```
# VexCode-2020
Codebase for the Vex 2020-2021 Change Up season
## Included Projects
Autonomous Period Simulator

Code Generator

Macros for Documentation

PID Debugger

Code Printing

Robot Code

Automated Scouting

## License
[gpl-3.0](https://opensource.org/licenses/lgpl-3.0.html)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jun  7 10:22:26 2019

@author: aiden
"""

import os
import treelib


class DirectoryTree:
    """
    contains methods for making a treelib tree structure of a
    given directory tree
    meant to be inherited but can be a stand alone class

    @params = None
    @return = None
    """

    def __init__(self):
        self.__directories = [] #will contain all directories and subdirectories
        self.__directory_tree = treelib.Tree() #will contain all directories ordered according
                                            #to "config.txt"



    def __get_dirs(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list given

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]

        for directory in dirs:
            storage_list.append(directory)
            self.__get_dirs(directory, storage_list)




    def __get_all(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list given

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]
        files = [f.path for f in os.scandir(parent_directory)]

        for file in files:
            storage_list.append(file)

        for directory in dirs:
            storage_list.append(directory)
            self.__get_all(directory, storage_list)



    def __make_tree(self):
        """
        add directories found to a treelib structure
        the parent directory is set in "config.txt"

        @params = None
        @return = None
        """
        depth = self.__directory_tree.depth()

        #splits path of directory and makes nodes of partial paths until
        #it reaches the end of the path given
        #the resulting tree is saved in "self.__directory_tree"
        for path in self.__directories:
            path_split = path.split("/")

            i = depth

            while i < len(path_split):
                try:
                    if i > 0:
                        name = path_split[i]
                        node_id = "/".join(path_split[0:i + 1])
                        parent = "/".join(path_split[0:i])
                    else:
                        name = path_split[i]
                        node_id = path_split[i]
                        parent = None

                    self.__directory_tree.create_node(tag=name, identifier=node_id, parent=parent)

                except treelib.exceptions.NodeIDAbsentError:
                    pass
                except treelib.exceptions.DuplicatedNodeIdError:
                    pass

                i = i + 1



    def return_directory_tree(self, parent_directory):
        """
        makes and returns a treelib tree structure of a directory
        tree based on a parent directory given only directories are
        included
```

```python

        @params = string of parent directory to start node at
        @return = treelib tree structure of directory tree,
                list of directories found
        """
        self.__directory_tree = treelib.Tree()
        self.__directories = []

        self.__get_dirs(parent_directory, self.__directories)
        self.__make_tree()

        return self.__directory_tree, self.__directories



    def return_tree(self, parent_directory):
        """
        makes and returns a treelib tree structure of a directory
        tree based on a parent directory given
        everything is included

        @params = string of parent directory to start node at
        @return = treelib tree structure of directory tree,
                list of directories found
        """
        self.__directory_tree = treelib.Tree()
        self.__directories = []

        self.__get_all(parent_directory, self.__directories)
        self.__make_tree()

        return self.__directory_tree, self.__directories
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jun  5 19:13:03 2019

@author: aiden
"""

import math
import os
import shutil


class PostScript:
    """
    class for converting code to pictures
    """
    def __init__(self):
        self.alldirs = [".."] #list declaration that will contain all
                              #directories. Parent directory is included in it
                              #because there is code to be printed in it
                              #
                              #list will contain all directories that contain
                              #files to be printed

        self.allfiles = [] #list declaration that will contain all files that
                           #must be converted to postscript

        self.pictures_dir = "../Pictures/" #parent directory of where pictures
                                           #will be placed

        self.pictures = [] #contains locations of pictures

        self.directory_exceptions = [ #directories that contain no files to
                                      #be printed
            "../AutonSimulator/__pycache__",
            "../CodeGenerator/__pycache__",
            "../Pictures",
            "../PIDDebugging/__pycache__",
            "../Print/__pycache__",
            "../Prototypes",
            "../RobotCode/bin",
            "../RobotCode/firmware",
            "../RobotCode/include/display",
            "../RobotCode/include/okapi",
            "../RobotCode/include/pros",
            "../RobotCode/lib",
            "../RobotCode/.d",
            "../RobotCode/src/JSONLibrary",
            "../RobotCode/src/objects/lcdCode/fonts",
            "../Scouting/__pycache__",
            "../Cascades",
            "../.git"
            ]

        self.file_exceptions = [  #files in included directories of acceptable
                                 #extensions that should not be converted to
                                 #postscript to be printed
            "../license.html",
            "../PIDDebugging/ut.txt",
            "../PIDDebugging/test.txt",
            "../Print/tree.txt",
            "../Print/tree.rtf",
            "../RobotCode/include/main.h",
            "../RobotCode/include/api.h",
            "../RobotCode/compile_commands.json",
            "../RobotCode/src/objects/lcdCode/DriverControl/logo.c"
            ]

        self.valid_extensions = [ #files with this extension will be allowed to
                                 #be printed
            ".py",
            ".c",
            ".cpp",
            ".hpp",
            ".h",
            ".sh",
            ".txt",
            ".json",
            ".md"
            ]

        if not os.path.isdir(self.pictures_dir):
            os.mkdir(self.pictures_dir)  #makes directory for pictures


    def __get_directories_recursively(self, parent_directory, storage_list):
        """
        recursively prints all files in parent directory
        put in list "alldirs"

        @params = directory to start at, directory to write output to
        @return = None
        """

        dirs = [f.path for f in os.scandir(parent_directory) if f.is_dir()]

        for directory in dirs:
            storage_list.append(directory)
            self.__get_directories_recursively(directory, storage_list)



    def __find_files(self, directory):
        """
        finds all files that are not subdirectories in a given directory that
        are of a specific type

        @params = directory to look through
        @return = type list of files that are in the directory that are
                  not sub directories
        """

        files = [] #contains all files that will be returned

        contents = [f.path for f in os.scandir(directory) if not f.is_dir()]
            #returns everything in directory including
            #sub directories

        for file in contents: #checks to make sure that only files that are not
                              #directories and have a valid extension are added
                              #to the list of files
            _, extension = os.path.splitext(file)
            if extension in self.valid_extensions:
                files.append(file)

        return files





    def __convert_to_postscript(self, file):
        """
        converts the specified file to postscript format which is a printable
        code type. The files are placed inside of "../pictures/" in the same
        directories that they were originally in

        @params = relative path of file to convert
        @return = None
        """

        _, extension = os.path.splitext(file)
        output_name = ((self.pictures_dir + file.split(extension)[0]
                       + extension.upper().split(".")[1]).replace("/../", "/")
                       + ".ps")

        self.pictures.append(output_name)

        #get the highlight color that will be used by shell command
        #based on the type of file it is
        if extension == ".py":
            highlight_color = "--highlight=python"

        elif extension in [".cpp", ".hpp"]:
            highlight_color = "--highlight=cpp"

        elif extension in [".c", ".h"]:
            highlight_color = "--highlight=c"

        elif extension == ".sh":
            highlight_color = "--highlight=bash"

        else:
            highlight_color = "--highlight=mail"

        #runs command that will convert a file to postscript
        #the output file location is in self.pictures
        os.system("enscript -G --line-numbers -o "
                  + output_name + " "
                  + highlight_color
                  + " --color=1 -f Palatino-Roman5 --columns 1 "
                  + file)


    def prepare_code(self):
        """
        prepares code to be printed by converting all code to
        postscript

        @params = None
        @return = None
        """
        self.allfiles = [] #list declaration that will contain all files that
                           #must be converted to postscript

        self.pictures_dir = "../Pictures/" #parent directory of where pictures
                                           #will be placed

        self.pictures = [] #contains locations of pictures
        shutil.rmtree(self.pictures_dir)  #clean out pictures folder so that
                                          #nothing is there that shouldn't be
        os.mkdir(self.pictures_dir)

        self.__get_directories_recursively("..", self.alldirs)
                             #gets unrefined list of directories
                             #saved in list "alldirs"
        to_remove = []
        for directory in self.alldirs: #iterates through each directory
                                       #found
            for exception in self.directory_exceptions: #checks to see if
                                                        #directory should
                                                        #be excluded
                if exception in directory:
                    to_remove.append(directory)  #do not remove items from list
                                                 #because changing the size
                                                 #while iterating through the
                                                 #list will cause
                                                 #unexpected results

                    break #end looking for exception because if one has
                          #been found it is already in the to remove list
                          #and no other cases will match it

        alldirs = set(self.alldirs) #convert each list to type set
                                    #and find the difference between them
                                    #the difference will always be the
                                    #directories wanted because all the items
                                    #in "to_remove" will be in "alldirs" and all
                                    #that is left is the directories to keep

        to_remove = set(to_remove)

        alldirs = list(alldirs - to_remove) #find difference of the two
                                            #sets and convert back to type
                                            #list remaining list will be
                                           #the directories that contain
                                           #code to be printed





        for directory in alldirs:
            self.allfiles.append(self.__find_files(directory))
                                                  #by adding
                                                  #the list
                                                  #directly to the all
                                                  #files list, list will
                                                  #be segmented and can
                                                  #be printed in order
                                                  #easier

            #make directory under pictures directory for a picture of the code
            #to go
            if not os.path.isdir((self.pictures_dir + directory).replace("/../", "/")):
                os.makedirs((self.pictures_dir + directory).replace("/../", "/"))

        for fileset in self.allfiles:
            for file in fileset:
                if not file in self.file_exceptions:
                    self.__convert_to_postscript(file)



    def get_page_count(self, files=None):
        """
        gets total size of all post script files in
        "Pictures" directory that was made on config file
        in "Pictures" directory

        @params = (optional) type list of paths of files to print
        @return = int amount of pages
        """
        if files:
            self.pictures = files

        elif not self.pictures and not files:
                              #if pictures list contains any file locations
                              #and if no files have been specified
                              #if it does not, then program will find all the
                              #files that are currently in the directory
                              #useful if user does not want to overwrite
                              #what is currently there

            self.pictures.append(self.pictures_dir) #add parent directory
                                                   #because it will not be
                                                   #there to start
            self.__get_directories_recursively(self.pictures_dir, self.pictures)

            self.valid_extensions.append(".ps") #temporarily add postscript to
                                               #acceptable extensions so that the
                                               #function used will not exclude it
                                               #entry is removed later
            picture_files = []
            for directory in self.pictures: #find all files that are not
                                           #subdirectories
                for file in self.__find_files(directory):
                    picture_files.append(file)

            self.valid_extensions.remove(".ps") #remove postscript from
                                               #acceptable extensions because
                                               #it is no longer needed

            self.pictures = picture_files #”self.pictures” now contains
                                         #locations of files

        total_pages = 0
        for picture in self.pictures: #iterate through and parse each file
                                      #looking for a keyword
                                      #keyword appears twice once with
                                      #parenthases and once without, so
                                      #algorithm makes sure it contains no
                                      #parenthases
            file = open(picture, "r")
            for line in file.readlines():
                if "%%Pages: " in line and "(" not in line:
                    total_pages = (total_pages + math.ceil(int(line.split("%%Pages: ")[1].split("\n")[0]) / 2))

        return total_pages
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jun  6 15:20:10 2019

@author: aiden
"""

import os
import treelib

import DirectoryTree


class Print(DirectoryTree.DirectoryTree): #inherits DirectoryTree so that
                                         #a directory tree can be made
    """
    contains methods for printing code based on
    "config.txt"
    """
    def __init__(self):
        DirectoryTree.DirectoryTree.__init__(self)

        self.directory_tree = treelib.Tree() #will contain all directories ordered according
                                            #to "config.txt"

        self.swap_tree = treelib.Tree() #used to re-arrange directories in
                                       #other node

        self.parent = "." #will contain parent to append to file paths
                           #in "config.txt"
        self.headers_only = 0 #option set in "config.txt"
        self.headers_first = 1 #option set in "config.txt"

        self.dir_order = [] #will contain order of directories set in "config.txt"
        self.file_order = [] #will contain order of files set in "config.txt"

        self.print_order = [] #will contain paths of all print items in order




    @classmethod
    def __get_file_type(cls, path):
        """
        takes a file converted to postscript and finds what file type it
        is (.cpp, .hpp, .sh)

        @params = path name of file to get the type of
        @return = type string of extension (ex. ".cpp")
                  type string of root path (ex. no directories and file type (CPP) cut off)
                  type string of path without root path
        """
        valid_extensions = [ #files with this extension will be allowed to
                             #be printed
            "PY",
            "C",
            "CPP",
            "HPP",
            "H",
            "SH",
            "TXT",
            "JSON",
            "MD"
            ]

        filename, _ = os.path.splitext(path) #remove extension
        filename = filename.split("/") #remove other directories
        filename = filename[len(filename) - 1]

        one_char = filename[-1:]
        two_char = filename[-2:]
        three_char = filename[-3:]
        four_char = filename[-4:]

        root_name = ""

        if one_char in valid_extensions and two_char not in valid_extensions:
            #added checking for two char as well to fix cases SH and H
            extension = one_char
        elif two_char in valid_extensions:
            extension = two_char
        elif three_char in valid_extensions:
            extension = three_char
        elif four_char in valid_extensions:
            extension = four_char
        else:
            extension = "invalid"

        if extension != "invalid":
            root_name = filename.split(extension)[0]

        root_path = path.split(root_name + extension)[0]

        return extension, root_name, root_path




    def __get_rules(self):
        """
        gets rules based on "config.txt"

        @params = None
        @return = None
        """
        #extract parent from "config.txt"
        with open("config.txt", "r") as config: #file closed at end of block
            for line in config.readlines():
                #makes sure all required elements are in the line
                #and that it is not a comment
                if "parent" in line and "=" in line and "#" not in line:
                    self.parent = line.split("parent")[1].strip().strip('\"')
                    self.parent = self.parent.split("=")[1].strip().strip('\"')

        #extract other parameters from "config.txt"
        param_names = ["HEADERS_FIRST", "ONLY_HEADERS"]
        with open("config.txt", "r") as config: #file closed at end of block
            for line in config.readlines():
                #makes sure all required elements are in the line
                #and that it is not a comment
                if param_names[0] in line and "#" not in line:
                    line = line.split(param_names[0])[1].strip()
                    self.headers_first = int(line)

                elif param_names[1] in line and "#" not in line:
                    line = line.split(param_names[1])[1].strip()
                    self.headers_only = int(line.strip())

        #read rules
        with open("config.txt", "r") as config: #file closed at end of block
            for line in config.readlines():
                if '#' not in line:
                    if "dir" in line:
                        #convert to standard os path
                        directory = self.parent + "/" + (line.split("dir ")[1].rstrip())
                        directory = os.path.normpath(directory)
                        self.dir_order.append(directory)
                    elif "file" in line:
                        #convert to standard os path
                        file = self.parent + "/" + (line.split("file ")[1].rstrip())
                        file = os.path.normpath(file)
                        self.file_order.append(file)




    def __order_directories(self):
        """
        orders directories in "self.directory_tree"
        based on settigns in "config.txt"
        makes new tree stored in "self.swap_tree"

        @params = None
        @return = None
        """
        #iterate through each branch of the tree and sort it
        #moving directories to needed spot in "self.swap_tree"
        #standard path will be made from os module
        #so that no conflicts occur from directory separators
        #a two number string is added to all the tags in the node so that
        #the order is maintained. The order for nodes is in
        #alphabetical order, so no changes would occur if the numbers were
        #not added

        #finds nodes in tree that have children
        parent_nodes = []
        for node in self.directory_tree.all_nodes():
            if not node.is_leaf(): #a leaf has no children
                parent_nodes.append(node)

        #adds the root node to the swap tree
        self.swap_tree.create_node("00" + parent_nodes[0].tag,
                                   parent_nodes[0].identifier)
        parent_nodes.pop(0)

        #sorts childrent of parent node
        #
        #a string of two numbers is added to the tag so that the tree is
        #in order, the id is not affected however
        #this means the max amount of directories in one node can only be 99
        #this should probably not be exceded
        for node in parent_nodes:
            num = 0
            children = self.directory_tree.children(node.identifier)
            try: #adds parent of the node to the tree if it is not already there
                self.swap_tree.create_node("00" + node.tag,
                                           node.identifier,
                                           self.directory_tree.parent(node.identifier).identifier)
            except treelib.exceptions.DuplicatedNodeIdError: #node already exists
                pass

            #get applicable rules
            applicable_rules = [] #contains identifier for nodes in order to
                                  #be sorted
            for rule in self.dir_order:
                dir_length = len(rule.split("/")) #checks to see if amount of
                                                  #directories in path is the
                                                  #same as the amount in the
                                                  #tree
                                                  #if it is, it is added to
                                                  #applicable rules

                offset = len(self.parent.split("/")) #offset is added to the
                                                     #tree level because the
                                                     #rules in "config.txt" do
                                                     #not contain the parent
                                                     #directory
                children_names = list(map(lambda x: x.identifier, children))
                tree_level = self.directory_tree.level(node.identifier) + offset
                if rule in children_names and dir_length == tree_level:
                    applicable_rules.append(rule)

            for child in applicable_rules: #add nodes to the tree that appear
                                           #in the given rules so that the order
                                           #wanted is kept
                num = str(num) #makes sure that "num" is two characters
                if len(num) < 2:
                    num = "0" + str(num)

                nid = self.directory_tree.get_node(child)
                self.swap_tree.create_node(str(num) + nid.tag,
                                           nid.identifier,
                                           node.identifier)
                num = int(num) + 1
                children.remove(self.directory_tree.get_node(child))
            while children: #adds other children to the new tree
                            #order does not matter so they are added in order
                            #of appearance
                num = str(num) #makes sure that "num" is two characters
                if len(num) < 2:
                    num = "0" + num

                self.swap_tree.create_node(str(num) + children[0].tag,
                                           children[0].identifier,
                                           parent=node.identifier)
                num = int(num) + 1
                children.pop(0)




    def __order_headers(self, files):
        """
        takes a list of postscript files and orders it so that headers
        appear before implementation files in the list

        @params = list of postscript files to sort putting headers before
                  implementation file
        @return = sorted list of postscript files
        """
        #create dictionary with key of the root file name and then a value of
        #a list of the extensions with that root name
        file_dict = {}

        for file in files:
            extension, root_name, root_path = self.__get_file_type(file)
            if root_name not in file_dict.keys():
                file_dict.update({root_name : [extension]})
            else: #if key exists then add value to the list
                file_dict[root_name].append(extension)

        #iterate through dictionary placing hpp files ahead of cpp
        for extension_list in file_dict.values():
            extension_list.sort(reverse=self.headers_first)

        #re construct list
        ordered_files = []
        for item in file_dict:
            extensions = file_dict.get(item)
            for extension in extensions:
                if extension != "invalid":
                    if not self.headers_only:
                        file = root_path + item + extension + ".ps"
                        ordered_files.append(file)
                    else: #if only headers are to be printed
                        if extension != "CPP" or len(extensions) == 1:
                            file = root_path + item + extension + ".ps"
                            ordered_files.append(file)

        return ordered_files




    def __order_files(self):
        """
        creates a list of file names in the order that they are to be printed

        @params = None
        @return = None
        """
        #get directories to look through
        directories = list(self.swap_tree.expand_tree())
        ordered_files = [] #holds list of all ordered files

        for directory in directories:
            ordered_directory = [] #holds list of ordered files in a directory

            #gets the files in a given directory
            files = list([f.path for f in os.scandir(directory) if not f.is_dir()])
            files.sort() #makes paths in aphabetical order

            #gets rules for files in a directory
            applicable_rules = []

            for rule in self.file_order:
                rule_directory = rule
                rule_directory = rule_directory.split("/")
                rule_directory.pop(-1)
                rule_directory = "/".join(rule_directory)

                rule_dirs = len(rule.split("/")) - 1

                #add one to the tree level because the level starts at 0 instead
                #of 1 like the method to find the height of other directories
                tree_level = self.swap_tree.level(self.swap_tree.get_node(directory).identifier) + 1

                #if rule applies
                if rule_directory == directory and rule_dirs == tree_level:
                    applicable_rules.append(rule)

            #adds files where a specified order has been given
            for file in applicable_rules:
                ordered_directory.append(file)
                files.remove(file)

            #adds rest of files
            while files:
                ordered_directory.append(files[0])
                files.pop(0)

            #sorts so that headers are first if that option
            #has been given
            #also checks if only headers will be printed
            ordered_directory = self.__order_headers(ordered_directory)

            #adds item in the ordered files in a directory to all the ordered
            #files
            for item in ordered_directory:
                ordered_files.append(item)


        #adds items in ordered_files to the final print order if the
        #extension is .ps
        for item in ordered_files:
            _, extension = os.path.splitext(item)
            if extension == ".ps":
                self.print_order.append(item)



    def order(self):
        """
        orders code based on "config.txt" with all the parameters set in
        it

        @params = None
        @return = type list of files to print in order
        """
        #reset attributes
        self.directory_tree = treelib.Tree() #will contain all directories ordered according
                                            #to "config.txt"
        self.swap_tree = treelib.Tree() #used to re-arrange directories in
                                       #other node
        self.headers_only = 0 #option set in "config.txt"
        self.headers_first = 1 #option set in "config.txt"
        self.dir_order = [] #will contain order of directories set in "config.txt"
        self.file_order = [] #will contain order of files set in "config.txt"
        self.print_order = [] #will contain paths of all print items in order

        #order code functions
        self.__get_rules()
        self.directory_tree, _ = self.return_directory_tree(self.parent)

        self.__order_directories()
        self.__order_files()

        return self.print_order




    def print_code(self, files=None):
        """
        prints code by file and also barriers if they are in the list

        @params = (optional) type list of path of files to print
        @return = None
        """

        if files:
            self.print_order = files

        for file in self.print_order: #adds print job to the printer queue
                                     #with shell command
            print("queuing ", file)
            command = "lpr -P HP-Color-LaserJet-M553 -o sides=two-sided-long-edge " + str(file)
            os.system(command)

    def output_pdf(self, files=None):
        """
        merges code by file and also barriers if they are in the list

        @params = (optional) type list of path of files to print
        @return = None
        """

        if not files:
            files - self.print_order

        command = "psmerge " + " ".join(files) + " > all.ps; ps2pdf all.ps; rm all.ps"
        print(command)
        #os.system(command)

```

../Print/config.txt

```
#contains rules for print order
#all files are printed alphabetically unless otherwise specified

#parent will specify which directory to start at. It should be set to where pictures are
parent = ../Pictures


#for rules: algorithm first finds all directories and sorts them according
#to rules. Directory rules starts with "dir" (can be in any order).

#Once all directories and their subdirectories are sorted, algorithm
#will look for individual files rules specifies with "file"

#formatting will look best as a directory tree
#only one rule per line

#order for print goes directory, files in that directory, subdirectory,
#files in the subdirectories of that parent


#set to "1" if user wants headers to be printed before implementation files
HEADERS_FIRST 1

#set to "1" if user only wants header files to be printed to save paper
#this will only keep implementation files that have no header
ONLY_HEADERS 0


#rules

#for rules involving headers and implementation files
#specify both the HPP and CPP file
#by setting HEADERS_FIRST the HPP file will be printed first
#even if the CPP file is listed first

dir Print
dir PIDDebugging
    file PIDDebugging/mainPY.ps
dir Prototypes
dir DocumentationMacros
dir CodeGenerator
    file CodeGenerator/code_genPY.ps
    file CodeGenerator/cpp_typesPY.ps
    file CodeGenerator/exceptionsPY.ps
    file CodeGenerator/configPY.ps
dir AutonSimulator
    file AutonSimulator/mainPY.ps
dir Scouting
    file Scouting/ReadmeMD.ps
dir RobotCode
    file RobotCode/lcdTestSH.ps
    file RobotCode/configJSON.ps
    file RobotCode/stacktraceSH.ps
    dir RobotCode/include
    dir RobotCode/src
        file RobotCode/src/mainCPP.ps
        dir RobotCode/src/objects/controller
        dir RobotCode/src/objects/motors
        dir RobotCode/src/objects/sensors
        dir RobotCode/src/objects/robotChassis
        dir RobotCode/src/objects/lift
        dir RobotCode/src/objects/tilter
        dir RobotCode/src/objects/writer
        dir RobotCode/src/objects/lcdCode
            file RobotCode/src/objects/lcdCode/Debug/DebugHPP.ps
            file RobotCode/src/objects/lcdCode/Debug/DebugCPP.ps
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 11 14:26:00 2019

@author: aiden
"""

import copy
import os

import PostScript
import Print
import DirectoryTree


print("starting print process")
print("please be very careful because there is a lot of code")
print()


help_dict = {
    "help":"print help",
    "postscript":"convert files to postscript",
    "page_count":"prints amount of pages that code will be",
    "order":"prepare order to print in",
    "show_order":"shows current order of files to be printed in",
    "tree":"add cover sheet of tree to files",
    "print":"prints code",
    "pdf":"converts all code into one large pdf",
    "cancel":"cancels all printer jobs",
    "status":"shows current print jobs",
    "exit":"ends session"
    }


ps = PostScript.PostScript()
printer = Print.Print()
Tree = DirectoryTree.DirectoryTree()


print_order = []


while 1:
    command = input("enter command ")

    if command.upper() == "HELP":
        print()

        for item in help_dict:
            string = item
            while len(string) < 20: #aligns help options with each other
                string = string + "."
            print(string, help_dict.get(item), sep=".")


    elif command.upper() == "POSTSCRIPT":
        ps.prepare_code()

    elif command.upper() == "PAGE_COUNT":
        if not print_order:
            pages = ps.get_page_count()
        else:
            pages = ps.get_page_count(print_order)

        print("number of pages: ", pages)

    elif command.upper() == "ORDER":
        if len(print_order) > 1:
            print_order = []

        to_add = printer.order()

        for item in to_add:
            print_order.append(item)
        print()
        print("please review this order carefully before printing")
        print("do not waste paper")
        print()

        for item in print_order:
            print(item)


    elif command.upper() == "SHOW_ORDER":
        for item in print_order:
            print(item)


    elif command.upper() == "TREE":
        file_tree, _ = Tree.return_tree("..")
        filtered_tree = copy.deepcopy(file_tree)

        #sorts out nodes that are not being printed in three ways
        #Tree is copies to a new tree so that the tree does not
        #change during iteration
        #original is looked through and changes are made to the copy
        for node in file_tree.all_nodes():
            nid = node.identifier
            if os.path.isdir(nid):  #sort out nodes by directory
                if nid in ps.directory_exceptions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)
            else:
                _, extension = os.path.splitext(nid)

                #sort out nodes by file
                if nid in ps.file_exceptions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)

                #sort out nodes by extension
                elif extension not in ps.valid_extensions and filtered_tree.get_node(nid):
                    filtered_tree.remove_node(nid)

        filtered_tree.get_node("..").tag = "VexCode-2019"
        filtered_tree.show()
        os.remove("tree.rtf")

        filtered_tree.save2file("tree.rtf")
        filtered_tree.save2file("tree.txt")
        print_order.insert(0, "tree.rtf")


    elif command.upper() == "PRINT":
        printer.print_code(print_order)

    elif command.upper() == "PDF":
        printer.output_pdf(print_order)

    elif command.upper() == "CANCEL":
        os.system("lprm -P HP-Color-LaserJet-M553 -")

    elif command.upper() == "STATUS":
        os.system("lpstat -P HP-Color-LaserJet-M553")

    elif command.upper() == "EXIT":
        break

    else:
        print("invalid command")

    print()

```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  5 17:13:31 2020

@author: aiden
"""
import streamlit as st

import data_parser
import graph

# if len(sys.argv) == 1:
#     file = input("enter file to parse: ")
# else:
#     file = sys.argv[1]
file = "./log.txt"

# parser.gen_sample_data()
p = data_parser.Parser()
p.parse_file(file)
p.print_data()
g = graph.DebugGraph(
    p.get_data(),
    {
        "kP":p.get_data()["pid_constants"]["kP"],
        "kI":p.get_data()["pid_constants"]["kI"],
        "kD":p.get_data()["pid_constants"]["kD"],
        "I_max":p.get_data()["pid_constants"]["I_max"],
        "brakemode":p.get_data()["brakemode"],
        "gearset":p.get_data()["gearset"],
        "slew":p.get_data()["slew_rate"]
    }
)
y1 = st.sidebar.selectbox("Y1 data", ["velocity", "voltage", "heading", "position", "integral", "correction", "acceleration"], 1)
track_y1 = st.sidebar.checkbox("Graph Y1 setpoint")
y2 = st.sidebar.selectbox("Y2 data", ["velocity", "voltage", "heading", "position", "None"], 4)
track_y2 = st.sidebar.checkbox("Graph Y2 setpoint")
plot = g.make_graph(y1, y2, track_y1, track_y2)
st.plotly_chart(plot, use_container_width=True)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan  5 15:35:01 2020

@author: aiden
"""
import math
import random
import os


class Parser:
    """
    parses data from motors so that it can be graphed
    """
    def __init__(self):
        self.__voltage_data = {
            "back_right":[],
            "back_left":[],
            "front_right":[],
            "front_left":[]
        }
        self.__velocity_data = {
            "back_right":[],
            "back_left":[],
            "front_right":[],
            "front_left":[]
        }
        self.__time_data = []
        self.__integral_data = []
        self.__vel_setpoint_data = []
        self.__heading_sp_data = []
        self.__heading_data = []
        self.__position_sp = []
        self.__position_l_data = []
        self.__position_r_data = []

        self.__correction_data = []

        self.__brakemode = None
        self.__gearset = None
        self.__slew = 0
        self.__pid = {
            "kP":0,
            "kI":0,
            "kD":0,
            "I_max":0
            }

        self.__brakemode_names = {
            0:"Coast",
            1:"Brake",
            2:"Hold",

        }

        self.__gearset_names = {
            0:"36:1",
            1:"18:1",
            2:"6:1"
        }


    def __get_data_point(self, line):
        """
        parses a line from the file and returns the data of voltage,
        velocity, integral value, and time in the form of a dictionary

        sample line:
            [INFO] Motor 1, Brakemode: xxxx, Actual_Voltage: xxx, ...
        """
        try:
            line = line.split("[INFO]")[1]
            data = line.split(",")
        except IndexError:  #ensures that line is an actual data line
            return 0
        try:
            voltage1 = [ item.strip().split("Actual_Vol1")[1].split(": ")[1] for item in data if "Actual_Vol1:" in item ]
            voltage2 = [ item.strip().split("Actual_Vol2")[1].split(": ")[1] for item in data if "Actual_Vol2:" in item ]
            voltage3 = [ item.strip().split("Actual_Vol3")[1].split(": ")[1] for item in data if "Actual_Vol3:" in item ]
            voltage4 = [ item.strip().split("Actual_Vol4")[1].split(": ")[1] for item in data if "Actual_Vol4:" in item ]

            velocity1 = [ item.strip().split("Actual_Vel1")[1].split(": ")[1] for item in data if "Actual_Vel1:" in item ]
            velocity2 = [ item.strip().split("Actual_Vel2")[1].split(": ")[1] for item in data if "Actual_Vel2:" in item ]
            velocity3 = [ item.strip().split("Actual_Vel3")[1].split(": ")[1] for item in data if "Actual_Vel3:" in item ]
            velocity4 = [ item.strip().split("Actual_Vel4")[1].split(": ")[1] for item in data if "Actual_Vel4:" in item ]

            time = [ item.strip().split("Time:")[1] for item in data if "Time:" in item ]
            integral = [ item.strip().split("I:")[1] for item in data if "I:" in item ]
            vel_sp = [ item.strip().split("Vel_Sp:")[1] for item in data if "Vel_Sp:" in item ]
            heading_sp = [ item.strip().split("Heading_Sp:")[1] for item in data if "Heading_Sp:" in item ]
            relative_heading = [ item.strip().split("Relative_Heading:")[1] for item in data if "Relative_Heading:" in item ]
            position_sp = [ item.strip().split("Position_Sp:")[1] for item in data if "Position_Sp:" in item ]
            position_r = [ item.strip().split("position_r:")[1] for item in data if "position_r:" in item ]
            position_l = [ item.strip().split("position_l:")[1] for item in data if "position_l:" in item ]

            correction = [ item.strip().split("Correction:")[1] for item in data if "Correction:" in item ]

            data_dict = {
                "voltage1":float(voltage1[0].strip()),
                "voltage2":float(voltage2[0].strip()),
                "voltage3":float(voltage3[0].strip()),
                "voltage4":float(voltage4[0].strip()),
                "velocity1":float(velocity1[0].strip()),
                "velocity2":float(velocity2[0].strip()),
                "velocity3":float(velocity3[0].strip()),
                "velocity4":float(velocity4[0].strip()),
                "time":float(time[0].strip()),
                "integral":float(integral[0].strip()),
                "heading_sp":float(heading_sp[0].strip()),
                "relative_heading":float(relative_heading[0].strip()),
                "position_sp":float(position_sp[0].strip()),
                "position_r_data":float(position_r[0].strip()),
                "correction":float(correction[0].strip()),
                "position_l_data":float(position_l[0].strip())
                }
        except:
                return 0

        try:
            data_dict.update({"vel_setpoint":float(vel_sp[0].strip())})
        except IndexError:
            data_dict.update({"vel_setpoint":[]})


        return data_dict
        # except IndexError:
        #     return 0


    def parse_file(self, file):
        """
        parses a file line by line and adds data to list
        """
        is_first_line = True
        with open(file) as f:
            #find first valid line
            for line in f:
                if is_first_line:
                    data = self.__get_data_point(line)
                    print(data)
                    if data:
                        self.__voltage_data["back_right"].append(data.get("voltage4"))
                        self.__voltage_data["back_left"].append(data.get("voltage3"))
                        self.__voltage_data["front_right"].append(data.get("voltage2"))
                        self.__voltage_data["front_left"].append(data.get("voltage1"))

                        self.__velocity_data["back_right"].append(data.get("velocity4"))
                        self.__velocity_data["back_left"].append(data.get("velocity3"))
                        self.__velocity_data["front_right"].append(data.get("velocity2"))
                        self.__velocity_data["front_left"].append(data.get("velocity1"))

                        self.__time_data.append(data.get("time"))
                        self.__integral_data.append(data.get("integral"))
                        self.__vel_setpoint_data.append(data.get("vel_setpoint"))
                        self.__heading_sp_data.append(data.get("heading_sp"))
                        self.__heading_data.append(data.get("relative_heading"))
                        self.__position_sp.append(data.get("position_sp"))
                        self.__position_r_data.append(data.get("position_r_data"))
                        self.__position_l_data.append(data.get("position_l_data"))

                        self.__correction_data.append(data.get("correction"))

                        first_line = line.split("[INFO]")[1]
                        data = first_line.split(",")

                        self.__brakemode = int([ item.strip().split("Brake:")[1].strip() for item in data if "Brake:" in item ][0])
                        self.__gearset = int([ item.strip().split("Gear:")[1].strip() for item in data if "Gear:" in item ][0])
                        self.__slew = int([ int(item.strip().split("Slew:")[1].strip()) for item in data if "Slew:" in item ][0])
                        # self.__slew = 120


                        self.__brakemode = self.__brakemode_names.get(self.__brakemode, "???")
                        self.__gearset = self.__gearset_names.get(self.__gearset, "???")

                        self.__pid["kP"] = float([ float(item.strip().split("kP:")[1].strip()) for item in data if "kP:" in item ][0])
                        self.__pid["kI"] = float([ float(item.strip().split("kI:")[1].strip()) for item in data if "kI:" in item ][0])
                        self.__pid["kD"] = float([ float(item.strip().split("kD:")[1].strip()) for item in data if "kD:" in item ][0])
                        self.__pid["I_max"] = float([ float(item.strip().split("I_max:")[1].strip()) for item in data if "I_max:" in item ][0])
                        is_first_line = False
                    continue


                data = self.__get_data_point(line)
                if data:
                    self.__voltage_data["back_right"].append(data.get("voltage4"))
                    self.__voltage_data["back_left"].append(data.get("voltage3"))
                    self.__voltage_data["front_right"].append(data.get("voltage2"))
                    self.__voltage_data["front_left"].append(data.get("voltage1"))

                    self.__velocity_data["back_right"].append(data.get("velocity4"))
                    self.__velocity_data["back_left"].append(data.get("velocity3"))
                    self.__velocity_data["front_right"].append(data.get("velocity2"))
                    self.__velocity_data["front_left"].append(data.get("velocity1"))

                    self.__time_data.append(data.get("time"))
                    self.__integral_data.append(data.get("integral"))
                    self.__vel_setpoint_data.append(data.get("vel_setpoint"))
                    self.__heading_sp_data.append(data.get("heading_sp"))
                    self.__heading_data.append(data.get("relative_heading"))
                    self.__position_sp.append(data.get("position_sp"))
                    self.__position_r_data.append(data.get("position_r_data"))
                    self.__position_l_data.append(data.get("position_l_data"))
                    self.__correction_data.append(data.get("correction"))


    def print_data(self):
        """
        prints data

        useful for debugging
        """
        print("\nVoltage Data:", len(self.__voltage_data), "data points")
        for item in self.__voltage_data:
            print(item)

        print("\nVelocity Data:", len(self.__velocity_data), "data points")
        for item in self.__velocity_data:
            print(item)

        print("\nIntegral Data:", len(self.__integral_data), "data points")
        for item in self.__integral_data:
            print(item)


        print("\nTime Data:", len(self.__time_data), "data points")
        for item in self.__time_data:
            print(item)

        print("\nVelocity Setpoint Data:", len(self.__vel_setpoint_data), "data points")
        for item in self.__vel_setpoint_data:
            print(item)

        print("\nHeading Setpoint Data:", len(self.__heading_sp_data), "data points")
        for item in self.__heading_sp_data:
            print(item)

        print("\Relative Heading Data:", len(self.__heading_data), "data points")
        for item in self.__heading_data:
            print(item)

        print("\Position Setpoint:", len(self.__position_sp), "data points")
        for item in self.__position_sp:
            print(item)

        # print("\Position Data:", len(self.__position_data), "data points")
        # for item in self.__position_data:
        #     print(item)


        print("\nPID constants:")
        for key in self.__pid:
            print(key, ":", self.__pid[key])


        print("\nBrakemode: ", self.__brakemode)
        print("Gearset: ", self.__gearset)
        print("Slew Rate: ", self.__slew)


    def get_data(self):
        """
        returns a dictionary of the data that was parsed
        """
        data = {
            "voltage":self.__voltage_data,
            "velocity":self.__velocity_data,
            "integral":self.__integral_data,
            "vel_setpoint":self.__vel_setpoint_data,
            "heading_setpoint":self.__heading_sp_data,
            "heading_data":self.__heading_data,
            "position_sp":self.__position_sp,
            "position_r":self.__position_r_data,
            "position_l":self.__position_l_data,
            "time":self.__time_data,
            "brakemode":self.__brakemode,
            "gearset":self.__gearset,
            "slew_rate":self.__slew,
            "correction":self.__correction_data,
            "pid_constants":self.__pid
        }

        return data



def gen_sample_data(num_data_pts=1000, setpoint=100, file="ut.txt"):
    """
    makes a random set of data that can be used for a unit test
    """
    if os.path.isfile(file):
        os.remove(file)
    f = open(file, "a")

    step = setpoint / num_data_pts #setpoint / ...
    min_vel = 0
    for i in range(num_data_pts):
        vel = (random.randint(int(min_vel), int(min_vel + step)) ** 1/((i+setpoint)/num_data_pts)) + setpoint/10
        vol = ((((vel + 200) * (12000 + 12000)) / (200 + 200)) - 12000) + random.randint(-600, 600); #scale vel to voltage range and add jitter
        data = "[INFO] Motor 1, Actual_Vol: " + str(vol)
        data += ", Brake: 1, Gear: 1, I_max: 1000, I: 100, kD: 0, kI: 0, "
        data += "kP: 1.0, Slew: 40, Time: " + str(i)
        data += ", Heading_Sp: 0.0, Relative_Heading: " + str((0 + (random.randrange(-100, 100, 1) / 100)))
        data += ", Position_Sp: 1000, Position: 1000"
        data += ", Vel_Sp: " + str(setpoint) + ", Vel: " + str(vel) + "\n"

        f.write(data)

        min_vel += step


    f.close()

# unit test

# gen_sample_data(file="test.txt")
# P = Parser()
# P.parse_file("test.txt")
# P.print_data()


```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 29 12:38:16 2019

@author: aiden
"""
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import numpy as np
import math

class DebugGraph:
    """
    class for making a graph for debugging PID data
    """
    def __init__(self, data_dict, parameters):
        self.time_data = data_dict.get("time")
        self.vel_data = data_dict.get("velocity")
        self.vol_data = data_dict.get("voltage")
        self.vel_sp = data_dict.get("vel_sp")
        self.heading_sp = data_dict.get("heading_sp")
        self.heading_data = data_dict.get("heading_data")
        self.position_sp = data_dict.get("position_sp")
        self.position_r_data = data_dict.get("position_r")
        self.position_l_data = data_dict.get("position_l")
        self.integral_data = data_dict.get("integral")
        self.correction_data = data_dict.get("correction")

        #add legend for constants
        self.constants_text = "kP : " + str(parameters.get("kP")) + "\n"
        self.constants_text += "kI : " + str(parameters.get("kI")) + "\n"
        self.constants_text += "kD: " + str(parameters.get("kD")) + "\n"
        self.constants_text += "Integral Max: " + str(parameters.get("I_max")) + "\n"
        self.constants_text += "\n"
        self.constants_text += "Brakemode: " + str(parameters.get("brakemode")) + "\n"
        self.constants_text += "Gearset: " + str(parameters.get("gearset")) + "\n"
        self.constants_text += "Slew Rate (mV/ms): " + str(parameters.get("slew")) + "\n"

    def __to_in_per_sec2(self, degrees_per_ms2, wheel_diameter=3.25):
        circumference = math.pi * wheel_diameter
        revolutions_per_ms2 = degrees_per_ms2 / 360
        linear_distance = revolutions_per_ms2 * circumference
        in_per_sec2 = linear_distance * 1000  # convert ms to sec
        return in_per_sec2

    def make_graph(self, y1, y2=None, track_y1_sp=True, track_y2_sp=True):
        """
        returns a graph object with given axis parameters
        """
        title = "PID Tuning - "
        x = self.time_data
        if y1 == "velocity":
            y1_data = [self.vel_data.get("back_right"), self.vel_data.get("front_right"), self.vel_data.get("back_left"), self.vel_data.get("front_left")]
            y1_sp = self.vel_sp
            y1_title = "Velocity (RPM)"
            name1 = ["Back Right Velocity", "Front Right Velocity", "Back Left Velocity", "Front Left Velocity"]
            title += "Velocity"
        elif y1 == "voltage":
            y1_data = [self.vol_data.get("back_right"), self.vol_data.get("front_right"), self.vol_data.get("back_left"), self.vol_data.get("front_left")]
            y1_sp = []
            y1_title = "Voltage (mV)"
            name1 = ["Back Right Voltage", "Front Right Voltage", "Back Left Voltage", "Front Left Voltage"]
            title += "Voltage"
        elif y1 == "heading":
            y1_data = [self.heading_data]
            y1_sp = self.heading_sp
            y1_title = "Relative Heading (Degrees)"
            name1 = ["Relative Heading of Robot"]
            title += "Relative Heading of Robot"
        elif y1 == "position":
            y1_data = [self.position_l_data, self.position_r_data]
            y1_sp = self.position_sp
            y1_title = "Position"
            name1 = ["Position of Right Sensor", "Position of Left Sensor"]
            title += "Position of Sensor"
        elif y1 == "integral":
            y1_data = [self.integral_data]
            y1_title = "Integral"
            y1_sp = []
            name1 = ["Integral Value"]
            title += "Integral"
        elif y1 == "correction":
            y1_data = [self.correction_data]
            y1_title = "correction"
            y1_sp = []
            name1 = ["Correction"]
            title += "Correction"
        elif y1 == "acceleration":
            vel_data_l = np.diff(self.position_l_data) / np.diff(x)
            vel_data_r = np.diff(self.position_r_data) / np.diff(x)
            accel_data_l = []
            accel_data_r = []
            for l, r in zip(np.diff(vel_data_l) / np.diff(x[:-1]), np.diff(vel_data_r) / np.diff(x[:-1])):
                accel_data_l.append(self.__to_in_per_sec2(l))
                accel_data_r.append(self.__to_in_per_sec2(r))
            y1_data = [accel_data_l, accel_data_r]
            x = x[:-2]
            y1_title = "Acceleration"
            y1_sp = []
            name1 = ["Acceleration of Right Sensor", "Acceleration of Left Sensor"]
            title += "Acceleration"
        else:
            y1_data = []
            y1_sp = []
            y1_title = ""
            name1 = ""


        if y2 == "velocity":
            y2_data = [self.vel_data.get("back_right"), self.vel_data.get("front_right"), self.vel_data.get("back_left"), self.vel_data.get("front_left")]
            y2_sp = self.vel_sp
            y2_title = "Velocity (RPM)"
            name2 = ["Back Right Velocity", "Front Right Velocity", "Back Left Velocity", "Front Left Velocity"]
            title += "Velocity"
        elif y2 == "voltage":
            y2_data = [self.vol_data.get("back_right"), self.vol_data.get("front_right"), self.vol_data.get("back_left"), self.vol_data.get("front_left")]
            y2_sp = []
            y2_title = "Voltage (mV)"
            name2 = ["Back Right Voltage", "Front Right Voltage", "Back Left Voltage", "Front Left Voltage"]
            title += "Voltage"
        elif y2 == "heading":
            y2_data = [self.heading_data]
            y2_sp = self.heading_sp
            y2_title = "Relative Heading (Degrees)"
            name2 = ["Relative Heading of Robot"]
            title += "Relative Heading of Robot"
        elif y2 == "position":
            y2_data = [self.position_l_data, self.position_r_data]
            y2_sp = self.position_sp
            y2_title = "Position"
            name2 = ["Position of Right Sensor", "Position of Left Sensor"]
            title += "Position of Sensor"
        elif y2 == "integral":
            y2_data = [self.integral_data]
            y2_sp = []
            name2 = ["Integral Value"]
            title += "Integral"
        else:
            y2_data = []
            y2_sp = []
            y2_title = ""
            name2 = ""

        if y2_data:
            plot = make_subplots(specs=[[{"secondary_y": True}]])
        else:
            plot = go.Figure()

        for data, name in zip(y1_data, name1):
            plot.add_trace(
                go.Scatter(
                    x=x,
                    y=data,
                    mode="lines",
                    line={'dash': 'solid', 'color': 'blue'},
                    name=name,
                    yaxis="y1",
                )
            )
        if track_y1_sp:
            plot.add_trace(
                go.Scatter(
                    x=x,
                    y=y1_sp,
                    mode="lines",
                    line={'dash': 'dash', 'color': 'blue'},
                    name="Setpoint",
                    yaxis="y1",
                )
            )


        if y2_data:
            for data, name in zip(y2_data, name2):
                plot.add_trace(
                    go.Scatter(
                        x=x,
                        y=data,
                        mode="lines",
                        line={'dash': 'solid', 'color': 'green'},
                        name=name,
                    ),
                    secondary_y=True
                )
            if track_y2_sp:
                plot.add_trace(
                    go.Scatter(
                        x=x,
                        y=y2_sp,
                        mode="lines",
                        line={'dash': 'dash', 'color': 'green'},
                        name="Setpoint",
                    ),
                    secondary_y=True
                )

        title += " vs Time"
        plot.update_layout(
            title=title,
            xaxis=dict(
                title="Time (ms)"
            ),
            font=dict(
                family="Courier New, monospace",
                size=14,
                color="#7f7f7f"
            ),
            showlegend=True,
            legend=dict(x=1.2, y=1.1),
            plot_bgcolor="rgb(255, 255, 255)"
        )


        plot.update_xaxes(
            tickangle=-45,
            showgrid=False,
            mirror=True,
            ticks='outside',
            showline=True

        )
        plot.update_yaxes(
            tickangle=0,
            showgrid=False,
            zeroline=True,
            zerolinewidth=2,
            zerolinecolor="black",
            mirror=True,
            ticks=’outside’,
            showline=True

        )

        if y2_data:
            plot.update_yaxes(title_text=y1_title, secondary_y=False, color="blue")
            plot.update_yaxes(title_text=y2_title, secondary_y=True, color="green")
        else:
            plot.update_yaxes(title_text=y1_title)
            pass

        return plot


    #unit test
    #
    #import numpy as np
    #
    #time = range(50)
    #velocity = [20 * np.sin(x) for x in time]
    #voltage = [x * 60 for x in velocity]
    #pid = {”kP”:1.0,”kI”:.001,”kD”:.25,”kI_max”:5,”brakemode”:”Brake”,”gearset”:”18:1”,”slew”:400}
    #x = DebugGraph(time, velocity, voltage, 12, pid)
    #x.get_graph().show()
    #x.get_graph().savefig(”test.png”, bbox_inches=’tight’)

```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Jan  6 22:38:31 2020

@author: aiden
"""
class LCD:
    """
    class for communicating with the vex lcd via serial communication
    """
    def __init__(self, serial_write, serial_read):
        self.lcd_screen = serial_write
        self.lcd_buttons = serial_read

        self.__flags = 0x00
        self.__line_1 = " " * 16
        self.__line_2 = " " * 16

        self.__num_chars = 16


    def __write_line(self):
        """
        Writes to the lcd. Runs from thread, not called by user

        Returns
        -------
        int
            1 on success.

        """
        #line 1 bytearray
        send_array = bytearray()
        send_array.append(0xAA)
        send_array.append(0x55)
        send_array.append(0x1E)
        send_array.append(0x12)
        send_array.append(0x00)
        checksum = 0x00

        for char in self.__line_1:
            send_array.append(ord(char))
            checksum += (ord(char))

        self.lcd_screen.write(send_array)

        #line 2 bytearray
        send_array = bytearray()
        send_array.append(0xAA)
        send_array.append(0x55)
        send_array.append(0x1E)
        send_array.append(0x12)
        send_array.append(0x01)
        checksum = 0x01

        for char in self.__line_1:
            send_array.append(ord(char))
            checksum += (ord(char))

        self.lcd_screen.write(send_array)

        return 1




    def write_string(self, string, ln=0, align="left"):
        """
        writes a string to the lcd
        handles multiline writing by keeping track of line one and line two
        and adding a newline character between them

        returns 0 on unsucessful write and 1 if completed successfully
        """
        buffer = self.__num_chars - len(string)
        if align == "center":
            left_spaces = round(buffer/2, 0)
            right_spaces = self.__num_chars - len(string) - left_spaces
            if right_spaces < 0:
                right_spaces = 0
            string = (" " * left_spaces) + string + (" " * right_spaces)

        elif align == "right":
            left_spaces = self.__num_chars - len(string)
            if left_spaces < 0:
                left_spaces = 0
            string = (" " * left_spaces) + string

        elif align == "left":
            string = string

        else:
            return 0

        if len(string) > self.__num_chars:  #cap string length to the number of characters on the lcd
            string = string[:self.__num_chars]


        if ln == 0:
            self.__line_1 = string
        elif ln == 1:
            self.__line_2 = string

        return 1




    def clear(self, line):
        """
        clears a line on the lcd by sending spaces

```

```
Parameters
----------
line : int
    the line to clear.

Returns
-------
int
    1 on success, 0 on failure.

"""
string = " " * 16
ret = self.write_string(string, ln=line)

if not ret:
    return 0

return 1


```

../DocumentationMacros/function_header.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:14:17 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @param:")
pyautogui.press("enter")

pyautogui.typewrite("* @param:")
pyautogui.press("enter")

pyautogui.typewrite("* @return:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_function_line_3")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:20:16 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * how_function_works_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* how_function_works_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* how_function_works_line_3")
pyautogui.press("enter")


pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 21:23:07 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @see:")
pyautogui.press("enter")

pyautogui.typewrite("* @see:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* purpose_of_class_line_3")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 13 20:50:03 2019

@author: aiden
"""

import time
import pyautogui

time.sleep(.5)

pyautogui.typewrite("/**")
pyautogui.press("enter")

pyautogui.typewrite(" * @file:")
pyautogui.press("enter")

pyautogui.typewrite("* @author:")
pyautogui.press("enter")

pyautogui.typewrite("* @reviewed_on:")
pyautogui.press("enter")

pyautogui.typewrite("* @reviewed_by:")
pyautogui.press("enter")

pyautogui.typewrite("* TODO:")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_1")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_2")
pyautogui.press("enter")

pyautogui.typewrite("* description_of_contents_line_3")
pyautogui.press("enter")

pyautogui.typewrite("*")
pyautogui.press("enter")

pyautogui.typewrite("*/")
pyautogui.press("enter")

pyautogui.press("backspace")
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 15 10:16:03 2019

@author: aiden
"""
import inspect
import colorama
import fcntl
import os
import readline
import struct
import termios

import cpp_types
import config
import exceptions


class HeaderGen:
    """
    Wrapper class for working with cpp types and header files
    used to easily generate code based on user input
    """
    def __init__(self, header_obj):
        self.header = header_obj
        self.children = header_obj.get_children()
        self.focus = header_obj
        self.current_type = "header"
        self.loc = "./" + self.header.file_name

        self.commands = {
            "ls":self.__ls,
            "view":self.__view,
            "cd":self.__change_focus,
            "exit":self.__exit,
            "new":self.__new,
            "write":self.__write,
            "add":self.__add,
            "help":self.__help
            }


    @classmethod
    def __exit(cls, *_):
        """
        exit function for shell
        """
        raise exceptions.Exit

    def __ls(self, *_):
        """
        lists data on focused object
        """
        print(self.focus.list_data() + "\n")

    def __view(self, *args):
        """
        shows the generated text of the focused object
        param bool header_text sets the view to either
        the generated header text or the generated implementation
        file text
        """
        if args[0] == []:
            header_text = True
        elif str(args[0][0]).upper() == "HEADER":
            header_text = True
        elif str(args[0][0]).upper() == "IMPLEMENTATION":
            header_text = False
        else:
            raise exceptions.UnknownOption

        text = self.focus.gen_header_text() if header_text else self.focus.gen_impl_text()
        print(text + "\n")

    def __change_focus(self, *args):
        """
        changes focus to user specified input
        no return
        """
        name = args[0][0]
        if name == "..":
            self.focus = self.focus.parent
            self.update_type(self.focus)
            self.children = self.focus.get_children()
        elif self.focus.has_children:
            if name in self.children.keys():
                self.focus = self.children.get(name)
                self.update_type(self.focus)
                self.children = self.focus.get_children()
            else:
                print("invalid selection")
        else:
            print("focused object has no children")


        path = [self.focus]
        while len(path) == len(set(path)):
            path.insert(0, path[0].parent)
        path.pop(0)
        path.pop(0)
        self.loc = "./" + self.header.file_name + "/" + "/".join(x.name for x in path)



    def __new_class(self, name):
        """
        adds a class object to a header file object
        param name = type str of name of the class

        throws invalid addition if not currently focused on header object
        """
```

```python
        if self.current_type == "header":
            obj = cpp_types.cpp_class(name, self.focus)
            self.header.classes.append(obj)
        else:
            raise exceptions.InvalidAddition


    def __new_func(self, loc, return_type, name, static=False):
        """
        adds a function object to a header file or class object
        param name = type str of name of the function
        param return_type = type of function
        param static = is the function static or not

        throws invalid addition if incorrect params are passed
        """
        obj = cpp_types.cpp_func(name, return_type, static, self.focus)

        if self.current_type == "header":
            self.focus.funcs.append(obj)
        elif self.current_type == "class":
            if static:
                if loc.upper() == "PUBLIC":
                    self.focus.public["static_func"].append(obj)
                elif loc.upper() in ["PROT", "PROTECTED"]:
                    self.focus.protected["static_func"].append(obj)
                else:
                    self.focus.private["static_func"].append(obj)
            else:
                if loc.upper() == "PUBLIC":
                    self.focus.public["func"].append(obj)
                elif loc.upper() in ["PROT", "PROTECTED"]:
                    self.focus.protected["func"].append(obj)
                else:
                    self.focus.private["func"].append(obj)
        else:
            raise exceptions.InvalidAddition

    def __new_var(self, loc, var_type, name, static=False):
        """
        adds a function object to a header file or class object
        param name = type str of name of the function
        param var_type = type of variable
        param static = is the variable static or not

        throws invalid addition if incorrect params are passed
        """
        obj = cpp_types.cpp_variable(name, var_type, static, self.focus)

        if self.current_type == "header":
            self.focus.static_vars.append(obj)
        elif self.current_type == "class":
            if static == "static":
                if loc.upper() == "PUBLIC":
                    self.focus.public["static_var"].append(obj)
                elif loc.upper() in ["PROT", "PROTECTED"]:
                    self.focus.protected["static_var"].append(obj)
                else:
                    self.focus.private["static_var"].append(obj)
            else:
                if loc.upper() == "PUBLIC":
                    self.focus.public["var"].append(obj)
                elif loc.upper() in ["PROT", "PROTECTED"]:
                    self.focus.protected["var"].append(obj)
                else:
                    self.focus.private["var"].append(obj)
        else:
            raise exceptions.InvalidAddition



    def __new_include(self, include_type, name):
        """
        adds an include to a header file
        param type (lib, user) - type of include
        param name - name of include

        throws invalid addition if incorrect params are passed
        """
        if include_type == "user":
            self.header.user_includes.append(name.strip())
        elif include_type == "lib":
            self.header.lib_includes.append(name.strip())
        else:
            raise exceptions.InvalidAddition



    def __new(self, *args):
        """
        adds a type of object to another type of object
        checks to see if addition is valid
        ex. if creating a class the parent must be of type header

        throws InvalidAddition if the addition failed
        """
        func_dict = {
            "class":self.__new_class,
            "var":self.__new_var,
            "func":self.__new_func,
            "function":self.__new_func,
            "include":self.__new_include
            }

        if not args[0]:
            raise exceptions.InvalidAddition("invalid parameters were passed")

        obj_type = args[0][0].strip()
        params = list(map(lambda x: x.strip(), args[0][1:]))

        func = func_dict.get(obj_type)
        if func:
            num_args = len(inspect.signature(func).parameters)
            if num_args > len(params):
                params.insert(0, "")
            while num_args > len(params):
                params.append("")

        elif func_dict.get(args[0][1].strip()): #if argument in second position
                                                #is a valid command then switch
                                                #param in first position to front
                                                #of params list
            func = func_dict.get(args[0][1].strip())
            params = list(map(lambda x: x.strip(), args[0][2:]))

            num_args = len(inspect.signature(func).parameters)
            if num_args > len(params):
                params.insert(0, args[0][0].strip())
            while num_args > len(params):
                params.append("")

        else:
            raise exceptions.InvalidAddition("invalid function call")

        func(*params[:num_args])
        self.children = self.focus.get_children()



    def __add_function_param(self, param_type, param_name):
        """
        adds parameters to a function
        @param param_type - type str of the cpp type
        @param param_name - name of the parameter
        throws Invalid Addition on failure or invalid params
        """
        if self.current_type != "function":
            raise exceptions.InvalidAddition

        param = param_type.strip() + " " + param_name.strip()
        self.focus.params.append(param)


    def __add(self, *args):
        """
        used to add attributes such as parameters to a function

        throws invalid addition if the addition failed
        """
        func_dict = {
            "param":self.__add_function_param
            }

        if not args[0]:
            raise exceptions.InvalidAddition("invalid parameters were passed")

        obj_type = args[0][0].strip()
        params = list(map(lambda x: x.strip(), args[0][1:]))

        func = func_dict.get(obj_type)

        if func:
            num_args = len(inspect.signature(func).parameters)
            while num_args > len(params):
                params.append("")

            func(*params[:num_args])
            self.children = self.focus.get_children()

        else:
            raise exceptions.InvalidAddition("invalid function call")




    def __write(self, *_):
        """
        writes the text generated from the header file into an actual file
        as well as generating and writing the text for an implementation
        file
        """
        header_file_name = self.header.file_name
        impl_file_name, _ = os.path.splitext(self.header.file_name)
        impl_file_name += ".cpp"

        with open(header_file_name, "a") as file:
            file.write(self.header.gen_header_text())

        with open(impl_file_name, "a") as file:
            file.write(self.header.gen_impl_text())




    def __help(self, *_):
        """
        prints docstrings for each function
        """
        #get terminal size to set max chars per line
        _, columns, _, _ = struct.unpack('HHHH',
                                        fcntl.ioctl(0, termios.TIOCGWINSZ,
                                        struct.pack('HHHH', 0, 0, 0, 0)))

        max_chars = columns - 5
        help_msg = ""
        spaces = len(max(list(self.commands.keys()), key=len))

        for key in self.commands:
            doc_str = str(self.commands.get(key).__doc__).strip().replace("\n", " ")
            words = doc_str.split(" ")
            words = list(filter(lambda a: a != "", words))

            line = key + (" " * (spaces - len(key))) + "  -  "
            indentation = len(line) * " "
            for word in words:
                if (len(word) + len(line)) < max_chars:
                    line += word + " "
                else:
                    help_msg += line + "\n"
                    line = indentation + word + " "
            help_msg += line + "\n\n"

        print(help_msg)




    def update_type(self, obj):
        """
        updates self.current_type to the type of obj
        """
        types = {cpp_types.cpp_class:"class",
                 cpp_types.cpp_func:"function",
                 cpp_types.HeaderFile:"header",
                 cpp_types.cpp_variable:"variable"
                }
        self.current_type = types.get(type(obj))




    def execute_command(self, command):
        """
        executes a command from the given api
        api commands are stored in self.commands
        """
        cmd = self.commands.get(command.split(" ")[0].strip())
        args = command.split(" ")[1:]
        for arg in args:
            arg.strip()

        try:
            if not cmd:
                raise exceptions.UnknownOption
            cmd(args)
        except exceptions.UnknownOption:
            pass
        except exceptions.InvalidAddition:
            pass



#TODO: add dynamically changing autocomplete
class Shell:
    """
    contains shell like interface for generating code
    """
    def __init__(self):
        self.loc = "./"
        self.functions = sorted(["new",
                                 "class",
                                 "edit",
                                 "ls",
                                 "include",
                                 "user",
                                 "lib",
                                 "func",
                                 "view",
                                 "exit",
                                 "add"])


    def auto_complete(self, text, state):
        """
        function that will attempt to autocomplete user input
        on <tab> to s member in self.functions
        returns type str of first matched string
        """
        if state == 0:  # on first trigger, build possible matches
            if text:  # cache matches (entries that start with entered text)
                matches = [s for s in self.functions if s and s.startswith(text)]
            else:  # no text entered, all matches possible
                matches = self.functions[:]

        # return match indexed by state
        try:
            return matches[state]
        except IndexError:
            return None


    def get_command(self):
        """
        gets command from user
        returns type str of command
        """
        print("")

        colorama.init()
        readline.set_completer(self.auto_complete)
        readline.parse_and_bind('tab: complete')

        command = input((config.SHELL_COLOR
                         + "\n"
                         + config.CURSOR_UP_ONE
                         + config.ERASE_LINE
                         + self.loc + " <command> "
                         + colorama.Fore.RESET))
        return command



#def unit_test():
#    h = cpp_types.HeaderFile("MyHeader.hpp")
#
#    c = cpp_types.cpp_class("MyClass")
#    c.protected["func"].append(cpp_types.cpp_func("my_func", "int", 1))
#    c.public["var"].append("int x")
#    h.static_vars.append("int y")
```

```python
#    h.classes.append(c)
#
#    t = h.gen_header_text()
#    print(t)




file_name = input(config.SHELL_COLOR
                  + "\n"
                  + config.CURSOR_UP_ONE
                  + config.ERASE_LINE
                  + "Enter name of Header File to Create "
                  + colorama.Fore.RESET)

if not any(s in file_name and s[-len(s):] == file_name[-len(s):] for s in ['.hpp', '.h']):
    raise exceptions.InvalidFileName



header = cpp_types.HeaderFile(file_name)
s = Shell()
s.loc += file_name + "/"
header_gen = HeaderGen(header)


while 1:
    try:
        try:
            usr_command = s.get_command()
        except KeyboardInterrupt:
            print()
            raise exceptions.Exit

        header_gen.execute_command(usr_command)
        s.loc = header_gen.loc

    except exceptions.Exit:
        break



#unit_test()

```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 17 19:52:41 2019

@author: aiden
"""
import os
import config

class cpp_class:
    """
    contains methods and data for a cpp class
    generates text for header and implementation files
    """
    def __init__(self, name, parent):
        self.name = name

        self.has_children = True
        self.parent = parent

        self.private = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }
        self.protected = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }
        self.public = {
            "func":[],
            "static_func":[],
            "var":[],
            "static_var":[]
            }



    @classmethod
    def __keys_empty(cls, section):
        """
        checks if all keys in a dictionary have no value
        returns True if empty, False otherwise
        """
        for key in section:
            if section.get(key):
                return False

        return True




    def list_data(self):
        """
        lists the data in the class by category
        returns type str of data
        """
        text = ""
        sections = [self.private,
                    self.protected,
                    self.public]
        section_names = ["private:\n",
                         "protected:\n",
                         "public:\n"]

        for i, section in enumerate(sections):
            text += section_names[i] + "\tstatic variables\n"
            for static_var in section.get("static_var"):
                text += "\t\t" + static_var.var_type + " " + static_var.name + "\n"

            text += "\tvariables\n"
            for var in section.get("var"):
                text += "\t\t" + var.var_type + " " + var.name + "\n"

            text += "\tstatic functions\n"
            for static_func in section.get("static_func"):
                text += "\t\t" + static_func.type + " " + static_func.name + "\n"

            text += "\tfunctions\n"
            for func in section.get("func"):
                text += "\t\t" + func.type + " " + func.name + "\n"

        return text




    def get_children(self):
        """
        returns a dict of children names and their object
        ex. {name1:obj1,
             name2:obj2,
             name3:obj3,
             ...}
        """
        children = {}
        sections = [self.private, self.protected, self.public]

        for section in sections:
            for static_var in section.get("static_var"):
                children.update({static_var.name:static_var})

            for var in section.get("var"):
                children.update({var.name:var})

            for static_func in section.get("static_func"):
                children.update({static_func.name:static_func})

            for func in section.get("func"):
                children.update({func.name:func})

        return children


    def gen_header_text(self):
        """
        takes data in the class and generates text for a class in
        a header file
        returns type str of text
        """
        text = "class " + self.name + "\n{\n\tprivate:\n"

        #TODO: condense, function too long, too many branches
        for member in self.private.get("static_func"):
            text += "\t\t" + member.gen_header_text()
        if self.private.get("static_func"):
            text += "\n\n"

        for member in self.private.get("func"):
            text += "\t\t" + member.gen_header_text()
        if self.private.get("func"):
            text += "\n\n"

        for member in self.private.get("static_var"):
            text += "\t\t" + member.gen_header_text()
        if self.private.get("static_var"):
            text += "\n\n"

        for member in self.private.get("var"):
            text += "\t\t" + member.gen_header_text()
        if self.private.get("var"):
            text += "\n\n"


        prot_txt = ""
        if not self.__keys_empty(self.protected):
            #if not empty
            prot_txt = "\tprotected:\n"
            for member in self.protected.get("static_func"):
                prot_txt += "\t\t" + member.gen_header_text()
            if self.protected.get("static_func"):
                prot_txt += "\n\n"

            for member in self.protected.get("func"):
                prot_txt += "\t\t" + member.gen_header_text()
            if self.protected.get("func"):
                prot_txt += "\n\n"

            for member in self.protected.get("static_var"):
                prot_txt += "\t\t" + member.gen_header_text()
            if self.protected.get("static_var"):
                prot_txt += "\n\n"

            for member in self.protected.get("var"):
                prot_txt += "\t\t" + member.gen_header_text()
            if self.protected.get("var"):
                prot_txt += "\n\n"
        elif self.__keys_empty(self.protected) and not config.REMOVE_PROTECTED_IF_EMPTY:
            #if empty and dont remove protected
            prot_txt = "\tprotected:\n"

        text += prot_txt + "\tpublic:\n\t\t" + self.name + "();\n\t\t~" + self.name + "();\n\n"


        for member in self.public.get("static_func"):
            text += "\t\t" + member.gen_header_text()
        if self.public.get("static_func"):
            text += "\n\n"

        for member in self.public.get("func"):
            text += "\t\t" + member.gen_header_text()
        if self.public.get("func"):
            text += "\n\n"

        for member in self.public.get("static_var"):
            text += "\t\t" + member.gen_header_text()
        if self.public.get("static_var"):
            text += "\n\n"

        for member in self.public.get("var"):
            text += "\t\t" + member.gen_header_text()
        if self.public.get("var"):
            text += "\n\n"

        text += "};\n"

        tab = ""
        for _ in range(config.TAB_SIZE):
            tab += " "
        text = text.replace("\t", tab)

        return text




    def gen_impl_text(self):
        """
        generates text for a class in an implementation file
        returns type str of text
        """
        text = ""

        sections = [self.private, self.protected, self.public]
        for section in sections:
            for static_var in section.get("static_var"):
                text += static_var.gen_impl_text()

        text += "\n\n\n" + self.name + "::" + self.name + "( )\n{\n\n}\n"
        text += "\n\n\n" + self.name + "::~" + self.name + "( )\n{\n\n}\n\n\n\n"

        for section in sections:
            for static_func in section.get("static_func"):
                text += static_func.gen_impl_text() + "\n\n\n"

            for func in section.get("func"):
                text += func.gen_impl_text() + "\n\n\n"

        return text




class cpp_func:
    """
    contains data about a cpp function type
    can be either static or not and has methods to
    generate text in both header and implementation file
    """
    def __init__(self, name, return_type, static, parent):
        self.type = return_type
        self.name = name
        self.static = bool(static)

        self.has_children = False
        self.parent = parent

        self.params = []




    def get_children(self):
        """
        returns a dict of children names and their object
        ex. {name1:obj1,
             name2:obj2,
             name3:obj3,
             ...}
        since there are no accesible children, an empty
        list is returned
        """
        return {}



    def list_data(self):
        """
        lists the data in the function by category
        returns type str of data
        """
        text = "type:\n\t" + self.type + "\nname:\n\t" + self.name + "\nstatic:\n\t"
        text += "yes\n" if self.static else "no\n"
        text += "parameters:\n"
        for param in self.params:
            text += "\t" + param + "\n"

        return text




    def gen_header_text(self):
        """
        generates text for a function with given dataset to be placed
        in a header file
        returns type str of text for header file
        """
        text = ""
        if self.static:
            text += "static "

        text += self.type + " "
        text += self.name + "( "
        for param in self.params:
            text += param + ", "
        k = text.rfind(",")
        if k > 0:
            text = text[:k] + text[k+1:]
        text += ");\n"

        return text




    def gen_impl_text(self, class_name=""):
        """
        generates text for a function with a given dataset to be placed
        in an implementation file
        returns type str of text for implementation file
        """
        text = ""

        text += self.type + " "
        if class_name:
            text += class_name + "::" + self.name + "( "
        else:
            text += self.name + "( "

        for param in self.params:
            text += param + ", "
        k = text.rfind(",")
        if k > 0:
            text = text[:k] + text[k+1:]

        text += ")\n{\n\n}"

        return text




class cpp_variable:
```

```python
    """
    contains data about a cpp variable type
    can be either static or not and has methods to
    generate text in both header and implementation file
    """
    def __init__(self, name, var_type, static, parent):
        self.var_type = var_type
        self.name = name
        self.static = bool(static)

        self.has_children = False
        self.parent = parent



    def get_children(self):
        """
        returns a dict of children names and their object
        ex. {name1:obj1,
            name2:obj2,
            name3:obj3,
            ...}
        since there are no accesible children, an empty
        list is returned
        """
        return {}




    def list_data(self):
        """
        lists the data in the variable by category
        returns type str of data
        """
        text = "type:\n\t" + self.var_type + "\nname:\n\t" + self.name + "\nstatic:\n\t"
        text += "yes\n" if self.static else "no\n"

        return text




    def gen_header_text(self):
        """
        generates text for a variable with given dataset to be placed
        in a header file
        returns type str of text for header file
        """
        text = "static " if self.static else ""

        text += self.var_type + " " + self.name + ";"

        return text




    def gen_impl_text(self):
        """
        generates text for a variable with a given dataset to be placed
        in an implementation file
        returns type str of text for implementation file
        """
        return self.var_type + " " + self.name + " = ;\n"





class HeaderFile:
    """
    contains data for a cpp header file
    data can also be used to generate an implementation file
    """
    def __init__(self, file_name):
        self.file_name = file_name

        self.parent = self
        self.has_children = True

        self.user_includes = []
        self.lib_includes = []
        self.classes = []
        self.static_vars = []
        self.funcs = []




    def list_data(self):
        """
        lists the data in the header file by category
        returns type str of data
        """
        text = "file name:\n\t" + self.file_name + "\nlibrary includes:\n"
        for include in self.lib_includes:
            text += "\t" + include + "\n"

        text += "user includes:\n"
        for include in self.user_includes:
            text += "\t" + include + "\n"

        text += "classes:\n"
        for cpp_cls in self.classes:
            text += "\t" + cpp_cls.name + "\n"

        text += "variables:\n"
        for static_var in self.static_vars:
            text += "\t" + static_var.gen_header_text() + "\n"

        text += "functions:\n"
        for func in self.funcs:
            text += "\t" + func.type + " " + func.name + "\n"

        return text




    def get_children(self):
        """
        returns a dict of children names and their object
        ex. {name1:obj1,
             name2:obj2,
             name3:obj3,
             ...}
        """
        children = {}

        for cpp_cls in self.classes:
            children.update({cpp_cls.name:cpp_cls})

        for static_var in self.static_vars:
            children.update({static_var.name:static_var})

        for func in self.funcs:
            children.update({func.name:func})

        return children




    def gen_header_text(self):
        """
        generates text for a header file
        returns type str of text
        """
        #add header guards
        guard, _ = os.path.splitext(self.file_name)
        guard = guard.split("/")[-1]

        text = "#ifndef __" + guard.upper() + "_HPP__\n"
        text += "#define __" + guard.upper() + "_HPP__\n\n"

        for include in sorted(self.lib_includes):
            text += "#include <" + include + ">\n"

        text += '\n#include "main.h"\n\n'

        for include in sorted(self.user_includes):
            text += '#include "' + include + '"\n'

        text += "\n\n"

        for c in self.classes:
            text += c.gen_header_text() + "\n\n\n\n"

        for func in self.funcs:
            text += func.gen_header_text() + "\n"
        if self.funcs:
            text += "\n\n\n\n"

        for var in self.static_vars:
            text += var.gen_header_text()
        if self.static_vars:
            text += "\n\n\n\n"

        text += "#endif\n"

        return text




    def gen_impl_text(self):
        """
        generates text for an implementation file
        returns type str of text
        """
        text = ""
        for include in sorted(self.lib_includes):
            text += "#include <" + include + ">\n"

        text += '\n#include "main.h"\n\n'

        text += '#include "' + self.file_name + '"\n'

        for include in sorted(self.user_includes):
            text += '#include "' + include + '"\n'

        text += "\n\n"

        for var in self.static_vars:
            text += var.gen_impl_text()
        if self.static_vars:
            text += "\n\n\n\n"

        for c in self.classes:
            text += c.gen_impl_text() + "\n\n\n\n"

        for func in self.funcs:
            text += func.gen_impl_text()
        if self.funcs:
            text += "\n\n\n\n"

        return text
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 20 16:48:22 2019

@author: aiden
"""

class Exit(Exception):
    """
    thrown for graceful program termination
    stack trace might not be shown if raised
    """
    pass

class UnknownOption(Exception):
    """
    thrown when an invalid parameter or and invalid command is given
    """
    def __init__(self):
        msg = "Unknown option given"
        super().__init__(msg)
        print(msg)

class InvalidFileName(Exception):
    """
    exception for and invalid file name given
    """
    def __init__(self):
        msg = ("Invalid file name given:\n"
                + "must end in valid header extension type ex.’.hpp’ or ’.h’")

        super().__init__(msg)
        print(msg)
        raise Exit

class InvalidAddition(Exception):
    """
    addition of object failed
    """
    def __init__(self, msg=None):
        if not msg:
            msg = ("adding of object failed: check to see that parent is of a valid type\n"
                    + "and that parameters were passed correctly")
        super().__init__(msg)
        print(msg)
```

../CodeGenerator/config.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 17 19:53:35 2019

@author: aiden
"""
import colorama

TAB_SIZE = 4
REMOVE_PROTECTED_IF_EMPTY = True
SHELL_COLOR = colorama.Fore.LIGHTGREEN_EX
CURSOR_UP_ONE = ’\x1b[1A’
ERASE_LINE = ’\x1b[2K’
```

../CodeGenerator/codegen.sh

```
#!/bin/bash
#move this to /bin and give executable permissions
#to be able to run python script from anywhere
python3 /home/aiden/Documents/VexCode-2019/CodeGen/code_gen.py
```

```python
import tkinter as tk

import robot
import field
import fieldObjects
import autonomous
import runningFrame
import robotInfoFrame
import controlPanelFrame


#user defines
X_RES = 1600
Y_RES = 900
ROBOT_START_TILE = [0, 1]
ROBOT_START = [0, 6]
ROBOT_START_ANGLE = 0



#sets up and draws field
f = field.Field(X_RES, Y_RES)
f.drawField()
f.drawGrid()
f.fieldInfo()
field = f.field
fieldSize = f.distance
robotCoordinates = field[ROBOT_START_TILE[0]][ROBOT_START_TILE[1]].placeRobot([ROBOT_START[0], ROBOT_START[1]])

fieldObjects.main(field)



#gets canvas and tkinter object
canvas = f.canvas
master = f.master



#sets up GUI further
master.title("Autonomous Simulator")
master.wm_title("Autonomous Simulator")
icon = tk.Image("photo", file="autonSimulator.png")
# master.tk.call('wm', 'iconphoto', master._w, icon)
master.resizable(0, 0)


#sets up buttons and labels
runningPane = runningFrame.runningFrame(master)
robotInfoPane = robotInfoFrame.robotInfoFrame(master)
controlPanelPane = controlPanelFrame.controlPanelFrame(master, runningPane)

runningPane.placeObjects()
robotInfoPane.placeObjects()
controlPanelPane.placeObjects()



#moves robot to start location
bot = robot.robot(fieldSize=fieldSize, tkobj=master, canvas=canvas, controlPanelFrame=controlPanelPane, robotInfoFrame=robotInfoPane)
bot.show(angle=ROBOT_START_ANGLE, position=robotCoordinates)


def motion(event):
    x, y = event.x, event.y
    screen_offset_x = (X_RES - (Y_RES - 100)) / 2
    screen_offset_y = 50
    offset_x = bot.x_offset_in
    offset_y = bot.y_offset_in
    # print(bot.inches(x), bot.x_offset_in)
    corrected_coords = [bot.inches(x) - bot.y_offset_in, bot.inches(y) - bot.x_offset_in]
    print('{}, {}'.format(corrected_coords[0], corrected_coords[1]))

def forward(event):
    bot.forward(25)

def backward(event):
    bot.backward(25)

def turnRight(event):
    bot.turnRight(2)

def turnLeft(event):
    bot.turnLeft(2)


#runs autonomous
try:
    master.bind('<Motion>', motion)
    master.bind("w", forward)
    master.bind("s", backward)
    master.bind("a", turnLeft)
    master.bind("d", turnRight)

    auton = autonomous.auton(bot, master, controlPanelPane, robotInfoPane)
    auton.commands()

except tk.TclError:
    pass

finally:
    try: #finished all the way through
        controlPanelPane.disable()

        controlPanelPane.idle()

    except tk.TclError: #window closed
        print("finished")
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import time

class auton:
    """
    contains autonomous commands
    """
    def __init__(self, robotObj, master, controlPanelFrame, robotInfoFrame):
        self.controlPanelFrame = controlPanelFrame
        self.robotInfoFrame = robotInfoFrame
        self.robot = robotObj
        self.master = master

        self.orientationLabelText = self.robotInfoFrame.orientationLabelText
        self.orientationLabelText.set("orientation: " + str(self.robot.orientationDegrees))

        self.controlPanelFrame.runningLabelText.set(self.controlPanelFrame.options.get(str(self.controlPanelFrame.keepRunning)))



    def nextFrame(self, waitTime=0):
        """
        updates tkinter canvas and allows for a wait
        """

        self.master.update()

        timeSlept = 0
        while timeSlept <= waitTime:
            if self.controlPanelFrame.keepRunning:
                time.sleep(0.1)
                timeSlept += 0.1
                self.master.update()
            else: #sets robot to idle
                time.sleep(0.1)
                self.master.update()



    def intakeStart(self, rotations):
        """
        for translator
        """
        pass

    def outakeStart(self, rotations):
        """
        for translator
        """
        pass

    def intakeEnd(self, Time):
        """
        for translator
        """
        pass

    def catapult(self, rotations):
        """
        for translator
        """
        pass

    def capFlipper(self, rotations):
        """
        for translator
        """
        pass


    def commands(self):
        """
        autonomous commands
        """
        self.nextFrame(1)

        self.robot.drive_to_point(0, 27)
        self.nextFrame(.25)

        self.robot.drive_to_point(27.2, 5.5, 1)
        self.nextFrame(.25)

        self.robot.drive_to_point(0, 27)
        self.nextFrame(.25)

        self.robot.drive_to_point(-22.5, 4.8, 1)
        self.nextFrame(.25)

        # self.robot.forward(350)
        # self.nextFrame(.25)

        #self.robot.turnLeft(10)
        #self.nextFrame(.25)


        # self.robot.drive_to_point(0, 30)
        # self.nextFrame(.25)


        # self.robot.drive_to_point(0, 0)
        # self.nextFrame(1)






######################## unit test ########################
#        import random                                    #
#        for x in range(0,4):                             #
#            self.robot.forward(500)                      #
#            self.nextFrame(1)                            #
#                                                 #
#           self.robot.turnLeft(90)               #
#           self.nextFrame(1)                     #
#                                                 #
#       self.robot.turnLeft(45)                   #
#       self.nextFrame(1)                         #
#                                                 #
#       for x in range(0,4):                      #
#           self.robot.forward(400)               #
#           self.nextFrame(1)                     #
#                                                 #
#           self.robot.turnRight(90)              #
#           self.nextFrame(1)                     #
#                                                 #
#       self.robot.turnRight(45)                  #
#       self.nextFrame(1)                         #
#                                                 #
#                                                 #
#       for x in range(0, 6):                     #
#           self.robot.forward(500)               #
#           self.nextFrame(1)                     #
#                                                 #
#           self.robot.turnRight(random.randint(10,90))    #
#           self.nextFrame(1)                     #
#                                                 #
###################################################################
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import smtplib
import getpass


SEND_EMAIL = 0
FILE = 'autonomous.py'


class email():
    def __init__(self):
        self.passwd = None
        self.user_email = 'jg570144@gmail.com'
        self.recipients = []

        self.message = None

        self.smtpObj = None
        self.check = None
        self.subject = None
        self.body = None
        self.sent = 0


    def getCredentials(self):
        #self.user_email = input("email address:\n")
        self.passwd = getpass.getpass()


    def getMessage(self, oldCode, newCode):
        r = int(input("\nEnter number of recipients\n"))

        for x in range(r):
            remail = input("enter recipient email\n")
            self.recipients.append(remail)

        self.subject = "translated autonomous code"

        self.body = "old auton code:\n\n"

        for i in oldCode:
            self.body = self.body + i + '\n'
            print("")

        self.body = self.body + '\n\n\n\nnew auton code:\n\n'

        for i in finalCode:
            self.body = self.body + i + '\n'

        self.body = self.body + '\n\n\nthis is an automated message\ncourtesy of Aiden'
        print("")



    def login(self):
        self.smtpObj = smtplib.SMTP("smtp.gmail.com:587")
        self.smtpObj.starttls()
        self.smtpObj.ehlo()
        self.smtpObj.login(self.user_email, self.passwd)


    def send(self):
        print("sent: ")

        message = "Subject: " + self.subject + "\n" + self.body

        self.check = self.smtpObj.sendmail(self.user_email, self.recipients, message)

    def close(self):
        self.smtpObj.quit()












class Translator:

    def __init__(self, file):
        self.file = open(file)
        self.code = self.file.readlines()

        self.oldCode = []
        self.newCode = []

        self.conversion = {
            "turnRight": "turnRightV",
            "turnLeft": "turnLeftV",
            "rightSide": "rs",
            "leftSide": "ls",
            "forward": "driveForward",
            "backward": "driveForward",
            "reverse": "changeDirection",
            "intakeStart": "intakeStart",
            "intakeEnd": "intakeEnd",
            "outake": "intakeStart",
            "catapult": "shootBall",
            "capFlipper": "flip"
            }


        self.functionMap = {
            "value": self.value,
            "sleep100RPM": self.sleep100RPM,
                "sleep200RPM": self.sleep200RPM,
                "sleep600RPM": self.sleep600RPM,
            "negativeValue": self.negativeValue,
            "noParam": self.noParam
             }


        self.specialInstructions = {
            "forward": ["value", "sleep200RPM"],
            "backward": ["negativeValue", "sleep200RPM"],
            "turnLeft": ["noParam"],
            "turnRight": ["noParam"],
            "intakeStart": ["negativeValue"],
            "outakeStart": ["value"],
            "intakeEnd": ["sleep600"],
            "catapult": ["value", "sleep100"],
            "capFlipper": ["value"],
            }



        self.parameterValue = ""
        self.sleepTime = ""
        self.separator = ""
        self.instructions = []


    def value(self):
        self.parameterValue = str(self.parameterValue)
        self.sleepTime = ""
        self.separator = ""

    def sleep100RPM(self):
        RPM = 100
        RPS = RPM / 60
        revolutions = (abs(float(self.parameterValue)) / 360)
        self.sleepTime = (1000 * (revolutions / RPS)) + 50
        #self.sleepTime = str(int((1000 * abs((float(self.parameterValue))/360))) + 200)
        if len(self.instructions) == 1:
            self.parameterValue = ""
            self.separator = ""
        else:
            self.separator = ", "

    def sleep200RPM(self):
        RPM = 200
        RPS = RPM / 60
        revolutions = (abs(float(self.parameterValue)) / 360)
        self.sleepTime = (1000 * (revolutions / RPS)) + 50
        if len(self.instructions) == 1:
            self.parameterValue = ""
            self.separator = ""
        else:
            self.separator = ", "

    def sleep600RPM(self):
        RPM = 600
        RPS = RPM / 60
        revolutions = (abs(float(self.parameterValue)) / 360)
        self.sleepTime = (1000 * (revolutions / RPS)) + 50
        if len(self.instructions) == 1:
            self.parameterValue = ""
            self.separator = ""
        else:
            self.separator = ", "


    def negativeValue(self):
        self.parameterValue = str(0 - float(self.parameterValue))
        self.sleepTime = ""
        self.separator = ""

    def noParam(self):
        self.parameterValue = ""
        self.sleepTime = ""
        self.separator = ""





    def translate(self):

        for i in self.code:

            try:
                oldCommand = i.split(".")[1]
                try:
                    rawCommand = i.split(".")[2]
                    oldCommand = oldCommand + "." + rawCommand
                except:
                    rawCommand = i.split(".")[1]

                oldCommand = oldCommand.split(")")[0]
                oldCommand = 'self.' + oldCommand + ")"

                command = rawCommand.split("(")[0]
                value = rawCommand.split("(")[1]
                self.parameterValue = value.split(")")[0]

                self.instructions = list(self.specialInstructions.get(command))
                for j in self.instructions:
                    self.functionMap[str(j)]()

                newCommand = self.conversion.get(command)
                newCommand = (newCommand
                            + "(" + str(self.parameterValue)
                            + str(self.separator)
                            + str(self.sleepTime)
                            + ");"
                            )

                self.oldCode.append(oldCommand)
```

```python
                self.newCode.append(newCommand)
                self.parameterValue = ""
                self.sleep = ""
                self.separator = ""

            except:
                self.parameterValue = ""
                self.sleep = ""
                self.separator = ""

        return self.oldCode, self.newCode



t = Translator(FILE)
oldCode, finalCode = t.translate()


if SEND_EMAIL:
    s = email()
    s.getCredentials()
    s.getMessage(oldCode, finalCode)
    s.login()
    s.send()
    s.close()

    print("")
    print("")
    print("done")

else:
    for i in oldCode:
        print(i)
    print("")
    for i in finalCode:
        print(i)
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import tkinter as tk
import time

class controlPanelFrame:
    """
    makes frame object for all labels and buttons
    """
    def __init__(self, master, runningFrame):
        self.runningLabelText = runningFrame.runningLabelText
        self.master = master

        self.guiFrame = tk.Frame(self.master)
        self.guiFrame.grid(row=1, column=0, sticky='news')

        self.speed = 7
        self.keepRunning = True

        self.options = {
            "True": "running",
            "False":"paused "
            }

        self.pauseButtonTextOptions = {
            "True": "pause ",
            "False":"resume"
             }


        ##### labels #####

        #speed label
        self.speedLabelText = tk.StringVar()
        self.speedLabel = tk.Label(self.guiFrame, textvariable=self.speedLabelText)


        ##### buttons #####

        self.fasterButton = tk.Button(self.guiFrame, text=' + ', command=self.__faster)
        self.slowerButton = tk.Button(self.guiFrame, text=' - ', command=self.__slower)

        self.pauseButtonText = tk.StringVar()
        self.pauseButton = tk.Button(self.guiFrame, textvariable=self.pauseButtonText, command=self.__pause)
        self.master.bind("<space>", self.__pause)




    def __changeSpeedLabel(self):
        self.speedLabelText.set("speed: " + str(self.speed))



    def __faster(self):
        if self.speed < 10:
            self.speed = self.speed + 1
            self.__changeSpeedLabel()

    def __slower(self):
        if self.speed > 1:
            self.speed = self.speed - 1
            self.__changeSpeedLabel()

    def __pause(self, event=None):
        """
        pauses robot actions
        """
        self.keepRunning = not self.keepRunning

        self.runningLabelText.set(self.options.get(str(self.keepRunning)))
        self.pauseButtonText.set(self.pauseButtonTextOptions.get(str(self.keepRunning)))

        self.master.update()


    def idle(self):
        """
        sends canvas into idle state
        """
        while 1:
            time.sleep(0.1)
            self.master.update()

    def disable(self):
        self.pauseButton.config(state="disabled")
        self.fasterButton.config(state="disabled")
        self.slowerButton.config(state="disabled")

        self.master.unbind("<space>")

    def placeObjects(self):
        """
        places all buttons and labels
        """

        self.pauseButton.grid(row=0, column=2, columnspan=4, rowspan=2, sticky='news', ipadx=30)

        self.speedLabel.grid(row=2, column=2, columnspan=4, rowspan=1, sticky='news')

        self.slowerButton.grid(row=0, column=0, rowspan=3, columnspan=2, sticky='news')
        self.fasterButton.grid(row=0, column=6, rowspan=3, columnspan=2, sticky='news')


        #sets default value of labels
        self.pauseButtonText.set(self.pauseButtonTextOptions.get(str(self.keepRunning)))
        self.speedLabelText.set("speed: " + str(self.speed))
        self.runningLabelText.set(self.options.get(str(self.keepRunning)))


```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


def main(field):
    """
    creates field elements
    """
    for num in range(0,6): #draw white line auton line
        field[2][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=10, outline="#ffffff")

    for num in range(0,6): #draw red alliance starting line
        field[4][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=7, outline="#ffffff")

    for num in range(0,6): #draw blue alliance starting line
        field[0][num].drawRectangle(vertexes=[[47,0],[47,47],[47,47],[47,0]], color="#ffffff", width=7, outline="#ffffff")

    field[0][3].drawRectangle(vertexes=[[0,0],[47,0],[47,0],[0,0]], color="#ffffff", width=10, outline="#ffffff")
    field[5][3].drawRectangle(vertexes=[[0,0],[47,0],[47,0],[0,0]], color="#ffffff", width=10, outline="#ffffff")

    #corner goal
    field[0][0].drawObjectFieldElementCircle(position=[11, 11], color='#919191', size=10)
    field[5][0].drawObjectFieldElementCircle(position=[36, 10], color='#919191', size=10)
    field[0][5].drawObjectFieldElementCircle(position=[11, 38], color='#919191', size=10)
    field[5][5].drawObjectFieldElementCircle(position=[36, 38], color='#919191', size=10)

    #cross goals
    field[3][0].drawObjectFieldElementCircle(position=[0, 10], color='#919191', size=10)
    field[3][3].drawObjectFieldElementCircle(position=[0, 0], color='#919191', size=10)
    field[3][5].drawObjectFieldElementCircle(position=[0, 38], color='#919191', size=10)

    field[0][3].drawObjectFieldElementCircle(position=[10, 0], color='#919191', size=10)
    field[5][3].drawObjectFieldElementCircle(position=[36, 0], color='#919191', size=10)


    #game objects

    #corner balls
    field[0][0].drawObjectFieldElementCircle(position=[24, 24], color='blue', size=7)
    field[0][5].drawObjectFieldElementCircle(position=[24, 24], color='blue', size=7)

    field[5][0].drawObjectFieldElementCircle(position=[24, 24], color='red', size=7)
    field[5][5].drawObjectFieldElementCircle(position=[24, 24], color='red', size=7)

    #middle balls
    field[3][2].drawObjectFieldElementCircle(position=[0, 29], color='blue', size=7)
    field[3][3].drawObjectFieldElementCircle(position=[18, 0], color='blue', size=7)

    field[3][3].drawObjectFieldElementCircle(position=[0, 18], color='red', size=7)
    field[2][3].drawObjectFieldElementCircle(position=[29, 0], color='red', size=7)

    #other balls
    field[3][1].drawObjectFieldElementCircle(position=[0, 24], color='blue', size=7)
    field[3][4].drawObjectFieldElementCircle(position=[0, 24], color='red', size=7)
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import tkinter as tk


class Tile:
    """
    gives ability to draw objects on each tile
    """
    def __init__(self, x, y, distance, canvas):
        self.canvas = canvas

        self.distance = distance

        self.P1 = [x, y]
        self.P2 = [x + distance, y]
        self.P3 = [x, y + distance]
        self.P4 = [x + distance, y + distance]


        self.grid = []
        x = self.P1[0]
        increment = self.distance/47
        for i in range(0, 48): #makes grid with coordinates
            column = []
            y = self.P1[1]
            for j in range(0, 48):
                coords = [x, y]
                column.append(coords)
                y = y + increment
            x = x + increment
            self.grid.append(column)



    def __makeCenteredGrid(self, size):
        centeredGrid = []
        x = self.P1[0]
        increment = (self.distance - size) / 2
        for i in range(0, 3): #makes grid with coordinates
            column = []
            y = self.P1[1]
            for j in range(0, 3):
                coords = [x, y]
                column.append(coords)
                y = y + increment
            x = x + increment
            centeredGrid.append(column)

        return centeredGrid


    def drawObjectTiles(self, position=[23, 23], size=24, color="#cccccc", outline=None, width=None):
        """
        draws object to fill a square of the grid in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)



    def drawObjectFieldElementSquare(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
        """
        draws object centered on the coordinate chosen in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)


    def drawObjectFieldElementCircle(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
        """
        draws object centered on the coordinate chosen in inches
        """
        x1 = self.grid[position[0]][position[1]][0]
        y1 = self.grid[position[0]][position[1]][1]

        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        x1 = x1 - (s/2)
        y1 = y1 - (s/2)

        x2 = self.grid[position[0]][position[1]][0] + (s/2)
        y2 = self.grid[position[0]][position[1]][1] + (s/2)

        self.canvas.create_oval(x1, y1, x2, y2, outline=outline, fill=color, width=width)




    def drawObjectFieldElementCentered(self, position=[1, 1], size=8, color="#cccccc", outline=None, width=None):
        """
        draws object contained only in tile in inches
        """
        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        centeredGrid = self.__makeCenteredGrid(s)

        x1 = centeredGrid[position[0]][position[0]][0]
        y1 = centeredGrid[position[0]][position[1]][1]

        x2 = centeredGrid[position[0]][position[0]][0] + s
        y2 = centeredGrid[position[0]][position[1]][1] + s

        self.canvas.create_rectangle(x1, y1, x2, y2, outline=outline, fill=color, width=width)


    def drawRectangle(self, vertexes=[[0, 1], [2, 1], [2, 1], [0, 1]], color="#cccccc", outline="white", width=3):
        """
        draws a rectangle given four vertices
        """
        points = [
            [
                self.grid[vertexes[0][0]][vertexes[0][0]][0],
                self.grid[vertexes[0][1]][vertexes[0][1]][1]
            ],
            [
                self.grid[vertexes[1][0]][vertexes[1][0]][0],
                self.grid[vertexes[1][1]][vertexes[1][1]][1]
            ],
            [
                self.grid[vertexes[2][0]][vertexes[2][0]][0],
                self.grid[vertexes[2][1]][vertexes[2][1]][1]
            ],
            [
                self.grid[vertexes[3][0]][vertexes[3][0]][0],
                self.grid[vertexes[3][1]][vertexes[3][1]][1]
            ]

            ]

        self.canvas.create_polygon(points, outline=outline, fill=color, width=width)



    def placeRobot(self, position=[1, 1], size=17):
        """
        draws robot in tile in inches
        """
        s = self.distance/25
        for i in range(0, size):
            s = s + (self.distance/25)

        #centeredGrid = self.__makeCenteredGrid(s)

        x1 = self.grid[position[0]][position[0]][0]
        y1 = self.grid[position[0]][position[1]][1]

        x2 = self.grid[position[0]][position[0]][0] + s
        y2 = self.grid[position[0]][position[1]][1] + s

        coords = [[x1, y1], [x2, y2]]
        return coords








class Field:
    """
    creates the field
    """
    def __init__(self, width, height):
        master = tk.Tk()
        self.master = master
        self.canvas = tk.Canvas(master, width=width, height=height)
        self.canvas.grid(row=0, column=0, columnspan=3)

        self.canvas.create_rectangle(0, 0, width, height, fill="#ffffff")

        distance = height - 100
        self.P1 = [((width-distance)/2), 50]
        self.P2 = [(((width-distance)/2) + (distance)), 50]
        self.P3 = [((width-distance)/2), (50+distance)]
        self.P4 = [(((width-distance)/2) + (distance)), (50+distance)]

        self.width = width
        self.height = height
        self.distance = distance

        self.field = []




    def drawField(self):
        """
        draws the field based on the resolution given
        """
        self.canvas.create_rectangle(self.P1[0], self.P1[1], self.P4[0], self.P4[1], fill="#cccccc")

        self.canvas.create_line(self.P1[0], self.P1[1], self.P2[0], self.P2[1], fill="black", width=7) #horizontal
        self.canvas.create_line(self.P1[0], self.P1[1] - 3, self.P3[0], self.P3[1] + 4, fill="black", width=7) #vertical

        self.canvas.create_line(self.P2[0], self.P2[1] - 3, self.P4[0], self.P4[1] + 4, fill="black", width=7) #vertical
        self.canvas.create_line(self.P3[0], self.P3[1], self.P4[0], self.P4[1], fill="black", width=7) #horizontal
    def drawGrid(self):
        """
        draws grid on the field
        """
        start = (self.width - self.distance) / 2
        increment = self.distance / 6

        for x in range(0, 6): #verical lines
            P1 = [start + (increment*x), 50]
            P2 = [start + (increment*x), (50+self.distance)]

            self.canvas.create_line(P1[0], P1[1], P2[0], P2[1], fill="black", width=2)

        start = 50
        for x in range(0, 6): #horizontal lines
            P1 = [((self.width - self.distance) / 2), start + (increment*x)]
            P2 = [(((self.width - self.distance) / 2) + self.distance), start + (increment*x)]

            self.canvas.create_line(P1[0], P1[1], P2[0], P2[1], fill="black", width=2)


    def fieldInfo(self):
        """
        creates objects of each tile and appends them to a list
        so that it is possible to create objects on the field
        """
        distance = self.distance/6
        x = ((self.width-self.distance)/2)

        for i in range(0, 6): #columns
            column = []
            y = 50
            for j in range(0, 6): #rows
                tileObject = Tile(x, y, distance, self.canvas)
                column.append(tileObject)

                y = y + distance
            x = x + distance

            self.field.append(column)
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jan  5 08:40:28 2021

