

Protocol Macro  
MODBUS for  
CS1/CJ1 series

# ModbusPM

# Operation Manual

OMRON

## Table of contents

<b>1. SPECIFICATIONS .....</b>	<b>3</b>
<b>2. MODBUSPM MASTER INTERFACE.....</b>	<b>4</b>
2.1. PRINCIPE.....	4
2.2. LADDER PROGRAM .....	5
2.3. READ 1 BIT (FUNCTION 01 OR 02).....	6
2.4. READ N WORDS (FUNCTION 03 OR 04) .....	6
2.5. WRITE 1 BIT (FUNCTION 05) .....	7
2.6. WRITE 1 BIT TO N SLAVES .....	7
2.7. WRITE 1 WORD (FUNCTION 06).....	8
2.8. WRITE 1 WORD TO N SLAVES .....	8
2.9. WRITE N WORDS TO 1 SLAVE (FUNCTION 10).....	9
2.10. WRITE N WORDS TO N SLAVES .....	9
<b>3. MODBUSPM SLAVE INTERFACE.....</b>	<b>10</b>
3.1. PRINCIPE.....	10
3.2. OPERATION.....	11
3.3. PMSU CONFIGURATION.....	11
<b>4. DOWNLOADING THE PROTOCOL MACRO.....</b>	<b>12</b>
4.1. PROGRAM ADAPTATIONS.....	12
4.2. FLAGS AND CONTROL BITS OF PMSU .....	14

## 1. Specifications

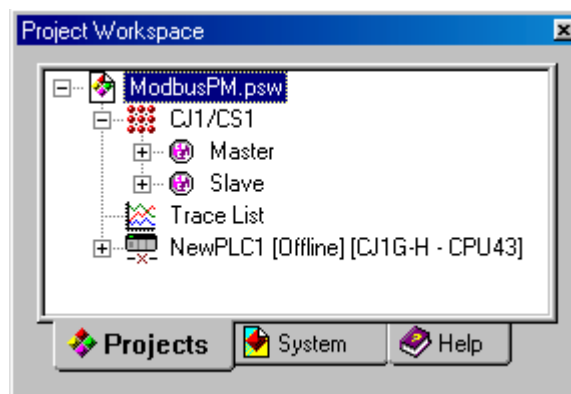
ModbusPM is a protocol macro developed for CJ1/CS1 series providing master and slave Modbus interfaces.

The following functions codes are implemented :

- 01 : Read n bit (output)
- 02 : Read n bit (input)
- 03 : Read n words (register)
- 04 : Read n words (input)
- 05 : Write 1 bit
- 06 : Write 1 word
- 10 : Write n words

Write command (05-06-10) can be sent to N consecutive slave

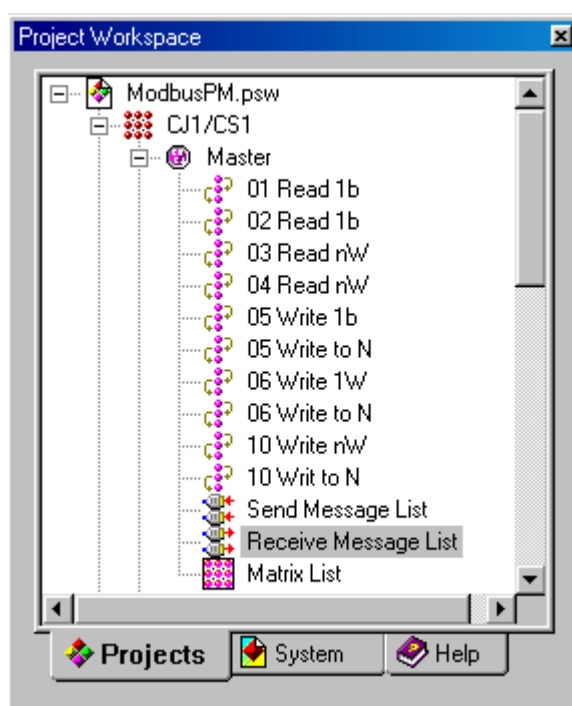
The Slave interface needs a ladder program (ModbusPM.cxp) to prepare data according to the read/write master request. The Read/Write command are executed in the DM area



