How to Use the MGate 4101-MB-PBS Paging Function

Moxa Technical Support Team

support@moxa.com

Contents

Released on Oct 13, 2023

About Moxa

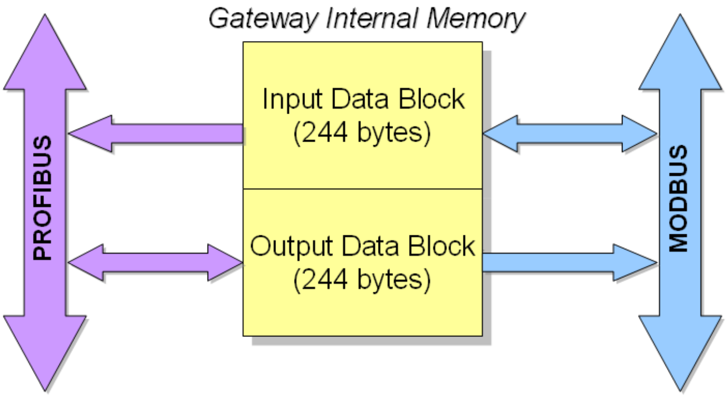
Moxa is a leading provider of edge connectivity, industrial computing, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With 35 years of industry experience, Moxa has connected more than 82 million devices worldwide and has a distribution and service network that reaches customers in more than 80 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa's solutions is available at www.moxa.com.

How to Contact Moxa

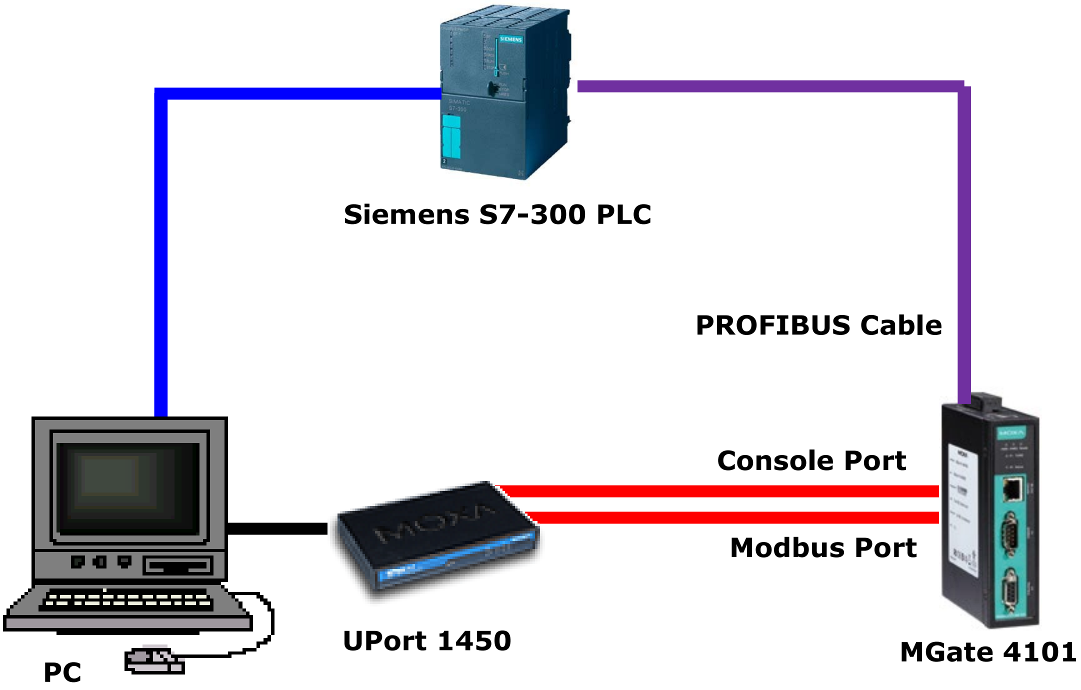
Tel: 1-714-528-6777
Fax: 1-714-528-6778

1 Application Description

PROFIBUS allows a maximum size of 244 bytes of input data, 244 bytes of output data, and 400 bytes of I/O data. But what options are available if your application needs to exceed the maximum allowed data size? With the MGate 4101 paging function, you can separate PROFIBUS I/O modules into different pages. The MGate 4101 supports up to 32 pages, and each page allows a maximum of 244 bytes of input/output data. So with multiple pages, you can overcome this restriction on data size. But you need to program the PLC to change the page number, and you can exchange only one page of data at a time via PROFIBUS. Sequential programming is crucial for accessing data from all pages. This document illustrates how to program the Siemens S7-300 PLC using the MGate 4101 paging function.



