How to control an ATV61 / ATV71 with a M340 CPU with built-in Ethernet port using the DATA_EXCH function

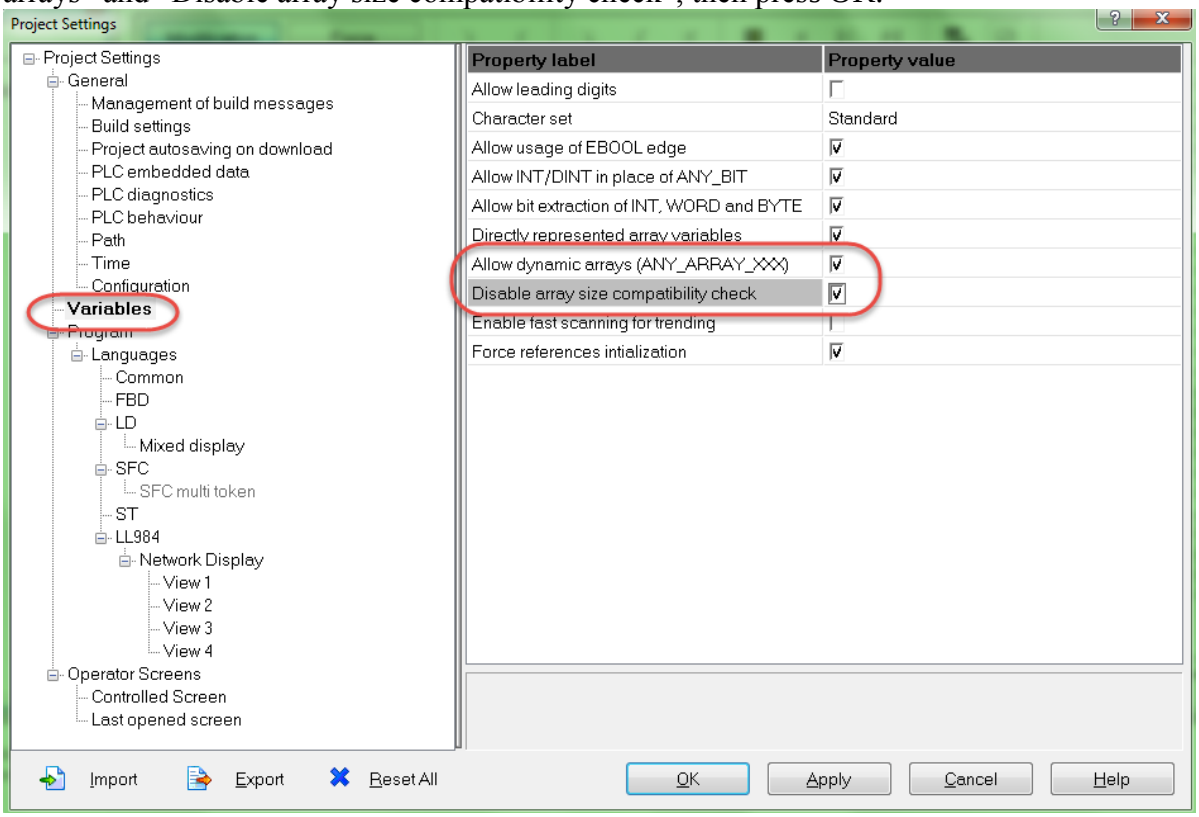
This document describes an application using a M340 CPU with a built-in Ethernet port connected to an ATV71 Ethernet card VW3A3310D and using the DATA_EXCH function. The M340 will be communicating with the IO Scanner service in the drive.

Hardware and software used:

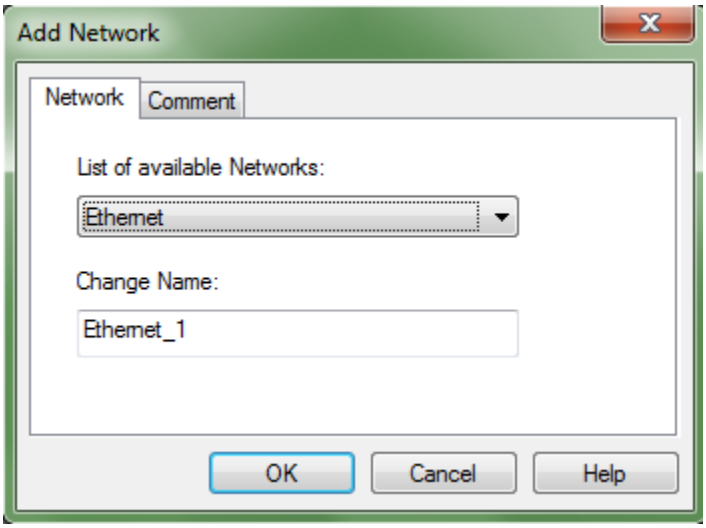
- BMXP3420302 CPU
- Altivar 71 drive
- VW3A3310D Modbus TCP/IP Daisy Chain Ethernet card
- One crossover cable connected directly from the BMXP3420302 CPU to the Altivar 71 Ethernet card VW3A3310D
- Unity Pro 8.1