## 2. ModbusPM Master interface

### 2.1. Principe

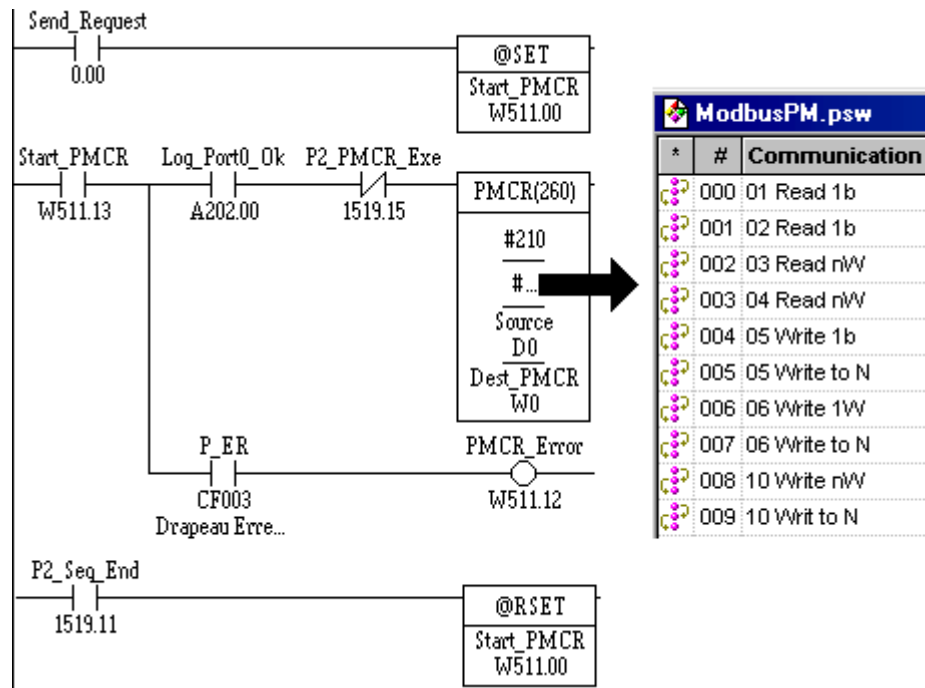
ModbusPM Master interface provide 10 sequences of read/write command.  
Write to **N** commands can be sent to several slaves which should be consecutive.




Command should be specified through operand C2 of PMCR instruction.

sequence	Designation	Operation
000	01 Read 1b	Read 1 bit
001	02 Read 1b	Read 1 bit
002	03 Read nW	Read n mots
003	04 Read nW	Read n words
004	05 Write 1b	Write 1 bit
005	05 Write to N	Write 1bit to N slaves
006	06 Write 1W	Write 1 word
007	06 Write to N	Write 1 word to N slaves
008	10 Write nW	Write n words
009	10 Writ to N	Write n words to N slaves

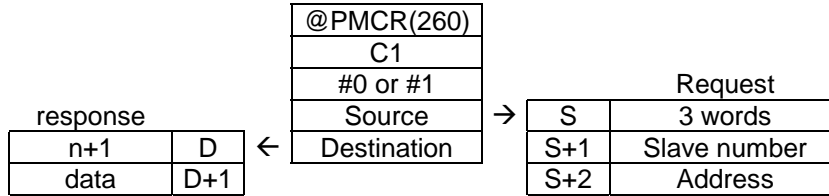
## 2.2. Ladder program



In case of no response or wrong response, the sequence is cancelled and Sequence End Flag does not change.

