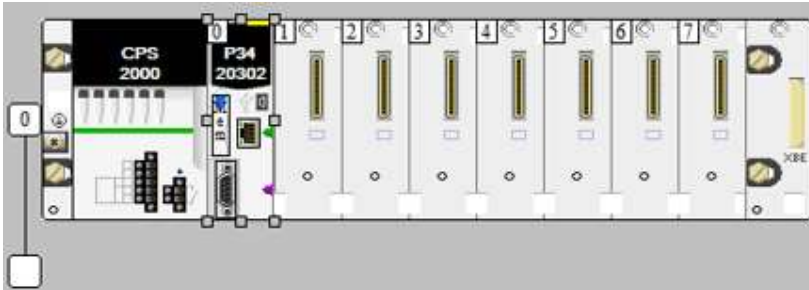


M340 to IE-NODE

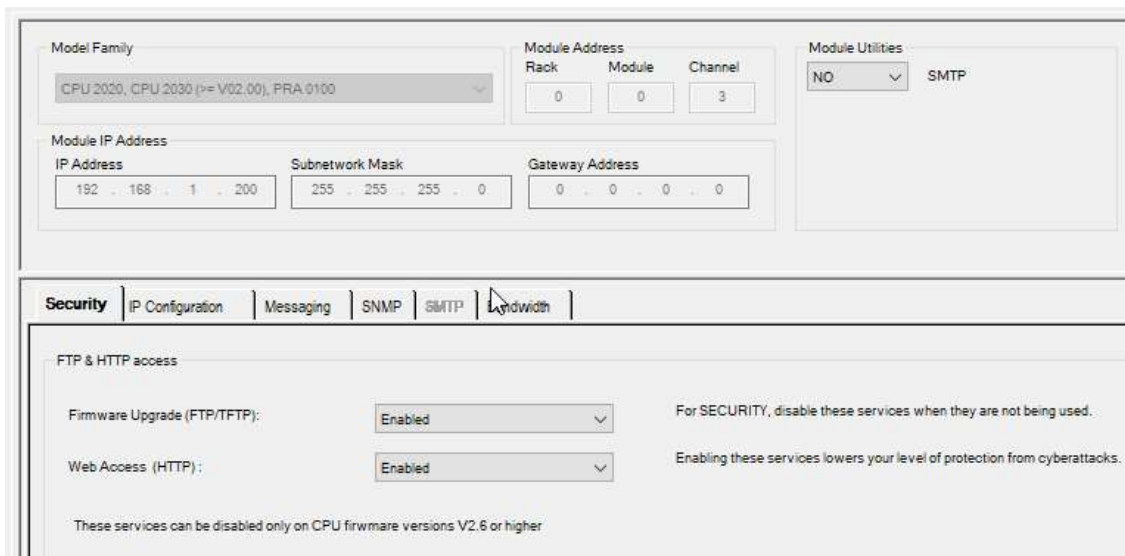
The following was done using an BMXP3420302 V2.5 and Control expert V15.1.

PLC architecture



Hardware setup

Dependant on the version of Control Expert and CPU firmware may mean you have to enable the firmware and web access. This should be OK if your M340 firmware is above V2.6 as it says in the image below.



Create an Ethernet network in the communications folder – this has the default name of Ethernet_1.



Setup the comms as below:

Model Family CPU 2020, CPU 2030 (>= V02.00), FRA.0100		Module Address Rack: 0, Module: 0, Channel: 3			Module Utilities NO: [v] SMTP
Module IP Address IP Address: 192 . 168 . 1 . 200		Subnetwork Mask: 255 . 255 . 255 . 0		Gateway Address: 0 . 0 . 0 . 0	

