## Codesys Ethernet/IP Implicit and Explicit Messaging to the Yaskawa YRC1000 Robot Controller



Copyright (c) 2023 CODESYS Development GmbH

and the second se	
Version Into	License Info

\*Program was made using Codesys V3.5 SP20 + (64bit)

\*Program was ran using Codesys Control Win V3 x64

\*Achieved on a YRC1000 with the following version information

VERSION I	NFORMATION			
SYSTEM	: YAS4.90.00A(JP/EN)-0	00		
PARAMETER	: 4.28/4.28			
MODEL	: 1-06VXH12-A0*(AR1440	))		
APPLI	: ARC WELD			
LANGUAGE	: 4.90-00-00/ 4.90-00	)-00		
CPU	SYSTEM ROM(CPU1/CPU2)	) BOOT ROM	OS/FPGA	
ACP01	4.90.00		1.05-10	
AIF01			16092200	
PP	3.00-00	2.11-00	1.14	
ASF01#0	2.26-00 / 2.26-00	1.00	16122200	
ASF01#1	DummyMsf / DummyMsf	DummyMsf	15010100	
ASF01#2	DummyMsf / DummyMsf	DummyMsf	15010100	
ASF01#3	DummyMsf / DummyMsf	DummyMsf	15010100	
ASF01#4	DummyMsf / DummyMsf	DummyMsf	15010100	

Michael Allen

Michael.allen@motoman.com

01/23/2025

## Overview

This document shows how to configure a Codesys Windows Soft PLC to communicate over EIP using both implicit and explicit messaging to a Yaskawa YRC1000 Robot Controller. In this example 8 bytes of input and output data was created and a position variable was transferred over the explicit connection.

## **PLC Setup**

 Download Codesys Control Win V3 x64 and install it and then run it. This is the "PLC", you will download projects to it. Once installed, run it from the start menu, you will see a popup that will look like the image below, this indicates that the PLC is running, you cannot close this.

CODESYS Control Win V3 - x64	-		$\times$
			_/_^
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	_/_/_/		1_1
	      	_I I I I   	, '/ / _/
<pre>2025-01-23T17:24:50.712Z: Cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <logoptions> 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <tpre>topable&gt;type&gt;topable&gt; 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <tpre>topable&gt;type&gt;topable&gt; 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <tisable>type&gt;topable&gt;type&gt;topable&gt; 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <disable>type&gt;type&gt;type&gt;type&gt; 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <disable>type&gt;type&gt;type&gt;type&gt;type&gt; 2025-01-23T17:24:50.712Z: cmp=CmpLog, Class=1, Error=0, Info=0, pszInfo= <disable>type&gt;type&gt;type&gt;type&gt;type&gt;type&gt;type&gt;typ</disable></disable></disable></tisable></tpre></tpre></logoptions></pre>	_'_'_'	amp>	

- Download, install and run Codesys V3.5 SP20, this is the programming software. Select File->New Project.
- 3. Click standard project, give it a name and then press OK.

New Pro	ject				
Categories		Templates			
Lib	raries ojects	Empty project	HMI project	Standard project	Standard project w
A project of	ontaining one device		amoty implement	tation for PLC	PPC
A project o	,	one application, and arre	impry implement		j ku
Name	Untitled 1				
Location	C: \Users \allenmi \Do	ocuments			×
				ОК	Cancel

4. It will then ask you the device type and the programming type you would like to use. You can select whichever device type you would like to use however in the example I used Codesys Control Win V3 x64 which is the Windows runtime that can operate on any windows computer. I wanted to be able to test this on my laptop which is why I used this as the device. Choose Structured Text for the program type and press OK.

Standard P	roject		×
1	You are abou objects within - One program - A program F - A cyclic task - A reference	t to create a new standard project. This wizard will create the following this project: nmable device as specified below LC_PRG in the language specified below which calls PLC_PRG to the newest version of the Standard library currently installed.	
	Device PLC_PRG in	CODESYS Control Win V3 x64 (CODESYS) Structured Text (ST) OK Cancel	>

5. At the top of the screen press the little gear with a plug to go online with the PLC. Make sure you have started Codesys Control Win V3 x64 and you see the command line like page running.

ols Window Help		
🤋 케 레   🛱   🛅 - 📑   🛗   App	lication [Device: PLC Logic] 🔻 🙀 🧐 🕞 🔳 🤻 🗍	v i i i i i i i i i i i i i i i i i i i
Device X Ethernet	Login (Alt+F8)	
Communication Settings Scan N	Network Gateway - Device -	
Applications		
Backup and Restore	· · · · · · · · · · · · · · · · · · ·	
Files		
.og	Gateway	
LC Settings	Gateway-1 TP-Address:	[002E] (active) Device Name:
LC Shell	localhost	Serie Hand
lears and Groups	Port: 1217	
, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
 \ \		

6. Once you press the gear it will ask you if you want to go online. Press yes and if required put in a password when running the PLC the first time. Your screen should

look like the image below with a green strip over the device and a red bar at the bottom saying it is in Stop mode.



7. No go offline by pressing the gear beside the go online gear, then double click the Device to get to the device screen.

0	
	Untitled1
	🖹 👚 📆 Device (CODESYS Control Win V3 x64)
	🖮 🗐 PLC Logic 🛛 🖓 📗
	Application     Device (CODESYS Control Wir
	Library Manager
	PLC_PRG (PRG)
Application Device: DICLogic L 🗸 🕮 🕅 📐 🕳 🕸 🗐 🖉 🖿	🖃 🎆 Task Configuration
	🖹 🕸 MainTask (IEC-Tasks)
Logout (Ctrl+F8)	PLC_PRG
3	

8. On the device screen you should see the gateway and the PLC should have (active) beside the PLC name, in my case it is 002E (active).

Communication Settings	Scan Network	Gateway 👻 Dev	ice 🔹				
Applications		_				_	
Backup and Restore				•			
Files						•	
Log		6	Gateway-1	Gateway	~	[002E] (active)	~
PLC Settings		1	P-Address:			Device Name:	
PLC Shell		F	ort:				
Users and Groups		1	217				
Access Rights							
Symbol Rights							
Software metrics for license determination							
IEC Objects							
Task Deployment							
Status							
Information							

9. On the device screen press PLC Settings.



10. Change "Always Update Variables" from disabled to Enabled 1 so you can see the bits being transferred between the robot and the PLC without needing to use them in the program. Otherwise if you just try and toggle them on in the watch table they will not work!