Given below is a step by step procedure.

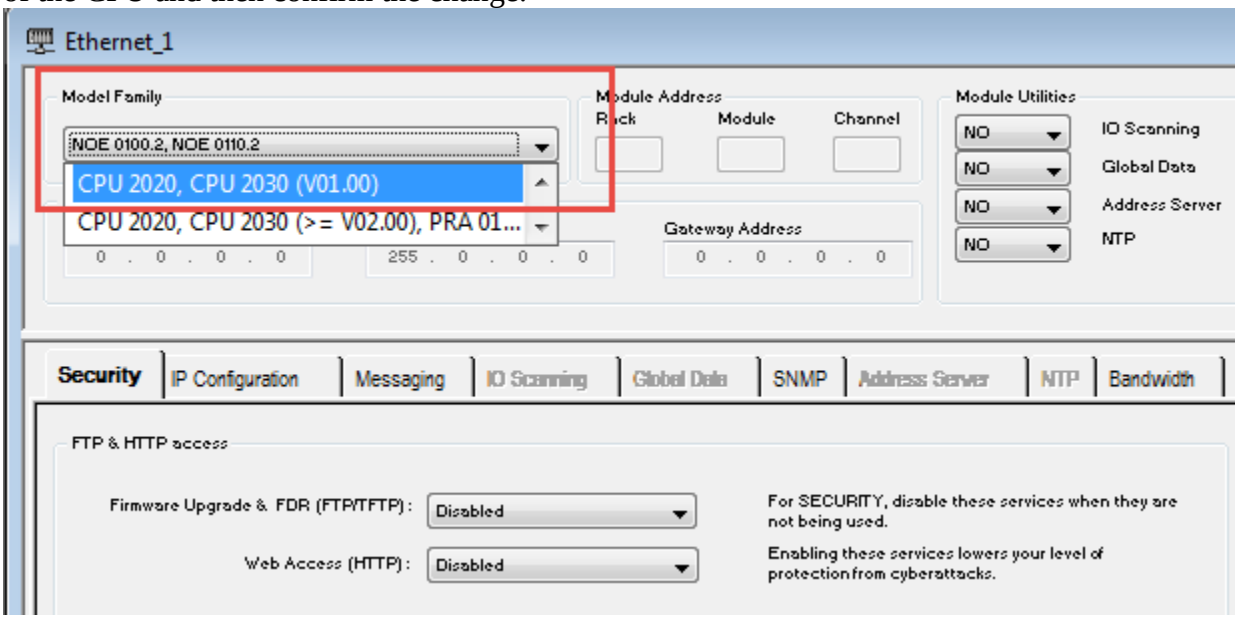
1. After creating a new project go to Tools / Project Setting and check “Allow dynamic arrays” and “Disable array size compatibility check”, then press OK.



