Introduction to Yaskawa Motors, Controller, and MotionWorks

Objective
The objective of this lab is to provide an introduction to Yaskawa motor/controller hardware and its Motionworks IEC 3 software. The contents of this lab will include:
- Description and set up of motors and controller
- How to use 'Hardware Configuration' in Motionworks
- How to test motor movements
- How to create and execute simple block diagram program with physical input/output

Background
Yaskawa is a Japanese electronics manufacturing company who manufacture servos, motion controllers, AC motor drives, switches, and industrial robots. The American division, Yaskawa America, has multiple locations in California and supports the activities and engineers of Cal Poly with funding and lab materials such as the equipment for this lab session.

The system works through the communication of our computer to the controller and the controller to the motors. We are mainly interfacing with the MP2300Siec via the computer's MotionWorks software which communicates through Ethernet/IP while the MP2300Siec communicates with the motors through Yaskawa's Mechatrolink as seen in Figure 1.

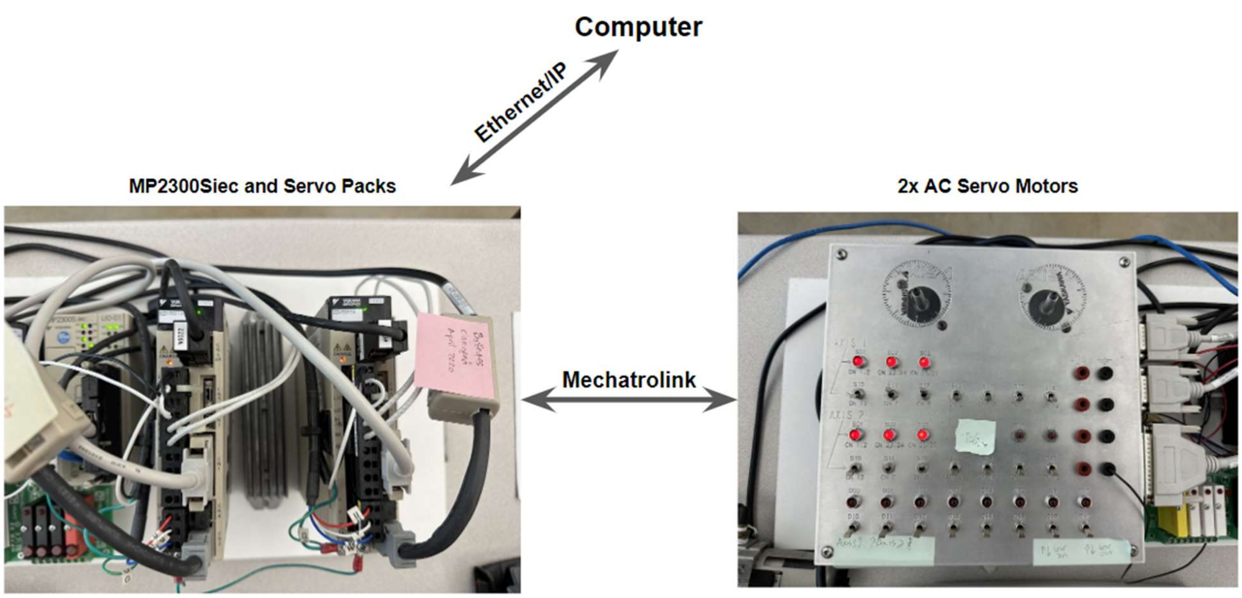

Figure 1: Integration of MPS2300Siec and Servo Motors

Step 1: Connect and Configure Controller in Motionworks
- Search for the MotionWorks IEC in the computer and launch the application. If your window looks like Figure 3 instead of Figure 2, then go to the top left and hit File>New Project and Figure 2 should open.

