# Acontis Stack PowerPMAC setups

# V1.8

November 24, 2017

**Davis Gentry** 

This document is intended to allow an engineer who has basic familiarity with the PowerPMAC and with Sysmac Studio to commission a PowerPMAC using an EtherCAT network using the Acontis EtherCAT stack. At the time of writing the Acontis stack is used in two Omron PowerPMAC products – the µPowerPMAC (CK3E series motion controllers) and the PowerPMAC IPC (NY512-A600 series IPC with motion control). In this document the term 'PowerPMAC' will be used generically to refer to any Omron Delta Tau motion controllers using the Acontis stack.

Note: one powerful method of troubleshooting is to connect a micro USB cable to the diagnostics port on the PowerPMAC (beside the power connector on the CK3E). The PC will detect it as a serial port - COM27 on my PC. Connect using the software tool PuTTy, serial type, 115200 baud. Default settings with the configuration shown will work:

Repute the second secon	X
Category:	
Category: Session Logging Terminal Keyboard Bell Features Window Peetares Window Selection Selection Colours Colours Colours Colours Connection Data Proxy Telnet Rlogin Sele	Basic options for your PuTTY session         Specify the destination you want to connect to         Serial line       Speed         COM27       115200         Connection type:       Raw         Raw       Telnet       Rlogin         Saved Sessions       Saved         ppmac       Load         Default Settings       Load         save       Delete
About	Close window on exit Always Never Only on clean exit

Once it starts, press the PC enter key once, log in with ID: root, Password: deltatau

# <u>Step 1</u>

Install the latest PowerPMAC IDE. As a part of that EC Engineer will be installed – this is the software needed to set up the Acontis stack.

Open the PowerPMAC IDE and connect to the PowerPMAC from your computer. I suggest opening a Watch window and adding ECAT[0].Enable and ECAT[0].MasterState. Make sure that ECAT[0].Enable=0 before making any changes to the EtherCAT configuration (can be changed in the Terminal window if connected). Once the EtherCAT configuration is set and downloaded to the PowerPMAC, and a 'save' (write working memory to permanent memory) and '\$\$\$' (reset)

has been done ECAT[0].MasterState should be greater than 0. 1, 2, or 8 are acceptable values. After setting ECAT[0].Enable = 1 from the terminal window (or code) there will be a brief delay, then you will see ECAT[0].Enable=1 in the Watch window. If there are any errors, or if the network is incorrectly set up, then ECAT[0].Enable will immediately reset to zero after you set it to one. This typically means that something about the EtherCAT configuration in the PowerPMAC does not match the configuration in the EtherCAT network. It could be the clocks, the network order, or anything that causes an EtherCAT error.

# Step 2

Connect the slave(s) to the EtherCAT network and power the slave(s) and the PowerPMAC.

At this point issuing the 'ecat slaves' (ECAT[0].enable can equal 0 or 1 and still see the same response) from the terminal window should show everything on the network. Vendor IDs and Product Codes are shown under the hardware sections.



# **EC Engineer**

Make sure that you have added the Omron ESI files – the best way is to go to the ESI Manager from the File menu item in EC Engineer. Click the 'Add Folder' button at the bottom.



Navigate to C:\Program Files (x86)\OMRON\Sysmac Studio\IODeviceProfiles\EsiFiles and choose the SystemEsiFiles directory.

Browse For Folder	X
Select the folder which contains the ESI files:	
▲ 📙 IODeviceProfiles	•
Description Descripti Description Description Description Description Descr	
📙 DBConnection	
📙 DdfFiles	
📜 DeviceParameters	
EipConnection	
⊿ 📜 EsiFiles	=
📙 SystemEsiFiles	
📙 UserEsiFiles	
📜 xsd	
	▼
	OK Cancel

It may tell you that one or more ESI files already exist and ask if you want to add them. Say yes. It may have errors in some of the ESI files – for example the Omron E3NW-ECT.xml file cannot be read at the date of writing. If you have added several items by hand it may be easier to delete all and start from scratch. When finished you should have all of the Omron devices:

Jeice	t an E	SI file which should be delete	ed or exported o	r add new ESI files.		
•	10010	OMRON Corporation				
	•	Omron GX-Digital IO.xml				
	•	Omron GX-Encoder.xml				
	►	Omron GX-IO-Link.xml				_
	•	Omron GX-JC.xml				
	►	Omron GX-JC06-H.xml				
	►	Omron NX_Coupler.xml				
	►	Omron R88D-1SN01H-ECT.xn	h			
	►	Omron R88D-1SN01L-ECT.xm	I			
	►	Omron R88D-1SN02H-ECT.xn	h			
	►	Omron R88D-1SN02L-ECT.xm	I.			
	►	Omron R88D-1SN04H-ECT.xn	าไ			
	►	Omron R88D-1SN04L-ECT.xm	I			
	►	Omron R88D-1SN06F-ECT.xm	l.			
	•	Omron R88D-1SN08H-ECT.xn	h			
					Number of ESI fil	les: 10
					Number of device	es: 202

With the PowerPMAC and the EtherCAT devices connected and powered open EC Engineer and choose EtherCAT Master Unit (Delta Tau).



Click on the Class-A-Master in the Project Explorer (upper left pane) and set your network cycle time in microseconds – note that the example below is set to 500 usec, or 2000 Hz. This should match the value in Sys.ServoPeriod in the Power PMAC – if Sys.ServoPeriod = 0.5, then cycle time should by 500 usec. Also set the IP address of your PowerPMAC – note that 192.168.1.200 is the default address for the PowerPMAC. Then click the 'Select' button to the right of the 'Master-Instance' block in the 'Slaves connected to remote system section'.

➡ EC-Engineer []	to an a state from a		X
File View Network Settings Help			
Configuration Mode Happort ENI Export ENI			5
Project Explorer Device Editor			
Class-A Master Master			
General			
Unit Name	Class-A Master		
Cycle Time [us]	500		
Source MAC address			Ī.
Slaves connected to local system			
Network Adapter	Local Area Connection (Intel/D) Ethernet Connection 1017 (M.)		
netroit / deper	Local Area Connection (Intel(R) Ethernet Connection 1217-LM )	Calaat	
Slaves connected to remote syste	em		
IP Address	192.168.1.200		
Port	6000		Ĩ
Master-Instance	0	Deselect	

Add a line to the Global Definitions.pmh file to set the servo (and EtherCAT) frequency. This must match the value used in EC Engineer. This will look like:

Sys.ServoPeriod=0.5 //2 kHz 1.0 //1kHz

Remember to add a startup PLC (see appendix D) when done.

On ALL devices it is best to set the Check Identification on the Advanced Options tab. This forces it to check to make sure that the node is in the correct position on the network.

Device Editor	_	
General Modules PDO Mapping	Variables	Advanced Options
Startup Checking		
Check Vendor ID		
Check Product Code		
Check Revision Number		
==	]	
Check Serial Number		
Identification Checking Check Identification 2 Dec Hex	2	

Some gotchas:

- 1) Make sure that the node address switches are set. For best results be sure that the nodes number from 1 up, with node 1 providing the clock and being the first node in the physical network.
- 2) Node 1 should always be set up as a Potential Reference Clock.

# Step 3

Set up your various hardware as in the next sections.

# 1S and G5 Drives

If the drive has previously been connected to a controller do an initialization (see appendix A). You will also need to do this before changing EtherCAT update frequencies. Using Sysmac Studio, set any desired parameters in the drive (Appendix B). You will probably need to change the Communications error counts (either 0 or f). Set the desired nodes using the rotary switches on the drive. Note that the motor setup utility in the PowerPMAC IDE gives unreliable setup parameters in some cases – easiest to set everything manually. Example settings are given in this document.

The 'ecat slaves' command with a 100 W 100V 1S drive will show Omron VID = \$83 (0x83), and PC = \$AB (0xAB).

The Cyclic Position mode will get your drives running well, with the only caveat being that they must be tuned in the drive rather than the PowerPMAC. If the application requires very tight coordination of multiple axes during the moves you may want to increase the servo frequency and go to Cyclic Torque control. This will be covered after Cyclic Position setup.

Open EC Engineer and scan the network (Appendix C). Click on the drive (Slave\_001 in this example), then click on the Advanced Options tab in the Device Editor pane. Make sure that Potential Reference Clock is checked in the Distributed Clocks settings.

🛹 EC-Engineer []	And have been strated on the local factor of		
File View Network Settings Help			
Configuration Mode 🚽 Export ENI	Kan Diagnosis Mode		
Project Explorer	Device Editor		
<ul> <li>Class-A Master</li> </ul>	General Modules PDO Mapping Variables Advanced Options Distrib	buted Clock   Init Commands   CoE Object-Diction	onary
Slave_001 (0001)			
	Startup Checking	Timeouts	
	Check Vendor ID	SDO Access:	0 🚭 [ms]
	Check Product Code	Init->Pre-Op/Init->Bootstrap:	3000 🖶 [ms]
	Check Revision Number	Pre-Op->Safe-Op/Safe-Op->Op:	10000 😅 [ms]
	==	Back to Pre-Op, Init:	5000 🗢 [ms]
	Check Serial Number	Op->Safe-Op:	200 😂 [ms]
	Identification Checking	Mailbox Mode	
	Check Identification	Cyclic	10 [ms]
	1 Hex Write to EEPROM	State Change	
	Process Data Mode	Overwrite Mailbox Size	
	Disable LRW	Output Size:	E[bytes]
		Input Size:	[bytes]
	Overwrite Watchdog		
	✓ Set Multiplier (Reg.: 0x400): 2498		
	✓ Set PDI Watchdog (Reg.: 0x410): 1000 🚭 (100.000 ms)		
	✓ Set SM Watchdog (Reg.: 0x420): 1000 🚭 (100.000 ms)		
	Distributed Clasks		
Classic View Flat View Topology View	Potential Reference Clock		

Click on the Distributed Clock tab in the Device Editor pane. You should see that the Sync Unit Cycle (us) is the same as the Master cycle time set prior to the scan. Check the Overwrite Mode box. Under Sync Unit 0 set the Cycle Time to User Defined and make sure that it matches the Sync Unit Cycle (us), then set the Shift Time (us) to 125.00.