@author: aiden
"""
import math
def to_radians(degrees):
    return ((degrees * math.pi) / 180)

def to_degrees(radians):
    return ((radians * 180) / math.pi)

current_x = -36
current_y = 0
current_angle = to_radians(-135)

end_x = 0
end_y = 0


dx = end_x - current_x
dy = end_y - current_y
print(to_degrees(math.atan2(dy, dx)))
dtheta = (math.atan2(dy, dx))
if dtheta < 0:
    dtheta += 2 * math.pi

if current_angle < 0:
    current_angle += 2 * math.pi

current_angle = -current_angle + (math.pi/2)
to_turn = current_angle - dtheta
print(to_degrees(current_angle), to_degrees(dtheta))
if to_turn > math.pi:
    to_turn = (-2 * math.pi) + to_turn
if to_turn < -math.pi:
    to_turn = (2 * math.pi) + to_turn

print(to_degrees(to_turn))
```

```
smtplib
getpass
tkinter

```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import tkinter as tk

class robotInfoFrame:
    """
    makes frame object for all labels and buttons
    """
    def __init__(self, master):
        self.master = master

        self.guiFrame = tk.Frame(self.master)
        self.guiFrame.grid(row=1, column=2, sticky='news')


        ##### labels #####

        # position label
        self.positionLabelText = tk.StringVar()
        self.positionLabel = tk.Label(self.guiFrame, textvariable=self.positionLabelText)

        #orientation label
        self.orientationLabelText = tk.StringVar()
        self.orientationLabel = tk.Label(self.guiFrame, textvariable=self.orientationLabelText)

        #distance moved label
        self.distanceMovedLabelText = tk.StringVar()
        self.distanceMovedLabel = tk.Label(self.guiFrame, textvariable=self.distanceMovedLabelText)

        #current command label
        self.commandLabelText = tk.StringVar()
        self.commandLabel = tk.Label(self.guiFrame, textvariable=self.commandLabelText)

        #white space
        self.whiteSpaceLabel = tk.Label(self.guiFrame)

        #trailing white space
        self.trailingWhiteSpaceLabel = tk.Label(self.guiFrame)




    def placeObjects(self):
        """
        places all labels and sets default value
        """
        self.positionLabel.grid(row=0, column=8, columnspan=1, sticky='w')
        self.orientationLabel.grid(row=1, column=8, columnspan=1, sticky='w')
        self.distanceMovedLabel.grid(row=2, column=8, columnspan=1, sticky='w')
        self.commandLabel.grid(row=3, column=8, columnspan=1, sticky='w')

        self.whiteSpaceLabel.grid(row=0, column=0, columnspan=7, rowspan=3, sticky='news')
        self.trailingWhiteSpaceLabel.grid(row=0, column=4, sticky='news')


        #configures trailing white space to be eaten
        self.guiFrame.grid_columnconfigure(4, weight=1)

        self.positionLabelText.set("(x: ?, y: ?)")
        self.orientationLabelText.set("orientation: ")
        self.distanceMovedLabelText.set("distance moved: N/A              ")
        self.commandLabelText.set("command: ")
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from PIL import Image
import io
import time
import math
import stopwatch


class robot:
    """
    contains all robot move functions
    """
    def __init__(self, fieldSize=None, tkobj=None, canvas=None, diameterOfWheel=4.1, controlPanelFrame=None, robotInfoFrame=None):
        self.controlPanelFrame = controlPanelFrame
        self.robotInfoFrame = robotInfoFrame

        self.canvas = canvas
        self.master = tkobj

        self.diameterOfWheel = diameterOfWheel
        self.fieldSize = fieldSize
        self.reversed = 0

        self.sqaure = None
        self.line = None

        self.squareVertexes = []
        self.lineVertexes = []

        self.iteration = 0

        self.x_offset_in = 0
        self.y_offset_in = 0
        self.theta_offset_deg = 0

    def __calcSleepTime(self, distance, iterations):
        return (distance / (10*(2 ** self.controlPanelFrame.speed))) / iterations


    def __generate_postscript(self):
        # ps = self.canvas.postscript(colormode="color")
        # im = Image.open(io.BytesIO(ps.encode('utf-8')))
        # im.save("ps/test" + str(self.iteration) + ".jpg")
        # self.iteration += 1
        pass

    def __calcCenters(self):
        """
        returns center of each polygon
        """
        xVals = []
        yVals = []

        for i in range(0, len(self.squareVertexes)): #calculates square center
            xVals.append(self.squareVertexes[i][0])
            yVals.append(self.squareVertexes[i][1])

        xVals.sort(reverse=True)
        yVals.sort(reverse=True)

        greatestX = xVals[0]
        greatestY = yVals[0]

        xVals.sort()
        yVals.sort()

        leastX = xVals[0]
        leastY = yVals[0]

        x = abs(((greatestX - leastX)/2)) + leastX
        y = abs(((greatestY - leastY)/2)) + leastY

        center = [x, y]


        return center





    def __rotate(self, Angle, pivotPoint):
        """
        rotates the robot simulating one side of the
        chassis moving
        """
        center = (pivotPoint[0], pivotPoint[1])
        angle = math.radians(Angle)
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center

        new_points = []
        for x_old, y_old in self.squareVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
            new_points.append([x_new + cx, y_new + cy])


        angle = math.radians(Angle)
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center

        new_points2 = []
        for x_old, y_old in self.lineVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
            new_points2.append([x_new + cx, y_new + cy])

        self.squareVertexes = new_points
        self.lineVertexes = new_points2




    def __rotateInPlace(self, Angle):
        """
        rotates the robot in place simulating both sides of the chassis
        moving
        """
        center = self.__calcCenters()


        center = (center[0], center[1])

        angle = math.radians(Angle) #moves square
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center
        new_points = []
        for x_old, y_old in self.squareVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
            new_points.append([x_new + cx, y_new + cy])

        angle = math.radians(Angle) #moves line
        cos_val = math.cos(angle)
        sin_val = math.sin(angle)
        cx, cy = center
        new_points2 = []
        for x_old, y_old in self.lineVertexes:
            x_old -= cx
            y_old -= cy
            x_new = x_old * cos_val - y_old * sin_val
            y_new = x_old * sin_val + y_old * cos_val
            new_points2.append([x_new + cx, y_new + cy])


        self.squareVertexes = new_points
        self.lineVertexes = new_points2



    def __move(self, units):
        """
        simulates the robot moving in a straight line
        """
        quadrants = {  #quadrant: [xVal, yVal]
            1:[1, 1],
            2:[-1, 1],
            3:[-1, -1],
            4:[1, -1]
            }


        #determine quadrant of final position
        #used to determine if robot needs to add or subtract
        #x and y value to move in that direction

        if units > 0 and (self.orientationDegrees >= 0 and self.orientationDegrees < 90):
            quadrant = 1
        elif units < 0 and (self.orientationDegrees >= 180 and self.orientationDegrees <= 270):
            quadrant = 1

        elif units > 0 and (self.orientationDegrees >= 90 and self.orientationDegrees <= 180):
            quadrant = 2
        elif units < 0 and (self.orientationDegrees > 270 and self.orientationDegrees < 360):
            quadrant = 2

        elif units > 0 and (self.orientationDegrees >= 180 and self.orientationDegrees < 270):
            quadrant = 3
        elif units < 0 and (self.orientationDegrees >= 0 and self.orientationDegrees <= 90):
            quadrant = 3

        elif units > 0 and (self.orientationDegrees >= 270 and self.orientationDegrees < 360):
            quadrant = 4
        elif units < 0 and (self.orientationDegrees > 90 and self.orientationDegrees < 180):
            quadrant = 4



        vals = quadrants.get(quadrant)
        xPol = vals[0]
        yPol = vals[1]

        #absolute value simulates reference angle
        #trig functions of reference angles are positive
        x = xPol * abs(math.cos(math.radians(self.orientationDegrees)))
        y = yPol * abs(math.sin(math.radians(self.orientationDegrees)))

        d = (math.sqrt((x**2) + (y**2)))
        distanceMoved = 0

        stp = stopwatch.stopwatch()
        stp.start()

        iterations = int(abs(units)/d)
        for i in range(0, iterations): #move animation
            if not self.controlPanelFrame.keepRunning: #allows for pause
                while not self.controlPanelFrame.keepRunning:
                    time.sleep(0.1)
                    self.master.update()

            self.canvas.move(self.square, x, y)
            self.canvas.move(self.line, x, y)

            self.master.update()
            self.__generate_postscript()
            time.sleep(self.__calcSleepTime(self.__encoderTicks(abs(units)), iterations))

            eus = self.__encoderTicks(d)  # shows distance moved
            distanceMoved = distanceMoved + eus

            #updates vertices of square and line
            xChange = (xPol * abs(math.cos(math.radians(self.orientationDegrees)) * abs(units))) / iterations
            yChange = (yPol * abs(math.sin(math.radians(self.orientationDegrees)) * abs(units))) / iterations

            self.squareVertexes = [
                [self.squareVertexes[0][0] + xChange, self.squareVertexes[0][1] + yChange],
                [self.squareVertexes[1][0] + xChange, self.squareVertexes[1][1] + yChange],
                [self.squareVertexes[2][0] + xChange, self.squareVertexes[2][1] + yChange],
                [self.squareVertexes[3][0] + xChange, self.squareVertexes[3][1] + yChange]
                ]

            self.lineVertexes = [
                [self.lineVertexes[0][0] + xChange, self.lineVertexes[0][1] + yChange],
                [self.lineVertexes[1][0] + xChange, self.lineVertexes[1][1] + yChange],
                [self.lineVertexes[2][0] + xChange, self.lineVertexes[2][1] + yChange],
                [self.lineVertexes[3][0] + xChange, self.lineVertexes[3][1] + yChange]
                ]

            self.__update()

            self.__updateDistanceLabel(str(round(distanceMoved, 2)) ,"encoder ticks")
            self.__update_position_label()





    def __update(self):
        """
        updates tkinter canvas so robot can be seen moving
        and also allows for a pause
        """
        self.canvas.delete(self.square)
        self.canvas.delete(self.line)

        self.square = self.canvas.create_polygon(self.squareVertexes,
                                                 outline="black",
                                                 fill="#949596",
                                                 width=3)

        self.line = self.canvas.create_polygon(self.lineVertexes, width=3)


        if not self.controlPanelFrame.keepRunning: #allows for pause during turns
            while not self.controlPanelFrame.keepRunning:
                time.sleep(0.1)
                self.master.update()

        self.master.update()


    def __updateDistanceLabel(self, distance=0, units=""):
        """
        updates distance travelled label
        """
        text = "distance moved: " + str(distance) + " " + units
        while len(text) < 38:
            text = text + " "
        self.robotInfoFrame.distanceMovedLabelText.set(text)

        self.master.update()

    def __update_orientation_label(self, units="degrees"):
        angle = self.orientationDegrees
        if units == "radians":
            angle = self.__to_radians(angle)
        text = "Orientation: " + str(angle)
        self.robotInfoFrame.orientationLabelText.set(text)


    def __update_position_label(self):
        x = round(self.inches(self.__calcCenters()[1]) - self.x_offset_in, 2)
        y = round(self.inches(self.__calcCenters()[0]) - self.y_offset_in, 2)

        text = "(x: " + str(x) + ", y: " + str(y) + ")"

        self.robotInfoFrame.positionLabelText.set(text)


    def __pixels(self, rotationUnits):
        """
        converts encoder ticks to pixels
        """
        revolutions = rotationUnits / 360
        inches = revolutions * (self.diameterOfWheel * math.pi)
        pixelsToMove = (inches * self.fieldSize) / 144
        # print(rotationUnits, pixelsToMove)
        return pixelsToMove


    def __encoderTicks(self, pixels):
        """
        converts pixels to encoder ticks
        """
        inches = pixels * (144 / self.fieldSize)
        revolutions = inches / (self.diameterOfWheel * math.pi)
        encoderTicks = revolutions * 360

        return encoderTicks

    def __to_radians(self, degrees):
        return ((degrees * math.pi) / 180)

    def __to_degrees(self, radians):
        return ((radians * 180) / math.pi)

    def __in_to_encoder_ticks(self, inches):
        circumference = (self.diameterOfWheel * math.pi);
        revolutions = inches / circumference;
        encoder_ticks = revolutions * 360;

        return encoder_ticks;


    def inches(self, pixels):
        """
        converts pixels to inches
        """
        inches = pixels * (144 / self.fieldSize)

        return inches


    def show(self, angle=0, position=[[0, 0], [0, 0]]):
        """
        starting function that shows the robot based on two coordinates
        if more than two coordinates are given use different show function
        """

        #if len(position) > 2:
        if 1:
            x1 = position[0][0]
            y1 = position[0][1]
            x2 = position[1][0]
            y2 = position[1][1]

            self.orientationDegrees = 90
            self.squareVertexes = [
                [x1, y1],
                [x2, y1],
                [x2, y2],
                [x1, y2]
                ]

            self.sizeOfSquare = abs(self.squareVertexes[0][0] - self.squareVertexes[1][0])

            y2 = (y2 - y1) / 4
            self.lineVertexes = [
                [x1, y1+y2],
                [x2, y1+y2],
                [x2, y1+y2+4],
                [x1, y1+y2+4]
                ]

#        else:
#            x1 = position[0][0]
#            x2 = position[1][0]
#            x3 = position[2][0]
#            x4 = position[3][0]
#
#            y1 = position[0][1]
#            y2 = position[1][1]
#            y3 = position[2][1]
#            y4 = position[3][1]
#
#            self.squareVertexes = position
#            self.lineVertexes = []


        self.square = self.canvas.create_polygon(self.squareVertexes,
                                                 outline="black",
                                                 fill="#949596",
                                                 width=3)

        turn = 90 - (angle % 360)

        self.line = self.canvas.create_polygon(self.lineVertexes, width=3)
        self.__rotateInPlace(turn)
        self.__update()
        self.__generate_postscript()

        self.orientationDegrees = angle % 360


        self.x_offset_in = self.inches(self.__calcCenters()[1])
        self.y_offset_in = self.inches(self.__calcCenters()[0])
        self.theta_offset_deg = self.orientationDegrees




    def reverse(self):
        """
        reverses orientation of the robot
        """
        self.reversed = not(self.reversed)




    def forward(self, rotationUnits):
        """
        moves the robot forward and straight
        """
        self.robotInfoFrame.commandLabelText.set(("forward " + str(rotationUnits)))

        if self.reversed:
            rotationUnits = 0 - rotationUnits

        pixelsToMove = self.__pixels((rotationUnits))

        self.__move(pixelsToMove)
        self.__update()




    def backward(self, rotationUnits):
        """
        moves the robot backwards and straight
        """
        self.robotInfoFrame.commandLabelText.set(("backward " + str(rotationUnits)))

        rotationUnits = 0 - rotationUnits
        if self.reversed:
            rotationUnits = 0 - rotationUnits

        pixelsToMove = self.__pixels(rotationUnits)


        self.__move(pixelsToMove)
        self.__update()


    def leftSide(self, angle):
        """
        turns the robot right so that only one side is moving
        """
        self.robotInfoFrame.commandLabelText.set(("leftSide " + str(angle)))

        if self.reversed:
            angle = 0 - angle

        pivotPoints = self.squareVertexes[2]

        turned = 0
        orientation = angle / abs(angle) #to account for negative turns
        toMove = orientation * .5
        while turned < abs(angle): #turn to specified angle
            self.__rotate(toMove, pivotPoints)
            self.__update()
            self.__generate_postscript()

            time.sleep(self.__calcSleepTime(angle, angle))

            turned += .5
            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")

            self.orientationDegrees = (self.orientationDegrees - toMove) % 360
            self.robotInfoFrame.orientationLabelText.set("orientation: " + str(self.orientationDegrees))
            self.__update_position_label()


    def rightSide(self, angle):
        """
        turns the robot left so that only the right side is moving
        """
        self.robotInfoFrame.commandLabelText.set(("rightSide " + str(angle)))

        angle = 0 - angle
        if self.reversed:
            angle = 0 - angle

        pivotPoints = self.squareVertexes[3]

        turned = 0
        orientation = angle / abs(angle) #to account for negative turns
        toMove = (-1 * orientation * .5)
        while turned < abs(angle): #turn to specified angle
            self.__rotate(toMove, pivotPoints)
            self.__update()
            self.__generate_postscript()

            time.sleep(self.__calcSleepTime(angle, angle))
            turned += .5
            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")

            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
            self.__update_orientation_label()
            self.__update_position_label()



    def turnLeft(self, angle):
        """
        turn in place left
        """
        self.robotInfoFrame.commandLabelText.set(("turnLeft " + str(angle)))

        if self.reversed:
            angle = 0 - angle

        turned = 0
        orientation = angle / abs(angle) #to account for negative turns
        toMove = -1 * orientation * .5

        while turned < abs(angle): #turn to specified angle
            self.__rotateInPlace(toMove)
            self.__update()
            self.__generate_postscript()

            time.sleep(self.__calcSleepTime(angle, angle))
            turned += .5
            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")

            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
            self.__update_orientation_label()
            self.__update_position_label()




    def turnRight(self, angle):
        """
        turn in place right
        """
        self.robotInfoFrame.commandLabelText.set(("turnRight " + str(angle)))

        if self.reversed:
            angle = 0 - angle

        turned = 0
        orientation = angle / abs(angle) #to account for negative turns
        toMove = .5 * orientation

        while turned < abs(angle): #turn to specified angle
            self.__rotateInPlace(toMove)
            self.__update()
            self.__generate_postscript()

            time.sleep(self.__calcSleepTime(angle*2, angle))
            turned += .5
            self.__updateDistanceLabel(str(round(turned, 2)), "degrees")


            self.orientationDegrees = (self.orientationDegrees + toMove) % 360
            self.__update_orientation_label()
            self.__update_position_label()




    def drive_to_point(self, x, y, explicit_direction=0):
        """
        drive to a point
        """
        self.robotInfoFrame.commandLabelText.set(("drive_to_point (" + str(x) + ", " + str(y) + ")"))


        current_x = self.inches(self.__calcCenters()[1]) - self.x_offset_in
        current_y = self.inches(self.__calcCenters()[0]) - self.y_offset_in
        current_angle = self.__to_radians(self.orientationDegrees - self.theta_offset_deg)

        end_x = x
        end_y = y


        dx = end_x - current_x
        dy = end_y - current_y
        print("current coords: ", current_x, current_y, self.orientationDegrees - self.theta_offset_deg)
        print("dx,dy:", dx, dy)
        print("end coords:", end_x, end_y)
        # print(self.__to_degrees(math.atan2(dy, dx)))
        dtheta = (math.atan2(dy, dx))
        if dtheta < 0:
            dtheta += 2 * math.pi

        if current_angle < 0:
            current_angle += 2 * math.pi

        current_angle = -current_angle + (math.pi/2)
        to_turn_face_forwards = current_angle - dtheta
        to_turn_face_backwards = to_turn_face_forwards - math.pi
        # print(self.__to_degrees(current_angle), self.__to_degrees(dtheta))
        if to_turn_face_forwards > math.pi:
            to_turn_face_forwards = (-2 * math.pi) + to_turn_face_forwards
        elif to_turn_face_forwards < -math.pi:
            to_turn_face_forwards = (2 * math.pi) + to_turn_face_forwards

        if to_turn_face_backwards > math.pi:
            to_turn_face_backwards = (-2 * math.pi) + to_turn_face_backwards
        elif to_turn_face_backwards < -math.pi:
            to_turn_face_backwards = (2 * math.pi) + to_turn_face_backwards

        if explicit_direction == 1:
            to_turn = to_turn_face_forwards
            direction = 1
        elif explicit_direction == -1:
            to_turn = to_turn_face_backwards
            direction = -1
        elif abs(to_turn_face_forwards) < abs(to_turn_face_backwards):
            to_turn = to_turn_face_forwards
            direction = 1
        else:
            to_turn = to_turn_face_backwards
            direction = -1

        to_drive_inches = math.sqrt((dx**2) + (dy**2))
        to_drive_enc = direction * self.__in_to_encoder_ticks(to_drive_inches)


        # perform turn command
        angle = self.__to_degrees(to_turn)
        print("to turn: ", angle)
        if angle != 0:
            if self.reversed:
                angle = 0 - angle

            turned = 0
            orientation = angle / abs(angle) #to account for negative turns
            toMove = .5 * orientation

            while turned < abs(angle): #turn to specified angle
                self.__rotateInPlace(toMove)
                self.__update()
                self.__generate_postscript()

                time.sleep(self.__calcSleepTime(angle*2, angle))
                turned += .5
                self.__updateDistanceLabel(str(round(turned, 2)), "degrees")


                self.orientationDegrees = (self.orientationDegrees + toMove) % 360
                self.__update_orientation_label()
                self.__update_position_label()



        # perform drive command
        if self.reversed:
            to_drive_enc = 0 - to_drive_enc

        pixelsToMove = self.__pixels((to_drive_enc))

        self.__move(pixelsToMove)
        self.__update()
```

../AutonSimulator/runningFrame.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import tkinter as tk

class runningFrame:
    """
    makes frame for running label
    """
    def __init__(self, master):
        self.master = master

        self.guiFrame = tk.Frame(self.master)
        self.guiFrame.grid(row=2, column=0, sticky='news')


        ##### labels #####

        #running label
        self.runningLabelText = tk.StringVar()
        self.runningLabel = tk.Label(self.guiFrame, textvariable=self.runningLabelText, font=("Courier", 15))



    def placeObjects(self):
        """
        places labels
        """

        self.runningLabel.grid(sticky='w', row=0, column=0)

        self.guiFrame.grid_columnconfigure(0, weight=1)
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import time

class stopwatch:

    def __init__(self):
        self.beginning = None

    def start(self):
        """
        starts stopwatch
        """
        self.beginning = time.time()

    def stop(self):
        """
        stops stopwatch and returns
        time elapsed
        """
        end = time.time()
        timeElapsed = end - self.beginning

        return timeElapsed
```

```
Update creds.json for google docs spreadsheet that will be modified

in "Scout.py" change sku in class constructor to sku of tournament
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 24 22:08:27 2019

@author: aiden
"""

import requests
import json
import stopwatch
from usefultools import split
import multiprocessing as mp
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
import time


class getData:
    def __init__(self):
        self.sku = ""

        self.sheet_name = ""

        self.matchesWonUrl = 'https://api.vexdb.io/v1/get_matches?round=2'
        self.skillsUrl = 'https://api.vexdb.io/v1/get_skills?'
        self.pointsUrl = 'https://api.vexdb.io/v1/get_rankings?sku=' + self.sku

        self.teams = []
        self.allTeams = []
        self.ranges ={}

        self.matchesWonData = {}
        self.matchesWonOverallData = {}
        self.combinedSkillsScoreData = {}
        self.driverSkillsScoreData = {}
        self.pointsData = {}
        self.rankingData = {}

        self.NUM_PROCESSES = 20
        self.TEAM_NUM_COL = 2

        self.sheet = None


    def __calcWin(self, team, data):
        """
        calculates if a match was won or not by finding what
        color a team was and the score of the match
        it is a win if the teams color score more points
        """
        colors = ['red1', 'red2', 'blue1', 'blue2']
        teamColor = ''
        for color in colors:
            if data.get(color) == team:
                c = color.split('1')[0]
                c = c.split('2')[0]
                teamColor = c
                break

        redScore = data.get('redscore')
        blueScore = data.get('bluescore')

        if redScore != blueScore:
            if redScore > blueScore:
                winner = 'red'
            else:
                winner = 'blue'

            if teamColor == 'red' and winner == 'red':
                return 1
            elif teamColor == 'blue' and winner == 'blue':
                return 1
            else:
                return 0

        elif redScore == blueScore and (redScore != 0 and blueScore != 0):
            return -1

        else:
            return None




    def __getMatchesWon(self, teams, q):
        """
        gets win/loss/tie data for a team
        """
        for team in teams:
            team = team.split('\n')[0]
            url = self.matchesWonUrl + '&team=' + team + '&season=Tower Takeover'

            data = requests.get(url)
            if data.status_code == 200 and team != '':

                winStruct = {'wins':0,
                             'losses':0,
                             'ties':0
                            }

                data = json.loads(data.content.decode('utf-8'))
                data = data.get('result')

                for entry in data:
#                   print(team, entry)
                    x = self.__calcWin(team, entry)
                    if x == 1:
                        winStruct['wins'] = winStruct.get('wins') + 1
                    elif x == 0:
                        winStruct['losses'] = winStruct.get('losses') + 1
                    elif x == -1:
                        winStruct['ties'] = winStruct.get('ties') + 1
                    else:
                        pass
                q.put({team: winStruct})



    def __getDriverSkillsScore(self, teams, q):
        """
        gets driver skills score for a team
        """
        for team in teams:
            team = team.split('\n')[0]
            url = self.skillsUrl + 'team=' + team + '&season=Tower Takeover&type=0'

            data = requests.get(url)
            if data.status_code == 200:

                data = json.loads(data.content.decode('utf-8'))
                data = data.get('result')
                highest = 0

                for item in data:
                    value = item.get('score')
                    if value > highest:
                        highest = value

                q.put({team: highest})


    def __getCombinedSkillsScore(self, teams, q):
        """
        gets the combined skills score for a team
        (driver and programming)
        """
        for team in teams:
            team = team.split('\n')[0]
            url2 = self.skillsUrl + 'team=' + team + '&season=Tower Takeover&type=2'

            data = requests.get(url2)
            if data.status_code == 200:


                data = json.loads(data.content.decode('utf-8'))
                data = data.get('result')
                highest = 0

                for item in data:
                    value = item.get('score')
                    if value > highest:
                        highest = value
                q.put({team: highest})


    def __getPoints(self, teams, q):
        """
        gets win points/auton points/strength points/calculated
        contribution to win margin data for a team
        """
        for team in teams:
            team = team.split('\n')[0]
            url = self.pointsUrl + '&team=' + team + '&season=Tower Takeover&type=2'

            data = requests.get(url)
            if data.status_code == 200:
                try:
                    data = json.loads(data.content.decode('utf-8'))
                    data = data.get('result')[0]
                    pointsStruct = {
                            'wp':data.get('wp'),
                            'ap':data.get('ap'),
                            'sp':data.get('sp'),
                            'ccwm':data.get('ccwm')
                            }
                    q.put({team: pointsStruct})

                except IndexError:
                    q.put({team: 'N/A'})

    def __getRanking(self, teams, q):
        """
        gets the ranking of a team at an event
        """
        for team in teams:
            team = team.split('\n')[0]
            url = self.pointsUrl + '&team=' + team + '&season=Tower Takeover&type=2'
            #print(url)

            data = requests.get(url)
            if data.status_code == 200:

                data = json.loads(data.content.decode('utf-8'))
                try:
                    data = data.get('result')[0]
                    q.put({team: data.get('rank')})

                except IndexError:
                    q.put({team: 'N/A'})


    def __parralellise(self, func, returnDict):
        """
        processes the data in parrallel so that if there is a lot of
        teams the operation can be performed quickly
        """
        queues = []
        processes = []
        for i in range(len(self.teams)):
            queues.append(mp.Queue())
            p = mp.Process(target=func, args=(self.teams[i], queues[i],))
            processes.append(p)
            p.daemon = True
            p.start()
        for process in processes:
            process.join()

        for q in queues:
            while not q.empty():
                returnDict.update(q.get(timeout=.1))





    def openSheet(self):
        """
        opens the sheet
        need to change workbook to the sheet that will be edited
        and the sheet name to the sheet that corresponds with the
        data that will be collected
        """
        json_key = json.load(open('/home/aiden/Documents/google_credentials/creds.json'))
        scope = ['https://spreadsheets.google.com/feeds',
        'https://www.googleapis.com/auth/drive']

        credentials = SignedJwtAssertionCredentials(json_key['client_email'], json_key['private_key'].encode(), scope)

        file = gspread.authorize(credentials)
        workbook = file.open("536C_Scouting")
        self.sheet = workbook.worksheet(self.sheet_name)





    def getTeams(self):
        """
        gets the teams by reading the data in the sheet
        this looks for data to find by comparing the first
        row to a list of valid headers then removes the headers
        from the list of cells to be updated
        """
        valid_headers = [
                'Rank',
                'WP/AP/SP/CCWM',
                'W-L-T (today)',
                'W-L-T (overall season)',
                'driver skills',
                'combined skills score'
                ]

        self.allTeams = self.sheet.col_values(1) #column that teams are stored in
        self.allTeams.pop(0) #remove header
        headers = self.sheet.row_values(1) #remove header
        row = 2
        column = 1
        for header in headers:
            if header in valid_headers:
                start = chr(ord('@')+column) + str(row)
                end = chr(ord('@')+column) + str((len(self.allTeams) + 1))
                string = start + ':' + end
                range_= self.sheet.range(string)
                self.ranges.update({header:range_})
            column += 1

        #limits num processes to the number of teams
        if self.NUM_PROCESSES > len(self.allTeams):
            self.NUM_PROCESSES = len(self.allTeams)

        self.teams = list(split.split(self.allTeams, self.NUM_PROCESSES))





    def collect(self):
        """
        collects all the data if that data is a valid header
        """
        if 'W-L-T (overall season)' in self.ranges.keys():
            self.__parralellise(self.__getMatchesWon, self.matchesWonOverallData)

        self.matchesWonUrl = self.matchesWonUrl + '&sku=' + self.sku

        if 'W-L-T (today)' in self.ranges.keys():
            self.__parralellise(self.__getMatchesWon, self.matchesWonData)

        if 'combined skills score' in self.ranges.keys():
            self.__parralellise(self.__getCombinedSkillsScore, self.combinedSkillsScoreData)
        if 'driver skills' in self.ranges.keys():
            self.__parralellise(self.__getDriverSkillsScore, self.driverSkillsScoreData)
        if 'WP/AP/SP/CCWM' in self.ranges.keys():
            self.__parralellise(self.__getPoints, self.pointsData)
        if 'Rank' in self.ranges.keys():
            self.__parralellise(self.__getRanking, self.rankingData)



    def printData(self):
        """
        prints the data for each team to be used for debugging
        purposes
        """
        if self.matchesWonData:
            #print("w-l-t")
            for team in self.matchesWonData.keys():
                try:
                    wins = self.matchesWonData.get(team)
                    record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
                    formattedRecord = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])
                    print(formattedRecord)
                except AttributeError:
                    print('N/A')
            for i in range(20):
                print("")

        if self.combinedSkillsScoreData:
            print("combined skills")
            for team in self.combinedSkillsScoreData.keys():
                print(self.combinedSkillsScoreData.get(team))


            for i in range(20):
                print("")

        if self.driverSkillsScoreData:
            print("driver skills")
            for team in self.driverSkillsScoreData.keys():
                print(self.driverSkillsScoreData.get(team))

            for i in range(20):
                print("")


        if self.rankingData:
            print("ranking")
            for team in self.rankingData.keys():
                print(self.rankingData.get(team))

            for i in range(20):
                print("")


        if self.pointsData:
            print("point values")
            for team in self.pointsData.keys():
                try:
                    val = self.pointsData.get(team)
                    points = (str(val.get('wp'))  + ' / '
                            + str(val.get('ap')) + ' / '
                            + str(val.get('sp')) + ' / '
                            + str(val.get('ccwm'))
                            )
                    print(points)
                except AttributeError:
                    print('N/A')
            for i in range(6):
                print("")





    def writeData(self):
        """
        writes the data in one chunk because the google api
        only allows so many edits
        by making it only one edit once all the data is collected
        this constraing can be worked around
        """
        cell_list = self.ranges.get('Rank') #write rank
        row = 0
        for cell in cell_list:
            try:
                team = self.allTeams[row].split('\n')[0]
                data = self.rankingData.get(team)
            except:
                data = "N/A"
            cell.value = data
            row += 1

        self.sheet.update_cells(cell_list)


        cell_list = self.ranges.get('WP/AP/SP/CCWM') #write points
        row = 0
        for cell in cell_list:
            try:
                team = self.allTeams[row].split('\n')[0]
                val = self.pointsData.get(team)
                data = (str(val.get('wp'))  + ' / '
                        + str(val.get('ap')) + ' / '
                        + str(val.get('sp')) + ' / '
                        + str(val.get('ccwm'))
                        )
                cell.value = data
            except AttributeError:
                cell.value = 'N/A'
            except IndexError:
                cell.value = 'N/A'
            row += 1
        self.sheet.update_cells(cell_list)



        cell_list = self.ranges.get('W-L-T (today)') #write w-l-t
        row = 0
        for cell in cell_list:
            try:
                team = self.allTeams[row].split('\n')[0]
                wins = self.matchesWonData.get(team)
                record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
                data = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])

                cell.value = data
            except AttributeError:
                cell.value = 'N/A'
            row += 1

        self.sheet.update_cells(cell_list)



        cell_list = self.ranges.get('W-L-T (overall season)') #write w-l-t for
        row = 0                                                #overall season
        for cell in cell_list:
            try:
                team = self.allTeams[row].split('\n')[0]
                wins = self.matchesWonOverallData.get(team)
                record = [wins.get('wins'), wins.get('losses'), wins.get('ties')]
                data = str(record[0]) + '-' + str(record[1]) + '-' + str(record[2])

                cell.value = data
            except AttributeError:
                cell.value = 'N/A'
            row += 1

        self.sheet.update_cells(cell_list)




        cell_list = self.ranges.get('driver skills')#write driver skills
        row = 0
        for cell in cell_list:
            team = self.allTeams[row].split('\n')[0]
            data = self.driverSkillsScoreData.get(team)
            cell.value = data

            row += 1

        self.sheet.update_cells(cell_list)



        cell_list = self.ranges.get('combined skills score') #write combined skills
        row = 0
        for cell in cell_list:
            team = self.allTeams[row].split('\n')[0]
            data = self.combinedSkillsScoreData.get(team)
            cell.value = data

            row += 1

        self.sheet.update_cells(cell_list)









while 1: #collect data every so often
    stp = stopwatch.stopwatch()
    stp.start()
    d = getData()
    d.openSheet()

    d.getTeams()
    d.collect()
    #print(d.rankingData)
    d.writeData()
    print("time taken:", stp.stop(), "sec")

    time.sleep(45)



```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 28 11:36:52 2019

@author: aiden
"""
import requests
import json
import stopwatch
from usefultools import split
import multiprocessing as mp
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
import time


class getTeams:
    """
    class for getting teams at a tournament
    based on the tournaments sku
    """
    def __init__(self):
        self.sheet_name = ""

        sku = ""

        self.url = "https://api.vexdb.io/v1/get_teams?round=5?&sku=" + sku
        self.elimsUrl = "https://api.vexdb.io/v1/get_teams?round=5?&sku=" + sku + '&matchnum='

        #legacy version that only works if matches have started
        #self.url = 'https://api.vexdb.io/v1/get_matches?round=2?&sku=' + sku
        #self.elimsUrl = 'https://api.vexdb.io/v1/get_matches?&sku=' + sku + '&matchnum='

        self.allTeams = []
        self.elimTeams = []

        self.COLLECT_ELIMS = 0
        self.COLLECT_TEAMS = 1

        self.NUM_PROCESSES = 50

        self.sheet = None

    def __getAllTeams(self, dicts, queue):
        """
        gets all teams that are registered
        """
        for entry in dicts:
            print(entry.get("number"))
            queue.put(entry.get("number"))



    def __getElimTeams(self, dicts, queue):
        """
        gets teams that are in the elimination matches
        does not work need to update
        """
        print(dicts)
        for entry in dicts:
            print(entry.get("number"))
            queue.put(entry.get("number"))


    def __parralellise(self, func, returnList, dicts):
        """
        starts threads that look through lists of entries
        for teams that are at the tournament
        this is used so that at events like worlds it does
        not take forever to run
        """
        queues = []
        processes = []
        for i in range(len(dicts)):
            queues.append(mp.Queue())
            p = mp.Process(target=func, args=(dicts[i], queues[i],))
            processes.append(p)
            p.daemon = True
            p.start()
        for process in processes:
            process.join()

        for q in queues:
            while not q.empty():
                returnList.append(q.get(timeout=.1))


    def collect(self):
        """
        collects the data from vexdb and splits it into
        entries based on self.NUM_PROCESSES so that data
        can be parsed faster especially for events like worlds
        """
        data = requests.get(self.url)
        if data.status_code == 200 and self.COLLECT_TEAMS:
            data = json.loads(data.content.decode('utf-8'))
            data = data.get('result')
            data = list(split.split(data, self.NUM_PROCESSES))

            self.__parralellise(self.__getAllTeams, self.allTeams, data)

            self.allTeams = list(set(self.allTeams))


        if self.COLLECT_ELIMS:
            links = []
            for num in range(1, 9):
                match = 'R16 #' + str(num) + '-1'
                link = self.elimsUrl + match
                links.append(link)

            data = []

            for url in links:
                response = requests.get(url)
                if response.status_code == 200:
                    response = json.loads(response.content.decode(’utf-8’))
                    allData = response.get(’result’) + data #merge lists


    #       data = requests.get(self.elimsUrl)
    #       if data.status_code == 200 and self.COLLECT_ELIMS:
    #           data = json.loads(data.content.decode(’utf-8’))
    #           data = data.get(’result’)
            data = list(split.split(allData, self.NUM_PROCESSES))

            self.__parralellise(self.__getElimTeams, self.elimTeams, data)

            self.elimTeams = list(set(self.elimTeams))









    def printTeams(self):
        """
        prints the teams out
        used for debugging
        """
        for team in self.allTeams:
            print(team)

        print("")
        print("")

        for team in self.elimTeams:
            print(team)



    def openSheet(self):
        """
        opens the sheet and loads the work book
        need to change the workbook to the file name
        and the sheet to the sheet that will be edited
        """
        json_key = json.load(open(’/home/aiden/Documents/google_credentials/creds.json’))
        scope = scope = [’https://spreadsheets.google.com/feeds’,
                         ’https://www.googleapis.com/auth/drive’]

        credentials = SignedJwtAssertionCredentials(json_key[’client_email’], json_key[’private_key’].encode(), scope)

        file = gspread.authorize(credentials)
        workbook = file.open("536C_Scouting")
        self.sheet = workbook.worksheet(self.sheet_name)


    def writeData(self, column):
        """
        writes the data in one chunk because the api only allows
        so many operations
        """
        if self.COLLECT_TEAMS:
            #leave room for header by setting to two and adding 1
            start = chr(ord(’@’)+column) + ’2’
            end = chr(ord(’@’)+column) + str(len(self.allTeams) + 1)
            rang = start + ’:’ + end
            print(rang)
            cell_list = self.sheet.range(rang)

            x = 0
            for cell in cell_list:
                cell.value = self.allTeams[x]
                x += 1

            self.sheet.update_cells(cell_list)


        if self.COLLECT_ELIMS:
            start = chr(ord(’@’)+column) + ’2’
            end = chr(ord(’@’)+column) + str(len(self.elimTeams) + 1)
            rang = start + ’:’ + end
            print(rang)
            cell_list = self.sheet.range(rang)

            x = 0
            for cell in cell_list:
                cell.value = self.elimTeams[x]
                x += 1

            self.sheet.update_cells(cell_list)




g = getTeams()
g.collect()
g.printTeams()
g.openSheet()
print("sheet opened")
g.writeData(1)
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import time

class stopwatch:

    def __init__(self):
        self.beginning = None

    def start(self):
        """
        starts stopwatch
        """
        self.beginning = time.time()

    def stop(self):
        """
        stops stopwatch and returns
        time elapsed
        """
        end = time.time()
        timeElapsed = end - self.beginning

        return timeElapsed
```

```
#!/bin/bash
prosv5 make
prosv5 upload --slot 2
prosv5 v5 run 2
prosv5 terminal
```

```
{
    ”internal_motor_pid”:[1, 0, 0, 0],
    ”tilter_pid_consts”:[1, 0, 0, 0],

    "front_right_port":20,
    "back_left_port":4,
    "front_left_port":18,
    "back_right_port":2,
    "left_intake_port":6,
    "right_intake_port":8,
    "tilter_port":19,
    "lift_port":9,

    "front_right_reversed":0,
    "back_left_reversed":1,
    "front_left_reversed":1,
    "back_right_reversed":0,
    "left_intake_reversed":1,
    "right_intake_reversed":0,
    "tilter_reversed":1,
    "lift_reversed":0,

    "tilter_setpoints":[100, 300, 400, 500],
    "lift_setpoints":[100, 300, 400, 500],
    "intake_speeds":[-63, -30, 0, 30, 63],

    "autons":{
        "auton1":"location_of_auton1.json",
        "auton2":"location_of_auton2.json",
        "auton3":"location_of_auton3.json",
        "auton4":"location_of_auton4.json"
    }
}
```

```
#!/bin/bash
echo "paste stack locations"
while true;
do
   read -p "" stack
   arm-none-eabi-addr2line --demangle --inlines -faps -e bin/monolith.elf $stack
done

```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar  1 13:05:32 2020

@author: aiden
"""

import datetime
import glob

header_files = glob.glob("src/**/" + ’*.hpp’, recursive=True)
impl_files = glob.glob("src/**/" + ’*.cpp’, recursive=True)

files = sorted(header_files + impl_files)
todo_comments = {}
review_dates = {}
for file in files:
    todo_comments.update({file:[]})
    with open(file) as f:
        for line in f.readlines():
            if "TODO: " in line:
                comment = line.split("TODO: ")[-1].strip()
                todo_comments[file].append(comment)
            elif "@reviewed_on: " in line:
                date = line.split("@reviewed_on: ")[-1].strip()
                if date:
                    try:
                        date = datetime.datetime.strptime(date, "%m/%d/%Y")
                    except ValueError:
                        date = datetime.datetime.strptime(date, "%m/%d/%y")
                else:
                    date = False
                review_dates.update({file:date})

i1 = 0
i2 = 0

for file in todo_comments:
    if todo_comments.get(file):
        print(file)
        for comment in todo_comments.get(file):
            print("\t" + comment)
            i1 += 1

to_review = []
for file in review_dates:
    if not review_dates.get(file):  # skip if no date is provided
        to_review.append([file, 999999999])
        continue
    days_elapsed = abs(datetime.datetime.now() - review_dates.get(file)).days
    if days_elapsed > 30:
        to_review.append([file, days_elapsed])

print("\n")

to_review = sorted(to_review, key=lambda x: (x[1], x[0]))
for file in to_review:
    if file[1] == 999999999:
        print(file[0], ": ", "never", sep="")
    else:
        print(file[0], ": ", file[1], " days", sep="")
    i2 += 1

print("\n")

print("number of todo comments:", i1)
print("number of files needing review:", i2)

```

```
24953 [INFO] 24953 Byte read from stdin: '
24953 [INFO] 24953 Byte read from stdin: s
24953 [INFO] 24953 Byte read from stdin: o
24953 [INFO] 24953 Byte read from stdin: u
24953 [INFO] 24953 Byte read from stdin: t
24953 [INFO] 24953 Byte read from stdin: -
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin: .
24953 [INFO] 24953 Byte read from stdin: 8
24953 [INFO] 24953 Byte read from stdin: 6
24953 [INFO] 24953 Byte read from stdin: 4
24953 [INFO] 24953 Byte read from stdin: 5
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin: 
24953 [INFO] 24953 Byte read from stdin: 3
24953 [INFO] 24953 Byte read from stdin: .



  _____  _____   ____   _____
 |  __ \|  __ \ / __ \ / ____|        Powered by PROS for VEX V5
 | |__) | |__) | |  | | (___     Version:                  3.3.1
 |  ___/|  _  /| |  | |\___ \    Uptime:                   24.974 s
 | |    | | \ \| |__| |____) |   Compiled:                 Unknown
 |_|    |_|  \_\\____/|_____/    Directory:                Unknown


27766 [INFO], 2, motor added at 0x38e63d8
27766 [INFO], 2, motor added at 0x38e6370
27766 [INFO], 2, motor added at 0x38e6308
27766 [INFO], 2, motor added at 0x38e66a8
27766 [INFO], 2, motor added at 0x38e6440
27766 [INFO], 2, motor added at 0x38e64a8
27766 [INFO], 2, motor added at 0x38e6578
27766 [INFO], 2, motor added at 0x38e6510
27766 [INFO] CHASSIS_PID_TURN, Time: 2118, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 0.000000, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.000000, Absolute Angle: -0.000144, error history: 1, history size: 20, time out time: -2147481531, error difference: 0.000000, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27766 [INFO] CHASSIS_PID_TURN, Time: 2128, Actual_Vol1: -6.000000, Actual_Vol2: -6.000000, Actual_Vol3: -12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 900.011435, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.001143, Absolute Angle: -0.000164, error history: 2, history size: 20, time out time: -2147481531, error difference: 0.001143, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27766 [INFO] CHASSIS_PID_TURN, Time: 2138, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1800.035220, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.002378, Absolute Angle: -0.000186, error history: 3, history size: 20, time out time: -2147481531, error difference: 0.002378, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2148, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: -12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2700.071856, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.003664, Absolute Angle: -0.000208, error history: 4, history size: 20, time out time: -2147481531, error difference: 0.003664, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2158, Actual_Vol1: 0.000000, Actual_Vol2: 6.000000, Actual_Vol3: 74.000000, Actual_Vol4: -80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3600.122100, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.005024, Absolute Angle: -0.000232, error history: 5, history size: 20, time out time: -2147481531, error difference: 0.005024, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27768 [INFO] CHASSIS_PID_TURN, Time: 2168, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 277.000000, Actual_Vol4: -277.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4500.187036, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.006494, Absolute Angle: -0.000258, error history: 6, history size: 20, time out time: -2147481531, error difference: 0.006494, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2178, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 320.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5400.267413, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.008038, Absolute Angle: -0.000285, error history: 7, history size: 20, time out time: -2147481531, error difference: 0.008038, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2188, Actual_Vol1: 265.000000, Actual_Vol2: -259.000000, Actual_Vol3: 653.000000, Actual_Vol4: -659.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6300.364161, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.009675, Absolute Angle: -0.000313, error history: 8, history size: 20, time out time: -2147481531, error difference: 0.009675, over slew: 1, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 8.200000, Actual_Vel4: 0.000000
27770 [INFO] CHASSIS_PID_TURN, Time: 2198, Actual_Vol1: 499.000000, Actual_Vol2: -499.000000, Actual_Vol3: 770.000000, Actual_Vol4: -912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7200.477785, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.011362, Absolute Angle: -0.000343, error history: 9, history size: 20, time out time: -2147481531, error difference: 0.011362, over slew: 1, Actual_Vel1: 20.800000, Actual_Vel2: -0.000000, Actual_Vel3: 12.200000, Actual_Vel4: -12.200000
27772 [INFO] CHASSIS_PID_TURN, Time: 2208, Actual_Vol1: 795.000000, Actual_Vol2: -869.000000, Actual_Vol3: 1195.000000, Actual_Vol4: -1220.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8100.608902, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.013112, Absolute Angle: -0.000373, error history: 10, history size: 20, time out time: -2147481531, error difference: 0.013112, over slew: 1, Actual_Vel1: 20.800000, Actual_Vel2: -0.000000, Actual_Vel3: 17.400000, Actual_Vel4: -18.400000
27772 [INFO] CHASSIS_PID_TURN, Time: 2218, Actual_Vol1: 1318.000000, Actual_Vol2: -1392.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -1614.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9000.758792, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.014989, Absolute Angle: -0.000406, error history: 11, history size: 20, time out time: -2147481531, error difference: 0.014989, over slew: 1, Actual_Vel1: 20.000000, Actual_Vel2: -7.200000, Actual_Vel3: 36.200000, Actual_Vel4: -31.400000
27772 [INFO] CHASSIS_PID_TURN, Time: 2228, Actual_Vol1: 1836.000000, Actual_Vol2: -1829.000000, Actual_Vol3: 1940.000000, Actual_Vol4: -2088.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9900.927480, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.016869, Absolute Angle: -0.000439, error history: 12, history size: 20, time out time: -2147481531, error difference: 0.016869, over slew: 1, Actual_Vel1: 55.200000, Actual_Vel2: -19.800000, Actual_Vel3: 70.000000, Actual_Vel4: -56.000000
27774 [INFO] CHASSIS_PID_TURN, Time: 2238, Actual_Vol1: 2285.000000, Actual_Vol2: -2353.000000, Actual_Vol3: 2248.000000, Actual_Vol4: -2384.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10801.111978, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.018450, Absolute Angle: -0.000466, error history: 13, history size: 20, time out time: -2147481531, error difference: 0.018450, over slew: 1, Actual_Vel1: 91.200000, Actual_Vel2: -51.600000, Actual_Vel3: 88.000000, Actual_Vel4: -90.200000
27774 [INFO] CHASSIS_PID_TURN, Time: 2248, Actual_Vol1: 2643.000000, Actual_Vol2: -2920.000000, Actual_Vol3: 2513.000000, Actual_Vol4: -2766.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11701.312031, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.020005, Absolute Angle: -0.000493, error history: 14, history size: 20, time out time: -2147481531, error difference: 0.020005, over slew: 1, Actual_Vel1: 105.800000, Actual_Vel2: -81.400000, Actual_Vel3: 101.000000, Actual_Vel4: -106.800000
27774 [INFO] CHASSIS_PID_TURN, Time: 2258, Actual_Vol1: 2914.000000, Actual_Vol2: -3333.000000, Actual_Vol3: 2834.000000, Actual_Vol4: -3006.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12601.525177, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.021315, Absolute Angle: -0.000516, error history: 15, history size: 20, time out time: -2147481531, error difference: 0.021315, over slew: 1, Actual_Vel1: 131.800000, Actual_Vel2: -105.400000, Actual_Vel3: 74.200000, Actual_Vel4: -131.600000
27776 [INFO] CHASSIS_PID_TURN, Time: 2268, Actual_Vol1: 3049.000000, Actual_Vol2: -3653.000000, Actual_Vol3: 3616.000000, Actual_Vol4: -3160.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13501.632044, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: -0.010687, Absolute Angle: -0.000331, error history: 16, history size: 20, time out time: -2147481531, error difference: 0.021315, over slew: 1, Actual_Vel1: 126.800000, Actual_Vel2: -96.800000, Actual_Vel3: 35.600000, Actual_Vel4: -117.400000
27776 [INFO] CHASSIS_PID_TURN, Time: 2278, Actual_Vol1: 3129.000000, Actual_Vol2: -3881.000000, Actual_Vol3: 4306.000000, Actual_Vol4: -3345.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14401.451417, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.018063, Absolute Angle: 0.000171, error history: 17, history size: 20, time out time: -2147481531, error difference: 0.039377, over slew: 1, Actual_Vel1: 103.600000, Actual_Vel2: -87.200000, Actual_Vel3: 25.600000, Actual_Vel4: -83.600000
27776 [INFO] CHASSIS_PID_TURN, Time: 2288, Actual_Vol1: 3511.000000, Actual_Vol2: -4503.000000, Actual_Vol3: 4990.000000, Actual_Vol4: -3665.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15300.670815, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.078060, Absolute Angle: 0.001218, error history: 18, history size: 20, time out time: -2147481531, error difference: 0.099375, over slew: 1, Actual_Vel1: 68.800000, Actual_Vel2: -33.600000, Actual_Vel3: 33.400000, Actual_Vel4: -70.800000
27778 [INFO] CHASSIS_PID_TURN, Time: 2298, Actual_Vol1: 4121.000000, Actual_Vol2: -4977.000000, Actual_Vol3: 5624.000000, Actual_Vol4: -4059.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16198.571932, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.209888, Absolute Angle: 0.003519, error history: 19, history size: 20, time out time: -2147481531, error difference: 0.231203, over slew: 1, Actual_Vel1: 28.600000, Actual_Vel2: -33.600000, Actual_Vel3: 57.200000, Actual_Vel4: -79.600000
27778 [INFO] CHASSIS_PID_TURN, Time: 2308, Actual_Vol1: 4916.000000, Actual_Vol2: -5735.000000, Actual_Vol3: 6222.000000, Actual_Vol4: -4491.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17094.890460, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.368147, Absolute Angle: 0.006281, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.389462, over slew: 1, Actual_Vel1: 65.000000, Actual_Vel2: -23.400000, Actual_Vel3: 64.200000, Actual_Vel4: -90.200000
27778 [INFO] CHASSIS_PID_TURN, Time: 2318, Actual_Vol1: 5433.000000, Actual_Vol2: -6462.000000, Actual_Vol3: 6733.000000, Actual_Vol4: -4792.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17989.316281, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 0.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.557418, Absolute Angle: 0.009585, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578732, over slew: 1, Actual_Vel1: 79.800000, Actual_Vel2: -52.400000, Actual_Vel3: 78.400000, Actual_Vel4: -111.000000
27780 [INFO] CHASSIS_PID_TURN, Time: 2328, Actual_Vol1: 5883.000000, Actual_Vol2: -6991.000000, Actual_Vol3: 7164.000000, Actual_Vol4: -5094.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18881.531910, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 1.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 0.778437, Absolute Angle: 0.013442, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.799752, over slew: 1, Actual_Vel1: 69.200000, Actual_Vel2: -57.200000, Actual_Vel3: 101.600000, Actual_Vel4: -138.600000
27780 [INFO] CHASSIS_PID_TURN, Time: 2338, Actual_Vol1: 6437.000000, Actual_Vol2: -7626.000000, Actual_Vol3: 7515.000000, Actual_Vol4: -5291.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19769.960788, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 2.000000, position_r: 0.000000, Heading_Sp: 90.000000, Relative_Heading: 1.157112, Absolute Angle: 0.020051, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.178427, over slew: 1, Actual_Vel1: 72.200000, Actual_Vel2: -56.800000, Actual_Vel3: 111.600000, Actual_Vel4: -162.200000
27780 [INFO] CHASSIS_PID_TURN, Time: 2348, Actual_Vol1: 6942.000000, Actual_Vol2: -8020.000000, Actual_Vol3: 7872.000000, Actual_Vol4: -5390.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20653.967137, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 4.000000, position_r: -1.000000, Heading_Sp: 90.000000, Relative_Heading: 1.599365, Absolute Angle: 0.027770, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.620680, over slew: 1, Actual_Vel1: 100.600000, Actual_Vel2: -62.200000, Actual_Vel3: 120.800000, Actual_Vel4: -187.400000
27782 [INFO] CHASSIS_PID_TURN, Time: 2358, Actual_Vol1: 7330.000000, Actual_Vol2: -6376.000000, Actual_Vol3: 7928.000000, Actual_Vol4: -5371.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21531.034134, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 5.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 2.293300, Absolute Angle: 0.039881, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.314615, over slew: 1, Actual_Vel1: 114.000000, Actual_Vel2: -63.600000, Actual_Vel3: 144.400000, Actual_Vel4: -193.400000
27782 [INFO] CHASSIS_PID_TURN, Time: 2368, Actual_Vol1: 7669.000000, Actual_Vol2: -6252.000000, Actual_Vol3: 7706.000000, Actual_Vol4: -5125.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22401.649649, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 8.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 2.938449, Absolute Angle: 0.051141, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.959763, over slew: 1, Actual_Vel1: 135.000000, Actual_Vel2: -46.400000, Actual_Vel3: 146.200000, Actual_Vel4: -180.200000
```

```
27782 [INFO] CHASSIS_PID_TURN, Time: 2378, Actual_Vol1: 7774.000000, Actual_Vol2: -6733.000000, Actual_Vol3: 7632.000000, Actual_Vol4: -4971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23264.271918, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 10.000000, position_r: -2.000000, Heading_Sp: 90.000000, Relative_Heading: 3.737773, Absolute Angle: 0.065092, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.759088, o
ver slew: 1, Actual_Vel1: 145.800000, Actual_Vel2: -58.000000, Actual_Vel3: 140.200000, Actual_Vel4: -173.200000
27784 [INFO] CHASSIS_PID_TURN, Time: 2388, Actual_Vol1: 7903.000000, Actual_Vol2: -6930.000000, Actual_Vol3: 7798.000000, Actual_Vol4: -4601.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24119.287181, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 13.000000, position_r: -3.000000, Heading_Sp: 90.000000, Relative_Heading: 4.498474, Absolute Angle: 0.078369, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.519788, o
ver slew: 1, Actual_Vel1: 168.200000, Actual_Vel2: -84.200000, Actual_Vel3: 159.400000, Actual_Vel4: -144.200000
27784 [INFO] CHASSIS_PID_TURN, Time: 2398, Actual_Vol1: 7835.000000, Actual_Vol2: -7503.000000, Actual_Vol3: 7798.000000, Actual_Vol4: -4571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24964.056953, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 16.000000, position_r: -4.000000, Heading_Sp: 90.000000, Relative_Heading: 5.523023, Absolute Angle: 0.096251, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.544337, o
ver slew: 1, Actual_Vel1: 168.800000, Actual_Vel2: -94.800000, Actual_Vel3: 154.200000, Actual_Vel4: -140.000000
27784 [INFO] CHASSIS_PID_TURN, Time: 2408, Actual_Vol1: 7983.000000, Actual_Vol2: -7232.000000, Actual_Vol3: 8039.000000, Actual_Vol4: -4564.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25797.979380, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 19.000000, position_r: -5.000000, Heading_Sp: 90.000000, Relative_Heading: 6.607757, Absolute Angle: 0.115183, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.629072, o
ver slew: 1, Actual_Vel1: 183.800000, Actual_Vel2: -99.600000, Actual_Vel3: 180.200000, Actual_Vel4: -139.200000
27786 [INFO] CHASSIS_PID_TURN, Time: 2418, Actual_Vol1: 7694.000000, Actual_Vol2: -7515.000000, Actual_Vol3: 8193.000000, Actual_Vol4: -4817.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26620.547837, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 22.000000, position_r: -6.000000, Heading_Sp: 90.000000, Relative_Heading: 7.743154, Absolute Angle: 0.134999, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.764469, o
ver slew: 1, Actual_Vel1: 183.600000, Actual_Vel2: -97.800000, Actual_Vel3: 201.400000, Actual_Vel4: -155.600000
27786 [INFO] CHASSIS_PID_TURN, Time: 2428, Actual_Vol1: 7503.000000, Actual_Vol2: -7373.000000, Actual_Vol3: 8082.000000, Actual_Vol4: -4854.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27431.227228, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 25.000000, position_r: -6.000000, Heading_Sp: 90.000000, Relative_Heading: 8.932061, Absolute Angle: 0.155750, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.953376, o
ver slew: 1, Actual_Vel1: 187.000000, Actual_Vel2: -110.600000, Actual_Vel3: 203.400000, Actual_Vel4: -168.800000
27786 [INFO] CHASSIS_PID_TURN, Time: 2438, Actual_Vol1: 7226.000000, Actual_Vol2: -7614.000000, Actual_Vol3: 7706.000000, Actual_Vol4: -4805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28230.461524, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 29.000000, position_r: -8.000000, Heading_Sp: 90.000000, Relative_Heading: 10.076570, Absolute Angle: 0.175725, error history: 20, history size: 20, time out time: -2147481531, error difference: 10.097885,
over slew: 1, Actual_Vel1: 197.000000, Actual_Vel2: -124.200000, Actual_Vel3: 195.000000, Actual_Vel4: -174.000000
27788 [INFO] CHASSIS_PID_TURN, Time: 2448, Actual_Vol1: 7059.000000, Actual_Vol2: -7595.000000, Actual_Vol3: 7343.000000, Actual_Vol4: -4583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29014.765327, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 33.000000, position_r: -8.000000, Heading_Sp: 90.000000, Relative_Heading: 11.569620, Absolute Angle: 0.201784, error history: 20, history size: 20, time out time: -2147481531, error difference: 11.590934,
over slew: 1, Actual_Vel1: 192.400000, Actual_Vel2: -123.600000, Actual_Vel3: 188.000000, Actual_Vel4: -180.200000
27788 [INFO] CHASSIS_PID_TURN, Time: 2458, Actual_Vol1: 6733.000000, Actual_Vol2: -7922.000000, Actual_Vol3: 7102.000000, Actual_Vol4: -4410.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29785.610472, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 37.000000, position_r: -9.000000, Heading_Sp: 90.000000, Relative_Heading: 12.915486, Absolute Angle: 0.225274, error history: 20, history size: 20, time out time: -2147481531, error difference: 12.926172,
over slew: 1, Actual_Vel1: 188.600000, Actual_Vel2: -124.600000, Actual_Vel3: 199.400000, Actual_Vel4: -183.000000
27788 [INFO] CHASSIS_PID_TURN, Time: 2468, Actual_Vol1: 6548.000000, Actual_Vol2: -7657.000000, Actual_Vol3: 6856.000000, Actual_Vol4: -4300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30541.645449, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 42.000000, position_r: -10.000000, Heading_Sp: 90.000000, Relative_Heading: 14.396502, Absolute Angle: 0.251122, error history: 20, history size: 20, time out time: -2147481531, error difference: 14.378440
, over slew: 1, Actual_Vel1: 185.800000, Actual_Vel2: -116.200000, Actual_Vel3: 221.800000, Actual_Vel4: -190.400000
27790 [INFO] CHASSIS_PID_TURN, Time: 2478, Actual_Vol1: 6271.000000, Actual_Vol2: -7614.000000, Actual_Vol3: 6462.000000, Actual_Vol4: -4090.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31281.596530, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 46.000000, position_r: -10.000000, Heading_Sp: 90.000000, Relative_Heading: 16.004892, Absolute Angle: 0.279194, error history: 20, history size: 20, time out time: -2147481531, error difference: 15.926832
, over slew: 1, Actual_Vel1: 188.600000, Actual_Vel2: -125.000000, Actual_Vel3: 206.200000, Actual_Vel4: -192.600000
27790 [INFO] CHASSIS_PID_TURN, Time: 2488, Actual_Vol1: 6043.000000, Actual_Vol2: -8002.000000, Actual_Vol3: 6098.000000, Actual_Vol4: -3838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32007.198633, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 50.000000, position_r: -12.000000, Heading_Sp: 90.000000, Relative_Heading: 17.439790, Absolute Angle: 0.304238, error history: 20, history size: 20, time out time: -2147481531, error difference: 17.229901
, over slew: 1, Actual_Vel1: 206.800000, Actual_Vel2: -135.000000, Actual_Vel3: 190.200000, Actual_Vel4: -195.600000
27790 [INFO] CHASSIS_PID_TURN, Time: 2498, Actual_Vol1: 5772.000000, Actual_Vol2: -8051.000000, Actual_Vol3: 5914.000000, Actual_Vol4: -3647.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32716.206349, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 54.000000, position_r: -13.000000, Heading_Sp: 90.000000, Relative_Heading: 19.099228, Absolute Angle: 0.333200, error history: 20, history size: 20, time out time: -2147481531, error difference: 18.731081
, over slew: 1, Actual_Vel1: 210.000000, Actual_Vel2: -147.200000, Actual_Vel3: 194.000000, Actual_Vel4: -182.400000
27792 [INFO] CHASSIS_PID_TURN, Time: 2508, Actual_Vol1: 5562.000000, Actual_Vol2: -7971.000000, Actual_Vol3: 5673.000000, Actual_Vol4: -3431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33409.465182, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 59.000000, position_r: -14.000000, Heading_Sp: 90.000000, Relative_Heading: 20.674117, Absolute Angle: 0.360687, error history: 20, history size: 20, time out time: -2147481531, error difference: 20.116699
, over slew: 1, Actual_Vel1: 189.400000, Actual_Vel2: -136.000000, Actual_Vel3: 210.000000, Actual_Vel4: -179.600000
27792 [INFO] CHASSIS_PID_TURN, Time: 2518, Actual_Vol1: 5217.000000, Actual_Vol2: -8242.000000, Actual_Vol3: 5365.000000, Actual_Vol4: -3259.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34085.853712, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 63.000000, position_r: -16.000000, Heading_Sp: 90.000000, Relative_Heading: 22.361147, Absolute Angle: 0.390131, error history: 20, history size: 20, time out time: -2147481531, error difference: 21.582710
, over slew: 1, Actual_Vel1: 179.200000, Actual_Vel2: -153.000000, Actual_Vel3: 216.600000, Actual_Vel4: -178.800000
27792 [INFO] CHASSIS_PID_TURN, Time: 2528, Actual_Vol1: 5039.000000, Actual_Vol2: -8267.000000, Actual_Vol3: 5026.000000, Actual_Vol4: -3018.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34745.199622, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 67.000000, position_r: -17.000000, Heading_Sp: 90.000000, Relative_Heading: 24.065409, Absolute Angle: 0.419876, error history: 20, history size: 20, time out time: -2147481531, error difference: 22.908297
, over slew: 1, Actual_Vel1: 183.200000, Actual_Vel2: -142.600000, Actual_Vel3: 201.600000, Actual_Vel4: -172.600000
27794 [INFO] CHASSIS_PID_TURN, Time: 2538, Actual_Vol1: 4829.000000, Actual_Vol2: -8193.000000, Actual_Vol3: 4682.000000, Actual_Vol4: -2852.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35388.418998, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 71.000000, position_r: -18.000000, Heading_Sp: 90.000000, Relative_Heading: 25.678062, Absolute Angle: 0.448022, error history: 20, history size: 20, time out time: -2147481531, error difference: 24.078697
, over slew: 1, Actual_Vel1: 189.600000, Actual_Vel2: -138.400000, Actual_Vel3: 184.600000, Actual_Vel4: -169.400000
27794 [INFO] CHASSIS_PID_TURN, Time: 2548, Actual_Vol1: 4614.000000, Actual_Vol2: -8013.000000, Actual_Vol3: 4392.000000, Actual_Vol4: -2772.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36015.456502, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 76.000000, position_r: -20.000000, Heading_Sp: 90.000000, Relative_Heading: 27.296250, Absolute Angle: 0.476265, error history: 20, history size: 20, time out time: -2147481531, error difference: 25.002949
, over slew: 1, Actual_Vel1: 195.600000, Actual_Vel2: -143.000000, Actual_Vel3: 177.400000, Actual_Vel4: -162.000000
27794 [INFO] CHASSIS_PID_TURN, Time: 2558, Actual_Vol1: 4226.000000, Actual_Vol2: -8217.000000, Actual_Vol3: 4183.000000, Actual_Vol4: -2729.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36624.418762, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 79.000000, position_r: -22.000000, Heading_Sp: 90.000000, Relative_Heading: 29.103774, Absolute Angle: 0.507812, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.165325
, over slew: 0, Actual_Vel1: 175.400000, Actual_Vel2: -153.200000, Actual_Vel3: 182.000000, Actual_Vel4: -162.200000
27796 [INFO] CHASSIS_PID_TURN, Time: 2568, Actual_Vol1: 4084.000000, Actual_Vol2: -8402.000000, Actual_Vol3: 4010.000000, Actual_Vol4: -2667.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37217.332056, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 83.000000, position_r: -24.000000, Heading_Sp: 90.000000, Relative_Heading: 30.708671, Absolute Angle: 0.535823, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.970898
, over slew: 0, Actual_Vel1: 169.200000, Actual_Vel2: -164.000000, Actual_Vel3: 186.000000, Actual_Vel4: -155.800000
27796 [INFO] CHASSIS_PID_TURN, Time: 2578, Actual_Vol1: 3850.000000, Actual_Vol2: -8279.000000, Actual_Vol3: 3819.000000, Actual_Vol4: -2661.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37793.115962, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 87.000000, position_r: -26.000000, Heading_Sp: 90.000000, Relative_Heading: 32.421609, Absolute Angle: 0.565720, error history: 20, history size: 20, time out time: -2147481531, error difference: 27.923136
, over slew: 0, Actual_Vel1: 171.400000, Actual_Vel2: -162.200000, Actual_Vel3: 182.400000, Actual_Vel4: -153.800000
27796 [INFO] CHASSIS_PID_TURN, Time: 2588, Actual_Vol1: 3665.000000, Actual_Vol2: -8273.000000, Actual_Vol3: 3671.000000, Actual_Vol4: -2624.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38351.699728, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 89.000000, position_r: -28.000000, Heading_Sp: 90.000000, Relative_Heading: 34.141623, Absolute Angle: 0.595739, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.618601
, over slew: 0, Actual_Vel1: 170.600000, Actual_Vel2: -174.400000, Actual_Vel3: 167.400000, Actual_Vel4: -150.000000
27798 [INFO] CHASSIS_PID_TURN, Time: 2598, Actual_Vol1: 3591.000000, Actual_Vol2: -8642.000000, Actual_Vol3: 3542.000000, Actual_Vol4: -2581.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38895.055902, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 92.000000, position_r: -32.000000, Heading_Sp: 90.000000, Relative_Heading: 35.664383, Absolute Angle: 0.622317, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.056625
, over slew: 0, Actual_Vel1: 173.600000, Actual_Vel2: -174.400000, Actual_Vel3: 163.000000, Actual_Vel4: -147.800000
27798 [INFO] CHASSIS_PID_TURN, Time: 2608, Actual_Vol1: 3437.000000, Actual_Vol2: -8667.000000, Actual_Vol3: 3456.000000, Actual_Vol4: -2563.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39420.325254, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 96.000000, position_r: -34.000000, Heading_Sp: 90.000000, Relative_Heading: 37.473065, Absolute Angle: 0.653884, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.729910
, over slew: 0, Actual_Vel1: 158.600000, Actual_Vel2: -172.400000, Actual_Vel3: 157.400000, Actual_Vel4: -144.800000
27798 [INFO] CHASSIS_PID_TURN, Time: 2618, Actual_Vol1: 3351.000000, Actual_Vol2: -8285.000000, Actual_Vol3: 3400.000000, Actual_Vol4: -2593.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39928.568954, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 99.000000, position_r: -37.000000, Heading_Sp: 90.000000, Relative_Heading: 39.175630, Absolute Angle: 0.683599, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.243569
, over slew: 0, Actual_Vel1: 143.600000, Actual_Vel2: -167.200000, Actual_Vel3: 161.800000, Actual_Vel4: -145.200000
27800 [INFO] CHASSIS_PID_TURN, Time: 2628, Actual_Vol1: 3302.000000, Actual_Vol2: -8341.000000, Actual_Vol3: 3308.000000, Actual_Vol4: -2630.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40419.826318, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 102.000000, position_r: -40.000000, Heading_Sp: 90.000000, Relative_Heading: 40.874264, Absolute Angle: 0.713246, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.79769
3, over slew: 0, Actual_Vel1: 140.800000, Actual_Vel2: -172.000000, Actual_Vel3: 158.600000, Actual_Vel4: -143.800000
27800 [INFO] CHASSIS_PID_TURN, Time: 2638, Actual_Vol1: 3382.000000, Actual_Vol2: -8846.000000, Actual_Vol3: 3179.000000, Actual_Vol4: -2587.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40894.107657, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 104.000000, position_r: -44.000000, Heading_Sp: 90.000000, Relative_Heading: 42.571866, Absolute Angle: 0.742875, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.00224
6, over slew: 0, Actual_Vel1: 148.400000, Actual_Vel2: -197.800000, Actual_Vel3: 150.600000, Actual_Vel4: -145.600000
27800 [INFO] CHASSIS_PID_TURN, Time: 2648, Actual_Vol1: 3376.000000, Actual_Vol2: -8778.000000, Actual_Vol3: 3043.000000, Actual_Vol4: -2519.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41351.548194, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 107.000000, position_r: -46.000000, Heading_Sp: 90.000000, Relative_Heading: 44.255946, Absolute Angle: 0.772268, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.34046
1, over slew: 0, Actual_Vel1: 151.200000, Actual_Vel2: -186.800000, Actual_Vel3: 142.200000, Actual_Vel4: -142.600000
27802 [INFO] CHASSIS_PID_TURN, Time: 2658, Actual_Vol1: 3203.000000, Actual_Vol2: -8575.000000, Actual_Vol3: 3012.000000, Actual_Vol4: -2482.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41793.148995, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 109.000000, position_r: -50.000000, Heading_Sp: 90.000000, Relative_Heading: 45.839920, Absolute Angle: 0.799913, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.44341
8, over slew: 0, Actual_Vel1: 145.000000, Actual_Vel2: -167.400000, Actual_Vel3: 143.800000, Actual_Vel4: -144.800000
27802 [INFO] CHASSIS_PID_TURN, Time: 2668, Actual_Vol1: 3000.000000, Actual_Vol2: -8753.000000, Actual_Vol3: 3006.000000, Actual_Vol4: -2458.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42217.863240, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 111.000000, position_r: -54.000000, Heading_Sp: 90.000000, Relative_Heading: 47.528576, Absolute Angle: 0.829386, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.52368
4, over slew: 0, Actual_Vel1: 132.600000, Actual_Vel2: -185.400000, Actual_Vel3: 143.200000, Actual_Vel4: -137.600000
27802 [INFO] CHASSIS_PID_TURN, Time: 2678, Actual_Vol1: 2981.000000, Actual_Vol2: -8310.000000, Actual_Vol3: 2963.000000, Actual_Vol4: -2433.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42625.742256, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 113.000000, position_r: -58.000000, Heading_Sp: 90.000000, Relative_Heading: 49.212098, Absolute Angle: 0.858769, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.77230
9, over slew: 0, Actual_Vel1: 132.000000, Actual_Vel2: -179.200000, Actual_Vel3: 141.000000, Actual_Vel4: -139.400000
27804 [INFO] CHASSIS_PID_TURN, Time: 2688, Actual_Vol1: 2926.000000, Actual_Vol2: -8630.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -2291.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43016.881858, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 114.000000, position_r: -61.000000, Heading_Sp: 90.000000, Relative_Heading: 50.886040, Absolute Angle: 0.887985, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.78681
1, over slew: 0, Actual_Vel1: 126.000000, Actual_Vel2: -201.600000, Actual_Vel3: 134.200000, Actual_Vel4: -133.800000
27804 [INFO] CHASSIS_PID_TURN, Time: 2698, Actual_Vol1: 2907.000000, Actual_Vol2: -8772.000000, Actual_Vol3: 2797.000000, Actual_Vol4: -2211.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43393.273086, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 116.000000, position_r: -65.000000, Heading_Sp: 90.000000, Relative_Heading: 52.360877, Absolute Angle: 0.913725, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.68676
1, over slew: 0, Actual_Vel1: 133.000000, Actual_Vel2: -208.800000, Actual_Vel3: 125.200000, Actual_Vel4: -129.600000
27804 [INFO] CHASSIS_PID_TURN, Time: 2708, Actual_Vol1: 2846.000000, Actual_Vol2: -8612.000000, Actual_Vol3: 2643.000000, Actual_Vol4: -2174.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43753.086264, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 116.000000, position_r: -70.000000, Heading_Sp: 90.000000, Relative_Heading: 54.018682, Absolute Angle: 0.942660, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.65753
5, over slew: 0, Actual_Vel1: 124.200000, Actual_Vel2: -192.400000, Actual_Vel3: 115.400000, Actual_Vel4: -123.400000
27806 [INFO] CHASSIS_PID_TURN, Time: 2718, Actual_Vol1: 2821.000000, Actual_Vol2: -8297.000000, Actual_Vol3: 2729.000000, Actual_Vol4: -2113.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44097.249899, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -77.000000, Heading_Sp: 90.000000, Relative_Heading: 55.583636, Absolute Angle: 0.969973, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.51822
7, over slew: 0, Actual_Vel1: 115.000000, Actual_Vel2: -189.400000, Actual_Vel3: 110.800000, Actual_Vel4: -125.400000
27806 [INFO] CHASSIS_PID_TURN, Time: 2728, Actual_Vol1: 2735.000000, Actual_Vol2: -7885.000000, Actual_Vol3: 2618.000000, Actual_Vol4: -1996.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44421.920395, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -82.000000, Heading_Sp: 90.000000, Relative_Heading: 57.532950, Absolute Angle: 1.003995, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.85488
8, over slew: 0, Actual_Vel1: 104.800000, Actual_Vel2: -198.400000, Actual_Vel3: 113.200000, Actual_Vel4: -119.000000
27806 [INFO] CHASSIS_PID_TURN, Time: 2738, Actual_Vol1: 2661.000000, Actual_Vol2: -7361.000000, Actual_Vol3: 2452.000000, Actual_Vol4: -1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44731.175417, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 118.000000, position_r: -87.000000, Heading_Sp: 90.000000, Relative_Heading: 59.074498, Absolute Angle: 1.030900, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.77824
8, over slew: 0, Actual_Vel1: 97.000000, Actual_Vel2: -201.400000, Actual_Vel3: 105.800000, Actual_Vel4: -111.000000
27808 [INFO] CHASSIS_PID_TURN, Time: 2748, Actual_Vol1: 2741.000000, Actual_Vol2: -6850.000000, Actual_Vol3: 2396.000000, Actual_Vol4: -1786.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45025.089547, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 119.000000, position_r: -91.000000, Heading_Sp: 90.000000, Relative_Heading: 60.608587, Absolute Angle: 1.057675, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.50481
```

```
3, over slew: 0, Actual_Vel1: 99.200000, Actual_Vel2: -173.400000, Actual_Vel3: 104.600000, Actual_Vel4: -108.000000
    100:   27808 [INFO] CHASSIS_PID_TURN, Time: 2758, Actual_Vol1: 2723.000000, Actual_Vol2: -6326.000000, Actual_Vol3: 2378.000000, Actual_Vol4: -1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45303.874626, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -95.000000, Heading_Sp: 90.000000, Relative_Heading: 62.121492, Absolute Angle: 1.084080, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.41282
1, over slew: 0, Actual_Vel1: 98.800000, Actual_Vel2: -173.000000, Actual_Vel3: 106.800000, Actual_Vel4: -100.600000
    101:   27808 [INFO] CHASSIS_PID_TURN, Time: 2768, Actual_Vol1: 2575.000000, Actual_Vol2: -5852.000000, Actual_Vol3: 2279.000000, Actual_Vol4: -1608.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45567.733878, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -101.000000, Heading_Sp: 90.000000, Relative_Heading: 63.614075, Absolute Angle: 1.110131, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.1924
66, over slew: 0, Actual_Vel1: 95.200000, Actual_Vel2: -164.800000, Actual_Vel3: 108.400000, Actual_Vel4: -94.000000
    102:   27810 [INFO] CHASSIS_PID_TURN, Time: 2778, Actual_Vol1: 2556.000000, Actual_Vol2: -5359.000000, Actual_Vol3: 2187.000000, Actual_Vol4: -1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45815.884714, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 120.000000, position_r: -106.000000, Heading_Sp: 90.000000, Relative_Heading: 65.184916, Absolute Angle: 1.137547, error history: 20, history size: 20, time out time: -2147481531, error difference: 31.0432
93, over slew: 0, Actual_Vel1: 98.400000, Actual_Vel2: -165.600000, Actual_Vel3: 102.200000, Actual_Vel4: -92.000000
    103:   27810 [INFO] CHASSIS_PID_TURN, Time: 2788, Actual_Vol1: 2470.000000, Actual_Vol2: -4885.000000, Actual_Vol3: 2045.000000, Actual_Vol4: -1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46049.611903, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 121.000000, position_r: -110.000000, Heading_Sp: 90.000000, Relative_Heading: 66.627281, Absolute Angle: 1.162721, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.9628
98, over slew: 0, Actual_Vel1: 97.200000, Actual_Vel2: -158.000000, Actual_Vel3: 91.800000, Actual_Vel4: -98.000000
    104:   27810 [INFO] CHASSIS_PID_TURN, Time: 2798, Actual_Vol1: 2365.000000, Actual_Vol2: -4386.000000, Actual_Vol3: 1885.000000, Actual_Vol4: -1417.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46269.140671, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 121.000000, position_r: -114.000000, Heading_Sp: 90.000000, Relative_Heading: 68.047123, Absolute Angle: 1.187502, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.5740
58, over slew: 0, Actual_Vel1: 91.000000, Actual_Vel2: -158.600000, Actual_Vel3: 83.000000, Actual_Vel4: -85.800000
    105:   27812 [INFO] CHASSIS_PID_TURN, Time: 2808, Actual_Vol1: 2187.000000, Actual_Vol2: -3905.000000, Actual_Vol3: 1780.000000, Actual_Vol4: -1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46475.692715, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -117.000000, Heading_Sp: 90.000000, Relative_Heading: 69.344796, Absolute Angle: 1.210151, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.1691
66, over slew: 0, Actual_Vel1: 82.800000, Actual_Vel2: -144.400000, Actual_Vel3: 81.200000, Actual_Vel4: -95.000000
    106:   27812 [INFO] CHASSIS_PID_TURN, Time: 2818, Actual_Vol1: 2094.000000, Actual_Vol2: -3055.000000, Actual_Vol3: 1700.000000, Actual_Vol4: -1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46661.433960, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -121.000000, Heading_Sp: 90.000000, Relative_Heading: 71.425876, Absolute Angle: 1.246472, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.5516
12, over slew: 0, Actual_Vel1: 73.200000, Actual_Vel2: -117.800000, Actual_Vel3: 77.600000, Actual_Vel4: -91.000000
    107:   27812 [INFO] CHASSIS_PID_TURN, Time: 2828, Actual_Vol1: 2057.000000, Actual_Vol2: -2686.000000, Actual_Vol3: 1663.000000, Actual_Vol4: -979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46835.470646, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -125.000000, Heading_Sp: 90.000000, Relative_Heading: 72.596331, Absolute Angle: 1.266901, error history: 20, history size: 20, time out time: -2147481531, error difference: 30.02446
5, over slew: 0, Actual_Vel1: 65.000000, Actual_Vel2: -107.800000, Actual_Vel3: 72.800000, Actual_Vel4: -82.000000
    108:   27814 [INFO] CHASSIS_PID_TURN, Time: 2838, Actual_Vol1: 1990.000000, Actual_Vol2: -2384.000000, Actual_Vol3: 1651.000000, Actual_Vol4: -825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46998.214738, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -129.000000, Heading_Sp: 90.000000, Relative_Heading: 73.725591, Absolute Angle: 1.286610, error history: 20, history size: 20, time out time: -2147481531, error difference: 29.46964
5, over slew: 0, Actual_Vel1: 63.200000, Actual_Vel2: -99.000000, Actual_Vel3: 67.800000, Actual_Vel4: -75.600000
    109:   27814 [INFO] CHASSIS_PID_TURN, Time: 2848, Actual_Vol1: 1854.000000, Actual_Vol2: -2057.000000, Actual_Vol3: 1639.000000, Actual_Vol4: -764.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47150.095294, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -132.000000, Heading_Sp: 90.000000, Relative_Heading: 74.811944, Absolute Angle: 1.305571, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.97202
4, over slew: 0, Actual_Vel1: 55.600000, Actual_Vel2: -87.600000, Actual_Vel3: 64.400000, Actual_Vel4: -70.800000
    110:   27814 [INFO] CHASSIS_PID_TURN, Time: 2858, Actual_Vol1: 2008.000000, Actual_Vol2: -1817.000000, Actual_Vol3: 1466.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47292.502181, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 122.000000, position_r: -135.000000, Heading_Sp: 90.000000, Relative_Heading: 75.759311, Absolute Angle: 1.322105, error history: 20, history size: 20, time out time: -2147481531, error difference: 28.23073
6, over slew: 0, Actual_Vel1: 51.000000, Actual_Vel2: -77.200000, Actual_Vel3: 63.000000, Actual_Vel4: -66.800000
    111:   27816 [INFO] CHASSIS_PID_TURN, Time: 2868, Actual_Vol1: 1971.000000, Actual_Vol2: -1725.000000, Actual_Vol3: 1435.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47425.872707, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 123.000000, position_r: -138.000000, Heading_Sp: 90.000000, Relative_Heading: 76.662947, Absolute Angle: 1.337877, error history: 20, history size: 20, time out time: -2147481531, error difference: 27.45084
9, over slew: 0, Actual_Vel1: 51.200000, Actual_Vel2: -70.600000, Actual_Vel3: 55.800000, Actual_Vel4: -61.600000
    112:   27816 [INFO] CHASSIS_PID_TURN, Time: 2878, Actual_Vol1: 1823.000000, Actual_Vol2: -1558.000000, Actual_Vol3: 1398.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47549.613356, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 123.000000, position_r: -141.000000, Heading_Sp: 90.000000, Relative_Heading: 77.625935, Absolute Angle: 1.354684, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.73989
5, over slew: 0, Actual_Vel1: 48.400000, Actual_Vel2: -65.400000, Actual_Vel3: 53.400000, Actual_Vel4: -57.400000
    113:   27816 [INFO] CHASSIS_PID_TURN, Time: 2888, Actual_Vol1: 2008.000000, Actual_Vol2: -1398.000000, Actual_Vol3: 1337.000000, Actual_Vol4: -437.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47665.214249, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 124.000000, position_r: -143.000000, Heading_Sp: 90.000000, Relative_Heading: 78.439911, Absolute Angle: 1.368890, error history: 20, history size: 20, time out time: -2147481531, error difference: 26.07903
3, over slew: 0, Actual_Vel1: 51.200000, Actual_Vel2: -59.600000, Actual_Vel3: 54.400000, Actual_Vel4: -54.600000
    114:   27818 [INFO] CHASSIS_PID_TURN, Time: 2898, Actual_Vol1: 1983.000000, Actual_Vol2: -1312.000000, Actual_Vol3: 1318.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47773.174973, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 124.000000, position_r: -146.000000, Heading_Sp: 90.000000, Relative_Heading: 79.203928, Absolute Angle: 1.382225, error history: 20, history size: 20, time out time: -2147481531, error difference: 25.18524
5, over slew: 0, Actual_Vel1: 55.200000, Actual_Vel2: -55.000000, Actual_Vel3: 52.400000, Actual_Vel4: -56.600000
    115:   27818 [INFO] CHASSIS_PID_TURN, Time: 2908, Actual_Vol1: 1756.000000, Actual_Vol2: -1220.000000, Actual_Vol3: 1238.000000, Actual_Vol4: -129.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47873.948486, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 125.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 79.922649, Absolute Angle: 1.394769, error history: 20, history size: 20, time out time: -2147481531, error difference: 24.33901
2, over slew: 0, Actual_Vel1: 53.000000, Actual_Vel2: -54.600000, Actual_Vel3: 53.000000, Actual_Vel4: -55.200000
    116:   27818 [INFO] CHASSIS_PID_TURN, Time: 2918, Actual_Vol1: 1657.000000, Actual_Vol2: -1103.000000, Actual_Vol3: 1084.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47968.036170, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 126.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 80.591232, Absolute Angle: 1.406438, error history: 20, history size: 20, time out time: -2147481531, error difference: 23.058281
, over slew: 0, Actual_Vel1: 47.200000, Actual_Vel2: -52.800000, Actual_Vel3: 50.800000, Actual_Vel4: -50.600000
    117:   27820 [INFO] CHASSIS_PID_TURN, Time: 2928, Actual_Vol1: 1632.000000, Actual_Vol2: -942.000000, Actual_Vol3: 1023.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48057.897305, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 126.000000, position_r: -148.000000, Heading_Sp: 90.000000, Relative_Heading: 81.013886, Absolute Angle: 1.413815, error history: 20, history size: 20, time out time: -2147481531, error difference: 21.939389,
over slew: 0, Actual_Vel1: 42.600000, Actual_Vel2: -48.800000, Actual_Vel3: 44.600000, Actual_Vel4: -42.000000
    118:   27820 [INFO] CHASSIS_PID_TURN, Time: 2938, Actual_Vol1: 1626.000000, Actual_Vol2: -862.000000, Actual_Vol3: 992.000000, Actual_Vol4: -12.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48144.969708, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 81.292760, Absolute Angle: 1.418682, error history: 20, history size: 20, time out time: -2147481531, error difference: 20.684173, o
ver slew: 0, Actual_Vel1: 40.400000, Actual_Vel2: -42.800000, Actual_Vel3: 39.400000, Actual_Vel4: -32.400000
    119:   27820 [INFO] CHASSIS_PID_TURN, Time: 2948, Actual_Vol1: 1620.000000, Actual_Vol2: -918.000000, Actual_Vol3: 1023.000000, Actual_Vol4: -142.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48225.722402, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 81.924731, Absolute Angle: 1.429712, error history: 20, history size: 20, time out time: -2147481531, error difference: 19.803239
, over slew: 0, Actual_Vel1: 38.200000, Actual_Vel2: -32.200000, Actual_Vel3: 37.600000, Actual_Vel4: -27.800000
    120:   27822 [INFO] CHASSIS_PID_TURN, Time: 2958, Actual_Vol1: 1614.000000, Actual_Vol2: -1016.000000, Actual_Vol3: 1047.000000, Actual_Vol4: -154.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48304.506101, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 128.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.121630, Absolute Angle: 1.433149, error history: 20, history size: 20, time out time: -2147481531, error difference: 18.50755
5, over slew: 0, Actual_Vel1: 32.600000, Actual_Vel2: -27.800000, Actual_Vel3: 34.600000, Actual_Vel4: -26.000000
    121:   27822 [INFO] CHASSIS_PID_TURN, Time: 2968, Actual_Vol1: 1657.000000, Actual_Vol2: -1060.000000, Actual_Vol3: 1090.000000, Actual_Vol4: -246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48381.652920, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 129.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.285318, Absolute Angle: 1.436005, error history: 20, history size: 20, time out time: -2147481531, error difference: 17.10040
2, over slew: 0, Actual_Vel1: 32.600000, Actual_Vel2: -26.800000, Actual_Vel3: 30.200000, Actual_Vel4: -23.400000
    122:   27822 [INFO] CHASSIS_PID_TURN, Time: 2978, Actual_Vol1: 1873.000000, Actual_Vol2: -1214.000000, Actual_Vol3: 1207.000000, Actual_Vol4: -339.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48456.512630, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 130.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.514029, Absolute Angle: 1.439997, error history: 20, history size: 20, time out time: -2147481531, error difference: 15.88674
8, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -28.800000, Actual_Vel3: 28.800000, Actual_Vel4: -22.800000
    123:   27824 [INFO] CHASSIS_PID_TURN, Time: 2988, Actual_Vol1: 1990.000000, Actual_Vol2: -1275.000000, Actual_Vol3: 1257.000000, Actual_Vol4: -561.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48529.313264, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 130.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.719937, Absolute Angle: 1.443591, error history: 20, history size: 20, time out time: -2147481531, error difference: 14.67281
3, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -34.000000, Actual_Vel3: 25.600000, Actual_Vel4: -20.200000
    124:   27824 [INFO] CHASSIS_PID_TURN, Time: 2998, Actual_Vol1: 2027.000000, Actual_Vol2: -1281.000000, Actual_Vol3: 1349.000000, Actual_Vol4: -788.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48601.212909, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 132.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 82.810036, Absolute Angle: 1.445164, error history: 20, history size: 20, time out time: -2147481531, error difference: 13.46524
0, over slew: 0, Actual_Vel1: 27.400000, Actual_Vel2: -35.800000, Actual_Vel3: 24.400000, Actual_Vel4: -19.000000
    125:   27824 [INFO] CHASSIS_PID_TURN, Time: 3008, Actual_Vol1: 2107.000000, Actual_Vol2: -1287.000000, Actual_Vol3: 1441.000000, Actual_Vol4: -856.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48670.414490, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 132.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.079842, Absolute Angle: 1.449873, error history: 20, history size: 20, time out time: -2147481531, error difference: 11.65396
6, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -34.200000, Actual_Vel3: 25.000000, Actual_Vel4: -19.200000
    126:   27826 [INFO] CHASSIS_PID_TURN, Time: 3018, Actual_Vol1: 2242.000000, Actual_Vol2: -1380.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48739.071279, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 133.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.134321, Absolute Angle: 1.450823, error history: 20, history size: 20, time out time: -2147481531, error difference: 10.53799
0, over slew: 0, Actual_Vel1: 23.000000, Actual_Vel2: -32.400000, Actual_Vel3: 26.400000, Actual_Vel4: -20.200000
    127:   27826 [INFO] CHASSIS_PID_TURN, Time: 3028, Actual_Vol1: 2458.000000, Actual_Vol2: -1540.000000, Actual_Vol3: 1632.000000, Actual_Vol4: -998.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48806.355596, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 134.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.271568, Absolute Angle: 1.453219, error history: 20, history size: 20, time out time: -2147481531, error difference: 9.545978
, over slew: 0, Actual_Vel1: 23.200000, Actual_Vel2: -27.200000, Actual_Vel3: 26.200000, Actual_Vel4: -29.000000
    128:   27826 [INFO] CHASSIS_PID_TURN, Time: 3038, Actual_Vol1: 2526.000000, Actual_Vol2: -1571.000000, Actual_Vol3: 1756.000000, Actual_Vol4: -1115.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48872.371561, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 134.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.398404, Absolute Angle: 1.455432, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.58645
9, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -23.400000, Actual_Vel3: 25.000000, Actual_Vel4: -36.200000
    129:   27828 [INFO] CHASSIS_PID_TURN, Time: 3048, Actual_Vol1: 2587.000000, Actual_Vol2: -1774.000000, Actual_Vol3: 1928.000000, Actual_Vol4: -1115.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48938.171963, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 135.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.419960, Absolute Angle: 1.455809, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.66064
9, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -23.400000, Actual_Vel3: 25.800000, Actual_Vel4: -45.400000
    130:   27828 [INFO] CHASSIS_PID_TURN, Time: 3058, Actual_Vol1: 2667.000000, Actual_Vol2: -1885.000000, Actual_Vol3: 1990.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49002.811180, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 136.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.536078, Absolute Angle: 1.457835, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.87313
1, over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -23.400000, Actual_Vel3: 29.400000, Actual_Vel4: -47.600000
    131:   27828 [INFO] CHASSIS_PID_TURN, Time: 3068, Actual_Vol1: 2747.000000, Actual_Vol2: -2273.000000, Actual_Vol3: 1996.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49066.288213, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 137.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.652297, Absolute Angle: 1.459864, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.02636
2, over slew: 0, Actual_Vel1: 27.800000, Actual_Vel2: -23.400000, Actual_Vel3: 28.200000, Actual_Vel4: -51.400000
    132:   27830 [INFO] CHASSIS_PID_TURN, Time: 3078, Actual_Vol1: 2889.000000, Actual_Vol2: -2519.000000, Actual_Vol3: 2181.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49128.569187, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 138.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.771903, Absolute Angle: 1.461951, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.33199
2, over slew: 0, Actual_Vel1: 25.800000, Actual_Vel2: -23.400000, Actual_Vel3: 27.400000, Actual_Vel4: -50.600000
    133:   27830 [INFO] CHASSIS_PID_TURN, Time: 3088, Actual_Vol1: 3043.000000, Actual_Vol2: -2815.000000, Actual_Vol3: 2445.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49189.569887, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 138.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.899930, Absolute Angle: 1.464186, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.69600
2, over slew: 0, Actual_Vel1: 26.200000, Actual_Vel2: -23.600000, Actual_Vel3: 28.400000, Actual_Vel4: -51.800000
    134:   27830 [INFO] CHASSIS_PID_TURN, Time: 3098, Actual_Vol1: 3197.000000, Actual_Vol2: -3117.000000, Actual_Vol3: 2519.000000, Actual_Vol4: -1103.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49250.108965, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 140.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 83.946092, Absolute Angle: 1.464991, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.02344
4, over slew: 0, Actual_Vel1: 28.400000, Actual_Vel2: -23.600000, Actual_Vel3: 36.400000, Actual_Vel4: -52.800000
    135:   27832 [INFO] CHASSIS_PID_TURN, Time: 3108, Actual_Vol1: 3308.000000, Actual_Vol2: -3326.000000, Actual_Vol3: 2532.000000, Actual_Vol4: -1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49308.070553, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 140.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.203841, Absolute Angle: 1.469490, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.61261
0, over slew: 0, Actual_Vel1: 28.400000, Actual_Vel2: -23.600000, Actual_Vel3: 46.200000, Actual_Vel4: -48.200000
    136:   27832 [INFO] CHASSIS_PID_TURN, Time: 3118, Actual_Vol1: 3419.000000, Actual_Vol2: -3653.000000, Actual_Vol3: 2532.000000, Actual_Vol4: -1053.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49365.234823, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 141.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.283573, Absolute Angle: 1.470882, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.26968
6, over slew: 0, Actual_Vel1: 27.800000, Actual_Vel2: -19.000000, Actual_Vel3: 44.000000, Actual_Vel4: -43.400000
    137:   27832 [INFO] CHASSIS_PID_TURN, Time: 3128, Actual_Vol1: 3511.000000, Actual_Vol2: -3979.000000, Actual_Vol3: 2624.000000, Actual_Vol4: -1109.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49420.458430, kD: 35.00000
```

```
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 142.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.477639, Absolute Angle: 1.474269, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.18488
0, over slew: 0, Actual_Vel1: 30.200000, Actual_Vel2: -19.000000, Actual_Vel3: 33.200000, Actual_Vel4: -39.000000
    27834 [INFO] CHASSIS_PID_TURN, Time: 3138, Actual_Vol1: 3647.000000, Actual_Vol2: -4263.000000, Actual_Vol3: 2790.000000, Actual_Vol4: -1214.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49473.600384, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 143.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.685805, Absolute Angle: 1.477902, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.76107
4, over slew: 0, Actual_Vel1: 32.400000, Actual_Vel2: -19.000000, Actual_Vel3: 32.600000, Actual_Vel4: -30.600000
    27834 [INFO] CHASSIS_PID_TURN, Time: 3148, Actual_Vol1: 3776.000000, Actual_Vol2: -4466.000000, Actual_Vol3: 2852.000000, Actual_Vol4: -1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49524.480807, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 144.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 84.911958, Absolute Angle: 1.481849, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.79032
8, over slew: 0, Actual_Vel1: 34.600000, Actual_Vel2: -11.600000, Actual_Vel3: 39.600000, Actual_Vel4: -24.600000
    27834 [INFO] CHASSIS_PID_TURN, Time: 3158, Actual_Vol1: 3807.000000, Actual_Vol2: -4688.000000, Actual_Vol3: 2852.000000, Actual_Vol4: -1595.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49572.891563, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 145.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.158924, Absolute Angle: 1.486159, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.87360
6, over slew: 0, Actual_Vel1: 43.400000, Actual_Vel2: -9.600000, Actual_Vel3: 41.800000, Actual_Vel4: -20.600000
    27836 [INFO] CHASSIS_PID_TURN, Time: 3168, Actual_Vol1: 3819.000000, Actual_Vol2: -4897.000000, Actual_Vol3: 2889.000000, Actual_Vol4: -1669.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49618.651791, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 146.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.423977, Absolute Angle: 1.490785, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.90994
8, over slew: 0, Actual_Vel1: 44.600000, Actual_Vel2: -9.600000, Actual_Vel3: 41.200000, Actual_Vel4: -22.600000
    27836 [INFO] CHASSIS_PID_TURN, Time: 3178, Actual_Vol1: 3838.000000, Actual_Vol2: -5168.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1780.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49661.641180, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 148.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 85.701061, Absolute Angle: 1.495621, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.98112
4, over slew: 0, Actual_Vel1: 44.400000, Actual_Vel2: -9.600000, Actual_Vel3: 46.000000, Actual_Vel4: -33.000000
    27836 [INFO] CHASSIS_PID_TURN, Time: 3188, Actual_Vol1: 3942.000000, Actual_Vol2: -5341.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49700.753904, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 149.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.088728, Absolute Angle: 1.502387, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.27869
2, over slew: 0, Actual_Vel1: 48.600000, Actual_Vel2: -16.200000, Actual_Vel3: 50.800000, Actual_Vel4: -37.800000
    27838 [INFO] CHASSIS_PID_TURN, Time: 3198, Actual_Vol1: 3942.000000, Actual_Vol2: -5378.000000, Actual_Vol3: 2932.000000, Actual_Vol4: -1940.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49736.784071, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 150.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.396983, Absolute Angle: 1.507768, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.31714
1, over slew: 0, Actual_Vel1: 56.600000, Actual_Vel2: -21.800000, Actual_Vel3: 49.200000, Actual_Vel4: -41.000000
    27838 [INFO] CHASSIS_PID_TURN, Time: 3208, Actual_Vol1: 3702.000000, Actual_Vol2: -5513.000000, Actual_Vol3: 2901.000000, Actual_Vol4: -1947.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49769.467573, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 151.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 86.731650, Absolute Angle: 1.513609, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.59732
9, over slew: 0, Actual_Vel1: 57.800000, Actual_Vel2: -34.400000, Actual_Vel3: 49.200000, Actual_Vel4: -43.000000
    27838 [INFO] CHASSIS_PID_TURN, Time: 3218, Actual_Vol1: 3708.000000, Actual_Vol2: -5692.000000, Actual_Vol3: 2858.000000, Actual_Vol4: -1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49798.618332, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 152.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.084924, Absolute Angle: 1.519774, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.81335
6, over slew: 0, Actual_Vel1: 46.800000, Actual_Vel2: -31.200000, Actual_Vel3: 50.000000, Actual_Vel4: -47.000000
    27840 [INFO] CHASSIS_PID_TURN, Time: 3228, Actual_Vol1: 3788.000000, Actual_Vol2: -5760.000000, Actual_Vol3: 2778.000000, Actual_Vol4: -1829.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49824.108344, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 153.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.450999, Absolute Angle: 1.526164, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.05259
5, over slew: 0, Actual_Vel1: 52.200000, Actual_Vel2: -39.600000, Actual_Vel3: 51.200000, Actual_Vel4: -48.600000
    27840 [INFO] CHASSIS_PID_TURN, Time: 3238, Actual_Vol1: 3585.000000, Actual_Vol2: -5723.000000, Actual_Vol3: 2618.000000, Actual_Vol4: -1688.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49845.783434, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 154.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 87.832491, Absolute Angle: 1.532822, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.41253
1, over slew: 0, Actual_Vel1: 58.800000, Actual_Vel2: -40.400000, Actual_Vel3: 52.600000, Actual_Vel4: -50.600000
    27840 [INFO] CHASSIS_PID_TURN, Time: 3248, Actual_Vol1: 3456.000000, Actual_Vol2: -5723.000000, Actual_Vol3: 2569.000000, Actual_Vol4: -1669.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49863.436876, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 156.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 88.234656, Absolute Angle: 1.539841, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.69857
7, over slew: 0, Actual_Vel1: 49.000000, Actual_Vel2: -39.200000, Actual_Vel3: 56.200000, Actual_Vel4: -50.200000
    27842 [INFO] CHASSIS_PID_TURN, Time: 3258, Actual_Vol1: 3394.000000, Actual_Vol2: -5667.000000, Actual_Vol3: 2489.000000, Actual_Vol4: -1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49875.917217, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 157.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 88.751966, Absolute Angle: 1.548870, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.09966
9, over slew: 0, Actual_Vel1: 50.800000, Actual_Vel2: -40.800000, Actual_Vel3: 58.600000, Actual_Vel4: -50.400000
    27842 [INFO] CHASSIS_PID_TURN, Time: 3268, Actual_Vol1: 3197.000000, Actual_Vol2: -5544.000000, Actual_Vol3: 2372.000000, Actual_Vol4: -1368.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49884.109442, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 158.000000, position_r: -150.000000, Heading_Sp: 90.000000, Relative_Heading: 89.180778, Absolute Angle: 1.556354, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.40887
5, over slew: 0, Actual_Vel1: 52.600000, Actual_Vel2: -45.000000, Actual_Vel3: 60.600000, Actual_Vel4: -50.200000
    27842 [INFO] CHASSIS_PID_TURN, Time: 3278, Actual_Vol1: 3018.000000, Actual_Vol2: -5365.000000, Actual_Vol3: 2137.000000, Actual_Vol4: -1287.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49887.999861, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 158.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 89.610958, Absolute Angle: 1.563862, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.71102
8, over slew: 0, Actual_Vel1: 52.000000, Actual_Vel2: -46.600000, Actual_Vel3: 62.400000, Actual_Vel4: -47.800000
    27844 [INFO] CHASSIS_PID_TURN, Time: 3288, Actual_Vol1: 2969.000000, Actual_Vol2: -5267.000000, Actual_Vol3: 1953.000000, Actual_Vol4: -1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49887.538427, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 160.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 90.046143, Absolute Angle: 1.571457, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.10005
1, over slew: 0, Actual_Vel1: 52.600000, Actual_Vel2: -42.600000, Actual_Vel3: 58.400000, Actual_Vel4: -43.800000
    27844 [INFO] CHASSIS_PID_TURN, Time: 3298, Actual_Vol1: 2895.000000, Actual_Vol2: -5224.000000, Actual_Vol3: 1774.000000, Actual_Vol4: -1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49881.623834, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 161.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 90.591459, Absolute Angle: 1.580975, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.38761
8, over slew: 0, Actual_Vel1: 54.000000, Actual_Vel2: -44.800000, Actual_Vel3: 56.000000, Actual_Vel4: -43.800000
    27844 [INFO] CHASSIS_PID_TURN, Time: 3308, Actual_Vol1: 2784.000000, Actual_Vol2: -5156.000000, Actual_Vol3: 1602.000000, Actual_Vol4: -1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49871.295571, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 162.000000, position_r: -151.000000, Heading_Sp: 90.000000, Relative_Heading: 91.032826, Absolute Angle: 1.588678, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.74925
3, over slew: 0, Actual_Vel1: 52.800000, Actual_Vel2: -49.000000, Actual_Vel3: 53.400000, Actual_Vel4: -45.600000
    27846 [INFO] CHASSIS_PID_TURN, Time: 3318, Actual_Vol1: 2563.000000, Actual_Vol2: -5063.000000, Actual_Vol3: 1435.000000, Actual_Vol4: -1060.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49856.597414, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 163.000000, position_r: -152.000000, Heading_Sp: 90.000000, Relative_Heading: 91.469816, Absolute Angle: 1.596305, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.99217
6, over slew: 0, Actual_Vel1: 47.200000, Actual_Vel2: -47.200000, Actual_Vel3: 50.000000, Actual_Vel4: -46.800000
    27846 [INFO] CHASSIS_PID_TURN, Time: 3328, Actual_Vol1: 2519.000000, Actual_Vol2: -4891.000000, Actual_Vol3: 1306.000000, Actual_Vol4: -979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49836.541888, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 164.000000, position_r: -153.000000, Heading_Sp: 90.000000, Relative_Heading: 92.005553, Absolute Angle: 1.605656, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.319748
, over slew: 0, Actual_Vel1: 44.800000, Actual_Vel2: -46.000000, Actual_Vel3: 47.800000, Actual_Vel4: -46.200000
    27846 [INFO] CHASSIS_PID_TURN, Time: 3338, Actual_Vol1: 2482.000000, Actual_Vol2: -4811.000000, Actual_Vol3: 1207.000000, Actual_Vol4: -893.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49811.233384, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 165.000000, position_r: -154.000000, Heading_Sp: 90.000000, Relative_Heading: 92.530850, Absolute Angle: 1.614824, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.618893
, over slew: 0, Actual_Vel1: 46.200000, Actual_Vel2: -51.400000, Actual_Vel3: 43.800000, Actual_Vel4: -45.600000
    27848 [INFO] CHASSIS_PID_TURN, Time: 3348, Actual_Vol1: 2322.000000, Actual_Vol2: -4700.000000, Actual_Vol3: 1146.000000, Actual_Vol4: -721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49780.751273, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 165.000000, position_r: -154.000000, Heading_Sp: 90.000000, Relative_Heading: 93.048211, Absolute Angle: 1.623853, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.889287
, over slew: 0, Actual_Vel1: 44.000000, Actual_Vel2: -51.200000, Actual_Vel3: 40.600000, Actual_Vel4: -44.200000
    27848 [INFO] CHASSIS_PID_TURN, Time: 3358, Actual_Vol1: 2131.000000, Actual_Vol2: -4521.000000, Actual_Vol3: 1029.000000, Actual_Vol4: -567.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49747.189737, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -155.000000, Heading_Sp: 90.000000, Relative_Heading: 93.356154, Absolute Angle: 1.629228, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.932176
, over slew: 0, Actual_Vel1: 38.200000, Actual_Vel2: -48.600000, Actual_Vel3: 39.600000, Actual_Vel4: -38.800000
    27848 [INFO] CHASSIS_PID_TURN, Time: 3368, Actual_Vol1: 2107.000000, Actual_Vol2: -4337.000000, Actual_Vol3: 942.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49708.553689, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -156.000000, Heading_Sp: 90.000000, Relative_Heading: 93.863605, Absolute Angle: 1.638085, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.162544,
over slew: 0, Actual_Vel1: 36.200000, Actual_Vel2: -45.400000, Actual_Vel3: 37.200000, Actual_Vel4: -34.000000
    27850 [INFO] CHASSIS_PID_TURN, Time: 3378, Actual_Vol1: 2088.000000, Actual_Vol2: -4115.000000, Actual_Vol3: 856.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49666.055912, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 166.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 94.249778, Absolute Angle: 1.644825, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.161050,
over slew: 0, Actual_Vel1: 36.800000, Actual_Vel2: -43.600000, Actual_Vel3: 35.000000, Actual_Vel4: -31.000000
    27850 [INFO] CHASSIS_PID_TURN, Time: 3388, Actual_Vol1: 1959.000000, Actual_Vol2: -3881.000000, Actual_Vol3: 696.000000, Actual_Vol4: -394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49619.829748, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 94.622616, Absolute Angle: 1.651332, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.225633,
over slew: 0, Actual_Vel1: 33.000000, Actual_Vel2: -41.600000, Actual_Vel3: 33.600000, Actual_Vel4: -24.600000
    27850 [INFO] CHASSIS_PID_TURN, Time: 3398, Actual_Vol1: 1663.000000, Actual_Vol2: -3782.000000, Actual_Vol3: 616.000000, Actual_Vol4: -283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49569.077404, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 95.075234, Absolute Angle: 1.659232, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.343585,
over slew: 0, Actual_Vel1: 30.600000, Actual_Vel2: -39.600000, Actual_Vel3: 29.600000, Actual_Vel4: -21.400000
    27852 [INFO] CHASSIS_PID_TURN, Time: 3408, Actual_Vol1: 1546.000000, Actual_Vol2: -3560.000000, Actual_Vol3: 536.000000, Actual_Vol4: -234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49515.006829, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 95.407058, Absolute Angle: 1.665023, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.322133,
over slew: 0, Actual_Vel1: 29.200000, Actual_Vel2: -38.400000, Actual_Vel3: 25.600000, Actual_Vel4: -21.400000
    27852 [INFO] CHASSIS_PID_TURN, Time: 3418, Actual_Vol1: 1411.000000, Actual_Vol2: -3413.000000, Actual_Vol3: 480.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49457.745586, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 95.726124, Absolute Angle: 1.670592, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.275126,
over slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -38.600000, Actual_Vel3: 24.600000, Actual_Vel4: -21.400000
    27852 [INFO] CHASSIS_PID_TURN, Time: 3428, Actual_Vol1: 1374.000000, Actual_Vol2: -3357.000000, Actual_Vol3: 419.000000, Actual_Vol4: -388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49398.431182, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 95.931440, Absolute Angle: 1.674175, error history: 20, history size: 20, time out time: -2147481531, error difference: 8.098949,
over slew: 0, Actual_Vel1: 23.400000, Actual_Vel2: -32.200000, Actual_Vel3: 21.800000, Actual_Vel4: -21.400000
    27854 [INFO] CHASSIS_PID_TURN, Time: 3438, Actual_Vol1: 1306.000000, Actual_Vol2: -3283.000000, Actual_Vol3: 407.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49336.244616, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 96.218657, Absolute Angle: 1.679188, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.984001,
over slew: 0, Actual_Vel1: 20.800000, Actual_Vel2: -27.000000, Actual_Vel3: 20.200000, Actual_Vel4: -21.000000
    27854 [INFO] CHASSIS_PID_TURN, Time: 3448, Actual_Vol1: 1275.000000, Actual_Vol2: -3043.000000, Actual_Vol3: 265.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49272.297991, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 96.394662, Absolute Angle: 1.682260, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.642697,
over slew: 0, Actual_Vel1: 19.600000, Actual_Vel2: -27.200000, Actual_Vel3: 20.200000, Actual_Vel4: -21.000000
    27854 [INFO] CHASSIS_PID_TURN, Time: 3458, Actual_Vol1: 1220.000000, Actual_Vol2: -2907.000000, Actual_Vol3: 148.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49205.783535, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 96.651446, Absolute Angle: 1.686742, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.470668,
over slew: 0, Actual_Vel1: 19.600000, Actual_Vel2: -26.600000, Actual_Vel3: 18.200000, Actual_Vel4: -16.000000
    27856 [INFO] CHASSIS_PID_TURN, Time: 3468, Actual_Vol1: 1170.000000, Actual_Vol2: -2852.000000, Actual_Vol3: 111.000000, Actual_Vol4: -382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49136.865417, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 96.891812, Absolute Angle: 1.690937, error history: 20, history size: 20, time out time: -2147481531, error difference: 7.280854,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -26.600000, Actual_Vel3: 18.200000, Actual_Vel4: -16.000000
    27856 [INFO] CHASSIS_PID_TURN, Time: 3478, Actual_Vol1: 1164.000000, Actual_Vol2: -2834.000000, Actual_Vol3: 111.000000, Actual_Vol4: -400.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 49066.720630, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 97.014479, Absolute Angle: 1.693078, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.968335,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -21.800000, Actual_Vel3: 18.200000, Actual_Vel4: -12.400000
    27856 [INFO] CHASSIS_PID_TURN, Time: 3488, Actual_Vol1: 1146.000000, Actual_Vol2: -2784.000000, Actual_Vol3: 197.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48995.496073, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 97.122456, Absolute Angle: 1.694962, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.530996,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -20.200000, Actual_Vel3: 18.200000, Actual_Vel4: -12.400000
    27858 [INFO] CHASSIS_PID_TURN, Time: 3498, Actual_Vol1: 1152.000000, Actual_Vol2: -2747.000000, Actual_Vol3: 290.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48922.389438, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.310663, Absolute Angle: 1.698247, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.277837,
over slew: 0, Actual_Vel1: 16.800000, Actual_Vel2: -18.600000, Actual_Vel3: 18.200000, Actual_Vel4: -10.600000
```