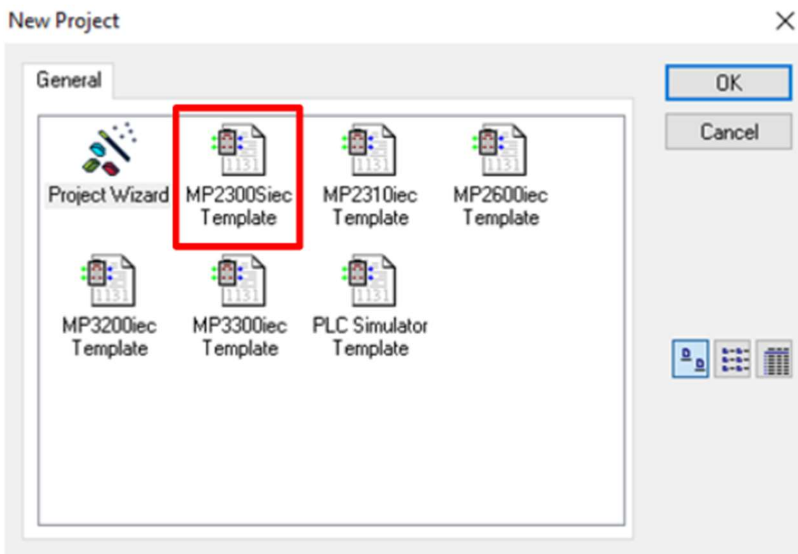

Figure 2: New Project Prompt Window

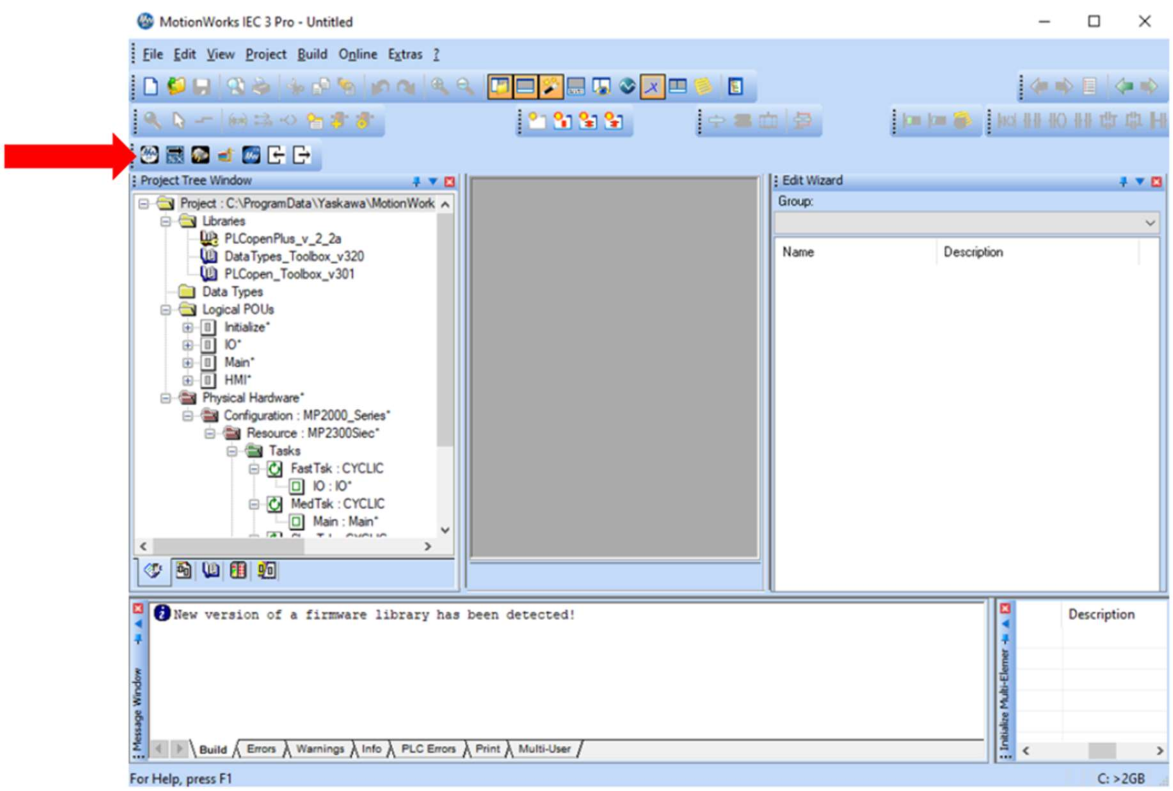

Figure 3: MotionWorks IEC Workspace with Arrow to Hardware Configuration

- Once in the Hardware Configuration Window, ensure that the IP in the top right corner matches what is shown in Figure 4. Then hit the 'Connect' button to proceed.

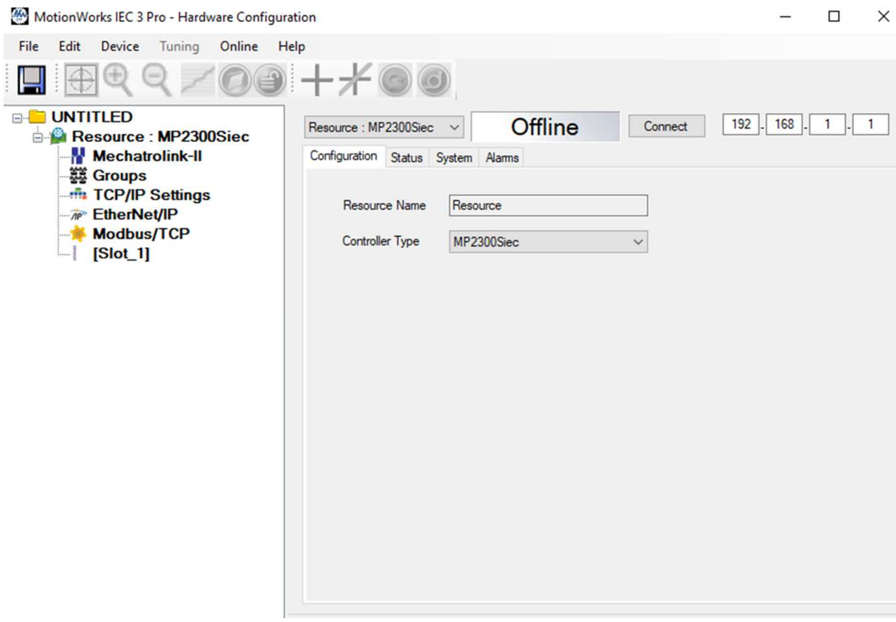

Figure 4: Hardware Configuration Window

- There should be a Startup Configuration already on the controller, so choose the Startup Configuration on the Controller as highlighted in Figure 5.

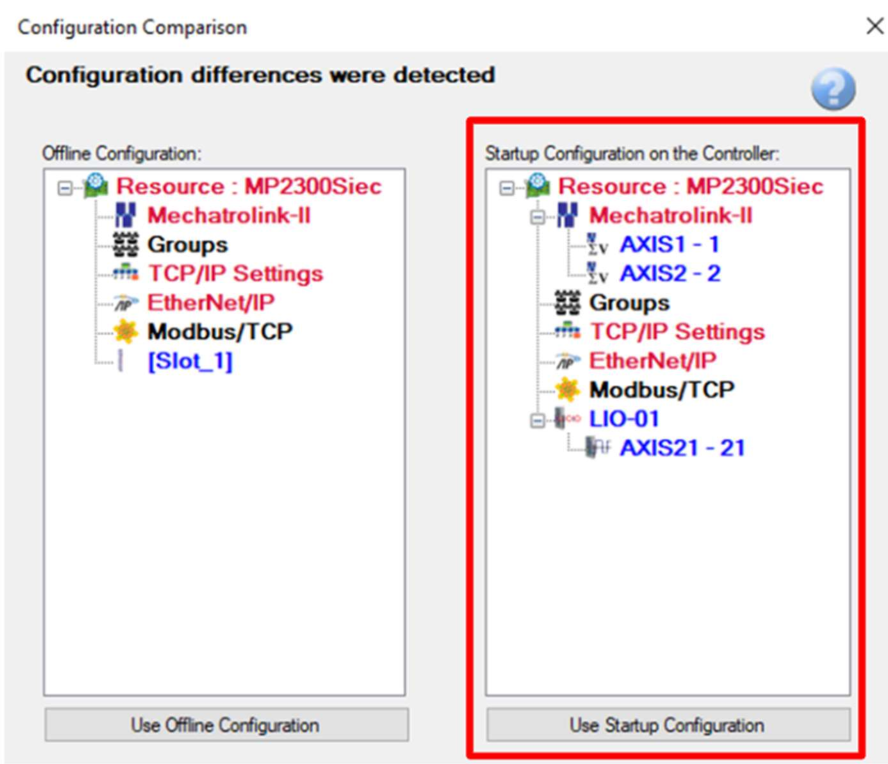

Figure 5: Configuration Comparison Window

Step 2: Test Motor Movement

- Once successfully connected to the motors, make your way to the Tuning Tab as shown in Figure 6 and click the 'Test Move Setup' Button.