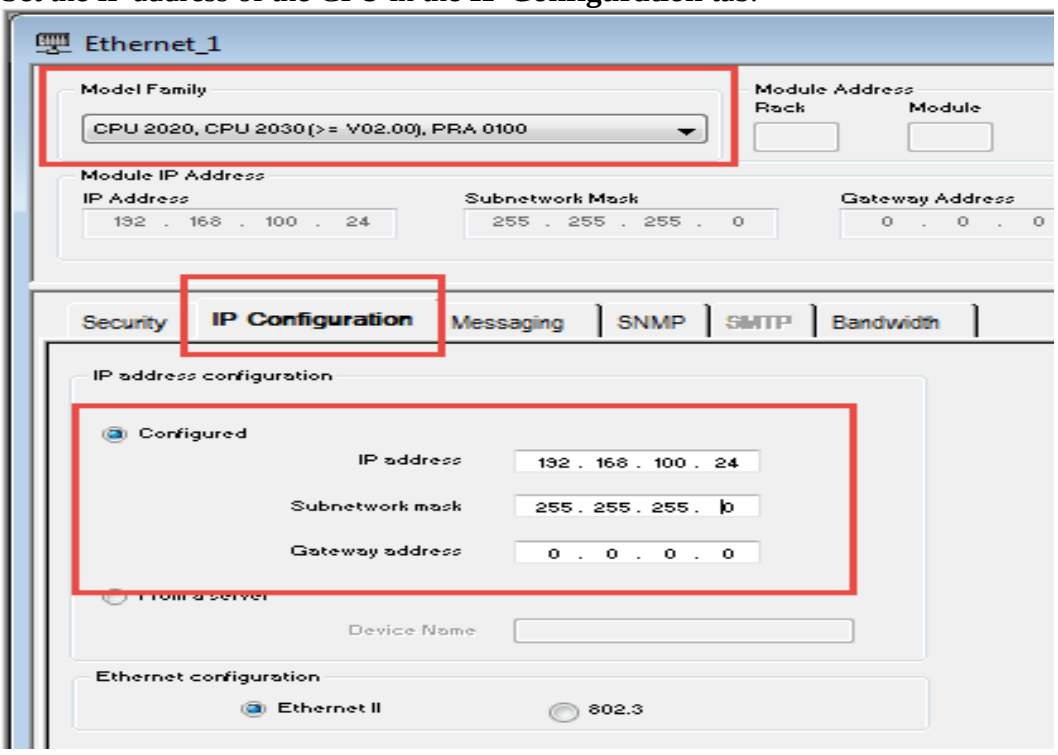
2. Go to the **Project Browser / Communication / Networks**, right-click on Networks and select **New Network...** For **List of available Networks** select **Ethernet** then press OK.



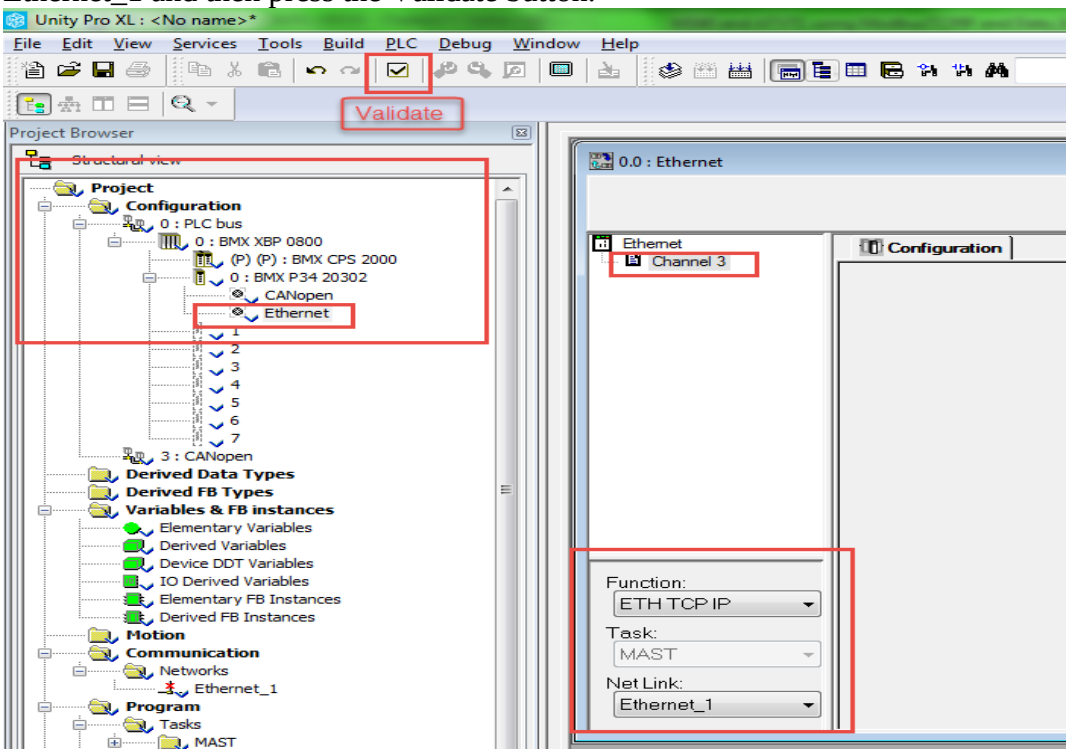
3. Go to **Project Browser / Communication / Networks** and double-click on Ethernet_1. For the Model Family select **CPU2020, CPU2030 (>= V02.00), PRA011** or your version of the CPU and then confirm the change.