```
27858 [INFO] CHASSIS_PID_TURN, Time: 3508, Actual_Vol1: 1220.000000, Actual_Vol2: -2569.000000, Actual_Vol3: 308.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48847.519744, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.486969, Absolute Angle: 1.701324, error history: 20, history size: 20, time out time: -2147481531, error difference: 6.017154,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -18.600000, Actual_Vel3: 16.000000, Actual_Vel4: -10.600000
27858 [INFO] CHASSIS_PID_TURN, Time: 3518, Actual_Vol1: 1250.000000, Actual_Vol2: -2655.000000, Actual_Vol3: 388.000000, Actual_Vol4: -382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48772.094316, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.542543, Absolute Angle: 1.702294, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.536990,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -18.600000, Actual_Vel3: 16.000000, Actual_Vel4: -10.600000
27860 [INFO] CHASSIS_PID_TURN, Time: 3528, Actual_Vol1: 1257.000000, Actual_Vol2: -2717.000000, Actual_Vol3: 437.000000, Actual_Vol4: -265.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48696.299860, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.579446, Absolute Angle: 1.702938, error history: 20, history size: 20, time out time: -2147481531, error difference: 5.048595,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -19.200000, Actual_Vel3: 13.000000, Actual_Vel4: -8.800000
27860 [INFO] CHASSIS_PID_TURN, Time: 3538, Actual_Vol1: 1318.000000, Actual_Vol2: -2606.000000, Actual_Vol3: 419.000000, Actual_Vol4: -216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48620.253571, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.604629, Absolute Angle: 1.703378, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.556418,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -19.200000, Actual_Vel3: 13.000000, Actual_Vel4: -8.800000
27860 [INFO] CHASSIS_PID_TURN, Time: 3548, Actual_Vol1: 1337.000000, Actual_Vol2: -2673.000000, Actual_Vol3: 419.000000, Actual_Vol4: -302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48544.124725, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.612885, Absolute Angle: 1.703522, error history: 20, history size: 20, time out time: -2147481531, error difference: 4.256731,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.800000
27862 [INFO] CHASSIS_PID_TURN, Time: 3558, Actual_Vol1: 1343.000000, Actual_Vol2: -2544.000000, Actual_Vol3: 394.000000, Actual_Vol4: -320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48468.107629, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.601710, Absolute Angle: 1.703327, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.749280,
over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.800000
27862 [INFO] CHASSIS_PID_TURN, Time: 3568, Actual_Vol1: 1411.000000, Actual_Vol2: -2513.000000, Actual_Vol3: 314.000000, Actual_Vol4: -400.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48392.262733, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.584490, Absolute Angle: 1.703026, error history: 20, history size: 20, time out time: -2147481531, error difference: 3.363107,
over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000
27862 [INFO] CHASSIS_PID_TURN, Time: 3578, Actual_Vol1: 1435.000000, Actual_Vol2: -2649.000000, Actual_Vol3: 394.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48316.540486, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.572225, Absolute Angle: 1.702812, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.990268,
over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -20.000000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000
27864 [INFO] CHASSIS_PID_TURN, Time: 3588, Actual_Vol1: 1540.000000, Actual_Vol2: -2717.000000, Actual_Vol3: 419.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48240.898340, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.564215, Absolute Angle: 1.702673, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.537650,
over slew: 0, Actual_Vel1: 9.400000, Actual_Vel2: -17.600000, Actual_Vel3: 10.400000, Actual_Vel4: -8.400000
27864 [INFO] CHASSIS_PID_TURN, Time: 3598, Actual_Vol1: 1565.000000, Actual_Vol2: -2710.000000, Actual_Vol3: 468.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48165.358755, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.553959, Absolute Angle: 1.702494, error history: 20, history size: 20, time out time: -2147481531, error difference: 2.205827,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -17.600000, Actual_Vel3: 9.600000, Actual_Vel4: -8.000000
27864 [INFO] CHASSIS_PID_TURN, Time: 3608, Actual_Vol1: 1577.000000, Actual_Vol2: -2784.000000, Actual_Vol3: 511.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48089.902415, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.545634, Absolute Angle: 1.702348, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.886760,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -17.600000, Actual_Vel3: 9.600000, Actual_Vel4: -8.000000
27866 [INFO] CHASSIS_PID_TURN, Time: 3618, Actual_Vol1: 1700.000000, Actual_Vol2: -2797.000000, Actual_Vol3: 517.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 48014.533633, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.536878, Absolute Angle: 1.702195, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.681444,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -7.400000
27866 [INFO] CHASSIS_PID_TURN, Time: 3628, Actual_Vol1: 1743.000000, Actual_Vol2: -2790.000000, Actual_Vol3: 561.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47939.225646, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.530799, Absolute Angle: 1.702089, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.394228,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000
27866 [INFO] CHASSIS_PID_TURN, Time: 3638, Actual_Vol1: 1823.000000, Actual_Vol2: -2809.000000, Actual_Vol3: 598.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47863.939426, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 97.528622, Absolute Angle: 1.702051, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.218222,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -18.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000
27868 [INFO] CHASSIS_PID_TURN, Time: 3648, Actual_Vol1: 1860.000000, Actual_Vol2: -2883.000000, Actual_Vol3: 690.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47788.657116, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.528231, Absolute Angle: 1.702044, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.961439,
over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 9.600000, Actual_Vel4: -9.000000
27868 [INFO] CHASSIS_PID_TURN, Time: 3658, Actual_Vol1: 1866.000000, Actual_Vol2: -2889.000000, Actual_Vol3: 862.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47712.360770, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.629635, Absolute Angle: 1.703814, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.737823,
over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 10.000000, Actual_Vel4: -8.800000
27868 [INFO] CHASSIS_PID_TURN, Time: 3668, Actual_Vol1: 2027.000000, Actual_Vol2: -3037.000000, Actual_Vol3: 912.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47636.060306, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.630046, Absolute Angle: 1.703822, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.615568,
over slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: -11.600000, Actual_Vel3: 10.000000, Actual_Vel4: -11.200000
27870 [INFO] CHASSIS_PID_TURN, Time: 3678, Actual_Vol1: 1940.000000, Actual_Vol2: -3166.000000, Actual_Vol3: 967.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47559.727195, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.633311, Absolute Angle: 1.703878, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.510855,
over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -11.200000
27870 [INFO] CHASSIS_PID_TURN, Time: 3688, Actual_Vol1: 2014.000000, Actual_Vol2: -3265.000000, Actual_Vol3: 967.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47483.354396, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.637280, Absolute Angle: 1.703948, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.326616,
over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -12.400000
27870 [INFO] CHASSIS_PID_TURN, Time: 3698, Actual_Vol1: 2057.000000, Actual_Vol2: -3289.000000, Actual_Vol3: 973.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47405.941914, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.741248, Absolute Angle: 1.705762, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.254279,
over slew: 0, Actual_Vel1: 7.600000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -12.800000
27872 [INFO] CHASSIS_PID_TURN, Time: 3708, Actual_Vol1: 2070.000000, Actual_Vol2: -3302.000000, Actual_Vol3: 1016.000000, Actual_Vol4: -419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47328.432484, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.750943, Absolute Angle: 1.705932, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.222712
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 9.000000, Actual_Vel4: -14.000000
27872 [INFO] CHASSIS_PID_TURN, Time: 3718, Actual_Vol1: 2070.000000, Actual_Vol2: -3363.000000, Actual_Vol3: 1010.000000, Actual_Vol4: -419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47250.835594, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.759689, Absolute Angle: 1.706084, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.231458
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -14.000000
27872 [INFO] CHASSIS_PID_TURN, Time: 3728, Actual_Vol1: 2070.000000, Actual_Vol2: -3382.000000, Actual_Vol3: 1016.000000, Actual_Vol4: -388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47173.122408, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.771319, Absolute Angle: 1.706287, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.243088
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -13.400000
27874 [INFO] CHASSIS_PID_TURN, Time: 3738, Actual_Vol1: 2070.000000, Actual_Vol2: -3610.000000, Actual_Vol3: 1053.000000, Actual_Vol4: -376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47095.335467, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.778694, Absolute Angle: 1.706416, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.250463
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -13.400000
27874 [INFO] CHASSIS_PID_TURN, Time: 3748, Actual_Vol1: 2193.000000, Actual_Vol2: -3665.000000, Actual_Vol3: 1096.000000, Actual_Vol4: -370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 47017.507757, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.782771, Absolute Angle: 1.706487, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.254540
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 15.400000, Actual_Vel4: -12.600000
27874 [INFO] CHASSIS_PID_TURN, Time: 3758, Actual_Vol1: 2242.000000, Actual_Vol2: -3665.000000, Actual_Vol3: 1103.000000, Actual_Vol4: -363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46939.633743, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.787401, Absolute Angle: 1.706568, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.259170
, over slew: 0, Actual_Vel1: 10.800000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -12.600000
27876 [INFO] CHASSIS_PID_TURN, Time: 3768, Actual_Vol1: 2267.000000, Actual_Vol2: -3745.000000, Actual_Vol3: 1103.000000, Actual_Vol4: -246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46861.719210, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.791453, Absolute Angle: 1.706639, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.263222
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000
27876 [INFO] CHASSIS_PID_TURN, Time: 3778, Actual_Vol1: 2261.000000, Actual_Vol2: -3696.000000, Actual_Vol3: 1096.000000, Actual_Vol4: -203.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46783.717663, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.800155, Absolute Angle: 1.706790, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.271924
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -0.000000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000
27876 [INFO] CHASSIS_PID_TURN, Time: 3788, Actual_Vol1: 2137.000000, Actual_Vol2: -3819.000000, Actual_Vol3: 1177.000000, Actual_Vol4: -314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46705.598880, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.811878, Absolute Angle: 1.706995, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.283647
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000
27878 [INFO] CHASSIS_PID_TURN, Time: 3798, Actual_Vol1: 2205.000000, Actual_Vol2: -3714.000000, Actual_Vol3: 1226.000000, Actual_Vol4: -407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46627.353858, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 97.824502, Absolute Angle: 1.707215, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.296271
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 19.000000, Actual_Vel4: -10.600000
27878 [INFO] CHASSIS_PID_TURN, Time: 3808, Actual_Vol1: 2242.000000, Actual_Vol2: -3868.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46548.984715, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.836914, Absolute Angle: 1.707432, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.308683
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 46.800000, Actual_Vel4: -10.600000
27878 [INFO] CHASSIS_PID_TURN, Time: 3818, Actual_Vol1: 2341.000000, Actual_Vol2: -3887.000000, Actual_Vol3: 1250.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46469.472234, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.951248, Absolute Angle: 1.709428, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.423017
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 46.800000, Actual_Vel4: -11.200000
27880 [INFO] CHASSIS_PID_TURN, Time: 3828, Actual_Vol1: 2372.000000, Actual_Vol2: -3961.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46389.847372, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.962486, Absolute Angle: 1.709624, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.434255
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -11.200000
27880 [INFO] CHASSIS_PID_TURN, Time: 3838, Actual_Vol1: 2433.000000, Actual_Vol2: -3979.000000, Actual_Vol3: 1250.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46310.051927, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 97.979544, Absolute Angle: 1.709921, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.451313
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -11.200000
27880 [INFO] CHASSIS_PID_TURN, Time: 3848, Actual_Vol1: 2458.000000, Actual_Vol2: -4115.000000, Actual_Vol3: 1244.000000, Actual_Vol4: -487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46229.120042, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.093189, Absolute Angle: 1.711905, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.463554
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -11.600000, Actual_Vel3: 20.600000, Actual_Vel4: -18.200000
27882 [INFO] CHASSIS_PID_TURN, Time: 3858, Actual_Vol1: 2464.000000, Actual_Vol2: -4121.000000, Actual_Vol3: 1312.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46148.036550, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.108349, Absolute Angle: 1.712169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.478303
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 20.600000, Actual_Vel4: -18.200000
27882 [INFO] CHASSIS_PID_TURN, Time: 3868, Actual_Vol1: 2476.000000, Actual_Vol2: -4004.000000, Actual_Vol3: 1324.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 46066.794142, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.124241, Absolute Angle: 1.712447, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.490930
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -18.200000
27882 [INFO] CHASSIS_PID_TURN, Time: 3878, Actual_Vol1: 2470.000000, Actual_Vol2: -4109.000000, Actual_Vol3: 1404.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45985.363198, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.143094, Absolute Angle: 1.712776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.505814
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -34.600000
    27884 [INFO] CHASSIS_PID_TURN, Time: 3888, Actual_Vol1: 2581.000000, Actual_Vol2: -4127.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45903.733560, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.162964, Absolute Angle: 1.713123, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.421716
, over slew: 0, Actual_Vel1: 10.400000, Actual_Vel2: -8.600000, Actual_Vel3: 17.200000, Actual_Vel4: -34.600000
    27884 [INFO] CHASSIS_PID_TURN, Time: 3898, Actual_Vol1: 2717.000000, Actual_Vol2: -4096.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45821.844177, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.188938, Absolute Angle: 1.713576, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.437995
, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
    27884 [INFO] CHASSIS_PID_TURN, Time: 3908, Actual_Vol1: 2760.000000, Actual_Vol2: -4146.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45739.650011, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 98.219417, Absolute Angle: 1.714108, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.459728
, over slew: 0, Actual_Vel1: 14.400000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
    27886 [INFO] CHASSIS_PID_TURN, Time: 3918, Actual_Vol1: 2784.000000, Actual_Vol2: -4146.000000, Actual_Vol3: 1552.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45657.167732, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.248228, Absolute Angle: 1.714611, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.476909
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 17.200000, Actual_Vel4: -28.000000
    27886 [INFO] CHASSIS_PID_TURN, Time: 3928, Actual_Vol1: 2556.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1571.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45573.456799, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.371093, Absolute Angle: 1.716755, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.592399
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 8.000000, Actual_Vel4: -32.200000
    27886 [INFO] CHASSIS_PID_TURN, Time: 3938, Actual_Vol1: 2476.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1571.000000, Actual_Vol4: -474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45489.508852, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.394795, Absolute Angle: 1.717169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.612024
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -12.600000, Actual_Vel3: 9.400000, Actual_Vel4: -29.800000
    27888 [INFO] CHASSIS_PID_TURN, Time: 3948, Actual_Vol1: 2470.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1448.000000, Actual_Vol4: -468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45405.290958, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.421789, Absolute Angle: 1.717640, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.634388
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
    27888 [INFO] CHASSIS_PID_TURN, Time: 3958, Actual_Vol1: 2587.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1423.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45319.727269, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.556369, Absolute Angle: 1.719989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.764916
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
    27888 [INFO] CHASSIS_PID_TURN, Time: 3968, Actual_Vol1: 2717.000000, Actual_Vol2: -4139.000000, Actual_Vol3: 1540.000000, Actual_Vol4: -450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45233.833637, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.589363, Absolute Angle: 1.720565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.789208
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.000000, Actual_Vel3: 9.400000, Actual_Vel4: -17.600000
    27890 [INFO] CHASSIS_PID_TURN, Time: 3978, Actual_Vol1: 2753.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1429.000000, Actual_Vol4: -456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45147.635518, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.619812, Absolute Angle: 1.721096, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.807934
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.200000, Actual_Vel3: 12.400000, Actual_Vel4: -17.600000
    27890 [INFO] CHASSIS_PID_TURN, Time: 3988, Actual_Vol1: 2821.000000, Actual_Vol2: -4152.000000, Actual_Vol3: 1417.000000, Actual_Vol4: -480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 45061.136110, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.649941, Absolute Angle: 1.721622, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.825439
, over slew: 0, Actual_Vel1: 14.200000, Actual_Vel2: -11.200000, Actual_Vel3: 12.400000, Actual_Vel4: -17.600000
    27890 [INFO] CHASSIS_PID_TURN, Time: 3998, Actual_Vol1: 2852.000000, Actual_Vol2: -4158.000000, Actual_Vol3: 1349.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44974.370561, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.676555, Absolute Angle: 1.722087, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.839641
, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 11.600000, Actual_Vel4: -17.600000
    27892 [INFO] CHASSIS_PID_TURN, Time: 4008, Actual_Vol1: 2587.000000, Actual_Vol2: -4176.000000, Actual_Vol3: 1411.000000, Actual_Vol4: -536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44887.340193, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.703037, Absolute Angle: 1.722549, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.751789
, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 11.600000, Actual_Vel4: -17.600000
    27892 [INFO] CHASSIS_PID_TURN, Time: 4018, Actual_Vol1: 2495.000000, Actual_Vol2: -4183.000000, Actual_Vol3: 1423.000000, Actual_Vol4: -524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44800.018468, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.732172, Absolute Angle: 1.723057, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.769686
, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
    27892 [INFO] CHASSIS_PID_TURN, Time: 4028, Actual_Vol1: 2612.000000, Actual_Vol2: -4170.000000, Actual_Vol3: 1343.000000, Actual_Vol4: -591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44712.452141, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.756633, Absolute Angle: 1.723484, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.777088
, over slew: 0, Actual_Vel1: 21.000000, Actual_Vel2: -11.200000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
    27894 [INFO] CHASSIS_PID_TURN, Time: 4038, Actual_Vol1: 2741.000000, Actual_Vol2: -4164.000000, Actual_Vol3: 1324.000000, Actual_Vol4: -678.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44624.665178, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.778696, Absolute Angle: 1.723869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.685508
, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 12.800000, Actual_Vel4: -17.600000
    27894 [INFO] CHASSIS_PID_TURN, Time: 4048, Actual_Vol1: 2544.000000, Actual_Vol2: -4275.000000, Actual_Vol3: 1331.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44536.591306, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.807387, Absolute Angle: 1.724370, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.699038
, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 15.600000, Actual_Vel4: -10.200000
    27894 [INFO] CHASSIS_PID_TURN, Time: 4058, Actual_Vol1: 2482.000000, Actual_Vol2: -4293.000000, Actual_Vol3: 1337.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44447.304447, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928686, Absolute Angle: 1.726487, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.804445
, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.800000, Actual_Vel3: 15.600000, Actual_Vel4: -10.200000
    27896 [INFO] CHASSIS_PID_TURN, Time: 4068, Actual_Vol1: 2470.000000, Actual_Vol2: -4275.000000, Actual_Vol3: 1331.000000, Actual_Vol4: -647.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44357.807529, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.949692, Absolute Angle: 1.726854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.806597
, over slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: -7.000000, Actual_Vel3: 15.600000, Actual_Vel4: -11.000000
    27896 [INFO] CHASSIS_PID_TURN, Time: 4078, Actual_Vol1: 764.000000, Actual_Vol2: -1318.000000, Actual_Vol3: 296.000000, Actual_Vol4: -117.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44268.144489, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.966304, Absolute Angle: 1.727144, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.803340, o
ver slew: 0, Actual_Vel1: 24.400000, Actual_Vel2: 17.600000, Actual_Vel3: 15.600000, Actual_Vel4: -12.400000
    27896 [INFO] CHASSIS_PID_TURN, Time: 4088, Actual_Vol1: 203.000000, Actual_Vol2: -425.000000, Actual_Vol3: 62.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44178.369090, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.977540, Absolute Angle: 1.727340, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.788602, over
 slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 17.600000, Actual_Vel3: 15.600000, Actual_Vel4: -12.400000
    27898 [INFO] CHASSIS_PID_TURN, Time: 4098, Actual_Vol1: 117.000000, Actual_Vol2: -228.000000, Actual_Vol3: 80.000000, Actual_Vol4: -259.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 44088.510075, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.985901, Absolute Angle: 1.727486, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.766485, ov
er slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 15.600000, Actual_Vel4: -4.000000
    27898 [INFO] CHASSIS_PID_TURN, Time: 4108, Actual_Vol1: 105.000000, Actual_Vol2: -179.000000, Actual_Vol3: 271.000000, Actual_Vol4: -407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43998.622738, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.988734, Absolute Angle: 1.727535, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.740506, ov
er slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
    27898 [INFO] CHASSIS_PID_TURN, Time: 4118, Actual_Vol1: 105.000000, Actual_Vol2: -154.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43908.789195, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.983354, Absolute Angle: 1.727441, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617640, ov
er slew: 0, Actual_Vel1: 8.600000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
    27900 [INFO] CHASSIS_PID_TURN, Time: 4128, Actual_Vol1: 92.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43819.109721, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967947, Absolute Angle: 1.727172, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.593939, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 13.000000, Actual_Vel4: -4.000000
    27900 [INFO] CHASSIS_PID_TURN, Time: 4138, Actual_Vol1: 92.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43729.547062, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956266, Absolute Angle: 1.726968, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.566944, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
    27900 [INFO] CHASSIS_PID_TURN, Time: 4148, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43640.064256, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948281, Absolute Angle: 1.726829, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.432365, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
    27902 [INFO] CHASSIS_PID_TURN, Time: 4158, Actual_Vol1: 105.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43550.586438, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.947782, Absolute Angle: 1.726820, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.399371, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 15.400000, Actual_Vel4: -4.000000
    27902 [INFO] CHASSIS_PID_TURN, Time: 4168, Actual_Vol1: 86.000000, Actual_Vol2: -142.000000, Actual_Vol3: 302.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43461.105789, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948065, Absolute Angle: 1.726825, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.368922, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
    27902 [INFO] CHASSIS_PID_TURN, Time: 4178, Actual_Vol1: 86.000000, Actual_Vol2: -129.000000, Actual_Vol3: 308.000000, Actual_Vol4: -444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43371.613608, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.949218, Absolute Angle: 1.726845, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.338793, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.800000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
    27904 [INFO] CHASSIS_PID_TURN, Time: 4188, Actual_Vol1: 99.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43282.054588, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955902, Absolute Angle: 1.726962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.312179, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: 77.000000, Actual_Vel4: 0.000000
    27904 [INFO] CHASSIS_PID_TURN, Time: 4198, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 302.000000, Actual_Vol4: -314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43192.466032, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958856, Absolute Angle: 1.727014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.285697, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27904 [INFO] CHASSIS_PID_TURN, Time: 4208, Actual_Vol1: 86.000000, Actual_Vol2: -136.000000, Actual_Vol3: 308.000000, Actual_Vol4: -308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43102.879321, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958671, Absolute Angle: 1.727010, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.256561, ov
er slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 16.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27906 [INFO] CHASSIS_PID_TURN, Time: 4218, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 413.000000, Actual_Vol4: -413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43013.286791, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959253, Absolute Angle: 1.727021, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.232101, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27906 [INFO] CHASSIS_PID_TURN, Time: 4228, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42923.707792, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957900, Absolute Angle: 1.726997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.210037, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    27906 [INFO] CHASSIS_PID_TURN, Time: 4238, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42834.101375, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960642, Absolute Angle: 1.727045, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.181347, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    27908 [INFO] CHASSIS_PID_TURN, Time: 4248, Actual_Vol1: 86.000000, Actual_Vol2: -129.000000, Actual_Vol3: 431.000000, Actual_Vol4: -431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42744.553756, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.954762, Absolute Angle: 1.726942, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.060048, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    27908 [INFO] CHASSIS_PID_TURN, Time: 4258, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 425.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42654.974120, kD: 35.000000, kI:
```

```
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957964, Absolute Angle: 1.726998, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    251:   27908 [INFO] CHASSIS_PID_TURN, Time: 4268, Actual_Vol1: 92.000000, Actual_Vol2: -136.000000, Actual_Vol3: 431.000000, Actual_Vol4: -425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42565.377039, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959708, Absolute Angle: 1.727028, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    252:   27910 [INFO] CHASSIS_PID_TURN, Time: 4278, Actual_Vol1: 191.000000, Actual_Vol2: -216.000000, Actual_Vol3: 499.000000, Actual_Vol4: -505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42475.774832, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960221, Absolute Angle: 1.727037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    253:   27910 [INFO] CHASSIS_PID_TURN, Time: 4288, Actual_Vol1: 222.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42386.182342, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959249, Absolute Angle: 1.727020, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    254:   27910 [INFO] CHASSIS_PID_TURN, Time: 4298, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42296.609836, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957251, Absolute Angle: 1.726986, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040952, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    255:   27912 [INFO] CHASSIS_PID_TURN, Time: 4308, Actual_Vol1: 222.000000, Actual_Vol2: -240.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42207.050096, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955974, Absolute Angle: 1.726963, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035572, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -11.200000
    256:   27912 [INFO] CHASSIS_PID_TURN, Time: 4318, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42117.481242, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956885, Absolute Angle: 1.726979, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020166, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    257:   27912 [INFO] CHASSIS_PID_TURN, Time: 4328, Actual_Vol1: 222.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 42027.942363, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953888, Absolute Angle: 1.726927, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    258:   27914 [INFO] CHASSIS_PID_TURN, Time: 4338, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 524.000000, Actual_Vol4: -511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41938.398771, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.954359, Absolute Angle: 1.726935, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    259:   27914 [INFO] CHASSIS_PID_TURN, Time: 4348, Actual_Vol1: 228.000000, Actual_Vol2: -246.000000, Actual_Vol3: 517.000000, Actual_Vol4: -517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41848.864117, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953465, Absolute Angle: 1.726920, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012860, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    260:   27914 [INFO] CHASSIS_PID_TURN, Time: 4358, Actual_Vol1: 302.000000, Actual_Vol2: -308.000000, Actual_Vol3: 598.000000, Actual_Vol4: -591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41759.300926, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956319, Absolute Angle: 1.726969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012577, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    261:   27916 [INFO] CHASSIS_PID_TURN, Time: 4368, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 610.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41669.698142, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960278, Absolute Angle: 1.727038, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011424, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    262:   27916 [INFO] CHASSIS_PID_TURN, Time: 4378, Actual_Vol1: 308.000000, Actual_Vol2: -370.000000, Actual_Vol3: 622.000000, Actual_Vol4: -604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41580.074952, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962319, Absolute Angle: 1.727074, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008854, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    263:   27916 [INFO] CHASSIS_PID_TURN, Time: 4388, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 616.000000, Actual_Vol4: -610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41490.451738, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962321, Absolute Angle: 1.727074, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008856, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    264:   27918 [INFO] CHASSIS_PID_TURN, Time: 4398, Actual_Vol1: 308.000000, Actual_Vol2: -333.000000, Actual_Vol3: 616.000000, Actual_Vol4: -610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41400.810398, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964134, Absolute Angle: 1.727106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010669, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    265:   27918 [INFO] CHASSIS_PID_TURN, Time: 4408, Actual_Vol1: 314.000000, Actual_Vol2: -326.000000, Actual_Vol3: 616.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41311.139810, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967059, Absolute Angle: 1.727157, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    266:   27918 [INFO] CHASSIS_PID_TURN, Time: 4418, Actual_Vol1: 308.000000, Actual_Vol2: -326.000000, Actual_Vol3: 622.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41221.531368, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960844, Absolute Angle: 1.727048, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    267:   27920 [INFO] CHASSIS_PID_TURN, Time: 4428, Actual_Vol1: 314.000000, Actual_Vol2: -333.000000, Actual_Vol3: 622.000000, Actual_Vol4: -616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41131.927400, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960397, Absolute Angle: 1.727041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    268:   27920 [INFO] CHASSIS_PID_TURN, Time: 4438, Actual_Vol1: 314.000000, Actual_Vol2: -363.000000, Actual_Vol3: 610.000000, Actual_Vol4: -622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 41042.356648, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957075, Absolute Angle: 1.726983, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    269:   27920 [INFO] CHASSIS_PID_TURN, Time: 4448, Actual_Vol1: 388.000000, Actual_Vol2: -333.000000, Actual_Vol3: 610.000000, Actual_Vol4: -604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40952.777543, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957910, Absolute Angle: 1.726997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    270:   27922 [INFO] CHASSIS_PID_TURN, Time: 4458, Actual_Vol1: 425.000000, Actual_Vol2: -431.000000, Actual_Vol3: 690.000000, Actual_Vol4: -653.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40863.135917, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964163, Absolute Angle: 1.727106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    271:   27922 [INFO] CHASSIS_PID_TURN, Time: 4468, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -702.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40773.476558, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965936, Absolute Angle: 1.727137, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013593, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    272:   27922 [INFO] CHASSIS_PID_TURN, Time: 4478, Actual_Vol1: 431.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -702.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40683.805494, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.967106, Absolute Angle: 1.727158, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013641, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    273:   27924 [INFO] CHASSIS_PID_TURN, Time: 4488, Actual_Vol1: 437.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40594.176729, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962876, Absolute Angle: 1.727084, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013641, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    274:   27924 [INFO] CHASSIS_PID_TURN, Time: 4498, Actual_Vol1: 437.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40504.473495, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970323, Absolute Angle: 1.727214, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016858, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    275:   27924 [INFO] CHASSIS_PID_TURN, Time: 4508, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40414.745208, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.972829, Absolute Angle: 1.727257, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    276:   27926 [INFO] CHASSIS_PID_TURN, Time: 4518, Actual_Vol1: 437.000000, Actual_Vol2: -444.000000, Actual_Vol3: 708.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40325.055298, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.968991, Absolute Angle: 1.727191, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    277:   27926 [INFO] CHASSIS_PID_TURN, Time: 4528, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 715.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40235.343287, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.971201, Absolute Angle: 1.727229, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    278:   27926 [INFO] CHASSIS_PID_TURN, Time: 4538, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40145.617896, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.972539, Absolute Angle: 1.727252, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019363, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    279:   27928 [INFO] CHASSIS_PID_TURN, Time: 4548, Actual_Vol1: 437.000000, Actual_Vol2: -450.000000, Actual_Vol3: 708.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 40055.909833, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970806, Absolute Angle: 1.727222, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016510, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    280:   27928 [INFO] CHASSIS_PID_TURN, Time: 4558, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39966.260268, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964956, Absolute Angle: 1.727120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    281:   27928 [INFO] CHASSIS_PID_TURN, Time: 4568, Actual_Vol1: 431.000000, Actual_Vol2: -450.000000, Actual_Vol3: 702.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39876.647964, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961230, Absolute Angle: 1.727055, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    282:   27930 [INFO] CHASSIS_PID_TURN, Time: 4578, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 708.000000, Actual_Vol4: -708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39787.018388, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962958, Absolute Angle: 1.727085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    283:   27930 [INFO] CHASSIS_PID_TURN, Time: 4588, Actual_Vol1: 431.000000, Actual_Vol2: -456.000000, Actual_Vol3: 702.000000, Actual_Vol4: -715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39697.403715, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961467, Absolute Angle: 1.727059, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    284:   27930 [INFO] CHASSIS_PID_TURN, Time: 4598, Actual_Vol1: 92.000000, Actual_Vol2: -105.000000, Actual_Vol3: 136.000000, Actual_Vol4: -123.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39607.754314, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.964940, Absolute Angle: 1.727120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    285:   27932 [INFO] CHASSIS_PID_TURN, Time: 4608, Actual_Vol1: 18.000000, Actual_Vol2: -25.000000, Actual_Vol3: 31.000000, Actual_Vol4: -18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39518.044655, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970966, Absolute Angle: 1.727225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over sl
ew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    286:   27932 [INFO] CHASSIS_PID_TURN, Time: 4618, Actual_Vol1: 0.000000, Actual_Vol2: -12.000000, Actual_Vol3: 12.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39428.331340, kD: 35.000000, kI: 0.0007
00, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.971332, Absolute Angle: 1.727231, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015754, over sle
w: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    287:   27932 [INFO] CHASSIS_PID_TURN, Time: 4628, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39338.579359, kD: 35.000000, kI: 0.000700
, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.975198, Absolute Angle: 1.727299, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018123, over slew:
0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
```

```
    288:   27934 [INFO] CHASSIS_PID_TURN, Time: 4638, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39248.876868, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.970249, Absolute Angle: 1.727212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017288, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    289:   27934 [INFO] CHASSIS_PID_TURN, Time: 4648, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39159.193703, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.968317, Absolute Angle: 1.727179, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    290:   27934 [INFO] CHASSIS_PID_TURN, Time: 4658, Actual_Vol1: -6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 39069.539490, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965421, Absolute Angle: 1.727128, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    291:   27936 [INFO] CHASSIS_PID_TURN, Time: 4668, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38979.882424, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.965707, Absolute Angle: 1.727133, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    292:   27936 [INFO] CHASSIS_PID_TURN, Time: 4678, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38890.263209, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961921, Absolute Angle: 1.727067, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013968, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    293:   27936 [INFO] CHASSIS_PID_TURN, Time: 4688, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38800.688767, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957444, Absolute Angle: 1.726989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017754, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    294:   27938 [INFO] CHASSIS_PID_TURN, Time: 4698, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38711.135370, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955340, Absolute Angle: 1.726952, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0,
  Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    295:   27938 [INFO] CHASSIS_PID_TURN, Time: 4708, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38621.513387, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962198, Absolute Angle: 1.727072, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    296:   27938 [INFO] CHASSIS_PID_TURN, Time: 4718, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38531.900666, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.961272, Absolute Angle: 1.727056, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    297:   27940 [INFO] CHASSIS_PID_TURN, Time: 4728, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38442.322121, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957855, Absolute Angle: 1.726996, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    298:   27940 [INFO] CHASSIS_PID_TURN, Time: 4738, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38352.749563, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957256, Absolute Angle: 1.726986, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    299:   27940 [INFO] CHASSIS_PID_TURN, Time: 4748, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: -6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38263.190770, kD: 35.000000, kI: 0.00070
 0, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.955879, Absolute Angle: 1.726962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew:
  0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    300:   27942 [INFO] CHASSIS_PID_TURN, Time: 4758, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38173.615329, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957544, Absolute Angle: 1.726991, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew: 0,
  Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    301:   27942 [INFO] CHASSIS_PID_TURN, Time: 4768, Actual_Vol1: 6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 38084.020793, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.959454, Absolute Angle: 1.727024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    302:   27942 [INFO] CHASSIS_PID_TURN, Time: 4778, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37994.416885, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.960391, Absolute Angle: 1.727040, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019858, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    303:   27944 [INFO] CHASSIS_PID_TURN, Time: 4788, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37904.878121, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953876, Absolute Angle: 1.726927, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021322, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
    304:   27944 [INFO] CHASSIS_PID_TURN, Time: 4798, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37815.349362, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.952876, Absolute Angle: 1.726909, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew: 0,
  Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    305:   27944 [INFO] CHASSIS_PID_TURN, Time: 4808, Actual_Vol1: -6.000000, Actual_Vol2: 0.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37725.779093, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957027, Absolute Angle: 1.726982, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    306:   27946 [INFO] CHASSIS_PID_TURN, Time: 4818, Actual_Vol1: -6.000000, Actual_Vol2: -6.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37636.208280, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.957081, Absolute Angle: 1.726983, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022322, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    307:   27946 [INFO] CHASSIS_PID_TURN, Time: 4828, Actual_Vol1: 0.000000, Actual_Vol2: 6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37546.676796, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953148, Absolute Angle: 1.726914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017373, over slew: 0,
  Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    308:   27946 [INFO] CHASSIS_PID_TURN, Time: 4838, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37457.204253, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.947254, Absolute Angle: 1.726811, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021062, over slew: 0,
  Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    309:   27948 [INFO] CHASSIS_PID_TURN, Time: 4848, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37367.680406, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.952385, Absolute Angle: 1.726901, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018452, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    310:   27948 [INFO] CHASSIS_PID_TURN, Time: 4858, Actual_Vol1: 0.000000, Actual_Vol2: 0.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37278.144835, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.953557, Absolute Angle: 1.726921, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018452, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    311:   27948 [INFO] CHASSIS_PID_TURN, Time: 4868, Actual_Vol1: 0.000000, Actual_Vol2: -6.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37188.561212, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958362, Absolute Angle: 1.727005, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014944, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    312:   27950 [INFO] CHASSIS_PID_TURN, Time: 4878, Actual_Vol1: 6.000000, Actual_Vol2: 0.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37098.921311, kD: 35.000000, kI: 0.000700,
 kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.963990, Absolute Angle: 1.727103, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew: 0
 , Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    313:   27950 [INFO] CHASSIS_PID_TURN, Time: 4888, Actual_Vol1: -6.000000, Actual_Vol2: 62.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 37009.340856, kD: 35.000000, kI: 0.00070
 0, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958046, Absolute Angle: 1.726999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew:
  0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    314:   27950 [INFO] CHASSIS_PID_TURN, Time: 4898, Actual_Vol1: -80.000000, Actual_Vol2: 62.000000, Actual_Vol3: 6.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36919.752900, kD: 35.000000, kI: 0.0007
 00, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958796, Absolute Angle: 1.727013, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sle
 w: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    315:   27952 [INFO] CHASSIS_PID_TURN, Time: 4908, Actual_Vol1: -18.000000, Actual_Vol2: 12.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36830.127824, kD: 35.000000, kI: 0.0007
 00, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.962508, Absolute Angle: 1.727077, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sle
 w: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    316:   27952 [INFO] CHASSIS_PID_TURN, Time: 4918, Actual_Vol1: 0.000000, Actual_Vol2: 68.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36740.541347, kD: 35.000000, kI: 0.000700
 , kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958648, Absolute Angle: 1.727010, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over slew:
 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    317:   27952 [INFO] CHASSIS_PID_TURN, Time: 4928, Actual_Vol1: -74.000000, Actual_Vol2: 209.000000, Actual_Vol3: 0.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36650.976928, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956442, Absolute Angle: 1.726971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sle
 w: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    318:   27954 [INFO] CHASSIS_PID_TURN, Time: 4938, Actual_Vol1: -277.000000, Actual_Vol2: 320.000000, Actual_Vol3: -6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36561.409907, kD: 35.000000, kI: 0.0
 00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956702, Absolute Angle: 1.726976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over s
 lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    319:   27954 [INFO] CHASSIS_PID_TURN, Time: 4948, Actual_Vol1: -142.000000, Actual_Vol2: 370.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36471.829030, kD: 35.000000, kI: 0.00
 0700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.958088, Absolute Angle: 1.727000, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sl
 ew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    320:   27954 [INFO] CHASSIS_PID_TURN, Time: 4958, Actual_Vol1: -105.000000, Actual_Vol2: 376.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36382.268935, kD: 35.000000, kI: 0.00
 0700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.956009, Absolute Angle: 1.726964, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sl
 ew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    321:   27956 [INFO] CHASSIS_PID_TURN, Time: 4968, Actual_Vol1: -99.000000, Actual_Vol2: 382.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36292.779706, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.948923, Absolute Angle: 1.726840, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016736, over sle
 w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    322:   27956 [INFO] CHASSIS_PID_TURN, Time: 4978, Actual_Vol1: -99.000000, Actual_Vol2: 370.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36203.318032, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.946167, Absolute Angle: 1.726792, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017823, over sle
 w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    323:   27956 [INFO] CHASSIS_PID_TURN, Time: 4988, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36113.907236, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.941080, Absolute Angle: 1.726703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022910, over sle
 w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    324:   27958 [INFO] CHASSIS_PID_TURN, Time: 4998, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 36024.521528, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938571, Absolute Angle: 1.726660, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025419, over sle
 w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    325:   27958 [INFO] CHASSIS_PID_TURN, Time: 5008, Actual_Vol1: -92.000000, Actual_Vol2: 326.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35935.199594, kD: 35.000000, kI: 0.000
 700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.932193, Absolute Angle: 1.726548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031797, over sle
```

```
w: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27958 [INFO] CHASSIS_PID_TURN, Time: 5018, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35845.904096, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929550, Absolute Angle: 1.726502, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.034440, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27960 [INFO] CHASSIS_PID_TURN, Time: 5028, Actual_Vol1: -86.000000, Actual_Vol2: 320.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35756.633572, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927052, Absolute Angle: 1.726459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27960 [INFO] CHASSIS_PID_TURN, Time: 5038, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35667.345778, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928779, Absolute Angle: 1.726489, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27960 [INFO] CHASSIS_PID_TURN, Time: 5048, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35578.064756, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928102, Absolute Angle: 1.726477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27962 [INFO] CHASSIS_PID_TURN, Time: 5058, Actual_Vol1: -92.000000, Actual_Vol2: 333.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35488.754425, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.931033, Absolute Angle: 1.726528, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27962 [INFO] CHASSIS_PID_TURN, Time: 5068, Actual_Vol1: -92.000000, Actual_Vol2: 320.000000, Actual_Vol3: 0.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35399.396677, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.935775, Absolute Angle: 1.726611, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036938, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27962 [INFO] CHASSIS_PID_TURN, Time: 5078, Actual_Vol1: -92.000000, Actual_Vol2: 326.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35310.014605, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938207, Absolute Angle: 1.726653, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: -24.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27964 [INFO] CHASSIS_PID_TURN, Time: 5088, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 6.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35220.660700, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.935390, Absolute Angle: 1.726604, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27964 [INFO] CHASSIS_PID_TURN, Time: 5098, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35131.324033, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.933667, Absolute Angle: 1.726574, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035455, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27964 [INFO] CHASSIS_PID_TURN, Time: 5108, Actual_Vol1: -86.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 35041.977290, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.934674, Absolute Angle: 1.726592, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031595, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27966 [INFO] CHASSIS_PID_TURN, Time: 5118, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: 0.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34952.593748, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938354, Absolute Angle: 1.726656, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27966 [INFO] CHASSIS_PID_TURN, Time: 5128, Actual_Vol1: -92.000000, Actual_Vol2: 363.000000, Actual_Vol3: -6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34863.210454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938329, Absolute Angle: 1.726655, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27966 [INFO] CHASSIS_PID_TURN, Time: 5138, Actual_Vol1: -92.000000, Actual_Vol2: 370.000000, Actual_Vol3: 6.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34773.821228, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.938923, Absolute Angle: 1.726666, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031035, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27968 [INFO] CHASSIS_PID_TURN, Time: 5148, Actual_Vol1: -92.000000, Actual_Vol2: 407.000000, Actual_Vol3: 6.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34684.401022, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.942021, Absolute Angle: 1.726720, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028957, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27968 [INFO] CHASSIS_PID_TURN, Time: 5158, Actual_Vol1: -265.000000, Actual_Vol2: 437.000000, Actual_Vol3: -80.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34594.998793, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.940223, Absolute Angle: 1.726688, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021871, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27968 [INFO] CHASSIS_PID_TURN, Time: 5168, Actual_Vol1: -136.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34505.561789, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.943700, Absolute Angle: 1.726749, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019115, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27970 [INFO] CHASSIS_PID_TURN, Time: 5178, Actual_Vol1: -111.000000, Actual_Vol2: 450.000000, Actual_Vol3: -18.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34416.157872, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.940392, Absolute Angle: 1.726691, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27970 [INFO] CHASSIS_PID_TURN, Time: 5188, Actual_Vol1: -92.000000, Actual_Vol2: 450.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34326.792150, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.936572, Absolute Angle: 1.726625, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27970 [INFO] CHASSIS_PID_TURN, Time: 5198, Actual_Vol1: -92.000000, Actual_Vol2: 444.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34237.475382, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.931677, Absolute Angle: 1.726539, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27972 [INFO] CHASSIS_PID_TURN, Time: 5208, Actual_Vol1: -99.000000, Actual_Vol2: 456.000000, Actual_Vol3: 0.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34148.175501, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929988, Absolute Angle: 1.726510, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27972 [INFO] CHASSIS_PID_TURN, Time: 5218, Actual_Vol1: -92.000000, Actual_Vol2: 456.000000, Actual_Vol3: 0.000000, Actual_Vol4: 99.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 34058.895145, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928036, Absolute Angle: 1.726476, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016648, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27972 [INFO] CHASSIS_PID_TURN, Time: 5228, Actual_Vol1: -99.000000, Actual_Vol2: 450.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33969.610486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.928466, Absolute Angle: 1.726483, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015665, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27974 [INFO] CHASSIS_PID_TURN, Time: 5238, Actual_Vol1: -86.000000, Actual_Vol2: 444.000000, Actual_Vol3: 0.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33880.362357, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.924813, Absolute Angle: 1.726419, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018887, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27974 [INFO] CHASSIS_PID_TURN, Time: 5248, Actual_Vol1: -86.000000, Actual_Vol2: 493.000000, Actual_Vol3: 6.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33791.121646, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.924071, Absolute Angle: 1.726407, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27974 [INFO] CHASSIS_PID_TURN, Time: 5258, Actual_Vol1: -283.000000, Actual_Vol2: 530.000000, Actual_Vol3: -80.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33701.849316, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927233, Absolute Angle: 1.726462, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27976 [INFO] CHASSIS_PID_TURN, Time: 5268, Actual_Vol1: -363.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33612.548349, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.930097, Absolute Angle: 1.726512, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27976 [INFO] CHASSIS_PID_TURN, Time: 5278, Actual_Vol1: -382.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33523.227118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.932123, Absolute Angle: 1.726547, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27976 [INFO] CHASSIS_PID_TURN, Time: 5288, Actual_Vol1: -376.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33433.885374, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.934174, Absolute Angle: 1.726583, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -12.600000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27978 [INFO] CHASSIS_PID_TURN, Time: 5298, Actual_Vol1: -376.000000, Actual_Vol2: 548.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33344.551868, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.933351, Absolute Angle: 1.726568, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -6.800000, Actual_Vel2: -0.000000, Actual_Vel3: -20.200000, Actual_Vel4: 0.000000
27978 [INFO] CHASSIS_PID_TURN, Time: 5308, Actual_Vol1: -148.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33255.254778, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.929709, Absolute Angle: 1.726505, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27978 [INFO] CHASSIS_PID_TURN, Time: 5318, Actual_Vol1: -105.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33165.977957, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.927682, Absolute Angle: 1.726470, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27980 [INFO] CHASSIS_PID_TURN, Time: 5328, Actual_Vol1: -99.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 33076.723304, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.925465, Absolute Angle: 1.726431, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019629, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27980 [INFO] CHASSIS_PID_TURN, Time: 5338, Actual_Vol1: -92.000000, Actual_Vol2: 585.000000, Actual_Vol3: -99.000000, Actual_Vol4: 209.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32987.518750, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.920455, Absolute Angle: 1.726343, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023245, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
27980 [INFO] CHASSIS_PID_TURN, Time: 5348, Actual_Vol1: -92.000000, Actual_Vol2: 591.000000, Actual_Vol3: -92.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32898.318062, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.920069, Absolute Angle: 1.726337, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023632, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
27982 [INFO] CHASSIS_PID_TURN, Time: 5358, Actual_Vol1: -86.000000, Actual_Vol2: 554.000000, Actual_Vol3: -99.000000, Actual_Vol4: 117.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32809.181388, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.913667, Absolute Angle: 1.726225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030033, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
27982 [INFO] CHASSIS_PID_TURN, Time: 5368, Actual_Vol1: -92.000000, Actual_Vol2: 542.000000, Actual_Vol3: -86.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32720.065457, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.911593, Absolute Angle: 1.726189, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028799, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
27982 [INFO] CHASSIS_PID_TURN, Time: 5378, Actual_Vol1: -86.000000, Actual_Vol2: 542.000000, Actual_Vol3: -86.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32630.903980, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.916148, Absolute Angle: 1.726268, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024979, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
27984 [INFO] CHASSIS_PID_TURN, Time: 5388, Actual_Vol1: -86.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32541.745763, kD: 35.000000, kI: 0.0
```

```
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.915822, Absolute Angle: 1.726263, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022581, over s
lew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27984 [INFO] CHASSIS_PID_TURN, Time: 5398, Actual_Vol1: -99.000000, Actual_Vol2: 536.000000, Actual_Vol3: -160.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32452.632276, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.911349, Absolute Angle: 1.726184, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022826, over
 slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27984 [INFO] CHASSIS_PID_TURN, Time: 5408, Actual_Vol1: -290.000000, Actual_Vol2: 579.000000, Actual_Vol3: -216.000000, Actual_Vol4: 209.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32363.534246, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.909803, Absolute Angle: 1.726157, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, ov
er slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27986 [INFO] CHASSIS_PID_TURN, Time: 5418, Actual_Vol1: -326.000000, Actual_Vol2: 634.000000, Actual_Vol3: -240.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32274.386473, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.914777, Absolute Angle: 1.726244, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, ov
er slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27986 [INFO] CHASSIS_PID_TURN, Time: 5428, Actual_Vol1: -370.000000, Actual_Vol2: 641.000000, Actual_Vol3: -234.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32185.245290, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.914118, Absolute Angle: 1.726233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, ov
er slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27986 [INFO] CHASSIS_PID_TURN, Time: 5438, Actual_Vol1: -376.000000, Actual_Vol2: 610.000000, Actual_Vol3: -228.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32096.139350, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.910594, Absolute Angle: 1.726171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, ov
er slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27988 [INFO] CHASSIS_PID_TURN, Time: 5448, Actual_Vol1: -148.000000, Actual_Vol2: 561.000000, Actual_Vol3: -228.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 32007.018162, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.912119, Absolute Angle: 1.726198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024371, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27988 [INFO] CHASSIS_PID_TURN, Time: 5458, Actual_Vol1: -111.000000, Actual_Vol2: 548.000000, Actual_Vol3: -265.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31917.952951, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.906521, Absolute Angle: 1.726100, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027653, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27988 [INFO] CHASSIS_PID_TURN, Time: 5468, Actual_Vol1: -99.000000, Actual_Vol2: 598.000000, Actual_Vol3: -314.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31828.889803, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.906315, Absolute Angle: 1.726097, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027860, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.600000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27990 [INFO] CHASSIS_PID_TURN, Time: 5478, Actual_Vol1: -86.000000, Actual_Vol2: 628.000000, Actual_Vol3: -363.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31739.833474, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.905633, Absolute Angle: 1.726085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028542, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 37.000000
    27990 [INFO] CHASSIS_PID_TURN, Time: 5488, Actual_Vol1: -86.000000, Actual_Vol2: 647.000000, Actual_Vol3: -363.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31650.807550, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.902592, Absolute Angle: 1.726032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030758, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27990 [INFO] CHASSIS_PID_TURN, Time: 5498, Actual_Vol1: -92.000000, Actual_Vol2: 647.000000, Actual_Vol3: -363.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31561.790121, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.901743, Absolute Angle: 1.726017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027966, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27992 [INFO] CHASSIS_PID_TURN, Time: 5508, Actual_Vol1: -86.000000, Actual_Vol2: 647.000000, Actual_Vol3: -326.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31472.802727, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.898739, Absolute Angle: 1.725964, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028943, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27992 [INFO] CHASSIS_PID_TURN, Time: 5518, Actual_Vol1: -86.000000, Actual_Vol2: 684.000000, Actual_Vol3: -407.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31383.867080, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893565, Absolute Angle: 1.725874, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031901, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27992 [INFO] CHASSIS_PID_TURN, Time: 5528, Actual_Vol1: -283.000000, Actual_Vol2: 752.000000, Actual_Vol3: -450.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31294.936846, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893023, Absolute Angle: 1.725865, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027432, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    27994 [INFO] CHASSIS_PID_TURN, Time: 5538, Actual_Vol1: -314.000000, Actual_Vol2: 758.000000, Actual_Vol3: -456.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31205.942991, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899386, Absolute Angle: 1.725976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027045, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 0.000000
    27994 [INFO] CHASSIS_PID_TURN, Time: 5548, Actual_Vol1: -326.000000, Actual_Vol2: 770.000000, Actual_Vol3: -419.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31116.886267, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.905672, Absolute Angle: 1.726085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 0.000000
    27994 [INFO] CHASSIS_PID_TURN, Time: 5558, Actual_Vol1: -370.000000, Actual_Vol2: 770.000000, Actual_Vol3: -382.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31027.896701, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.898957, Absolute Angle: 1.725968, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, ov
er slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.200000, Actual_Vel3: -306.200000, Actual_Vel4: 28.800000
    27996 [INFO] CHASSIS_PID_TURN, Time: 5568, Actual_Vol1: -376.000000, Actual_Vol2: 819.000000, Actual_Vol3: -419.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30938.887355, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.900935, Absolute Angle: 1.726003, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023124, ov
er slew: 0, Actual_Vel1: -17.600000, Actual_Vel2: -0.000000, Actual_Vel3: -306.200000, Actual_Vel4: 28.800000
    27996 [INFO] CHASSIS_PID_TURN, Time: 5578, Actual_Vol1: -456.000000, Actual_Vol2: 856.000000, Actual_Vol3: -456.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30849.851418, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.903594, Absolute Angle: 1.726049, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022798, ov
er slew: 0, Actual_Vel1: -19.600000, Actual_Vel2: -0.000000, Actual_Vel3: -306.200000, Actual_Vel4: 14.800000
    27996 [INFO] CHASSIS_PID_TURN, Time: 5588, Actual_Vol1: -394.000000, Actual_Vol2: 862.000000, Actual_Vol3: -480.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30760.817773, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.903364, Absolute Angle: 1.726045, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, ov
er slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -19.400000, Actual_Vel4: 13.800000
    27998 [INFO] CHASSIS_PID_TURN, Time: 5598, Actual_Vol1: -148.000000, Actual_Vol2: 869.000000, Actual_Vol3: -456.000000, Actual_Vol4: 277.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30671.768218, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.904956, Absolute Angle: 1.726073, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, ov
er slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -19.400000, Actual_Vel4: 13.800000
    27998 [INFO] CHASSIS_PID_TURN, Time: 5608, Actual_Vol1: -111.000000, Actual_Vol2: 862.000000, Actual_Vol3: -456.000000, Actual_Vol4: 246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30582.772630, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899559, Absolute Angle: 1.725979, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021754, ov
er slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.800000
    27998 [INFO] CHASSIS_PID_TURN, Time: 5618, Actual_Vol1: -92.000000, Actual_Vol2: 942.000000, Actual_Vol3: -444.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30493.780574, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899206, Absolute Angle: 1.725973, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021095, ov
er slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.800000
    28000 [INFO] CHASSIS_PID_TURN, Time: 5628, Actual_Vol1: -290.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -320.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30404.784180, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.899639, Absolute Angle: 1.725980, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019095, o
ver slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000
    28000 [INFO] CHASSIS_PID_TURN, Time: 5638, Actual_Vol1: -314.000000, Actual_Vol2: 1066.000000, Actual_Vol3: -271.000000, Actual_Vol4: 271.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30315.813358, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.897082, Absolute Angle: 1.725935, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019095, o
ver slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: 11.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000
    28000 [INFO] CHASSIS_PID_TURN, Time: 5648, Actual_Vol1: -370.000000, Actual_Vol2: 961.000000, Actual_Vol3: -259.000000, Actual_Vol4: 253.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30226.877258, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.893610, Absolute Angle: 1.725875, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013498, ov
er slew: 0, Actual_Vel1: -15.000000, Actual_Vel2: 17.000000, Actual_Vel3: -14.200000, Actual_Vel4: 16.200000
    28002 [INFO] CHASSIS_PID_TURN, Time: 5658, Actual_Vol1: -154.000000, Actual_Vol2: 856.000000, Actual_Vol3: -283.000000, Actual_Vol4: 240.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30137.966537, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.891072, Absolute Angle: 1.725831, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015243, ov
er slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 16.600000, Actual_Vel3: -10.800000, Actual_Vel4: 16.200000
    28002 [INFO] CHASSIS_PID_TURN, Time: 5668, Actual_Vol1: -99.000000, Actual_Vol2: 807.000000, Actual_Vol3: -283.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 30049.050954, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.891558, Absolute Angle: 1.725839, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014600, ov
er slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000
    28002 [INFO] CHASSIS_PID_TURN, Time: 5678, Actual_Vol1: -99.000000, Actual_Vol2: 684.000000, Actual_Vol3: -246.000000, Actual_Vol4: 246.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29960.158863, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.889209, Absolute Angle: 1.725798, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016463, ov
er slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000
    28004 [INFO] CHASSIS_PID_TURN, Time: 5688, Actual_Vol1: -92.000000, Actual_Vol2: 641.000000, Actual_Vol3: -228.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29871.355420, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.880344, Absolute Angle: 1.725643, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025328, ov
er slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000
    28004 [INFO] CHASSIS_PID_TURN, Time: 5698, Actual_Vol1: -209.000000, Actual_Vol2: 708.000000, Actual_Vol3: -265.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29782.571045, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.878438, Absolute Angle: 1.725610, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027235, ov
er slew: 0, Actual_Vel1: -14.800000, Actual_Vel2: 19.200000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000
    28004 [INFO] CHASSIS_PID_TURN, Time: 5708, Actual_Vol1: -314.000000, Actual_Vol2: 721.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29693.834033, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.873701, Absolute Angle: 1.725527, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031971, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -14.800000, Actual_Vel4: 24.600000
    28006 [INFO] CHASSIS_PID_TURN, Time: 5718, Actual_Vol1: -363.000000, Actual_Vol2: 758.000000, Actual_Vol3: -326.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29605.189681, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.864435, Absolute Angle: 1.725366, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041237, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -21.000000, Actual_Vel4: 24.600000
    28006 [INFO] CHASSIS_PID_TURN, Time: 5728, Actual_Vol1: -370.000000, Actual_Vol2: 764.000000, Actual_Vol3: -326.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29516.626042, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.856364, Absolute Angle: 1.725225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049309, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 25.600000, Actual_Vel3: -21.000000, Actual_Vel4: 50.800000
    28006 [INFO] CHASSIS_PID_TURN, Time: 5738, Actual_Vol1: -370.000000, Actual_Vol2: 832.000000, Actual_Vol3: -400.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29428.119101, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.850694, Absolute Angle: 1.725126, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054978, ov
er slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 22.200000, Actual_Vel3: -21.000000, Actual_Vel4: 50.800000
    28008 [INFO] CHASSIS_PID_TURN, Time: 5748, Actual_Vol1: -216.000000, Actual_Vol2: 856.000000, Actual_Vol3: -437.000000, Actual_Vol4: 437.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29339.643864, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.847524, Absolute Angle: 1.725070, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 22.200000, Actual_Vel3: -43.600000, Actual_Vel4: 50.800000
    28008 [INFO] CHASSIS_PID_TURN, Time: 5758, Actual_Vol1: -123.000000, Actual_Vol2: 856.000000, Actual_Vol3: -400.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29251.127663, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851620, Absolute Angle: 1.725142, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 22.200000, Actual_Vel3: -40.400000, Actual_Vel4: 98.400000
```

```
28008 [INFO] CHASSIS_PID_TURN, Time: 5768, Actual_Vol1: -92.000000, Actual_Vol2: 856.000000, Actual_Vol3: -376.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29162.593383, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853428, Absolute Angle: 1.725174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, ov
er slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -40.400000, Actual_Vel4: 98.400000
28010 [INFO] CHASSIS_PID_TURN, Time: 5778, Actual_Vol1: -222.000000, Actual_Vol2: 1004.000000, Actual_Vol3: -431.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 29074.046506, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854688, Absolute Angle: 1.725196, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, o
ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -12.400000, Actual_Vel4: 98.400000
28010 [INFO] CHASSIS_PID_TURN, Time: 5788, Actual_Vol1: -308.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -444.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28985.430171, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.861633, Absolute Angle: 1.725317, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057432, o
ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 37.400000, Actual_Vel3: -14.600000, Actual_Vel4: 98.400000
28010 [INFO] CHASSIS_PID_TURN, Time: 5798, Actual_Vol1: -326.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -320.000000, Actual_Vol4: 561.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28896.820636, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860954, Absolute Angle: 1.725305, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: -0.000000, Actual_Vel3: -14.600000, Actual_Vel4: 98.400000
28012 [INFO] CHASSIS_PID_TURN, Time: 5808, Actual_Vol1: -326.000000, Actual_Vol2: 973.000000, Actual_Vol3: -259.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28808.202645, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.861799, Absolute Angle: 1.725320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, ov
er slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000
28012 [INFO] CHASSIS_PID_TURN, Time: 5818, Actual_Vol1: -413.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -283.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28719.579633, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.862301, Absolute Angle: 1.725328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052116, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000
28012 [INFO] CHASSIS_PID_TURN, Time: 5828, Actual_Vol1: -450.000000, Actual_Vol2: 1053.000000, Actual_Vol3: -326.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28630.910586, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.866905, Absolute Angle: 1.725409, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049558, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.600000, Actual_Vel4: 20.400000
28014 [INFO] CHASSIS_PID_TURN, Time: 5838, Actual_Vol1: -456.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -302.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28542.318690, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.859190, Absolute Angle: 1.725274, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.046086, o
ver slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.000000, Actual_Vel4: 13.800000
28014 [INFO] CHASSIS_PID_TURN, Time: 5848, Actual_Vol1: -413.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -302.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28453.734081, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.858461, Absolute Angle: 1.725261, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044035, o
ver slew: 0, Actual_Vel1: -16.800000, Actual_Vel2: 52.800000, Actual_Vel3: -14.000000, Actual_Vel4: 14.400000
28014 [INFO] CHASSIS_PID_TURN, Time: 5858, Actual_Vol1: -376.000000, Actual_Vol2: 1152.000000, Actual_Vol3: -302.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28365.193451, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854063, Absolute Angle: 1.725185, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044035, o
ver slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000
28016 [INFO] CHASSIS_PID_TURN, Time: 5868, Actual_Vol1: -370.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -265.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28276.625047, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.856840, Absolute Angle: 1.725233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041685, o
ver slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000
28016 [INFO] CHASSIS_PID_TURN, Time: 5878, Actual_Vol1: -333.000000, Actual_Vol2: 1183.000000, Actual_Vol3: -246.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28188.079172, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854588, Absolute Angle: 1.725194, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032821, o
ver slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 14.400000
28016 [INFO] CHASSIS_PID_TURN, Time: 5888, Actual_Vol1: -407.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -277.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28099.547665, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853151, Absolute Angle: 1.725169, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030914, o
ver slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 52.800000, Actual_Vel3: -11.600000, Actual_Vel4: 13.200000
28018 [INFO] CHASSIS_PID_TURN, Time: 5898, Actual_Vol1: -437.000000, Actual_Vol2: 1207.000000, Actual_Vol3: -314.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 28011.020204, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852746, Absolute Angle: 1.725162, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026177, o
ver slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 53.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
28018 [INFO] CHASSIS_PID_TURN, Time: 5908, Actual_Vol1: -450.000000, Actual_Vol2: 1195.000000, Actual_Vol3: -363.000000, Actual_Vol4: 370.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27922.498320, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852188, Absolute Angle: 1.725152, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, o
ver slew: 0, Actual_Vel1: -12.800000, Actual_Vel2: 53.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
28018 [INFO] CHASSIS_PID_TURN, Time: 5918, Actual_Vol1: -450.000000, Actual_Vol2: 1090.000000, Actual_Vol3: -425.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27833.979691, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851863, Absolute Angle: 1.725146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, o
ver slew: 0, Actual_Vel1: -12.800000, Actual_Vel2: 19.800000, Actual_Vel3: -11.200000, Actual_Vel4: 12.200000
28020 [INFO] CHASSIS_PID_TURN, Time: 5928, Actual_Vol1: -394.000000, Actual_Vol2: 1072.000000, Actual_Vol3: -431.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27745.435924, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854377, Absolute Angle: 1.725190, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, o
ver slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 19.800000, Actual_Vel3: -11.200000, Actual_Vel4: 12.200000
28020 [INFO] CHASSIS_PID_TURN, Time: 5938, Actual_Vol1: -320.000000, Actual_Vol2: 924.000000, Actual_Vol3: -320.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27656.902057, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.853387, Absolute Angle: 1.725173, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019381, ov
er slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000
28020 [INFO] CHASSIS_PID_TURN, Time: 5948, Actual_Vol1: -308.000000, Actual_Vol2: 807.000000, Actual_Vol3: -302.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27568.343562, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.855849, Absolute Angle: 1.725216, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015285, ov
er slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000
28022 [INFO] CHASSIS_PID_TURN, Time: 5958, Actual_Vol1: -394.000000, Actual_Vol2: 838.000000, Actual_Vol3: -363.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27479.759651, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.858391, Absolute Angle: 1.725260, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, ov
er slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -11.800000, Actual_Vel4: 12.200000
28022 [INFO] CHASSIS_PID_TURN, Time: 5968, Actual_Vol1: -431.000000, Actual_Vol2: 862.000000, Actual_Vol3: -290.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27391.215698, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854395, Absolute Angle: 1.725190, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, ov
er slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 16.600000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000
28022 [INFO] CHASSIS_PID_TURN, Time: 5978, Actual_Vol1: -431.000000, Actual_Vol2: 856.000000, Actual_Vol3: -246.000000, Actual_Vol4: 376.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27302.614759, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860094, Absolute Angle: 1.725290, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, ov
er slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000
28024 [INFO] CHASSIS_PID_TURN, Time: 5988, Actual_Vol1: -474.000000, Actual_Vol2: 1004.000000, Actual_Vol3: -283.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27214.009165, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.860559, Absolute Angle: 1.725298, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -17.400000, Actual_Vel4: 12.000000
28024 [INFO] CHASSIS_PID_TURN, Time: 5998, Actual_Vol1: -517.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -296.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27125.465832, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.854333, Absolute Angle: 1.725189, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 21.800000, Actual_Vel3: -22.600000, Actual_Vel4: 15.000000
28024 [INFO] CHASSIS_PID_TURN, Time: 6008, Actual_Vol1: -524.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -265.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 27036.943682, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.852215, Absolute Angle: 1.725152, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015042, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 15.000000
28026 [INFO] CHASSIS_PID_TURN, Time: 6018, Actual_Vol1: -579.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -283.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26948.429185, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.851450, Absolute Angle: 1.725139, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015455, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 11.800000
28026 [INFO] CHASSIS_PID_TURN, Time: 6028, Actual_Vol1: -628.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -320.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26859.965313, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.846387, Absolute Angle: 1.725051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014172, o
ver slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 18.600000, Actual_Vel3: -22.600000, Actual_Vel4: 11.800000
28026 [INFO] CHASSIS_PID_TURN, Time: 6038, Actual_Vol1: -647.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -314.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26771.509297, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.845602, Absolute Angle: 1.725037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014958, o
ver slew: 0, Actual_Vel1: -450.400000, Actual_Vel2: 28.000000, Actual_Vel3: -22.600000, Actual_Vel4: 14.000000
28028 [INFO] CHASSIS_PID_TURN, Time: 6048, Actual_Vol1: -598.000000, Actual_Vol2: 1244.000000, Actual_Vol3: -400.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26683.064576, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.844472, Absolute Angle: 1.725017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -55.400000, Actual_Vel2: 28.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000
28028 [INFO] CHASSIS_PID_TURN, Time: 6058, Actual_Vol1: -561.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -450.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26594.569134, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849544, Absolute Angle: 1.725106, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -55.400000, Actual_Vel2: 28.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000
28028 [INFO] CHASSIS_PID_TURN, Time: 6068, Actual_Vol1: -517.000000, Actual_Vol2: 1281.000000, Actual_Vol3: -487.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26506.069906, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849923, Absolute Angle: 1.725112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -21.200000, Actual_Vel2: -0.000000, Actual_Vel3: -42.400000, Actual_Vel4: 14.000000
28030 [INFO] CHASSIS_PID_TURN, Time: 6078, Actual_Vol1: -474.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -530.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26417.595323, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.847458, Absolute Angle: 1.725069, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 15.800000
28030 [INFO] CHASSIS_PID_TURN, Time: 6088, Actual_Vol1: -462.000000, Actual_Vol2: 1368.000000, Actual_Vol3: -505.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26329.113469, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.848185, Absolute Angle: 1.725082, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 15.800000
28030 [INFO] CHASSIS_PID_TURN, Time: 6098, Actual_Vol1: -499.000000, Actual_Vol2: 1368.000000, Actual_Vol3: -505.000000, Actual_Vol4: 283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26240.620321, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.849315, Absolute Angle: 1.725102, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000
28032 [INFO] CHASSIS_PID_TURN, Time: 6108, Actual_Vol1: -524.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -536.000000, Actual_Vol4: 283.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26152.155133, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.846519, Absolute Angle: 1.725053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016087, o
ver slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000
28032 [INFO] CHASSIS_PID_TURN, Time: 6118, Actual_Vol1: -530.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -548.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 26063.742958, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.841217, Absolute Angle: 1.724960, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019342, o
ver slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: -0.000000, Actual_Vel3: -238.400000, Actual_Vel4: 11.800000
28032 [INFO] CHASSIS_PID_TURN, Time: 6128, Actual_Vol1: -536.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -604.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25975.303031, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.843993, Absolute Angle: 1.725009, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019342, o
ver slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: -0.000000, Actual_Vel3: -21.400000, Actual_Vel4: 11.800000
28034 [INFO] CHASSIS_PID_TURN, Time: 6138, Actual_Vol1: -542.000000, Actual_Vol2: 1614.000000, Actual_Vol3: -616.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25886.906303, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.839673, Absolute Angle: 1.724933, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020887, o
```

```
ver slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 31.800000, Actual_Vel3: -21.400000, Actual_Vel4: 11.800000
28034 [INFO] CHASSIS_PID_TURN, Time: 6148, Actual_Vol1: -499.000000, Actual_Vol2: 1435.000000, Actual_Vol3: -542.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25798.502533, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.840377, Absolute Angle: 1.724946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020887, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 27.400000, Actual_Vel3: -17.800000, Actual_Vel4: 11.800000
28034 [INFO] CHASSIS_PID_TURN, Time: 6158, Actual_Vol1: -511.000000, Actual_Vol2: 1411.000000, Actual_Vol3: -487.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25710.116081, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.838645, Absolute Angle: 1.724916, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021914, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 31.400000, Actual_Vel3: -17.800000, Actual_Vel4: 14.200000
28036 [INFO] CHASSIS_PID_TURN, Time: 6168, Actual_Vol1: -530.000000, Actual_Vol2: 1331.000000, Actual_Vol3: -480.000000, Actual_Vol4: 542.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25621.753442, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.836264, Absolute Angle: 1.724874, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024296, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 29.800000, Actual_Vel3: -17.800000, Actual_Vel4: 14.200000
28036 [INFO] CHASSIS_PID_TURN, Time: 6178, Actual_Vol1: -585.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -505.000000, Actual_Vol4: 554.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25533.447218, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.830622, Absolute Angle: 1.724776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029937, over slew: 0, Actual_Vel1: -12.400000, Actual_Vel2: 29.800000, Actual_Vel3: -17.800000, Actual_Vel4: 10.800000
28036 [INFO] CHASSIS_PID_TURN, Time: 6188, Actual_Vol1: -622.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -505.000000, Actual_Vol4: 567.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25445.140879, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.830634, Absolute Angle: 1.724776, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023711, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 10.200000
28038 [INFO] CHASSIS_PID_TURN, Time: 6198, Actual_Vol1: -616.000000, Actual_Vol2: 1072.000000, Actual_Vol3: -456.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25356.865868, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.827501, Absolute Angle: 1.724721, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024714, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 10.200000
28038 [INFO] CHASSIS_PID_TURN, Time: 6208, Actual_Vol1: -665.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -474.000000, Actual_Vol4: 487.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25268.601636, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.826423, Absolute Angle: 1.724702, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025026, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 21.600000, Actual_Vel3: -10.800000, Actual_Vel4: 11.000000
28038 [INFO] CHASSIS_PID_TURN, Time: 6218, Actual_Vol1: -653.000000, Actual_Vol2: 1177.000000, Actual_Vol3: -517.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25180.345854, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.825578, Absolute Angle: 1.724687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024345, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.800000, Actual_Vel4: 12.600000
28040 [INFO] CHASSIS_PID_TURN, Time: 6228, Actual_Vol1: -665.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -567.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25092.212236, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.813362, Absolute Angle: 1.724474, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.036561, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.600000, Actual_Vel4: 12.600000
28040 [INFO] CHASSIS_PID_TURN, Time: 6238, Actual_Vol1: -708.000000, Actual_Vol2: 1244.000000, Actual_Vol3: -616.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 25004.088807, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.812343, Absolute Angle: 1.724456, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.037580, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 18.000000, Actual_Vel3: -10.600000, Actual_Vel4: 13.200000
28040 [INFO] CHASSIS_PID_TURN, Time: 6248, Actual_Vol1: -702.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -622.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24916.008324, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.808048, Absolute Angle: 1.724382, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041875, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -10.600000, Actual_Vel4: 13.200000
28042 [INFO] CHASSIS_PID_TURN, Time: 6258, Actual_Vol1: -838.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -659.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24827.967721, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.804060, Absolute Angle: 1.724312, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045862, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -9.600000, Actual_Vel4: 11.400000
28042 [INFO] CHASSIS_PID_TURN, Time: 6268, Actual_Vol1: -912.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -708.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24739.990815, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.797691, Absolute Angle: 1.724201, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051624, over slew: 0, Actual_Vel1: -10.200000, Actual_Vel2: 16.800000, Actual_Vel3: -9.600000, Actual_Vel4: 11.400000
28042 [INFO] CHASSIS_PID_TURN, Time: 6278, Actual_Vol1: -936.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -708.000000, Actual_Vol4: 388.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24652.064836, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.792598, Absolute Angle: 1.724112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056717, over slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 11.400000
28044 [INFO] CHASSIS_PID_TURN, Time: 6288, Actual_Vol1: -936.000000, Actual_Vol2: 1435.000000, Actual_Vol3: -721.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24564.204620, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.786022, Absolute Angle: 1.723997, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063293, over slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
28044 [INFO] CHASSIS_PID_TURN, Time: 6298, Actual_Vol1: -936.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -708.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24476.378882, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782574, Absolute Angle: 1.723937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063945, over slew: 0, Actual_Vel1: -8.000000, Actual_Vel2: 10.200000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
28044 [INFO] CHASSIS_PID_TURN, Time: 6308, Actual_Vol1: -936.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -788.000000, Actual_Vol4: 407.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24388.542597, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.783629, Absolute Angle: 1.723955, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.061419, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: -0.000000, Actual_Vel3: -8.400000, Actual_Vel4: 9.600000
28046 [INFO] CHASSIS_PID_TURN, Time: 6318, Actual_Vol1: -832.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -875.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24300.697033, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.784556, Absolute Angle: 1.723972, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.061419, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 9.600000
28046 [INFO] CHASSIS_PID_TURN, Time: 6328, Actual_Vol1: -862.000000, Actual_Vol2: 1602.000000, Actual_Vol3: -967.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24212.869739, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782729, Absolute Angle: 1.723940, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057803, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 9.600000
28046 [INFO] CHASSIS_PID_TURN, Time: 6338, Actual_Vol1: -912.000000, Actual_Vol2: 1675.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24125.041576, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.782816, Absolute Angle: 1.723941, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057803, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -11.200000, Actual_Vel4: 10.600000
28048 [INFO] CHASSIS_PID_TURN, Time: 6348, Actual_Vol1: -924.000000, Actual_Vol2: 1694.000000, Actual_Vol3: -992.000000, Actual_Vol4: 511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 24037.247026, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.779455, Absolute Angle: 1.723882, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.059190, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 10.600000
28048 [INFO] CHASSIS_PID_TURN, Time: 6358, Actual_Vol1: -986.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -992.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23949.478992, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.776803, Absolute Angle: 1.723836, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.059460, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 10.600000
28048 [INFO] CHASSIS_PID_TURN, Time: 6368, Actual_Vol1: -1029.000000, Actual_Vol2: 1793.000000, Actual_Vol3: -998.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23861.754896, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.772410, Absolute Angle: 1.723760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.058224, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 9.800000
28050 [INFO] CHASSIS_PID_TURN, Time: 6378, Actual_Vol1: -1078.000000, Actual_Vol2: 1786.000000, Actual_Vol3: -986.000000, Actual_Vol4: 474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23774.139047, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.761585, Absolute Angle: 1.723571, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069049, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
28050 [INFO] CHASSIS_PID_TURN, Time: 6388, Actual_Vol1: -1066.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23686.636837, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.750221, Absolute Angle: 1.723372, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.077280, over slew: 0, Actual_Vel1: -21.800000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
28050 [INFO] CHASSIS_PID_TURN, Time: 6398, Actual_Vol1: -986.000000, Actual_Vol2: 1885.000000, Actual_Vol3: -1072.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23599.252516, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.738432, Absolute Angle: 1.723166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.087991, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.600000, Actual_Vel4: 12.600000
28052 [INFO] CHASSIS_PID_TURN, Time: 6408, Actual_Vol1: -986.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 511.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23511.961261, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.729126, Absolute Angle: 1.723004, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.096453, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.000000, Actual_Vel4: 10.400000
28052 [INFO] CHASSIS_PID_TURN, Time: 6418, Actual_Vol1: -1016.000000, Actual_Vol2: 2014.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23424.708154, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.725311, Absolute Angle: 1.722937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.088051, over slew: 0, Actual_Vel1: -16.400000, Actual_Vel2: -0.000000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
28052 [INFO] CHASSIS_PID_TURN, Time: 6428, Actual_Vol1: -1072.000000, Actual_Vol2: 2168.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 474.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23337.549230, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.715892, Absolute Angle: 1.722773, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.096451, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 8.000000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
28054 [INFO] CHASSIS_PID_TURN, Time: 6438, Actual_Vol1: -1146.000000, Actual_Vol2: 2076.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23250.513279, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.703595, Absolute Angle: 1.722558, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.104453, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
28054 [INFO] CHASSIS_PID_TURN, Time: 6448, Actual_Vol1: -1201.000000, Actual_Vol2: 1977.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 493.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23163.594978, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.691830, Absolute Angle: 1.722353, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.112230, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 9.800000
28054 [INFO] CHASSIS_PID_TURN, Time: 6458, Actual_Vol1: -1244.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 23076.780670, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.681431, Absolute Angle: 1.722172, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.116260, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -9.000000, Actual_Vel4: 12.000000
28056 [INFO] CHASSIS_PID_TURN, Time: 6468, Actual_Vol1: -1300.000000, Actual_Vol2: 2008.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22990.091853, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.668882, Absolute Angle: 1.721953, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.123716, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 15.400000, Actual_Vel3: -258.200000, Actual_Vel4: 12.000000
28056 [INFO] CHASSIS_PID_TURN, Time: 6478, Actual_Vol1: -1331.000000, Actual_Vol2: 2020.000000, Actual_Vol3: -1398.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22903.544889, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.654696, Absolute Angle: 1.721705, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.131325, over slew: 0, Actual_Vel1: -15.600000, Actual_Vel2: 16.200000, Actual_Vel3: -258.200000, Actual_Vel4: 16.400000
28056 [INFO] CHASSIS_PID_TURN, Time: 6488, Actual_Vol1: -1392.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1368.000000, Actual_Vol4: 462.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22817.177327, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.636756, Absolute Angle: 1.721392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.147800, over slew: 0, Actual_Vel1: -21.400000, Actual_Vel2: 16.200000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
28058 [INFO] CHASSIS_PID_TURN, Time: 6498, Actual_Vol1: -1435.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1374.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22731.025856, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -168.000000, Heading_Sp: 90.000000, Relative_Heading: 98.615147, Absolute Angle: 1.721015, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.169409, over slew: 0, Actual_Vel1: -21.800000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
28058 [INFO] CHASSIS_PID_TURN, Time: 6508, Actual_Vol1: -1349.000000, Actual_Vol2: 1940.000000, Actual_Vol3: -1368.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22645.041083, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.598477, Absolute Angle: 1.720724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.186079, over slew: 0, Actual_Vel1: -27.600000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.400000
28058 [INFO] CHASSIS_PID_TURN, Time: 6518, Actual_Vol1: -1324.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1411.000000, Actual_Vol4: 579.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22560.175881, kD: 35.000000
```

```
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.486520, Absolute Angle: 1.718770, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.296296
, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000
    477:   28060 [INFO] CHASSIS_PID_TURN, Time: 6528, Actual_Vol1: -1294.000000, Actual_Vol2: 2014.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22475.498954, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.467693, Absolute Angle: 1.718441, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.315124
, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000
    478:   28060 [INFO] CHASSIS_PID_TURN, Time: 6538, Actual_Vol1: -1318.000000, Actual_Vol2: 2144.000000, Actual_Vol3: -1552.000000, Actual_Vol4: 671.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22390.979511, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.451944, Absolute Angle: 1.718166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.327511
, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 16.400000, Actual_Vel3: -24.600000, Actual_Vel4: 16.800000
    479:   28060 [INFO] CHASSIS_PID_TURN, Time: 6548, Actual_Vol1: -1337.000000, Actual_Vol2: 2193.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22306.596291, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.438322, Absolute Angle: 1.717929, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.338482
, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 15.600000, Actual_Vel3: -24.600000, Actual_Vel4: 19.200000
    480:   28062 [INFO] CHASSIS_PID_TURN, Time: 6558, Actual_Vol1: -1417.000000, Actual_Vol2: 2359.000000, Actual_Vol3: -1669.000000, Actual_Vol4: 678.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22222.315306, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.428099, Absolute Angle: 1.717750, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.344311
, over slew: 0, Actual_Vel1: -18.400000, Actual_Vel2: 15.600000, Actual_Vel3: -24.600000, Actual_Vel4: 19.200000
    481:   28062 [INFO] CHASSIS_PID_TURN, Time: 6568, Actual_Vol1: -1429.000000, Actual_Vol2: 2396.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22138.147880, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.416743, Absolute Angle: 1.717552, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.344842
, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 15.600000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000
    482:   28062 [INFO] CHASSIS_PID_TURN, Time: 6578, Actual_Vol1: -1361.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 22054.139384, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.400850, Absolute Angle: 1.717275, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.349371
, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 11.600000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000
    483:   28064 [INFO] CHASSIS_PID_TURN, Time: 6588, Actual_Vol1: -1417.000000, Actual_Vol2: 2458.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21970.283071, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.385631, Absolute Angle: 1.717009, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.352801
, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 10.400000, Actual_Vel3: -16.000000, Actual_Vel4: 15.400000
    484:   28064 [INFO] CHASSIS_PID_TURN, Time: 6598, Actual_Vol1: -1435.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -1712.000000, Actual_Vol4: 825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21886.548662, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.373441, Absolute Angle: 1.716796, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.355685
, over slew: 0, Actual_Vel1: -14.600000, Actual_Vel2: 10.400000, Actual_Vel3: -10.600000, Actual_Vel4: 15.400000
    485:   28064 [INFO] CHASSIS_PID_TURN, Time: 6608, Actual_Vol1: -1558.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -1706.000000, Actual_Vol4: 881.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21803.005613, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.354305, Absolute Angle: 1.716462, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.371006
, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -10.600000, Actual_Vel4: 15.400000
    486:   28066 [INFO] CHASSIS_PID_TURN, Time: 6618, Actual_Vol1: -1571.000000, Actual_Vol2: 2519.000000, Actual_Vol3: -1688.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21719.668685, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.333693, Absolute Angle: 1.716102, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.382200
, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 15.400000
    487:   28066 [INFO] CHASSIS_PID_TURN, Time: 6628, Actual_Vol1: -1651.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1688.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21636.589191, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.307949, Absolute Angle: 1.715653, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.395646
, over slew: 0, Actual_Vel1: -13.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 15.400000
    488:   28066 [INFO] CHASSIS_PID_TURN, Time: 6638, Actual_Vol1: -1669.000000, Actual_Vol2: 2519.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21553.728296, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.286090, Absolute Angle: 1.715272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.405741
, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 14.800000
    489:   28068 [INFO] CHASSIS_PID_TURN, Time: 6648, Actual_Vol1: -1731.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21471.152349, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.257595, Absolute Angle: 1.714774, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.423836
, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.400000, Actual_Vel4: 14.800000
    490:   28068 [INFO] CHASSIS_PID_TURN, Time: 6658, Actual_Vol1: -1749.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21388.770019, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.238233, Absolute Angle: 1.714436, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.430649
, over slew: 0, Actual_Vel1: -9.800000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 14.800000
    491:   28068 [INFO] CHASSIS_PID_TURN, Time: 6668, Actual_Vol1: -1756.000000, Actual_Vol2: 2599.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21306.580032, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.218999, Absolute Angle: 1.714101, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.43569
8, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 14.800000
    492:   28070 [INFO] CHASSIS_PID_TURN, Time: 6678, Actual_Vol1: -1817.000000, Actual_Vol2: 2655.000000, Actual_Vol3: -1823.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21224.569612, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.201042, Absolute Angle: 1.713787, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.43571
4, over slew: 0, Actual_Vel1: -15.400000, Actual_Vel2: 11.200000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000
    493:   28070 [INFO] CHASSIS_PID_TURN, Time: 6688, Actual_Vol1: -1793.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21142.721531, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.184808, Absolute Angle: 1.713504, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.43033
9, over slew: 0, Actual_Vel1: -15.400000, Actual_Vel2: 10.600000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000
    494:   28070 [INFO] CHASSIS_PID_TURN, Time: 6698, Actual_Vol1: -1768.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -1953.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 21061.113085, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.160845, Absolute Angle: 1.713086, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.43763
3, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -11.600000, Actual_Vel4: 28.400000
    495:   28072 [INFO] CHASSIS_PID_TURN, Time: 6708, Actual_Vol1: -1768.000000, Actual_Vol2: 2618.000000, Actual_Vol3: -1953.000000, Actual_Vol4: 1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20979.799318, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.131377, Absolute Angle: 1.712571, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.35514
3, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -15.400000, Actual_Vel4: 28.400000
    496:   28072 [INFO] CHASSIS_PID_TURN, Time: 6718, Actual_Vol1: -1848.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1959.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20898.764056, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.103526, Absolute Angle: 1.712085, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.36416
7, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 11.600000, Actual_Vel3: -15.400000, Actual_Vel4: 28.400000
    497:   28072 [INFO] CHASSIS_PID_TURN, Time: 6728, Actual_Vol1: -1860.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20818.037827, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.072623, Absolute Angle: 1.711546, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.37932
1, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 13.800000, Actual_Vel3: -15.400000, Actual_Vel4: 20.200000
    498:   28074 [INFO] CHASSIS_PID_TURN, Time: 6738, Actual_Vol1: -1873.000000, Actual_Vol2: 2766.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20737.683655, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 98.035417, Absolute Angle: 1.710897, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.40290
5, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 13.800000, Actual_Vel3: -16.600000, Actual_Vel4: 20.200000
    499:   28074 [INFO] CHASSIS_PID_TURN, Time: 6748, Actual_Vol1: -1965.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20657.687055, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.999660, Absolute Angle: 1.710272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.42843
9, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 13.800000, Actual_Vel3: -17.200000, Actual_Vel4: 20.200000
    500:   28074 [INFO] CHASSIS_PID_TURN, Time: 6758, Actual_Vol1: -1873.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20578.007844, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.967921, Absolute Angle: 1.709719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.44882
1, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 13.800000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000
    501:   28076 [INFO] CHASSIS_PID_TURN, Time: 6768, Actual_Vol1: -1965.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20498.584374, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.942347, Absolute Angle: 1.709272, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.45850
3, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000
    502:   28076 [INFO] CHASSIS_PID_TURN, Time: 6778, Actual_Vol1: -1990.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20419.341475, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.924290, Absolute Angle: 1.708957, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46134
1, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000
    503:   28076 [INFO] CHASSIS_PID_TURN, Time: 6788, Actual_Vol1: -2131.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2119.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20340.304506, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.903697, Absolute Angle: 1.708598, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46974
4, over slew: 0, Actual_Vel1: -9.600000, Actual_Vel2: 18.600000, Actual_Vel3: -17.200000, Actual_Vel4: 21.200000
    504:   28078 [INFO] CHASSIS_PID_TURN, Time: 6798, Actual_Vol1: -2094.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2137.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20261.400795, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.890371, Absolute Angle: 1.708365, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46393
4, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000
    505:   28078 [INFO] CHASSIS_PID_TURN, Time: 6808, Actual_Vol1: -2008.000000, Actual_Vol2: 2834.000000, Actual_Vol3: -2020.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20182.675766, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.872503, Absolute Angle: 1.708053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46119
0, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000
    506:   28078 [INFO] CHASSIS_PID_TURN, Time: 6818, Actual_Vol1: -2205.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -2119.000000, Actual_Vol4: 1287.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20104.264479, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.841129, Absolute Angle: 1.707506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46682
1, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.600000, Actual_Vel3: -13.600000, Actual_Vel4: 20.200000
    507:   28080 [INFO] CHASSIS_PID_TURN, Time: 6828, Actual_Vol1: -2236.000000, Actual_Vol2: 2852.000000, Actual_Vol3: -2261.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 20026.145173, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.811931, Absolute Angle: 1.706996, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.47415
9, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 9.600000, Actual_Vel3: -20.200000, Actual_Vel4: 20.200000
    508:   28080 [INFO] CHASSIS_PID_TURN, Time: 6838, Actual_Vol1: -2255.000000, Actual_Vol2: 2914.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19948.392846, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.775233, Absolute Angle: 1.706355, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.48236
2, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 9.600000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000
    509:   28080 [INFO] CHASSIS_PID_TURN, Time: 6848, Actual_Vol1: -2248.000000, Actual_Vol2: 2871.000000, Actual_Vol3: -2285.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19871.019388, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.737346, Absolute Angle: 1.705694, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.50088
7, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000
    510:   28082 [INFO] CHASSIS_PID_TURN, Time: 6858, Actual_Vol1: -2119.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19794.025191, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.699420, Absolute Angle: 1.705032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.51957
9, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -20.200000, Actual_Vel4: 27.800000
    511:   28082 [INFO] CHASSIS_PID_TURN, Time: 6868, Actual_Vol1: -2236.000000, Actual_Vol2: 2901.000000, Actual_Vol3: -2359.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19717.425872, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.659932, Absolute Angle: 1.704343, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.54111
0, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -44.600000, Actual_Vel4: 27.800000
    512:   28082 [INFO] CHASSIS_PID_TURN, Time: 6878, Actual_Vol1: -2273.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2168.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19641.175841, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.625003, Absolute Angle: 1.703733, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.55980
5, over slew: 0, Actual_Vel1: -20.400000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 24.200000
    513:   28084 [INFO] CHASSIS_PID_TURN, Time: 6888, Actual_Vol1: -2353.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2156.000000, Actual_Vol4: 1398.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19565.258232, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.591761, Absolute Angle: 1.703153, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.56908
4, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 24.200000
```

```
28084 [INFO] CHASSIS_PID_TURN, Time: 6898, Actual_Vol1: -2113.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -2248.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19489.750380, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.550785, Absolute Angle: 1.702438, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.58059
1, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -44.600000, Actual_Vel4: 21.000000
28084 [INFO] CHASSIS_PID_TURN, Time: 6908, Actual_Vol1: -2094.000000, Actual_Vol2: 3000.000000, Actual_Vol3: -2279.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19414.628522, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.512186, Absolute Angle: 1.701764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.59134
0, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000
28086 [INFO] CHASSIS_PID_TURN, Time: 6918, Actual_Vol1: -2211.000000, Actual_Vol2: 3135.000000, Actual_Vol3: -2156.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19339.893009, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.473551, Absolute Angle: 1.701090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.59907
2, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 12.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000
28086 [INFO] CHASSIS_PID_TURN, Time: 6928, Actual_Vol1: -2255.000000, Actual_Vol2: 3172.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19265.584254, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.430875, Absolute Angle: 1.700345, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.60454
2, over slew: 0, Actual_Vel1: -20.200000, Actual_Vel2: 15.000000, Actual_Vel3: -36.200000, Actual_Vel4: 21.000000
28086 [INFO] CHASSIS_PID_TURN, Time: 6938, Actual_Vol1: -2353.000000, Actual_Vol2: 3092.000000, Actual_Vol3: -2211.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19191.800149, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.378411, Absolute Angle: 1.699430, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.62124
9, over slew: 0, Actual_Vel1: -42.400000, Actual_Vel2: 20.800000, Actual_Vel3: -17.000000, Actual_Vel4: 27.400000
28088 [INFO] CHASSIS_PID_TURN, Time: 6948, Actual_Vol1: -2255.000000, Actual_Vol2: 2932.000000, Actual_Vol3: -2051.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19118.534870, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.326528, Absolute Angle: 1.698524, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.64139
3, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000
28088 [INFO] CHASSIS_PID_TURN, Time: 6958, Actual_Vol1: -2107.000000, Actual_Vol2: 2994.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 19045.788150, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.274672, Absolute Angle: 1.697619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.66767
5, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000
28088 [INFO] CHASSIS_PID_TURN, Time: 6968, Actual_Vol1: -2230.000000, Actual_Vol2: 3154.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18973.535126, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.225302, Absolute Angle: 1.696757, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.69898
8, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 20.800000, Actual_Vel3: -15.400000, Actual_Vel4: 23.000000
28090 [INFO] CHASSIS_PID_TURN, Time: 6978, Actual_Vol1: -2255.000000, Actual_Vol2: 3105.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1392.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18901.800732, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.173439, Absolute Angle: 1.695852, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.73025
8, over slew: 0, Actual_Vel1: -29.800000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.000000
28090 [INFO] CHASSIS_PID_TURN, Time: 6988, Actual_Vol1: -2267.000000, Actual_Vol2: 2938.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18830.648053, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.115268, Absolute Angle: 1.694837, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.77510
3, over slew: 0, Actual_Vel1: -27.000000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.800000
28090 [INFO] CHASSIS_PID_TURN, Time: 6998, Actual_Vol1: -2255.000000, Actual_Vol2: 2920.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1404.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18760.018225, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.062983, Absolute Angle: 1.693924, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.80952
0, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -10.400000, Actual_Vel4: 23.800000
28092 [INFO] CHASSIS_PID_TURN, Time: 7008, Actual_Vol1: -2107.000000, Actual_Vol2: 2932.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1355.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18689.854057, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 97.016417, Absolute Angle: 1.693112, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.82471
2, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -8.800000, Actual_Vel4: 18.600000
28092 [INFO] CHASSIS_PID_TURN, Time: 7018, Actual_Vol1: -2082.000000, Actual_Vol2: 3000.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1349.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18620.222499, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.963156, Absolute Angle: 1.692182, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.84877
5, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 22.800000, Actual_Vel3: -11.400000, Actual_Vel4: 18.600000
28092 [INFO] CHASSIS_PID_TURN, Time: 7028, Actual_Vol1: -2230.000000, Actual_Vol2: 3160.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18551.055688, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.916681, Absolute Angle: 1.691371, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.85855
2, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 23.200000, Actual_Vel3: -11.400000, Actual_Vel4: 12.400000
28094 [INFO] CHASSIS_PID_TURN, Time: 7038, Actual_Vol1: -2248.000000, Actual_Vol2: 3043.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18482.350623, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.870507, Absolute Angle: 1.690565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.86683
9, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 28.800000, Actual_Vel3: -9.800000, Actual_Vel4: 12.400000
28094 [INFO] CHASSIS_PID_TURN, Time: 7048, Actual_Vol1: -2236.000000, Actual_Vol2: 2901.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18414.125482, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.822514, Absolute Angle: 1.689727, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.87690
6, over slew: 0, Actual_Vel1: -25.600000, Actual_Vel2: 28.800000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000
28094 [INFO] CHASSIS_PID_TURN, Time: 7058, Actual_Vol1: -2347.000000, Actual_Vol2: 2907.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1306.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18346.385799, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.773968, Absolute Angle: 1.688880, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.88596
4, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 28.800000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000
28096 [INFO] CHASSIS_PID_TURN, Time: 7068, Actual_Vol1: -2255.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18279.163702, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.722210, Absolute Angle: 1.687977, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.90279
3, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 42.200000, Actual_Vel3: -13.000000, Actual_Vel4: 10.800000
28096 [INFO] CHASSIS_PID_TURN, Time: 7078, Actual_Vol1: -2119.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18212.465678, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.669802, Absolute Angle: 1.687062, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.92195
9, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 31.800000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
28096 [INFO] CHASSIS_PID_TURN, Time: 7088, Actual_Vol1: -2230.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1263.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18146.374480, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.609120, Absolute Angle: 1.686003, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.94166
5, over slew: 0, Actual_Vel1: -30.000000, Actual_Vel2: 31.800000, Actual_Vel3: -15.400000, Actual_Vel4: 9.400000
28098 [INFO] CHASSIS_PID_TURN, Time: 7098, Actual_Vol1: -2261.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18080.880736, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.549374, Absolute Angle: 1.684960, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.96281
1, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 31.800000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000
28098 [INFO] CHASSIS_PID_TURN, Time: 7108, Actual_Vol1: -2242.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -1940.000000, Actual_Vol4: 1214.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18015.977599, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.490314, Absolute Angle: 1.683929, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.98323
8, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000
28098 [INFO] CHASSIS_PID_TURN, Time: 7118, Actual_Vol1: -2113.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1885.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17951.665703, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.431190, Absolute Angle: 1.682898, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.99968
6, over slew: 0, Actual_Vel1: -31.800000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 8.800000
28100 [INFO] CHASSIS_PID_TURN, Time: 7128, Actual_Vol1: -2230.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1220.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17887.961687, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.370402, Absolute Angle: 1.681837, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.00800
9, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 38.400000, Actual_Vel3: -16.200000, Actual_Vel4: 11.600000
28100 [INFO] CHASSIS_PID_TURN, Time: 7138, Actual_Vol1: -2113.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1207.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17824.790376, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.317131, Absolute Angle: 1.680907, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.00939
7, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 21.600000, Actual_Vel3: -21.600000, Actual_Vel4: 11.600000
28100 [INFO] CHASSIS_PID_TURN, Time: 7148, Actual_Vol1: -2082.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1201.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17762.115028, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.267535, Absolute Angle: 1.680041, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.00713
7, over slew: 0, Actual_Vel1: -19.000000, Actual_Vel2: 21.600000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000
28102 [INFO] CHASSIS_PID_TURN, Time: 7158, Actual_Vol1: -2082.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1805.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17699.898180, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.221685, Absolute Angle: 1.679241, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.00361
8, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 14.200000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000
28102 [INFO] CHASSIS_PID_TURN, Time: 7168, Actual_Vol1: -2070.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17638.145174, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.175301, Absolute Angle: 1.678431, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.99813
9, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 14.200000, Actual_Vel3: -22.000000, Actual_Vel4: 12.400000
28102 [INFO] CHASSIS_PID_TURN, Time: 7178, Actual_Vol1: -2076.000000, Actual_Vol2: 2827.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17576.895138, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.125004, Absolute Angle: 1.677554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.99026
4, over slew: 0, Actual_Vel1: -17.200000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 19.000000
28104 [INFO] CHASSIS_PID_TURN, Time: 7188, Actual_Vol1: -2057.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17516.165089, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.073005, Absolute Angle: 1.676646, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.98997
8, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000
28104 [INFO] CHASSIS_PID_TURN, Time: 7198, Actual_Vol1: -1922.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17455.895975, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 96.026911, Absolute Angle: 1.675842, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.98950
5, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 10.800000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000
28104 [INFO] CHASSIS_PID_TURN, Time: 7208, Actual_Vol1: -1971.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17396.082127, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.981385, Absolute Angle: 1.675047, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.98177
1, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 8.000000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000
28106 [INFO] CHASSIS_PID_TURN, Time: 7218, Actual_Vol1: -2057.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17336.642565, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.943956, Absolute Angle: 1.674394, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.97272
5, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 9.000000, Actual_Vel3: -22.000000, Actual_Vel4: 23.200000
28106 [INFO] CHASSIS_PID_TURN, Time: 7228, Actual_Vol1: -1996.000000, Actual_Vol2: 2630.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17277.602454, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.904011, Absolute Angle: 1.673697, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.96649
5, over slew: 0, Actual_Vel1: -13.400000, Actual_Vel2: 9.000000, Actual_Vel3: -22.000000, Actual_Vel4: 19.800000
28106 [INFO] CHASSIS_PID_TURN, Time: 7238, Actual_Vol1: -1990.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17218.940171, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.866228, Absolute Angle: 1.673037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.95628
6, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.800000
28108 [INFO] CHASSIS_PID_TURN, Time: 7248, Actual_Vol1: -2002.000000, Actual_Vol2: 2532.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 992.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17160.582078, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.835809, Absolute Angle: 1.672506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.938159
, over slew: 0, Actual_Vel1: -11.600000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000
28108 [INFO] CHASSIS_PID_TURN, Time: 7258, Actual_Vol1: -2205.000000, Actual_Vol2: 2599.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17102.514432, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.806765, Absolute Angle: 1.671999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.915445
, over slew: 0, Actual_Vel1: -10.800000, Actual_Vel2: 9.000000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000
28108 [INFO] CHASSIS_PID_TURN, Time: 7268, Actual_Vol1: -2088.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 17044.654985, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.785945, Absolute Angle: 1.671636, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.883858
, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -16.200000, Actual_Vel4: 19.400000
    552:   28110 [INFO] CHASSIS_PID_TURN, Time: 7278, Actual_Vol1: -1879.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16987.074088, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.758090, Absolute Angle: 1.671150, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.851030
, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -18.200000, Actual_Vel4: 19.400000
    553:   28110 [INFO] CHASSIS_PID_TURN, Time: 7288, Actual_Vol1: -1866.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1016.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16929.739933, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.733416, Absolute Angle: 1.670719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.81595
9, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 8.800000, Actual_Vel3: -18.200000, Actual_Vel4: 19.400000
    554:   28110 [INFO] CHASSIS_PID_TURN, Time: 7298, Actual_Vol1: -1860.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16872.628174, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.711176, Absolute Angle: 1.670331, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.77913
8, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 14.800000
    555:   28112 [INFO] CHASSIS_PID_TURN, Time: 7308, Actual_Vol1: -1971.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 979.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16815.792085, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.683609, Absolute Angle: 1.669850, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.747581
, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
    556:   28112 [INFO] CHASSIS_PID_TURN, Time: 7318, Actual_Vol1: -2051.000000, Actual_Vol2: 2747.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16759.239316, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.655277, Absolute Angle: 1.669355, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.715125
, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 24.600000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
    557:   28112 [INFO] CHASSIS_PID_TURN, Time: 7328, Actual_Vol1: -2070.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16702.943962, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.629535, Absolute Angle: 1.668906, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.687596
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -18.200000, Actual_Vel4: 17.800000
    558:   28114 [INFO] CHASSIS_PID_TURN, Time: 7338, Actual_Vol1: -2076.000000, Actual_Vol2: 2618.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16646.846913, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.609705, Absolute Angle: 1.668560, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.657830
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
    559:   28114 [INFO] CHASSIS_PID_TURN, Time: 7348, Actual_Vol1: -2082.000000, Actual_Vol2: 2532.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16591.008960, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.583795, Absolute Angle: 1.668108, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.637890
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
    560:   28114 [INFO] CHASSIS_PID_TURN, Time: 7358, Actual_Vol1: -2224.000000, Actual_Vol2: 2593.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16535.441837, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.556712, Absolute Angle: 1.667635, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.618588
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 54.200000, Actual_Vel3: -31.600000, Actual_Vel4: 21.800000
    561:   28116 [INFO] CHASSIS_PID_TURN, Time: 7368, Actual_Vol1: -2242.000000, Actual_Vol2: 2667.000000, Actual_Vol3: -1842.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16480.104536, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.533730, Absolute Angle: 1.667234, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.591274
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 21.800000
    562:   28116 [INFO] CHASSIS_PID_TURN, Time: 7378, Actual_Vol1: -1959.000000, Actual_Vol2: 2717.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16425.034990, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.506955, Absolute Angle: 1.666767, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.566050
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 21.800000
    563:   28116 [INFO] CHASSIS_PID_TURN, Time: 7388, Actual_Vol1: -1866.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16370.254565, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.478043, Absolute Angle: 1.666262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.54886
9, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 43.600000
    564:   28118 [INFO] CHASSIS_PID_TURN, Time: 7398, Actual_Vol1: -1860.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 986.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16315.804386, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.445018, Absolute Angle: 1.665686, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536367
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -14.200000, Actual_Vel4: 33.000000
    565:   28118 [INFO] CHASSIS_PID_TURN, Time: 7408, Actual_Vol1: -1959.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16261.657866, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.414652, Absolute Angle: 1.665156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.529304
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.000000
    566:   28118 [INFO] CHASSIS_PID_TURN, Time: 7418, Actual_Vol1: -1971.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16207.772419, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.388545, Absolute Angle: 1.664700, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.515466
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 54.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.000000
    567:   28120 [INFO] CHASSIS_PID_TURN, Time: 7428, Actual_Vol1: -2045.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 967.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16154.081454, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.369096, Absolute Angle: 1.664360, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.497132
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 32.200000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
    568:   28120 [INFO] CHASSIS_PID_TURN, Time: 7438, Actual_Vol1: -2064.000000, Actual_Vol2: 2624.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1004.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16100.624707, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.345675, Absolute Angle: 1.663952, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.49013
4, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
    569:   28120 [INFO] CHASSIS_PID_TURN, Time: 7448, Actual_Vol1: -2070.000000, Actual_Vol2: 2526.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 16047.419153, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.320555, Absolute Angle: 1.663513, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.486209
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
    570:   28122 [INFO] CHASSIS_PID_TURN, Time: 7458, Actual_Vol1: -2230.000000, Actual_Vol2: 2612.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15994.363039, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.305611, Absolute Angle: 1.663252, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.480333
, over slew: 0, Actual_Vel1: -22.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
    571:   28122 [INFO] CHASSIS_PID_TURN, Time: 7468, Actual_Vol1: -2193.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15941.535097, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.282794, Absolute Angle: 1.662854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.475295
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -15.200000, Actual_Vel4: 33.800000
    572:   28122 [INFO] CHASSIS_PID_TURN, Time: 7478, Actual_Vol1: -2101.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15888.930142, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.260496, Absolute Angle: 1.662465, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.472920
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 22.800000
    573:   28124 [INFO] CHASSIS_PID_TURN, Time: 7488, Actual_Vol1: -2088.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15836.518122, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.241202, Absolute Angle: 1.662128, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.469974
, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
    574:   28124 [INFO] CHASSIS_PID_TURN, Time: 7498, Actual_Vol1: -2020.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15784.340827, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.217729, Absolute Angle: 1.661719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.465879
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
    575:   28124 [INFO] CHASSIS_PID_TURN, Time: 7508, Actual_Vol1: -1971.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15732.392410, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.194842, Absolute Angle: 1.661319, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.460435
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -10.800000, Actual_Vel4: 19.200000
    576:   28126 [INFO] CHASSIS_PID_TURN, Time: 7518, Actual_Vol1: -2020.000000, Actual_Vol2: 2784.000000, Actual_Vol3: -1632.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15680.758196, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.163421, Absolute Angle: 1.660771, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.466114
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 19.400000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
    577:   28126 [INFO] CHASSIS_PID_TURN, Time: 7528, Actual_Vol1: -1977.000000, Actual_Vol2: 2821.000000, Actual_Vol3: -1602.000000, Actual_Vol4: 850.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15629.415303, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.134289, Absolute Angle: 1.660262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.475416
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
    578:   28126 [INFO] CHASSIS_PID_TURN, Time: 7538, Actual_Vol1: -2002.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -1743.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15578.360528, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.105478, Absolute Angle: 1.659759, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.478318
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -19.200000, Actual_Vel4: 26.800000
    579:   28128 [INFO] CHASSIS_PID_TURN, Time: 7548, Actual_Vol1: -2057.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15527.519889, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.084064, Absolute Angle: 1.659386, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.472649
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 11.000000, Actual_Vel3: -14.000000, Actual_Vel4: 26.800000
    580:   28128 [INFO] CHASSIS_PID_TURN, Time: 7558, Actual_Vol1: -2070.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15476.916654, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.060324, Absolute Angle: 1.658971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.473406
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 26.800000
    581:   28128 [INFO] CHASSIS_PID_TURN, Time: 7568, Actual_Vol1: -2131.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15426.495463, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.042119, Absolute Angle: 1.658654, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.464836
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
    582:   28130 [INFO] CHASSIS_PID_TURN, Time: 7578, Actual_Vol1: -2230.000000, Actual_Vol2: 2784.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15376.238192, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.025727, Absolute Angle: 1.658368, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.452315
, over slew: 0, Actual_Vel1: -7.400000, Actual_Vel2: 7.400000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
    583:   28130 [INFO] CHASSIS_PID_TURN, Time: 7588, Actual_Vol1: -2255.000000, Actual_Vol2: 2760.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 825.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15326.107341, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.013085, Absolute Angle: 1.658147, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.431933
, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
    584:   28130 [INFO] CHASSIS_PID_TURN, Time: 7598, Actual_Vol1: -2248.000000, Actual_Vol2: 2741.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15276.052654, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 95.005469, Absolute Angle: 1.658014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.409183
, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -14.000000, Actual_Vel4: 43.400000
    585:   28132 [INFO] CHASSIS_PID_TURN, Time: 7608, Actual_Vol1: -2261.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15226.086710, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.996594, Absolute Angle: 1.657859, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.391950
, over slew: 0, Actual_Vel1: -7.600000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 43.400000
    586:   28132 [INFO] CHASSIS_PID_TURN, Time: 7618, Actual_Vol1: -2255.000000, Actual_Vol2: 2735.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15176.222639, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.986407, Absolute Angle: 1.657681, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.382689
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 23.800000
    587:   28132 [INFO] CHASSIS_PID_TURN, Time: 7628, Actual_Vol1: -2255.000000, Actual_Vol2: 2729.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15126.466296, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.975634, Absolute Angle: 1.657493, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.370041
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 8.000000, Actual_Vel3: -35.000000, Actual_Vel4: 23.800000
    588:   28134 [INFO] CHASSIS_PID_TURN, Time: 7638, Actual_Vol1: -2310.000000, Actual_Vol2: 2575.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15076.897263, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.956903, Absolute Angle: 1.657166, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.363652
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 23.800000
    589:   28134 [INFO] CHASSIS_PID_TURN, Time: 7648, Actual_Vol1: -2359.000000, Actual_Vol2: 2624.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 15027.518531, kD: 35.000000
```