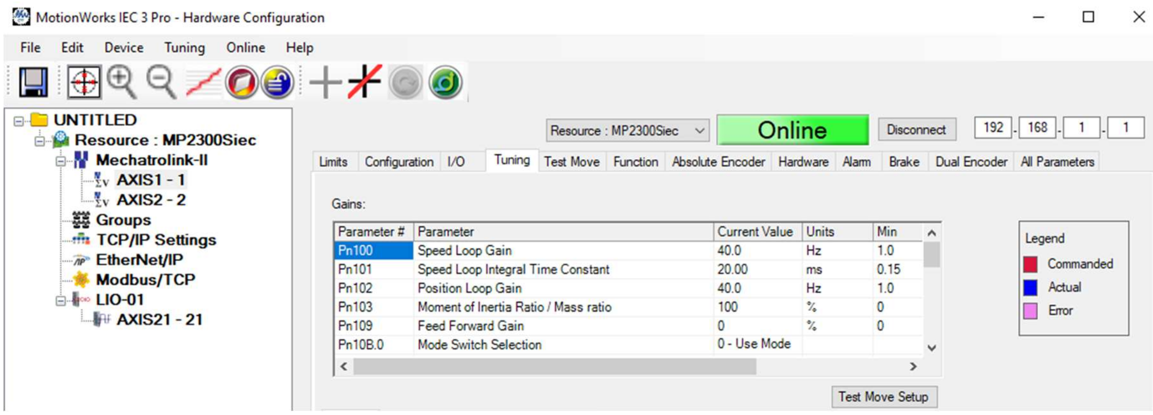

Figure 6: Tuning Tab

- Configure your Test Move Setup Window to look like Figure 7, then click start.

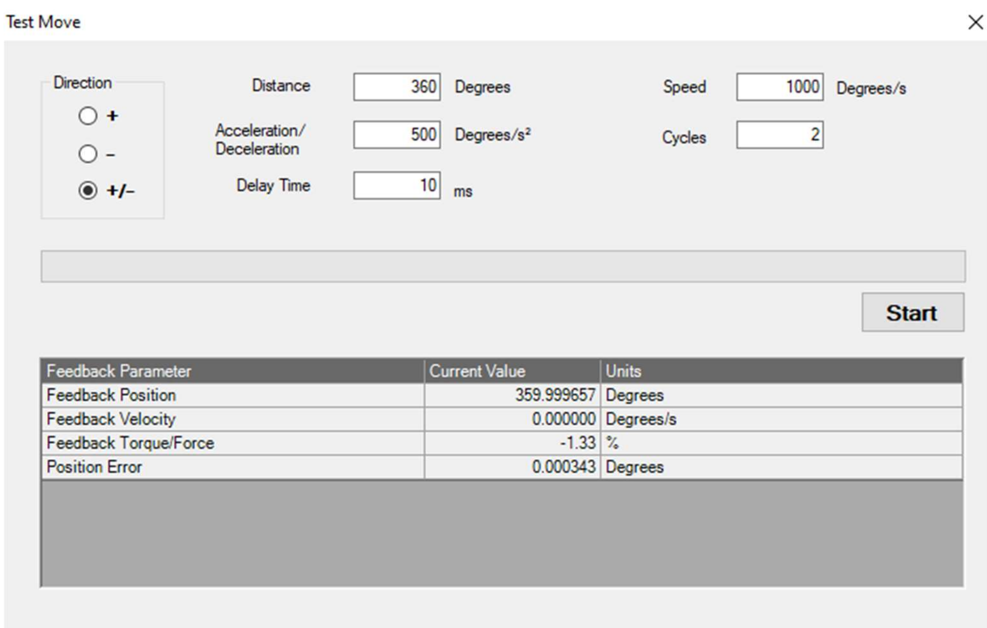

Figure 7: Test Move Setup Window

- After the test finishes and the motor stops moving, a graph similar to Figure 8 should appear. If so save the Hardware Configuration through the button in Figure 9.