Project Explorer	Device Ed	itor		_			_				
<ul> <li>Class-A Master</li> </ul>	General	Modules	PDO Mapping	Variables	Advanced Options	Distributed Clock	Init Co				
Slave_001 (0001)	Distribu	ited Clock									
	Opera	tion Mode		DC for syn	chronization	~					
	Sync L	Jnit Cycle (u	s)	500							
	Overw	rite Mode		1							
	V S	ync Units									
	Sync Unit 0										
		Cycle	Time								
		C	Sync Unit Cycle	e	x 1 50	0 us					
		C	User defined		500						
		Shift T	ime (us)		125						
		Svnc Un	it 1								

Click on the PDO Mapping in the Device Editor pane. See note 1.

If you are using Cyclic Position mode, no EtherCAT safety, and do not need any data not included in the 258 PDOs, make no changes here. If you plan to use EtherCAT Safety and already have the safety controller set up, add the 273 PDO mappings. The 258 PDO mappings include:

puts				- 1	Output	Jutputs			
Digital inputs	0x60FD:00	32		•	-	Ist receive PDO Mapping (excluded by 0x1701)			0x1600
▼ 🛛 258th transmit PDO N	Mapping		0x1B01			Name	Index	Bit Length	
Name	Index	Bit Length				Controlword	0x6040:00	16	
Error code	0x603F:00	16		=		Target position	0x607A:00	32	
Statusword	0x6041:00	16				Touch probe function	0x60B8:00	16	
Position actual value	0x6064:00	32			•	▼			0x1701
Torque actual value	0x6077:00	16				Name	Index	Bit Length	
Following error actual va	0x60F4:00	32				Controlword	0x6040:00	16	
Touch probe status	0x60B9:00	16				Target position	0x607A:00	32	
Touch probe pos1 pos v	0x60BA:00	32				Touch probe function	0x60B8:00	16	
Touch probe pos2 pos vi	0x60BC:00	32				Physical outputs	0x60FE:01	32	
Digital inputs	0x60FD:00	32			•	259th receive PDO N	/apping (excluded by 0x17	01)	0x1702

If planning to use Cyclic Torque or Cyclic Velocity control I suggest leaving the 259 transmit PDO mapping, deselecting the 258<sup>th</sup> receive PDO mapping, and selecting the 261th receive PDO Mapping.

Inputs					•	Dutpu	its				
•	Ist transmit PDO Mapping (excluded by 0x1B01)     0x1A00		-	-	261th receive PDO Mapping						
	Name	Index	Bit Length				Name	Index	Bit Length		
	Statusword	0x6041:00	16				Controlword	0x6040:00	16		
	Position actual value	0x6064:00	32				Target position	0x607A:00	32		
	Touch probe status	0x60B9:00	16		=		Target velocity	0x60FF:00	32		
	Touch probe pos1 pos value	0x60BA:00	32				Target torque	0x6071:00	16		
	Touch probe pos2 pos value	0x60BC:00	32				Modes of operation	0x6060:00	8		=
	Error code	0x603F:00	16				Touch probe function	0x60B8:00	16		
	Digital inputs	0x60FD:00	32				Max profile velocity	0x607F:00	32		
-	258th transmit PDO Ma	apping		0x1B01			Positive torque limit value	0x60E0:00	16		
	Name	Index	Bit Length				Negative torque limit value	0x60E1:00	16		
	Error code	0x603F:00	16			-	262th receive PDO Ma	pping (excluded by 0x1704)		0x1705	
	Statusword	0x6041:00	16				Name	Index	Bit Length		
	Position actual value	0x6064:00	32		*		Controlword	0x6040:00	16		Ŧ

The default is synchronous position mode. If not using synchronous position mode, go to the Init Commands tab, and set the x6060 parameter to according to:

```
8 = Cyclic synchronous position mode (csp)
9 = Cyclic synchronous velocity mode (csv)
10 = Cyclic synchronous torque mode (cst)
```

Project Explorer	Device Edi	itor					_		
<ul> <li>Class-A Master</li> <li>Slave_001 [R88D-1SN01L-ECT] (0001)</li> <li>001: Module 1 (Safety Process Data)</li> </ul>	General Init Con	Modules	PDO Mappi	ng Variabl	es Advanced Options	Distributed Clock	Init Commands	CoE Object-Diction	
	Transi	tion	Protocol	Index	Value				
	Pre-O	p->Safe-Op	CoE	0x1C12:000	01 00 04 17				
	Pre-O	p->Safe-Op	CoE	0x1C13:000	01 00 01 1B	1 00 01 1B			
	Pre-O	p->Safe-Op	CoE	0x1A00:000	07 00 10 00 41 60 20 00 6	64 60 10 00 B9 60 20	00 BA 60 20 00 BC 6	60 10 00 3F 60 20 00 F	
	Pre-O	p->Safe-Op	CoE	0x1600:000	03 00 10 00 40 60 20 00 7	7A 60 10 00 B8 60			
	Pre-O	p->Safe-Op	CoE	0x1C12:000	01 00 01 17				
	Pre-O	p->Safe-Op	CoE	0x1C13:000	01 00 01 1B				
	Pre-O	p->Safe-Op	CoE	0x6060:000	10				
	Pre-O	p->Safe-Op	CoE	0x2002:002	1				

Use System Setup tool (after enabling Ecat[0].Enable=1) to set the motor up. Choose Amplifier 131xxxx, ID \$AB. If cannot get enable to stay on once set up, set Motor[x].EcatAmpFaultLimit to 500 or above. If system setup is giving

problems try closing the IDE and restarting it. If you have used this drive before, especially with NX Safety, you may need to hook up THROUGH an NJ using EtherCAT and Sysmac Studio and initialize, build new project, and enable the drive once before you can get the 87 (ESTO) error to clear and / or the motor to enable.

This assumes that the EtherCAT header file created from the EC Engineer xml file looks something like (sample is 261 receive PDO mapping):

```
#define Slave_0_6040_0_Controlword ECAT[0].IO[0].Data
#define Slave_0_607A_0_Targetposition ECAT[0].IO[1].Data
#define Slave_0_60FF_0_Targetvelocity ECAT[0].IO[2].Data
#define Slave_0_6071_0_Targettorque ECAT[0].IO[3].Data
#define Slave 0 6060 0 Modesofoperation ECAT[0].IO[4].Data
#define Slave_0_60B8_0_Touchprobefunctio ECAT[0].IO[5].Data
#define Slave_0_607F_0_Maxprofilevelocit ECAT[0].IO[6].Data
#define Slave 0 603F 0 Errorcode ECAT[0].IO[4096].Data
#define Slave_0_6041_0_Statusword ECAT[0].IO[4097].Data
#define Slave_0_6064_0_Positionactualval ECAT[0].IO[4098].Data
#define Slave_0_6077_0_Torqueactualvalue ECAT[0].IO[4099].Data
#define Slave_0_6061_0_Modesofoperationd ECAT[0].IO[4100].Data
#define Slave_0_60B9_0_Touchprobestatus ECAT[0].IO[4101].Data
#define Slave 0 60BA 0 Touchprobepos1pos ECAT[0].IO[4102].Data
#define Slave_0_60BC_0_Touchprobepos2pos ECAT[0].IO[4103].Data
#define Slave_0_60FD_0_Digitalinputs ECAT[0].IO[4104].Data
```

When using the 261 receive PDO mapping you will need to set a few variables in order to run your motors. Assuming the mapping above, add the following to your startup PLC (or in operational PLC for dynamic changes):

```
//Set up for torque control
Slave_0_607F_0_Maxprofilevelocit=838860000; //6000 rpm units are cts/ms 419430000; //3000 rpm units
are cts/ms max velocity
Slave_0_60E0_0_Positivetorquelim=3000; //300% positive torque limit
Slave_0_60E1_0_Negativetorquelim=3000; //300% negative torque limit
Slave_0_6060_0_Modesofoperation=10; //set for torque mode
/*Mode of Operation value Description
0 Not specified
1 Profile position mode (pp)
3 Profile velocity mode (pv)
6 Homing mode (hm)
8 Cyclic synchronous position mode (csp)
9 Cyclic synchronous velocity mode (csv)
10 Cyclic synchronous torque mode (cst)
*/
```

Note that once you have these basics done, you can use the Setup tool to set the motor up, and the Tuning tool to tune it once the setup is done. Note that everything shown here is using motor units of counts.

