ChoreoLib

LabVIEW

Reference

Version 2026.0.0

Table of Contents

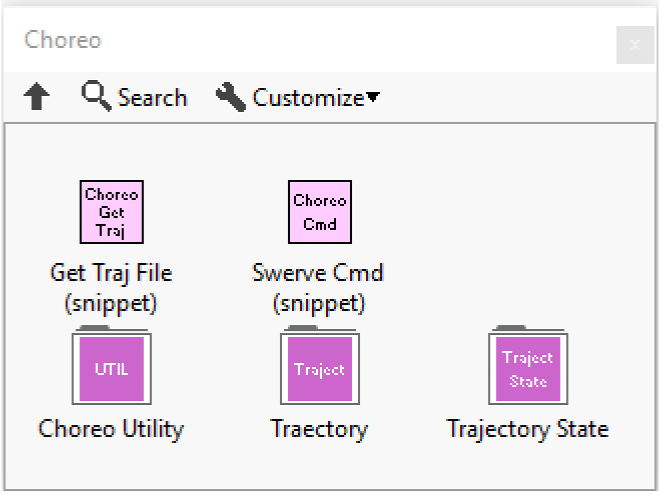
Introduction

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

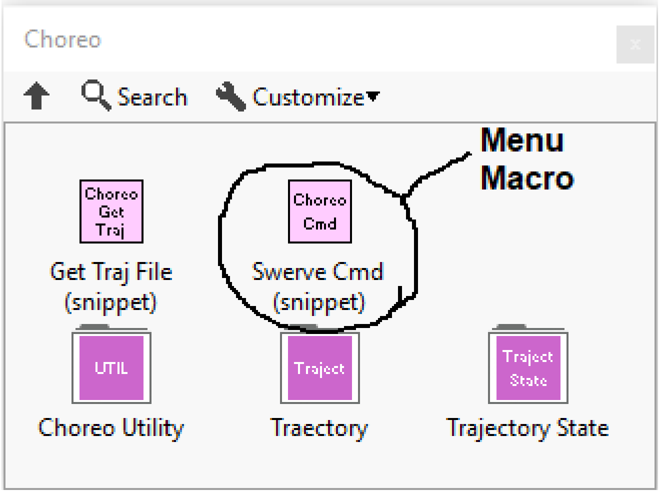
Function Menus

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



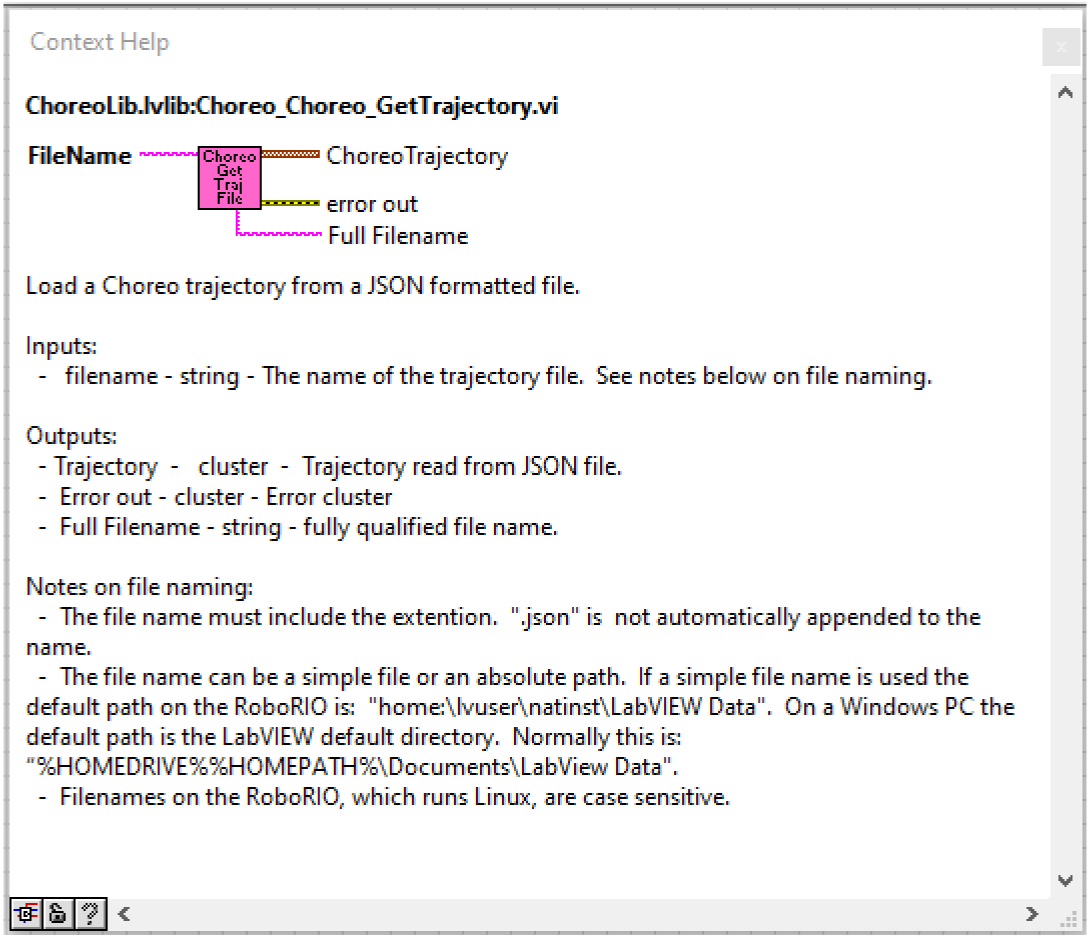
Menu Macros

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).



ChoreoLib.lvlib:Choreo_Choreo_GetTrajectory.vi

Load a Choreo trajectory from a JSON formatted file.

