ChoreoLib

LabVIEW

Reference

Version 2026.0.0

Table of Contents

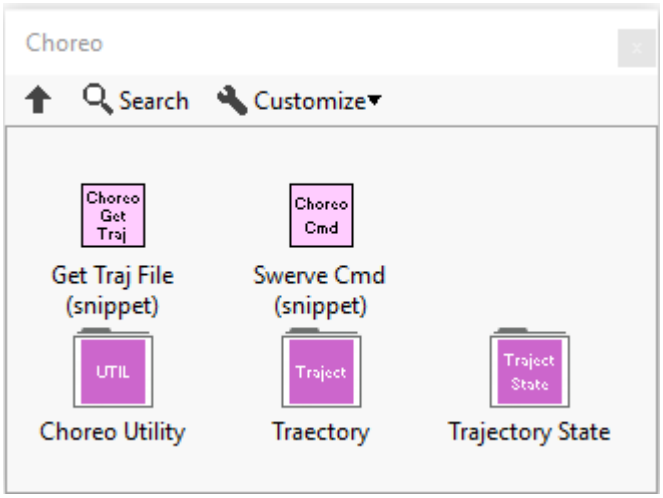
Introduction

The ChoreoLib LabVIEW library provides utility functions to read, sample, and follow Choreo trajectories.

The library source code, package build specifications, and test package can be found here https://github.com/jsimpso81/ChoreoLabVIEW

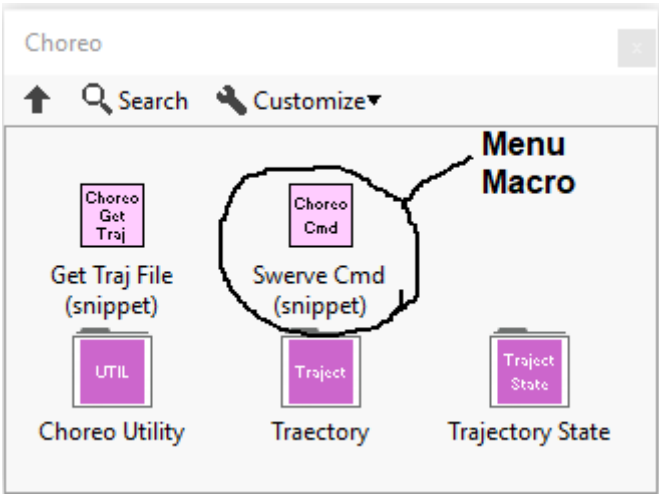
Function Menus

A Choreo function palette contains the Choreo functions and type definitions. This palette can be accessed from the WPI Robotics Library Third Party pallete.



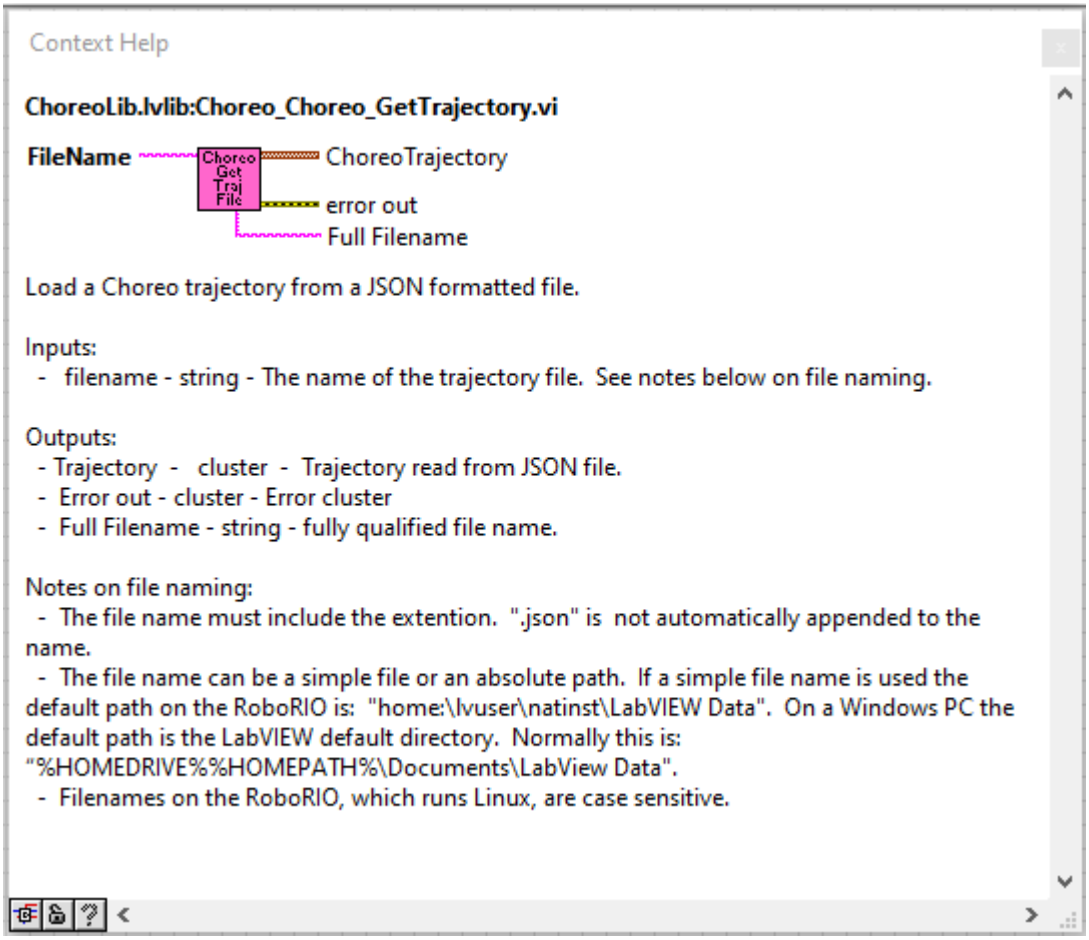
Menu Macros

Some of the menu items place “snippets” of code into a VI. These “macros” can greatly speed up development by placing large sections of mostly completed code in a VI. Usually macros have a different color menu palette icon and may contain “(snippet)” in the description.



