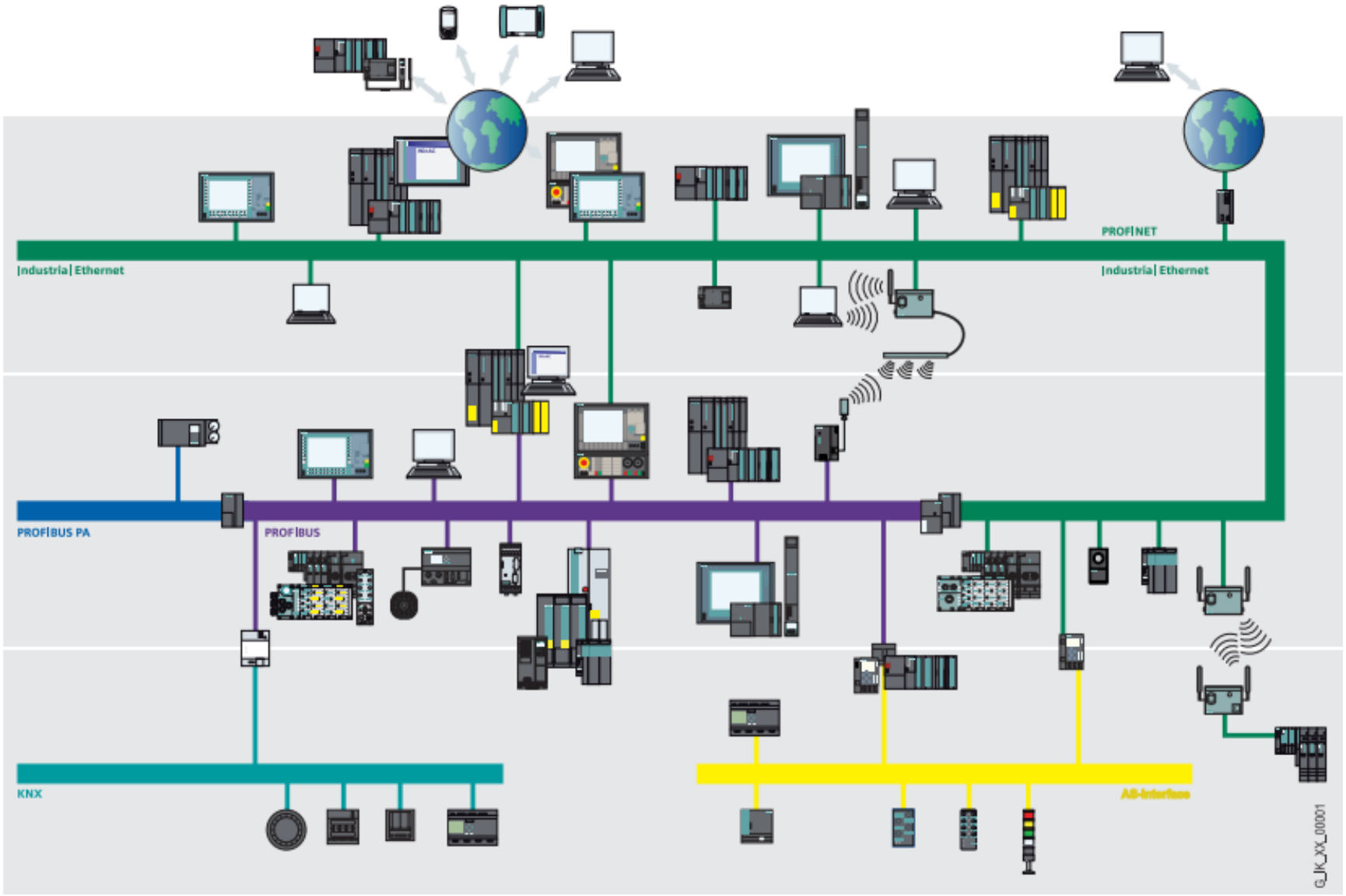
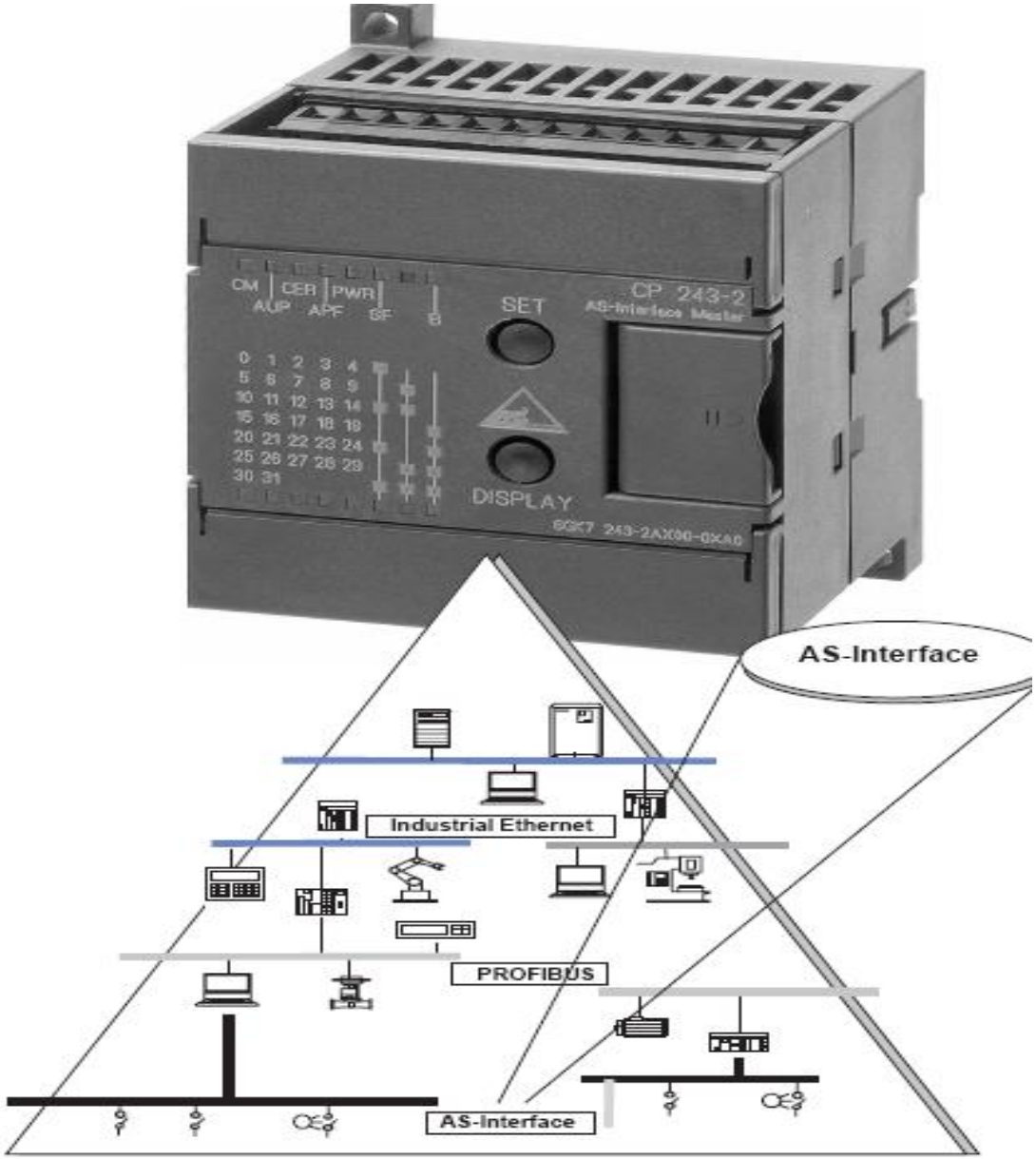
SIEMENS INDUSTRIAL NETWORKS



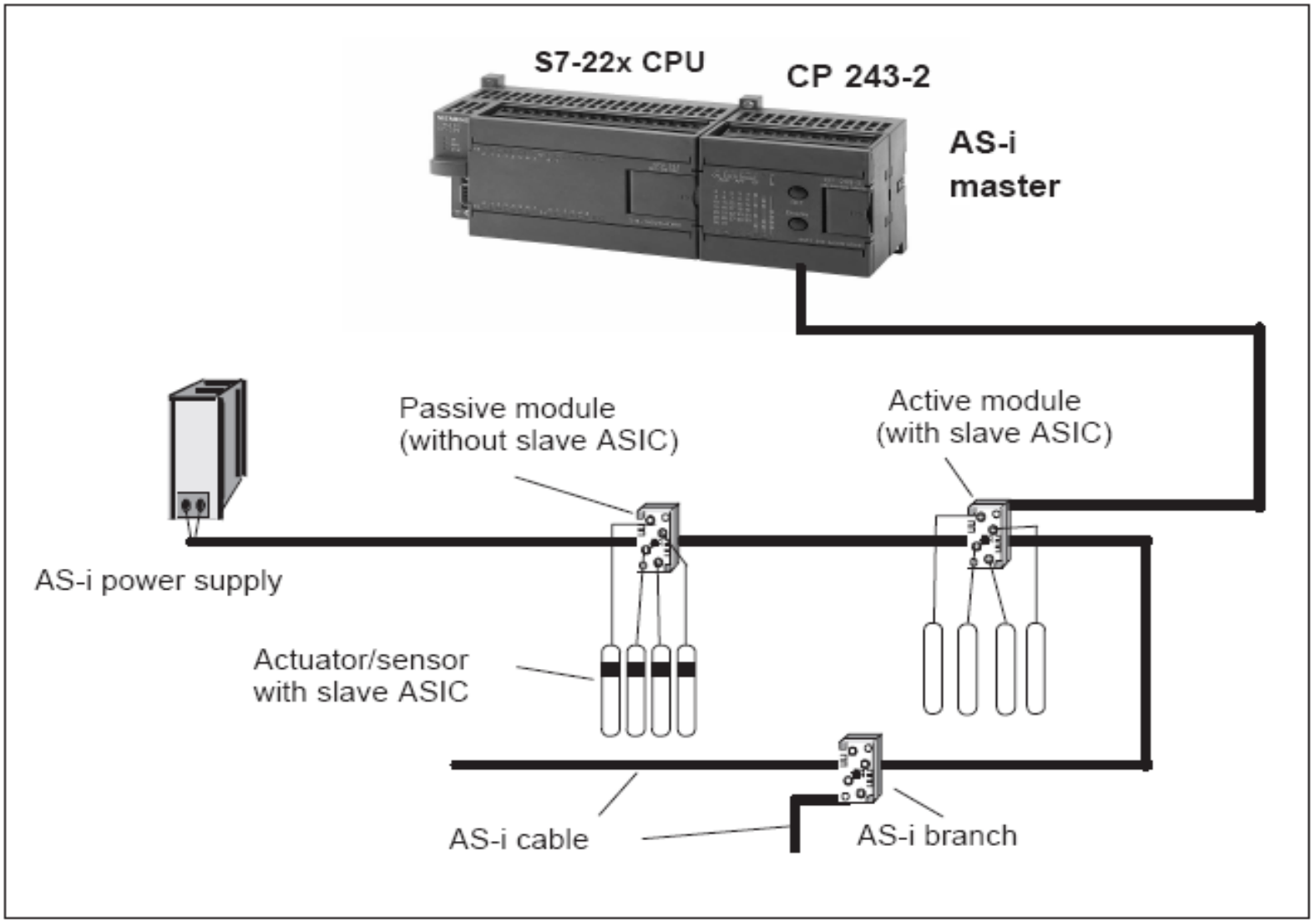
ASI – NETWORK INTRODUCTION

Number of slaves	Up to 62
Number of I/Os	Up to 496 inputs and 496 outputs
Medium	Unshielded two-wire line for data and energy
Line length	100 m as a standard, extendable to 600 m with Repeater and Extension Plug
Cycle time	5 ms (typical) with 31 Slave, 10ms with 62 Slave
Data transfer	Digital and analog (16 bit)

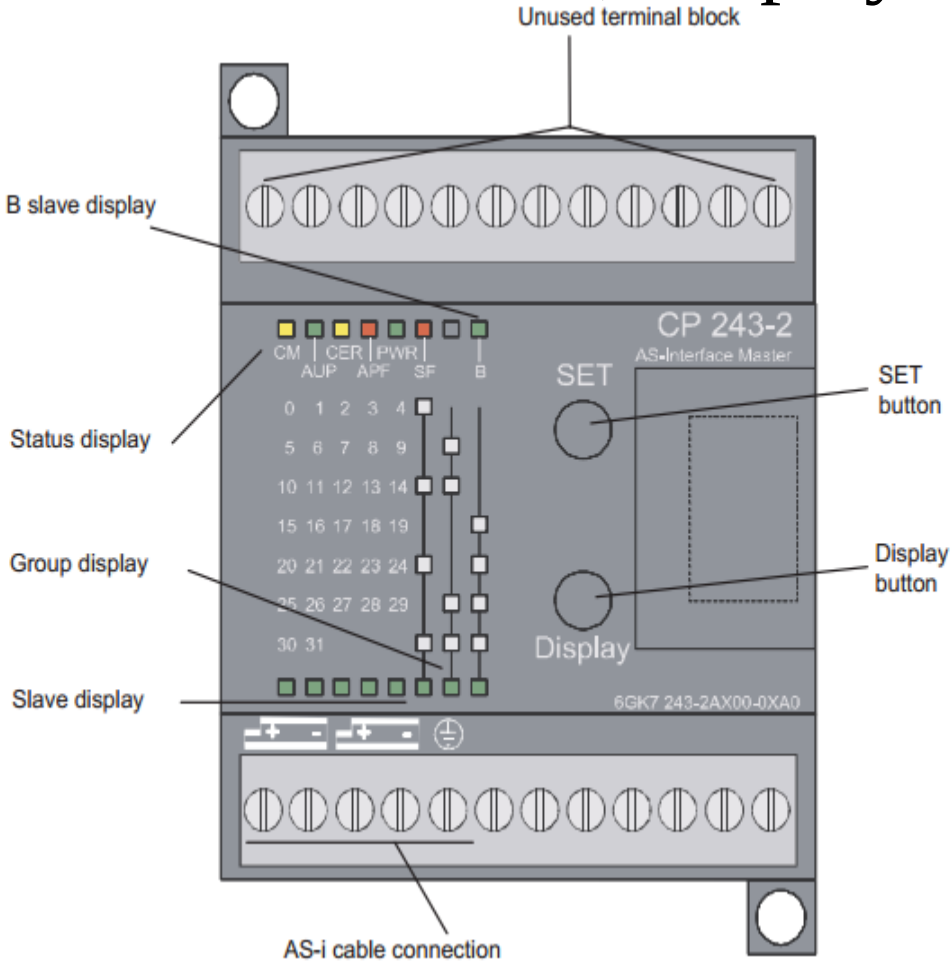
CP 243-2 - AS Interface Master

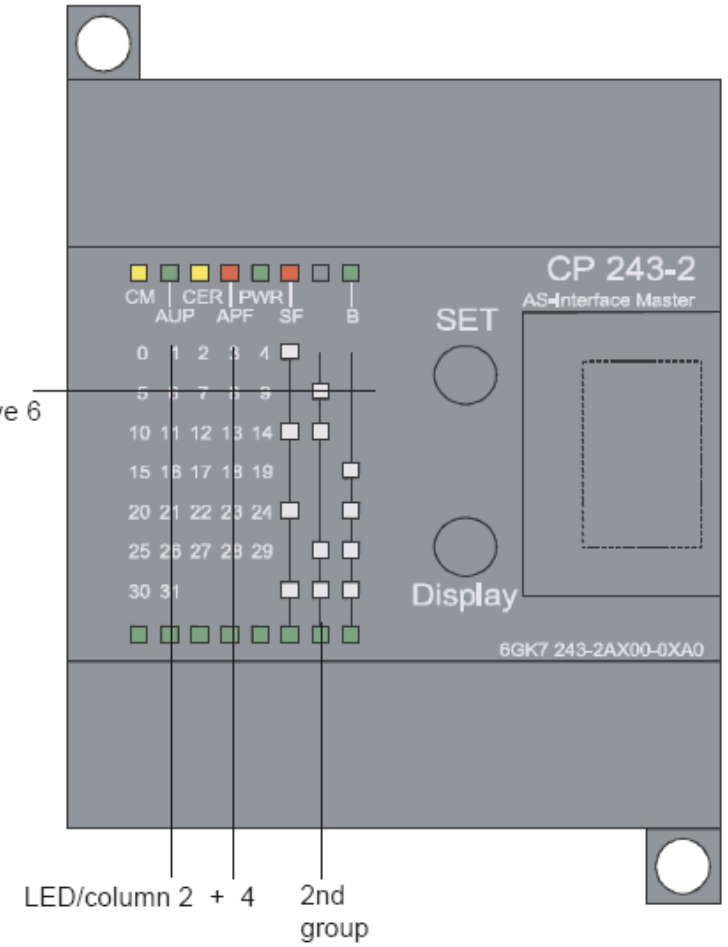


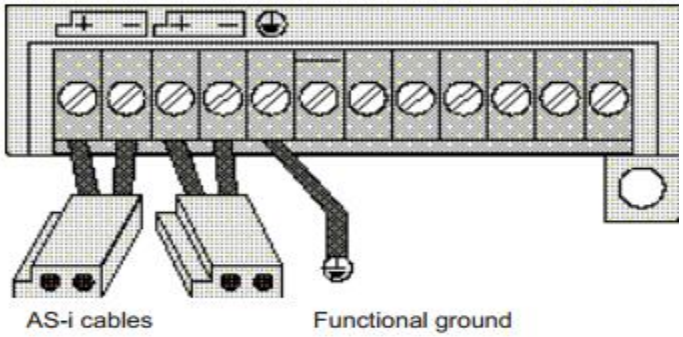
Operation of AS Interface Master in plc s7200



