PLCopen Basics with MPiec Controllers
Remote Demo Connection

Connection Concept
Secomea
Login
Connect to MPiec controller
Connect to IP camera
Demo Overview
Tips
Remote Demo Connection

- **Site Manager**
- **Gate Manager**
- **Link Manager**
  - Application on your PC
- **Remote Connection**
  - Login credentials
  - Electronic certificate
Remote Demo Connection

- More Information at www.secomea.com
Remote Demo Connection

- **Contact Yaskawa**
  - training@Yaskawa.com
  - www.Yaskawa.com Request Training
- **Receive email**
- **Install Link Manager software**
- **First-time Login**
  - Start Link Manager
  - Certificate
  - Password

**Process Summary**

- **Test Mpiec connection**
  - Internet Explorer browser
- **Test IP Camera connection**
  - Port 88
  - Plugin installation
    - Internet Explorer 11
    - FireFox
    - Chrome
    - Opera
Hello eLV Student

Hello from Yaskawa Technical Training Services

This mail contains your X.509 user certificate for the Secomea LinkManager. The password associated with the certificate is: dKQuquAK4809

You must save the attached file, eLV_Student.lmc, to your local hard drive (or other suitable storage) before you can import it into the LinkManager.

Installing a new LinkManager

First you need to download and install the latest LinkManager software for your Windows system:

For 32-bit windows: http://ftp.secomea.com/pub/LinkManager-Setup.exe
For 64-bit windows: http://ftp.secomea.com/pub/LinkManager-x64-Setup.exe

When the installation completes, you will be asked to install the user certificate, after which you can proceed to Login.

Installing a LinkManager User certificate
Remote Demo Connection

Install Link Manager Software
Remote Demo Connection

- **Start Link Manager**
Remote Demo Connection

- **Certificate**

Please install LinkManager User Certificate.

The GateManager administrator has sent you an email which contains a LinkManager User Certificate file (file type is .imc).

Press the "Browse" button to select the certificate file from your local computer, fill in the certificate's password, and press "Install".

- Certificate file: 
- Password: 
  - Remember password

Install  |  About

eLV_Student.imc
3.3 KB
Remote Demo Connection

- **Password**

---

First Time Login

---

Please install LinkManager User Certificate

The GateManager administrator has sent you an email which contains a LinkManager User Certificate file.

Press the "Browse" button to select the certificate file from your local computer, fill in the certificate’s password, and press "Install".

Certificate file: C:\Users\student\Downloads\eLV_Student.lmc

Password: 

- Remember password

First you need to download and install the latest LinkManager software for your Windows system.
Remote Demo Connection

First Time Login

- eLV Student [TRAIN15-E6420]
  - station2: MP2300SiecDemo3 (SiteManager) - 192.168.15.23
  - station2: TTScam2 (SiteManager) - 192.168.15.242

- Show all
- Refresh

<table>
<thead>
<tr>
<th>Agent</th>
<th>Address</th>
<th>Status</th>
<th>Connects</th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>ok</td>
<td>tx</td>
<td>rx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fail</td>
<td>tx</td>
<td>rx</td>
</tr>
<tr>
<td>MP2300SiecDemo3</td>
<td>192.168.15.23</td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(udp)</td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TRAIN15-E6420</td>
<td></td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TTScam2</td>
<td>192.168.15.242</td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(udp)</td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>TRAIN15-E6420</td>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Round-trip time: Min: 21.5 ms, Avg: 40.9 ms, Max: 104.7 ms
Remote Demo Connection

Test MPiec Connection

WebServer eLearning Modules
- eLV.Mpiec.01.WebServer
- eLV.MP2300iec.01.Setup
- eLV.MP2600iec.01.Setup
- eLV.MPiec.01.ProjectArchive
- eLV.MPiec.01.Firmware
Remote Demo Connection

- Port 88
Remote Demo Connection

- **Plugin – ie 11**
Remote Demo Connection

- **Plugin – Firefox**
Remote Demo Connection

- **Plugin – Chrome**
Remote Demo Connection

- Plugin – Opera
Remote Demo Connection

Test IP Camera Connection

Indoor Pan/Tilt/Zoom IP Camera
Driving Value

Class Project Template

Purpose
Save the Hardware Configuration
Hardware Configuration Summary
Project Overview
Run Project
Toggle Boolean Interface
Starting Point for PLCopen training

Hardware Configuration knowledge not required
- You will learn some basics anyway

Provide an input interface
- IP camera can’t turn on the switches!
• **Secomea connected**
  - MPiec web page in i.e.
  - IP camera in Opera

• **Class project file *.zwt downloaded from description page**
  - PLCopen2300sPro2 RevX
  - PLCopen2300sPro3 RevX
  - PLCopen2600Pro2 RevX
  - PLCopen2600Pro3 RevX

• **MotionWorks IEC Pro installed**
  - Prefer Version 3.x
  - Version 2.x very similar
Class Project Template