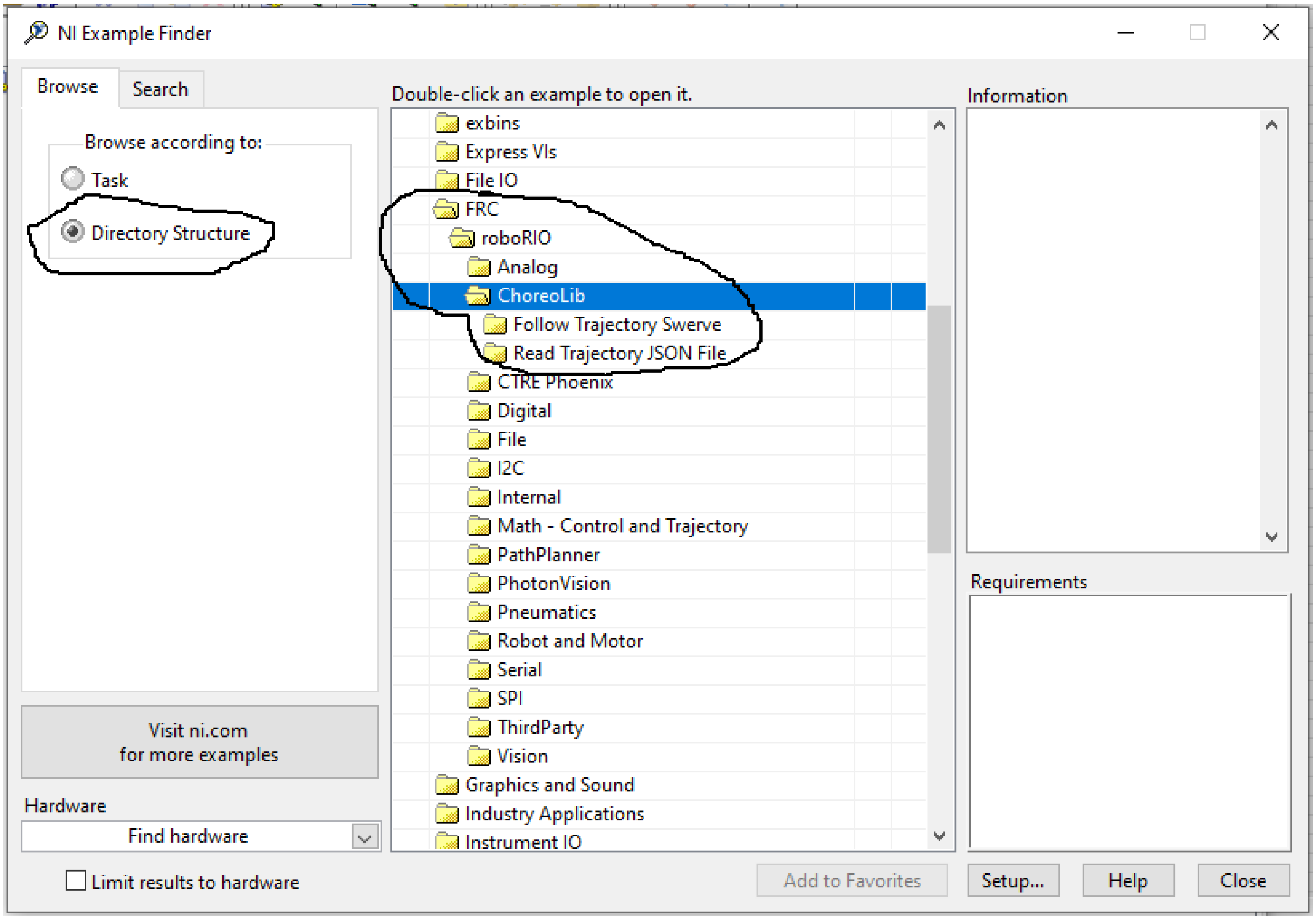
Inputs:
- filename - string - The name of the trajectory file. See notes below on file naming.

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 name must include the extention. ".json" is not automatically appended to the name.
- 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". On a Windows PC the default path is the LabVIEW default directory. Normally this is: "%HOMEDRIVE%%HOMEPATH%\Documents\LabView Data".
- Filenames on the RoboRIO, which runs Linux, are case sensitive.

Function Examples

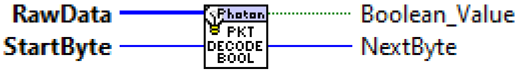
Many of the functions have examples that can be found under the LabVIEW "Find examples..." function. (Help -> Find Examples...). The function examples are easiest to find when "Directory Structure" is selected.


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.

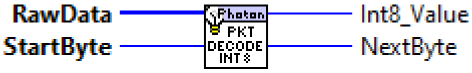
- StartByte -- The starting byte number (the first byte starts at 0)

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.

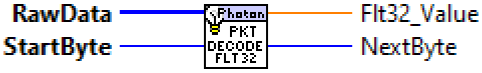
- StartByte -- The starting byte number (the first byte starts at 0)

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.

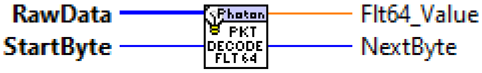
- StartByte -- The starting byte number (the first byte starts at 0)

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.

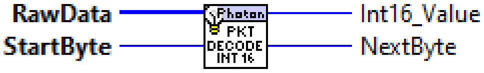
- StartByte -- The starting byte number (the first byte starts at 0)

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.

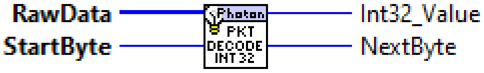
- StartByte -- The starting byte number (the first byte starts at 0)

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.