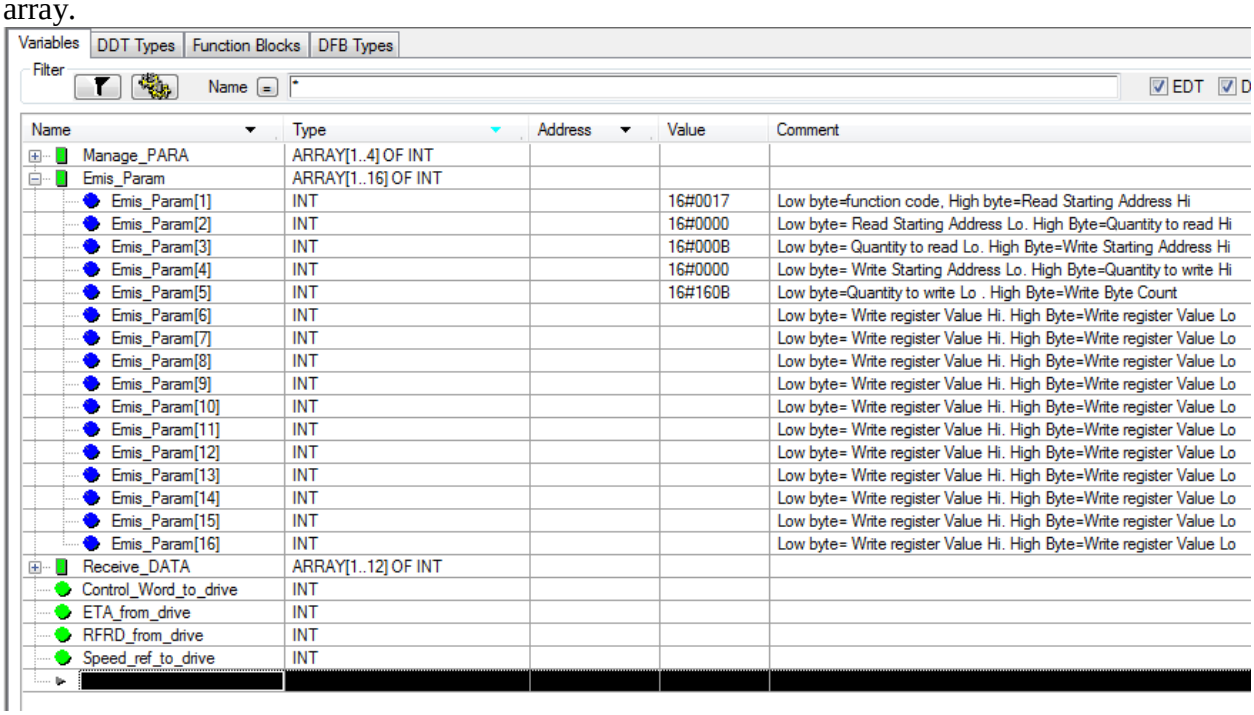
4. Set the IP address of the CPU in the **IP Configuration** tab.



5. Go to the **Project Browser / Configuration / PLC Bus / BMXP3420302 / Ethernet** and double-click on **Ethernet**. Select **Channel 3, Function ETH TCP IP** and **Net Link** to Ethernet_1 and then press the Validate button.

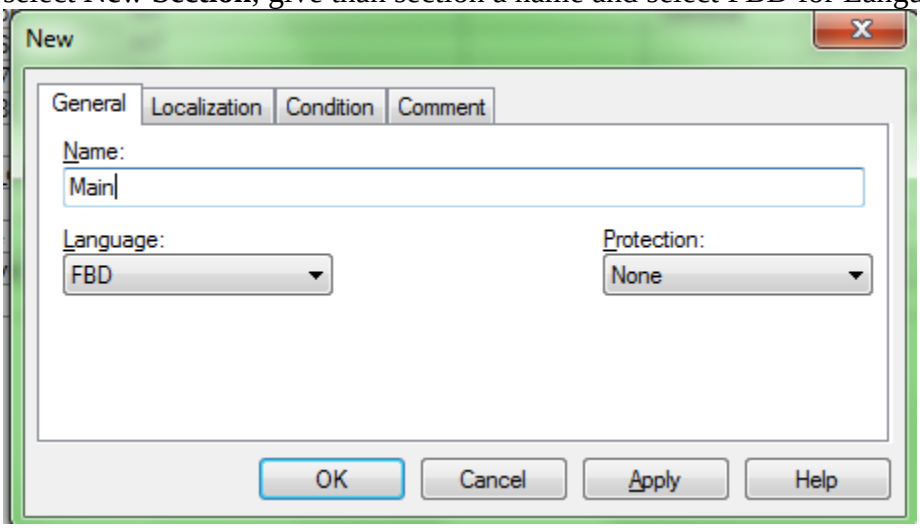


6. In the **Project Browser / Variables and FB Instances** create the following variables and make sure you assign the shown values to the first five elements of the Emis_Param array.