- **Hardware Configuration**
  - On the Yaskawa toolbar – Move the toolbar
- **Save Project Hardware Configuration to Controller**
  - IP address, Connect HC to controller
  - Use Offline Configuration
  - SAVE online
  - Reboot controller
Hardware Configuration Summary: Near Default
- Made from default controller with default servos
- Encoders set for incremental mode
- Axes
  - Screw, 360mm/rev
  - Rotary, 360 deg/rev, 360 mach cycle
  - External, pulses
- OT disabled
Tour the Class Template Project

- Based on File-New templates
- Libraries
  - One library is used exclusively for data types
- DataTypes
  - No local datatypes
- Logical POUs
  - Toggle function block
  - HMI with TB inputs to Global variables for your use
- Tasks
  - Template tasks
- Global Variables
  - Servo axis
  - L-IO
  - Created by Hardware Configuration
• **MotionWorks IEC-Pro Version 3**
  - Add Download changes button
    - Extras – Options – Commands – Compile/Debug
    - Drag to toolbar
Class Project Template

- Make
- Download
- Coldstart
Class Project Template

- **Debug Mode**
- **Open Worksheet**
- **Toggle Boolean**
  - Setting is lost when worksheet is closed
PLCopen Overview

- Initiatives
- Summary
- Motion State Diagram
- General Rules
- Initial Value
- Done Output
PLCopen Overview

PLCopen
for efficiency in automation

PLCopen Motion Control
PLCopen Safety
PLCopen Benchmarking
PLCopen XML
PLCopen Reusability Level
PLCopen Conformity Level

IEC 61131-3

Certified Training

No. 0000
21.06.2001
LD
Datatypes supported: 26 of 26
PLCopen Overview

- **Defines libraries of Function Blocks**
  - Motion control specification
- **YASKAWA MotionWorks IEC**
  - Complies with PLCopen
  - Proprietary internal algorithms

**TC2 – Motion Control**

**Overview**

- **PLCopen**
- **Implementation**

```
<table>
<thead>
<tr>
<th>Axis Ref</th>
<th>Axis</th>
<th>Axis</th>
<th>Axis Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOL</td>
<td>Execute</td>
<td>Done</td>
<td>BOOL</td>
</tr>
<tr>
<td>REAL</td>
<td>Position</td>
<td>CommandAborted</td>
<td>BOOL</td>
</tr>
<tr>
<td>REAL</td>
<td>Velocity</td>
<td>Error</td>
<td>BOOL</td>
</tr>
<tr>
<td>REAL</td>
<td>Acceleration</td>
<td>ErrorID</td>
<td>WORD</td>
</tr>
<tr>
<td>REAL</td>
<td>Deceleration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REAL</td>
<td>Jerk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MC Direction</td>
<td>Direction</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Axis</th>
<th>Execute</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Position</td>
<td>Busy</td>
</tr>
<tr>
<td></td>
<td>Velocity</td>
<td>Active</td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
<td>CommandAborted</td>
</tr>
<tr>
<td></td>
<td>Deceleration</td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>Jerk</td>
<td>ErrorID</td>
</tr>
<tr>
<td></td>
<td>Direction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BufferMode</td>
<td></td>
</tr>
</tbody>
</table>
```
• **MPiec Controllers**
  - MP2300iec – 20-Axis, I/O slot x1
  - MP2310iec – 20-Axis, I/O slot x3
  - MP2600iec – 1-Axis, Multi-function I/O
  - MP3200iec – 62-Axis, Mechatrolink III
  - MP3300iec – 16-Axis, Mechatrolink III

PLCopen programming is identical in each of the MPiec controllers.
PLCopen Overview

- **Firmware Level Operation**
  - Synchronized with Mechatrolink

- **MC_**
  - Defined by PLCopen

- **Y_**
  - Yaskawa Specific
PLCopen Overview

- **Motion States**
  - Synchronized Motion
  - Discrete Motion
  - Continuous Motion
  - Stopping
  - ErrorStop
  - Homing
  - Standstill
  - Disabled

- **PLCopen describes which blocks have priority and which state is active**
PLCopen Overview

- **PLCopen 2.4.1 specification**
  - Basic and important rules for how the motion control function blocks work
- **Use the following pages as reference**
  - PDF available
    - “tf_mc_part1_version10.pdf”
  - Download most recent versions at www.plcopen.org
### General Rules Summary