```
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.937873, Absolute Angle: 1.656834, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.367738
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 23.800000
    28134 [INFO] CHASSIS_PID_TURN, Time: 7658, Actual_Vol1: -2365.000000, Actual_Vol2: 2698.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14978.297407, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -167.000000, Heading_Sp: 90.000000, Relative_Heading: 94.922112, Absolute Angle: 1.656559, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.360682
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 11.000000, Actual_Vel3: -0.000000, Actual_Vel4: 15.800000
    28136 [INFO] CHASSIS_PID_TURN, Time: 7668, Actual_Vol1: -2365.000000, Actual_Vol2: 2575.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14929.189940, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.910747, Absolute Angle: 1.656361, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.349749
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000
    28136 [INFO] CHASSIS_PID_TURN, Time: 7678, Actual_Vol1: -2341.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -1928.000000, Actual_Vol4: 850.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14881.243700, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.794624, Absolute Angle: 1.654334, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.446578
, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000
    28136 [INFO] CHASSIS_PID_TURN, Time: 7688, Actual_Vol1: -2267.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14833.455263, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.778844, Absolute Angle: 1.654059, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.438886
, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 9.000000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000
    28138 [INFO] CHASSIS_PID_TURN, Time: 7698, Actual_Vol1: -610.000000, Actual_Vol2: 782.000000, Actual_Vol3: -425.000000, Actual_Vol4: 179.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14785.806834, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764843, Absolute Angle: 1.653814, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.429999, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 4.200000, Actual_Vel3: -33.600000, Actual_Vel4: 15.800000
    28138 [INFO] CHASSIS_PID_TURN, Time: 7708, Actual_Vol1: -136.000000, Actual_Vol2: 197.000000, Actual_Vol3: -80.000000, Actual_Vol4: 37.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14738.271980, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.753485, Absolute Angle: 1.653616, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.409936, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 15.800000
    28138 [INFO] CHASSIS_PID_TURN, Time: 7718, Actual_Vol1: -92.000000, Actual_Vol2: 296.000000, Actual_Vol3: -18.000000, Actual_Vol4: 12.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14690.784542, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.748744, Absolute Angle: 1.653533, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.385545, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.000000
    28140 [INFO] CHASSIS_PID_TURN, Time: 7728, Actual_Vol1: -277.000000, Actual_Vol2: 431.000000, Actual_Vol3: -80.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14643.245777, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.753877, Absolute Angle: 1.653623, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.356734, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 6.800000, Actual_Vel4: 8.000000
    28140 [INFO] CHASSIS_PID_TURN, Time: 7738, Actual_Vol1: -308.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14595.546183, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.769959, Absolute Angle: 1.653904, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.335320, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 8.000000
    28140 [INFO] CHASSIS_PID_TURN, Time: 7748, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14547.612216, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793397, Absolute Angle: 1.654313, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311580, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000
    28142 [INFO] CHASSIS_PID_TURN, Time: 7758, Actual_Vol1: -320.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14499.526499, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808572, Absolute Angle: 1.654577, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.293375, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -4.200000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000
    28142 [INFO] CHASSIS_PID_TURN, Time: 7768, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14451.318271, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.820823, Absolute Angle: 1.654791, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.276983, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000
    28142 [INFO] CHASSIS_PID_TURN, Time: 7778, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14403.092581, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.822569, Absolute Angle: 1.654822, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.264341, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 6.800000, Actual_Vel4: 0.000000
    28144 [INFO] CHASSIS_PID_TURN, Time: 7788, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14354.863668, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.822891, Absolute Angle: 1.654827, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.256725, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -24.400000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000
    28144 [INFO] CHASSIS_PID_TURN, Time: 7798, Actual_Vol1: -320.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14306.645176, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.821849, Absolute Angle: 1.654809, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.247851, over s
lew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000
    28144 [INFO] CHASSIS_PID_TURN, Time: 7808, Actual_Vol1: -314.000000, Actual_Vol2: 456.000000, Actual_Vol3: -86.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14258.452662, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819251, Absolute Angle: 1.654764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.237663, over s
lew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 17.200000, Actual_Vel4: 0.000000
    28146 [INFO] CHASSIS_PID_TURN, Time: 7818, Actual_Vol1: -314.000000, Actual_Vol2: 493.000000, Actual_Vol3: -92.000000, Actual_Vol4: -6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14210.350980, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.810168, Absolute Angle: 1.654605, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.226890, over
 slew: 0, Actual_Vel1: 8.800000, Actual_Vel2: -17.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000
    28146 [INFO] CHASSIS_PID_TURN, Time: 7828, Actual_Vol1: -413.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14162.229456, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.812152, Absolute Angle: 1.654640, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.208160, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000
    28146 [INFO] CHASSIS_PID_TURN, Time: 7838, Actual_Vol1: -431.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14114.110500, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811896, Absolute Angle: 1.654636, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.189129, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 9.800000, Actual_Vel4: 0.000000
    28148 [INFO] CHASSIS_PID_TURN, Time: 7848, Actual_Vol1: -437.000000, Actual_Vol2: 536.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14065.923561, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.818694, Absolute Angle: 1.654754, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.173369, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28148 [INFO] CHASSIS_PID_TURN, Time: 7858, Actual_Vol1: -431.000000, Actual_Vol2: 542.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 14017.770319, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815324, Absolute Angle: 1.654695, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.162003, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28148 [INFO] CHASSIS_PID_TURN, Time: 7868, Actual_Vol1: -474.000000, Actual_Vol2: 585.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13969.580198, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819012, Absolute Angle: 1.654760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28150 [INFO] CHASSIS_PID_TURN, Time: 7878, Actual_Vol1: -499.000000, Actual_Vol2: 628.000000, Actual_Vol3: -191.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13921.390719, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.818948, Absolute Angle: 1.654759, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28150 [INFO] CHASSIS_PID_TURN, Time: 7888, Actual_Vol1: -517.000000, Actual_Vol2: 641.000000, Actual_Vol3: -222.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13873.232923, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815780, Absolute Angle: 1.654703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28150 [INFO] CHASSIS_PID_TURN, Time: 7898, Actual_Vol1: -511.000000, Actual_Vol2: 634.000000, Actual_Vol3: -228.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13825.090933, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814199, Absolute Angle: 1.654676, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28152 [INFO] CHASSIS_PID_TURN, Time: 7908, Actual_Vol1: -511.000000, Actual_Vol2: 634.000000, Actual_Vol3: -222.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13776.958291, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.813264, Absolute Angle: 1.654659, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074148, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28152 [INFO] CHASSIS_PID_TURN, Time: 7918, Actual_Vol1: -554.000000, Actual_Vol2: 678.000000, Actual_Vol3: -222.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13728.866303, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.809199, Absolute Angle: 1.654588, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069015, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28152 [INFO] CHASSIS_PID_TURN, Time: 7928, Actual_Vol1: -610.000000, Actual_Vol2: 715.000000, Actual_Vol3: -283.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13680.792486, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807382, Absolute Angle: 1.654557, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.052932, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28154 [INFO] CHASSIS_PID_TURN, Time: 7938, Actual_Vol1: -616.000000, Actual_Vol2: 752.000000, Actual_Vol3: -302.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13632.751935, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804055, Absolute Angle: 1.654499, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029495, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28154 [INFO] CHASSIS_PID_TURN, Time: 7948, Actual_Vol1: -616.000000, Actual_Vol2: 758.000000, Actual_Vol3: -308.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13584.681295, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807064, Absolute Angle: 1.654551, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28154 [INFO] CHASSIS_PID_TURN, Time: 7958, Actual_Vol1: -616.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13536.540629, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814067, Absolute Angle: 1.654673, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28156 [INFO] CHASSIS_PID_TURN, Time: 7968, Actual_Vol1: -653.000000, Actual_Vol2: 838.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13488.373965, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.816666, Absolute Angle: 1.654719, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28156 [INFO] CHASSIS_PID_TURN, Time: 7978, Actual_Vol1: -690.000000, Actual_Vol2: 924.000000, Actual_Vol3: -407.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13440.230935, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814303, Absolute Angle: 1.654678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018836, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28156 [INFO] CHASSIS_PID_TURN, Time: 7988, Actual_Vol1: -702.000000, Actual_Vol2: 942.000000, Actual_Vol3: -425.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13392.073555, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815738, Absolute Angle: 1.654703, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017794, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28158 [INFO] CHASSIS_PID_TURN, Time: 7998, Actual_Vol1: -702.000000, Actual_Vol2: 942.000000, Actual_Vol3: -431.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13343.880530, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.819303, Absolute Angle: 1.654765, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28158 [INFO] CHASSIS_PID_TURN, Time: 8008, Actual_Vol1: -702.000000, Actual_Vol2: 936.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13295.756307, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.812422, Absolute Angle: 1.654645, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    28158 [INFO] CHASSIS_PID_TURN, Time: 8018, Actual_Vol1: -819.000000, Actual_Vol2: 992.000000, Actual_Vol3: -431.000000, Actual_Vol4: 333.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13247.690020, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.806629, Absolute Angle: 1.654544, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
```

```
28160 [INFO] CHASSIS_PID_TURN, Time: 8028, Actual_Vol1: -912.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -505.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13199.646593, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804343, Absolute Angle: 1.654504, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28160 [INFO] CHASSIS_PID_TURN, Time: 8038, Actual_Vol1: -924.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 450.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13151.568288, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807830, Absolute Angle: 1.654565, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28160 [INFO] CHASSIS_PID_TURN, Time: 8048, Actual_Vol1: -924.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -517.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13103.490114, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807817, Absolute Angle: 1.654564, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28162 [INFO] CHASSIS_PID_TURN, Time: 8058, Actual_Vol1: -967.000000, Actual_Vol2: 1090.000000, Actual_Vol3: -517.000000, Actual_Vol4: 456.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13055.409430, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808068, Absolute Angle: 1.654569, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28162 [INFO] CHASSIS_PID_TURN, Time: 8068, Actual_Vol1: -1010.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -598.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 13007.267906, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.814152, Absolute Angle: 1.654675, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28162 [INFO] CHASSIS_PID_TURN, Time: 8078, Actual_Vol1: -1029.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -616.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12959.153057, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811485, Absolute Angle: 1.654628, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28164 [INFO] CHASSIS_PID_TURN, Time: 8088, Actual_Vol1: -1016.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -610.000000, Actual_Vol4: 542.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12911.038301, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.811476, Absolute Angle: 1.654628, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28164 [INFO] CHASSIS_PID_TURN, Time: 8098, Actual_Vol1: -1023.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -610.000000, Actual_Vol4: 536.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12862.882038, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.815626, Absolute Angle: 1.654701, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28164 [INFO] CHASSIS_PID_TURN, Time: 8108, Actual_Vol1: -1060.000000, Actual_Vol2: 1214.000000, Actual_Vol3: -610.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12814.804135, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807790, Absolute Angle: 1.654564, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015247, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28166 [INFO] CHASSIS_PID_TURN, Time: 8118, Actual_Vol1: -1103.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12766.771454, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.803268, Absolute Angle: 1.654485, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016034, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28166 [INFO] CHASSIS_PID_TURN, Time: 8128, Actual_Vol1: -1152.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -702.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12718.777104, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.799435, Absolute Angle: 1.654418, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28166 [INFO] CHASSIS_PID_TURN, Time: 8138, Actual_Vol1: -1152.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12670.782526, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.799458, Absolute Angle: 1.654418, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28168 [INFO] CHASSIS_PID_TURN, Time: 8148, Actual_Vol1: -1183.000000, Actual_Vol2: 1312.000000, Actual_Vol3: -702.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12622.743310, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.803922, Absolute Angle: 1.654496, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28168 [INFO] CHASSIS_PID_TURN, Time: 8158, Actual_Vol1: -1238.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -893.000000, Actual_Vol4: 696.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12574.691970, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.805134, Absolute Angle: 1.654517, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28168 [INFO] CHASSIS_PID_TURN, Time: 8168, Actual_Vol1: -1238.000000, Actual_Vol2: 1324.000000, Actual_Vol3: -764.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12526.648875, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.804309, Absolute Angle: 1.654503, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28170 [INFO] CHASSIS_PID_TURN, Time: 8178, Actual_Vol1: -1250.000000, Actual_Vol2: 1275.000000, Actual_Vol3: -708.000000, Actual_Vol4: 721.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12478.575820, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807306, Absolute Angle: 1.654555, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28170 [INFO] CHASSIS_PID_TURN, Time: 8188, Actual_Vol1: -1287.000000, Actual_Vol2: 1306.000000, Actual_Vol3: -702.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12430.496531, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807929, Absolute Angle: 1.654566, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019868, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28170 [INFO] CHASSIS_PID_TURN, Time: 8198, Actual_Vol1: -1324.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -899.000000, Actual_Vol4: 906.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12382.387943, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.810859, Absolute Angle: 1.654617, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28172 [INFO] CHASSIS_PID_TURN, Time: 8208, Actual_Vol1: -1337.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 918.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12334.288670, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.809927, Absolute Angle: 1.654601, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28172 [INFO] CHASSIS_PID_TURN, Time: 8218, Actual_Vol1: -1337.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12286.207463, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808121, Absolute Angle: 1.654570, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28172 [INFO] CHASSIS_PID_TURN, Time: 8228, Actual_Vol1: -1374.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -936.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12238.120661, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.808680, Absolute Angle: 1.654579, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28174 [INFO] CHASSIS_PID_TURN, Time: 8238, Actual_Vol1: -1423.000000, Actual_Vol2: 1448.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 1010.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12190.047165, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.807350, Absolute Angle: 1.654556, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28174 [INFO] CHASSIS_PID_TURN, Time: 8248, Actual_Vol1: -1429.000000, Actual_Vol2: 1448.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12142.019942, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.802722, Absolute Angle: 1.654475, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016191, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 0.000000
28174 [INFO] CHASSIS_PID_TURN, Time: 8258, Actual_Vol1: -1441.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1041.000000, Actual_Vol4: 961.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12094.056572, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.796337, Absolute Angle: 1.654364, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019289, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000
28176 [INFO] CHASSIS_PID_TURN, Time: 8268, Actual_Vol1: -1466.000000, Actual_Vol2: 1534.000000, Actual_Vol3: -1035.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 12046.129698, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.792687, Absolute Angle: 1.654300, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000
28176 [INFO] CHASSIS_PID_TURN, Time: 8278, Actual_Vol1: -1558.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1146.000000, Actual_Vol4: 1016.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11998.159657, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.797004, Absolute Angle: 1.654376, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000
28176 [INFO] CHASSIS_PID_TURN, Time: 8288, Actual_Vol1: -1565.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11950.202759, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.795690, Absolute Angle: 1.654353, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022939, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -12.200000, Actual_Vel4: 7.600000
28178 [INFO] CHASSIS_PID_TURN, Time: 8298, Actual_Vol1: -1565.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1164.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11902.273446, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.792931, Absolute Angle: 1.654305, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 44.200000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28178 [INFO] CHASSIS_PID_TURN, Time: 8308, Actual_Vol1: -1602.000000, Actual_Vol2: 1639.000000, Actual_Vol3: -1164.000000, Actual_Vol4: 1041.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11854.335486, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793796, Absolute Angle: 1.654320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28178 [INFO] CHASSIS_PID_TURN, Time: 8318, Actual_Vol1: -1657.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1238.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11806.405436, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.793005, Absolute Angle: 1.654306, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018171, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28180 [INFO] CHASSIS_PID_TURN, Time: 8328, Actual_Vol1: -1663.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1164.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11758.527678, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.787776, Absolute Angle: 1.654215, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28180 [INFO] CHASSIS_PID_TURN, Time: 8338, Actual_Vol1: -1669.000000, Actual_Vol2: 1694.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11710.621645, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.790603, Absolute Angle: 1.654264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28180 [INFO] CHASSIS_PID_TURN, Time: 8348, Actual_Vol1: -1706.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1263.000000, Actual_Vol4: 1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11662.681530, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.794012, Absolute Angle: 1.654323, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28182 [INFO] CHASSIS_PID_TURN, Time: 8358, Actual_Vol1: -1749.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1263.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11614.792331, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.788920, Absolute Angle: 1.654234, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023083, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28182 [INFO] CHASSIS_PID_TURN, Time: 8368, Actual_Vol1: -1762.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1269.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11566.939926, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.785241, Absolute Angle: 1.654170, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025618, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 7.600000
28182 [INFO] CHASSIS_PID_TURN, Time: 8378, Actual_Vol1: -1799.000000, Actual_Vol2: 1823.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11519.110834, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.782909, Absolute Angle: 1.654130, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027950, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28184 [INFO] CHASSIS_PID_TURN, Time: 8388, Actual_Vol1: -1842.000000, Actual_Vol2: 1879.000000, Actual_Vol3: -1435.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11471.280196, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.783064, Absolute Angle: 1.654132, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027950, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
28184 [INFO] CHASSIS_PID_TURN, Time: 8398, Actual_Vol1: -1854.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 1275.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11423.425110, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.785509, Absolute Angle: 1.654175, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02701
```

```
8, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28184 [INFO] CHASSIS_PID_TURN, Time: 8408, Actual_Vol1: -1860.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1460.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11375.607290, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.781782, Absolute Angle: 1.654110, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02689
8, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28186 [INFO] CHASSIS_PID_TURN, Time: 8418, Actual_Vol1: -1928.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1454.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11327.823362, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.778393, Absolute Angle: 1.654051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03028
7, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28186 [INFO] CHASSIS_PID_TURN, Time: 8428, Actual_Vol1: -1965.000000, Actual_Vol2: 1977.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1324.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11280.088308, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.773505, Absolute Angle: 1.653965, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03384
4, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28186 [INFO] CHASSIS_PID_TURN, Time: 8438, Actual_Vol1: -1977.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1589.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11232.375681, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.771263, Absolute Angle: 1.653926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03146
0, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28188 [INFO] CHASSIS_PID_TURN, Time: 8448, Actual_Vol1: -2088.000000, Actual_Vol2: 1959.000000, Actual_Vol3: -1589.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11184.709671, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.766601, Absolute Angle: 1.653845, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03040
3, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28188 [INFO] CHASSIS_PID_TURN, Time: 8458, Actual_Vol1: -2218.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1737.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11137.064974, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764470, Absolute Angle: 1.653808, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03253
4, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28188 [INFO] CHASSIS_PID_TURN, Time: 8468, Actual_Vol1: -2242.000000, Actual_Vol2: 2008.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11089.393564, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.767141, Absolute Angle: 1.653854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03253
4, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28190 [INFO] CHASSIS_PID_TURN, Time: 8478, Actual_Vol1: -2248.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1448.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 11041.753320, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.764024, Absolute Angle: 1.653800, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03166
5, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28190 [INFO] CHASSIS_PID_TURN, Time: 8488, Actual_Vol1: -2261.000000, Actual_Vol2: 2107.000000, Actual_Vol3: -1780.000000, Actual_Vol4: 1478.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10994.139135, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.761419, Absolute Angle: 1.653755, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03259
3, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28190 [INFO] CHASSIS_PID_TURN, Time: 8498, Actual_Vol1: -2248.000000, Actual_Vol2: 2236.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10946.616701, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.752243, Absolute Angle: 1.653594, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.04176
8, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -0.000000, Actual_Vel4: 18.800000
    28192 [INFO] CHASSIS_PID_TURN, Time: 8508, Actual_Vol1: -2255.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10899.215497, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.740120, Absolute Angle: 1.653383, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.05389
1, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 18.800000
    28192 [INFO] CHASSIS_PID_TURN, Time: 8518, Actual_Vol1: -2322.000000, Actual_Vol2: 2335.000000, Actual_Vol3: -1793.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10851.935681, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.727982, Absolute Angle: 1.653171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.06603
0, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 18.800000
    28192 [INFO] CHASSIS_PID_TURN, Time: 8528, Actual_Vol1: -2378.000000, Actual_Vol2: 2384.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10804.730562, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.720512, Absolute Angle: 1.653041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.07350
0, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -121.400000, Actual_Vel4: 8.000000
    28194 [INFO] CHASSIS_PID_TURN, Time: 8538, Actual_Vol1: -2365.000000, Actual_Vol2: 2396.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10757.693002, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.703756, Absolute Angle: 1.652748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.09025
6, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 17.800000, Actual_Vel3: -53.800000, Actual_Vel4: 8.000000
    28194 [INFO] CHASSIS_PID_TURN, Time: 8548, Actual_Vol1: -2378.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1786.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10710.854184, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -166.000000, Heading_Sp: 90.000000, Relative_Heading: 94.683882, Absolute Angle: 1.652401, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.10503
8, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
    28194 [INFO] CHASSIS_PID_TURN, Time: 8558, Actual_Vol1: -2384.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10664.258594, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.659559, Absolute Angle: 1.651977, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.12595
0, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
    28196 [INFO] CHASSIS_PID_TURN, Time: 8568, Actual_Vol1: -2384.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10618.902689, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.535590, Absolute Angle: 1.649813, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.24991
8, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
    28196 [INFO] CHASSIS_PID_TURN, Time: 8578, Actual_Vol1: -2384.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10573.802672, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.510002, Absolute Angle: 1.649366, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.27550
7, over slew: 0, Actual_Vel1: -7.200000, Actual_Vel2: 74.400000, Actual_Vel3: -53.800000, Actual_Vel4: 9.400000
    28196 [INFO] CHASSIS_PID_TURN, Time: 8588, Actual_Vol1: -2415.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10529.004126, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.479855, Absolute Angle: 1.648840, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.30565
4, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -53.800000, Actual_Vel4: 14.000000
    28198 [INFO] CHASSIS_PID_TURN, Time: 8598, Actual_Vol1: -2427.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1657.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10484.548980, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.445515, Absolute Angle: 1.648241, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.33626
7, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -53.800000, Actual_Vel4: 14.000000
    28198 [INFO] CHASSIS_PID_TURN, Time: 8608, Actual_Vol1: -2384.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10440.434188, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.411479, Absolute Angle: 1.647647, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.36691
4, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -32.600000, Actual_Vel4: 14.000000
    28198 [INFO] CHASSIS_PID_TURN, Time: 8618, Actual_Vol1: -2427.000000, Actual_Vol2: 2445.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10396.593890, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.384030, Absolute Angle: 1.647168, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.38947
6, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 44.600000, Actual_Vel3: -32.600000, Actual_Vel4: 14.000000
    28200 [INFO] CHASSIS_PID_TURN, Time: 8628, Actual_Vol1: -2445.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1645.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10352.954411, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.363948, Absolute Angle: 1.646817, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.40731
5, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 14.000000
    28200 [INFO] CHASSIS_PID_TURN, Time: 8638, Actual_Vol1: -2427.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1885.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10309.553997, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.340041, Absolute Angle: 1.646400, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.42710
0, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 14.000000
    28200 [INFO] CHASSIS_PID_TURN, Time: 8648, Actual_Vol1: -2384.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1866.000000, Actual_Vol4: 1682.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10266.362055, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.319194, Absolute Angle: 1.646036, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.44794
7, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
    28202 [INFO] CHASSIS_PID_TURN, Time: 8658, Actual_Vol1: -2390.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1589.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10223.301003, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.306105, Absolute Angle: 1.645808, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46103
6, over slew: 0, Actual_Vel1: -11.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
    28202 [INFO] CHASSIS_PID_TURN, Time: 8668, Actual_Vol1: -2421.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10180.404155, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.289685, Absolute Angle: 1.645521, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.47434
0, over slew: 0, Actual_Vel1: -16.000000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 15.200000
    28202 [INFO] CHASSIS_PID_TURN, Time: 8678, Actual_Vol1: -2464.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10137.679945, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.272421, Absolute Angle: 1.645220, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.48899
8, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 44.600000, Actual_Vel3: -14.600000, Actual_Vel4: 18.400000
    28204 [INFO] CHASSIS_PID_TURN, Time: 8688, Actual_Vol1: -2384.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1749.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10095.174294, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.250565, Absolute Angle: 1.644838, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.50167
8, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 15.400000, Actual_Vel3: -11.200000, Actual_Vel4: 18.400000
    28204 [INFO] CHASSIS_PID_TURN, Time: 8698, Actual_Vol1: -2378.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1768.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10052.936666, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.223763, Absolute Angle: 1.644371, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.51635
8, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 15.400000, Actual_Vel3: -11.200000, Actual_Vel4: 15.000000
    28204 [INFO] CHASSIS_PID_TURN, Time: 8708, Actual_Vol1: -2445.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1688.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 10010.960418, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.197625, Absolute Angle: 1.643914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.53035
7, over slew: 0, Actual_Vel1: -14.400000, Actual_Vel2: 8.400000, Actual_Vel3: -11.200000, Actual_Vel4: 15.000000
    28206 [INFO] CHASSIS_PID_TURN, Time: 8718, Actual_Vol1: -2476.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1682.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9969.283510, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.167691, Absolute Angle: 1.643392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.552821
, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 15.000000
    28206 [INFO] CHASSIS_PID_TURN, Time: 8728, Actual_Vol1: -2482.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9927.929555, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.135396, Absolute Angle: 1.642828, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.568360
, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
    28206 [INFO] CHASSIS_PID_TURN, Time: 8738, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9886.872932, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.105662, Absolute Angle: 1.642309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578220
, over slew: 0, Actual_Vel1: -15.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
    28208 [INFO] CHASSIS_PID_TURN, Time: 8748, Actual_Vol1: -2482.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9846.058266, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -165.000000, Heading_Sp: 90.000000, Relative_Heading: 94.081467, Absolute Angle: 1.641887, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578092
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 9.400000
    28208 [INFO] CHASSIS_PID_TURN, Time: 8758, Actual_Vol1: -2476.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9805.406102, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 94.065216, Absolute Angle: 1.641603, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.470374
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 12.000000
    28208 [INFO] CHASSIS_PID_TURN, Time: 8768, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9765.884228, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -164.000000, Heading_Sp: 90.000000, Relative_Heading: 93.952187, Absolute Angle: 1.639631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.557814
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.400000, Actual_Vel3: -13.000000, Actual_Vel4: 12.000000
    28210 [INFO] CHASSIS_PID_TURN, Time: 8778, Actual_Vol1: -2470.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2002.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9726.507020, kD: 35.000000
```

```
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.937721, Absolute Angle: 1.639378, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.542134
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 8.800000, Actual_Vel3: -30.800000, Actual_Vel4: 12.000000
    703:   28210 [INFO] CHASSIS_PID_TURN, Time: 8788, Actual_Vol1: -2464.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9688.250017, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825700, Absolute Angle: 1.637423, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.619814
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 12.000000
    704:   28210 [INFO] CHASSIS_PID_TURN, Time: 8798, Actual_Vol1: -1423.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -2014.000000, Actual_Vol4: 1491.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9650.108517, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.814150, Absolute Angle: 1.637222, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.597329
, over slew: 0, Actual_Vel1: -12.200000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 14.600000
    705:   28212 [INFO] CHASSIS_PID_TURN, Time: 8808, Actual_Vol1: -308.000000, Actual_Vol2: 610.000000, Actual_Vol3: -505.000000, Actual_Vol4: 326.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9612.111502, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.799702, Absolute Angle: 1.636969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.584328, ov
er slew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 9.200000, Actual_Vel3: -30.800000, Actual_Vel4: 14.600000
    706:   28212 [INFO] CHASSIS_PID_TURN, Time: 8818, Actual_Vol1: -209.000000, Actual_Vol2: 142.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9574.284811, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.782669, Absolute Angle: 1.636672, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.581279, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: -0.000000, Actual_Vel4: 7.000000
    707:   28212 [INFO] CHASSIS_PID_TURN, Time: 8828, Actual_Vol1: -376.000000, Actual_Vol2: 142.000000, Actual_Vol3: -18.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9536.572311, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.771250, Absolute Angle: 1.636473, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.568791, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000
    708:   28214 [INFO] CHASSIS_PID_TURN, Time: 8838, Actual_Vol1: -431.000000, Actual_Vol2: 290.000000, Actual_Vol3: -86.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9498.876339, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.769597, Absolute Angle: 1.636444, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.549597, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000
    709:   28214 [INFO] CHASSIS_PID_TURN, Time: 8848, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -86.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9461.090957, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.778538, Absolute Angle: 1.636600, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536508, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 7.000000
    710:   28214 [INFO] CHASSIS_PID_TURN, Time: 8858, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9423.068570, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802239, Absolute Angle: 1.637014, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.520088, over
 slew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000
    711:   28216 [INFO] CHASSIS_PID_TURN, Time: 8868, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9384.852787, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.821578, Absolute Angle: 1.637351, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.502824, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: 3.400000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000
    712:   28216 [INFO] CHASSIS_PID_TURN, Time: 8878, Actual_Vol1: -431.000000, Actual_Vol2: 376.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9346.463682, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838910, Absolute Angle: 1.637654, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.480968, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -0.000000, Actual_Vel3: 13.800000, Actual_Vel4: 6.200000
    713:   28216 [INFO] CHASSIS_PID_TURN, Time: 8888, Actual_Vol1: -431.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9308.043660, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.842002, Absolute Angle: 1.637708, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.454165, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -0.000000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000
    714:   28218 [INFO] CHASSIS_PID_TURN, Time: 8898, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9269.638012, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.840565, Absolute Angle: 1.637683, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.428028, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000
    715:   28218 [INFO] CHASSIS_PID_TURN, Time: 8908, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9231.228000, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.841001, Absolute Angle: 1.637690, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.398094, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: 0.000000
    716:   28218 [INFO] CHASSIS_PID_TURN, Time: 8918, Actual_Vol1: -474.000000, Actual_Vol2: 407.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9192.822930, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.840507, Absolute Angle: 1.637682, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.365798, over s
lew: 0, Actual_Vel1: 7.000000, Actual_Vel2: -9.800000, Actual_Vel3: 13.800000, Actual_Vel4: -10.000000
    717:   28220 [INFO] CHASSIS_PID_TURN, Time: 8928, Actual_Vol1: -511.000000, Actual_Vol2: 444.000000, Actual_Vol3: -92.000000, Actual_Vol4: 99.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9154.528393, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829454, Absolute Angle: 1.637489, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.336065, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: -10.000000
    718:   28220 [INFO] CHASSIS_PID_TURN, Time: 8938, Actual_Vol1: -517.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9116.278227, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825017, Absolute Angle: 1.637411, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311869, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: -10.000000
    719:   28220 [INFO] CHASSIS_PID_TURN, Time: 8948, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9078.024683, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825354, Absolute Angle: 1.637417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.295619, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 10.400000, Actual_Vel4: 0.000000
    720:   28222 [INFO] CHASSIS_PID_TURN, Time: 8958, Actual_Vol1: -524.000000, Actual_Vol2: 493.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9039.695346, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832934, Absolute Angle: 1.637549, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.182590, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    721:   28222 [INFO] CHASSIS_PID_TURN, Time: 8968, Actual_Vol1: -567.000000, Actual_Vol2: 530.000000, Actual_Vol3: -203.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 9001.363272, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833207, Absolute Angle: 1.637554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.168124, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    722:   28222 [INFO] CHASSIS_PID_TURN, Time: 8978, Actual_Vol1: -598.000000, Actual_Vol2: 536.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8963.066825, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829645, Absolute Angle: 1.637492, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    723:   28224 [INFO] CHASSIS_PID_TURN, Time: 8988, Actual_Vol1: -616.000000, Actual_Vol2: 542.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8924.735179, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833165, Absolute Angle: 1.637553, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    724:   28224 [INFO] CHASSIS_PID_TURN, Time: 8998, Actual_Vol1: -616.000000, Actual_Vol2: 536.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8886.404676, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833050, Absolute Angle: 1.637551, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    725:   28224 [INFO] CHASSIS_PID_TURN, Time: 9008, Actual_Vol1: -659.000000, Actual_Vol2: 579.000000, Actual_Vol3: -234.000000, Actual_Vol4: 234.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8848.145580, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825910, Absolute Angle: 1.637427, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    726:   28226 [INFO] CHASSIS_PID_TURN, Time: 9018, Actual_Vol1: -690.000000, Actual_Vol2: 628.000000, Actual_Vol3: -290.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8809.848196, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829738, Absolute Angle: 1.637494, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    727:   28226 [INFO] CHASSIS_PID_TURN, Time: 9028, Actual_Vol1: -702.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8771.511791, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833640, Absolute Angle: 1.637562, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072405, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    728:   28226 [INFO] CHASSIS_PID_TURN, Time: 9038, Actual_Vol1: -702.000000, Actual_Vol2: 634.000000, Actual_Vol3: -314.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8733.135983, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.837581, Absolute Angle: 1.637631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063464, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    729:   28228 [INFO] CHASSIS_PID_TURN, Time: 9048, Actual_Vol1: -825.000000, Actual_Vol2: 671.000000, Actual_Vol3: -394.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8694.792686, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834330, Absolute Angle: 1.637574, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039764, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    730:   28228 [INFO] CHASSIS_PID_TURN, Time: 9058, Actual_Vol1: -899.000000, Actual_Vol2: 715.000000, Actual_Vol3: -425.000000, Actual_Vol4: 413.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8656.407600, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838509, Absolute Angle: 1.637647, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.020424, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    731:   28228 [INFO] CHASSIS_PID_TURN, Time: 9068, Actual_Vol1: -918.000000, Actual_Vol2: 758.000000, Actual_Vol3: -431.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8618.022995, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.838460, Absolute Angle: 1.637646, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016986, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    732:   28230 [INFO] CHASSIS_PID_TURN, Time: 9078, Actual_Vol1: -924.000000, Actual_Vol2: 758.000000, Actual_Vol3: -431.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8579.668747, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.835425, Absolute Angle: 1.637593, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016986, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    733:   28230 [INFO] CHASSIS_PID_TURN, Time: 9088, Actual_Vol1: -973.000000, Actual_Vol2: 838.000000, Actual_Vol3: -474.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8541.350433, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.831831, Absolute Angle: 1.637530, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015985, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    734:   28230 [INFO] CHASSIS_PID_TURN, Time: 9098, Actual_Vol1: -1016.000000, Actual_Vol2: 918.000000, Actual_Vol3: -517.000000, Actual_Vol4: 499.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8503.000448, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834999, Absolute Angle: 1.637585, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015985, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    735:   28232 [INFO] CHASSIS_PID_TURN, Time: 9108, Actual_Vol1: -1023.000000, Actual_Vol2: 942.000000, Actual_Vol3: -524.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8464.651458, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.834899, Absolute Angle: 1.637584, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.015490, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    736:   28232 [INFO] CHASSIS_PID_TURN, Time: 9118, Actual_Vol1: -1023.000000, Actual_Vol2: 942.000000, Actual_Vol3: -517.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8426.368258, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.828320, Absolute Angle: 1.637469, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.013492, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    737:   28232 [INFO] CHASSIS_PID_TURN, Time: 9128, Actual_Vol1: -1066.000000, Actual_Vol2: 992.000000, Actual_Vol3: -567.000000, Actual_Vol4: 524.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8388.125158, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.824310, Absolute Angle: 1.637399, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    738:   28234 [INFO] CHASSIS_PID_TURN, Time: 9138, Actual_Vol1: -1109.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -610.000000, Actual_Vol4: 591.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8349.880613, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.824455, Absolute Angle: 1.637401, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    739:   28234 [INFO] CHASSIS_PID_TURN, Time: 9148, Actual_Vol1: -1146.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -567.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8311.605127, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.827549, Absolute Angle: 1.637455, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
```

```
28234 [INFO] CHASSIS_PID_TURN, Time: 9158, Actual_Vol1: -1152.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -530.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8273.282705, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832242, Absolute Angle: 1.637537, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28236 [INFO] CHASSIS_PID_TURN, Time: 9168, Actual_Vol1: -1195.000000, Actual_Vol2: 1084.000000, Actual_Vol3: -573.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8234.952886, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832982, Absolute Angle: 1.637550, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28236 [INFO] CHASSIS_PID_TURN, Time: 9178, Actual_Vol1: -1238.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -610.000000, Actual_Vol4: 696.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8196.613676, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833921, Absolute Angle: 1.637567, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28236 [INFO] CHASSIS_PID_TURN, Time: 9188, Actual_Vol1: -1244.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -616.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8158.281592, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833208, Absolute Angle: 1.637554, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28238 [INFO] CHASSIS_PID_TURN, Time: 9198, Actual_Vol1: -1244.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -622.000000, Actual_Vol4: 715.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8119.942452, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.833914, Absolute Angle: 1.637567, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28238 [INFO] CHASSIS_PID_TURN, Time: 9208, Actual_Vol1: -1294.000000, Actual_Vol2: 1207.000000, Actual_Vol3: -659.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8081.592165, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.835029, Absolute Angle: 1.637586, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28238 [INFO] CHASSIS_PID_TURN, Time: 9218, Actual_Vol1: -1343.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8043.263630, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.832854, Absolute Angle: 1.637548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28240 [INFO] CHASSIS_PID_TURN, Time: 9228, Actual_Vol1: -1337.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 8004.978617, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.828501, Absolute Angle: 1.637472, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014199, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28240 [INFO] CHASSIS_PID_TURN, Time: 9238, Actual_Vol1: -1343.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -708.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7966.739377, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.823924, Absolute Angle: 1.637392, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014585, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28240 [INFO] CHASSIS_PID_TURN, Time: 9248, Actual_Vol1: -1380.000000, Actual_Vol2: 1300.000000, Actual_Vol3: -832.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7928.478380, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.826100, Absolute Angle: 1.637430, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014585, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28242 [INFO] CHASSIS_PID_TURN, Time: 9258, Actual_Vol1: -1429.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -912.000000, Actual_Vol4: 1004.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7890.183104, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829528, Absolute Angle: 1.637490, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014536,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28242 [INFO] CHASSIS_PID_TURN, Time: 9268, Actual_Vol1: -1435.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -930.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7851.886783, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.829632, Absolute Angle: 1.637492, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011501,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.800000, Actual_Vel4: 0.000000
28242 [INFO] CHASSIS_PID_TURN, Time: 9278, Actual_Vol1: -1435.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -930.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7813.620635, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.826615, Absolute Angle: 1.637439, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28244 [INFO] CHASSIS_PID_TURN, Time: 9288, Actual_Vol1: -1478.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -973.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7775.369359, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825128, Absolute Angle: 1.637413, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28244 [INFO] CHASSIS_PID_TURN, Time: 9298, Actual_Vol1: -1571.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 1096.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7737.115916, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.825344, Absolute Angle: 1.637417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011105
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28244 [INFO] CHASSIS_PID_TURN, Time: 9308, Actual_Vol1: -1577.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7698.915362, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820055, Absolute Angle: 1.637325, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014973
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28246 [INFO] CHASSIS_PID_TURN, Time: 9318, Actual_Vol1: -1577.000000, Actual_Vol2: 1460.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 1047.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7660.746233, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.816913, Absolute Angle: 1.637270, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28246 [INFO] CHASSIS_PID_TURN, Time: 9328, Actual_Vol1: -1620.000000, Actual_Vol2: 1540.000000, Actual_Vol3: -1072.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7622.538661, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820757, Absolute Angle: 1.637337, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28246 [INFO] CHASSIS_PID_TURN, Time: 9338, Actual_Vol1: -1657.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1146.000000, Actual_Vol4: 1103.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7584.335960, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.820270, Absolute Angle: 1.637328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018116
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28248 [INFO] CHASSIS_PID_TURN, Time: 9348, Actual_Vol1: -1669.000000, Actual_Vol2: 1589.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7546.201214, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.813475, Absolute Angle: 1.637210, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021554
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28248 [INFO] CHASSIS_PID_TURN, Time: 9358, Actual_Vol1: -1663.000000, Actual_Vol2: 1632.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7508.152231, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.804898, Absolute Angle: 1.637060, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030130
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28248 [INFO] CHASSIS_PID_TURN, Time: 9368, Actual_Vol1: -1712.000000, Actual_Vol2: 1669.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1226.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7470.125382, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802685, Absolute Angle: 1.637021, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032344
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28250 [INFO] CHASSIS_PID_TURN, Time: 9378, Actual_Vol1: -1743.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1244.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7432.099997, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.802539, Absolute Angle: 1.637019, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.032490
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28250 [INFO] CHASSIS_PID_TURN, Time: 9388, Actual_Vol1: -1756.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7394.159516, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.794048, Absolute Angle: 1.636871, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.040981
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28250 [INFO] CHASSIS_PID_TURN, Time: 9398, Actual_Vol1: -1793.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7356.253926, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.790559, Absolute Angle: 1.636810, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044470
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28252 [INFO] CHASSIS_PID_TURN, Time: 9408, Actual_Vol1: -1842.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1337.000000, Actual_Vol4: 1324.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7318.380692, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.787323, Absolute Angle: 1.636753, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045530
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28252 [INFO] CHASSIS_PID_TURN, Time: 9418, Actual_Vol1: -1854.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7280.530860, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.784983, Absolute Angle: 1.636712, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.044649
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 8.600000
28252 [INFO] CHASSIS_PID_TURN, Time: 9428, Actual_Vol1: -1860.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7242.752476, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.777838, Absolute Angle: 1.636588, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28254 [INFO] CHASSIS_PID_TURN, Time: 9438, Actual_Vol1: -1922.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7204.964227, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.778825, Absolute Angle: 1.636605, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28254 [INFO] CHASSIS_PID_TURN, Time: 9448, Actual_Vol1: -1965.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1423.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7167.167834, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.779639, Absolute Angle: 1.636619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051794
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28254 [INFO] CHASSIS_PID_TURN, Time: 9458, Actual_Vol1: -1971.000000, Actual_Vol2: 1774.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7129.412958, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.775488, Absolute Angle: 1.636547, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054145
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28256 [INFO] CHASSIS_PID_TURN, Time: 9468, Actual_Vol1: -1965.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1540.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7091.709969, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.770299, Absolute Angle: 1.636456, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056316
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28256 [INFO] CHASSIS_PID_TURN, Time: 9478, Actual_Vol1: -2082.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1558.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7054.005175, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.770479, Absolute Angle: 1.636459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.055045
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28256 [INFO] CHASSIS_PID_TURN, Time: 9488, Actual_Vol1: -2144.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 7016.319773, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.768540, Absolute Angle: 1.636426, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.056804
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28258 [INFO] CHASSIS_PID_TURN, Time: 9498, Actual_Vol1: -2051.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1595.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6978.689419, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.763035, Absolute Angle: 1.636329, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.057722
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28258 [INFO] CHASSIS_PID_TURN, Time: 9508, Actual_Vol1: -2119.000000, Actual_Vol2: 1947.000000, Actual_Vol3: -1675.000000, Actual_Vol4: 1583.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6941.090534, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.759889, Absolute Angle: 1.636275, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.060869
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 23.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28258 [INFO] CHASSIS_PID_TURN, Time: 9548, Actual_Vol1: -2372.000000, Actual_Vol2: 2242.000000, Actual_Vol3: -1749.000000, Actual_Vol4: 1854.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6791.222618, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.740561, Absolute Angle: 1.635937, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.064337
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000
28262 [INFO] CHASSIS_PID_TURN, Time: 9558, Actual_Vol1: -2384.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1762.000000, Actual_Vol4: 1873.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6753.883455, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.733916, Absolute Angle: 1.635821, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.068769
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000
    28262 [INFO] CHASSIS_PID_TURN, Time: 9568, Actual_Vol1: -2384.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1866.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6716.564862, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.731859, Absolute Angle: 1.635785, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.070679
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 0.000000
    28262 [INFO] CHASSIS_PID_TURN, Time: 9578, Actual_Vol1: -2445.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1811.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6679.228762, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.733610, Absolute Angle: 1.635816, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.062189
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 7.200000
    28264 [INFO] CHASSIS_PID_TURN, Time: 9588, Actual_Vol1: -2464.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6641.997507, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.723126, Absolute Angle: 1.635633, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.067433
, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 7.200000
    28264 [INFO] CHASSIS_PID_TURN, Time: 9598, Actual_Vol1: -2470.000000, Actual_Vol2: 2409.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6604.854088, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.714342, Absolute Angle: 1.635480, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072982
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
    28264 [INFO] CHASSIS_PID_TURN, Time: 9608, Actual_Vol1: -2464.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1780.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6567.803698, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.705039, Absolute Angle: 1.635317, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.079944
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
    28266 [INFO] CHASSIS_PID_TURN, Time: 9618, Actual_Vol1: -2458.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 1836.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6530.813847, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.698985, Absolute Angle: 1.635212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.080654
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: -0.000000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
    28266 [INFO] CHASSIS_PID_TURN, Time: 9628, Actual_Vol1: -2458.000000, Actual_Vol2: 2464.000000, Actual_Vol3: -2033.000000, Actual_Vol4: 1866.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6494.001859, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.681199, Absolute Angle: 1.634901, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098441
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
    28266 [INFO] CHASSIS_PID_TURN, Time: 9638, Actual_Vol1: -2464.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6457.294575, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -163.000000, Heading_Sp: 90.000000, Relative_Heading: 93.670728, Absolute Angle: 1.634718, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.108911
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -40.400000, Actual_Vel4: 8.800000
    28268 [INFO] CHASSIS_PID_TURN, Time: 9648, Actual_Vol1: -2464.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6420.712424, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.658215, Absolute Angle: 1.634500, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.117273
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -14.400000, Actual_Vel4: 8.800000
    28268 [INFO] CHASSIS_PID_TURN, Time: 9658, Actual_Vol1: -2464.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2057.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6385.265999, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.544642, Absolute Angle: 1.632518, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.225837
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 40.800000, Actual_Vel3: -14.400000, Actual_Vel4: 10.000000
    28268 [INFO] CHASSIS_PID_TURN, Time: 9668, Actual_Vol1: -2464.000000, Actual_Vol2: 2464.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1996.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6350.005535, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.526046, Absolute Angle: 1.632193, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.244433
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
    28270 [INFO] CHASSIS_PID_TURN, Time: 9678, Actual_Vol1: -2599.000000, Actual_Vol2: 2452.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1990.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6314.960514, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -162.000000, Heading_Sp: 90.000000, Relative_Heading: 93.504502, Absolute Angle: 1.631817, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.264038
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.000000
    28270 [INFO] CHASSIS_PID_TURN, Time: 9688, Actual_Vol1: -2636.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 2002.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6280.120484, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.484003, Absolute Angle: 1.631459, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.279032
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
    28270 [INFO] CHASSIS_PID_TURN, Time: 9698, Actual_Vol1: -2649.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 1811.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6246.528634, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.359185, Absolute Angle: 1.629281, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.400704
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
    28272 [INFO] CHASSIS_PID_TURN, Time: 9708, Actual_Vol1: -2753.000000, Actual_Vol2: 2513.000000, Actual_Vol3: -2051.000000, Actual_Vol4: 1774.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6213.235072, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.329356, Absolute Angle: 1.628760, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.425598
, over slew: 0, Actual_Vel1: -97.600000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
    28272 [INFO] CHASSIS_PID_TURN, Time: 9718, Actual_Vol1: -2698.000000, Actual_Vol2: 2507.000000, Actual_Vol3: -2008.000000, Actual_Vol4: 1848.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6180.200307, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.303476, Absolute Angle: 1.628309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.443752
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
    28272 [INFO] CHASSIS_PID_TURN, Time: 9728, Actual_Vol1: -2513.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2045.000000, Actual_Vol4: 1959.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6147.399769, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.280054, Absolute Angle: 1.627900, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.463993
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -20.000000, Actual_Vel4: 12.400000
    28274 [INFO] CHASSIS_PID_TURN, Time: 9738, Actual_Vol1: -2482.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6114.828538, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.257123, Absolute Angle: 1.627500, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.483438
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
    28274 [INFO] CHASSIS_PID_TURN, Time: 9748, Actual_Vol1: -2612.000000, Actual_Vol2: 2569.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6082.525094, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.230344, Absolute Angle: 1.627032, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.503572
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 24.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
    28274 [INFO] CHASSIS_PID_TURN, Time: 9758, Actual_Vol1: -2643.000000, Actual_Vol2: 2643.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1947.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6050.522356, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.200274, Absolute Angle: 1.626507, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.533336
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -21.800000, Actual_Vel4: 16.800000
    28276 [INFO] CHASSIS_PID_TURN, Time: 9768, Actual_Vol1: -2636.000000, Actual_Vol2: 2593.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 6018.795426, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.172693, Absolute Angle: 1.626026, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.560917
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
    28276 [INFO] CHASSIS_PID_TURN, Time: 9778, Actual_Vol1: -2649.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1879.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5987.352611, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.144281, Absolute Angle: 1.625530, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.578844
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
    28276 [INFO] CHASSIS_PID_TURN, Time: 9788, Actual_Vol1: -2649.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 1793.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5956.168541, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 171.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.118407, Absolute Angle: 1.625079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.595935
, over slew: 0, Actual_Vel1: -42.200000, Actual_Vel2: 12.800000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
    28278 [INFO] CHASSIS_PID_TURN, Time: 9798, Actual_Vol1: -2655.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2082.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5925.250205, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.091834, Absolute Angle: 1.624615, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.613205
, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.000000, Actual_Vel3: -37.000000, Actual_Vel4: 20.800000
    28278 [INFO] CHASSIS_PID_TURN, Time: 9808, Actual_Vol1: -2643.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1940.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5895.534956, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.971525, Absolute Angle: 1.622515, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.727460
, over slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: 10.000000, Actual_Vel3: -38.200000, Actual_Vel4: 20.800000
    28278 [INFO] CHASSIS_PID_TURN, Time: 9818, Actual_Vol1: -2519.000000, Actual_Vol2: 1608.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5866.000235, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.953472, Absolute Angle: 1.622200, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.727727
, over slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: 10.000000, Actual_Vel3: -38.200000, Actual_Vel4: 42.400000
    28280 [INFO] CHASSIS_PID_TURN, Time: 9828, Actual_Vol1: -690.000000, Actual_Vol2: 388.000000, Actual_Vol3: -524.000000, Actual_Vol4: 468.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5836.624652, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.937558, Absolute Angle: 1.621922, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.733170, ov
er slew: 0, Actual_Vel1: -13.200000, Actual_Vel2: -0.000000, Actual_Vel3: -38.200000, Actual_Vel4: 42.400000
    28280 [INFO] CHASSIS_PID_TURN, Time: 9838, Actual_Vol1: -154.000000, Actual_Vol2: 179.000000, Actual_Vol3: -105.000000, Actual_Vol4: 191.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5807.398728, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.922592, Absolute Angle: 1.621661, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.735623, ov
er slew: 0, Actual_Vel1: -5.000000, Actual_Vel2: -0.000000, Actual_Vel3: -8.800000, Actual_Vel4: 9.600000
    28280 [INFO] CHASSIS_PID_TURN, Time: 9848, Actual_Vol1: -376.000000, Actual_Vol2: 234.000000, Actual_Vol3: -25.000000, Actual_Vol4: 25.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5778.172805, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.922592, Absolute Angle: 1.621661, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.622050, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
    28282 [INFO] CHASSIS_PID_TURN, Time: 9858, Actual_Vol1: -480.000000, Actual_Vol2: 320.000000, Actual_Vol3: -80.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5748.850784, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.932202, Absolute Angle: 1.621829, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.603454, over sl
ew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
    28282 [INFO] CHASSIS_PID_TURN, Time: 9868, Actual_Vol1: -511.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5719.286856, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.956393, Absolute Angle: 1.622251, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.581910, over sl
ew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 9.600000
    28282 [INFO] CHASSIS_PID_TURN, Time: 9878, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -92.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5689.438042, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984881, Absolute Angle: 1.622748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.561411, over sl
ew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
    28284 [INFO] CHASSIS_PID_TURN, Time: 9888, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5659.361393, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007665, Absolute Angle: 1.623146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.436593, over sl
ew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
    28284 [INFO] CHASSIS_PID_TURN, Time: 9898, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5629.179734, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.018166, Absolute Angle: 1.623329, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.406764, over sl
ew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 4.000000, Actual_Vel4: 0.000000
    28284 [INFO] CHASSIS_PID_TURN, Time: 9908, Actual_Vol1: -517.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5598.977349, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.020239, Absolute Angle: 1.623365, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.380884, over sl
ew: 0, Actual_Vel1: 10.200000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
    28286 [INFO] CHASSIS_PID_TURN, Time: 9918, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -99.000000, Actual_Vol4: 6.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5568.800803, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.017655, Absolute Angle: 1.623320, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.357461, over sl
ew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
    28286 [INFO] CHASSIS_PID_TURN, Time: 9928, Actual_Vol1: -524.000000, Actual_Vol2: 370.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5538.670154, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.013065, Absolute Angle: 1.623240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.334531, over sl
ew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 9.400000, Actual_Vel4: -17.800000
    28286 [INFO] CHASSIS_PID_TURN, Time: 9938, Actual_Vol1: -517.000000, Actual_Vol2: 400.000000, Actual_Vol3: -92.000000, Actual_Vol4: 0.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5508.659824, kD: 35.000000, kI: 0.00
```

```
0700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001033, Absolute Angle: 1.623030, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.307752, over slew: 0, Actual_Vel1: 45.000000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000
    816:   28288 [INFO] CHASSIS_PID_TURN, Time: 9948, Actual_Vol1: -591.000000, Actual_Vol2: 444.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5478.744807, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991502, Absolute Angle: 1.622864, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.277681, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000
    817:   28288 [INFO] CHASSIS_PID_TURN, Time: 9958, Actual_Vol1: -610.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5448.801152, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.994365, Absolute Angle: 1.622914, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.250101, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: 15.800000, Actual_Vel4: -17.800000
    818:   28288 [INFO] CHASSIS_PID_TURN, Time: 9968, Actual_Vol1: -604.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5418.822919, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997823, Absolute Angle: 1.622974, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.221689, over slew: 0, Actual_Vel1: 24.200000, Actual_Vel2: -179.800000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    819:   28290 [INFO] CHASSIS_PID_TURN, Time: 9978, Actual_Vol1: -616.000000, Actual_Vol2: 493.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5388.816034, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.000689, Absolute Angle: 1.623024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.195815, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    820:   28290 [INFO] CHASSIS_PID_TURN, Time: 9988, Actual_Vol1: -678.000000, Actual_Vol2: 536.000000, Actual_Vol3: -203.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5358.829322, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998671, Absolute Angle: 1.622989, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.169241, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    821:   28290 [INFO] CHASSIS_PID_TURN, Time: 9998, Actual_Vol1: -702.000000, Actual_Vol2: 536.000000, Actual_Vol3: -216.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5328.843241, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998608, Absolute Angle: 1.622988, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    822:   28292 [INFO] CHASSIS_PID_TURN, Time: 10008, Actual_Vol1: -702.000000, Actual_Vol2: 542.000000, Actual_Vol3: -222.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5298.832414, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001083, Absolute Angle: 1.623031, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    823:   28292 [INFO] CHASSIS_PID_TURN, Time: 10018, Actual_Vol1: -696.000000, Actual_Vol2: 579.000000, Actual_Vol3: -222.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5268.782984, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004943, Absolute Angle: 1.623098, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    824:   28292 [INFO] CHASSIS_PID_TURN, Time: 10028, Actual_Vol1: -875.000000, Actual_Vol2: 628.000000, Actual_Vol3: -290.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5238.708014, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007497, Absolute Angle: 1.623143, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    825:   28294 [INFO] CHASSIS_PID_TURN, Time: 10038, Actual_Vol1: -912.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5208.611632, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009638, Absolute Angle: 1.623180, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.097646, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    826:   28294 [INFO] CHASSIS_PID_TURN, Time: 10048, Actual_Vol1: -918.000000, Actual_Vol2: 641.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5178.544147, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.006749, Absolute Angle: 1.623130, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.088036, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    827:   28294 [INFO] CHASSIS_PID_TURN, Time: 10058, Actual_Vol1: -918.000000, Actual_Vol2: 678.000000, Actual_Vol3: -308.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5148.438763, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.010538, Absolute Angle: 1.623196, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.063846, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    828:   28296 [INFO] CHASSIS_PID_TURN, Time: 10068, Actual_Vol1: -992.000000, Actual_Vol2: 715.000000, Actual_Vol3: -413.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5118.346091, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009267, Absolute Angle: 1.623174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035357, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    829:   28296 [INFO] CHASSIS_PID_TURN, Time: 10078, Actual_Vol1: -1016.000000, Actual_Vol2: 764.000000, Actual_Vol3: -431.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5088.216881, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.012921, Absolute Angle: 1.623237, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    830:   28296 [INFO] CHASSIS_PID_TURN, Time: 10088, Actual_Vol1: -1023.000000, Actual_Vol2: 758.000000, Actual_Vol3: -444.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5058.053537, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.016334, Absolute Angle: 1.623297, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    831:   28298 [INFO] CHASSIS_PID_TURN, Time: 10098, Actual_Vol1: -1023.000000, Actual_Vol2: 844.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5027.903553, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.014998, Absolute Angle: 1.623274, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028737, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    832:   28298 [INFO] CHASSIS_PID_TURN, Time: 10108, Actual_Vol1: -1096.000000, Actual_Vol2: 930.000000, Actual_Vol3: -431.000000, Actual_Vol4: 320.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4997.752478, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.015107, Absolute Angle: 1.623276, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026153, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    833:   28298 [INFO] CHASSIS_PID_TURN, Time: 10118, Actual_Vol1: -1140.000000, Actual_Vol2: 942.000000, Actual_Vol3: -511.000000, Actual_Vol4: 394.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4967.607023, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.014545, Absolute Angle: 1.623266, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    834:   28300 [INFO] CHASSIS_PID_TURN, Time: 10128, Actual_Vol1: -1152.000000, Actual_Vol2: 949.000000, Actual_Vol3: -517.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4937.507901, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.009912, Absolute Angle: 1.623185, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    835:   28300 [INFO] CHASSIS_PID_TURN, Time: 10138, Actual_Vol1: -1146.000000, Actual_Vol2: 992.000000, Actual_Vol3: -524.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4907.455496, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.005241, Absolute Angle: 1.623103, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024833, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    836:   28300 [INFO] CHASSIS_PID_TURN, Time: 10148, Actual_Vol1: -1226.000000, Actual_Vol2: 1035.000000, Actual_Vol3: -524.000000, Actual_Vol4: 444.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4877.416506, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.003899, Absolute Angle: 1.623080, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021969, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    837:   28302 [INFO] CHASSIS_PID_TURN, Time: 10158, Actual_Vol1: -1244.000000, Actual_Vol2: 1047.000000, Actual_Vol3: -598.000000, Actual_Vol4: 480.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4847.394118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.002239, Absolute Angle: 1.623051, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018511, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    838:   28302 [INFO] CHASSIS_PID_TURN, Time: 10168, Actual_Vol1: -1238.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -610.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4817.399702, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.999442, Absolute Angle: 1.623002, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    839:   28302 [INFO] CHASSIS_PID_TURN, Time: 10178, Actual_Vol1: -1250.000000, Actual_Vol2: 1084.000000, Actual_Vol3: -616.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4787.380155, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.001955, Absolute Angle: 1.623046, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    840:   28304 [INFO] CHASSIS_PID_TURN, Time: 10188, Actual_Vol1: -1312.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -616.000000, Actual_Vol4: 530.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4757.346402, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.003375, Absolute Angle: 1.623071, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017726, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    841:   28304 [INFO] CHASSIS_PID_TURN, Time: 10198, Actual_Vol1: -1337.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -684.000000, Actual_Vol4: 573.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4727.299867, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004654, Absolute Angle: 1.623093, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    842:   28304 [INFO] CHASSIS_PID_TURN, Time: 10208, Actual_Vol1: -1324.000000, Actual_Vol2: 1170.000000, Actual_Vol3: -708.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4697.212153, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.008771, Absolute Angle: 1.623165, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    843:   28306 [INFO] CHASSIS_PID_TURN, Time: 10218, Actual_Vol1: -1337.000000, Actual_Vol2: 1214.000000, Actual_Vol3: -702.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4667.167192, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004496, Absolute Angle: 1.623090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    844:   28306 [INFO] CHASSIS_PID_TURN, Time: 10228, Actual_Vol1: -1411.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -708.000000, Actual_Vol4: 628.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4637.126608, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004058, Absolute Angle: 1.623083, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    845:   28306 [INFO] CHASSIS_PID_TURN, Time: 10238, Actual_Vol1: -1423.000000, Actual_Vol2: 1269.000000, Actual_Vol3: -887.000000, Actual_Vol4: 665.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4607.050075, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.007653, Absolute Angle: 1.623146, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    846:   28308 [INFO] CHASSIS_PID_TURN, Time: 10248, Actual_Vol1: -1435.000000, Actual_Vol2: 1263.000000, Actual_Vol3: -924.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4577.001531, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.004854, Absolute Angle: 1.623097, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    847:   28308 [INFO] CHASSIS_PID_TURN, Time: 10258, Actual_Vol1: -1435.000000, Actual_Vol2: 1306.000000, Actual_Vol3: -924.000000, Actual_Vol4: 708.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4546.978236, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.002330, Absolute Angle: 1.623053, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    848:   28308 [INFO] CHASSIS_PID_TURN, Time: 10268, Actual_Vol1: -1546.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -930.000000, Actual_Vol4: 795.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4516.981087, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.999715, Absolute Angle: 1.623007, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.016893, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    849:   28310 [INFO] CHASSIS_PID_TURN, Time: 10278, Actual_Vol1: -1565.000000, Actual_Vol2: 1349.000000, Actual_Vol3: -1004.000000, Actual_Vol4: 887.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4487.001906, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997918, Absolute Angle: 1.622976, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018416, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    850:   28310 [INFO] CHASSIS_PID_TURN, Time: 10288, Actual_Vol1: -1577.000000, Actual_Vol2: 1355.000000, Actual_Vol3: -1023.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4457.083575, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991833, Absolute Angle: 1.622869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023274, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    851:   28310 [INFO] CHASSIS_PID_TURN, Time: 10298, Actual_Vol1: -1577.000000, Actual_Vol2: 1398.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4427.166534, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991704, Absolute Angle: 1.622867, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023403, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    852:   28312 [INFO] CHASSIS_PID_TURN, Time: 10308, Actual_Vol1: -1651.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 942.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4397.216054, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.995048, Absolute Angle: 1.622926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022841, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
```

```
28312 [INFO] CHASSIS_PID_TURN, Time: 10318, Actual_Vol1: -1669.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1096.000000, Actual_Vol4: 992.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4367.297690, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.991836, Absolute Angle: 1.622869, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018208, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28312 [INFO] CHASSIS_PID_TURN, Time: 10328, Actual_Vol1: -1675.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4337.347626, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.995006, Absolute Angle: 1.622925, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28314 [INFO] CHASSIS_PID_TURN, Time: 10338, Actual_Vol1: -1675.000000, Actual_Vol2: 1534.000000, Actual_Vol3: -1047.000000, Actual_Vol4: 1023.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4307.376159, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.997147, Absolute Angle: 1.622962, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28314 [INFO] CHASSIS_PID_TURN, Time: 10348, Actual_Vol1: -1743.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1103.000000, Actual_Vol4: 1072.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4277.369182, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 93.000698, Absolute Angle: 1.623024, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28314 [INFO] CHASSIS_PID_TURN, Time: 10358, Actual_Vol1: -1756.000000, Actual_Vol2: 1595.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4247.381476, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.998771, Absolute Angle: 1.622991, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28316 [INFO] CHASSIS_PID_TURN, Time: 10368, Actual_Vol1: -1756.000000, Actual_Vol2: 1589.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1158.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4217.443767, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.993771, Absolute Angle: 1.622903, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.017067, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28316 [INFO] CHASSIS_PID_TURN, Time: 10378, Actual_Vol1: -1799.000000, Actual_Vol2: 1632.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4187.541115, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.990265, Absolute Angle: 1.622842, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28316 [INFO] CHASSIS_PID_TURN, Time: 10388, Actual_Vol1: -1848.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1232.000000, Actual_Vol4: 1195.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4157.609994, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.993112, Absolute Angle: 1.622892, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28318 [INFO] CHASSIS_PID_TURN, Time: 10398, Actual_Vol1: -1854.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4127.704872, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.990512, Absolute Angle: 1.622846, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018506, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28318 [INFO] CHASSIS_PID_TURN, Time: 10408, Actual_Vol1: -1854.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4097.847226, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.985765, Absolute Angle: 1.622764, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.021889, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28318 [INFO] CHASSIS_PID_TURN, Time: 10418, Actual_Vol1: -1928.000000, Actual_Vol2: 1719.000000, Actual_Vol3: -1257.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4068.000169, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984706, Absolute Angle: 1.622745, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022948, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28320 [INFO] CHASSIS_PID_TURN, Time: 10428, Actual_Vol1: -1965.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1331.000000, Actual_Vol4: 1300.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4038.215873, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.978430, Absolute Angle: 1.622635, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029224, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28320 [INFO] CHASSIS_PID_TURN, Time: 10438, Actual_Vol1: -1940.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 4008.369396, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984648, Absolute Angle: 1.622744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026425, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -7.000000, Actual_Vel4: 0.000000
28320 [INFO] CHASSIS_PID_TURN, Time: 10448, Actual_Vol1: -1866.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3978.523195, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984620, Absolute Angle: 1.622744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023900, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28322 [INFO] CHASSIS_PID_TURN, Time: 10458, Actual_Vol1: -1934.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1355.000000, Actual_Vol4: 1337.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3948.677935, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.984526, Absolute Angle: 1.622742, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28322 [INFO] CHASSIS_PID_TURN, Time: 10468, Actual_Vol1: -1965.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1429.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3918.827226, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.985071, Absolute Angle: 1.622751, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28322 [INFO] CHASSIS_PID_TURN, Time: 10478, Actual_Vol1: -1977.000000, Actual_Vol2: 1879.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1435.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3889.000529, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.982670, Absolute Angle: 1.622709, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.022268, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28324 [INFO] CHASSIS_PID_TURN, Time: 10488, Actual_Vol1: -2101.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1441.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3859.224650, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.977588, Absolute Angle: 1.622621, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023110, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28324 [INFO] CHASSIS_PID_TURN, Time: 10498, Actual_Vol1: -2224.000000, Actual_Vol2: 1983.000000, Actual_Vol3: -1558.000000, Actual_Vol4: 1485.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3829.483416, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.974123, Absolute Angle: 1.622560, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026574, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28324 [INFO] CHASSIS_PID_TURN, Time: 10508, Actual_Vol1: -2261.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3799.753510, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.972991, Absolute Angle: 1.622541, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027707, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28326 [INFO] CHASSIS_PID_TURN, Time: 10518, Actual_Vol1: -2267.000000, Actual_Vol2: 1959.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1540.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3770.060069, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.969344, Absolute Angle: 1.622477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.031354, over slew: 0, Actual_Vel1: -16.600000, Actual_Vel2: 98.600000, Actual_Vel3: -0.000000, Actual_Vel4: 218.200000
28326 [INFO] CHASSIS_PID_TURN, Time: 10528, Actual_Vol1: -2335.000000, Actual_Vol2: 1953.000000, Actual_Vol3: -1577.000000, Actual_Vol4: 1454.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3740.411832, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.964824, Absolute Angle: 1.622398, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035874, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -0.000000, Actual_Vel4: 218.200000
28326 [INFO] CHASSIS_PID_TURN, Time: 10538, Actual_Vol1: -2341.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1725.000000, Actual_Vol4: 1485.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3710.841625, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.957021, Absolute Angle: 1.622262, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043677, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28328 [INFO] CHASSIS_PID_TURN, Time: 10548, Actual_Vol1: -2267.000000, Actual_Vol2: 1996.000000, Actual_Vol3: -1626.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3681.358118, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.948351, Absolute Angle: 1.622111, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050420, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28328 [INFO] CHASSIS_PID_TURN, Time: 10558, Actual_Vol1: -2335.000000, Actual_Vol2: 2107.000000, Actual_Vol3: -1595.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3651.954690, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.940343, Absolute Angle: 1.621971, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.053428, over slew: 0, Actual_Vel1: -11.800000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28328 [INFO] CHASSIS_PID_TURN, Time: 10568, Actual_Vol1: -2372.000000, Actual_Vol2: 2230.000000, Actual_Vol3: -1725.000000, Actual_Vol4: 1657.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3622.606995, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.934770, Absolute Angle: 1.621873, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.058343, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28330 [INFO] CHASSIS_PID_TURN, Time: 10578, Actual_Vol1: -2372.000000, Actual_Vol2: 2261.000000, Actual_Vol3: -1756.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3593.359998, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.924700, Absolute Angle: 1.621698, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.068412, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28330 [INFO] CHASSIS_PID_TURN, Time: 10588, Actual_Vol1: -2378.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1756.000000, Actual_Vol4: 1756.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3564.150596, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.920940, Absolute Angle: 1.621632, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.069572, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28330 [INFO] CHASSIS_PID_TURN, Time: 10598, Actual_Vol1: -2409.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1799.000000, Actual_Vol4: 1768.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3534.992946, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.915765, Absolute Angle: 1.621542, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.070000, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28332 [INFO] CHASSIS_PID_TURN, Time: 10608, Actual_Vol1: -2458.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1848.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3505.924751, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.906819, Absolute Angle: 1.621386, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.078251, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28332 [INFO] CHASSIS_PID_TURN, Time: 10618, Actual_Vol1: -2470.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1854.000000, Actual_Vol4: 1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3476.934234, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899052, Absolute Angle: 1.621250, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28332 [INFO] CHASSIS_PID_TURN, Time: 10628, Actual_Vol1: -2470.000000, Actual_Vol2: 2433.000000, Actual_Vol3: -1934.000000, Actual_Vol4: 1928.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3447.939593, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899464, Absolute Angle: 1.621257, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28334 [INFO] CHASSIS_PID_TURN, Time: 10638, Actual_Vol1: -2470.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -2020.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3418.920448, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.901914, Absolute Angle: 1.621300, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -65.000000, Actual_Vel4: 218.200000
28334 [INFO] CHASSIS_PID_TURN, Time: 10648, Actual_Vol1: -2458.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1983.000000, Actual_Vol4: 2051.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3389.921917, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.899853, Absolute Angle: 1.621264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.086019, over slew: 0, Actual_Vel1: -9.200000, Actual_Vel2: 98.600000, Actual_Vel3: -14.200000, Actual_Vel4: 218.200000
28334 [INFO] CHASSIS_PID_TURN, Time: 10658, Actual_Vol1: -2470.000000, Actual_Vol2: 2482.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3360.950346, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.897157, Absolute Angle: 1.621217, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.087914, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 8.000000
28336 [INFO] CHASSIS_PID_TURN, Time: 10668, Actual_Vol1: -2544.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -1971.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3332.051526, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.889882, Absolute Angle: 1.621090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.092788, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.200000, Actual_Vel4: 8.000000
28336 [INFO] CHASSIS_PID_TURN, Time: 10678, Actual_Vol1: -2624.000000, Actual_Vol2: 2489.000000, Actual_Vol3: -2057.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3303.204743, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.884678, Absolute Angle: 1.620999, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.092910, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 8.000000
28336 [INFO] CHASSIS_PID_TURN, Time: 10688, Actual_Vol1: -2636.000000, Actual_Vol2: 2482.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3274.465916, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.873883, Absolute Angle: 1.620811, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.10024
```

```
1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 8.000000
    891:   28338 [INFO] CHASSIS_PID_TURN, Time: 10698, Actual_Vol1: -2717.000000, Actual_Vol2: 2612.000000, Actual_Vol3: -1990.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3245.901738, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.856418, Absolute Angle: 1.620506, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.11657
3, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 14.200000
    892:   28338 [INFO] CHASSIS_PID_TURN, Time: 10708, Actual_Vol1: -2723.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3217.480992, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.842075, Absolute Angle: 1.620256, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.12726
9, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: -0.000000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    893:   28338 [INFO] CHASSIS_PID_TURN, Time: 10718, Actual_Vol1: -2649.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3189.241403, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.823959, Absolute Angle: 1.619939, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.14086
5, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    894:   28340 [INFO] CHASSIS_PID_TURN, Time: 10728, Actual_Vol1: -2643.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1983.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3161.121995, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -161.000000, Heading_Sp: 90.000000, Relative_Heading: 92.811941, Absolute Angle: 1.619730, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.14508
0, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    895:   28340 [INFO] CHASSIS_PID_TURN, Time: 10738, Actual_Vol1: -2704.000000, Actual_Vol2: 2643.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3133.175294, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.794670, Absolute Angle: 1.619428, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.15368
1, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    896:   28340 [INFO] CHASSIS_PID_TURN, Time: 10748, Actual_Vol1: -2797.000000, Actual_Vol2: 2667.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2008.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3106.339340, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.683595, Absolute Angle: 1.617490, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.25674
7, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    897:   28342 [INFO] CHASSIS_PID_TURN, Time: 10758, Actual_Vol1: -2846.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2070.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3079.633009, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.670633, Absolute Angle: 1.617263, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.26413
6, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 7.800000, Actual_Vel3: -14.400000, Actual_Vel4: 13.600000
    898:   28342 [INFO] CHASSIS_PID_TURN, Time: 10768, Actual_Vol1: -2846.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -2150.000000, Actual_Vol4: 2039.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3053.101534, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -160.000000, Heading_Sp: 90.000000, Relative_Heading: 92.653148, Absolute Angle: 1.616958, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.27155
2, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.200000, Actual_Vel3: -14.400000, Actual_Vel4: 21.000000
    899:   28342 [INFO] CHASSIS_PID_TURN, Time: 10778, Actual_Vol1: -2858.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2230.000000, Actual_Vol4: 1990.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3026.763809, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.633772, Absolute Angle: 1.616620, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.28716
8, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 10.200000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
    900:   28344 [INFO] CHASSIS_PID_TURN, Time: 10788, Actual_Vol1: -2864.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 1977.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 3001.576437, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.518737, Absolute Angle: 1.614612, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.39702
8, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
    901:   28344 [INFO] CHASSIS_PID_TURN, Time: 10798, Actual_Vol1: -2852.000000, Actual_Vol2: 2544.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2976.553486, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.502295, Absolute Angle: 1.614325, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.40452
4, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -76.000000, Actual_Vel4: 17.000000
    902:   28344 [INFO] CHASSIS_PID_TURN, Time: 10808, Actual_Vol1: -2895.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2248.000000, Actual_Vol4: 1928.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2951.734750, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.481874, Absolute Angle: 1.613969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.42004
1, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 12.400000
    903:   28346 [INFO] CHASSIS_PID_TURN, Time: 10818, Actual_Vol1: -2926.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2322.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2927.123046, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.461170, Absolute Angle: 1.613608, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.44074
4, over slew: 0, Actual_Vel1: -16.200000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
    904:   28346 [INFO] CHASSIS_PID_TURN, Time: 10828, Actual_Vol1: -2938.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 1731.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2902.697655, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.442539, Absolute Angle: 1.613282, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.45937
5, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.000000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
    905:   28346 [INFO] CHASSIS_PID_TURN, Time: 10838, Actual_Vol1: -2920.000000, Actual_Vol2: 2704.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 1836.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2878.388887, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.430877, Absolute Angle: 1.613079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.46897
6, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -0.000000, Actual_Vel4: 13.200000
    906:   28348 [INFO] CHASSIS_PID_TURN, Time: 10848, Actual_Vol1: -2852.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 1860.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2854.201928, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.418696, Absolute Angle: 1.612866, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.47846
1, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -16.600000, Actual_Vel4: 13.200000
    907:   28348 [INFO] CHASSIS_PID_TURN, Time: 10858, Actual_Vol1: -2889.000000, Actual_Vol2: 2778.000000, Actual_Vol3: -2279.000000, Actual_Vol4: 1725.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2830.123628, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.407830, Absolute Angle: 1.612677, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.48205
2, over slew: 0, Actual_Vel1: -10.600000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
    908:   28348 [INFO] CHASSIS_PID_TURN, Time: 10868, Actual_Vol1: -2932.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1595.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2806.213571, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 170.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.391006, Absolute Angle: 1.612383, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.49367
3, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
    909:   28350 [INFO] CHASSIS_PID_TURN, Time: 10878, Actual_Vol1: -2895.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1700.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2782.513562, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.370001, Absolute Angle: 1.612016, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.50388
2, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
    910:   28350 [INFO] CHASSIS_PID_TURN, Time: 10888, Actual_Vol1: -2858.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1823.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2759.986142, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.252742, Absolute Angle: 1.609970, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.60367
6, over slew: 0, Actual_Vel1: -20.000000, Actual_Vel2: 9.800000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
    911:   28350 [INFO] CHASSIS_PID_TURN, Time: 10898, Actual_Vol1: -2852.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 1842.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2737.676877, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.230927, Absolute Angle: 1.609589, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.61114
8, over slew: 0, Actual_Vel1: -12.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 12.800000
    912:   28352 [INFO] CHASSIS_PID_TURN, Time: 10908, Actual_Vol1: -696.000000, Actual_Vol2: 702.000000, Actual_Vol3: -505.000000, Actual_Vol4: 382.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2715.568874, kD: 35.000000, kI:
 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.210800, Absolute Angle: 1.609238, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.613159, ov
er slew: 0, Actual_Vel1: -5.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 7.200000
    913:   28352 [INFO] CHASSIS_PID_TURN, Time: 10918, Actual_Vol1: -166.000000, Actual_Vol2: 209.000000, Actual_Vol3: -99.000000, Actual_Vol4: 74.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2693.628761, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.194011, Absolute Angle: 1.608945, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617929, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.400000, Actual_Vel3: -15.400000, Actual_Vel4: 7.200000
    914:   28352 [INFO] CHASSIS_PID_TURN, Time: 10928, Actual_Vol1: -277.000000, Actual_Vol2: 277.000000, Actual_Vol3: -62.000000, Actual_Vol4: 18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2671.717675, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.191109, Absolute Angle: 1.608894, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.603561, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: 7.200000
    915:   28354 [INFO] CHASSIS_PID_TURN, Time: 10938, Actual_Vol1: -413.000000, Actual_Vol2: 425.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2649.722701, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.199497, Absolute Angle: 1.609041, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.492487, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
    916:   28354 [INFO] CHASSIS_PID_TURN, Time: 10948, Actual_Vol1: -444.000000, Actual_Vol2: 450.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2627.550259, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.217244, Absolute Angle: 1.609350, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.479524, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
    917:   28354 [INFO] CHASSIS_PID_TURN, Time: 10958, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2605.145496, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.240476, Absolute Angle: 1.609756, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.462039, over
 slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -5.200000
    918:   28356 [INFO] CHASSIS_PID_TURN, Time: 10968, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2582.489090, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265641, Absolute Angle: 1.610195, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.442664, over
 slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: -0.000000, Actual_Vel4: -14.800000
    919:   28356 [INFO] CHASSIS_PID_TURN, Time: 10978, Actual_Vol1: -444.000000, Actual_Vol2: 456.000000, Actual_Vol3: -86.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2559.674045, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.281505, Absolute Angle: 1.610472, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.327629, over
 slew: 0, Actual_Vel1: 4.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
    920:   28356 [INFO] CHASSIS_PID_TURN, Time: 10988, Actual_Vol1: -437.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2536.808132, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.286591, Absolute Angle: 1.610561, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.311187, over
 slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
    921:   28358 [INFO] CHASSIS_PID_TURN, Time: 10998, Actual_Vol1: -437.000000, Actual_Vol2: 444.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2513.908787, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.289934, Absolute Angle: 1.610619, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.290765, over
 slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
    922:   28358 [INFO] CHASSIS_PID_TURN, Time: 11008, Actual_Vol1: -437.000000, Actual_Vol2: 456.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2491.052805, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.285598, Absolute Angle: 1.610543, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.270062, over
 slew: 0, Actual_Vel1: 25.000000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
    923:   28358 [INFO] CHASSIS_PID_TURN, Time: 11018, Actual_Vol1: -431.000000, Actual_Vol2: 487.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2468.269011, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.278379, Absolute Angle: 1.610417, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.251430, over
 slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -14.800000
    924:   28360 [INFO] CHASSIS_PID_TURN, Time: 11028, Actual_Vol1: -499.000000, Actual_Vol2: 536.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2445.586886, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268213, Absolute Angle: 1.610240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.239768, over
 slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
    925:   28360 [INFO] CHASSIS_PID_TURN, Time: 11038, Actual_Vol1: -530.000000, Actual_Vol2: 536.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2422.990649, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.259624, Absolute Angle: 1.610090, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.227587, over
 slew: 0, Actual_Vel1: 25.400000, Actual_Vel2: -8.800000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
    926:   28360 [INFO] CHASSIS_PID_TURN, Time: 11048, Actual_Vol1: -524.000000, Actual_Vol2: 542.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2400.375022, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.261563, Absolute Angle: 1.610124, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.216721, over
 slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 168.400000, Actual_Vel4: -8.800000
    927:   28362 [INFO] CHASSIS_PID_TURN, Time: 11058, Actual_Vol1: -524.000000, Actual_Vol2: 579.000000, Actual_Vol3: -154.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2377.732496, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264253, Absolute Angle: 1.610171, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.199897, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: 168.400000, Actual_Vel4: 0.000000
    928:   28362 [INFO] CHASSIS_PID_TURN, Time: 11068, Actual_Vol1: -598.000000, Actual_Vol2: 622.000000, Actual_Vol3: -222.000000, Actual_Vol4: 197.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2355.044444, kD: 35.000000, kI:
```

```
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268805, Absolute Angle: 1.610250, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.178892, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    929:   28362 [INFO] CHASSIS_PID_TURN, Time: 11078, Actual_Vol1: -610.000000, Actual_Vol2: 628.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2332.375137, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266931, Absolute Angle: 1.610217, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    930:   28364 [INFO] CHASSIS_PID_TURN, Time: 11088, Actual_Vol1: -616.000000, Actual_Vol2: 634.000000, Actual_Vol3: -228.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2309.657884, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271725, Absolute Angle: 1.610301, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    931:   28364 [INFO] CHASSIS_PID_TURN, Time: 11098, Actual_Vol1: -616.000000, Actual_Vol2: 671.000000, Actual_Vol3: -234.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2286.921791, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273609, Absolute Angle: 1.610334, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    932:   28364 [INFO] CHASSIS_PID_TURN, Time: 11108, Actual_Vol1: -622.000000, Actual_Vol2: 721.000000, Actual_Vol3: -265.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2264.216998, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.270479, Absolute Angle: 1.610279, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    933:   28366 [INFO] CHASSIS_PID_TURN, Time: 11118, Actual_Vol1: -684.000000, Actual_Vol2: 752.000000, Actual_Vol3: -302.000000, Actual_Vol4: 296.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2241.562151, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265485, Absolute Angle: 1.610192, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098826, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    934:   28366 [INFO] CHASSIS_PID_TURN, Time: 11128, Actual_Vol1: -702.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2218.878395, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268376, Absolute Angle: 1.610243, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.090437, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    935:   28366 [INFO] CHASSIS_PID_TURN, Time: 11138, Actual_Vol1: -702.000000, Actual_Vol2: 752.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2196.144145, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273425, Absolute Angle: 1.610331, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.072690, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    936:   28368 [INFO] CHASSIS_PID_TURN, Time: 11148, Actual_Vol1: -696.000000, Actual_Vol2: 832.000000, Actual_Vol3: -382.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2173.422322, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.272182, Absolute Angle: 1.610309, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.049458, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    937:   28368 [INFO] CHASSIS_PID_TURN, Time: 11158, Actual_Vol1: -887.000000, Actual_Vol2: 918.000000, Actual_Vol3: -419.000000, Actual_Vol4: 419.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2150.709711, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271261, Absolute Angle: 1.610293, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    938:   28368 [INFO] CHASSIS_PID_TURN, Time: 11168, Actual_Vol1: -918.000000, Actual_Vol2: 942.000000, Actual_Vol3: -431.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2127.971100, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273861, Absolute Angle: 1.610338, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    939:   28370 [INFO] CHASSIS_PID_TURN, Time: 11178, Actual_Vol1: -924.000000, Actual_Vol2: 949.000000, Actual_Vol3: -431.000000, Actual_Vol4: 431.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2105.259838, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271126, Absolute Angle: 1.610291, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    940:   28370 [INFO] CHASSIS_PID_TURN, Time: 11188, Actual_Vol1: -924.000000, Actual_Vol2: 986.000000, Actual_Vol3: -480.000000, Actual_Vol4: 425.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2082.598550, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266129, Absolute Angle: 1.610203, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.030311, ov
er slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    941:   28370 [INFO] CHASSIS_PID_TURN, Time: 11198, Actual_Vol1: -998.000000, Actual_Vol2: 1029.000000, Actual_Vol3: -511.000000, Actual_Vol4: 505.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2059.902346, kD: 35.000000, k
I: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.269620, Absolute Angle: 1.610264, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.025975, o
ver slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    942:   28372 [INFO] CHASSIS_PID_TURN, Time: 11208, Actual_Vol1: -1016.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2037.188879, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.271347, Absolute Angle: 1.610295, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.018756,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    943:   28372 [INFO] CHASSIS_PID_TURN, Time: 11218, Actual_Vol1: -1023.000000, Actual_Vol2: 1041.000000, Actual_Vol3: -517.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 2014.503224, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268565, Absolute Angle: 1.610246, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014237,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    944:   28372 [INFO] CHASSIS_PID_TURN, Time: 11228, Actual_Vol1: -1029.000000, Actual_Vol2: 1078.000000, Actual_Vol3: -561.000000, Actual_Vol4: 517.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1991.829857, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.267337, Absolute Angle: 1.610225, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.014237,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    945:   28374 [INFO] CHASSIS_PID_TURN, Time: 11238, Actual_Vol1: -1096.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -604.000000, Actual_Vol4: 598.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1969.141539, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268832, Absolute Angle: 1.610251, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012298,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    946:   28374 [INFO] CHASSIS_PID_TURN, Time: 11248, Actual_Vol1: -1146.000000, Actual_Vol2: 1158.000000, Actual_Vol3: -604.000000, Actual_Vol4: 604.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1946.408679, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.273286, Absolute Angle: 1.610328, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009608,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    947:   28374 [INFO] CHASSIS_PID_TURN, Time: 11258, Actual_Vol1: -1146.000000, Actual_Vol2: 1164.000000, Actual_Vol3: -610.000000, Actual_Vol4: 610.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1923.709406, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.269927, Absolute Angle: 1.610270, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    948:   28376 [INFO] CHASSIS_PID_TURN, Time: 11268, Actual_Vol1: -1152.000000, Actual_Vol2: 1201.000000, Actual_Vol3: -659.000000, Actual_Vol4: 616.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1901.031231, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.267818, Absolute Angle: 1.610233, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    949:   28376 [INFO] CHASSIS_PID_TURN, Time: 11278, Actual_Vol1: -1220.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -696.000000, Actual_Vol4: 684.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1878.344409, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268682, Absolute Angle: 1.610248, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    950:   28376 [INFO] CHASSIS_PID_TURN, Time: 11288, Actual_Vol1: -1238.000000, Actual_Vol2: 1257.000000, Actual_Vol3: -702.000000, Actual_Vol4: 819.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1855.659189, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268522, Absolute Angle: 1.610245, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.008376,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    951:   28378 [INFO] CHASSIS_PID_TURN, Time: 11298, Actual_Vol1: -1250.000000, Actual_Vol2: 1269.000000, Actual_Vol3: -696.000000, Actual_Vol4: 653.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1833.012143, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264705, Absolute Angle: 1.610179, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009157,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    952:   28378 [INFO] CHASSIS_PID_TURN, Time: 11308, Actual_Vol1: -1244.000000, Actual_Vol2: 1300.000000, Actual_Vol3: -819.000000, Actual_Vol4: 622.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1810.374272, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263787, Absolute Angle: 1.610163, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    953:   28378 [INFO] CHASSIS_PID_TURN, Time: 11318, Actual_Vol1: -1318.000000, Actual_Vol2: 1343.000000, Actual_Vol3: -912.000000, Actual_Vol4: 801.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1787.691954, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.268232, Absolute Angle: 1.610240, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    954:   28380 [INFO] CHASSIS_PID_TURN, Time: 11328, Actual_Vol1: -1343.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -936.000000, Actual_Vol4: 838.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1765.031208, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266075, Absolute Angle: 1.610203, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.010074,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    955:   28380 [INFO] CHASSIS_PID_TURN, Time: 11338, Actual_Vol1: -1337.000000, Actual_Vol2: 1361.000000, Actual_Vol3: -924.000000, Actual_Vol4: 844.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1742.410826, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.262038, Absolute Angle: 1.610132, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    956:   28380 [INFO] CHASSIS_PID_TURN, Time: 11348, Actual_Vol1: -1386.000000, Actual_Vol2: 1404.000000, Actual_Vol3: -967.000000, Actual_Vol4: 832.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1719.776579, kD: 35.000000,
kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263425, Absolute Angle: 1.610156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823,
over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    957:   28382 [INFO] CHASSIS_PID_TURN, Time: 11358, Actual_Vol1: -1429.000000, Actual_Vol2: 1441.000000, Actual_Vol3: -1016.000000, Actual_Vol4: 912.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1697.137121, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263946, Absolute Angle: 1.610165, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011823
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    958:   28382 [INFO] CHASSIS_PID_TURN, Time: 11368, Actual_Vol1: -1435.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 924.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1674.502773, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.263435, Absolute Angle: 1.610156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    959:   28382 [INFO] CHASSIS_PID_TURN, Time: 11378, Actual_Vol1: -1435.000000, Actual_Vol2: 1454.000000, Actual_Vol3: -1029.000000, Actual_Vol4: 936.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1651.858273, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.264450, Absolute Angle: 1.610174, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    960:   28384 [INFO] CHASSIS_PID_TURN, Time: 11388, Actual_Vol1: -1472.000000, Actual_Vol2: 1546.000000, Actual_Vol3: -1066.000000, Actual_Vol4: 930.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1629.192063, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.266621, Absolute Angle: 1.610212, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    961:   28384 [INFO] CHASSIS_PID_TURN, Time: 11398, Actual_Vol1: -1546.000000, Actual_Vol2: 1577.000000, Actual_Vol3: -1096.000000, Actual_Vol4: 973.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1606.534072, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.265799, Absolute Angle: 1.610198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.011248
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    962:   28384 [INFO] CHASSIS_PID_TURN, Time: 11408, Actual_Vol1: -1571.000000, Actual_Vol2: 1552.000000, Actual_Vol3: -1158.000000, Actual_Vol4: 1035.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1583.920735, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.261334, Absolute Angle: 1.610120, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01195
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.200000
    963:   28386 [INFO] CHASSIS_PID_TURN, Time: 11418, Actual_Vol1: -1571.000000, Actual_Vol2: 1472.000000, Actual_Vol3: -1152.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1561.298266, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.262247, Absolute Angle: 1.610136, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01195
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    964:   28386 [INFO] CHASSIS_PID_TURN, Time: 11428, Actual_Vol1: -1614.000000, Actual_Vol2: 1546.000000, Actual_Vol3: -1201.000000, Actual_Vol4: 1029.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1538.708417, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258985, Absolute Angle: 1.610079, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01430
1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
    965:   28386 [INFO] CHASSIS_PID_TURN, Time: 11438, Actual_Vol1: -1657.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1078.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1516.136456, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.257196, Absolute Angle: 1.610048, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01609
0, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
```