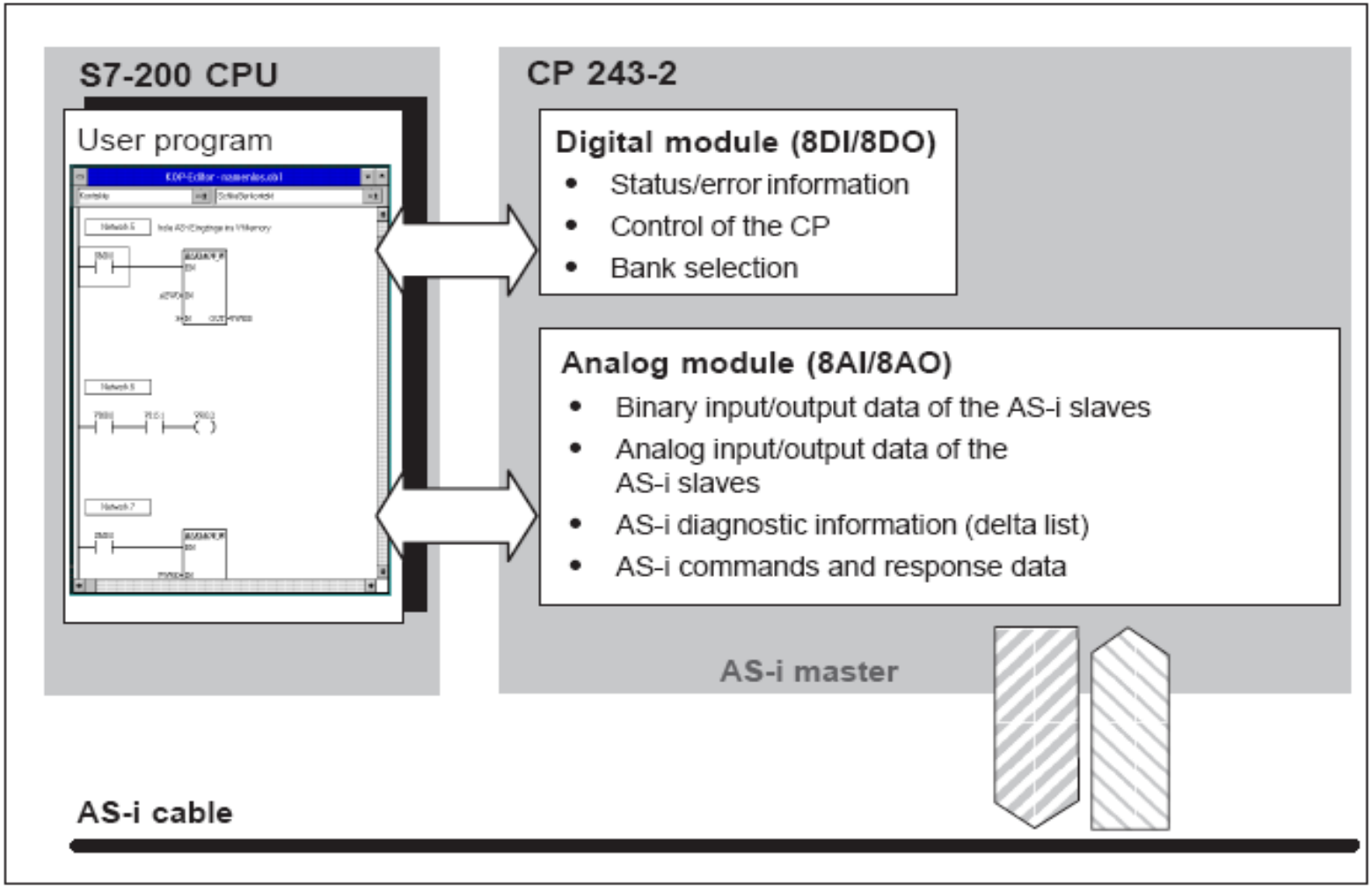
Connection, Display and Control Elements







Interface to user program in the S7200 CPU



ASI Master CP is as expansion modules in the cpu s7200

Addressing the CP in the S7200 CPU

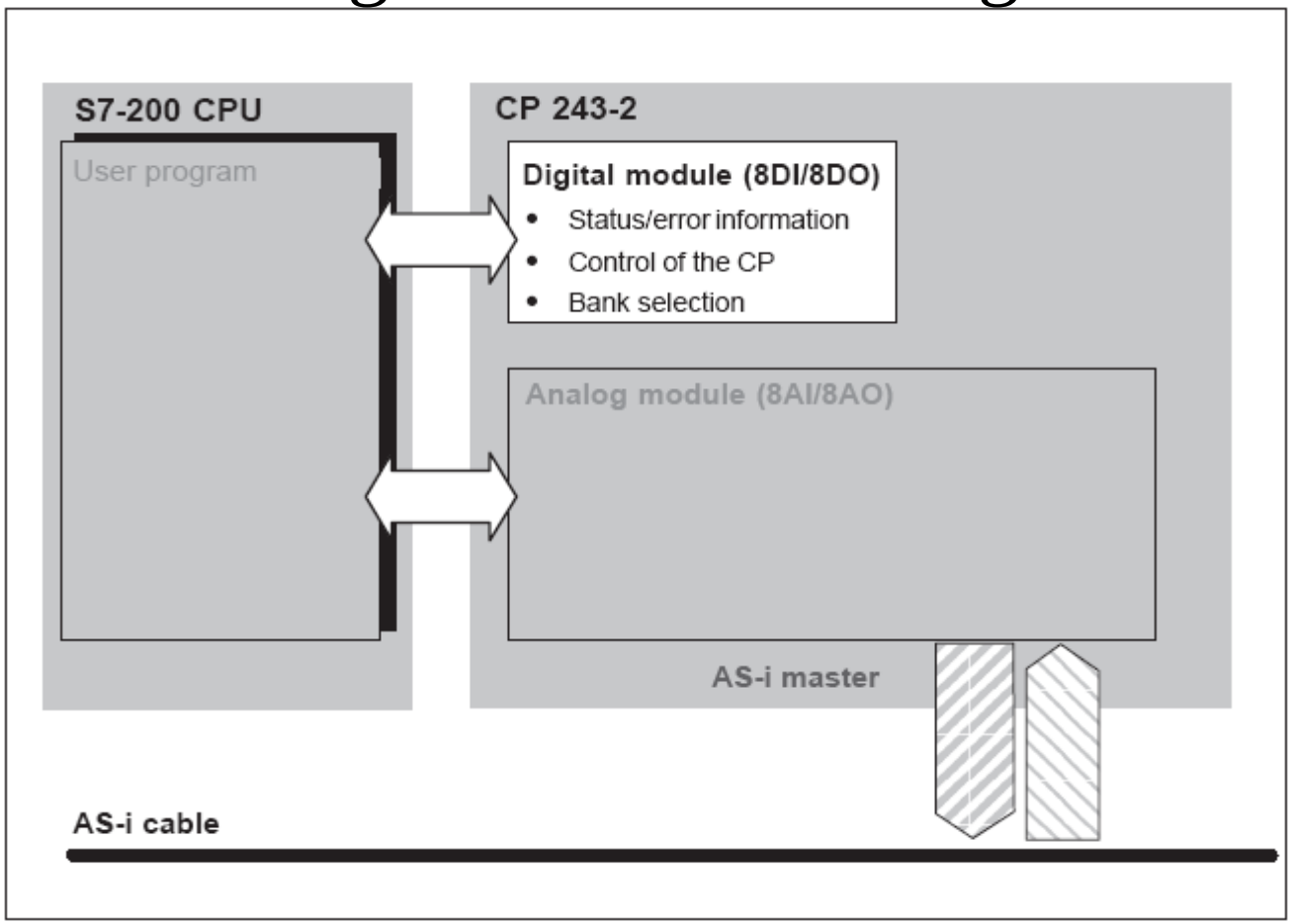
CPU 222		CP 243-2			
8DI	8 DO	8DI	8 DO	8AI	8AO
I0.0	Q0.0	I1.0	Q1.0	AIW0	AQW0
I0.1	Q0.1	I1.1	Q1.1	AIW2	AQW2
I0.2	Q0.2	I1.2	Q1.2	AIW4	AQW4
I0.3	Q0.3	I1.3	Q1.3	AIW6	AQW6
I0.4	Q0.4	I1.4	Q1.4	AIW8	AQW8
I0.5	Q0.5	I1.5	Q1.5	AIW10	AQW10
I0.6	Q0.6	I1.6	Q1.6	AIW12	AQW12
I0.7	Q0.7	I1.7	Q1.7	AIW14	AQW14

CPU 224		CP 243-2			
14 DI	10 DO	8DI	8 DO	8AI	8AO
I0.0	Q0.0	I2.0	Q2.0	AIW0	AQW0
I0.1	Q0.1	I2.1	Q2.1	AIW2	AQW2
I0.2	Q0.2	I2.2	Q2.2	AIW4	AQW4
I0.3	Q0.3	I2.3	Q2.3	AIW6	AQW6
I0.4	Q0.4	I2.4	Q2.4	AIW8	AQW8
I0.5	Q0.5	I2.5	Q2.5	AIW10	AQW10
I0.6	Q0.6	I2.6	Q2.6	AIW12	AQW12
I0.7	Q0.7	I2.7	Q2.7	AIW14	AQW14
I1.0	Q1.0				
I1.1	Q1.1				
I1.2					
I1.3					
I1.4					
I1.5					

Addressing the CP in the S7200 CPU

CPU 224		CP 243-2				CP 243-2			
14 DI	10 DO	8DI	8 DO	8AI	8AO	8DI	8 DO	8AI	8AO
I0.0	Q0.0	I2.0	Q2.0	AIW0	AQW0	I3.0	Q3.0	AIW16	AQW16
I0.1	Q0.1	I2.1	Q2.1	AIW2	AQW2	I3.1	Q3.1	AIW18	AQW18
I0.2	Q0.2	I2.2	Q2.2	AIW4	AQW4	I3.2	Q3.2	AIW20	AQW20
I0.3	Q0.3	I2.3	Q2.3	AIW6	AQW6	I3.3	Q3.3	AIW22	AQW22
I0.4	Q0.4	I2.4	Q2.4	AIW8	AQW8	I3.4	Q3.4	AIW24	AQW24
I0.5	Q0.5	I2.5	Q2.5	AIW10	AQW10	I3.5	Q3.5	AIW26	AQW26
I0.6	Q0.6	I2.6	Q2.6	AIW12	AQW12	I3.6	Q3.6	AIW28	AQW28
I0.7	Q0.7	I2.7	Q2.7	AIW14	AQW14	I3.7	Q3.7	AIW30	AQW30
I1.0	Q1.0								
I1.1	Q1.1								
I1.2									
I1.3									
I1.4									
I1.5									

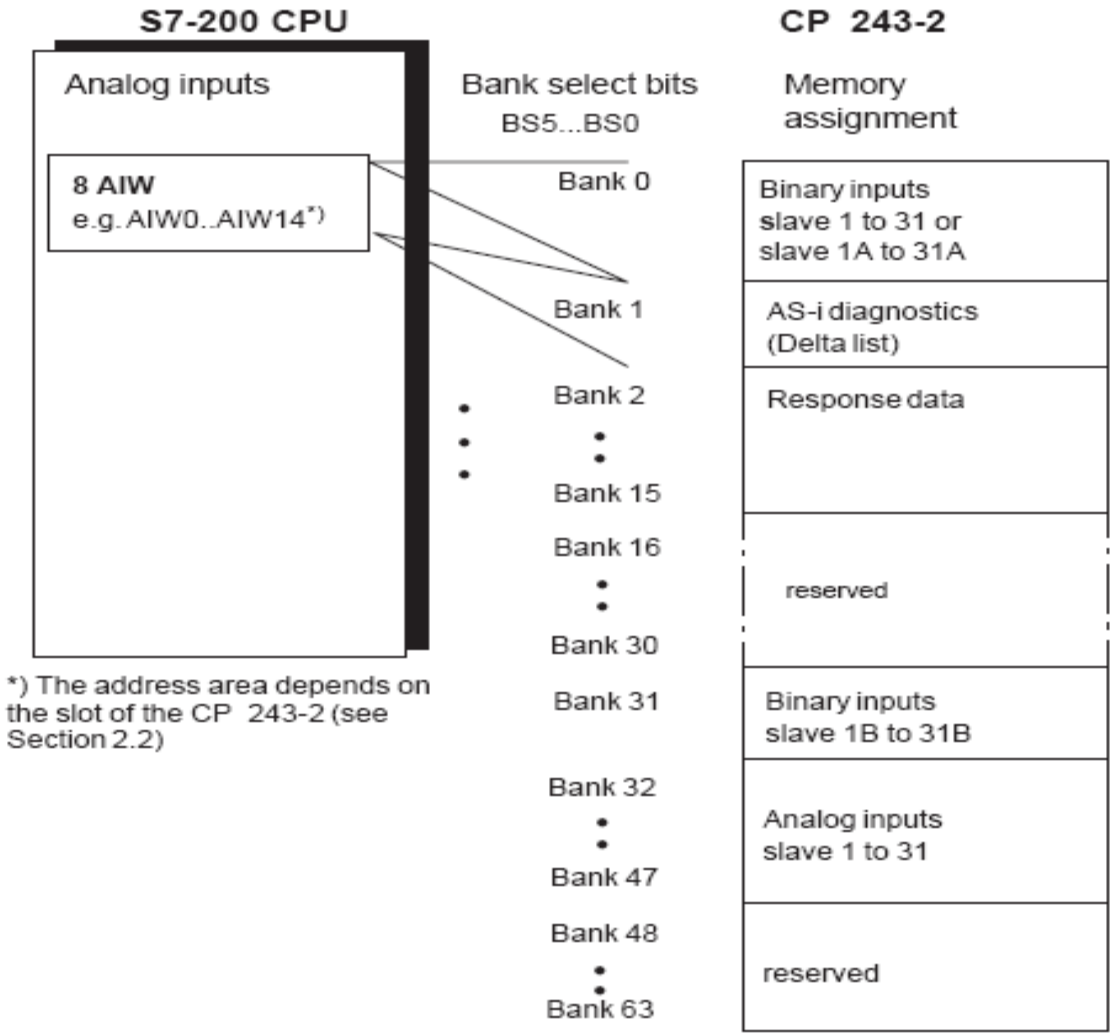
Meaning of Data in the Digital Module



- Error register (8 bit, special memory, SMB8)
- Identification register(8 bit, special memory SMB9)
- Input register 8DI (Status byte of the CP)
- Output register 8DO(Control byte of the CP)

