NJ Programming Best Practices

Johnston Hall

October 15, 2014

NJ Best Practices V1.1

Contents

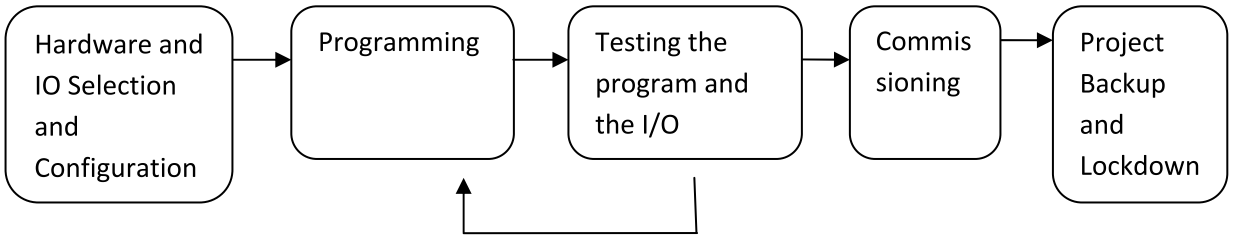
Disclaimer

This startup guide does not replace the Omron manuals concerning the safe startup of equipment.

Summary

The purpose of this paper

Use this as a guideline when configuring and programming an NJ controller so that you get the most out of the system in the least amount of time. The document is created from our programming experiences and may not be ideal for everyone but should provide a good start. This document covers the following stages of programming.


Hardware and IO Selection and Configuration
Programming
Testing the program and the I/O
Commis sioning
Project Backup and Lockdown

Examples would be:

- Best addresses to use for servo and I/O style remote IO
- When to use remote or local I/O (not what you would think)
- How do I address the IO
- Best addressing conventions for I/O
- How best to store permanent setup data.

The structure is set out as follows:

- We mention networks, which ones are available to the NJ.
- Then the I/O is tackled. With tag based programming you need to set up the I/O to create the IO tag names. Some basic rules are given for I/O tag names since a lot of Omron programmers have not used them before. Local and remote I/O are discussed because local I/O on an NJ has no advantage over EtherCAT remote I/O.
- The next section talks about how to store permanent data within the program. Normally this would be done in battery backed memory but with the SD memory card there are better options.
- Next are the tasks. The NJ is a real time processor so it uses tasks instead of interrupts. This can drastically change your programming style so it is important to understand before you program.
- The next section talks about how to add or remove IO to an already installed system. This can affect how you set up the communications wiring for the system.
- Simulation tools for the NJ. Especially for new users it will prove advantageous to test your program to see how the variables work before you write a lot of code.
- Then last of all is how to back-up the program and add passwords.

Fastest Way to Get Started

The NJ uses tag based programming which may require a slightly different startup procedure than what you used on the CJ platform (similar to the Rockwell platform).
The Network configuration works a little different. You can add global variables at any time then change the variables “Network Publish” attribute to Publish, input, or output. So you do not need to create all you Network variables before you program.
The largest change is the naming of the I/O. The NJ does not use addresses like the CJ. Each I/O is assigned a name. To do this properly you need to create the IO table first. Creating the I/O table allows you to create the I/O names which in turn create the Global I/O variables you need to program.

Hardware and I/O Configuration

Network Configuration – Which Network to Use

NJ CPU Ethernet Port

If the NJ CPU is to be connected to the office network then this is the best port to use.

- The NJ SQL version uses this port to talk to the SQL server
- The FTP client and server services are attached to this port.
- The NJ NTP time service for the NJ Clock is on this port.

Other features of this port include:

- This is the only Ethernet port which supports programming of the NJ.
- This port supports socket services which allow you to communicate to devices that do not support Omron or CIP (EtherNet/IP) protocols.

NJ EtherCAT Port

Use the EtherCAT port for remote I/O and connecting to servos, drives, vision systems, and sensors.
All devices on the EtherCAT network - made by Omron - can be programmed using Sysmac Studio via the USB or Ethernet Ports on the CPU.
All data on this network is given highest priority and performance is in the order of 0.5 to 2ms to communicate with all the devices on this network.

Rack Mounted EtherNet/IP Cards

These CJ1W-EIP21 cards are high performance EtherNet/IP cards with and are best used to connect to other EtherNet/IP devices. They offer fast connection (0.5ms), up to 256 connections each, and up to 1444 byte packets. You can add up to 3 of these cards. You cannot program the NJ through these ports. Please note that even though the EtherNet/IP card can log at 0.5ms it cannot talk to the NJ CPU that fast.

- Ideal for connect to OS32C-DM Safety Scanners
- Best for NJ to NJ or NJ to CJ connections
- Will connect to third party EtherNet/IP devices
- Can be used for HMI connections
- Supports older Omron FINS communications for use with older screens and SCADA.

EtherCAT and EtherNet/IP Cable Colours

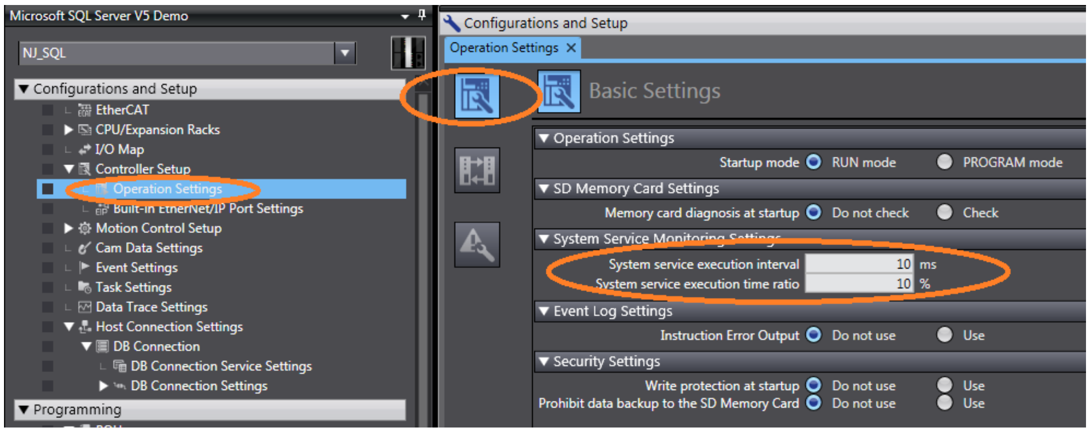
EtherNet/IP / EtherCAT cable colours.

- Blue for EtherCAT (because the external cables only come in blue). The Omron XS5W cables are rated for outside the control panel and are well made. Third party Ethernet cables have been problematic to this point in time on our EtherCAT devices. The cable family for inside the panels is the XS6W part numbers.
- Yellow for EtherNet/IP (because yellow is easy to get made locally). Locally made cables have been OK for EtherNet/IP outside the panel. Inside the panel we recommend the XS5W or XS6W cables.

Assigning More CPU Time to Communications

There are times when a lot of the work the CPU is trying to do is communications. Omron allows you to assign a greater percentage of the CPU time to communications (rather than the ladder and motion logic). This is done by changing the System Service Monitoring Settings.
A couple of things to note:

1) Communications are not serviced every scan (default is 10ms)
2) Communications are set as a percentage of the time used for ladder and motion.
3) This setting increases the time spent on EtherNet/IP ports and local IO and communications cards.
4) As you increase the percentage or decrease the interval you will affect the tasks workload. Use the task workload monitoring to make sure you are not overloading the tasks.



Operation Settings
Startup mode [✓ RUN mode] PROGRAM mode
SD Memory Card Settings
Memory card diagnosis at startup [✓ Do not check] Check
System Service Monitoring Settings
System service execution interval 10 ms
System service execution time ratio 10 %
Event Log Settings
Instruction Error Output [✓ Do not use] Use
Security Settings
Write protection at startup [✓ Do not use] Use
Prohibit data backup to the SD Memory Card [✓ Do not use] Use

I/O Configuration

Selecting I/O Cards and Power for them

The NJ primarily supports many I/O systems but the two main systems are: The local CJ type I/O and the remote NX style I/O on EtherCAT.

The local CJ type I/O includes communication cards, digital I/O and analog I/O cards.

The communications cards make the NJ very versatile and include EtherNet/IP (which is also supports regular Ethernet with FIN's protocol), PROFINET Master, DeviceNet Master, RS232, RS485 and RS422 dual port cards, and CANOPEN cards.

The local I/O cards use the older CJ type bus and the I/O type cards offer no advantage over the NX I/O in price, performance, or wiring density. There is no real reason to use them other than the fact that you might already own them or you prefer to wire to terminals instead of insertion blocks. The NX I/O are synchronous to scan while the local I/O can take several scans to reach the ladder processor.

The NX style I/O connects to an EtherCAT coupler which in turn is connected by an Ethernet type cable (very well shielded) to the EtherCAT port on the NJ. These couplers are typically daisy chained together on the EtherCAT network since each coupler has an in and out communications port. A "Junction Slave" (looks like an Ethernet switch but works in a very different way – the two are not interchangeable) can be used to create an octopus type connection which comes in useful if you wish to turn off or remove components from the network without disconnecting downward I/O.

The NX I/O is very price competitive and performance oriented. NX I/O is designed to update in a little as 250µs. Each NX I/O group typically has a coupler at the left end to connect to EtherCAT and then up to 63 I/O cards can be connected to each coupler. This makes for a potentially very large I/O system.

There are two power buses on this I/O system. One is for the I/O logic. After about ten I/O cards this bus reaches its limit and a bus power feed module is required to break the bus and supply power to all modules to the right of the power feed module.

The NX I/O also has I/O power on its backplane. Combined with I/O power modules many 0V, 24V, and shield connections can be brought to the front of the units eliminating the need for terminals and greatly increasing the neatness and density of the wiring.

How to Configure the I/O Bus Power

- If there are less than 8 I/O per coupler then you could assemble the I/O – configure the I/O using Sysmac Studio – connect to the NJ – “Compare and Merge with Actual Network Configuration” – to get the right version numbers of the I/O. (Why 8 I/O units per coupler – you will likely not need bus power units with 8 I/O or less.
- If there are more than 8 I/O units per coupler – configure the I/O using Sysmac Studio – Sysmac Studio will show you when to add bus power couplers to the system by showing a yellow triangle under the I/O cards where there is not enough power on the backplane. Please note that you must pay attention to the version numbers of the I/O. After configuring the I/O and attaching the I/O to the NJ then - “Compare and Merge with Actual Network Configuration” – to confirm the right version numbers of the I/O. Now you can name the I/O and configure the servo drives. Be sure to “Rebuild Controller” and download the program to the NJ after you “Compare and Merge with Actual Network”. Otherwise you will get an error because the I/O Tag names have not been updated in the NJ.

I/O Power (New and Important)

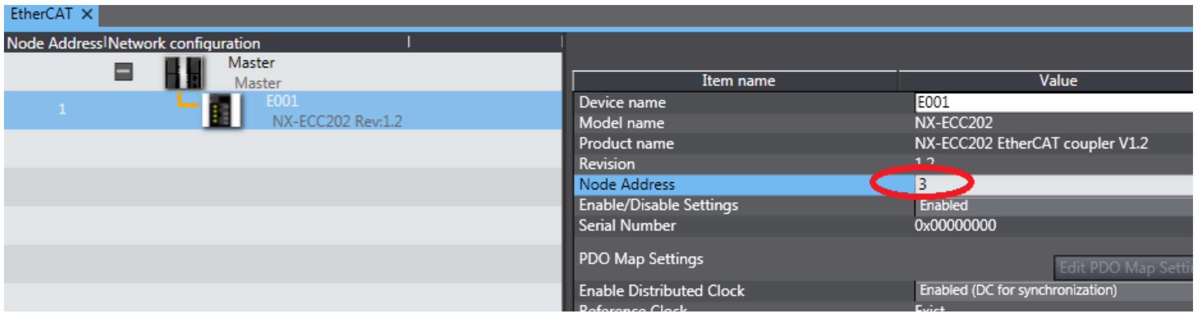
- The I/O has two basic inputs for power (Unit Power and I/O Power). By using separate power supplies for these two inputs you increase the noise resistance of the system. I/O can create noise that will reduce the ruggedness of the bus. (Some customers prefer to wire the inputs with the device power and keep the outputs on their own power supply. The disadvantage to inputs being on the same power as the bus is that a short on an input can cause the bus to go down or become intermittent making the job of finding the short much more difficult.
- I/O power also travels on the backplane. I/O current can only be determined when you know what the I/O is. Each IP Power Supply unit can feed 4 amps to the backplane (some can deliver 10 amps). To determine how many of these units you will need to know the loads being connected to the I/O.
- I/O power modules are also a great way to separate input, output, and analog power supplies. Do not forget the shield unit which gives you a place to connect the shield wiring – and you can put one beside each analog unit greatly improving the look of the wiring and reduce the time it takes to put extensions on the shield wiring to get it to a ground terminal.

There are Two Ways to Address the EtherCAT I/O Units

- One is to use the rotary switches on the I/O to create I/O addresses. Note that 0 (default) is not valid. Addresses 1 to 64 should be reserved for drives and servo units. These can be reassigned to I/O if drives and servos do not use them all. Addresses 65 to 99 should be used for I/O (drives and servo units cannot use these addresses).
- The other way is to use Sysmac Studio to assign addresses. Set the I/O Unit's rotary switch to 00. With this system the drives are still primarily 1 to 64 but the remote I/O can go to 192. (Both methods require that the units be wired (communications) in the correct order which is the same order as seen in the Sysmac Studio configuration screen.
- Drives can only be node numbers 1 to 99. (Section 5-1-1 of I576) Only node numbers 1 to 64 can be mapped to an axis.

When Adding a Junction Unit (Splitter)

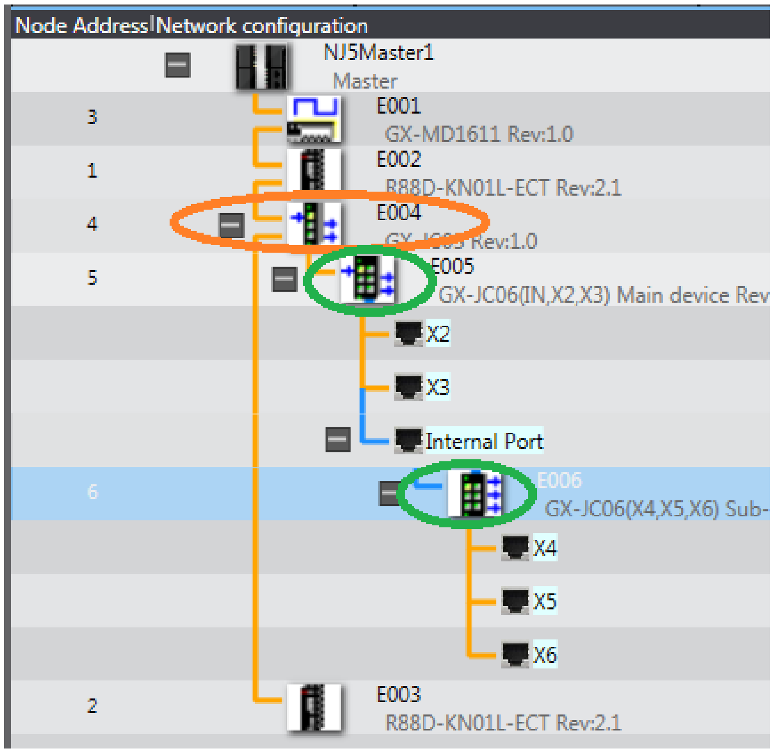
You need to assign an IP address manually. There are no node number switches on the Junction Unit and Sysmac Studio will not assign one automatically.