<table>
<thead>
<tr>
<th>Output status</th>
<th>The Done, InGear, InVelocity, Error, ErrorID and CommandAborted outputs are reset with the falling edge of execute. It must be guaranteed that they are set for at least one cycle if the corresponding situation occurs, even if execute was reset before. Done and Error outputs are mutually exclusive (cannot be true at the same time). If an instance of a FB receives a new execute before it finished (as a series of commands on the same instance), the FB won't return any feedback, like ‘Done’ or ‘CommandAborted’, for the previous action.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input parameters</td>
<td>The parameters are used with the rising edge of the execute input. To modify any parameter it is necessary to put the correct set of values and to trigger the motion again.</td>
</tr>
<tr>
<td>Missing input parameters</td>
<td>According to IEC 61131-3, if any parameter of a function block input is missing (“open”) then the value from the previous invocation of this instance will be used. In the first invocation the initial value is applied.</td>
</tr>
<tr>
<td>Position versus distance</td>
<td>“Position” is a value defined within a coordinate system. “Distance” is a relative measure related to technical units. “Distance” is the difference between two positions.</td>
</tr>
<tr>
<td>Sign rules</td>
<td>The Velocity, Acceleration, Deceleration and Jerk are always positive values. Position and Distance can be both positive and negative.</td>
</tr>
<tr>
<td>Error Handling Behavior</td>
<td>All blocks have two outputs which are dealing with errors that can occur while executing a Function Block. These outputs are defined as follows:</td>
</tr>
<tr>
<td></td>
<td><strong>Error</strong> Rising edge of Error informs that an error occurred during the execution of the Function Block.</td>
</tr>
<tr>
<td></td>
<td><strong>ErrorID</strong> Error number</td>
</tr>
<tr>
<td>Types of errors:</td>
<td>• Function blocks (e.g. parameters outside range, state machine) • Communication • Drive</td>
</tr>
<tr>
<td></td>
<td>Instance errors are not always resulting in an axis error (bringing the axis to standstill)</td>
</tr>
<tr>
<td>FB Naming</td>
<td>In case of multiple libraries within one system (to support multiple drive / motion control systems), the FB naming may be changed to “MC_FBname_SupplierID”.</td>
</tr>
<tr>
<td>Behavior of Done output</td>
<td>The Done output (as well as InGear, InVelocity,…) is set when the commanded action has been completed successfully. With multiple Function Blocks working on the same axis in a sequence, the following applies: when one movement on an axis is interrupted with another movement on the same axis without having reached the final goal, Done of the first FB will not be set.</td>
</tr>
<tr>
<td>Behavior of CommandAborted output</td>
<td>CommandAborted is set, when a commanded motion is interrupted by another motion command or MC_Stop. The reset-behavior of CommandAborted is like Done. When CommandAborted occurs, the other output signals like InVelocity are reset.</td>
</tr>
</tbody>
</table>
PLCopen Overview

• Examples of PLCopen Specification
  - Read inputs at rising edge only
  - Done exclusive of Error
  - Positive Velocity, Acceleration, Deceleration
  - Default input values

General Rules Summary

> 0

Done & Error never on at the same time
Right-Click any block for help

- The “Default” column is the initial value that will be used by the function block input if nothing is connected.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR_IN_OUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B Axis</td>
<td>AXIS_REF</td>
<td>Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).</td>
</tr>
<tr>
<td>VAR_INPUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B Execute</td>
<td>BOOL</td>
<td>Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.</td>
</tr>
<tr>
<td>B Distance</td>
<td>LREAL</td>
<td>Incremental distance (in user units)</td>
</tr>
<tr>
<td>E Velocity</td>
<td>LREAL</td>
<td>Absolute value of the velocity in user units/second</td>
</tr>
<tr>
<td>E Acceleration</td>
<td>LREAL</td>
<td>Value of the acceleration in user units/second ^ 2 (acceleration is applicable with same sign of torque and velocity)</td>
</tr>
</tbody>
</table>

Initial Value (Default)
PLCopen Overview

- **Done bit turns on**
  - At least 1 scan
  - At command completion
- **Done ≠ Position Complete**
  - AXn_PSET (global variable)
    - `/COIN`
    - `Pn522`

![Diagram](image)
Driving Value

Axis_Ref

Usage and Purpose
Definition
Axis Name and Number
Initialize Axis Variable
Axis

- **Data type AXIS_REF**
- **Data Structure**
- Allows for vendor-specific data to be combined into one variable
- **VAR_IN_OUT**
  - Input function
  - Data not copied in memory
- Required by all PLCopen function blocks

### Usage and Purpose

**Axis_Ref**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR_IN_OUT</td>
<td>AXIS_REF</td>
</tr>
</tbody>
</table>
### Usage and Purpose

**Structure**
- Many data elements in one variable

(Not Yaskawa) Possible Definition of the AXIS_REF derived data type

<table>
<thead>
<tr>
<th>AXIS_REF</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td></td>
</tr>
<tr>
<td>Axis Number</td>
<td>UINT</td>
</tr>
<tr>
<td>Max Speed</td>
<td>LREAL</td>
</tr>
<tr>
<td>Encoder Resolution</td>
<td>DINT</td>
</tr>
</tbody>
</table>

**LeftMotor**

<table>
<thead>
<tr>
<th>Data</th>
<th>Data Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UINT</td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>LREAL</td>
<td></td>
</tr>
<tr>
<td>8192</td>
<td>DINT</td>
<td></td>
</tr>
</tbody>
</table>

Variable named LeftMotor of data type AXIS_REF
AXIS_REF is defined under Data Types in DataTypes_Toolbox

One Element In the Structure

- More elements may be added by Yaskawa in the future

What is the data type of the AxisNum element?