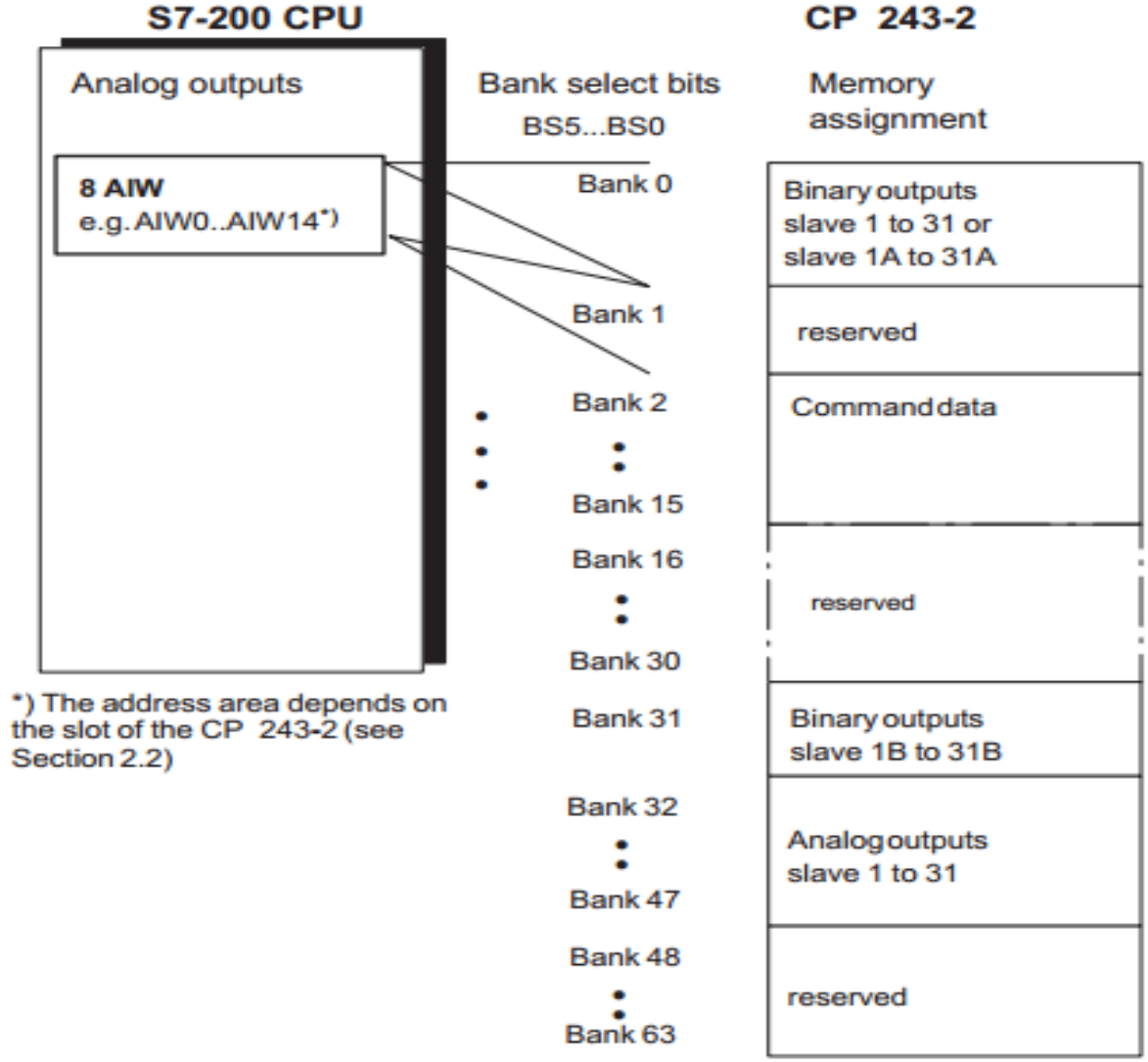
Control Byte (Output register 8DO)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLC_RUN	**ASI_COM**	**BS5**	**BS4**	**BS3**	**BS2**	**BS1**	**BS0**



Control Byte (Output register 8DO)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLC_RUN	ASI_COM	BS5	BS4	BS3	BS2	BS1	BS0



Control byte (Output register 8DO)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLC_RUN	**ASI_COM**	**BS5**	**BS4**	**BS3**	**BS2**	**BS1**	**BS0**

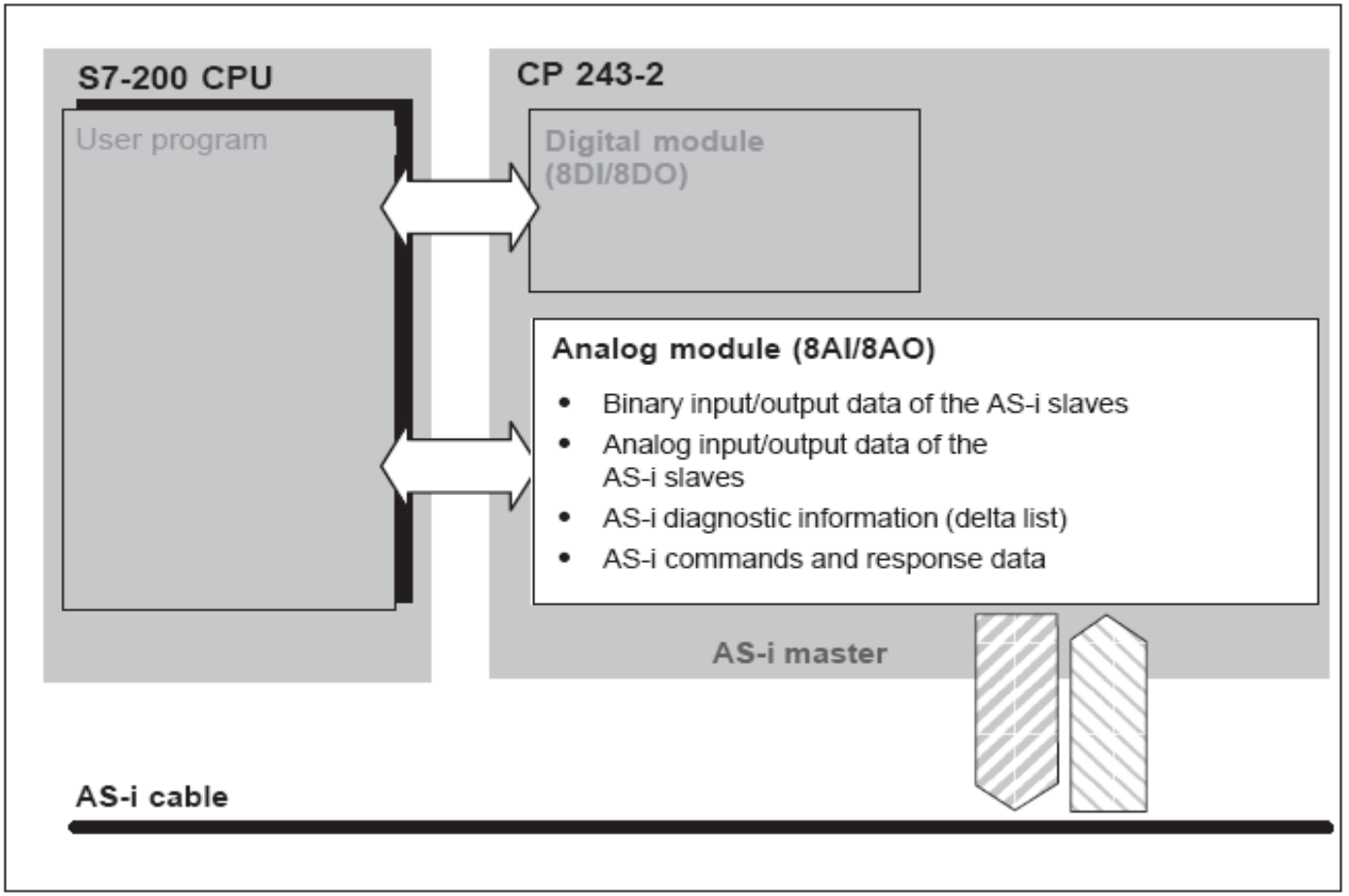
Bit	Value	Meaning
BS0..BS5	0 ... 63 dec.	Bank select bits for changing the bank in the analog module (see Section 2.5).
ASI_COM	0/1	Job bit for the AS-i command interface (see Section 5.1).
PLC_RUN	In the STOP mode of the S7-200 CPU, the CP must send defined values to all AS-i slaves (see Chapter 3). Since the AS-i slave data are transferred via the analog area and the S7-200 CPU does not set this area to '0' when it changes from RUN to STOP, the CPU mode must be signaled to the CP 243-2 using the PLC_RUN bit as follows:	
	0	Signal to the CP 243-2 that the S7-200 CPU is in the STOP mode. The CP 243-2 sends '0' to all AS-i binary slaves. The analog value transfer to analog output slaves is interrupted. The S7-200 CPU sets the bit automatically to "0" at a change from RUN to STOP.
	1	Signals to the CP 243-2 that the S7-200 CPU is in the RUN mode. The CP 243-2 sends the content of output bank 0 to all AS-i slaves (see Section 2.4). The user program must set this bit to "1" during startup (first scan). **Do not set the PLC_RUN bit permanently to "1" with the S7-200 operating system functions such as "CPU configuration/setting the outputs" or "force outputs".**

Status byte (Input register 8DI)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	**ASI_RESP**	0	0	0	0	**CP_READY**	**ASI_MODE**

Bit	Value	Meaning
ASI_MODE	0	The CP 243-2 is in the protected mode.
	1	The CP 243-2 is in the configuration mode.
CP_READY	0	The CP 243-2 is not yet operational after turning on the power supply. Evaluation of the I/O data or other information from the CP is not permitted.
	1	The CP 243-2 is operational.
ASI_RESP	0/1	Response bit for the AS-i command interface (see Section 5.1).

Meaning of Data in Analog Module



- Error register (8 bit, special memory, SMB10)
- Identification register(8 bit, special memory SMB11)
- 8 Analog Input Word (8AIW)
- 8 Analog Output Word (8AQW)

Meaning of Data in Input Analog Module

Assignment of the AS-i Input Data of the Standard or A Slaves (Bank 0 in the Analog Input Area of the CP)

Bank	Byte no.	Bit 7–4	Bit 3–0
0	m+0	reserved	Slave 1 or slave 1A Bit 3 | Bit 2 | Bit 1 | Bit 0
0	m+1	Slave 2 or slave 2A	Slave 3 or slave 3A
0	m+2	Slave 4 or slave 4A	Slave 5 or slave 5A
0	m+3	Slave 6 or slave 6A	Slave 7 or slave 7A
0	m+4	Slave 8 or slave 8A	Slave 9 or slave 9A
0	m+5	Slave 10 or slave 10A	Slave 11 or slave 11A
0	m+6	Slave 12 or slave 12A	Slave 13 or slave 13A
0	m+7	Slave 14 or slave 14A	Slave 15 or slave 15A
0	m+8	Slave 16 or slave 16A	Slave 17 or slave 17A
0	m+9	Slave 18 or slave 18A	Slave 19 or slave 19A
0	m+10	Slave 20 or slave 20A	Slave 21 or slave 21A
0	m+11	Slave 22 or slave 22A	Slave 23 or slave 23A
0	m+12	Slave 24 or slave 24A	Slave 25 or slave 25A
0	m+13	Slave 26 or slave 26A	Slave 27 or slave 27A
0	m+14	Slave 28 or slave 28A	Slave 29 or slave 29A
0	m+15	Slave 30 or slave 30A Bit 3 | Bit 2 | Bit 1 | Bit 0	Slave 31 or slave 31A Bit 3 | Bit 2 | Bit 1 | Bit 0

m = start address of the CP analog module in the input direction

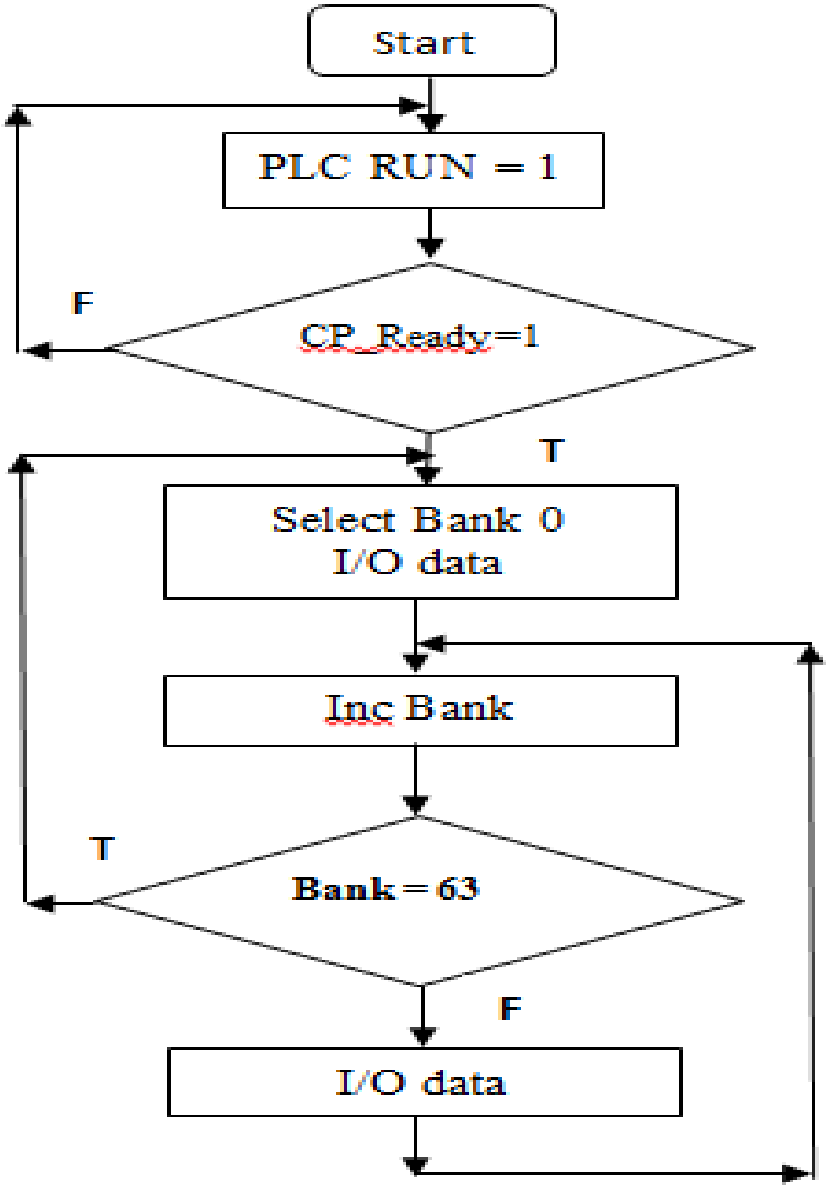
Meaning of Data in Output Analog Module

Assignment of the AS-i Output Data of the Standard or A Slaves (Bank 0 in the Analog Output Area of the CP)