Security | **IP Configuration** | Messaging | SNMP | SMTP | Bandwidth

IP address configuration

Configured

IP address: 192 . 168 . 1 . 200

Subnetwork mask: 255 . 255 . 255 . 0

Gateway address: 0 . 0 . 0 . 0

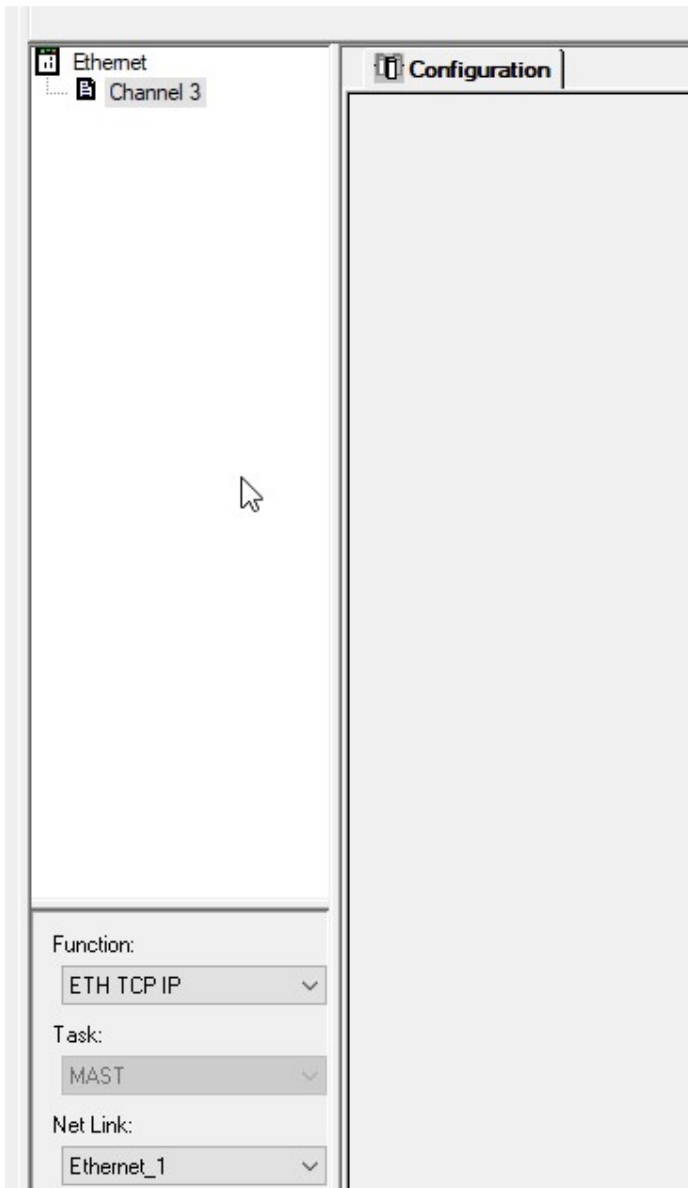
From a server

Device Name: []

Ethernet configuration

Ethernet II 802.3

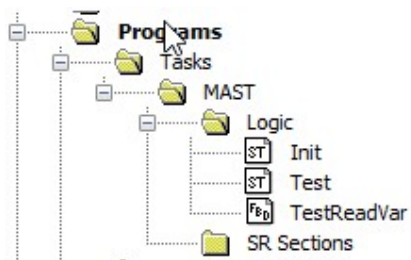
The model is chosen to be the port out of the CPU, and the IP address is set to 192.168.1.200.



You must change the function and attach the net link to the previously configured Ethernet_1 network.

Program:

There are 3 program sections:



Init: This is the section that sets up some of the communication parameters needed to communicate with the IE-NODE.

Test: This is a simple counter to make sure the program is running (the user can animate the counter)

TestReadVar: This is the communication section where you will find the FBs to perform the read of the information in the IE-NODE.