EtherCAT

Node Address	Network configuration
	Master Master
1	E001 NX-ECC202 Rev:1.2

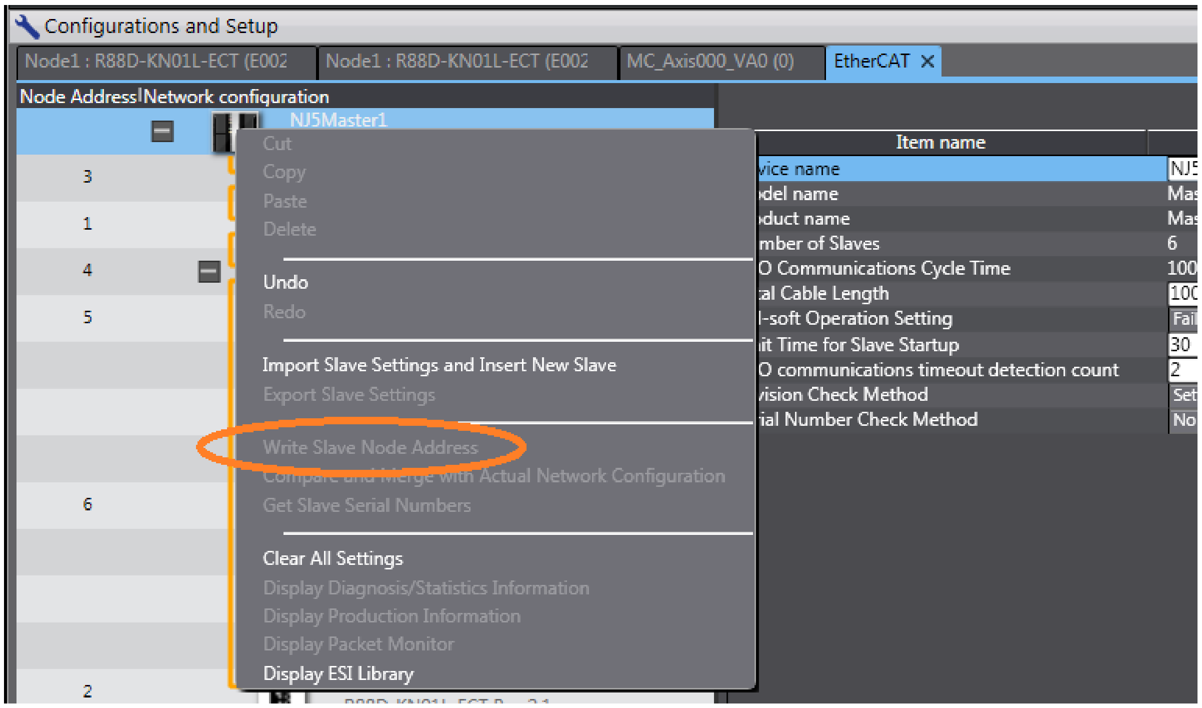
Item name	Value
Device name	E001
Model name	NX-ECC202
Product name	NX-ECC202 EtherCAT coupler V1.2
Revision	1.2
Node Address	3
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	Edit PDO Map Setti
Enable Distributed Clock	Enabled (DC for synchronization)



Node Address	Network configuration
	NJ5Master1 Master
3	E001 GX-MD1611 Rev:1.0
1	E002 R88D-KN01L-ECT Rev:2.1
4	E004 GX-JC03 Rev:1.0
5	E005 GX-JC06(IN,X2,X3) Main device Rev
	X2
	X3
	Internal Port
6	E006 GX-JC06(X4,X5,X6) Sub-
	X4
	X5
	X6
2	E003 R88D-KN01L-ECT Rev:2.1

Please note that the 6-port Junction Slave uses two node addresses as seen in green at left. The 3-port Junction Slave is circled in orange and only has one node address.

Once you have set the addresses you should click on the NJ then right click and select "Write Slave Node address.


Configurations and Setup
Node1 : R88D-KN01L-ECT (E002
Node1 : R88D-KN01L-ECT (E002
MC_Axis000_VA0 (0)
EtherCAT
Node Address Network configuration
NJ5Master1
3
1
4
5
6
2
Cut
Copy
Paste
Delete
Undo
Redo
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library

Item name	
vice name	NJ5
del name	Mas
duct name	Mas
mber of Slaves	6
O Communications Cycle Time	100
al Cable Length	100
l-soft Operation Setting	Fail
it Time for Slave Startup	30
O communications timeout detection count	2
vision Check Method	Set
ial Number Check Method	No

Physical Order and Location of I/O Considerations

- On the EtherCAT network order typically does not matter but within the coupler - by grouping inputs – outputs – and analog I/O – you can give each of them their own power supply. This reduces noise into the analog circuits from the digital circuits and stops outputs from causing spikes and tripping inputs. (NX-PF0630 or NX-PF0730) Pay attention to the maximum current on these modules PF630 is 4A and the PF730 is 10A.
- Using the power supply connection I/O units (NX-PC0010, NX-PC0020 and NX-PC0030) right beside the input or output card allows the 3 wire sensor cable sheath to be left on the wire right up to the input module.
- Using the Shielding Power Supply Connection (NX-TBX01) allows the shielding wire to follow the analog signal wires right up to the input module. Much neater and better shielding.
- The Safety CPU and I/O can be placed anywhere on the NX network. But do not put the safety CPU on a node you may wish to disable and try not to put safety I/O on nodes that you want to disable now and then. The safety program can detect disabled nodes and not run that part of the program – but it makes the program harder to write.

EtherCAT I/O and Task Synchronization

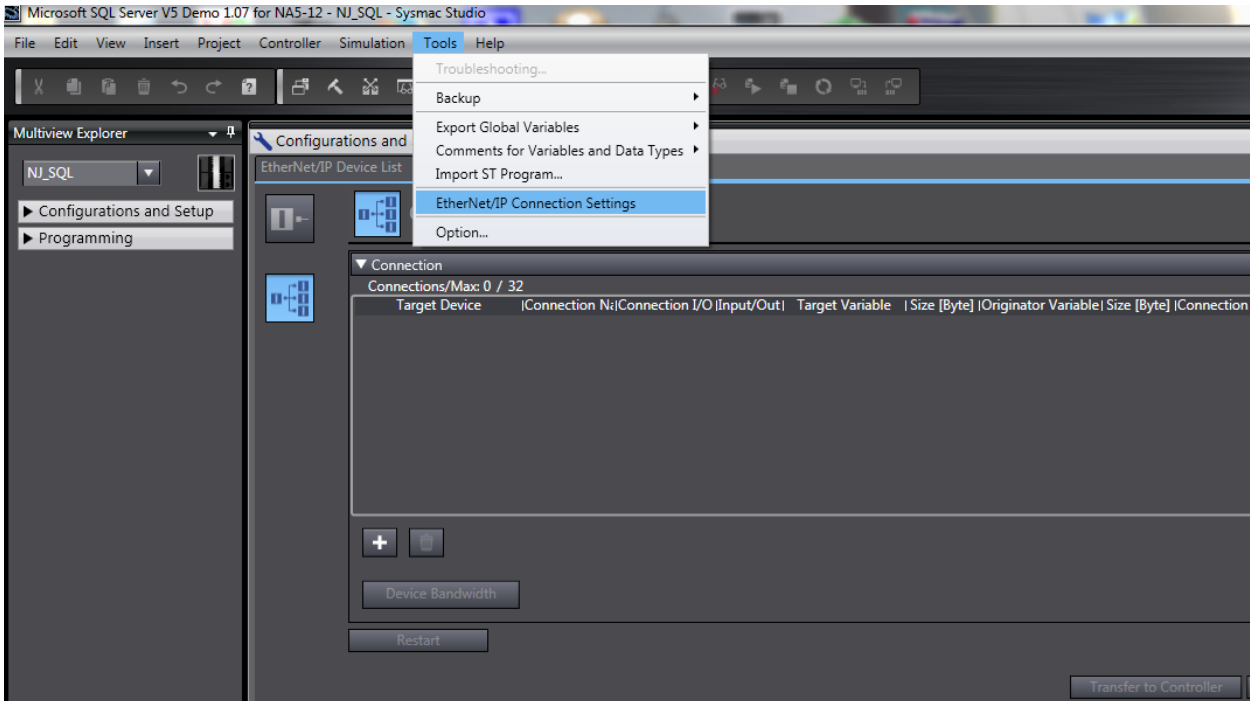
Omron EtherCAT I/O is synchronous with the main task by default. This is true even if the main task is set to 0.25µs (0.5µs in the slower CPU's). You can attach an I/O block to one of the other tasks which means it would be synchronous with that task. The I/O block/servo can only be used in the task it was assigned to be synchronous with.

EtherNet/IP Port for EtherNet/IP Remote I/O

The EtherNet/IP port supports socket services (no protocol UDP or TCP/IP communications) which is very useful for third party Ethernet products.
The same port supports EtherNet/IP communications. Omron uses EDS files for I/O.
The polling rate for each I/O is set independently down to 1ms. The EtherNet/IP I/O is synchronous with the main task but may not refresh every execution of the main task. (The I/O will not change state during the main tasks execution.)

Configuring EtherNet/IP IO

Starting with version 1.10 Sysmac Studio can configure EtherNet/IP connections.


Microsoft SQL Server V5 Demo 1.07 for NA5-12 - NJ_SQL - Sysmac Studio
File Edit View Insert Project Controller Simulation Tools Help
Troubleshooting...
Backup
Export Global Variables
Comments for Variables and Data Types
Import ST Program...
EtherNet/IP Connection Settings
Option...
Multiview Explorer
NJ_SQL
Configurations and Setup
Programming
Configurations and
EtherNet/IP Device List
Connection
Connections/Max: 0 / 32
Target Device | Connection N | Connection I/O | Input/Out | Target Variable | Size [Byte] | Originator Variable | Size [Byte] | Connection
Device Bandwidth
Restart
Transfer to Controller

Other Remote I/O for the NJ

Most of the CJ communications cards work on the NJ and make the NJ very versatile and include EtherNet/IP (which also supports regular Ethernet with FIN's protocol), PROFINET Master, DeviceNet Master, RS232, RS485 and RS422 dual port cards, and CANOPEN cards.

The NJ Ethernet port also supports socket services which allow the user to build his own protocol and talk to third party devices which do not support EIP or Omron protocols. The UDP and TCP/IP sockets are very useful for Ethernet communications to bar code readers, vision systems, and printers.

I/O Labeling Conventions

For All Users, but Especially Omron CJ Users

The NJ does not use rack, slot, and I/O numbering systems so the following is recommended for the tag names.

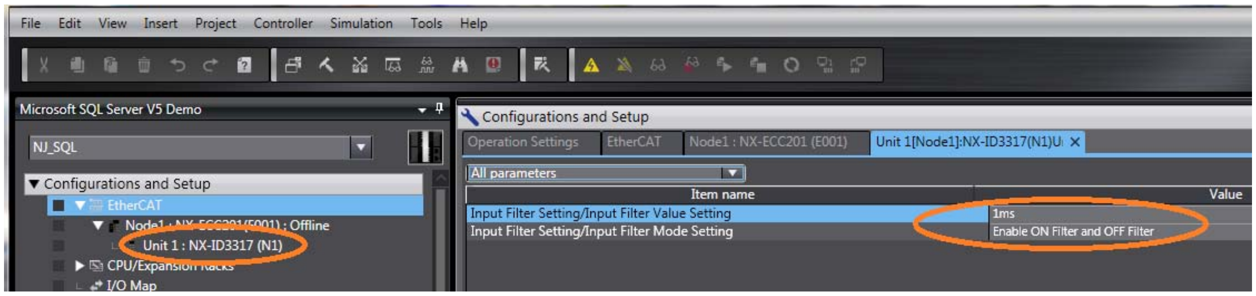
1.	For EtherCAT I/O:	ect_<Ndxxx..f.yy.zz> (xxx=node, f=I for in and Q for out, yy=slot, zz=I/O) this is not how Sysmac Studio auto names the I/O. When you add IO to EtherCAT it will automatically show up in the I/O map. This includes servo, drive, NX I/O, vision, and measurement sensors.
2.	For EtherNet/IP I/O:	eip_<Ndxxx..f.yy.zz> (xxx=node, f=I for in and Q for out, yy=slot, zz=I/O)
3.	For the HMI:	hmi_ (please note that the “create device variable” automatic naming tool in Sysmac Studio adds the “name” of the NJ processor to the front of the variable names from the NJ in the NA variable mapping.)
4.	For DeviceNet:	dn_<Ndxxx..f.yy.zz> (xxx=node, f=I for in and Q for out, yy=slot, zz=I/O)
5.	For PROFINET:	pn_<Ndxxx..f.yy.zz> (xxx=node, f=I for in and Q for out, yy=slot, zz=I/O)
6.	For PROFIBUS:	pb_<Ndxxx..f.yy.zz> (xxx=node, f=I for in and Q for out, yy=slot, zz=I/O)

Especially for Rockwell Programmers

Some programmers add code at the beginning of the Rockwell program to capture the state of the I/O at the beginning of the program execution and keeping the state of the I/O the same for the entire execution of the program. This is done because the I/O and the program are not synchronous.

- Omron allows you to connect the I/O to a task. Programs are put in the tasks. In this way the I/O status will not change during the execution of a program. The task and the I/O refresh will become synchronous. The above Rockwell code is not needed.
- Typically all I/O is attached to the main task and therefore by default is synchronous with the main task and all programs in it.

For more information - see “Programs and Tasks” – “I/O Task Relationships” later in this manual.
Some programmers add timers at the beginning of the program to capture the state of the I/O and put filters on the I/O. Similar to the point above, this adds the filter option at the same time.



Item name	Value
Input Filter Setting/Input Filter Value Setting	1ms
Input Filter Setting/Input Filter Mode Setting	Enable ON Filter and OFF Filter

- Assign the I/O block to the task that contains the program that will use the I/O to fix the first problem.
- Each I/O has a filter setup in the I/O setup area. When looking at the I/O go into the “Edit Unit Operation Settings” area. Typical settings are from no filter, 0.25µs, 0.5µs … up to 256ms. This removes the need for the timers.
- Note that you can have just an Off filter. You can use this to catch a pulse and hold it On until the NJ has time to scan the I/O.

The best way to store permanent data like Recipes or setup parameters on an NJ processor is to put them all in one variable or array and store them to the SD card using a program in the NJ.

This method offers a couple of advantages over other methods.

1. The data is stored on the SD card and will not be lost even if the CPU dies.
2. You can store several versions of the data which allows you to go back and retrieve previous versions if the current version seems incorrect. (To do this you should make part of the file name a date stamp.)
3. The program can also be stored on the same SD card. By putting the SD card in a new processor the program, setup, and backed up data can be restored without using Sysmac Studio.
4. You can remove the SD card to copy the files to your laptop.
5. You can transfer the files out of the SD card using FTP from a remote laptop.
6. Using an event input you can make the NJ transfer the file automatically via FTP to a server so you have a record of all the changes. You may want to add operator name to the file.
7. You can read the data back into the variable area using code and a button on the operator screen.

Especially for Siemens Programmers

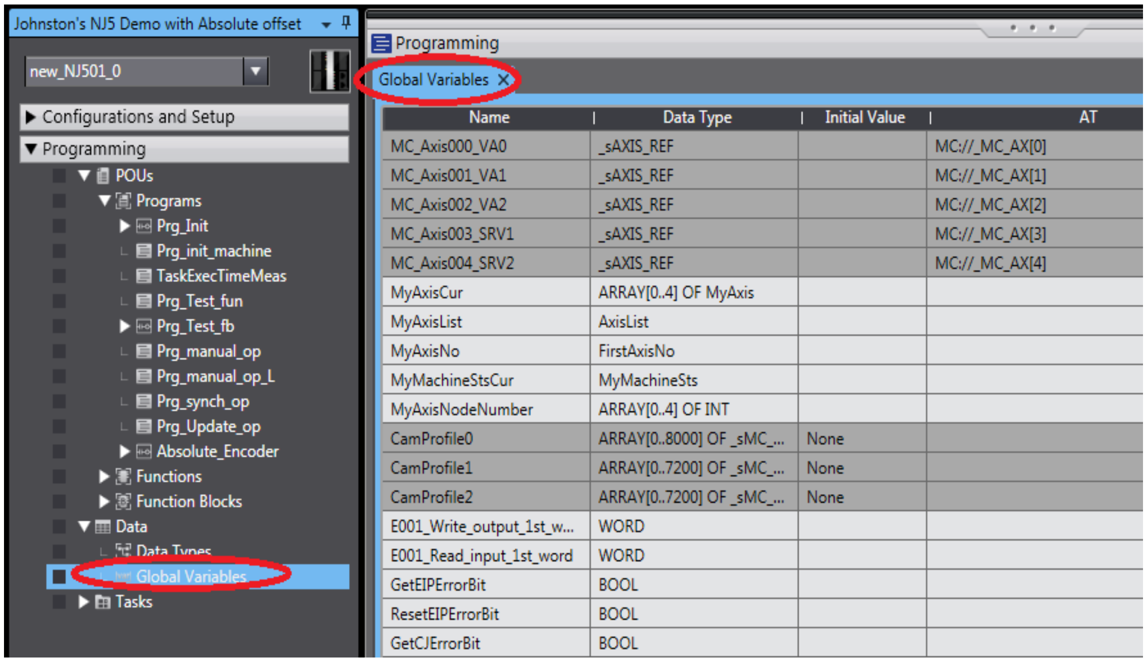
There is a difference in specifying structured variables.
In the Siemens Portal V13 software, when you are trying to use a “TUSEND” instruction (or one like it) your structured variable must be created as a “data block” under the “program blocks”. This happens because some of the data types required by the “TUSEND” type instruction are only found in the “data block” – they are not found in the “PLC Tags” area.
In Omron there are a couple of differences when using this type of instruction.