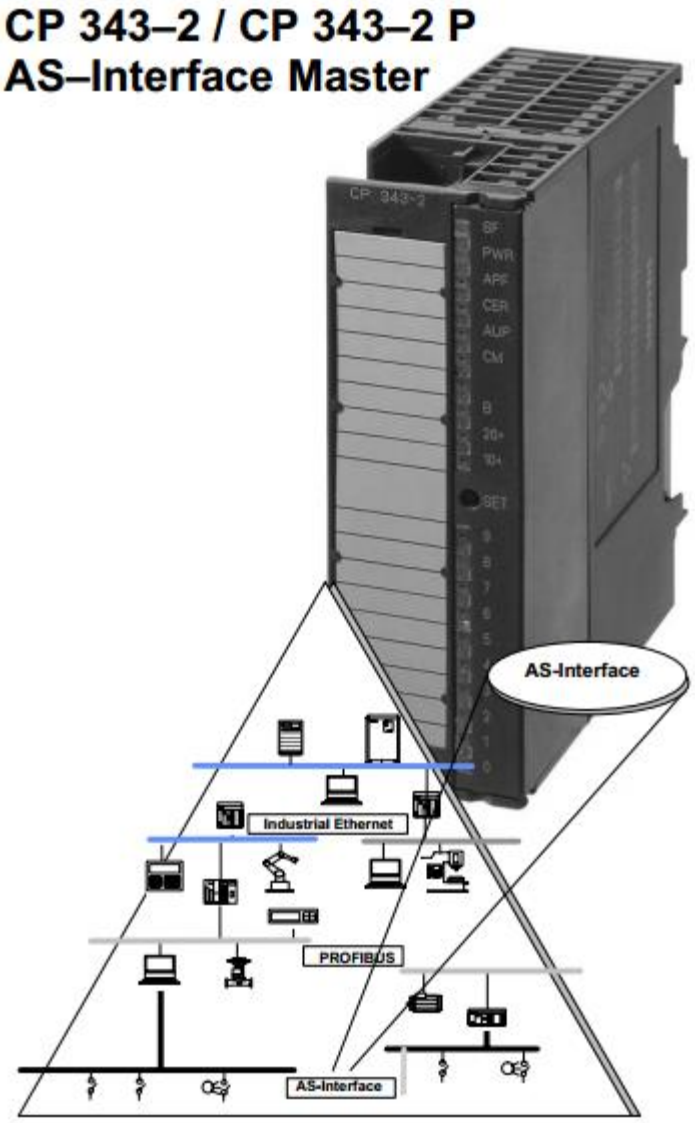
Bank	Byte no.	Bit 7–4	Bit 3–0
0	n+0	reserved	Slave 1 or slave 1A Bit 3 | Bit 2 | Bit 1 | Bit 0
0	n+1	Slave 2 or slave 2A	Slave 3 or slave 3A
0	n+2	Slave 4 or slave 4A	Slave 5 or slave 5A
0	n+3	Slave 6 or slave 6A	Slave 7 or slave 7A
0	n+4	Slave 8 or slave 8A	Slave 9 or slave 9A
0	n+5	Slave 10 or slave 10A	Slave 11 or slave 11A
0	n+6	Slave 12 or slave 12A	Slave 13 or slave 13A
0	n+7	Slave 14 or slave 14A	Slave 15 or slave 15A
0	n+8	Slave 16 or slave 16A	Slave 17 or slave 17A
0	n+9	Slave 18 or slave 18A	Slave 19 or slave 19A
0	n+10	Slave 20 or slave 20A	Slave 21 or slave 21A
0	n+11	Slave 22 or slave 22A	Slave 23 or slave 23A
0	n+12	Slave 24 or slave 24A	Slave 25 or slave 25A
0	n+13	Slave 26 or slave 26A	Slave 27 or slave 27A
0	n+14	Slave 28 or slave 28A	Slave 29 or slave 29A
0	n+15	Slave 30 or slave 30A Bit 3 | Bit 2 | Bit 1 | Bit 0	Slave 31 or slave 31A Bit 3 | Bit 2 | Bit 1 | Bit 0

n = start address of the CP analog module in the output direction

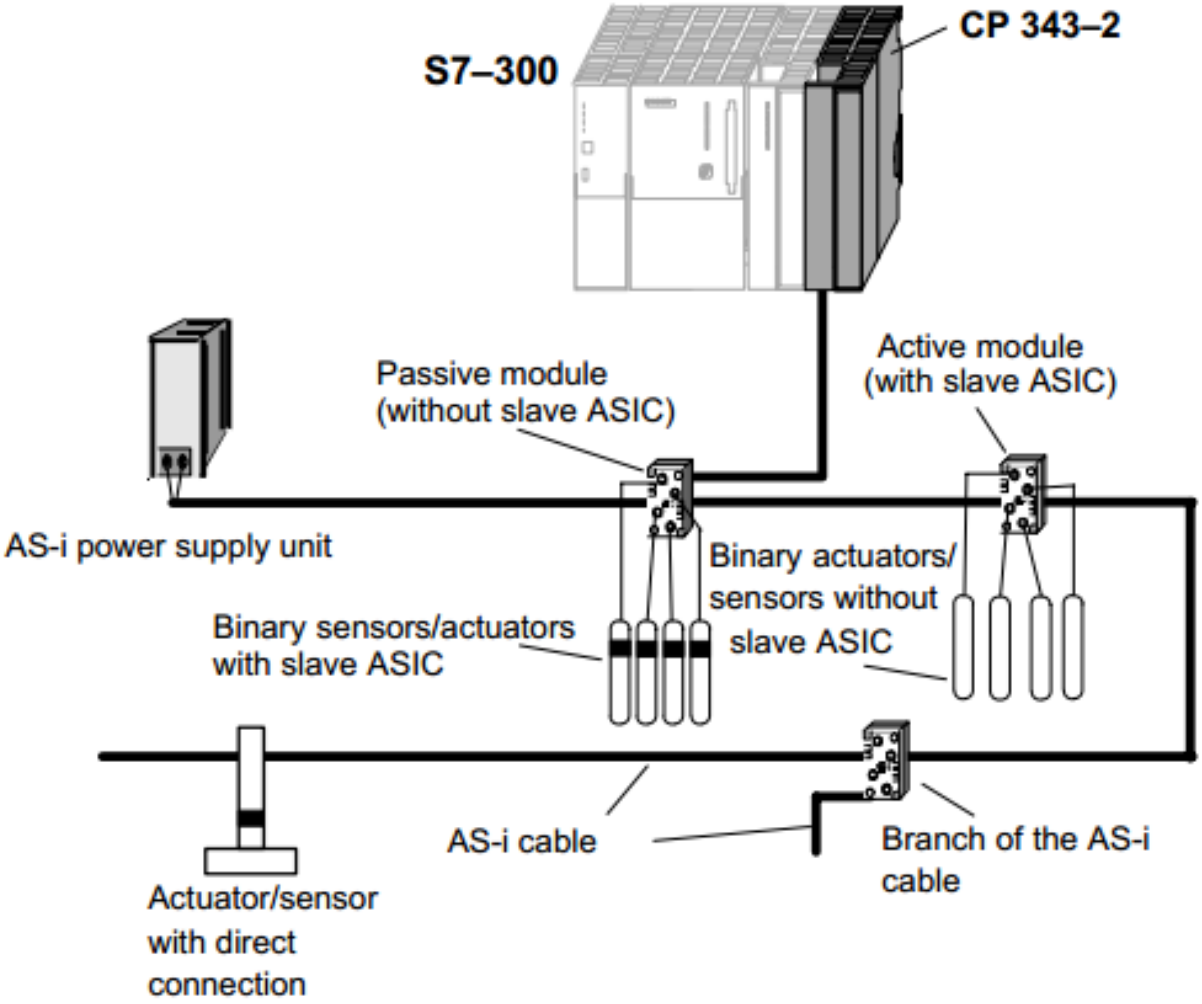
I/O data via ASI Network



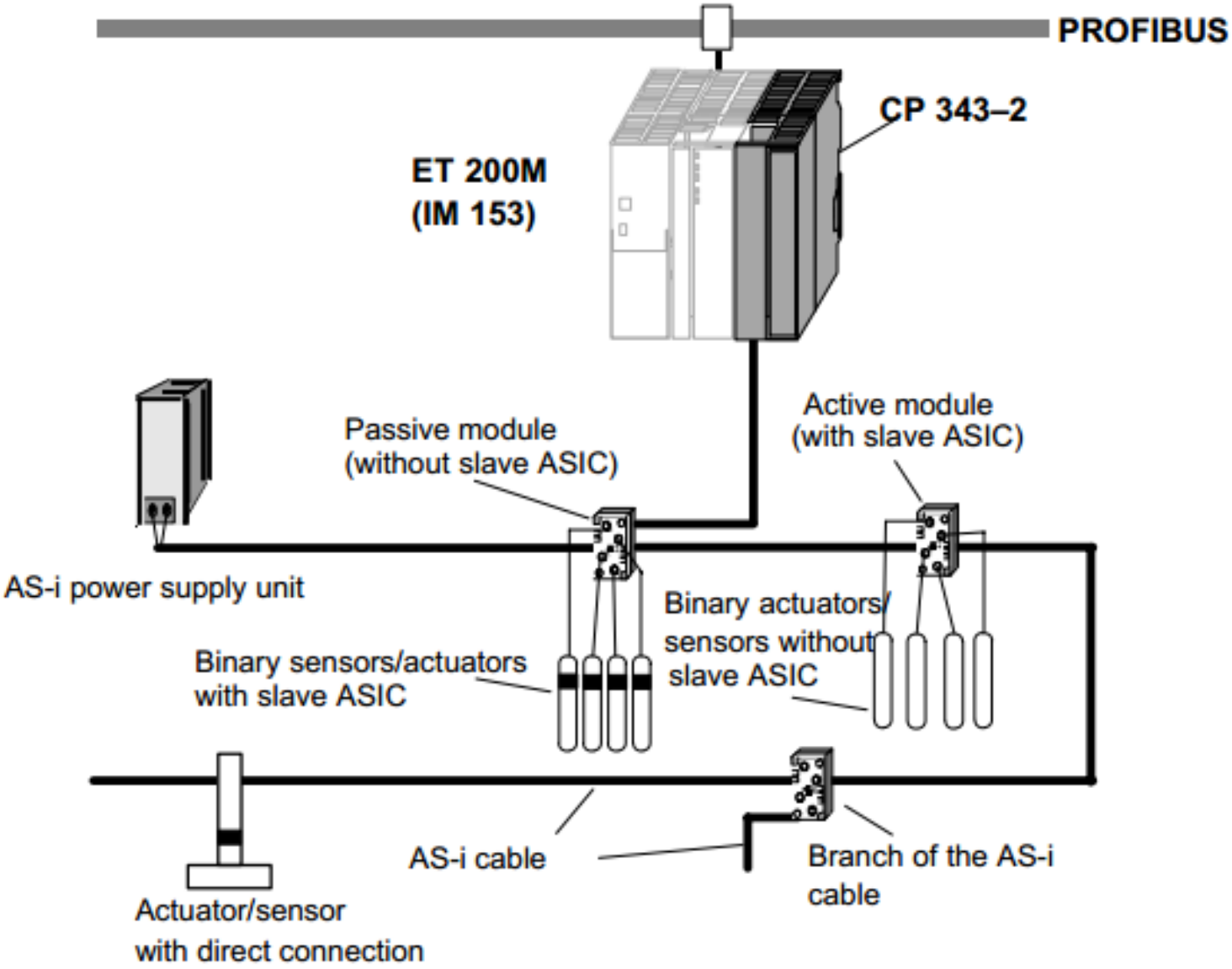
AS Interface Master CP343-2



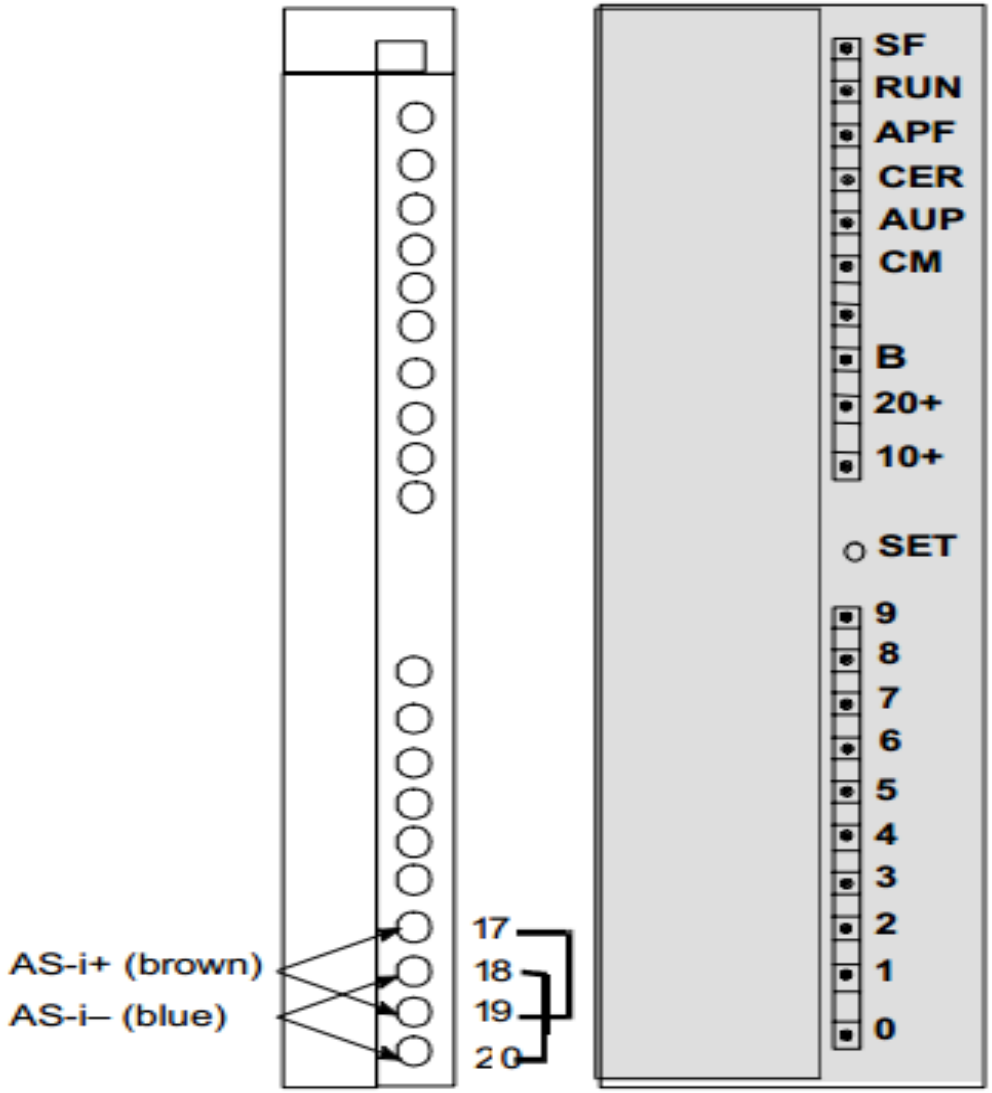
Using the CP Master Module in the S7-300