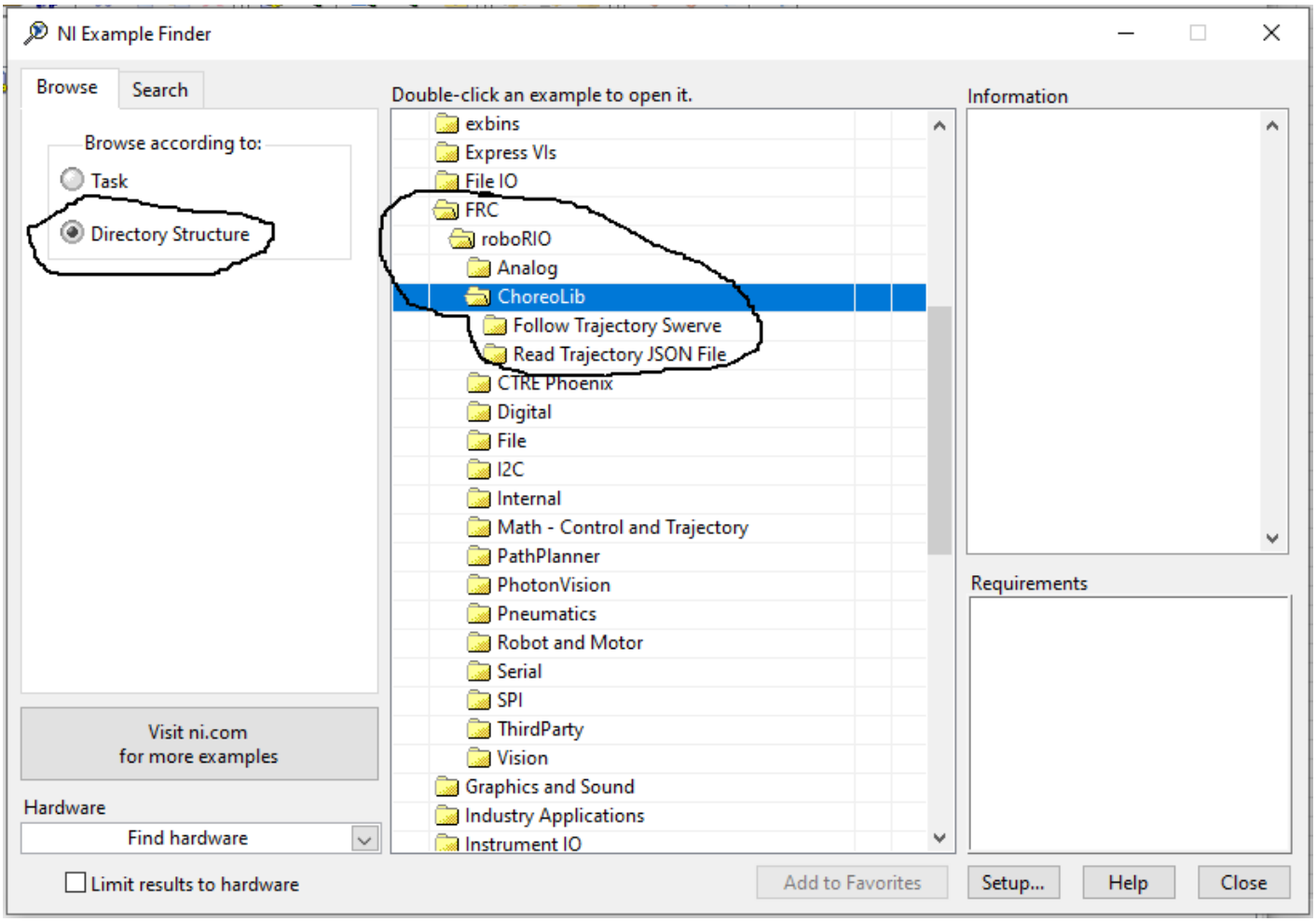
Function Help

Each VI includes help that can be accessed using the standard LabVIEW help toggle (Ctrl H).



Function Examples

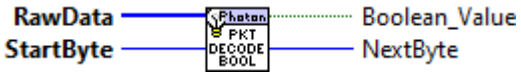
Many of the functions have examples that can be found under the LabVIEW "Find examples..." function. (Help -> Find Examples...). The function examples are easiest to find when "Directory Structure" is selected.


NI Example Finder
Browse
Search
Browse according to:
Task
Directory Structure
Visit ni.com
for more examples
Hardware
Find hardware
Limit results to hardware
Double-click an example to open it.
exbins
Express VIs
File IO
FRC
roboRIO
Analog
ChoreoLib
Follow Trajectory Swerve
Read Trajectory JSON File
CTRE Phoenix
Digital
File
I2C
Internal
Math - Control and Trajectory
PathPlanner
PhotonVision
Pneumatics
Robot and Motor
Serial
SPI
ThirdParty
Vision
Graphics and Sound
Industry Applications
Instrument IO
Information
Requirements
Add to Favorites
Setup...
Help
Close

Function Groups

ByteBuffer

Choreo_ByteBuffer_decode_Boolean



Decode (or strip out) a boolean from an array of bytes. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type. A TRUE boolean is stroed as an integer value = 1.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- Boolean_Value -- Boolean value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_ByteI8

Decode (or strip out) an unsigned byte from an array of bytes. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- Int8_Value -- Byte in the form of an INT 8 value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_Flt32

Decode (or strip out) a single precision floating point value FLT 32. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- FLT32_Value -- Extracted float 32 (single float) value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_Flt64

Decode (or strip out) a double precision floating point value FLT 64. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- FLT64_Value -- Extracted float 64 (double) value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_Int16

Decode (or strip out) a 16 bit integer value from the provided byte array. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- INT16_Value -- Extracted 16 bit integer value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_Int32

Decode (or strip out) a 32 bit integer value from the provided byte array. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- INT32_Value -- Extracted 32 bit integer value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_decode_Int64

Decode (or strip out) a 64 bit integer value from the provided byte array. No conversion or decoding is actually done, the memory is just copied, and the value is cast to the correct type.

Inputs

- RawData -- Array of bytes containing the data to be extracted.

- StartByte -- The starting byte number (the first byte starts at 0)

Outputs

- INT64_Value -- Extracted 64 bit integer value.

- NextByte -- The starting byte index for the next value (regardless of type).

Choreo_ByteBuffer_encode_Boolean

Encode (or append) a boolean to the end of an array of bytes. The boolean is stored as an unsigned integer byte, where TRUE = 1, FALSE = 0.

Inputs

- RawData In -- Array of bytes to be appended to..

- Boolean_Value -- Boolean value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_ByteI8

Encode (or append) an unsigned byte at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..
- Int8_Value -- Byte in the form of an INT 8 value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_FLT32

Encode (or append) a single floating point value (FLT 32) at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..
- FLT32_Value -- Single floating point value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_FLT64

Encode (or append) a double floating point value (FLT 64) at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..

- FLT64_Value -- Double floating point value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_Int16

Encode (or append) a 16 bit integer at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..

- Int16_Value -- 16 bit integer the form of an INT 16 value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_Int32

Encode (or append) a 32 bit integer at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..

- In32_Value -- 32 bit integer the form of an INT 32 value.

Outputs

- RawData Out -- Array of bytes containing the appended value.

Choreo_ByteBuffer_encode_Int64

Encode (or append) a 64 bit integer at the end of an array of bytes. No conversion is actually done, the memory is just copied to the end of the array.

Inputs

- RawData In -- Array of bytes to be appended to..

- In64_Value -- 64 bit integer the form of an INT 64 value.

Outputs

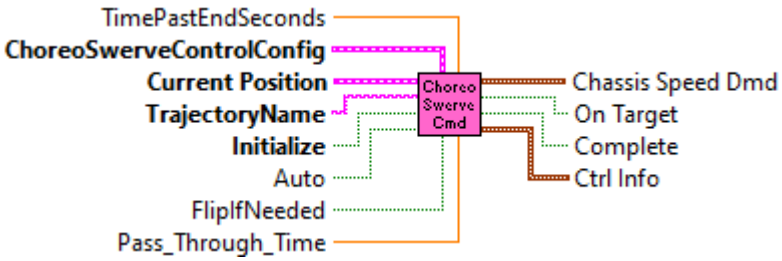
- RawData Out -- Array of bytes containing the appended value.

Choreo

Choreo_Choreo_Globals

Global variables for the Choreo Trajectory following system.

Choreo_Choreo_SwerveCommand



Command to assist in the execution of a Choreo Trajectory. This wraps the ChoreoSwerveCtrl VI.

Inputs:

- TimePastEndSeconds - double - Number of seconds past the end of the trajectory to continue to control robot position. (Optional. Default: 3.0)

- ChoreoSwerveControlConfig - cluster - Controller configuration data

- Current Pose - pose2d - Current robot position pose calcualted by either Odometry or Pose Estimation.

- TrajectoryName - string - Name of trajectory to load from cache. If it isn't in cache, it will try to be loaded from the file system.