1. All instructions are in the Sysmac Studio editor – they do not have to be brought into the project.
2. All data types for all instructions are available at all times.
3. Sysmac Studio does not have data blocks. You can create a structure and then create a variable (global or local) that uses that structure.
4. In Sysmac Studio a structure can be made up of internal variables and other structures.

Creating Global Variables

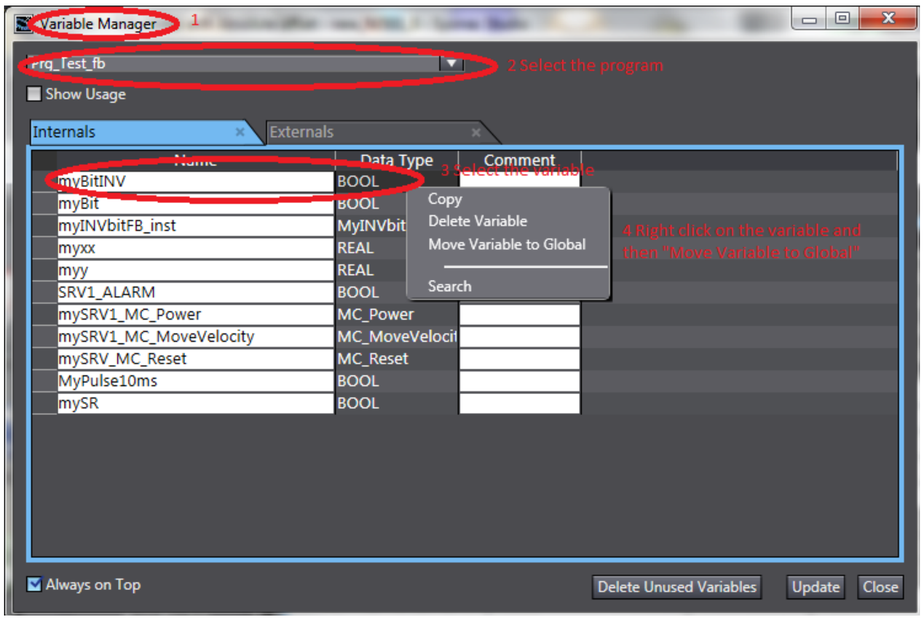
The best way to create global variables is to create them in the global variable area first. When editing online, create a global variable by adding it to the global variable area before using it in the program.

- Motion variables (tags) are created automatically as shown on the first 5 lines below.
- Cam variables are created automatically as shown on lines 11 to 13 below.
- Local I/O and EtherCAT I/O are created in the I/O table – go there to give them names.
- The variables (tags) that you create here are for internal global use, for variables used to connect the NA and NS HMIs, and for EtherNet/IP variables.



Name	Data Type	Initial Value	AT
MC_Axis000_VA0	_sAXIS_REF		MC://_MC_AX[0]
MC_Axis001_VA1	_sAXIS_REF		MC://_MC_AX[1]
MC_Axis002_VA2	_sAXIS_REF		MC://_MC_AX[2]
MC_Axis003_SRV1	_sAXIS_REF		MC://_MC_AX[3]
MC_Axis004_SRV2	_sAXIS_REF		MC://_MC_AX[4]
MyAxisCur	ARRAY[0..4] OF MyAxis		
MyAxisList	AxisList		
MyAxisNo	FirstAxisNo		
MyMachineStsCur	MyMachineSts		
MyAxisNodeNumber	ARRAY[0..4] OF INT		
CamProfile0	ARRAY[0..8000] OF _sMC_...	None	
CamProfile1	ARRAY[0..7200] OF _sMC_...	None	
CamProfile2	ARRAY[0..7200] OF _sMC_...	None	
E001_Write_output_1st_w...	WORD		
E001_Read_input_1st_word	WORD		
GetEIPErrorBit	BOOL		
ResetEIPErrorBit	BOOL		
GetCJErrorBit	BOOL		

To Move a Local Variable to Global Variable



Name	Data Type	Comment
myBitINV	BOOL	
myBit	BOOL	
myINVbitFB_inst	MyINVbit	
myxx	REAL	
myy	REAL	
SRV1_ALARM	BOOL	
mySRV1_MC_Power	MC_Power	
mySRV1_MC_MoveVelocity	MC_MoveVelocit	
mySRV_MC_Reset	MC_Reset	
MyPulse10ms	BOOL	
mySR	BOOL	

1. View
2. Variable Manager
3. In the top left corner select the local program
4. Left click on the variable in question
5. Right click and select “Move Variable to Global”

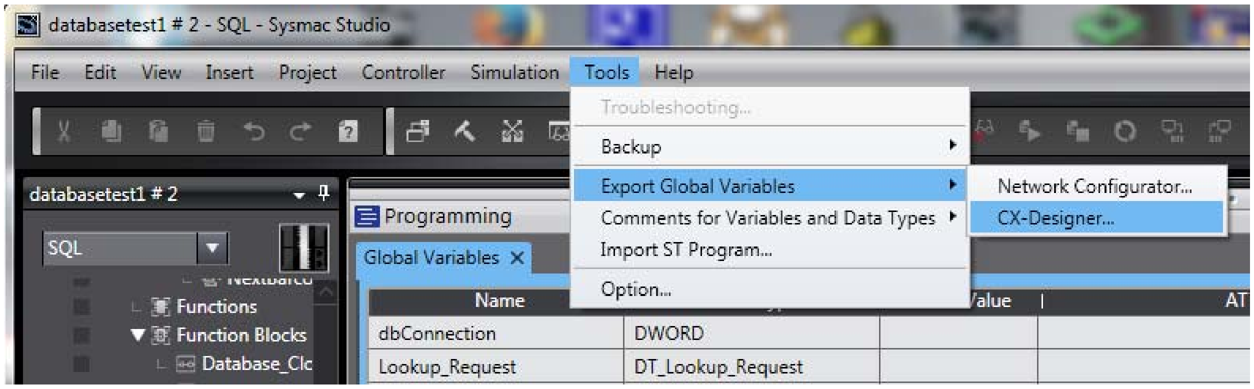
Notes:
You can put the variables in alphabetical order by clicking on the “Name” title.
You can delete unused variables on this screen.

Global Variables for the NA HMI

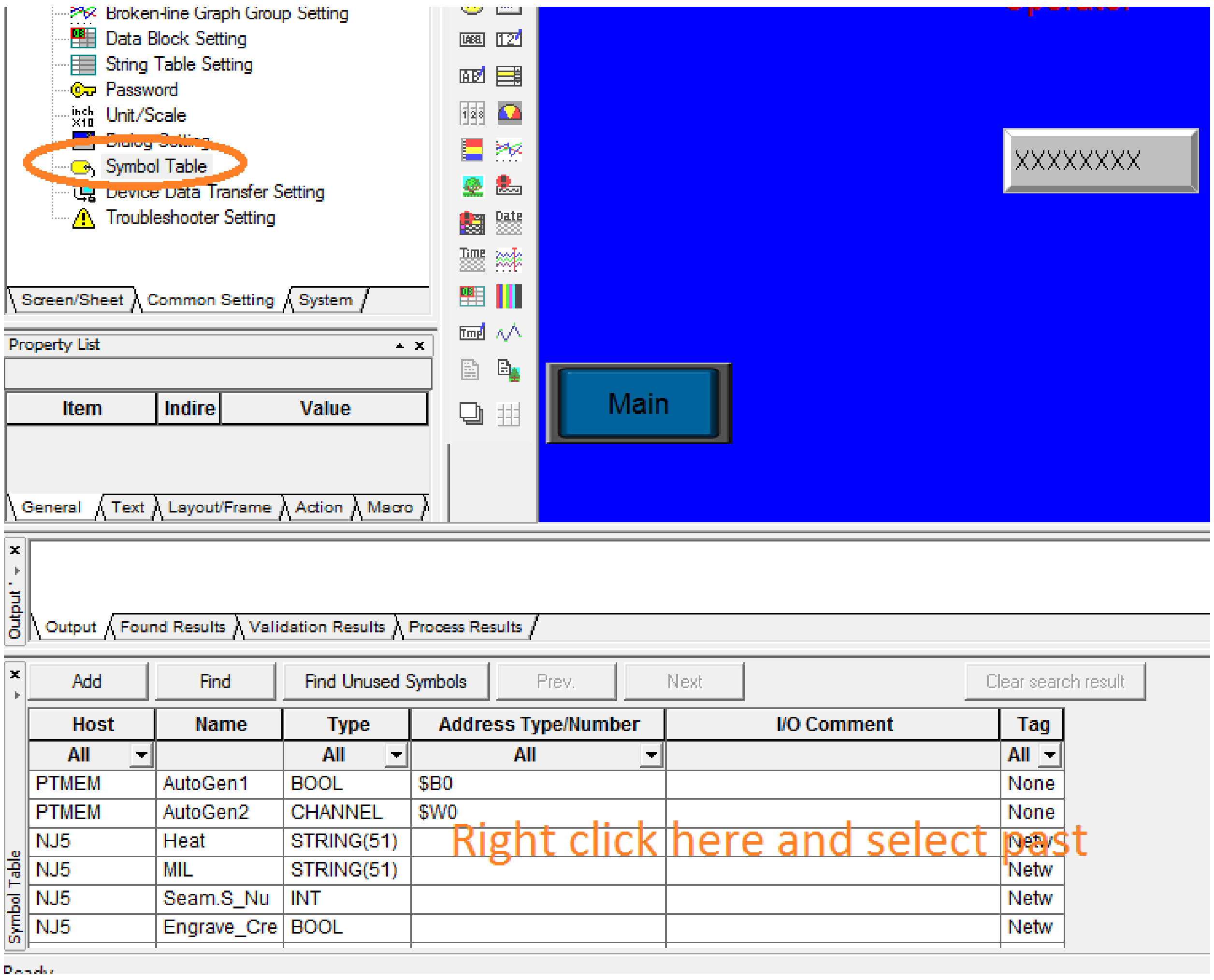
All global variables in the NJ program are visible in the NA variable table. They do not need to be published.

Global Variables for the NS Screen

You set the variables you need to “Publish Only”, “Input”, or “Output”. You then go “tools”, “Export Global Variables”, “CX-Designer”. Publish Only means in and out.



Now you go to CX-Designer and import the global variables



Host	Name	Type	Address Type/Number	I/O Comment	Tag
All		All	All		All
PTMEM	AutoGen1	BOOL	$B0		None
PTMEM	AutoGen2	CHANNEL	$W0		None
NJ5	Heat	STRING(51)			Netw
NJ5	MIL	STRING(51)			Netw
NJ5	Seam.S_Nu	INT			Netw
NJ5	Engrave_Cre	BOOL			Netw

Right click here and select past

Global Variables for EtherNet/IP Data Link

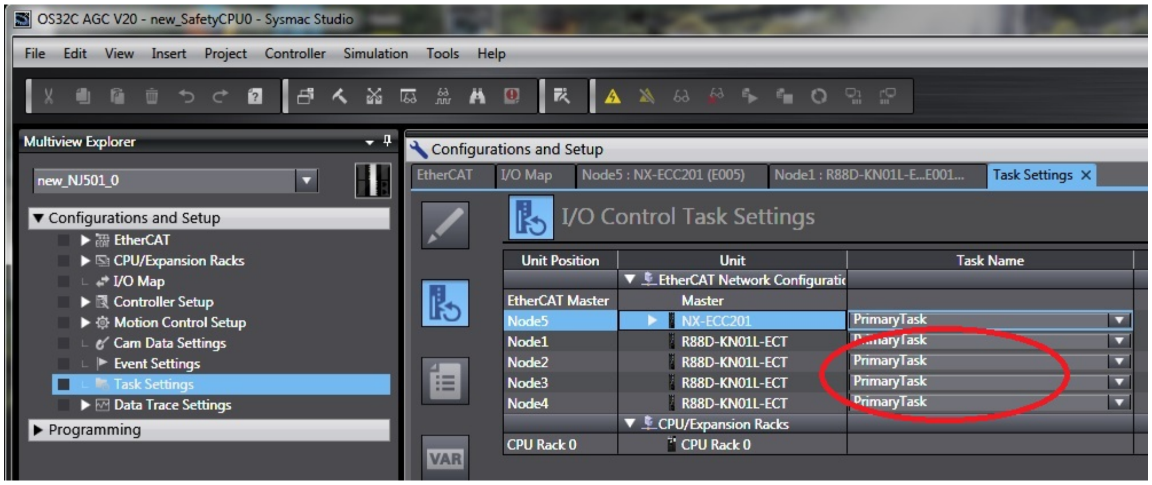
You define a global variable under “Programming – Data – Global Variables”.
If you are using EtherNet/IP data link then you must specify input if it is being used as an input or output. (It cannot be both).

Programming

Programs and Tasks

Primary Task (0.5ms to 4ms – only one primary task)

- All motion commands must be in the primary task.
- All motion drives and servo amplifiers must be in the primary task. (A drive not attached to a motion profile does not have to be in the primary task.) This is set in “Configurations and Setup” – “Task Settings” – “I/O Control Task Settings” – under the EtherCAT master.



I/O Control Task Settings

Unit Position	Unit	Task Name
	EtherCAT Network Configuratic	
EtherCAT Master	Master	
Node5	NX-ECC201	PrimaryTask
Node1	R88D-KN01L-ECT	PrimaryTask
Node2	R88D-KN01L-ECT	PrimaryTask
Node3	R88D-KN01L-ECT	PrimaryTask
Node4	R88D-KN01L-ECT	PrimaryTask
	CPU/Expansion Racks	
CPU Rack 0	CPU Rack 0	

Tasks with Priority 16, 17, and 18 (One task for each priority 1 to 100ms)

These are periodic tasks and useful to reduce the load on the primary task. Useful for slower communications programs, HMI routines, and slower push button type I/O.

Event Tasks with Priority 8 or 48 (32 of them)

At the moment these tasks are triggered by an instruction or when a variable expression is true. There are no I/O triggered even tasks at this time.

I/O Task Relationships

- I/O can only be used in a program assigned to the same task as the I/O.
- You would assign the program and I/O to a slower task in order to keep the main task running fast and because the program you are running does not need to run that fast. An example is a remote push button station. The I/O and task could run as slow as 100ms – nobody would notice. Another example would be the code to control the HMI. A third example would be a lot of math for setup calibration.
- Why assign IO to a slower task? So that it does not change state halfway through the task. If the I/O is refreshed with a faster task – the I/O can change state every time that task runs. Programmers typically do not like an IO changing state partially through a program.