Distributed use of the CP Master Module in ET200



Displays and Operator controls of the CP 343-2



Addressing the CP on the programmable Controller

Rack 0

Module	PS	CPU	IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number	1	2	3	4	5	6	7	8	9	10	11
Base address	1	2	3	256	272	288	304	320	336	352	368

Rack 1

Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				384	400	416	432	448	464	480	496

Rack 2

Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				512	528	544	560	576	592	608	624

Rack 3

Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				640	656	672	688	704	720	736	752

16 input and output byte in the I/O address space of the S7 programmable controller(Analog area on the plc)

Addressing the CP on the programmable Controller

Rack 0

Module	PS	CPU	IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number	1	2	3	4	5	6	7	8	9	10	11
Base address	1	2	3	256	272	288	304	320	336	352	368

Rack 1

Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				384	400	416	432	448	464	480	496

Rack 2

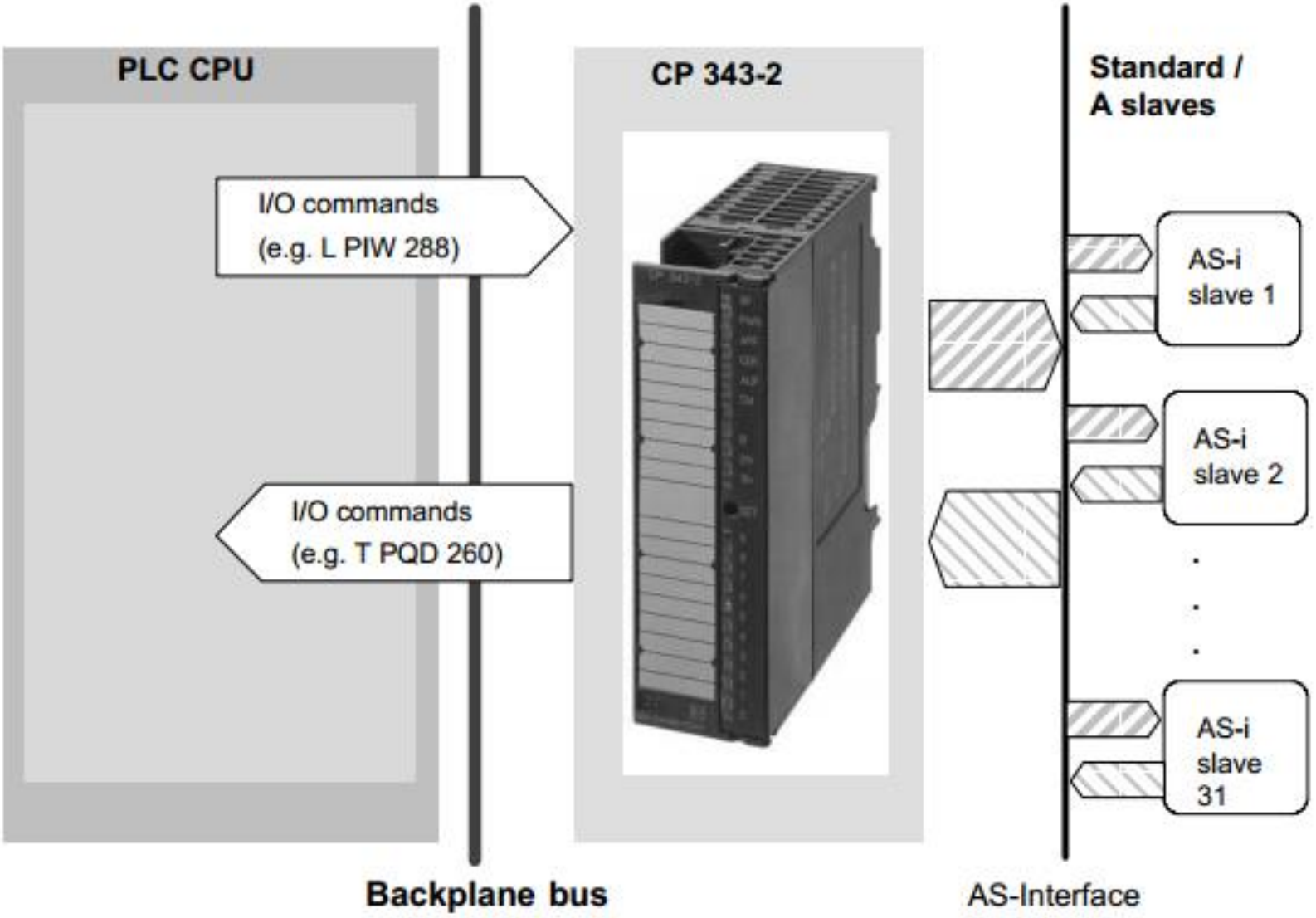
Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				512	528	544	560	576	592	608	624

Rack 3

Module			IM	CP	CP	CP	CP	CP	CP	CP	CP
Slot number			3	4	5	6	7	8	9	10	11
Base address				640	656	672	688	704	720	736	752

16 input and output byte in the I/O address space of the S7 programmable controller(Analog area on the plc)

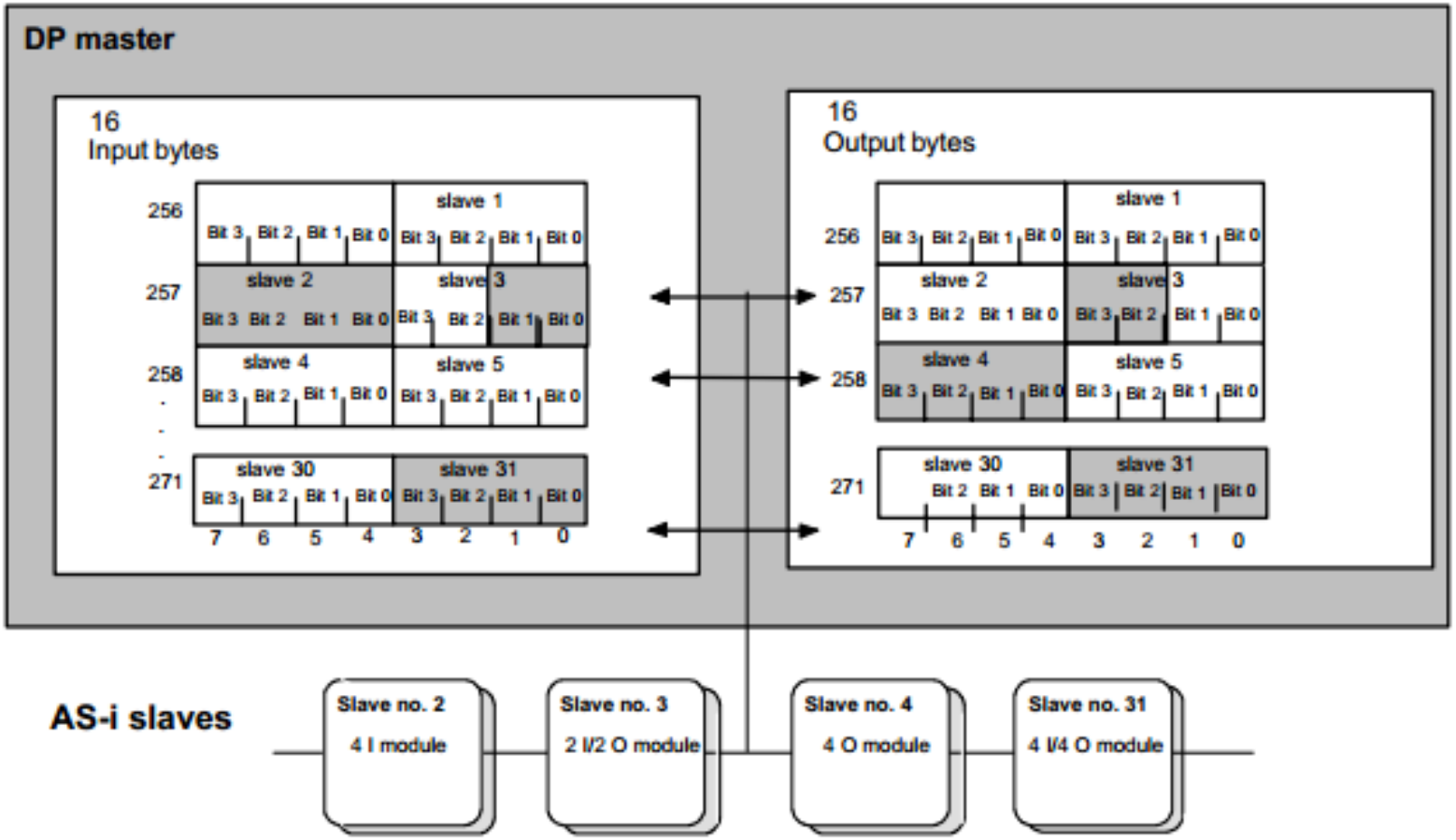
Data exchange between user program and asi slave



Addressing the standard or A slave with the PLC

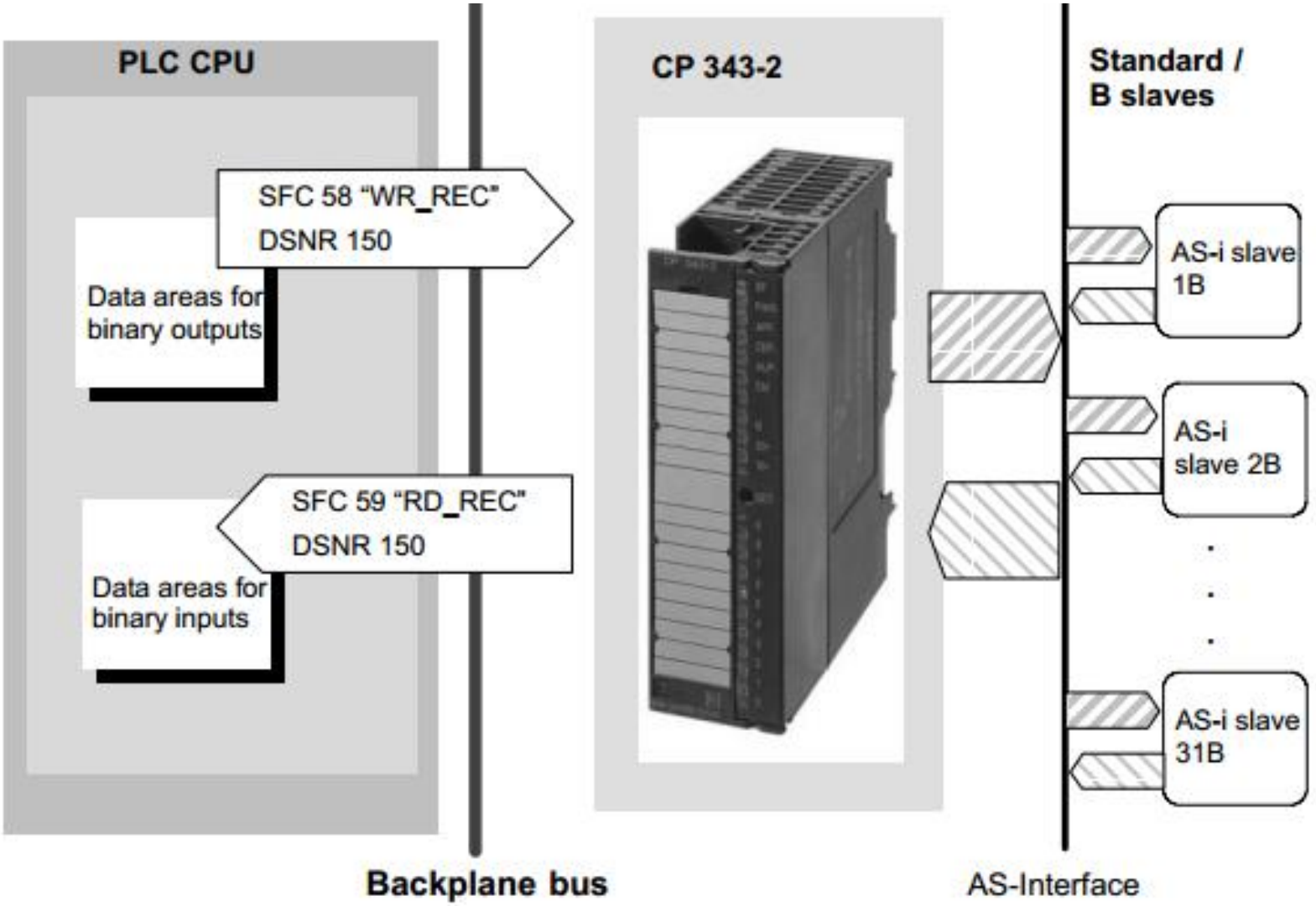
I/O byte number	Bit 7–4	Bit 3–0
n+0	reserved	Slave 1 or 1A Bit 3 | Bit 2 | Bit 1 | Bit 0
n+1	Slave 2 or 2A	Slave 3 or 3A
n+2	Slave 4 or 4A	Slave 5 or 5A
n+3	Slave 6 or 6A	Slave 7 or 7A
n+4	Slave 8 or 8A	Slave 9 or 9A
n+5	Slave 10 or 10A	Slave 11 or 11A
n+6	Slave 12 or 12A	Slave 13 or 13A
n+7	Slave 14 or 14A	Slave 15 or 15A
n+8	Slave 16 or 16A	Slave 17 or 17A
n+9	Slave 18 or 18A	Slave 19 or 19A
n+10	Slave 20 or 20A	Slave 21 or 21A
n+11	Slave 22 or 22A	Slave 23 or 23A
n+12	Slave 24 or 24A	Slave 25 or 25A
n+13	Slave 26 or 26A	Slave 27 or 27A
n+14	Slave 28 or 28A	Slave 29 or 29A
n+15	Slave 30 or 30A Bit 3 | Bit 2 | Bit 1 | Bit 0	Slave 31 or 31A Bit 3 | Bit 2 | Bit 1 | Bit 0