2 System Topology



1. **Modbus Side:**

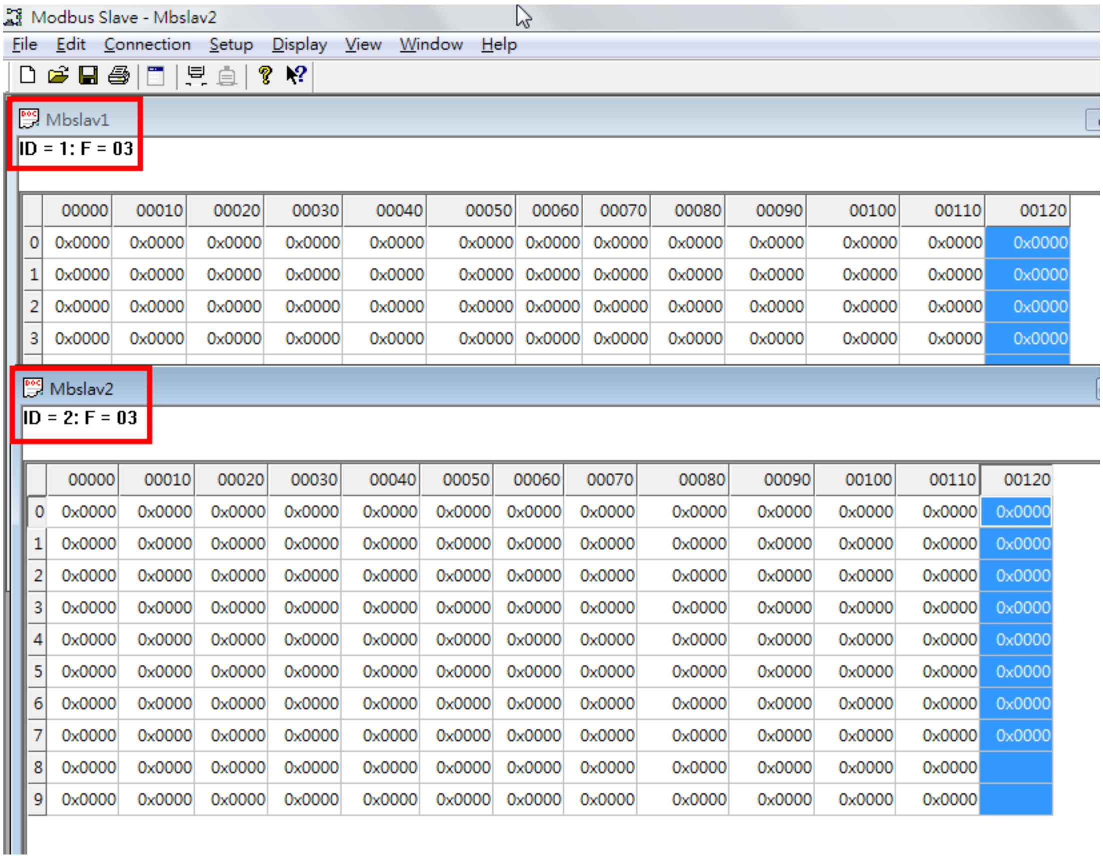
   The MGate 4101 Modbus Port connects to a UPort and a PC, simulating two Modbus server (slave) devices, ID 1 and ID 2. Each Modbus server (slave) simulates 64 Modbus registers for read and 64 registers for write. So, the total is (64+64) x 2 bytes x 2 devices = 512 bytes, exceeding the maximum 400 bytes of I/O data allowed by PROFIBUS. In this illustration, Modbus data of server (slave) ID 1 is mapped to PROFIBUS **Page 1** in MGate 4101, and data of server (slave) ID 2 is mapped to PROFIBUS **Page 2**.

2. **PROFIBUS Side:**

   The MGate 4101 PROFIBUS Port connects to the Siemens S7-300 PLC, running PROFIBUS protocol. The PLC connects to the PC because Siemens Step 7 program development software is installed on the PC, and the program will be later downloaded from the PC to the PLC. The PROFIBUS I/O setting includes 1 word of input module and 1 word of output module to change the pages, and 64 words of input modules and 64 words of output modules for data mapping.

Modbus Server (Slave) Settings

Execute the Modbus server (slave) simulation tool on the PC. Create Server (Slave) ID 1 and ID 2, simulating two Modbus server (slave) devices. Register addresses **0 to 63 are for write**, and register addresses **64 to 127 are for read**.



Mbslav1

ID = 1: F = 03

	00000	00010	00020	00030	00040	00050	00060	00070	00080	00090	00100	00110	00120
0	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
1	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
2	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
3	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000