<table>
<thead>
<tr>
<th>AXIS_REF</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Data Type</td>
</tr>
<tr>
<td>AxisNum</td>
<td>UINT</td>
</tr>
</tbody>
</table>
Axis_Name & Number

- **Logical Axis Number**
  - Hardware Configuration
  - Axis Parameter #1831
  - NOT network node number

- **Axis Name**
  - Can be changed
  - Used to create axis ref variable

Axis variable is automatically created by Hardware Configuration.
**View – Initialize Multi-Element Variable Window**

- Enter the logical axis number

Type the initial value of the structure element in this window
- **MotionWorks IEC 2**
  - Manually create axis variables (with Axis_Ref datatype)
  - Initialize axis variables in ST program
  - Refer to Quick Start Videos

This text command loads the unsigned integer “1” into the AxisNum element of variable LeftMotor.
Axis_Ref

- **Confirm Operation**
  - Add each axis to the Watch Window
  - Set initial value
  - Warm Start vs Cold Start

---

MP2600iec: Rotary axis is also a virtual axis, AxisNum = 27
Servo Enable

- **HMI_I**
  - Part of class project
Servo Enable

Enable Program

- Create Enable POU
  - POU Type: Program
  - Language: LD
  - Run in the Slow task
- Add MC_Power
  - Axis
  - Enable
• **Shortcut Button**
  - Extras – Options - Commands
- **Use Debug Mode**
  - Program and test Screw
  - Program and test RightMotor

- **MC_Power.Status**
  - Status of the command
  - Updates at the application scan rate
Servo Enable

- **ErrorID Output**
  - Right-click for help

### Troubleshooting

- **ErrorID Output**
  - **4414**: MECHATROLINK Communications to the device failed. Troubleshoot the connection.
  - **4625**: Axis ID does not correspond to an axis number in the configuration. Tip: Make sure the relevant POU's match.
  - **4641**: Buffer mode does not correspond to a valid axis ID. Check the configuration.
  - **4893**: The specified external axis may not be accessible. Review the axis configuration.

See HELP
Servo Enable

- **Confirm**
  - Screw
  - Rotary
  - Virtual
Positioning

- MC_MoveRelative
- MC_MoveAbsolute
- Timers
- Move Sequence
Create Positioning (program POU)

- What task is most appropriate? (Fast, Med, Slow)

Refer to Quick Start Video
Positioning

Positioning (program POU)

- **MC_MoveRelative Function Block**

## Screw Move Profile

<table>
<thead>
<tr>
<th>Input</th>
<th>Initial Value</th>
<th>Unit</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>-270.0</td>
<td>mm</td>
<td>Use Variable</td>
</tr>
<tr>
<td>Velocity</td>
<td>180.0</td>
<td>mm/sec</td>
<td>Use Variable</td>
</tr>
<tr>
<td>Accel</td>
<td>360000.0</td>
<td>mm/sec²</td>
<td>Use Literal LREAL#360_000.0</td>
</tr>
<tr>
<td>Decel</td>
<td>360000.0</td>
<td>mm/sec²</td>
<td>Use Literal LREAL#360_000.0</td>
</tr>
</tbody>
</table>

Warm Start required if initial value is changed.
• Quick Zero Set (optional)
  - Repeat relative moves until at mechanical zero
  - Use MC_SetPosition
  - Repeat for Rotary

Change distance by trial and error. Re execute until arrow on motor wheel is pointing up
Positioning

- **Add MC_MoveAbsolute**
  - Create a move sequence

<table>
<thead>
<tr>
<th>Input</th>
<th>Initial Value</th>
<th>Unit</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>0.0</td>
<td>mm</td>
<td>Use Variable</td>
</tr>
<tr>
<td>Velocity</td>
<td>180.0</td>
<td>mm/sec</td>
<td>Use Variable</td>
</tr>
<tr>
<td>Accel</td>
<td>360000.0</td>
<td>mm/sec²</td>
<td>Use Literal LREAL#360_000.0</td>
</tr>
<tr>
<td>Decel</td>
<td>360000.0</td>
<td>mm/sec²</td>
<td>Use Literal LREAL#360_000.0</td>
</tr>
</tbody>
</table>

Relative

Absolute
Partial Solution

- Jerk, Direction, BufferMode can be disconnected and default values are used.
  - Jerk exists as a parameter in HWConfig “Moving Average Filter” #1300, #1301
  - BufferMode and Direction are “Enumerated Data Types” (more information later)
• **Repeating Sequence**

Timers count how long the IN input is true. Use a N.C. contact to repeat.
Adjust the program to operate as follows:

- Screw moves, wait 500ms
- Rotary moves, wait 500ms
- Screw returns, wait 500ms
- Rotary returns, wait 500ms
- Repeat sequence
• **Solution Concept**
Stop & Alarm

- MC_Stop
- MC_Reset
- MC_ReadAxisError
- Alarm Code Diagnosis
- Task Execution Adjustment
• Create Program POU “Stop”
  - Instance: SlowTsk
Stop & Alarm