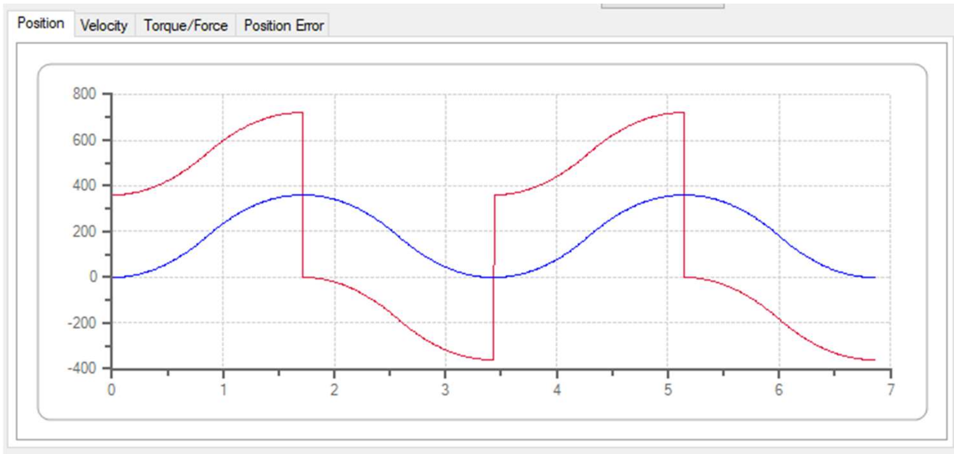

Figure 8: Resulting Position Graph from Test Move

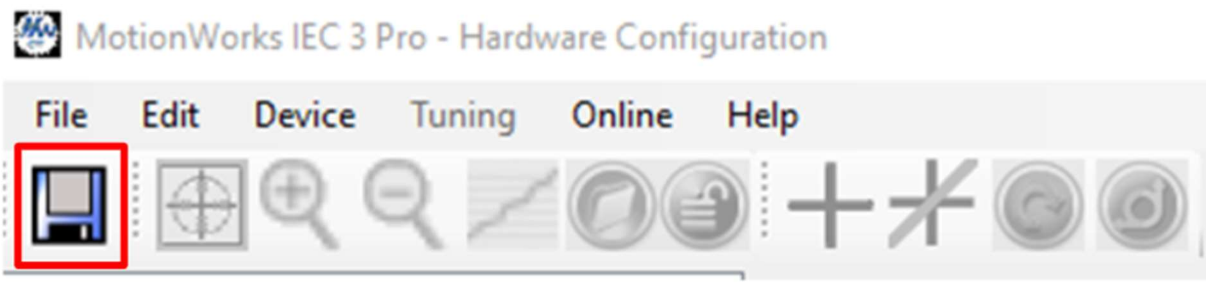

Figure 9: Save Hardware Configuration Button

Step 3: Verify Hardware Integration with MotionWorks

- Finally, check that the Hardware Configuration is saved in your main program. To do this double-click the Global Variables in the Project Tree Window as shown in Figure 10.

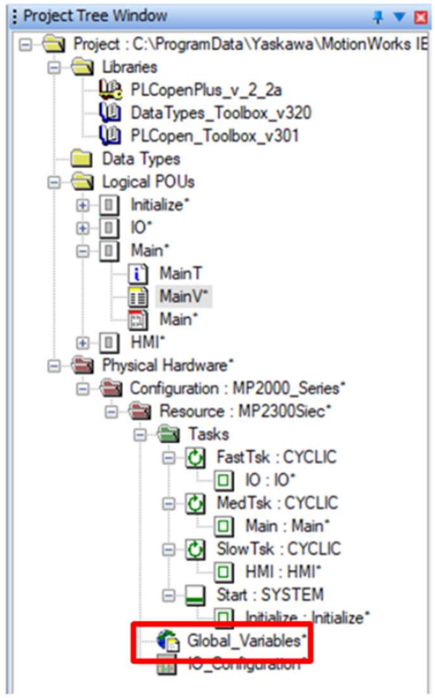

Figure 10: Project Tree Window

- Once there look for AXIS1 and AXIS 2 and they should look like they do in Figure 11

AXIS1	AXIS_REF	VAR_GLOBAL	SGDV Rotary - 1 (* Do Not Modify!! *)		(AxisNu...
⊟ AXIS2 <SGDV Rotary> - Sigma-V Rotary Servo Amplifier - 1:2 (* Modify Variable Names, Not Group Name. *)					
AXIS2_SI1_POT	BOOL	VAR_GLOBAL	POT, default on pin #7, configurable by Pn50A.3	%IX53312.0	
AXIS2_SI2_NOT	BOOL	VAR_GLOBAL	NOT, default on pin #8, configurable by Pn50B.0	%IX53312.1	
AXIS2_SI3_DEC	BOOL	VAR_GLOBAL	DEC, default on pin #9, configurable by Pn511.0	%IX53312.2	
AXIS2_SI4_EXT1	BOOL	VAR_GLOBAL	EXT1, default on pin #10, configurable by Pn511.1	%IX53312.6	
AXIS2_SI5_EXT2	BOOL	VAR_GLOBAL	EXT2, default on pin #11, configurable by Pn511.2	%IX53312.7	
AXIS2_SI6_EXT3	BOOL	VAR_GLOBAL	EXT3, default on pin #12, configurable by Pn511.3	%IX53313.0	
AXIS2_BRK	BOOL	VAR_GLOBAL	Brake Output Status	%IX53313.1	
AXIS2_HBB	BOOL	VAR_GLOBAL	HBB, Stop Signal Input	%IX53313.2	
AXIS2_SI0_IO12	BOOL	VAR_GLOBAL	Configurable by Pn81E.0, default is unallocated	%IX53313.4	
AXIS2_SI1_IO13	BOOL	VAR_GLOBAL	Configurable by Pn81E.1, default is unallocated	%IX53313.5	
AXIS2_SI2_IO14	BOOL	VAR_GLOBAL	Configurable by Pn81E.2, default is unallocated	%IX53313.6	
AXIS2_SI3_IO15	BOOL	VAR_GLOBAL	Configurable by Pn81E.3, default is unallocated	%IX53313.7	
AXIS2_ALM	BOOL	VAR_GLOBAL	Alarm On Drive	%IX53316.0	
AXIS2_WARNG	BOOL	VAR_GLOBAL	Warning On Drive	%IX53316.1	
AXIS2_SVON	BOOL	VAR_GLOBAL	Servo On	%IX53316.3	
AXIS2_PON	BOOL	VAR_GLOBAL	Main Circuit Power On	%IX53316.4	
AXIS2_PSET	BOOL	VAR_GLOBAL	Positioning Completed	%IX53316.7	
AXIS2_SO1	BOOL	VAR_GLOBAL	SO1, pins 1 and 2, configurable by Pn82E, Pn50E, Pn50F,...	%QX53312.0	
AXIS2_SO2	BOOL	VAR_GLOBAL	SO2, pins 23 and 24, configurable by Pn82E, Pn50E, Pn5...	%QX53312.1	
AXIS2_SO3	BOOL	VAR_GLOBAL	SO3, pins 25 and 26, configurable by Pn82E, Pn50E, Pn5...	%QX53312.2	
AXIS2	AXIS_REF	VAR_GLOBAL	SGDV Rotary - 2 (* Do Not Modify!! *)		(AxisNu...