We also use the following variables in the program:

Name	Type
address	string[30]
counter	INT
management	ARRAY[0..3] OF INT
numbertoread	INT
obtype	STRING
readbuffer	ARRAY[0..74] OF INT
serverreadreg	DINT
start	BOOL

The Init Section:

This is a set of commands to load into a FB called ADDM in the Estrada section. It is described as follows:

```

(*Initialise communications to the IE Node.*)
IF %s13 THEN                                (* if first scan initialize variables *)
  obtype      := '%IW';                      (* Reading registers, %IW's *)
  serverreadreg := 0;                        (* we want to read %IW0 *)
  numbertoread := 75;                       (*reading 75 registers *)
  management[2] := 50;                       (*The timeout value *)
  management[3] := 75;                       (*length again.*/)
  start       := TRUE;                       (*Set the start variable to TRUE to receive information every scan*)
END_IF;

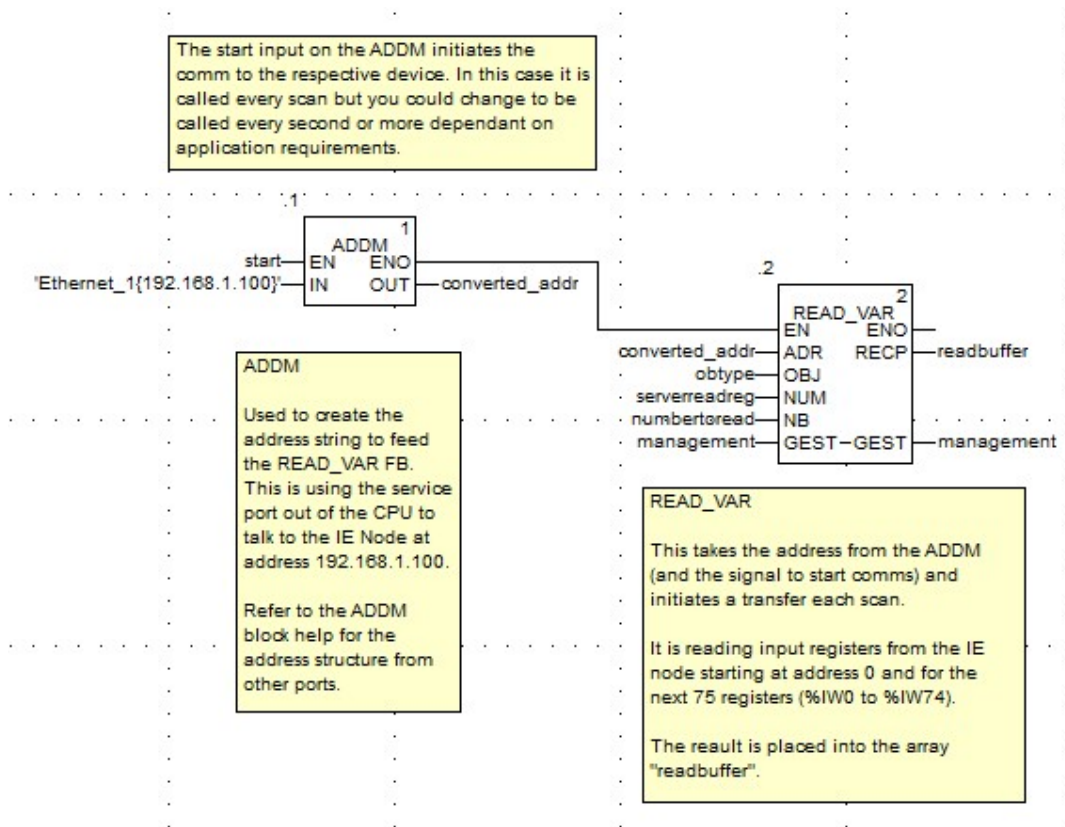
```

%S13 is a system bit that is on the first PLC scan only, hence these registers are setup when the PLC boots up only, kind of an initialisation like the routine says!

The object type needed for the IE-NODE is %IW as the data in the device is located in input registers (3x registers). The information starts at location 0 and ends at location 74. So, the start address is set to 0, and the size of the read allocation is 75 variables. There is a timeout value set at 5 seconds (50x 100ms).