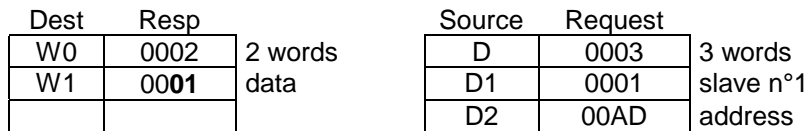
 If you change the sequence number, be sure to use hexadecimal format in PMCR instruction.

**2.3. Read 1 bit (function 01 or 02)**

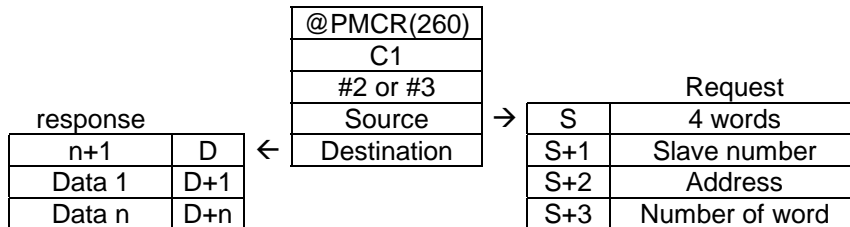


Example :Read bit 0 from slave n°1 address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request : 01 01 00AD 0001 6C2B  
 Response : 01 01 01 01 9048

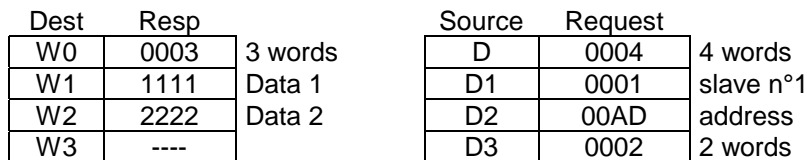


**2.4. Read n words (function 03 or 04)**

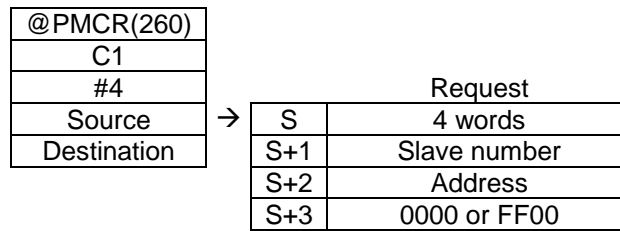


Example :Read 2 words from slave n°1 address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request : 01 03 00AD 0002 55EA  
 Response : 01 03 0004 1111 2222 8E87



### 2.5. Write 1 bit (function 05)



Example: Write to ON bit 00AD.00 of slave n°1.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

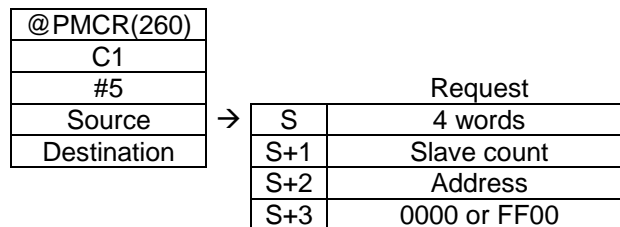
Request : 01 05 00AD FF00 1DDB  
 Response : 01 05 00AD FF00 1DDB (Response similar to the request)

Source	
D0	0004
D1	0001
D2	00AD
D3	FF00

4 words  
 slave n°1  
 Address  
 Write data

Write data : FF00= ON 0000= OFF

### 2.6. Write 1 bit to N slaves



Example: write to ON bit 00AD.00 of slaves n°1,2 and 3.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