- Initialize - boolean - Set to true for one scan cycle when starting a trajectory. Resets the elapsed time and controllers.

- Auto - boolean - When TRUE closed loop control is used to control trajectory execution. This is mostly for debugging and robot tuning. (Optional. Default: TRUE)

- Mirror For Red Alliance - boolean - Whether or not to mirror the path based on alliance (this assumes the path is created for the blue alliance) If TRUE, the alliance color is queried and if RED, the trajectory is flipped for execution.

- PassThroughTime - double - Continueually counting time (seconds) read from the FGPA on roboRIO or local computer when running on PC. (Optional: Default: Read FGPA time)

Outputs:

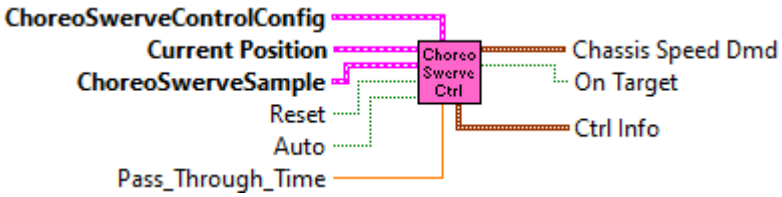
- Choreo Speed Demand - chassis speed - Desired chassis speed (M/S, Rad/Sec)

- On Target - boolean - TRUE when robot position is within tolerance of trajectory position

- Complete - boolean - TRUE when trajectory time has elapsed and robot is on target or when trajectory time is "TimePastEndSeconds" past trajectory end time regardless of On Target status.

- Ctrl Info - cluster - Information on control. Mostly for diagnostics.

Choreo_Choreo_SwerveController



Creates and executes a swerve drive control function to execute a Choreo Trajectory sample. The controller consists of three advanced PID controllers, one for X position, Y position, and Rotation.

Inputs:

- ChoreoSwerveControlConfig - cluster - Controller configuration data

- Current Pose - pose2d - Current robot position pose calcualted by either Odometry or Pose Estimation.

- ChoreoTrajectoryState - cluster - Trajectory state to execute

- Reset - boolean - Resets the controllers.

- Auto - boolean - When TRUE closed loop control is used to control trajectory execution. This is mostly for debugging and robot tuning. (Optional. Default: TRUE)

- PassThroughTime - double - Continueually counting time (seconds) read from the FGPA on roboRIO or local computer when running on PC. (Optional: Default: Read FGPA time)

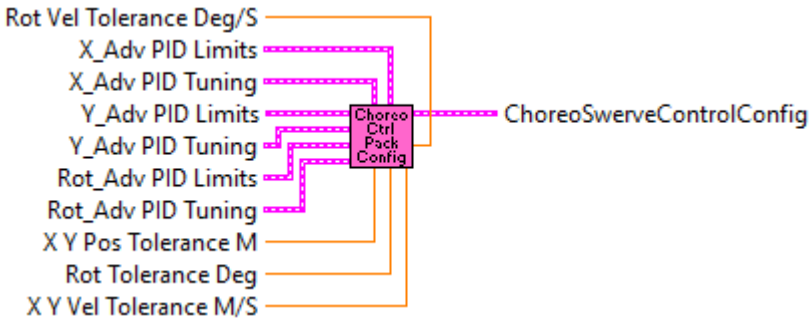
Outputs:

- Choreo Speed Demand - chassis speed - Desired chassis speed (M/S, Rad/Sec)

- On Target - boolean - TRUE when robot position is within tolerance of trajectory position

- Ctrl Info - cluster - Information on control. Mostly for diagnostics.

Choreo_Choreo_SwerveControllerPackConfig



Packs the configuration information for a Choreo Swerve Controller into a cluster for use by the Choreo Swerve Controller or the Choreo Swerve Command functions.

Note:

- Defaults have been chosen for the inputs, however tuning based on actual performance is advised.

Inputs:

- X Adv PID Limits - cluster - Input and output limits for the X position PID (Optional.)

- X Adv PID Tuning - cluster - Tuning values for the X position PID (optional)

- Y Adv PID Limits - cluster - Input and output limits for the Y position PID (Optional.)

- Y Adv PID Tuning - cluster - Tuning values for the Y position PID (optional)

- Rot Adv PID Limits - cluster - Input and output limits for the rotation position PID. Note this PID must be set to continuous (Optional.)

- Rot Adv PID Tuning - cluster - Tuning values for the rotational position PID (optional)

- X Y Pos Tolerance - double - Distance (Meters) to be considered on target. (Defaultt 0.0381)

- Rot Pos Tolerance - double - Rotation (Degrees) to be considered on target. (Defaultt 10.0)

- X Y Vel Tolerance - double - Velocity (Meters/Second) to be considered on target. (Defaultt 1.0)

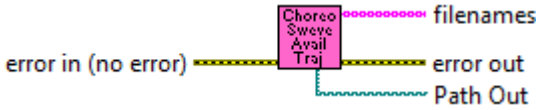
- Rot Vel Tolerance - double - Velocity (Degrees/Second) to be considered on target. (Defaultt 30.0)

Outputs:

- ChoreoSwerveControlConfig - cluster - packed controller configuration.

ChoreoSwerve

Choreo_ChoreoSwerve_AvailableTrajectories



Fetches the names of all available trajectories in the deploy directory. All trajectories are assumed to be type Swerve.

Inputs:

- ErrorIn - cluster - standard error cluster. Can be used to force execution order.

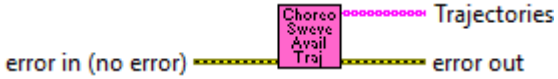
Outputs:

- Filenames - string array - List of *.traj files found.

- PathOut - path - the path that was searched.

- ErrorOut - cluster - standard error cluster. Can be used to force execution order and check for errors.

Choreo_ChoreoSwerve_LoadAllTrajectories



Load all *.traj files found in the default choreo deploy directory into cache. This is an easy way to load everything into memory when the robot is started. All trajectories are expected to be of type Swerve.

Inputs:

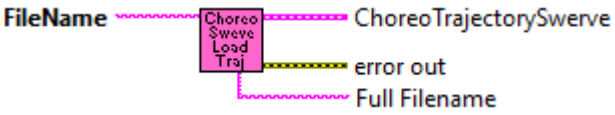
- ErrorIn - cluster - standard error cluster. Can be used to force execution order.

Outputs:

- Trajectories - string array - List of the names of the trajectories found and loaded. This is just for information.

- ErrorOut - cluster - standard error cluster. Can be used to force execution order and check for errors.

Choreo_ChoreoSwerve_LoadTrajectory



Load a Swerve trajectory from the file system deploy directory. Choreolib expects .traj files to be placed in /deploy/choreo/.traj.

Inputs:

- filename - string - The name of the trajectory file. See notes below on file naming.

Outputs:

- Trajectory - cluster - Trajectory read from JSON file.

- Error out - cluster - Error cluster

- Full Filename - string - fully qualified file name.

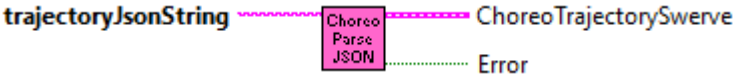
Notes on file naming:

- The file does not need to include the ".traj" extension. This is added if not present.

- The file name can be a simple file or an absolute path. If a simple file name is used the default path on the RoboRIO is: "home:\lvuser\natinst\LabVIEW Data\choreo". On a Windows PC the default path is the LabVIEW default directory\choreo. Normally this is: %HOMEDRIVE%%HOMEPATH%\Documents\LabView Data\choreo".