Figure 11: Global Variables Window

Step 4: Simple MotionWorks Program

- Now we will develop and execute a simple MotionWorks Program with just a Count Up block.
- On the left side of the screen in the Project Tree Window, navigate to Project > Logical POUs > Main > Main and double click it as highlighted in Figure 12 to get to the correct workspace.

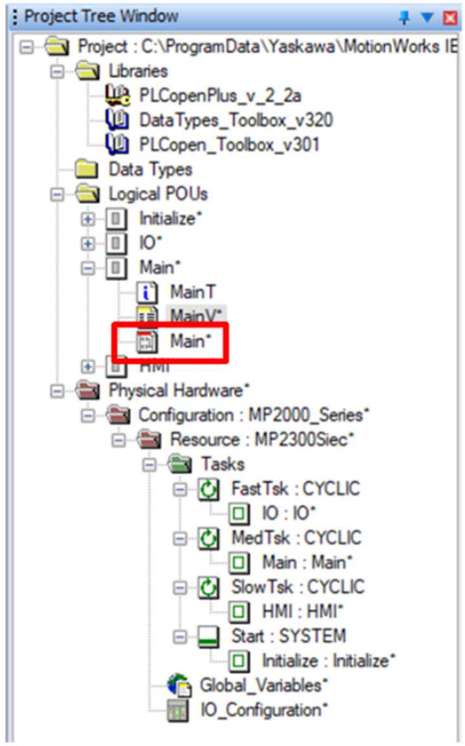

Figure 12: Navigation to Main Program Workspace

- On the right side of the screen in the **Edit Wizard**, click and drag **Counter Up (CTU)** out onto the blank workspace in the middle of the window as highlighted in Figure 13.
- The window in Figure 14 should pop up but we can ignore it and just hit OK.

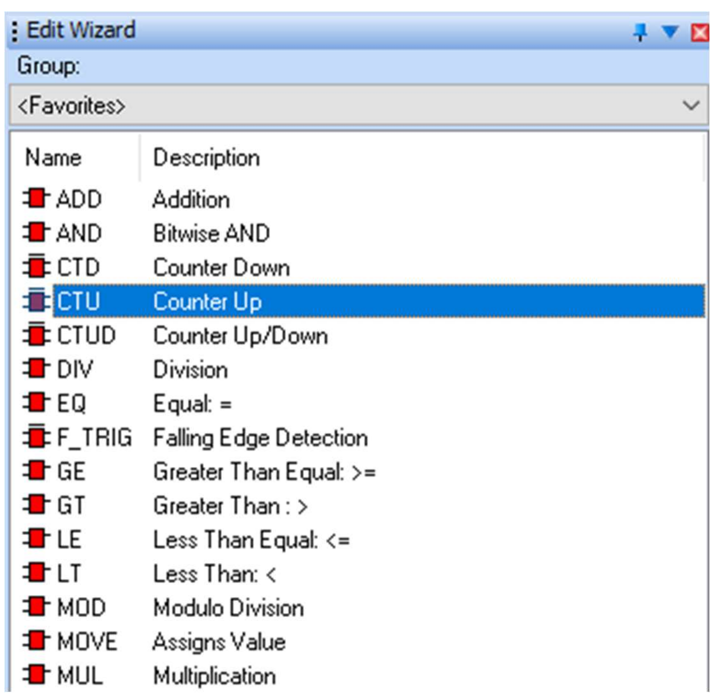

Figure 13: Counter Up Navigation in Edit Wizard

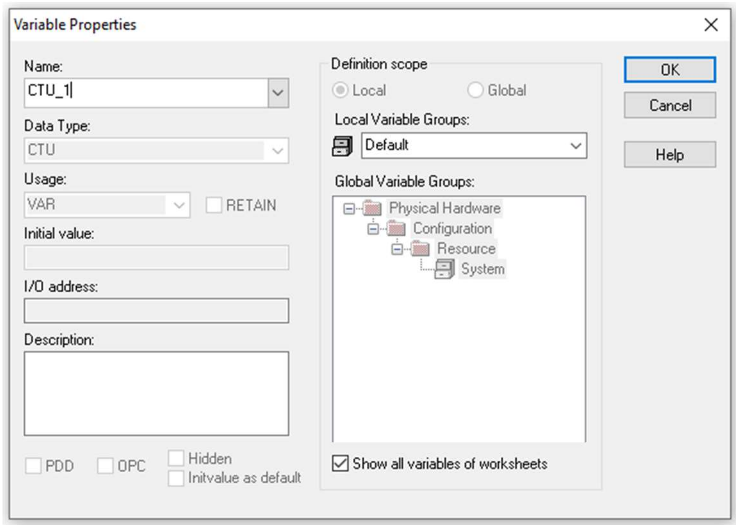

Figure 14: Counter Up Variable Properties

- Double-click ‘CU’ on the CTU block and name it so that it matches with Figure 15 and repeat the process for all the variables to match with Figure 17.
- The PV Variable does have an initial value so make sure PV Variable Settings match Figure 16.