Device X	
Communication Settings	Application for I/O handling Application
Applications	PLC Settings
Backup and Restore	Behavior for outputs in stop Keep current values
Files	Always update variables Disabled (update only if used in a task)
Log	Bus Cycle Options Enabled II Use bus cycle tack (Finot used in any tack) Bus cycle tack
PLC Settings	
PLC Shell	Additional Settings
Users and Groups	Show I/O warnings as errors Enable symbolic access for IOs
Access Rights	
Symbol Rights	
Software metrics for license determination	
IEC Objects	
Task Deployment	
Status	
Information	

11. Right click on the device again and select add device.



12. Click Fieldbuses, Ethernet Adapter, Ethernet and then click Add Device on the bottom right hand corner of the page.

Name Miscellaneous Can CaNbus Can Can Can Canbus Can Can Can Can Canbus Can Can Can Can Can Canbus Can Can Can Can Can Can Can Can Can Can	Vendo	r Version	Description		
m. III Ethornot Adapter					
Citement     Citement	CODES	rs 4.2.0.0	Ethernet Link. CIFX EtherNet/IP Scanner		
Group by category Display al Name: Ethernet Vendor: CODESYS Categories: Ethernet Adapte Version: 4.2.0.0 Order Humber - Description: Ethernet Link.	r, Ethernet A	or experts only)	Display outdated versions	*	

13. Double click the ethernet device and under the general tab select "Browse." beside the Network Interface to select the network interface you would like to use. In my case I want Ethernet 2 so I select it and press OK.

IP address         192 . 168 . 0 . 1           Subnet mask         255 . 255 . 255 . 0           Default gateway         0 . 0 . 0 . 0	Network interface	Network interface	rface	Brov
Subnet mask         255         255         0           Default gateway         0         .         0         .         0	IP address	IP address	192 . 168 . 0 . 1	
Default gateway 0 . 0 . 0 . 0	Subnet mask	Subnet mask	255 . 255 . 255 . 0	
	Default gateway	Default gateway	way 0.0.0.0	
Adjust operating system settings	Adjust operating	Adjust operating	perating system settings	

Name		Description	IP address
Bluetooth Network C	Connection	Bluetooth Device (Personal Area Network)	0.0.0.0
Ethernet 2		Realtek USB GbE Family Controller	192.16
Ethernet 5		VirtualBox Host-Only Ethemet Adapter	192.16
vEthernet (Ethernet	2)	Hyper-V Virtual Ethernet Adapter	172.22
vEthernet (Ethernet	5)	Hyper-V Virtual Ethernet Adapter #2	172.26
IP address	192 . 1	68 . 1 . 46	
Subnet mask	255 . 2	55 . 255 . 0	
Default gateway	192 . 1	68.1.1	
MAC address	5C:28:86	:3E:A6:8A	

14. Right click the Ethernet device and click add device.

<b>#</b> 1	PLC_	PRG	Status
Ethernet (Ether	X	Cut	mation
	e <sub>e</sub>	Сору	macion
	e	Paste	
	$\times$	Delete	
		Refactoring	•
	G.	Properties	
	51.2 10.00 10.00 10.00	Add Object	
		Add Folder	
		Add Devise	
		Insert Device	
		Disable Device	
		Update Device	
	D°	Edit Object	
		Edit Object With	
		Edit IO mapping	
		Import mappings from CSV	
		Export mappings to CSV	es - Total C
	_	[ [	ownload

15. Under Fieldbuses, EtherNet/IP, EtherNet/IP Scanner select EtherNet/IP Scanner and then press Add Device.

e cherver_iP_scanner			
ion			
Append device 🔿 Insert device 🕥	Plug device O	Update dev	ice
ng for a full text search	Vendor	<all th="" vend<=""><th>ors&gt;</th></all>	ors>
ame	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Local Adapter			
= - EtherNet/IP Scanner	0005010		The state ID Courses
Ethenvet/IP Scanner	CODESTS	4.5.1.0	Etherwet/ur Scanner
E PROFINET IO			
IN PROPINEL IO			
iroup by category 🗌 Display all Vers	ions (for experts	only) 🔲 🛙	Display outdated versions
Siroup by category Display all vers	ions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: Dharket/Dr Scanner Vender: COFEN	ions (for experts	only) 🔲 🛙	Display outdated versions
Sroup by category Display all vers Name: Etherlet/IP Scamer Vendor: COESYS Categories: Etherlet/IP Scamer	ions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: Dharket/D Scame Vendor: COESTS Categories: EtherNet/D Scame Version: 43:1.0	ions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: EtherNet/IP Scamer Vendor: CODEYS Categories: EtherNet/IP Scamer Version: 4.5.1.0 Order: Humber: 1	ions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: EtherNet/D Scarner Vendor: COSEYS Categories: EtherNet/D Scarner Version: 43.5.0 Order Humber: 1 Descruption: EtherNet/D Scarner	iions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: EtherNet/IP Scamer Vendor: CODEX/S Categories: EtherNet/IP Scamer Version: 4.5.1.0 Order, Imailien: 1 Description: EtherNet/IP Scamer	iions (for experts	only) 🗌 [	Display outdated versions
Sroup by category Display all vers Name: EtherHeV/D Scarner Vendor: COESYS Categories: EtherHeV/D Scarner Version: 4.5.1.0 Order Number: 1 Description: EtherHeV/D Scarner	ions (for experts	only) 🗌 [	Display outdated versions
Sroup by category Display all vers Name: EtherNet/JP Scanner Vendor: CODEX'S Categories: EtherNet/JP Scanner Order: Number: 1 Description: EtherNet/JP Scanner	iions (for experts	only) 🗌 [	Display outdated versions
Sroup by category Display all vers Name: EtherHeV/D Scarner Vendor: COEPS'S Categories: EtherHeV/D Scarner Version: 4.5.0 Order, Namber: 1 Description: EtherHeV/D Scarner	iions (for experts	only) 🗌 [	Display outdated versions
iroup by category Display all vers Name: EtherNet/JP Scamer Vendor CO2E315 Version: 45.14/JP Scamer Version: 45.14/JP Scamer Description: EtherNet/JP Scamer	ions (for experts	only) 🔲 [	Display outdated versions
Incup by category Display all vers Name: EtherHe/I/P Scamer Vendor: COESYS Categories: EtherHe/I/P Scamer Order Namber: 1 Description: EtherNe/I/P Scamer Incurst and Action Scamer Description: Action Scamer Description	iions (for experts	only) [][	Display outdated versions
iroup by category Display all vers Nume: EtherNet/IP Scamer Vendors COSENS Categories: EtherNet/IP Scamer Version: 45.10 Order Humber: 1 Description: EtherNet/IP Scamer end velocted device as last child o zmet	iions (for experts	only) [][	Display outdated versions

16. Rick click the EtherNet\_IP\_Scanner and press Add Device just like we did for the scanner.



17. Under Fieldbuses, EtherNet/IP, EtherNet/IP Remote Adapter select Generic EtherNet/IP device and then select add device. Double click on the newly added device.