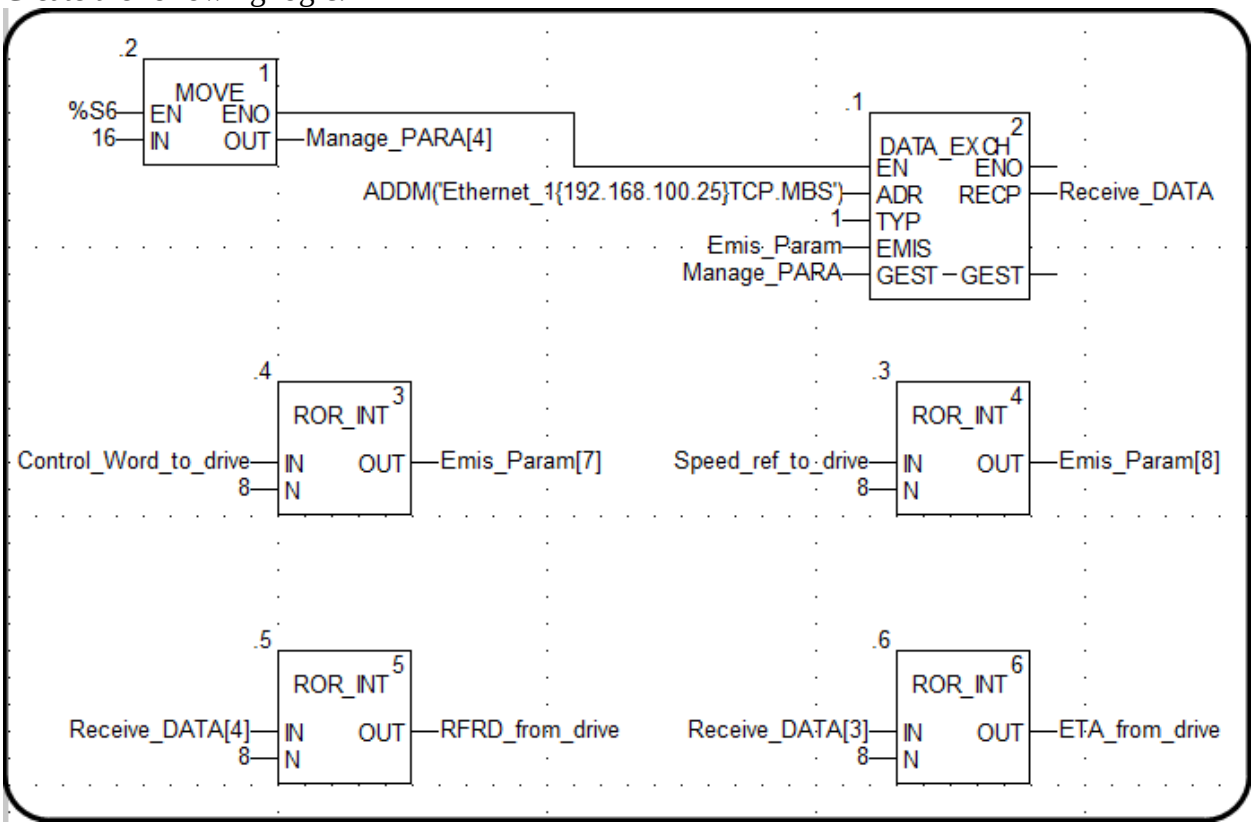


Name	Type	Address	Value	Comment
Manage_PARA	ARRAY[1..4] OF INT			
Emis_Param	ARRAY[1..16] OF INT			
Emis_Param[1]	INT		16#0017	Low byte=function code, High byte=Read Starting Address Hi
Emis_Param[2]	INT		16#0000	Low byte= Read Starting Address Lo. High Byte=Quantity to read Hi
Emis_Param[3]	INT		16#000B	Low byte= Quantity to read Lo. High Byte=Write Starting Address Hi
Emis_Param[4]	INT		16#0000	Low byte= Write Starting Address Lo. High Byte=Quantity to write Hi
Emis_Param[5]	INT		16#160B	Low byte=Quantity to write Lo . High Byte=Write Byte Count
Emis_Param[6]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[7]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[8]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[9]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[10]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[11]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[12]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[13]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[14]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[15]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Emis_Param[16]	INT			Low byte= Write register Value Hi. High Byte=Write register Value Lo
Receive_DATA	ARRAY[1..12] OF INT			
Control_Word_to_drive	INT			
ETA_from_drive	INT			
RFRD_from_drive	INT			
Speed_ref_to_drive	INT			

7. In the **Project Browser / Program / Task / Mast / Sections** right-click on **Sections,** select **New Section**, give than section a name and select FBD for Language.



8. Create the following logic.



9. **<u>DATA_EXCH parameters explanation:</u>**

- **ADR**: ADDM('Ethernet_1{192.168.100.25}TCP.MBS'). This pin has the ADDM function that will direct the message to Ethernet_1 network and then to the IP address of the ATV61 / ATV71. Change this with your ATV61 / AV71 IP address keeping the same syntax

- **TYP**:1. This is the only possible value, do not change

- **EMIS**: Emis_Para, array of 16 INT for data to send. This is where the Modbus message to be sent is assembled. The screenshot in Step 6, Comment area, explains what each register is. It is based on the Modbus Application Protocol Specification document, Modbus function code 23, Read/Write Multiple registers that can be downloaded from http://www.modbus.org

Below there is a screenshot from that document. In our case we need to read 11 words and write 11 words. The first word read and the first word written are reserved and not used in our program.

