Driving Value

# PLCopen Basics with MPiec Controllers

Class No. TRM010-PLCopen-Basic
Doc No. eLV.Mpiec.01.PLCopenBasic
Rev A.00
Date: February 24, 2015

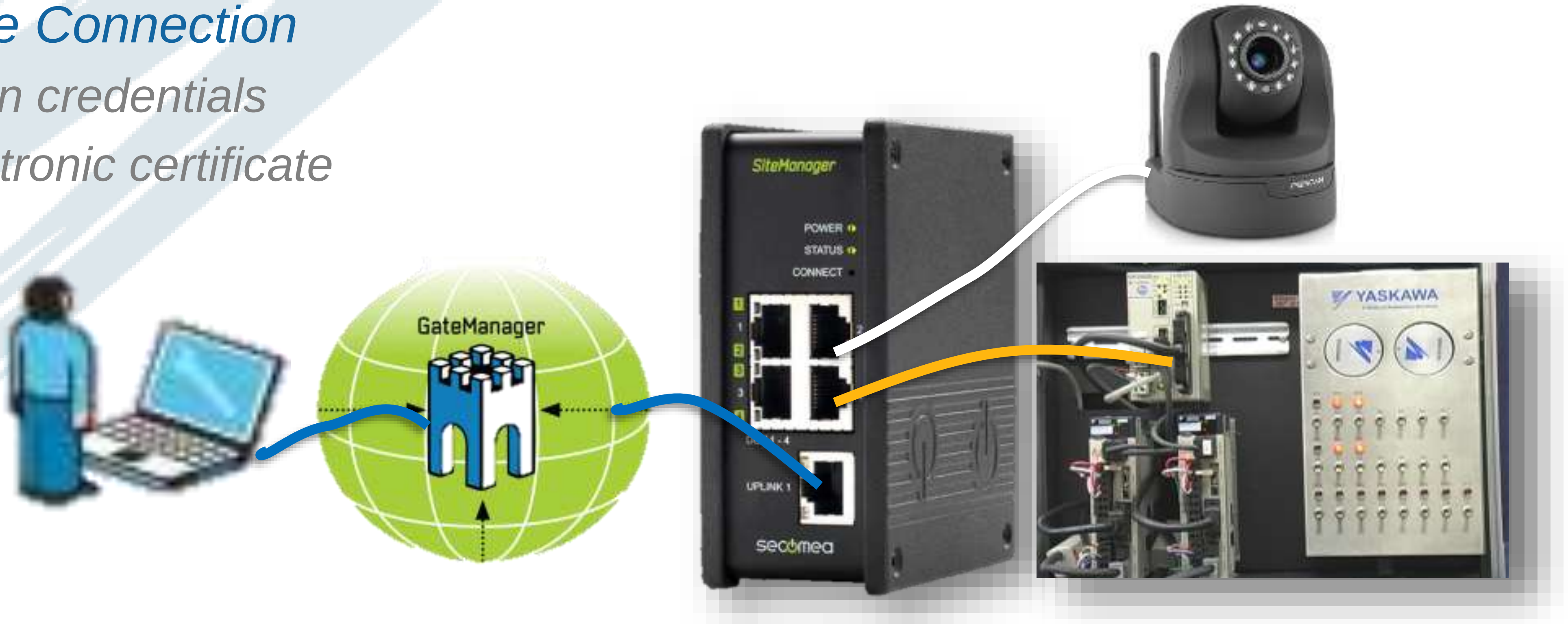
YASKAWA

# Remote Demo Connection

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

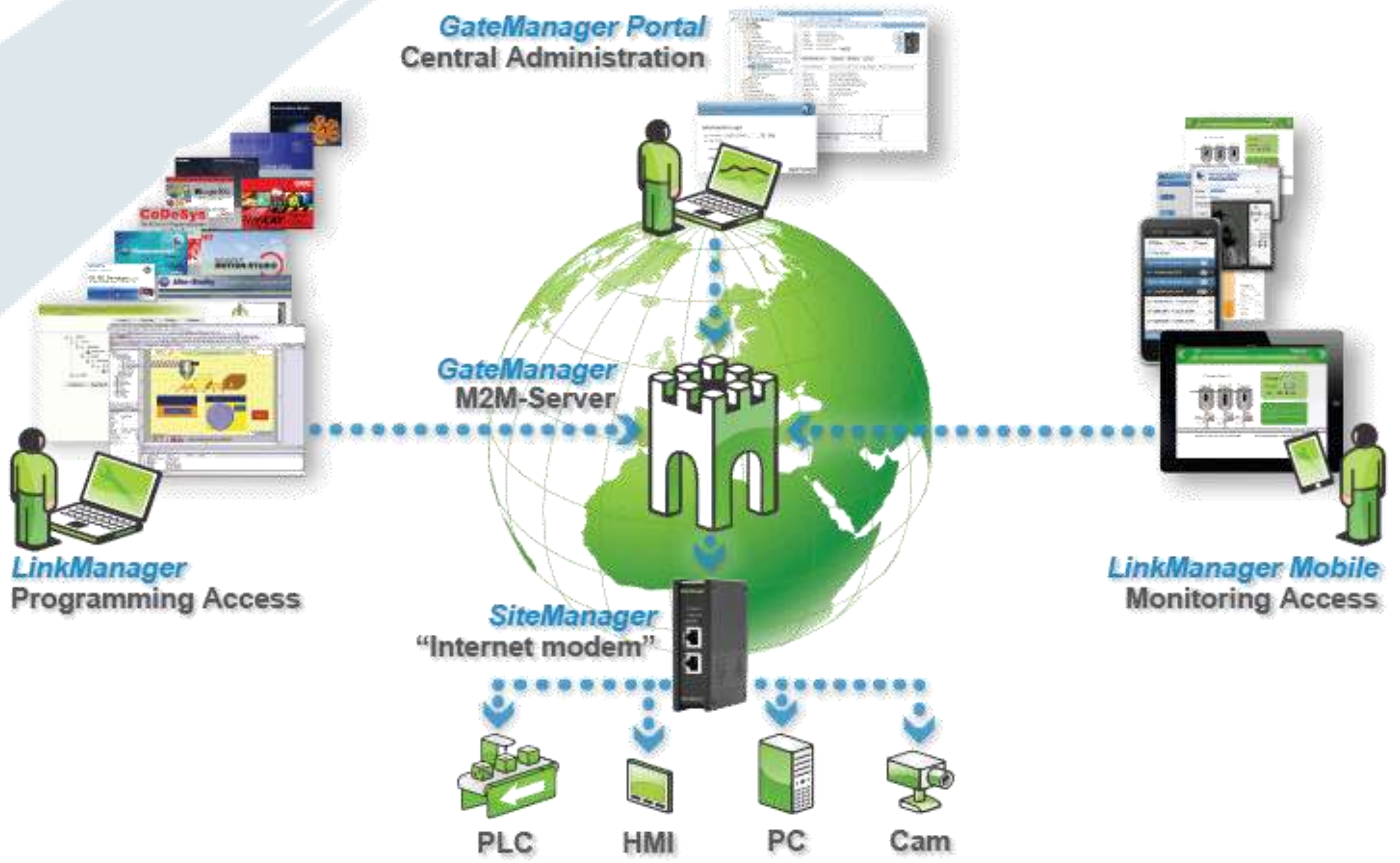
## Connection Concept

- *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*

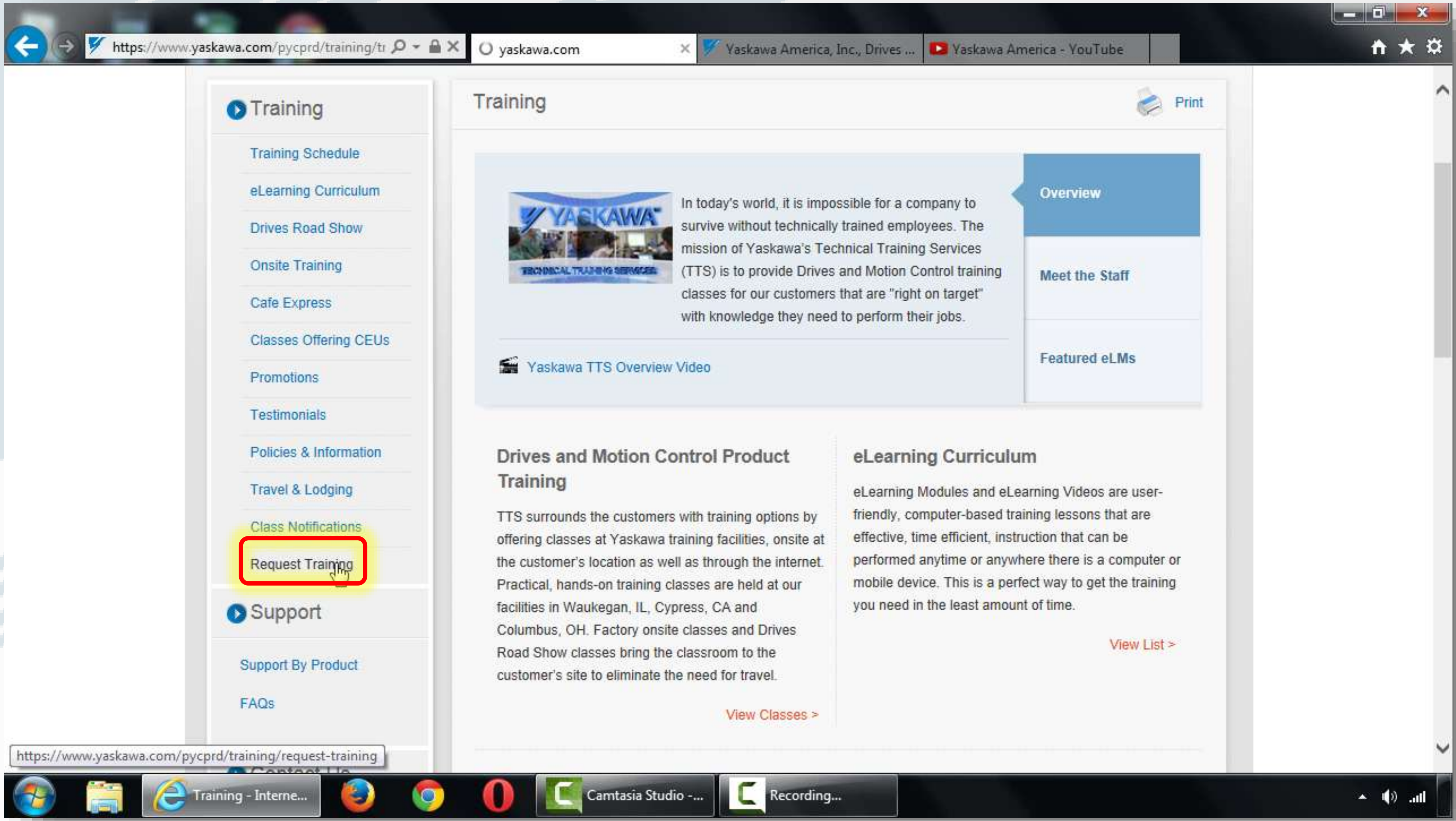


## Process Summary

- *Contact Yaskawa*
  - *training@Yaskawa.com*
  - *www.Yaskawa.com* *Request Training*
- *Receive email*
- *Install Link Manager software*
- *First-time Login*
  - *Start Link Manager*
  - *Certificate*
  - *Password*
- *Test Mpiec connection*
  - *Internet Explorer browser*
- *Test IP Camera connection*
  - *Port 88*
  - *Plugin installation*
    - » *Internet Explorer 11*
    - » *FireFox*
    - » *Chrome*
    - » *Opera*

## Remote Demo Connection

## Contact Yaskawa


https://www.yaskawa.com/pycprd/training/tr

yaskawa.com

Yaskawa America, Inc., Drives ...

Yaskawa America - YouTube

Training

- Training Schedule
- eLearning Curriculum
- Drives Road Show
- Onsite Training
- Cafe Express
- Classes Offering CEUs
- Promotions
- Testimonials
- Policies & Information
- Travel & Lodging
- Class Notifications
- Request Training

Support

- Support By Product
- FAQs

Training

Print

In today's world, it is impossible for a company to survive without technically trained employees. The mission of Yaskawa's Technical Training Services (TTS) is to provide Drives and Motion Control training classes for our customers that are "right on target" with knowledge they need to perform their jobs.

Yaskawa TTS Overview Video

Overview

Meet the Staff

Featured eLMs

Drives and Motion Control Product Training

TTS surrounds the customers with training options by offering classes at Yaskawa training facilities, onsite at the customer's location as well as through the internet. Practical, hands-on training classes are held at our facilities in Waukegan, IL, Cypress, CA and Columbus, OH. Factory onsite classes and Drives Road Show classes bring the classroom to the customer's site to eliminate the need for travel.

View Classes >

eLearning Curriculum

eLearning Modules and eLearning Videos are user-friendly, computer-based training lessons that are effective, time efficient, instruction that can be performed anytime or anywhere there is a computer or mobile device. This is a perfect way to get the training you need in the least amount of time.

View List >

https://www.yaskawa.com/pycprd/training/request-training

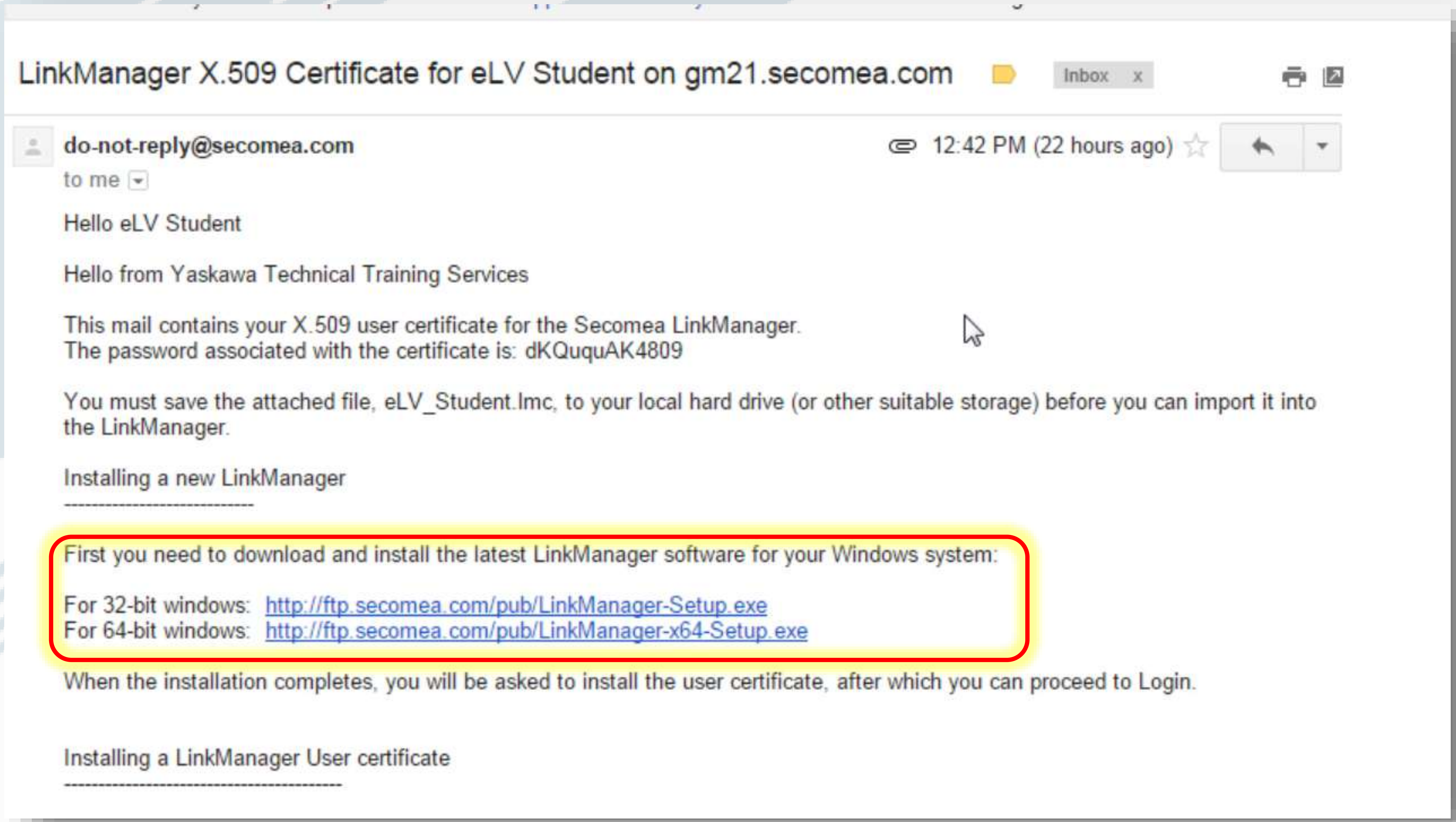
Training - Interne...

Camtasia Studio -...

Recording...

## Remote Demo Connection

## Receive eMail



LinkManager X.509 Certificate for eLV Student on gm21.secomea.com

do-not-reply@secomea.com

to me

12:42 PM (22 hours ago)

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
-----------------------------

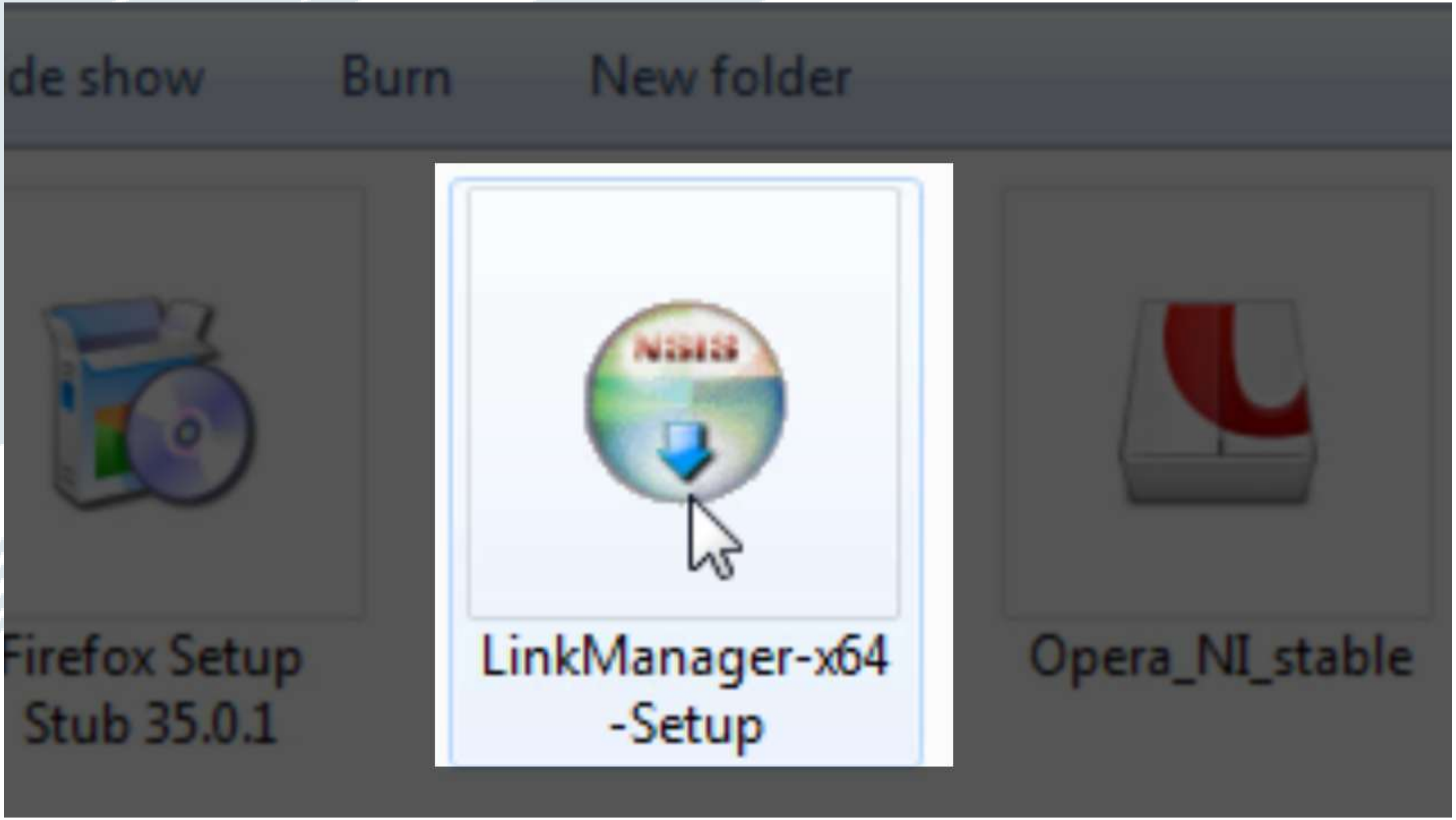
First you need to download and install the latest LinkManager software for your Windows system:

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.

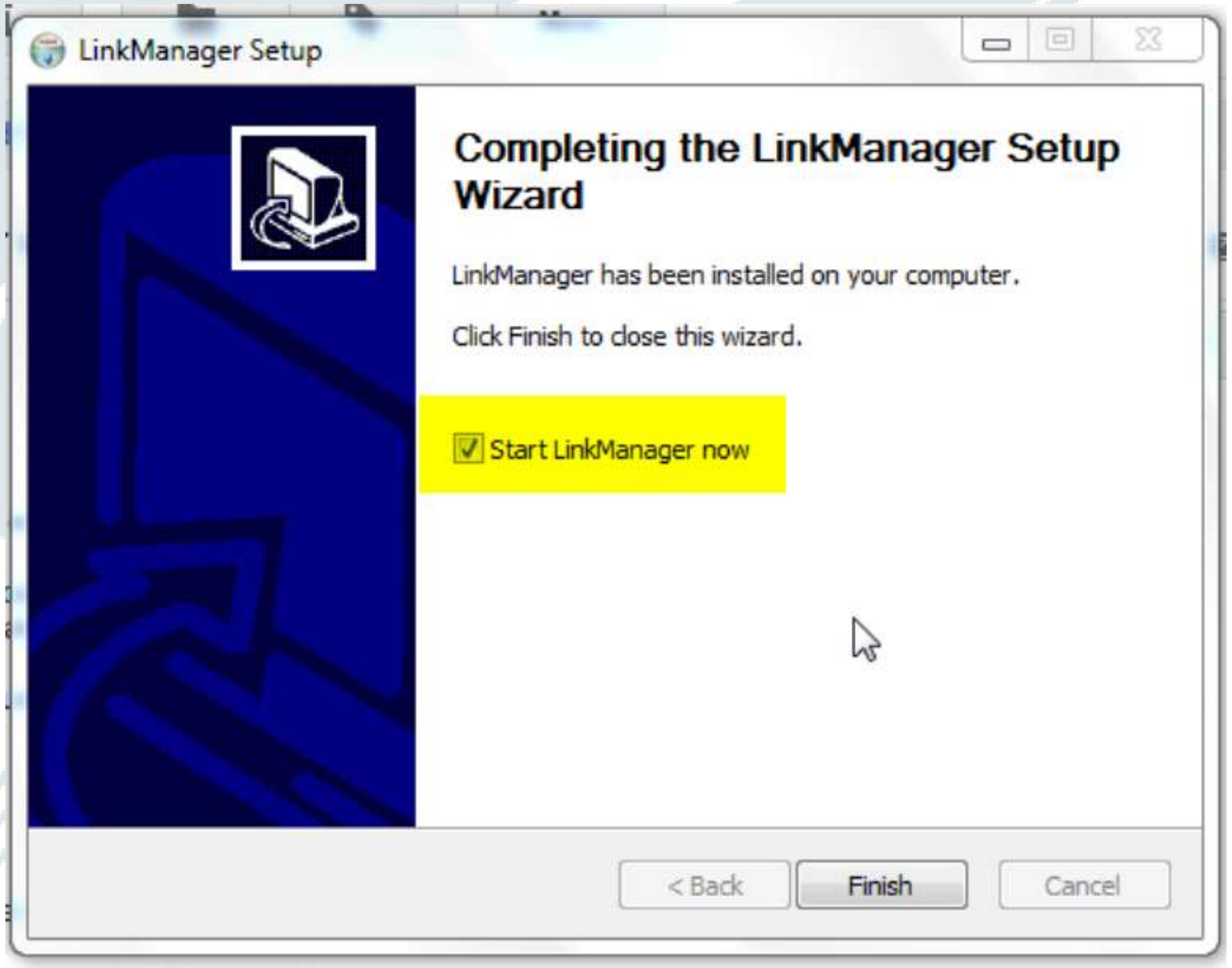
Installing a LinkManager User certificate
-------------------------------------------

## Install Link Manager Software



## First Time Login

- *Start Link Manager*



LinkManager Setup

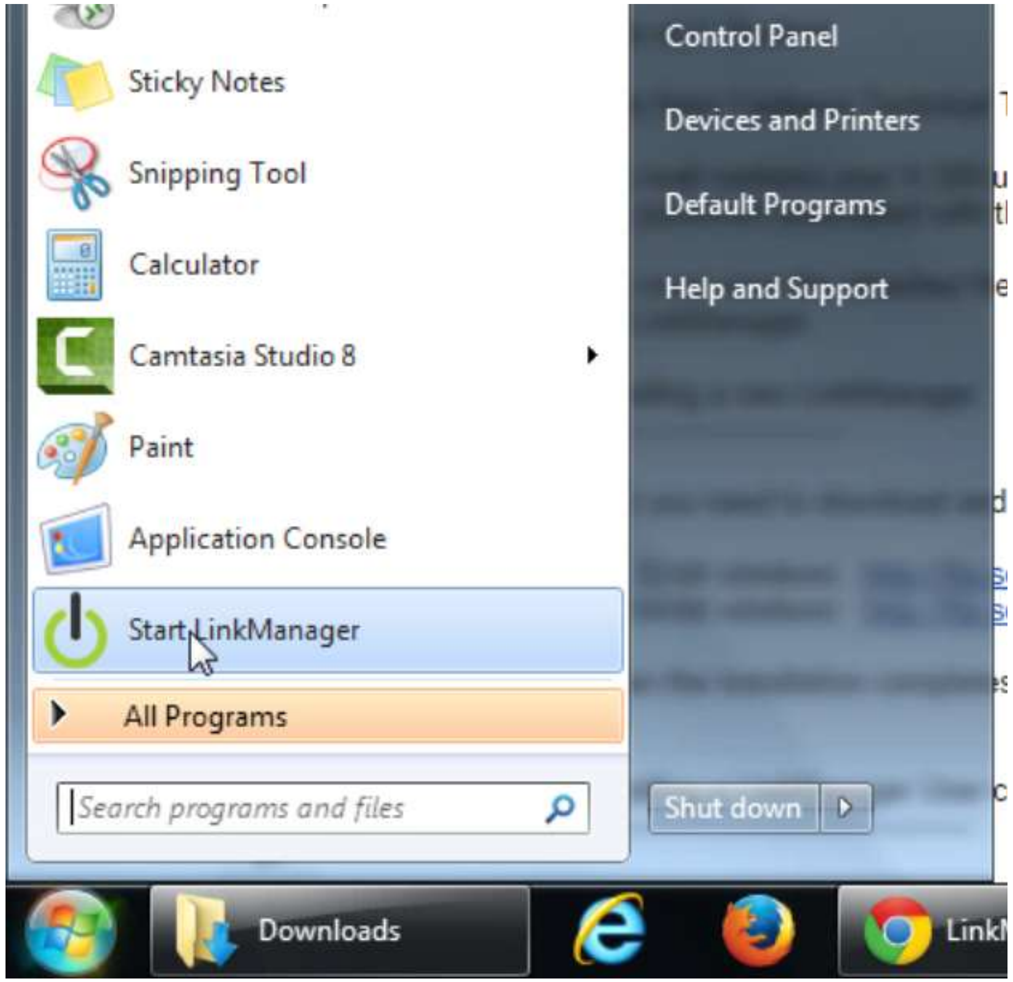
**Completing the LinkManager Setup Wizard**

LinkManager has been installed on your computer.

Click Finish to close this wizard.

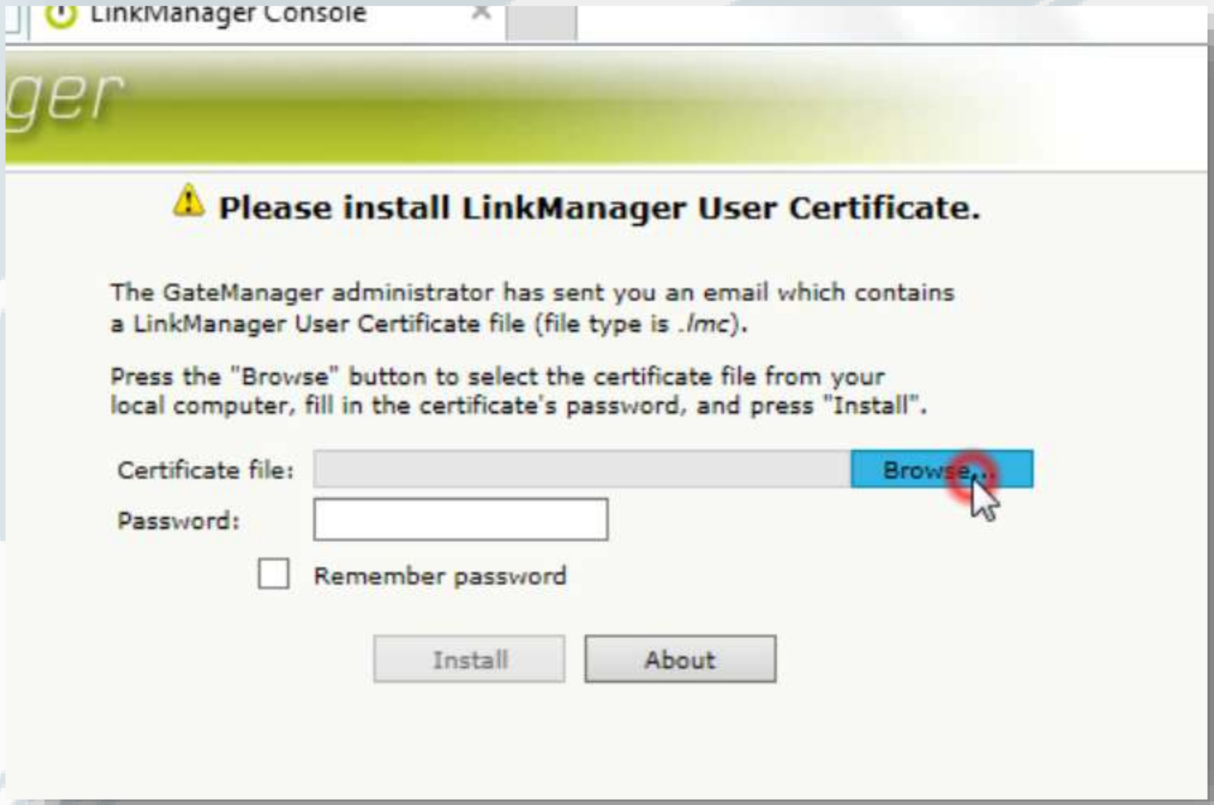
[✓] Start LinkManager now

< Back | Finish | Cancel



## First Time Login

- *Certificate*



LinkManager Console

ger

**Please install LinkManager User Certificate.**

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

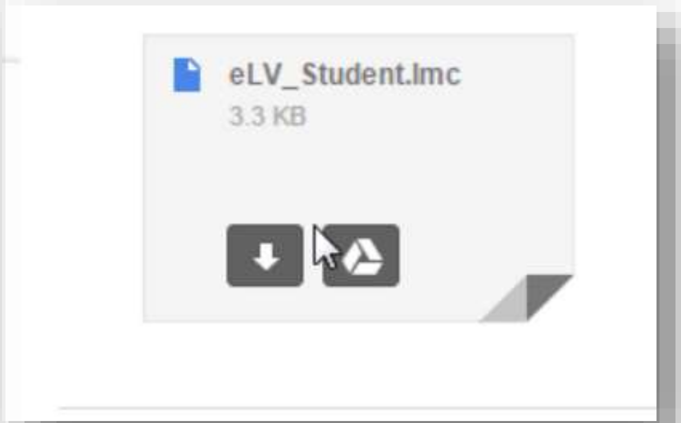
Press the "Browse" button to select the certificate file from your local computer, fill in the certificate's password, and press "Install".

Certificate file: Browse...

Password:

☐ Remember password

Install About



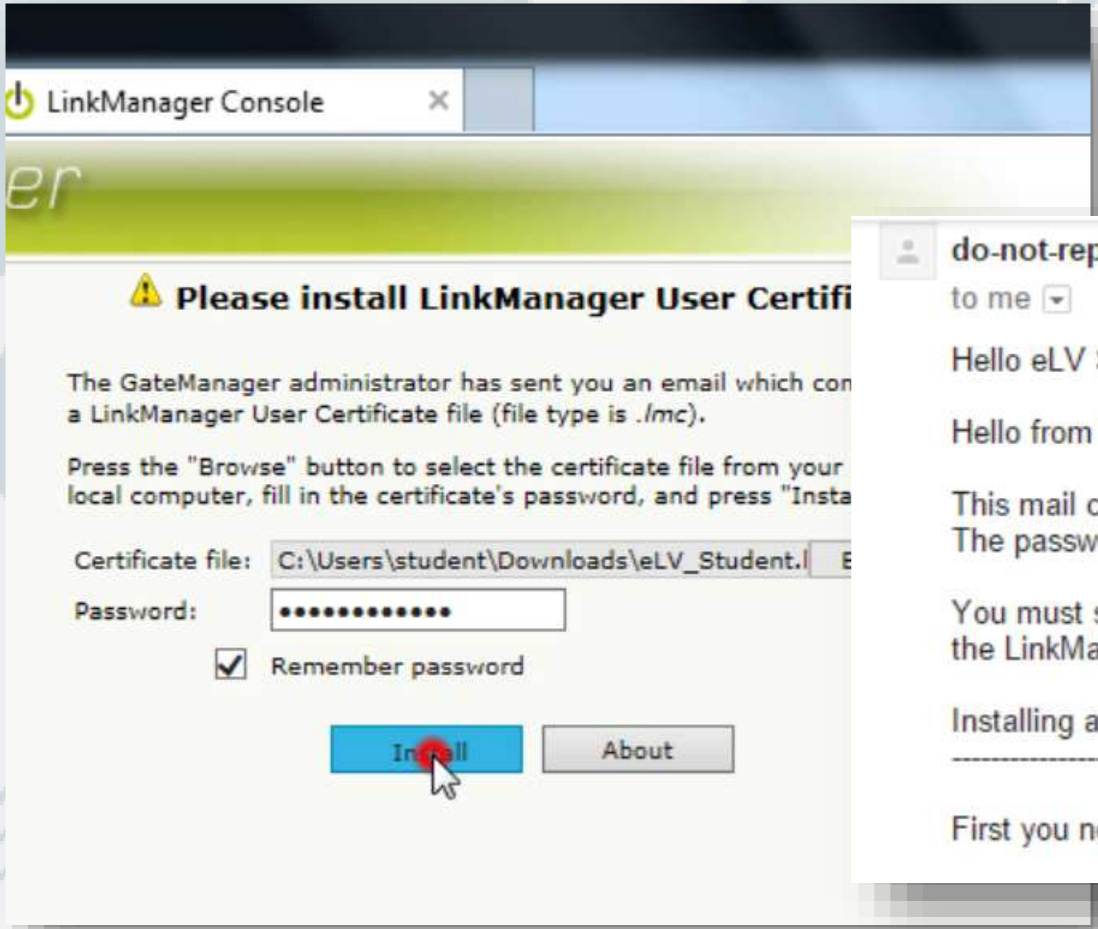
eLV_Student.lmc

3.3 KB

# Remote Demo Connection

## First Time Login

- *Password*



LinkManager Console

Please install LinkManager User Certifi

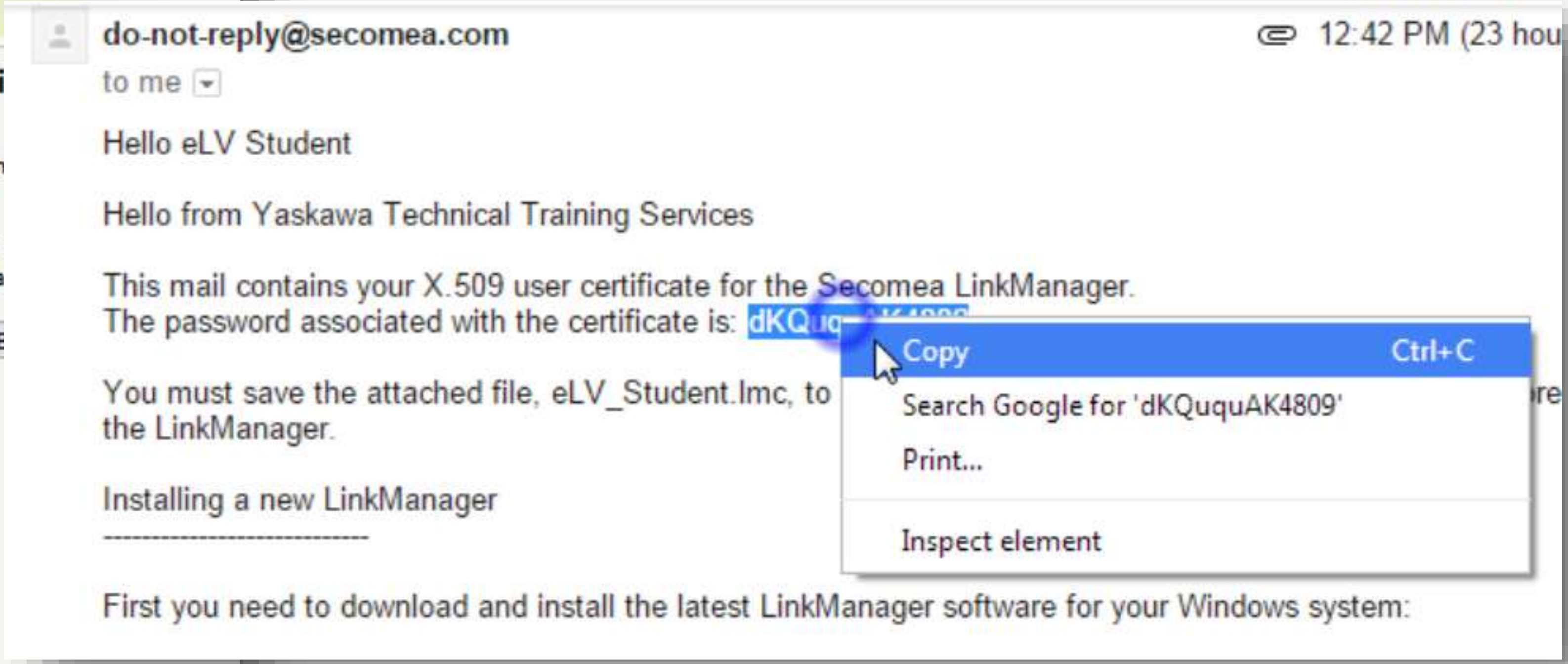
The GateManager administrator has sent you an email which con
a LinkManager User Certificate file (file type is *.lmc*).

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

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

Password: ••••••••••••

☑ Remember password



do-not-reply@secomea.com 12:42 PM (23 hou

to me

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: dKQu

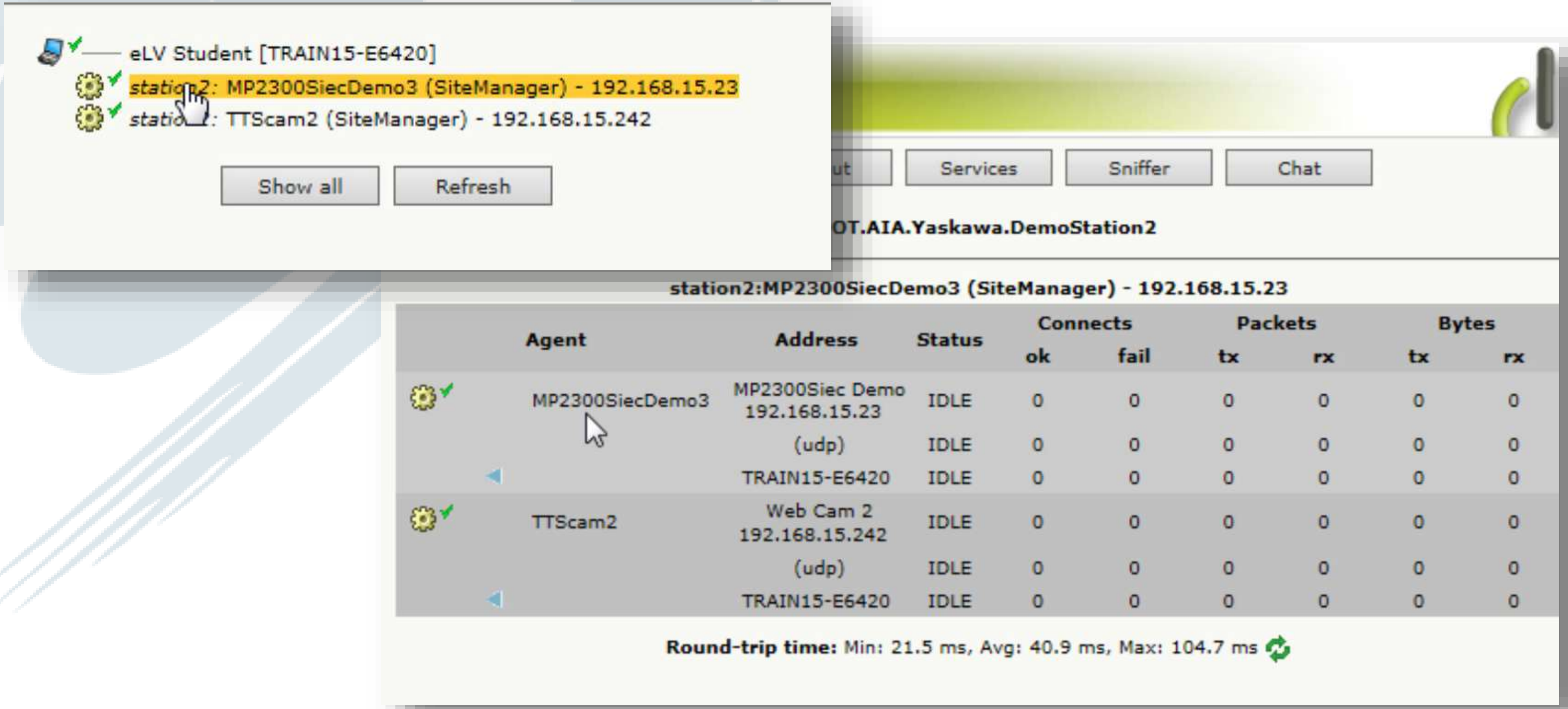
You must save the attached file, eLV_Student.lmc, to
the LinkManager.

Installing a new LinkManager

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

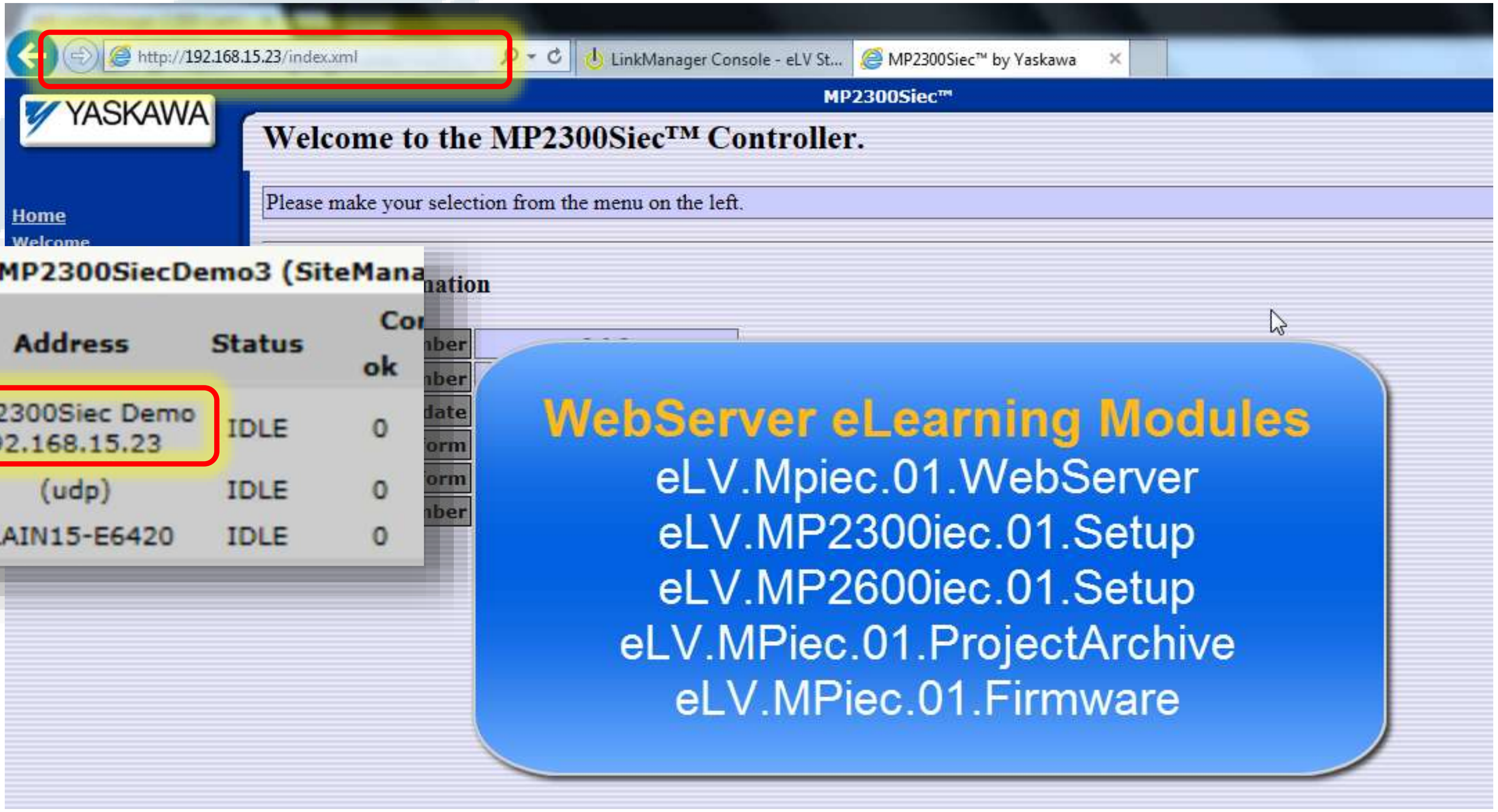
OT.AIA.Yaskawa.DemoStation2

station2:MP2300SiecDemo3 (SiteManager) - 192.168.15.23

| Agent | Address | Status | Connects | | Packets | | Bytes | |
|---|---|---|---|---|---|---|---|---|
| | | | ok | fail | tx | rx | tx | rx |
| MP2300SiecDemo3 | MP2300Siec Demo 192.168.15.23 | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| | (udp) | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| | TRAIN15-E6420 | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| TTScam2 | Web Cam 2 192.168.15.242 | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| | (udp) | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |
| | TRAIN15-E6420 | IDLE | 0 | 0 | 0 | 0 | 0 | 0 |

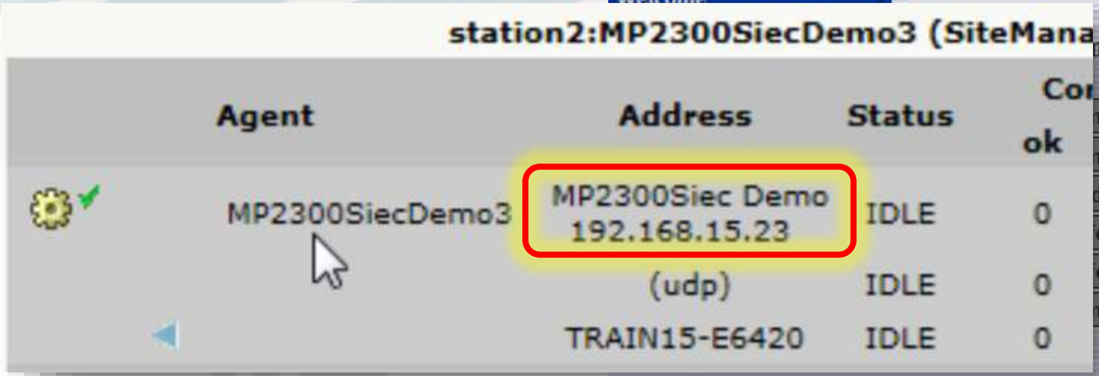
**Round-trip time:** Min: 21.5 ms, Avg: 40.9 ms, Max: 104.7 ms

## Test MPiec Connection


http://192.168.15.23/index.xml
LinkManager Console - eLV St...
MP2300Siec™ by Yaskawa
MP2300Siec™
YASKAWA
Home
Welcome

Welcome to the MP2300Siec™ Controller.