Generic_Etherivet_IP_device					
ction					
) Append device () Insert device () Plug	device O	Jpdate devic	e		
ring for a full text search	Vendor	<all th="" vendor<=""><th>\$&gt;</th><th></th><th></th></all>	\$>		
lame	Vendor		Version	Description	
Fieldbuses					
EtherNet/IP					
EtherNet/IP Remote Adapter	CODESY		Major Dovision - 16#1 Minor Dovision - 16#1	EthorNot/ID Target imported fr	om EDC Eilou
Etherweiter Adapter	STCK AG	3	Major Revision=16#1, Minor Revision = 16#1 Major Revision=16#1, Minor Revision = 16#2	EtherNet/IP Target imported fr	om EDS File:
Generic EtherNet/IP device	CODESY	s	4.1.0.0	EtherNet/IP Target for a gener	ic Device
IO-Link master EIP 8P IP 20	ifm elect	ronic gmbh	Major Revision=16#1, Minor Revision = 16#4	EtherNet/IP Target imported fr	om EDS File:
Group by category 🗌 Display all versions	(for experts	only) 🗌 Dis	splay outdated versions		
Group by category Display all versions Vendor: CODESYS Categories: EtherNet/IP Remote Adapt Version: 4.1.0.0 Order Number: Description: EtherNet/IP Target for a c	(for experts of er eneric Device	only) 🗌 Dis	splay outdated versions	<b>V</b> 0	

18. On the general tab of the Generic EtherNet IP Device change the IP address to the address of the robot, in my case it was 192.168.1.31. The Vendor ID is 44, Device Type is 12 and Product Code is 1283 for the YRC 1000.



19. Click the Connections tab and then select Add Connection...



20. The Configuration Assembly Instance is 96. The Consuming Assembly Instance is 32 and the Producing Assembly Instance is 64. The Class ID is 4 for all instances and the Attribute ID is 3 for all instances. The O→T size and T→O sizes are set to 8 bytes and the RPI(ms) is set to 10.

Edit Connection				
Connection Path Setting	IS			OK
<ul> <li>Automatically ger</li> </ul>	nerated path			OK
Configuration	n assembly			Cancel
Class ID: 16	# 4 Instance ID: 16# 96	Attribute ID: 16#3		
Consuming as	ssembly (O>T)			
Class ID: 16	#4 Instance ID: 16# 32	Attribute ID: 16# 3		
Producing as	sembly (T>0)			
Class ID: 16	#4 Instance ID: 16#64	Attribute ID: 16#3		
O Dath defined by a	umbolicopmo			
O Path defined by s	yndonename			
General Parameters				
Connection Path	20 04 24 96 2C 32 2C 64			
			No. 1	
Trigger type	Cyclic ~	RPI (ms)	10	
Transport type	Exclusive owner $\checkmark$	Timeout multiplier	4 ~	
Scanner to Target (Outp	put)	Target to Scanner (Inp	out)	
0>T size (bytes)	8	T>0 size (bytes)	8	
Provy config size (by	tes) 0			
Treast seefin size (b)	(to) 0			
Target config size (b)	(tes) 0			
Connection type	Point to Point V	Connection type	Multicast $\checkmark$	
Connection priority	Scheduled $\sim$	Connection priority	Scheduled $\checkmark$	
Fixed/Variable	Fixed $\checkmark$	Fixed/Variable	Fixed $\checkmark$	
Transfer format	32-bit run/idle 🗸 🗸	Transfer format	Pure data $\sim$	
Inhibit time (ms)	0	Inhibit time (ms)	0	
Heartbeat multiplier	1			

21. Click the assemblies tab and change the data types of the outputs and inputs to SINT from BYTE.

+ # X	Device Ethernet	EtherNet_IP_Scan	ner 🖉	Generic_Eth	erNet_IP_dev	ke 🗙 📄 PLC_PRG					
Device (CODESYS Control Win V3 x64)	General	Connections									
PLC Logic     PLC Logic     Application	Connections	Connection Nam	e O>TSiz	e (Bytes)	T>O Size (By	tes) Proxy Config Size (Bytes)	Target Config S	ize (Bytes)			
Library Manager     PLC_PRG (PRG)	Assemblies	Generic connection	8		5						
Task Configuration     Set BNIPScamerIOTask (IEC-Tasks)	User-Defined Parameters	Consuming Assembl	y "Output" (O	>T)			Producing Assen	bly "Input" (T	>0)		
EtherNet_IP_Scanner_IOCycle	Log	Add X Delete	e 🕸 Movel	Jp   Movel	Jown	Line Christe	Add X De	lete @ Mov	eUp ⊕ Mov	: Down	The Original
EtherNet_IP_Scamer.ServiceCyde	EtherNet/IP I/O Mapping	Output_Param0	SINT SINT	8	Unit	nep string	Input_Perem0	SINT	8	Onic	Hep String
HainTask (JEC-Tasks) - Office_PRG	EtherNet/IP IEC Objects	Output_Param1 Output_Param2	SINT	8			Input_Param1 Input_Param2	SINT	8		
Ethernet (Ethernet)     EtherNet /P_Scanner (EtherNet//P Scanner)	Status	Output_Param3 Output_Param3	SINT	8			Input_Param3	SINT	8		
Generic_EtherNet_JP_device (Generic EtherN	Information	Output_Param5	SINT	8			Input_Param5	SINT	8		
		Output_Param7	SINT	8			Input_Param7	SINT	8		

22. Now you need to complete the setup on the robot. Follow SSGW-509 for Ethernet IP setup. In Maintenance->System->Setup->OptionFunction->EtherNet/IP(CPU Board) press Detail. Press detail under Adapter. Enabled the Adapter and put in the settings below. Then complete the EtherNet/IP setup by pressing Enter and Modify and auto allocate the IO.

		1	
SYSTEM	ADAPTER		
	ADAPTER	ENABLE	
FILE	OUTPUT SIZE	8 byte	
	CONFIGURATION SIZE	0 word 50	
SD	OUTPUT INSTANCE	100 150	
MotoPlus APL.	CONTROLING INCIDENCE	100	
<u> </u>			
DISPLAY SETUP			
Aa			
REMOTE PENDANT			
Main Heru Simple	Nerro I/F Panel Mainten	ance mode	

Enabled adapter and change instances and IO size, press enter.



