March 18, 2013

Team 19 Support Tool Description

Laboratory # 4: Laboratory Name

Morgan, Laura
Miaw, Jireh
Hauser, Steven
Dworak, Catherine
Bertoglio, David

Work Product

A description of the On-board Test Support Tool

Document Revision Information

Created 3/18/2013

Approval Sheet

All group members whose names are listed below approve of the document and contributed fairly.

Member Names

Morgan, Laura
Miaw, Jireh
Hauser, Steven
Dworak, Catherine
Bertoglio, David

Pledge

On my honor, as a student, I have neither given nor received unauthorized aid on this assignment.

Names

Morgan, Laura
Miaw, Jireh
Hauser, Steven
Dworak, Catherine
Bertoglio, David

Contents

Overview 4

Methods 4

Overview

The on-board robot test tool is a simple program written in Java that will be used to test the on-board robot system. It contains various methods that create commands based on user input and send them to the on-board system. The engineer will type a command in natural language from a list of pre-set commands to perform one of the 7 actions: move straight, move in an arc, turn stationary, stop, set speed, read sensor, or no operation. This command will be turned into a 10 byte message following the communications protocol to test how the on-board system responds to messages, both correct and incorrect.

Methods

The test tool has a main method and many helper methods:

Main

The main method creates the Bluetooth connection between the computer and the robot and requests a command from user input, which is then sent to the createComand method.

String createCommand(string)

The createCommand method takes a string as input and returns a string of length 10 to be sent via Bluetooth to the robot. It splits the command into a string array, and calls methods to create specific messages based on the first word of the command, passing additional arguments for longer commands.

String[] getCommandArguments(String [])

This method returns a string array without the first word of the command, to be passed as an argument for longer commands.

String getCommand(String[])

This method returns the first word in a command, which is used to decide which method to send the command to create the correct message.

Boolean isNumeric(String)

This method parses a string to an integer.

All createMessage methods to create individual messages return a string command of length 10. Some also take in a String Array with additional arguments, such as movement backward or forward, left or right, or a number.

String createMoveCommand(String[])
String createArcCommand(String[])
String createTurnCommand(String[])
String createStopCommand
String createSetSpeedMessage(String[])
String createReadSensorMessage(String[])
String createNoOpMessage()

String createMalformedMessage()

This method creates a malformed message to test whether the onboard software detects malformed messages/fixes them.

String getCommandHelp()

This method prints various commands so the user will know what commands they may enter.

Messages:

Move Straight: “Move, forward/backward, (number)”
Move Arc “Arc, forward/backward, left/right”
Turn: “Turn, left/right, (degrees)”
Stop: “Stop”
Set Speed: “setspeed, motor, speed”
Read sensors: “read, all/u/t/m/l”
NoOp: “none”