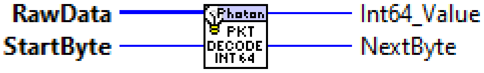
- StartByte -- The starting byte number (the first byte starts at 0)

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.

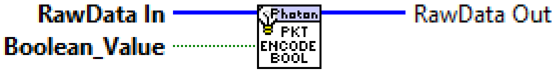
Inputs

- 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.

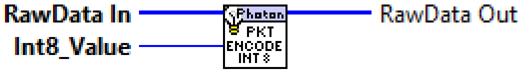
Inputs

- 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

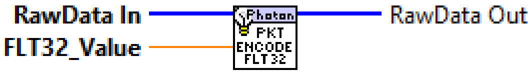
- RawData In -- Array of bytes to be appended to..

- 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

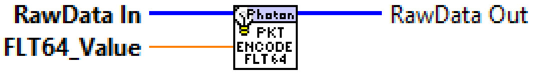
- RawData In -- Array of bytes to be appended to..

- 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

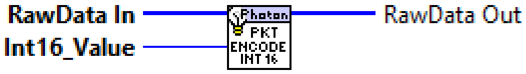
- RawData In -- Array of bytes to be appended to..

- 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

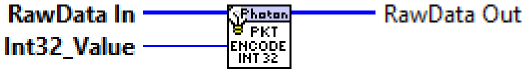
- RawData In -- Array of bytes to be appended to..

- 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.

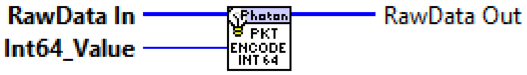
Inputs

- 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

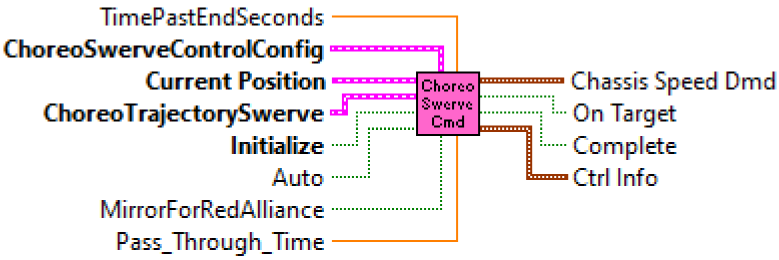
- RawData In -- Array of bytes to be appended to..
- In64_Value -- 64 bit integer the form of an INT 64 value.

Outputs

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

Choreo

Choreo_Choreo_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.

- ChoreoTrajectory - cluster - Choreotrajectory list

- 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:

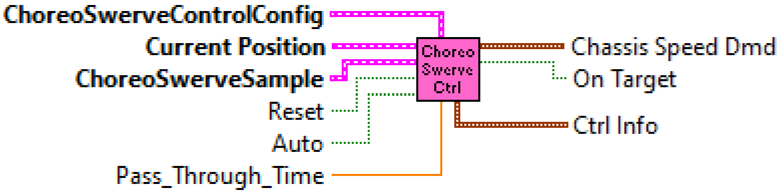
- Choreo Speed Demand - chassis speed - Desired chassis speed (M/S, Rad/Sec)

- 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)

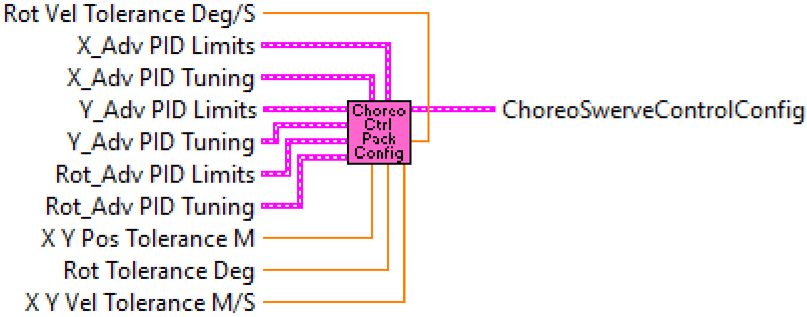
Outputs:

- 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:

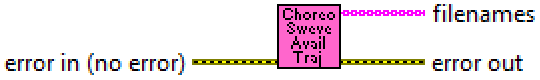
- X Adv PID Limits - cluster - Input and output limits for the X position PID (Optional.)
- X Adv PID Tuning - cluster - Tuning values for the X position PID (optional)
- Y Adv PID Limits - cluster - Input and output limits for the Y position PID (Optional.)
- Y Adv PID Tuning - cluster - Tuning values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



Load a Choreo trajectory from a JSON formatted file.   Note that for newer versions of Choreo, the file has a ".traj" extension. Internally this is formatted as a JSON file.

Inputs:

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

Outputs:

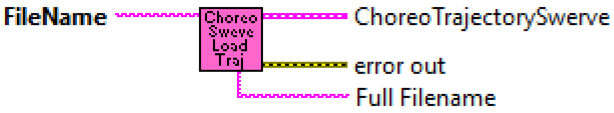
- Trajectory -   cluster - Trajectory read from JSON file.
- Error out - cluster - Error cluster
- Full Filename - string - fully qualified file name.

Notes on file naming:

- The file name must include the extention. ".json", or ".traj" is not automatically appended to the name.
- 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". On a Windows PC the default path is the LabVIEW default directory. Normally this is: %HOMEDRIVE%%HOMEPATH%\Documents\LabView Data".
- Filenames on the RoboRIO, which runs Linux, are case sensitive.

Choreo_ChoreoSwerve_Globals

Choreo_ChoreoSwerve_LoadTrajectory



Load a Choreo trajectory from a JSON formatted file. Note that for newer versions of Choreo, the file has a ".traj" extension. Internally this is formatted as a JSON file.