Press enter and then Modify.



Press enter twice and then Modify.



Set Allocation Mode to AUTO and press enter.

SYSTEM	EXTERNAL S	IO ALLO T# CH	DCATION(I MAC ID	NPUT) ADOR	BYTE	NAME	
FILE	#20010 #20060 #20070	0 0 15 0 15 0	0 254 0	0 0 1	5 1 8	ASF01 Ethernet/IP C Ethernet/IP C	PU PU
EX. MENORY	#25110	2 0	Ő	0	2	AEW01	
DISPLAY SETUP							
Hain Menu Simple	Menu I/F Pa	nel 🗄	à i	Select	'Safety	Board FLASH Res	et'.

Take note where the 8 bytes of input data get mapped to, in this case they start at #20070, press enter.

								1	
SYSTEM		EXTERN/	L IO ST#	ALLO CH	CATION(O MAC ID	utput) Ador	BYTE	NAME	
FILE	٦	#30010 #30060 #30070	0 15 15	0 0 0	0 254 0	0 0 1	5 1 8	ASF01 Ethernet/IP Ethernet/IP	CPU CPU
EX. MEMOR	87	#35110	2	0	Û	0	2	AEWUT	
MotoPlus Af	PL.								
DISPLAY SE	TUP								
	DANT	_				_	1		
			_			_			
Main Menu		Menu 17	Panel	- F		Select	'Safety	Board FLASH R	eset'.

Take note where the 8 bytes of output data get mapped to, in this case they start at #30070, press enter and Modify.



Perform safety flash reset and then reboot the controller in online mode.

23. Once the robot boots back up you can go online with the PLC by pressing the gear and then downloading to the PLC. Then press the play button to start the PLC. Your screen should look like the image below. Note you should see green circles around all devices if connection



24. Now to test implicit connections we are going to turn on some bits on both the robot and PLC and make sure the other can see them. Click on the EtherNet/IP I/O Mapping tab and notice the first Input and first output addresses. Mine are located at %IB0 and %QB0.

Nevices 🗸 🗸 🗙	Device Ethernet	EtherNet_IP_Scanner	/ 🕤 Generi	c_EtherNet_IP_de	vice 🗙 📄	PLC_PRG					
Untitled1     Untitled1     Device [connected] (CODESYS Control Win V3 x64)	General	Find		Filter Show all			🕶 🖶 Add FB for IO (	🔹 🕂 Add FB for IO Channel 🔶 Go to Instance			
PLC Logic     Application [run]	Connections	Variable	Mapping	Channel	Address	Туре	Current Value	Prepared Value	Unit	Description	
ibrary Manager	Assemblies	ii - 🍅		Input_Param0	%IB0	SINT	0				
□       □		😐 🐐		Input_Param1	%IB1	SINT	0				
	User-Defined Parameters	🖷 - 🍫		Input_Param2	%IB2	SINT	0				
	-	i≣*≱		Input_Param3	%IB3	SINT	0				
	Log	🖷 - 🍫		Input_Param4	%IB4	SINT	0				
ENIPScannerServiceTask (IEC-Tasks)	51	÷ *		Input_Param5	%IB5	SINT	0				
EtherNet_IP_Scanner.ServiceCycle	Ethenvet/IP I/O Mapping	ii 🖗 - 🍫		Input_Param6	%IB6	SINT	0				
AlinTask (IEC-Tasks)	EtherNet/IR IEC Objects	· · · · · · · · · · · · · · · · · · ·		Input_Param7	%IB7	SINT	0				
PLC_PRG	Ethenvely in ince objects	÷- 🍫		Output_Param0	%QB0	SINT	0				
Ethernet (Ethernet)	Status	÷-*•		Output_Param1	%QB1	SINT	0				
EtherNet_IP_Scanner (EtherNet/IP Scanner)		🖷 - 🍫		Output_Param2	%QB2	SINT	0				
🧐 🛄 Generic_EtherNet_IP_device (Generic Ethe	Information	÷-*>		Output_Param3	%QB3	SINT	0				
		÷ *>		Output_Param4	%QB4	SINT	0				
		÷- *>		Output_Param5	%QB5	SINT	0				
				Output_Param6	%QB6	SINT	0				
		iii <b>*</b> ≱		Output_Param7	%QB7	SINT	0				

25. In the watch table add %IBO and %QBO. Click on prepared value on %QBO and change it to 1. Then right click and select Write Values, this will write a value of 1 to the first bit in the first byte of output data to the robot.

Watch 1								<b>↓</b> ₽	;
Expression	Application	Туре	Value	P	Prepared val	ue Execution point			
%IB0	Device.Application	BYTE	0			Cyclic Monitoring			9
%QB0	Device. Application	BYTE	0	1	L X	Cut			9
						Сору			
					(fill)	Paste			
					×	Delete			
						Select All			
						Browse	•		
					2	Input Assistant			
						Write Slues			

26. Now on the pendant click IN/OUT->External Input. Look at the first bit that was mapped when you did the auto allocation, in my case it was #20070. It should now be a 1 instead of a 0.

DATA	EDIT	DISPLAY	UTILITY	12 🗷 📶 🥫	👌 🔟 🥥 (	Ð
	EXTE	RNAL INPUT	54 3210			
ARC WELDI	NG #200 #200 #200	12X 00 13X 00 14X 00	000_0000 000_0000 000_0000			
VARIABLE	#200 #200	15X 00 16X 00 17X 00	00_0000 00_0000 00_0000			
	#200 #200 #201	18X 00 19X 00	00_0000 000_0000 000_0000			
ROBOT	#201 #201 #201	1X 00 2X 00				
SYSTEM IN	F0 #201	4X 00	100_0000			
		Ĩ	-).			
Main Menu	Simple Menu	I/F Panel	S 🔁	afety guard is ope	n	MotoLaser

27. Now to test outputs go to IN/OUT->External Output. Click Display->Detail and then go to the start of your EtherNet/IP Outputs, which in my case was #30070. Hover over the box, press select to turn on Sim and then hover over the circle and press Interlock and Select at the same time to simulate the output. On the PLC program you should see %QB0 change from 0 to 1.

	<					
			Reset Map	ping Always update	use parent device se	tting
Watch 1						<b>→</b> ₽ X
Expression	Application	Туре	Value	Prepared value	Execution point	Ad
%IB0	Device.Application	BYTE	1		Cyclic Monitoring	%1
%QB0	Device. Application	BYTE	1		Cyclic Monitoring	%0

28. Now to test explicit communication we are going to read a position variable and then write it to another position variable. Please go offline with the PLC, copy the logic and variables to your PLC and then download and go back online with the PLC.