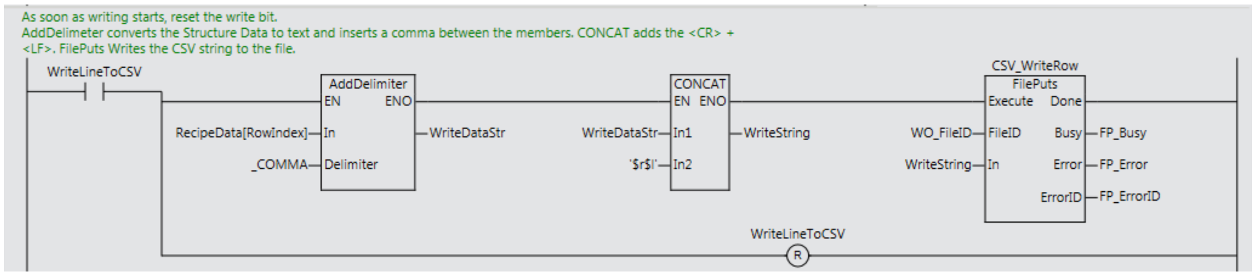
Storing Permanent Data on the NJ

The best way to store permanent data on the NJ is to put it in one variable and then store it on the memory card using the File read and Write commands. Store the current data to a fixed file name. That way you know which file to read to get the data back. By creating another file with the date as part of the filename – you can have a history of setup data which may come in handy if you the data becomes contaminated. Sysmac Studio can look at the files – then you can just rename the file you want so the CPU can read it.
When you upgrade the firmware version of the CPU and sometimes the program – you can lose permanent data on the CPU.
The file memory card (SD card) can then be also be put in a new CPU and the variables restored. The memory card can also have a backup program with the program and NJ and I/O configuration data.
The next chapter shows how to send data to and from the SD card in the NJ CPU.

Reading Data from a File (SD card located in SD slot on NJ CPU)

File Read and Write

FileOpen is used to open the file. FileSeek can be used to point at a given line in the file. FilePut and FileGet are used to write or read data to/from the file. Here is an example of a file write where the first instruction changes the variable of user data type to a string with commas between the variables, the second box adds the carriage return line feed and the third box actually writes the data to a file.

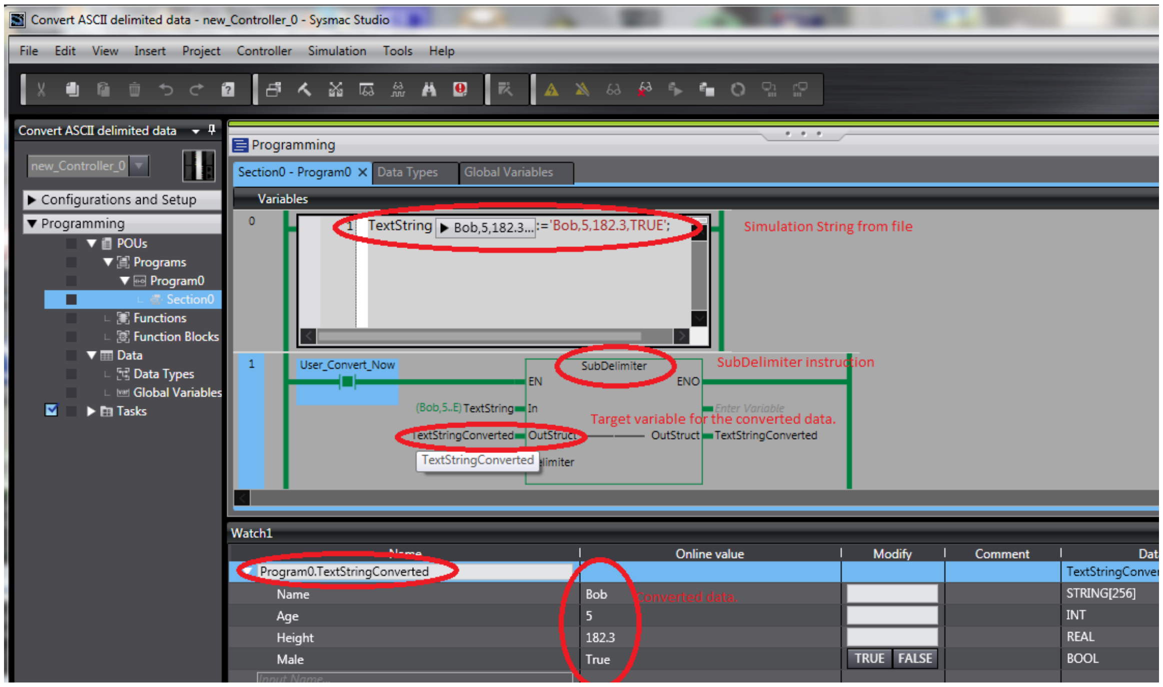


As soon as writing starts, reset the write bit.
AddDelimeter converts the Structure Data to text and inserts a comma between the members. CONCAT adds the <CR> + <LF>. FilePuts Writes the CSV string to the file.

Converting Read and Write Text Strings to Variable Data

The NJ has a very powerful instruction for interpreting data from a .csv file. The “SubDelimiter” function. You can also take a variable of custom data type and convert to text ready to be sent to a file using the AddDelimiter function.

For example:
The first line of the file is Bob,5,182.3,True
With the SubDelimiter instruction you create a variable with a custom data type like so:
Variable: TextStringConverted which would be of type TextStringConvereted_Struct
Data Type: TextStringConvereted_Struct with members
Name of base type “String[256]
  Age of base type “INT”
  Height of base type “REAL”
  Male of base type “BOOL”



```
TextString ▶ Bob,5,182.3... :='Bob,5,182.3,TRUE';
```

Name	Online value	Modify	Comment	Dat
Program0.TextStringConverted				TextStringConve
Name	Bob			STRING[256]
Age	5			INT
Height	182.3			REAL
Male	True	TRUE FALSE		BOOL

Scan Times

How a PID instruction Affects Scan Time

My testing showed 10 loops ran an average of 218µs. Peak was 315µs.

How “Auto Tune” Affects Scan Time

There is no scan time value for instructions on the NJ. But testing at the factory has shown that initiating 90 auto tune loops at the same time added about 3ms to the scan time. This could be a problem if you put them in the main task. It is also important to know that you should not run a PID loop (actually the auto tune PID loops) in a scan that runs slower than 8ms.
My actual testing showed 10 loops auto tuning gave a scan time of 259µs (up from 218µs for just the PID loop) for about two seconds till the PID tuning was complete.

Adding and Removing EtherCAT I/O

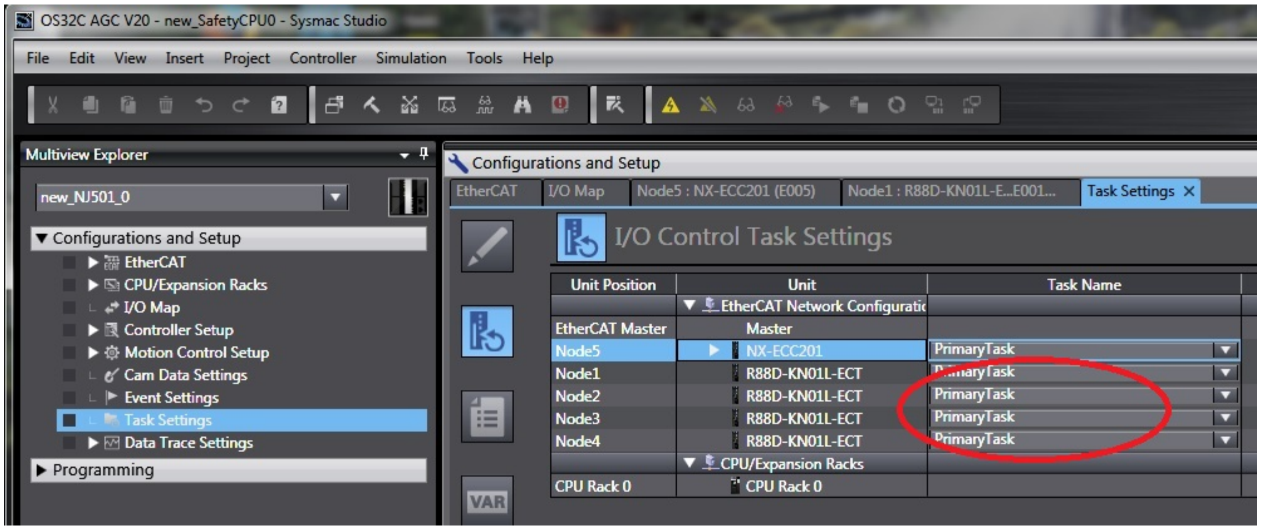
Adding NX I/O

Notes:

- You cannot add an NX I/O bank online if it was not previously in the I/O table.
- You cannot add an NX I/O card online.
- You can add an NX I/O bank – disable it while online or offline – remove the bank - run the machine without it – then put the NX I/O bank back in (online) at a later time.
- You can put a blank in the I/O table and then add the I/O card later. This allows you to insert a known card into the bank at a later date.
- To add an NX I/O card you can add the card. Compare and merge with a scan of the network or add it manually. **You need to compile the program and then download it to the NJ**. Otherwise you will keep getting an I/O card not configured properly type error.

G5 Servos

Addressing



Unit Position	Unit	Task Name
	EtherCAT Network Configuratic	
EtherCAT Master	Master	
Node5	NX-ECC201	PrimaryTask
Node1	R88D-KN01L-ECT	PrimaryTask
Node2	R88D-KN01L-ECT	PrimaryTask
Node3	R88D-KN01L-ECT	PrimaryTask
Node4	R88D-KN01L-ECT	PrimaryTask
	CPU/Expansion Racks	
CPU Rack 0	CPU Rack 0	

Servos should be nodes 1 to 64 (or the maximum number your NJ can handle). The "Motion Control Setup" attaches the servos to axis so the order of the node numbers does not matter unless you want the node number to match the "Axis" number. The servo will be referred to by an axis number and name in the program.
Servo motors used with the motion instructions must be attached to the main axis in the program for I/O refresh. The servo programs must always be in the primary task.

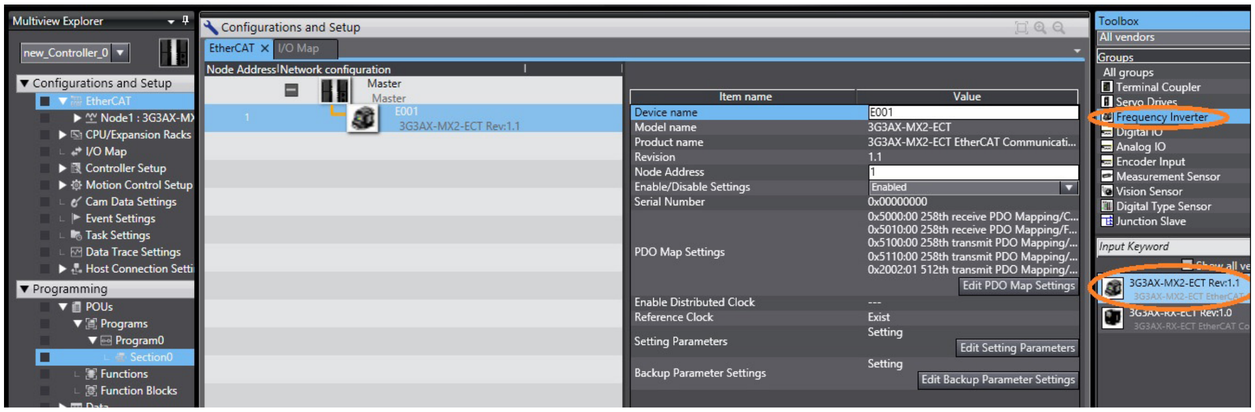
Adding to I/O Table

Just add the servo to the EtherCAT network like any other I/O. G5 servos can only be added to the EtherCAT network. Servos to be used with the motion instructions can only be added to the EtherCAT network.

MX2 Inverter Drive

Add the Drive to I/O Table

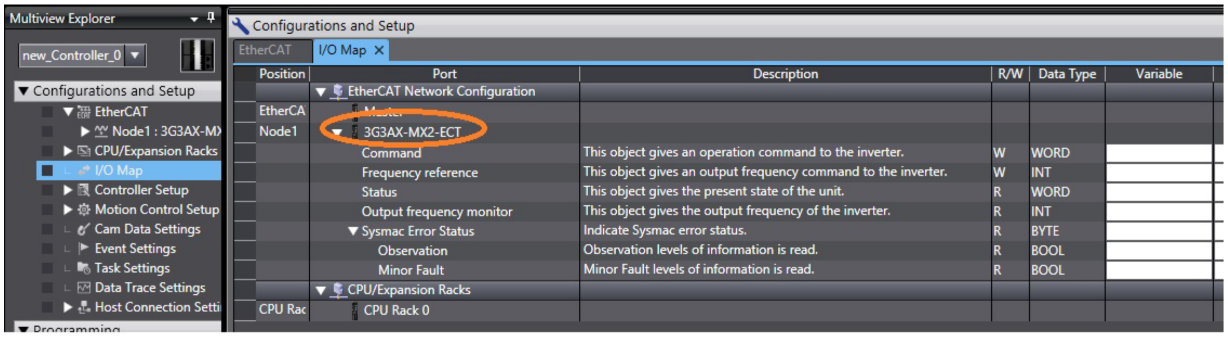
There is a quick start manual P521 for starting an MX2 on EtherCAT.
Manual I574 is the user manual for the MX2 EtherCAT module.
MX2 Drives should be node numbers 00 to 99 (all the rotary switches will accept).
Go to "Configurations and Setup – EtherCAT" and then go to the right side of the page and pick "Frequency Inverter" and then in the lower box pick "3G3AX-MX2-ECT".



Item name	Value
Device name	E001
Model name	3G3AX-MX2-ECT
Product name	3G3AX-MX2-ECT EtherCAT Communicati...
Revision	1.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
PDO Map Settings	0x5000:00 258th receive PDO Mapping/C... 0x5010:00 258th receive PDO Mapping/F... 0x5100:00 258th transmit PDO Mapping/... 0x5110:00 258th transmit PDO Mapping/... 0x2002:01 512th transmit PDO Mapping/... Edit PDO Map Settings
Enable Distributed Clock	---
Reference Clock	Exist
Setting Parameters	Setting Edit Setting Parameters
Backup Parameter Settings	Setting Edit Backup Parameter Settings

Configure the Drive I/O

Then you go to "Configurations and Setup – I/O Map"
You will find the MX2 and see a list of variables available for the MX2. To use these variables in your program you must give a tag name in the "Variable" column.



Position	Port	Description	R/W	Data Type	Variable
	EtherCAT Network Configuration				
EtherCA	Master				
Node1	3G3AX-MX2-ECT				
	Command	This object gives an operation command to the inverter.	W	WORD	
	Frequency reference	This object gives an output frequency command to the inverter.	W	INT	
	Status	This object gives the present state of the unit.	R	WORD	
	Output frequency monitor	This object gives the output frequency of the inverter.	R	INT	
	Sysmac Error Status	Indicate Sysmac error status.	R	BYTE	
	Observation	Observation levels of information is read.	R	BOOL	
	Minor Fault	Minor Fault levels of information is read.	R	BOOL	
	CPU/Expansion Racks				
CPU Rac	CPU Rack 0				

The "W" type command word:
Bit 0: 0=stop, 1=Forward
Bit 1: 0=stop, 1=Reverse
Bit 7: 1=Reset errors on ECAT unit and MX2. (Rising Edge)
The "W" type Frequency Reference is in 0.01Hz units.
The "R" type Status Word
Bit 0: 0 for stop or reverse, 1 for forward
Bit 1: 0 for stop or forward, 1 for reverse
Bit 3: Fault
Bit 7: Warning
Bit 9: 0=Local (Digital Inputs controlled), 1=Remote (EtherCAT controlled)
Bit 12: 0= accel or decel, 1=at set speed

Bit 15: 0=ok, 1=EtherCAT communications lost.
The “R” type Output Frequency Monitor in 0.00Hz units.
Note: If you put the drive in high speed mode the Frequency units will be 000.0Hz.

Configure the Drive

In the drive set:
A001 = 04 option board
A002 = 04 option board
C102 = 03 Trip reset only (you might need to set B037=1 to see this parameter.
P012 can be 00 or 02.
You cannot control motor 2 with the EtherCAT option board.

