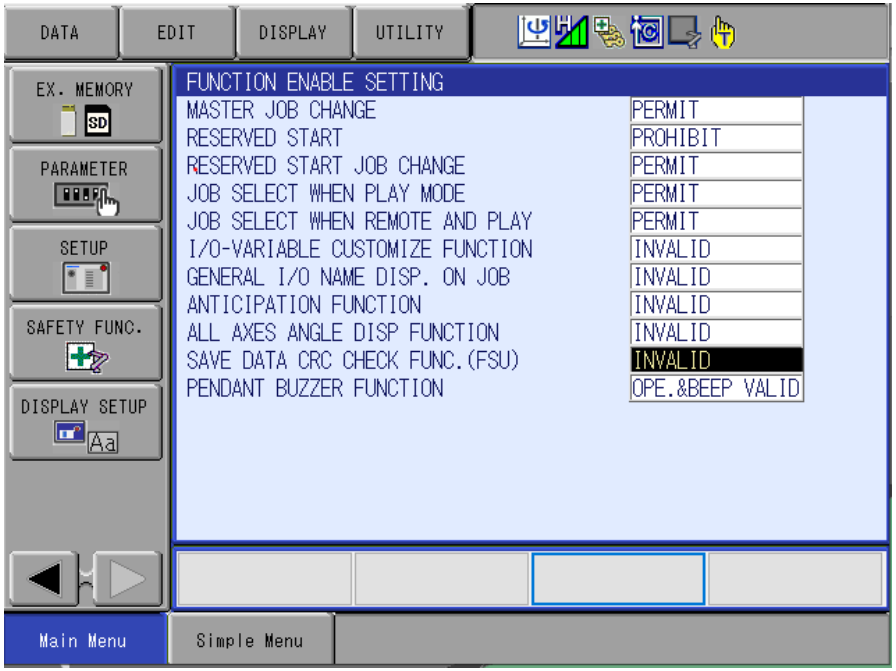
YASKAWA
PARTNER SUPPORT
Shared Integration Experience

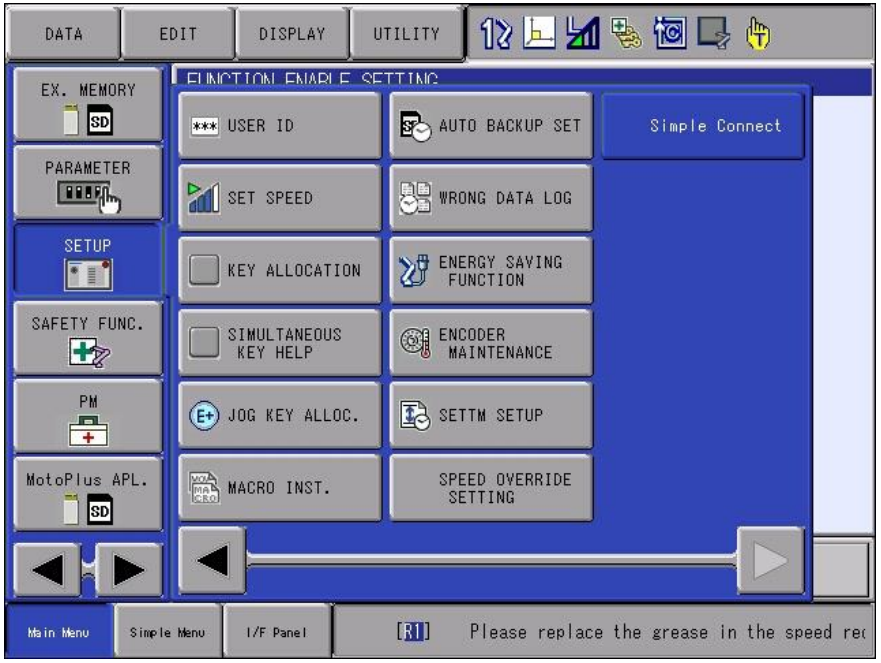
This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

Enabling YAI Standard PLC Map Via Simple Connect

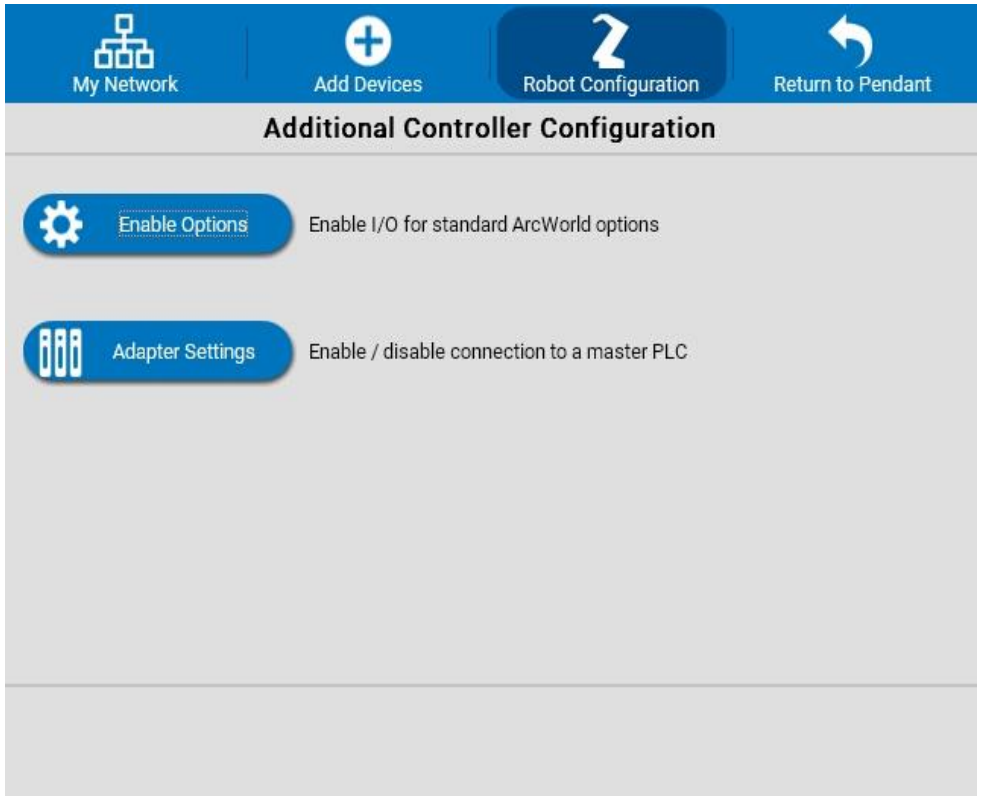
YRC1000 arc welding applications are equipped with a feature known as Simple Connect. Simple Connect is a combination of hardware and software allowing for easy installation and configuration of common robotic devices and accessories with minimal effort and time. This document will guide you through enabling a PLC map VIA the Simple Connect Interface. This option allows for an easy experience configuring a PLC as a master to the Yaskawa YRC1000 Controller acting as the slave. This also will provide standard signaling without the hassle of remapping the Concurrent IO.

1. If the FSU function is enabled ensure you are in-SAFETY mode and the CRC check is set to INVALID. Then navigate to the Simple Connect icon.