Please make your selection from the menu on the left.



station2:MP2300SiecDemo3 (SiteMana

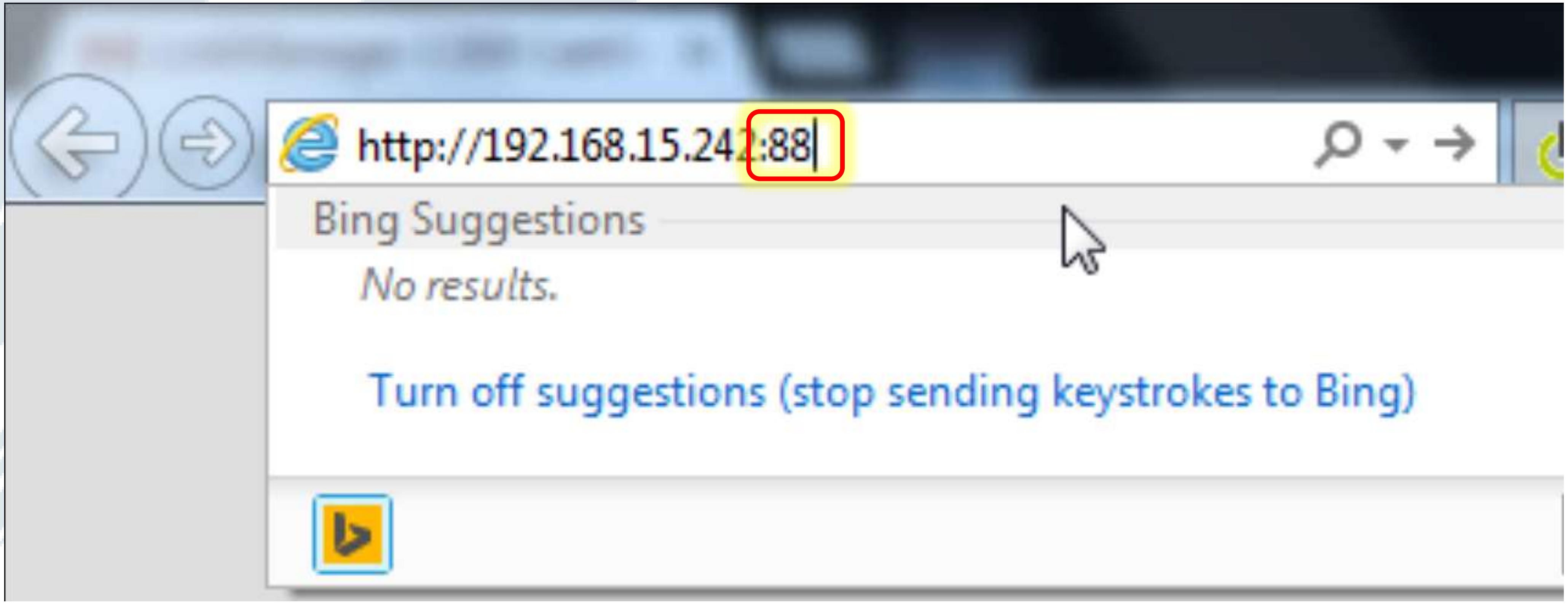
| | Agent | Address | Status | Co ok |
|---|---|---|---|---|
| | MP2300SiecDemo3 | MP2300Siec Demo 192.168.15.23 | IDLE | 0 |
| | | (udp) | IDLE | 0 |
| | | TRAIN15-E6420 | IDLE | 0 |

**WebServer eLearning Modules**

eLV.Mpiec.01.WebServer
eLV.MP2300iec.01.Setup
eLV.MP2600iec.01.Setup
eLV.MPiec.01.ProjectArchive
eLV.MPiec.01.Firmware

## Test IP Camera Connection

- *Port 88*


http://192.168.15.242:88

Bing Suggestions

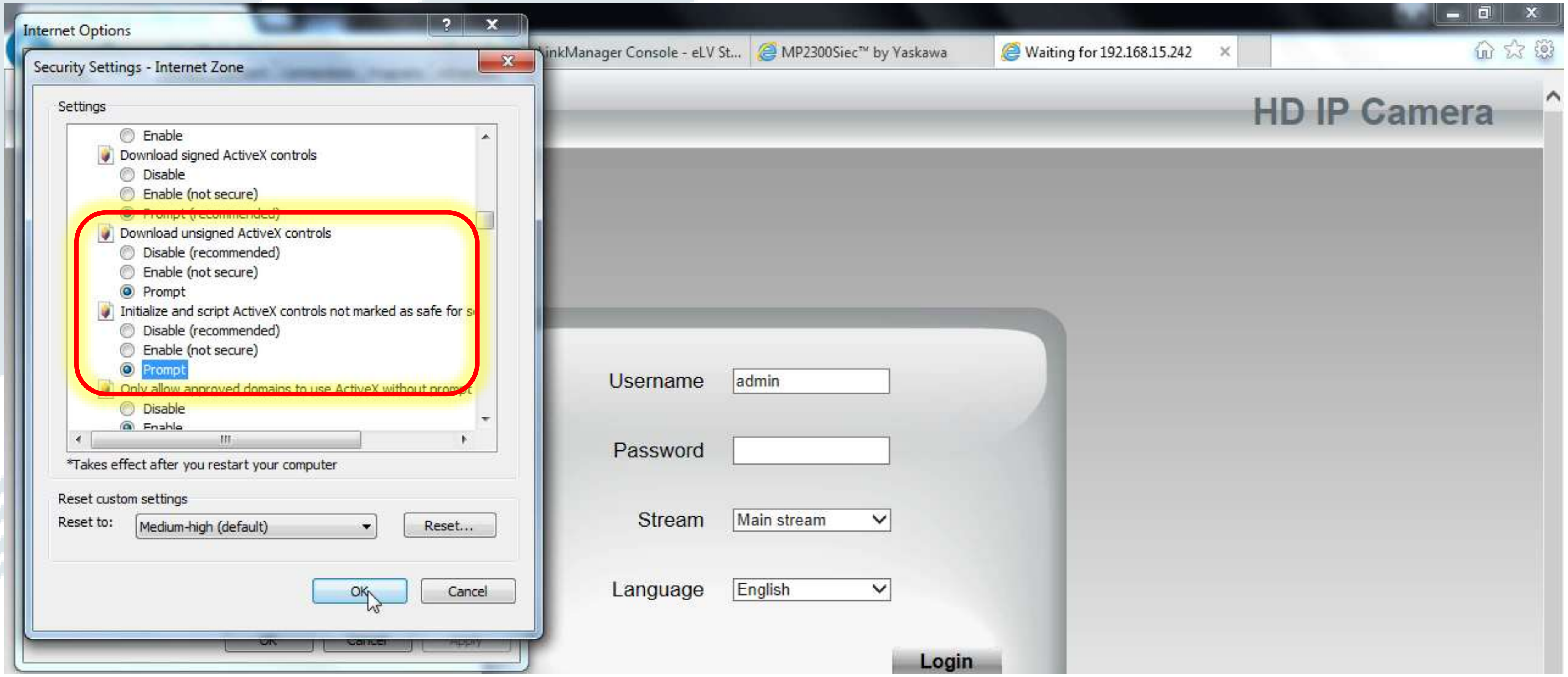
No results.

Turn off suggestions (stop sending keystrokes to Bing)

Remote Demo Connection

## Test IP Camera Connection

- *Plugin – ie 11*


Internet Options

Security Settings - Internet Zone

Settings

- Enable
- Download signed ActiveX controls
  - Disable
  - Enable (not secure)
  - Prompt (recommended)
- Download unsigned ActiveX controls
  - Disable (recommended)
  - Enable (not secure)
  - [✓ Prompt]
- Initialize and script ActiveX controls not marked as safe for s
  - Disable (recommended)
  - Enable (not secure)
  - [✓ Prompt]
- Only allow approved domains to use ActiveX without prompt
  - Disable
  - [✓ Enable]

*Takes effect after you restart your computer

Reset custom settings

Reset to: Medium-high (default)

Username: admin

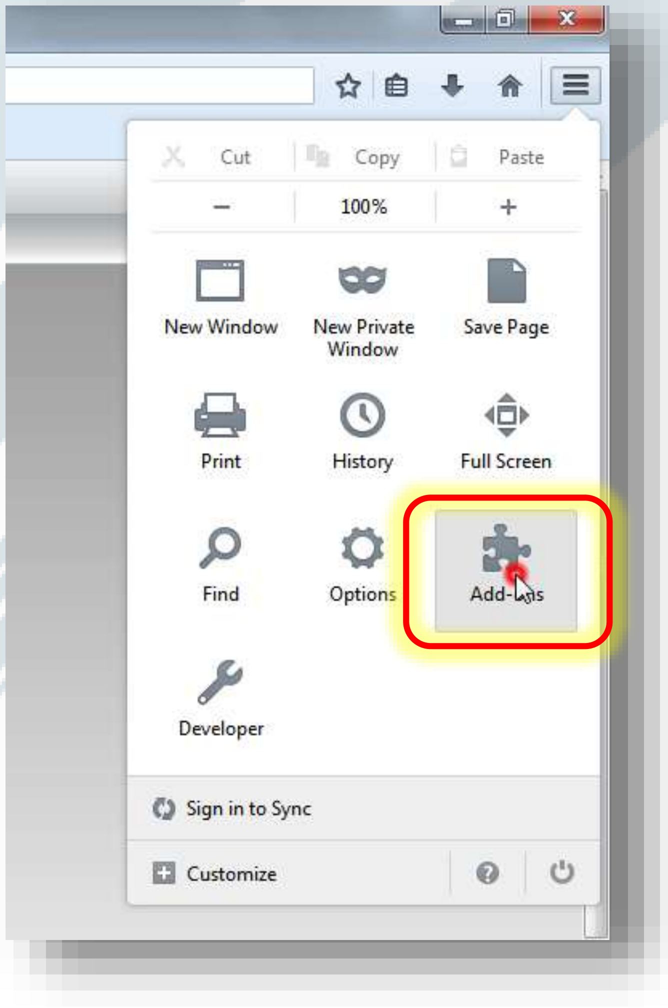
Password:

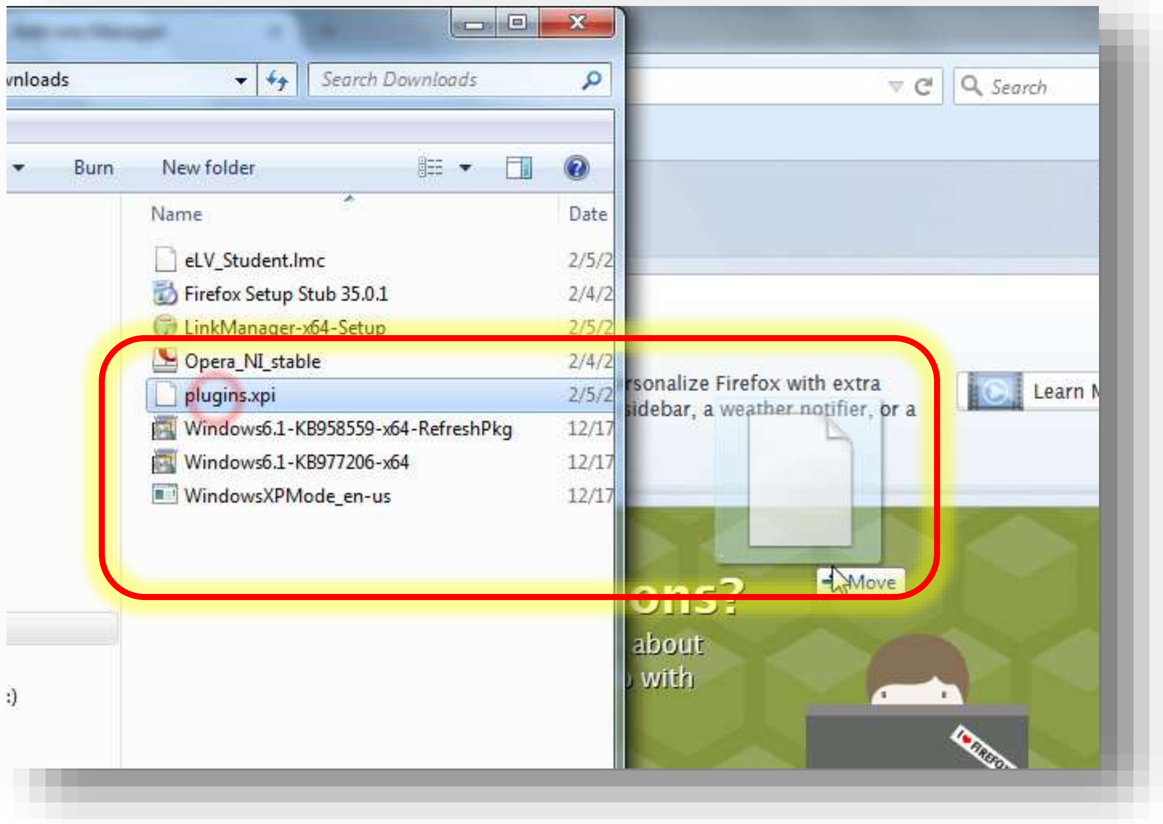
Stream: Main stream

Language: English

## Test IP Camera Connection

- *Plugin – Firefox*

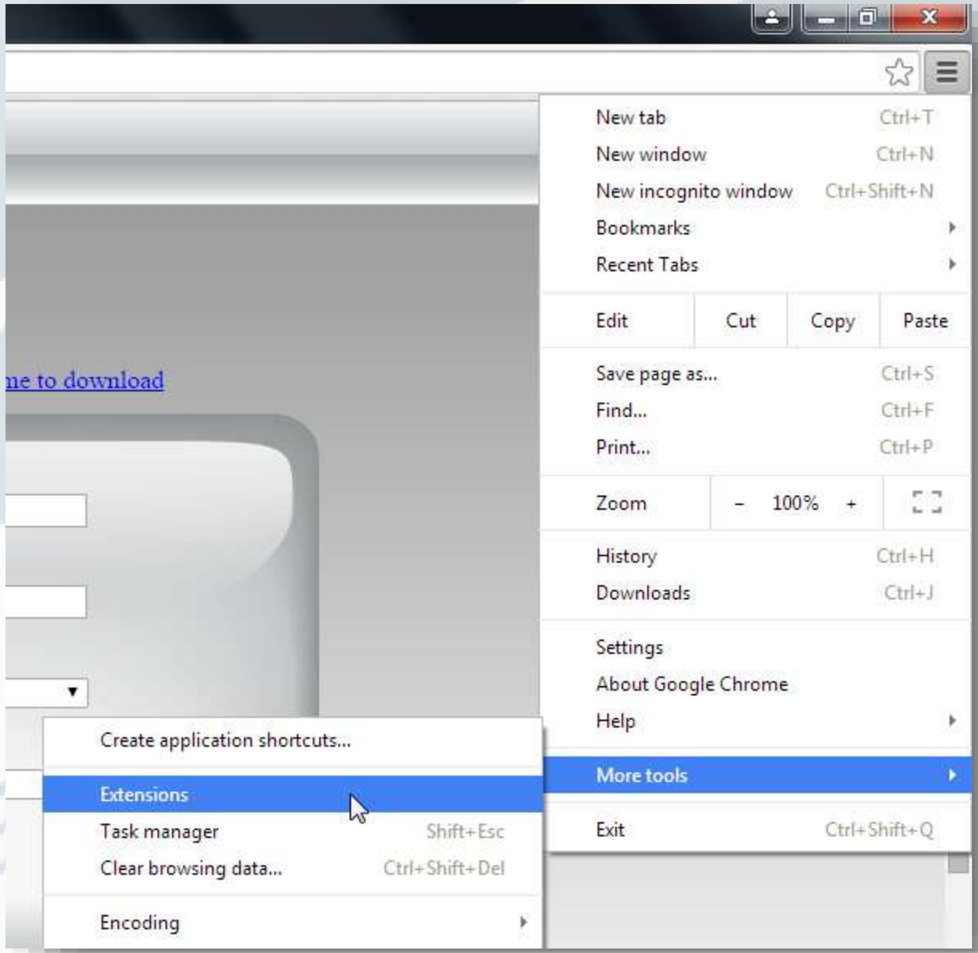


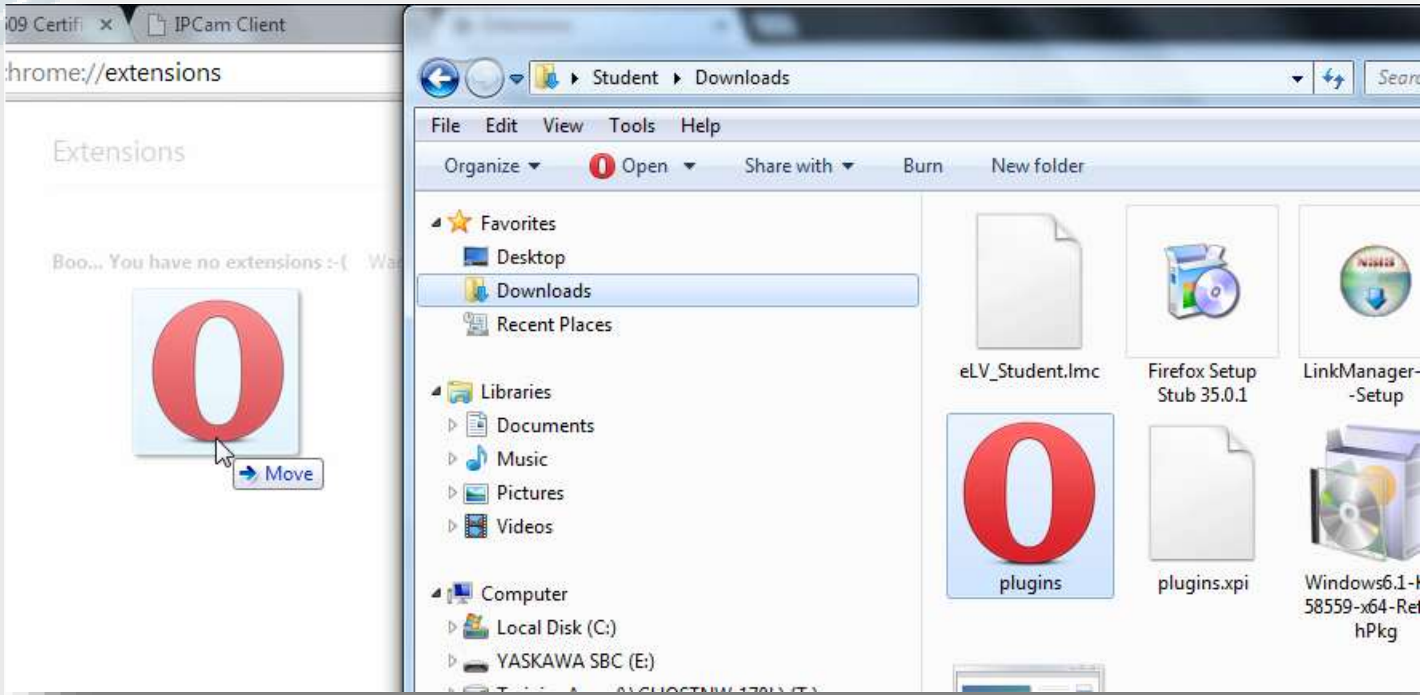


eLV_Student.lmc
Firefox Setup Stub 35.0.1
LinkManager-x64-Setup
Opera_NI_stable
plugins.xpi
Windows6.1-KB958559-x64-RefreshPkg
Windows6.1-KB977206-x64
WindowsXPMode_en-us

## Test IP Camera Connection

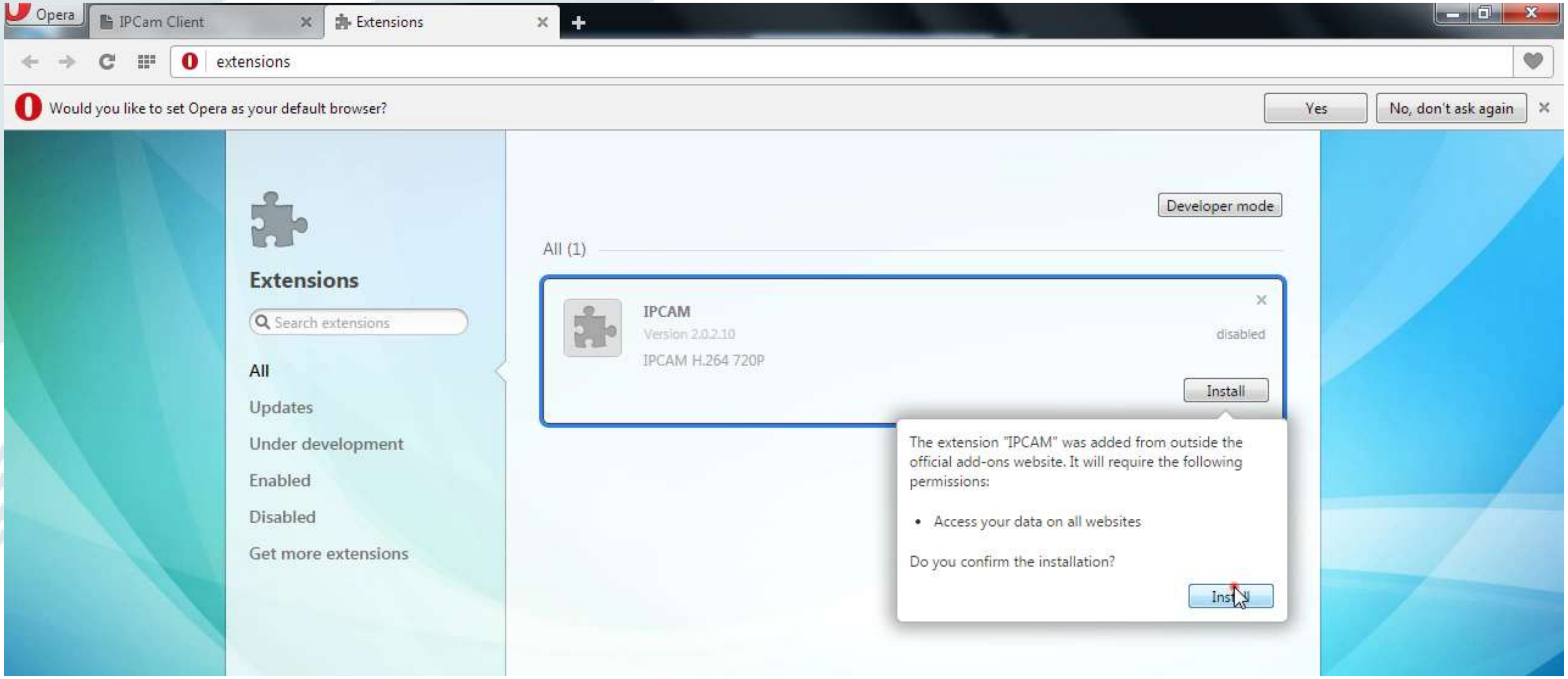
- *Plugin – Chrome*





## Test IP Camera Connection

- *Plugin – Opera*



Would you like to set Opera as your default browser?

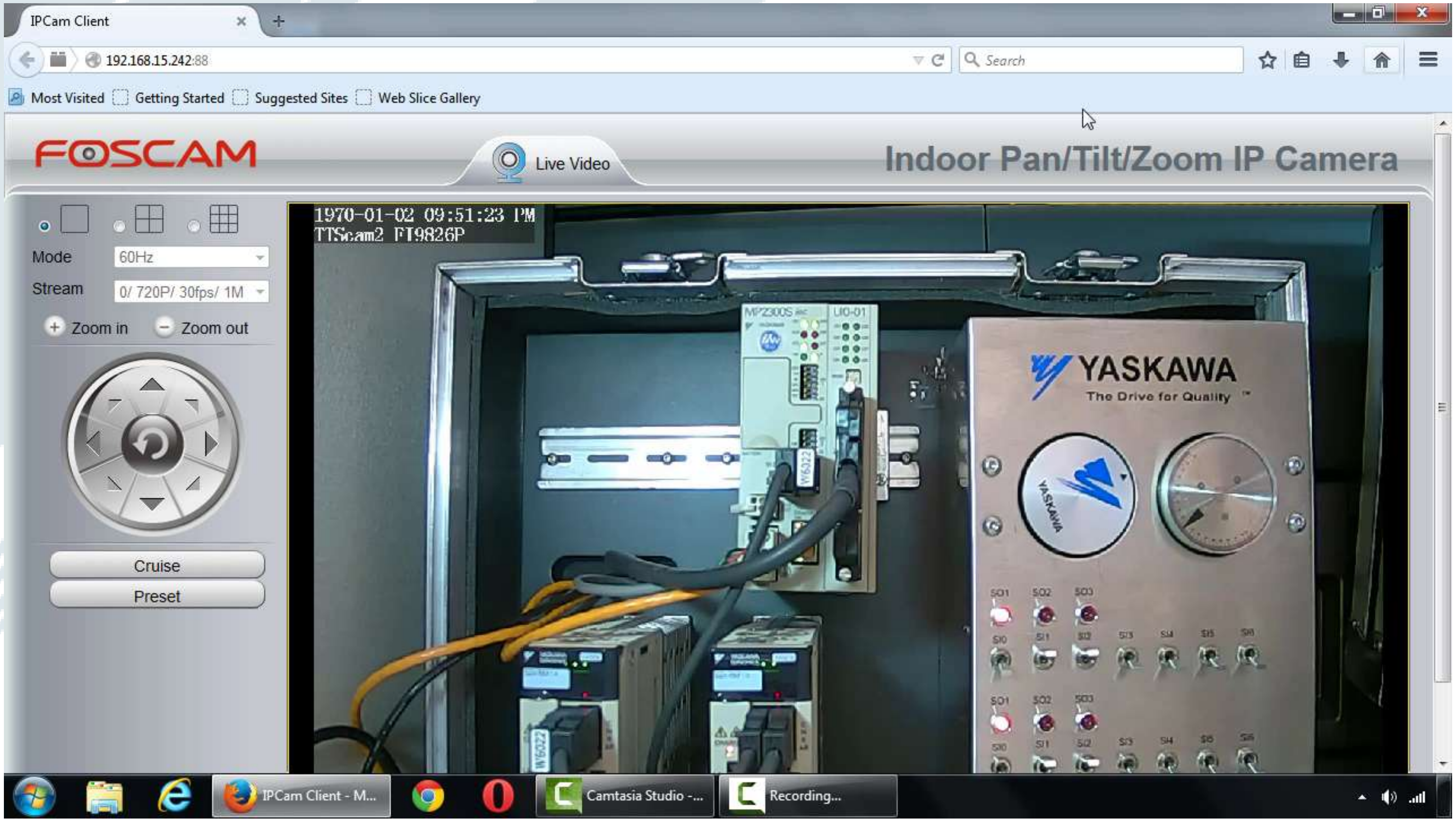
IPCAM
Version 2.0.2.10
IPCAM H.264 720P
disabled

The extension "IPCAM" was added from outside the official add-ons website. It will require the following permissions:

- Access your data on all websites

Do you confirm the installation?

## Test IP Camera Connection


IPCam Client

192.168.15.242:88

Most Visited Getting Started Suggested Sites Web Slice Gallery

FOSCAM Live Video Indoor Pan/Tilt/Zoom IP Camera

Mode 60Hz

Stream 0/ 720P/ 30fps/ 1M

1970-01-02 09:51:23 PM
TTScam2 FI9826P

Driving Value

# Class Project Template

- Purpose
- Save the Hardware Configuration
- Hardware Configuration Summary
- Project Overview
- Run Project
- Toggle Boolean Interface

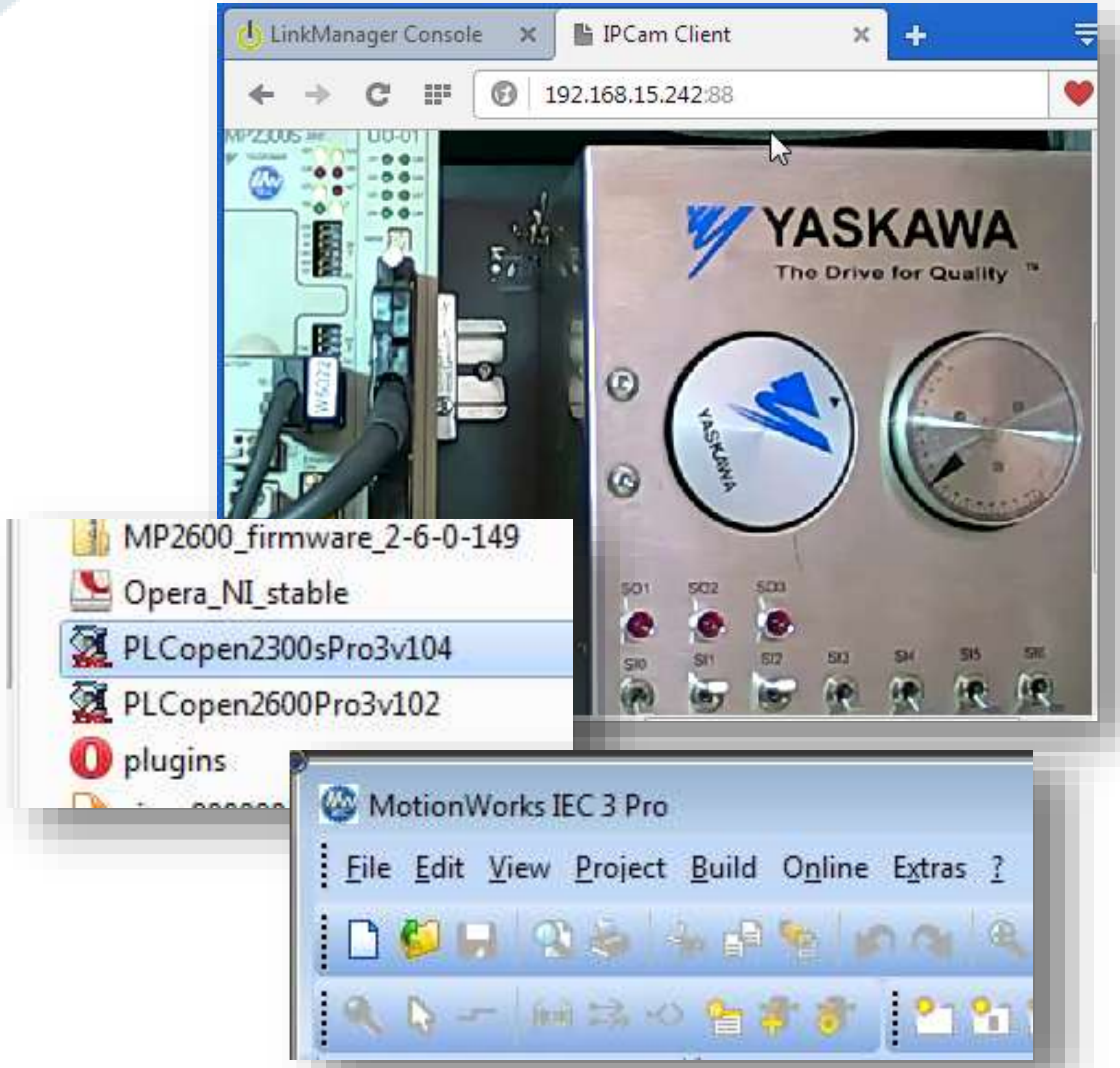
YASKAWA

## Purpose

- *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!*

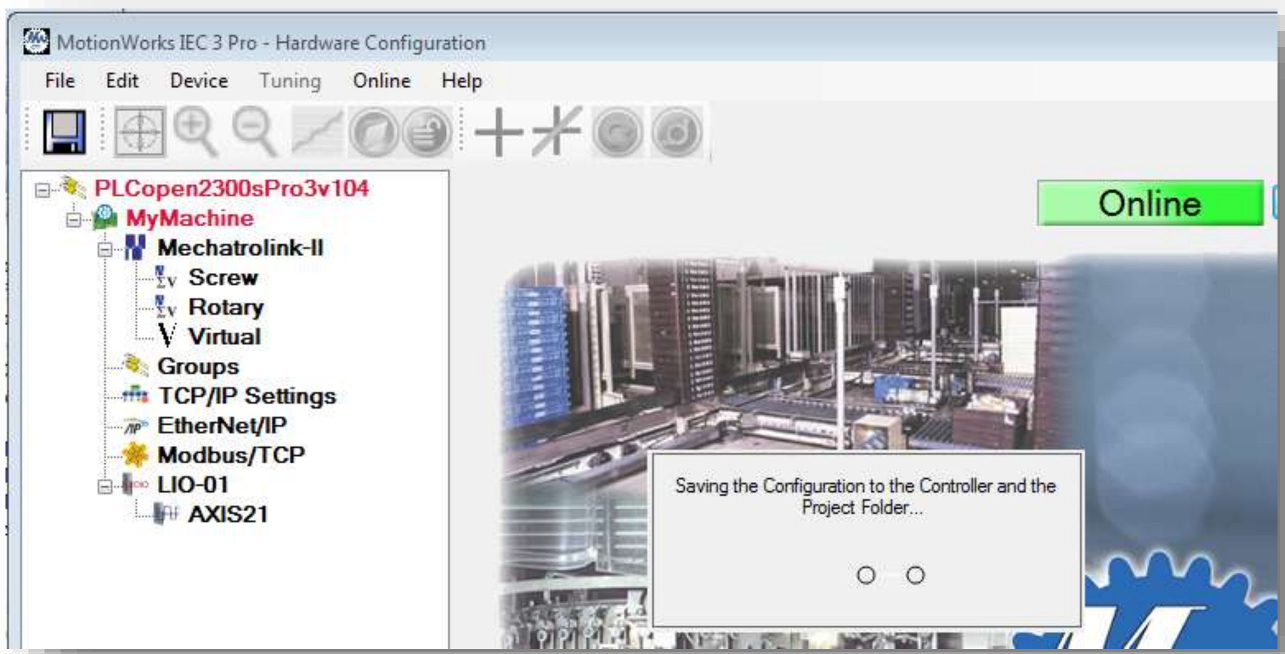
# Requirements

- *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*



## Save Hardware Configuration

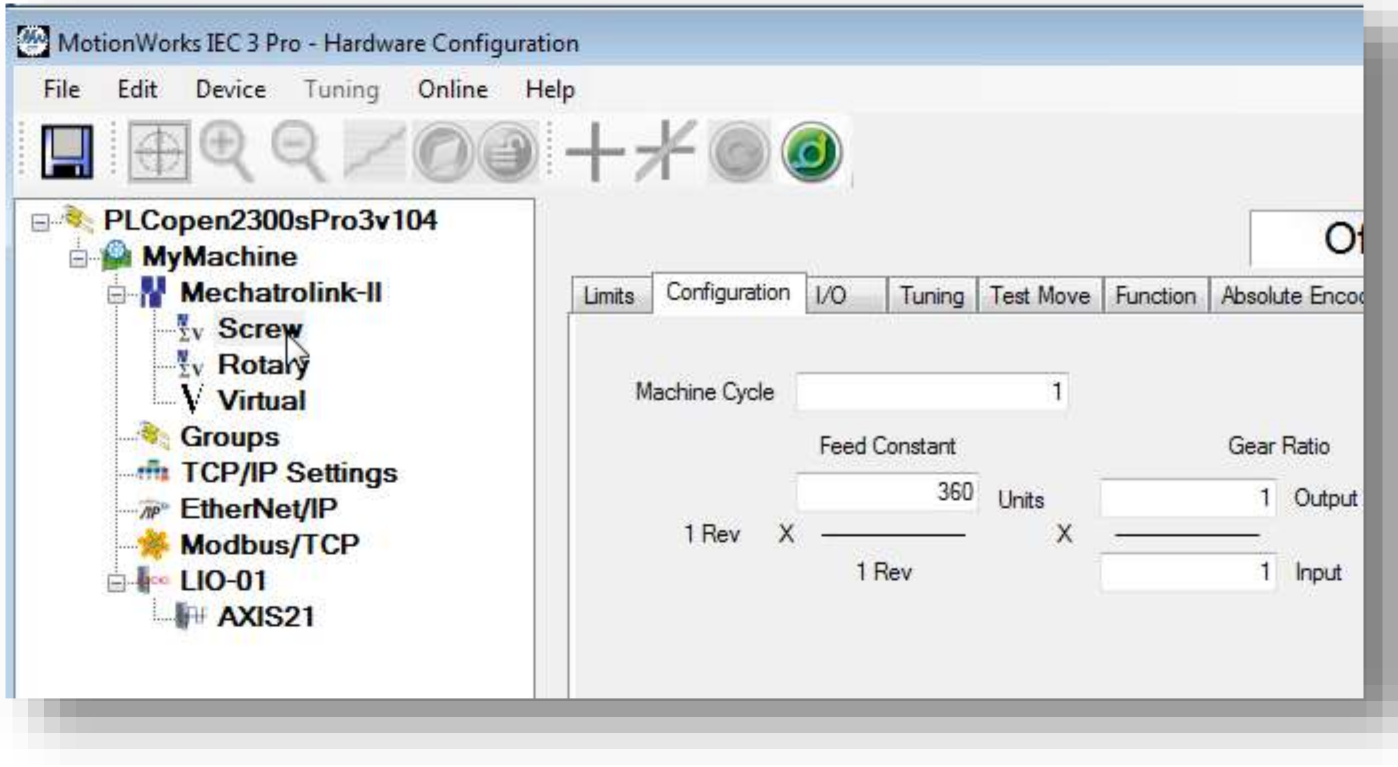
- *Hardware Configuration*
  - *On the Yaskawa toolbar – Move the toolbar*
- *Save Project Hardware Configuration to Controller*
  - *IP address, Connect HC to controller*
  - *Use Offline Configuration*
  - *SAVE online*
  - *Reboot controller*



Saving the Configuration to the Controller and the Project Folder...

## Configuration Summary

- *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*



Machine Cycle: 1

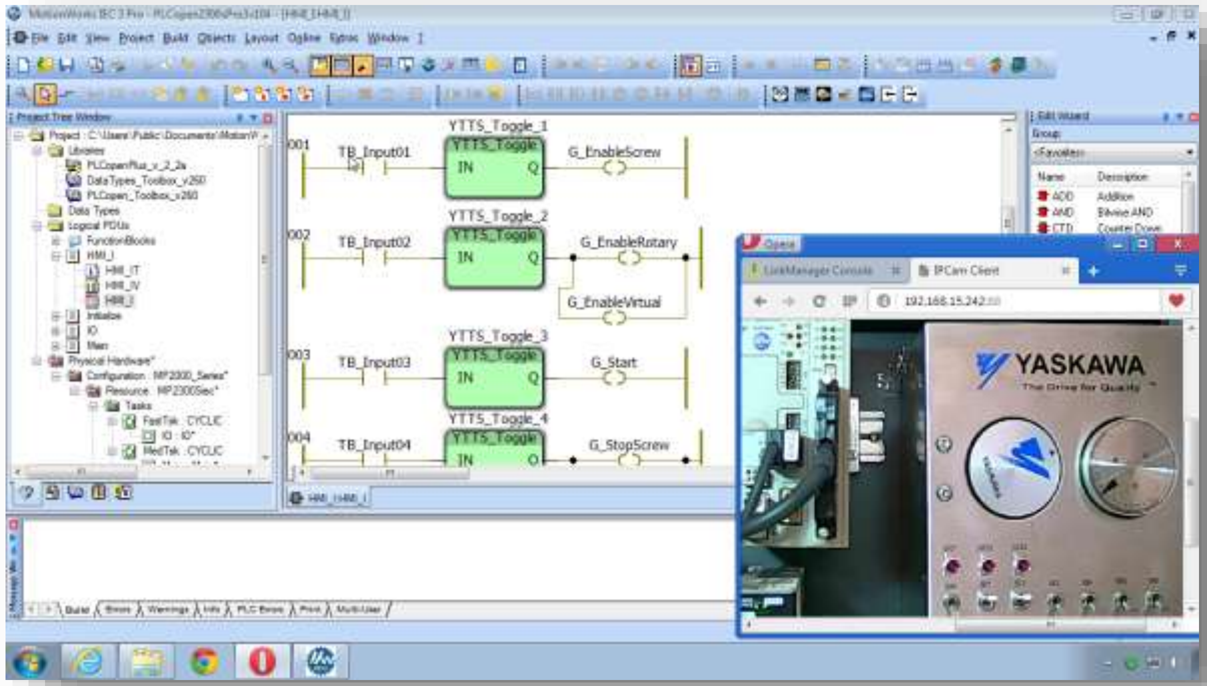
Feed Constant: 1 Rev X 360 Units / 1 Rev

Gear Ratio: X Output 1 / Input 1

Class Project Template

## Project Overview

- *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*

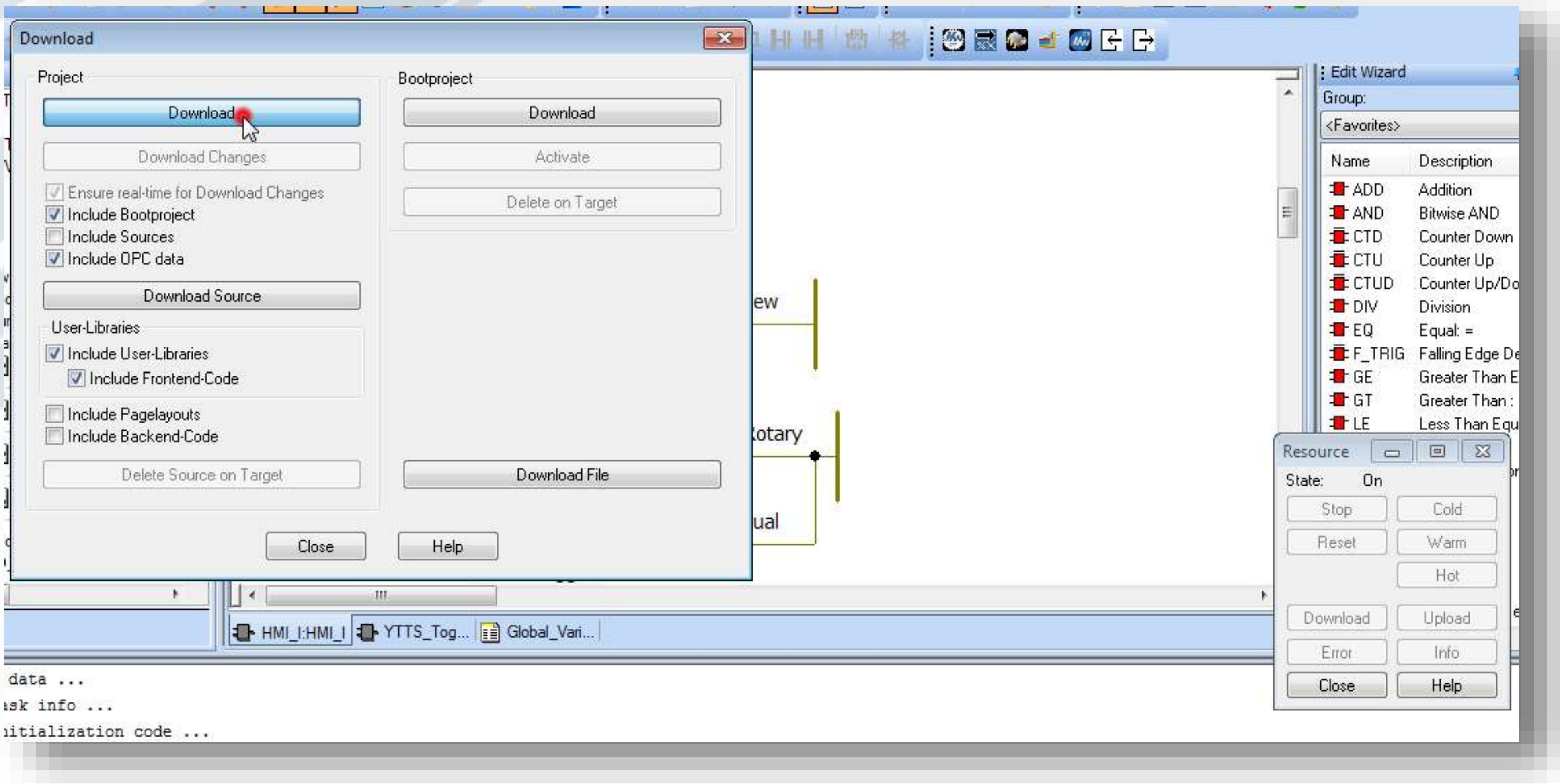


## Software Configuration

- *MotionWorks IEC-Pro Version 3*
  - *Add Download changes button*
    - » *Extras – Options – Commands – Compile/Debug*
    - » *Drag to toolbar*

# Run Project

- *Make*
- *Download*
- *Coldstart*



Ensure real-time for Download Changes [✓ Include Bootproject] Include Sources [✓ Include OPC data]

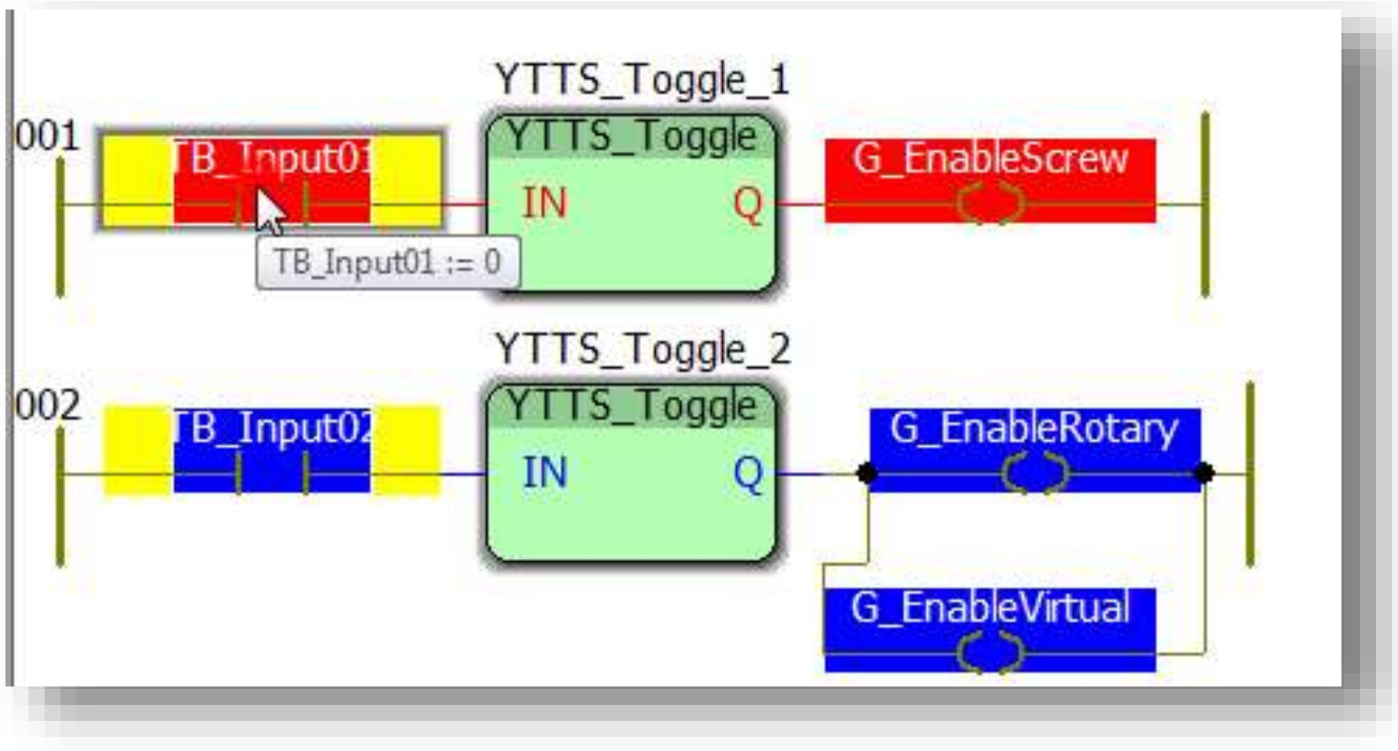
User-Libraries: [✓ Include User-Libraries] [✓ Include Frontend-Code] Include Pagelayouts Include Backend-Code

| Name | Description |
|---|---|
| ADD | Addition |
| AND | Bitwise AND |
| CTD | Counter Down |
| CTU | Counter Up |
| CTUD | Counter Up/Do |
| DIV | Division |
| EQ | Equal: = |
| F_TRIG | Falling Edge De |
| GE | Greater Than E |
| GT | Greater Than : |
| LE | Less Than Equ |

```
data ...
sk info ...
itialization code ...
```