- Filenames on the RoboRIO, which runs Linux, are case sensitive.

Choreo_ChoreoSwerve_LoadTrajectoryString



Parse a Swerve choreo trajectory from a JSON formatted string.

Inputs:

- trajectoryJsonString - string- The string containing the choreo trajectory. Note that files may have a .TRAJ extension. These are actually internally formatted as JSON files.

Outputs:

- ChoreoTrajectorySwerve - cluster - Trajectory read from JSON file.
- Error - boolean - TRUE if an error occured.

EventMarker

Choreo_EventMarker_Equals

Check to see if two event markers are equal

Inputs:

-- EventMarker - cluster - The event marker data cluster

-- OtherEventMarker - cluster - The other event marker data cluster to check

Outputs:

-- equal - boolean - TRUE if event markers are equal

Choreo_EventMarker_FromJSON

Create an array of event markers from json string

Inputs:

- JSONString - string - String potentially containing an event marker

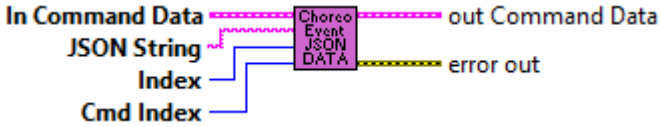
Outputs:

- EventMarkers - array of cluster - The event markers defined by the given json object
- Exists - boolean- TRUE if any event markers were found in the JSON string.

Notes:

Currently actual command data is ignored. Command will be issed based on the "name" of this event marker. Currently the only type of commands that can be issued are Boolean commands. The value is forced to TRUE when the command is issued.

Choreo_EventMarker_FromJSON_Data



Internal function to parse JSON data for Event Markers. This is data that is different for each different type of Event Marker command.

Notes:

Currently this is not used. Commands are issued soley on the "name" of the event marker.

Choreo_EventMarker_New

Constructs an EventMarker with the specified parameters.

Inputs:

-- timestamp - double - The timestamp of the event. (seconds)

-- event - string - The event data string

Outputs:

-- EventMarker - cluster - the constructed event marker

Choreo_EventMarker_OffsetBy

Returns a new EventMarker with the timestamp offset by the specified amount.

Inputs:

-- EventMarker - cluster - The event marker data to be offset.

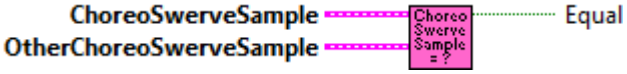
-- timestampOffset - double - The amount to offset the timestamp by.

Outputs:

-- offsetEventMarker - cluster - A new EventMarker with the timestamp offset by the specified amount.

SwerveSample

Choreo_SwerveSample_Equals



Determine if two Swerve Trajectory Samples are equal

Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

-- OtherChoreoSwerveSample - cluster - Swerve sample to compare.

Outputs"

-- Equal - boolean - TRUE when the samples are the same

Choreo_SwerveSample_Flipped

Returns this sample, mirrored across the field midline.

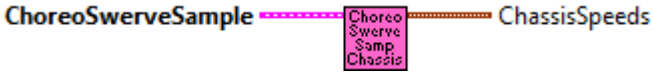
Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

-- FlippedSwerveSample - cluster - This sample, mirrored across the field midline.

Choreo_SwerveSample_GetChassisSpeeds



Returns the field-relative chassis speeds of this sample.

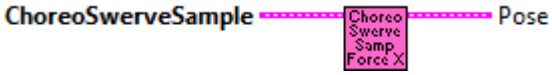
Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

-- ChassisSpeed - ChassisSpeed cluster - The field-relative chassis speeds of this sample.

Choreo_SwerveSample_GetPose



Returns the pose at this sample.

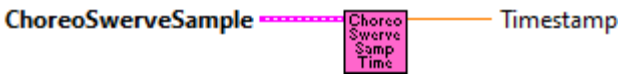
Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

-- Pose - Pose2d - The pose at this sample.

Choreo_SwerveSample_GetTimestamp



Returns the timestamp of this sample.

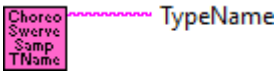
Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

-- TimeStamp - double - the timestamp of this sample. (seconds)

Choreo_SwerveSample_GetTypeName



Return the type name of this sample. The response string is "SwerveSample".

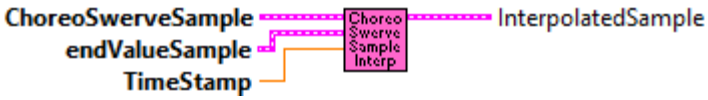
Inputs:

-- None --

Outputs:

-- TypeName - string - Returned name of this type of trajectory sample.

Choreo_SwerveSample_Interpolate



Interpolate between two Swerve Trajectory Samples.

Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

-- endChoreoSwerveSample - cluster - The end Swerve sample

-- TimeStamp - double - desired time stamp

Outputs:

-- InterpolatedSample - cluster - Interpolated Swerve sample.

Choreo_SwerveSample_ModuleForcesX

Retrieves the X module forces from a Swerve Trajectory Sample

Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

Choreo_SwerveSample_ModuleForcesY

Retrieves the Y module forces from a Swerve Trajectory Sample

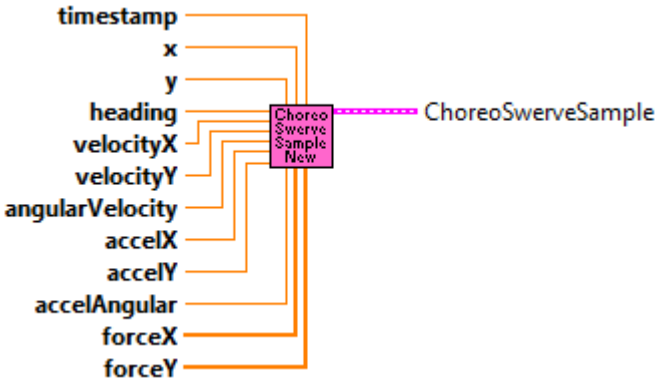
Inputs:

-- ChoreoSwerveSample - cluster - Swerve sample.

Outputs:

-- ForcesY - double array - The force on each swerve module in the Y direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

Choreo_SwerveSample_New



Constructs a SwerveSample with the specified parameters.

Inputs:

- timestamp - double - The timestamp of this sample, relative to the beginning of the trajectory.

- x - double - The X position of the sample in meters.

- y - double - The Y position of the sample in meters.

- heading - double - The heading of the sample in radians, with 0 being in the +X direction.

- velocityX - double - The velocity of the sample in the X direction in m/s.

- velocityY - double - The velocity of the sample in the Y direction in m/s.

- angularVelocity - double - The angular velocity of the sample in rad/s.

- accelX - double - The acceleration of the sample in the X direction in m/s².

- accelY - double - The acceleration of the sample in the Y direction in m/s².

- accelAngular - double - The angular acceleration of the sample in rad/s².

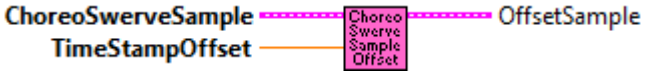
- ForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- ForcesY - double array - The force on each swerve module in the Y direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

Outputs:

- ChoreoSwerveSample - cluster - Cluster containing sample data

Choreo_SwerveSample_OffsetBy



Returns this sample, offset by the given timestamp.

Inputs:

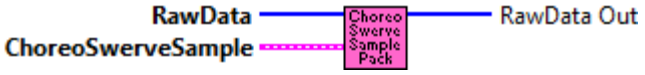
-- ChoreoSwerveSample - cluster - Swerve sample.