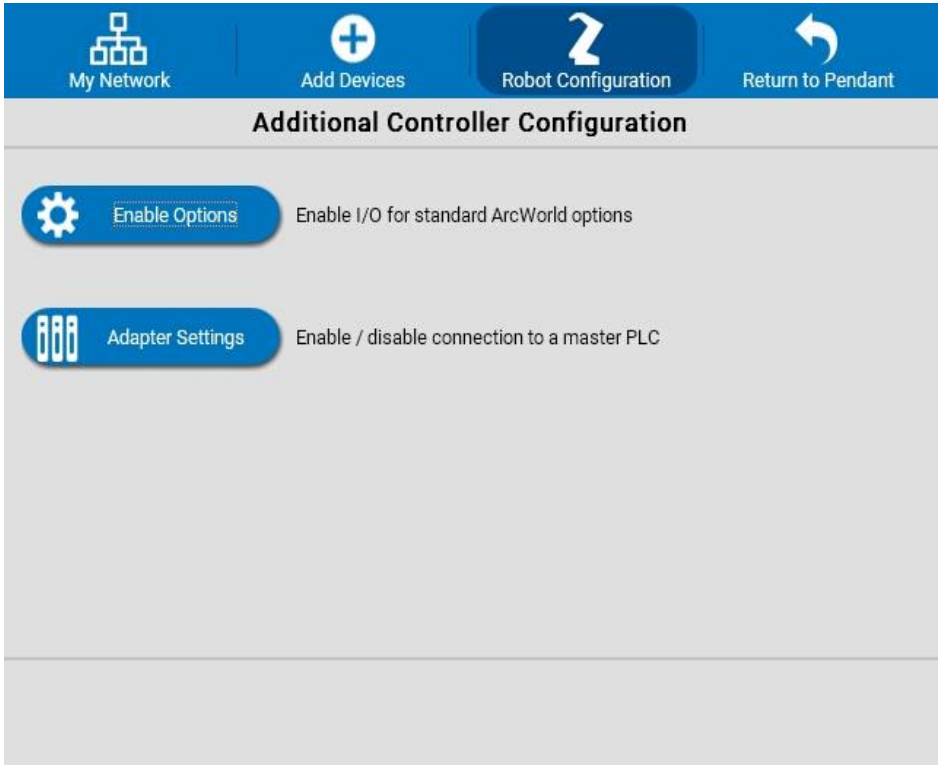


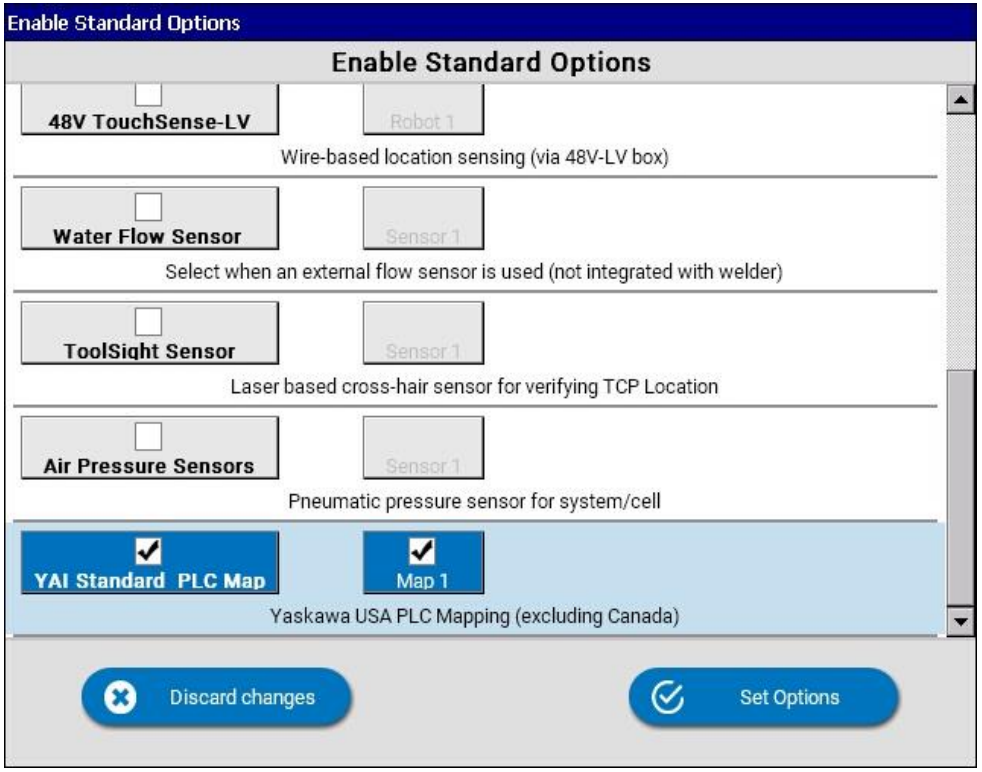


2. Once the application has launched, select the robot configuration tab.

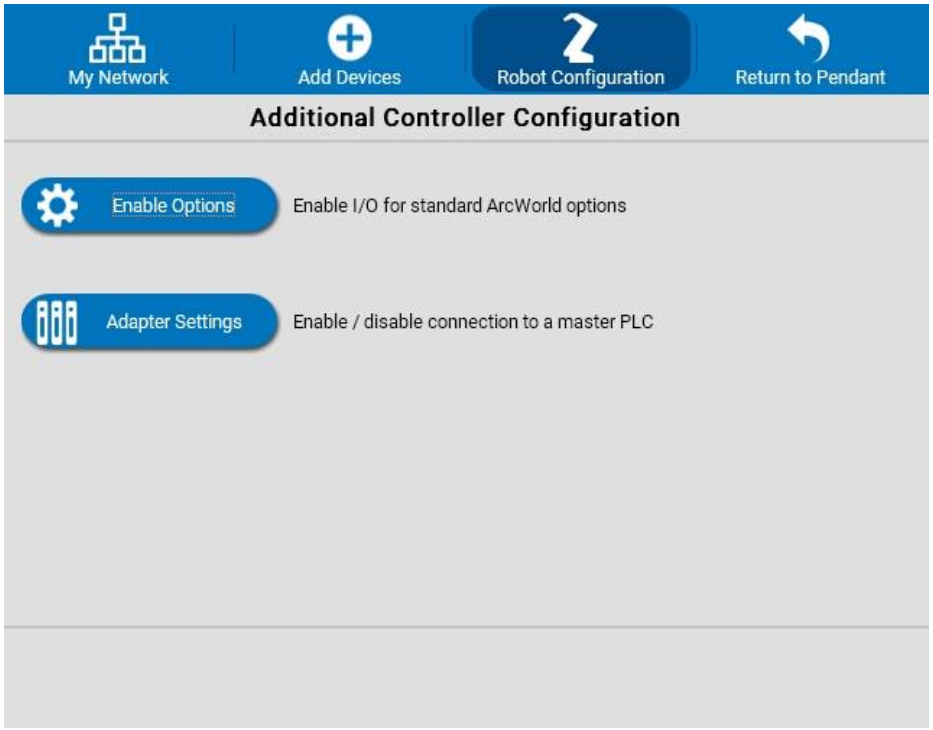


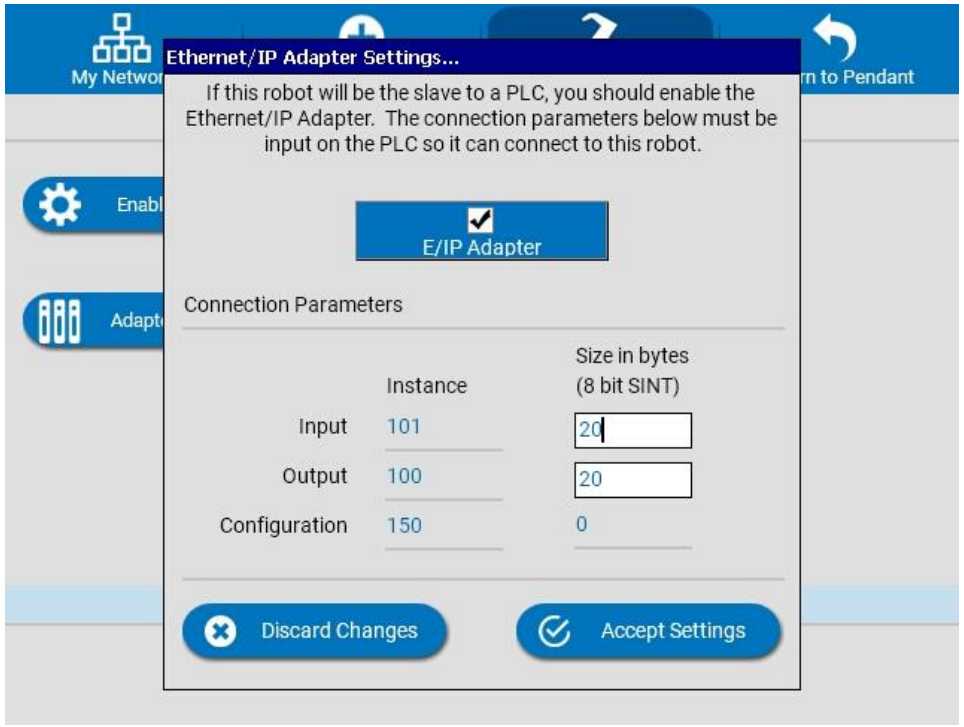
3. Within the robot configuration tab select enable options and enable the YAI Standard PLC Map and select the set options tab. This sets an activation bit VIA the Concurrent IO.





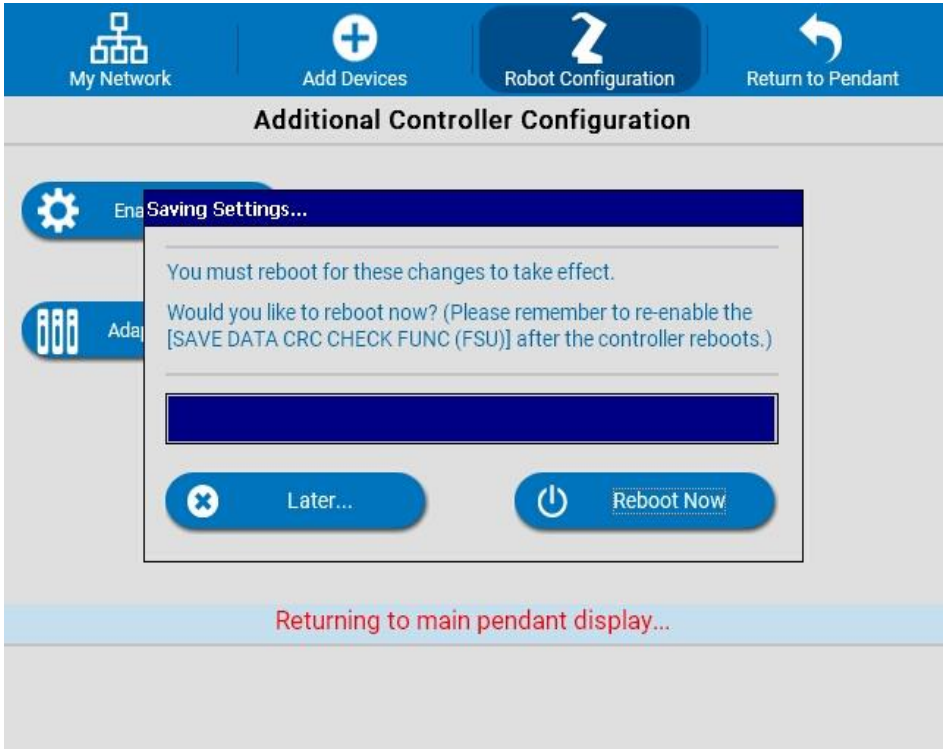
4. Select the adapter settings tab. Within the settings tab, the only thing that is configurable is the IO byte size. Your PLC must match the instance configuration. ***Set accordingly: robot's input are the PLC's outputs, robot's outputs are the PLC's Inputs.***