Addressing the standard or A slave with the PLC



Start Address of Analog Module in the PLC: m = 256

Exchanging As-i Binary Value with slave B



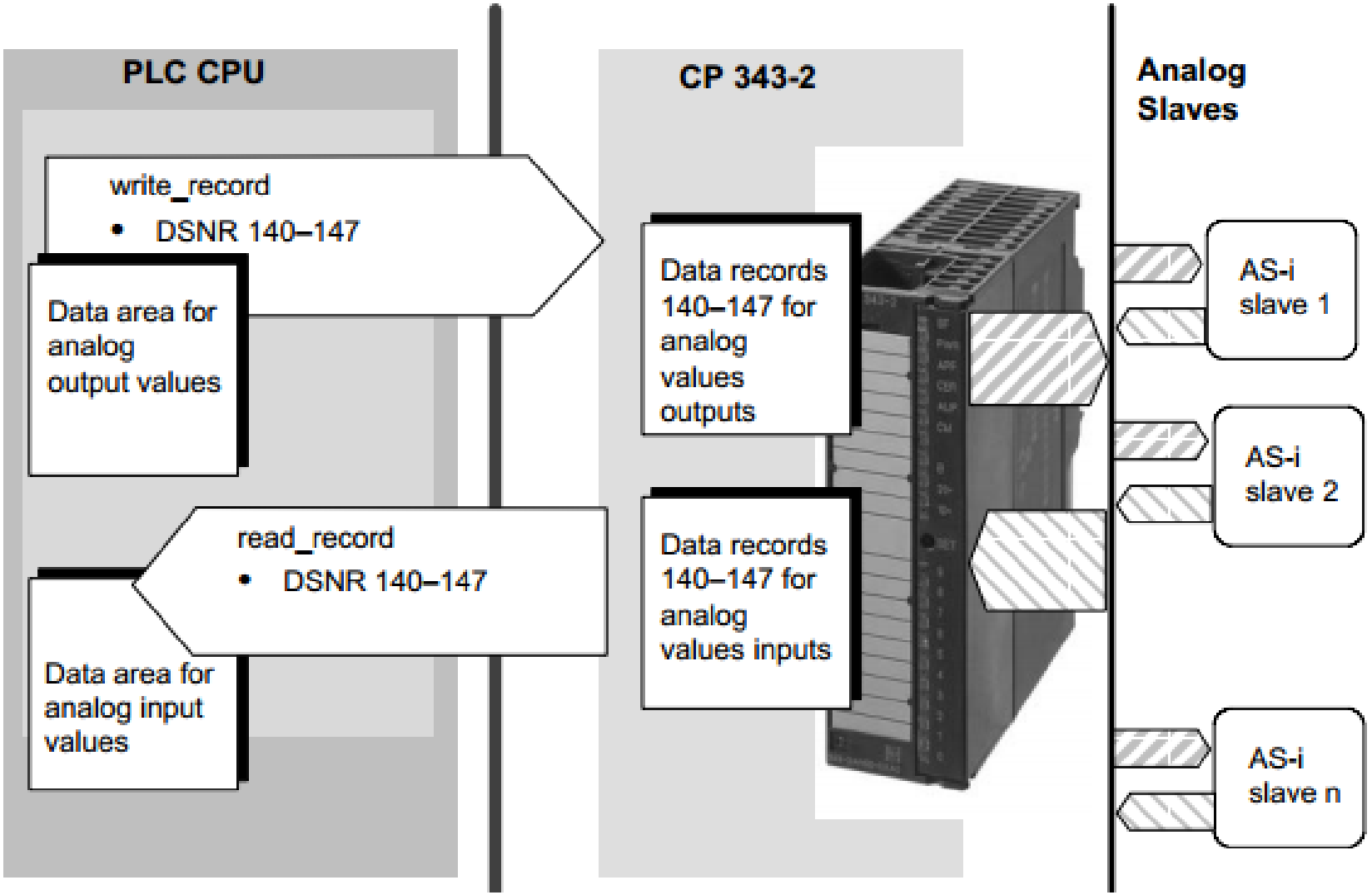
Exchanging As-i Binary Value with slave B

```
STL                                                  Explanation
//Read in binary input data of the B slaves:
        CALL SFC 59                                  //RD_REC
        REQ     :=TRUE                               //Permanent trigger
        IOID    :=B#16#54                            //Fixed value
        LADDR   :=W#16#100                           //CP address (here 256 dec.)
        RECNUM  :=B#16#96                            //DSNR=150 (binary data B slaves)
        RET_VAL :=MW10
        BUSY    :=M9.0
        RECORD  :=P#DB20.DBX16.0 Byte 16             //Dest. area binary data

//Examples: Binary access to B slaves:
        A       DBX 17.4                             //Slave 2B, terminal 1
        S       DBX 49.6                             //Slave 2B, terminal 3
        A       DBX 17.0                             //Slave 3B, terminal 1
        R       DBX 49.6                             //Slave 3B, terminal 3

//Output binary output data of the B slaves:
        CALL SFC 58                                  //WR_REC
        REQ     :=TRUE                               //Permanent trigger
        IOID    :=B#16#54                            //Fixed value
        LADDR   :=W#16#100                           //CP address (here 256 dec.)
        RECNUM  :=B#16#96                            //DSNR=150 (binary data B slaves)
        RECORD  :=P#DB20.DBX48.0 Byte 16             //Source area binary data
        RET_VAL :=MW12
        BUSY    :=M9.1
```

Transfering AS-I analog value



Transfering AS-I analog value

	Byte addresses occupied by analog values in the data record							
AS-i slave address	DS 140	DS 141	DS 142	DS 143	DS 144	DS 145	DS 146	DS 147
1	0–7							
2	8–15							
3	16–23							
4	24–31							
5	32–39	0–7						
6	40–47	8–15						
7	48–55	16–23						
8	56–63	24–31						
9	64–71	32–39	0–7					
10	72–79	40–47	8–15					
11	80–87	48–55	16–23					
12	88–95	56–63	24–31					
13	96–103	64–71	32–39	0–7				
14	104–111	72–79	40–47	8–15				
15	112–119	80–87	48–55	16–23				
16	120–127	88–95	56–63	24–31				
17		96–103	64–71	32–39	0–7			
18		104–111	72–79	40–47	8–15			
19		112–119	80–87	48–55	16–23			
20		120–127	88–95	56–63	24–31			
21			96–103	64–71	32–39	0–7		
22			104–111	72–79	40–47	8–15		
23			112–119	80–87	48–55	16–23		
24			120–127	88–95	56–63	24–31		
25				96–103	64–71	32–39	0–7	
26				104–111	72–79	40–47	8–15	
27				112–119	80–87	48–55	16–23	
28				120–127	88–95	56–63	24–31	
29					96–103	64–71	32–39	0–7
30					104–111	72–79	40–47	8–15
31					112–119	80–87	48–55	16–23

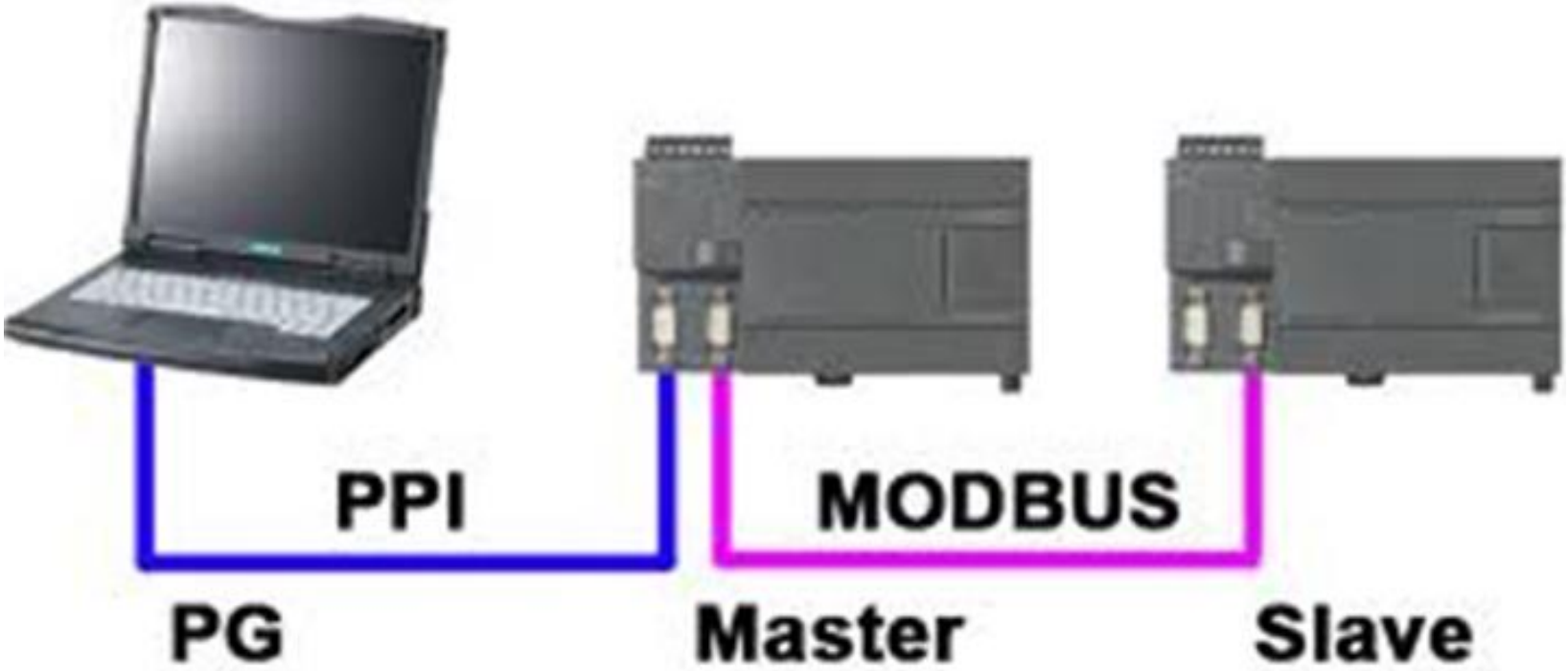
Example of transfering AS-I analog value

```
STL                                              Explanation
//Read in analog input data for slave 5:
       Call SFC 59                               //RD_REC
       REQ     :=TRUE                            //Permanent trigger
       IOID    :=B#16#54                         //Fixed value
       LADDR   :=W#16#100                        //CP address (here 256 dec.)
       RECNUM  :=B#16#8D                         //DSNR=141 (analog data slave 5...)
       RET_VAL :=MW14
       BUSY    :=M9.2
       RECORD  :=P#DB20.DBX64.0 BYTE 8           //Dest. area analog input data

//Examples: processing analog values:
       L       DB20DBW 64                        //Slave 5, input channel 1
       +       400
       T       DB20.DBW 88                       //Slave 6, output channel 1
       T       DB20.DBW 106                      //Slave 8, output channel 2

//Output analog output data for slave 5..8:
       CALL SFC 58                               //WR_REC
       REQ     :=TRUE                            //Permanent trigger
       IOID    :=B#16#54                         //Fixed value
       LADDR   :=W#16#100                        //CP address (here 256 dec.)
       RECNUM  :=B#16#8D                         //DSNR=141 (analog data slave 5...)
       RECORD  :=P#DB20.DBX80.0 Byte 32          //Source area analog output data
       RET_VAL::=MW16
       BUSY    :=M9.3
```