## Toggle Boolean Interface

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



# PLCopen Overview

## PLCopen Intiatives

PLCopen
for efficiency in automation

PLCopen
motion control

PLCopen
safety

PLCopen
Benchmarking

PLCopen
Certified Training
IEC 61131-3

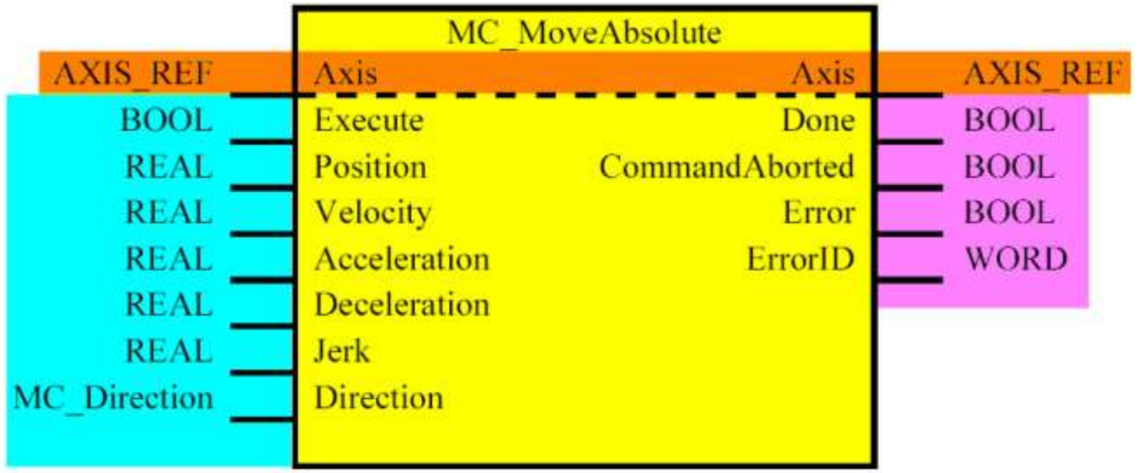
PLCopen
XML
EXTENSIBLE
< MARKUP >
LANGUAGE

PLCopen
reusability level
No. 0000
21.06.2001
DS: 26 of 26

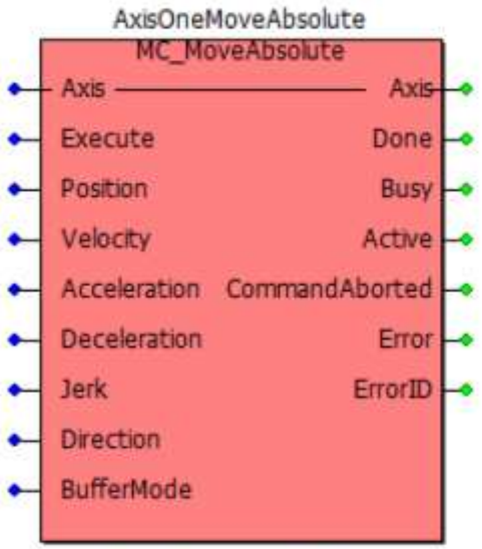
PLCopen
CONFORMITY LEVEL
No. 0000
LD
21.06.2001
Datatypes supported: 26 of 26

## TC2 – Motion Control

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



PLCopen



Implementation

## MPiec Controllers

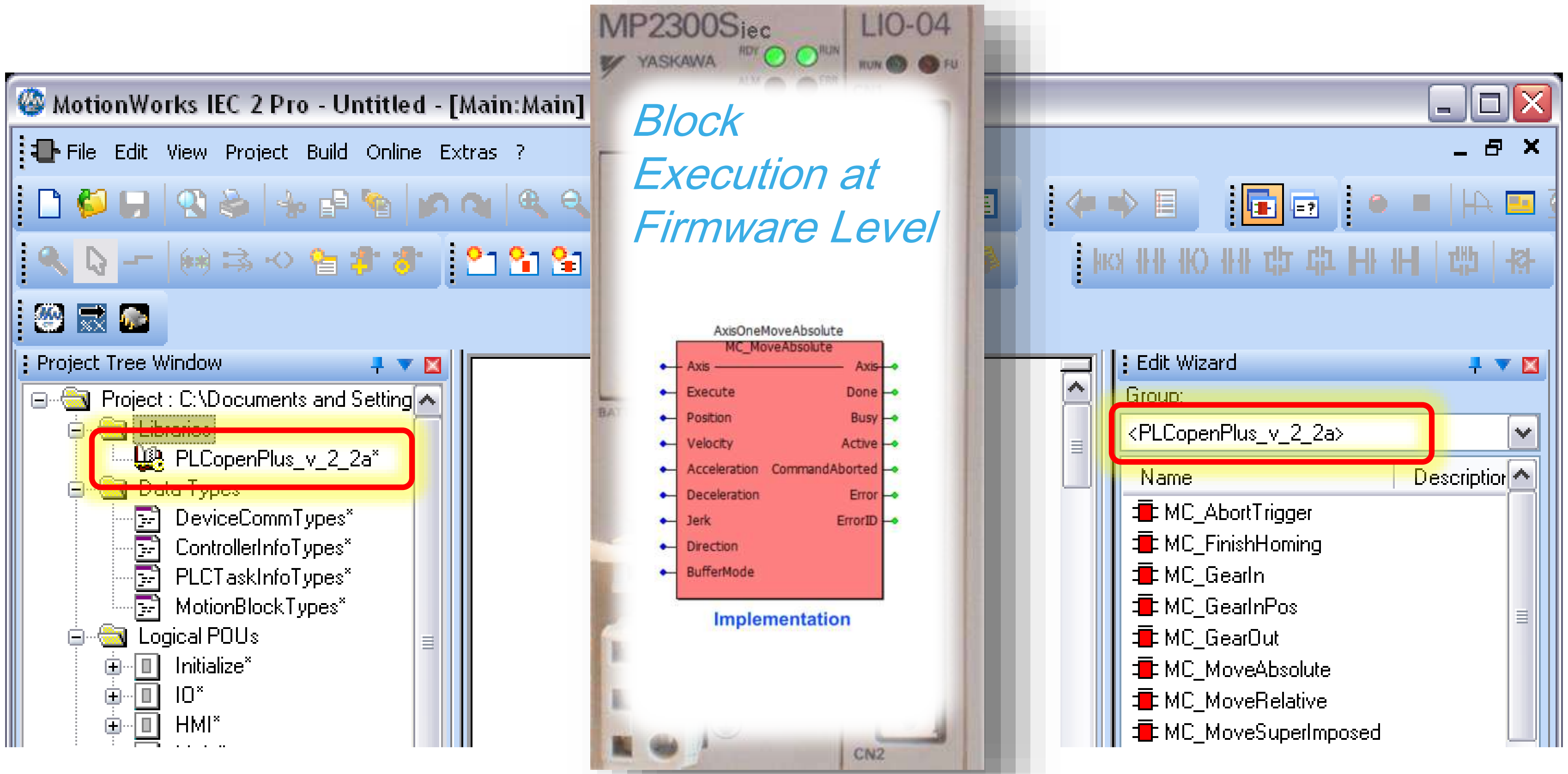
- *MPiec Controllers*
  - *MP2300Siec – 20-Axis, I/O slot x1*
  - *MP2310iec – 20-Axis, I/O slot x3*
  - *MP2600iec – 1-Axis, Multi-function I/O*
  - *MP3200iec – 62-Axis, Mechatrolink III*
  - *MP3300iec – 16-Axis, Mechatrolink III*

PLCopen programming is identical in each of the MPiec controllers.

## PLCopen Summary

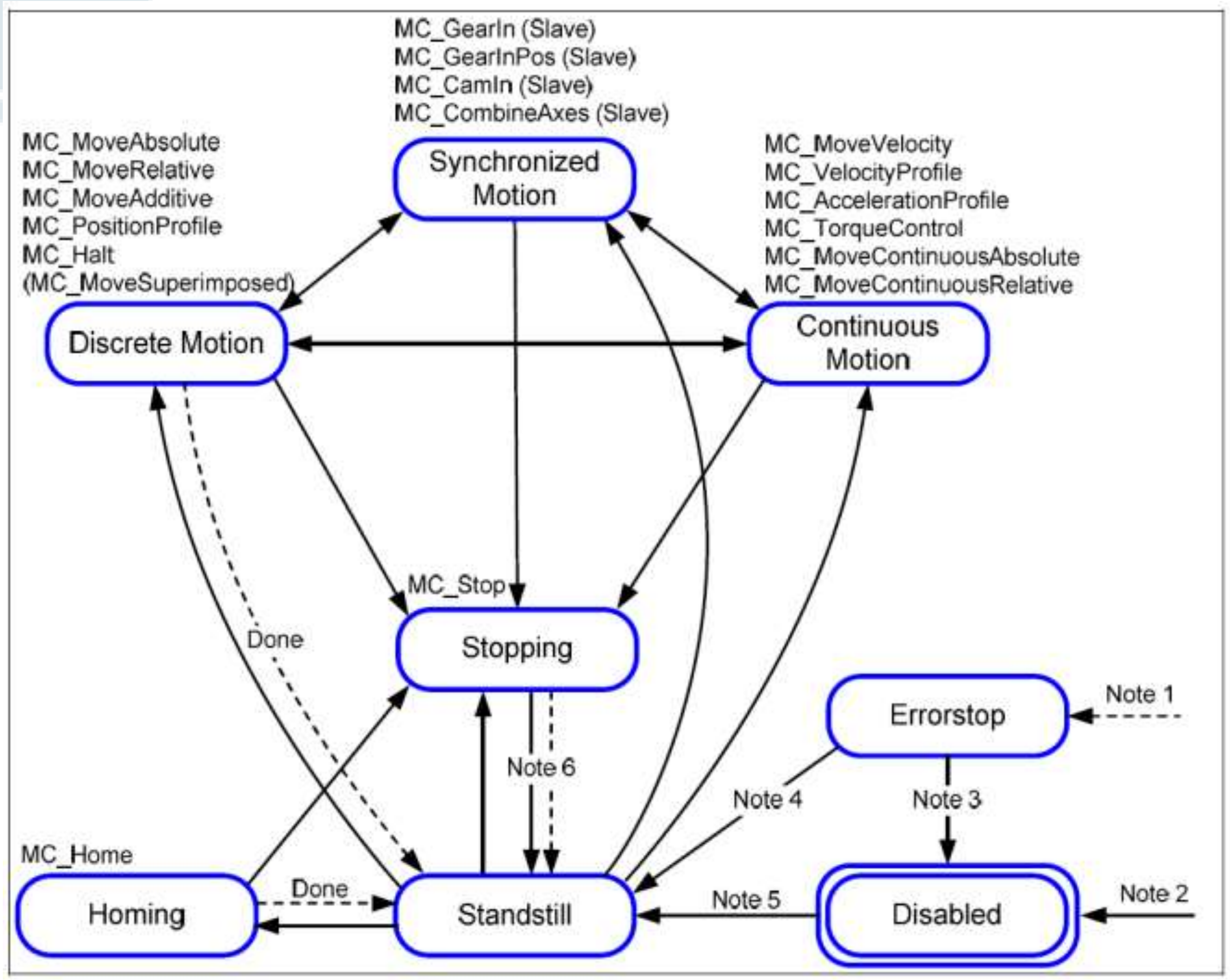
- *Firmware Level Operation*
  - *Synchronized with Mechatrolink*

- *MC_*
  - *Defined by PLCopen*
- *Y_*
  - *Yaskawa Specific*



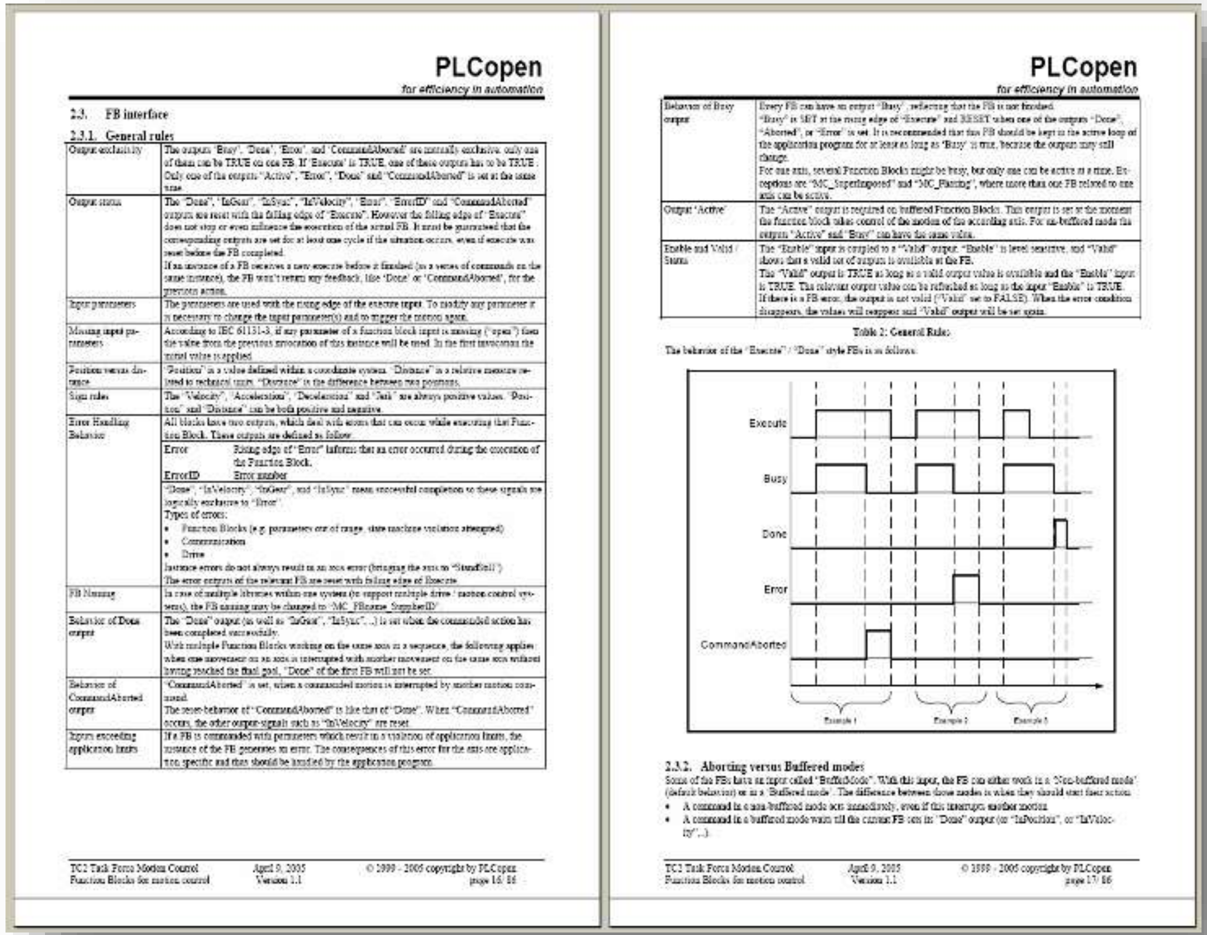
## Motion State Diagram

- *Motion States*
  - *Synchronized Motion*
  - *Discrete Motion*
  - *Continuous Motion*
  - *Stopping*
  - *ErrorStop*
  - *Homing*
  - *Standstill*
  - *Disabled*
- *PLCopen describes which blocks have priority and which state is active*



## Documentation

- *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*



PLCopen
*for efficiency in automation*

### 2.3. FB interface

#### 2.3.1. General rules

| | |
|---|---|
| Output exclusivity | The outputs 'Busy', 'Done', 'Error', and 'CommandAborted' are mutually exclusive: only one of them can be TRUE on one FB. If 'Execute' is TRUE, one of these outputs has to be TRUE. Only one of the outputs "Active", "Error", "Done" and "CommandAborted" is set at the same time. |
| Output status | The "Done", "InGear", "InSync", "InVelocity", "Error", "ErrorID" and "CommandAborted" outputs are reset with the falling edge of "Execute". However the falling edge of "Execute" does not stop or even influence the execution of the actual FB. It must be guaranteed that the corresponding outputs are set for at least one cycle if the situation occurs, even if execute was reset before the FB completed. 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. |
| Input parameters | The parameters are used with the rising edge of the execute input. To modify any parameter it is necessary to change the input parameter(s) and to trigger the motion again. |
| Missing input parameters | 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. |
| Position versus distance | "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. |
| Sign rules | The "Velocity", "Acceleration", "Deceleration" and "Jerk" are always positive values. "Position" and "Distance" can be both positive and negative. |
| Error Handling Behavior | All blocks have two outputs, which deal with errors that can occur while executing that Function Block. These outputs are defined as follow: Error: Rising edge of "Error" informs that an error occurred during the execution of the Function Block. ErrorID: Error number. "Done", "InVelocity", "InGear", and "InSync" mean successful completion so these signals are logically exclusive to "Error". Types of errors: • Function Blocks (e.g. parameters out of range, state machine violation attempted) • Communication • Drive. Instance errors do not always result in an axis error (bringing the axis to "StandStill"). The error outputs of the relevant FB are reset with falling edge of Execute. |
| FB Naming | 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". |
| Behavior of Done output | The "Done" output (as well as "InGear", "InSync", ..) 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. |
| Behavior of CommandAborted output | "CommandAborted" is set, when a commanded motion is interrupted by another motion command. The reset-behavior of "CommandAborted" is like that of "Done". When "CommandAborted" occurs, the other output-signals such as "InVelocity" are reset. |
| Inputs exceeding application limits | If a FB is commanded with parameters which result in a violation of application limits, the instance of the FB generates an error. The consequences of this error for the axis are application specific and thus should be handled by the application program. |

TC2 Task Force Motion Control
Function Blocks for motion control
April 9, 2005
Version 1.1
© 1999 - 2005 copyright by PLCopen
page 16/ 86

PLCopen
*for efficiency in automation*

| | |
|---|---|
| Behavior of Busy output | Every FB can have an output "Busy", reflecting that the FB is not finished. "Busy" is SET at the rising edge of "Execute" and RESET when one of the outputs "Done", "Aborted", or "Error" is set. It is recommended that this FB should be kept in the active loop of the application program for at least as long as 'Busy' is true, because the outputs may still change. For one axis, several Function Blocks might be busy, but only one can be active at a time. Exceptions are "MC_SuperImposed" and "MC_Phasing", where more than one FB related to one axis can be active. |
| Output 'Active' | The "Active" output is required on buffered Function Blocks. This output is set at the moment the function block takes control of the motion of the according axis. For un-buffered mode the outputs "Active" and "Busy" can have the same value. |
| Enable and Valid / Status | The "Enable" input is coupled to a "Valid" output. "Enable" is level sensitive, and "Valid" shows that a valid set of outputs is available at the FB. The "Valid" output is TRUE as long as a valid output value is available and the "Enable" input is TRUE. The relevant output value can be refreshed as long as the input "Enable" is TRUE. If there is a FB error, the output is not valid ("Valid" set to FALSE). When the error condition disappears, the values will reappear and "Valid" output will be set again. |

Table 2: General Rules

The behavior of the "Execute" / "Done" style FBs is as follows:

#### 2.3.2. Aborting versus Buffered modes

Some of the FBs have an input called "BufferMode". With this input, the FB can either work in a 'Non-buffered mode' (default behavior) or in a 'Buffered mode'. The difference between those modes is when they should start their action:

- A command in a non-buffered mode acts immediately, even if this interrupts another motion.
- A command in a buffered mode waits till the current FB sets its "Done" output (or "InPosition", or "InVelocity",..).

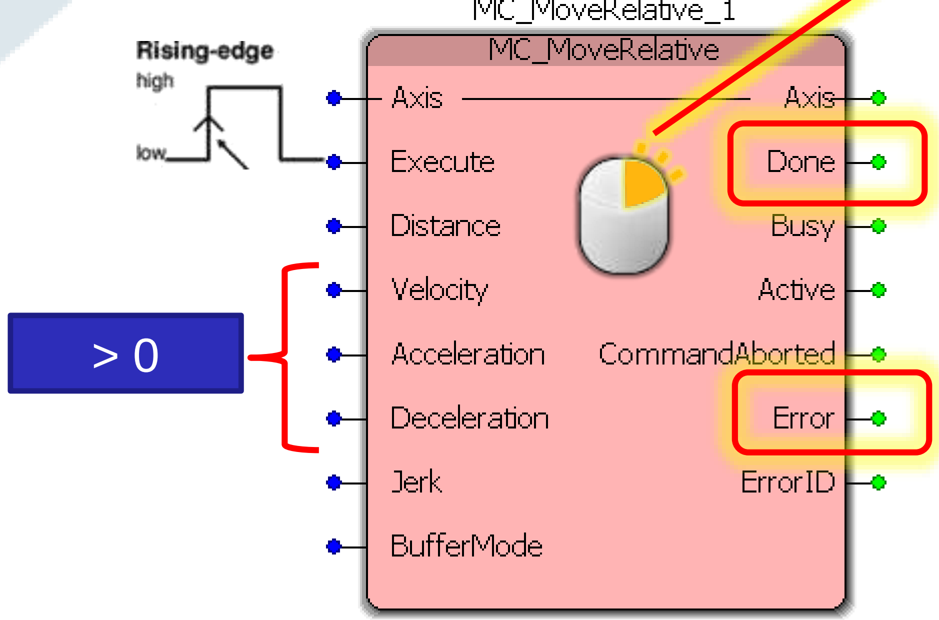
TC2 Task Force Motion Control
Function Blocks for motion control
April 9, 2005
Version 1.1
© 1999 - 2005 copyright by PLCopen
page 17/ 86

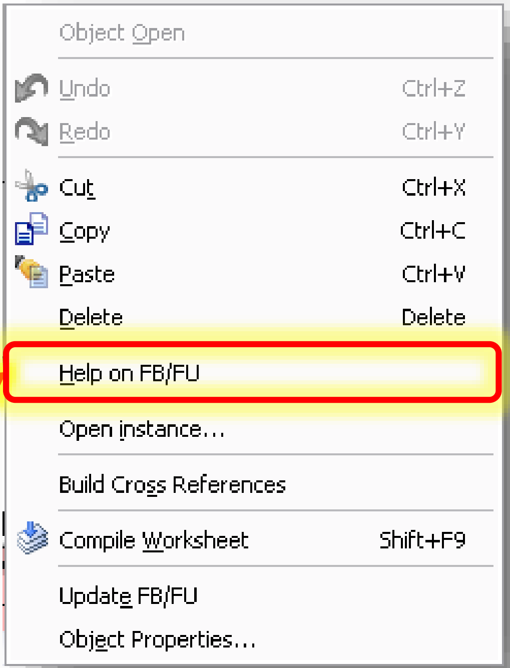
# General Rules Summary

| | |
|---|---|
| Output status | The Done, InGear, InSync, InVelocity, Error, ErrorID and CommandAborted outputs are reset with the falling edge of execute. In must be guaranteed that they are set for at least one cycle if the corresponding situation occurs, even if execute was reset before.<br>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. |
| Input parameters | 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 |
| Missing input parameters | 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. |
| Position versus distance | "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. |
| Sign rules | The Velocity, Acceleration, Deceleration and Jerk are always positive values. Position and Distance can be both positive and negative |
| Error Handling Behavior | All blocks have two outputs which are dealing with errors that can occur while executing a Function Block. These outputs are defined as follow: |
| | **Error** Rising edge of Error informs that an error occurred during the execution of the Function Block.<br>**ErrorID** Error number |
| | Done, InVelocity, InGear, and InSync mean successful completion so these signals are logically exclusive to Error.<br><br>Types of errors:<br>• Function blocks (e.g. parameters outside range, state machine)<br>• Communication<br>• Drive<br>Instance errors are not always resulting in an axis error (bringing the axis to standstill) |
| FB Naming | 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". |
| Behavior of Done output | The Done output (as well as InGear, InSync, ..) is set when the commanded action has been completed successfully.<br>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. |
| Behavior of CommandAborted output | CommandAborted is set, when a commanded motion is interrupted by another motion command or MC_Stop.<br>The reset-behavior of CommandAborted is like Done. When CommandAborted occurs, the other output-signals like InVelocity are reset. |

# General Rules Summary

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

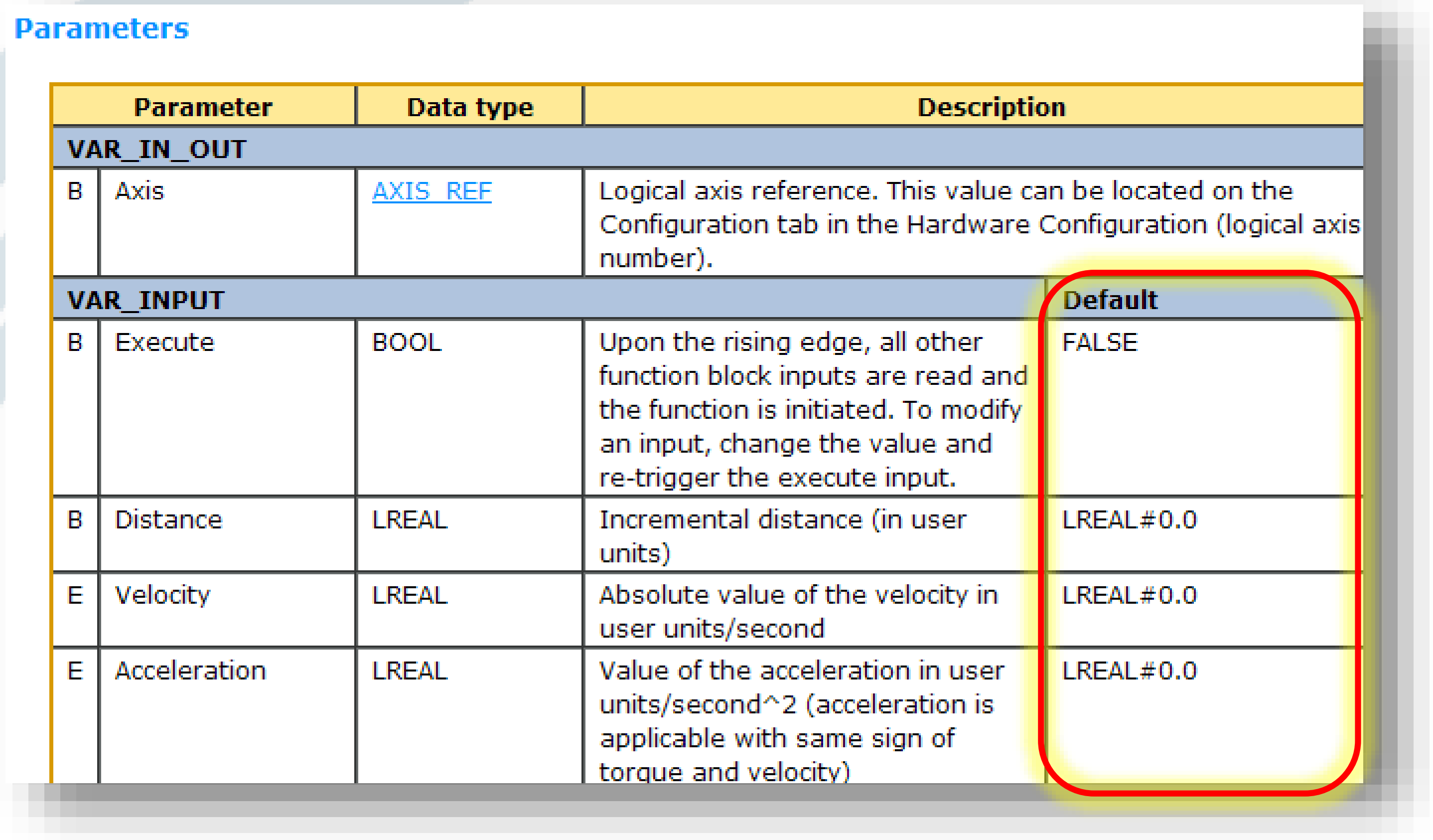




Done & Error never on at the same time

## Initial Value (Default)

- *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*

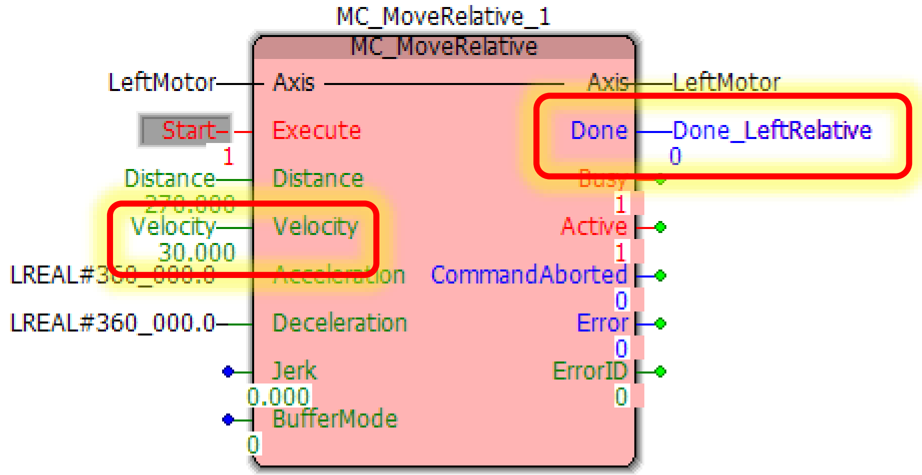


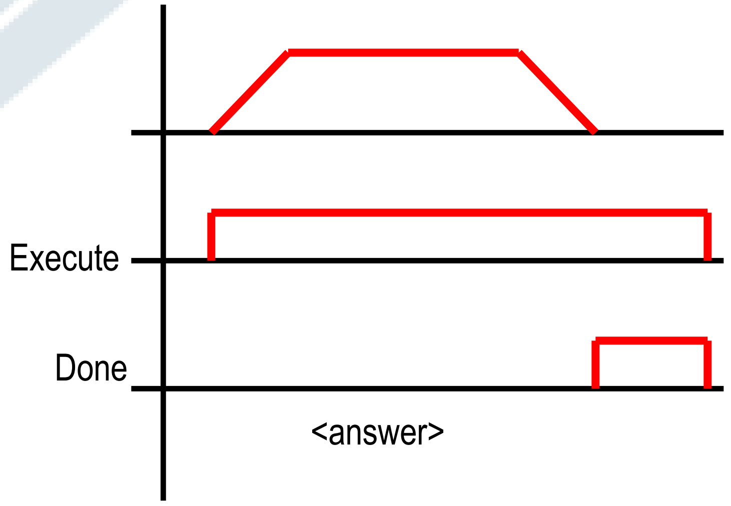
**Parameters**

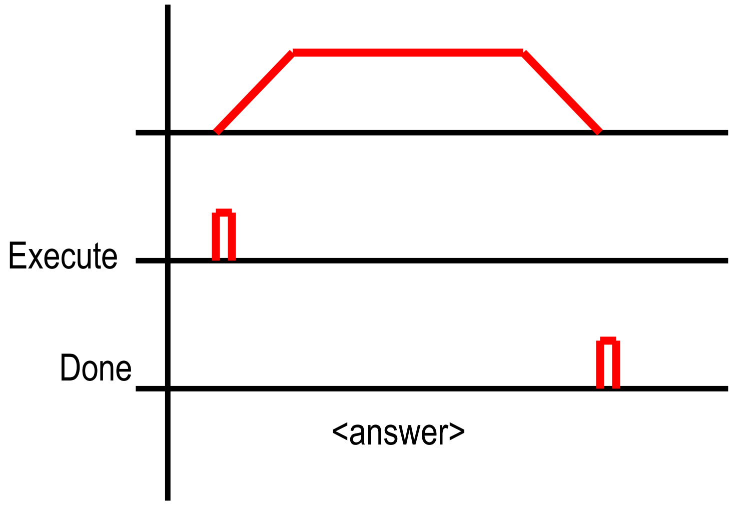
| | Parameter | Data type | Description | |
|---|---|---|---|---|
| **VAR_IN_OUT** | | | | |
| B | Axis | AXIS_REF | Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number). | |
| **VAR_INPUT** | | | | **Default** |
| B | Execute | BOOL | 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. | FALSE |
| B | Distance | LREAL | Incremental distance (in user units) | LREAL#0.0 |
| E | Velocity | LREAL | Absolute value of the velocity in user units/second | LREAL#0.0 |
| E | Acceleration | LREAL | Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity) | LREAL#0.0 |

## DONE Output

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





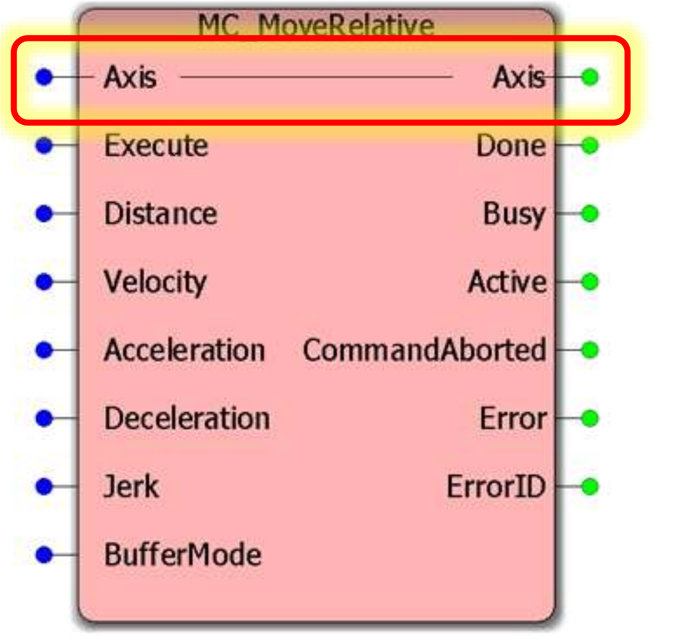


## Axis_Ref

Usage and Purpose
Definition
Axis Name and Number
Initialize Axis Variable

## Usage and Purpose

- ***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*



This Function Block commands a controlled
the time of the execution.

**Parameters**

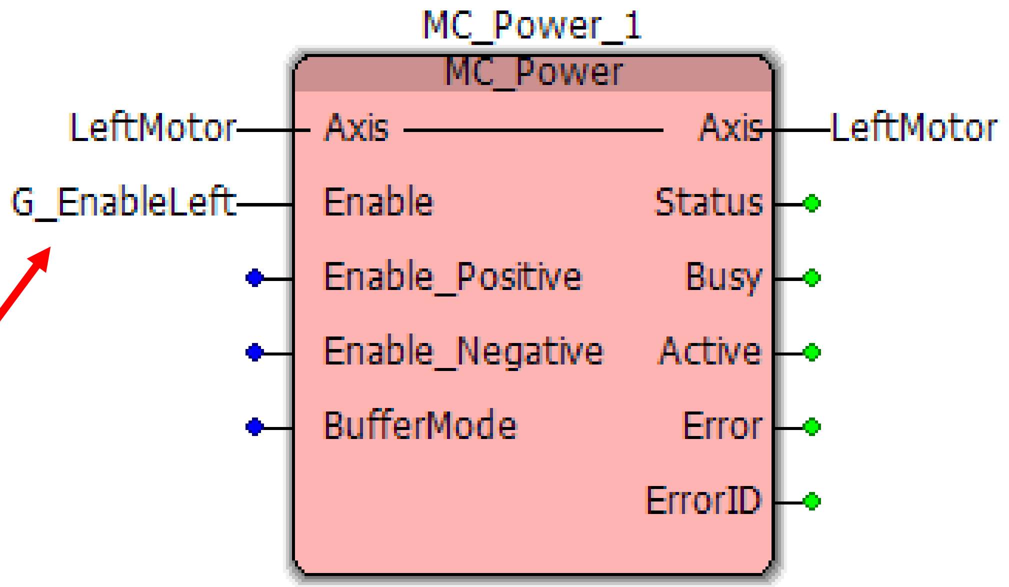
| * | Parameter | Data type |
|---|---|---|
| **VAR_IN_OUT** | | |
| B | Axis | AXIS_REF |

## Usage and Purpose

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

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

| **AXIS_REF** | |
|---|---|
| **Element** | **Data Type** |
| Axis Number | UINT |
| Max Speed | LREAL |
| Encoder Resolution | DINT |

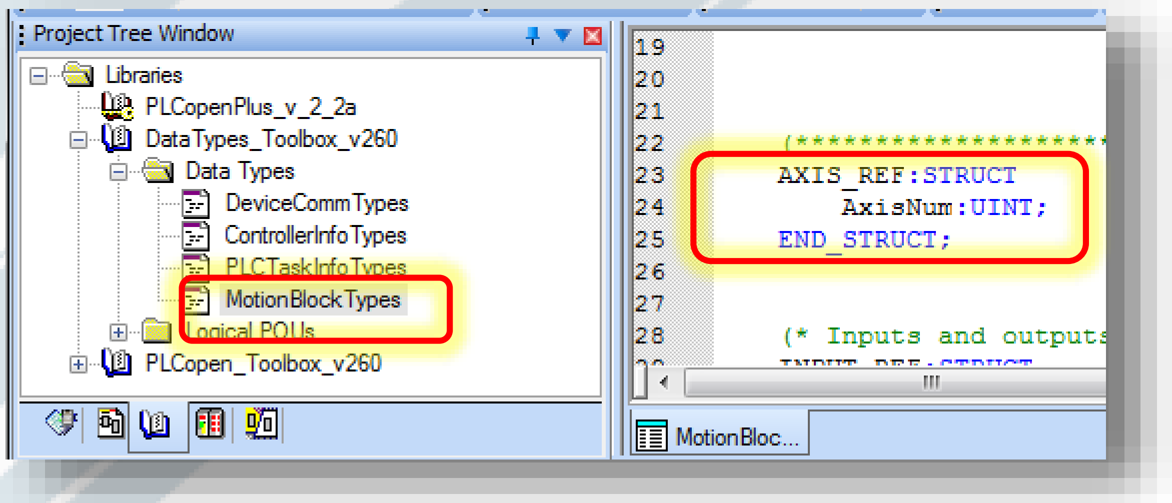


Variable named LeftMotor of data type AXIS_REF

| **LeftMotor** | | |
|---|---|---|
| **Element** | **Data Type** | **Data** |
| Axis Number | UINT | 1 |
| Max Speed | LREAL | 6000 |
| Encoder Resolution | DINT | 8192 |

## Definition

*AXIS_REF is defined under Data Types in DataTypes_Toolbox*



```
(******************
AXIS_REF:STRUCT
    AxisNum:UINT;
END_STRUCT;

(* Inputs and outputs
```

*One Element In the Structure*

- *More elements may be added by Yaskawa in the future*

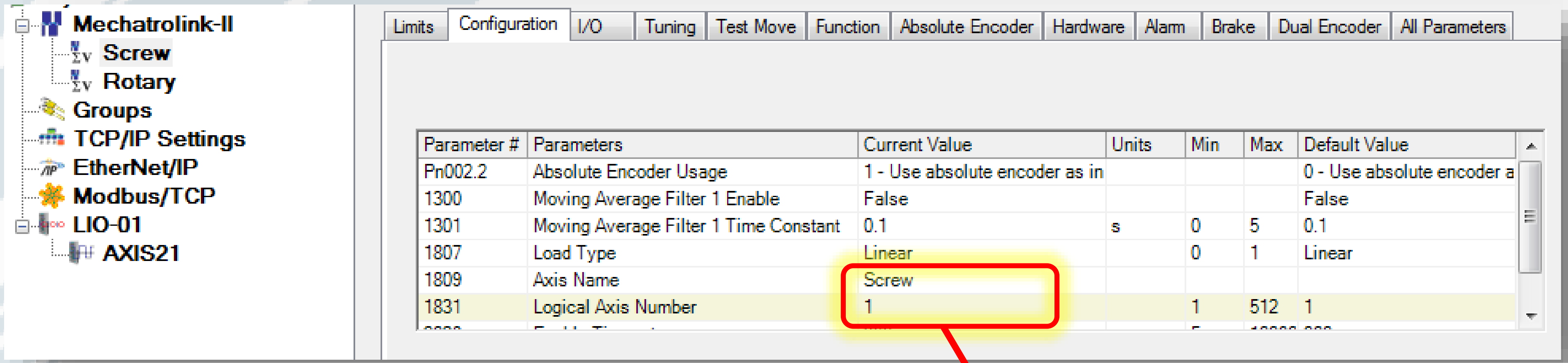
What is the data type of the AxisNum element?

| AXIS_REF | |
|---|---|
| **Element** | **Data Type** |
| AxisNum | UINT |

## Axis Name & Number

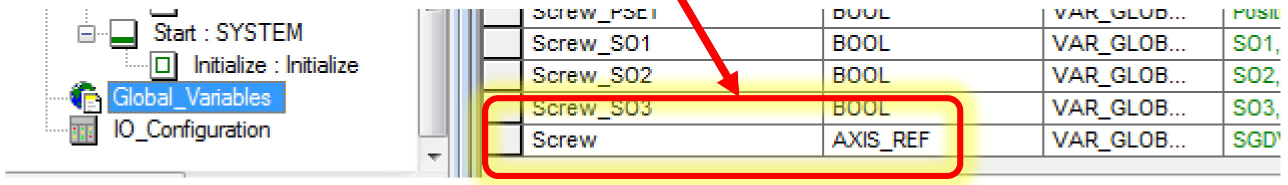
- *Logical Axis Number*
  - *Hardware Configuration*
  - *Axis Parameter #1831*
  - *NOT network node number*

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



| Parameter # | Parameters | Current Value | Units | Min | Max | Default Value |
|---|---|---|---|---|---|---|
| Pn002.2 | Absolute Encoder Usage | 1 - Use absolute encoder as in | | | | 0 - Use absolute encoder a |
| 1300 | Moving Average Filter 1 Enable | False | | | | False |
| 1301 | Moving Average Filter 1 Time Constant | 0.1 | s | 0 | 5 | 0.1 |
| 1807 | Load Type | Linear | | 0 | 1 | Linear |
| 1809 | Axis Name | Screw | | | | |
| 1831 | Logical Axis Number | 1 | | 1 | 512 | 1 |

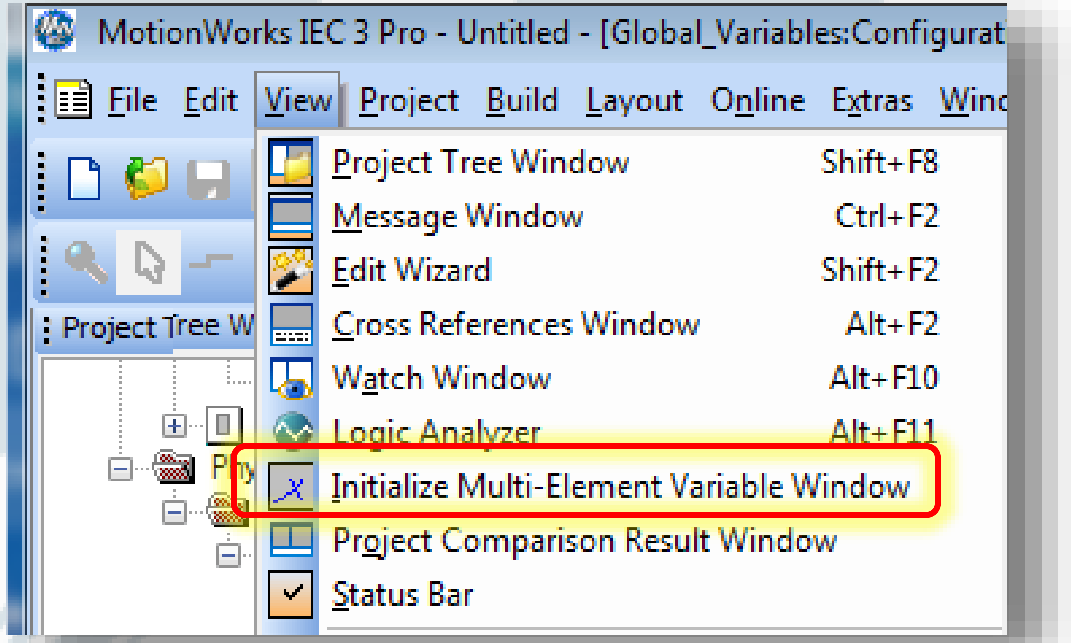
Axis variable is automatically created by Hardware Configuration



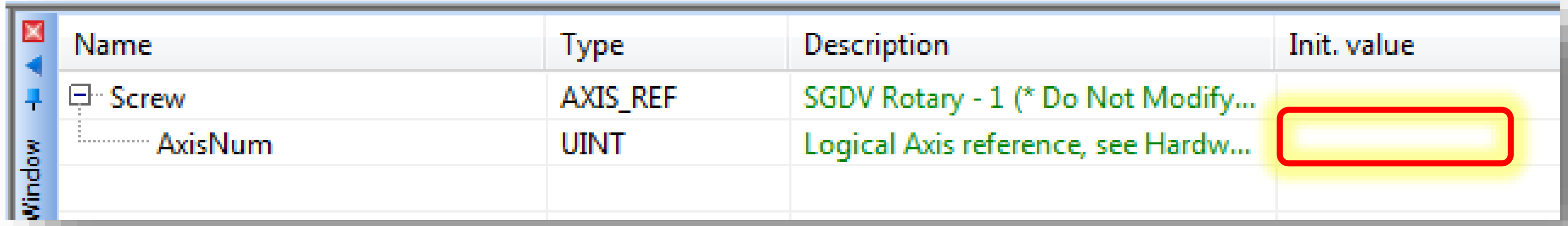
| | | | |
|---|---|---|---|
| Screw_PSET | BOOL | VAR_GLOB... | Posi |
| Screw_SO1 | BOOL | VAR_GLOB... | SO1, |
| Screw_SO2 | BOOL | VAR_GLOB... | SO2, |
| Screw_SO3 | BOOL | VAR_GLOB... | SO3, |
| Screw | AXIS_REF | VAR_GLOB... | SGD' |

## Initialize Axis Variables

- *View – Initialize Multi-Element Variable Window*
  - *Enter the logical axis number*


MotionWorks IEC 3 Pro - Untitled - [Global_Variables:Configurat
File Edit View Project Build Layout Online Extras Winc
Project Tree W
Project Tree Window Shift+F8
Message Window Ctrl+F2
Edit Wizard Shift+F2
Cross References Window Alt+F2
Watch Window Alt+F10
Logic Analyzer Alt+F11
Initialize Multi-Element Variable Window
Project Comparison Result Window
Status Bar

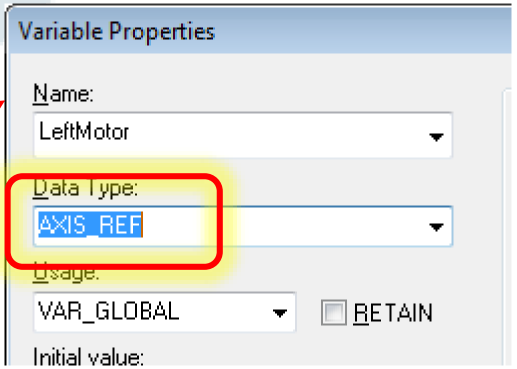
Type the initial value of the structure element in this window



| Name | Type | Description | Init. value |
|---|---|---|---|
| Screw | AXIS_REF | SGDV Rotary - 1 (* Do Not Modify... | |
| AxisNum | UINT | Logical Axis reference, see Hardw... | |

## Text Initialization

- *MotionWorks IEC 2*
  - *Manually create axis variables (with Axis_Ref datatype)*
  - *Initialize axis variables in ST program*
  - *Refer to Quick Start Videos*


Variable Properties

Name: LeftMotor

Data Type: AXIS_REF

Usage: VAR_GLOBAL

☐ RETAIN

Initial value:

**Initialize** (ST)

```
LeftMotor.AxisNum:=UINT#1;
```

```
LeftMotor.AxisNum:=UINT#1;
```

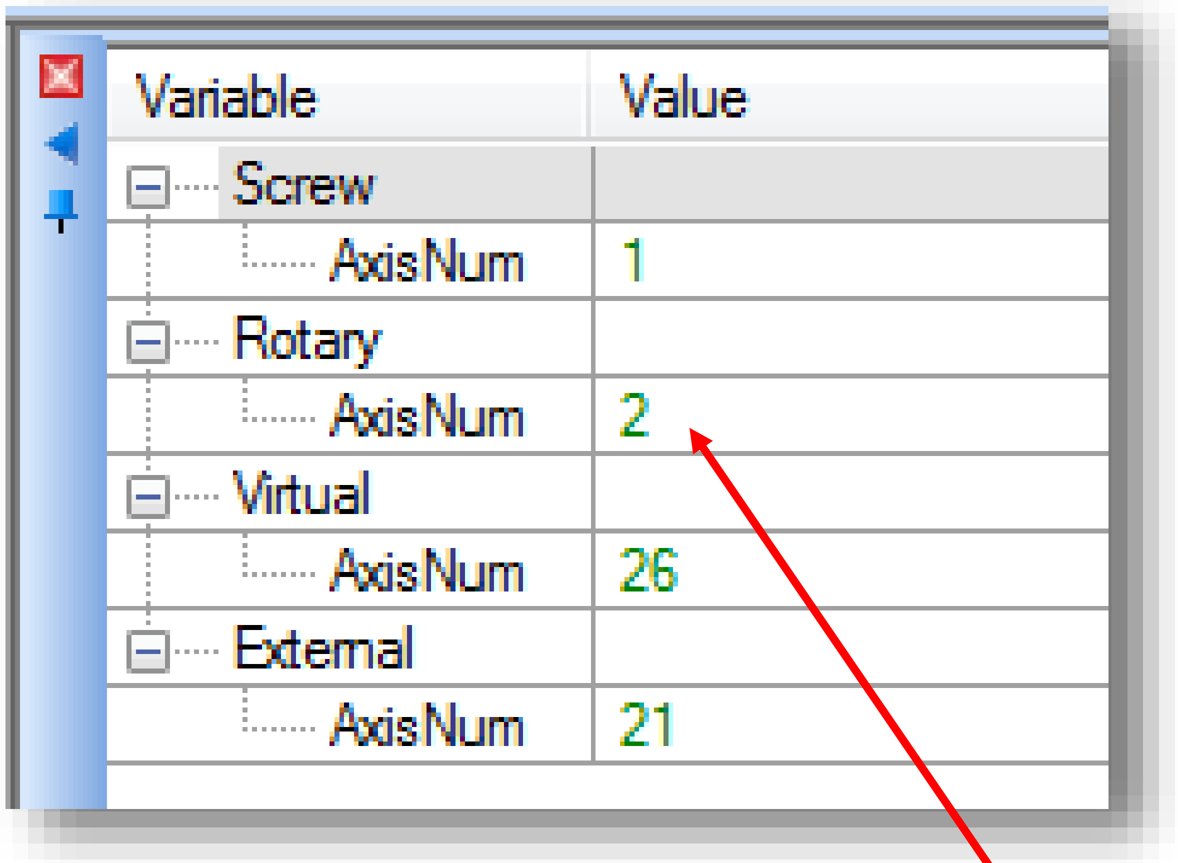
**LeftMotor**

| Element | Data Type | Data |
|---|---|---|
| AxisNum | UINT | 1 |

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

## Confirmation

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



| Variable | Value |
|---|---|
| Screw | |
| AxisNum | 1 |
| Rotary | |
| AxisNum | 2 |
| Virtual | |
| AxisNum | 26 |
| External | |
| AxisNum | 21 |

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

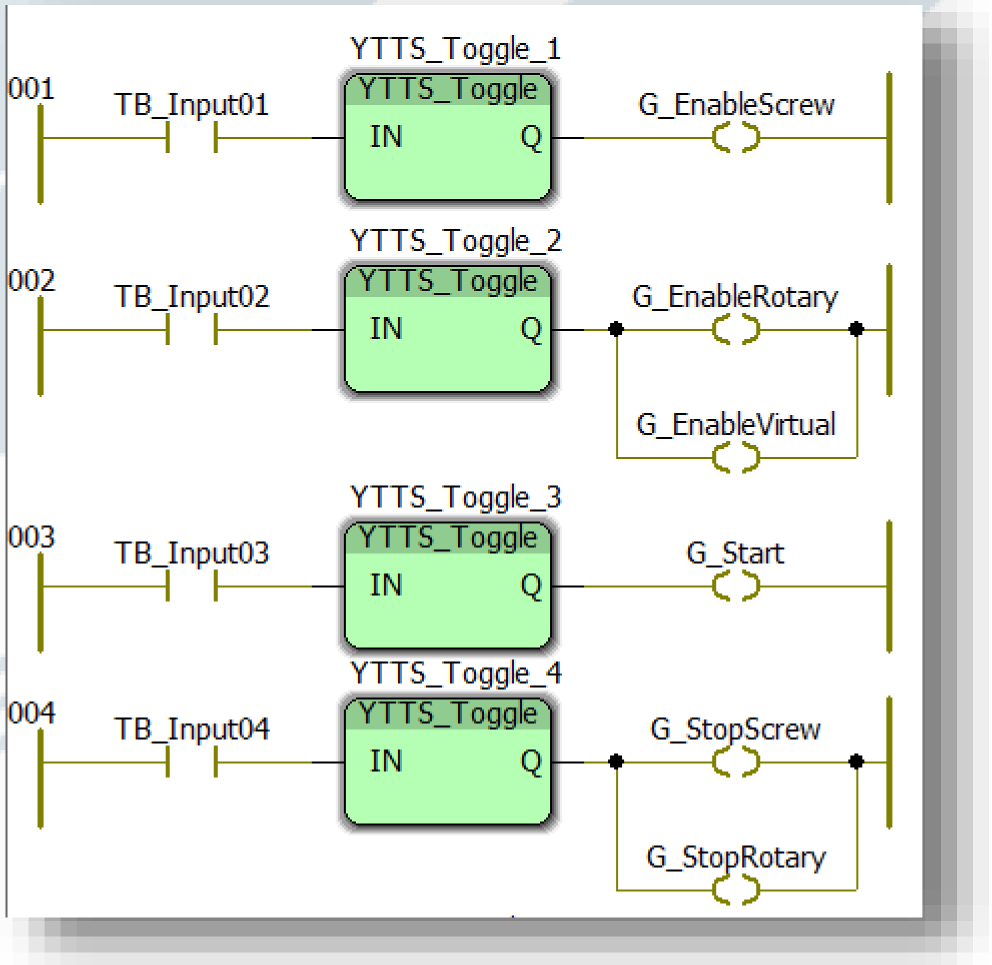
## Servo Enable

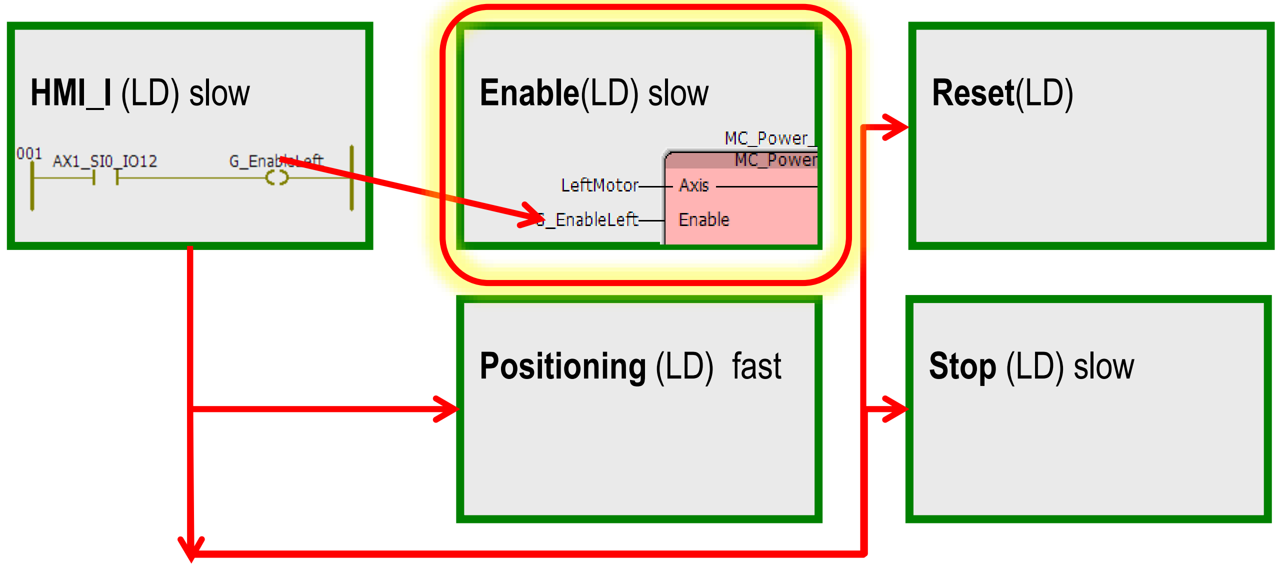
YASKAWA

# Servo Enable

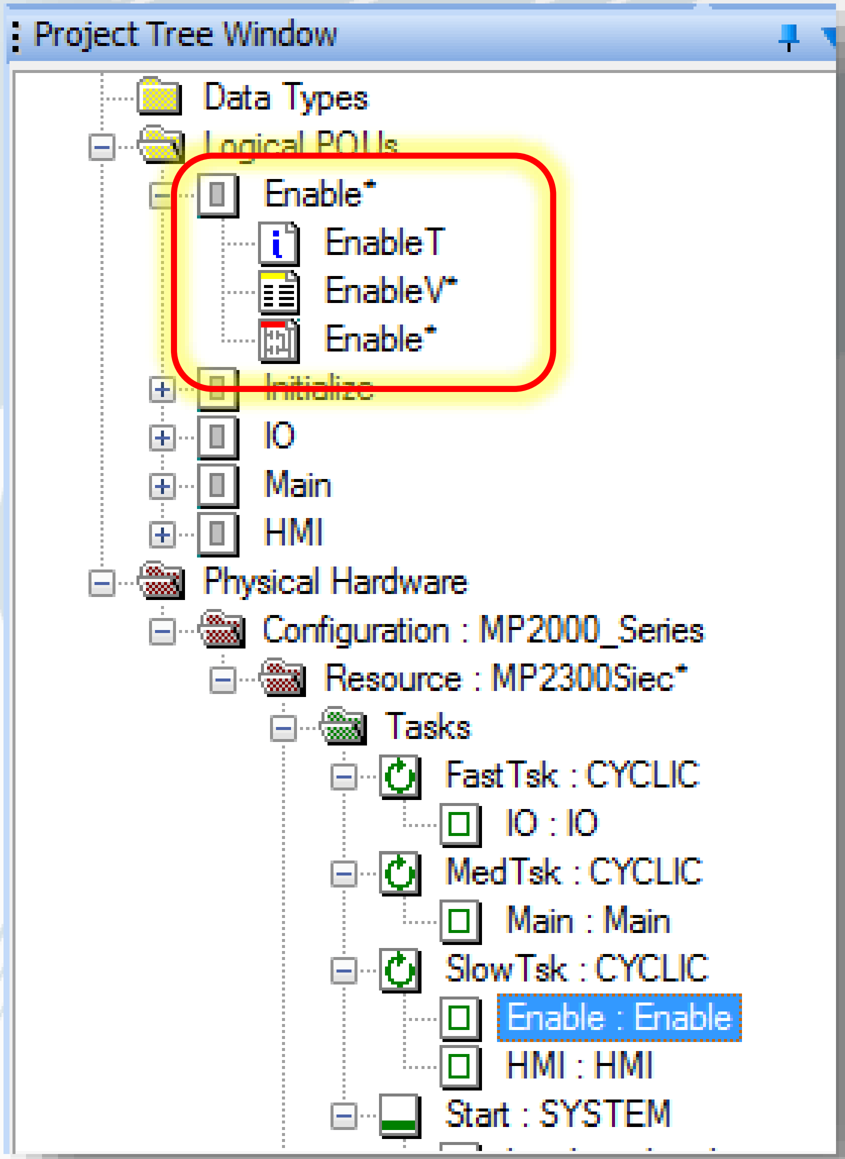
## Program Map

- *HMI_I*
  - *Part of class project*





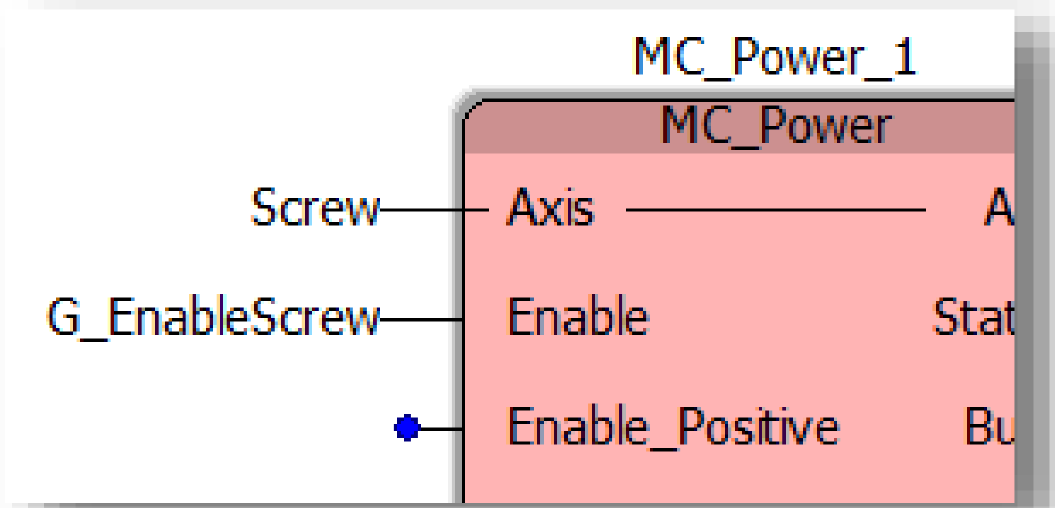
## Enable Program



Project Tree Window

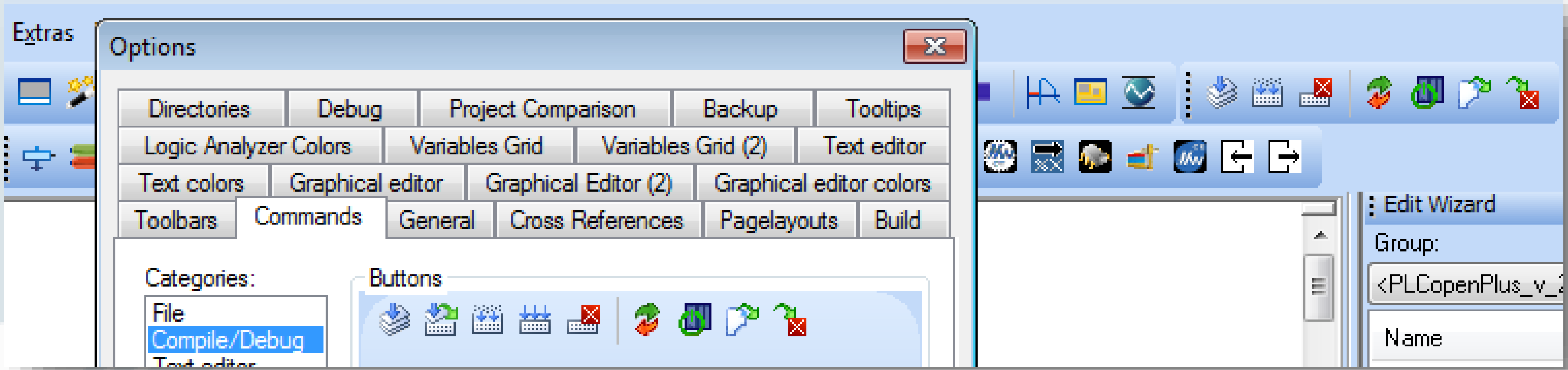
```
Data Types
Logical POUs
  Enable*
    EnableT
    EnableV*
    Enable*
  Initialize
  IO
  Main
  HMI
Physical Hardware
  Configuration : MP2000_Series
    Resource : MP2300Siec*
      Tasks
        FastTsk : CYCLIC
          IO : IO
        MedTsk : CYCLIC
          Main : Main
        SlowTsk : CYCLIC
          Enable : Enable
          HMI : HMI
        Start : SYSTEM
```

- *Create Enable POU*
  - *POU Type: Program*
  - *Language: LD*
  - *Run in the Slow task*
- *Add MC_Power*
  - *Axis*
  - *Enable*



## Download Changes

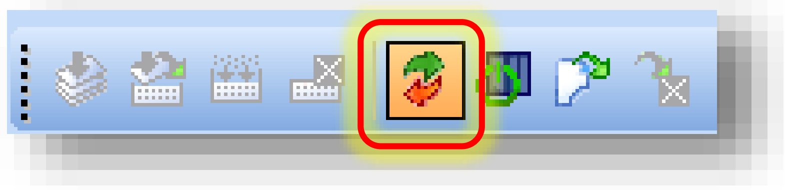
- *Shortcut Button*
  - *Extras – Options - Commands*



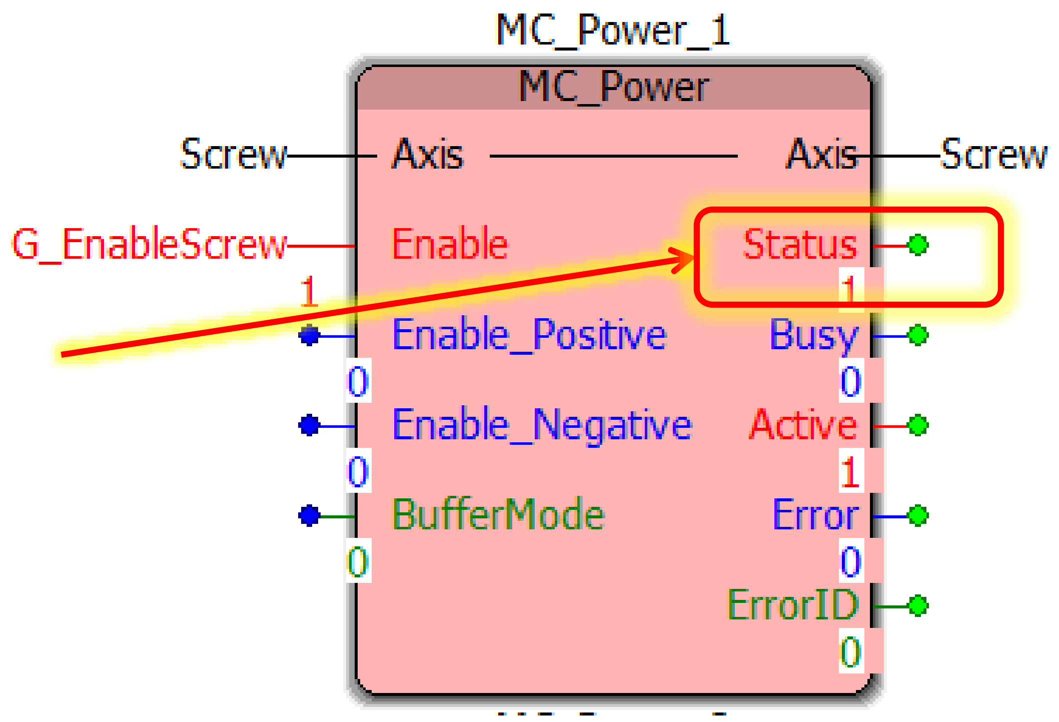
# Servo Enable

# MC_Power

- *Use Debug Mode*
  - *Program and test Screw*
  - *Program and test RightMotor*

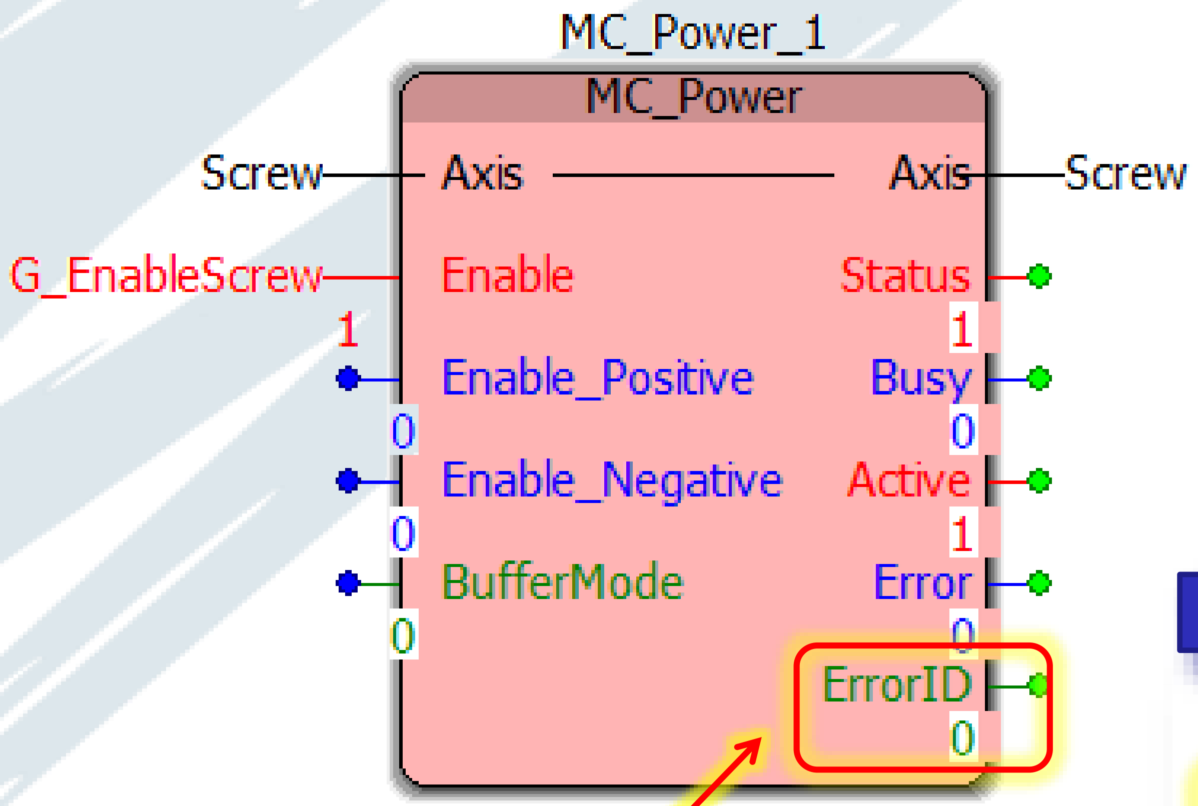


- *MC_Power.Status*
  - *Status of the command*
  - *Updates at the application scan rate*



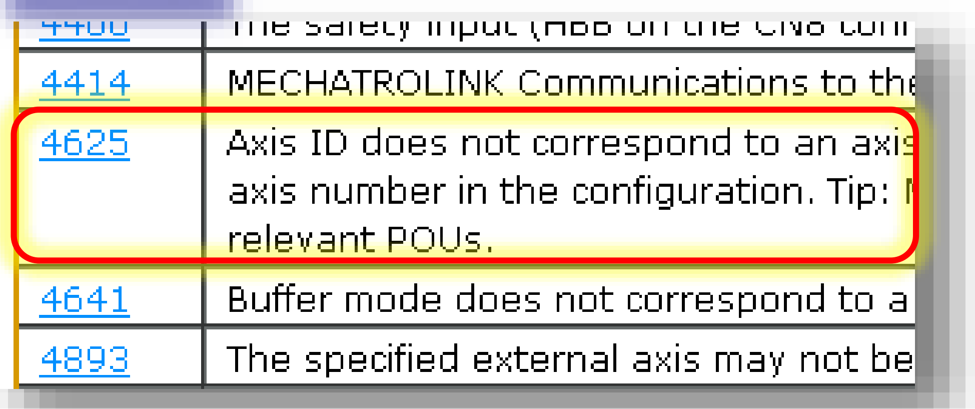
## Servo Enable

## Troubleshooting



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

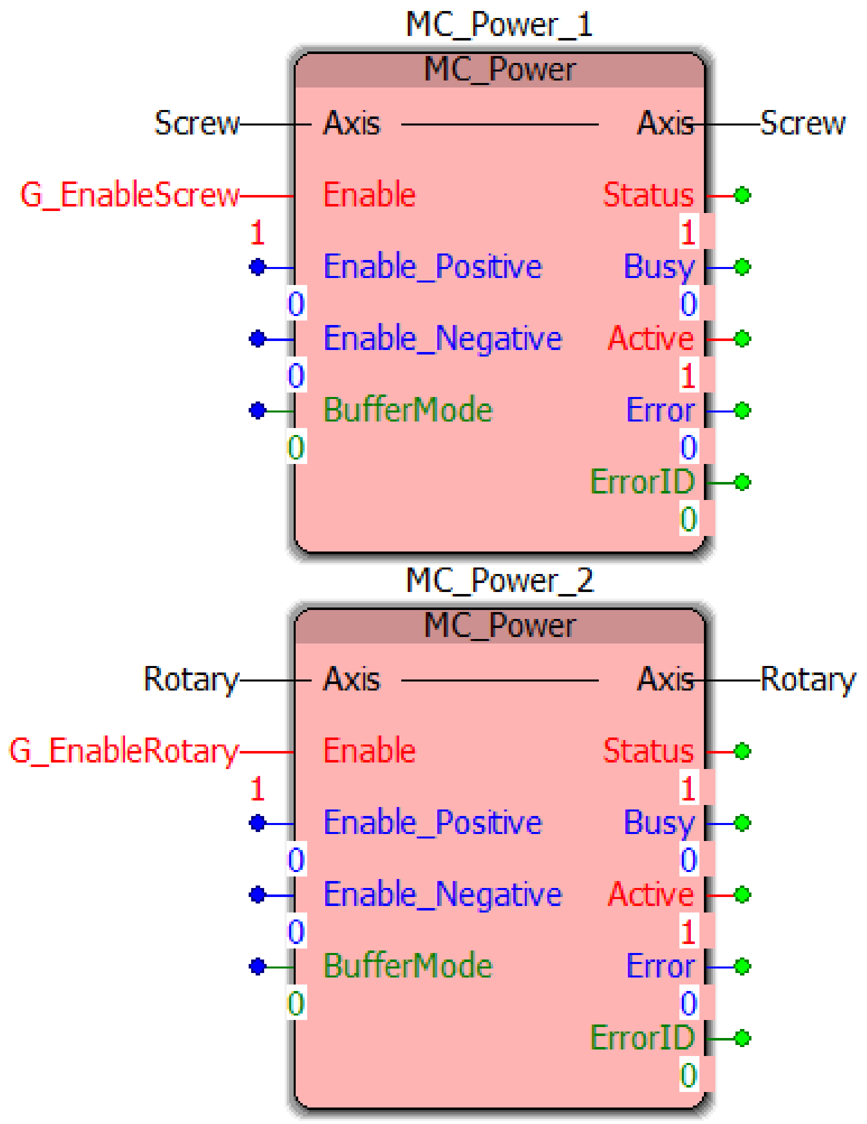
See HELP



| | |
|---|---|
| 4400 | The safety input (HBB on the CN8 conn |
| 4414 | MECHATROLINK Communications to the |
| 4625 | Axis ID does not correspond to an axis axis number in the configuration. Tip: relevant POUs. |
| 4641 | Buffer mode does not correspond to a |
| 4893 | The specified external axis may not be |

# Enable All Axes

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



## Positioning

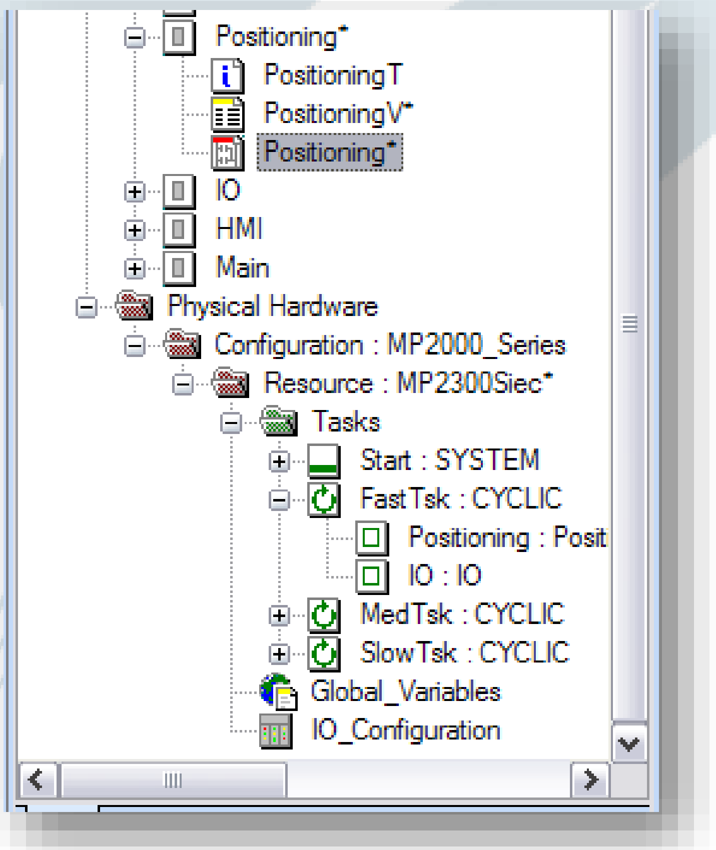
MC_MoveRelative
MC_MoveAbsolute
Timers
Move Sequence

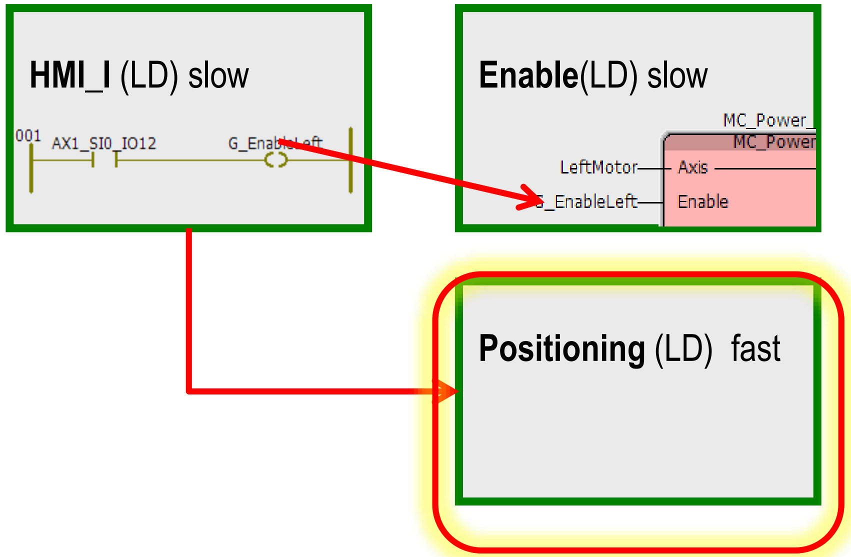
## Positioning Program

*Create Positioning (program POU)*

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

*Refer to Quick Start Video*



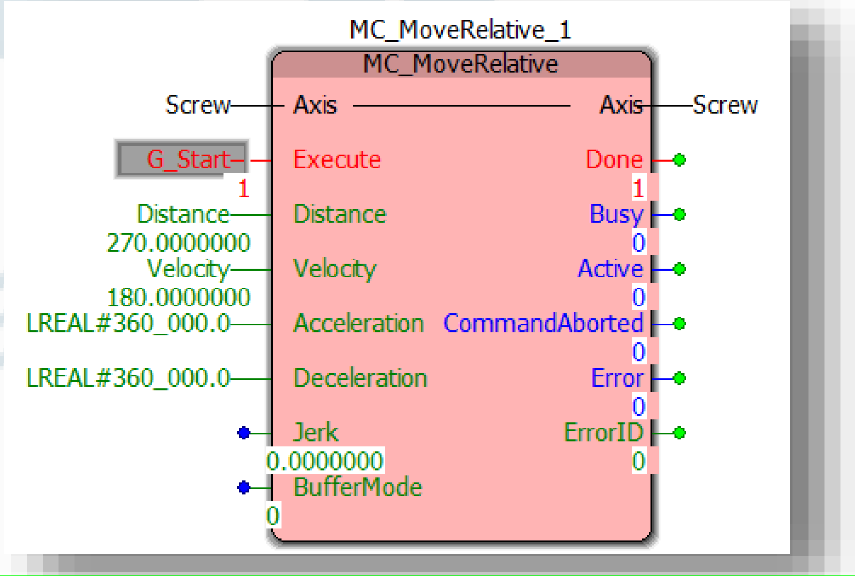


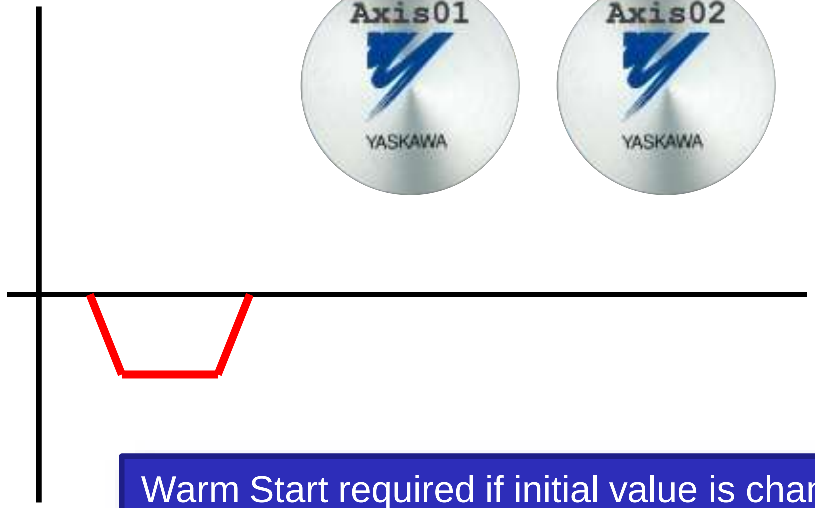
# MC_MoveRelative

- *Positioning (program POU)*
  - *MC_MoveRelative Function Block*

**Screw Move Profile**

| Input | Initial Value | Unit | Note |
|---|---:|---|---|
| Distance | -270.0 | mm | Use Variable |
| Velocity | 180.0 | mm/sec | Use Variable |
| Accel | 360000.0 | mm/sec² | Use Literal LREAL#360_000.0 |
| Decel | 360000.0 | mm/sec² | Use Literal LREAL#360_000.0 |

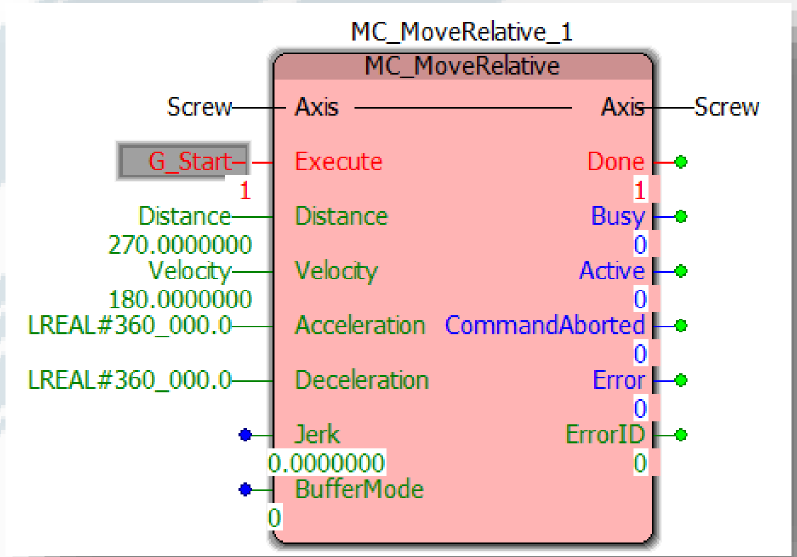


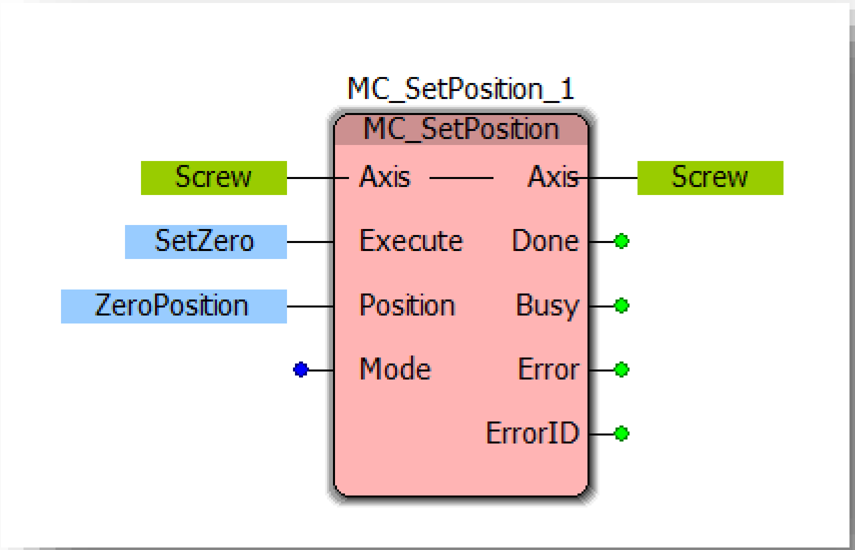


Warm Start required if initial value is changed.

## Set Zero Position

- *Quick Zero Set (optional)*
  - *Repeat relative moves until at mechanical zero*
  - *Use MC_SetPosition*
  - *Repeat for Rotary*





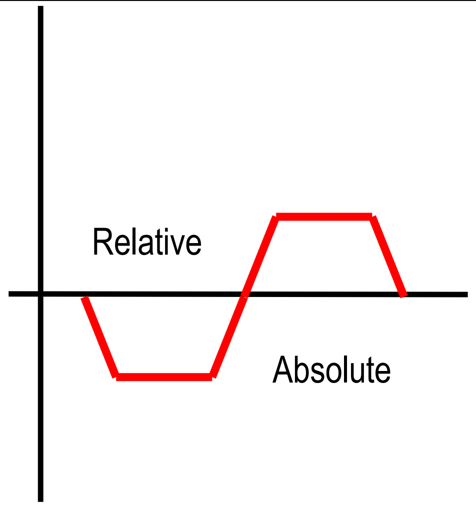
Change distance by trial and error. Re execute until arrow on motor wheel is pointing up

## MC_MoveAbsolute

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

**LeftMotor Move Profile**

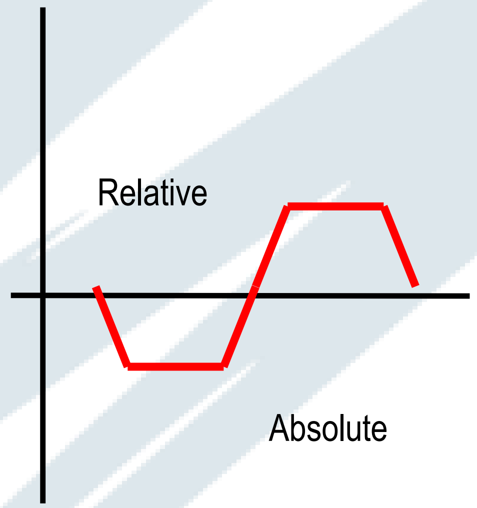
| **Input** | **Initial Value** | **Unit** | **Note** |
|---|---:|---|---|
| Position | 0.0 | mm | Use Variable |
| Velocity | 180.0 | mm/sec | Use Variable |
| Accel | 360000.0 | mm/sec² | Use Literal LREAL#360_000.0 |
| Decel | 360000.0 | mm/sec² | Use Literal LREAL#360_000.0 |



Positioning

# MC_MoveAbsolute

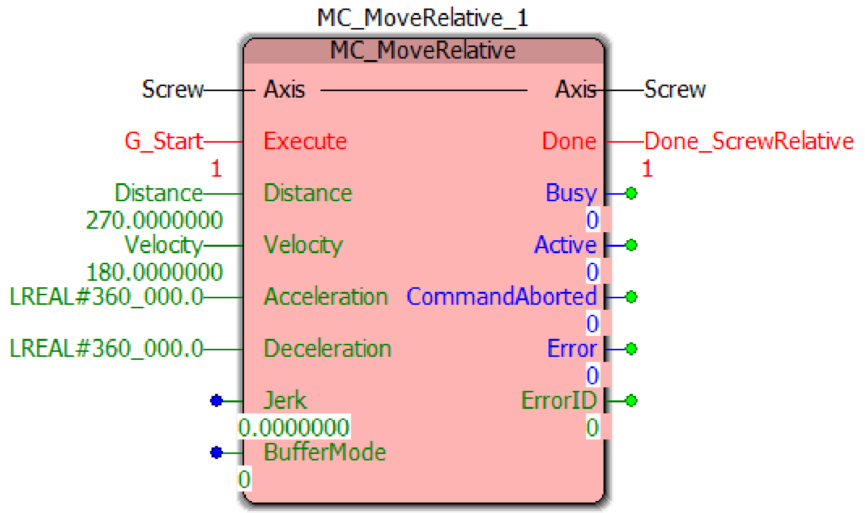
- *Partial Solution*



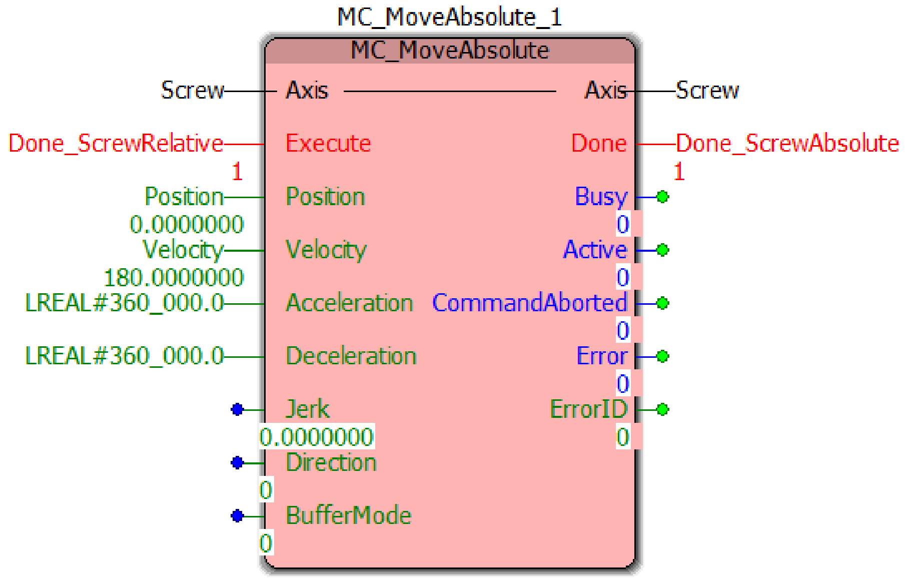
Jerk, Direction, BufferMode can be disconnected and default values are used.