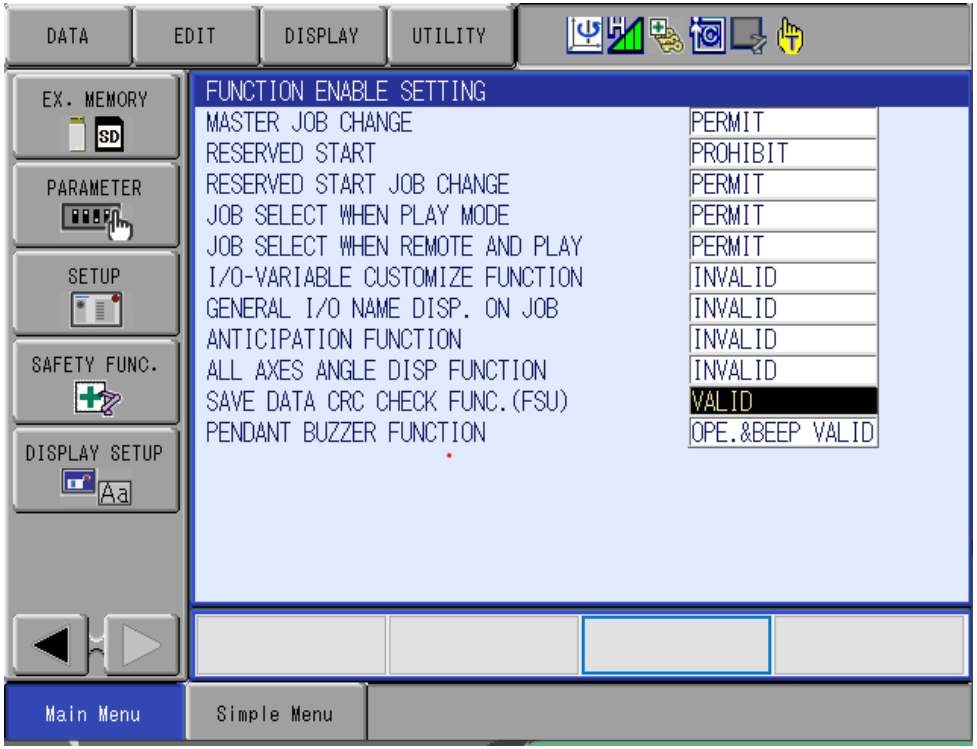




NOTE → *Up to 72 bytes could be allocated in the EtherNet/IP adapter setting. This guide will utilize 20 bytes for all examples. The techniques to increase the PLC allocation beyond 20 bytes are left to the user. Change the PLC connection and expanding the UDT would be easy following the examples provided.*

5. After accepting the adapter settings, select return to pendant. This will accept and save changes, then prompts for a controller reboot. Also remember to set the CRC check back to valid upon reboot.