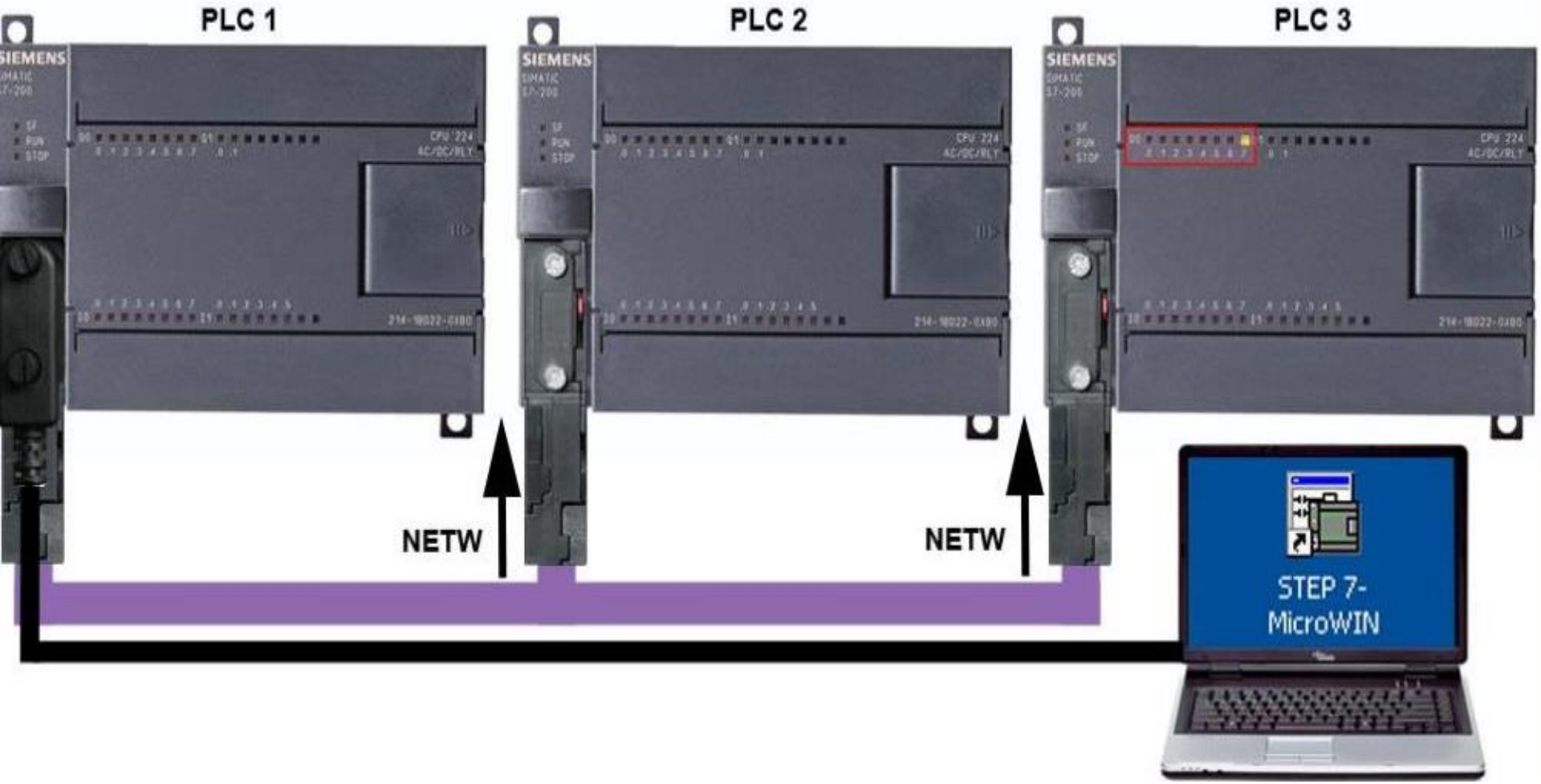
S7-200 MODBUS COMMUNICATION



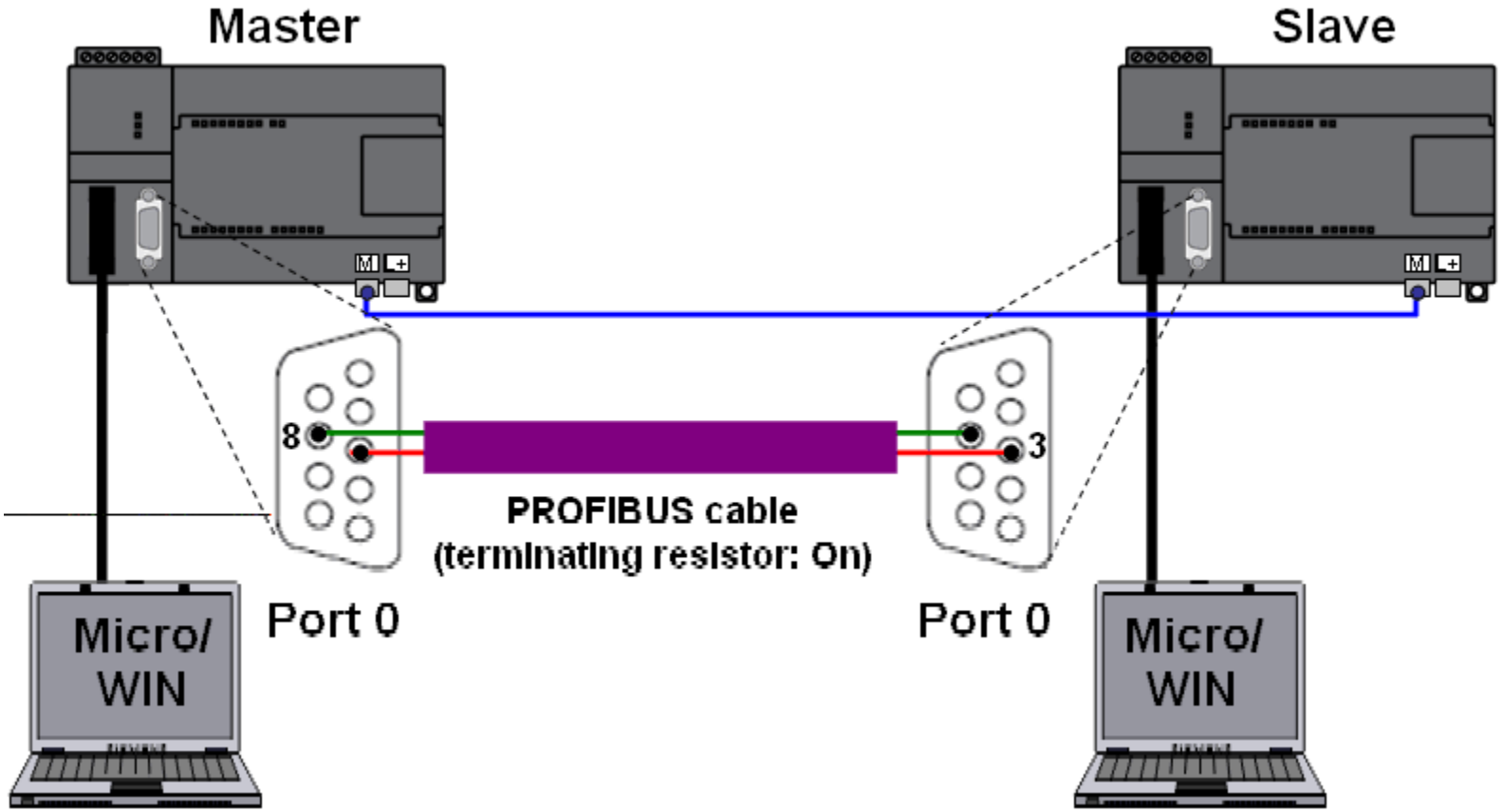
Giới thiệu về Modbus

- **Được sử dụng để thiết lập kết nối giữa Master/slave giữa 2 thiết bị thông minh**
- **Chỉ có một Master chiếm quyền điều khiển các Slave trong mạng.**
- **Slave chỉ đáp bằng cách cung cấp dữ liệu được yêu cầu đến Master**
- **Các Slave có thể là các cảm biến, valve, driver, thiết bị đo lường**

S7-200 MODBUS COMMUNICATION



S7-200 MODBUS CONNECTION



Các thiết bị kết nối với nhau qua Modbus theo chuẩn RS485, sử dụng cáp kết nối MPI, Profibus có điện trở đầu cuối cho trạm đầu và trạm cuối

S7-200 MODBUS COMMUNICATION

- Chỉ có một Master trong mạng
- Số lượng Slave 247
- Master và Slave trao đổi dữ liệu với nhau thông qua các hàm truyền thông modbus.

Master

MBUS Control: Khởi tạo modbus tại MS.

MBUS MSG: Truyền nhận dữ liệu

Slave

MBUS INT: Khởi tạo modbus tại SL

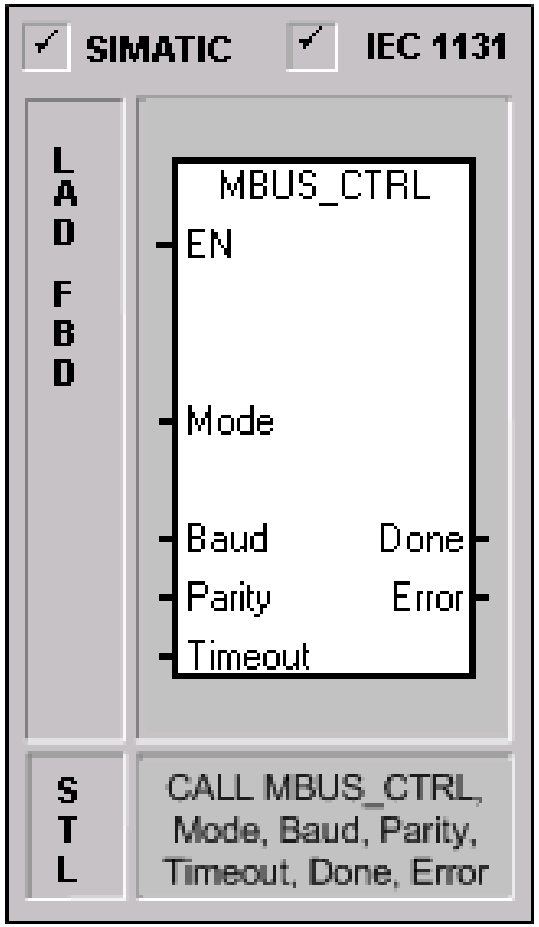
MODBUS SLAVE: Đáp ứng yêu cầu của MS

ĐIA CHỈ MODBUS TRONG S7200

Modbus Address	S7-200 Address	Modbus Address	S7-200 Address
00001	Q0.0	30001	AIW0
00002	Q0.1	30002	AIW2
00003	Q0.2	30003	AIW4
...	...	...	...
00127	Q15.6	30032	AIW62
00128	Q15.7		
		40001	Hold Start
10001	I0.0	40002	HoldStart+2
10002	I0.1	40003	HoldStart+4
10003	I0.2	...	...
...	...	4xxxx	HoldStart+2 x (xxxx-1)
10127	I15.6		
10128	I15.7		

HÀM MODBUS TẠI MASTER

MBUS Control: Khởi tạo modbus tại MS.



The MBUS_CTRL instruction for S7-200 port 0 (or MBUS_CTRL_P1 for port 1) is used to initialize, monitor, or to disable Modbus communications. Before the MBUS_MSG instruction can be used, the MBUS_CTRL instruction must be executed without errors. The instruction completes and the Done bit is set immediately before continuing to the next instruction. This instruction is executed on each scan when the **EN** input is on.

The MBUS_CTRL instruction must be called every scan (including the first scan) to allow it to monitor the progress of any outstanding messages initiated with the MBUS_MSG instruction. The Modbus Master Protocol will not operate correctly unless MBUS_CTRL is called every scan.

The value for the **Mode** input selects the communications protocol. An input value of 1 assigns the CPU port to Modbus protocol and enables the protocol. An input value of 0 assigns the CPU port to PPI system protocol and disables Modbus protocol.

The parameter **Baud** sets the baud rate to 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200 baud.

HÀM MODBUS TẠI MASTER

Các tham số ngõ vào hàm khởi tạo modbus

- EN: Cho phép.
- Mode: Chọn lựa giao thức truyền thông cho cổng giao tiếp của PLC
  + Mode = 0:Chuẩn PPI
  + Mode =1 : Chuẩn modbus
- Baud: Tốc độ truyền thông
- Parity: Kiểm tra chẳn lẻ
- Timeout: Thời gian đợi đáp ứng từ slave.

HÀM MODBUS TẠI MASTER

MBUS MSG: Truyền nhận dữ liệu

The MBUS_MSG instruction (or MBUS_MSG_P1 for port 1) is used to initiate a request to a Modbus slave and process the response.

The MBUS_MSG instruction initiates a request to a Modbus slave when both the EN input and the First inputs are on. Sending the request, waiting for the response, and processing the response usually requires several scans. The EN input must be on to enable the sending of the request, and should remain on until the Done bit is set.

Note: Only one MBUS_MSG instruction can be active at a time. If there is more than one MBUS_MSG instruction enabled, the first MBUS_MSG instruction executed will be processed and all subsequent MBUS_MSG instructions will abort with an error code 6.

The parameter **First** should be on for only one scan when there is a new request to send. The First input should be pulsed on through an edge detection element (i.e. Positive Edge) which will cause the request to be transmitted one time. See the example program.

The parameter **Slave** is the address of the Modbus slave device. The allowed range is 0 through 247. Address 0 is the broadcast address and can only be used for write requests. There is no response to a broadcast request to address 0. Not all slave devices will support the broadcast address. The S7-200 Modbus Slave Library does not support the broadcast address.

HÀM MODBUS TẠI MASTER

Các tham số ngõ vào hàm MBUS MSG:

- EN: Cho phép hàm hoạt động
- First: Kích hoạt việc truyền nhận dữ liệu.
- Slave: Địa chỉ của slave 1 đến 247
- RW: Ngõ vào điều khiển đọc ghi dữ liệu.
  RW = 0: Đọc dữ liệu từ slave về master
  RW = 1: Ghi dữ liệu từ master đến slave
- Address: Địa chỉ modbus trong Slave.

HÀM MODBUS TẠI MASTER

Các tham số ngõ vào hàm MBUS MSG:

- **Count**:Số lượng bit hay word dữ liệu được đọc hay ghi (Phụ thuộc vào loại dữ liệu được sử dụng)
- **DataPtr**: Con trỏ địa chỉ: Chỉ địa chỉ bắt đầu của vùng nhớ V trong S7 200 tại Master.
- **Done**: Ngõ ra báo hiệu việc ghi hay đọc đã hoàn tất.
- **Error**:Báo mã lỗi trong trường hợp xãy ra lỗi.

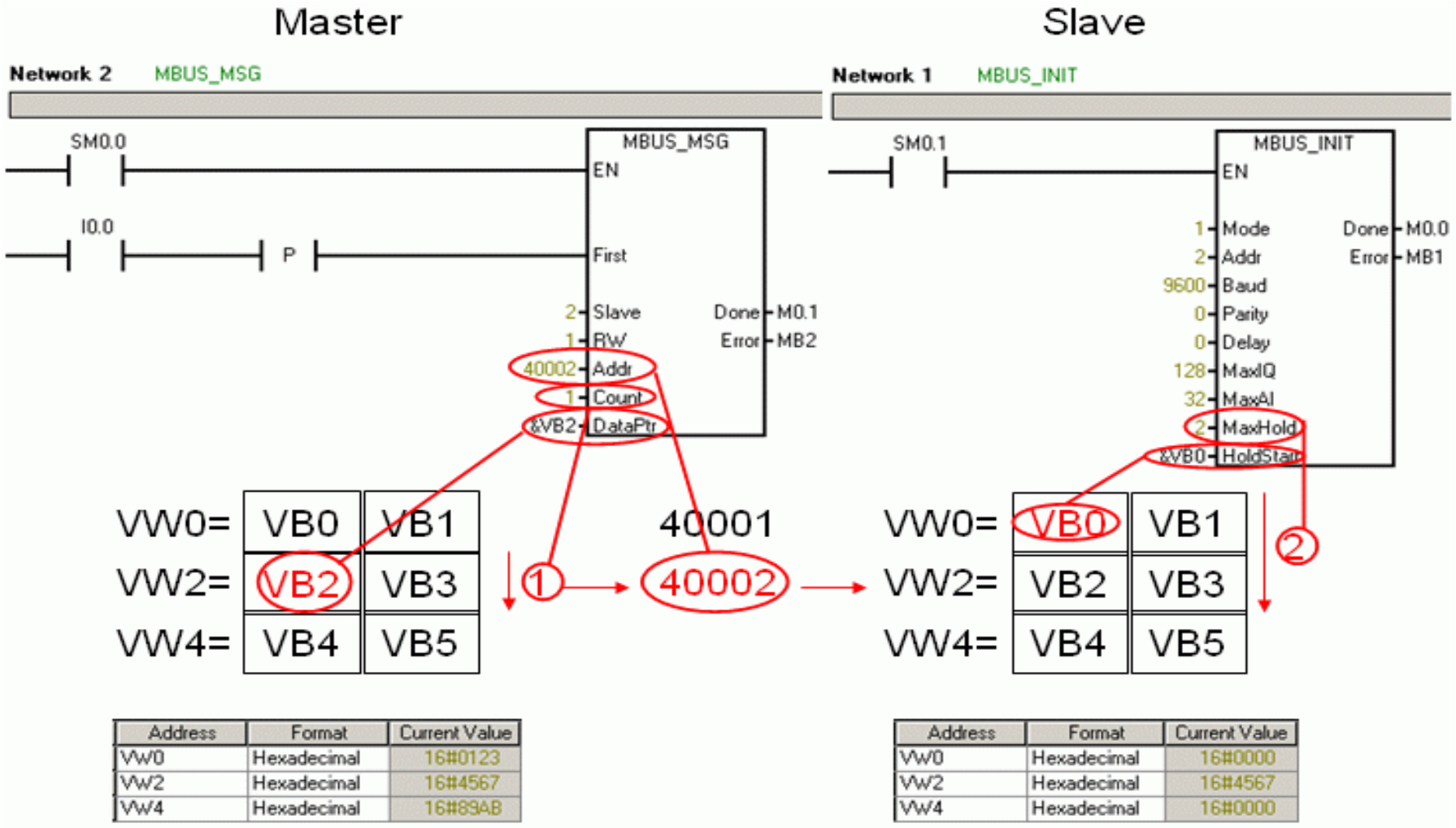
MÃ LỖI XÃY CỦA HÀM MBUS MGS TẠI MASTER

MBUS_MSG Error Codes	Description
0	No error
1	Parity error in response: This is only possible if even or odd parity is used. The transmission was disturbed and possibly incorrect data was received. This error is usually caused by an electrical problem such as incorrect wiring or electrical noise affecting the communication.
2	Not used
3	Receive timeout: There was no response from the slave within the Timeout time. Some possible causes are bad electrical connection to the slave device, master and slave are set to a different baud rate / parity setting, and incorrect slave address.
4	Error in request parameter: One or more of the input parameters (Slave, RW, Addr, or Count) is set to an illegal value. Check the documentation for allowed values for the input parameters.
5	Modbus master not enabled: Call MBUS_CTRL on every scan prior to calling MBUS_MSG.
6	Modbus is busy with another request: Only one MBUS_MSG instruction can be active at a time.
7	Error in response: The response received does not correspond to the request. This indicates some problem in the slave device or that the wrong slave device answered the request.
8	CRC error in response: The transmission was disturbed and possibly incorrect data was received. This error is usually caused by an electrical problem such as incorrect wiring or electrical noise affecting the communication.

MÃ LỖI XÃY CỦA HÀM MBUS MGS TẠI MASTER

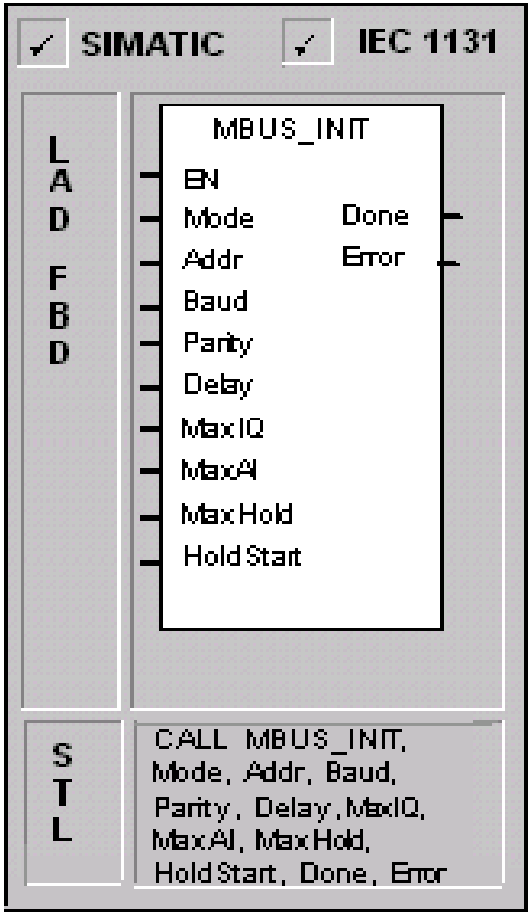
101	Slave does not support the requested function at this address: See the required Modbus slave function support table in the "Using the Modbus master Instructions" help topic.
102	Slave does not support the data address: The requested address range of Addr plus Count is outside the allowed address range of the slave.
103	Slave does not support the data type: The Addr type is not supported by the slave device.
104	Slave device failure
105	Slave accepted the message but the response is delayed:: This is an error for MBUS_MSG and the user program should resend the request at a later time.
106	Slave is busy and rejected the message: You can try the same request again to get a response.
107	Slave rejected the message for an unknown reason
108	Slave memory parity error: There is an error in the slave device.