•Jerk exists as a parameter in HWConfig “Moving Average Filter” #1300, #1301

•BufferMode and Direction are “Enumerated Data Types” (more information later)



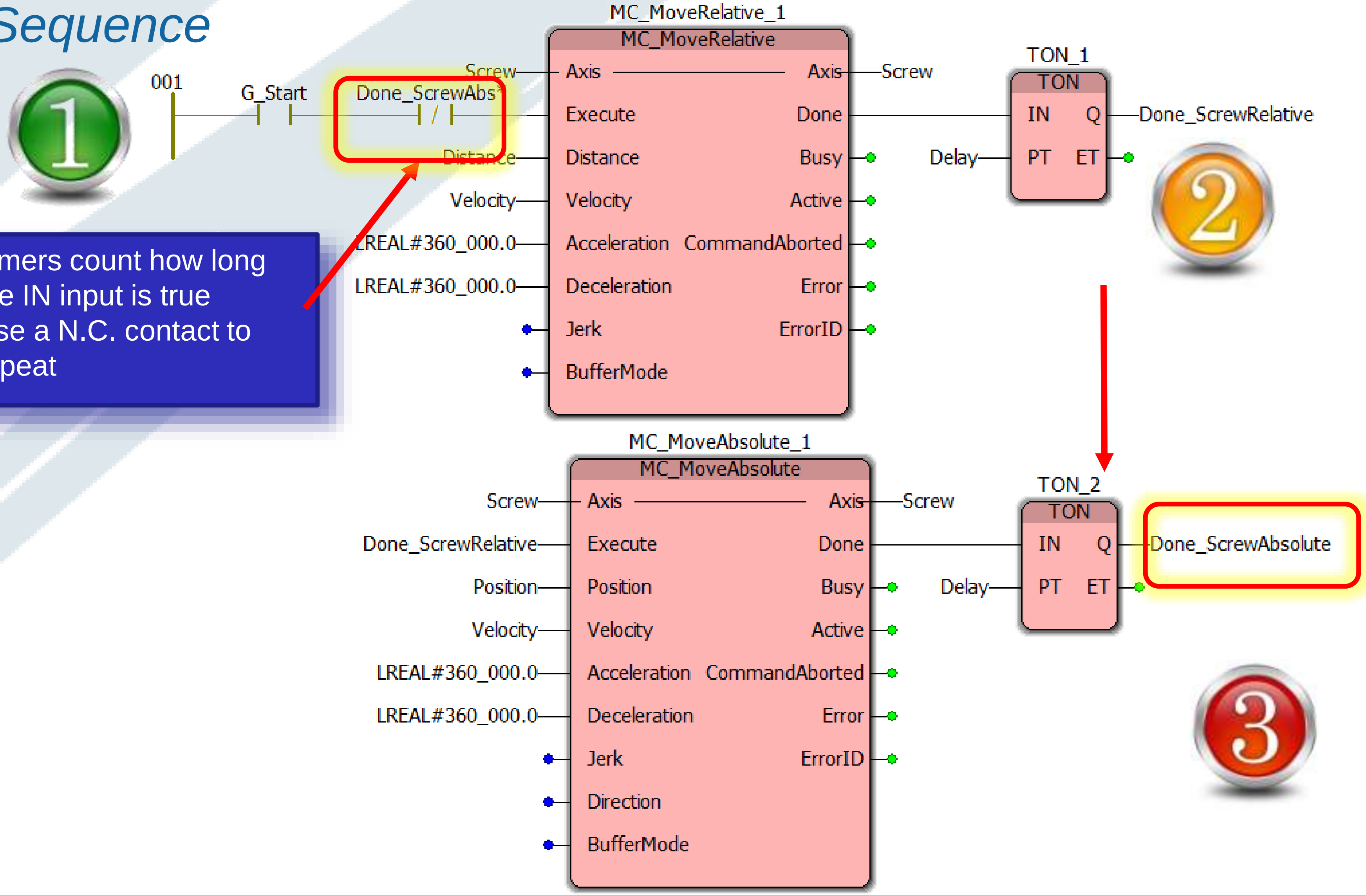
```
                         MC_MoveRelative_1
                          MC_MoveRelative
            Screw—— Axis                       Axis ——Screw
          G_Start—— Execute                    Done ——Done_ScrewRelative
                1                                 1
         Distance—— Distance                   Busy ——
      270.0000000                                 0
         Velocity—— Velocity                 Active ——
      180.0000000                                 0
LREAL#360_000.0—— Acceleration     CommandAborted ——
                                                  0
LREAL#360_000.0—— Deceleration                Error ——
                                                  0
                —— Jerk                     ErrorID ——
                  0.0000000                       0
                —— BufferMode
                  0
```



```
                         MC_MoveAbsolute_1
                          MC_MoveAbsolute
            Screw—— Axis                       Axis ——Screw
Done_ScrewRelative—— Execute                   Done ——Done_ScrewAbsolute
                1                                 1
         Position—— Position                   Busy ——
        0.0000000                                 0
         Velocity—— Velocity                 Active ——
      180.0000000                                 0
LREAL#360_000.0—— Acceleration     CommandAborted ——
                                                  0
LREAL#360_000.0—— Deceleration                Error ——
                                                  0
                —— Jerk                     ErrorID ——
                  0.0000000                       0
                —— Direction
                  0
                —— BufferMode
                  0
```

## Timers Between Moves

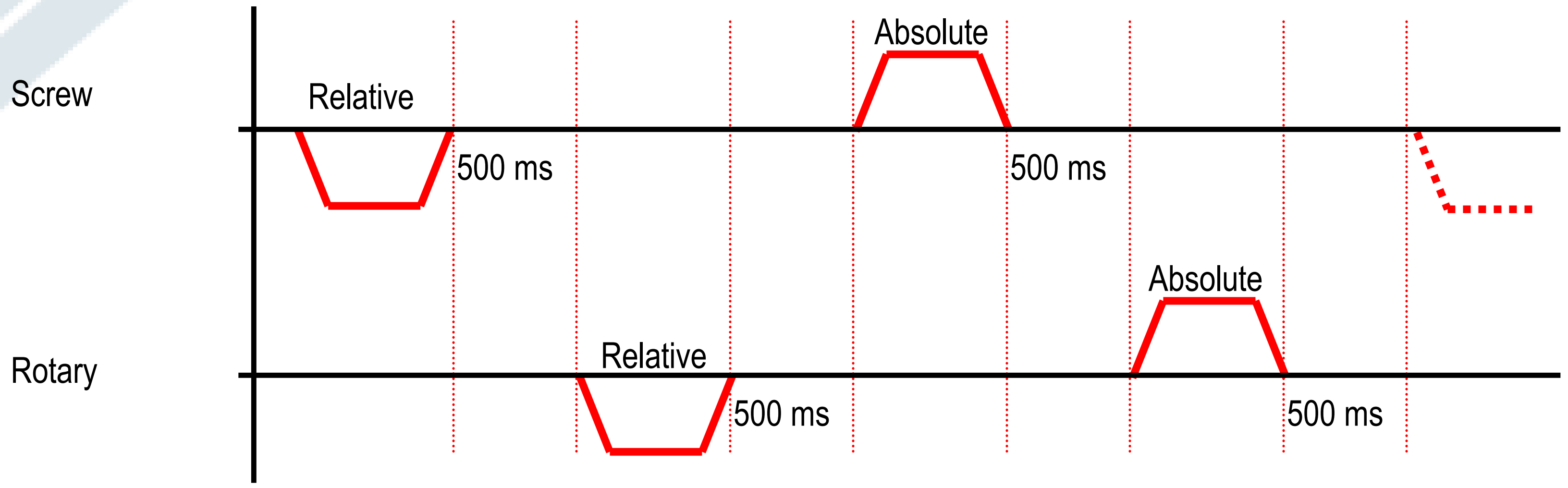
- *Repeating Sequence*



Timers count how long the IN input is true
Use a N.C. contact to repeat

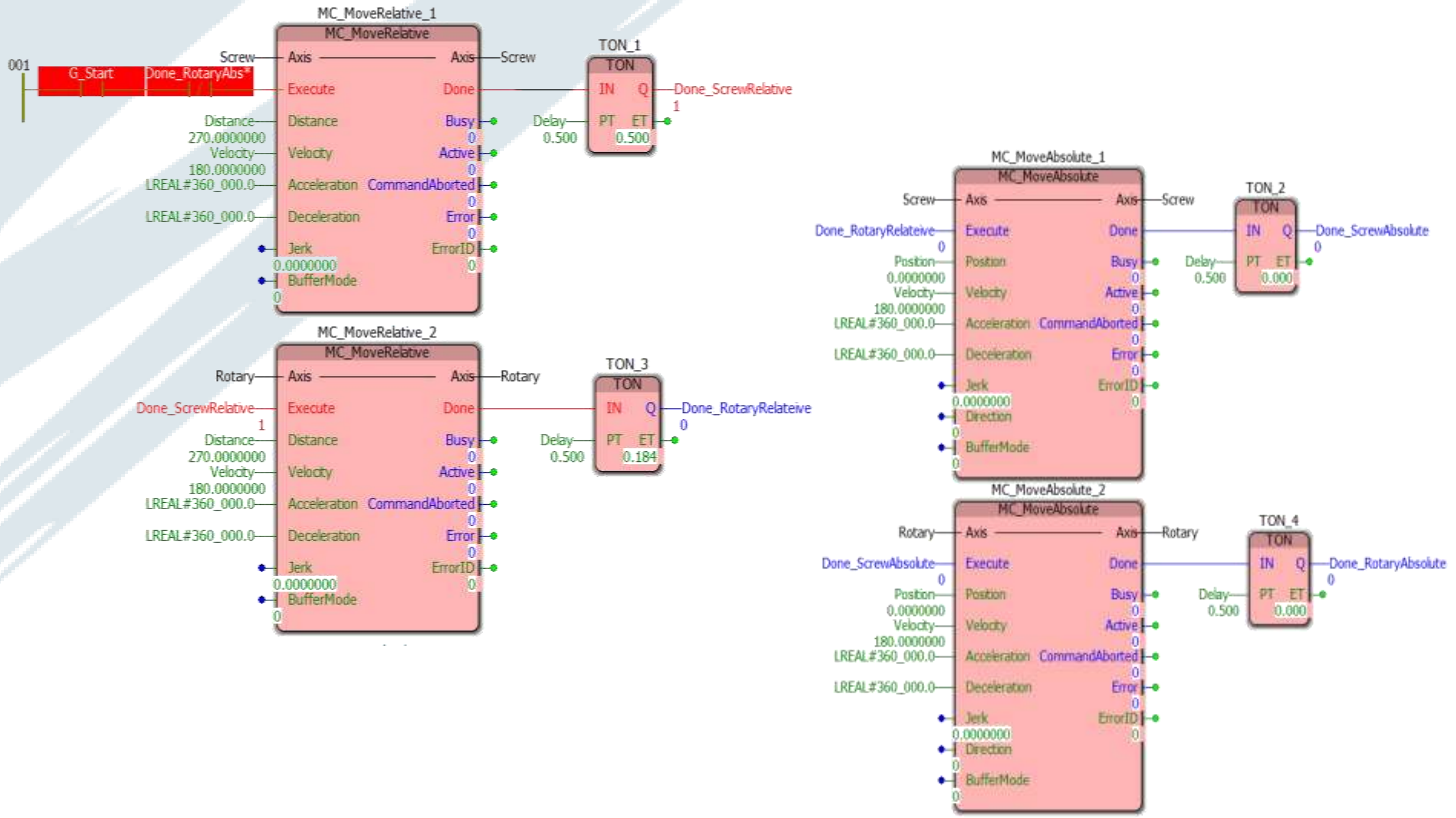
## Move Sequence

- *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*



# Move Sequence

- *Solution Concept*

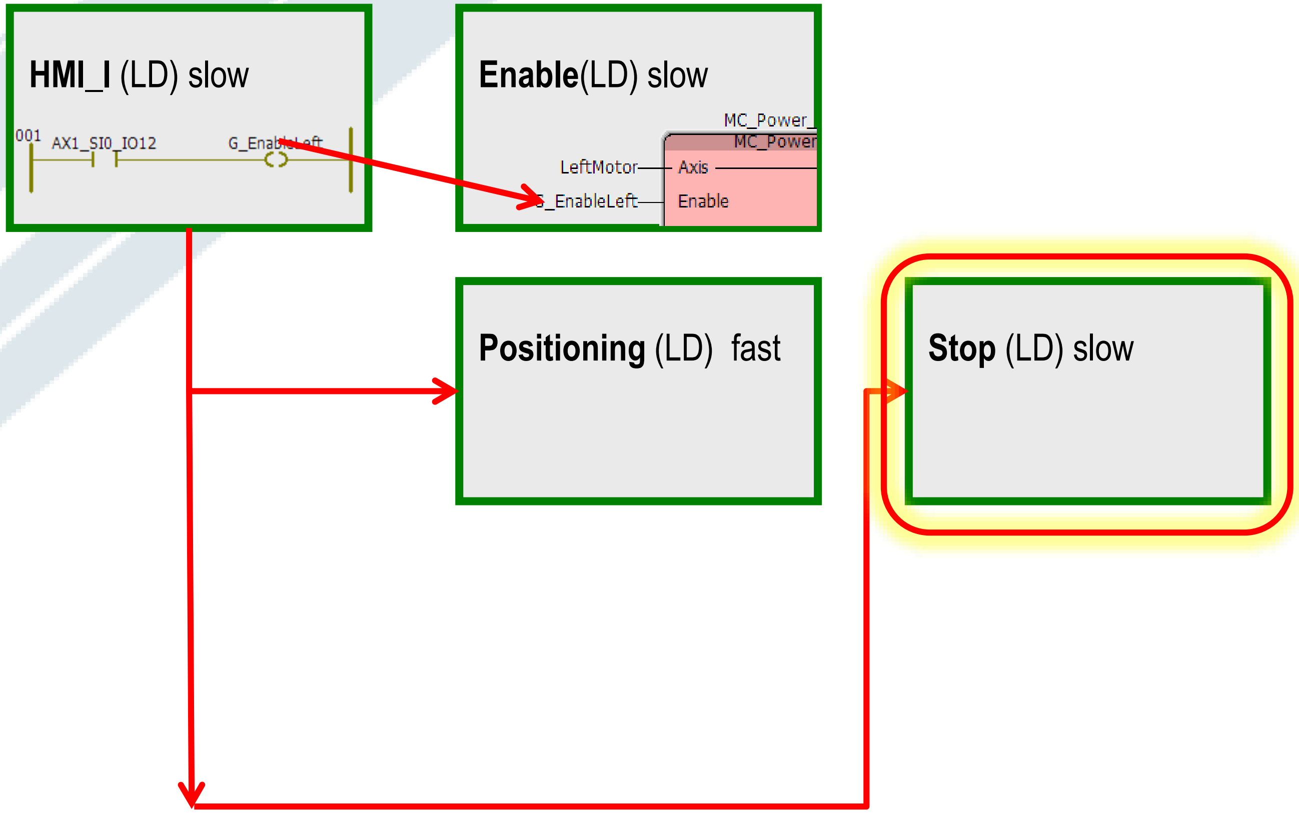


## Stop & Alarm

MC_Stop
MC_Reset
MC_ReadAxisError
Alarm Code Diagnosis
Task Execution Adjustment

## STOP Program

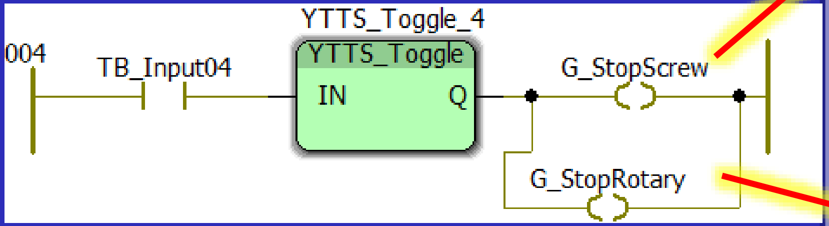
- *Create Program POU "Stop"*
  - *Instance: SlowTsk*



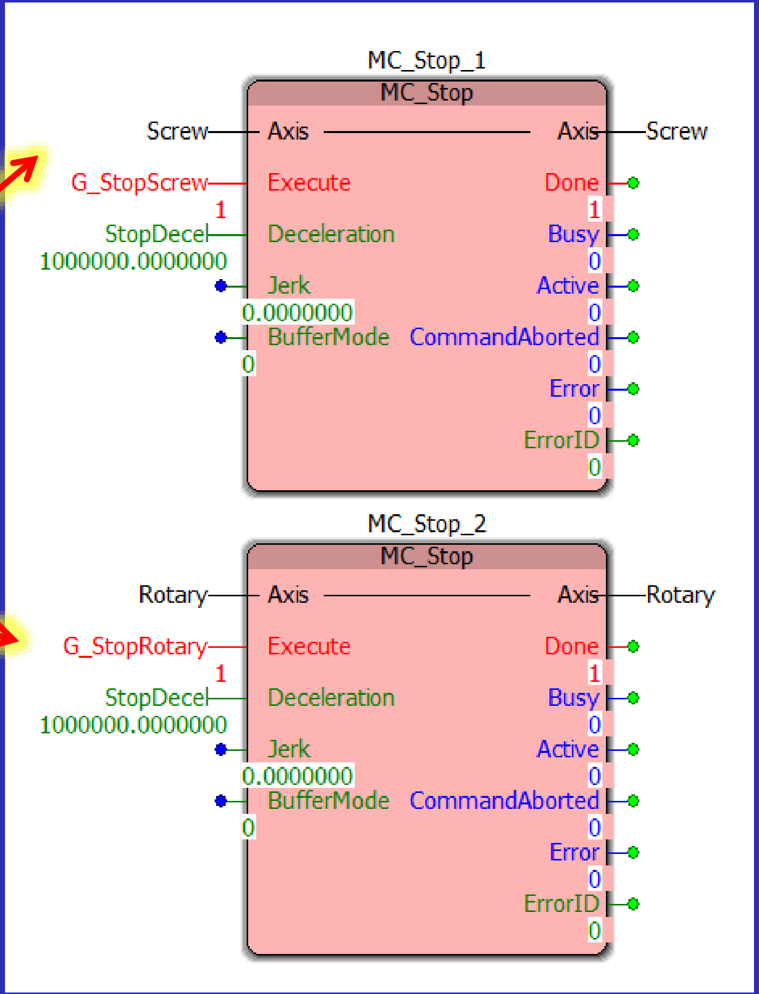
## MC_Stop

- *Partial Solution*

“HMI_I” Program POU



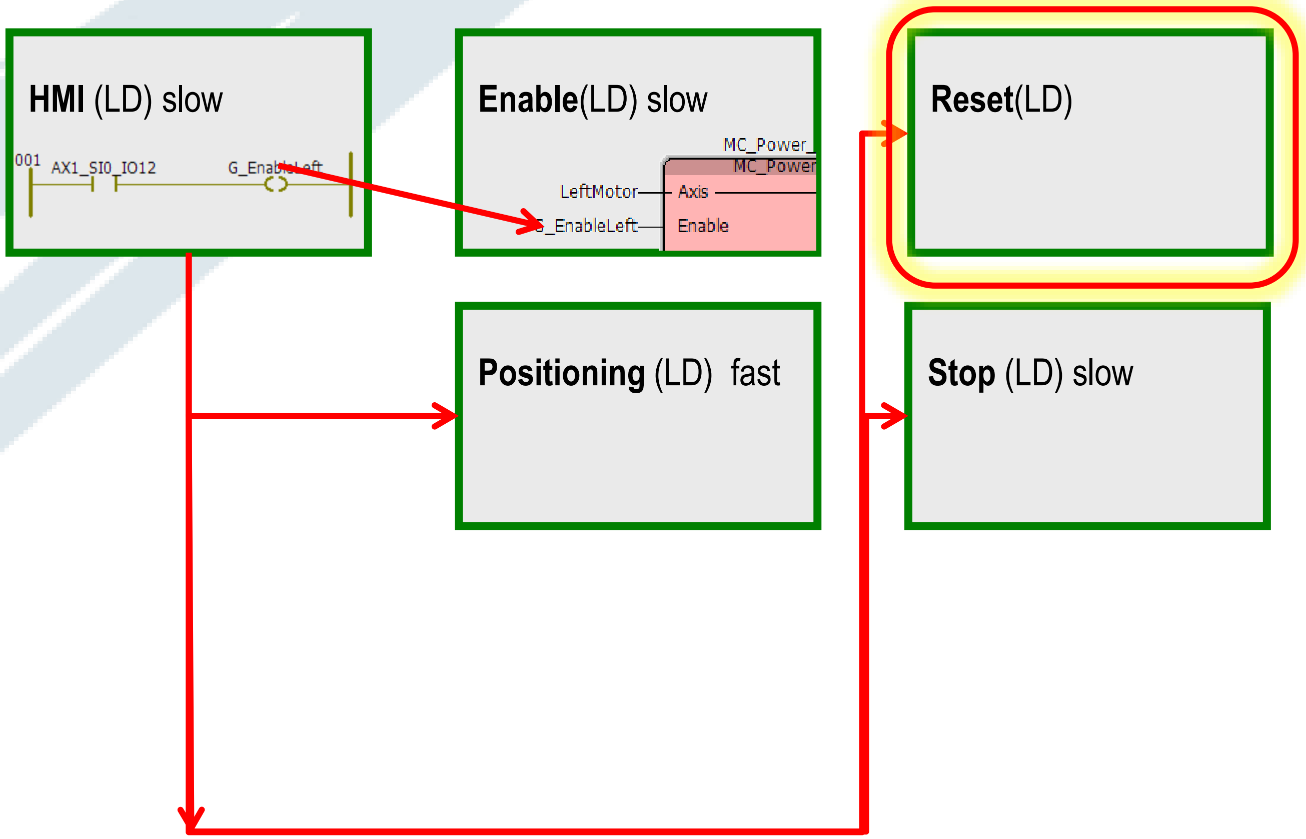
“Stop” Program POU



Stop & Alarm

## RESET Program

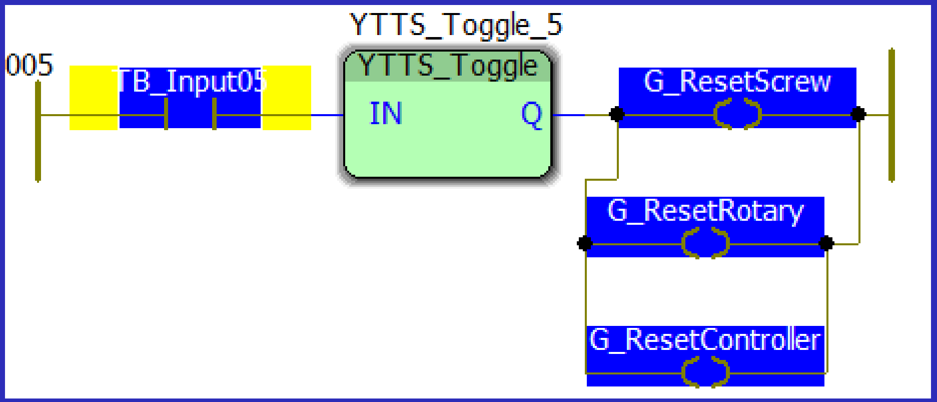
- *Create Program POU "Reset"*
  - *Instance: SlowTsk*

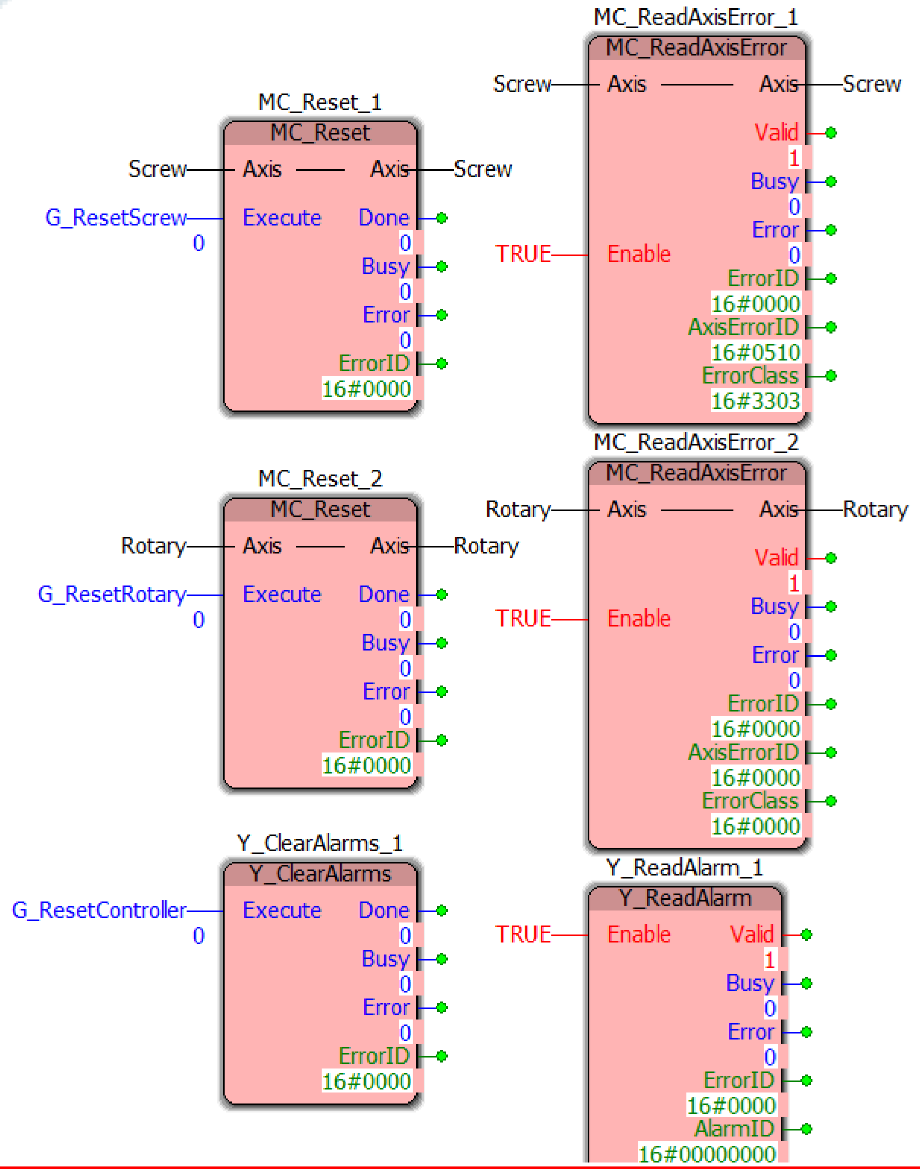


# MC_Reset, MC_ReadAxisError

- *Partial Solution*

"HMI_I" Program POU

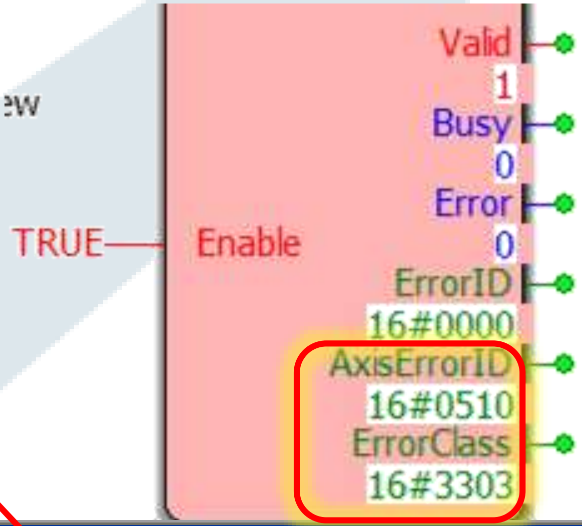




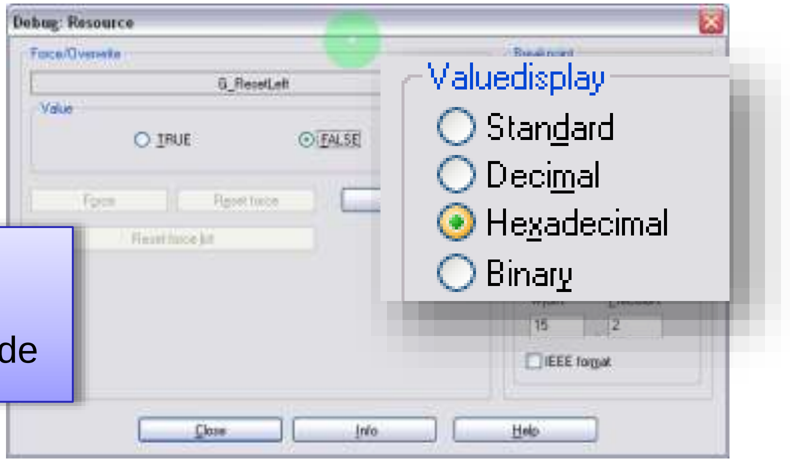
## Yaskawa Alarm Code

**Produce Alarm**
Speed = 40_000.0
Distance = 40_000.0

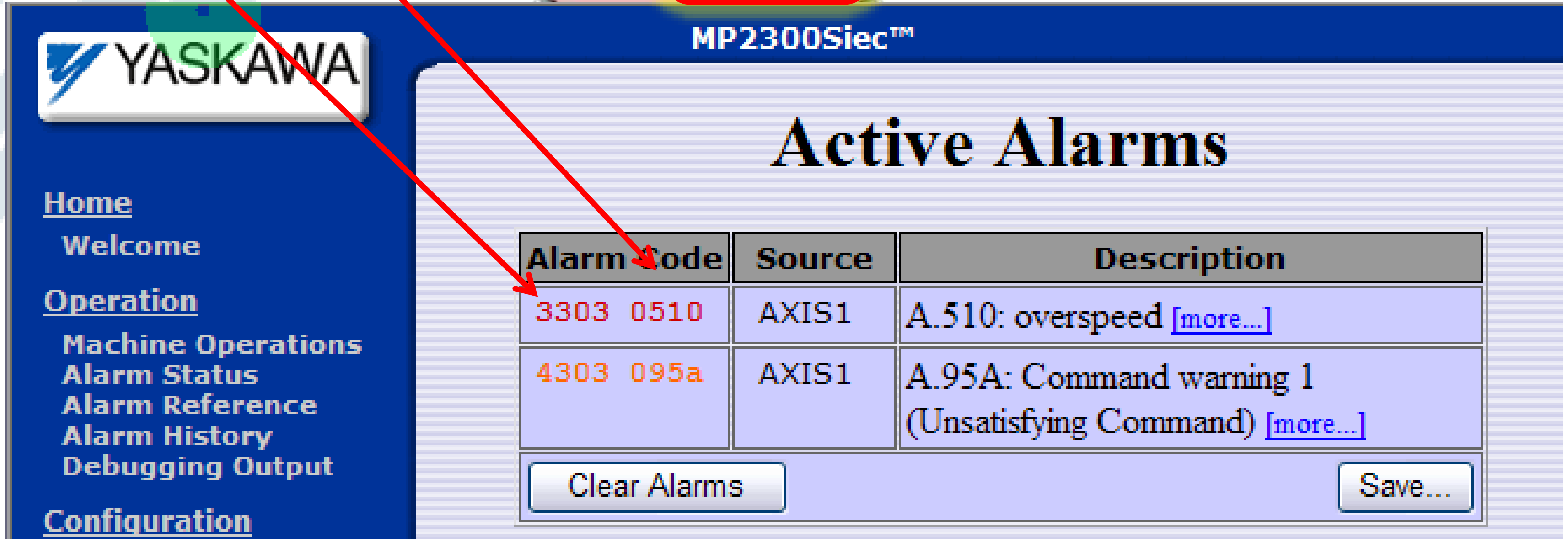
- *Axis Error ID (Hex)*
- *Error Class (Hex)*



Open any variable in DEBUG mode



Valuedisplay
Standard
Decimal
[✓ Hexadecimal]
Binary



MP2300Siec™

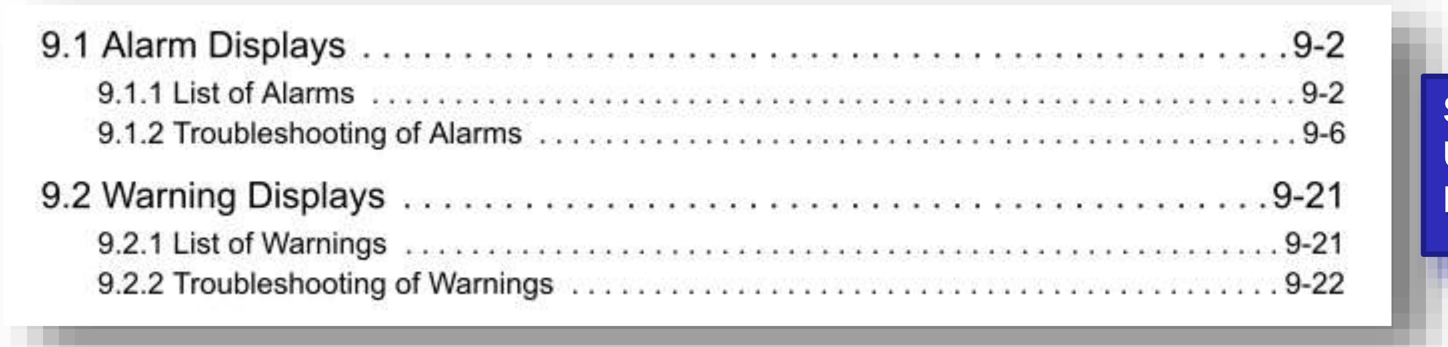
### Active Alarms

| Alarm Code | Source | Description |
|---|---|---|
| 3303 0510 | AXIS1 | A.510: overspeed [more...] |
| 4303 095a | AXIS1 | A.95A: Command warning 1 (Unsatisfying Command) [more...] |

## Yaskawa Alarm Diagnosis

- *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*

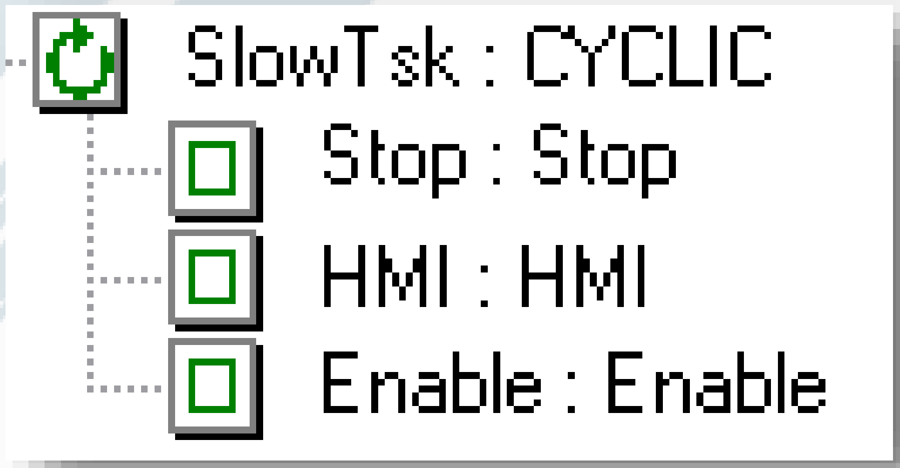
- *Warning*
  - *Future alarm under current conditions*
  - *Servo remains enabled*
  - *Display Code A. 9 □ □*
  - *Examples*
    - » *A.900 Position Error Overflow*
    - » *A.910 Overload*
    - » *A.95A Command Warning*
    - » *A.971 Undervoltage*



**Servo User Manual**

## Task Execution Adjustment

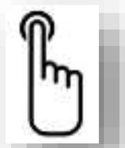
- *Adjust POU order in task, top to bottom*
  - *Logical sequence*
- *I/O Module Task Assignment*
  - *Assign to same task as application code that uses the %I %Q*
  - *Use Hardware Configuration*



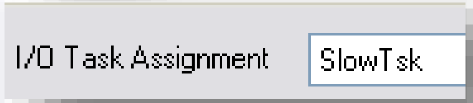
SlowTsk : CYCLIC
Stop : Stop
HMI : HMI
Enable : Enable

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

Human operation of physical inputs



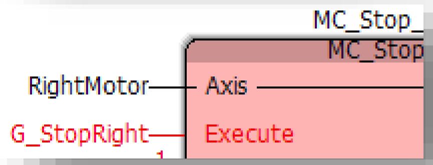
Inputs read by assigned task



I/O Task Assignment SlowTsk

Global variables written by HMI

Global variable executes MC_Stop



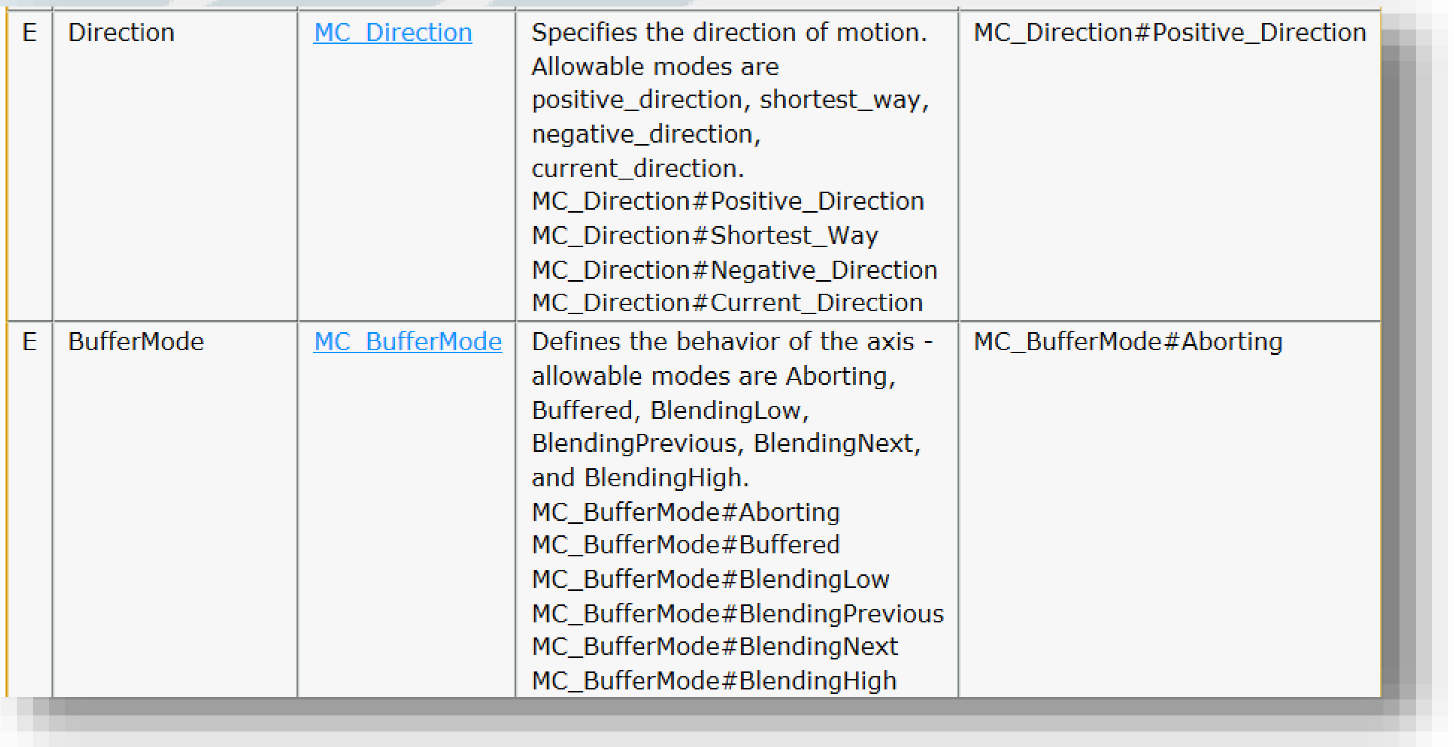
## Enumerated Data Types

Definition
Data Types Folder
MC_Direction
Enumerated Types as Literal and Variable
MC_Direction for Rotary Axis
MC_BufferMode to Create Blended Moves

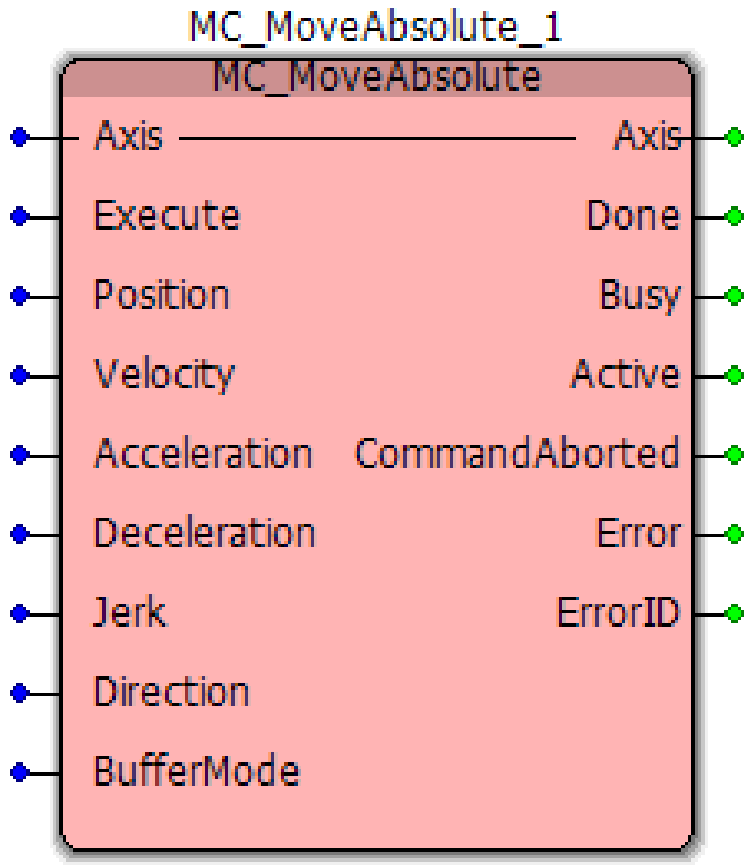
YASKAWA

# Enumerated Data Types

- *What is an enumerated data type?*
  - *A NAME for a NUMBER*
  - *Code reads easily*
  - *Reduced mistakes*



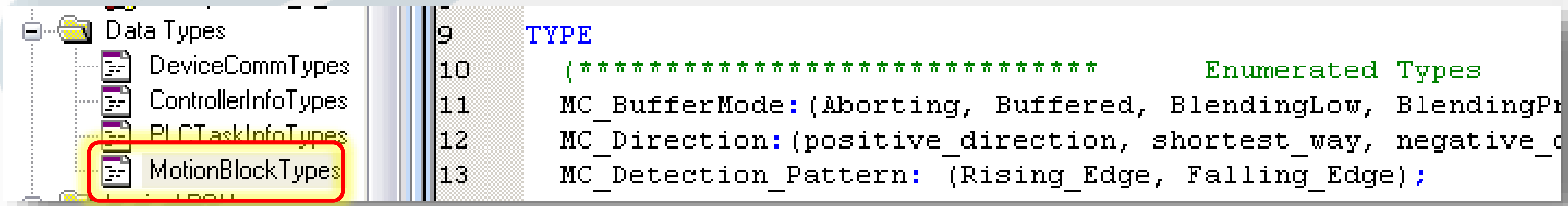
| | | | | |
|---|---|---|---|---|
| E | Direction | MC_Direction | Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction | MC_Direction#Positive_Direction |
| E | BufferMode | MC_BufferMode | Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh. MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh | MC_BufferMode#Aborting |