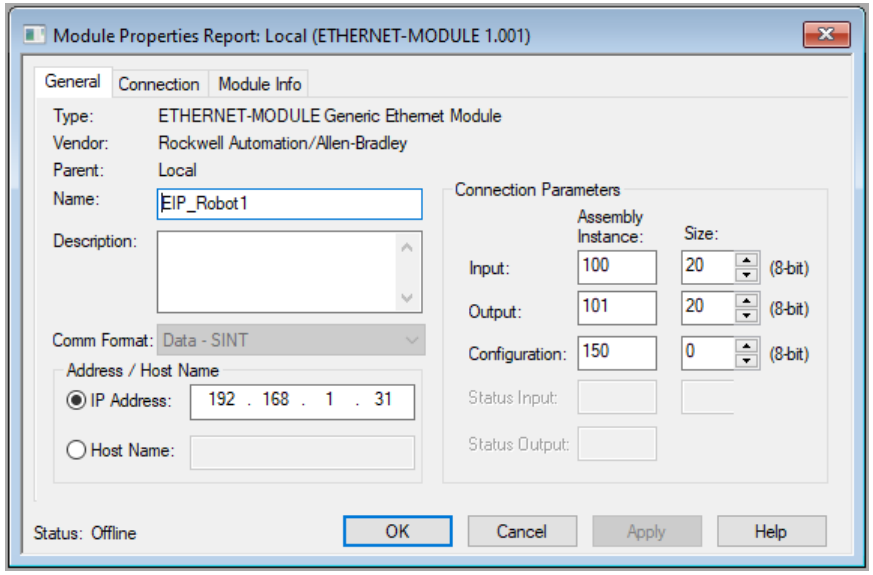


Settings for Rockwell PLC

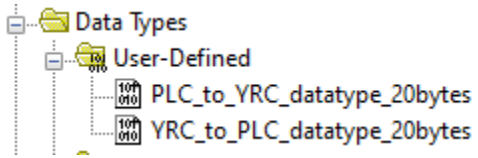
Obtain the file [YRC1000_ArcWorld_PLC_Map_Example_rev0.ACD] from Yaskawa, typically the file will be packaged along with this guide. The file will have the correct settings to match the YRC1000 controller settings established earlier. Additionally, this file will include user defined data types (UDTs) which will provide fully commented data and data with friendly names. The steps below can be skipped if the file was provided. The steps below provide an overview of the functionality in this PLC project.

Verify the Generic Adapter Settings:

Change the IP address if your robot controller is not default → 192.168.1.31



Notice 2 User Defined Data Types are defined:

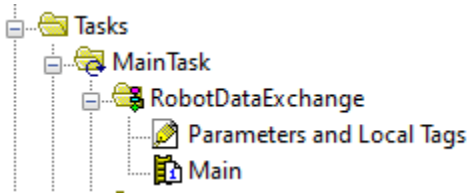


This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

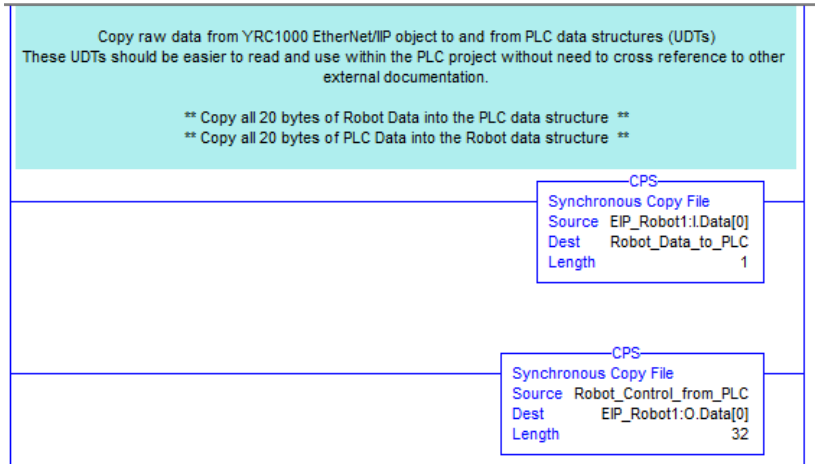
Both data types are used to define Controller scoped Tags (variables):

Name	Alias For	Base Tag	Data Type
Robot_Control_from_PLC			PLC_to_YRC_datatype_20bytes
Robot_Data_to_PLC			YRC_to_PLC_datatype_20bytes

Next, this solution maps the generic adapter data into these variables by ladder logic.



The ladder logic is simple, two Copy statements. These must be executed all the time or at a rate corresponding to the RPI.



The UDT structures provide mapping which correspond to the Excel spreadsheet [YRC_Master_resource-2021-03-12-Standard_PLC.xlsm] which is also available with this guide.

- Robot_Control_from_PLC
  - Robot_Control_from_PLC.External_Start
  - Robot_Control_from_PLC.Ext_Servo_On
  - Robot_Control_from_PLC.Ext_Servo_Off
  - Robot_Control_from_PLC.Call_Master_Job
  - Robot_Control_from_PLC.Alarm_Reset
  - Robot_Control_from_PLC.avail_byte00_bit5
  - Robot_Control_from_PLC.avail_byte00_bit6
  - Robot_Control_from_PLC.avail_byte00_bit7
  - Robot_Control_from_PLC.Go_Home
  - Robot_Control_from_PLC.Ext_Hold
  - Robot_Control_from_PLC.Engage_Safety_Speed
  - Robot_Control_from_PLC.avail_byte01_bit3
  - Robot_Control_from_PLC.avail_byte01_bit4
  - Robot_Control_from_PLC.avail_byte01_bit5
  - Robot_Control_from_PLC.avail_byte01_bit6
  - Robot_Control_from_PLC.Check_Mode_Select

- Robot_Data_to_PLC
  - Robot_Data_to_PLC.Operating
  - Robot_Data_to_PLC.Servos_are_On
  - Robot_Data_to_PLC.Servos_are_Off
  - Robot_Data_to_PLC.At_Top_Default_Job
  - Robot_Data_to_PLC.Alarm_Occurred
  - Robot_Data_to_PLC.In_Remote_Mode
  - Robot_Data_to_PLC.In_Play_Mode
  - Robot_Data_to_PLC.In_Teach_Mode
  - Robot_Data_to_PLC.At_Home_for_Shutdown
  - Robot_Data_to_PLC.In_Hold
  - Robot_Data_to_PLC.In_Safety_Speed
  - Robot_Data_to_PLC.Safety_Circuit_Closed
  - Robot_Data_to_PLC.External_EStop_OK
  - Robot_Data_to_PLC.Programming_Pendant_EStop_OK
  - Robot_Data_to_PLC.Manipulator_was_Jogged
  - Robot_Data_to_PLC.Check_Mode

uIn#	IG#	IGH#	Contact	EX I/O Name
1001	126	251	21280	PLC: Ext Start
1002	126	251	21281	PLC: Ext Servo On
1003	126	251	21282	PLC:Ext Servo Off
1004	126	251	21283	PLC:Call Mst JBI
1005	126	252	21284	PLC:Ext Alarm Rst
1006	126	252	21285	PLC:
1007	126	252	21286	PLC:
1008	126	252	21287	PLC:
1009	127	253	21290	PLC: Go Home
1010	127	253	21291	PLC:Ext Hold