Testing Offline

Simulation

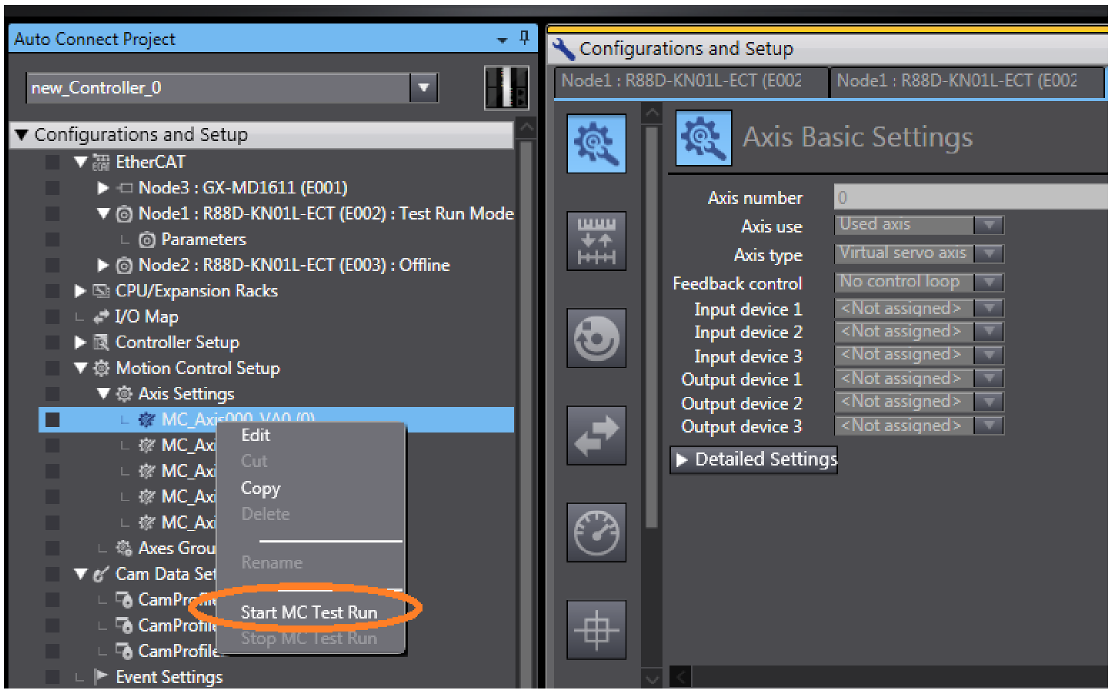
Sysmac Studio can program the NJ and the HMI (NA or NS).
Sysmac Studio can also run all three in simulation mode.
Simulation mode is useful for getting used to the NJ instruction set and to test the HMI abilities and communications to the NJ. The NA is much more capable at moving all the NJ data types back and forth so if you are using an NS the simulation is great for testing the data types.
Sysmac Studio can even make the simulations talk to each other (NJ to NS or NJ to NA). Sysmac Studio directly programs the NJ and NA and the variables are automatically mapped from NJ to NA. With the NS you need to export and import the variables.

It is important to note that the NJ simulation can run in continuous, single scan, or even with breakpoints.

Commissioning

Commissioning G5 Servos

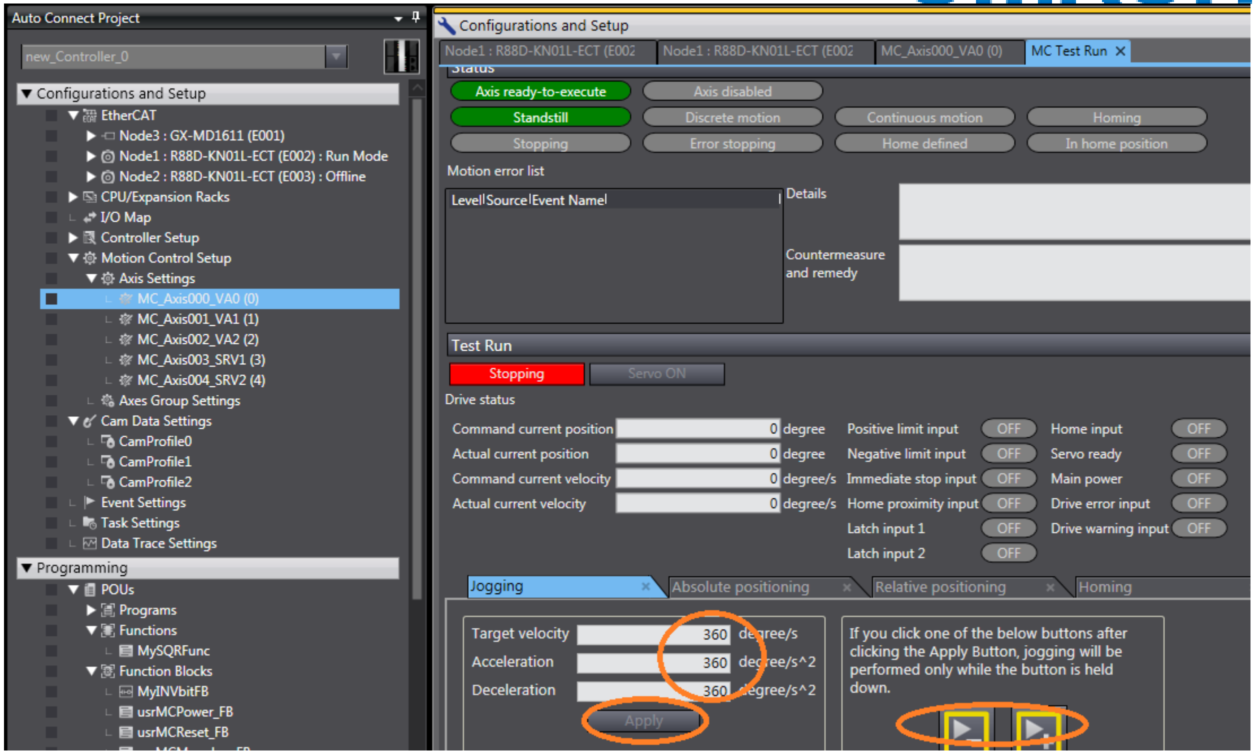
To test the servo this is the screen we want – where we can manually control the motor to test speed, calibration, and direction.



Axis number	0
Axis use	Used axis
Axis type	Virtual servo axis
Feedback control	No control loop
Input device 1	<Not assigned>
Input device 2	<Not assigned>
Input device 3	<Not assigned>
Output device 1	<Not assigned>
Output device 2	<Not assigned>
Output device 3	<Not assigned>

You should get a screen like this one that shows you the status of your end limits and origin sensors etc. At the bottom of the screen are some numbers you need to enter. The Target Velocity, Acceleration, and Deceleration need to be set. Set a value that will turn the motor at about 1 revolution per second. That way the jog buttons just to the right can be used to test the motor to see if it turns, does it turn the right direction, etc.

- For testing you may find that turning the torque down on the motor (setup) to about 10 or 20% will go a long way to protecting the mechanical hardware till you get the direction, speed, end sensors, and scaling correct.
- Take the time to make sure the end limits are at the correct end. You can always trip them with a screw driver and make sure the drive stops moving in that direction.

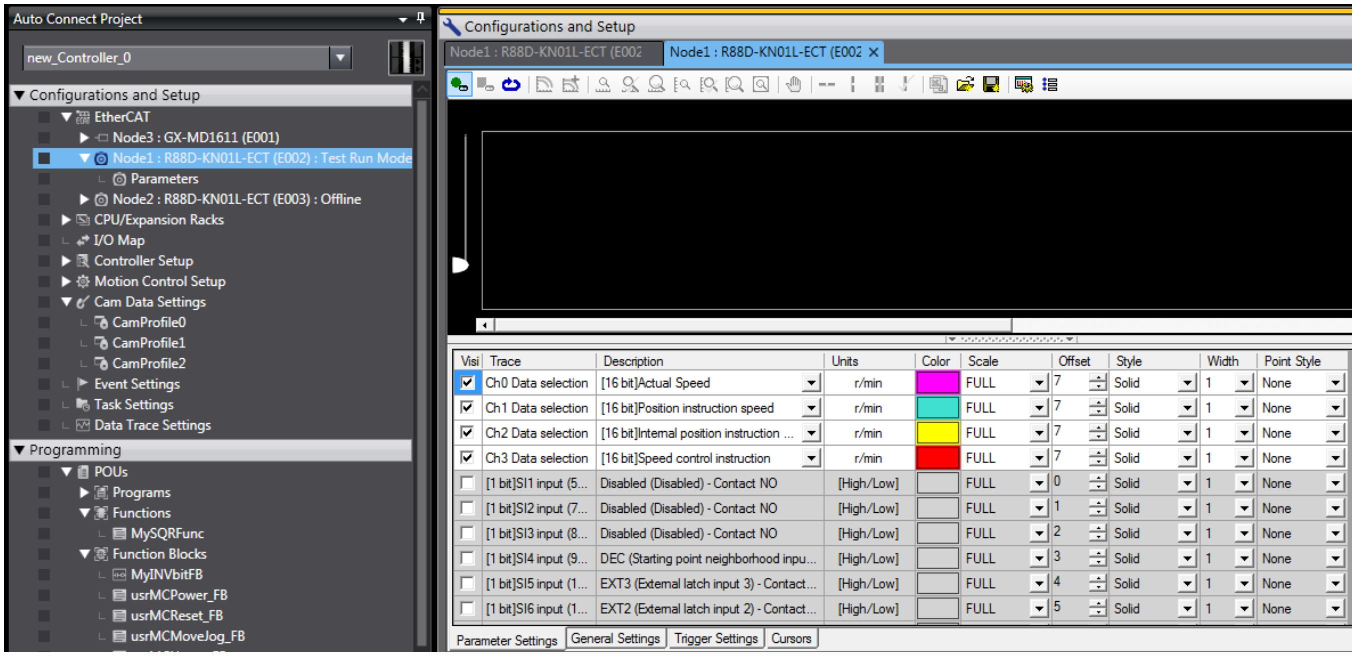


Command current position	0 degree	Positive limit input	OFF	Home input	OFF
Actual current position	0 degree	Negative limit input	OFF	Servo ready	OFF
Command current velocity	0 degree/s	Immediate stop input	OFF	Main power	OFF
Actual current velocity	0 degree/s	Home proximity input	OFF	Drive error input	OFF
		Latch input 1	OFF	Drive warning input	OFF
		Latch input 2	OFF		

Target velocity	360	degree/s
Acceleration	360	degree/s^2
Deceleration	360	degree/s^2

If you click one of the below buttons after clicking the Apply Button, jogging will be performed only while the button is held down.

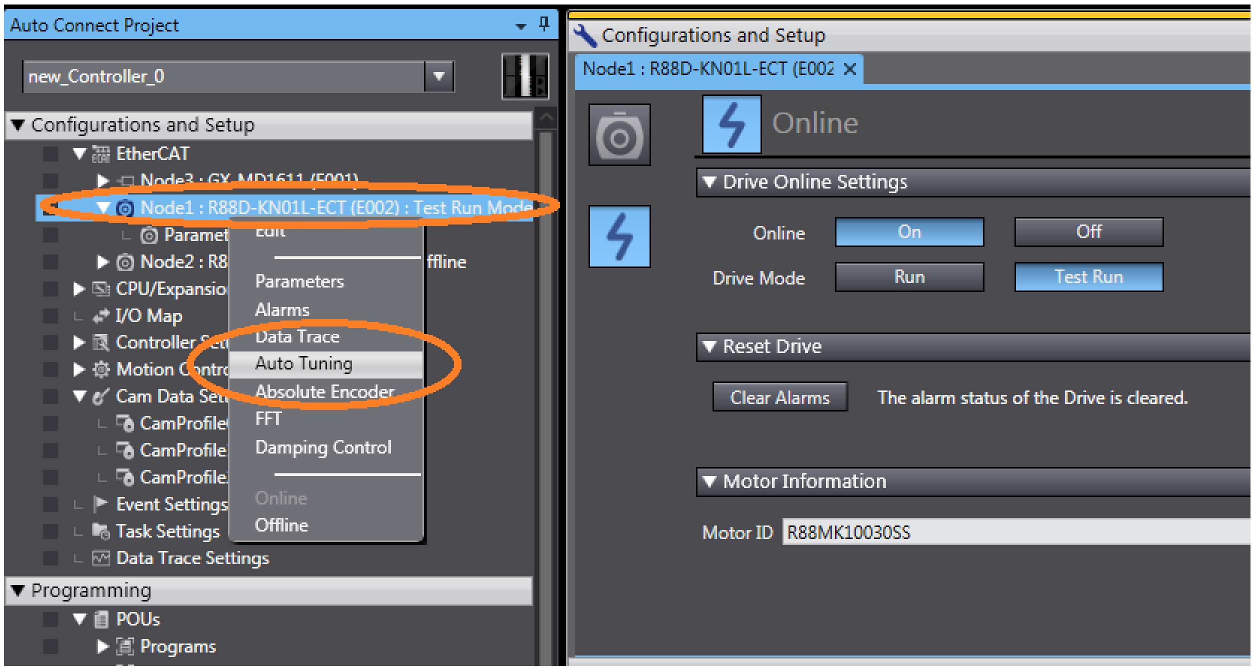
This is where you can set the G5 Drive Parameters



Visi	Trace	Description	Units	Color	Scale	Offset	Style	Width	Point Style
✓	Ch0 Data selection	[16 bit]Actual Speed	r/min	🟪	FULL	7	Solid	1	None
✓	Ch1 Data selection	[16 bit]Position instruction speed	r/min	🟦	FULL	7	Solid	1	None
✓	Ch2 Data selection	[16 bit]Internal position instruction ...	r/min	🟨	FULL	7	Solid	1	None
✓	Ch3 Data selection	[16 bit]Speed control instruction	r/min	🟥	FULL	7	Solid	1	None
	[1 bit]SI1 input (5...	Disabled (Disabled) - Contact NO	[High/Low]		FULL	0	Solid	1	None
	[1 bit]SI2 input (7...	Disabled (Disabled) - Contact NO	[High/Low]		FULL	1	Solid	1	None
	[1 bit]SI3 input (8...	Disabled (Disabled) - Contact NO	[High/Low]		FULL	2	Solid	1	None
	[1 bit]SI4 input (9...	DEC (Starting point neighborhood inpu...	[High/Low]		FULL	3	Solid	1	None
	[1 bit]SI5 input (1...	EXT3 (External latch input 3) - Contact...	[High/Low]		FULL	4	Solid	1	None
	[1 bit]SI6 input (1...	EXT2 (External latch input 2) - Contact...	[High/Low]		FULL	5	Solid	1	None

Tuning

The best way to tune a servo motor is to create a small program to move the motor back and forth about one turn.
Then go to the “Data Trace Settings” and add a data trace. Add the MC_Axis001_VA1.Act.Pos, MC_Axis001_VA1.Act.Trq, MC_Axis001_VA1.Cmd.Pos variables. Turn the trace on and now you can see how the servo is performing.



The alarm status of the Drive is cleared.

Motor ID R88MK10030SS

Auto Tune

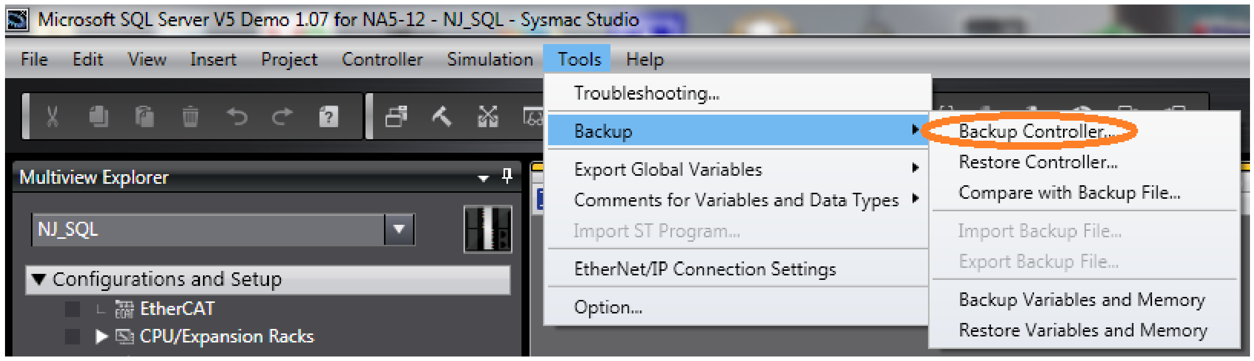
This is how you can auto tune. Auto tune works best on screw type drive systems. Low backslap and stiff.
You can watch the auto tune this way – same as above but select “Data Trace” instead of “Auto Tuning”. You could also just use the “Data Trace” you setup above.