Here is an example of a request to read six registers starting at register 4, and to write three registers starting at register 15:

Request		Response	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	17	Function	17
Read Starting Address Hi	00	Byte Count	0C
Read Starting Address Lo	03	Read Registers value Hi	00
Quantity to Read Hi	00	Read Registers value Lo	FE
Quantity to Read Lo	06	Read Registers value Hi	0A
Write Starting Address Hi	00	Read Registers value Lo	CD
Write Starting address Lo	0E	Read Registers value Hi	00
Quantity to Write Hi	00	Read Registers value Lo	01
Quantity to Write Lo	03	Read Registers value Hi	00
Write Byte Count	06	Read Registers value Lo	03
Write Registers Value Hi	00	Read Registers value Hi	00
Write Registers Value Lo	FF	Read Registers value Lo	0D
Write Registers Value Hi	00	Read Registers value Hi	00
Write Registers Value Lo	FF	Read Registers value Lo	FF
Write Registers Value Hi	00		
Write Registers Value Lo	FF		

In our example:

Emis_Param[1] = 16#0017 where 17 hex is Modbus function code 23 dec (Read/Write) and 00 hex is the Read Starting Address Hi byte

Emis_Param[2] = 16#0000 where 00 hex (low byte) is Read Starting Address Lo byte and 00 hex (high byte) is Quantity to read Hi byte

Emis_Param[3] = 16#000B where 0B hex (11dec) is Quantity to read Lo. and 00 hex is Write Starting Address Hi

Emis_Param[4] = 16#0000 00 hex (low byte) is Write Starting Address Lo byte and 00 hex (high byte) is Quantity to write Hi byte

Emis_Param[5] = 16#160B where 0B hex is Quantity to write Hi byte, 11 word in dec and 16hex is the Write Byte Count, 22 bytes in dec

Emis_Param[6] = value of the first word to be written to the drive – reserved, do not use

Emis_Param[7] = Control Word (CMD). The bytes are swapped, that is why ROR_INT is used, it swaps the bytes

Emis_Param[8]= Speed Ref (LFRD). The bytes are swapped, that is why ROR_INT is used, it swaps the bytes

Emis_Param[9] to Emis_Param[18] = additional parameters that can be configured in the drive and written from the program. The byte will also have to be swapped.

Note:
The IO Scanner table has the read starting address and write starting address as zero.

- **GEST**: Manage Para, array of 4 INT for the management parameters.
  We only write to the fourth word (length) the number of bytes to be sent when we trigger the DATA_EXCH. In our case there are 32 bytes (16 words). When the message is received from the drive, it will overwrite this field with the number of received words (different number, 24 bytes in our case). This is why we need to write this word every time we execute DATA_EXCH.
  Writing the third word (timeout) is optional.