• Partial Solution

“HMI_I” Program POU

“Stop” Program POU

MC_Stop
• Create Program POU “Reset”
  ■ Instance: SlowTsk

Stop & Alarm
Stop & Alarm

**Partial Solution**

“HMI_I” Program POU

- **MC_Reset**
  - G_ResetScrew
  - G_ResetRotary
  - G_ResetController

- **MC_ReadAxisError**
  - MC_Reset_1
    - Screw
    - Axis
    - Execute: Done, Busy, Error
    - ErrorID: 16#0000
  - MC_Reset_2
    - Rotary
    - Axis
    - Execute: Done, Busy, Error
    - ErrorID: 16#0000
  - MC_ReadAxisError_1
    - Valid
    - Busy
    - Error
    - ErrorID
    - ErrorClass: 16#0000
  - MC_ReadAxisError_2
    - Valid
    - Busy
    - Error
    - ErrorID
    - ErrorClass: 16#0000
  - M_ClearAlarms_1
    - G_ResetController
    - Execute: Done, Busy, Error
    - ErrorID: 16#0000
  - M_ClearAlarm_1
    - G_ResetController
    - Execute: Done, Busy, Error
    - ErrorID: 16#0000

Stop & Alarm
Stop & Alarm

Yaskawa Alarm Code

- **Produce Alarm**
  
  Speed = 40.000.0
  Distance = 40.000.0

- **Axis Error ID (Hex)**
- **Error Class (Hex)**

Open any variable in DEBUG mode
Stop & Alarm

---

**Alarm**
- Motion cannot continue under current conditions
- Disable Servo
  - Alarm Stop Mode
- Display Code A. □ □ □
- Examples
  - A.400 Overvoltage
  - A.510 Overspeed
  - A.710 Overload: High Load
  - A.860 Encoder Overheat

---

**Warning**
- Future alarm under current conditions
- Servo remains enabled
- Display Code A. 9 □ □
- Examples
  - A.900 Position Error Overflow
  - A.910 Overload
  - A.95A Command Warning
  - A.971 Undervoltage

---

9.1 Alarm Displays ................................................................. 9-2
  9.1.1 List of Alarms .......................................................... 9-2
  9.1.2 Troubleshooting of Alarms ........................................... 9-6

9.2 Warning Displays ............................................................. 9-21
  9.2.1 List of Warnings ...................................................... 9-21
  9.2.2 Troubleshooting of Warnings ........................................ 9-22
• Adjust POU order in task, top to bottom
  - Logical sequence
• I/O Module Task Assignment
  - Assign to same task as application code that uses the %I %Q
  - Use Hardware Configuration

When inputs are controlled by the machine (not human operation), then use FastTsk to stop.
Enumerated Data Types

Definition
Data Types Folder
MC_Direction
Enumerated Types as Literal and Variable
MC_Direction for Rotary Axis
MC_BufferMode to Create Blended Moves
### What is an enumerated data type?

- A **NAME** for a **NUMBER**
- **Code reads easily**
- **Reduced mistakes**

---

**Enumerated Data Types**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction</td>
<td>Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction.</td>
<td><img src="image" alt="Direction Table" /></td>
</tr>
<tr>
<td>BufferMode</td>
<td>Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.</td>
<td><img src="image" alt="BufferMode Table" /></td>
</tr>
</tbody>
</table>

---

**MC_MoveAbsolute**

- Axis
- Execute
- Position
- Velocity
- Acceleration
- Deceleration
- Jerk
- Direction
- BufferMode

**MC_MoveAbsolute 1**

- Done
- Busy
- Active
- CommandAborted
- Error
- ErrorID
Enumerated Data Types

DataTypes Toolbox

- “Data Types” folder
- “MotionBlock Types”
- Other Enumerated types exist

```plaintext
(*---------------------------*  Enumerated Types
MC_BufferMode: (Aborting, Buffered, BlendingLow, BlendingPr
MC_Direction: (positive_direction, shortest_way, negative_d
MC_Detection_Pattern: (Rising_Edge, Falling_Edge);
```
### MC_Direction

- **MC_Direction**
  - Absolute Positioning of Rotary Loads
  - Four possible “directions”
  - Example: Rotary Table
  - Position 0 = Position 360

**MC_Direction** does not apply to
- Relative Moves (MC_MoveRelative)
- Linear Loads

MC_MoveVelocity uses only positive_direction and negative_direction. Other values are ignored.

<table>
<thead>
<tr>
<th>MC_Direction#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>positive_direction In a rotary application, forces the axis to move in a positive direction.</td>
</tr>
<tr>
<td>1</td>
<td>shortest_way For use in applications where the Load Type is configured as a rotary or modularized axis.</td>
</tr>
<tr>
<td>2</td>
<td>negative-direction In a rotary application, forces the axis to move in a negative direction</td>
</tr>
<tr>
<td>3</td>
<td>current_direction For use in applications where the Load Type is configured as a rotary or modularized axis. Only applies if an existing move is in progress and another function block such as MC_MoveAbsolute or MC_MoveRelative is executed. Once the axis is at StandStill, using MC_Direction_CurrentDirection will default to the positive direction.</td>
</tr>
</tbody>
</table>
Enumerated Data Types