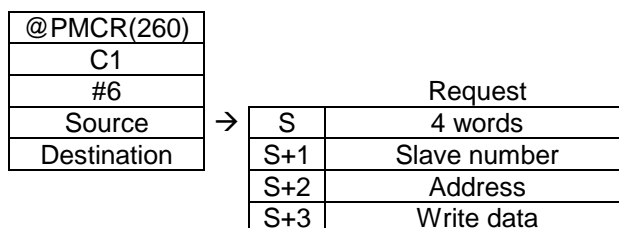
Request : 01 05 00AD FF00 1DDB then 02 05 00AD FF00 1DE8 then 03....  
 Response : 01 05 00AD FF00 1DDB (Response similar to the request)

Source	
D0	0004
D1	0003
D2	00AD
D3	FF00

4 words  
 3 slaves  
 Address  
 Write data

Each response will be written to Destination area. If one of the slaves does not respond, sequence will be aborted. Repeat counter value will designate the slave in error.

### 2.7. Write 1 word (function 06)



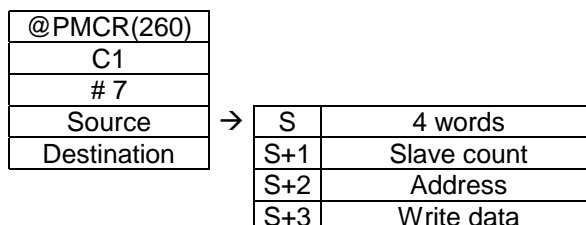
Example: Write 1111 in to slave n°1 at address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request : 01 06 00AD 1111 D477  
 Response : 01 06 00AD 1111 D477 (Response similar to the request)

Source	
D0	0004
D1	0001
D2	00AD
D3	1111

4 words  
 slave n°1  
 Address  
 Write data

### 2.8. Write 1 word to N slaves



Example :Write 1111 in to slaves n°1, 2 and 3 at address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request :01 06 00AD 1111 D477  
 Response :01 06 00AD 1111 D477 (Response similar to the request)

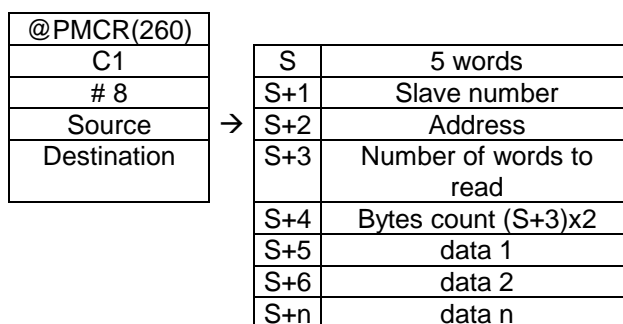
Source	
D0	0004
D1	0003
D2	00AD
D3	1111

4 words  
 3 slaves  
 Address  
 Write data

Each response will be written to Destination area. If one of the slaves does not respond, sequence will be aborted. Repeat counter value will designate the slave in error.

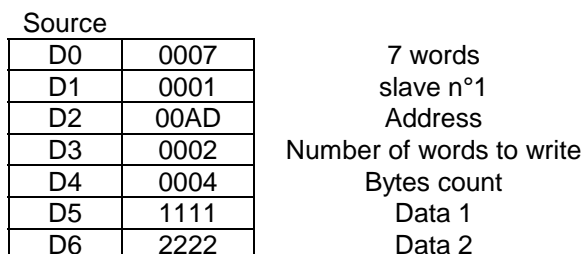


### 2.9. Write n words to 1 slave (function 10)

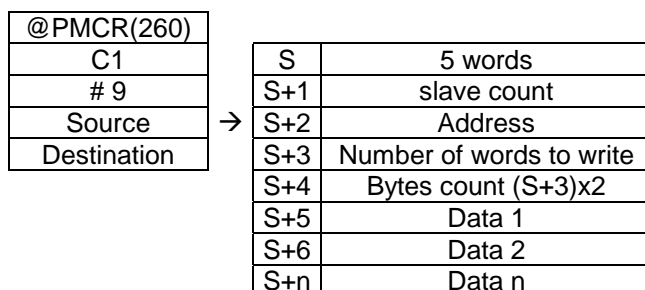


Example : Write 1111 and 2222 in to slave n°1 at address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request : 01 10 00AD 0002 04 1111 2222 F5CE  
 Response : 01 10 00AD 0002 D029