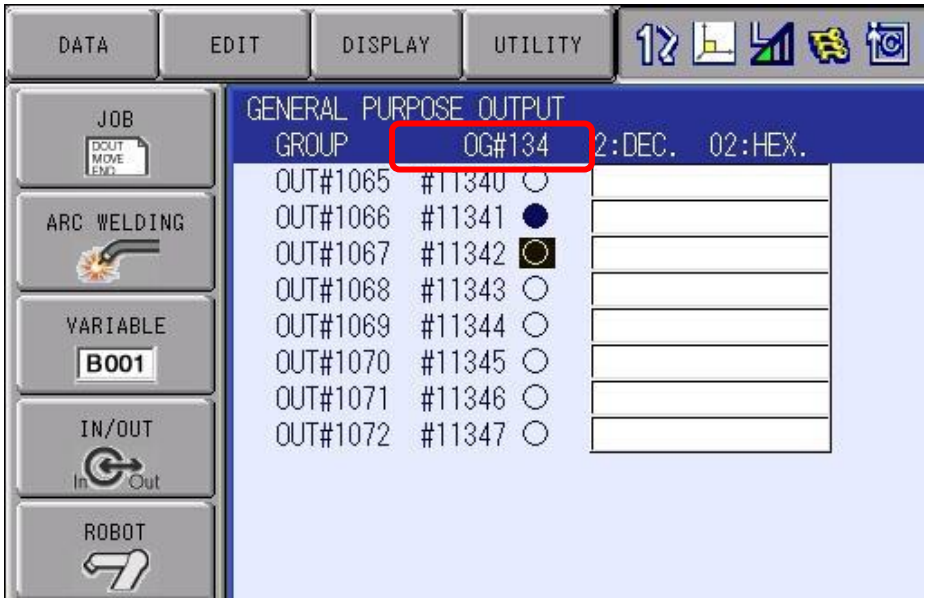
uOut#	OG#	OGH#	Contact	I/O Name
1001	126	251	11260	EIP STD PLC: Operating
1002	126	251	11261	EIP STD PLC: Servos On
1003	126	251	11262	EIP STD PLC: Servos Off
1004	126	251	11263	EIP STD PLC: At Top of Master Job
1005	126	252	11264	EIP STD PLC: Alarm Occurred
1006	126	252	11265	EIP STD PLC: Remote Mode
1007	126	252	11266	EIP STD PLC: Play Mode
1008	126	252	11267	EIP STD PLC: Teach Mode
1009	127	253	11270	EIP STD PLC: At Home for Shutdown
1010	127	253	11271	EIP STD PLC: Hold

Testing PLC and YRC1000 Communication

A Task/Ladder is included in the PLC project file [YRC1000_ArcWorld_PLC_Map_Example_rev0.ACD]. The following screenshots use this logic.

- **NOTE: This Task/Ladder should be deactivated when actual cell logic is ready to be written.**

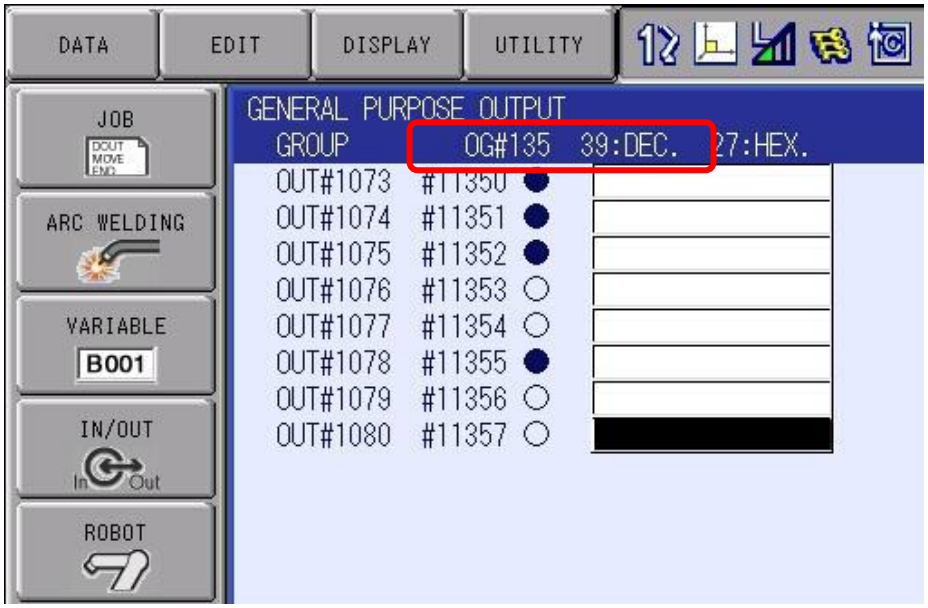
Output from Robot Controller

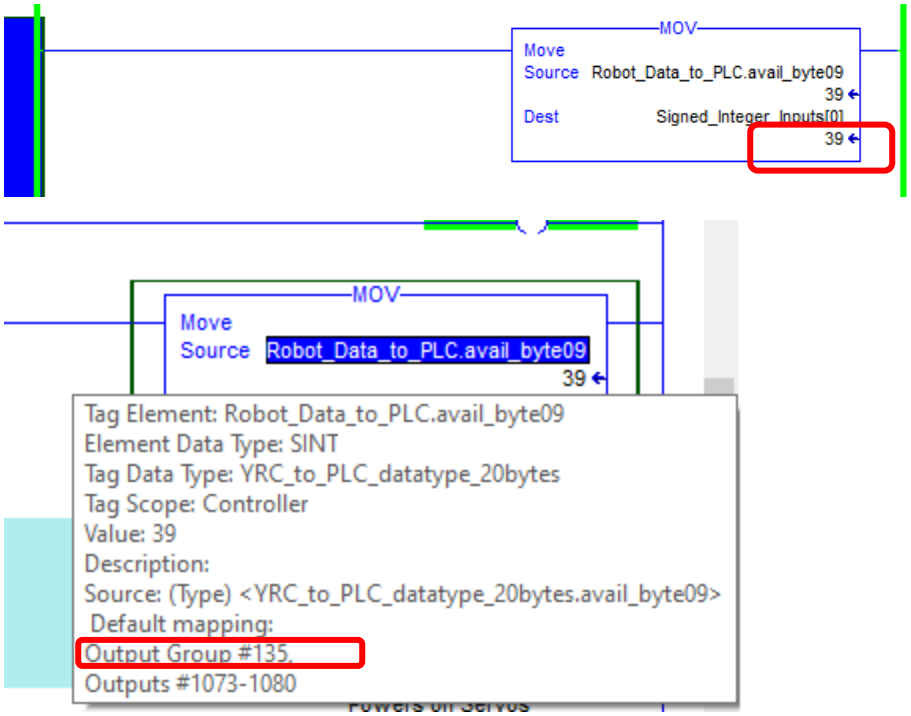


Input to PLC

The comment in the PLC shows that this bit came from Output Group (OG) #134. Looking to the left we can immediately see this correlation. Bit 1 is active. Rockwell PLC starts array counting with 0.