Inputs:

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

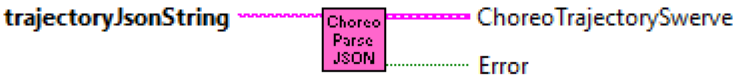
Outputs:

- 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 name must include the extention. ".json", or ".traj" is not automatically appended to the name.
- 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". On a Windows PC the default path is the LabVIEW default directory. Normally this is: %HOMEDRIVE%%HOMEPATH%\Documents\LabView Data".
- Filenames on the RoboRIO, which runs Linux, are case sensitive.

Choreo_ChoreoSwerve_LoadTrajectoryString



Parse a choreo trajectory from a JSON formatted string.

Inputs:

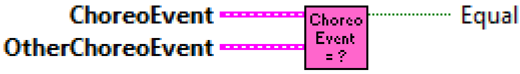
- JSON String - string- The string containing the choreo trajectory. Note that files may have a .TRAJ extension. These are actually internally formatted as JSON files.

Outputs:

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

EventMarker

Choreo_EventMarker_Equals



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

- ax - double - The acceleration of the sample in the X direction in m/s^2.

- ay - double - The acceleration of the sample in the Y direction in m/s^2.

- alpha - double - The angular acceleration of the sample in rad/s^2.

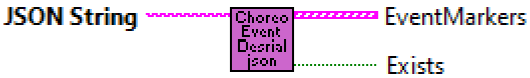
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_EventMarker_FromJSON



Create a list 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.

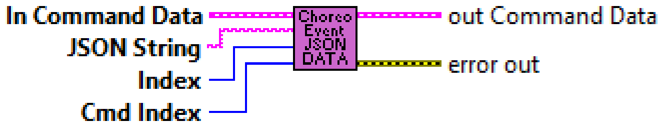
Notes:

Currently the only type of commands that can be issued are Boolean commands. The value is forced to TRUE when the command is issued.

The following table lists the actions performed for different types of commands:

- Unknown - nothing - not supported.
- Wait - nothing - not supported
- Named - Issue boolean command with TRUE value using the "name" as the command name.
- Path - nothing - not supported
- Sequential, Parallel, Race, Deadline - Issue boolean command with TRUE value for each of the commands contained in the "commands" array.   The value for "name" is used as the command name.

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 the only type of commands that can be issued are Boolean commands. The value is forced to TRUE when the command is issued.

The following table lists the actions performed for different types of commands:

- Unknown - nothing - not supported.

- Wait - nothing - not supported

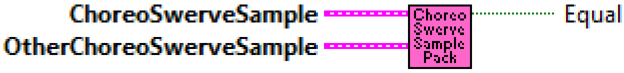
- Named - Issue boolean command with TRUE value using the "name" as the command name.

- Path - nothing - not supported

- Sequential, Parallel, Race, Deadline - Issue boolean command with TRUE value for each of the commands contained in the "commands" array. The value for "name" is used as the command name.

SwerveSample

Choreo_SwerveSample_Equals



Constructs a SwerveSample with the specified parameters.

*/

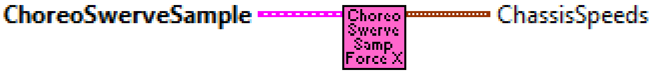
Inputs:

- t - 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.
- vx - double - The velocity of the sample in the X direction in m/s.
- vy - double - The velocity of the sample in the Y direction in m/s.
- omega - double - The angular velocity of the sample in rad/s.
- ax - double - The acceleration of the sample in the X direction in m/s^2.
- ay - double - The acceleration of the sample in the Y direction in m/s^2.
- alpha - double - The angular acceleration of the sample in rad/s^2.
- 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].
- moduleForcesY - 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_GetChassisSpeeds



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

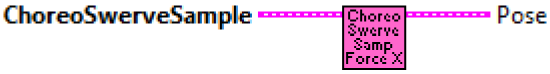
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_GetPose



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

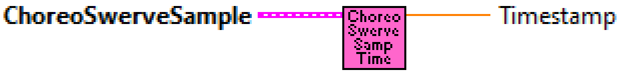
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_GetTimestamp



Constructs a SwerveSample with the specified parameters.

*/

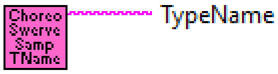
Inputs:

- t - 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.
- vx - double - The velocity of the sample in the X direction in m/s.
- vy - double - The velocity of the sample in the Y direction in m/s.
- omega - double - The angular velocity of the sample in rad/s.
- ax - double - The acceleration of the sample in the X direction in m/s^2.
- ay - double - The acceleration of the sample in the Y direction in m/s^2.
- alpha - double - The angular acceleration of the sample in rad/s^2.
- 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].
- moduleForcesY - 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_GetTypeName



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

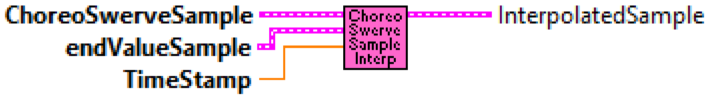
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_Interpolate



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

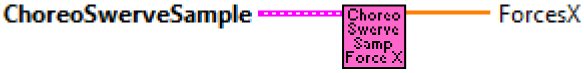
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_ModuleForcesX



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

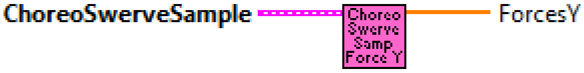
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_ModuleForcesY



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

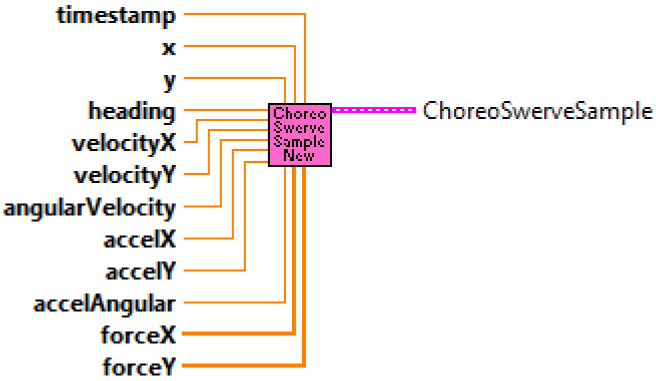
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_New



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

- ax - double - The acceleration of the sample in the X direction in m/s^2.