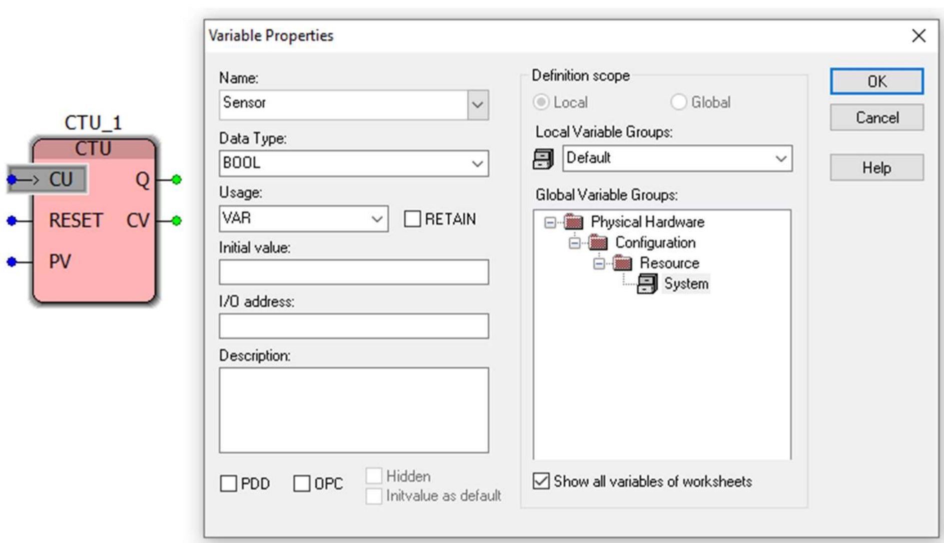

Figure 15: CTU Count Up (CU) Variable Properties

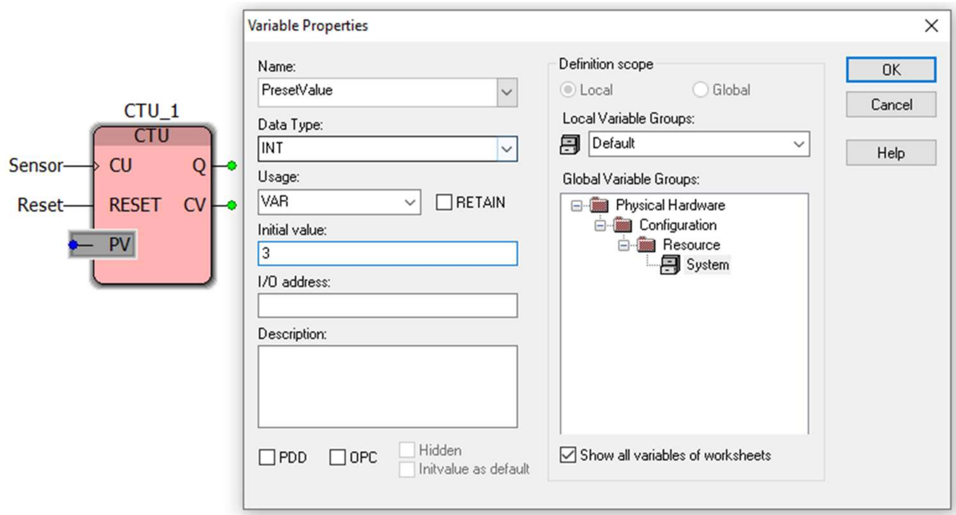

Figure 16: CTU Preset Value (PV) Special Variable Properties

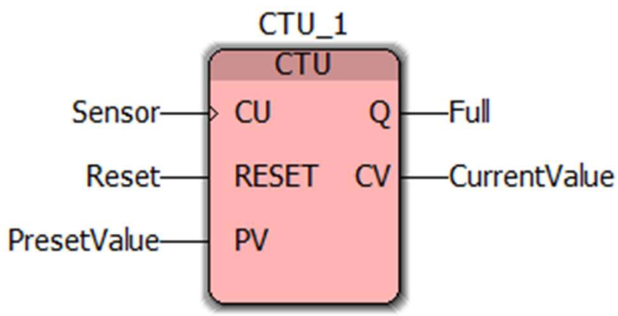

Figure 17: CTU Current Value (CV) Variable Properties

- Once you are done configuring all the variables and the block matches Figure 17, click ‘Download Changes’ which is shown in Figure 18.

Figure 18: Arrow to “Download Changes”

- After a few seconds the window on the left of Figure 19 should appear, hit Download and the window on the right of Figure 19 should appear. Hit Download again.
- After downloading, the window on the left of Figure 20 should appear; hit warm and the window should change to look like the window on the right of Figure 20 and the program is ready to run.

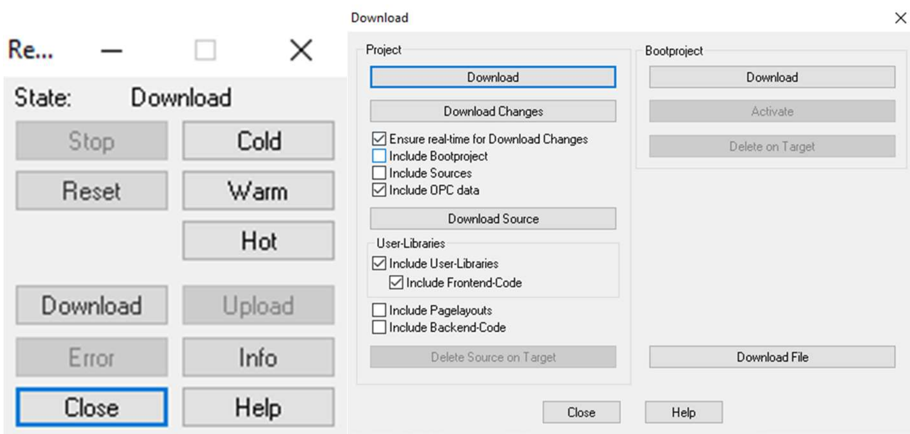

Figure 19: Download and Run Series 1

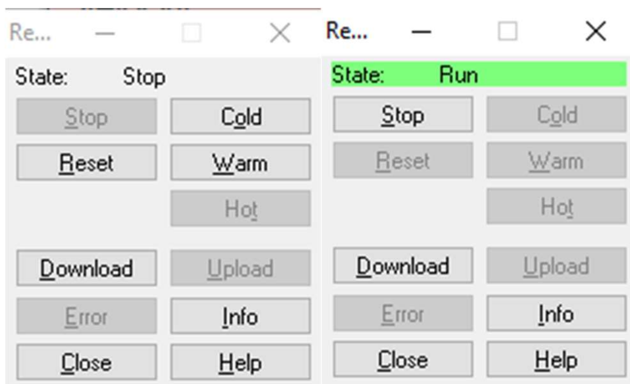

Figure 20: Download and Run Series 2

- Enter Debug Mode clicking the Debug Toggle as shown in Figure 21 and it should look like Figure 22

Figure 21: Arrow to "Debug Toggle"

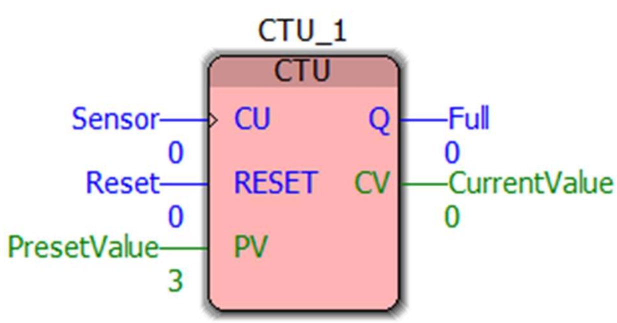

Figure 22: Program in Debug Mode

- Double-click 'Sensor' and once in the window in Figure 23, hit Overwrite to change the value so that the blocks look like they do in Figure 24.