-- timestampOffset -- double - The offset to apply to the timestamp. (seconds)

Outputs:

-- OffsetSample - cluster - this sample, offset by the given timestamp.

Choreo_SwerveSample_Pack



Pack a Swerve Trajectory Sample cluster into a binary byte stream. New data will be added to the end of existing data.

Inputs:

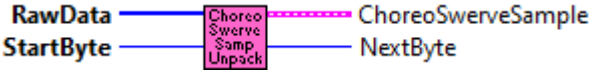
-- ChoreoSwerveSample - cluster - Swerve trajectory sample data

-- RawData - Byte Array - Input byte stream

OutputsL:

-- RawData - Byte Array - Updated array of data.

Choreo_SwerveSample_Unpack



Unpack a binary byte stream into a Swerve Trajectory Sample.

Inputs:

-- RawData - Byte Array - Input byte stream

-- StartByte - Integer - Starting byte (0 relative) to parse

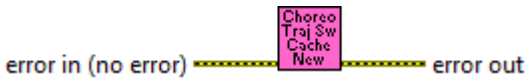
OutputsL:

-- ChoreoSwerveSample - cluster - Swerve trajectory sample data

-- NextByte - integer - Index into next byte to parse

TrajectoryCacheSwerve

Choreo_TrajectoryCacheSwerve_Clear



Clear the swerve memory cache

Inputs:

-- ErrorIn -- cluster -- standard error cluster. Can be used to force execution sequence.

Outputs:

-- ErrorOut -- cluster -- standard error cluster. Can be used to force execution sequence and check for error.

Choreo_TrajectoryCacheSwerve_LoadTrajectory

Load a Swerve trajectory from the memory cache. If the trajectory doesn't exist in the cache is is read from the file system default choreo directory. Choreolib expects .traj files to be placed in /deploy/choreo/.traj.

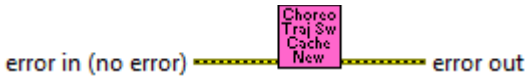
Inputs:

- TrajectoryName - string - The name of the trajectory. This should NOT have a .traj file extension.
- ErrorIn - cluster - standard error cluster

Outputs:

- ChoreoSwerveTrajectory - cluster - The trajectory obtained from the cache or from a file.
- InCache - boolean - TRUE if the trajectory was found in the cache.
- ErrorOut - cluster - Error cluster

Choreo_TrajectoryCacheSwerve_New



Initialize and clear the swerve memory cache

Inputs:

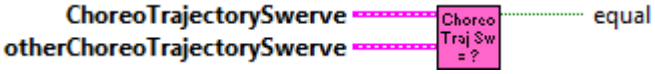
-- ErrorIn -- cluster -- standard error cluster. Can be used to force execution sequence.

Outputs:

-- ErrorOut -- cluster -- standard error cluster. Can be used to force execution sequence and check for error.

TrajectorySwerve

Choreo_TrajectorySwerve_Equals



Determine if two trajectories are equal

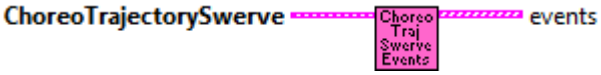
Inputs:

- ChoreoTrajectory - cluster - Input trajectory
- Other ChoreoTrajectory - cluster - Trajectory to compare

Outputs:

- Equals - boolean - TRUE if trajectories are equal.

Choreo_TrajectorySwerve_Events



Returns the events in the trajectory.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

Outputs:

-- Events - cluster array - the events in the trajectory.

Choreo_TrajectorySwerve_Flipped

Returns this trajectory, mirrored across the field midline.

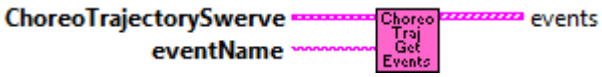
Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

Outputs:

-- FlippedTrajectorySwerve - cluster - this trajectory, mirrored across the field midline.

Choreo_TrajectorySwerve_GetEvents



Returns a list of all events with the given name in the trajectory.

Inputs:

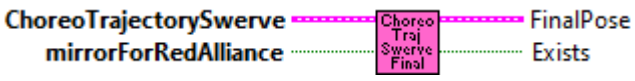
-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

-- eventName - string - The name of the event.

Outputs:

-- Events - cluster array - A list of all events with the given name in the trajectory, if no events are found, an empty list is returned.

Choreo_TrajectorySwerve_GetFinalPose



Returns the final pose in the trajectory.

This function will return empty and Exists will be false.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

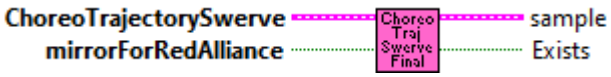
-- mirrorForRedAlliance - boolean - whether or not to return the sample as mirrored across the field

Outputs:

-- FinalPose - pose2d - The final pose in the trajectory.

-- exists - boolean - the sample exits.

Choreo_TrajectorySwerve_GetFinalSample



Returns the final swerve sample in the trajectory.

This function will return empty and Exists will be false.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

-- mirrorForRedAlliance - boolean - whether or not to return the sample as mirrored across the field

Outputs:

-- Sample - cluster - The final swerve sample in the trajectory.

-- exists - boolean - the sample exits.

Choreo_TrajectorySwerve_GetInitialPose

Returns the first pose in the trajectory.

This function will return empty and Exists will be false.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

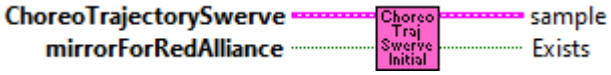
-- mirrorForRedAlliance - boolean - whether or not to return the sample as mirrored across the field

Outputs:

-- InitialPose - pose2d - The first pose2d in the trajectory.

-- exists - boolean - the sample exits.

Choreo_TrajectorySwerve_GetInitialSample



Returns the first swerve sample in the trajectory.

This function will return empty and Exists will be false.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

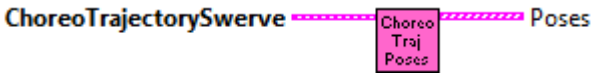
-- mirrorForRedAlliance - boolean - whether or not to return the sample as mirrored across the field

Outputs:

-- Sample - cluster - The first swerve sample in the trajectory.

-- exists - boolean - the sample exits.

Choreo_TrajectorySwerve_GetPoses



Return all posse of this trajectory

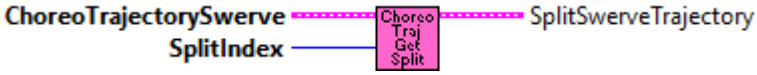
Inputs:

- ChoreoTrajectory - cluster - Input trajectory

Outputs:

- Poses - pose2d array - all, non-mirrored poses of the trajectory.

Choreo_TrajectorySwerve_GetSplit



Returns a choreo trajectory that represents the split of the trajectory at the given index.

Inputs:

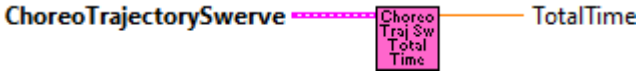
-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

-- splitIndex - integer - the index of the split trajectory to return.

Outputs:

-- SplitSwerveTrajectory - cluster - a choreo trajectory that represents the split of the trajectory at the given index.

Choreo_TrajectorySwerve_GetTotalTime



Returns the total time of the trajectory (the timestamp of the last sample). This will return 0 if the trajectory is empty.