Johne	Ethernet	EtherNet_IP_Scanner Generic_EtherNet_IP_device PLC_PRG X
1 PROG	RAM PLC_PRG	
2 VAR	-	
3	reset	: ENIP.Reset; (* fb to reset a cip class *)
4	getAttributeSingle	: ENIP.Get_Attribute_Single; (* fb to get a single attribute from a cip class *)
5	setAttributeSingle	: ENIP.Set Attribute Single; (* fb to set a single attribute of a cip class *)
6	getAttributeAll	: ENIP.Get Attributes All; (* fb to get all attributes of a cip class *)
7	- setAttributeAll	: ENIP.Set Attributes All; (* fb to set all attributes of a cip class *)
8	udiReceivedData	: UDINT:
9	done	: BOOL:
10	busv	: BOOL:
11	vFrror	BOOL-
12	startTransfer	- BOOL-
13	svag	
14 10	anes Man	· ARAT[013] OF DWORD,
	VAR	
		Variable declaration
		Variable declaration
		Variable declaration
getAttribut	eAll(xExecute:= startTrans	variable declaration
getAttribut itfEthe	eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth	sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)
getAttribut itfEthe eClass: dwInsta	eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 16477, nee:= 1	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *) (* cip class which contains the desired attribute *) (* cip class which contains the desired attribute *)</pre>
getAttribut itfEthe eClass: dwInsta pData:=	eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 16#7F, nce:= 1, ADR(axes),	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itfEthe eClass: dwInsta pData:= udiData	eAll(xExecute:= startTrans NMetIPDevice:= Generic_Eth = 16#7F, nce:= 1, ADR(axes), Size:= SIZEOF(axes),	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itfEthe eClass: dwInsta pData:= udiData xDone=> *Busy=>	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 1647F, nce:= 1, ADR(axes), Size:= SIZDOP(axes), done, here</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itfEthe eClass: dwInsta udiData;= udiData xDone=> xBusy=> xErro=	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 1647F, nce:= 1, ADR(axes), Size:= SIZEOF(axes), done, busy, &gt; xError,</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itfEthe eClass: dwInsta pData:= xDone=> xBusy=> xFrror eError=	eAll(xExecute:= startTrans NetIPDevice:= Generic_Eth = 160PT, note:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, > xError, > ,	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *) (* cip class which contains the desired attribute *) (* value of 0 is class level, range from 1x is instance level *) (* data buffer *) (* size of the data buffer *) (* size of the data buffer *) (* attribute no. 6 of the tcp/ip interface object is the Host</pre>
getAttribut itfEthe eClass: dvInsta pData:= udiData xBusy=> xErroz= eErroz= udiRece	<pre>eAll(xExecute:= startTrans NMetIPDevice:= Generic_Eth = 160F, nce:= 1, ADR(axes), Size:= SIEBOF(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived</pre>	<pre>dData);</pre>
getAttribut itffthe eClass: dwInsts xDone=> xDusy=> xError= eError= udiRece setAttribut	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 169T, nce:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll(</pre>	<pre>dData);</pre>
getAttribut itffthe eClass: dWInsta udiData: udiData xBusy=> xBusy=> xError= eError= udiRece setAttribut	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 16#7F, nce:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, viwedDataSize=&gt; udlReceived eAll( e:= startTransfer,</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itffthe eClass: dwInsta udiPata: udiPata: xBusy=> xRusy=> wfrror= eError= udiRece setAttribut xExecut itffthe eClass	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 16#7F, nce:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, tredDataSize=&gt; udiReceived eAll( e:= startTransfer, rNetIPDevice:= Generic_Eth = 16#7F</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
<pre>getAttribut itfEthe eClass: dvInsta pData:= udiData xDone=&gt; xBrror xBryror xError= udiRece setAttribut xExecut itfEthe eClass: dvInsta</pre>	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 1647F, noc:= 1, ADR(axes), Size:= SIZEOF(axes), done, busy, &gt; xError, &gt; xtrror, &gt; xtrror, &gt; vivedDataSize=&gt; udiReceived eAll( e:= startTransfer, rNetIPDevice:= Generic_Eth = 1647F, noc:= 2,</pre>	<pre>dData); value of the device (instance is found in the I/O Image of the device) ')     (* cip class which contains the desired attribute *)     (* value of 0 is class level, range from 1x is instance level *)     (* data buffer *)     (* asize of the data buffer *)     (* attribute no. 6 of the tcp/ip interface object is the Host  dData); herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)     (* cip class which contains the desired attribute *)     (* value of 0 is class level, range from 1x is instance level *)     (* value of 0 is class level, range from 1x is instance level *)     (* value of 0 is class level, range from 1x is instance level *) </pre>
getAttribut itfEthe eClass: dwInsta yData:= udiData xBusy=> xError= eError= udiRece setAttribut itfEthe eClass: dwInsta pData:=	<pre>eAll(xExecute:= gtartTrans thetPDevice:= Generic_Eth = 16fPT, note:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll( e1= startTransfer, ThetPDevice:= Generic_Eth = 16fPT, note:= 2, ADR(axes),</pre>	<pre>ster, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *) (* cip class which contains the desired attribute *) (* value of 0 is class level, range from 1x is instance level *) (* data buffer *) (* size of the data buffer *) (* attribute no. 6 of the tcp/ip interface object is the Host dData); herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *) (* value of 0 is class level, range from 1x is instance level *) (* value of 0 is class level, range from 1x is instance level *) (* value of 0 is class level, range from 1x is instance level *) (* data buffer *)</pre>
getAttribut itfEthe eClass: dvInsta pData:= udiData eError= udiRece setAttribut itfEthe eClass: dvInsta pData:= udiData	<pre>eAll(xExecute:= startTrans NetIPDevice:= Generic_Eth = 160FF, nce:= 1, ADR(axes), Size:= SIEBOF(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll( e:= startTransfer, ThetIPDevice:= Generic_Eth = 160FF, nce:= 2, ADR(axes), Size:= SIZEOF(axes), </pre>	<pre>define the state of the device (instance is found in the I/O Image of the device) ')     (' cip class which contains the desired attribute ')     (' value of 0 is class level, range from 1x is instance level ')     (' data buffer ')     (' airs of the data buffer ')     (' airs of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host  dData);  herNet_IP_Device, (' instance of the device (instance is found in the I/O Image of the device) ')     (' cip class which contains the desired attribute ')     (' attribute no. 6 of the device) ')     (' cip class which contains the desired attribute ')     (' value of 0 is class level, range from 1x is instance level ')     (' data buffer ')     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' size of the data buffer ')     (' attribute no. 6 of the top/ip interface object is the Host     (' attribute no. 6 of the top/ip interface object is the Host     (' attribute no. 6 of the top/ip interface object is the Host </pre>
getAttribut itffthe eClass: dwInsts xDone=> xBusy=> xError= udiRece setAttribut itfEthe eClass: dwInsts pData:= pData: udiRece xRecut xfxecut itfEthe eClass: dwInsts pData: xDone=> xBusy=>	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 160FF, nce:= 1, ADR(axes), Size:= SIZEOF(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll( e:= startTransfer, rNetIPDevice:= Generic_Eth = 160FF, nce:= 2, ADR(axes), Size:= SIZEOF(axes), done, busy.</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itffthe eClass: dwInsta xDone=> xBusy=> xError= eFrror= udiRece setAttribut xExecut itffthe eClass: dwInsta pData:= udiData xDone=> xEuro=	<pre>eAll(xExecute:= startTrans INetIPDevice:= Generic_Eth = 16#7F, nce:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, vedDataSize=&gt; udiReceived eAll( e:= startTransfer, INetIPDevice:= Generic_Eth = 16#7F, nce:= 2, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError,</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
getAttribut itfEthe eClass: dwInsta pData:= udiData xBusyo- xError= udiRece setAttribut xExecut itfEthe eClass: dwInsta xDone> xBusyo- xError= eError=	<pre>eAll(xExecute:= startTrans rNetIPDevice:= Generic_Eth = 16000000000000000000000000000000000000</pre>	<pre>sfer, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
<pre>getAttribut itfEthe eClass: dwInsta whose&gt; xBuay=&gt; xBuay=&gt; xError= udiRece setAttribut itfEthe eClass: dwInsta pData:= udiRece xBuay=&gt; xBuay=&gt; xError= eError= xBuay=&gt; xError= eError= y;</pre>	<pre>eAll(xExecute:= gtartTrans NWetIPDevice:= Generic_Eth = 160PT, noc:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll( et= startTransfer, NWetIPDevice:= Generic_Eth = 160PT, noc:= 2, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;</pre>	<pre>ster, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>
<pre>getAttribut itfEthe eClass: dwInsta whose xBusy=&gt; xError= udiRece setAttribut itfEthe eClass: dwInsta cPlata:= udiRece xExecut itfEthe eClass: dwInsta xData:= udiRece xError= xBusy=&gt; xError= eFror= eFror= );</pre>	<pre>eAll(xExecute:= gtartTrans NWetIPDevice:= Generic_Eth = 16fPT, noc:= 1, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;, ivedDataSize=&gt; udiReceived eAll( e:= startTransfer, TWetIPDevice:= Generic_Eth = 16fPT, noc:= 2, ADR(axes), Size:= SIZEOP(axes), done, busy, &gt; xError, &gt;</pre>	<pre>ster, herNet_IP_Device, (* instance of the device (instance is found in the I/O Image of the device) *)</pre>