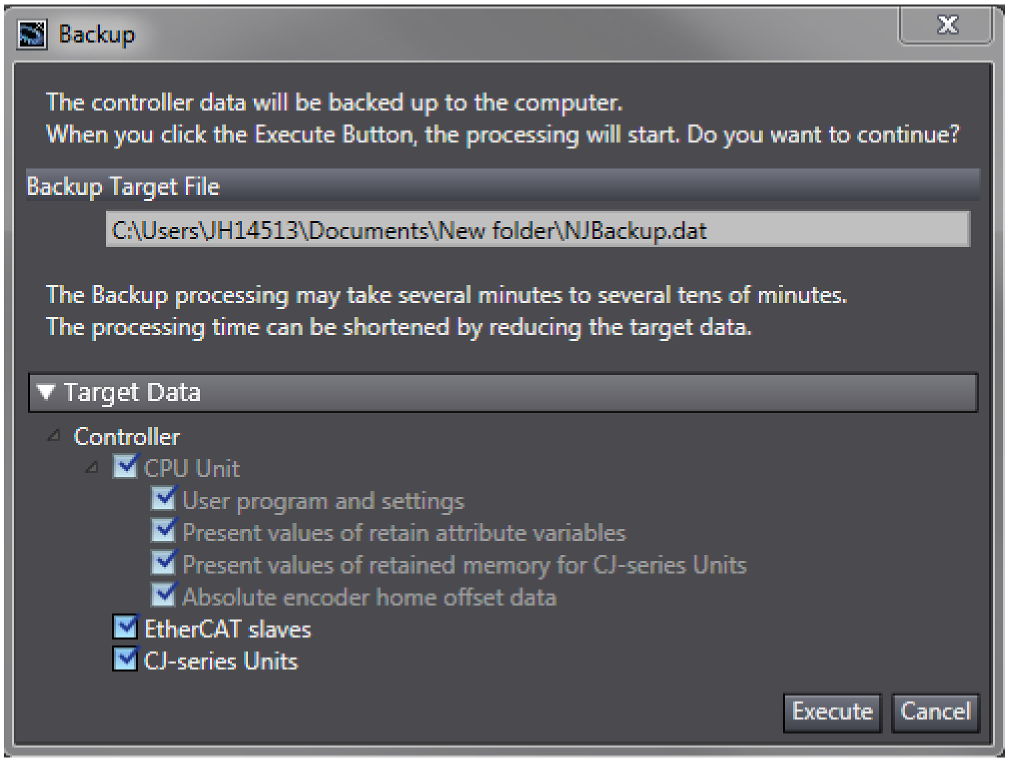
Locking the Project Down (Passwords and Backups)

Backup Configuration on the NJ

Backup to Laptop Program and All I/O Setup

Go online with NJ.
Notice that you can backup program and memory or just the variables.




Backup

The controller data will be backed up to the computer.
When you click the Execute Button, the processing will start. Do you want to continue?

Backup Target File

C:\Users\JH14513\Documents\New folder\NJBackup.dat

The Backup processing may take several minutes to several tens of minutes.
The processing time can be shortened by reducing the target data.

Target Data

- Controller
  - ☑ CPU Unit
    - ☑ User program and settings
    - ☑ Present values of retain attribute variables
    - ☑ Present values of retained memory for CJ-series Units
    - ☑ Absolute encoder home offset data
  - ☑ EtherCAT slaves
  - ☑ CJ-series Units

Notice that you can specify file name and location. You can specify what to backup including EtherCAT slave configuration data.

This backup takes a long time.

Backup Configuration on the NX Coupler

This is a future capability.

Backup All on the Memory Card

To Back-up the Program (no program tools) **W501 Section 9-1-2**

Insert SD card.
Set DIP switch pins 1-Off, 2-Off, 3-On, 4-Off
Press the button beside the SD card slot for 3 second. SD power indicator will flash. On 3 seconds off 0.5 seconds until it is done. The SD power light will turn off when done. Turn off all DIP switches.
To remove the memory card press the SD card PWR switch.

To Restore the Program

Turn off the NJ and EtherCAT slaves.
Insert SD card 9 (with backup program)
Set DIP switch to 1-Off, 2-Off, 3-On, 4-On
Turn on the NJ and EtherCAT slave power.
Backup file should be in the root directory.

The NJ Program Can Also Backup the Program and Data

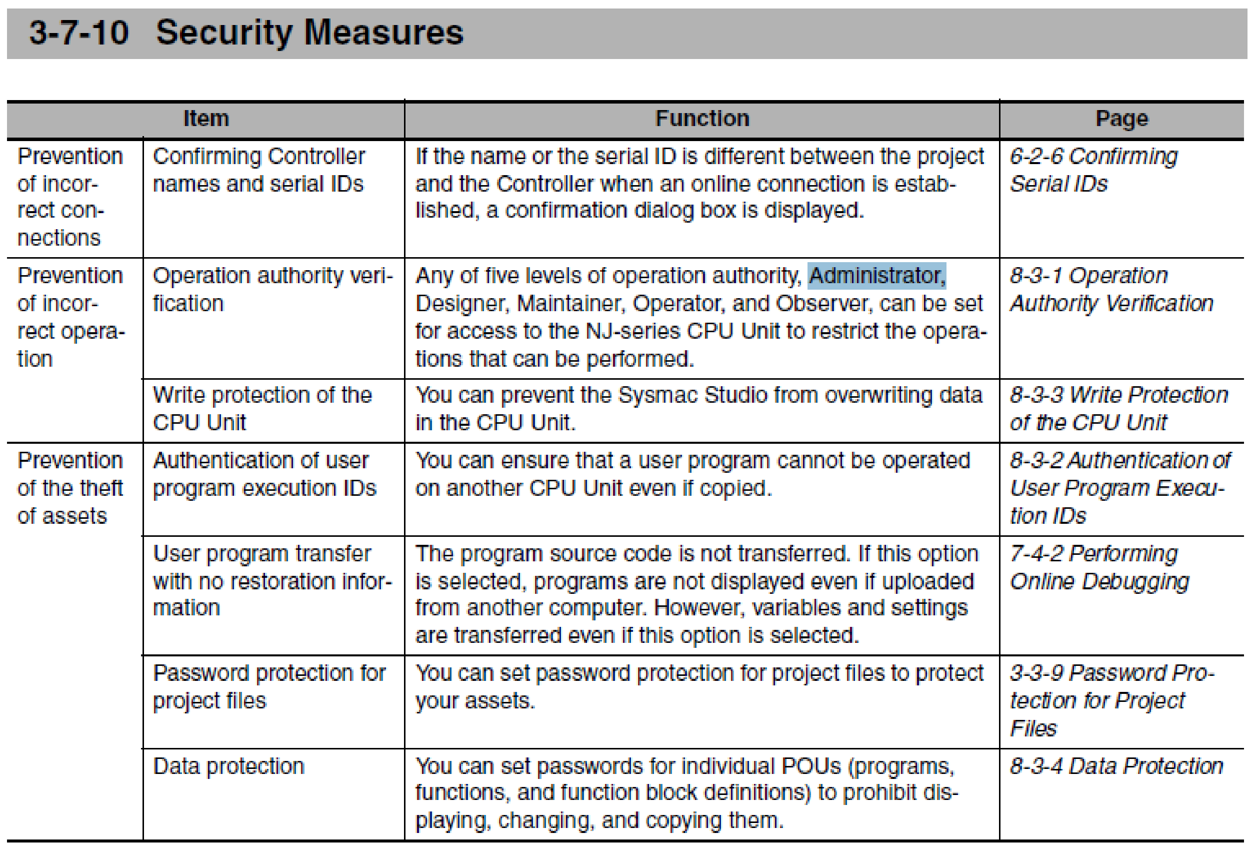
To backup the program and all configuration data to the SD card on the NJ CPU you will use the "BackupToMemoryCard" instruction. You do not have control of the file name so you will have to change the directory name. Note that restore uses the root directory.

Backup All on the Computer

Password Protection

NJ: Security

Here is a good overview of the security options on an NJ.

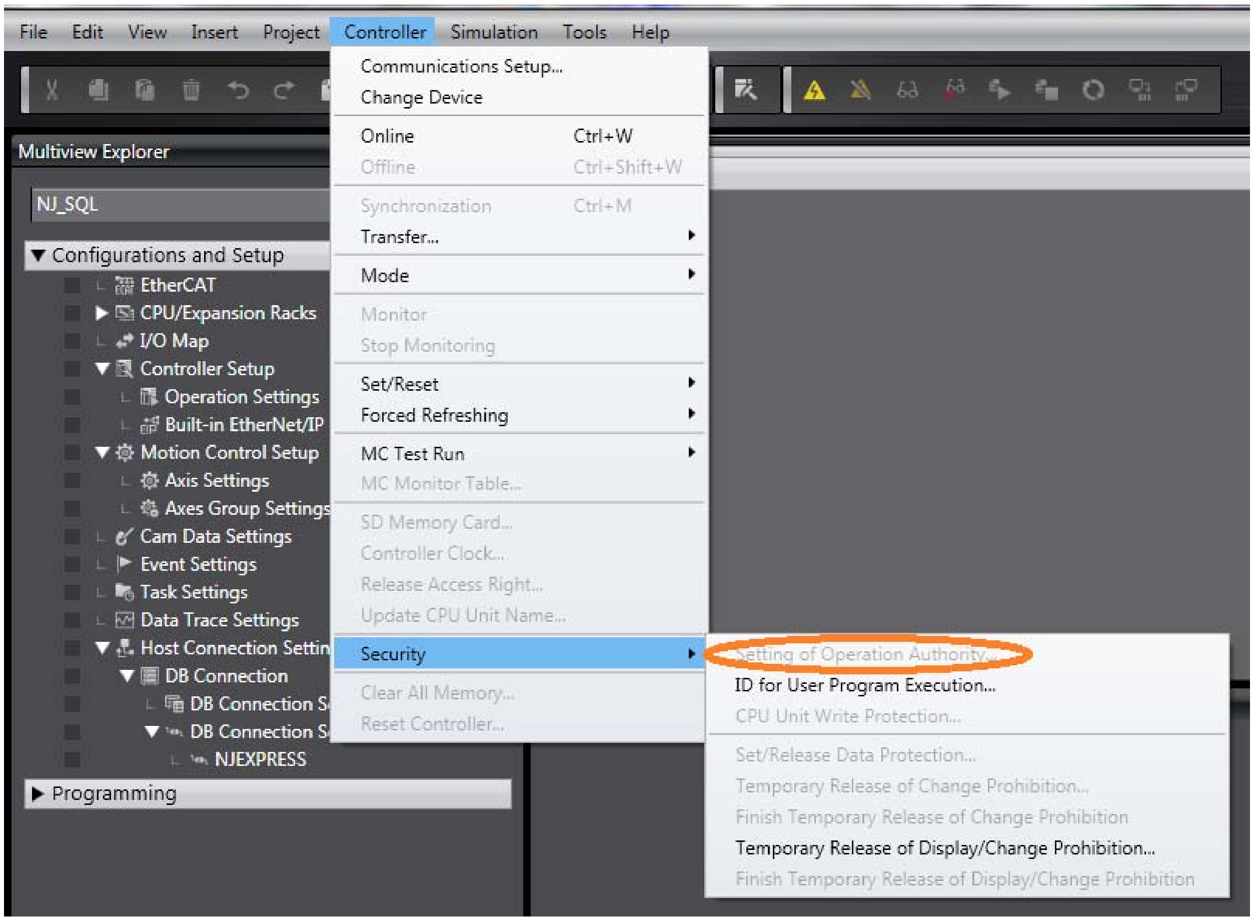


3-7-10 Security Measures

Item		Function	Page
Prevention of incorrect connections	Confirming Controller names and serial IDs	If the name or the serial ID is different between the project and the Controller when an online connection is established, a confirmation dialog box is displayed.	*6-2-6 Confirming Serial IDs*
Prevention of incorrect operation	Operation authority verification	Any of five levels of operation authority, Administrator, Designer, Maintainer, Operator, and Observer, can be set for access to the NJ-series CPU Unit to restrict the operations that can be performed.	*8-3-1 Operation Authority Verification*
	Write protection of the CPU Unit	You can prevent the Sysmac Studio from overwriting data in the CPU Unit.	*8-3-3 Write Protection of the CPU Unit*
Prevention of the theft of assets	Authentication of user program execution IDs	You can ensure that a user program cannot be operated on another CPU Unit even if copied.	*8-3-2 Authentication of User Program Execution IDs*
	User program transfer with no restoration information	The program source code is not transferred. If this option is selected, programs are not displayed even if uploaded from another computer. However, variables and settings are transferred even if this option is selected.	*7-4-2 Performing Online Debugging*
	Password protection for project files	You can set password protection for project files to protect your assets.	*3-3-9 Password Protection for Project Files*
	Data protection	You can set passwords for individual POUs (programs, functions, and function block definitions) to prohibit displaying, changing, and copying them.	*8-3-4 Data Protection*

Operation Authority Verification

Basically you can limit the access of the user to the program, online edit, etc. based on the user login.
This is described in section 8-3-1 of the manual W504.
This is where you set the user rights.