Inputs:

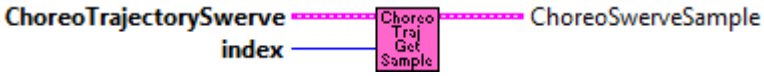
- ChoreoTrajectory - cluster - Input trajectory

Outputs:

- TotalTime - double - The total time duration of this trajectory (seconds).

.

Choreo_TrajectorySwerve_GetTrajSample



Return a particular swerve trajectory sample

Inputs:

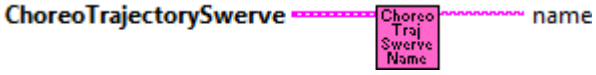
-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

-- index - integer - zero relative index into the array of samples

Outputs:

-- ChoreoSwerveSample - cluster - selected trajectory sample

Choreo_TrajectorySwerve_Name



Returns the name of the trajectory.

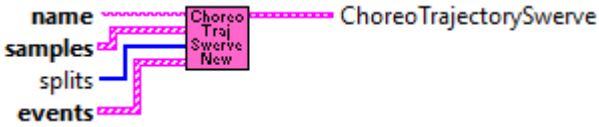
Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

Outputs:

-- name - string - the name of the trajectory.

Choreo_TrajectorySwerve_New



Constructs a Swerve Trajectory with the specified parameters.

Inputs:

-- name - string - The name of the trajectory.

-- samples - cluster array - The samples of the trajectory.

-- splits - integer array - The indices of the splits in the trajectory.

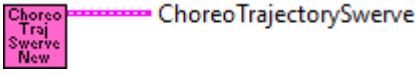
-- events - cluster array - The events in the trajectory.

Outputs:

- ChoreoTrajectory - cluster - Created trajectory stat cluster

---

Choreo_TrajectorySwerve_New_Empty



Create an empty trajectory

Inputs:

Outputs:

- ChoreoSwerveTrajectory - cluster - Empty swerve trajectory data cluster.

Choreo_TrajectorySwerve_SampleAt

Return an interpolated sample of the trajectory at the given timestamp.

Inputs:

- ChoreoTrajectory - cluster - The trajectory to sample.

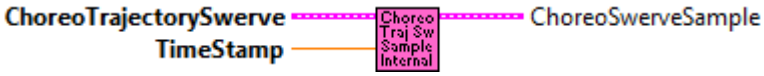
- timestamp - double - The timestamp of this sample relative to the beginning of the trajectory.

- mirrorForRedAlliance - boolean - whether or not to return the sample as mirrored across the field midline (as in 2023). (Optional. Default: false)

Outputs:

- ChoreoTrajectoryState - cluster - The ChoreoTrajectoryState at the given time.

Choreo_TrajectorySwerve_SampleInternal



Internal routine to sample a swerve trajectory at a particular point in time.

Inputs:

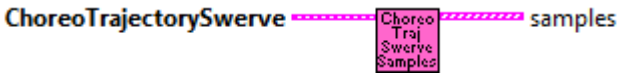
-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

-- timestamp - double - time to sample

Outputs:

-- ChoreoSwerveSample - cluster - interpolated sample

Choreo_TrajectorySwerve_Samples



Returns the samples of the trajectory.

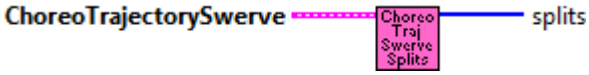
Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

Outputs:

-- samples - SwerveSample - the samples of the trajectory.

Choreo_TrajectorySwerve_Splits



Returns the indices of the splits in the trajectory.

Inputs:

-- ChoreoTrajectorySwerve - cluster - data cluster for this trajectory

Outputs:

-- Splits - integer array - the indices of the splits in the trajectory.

Util

Choreo_Util_AllianceFlipGetFlipper

Get the type of flipping to be done for the currently selected competition year.

Inputs:

- none -

Outputs:

- Flipper - enum - Type of flipping to be done - MIRRORED, or ROTATE_AROUND.

Choreo_Util_AllianceFlipGetYearInfoNOT_DONE

trajectory, mirrored across the field midline.

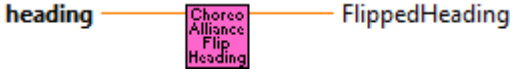
Inputs:

- ChoreoTrajectory - cluster - Input trajectory

Outputs:

- Flipped Trajectory - cluster - trajectory, mirrored across the field midline.

Choreo_Util_AllianceFlipHeading



Flips a heading

Inputs:

-- heading - double - The heading to flip. (radians)

Outputs:

-- FlippedHeading - double - The flipped heading (radians)

Choreo_Util_AllianceFlipPose2d

Flips a pose2d cluster.

Inputs:

-- pose - pose2d - The pose to flip.

Outputs:

-- FlippedPose - pose2d - The flipped cluster

Choreo_Util_AllianceFlipPose3d

Flips a pose3d cluster.

Inputs:

-- pose - pose3d - The pose to flip.

Outputs:

-- FlippedPose - pose3d - The flipped cluster

Choreo_Util_AllianceFlipRotation2d

Flips a rotation2d cluster.

Inputs:

-- rotation - rotation2d - The rotation to flip.

Outputs:

-- FlippedRotation - rotation2d - The flipped cluster

Choreo_Util_AllianceFlipRotation3d

Flips a rotation3d cluster.

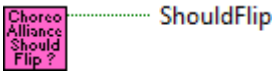
Inputs:

-- rotation - rotation3d - The rotation to flip.

Outputs:

-- FlippedRotation - rotation3d - The flipped cluster

Choreo_Util_AllianceFlipShouldFlip



Returns if a trajectory should be flipped. It returns true if the current alliance is RED. This calls the driver station to get the alliance color. As such, it should be called sparingly.

Inputs:

- none -

Output:

- ShouldFlip - boolean - TRUE if this robot is part of the RED alliance.

Choreo_Util_AllianceFlipTranslation2d

Flips a translation2d cluster.

Inputs:

-- translation - translation2d - The translation to flip.

Outputs:

-- FlippedTranslation - Translation2d - The flipped cluster

Choreo_Util_AllianceFlipTranslation3d

Flips a translation3d cluster.

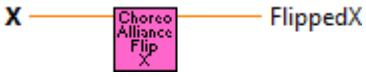
Inputs:

-- translation - translation3d - The translation to flip.

Outputs:

-- FlippedTranslation - Translation3d - The flipped cluster

Choreo_Util_AllianceFlipX



Flips the X coordinate.

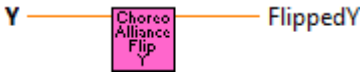
Inputs:

-- x - double - The X coordinate to flip.

Outputs:

-- FlippedX - double - The flipped X coordinate.

Choreo_Util_AllianceFlipY



Flips the Y coordinate.

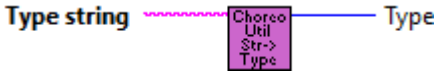
Inputs:

-- y - double - The Y coordinate to flip.

Outputs:

-- FlippedY - double - The flipped Y coordinate.

Choreo_Util_CommandTypeFromString



Get the command utility type enum from a string

If the string is not a valid command type, then "Unknown" is used.

Inputs:

-- Type string - string - string to evaluate for command util type

Outputs:

-- type - enum - Evaluated command util type.

Choreo_Util_DefaultDir

If given path is relative insert the default data directory for choreo trajectories. If given path is absolute, pass it through unchanged. The resulting path is always absolute.

Inputs:

-- PathIn - path - The path to check for absolute or relative.

Outputs:

-- PathOut - path - The resulting absolute path

Choreo_Util_FieldDimensions

Selector VI to get the field dimensions based on the competition year. The dimensions are in meters.