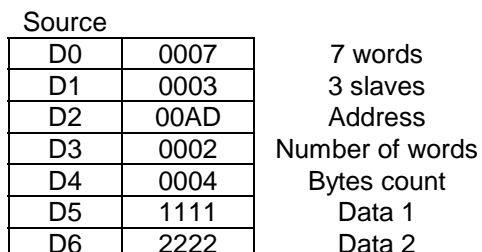


### 2.10. Write n words to N slaves



Example : write 1111 et 2222 in to slaves n°1,2 and 3 at address 00AD.  
 Logical port n°0, Port RS232C n°2, Unit CJ1W-SCU41 n° 0 (SCU +10)

Request : 01 10 00AD 0002 04 1111 2222 F5CE  
 Response : 01 10 00AD 0002 D029



### 3. ModbusPM slave interface

#### 3.1. Principe

ModbusPM slave provide 2 sequences :

- Read & Write  
 Functions 03-04-06-10h are implemented with the following restriction :  
**The write n word commands (10h) checksum is not checked.**
- Write with CRC  
 This sequence allow only one function: write n word command (10h) but CRC is controlled.

The raison of such division is that sequence Read & Write use a matrix to compare the requests to identify the function then execute the appropriate step. Matrix case works only in these cases :

- 1- similar Header and terminator.
- 2- header and size information in the request.
- 3- header and same size.

Modbus protocol does not use terminator, and most function do not use bytes count information. The third case could be used but request such Write n word are bigger than the others. This is why function 10h is treated in two steps. The frame is truncated by the matrix first and the rest of the frame is read by another step which cannot use CRC control because one part of the frame is missing.

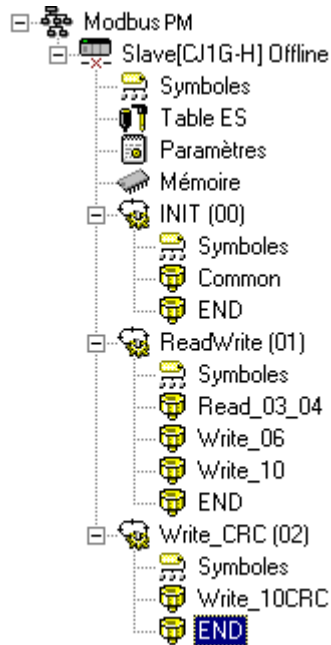
Receive Message	Data
03_Read_nW	<a>+[03]+(W(1),4)+<c>
04_Read_nW	<a>+[04]+(W(1),4)+<c>
06_Writ_1W	<a>+[06]+(W(1),4)+<c>
10_Writ_nW	<a>+[10]+(W(1),6)
10_Wr_Data	~(W(4),1)+(W(5),R(DM 29602))
10_Wrt_CRC	<a>+[10]+(W(1),4)+<l>+(W(4),*)+<c>

The write request is truncated after 8 bytes and compared in the matrix: <a>+[10]+(W(1),6))

This part contains address, number of words, bytes count but also the 1<sup>st</sup> byte of the 1<sup>st</sup> data to be written. The ladder program recovers this bytes and place it into the MSB od word W4. The rest of the frame is written starting from LSB of W4 ( "~" before (W(4),1) ).

### 3.2. Operation

Unlike the master interface, Slave interface should be assisted by a ladder program to prepare data from/to the address specified by the request. Read/Write is executed into the DM area starting from 00000 to DM 32767 (7FFF) with a maximum of 96 words (60h) by command.