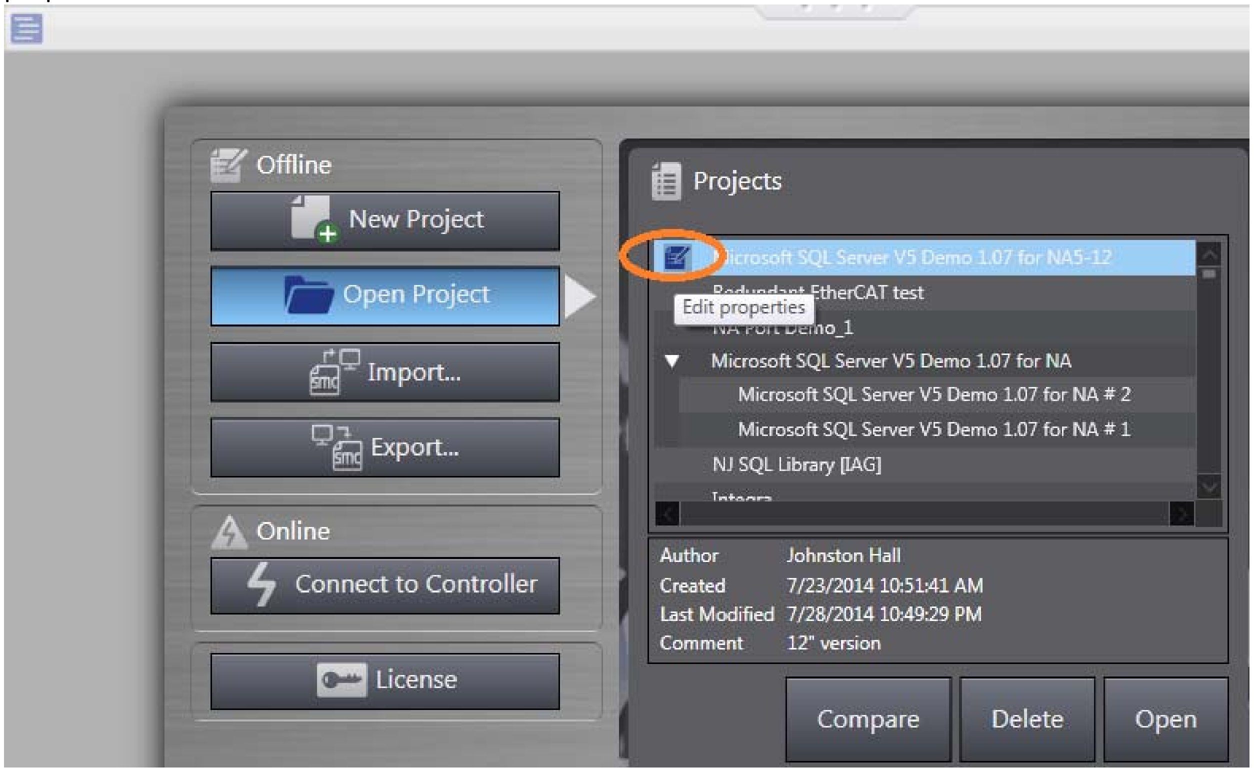


Passwords

Note: Administrator password cannot be bypassed or recovered.
W501 section 9-4

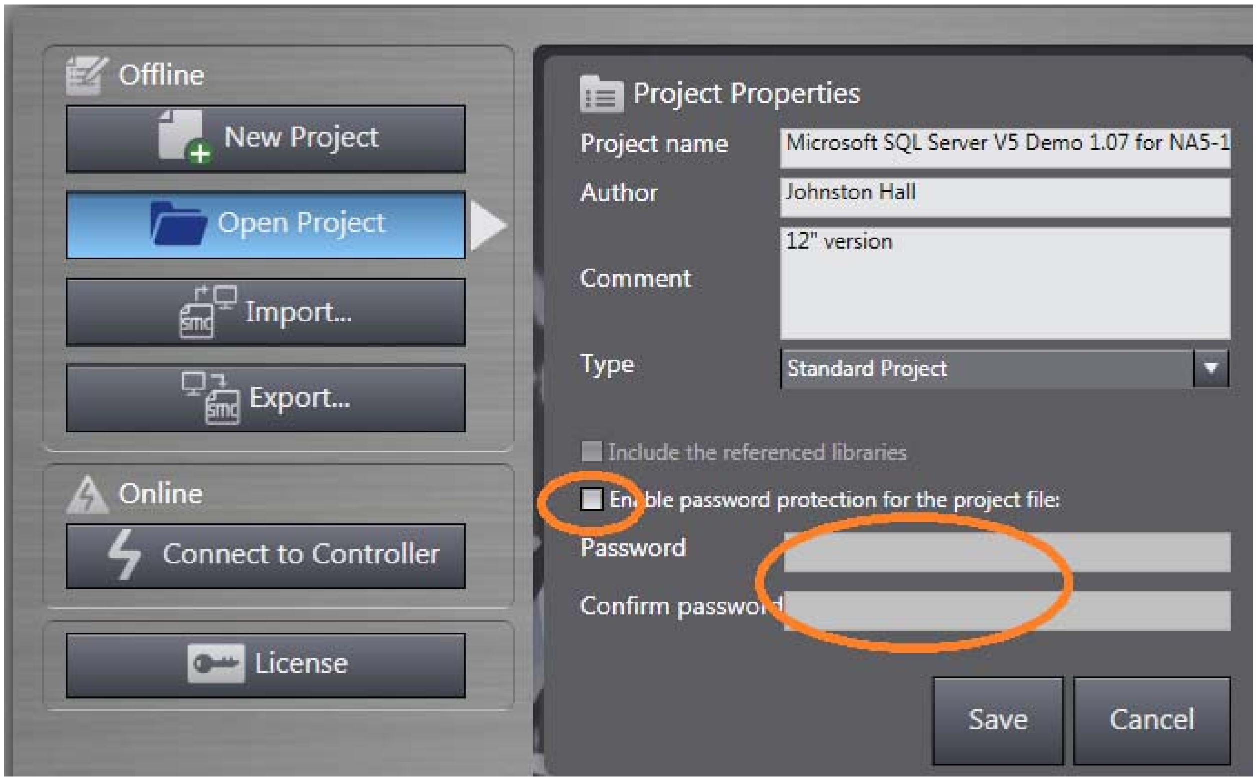
To Set and Reset the Project Password

When you open the projects you will see this page. Go to the left of the project name to edit the project properties as shown below and left click.



Author	Johnston Hall
Created	7/23/2014 10:51:41 AM
Last Modified	7/28/2014 10:49:29 PM
Comment	12" version

Next you will get this window so you can enter the password.



Project name	Microsoft SQL Server V5 Demo 1.07 for NA5-1
Author	Johnston Hall
Comment	12" version
Type	Standard Project

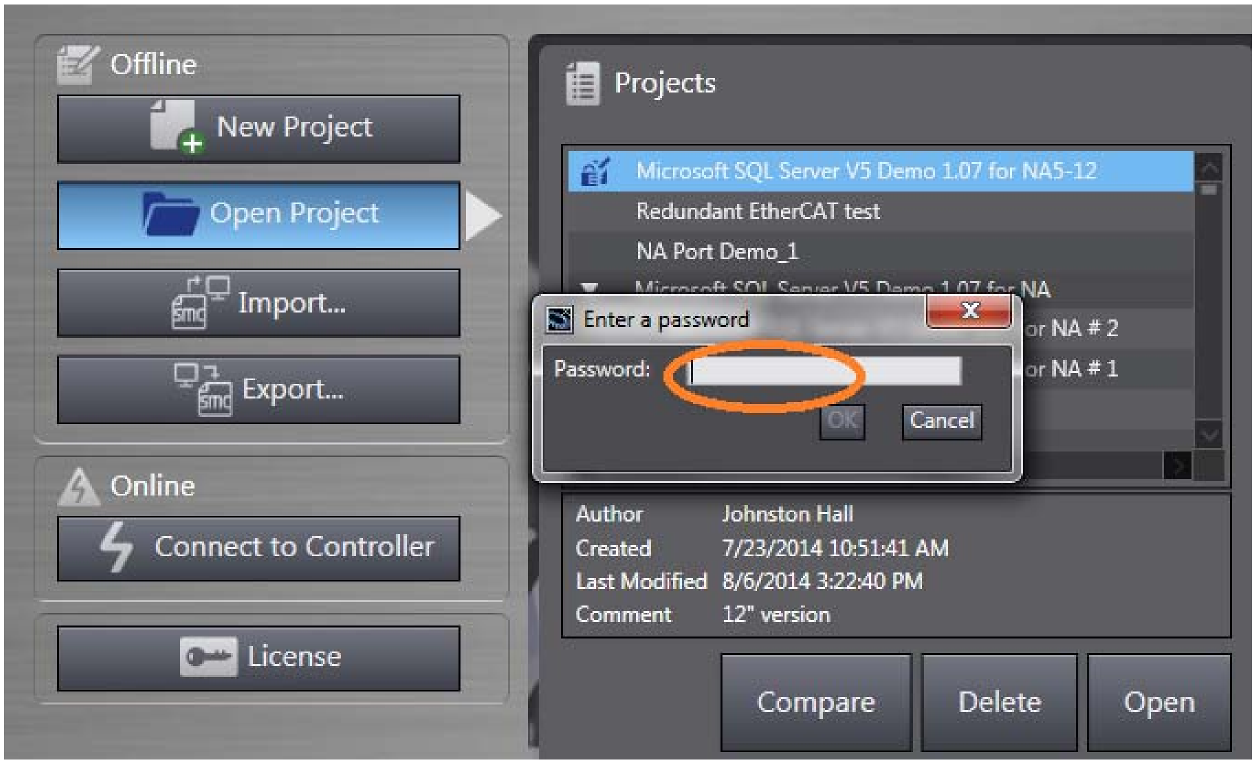
☐ Include the referenced libraries

☐ Enable password protection for the project file:

Password

Confirm password

From then on when you try to open the project it will prompt you for the password



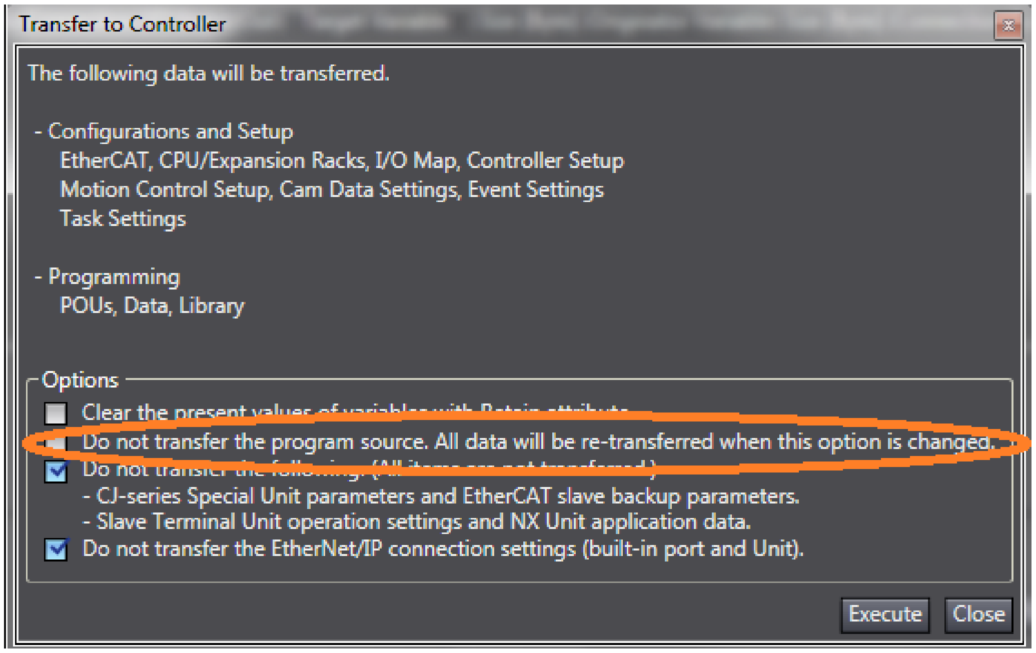
Microsoft SQL Server V5 Demo 1.07 for NA5-12
Redundant EtherCAT test
NA Port Demo_1
Microsoft SQL Server V5 Demo 1.07 for NA
or NA # 2
or NA # 1

Author	Johnston Hall
Created	7/23/2014 10:51:41 AM
Last Modified	8/6/2014 3:22:40 PM
Comment	12" version

To get rid of the password, go back to the edit properties for the project and disable the password protection.

Transfer the Program Without Source Data so it Cannot be Uploaded

This is the simplest way to prevent people from uploading your program out of the NJ.
Manual W501 section 9-4-2



Transfer to Controller

The following data will be transferred.

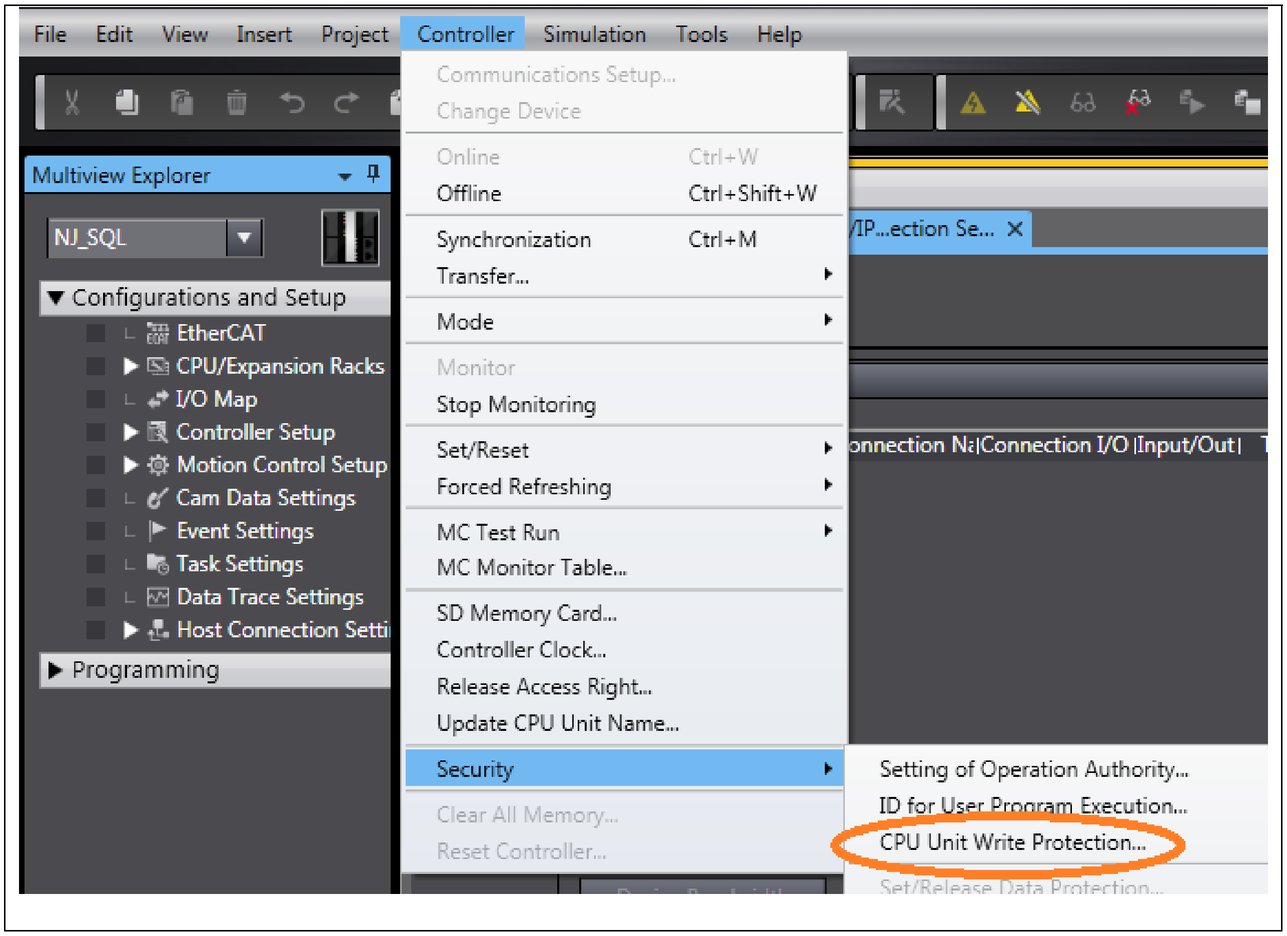
- Configurations and Setup
  EtherCAT, CPU/Expansion Racks, I/O Map, Controller Setup
  Motion Control Setup, Cam Data Settings, Event Settings
  Task Settings

- Programming
  POUs, Data, Library

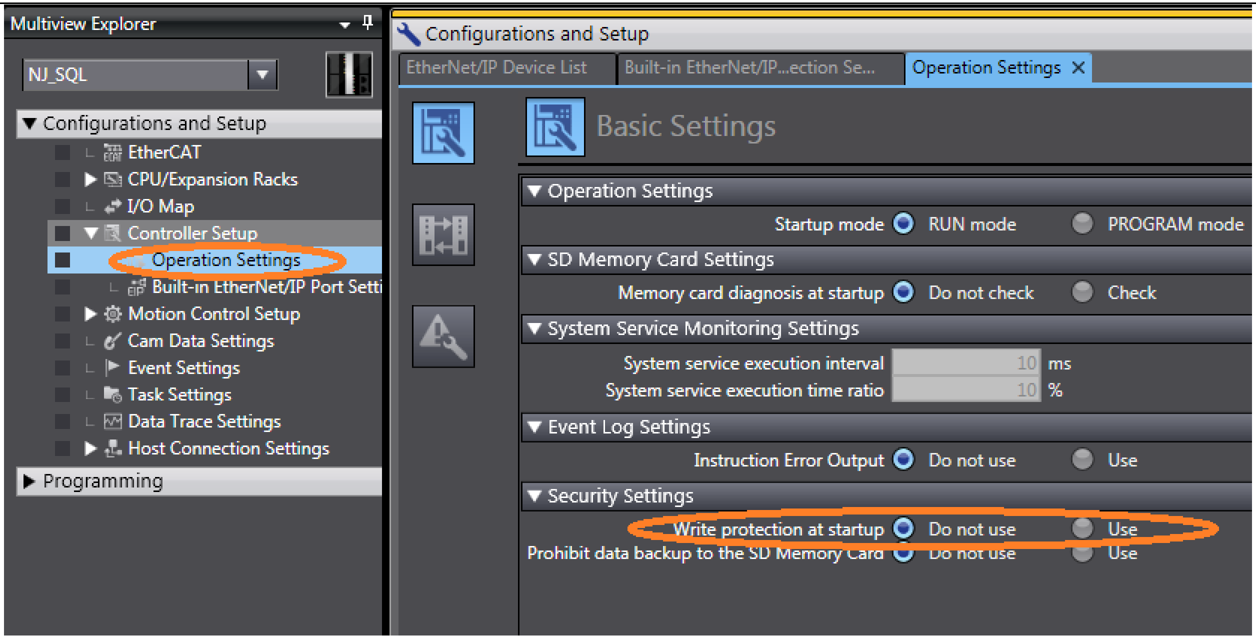
Options

- [ ] Clear the present values of variables with Retain attribute.
- Do not transfer the program source. All data will be re-transferred when this option is changed.
- [x] Do not transfer the following. (All items are not transferred.)
  - CJ-series Special Unit parameters and EtherCAT slave backup parameters.
  - Slave Terminal Unit operation settings and NX Unit application data.
- [x] Do not transfer the EtherNet/IP connection settings (built-in port and Unit).