Some variables to set after the initial motor setup is done are:

```
Motor[1].JogTa=- -0.002 //0.002 ms^2/ct
Motor[1].JogTs= -0.0005 //0.0005 ms^3/ct
Motor[1].AbortTs=25
Motor[1].MaxSpeed=400000
Motor[1].JogSpeed=280000
```

One potential gotcha – there is a one cycle delay after enabling the motor – so if the motor is not enabled before telling it to move you may get a fatal following error. Best way to fix this is to enable your motors at the start of operation, wait a couple of milliseconds, then start your moves. If the motor is ever killed, repeat this.

Some PowerPMAC variables for a 1S (with a demo motor) sitting at the first node and setup on Motor[1] are:

//Motor 1 – note this setting is for PDO map 259 target position Motor[1].Ctrl=Sys.PosCtrl // Sys.ServoCtrl for torque control Motor[1].pDac=ECAT[0].IO[1].Data.a //ECAT[0].IO[3].Data for torque control Motor[1].MaxDac=3000 //300% Motor[1].JogTa=- -0.002 Motor[1].JogTs= -0.0005 Motor[1].AbortTs=25 Motor[1].MaxSpeed=400000 Motor[1].JogSpeed=280000 Motor[1].pEnc2=EncTable[32].a Motor[1].AmpEnableBit=0 Motor[1].LimitBits=64 Motor[1].pAmpEnable=ECAT[0].IO[0].Data.a Motor[1].pAmpFault=ECAT[0].IO[4097].Data.a Motor[1].pLimits=ECAT[0].IO[4104].Data.a Motor[1].AmpFaultBit=3 Motor[1].pEnc=EncTable[32].a motor[1].FatalFeLimit=8000000 Motor[1].Control[0]=\$10000f00 Motor[1].Control[1]=\$0 EncTable[32].type=1 EncTable[32].index1=0 EncTable[32].index2=0 EncTable[32].index3=0 EncTable[32].index4=0 EncTable[32].index5=0 EncTable[32].pEnc1=Sys.pushm EncTable[32].pEnc=ECAT[0].IO[4098].Data.a EncTable[32].MaxDelta=0 EncTable[32].ScaleFactor=1 EncTable[32].index6=0

# NX I/O

# NOTE: Major caveat – using the revision 1.5 NX-ECC203 EtherCAT coupler:

This version of the NX-ECC203 card can only have ONE (total) of any of the following cards in the rack, and that one card must be in slot 1: position interface units, safety CPUs. When using these cards (and also any safety i/o cards) It must also be put in distributed clock more – NOT free run.

Note: ECC201 and ECC202 support only Free Run mode. To use with synchronized clocks MUST have an ECC203. Must use with synchronized clocks in order to use NX i/o to input motor feedback or to do motor control. Apply power and open Sysmac Studio. Going to the Configurations and Setup section double click on the EtherCAT sub section. If not already present, drag the NX-ECC20x which you are using into the EtherCAT network and drop it. In the example below I am using an older revision coupler, so I checked the 'Show all versions' box in the toolbox to see the Rev 1.1.

File Edit View Insert Project	Controller Simulation	Tools Help			
	a 5 × 3 5	# # 🖲 🕏 🔺 🔌	63 🌮 庵 🖬 🖸 🖫 🕼		
Multiview Explorer 🗸 👎	Node2 : NX-ECC201 (E	002) TherCAT X		•	Toolbox 🗸 🖡
new_Controller_0  Configurations and Setup	Node Address Network cor	nfiguration Master Master E001	Item name	Value	All vendors  Groups All groups
EtherCAT	1	R88D-1SN30F-ECT Rev:1.0	Model name Product name	Master	Servo Drives
▼      Node1 : R88D-1SN30F- ∟      Parameters	2	NX-ECC201 Rev:1.1	Number of Slaves PDO Communications Cycle	2 1000 us	<ul> <li>Frequency Inverter</li> <li>Digital IO</li> <li>Analog IO</li> </ul>
∟ 🗟 FFT ∟ 🖾 Data Trace Settings			Reference Clock Total Cable Length	Not exist 1000 m	Encoder Input
L Node2 : NX-ECC201 (EC ► S CPU/Expansion Racks			Fail-soft Operation Setting Wait Time for Slave Startup	Fail-soft operation	Input Keyword Show all versions
I/O Map			Revision Check Method Serial Number Check Method	Setting <= Actual device  No check	NX-ECC201 Rev:1.0 NX-ECC201 EtherCAT coupler V1.0
L □ Operation Settings			DC Synchronous Correction	Disable slave monitoring option 🛛 🔻	NX-ECC201 Rev:1.1 NX-ECC201 EtherCAT coupler V1.1
L # Built-in EtherNet/IP Por ★ Motion Control Setup					NX-ECC201 Rev:1.2 NX-ECC201 EtherCAT coupler V1.2
er Cam Data Settings					NX-ECC202 Rev:1.2 NX-ECC202 EtherCAT coupler V1.2
Task Settings					NX-ECC203 Rev:1.3 NX-ECC203 EtherCAT coupler V1.3
🖂 Data Trace Settings					NX-ECC203 Rev:1.4 NX-ECC203 EtherCAT coupler V1.4

Connect to the NX coupler using a USB cable, and disconnect the EtherCAT cable. Right click on the NX node and select Coupler Connection (USB)-> Online. Double click on the NX node to open the NX configuration page.

Multiview Explorer 🚽 👎	Mode2 : NX-ECC201 (E002) 🗙			
new_Controller_0		Item name		Value
<ul> <li>Configurations and Setup</li> </ul>		Device name	E002	
▼   EtherCAT		Model name	NX-ECC201	
▼ @ Node1:R88D-1SN30F-		Product name	EtherCAT Coupler	
		Unit version	1.1	
		NX Unit Number	0	
		NX Unit Mounting Setting		
L I Data Trace Settings		Serial Number		
Node2 : NX-ECC201 (ECC201)		Supply Power/Available Power	0.00 / 10.00	w
		Unit width	46	mm
<ul> <li>CPU/Expansion Racks</li> <li>I/O Map</li> <li>Controller Setup</li> </ul>		I/O allocation settings	NX Unit Registration NX Unit I/O Data Acti Sysmac Error Status :	Status 125 : 128 [bits] ive Status 125 : 128 [bits] 8 [bits]
V R Controller Setup				Edit I/O Allocation Settings
L □ Operation Settings		11-14		
∟ # Built-in EtherNet/IP Por		Unit operation settings		Edit Unit Operation Settings
►		Number of mounted Units		
Cam Data Settings</th <td></td> <td>NX Unit Connection Time</td> <td></td> <td>sec</td>		NX Unit Connection Time		sec
g Cam Data Settings		Serial Number Check Method	No check	
Event Settings				
Task Settings		Coupler Connection (USB)		Online
Data Trace Settings				Offline
Programming				

Right click on the coupler in the node tab and choose 'clear all memory'. Accept clearing coupler and all modules. Depending on modules attached this may take a couple of minutes to complete. Right click on the NX node and select Coupler Connection (USB)-> Offline. Cycle power on the unit. When this completes, right click on the NX node and select Coupler Connection (USB)-> Online. Right click on the coupler in the node tab and choose 'Compare and Merge with Actual Unit Configuration'. A window with the existing modules should pop up.

Sompare and Merge with Actual Unit Configuration			
Configuration on Sysmac Studio	Configuration on Sysmac Studiol NX-ECC201 Ver.1.1	Result Matched Added Added Added Added Added Added	Actual Unit C NX-ECC201 NX-0D4256 NX-AD2608 NX-DA2605 NX-AD3208 NX-EC0142 ' NX-SL3300 \
Actual Unit Configuration	۲		

Click the 'Apply Actual Unit Configuration' button in the middle. Everything should match now.



Click the 'OK' button.

Now you can click on the various modules in the Node tab and select 'Edit Unit Operation Settings' for modules as needed. When finished editing for each module be sure to press the 'Transfer to Unit' button.

Node2 : NX-ECC201 (E002) ×	Unit 2[Node2]:NX-AD2ra	•
	Item name	Value
	Device name	N2
	Model name	NX-AD2608
	Product name	Analog Input Unit
	Unit version	1.0
	NX Unit Number	2
	NX Unit Mounting Setting	Enabled
	Serial Number	0x0000000
	Power consumption	1.05 W
	Unit width	12 mm
		Ch1 Analog Input Value : 16 [bits]
	I/O allocation settings	Ch2 Analog Input Value : 16 [bits]
		Edit I/O Allocation Settings
	Unit exerction acttings	
	Unit operation settings	Edit Unit Operation Settings

One important Analog Input module note – if you do not have inputs connected you must disable the input to avoid faults.



When all modules are configured, right click on the NX node and select Coupler Connection (USB)->Transfer to Coupler. Select the 'Configuration information + Unit operations settings + Unit application data' in the popup.



If you have a safety controller you may now select the new\_SafetyCPU in the Multiview Explorer pane and configure and program it.



When done with the SafetyCPU (if any) go back to the controller, right click on the NX node and select Coupler Connection (USB)-> Offline. Power down, disconnect the USB cable, connect the EtherCAT cable(s), and power up again.

The 'ecat slaves' command with a NX-ECC201 EtherCAT coupler will show Omron VID as \$83 (0x83) and PC as \$83 (0x83).

Open EC Engineer and scan the network (Appendix C). Click on the NX coupler (Slave\_001 in this example) and open the subsections to see the modules.