Common task store the slave number to HR1 which is used by C3 of PMCR. The other tasks transfer the data to the address specified by the master request. When data are ready, the "Wait Release" flag is activated by the task to allow the PMSU to continue the execution of the sequence. Only one task should be activated. Use the property windows to de-activate the task (01 or 02) according to the sequence used Read & Write or Write with CRC

Procedure :

1. Adapt the ladder program ModbusPM.cxp to the PMSU used (see Ch. flags and control bits).
2. Activate the Task 01 or 02.
3. Specify the slave number in the common task.

### 3.3. PMSU configuration.

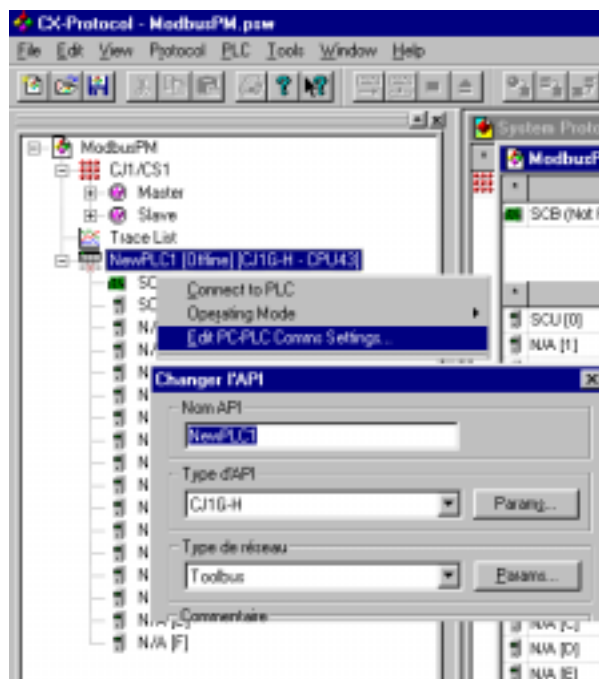
Reference	Serial Communications Boards		Serial Communications Unit	
	CS1W-SCB21	CS1W-SCB41	CS1W-SCU21	CJ1W-SCU41
Port 1	RS232C	RS232C	RS232C	RS-422A/485
Port 2	RS232C	RS-422A/485	RS232C	RS232C
API compatibles	CS1H-CPU67/CPU66/CPU65/CPU64/CPU63 CS1G-CPU45/CPU44/CPU43/CPU42		CJ1G-CPU45 CPU44	

Communication port and Unit number (for SCU unit) determine the content of C1 of PMCR instruction.