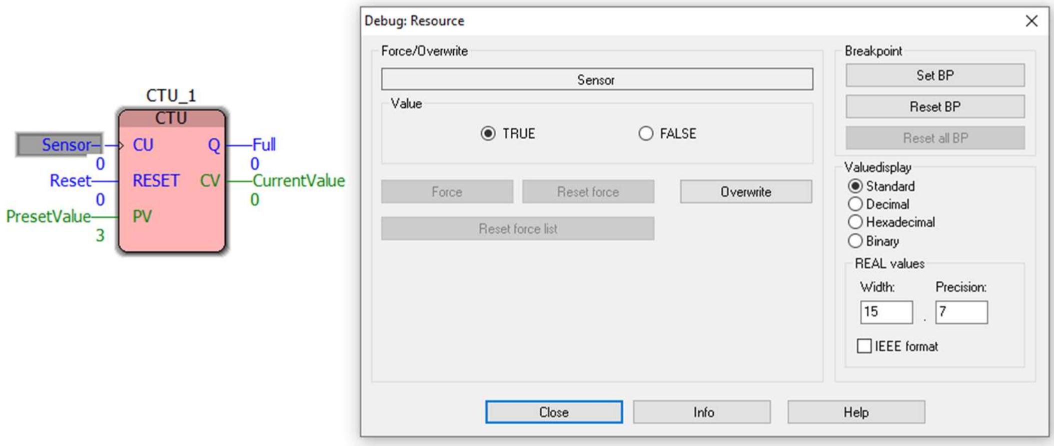

Figure 23: Sensor Execute Overwrite

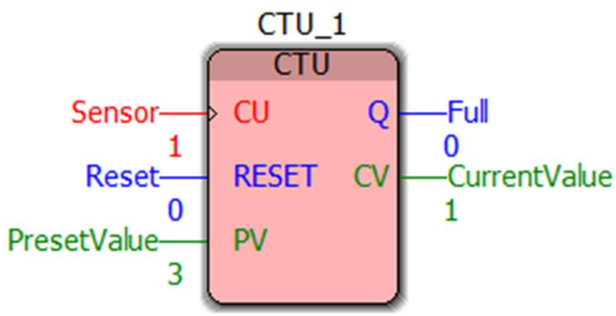

Figure 24: CTU Block after Sensor Overwrite

- Repeat the overwrite of the sensor until the Counter block is full and resembles Figure 25, then show to the instructor.

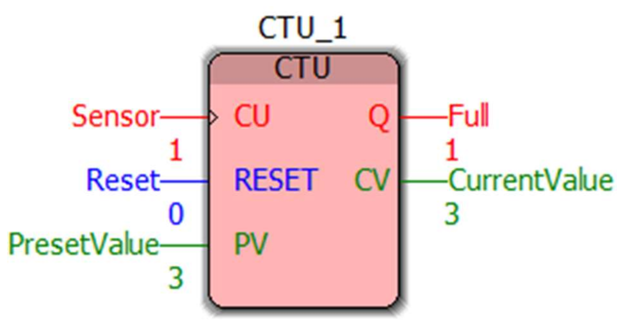

Figure 25: CTU Block Full

Step 5: Adding Physical Inputs/Outputs

- Since we now know how the function block operates, we can add physical inputs to the function block.
- Be sure to take the program out of Debug Mode and click on the Ladder Logic button at the top as shown in Figure 26 and the program should look like Figure 27.

Figure 26: Insert Ladder Logic Button

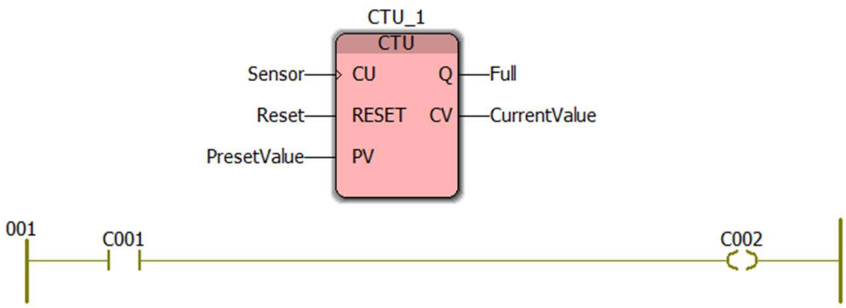

Figure 27: Ladder Logic + Block Diagram

- Similar to the Rockwell software for the PLC's you have been using, the inputs are already linked as variables such as 'MO1_DI_00', DI meaning input and 00 meaning the reference number.
- Clicking on C001 and typing 'MO1_DI_00' as the name, should autofill the I/O address as seen in Figure 28.