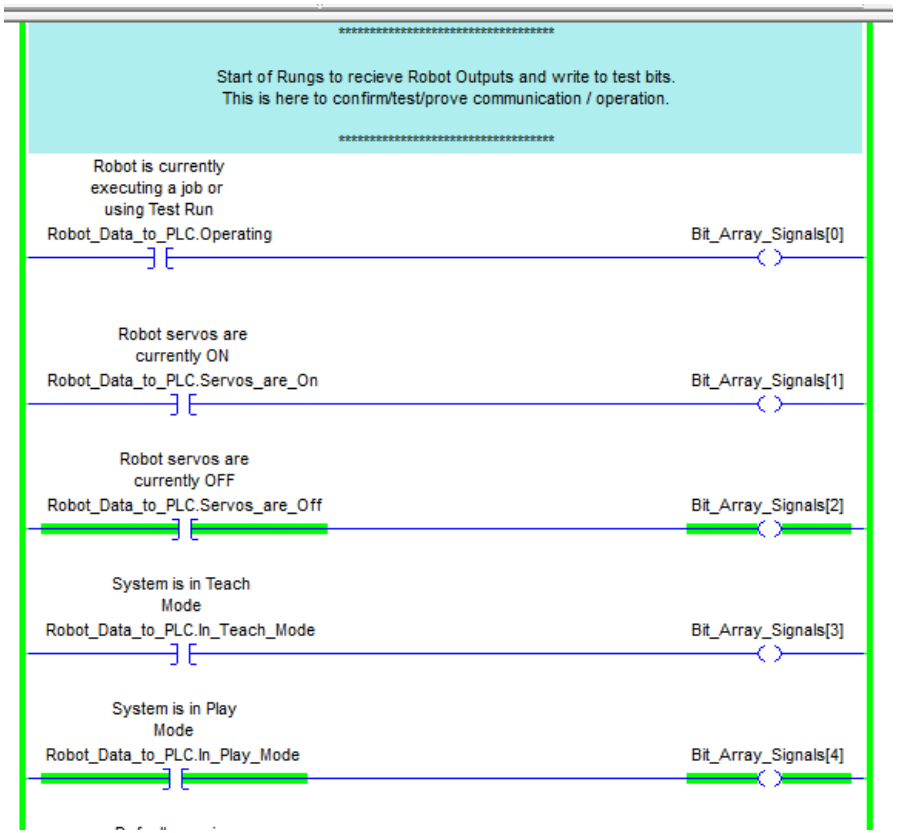
This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.





To see the correlation of the robot output to the PLC usage, use the "mouse over" to get the dialog to show up. In this dialog we see the comment displays Output Group (OG) #135.

This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.



These are specific control signals which are pre-mapped in the concurrent IO ladder. Your system may not have NAME's entered in these fields. Use the Excel data to review the correlation. The intent of this is to show output status from the robot mapped into the PLC. Using the signals at the PLC side is very intuitive due to the UDT structure and naming.

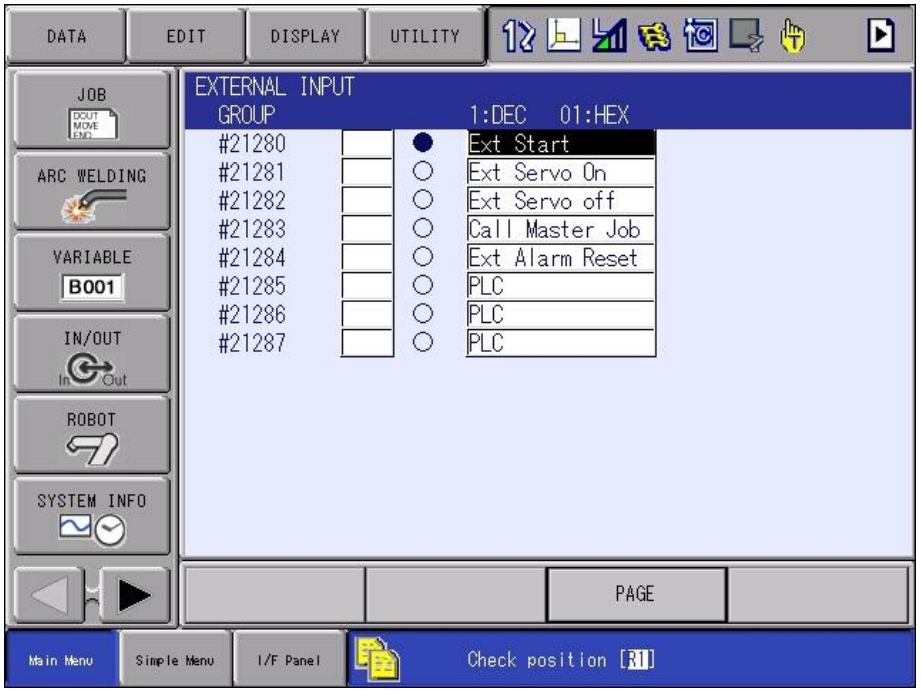
This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

Output from PLC

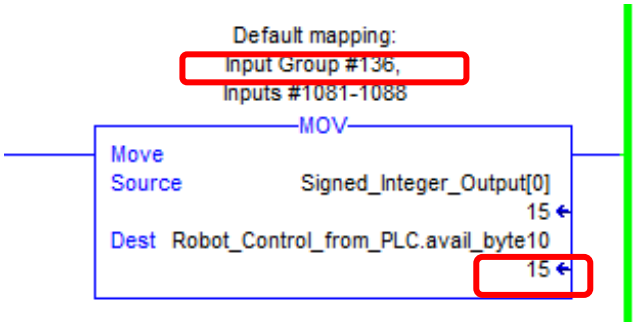
Using specifical control output signals the PLC can control the YRC1000 robot controller into certain modes. In this example the PLC is requesting the robot to start a job. (Other conditions may be necessary – servo on, play, remote mode, etc.). It will be up to the PLC programmer to check those conditions, which can be done by combining PLC inputs, status signals and other logic.

Your system may not have NAME's entered in these fields, use the Excel data to review the correlation.