HÀM TRUYỀN NHẬN MODBUS



HÀM MODBUS TẠI SLAVE

MBUS INTI: Khởi tạo modbus tại SLAVE.



The MBUS_INIT instruction is used to enable and initialize, or to disable Modbus communications. Before the MBUS_SLAVE instruction can be used, the MBUS_INIT instruction must be executed without errors. The instruction completes and the Done bit is set immediately, before continuing to the next instruction.

The instruction is executed on each scan when the EN input is on.

The MBUS_INIT instruction should be executed exactly once for each change in communications state. Therefore, the EN input should be pulsed on through an edge detection element, or executed only on the first scan.

The value for the **Mode** input selects the communications protocol: an input value of 1 assigns port 0 to Modbus protocol and enables the protocol, and an input value of 0 assigns port 0 to PPI and disables Modbus protocol.

The parameter **Baud** sets the baud rate at 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200.

The parameter **Addr** sets the address at inclusive values between 1 and 247.

The parameter **Parity** is set to match the parity of the Modbus master. The accepted values are:

0-no parity

1-odd parity

2-even parity

HÀM MODBUS TẠI SLAVE

Các tham số ngõ vào hàm MBUS INIT

EN: Cho phép khởi tạo

Mode: Chọn chế độ truyền thông

Address: Địa chỉ của slave (từ 1 đến 247)

Baud: Tốc độ truyền thông: Từ 1200 đến 115200.

Parity: Bit kiểm tra chẳn lẻ.

Delay: Thời gian chờ để nhận dữ liệu (0 đến 32767 ms)

HÀM MODBUS TẠI SLAVE

Các tham số ngõ vào hàm MBUS INIT

MaxIQ: Số lượng ngõ vào, ngõ ra cho phép đọc, ghi (0 đến 128)

MaxAI: Số lượng analog cho phép ghi 0 đến 32)

Maxhold:Số lượng word tối đa cho phép truy xuất trong slave

HoldStart: Địa chỉ bắt đầu của vùng nhớ V trong slave cho phép master truy xuất.

Done: Báo trạng thái hàm khởi tạo hoàn thành hay chưa

Error: Báo mã lỗi nếu khởi hàm khởi tạo bị lỗi

MÃ LỖI CỦA HÀM MBIS INIT TẠI SLAVE

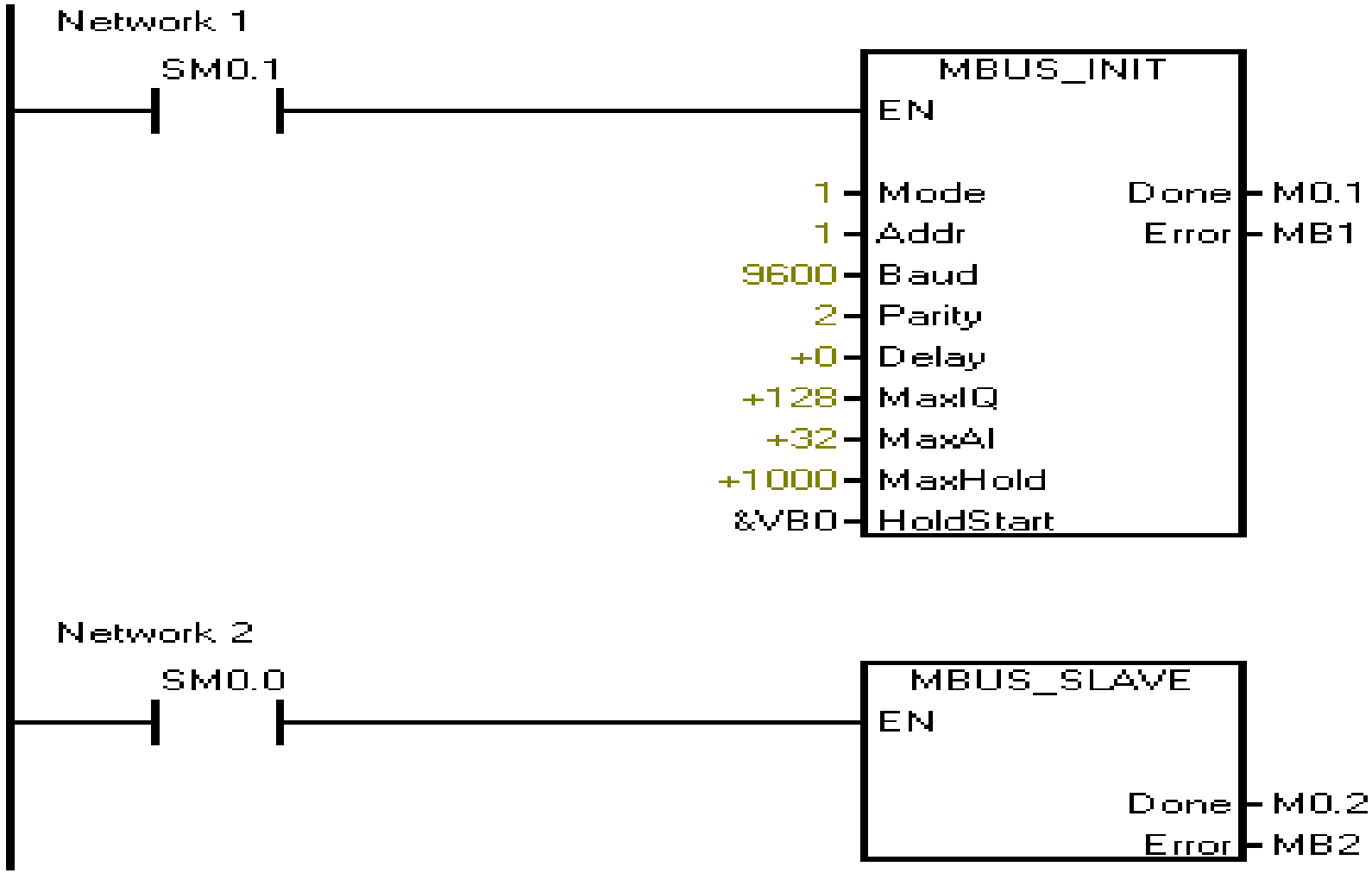
Modbus Slave Protocol Execution Error Codes

Error Codes	Description
0	No Error
1	Memory range error
2	Illegal baud rate or parity
3	Illegal slave address
4	Illegal value for Modbus parameter
5	Holding registers overlap Modbus Slave symbols
6	Receive parity error
7	Receive CRC error
8	Illegal function request/function not supported
9	Illegal memory address in request
10	Slave function not enabled

CÁC HÀM MBUS ĐƯỢC SỬ DỤNG TẠI SLAVE

Program Example

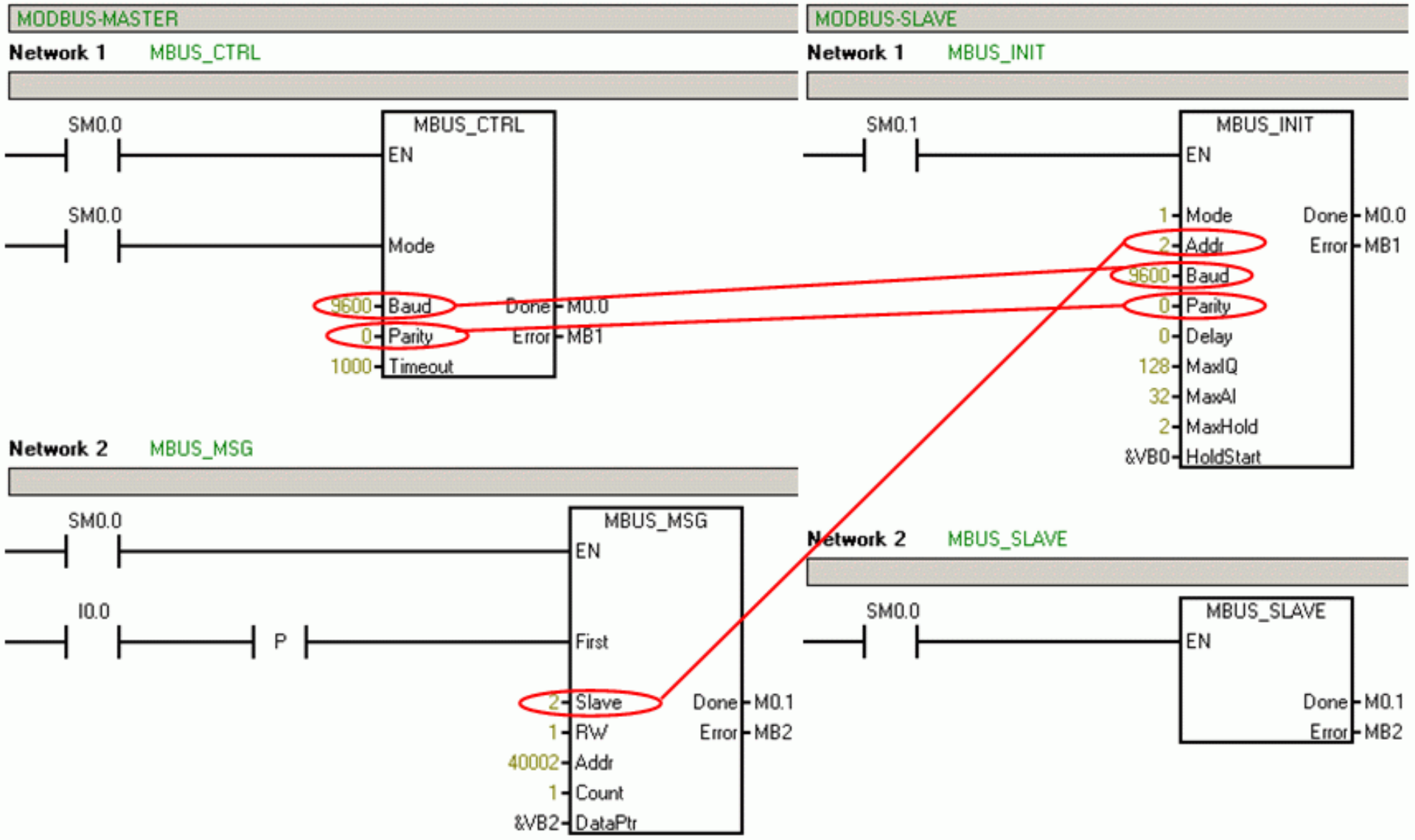
LAD



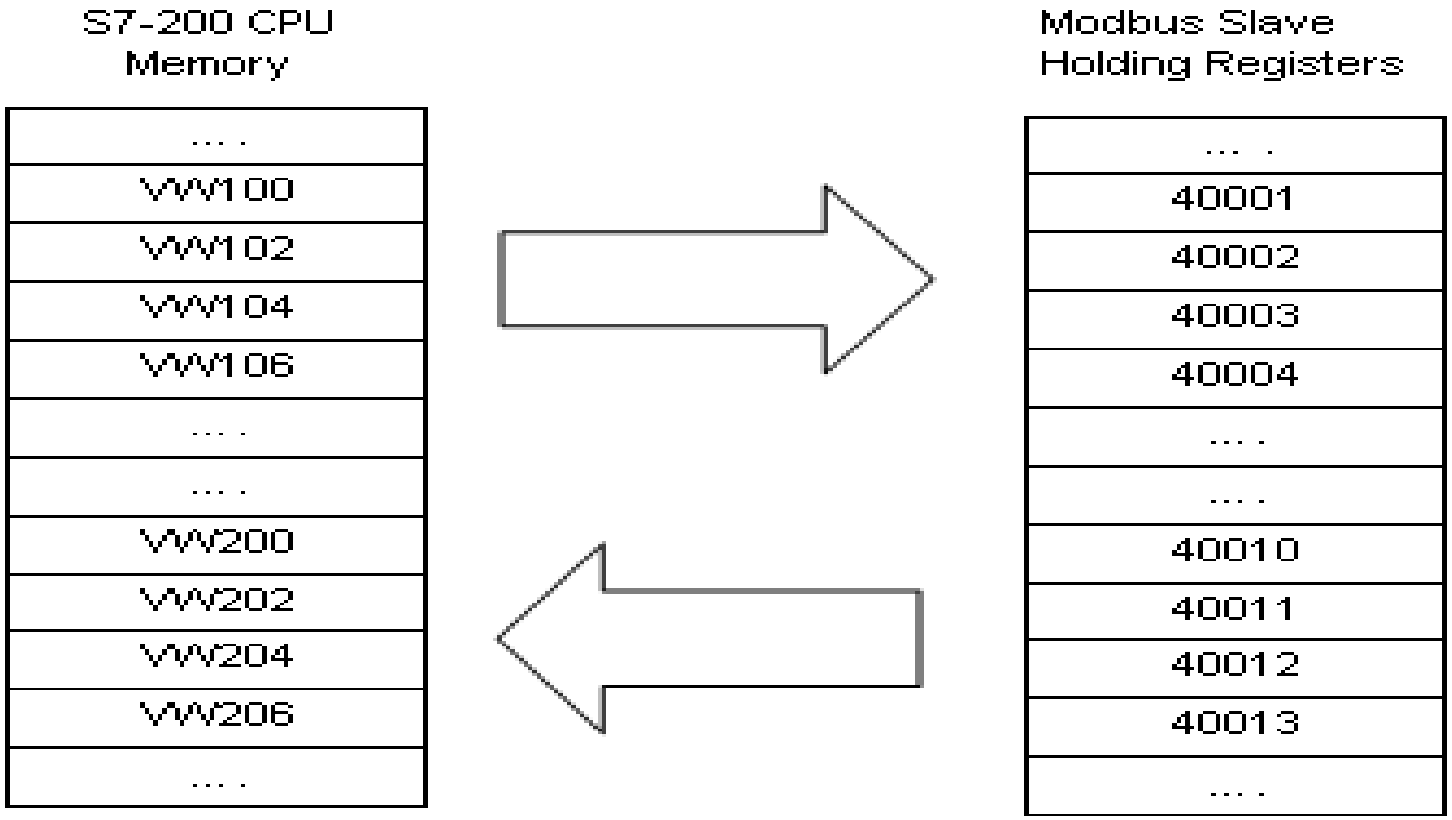
HÀM MBUS SLAVE

- MBUS SLAVE: Được sử dụng để phục vụ yêu cầu từ master.
- MBUS SLAVE: Phải được gọi mỗi chu kỳ quét của chương trình.
- Nếu slave đáp ứng các yêu cầu của master thì bit **Done = 1**, ngược lại **Done** = 0.
- Error được sử dụng để báo lỗi trong trường hợp xãy ra lỗi.

TRUYỀN NHẬN DỮ LIỆU MODBUS



TRUYỀN NHẬN DỮ LIỆU MODBUS



- ✓ **Viết chương trình cho Master và Slave để truyền và nhận dữ liệu như hình trên.**
- ✓ **Kết nối PLC, download chương trình để kiểm tra**