You should now see all the modules on the NX coupler. If the modules are not there and you try to start up using the configuration PuTTy will show the following error (on node 6) when you try to set Ecat[0].Enable=1:

COM27 - PuTTY		X
ioctl waiting for reguest		
ecats	et	
rStateEx() Error 0x98110024 in McSm state 'START' for requested state 'START'		
EcMa	st	
rt: ecatSetMasterState to SAFEOP returned with code 0x98110024		
Cannot set master s	ta	
SAFEOP (Result = ERROR: Slave error (0x98110024))		
Cannot get DCM status! ERROR: F	ea	
not supported (0x98110001)		
ECMASterStart: Job times during startup <init> to <preo< td=""><td>₽&gt;</td><td></td></preo<></init>	₽>	
starstart. Getting ecatGetMastarState		
EcMasterStart: Latest ecatMasterState: 2		
Echapocipolitici Latobo contrabioritatici. L	st	
rt: Getting ecatGetMasterState		
EcMasterStart: Latest ecatMasterState: 2		
Slave error		
006 [NX-ECC203]": - EtherCAT address=6 - State <pre error="" operational="">(0x12), co</pre>	nt 🖌	
tatus <invalid configuration="" output="">(0x1d)</invalid>		
RAS Connection changed: Established!		
RAS	C	

If this happens go back to EC-Engineer and press the Load Modules button in the Device Editor pane:



Click on the NX slave in the Project Explorer pane.

Project Explorer	Device Editor						
<ul> <li>Class-A Master</li> </ul>	General Modules PDO Mapping Va	riables Advanced Options Distributed Clock Init Commands CoE Object-Dictionary					
<ul> <li>Slave_001 (0001)</li> <li>001: Module 1 (NX-OD4256)</li> <li>002: Module 2 (NX-AD2608)</li> <li>003: Module 3 (NX-DA2605)</li> </ul>	Address Station Address Information						
004: Module 4 (NX-AD3208)	Name	Slave_001					
005: Module 5 (NX-EC0142)	Description	NX-ECC201 EtherCAT coupler V1.1					
006: Module 6 (NX-SL3300)	Vendor	OMRON Corporation (0x00000083)					
	Product Code	0x00000083 (131)					
	Revision Number	0x00010001 (65537)					
	ESI File	C:\ProgramData\EC-Engineer\EtherCAT\Omron NX_Coupler.xml					
	Identification Value	Not Used					
	Ports						
	A	Class-A Master					
	D	Not Available					
	в	Not Connected					
	C 🌒	Not Available					

Click on the 'Modules' tab in the Device Editor pane. You should see all of the correct modules.

Click on the 'PDO Mapping' tab in the Device Editor pane. Select the appropriate Input and Output PDOs for your modules. By default all of the correct PDOs are already selected. See note 1.

Click on the 'Modules' tab in the Device Editor pane. You should see all of the variables transmitted for each PDO selected. If any data here is wrong you will need to check your ESI file

Click on the 'Advanced Options' tab in the Device Editor pane. If you have a drive in the network, or are using the ECC201 coupler, make sure that Potential Reference Clock is NOT checked in the Distributed Clocks settings.



Click on the 'Distributed Clock' tab in the Device Editor pane. Set the Operation Mode to 'Free Run'.

Project Explorer	[	Device Edi	tor					
<ul> <li>Class-A Master</li> </ul>		General	Modules	PDO Mapping	Variables	Advanced Options	Distributed Clock	i II
Slave_001 (0001)								
001: Module 1 (NX-OD4256)		Distribu	ted Clock					
002: Module 2 (NX-AD2608)		Operat	tion Mode		FreeRun			
003: Module 3 (NX-DA2605)		Sync U	Init Cycle (u	s)	500			
004: Module 4 (NX-AD3208)		Overw	rite Mode					
005: Module 5 (NX-EC0142)		5	une Unite					
006: Module 6 (NX-SL3300)								
			Sync Uni	it 0				

The EtherCAT header file created from the EC Engineer xml file looks like:

<pre>#define Slave_0_7040_2_Ch2AnalogOutputVa pshm-&gt;ECAT[0].I0[2].Data #define Slave_0_7088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].I0[3].Data #define Slave_0_7084_1_Ch1LatchFunction pshm-&gt;ECAT[0].I0[4].Data #define Slave_0_3003_4_NXUnitRegistratio(x) pshm-&gt;ECAT[0].I0Buffer[0 + x ] #define Slave_0_3006_4_NXUnitIODataActiv(x) pshm-&gt;ECAT[0].IOBuffer[17 + x ] #define Slave_0_2022_1_SysmacErrorStatus pshm-&gt;ECAT[0].I0[4098].Data #define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4099].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4101].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].I0[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].I0[4104].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].I0[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].I0[4101].Data</pre>	#define	Slave_0_7040_1_Ch1AnalogOutpu	tVa pshm->ECAT[0].IO[1].Data
<pre>#define Slave_0_7088_1_ChlPulsePeriodMea pshm-&gt;ECAT[0].I0[3].Data #define Slave_0_7084_1_ChlLatchFunction pshm-&gt;ECAT[0].I0[4].Data #define Slave_0_3003_4_NXUnitRegistratio(x) pshm-&gt;ECAT[0].I0Buffer[0 + x ] #define Slave_0_3006_4_NXUnitIODataActiv(x) pshm-&gt;ECAT[0].I0Buffer[17 + x ] #define Slave_0_2002_1_SysmacErrorStatus pshm-&gt;ECAT[0].I0[4098].Data #define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4099].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].I0[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].I0[4104].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].I0[4106].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].I0[4108].Data #define Slave_0_6084_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6085_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].I0[4110].Data</pre>	#define	Slave_0_7040_2_Ch2AnalogOutpu	tVa pshm->ECAT[0].IO[2].Data
<pre>#define Slave_0_7084_1_ChlLatchFunction pshm-&gt;ECAT[0].I0[4].Data #define Slave_0_3003_4_NXUnitRegistratio(x) pshm-&gt;ECAT[0].I0Buffer[0 + x ] #define Slave_0_3006_4_NXUnitIODataActiv(x) pshm-&gt;ECAT[0].I0Buffer[17 + x ] #define Slave_0_2022_1_SysmacErrorStatus pshm-&gt;ECAT[0].I0[4098].Data #define Slave_0_6020_1_ChlAnalogInputVal pshm-&gt;ECAT[0].I0[4099].Data #define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4101].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].I0[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].I0[4103].Data #define Slave_0_6060_4_Ch4AnalogInputVal pshm-&gt;ECAT[0].I0[4104].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].I0[4105].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].I0[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].I0[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].I0[4110].Data</pre>	#define	Slave_0_7088_1_Ch1PulsePeriod	Mea pshm->ECAT[0].IO[3].Data
<pre>#define Slave_0_3003_4_NXUnitRegistratio(x) pshm-&gt;ECAT[0].IOBuffer[0 + x ] #define Slave_0_3006_4_NXUnitIODataActiv(x) pshm-&gt;ECAT[0].IOBuffer[17 + x ] #define Slave_0_2002_1_SysmacErrorStatus pshm-&gt;ECAT[0].IO[4098].Data #define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4099].Data #define Slave_0_6060_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4101].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat</pre>	#define	Slave_0_7084_1_Ch1LatchFunction	on pshm->ECAT[0].IO[4].Data
<pre>#define Slave_0_3006_4_NXUnitIODataActiv(x) pshm-&gt;ECAT[0].IOBuffer[17 + x ] #define Slave_0_2002_1_SysmacErrorStatus pshm-&gt;ECAT[0].IO[4098].Data #define Slave_0_6020_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4099].Data #define Slave_0_6060_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4101].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6081_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6088_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat #define Slave_0_6086_1_Ch1LatchInput2Dat</pre>	#define	Slave_0_3003_4_NXUnitRegistra	<pre>tio(x) pshm-&gt;ECAT[0].IOBuffer[0 + x ]</pre>
<pre>#define Slave_0_2002_1_SysmacErrorStatus pshm-&gt;ECAT[0].IO[4098].Data #define Slave_0_6020_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4099].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4100].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat #define Slave_0_6086_1_Ch1LatchInput2Dat</pre>	#define	Slave 0 3006 4 NXUnitIODataAc	<pre>stiv(x) pshm-&gt;ECAT[0].IOBuffer[17 + x ]</pre>
<pre>#define Slave_0_6020_1_ChlAnalogInputVal pshm-&gt;ECAT[0].IO[4099].Data #define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4100].Data #define Slave_0_6060_1_ChlAnalogInputVal pshm-&gt;ECAT[0].IO[4101].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_4_Ch4AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6081_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_2002_1_SysmacErrorSta	tus pshm->ECAT[0].IO[4098].Data
<pre>#define Slave_0_6020_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4100].Data #define Slave_0_6060_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4101].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_4_Ch4AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6081_1_Ch1ResetExternalI pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6020_1_Ch1AnalogInput	Val pshm->ECAT[0].IO[4099].Data
<pre>#define Slave_0_6060_1_Ch1AnalogInputVal pshm-&gt;ECAT[0].IO[4101].Data #define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6081_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6020_2_Ch2AnalogInput	Val pshm->ECAT[0].IO[4100].Data
<pre>#define Slave_0_6060_2_Ch2AnalogInputVal pshm-&gt;ECAT[0].IO[4102].Data #define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6081_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6060_1_Ch1AnalogInput	Val pshm->ECAT[0].IO[4101].Data
<pre>#define Slave_0_6060_3_Ch3AnalogInputVal pshm-&gt;ECAT[0].IO[4103].Data #define Slave_0_6060_4_Ch4AnalogInputVal pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6081_1_Ch1ResetExternalI pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6060_2_Ch2AnalogInput	Val pshm->ECAT[0].IO[4102].Data
<pre>#define Slave_0_6060_4_Ch4AnalogInputVal pshm-&gt;ECAT[0].IO[4104].Data #define Slave_0_6080_1_Ch1EncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6081_1_Ch1ResetExternalI pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6060_3_Ch3AnalogInput	Val pshm->ECAT[0].IO[4103].Data
<pre>#define Slave_0_6080_1_ChlEncoderCounter pshm-&gt;ECAT[0].IO[4105].Data #define Slave_0_6081_1_ChlResetExternalI pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_ChlEncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_ChlPulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_ChlLatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_ChlLatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_ChlLatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6060_4_Ch4AnalogInput	Val pshm->ECAT[0].IO[4104].Data
<pre>#define Slave_0_6081_1_Ch1ResetExternalI pshm-&gt;ECAT[0].IO[4106].Data #define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].IO[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave_0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6080_1_Ch1EncoderCour	ter pshm->ECAT[0].IO[4105].Data
<pre>#define Slave_0_6082_1_Ch1EncoderPresent pshm-&gt;ECAT[0].I0[4107].Data #define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].I0[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].I0[4110].Data #define Slave 0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].I0[4111].Data</pre>	#define	Slave 0 6081 1 Ch1ResetExterr	all pshm->ECAT[0].IO[4106].Data
<pre>#define Slave_0_6088_1_Ch1PulsePeriodMea pshm-&gt;ECAT[0].IO[4108].Data #define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].IO[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave 0_6086_1_Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6082_1_Ch1EncoderPres	ent pshm->ECAT[0].IO[4107].Data
<pre>#define Slave_0_6084_1_Ch1LatchStatus pshm-&gt;ECAT[0].I0[4109].Data #define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].I0[4110].Data #define Slave 0 6086 1 Ch1LatchInput2Dat pshm-&gt;ECAT[0].I0[4111].Data</pre>	#define	Slave_0_6088_1_Ch1PulsePeriod	Mea pshm->ECAT[0].IO[4108].Data
<pre>#define Slave_0_6085_1_Ch1LatchInput1Dat pshm-&gt;ECAT[0].IO[4110].Data #define Slave 0 6086 1 Ch1LatchInput2Dat pshm-&gt;ECAT[0].IO[4111].Data</pre>	#define	Slave_0_6084_1_Ch1LatchStatus	pshm->ECAT[0].IO[4109].Data
#define Slave 0 6086 1 Ch1LatchInput2Dat pshm->ECAT[0].IO[4111].Data	#define	Slave_0_6085_1_Ch1LatchInput1	Dat pshm->ECAT[0].IO[4110].Data
	#define	Slave_0_6086_1_Ch1LatchInput2	Dat pshm->ECAT[0].IO[4111].Data