```
28388 [INFO] CHASSIS_PID_TURN, Time: 11448, Actual_Vol1: -1663.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1244.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1493.587058, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254940, Absolute Angle: 1.610008, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01498
7, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28388 [INFO] CHASSIS_PID_TURN, Time: 11458, Actual_Vol1: -1669.000000, Actual_Vol2: 1583.000000, Actual_Vol3: -1214.000000, Actual_Vol4: 1152.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1471.005361, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258170, Absolute Angle: 1.610065, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01374
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28388 [INFO] CHASSIS_PID_TURN, Time: 11468, Actual_Vol1: -1712.000000, Actual_Vol2: 1626.000000, Actual_Vol3: -1207.000000, Actual_Vol4: 1146.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1448.421418, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.258394, Absolute Angle: 1.610068, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01374
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28390 [INFO] CHASSIS_PID_TURN, Time: 11478, Actual_Vol1: -1756.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1238.000000, Actual_Vol4: 1195.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1425.873969, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254745, Absolute Angle: 1.610005, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01377
7, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28390 [INFO] CHASSIS_PID_TURN, Time: 11488, Actual_Vol1: -1762.000000, Actual_Vol2: 1682.000000, Actual_Vol3: -1250.000000, Actual_Vol4: 1238.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1403.360266, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.251370, Absolute Angle: 1.609946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01686
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28390 [INFO] CHASSIS_PID_TURN, Time: 11498, Actual_Vol1: -1768.000000, Actual_Vol2: 1688.000000, Actual_Vol3: -1250.000000, Actual_Vol4: 1250.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1380.809447, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.255082, Absolute Angle: 1.610011, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01686
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28392 [INFO] CHASSIS_PID_TURN, Time: 11508, Actual_Vol1: -1805.000000, Actual_Vol2: 1731.000000, Actual_Vol3: -1300.000000, Actual_Vol4: 1257.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1358.268386, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.254106, Absolute Angle: 1.609994, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01686
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28392 [INFO] CHASSIS_PID_TURN, Time: 11518, Actual_Vol1: -1848.000000, Actual_Vol2: 1768.000000, Actual_Vol3: -1331.000000, Actual_Vol4: 1294.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1335.707196, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.256119, Absolute Angle: 1.610029, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01525
1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 7.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28392 [INFO] CHASSIS_PID_TURN, Time: 11528, Actual_Vol1: -1854.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1343.000000, Actual_Vol4: 1331.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1313.180007, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.252719, Absolute Angle: 1.609969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01525
1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28394 [INFO] CHASSIS_PID_TURN, Time: 11538, Actual_Vol1: -1860.000000, Actual_Vol2: 1780.000000, Actual_Vol3: -1349.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1290.687918, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.249209, Absolute Angle: 1.609908, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01741
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28394 [INFO] CHASSIS_PID_TURN, Time: 11548, Actual_Vol1: -1922.000000, Actual_Vol2: 1817.000000, Actual_Vol3: -1386.000000, Actual_Vol4: 1343.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1268.195764, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.249215, Absolute Angle: 1.609908, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.01741
2, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28394 [INFO] CHASSIS_PID_TURN, Time: 11558, Actual_Vol1: -1965.000000, Actual_Vol2: 1866.000000, Actual_Vol3: -1435.000000, Actual_Vol4: 1386.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1245.734440, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.246132, Absolute Angle: 1.609854, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02048
9, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28396 [INFO] CHASSIS_PID_TURN, Time: 11568, Actual_Vol1: -1971.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1448.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1223.333931, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.240051, Absolute Angle: 1.609748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02657
0, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -9.400000, Actual_Vel4: 0.000000
28396 [INFO] CHASSIS_PID_TURN, Time: 11578, Actual_Vol1: -1965.000000, Actual_Vol2: 1873.000000, Actual_Vol3: -1441.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1200.952173, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238176, Absolute Angle: 1.609716, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02844
5, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28396 [INFO] CHASSIS_PID_TURN, Time: 11588, Actual_Vol1: -2088.000000, Actual_Vol2: 1940.000000, Actual_Vol3: -1478.000000, Actual_Vol4: 1429.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1178.568052, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238412, Absolute Angle: 1.609720, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02762
3, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28398 [INFO] CHASSIS_PID_TURN, Time: 11598, Actual_Vol1: -2218.000000, Actual_Vol2: 1983.000000, Actual_Vol3: -1571.000000, Actual_Vol4: 1478.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1156.181603, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.238645, Absolute Angle: 1.609724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02407
1, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28398 [INFO] CHASSIS_PID_TURN, Time: 11608, Actual_Vol1: -2168.000000, Actual_Vol2: 1990.000000, Actual_Vol3: -1571.000000, Actual_Vol4: 1565.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1133.829209, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.235239, Absolute Angle: 1.609664, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02700
8, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28398 [INFO] CHASSIS_PID_TURN, Time: 11618, Actual_Vol1: -2039.000000, Actual_Vol2: 2002.000000, Actual_Vol3: -1583.000000, Actual_Vol4: 1571.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1111.496136, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.233307, Absolute Angle: 1.609631, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.02567
8, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28400 [INFO] CHASSIS_PID_TURN, Time: 11628, Actual_Vol1: -2101.000000, Actual_Vol2: 2119.000000, Actual_Vol3: -1663.000000, Actual_Vol4: 1577.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1089.213579, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.228256, Absolute Angle: 1.609542, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03013
9, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28400 [INFO] CHASSIS_PID_TURN, Time: 11638, Actual_Vol1: -2230.000000, Actual_Vol2: 2248.000000, Actual_Vol3: -1749.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1066.968559, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.224502, Absolute Angle: 1.609477, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03389
2, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28400 [INFO] CHASSIS_PID_TURN, Time: 11648, Actual_Vol1: -2242.000000, Actual_Vol2: 2267.000000, Actual_Vol3: -1768.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1044.768404, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.220015, Absolute Angle: 1.609399, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03837
9, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28402 [INFO] CHASSIS_PID_TURN, Time: 11658, Actual_Vol1: -2248.000000, Actual_Vol2: 2273.000000, Actual_Vol3: -1774.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1022.596289, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.217212, Absolute Angle: 1.609350, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.04118
3, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28402 [INFO] CHASSIS_PID_TURN, Time: 11668, Actual_Vol1: -2316.000000, Actual_Vol2: 2341.000000, Actual_Vol3: -1817.000000, Actual_Vol4: 1589.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 1000.427246, kD: 35.00000
0, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.216904, Absolute Angle: 1.609344, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.03921
5, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28402 [INFO] CHASSIS_PID_TURN, Time: 11678, Actual_Vol1: -2353.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1860.000000, Actual_Vol4: 1663.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 978.316339, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.211091, Absolute Angle: 1.609243, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045028
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28404 [INFO] CHASSIS_PID_TURN, Time: 11688, Actual_Vol1: -2359.000000, Actual_Vol2: 2402.000000, Actual_Vol3: -1873.000000, Actual_Vol4: 1743.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 956.231250, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.208509, Absolute Angle: 1.609198, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.047610
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28404 [INFO] CHASSIS_PID_TURN, Time: 11698, Actual_Vol1: -2359.000000, Actual_Vol2: 2390.000000, Actual_Vol3: -1879.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 934.180666, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.205058, Absolute Angle: 1.609138, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.051061
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28404 [INFO] CHASSIS_PID_TURN, Time: 11708, Actual_Vol1: -2402.000000, Actual_Vol2: 2470.000000, Actual_Vol3: -1940.000000, Actual_Vol4: 1762.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 912.161550, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.201912, Absolute Angle: 1.609083, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.054207
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28406 [INFO] CHASSIS_PID_TURN, Time: 11718, Actual_Vol1: -2445.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -1996.000000, Actual_Vol4: 1805.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 890.112381, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.204917, Absolute Angle: 1.609135, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050807
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28406 [INFO] CHASSIS_PID_TURN, Time: 11728, Actual_Vol1: -2452.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2033.000000, Actual_Vol4: 1885.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 868.094528, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.201785, Absolute Angle: 1.609080, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.047430
, over slew: 0, Actual_Vel1: -11.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28406 [INFO] CHASSIS_PID_TURN, Time: 11738, Actual_Vol1: -2452.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2076.000000, Actual_Vol4: 1965.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 846.084590, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.200994, Absolute Angle: 1.609067, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.048222
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28408 [INFO] CHASSIS_PID_TURN, Time: 11748, Actual_Vol1: -2452.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 1971.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 824.063069, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.202152, Absolute Angle: 1.609087, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.045139
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28408 [INFO] CHASSIS_PID_TURN, Time: 11758, Actual_Vol1: -2464.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2088.000000, Actual_Vol4: 2020.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 802.022994, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.204007, Absolute Angle: 1.609119, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039057
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28408 [INFO] CHASSIS_PID_TURN, Time: 11768, Actual_Vol1: -2464.000000, Actual_Vol2: 2501.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2070.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 780.021673, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.200132, Absolute Angle: 1.609052, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.038513
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 10.000000
28410 [INFO] CHASSIS_PID_TURN, Time: 11778, Actual_Vol1: -2458.000000, Actual_Vol2: 2495.000000, Actual_Vol3: -2094.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 758.067472, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.195420, Absolute Angle: 1.608969, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043225
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28410 [INFO] CHASSIS_PID_TURN, Time: 11788, Actual_Vol1: -2532.000000, Actual_Vol2: 2636.000000, Actual_Vol3: -2174.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 736.118943, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.194853, Absolute Angle: 1.608959, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.043792
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28410 [INFO] CHASSIS_PID_TURN, Time: 11798, Actual_Vol1: -2618.000000, Actual_Vol2: 2661.000000, Actual_Vol3: -2255.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 714.179503, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193944, Absolute Angle: 1.608944, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.041295
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28412 [INFO] CHASSIS_PID_TURN, Time: 11808, Actual_Vol1: -2630.000000, Actual_Vol2: 2698.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 2082.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 692.242497, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193701, Absolute Angle: 1.608939, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.039607
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28412 [INFO] CHASSIS_PID_TURN, Time: 11818, Actual_Vol1: -2636.000000, Actual_Vol2: 2710.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 670.335964, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.190653, Absolute Angle: 1.608886, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.037602
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1004:   28412 [INFO] CHASSIS_PID_TURN, Time: 11828, Actual_Vol1: -2735.000000, Actual_Vol2: 2772.000000, Actual_Vol3: -2347.000000, Actual_Vol4: 2156.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 648.406536, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.192943, Absolute Angle: 1.608926, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033849
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1005:   28414 [INFO] CHASSIS_PID_TURN, Time: 11838, Actual_Vol1: -2815.000000, Actual_Vol2: 2790.000000, Actual_Vol3: -2390.000000, Actual_Vol4: 2236.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 626.468837, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.193770, Absolute Angle: 1.608941, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.029362
, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1006:   28414 [INFO] CHASSIS_PID_TURN, Time: 11848, Actual_Vol1: -2747.000000, Actual_Vol2: 2797.000000, Actual_Vol3: -2402.000000, Actual_Vol4: 2261.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 604.545971, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.192287, Absolute Angle: 1.608915, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026558
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1007:   28414 [INFO] CHASSIS_PID_TURN, Time: 11858, Actual_Vol1: -2766.000000, Actual_Vol2: 2797.000000, Actual_Vol3: -2427.000000, Actual_Vol4: 2261.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 582.637322, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.190865, Absolute Angle: 1.608890, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.026251
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1008:   28416 [INFO] CHASSIS_PID_TURN, Time: 11868, Actual_Vol1: -2840.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2421.000000, Actual_Vol4: 2335.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 560.773214, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.186411, Absolute Angle: 1.608812, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024680
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -15.800000, Actual_Vel4: 117.800000
  1009:   28416 [INFO] CHASSIS_PID_TURN, Time: 11878, Actual_Vol1: -2846.000000, Actual_Vol2: 2883.000000, Actual_Vol3: -2396.000000, Actual_Vol4: 2328.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 538.959165, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.181405, Absolute Angle: 1.608725, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.027104
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: -0.000000, Actual_Vel3: -15.800000, Actual_Vel4: 117.800000
  1010:   28416 [INFO] CHASSIS_PID_TURN, Time: 11888, Actual_Vol1: -2852.000000, Actual_Vol2: 2895.000000, Actual_Vol3: -2396.000000, Actual_Vol4: 2267.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 517.246283, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.171288, Absolute Angle: 1.608548, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.033770
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 238.000000, Actual_Vel3: -15.800000, Actual_Vel4: 18.000000
  1011:   28418 [INFO] CHASSIS_PID_TURN, Time: 11898, Actual_Vol1: -2858.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 2162.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 495.699678, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.154660, Absolute Angle: 1.608258, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.050256
, over slew: 0, Actual_Vel1: -9.000000, Actual_Vel2: 238.000000, Actual_Vel3: -31.600000, Actual_Vel4: 19.200000
  1012:   28418 [INFO] CHASSIS_PID_TURN, Time: 11908, Actual_Vol1: -2846.000000, Actual_Vol2: 2864.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 2088.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 474.395587, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 169.000000, position_r: -159.000000, Heading_Sp: 90.000000, Relative_Heading: 92.130409, Absolute Angle: 1.607835, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.074508
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -31.600000, Actual_Vel4: 19.200000
  1013:   28418 [INFO] CHASSIS_PID_TURN, Time: 11918, Actual_Vol1: -2840.000000, Actual_Vol2: 2871.000000, Actual_Vol3: -2384.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 453.413222, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 92.098237, Absolute Angle: 1.607273, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.105771
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 16.000000
  1014:   28420 [INFO] CHASSIS_PID_TURN, Time: 11928, Actual_Vol1: -2846.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -2378.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 434.834477, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.857875, Absolute Angle: 1.603078, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.346133
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 14.000000
  1015:   28420 [INFO] CHASSIS_PID_TURN, Time: 11938, Actual_Vol1: -2846.000000, Actual_Vol2: 2858.000000, Actual_Vol3: -2372.000000, Actual_Vol4: 2064.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 416.784222, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.805025, Absolute Angle: 1.602156, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.398982
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -16.200000, Actual_Vel4: 14.000000
  1016:   28420 [INFO] CHASSIS_PID_TURN, Time: 11948, Actual_Vol1: -2852.000000, Actual_Vol2: 2877.000000, Actual_Vol3: -2328.000000, Actual_Vol4: 2076.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 399.227879, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.755634, Absolute Angle: 1.601294, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.448373
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 16.800000, Actual_Vel3: -12.200000, Actual_Vel4: 13.400000
  1017:   28422 [INFO] CHASSIS_PID_TURN, Time: 11958, Actual_Vol1: -2852.000000, Actual_Vol2: 2889.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 2027.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 382.180594, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.704728, Absolute Angle: 1.600405, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.495404
, over slew: 0, Actual_Vel1: -8.600000, Actual_Vel2: 15.600000, Actual_Vel3: -12.200000, Actual_Vel4: 13.600000
  1018:   28422 [INFO] CHASSIS_PID_TURN, Time: 11968, Actual_Vol1: -2852.000000, Actual_Vol2: 2809.000000, Actual_Vol3: -2273.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 365.587579, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -158.000000, Heading_Sp: 90.000000, Relative_Heading: 91.659302, Absolute Angle: 1.599612, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.536119
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 13.600000
  1019:   28422 [INFO] CHASSIS_PID_TURN, Time: 11978, Actual_Vol1: -2790.000000, Actual_Vol2: 2858.000000, Actual_Vol3: -2261.000000, Actual_Vol4: 1934.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 349.430636, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.615694, Absolute Angle: 1.598851, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.579159
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 13.600000
  1020:   28424 [INFO] CHASSIS_PID_TURN, Time: 11988, Actual_Vol1: -2772.000000, Actual_Vol2: 2815.000000, Actual_Vol3: -2267.000000, Actual_Vol4: 1891.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 334.738453, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.469218, Absolute Angle: 1.596295, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.724726
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.200000, Actual_Vel4: 12.400000
  1021:   28424 [INFO] CHASSIS_PID_TURN, Time: 11998, Actual_Vol1: -2710.000000, Actual_Vol2: 2803.000000, Actual_Vol3: -2168.000000, Actual_Vol4: 1712.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 320.515438, kD: 35.000000
, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 168.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.422302, Absolute Angle: 1.595476, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.771468
, over slew: 0, Actual_Vel1: -13.800000, Actual_Vel2: 15.600000, Actual_Vel3: -11.000000, Actual_Vel4: 12.400000
  1022:   28424 [INFO] CHASSIS_PID_TURN, Time: 12008, Actual_Vol1: -776.000000, Actual_Vol2: 715.000000, Actual_Vol3: -499.000000, Actual_Vol4: 363.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 306.804051, kD: 35.000000, kI:
0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.371139, Absolute Angle: 1.594583, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.822631, ov
er slew: 0, Actual_Vel1: -17.800000, Actual_Vel2: 25.400000, Actual_Vel3: -6.400000, Actual_Vel4: 12.400000
  1023:   28426 [INFO] CHASSIS_PID_TURN, Time: 12018, Actual_Vol1: -179.000000, Actual_Vol2: 216.000000, Actual_Vol3: -99.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 294.551132, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.225292, Absolute Angle: 1.592037, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.968478, over s
lew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 12.400000
  1024:   28426 [INFO] CHASSIS_PID_TURN, Time: 12028, Actual_Vol1: -271.000000, Actual_Vol2: 246.000000, Actual_Vol3: -99.000000, Actual_Vol4: 18.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 282.604818, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.194631, Absolute Angle: 1.591502, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.999138, over s
lew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
  1025:   28426 [INFO] CHASSIS_PID_TURN, Time: 12038, Actual_Vol1: -413.000000, Actual_Vol2: 320.000000, Actual_Vol3: -105.000000, Actual_Vol4: 68.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 270.726472, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.187835, Absolute Angle: 1.591384, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.004452, over
 slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
  1026:   28428 [INFO] CHASSIS_PID_TURN, Time: 12048, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 258.758599, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.196787, Absolute Angle: 1.591540, error history: 20, history size: 20, time out time: -2147481531, error difference: 1.003030, over
 slew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: -0.000000, Actual_Vel4: 8.400000
  1027:   28428 [INFO] CHASSIS_PID_TURN, Time: 12058, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 246.557861, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.220074, Absolute Angle: 1.591946, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.998576, over s
lew: 0, Actual_Vel1: -7.800000, Actual_Vel2: 25.400000, Actual_Vel3: 8.000000, Actual_Vel4: 8.400000
  1028:   28428 [INFO] CHASSIS_PID_TURN, Time: 12068, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 234.049552, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.250831, Absolute Angle: 1.592483, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.993570, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 8.000000, Actual_Vel4: 8.400000
  1029:   28430 [INFO] CHASSIS_PID_TURN, Time: 12078, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -99.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 221.312258, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.273729, Absolute Angle: 1.592883, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.983454, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 8.000000, Actual_Vel4: 5.200000
  1030:   28430 [INFO] CHASSIS_PID_TURN, Time: 12088, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 208.452981, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.285928, Absolute Angle: 1.593096, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.966826, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: 26.600000, Actual_Vel3: 16.800000, Actual_Vel4: 5.200000
  1031:   28430 [INFO] CHASSIS_PID_TURN, Time: 12098, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 195.638810, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.281417, Absolute Angle: 1.593017, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.942575, over s
lew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 16.800000, Actual_Vel4: 5.200000
  1032:   28432 [INFO] CHASSIS_PID_TURN, Time: 12108, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -92.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 182.832356, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.280645, Absolute Angle: 1.593004, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.910402, over s
lew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 16.800000, Actual_Vel4: 0.000000
  1033:   28432 [INFO] CHASSIS_PID_TURN, Time: 12118, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 170.062333, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.277002, Absolute Angle: 1.592940, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.670040, over
 slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: 0.000000
  1034:   28432 [INFO] CHASSIS_PID_TURN, Time: 12128, Actual_Vol1: -431.000000, Actual_Vol2: 326.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 157.370291, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.269204, Absolute Angle: 1.592804, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.617191, over s
lew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: 0.000000
  1035:   28434 [INFO] CHASSIS_PID_TURN, Time: 12138, Actual_Vol1: -437.000000, Actual_Vol2: 363.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 144.751673, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261862, Absolute Angle: 1.592676, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.567800, over s
lew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: 20.600000, Actual_Vel4: -39.200000
  1036:   28434 [INFO] CHASSIS_PID_TURN, Time: 12148, Actual_Vol1: -431.000000, Actual_Vol2: 363.000000, Actual_Vol3: -105.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 132.181168, kD: 35.000000, kI: 0.
000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.257051, Absolute Angle: 1.592592, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.516894, over
 slew: 0, Actual_Vel1: 12.800000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -39.200000
  1037:   28434 [INFO] CHASSIS_PID_TURN, Time: 12158, Actual_Vol1: -493.000000, Actual_Vol2: 400.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 119.576957, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.260421, Absolute Angle: 1.592651, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.471467, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: -39.200000
  1038:   28436 [INFO] CHASSIS_PID_TURN, Time: 12168, Actual_Vol1: -511.000000, Actual_Vol2: 444.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 106.943315, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263364, Absolute Angle: 1.592702, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.427860, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1039:   28436 [INFO] CHASSIS_PID_TURN, Time: 12178, Actual_Vol1: -511.000000, Actual_Vol2: 450.000000, Actual_Vol3: -105.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 94.278076, kD: 35.000000, kI: 0.0
00700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266524, Absolute Angle: 1.592757, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.281384, over s
lew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1040:   28436 [INFO] CHASSIS_PID_TURN, Time: 12188, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 86.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 81.615794, kD: 35.000000, kI: 0.00
0700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266228, Absolute Angle: 1.592752, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.234467, over sl
ew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
  1041:   28438 [INFO] CHASSIS_PID_TURN, Time: 12198, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -99.000000, Actual_Vol4: 92.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 68.985541, kD: 35.000000, kI: 0.00
```

```
0700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263025, Absolute Angle: 1.592696, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.183304, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28438 [INFO] CHASSIS_PID_TURN, Time: 12208, Actual_Vol1: -517.000000, Actual_Vol2: 450.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 56.325654, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265989, Absolute Angle: 1.592748, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28438 [INFO] CHASSIS_PID_TURN, Time: 12218, Actual_Vol1: -579.000000, Actual_Vol2: 493.000000, Actual_Vol3: -92.000000, Actual_Vol4: 80.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 43.682894, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264276, Absolute Angle: 1.592718, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28440 [INFO] CHASSIS_PID_TURN, Time: 12228, Actual_Vol1: -610.000000, Actual_Vol2: 530.000000, Actual_Vol3: -197.000000, Actual_Vol4: 191.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 31.063019, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261987, Absolute Angle: 1.592678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.098093, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28440 [INFO] CHASSIS_PID_TURN, Time: 12238, Actual_Vol1: -604.000000, Actual_Vol2: 536.000000, Actual_Vol3: -216.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 18.443358, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.261966, Absolute Angle: 1.592678, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.089140, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28440 [INFO] CHASSIS_PID_TURN, Time: 12248, Actual_Vol1: -604.000000, Actual_Vol2: 530.000000, Actual_Vol3: -228.000000, Actual_Vol4: 228.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: 5.796962, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264640, Absolute Angle: 1.592724, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.065854, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28442 [INFO] CHASSIS_PID_TURN, Time: 12258, Actual_Vol1: -604.000000, Actual_Vol2: 536.000000, Actual_Vol3: -222.000000, Actual_Vol4: 222.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -6.812050, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.260901, Absolute Angle: 1.592659, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.035097, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28442 [INFO] CHASSIS_PID_TURN, Time: 12268, Actual_Vol1: -616.000000, Actual_Vol2: 536.000000, Actual_Vol3: -234.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -19.411581, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.259953, Absolute Angle: 1.592642, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028877, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28442 [INFO] CHASSIS_PID_TURN, Time: 12278, Actual_Vol1: -616.000000, Actual_Vol2: 579.000000, Actual_Vol3: -228.000000, Actual_Vol4: 216.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -32.036665, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.262508, Absolute Angle: 1.592687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.028877, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28444 [INFO] CHASSIS_PID_TURN, Time: 12288, Actual_Vol1: -684.000000, Actual_Vol2: 628.000000, Actual_Vol3: -296.000000, Actual_Vol4: 290.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -44.668432, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.263177, Absolute Angle: 1.592699, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.024367, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28444 [INFO] CHASSIS_PID_TURN, Time: 12298, Actual_Vol1: -696.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -57.293692, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.262526, Absolute Angle: 1.592687, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.023595, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28444 [INFO] CHASSIS_PID_TURN, Time: 12308, Actual_Vol1: -696.000000, Actual_Vol2: 634.000000, Actual_Vol3: -302.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -69.955795, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.266210, Absolute Angle: 1.592752, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.019952, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28446 [INFO] CHASSIS_PID_TURN, Time: 12318, Actual_Vol1: -696.000000, Actual_Vol2: 628.000000, Actual_Vol3: -308.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -82.603629, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.264783, Absolute Angle: 1.592727, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.012154, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28446 [INFO] CHASSIS_PID_TURN, Time: 12328, Actual_Vol1: -702.000000, Actual_Vol2: 628.000000, Actual_Vol3: -308.000000, Actual_Vol4: 302.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -95.261112, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265748, Absolute Angle: 1.592744, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009473, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28446 [INFO] CHASSIS_PID_TURN, Time: 12338, Actual_Vol1: -708.000000, Actual_Vol2: 634.000000, Actual_Vol3: -308.000000, Actual_Vol4: 314.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -107.916924, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265581, Absolute Angle: 1.592741, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.009473, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
28448 [INFO] CHASSIS_PID_TURN, Time: 12348, Actual_Vol1: -813.000000, Actual_Vol2: 671.000000, Actual_Vol3: -308.000000, Actual_Vol4: 308.000000, Slew: 15.000000, Brake: 1, Gear: 2, I_max: 2147483647.000000, I: -120.568364, kD: 35.000000, kI: 0.000700, kP: 3.000000, Position_Sp: 0, position_l: 167.000000, position_r: -157.000000, Heading_Sp: 90.000000, Relative_Heading: 91.265144, Absolute Angle: 1.592733, error history: 20, history size: 20, time out time: -2147481531, error difference: 0.006571, over slew: 0, Actual_Vel1: 0.000000, Actual_Vel2: -0.000000, Actual_Vel3: -0.000000, Actual_Vel4: 0.000000
WARNING - pros.serial.terminal.terminal:stop - Stopping terminal
```

```c
/**
 * @file: ./RobotCode/include/fonts/fonts.h
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: ../src/objects/lcdCode/fonts/
 *
 * contains definitions for fonts to use for lvgl
 * this is used to make gui more interesting and provide more contrast and have
 * the ability to fit more data because of a smaller font
 */

#ifndef __FONTS_H__
#define __FONTS_H__


#ifdef USE_DEJAVU_9
LV_FONT_DECLARE(dejavu_9);
extern lv_font_t dejavu_9;
#endif

#ifdef USE_DEJAVU_12
LV_FONT_DECLARE(dejavu_12);
extern lv_font_t dejavu_12;
#endif

#ifdef USE_DEJAVU_16
LV_FONT_DECLARE(dejavu_16);
extern lv_font_t dejavu_16;
#endif

#endif
```

```cpp
#include <csignal>

#include "main.h"

#include "Autons.hpp"
#include "Configuration.hpp"
#include "DriverControl.hpp"
#include "objects/controller/controller.hpp"
#include "objects/lcdCode/DriverControl/AutonomousLCD.hpp"
#include "objects/lcdCode/DriverControl/DriverControlLCD.hpp"
#include "objects/lcdCode/gui.hpp"
#include "objects/lcdCode/TemporaryScreen.hpp"
#include "objects/motors/Motors.hpp"
#include "objects/motors/MotorThread.hpp"
#include "objects/position_tracking/PositionTracker.hpp"
#include "objects/serial/Logger.hpp"
#include "objects/serial/Server.hpp"
#include "objects/subsystems/chassis.hpp"

int final_auton_choice;
AutonomousLCD auton_lcd;

/**
 * Runs initialization code. This occurs as soon as the program is started.
 *
 * All other competition modes are blocked by initialize; it is recommended
 * to keep execution time for this mode under a few seconds.
 */
 void initialize()
 {
    pros::c::serctl(SERCTL_ACTIVATE, 0);  // I think this enables stdin (necessary to start server)

    Motors::register_motors();
    MotorThread::get_instance()->start_thread();

    pros::delay(100); //wait for terminal to start and lvgl
    Configuration* config = Configuration::get_instance();
    config->init();
    config->print_config_options();

    final_auton_choice = chooseAuton();
    Autons auton;
    config->filter_color = auton.AUTONOMOUS_COLORS.at(final_auton_choice);
    auton.set_autonomous_number(final_auton_choice);

    Sensors::calibrate_imu();


    // std::cout << OptionsScreen::cnfg.use_hardcoded << '\n';
    // std::cout << OptionsScreen::cnfg.gyro_turn << '\n';
    // std::cout << OptionsScreen::cnfg.accelleration_ctrl << '\n';
    // std::cout << OptionsScreen::cnfg.check_motor_tmp << '\n';
    // std::cout << OptionsScreen::cnfg.use_previous_macros << '\n';
    // std::cout << OptionsScreen::cnfg.record << '\n';

    std::cout << "initalize finished" << "\n";
    lv_scr_load(tempScreen::temp_screen);
}



/**
 * Runs while the robot is in the disabled state of Field Management System or
 * the VEX Competition Switch, following either autonomous or opcontrol. When
 * the robot is enabled, this task will exit.
 */
void disabled() {}



/**
 * Runs after initialize(), and before autonomous when connected to the Field
 * Management System or the VEX Competition Switch. This is intended for
 * competition-specific initialization routines, such as an autonomous selector
 * on the LCD.
 *
 * This task will exit when the robot is enabled and autonomous or opcontrol
 * starts.
 */
void competition_initialize() {
    Autons auton;
    auton_lcd.update_labels(auton.get_autonomous_number());
}



/**
 * Runs the user autonomous code. This function will be started in its own task
 * with the default priority and stack size whenever the robot is enabled via
 * the Field Management System or the VEX Competition Switch in the autonomous
 * mode. Alternatively, this function may be called in initialize or opcontrol
 * for non-competition testing purposes.
 *
 * If the robot is disabled or communications is lost, the autonomous task
 * will be stopped. Re-enabling the robot will restart the task, not re-start it
 * from where it left off.
 */
void autonomous() {
    Autons auton;
    auton.run_autonomous();
}


 void log_thread_fn( void* )
 {
     // TODO: add back imu functionality when imu is mounted
     Logger logger;
     Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);

     Configuration* config = Configuration::get_instance();
     double kP = config->chassis_pid.kP;
     double kI = config->chassis_pid.kI;
     double kD = config->chassis_pid.kD;
     double I_max = config->chassis_pid.I_max;

     int l_id = Sensors::left_encoder.get_unique_id();
     int r_id = Sensors::right_encoder.get_unique_id();
     // double prev_l_encoder = std::get<0>(chassis.get_average_encoders(l_id, r_id));
     // double prev_r_encoder = std::get<1>(chassis.get_average_encoders(l_id, r_id));
     // double intitial_angle = Sensors::imu.get_heading();
     // double prev_angle = Sensors::imu.get_heading();
     // double relative_angle = 0;
     //
     while ( 1 )
     {
         // double delta_l = std::get<0>(chassis.get_average_encoders(l_id, r_id)) - prev_l_encoder;
         // double delta_r = std::get<1>(chassis.get_average_encoders(l_id, r_id)) - prev_r_encoder;
         // double delta_theta = chassis.calc_delta_theta(prev_angle, delta_l, delta_r);
         // prev_angle = prev_angle + delta_theta;
         // relative_angle = relative_angle + delta_theta;
         //

         std::string msg = (
             "[INFO] " + std::string("CHASSIS_PID ")
             + ", Actual_Vol1: " + std::to_string(Motors::front_left.get_actual_voltage())
             + ", Actual_Vol2: " + std::to_string(Motors::front_right.get_actual_voltage())
             + ", Actual_Vol3: " + std::to_string(Motors::back_left.get_actual_voltage())
             + ", Actual_Vol4: " + std::to_string(Motors::back_right.get_actual_voltage())
             + ", Slew: " + std::to_string(0)
             + ", Brake: " + std::to_string(Motors::front_left.get_brake_mode())
             + ", Gear: " + std::to_string(Motors::front_left.get_gearset())
             + ", I_max: " + std::to_string(I_max)
             + ", I: " + std::to_string(0)
             + ", kD: " + std::to_string(kD)
             + ", kI: " + std::to_string(kI)
             + ", kP: " + std::to_string(kP)
             + ", Time: " + std::to_string(pros::millis())
             + ", Position_Sp: " + std::to_string(1269.32)
             + ", position_l: " + std::to_string(Sensors::left_encoder.get_position(l_id))
             + ", position_r: " + std::to_string(Sensors::right_encoder.get_position(r_id))
             + ", Heading_Sp: " + std::to_string(0)
             + ", Relative_Heading: " + std::to_string(0)
             + ", Actual_Vel1: " + std::to_string(Motors::front_left.get_actual_velocity())
             + ", Actual_Vel2: " + std::to_string(Motors::front_right.get_actual_velocity())
             + ", Actual_Vel3: " + std::to_string(Motors::back_left.get_actual_velocity())
             + ", Actual_Vel4: " + std::to_string(Motors::back_right.get_actual_velocity())
         );
         // std::cout << msg << "\n";
         pros::delay(10);
     }
 }



 void Exit( int signal )
 {
     //Writer writer;
     std::cerr << "program caught " << signal << "\n" << std::flush;
     std::cerr << "errno: " << errno << "\n" << std::flush;
     std::cerr << "strerror: " << std::strerror(errno) << "\n" << std::flush;
     pros::delay(100); // wait for stdout to be flushed
     raise(signal);
 }



/**
 * Runs the operator control code. This function will be started in its own task
 * with the default priority and stack size whenever the robot is enabled via
 * the Field Management System or the VEX Competition Switch in the operator
 * control mode.
 *
 * If no competition control is connected, this function will run immediately
 * following initialize().
 *
 * If the robot is disabled or communications is lost, the
 * operator control task will be stopped. Re-enabling the robot will restart the
 * task, not resume it from where it left off.
 */
void opcontrol() {
     // pros::ADIAnalogIn l1 (1);
     // pros::ADIAnalogIn l2 (2);
     // pros::ADIAnalogIn l3 (3);
     // while(1) {
     //     std::cout << l1.get_value() << " " << l2.get_value() << " " << l3.get_value() << "\n";
     //     pros::delay(50);
     // }
    // Logger logger;
    // pros::ADIDigitalIn limit_switch('A');


    // pros::Task write_task (log_thread_fn,
    //                        (void*)NULL,
    //                        TASK_PRIORITY_DEFAULT,
    //                        TASK_STACK_DEPTH_DEFAULT,
    //                        "logger_thread");


    Server server;
    server.clear_stdin();
    server.start_server();
    server.set_debug_mode(true);

    // int stop = pros::millis() + 8000;
    //
    // Lift lift(Motors::lift, {0, 800});
    // while ( pros::millis() < stop )
    // {
    //     lift.move_to(900, false, true);
    //     pros::delay(10);
    // }
    // stop = pros::millis() + 2000;
    // while ( pros::millis() < stop )
    // {
    //     lift.move_to(0, false, true);
```

```
     //     pros::delay(10);
     // }
     // logger.dump();
     // logger.dump();
     // logger.dump();
     // logger.dump();
     // logger.dump();
     // logger.dump();
     // logger.dump();
     // logger.dump();
     //
     // Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, 12.4 );
     // int stop = pros::millis() + 8000;
     //
     // chassis.turn_left(13, 12000, INT32_MAX, true, false, true);
     //
     // while ( pros::millis() < stop )
     // {
     //     chassis.turn_left(13, 12000, INT32_MAX, false, false, true );
     //     pros::delay(10);
     // }

    std::cout << "opcontrol started\n";

    std::signal(SIGSEGV, Exit);
    std::signal(SIGTERM, Exit);
    std::signal(SIGINT, Exit);
    std::signal(SIGILL, Exit);
    std::signal(SIGABRT, Exit);
    std::signal(SIGFPE, Exit);
    std::signal(SIGBUS, Exit);
    std::signal(SIGALRM, Exit);
    std::signal(SIGSTOP, Exit);
    std::signal(SIGUSR1, Exit);
    std::signal(SIGUSR2, Exit);
    std::signal(SIGKILL, Exit);

    pros::delay(100);

    lv_scr_load(tempScreen::temp_screen);


    Controller controllers;
    // while(1) {
    //     Motors::left_intake.user_move(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y));
    //     Motors::right_intake.user_move(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y));
    //     std::cout << Motors::left_intake.get_torque() << " " << Motors::left_intake.get_efficiency() << " " << Motors::left_intake.get_actual_voltage() << "\n";
    //     std::cout << Motors::right_intake.get_torque() << " " << Motors::right_intake.get_efficiency() << " " << Motors::right_intake.get_actual_voltage() << "\n";
    //     std::cout << "\n";
    //     pros::delay(10);
    // }


     // Motors motors;
     // Motors::record_macro();
     //
     // Writer writer;
     // while( writer.get_count() > 0 )
     // {
     //     std::cout << pros::millis() << " " << writer.get_count() << "\n";
     //     pros::delay(1);
     // }
     //
     // std::cout << "done\n";

     //update controller with color of cube and if it is loaded or not

    // Controller controllers;
    // std::string controller_text = "no cube loaded";
    // std::string prev_controller_text = "";

    // PositionTracker* tracker = PositionTracker::get_instance();
    // tracker->start_thread();
    // std::cout << pros::Task::get_count() << "\n";
    // Chassis chassis(Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, Sensors::imu, 12.75, 5/3, 3.25);
    DriverControlLCD lcd;
    // lcd.update_labels();
    // chassis.generate_profiles();


    // chassis.turn_left(90);  // passing
    // pros::delay(1000);
    //
    // chassis.turn_right(90); // passing
    // pros::delay(1000);
    //
    // chassis.turn_left(45); // passing
    // pros::delay(1000);
    //
    // chassis.turn_right(45); // passing
    // pros::delay(1000);
    //
    // chassis.turn_left(30); // passing
    // pros::delay(1000);
    //
    // chassis.turn_right(30); // passing
    // pros::delay(1000);
    //
    // chassis.turn_left(10); // passing
    // pros::delay(1000);
    //
    // chassis.turn_right(10); // passing
    // pros::delay(1000);

    // chassis.turn_right(270); // passing
    // pros::delay(1000);
    //
    // chassis.turn_left(270); // passing
    // pros::delay(1000);
    //

    // chassis.turn_to_angle(90);  // passing
    // pros::delay(1000);
    //
    // chassis.turn_to_angle(270);  // passing
    // pros::delay(1000);
    //
    // chassis.turn_to_angle(0);  // passing
    // pros::delay(1000);
    //
    // chassis.turn_to_angle(-90);  // passing
    // pros::delay(1000);

    // chassis.turn_to_point(36, 0);  // passing
    // pros::delay(1000);
    // chassis.turn_to_point(-36, 0);  // passing
    // pros::delay(1000);
    // chassis.turn_to_point(0, 36);  // passing
    // pros::delay(1000);
    // chassis.turn_to_point(-36, -36);  // passing

    // chassis.drive_to_point(0, 36);  // passing
    // pros::delay(1000);
    // chassis.drive_to_point(36, 36);  // passing
    // pros::delay(1000);
    // chassis.drive_to_point(36, 0);  // passing
    // pros::delay(1000);
    // chassis.drive_to_point(0, 0);  // passing

    // tracker->stop_logging();
    lcd.update_labels();
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->enable_imu();
    tracker->start_thread();
    // chassis.turn_left(90);
    Autons autons;
    autons.skills2();

    // gather data from position tracker
    // tracker->start_logging();
    // while(1) {
    //     pros::delay(10);
    // }

    pros::Task driver_control_task (driver_control,
                                  (void*)NULL,
                                  TASK_PRIORITY_DEFAULT,
                                  TASK_STACK_DEPTH_DEFAULT,
                                  "DriverControlTask");



    // double prev_angle = std::fmod(Sensors::imu.get_heading() + 360, 360);
    // double ref_angle = std::fmod(Sensors::imu.get_heading() + 360, 360);
    int l_id = Sensors::left_encoder.get_unique_id();
    int r_id = Sensors::right_encoder.get_unique_id();
    int s_id = Sensors::strafe_encoder.get_unique_id();
    // double prev_l = Sensors::left_encoder.get_position(l_id);
    // double prev_r = Sensors::right_encoder.get_position(r_id);
    // pros::delay(1);
    // chassis.straight_drive(-1000, 0, 12000, 10000);
    // Sensors::ball_detector.start_logging();
    // Logger::stop_queueing();
    while(1)
    {
        // print encoder values

        // std::cout << "r: " << Sensors::right_encoder.get_position(r_id) << " | l: " << Sensors::left_encoder.get_position(l_id) << " | s: " << Sensors::strafe_encoder.get_position(s_id) << "\n";
        // double delta_theta = chassis.calc_delta_theta(prev_angle, ref_angle, Sensors::left_encoder.get_position(l_id) - prev_l, Sensors::right_encoder.get_position(r_id) - prev_r);
        // prev_angle = prev_angle + delta_theta;
        // prev_l = Sensors::left_encoder.get_position(l_id);
        // prev_r = Sensors::right_encoder.get_position(r_id);
        //
        // std::cout << "delta theta: " << delta_theta << "  |  new angle: " << prev_angle << "\n";
        // std::cout << tracker->get_position().x_pos << " " << tracker->get_position().y_pos << " " << tracker->to_degrees(tracker->get_position().theta) << "\n";
        lcd.update_labels();
        // server.handle_requests(50);
        // std::cout << "handling requests\n";
        // logger.dump();

        pros::delay(20);
    }
}
```

```cpp
/**
 * @file: ./RobotCode/src/Autons.hpp
 * @author: Aiden Carney
 * @reviewed_on: 12/5/19
 * @reviewed_by: Aiden Carney
 *
 * contains class that holds data about the autonomous period as well as
 * structs for configuration data
 */

#ifndef __AUTONS_HPP__
#define __AUTONS_HPP__

#include <unordered_map>

#include "main.h"


typedef struct
{
    bool use_hardcoded = 0;
    bool gyro_turn = 1;
    bool accelleration_ctrl = 1;
    bool check_motor_tmp = 0;
    bool use_previous_macros = 1;
    bool record = 0;
} autonConfig;



/**
 * @see: Motors.hpp
 * @see: ./objects/lcdCode
 *
 * contains data for the autonomous period as well as functions to run the
 * selected autonomous
 */
class Autons
{
    private:
        static int selected_number;

    public:
        Autons();
        ~Autons();


        int debug_auton_num;      //change if more autons are added
                                  //debugger should be last option
        int driver_control_num;

        const std::unordered_map <int, const char*> AUTONOMOUS_NAMES = {
            {1, "Driver Control"},           //used to find name of auton
            {2, "one_pt"},                   //to keep title the same
            {3, "skills-47"},
            {4, "skills-66"},
            {5, "blue north"},
            {6, "blue north 2"},
            {7, "red north"},
            {8, "Debugger"}
        };
        const std::unordered_map <int, const char*> AUTONOMOUS_DESCRIPTIONS = {   //used to find color of auton
            {1, "goes directly to\ndriver control"},                              //selected to keep background the same
            {2, "drives forward and\nbackwards"},
            {3, "skills auton that scores 47 points"},
            {4, "skills auton that scores 66 points"},
            {5, "Goes to cap middle wall\ntower and then cycle\nstarting tower"},
            {6, "cycles closest tower, turns left"},
            {7, "cylces closest tower, turns right"},
            {8, "opens debugger"}
        };
        const std::unordered_map <int, std::string> AUTONOMOUS_COLORS = {
            {1, "none"},                     //used to find color of auton
            {2, "none"},                     //selected to keep background the same
            {3, "none"},
            {4, "none"},
            {5, "blue"},
            {6, "blue"},
            {7, "red"},
            {8, "none"}
        };

        void set_autonomous_number(int n);
        int get_autonomous_number();


        /**
         * @return: None
         *
         * @see: Motors.hpp
         * @see: Chassis.hpp
         *
         * get robot ready for auton
         */
        void deploy();

        /**
         * @param: autonConfig cnfg -> the configuration to use for the auton
         * @return: None
         *
         * @see: Motors.hpp
         *
         * drives forward
         */
        void one_pt();

        /**
         * @param: autonConfig cnfg -> the configuration to use for the auton
         * @return: None
         *
         * @see: Motors.hpp
         *
         * runs skills
```

```
     */
    void skills();

    void skills2();


    void blue_north();
    void blue_north_2();
    void red_north();

    void run_autonomous();
};




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/Autons.cpp
 * @author: Aiden Carney
 * @reviewed_on: 12/5/19
 * @reviewed_by: Aiden Carney
 *
 * @see: Autons.hpp
 *
 * contains implementation for autonomous options
 */

#include <unordered_map>

#include "main.h"

#include "Autons.hpp"
#include "objects/motors/Motors.hpp"
#include "objects/motors/MotorThread.hpp"
#include "objects/position_tracking/PositionTracker.hpp"
#include "objects/subsystems/chassis.hpp"
#include "objects/subsystems/Indexer.hpp"
#include "objects/subsystems/intakes.hpp"
#include "objects/lcdCode/DriverControl/AutonomousLCD.hpp"


int Autons::selected_number = 1;

Autons::Autons( )
{
    debug_auton_num = 8;
    driver_control_num = 1;
}



Autons::~Autons( ) {

}

void Autons::set_autonomous_number(int n) {
    selected_number = n;
}

int Autons::get_autonomous_number() {
    return selected_number;
}

/**
 * deploys by outtaking and running top roller
 */
void Autons::deploy() {
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "none");
    Intakes intakes(Motors::left_intake, Motors::right_intake);

    intakes.rocket_outwards();
    indexer.run_upper_roller();

    pros::delay(1000);

    intakes.stop();
    indexer.stop();
}


/**
 * drives forward to score in the zone, then drive backward
 * to stop touching the cube
 */
void Autons::one_pt() {

    deploy();
}




/**
 * runs unit test
 * 180 degree, 90 degree, 45 degree, 45 degree
 * tilter movement
 * straight drive moving
 */
void Autons::skills() {
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
    Intakes intakes(Motors::left_intake, Motors::right_intake);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->start_thread();
    tracker->enable_imu();
    tracker->set_log_level(0);
    tracker->set_position({0, 0, 0});

    deploy();

    // tower one
    int uid = chassis.profiled_straight_drive(1000, 450, 3000, true);
    while(!chassis.is_finished(uid)) {
        intakes.intake();
        indexer.auto_increment();
        pros::delay(10);
    }

    intakes.stop();
    indexer.stop();

    chassis.turn_left(82, 450, 2500);

    chassis.profiled_straight_drive(975, 450, 2500);

    indexer.index();
    pros::delay(300);
    indexer.stop();

    // tower two
    chassis.profiled_straight_drive(-1000, 450, 2500);
    chassis.turn_right(107, 450, 3500);

    intakes.hold_outward();
    chassis.pid_straight_drive(1225, 0, 450, 3500);

    uid = chassis.pid_straight_drive(500, 0, 300, 3000, true);
    while(!chassis.is_finished(uid)) {
        intakes.intake();
        indexer.auto_increment();
        pros::delay(10);
    }

    intakes.stop();
    indexer.stop();

    chassis.turn_left(62, 450, 2000);

    chassis.pid_straight_drive(345, 0, 450, 2000);

    indexer.index();
    pros::delay(300);
    indexer.stop();

    pros::delay(500);

    indexer.index();
    pros::delay(300);
    indexer.stop();

    // tower 3
    chassis.pid_straight_drive(-450, 0, 450, 2000);
    chassis.turn_right(78, 450, 2000);

    intakes.hold_outward();
    chassis.pid_straight_drive(1500, 0, 450, 3500);

    uid = chassis.pid_straight_drive(600, 0, 300, 3000, true);
    while(!chassis.is_finished(uid)) {
        intakes.intake();
        indexer.auto_increment();
        pros::delay(10);
    }


    chassis.turn_left(62, 450, 2000);

    chassis.pid_straight_drive(1175, 0, 450, 2000);

    indexer.index();
    pros::delay(300);
    indexer.stop();

    //
    // chassis.drive_to_point(15, 15, 0, 0, 150, INT32_MAX, false);
    // chassis.pid_straight_drive(400);
}

void Autons::skills2() {
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
    Intakes intakes(Motors::left_intake, Motors::right_intake);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->start_thread();
    tracker->enable_imu();
    tracker->set_log_level(0);
    tracker->set_position({0, 0, 0});

    deploy();

    // tower one
    int uid = chassis.profiled_straight_drive(1000, 450, 3000, true);
    while(!chassis.is_finished(uid)) {
        intakes.intake();
        indexer.auto_increment();
        pros::delay(10);
    }

    intakes.stop();
    indexer.stop();

    chassis.turn_left(82, 450, 2500);

    chassis.profiled_straight_drive(975, 450, 2500);

    indexer.index();
    pros::delay(300);
    indexer.stop();

    // tower two
    chassis.profiled_straight_drive(-1000, 450, 2500);
    chassis.turn_right(107, 450, 3500);

    uid = chassis.pid_straight_drive(2000, 0, 450, 5000, true);
    while(!chassis.is_finished(uid)) {
        intakes.intake();
        indexer.auto_increment();
        pros::delay(10);
    }
}



void Autons::blue_north() {
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
    indexer.update_filter_color("red");
    Intakes intakes(Motors::left_intake, Motors::right_intake);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->start_thread();
    tracker->enable_imu();
    tracker->set_log_level(0);
    tracker->set_position({0, 0, 0});

    deploy();

    chassis.drive_to_point(0, 27);

    chassis.drive_to_point(27.2, 5.5, 0, 1, 125, INT32_MAX, false);

    chassis.turn_right(51);
    chassis.pid_straight_drive(235);

    // chassis.pid_straight_drive(200, 0, 80, INT32_MAX, false, false);

    for(int i=0; i < 50; i++) {  // score preload
        indexer.auto_index();
        pros::delay(10);
    }

    chassis.drive_to_point(0, 27);

    intakes.hold_outward();

    chassis.drive_to_point(-17.7, 9.7, 0, 1, 100, INT32_MAX, false);
    // chassis.pid_straight_drive(1100, 0, 150, INT32_MAX, false, false);

    int uid = chassis.pid_straight_drive(450, 0, 80, 2000, true, false);
    while(!chassis.is_finished(uid)) {
        indexer.auto_increment();
        intakes.intake();
    }

    // intakes.intake();
    // indexer.index_until_filtered();
    // pros::delay(100);
    // intakes.stop();
    //
    for(int i=0; i < 100; i++) {  // index a little bit longer
        indexer.auto_index();
        pros::delay(10);
    }

    chassis.pid_straight_drive(-500, 0, 200, INT32_MAX, false, false);
}


void Autons::blue_north_2() {
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
    indexer.update_filter_color("blue");
    Intakes intakes(Motors::left_intake, Motors::right_intake);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->start_thread();
    tracker->enable_imu();
    tracker->set_log_level(0);
    tracker->set_position({0, 0, 0});

    deploy();

    chassis.pid_straight_drive(560, 0, 100);
    chassis.turn_left(136, 100, 5000);

    intakes.hold_outward();
    // chassis.pid_straight_drive(300, 0, 80, 2000, false, true);


    intakes.intake();
    indexer.auto_increment();
    chassis.pid_straight_drive(900, 0, 80, 2000, false, false);
    indexer.stop();
    // indexer.index_until_filtered();
    intakes.stop();

    for(int i=0; i < 150; i++) {  // index a little bit longer
        // intakes.intake();
        indexer.index_no_backboard();
        pros::delay(10);
    }

    //
    // for(int i=0; i < 100; i++) {  // index a little bit longer
    //     // intakes.intake();
    //     indexer.index();
    //     pros::delay(10);
    // }

    indexer.fix_ball();
    pros::delay(1000);
    indexer.stop();
    for(int i=0; i < 40; i++) {  // index a little bit longer
        intakes.intake();
        // indexer.auto_index();
        pros::delay(10);
    }
    intakes.stop();

    intakes.hold_outward();
    chassis.pid_straight_drive(-400, 0, 80);
}


void Autons::red_north() {
    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, "blue");
    indexer.update_filter_color("blue");
    Intakes intakes(Motors::left_intake, Motors::right_intake);
    PositionTracker* tracker = PositionTracker::get_instance();
    tracker->start_thread();
    tracker->enable_imu();
    tracker->set_log_level(0);
    tracker->set_position({0, 0, 0});

    deploy();

    chassis.pid_straight_drive(560, 0, 100);
    chassis.turn_right(136, 100, 5000);

    intakes.hold_outward();
    // chassis.pid_straight_drive(300, 0, 80, 2000, false, true);
    

    intakes.intake();
    indexer.auto_increment();
    chassis.pid_straight_drive(900, 0, 80, 2000, false, false);
    indexer.stop();
    // indexer.index_until_filtered();
    intakes.stop();

    for(int i=0; i < 150; i++) {  // index a little bit longer
        // intakes.intake();
        indexer.index_no_backboard();
        pros::delay(10);
    }

    //
    // for(int i=0; i < 100; i++) {  // index a little bit longer
    //     // intakes.intake();
    //     indexer.index();
    //     pros::delay(10);
    // }

    indexer.fix_ball();
    pros::delay(1000);
    indexer.stop();
    for(int i=0; i < 40; i++) {  // index a little bit longer
        intakes.intake();
        // indexer.auto_index();
        pros::delay(10);
    }
    intakes.stop();

    intakes.hold_outward();
    chassis.pid_straight_drive(-400, 0, 80);
}

void Autons::run_autonomous() {
    switch(selected_number)
    {
        case 1:
           break;

        case 2:
            one_pt();
            break;

        case 3:
            skills();
            break;

        case 4:
            skills2();
            break;

        case 5:
            blue_north();
            break;

        case 6:
            blue_north_2();
            break;

        case 7:
            red_north();
    }
}
```

```cpp
/**
 * @file: ./RobotCode/src/Configuration.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains class static variables for runtime configuration
 *
 */

#ifndef __CONFIGURATION_HPP__
#define __CONFIGURATION_HPP__

#include <iostream>
#include <vector>

#include "main.h"

#include "../lib/json.hpp"


#define LEFT_ENC_TOP_PORT        'G'
#define LEFT_ENC_BOTTOM_PORT     'H'
#define RIGHT_ENC_TOP_PORT       'A'
#define RIGHT_ENC_BOTTOM_PORT    'B'
#define STRAFE_ENC_TOP_PORT      'E'
#define STRAFE_ENC_BOTTOM_PORT   'F'
#define DETECTOR_MIDDLE_PORT     'C'
#define POTENTIOMETER_PORT       'Z'

#define DETECTOR_BOTTOM_PORT     'Z'  // no port available but still wanted in code
#define DETECTOR_TOP_PORT        'D'  // no port available but still wanted in code

#define OPTICAL_PORT             5
#define IMU_PORT                 10


typedef struct
{
    double kP = 0;
    double kI = 0;
    double kD = 0;
    double I_max = 0;
    void print() {
        std::cout << "kP: " << this->kP << "\n";
        std::cout << "kI: " << this->kI << "\n";
        std::cout << "kD: " << this->kD << "\n";
        std::cout << "I_max: " << this->I_max << "\n";
    };
} pid;



/**
 * @see: ../lib/json.hpp
 *
 * Singleton class
 * contains class to read data from config file on sd card for better runtime config
 * useful so that a clean build is not always necessary
 * contains static variables used throughout rest of project
 */
class Configuration
{
    private:
        Configuration();
        static Configuration *config_obj;

    public:
        ~Configuration();

        /**
         * @return: Configuration -> instance of class to be used throughout program
         *
         * give user the instance of the singleton class or creates it if it does
         * not yet exist
         */
        static Configuration* get_instance();

        pid internal_motor_pid;
        pid lift_pid;
        pid chassis_pid;

        int front_right_port;
        int back_left_port;
        int front_left_port;
        int back_right_port;
        int left_intake_port;
        int right_intake_port;
        int upper_indexer_port;
        int lower_indexer_port;

        bool front_right_reversed;
        bool back_left_reversed;
        bool front_left_reversed;
        bool back_right_reversed;
        bool left_intake_reversed;
        bool right_intake_reversed;
        bool upper_indexer_reversed;
        bool lower_indexer_reversed;

        std::vector<int> lift_setpoints;
        std::vector<int> tilter_setpoints;
        std::vector<int> intake_speeds;

        int filter_threshold;
        std::string filter_color;  // color to remove


        /**
         * @return: int -> 1 if file was successfully read, 0 if no changes were made
         *
         * @see: ../lib/json.hpp
         *
```

```
     * parses json file looking for data to set variables to
     */
    int init();


    /**
     * @return: None
     *
     * @see: typedef struct pid
     *
     * prints all the variables in the class
     * used for debugging to make sure values are what they are
     * supposed to be
     */
    void print_config_options();

};




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/Configuration.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * @see: Configuration.hpp
 *
 * contains implementation for configuration class
 */

#include <vector>
#include <fstream>

#include "main.h"

#include "../lib/json.hpp"
#include "Configuration.hpp"


Configuration *Configuration::config_obj = NULL;


Configuration::Configuration( )
{
    //set default values for constants in case file can’t be read
    internal_motor_pid.kP = 30;
    internal_motor_pid.kI = 37;
    internal_motor_pid.kD = 11;
    internal_motor_pid.I_max = INT32_MAX;

    lift_pid.kP = .1;
    lift_pid.kI = 0.0001;
    lift_pid.kD = 0;
    lift_pid.I_max = INT32_MAX;

    chassis_pid.kP = .0035;
    chassis_pid.kI = 0;
    chassis_pid.kD = 0;
    chassis_pid.I_max = INT32_MAX;

    //536C motor config
    front_right_port = 12;
    back_left_port = 15;
    front_left_port = 16;
    back_right_port = 13;
    left_intake_port = 8;
    right_intake_port = 7;
    upper_indexer_port = 9;
    lower_indexer_port = 17;

    front_right_reversed = 1;
    back_left_reversed = 1;
    front_left_reversed = 0;
    back_right_reversed = 0;
    left_intake_reversed = 0;
    right_intake_reversed = 1;
    upper_indexer_reversed = 0;
    lower_indexer_reversed = 0;

    filter_threshold = 2880;
    filter_color = "blue";


    //536D motor config
    // front_right_port = 13;
    // back_left_port = 1;
    // front_left_port = 20;
    // back_right_port = 19;
    // left_intake_port = 2;
    // right_intake_port = 11;
    // tilter_port = 17;
    // lift_port = 12;
    //
    // front_right_reversed = 1;
    // back_left_reversed = 0;
    // front_left_reversed = 0;
    // back_right_reversed = 1;
    // left_intake_reversed = 0;
    // right_intake_reversed = 1;
    // tilter_reversed = 1;
    // lift_reversed = 0;


    //536A motor config
    // front_right_port = 20;
    // back_left_port = 2;
    // front_left_port = 5;
    // back_right_port = 4;
    // left_intake_port = 8;
    // right_intake_port = 1;
    // tilter_port = 13;
    // lift_port = 11;
    //
    // front_right_reversed = 1;
    // back_left_reversed = 1;
    // front_left_reversed = 1;
    // back_right_reversed = 1;
    // left_intake_reversed = 0;
    // right_intake_reversed = 1;
    // tilter_reversed = 0;
    // lift_reversed = 0;

    std::vector<int> vec1 {100, 300, 400, 500};
    std::vector<int> vec2 {100, 300, 400, 500};
    std::vector<int> vec3 {-63, -30, 0, 30, 63};

    tilter_setpoints = vec1;
    lift_setpoints = vec2;
    intake_speeds = vec3;
}


```

```
Configuration::~Configuration( )
{

}



/**
 * inits object if object is not already initialized based on a static bool
 * sets bool if it is not set
 */
Configuration* Configuration::get_instance()
{
    if ( config_obj == NULL )
    {
        config_obj = new Configuration;
    }
    return config_obj;
}


/**
 * reads json file into memory in the form of a json object supported by
 * a library
 * parses json array to get pid constants and setpoints by looking at the size
 * sets other variables by looking at their value
 */
int Configuration::init()
{
    std::ifstream input("/usd/config.json"); //open file with library
    if ( input.fail() )
    {
        std::cerr << "[ERROR], " << pros::millis() << ", configuration file could not be opened\n";
        return 0;
    }
    nlohmann::json contents;
    input >> contents;


    std::vector<double> constants1;  //read pid constants for different systems
    std::vector<double> constants2;

    for ( int i1 = 0; i1 < 4; i1++)
    {
        double value1 = contents["internal_motor_pid"][i1];
        double value2 = contents["lift_pid"][i1];

        std::cout << value1 << "\n";
        constants1.push_back(value1);
        constants2.push_back(value2);
    }

    internal_motor_pid.kP = constants1.at(0);
    internal_motor_pid.kI = constants1.at(1);
    internal_motor_pid.kD = constants1.at(2);
    internal_motor_pid.I_max = constants1.at(3);

    lift_pid.kP = constants2.at(0);
    lift_pid.kI = constants2.at(1);
    lift_pid.kD = constants2.at(2);
    lift_pid.I_max = constants2.at(3);




    front_right_port = contents["front_right_port"];  //read motor port definitions
    back_left_port = contents["back_left_port"];
    front_left_port = contents["front_left_port"];
    back_right_port = contents["back_right_port"];
    left_intake_port = contents["left_intake_port"];
    right_intake_port = contents["right_intake_port"];
    upper_indexer_port = contents["upper_indexer_port"];
    lower_indexer_port = contents["lower_indexer_port"];

    front_right_reversed = contents["front_right_reversed"] == 1 ? true : false; //read motor port reversals
    back_left_reversed = contents["back_left_reversed"] == 1 ? true : false;
    front_left_reversed = contents["front_left_reversed"] == 1 ? true : false;
    back_right_reversed = contents["back_right_reversed"] == 1 ? true : false;
    left_intake_reversed = contents["left_intake_reversed"] == 1 ? true : false;
    right_intake_reversed = contents["right_intake_reversed"] == 1 ? true : false;
    upper_indexer_reversed = contents["upper_indexer_reversed"] == 1 ? true : false;
    lower_indexer_reversed = contents["lower_indexer_reversed"] == 1 ? true : false;

    filter_threshold = contents["filter_threshold"];

    lift_setpoints.clear();
    for ( int i2 = 0; i2 < contents["lift_setpoints"].size(); i2++)
    {
        lift_setpoints.push_back(contents["lift_setpoints"][i2]);

    }

    intake_speeds.clear();
    for ( int i3 = 0; i3 < contents["intake_speeds"].size(); i3++)
    {
        intake_speeds.push_back(contents["intake_speeds"][i3]);

    }

    return 1;
}




/**
 * prints all the variables and what they are so that they can be debugged
 * makes use of internal pid print function
 */
void Configuration::print_config_options()
{
    std::cout << "drive PID constants\n";
```

```cpp
    internal_motor_pid.print();
    std::cout << "lift PID constants\n";
    lift_pid.print();

    std::cout << "\n";

    std::cout << "front_right_port: " << front_right_port << "\n";
    std::cout << "back_left_port: " << back_left_port << "\n";
    std::cout << "front_left_port: " << front_left_port << "\n";
    std::cout << "back_right_port: " << back_right_port << "\n";
    std::cout << "left_intake_port: " << left_intake_port << "\n";
    std::cout << "right_intake_port: " << right_intake_port << "\n";
    std::cout << "upper_indexer_port: " << upper_indexer_port << "\n";
    std::cout << "lower_indexer_port: " << lower_indexer_port << "\n";

    std::cout << "front_right_reversed: " << front_right_reversed << "\n";
    std::cout << "back_left_reversed: " << back_left_reversed << "\n";
    std::cout << "front_left_reversed: " << front_left_reversed << "\n";
    std::cout << "back_right_reversed: " << back_right_reversed << "\n";
    std::cout << "left_intake_reversed: " << left_intake_reversed << "\n";
    std::cout << "right_intake_reversed: " << right_intake_reversed << "\n";
    std::cout << "upper_indexer_reversed: " << upper_indexer_reversed << "\n";
    std::cout << "lower_indexer_reversed: " << lower_indexer_reversed << "\n";

    std::cout << "\nfilter threshold: " << filter_threshold << "\n";


    std::cout << "\nlift_setpoints: ";
    for ( int i = 0; i < lift_setpoints.size() - 1; i++ )
    {
        std::cout << lift_setpoints.at(i) << ", ";
    }
    std::cout << lift_setpoints.at(lift_setpoints.size() - 1) << "\n";


    std::cout << "\nintake_speeds: ";
    for ( int i = 0; i < intake_speeds.size() - 1; i++ )
    {
        std::cout << intake_speeds.at(i) << ", ";
    }
    std::cout << intake_speeds.at(intake_speeds.size() - 1) << "\n";
}
```

```
/**
 * @file: ./RobotCode/src/DriverControl.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * Contains robot move functions. Meant to be run in pros task
 *
 */

#ifndef __DRIVERCONTROL_HPP__
#define __DRIVERCONTROL_HPP__

#include <cstdlib>

#include "../include/main.h"

#include "objects/subsystems/chassis.hpp"
#include "objects/controller/controller.hpp"
#include "objects/motors/Motors.hpp"

#define AUTON_DEBUG


/**
 * @param: void* -> not used
 * @return: None
 *
 * @see: Motors.hpp
 * @see: Controller.hpp
 *
 * meant to be run on task
 * function cycles through and allows user to controll robot
 *
 */
void driver_control(void*);


#endif
```

```

/**
 * @file: ./RobotCode/src/DriverControl.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: DriverControl.hpp
 *
 */

#include <cstdlib>
#include <cmath>

#include "../include/main.h"

#include "objects/lcdCode/gui.hpp"
#include "objects/subsystems/chassis.hpp"
#include "objects/subsystems/Indexer.hpp"
#include "objects/subsystems/intakes.hpp"
#include "objects/controller/controller.hpp"
#include "objects/motors/Motors.hpp"
#include "objects/sensors/Sensors.hpp"
#include "Configuration.hpp"
#include "DriverControl.hpp"
#include "Configuration.hpp"



/**
 * uses if statements to control motor based on controller settings
 * checks to set it to zero based on if static var in Motors class allows it
 * this is to make sure that other tasks can controll Motors too
 */
void driver_control(void*)
{
    Configuration *config = Configuration::get_instance();

    Controller controllers;

    Chassis chassis( Motors::front_left, Motors::front_right, Motors::back_left, Motors::back_right, Sensors::left_encoder, Sensors::right_encoder, 16, 3/5);
    Indexer indexer(Motors::upper_indexer, Motors::lower_indexer, Sensors::ball_detector, config->filter_color);
    Intakes intakes(Motors::left_intake, Motors::right_intake);

    int left_analog_y = 0;
    int right_analog_y = 0;

    bool auto_filter = true;
    bool hold_intakes_out = true;
    int intake_start_time = 0;  // no possible way to think indexer should run at the start of driver control

    controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);

    while ( true ) {
        controllers.update_button_history();

    // section for front roller intake movement
        if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_R1)) {  // define velocity for main intake
            intakes.intake();
            intake_start_time = pros::millis();
        } else if(hold_intakes_out){  // rest state is outward with motor power
            intakes.hold_outward();
        } else {  // rest state is no motor power
            intakes.stop();
        }

        if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_R2)) {
            hold_intakes_out = !hold_intakes_out;
        }

    // section for indexer motion
        if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L1) && auto_filter) {  // define movement for indexer subsystem
            indexer.auto_index();
        } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L1) && !auto_filter) {
            indexer.index();
        } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_LEFT)) {
            indexer.filter();
        } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L2) && auto_filter) {
            indexer.auto_increment();
        } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_L2) && !auto_filter) {
            indexer.increment();
        } else if(controllers.master.get_digital_new_press(pros::E_CONTROLLER_DIGITAL_RIGHT)) {
            indexer.fix_ball(true);
        } else if(controllers.btn_is_pressing(pros::E_CONTROLLER_DIGITAL_X)) {
            indexer.index_no_backboard();
        } else if (pros::millis() < intake_start_time + 1000) {
            indexer.auto_increment();
        } else {
            indexer.hard_stop();
        }

        if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_LEFT)) {
            auto_filter = !auto_filter;
            if(auto_filter) {  // give different message if not auto filtering
                controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);
            } else {
                controllers.master.print(0, 0, "Man Filter %s     ", config->filter_color);
            }
        }

    // section for setting filter color
        if(controllers.btn_get_release(pros::E_CONTROLLER_DIGITAL_A)) {  // cycle filter colors
            if(config->filter_color == "red") {
                config->filter_color = "blue";
            } else if(config->filter_color == "blue") {
                config->filter_color = "none";
            } else if(config->filter_color == "none") {
                config->filter_color = "red";
            }

            if(auto_filter) {  // give different message if not auto filtering
                controllers.master.print(0, 0, "Auto Filter %s     ", config->filter_color);
            } else {
                controllers.master.print(0, 0, "Man Filter %s     ", config->filter_color);
            }
            std::cout << "filtering " << config->filter_color << "\n";
            indexer.update_filter_color(config->filter_color);
        }


    // section for chassis movement
        if(std::abs(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) < 5) {   // define deadzone for left analog input on the y axis
            left_analog_y = 0;
        } else {
            left_analog_y = controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y);
        }

        if(std::abs(controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) < 5) {   // define deadzone for right analog input on the y axis
            right_analog_y = 0;
        } else {
            right_analog_y = controllers.master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y);
        }

        // float corrected_speed = ( .000043326431866017 * std::pow( leftDriveSpeed, 3 ) ) + ( 0.29594689028631 * leftDriveSpeed);
        Motors::front_left.user_move(left_analog_y);
        Motors::back_left.user_move(left_analog_y);

        // float corrected_speed = ( .000043326431866017 * std::pow( rightDriveSpeed, 3 ) ) + ( 0.29594689028631 * rightDriveSpeed);
        Motors::front_right.user_move(right_analog_y);
        Motors::back_right.user_move(right_analog_y);


        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN) ) {
            Autons auton;
            auton.deploy();
        }

        pros::delay(5);

    }
}
```

```
/**
 * @file: ./RobotCode/src/controller/controller.hpp
 * @author: Aiden Carney
 * @reviewed_on: 11/8/19
 * @reviewed_by: Aiden Carney
 * TODO: remove static members and replace with singleton class
 *
 * contains class that has data about controllers
 */

#include <unordered_map>
#include <string>
#include <deque>

#include "../../../include/main.h"
#include "../../../include/api.h"
#include "../../../include/pros/rtos.hpp"
#include "../../../include/pros/motors.hpp"

#ifndef __CONTROLLER_HPP__
#define __CONTROLLER_HPP__

/**
 * contains controller objects as well as unordered maps that hold information
 * about what the controller does
 * TODO: move unordered maps to configuration and make more robust so it does not have to be updated
 */
class Controller
{
    private:
        static std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > master_btn_history;
        static std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > partner_btn_history;

    public:
        Controller();
        ~Controller();

        static pros::Controller master;
        static pros::Controller partner;

        static std::unordered_map <pros::controller_analog_e_t, std::string> MASTER_CONTROLLER_ANALOG_MAPPINGS;
        static std::unordered_map <pros::controller_analog_e_t, std::string> PARTNER_CONTROLLER_ANALOG_MAPPINGS;
        static std::unordered_map <pros::controller_digital_e_t, std::string> MASTER_CONTROLLER_DIGITAL_MAPPINGS;
        static std::unordered_map <pros::controller_digital_e_t, std::string>  PARTNER_CONTROLLER_DIGITAL_MAPPINGS;

        void update_button_history();

        bool btn_get_release(pros::controller_digital_e_t btn, int controller=0);
        bool btn_get_start_press(pros::controller_digital_e_t btn, int controller=0);
        bool btn_is_pressing(pros::controller_digital_e_t btn, int controller=0);

};




#endif
```

```
/**
 * @file: ./RobotCode/src/controller/controller.cpp
 * @author: Aiden Carney
 * @reviewed_on: 11/8/19
 * @reviewed_by: Aiden Carney
 *
 * @see: controller.hpp
 *
 * contains definitions for static members of class
 */

#include <unordered_map>
#include <string>

#include "../../../include/main.h"
#include "../../../include/api.h"
#include "../../../include/pros/rtos.hpp"
#include "../../../include/pros/motors.hpp"

#include "controller.hpp"



std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > Controller::master_btn_history = {
    {pros::E_CONTROLLER_DIGITAL_L1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_L2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_UP, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_DOWN, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_LEFT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_X, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_B, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_Y, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_A, new std::deque<bool>}
};

std::unordered_map <pros::controller_digital_e_t, std::deque<bool>* > Controller::partner_btn_history = {
    {pros::E_CONTROLLER_DIGITAL_L1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_L2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R2, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_R1, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_UP, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_DOWN, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_LEFT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_X, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_B, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_Y, new std::deque<bool>},
    {pros::E_CONTROLLER_DIGITAL_A, new std::deque<bool>}
};

pros::Controller Controller::master(pros::E_CONTROLLER_MASTER);
pros::Controller Controller::partner(pros::E_CONTROLLER_PARTNER);

//mappings for each controller
std::unordered_map <pros::controller_analog_e_t, std::string> Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS = {
    {pros::E_CONTROLLER_ANALOG_LEFT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_LEFT_Y, "Left Side Chassis"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_Y, "Right Side Chassis"}
};

std::unordered_map <pros::controller_analog_e_t, std::string> Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS = {
    {pros::E_CONTROLLER_ANALOG_LEFT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_LEFT_Y, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_X, "None"},
    {pros::E_CONTROLLER_ANALOG_RIGHT_Y, "None"}
};

std::unordered_map <pros::controller_digital_e_t, std::string> Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS = {
    {pros::E_CONTROLLER_DIGITAL_L1, "lift up"},
    {pros::E_CONTROLLER_DIGITAL_L2, "lift down"},
    {pros::E_CONTROLLER_DIGITAL_R2, "Outake"},
    {pros::E_CONTROLLER_DIGITAL_R1, "Intake"},
    {pros::E_CONTROLLER_DIGITAL_UP, "Tilter up"},
    {pros::E_CONTROLLER_DIGITAL_DOWN, "Tilter down"},
    {pros::E_CONTROLLER_DIGITAL_LEFT, "None"},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, "None"},
    {pros::E_CONTROLLER_DIGITAL_X, "auto deploy"},
    {pros::E_CONTROLLER_DIGITAL_B, "Toggle brakes"},
    {pros::E_CONTROLLER_DIGITAL_Y, "auto dump"},
    {pros::E_CONTROLLER_DIGITAL_A,  "run auton"}
};

std::unordered_map <pros::controller_digital_e_t, std::string>  Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS = {
    {pros::E_CONTROLLER_DIGITAL_L1, "None"},
    {pros::E_CONTROLLER_DIGITAL_L2, "None"},
    {pros::E_CONTROLLER_DIGITAL_R2, "None"},
    {pros::E_CONTROLLER_DIGITAL_R1, "None"},
    {pros::E_CONTROLLER_DIGITAL_UP, "None"},
    {pros::E_CONTROLLER_DIGITAL_DOWN, "None"},
    {pros::E_CONTROLLER_DIGITAL_LEFT, "None"},
    {pros::E_CONTROLLER_DIGITAL_RIGHT, "None"},
    {pros::E_CONTROLLER_DIGITAL_X, "None"},
    {pros::E_CONTROLLER_DIGITAL_B, "None"},
    {pros::E_CONTROLLER_DIGITAL_Y, "None"},
    {pros::E_CONTROLLER_DIGITAL_A,  "None"}
};




Controller::Controller()
{

}

Controller::~Controller()
{

}
```

```


void Controller::update_button_history()
{
    for (std::pair<pros::controller_digital_e_t, std::deque<bool>*> element : master_btn_history) {
        element.second->push_back(master.get_digital(element.first));
        if(element.second->size() > 3) {
            element.second->pop_front();
        }

        while(element.second->size() < 3) {  // forces deque size to be no less than 3
            element.second->push_back(master.get_digital(element.first));
        }
    }

    for (std::pair<pros::controller_digital_e_t, std::deque<bool>*> element : partner_btn_history) {
        element.second->push_back(master.get_digital(element.first));
        if(element.second->size() > 3) {
            element.second->pop_front();
        }

        while(element.second->size() < 3) {  // forces deque size to be no less than 3
            element.second->push_back(master.get_digital(element.first));
        }
    }
}


bool Controller::btn_get_release(pros::controller_digital_e_t btn, int controller /** 0 **/) {
    bool pressed_and_released = false;
    if(!controller) {
        if(master_btn_history.at(btn)->at(1) and !master_btn_history.at(btn)->at(2)) {
            pressed_and_released = true;
        }
    }
    else {
        if(partner_btn_history.at(btn)->at(1) and !partner_btn_history.at(btn)->at(2)) {
            pressed_and_released = true;
        }
    }
    return pressed_and_released;
}


bool Controller::btn_get_start_press(pros::controller_digital_e_t btn, int controller /** 0 **/) {
    bool press_start = false;
    if(!controller) {
        if(!master_btn_history.at(btn)->at(1) and master_btn_history.at(btn)->at(2)) {
            press_start = true;
        }
    }
    else {
        if(!partner_btn_history.at(btn)->at(1) and partner_btn_history.at(btn)->at(2)) {
            press_start = true;
        }
    }

    return press_start;
}


bool Controller::btn_is_pressing(pros::controller_digital_e_t btn, int controller /** 0 **/) {
    bool pressing = false;
    if(!controller) {
        pressing = master_btn_history.at(btn)->at(2);
    }
    else {
        pressing = partner_btn_history.at(btn)->at(2);
    }

    return pressing;
}
```

```
/**
 * @file: ./RobotCode/src/objects/motors/Motor.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 * TODO:
 *
 * contains a wrapper class for a pros::Motor
 */

#ifndef __MOTOR_HPP__
#define __MOTOR_HPP__

#include <atomic>

#include "main.h"

#include "../../Configuration.hpp"


typedef enum {
    e_builtin_velocity_pid,
    e_voltage,
    e_custom_velocity_pid
} motor_mode;

/**
 * @see: pros::Motor
 * @see: ../../Configuration.hpp
 *
 * wrapper class for pros::Motor
 * contains implementation for better runtime port configuration
 * contains easier implementation for slew rate control
 * contains a pid velocity controller that can be enabled for consistent motor output
 */
class Motor
{
    private:
        int motor_port;

        pros::Motor *motor;

        int log_level;

        bool slew_enabled;
        int slew_rate;

        bool velocity_pid_enabled;
        int prev_velocity;
        pid internal_motor_pid;
        double integral;
        double prev_error;

        motor_mode mode = e_voltage;
        int voltage_setpoint;
        int prev_voltage_setpoint;
        int velocity_setpoint;


        int to_voltage(int velocity);
        int to_velocity(int voltage);

        /**
         * @param: int voltage -> the voltage to set the motor to
         * @return: int -> if setting motor voltage was successful or not
         *
         * @see: pros::Motor
         *
         * sets voltage of motor on interval [-12000,12000]
         */
        int set_voltage_setpoint( int voltage );


        int set_velocity_setpoint(int new_velocity);


        /**
         * @param int target -> the new voltage that could be requested
         * @param int previous -> the previous voltage to calculate change in voltage over time
         * @param in delta_t -> the time that has elapsed
         * @return: int -> the rate of the voltage set based on time elapsed and previous voltage
         *
         * @see: set_voltage_setpoint()
         *
         * calculates the rate of change of the voltage (mv/ms) that the new
         * voltage is trying to reach
         */
        int calc_target_rate( int target, int previous, int delta_t );

        /**
         * @param: int voltage -> a possible motor voltage in mv on interval [-12000,12000]
         * @return: int -> the corresponding velocity for a given voltage
         *
         * TODO: add checking if the voltage is not on the interval
         *
         * calculates the corresponding velocity for a given voltage in mv
         * the velocity range corresponds to the gearset of the motor
         * velocity ranges are ~20% higher than what they are rated for
         * because motors can achieve this velocity when supplied 12V
         */
        int calc_target_velocity( int voltage );

        /**
         * @return: int -> the voltage that the motor will be set at
         *
         * @see: slew rate functions contained in this class
         *
         * returns the target voltage based on the voltage set by the user
         * but that is either increased or decreased by the velocity pid if that
         * is enabled, or the slew rate code which limits the rate that the
         * voltage can increase
         */
        int get_target_voltage( int delta_t );


        std::atomic<bool> lock;  //protect motor functions from concurrent access
        bool allow_driver_control;


    public:
        Motor(int port, pros::motor_gearset_e_t gearset, bool reversed);
        Motor(int port, pros::motor_gearset_e_t gearset, bool reversed, pid pid_consts);
        ~Motor();

    //accessor functions

        /**
         * @return: double -> the actual velocity of the motor
         *
         * @see: pros::Motor
         *
         * returns the actual velocity of the motor as calculated internally by
         * the pros::Motor
         */
        double get_actual_velocity( );

        /**
         * @return: double -> the actual voltage of the motor
         *
         * @see: pros::Motor
         *
         * returns the actual voltage of the motor as calculated internally by
         * the pros::Motor
         */
        double get_actual_voltage( );

        /**
         * @return: int -> the actual current being supplied to the motor
         *
         * @see: pros::Motor
         *
         * returns the actual current being supplied to the motor as calculated internally by
         * the pros::Motor
         */
        int get_current_draw( );

        /**
         * @return: double -> the encoder value of the motor
         *
         * @see: pros::Motor
         *
         * returns the encoder position of the motor in degrees as calculated internally by
         * the pros::Motor
         */
        double get_encoder_position( );

        /**
         * @return: pros::motor_gearset_e_t -> the gearing of the motor
         *
         * @see: pros::Motor
         *
         * returns the gearset internally used by the motor per the pros::Motor
         */
        pros::motor_gearset_e_t get_gearset( );

        /**
         * @return: pros::motor_brake_mode_e_t -> the brakemode of the motor
         *
         * @see: pros::Motor
         *
         * returns the brakemode internally used by the motor per the pros::Motor
         */
        pros::motor_brake_mode_e_t get_brake_mode( );

        /**
         * @return: int -> the port of the motor
         *
         * returns the port that the motor is set on
         */
        int get_port( );

        /**
         * @return: pid -> struct of pid constants
         *
         * returns the pid constants in use by the motor
         */
        pid get_pid( );

        /**
         * @return: int -> the slew rate in use by the motor
         *
         * returns the slew rate in mV/ms in use by the motor
         */
        int get_slew_rate( );

        /**
         * @return: double -> the power drawn by the motor
         *
         * @see: pros::Motor
         *
         * returns the power that the motor is drawing in Watts
         */
        double get_power( );

        /**
         * @return: double -> the temperature of the motor
         *
         * @see: pros::Motor
         *
         * returns the temperature of the motor in degrees C
         */
        double get_temperature( );

        /**
         * @return: double -> the torque output of the motor
         *
```

```
     * @see: pros::Motor
     *
     * returns the torque output of the motor in Nm
     */
    double get_torque( );

    /**
     * @return: int -> the direction the motor is spinning
     *
     * @see: pros::Motor
     *
     * returns the direction of the motor
     * 1 for moving in the positive direction
     * -1 for moving in the negative direction
     */
    int get_direction( );

    /**
     * @return: int -> the efficiency of the motor
     *
     * @see: pros::Motor
     *
     * returns the efficiency of the motor as a percentage
     */
    int get_efficiency( );

    /**
     * @return: int -> if the motor is a rest
     *
     * @see: pros::Motor
     *
     * returns 1 if the motor is not moving and 0 if the motor is moving
     */
    int is_stopped( );

    /**
     * @return: int -> if the motor has been reversed or not
     *
     * @see: pros::Motor
     *
     * returns 1 if the motor has been reversed and 0 if the motor was not reversed
     */
    int is_reversed( );




  //setter functions
    /**
     * @param: int port -> the new port for the motor
     * @return: int -> if the change was successful or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int set_port( int port );

    /**
     * @return: int -> if function was successful or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int tare_encoder( );

    /**
     * @param: pros::motor_brake_mode_e_t -> the new brake mode for the motor
     * @return: int -> if the change was successful or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int set_brake_mode( pros::motor_brake_mode_e_t brake_mode );

    /**
     * @param: pros::motor_gearset_e_t -> the new gearset for the motor
     * @return: int -> if the change was successful or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int set_gearing( pros::motor_gearset_e_t gearset );

    /**
     * @return: int -> if motor was reversed or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int reverse_motor( );

    /**
     * @param: pid pid_consts -> the new pid constants for the motor
     * @return: int -> if the change was successful or not
     *
     * @see: pros::Motor
     *
     * returns 1 on success
     */
    int set_pid( pid pid_consts );

    /**
     * @param: int logging -> the new log level, 0-5, 5 is most verbose
     * @return: None
     *
     * @see: pros::Motor
     *
     * updates how verbose the logging is, 0 is no logging, 5 is very
```

```
         * verbose
         */
        void set_log_level( int logging );




    //movement functions
        /**
         * @param: int voltage -> the voltage to set the motor to
         * @return: int -> if setting motor voltage was successful or not
         *
         * @see: pros::Motor
         *
         * takes range [-127,127] and scales it to [-12000,12000]
         * makes it easier to map to controller input
         */
        int move( int voltage );

        /**
         * @param: int voltage -> the voltage to set the motor to
         * @return: int -> if the motor voltage was set or not
         *
         * @see: pros::Motor
         *
         * takes range [-127,127] and scales it to [-12000,12000]
         * makes it easier to map to controller input. This function acts as
         * a wrapper to move but with a check to see if the driver control lock
         * is taken. Use this function when using the controller in driver control
         * to set the motor value.
         */
        int user_move( int voltage );

        /**
         * @param: int velocity -> the velocity to set the motor to
         * @return: int -> if setting motor velocity was successful or not
         *
         * @see: pros::Motor
         *
         * takes range [-gearset_min + ~20%, gearset_min + ~20%] and scales it
         * to [-12000,12000]
         * used to make motor performance more consistent when velocity pid is
         * enabled
         * doesn't use built in pid because max motor ouput is limited by
         * approximately 20%
         */
        int move_velocity( int velocity );

        int set_voltage(int voltage);


    //slew rate control functions
        /**
         * @param: int rate -> the new slew rate in mv/ms
         * @return: int -> 1 on success
         *
         * @see: pros::Motor
         *
         * sets the new rate that the voltage can increase at
         * used for either acceleration control for less wheel slippage
         * or to protect motors from voltage spikes
         */
        int set_slew( int rate );

        /**
         * @return: None
         *
         * @see: pros::Motor
         *
         * sets slew rate code to be used to limit voltage change rate
         */
        void enable_slew( );

        /**
         * @return: None
         *
         * @see: pros::Motor
         *
         * sets slew rate code to not be used to limit voltage change rate
         */
        void disable_slew( );




    //velocity pid control functions
        /**
         * @return: None
         *
         * @see: pros::Motor
         *
         * sets new mode for the motor to follow
         */
        void set_motor_mode(motor_mode new_mode);



    //driver control lock setting and clearing functions
        /**
         * @return: None
         *
         * @see: pros::Motor
         *
         * sets a lock that can be used to prevent controller from being able
         * to set motor voltage
         */
        void enable_driver_control( );

        /**
         * @return: None
         *
         * @see: pros::Motor
         *
         * clears a lock that can be used to prevent controller from being able
         * to set motor voltage
         */
        void disable_driver_control( );

        /**
         * @return: int -> if driver control lock is cleared
         *
         * @see: pros::Motor
         *
         * returns 1 if lock is cleared, 0 otherwise
         */
        int driver_control_allowed( );


    //function to run on thread
        /**
         * @param: int delta_t -> the amount of time elapsed since the last time the function was called
         * @return: int -> the voltage to set the motor to
         *
         * @see: pros::Motor
         *
         * used to be run on long living thread so that motor can have PID and
         * slew rate code built into it easier
         * also contains logging implementation and adds to logger queue
         */
        int run( int delta_t );
};




#endif
```

```
/**
 * @file: ./RobotCode/src/objects/motors/Motor.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 * TODO: Clean up how logging message is set
 *
 * contains a implementation for wrapper class for a pros::Motor
 */

#include <atomic>

#include "main.h"

#include "../../Configuration.hpp"
#include "../serial/Logger.hpp"
#include "Motor.hpp"



Motor::Motor( int port, pros::motor_gearset_e_t gearset, bool reversed )
{
    lock = ATOMIC_VAR_INIT(false);
    allow_driver_control = true;

    while ( lock.exchange( true ) ); //aquire motor lock

    motor_port = port;

    motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);

    prev_velocity = 0;

    log_level = 0;

    slew_enabled = false;  // default slew rate to false
    slew_rate = 30;  //approx. 5% voltage per 20ms == 400ms to reach full voltage

    prev_voltage_setpoint = 0;
    voltage_setpoint = 0;
    velocity_setpoint = 0;

    Configuration *configuration = Configuration::get_instance();
    internal_motor_pid.kP = configuration->internal_motor_pid.kP;
    internal_motor_pid.kI = configuration->internal_motor_pid.kI;
    internal_motor_pid.kD = configuration->internal_motor_pid.kD;
    internal_motor_pid.I_max = configuration->internal_motor_pid.I_max;
    integral = 0;
    prev_error = 0;

    lock.exchange(false);
}


Motor::Motor(int port, pros::motor_gearset_e_t gearset, bool reversed, pid pid_consts)
{
    lock = ATOMIC_VAR_INIT(false);
    allow_driver_control = true;

    while ( lock.exchange( true ) ); //aquire motor lock

    motor_port = port;

    motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);

    prev_velocity = 0;

    log_level = 0;

    slew_enabled = false;
    slew_rate = 30;  //approx. 5% voltage per 20ms == 400ms to reach full voltage

    prev_voltage_setpoint = 0;
    voltage_setpoint = 0;
    velocity_setpoint = 0;

    internal_motor_pid.kP = pid_consts.kP;
    internal_motor_pid.kI = pid_consts.kI;
    internal_motor_pid.kD = pid_consts.kD;
    internal_motor_pid.I_max = pid_consts.I_max;
    integral = 0;
    prev_error = 0;

    lock.exchange(false);
}


Motor::~Motor( )
{
    delete motor;
}



int Motor::to_voltage(int velocity) {
    pros::motor_gearset_e_t gearset = motor->get_gearing();

    int prev_max;
    int prev_min;

    if ( gearset == pros::E_MOTOR_GEARSET_36 )  //100 RPM Motor
    {
        prev_max = 120;
        prev_min = -120;
    }
    if ( gearset == pros::E_MOTOR_GEARSET_06 ) //600 RPM Motor
    {
        prev_max = 720;
        prev_min = -720;
    }
    else //default to 200 RPM motor because that is most commonly used
    {
        prev_max = 240;
        prev_min = -240;
    }

    int new_max = 12000;
    int new_min = -12000;

    int voltage = (((velocity - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;

    return voltage;
}


int Motor::to_velocity(int voltage) {
    int prev_max = 12000;
    int prev_min = -12000;

    pros::motor_gearset_e_t gearset = motor->get_gearing();

    int new_max;
    int new_min;

    if ( gearset == pros::E_MOTOR_GEARSET_36 )  //100 RPM Motor
    {
        new_max = 120;
        new_min = -120;
    }
    if ( gearset == pros::E_MOTOR_GEARSET_06 ) //600 RPM Motor
    {
        new_max = 720;
        new_min = -720;
    }
    else //default to 200 RPM motor because that is most commonly used
    {
        new_max = 240;
        new_min = -240;
    }

    int velocity = (((voltage - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;


    return velocity;
}


/**
 * calculates the rate that the motor would be set to with a target the previous
 * voltage, and how much time has passed
 * returns rate in mv/ms
 */
int Motor::calc_target_rate( int target, int previous, int delta_t)
{
    int delta_v = target - previous;
    int rate;
    if ( delta_t == 0 && delta_v == 0 )
    {
        rate = 0;
    }
    else if ( delta_t == 0 && delta_v != 0 )
    {
        rate = INT32_MAX;  //essentially undefined but still represented as integer
    }
    else
    {
        rate = delta_v / delta_t;
    }

    return rate;
}


/**
 * returns the target voltage set to the motor after performing PID and slew rate
 * calculations on it
 */
int Motor::get_target_voltage( int delta_t )
{
    double kP = internal_motor_pid.kP;
    double kI = internal_motor_pid.kI;
    double kD = internal_motor_pid.kD;
    double I_max = internal_motor_pid.I_max;

    int voltage;
    int calculated_target_voltage = voltage_setpoint;

    //velocity pid is enabled when the target voltage does not change
    if ( mode == e_custom_velocity_pid && voltage_setpoint == prev_voltage_setpoint )
    {
        int error =  to_velocity(voltage_setpoint) - get_actual_velocity();
        if ( std::abs(integral) > I_max )
        {
            integral = 0;
        }
        else
        {
            integral = integral + error;
        }
        double derivative = error - prev_error;
        prev_error = error;


        calculated_target_voltage = (kP * error) + (kI * integral) + (kD * derivative);
    }

    //ensure that voltage range is allowed by the slew rate set
    int rate = calc_target_rate(calculated_target_voltage, get_actual_voltage(), delta_t);
    if ( slew_enabled && std::abs(rate) > slew_rate )
    {
        int max_delta_v = slew_rate * delta_t;
        int polarity = 1;               // rate will be positive or negative if motor is gaining
        if ( rate < 0 )                 // or losing velocity
        {                               // the polarity ensures that the max voltage is added
            polarity = -1;              // in the correct direction so that the motor's velocity
        }                               // will increase in the correct direction

        voltage = get_actual_voltage() + (polarity * max_delta_v);
    }
    else if ( voltage_setpoint == 0 )
    {
        voltage = 0;
    }
    else if ( calculated_target_voltage != 0 )
    {
        voltage = calculated_target_voltage;
    }

    prev_voltage_setpoint = voltage_setpoint;


    return voltage;
}




//accessor functions

/**
 * returns velocity of motor
 */
double Motor::get_actual_velocity( )
{
    return motor->get_actual_velocity();
}


/**
 * returns voltage of motor
 */
double Motor::get_actual_voltage( )
{
    return motor->get_voltage();
}


/**
 * returns current drawn by motor in mA
 */
int Motor::get_current_draw( )
{
    return motor->get_current_draw();
}


/**
 * returns encoder position of motor in degrees
 */
double Motor::get_encoder_position( )
{
    return motor->get_position();
}


/**
 * returns gearset of motor
 */
pros::motor_gearset_e_t Motor::get_gearset( )
{
    return motor->get_gearing();
}


/**
 * returns brake mode of motor
 */
pros::motor_brake_mode_e_t Motor::get_brake_mode( )
{
    return motor->get_brake_mode();
}


/**
 * returns port of motor
 */
int Motor::get_port( )
{
    return motor_port;
}


/**
 * returns pid constants used by motor
 */
pid Motor::get_pid( )
{
    return internal_motor_pid;
}


/**
 * returns slew rate used by motor
 */
int Motor::get_slew_rate( )
{
    return slew_rate;
}


/**
 * returns power of motor in watts
 */
double Motor::get_power( )
{
    return motor->get_power();
}


```

```
/**
 * returns temperature of motor in degrees C
 */
double Motor::get_temperature( )
{
    return motor->get_temperature();
}


/**
 * returns torque of motor in Nm
 */
double Motor::get_torque( )
{
    return motor->get_torque();
}


/**
 * returns direction motor is spinning
 */
int Motor::get_direction( )
{
    return motor->get_direction();
}


/**
 * returns efficiency of motor as a percent
 */
int Motor::get_efficiency( )
{
    return motor->get_efficiency();
}


/**
 * returns true if motor is at reast
 */
int Motor::is_stopped( )
{
    return motor->is_stopped();
}


/**
 * returns true if the motor has been reversed internally
 */
int Motor::is_reversed( )
{
    return motor->is_reversed();
}





//setter functions

/**
 * aquires lock and creates a new motor on a different port
 * exception safe to always release lock
 */
int Motor::set_port( int port )
{
    pros::motor_gearset_e_t gearset = motor->get_gearing();
    bool reversed = motor->is_reversed();

    while ( lock.exchange( true ) );

    try
    {
        delete motor;
        motor = new pros::Motor(port, gearset, reversed, pros::E_MOTOR_ENCODER_DEGREES);
        motor_port = port;
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set port on motor port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);
    return 1;
}


/**
 * aquires lock and sets zero position of motor
 * exception safe to always release lock
 */
int Motor::tare_encoder( )
{
    while ( lock.exchange( true ) );

    try
    {
        motor->tare_position();
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not tare encoder on motor port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);

    return 1;
}


/**
 * aquires lock and sets new brake mode for motor
 * exception safe to always release lock
 */
int Motor::set_brake_mode( pros::motor_brake_mode_e_t brake_mode )
{
    while ( lock.exchange( true ) );

    try
    {
        motor->set_brake_mode(brake_mode);
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set brakemode on motor port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);

    return 1;
}


/**
 * aquires lock and sets new gearing for motor
 * exception safe to always release lock
 */
int Motor::set_gearing( pros::motor_gearset_e_t gearset )
{
    while ( lock.exchange( true ) );

    try
    {
        motor->set_gearing(gearset);
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set gearing on motor port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);

    return 1;
}


/**
 * aquires lock and internally reverses motor
 * exception safe to always release lock
 */
int Motor::reverse_motor( )
{
    while ( lock.exchange( true ) );

    try
    {
        motor->set_reversed(!motor->is_reversed());
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not reverse motor on port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);

    return 1;
}


/**
 * aquires lock and sets new PID constants for the motor
 * exception safe to always release lock
 */
int Motor::set_pid( pid pid_consts )
{
    while ( lock.exchange( true ) );

    try
    {
        internal_motor_pid.kP = pid_consts.kP;
        internal_motor_pid.kI = pid_consts.kI;
        internal_motor_pid.kD = pid_consts.kD;
```

```cpp
        internal_motor_pid.I_max = pid_consts.I_max;
    }
    catch(...) //ensure lock will be released
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not set motor pid on motor port " + std::to_string(motor_port);
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);

    return 1;
}


/**
 * sets a new log level for the motor, caps it between 0 and 5
 */
void Motor::set_log_level( int logging )
{
    if ( logging > 5 )
    {
        log_level = 5;
    }
    else if ( logging < 0 )
    {
        log_level = 0;
    }
    else
    {
        log_level = logging;
    }
}




//movement functions

/**
 * sets new voltage by scaling from interval +/- 127 to +/- 12000
 */
int Motor::move( int voltage )
{
    int prev_max = 127;
    int prev_min = -127;
    int new_max = 12000;
    int new_min = -12000;

    int scaled_voltage = (((voltage - prev_min) * (new_max - new_min)) / (prev_max - prev_min)) + new_min;
    set_voltage_setpoint(scaled_voltage); //dont aquire lock because it will be acquired in this function
    set_velocity_setpoint(to_velocity(scaled_voltage));

    return 1;
}

int Motor::user_move( int voltage ) {
    if(allow_driver_control) {
        move(voltage);
        return 1;
    }

    return 0;
}


/**
 * sets new voltage by scaling from gearset interval to voltage range
 * of +/- 12000
 */
int Motor::move_velocity( int velocity )
{
    set_velocity_setpoint(velocity);
    set_voltage_setpoint(to_voltage(velocity));

    return 1;
}


int Motor::set_voltage(int voltage) {
    set_voltage_setpoint(voltage);
    set_velocity_setpoint(to_velocity(voltage));

    return 1;
}


/**
 * aquires lock and sets new voltage setpoint for the motor
 * exception safe to always release lock
 */
int Motor::set_voltage_setpoint( int voltage )
{
    while ( lock.exchange( true ) );
    voltage_setpoint = voltage;
    if ( voltage_setpoint != prev_voltage_setpoint )  //reset integral for new setpoint
    {
        integral = 0;
    }
    lock.exchange(false);

    return 1;
}


int Motor::set_velocity_setpoint(int new_velocity) {
    while ( lock.exchange( true ) );
    velocity_setpoint = new_velocity;
    lock.exchange(false);

    return 1;
}




//velocity pid control functions

/**
 * aquires lock and sets flag for using velocity PID
 */
void Motor::set_motor_mode(motor_mode new_mode)
{
    while ( lock.exchange( true ) );
    mode = new_mode;
    lock.exchange(false);
}



//slew control functions

/**
 * aquires lock and sets new slew rate to be used in calculations
 */
int Motor::set_slew( int rate )
{
    while ( lock.exchange( true ) );
    slew_rate = rate;
    lock.exchange(false);

    return 1;
}


/**
 * aquires lock and sets flag for using slew rate
 */
void Motor::enable_slew( )
{
    while ( lock.exchange( true ) );
    slew_enabled = true;
    lock.exchange(false);
}


/**
 * aquires lock and clears flag for using slew rate
 */
void Motor::disable_slew( )
{
    while ( lock.exchange( true ) );
    slew_enabled = false;
    lock.exchange(false);
}




//driver control lock setting and clearing functions

/**
 * aquires lock and sets flag for allowing driver control
 */
void Motor::enable_driver_control()
{
    while ( lock.exchange( true ) );
    allow_driver_control = true;
    lock.exchange(false);
}


/**
 * aquires lock and clears flag for allowing driver control
 */
void Motor::disable_driver_control()
{
    while ( lock.exchange( true ) );
    allow_driver_control = false;
    lock.exchange(false);
}


/**
 * returns flag for allowing driver control
 */
int Motor::driver_control_allowed()
{
    if ( allow_driver_control )
    {
        return 1;
    }
    else
    {
        return 0;
    }
}




/**
 * gets the voltage to set the motor to based on pid and slew rate calculations
 * and internally sets motor voltage
 * calculates what log message is to be based on the log level set and adds it to
 * the logger queue
 */
int Motor::run( int delta_t )
{
    switch(mode) {
        case e_builtin_velocity_pid: {
```

```
        motor->move_velocity(velocity_setpoint);
        break;
    } case e_voltage: {
        motor->move_voltage(voltage_setpoint);
        break;
    } case e_custom_velocity_pid: {
        int voltage = get_target_voltage( delta_t );
        motor->move_voltage(voltage);
        break;
    }
}



std::string log_msg;
switch ( log_level )
{
    case 0:
        log_msg = "";
        break;

    case 1:
        log_msg = (
            "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
            + ", Actual_Vol: " + std::to_string(get_actual_voltage())
            + ", Brake: " + std::to_string(get_brake_mode())
            + ", Gear: " + std::to_string(get_gearset())
            + ", I_max: " + std::to_string(internal_motor_pid.I_max)
            + ", I: " + std::to_string(integral)
            + ", kD: " + std::to_string(internal_motor_pid.kD)
            + ", kI: " + std::to_string(internal_motor_pid.kI)
            + ", kP: " + std::to_string(internal_motor_pid.kP)
            + ", Slew: " + std::to_string(get_slew_rate())
            + ", Time: " + std::to_string(pros::millis())
            + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
            + ", Vel: " + std::to_string(get_actual_velocity())
        );
        break;

    case 2:
        log_msg = (
            "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
            + ", Actual_Vol: " + std::to_string(get_actual_voltage())
            + ", Brake: " + std::to_string(get_brake_mode())
            // + ", Calc_Target_Vol: " + std::to_string(voltage)
            + ", Gear: " + std::to_string(get_gearset())
            + ", I_max: " + std::to_string(internal_motor_pid.I_max)
            + ", I: " + std::to_string(integral)
            + ", kD: " + std::to_string(internal_motor_pid.kD)
            + ", kI: " + std::to_string(internal_motor_pid.kI)
            + ", kP: " + std::to_string(internal_motor_pid.kP)
            + ", Slew: " + std::to_string(get_slew_rate())
            + ", Target_Vol: " + std::to_string(voltage_setpoint)
            + ", Time: " + std::to_string(pros::millis())
            + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
            + ", Vel: " + std::to_string(get_actual_velocity())
        );
        break;

    case 3:
        log_msg = (
            "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
            + ", Actual_Vol: " + std::to_string(get_actual_voltage())
            + ", Brake: " + std::to_string(get_brake_mode())
            // + ", Calc_Target_Vol: " + std::to_string(voltage)
            + ", Gear: " + std::to_string(get_gearset())
            + ", I_max: " + std::to_string(internal_motor_pid.I_max)
            + ", I: " + std::to_string(integral)
            + ", IME: " + std::to_string(get_encoder_position())
            + ", kD: " + std::to_string(internal_motor_pid.kD)
            + ", kI: " + std::to_string(internal_motor_pid.kI)
            + ", kP: " + std::to_string(internal_motor_pid.kP)
            + ", Slew: " + std::to_string(get_slew_rate())
            + ", Target_Vol: " + std::to_string(voltage_setpoint)
            + ", Time: " + std::to_string(pros::millis())
            + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
            + ", Vel: " + std::to_string(get_actual_velocity())
        );
        break;

    case 4:
        log_msg = (
            "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
            + ", Actual_Vol: " + std::to_string(get_actual_voltage())
            + ", Brake: " + std::to_string(get_brake_mode())
            // + ", Calc_Target_Vol: " + std::to_string(voltage)
            + ", Dir: " + std::to_string(get_direction())
            + ", Gear: " + std::to_string(get_gearset())
            + ", I_max: " + std::to_string(internal_motor_pid.I_max)
            + ", I: " + std::to_string(integral)
            + ", IME: " + std::to_string(get_encoder_position())
            + ", kD: " + std::to_string(internal_motor_pid.kD)
            + ", kI: " + std::to_string(internal_motor_pid.kI)
            + ", kP: " + std::to_string(internal_motor_pid.kP)
            + ", Reversed: " + std::to_string(is_reversed())
            + ", Slew: " + std::to_string(get_slew_rate())
            + ", Target_Vol: " + std::to_string(voltage_setpoint)
            + ", Time: " + std::to_string(pros::millis())
            + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
            + ", Vel: " + std::to_string(get_actual_velocity())
        );
        break;

    case 5:
        log_msg = (
            "[INFO]," + std::string(" Motor ") + std::to_string(motor_port)
            + ", Actual_Vol: " + std::to_string(get_actual_voltage())
            + ", Brake: " + std::to_string(get_brake_mode())
            // + ", Calc_Target_Vol: " + std::to_string(voltage)
            + ", Current: " + std::to_string(get_current_draw())
            + ", Dir: " + std::to_string(get_direction())
            + ", Gear: " + std::to_string(get_gearset())
            + ", I_max: " + std::to_string(internal_motor_pid.I_max)
```

```
                + ", I: " + std::to_string(integral)
                + ", IME: " + std::to_string(get_encoder_position())
                + ", kD: " + std::to_string(internal_motor_pid.kD)
                + ", kI: " + std::to_string(internal_motor_pid.kI)
                + ", kP: " + std::to_string(internal_motor_pid.kP)
                + ", Reversed: " + std::to_string(is_reversed())
                + ", Slew: " + std::to_string(get_slew_rate())
                + ", Target_Vol: " + std::to_string(voltage_setpoint)
                + ", Temp: " + std::to_string(get_temperature())
                + ", Time: " + std::to_string(pros::millis())
                + ", Torque: " + std::to_string(get_torque())
                + ", Vel_Sp: " + std::to_string(to_velocity(voltage_setpoint))
                + ", Vel: " + std::to_string(get_actual_velocity())
            );
            break;

    }

    Logger logger;
    log_entry entry;
    entry.content = log_msg;
    entry.stream = "clog";
    logger.add(entry);

    return 1;
}
```

```
/**
 * @file: ./RobotCode/src/objects/motors/MotorThread.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 * TODO:
 *
 * contains functions that handle motor functions
 */

#ifndef __MOTORTHREAD_HPP__
#define __MOTORTHREAD_HPP__

#include <vector>
#include <atomic>

#include "main.h"

#include "../../Configuration.hpp"
#include "Motor.hpp"


/**
 * @see: Motor.hpp
 *
 * contains singleton class for using motors in a thread
 * motors are added to a vector and iterated over in a thread so that the voltage
 * can be set
 */
class MotorThread
{
    private:
        MotorThread();
        static MotorThread *thread_obj;

        static std::vector<Motor*> motors;
        static std::atomic<bool> lock;  //protect vector from concurrent access


        /**
         * @param: void* -> not used, but necessary to follow thread making constructor
         * @return: None
         *
         * the function to be run on a thread that calls the run function for
         * each motor that sets the voltage and performs logging
         */
        static void run(void*);

        pros::Task *thread;  // the motor thread


    public:
        ~MotorThread();

        /**
         * @return: MotorThread -> instance of class to be used throughout program
         *
         * give the instance of the singleton class or creates it if it does
         * not yet exist
         */
        static MotorThread* get_instance();




        /**
         * @return: None
         *
         * starts the thread or resmes it if it was stopped
         */
        void start_thread();

        /**
         * @return: None
         *
         * stops the thread from being scheduled
         */
        void stop_thread();




        /**
         * @param: Motor &motor -> the motor to add to the vector
         * @return: int -> 1 if motor was successfully added, 0 otherwise
         *
         * adds a motor to the vector of motors to operate
         * logs that the motor was added to the logger queue
         */
        int register_motor( Motor &motor );

        /**
         * @param: Motor &motor -> the motor to remove from the vector
         * @return: int -> 1 if motor was successfully added, 0 otherwise
         *
         * removes a motor from the vector of motors to operate
         * logs that the motor was removed to the logger queue
         */
        int unregister_motor( Motor &motor );

        int is_registered(Motor &motor);


};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/motors/MotorThread.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for functions that handle motor functions
 */

#include <atomic>
#include <stdio.h>
#include <vector>

#include "main.h"

#include "../serial/Logger.hpp"
#include "Motor.hpp"
#include "MotorThread.hpp"


MotorThread *MotorThread::thread_obj = NULL;
std::vector<Motor*> MotorThread::motors;
std::atomic<bool> MotorThread::lock = ATOMIC_VAR_INIT(false);


MotorThread::MotorThread()
{
    thread = new pros::Task( run, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "motor_thread");
    thread->suspend();
}


MotorThread::~MotorThread()
{
    thread->remove();
    delete thread;
}


void MotorThread::run(void*)
{
    int start = pros::millis();
    while (1) {
        while ( lock.exchange( true ) );
        for ( int i = 0; i < motors.size(); i++ ) {
            motors.at(i)->run( pros::millis() - start );
        }
        start = pros::millis();
        lock.exchange(false);
        pros::delay(5);
    }
}



/**
 * inits object if object is not already initialized based on a static bool
 * sets bool if it is not set
 */
MotorThread* MotorThread::get_instance() {
    if ( thread_obj == NULL ) {
        thread_obj = new MotorThread;
    }
    return thread_obj;
}


void MotorThread::start_thread() {
    thread->resume();
}

void MotorThread::stop_thread() {
    thread->suspend();
}


int MotorThread::register_motor( Motor &motor ) {
    while ( lock.exchange( true ) );

    Logger logger;
    log_entry entry;
    char buffer[10];


    try
    {
        motors.push_back(&motor);

        sprintf(buffer, "%p", &motor);
        entry.stream = "clog";
        entry.content = "[INFO], " + std::to_string(pros::millis()) +  ", motor added at " + buffer;
        logger.add(entry);
    }
    catch ( ... )
    {
        sprintf(buffer, "%p", &motor);
        entry.content = "[WARNING], " + std::to_string(pros::millis()) +  ", could not add motor at " + buffer;
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);
    return 1;
}


int MotorThread::unregister_motor( Motor &motor )
{
    while ( lock.exchange( true ) );

    Logger logger;
    log_entry entry;
    char buffer[10];

    auto element = std::find(begin(motors), end(motors), &motor);
    if ( element != motors.end())
    {
        motors.erase(element);

        sprintf(buffer, "%p", &motor);
        entry.stream = "clog";
        entry.content = "[INFO] " + std::to_string(pros::millis()) + ", motor removed at " + buffer;
        logger.add(entry);
    }
    else
    {
        sprintf(buffer, "%p", &motor);
        entry.content = "[WARNING] " + std::to_string(pros::millis()) +  ", could not remove motor at " + buffer;
        entry.stream = "cerr";
        logger.add(entry);

        lock.exchange(false);
        return 0;
    }

    lock.exchange(false);
    return 1;

}


int MotorThread::is_registered(Motor &motor) {
    int registered = 0;

    while ( lock.exchange( true ) );

    auto element = std::find(begin(motors), end(motors), &motor);
    if ( element != motors.end()) {
        registered = 1;
    }

    lock.exchange(false);

    return registered;
}
```

../RobotCode/src/objects/motors/Motors.hpp

```
/**
 * @file: ./RobotCode/src/motors/Motors.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * contains global struct for all motors
 */

#ifndef __MOTORS_HPP__
#define __MOTORS_HPP__

#include <array>

#include "main.h"

#include "../../Configuration.hpp"
#include "Motor.hpp"


namespace Motors
{
    extern Motor front_right;
    extern Motor front_left;
    extern Motor back_right;
    extern Motor back_left;
    extern Motor left_intake;
    extern Motor right_intake;
    extern Motor upper_indexer;
    extern Motor lower_indexer;

    extern double chassis_gear_ratio;

    extern std::array<Motor*, 8> motor_array;
    extern std::array<std::string, 8> motor_names_array;

    void enable_driver_control();
    void disable_driver_control();
    void set_brake_mode(pros::motor_brake_mode_e_t new_brakemode);
    void stop_all_motors();
    void set_log_level(int log_level);
    void register_motors();
    void unregister_motors();
};


#endif
```

```cpp
/**
 * @file: ./RobotCode/src/motors/Motors.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * contains definition of global struct
 */

#include "Motors.hpp"
#include "MotorThread.hpp"

namespace Motors
{
    Motor front_right {Configuration::get_instance()->front_right_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->front_right_reversed};
    Motor front_left {Configuration::get_instance()->front_left_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->front_left_reversed};
    Motor back_right {Configuration::get_instance()->back_right_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->back_right_reversed};
    Motor back_left {Configuration::get_instance()->back_left_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->back_left_reversed};
    Motor left_intake {Configuration::get_instance()->left_intake_port, pros::E_MOTOR_GEARSET_36, Configuration::get_instance()->left_intake_reversed};
    Motor right_intake {Configuration::get_instance()->right_intake_port, pros::E_MOTOR_GEARSET_36, Configuration::get_instance()->right_intake_reversed};
    Motor upper_indexer {Configuration::get_instance()->upper_indexer_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->upper_indexer_reversed};
    Motor lower_indexer {Configuration::get_instance()->lower_indexer_port, pros::E_MOTOR_GEARSET_06, Configuration::get_instance()->lower_indexer_reversed};

    double chassis_gear_ratio = 3 / 5;

    std::array<Motor*, 8> motor_array = {
        &front_right,
        &front_left,
        &back_right,
        &back_left,
        &left_intake,
        &right_intake,
        &upper_indexer,
        &lower_indexer,
    };

    std::array<std::string, 8> motor_names_array = {
        "Front Left",
        "Front Right",
        "Back Left",
        "Back Right",
        "Left Intake",
        "Right Intake",
        "Upper Indexer",
        "Lower Indexer"
    };

    void enable_driver_control() {
        Motors::front_left.enable_driver_control();
        Motors::front_left.enable_driver_control();
        Motors::back_right.enable_driver_control();
        Motors::back_left.enable_driver_control();
        Motors::left_intake.enable_driver_control();
        Motors::right_intake.enable_driver_control();
        Motors::upper_indexer.enable_driver_control();
        Motors::lower_indexer.enable_driver_control();
    }

    void disable_driver_control() {
        Motors::front_left.disable_driver_control();
        Motors::front_left.disable_driver_control();
        Motors::back_right.disable_driver_control();
        Motors::back_left.disable_driver_control();
        Motors::left_intake.disable_driver_control();
        Motors::right_intake.disable_driver_control();
        Motors::upper_indexer.disable_driver_control();
        Motors::lower_indexer.disable_driver_control();
    }

    void set_brake_mode(pros::motor_brake_mode_e_t new_brakemode) {
        Motors::front_left.set_brake_mode(new_brakemode);
        Motors::front_left.set_brake_mode(new_brakemode);
        Motors::back_right.set_brake_mode(new_brakemode);
        Motors::back_left.set_brake_mode(new_brakemode);
        Motors::left_intake.set_brake_mode(new_brakemode);
        Motors::right_intake.set_brake_mode(new_brakemode);
        Motors::upper_indexer.set_brake_mode(new_brakemode);
        Motors::lower_indexer.set_brake_mode(new_brakemode);
    }

    void stop_all_motors() {
        Motors::front_left.move(0);
        Motors::front_left.move(0);
        Motors::back_right.move(0);
        Motors::back_left.move(0);
        Motors::left_intake.move(0);
        Motors::right_intake.move(0);
        Motors::upper_indexer.move(0);
        Motors::lower_indexer.move(0);
    }

    void set_log_level(int log_level) {
        Motors::front_right.set_log_level(log_level);
        Motors::front_left.set_log_level(log_level);
        Motors::back_right.set_log_level(log_level);
        Motors::back_left.set_log_level(log_level);
        Motors::left_intake.set_log_level(log_level);
        Motors::right_intake.set_log_level(log_level);
        Motors::upper_indexer.set_log_level(log_level);
        Motors::lower_indexer.set_log_level(log_level);
    }

    void register_motors() {
        MotorThread* motor_thread = MotorThread::get_instance();
        motor_thread->register_motor(Motors::front_right);
        motor_thread->register_motor(Motors::front_left);
        motor_thread->register_motor(Motors::back_right);
        motor_thread->register_motor(Motors::back_left);
        motor_thread->register_motor(Motors::left_intake);
        motor_thread->register_motor(Motors::right_intake);
        motor_thread->register_motor(Motors::upper_indexer);
        motor_thread->register_motor(Motors::lower_indexer);
    }

    void unregister_motors() {
        MotorThread* motor_thread = MotorThread::get_instance();
        motor_thread->unregister_motor(Motors::front_right);
        motor_thread->unregister_motor(Motors::front_left);
        motor_thread->unregister_motor(Motors::back_right);
        motor_thread->unregister_motor(Motors::back_left);
        motor_thread->unregister_motor(Motors::left_intake);
        motor_thread->unregister_motor(Motors::right_intake);
        motor_thread->unregister_motor(Motors::upper_indexer);
        motor_thread->unregister_motor(Motors::lower_indexer);
    }