- ay - double - The acceleration of the sample in the Y direction in m/s^2.

- alpha - double - The angular acceleration of the sample in rad/s^2.

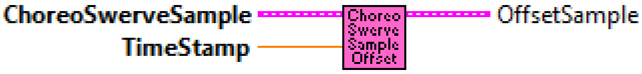
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

- ax - double - The acceleration of the sample in the X direction in m/s^2.

- ay - double - The acceleration of the sample in the Y direction in m/s^2.

- alpha - double - The angular acceleration of the sample in rad/s^2.

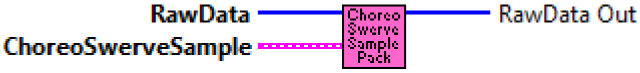
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_Pack



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

- ax - double - The acceleration of the sample in the X direction in m/s^2.

- ay - double - The acceleration of the sample in the Y direction in m/s^2.

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

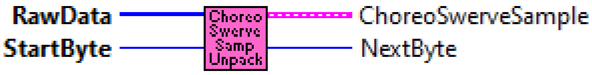
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_Unpack



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

- 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].

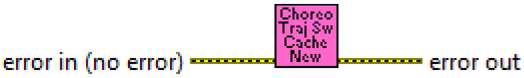
- moduleForcesY - double array - The force on each swerve module in the Y direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

Outputs:

- ChoreoSwerveSample - cluster - Cluster containing sample data

TrajectoryCacheSwerve

Choreo_TrajectoryCacheSwerve_Clear



Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

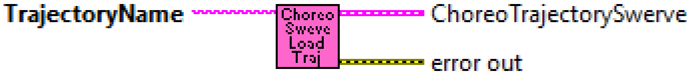
- moduleForcesX - double array - The force on each swerve module in the X direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

- moduleForcesY - 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_TrajectoryCacheSwerve_LoadTrajectory



Load a Choreo trajectory from a JSON formatted file.   Note that for newer versions of Choreo, the file has a ".traj" extension.   Internally this is formatted as a JSON file.

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 name must include the extention. ".json", or ".traj" is not automatically appended to the name.

- 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". On a Windows PC the default path is the LabVIEW default directory. Normally this is: %HOMEDRIVE%%HOMEPATH%\Documents\LabView Data".

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

Choreo_TrajectoryCacheSwerve_New

Constructs a SwerveSample with the specified parameters.

*/

Inputs:

- t - 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.

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

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

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

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

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

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

- 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].

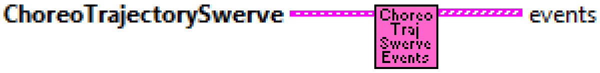
- moduleForcesY - double array - The force on each swerve module in the Y direction in Newtons. Module forces appear in the following order: [FL, FR, BL, BR].

Outputs:

- ChoreoSwerveSample - cluster - Cluster containing sample data

TrajectorySwerve

Choreo_TrajectorySwerve_Events



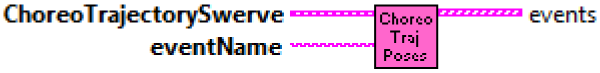
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_GetEvents



Return all posse of this trajectory

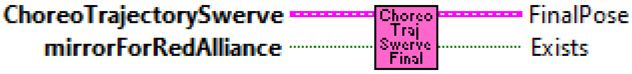
Inputs:

- ChoreoTrajectory - cluster - Input trajectory

Outputs:

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

Choreo_TrajectorySwerve_GetFinalPose



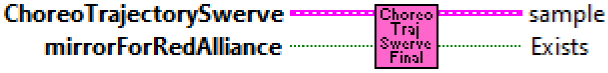
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_GetFinalSample



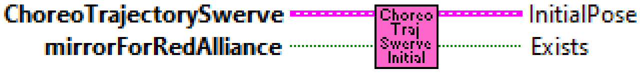
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_GetInitialPose



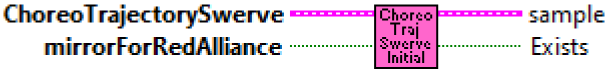
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_GetInitialSample



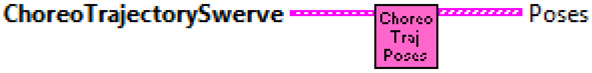
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_GetPoses



Return all posse of this trajectory

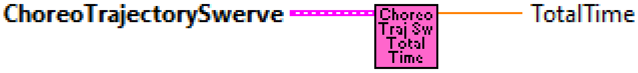
Inputs:

- ChoreoTrajectory - cluster - Input trajectory

Outputs:

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

Choreo_TrajectorySwerve_GetTotalTime



Return total time duration of this trajectory

Inputs:

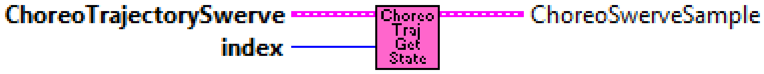
- ChoreoTrajectory - cluster - Input trajectory

Outputs:

- TotalTime - double - The total time duration of this trajectory.

.

Choreo_TrajectorySwerve_GetTrajSample



Return a single trajectory state from the trajectory.

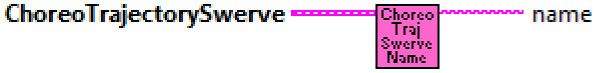
Inputs:

- ChoreoTrajectory - cluster - Input trajectory
- Index - integer - index into the trajectory state array. Beginning index is zero.