#### **NX Safety**

In Sysmac Studio go to the Safety CPU, then the Global Variables (Programming->Data->Global Variables). Define any desired variables to exchange between the NX Safety Controller and the PowerPMAC. Major note: At this time DO not use BOOLs – use INTs for data transferred between the NX Safety CPU and the PowerPMAC.

Unit 0[Node1]:NX-ECCratio	n 🜌 Global Varia	ibles 🗙 跖 Progra	im0	
Name	Data Type	Initial Value	Constant	Expose
sbConnectionStatus	SAFEBOOL	FALSE		Do not expose
sbOutputStatus	SAFEBOOL	FALSE		Do not expose
sbOut1	SAFEBOOL	FALSE		Do not expose
sbOut2	SAFEBOOL	FALSE		Do not expose
bSafetyStatus	INT	0		Output
bSafetyOutputStatus	INT	0		Output
bSafetyOut1	INT	0		Output
bSafetyOut2	INT	0		Output
bSafetyRun	INT	0		Input
bSafetyReset	INT	0		Input

In the I/O Map section (Configurations and Setup->I/O Map) build any hardware variables necessary for the application.

÷	I/O Map	×					
	Position	Port	R/W	Data Type	Variable	Variable Comment	Variable Type
		🔻 👰 EtherCAT Network					
	EtherCA	Master					
	Node1/I	▼ NX-SOH200					
		▼ Status					
		Safety Connection Status	R	SAFEBOOL	sbConnectionStatu		Global Variables
		Safety Output Terminal Status	R	SAFEBOOL	sbOutputStatus		Global Variables
		▼ Safety Outputs					
		So00 Output Value	w	SAFEBOOL	sbOut1		Global Variables
		So01 Output Value	w	SAFEBOOL	sbOut2		Global Variables

Program the Safety Controller as desired. Note that this document will not go into safety programming or general use of the NX Safety Controller. Note that the program shown here is NOT a valid safety program – it will run and will operate the safety output based on input from the PowerPMAC, but is NOT meant to be used in a safety system.

- d-11-NIV FCC anti-up hand Clark				
Gloc	ai variables	ogramu x		
	Namo		1	Data Turna
bSafetyRun 1	NT_TO_BOOL	lbSafetyRun <b>TR</b>	UE	
bSafetyReset 0	INT_TO_BOOL	— lbSafetyReset	FALSE	
	SF_OutCon	ntrol_0		
TRUE	Activate	Ready-	TRUE	sbOut1 TRUE
TRUE	S_SafeControl	S_OutControl		
lbSafetyRun TRUE	ProcessControl	Error	FALSE	
	StaticControl	DiagCode -	32768	
_	S StartReset			
	S_AutoReset			
lbSafetyReset DALSE	Reset			
	BOOL_TO_INT			
sbOutputStatus TRUE		- bSafetyOutp	outStatus	1

Validate the safety program, return to RUN mode, then go back to the NX controller side, save, build, and download to the controller. Afterwards cycle power on the entire system.

When all is done here, go to the Controller section in Sysmac Studio, double click on the EtherCAT section, then right click on the Master and select 'Export All Couplers' I/O Allocations'. Keep track of where you put the file created. Go to that directory in Windows Explorer and Extract All from the zip file. Open the CouplerMemoryMap.xml using Internet Explorer. It is very important that the Offsets seen here match those in later steps.

Device	PDO Mapping	PDO Entry												
Name	Name	Index	SM	Offset	Size	Name	Index	Data Type	Offset	Size				
E001	TxPDO													
	Slot0(NX-ECC203)505th transmit PDO Mapping	#x1BF8	3	0.00	16.00	NX Unit Registration Status 63	#x2003:03	ARRAY [07] OF BYTE	0.00	8.00				
						NX Unit I/O Data Active Status 63	#x2005:03	ARRAY [07] OF BYTE	8.00	8.00				
	Slot0(NX-ECC203)512th transmit PDO Mapping	#x1BFF	3	16.00	1.00	Sysmac Error Status	#x2001:01	USINT	16.00	1.00				
	Slot0(NX-ECC203)PaddingTxPdo	#x1BF4	3	17.00	1.00					1.00				
	Slot1(NX-SL3300)Input Data Set 1	#x1A00	3	18.00	6.00	Node1/Unit2	#x6000:01	ARRAY [05] OF BYTE	18.00	6.00				
	Slot1(NX-SL3300)Input Data Set 2	#x1A01	3	24.00	10.00	Safety CPU Status	#x6004:01	UINT	24.00	2.00				
						bSafetyStatus	#x6001:02	INT	26.00	2.00				
						bSafetyOutputStatus	#x6001:03	INT	28.00	2.00				
						bSafetyOut1	#x6001:04	INT	30.00	2.00				
						bSafetyOut2	# <b>x6001:0</b> 5	INT	32.00	2.00				
	Slot2(NX-SOH200)Input Data Set 1	# <b>x1A</b> 04	3	34.00	6.00	FSoE Slave CMD	#x6020:01	USINT	34.00	1.00				
						Safety Input 1st Byte	#x6021:01	USINT	35.00	1.00				
						FSoE Slave CRC_0	#x6020:03	UINT	36.00	2.00				
						FSoE Slave Conn_ID	#x6020:02	UINT	38.00	2.00				
	Slot2(NX-SOH200)Input Data Set 2	#x1A05	3	40.00	2.00	Standard Input 1st Byte	#x6022:01	USINT	40.00	1.00				
						Standard Input 2nd Byte	#x6022:02	USINT	41.00	1.00				
	RxPDO													
	Slot1(NX-SL3300)Output Data Set 1	#x1600	2	0.00	6.00	Node1/Unit2	#x7000:01	ARRAY [05] OF BYTE	0.00	6.00				
	Slot1(NX-SL3300)Output Data Set 2	#x1601	2	6.00	4.00	bSafetyRun	#x7001:01	INT	6.00	2.00				
						bSafetyReset	#x7001:02	INT	8.00	2.00				
	Slot2(NX-SOH200)Output Data Set 1	# <b>x16</b> 04	2	10.00	6.00	FSoE Master CMD	#x7020:01	USINT	10.00	1.00				
						Safety Output 1st Byte	#x7021:01	USINT	11.00	1.00				
						FSoE Master CRC_0	#x7020:03	UINT	12.00	2.00				
						FSoE Master Conn_ID	#x7020:02	UINT	14.00	2.00				
	Slot2(NX-SOH200)Output Data Set 2	# <b>x16</b> 05	2	16.00	2.00	Standard Output 1st Byte	#x7022:01	USINT	16.00	1.00				
						Standard Output 2nd Byte	#x7022:02	USINT	17.00	1.00				