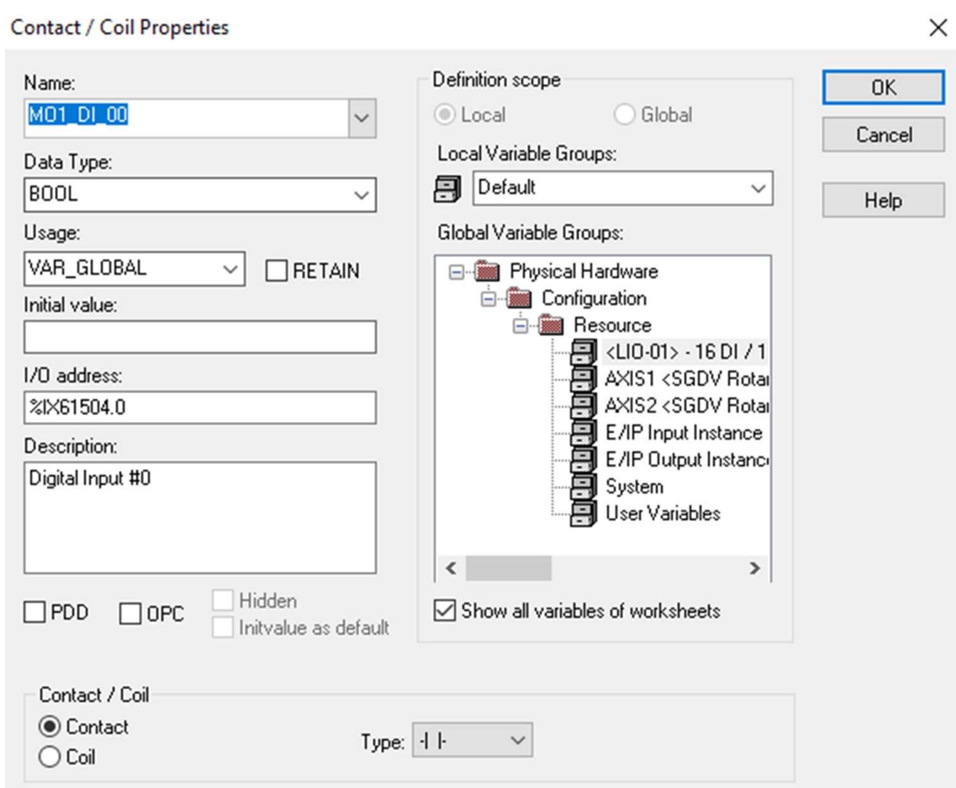

Figure 28: Example Change in Contact Properties

- Using your prerequisite knowledge of ladder logic from the Rockwell software and using the buttons in the same row as the Insert ladder logic button from Figure 26, make the program look like Figure 29

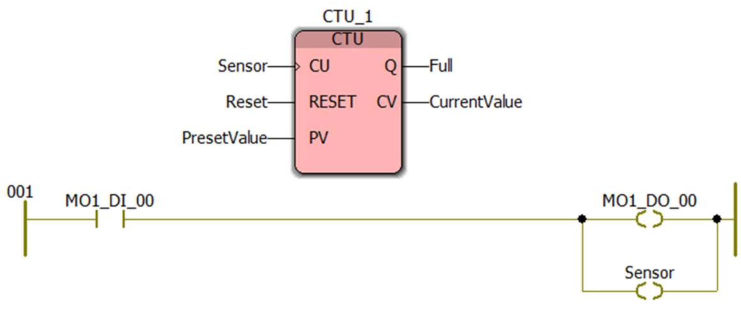

Figure 29: Final Product with Input Button and Output Light for 'Sensor'

- Once completed, ensure that all the switches are off (pointed down). Then Download Changes and switch to Debug Mode and it should look like Figure 30.
- Toggling the DI_00 switch should make the program look like Figure 31 as well as turn on the light above the DI_00 switch labeled DO_00.

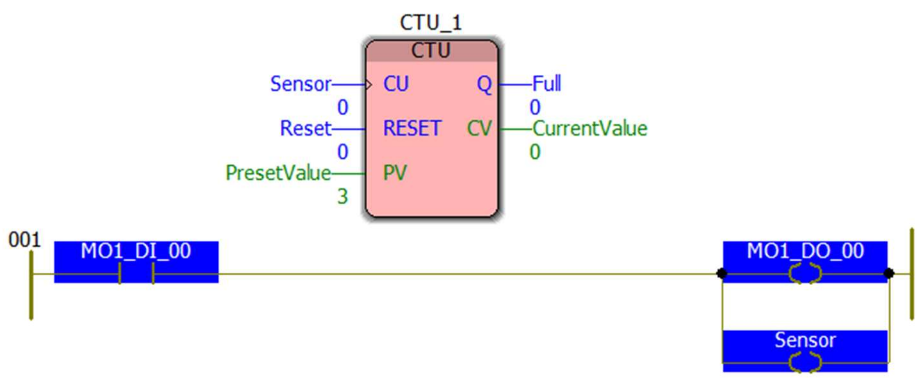

Figure 30: Program in Debug Mode and Button Switched Off

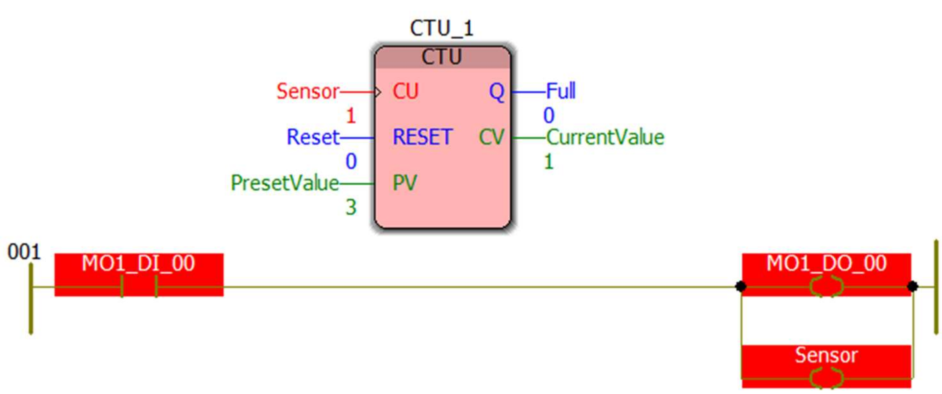

Figure 31: Program in Debug Mode and Button Switched On

Challenge: Completing the System

Systems are better understood with all significant inputs and outputs in the physical realm. Complete the system from Step 5 by:

- Adding an input button and output light for ‘Reset’
- Adding an output light for ‘Full’
- Adding output lights to show ‘CurrentValue’ (i.e. if ‘CurrentValue’ = 1 have 1 light on and if ‘CurrentValue’ = 2, have 2 lights on etc.)

Deliverables

Demonstrate challenge to the lab instructor.