The TestReadVar section:

This is the code to read the registers in the IE-NODE:

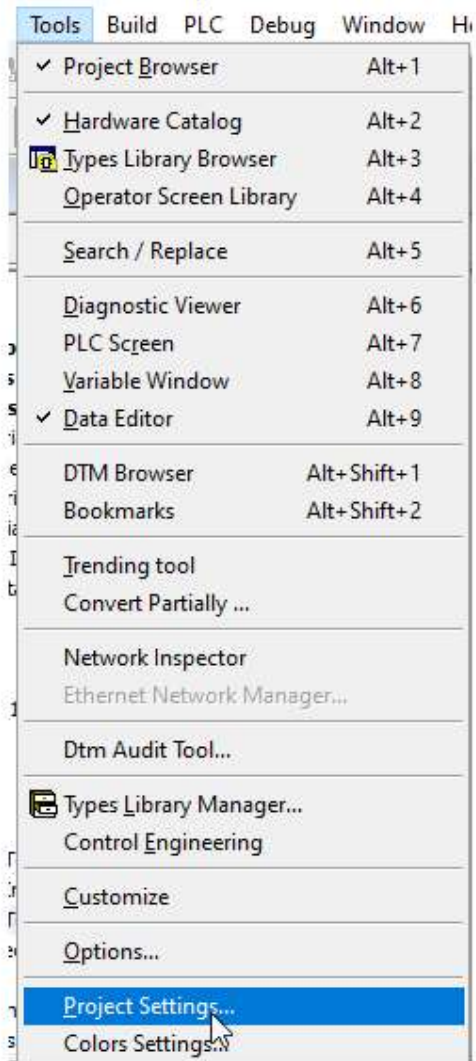


So, in the Init section we prepare the registers and then set the “start” Boolean to TRUE. This then means the ADDM function is performed and the address of the equipment to read is passed in the right format to the READ_VAR FB. The address in this case is Ethernet_1{192.168.1.100}. That is the default address of the IE-NODE but must be set to match. Be aware that the subnets also must match in both equipment. Once this is done it also then triggers the READ_VAR FB to perform a read of 75 input registers, from address 0 and put the result in the readbuffer array. It is important to note that in this example the communication is set up to use the service port of the PLC on the CPU as that is where we attached the Ethernet_1 network. Refer to the online help if you are using another port like in an in-rack module.

See below the result of the transaction in an animation table:

Name	Value	Type	Comme
readbuffer		ARRAY[0..74] O...	
readbuffer[0]	2	INT	
readbuffer[1]	1	INT	
readbuffer[2]	3281	INT	
readbuffer[3]	2	INT	
readbuffer[4]	4	INT	
readbuffer[5]	516	INT	
readbuffer[6]	0	INT	
readbuffer[7]	1026	INT	
readbuffer[8]	0	INT	
readbuffer[9]	0	INT	
readbuffer[10]	0	INT	
readbuffer[11]	193	INT	

Data is being received. If there is an error, then you should refer to the online help for the READ_VAR and specifically the management table as it will lead you to the cause of the fault. You may also get an error when you first build the project when you create it yourself that dynamic arrays are not enabled. You can enable them in the Tools menu under project settings:



And in the variables list there is a tick box to allow these.

Project Settings

	Property label	Property value
Project Settings	Allow leading digits	<input type="checkbox"/>
General	Character set	Standard
Management of build messages	Allow usage of EBOOL edge	<input checked="" type="checkbox"/>
Build settings	Allow INT/DINT in place of ANY_BIT	<input checked="" type="checkbox"/>
Project autosaving on download	Allow bit extraction of INT, WORD and BYTE	<input checked="" type="checkbox"/>
PLC embedded data	Directly represented array variables	<input type="checkbox"/>
PLC diagnostics	Allow dynamic arrays (ANY_ARRAY_XXX)	<input checked="" type="checkbox"/>
PLC behaviour	Disable array size compatibility check	<input type="checkbox"/>
Path	Enable fast scanning for trending	<input type="checkbox"/>
Time	Force references initialization	<input checked="" type="checkbox"/>
Configuration	Save Restore Data Request	<input type="checkbox"/>
Variables	Display private variables for DFB instances	<input type="checkbox"/>
Program		
Languages		
Common		

So, in summary the project does what it says on the tin and feel free to use.