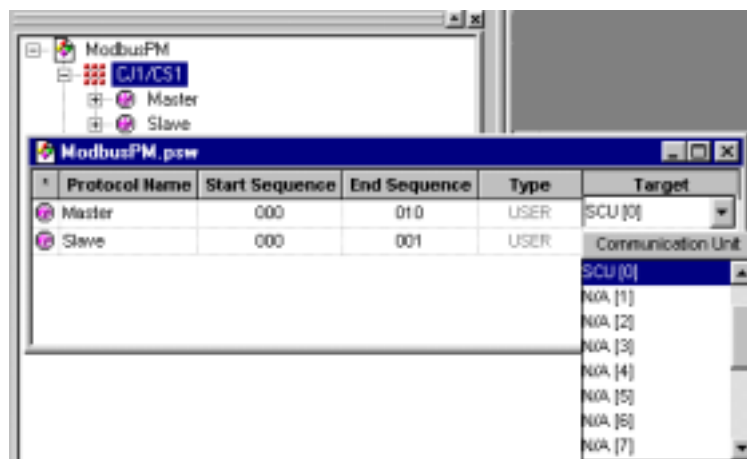
## 4. Downloading the protocol macro

### 4.1. program adaptations

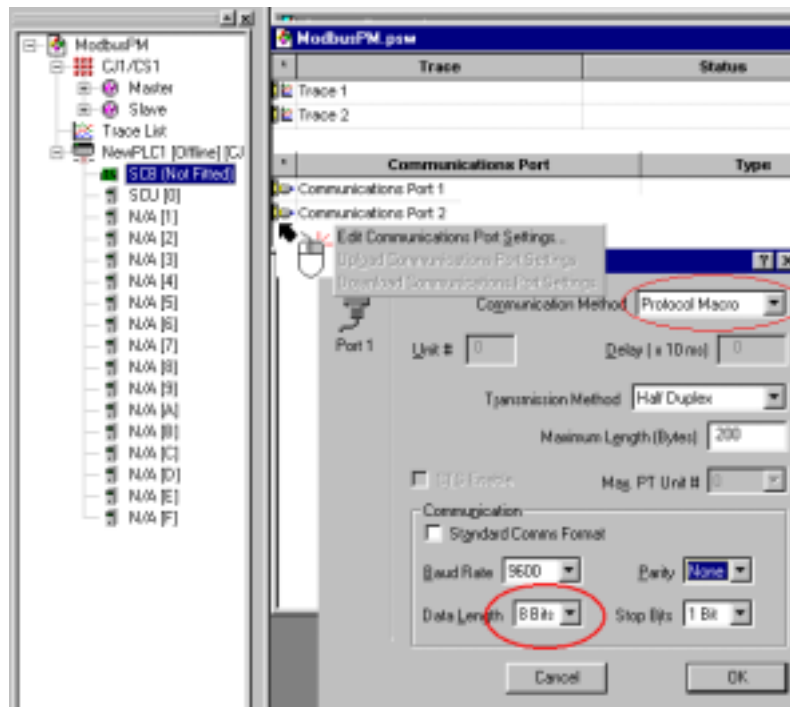
1- Specify the PLC type



2 specify the PMSU type



3 – Edit the communication Port settings (1 or 2) and select Protocol Macro and choose the correct format. (8 bits should be use for Modbus)



4- Connect then transfer the setup and the protocol (mastes or slave) into the PMSU.

## 4.2. Flags and control bits of PMSU

The address of flags and bits used by the ladder program are depending of the PMSU used.

Each unit use 25 words in CIO memory.

For SCB unit,  $n = 1900$

For SCU unit,  $n = \text{CIO } 1500 + (25 \times \text{Unit number})$

Port n°1

Symbol	Channel	Bit	SCB	SCU n°0	SCU n°1
P1_Wait_Release	n	00	1900.00	1500.00	1525.00
P1_Abort_Switch	n	03	1900.03	1500.03	1525.03
P1_Seq_End	n + 9	11	1909.11	1509.11	1534.11
P1_PMCR_Exe	n + 9	15	1909.15	1509.15	1534.15
P1_Case_0	n + 12	15	1912.00	1512.00	1537.00
P1_Case_1	n + 12	01	1912.01	1512.01	1537.01
P1_Case_2	n + 12	02	1912.02	1512.02	1537.02
P1_Case_3	n + 12	03	1912.03	1512.03	1537.03
P1_Step_8	n + 13	08	1913.08	1513.08	1537.08
P1_Step_9	n + 13	09	1913.09	1513.09	1537.09

Port n°2

Symbol	Channel	Bit	SCB	SCU n°0	SCU n°1
P2_Wait_Release	n	08	1900.08	1500.08	1525.08
P2_Abort_Switch	n	11	1900.11	1500.11	1525.11
P2_Seq_End	n + 19	11	1919.11	1519.11	1534.11
P2_PMCR_Exe	n + 19	15	1919.15	1519.15	1534.15
P2_Case_0	n + 22	00	1922.00	1522.00	1537.00
P2_Case_1	n + 22	01	1922.01	1522.01	1537.01
P2_Case_2	n + 22	02	1922.02	1522.02	1537.02
P2_Case_3	n + 22	03	1922.03	1522.03	1537.03
P2_Step_8	n + 23	08	1923.08	1523.08	1537.08
P2_Step_9	n + 23	09	1923.09	1523.09	1537.09