• **Programming with Enumerated Data Types**
  - As Literal Value
    - MC_BufferMode#Aborting
    - MC_Direction#Shortest_Way

**Compare:**
- UINT#1
- LREAL#1.0
- MC_Direction#Shortest_Way

**Format:**
- \(<\text{DataType}\>#<\text{data}>\)
Enumerated Data Types

- Correct Spelling of Enumerated Data Type
  - Function Block Help
  - Copy and paste
Enumerated Data Types

- **Online Hardware Configuration (Rotary)**
  - Rotary
  - Degrees
  - Online Save
  - Reboot
- **Application Program (Positioning)**
  - Connect a literal at direction input
- **Observe Result**

Use `MC_Direction` in Program

- Use `MC_Direction` in Program

- Connect a literal at direction input
Enumerated Data Types

- **Programming with Enumerated Data Types**
  - **Variable**
    - Set value in application code
  - **Not Supported:**
    - Data Type detection
    - Debug display
    - Initial value

As Variable

- **MC_MoveAbsolute**
  - Axis
  - Execute
  - Position
  - Velocity
  - Acceleration
  - Deceleration
  - Jerk
  - Direction
  - BufferMode

**Variable Properties**
- Name: MoveDirection
- Data Type: INT
- Usage: VAR

Enter variable name
MC_Direction data type not shown

Value loaded to variable within application code

Open existing variable
MC_Direction data type available
**MC_BufferMode**

- Move 2 waits for Move 1 to complete
- Create “blended moves”
- Use for registration applications

---

### Buffer mode

<table>
<thead>
<tr>
<th>Buffer mode</th>
<th>Short description</th>
<th>Important note: The meaning of each value may vary depending on the FB(s) involved. For this reason, please also refer to the individual parameter descriptions!</th>
<th>Input value at BufferMode *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborting</td>
<td>This is the Default mode. The FB aborts an ongoing motion and the command affects the axis immediately.</td>
<td></td>
<td>INT#0</td>
</tr>
<tr>
<td>Buffered</td>
<td>The FB affects the axis as soon as the previous movement is complete. The axis will stop between the movements.</td>
<td></td>
<td>INT#1</td>
</tr>
<tr>
<td>BlendingLow</td>
<td>The FB controls the axis after the previous FB has finished, but the axis will not stop between the movements. The velocity is blended with the lowest velocity of both commands.</td>
<td></td>
<td>INT#2</td>
</tr>
<tr>
<td>BlendingPrevious</td>
<td>The FB controls the axis after the previous FB has finished (equivalent to buffered), but the axis will not stop between the movements. Blending with the velocity of the previous move.</td>
<td></td>
<td>INT#3</td>
</tr>
<tr>
<td>BlendingNext</td>
<td>The FB controls the axis after the previous FB has finished, but the axis will not stop between the movements. Blending with velocity of this (next) function.</td>
<td></td>
<td>INT#4</td>
</tr>
<tr>
<td>BlendingHigh</td>
<td>The FB controls the axis after the previous FB has finished (equivalent to buffered), but the axis will not stop between the movements. Blending with highest velocity of the previous and this (next) function.</td>
<td></td>
<td>INT#5</td>
</tr>
</tbody>
</table>
• **Use Logic Analyzer**
  - Try different buffer modes

Enumerated Data Types

![Logic Analyzer Interface]

- **MC_ReadActualVelocity**
  - **Screw**
  - **Axis**
  - **Valid**
    - 1
  - **Busy**
  - 0
  - **Error**
  - 0
  - **ErrorID**
  - 0
- **ActualVelocity**
  - Velocity_Screw: 177.6695251

**Logic Analyzer Settings**

- **Sampling**
  - Pre-recording cycles: 50
  - Post-recording cycles: 300

**Trigger Conditions**

2. Operator: FE (falling edge)

**Data Collection**

- Synchronous with task: FastTask

Configuration: Resource - LADATA
Yaskawa “Toolbox” User Libraries

- Toolbox Concept
- PLCopen Toolbox
- Toolbox Installer
- Dependent Libraries
- Insert Additional Toolbox
• **Right Click Help**
  - When Toolbox is installed
  - Links to eLearning Videos and recorded webinars
Yaskawa “Toolbox” User Libraries

- **Insert Another Project**
  - Library = any project
  - *.mwt (or *.mwe)

- **Library Data Imported**
  - User FU & FB POUs
  - Program POUs
  - Data Types
    - **NOT** global variables!
    - **NOT** dependent libraries!