The following table describes the structure of the data in the communication management table:

	Order of the Word	Most Significant Byte	Least Significant Byte
Data managed by the system	1	Exchange number	Activity bit
	2	operation report	communication report
Data managed by the user	3	Timeout	
	4	Length	

- **RECP**: Receive Data, array of 12 INT for the received data.
  Receive_DATA[0] = Low byte is the function code, 17hex, 23 dec in our case
  High byte is the number of bytes received, 22 dec in our case
  Receive_DATA[1] = First read word from the drive, reserved, do not use
  Receive_DATA[2] = Second read word from the drive, ETA Status word, bytes are swapped, that is why ROR_INT is used
  Receive_DATA[3] = Third read word from the drive, RFRD actual speed word, bytes are swapped, that is why ROR_INT is used
  Receive_DATA[4] to Receive_DATA[12] = additional parameters that can be configure in the drive and read from the program. The byte will also have to be swapped.

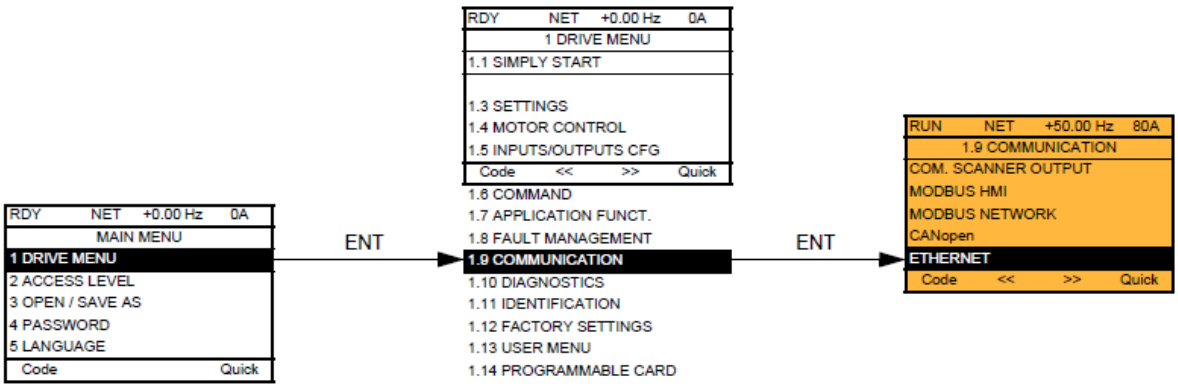
This completes the setting on the Unity / M340 side. The next steps are for configuring the ATV61 / ATV71.

10. Using the graphic display terminal set the IP address of the drive.

The [ETHERNET] submenu is used to configure and display the Ethernet card parameters and can be accessed via the [1.9 - COMMUNICATION] menu.

If you are using the FDR (Faulty Device Replacement) function, you must also configure the device name in the [7. DISPLAY CONFIG.] menu, [7.1 USER PARAMETERS] submenu, [DEVICE NAME] submenu.

This menu is only accessible in expert mode: In the [2 ACCESS LEVEL] (LAC-) menu, set the level to [expert] (EPr).



11. Reserving control (optional step)

It is strongly recommended that control should be reserved for a single master device

To configure this reservation, enter an IP address other than [0.0.0.0] (0) (0) (0) (0) in the [1.9 COMMUNICATION] (COM-) menu, [ETHERNET] (EtH-) submenu, [IP Master] submenu.

If control has been reserved: only the control word (CMd) written by the master with control will be accepted via IO Scanning or via Modbus TCP messaging.