## Data Types Folder

*DataTypes Toolbox*

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



```
TYPE
  (******************************         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.

0|360

180

How to get to 180?

| MC_Direction# | | |
|---|---|---|
| 0 | positive_direction | In a rotary application, forces the axis to move in a positive direction. |
| 1 | shortest_way | For use in applications where the Load Type is configured as a rotary or modularized axis. |
| 2 | negative_direction | In a rotary application, forces the axis to move in a negative direction |
| 3 | 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 |

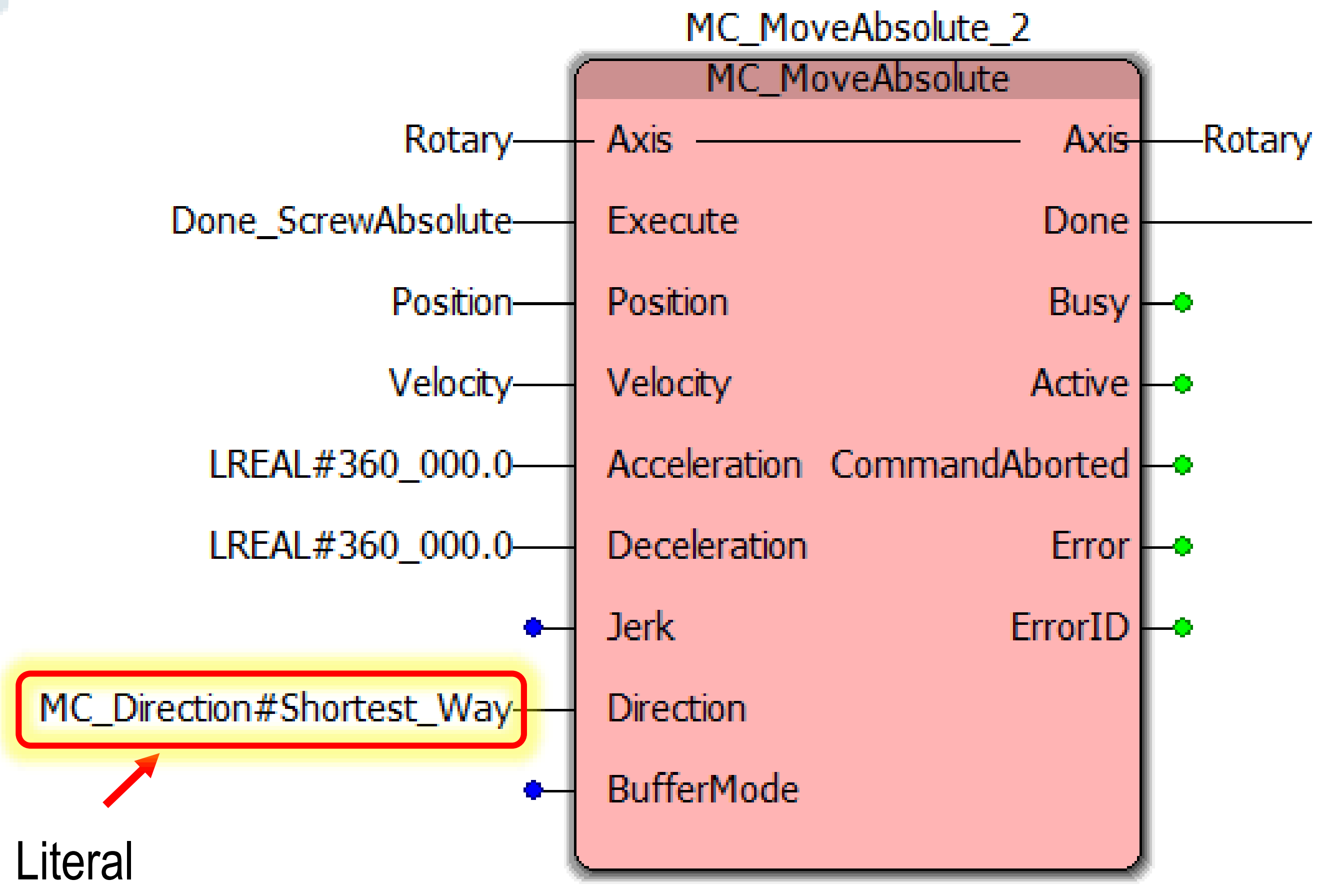
# Enumerated Data Types — As Literal

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

**Compare**:

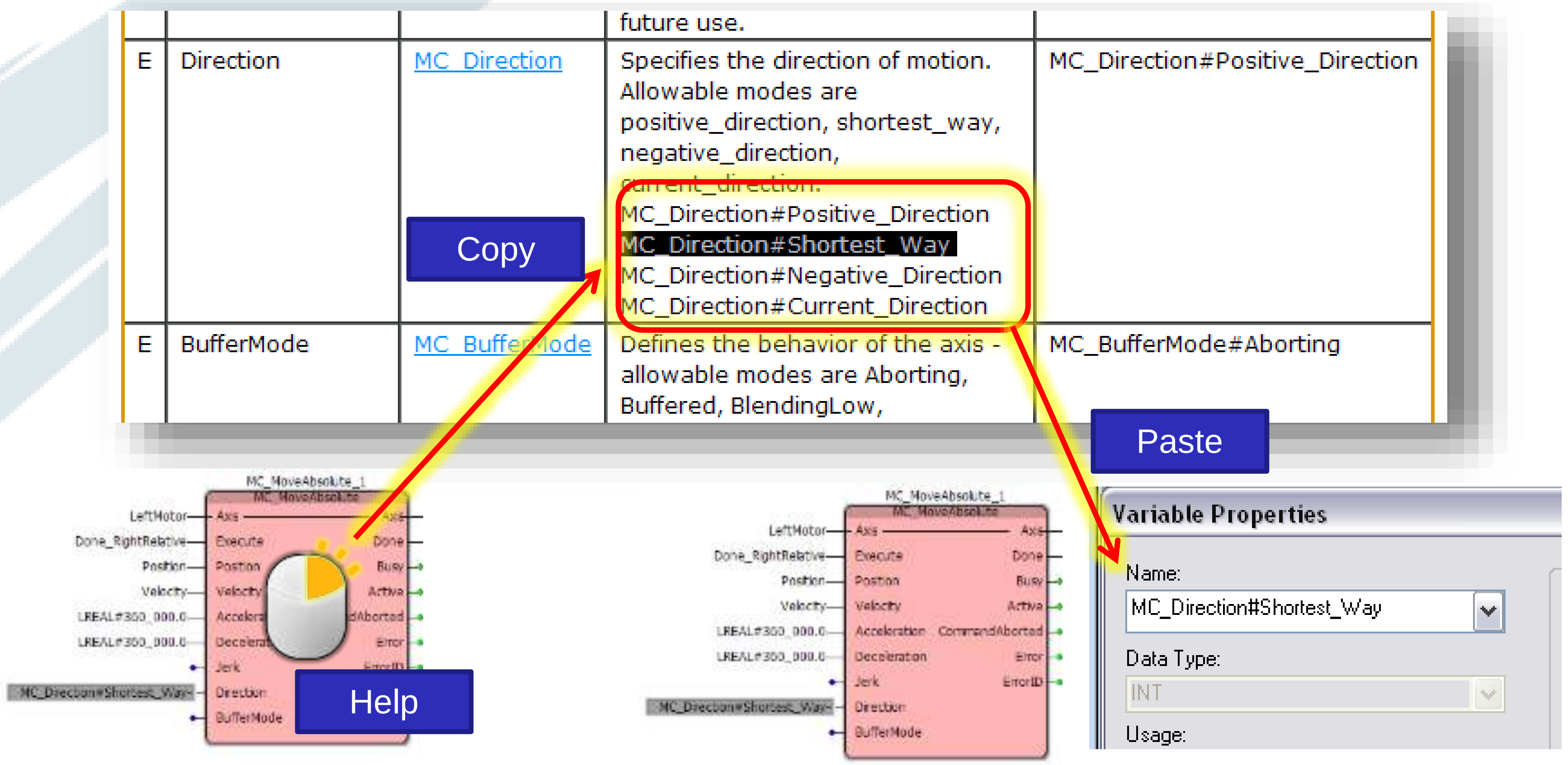
UINT#1
LREAL#1.0
MC_Direction#Shortest_Way

**Format**:
<DataType>#<data>



# Enumerated Data Types

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



| | | | | |
|---|---|---|---|---|
| | | | future use. | |
| E | Direction | MC_Direction | Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction | MC_Direction#Positive_Direction |
| E | BufferMode | MC_BufferMode | Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, | MC_BufferMode#Aborting |

Variable Properties

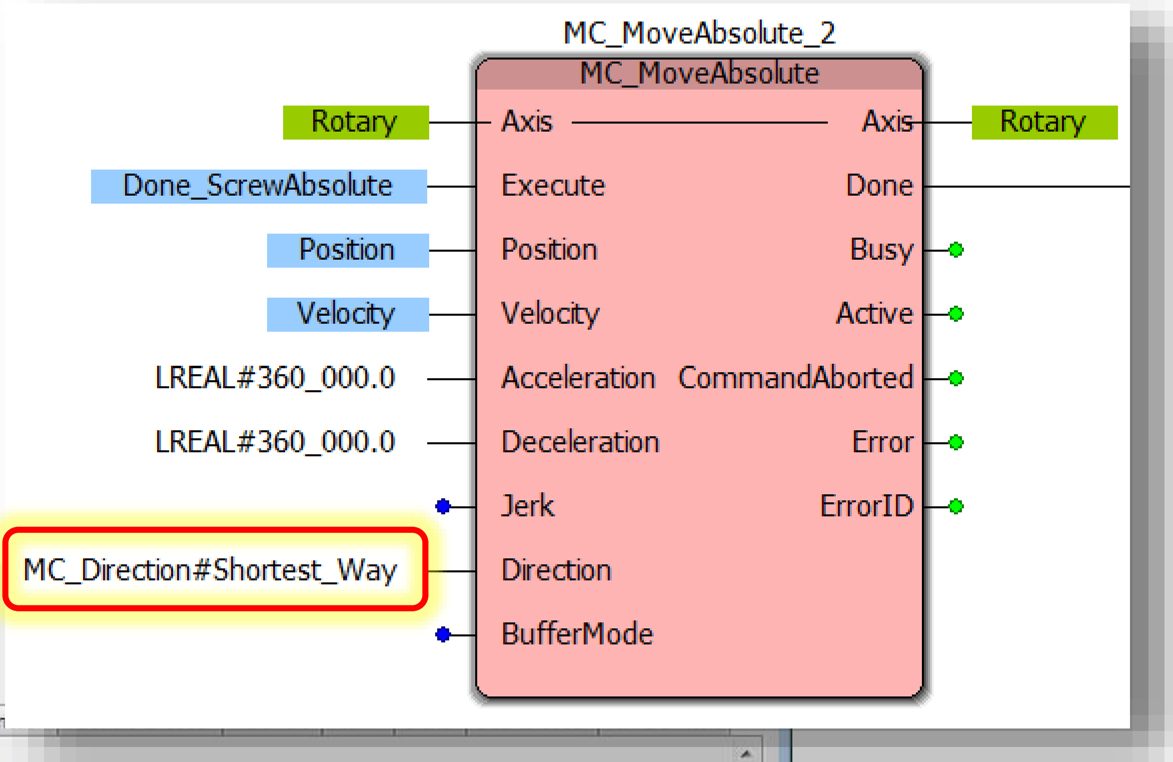
Name: MC_Direction#Shortest_Way

Data Type: INT

Usage:

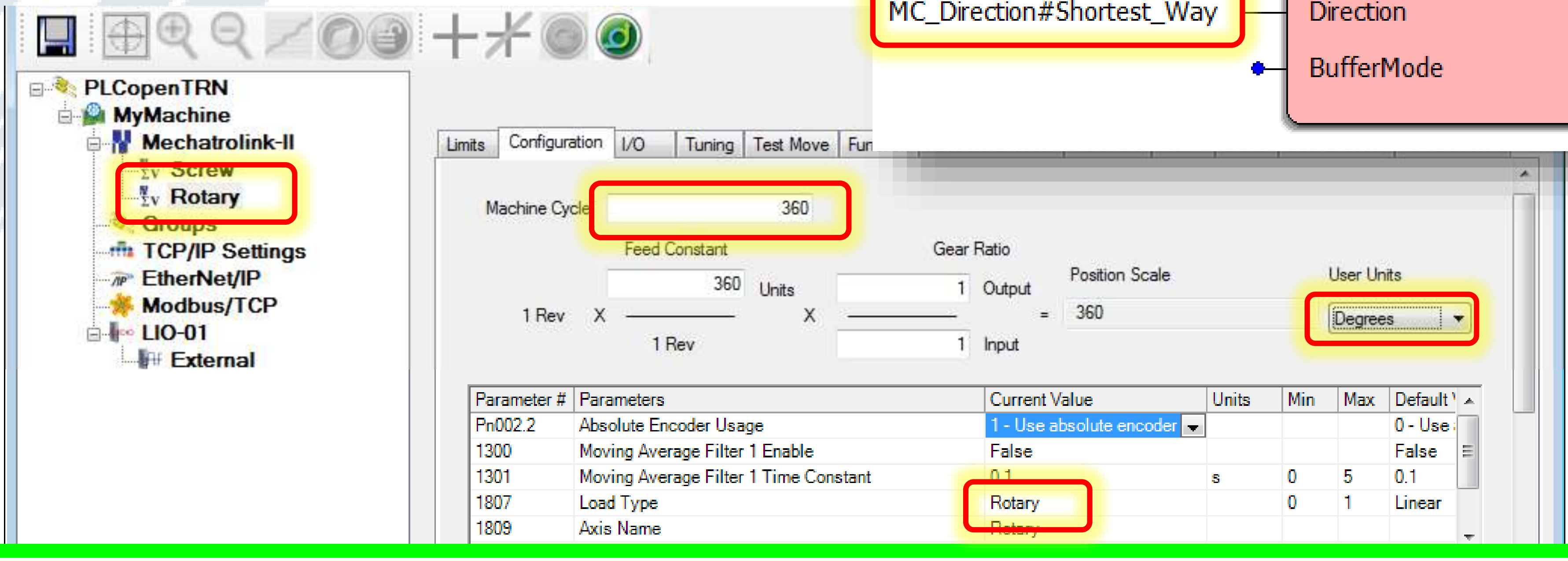
## Use MC_Direction in Program

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


MC_MoveAbsolute_2

MC_MoveAbsolute

| Input | Pin | Pin | Output |
|---|---|---|---|
| Rotary | Axis | Axis | Rotary |
| Done_ScrewAbsolute | Execute | Done | |
| Position | Position | Busy | |
| Velocity | Velocity | Active | |
| LREAL#360_000.0 | Acceleration | CommandAborted | |
| LREAL#360_000.0 | Deceleration | Error | |
| | Jerk | ErrorID | |
| MC_Direction#Shortest_Way | Direction | | |
| | BufferMode | | |



Machine Cycle 360

Feed Constant: 1 Rev X 360 Units / 1 Rev

Gear Ratio: X 1 Output / 1 Input

Position Scale = 360

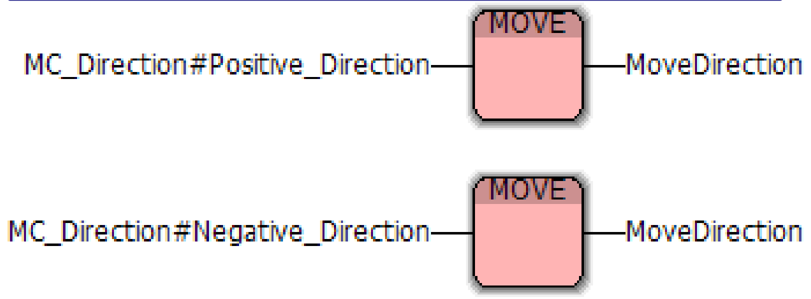
User Units: Degrees

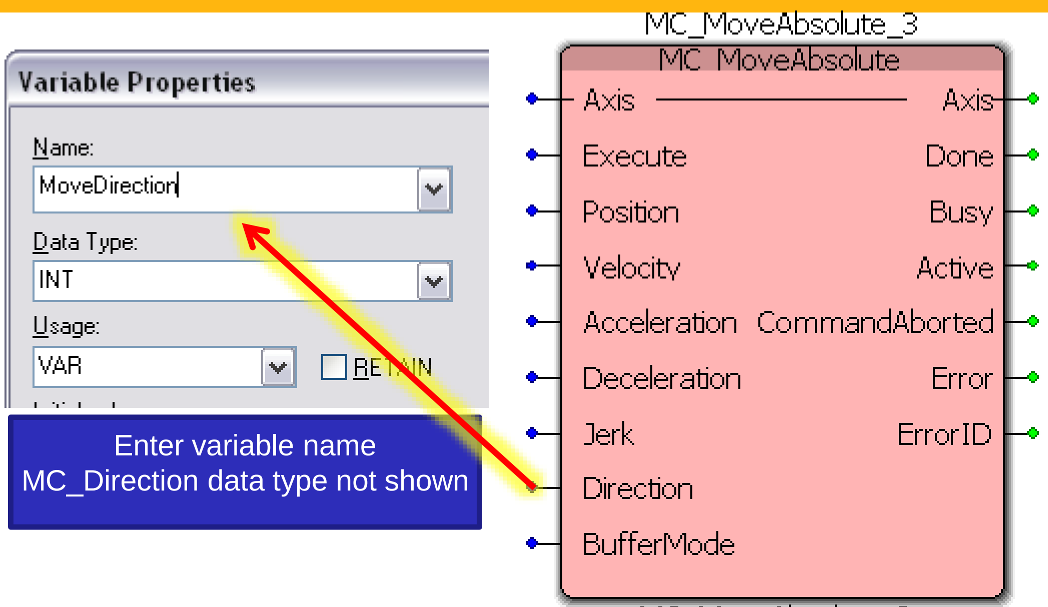
| Parameter # | Parameters | Current Value | Units | Min | Max | Default V |
|---|---|---|---|---|---|---|
| Pn002.2 | Absolute Encoder Usage | 1 - Use absolute encoder | | | | 0 - Use a |
| 1300 | Moving Average Filter 1 Enable | False | | | | False |
| 1301 | Moving Average Filter 1 Time Constant | 0.1 | s | 0 | 5 | 0.1 |
| 1807 | Load Type | Rotary | | 0 | 1 | Linear |
| 1809 | Axis Name | Rotary | | | | |

# Enumerated Data Types

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

Value loaded to variable within application code


MC_Direction#Positive_Direction
MOVE
MoveDirection
MC_Direction#Negative_Direction
MOVE
MoveDirection


Variable Properties

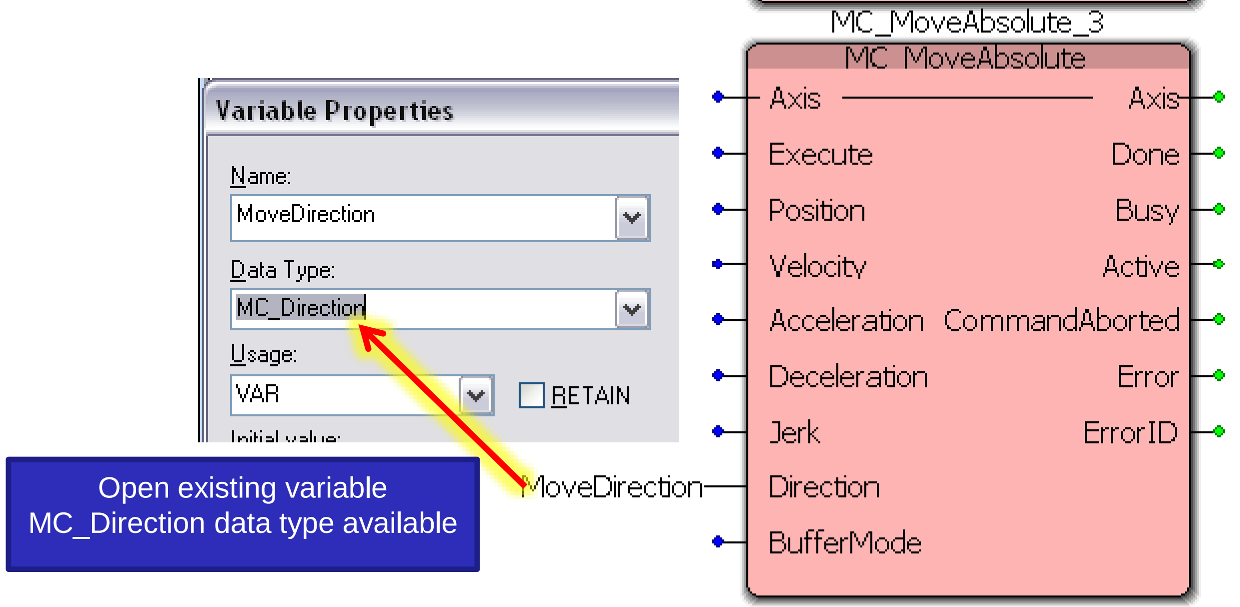
Name: MoveDirection

Data Type: INT

Usage: VAR

☐ RETAIN

Enter variable name
MC_Direction data type not shown


Variable Properties

Name: MoveDirection

Data Type: MC_Direction

Usage: VAR

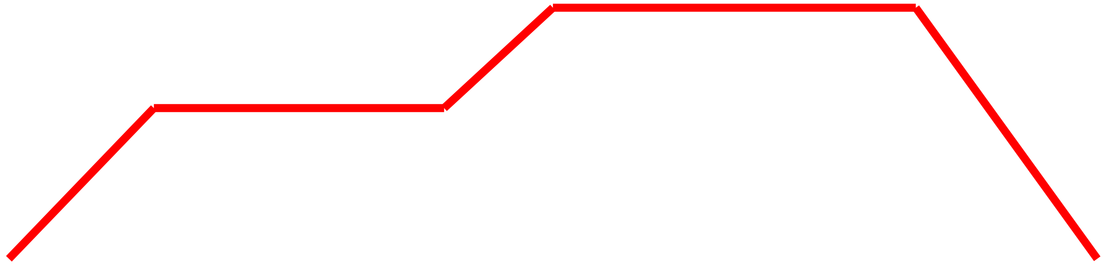
☐ RETAIN

Initial value:

Open existing variable
MC_Direction data type available

# Enumerated Data Types MC_BufferMode

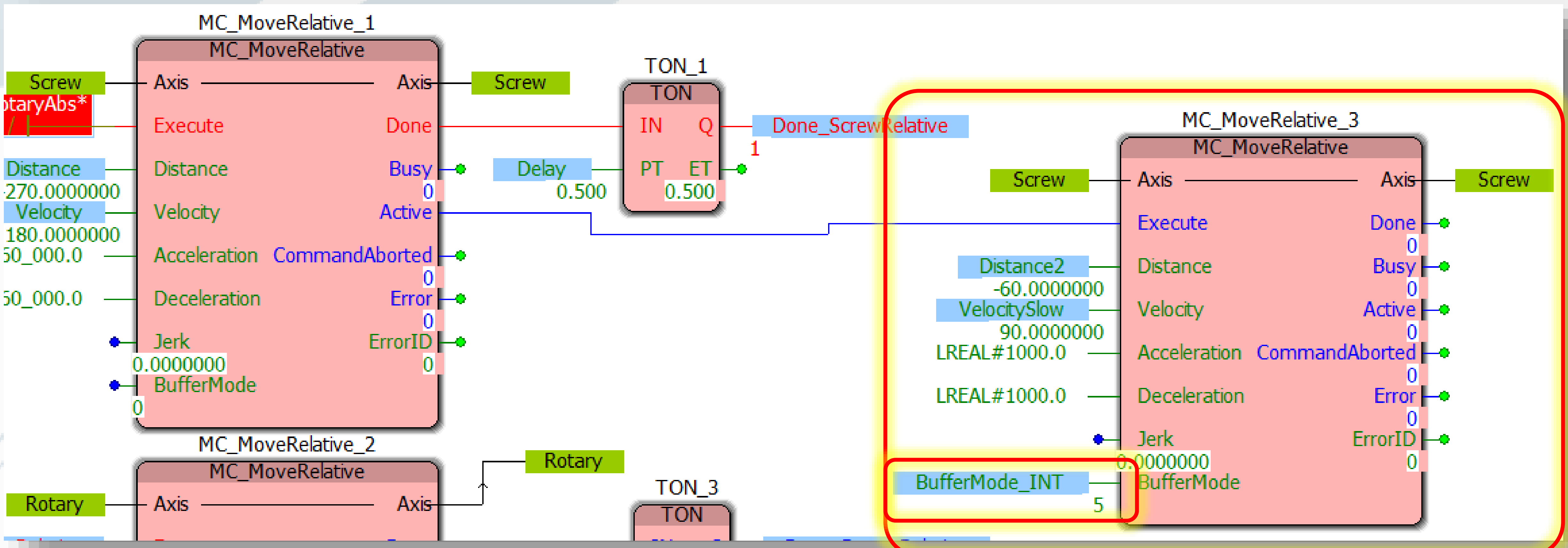
- *MC_BufferMode*
  - *Move 2 waits for Move 1 to complete*
  - *Create “blended moves”*
  - *Use for registration applications*



| Buffer mode | Short description 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! | Input value at BufferMode * |
|---|---|---|
| Aborting | This is the Default mode. The FB aborts an ongoing motion and the command affects the axis immediately. | INT#0 |
| Buffered | The FB affects the axis as soon as the previous movement is complete. The axis will stop between the movements. | INT#1 |
| BlendingLow | 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. | INT#2 |
| BlendingPrevious | The FB controls the axis after the previous FB has finished (equivalent to buffered), but the axis wi exll not stop between the movements. Blending with the velocity of the previous move. | INT#3 |
| BlendingNext | 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. | INT#4 |
| BlendingHigh | 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. | INT#5 |

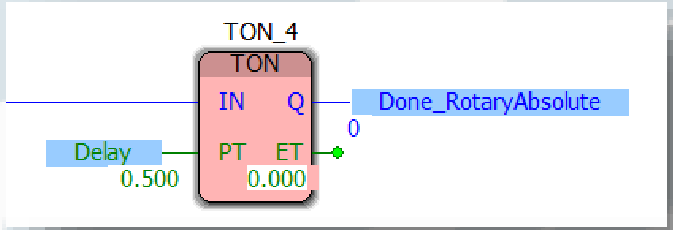
# Blended Move

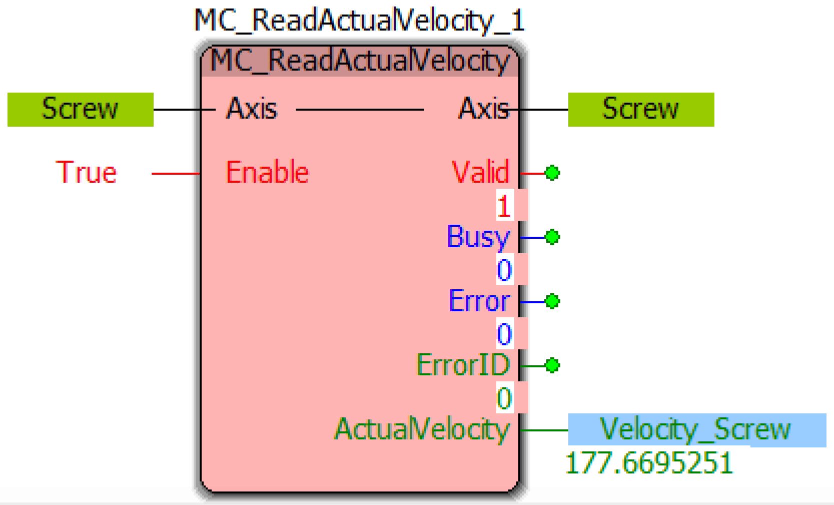
- *Edit Positioning POU*

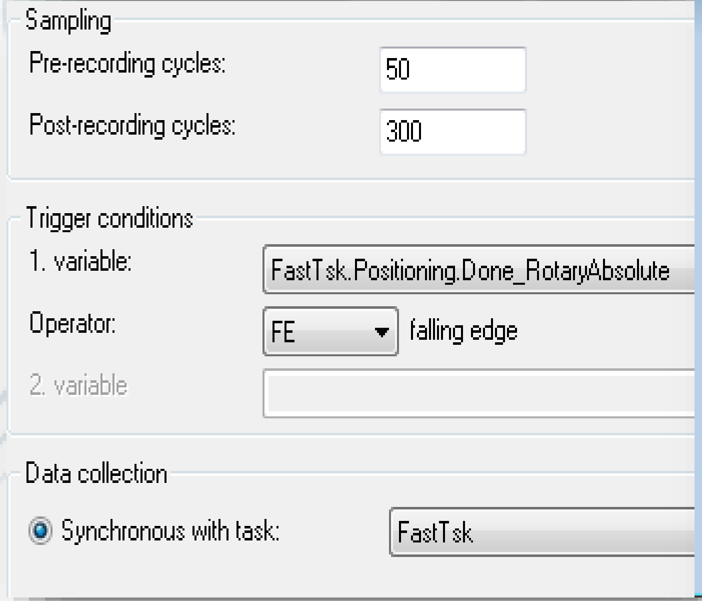


## Logic Analyzer

- *Use Logic Analyzer*
  - *Try different buffer modes*







Sampling

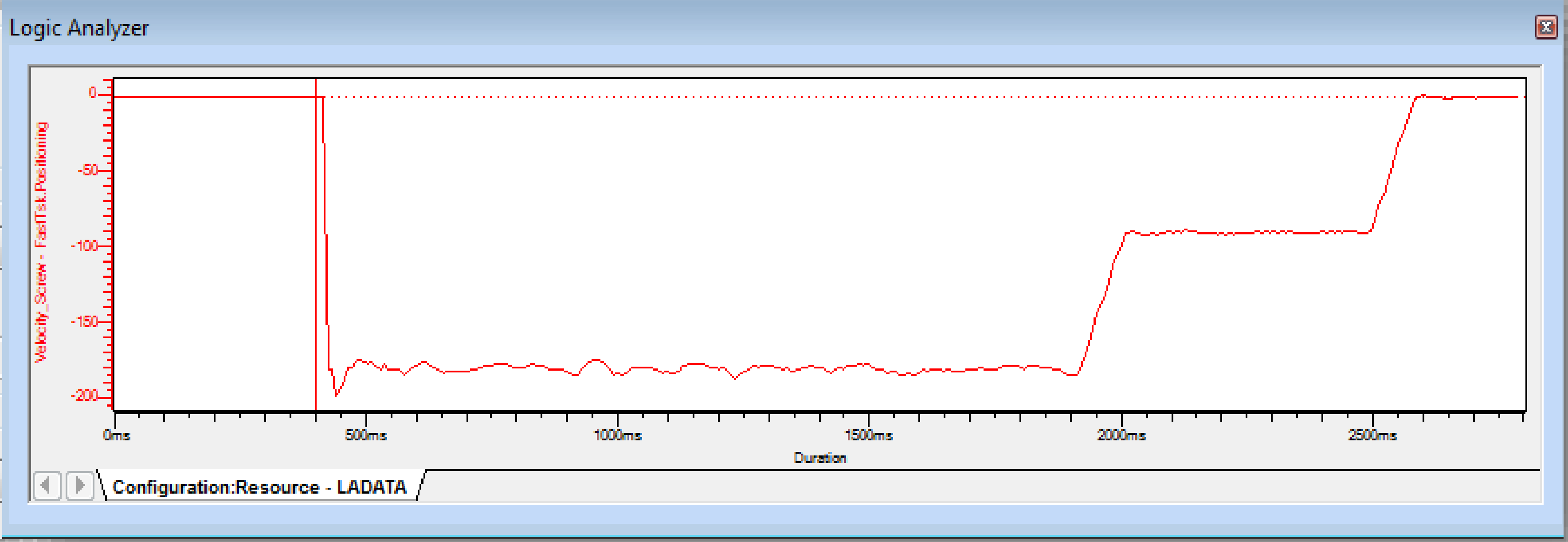
| | |
|---|---|
| Pre-recording cycles: | 50 |
| Post-recording cycles: | 300 |

Trigger conditions

| | |
|---|---|
| 1. variable: | FastTsk.Positioning.Done_RotaryAbsolute |
| Operator: | FE falling edge |
| 2. variable | |

Data collection

(●) Synchronous with task: FastTsk

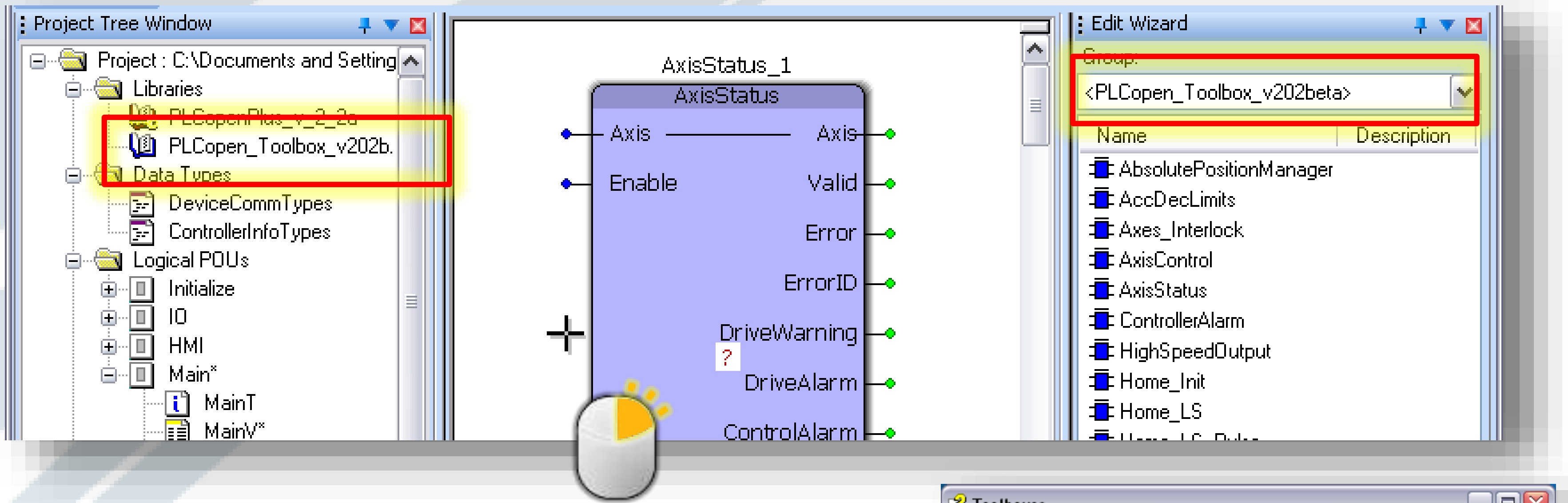


## Yaskawa "Toolbox" User Libraries

- Toolbox Concept
- PLCopen Toolbox
- Toolbox Installer
- Dependent Libraries
- Insert Additional Toolbox

# Yaskawa "Toolbox" User Libraries

PLCopen Toolbox



Project Tree Window

Project : C:\Documents and Setting
Libraries
PLCopenPlus_v_2_2a
PLCopen_Toolbox_v202b.
Data Types
DeviceCommTypes
ControllerInfoTypes
Logical POUs
Initialize
IO
HMI
Main*
MainT
MainV*

AxisStatus_1

| AxisStatus | |
|---|---|
| Axis | Axis |
| Enable | Valid |
| | Error |
| | ErrorID |
| | DriveWarning |
| | DriveAlarm |
| | ControlAlarm |

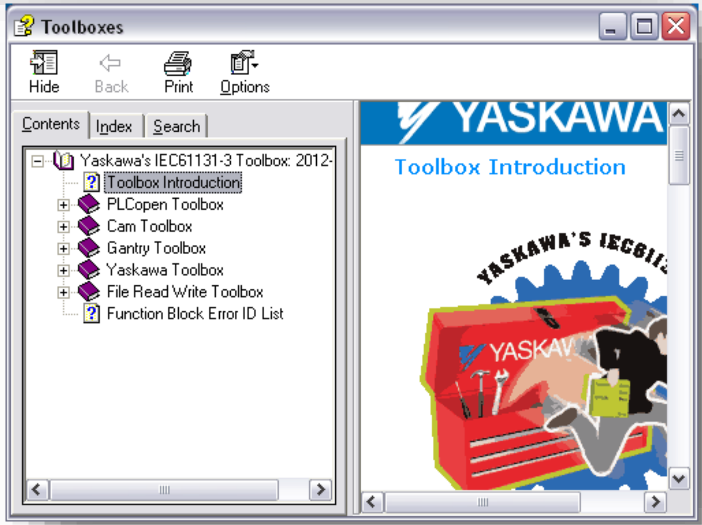
Edit Wizard

Group:
<PLCopen_Toolbox_v202beta>

| Name | Description |
|---|---|
| AbsolutePositionManager | |
| AccDecLimits | |
| Axes_Interlock | |
| AxisControl | |
| AxisStatus | |
| ControllerAlarm | |
| HighSpeedOutput | |
| Home_Init | |
| Home_LS | |

Help

- *Right Click Help*
  - *When Toolbox is installed*
  - *Links to eLearning Videos and recorded webinars*



Toolboxes

Hide Back Print Options

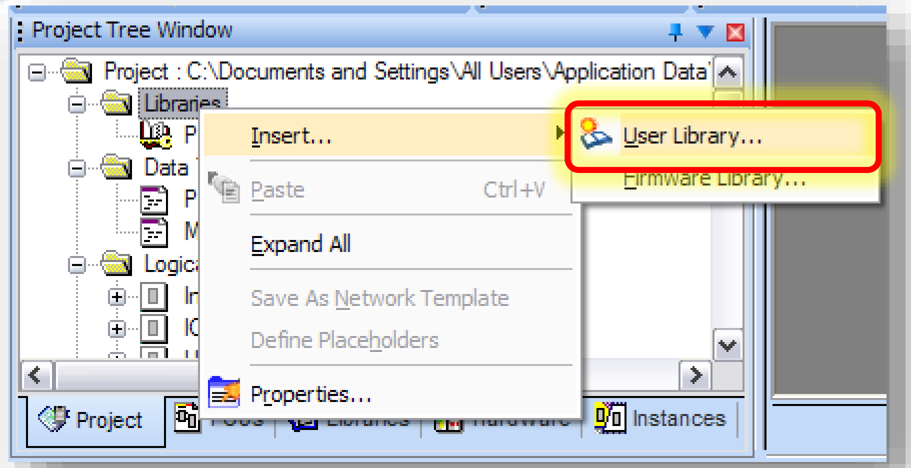
Contents Index Search

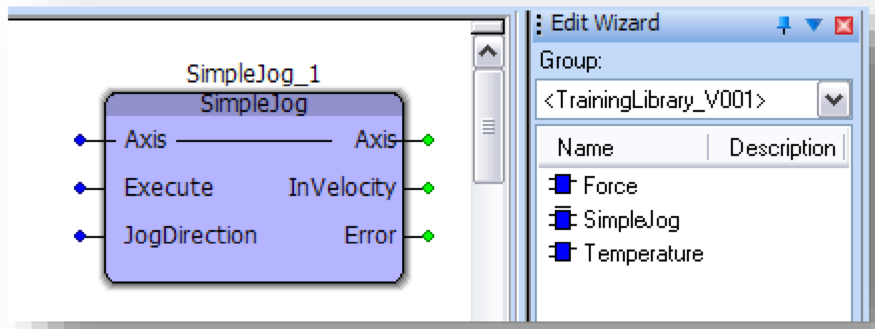
Yaskawa's IEC61131-3 Toolbox: 2012-
Toolbox Introduction
PLCopen Toolbox
Cam Toolbox
Gantry Toolbox
Yaskawa Toolbox
File Read Write Toolbox
Function Block Error ID List

YASKAWA
Toolbox Introduction

## Library Concept Review

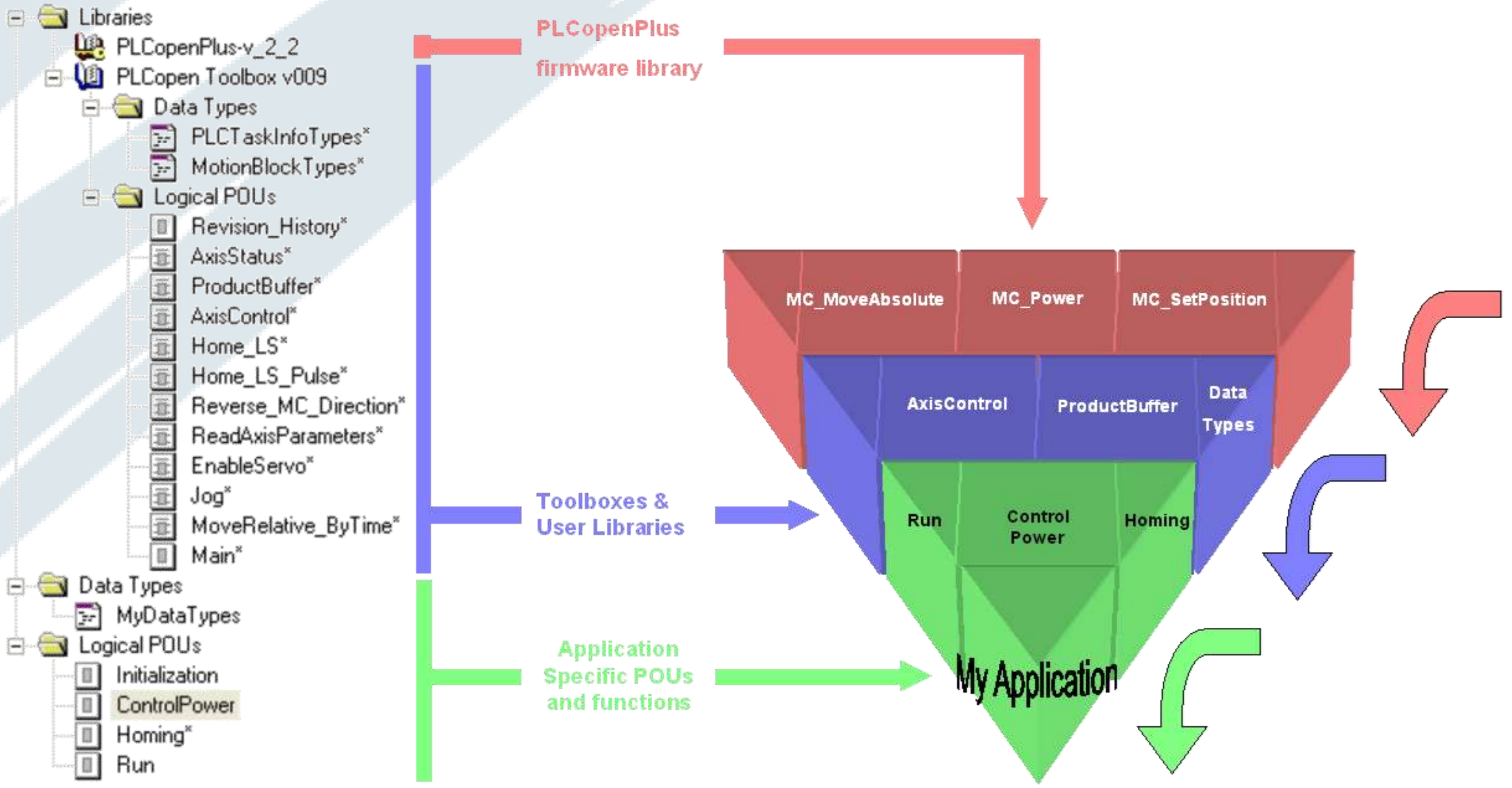
- *Insert Another Project*
  - *Library = any project*
  - **.mwt (or *.mwe)*
- *Library Data Imported*
  - *User FU & FB POUs*
  - *Program POUs*
  - *Data Types*
  - *NOT global variables!*
  - *NOT dependent libraries!*
- *Organization*
  - *Specific projects for library use*
  - *Revision number in project name*
  - *Prefix (ex: YTTS_)*





# Yaskawa "Toolbox" User Libraries

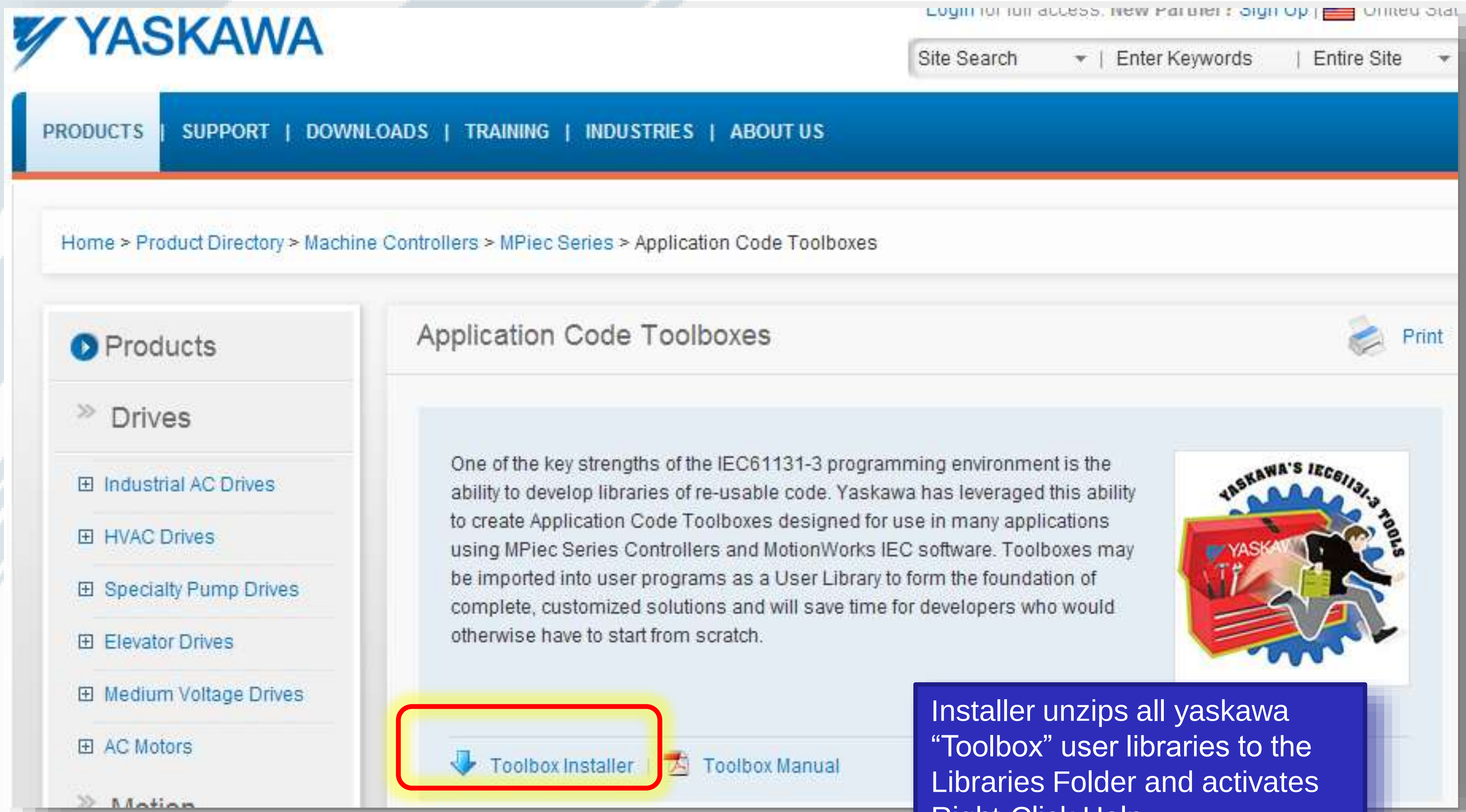
- *Yaskawa Tech Note: TN.MCD.08.130*



All Functions, Function Blocks, and DataTypes from the libraries are available for the application.

## Toolbox Installer

- *Yaskawa.com/iectb*



Login for full access. New Partner? Sign Up | United Stat

YASKAWA

Site Search | Enter Keywords | Entire Site

PRODUCTS | SUPPORT | DOWNLOADS | TRAINING | INDUSTRIES | ABOUT US

Home > Product Directory > Machine Controllers > MPiec Series > Application Code Toolboxes

Products

Drives

- Industrial AC Drives
- HVAC Drives
- Specialty Pump Drives
- Elevator Drives
- Medium Voltage Drives
- AC Motors

Motion

Application Code Toolboxes

Print