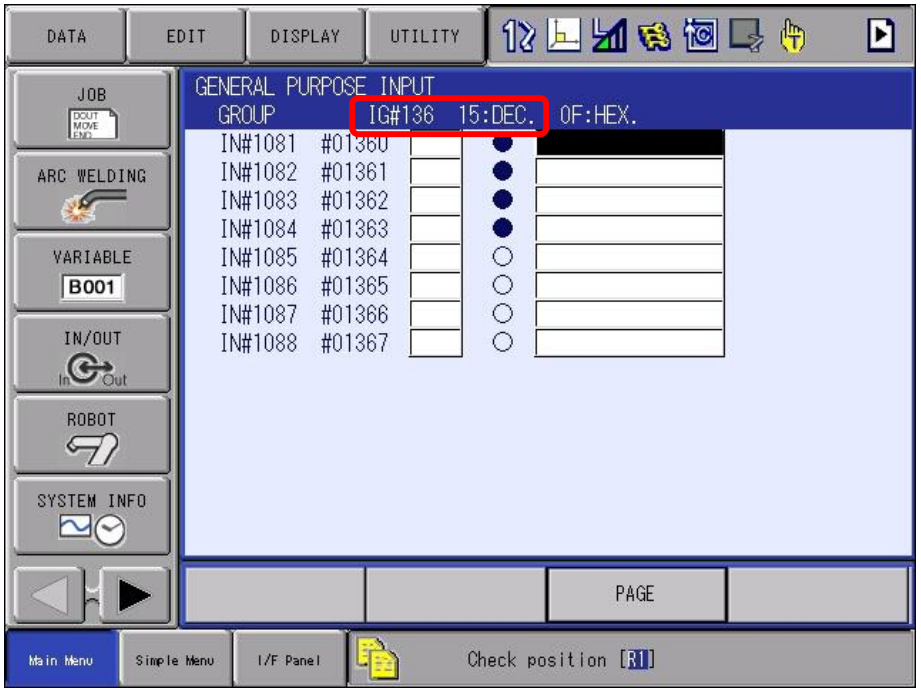
Input to Robot Controller



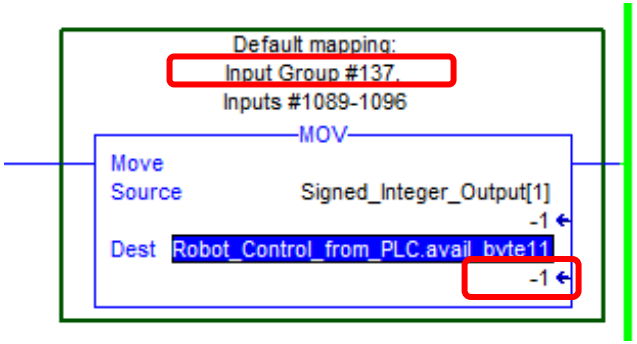
This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.



The comment in the PLC shows that this byte maps to the Input Group (IG) #136. Looking to the left we can immediately see this correlation. The total value is also 15.



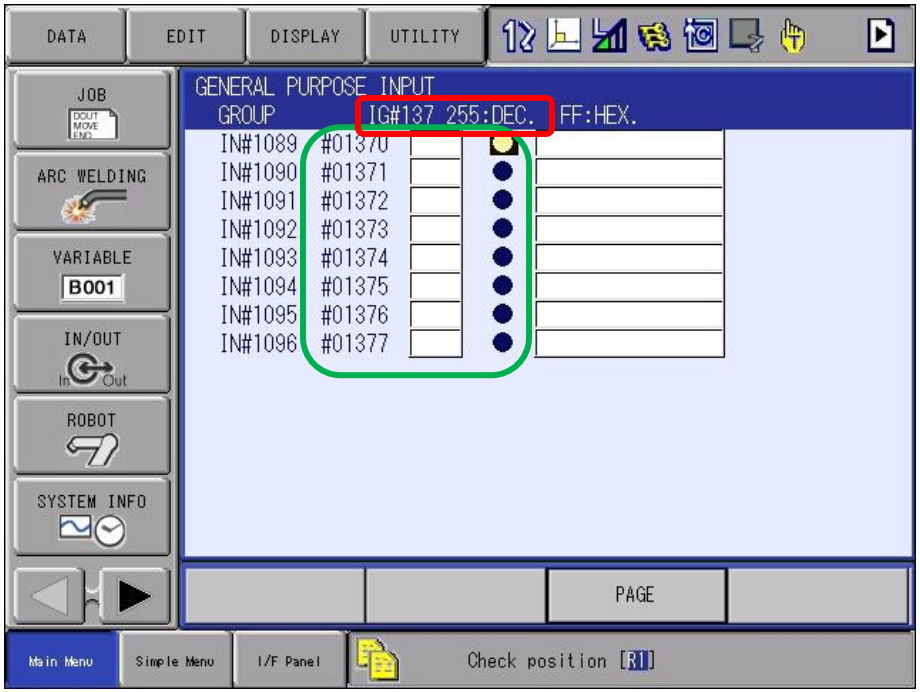
This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.



The comment in the PLC shows that this byte maps to the Input Group (IG) #137. Since this is a negative number, we need extra consideration.

The PLC represents the 8-bit number as a SINT – signed integer. However, the YRC1000 represents the 8-bit number as a bit INT (no negative). A negative number uses "2s complement" formatting. Below shows the PLC's bit-wise representation. This value directly matches the 8 bits in the YRC1000, although the total value is different due to "signing".

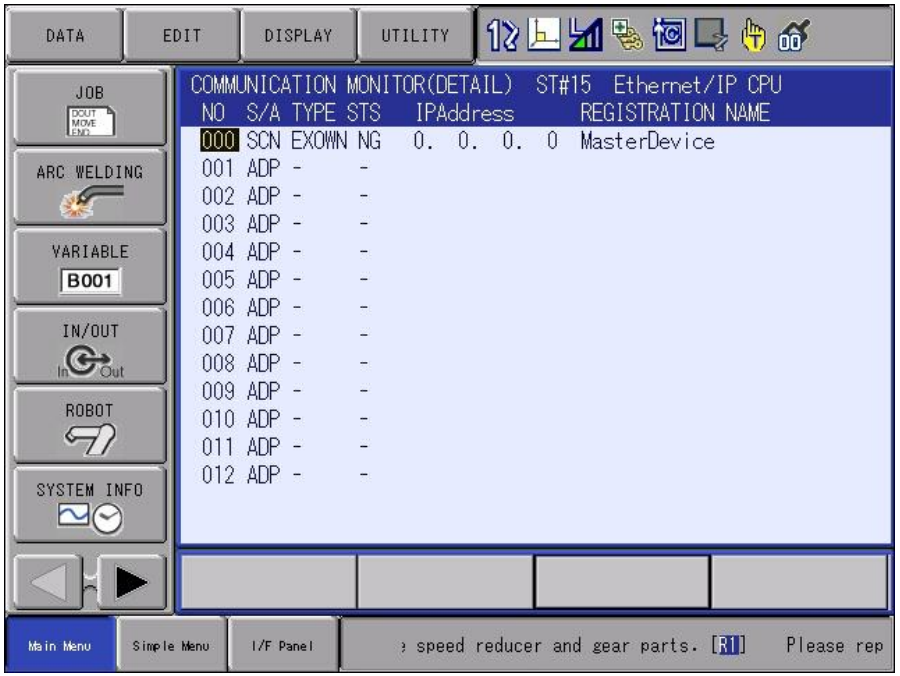
Name	Value
Robot_Control_from_PLC.avail_byte11	-1
Robot_Control_from_PLC.avail_byte11.0	1
Robot_Control_from_PLC.avail_byte11.1	1
Robot_Control_from_PLC.avail_byte11.2	1
Robot_Control_from_PLC.avail_byte11.3	1
Robot_Control_from_PLC.avail_byte11.4	1
Robot_Control_from_PLC.avail_byte11.5	1
Robot_Control_from_PLC.avail_byte11.6	1
Robot_Control_from_PLC.avail_byte11.7	1