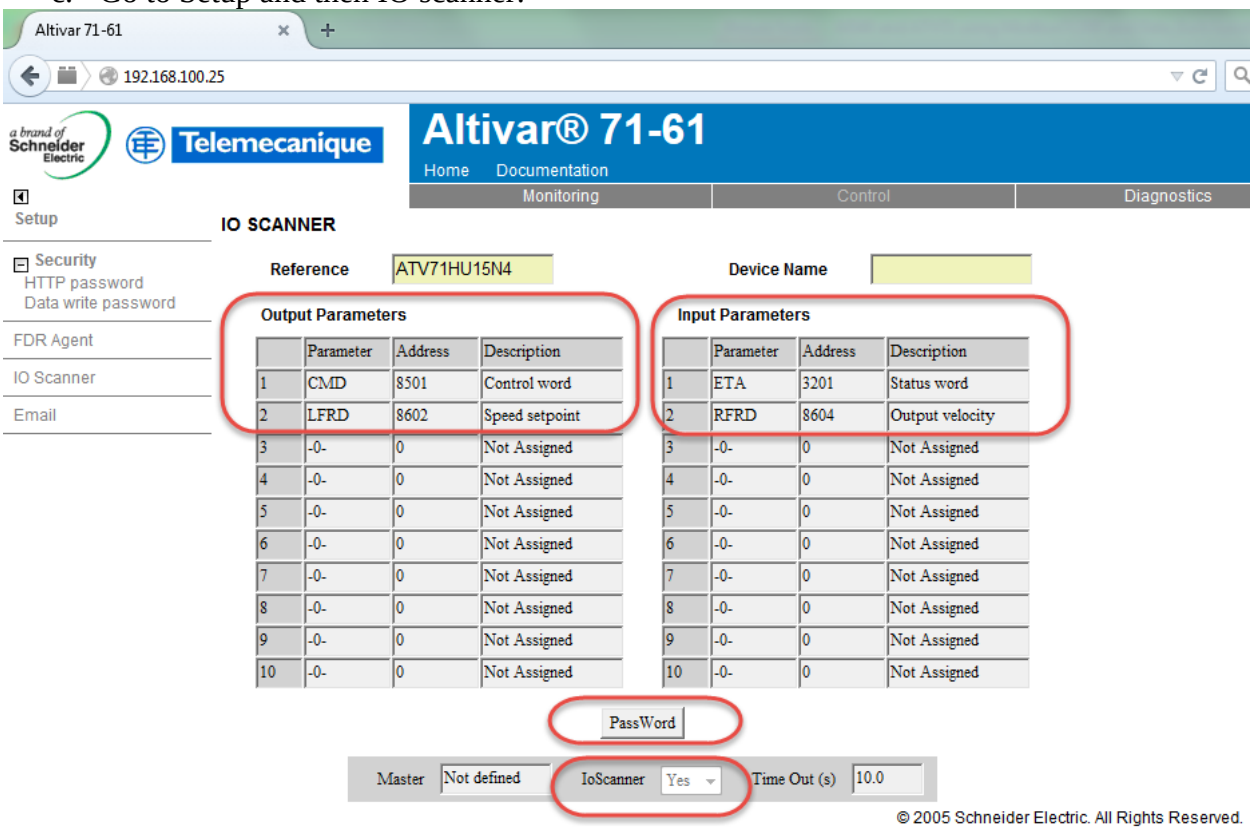
If control has not been reserved ([IP Master] = [0.0.0.0] (0) (0) (0) (0)), control can come from any IP address.

12. Configure the IO scanner variables.

There are two ways of doing that: by using the integrated web server (detailed in this document) or by using the Powersuite or SoMove software, free to download from www.schneider-electric.com. In both cases the IO scanner has to be turned off to be able to change these variables.

a. To connect to the Web server of a drive located, for example, at IP address 192.168.100.25 enter the URL **192.168.100.25** in the address bar of a Web browser.
b. When the browser first connects to the drive Web server, it will ask for a user name and password. By default, the user name and the password are both "USER" (upper case).

c. Go to Setup and then IO scanner:



	Parameter	Address	Description
1	CMD	8501	Control word
2	LFRD	8602	Speed setpoint
3	-0-	0	Not Assigned
4	-0-	0	Not Assigned
5	-0-	0	Not Assigned
6	-0-	0	Not Assigned
7	-0-	0	Not Assigned
8	-0-	0	Not Assigned
9	-0-	0	Not Assigned
10	-0-	0	Not Assigned

	Parameter	Address	Description
1	ETA	3201	Status word
2	RFRD	8604	Output velocity
3	-0-	0	Not Assigned
4	-0-	0	Not Assigned
5	-0-	0	Not Assigned
6	-0-	0	Not Assigned
7	-0-	0	Not Assigned
8	-0-	0	Not Assigned
9	-0-	0	Not Assigned
10	-0-	0	Not Assigned

d. To change these variables, click on PassWord and enter “USER”, change the IOScanner to “No”, add, change or delete your parameters, click on “Save” and then turn the IO Scanner back to “Yes”.

In our example:
Output Parameter CMD will correspond to Emis_Param[7] and LFRD to Emis_Param[8].
Input parameters, ETA will correspond Receive_DATA[2] and RFRD to Receive_DATA[3].

This completes the step by step procedure on how to control an ATV61 / ATV71 with a M340 CPU with built-in Ethernet port using the DATA_EXCH function.