- **Organization**
  - Specific projects for library use
  - Revision number in project name
  - Prefix (ex: YTTS_)
Yaskawa “Toolbox” User Libraries

• **Yaskawa Tech Note: TN.MCD.08.130**

All Functions, Function Blocks, and DataTypes from the libraries are available for the application.
Yaskawa “Toolbox” User Libraries

- Yaskawa.com/iectb

Installer unzips all yaskawa “Toolbox” user libraries to the Libraries Folder and activates Right-Click Help.
Yaskawa “Toolbox” User Libraries

- Refer to Quick Reference Guide
  - Steps 1 & 2 completed by Toolbox installer

Refer to the Quick Reference Guide

### Use a Library

<table>
<thead>
<tr>
<th>Step Description</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Acquire a ZWE (Express) or ZVT (Pro) file</td>
<td>Use your own, or download from Yaskawa.com Product Page. Follow links to save the file. In Windows Explorer, copy the file to C:/Documents and Settings/All Users/Documents/MotionWorks IEC 61131-3 Libraries (For organization purposes).</td>
</tr>
<tr>
<td>2. Unzip the library project to the library directory</td>
<td>In MotionWorks IEC 61131-3 File-&gt;Open Project / Unzip Project click &quot;Yes&quot; to unzip to the Library directory (file was copied here in previous step) or click &quot;No&quot; if opening directly from CD or Download folder &quot;Skip All&quot; to Extracting Firmware Libraries dialog &quot;Yes to All&quot; to Overwrite Page Layout.</td>
</tr>
<tr>
<td>4. Start new / open existing project</td>
<td>File -&gt; New, or File -&gt; Open</td>
</tr>
<tr>
<td>5. Insert the Library and any dependent libraries</td>
<td>In Project Tree, &quot;Project&quot; tab, R-Click &quot;Libraries&quot; -&gt; Insert -&gt; User Libraries. Navigate to find the Library (if you unzipped it to the &quot;Libraries&quot; folder, you will see it right away. Also insert any dependent libraries noted in Step 3.</td>
</tr>
<tr>
<td>6. Delete duplicate project data types</td>
<td>In Project Tree, &quot;Project&quot; tab, expand &quot;Data Types&quot; folder for both the user library and the project library. Delete any duplicates of &quot;PLCTaskInfoTypes&quot; or &quot;MotionBlockTypes&quot; from the project library. R-Click -&gt; delete (or open, delete text) These data types are already defined within the imported library. Repeating the definition here causes compile errors since the same data types were defined two times, even though the definitions are identical.</td>
</tr>
<tr>
<td>7. Use FB from new group in edit wizard</td>
<td>Click on programming worksheet whitespace. Open Edit Wizard and the group dropdown list will have the library name. User Library blocks appear as Blue by default. Help for Yaskawa Application Code Toolbox user libraries is available on the website, but is not integrated with the Right-Click menu as it is for the pink colored Firmware Library function blocks.</td>
</tr>
</tbody>
</table>
Yaskawa “Toolbox” User Libraries

- **Open CamToolbox Library Project**

- **Dependent Libraries**
  - PLCopen Plus
  - Ymotion
  - DataTypes_Toolbox
  - Math_Toolbox

- **Note the Libraries used by the Toolbox**
  - Note the order top to bottom – increasing complexity and dependence

Dependent libraries in your project must appear in the same order, above the PLCopen Toolbox library.
Yaskawa “Toolbox” User Libraries

- **Add Cam Toolbox and dependent libraries to your project**
  - Must appear in order of dependency from top to bottom

![Project Tree Window]

- R-click insert on Libraries Folder: Library inserted at the bottom
- R-click insert on existing Library: Library inserted above
- Click and drag to re-order (NEW in Version 3)
Yaskawa “Toolbox” User Libraries

- Run new installer
  - Yaskawa.com/iecTB
- Insert new versions
  - in same order
- Remove old versions
- Make

- Alternate
  - Newest version may be available individually (not part of installer)
  - Download ZWT, extract and insert
    » See Quick Reference Guide
  - Help will be disabled for that library
    » Manual process to move help file to new directory

Before Starting Project
– Please update to the most recent Toolbox user libraries

During Project Development
– You may wish to update certain Toolbox user libraries in order to use new features

After Project Development
– Toolbox update is not recommended
PLCopen Toolbox
Programming Introduction

Class Project #2
AxisStruct datatype
AxisControl function block
Jog function block
ReadAxisParameters function block
• Program Map for Second Project
  ■ Using PLCopen Toolbox
Back Up Existing Project

Use the Class Project Template to create a new project

- File-Unzip PLCopen*.zwt to new project name
- Adjust IP address
- Same Hardware Configuration – no update required
- Open original project in another instance of MotionWorks IEC
- Copy/Paste Logical POUs
  - Positioning
  - Stop
- Insert Program Instances
  - Positioning
  - Stop
Global Variables: Change AxisRef to “AxisStruct” data type

AxisStruct comes from PLCopen Toolbox user library
• Update Axis Var_In_Out of copied code
- Create the Control POU
  - Run in SlowTsk
PLCopen Toolbox Programming Introduction

- **Use AxisControl for Screw, Rotary, Virtual**

Notice the ".Ref"

The AxisStruct datatype already contains elements for warnings and alarms
Create the Manual POU

- Run in SlowTsk
• Implement JOG function block from PLCopen Toolbox
  ▪ New Program “Manual”
• Jog Screw
• Jog Virtual Axis
• Initialize AxisStruct elements
PLCopen Toolbox Programming Introduction