One of the key strengths of the IEC61131-3 programming environment is the ability to develop libraries of re-usable code. Yaskawa has leveraged this ability to create Application Code Toolboxes designed for use in many applications using MPiec Series Controllers and MotionWorks IEC software. Toolboxes may be imported into user programs as a User Library to form the foundation of complete, customized solutions and will save time for developers who would otherwise have to start from scratch.

YASKAWA'S IEC61131-3 TOOLS

Toolbox Installer | Toolbox Manual

Installer unzips all yaskawa "Toolbox" user libraries to the Libraries Folder and activates Right-Click Help

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

Refer to the Quick Reference Guide

YASKAWA

Quick Reference Guide

MPiec Series Controllers

Contents

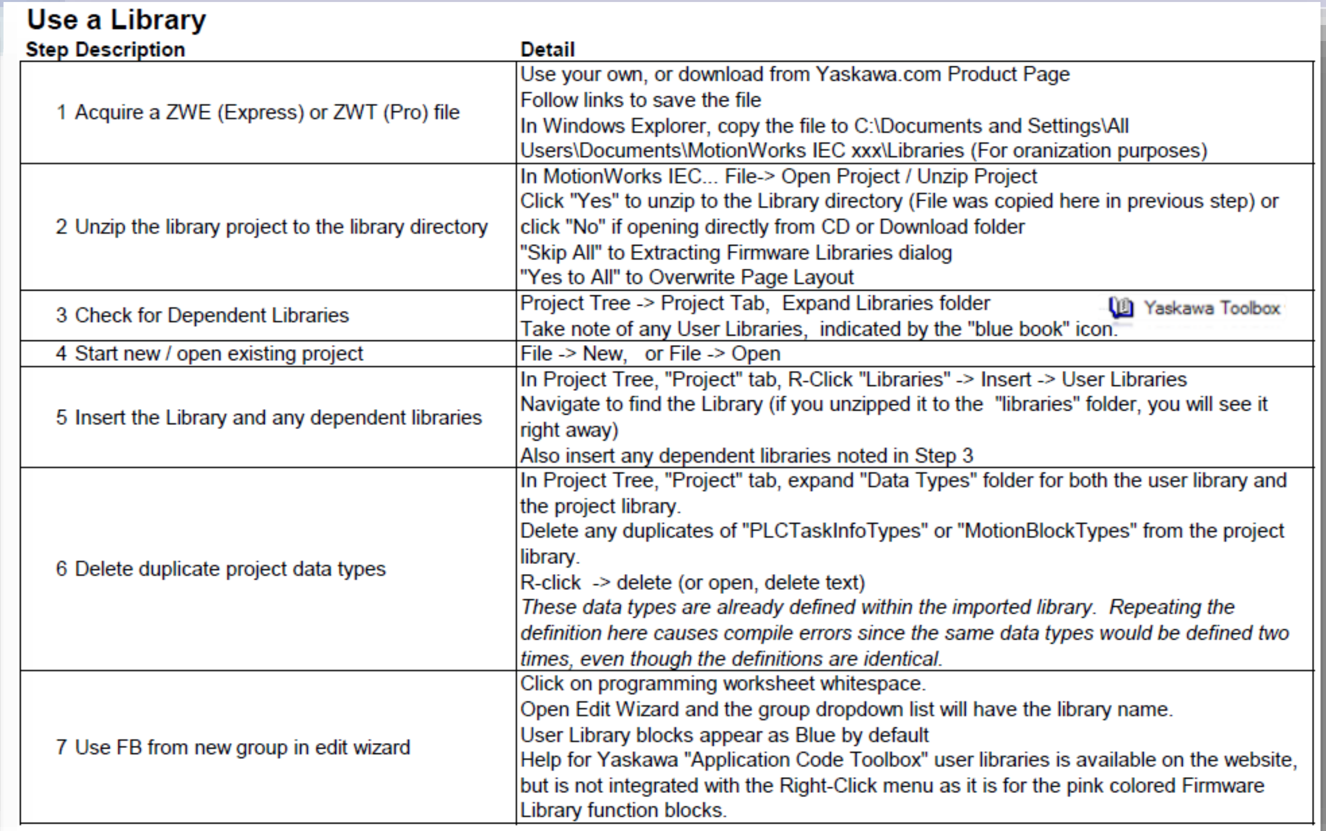
**Startup Procedures**
- Set the Front Panel Switches
- Open a saved project
- Start a new project
- Establish ethernet communication with controller (web interface)
- Go online with controller
- Download project to controller
- Set program to auto-start
- Reset an MP2300Siec system back to factory settings?
- Confirm motion is possible on each axis
- Tune the motors

**Maintenance Procedures**
- Update the firmware
- Determine/Set unknown IP address in controller
- Extract/Open the project from controller
- Replace the controller
- Replace the servopack
- Replace the servo motor

**Analysis & Troubleshooting**
- Monitor Program
- Watch Variables in a List (Monitor window)
- Use Logic Analyzer (AKA: graph, trace, scope)
- Diagnose Alarm Status

**Modbus/TCP and Ethernet/IP**
**User Libraries**
**OPC Server**

MotionWorks IEC 2
PRO
IEC

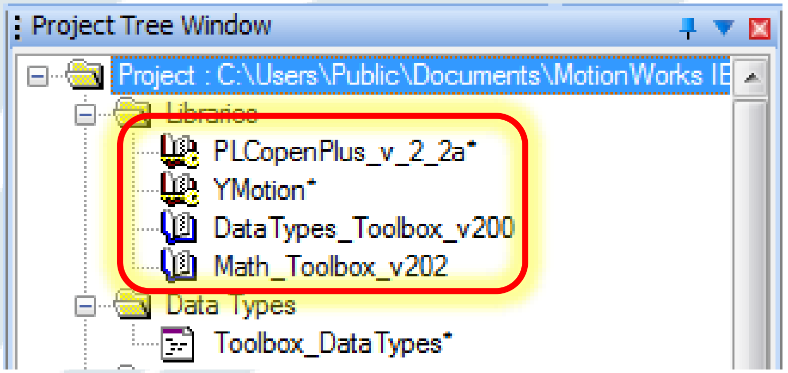


**Use a Library**

| Step Description | Detail |
|---|---|
| 1 Acquire a ZWE (Express) or ZWT (Pro) file | Use your own, or download from Yaskawa.com Product Page<br>Follow links to save the file<br>In Windows Explorer, copy the file to C:\Documents and Settings\All Users\Documents\MotionWorks IEC xxx\Libraries (For oranization purposes) |
| 2 Unzip the library project to the library directory | In MotionWorks IEC... File-> Open Project / Unzip Project<br>Click "Yes" to unzip to the Library directory (File was copied here in previous step) or click "No" if opening directly from CD or Download folder<br>"Skip All" to Extracting Firmware Libraries dialog<br>"Yes to All" to Overwrite Page Layout |
| 3 Check for Dependent Libraries | Project Tree -> Project Tab, Expand Libraries folder Yaskawa Toolbox<br>Take note of any User Libraries, indicated by the "blue book" icon. |
| 4 Start new / open existing project | File -> New, or File -> Open |
| 5 Insert the Library and any dependent libraries | In Project Tree, "Project" tab, R-Click "Libraries" -> Insert -> User Libraries<br>Navigate to find the Library (if you unzipped it to the "libraries" folder, you will see it right away)<br>Also insert any dependent libraries noted in Step 3 |
| 6 Delete duplicate project data types | In Project Tree, "Project" tab, expand "Data Types" folder for both the user library and the project library.<br>Delete any duplicates of "PLCTaskInfoTypes" or "MotionBlockTypes" from the project library.<br>R-click -> delete (or open, delete text)<br>*These data types are already defined within the imported library. Repeating the definition here causes compile errors since the same data types would be defined two times, even though the definitions are identical.* |
| 7 Use FB from new group in edit wizard | Click on programming worksheet whitespace.<br>Open Edit Wizard and the group dropdown list will have the library name.<br>User Library blocks appear as Blue by default<br>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. |

# Dependent Libraries

- *Open CamToolbox Library Project*


Project Tree Window

Project : C:\Users\Public\Documents\MotionWorks IE
Libraries
PLCopenPlus_v_2_2a*
YMotion*
DataTypes_Toolbox_v200
Math_Toolbox_v202
Data Types
Toolbox_DataTypes*

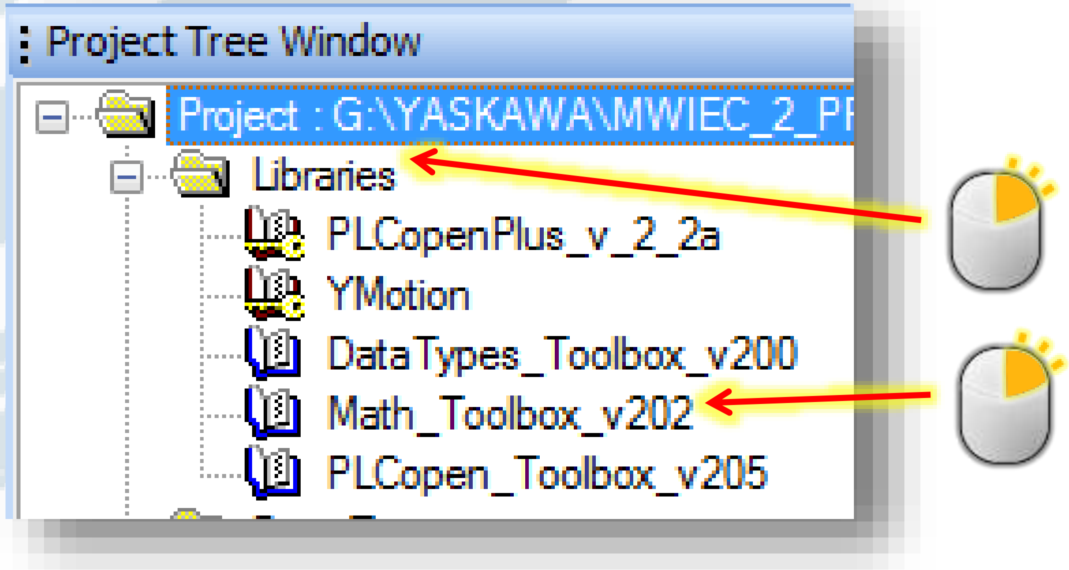
- *Dependent Libraries*
  - *PLCopen Plus*
  - *Ymotion*
  - *DataTypes_Toolbox*
  - *Math_Toolbox*

- *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**

## Insert Libraries

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


Project Tree Window

Project : G:\YASKAWA\MWIEC_2_PF

Libraries

PLCopenPlus_v_2_2a

YMotion

DataTypes_Toolbox_v200

Math_Toolbox_v202

PLCopen_Toolbox_v205

**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)**

## Updating Toolboxes

- *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

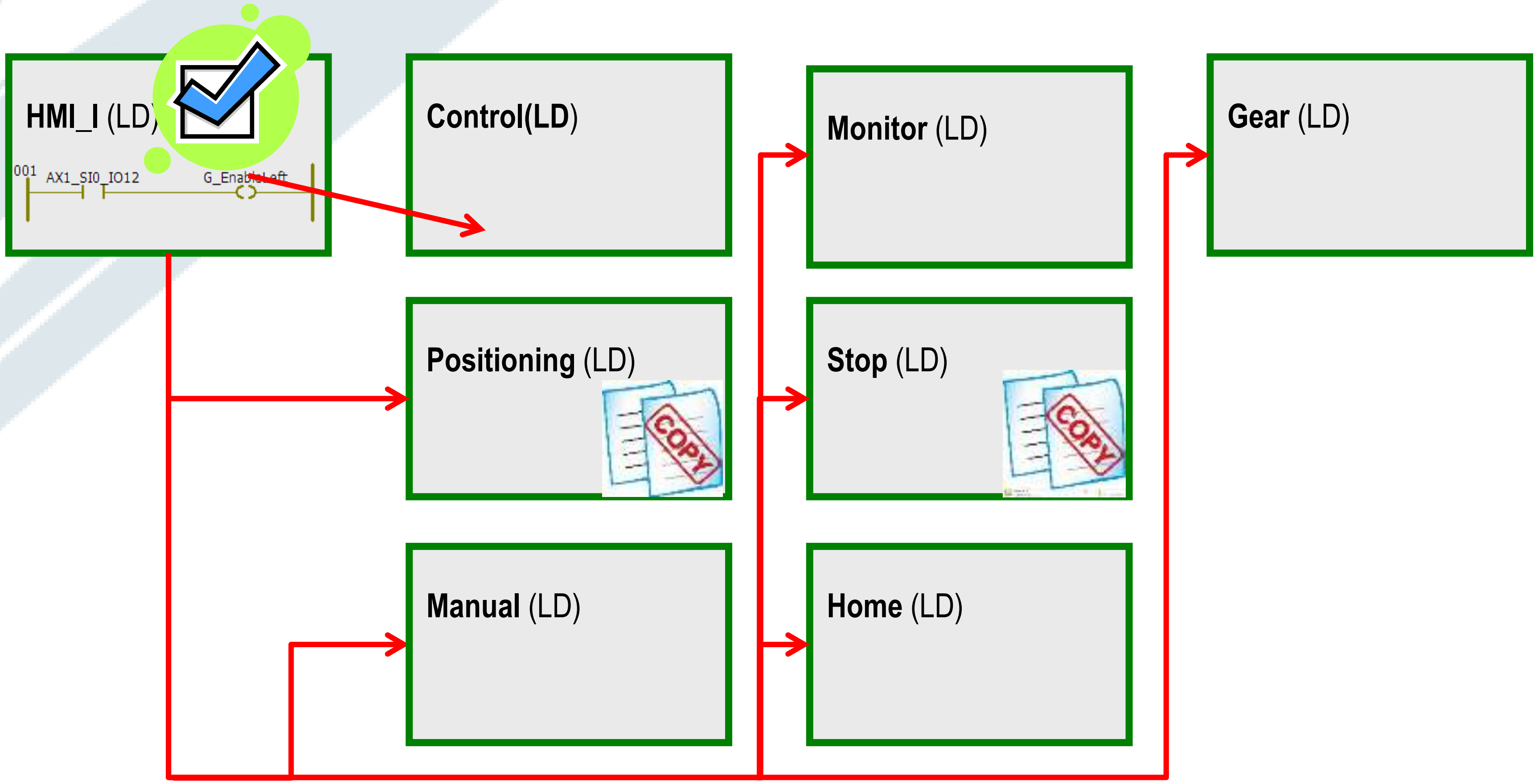
Jog function block

ReadAxisParameters function block

YASKAWA

## Class Project #2

- *Program Map for Second Project*
  - *Using PLCopen Toolbox*

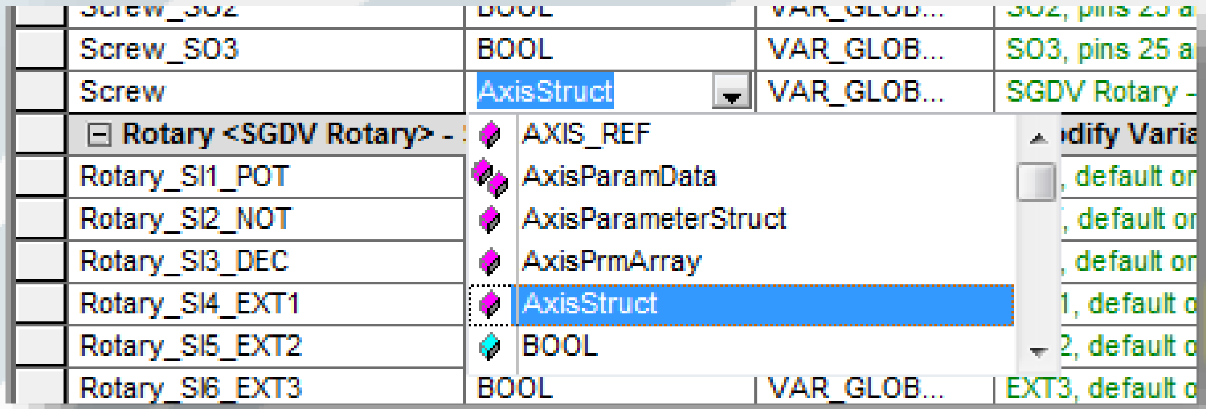


## Class Project #2

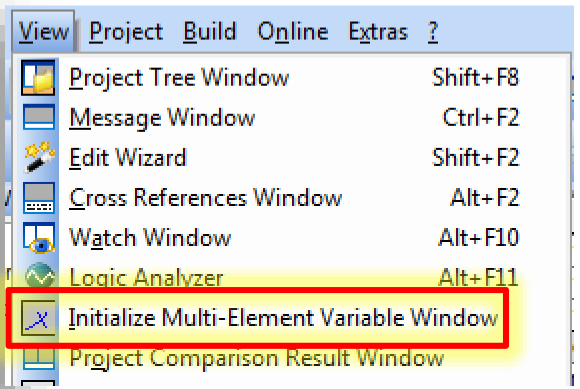
- *Back Up Existing Project*
- *Use the Class Project Template to create a new project*
  - *File-Unzip PLCopen*.zwt to new project name*
  - *Adjust IP address*
  - *Same Hardware Configuration – no update required*
  - *Open original project in another instance of MotionWorks IEC*
  - *Copy/Paste Logical POUs*
    - *Positioning*
    - *Stop*
  - *Insert Program Instances*
    - *Positioning*
    - *Stop*

## AxisStruct Data Type

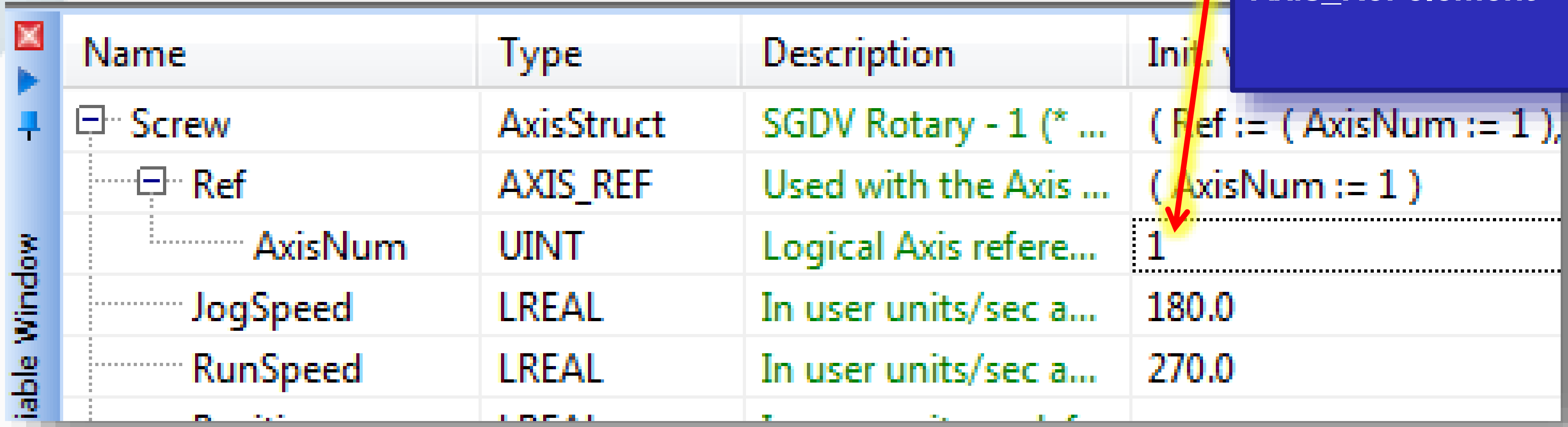
- *Global Variables: Change AxisRef to "AxisStruct" data type*
  - *AxisStruct comes from PLCopen Toolbox user library*



| | | | |
|---|---|---|---|
| Screw_SO3 | BOOL | VAR_GLOB... | SO3, pins 25 a |
| Screw | AxisStruct | VAR_GLOB... | SGDV Rotary - |
| ⊟ Rotary <SGDV Rotary> - | | | odify Varia |
| Rotary_SI1_POT | | | , default or |
| Rotary_SI2_NOT | | | , default or |
| Rotary_SI3_DEC | | | , default or |
| Rotary_SI4_EXT1 | | | 1, default o |
| Rotary_SI5_EXT2 | | | 2, default o |
| Rotary_SI6_EXT3 | BOOL | VAR_GLOB... | EXT3, default o |



View Project Build Online Extras ?
Project Tree Window Shift+F8
Message Window Ctrl+F2
Edit Wizard Shift+F2
Cross References Window Alt+F2
Watch Window Alt+F10
Logic Analyzer Alt+F11
Initialize Multi-Element Variable Window
Project Comparison Result Window

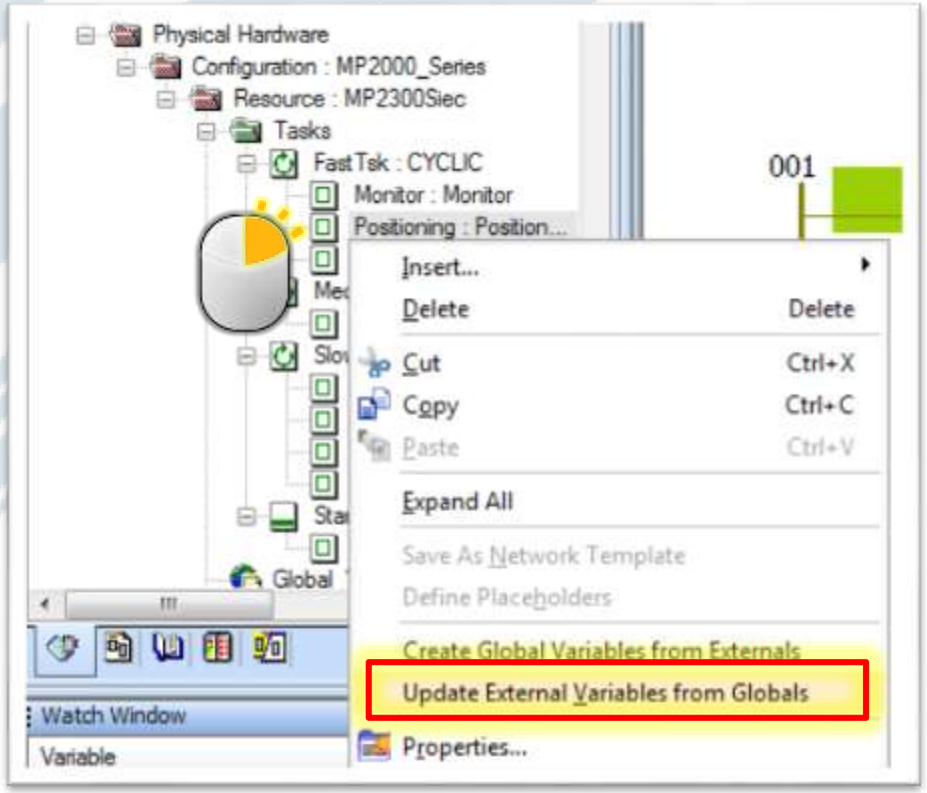


| Name | Type | Description | Init. v |
|---|---|---|---|
| ⊟ Screw | AxisStruct | SGDV Rotary - 1 (* ... | ( Ref := ( AxisNum := 1 ), |
| ⊟ Ref | AXIS_REF | Used with the Axis ... | ( AxisNum := 1 ) |
| AxisNum | UINT | Logical Axis refere... | 1 |
| JogSpeed | LREAL | In user units/sec a... | 180.0 |
| RunSpeed | LREAL | In user units/sec a... | 270.0 |

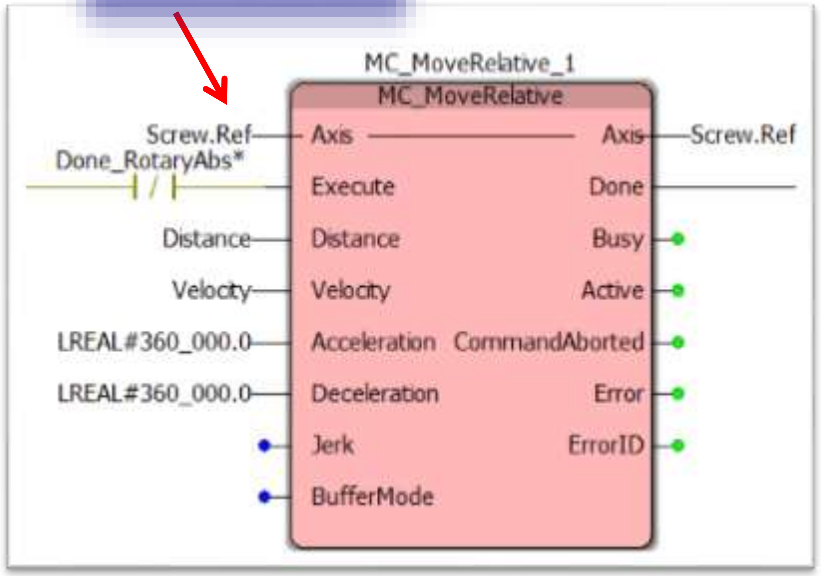
Initialize the Axis_Ref element

## AxisStruct Data Type

- *Update Axis Var_In_Out of copied code*

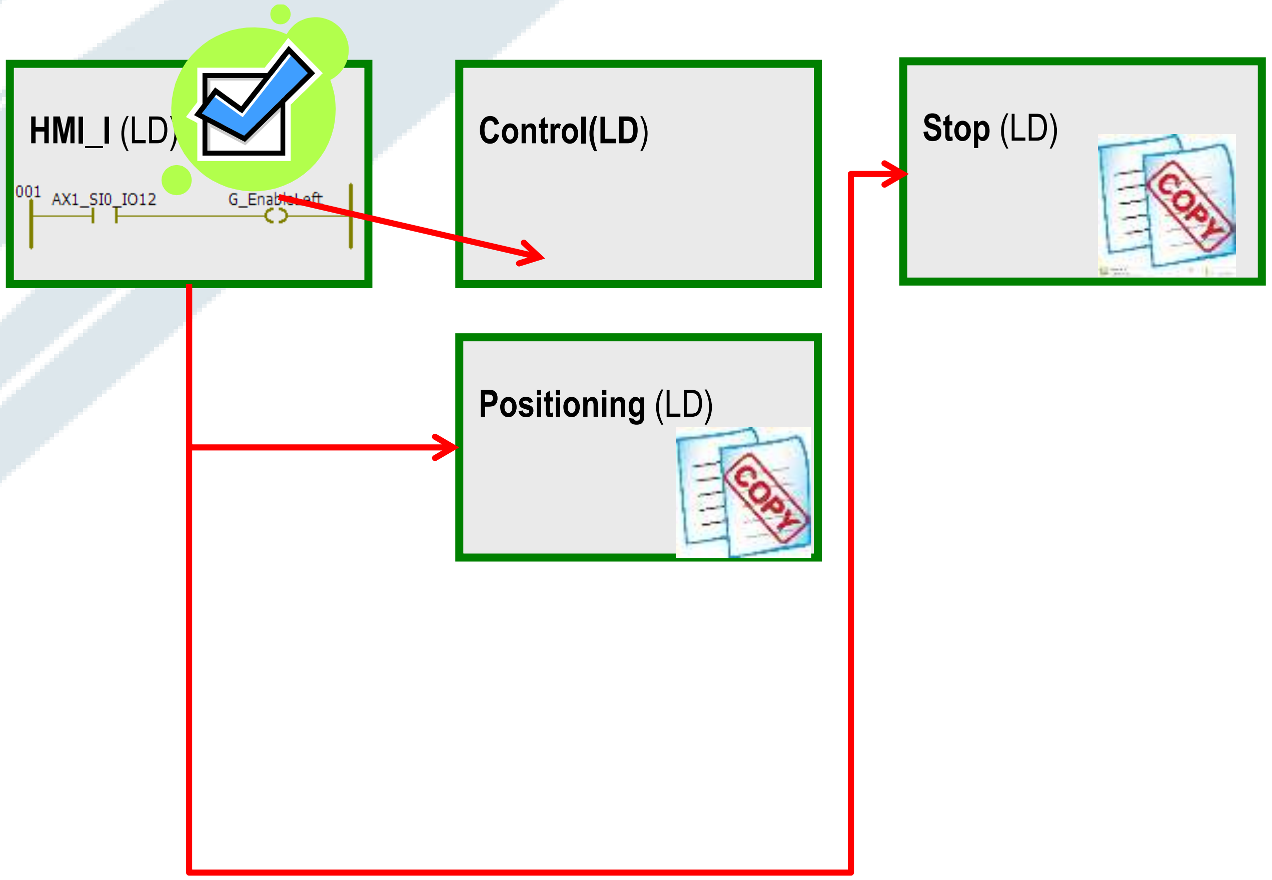


Notice the ".Ref"



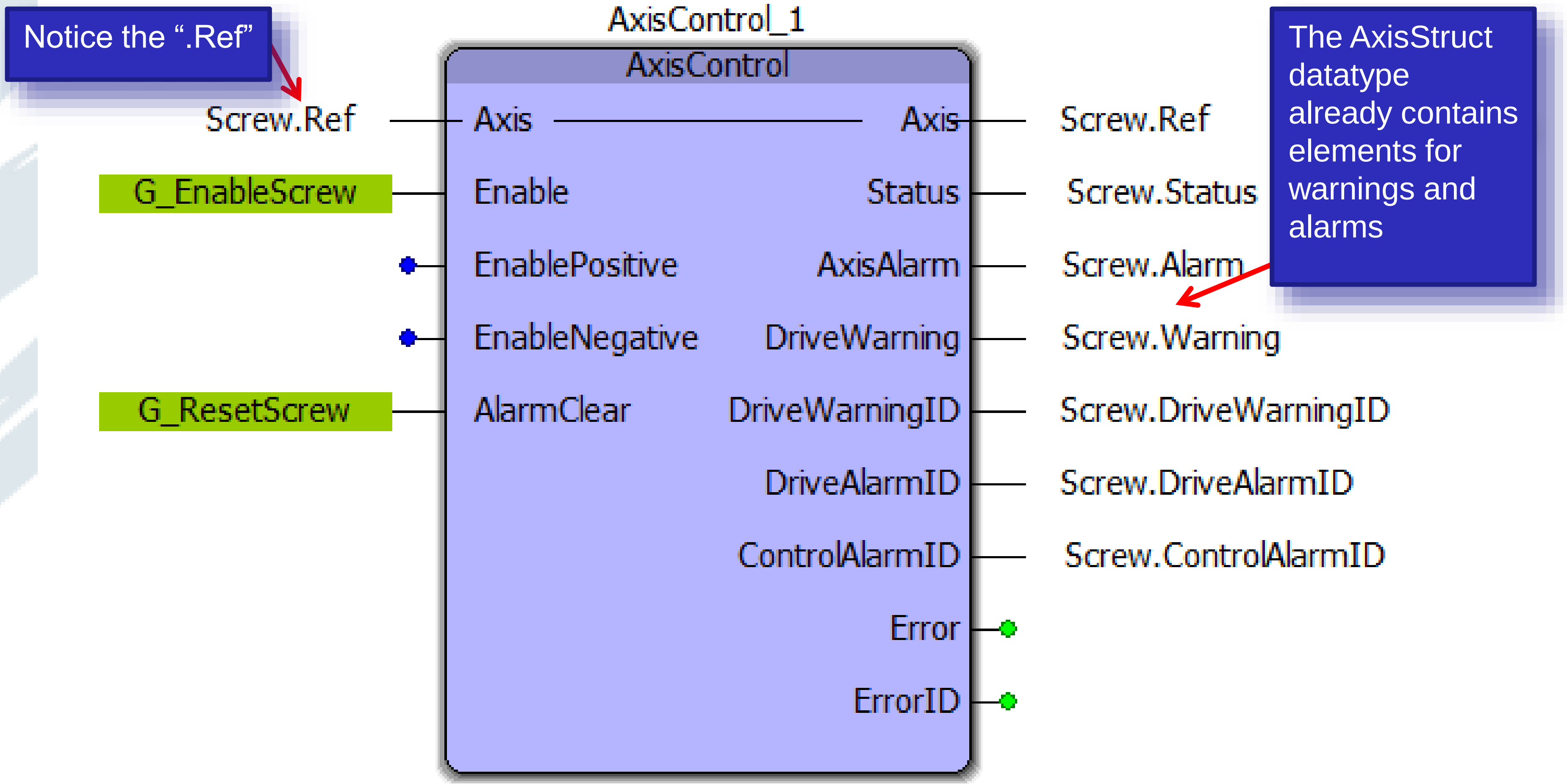
## Axis Control

- *Create the Control POU*
  - *Run in SlowTsk*



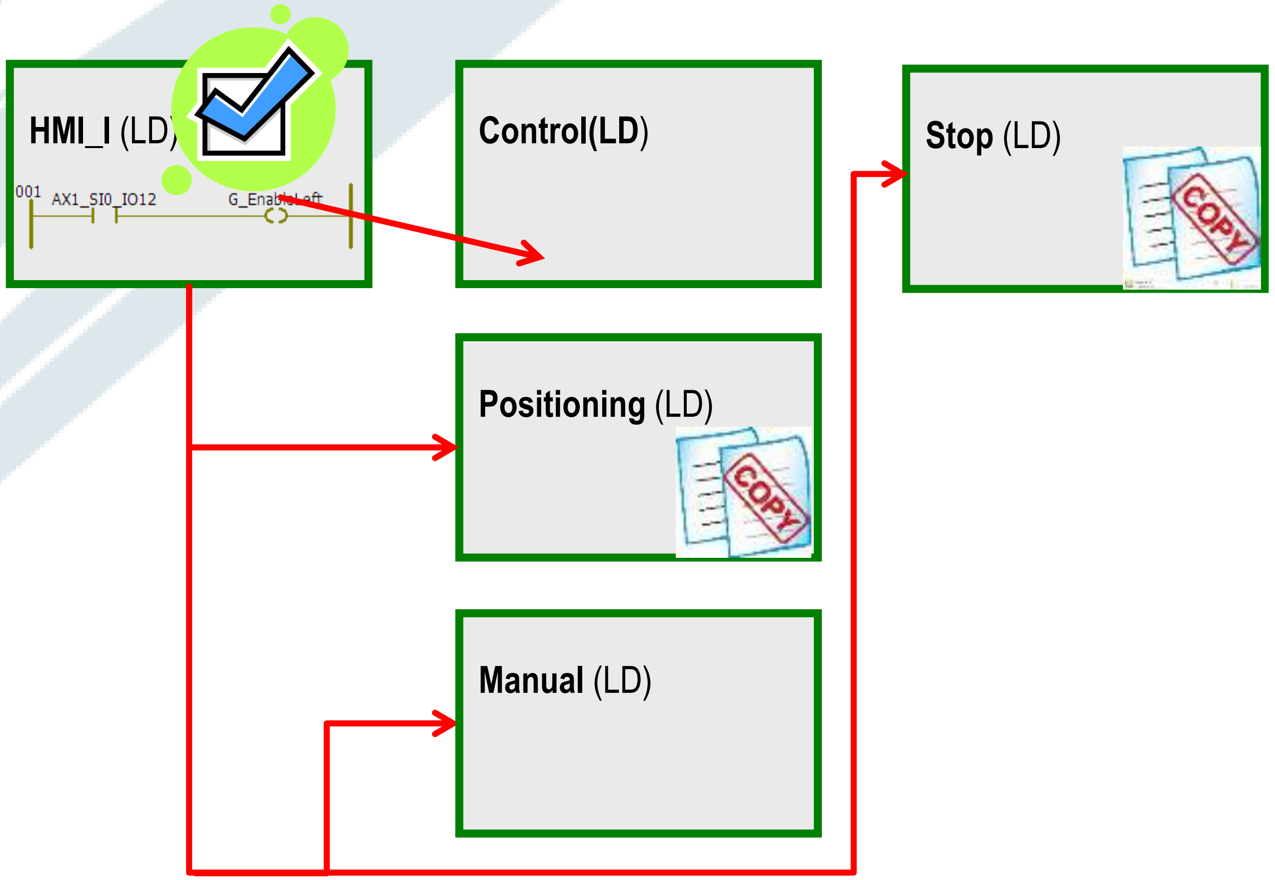
## Implement AxisControl

- *Use AxisControl for Screw, Rotary, Virtual*



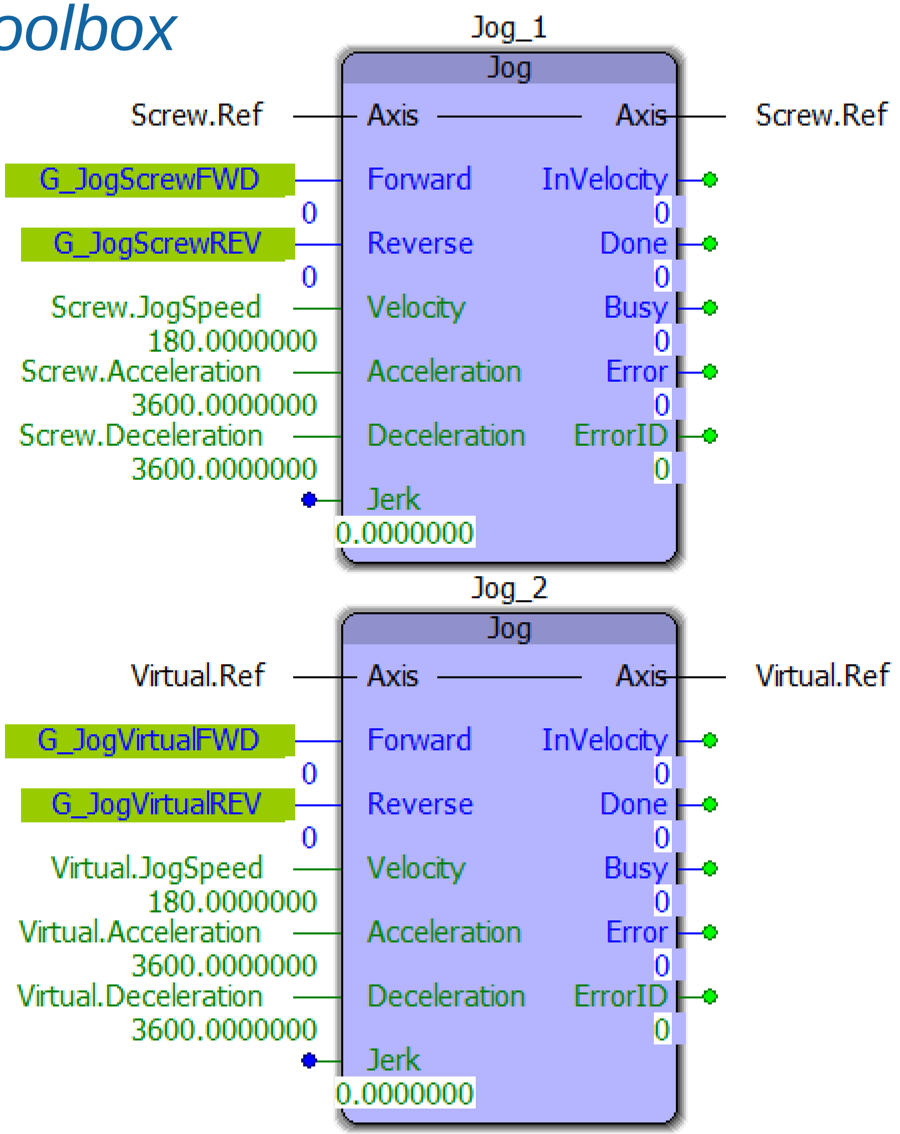
## Manual POU

- *Create the Manual POU*
  - *Run in SlowTsk*



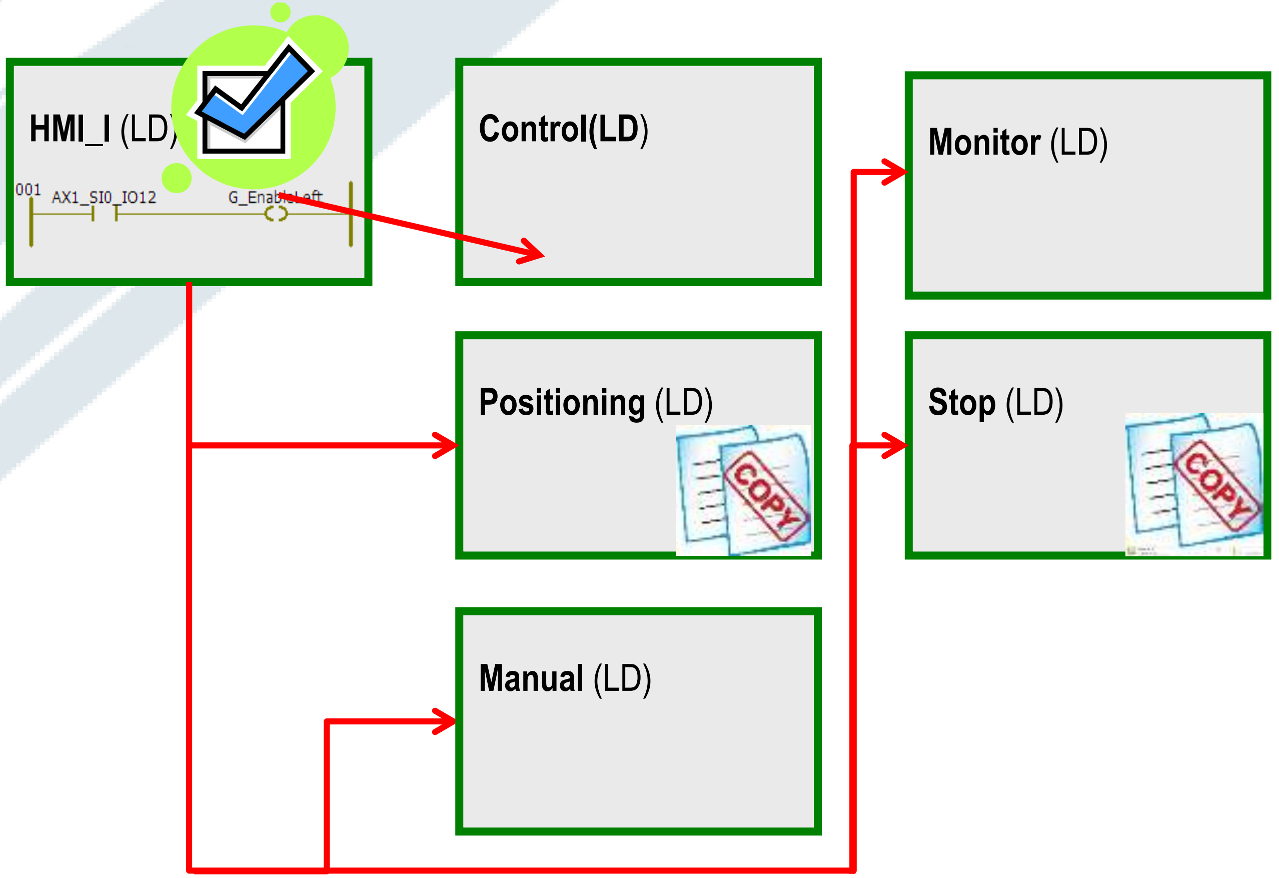
## Jog Function Block

- *Implement JOG function block from PLCopen Toolbox*
  - *New Program "Manual"*
- *Jog Screw*
- *Jog Virtual Axis*
- *Initialize AxisStruct elements*



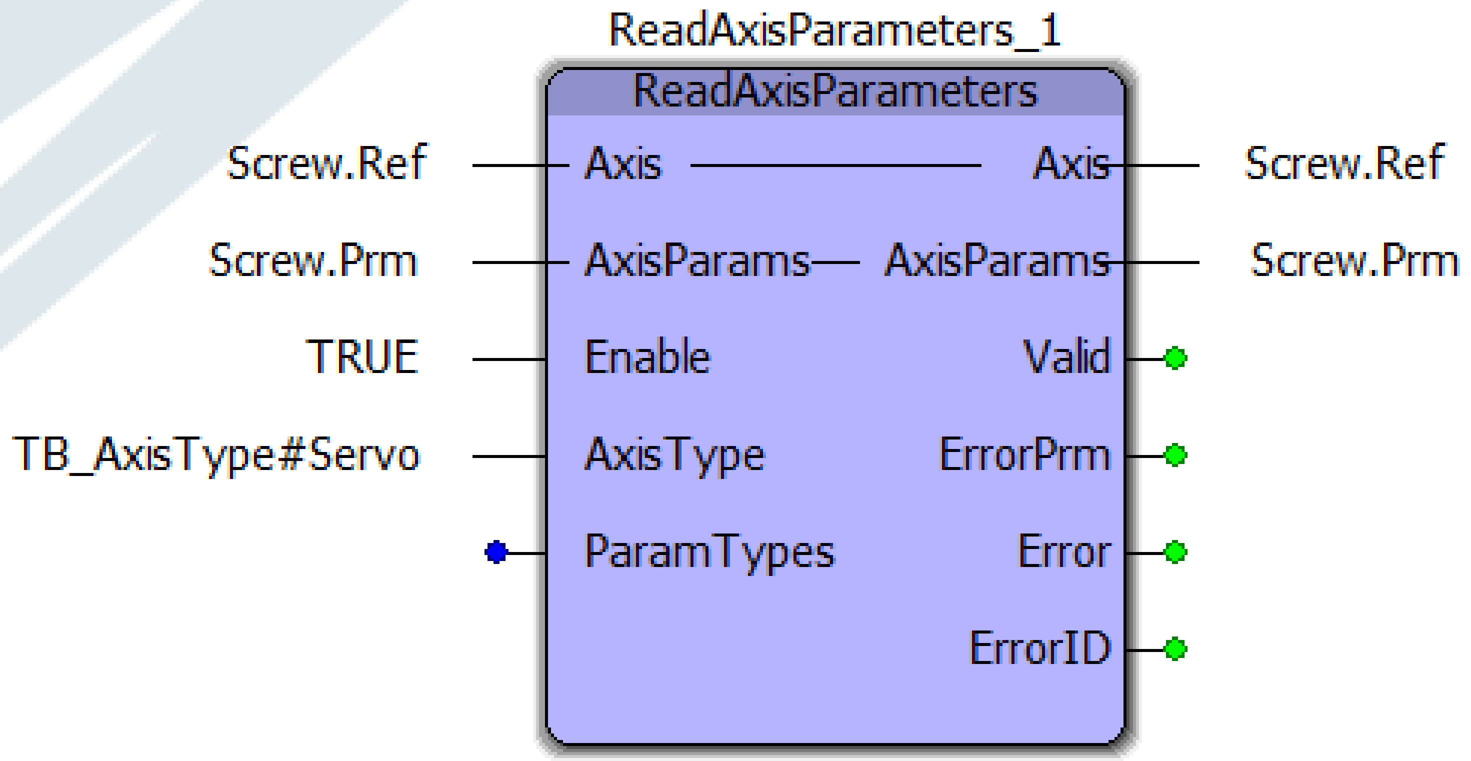
# Monitor POU

- *Create Monitor POU*
  - *Run in FastTsk*



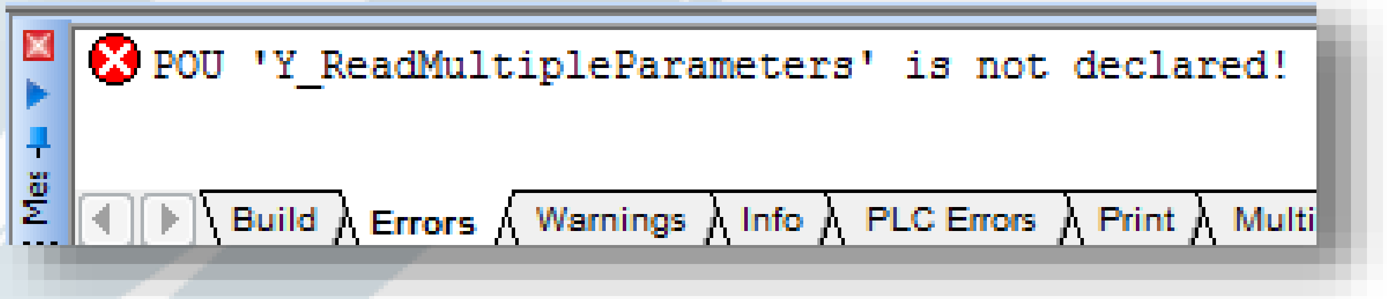
## ReadAxisParameters

- *Implement ReadAxisParameters*
  - *AxisType is an enumerated type (Right-click Help)*



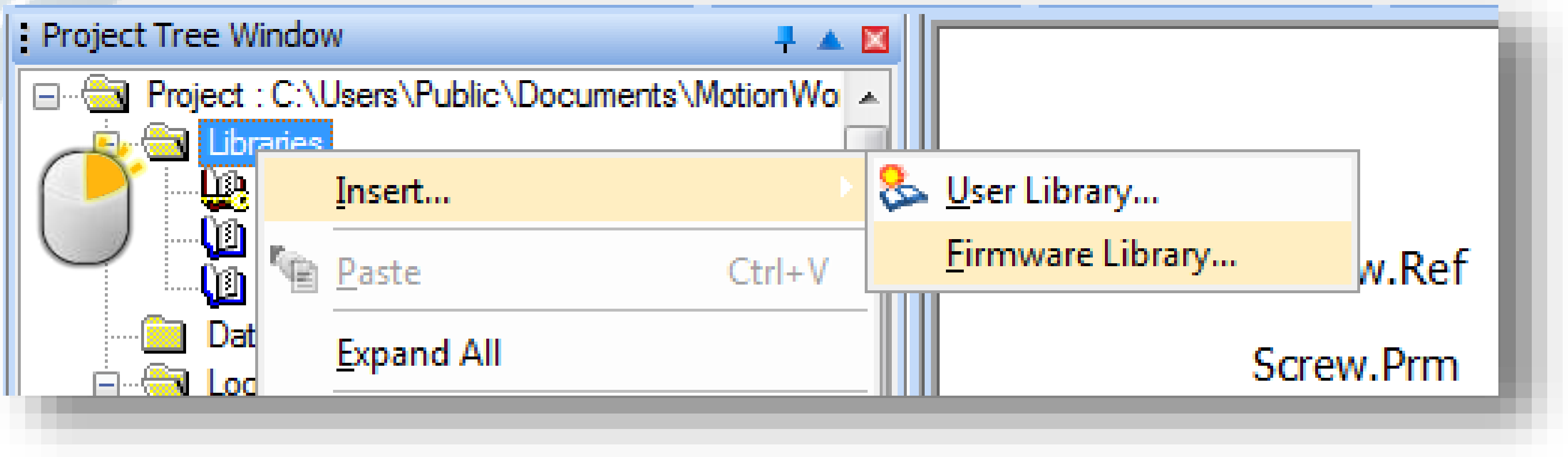
## ReadAxisParameters

- *ReadAxisParameters requires Y_Motion Firmware Library*