Selector:

-- competition year --

Inputs:

-- none --

Outputs:

-- length - double - length of playing field (meters)

-- width - double - width of playing field (meters)

Choreo_Util_FieldDimensions2026

Get the field dimensions for competition year 2026. The dimensions are in meters.

Inputs:

-- none --

Outputs:

-- length - double - length of playing field (meters)

-- width - double - width of playing field (meters)

Type Definitions

TypeDef

TypeDef-ChoreoAllianceFlipYearInfo

Choreo Alliance Flip Year info. Data needed to flip all or parts of a trajectory. Generally this is stored in a global variable. It can be updated to allow previous year's robots to continue to operate.

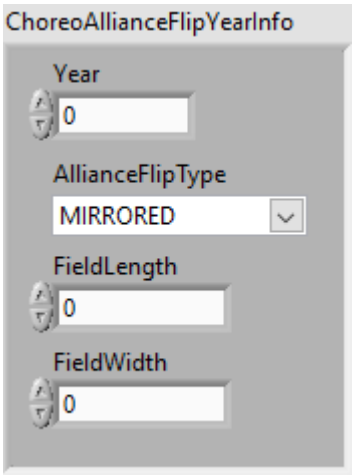
Contains:

- year - integer - year of this flip data.

- AllianceFlipType - enum - The type of flip to be performed.

- FieldLength - double - Field length (meters)

- FieldWidth - double - Field width (meters)



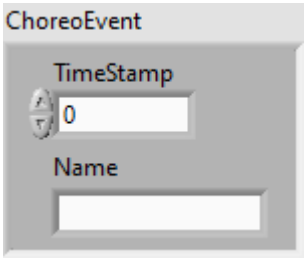
TypeDef-ChoreoEventMarker

Choreo Event data cluster

Contains:

- timestamp - double - timestamp (seconds) of this event.

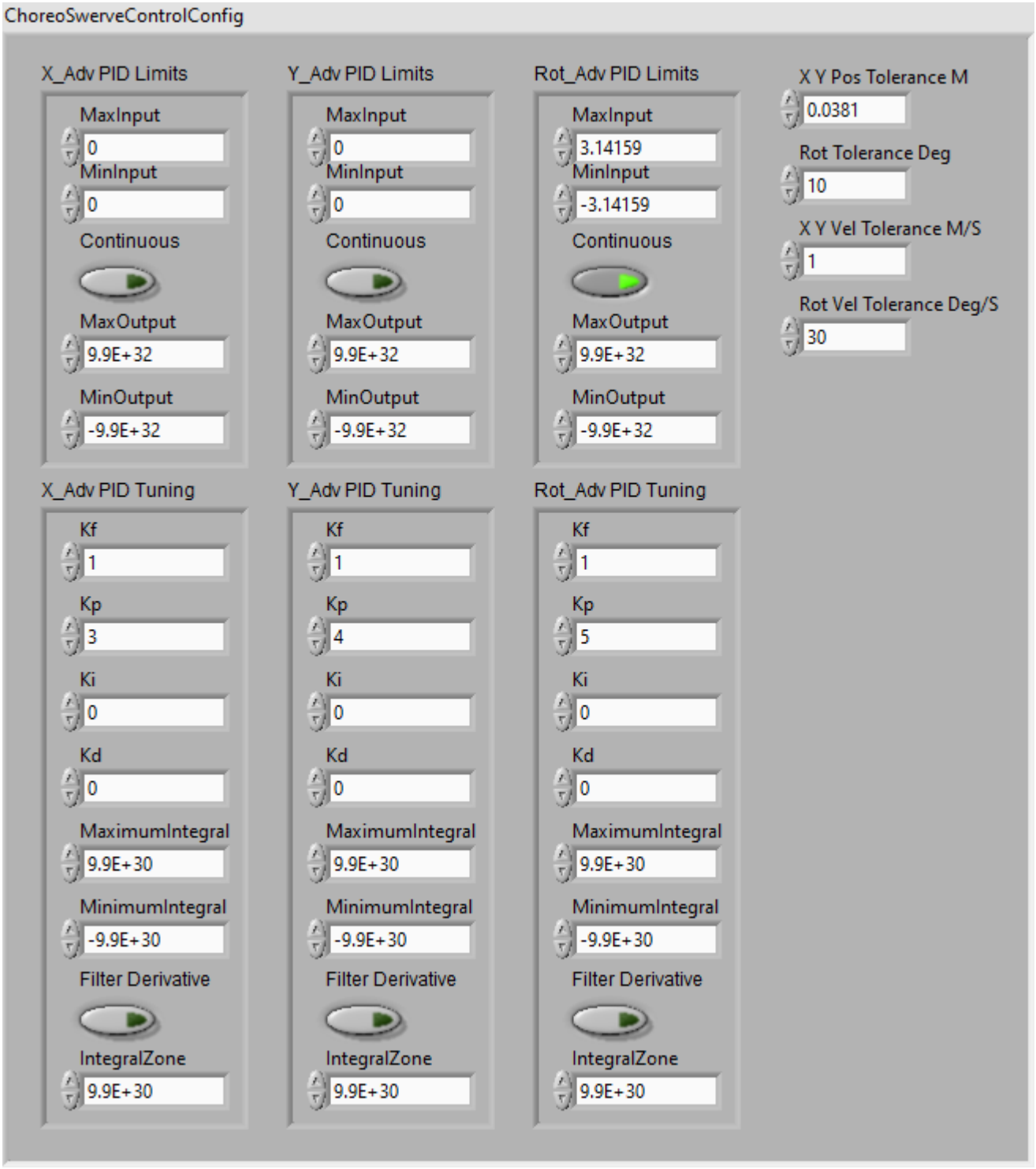
- name - string - the name of the event. Generally this will be the name of the command to issue.



TypeDef-ChoreoSwerveControlConfig

Controller configuration cluster for the ChoreoSwerveController

This cluster contains the configuration data for the 3 advanced PIDs. This includes limits, tuning, and tolerance configuration.



TypeDef-ChoreoSwerveSample

A single swerve robot sample in a Choreo Trajectory.

Contains:

- timestamp - double - The timestamp of this sample, relative to the beginning of the trajectory.

double

- x - double - The X position of the sample relative to the blue alliance wall origin in meters.

- y - double - The Y position of the sample relative to the blue alliance wall origin in meters.

- heading - double - The heading of the sample in radians, with 0 being in the +X direction.

- velocityX - double - The velocity of the sample in the X direction in m/s.

- velocityY - double - The velocity of the sample in the Y direction in m/s.

- angularVelocity - double - The angular velocity of the sample in rad/s.

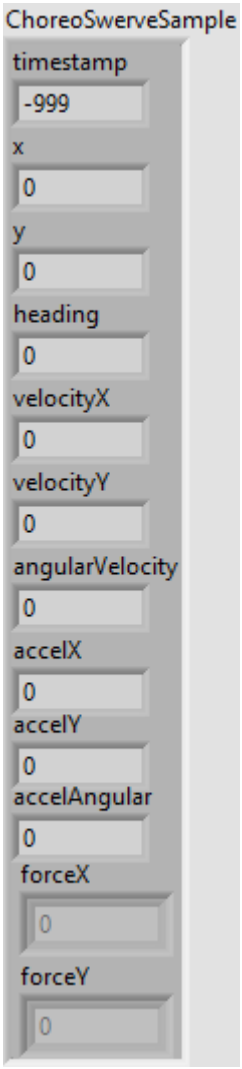
- accelX - double - The acceleration of the sample in the X direction in m/s².

- accelY - double - The acceleration of the sample in the Y direction in m/s².