CPU Unit Write Protection (Simple Protection with No Password)
This is a simple protection from people over writing the program. No password is used.
W501 section 9-4-3
First enable the protection in the NJ setup.



Then you can disable the protection in Sysmac Studio without a password.


Multiview Explorer

NJ_SQL

- Configurations and Setup
  - EtherCAT
  - CPU/Expansion Racks
  - I/O Map
  - Controller Setup
    - Operation Settings
    - Built-in EtherNet/IP Port Setti
  - Motion Control Setup
  - Cam Data Settings
  - Event Settings
  - Task Settings
  - Data Trace Settings
  - Host Connection Settings
- Programming

Configurations and Setup

EtherNet/IP Device List | Built-in EtherNet/IP...ection Se... | Operation Settings

Basic Settings

Operation Settings

Startup mode [✓ RUN mode] PROGRAM mode

SD Memory Card Settings

Memory card diagnosis at startup [✓ Do not check] Check

System Service Monitoring Settings

System service execution interval 10 ms
System service execution time ratio 10 %

Event Log Settings

Instruction Error Output [✓ Do not use] Use

Security Settings

Write protection at startup [✓ Do not use] Use
Prohibit data backup to the SD Memory Card [✓ Do not use] Use

User Program Execution ID (Set CPU to Program – Prevents Wrong Program Transfer)

W501 User Manual, Section 9-4-4. Every program has a “user program execution ID”. You can transfer the ID into the NJ so that it will only run that program. You can view edit the program. The most likely use of this feature is when more than one NJ is in the project and there is a possibility of the user transferring the wrong NJ program to the wrong NJ.

CPU Unit Names

Each CPU remembers it’s name and serial ID. If you try to transfer the wrong program the first thing the download will tell you is that the name or serial ID does not match. You can continue but you now know that this is not the CPU you sent the program to last time.

Function Blocks

A future enhancement.

NA HMI Program

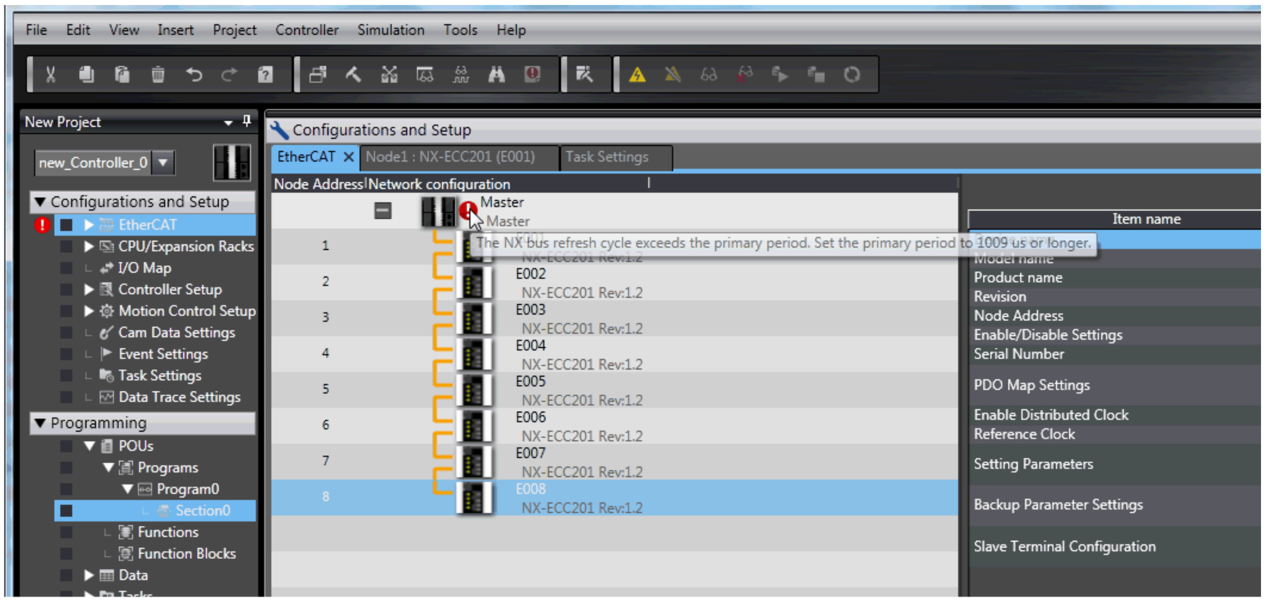
At this time the NA program cannot be uploaded and therefore password protection is not needed.

Appendix A

Calculating Maximum EtherCAT I/O

We know:
Maximum number of slaves is 192
The maximum bytes per slave is 1434 bytes in and 1434 bytes out.
The maximum bytes per master is 5736 bytes in and 5736 bytes out.

Sysmac Studio Warns you if Scan Time Cannot be Met Due to I/O Size



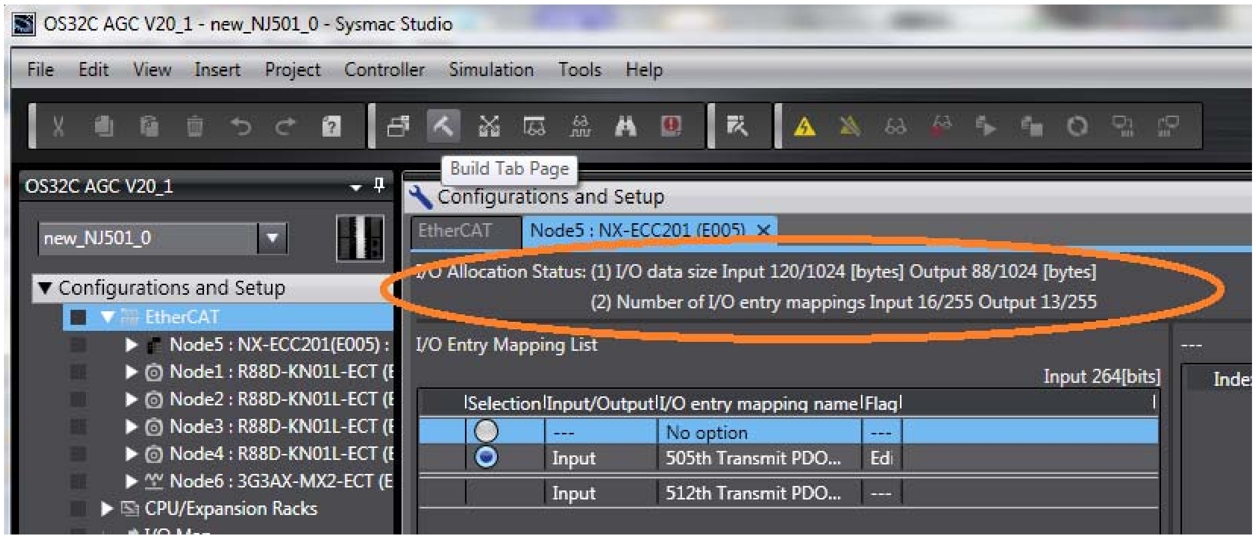
The NX bus refresh cycle exceeds the primary period. Set the primary period to 1009 us or longer.

Node Address	Network configuration
	Master Master
1	E001 NX-ECC201 Rev:1.2
2	E002 NX-ECC201 Rev:1.2
3	E003 NX-ECC201 Rev:1.2
4	E004 NX-ECC201 Rev:1.2
5	E005 NX-ECC201 Rev:1.2
6	E006 NX-ECC201 Rev:1.2
7	E007 NX-ECC201 Rev:1.2
8	E008 NX-ECC201 Rev:1.2

Sysmac Studio Warns you if the I/O Count is Too High

NX Coupler

The number of I/O on each coupler is configurable so there is no fixed I/O size for each coupler. Sysmac Studio is good at showing us how many bytes and I/O have been used on each NX EtherCAT coupler as shown below.


OS32C AGC V20_1 - new_NJ501_0 - Sysmac Studio
File Edit View Insert Project Controller Simulation Tools Help
Build Tab Page
OS32C AGC V20_1
new_NJ501_0
Configurations and Setup
EtherCAT
Node5 : NX-ECC201(E005) :
Node1 : R88D-KN01L-ECT (E
Node2 : R88D-KN01L-ECT (E
Node3 : R88D-KN01L-ECT (E
Node4 : R88D-KN01L-ECT (E
Node6 : 3G3AX-MX2-ECT (E
CPU/Expansion Racks
Configurations and Setup
EtherCAT
Node5 : NX-ECC201 (E005)

I/O Allocation Status: (1) I/O data size Input 120/1024 [bytes] Output 88/1024 [bytes]
(2) Number of I/O entry mappings Input 16/255 Output 13/255

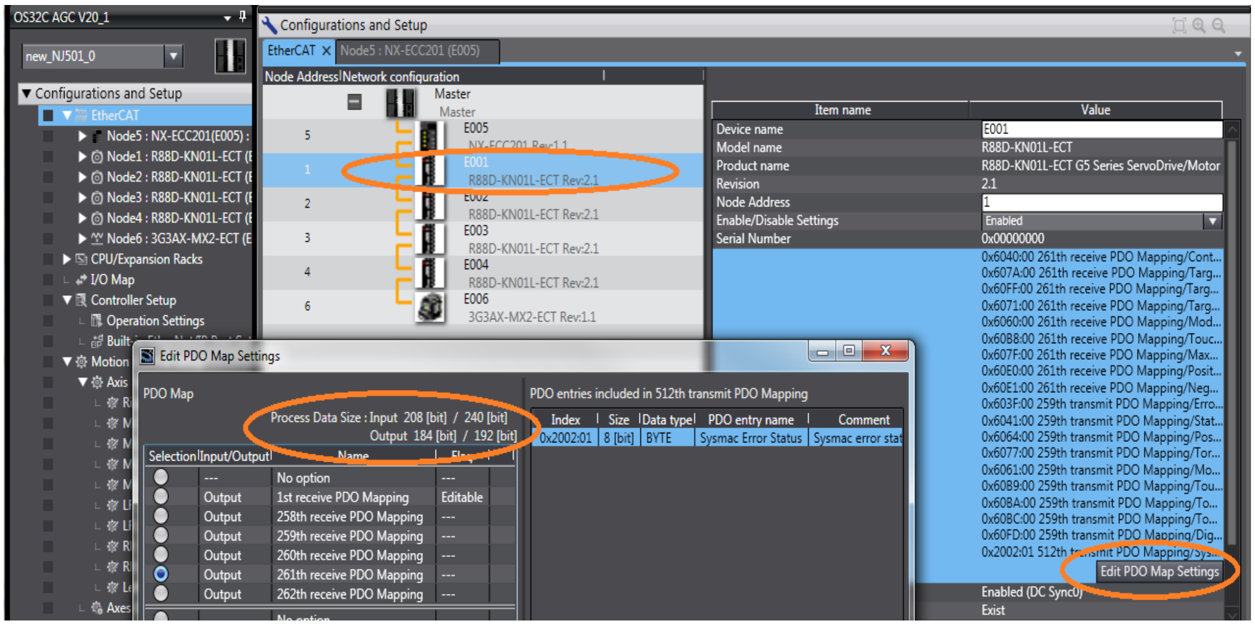
I/O Entry Mapping List

Input 264[bits]

Selection	Input/Output	I/O entry mapping name	Flag
○	---	No option	---
●	Input	505th Transmit PDO...	Edi
	Input	512th Transmit PDO...	---

G5 Servo Drive

The maximum packet size is 240 bits in (30 bytes) and 192 bits out (24 bytes).
And Sysmac Studio shows how many bytes are being used by the G5 servo drive.



Node Address	Network configuration
	Master Master
5	E005 NX-ECC201 Rev:1.1
1	E001 R88D-KN01L-ECT Rev:2.1
2	E002 R88D-KN01L-ECT Rev:2.1
3	E003 R88D-KN01L-ECT Rev:2.1
4	E004 R88D-KN01L-ECT Rev:2.1
6	E006 3G3AX-MX2-ECT Rev:1.1

Item name	Value
Device name	E001
Model name	R88D-KN01L-ECT
Product name	R88D-KN01L-ECT G5 Series ServoDrive/Motor
Revision	2.1
Node Address	1
Enable/Disable Settings	Enabled
Serial Number	0x00000000
	0x6040:00 261th receive PDO Mapping/Cont... 0x607A:00 261th receive PDO Mapping/Targ... 0x60FF:00 261th receive PDO Mapping/Targ... 0x6071:00 261th receive PDO Mapping/Targ... 0x6060:00 261th receive PDO Mapping/Mod... 0x60B8:00 261th receive PDO Mapping/Touc... 0x607F:00 261th receive PDO Mapping/Max... 0x60E0:00 261th receive PDO Mapping/Posit... 0x60E1:00 261th receive PDO Mapping/Neg... 0x603F:00 259th transmit PDO Mapping/Erro... 0x6041:00 259th transmit PDO Mapping/Stat... 0x6064:00 259th transmit PDO Mapping/Pos... 0x6077:00 259th transmit PDO Mapping/Tor... 0x6061:00 259th transmit PDO Mapping/Mo... 0x60B9:00 259th transmit PDO Mapping/Tou... 0x60BA:00 259th transmit PDO Mapping/To... 0x60BC:00 259th transmit PDO Mapping/To... 0x60FD:00 259th transmit PDO Mapping/Dig... 0x2002:01 512th transmit PDO Mapping/Sys...
	Enabled (DC Sync0)
	Exist

Process Data Size : Input 208 [bit] / 240 [bit]
Output 184 [bit] / 192 [bit]

Selection	Input/Output	Name	Flag
○	---	No option	---
○	Output	1st receive PDO Mapping	Editable
○	Output	258th receive PDO Mapping	---
○	Output	259th receive PDO Mapping	---
○	Output	260th receive PDO Mapping	---
●	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapping	---
○		No option	

PDO entries included in 512th transmit PDO Mapping

Index	Size	Data type	PDO entry name	Comment
0x2002:01	8 [bit]	BYTE	Sysmac Error Status	Sysmac error stat

MX2 Inverter Drive

The maximum packet size is 320 bits (40 bytes in) and 320 bits out (40 bytes).
Sysmac Studio will show the data size for each MX2.

Glossary

EtherNet/IP	An Ethernet based communication system using the CIP protocol. The NJ uses
EtherNet/IP	For NJ to computer, NJ to HMI, NJ to NJ, NJ to some IO and drives, communications.
Global Variable	This is a variable that should be accessible from many programs or devices.
Local I/O	I/O connected physically to the NJ CPU.
Local Variable	A variable that is only accessible from the program it was declared in.
NA	The Omron NA series of touch screens for use with NJ and CJ controllers.
NJ	The Omron NJ series CPU's
Remote I/O	I/O connected to the CPU by a communication cable (example: EtherCAT)
Tags	Instead of using fixed numbering system for I/O the NJ uses a free allocation of memory and you give each I/O or variable a name.
Task	The NJ series controller runs several tasks (Program and IO refresh) in a time slice manner. Each task gets so many microseconds (µs) of time to execute then the processor moves on to the next task (comes back to the current task later).

Revision History

Rev 1.0	First Release.	October 2, 2014
Rev 1.1	Added MX2 drive setup on EtherCAT.	October 15, 2014

P37I-E-01 10/14 Version 1.1 Note: Specifications are subject to change. © 2014 Omron Electronics LLC Printed in U.S.A.

Printed on recycled paper.