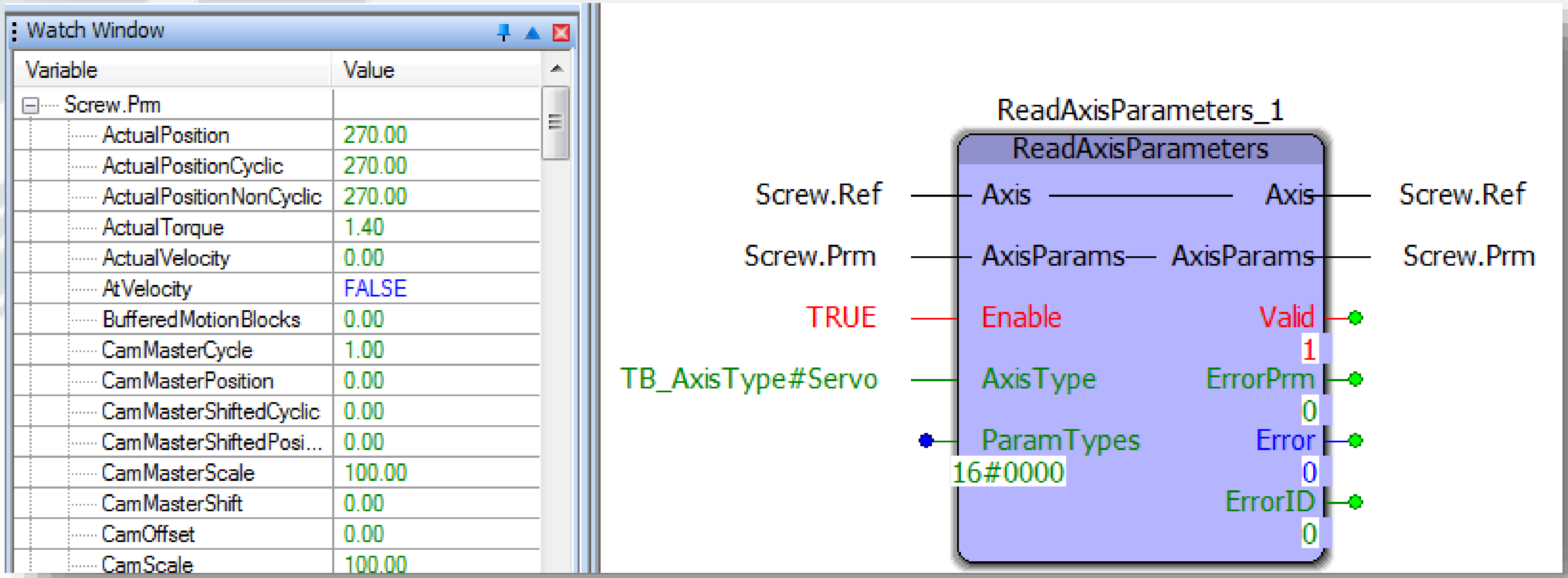
```
POU 'Y_ReadMultipleParameters' is not declared!
```

- *Insert Y_Motion Firmware Library*



## ReadAxisParameters

- *View Axis Parameters in Watch Window*
  - *"Parameter" = "feedback data" in PLCopen*


Watch Window

| Variable | Value |
|---|---|
| Screw.Prm | |
| ActualPosition | 270.00 |
| ActualPositionCyclic | 270.00 |
| ActualPositionNonCyclic | 270.00 |
| ActualTorque | 1.40 |
| ActualVelocity | 0.00 |
| AtVelocity | FALSE |
| BufferedMotionBlocks | 0.00 |
| CamMasterCycle | 1.00 |
| CamMasterPosition | 0.00 |
| CamMasterShiftedCyclic | 0.00 |
| CamMasterShiftedPosi... | 0.00 |
| CamMasterScale | 100.00 |
| CamMasterShift | 0.00 |
| CamOffset | 0.00 |
| CamScale | 100.00 |

## Further Highlights

- *ProductBuffer*
- *MoveRelativeByTime*

- *PLCopen Toolbox User Library for MotionWorks IEC*
  - *Tutorial Videos Playlist on YouTube Channel*
    - » *https://www.youtube.com/playlist?list=PLNAENlyEDCkybLQ25iijwcRAZyG4NGBPb*
  - *Help contains video links*

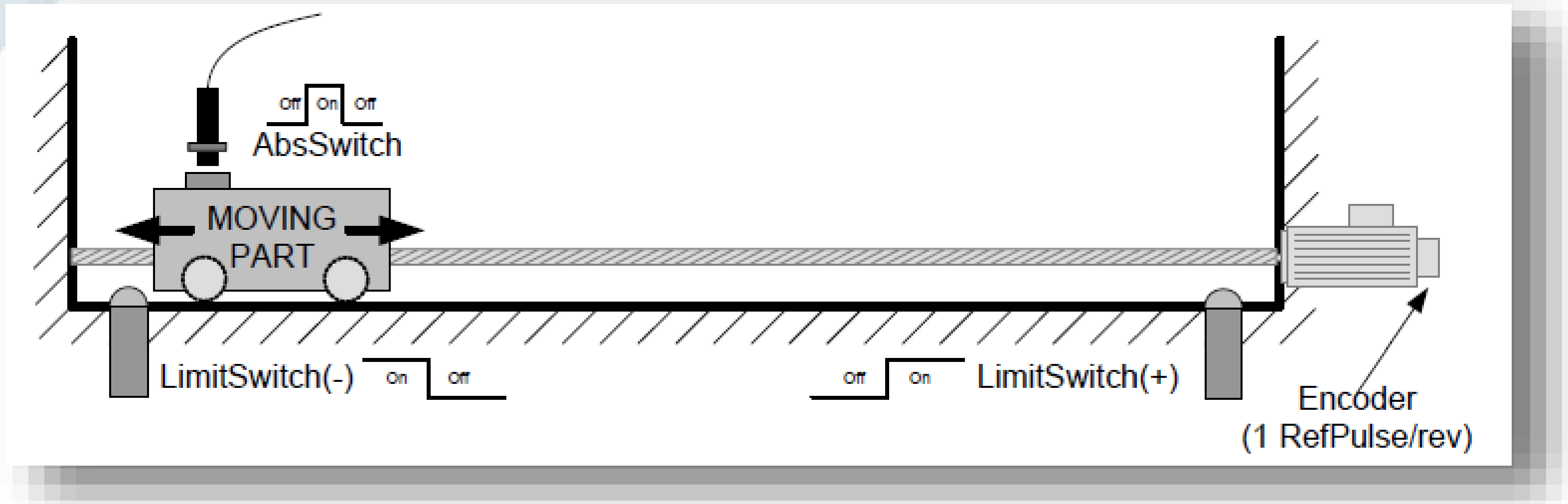
## Homing

YASKAWA

## Introduction

### *What is Homing?*

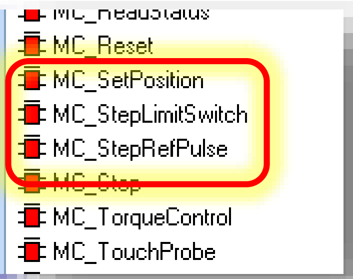
- *A repeatable move sequence to move the axis from an unknown position to a known position*
- *Executed at every power-up (incremental encoder)*
- *Executed once when axis is commissioned (absolute encoder)*
- *Usually done at slow speed*
- *May involve proximity sensors, encoder reference pulse, hard stops, limit switches, torque limits*

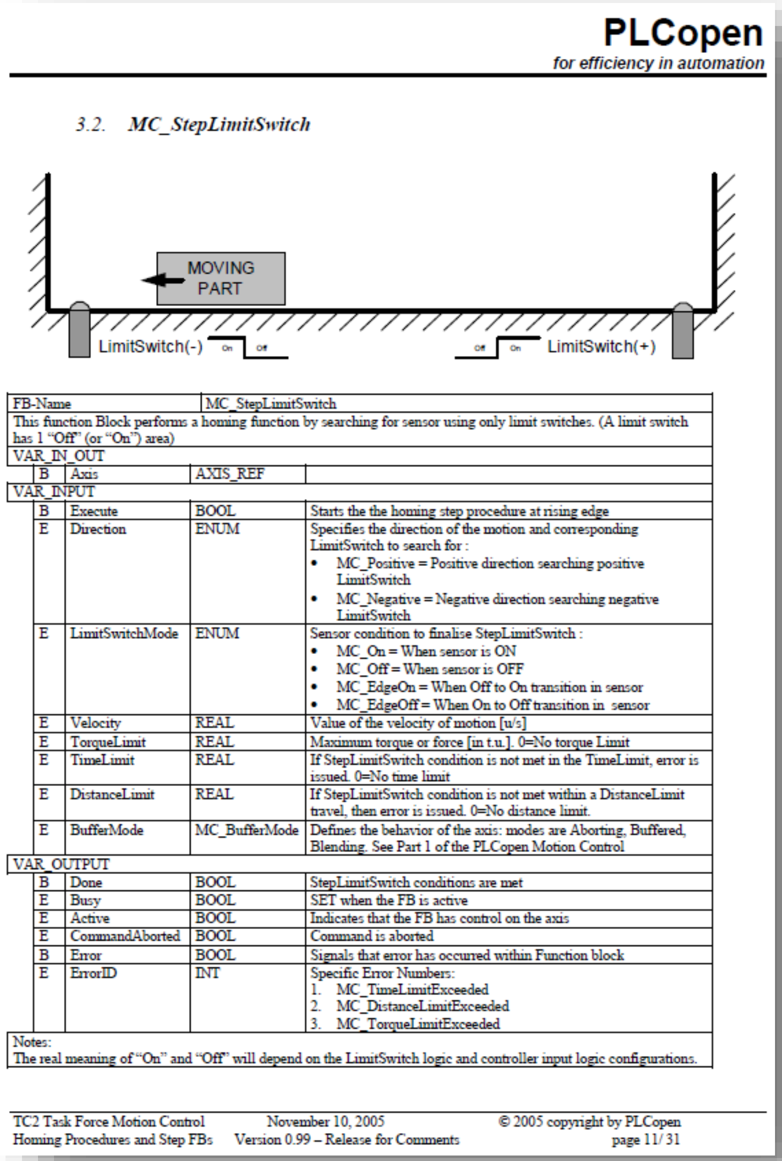


# PLCopen Homing

- *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_ReadStatus
MC_Reset
MC_SetPosition
MC_StepLimitSwitch
MC_StepRefPulse
MC_Stop
MC_TorqueControl
MC_TouchProbe


MOVING PART
LimitSwitch(-)
On
Off
Off
On
LimitSwitch(+)

PLCopen
for efficiency in automation

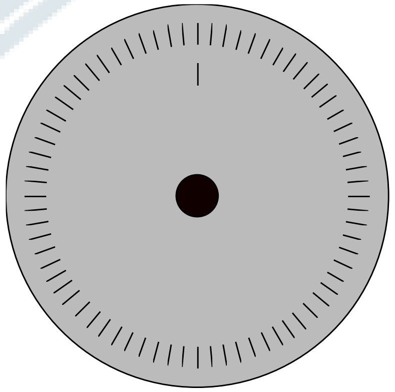
3.2. *MC_StepLimitSwitch*

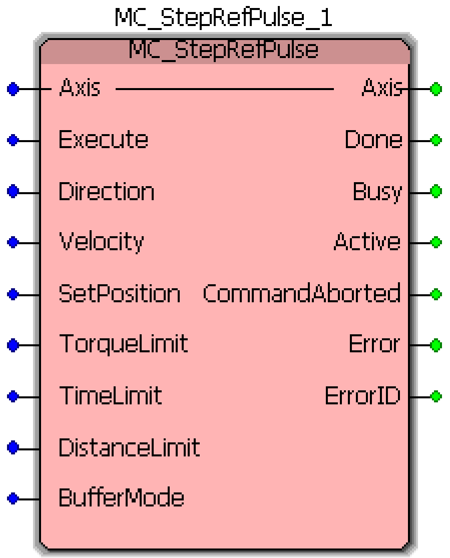
| | | | |
|---|---|---|---|
| FB-Name | MC_StepLimitSwitch | | |
| This function Block performs a homing function by searching for sensor using only limit switches. (A limit switch has 1 "Off" (or "On") area) | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | |
| VAR_INPUT | | | |
| B | Execute | BOOL | Starts the the homing step procedure at rising edge |
| E | Direction | ENUM | Specifies the direction of the motion and corresponding LimitSwitch to search for :<br>• MC_Positive = Positive direction searching positive LimitSwitch<br>• MC_Negative = Negative direction searching negative LimitSwitch |
| E | LimitSwitchMode | ENUM | Sensor condition to finalise StepLimitSwitch :<br>• MC_On = When sensor is ON<br>• MC_Off = When sensor is OFF<br>• MC_EdgeOn = When Off to On transition in sensor<br>• MC_EdgeOff = When On to Off transition in sensor |
| E | Velocity | REAL | Value of the velocity of motion [u/s] |
| E | TorqueLimit | REAL | Maximum torque or force [in t.u.]. 0=No torque Limit |
| E | TimeLimit | REAL | If StepLimitSwitch condition is not met in the TimeLimit, error is issued. 0=No time limit |
| E | DistanceLimit | REAL | If StepLimitSwitch condition is not met within a DistanceLimit travel, then error is issued. 0=No distance limit. |
| E | BufferMode | MC_BufferMode | Defines the behavior of the axis: modes are Aborting, Buffered, Blending. See Part 1 of the PLCopen Motion Control |
| VAR_OUTPUT | | | |
| B | Done | BOOL | StepLimitSwitch conditions are met |
| E | Busy | BOOL | SET when the FB is active |
| E | Active | BOOL | Indicates that the FB has control on the axis |
| E | CommandAborted | BOOL | Command is aborted |
| B | Error | BOOL | Signals that error has occurred within Function block |
| E | ErrorID | INT | Specific Error Numbers:<br>1. MC_TimeLimitExceeded<br>2. MC_DistanceLimitExceeded<br>3. MC_TorqueLimitExceeded |
| Notes:<br>The real meaning of "On" and "Off" will depend on the LimitSwitch logic and controller input logic configurations. | | | |

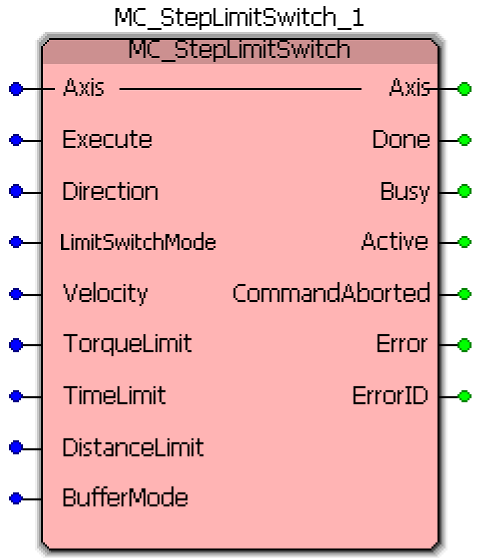
TC2 Task Force Motion Control November 10, 2005 © 2005 copyright by PLCopen
Homing Procedures and Step FBs Version 0.99 – Release for Comments page 11/ 31

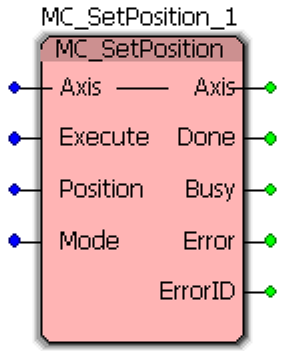
## Supported Function Blocks

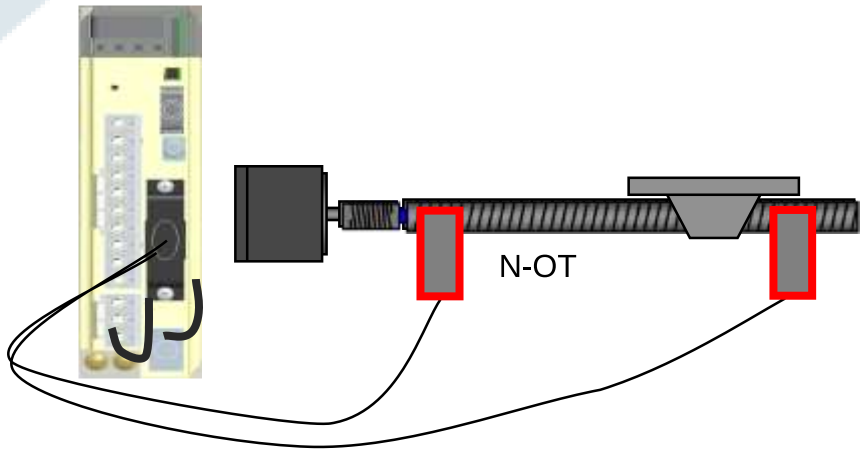
- *MC_StepRefPulse*
  - *PLCopen p.16-17*
  - *C-pulse (Index, Reference)*
- *MC_StepLimitSwitch*
  - *PLCopen p.11-12*
  - *P-OT, N-OT*
- *MC_FinishHoming*
  - *PLCopen p.15*
- *MC_SetPosition*
  - *Current position = any value*





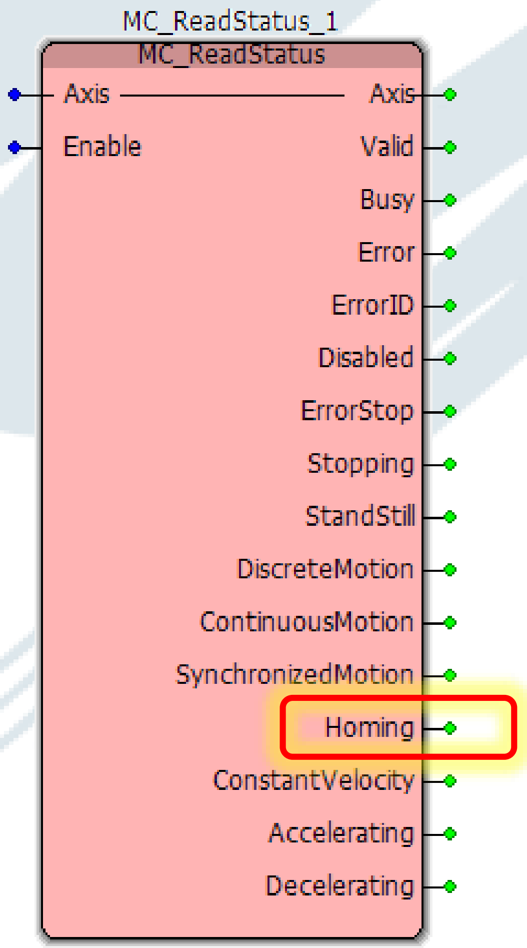


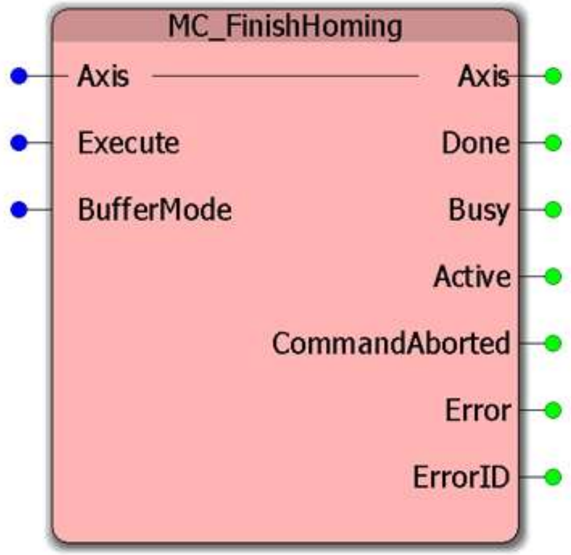


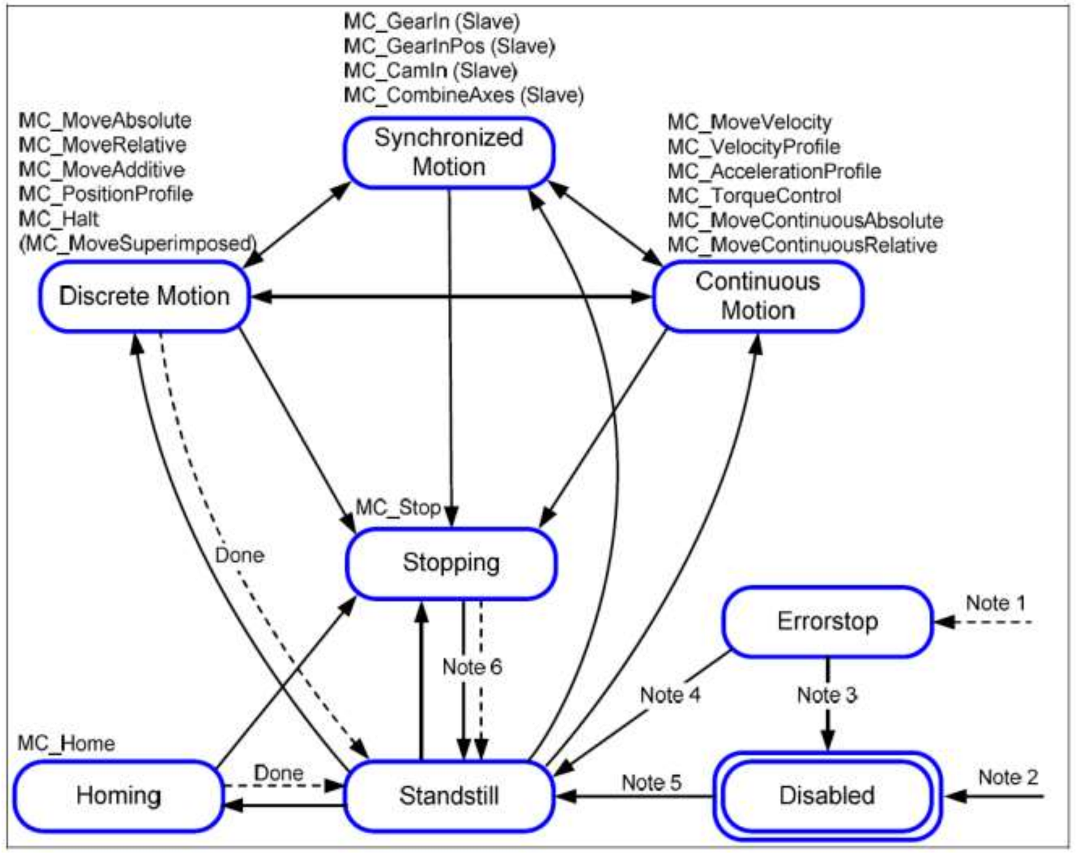


## Homing State

- *Homing is a State of PLCopen*
- *Monitor: MC_ReadAxisStatus*

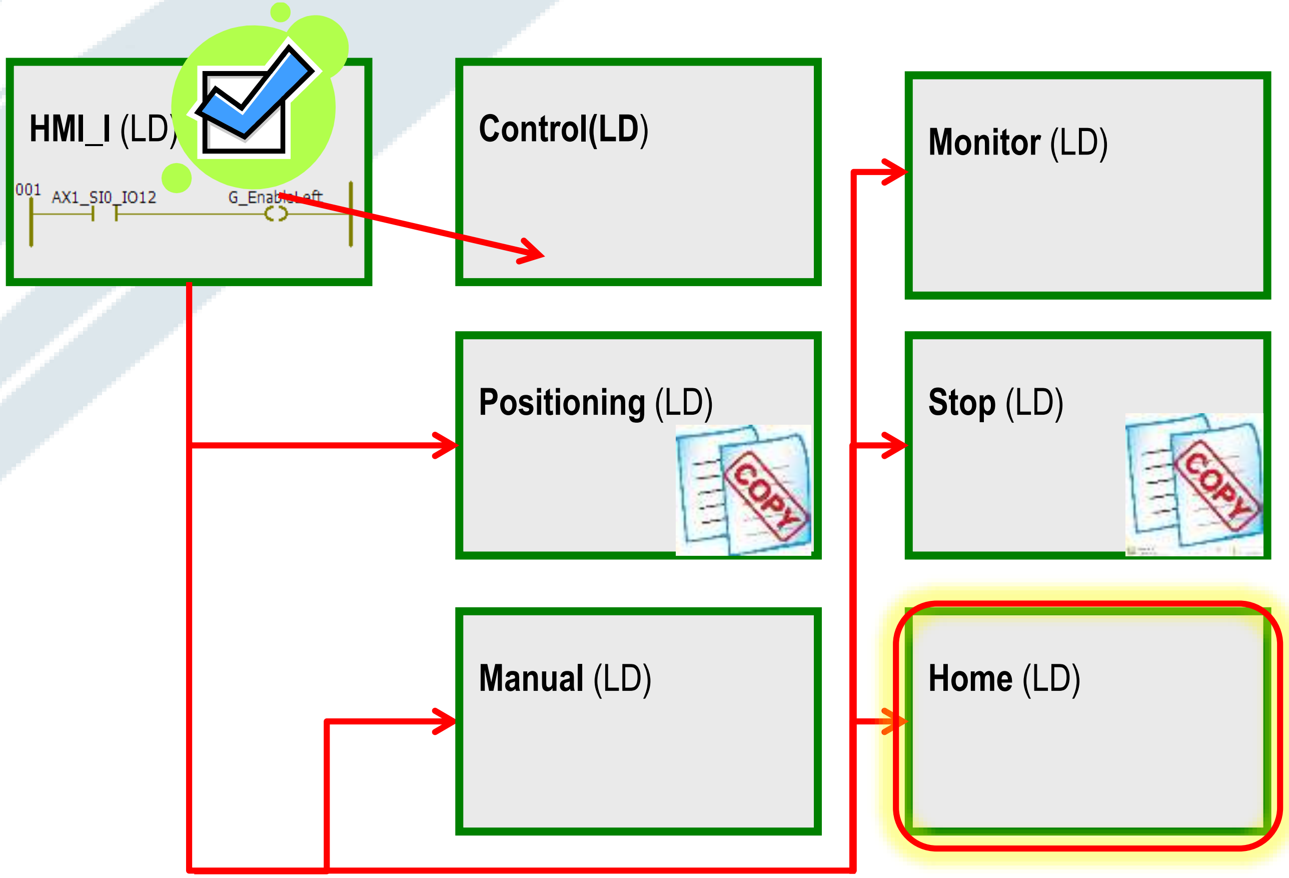






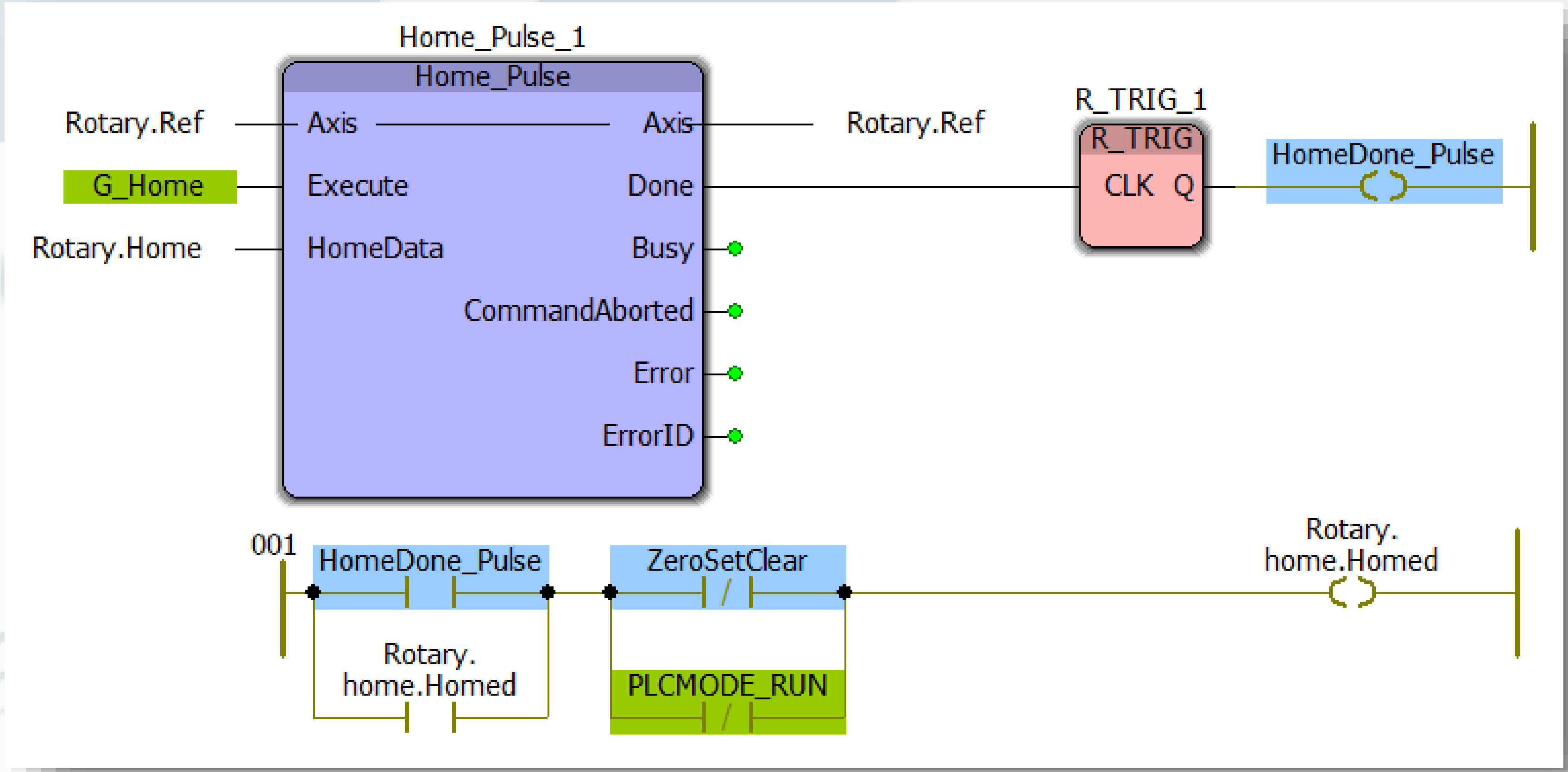
## Home POU

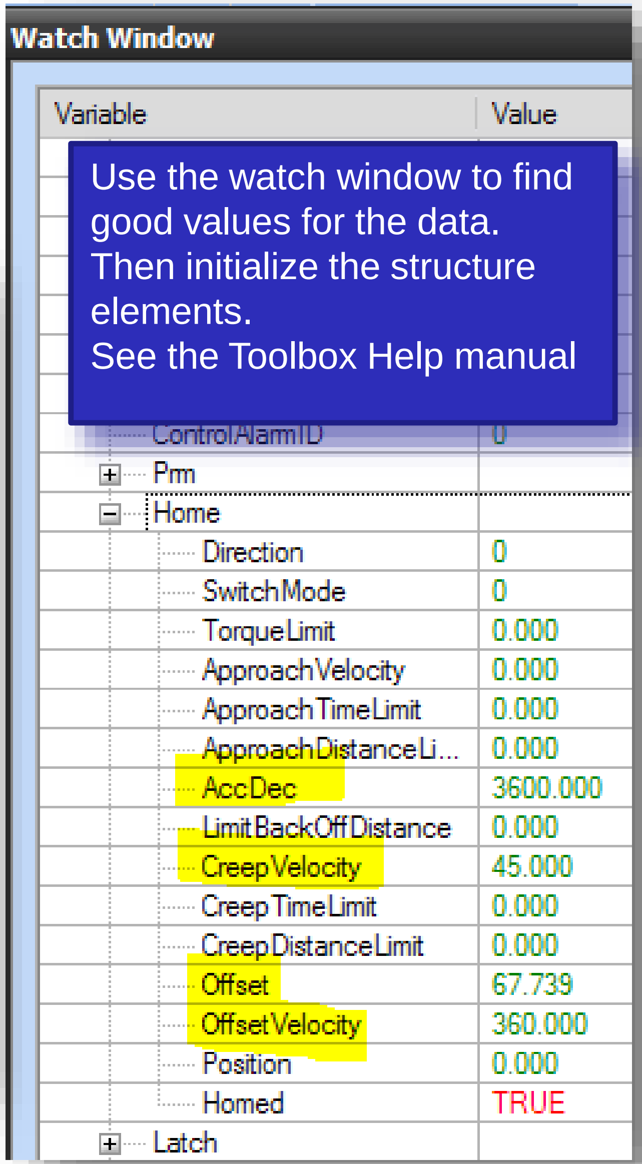
- *Add Home Program POU*
  - *Run in MedTsk*



## PLCopen Toolbox Homing

- *Home_Pulse for Rotary*
  - *"HomeStruct" data type for Home Data*



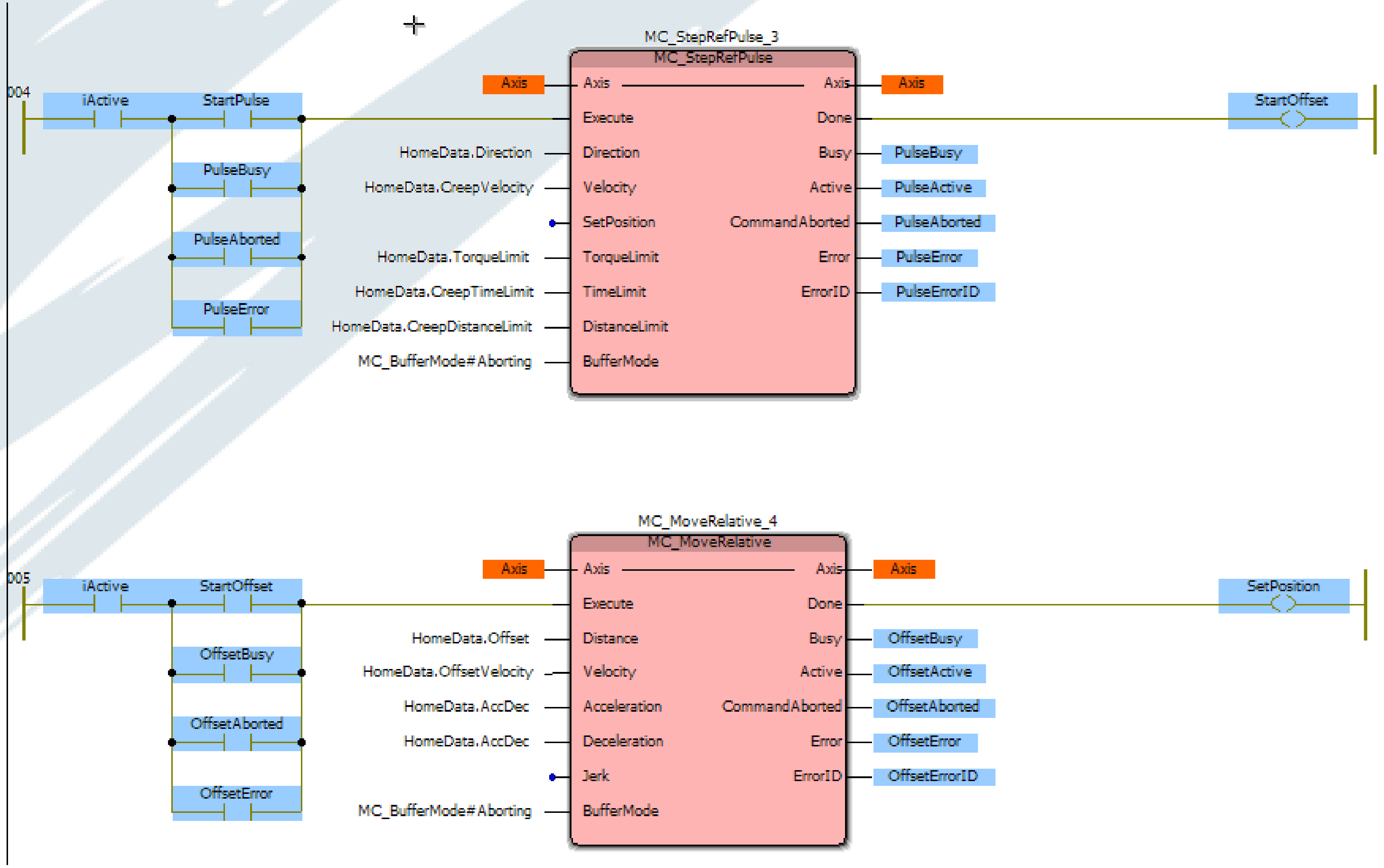

Watch Window

Use the watch window to find good values for the data. Then initialize the structure elements.
See the Toolbox Help manual

| Variable | Value |
|---|---|
| ControlAlarmID | 0 |
| Prm | |
| Home | |
| Direction | 0 |
| SwitchMode | 0 |
| TorqueLimit | 0.000 |
| ApproachVelocity | 0.000 |
| ApproachTimeLimit | 0.000 |
| ApproachDistanceLi... | 0.000 |
| AccDec | 3600.000 |
| LimitBackOffDistance | 0.000 |
| CreepVelocity | 45.000 |
| CreepTimeLimit | 0.000 |
| CreepDistanceLimit | 0.000 |
| Offset | 67.739 |
| OffsetVelocity | 360.000 |
| Position | 0.000 |
| Homed | TRUE |
| Latch | |

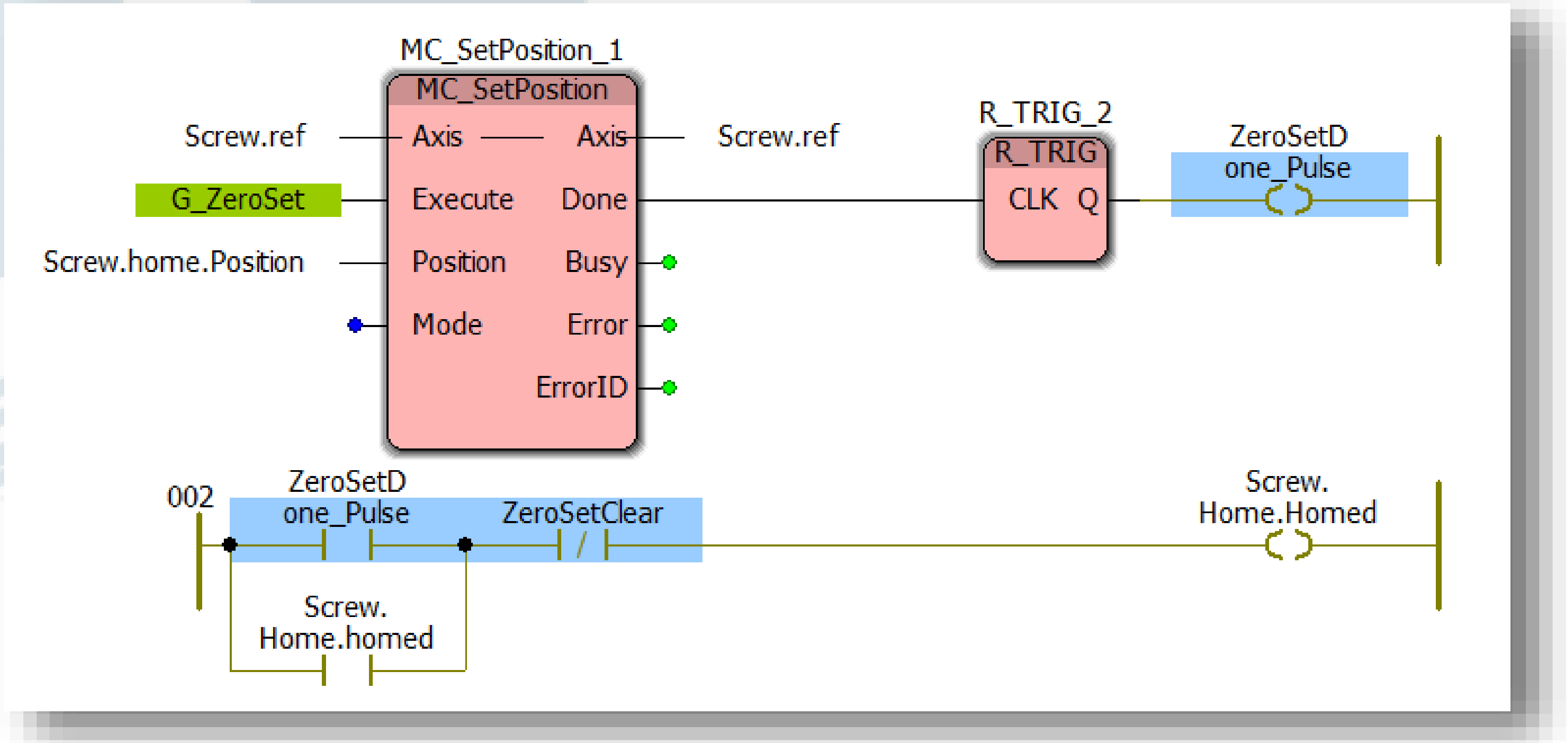
# Homing

## Inside Home_Pulse



## Zero Set

- *Simple Zero Set Example (For Screw)*
  - *Arbitrarily set position to zero (visual calibration)*
  - *A one-time "zero set" for absolute encoders*



## Electronic Gear

- Overview
- PLCopen Gearing
- Gear program POU
- Program Example
- Program Test

YASKAWA

## Electronic Gear

## Overview

- *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*

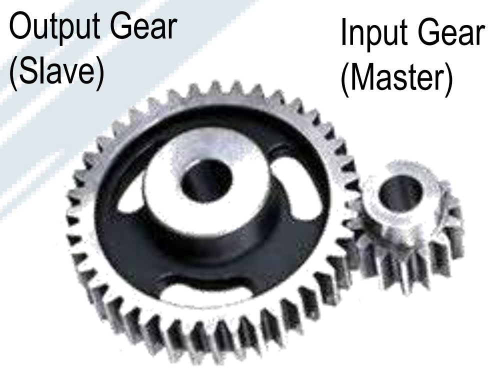
$$GearRatio = \frac{\text{Slave Units (Output)}}{\text{Master Units (Input)}}$$

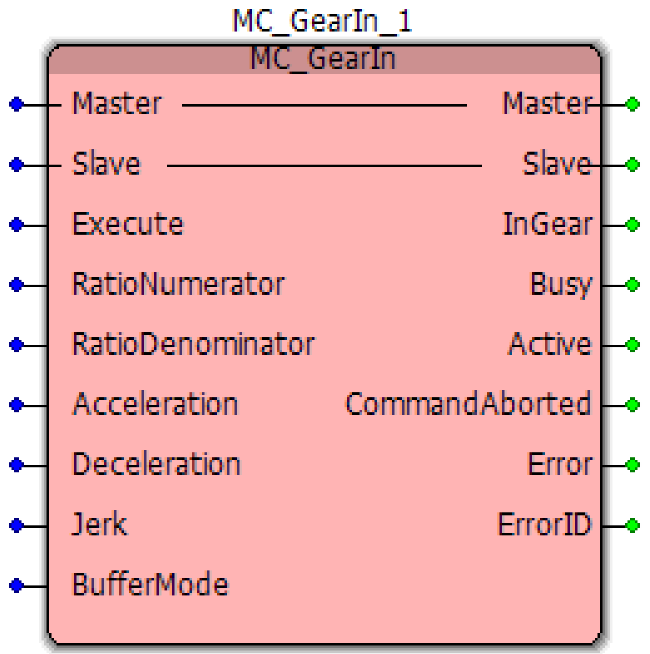
Servo Axis

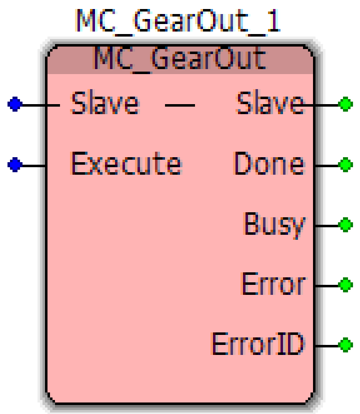
External Encoder

# PLCopen Gearing

- *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*



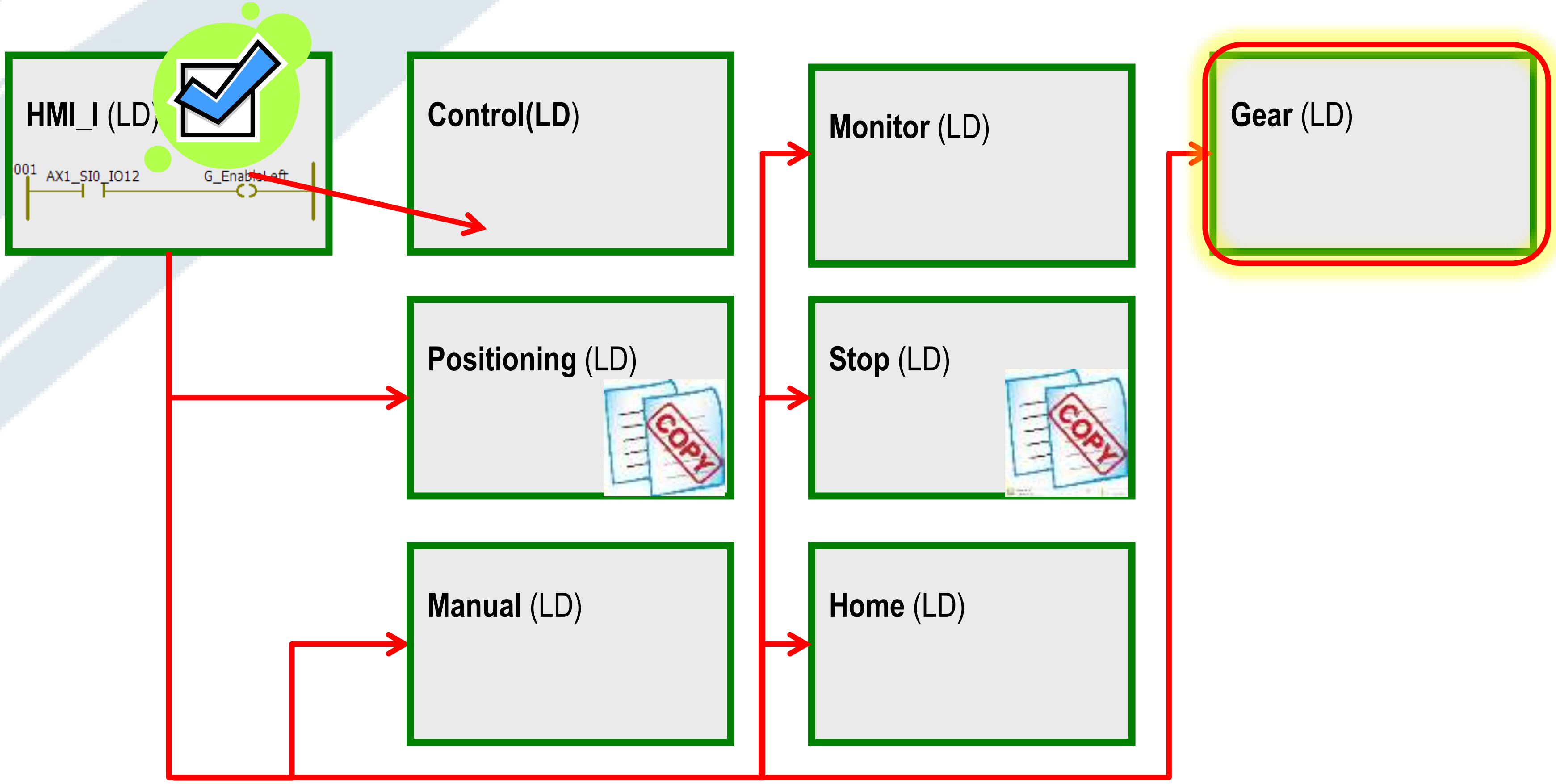




Electronic Gear

# Class Project #2

- *Create Gear (LD) POU*
  - *Run in MedTsk*

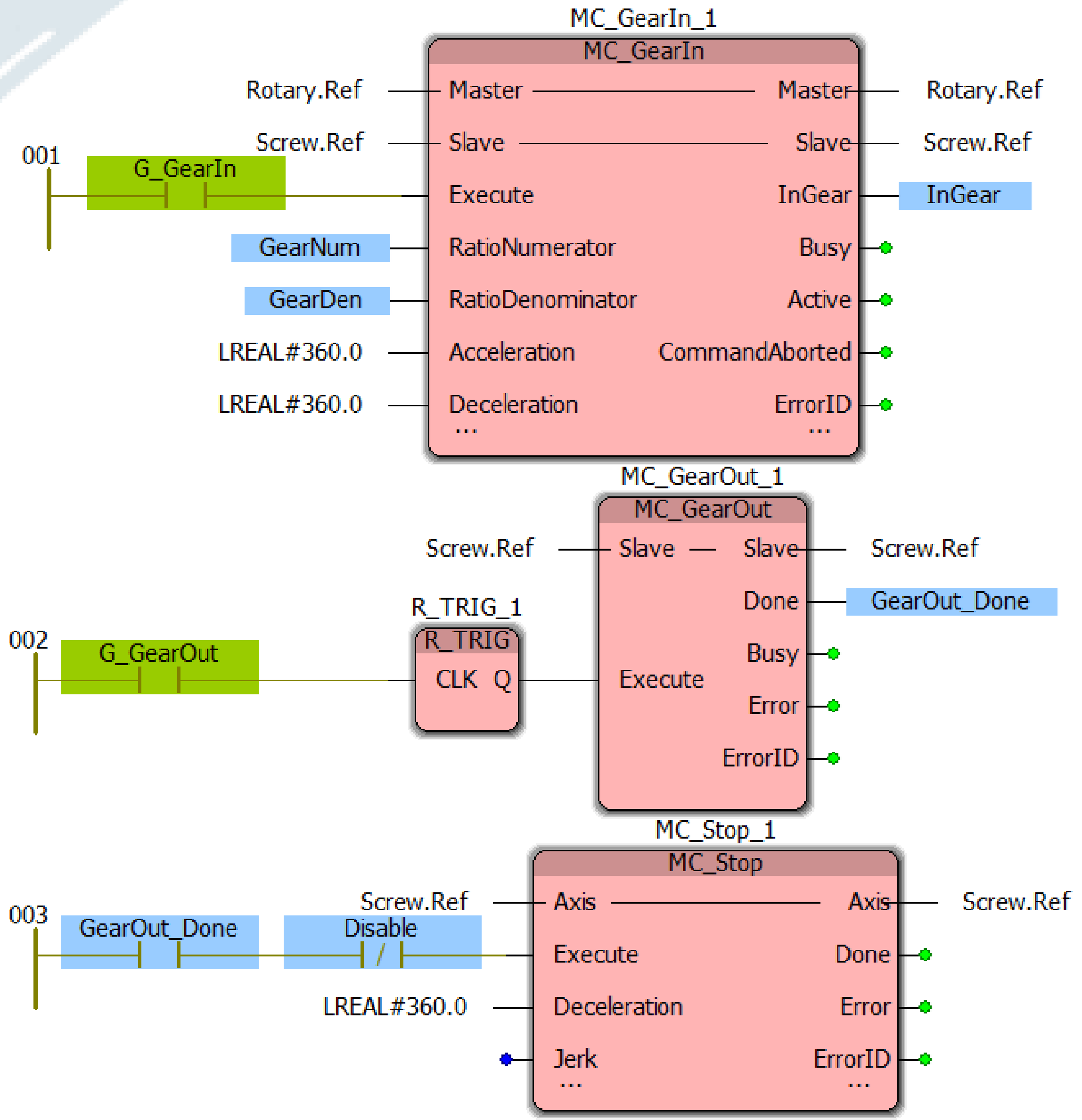


Electronic Gear

# Program Example

## *Test the program*

- *Jog the master*
- *Execute MC_GearIn*
  - *While master is moving*
  - *While master is stopped*
- *Adjust Ratio*
- *Observe the InGear output*



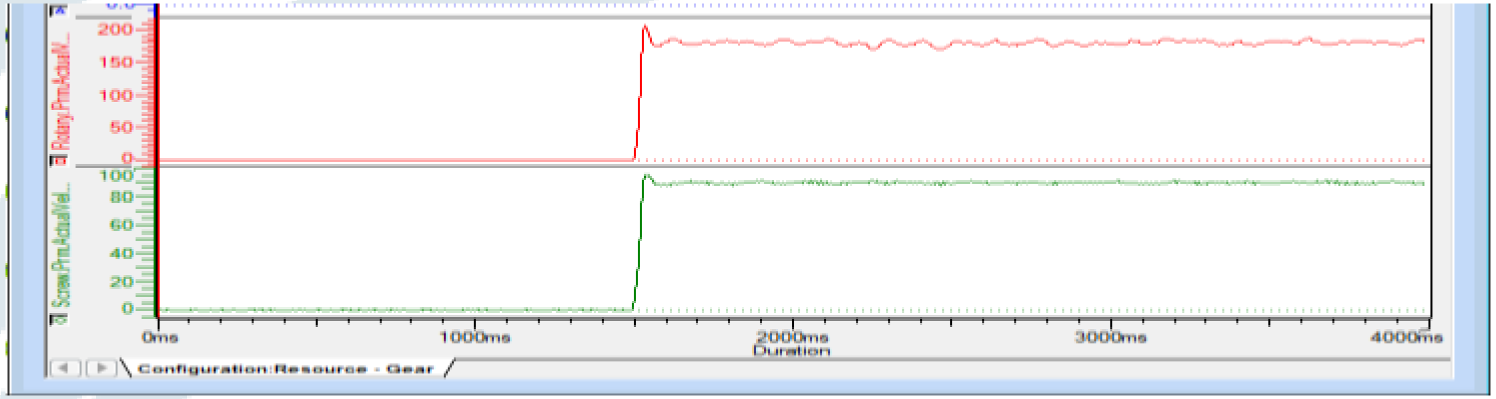
## Program Test

*Use the logic analyzer to determine the following*

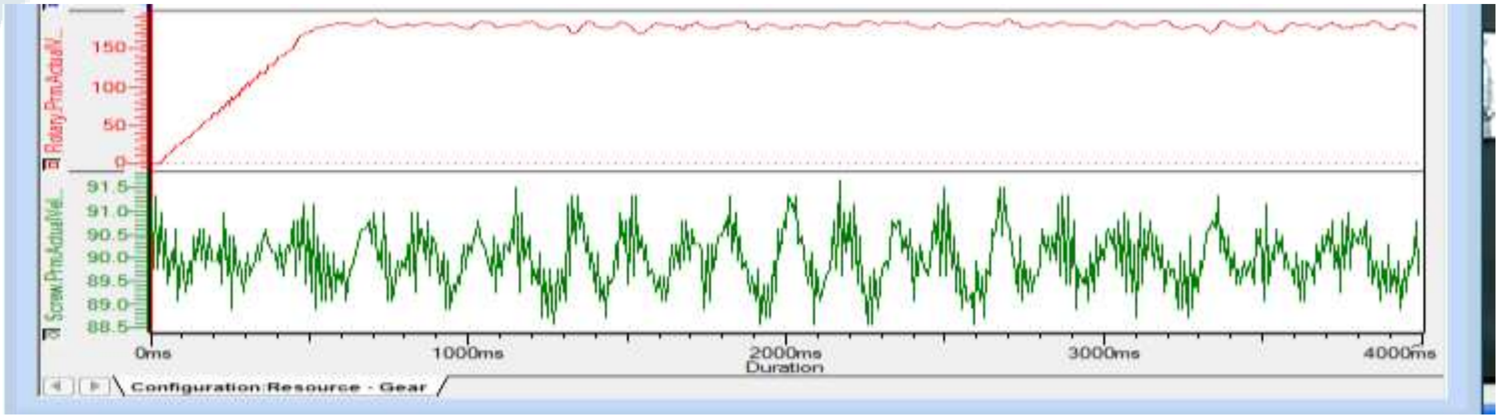
1. *What is the difference between executing MC_GearIn when the master is already moving vs when the master is stopped? Use Logic Analyzer (master speed, slave speed, InGear)*
2. *How can the gear ratio be changed without stopping the slave?*
3. *Under what conditions does the slave disengage and no 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?*

## Logic Analyzer Result #1

- *Master stopped, Gear In, Start Master*



- *Master Running, Gear In*



ENGINEERING EXPERTISE

EASY TO
WORK WITH

YASKAWA™

IT'S PERSONAL
YASKAWA

NOLOGICAL
INNOVATION

QUALITY
PRODUCT