Program

29. In this program we are calling a getAttributeAll function that is in the standard EtherNet IP Library included with Codesys, if your program does not recognize it you may have to download it and install it. The CIP class for Position Variables is 16#7F as stated in the YRC1000 EtherNet/IP Manual. There is 13 x 4 Byte values that will be returned corresponding to the different axis and rotation values for the position. I made an array called Axes that will store all the values.

Table 5-16(a): Required Format								
Division	Outline	Size	Data	Explanation				
Header	Class	2 Byte	0x7F	Vendor-specific				
	Instance	2 Byte	Specify the variable P to read/write From 0 or from 1	Specify the variable P number (Parameter: RS022=1)				
				Specify the variable P number +1 (Parameter: RS022=0)				
	Attribute	1 Byte	Specify the read data type 1 to 13	Specify the data number of the position information 1: Data type to 13: 8th axis data				
	Service	1 Byte	Specify the data access method 0x01: Get Attribute All 0x0E: Get Attribute Single 0x02: Set Attribute All 0x10: Set Attribute Single					

## Read and write a robot position-type variable (P)

Table 5-16(b): Response Format (When Reading All Attributes/Omitting Head					
Division	Outline	Size	Explanation		

Division	Outline	Size	Explanation			
Data	Data type	4 Byte	The position data type will be output 0: Pulse 16: Base 17: Robot 18: Tool 19: User coordinates			
	Configuration	4 Byte	The configuration will be output bit 0: Back bit 1: Lower arm bit 2: No flip bit 3: R axis $\geq$ 180° bit 4: T axis $\geq$ 180° bit 5: S axis $\geq$ 180°			
	Tool number	4 Byte				
	User coordinate number	4 Byte				
	Extended configuration	4 Byte	Output the 7-axis robot extended configuration bit 0: $\Theta \ge 180^{\circ}$ bit 1: $\Theta \ge 180^{\circ}$ bit 2: $\Theta \ge 180^{\circ}$ bit 3: $\Theta \ge 180^{\circ}$ bit 4: $\Theta W \ge 180^{\circ}$			
	1st axis data	4 Byte	The following values will be output:			
	2nd axis data	4 Byte	For pulse: Fach axis' pulse value			
	3rd axis data	4 Byte	For base:			
	4th axis data	4 Byte	Length (µm)			
	5th axis data	4 Byte	Angle (0.0001°)			
	6th axis data	4 Byte	Data on each axis is set in			
	7th axis data	4 Byte	ascending order.			
	8th axis data	4 Byte	existing axis.			

30. In the watch table click a new line, click the ... and then select the Axes array variable and the Start Transfer bools to be added to the watch table. Press OK after highlighting both.



31. Go online by pressing the gear, download the program and then press the play button to run the PLC. Your screen should look like this after you are done.