Outputs:

- Trajectory state - cluster - the selectred trajectory state..

Choreo_TrajectorySwerve_Name



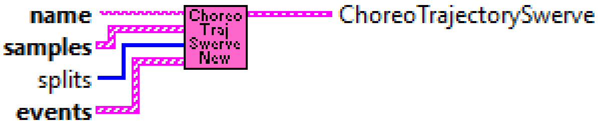
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_New



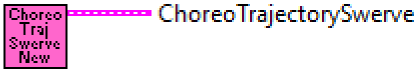
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_New_Empty



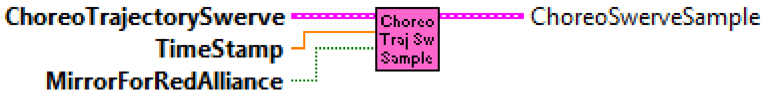
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_SampleAt



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

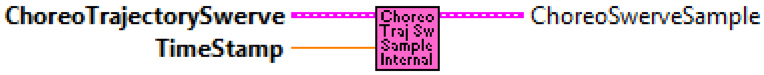
Inputs:

- 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



This is an internal routine. It should NOT be called by the end user. Return an interpolated sample of the trajectory at the given timestamp.

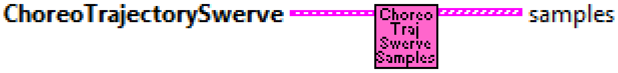
Inputs:

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

Outputs:

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

Choreo_TrajectorySwerve_Samples



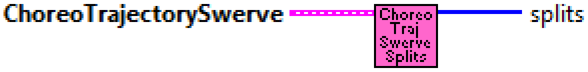
Create an empty trajectory

Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Choreo_TrajectorySwerve_Splits



Create an empty trajectory

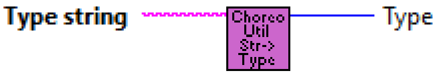
Inputs:

Outputs:

- ChoreoTrajectory - array - Empty array of Trajectory states.

Util

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.

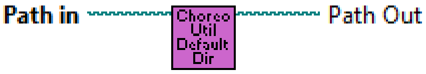
Inputs:

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

Outputs:

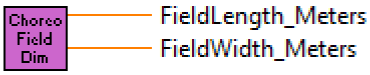
-- type - enum - Evaluated command util type.

Choreo_Util_DefaultDir

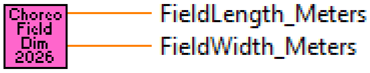


If given path is relative insert the default data directory. If given path is absolute, pass it through unchanged. The resulting path is always absolute.

Choreo_Util_FieldDimensions



Choreo_Util_FieldDimensions2026



trajectory, mirrored across the field midline.

Inputs:

- ChoreoTrajectory - cluster - Input trajectory

Outputs:

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

Type Definitions

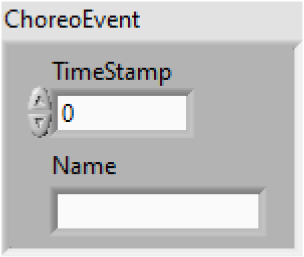
TypeDef

TypeDef-ChoreoEventMarker

Choreo
Event

Controller configuration cluster for the ChoreoSwerveController

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



ChoreoEvent

TimeStamp: 0

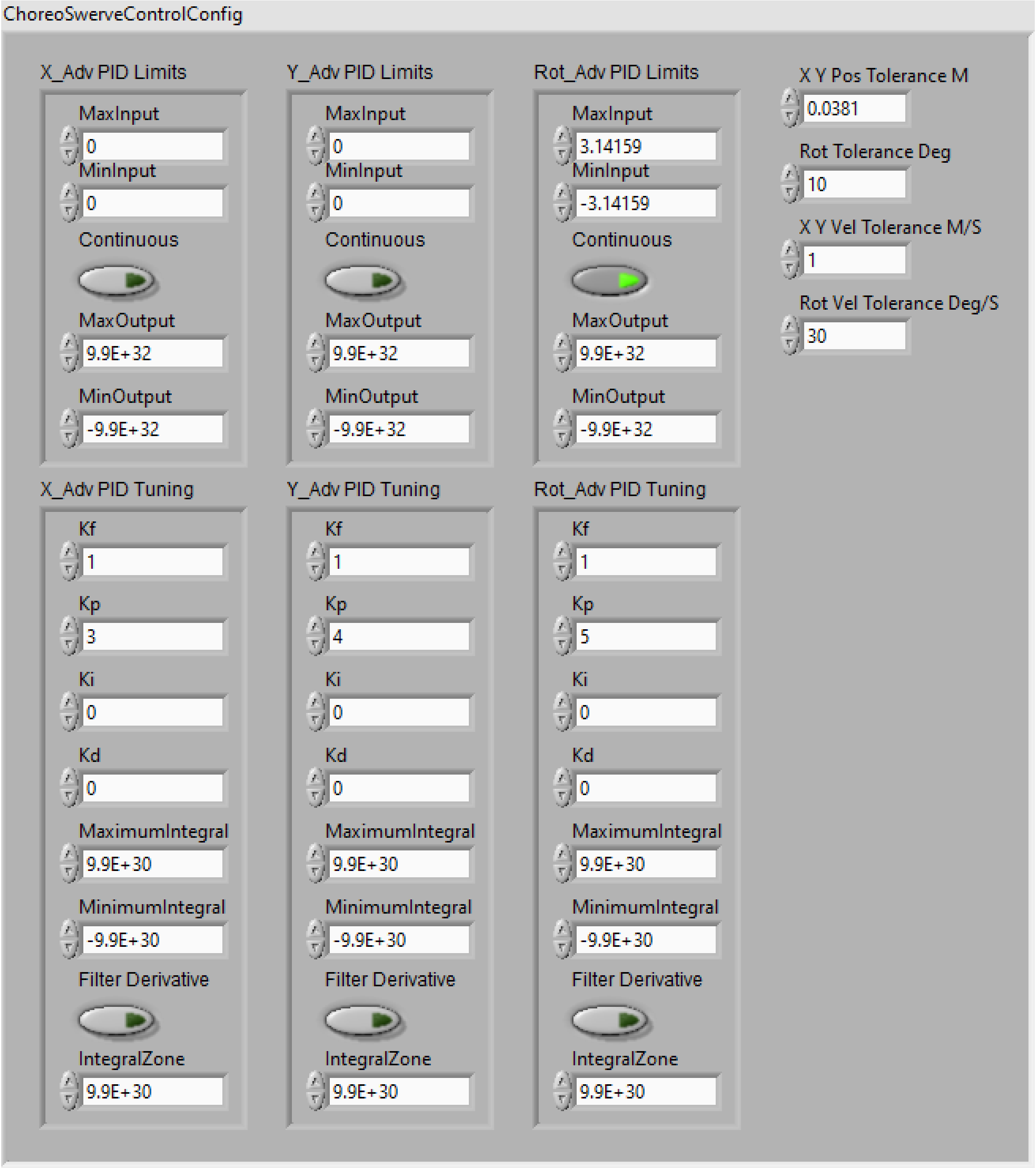
Name:

TypeDef-ChoreoSwerveControlConfig

Choreo
Swerve
Control
Config

Controller configuration cluster for the ChoreoSwerveController

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



ChoreoSwerveControlConfig

X_Adv PID Limits

- MaxInput: 0
- MinInput: 0
- Continuous
- MaxOutput: 9.9E+32
- MinOutput: -9.9E+32

Y_Adv PID Limits

- MaxInput: 0
- MinInput: 0
- Continuous
- MaxOutput: 9.9E+32
- MinOutput: -9.9E+32

Rot_Adv PID Limits

- MaxInput: 3.14159
- MinInput: -3.14159
- Continuous
- MaxOutput: 9.9E+32
- MinOutput: -9.9E+32

X Y Pos Tolerance M: 0.0381

Rot Tolerance Deg: 10

X Y Vel Tolerance M/S: 1

Rot Vel Tolerance Deg/S: 30

X_Adv PID Tuning

- Kf: 1
- Kp: 3
- Ki: 0
- Kd: 0
- MaximumIntegral: 9.9E+30
- MinimumIntegral: -9.9E+30
- Filter Derivative
- IntegralZone: 9.9E+30

Y_Adv PID Tuning

- Kf: 1
- Kp: 4
- Ki: 0
- Kd: 0
- MaximumIntegral: 9.9E+30
- MinimumIntegral: -9.9E+30
- Filter Derivative
- IntegralZone: 9.9E+30

Rot_Adv PID Tuning

- Kf: 1
- Kp: 5
- Ki: 0
- Kd: 0
- MaximumIntegral: 9.9E+30
- MinimumIntegral: -9.9E+30
- Filter Derivative
- IntegralZone: 9.9E+30

TypeDef-ChoreoSwerveSample

Choreo
Swerve
Sample

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.

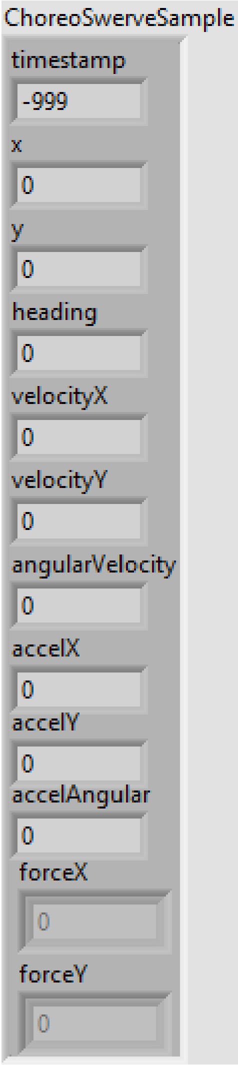
- accelX - double - The acceleration of the sample in the X direction in m/s².

- 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].