};
```

../RobotCode/src/objects/sensors/AnalogInSensor.hpp

```
/**
 * @file: ./RobotCode/src/objects/sensors/AnalogInSensor.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains a wrapper class for ADI analog in sensor
 */

#ifndef __ANALOGINSENSOR_HPP__
#define __ANALOGINSENSOR_HPP__

#include <atomic>
#include <vector>

#include "main.h"


class AnalogInSensor
{
    private:
        pros::ADIAnalogIn *sensor;
        bool calibrated;

    public:
        AnalogInSensor();
        AnalogInSensor(char port);
        AnalogInSensor(pros::ext_adi_port_pair_t port_pair);
        ~AnalogInSensor();

        void set_port(char port);
        void set_port(pros::ext_adi_port_pair_t port_pair);

        double get_raw_value();
        double get_value(bool high_res);

        void calibrate();
        bool is_calibrated();
};





#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/sensors/AnalogInSensor.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for wrapper class for analog in sensor
 */

#include "../serial/Logger.hpp"
#include "AnalogInSensor.hpp"



AnalogInSensor::AnalogInSensor() {
    sensor = NULL;
}

AnalogInSensor::AnalogInSensor(char port) {
    sensor = new pros::ADIAnalogIn(port);
}

AnalogInSensor::AnalogInSensor(pros::ext_adi_port_pair_t port_pair) {
    sensor = new pros::ADIAnalogIn(port_pair);
}

AnalogInSensor::~AnalogInSensor()
{
    if(sensor != NULL) {
        delete sensor;
    }
}


void AnalogInSensor::set_port(char port) {
    if(sensor != NULL) {
        delete sensor;
    }

    sensor = new pros::ADIAnalogIn(port);
}


void AnalogInSensor::set_port(pros::ext_adi_port_pair_t port_pair) {
    if(sensor != NULL) {
        delete sensor;
    }

    sensor = new pros::ADIAnalogIn(port_pair);
}


double AnalogInSensor::get_raw_value() {
    double value = sensor->get_value();
    return value;

}


double AnalogInSensor::get_value(bool high_res) {
    if(!calibrated)
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not read analog sensor (not calibrated) ";
        entry.stream = "cerr";

        logger.add(entry);

        return INT32_MAX;
    }

    if(high_res)
    {
        return sensor->get_value_calibrated_HR();
    }
    else
    {
        return sensor->get_value_calibrated();
    }
}


void AnalogInSensor::calibrate() {
    sensor->calibrate();
    calibrated = true;

}


bool AnalogInSensor::is_calibrated() {
    return calibrated;
}
```

```
/**
 * @file: ./RobotCode/src/objects/sensors/BallDetector.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains a wrapper class for the encoders
 */

#ifndef __BALLDETECTOR_HPP__
#define __BALLDETECTOR_HPP__

#include <tuple>

#include "main.h"

#include "AnalogInSensor.hpp"


class BallDetector
{
    private:
        AnalogInSensor ball_detector_top;
        AnalogInSensor ball_detector_filter;
        AnalogInSensor ball_detector_bottom;
        pros::Optical* optical_sensor;

        int time_since_last_ball;
        bool log_data;

        int threshold;

    public:
        BallDetector(
            AnalogInSensor& detector_top_left,
            AnalogInSensor& detector_filter,
            AnalogInSensor& detector_bottom,
            int optical_port,
            int detector_threshold
        );
        ~BallDetector();

        int set_threshold(int new_threshold);
        int check_filter_level();
        std::vector<bool> locate_balls();

        void set_led_brightness(int pct);
        void auto_set_led_brightness();

        std::tuple<int, int> debug_color();
        void start_logging();
        void stop_logging();

};





#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/sensors/BallDetector.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for ball detector class
 */

#include "../serial/Logger.hpp"
#include "BallDetector.hpp"



BallDetector::BallDetector(
    AnalogInSensor& detector_top_left,
    AnalogInSensor& detector_filter,
    AnalogInSensor& detector_bottom,
    int optical_port,
    int detector_threshold
) {
    ball_detector_top = detector_top_left;
    ball_detector_filter = detector_filter;
    ball_detector_bottom = detector_bottom;
    optical_sensor = new pros::Optical(optical_port);

    optical_sensor->disable_gesture();
    optical_sensor->set_led_pwm(50);

    threshold = detector_threshold;
    time_since_last_ball = 0;
    log_data = false;
}

BallDetector::~BallDetector() {
    delete optical_sensor;
}


int BallDetector::set_threshold(int new_threshold) {
    threshold = new_threshold;
    return 1;
}


int BallDetector::check_filter_level() {
    int return_code = 0;
    if(ball_detector_filter.get_raw_value() < threshold) {  // ball is detected
        time_since_last_ball = 0;  // ball detected so there is no time since last ball

        double hue = optical_sensor->get_hue();
        if(hue > 170 && hue < 260) {  // color is blue
            return_code = 1;
        } else if(hue > 335 || hue < 25) {  // color is red
            return_code = 2;
        } else {  // could not determine color based on ranges
            return_code = -1;
        }
    } else {
        time_since_last_ball = pros::millis() - time_since_last_ball;  // get time elapsed
    }

    if(log_data) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("BALL_DETECT_MIDDLE")
            + ", Time: " + std::to_string(pros::millis())
            + ", ball_detected: " + std::to_string(return_code)
            + ", time_since_last_ball " + std::to_string(time_since_last_ball)
            + ", line_detector: " + std::to_string(ball_detector_filter.get_raw_value())
            + ", threshold: " + std::to_string(threshold)

        );
        entry.stream = "clog";
        logger.add(entry);
    }


    return return_code;  // no ball is detected
}


std::vector<bool> BallDetector::locate_balls() {
    std::vector<bool> locations;
    if(ball_detector_top.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(ball_detector_filter.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(ball_detector_bottom.get_raw_value() < threshold) {
        locations.push_back(true);
    } else {
        locations.push_back(false);
    }

    if(log_data) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("BALL_DETECT_MIDDLE")
            + ", time: " + std::to_string(pros::millis())
            + ", top_present: " + std::to_string(locations.at(0))
            + ", middle_present: " + std::to_string(locations.at(1))
            + ", bottom_present: " + std::to_string(locations.at(2))
            + ", top: " + std::to_string(ball_detector_top.get_raw_value())
            + ", middle: " + std::to_string(ball_detector_filter.get_raw_value())
            + ", bottom: " + std::to_string(ball_detector_bottom.get_raw_value())
            + ", threshold: " + std::to_string(threshold)
        );
        entry.stream = "clog";
        logger.add(entry);
    }


    return locations;
}


void BallDetector::set_led_brightness(int pct) {
    optical_sensor->set_led_pwm(pct);
}


void BallDetector::auto_set_led_brightness() {
    int pct = 100 * std::abs(1 - optical_sensor->get_brightness());  // 1 to 1 scale
    optical_sensor->set_led_pwm(pct);
}


std::tuple<int, int> BallDetector::debug_color() {
    return std::make_tuple(ball_detector_filter.get_raw_value(), check_filter_level());
}

void BallDetector::start_logging() {
    log_data = true;
}

void BallDetector::stop_logging() {
    log_data = false;
}
```

```
/**
 * @file: ./RobotCode/src/objects/sensors/Encoder.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains a wrapper class for the encoders
 */

#ifndef __ENCODER_HPP__
#define __ENCODER_HPP__

#include <atomic>
#include <unordered_map>

#include "main.h"


class Encoder
{
    private:
        pros::ADIEncoder *encoder;

        std::atomic<bool> lock;  // protect map from concurrent access
        int latest_uid;
        std::unordered_map<int, double> zero_positions;

    public:
        Encoder(char upper_port, char lower_port, bool reverse);
        ~Encoder();

        int get_unique_id(bool zero=false);

        double get_position(int unique_id);
        double get_absolute_position(bool scaled);

        int reset(int unique_id);

        void forget_position(int unique_id);

};





#endif
```

```
/**
 * @file: ./RobotCode/src/objects/sensors/Encoder.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for wrapper class for Encoder
 */

#include <atomic>
#include <vector>

#include "main.h"

#include "../serial/Logger.hpp"
#include "Encoder.hpp"



Encoder::Encoder( char upper_port, char lower_port, bool reverse )
{
    lock = ATOMIC_VAR_INIT(false);

    encoder = new pros::ADIEncoder(upper_port, lower_port, reverse);


    while ( lock.exchange( true ) ); //aquire lock
    latest_uid = 0;
    zero_positions[0] = encoder->get_value();
    lock.exchange(false);  //release lock
}


Encoder::~Encoder()
{
    // TODO: figure out why checking for null pointer needs to be present to not crash when program starts
    if(encoder != NULL)  // causes segfault when program begins if this is not present
    {
        delete encoder;
    }
}




int Encoder::get_unique_id(bool zero /*false*/)
{
    while ( lock.exchange( true ) ); //aquire lock

    latest_uid += 1;
    int id = latest_uid;
    zero_positions[id] = zero_positions.at(0);
    lock.exchange(false);  //release lock

    if(zero) {
        reset(id);
    }

    return id;
}




double Encoder::get_position(int unique_id)
{
    if(zero_positions.find(unique_id) == zero_positions.end())
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not get encoder position with unique id " + std::to_string(unique_id);
        entry.stream = "cerr";

        logger.add(entry);

        return INT32_MAX;
    }

    double position = get_absolute_position(false) - zero_positions.at(unique_id);
    return position;

}



double Encoder::get_absolute_position(bool scaled)
{
    double position = encoder->get_value() - zero_positions.at(0);

    if(scaled)
    {
        position = ((int)position % 720) - 360;  // scales to interval [-360,360]
    }

    return position;
}




int Encoder::reset(int unique_id)
{
    if(zero_positions.find(unique_id) == zero_positions.end() || unique_id == 0)
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not get encoder position with unique id " + std::to_string(unique_id);
        entry.stream = "cerr";

        logger.add(entry);

        return 0;
    }

    zero_positions.at(unique_id) = encoder->get_value();
    return 1;
}


void Encoder::forget_position(int unique_id) {
    if(zero_positions.find(unique_id) == zero_positions.end() || unique_id == 0)
    {
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", could not remove zero position with unique id " + std::to_string(unique_id);
        entry.stream = "cerr";

        logger.add(entry);

    }
    zero_positions.erase(unique_id);
}
```

```
/**
 * @file: ./RobotCode/src/objects/sensors/Sensors.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/29/2020
 * @reviewed_by: Aiden Carney
 *
 * contains a class for interacting with the ADI sensors on the robot
 */

#ifndef __SENSORS_HPP__
#define __SENSORS_HPP__

#include "main.h"

#include "BallDetector.hpp"
#include "Encoder.hpp"
#include "AnalogInSensor.hpp"



namespace Sensors
{
    extern Encoder right_encoder;
    extern Encoder left_encoder;
    extern Encoder strafe_encoder;

    extern AnalogInSensor line_tracker_top;
    extern AnalogInSensor line_tracker_middle;
    extern AnalogInSensor line_tracker_bottom;
    extern pros::Optical optical;
    extern BallDetector ball_detector;

    extern pros::Imu imu;
    extern bool imu_is_calibrated;

    void calibrate_imu();
    void log_data();
    std::tuple<double, double> get_average_encoders(int l_id, int r_id);
}




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/sensors/Sensors.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/29/2020
 * @reviewed_by: Aiden Carney
 *
 * @see: Sensors.hpp
 *
 * contains definitions for sensors and implementation for sensor class
 */

#include "Sensors.hpp"
#include "../motors/Motors.hpp"
#include "../../Configuration.hpp"
#include "../serial/Logger.hpp"


namespace Sensors
{
    Encoder right_encoder{RIGHT_ENC_TOP_PORT, RIGHT_ENC_BOTTOM_PORT, false};
    Encoder left_encoder{LEFT_ENC_TOP_PORT, LEFT_ENC_BOTTOM_PORT, false};
    Encoder strafe_encoder{STRAFE_ENC_TOP_PORT, STRAFE_ENC_BOTTOM_PORT, true};

    AnalogInSensor line_tracker_top{DETECTOR_TOP_PORT};
    AnalogInSensor line_tracker_middle{DETECTOR_MIDDLE_PORT};
    AnalogInSensor line_tracker_bottom{DETECTOR_BOTTOM_PORT};

    // BallDetector ball_detector{DETECTOR_TOP_PORT, DETECTOR_MIDDLE_PORT, DETECTOR_BOTTOM_PORT, VISIONSENSOR_PORT, Configuration::get_instance()->filter_threshold};
    BallDetector ball_detector{
        line_tracker_top,
        line_tracker_middle,
        line_tracker_bottom,
        OPTICAL_PORT,
        Configuration::get_instance()->filter_threshold
    };

    pros::Imu imu{IMU_PORT};
    bool imu_is_calibrated = false;

    void calibrate_imu() {
        bool calibrated = false;
        while(!calibrated) {  // block until imu is connected and calibrated
            imu.reset();  // calibrate imu
            while(imu.is_calibrating()) {
                pros::delay(10);
                calibrated = true;
            }
        }
        imu_is_calibrated = true;
    }

    void log_data() {
        Logger logger;
        log_entry entry;
        entry.content = ("[INFO], " + std::to_string(pros::millis())
            + ", Sensor Data"
            +  ", Right_Enc: " + std::to_string(right_encoder.get_absolute_position(false))
            +  ", Left_Enc: " + std::to_string(left_encoder.get_absolute_position(false))
            +  ", Top Detector" + std::to_string(ball_detector.locate_balls().at(0))
            +  ", Middle Detector" + std::to_string(ball_detector.locate_balls().at(1))
            +  ", Bottom Detector" + std::to_string(ball_detector.locate_balls().at(2))
        );
        entry.stream = "clog";
        logger.add(entry);
    }

    /**
     * takes the average of each side of the drive encoders
     * hopefully to reduce error of encoders
     * returns tuple of encoder values
     */
    std::tuple<double, double> get_average_encoders(int l_id, int r_id) {
        // use a weighted average to merge all encoders on the robot for a hopefully more accurate reading
        double left_encoder_val = (0 * Motors::front_left.get_encoder_position() * Motors::chassis_gear_ratio) + (0 * Motors::back_left.get_encoder_position() * Motors::chassis_gear_ratio) + (1 * Sensors::left_encoder.get_position(l_id));
        double right_encoder_val = (0 * Motors::front_right.get_encoder_position() * Motors::chassis_gear_ratio) + (0 * Motors::back_right.get_encoder_position() * Motors::chassis_gear_ratio) + (1 * Sensors::right_encoder.get_position(r_id));

        return {left_encoder_val, right_encoder_val};
    }


}
```

```cpp
/**
 * @file: ./RobotCode/src/objects/lcdCode/Gimmicks.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: fix loading screen, it sometimes does not work
 *
 * contains lcd gimmicks that are used to enhance interface
 *
 */

#ifndef __GIMMICKS_HPP__
#define __GIMMICKS_HPP__

#include <string>

#include "../../../include/main.h"

#include "Styles.hpp"


/**
 * @see: Styles.hpp
 * @see: ./lcdCode
 *
 * used to display warning box
 */
class WarningMessage : virtual Styles
{
    protected:
        /**
         * @param: lv_obj_t* mbox -> message box object
         * @param: const char* txt -> text for message box
         * @return: LV_RES_OK -> if finishes successfully
         *
         * @see: Styles.hpp
         * @see: ./lcdCode
         *
         * sets static int option to positive or negative based on feedback
         *
         */
        static lv_res_t mbox_apply_action(lv_obj_t * mbox, const char * txt);
        static const char* buttons[];
        static int option;

        lv_obj_t *warn_box;


    public:
        WarningMessage();
        virtual ~WarningMessage();

        /**
         * @param: std::string warn_msg -> message that will appear as option
         * @param: lv_obj_t* parent -> the parent that the message box will appear on
         * @return: bool -> if user selected yes or no
         *
         * returns true or false based on what user selects
         * implementation of this is up to user
         *
         */
        bool warn(std::string warn_msg, lv_obj_t *parent);

};


/**
 * @see: Styles.hpp
 * @see: ./lcdCode
 *
 * methods and objects for a loading bar
 */
class Loading : virtual Styles
{
    protected:
        lv_obj_t *loader;

    public:
        Loading();
        ~Loading();

        /**
         * @param: int estimated_duration -> duration that loading should take used to set speed of bar
         * @param: lv_obj_t* parent -> parent object that loading bar will go on
         * @param: int x -> x position of loading bar relative to parent
         * @param int y -> y position of loading bar relative to parent
         * @return: None
         *
         * shows the loader and starts the action of it moving
         *
         */
        void show_load(int estimated_duration, lv_obj_t *parent, int x, int y); //starts the loader

        /**
         * @return: None
         *
         * hides the loader
         * this should be about when the loader is finished
         * Used to keep a smooth transition
         *
         */
        void hide_load(); //ends the loader and hides it
};








#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/lcdCode/Gimmicks.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: Gimmicks.hpp
 *
 * contains implementation for header file
 *
 */

#include "../../../include/main.h"
#include "../../../include/api.h"

#include "Styles.hpp"
#include "Gimmicks.hpp"



const char* WarningMessage::buttons[] = {"Back", "Continue", ""};

//base classes
int WarningMessage::option = 0;

WarningMessage::WarningMessage()
{
    option = 0;

    warn_box = lv_mbox_create(lv_scr_act(), NULL);
    lv_mbox_set_text(warn_box, "None");
    lv_mbox_add_btns(warn_box, buttons, NULL);
    lv_mbox_set_action(warn_box, mbox_apply_action);

    lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BG, &warn_box_bg);
    lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BTN_REL, &warn_box_released);
    lv_mbox_set_style(warn_box, LV_MBOX_STYLE_BTN_PR, &warn_box_pressed);

    lv_obj_set_width(warn_box, 400);
    lv_obj_set_height(warn_box, 140);

    lv_obj_align(warn_box, NULL, LV_ALIGN_CENTER, 0, -50);


}

WarningMessage::~WarningMessage()
{
    lv_obj_del(warn_box);
}

/**
 * compares text of message to set option to positive or negative
 */
lv_res_t WarningMessage::mbox_apply_action(lv_obj_t * mbox, const char * txt)
{
    if ( txt == "Continue" )
    {
        option = 1;
    }

    else if ( txt == "Back" )
    {
        option = -1;
    }

    return LV_RES_OK;
}

/**
 * dislays a message box and sets the text
 * user can choose "continue" or "back"
 * how function works line 3
 */
bool WarningMessage::warn( std::string warn_msg, lv_obj_t *parent )
{
    option = 0;

    lv_obj_set_hidden(warn_box, false);
    lv_obj_set_parent(warn_box, parent);
    lv_mbox_set_text(warn_box, warn_msg.c_str());

    while ( !(option) )
    {
        pros::delay(50);
    }

    if ( option == 1 )
    {
        lv_obj_set_hidden(warn_box, true);
        return true;
    }

    else
    {
        lv_obj_set_hidden(warn_box, true);
        return false;
    }

}






Loading::Loading()
{
    loader = lv_bar_create(lv_scr_act(), NULL);
    lv_obj_set_size(loader, 100, 20);
    lv_bar_set_value(loader, 1);
}

```

```
Loading::~Loading()
{
    lv_obj_del(loader);
}

/**
 * loader is shown on a parent at specified location
 * animation time is set by user, so this function only works if user knows
 * about how long the function will take
 * this function is meant as a filler so that if some initialization occurs
 * the gui does not appear like its hanging for no reason
 */
void Loading::show_load(int estimated_duration, lv_obj_t *parent, int x, int y)
{
    lv_obj_set_hidden(loader, false);
    lv_bar_set_value(loader, 1);
    lv_obj_set_parent(loader, parent);
    lv_obj_set_top(loader, true);

    lv_obj_set_pos(loader, x, y);

    lv_bar_set_value_anim(loader, 100, estimated_duration);
}

/**
 * hides the loader for when the initialization by user is finished
 */
void Loading::hide_load()
{
    lv_obj_set_hidden(loader, true);
}
```

../RobotCode/src/objects/lcdCode/Styles.hpp

```
/**
 * @file: ./RobotCode/src/objects/lcdCode/Styles.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * contains base class for styles of gui objects
 *
 */

#ifndef __STYLES__
#define __STYLES__


#include "../../../include/main.h"

//defines colors to use for each style
#define BLUE_BORDER LV_COLOR_BLUE
#define RED_BORDER LV_COLOR_RED
#define BG LV_COLOR_GRAY
#define BUTTON_REL LV_COLOR_SILVER
#define BUTTON_PR LV_COLOR_NAVY
#define TEXT LV_COLOR_WHITE
#define BODY_TEXT LV_COLOR_BLACK
#define SW_INDIC LV_COLOR_HEX(0x9fc8ef)

//allows use of other fonts
#define USE_DEJAVU_12
#define USE_DEJAVU_16
#include "../../../include/fonts/fonts.h"


/**
 * @see: ../../include/fonts/fonts.hpp
 * @see ../fonts/
 *
 * base class that contains different colors and styles to be used throughout
 * the gui
 * designed so that there is no repetion of styles and so they are all in one place
 * designed to be inherited
 */
class Styles
{
    protected:
        //styles
        lv_style_t blue;
        lv_style_t red;
        lv_style_t gray;

        lv_style_t toggle_btn_released;
        lv_style_t toggle_btn_pressed;

        lv_style_t toggle_tabbtn_released;
        lv_style_t toggle_tabbtn_pressed;

        lv_style_t sw_toggled;
        lv_style_t sw_off;
        lv_style_t sw_bg;
        lv_style_t sw_indic;

        lv_style_t heading_text;
        lv_style_t body_text;
        lv_style_t subheading_text;

        lv_style_t lines;

        lv_style_t warn_box_bg;
        lv_style_t warn_box_pressed;
        lv_style_t warn_box_released;

        lv_style_t loader_style;

    public:
        Styles();
        virtual ~Styles();

};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/lcdCode/Styles.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: Styles.hpp
 *
 * contains base class for styles for gui
 *
 */

#include "../../../include/main.h"
#include "../../../include/api.h"

#include "Styles.hpp"


Styles::Styles()
{
//red style
    lv_style_copy(&red, &lv_style_scr);
    red.body.main_color = LV_COLOR_RED;
    red.body.grad_color = LV_COLOR_RED;
    red.body.border.color = LV_COLOR_RED;

//blue style
    lv_style_copy(&blue, &lv_style_scr);
    blue.body.main_color = LV_COLOR_BLUE;
    blue.body.grad_color = LV_COLOR_BLUE;
    blue.body.border.color = LV_COLOR_BLUE;

//gray style
    lv_style_copy(&gray, &lv_style_scr);
    gray.body.main_color = BG;
    gray.body.grad_color = BG;
    gray.body.border.color = BG;
    gray.body.border.width = 10;

//style for when the button is not pressed
    lv_style_copy(&toggle_btn_released, &lv_style_plain);
    toggle_btn_released.body.main_color = BUTTON_REL;
    toggle_btn_released.body.grad_color = BUTTON_REL;
    toggle_btn_released.body.border.color = BUTTON_REL;
    toggle_btn_released.body.border.width = 2;
    toggle_btn_released.body.border.opa = LV_OPA_0;
    toggle_btn_released.body.radius = 5;
    toggle_btn_released.text.color = TEXT;

//style for when the button is pressed
    lv_style_copy(&toggle_btn_pressed, &lv_style_plain);
    toggle_btn_pressed.body.main_color = BUTTON_PR;
    toggle_btn_pressed.body.grad_color = BUTTON_PR;
    toggle_btn_pressed.body.border.color = BUTTON_REL;
    toggle_btn_pressed.text.color = TEXT;

//style for when tabview button is not pressed
    lv_style_copy(&toggle_tabbtn_released, &lv_style_plain);
    toggle_tabbtn_released.body.main_color = BUTTON_REL;
    toggle_tabbtn_released.body.grad_color = BUTTON_REL;
    toggle_tabbtn_released.body.border.color = BUTTON_REL;
    toggle_tabbtn_released.body.border.width = 2;
    toggle_tabbtn_released.body.border.opa = LV_OPA_0;
    toggle_tabbtn_released.text.color = TEXT;
    toggle_tabbtn_released.text.font = &dejavu_12;

//style for when tabview button is pressed
    lv_style_copy(&toggle_tabbtn_pressed, &lv_style_plain);
    toggle_tabbtn_pressed.body.main_color = BUTTON_PR;
    toggle_tabbtn_pressed.body.grad_color = BUTTON_PR;
    toggle_tabbtn_pressed.body.border.color = BUTTON_REL;
    toggle_tabbtn_pressed.text.color = TEXT;
    toggle_tabbtn_pressed.text.font = &dejavu_12;

//switch on
    lv_style_copy(&sw_toggled, &lv_style_pretty_color);
    sw_toggled.body.radius = LV_RADIUS_CIRCLE;
    sw_toggled.body.shadow.width = 4;
    sw_toggled.body.shadow.type = LV_SHADOW_BOTTOM;

//switch off
    lv_style_copy(&sw_off, &lv_style_pretty);
    sw_off.body.radius = LV_RADIUS_CIRCLE;
    sw_off.body.shadow.width = 4;
    sw_off.body.shadow.type = LV_SHADOW_BOTTOM;

//switch background
    lv_style_copy(&sw_bg, &lv_style_pretty);
    sw_bg.body.radius = LV_RADIUS_CIRCLE;

//switch indicator
    lv_style_copy(&sw_indic, &lv_style_pretty_color);
    sw_indic.body.radius = LV_RADIUS_CIRCLE;
    sw_indic.body.main_color = SW_INDIC;
    sw_indic.body.grad_color = SW_INDIC;
    sw_indic.body.padding.hor = 0;
    sw_indic.body.padding.ver = 0;

//heading text
    lv_style_copy(&heading_text, &lv_style_plain);
    heading_text.text.letter_space = 2;
    heading_text.text.line_space = 1;
    heading_text.text.color = TEXT;
    heading_text.text.font = &lv_font_dejavu_20;

//body text
    lv_style_copy(&body_text, &lv_style_plain);
    body_text.text.letter_space = 2;
    body_text.text.line_space = 1;
    body_text.text.color = BODY_TEXT;
    body_text.text.font = &dejavu_12;

//subheading text
```

```
    lv_style_copy(&subheading_text, &lv_style_plain);
    subheading_text.text.letter_space = 2;
    subheading_text.text.line_space = 1;
    subheading_text.text.color = BODY_TEXT;
    subheading_text.text.font = &dejavu_16;

//style for lines
    lv_style_copy(&lines, &lv_style_plain);
    lines.line.color = BUTTON_PR;
    lines.line.width = 5;

//styles for warning box
    //background
    lv_style_copy(&warn_box_bg, &lv_style_pretty);
    warn_box_bg.body.main_color = LV_COLOR_MAKE(0xf5, 0x45, 0x2e);
    warn_box_bg.body.grad_color = LV_COLOR_MAKE(0xb9, 0x1d, 0x09);
    warn_box_bg.body.border.color = LV_COLOR_MAKE(0x3f, 0x0a, 0x03);
    warn_box_bg.text.color = LV_COLOR_WHITE;
    warn_box_bg.body.padding.hor = 12;
    warn_box_bg.body.padding.ver = 8;
    warn_box_bg.body.shadow.width = 8;

    //button not pressed
    lv_style_copy(&warn_box_released, &lv_style_btn_rel);
    warn_box_released.body.empty = 1;
    warn_box_released.body.border.color = LV_COLOR_WHITE;
    warn_box_released.body.border.width = 2;
    warn_box_released.body.border.opa = LV_OPA_70;
    warn_box_released.body.padding.hor = 12;
    warn_box_released.body.padding.ver = 8;

    //button being pressed
    lv_style_copy(&warn_box_pressed, &warn_box_released);
    warn_box_pressed.body.empty = 0;
    warn_box_pressed.body.main_color = LV_COLOR_MAKE(0x5d, 0x0f, 0x04);
    warn_box_pressed.body.grad_color = LV_COLOR_MAKE(0x5d, 0x0f, 0x04);

//style for loader
    lv_style_copy(&loader_style, &lv_style_plain);
    loader_style.line.width = 10;  //10 px thick arc
    loader_style.line.color = LV_COLOR_HEX3(0x258); //Blueish arc color

    loader_style.body.border.color = LV_COLOR_HEX3(0xBBB); //Gray background color
    loader_style.body.border.width = 10;
    loader_style.body.padding.hor = 0;

}

Styles::~Styles()
{

}
```

../RobotCode/src/objects/lcdCode/TemporaryScreen.hpp

```
/**
 * @file: ./RobotCode/src/objects/lcdCode/TemporaryScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: deprecate, possibly move somewhere else, file does very little and could be merged elsewhere
 *
 * contains a global static screen that can be loaded so that the one screen needs to
 * be loaded at all times rule is not broken
 *
 */

#ifndef __TEMPORARYSCREEN_HPP__
#define __TEMPORARYSCREEN_HPP__

#include "../../../include/main.h"


struct tempScreen
{
    static lv_obj_t *temp_screen;
};



#endif
```

../RobotCode/src/objects/lcdCode/TemporaryScreen.cpp

```
/**
 * @file: ./RobotCode/src/objects/lcdCode/TemporaryScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: TemporaryScreen.hpp
 *
 * global screen part of a struct that can be loaded
 * has no parent so that it is always valid
 *
 */

#include "TemporaryScreen.hpp"
#include "Styles.hpp"
#include "../../../include/main.h"


lv_obj_t *tempScreen::temp_screen = lv_obj_create(NULL, NULL);
```

```
/**
 * @file: ./RobotCode/src/objects/lcdCode/gui.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: clean up conditionals, add config file
 *
 * contains auton selector gui selection all put together in one function
 *
 */
#ifndef __GUI_HPP__
#define __GUI_HPP__


#include "../include/main.h"

#include "AutonSelection/SelectionScreen.hpp"
#include "AutonSelection/OptionsScreen.hpp"
#include "AutonSelection/PrepScreen.hpp"
#include "AutonSelection/ActionsScreen.hpp"
#include "DriverControl/DriverControlLCD.hpp"
#include "../../DriverControl.hpp"
#include "Debug/Debug.hpp"
#include "TemporaryScreen.hpp"


/**
 * @return: int -> number of auton selected
 *
 * @see: ./AutonSelection
 * @see: ./Debug
 *
 * TODO: add more meaningful config options, clean up conditionals
 *
 * iterates and interacts with user to find final auton choice, and config options
 *
 */
int chooseAuton();





#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/lcdCode/gui.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/25/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: gui.hpp
 *
 * contains implementation of gui
 *
 */

#include "../../../include/main.h"

#include "gui.hpp"
#include "../../Autons.hpp"
#include "../motors/MotorThread.hpp"
#include "../../DriverControl.hpp"
#include "TemporaryScreen.hpp"


/**
 * iterates through selecting for user to go through stages selecting an auton or the debugger
 * and then all the config options
 * loads all screens at start so there are no mem management issues
 * finishes when all options are chosen
 */
int chooseAuton()
{
    Autons auton_data;

    //init screens so that loading time is faster
    SelectionScreen scr1;
    OptionsScreen scr2;
    PrepScreen scr3;

    int finalAutonChoice = 0;
    int auton = 1;
    bool confirm = false;
    int interval = 20;

    while ( !(finalAutonChoice) ) //allows user to go to previous screen
    {
        scr2.back = false;

        auton = scr1.selectAuton( auton ); //get auton option

        if ( auton == auton_data.driver_control_num ) {  //if prog with no auton is selected
            finalAutonChoice = 1;
        } else if ( auton == auton_data.debug_auton_num ) {  //if debugger is selected
            //starts driver control for debugging purposes
            MotorThread* motor_thread = MotorThread::get_instance();
            Motors::register_motors();
            motor_thread->start_thread();

            pros::Task driver_control_task (driver_control,
                                            (void*)NULL,
                                            TASK_PRIORITY_DEFAULT,
                                            TASK_STACK_DEPTH_DEFAULT,
                                            "DriverControlTask");

            debug();

            //ends driver control because it should not be enabled when
            //auton is being selected
            driver_control_task.remove();

            Motors::unregister_motors();
            motor_thread->stop_thread();
        } else {
            finalAutonChoice = auton;
        }

        // selection screen has been removed temporarily because options are not in use
        // else
        // {
        //     while ( !(scr2.back) && !(finalAutonChoice) )
        //     //if user selects a program with an auton
        //     {
        //         autonConfig cnfg = scr2.getOptions( auton ); //get config options
        //
        //         if ( !(scr2.back) ) //if user does not want to go back from screen 2
        //         {
        //
        //             scr3.getConfirmation( auton ); //gets confirmation from user
        //             if ( scr3.confirm )
        //             {
        //                 finalAutonChoice = auton;
        //             }
        //
        //
        //         }
        //         else
        //         {
        //             break;
        //         }
        //
        //
        //     }
        // }

    }



    return finalAutonChoice;
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/DriverControl/AutonomousLCD.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains the lcd screen used during auton
 *
 */
#ifndef __AUTONOMOUSLCD_HPP__
#define __AUTONOMOUSLCD_HPP__

#include "main.h"

#include "../Styles.hpp"


/**
 * @see: ../Styles.hpp
 *
 * contains lcd to be used during driver control
 */
class AutonomousLCD : private Styles
{
    private:
        lv_obj_t *screen;

        lv_obj_t *logo_img;


        //labels
        lv_obj_t *title_label;
        lv_obj_t *auton_label;
        lv_obj_t *description_label;


    public:
        AutonomousLCD();
        ~AutonomousLCD();


        /**
         * @param: int auton_number -> the autonomous number
         * @return: None
         *
         * TODO: add actual content to be updated
         *
         * function to be used to update the gui to keep data relevant
         */
        void update_labels(int auton_number);


        void log_to_lcd(std::string msg);


};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/DriverControl/AutonomousLCD.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * @see: DriverControlLCD.hpp
 *
 * contains methods for driver control lcd
 */

#include "main.h"

#include "../../../Autons.hpp"
#include "../../serial/Logger.hpp"
#include "../../position_tracking/PositionTracker.hpp"
#include "../AutonSelection/OptionsScreen.hpp"
#include "../Debug/Debug.hpp"
#include "AutonomousLCD.hpp"

LV_IMG_DECLARE(logo);


AutonomousLCD::AutonomousLCD()
{
    screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(screen, &gray);

// init labels
    title_label = lv_label_create(screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Autonomous");


    auton_label = lv_label_create(screen, NULL);
    lv_obj_set_style(auton_label, &subheading_text);
    lv_obj_set_width(auton_label, 300);
    lv_obj_set_height(auton_label, 20);
    lv_label_set_align(auton_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(auton_label, "");


    description_label = lv_label_create(screen, NULL);
    lv_obj_set_style(description_label, &subheading_text);
    lv_obj_set_width(description_label, 300);
    lv_obj_set_height(description_label, 20);
    lv_label_set_align(description_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(description_label, "");


// init image area
    logo_img = lv_img_create(screen, NULL);
    lv_img_set_src(logo_img, &logo);
    lv_img_set_auto_size(logo_img, false);
    lv_obj_set_width(logo_img, 210);
    lv_obj_set_height(logo_img, 150);


// place objects
    lv_obj_set_pos(title_label, 180, 9);

    lv_obj_set_pos(logo_img, 30, 40);

    lv_obj_set_pos(auton_label, 280, 40);
    lv_obj_set_pos(description_label, 280, 80);

}



AutonomousLCD::~AutonomousLCD()
{
    lv_obj_del(screen);
}




/**
 * updates colors and borders during driver control
 * keeps data relevent
 */
void AutonomousLCD::update_labels(int auton_number)
{
    Autons autons;

    lv_scr_load(screen);

    lv_label_set_text(auton_label, autons.AUTONOMOUS_NAMES.at(auton_number));
    lv_label_set_text(description_label, autons.AUTONOMOUS_DESCRIPTIONS.at(auton_number));
}


void AutonomousLCD::log_to_lcd(std::string msg) {
    lv_scr_load(screen);
    lv_label_set_text(description_label, msg.c_str());
}
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/DriverControl/DriverControlLCD.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: add actual content instead of blank screen
 *
 * contains the lcd screen used during driver control
 *
 */
#ifndef __DRIVERCONTROLLCD_HPP__
#define __DRIVERCONTROLLCD_HPP__

#include "main.h"

#include "../Styles.hpp"


/**
 * @see: ../Styles.hpp
 *
 * contains lcd to be used during driver control
 */
class DriverControlLCD : private Styles
{
    private:
        static bool log_data;
        static bool open_debugger;
        static std::string toggle_logging_text;

        lv_obj_t *screen;

        lv_obj_t *logo_img;


        //labels
        lv_obj_t *title_label;
        lv_obj_t *queue_size_label;

        //buttons
        lv_obj_t *btn_debugger;
        lv_obj_t *btn_run_auton;
        lv_obj_t *btn_toggle_logging;
        lv_obj_t *btn_flush_queue;

        lv_obj_t *btn_debugger_label;
        lv_obj_t *btn_run_auton_label;
        lv_obj_t *btn_toggle_logging_label;
        lv_obj_t *btn_flush_queue_label;


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to open debugger
         */
        static lv_res_t btn_debugger_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to run auton
         */
        static lv_res_t btn_run_auton_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to toggle logging of motors and other various items
         */
        static lv_res_t btn_toggle_logging_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to flush the logging queue
         */
        static lv_res_t btn_flush_queue_action(lv_obj_t *btn);


    public:
        DriverControlLCD();
        ˜DriverControlLCD();


        /**
         * @return: None
         *
         * TODO: add actual content to be updated
         *
         * function to be used to update the gui to keep data relevant
         */
        void update_labels();


};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/DriverControl/DriverControlLCD.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: DriverControlLCD.hpp
 *
 * contains methods for driver control lcd
 */

#include "main.h"

#include "../../../Autons.hpp"
#include "../../serial/Logger.hpp"
#include "../../position_tracking/PositionTracker.hpp"
#include "../AutonSelection/OptionsScreen.hpp"
#include "../Debug/Debug.hpp"
#include "DriverControlLCD.hpp"


bool DriverControlLCD::log_data = false;
bool DriverControlLCD::open_debugger = false;
std::string DriverControlLCD::toggle_logging_text = "Start Logging";
LV_IMG_DECLARE(logo);


DriverControlLCD::DriverControlLCD()
{
    log_data = false;

    screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(screen, &gray);

// init labels
    queue_size_label = lv_label_create(screen, NULL);
    lv_obj_set_style(queue_size_label, &subheading_text);
    lv_obj_set_width(queue_size_label, 300);
    lv_obj_set_height(queue_size_label, 20);
    lv_label_set_align(queue_size_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(queue_size_label, "Logger Queue Size: ");


    title_label = lv_label_create(screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Driver Control");

// init image area
    logo_img = lv_img_create(screen, NULL);
    lv_img_set_src(logo_img, &logo);
    lv_img_set_auto_size(logo_img, false);
    lv_obj_set_width(logo_img, 210);
    lv_obj_set_height(logo_img, 150);

// init buttons
    //button
    btn_debugger = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_debugger, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_debugger, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_debugger, LV_BTN_ACTION_CLICK, btn_debugger_action);
    lv_obj_set_width(btn_debugger, 180);
    lv_obj_set_height(btn_debugger, 25);

    //label
    btn_debugger_label = lv_label_create(btn_debugger, NULL);
    lv_obj_set_style(btn_debugger_label, &subheading_text);
    lv_label_set_text(btn_debugger_label, "Open Debugger");


    //button
    btn_run_auton = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_run_auton, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_run_auton, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_run_auton, LV_BTN_ACTION_CLICK, btn_run_auton_action);
    lv_obj_set_width(btn_run_auton, 180);
    lv_obj_set_height(btn_run_auton, 25);

    //label
    btn_run_auton_label = lv_label_create(btn_run_auton, NULL);
    lv_obj_set_style(btn_run_auton_label, &subheading_text);
    lv_label_set_text(btn_run_auton_label, "Run Auton");


    //button
    btn_toggle_logging = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_toggle_logging, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_toggle_logging, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_toggle_logging, LV_BTN_ACTION_CLICK, btn_toggle_logging_action);
    lv_obj_set_width(btn_toggle_logging, 180);
    lv_obj_set_height(btn_toggle_logging, 25);

    //label
    btn_toggle_logging_label = lv_label_create(btn_toggle_logging, NULL);
    lv_obj_set_style(btn_toggle_logging_label, &subheading_text);
    lv_label_set_text(btn_toggle_logging_label, toggle_logging_text.c_str());


    //button
    btn_flush_queue = lv_btn_create(screen, NULL);
    lv_btn_set_style(btn_flush_queue, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_flush_queue, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_flush_queue, LV_BTN_ACTION_CLICK, btn_flush_queue_action);
    lv_obj_set_width(btn_flush_queue, 180);
    lv_obj_set_height(btn_flush_queue, 25);

    //label
    btn_flush_queue_label = lv_label_create(btn_flush_queue, NULL);
    lv_obj_set_style(btn_flush_queue_label, &subheading_text);
    lv_label_set_text(btn_flush_queue_label, "Flush Logger Queue");

```

```
// place objects
    lv_obj_set_pos(title_label, 180, 9);
    lv_obj_set_pos(queue_size_label, 280, 200);
    lv_obj_set_pos(logo_img, 30, 40);

    lv_obj_set_pos(btn_debugger, 280, 40);
    lv_obj_set_pos(btn_run_auton, 280, 80);
    lv_obj_set_pos(btn_toggle_logging, 280, 120);
    lv_obj_set_pos(btn_flush_queue, 280, 160);

}



DriverControlLCD::~DriverControlLCD()
{
    lv_obj_del(screen);
}




lv_res_t DriverControlLCD::btn_debugger_action(lv_obj_t *btn)
{
    open_debugger = true;
    return LV_RES_OK;
}



lv_res_t DriverControlLCD::btn_run_auton_action(lv_obj_t *btn)
{
    pros::delay(3000);

    Autons auton_obj;

    Motors::disable_driver_control();

    auton_obj.run_autonomous();

    Motors::enable_driver_control();

    return LV_RES_OK;
}



lv_res_t DriverControlLCD::btn_toggle_logging_action(lv_obj_t *btn)
{
    if(log_data)
    {
        log_data = false;
    }
    else
    {
        log_data = true;
    }
    return LV_RES_OK;
}



lv_res_t DriverControlLCD::btn_flush_queue_action(lv_obj_t *btn)
{
    Logger logger;
    while(logger.get_count() > 0)
    {
        logger.dump();
    }
    return LV_RES_OK;
}





/**
 * updates colors and borders during driver control
 * keeps data relevent
 */
void DriverControlLCD::update_labels()
{
    lv_scr_load(screen);

    Logger logger;
    // PositionTracker* pos_tracker = PositionTracker::get_instance();

    // update toggle logging label text
    if(log_data)
    {
        lv_label_set_text(btn_toggle_logging_label, "Stop Logging");
        Motors::set_log_level(1);

        // pos_tracker->start_logging();

        Sensors::log_data();
    }
    else
    {
        lv_label_set_text(btn_toggle_logging_label, "Start Logging");
        Motors::set_log_level(0);

        // pos_tracker->stop_logging();
    }

    if(open_debugger)
    {
        debug();
        open_debugger = false;
    }

    //update logger queue size label
    std::string text = std::string("Logger Queue Size: ") + std::to_string(logger.get_count());
    lv_label_set_text(queue_size_label, text.c_str());
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/ActionsScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: add actual content for when actions are decided on
 *
 * does nothing
 *
 */

#ifndef __ACTIONSSCREEN_HPP__
#define __ACTIONSSCREEN_HPP__


#include "../include/main.h"




#endif
```

../RobotCode/src/objects/lcdCode/AutonSelection/ActionsScreen.cpp

```
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/ActionsScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: ActionsScreen.hpp
 *
 * does nothing
 *
 */
```

```
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/OptionsScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: add correct options when they are decided on, deprecate static options to reduce coupling
 *
 * contains class with methods to decide on options for auton
 *
 */

#ifndef __OPTIONSSCREEN_HPP__
#define __OPTIONSSCREEN_HPP__


#include "../../../../include/main.h"

#include "../../../Autons.hpp"
#include "../Styles.hpp"

/**
 * @see: ../Styles.hpp
 * @see: ./AutonSelection
 * @see: ../gui.hpp
 *
 * contains methods to get options for auton period
 */
class OptionsScreen : private Styles
{
    private:
        //screen
        lv_obj_t *options_screen;

        //labels
        lv_obj_t *title_label;

        lv_obj_t *sw_use_hardcoded_label;
        lv_obj_t *sw_gyro_turn_label;
        lv_obj_t *sw_accelleration_ctrl_label;
        lv_obj_t *sw_check_motor_tmp_label;
        lv_obj_t *sw_use_previous_macros_label;
        lv_obj_t *sw_record_label;

        //buttons
        lv_obj_t *btn_confirm;
        lv_obj_t *btn_back;

        //button labels
        lv_obj_t *btn_confirm_label;
        lv_obj_t *btn_back_label;


        //switches
        lv_obj_t *sw_use_hardcoded;
        lv_obj_t *sw_gyro_turn;
        lv_obj_t *sw_accelleration_ctrl;
        lv_obj_t *sw_check_motor_tmp;
        lv_obj_t *sw_use_previous_macros;
        lv_obj_t *sw_record;


    public:
        OptionsScreen();
        ~OptionsScreen();

        static autonConfig cnfg;
        static bool nextScreen;
        static bool back;


        //button functions

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui continues
         * to the next stage
         */
        static lv_res_t btn_confirm_action(lv_obj_t *btn);

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui goes back to
         * the previous stage
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


        //switch functions

        /**
         * @param: lv_obj_t* sw -> switch object that was selected
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         * TODO: merge with other functions to condense code and make it more expandable
         *
         * sets configuration option for using a compiled auton vs auton written on sd card
         */
        static lv_res_t sw_use_hardcoded_action(lv_obj_t *sw);

        /**
         * @param: lv_obj_t* sw -> switch object that was selected
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         * TODO: merge with other functions to condense code and make it more expandable
         *
         * sets configuration option for using gyro turns in auton
         */
        static lv_res_t sw_gyro_turn_action(lv_obj_t *sw);

        /**
         * @param: lv_obj_t* sw -> switch object that was selected
```

```
     * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
     * TODO: merge with other functions to condense code and make it more expandable
     *
     * sets configuration option for using acceleration control code
     */
    static lv_res_t sw_accelleration_ctrl_action(lv_obj_t *sw);

    /**
     * @param: lv_obj_t* sw -> switch object that was selected
     * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
     * TODO: merge with other functions to condense code and make it more expandable
     *
     * sets configuration option for limiting motor output based on temperature during the match
     */
    static lv_res_t sw_check_motor_tmp_action(lv_obj_t *sw);

    /**
     * @param: lv_obj_t* sw -> switch object that was selected
     * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
     * TODO: merge with other functions to condense code and make it more expandable
     *
     * sets configuration option for allowing the use of previously recorded macros
     */
    static lv_res_t sw_use_previous_macros_action(lv_obj_t *sw);

    /**
     * @param: lv_obj_t* sw -> switch object that was selected
     * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
     * TODO: merge with other functions to condense code and make it more expandable
     *
     * sets configuration option for recording match in the macro format
     */
    static lv_res_t sw_record_action(lv_obj_t *sw);



    /**
     * @param: int auton -> number of auton selected, used to set color background of lcd
     * @return: autonConfig -> configuration struct with options based on how the switches were set
     *
     * @see: ../Structs.hpp
     * @see: ../Gui.hpp
     *
     * allows user to interact with the switches to set configuration options
     * user can choose to go back or continue with the options selected
     */
    autonConfig getOptions( int auton );

};


#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/OptionsScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: OptionsScreen.hpp
 *
 * contains class with methods for getting a configuration struct for how
 * auton will be run
 */

#include <unordered_map>

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../../../Autons.hpp"
#include "../../controller/controller.hpp"
#include "OptionsScreen.hpp"


autonConfig OptionsScreen::cnfg;
bool OptionsScreen::nextScreen = false;
bool OptionsScreen::back = false;



OptionsScreen::OptionsScreen()
{
    nextScreen = false;
    back = false;

    options_screen = lv_obj_create(NULL, NULL);

//use hard coded autonomous
    //switch
    sw_use_hardcoded = lv_sw_create(options_screen, NULL); //switch template
    lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_BG, &sw_bg);
    lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_INDIC, &sw_indic);
    lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_KNOB_ON, &sw_toggled);
    lv_sw_set_style(sw_use_hardcoded, LV_SW_STYLE_KNOB_OFF, &sw_off);

    lv_sw_set_action(sw_use_hardcoded, sw_use_hardcoded_action); //map action
    lv_obj_set_width(sw_use_hardcoded, 40); //witdth
    lv_obj_set_height(sw_use_hardcoded, 20); //height

    //label
    sw_use_hardcoded_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_use_hardcoded_label, &heading_text);
    lv_obj_set_width(sw_use_hardcoded_label, 10);
    lv_obj_set_height(sw_use_hardcoded_label, 20);
    lv_label_set_align(sw_use_hardcoded_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_use_hardcoded_label, "Use Hardcoded Auton");



//gyro turns
    //switch
    sw_gyro_turn = lv_sw_create(options_screen, sw_use_hardcoded);
    lv_sw_set_action(sw_gyro_turn, sw_gyro_turn_action);
    lv_obj_set_width(sw_gyro_turn, 40);
    lv_obj_set_height(sw_gyro_turn, 20);

    //label
    sw_gyro_turn_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_gyro_turn_label, &heading_text);
    lv_obj_set_width(sw_gyro_turn_label, 300);
    lv_obj_set_height(sw_gyro_turn_label, 20);
    lv_label_set_align(sw_gyro_turn_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_gyro_turn_label, "Use Gyro Turns");


//acceleration control
    //switch
    sw_accelleration_ctrl = lv_sw_create(options_screen, sw_use_hardcoded);
    lv_sw_set_action(sw_accelleration_ctrl, sw_accelleration_ctrl_action);
    lv_obj_set_width(sw_accelleration_ctrl, 40);
    lv_obj_set_height(sw_accelleration_ctrl, 20);

    //label
    sw_accelleration_ctrl_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_accelleration_ctrl_label, &heading_text);
    lv_obj_set_width(sw_accelleration_ctrl_label, 300);
    lv_obj_set_height(sw_accelleration_ctrl_label, 20);
    lv_label_set_align(sw_accelleration_ctrl_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_accelleration_ctrl_label, "Use Acceleration Control");


//check motor temp
    //switch
    sw_check_motor_tmp = lv_sw_create(options_screen, sw_use_hardcoded);
    lv_sw_set_action(sw_check_motor_tmp, sw_check_motor_tmp_action);
    lv_obj_set_width(sw_check_motor_tmp, 40);
    lv_obj_set_height(sw_check_motor_tmp, 20);

    //label
    sw_check_motor_tmp_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_check_motor_tmp_label, &heading_text);
    lv_obj_set_width(sw_check_motor_tmp_label, 300);
    lv_obj_set_height(sw_check_motor_tmp_label, 20);
    lv_label_set_align(sw_check_motor_tmp_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_check_motor_tmp_label, "Limit Motor Temp");


//use previous macros
    //switch
    sw_use_previous_macros = lv_sw_create(options_screen, sw_use_hardcoded);
    lv_sw_set_action(sw_use_previous_macros, sw_use_previous_macros_action);
    lv_obj_set_width(sw_use_previous_macros, 40);
    lv_obj_set_height(sw_use_previous_macros, 20);

    //label
    sw_use_previous_macros_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_use_previous_macros_label, &heading_text);
    lv_obj_set_width(sw_use_previous_macros_label, 300);
    lv_obj_set_height(sw_use_previous_macros_label, 20);
    lv_label_set_align(sw_use_previous_macros_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_use_previous_macros_label, "Use Previously Recorded Macros");


//record
    //switch
    sw_record = lv_sw_create(options_screen, sw_use_hardcoded);
    lv_sw_set_action(sw_record, sw_record_action);
    lv_obj_set_width(sw_record, 40);
    lv_obj_set_height(sw_record, 20);

    //label
    sw_record_label = lv_label_create(options_screen, NULL);
    lv_label_set_style(sw_record_label, &heading_text);
    lv_obj_set_width(sw_record_label, 300);
    lv_obj_set_height(sw_record_label, 20);
    lv_label_set_align(sw_record_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(sw_record_label, "Record Motor Movements");


//confirm button
    //button
    btn_confirm = lv_btn_create(options_screen, NULL);
    lv_btn_set_style(btn_confirm, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_confirm, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_confirm, LV_BTN_ACTION_CLICK, btn_confirm_action);
    lv_obj_set_width(btn_confirm, 300);
    lv_obj_set_height(btn_confirm, 25);

    //label
    btn_confirm_label = lv_label_create(btn_confirm, NULL);
    lv_obj_set_style(btn_confirm_label, &heading_text);
    lv_label_set_text(btn_confirm_label, "Confirm");

//back button
    //button
    btn_back = lv_btn_create(options_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 50);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");


//title label
    title_label = lv_label_create(options_screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(title_label, "Auton");

//set postition of widgets
    lv_obj_set_pos(sw_use_hardcoded, 400, 40);
    lv_obj_set_pos(sw_gyro_turn, 400, 65);
    lv_obj_set_pos(sw_accelleration_ctrl, 400, 90);
    lv_obj_set_pos(sw_check_motor_tmp, 400, 115);
    lv_obj_set_pos(sw_use_previous_macros, 400, 140);
    lv_obj_set_pos(sw_record, 400, 165);

    lv_obj_set_pos(sw_use_hardcoded_label, 20, 40);
    lv_obj_set_pos(sw_gyro_turn_label, 20, 65);
    lv_obj_set_pos(sw_accelleration_ctrl_label, 20, 90);
    lv_obj_set_pos(sw_check_motor_tmp_label, 20, 115);
    lv_obj_set_pos(sw_use_previous_macros_label, 20, 140);
    lv_obj_set_pos(sw_record_label, 20, 165);

    lv_obj_set_pos(btn_back, 40, 200);
    lv_obj_set_pos(btn_confirm, 100, 200);
    lv_obj_set_pos(title_label, 210, 20);
}


OptionsScreen::~OptionsScreen()
{
    lv_obj_del(sw_use_hardcoded_label);
    lv_obj_del(sw_gyro_turn_label);
    lv_obj_del(sw_accelleration_ctrl_label);
    lv_obj_del(sw_check_motor_tmp_label);
    lv_obj_del(sw_use_previous_macros_label);
    lv_obj_del(sw_record_label);

    lv_obj_del(sw_use_hardcoded);
    lv_obj_del(sw_gyro_turn);
    lv_obj_del(sw_accelleration_ctrl);
    lv_obj_del(sw_check_motor_tmp);
    lv_obj_del(sw_use_previous_macros);
    lv_obj_del(sw_record);

    lv_obj_del(btn_back_label);
    lv_obj_del(btn_confirm_label);
    lv_obj_del(title_label);

    lv_obj_del(btn_back);
    lv_obj_del(btn_confirm);

    lv_obj_del(options_screen);

}



/**
 * sets nextScreen so that main loop will break and go to next stage
 */
lv_res_t OptionsScreen::btn_confirm_action(lv_obj_t *btn)
{
   nextScreen = true;
   back = false;

   return LV_RES_OK;
}

/**
 * sets nextScreen so that main loop will break and go to the previous stage
 */
lv_res_t OptionsScreen::btn_back_action(lv_obj_t *btn)
{
   nextScreen = true;
   back = true;

   return LV_RES_OK;
}




/**
 * sets or clears config option for using hard coded autons based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_use_hardcoded_action(lv_obj_t *sw)
{
   cnfg.use_hardcoded = !(cnfg.use_hardcoded);
   return LV_RES_OK;
}

/**
 * sets or clears config option for using gyro turns based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_gyro_turn_action(lv_obj_t *sw)
{
   cnfg.gyro_turn = !(cnfg.gyro_turn);
   return LV_RES_OK;
}


/**
 * sets or clears config option for using acceleration control code based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_accelleration_ctrl_action(lv_obj_t *sw)
{
   cnfg.accelleration_ctrl = !(cnfg.accelleration_ctrl);
   return LV_RES_OK;
}


/**
 * sets or clears config option for limiting motor output based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_check_motor_tmp_action(lv_obj_t *sw)
{
   cnfg.check_motor_tmp = !(cnfg.check_motor_tmp);
   return LV_RES_OK;
}


/**
 * sets or clears config option for allowing use of previously recorded macros based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_use_previous_macros_action(lv_obj_t *sw)
{
   cnfg.use_previous_macros = !(cnfg.use_previous_macros);
   return LV_RES_OK;
}


/**
 * sets or clears config option for recorded the match as a macro based on the
 * switches previous state
 */
lv_res_t OptionsScreen::sw_record_action(lv_obj_t *sw)
{
   cnfg.record = !(cnfg.record);
   return LV_RES_OK;
}



/**
 * runs loop where user can set auton configuration options with digital switches
 * loop breaks when user clicks the back or continue button
 * if user click the back button then the back flag is set
 */
autonConfig OptionsScreen::getOptions( int auton )
{
   Controller controllers;
   Autons auton_data;

   lv_sw_off(sw_use_hardcoded); //reset switches and values
   lv_sw_on(sw_gyro_turn);
   lv_sw_on(sw_accelleration_ctrl);
   lv_sw_off(sw_check_motor_tmp);
   lv_sw_on(sw_use_previous_macros);
   lv_sw_off(sw_record);

   cnfg.use_hardcoded = 0;
   cnfg.gyro_turn = 1;
   cnfg.accelleration_ctrl = 1;
   cnfg.check_motor_tmp = 0;
   cnfg.use_previous_macros = 1;
   cnfg.record = 0;

```

```

    lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton));

    //load screen
    lv_scr_load(options_screen);

    //set color of border
    std::string color = auton_data.AUTONOMOUS_COLORS.at(auton);
    if (color == "blue")
    {
        gray.body.border.color = BLUE_BORDER;
    }
    else if (color == "red")
    {
        gray.body.border.color = RED_BORDER;
    }
    else
    {
        gray.body.border.color = BG;
    }

    lv_obj_set_style(options_screen, &gray);


    back = false;
    nextScreen = false;

    pros::delay( 100 );  //add delay so that previous button clicks do not register

    while ( !(nextScreen) )
    {
        //allow controller to press the buttons as well
        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
        {
            btn_confirm_action( NULL );
            pros::delay(100);
        }
        else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
        {
            btn_back_action( NULL );
            pros::delay(100);
        }

        pros::delay(20);
    }

    return cnfg;

}
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/PrepScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: add actual preparation steps text
 * TODO: decouple initilization string, make it more configurable
 *
 * contains class with methods for showing the things that will occur
 * before the auton is run
 */

#ifndef __PREP_SCREEN__
#define __PREP_SCREEN__


#include "../include/main.h"

#include "../Styles.hpp"
#include "OptionsScreen.hpp"


/**
 * @see: ../Styles.hpp
 * @see: ../gui.hpp
 *
 * final confirmation steps for auton
 * shows the initialization that will occur after the user clicks continue
 */
class PrepScreen : private Styles
{
    private:
        //screen
        lv_obj_t *prep_screen;

        //labels
        lv_obj_t *title_label;

        lv_obj_t *actions_label;

        //buttons
        lv_obj_t *btn_confirm;
        lv_obj_t *btn_back;

        //button labels
        lv_obj_t *btn_confirm_label;
        lv_obj_t *btn_back_label;


    public:
        PrepScreen();
        ~PrepScreen();

        static bool nextScreen;
        static bool confirm;

        //button functions

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui continues
         * to the next stage
         */
        static lv_res_t btn_confirm_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui goes
         * to the previous stage
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


        /**
         * @param: int auton -> auton number selected, used to set border color of gui based on side of color the auton is run on
         * @return: None
         *
         * @see: ../Structs.hpp
         * @see: ../Styles.hpp
         *
         * gunction used to get confirmation from user to continue to next stage
         * of the selection process
         */
        void getConfirmation( int auton );

};
#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/PrepScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: PrepScreen.hpp
 *
 * contains class methods seeking confirmation from user
 */

#include <sstream>
#include <string>

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../../../Autons.hpp"
#include "../../controller/controller.hpp"
#include "PrepScreen.hpp"


bool PrepScreen::nextScreen = false;
bool PrepScreen::confirm = false;



PrepScreen::PrepScreen()
{
    nextScreen = false;
    confirm = false;

    prep_screen = lv_obj_create(NULL, NULL);

//actions_label
    actions_label = lv_label_create(prep_screen, NULL);
    lv_obj_set_style(actions_label, &heading_text);
    lv_obj_set_width(actions_label, 300);
    lv_obj_set_height(actions_label, 160);
    lv_label_set_align(actions_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(actions_label, "actions");



//confirm button
    //button
    btn_confirm = lv_btn_create(prep_screen, NULL);
    lv_btn_set_style(btn_confirm, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_confirm, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_confirm, LV_BTN_ACTION_CLICK, btn_confirm_action);
    lv_obj_set_width(btn_confirm, 300);
    lv_obj_set_height(btn_confirm, 25);

    //label
    btn_confirm_label = lv_label_create(btn_confirm, NULL);
    lv_obj_set_style(btn_confirm_label, &heading_text);
    lv_label_set_text(btn_confirm_label, "Confirm");

//back button
    //button
    btn_back = lv_btn_create(prep_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 50);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");


//title label
    title_label = lv_label_create(prep_screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(title_label, "Auton");


    lv_obj_set_pos(btn_back, 40, 200);
    lv_obj_set_pos(btn_confirm, 100, 200);
    lv_obj_set_pos(title_label, 210, 20);
    lv_obj_set_pos(actions_label, 40, 50);
}




PrepScreen::~PrepScreen()
{
    lv_obj_del(btn_back_label);
    lv_obj_del(btn_confirm_label);
    lv_obj_del(title_label);

    lv_obj_del(btn_back);
    lv_obj_del(btn_confirm);

    lv_obj_del(prep_screen);
}




/**
 * sets nextScreen so that main loop will break and go to next stage
 */
lv_res_t PrepScreen::btn_confirm_action(lv_obj_t *btn)
{
    nextScreen = true;
    confirm = true;

    return LV_RES_OK;
}

/**
 * sets nextScreen so that main loop will break and go to the previous stage
 */
lv_res_t PrepScreen::btn_back_action(lv_obj_t *btn)
{
    nextScreen = true;
    confirm = false;

    return LV_RES_OK;
}



/**
 * runs loop where user can see what operations will be performed
 * loop breaks when user clicks the back or continue button
 * if user click the back button then the back flag is set
 */
void PrepScreen::getConfirmation( int auton )
{
    Controller controllers;
    Autons auton_data;

    lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton));

    //load screen
    lv_scr_load(prep_screen);

    //set color of border
    std::string color = auton_data.AUTONOMOUS_COLORS.at(auton);
    if (color == "blue")
    {
        gray.body.border.color = BLUE_BORDER;
    }
    else if (color == "red")
    {
        gray.body.border.color = RED_BORDER;
    }
    else
    {
        gray.body.border.color = BG;
    }

    lv_obj_set_style(prep_screen, &gray);

    nextScreen = false;


    std::string label =
        "Initialize and Calibrate Gyro\n"
        "Initialize Other Sensors\n"
        "Initialize Motors\n"
        "Zero Motor Encoders\n"
        "Initialize Controllers\n";
    if ( OptionsScreen::cnfg.record )
    {
        label = label + "Start Recording Thread";
    }

    //cast std::string to const char* and set text
    std::ostringstream text;
    text << label;
    lv_label_set_text(actions_label, text.str().c_str());

    pros::delay( 100 );  //add delay so that button press from previous stage does not register

    while ( !(nextScreen) )
    {
        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
        {
            btn_confirm_action( NULL );
            pros::delay(200);
        }
        else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
        {
            btn_back_action( NULL );
            pros::delay(200);
        }
        pros::delay(20);
    }

}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/SelectionScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: add ability for controller to make selections
 *
 * contains first stage of auton selection--selecting the auton number
 *
 */

#ifndef __SELECTIONSCREEN_HPP__
#define __SELECTIONSCREEN_HPP__


#include "../../../../include/main.h"

#include "../Styles.hpp"




/**
 * @see: ../Styles.hpp
 *
 * contains methods for going through each auton and providing description
 * for user
 */
class SelectionScreen : private Styles
{
    private:

        //labels
        lv_obj_t *title_label;
        lv_obj_t *description_label;
        lv_obj_t *auton_number_label;

        //buttons
        lv_obj_t *btn_right;
        lv_obj_t *btn_left;
        lv_obj_t *btn_select;

        //button labels
        lv_obj_t *btn_right_label;
        lv_obj_t *btn_left_label;
        lv_obj_t *btn_select_label;


    public:
        //screens
        static lv_obj_t *selection_screen;
        //variables
        static int auton_choice;
        static int final_choice;
        static bool update;

        //button action functions

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui moves
         * autons to the right (increasing auton number by one and looping at end)
         */
        static lv_res_t btn_right_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set variable so that gui moves
         * autons to the left (decreasing auton number by one and looping at end)
         */
        static lv_res_t btn_left_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to select the auton so that the gui continues
         * to the next stage
         */
        static lv_res_t btn_select_action(lv_obj_t *btn);

        SelectionScreen();
        ~SelectionScreen();



        /**
         * @return: None
         *
         * @see: int selectAuton()
         * @see: ../Structs.hpp
         *
         * sets the description, color, title, and auton number on the screen
         * used to update the auton selection based on the current number
         */
        void showSelection();


        /**
         * @param: int auton -> auton number to start the screen at
         * @return: int -> auton number that the screen was on when the user hit the select button
         *
         * loops through waiting for the auton number to change to update the screen
         */
        int selectAuton( int auton );


};



#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/AutonSelecton/SelectionScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: SelectionScreen.hpp
 *
 * contains methods for class that give user the ability to select an auton number
 */

#include <sstream>
#include <string>

#include "../../../../include/main.h"

#include "../../../Autons.hpp"
#include "SelectionScreen.hpp"
#include "../../controller/controller.hpp"

//init static vars
lv_obj_t *SelectionScreen::selection_screen;
int SelectionScreen::auton_choice = 1;
int SelectionScreen::final_choice = 0;
bool SelectionScreen::update = false;


//constructor
SelectionScreen::SelectionScreen()

{
    auton_choice = 1;
    final_choice = 0;
    update = false;

    selection_screen = lv_obj_create(NULL, NULL);


    //init buttons and labels
    title_label = lv_label_create(selection_screen, NULL);
    description_label = lv_label_create(selection_screen, NULL);
    auton_number_label = lv_label_create(selection_screen, NULL);

    btn_right = lv_btn_create(selection_screen, NULL);
    btn_left = lv_btn_create(selection_screen, NULL);
    btn_select = lv_btn_create(selection_screen, NULL);

    btn_right_label = lv_label_create(btn_right, NULL);
    btn_left_label = lv_label_create(btn_left, NULL);
    btn_select_label = lv_label_create(btn_select, NULL);


    //sets style for widgets
    lv_obj_set_style(selection_screen, &gray);

    lv_btn_set_style(btn_right, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_left, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_select, LV_BTN_STYLE_REL, &toggle_btn_released);

    lv_btn_set_style(btn_right, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_style(btn_left, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_style(btn_select, LV_BTN_STYLE_PR, &toggle_btn_pressed);

    lv_obj_set_style(btn_right_label, &heading_text);
    lv_obj_set_style(btn_left_label, &heading_text);
    lv_obj_set_style(btn_select_label, &heading_text);

    lv_label_set_style(title_label, &heading_text);
    lv_label_set_style(description_label, &heading_text);
    lv_label_set_style(auton_number_label, &heading_text);

    lv_label_set_long_mode(description_label, LV_LABEL_LONG_BREAK);
    lv_label_set_align(auton_number_label, LV_LABEL_ALIGN_CENTER);

    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_align(description_label, LV_LABEL_ALIGN_CENTER);


    //set size of widgets
    lv_obj_set_width(btn_right, 80);
    lv_obj_set_width(btn_left, 80);
    lv_obj_set_width(btn_select, 120);
    lv_obj_set_width(auton_number_label, 40);
    lv_obj_set_width(title_label, 400);
    lv_obj_set_width(description_label, 400);

    lv_obj_set_height(btn_right, 80);
    lv_obj_set_height(btn_left, 80);
    lv_obj_set_height(btn_select, 40);
    lv_obj_set_height(auton_number_label, 40);
    lv_obj_set_height(title_label, 30);
    lv_obj_set_height(description_label, 80);


    //set default text and move widgets to start location
    lv_label_set_text(btn_right_label, SYMBOL_RIGHT);
    lv_label_set_text(btn_left_label, SYMBOL_LEFT);
    lv_label_set_text(btn_select_label, "Select");

    lv_obj_set_pos(btn_right, 390, 150);
    lv_obj_set_pos(btn_left, 10, 150);
    lv_obj_set_pos(btn_select, 180, 180);
    lv_obj_set_pos(auton_number_label, 440, 20);
    lv_obj_set_pos(title_label, 210, 20);
    lv_obj_set_pos(description_label, 40, 60);


    //set action for buttons
    lv_btn_set_action(btn_right, LV_BTN_ACTION_CLICK, btn_right_action);
    lv_btn_set_action(btn_left, LV_BTN_ACTION_CLICK, btn_left_action);
    lv_btn_set_action(btn_select, LV_BTN_ACTION_CLICK, btn_select_action);

}
```

```cpp


//destructor
SelectionScreen::~SelectionScreen()
{
    lv_obj_del(auton_number_label);
    lv_obj_del(title_label);
    lv_obj_del(description_label);
    lv_obj_del(btn_right);
    lv_obj_del(btn_left);
    lv_obj_del(btn_select);

    lv_obj_del(selection_screen);
}



//button action functions

/**
 * called when left button is clicked
 * decrements auton_choice and loops it back in range if not in range
 */
lv_res_t SelectionScreen::btn_left_action(lv_obj_t *btn)
{
    Autons auton_data;

    auton_choice -= 1;
    if (auton_choice < 1)
    {
        auton_choice = auton_data.AUTONOMOUS_NAMES.size();
    }

    update = true;

    return LV_RES_OK;
}


/**
 * called when left button is clicked
 * increments auton_choice and loops it back in range if not in range
 */
lv_res_t SelectionScreen::btn_right_action(lv_obj_t *btn)
{
    std::cout << "function called\n";
    Autons auton_data;

    auton_choice += 1;
    if (auton_choice > auton_data.AUTONOMOUS_NAMES.size())
    {
        auton_choice = 1;
    }

    update = true;

    return LV_RES_OK;
}


/**
 * breaks main loop by setting the final auton choice so that gui continues
 */
lv_res_t SelectionScreen::btn_select_action(lv_obj_t *btn)
{
    final_choice = auton_choice;
    update = true;

    return LV_RES_OK;
}




//other functions

/**
 * updates background color by looking at std::unordered_map
 * updates auton number label
 * waits for there to be an update to be implemented by the buttons before exiting
 */
void SelectionScreen::showSelection()
{
    Controller controllers;
    Autons auton_data;

    lv_label_set_text(title_label, auton_data.AUTONOMOUS_NAMES.at(auton_choice));
    lv_label_set_text(description_label, auton_data.AUTONOMOUS_DESCRIPTIONS.at(auton_choice));


    //get color
    std::string color = auton_data.AUTONOMOUS_COLORS.at(auton_choice);
    if (color == "blue")
    {
        gray.body.border.color = BLUE_BORDER;
    }
    else if (color == "red")
    {
        gray.body.border.color = RED_BORDER;
    }
    else
    {
        gray.body.border.color = BG;
    }

    lv_obj_set_style(selection_screen, &gray); //update background

    //cast int of auton choice to string
    std::string str_auton_choice;
    str_auton_choice = std::to_string(auton_choice);
    lv_label_set_text(auton_number_label, str_auton_choice.c_str());


    controllers.master.print( 0, 0, "                    ");
    pros::delay(50);
    controllers.master.print( 0, 0, auton_data.AUTONOMOUS_NAMES.at(auton_choice) );

    while ( !(update) && !(final_choice) ) //waits for screen to change
                                           //so that time is not wasted
    {
        //allow controller to press the buttons as well
        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_R1) )
        {
            btn_right_action( NULL );
            pros::delay(200);
        }
        else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_L1) )
        {
            btn_left_action( NULL );
            pros::delay(200);
        }
        else if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_A) )
        {
            btn_select_action( NULL );
            pros::delay(200);
        }
        pros::delay(100);
    }
    update = false;
}




/**
 * waits in a loop for there to be an update to the gui implemented by
 * button callback functions
 * in the loop, the gui is updated until a final selection is made
 * everytime there is a change ie. when a button is clicked
 */
int SelectionScreen::selectAuton( int auton )
{
    auton_choice = auton;
    final_choice = 0;
    update = false;

    //load screen
    lv_scr_load(selection_screen);

    while ( !(final_choice) ) //waits for user to select an auton
    {                         //before going to next screen
        showSelection(); //showSelection contains delay
    }

    gray.body.border.color = BG; //reset gray style

    return final_choice;

}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/Debug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * gives user the option to visit debugger tabs by selecting an option from a button
 * matrix
 */

#ifndef __DEBUG_HPP__
#define __DEBUG_HPP__

#include "BatteryDebug.hpp"
#include "ControllerDebug.hpp"
#include "FieldControlDebug.hpp"
#include "InternalMotorDebug.hpp"
#include "MotorsDebug.hpp"
#include "SensorsDebug.hpp"
#include "TitleScreen.hpp"
#include "Wiring.hpp"



/**
 * @return: None
 *
 * @see: TitleScreen.hpp
 *
 * loads screens and switches the debugger option based on a what is clicked from
 * a button matrix
 */
void debug();


#endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/Debug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: Debug.hpp
 *
 * contains function for selecting debug screen
 *
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "Debug.hpp"



/**
 * loads all screens at beginning
 * when on titlescreen a tab number is selected and a switch statement is used
 * to let a tab take over
 */
void debug()
{
    bool cont = true;

    TitleScreen dbg1;
    MotorsDebug dbgM;
    SensorsDebug dbgS;
    ControllerDebug dbgC;
    BatteryDebug dbgB;
    FieldControlDebug dbgF;
    Wiring dbgW;
    InternalMotorDebug dbgP;


    while ( cont )
    {
        dbg1.chooseOption();
        if ( dbg1.option == -1 ) //-1 means go back
        {
            cont = false;
            break;
        }

        switch (dbg1.option) //go to selected debug screen
        {
            case 1:
                dbgM.debug();
                break;

            case 2:
                dbgS.debug();
                break;

            case 3:
                dbgC.debug();
                break;

            case 4:
                dbgB.debug();
                break;

            case 5:
                dbgF.debug();
                break;

            case 6:
                dbgW.debug();
                break;
            case 7:
                dbgP.debug();
                break;
        }

    }

}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/BatteryDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * contains class for debugging the battery
 */

#ifndef __BATTERYDEBUG_HPP__
#define __BATTERYDEBUG_HPP__


#include "../../../../include/main.h"

#include "../Styles.hpp"


/**
 * @see: ../Styles.hpp
 *
 * contains methods that show user data about the battery
 */
class BatteryDebug : private Styles
{
    private:
        lv_obj_t *battery_screen;
        lv_obj_t *title_label;

        lv_obj_t *labels_label;
        lv_obj_t *info_label;


        //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);

    public:
        static bool cont;

        BatteryDebug();
        ~BatteryDebug();


        /**
         * @return: None
         *
         * allows user to see information about the state of field control
         */
        void debug();

};




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/BatteryDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see BatteryDebug.hpp
 *
 * contains implementation for class for debugging the battery
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "BatteryDebug.hpp"


bool BatteryDebug::cont = true;

BatteryDebug::BatteryDebug()
{
    cont = true;

//screen
    battery_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(battery_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(battery_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Battery - Debug");

//init headers label
    labels_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(labels_label, &subheading_text);
    lv_obj_set_width(labels_label, 220);
    lv_obj_set_height(labels_label, 200);
    lv_label_set_align(labels_label, LV_LABEL_ALIGN_LEFT);

    std::string labels_label_text = (
            "battery percentage\n"
            "current\n"
            "voltage\n"
            "temperature"
    );

    lv_label_set_text(labels_label, labels_label_text.c_str());

//init values label
    info_label = lv_label_create(battery_screen, NULL);
    lv_label_set_style(info_label, &subheading_text);
    lv_obj_set_width(info_label, 220);
    lv_obj_set_height(info_label, 200);
    lv_label_set_align(info_label, LV_LABEL_ALIGN_LEFT);

    std::string info_label_text = (
        "None\n"
        "None\n"
        "None\n"
        "None"
    );

    lv_label_set_text(info_label, info_label_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_align(title_label, battery_screen, LV_ALIGN_IN_TOP_MID, 0, 10);

    lv_obj_set_pos(labels_label, 20, 40);
    lv_obj_set_pos(info_label, 360, 40);


}



BatteryDebug::~BatteryDebug()
{
    lv_obj_del(battery_screen);
}



/**
 * sets cont to false to break main loop so main function returns
 */
lv_res_t BatteryDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}
```

```
/**
 * main loop that updates the battery information
 */
void BatteryDebug::debug()
{
    cont = true;

    lv_scr_load(battery_screen);

    while ( cont )
    {
        std::string info_label_text = (
            std::to_string(pros::battery::get_capacity()) + "%\n"
            + std::to_string(pros::battery::get_current()) + "\n"
            + std::to_string(pros::battery::get_voltage()) + "\n"
            + std::to_string(pros::battery::get_temperature()) + "\n"
        );

        lv_label_set_text(info_label, info_label_text.c_str());


        pros::delay(100);
    }
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/ControllerDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 * TODO:
 *
 * contains classes to show data about controller on gui
 */

#ifndef __CONTROLLERDEBUG_HPP__
#define __CONTROLLERDEBUG_HPP__

#include <unordered_map>

#include "../../../../include/main.h"

#include "../Styles.hpp"

//user defines

//sets size of container
#define CONTROLLER_CONTAINER_WIDTH 440
#define CONTROLLER_CONTAINER_HEIGHT 120

//Base classes
//base classes are the tabs that will be loaded by the derived class
//this makes it easy to add new tabs while keeping the amount that has
//to go in one class to a minimum, especially since lvgl is not light



/**
 * @see: ../Styels.hpp
 * @see: ../../controller/controller.hpp
 *
 * contains general information about controllers and also allow
 * to test communication by sending rumbles and test strings
 */
class GeneralControllerDebug : virtual Styles
{
    private:
        lv_obj_t *container;

        lv_obj_t *controller_column;
        lv_obj_t *connected_column;
        lv_obj_t *capacity_column;
        lv_obj_t *level_column;



        lv_obj_t *btn_test_string;
        lv_obj_t *btn_test_string_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         * TODO: sometimes string does not send to the write location, fix
         *
         * sends a simple string to the controller lcd to test where output is
         * when the user clicks a button
         */
        static lv_res_t btn_test_string_action(lv_obj_t *btn);



        lv_obj_t *btn_clear_scr;
        lv_obj_t *btn_clear_scr_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         * TODO: sometimes doesn't actually work
         *
         * clears any output on the lcd controller when user clicks a button
         */
        static lv_res_t btn_clear_scr_action(lv_obj_t *btn);




        lv_obj_t *btn_test_rumble;
        lv_obj_t *btn_test_rumble_label;

        /**
         * @param: lv_obj_t* btn -> button that called the function
         * @return: lv_res_t -> return LV_RES_OK because object still exists
         *
         * tells controller to rumble
         */
        static lv_res_t btn_test_rumble_action(lv_obj_t *btn);



        lv_obj_t *controls_info;
        lv_obj_t *motor2_info;

    protected:
        /**
         * @return: None
         *
         * @see: ../../controller/controller.hpp
         *
         * updates data for the controllers such as connected or not, and battery
         */
        void update_general_info();

    public:
        GeneralControllerDebug();
        virtual ~GeneralControllerDebug();

        /**
         * @param: lv_obj_t* parent -> new parent for the main container
         * @return: None
         * TODO: depracate and fix method of inheritence, current method is not implemented well
         *
         * changes parent of containter
         */
        void GeneralControllerDebugInit(lv_obj_t *parent);

};





/**
 * @see: ../Styels.hpp
 * @see: ../../controller/controller.hpp
 *
 * generic class for showing the functions or values of the controller for each button
 */
class ControllerTab : virtual Styles
{
    private:
        lv_obj_t *container;

        //separated into two columns because LCD is not big enough
        //column one widgets
        lv_obj_t *button_names_one;
        lv_obj_t *button_col_one;

        //column two widgets
        lv_obj_t *button_names_two;
        lv_obj_t *button_col_two;

    public:
        ControllerTab(lv_obj_t *parent);
        virtual ˜ControllerTab();


        /**
         * @param: pros::controller_id_e_t controller -> controller that the tab is looking at
         * @param: bool showing_values -> bool for if values should be shown or not
         * @return: None
         * TODO: make controller a member so that controller does not have to be a parameter
         *
         * shows either the values or the function the controller calls on the gui
         */
        void update(pros::controller_id_e_t controller, bool showing_values);
};





//derived class


/**
 * @see: ../Styles.hpp
 *
 * tab for showing data about controllers
 */
class ControllerDebug :
    virtual private Styles,
    private GeneralControllerDebug
{
    private:
        static bool showing_values;

        //screen
        lv_obj_t *controller_debug_screen;

        //title label
        lv_obj_t *title_label;



        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);



        lv_obj_t *btn_show_values;
        static lv_obj_t *btn_show_values_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function switch between showing the functions or values of
         * for the controller
         */
        static lv_res_t btn_show_values_action(lv_obj_t *btn);



        static lv_obj_t *tabview; //tabview object

        //individual tabs
        //content will come from base classes
        lv_obj_t *general_tab;
        lv_obj_t *master_tab;
        lv_obj_t *partner_tab;


    public:
        ControllerDebug();
        ˜ControllerDebug();

        static bool cont; //checks whether to keep letting user
                          //cycle through tabs


        /**
        * @return: None
        *
        * allows user to see information about the controller
        */
        void debug();
};




#endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/ControllerDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: ControllerDebug.hpp
 *
 * contains implementation for classes that show information about the controller
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include <unordered_map>

#include "../Styles.hpp"
#include "../../controller/controller.hpp"
#include "ControllerDebug.hpp"


//declare static members of all classes
bool ControllerDebug::showing_values = 0;
bool ControllerDebug::cont = true;
lv_obj_t *ControllerDebug::tabview;
lv_obj_t *ControllerDebug::btn_show_values_label;



GeneralControllerDebug::GeneralControllerDebug()
{
//init container
    container = lv_cont_create(lv_scr_act(), NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, CONTROLLER_CONTAINER_WIDTH);
    lv_obj_set_height(container, CONTROLLER_CONTAINER_HEIGHT);

//default text for each column
    std::string text1 = (
        "Controller\n"
        "Master\n"
        "Partner"
    );

    std::string text2 = (
        "Connected\n"
        "no\n"
        "no"
    );

    std::string text3 = (
        "Battery Capacity\n"
        "0\n"
        "0"
    );

    std::string text4 = (
        "Battery Percent\n"
        "0\n"
        "0"
    );

//init controller column label
    controller_column = lv_label_create(container, NULL);
    lv_obj_set_style(controller_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(controller_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(controller_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(controller_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(controller_column, text1.c_str());

//init connected column label
    connected_column = lv_label_create(container, NULL);
    lv_obj_set_style(connected_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(connected_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(connected_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(connected_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(connected_column, text2.c_str());

//init capacity column label
    capacity_column = lv_label_create(container, NULL);
    lv_obj_set_style(capacity_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(capacity_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(capacity_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(capacity_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(capacity_column, text3.c_str());

//init battery percentage column label
    level_column = lv_label_create(container, NULL);
    lv_obj_set_style(level_column, &toggle_tabbtn_pressed);
    lv_obj_set_width(level_column, (CONTROLLER_CONTAINER_WIDTH / 4));
    lv_obj_set_height(level_column, CONTROLLER_CONTAINER_HEIGHT);
    lv_label_set_align(level_column, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(level_column, text4.c_str());


//init send test string button
    //button
    btn_test_string = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_test_string, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_test_string, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_test_string, LV_BTN_ACTION_CLICK, btn_test_string_action);
    lv_obj_set_width(btn_test_string, 130);
    lv_obj_set_height(btn_test_string, 25);

    //label
    btn_test_string_label = lv_label_create(btn_test_string, NULL);
    lv_obj_set_style(btn_test_string_label, &subheading_text);
    lv_label_set_text(btn_test_string_label, "Send Test String");


//init clear screen button
```

```
    //button
    btn_clear_scr = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_clear_scr, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_clear_scr, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_clear_scr, LV_BTN_ACTION_CLICK, btn_clear_scr_action);
    lv_obj_set_width(btn_clear_scr, 130);
    lv_obj_set_height(btn_clear_scr, 25);

    //label
    btn_clear_scr_label = lv_label_create(btn_clear_scr, NULL);
    lv_obj_set_style(btn_clear_scr_label, &subheading_text);
    lv_label_set_text(btn_clear_scr_label, "Clear Screen");


//init send test rumble button
    //button
    btn_test_rumble = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_test_rumble, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_test_rumble, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_test_rumble, LV_BTN_ACTION_CLICK, btn_test_rumble_action);
    lv_obj_set_width(btn_test_rumble, 130);
    lv_obj_set_height(btn_test_rumble, 25);

    //label
    btn_test_rumble_label = lv_label_create(btn_test_rumble, NULL);
    lv_obj_set_style(btn_test_rumble_label, &subheading_text);
    lv_label_set_text(btn_test_rumble_label, "Send Test Rumble");


//set positions relative to container
    lv_obj_align(controller_column, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(connected_column, container, LV_ALIGN_IN_TOP_MID, -80, 10);
    lv_obj_align(capacity_column, container, LV_ALIGN_IN_TOP_MID, 15, 10);
    lv_obj_align(level_column, container, LV_ALIGN_IN_TOP_RIGHT, -30, 10);


    lv_obj_align(btn_test_string, container, LV_ALIGN_IN_BOTTOM_LEFT, 20, -30);
    lv_obj_align(btn_clear_scr, container, LV_ALIGN_IN_BOTTOM_MID, 0, -30);
    lv_obj_align(btn_test_rumble, container, LV_ALIGN_IN_BOTTOM_RIGHT, -20, -30);

}


GeneralControllerDebug::~GeneralControllerDebug()
{

}




/**
 * sends test string to controller
 */
lv_res_t GeneralControllerDebug::btn_test_string_action(lv_obj_t *btn)
{
    Controller::master.print(0, 0, "This is a test message");
    Controller::partner.print(0, 0, "This is a test message");
    return LV_RES_OK;
}


/**
 * clears the screen on the controller
 */
lv_res_t GeneralControllerDebug::btn_clear_scr_action(lv_obj_t *btn)
{
    Controller::master.print(0, 0, "");
    Controller::partner.print(0, 0, "");
    Controller::master.clear_line(0);
    Controller::partner.clear_line(0);
    return LV_RES_OK;
}


/**
 * sends a test rumble to the controller
 */
lv_res_t GeneralControllerDebug::btn_test_rumble_action(lv_obj_t *btn)
{
    Controller::master.rumble(". - . - ");
    Controller::partner.rumble(". - . - ");
    return LV_RES_OK;
}




/**
 * updates data on the tab
 */
void GeneralControllerDebug::update_general_info()
{
    std::string text1 = (
        "Controller\n"
        "Master\n"
        "Partner"
    );

    std::string master_text = "no";
    std::string partner_text = "no";
    if ( Controller::master.is_connected() )
    {
        master_text = "yes";
    }
    if ( Controller::partner.is_connected() )
    {
        partner_text = "yes";
    }

    std::string text2 = (
        "Connected\n"
        + master_text + "\n"
        + partner_text
    );


    std::string text3 = (
        "Battery Capacity\n"
        + std::to_string(Controller::master.get_battery_level()) + "\n"
        + std::to_string(Controller::partner.get_battery_level())
    );

    std::string text4 = (
        "Battery Percent\n"
        + std::to_string(Controller::master.get_battery_capacity()) + "\n"
        + std::to_string(Controller::partner.get_battery_capacity())
    );

    lv_label_set_text(controller_column, text1.c_str());
    lv_label_set_text(connected_column, text2.c_str());
    lv_label_set_text(capacity_column, text3.c_str());
    lv_label_set_text(level_column, text4.c_str());
}



/**
 * changes parent of all objects
 */
void GeneralControllerDebug::GeneralControllerDebugInit(lv_obj_t *parent)
{
    //sets parent of container to pointer of new parent
    //this is to allow seperation of tabs into seperate classes
    //reduce the quantity in one class and to allow for ease of adding
    //new or different tabs

    lv_obj_set_parent(container, parent);

}




ControllerTab::ControllerTab(lv_obj_t *parent)
{
//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, CONTROLLER_CONTAINER_WIDTH);
    lv_obj_set_height(container, CONTROLLER_CONTAINER_HEIGHT);

//text for names
    std::string ctrl_col1 = (
        "Analog Left X\n"
        "Analog Left Y\n"
        "Analog Right X\n"
        "Analog Right Y\n"
        "Digital L1\n"
        "Digital L2\n"
        "Digital R1\n"
        "Digital R2"
    );

    std::string ctrl_col2 = (
        "Digital Up\n"
        "Digital Down\n"
        "Digital Left\n"
        "Digital Right\n"
        "Digital X\n"
        "Digital B\n"
        "Digital Y\n"
        "Digital A\n"
    );

//column one button names
    button_names_one = lv_label_create(container, NULL);
    lv_label_set_style(button_names_one, &toggle_tabbtn_pressed);
    lv_obj_set_width(button_names_one, CONTROLLER_CONTAINER_WIDTH / 4);
    lv_obj_set_height(button_names_one, 20);
    lv_label_set_align(button_names_one, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(button_names_one, ctrl_col1.c_str());


//column two button names
    button_names_two = lv_label_create(container, NULL);
    lv_label_set_style(button_names_two, &toggle_tabbtn_pressed);
    lv_obj_set_width(button_names_two, CONTROLLER_CONTAINER_WIDTH / 4);
    lv_obj_set_height(button_names_two, 20);
    lv_label_set_align(button_names_two, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(button_names_two, ctrl_col2.c_str());


//column one second part that contains function or values
    button_col_one = lv_label_create(container, NULL);
    lv_label_set_style(button_col_one, &toggle_tabbtn_pressed);
    lv_obj_set_width(button_col_one, CONTROLLER_CONTAINER_WIDTH / 4);
    lv_obj_set_height(button_col_one, 20);
    lv_label_set_align(button_col_one, LV_LABEL_ALIGN_LEFT);


//column two second part that contains function or values
    button_col_two = lv_label_create(container, NULL);
    lv_label_set_style(button_col_two, &toggle_tabbtn_pressed);
    lv_obj_set_width(button_col_two, CONTROLLER_CONTAINER_WIDTH / 4);
    lv_obj_set_height(button_col_two, 20);
    lv_label_set_align(button_col_two, LV_LABEL_ALIGN_LEFT);


//set positions relative to container
    lv_obj_align(button_names_one, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(button_col_one, container, LV_ALIGN_IN_TOP_MID, -80, 10);

    lv_obj_align(button_names_two, container, LV_ALIGN_IN_TOP_MID, 60, 10);
    lv_obj_align(button_col_two, container, LV_ALIGN_IN_TOP_RIGHT, -70, 10);
```

```

}


ControllerTab::~ControllerTab()
{
    lv_obj_del(button_names_one);
    lv_obj_del(button_names_two);
    lv_obj_del(button_col_one);
    lv_obj_del(button_col_two);

    lv_obj_del(container);
}



/**
 * updates the data for the controller tab for either the value of each button
 * or the function each button performs by looking at an std::unordered_map contained in
 * controller.hpp
 */
void ControllerTab::update(pros::controller_id_e_t controller, bool showing_values)
{
    std::string functions_col1 = "";
    std::string functions_col2 = "";

    std::string values_col1 = "";
    std::string values_col2 = "";

    if ( controller == pros::E_CONTROLLER_MASTER )
    {
        //text for functions
        functions_col1 = (
            Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_X) + "\n"
            + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_Y) + "\n"
            + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_X) + "\n"
            + Controller::MASTER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_Y) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L1) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L2) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R2) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R1)
        );

        functions_col2 = (
            Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_UP) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_DOWN) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_LEFT) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_RIGHT) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_X) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_B) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_Y) + "\n"
            + Controller::MASTER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_A)
        );

        values_col1 = (
            std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_X)) + "\n"
            + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) + "\n"
            + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X)) + "\n"
            + std::to_string(Controller::master.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_R1))
        );

        values_col2 = (
            std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_UP)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_LEFT)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_RIGHT)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_X)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_B)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_Y)) + "\n"
            + std::to_string(Controller::master.get_digital(pros::E_CONTROLLER_DIGITAL_A))
        );


    }

    else if ( controller == pros::E_CONTROLLER_PARTNER )
    {
        //text for functions
        functions_col1 = (
            Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_X) + "\n"
            + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_LEFT_Y) + "\n"
            + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_X) + "\n"
            + Controller::PARTNER_CONTROLLER_ANALOG_MAPPINGS.at(pros::E_CONTROLLER_ANALOG_RIGHT_Y) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L1) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_L2) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R2) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_R1)
        );

        functions_col2 = (
            Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_UP) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_DOWN) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_LEFT) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_RIGHT) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_X) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_B) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_Y) + "\n"
            + Controller::PARTNER_CONTROLLER_DIGITAL_MAPPINGS.at(pros::E_CONTROLLER_DIGITAL_A)
        );


        values_col1 = (
            std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_X)) + "\n"
            + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_LEFT_Y)) + "\n"
            + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_X)) + "\n"
            + std::to_string(Controller::partner.get_analog(pros::E_CONTROLLER_ANALOG_RIGHT_Y)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_L1)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_L2)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_R2)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_R1))
        );

        values_col2 = (
            std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_UP)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_DOWN)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_LEFT)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_RIGHT)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_X)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_B)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_Y)) + "\n"
            + std::to_string(Controller::partner.get_digital(pros::E_CONTROLLER_DIGITAL_A))
        );
    }

    if ( showing_values )
    {
        lv_label_set_text(button_col_one, values_col1.c_str());
        lv_label_set_text(button_col_two, values_col2.c_str());
    }

    else
    {
        lv_label_set_text(button_col_one, functions_col1.c_str());
        lv_label_set_text(button_col_two, functions_col2.c_str());
    }

}






ControllerDebug::ControllerDebug()
{
//reset statics
    cont = true;
    showing_values = 0;


//init screen
    controller_debug_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(controller_debug_screen, &gray);

//init title label
    title_label = lv_label_create(controller_debug_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, CONTROLLER_CONTAINER_WIDTH);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Controller - Debug");

//init tabview
    tabview = lv_tabview_create(controller_debug_screen, NULL);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
    //lv_tabview_set_tab_load_action(tabview,  tab_load_action);
    lv_obj_set_width(tabview, CONTROLLER_CONTAINER_WIDTH);
    lv_obj_set_height(tabview, 200);

//init tabs
    general_tab = lv_tabview_add_tab(tabview, "General");
    master_tab = lv_tabview_add_tab(tabview, "Master Controller");
    partner_tab = lv_tabview_add_tab(tabview, "Partner Controller");


//init back button
    //button
    btn_back = lv_btn_create(controller_debug_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");


//init button to switch between showing values and functions
    //button
    btn_show_values = lv_btn_create(controller_debug_screen, NULL);
    lv_btn_set_style(btn_show_values, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_show_values, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_show_values, LV_BTN_ACTION_CLICK, btn_show_values_action);
    lv_obj_set_width(btn_show_values, 150);
    lv_obj_set_height(btn_show_values, 25);

    //label
    btn_show_values_label = lv_label_create(btn_show_values, NULL);
    lv_obj_set_style(btn_show_values_label, &heading_text);
    lv_label_set_text(btn_show_values_label, "Show Values");

    lv_obj_set_hidden(btn_show_values, true);  //set hidden because button is
                            //not needed on the default tab


//init tabs from other classes
    GeneralControllerDebugInit(general_tab);


//set positions
    lv_obj_set_pos(btn_back, 30, 210);
    lv_obj_align(btn_show_values, controller_debug_screen, LV_ALIGN_IN_BOTTOM_MID, 0, -5);

    lv_obj_set_pos(title_label, 180, 5);

    lv_obj_set_pos(tabview, 20, 25);
```

```cpp
}




ControllerDebug::~ControllerDebug()
{
    lv_obj_del(controller_debug_screen);
}




/**
 * switches between showing values or function by setting variable to the
 * opposite of itself and then it updates the text label based on the new value
 */

lv_res_t ControllerDebug::btn_show_values_action(lv_obj_t *btn)
{
    showing_values = !showing_values;
    if ( showing_values )
    {
        lv_label_set_text(btn_show_values_label, "Show Functions");
    }
    else
    {
        lv_label_set_text(btn_show_values_label, "Show Values");
    }

    return LV_RES_OK;
}




/**
 * callback funciton that exits main loop when button is pressed
 */
lv_res_t ControllerDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}




/**
 * main loop that updates controller information
 */
void ControllerDebug::debug()
{
    //used to check if user wants to continue cycling through
    //tabs. Will be set to zero and loop will break if user hits
    //the back button
    cont = true;

    lv_tabview_set_tab_act(tabview, 0, NULL);
    lv_scr_load(controller_debug_screen);

    //init tabs from other classes
    ControllerTab controller_tab(master_tab);
    ControllerTab controller_tab2(partner_tab);

    while ( cont )
    {
        switch ( lv_tabview_get_tab_act(tabview) ) //switches to tab user wants to go to
        {
            case 0:
                lv_obj_set_hidden(btn_show_values, true);
                update_general_info();
                break;
            case 1:
                lv_obj_set_hidden(btn_show_values, false);
                controller_tab.update(pros::E_CONTROLLER_MASTER, showing_values);
                break;
            case 2:
                lv_obj_set_hidden(btn_show_values, false);
                controller_tab2.update(pros::E_CONTROLLER_PARTNER, showing_values);
                break;
        }
        pros::delay(200);
    }

}
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/FieldControlDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * contains class with methods that allow the user to see info about the state
 * od the field control
 */

#ifndef __FIELDCONTROLDEBUG_HPP__
#define __FIELDCONTROLDEBUG_HPP__


#include "../../../../include/main.h"

#include "../Styles.hpp"


/**
 * @see: ../Styles.hpp
 *
 * contains methods that show user data about the field control
 */
class FieldControlDebug : private Styles
{
    private:
        lv_obj_t *field_ctrl_screen;
        lv_obj_t *title_label;

        lv_obj_t *labels_label;
        lv_obj_t *info_label;

        //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);

    public:
        static bool cont;

        FieldControlDebug();
        ~FieldControlDebug();


        /**
         * @return: None
         * TODO: use ternary operator to condense and make more readable
         *
         * allows user to see information about the state of field control
         */
        void debug();

};




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/FieldControlDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: FieldControlDebug.hpp
 *
 * contains implementation for class with field control data
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "FieldControlDebug.hpp"


bool FieldControlDebug::cont = true;

FieldControlDebug::FieldControlDebug()
{
    cont = true;

//screen
    field_ctrl_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(field_ctrl_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(field_ctrl_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Field Control - Debug");

//init headers label
    labels_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(labels_label, &subheading_text);
    lv_obj_set_width(labels_label, 220);
    lv_obj_set_height(labels_label, 200);
    lv_label_set_align(labels_label, LV_LABEL_ALIGN_LEFT);

    std::string labels_label_text = (
            "connected to competition switch\n"
            "disabled\n"
            "game state"
    );

    lv_label_set_text(labels_label, labels_label_text.c_str());

//init values label
    info_label = lv_label_create(field_ctrl_screen, NULL);
    lv_label_set_style(info_label, &subheading_text);
    lv_obj_set_width(info_label, 220);
    lv_obj_set_height(info_label, 200);
    lv_label_set_align(info_label, LV_LABEL_ALIGN_LEFT);

    std::string info_label_text = (
        "no\n"
        "no\n"
        "no"
    );

    lv_label_set_text(info_label, info_label_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_align(title_label, field_ctrl_screen, LV_ALIGN_IN_TOP_MID, 0, 10);

    lv_obj_set_pos(labels_label, 20, 40);
    lv_obj_set_pos(info_label, 360, 40);


}

FieldControlDebug::~FieldControlDebug()
{
    lv_obj_del(field_ctrl_screen);
}


/**
 * sets cont to false to break main loop so main function returns
 */
lv_res_t FieldControlDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}


/**
 * main loop that updates the field control information
 */
void FieldControlDebug::debug()
{
    cont = true;

    lv_scr_load(field_ctrl_screen);

    while ( cont )
    {
        std::string info_label_text = "";

        if ( pros::competition::is_connected() )
        {
            info_label_text = info_label_text + "yes\n";
        }
        else
        {
            info_label_text = info_label_text + "no\n";
        }

        if ( pros::competition::is_disabled() )
        {
            info_label_text = info_label_text + "yes\n";
        }

        else
        {
            info_label_text = info_label_text + "no\n";
        }

        if ( pros::competition::is_autonomous() )
        {
            info_label_text = info_label_text + "autonomous\n";
        }

        else
        {
            info_label_text = info_label_text + "driver control\n";
        }

        lv_label_set_text(info_label, info_label_text.c_str());


        pros::delay(100);
    }
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/InternalMotorDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * contains class that for debugging the internal Motor PID constants
 * from the lcd without having to recompile and upload code
 */

#ifndef __INTERNALMOTORDEBUG_HPP__
#define __INTERNALMOTORDEBUG_HPP__

#include "main.h"

#include "../Styles.hpp"
#include "../../../Configuration.hpp"
#include "../../motors/Motor.hpp"


/**
 * @see: ../../motors/Motor.hpp
 *
 * allows user to tune pid constants by logging data and running unit tests
 * with given parameters
 */
class InternalMotorDebug : private Styles
{
    private:
        static bool cont;
        static bool run;

        Motor motor;

        lv_obj_t *main_screen;
        lv_obj_t *title_label;

    //parameter labels side one
        lv_obj_t *kp_label;
        lv_obj_t *kp_text_area;

        lv_obj_t *ki_label;
        lv_obj_t *ki_text_area;

        lv_obj_t *kd_label;
        lv_obj_t *kd_text_area;

        lv_obj_t *I_max_label;
        lv_obj_t *I_max_text_area;

        lv_obj_t *slew_label;
        lv_obj_t *slew_text_area;

        lv_obj_t *setpoint_label;
        lv_obj_t *setpoint_text_area;

        lv_obj_t *duration_label;
        lv_obj_t *duration_text_area;

    //parameter labels side two
        //port
        lv_obj_t *port_label;
        lv_obj_t *port_text_area;

        //gearset
        static pros::motor_gearset_e_t current_gearset;
        lv_obj_t *gearset_label;
        lv_obj_t *ddlist_gearset;

        /**
         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * sets the brake mode for the motor set which will be updated in the main loop
         */
        static lv_res_t ddlist_gearset_action(lv_obj_t *ddlist);

        //brake mode
        static pros::motor_brake_mode_e_t current_brake_mode;
        lv_obj_t *brakemode_label;
        lv_obj_t *ddlist_brake_mode;

        /**
         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * sets the brake mode for the motor set which will be updated in the main loop
         */
        static lv_res_t ddlist_brake_mode_action(lv_obj_t *ddlist);

    //information label
        lv_obj_t *information_label;

    //keyboard

        lv_obj_t *keyboard;


    //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


    //run unit test button
```

```
        lv_obj_t *btn_run;
        lv_obj_t *btn_run_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button to run unit_test and log data with the given parameters
         */
        static lv_res_t btn_run_action(lv_obj_t *btn);

    //actual unit test function
        /**
         * @return: int -> 1 if motor was successfully tested, 0 otherwise
         *
         * reads values from text areas and performs a unit test with the
         * given parameters
         */
        int run_unit_test( );


    public:
        InternalMotorDebug();
        ~InternalMotorDebug();

        /**
         * @return: None
         *
         * start debugger
         */
        void debug();

};



#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/InternalMotorDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * @see: InternalMotorDebug.hpp
 *
 * contains class implementation for tuning the motors internal velocity PID
 * controller
 */

#include <stdexcept>
#include <string>

#include "main.h"

#include "InternalMotorDebug.hpp"
#include "../Styles.hpp"
#include "../../../Configuration.hpp"
#include "../../motors/Motor.hpp"
#include "../../motors/MotorThread.hpp"
#include "../../serial/Logger.hpp"



bool InternalMotorDebug::cont = true;
bool InternalMotorDebug::run = false;
pros::motor_gearset_e_t InternalMotorDebug::current_gearset = pros::E_MOTOR_GEARSET_18;
pros::motor_brake_mode_e_t InternalMotorDebug::current_brake_mode = pros::E_MOTOR_BRAKE_COAST;