#### In EC Engineer select the safety controller, then click the Variables tab.

Device Editor											
MDP Slot Propert	ies Variables										
Variables											
Name			<b>^</b>	Datatype	Master Sync Unit	Offset	S				
Edit Variable				$\sim$							
Move Up	Move Down			New	Edit	Delet	e				
New Alias	Edit Alias	Delete Alias		$\sim$			Click 'New'.	Add an A	RRAY[01]	of BYTE fr	om SL300

Input Data Set 1, checking the 'Combine' box, Count=3. Add a UINT from SL3300 Input Data Set 2, using a Count of 1. Add an Array[0..1] of BYTE from SL3300 Input Data Set 2, using a count of 4. Add an ARRAY[0..1] of BYTE from SL300 Output Data Set 1, checking the 'Combine' box, Count=3. Add an Array[0..1] of BYTE from SL3300 Output Data Set 2, using a count of 2. The variables shown will initially be xxxxData Set y.Variable n. You can click on each variable, then click 'Edit' and rename the variable. Note: if have odd number in array size of BYTEs, can use type USINT, count = size of array, and check 'combine'. Here is final result:

#### Device Editor

.. . . .

MDP Slot Properties Variables

vai	Tables					
	Name	Datatype	Master Sync Unit	Offset	-	Size
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.Variable 0	ARRAY [02] OF UINT	Id 0: Default 0	IN :	18.0	6.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.Safety CPU Status	UINT	Id 0: Default 0	IN :	24.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyStatus	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	26.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOutputStatus	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	28.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOut1	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	30.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOut2	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	32.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Variable 0	ARRAY [02] OF UINT	Id 0: Default 0	OUT :	0.0	6.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 2.bSafetyRun	ARRAY [01] OF BYTE	Id 0: Default 0	OUT :	6.0	2.0
	Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 2.bSafetyReset	ARRAY [01] OF BYTE	Id 0: Default 0	OUT :	8.0	2.0

Do the same exercise for any other safety I/O modules in the rack. My rack has only one SOH200, and the variable list for that is auto created, looking like:

MDP Slot Properties Variables

.. . . .

les					
Name	Datatype	Master Sync Unit	Offset	-	Size
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 1.FSoE Slave CMD	USINT	Id 0: Default 0	IN :	34.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 1.Safety Input 1st Byte	USINT	Id 0: Default 0	IN :	35.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 1.FSoE Slave CRC_0	UINT	Id 0: Default 0	IN:	36.0	2.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 1.FSoE Slave Conn_ID	UINT	Id 0: Default 0	IN:	38.0	2.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 2.Standard Input 1st Byte	USINT	Id 0: Default 0	IN :	40.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Input Data Set 2.Standard Input 2nd Byte	USINT	Id 0: Default 0	IN :	41.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 1.FSoE Master CMD	USINT	Id 0: Default 0	OUT :	10.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 1.Safety Output 1st Byte	USINT	Id 0: Default 0	OUT :	11.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 1.FSoE Master CRC_0	UINT	Id 0: Default 0	OUT :	12.0	2.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 1.FSoE Master Conn_ID	UINT	Id 0: Default 0	OUT :	14.0	2.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 2.Standard Output 1st Byte	USINT	Id 0: Default 0	OUT :	16.0	1.0
Slave_001 [NX-ECC203].Module 2 (NX-SOH200).Output Data Set 2.Standard Output 2nd Byte	USINT	Id 0: Default 0	OUT :	17.0	1.0

NOTE: If you have added your own safety variables of type BOOL for the Safety Controller the autogenerated offsets may be wrong. Check the offsets to make sure they match those from the Sysmac Studio generated file. If not you will have to delete these variables and create new ones – you may have to change to an INT type.

Click on the Master in the Project Explorer, then the Slave to Slave tab. You will need to associate the Safety I/O slices with the Safety CPU. The Safety CPU Input Data Set 1 needs to be associated with EACH of the Safety I/O card's Output Data Set 1. The Safety CPU Output Data Set 1 needs to be associated with EACH of the Safety I/O card's Input Data Set 1. This is done by clicking on the Input Data Set in question in the left pane, then clicking on the Output Data Set in question in the right pane, then clicking on the '>>' between the panes. For one SOH200 it should look like this at the end:

Devi	ce Ed	itor			_							
Ma	ster	Process Data Image	Variables	Advanced Options	Slave to Slave	Distributed Clocks	Tasks + S	Sync Units				
Sla	Slave to Slave											
I	nputs						Outputs					
	*	Slave_001 [NX-EC	C203]				- 1	Slave_001 [NX-ECC203]				
	)	💉 505th transmit	t PDO Map	ping [128 Bits]				Source And the American Set 1 (1997) Set 2 (				
	)	💉 512th transmit	t PDO Map	ping [8 Bits]		>>		Module 1 (NX-SL3300).Output Data Set 2 [32 Bits]				
	)	💉 501st transmit	PDO Mapp	ing [8 Bits]		x		🛷 Module 2 (NX-SOH200).Output Data Set 1 [48 Bits]				
	)	Module 1 (NX	-SL3300).Inj	out Data Set 1 [48 Bit	s]			💉 Module 2 (NX-SOH200).Output Data Set 2 [16 Bits]				
	)	💉 Module 1 (NX	-SL3300).Inj	out Data Set 2 [80 Bit	s]							
	)	Module 2 (NX-	-SOH200).Ir	nput Data Set 1 [48 B	its]							
	)	💉 Module 2 (NX	-SOH200).Ir	nput Data Set 2 [16 B	its]							

# NX Safety – Local Rack safety I/O only

See above for example.

# NX Safety – Local Rack safety I/O plus one 1S drive

Use normal setup for the NX rack, including the safety above. Use normal setup for the 1S, but include the 273 transmit and receive PDOs in the PDO mapping (EC Engineer). When adding the variables for the Safety CPU you need to add variables for the 273 Transmit and Receive PDOs. The memory used for each drive will be divided by:

	INDEX	SIZE (BYTEs)	Description	ADDRESS	ТҮРЕ	Offset (from 0)	SIZE (BYTEs)	New Variable Type & Number		
273th	#x1B10	7								
transmit			FSoE Slave CMD	#xE600:01	USINT	0	1	USINT		
PDO			STO command	#x6640:00	BOOL	1	0.01			
Mapping			Padding - BOOL				0.06	BOOL 8		
			error acknowledge	#x6632:00	BOOL	1.07	0.01			
			Padding - BOOL				0.07			
			Safety Connection	#xE601:01	BOOL	2.07	0.01	BOOL 8		
			FSoE Slave CRC_0	#xE600:03	UINT	3	2	UINT		
			FSoE Slave Conn_ID	#xE600:02	UINT	5	2	UINT		
273th	#x1710	7								
receive			FSoE Master CMD	#xE700:01	USINT	0	1	USINT		
PDO			STO command	#x6640:00	BOOL	1.00	0.01			
Mapping			Padding - BOOL				0.06	BOOL 8		
			error acknowledge	#x6632:00	BOOL	1.07	0.01			
			Padding - BYTE			2	1	USINT		
			FSoE Master CRC_0	#xE700:03	UINT	3	2	UINT		
			FSoE Master	#xE700:02	UINT	5	2	UINT		

The drive is Node 2, and the CouplerMemoryMap.xml for this particular section looks like - Input data set:

Node2 #x6000:02 ARRAY [06] OF BYTH	18.00 6.00	ARRAY [05] OF BYTE 18	#x6000:01	Node1/Unit2	14.00	18.00	3	#x1A00	Slot1(NX-SL3300)Input Data Set 1
	24.00 7.00	ARRAY [06] OF BYTE 24	#x6000:02	Node2					
Padding #x6000:03 ARRAY [0.0] OF BYTH	31.00 1.00	ARRAY [00] OF BYTE 31	#x6000:03	Padding					

# Output data set:

Slot1(NX-SL3300)Output Data Set 1	#x1600 2	0.00	14.00	Node1/Unit2	#x7000:0	ARRAY [05] OF BYTE 0.00 6.00
				Node2	#x7000:02	ARRAY [06] OF BYTE 6.00 7.00
				Padding	#x7000:03	ARRAY [00] OF BYTE 13.00 1.00

Note that Dataset 2 in this example contains communications variables for use with the PowerPMAC, and that I have renamed the variables from default to give a logical description. So for a NX rack with one SL3300, one SOH200, and one 1S drive the Safety CPU input variables will look like:

Name	Datatype	Master Sync Unit	Offset	<b>^</b>	Size
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.Variable 0	ARRAY [02] OF UINT	Id 0: Default 0	IN :	18.0	6.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.DriveFSoECmd	USINT	Id 0: Default 0	IN:	24.0	1.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.DriveSTOCmd	BOOL	Id 0: Default 0	IN :	25.0	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool1	BOOL	Id 0: Default 0	IN :	25.1	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool2	BOOL	Id 0: Default 0	IN:	25.2	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool3	BOOL	Id 0: Default 0	IN :	25.3	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool4	BOOL	Id 0: Default 0	IN:	25.4	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool5	BOOL	Id 0: Default 0	IN :	25.5	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool6	BOOL	Id 0: Default 0	IN :	25.6	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.DriveFSoEErrorAck	BOOL	Id 0: Default 0	IN :	25.7	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool8	BOOL	Id 0: Default 0	IN:	26.0	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool9	BOOL	Id 0: Default 0	IN:	26.1	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool10	BOOL	Id 0: Default 0	IN:	26.2	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool11	BOOL	Id 0: Default 0	IN:	26.3	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool12	BOOL	Id 0: Default 0	IN:	26.4	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool13	BOOL	Id 0: Default 0	IN:	26.5	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PadBool14	BOOL	Id 0: Default 0	IN:	26.6	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.SafetyConnStatus	BOOL	Id 0: Default 0	IN :	26.7	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.unDriveCRC_0	UINT	Id 0: Default 0	IN:	27.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.unDriveConnId	UINT	Id 0: Default 0	IN:	29.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 1.PaddingByte	ARRAY [00] OF BYTE	Id 0: Default 0	IN:	31.0	1.0
t					
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.SafetyCPUStatus	UINT	Id 0: Default 0	IN :	32.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyStatus	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	34.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOutputStatus	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	36.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOut1	ARRAY [01] OF BYTE	Id 0: Default 0	IN :	38.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Input Data Set 2.bSafetyOut2	ARRAY [01] OF BYTE	Id 0: Default 0	IN:	40.0	2.0

### And output variables will look like:

Name	Datatype	Master Sync Unit	Offset	-	Size
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Variable 0	ARRAY [02] OF UINT	Id 0: Default 0	OUT :	0.0	6.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.DriveFSoEMasterCmd	USINT	Id 0: Default 0	OUT :	6.0	1.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.DriveSTOCmd	BOOL	Id 0: Default 0	OUT :	7.0	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad1	BOOL	Id 0: Default 0	OUT :	7.1	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad2	BOOL	Id 0: Default 0	OUT :	7.2	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad3	BOOL	Id 0: Default 0	OUT :	7.3	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad4	BOOL	Id 0: Default 0	OUT :	7.4	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad6	BOOL	Id 0: Default 0	OUT :	7.5	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.Pad7	BOOL	Id 0: Default 0	OUT :	7.6	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.DriveErrorAck	BOOL	Id 0: Default 0	OUT :	7.7	0.1
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.BytePad1	USINT	Id 0: Default 0	OUT :	8.0	1.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.DriveFSoEMasterCRC_0	UINT	Id 0: Default 0	OUT :	9.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.DriveFSoEMasterConnId	UINT	Id 0: Default 0	OUT :	11.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 1.BytePad2	USINT	Id 0: Default 0	OUT :	13.0	1.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 2.bSafetyRun	ARRAY [01] OF BYTE	Id 0: Default 0	OUT :	14.0	2.0
Slave_001 [NX-ECC203].Module 1 (NX-SL3300).Output Data Set 2.bSafetyReset	ARRAY [01] OF BYTE	Id 0: Default 0	OUT :	16.0	2.0

When setting the Slave to Slave mapping in EC engineer under the Class-A Master, set the NX mappings as above, then include the mapping of the NX Safety CPU to the 1S drive(s).

**\*Note 1:** You can check the PDO mappings for the various pieces of EtherCAT hardware by going into Sysmac Studio and comparing them. The PDO mapping of the same number should have the same variables with the same type.

# <u>Step 4</u>

Once all the Slaves are set up (including at a minimum PDOs and clocks), choose the 'Save' option from the File menu item at the upper left of the EC Engineer window. Save the file where desired. Then click the 'Export ENI' button in the upper section of the EC Engineer window. Note where this file is being saved and give it a name with no spaces in the name. From the PowerPMAC IDE **make certain that ECAT[0].Enable=0** – otherwise any attempt to complete the next steps will fail, but may not show an error.

# Step 5

From the PowerPMAC IDE choose 'System Setup' under the 'Tools' menu item. Click on the PowerPMAC object in the upper left corner of the System Setup window. Select the same servo frequency that was used in EC Engineer, and click 'Accept'. Here we are using 2 kHz:

PowerPMAC System Setup : Online[1]     File Database	92.168.1.200:SSH]	a Brains rate				l	_ 0 %
PowerPMAC (192 168 1 200) → Hardware Diagnosis ⊕ # Motors ★ Master[0] Deactivated							
		Glo	bal Clock Sett	ing			
	Phase Frequency	2	kHz	Items	New clock settings	PowerPMAC Value	Cancel
				Servo Period (msec) PhaseOverServoPeriod	0.500	0.500	-
	Servo Frequency	2.000 💌	kHz				
				EtherCAT option of Gate detected, usi sys.CPUTimerIntr ServoPeriod must On Accepting the d the terminal windo effect. Check 'sys	etected and no ng internal clock = 1. For EtherC be multiple of 6 clock issue save w for changes to ServoTime' to c	Master c by setting CAT the 2.5 usec. a and \$\$\$ in o take yount	Accept

Click on the 'Master[0] Deactivated' object in the left pane of the System Setup window. Now click the 'Browse' button in the center pane. Navigate to the XML file created by EC Engineer, click on it, and click 'OK'.

PowerPMAC System Setup : Online[192	168.1.200:SSH]
File Database	
PowerPMAC (192.168.1.200) Hardware Diagnosis H Motors Master[0] Deactivated	Ether CAT.
	EtherCAT License Status: EtherCAT IO Devices and axis counts up to 256 axes.
	EtherCAT Master type: ACONTIS
	C:\Users\DG096521\OneDrive\Documents\projects - dt\ipc testing\1s2kHzMultiMode xml Browse Download ENI file
	Slave status
	ECAT[0].SlaveCount=1
	ECAT[0].RxTime=0 Ecat[0].Erron=No Erron

Now click the 'Download ENI File' button in the center pane. Assuming that ECAT[0].Enable=0 in the PowerPMAC, there should be no errors in the Setup Messages pane at the lower side of the System Setup window. Right click on the 'Master[0] Deactivated' object in the left pane of the System Setup window, and select 'Export EtherCAT Variables'. Choose a header file name with NO spaces, and click the 'Save' button. If you are overwriting a file name tell it that that is ok when the popup windows appear. The Setup Messages pane should show no errors. Close the PowerPMAC System Setup window. Press the 'OK' button when the popup appears.

Go to the terminal window in the IDE. Type 'save' and enter. Type '\$\$\$' and enter. This saves everything to permanent memory and resets the PowerPMAC. At this point you should be able to type 'ECAT[0].Enable=1' and see that it remains at that value. Depending on how extensive changes are you may need to power cycle the entire system, or even initialize one or more of your slaves. If after doing all of this ECAT[0].Enable refuses to remain =1, there is something wrong with your EtherCAT configuration. Check for errors on your slaves to get an idea where the problem is. Also you can try typing 'ecat slaves' in the terminal window of the IDE – you should see all of your hardware listed. NOTE: I have seen a stubborn error where one drive had the same node address switch setting as the one beside it.

If you have slave errors, but are able to start EtherCAT communications, you may be able to get more information by going into Diagnosis Mode. Here I have chosen the 1S drive object and can see that the Node ID Sector matches the configured Alias value (which it should):

🛹 EC-Engineer []							- 0	X
File View Network Settings Help								
🔀 Configuration Mode 🛛 🚽 Export ENI	sis Mod	e						
Project Explorer	Device Gen Valu	e Editor eral Varia les	bles ESC Register EEPROM Extended Diagnosis DC Diagnosis CoE Objec	t-Dictior	History History Descr	ption fro	m ESI Single Ob	oject
M1 0001 *		Index	Name	Value	T	/pe	Flags	*
		1	Vendor ID		131 (0x83)	UDINT	( RO RO RO )	
		2	Product Code		171 (0xAB)	UDINT	( RO RO RO )	
		3	Revision Number		65536 (0x10000	UDINT	( RO RO RO )	
		4	Serial Number		3159 (0xC57)	UDINT	( RO RO RO )	
		<ul> <li>0x10E0</li> </ul>	Node Address Reload	3 (0x03	U	SINT	( RO RO RO )	
		SubInde	ex Name		Value	Type	Flags	
		1	Configured Station Alias value		1 (0x01)	UINT	( RW RW RW )	
		3	ID-Selector validation		1 (0x01)	UINT	( RW RW RW )	
		0x10F3	Diagnosis History	25 (0x1	9) U	SINT	( RO RO RO )	
		<ul> <li>0x10F9</li> </ul>	Present Time for Event Log	1 (0x01	) U	SINT	( RO RO RO )	Ŧ