File Edit Many Declark Build Online Debug Tech Mindaux Hide						
rile Edit view Project Build Online Debug loois Window Help						
物 📽 🖬 👝 😋 苯 🐘 能 🗙 🗰 馀 🏘 馀 👫 📜 🦉 🦉 🕼 🕅 🐄 🕥 🕮 Application (Device PI (	Looid + 08 08 .	A2103 93 45 13	이 이 봐. ㅠ	- 死/		
Devices • 🔻 🗙 💮 Device 😭 EtherNet_IP_Scanne	Generic_Ether	Net_IP_device	PLC_PRG X			
Chatted1 Device.Application.PLC_PRG						
= 😔 💮 Device [connected] (CODESYS Control Win V3 x64)				Type Va	lue Prenar	Address
P D PLC Logic				SHED D	inc inclusion	
Application [run]     A antAttributeSinde				ENTR C		
- III Library Manager				ENID Ce		
PLC_PRG (PRG)     A statistical				ENTR C		
Configuration		~ <b>v</b>		La var retro		
G      BUIPScanner10Task (IEC-Tasks)     1						
EtherNet_IP_Scanner.IOCyde						
🖶 😏 🌚 ENIPScannerServiceTask (IEC-Tasks) 🗎 3 💿 getAttributeAll (xExecuteTasks) := st	rtTransfer FALSE,					
EtherNet_IP_Scanner.ServiceCyde     iffEtherNetIPDevice:= Generic_I	therNet_IP_Device,	(* instance of the	device (instan	ce is found	in the I/O I	Image of the
<ul> <li>G MainTask (IEC-Tasks)</li> <li>Class 127 := 10 / 2, dvTasta 10 / 2,</li> </ul>		(* cip	Class which cos	ntains the d	lesired accri	instance 1
C_PRG     TOATA SAMOUTIPOLATALA	(es).	(4 data buffer	4)	ver, range r	100 1 10	And Cance A
Ethernet (Ethernet)	axes). (* )	tize of the data bu	ffer 4)			(* attribute
Some Section 1 (1) Scanner (EtherNet/IP Scanner) (1) xDone (2000) -> done (2000) (1) xDone (2000) (1) xDone (2000) -> done (2000) (1) xDone (2000)						
C      EtherNiet_IP_Scanner (EtherNiet_IP Scanner)     xDone [7155] > done [7155],     xBusy[7165] > busy[7165],     xBusy[7165] > busy[7165],						
G @ EtherNet_IP_Scamer (EtherNet_IP Scamer)     xDone (Intel®, IP device (Generic EtherNet_IP, device (Generic EtherNet_IP, device (Generic EtherNet_IP, device (Generic EtherNet_IP, device (Generic EtherNet_IP), device (Generic EtherNet_IP)						
S    S    S    S    S    S    S	udiReceivedData 0					
Control and the second se	udiReceivedData 0	7				
B • O ■ Therete, P. Samer Etheriet/P Same) • O ■ Genera, Etheriet, P. Jewice (Genera Etheriet, B. Jewice), and the structure (Genera Etheriet) and the structure (Genera Etheriet), and the s	udiReceivedData 0					
Control and the set of the s	udiReceivedData 0	); () instance of the	device (instan	ce is found	in the T/O I	Trace of the
■ O Therete, P. Source Etheric(P. Source) Source (Etheric, P. Jource (Etheric)) O G Generic, Etherict, P. Joives (Generic Ether Interverse (Etherict)) O G Generic, Etherict, P. Joives (Generic Ether Interverse) Interverse Intervers	udiReceivedData 0 DEEE, therNet_IP_Device,	; (* instance of the (* cim	device (instan: class which cou	ce is found	in the I/O I	Image of the
G G Generic Effectet (P, Source (Sected))     Source (Sected) Source()     Source (Sected) Source	udiReceivedData 0	(* instance of the (* cip (* value o	device (instan class which com f 0 is class let	ce is found ntains the c wel, range f	in the I/O I lesired attri rom 1x is	Image of the
S → O → Control (LP) Scare (Etherlet(JP) Scare) = x0000 (2000 + 3000 + 3000 (2000 + 3000 + 3000 (2000 + 3000 + 3000 (2000 + 3000 + 3000 + 3000 (2000 + 3000	udiReceivedData 0	(' instance of the (' cip (' value o (' data buffer	device (instan class which cou f is class let 4)	ce is found ntains the c wel, range f	in the I/O 1 lesired attri 'rom 1x is	Image of the ibute *) instance le
G G thereix(P, Scarver Etheric(P) Scarver)     Society (Scarver)     Society (Scar	udiReceivedData 0 ttes, ttherNet_IP_Device, txes), xxes), (* 1	(' instance of the (' cip (' value o (' data buffer size of the data bu	device (instan class which cou f 0 is class let *) ffer *)	ce is found ntains the c vel, range f	in the I/O I Wesired attri from 1x is	Image of the ibute *) instance le (* attribute
Control of the field P. Source (Sherled I) Sou	udiReceivedData 0 udi, therNet_IP_Device, tes), xxes), (* 4	; (* instance of the (* cip (* value o (* data buffer pize of the data bu	device (instan class which co f 0 is class let *) ffer *)	ce is found ntains the c vel, range f	in the I/O 1 lesired attri from 1x is	Image of the ibute ") instance le (" attribute
<pre>short future(P_Picaner Shorte(U) Samme) short future(P_Picaner Shorte(U) Samme) short future(P_Picaner Shorte(U) Samme) short future(Picaner Shorte(U) Samme) short future(Picaner</pre>	udiReceivedData 0 UGG, therNet_IP_Device, tes), xxes), (* 1	; (* instance of the (* cip (* value o (* data buffer pize of the data bu	device (instan class which con class is class let *) ffer *)	ce is found ntains the c rel, range f	in the I/O 1 lesired attri from 1x is	Image of the ibute *) instance le (* attribute
S G G Tennet D. Jonner Sherle(D Samer)     S G G G G G G G G G G G G G G G G G	udiReceivedData 6	(' instance of the (' cip (' value o (' data buffer pize of the data bu	device (instan class which co f 0 is class let *) ffor *)	ce is found ntains the c vel, range f	in the I/O 1 lesired attri from 1x is	Image of the ibute *) instance le (* attribute
<pre>s 0 0 therete P_score Shere(c)F Scorer)</pre>	udiReceivedData 0 253. therNet_IP_Device, tes), axes), (* 1	; (' instance of the (' cip (' value o (' data bu ise of the data bu	device (instam class which co f 0 is class let *) ffer *)	ce is found stains the c vel, range i	in the I/O I lesired attri 'rom 1x is (	Image of the ibute *) instance 10 (* attribute • 4
Control of the rest P. Source (Beneric)     Source (Steering P. Source)     Source (Steer	udiReceivedData 3	; (* instance of the (* circ) (* value o (* data buffer ize of the data bu	device (instan class which co f 0 is class let i) ffer *) Prepared value	ce is found ntains the c vel, range f Execution poin	in the I/O I legired attri 'rom 1x is (	Image of the Libute 4) instance 14 (4 attribute * 4
S G General Etherheit/P Source (Serverd)     S store <u>state</u> → door <u>state</u> → state = d	udiBeceivedData 6	); (* instance of the (* cip (* value o (* data but ise of the data but Value 1	device (instance) class which count for solars lev () Prepared value	ce is found ntains the o vel, range f Execution poin Cydd Monitoring	in the I/O 1 legined attri from 1x is	Image of the ibute 4) instance 10 (4 attribute - 9
S G G Tentret_P_Source (Beneric)     S S G G G G G G G G G G G G G G G G	udiReceivedData 8	); (* instance of the (* cip (* value o (* data but cise of the data but Value 1 0	device (instan class vhich cor fo is class let *) ffor *) Prepared value	ce is found stains the o rel, range f Execution poir Cyclc Monitoring Cyclc Monitoring	in the I/O I leaired attri from 1x is (	Image of the ibste ') instance 1/ (' attribute - 4
	udiReceivedData 8	); (' instance of the (' cip (' value o (' data bu value of the data bu Value 1 0 7442	device (instance) class which con f) is class let ') ffer ') Prepared value	ce is found ntains the o vel, zange f Execution poir Cycle Monitorin Cycle Monitorin	in the I/O 1 Menired attri from 1x is	Image of the ibute ') instance le (' attribute * 9
S G G thereite, P. Source (Bereite)     S source (Second)	udiBaceivedData 3	); (' instance of the (' cup (' value o (' data but value of the data but value 1 0 f4450	device (instan class which co f 0 is class let *) ffor *) Prepared value	ce is found stains the c vel, range f Execution poin Cycle Monitoring Cycle Monitoring Cycle Monitoring	in the I/O 1 esired attri rom 1x is (	Image of thi ibute *) instance 1: (* attribute * 3