- **Create Monitor POU**
  - Run in FastTsk
• Implement ReadAxisParameters
  - AxisType is an enumerated type (Right-click Help)
PLCopen Toolbox Programming Introduction

- **ReadAxisParameters** requires **Y_Motion Firmware Library**

- **Insert Y_Motion Firmware Library**
- **View Axis Parameters in Watch Window**
  - “Parameter” = “feedback data” in PLCopen
PLCopen Toolbox Programming Introduction

- **ProductBuffer**
- **MoveRelativeByTime**

**PLCopen Toolbox User Library for MotionWorks IEC**

- Tutorial Videos Playlist on YouTube Channel
  - [https://www.youtube.com/playlist?list=PLNAENlyEDCkybLQ25iiwRZyG4NGBPb](https://www.youtube.com/playlist?list=PLNAENlyEDCkybLQ25iiwRZyG4NGBPb)
- Help contains video links
Driving Value

Homing

Introduction
PLCopen Homing
Supported Function Blocks
Homing State
PLCopen Toolbox Homing
Homing/ZeroPoint Program Investigation
Homing

What is Homing?

- A repeatable move sequence to move the axis from an unknown position to a known position
- Executed at every power-up (incremental encoder)
- Executed once when axis is commissioned (absolute encoder)
- Usually done at slow speed
- May involve proximity sensors, encoder reference pulse, hard stops, limit switches, torque limits
• **PLCopen Part 5: Homing**
  - Refer to PDF of PLCopen part 5
  - PLCopen defines
    - Homing “Procedures”
    - Homing “Steps”

• **Homing Steps**
  - There is no one block that would satisfy all homing requirements
  - PLCopen defines the building blocks, or “Steps” of homing
    - Homing function blocks are named MC_StepXxxxx

---

Two Homing Steps are supported in the MP2000iec controllers

- MC_Reset
- MC_SetPosition
- MC_StepLimitSwitch
- MC_StepRefPulse
- MC_Grip
- MC_TorqueControl
- MC_TouchProbe
Homing

- **MC_StepRefPulse**
  - PLCopen p.16-17
  - C-pulse (Index, Reference)
- **MC_StepLimitSwitch**
  - PLCopen p.11-12
  - P-OT, N-OT
- **MC_FinishHoming**
  - PLCopen p.15
- **MC_SetPosition**
  - Current position = any value

Supported Function Blocks

- **MC_StepRefPulse**
  - Axes
  - Execute: Done
  - Direction: Busy
  - Velocity: Active
  - SetPosition: Command/Aborted
  - TorqueLimit: Error
  - TimeLimit: ErrorID
  - DistanceLimit
  - BufferMode

- **MC_StepLimitSwitch**
  - Axes
  - Execute: Done
  - Direction: Busy
  - LimitswitchMode: Active
  - Velocity: Command/Aborted
  - TorqueLimit: Error
  - TimeLimit: ErrorID
  - DistanceLimit
  - BufferMode

- **MC_SetPosition**
  - Axes
  - Execute: Done
  - Position: Busy
  - Mode: Error
  - ErrorID
  - N-OT
• **Homing is a State of PLCopen**
• **Monitor: MC_ReadAxisStatus**
• **Add Home Program POU**
  - Run in MedTsk
Homing

- **Home_Pulse for Rotary**
  - “HomeStruct” data type for Home Data

Use the watch window to find good values for the data. Then initialize the structure elements. See the Toolbox Help manual.
**Simple Zero Set Example (For Screw)**

- Arbitrarily set position to zero (visual calibration)
- A one-time “zero set” for absolute encoders
Driving Value

Electronic Gear

Overview
PLCopen Gearing
Gear program POU
Program Example
Program Test
**Electronic Gearing**

- Motor moves like the output gear – “slave”
- Input gear is another encoder – “master”
  - External Encoder
  - Servo Axis
  - Virtual Axis
- Gear Ratio
  - Numerator = Slave Units
  - Denominator = Master Units

\[
\text{Gear Ratio} = \frac{\text{Slave Units (Output)}}{\text{Master Units (Input)}}
\]
Electronic Gear

- **GearIn**
  - Engages the slave to the master
  - If the master is already moving, slave accelerates to speed, then matches position

- **GearOut**
  - Disengages slave from master
  - Slave will continue at the previous speed, as if a frictionless system
• **Create Gear (LD) POU**
  - Run in MedTsk
Test the program

- Jog the master
- Execute MC_GearIn
  - While master is moving
  - While master is stopped
- Adjust Ratio
- Observe the InGear output
Use the logic analyzer to determine the following

1. What is the difference between executing MC_GearIn when the master is already moving vs when the master is stopped? Use Logic Analyzer (master speed, slave speed, InGear)

2. How can the gear ratio be changed without stopping the slave?

3. Under what conditions does the slave disengage and no longer follow the master?

4. Disable execution of MC_Stop. How does this affect operation? Does the slave remain engaged?

5. Change the master to the virtual axis. What are the advantages and disadvantages?
Electronic Gear

- **Master stopped, Gear In, Start Master**

- **Master Running, Gear In**