- angularAccel - double - The angular acceleration of the sample in rad/s².

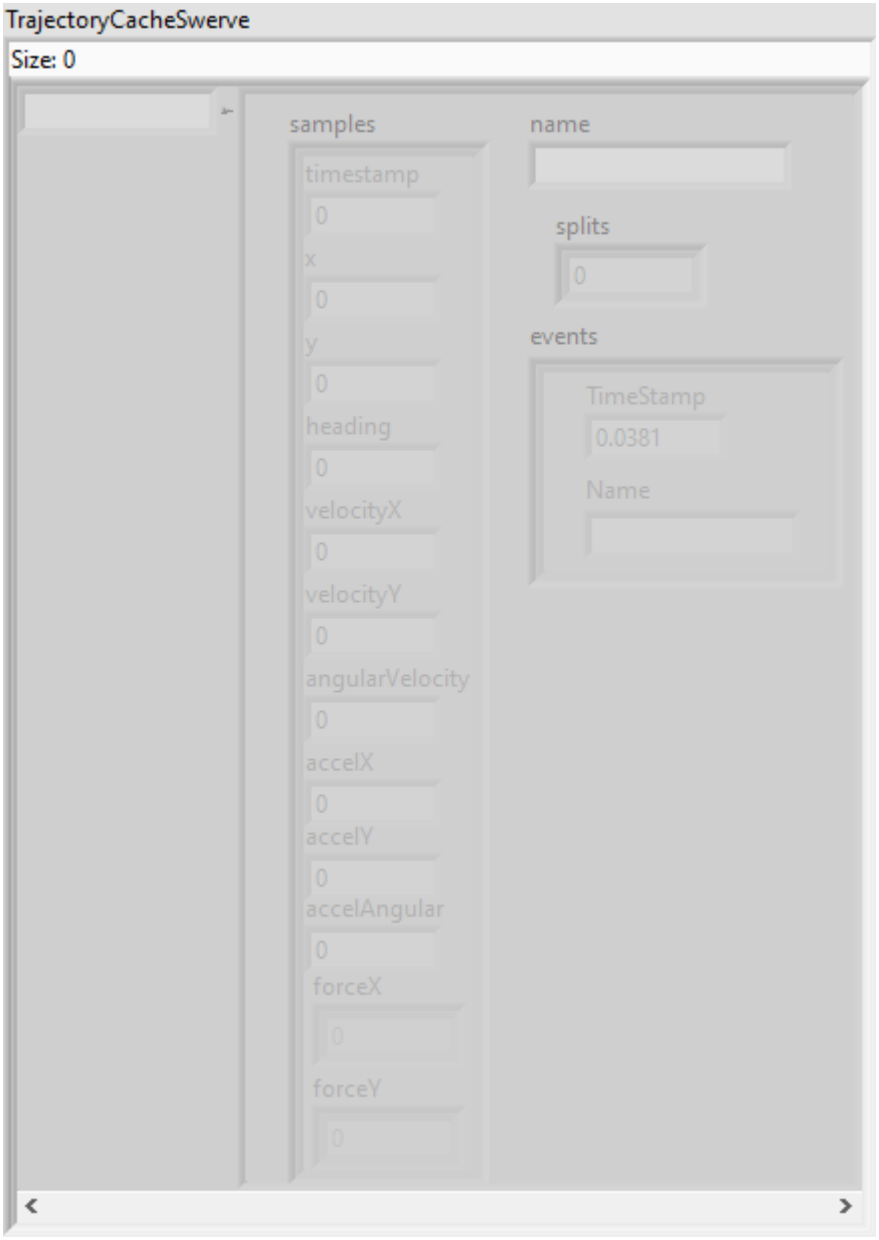
- forceX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- forceY - double array - The force on each swerve module in the Y direction in Newtons Module forces appear in the following order: [FL, FR, BL, BR].



TypeDef-ChoreoTrajectoryCacheSwerve

A map containing trajectories. This contains a hashed array of name, trajectory pairs. Trajectories can be access by name.



TypeDef-ChoreoTrajectorySwerve

A swerve trajectory loaded from Choreo.

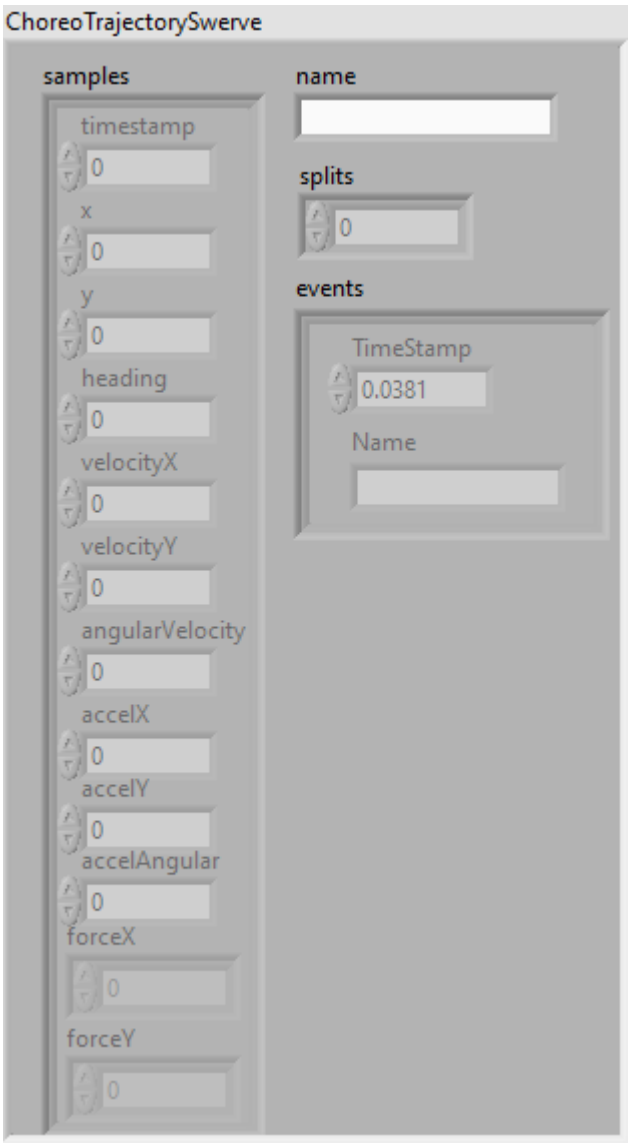
Contains:

- name - string - name of this trajectory. It should match the file name without the .traj extension

- splits - integer array - The indices of the splits in the trajectory

- events - cluster array - The events in the trajectory.

- samples - cluster array - The swerve trajectory samples of the trajectory.



Enumerated Type Definitions

Enum

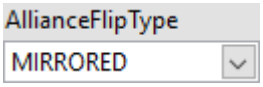
Enum-Choreo_Util_AllianceFlipType

Enumerated variable type for the type alliance flip to be performed.

The types are:

- MIRRORED - X becomes fieldLength - x, leaves the y coordinate unchanged, and heading becomes PI - heading.

- ROTATE_AROUND - X becomes fieldLength - x, Y becomes fieldWidth - y, and heading becomes PI - heading.



Enum-Choreo_Util_CommandType_ENUM

Enumerated variable type for the type of command contained in the Command Util cluster. This is converted from a string contained in the path JSON.

The types are:

- Unknown
- Wait
- Named
- Path
- Sequential
- Parallel

- Race
- Deadline
- PointTowardsZoneEnter
- PointTowardsZoneExit
- TriggerEventStart
- TriggerEventStop
- TriggerOneShot