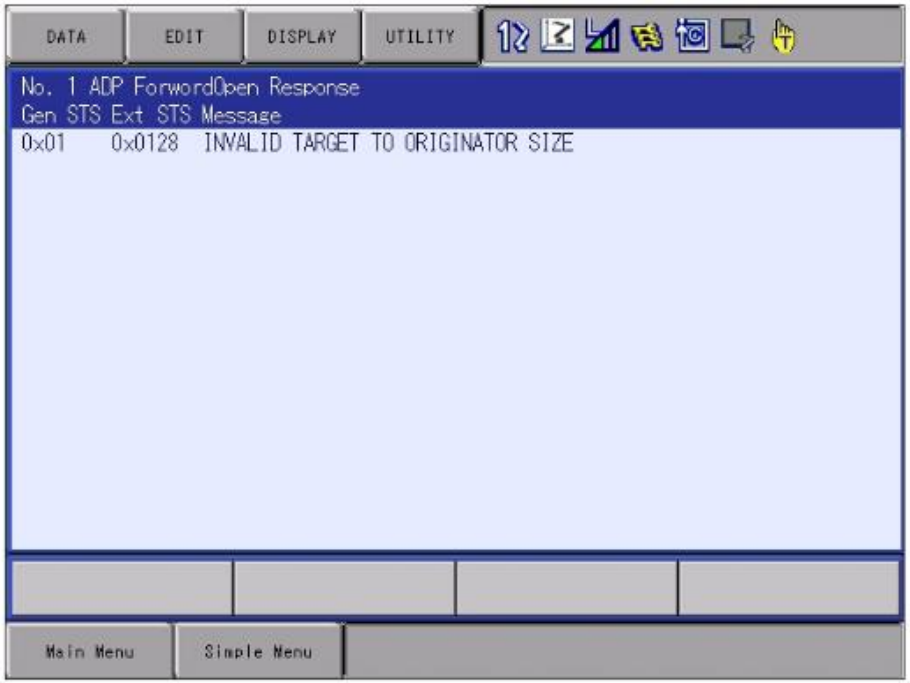


Troubleshooting Tips

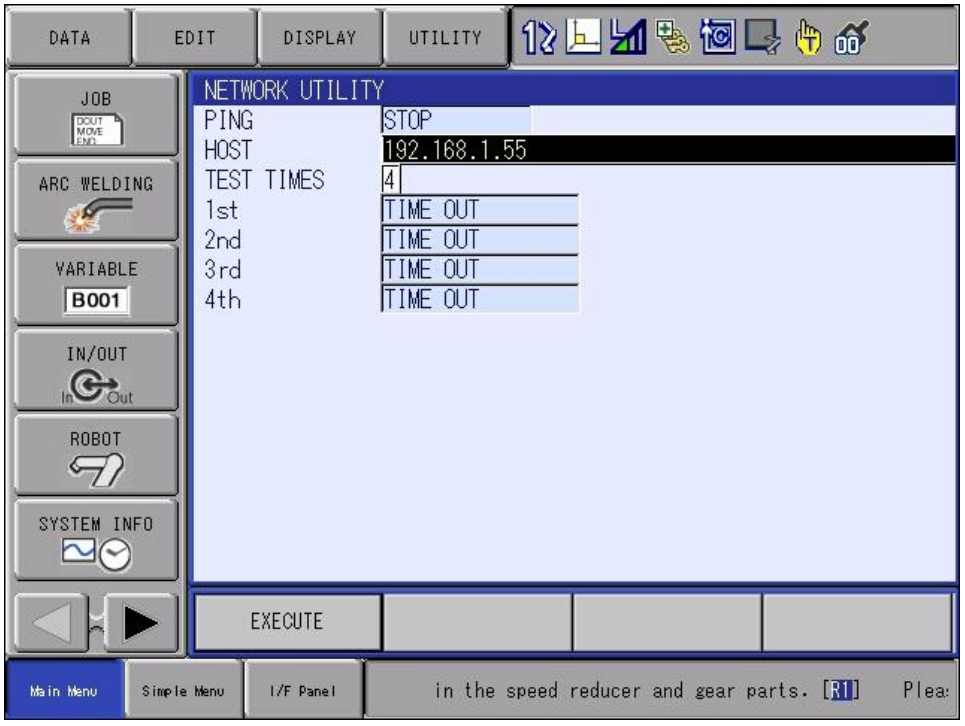
If establishing communication has failed, please follow troubleshooting tips below.

- Check the communication status of the PLC VIA the Communication Monitor residing under the IN/OUT tab. Note the General Status *STS* column there will be an error code refer to section 4.8.2 of the YRC1000 Ethernet IP manual for the error code explanation.

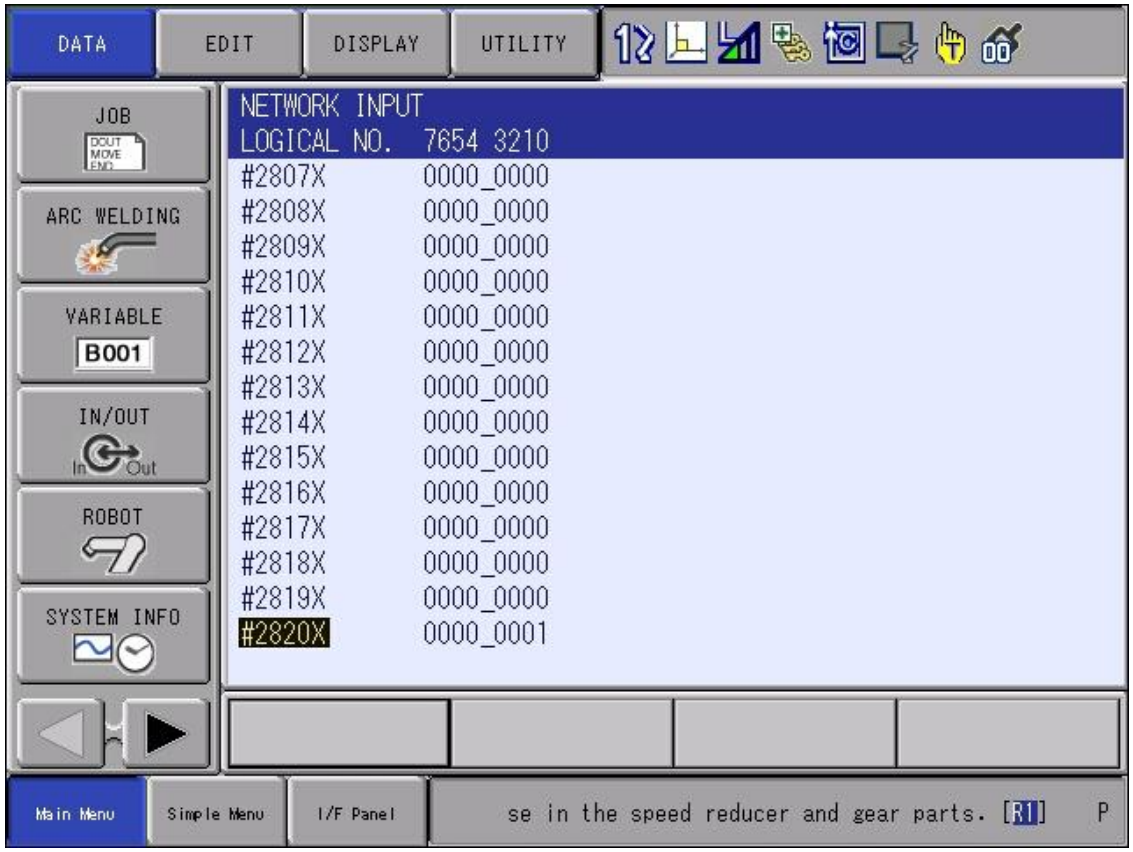




- Try pinging the PLC using the YRC1000 controller ping function. If the ping is unsuccessful verify ethernet cable is properly terminated and PLC ethernet settings are correct.



- If communication status is OK and specific control bits are not properly working. Verify the PLC enable bit is active: Network Bit 28200. Also, the robot will need to reside in Remote mode for the use of the robots control functions like EXT Start, Servos On, Call Master Job Etc.



This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

Files Referenced in this Document.

YRC1000_ArcWorld_
PLC_Map_Example_r

YRC_Master_resourc
e-2021-03-12-Standa