InternalMotorDebug::InternalMotorDebug() :
    motor(1, pros::E_MOTOR_GEARSET_18, false)
{
    MotorThread* motor_thread = MotorThread::get_instance();
    motor_thread->register_motor(motor);


    cont = true;
    run = false;

//screen
    main_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(main_screen, &gray);

//init title label
    title_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Velocity PID Controller - Debugger");


    //init parameters side two
    //port
    port_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(port_label, &heading_text);
    lv_obj_set_width(port_label, 100);
    lv_obj_set_height(port_label, 40);
    lv_label_set_align(port_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(port_label, "Port");

    port_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(port_text_area, &subheading_text);
    lv_ta_set_accepted_chars(port_text_area, "0123456789");
    lv_obj_set_size(port_text_area, 80, 20);
    lv_ta_set_text(port_text_area, "1");
    lv_ta_set_one_line(port_text_area, true);

    //gearset
    current_gearset = pros::E_MOTOR_GEARSET_18;

    gearset_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(gearset_label, &heading_text);
    lv_obj_set_width(gearset_label, 100);
    lv_obj_set_height(gearset_label, 40);
    lv_label_set_align(gearset_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(gearset_label, "Gearset");

    ddlist_gearset = lv_ddlist_create(main_screen, NULL);
    lv_ddlist_set_options(ddlist_gearset, "100\n200\n600");
    lv_obj_set_style(ddlist_gearset, &subheading_text);
    lv_obj_set_width(ddlist_gearset, 125);
    lv_obj_set_height(ddlist_gearset, 60);
    lv_ddlist_set_action(ddlist_gearset, ddlist_gearset_action);
    lv_ddlist_set_selected(ddlist_gearset, 1);

    //brakemode
    current_brake_mode = pros::E_MOTOR_BRAKE_COAST;

    brakemode_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(brakemode_label, &heading_text);
    lv_obj_set_width(brakemode_label, 100);
    lv_obj_set_height(brakemode_label, 40);
    lv_label_set_align(brakemode_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(brakemode_label, "Brakemode");

    ddlist_brake_mode = lv_ddlist_create(main_screen, NULL);
    lv_ddlist_set_options(ddlist_brake_mode, "Coast\n"
                          "Brake\n"
                          "PID Hold");
    lv_obj_set_style(ddlist_brake_mode, &subheading_text);
    lv_obj_set_width(ddlist_brake_mode, 125);
    lv_obj_set_height(ddlist_brake_mode, 20);
    lv_ddlist_set_action(ddlist_brake_mode, ddlist_brake_mode_action);


//init parameters side one
//kP
    kp_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(kp_label, &heading_text);
    lv_obj_set_width(kp_label, 440);
    lv_obj_set_height(kp_label, 20);
    lv_label_set_align(kp_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(kp_label, "kP");

    kp_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(kp_text_area, &subheading_text);
    lv_ta_set_accepted_chars(kp_text_area, ".0123456789");
    lv_obj_set_size(kp_text_area, 80, 15);
    lv_ta_set_text(kp_text_area, std::to_string(motor.get_pid().kP).c_str());
    lv_ta_set_one_line(kp_text_area, true);

//kI
    ki_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(ki_label, &heading_text);
    lv_obj_set_width(ki_label, 100);
    lv_obj_set_height(ki_label, 20);
    lv_label_set_align(ki_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(ki_label, "kI");

    ki_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(ki_text_area, &subheading_text);
    lv_ta_set_accepted_chars(ki_text_area, ".0123456789");
    lv_obj_set_size(ki_text_area, 80, 15);
    lv_ta_set_text(ki_text_area, std::to_string(motor.get_pid().kI).c_str());
    lv_ta_set_one_line(ki_text_area, true);

//kD
    kd_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(kd_label, &heading_text);
    lv_obj_set_width(kd_label, 100);
    lv_obj_set_height(kd_label, 20);
    lv_label_set_align(kd_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(kd_label, "kD");

    kd_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(kd_text_area, &subheading_text);
    lv_ta_set_accepted_chars(kd_text_area, ".0123456789");
    lv_obj_set_size(kd_text_area, 80, 15);
    lv_ta_set_text(kd_text_area, std::to_string(motor.get_pid().kD).c_str());
    lv_ta_set_one_line(kd_text_area, true);

//I max
    I_max_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(I_max_label, &heading_text);
    lv_obj_set_width(I_max_label, 100);
    lv_obj_set_height(I_max_label, 20);
    lv_label_set_align(I_max_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(I_max_label, "kI Max");

    I_max_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(I_max_text_area, &subheading_text);
    lv_ta_set_accepted_chars(I_max_text_area, ".0123456789");
    lv_obj_set_size(I_max_text_area, 80, 15);
    lv_ta_set_text(I_max_text_area, std::to_string(motor.get_pid().I_max).c_str());
    lv_ta_set_one_line(I_max_text_area, true);

//slew rate
    slew_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(slew_label, &heading_text);
    lv_obj_set_width(slew_label, 100);
    lv_obj_set_height(slew_label, 40);
    lv_label_set_align(slew_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(slew_label, "Slew Rate");

    slew_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(slew_text_area, &subheading_text);
    lv_ta_set_accepted_chars(slew_text_area, "0123456789");
    lv_obj_set_size(slew_text_area, 80, 15);
    lv_ta_set_text(slew_text_area, std::to_string(motor.get_slew_rate()).c_str());
    lv_ta_set_one_line(slew_text_area, true);

//setpoint_label
    setpoint_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(setpoint_label, &heading_text);
    lv_obj_set_width(setpoint_label, 100);
    lv_obj_set_height(setpoint_label, 40);
    lv_label_set_align(setpoint_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(setpoint_label, "Setpoint");

    setpoint_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(setpoint_text_area, &subheading_text);
    lv_ta_set_accepted_chars(setpoint_text_area, "0123456789");
    lv_obj_set_size(setpoint_text_area, 80, 15);
    lv_ta_set_text(setpoint_text_area, "200");
    lv_ta_set_one_line(setpoint_text_area, true);

//duration
    duration_label = lv_label_create(main_screen, NULL);
    lv_label_set_style(duration_label, &heading_text);
    lv_obj_set_width(duration_label, 100);
    lv_obj_set_height(duration_label, 40);
    lv_label_set_align(duration_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(duration_label, "Duration");

    duration_text_area = lv_ta_create(main_screen, NULL);
    lv_obj_set_style(duration_text_area, &subheading_text);
    lv_ta_set_accepted_chars(duration_text_area, "0123456789");
    lv_obj_set_size(duration_text_area, 80, 15);
    lv_ta_set_text(duration_text_area, "10000");
    lv_ta_set_one_line(duration_text_area, true);



//information label
    information_label = lv_label_create(main_screen, NULL);
    lv_obj_set_style(information_label, &subheading_text);
    lv_label_set_text(information_label, "Info");


//init back button
    //button
    btn_back = lv_btn_create(main_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");


//init run button
    //button
    btn_run = lv_btn_create(main_screen, NULL);
    lv_btn_set_style(btn_run, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_run, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_run, LV_BTN_ACTION_CLICK, btn_run_action);
    lv_obj_set_width(btn_run, 150);
    lv_obj_set_height(btn_run, 25);

    //label
    btn_run_label = lv_label_create(btn_run, NULL);
    lv_obj_set_style(btn_run_label, &heading_text);
    lv_label_set_text(btn_run_label, "Run Unit Test");


//set up keyboard
    keyboard = lv_kb_create(main_screen, NULL);
    // lv_kb_set_ta(keyboard, kp_text_area);
    // lv_kb_set_ta(keyboard, ki_text_area);
    // lv_kb_set_ta(keyboard, kd_text_area);
    // lv_kb_set_ta(keyboard, I_max_text_area);
    // lv_kb_set_ta(keyboard, slew_text_area);

    //lv_ta_set_action(port_text_area, LV_EVENT_PRESSED);

//set positions
//title
    lv_obj_set_pos(title_label, 100, 5);

//bottom buttons
    lv_obj_set_pos(btn_back, 30, 210);
    lv_obj_set_pos(btn_run, 300, 210);

//parameters side 1
    lv_obj_set_pos(kp_label, 20, 30);
    lv_obj_set_pos(ki_label, 20, 55);
    lv_obj_set_pos(kd_label, 20, 80);
    lv_obj_set_pos(I_max_label, 20, 105);
    lv_obj_set_pos(slew_label, 20, 130);
    lv_obj_set_pos(setpoint_label, 20, 155);
    lv_obj_set_pos(duration_label, 20, 180);

    lv_obj_set_pos(kp_text_area, 130, 23);
    lv_obj_set_pos(ki_text_area, 130, 48);
    lv_obj_set_pos(kd_text_area, 130, 73);
    lv_obj_set_pos(I_max_text_area, 130, 98);
    lv_obj_set_pos(slew_text_area, 130, 123);
    lv_obj_set_pos(setpoint_text_area, 130, 148);
    lv_obj_set_pos(duration_text_area, 130, 173);

//parameters side 2
    lv_obj_set_pos(port_label, 240, 40);
    lv_obj_set_pos(gearset_label, 240, 75);
    lv_obj_set_pos(brakemode_label, 240, 100);

    lv_obj_set_pos(port_text_area, 350, 33);
    lv_obj_set_pos(ddlist_gearset, 350, 75);
    lv_obj_set_pos(ddlist_brake_mode, 350, 100);

//information
    lv_obj_set_pos(information_label, 240, 140);



}

InternalMotorDebug::~InternalMotorDebug()
{
    MotorThread* motor_thread = MotorThread::get_instance();
    motor_thread->unregister_motor(motor);

    lv_obj_del(main_screen);
}


/**
 * sets cont to false signifying user wants to go back, main loop will exit
 */
lv_res_t InternalMotorDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}

lv_res_t InternalMotorDebug::btn_run_action(lv_obj_t *btn)
{
    lv_btn_set_state(btn, LV_BTN_STATE_INA);
    run = true;
    return LV_RES_OK;
}



/**
 * looks at the string of the current drop down list option and compares it to
 * a string to see what gearset the user wants
 * sets gearset to this value
 */
lv_res_t InternalMotorDebug::ddlist_gearset_action(lv_obj_t * ddlist)
{
    //checks what the drop down list string is
    char sel_cstr[32];
    lv_ddlist_get_selected_str(ddlist, sel_cstr);

    std::string sel_str = std::string(sel_cstr);  //convert to std::string so
                                            //that the strings can be
                                            //compared

    //sets brake mode for motor
    if ( sel_str == "100" )
    {
        current_gearset = pros::E_MOTOR_GEARSET_36;
    }

    else if ( sel_str == "600" )
    {
        current_gearset = pros::E_MOTOR_GEARSET_06;
    }

    else
    {
        current_gearset = pros::E_MOTOR_GEARSET_18;
    }

    return LV_RES_OK; //Return OK because the drop down list was not deleted
}




/**
 * looks at the string of the current drop down list option and compares it to
 * a string to see what brakemode the user wants
 * sets brake mode to this value
 */
lv_res_t InternalMotorDebug::ddlist_brake_mode_action(lv_obj_t * ddlist)
{
    //checks what the drop down list string is
    char sel_cstr[32];
    lv_ddlist_get_selected_str(ddlist, sel_cstr);

    std::string sel_str = std::string(sel_cstr);  //convert to std::string so
                                            //that the strings can be
                                            //compared

    //sets brake mode for motor
    if ( sel_str == "PID Hold" )
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_HOLD;
    }

    else if ( sel_str == "Brake" )
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_BRAKE;
    }

    else
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
    }

    return LV_RES_OK; //Return OK if the drop down list is not deleted
}




/**
 * reads values from text areas and performs data validation, exits on invalid data
 * starts unit test and logs data
 * updates labels on lcd while waiting for duration to finish
 */
int InternalMotorDebug::run_unit_test()
{
    Logger logger;

    pid pid_constants;

    int slew = 0;
    int setpoint = 0;
    int motor_port = 0;

    int duration = 0;

    //read info from text areas in exception safe way
    try
    {
        double kP = std::stod(lv_ta_get_text(kp_text_area));
        double kI = std::stod(lv_ta_get_text(ki_text_area));
        double kD = std::stod(lv_ta_get_text(kd_text_area));
        double I_max = std::stod(lv_ta_get_text(I_max_text_area));

        pid_constants.kP = kP;
        pid_constants.kI = kI;
        pid_constants.kD = kD;
        pid_constants.I_max = I_max;
    }
    catch ( const std::invalid_argument& )
    {
        run = false;

        log_entry entry;
        entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid pid constants given to internal motor unit test";
        entry.stream = "cerr";
        logger.add(entry);

        return 0;
    }

    try
    {
        slew = std::stoi(lv_ta_get_text(slew_text_area));
    }
    catch ( const std::invalid_argument& )
```

```
{
    run = false;

    log_entry entry;
    entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid slew rate given to internal motor unit test";
    entry.stream = "cerr";
    logger.add(entry);

    return 0;
}

try
{
    setpoint = std::stoi(lv_ta_get_text(setpoint_text_area));
}
catch ( const std::invalid_argument& )
{
    run = false;

    log_entry entry;
    entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid setpoint given to internal motor unit test";
    entry.stream = "cerr";
    logger.add(entry);

    return 0;
}

try
{
    motor_port = std::stoi(lv_ta_get_text(port_text_area));
}
catch ( const std::invalid_argument& )
{
    run = false;

    log_entry entry;
    entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid motor port given to internal motor unit test";
    entry.stream = "cerr";
    logger.add(entry);

    return 0;
}

try
{
    duration = std::stoi(lv_ta_get_text(duration_text_area));
}
catch ( const std::invalid_argument& )
{
    run = false;

    log_entry entry;
    entry.content = "[ERROR] " + std::to_string(pros::millis()) + " invalid duration given to internal motor unit test";
    entry.stream = "cerr";
    logger.add(entry);

    return 0;
}

//set motor information
motor.set_port(motor_port);
motor.set_gearing(current_gearset);
motor.set_brake_mode(current_brake_mode);
// if ( std::abs(slew) > 0 )
// {
// motor.enable_slew();
// motor.set_slew(30);
// }
// else
// {
     motor.disable_slew();
// }

//motor.disable_velocity_pid();
motor.disable_driver_control();
motor.set_pid( pid_constants );
motor.set_motor_mode(e_custom_velocity_pid);
motor.set_log_level(1);
std::cout << motor.get_pid().kP << "\n";

MotorThread * motor_thread = MotorThread::get_instance();
motor_thread->start_thread();

int ut_end_time = pros::millis() + duration;
//motor.move_velocity(setpoint);
motor.move_velocity(200);

//wait for unit test to finish and update gui in the meantime
while ( pros::millis() < ut_end_time )
{
    //update gui
    std::string info_str;
    info_str = "Voltage: " + std::to_string(motor.get_actual_voltage()) + "\n";
    info_str += "Velocity: " + std::to_string(motor.get_actual_velocity()) + "\n";
    info_str += "Error: " + std::to_string(setpoint - motor.get_actual_velocity());

    lv_label_set_text(information_label, info_str.c_str());
    logger.dump();
    pros::delay(50);
}

motor.set_voltage(0);
pros::delay(2000);
logger.dump();
logger.dump();
logger.dump();
logger.dump();
logger.dump();
logger.dump();
logger.dump();
logger.dump();
logger.dump();
motor.set_log_level(0);
```

```

    return 1;
}




/**
 * waits for cont to be false which occurs when the user hits the back button
 */
void InternalMotorDebug::debug()
{
    std::string error_str = "-";
    cont = true;
    run = false;


    lv_scr_load(main_screen);

    while ( cont )
    {
        //update information label
        std::string info_str;
        info_str = "Voltage: " + std::to_string(motor.get_actual_voltage()) + "\n";
        info_str += "Velocity: " + std::to_string(motor.get_actual_velocity()) + "\n";
        info_str += "Error: -";
        lv_label_set_text(information_label, info_str.c_str());

        if ( run )
        {
            run_unit_test();
            run = false;
            lv_btn_set_state(btn_run, LV_BTN_STYLE_REL);

        }


        pros::delay(100);
    }

}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/MotorsDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * contains class that loads tabs to debug motors
 */

#ifndef __MOTORDEBUG_HPP__
#define __MOTORDEBUG_HPP__

#include <vector>

#include "../../../../include/main.h"

#include "../Styles.hpp"
#include "../../motors/Motors.hpp"
#include "../../motors/Motor.hpp"

//user defines

//sets size of container
#define MOTORS_CONTAINER_WIDTH 440
#define MOTORS_CONTAINER_HEIGHT 100

//sets percent at which to step velocity at
//10 is reasonable because anything higher gives
//less control and anything lower will make it
//difficult to ramp up or down
#define STEP_PERCENT 10



/**
 * @see: ../Styles.hpp
 *
 * general tab for one or two motors max
 * contains methods to show data and set velocity of motors
 * on this tab
 */
class MotorsDebugTab : virtual Styles
{
    private:
        lv_obj_t *container;
        lv_obj_t *motor1_label;
        lv_obj_t *motor2_label;

        lv_obj_t *motor1_info;
        lv_obj_t *motor2_info;

        std::vector<Motor*> motors;
        std::vector<std::string> titles;

    public:
        MotorsDebugTab( std::vector<Motor*> motors_vec, std::vector<std::string> titles_vec, lv_obj_t *parent);
        ˜MotorsDebugTab();

        /**
         * @param: int target_velocity -> velocity the motor should be set to
         * @param: lv_obj_t* velocity_label -> label that current veolicty will be written to
         * @return: None
         *
         * @see: ../Styles.hpp
         * @see: ../../motors/Motors.hpp
         *
         * updates text for the motors that the class was instatiated with
         * also sets the velocity of the motor to int target_velocity
         * data shown is current drawn, voltage, reversed or not, temperature, encoder value,
         * and tourque
         */
        void update_label(int target_velocity, lv_obj_t *velocity_label);
};




/**
 * @see: class MotorsDebugTab
 * @see: ../Styles.hpp
 *
 * contatins debugger for motors
 * gives data for each motor set ie. left chassis, right chassis, intake, etc.
 */
class MotorsDebug : virtual Styles
{
    private:
        //screen
        lv_obj_t *motor_debug_screen;

        //title label
        lv_obj_t *title_label;

        //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


        static lv_obj_t *tabview;   //tabview object

        lv_obj_t *l_chassis_tab;    //individual tabs
        lv_obj_t *r_chassis_tab;    //content will come from base classes
        lv_obj_t *main_intake_tab;
        lv_obj_t *front_intake_tab;

        static uint16_t tab_loaded; // 0 = left chassis
                                    // 1 = right chassis
                                    // 2 = main intake
                                    // 3 = front intake

        /**
         * @param: lv_obj_t* tabview -> tabview object for callback function
         * @param: uint16_t -> id of active tab
         * @return: lv_res_t -> return LV_RES_OK since object was not deleted
         *
         * funtion to stop motor movements and set the ability for other threads
         * to limit the speed of the motor ie. set it to zero in driver control
         * also updates target velocity and the tab loaded
         */
        static lv_res_t tab_load_action(lv_obj_t *tabview, uint16_t act_id);


        //velocity setting buttons
        lv_obj_t *velocity_label;

        lv_obj_t *btn_pos_increase;
        lv_obj_t *btn_neg_increase;
        lv_obj_t *btn_stp;

        lv_obj_t *btn_pos_increase_label;
        lv_obj_t *btn_neg_increase_label;
        lv_obj_t *btn_stp_label;


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * @see: std::tuple<int, int> get_velocity_step()
         *
         * button callback function used to decrease the target velocity
         */
        static lv_res_t btn_pos_increase_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * @see: std::tuple<int, int> get_velocity_step()
         *
         * button callback function used to increase the target velocity
         */
        static lv_res_t btn_neg_increase_action(lv_obj_t *btn);


        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * @see: std::tuple<int, int> get_velocity_step()
         *
         * button callback function used to set target velocity to zero
         */
        static lv_res_t btn_stp_action(lv_obj_t *btn);


        /**
         * @return: std::tuple<int, int> -> tuple of step, a percentage of max velocity
         *          based on STEP_PERCENT, and max velocity of the motor
         * TOOD: update max velocity for motors and make more adaptable to changing motors
         *
         * gets the amount the step should be and the max velocity for the motor
         * the max velocity is higher than actual because the motor can go faster
         * than the specified RPM
         */
        static std::tuple<int, int> get_velocity_step();


        //static vars to help keep velocity
        //need to be static because they will be modified by
        //static function
        static int target_velocity;


        //brake mode option widgets
        lv_obj_t *brake_mode_label;
        lv_obj_t *ddlist_brake_mode;


        /**
         * @param: lv_obj_t* ddlist -> the dropdown list object for the callback function
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * sets the brake mode for the motor set which will be updated in the main loop
         */
        static lv_res_t ddlist_brake_mode_action(lv_obj_t *ddlist);

        static pros::motor_brake_mode_e_t current_brake_mode;

    public:
        MotorsDebug();
        ~MotorsDebug();

        static bool cont; //checks whether to keep letting user
                          //cycle through tabs

        /**
         * @return: None
         *
         * allows user to interact with tabs for each motor set that display
         * data about those motors
         */
        void debug();

};

```

```

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/MotorsDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/16/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: MotorsDebug.hpp
 *
 * contains classes and methods implementation that allow the gui to show
 * the user information about groups of motors seperated into tabs
 */

#include <cstdint>
#include <cmath>
#include <vector>

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "../../motors/Motors.hpp"
#include "MotorsDebug.hpp"


//declare static members of all classes
bool MotorsDebug::cont = true;
int MotorsDebug::target_velocity = 0;
lv_obj_t *MotorsDebug::tabview;
pros::motor_brake_mode_e_t MotorsDebug::current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
uint16_t MotorsDebug::tab_loaded = 0;



MotorsDebugTab::MotorsDebugTab( std::vector<Motor*> motors_vec, std::vector<std::string> titles_vec, lv_obj_t *parent)
{
    for( int i = 0; i < motors_vec.size(); i++ )
    {
        motors.push_back(motors_vec.at(i));
        titles.push_back(titles_vec.at(i));
    }
//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, MOTORS_CONTAINER_WIDTH);
    lv_obj_set_height(container, MOTORS_CONTAINER_HEIGHT);

//init motor 1 label
    motor1_label = lv_label_create(container, NULL);
    lv_obj_set_style(motor1_label, &toggle_tabbtn_pressed);
    lv_obj_set_width(motor1_label, (MOTORS_CONTAINER_WIDTH/2));
    lv_obj_set_height(motor1_label, 20);
    lv_label_set_align(motor1_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(motor1_label, titles.at(0).c_str());

//init motor 1 info label
    motor1_info = lv_label_create(container, NULL);
    lv_obj_set_style(motor1_info, &toggle_tabbtn_pressed);
    lv_obj_set_width(motor1_info, (MOTORS_CONTAINER_WIDTH/2));
    lv_obj_set_height(motor1_info, 20);
    lv_label_set_align(motor1_info, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(motor1_info, "None\nNone\nNone\nNone\nNone\nNone");

//init motor 2 label
    motor2_label = lv_label_create(container, NULL);
    lv_obj_set_style(motor2_label, &toggle_tabbtn_pressed);
    lv_obj_set_width(motor2_label, (MOTORS_CONTAINER_WIDTH/2));
    lv_obj_set_height(motor2_label, 20);
    lv_label_set_align(motor2_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(motor2_label, "");

    motor2_info = lv_label_create(container, NULL);
    lv_obj_set_style(motor2_info, &toggle_tabbtn_pressed);
    lv_obj_set_width(motor2_info, (MOTORS_CONTAINER_WIDTH/2));
    lv_obj_set_height(motor2_info, 20);
    lv_label_set_align(motor2_info, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(motor2_info, "None");



    if( motors.size() > 1 )
    {
        lv_label_set_text(motor2_label, titles.at(1).c_str());
    }

//align objects on container
    lv_obj_set_pos(motor1_label, 60, 0);
    lv_obj_set_pos(motor1_info, 10, 15);
    if( motors.size() > 1 )
    {
        lv_obj_set_pos(motor2_label, 315, 0);
        lv_obj_set_pos(motor2_info, 255, 15);
    }
}

MotorsDebugTab::~MotorsDebugTab()
{
    lv_obj_del(motor1_label);
    lv_obj_del(motor1_info);
    lv_obj_del(motor2_label);
    lv_obj_del(motor2_info);

    lv_obj_del(container);
}



/**
 * function to be called in main loop so that data about motors will be updated
 * sets velocity, updates data, and updates velocity label
 */
void MotorsDebugTab::update_label(int target_velocity, lv_obj_t *velocity_label)
```

```
{
    std::string info1 = "";
    std::string info2 = "";

//set velocity to move at
    std::int32_t vel = target_velocity;
    motors.at(0)->move_velocity(vel);
    if ( motors.size() > 1 )
    {
        motors.at(1)->move_velocity(vel);
    }

//info for first motor
    info1 += "Current Draw: " + std::to_string(motors.at(0)->get_current_draw()) + "\n";
    info1 += "Voltage (mV): " + std::to_string(motors.at(0)->get_actual_voltage()) + "\n";
    info1 += "State: ";
    info1 += motors.at(0)->is_reversed() ? "reversed\n" : "not reversed\n";
    info1 += "Temperature: " + std::to_string(motors.at(0)->get_temperature()) + "\n";
    info1 += "Encoder Position: " + std::to_string(motors.at(0)->get_encoder_position()) + "\n";
    info1 += "Torque (Nm): " + std::to_string(motors.at(0)->get_torque()) + "\n";

//info for second motor if it exists
    if ( motors.size() > 1 )
    {
        info2 += "Current Draw: " + std::to_string(motors.at(1)->get_current_draw()) + "\n";
        info2 += "Voltage (mV): " + std::to_string(motors.at(1)->get_actual_voltage()) + "\n";
        info2 += "State: ";
        info2 += motors.at(1)->is_reversed() ? "reversed\n" : "not reversed\n";
        info2 += "Temperature: " + std::to_string(motors.at(1)->get_temperature()) + "\n";
        info2 += "Encoder Position: " + std::to_string(motors.at(1)->get_encoder_position()) + "\n";
        info2 += "Torque (Nm): " + std::to_string(motors.at(1)->get_torque()) + "\n";
    }
//info for velocity label
    std::string velocity;
    velocity += titles.at(0) + ": " + std::to_string(motors.at(0)->get_actual_velocity()) + "\n";
    if ( motors.size() > 1 )
    {
        velocity += titles.at(1) + ": " + std::to_string(motors.at(1)->get_actual_velocity());
    }

//set labels
    //casts info strings to c strings to make them compatible with lvgl
    lv_label_set_text(motor1_info, info1.c_str());
    if ( motors.size() > 1 )
    {
        lv_label_set_text(motor2_info, info2.c_str());
    }
    lv_label_set_text(velocity_label, velocity.c_str());

}




MotorsDebug::MotorsDebug()
{
    //set default for statics
    cont = true;
    target_velocity = 0;
    tab_loaded = 0;

//init screen
    motor_debug_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(motor_debug_screen, &gray);

//init title label
    title_label = lv_label_create(motor_debug_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, MOTORS_CONTAINER_WIDTH);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Motors - Debug");

//init tabview
    tabview = lv_tabview_create(motor_debug_screen, NULL);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
    lv_tabview_set_tab_load_action(tabview, tab_load_action);
    lv_obj_set_width(tabview, MOTORS_CONTAINER_WIDTH);
    lv_obj_set_height(tabview, 200);

//init tabs
    l_chassis_tab = lv_tabview_add_tab(tabview, "Chassis (L)");
    r_chassis_tab = lv_tabview_add_tab(tabview, "Chassis (R)");
    main_intake_tab = lv_tabview_add_tab(tabview, "Main Intake");
    front_intake_tab = lv_tabview_add_tab(tabview, "Front Intakes");

//init back button
    //button
    btn_back = lv_btn_create(motor_debug_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init velocity label
    velocity_label = lv_label_create(motor_debug_screen, NULL);
    lv_obj_set_style(velocity_label, &subheading_text);
    lv_obj_set_width(velocity_label, 100);
    lv_obj_set_height(velocity_label, 40);
    lv_label_set_align(velocity_label, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(velocity_label, "Velocity: ");

//init velocity increase button
```

```
    //button
    btn_pos_increase =  lv_btn_create(motor_debug_screen, NULL);
    lv_btn_set_style(btn_pos_increase, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_pos_increase, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_pos_increase, LV_BTN_ACTION_CLICK, btn_pos_increase_action);
    lv_obj_set_width(btn_pos_increase, 40);
    lv_obj_set_height(btn_pos_increase, 25);

    //label
    btn_pos_increase_label = lv_label_create(btn_pos_increase, NULL);
    lv_obj_set_style(btn_pos_increase_label, &heading_text);
    lv_label_set_text(btn_pos_increase_label, SYMBOL_RIGHT);

//init velocity decrease button
    //button
    btn_neg_increase =  lv_btn_create(motor_debug_screen, NULL);
    lv_btn_set_style(btn_neg_increase, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_neg_increase, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_neg_increase, LV_BTN_ACTION_CLICK, btn_neg_increase_action);
    lv_obj_set_width(btn_neg_increase, 40);
    lv_obj_set_height(btn_neg_increase, 25);

    //label
    btn_neg_increase_label = lv_label_create(btn_neg_increase, NULL);
    lv_obj_set_style(btn_neg_increase_label, &heading_text);
    lv_label_set_text(btn_neg_increase_label, SYMBOL_LEFT);

//init zero velocity button
    //button
    btn_stp =  lv_btn_create(motor_debug_screen, NULL);
    lv_btn_set_style(btn_stp, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_stp, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_stp, LV_BTN_ACTION_CLICK, btn_stp_action);
    lv_obj_set_width(btn_stp, 40);
    lv_obj_set_height(btn_stp, 25);

    //label
    btn_stp_label = lv_label_create(btn_stp, NULL);
    lv_obj_set_style(btn_stp_label, &heading_text);
    lv_label_set_text(btn_stp_label, SYMBOL_STOP);

//init brake mode label
    brake_mode_label = lv_label_create(motor_debug_screen, NULL);
    lv_obj_set_style(brake_mode_label, &heading_text);
    lv_obj_set_width(brake_mode_label, 100);
    lv_obj_set_height(brake_mode_label, 20);
    lv_label_set_align(brake_mode_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(brake_mode_label, "Brakemode: ");

//init drop down list
    ddlist_brake_mode = lv_ddlist_create(motor_debug_screen, NULL);
    lv_ddlist_set_options(ddlist_brake_mode, "Coast\n"
                                    "Brake\n"
                                    "PID Hold");
    lv_obj_set_style(ddlist_brake_mode, &subheading_text);
    lv_obj_set_width(ddlist_brake_mode, 125);
    lv_obj_set_height(ddlist_brake_mode, 18);
    lv_ddlist_set_action(ddlist_brake_mode, ddlist_brake_mode_action);

//set positions
    lv_obj_set_pos(btn_back, 30, 210);
    lv_obj_set_pos(btn_pos_increase, 270, 210);
    lv_obj_set_pos(btn_stp, 220, 210);
    lv_obj_set_pos(btn_neg_increase, 170, 210);

    lv_obj_set_pos(velocity_label, 330, 177);

    lv_obj_set_pos(brake_mode_label, 60, 177);
    lv_obj_set_pos(ddlist_brake_mode, 170, 177);

    lv_obj_set_pos(title_label, 180, 5);

    lv_obj_set_pos(tabview, 20, 25);

}


MotorsDebug::~MotorsDebug()
{
    //sets motors to off
    Motors::stop_all_motors();

    //allow motor to go to zero for driver control if it is not set
    //already
    Motors::enable_driver_control();

    //deletes widgets instantiated by class
    lv_obj_del(title_label);

    lv_obj_del(btn_back_label);
    lv_obj_del(btn_back);

    lv_obj_del(l_chassis_tab);
    lv_obj_del(r_chassis_tab);
    lv_obj_del(main_intake_tab);
    lv_obj_del(front_intake_tab);
    lv_obj_del(tabview);

    lv_obj_del(velocity_label);

    lv_obj_del(btn_pos_increase_label);
    lv_obj_del(btn_neg_increase_label);
    lv_obj_del(btn_stp_label);
    lv_obj_del(btn_pos_increase);
    lv_obj_del(btn_neg_increase);
    lv_obj_del(btn_stp);

    lv_obj_del(brake_mode_label);
    lv_obj_del(ddlist_brake_mode);


    lv_obj_del(motor_debug_screen);
}
```

```


/**
 * set cont to false to break main loop
 */
lv_res_t MotorsDebug::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}


/**
 * callback function for when a new tab is selected
 * used to set motor to default ie. brakemode, velocity
 */
lv_res_t MotorsDebug::tab_load_action(lv_obj_t *tabview, uint16_t act_id)
{
    tab_loaded = act_id;
    target_velocity = 0;

    //sets motors to off
    Motors::stop_all_motors();

    //allow motor to go to zero for driver control if it is not set
    //already
    Motors::enable_driver_control();

    return LV_RES_OK;

}



/**
 * looks at the current tab loaded to decide on max velocity because the motor
 * can be determined from that
 * gets the step percent by looking at what STEP_PERCENT is defined as
 */
std::tuple<int, int> MotorsDebug::get_velocity_step()
{
    int index = tab_loaded;    // 0 = left chassis - 200RPM
                              // 1 = right chassis - 200RPM
                              // 2 = tilter - 100RPM
                              // 3 = intake - 100RPM
                              // 4 = lift - 100RPM
    int max;

    if ( index == 0 || index == 1 )
    {
        max = 250;
    }

    else if (index == 2 || index == 3 || index == 4)
    {
        max = 130;
    }

    else
    {
        max = 650;
    }

    int step = static_cast<int>(max / STEP_PERCENT);
    return std::make_tuple(step, max);
}


/**
 * increses velocity of motor by calling get_velocity_step but limits it to
 * the max velocity
 */
lv_res_t MotorsDebug::btn_pos_increase_action(lv_obj_t *btn)
{
    //increases velocity by user defined percent
    int step;
    int max;

    std::tie(step, max) = get_velocity_step();
    if ( target_velocity < max )
    {
        target_velocity = target_velocity + step;
    }

    return LV_RES_OK;
}


/**
 * decreases velocity of motor by calling get_velocity_step but limits it to
 * the max velocity in the negative direction
 */
lv_res_t MotorsDebug::btn_neg_increase_action(lv_obj_t *btn)
{
    //decreases velocity by user defined percent
    int step;
    int max;

    std::tie(step, max) = get_velocity_step();
    if ( target_velocity > 0-max )
    {
        target_velocity = target_velocity - step;
    }

    return LV_RES_OK;
}


/**
 * sets velocity of motor to zero, used so that user does not have to click
 * many times to stop the motor
 */
lv_res_t MotorsDebug::btn_stp_action(lv_obj_t *btn)
```

```
{
    target_velocity = 0;
    return LV_RES_OK;
}


/**
 * looks at the string of the current drop down list option and compares it to
 * a string to see what brakemode the user wants
 * sets brake mode to this value
 */
lv_res_t MotorsDebug::ddlist_brake_mode_action(lv_obj_t * ddlist)
{
    //checks what the drop down list string is
    char sel_cstr[32];
    lv_ddlist_get_selected_str(ddlist, sel_cstr);

    std::string sel_str = std::string(sel_cstr);  //convert to std::string so
                                                  //that the strings can be
                                                  //compared

    //sets brake mode for motor
    if ( sel_str == "PID Hold" )
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_HOLD;
    }

    else if ( sel_str == "Brake" )
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_BRAKE;
    }

    else
    {
        current_brake_mode = pros::E_MOTOR_BRAKE_COAST;
    }

    return LV_RES_OK; //Return OK if the drop down list is not deleted
}




/**
 * has a main loop that updates internal data as user cycles through tabs
 * to keep data relevent and motors following the function they are supposed to
 * loads tabs for each motor set
 */
void MotorsDebug::debug()
{
    //used to check if user wants to continue cycling through
    //tabs. Will be set to zero and loop will break if user hits
    //the back button
    cont = true;

    lv_tabview_set_tab_act(tabview, 0, NULL);
    lv_scr_load(motor_debug_screen);

    MotorsDebugTab l_chassis_tab_debug( {&Motors::front_left, &Motors::back_left}, {"Front Left", "Back Left"}, l_chassis_tab );
    MotorsDebugTab r_chassis_tab_debug( {&Motors::front_right, &Motors::back_right}, {"Front Right", "Back Right"}, r_chassis_tab );
    MotorsDebugTab main_intake_tab_debug( {&Motors::upper_indexer, &Motors::lower_indexer}, {"upper_indexer", "lower_indexer"}, main_intake_tab );
    MotorsDebugTab front_intake_tab_debug( {&Motors::left_intake, &Motors::right_intake}, {"Left Intake", "Right Intake"}, front_intake_tab );

    while ( cont )
    {

        switch ( tab_loaded ) //switches to tab user wants to go to
        {
            case 0:
                l_chassis_tab_debug.update_label(target_velocity, velocity_label);
                break;
            case 1:
                r_chassis_tab_debug.update_label(target_velocity, velocity_label);
                break;
            case 2:
                main_intake_tab_debug.update_label(target_velocity, velocity_label);
                break;
            case 3:
                front_intake_tab_debug.update_label(target_velocity, velocity_label);
                break;

        }


        Motors::set_brake_mode(current_brake_mode);



        pros::delay(200);
    }

    //reallow motor to hit zero velocity for driver controll
    Motors::enable_driver_control();
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/SensorsDebug.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 * TODO: condense, there are several classes that could be combined so that their is not so many
 *
 * contains classes for tabs of the sensors debugger tab
 */

#ifndef __SENSORDEBUG_HPP__
#define __SENSORDEBUG_HPP__

#include <string>

#include "../../../../include/main.h"

#include "../Styles.hpp"
#include "../Gimmicks.hpp"
#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"


//user defines

//sets size of container
#define SENSORS_CONTAINER_WIDTH 440
#define SENSORS_CONTAINER_HEIGHT 120


//Base classes
//base classes are the tabs that will be loaded by the derived class
//this makes it easy to add new tabs while keeping the amount that has
//to go in one class to a minimum, especially since lvgl is not light



// /**
//  * @see: ../Styles.hpp
//  *
//  * shows tab of IMEs and allows user to tare encoders and see values
//  */
// class IMEsDebugger :
//     virtual Styles
// {
//     private:
//         lv_obj_t *container;
//         lv_obj_t *title;
//         lv_obj_t *info;
//
//         lv_obj_t *btn_tare;
//         lv_obj_t *btn_tare_label;
//
//         /**
//          * @param: lv_obj_t* btn -> button that called the funtion
//          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
//          *
//          * button callback function used to tare all IMEs
//          */
//         static lv_res_t btn_tare_action(lv_obj_t *btn);
//
//
//     protected:
//         /**
//          * @return: None
//          *
//          * updates values for IMEs
//          */
//         void update_imes_info();
//
//     public:
//         IMEsDebugger();
//         virtual ~IMEsDebugger();
//
//         /**
//          * @param: lv_obj_t* parent -> parent of the tab
//          * @return: None
//          *
//          * objects are initially loaded onto a NULL parent to be updated later
//          * this sets it so that the parent of the objects is now the tab
//          */
//         void IMEsDebuggerInit(lv_obj_t *parent);
//
// };


// /**
//  * @see: ../Styles.
//  *
//  * show value for potentiometer
//  */
// class PotentiometerDebugger :
//     virtual Styles
// {
//     private:
//         lv_obj_t *container;
//
//         lv_obj_t *title1;
//         lv_obj_t *title2;
//         lv_obj_t *title3;
//
//         lv_obj_t *info1;
//         lv_obj_t *info2;
//         lv_obj_t *info3;
//
//         lv_obj_t *btn_calibrate;
//         lv_obj_t *btn_calibrate_label;
//
//         /**
//          * @param: lv_obj_t* btn -> button that called the funtion
//          * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
//          *
//          * button callback function used to calibrate sensor
//         */
//        static lv_res_t btn_calibrate_action(lv_obj_t *btn);
//
//    protected:
//        /**
//         * @return: None
//         *
//         * updates value of potentiometer
//         */
//        void update_pot_info();
//
//    public:
//        PotentiometerDebugger();
//        virtual ~PotentiometerDebugger();
//
//        /**
//         * @param: lv_obj_t* parent -> parent of the tab
//         * @return: None
//         *
//         * objects are initially loaded onto a NULL parent to be updated later
//         * this sets it so that the parent of the objects is now the tab
//         */
//        void PotentiometerDebuggerInit(lv_obj_t *parent);
// };
//
//
//
// /**
//  * @see: ../Styles.
//  *
//  * show value for limit switch
//  */
// class LimitSwitchDebugger :
//    virtual Styles
// {
// private:
//    lv_obj_t *container;
//
//    lv_obj_t *title1;
//    lv_obj_t *title2;
//
//    lv_obj_t *info1;
//    lv_obj_t *info2;
//
//    protected:
//        /**
//         * @return: None
//         *
//         * updates value of limit switch
//         */
//        void update_limit_switch_info();
//
//    public:
//        LimitSwitchDebugger();
//        virtual ~LimitSwitchDebugger();
//
//        /**
//         * @param: lv_obj_t* parent -> parent of the tab
//         * @return: None
//         *
//         * objects are initially loaded onto a NULL parent to be updated later
//         * this sets it so that the parent of the objects is now the tab
//         */
//        void LimitSwitchDebuggerInit(lv_obj_t *parent);
// };
//
//
//
// /**
//  * @see: ../Styles.
//  *
//  * starts new page with debugger info for vision sensor because it needs more room
//  */
// class VisionSensorDebugger : virtual Styles
// {
//    private:
//        lv_obj_t *title_label;
//
//        lv_obj_t *vision_sensor_screen;
//
//        //back button
//        lv_obj_t *btn_back;
//        lv_obj_t *btn_back_label;
//
//        /**
//         * @param: lv_obj_t* btn -> button that called the funtion
//         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
//         *
//         * button callback function used to go back from the new screen loaded by
//         * this tab because it is predicted to need more space
//         */
//        static lv_res_t btn_back_action(lv_obj_t *btn);
//
//        static bool cont;
//
//    protected:
//        /**
//         * @return: None
//         *
//         * loads a new page with debug info
//         */
//        void load_vision_sensor_page();
//
//    public:
//        VisionSensorDebugger();
//        virtual ~VisionSensorDebugger();
// };
//



//derived class

```

```

/**
 * @see: ../Styles.
 *
 * starts tab object with all the sensor tabs that the user
 * can switch between
 */
class SensorsDebug :
    virtual private Styles
{
    private:
        //screen
        lv_obj_t *sensors_debug_screen;

        //title label
        lv_obj_t *title_label;

        //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to go back from the debug screen to
         * the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


        static lv_obj_t *tabview; //tabview object

        lv_obj_t *imes_tab;    //individual tabs
        lv_obj_t *analog_in_tab;
        lv_obj_t *digital_in_tab;
        lv_obj_t *imu_tab;
        lv_obj_t *encoders_tab;
        lv_obj_t *vision_sensor_tab;


    public:
        SensorsDebug();
        ~SensorsDebug();

        static bool all_cont; //checks whether to allow user to
                              //cycle through tabs or not


        /**
         * @return: None
         *
         * contains methods for transition between tabs with checking sensors
         * for if they are calibrated or not
         * waits for user to go back in a loop while also switching tabs
         *
         */
        void debug();

};




#endif
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/SensorsDebug.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: SensorsDebug.hpp
 *
 * contains all methods for tabs that contain ways to debug and check sensors
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "../Gimmicks.hpp"
#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"
#include "SensorsDebug.hpp"
#include "sensor_tabs/AnalogInTab.hpp"
#include "sensor_tabs/DigitalInTab.hpp"
#include "sensor_tabs/EncoderTab.hpp"
#include "sensor_tabs/IMETab.hpp"
#include "sensor_tabs/IMUTab.hpp"
#include "sensor_tabs/VisionSensorTab.hpp"

//base classes
// bool VisionSensorDebugger::cont = true;
bool SensorsDebug::all_cont = true;
lv_obj_t *SensorsDebug::tabview;


SensorsDebug::SensorsDebug()
{
//set default for statics
    all_cont = true;

//init screen
    sensors_debug_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(sensors_debug_screen, &gray);

//init title label
    title_label = lv_label_create(sensors_debug_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, SENSORS_CONTAINER_WIDTH);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Sensors - Debug");

//init tabview
    tabview = lv_tabview_create(sensors_debug_screen, NULL);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BG, &gray);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_REL, &toggle_tabbtn_released);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_PR, &toggle_tabbtn_pressed);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_INDIC, &sw_indic);
    lv_tabview_set_style(tabview, LV_TABVIEW_STYLE_BTN_TGL_REL, &toggle_tabbtn_pressed);
    //lv_tabview_set_tab_load_action(tabview, tab_load_action);
    lv_obj_set_width(tabview, SENSORS_CONTAINER_WIDTH);
    lv_obj_set_height(tabview, 200);

//init tabs
    imes_tab = lv_tabview_add_tab(tabview, "IMEs");
    analog_in_tab = lv_tabview_add_tab(tabview, "Analog In");
    digital_in_tab = lv_tabview_add_tab(tabview, "Digital In");
    imu_tab = lv_tabview_add_tab(tabview, "IMU");
    encoders_tab = lv_tabview_add_tab(tabview, "Encoders");
    vision_sensor_tab = lv_tabview_add_tab(tabview, "Vision\nSensor");


//init back button
    //button
    btn_back = lv_btn_create(sensors_debug_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init tabs from other classes
    //IMEsDebuggerInit(imes_tab);
    //PotentiometerDebuggerInit(pot_tab);
    //LimitSwitchDebuggerInit(limit_tab);

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_set_pos(title_label, 180, 5);

    lv_obj_set_pos(tabview, 20, 25);
}


SensorsDebug::~SensorsDebug()
{
    //deletes widgets instantiated by class
    lv_obj_del(title_label);

    lv_obj_del(btn_back_label);
    lv_obj_del(btn_back);

    lv_obj_del(imes_tab);
    lv_obj_del(analog_in_tab);
    lv_obj_del(digital_in_tab);
    lv_obj_del(vision_sensor_tab);

    lv_obj_del(tabview);

    lv_obj_del(sensors_debug_screen);
```

```cpp
}



/**
 * callback funciton that exits main loop when button is pressed
 */
lv_res_t SensorsDebug::btn_back_action(lv_obj_t *btn)
{
    all_cont = 0;
    return LV_RES_OK;
}



/**
 * switches on tab loaded, this corresponds to a sensor tab
 * if this sensor needs to be calibrated then there is a warning box that
 * lets the user choosed to calibrate the sensor, and will not allow the user
 * to access the tab until the sensor is calibrated
 */
void SensorsDebug::debug()
{
    //used to check if user wants to continue cycling through
    //tabs. Will be set to zero and loop will break if user hits
    //the back button
    all_cont = 1;

    std::vector<Motor*> v1(Motors::motor_array.begin(), Motors::motor_array.end());
    std::vector<std::string> v2(Motors::motor_names_array.begin(), Motors::motor_names_array.end());

    IMEsDebugger imes_debug(
        imes_tab,
        SENSORS_CONTAINER_WIDTH,
        SENSORS_CONTAINER_HEIGHT,
        v1,
        v2
    );
    AnalogInDebugger analog_in_debug(analog_in_tab, SENSORS_CONTAINER_WIDTH, SENSORS_CONTAINER_HEIGHT, {&Sensors::line_tracker_top, &Sensors::line_tracker_middle, &Sensors::line_tracker_bottom}, {"Tracker Top", "Tracker Middle", "Tracker Bottom"});
    IMUDebugger imu_debug(imu_tab, SENSORS_CONTAINER_WIDTH, SENSORS_CONTAINER_HEIGHT, &Sensors::imu);
    EncoderDebugger encoder_debug(
        encoders_tab,
        SENSORS_CONTAINER_WIDTH,
        SENSORS_CONTAINER_HEIGHT,
        {
            &Sensors::right_encoder,
            &Sensors::left_encoder,
            &Sensors::strafe_encoder
        },
        {
            "R Encoder",
            "L Encoder",
            "S Encoder"
        }
    );

    lv_tabview_set_tab_act(tabview, 0, NULL);
    lv_scr_load(sensors_debug_screen);

    while ( all_cont )
    {
        switch ( lv_tabview_get_tab_act(tabview) ) //switches to tab user wants to go to
        {
            case 0:
                imes_debug.update_info();
                break;


            case 1:
                analog_in_debug.update_info();
                // if ( !(Sensors::potentiometer.is_calibrated()) )   //checks for sensor being
                //                                                    //calibrated. If not warning
                //                                                    //will appear
                // {
                //     lv_tabview_set_sliding(tabview, false); //dissallows changing
                //                                             //tab until user
                //                                             //has selected a
                //                                             //calibrate option
                //
                //     std::string msg = (
                //         "Potentiometer has not been calibrated.\n"
                //         "Click continue to calibrate, or back to\n"
                //         "return to a previous screen\n\n"
                //         "(Please keep sensor still while calibrating)\n"
                //     );
                //
                //     WarningMessage warnmsg;
                //     bool calibrated = warnmsg.warn(msg, sensors_debug_screen);
                //
                //     lv_tabview_set_sliding(tabview, true); //re-enables switching
                //                                            //tabs
                //
                //     if ( calibrated )
                //     {
                //         Loading load;
                //         load.show_load(500, sensors_debug_screen, 190, 125); //shows loading circle while calibrating
                //         Sensors::potentiometer.calibrate();
                //         load.hide_load();
                //
                //         update_pot_info();
                //     }
                //
                //     else
                //     {
                //         lv_tabview_set_tab_act(tabview, 0, NULL);
                //         //tab_loaded = 0;
                //     }
                // }
                //
                // else //if Accelerometer is already calibrated
                // {
                //     update_pot_info();
                // }

                break;

            // case 2:
            //     digital_in_debug.update_info();
            //     break;
            //
            // case 3:
            //     load_vision_sensor_page();
            //     lv_scr_load(sensors_debug_screen);
            //
            //     //switch to a different tab or user will be unable to leave
            //     //vision sensor debugger
            //     lv_tabview_set_tab_act(tabview, 0, NULL);
            //     //tab_loaded = 0;
            //     break;
            case 2:
                imu_debug.update_info();

            case 3:
                encoder_debug.update_info();

        }

        pros::delay(200);
    }
}
```

```
/**
 * @file: ./RobotCode/src/lcdCode/Debug/TitleScreen.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * contains class that allows user to select a debug tab
 *
 */

#ifndef __TITLESCREEN_HPP__
#define __TITLESCREEN_HPP__


#include "../../../../include/main.h"

#include "../Styles.hpp"


/**
 * @see: ../Styles.hpp
 * @see: Debug.hpp
 *
 * contains button matrix that has different debug tabs on it
 */
class TitleScreen : private Styles
{
    private:
        //screen
        lv_obj_t *title_screen;

        lv_obj_t *title_label;

        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set the debug option to -1 meaning the
         * user wants to go to the previous screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);


        lv_obj_t *button_matrix; //button matrix object
        static const char* btnm_map[]; //map for button matrix

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set option of debug screen that user wants to go to
         */
        static lv_res_t button_matrix_action(lv_obj_t *btnm, const char *btn_txt);


    public:
        TitleScreen();
        ~TitleScreen();

        static int option;

        /**
         * @return: None
         *
         * @see: btn_back_action
         * @see: button_matrix_action
         *
         * loads screen and waits in a loop with a delay for user to select
         * a button
         */
        void chooseOption();

};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/TitleScreen.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: TitleScreen.hpp
 *
 * contains class for selecting a debug screen or going to previous stage
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "TitleScreen.hpp"
#include "../../controller/controller.hpp"


int TitleScreen::option = 0;
const char* TitleScreen::btnm_map[] = {
        "Motors", "Sensors", "Controller", "Battery",
        "\n", "Field Control", "Wiring", "Internal\nMotor PID", ""
        };

TitleScreen::TitleScreen()
{
    option = 0;

    title_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(title_screen, &gray);

//init button matrix
    button_matrix = lv_btnm_create(title_screen, NULL);
    lv_btnm_set_map(button_matrix, btnm_map);
    lv_btnm_set_action(button_matrix, button_matrix_action);
    lv_obj_set_width(button_matrix, 440);
    lv_obj_set_height(button_matrix, 140);

    //set styles of button matrix
    lv_btnm_set_style(button_matrix, LV_BTNM_STYLE_BTN_REL, &toggle_btn_released);
    lv_btnm_set_style(button_matrix, LV_BTNM_STYLE_BTN_PR, &toggle_btn_pressed);

//init title label
    title_label = lv_label_create(title_screen, NULL);
    lv_obj_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 300);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Debugger");

//init back button
    //button
    btn_back = lv_btn_create(title_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");


//set postitions of widgets
    lv_obj_set_pos(btn_back, 210, 200);
    lv_obj_set_pos(title_label, 210, 20);
    lv_obj_set_pos(button_matrix, 20, 50);

}


TitleScreen::~TitleScreen()
{
    lv_obj_del(btn_back_label);
    lv_obj_del(btn_back);
    lv_obj_del(title_label);
    lv_obj_del(button_matrix);

    lv_obj_del(title_screen);
}



/**
 * compares text of button to text of label to see what button was clicked
 * sets int option to value based on the button clicked
 */
lv_res_t TitleScreen::button_matrix_action(lv_obj_t *btnm, const char *btn_txt)
{
    if (btn_txt == "Motors")
    {
        option = 1;
    }
    else if (btn_txt == "Sensors")
    {
        option = 2;
    }
    else if (btn_txt == "Controller")
    {
        option = 3;
    }
    else if (btn_txt == "Battery")
    {
        option = 4;
    }
    else if (btn_txt == "Field Control")
    {
        option = 5;
    }
    else if (btn_txt == "Wiring")
    {
        option = 6;
    }
    else if (btn_txt == "Internal\nMotor PID")
    {
        option = 7;
    }
    return LV_RES_OK;
}



/**
 * sets option to -1 which is to be interpreted as user wanting to go back
 */
lv_res_t TitleScreen::btn_back_action(lv_obj_t *btn)
{
    option = -1;
    return LV_RES_OK;
}



/**
 * waits for option to be non zero
 * this will happen once any button is clicked
 */
void TitleScreen::chooseOption()
{
    Controller controllers;
    option = 0;

    lv_scr_load(title_screen);
    while ( !option )
    {
        //allow controller to press the buttons as well
        if ( controllers.master.get_digital(pros::E_CONTROLLER_DIGITAL_B) )
        {
            btn_back_action( NULL );
            pros::delay(100);
        }
        pros::delay(20);
    }
}
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/Wiring.hpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * contains class that shows the current wiring of the robot
 */

#ifndef __WIRING_HPP__
#define __WIRING_HPP__

#include "../Styles.hpp"

#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"


/**
 * @see: ../Styles.hpp
 * @see: ../../motors/Motors.hpp
 * @see: ../../sensors/Sensors.hpp
 *
 * shows the ports that each motor or sensor is located on
 * purpose is to make it easier and more companct to wire the robot than having
 * to read off of separate computer screen
 */
class Wiring : private Styles
{
    private:
        lv_obj_t *wiring_screen;
        lv_obj_t *title_label;

        lv_obj_t *motor_info;
        lv_obj_t *sensors_info;

        //back button
        lv_obj_t *btn_back;
        lv_obj_t *btn_back_label;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to set cont to false meaning the
         * user wants to go to the title screen
         */
        static lv_res_t btn_back_action(lv_obj_t *btn);

    public:
        static bool cont;

        Wiring();
        ~Wiring();

        /**
         * @return: None
         *
         * passive screen -- loads text and wait for user to go back
         */
        void debug();

};



#endif
```

```cpp
/**
 * @file: ./RobotCode/src/lcdCode/Debug/Wiring.cpp
 * @author: Aiden Carney
 * @reviewed_on: 10/15/2019
 * @reviewed_by: Aiden Carney
 *
 * @see: Wiring.hpp
 *
 * contains class that shows wiring configuration
 */

#include "../../../../include/main.h"
#include "../../../../include/api.h"

#include "../Styles.hpp"
#include "Wiring.hpp"

#include "../../motors/Motors.hpp"
#include "../../sensors/Sensors.hpp"


bool Wiring::cont = true;

Wiring::Wiring()
{
    Configuration* config = Configuration::get_instance();

    cont = true;

//screen
    wiring_screen = lv_obj_create(NULL, NULL);
    lv_obj_set_style(wiring_screen, &gray);

//init back button
    //button
    btn_back = lv_btn_create(wiring_screen, NULL);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_back, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_back, LV_BTN_ACTION_CLICK, btn_back_action);
    lv_obj_set_width(btn_back, 75);
    lv_obj_set_height(btn_back, 25);

    //label
    btn_back_label = lv_label_create(btn_back, NULL);
    lv_obj_set_style(btn_back_label, &heading_text);
    lv_label_set_text(btn_back_label, "Back");

//init title label
    title_label = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(title_label, &heading_text);
    lv_obj_set_width(title_label, 440);
    lv_obj_set_height(title_label, 20);
    lv_label_set_align(title_label, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title_label, "Wiring");

//init motor info label
    motor_info = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(motor_info, &subheading_text);
    lv_obj_set_width(motor_info, 220);
    lv_obj_set_height(motor_info, 200);
    lv_label_set_align(motor_info, LV_LABEL_ALIGN_LEFT);

    std::string motors_text = (
            "front right     (200 RPM) - " + std::to_string(config->front_right_port) + "\n"
            "back right      (200 RPM) - " + std::to_string(config->back_left_port) + "\n"
            "front left      (200 RPM) - " + std::to_string(config->front_left_port) + "\n"
            "back left       (200 RPM) - " + std::to_string(config->back_right_port) + "\n"
            "left intake     (600 RPM) - " + std::to_string(config->left_intake_port) + "\n"
            "right intake    (600 RPM) - " + std::to_string(config->right_intake_port) + "\n"
            "upper_indexer   (600 RPM) - " + std::to_string(config->upper_indexer_port) + "\n"
            "lower_indexer   (600 RPM) - " + std::to_string(config->lower_indexer_port) + "\n"
    );

    lv_label_set_text(motor_info, motors_text.c_str());

//init motor info label
    sensors_info = lv_label_create(wiring_screen, NULL);
    lv_label_set_style(sensors_info, &subheading_text);
    lv_obj_set_width(sensors_info, 220);
    lv_obj_set_height(sensors_info, 200);
    lv_label_set_align(sensors_info, LV_LABEL_ALIGN_LEFT);

    std::string sensors_text = (
        std::string("right enc top     - ") + RIGHT_ENC_TOP_PORT + "\n" +
        "right enc bottom  - " + RIGHT_ENC_BOTTOM_PORT + "\n" +
        "left enc top      - " + LEFT_ENC_TOP_PORT + "\n" +
        "left enc bottom   - " + LEFT_ENC_BOTTOM_PORT + "\n" +
        "potentiometer     - " + POTENTIOMETER_PORT + "\n" +
        "top detector      - " + DETECTOR_TOP_PORT + "\n" +
        "middle detector   - " + DETECTOR_MIDDLE_PORT + "\n" +
        "bottom detector   - " + DETECTOR_BOTTOM_PORT + "\n" +
        "optical sensor    - " + std::to_string(OPTICAL_PORT) + "\n"
    );

    lv_label_set_text(sensors_info, sensors_text.c_str());

//set positions
    lv_obj_set_pos(btn_back, 30, 210);

    lv_obj_set_pos(title_label, 220, 5);

    lv_obj_set_pos(motor_info, 20, 25);
    lv_obj_set_pos(sensors_info, 300, 25);


}

Wiring::~Wiring()
{
    lv_obj_del(wiring_screen);
}


```

```
/**
 * sets cont to false signifying user wants to go back, main loop will exit
 */
lv_res_t Wiring::btn_back_action(lv_obj_t *btn)
{
    cont = false;
    return LV_RES_OK;
}


/**
 * waits for cont to be false which occurs when the user hits the back button
 */
void Wiring::debug()
{
    cont = true;

    lv_scr_load(wiring_screen);

    while ( cont )
    {
        pros::delay(100);
    }
}
```

../RobotCode/src/objects/lcdCode/Debug/sensor_tabs/AnalogInTab.hpp

```
#ifndef __ANALOGINTAB_HPP__
#define __ANALOGINTAB_HPP__

#include <string>
#include <vector>

#include "main.h"

#include "../../../sensors/AnalogInSensor.hpp"
#include "../../Styles.hpp"


class AnalogInDebugger :
    virtual Styles
{
    private:
        lv_obj_t *container;

        lv_obj_t *title1;
        lv_obj_t *title2;
        lv_obj_t *title3;

        lv_obj_t *info1;
        lv_obj_t *info2;
        lv_obj_t *info3;

        lv_obj_t *btn_calibrate;
        lv_obj_t *btn_calibrate_label;

        static std::vector<AnalogInSensor*> sensors;
        static std::vector<std::string> names;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to calibrate sensor
         */
        static lv_res_t btn_calibrate_action(lv_obj_t *btn);

    public:
        AnalogInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<AnalogInSensor*> sensors_vec, std::vector<std::string> names_vec);
        ~AnalogInDebugger();

        /**
         * @return: None
         *
         * updates value of sensors
         */
        void update_info();

};



#endif
```

```cpp
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "AnalogInTab.hpp"

std::vector<AnalogInSensor*> AnalogInDebugger::sensors;
std::vector<std::string> AnalogInDebugger::names;


AnalogInDebugger::AnalogInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<AnalogInSensor*> sensors_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < sensors_vec.size(); i++ )
    {
        sensors.push_back(sensors_vec.at(i));
        names.push_back(names_vec.at(i));
    }
//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//title for columns
    //1
    title1 = lv_label_create(container, NULL);
    lv_obj_set_style(title1, &toggle_tabbtn_pressed);
    lv_obj_set_width(title1, (x_dim));
    lv_obj_set_height(title1, 20);
    lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title1, "None");

    //2
    title2 = lv_label_create(container, NULL);
    lv_obj_set_style(title2, &toggle_tabbtn_pressed);
    lv_obj_set_width(title2, (x_dim/3));
    lv_obj_set_height(title2, 20);
    lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title2, "None");

    //3
    title3 = lv_label_create(container, NULL);
    lv_obj_set_style(title3, &toggle_tabbtn_pressed);
    lv_obj_set_width(title3, (x_dim/3));
    lv_obj_set_height(title3, 20);
    lv_label_set_align(title3, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title3, "None");

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

    //3
    info3 = lv_label_create(container, NULL);
    lv_obj_set_style(info3, &toggle_tabbtn_pressed);
    lv_obj_set_width(info3, (x_dim/3));
    lv_obj_set_height(info3, y_dim - 20);
    lv_label_set_align(info3, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info3, "None");

//calibrate button
    //button
    btn_calibrate = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_calibrate, LV_BTN_ACTION_CLICK, btn_calibrate_action);
    lv_obj_set_width(btn_calibrate, 110);
    lv_obj_set_height(btn_calibrate, 25);

    //label
    btn_calibrate_label = lv_label_create(btn_calibrate, NULL);
    lv_obj_set_style(btn_calibrate_label, &subheading_text);
    lv_label_set_text(btn_calibrate_label, "Calibrate");

//set positions relative to container
    lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);

    lv_obj_align(title2, container, LV_ALIGN_IN_TOP_MID, -15, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -15, 30);

    lv_obj_align(title3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
    lv_obj_align(info3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);

    lv_obj_align(btn_calibrate, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

AnalogInDebugger::~AnalogInDebugger()
{

}


/**
 * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
 */
```

```
lv_res_t AnalogInDebugger::btn_calibrate_action(lv_obj_t *btn)
{
    Loading load;
    for(int i=0; i < sensors.size(); i++)
    {
        load.show_load(500, lv_scr_act(), 190, 240); //shows loading bar while calibrating
        sensors.at(i)->calibrate();
        load.hide_load();
    }

    return LV_RES_OK;
}



/**
 * updates potentiometer data with raw and corrected values
 */
void AnalogInDebugger::update_info()
{
    std::string names_text = "";
    std::string raw_text = "";
    std::string corrected_text = "";
    for(int i=0; i < sensors.size(); i++)
    {
        names_text += names.at(i) + "\n";
        raw_text += std::to_string(sensors.at(i)->get_raw_value()) + "\n";
        corrected_text += std::to_string(sensors.at(i)->get_value(false)) + "\n";
    }

    lv_label_set_text(title1, "Sensor");
    lv_label_set_text(title2, "Raw Input");
    lv_label_set_text(title3, "Corrected Input");
    lv_label_set_text(info1, names_text.c_str());
    lv_label_set_text(info2, raw_text.c_str());
    lv_label_set_text(info3, corrected_text.c_str());
}
```

```
#ifndef __DIGITALINTAB_HPP__
#define __DIGITALINTAB_HPP__

#include <string>
#include <vector>

#include "main.h"

#include "../../Styles.hpp"


/**
 * @see: ../Styles.
 *
 * show value for limit switch
 */
class DigitalInDebugger :
    virtual Styles
{
private:
    lv_obj_t *container;

    lv_obj_t *title1;
    lv_obj_t *title2;

    lv_obj_t *info1;
    lv_obj_t *info2;

    static std::vector<pros::ADIDigitalIn*> sensors;
    static std::vector<std::string> names;

    public:
        DigitalInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<pros::ADIDigitalIn*> sensors_vec, std::vector<std::string> names_vec);
        ~DigitalInDebugger();

        /**
         * @return: None
         *
         * updates value of limit switch
         */
        void update_info();
};


#endif
```

../RobotCode/src/objects/lcdCode/Debug/sensor_tabs/DigitalInTab.cpp

```cpp
#include <string>
#include <vector>

#include "main.h"

#include "../../../motors/Motor.hpp"
#include "../../Styles.hpp"
#include "DigitalInTab.hpp"


std::vector<pros::ADIDigitalIn*> DigitalInDebugger::sensors;
std::vector<std::string> DigitalInDebugger::names;


DigitalInDebugger::DigitalInDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<pros::ADIDigitalIn*> sensors_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < sensors_vec.size(); i++ )
    {
        sensors.push_back(sensors_vec.at(i));
        names.push_back(names_vec.at(i));
    }

//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//title for columns
    //1
    title1 = lv_label_create(container, NULL);
    lv_obj_set_style(title1, &toggle_tabbtn_pressed);
    lv_obj_set_width(title1, (x_dim));
    lv_obj_set_height(title1, 20);
    lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title1, "None");

    //2
    title2 = lv_label_create(container, NULL);
    lv_obj_set_style(title2, &toggle_tabbtn_pressed);
    lv_obj_set_width(title2, (x_dim/3));
    lv_obj_set_height(title2, 20);
    lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title2, "None");

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim/3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");


//set positions relative to container
    lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);

    lv_obj_align(title2, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);

}

DigitalInDebugger::~DigitalInDebugger()
{

}


/**
 * shows value of limit switch as either 0 or 1
 */
void DigitalInDebugger::update_info()
{
    std::string names_text;
    std::string val_text;
    for(int i=0; i < sensors.size(); i++)
    {
        names_text += names.at(i) + "\n";
        val_text += std::to_string(sensors.at(i)->get_value()) + "\n";
    }

    lv_label_set_text(title1, "Limit Switch");
    lv_label_set_text(title2, "State");
    lv_label_set_text(info1, names_text.c_str());
    lv_label_set_text(info2, val_text.c_str());
}
```

```cpp
#ifndef __ENCODERTAB_HPP__
#define __ENCODERTAB_HPP__

#include <string>
#include <vector>

#include "main.h"

#include "../../../sensors/Encoder.hpp"
#include "../../Styles.hpp"


class EncoderDebugger :
    virtual Styles
{
    private:
        lv_obj_t *container;

        lv_obj_t *title1;
        lv_obj_t *title2;
        lv_obj_t *title3;

        lv_obj_t *info1;
        lv_obj_t *info2;
        lv_obj_t *info3;

        lv_obj_t *btn_tare;
        lv_obj_t *btn_tare_label;

        static std::vector<Encoder*> encoders;
        static std::vector<std::string> names;
        static std::vector<int> unique_ids;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to tare encoder
         */
        static lv_res_t btn_tare_action(lv_obj_t *btn);

    public:
        EncoderDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Encoder*> encoders_vec, std::vector<std::string> names_vec);
        ~EncoderDebugger();

        /**
         * @return: None
         *
         * updates value of sensors
         */
        void update_info();

};



#endif
```

```cpp
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "EncoderTab.hpp"

std::vector<Encoder*> EncoderDebugger::encoders;
std::vector<std::string> EncoderDebugger::names;
std::vector<int> EncoderDebugger::unique_ids;


EncoderDebugger::EncoderDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Encoder*> encoders_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < encoders_vec.size(); i++ )
    {
        encoders.push_back(encoders_vec.at(i));
        unique_ids.push_back(encoders_vec.at(i)->get_unique_id());
        names.push_back(names_vec.at(i));
    }
//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//title for columns
    //1
    title1 = lv_label_create(container, NULL);
    lv_obj_set_style(title1, &toggle_tabbtn_pressed);
    lv_obj_set_width(title1, (x_dim));
    lv_obj_set_height(title1, 20);
    lv_label_set_align(title1, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title1, "None");

    //2
    title2 = lv_label_create(container, NULL);
    lv_obj_set_style(title2, &toggle_tabbtn_pressed);
    lv_obj_set_width(title2, (x_dim/3));
    lv_obj_set_height(title2, 20);
    lv_label_set_align(title2, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title2, "None");

    //3
    title3 = lv_label_create(container, NULL);
    lv_obj_set_style(title3, &toggle_tabbtn_pressed);
    lv_obj_set_width(title3, (x_dim/3));
    lv_obj_set_height(title3, 20);
    lv_label_set_align(title3, LV_LABEL_ALIGN_CENTER);
    lv_label_set_text(title3, "None");

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

    //3
    info3 = lv_label_create(container, NULL);
    lv_obj_set_style(info3, &toggle_tabbtn_pressed);
    lv_obj_set_width(info3, (x_dim/3));
    lv_obj_set_height(info3, y_dim - 20);
    lv_label_set_align(info3, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info3, "None");

//calibrate button
    //button
    btn_tare = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_tare, LV_BTN_ACTION_CLICK, btn_tare_action);
    lv_obj_set_width(btn_tare, 110);
    lv_obj_set_height(btn_tare, 25);

    //label
    btn_tare_label = lv_label_create(btn_tare, NULL);
    lv_obj_set_style(btn_tare_label, &subheading_text);
    lv_label_set_text(btn_tare_label, "Calibrate");

//set positions relative to container
    lv_obj_align(title1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 30);

    lv_obj_align(title2, container, LV_ALIGN_IN_TOP_MID, -50, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -50, 30);

    lv_obj_align(title3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 10);
    lv_obj_align(info3, container, LV_ALIGN_IN_TOP_RIGHT, -100, 30);

    lv_obj_align(btn_tare, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

EncoderDebugger::~EncoderDebugger()
{

}


/**
```

```
 * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
 */
lv_res_t EncoderDebugger::btn_tare_action(lv_obj_t *btn)
{
    Loading load;
    for(int i=0; i < encoders.size(); i++)
    {
        load.show_load(500, lv_scr_act(), 190, 240); //shows loading bar while calibrating
        encoders.at(i)->reset(unique_ids.at(i));
        load.hide_load();
    }

    return LV_RES_OK;
}



/**
 * updates potentiometer data with raw and corrected values
 */
void EncoderDebugger::update_info()
{
    std::string names_text = "";
    std::string raw_text = "";
    std::string corrected_text = "";
    for(int i=0; i < encoders.size(); i++)
    {
        names_text += names.at(i) + "\n";
        raw_text += std::to_string(encoders.at(i)->get_absolute_position(false)) + "/" + std::to_string(encoders.at(i)->get_absolute_position(true)) + "\n";
        corrected_text += std::to_string(encoders.at(i)->get_position(unique_ids.at(i))) + "\n";
    }

    lv_label_set_text(title1, "Encoder");
    lv_label_set_text(title2, "Absolute Position");
    lv_label_set_text(title3, "Unique Position");
    lv_label_set_text(info1, names_text.c_str());
    lv_label_set_text(info2, raw_text.c_str());
    lv_label_set_text(info3, corrected_text.c_str());
}
```

```cpp
#ifndef __IMETAB_HPP__
#define __IMETAB_HPP__

#include <string>
#include <vector>

#include "main.h"

#include "../../Styles.hpp"
#include "../../../motors/Motor.hpp"

/**
 * @see: ../Styles.hpp
 *
 * shows tab of IMEs and allows user to tare encoders and see values
 */
class IMEsDebugger :
    virtual Styles
{
    private:
        lv_obj_t *container;
        lv_obj_t *title;
        lv_obj_t *info;

        lv_obj_t *btn_tare;
        lv_obj_t *btn_tare_label;

        static std::vector<Motor*> motors;
        static std::vector<std::string> names;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to tare all IMEs
         */
        static lv_res_t btn_tare_action(lv_obj_t *btn);

    public:
        IMEsDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Motor*> motors_vec, std::vector<std::string> names_vec);
        ~IMEsDebugger();

        /**
         * @return: None
         *
         * updates values for IMEs
         */
        void update_info();


};


#endif
```

```cpp
#include <string>
#include <vector>

#include "main.h"

#include "../../../motors/Motor.hpp"
#include "../../Styles.hpp"
#include "IMETab.hpp"


std::vector<Motor*> IMEsDebugger::motors;
std::vector<std::string> IMEsDebugger::names;

IMEsDebugger::IMEsDebugger(lv_obj_t *parent, int x_dim, int y_dim, std::vector<Motor*> motors_vec, std::vector<std::string> names_vec)
{
    for( int i = 0; i < motors_vec.size(); i++ )
    {
        motors.push_back(motors_vec.at(i));
        names.push_back(names_vec.at(i));
    }

//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//default text
    std::string text = (
        "front right   -\n"
        "back right    -\n"
        "front left    -\n"
        "back left     -\n"
        "right lift    -\n"
        "left lift     -\n"
        "intake        -\n"
        "lift          -  "
    );

//init integrated motor encoders label label
    info = lv_label_create(container, NULL);
    lv_obj_set_style(info, &toggle_tabbtn_pressed);
    lv_obj_set_width(info, (x_dim));
    lv_obj_set_height(info, y_dim);
    lv_label_set_align(info, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info, text.c_str());


//init tare encoders button
    //button
    btn_tare = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_tare, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_tare, LV_BTN_ACTION_CLICK, btn_tare_action);
    lv_obj_set_width(btn_tare, 110);
    lv_obj_set_height(btn_tare, 25);

    //label
    btn_tare_label = lv_label_create(btn_tare, NULL);
    lv_obj_set_style(btn_tare_label, &subheading_text);
    lv_label_set_text(btn_tare_label, "tare encoders");


//align objects on container
    lv_obj_set_pos(info, 10, 0);
    lv_obj_set_pos(btn_tare, 300, (y_dim - 30));
}

IMEsDebugger::~IMEsDebugger()
{

}


/**
 * tares encodes of all motors
 */
lv_res_t IMEsDebugger::btn_tare_action(lv_obj_t *btn)
{
    for( int i = 0; i < motors.size(); i++ )
    {
        motors.at(i)->tare_encoder();
    }

    return LV_RES_OK;
}


/**
 * updates for each motor to current values
 */
void IMEsDebugger::update_info()
{
    int max_characters = 0;
    for( int i = 0; i < names.size(); i++ )
    {
        if(names.at(i).length() > max_characters)
        {
            max_characters = names.at(i).length();
        }
    }

    std::string text;
    for( int i = 0; i < names.size(); i++ )
    {
        std::string spaces = "";
        for(int j = names.at(i).length(); j < max_characters + 3; j++)
        {
            spaces += " ";
        }
        text += names.at(i) + spaces + "-  " + std::to_string(motors.at(i)->get_encoder_position()) + "\n";
    }

    lv_label_set_text(info, text.c_str());
}
```

```cpp
#ifndef __IMUTAB_HPP__
#define __IMUTAB_HPP__

#include <string>
#include <vector>

#include "main.h"

#include "../../../sensors/AnalogInSensor.hpp"
#include "../../Styles.hpp"


class IMUDebugger :
    virtual Styles
{
    private:
        lv_obj_t *container;

        lv_obj_t *info1;
        lv_obj_t *info2;

        lv_obj_t *btn_calibrate;
        lv_obj_t *btn_calibrate_label;

        static pros::Imu *imu;

        /**
         * @param: lv_obj_t* btn -> button that called the funtion
         * @return: lv_res_t -> LV_RES_OK on successfull completion because object still exists
         *
         * button callback function used to calibrate sensor
         */
        static lv_res_t btn_calibrate_action(lv_obj_t *btn);

    public:
        IMUDebugger(lv_obj_t *parent, int x_dim, int y_dim, pros::Imu *imu_sensor);
        ~IMUDebugger();

        /**
         * @return: None
         *
         * updates value of sensors
         */
        void update_info();

};



#endif
```

```
#include <string>
#include <vector>

#include "main.h"

#include "../../Gimmicks.hpp"
#include "IMUTab.hpp"

pros::Imu *IMUDebugger::imu;

IMUDebugger::IMUDebugger(lv_obj_t *parent, int x_dim, int y_dim, pros::Imu *imu_sensor)
{
    imu = imu_sensor;

//init container
    container = lv_cont_create(parent, NULL);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_style(container, &gray);
    lv_cont_set_fit(container, false, false);
    lv_obj_set_width(container, x_dim);
    lv_obj_set_height(container, y_dim);

//info for columns
    //1
    info1 = lv_label_create(container, NULL);
    lv_obj_set_style(info1, &toggle_tabbtn_pressed);
    lv_obj_set_width(info1, (x_dim/3));
    lv_obj_set_height(info1, y_dim - 20);
    lv_label_set_align(info1, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info1, "None");

    //2
    info2 = lv_label_create(container, NULL);
    lv_obj_set_style(info2, &toggle_tabbtn_pressed);
    lv_obj_set_width(info2, (x_dim / 3));
    lv_obj_set_height(info2, y_dim - 20);
    lv_label_set_align(info2, LV_LABEL_ALIGN_LEFT);
    lv_label_set_text(info2, "None");

//calibrate button
    //button
    btn_calibrate = lv_btn_create(container, NULL);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_REL, &toggle_btn_released);
    lv_btn_set_style(btn_calibrate, LV_BTN_STYLE_PR, &toggle_btn_pressed);
    lv_btn_set_action(btn_calibrate, LV_BTN_ACTION_CLICK, btn_calibrate_action);
    lv_obj_set_width(btn_calibrate, 110);
    lv_obj_set_height(btn_calibrate, 25);

    //label
    btn_calibrate_label = lv_label_create(btn_calibrate, NULL);
    lv_obj_set_style(btn_calibrate_label, &subheading_text);
    lv_label_set_text(btn_calibrate_label, "Calibrate");

//set positions relative to container
    lv_obj_align(info1, container, LV_ALIGN_IN_TOP_LEFT, 10, 10);
    lv_obj_align(info2, container, LV_ALIGN_IN_TOP_MID, -15, 10);

    lv_obj_align(btn_calibrate, container, LV_ALIGN_IN_BOTTOM_RIGHT, -50, 0);
}

IMUDebugger::~IMUDebugger()
{

}


/**
 * calibrates potentiometer and adds loading bar show gui doesn't appear to hang
 */
lv_res_t IMUDebugger::btn_calibrate_action(lv_obj_t *btn)
{
    Loading load;

    load.show_load(2000, lv_scr_act(), 190, 240); //shows loading bar while calibrating
    if(!(imu->get_status() && 0xFF))
    {
        imu->reset();
        while(imu->is_calibrating())
        {
            pros::delay(25);
        }
        load.hide_load();
    }
    return LV_RES_OK;
}



/**
 * updates potentiometer data with raw and corrected values
 */
void IMUDebugger::update_info()
{
    std::string names_text;
    std::string data_text;

    names_text += "heading\nrotation\npitch\nroll\nyaw\naccel x\naccel y\naccel z";
    data_text += std::to_string(imu->get_heading()) + "\n";
    data_text += std::to_string(imu->get_rotation()) + "\n";
    data_text += std::to_string(imu->get_pitch()) + "\n";
    data_text += std::to_string(imu->get_roll()) + "\n";
    data_text += std::to_string(imu->get_yaw()) + "\n";
    data_text += std::to_string(imu->get_accel().x) + "\n";
    data_text += std::to_string(imu->get_accel().y) + "\n";
    data_text += std::to_string(imu->get_accel().z) + "\n";

    lv_label_set_text(info1, names_text.c_str());
    lv_label_set_text(info2, data_text.c_str());
}
```

```cpp
/**
 * @file: ./RobotCode/src/objects/serial/Logger.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/9/2020
 * @reviewed_by: Aiden Carney
 *
 * contains class for a writer queue that accepts writes and flushes them
 * to an output stream
 */

#ifndef __LOGGER_HPP__
#define __LOGGER_HPP__

#include <atomic>
#include <queue>
#include <string>


typedef struct
{
    std::string stream;
    std::string content;
} log_entry;


/**
 * Contains a queue that can be added to and dumped out so that data can
 * be gathered and exported
 *
 */
class Logger
{
    private:
        static std::queue<log_entry> logger_queue;
        static std::atomic<bool> lock;
        static bool use_queue;

        /**
         * @param: num_entries -> max number of entries to get
         * @return: std::vector<log_entry> -> the list of items gotton from queue
         *
         * gets an object from the logger queue
         * returns an empty string if the queue is empty
         */
        std::vector<log_entry> get_entries(int num_entries);

        /**
         * @param: log_entry contents -> what to log
         * @return: bool -> true if the file was actually written to, false if an error occured
         *
         * sends an entry on a given stream
         * currently supports cout, clog, and cerr
         */
        bool log( log_entry entry );

    public:
        Logger();
        ~Logger();

        /**
         * @param: log_entry test_item -> item to add to the writer queue
         * @return: bool -> true on success and false if an error occured in the process
         *
         * adds an item to the logger queue
         * the queue is protected using a spinlock implemented with an
         * std::atomic bool
         */
        bool add( log_entry entry );

        /**
         * @return: None
         *
         * builds up a cache of items from the queue for 50ms so that they can be
         * logged at closer to the max speed
         */
        void dump( );

        static void stop_queueing();
        static void start_queueing();



        /**
         * @return: int -> number of items in the logger queue
         *
         * returns the size of the logger queue
         */
        static int get_count();
};



#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/serial/Logger.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/9/2020
 * @reviewed_by: Aiden Carney
 *
 * @see Logger.hpp
 *
 * contains implementation for the logger class
 */

#include <atomic>
#include <iostream>
#include <queue>
#include <string>
#include <vector>

#include "main.h"

#include "Logger.hpp"

std::queue<log_entry> Logger::logger_queue;
std::atomic<bool> Logger::lock = ATOMIC_VAR_INIT(false);
bool Logger::use_queue = true;


Logger::Logger() { }




Logger::~Logger() { }




/**
 * fsends data on the given stream based on the log entry
 */
bool Logger::log( log_entry entry )
{
    if ( entry.stream == "cout" )
    {
        std::cout << pros::millis() << " " << entry.content << "\n";
    }
    else if ( entry.stream == "cerr" )
    {
        std::cerr << pros::millis() << " " << entry.content << "\n";
    }
    else if ( entry.stream == "clog" )
    {
        std::clog << pros::millis() << " " << entry.content << "\n";
    }
    else
    {
        return false;
    }

    return true;
}




/**
 * add item to the queue by aquiring and releasing atomic lock
 */
bool Logger::add( log_entry entry )
{
    if ( !entry.stream.empty() && !entry.content.empty() )
    {
        if(use_queue) {  // save the message in a queue to be viewed later
            while ( lock.exchange( true ) ); //aquire lock
            logger_queue.push( entry );
            lock.exchange( false ); //release lock
        } else {  // log the message right away
            log(entry);
        }

        return true;
    }

    return false;
}




/**
 * gets an item from the queue by acquiring the lock and releasing it
 */
std::vector<log_entry> Logger::get_entries(int num_entries)
{
    std::vector<log_entry> contents;

    while ( lock.exchange( true ) ); //aquire lock

    for(int i=0; i<num_entries; i++) {
        if ( !logger_queue.empty() ) {
            contents.push_back(logger_queue.front());
            logger_queue.pop();
        } else {
            break;
        }
    }

    lock.exchange( false ); //release lock because there is no more iteraction
                            //with the queue
    return contents;
}
```

```cpp

/**
 * builds up a cache of items
 * this is used so that data can be sent at closer to the max speed
 */
void Logger::dump( )
{
    std::vector<log_entry> entries = get_entries(50);

    for ( int i = 0; i < entries.size(); i++ )
    {
        log(entries.at(i));
    }

}


void Logger::start_queueing() {
    use_queue = true;
}

void Logger::stop_queueing() {
    use_queue = false;
}


/**
 * gets the size of the writer queue
 */
int Logger::get_count()
{
    return logger_queue.size();
}
```

```
/**
 * @file: ./RobotCode/src/objects/serial/Server.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains class for a server that works over serial communication
 */

#ifndef __SERVER_HPP__
#define __SERVER_HPP__

#include <atomic>
#include <queue>
#include <cstdint>


typedef struct
{
    uint16_t return_id;
    uint16_t command_id;
    std::string msg;
} server_request;

class Server
{
    private:
        static std::atomic<bool> lock;
        static std::queue<server_request> request_queue;

        static pros::Task *read_thread;  // the thread for reading stdin

        static void read_stdin(void*);

        static int num_instances;
        static bool debug;

        static int delay;

        int handle_request(server_request request);

    public:
        Server();
        ˜Server();

        /**
         * @return: None
         *
         * starts the thread or resmes it if it was stopped
         */
        void start_server();

        /**
         * @return: None
         *
         * stops the thread from being scheduled
         */
        void stop_server();

        void set_server_task_priority(int new_prio);

        void set_debug_mode(bool debug_mode);

        void clear_stdin();

        int handle_requests(int max_requests=10);
};


#endif
```

```
/**
 * @file: ./RobotCode/src/objects/serial/Server.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains implementation for server implementation
 */
#include <atomic>
#include <cstdint>
#include <queue>
#include <string>

#include "main.h"
#include "pros/apix.h"

#include "../../Configuration.hpp"
#include "../motors/Motors.hpp"
#include "../motors/MotorThread.hpp"
#include "Logger.hpp"
#include "Server.hpp"

std::queue<server_request> Server::request_queue;
std::atomic<bool> Server::lock = ATOMIC_VAR_INIT(false);
pros::Task *Server::read_thread = NULL;
int Server::num_instances = 0;
bool Server::debug = false;
int Server::delay = 100;


Server::Server() {
    if(read_thread == NULL) {
        read_thread = new pros::Task( read_stdin, (void*)NULL, 2, TASK_STACK_DEPTH_DEFAULT, "server_thread");
        read_thread->suspend();
    }

    num_instances += 1;
}




Server::~Server() {
    std::cout << "destructor called on server\n";
    num_instances -= 1;
    if(num_instances == 0) {
        read_thread->remove();
        delete read_thread;
        read_thread = NULL;
    }
}



void Server::read_stdin(void*) {
    int read_check = 0;
    Logger logger;
    log_entry entry;
    int wait_check = 0;

    while(1) {
        char byte = getchar_unlocked();

        if(debug) {
            entry.stream = "clog";
            entry.content = "[INFO] " + std::to_string(pros::millis()) + " Byte read from stdin: " + byte;
            logger.add(entry);
        }

        if(read_check == 0 && byte == ’\xAA’) {
            read_check = 1;
        } else if(read_check == 1 && byte == ’\x55’) {
            read_check = 2;
        } else if(read_check == 2 && byte == ’\x1E’) {
            read_check = 3;
        } else if(read_check == 3) {
            std::string msg;
            int len_msg = (int)byte - 4;  // byte read will be length of bytes to follow
                                          // subtract 4 because next four bytes are handled different
                                          // because they are identifiers
            uint8_t return_msb = getchar_unlocked();
            uint8_t return_lsb = getchar_unlocked();
            uint8_t command_msb = getchar_unlocked();
            uint8_t command_lsb = getchar_unlocked();

            uint16_t return_id = (return_msb << 8) | return_lsb;
            uint16_t command_id = (command_msb << 8) | command_lsb;

            for(int i=0; i<len_msg; i++) {  // read rest message
                msg.push_back(getchar_unlocked());
            }

            char checksum = getchar_unlocked();  // checksum is directly after end of message

            if(debug) {
                entry.stream = "clog";
                entry.content = (
                    "[INFO], "
                    + std::to_string(pros::millis())
                    + ", Return ID read: " + std::to_string(return_id)
                    + ", Command ID read: " + std::to_string(command_id)
                    + ", Msg read: " + msg
                    + ", Checksum read: " + checksum
                );
                logger.add(entry);
            }

            if(checksum == ’\xC6’)
            {
                server_request request;
                request.return_id = return_id;
                request.command_id = command_id;
                request.msg = msg;

                while ( lock.exchange( true ) ); //aquire lock
                request_queue.push(request);
                msg = '\0';
                lock.exchange( false ); //release lock
            }

            read_check = 0;
            len_msg = 0;
            msg[0] = '\0';

        } else {
            read_check = 0;
        }

        wait_check += 1;
        if(wait_check > 1024) {
            wait_check = 0;
            pros::delay(10);
        }
    }
}



int Server::handle_request(server_request request) {
    // cases are defined in commands.ods
    Logger logger;
    log_entry entry;
    entry.stream = "clog";

    std::string return_msg;
    return_msg.push_back('\xAA');
    return_msg.push_back('\x55');
    return_msg.push_back('\x1E');

    std::string return_msg_body;
    int status;

    switch(request.command_id) {
    // motor interaction post cases
        case 45232: {  //0xB0 0xB0  Set voltage
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int voltage = std::stoi(request.msg);

                status = Motors::motor_array.at(motor_number)->set_voltage(voltage);
            }
            break;

        case 45233: {  //0xB0 0xB1  Set Slew Rate
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int slew_rate = std::stoi(request.msg);

                status = Motors::motor_array.at(motor_number)->set_slew(slew_rate);
            }
            break;

        case 45234: {  //0xB0 0xB2  Set Port
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int port = std::stoi(request.msg);

                status = Motors::motor_array.at(motor_number)->set_port(port);
            }
            break;

        case 45235: {  //0xB0 0xB3  Tare IME
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);

                status = Motors::motor_array.at(motor_number)->tare_encoder();
            }
            break;

        case 45236: {  //0xB0 0xB4  Set Brakemode
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                pros::motor_brake_mode_e_t new_brake_mode = static_cast<pros::motor_brake_mode_e_t>(std::stoi(request.msg));

                status = Motors::motor_array.at(motor_number)->set_brake_mode(new_brake_mode);
            }
            break;

        case 45237: {  //0xB0 0xB5  Set Gearing
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                pros::motor_gearset_e_t new_gearing = static_cast<pros::motor_gearset_e_t>(std::stoi(request.msg));

                status = Motors::motor_array.at(motor_number)->set_gearing(new_gearing);
            }
            break;

        case 45238: {  //0xB0 0xB6  Set PID
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);

                char buffer1[8];
                char buffer2[8];
                char buffer3[8];
                char buffer4[8];

                std::copy(request.msg.begin(), request.msg.begin() + 8, buffer1);
                std::copy(request.msg.begin() + 8, request.msg.begin() + 16, buffer2);
                std::copy(request.msg.begin() + 16, request.msg.begin() + 24, buffer3);
                std::copy(request.msg.begin() + 24, request.msg.begin() + 32, buffer4);

                double n1 = *reinterpret_cast<double*>(buffer1);
                double n2 = *reinterpret_cast<double*>(buffer2);
                double n3 = *reinterpret_cast<double*>(buffer3);
                double n4 = *reinterpret_cast<double*>(buffer4);

                pid pid_constants;
                pid_constants.kP = n1;
                pid_constants.kI = n2;
                pid_constants.kD = n3;
                pid_constants.I_max = n4;

                status = Motors::motor_array.at(motor_number)->set_pid(pid_constants);
            }
            break;

        case 45239: {  //0xB0 0xB7  Reverse Motor
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int reveresed = std::stoi(request.msg);
                status = Motors::motor_array.at(motor_number)->reverse_motor();
            }
            break;
        case 45240: {  //0xB0 0xB8  Set Log Level
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int new_log_level = std::stoi(request.msg);
                Motors::motor_array.at(motor_number)->set_log_level(new_log_level);
                status = 1;
            }
            break;
        case 45241: {  //0xB0 0xB9  Set Slew enabled/disabled
                int motor_number = std::stoi(std::to_string(request.msg.at(0)));
                request.msg.erase(0);
                int enabled = std::stoi(request.msg);
                if(enabled) {
                    Motors::motor_array.at(motor_number)->enable_slew();
                } else {
                    Motors::motor_array.at(motor_number)->disable_slew();
                }
                status = 1;
            }
            break;


    // motor interaction get cases
        case 41120: {  // 0xA0 0xA0  Actual Velocity
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_actual_velocity());
            }
            break;

        case 41121: {  // 0xA0 0xA1  Actual Voltage
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_actual_voltage());
            }
            break;

        case 41122: {  // 0xA0 0xA2  Current Draw
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_current_draw());
            }
            break;

        case 41123: {  // 0xA0 0xA3  Encoder Position
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_encoder_position());
            }
            break;

        case 41124: {  // 0xA0 0xA4  Brakemode
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_brake_mode());
            }
            break;

        case 41125: {  // 0xA0 0xA5  Gearset
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_gearset());
            }
            break;

        case 41126: {  // 0xA0 0xA6  Port
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_port());
            }
            break;

        case 41127: {  // 0xA0 0xA7  PID Constants
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body += std::to_string(Motors::motor_array.at(motor_number)->get_pid().kP);
                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().kI);
                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().kD);
                return_msg_body += " " + std::to_string(Motors::motor_array.at(motor_number)->get_pid().I_max);
            }
            break;

        case 41128: { // 0xA0 0xA8  Slew Rate
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_slew_rate());
            }
            break;

        case 41129: { // 0xA0 0xA9  Power
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_power());
            }
            break;

        case 41130: { // 0xA0 0xAA  Temperature
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_temperature());
            }
            break;

        case 41131: { // 0xA0 0xAB  Torque
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_torque());
            }
            break;

        case 41132: { // 0xA0 0xAC  Direction
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_direction());
            }
            break;

        case 41133: { // 0xA0 0xAD  Efficiency
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->get_efficiency());
            }
            break;

        case 41134: { // 0xA0 0xAE  is stopped
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->is_stopped());
            }
            break;

        case 41135: { // 0xA0 0xAF  is reversed
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;
                return_msg_body = std::to_string(Motors::motor_array.at(motor_number)->is_reversed());
            }
            break;

        case 41376: { // 0xA1 0xA0  is registered
                int motor_number = request.msg.at(0) - 48;
                request.msg.erase(0);

                status = 1;

                MotorThread* motor_thread = MotorThread::get_instance();
                return_msg_body = std::to_string(motor_thread->is_registered(*Motors::motor_array.at(motor_number)));
            }
            break;

        // encoder interaction post cases
        // encoder iteraction get cases

        // analog in sensor interaction post cases
        // analog in sensor interaction get cases

        // imu interaction post cases
        // imu interaction get cases

        // position tracker post cases
        // position tracker get cases

        // sd card interaction post cases

        // misc
        case 43936:  // 0xAB 0xA0  debug
            status = 1;
            return_msg_body = " debug msg received: " + request.msg;
            break;

        case 43937:  // 0xAB 0xA1  init server
            status = 1;
            pros::c::serctl(SERCTL_DISABLE_COBS, NULL);
            set_server_task_priority(TASK_PRIORITY_DEFAULT);  // more messages are sure to follow so give read task more CPU time
            delay = 10; // lower delay because of expected messages
            return_msg_body = "server is running";
            break;

        case 43938:  // 0xAB 0xA2  shutdown server
            status = 1;
            pros::c::serctl(SERCTL_ENABLE_COBS, NULL);
            set_server_task_priority(2);
            delay = 100;
            return_msg_body = "server is no longer running";
            break;

        default:
            status = 1;
            return_msg_body = " [INFO], " + std::to_string(pros::millis()) + ", Invalid Command: " + request.msg;
            break;


    }

    return_msg.push_back(return_msg_body.length() + 2);
    return_msg.push_back((char)(request.return_id >> 8) & 0xFF);
    return_msg.push_back((char)request.return_id & 0xFF);
    return_msg += return_msg_body;
    // return_msg += std::to_string(pros::millis());
    return_msg.push_back(’\xC6’);

    entry.content = return_msg;

    logger.add(entry);

    return 1;
}




void Server::start_server() {
    read_thread->resume();
}

void Server::stop_server() {
    read_thread->suspend();
}

void Server::set_server_task_priority(int new_prio) {
    read_thread->set_priority(new_prio);
}

void Server::set_debug_mode(bool debug_mode) {
    debug = debug_mode;
}

void Server::clear_stdin() {
    fflush(stdin);
    std::cin.clear();
}



int Server::handle_requests(int max_requests) {
    std::vector<server_request> requests;

    if ( !request_queue.empty() ) {
        while ( lock.exchange( true ) ); //aquire lock
        for(int i=0; i<max_requests; i++) {
            if ( !request_queue.empty() ) {
                server_request request = request_queue.front();
                request_queue.pop();

                requests.push_back(request);
            }
        }
        lock.exchange( false ); //release lock
    }

    for (int i=0; i<requests.size(); i++) {
        handle_request(requests.at(i));
    }

    return requests.size();
}
```

```
/**
 * @file: ./RobotCode/src/objects/position_tracking/PositionTracker.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * contains functions for calculating robot position
 *
 */

#ifndef __POSITIONTRACKER_HPP__
#define __POSITIONTRACKER_HPP__

#include <atomic>

#include "main.h"


#define WHEEL_TRACK_R 2.47
#define WHEEL_TRACK_L 2.47
#define S_ENC_OFFSET 3.5

typedef struct
{
    long double x_pos = 0;
    long double y_pos = 0;
    long double theta = 0;
    void print() {
        std::cout << "x pos: " << this->x_pos << "\n";
        std::cout << "y pos: " << this->y_pos << "\n";
        std::cout << "angle: " << this->theta << "\n";
    };
} position;


class PositionTracker
{
    private:
        PositionTracker();
        static PositionTracker *tracker_obj;

        static position current_position;

        static long double initial_l_enc;
        static long double initial_r_enc;
        static long double initial_theta;
        static long double imu_offset;

        static long double prev_l_enc;
        static long double prev_r_enc;
        static long double delta_theta_rad;

        static int l_id;
        static int r_id;

        static std::atomic<bool> lock;  //protect position from concurrent access

        static int log_level;
        static bool use_imu;


        static void calc_position(void*);
        pros::Task *thread;  // the thread for keeping track of position


    public:
        ~PositionTracker();

        /**
         * @return: PositionTracker -> instance of class to be used throughout program
         *
         * give the instance of the singleton class or creates it if it does
         * not yet exist
         */
        static PositionTracker* get_instance();

        static long double to_inches( long double encoder_ticks, long double wheel_size );
        static long double to_encoder_ticks(long double inches, long double wheel_size);
        static long double to_radians(long double degrees);
        static long double to_degrees(long double radians);

        /**
         * @return: None
         *
         * starts the thread or resmes it if it was stopped
         */
        void start_thread();

        /**
         * @return: None
         *
         * stops the thread from being scheduled
         */
        void stop_thread();

        void set_log_level(int log_lvl);

        void enable_imu();
        void disable_imu();

        long double get_delta_theta_rad();
        long double get_heading_rad();

        position get_position();

        static void set_position(position robot_coordinates);
};

#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/position_tracking/PositionTracker.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 * TODO:
 *
 * contains implementation for functions that track position
 */

#include <atomic>

#include "main.h"

#include "../serial/Logger.hpp"
#include "../sensors/Sensors.hpp"
#include "PositionTracker.hpp"


PositionTracker *PositionTracker::tracker_obj = NULL;
std::atomic<bool> PositionTracker::lock = ATOMIC_VAR_INIT(false);
position PositionTracker::current_position;

long double PositionTracker::initial_l_enc;
long double PositionTracker::initial_r_enc;
long double PositionTracker::initial_theta;
long double PositionTracker::imu_offset;

long double PositionTracker::prev_l_enc;
long double PositionTracker::prev_r_enc;
long double PositionTracker::delta_theta_rad;

int PositionTracker::l_id = -1;  // -1 is used as an invalid id
int PositionTracker::r_id = -1;

int PositionTracker::log_level = 0;
bool PositionTracker::use_imu = false;


PositionTracker::PositionTracker() {
    thread = new pros::Task( calc_position, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "position_tracking");
    set_position({0, 0, 0});
    thread->suspend();
}


PositionTracker::~PositionTracker() {
    thread->remove();
    delete thread;
}


/**
 * inits object if object is not already initialized based on a static bool
 * sets bool if it is not set
 */
PositionTracker* PositionTracker::get_instance() {
    if ( tracker_obj == NULL )
    {
        tracker_obj = new PositionTracker;
    }
    return tracker_obj;
}




long double PositionTracker::to_inches( long double encoder_ticks, long double wheel_size ) {
    long double circumference = (wheel_size * M_PI);
    long double revolutions = encoder_ticks / 360.0;
    long double inches = circumference * revolutions;

    return inches;
}


long double PositionTracker::to_encoder_ticks(long double inches, long double wheel_size) {
    long double circumference = (wheel_size * M_PI);
    long double revolutions = inches / circumference;
    long double encoder_ticks = revolutions * 360;

    return encoder_ticks;
}


long double PositionTracker::to_degrees(long double radians) {
    return (radians * (180 / M_PI));
}


long double PositionTracker::to_radians(long double degrees) {
    return (degrees * (M_PI / 180));
}



void PositionTracker::calc_position(void*)
{
    int s_id = Sensors::strafe_encoder.get_unique_id();

    prev_l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
    prev_r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
    long double prev_s_enc = Sensors::strafe_encoder.get_position(s_id);

    while(1)
    {
        while ( lock.exchange( true ) );

        long double l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
        long double r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
        long double s_enc = Sensors::strafe_encoder.get_position(s_id);
        // std::cout << l_enc << " " << r_enc << " " << s_enc << "\n";
        long double delta_l_in = to_inches(l_enc - prev_l_enc, 3.25);  // calculate change in each encoder in inches
        long double delta_r_in = to_inches(r_enc - prev_r_enc, 3.25);
        long double delta_s_in = to_inches(s_enc - prev_s_enc, 3.25);

        prev_l_enc = l_enc;  // update previous encoder values
        prev_r_enc = r_enc;
        prev_s_enc = s_enc;

        // calculate total change in encoders
        long double delta_l_total = to_inches(l_enc, 3.25) - to_inches(initial_l_enc, 3.25);
        long double delta_r_total = to_inches(r_enc, 3.25) - to_inches(initial_r_enc, 3.25);
        // std::cout << "encoder data: " << delta_l_total << " " << delta_r_total << " " << initial_l_enc << " " << initial_r_enc << "\n";

        // calculate absolute orientation (unbounded)
        long double encoder_reading_rad = initial_theta + ((delta_l_total - delta_r_total) / (WHEEL_TRACK_L + WHEEL_TRACK_R));  // wheel track length
        // wrap angle to [-pi, pi]
        encoder_reading_rad = std::atan2(std::sin(encoder_reading_rad), std::cos(encoder_reading_rad));


        long double new_abs_theta_rad;
        long double imu_reading_rad;
        if(use_imu) {
            imu_reading_rad = imu_offset + to_radians(Sensors::imu.get_heading());
            imu_reading_rad = std::atan2(std::sin(imu_reading_rad), std::cos(imu_reading_rad));  // wrap angle to [-pi, pi]

            // make sure that imu_reading and theta from encoders have the same sign
            // to ensure that they are telling the same reading when merging
            // ie. imu = -359, enc = 1     == bad merge
            //     imu = -10,  enc = 2     == good merge
            if(encoder_reading_rad > 0 && imu_reading_rad < 0 && std::abs(encoder_reading_rad) + std::abs(imu_reading_rad) > (M_PI / 2)) {
                imu_reading_rad += 2 * M_PI;
            } else if(encoder_reading_rad < 0 && imu_reading_rad > 0 && std::abs(encoder_reading_rad) + std::abs(imu_reading_rad) > (M_PI / 2)) {
                imu_reading_rad -= 2 * M_PI;
            }

            new_abs_theta_rad = (.7 * imu_reading_rad) + (.3 * encoder_reading_rad);  // merge with imu
        } else {
            new_abs_theta_rad = encoder_reading_rad;
        }

        // calculate the change in angle from the previous position
        delta_theta_rad = new_abs_theta_rad - current_position.theta;

        // calculate local offset
        long double delta_local_x;
        long double delta_local_y;
        if(std::abs(delta_theta_rad) < 0.000001) {
            delta_local_x = delta_s_in;
            delta_local_y = delta_r_in;  // note: delta_l == delta_r
        } else {
            delta_local_x = (2 * std::sin((delta_theta_rad / 2))) * ((delta_s_in / delta_theta_rad) + S_ENC_OFFSET);
            delta_local_y = (2 * std::sin((delta_theta_rad / 2))) * ((delta_r_in / delta_theta_rad) + WHEEL_TRACK_R);
        }

        // calculate average orientation for the cycle
        double avg_theta_rad = current_position.theta + (delta_theta_rad / 2);

        // calculate global change in coordinates as the change in the local offset
        // rotated by -(avg_theta_rad)
        // Converts to polar coordinates, changes the angle, and converts back to cartesian
        long double radius_pol = std::sqrt((std::pow(delta_local_x, 2) + std::pow(delta_local_y, 2)));
        long double theta_pol = std::atan2(delta_local_y, delta_local_x);
        theta_pol = theta_pol - avg_theta_rad;
        long double delta_global_x = radius_pol * std::cos(theta_pol);
        long double delta_global_y = radius_pol * std::sin(theta_pol);

        if (std::isnan(delta_global_x)) {
          delta_global_x = 0;
        }

        if (std::isnan(delta_global_y)) {
          delta_global_y = 0;
        }

        if (std::isnan(new_abs_theta_rad)) {
          new_abs_theta_rad = 0;
        }

        // don't use built in method to update position because that resets encoders, which is not necessary
        current_position.x_pos = current_position.x_pos + delta_global_x;
        current_position.y_pos = current_position.y_pos + delta_global_y;
        current_position.theta = new_abs_theta_rad;