32. On the robot make sure you have a position value in #P1 other than all 0's so you know the value is read and then transferred. In my case I have the below position values.



33. Click the box beside the Start Transfer bool under the prepared value heading until it changes to TRUE. Then right click and write value. This will run the program one time, it will read P1 and copy it into the Axes array and then copy it back to the robot and place it in P2.



34. You should see the axes array populate with your position and if you look at P2 on the robot you should see the values from P1

Unböled I	Device.Application.PLC_PRG			201				
Device [connected] (CODESYS Control Win V3 x64)     Device [connected] (CODESYS Control Win V3 x64)	Expression					Type Value	Prepar Address	Comm_ ^
= O Application [run]	· reset					ENIP.R		fb to res
- 🎁 Library Manager	getAttributeSingle     setAttributeSingle					ENDP.G.,		fb to set
PLC_PRG (PRG)	a getAttributeAl					ENIP.G		fb to ge
Task Configuration     Set FNIPScamper[OTask (FIIC-Tasks)				A W			4	
Benerick_P_Same: Doyle     Oblight DDPSame: Doyle     Oblight DDPSame: Service Craiks)     Oblight DDPSame: ServiceCycle     Oblight DDPSame: ServiceCycle     Oblight DDPSame: ServiceCycle     Oblight DDPSame: DDPSame: ServiceCycle     Oblight DDPSame: ServiceCycle     ServiceCycle     Oblight DDPSame: Ser	Image: State	<pre>startTran vice:- Generic_EthrNet := 16975, := 1, stAta5_:= ADR(ares), st</pre>	afer <b>izzeta</b> . t_IP_Device, (* s	(* instance of (* (* valu (* dets bu: iie of the det	the device (insta cip class which c be of 0 is class 1 ffer ') a buffer ')	nce is found in t ontains the desir evel, range from	he I/O Image of the ed attribute ') 1x is instance is (' attribute	(device) *) (vel *) ( po, 6 of t)
	atron [Main]     atron [Main]       ii     atron [Main]       iii     udikestvelkaii       iii     atron [Main]       iiii     atron [Main]       iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	<pre>Error[ZTEG], Size 0 =&gt; udiRec startTransfer[ZTEG], vice:= Generic_ItherHel := 16f97, := 16f97, := 27, dirads := ADR(axes), sciented: sciented</pre>	eivedDate 6	(* instance of (* (* val (* deta bus vize of the data	the device (insta cip class which c or of 0 is class 1 ffer ") a buffer ")	nce is found in t catains the desir evel, range from	he I/O Image of the ed attribute 4) 1x is instance 10 (* attribute	• device) *) nvel *) • no. 6 of ti 100 % @
					_			
	Watch 1			1			• 4	X Messages -
	Expression	Application	Type	Value	Prepared value	Execution point		Ad Download
	%IB0	Device. Application	BYTE	1		Cyclic Monitoring		%I Descriptio
	14Q80	Device.Application	BYTE	0	_	Cyclic Monitoring	1	%Q A core
	# PLC_PRG.startTransfer	Device. Application	BOOL AND AND AND	PALSE	_	Cyclic Monitoring		
	PLC_PRG.axes	Device. Application	ARRAY [012]			Cyclic Monitoring		-
	axes(o)		DWICKD	10		Cyclic Monitoring		-
	aves[1]		DWIDED	0		Cyclic Monitoring		-
	accel_j		DWORD	0		Cyclic Monitoring		
	axes[4]		DWORD	0		Cyclic Monitoring		-
	avec[5]		DWORD	55101000		Cyclic Monitoring		-
	# axes[6]		DWORD	465000		Cyclic Monitoring		-
	aves[7]		DWORD	465000		Cyclic Manitoring		-
	axes[8]		DWORD	650000		Cyclic Monitoring		-
	a mar[9]		DWORD	640000		Cycle Menitoring		_
	* aves[10]		DWICRD	590000		Cyclic Monitoring		-
	@ aves[11]		DWORD	0		Cyclic Monitoring		-
	# axes[12]		DWORD	ů.		Cyclic Monitoring		-
DATA EDIT	b in axes array	<b>y.</b> UTILITY	12			10	) (†)	Þ
	OSITION VARIAB POO2 BASE	LE	NAME					
	1 :X 55101 Y 465	.000 000 TOOL	• <u>60</u>					
	Z 465	.000 <typ< td=""><td>E&gt;</td><td>1 67</td><td>100</td><td></td><td></td><td></td></typ<>	E>	1 67	100			
VARIABLE	Ry 64.1	0000 FR		- 185 R<	180			
B001	Rz 59.0	0000 FL	.IP	<u>T&lt;</u>	180			
IN/OUT								
inCut								
ROBOT								

Note the values in P2.

Simple Menu

SYSTEM INFO

35. Congratulations you have successfully set up Implicit and Explicit EtherNet/IP connections from a Codesys Control Win V3 X64 to a YRC1000. Codesys is the basis for a bunch of other PLC brands including but not limited to Wago, Festo, Lenze, Schneider, Eaton and Beckhoff so this documentation may help assist in setup for those PLC's as well.

LP .

1/F Panel

PAGE

MotoLaser

Safety guard is open