ChoreoSwerveSample

timestamp
-999

x
0

y
0

heading
0

velocityX
0

velocityY
0

angularVelocity
0

accelX
0

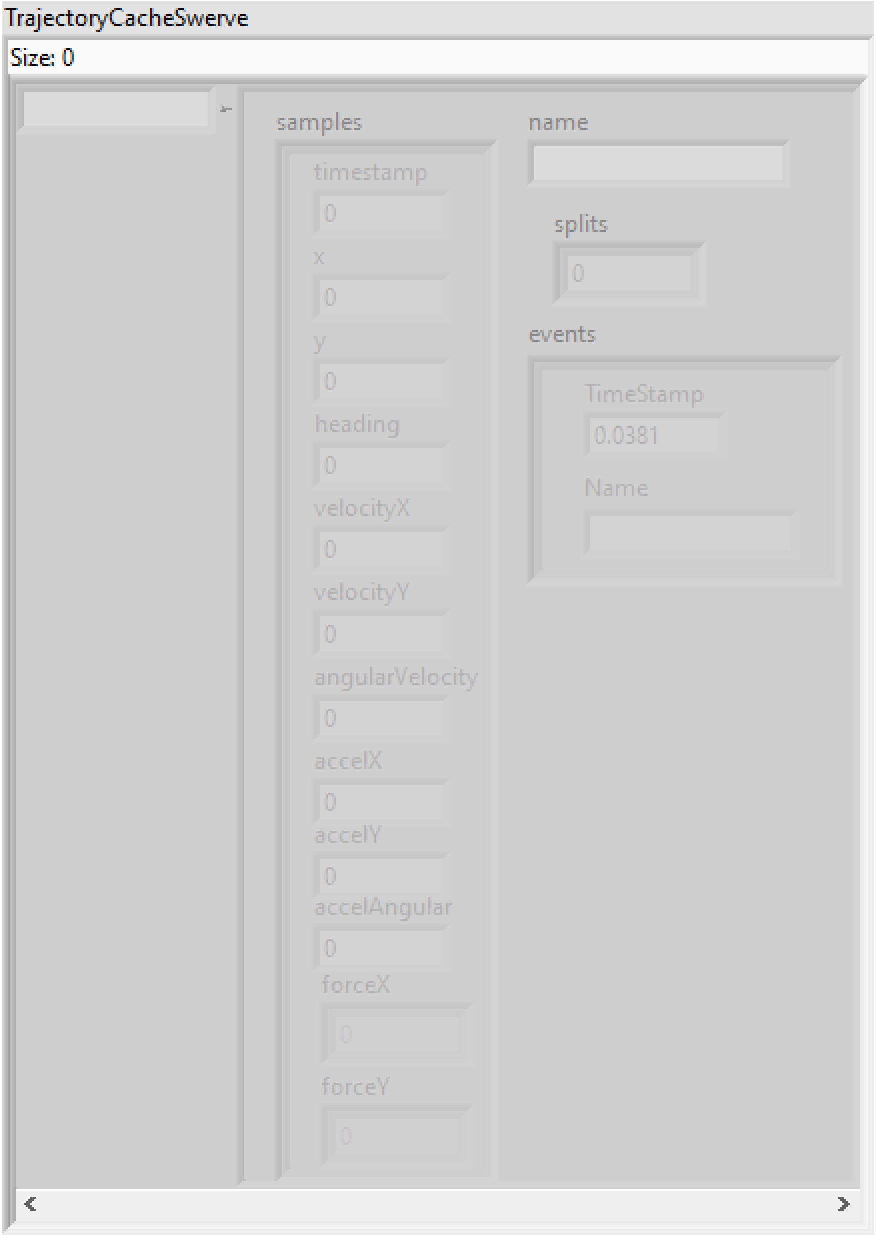
accelY
0

accelAngular
0

forceX
0

forceY
0

TypeDef-ChoreoTrajectoryCacheSwerve



TrajectoryCacheSwerve

Size: 0

samples

timestamp: 0

x: 0

y: 0

heading: 0

velocityX: 0

velocityY: 0

angularVelocity: 0

accelX: 0

accelY: 0

accelAngular: 0

forceX: 0

forceY: 0

name

splits: 0

events

TimeStamp: 0.0381

Name

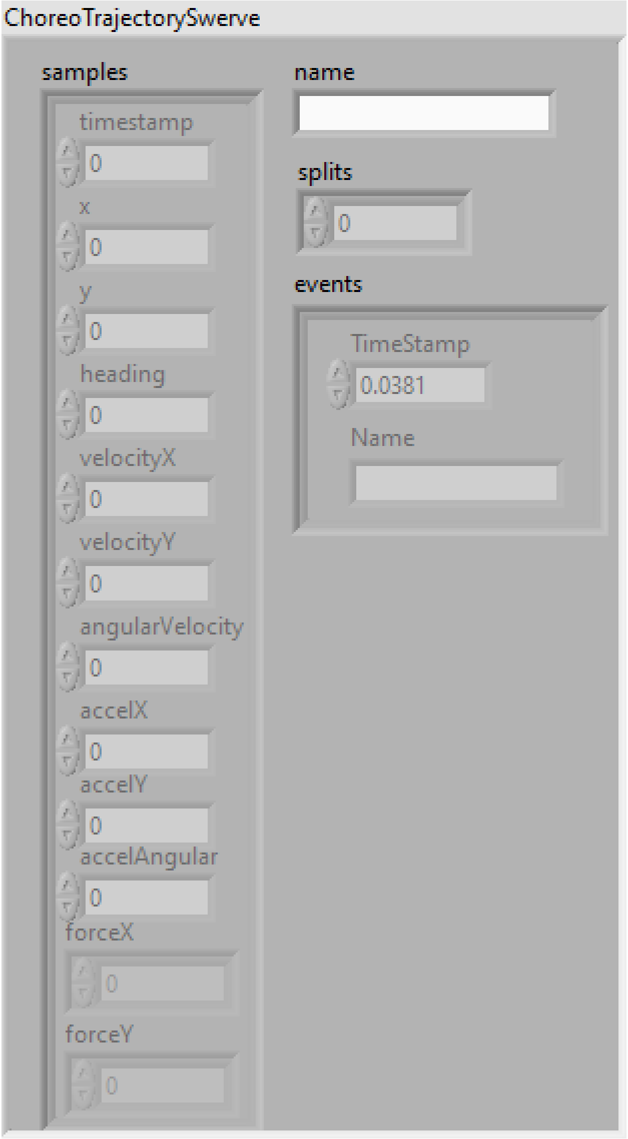
TypeDef-ChoreoTrajectorySwerve

Choreo Traj Swerve

A trajectory loaded from Choreo.

Contains:

- ChoreoTrajectory - array of ChoreoTrajectoryState



ChoreoTrajectorySwerve

samples

timestamp 0

x 0

y 0

heading 0

velocityX 0

velocityY 0

angularVelocity 0

accelX 0

accelY 0

accelAngular 0

forceX 0

forceY 0

name

splits 0

events

TimeStamp 0.0381

Name

Enumerated Type Definitions

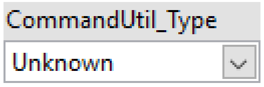
Enum

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



CommandUtil_Type

Unknown