Some PowerPMAC variables for a 1S sitting at the first node and setup on Motor[1] are:

```
//Motor 1 – note this setting is for PDO map 259 target position
Motor[1].ServoCtrl=1 //activate processing for motor 1
Motor[1].Ctrl=Sys.PosCtrl //Sys.ServoCtrl for torque control
Motor[1].pDac=ECAT[0].IO[1].Data.a //This address needs to match the address of the drive
                        //Target Position if controlling in Position mode.
Motor[1].MotorTa=-0.000239999993937090039
Motor[1].MotorTs=10
Motor[1].JogTa=-0.00023999999
Motor[1].JogTs=10
Motor[1].AbortTs=25
Motor[1].MaxSpeed=40000
Motor[1].JogSpeed=280000
Motor[1].pEnc2=EncTable[32].a
Motor[1].AmpEnableBit=0
Motor[1].LimitBits=64
                                //bit 0 is MLIM, bit 1 is PLIM
Motor[1].pAmpEnable=ECAT[0].IO[0].Data.a
Motor[1].pAmpFault=ECAT[0].IO[4097].Data.a
Motor[1].pLimits=ECAT[0].IO[4104].Data.a
Motor[1].AmpFaultBit=3
Motor[1].pEnc=EncTable[32].a
motor[1].FatalFeLimit=8000000
Motor[1].Control[0]=$10000f00
Motor[1].Control[1]=$0
EncTable[32].type=1
EncTable[32].index1=0
EncTable[32].index2=0
EncTable[32].index3=0
EncTable[32].index4=0
EncTable[32].index5=0
EncTable[32].pEnc1=Sys.pushm
EncTable[32].pEnc=ECAT[0].IO[4098].Data.a
EncTable[32].MaxDelta=0
EncTable[32].ScaleFactor=1
EncTable[32].index6=0
```

- #define Slave\_0\_6040\_0\_Controlword ECAT[0].IO[0].Data
- #define Slave\_0\_607A\_0\_Targetposition ECAT[0].IO[1].Data
- #define Slave\_0\_60FF\_0\_Targetvelocity ECAT[0].IO[2].Data
- #define Slave\_0\_6071\_0\_Targettorque ECAT[0].IO[3].Data
- #define Slave\_0\_6060\_0\_Modesofoperation ECAT[0].IO[4].Data
- #define Slave\_0\_60B8\_0\_Touchprobefunctio ECAT[0].IO[5].Data
- #define Slave\_0\_607F\_0\_Maxprofilevelocit ECAT[0].IO[6].Data
- #define Slave\_0\_603F\_0\_Errorcode ECAT[0].IO[4096].Data
- #define Slave\_0\_6041\_0\_Statusword ECAT[0].IO[4097].Data
- #define Slave\_0\_6064\_0\_Positionactualval ECAT[0].IO[4098].Data
- #define Slave\_0\_6077\_0\_Torqueactualvalue ECAT[0].IO[4099].Data
- #define Slave\_0\_6061\_0\_Modesofoperationd ECAT[0].IO[4100].Data
- #define Slave\_0\_60B9\_0\_Touchprobestatus ECAT[0].IO[4101].Data
- #define Slave\_0\_60BA\_0\_Touchprobepos1pos ECAT[0].IO[4102].Data
- #define Slave\_0\_60BC\_0\_Touchprobepos2pos ECAT[0].IO[4103].Data
- #define Slave\_0\_60FD\_0\_Digitalinputs ECAT[0].IO[4104].Data

# Appendix A

### **1S Drive Intialization**

Connect up with USB, start Sysmac Studio, add the correct drive to the EtherCAT network. Right click on the node in the Multiview explorer tab, click Direct Connection (USB) -> Online. Then right click on the node and pick Initialize. Choose Parameters and Safety.

▼ ₩ EtherCAT		
∟ 🗆 Node4 : GX-JC06-H(IN,X2,X3) N		
▶	Initializa driva	×
▶ @ Node2 : R88D-KN01L-ECT (E00	Initialize drive	
Node3 : NX-ECC202 (E006) : Of	Initialize drive	
∟ -□ Node5 : GX-JC06-H(X4,X5,X6) S	This function restores all parameters in the	he drive to the factory default settings.
► 🖙 CPU/Expansion Racks	Unit Name:	E003
* I/O Man	Model:	R88D-1SN01L-ECT
	Area:	Parameter objects
► R Controller Setup		Safety objects
▶	Exceptions:	Encoder data
🖌 Cam Data Settings		Event log
► Event Settings	Configuration	
📭 Task Settings	Area to restore:	Parameters and Safety 🔹
🖂 Data Trace Settings	Drive restart:	Yes
► ♣ Host Connection Settings	Transfer parameters from drive:	Yes
Programming		OK Cancel

You will need to set the safety variables now according to how you have the drive wired. If using no safety or if using EtherCAT safety, in the drive parameters set 4637.01 (Error Stop Input) = 0:No allocation.

If using no hard limits in the drive parameters set 4630.01 (Positive Drive Prohibition) = 0:No allocation and 4631.01 (Negative Drive Prohibition) = 0:No allocation.

All	para	meters > All (	Categories > Interface objects > Digital Analog Inp	ut and Output Area 🔻	
		OD 🔽	Description	Value 🔽	Drive Value
	=	4620.04	Encoder Dividing Pulse Output - Output Reverse	0 : Not reverse	
		4630.01	Positive Drive Prohibition Input - Port Selection	0 : No allocation	0
	=	4630.02	Positive Drive Prohibition Input - Logic Selection	1 : Negative logic (NC contact)	
	=	4631.01	Negative Drive Prohibition Input - Port Selection	0 : No allocation	0
	=	4631.02	Negative Drive Prohibition Input - Logic Selection	1 : Negative logic (NC contact)	
		4632.01	External Latch Input 1 - Port Selection	7 : General Input 7 (IN7)	
	=	4632.02	External Latch Input 1 - Logic Selection	0 : Positive logic (NO contact)	
		4633.01	External Latch Input 2 - Port Selection	8 : General Input 8 (IN8)	
	=	4633.02	External Latch Input 2 - Logic Selection	0 : Positive logic (NO contact)	
		4634.01	Home Proximity Input - Port Selection	4 : General Input 4 (IN4)	
	=	4634.02	Home Proximity Input - Logic Selection	0 : Positive logic (NO contact)	
	=	4635.01	Positive Torque Limit Input - Port Selection	0 : No allocation	
	=	4635.02	Positive Torque Limit Input - Logic Selection	0 : Positive logic (NO contact)	
		4636.01	Negative Torque Limit Input - Port Selection	0 : No allocation	
	=	4636.02	Negative Torque Limit Input - Logic Selection	0 : Positive logic (NO contact)	
	=	4637.01	Error Stop Input - Port Selection	0 : No allocation	0
	=	4637.02	Error Stop Input - Logic Selection	1 : Negative logic (NC contact)	

### Appendix **B**

### **1S Drive parameter setting**

Connect up with USB, start Sysmac Studio, add the correct drive to the EtherCAT network. Right click on the node in the Multiview explorer tab, click Direct Connection (USB) -> Online. Clicking on the right arrow beside the node will open the following tree:



Double click on the Parameters. This will open a tab with the drive parameters. The values can be changed. Then either choose the individual parameters, right click, and choose Selection to Drive, or click the Transfer Selection to Drive or the Transfer All to Drive buttons to the lower right:



### Appendix C

### Scan network with EC engineer.

Open EC Engineer. Select EtherCAT Master Unit (Delta Tau) on the Start Page tab. Set the Cycle time in uSeconds to match the Sys.ServoPeriod set in the PowerPMAC. For a 2 kHz servo period Sys.ServoPeriod=0.5, and Cycle Time will be 500. Set the IP Address of the PowerPMAC, keep Port=6000, and probably Master-Instance=0. Click the 'Select' button to the right of the Mast-Instance.

EC-Engineer []		
File View Network Settings Help		
Configuration Mode	🌉 Diagnosis Mode	
Project Explorer	Device Editor	
🚽 Class-A Master	Master	
	General	
	Unit Name	Class-A Master
	Cycle Time [us]	500
	Source MAC address	
	Slaves connected to local system	
	Network Adapter	Local Area Connection 3 ( Cisco AnyConnect Secure Mobility Client Virtual Miniport Ad
		Select
	Slaves connected to remote syste	em
	IP Address	192.168.0.200
	Port	6000
	Master-Instance	0 Select

Right click on the Class-A Master in the Project Explorer pane and select 'Scan EtherCAT Network'.



The slave devices in your network should now appear under the Class-A Master in the Project Explorer pane. If you get errors but are sure that your ESI files are correct, go into the ESI manager, delete the device you are trying to connect, then add the file again. This will clear up errors if a new version of a device was not in the ESI file when you first added it. Errors are shown in the Messages pane of EC engineer. A typical error might be that a device was not found for a particular VendorID (Omron is 0x83), ProductCode (a 100W/120V 1S drive is 0xAB), or Revision (at time of writing 0x10000 for the 1S drives).



Clicking on the slaves should show the station address and information, as well as the associated ESI file in the General tab of the Device Editor pane.

### Appendix D

### **Startup PLC**

In Solution Explorer right click on PMAC Script Language->PLC Programs->Add->New Item... Pick a name for the PLC and click Add button.

Open plc plcStartup local rStartupTimer; while(sys.ecatMasterReady==0){}; ECAT[0].Enable=1; rStartupTimer =Sys.Time+2; //time in seconds while(rStartupTimer >Sys.Time){}; cmd"&1enable"; rStartupTimer =Sys.Time+5; while(rStartupTimer >Sys.Time){}; //cmd"&1b1r"; disable plc 1;

close

Open pp\_startup.txt and add this line:

Enable plc plcStartup