        Logger logger;
        log_entry entry;

        for(int i = 0; i <= log_level; i++) {
            switch(i) {
                case 0:
                    entry.content = "";
                    break;
                case 1:
                    entry.content += ("[INFO], " + std::string("Position Tracking Data")
                        + ", Time: " + std::to_string(pros::millis())
                        + ", X_POS: " + std::to_string(current_position.x_pos)
                        + ", Y_POS: " + std::to_string(current_position.y_pos)
                        + ", Angle: " + std::to_string(to_degrees(current_position.theta))
                    );
                    break;
                case 2:
                    entry.content += (
                        "angle_from_imu_radians: " + std::to_string(imu_reading_rad)
                        + "angle_from_encoders_radians: " + std::to_string(encoder_reading_rad)
                        + "angle_from_imu_degrees: " + std::to_string(to_degrees(imu_reading_rad))
                        + "angle_from_encoders_degrees: " + std::to_string(to_degrees(encoder_reading_rad))
                    );
                    break;
                case 3:
                    entry.content += (
                        "local_delta_y: " + std::to_string(delta_local_y)
                        + "local_delta_x: " + std::to_string(delta_local_x)
                        + "global_delta_y: " + std::to_string(delta_global_y)
                        + "global_delta_x: " + std::to_string(delta_global_x)
                    );
                    break;
                case 4:
                    entry.content += (
                        "l_enc: " + std::to_string(l_enc)
                        + "r_enc: " + std::to_string(r_enc)
                        + "s_enc: " + std::to_string(s_enc)
                        + "delta_l_enc_in: " + std::to_string(delta_l_in)
                        + "delta_r_enc_in: " + std::to_string(delta_r_in)
                        + "delta_s_enc_in: " + std::to_string(delta_s_in)
                    );
                    break;
                case 5:
                    if(use_imu) {
                        entry.content += (
                            "imu_reading: " + std::to_string(Sensors::imu.get_heading())
                            + "imu_offset: " + std::to_string(imu_offset)
                        );
                    }
                    break;
            }
        }

        entry.stream = "clog";
        if(!entry.content.empty()) {
            logger.add(entry);
        }

        lock.exchange(false);

        pros::delay(5);
    }
}




void PositionTracker::start_thread()
{
    thread->resume();
}

void PositionTracker::stop_thread()
{
    thread->suspend();
}




void PositionTracker::set_log_level(int log_lvl) {
    while ( lock.exchange( true ) );
    log_level = log_lvl;
    lock.exchange(false);
}

void PositionTracker::enable_imu() {
    while ( lock.exchange( true ) );
    use_imu = true;
    lock.exchange(false);
}

void PositionTracker::disable_imu() {
    while ( lock.exchange( true ) );
    use_imu = false;
    lock.exchange(false);
}


long double PositionTracker::get_delta_theta_rad() {
    while ( lock.exchange( true ) );
    long double d_theta_rad = delta_theta_rad;
    lock.exchange(false);

    return d_theta_rad;
}

long double PositionTracker::get_heading_rad() {
    while ( lock.exchange( true ) );
    long double heading = current_position.theta;
    lock.exchange(false);
    return heading;
}

position PositionTracker::get_position()
{
    while ( lock.exchange( true ) );
    position pos;
    pos.x_pos = current_position.x_pos;
    pos.y_pos = current_position.y_pos;
    pos.theta = current_position.theta;
    lock.exchange(false);

    return pos;
}




void PositionTracker::set_position(position robot_coordinates)
{
    while ( lock.exchange( true ) );

    if(l_id != -1) {
        Sensors::left_encoder.forget_position(l_id);
    }
    if(r_id != -1) {
        Sensors::right_encoder.forget_position(r_id);
    }
    l_id = Sensors::left_encoder.get_unique_id(true);
    r_id = Sensors::right_encoder.get_unique_id(true);

    initial_l_enc = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
    initial_r_enc = std::get<1>(Sensors::get_average_encoders(l_id, r_id));
    initial_theta = robot_coordinates.theta;

    if(use_imu) {
        imu_offset = initial_theta - to_radians(Sensors::imu.get_heading());  // offset + imu_reading = initial_theta
    } else {
        imu_offset = initial_theta;
    }

    prev_l_enc = initial_l_enc;
    prev_r_enc = initial_r_enc;

    delta_theta_rad = 0;

    current_position = robot_coordinates;

    lock.exchange(false);
}
```

```
/**
 * @file: ./RobotCode/src/objects/subsystems/Indexer.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * Contains class for the differential subsystem
 * has methods for brake and indexing
 */

#ifndef __INDEXER_HPP__
#define __INDEXER_HPP__

#include <tuple>
#include <queue>

#include "main.h"

#include "../motors/Motor.hpp"
#include "../sensors/Sensors.hpp"
#include "../sensors/BallDetector.hpp"



struct ball_positions{
    bool top;
    bool middle;
    std::string middle_color;
    bool operator== (const ball_positions &state) {
        bool top_equal = (top == state.top);
        bool middle_equal = (middle == state.middle);
        bool middle_color_equal;
        if(middle_color == "present" && state.middle_color != "none") {
            middle_color_equal = true;
        } else if (middle_color == "any") {
            middle_color_equal = true;
        }else if (middle_color == "red" && state.middle_color == "red") {
            middle_color_equal = true;
        } else if (middle_color == "blue" && state.middle_color == "blue") {
            middle_color_equal = true;
        } else if (middle_color == "none" && state.middle_color == "none") {
            middle_color_equal = true;
        }

        return (top_equal && middle_equal && middle_color_equal);
    }
    bool operator!= (const ball_positions &state) {
        bool top_equal = (top == state.top);
        bool middle_equal = (middle == state.middle);
        bool middle_color_equal;
        if(middle_color == "any" && state.middle_color != "none") {
            middle_color_equal = true;
        } else if (middle_color == "red" && state.middle_color == "red") {
            middle_color_equal = true;
        } else if (middle_color == "blue" && state.middle_color == "blue") {
            middle_color_equal = true;
        } else if (middle_color == "none" && state.middle_color == "none") {
            middle_color_equal = true;
        }

        return !(top_equal && middle_equal && middle_color_equal);
    }
};

typedef enum e_indexer_command {
    e_index,
    e_filter,
    e_auto_index,
    e_index_no_backboard,
    e_index_until_filtered,
    e_increment,
    e_auto_increment,
    e_index_to_state,
    e_fix_ball,
    e_run_upper,
    e_run_lower,
    e_stop
} indexer_command;

typedef struct {
    ball_positions end_state;
    bool allow_filter;
}indexer_args;

typedef struct {
    int uid;
    indexer_command command;
    indexer_args args;
} indexer_action;

/**
 * @see: Motors.hpp
 *
 * contains methods to allow for control of the indexer
 */
class Indexer
{
    private:
        static Motor *upper_indexer;
        static Motor *lower_indexer;
        static BallDetector *ball_detector;
        static std::string filter_color;

        static int num_instances;

        pros::Task *thread;  // the motor thread
        static std::queue<indexer_action> command_queue;
        static std::vector<int> commands_finished;
        static std::atomic<bool> command_start_lock;
        static std::atomic<bool> command_finish_lock;

        int send_command(indexer_command command, indexer_args args={});

        static bool auto_filter_ball();
        static void indexer_motion_task(void*);

    public:
        Indexer(Motor &upper, Motor &lower, BallDetector &detector, std::string color);
        ~Indexer();

        void index();
        void filter();
        void auto_index();
        void index_no_backboard();
        int index_until_filtered(bool asynch=false);
        int index_to_state(bool allow_filter, ball_positions end_state, bool asynch=false);

        void increment();
        void auto_increment();

        void run_upper_roller();
        void run_lower_roller();

        int fix_ball(bool asynch=true);

        void hard_stop();
        void stop();

        static ball_positions get_state();

        void reset_command_queue();
        void update_filter_color(std::string new_color);

        void wait_until_finished(int uid);
        bool is_finished(int uid);


};




#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/subsystems/Indexer.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * Contains implementation for the differential subsystem
 * has methods for brake and indexing
 */

#include "main.h"


#include "../serial/Logger.hpp"
#include "../sensors/BallDetector.hpp"
#include "Indexer.hpp"

int Indexer::num_instances = 0;
std::queue<indexer_action> Indexer::command_queue;
std::vector<int> Indexer::commands_finished;
std::atomic<bool> Indexer::command_start_lock = ATOMIC_VAR_INIT(false);
std::atomic<bool> Indexer::command_finish_lock = ATOMIC_VAR_INIT(false);

Motor* Indexer::upper_indexer;
Motor* Indexer::lower_indexer;
BallDetector* Indexer::ball_detector;
std::string Indexer::filter_color;


Indexer::Indexer(Motor &upper, Motor &lower, BallDetector &detector, std::string color)
{
    upper_indexer = &upper;
    lower_indexer = &lower;
    ball_detector = &detector;
    filter_color = color;

    upper_indexer->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    lower_indexer->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    upper_indexer->set_motor_mode(e_voltage);
    lower_indexer->set_motor_mode(e_voltage);

    upper_indexer->disable_slew();
    lower_indexer->disable_slew();

    if(num_instances == 0 || thread == NULL) {
        thread = new pros::Task( indexer_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "indexer_thread");
    }

    num_instances += 1;
}


Indexer::˜Indexer() {
    num_instances -= 1;
    if(num_instances == 0) {
        delete thread;
    }
}




bool Indexer::auto_filter_ball() {
    int color = ball_detector->check_filter_level();
    if((color == 1 && filter_color == "blue") || (color == 2 && filter_color == "red")) {  // ball should be filtered
        upper_indexer->set_voltage(-12000);
        lower_indexer->set_voltage(12000);
        pros::delay(250);  // let ball filter out
        upper_indexer->set_voltage(0);
        lower_indexer->set_voltage(0);

        return true;  // did filter out ball
    } else if(color < 0) {  // ball was detected but color could not be determined: print error message and default to intaking
        Logger logger;
        log_entry entry;
        entry.content = "[ERROR], " + std::to_string(pros::millis()) + ", ball was detected but color could not be determined";
        entry.stream = "cerr";
        logger.add(entry);
    }

    return false;  // did not filter out ball
}




void Indexer::indexer_motion_task(void*) {
    while(1) {
        if(command_queue.empty()) {  // delay unitl there is a command in the queue
            pros::delay(5);
            continue;
        }

        // take lock and get command
        while ( command_start_lock.exchange( true ) ); //aquire lock
        indexer_action action = command_queue.front();
        command_queue.pop();
        command_start_lock.exchange( false ); //release lock

        // execute command
        switch(action.command) {
            case e_filter: {
                upper_indexer->set_voltage(-12000);
                lower_indexer->set_voltage(12000);
                break;
            } case e_auto_index: {
                auto_filter_ball();
                // fallthrough and index like normal now that it doesn’t need to filter
            } case e_index: {
                upper_indexer->set_voltage(12000);
                lower_indexer->set_voltage(12000);
                break;
            } case e_index_no_backboard: {
                upper_indexer->set_voltage(9000);
                lower_indexer->set_voltage(12000);
                break;
            } case e_index_until_filtered: {
                upper_indexer->set_voltage(12000);
                lower_indexer->set_voltage(12000);

                bool filtered = false;
                do {
                    filtered = auto_filter_ball();
                } while(!filtered);

                break;
            } case e_index_to_state: {
                ball_positions current_state = get_state();
                do {
                    current_state = get_state();
                    int color = ball_detector->check_filter_level();
                    if(action.args.allow_filter) {
                        auto_filter_ball();  // attempt to filter
                    }

                    if(current_state.top != action.args.end_state.top) {
                        upper_indexer->set_voltage(12000);
                    }

                    if(current_state.middle != action.args.end_state.middle) {
                        lower_indexer->set_voltage(12000);
                    }
                } while(current_state != action.args.end_state);

                break;
            } case e_auto_increment: {
                // try to filter out ball at second level if necessary
                auto_filter_ball();
                // fall through if there is nothing to filter out
            } case e_increment: {
                std::vector<bool> locations = ball_detector->locate_balls();

                if(!locations.at(0)) {  // move ball into top position
                    upper_indexer->set_voltage(7500);
                    lower_indexer->set_voltage(10500);
                } else if(locations.at(0) && !locations.at(1)) { // move ball from lowest/no position to middle position
                    upper_indexer->set_voltage(0);
                    lower_indexer->set_voltage(10500);
                } else { // indexer can't do anything to increment so don't run
                    upper_indexer->set_voltage(0);
                    lower_indexer->set_voltage(0);
                }
                break;
            } case e_fix_ball: {
                upper_indexer->set_voltage(-12000);
                pros::delay(250);
                upper_indexer->set_voltage(12000);
                pros::delay(500);
                upper_indexer->set_voltage(0);
                break;
            } case e_run_upper: {
                upper_indexer->set_voltage(12000);
                break;
            } case e_run_lower: {
                lower_indexer->set_voltage(12000);
                break;
            } case e_stop: {
                lower_indexer->set_voltage(0);
                upper_indexer->set_voltage(0);
                break;
            }
        }

        if(action.command == e_index_until_filtered || action.command == e_index_to_state || action.command == e_fix_ball) {
            while ( command_finish_lock.exchange( true ) ); //aquire lock
            commands_finished.push_back(action.uid);
            command_finish_lock.exchange( false ); //release lock
        }
    }
}

int Indexer::send_command(indexer_command command, indexer_args args /*{}*/) {
    while ( command_start_lock.exchange( true ) ); //aquire lock
    indexer_action action;
    action.command = command;
    action.args = args;
    action.uid = pros::millis() + lower_indexer->get_actual_voltage() + upper_indexer->get_actual_voltage();
    command_queue.push(action);
    command_start_lock.exchange( false ); //release lock

    return action.uid;
}

void Indexer::index() {
    send_command(e_index);
}

void Indexer::filter() {
    send_command(e_filter);
}

void Indexer::auto_index() {
    send_command(e_auto_index);
}

void Indexer::index_no_backboard() {
    send_command(e_index_no_backboard);
}

int Indexer::index_until_filtered(bool asynch /*false*/) {
    int uid = send_command(e_index_until_filtered);

    if(!asynch) {
        wait_until_finished(uid);
    }
```

```cpp

    return uid;
}

int Indexer::index_to_state(bool allow_filter, ball_positions end_state, bool asynch) {
    indexer_args args;
    args.allow_filter = allow_filter;
    args.end_state = end_state;
    int uid = send_command(e_index_to_state, args);

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}

void Indexer::increment() {
    send_command(e_increment);
}

void Indexer::auto_increment() {
    send_command(e_auto_increment);
}



void Indexer::run_upper_roller() {
    send_command(e_run_upper);
}


void Indexer::run_lower_roller() {
    send_command(e_run_lower);
}



int Indexer::fix_ball(bool asynch /*true*/) {
    int uid = send_command(e_fix_ball);

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



void Indexer::hard_stop() {
    reset_command_queue();
    send_command(e_stop);
}

void Indexer::stop() {
    send_command(e_stop);
}

ball_positions Indexer::get_state() {
    ball_positions state;

    int color = ball_detector->check_filter_level();
    std::vector<bool> ball_locations = ball_detector->locate_balls();
    if(ball_locations.at(0)) {
        state.top = true;
    } else {
        state.top = false;
    }
    if(ball_locations.at(1)) {
        state.middle = true;
    } else {
        state.middle = false;
    }

    if (color == 0) {
        state.middle_color = "none";
    } else if(color == 1) {
        state.middle_color = "blue";
    } else if (color == 2) {
        state.middle_color = "red";
    } else {
        state.middle_color = "unknown";
    }

    return state;
}

void Indexer::reset_command_queue() {
    while ( command_start_lock.exchange( true ) ); //aquire lock
    std::queue<indexer_action> empty_queue;
    std::swap( command_queue, empty_queue );  // replace command queue with an empty queue
    command_start_lock.exchange( false ); //release lock
}


void Indexer::update_filter_color(std::string new_color) {
    filter_color = new_color;
}


void Indexer::wait_until_finished(int uid) {
    while(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
        pros::delay(10);
    }
    while ( command_finish_lock.exchange( true ) ); //aquire lock
    commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
    command_finish_lock.exchange( false ); //release lock
}


bool Indexer::is_finished(int uid) {
    if(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
        while ( command_finish_lock.exchange( true ) ); //aquire lock
        commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
        command_finish_lock.exchange( false ); //release lock

        return false;  // command is not finished because it is not in the list
    }
    return true;
}
```

```cpp
/**
 * @file: ./RobotCode/src/objects/subsystems/chassis.hpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * Contains class for the chassis subsystem
 * has methods for driving during autonomous including turning and driving straight
 */

#ifndef __CHASSIS_HPP__
#define __CHASSIS_HPP__

#include <tuple>
#include <queue>

#include "main.h"

#include "../motors/Motor.hpp"
#include "../sensors/Sensors.hpp"


std::vector<double> generate_velocity_profile(int encoder_ticks, const std::function<double(double)>& max_acceleration, double max_decceleration, double max_velocity, double initial_velocity);


typedef enum {
    e_pid_straight_drive,
    e_profiled_straight_drive,
    e_turn,
    e_drive_to_point,
    e_turn_to_point,
    e_turn_to_angle
} chassis_commands;

typedef struct {
    long double x;
    long double y;
    long double dx;
    long double dy;
    long double radius;
    long double dtheta;
    std::string get_string() {
        std::string str = (
            + "{x: " + std::to_string(this->x)
            + " y: " + std::to_string(this->y)
            + " dx: " + std::to_string(this->dx)
            + " dy: " + std::to_string(this->dy)
            + " radius: " + std::to_string(this->radius)
            + " dtheta: " + std::to_string(this->dtheta)
            + "}"
        );
        return str;
    }
} waypoint;

typedef struct {
    double setpoint1=0;
    double setpoint2=0;
    double kP=1;
    double kI=.001;
    double kD=.001;
    double I_max=INT32_MAX;
    int max_velocity=150;
    int timeout=INT32_MAX;
    int recalculations=0;
    int explicit_direction=0;
    double motor_slew=INT32_MAX;
    bool correct_heading=true;
    bool log_data=false;
} chassis_params;

typedef struct {
    chassis_params args;
    int command_uid;
    chassis_commands command;
} chassis_action;


/**
 * @see: Motors.hpp
 *
 * contains methods to allow for easy control of the robot during
 * the autonomous period
 */
class Chassis
{
    private:
        static Motor *front_left_drive;
        static Motor *front_right_drive;
        static Motor *back_left_drive;
        static Motor *back_right_drive;

        static Encoder* left_encoder;
        static Encoder* right_encoder;

        pros::Task *thread;  // the motor thread
        static std::queue<chassis_action> command_queue;
        static std::vector<int> commands_finished;
        static std::atomic<bool> command_start_lock;
        static std::atomic<bool> command_finish_lock;
        static int num_instances;

        static void t_pid_straight_drive(chassis_params args);  // functions called by thread for asynchronous movement
        static void t_profiled_straight_drive(chassis_params args);
        static void t_turn(chassis_params args);
        static void t_move_to_waypoint(chassis_params args, waypoint point);

        static double wheel_diameter;
        static double width;
        static double gear_ratio;

        static void chassis_motion_task(void*);

    public:
        Chassis( Motor &front_left, Motor &front_right, Motor &back_left, Motor &back_right, Encoder &l_encoder, Encoder &r_encoder, double chassis_width, double gearing=1, double wheel_size=4.05);
        ˜Chassis();

        int pid_straight_drive(double encoder_ticks, int relative_heading=0, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, bool correct_heading=true, double slew=0.2, bool log_data=true);
        int profiled_straight_drive(double encoder_ticks, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, bool correct_heading=true, int relative_heading=0, bool log_data=true);
        int uneven_drive(double l_enc_ticks, double r_enc_ticks, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=10, bool log_data=false);
        int turn_right(double degrees, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=15, bool log_data=true);
        int turn_left(double degrees, int max_velocity=450, int timeout=INT32_MAX, bool asynch=false, double slew=15, bool log_data=true);
        int drive_to_point(double x, double y, int recalculations=0, int explicit_direction=0, int max_velocity=450, int timeout=INT32_MAX, bool correct_heading=true, bool asynch=false, double slew=10, bool log_data=true);
        int turn_to_point(double x, double y, int max_velocity=450, int timeout=INT32_MAX, bool asynch = false, double slew=10, bool log_data=true);
        int turn_to_angle(double theta, int max_velocity=450, int timeout=INT32_MAX, bool asynch = false, double slew=10, bool log_data=true);

        /**
         * @param: int voltage -> the voltage on interval [-127, 127] to set the motor to
         * @return: None
         *
         * sets voltage of chassis
         */
        void move( int voltage );

        /**
         * @param: pros::motor_brake_mode_e_t new_brake_mode -> the new brakemode for the chassis
         * @return: None
         *
         * sets brake mode of all motors
         */
        void set_brake_mode( pros::motor_brake_mode_e_t new_brake_mode );



        /**
         * @return: None
         *
         * @see: Motors.hpp
         *
         * changes the direction at the api motor level so that all the
         * motors in the chassis system are reversed
         * useful for allowing to change direction of drive in user control
         */
        void change_direction();

        /**
         * @param: int speed -> the new speed the slew rate controller
         * @return: None
         *
         * sets the internal slew rate of the motor and enables it
         */
        void enable_slew( int rate=120 );

        /**
         * @return: None
         *
         * disables internal slew rate of the motor
         */
        void disable_slew( );

        void wait_until_finished(int uid);
        bool is_finished(int uid);


};


#endif
```

```cpp
/**
 * @file: ./RobotCode/src/objects/subsystems/chassis.cpp
 * @author: Aiden Carney
 * @reviewed_on: 2/16/2020
 * @reviewed_by: Aiden Carney
 *
 * @see: chassis.hpp
 *
 * contains implementation for chassis subsytem class
 */

#include <cmath>
#include <algorithm>
#include <deque>
#include <type_traits>

#include "main.h"

#include "../serial/Logger.hpp"
#include "../position_tracking/PositionTracker.hpp"
#include "chassis.hpp"



std::vector<double> generate_velocity_profile(int encoder_ticks, const std::function<double(double)>& max_acceleration, double max_decceleration, double max_velocity, double initial_velocity) {
    if(encoder_ticks <= 0) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[ERROR] " + std::string("PROFILE_CALCULATION")
            + ", Time: " + std::to_string(pros::millis())
            + ", Could not generate profile with negative or 0 encoder ticks"
            + ", enc_ticks: " + std::to_string(encoder_ticks)
            + ", max_velocity: " + std::to_string(max_velocity)
        );
        entry.stream = "clog";
        logger.add(entry);
        pros::delay(100); // add delay for msg to be logged
        throw std::invalid_argument("Cannot generate profile with negative or 0 encoder ticks");
    }

    std::vector<double> profile = {initial_velocity};

    int i = 0;
    while(i < encoder_ticks) {
        int ticks_left = encoder_ticks - i;
        int ticks_to_deccelerate = profile.at(i) / max_decceleration;
        if(ticks_to_deccelerate < ticks_left) {
            double step = profile.at(i) + max_acceleration(i);
            if(step > max_velocity) {
                step = max_velocity;
            }
            profile.push_back(step);
        } else {
            profile.push_back(profile.at(i) - max_decceleration);
        }

        i += 1;
    }

    return profile;
}



int Chassis::num_instances = 0;
std::queue<chassis_action> Chassis::command_queue;
std::vector<int> Chassis::commands_finished;
std::atomic<bool> Chassis::command_start_lock = ATOMIC_VAR_INIT(false);
std::atomic<bool> Chassis::command_finish_lock = ATOMIC_VAR_INIT(false);

Motor* Chassis::front_left_drive;
Motor* Chassis::front_right_drive;
Motor* Chassis::back_left_drive;
Motor* Chassis::back_right_drive;

Encoder* Chassis::left_encoder;
Encoder* Chassis::right_encoder;
double Chassis::width;
double Chassis::gear_ratio;
double Chassis::wheel_diameter;


Chassis::Chassis( Motor &front_left, Motor &front_right, Motor &back_left, Motor &back_right, Encoder &l_encoder, Encoder &r_encoder, double chassis_width, double gearing /*1*/, double wheel_size /*4.05*/)
{
    front_left_drive = &front_left;
    front_right_drive = &front_right;
    back_left_drive = &back_left;
    back_right_drive = &back_right;

    left_encoder = &l_encoder;
    right_encoder = &r_encoder;

    wheel_diameter = wheel_size;
    gear_ratio = gearing;
    width = chassis_width;

    if(num_instances == 0 || thread == NULL) {
        thread = new pros::Task( chassis_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "chassis_thread");
    }

    num_instances += 1;

    front_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    front_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    front_left_drive->set_motor_mode(e_voltage);
    front_right_drive->set_motor_mode(e_voltage);
    back_left_drive->set_motor_mode(e_voltage);
    back_right_drive->set_motor_mode(e_voltage);

    front_left_drive->disable_slew();
    front_right_drive->disable_slew();
    back_left_drive->disable_slew();
    back_right_drive->disable_slew();
}




Chassis::~Chassis()
{
    num_instances -= 1;
    if(num_instances == 0) {
        delete thread;
    }
}


// void Chassis::generate_profiles() {
//     if(!profile_1.is_generated()) {
//         profile_1.generate_profile(
//             [](double n) -> double { return 183335300 + ((16.29262 - 183335300) / (1 + std::pow((n / 5807375), 1.381135))); },
//             [](double n) -> double { return 201.3993 - (0.07738232 * n) - (0.0001796556 * std::pow(n, 2)); },
//             // [](double n) -> double { return (201.3993 - (0.0967279 * n) - (0.0002807118 * std::pow(n, 2))); },
//             // [](double n) -> double { return -14.87879 + ((199.9947 - -14.87879) / (1 + std::pow((n / 216.5658), 1.756448))); },  // 700 endpoint
//             // [](double n) -> double { return -718.7411 + ((200.0491 - -718.7411) / (1 + std::pow((n / 3436.08), 0.8049193))); },  // 700 endpoint
//             // [](double n) -> double { return -39.01548 + ((195.369 - -39.01548) / (1 + std::pow((n / 978.0565), 2.266577))); },  // 1700 endpoint
//             // [](double n) -> double { return (-0.2 * n + 1000); },  // 1000 endpoint
//             250,
//             820,
//             200,
//             0
//         );
//     }
// }


void Chassis::chassis_motion_task(void*) {
    while(1) {
        if(command_queue.empty()) {  // delay unitl there is a command in the queue
            pros::delay(10);
            continue;
        }

        // take lock and get command
        while ( command_start_lock.exchange( true ) ); //aquire lock
        chassis_action action = command_queue.front();
        command_queue.pop();
        command_start_lock.exchange( false ); //release lock

        // execute command
        switch(action.command) {
            case e_pid_straight_drive:
                t_pid_straight_drive(action.args);
                break;
            case e_profiled_straight_drive:
                t_profiled_straight_drive(action.args);
                break;
            case e_turn:
                t_turn(action.args);
                break;
            case e_drive_to_point: {
                PositionTracker* tracker = PositionTracker::get_instance();
                std::vector<waypoint> waypoints;  // calculate waypoints based on starting position

                long double dx = action.args.setpoint1 - tracker->get_position().x_pos;
                long double dy = action.args.setpoint2 - tracker->get_position().y_pos;
                std::cout << tracker->get_position().x_pos << " " << tracker->get_position().y_pos << "\n";
                // convert end coordinates to polar and then calculate waypoints
                long double delta_radius_polar = std::sqrt((std::pow(dx, 2) + std::pow(dy, 2)));
                long double delta_theta_polar = std::atan2(dy, dx);

                for(int i=action.args.recalculations + 1; i > 0; i--) {  // calculate additional waypoints, start with last endpoint and go down
                    long double radius = (i * delta_radius_polar) / (action.args.recalculations + 1);
                    waypoint recalc_point;
                    recalc_point.x = tracker->get_position().x_pos + (radius * std::cos(delta_theta_polar));  // intital x + dx
                    recalc_point.y = tracker->get_position().y_pos + (radius * std::sin(delta_theta_polar));  // initial y + dy
                    std::cout << radius << " " << delta_theta_polar << " " << (radius * std::cos(delta_theta_polar)) << " " << (radius * std::sin(delta_theta_polar)) << "\n";
                    recalc_point.dx = radius * std::cos(delta_theta_polar);
                    recalc_point.dy = radius * std::sin(delta_theta_polar);
                    recalc_point.radius = radius;
                    recalc_point.dtheta = delta_theta_polar;
                    waypoints.insert(waypoints.begin(), recalc_point);
                }
                std::cout << "\n\n\n\n\n";
                if(action.args.log_data) {
                    Logger logger;
                    log_entry entry;
                    std::string msg = (
                        "[INFO] " + std::string("CHASSIS_ODOM")
                        + ", Time: " + std::to_string(pros::millis())
                        + ", dx: " + std::to_string(dx)
                        + ", dy: " + std::to_string(dy)
                        + ", delta_theta_polar: " + std::to_string(delta_theta_polar)
                        + ", current x: " + std::to_string(tracker->get_position().x_pos)
                        + ", current y: " + std::to_string(tracker->get_position().y_pos)
                        + ", current theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
                    );
                    int i = 0;
                    for(waypoint point : waypoints) {  // add waypoints to debug message
                        msg += ", waypoint " + std::to_string(i) + ": " + point.get_string();
                    }
                    entry.content = msg;
                    entry.stream = "clog";
                    logger.add(entry);
                }

                int start = pros::millis();
                for(waypoint point : waypoints) {  // move to each generated waypoint
                    if(pros::millis() - start > action.args.timeout) {  // end early if past the timeout point
                        break;
                    }
                    t_move_to_waypoint(action.args, point);
                }
                break;
            } case e_turn_to_point: {
                PositionTracker* tracker = PositionTracker::get_instance();

                long double dx = action.args.setpoint1 - tracker->get_position().x_pos;
                long double dy = action.args.setpoint2 - tracker->get_position().y_pos;

                // convert end coordinates to polar to find the change in angle
                // long double dtheta = std::fmod((-M_PI / 2) + std::atan2(dy, dx), (2 * M_PI));
                long double dtheta = std::atan2(dy, dx);
                if(dtheta < 0) {  // map to [0, 2pi]
                    dtheta += 2 * M_PI;
                }

                // current angle is bounded by [-pi, pi] re map it to [0, 2pi]
                long double current_angle = tracker->get_heading_rad();
                if(current_angle < 0) {
                    current_angle += 2 * M_PI;
                }
                current_angle = (-current_angle) + (M_PI / 2);

                // calculate how much the robot needs to turn to be at the angle
                long double to_turn_face_forwards = current_angle - dtheta;  // change in robot angle
                long double to_turn_face_backwards = (current_angle - dtheta) - M_PI;

                if(to_turn_face_forwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
                    to_turn_face_forwards = (-2 * M_PI) + to_turn_face_forwards;  // give negative value to turn left to point
                } else if(to_turn_face_forwards < -M_PI) {
                    to_turn_face_forwards = (2 * M_PI) + to_turn_face_forwards;  // give positive value to turn left to point
                }

                if(to_turn_face_backwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
                    to_turn_face_backwards = (-2 * M_PI) + to_turn_face_backwards;  // give negative value to turn left to point
                } else if(to_turn_face_backwards < -M_PI) {
                    to_turn_face_backwards = (2 * M_PI) + to_turn_face_backwards;  // give positive value to turn left to point
                }


                long double to_turn;
                int direction;
                if(action.args.explicit_direction == 1) {  // force positive direction
                    to_turn = to_turn_face_forwards;
                    direction = 1;
                } else if(action.args.explicit_direction == -1) {  // force negative direction
                    to_turn = to_turn_face_backwards;
                    direction = -1;
                } else if(std::abs(to_turn_face_forwards) < std::abs(to_turn_face_backwards)) {  // faster to go forwards
                    to_turn = to_turn_face_forwards;
                    direction = 1;
                } else {  // faster to go backwards
                    to_turn = to_turn_face_backwards;
                    direction = -1;
                }

                to_turn = tracker->to_degrees(to_turn);


                // set up turn
                chassis_params turn_args;
                turn_args.setpoint1 = to_turn;
                turn_args.max_velocity = action.args.max_velocity;
                turn_args.timeout = 15000; // TODO: add time estimation
                turn_args.kP = 2.8;
                turn_args.kI = 0.0005;
                turn_args.kD = 50;
                turn_args.I_max = INT32_MAX;
                turn_args.motor_slew = action.args.motor_slew;
                turn_args.log_data = action.args.log_data;

                // perform turn
                std::cout << "starting turn\n";
                std::cout << to_turn << "\n";
                t_turn(turn_args);
                std::cout << "turn done\n";

                if(action.args.log_data) {
                    Logger logger;
                    log_entry entry;
                    entry.content = (
                        "[INFO] " + std::string("CHASSIS_ODOM")
                        + ", Time: " + std::to_string(pros::millis())
                        + ", X " + std::to_string(action.args.setpoint1)
                        + ", Y " + std::to_string(action.args.setpoint2)
                        + ", ToTurnForwards: " + std::to_string(tracker->to_degrees(to_turn_face_forwards))
                        + ", ToTurnBackwards: " + std::to_string(tracker->to_degrees(to_turn_face_backwards))
                        + ", ToTurn: " + std::to_string(to_turn)
                        + ", Direction: " + std::to_string(direction)
                        + ", dx: " + std::to_string(dx)
                        + ", dy: " + std::to_string(dy)
                        + ", X: " + std::to_string(tracker->get_position().x_pos)
                        + ", Y: " + std::to_string(tracker->get_position().y_pos)
                        + ", Theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
                    );
                    entry.stream = "clog";
                    logger.add(entry);
                }

                pros::delay(100);  // add delay for extra settling

                break;
            } case e_turn_to_angle: {
                PositionTracker* tracker = PositionTracker::get_instance();

                // current angle is bounded by [-pi, pi] re map it to [0, pi]
                long double current_angle = tracker->get_heading_rad();
                if(current_angle < -M_PI) {
                    current_angle += M_PI;
                }

                // calculate how much the robot needs to turn to be at the angle
                long double to_turn = action.args.setpoint1 - current_angle;  // change in robot angle

```

```
                if(to_turn > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
                    to_turn = (-2 * M_PI) + to_turn;  // give negative value to turn left to point
                } else if(to_turn < -M_PI) {
                    to_turn = (2 * M_PI) + to_turn;  // give positive value to turn left to point
                }
                to_turn = tracker->to_degrees(to_turn);

                std::cout << current_angle << " " << to_turn << "\n";

                // set up turn
                chassis_params turn_args;
                turn_args.setpoint1 = to_turn;
                turn_args.max_velocity = action.args.max_velocity;
                turn_args.timeout = action.args.timeout; // TODO: add time estimation
                turn_args.kP = 2.8;
                turn_args.kI = 0.0005;
                turn_args.kD = 50;
                turn_args.I_max = INT32_MAX;
                turn_args.motor_slew = action.args.motor_slew;
                turn_args.log_data = action.args.log_data;

                if(action.args.log_data) {
                    Logger logger;
                    log_entry entry;
                    std::string msg = (
                        "[INFO] " + std::string("CHASSIS_ODOM")
                        + ", Time: " + std::to_string(pros::millis())
                        + ", turning: " + std::to_string(to_turn)
                        + ", Current re-bounded angle: " + std::to_string(tracker->to_degrees(current_angle))
                        + ", Current angle: " + std::to_string(tracker->to_degrees(tracker->get_heading_rad()))
                    );
                    entry.content = msg;
                    entry.stream = "clog";
                    logger.add(entry);
                }

                // perform turn
                t_turn(turn_args);

                break;
            }
        }

        while ( command_finish_lock.exchange( true ) ); //aquire lock
        commands_finished.push_back(action.command_uid);
        command_finish_lock.exchange( false ); //release lock
    }
}




void Chassis::t_pid_straight_drive(chassis_params args) {
    PositionTracker* tracker = PositionTracker::get_instance();
    Configuration* config = Configuration::get_instance();

    double kP_l = args.kP;
    double kI_l = args.kI;
    double kD_l = args.kD;
    double I_max_l = args.I_max;

    double kP_r = kP_l;
    double kI_r = kI_l;
    double kD_r = kD_l;
    double I_max_r = I_max_l;

    front_left_drive->disable_driver_control();
    front_right_drive->disable_driver_control();
    back_left_drive->disable_driver_control();
    back_right_drive->disable_driver_control();

    front_left_drive->set_motor_mode(e_builtin_velocity_pid);
    front_right_drive->set_motor_mode(e_builtin_velocity_pid);
    back_left_drive->set_motor_mode(e_builtin_velocity_pid);
    back_right_drive->set_motor_mode(e_builtin_velocity_pid);

    int r_id = right_encoder->get_unique_id(true);
    int l_id = left_encoder->get_unique_id(true);

    double integral_l = 0;
    double integral_r = 0;
    double prev_error_l = 0;
    double prev_error_r = 0;
    double prev_velocity_l = 0;
    double prev_velocity_r = 0;

    double prev_l_encoder = std::get<0>(Sensors::get_average_encoders(l_id, r_id));
    double prev_r_encoder = std::get<1>(Sensors::get_average_encoders(l_id, r_id));

    long double relative_angle = 0;
    long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
    long double prev_abs_angle = abs_angle;
    long double integral_heading = 0;
    double prev_heading_error = 0;

    bool settled = false;
    std::vector<double> previous_l_velocities;
    std::vector<double> previous_r_velocities;
    int velocity_history = 15;
    bool use_integral_l = true;
    bool use_integral_r = true;

    int current_time = pros::millis();
    int start_time = current_time;

    do {
        int dt = pros::millis() - current_time;
        // pid distance controller
        double error_l = args.setpoint1 - std::get<0>(Sensors::get_average_encoders(l_id, r_id));
        double error_r = args.setpoint2 - std::get<1>(Sensors::get_average_encoders(l_id, r_id));

        if ( std::abs(integral_l) > I_max_l | | !use_integral_l) {
            integral_l = 0;  // reset integral if greater than max allowable value
            use_integral_l = false;
        } else {
            integral_l = integral_l + (error_l * dt);
        }

        if ( std::abs(integral_r) > I_max_l || !use_integral_r) {
            integral_r = 0;  // reset integral if greater than max allowable value
            use_integral_r = false;
        } else {
            integral_r = integral_r + (error_r * dt);
        }

        current_time = pros::millis();

        double derivative_l = error_l - prev_error_l;
        double derivative_r = error_r - prev_error_r;
        prev_error_l = error_l;
        prev_error_r = error_r;

        double left_velocity = (kP_l * error_l) + (kI_l * integral_l) + (kD_l * derivative_l);
        double right_velocity = (kP_r * error_r) + (kI_r * integral_r) + (kD_r * derivative_r);


    // slew rate code
        double delta_velocity_l = left_velocity - prev_velocity_l;
        double delta_velocity_r = right_velocity - prev_velocity_r;
        double slew_rate = args.motor_slew;
        if(std::abs(delta_velocity_l) > (dt * slew_rate) && (std::signbit(delta_velocity_l) == std::signbit(left_velocity)) ) {  // ignore deceleration
            int sign = std::abs(delta_velocity_l) / delta_velocity_l;
            std::cout << "l over slew: " << sign << " " << dt << " " << slew_rate << "\n";
            left_velocity = prev_velocity_l + (sign * dt * slew_rate);
        }

        if(std::abs(delta_velocity_r) > (dt * slew_rate) && (std::signbit(delta_velocity_r) == std::signbit(right_velocity))) {
            int sign = std::abs(delta_velocity_r) / delta_velocity_r;
            std::cout << "r over slew: " << sign << " " << dt << " " << slew_rate << "\n";
            right_velocity = prev_velocity_r + (sign * dt * slew_rate);
        }


    // p controller heading correction
        abs_angle = tracker->get_heading_rad();
        abs_angle = std::atan2(std::sin(abs_angle), std::cos(abs_angle));
        long double delta_theta;
        // account for angle wrap around    ie. new = -1, prev = -359   == bad delta
        if(prev_abs_angle > 0 && abs_angle < 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
            delta_theta = tracker->to_degrees((2*M_PI) + abs_angle) - prev_abs_angle;
        } else if(prev_abs_angle < 0 && abs_angle > 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
            delta_theta = tracker->to_degrees(abs_angle - (2*M_PI)) - prev_abs_angle;
        } else {
            delta_theta = tracker->to_degrees(abs_angle) - prev_abs_angle;
        }

        relative_angle += delta_theta;
        prev_abs_angle = tracker->to_degrees(abs_angle);


        double heading_error = 0 - relative_angle;  // 0 is the setpoint because we want to drive straight
        integral_heading = integral_heading + (heading_error * dt);
        double d_heading_error = heading_error - prev_heading_error;
        prev_heading_error = heading_error;
        // std::cout << "delta_theta: " << delta_theta << " | prev_anlge: " << prev_angle << " | relative angle: " << relative_angle << " | heading_error: " << heading_error << "\n";
        int velocity_correction = (.05 * heading_error) + (0 * integral_heading) + (0 * d_heading_error);
        // int velocity_correction = (4 * heading_error) + (0 * integral_heading) + (54 * d_heading_error);
        if(args.correct_heading && heading_error > 0.00001) {  // veering left
            // velocity_correction = 5;
            right_velocity -= velocity_correction;
        } else if ( args.correct_heading && heading_error < -0.00001) {  // veering right
            // velocity_correction = 5;
            left_velocity -= velocity_correction;
        }


        // cap voltage to max voltage with regard to velocity
        if ( std::abs(left_velocity) > args.max_velocity ) {
            left_velocity = left_velocity > 0 ? args.max_velocity : -args.max_velocity;
        }
        if ( std::abs(right_velocity) > args.max_velocity ) {
            right_velocity = right_velocity > 0 ? args.max_velocity : -args.max_velocity;
        }

        if ( args.log_data ) {
            Logger logger;
            log_entry entry;
            entry.content = (
                "[INFO] " + std::string("CHASSIS_PID")
                + ", Time: " + std::to_string(pros::millis())
                + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
                + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
                + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
                + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
                + ", Slew: " + std::to_string(args.motor_slew)
                + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
                + ", Gear: " + std::to_string(front_left_drive->get_gearset())
                + ", I_max: " + std::to_string(I_max_l)
                + ", I: " + std::to_string(integral_l)
                + ", kD: " + std::to_string(kD_l)
                + ", kI: " + std::to_string(kI_l)
                + ", kP: " + std::to_string(kP_l)
                + ", Position_Sp: " + std::to_string(args.setpoint1)
                + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
                + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
                + ", Heading_Sp: " + std::to_string(args.setpoint2)
                + ", Relative_Heading: " + std::to_string(relative_angle)
                + ", Actual_Vel1: " + std::to_string(front_left_drive->get_actual_velocity())
                + ", Actual_Vel2: " + std::to_string(front_right_drive->get_actual_velocity())
                + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
                + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
            );
            entry.stream = "clog";
            logger.add(entry);
        }

        prev_velocity_l = left_velocity;
        prev_velocity_r = right_velocity;

        previous_l_velocities.push_back(left_velocity);
        previous_r_velocities.push_back(right_velocity);
        if(previous_l_velocities.size() > velocity_history) {
            previous_l_velocities.erase(previous_l_velocities.begin());
        }

        if(previous_r_velocities.size() > velocity_history) {
            previous_r_velocities.erase(previous_r_velocities.begin());
        }

        // settled is when error is almost zero and velocity is minimal
        double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
        double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
        std::cout << "difference: " << *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second << " " << previous_l_velocities.size() << "\n";
        if (
            std::abs(l_difference) < 2
            && previous_l_velocities.size() == velocity_history
            && std::abs(r_difference) < 2
            && previous_r_velocities.size() == velocity_history
            && left_velocity < 2
            && right_velocity < 2
        ) {
            break; // end before timeout
        }


        front_left_drive->move_velocity(left_velocity);
        front_right_drive->move_velocity(right_velocity);
        back_left_drive->move_velocity(left_velocity);
        back_right_drive->move_velocity(right_velocity);

        pros::delay(10);
    } while ( pros::millis() < start_time + args.timeout );

    front_left_drive->set_motor_mode(e_voltage);
    front_right_drive->set_motor_mode(e_voltage);
    back_left_drive->set_motor_mode(e_voltage);
    back_right_drive->set_motor_mode(e_voltage);

    front_left_drive->set_voltage(0);
    front_right_drive->set_voltage(0);
    back_left_drive->set_voltage(0);
    back_right_drive->set_voltage(0);

    front_left_drive->enable_driver_control();
    front_right_drive->enable_driver_control();
    back_left_drive->enable_driver_control();
    back_right_drive->enable_driver_control();

    right_encoder->forget_position(r_id);  // free up space in the encoders log
    left_encoder->forget_position(l_id);
}




void Chassis::t_profiled_straight_drive(chassis_params args) {
    PositionTracker* tracker = PositionTracker::get_instance();
    Configuration* config = Configuration::get_instance();

    double kP = args.kP;
    double kI = args.kI;
    double kD = args.kD;
    double I_max = INT32_MAX;

    front_left_drive->disable_driver_control();
    front_right_drive->disable_driver_control();
    back_left_drive->disable_driver_control();
    back_right_drive->disable_driver_control();

    front_left_drive->set_motor_mode(e_builtin_velocity_pid);
    front_right_drive->set_motor_mode(e_builtin_velocity_pid);
    back_left_drive->set_motor_mode(e_builtin_velocity_pid);
    back_right_drive->set_motor_mode(e_builtin_velocity_pid);

    int r_id = right_encoder->get_unique_id(true);
    int l_id = left_encoder->get_unique_id(true);

    long double relative_angle = 0;
    long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
    long double prev_abs_angle = abs_angle;
    long double integral = 0;
    long double prev_integral = 0;
    double prev_error = 0;
    bool use_integral = true;

    int current_time = pros::millis();
    int start_time = current_time;
    bool settled = false;
    bool was_at_target_l = false;
    bool was_at_target_r = false;

    std::vector<double> previous_l_velocities;
    std::vector<double> previous_r_velocities;
    int velocity_history = 15;

    auto accel_func = [](double n) -> double { return 0.005 * n; };
    // auto accel_func = [](double n) -> double { return 1; };
    std::vector<double> velocity_profile = generate_velocity_profile(std::abs(args.setpoint1), accel_func, .55, args.max_velocity, 50);  // .45 is decceleration, 10 is initial velocity

    do {
        int dt = pros::millis() - current_time;
        current_time = pros::millis();

        double velocity_l;
        double velocity_r;
        if(std::abs(std::get<0>(Sensors::get_average_encoders(l_id, r_id))) <= std::abs(args.setpoint1)) {
            velocity_l = velocity_profile.at(std::abs(std::get<0>(Sensors::get_average_encoders(l_id, r_id))));
        } else {
            was_at_target_l = true;
            velocity_l = 0;
        }

        if(std::abs(std::get<1>(Sensors::get_average_encoders(l_id, r_id))) <= std::abs(args.setpoint1)) {
            velocity_r = velocity_profile.at(std::abs(std::get<1>(Sensors::get_average_encoders(l_id, r_id))));
        } else {
            was_at_target_l = true;
            velocity_r = 0;
        }


        // double velocity;
        // if(velocity_l > velocity_r) {
        //     velocity_r = velocity_r;
        //     velocity_l = velocity_r;
        // } else {
        //     velocity_r = velocity_l;
        //     velocity_l = velocity_l;
        // }

        if(args.setpoint1 < 0) {
            velocity_l = -velocity_l;
            velocity_r = -velocity_r;
        }

        abs_angle = tracker->to_degrees(tracker->get_heading_rad());
        long double delta_theta = abs_angle - prev_abs_angle;
        relative_angle += delta_theta;
        prev_abs_angle = abs_angle;

        long double error = 0 - relative_angle;  // setpoint is 0 because we want to drive straight
        // long double error = std::get<0>(Sensors::get_average_encoders(l_id, r_id)) - std::get<1>(Sensors::get_average_encoders(l_id, r_id));
        std::cout << "relative angle: " << relative_angle << " | dtheta: " << delta_theta << "\n";
        // cap velocity to max velocity with regard to velocity
        integral = integral + (error * dt);
        if(integral > I_max) {
            integral = I_max;
        } else if (integral < -I_max) {
            integral = -I_max;
        }

        // if(std::signbit(error) != std::signbit(prev_error)) {
        //     std::cout << "halving " << integral << " " << prev_integral;
        //     integral = .5 * integral;
        // }
        prev_integral = integral;

        double derivative = error - prev_error;
        prev_error = error;


        // std::cout << error << " " << relative_angle << "\n";

    // pid heading correction
        // int velocity_correction = (kP * error) + (kI * integral) + (kD * derivative);

    // PI heading correction
        // double velocity_correction = std::abs(kI * integral);
        double velocity_correction = std::abs(args.kP * error + args.kI * integral + args.kD * derivative);
        std::cout << "integral: " << integral << " " << velocity_correction << "\n";
        if(args.correct_heading  && error > 0.000001) {  // veering off course, so correct
            velocity_r -= velocity_correction / 2;
            velocity_l += velocity_correction / 2;
        } else if(args.correct_heading && error < -0.000001) {
            velocity_l -= velocity_correction / 2;
            velocity_r += velocity_correction / 2;
        }


    // cap velocity to max velocity with regard to direction
        if ( std::abs(velocity_l) > args.max_velocity ) {
            velocity_l = velocity_l > 0 ? args.max_velocity : -args.max_velocity;
        }
        if ( std::abs(velocity_r) > args.max_velocity ) {
            velocity_r = velocity_r > 0 ? args.max_velocity : -args.max_velocity;
        }

        if ( args.log_data ) {
            Logger logger;
            log_entry entry;
            entry.content = (
                "[INFO] " + std::string("CHASSIS_PROFILED_STRAIGHT_DRIVE")
                + ", Time: " + std::to_string(pros::millis())
                + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
                + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
                + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
                + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
                + ", Slew: " + std::to_string(args.motor_slew)
                + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
                + ", Gear: " + std::to_string(front_left_drive->get_gearset())
                + ", I_max: " + std::to_string(I_max)
                + ", I: " + std::to_string(integral)
                + ", kD: " + std::to_string(kD)
                + ", kI: " + std::to_string(kI)
                + ", kP: " + std::to_string(kP)
                + ", Position_Sp: " + std::to_string(args.setpoint1)
                + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
                + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
                + ", Heading_Sp: " + std::to_string(args.setpoint2)
                + ", Relative_Heading: " + std::to_string(relative_angle)
                + ", Actual_Vel1: " + std::to_string(velocity_l)
                + ", Actual_Vel2: " + std::to_string(velocity_r)
                + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
                + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
                + ", Correction: " + std::to_string(velocity_correction)
            );
            entry.stream = "clog";
            logger.add(entry);
        }

        double error_l = std::abs(args.setpoint1 - std::get<0>(Sensors::get_average_encoders(l_id, r_id)));
        double error_r = std::abs(args.setpoint1 - std::get<1>(Sensors::get_average_encoders(l_id, r_id)));

        previous_l_velocities.push_back(velocity_l);
        previous_r_velocities.push_back(velocity_r);
        if(previous_l_velocities.size() > velocity_history) {
            previous_l_velocities.erase(previous_l_velocities.begin());
        }
        if(previous_r_velocities.size() > velocity_history) {
            previous_r_velocities.erase(previous_r_velocities.begin());
        }

        // settled is when error is almost zero and velocity is minimal
        double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
        double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
        if (
            std::abs(l_difference) < 2
            && previous_l_velocities.size() == velocity_history
            && std::abs(r_difference) < 2
            && previous_r_velocities.size() == velocity_history
            && std::abs(velocity_l) < 2
            && std::abs(velocity_r) < 2
        ) {
            break; // end before timeout
        }
        // if(error_l < 5 || error_r < 5) {  // shut off motors when one side reaches the setpoint
        //     velocity_l = 0;
        //     velocity_r = 0;
        //     break;
        // }

        std::cout << "velocity: " << velocity_l << " " << velocity_r << "\n";
        std::cout << "error: " << error_r << " " << error_l << " " << error << "\n";
        front_left_drive->move_velocity(velocity_l);
        front_right_drive->move_velocity(velocity_r);
        back_left_drive->move_velocity(velocity_l);
        back_right_drive->move_velocity(velocity_r);

        pros::delay(10);
    } while (pros::millis() < start_time + args.timeout);

    front_left_drive->set_motor_mode(e_voltage);
    front_right_drive->set_motor_mode(e_voltage);
    back_left_drive->set_motor_mode(e_voltage);
    back_right_drive->set_motor_mode(e_voltage);

    front_left_drive->set_voltage(0);
    front_right_drive->set_voltage(0);
    back_left_drive->set_voltage(0);
    back_right_drive->set_voltage(0);

    front_left_drive->enable_driver_control();
    front_right_drive->enable_driver_control();
    back_left_drive->enable_driver_control();
    back_right_drive->enable_driver_control();

    front_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    front_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_left_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    back_right_drive->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    right_encoder->forget_position(r_id);  // free up space in the encoders log
    left_encoder->forget_position(l_id);
}




void Chassis::t_turn(chassis_params args) {
    PositionTracker* tracker = PositionTracker::get_instance();
    Configuration* config = Configuration::get_instance();

    double kP = args.kP;
    double kI = args.kI;
    double kD = args.kD;
    double I_max = args.I_max;

    front_left_drive->disable_driver_control();
    front_right_drive->disable_driver_control();
    back_left_drive->disable_driver_control();
    back_right_drive->disable_driver_control();

    front_left_drive->set_motor_mode(e_builtin_velocity_pid);
    front_right_drive->set_motor_mode(e_builtin_velocity_pid);
    back_left_drive->set_motor_mode(e_builtin_velocity_pid);
    back_right_drive->set_motor_mode(e_builtin_velocity_pid);

    front_left_drive->move_velocity(0);
    front_right_drive->move_velocity(0);
    back_left_drive->move_velocity(0);
    back_right_drive->move_velocity(0);

    front_left_drive->tare_encoder();
    front_right_drive->tare_encoder();
    back_left_drive->tare_encoder();
    back_right_drive->tare_encoder();

    int r_id = right_encoder->get_unique_id();
    int l_id = left_encoder->get_unique_id();
    right_encoder->reset(r_id);
    left_encoder->reset(l_id);

    long double relative_angle = 0;
    long double abs_angle = tracker->to_degrees(tracker->get_heading_rad());
    long double prev_abs_angle = abs_angle;
    long double integral = 0;
    double prev_error = 0;
    bool use_integral = true;
    int current_time = pros::millis();
    int start_time = current_time;

    double prev_velocity_l = 0;
    double prev_velocity_r = 0;

    // std::vector<double> previous_l_velocities;
    // std::vector<double> previous_r_velocities;
// int velocity_history = 15;
std::vector<double> error_history;
int max_history_length = 20;

do {
    int dt = pros::millis() - current_time;

    abs_angle = tracker->get_heading_rad();
    abs_angle = std::atan2(std::sin(abs_angle), std::cos(abs_angle));
    long double delta_theta;
    // account for angle wrap around
    // ie. new = -1, prev = -359   == bad delta
    if(prev_abs_angle > 0 && abs_angle < 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
        delta_theta = tracker->to_degrees((2*M_PI) + abs_angle) - prev_abs_angle;
    } else if(prev_abs_angle < 0 && abs_angle > 0 && std::abs(tracker->to_radians(prev_abs_angle)) + std::abs(abs_angle) > (M_PI)) {
        delta_theta = tracker->to_degrees(abs_angle - (2*M_PI)) - prev_abs_angle;
    } else {
        delta_theta = tracker->to_degrees(abs_angle) - prev_abs_angle;
    }
    // long double delta_theta = abs_angle - prev_abs_angle;
    // long double delta_theta = tracker->to_degrees(tracker->get_delta_theta_rad());
    relative_angle += delta_theta;
    prev_abs_angle = tracker->to_degrees(abs_angle);

    long double error = args.setpoint1 - relative_angle;

    integral = integral + (error * dt);
    if(integral > I_max) {
        integral = I_max;
    } else if (integral < -I_max) {
        integral = -I_max;
    }

    double derivative = error - prev_error;
    prev_error = error;

    current_time = pros::millis();

    // std::cout << "relative angle: " << relative_angle << " | dtheta: " << delta_theta << "\n";
    // std::cout << error << " " << relative_angle << "\n";

    double abs_velocity = (kP * error) + (kI * integral) + (kD * derivative);
    double l_velocity = abs_velocity;
    double r_velocity = -abs_velocity;

    // slew rate code
    double delta_velocity_l = l_velocity - prev_velocity_l;
    double delta_velocity_r = r_velocity - prev_velocity_r;
    double slew_rate = args.motor_slew;
    int over_slew = 0;
    if(std::abs(delta_velocity_l) > (dt * slew_rate) && (std::signbit(delta_velocity_l) == std::signbit(l_velocity)) ) {  // ignore deceleration
        int sign = std::abs(delta_velocity_l) / delta_velocity_l;
        std::cout << "l over slew: " << sign << " " << dt << " " << slew_rate << "\n";
        l_velocity = prev_velocity_l + (sign * dt * slew_rate);
        over_slew = 1;
    }

    if(std::abs(delta_velocity_r) > (dt * slew_rate) && (std::signbit(delta_velocity_r) == std::signbit(r_velocity))) {
        int sign = std::abs(delta_velocity_r) / delta_velocity_r;
        std::cout << "r over slew: " << sign << " " << dt << " " << slew_rate << "\n";
        r_velocity = prev_velocity_r + (sign * dt * slew_rate);
        over_slew = 1;
    }


    // cap velocity to max velocity with regard to velocity
    if ( std::abs(l_velocity) > args.max_velocity ) {
        l_velocity = l_velocity > 0 ? args.max_velocity : -args.max_velocity;
    }
    if ( std::abs(r_velocity) > args.max_velocity ) {
        r_velocity = r_velocity > 0 ? args.max_velocity : -args.max_velocity;
    }

    // prev_velocity_l = l_velocity;
    // prev_velocity_r = r_velocity;
    //
    // previous_l_velocities.push_back(l_velocity);
    // previous_r_velocities.push_back(r_velocity);
    // if(previous_l_velocities.size() > velocity_history) {
    //     previous_l_velocities.erase(previous_l_velocities.begin());
    // }
    //
    // if(previous_r_velocities.size() > velocity_history) {
    //     previous_r_velocities.erase(previous_r_velocities.begin());
    // }
    error_history.push_back(prev_error);
    if(error_history.size() > max_history_length) {
         error_history.erase(error_history.begin());
    }


    std::cout << l_velocity << " " << r_velocity << " " << relative_angle << " " << error << "\n";
    // for(int i=0; i < previous_l_velocities.size(); i++) {
    //     std::cout << previous_l_velocities.at(i) << " ";
    // }
    // std::cout << "\n";
    double error_difference = *std::minmax_element(error_history.begin(), error_history.end()).second - *std::minmax_element(error_history.begin(), error_history.end()).first;

    if ( args.log_data ) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("CHASSIS_PID_TURN")
            + ", Time: " + std::to_string(pros::millis())
            + ", Actual_Vol1: " + std::to_string(front_left_drive->get_actual_voltage())
            + ", Actual_Vol2: " + std::to_string(front_right_drive->get_actual_voltage())
            + ", Actual_Vol3: " + std::to_string(back_left_drive->get_actual_voltage())
            + ", Actual_Vol4: " + std::to_string(back_right_drive->get_actual_voltage())
            + ", Slew: " + std::to_string(args.motor_slew)
            + ", Brake: " + std::to_string(front_left_drive->get_brake_mode())
            + ", Gear: " + std::to_string(front_left_drive->get_gearset())
            + ", I_max: " + std::to_string(I_max)
            + ", I: " + std::to_string(integral)
```

```
                + ", kD: " + std::to_string(kD)
                + ", kI: " + std::to_string(kI)
                + ", kP: " + std::to_string(kP)
                + ", Position_Sp: " + std::to_string(0)
                + ", position_l: " + std::to_string(std::get<0>(Sensors::get_average_encoders(l_id, r_id)))
                + ", position_r: " + std::to_string(std::get<1>(Sensors::get_average_encoders(l_id, r_id)))
                + ", Heading_Sp: " + std::to_string(args.setpoint1)
                + ", Relative_Heading: " + std::to_string(relative_angle)
                + ", Absolute Angle: " + std::to_string(abs_angle)
                + ", error history: " + std::to_string(error_history.size())
                + ", history size: " + std::to_string(max_history_length)
                + ", time out time: " + std::to_string(start_time + args.timeout)
                + ", error difference: " + std::to_string(error_difference)
                + ", over slew: " + std::to_string(over_slew)
                + ", Actual_Vel1: " + std::to_string(front_left_drive->get_actual_velocity())
                + ", Actual_Vel2: " + std::to_string(front_right_drive->get_actual_velocity())
                + ", Actual_Vel3: " + std::to_string(back_left_drive->get_actual_velocity())
                + ", Actual_Vel4: " + std::to_string(back_right_drive->get_actual_velocity())
            );
            entry.stream = "clog";
            logger.add(entry);
        }

        // settled is when error is almost zero and velocity is minimal
        // double l_difference = *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second - *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).first;
        // double r_difference = *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).second - *std::minmax_element(previous_r_velocities.begin(), previous_r_velocities.end()).first;
        // std::cout << "difference: " << *std::minmax_element(previous_l_velocities.begin(), previous_l_velocities.end()).second << " " << previous_l_velocities.size() << "\n";
        if (
            std::abs(error_difference) < .007
            && error_history.size() == max_history_length
            && pros::millis() > start_time + 500
            // std::abs(l_difference) < 2
            // && previous_l_velocities.size() == velocity_history
            // && std::abs(r_difference) < 2
            // && previous_r_velocities.size() == velocity_history
            // && l_velocity < 2
            // && r_velocity < 2
        ) {  // velocity change has been minimal, so stop
            front_left_drive->set_motor_mode(e_voltage);
            front_right_drive->set_motor_mode(e_voltage);
            back_left_drive->set_motor_mode(e_voltage);
            back_right_drive->set_motor_mode(e_voltage);

            front_left_drive->set_voltage(0);
            front_right_drive->set_voltage(0);
            back_left_drive->set_voltage(0);
            back_right_drive->set_voltage(0);
            std::cout << "ending\n";
            break; // end before timeout
        }

        front_left_drive->move_velocity(l_velocity);
        front_right_drive->move_velocity(r_velocity);
        back_left_drive->move_velocity(l_velocity);
        back_right_drive->move_velocity(r_velocity);


        pros::delay(10);
    } while ( pros::millis() < (start_time + args.timeout) );

    front_left_drive->set_motor_mode(e_voltage);
    front_right_drive->set_motor_mode(e_voltage);
    back_left_drive->set_motor_mode(e_voltage);
    back_right_drive->set_motor_mode(e_voltage);

    front_left_drive->set_voltage(0);
    front_right_drive->set_voltage(0);
    back_left_drive->set_voltage(0);
    back_right_drive->set_voltage(0);

    front_left_drive->enable_driver_control();
    front_right_drive->enable_driver_control();
    back_left_drive->enable_driver_control();
    back_right_drive->enable_driver_control();

    right_encoder->forget_position(r_id);  // free up space in the encoders log
    left_encoder->forget_position(l_id);
}



void Chassis::t_move_to_waypoint(chassis_params args, waypoint point) {
    PositionTracker* tracker = PositionTracker::get_instance();

    long double dx = point.x - tracker->get_position().x_pos;
    long double dy = point.y - tracker->get_position().y_pos;

    // convert end coordinates to polar to find the change in angle
    // long double dtheta = std::fmod((-M_PI / 2) + std::atan2(dy, dx), (2 * M_PI));
    long double dtheta = std::atan2(dy, dx);
    if(dtheta < 0) {  // map to [0, 2pi]
        dtheta += 2 * M_PI;
    }

    // current angle is bounded by [-pi, pi] re map it to [0, 2pi]
    long double current_angle = tracker->get_heading_rad();
    if(current_angle < 0) {
        current_angle += 2 * M_PI;
    }
    current_angle = (-current_angle) + (M_PI / 2);

    // calculate how much the robot needs to turn to be at the angle
    long double to_turn_face_forwards = current_angle - dtheta;  // change in robot angle
    long double to_turn_face_backwards = (current_angle - dtheta) - M_PI;

    if(to_turn_face_forwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
        to_turn_face_forwards = (-2 * M_PI) + to_turn_face_forwards;  // give negative value to turn left to point
    } else if(to_turn_face_forwards < -M_PI) {
        to_turn_face_forwards = (2 * M_PI) + to_turn_face_forwards;  // give positive value to turn right to point
    }

    if(to_turn_face_backwards > M_PI) {  // find minimal angle change and direction of change [-PI/2, PI/2]
        to_turn_face_backwards = (-2 * M_PI) + to_turn_face_backwards;  // give negative value to turn left to point
    } else if(to_turn_face_backwards < -M_PI) {
        to_turn_face_backwards = (2 * M_PI) + to_turn_face_backwards;  // give positive value to turn right to point
    }


    long double to_turn;
    int direction;
    if(args.explicit_direction == 1) {  // force positive direction
        to_turn = to_turn_face_forwards;
        direction = 1;
    } else if(args.explicit_direction == -1) {  // force negative direction
        to_turn = to_turn_face_backwards;
        direction = -1;
    } else if(std::abs(to_turn_face_forwards) < std::abs(to_turn_face_backwards)) {  // faster to go forwards
        to_turn = to_turn_face_forwards;
        direction = 1;
    } else {  // faster to go backwards
        to_turn = to_turn_face_backwards;
        direction = -1;
    }

    to_turn = tracker->to_degrees(to_turn);


    // set up turn
    chassis_params turn_args;
    turn_args.setpoint1 = to_turn;
    turn_args.max_velocity = args.max_velocity;
    turn_args.timeout = 15000; // TODO: add time estimation
    args.kP = 2.8;
    args.kI = 0.0005;
    args.kD = 50;
    args.I_max = INT32_MAX;
    turn_args.motor_slew = args.motor_slew;
    turn_args.log_data = args.log_data;

    // perform turn
    std::cout << "starting turn\n";
    std::cout << to_turn << "\n";
    t_turn(turn_args);
    std::cout << "turn done\n";

    // caclulate distance to move to point
    long double distance = std::sqrt((std::pow(dx, 2) + std::pow(dy, 2)));
    long double to_drive = direction * tracker->to_encoder_ticks(distance, wheel_diameter);

    // set up straight drive
    chassis_params drive_straight_args;
    drive_straight_args.setpoint1 = to_drive;
    drive_straight_args.setpoint2 = to_drive;
    drive_straight_args.max_velocity = 125;
    drive_straight_args.timeout = 15000;
    drive_straight_args.kP = .77;
    drive_straight_args.kI = 0.000002;
    drive_straight_args.kD = 7;
    drive_straight_args.I_max = INT32_MAX;
    drive_straight_args.motor_slew = args.motor_slew;
    drive_straight_args.correct_heading = args.correct_heading;
    drive_straight_args.log_data = args.log_data;
    //
    // std::cout << "starting drive\n";
    // std::cout << to_drive << "\n";
    // drive_straight_args.kP = .2;
    // drive_straight_args.kI = .001;
    // drive_straight_args.kD = 0;
    // drive_straight_args.I_max = 2000;
    t_pid_straight_drive(drive_straight_args);

    std::cout << "drive finished\n";
    if(args.log_data) {
        Logger logger;
        log_entry entry;
        entry.content = (
            "[INFO] " + std::string("CHASSIS_ODOM")
            + ", Time: " + std::to_string(pros::millis())
            + ", Waypoint: " + point.get_string()
            + ", ToTurnForwards: " + std::to_string(tracker->to_degrees(to_turn_face_forwards))
            + ", ToTurnBackwards: " + std::to_string(tracker->to_degrees(to_turn_face_backwards))
            + ", ToTurn: " + std::to_string(to_turn)
            + ", ToDrive: " + std::to_string(to_drive)
            + ", Direction: " + std::to_string(direction)
            + ", dx: " + std::to_string(dx)
            + ", dy: " + std::to_string(dy)
            + ", X: " + std::to_string(tracker->get_position().x_pos)
            + ", Y: " + std::to_string(tracker->get_position().y_pos)
            + ", Theta: " + std::to_string(tracker->to_degrees(tracker->get_position().theta))
        );
        entry.stream = "clog";
        logger.add(entry);
    }
}



int Chassis::pid_straight_drive(double encoder_ticks, int relative_heading /*0*/, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, bool correct_heading /*true*/, double slew /*0.2*/, bool log_data /*false*/) {
    chassis_params args;
    args.setpoint1 = encoder_ticks;
    args.setpoint2 = encoder_ticks;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.kP = .77;
    args.kI = 0.000002;
    args.kD = 7;
    args.I_max = INT32_MAX;
    args.motor_slew = slew;
    args.correct_heading = correct_heading;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(encoder_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_pid_straight_drive};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}

int Chassis::profiled_straight_drive(double encoder_ticks, int max_velocity  /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, bool correct_heading /*true*/, int relative_heading /*0*/, bool log_data /*false*/) {
    chassis_params args;
    args.setpoint1 = encoder_ticks;
    args.setpoint2 = relative_heading;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.kP = 2;
    args.kI = 0.0005;
    args.kD = 0.001;
    args.I_max = INT32_MAX;
    args.correct_heading = correct_heading;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(encoder_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_profiled_straight_drive};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



int Chassis::uneven_drive(double l_enc_ticks, double r_enc_ticks, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*false*/) {
    chassis_params args;
    args.setpoint1 = l_enc_ticks;
    args.setpoint2 = r_enc_ticks;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.kP = .77;
    args.kI = 0.000002;
    args.kD = 7;
    args.I_max = INT32_MAX;
    args.motor_slew = slew;
    args.correct_heading = false;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(l_enc_ticks) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_pid_straight_drive};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



int Chassis::turn_right(double degrees, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*15*/, bool log_data /*false*/) {
    chassis_params args;
    args.setpoint1 = degrees;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.kP = 2.8;
    args.kI = 0.0005;
    args.kD = 50;
    args.I_max = INT32_MAX;
    args.motor_slew = slew;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(degrees) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_turn};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



int Chassis::turn_left(double degrees, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*15*/, bool log_data /*false*/) {
    chassis_params args;
    args.setpoint1 = -degrees;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.kP = 2.8;
    args.kI = 0.0005;
    args.kD = 50;
    args.I_max = INT32_MAX;
    args.motor_slew = slew;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(degrees) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_turn};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}


int Chassis::drive_to_point(double x, double y, int recalculations /*0*/, int explicit_direction /*0*/, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool correct_heading /*true*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
    chassis_params args;
    args.setpoint1 = x;
    args.setpoint2 = y;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.recalculations = recalculations;
    args.explicit_direction = explicit_direction;
    args.motor_slew = slew;
    args.correct_heading = correct_heading;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(x) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_drive_to_point};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



int Chassis::turn_to_point(double x, double y, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
    chassis_params args;
    args.setpoint1 = x;
    args.setpoint2 = y;
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.motor_slew = slew;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(x) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_turn_to_point};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}



int Chassis::turn_to_angle(double theta, int max_velocity /*450*/, int timeout /*INT32_MAX*/, bool asynch /*false*/, double slew /*10*/, bool log_data /*true*/) {
    PositionTracker* tracker = PositionTracker::get_instance();
    chassis_params args;
    args.setpoint1 = tracker->to_radians(theta);
    args.max_velocity = max_velocity;
    args.timeout = timeout;
    args.motor_slew = slew;
    args.log_data = log_data;

    // generate a unique id based on time, parameters, and seemingly random value of the voltage of one of the motors
    int uid = pros::millis() * (std::abs(theta) + 1) + max_velocity + front_left_drive->get_actual_voltage();

    chassis_action command = {args, uid, e_turn_to_angle};
    while ( command_start_lock.exchange( true ) ); //aquire lock
    command_queue.push(command);
    command_start_lock.exchange( false ); //release lock

    if(!asynch) {
        wait_until_finished(uid);
    }

    return uid;
}


/**
 * sets scaled voltage of each drive motor
 */
void Chassis::move( int voltage )
{
    front_left_drive->move(voltage);
    front_right_drive->move(voltage);
    back_left_drive->move(voltage);
    back_right_drive->move(voltage);
}


/**
 * sets a new brakemode for each drive motor
 */
void Chassis::set_brake_mode( pros::motor_brake_mode_e_t new_brake_mode )
{
    front_left_drive->set_brake_mode(new_brake_mode);
```

```cpp
    front_right_drive->set_brake_mode(new_brake_mode);
    back_left_drive->set_brake_mode(new_brake_mode);
    back_right_drive->set_brake_mode(new_brake_mode);
}




/**
 * sets all chassis motors to the opposite direction that they were facing
 * ie. reversed is now normal and normal is now reversed
 */
void Chassis::change_direction()
{
    front_left_drive->reverse_motor();
    front_right_drive->reverse_motor();
    back_left_drive->reverse_motor();
    back_right_drive->reverse_motor();
}




/**
 * sets slew to enabled for each motor
 * sets the rate of the slew to the rate parameter
 */
void Chassis::enable_slew( int rate /*120*/ )
{
    front_left_drive->enable_slew();
    front_right_drive->enable_slew();
    back_left_drive->enable_slew();
    back_right_drive->enable_slew();

    front_left_drive->set_slew(rate);
    front_right_drive->set_slew(rate);
    back_left_drive->set_slew(rate);
    back_right_drive->set_slew(rate);
}




/**
 * sets slew to disabled for each motor
 */
void Chassis::disable_slew( )
{
    front_left_drive->disable_slew();
    front_right_drive->disable_slew();
    back_left_drive->disable_slew();
    back_right_drive->disable_slew();
}


void Chassis::wait_until_finished(int uid) {
    while(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
        pros::delay(10);
    }
    while ( command_finish_lock.exchange( true ) ); //aquire lock
    commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
    command_finish_lock.exchange( false ); //release lock
}


bool Chassis::is_finished(int uid) {
    if(std::find(commands_finished.begin(), commands_finished.end(), uid) == commands_finished.end()) {
        while ( command_finish_lock.exchange( true ) ); //aquire lock
        commands_finished.erase(std::remove(commands_finished.begin(), commands_finished.end(), uid), commands_finished.end());
        command_finish_lock.exchange( false ); //release lock

        return false;  // command is not finished because it is not in the list
    }
    return true;
}
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 22 12:49:32 2021

@author: aiden
"""

tower_colors = ["blue", "blue", "red"] # top, middle, bottom
indexer_colors = ["red", "none", "none"] #top, middle, bottom

final_tower = ["red", "red", "none"]

num_balls_to_cycle = 0
tower_initial = [i for i in tower_colors if i != "none"]
indexer_initial = [i for i in tower_colors if i != "none"]
cycle = indexer_colors + tower_colors
while cycle[3:] != final_tower:
    num_balls_to_cycle += 1
    cycle.insert(0, cycle.pop())
    in_tower = cycle[3:]
    in_indexer = cycle[:3]
    print(num_balls_to_cycle, in_indexer, in_tower)
```

```
/**
 * @file: ./RobotCode/src/objects/subsystems/intakes.hpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * Contains class for the front intakes
 * has methods for intaking
 */

#ifndef __INTAKES_HPP__
#define __INTAKES_HPP__

#include <tuple>
#include <queue>

#include "main.h"

#include "../motors/Motor.hpp"
#include "../sensors/Sensors.hpp"
#include "../sensors/BallDetector.hpp"


typedef enum e_intake_command {
    e_intake,
    e_stop_movement,
    e_secure,
    e_hold_outward,
    e_rocket_outwards
} intake_command;


/**
 * @see: Motors.hpp
 *
 * contains methods to allow for control of the indexer
 */
class Intakes
{
    private:
        static Motor *l_intake;
        static Motor *r_intake;

        static int num_instances;

        pros::Task *thread;  // the motor thread
        static std::queue<intake_command> command_queue;
        static std::atomic<bool> lock;

        static void intake_motion_task(void*);

    public:
        Intakes(Motor &left, Motor &right);
        ˜Intakes();

        void intake();
        void stop();
        void intake_until_secure();
        void hold_outward();
        void rocket_outwards();

        void reset_queue();
};




#endif
```

```
/**
 * @file: ./RobotCode/src/objects/subsystems/intakes.cpp
 * @author: Aiden Carney
 * @reviewed_on:
 * @reviewed_by:
 *
 * Contains implementation for the front intakes subsystem
 * has methods for intaking
 */

#include "main.h"


#include "../serial/Logger.hpp"
#include "intakes.hpp"

int Intakes::num_instances = 0;
std::queue<intake_command> Intakes::command_queue;
std::atomic<bool> Intakes::lock = ATOMIC_VAR_INIT(false);
Motor* Intakes::l_intake;
Motor* Intakes::r_intake;

Intakes::Intakes(Motor &left, Motor &right)
{
    l_intake = &left;
    r_intake = &right;

    l_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    r_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    l_intake->set_motor_mode(e_voltage);
    r_intake->set_motor_mode(e_voltage);

    l_intake->disable_slew();
    r_intake->disable_slew();

    if(num_instances == 0 | | thread == NULL) {
        thread = new pros::Task( intake_motion_task, (void*)NULL, TASK_PRIORITY_DEFAULT, TASK_STACK_DEPTH_DEFAULT, "intakes_thread");
    }

    num_instances += 1;
}


Intakes::~Intakes() {
    num_instances -= 1;
    if(num_instances == 0) {
        delete thread;
    }
}




void Intakes::intake_motion_task(void*) {
    l_intake->tare_encoder();
    r_intake->tare_encoder();
    l_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);
    r_intake->set_brake_mode(pros::E_MOTOR_BRAKE_BRAKE);

    int abs_position_l = 0;  // the absolute postions are calculated based on the change in encoder value
    int abs_position_r = 0;  // and capped to max and min values
    int prev_encoder_l = l_intake->get_encoder_position();
    int prev_encoder_r = r_intake->get_encoder_position();
    int integral_l = 0;
    int integral_r = 0;
    int dt = 0;
    int time = pros::millis();

    while(1) {
        if(command_queue.empty()) {  // delay unitl there is a command in the queue
            pros::delay(7);
            continue;
        }

        // take lock and get command
        while ( lock.exchange( true ) ); //aquire lock
        intake_command command = command_queue.front();
        command_queue.pop();
        lock.exchange( false ); //release lock

        if(command != e_hold_outward) {  // reset integral if no longer holding outwards
            integral_l = 0;
            integral_r = 0;
        }

        dt = pros::millis() - time;  // calculate change in time since last command
        time = pros::millis();

        int d_enc_l = l_intake->get_encoder_position() - prev_encoder_l;
        int d_enc_r = r_intake->get_encoder_position() - prev_encoder_r;
        prev_encoder_l = l_intake->get_encoder_position();
        prev_encoder_r = r_intake->get_encoder_position();
        abs_position_l += d_enc_l;
        abs_position_r += d_enc_r;

        // cap encoder values. This can be done because mechanical stops stop the motion of
        // the intakes
        if (abs_position_l > 0) {  // innermost value of the encoder
            abs_position_l = 0;
        }

        if (abs_position_r > 0) {  // innermost value of the encoder
            abs_position_r = 0;
        }
        // std::cout << abs_position_l << " " << l_intake->get_actual_voltage() << "\n";
        // execute command
        switch(command) {
            case e_intake: {
                l_intake->set_voltage(12000);
                r_intake->set_voltage(12000);
                break;
            } case e_stop_movement: {
                l_intake->set_voltage(0);
                r_intake->set_voltage(0);
                break;
            } case e_secure: {
                l_intake->set_voltage(12000);
                r_intake->set_voltage(12000);
                if ((l_intake->get_torque() + r_intake->get_torque()) / 2 > 1) { // wait a little bit and then say ball is secure
                    pros::delay(300);
                    l_intake->set_voltage(0);
                    r_intake->set_voltage(0);
                }
                break;
            } case e_hold_outward: {  // PI controller to hold outwards
                // double l_error = -37 - abs_position_l;  // set first number to encoder setpoint
                // double r_error = -37 - abs_position_r;  // set first number to encoder setpoint
                //
                // integral_l = integral_l + (l_error * dt);
                // integral_r = integral_r + (r_error * dt);
                //
                // int voltage_l = (40 * l_error) + (1 * integral_l);  // set first number to kP, second number to kI
                // int voltage_r = (40 * r_error) + (1 * integral_r);  // set first number to kP, second number to kI
                // if(abs_position_l > -30) {
                //     l_intake->set_voltage(-5000);
                // } else {
                //     l_intake->set_voltage(-1500);  // doesn't take a lot to keep it out, so less voltage
                // }
                //
                // if(abs_position_r > -30) {
                //     r_intake->set_voltage(-5000);
                // } else {
                //     r_intake->set_voltage(-1500);  // doesn't take a lot to keep it out, so less voltage
                // }
                //
                l_intake->set_voltage(-3500);
                r_intake->set_voltage(-3500);
                break;
            } case e_rocket_outwards: {
                l_intake->set_voltage(-12000);
                r_intake->set_voltage(-12000);
                break;
            }
        }

    }
}

void Intakes::intake() {
    while ( lock.exchange( true ) ); //aquire lock
    command_queue.push(e_intake);
    lock.exchange( false ); //release lock
}

void Intakes::stop() {
    reset_queue();
    while ( lock.exchange( true ) ); //aquire lock
    command_queue.push(e_stop_movement);
    lock.exchange( false ); //release lock
}

void Intakes::intake_until_secure() {
    while ( lock.exchange( true ) ); //aquire lock
    command_queue.push(e_secure);
    lock.exchange( false ); //release lock
}

void Intakes::hold_outward() {
    while ( lock.exchange( true ) ); //aquire lock
    command_queue.push(e_hold_outward);
    lock.exchange( false ); //release lock
}

void Intakes::rocket_outwards() {
    while ( lock.exchange( true ) ); //aquire lock
    command_queue.push(e_rocket_outwards);
    lock.exchange( false ); //release lock
}

void Intakes::reset_queue() {
    while ( lock.exchange( true ) ); //aquire lock
    std::queue<intake_command> empty_queue;
    std::swap( command_queue, empty_queue );  // replace command queue with an empty queue
    lock.exchange( false ); //release lock
}
```

../RobotCode/src/objects/subsystems/profile_tester.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan 17 12:47:59 2021

@author: aiden
"""


import matplotlib.pyplot as plt

def gen_profile(enc_ticks, max_acceleration, max_decceleration, max_velocity, initial_velocity):
    profile = [initial_velocity]

    i = 0
    while(i < enc_ticks):
        ticks_left = enc_ticks - i
        ticks_to_decclerate = profile[i] / max_decceleration
        if(ticks_to_decclerate < ticks_left):
            step = (profile[i] + max_acceleration(i))
            if(step > max_velocity):
                step = max_velocity
            profile.append(step)
        else:
            profile.append((profile[i] - max_decceleration))

        i += 1

    return profile

def accel_profile(x):
    vel = .005 * x
    print(vel)
    return vel

ticks = 1000

y = gen_profile(ticks, accel_profile, 0.8, 450, 50)
x = list(range(ticks + 1))
print(len(y))
plt.scatter(x, y)
plt.show()
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 31 14:01:59 2020

@author: aiden
"""

import kivy
from kivy.app import App
from kivy.clock import Clock
from kivy.uix.label import Label
from kivy.uix.gridlayout import GridLayout
from kivy.uix.textinput import TextInput
from kivy.uix.button import Button
from kivy.uix.widget import Widget
from kivy.properties import ObjectProperty
from kivy.uix.tabbedpanel import TabbedPanel
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.popup import Popup
from kivy.properties import StringProperty
from kivy.event import EventDispatcher
import sys


class Data:
    __instance = None

    motors = {
        0:"Front Right",
        1:"Front Left",
        2:"Back Right",
        3:"Back Left",
        4:"Main Intake",
        5:"Hoarding Intake",
        6:"Lift",
    }

    @staticmethod
    def get_instance():
       """ Static access method. """
       if Data.__instance == None:
          Data()
       return Data.__instance

    def __init__(self):
        if Data.__instance != None:
            raise RuntimeError("Constructor has already been called and exists at " + str(Data.__instance))
        else:
            Data.__instance = self
            self.motors_data = {}

    def api_interaction(self, byte1, byte2, msg):
        return None

    def retrieve_motor_data(self, *args):
        data = {}
        for i, motor in self.motors.items():
            motor_data = {
                "Actual Velocity":self.api_interaction(0xA0, 0xA0, i),
                "Actual Voltage":self.api_interaction(0xA0, 0xA1, i),
                "Current Draw":self.api_interaction(0xA0, 0xA2, i),
                "Encoder Position":self.api_interaction(0xA0, 0xA3, i),
                "Brakemode":self.api_interaction(0xA0, 0xA4, i),
                "Gearset":self.api_interaction(0xA0, 0xA5, i),
                "Port":self.api_interaction(0xA0, 0xA6, i),
                "PID Constants":self.api_interaction(0xA0, 0xA7, i),
                "Slew Rate":self.api_interaction(0xA0, 0xA8, i),
                "Power":self.api_interaction(0xA0, 0xA9, i),
                "Temperature":self.api_interaction(0xA0, 0xAA, i),
                "Torque":self.api_interaction(0xA0, 0xAB, i),
                "Direction":self.api_interaction(0xA0, 0xAC, i),
                "Efficiency":self.api_interaction(0xA0, 0xAD, i),
                "Is Stopped":self.api_interaction(0xA0, 0xAE, i),
                "Is Reversed":self.api_interaction(0xA0, 0xAF, i),
                "Is Registered":self.api_interaction(0xA1, 0xA0, i)
            }

            data.update({i:motor_data})
        self.motors_data = data


class Settings:
    __instance = None

    @staticmethod
    def get_instance():
       """ Static access method. """
       if Settings.__instance == None:
          Settings()
       return Settings.__instance

    def __init__(self):
        if Settings.__instance != None:
            raise RuntimeError("Constructor has already been called and exists at " + str(Settings.__instance))
        else:
            Settings.__instance = self
            self.ip_address = "127.0.0.1"
            self.motor_dashboard_selected = "Actual Velocity"
            self.motor_selected = 0

    def update_motor_dashboard_selected(self, new_value):
        self.motor_dashboard_selected = new_value

    def update_motor_selected(self, new_value):
        self.motor_selected = new_value
        if self.motor_selected not in Data.motors.keys():
            self.motor_selected = 0



class SettingsPopup(FloatLayout):
    def update_ip_addr(self, new_ip):
        print("New IP set: ", new_ip)



class MainScreen(FloatLayout):

    motor_title_label_text = StringProperty("")

    def __init__(self):
        super(MainScreen, self).__init__()

    def open_settings(self):
        s = SettingsPopup()
        popup_window = Popup(
            title="Settings",
            content=s,
            size_hint=(None,None),
            size=(self.width / 2, self.height / 2)
        )

        popup_window.open()

    @classmethod
    def update_motor_info_labels(cls):
        # self.ids.get("label0").text = str(Data.get_instance().motors_data.get(0, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label1").text = str(Data.get_instance().motors_data.get(1, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label2").text = str(Data.get_instance().motors_data.get(2, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label3").text = str(Data.get_instance().motors_data.get(3, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label4").text = str(Data.get_instance().motors_data.get(4, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label5").text = str(Data.get_instance().motors_data.get(5, {}).get(App.get_running_app().settings.motor_dashboard_selected))
        # self.ids.get("label6").text = str(Data.get_instance().motors_data.get(6, {}).get(App.get_running_app().settings.motor_dashboard_selected))

        cls.motor_title_label_text = str(Data.get_instance().motors.get(Settings.get_instance().motor_selected))


class VexServer(App):
    settings = Settings.get_instance()
    l0 = StringProperty("")
    l1 = StringProperty("")
    l2 = StringProperty("")
    l3 = StringProperty("")
    l4 = StringProperty("")
    l5 = StringProperty("")
    l6 = StringProperty("")
    motor_data_title = StringProperty("")
    motor_data_body = StringProperty("")


    def update_motor_data(self, *args):
        self.l0 = str(Data.get_instance().motors_data.get(0, {}).get(self.settings.motor_dashboard_selected))
        self.l1 = str(Data.get_instance().motors_data.get(1, {}).get(self.settings.motor_dashboard_selected))
        self.l2 = str(Data.get_instance().motors_data.get(2, {}).get(self.settings.motor_dashboard_selected))
        self.l3 = str(Data.get_instance().motors_data.get(3, {}).get(self.settings.motor_dashboard_selected))
        self.l4 = str(Data.get_instance().motors_data.get(4, {}).get(self.settings.motor_dashboard_selected))
        self.l5 = str(Data.get_instance().motors_data.get(5, {}).get(self.settings.motor_dashboard_selected))
        self.l6 = str(Data.get_instance().motors_data.get(6, {}).get(self.settings.motor_dashboard_selected))

        self.motor_data_title = str(Data.get_instance().motors.get(self.settings.motor_selected))
        body_text = ""
        for key, value in Data.get_instance().motors_data.get(self.settings.motor_selected, {}).items():
            body_text += key + ": " + str(value) + '\n'
        # print(self.motor_data_body)
        self.motor_data_body = body_text

    def build(self):
        Clock.schedule_interval(self.update_motor_data, 0.1)
        Clock.schedule_interval(Data.get_instance().retrieve_motor_data, 0.01)
        return MainScreen()


def mainloop(dt):
    # VexServer.set_more_info_labels(App.get_running_app().settings.motor_selected)
    # VexServer.set_comparison_labels(App.get_running_app().settings.motor_dashboard_selected)
    # print(App.get_running_app().l0)
    MainScreen.update_motor_info_labels()



if __name__ == "__main__":
    VexServer().run()
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jul 26 11:00:42 2020

@author: aiden
"""

import multiprocessing as mp
import serial
import subprocess
import threading
import time
import queue
from functools import wraps
import sys




def create_double(n):
    if n >= 0:
        sign = 0
    else:
        sign = 1

    bias = 1023
    interval = (0, 2048)
    exp_term = 0
    while not 1 <= exp_term < 2:
        mid = int(interval[0] + (interval[1] - interval[0]) / 2)
        exp_term = (abs(n) / (2**(mid - bias)))

        if exp_term < 1:  # use lower range
            interval = (interval[0], mid + 1)
            exp_guess = mid
        else:  # use upper range
            mid = int(interval[0] + (interval[1] - interval[0]) / 2)
            interval = (mid - 1, interval[1])
            exp_guess = mid

    significand = 0
    total = exp_term - 1
    for i in range(1, 53):
        if total - (2**(-i)) >= 0:
            total = total - (2**(-i))
            significand |= 2**(52 - i)

    byte_list = []
    # first 6 bytes are from significand
    for i in range(6):
        byte = significand & 0xff
        byte_list.append(byte)
        significand = significand >> 8

    # 7th byte is part significand (4 bits) and part exponent (4 bits)
    byte = significand
    byte |= ((exp_guess & 0x0f) << 4)
    exp_guess = exp_guess >> 4
    byte_list.append(byte)

    # 8th byte is part exponent (7 bits) and the sign bit (1 bit)
    byte = exp_guess & 0x7f
    byte |= sign << 7
    byte_list.append(byte)

    return byte_list



class Client:
    def __init__(self, uid):
        self.uid = uid
        self.send_queue = queue.Queue()
        self.send_queue_lock = threading.Lock()

        self.recv_queue = queue.Queue()
        self.recv_queue_lock = threading.Lock()


    def _send_message(self, id1, id2, msg=""):
        msg = id1 + id2 + msg
        with self.send_queue_lock:
            self.send_queue.put(msg)


    def _receive_message(self, max_wait=5, sent_message=""):
        if max_wait is None:
            max_wait = sys.maxsize - 1

        start = time.time()
        while 1:  # set max waiting time to 5 sec
            if not self.recv_queue.empty():
                with self.recv_queue_lock:
                    received = self.recv_queue.get()
                break

            end = time.time()

            if (end - start) > max_wait:
                error_msg = "No response returned from host in allotted time"
                if sent_message:
                    error_msg += " with sent message: " + sent_message
                raise TimeoutError(error_msg)

        return received


    def get_command(self, id1, id2, msg=""):
        self._send_message(id1, id2, msg)
        return self._receive_message(5, msg)

    def post_command(self, id1, id2, msg=""):
        pass

    def debug(self, debug_message):
        self._send_message('\xAB', '\xA0', debug_message)
        return self._receive_message(5, debug_message)


class ServerConnection:
    def __init__(self, debug=False, read_chunk_size=1024):
        self.connection = None
        self.debug = debug
        self.read_chunk_size = read_chunk_size

        self.__write_thread = threading.Thread(target=self.write_thread)
        self.run_writing_thread = False

        self.__read_thread = threading.Thread(target=self.read_server_stdout)
        self.run_reading_thread = False

        self.__write_thread.daemon = True
        self.__read_thread.daemon = True

        self.clients = []
        self.client_lock = threading.Lock()

        self.connection_lock = threading.Lock()

        self.__write_thread.start()
        self.__read_thread.start()



    def serial_exception_handler(func):
        @wraps(func)
        def inner_function(self, *args, **kwargs):
            while 1:
                try:
                    return func(self, *args, **kwargs)
                except serial.SerialException as e:
                    if self.debug:
                        print(e)
                except serial.serialutil.SerialException as e:
                    if self.debug:
                        print(e)
                except OSError as e:
                    if self.debug:
                        print(e)
                except AttributeError:
                    if self.debug:
                        print("Connection is not established; attemting to establish one")

                self.run_writing_thread = False
                self.run_reading_thread = False
                time.sleep(.5)
                self.connection = None

                with self.connection_lock:  # use a lock in case multiple threads are trying to establish a connection
                    while not self.mount_vex_brain():
                        time.sleep(.1)
                        if self.debug:
                            print("retrying connection", flush=True)

                self.run_writing_thread = True
                self.run_reading_thread = True


        return inner_function


    def mount_vex_brain(self):
        command = "./usb.sh"
        process = subprocess.Popen(command, stdout=subprocess.PIPE)

        ttys = []
        for i in process.stdout.readlines():
            i = i.decode("utf=8")
            if "VEX" in i:
                ttys.append(i.split(" ")[0])

        if not ttys:
            if self.debug:
                print("No mount points for the vex brain were found", flush=True)
            return 0

        mnts = sorted(ttys, reverse=True)
        tty = ""
        for i in mnts:
            if "ACM" in i:
                tty = i
                break

        try:
            self.connection = serial.Serial(
                tty,
                baudrate=115200,
                bytesize=serial.EIGHTBITS,
                parity=serial.PARITY_NONE,
                stopbits=serial.STOPBITS_ONE
            )

        except serial.SerialException:
            if self.debug:
                print("Failed to open Vex Brain on ", tty)
            return 0

        if self.debug:
            print("connection established")

        return 1


    @serial_exception_handler
    def read_bytes(self):
```

```python
        return self.connection.read(self.connection.in_waiting)

    @serial_exception_handler
    def write_bytes(self, send_array):
        self.connection.write(send_array)
        return 1


    def read_server_stdout(self):
        read_check = 0
        while 1:
            if self.run_reading_thread:
                bytes_read = iter(self.read_bytes())
                terminal_output = ""
                for byte in bytes_read:
                    if read_check == 0 and byte == 0xAA:
                        read_check = 1
                    elif read_check == 1 and byte == 0x55:
                        read_check = 2
                    elif read_check == 2 and byte == 0x1E:
                        read_check = 3
                    elif read_check == 3:
                        num_bytes_following = byte
                        uid_msb = next(bytes_read)
                        uid_lsb = next(bytes_read)
                        uid = (uid_msb << 8) | uid_lsb
                        msg = ""
                        for _ in range(num_bytes_following - 2):
                            try:
                                char = chr(next(bytes_read))
                            except UnicodeDecodeError:
                                if self.debug:
                                    print("failed to decode character")
                                char = ""

                            msg += char
                        if self.debug:
                            print("message received: ", msg, "at", time.time())

                        checksum = next(bytes_read)
                        if checksum == 0xC6:
                            # find server with that id and add message to its queue
                            for client in self.clients:
                                if client.uid == uid:
                                    with client.recv_queue_lock:
                                        client.recv_queue.put(msg)

                        elif checksum != 0xC6 and self.debug:
                            print("checksum failed - received: ", checksum)

                        read_check = 0

                    else:  # if response from server is not part of message send to stdout
                        read_check = 0
                        try:
                            char = chr(byte)
                        except UnicodeDecodeError:
                            print(byte)
                            char = ""
                        terminal_output += char
                        # print(char, end="")
                with open("log.txt", "a") as f:
                    f.write(terminal_output)
                    terminal_output = ""

                time.sleep(.1)


    def write_thread(self):
        while 1:
            if self.run_writing_thread:
                send_array = bytearray()
                for client in self.clients:
                    with client.send_queue_lock:
                        if not client.send_queue.empty():
                            to_write = client.send_queue.get()
                        else:
                            continue


                    send_array.append(0xAA)
                    send_array.append(0x55)
                    send_array.append(0x1E)
                    send_array.append(len(to_write) + 2)  # add two for the uid bytes

                    send_array.append((client.uid >> 8) & 0xFF)
                    send_array.append(client.uid & 0xFF)

                    for i in to_write:
                        send_array.append(ord(i))

                    send_array.append(0xC6)

                    if self.debug:
                        print("Message added to be sent: ", to_write, "at", time.time())

                if send_array:
                    self.write_bytes(send_array)
                    if self.debug:
                        print("Message array sent at", time.time())

                    # write garbage on the stream to help clear any blocking functions on server
                # send_array = bytearray()
                # send_array.append(0xFF)
                # send_array.append(0xFF)
                # send_array.append(0xFF)
                # send_array.append(0xFF)
                # send_array.append(0xFF)

                # self.write_bytes(send_array)

                # time.sleep(.01)


    def add_clients(self, *args):
        with self.client_lock:
            for client in args:
                self.clients.append(client)


    def start_server(self):
        self.run_writing_thread = True
        self.run_reading_thread = True

    def stop_server(self):
        self.run_writing_thread = False
        self.run_reading_thread = False


def handle_requests_async(connection, *args, **kwargs):
    clients = []
    for i in range(55000, 55000 + len(args)):
        client = Client(i)
        clients.append(client)

    connection.add_clients(*clients)

    for request, client in zip(args, clients):
        client._send_message(request[0], request[1], request[2])

    responses = [None for i in range(len(args))]
    i = 0
    start = time.time()
    dt = 0
    while None in responses and dt < kwargs.get("max_wait", 5):
        try:
            response = clients[i]._receive_message(max_wait=.001)
            responses[i] = response
        except TimeoutError:
            pass
        i += 1

        if i > len(clients) - 1:
            i = 0
        dt = time.time() - start

    return responses



if __name__ == "__main__":
    c = ServerConnection(debug=True)
    x = c.mount_vex_brain()
    c.start_server()
    client = Client(55000)
    c.add_client(client)
    while 1:
        print("starting debug msg at", time.time())
        client.get_command('\xA0', '\xA0', "0")
        # start = time.time()
        # print("start time:", start)
        # for i in range(20):
        #     client._send_message('\xAB', '\xA0', "test")

        # print("time to send 20 messages: ", time.time() - start)
        print("ended debug msg at", time.time())
        time.sleep(20)
```

../Serial/test.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug  6 14:30:19 2020

@author: aiden
"""


from kivy.app import App
from kivy.properties import StringProperty
from kivy.uix.floatlayout import FloatLayout
from kivy.clock import Clock
import random
import time as t

class MainScreen(FloatLayout):
    pass

class TestApp(App):
    time = ""

    def update(self, *args):
        self.time = str(t.asctime()) # + 'time'?

    def build(self):
        Clock.schedule_interval(self.update, 1)
        return MainScreen()



if __name__ == "__main__":
    TestApp().run()


```

../Serial/usb.sh

```bash
#!/bin/bash

for sysdevpath in $(find /sys/bus/usb/devices/usb*/ -name dev); do

    syspath="${sysdevpath%/dev}"
    devname="$(udevadm info -q name -p $syspath)"
    [[ "$devname" == "bus/"* ]] && continue
    eval "$(udevadm info -q property --export -p $syspath)"
    [[ -z "$ID_SERIAL" ]] && continue
    echo "/dev/$devname - $ID_SERIAL"

done
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 27 16:48:08 2020

@author: aiden
"""
import flask

import serial_client


# values = ”””0xA0 0xA0
# 0xA0 0xA1
# 0xA0 0xA2
# 0xA0 0xA3
# 0xA0 0xA4
# 0xA0 0xA5
# 0xA0 0xA6
# 0xA0 0xA7
# 0xA0 0xA8
# 0xA0 0xA9
# 0xA0 0xAA
# 0xA0 0xAB
# 0xA0 0xAC
# 0xA0 0xAD
# 0xA0 0xAE
# 0xA0 0xAF
# 0xA1 0xA0”””

# for line in values.split(”\n”):
#     byte_s = line.split(” ”)
#     msb = byte_s[0].strip()
#     lsb = byte_s[1].strip()

#     return_id = (int(msb, 0) << 8) | int(lsb, 0);
#     print(return_id)
motors = {
    0:"Front Right",
    1:"Front Left",
    2:"Back Right",
    3:"Back Left",
    4:"Main Intake",
    5:"Hoarding Intake",
    6:"Lift",
}

app = flask.Flask(__name__)


def get_motor_data(connection, motor_num):
    data = serial_client.handle_requests_async(connection,
        ('\xA0', '\xA0', motor_num),
        ('\xA0', '\xA1', motor_num),
        ('\xA0', '\xA2', motor_num),
        ('\xA0', '\xA3', motor_num),
        ('\xA0', '\xA4', motor_num),
        ('\xA0', '\xA5', motor_num),
        ('\xA0', '\xA6', motor_num),
        ('\xA0', '\xA7', motor_num),
        ('\xA0', '\xA8', motor_num),
        ('\xA0', '\xA9', motor_num),
        ('\xA0', '\xAA', motor_num),
        ('\xA0', '\xAB', motor_num),
        ('\xA0', '\xAC', motor_num),
        ('\xA0', '\xAD', motor_num),
        ('\xA0', '\xAE', motor_num),
        ('\xA0', '\xAF', motor_num),
        ('\xA1', '\xA0', motor_num)
    )
    motor_data = {
        "Actual Velocity":data[0],
        "Actual Voltage":data[1],
        "Current Draw":data[2],
        "Encoder Position":data[3],
        "Brakemode":data[4],
        "Gearset":data[5],
        "Port":data[6],
        "PID Constants":data[7],
        "Slew Rate":data[8],
        "Power":data[9],
        "Temperature":data[10],
        "Torque":data[11],
        "Direction":data[12],
        "Efficiency":data[13],
        "Is Stopped":data[14],
        "Is Reversed":data[15],
        "Is Registered":data[16]
    }
        # motor_data = {
        #     ”Actual Velocity”:client.get_command('\xA0', '\xA0', motor_num),
        #     ”Actual Voltage”:client.get_command('\xA0', '\xA1', motor_num),
        #     ”Current Draw”:client.get_command('\xA0', '\xA2', motor_num),
        #     ”Encoder Position”:client.get_command('\xA0', '\xA3', motor_num),
        #     ”Brakemode”:client.get_command('\xA0', '\xA4', motor_num),
        #     ”Gearset”:client.get_command('\xA0', '\xA5', motor_num),
        #     ”Port”:client.get_command('\xA0', '\xA6', motor_num),
        #     ”PID Constants”:client.get_command('\xA0', '\xA7', motor_num),
        #     ”Slew Rate”:client.get_command('\xA0', '\xA8', motor_num),
        #     ”Power”:client.get_command('\xA0', '\xA9', motor_num),
        #     ”Temperature”:client.get_command('\xA0', '\xAA', motor_num),
        #     ”Torque”:client.get_command('\xA0', '\xAB', motor_num),
        #     ”Direction”:client.get_command('\xA0', '\xAC', motor_num),
        #     ”Efficiency”:client.get_command('\xA0', '\xAD', motor_num),
        #     ”Is Stopped”:client.get_command('\xA0', '\xAE', motor_num),
        #     ”Is Reversed”:client.get_command('\xA0', '\xAF', motor_num),
        #     ”Is Registered”:client.get_command('\xA1', '\xA0', motor_num)
        # }
    # except TimeoutError as e:
        # print(e)
        # motor_data = {
        #     ”Actual Velocity”:None,
        #     ”Actual Voltage”:None,
        #     "Current Draw":None,
        #     "Encoder Position":None,
        #     "Brakemode":None,
        #     "Gearset":None,
        #     "Port":None,
        #     "PID Constants":None,
        #     "Slew Rate":None,
        #     "Power":None,
        #     "Temperature":None,
        #     "Torque":None,
        #     "Direction":None,
        #     "Efficiency":None,
        #     "Is Stopped":None,
        #     "Is Reversed":None,
        #     "Is Registered":None
        # }

    return motor_data


class InvalidUsage(Exception):
    status_code = 400

    def __init__(self, message, status_code=None, payload=None):
        Exception.__init__(self)
        self.message = message
        if status_code:
            self.status_code = status_code
        self.payload = payload

    def to_dict(self):
        rv = dict(self.payload or ())
        rv['message'] = self.message
        return rv


@app.errorhandler(InvalidUsage)
def handle_invalid_usage(error):
    response = flask.jsonify(error.to_dict())
    response.status_code = error.status_code
    return response


@app.route("/api/motor_data/<motor_number>", methods=["GET"])
def api_get_motor_data(motor_number):
    if int(motor_number) in motors.keys():
        data = get_motor_data(server_conn, motor_number)
        return flask.jsonify(data)

    else:
        raise InvalidUsage("Motor Number supplied was not valid", status_code=406)

# @app.route("/api/debug", methods=["GET"])
# def api_debug():
#     motor_client.debug("test message")


server_conn = serial_client.ServerConnection(debug=True)
x = server_conn.mount_vex_brain()
server_conn.start_server()



app.run(host='0.0.0.0')
```