Mbslav2

ID = 2: F = 03

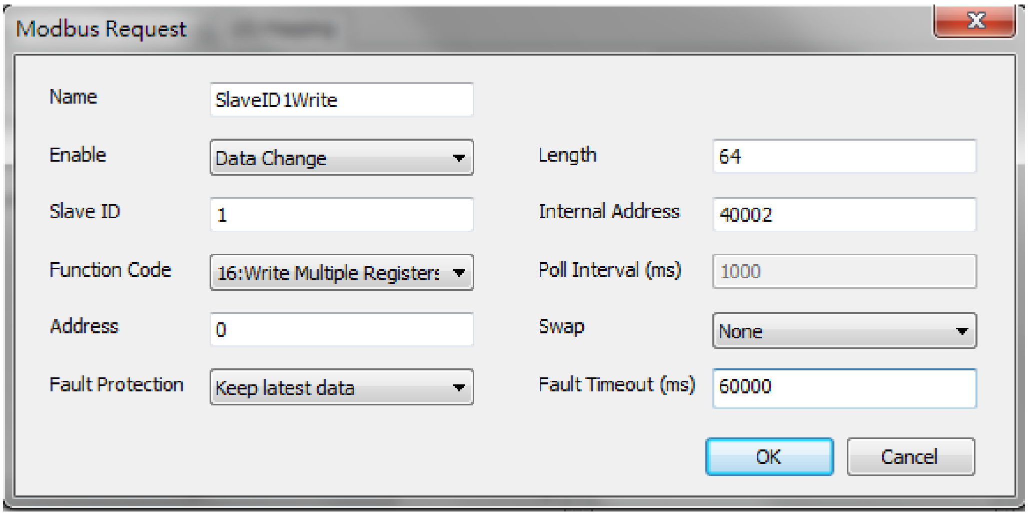
	00000	00010	00020	00030	00040	00050	00060	00070	00080	00090	00100	00110	00120
0	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
1	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
2	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
3	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
4	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
5	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
6	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
7	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
8	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	
9	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	

3 MGate 4101 Settings

3.1 Configure Modbus Commands

Start MGate Manager utility for the configuration of the MGate 4101. In the **IO Mapping** tab, add the following Modbus commands:

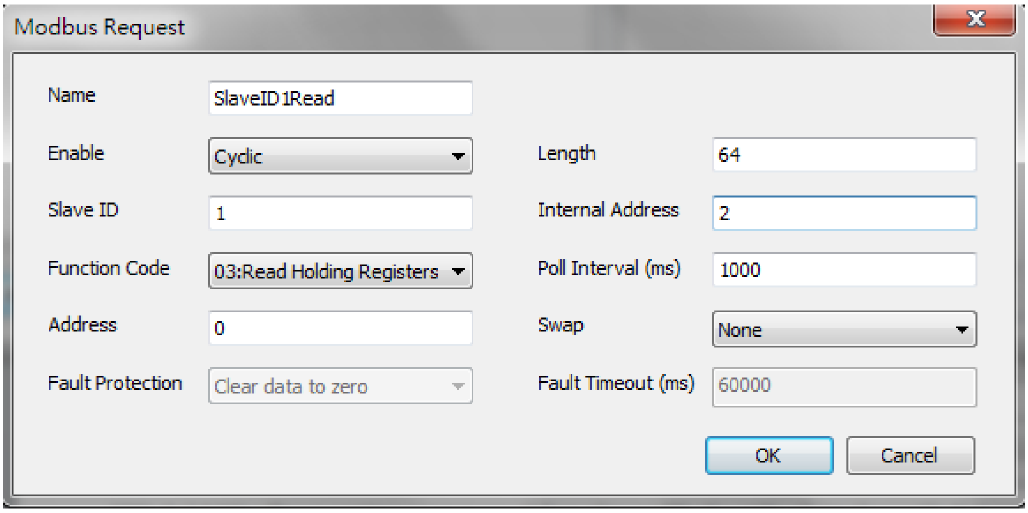
1. Add the Modbus write command for server (slave) ID 1 with a data length of 64 registers (register addresses 0 to 63). This data is stored in internal memory address **40002**, the start address of PROFIBUS page 1 output module.



Modbus Request

Name	SlaveID1Write		
Enable	Data Change	Length	64
Slave ID	1	Internal Address	40002
Function Code	16:Write Multiple Registers	Poll Interval (ms)	1000
Address	0	Swap	None
Fault Protection	Keep latest data	Fault Timeout (ms)	60000

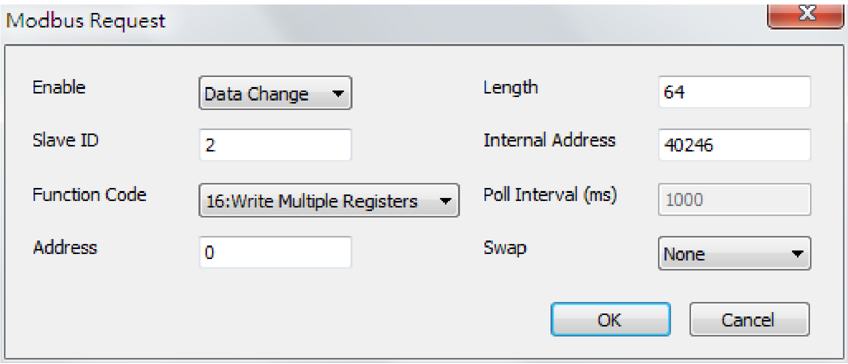
2. Add the Modbus read command for server (slave) ID 1 with a data length of 64 registers (register addresses 64 to 127). This data is stored in internal memory address **2**, the start address of PROFIBUS page 1 input module. The read polling time is set at 1,000 ms.



Modbus Request

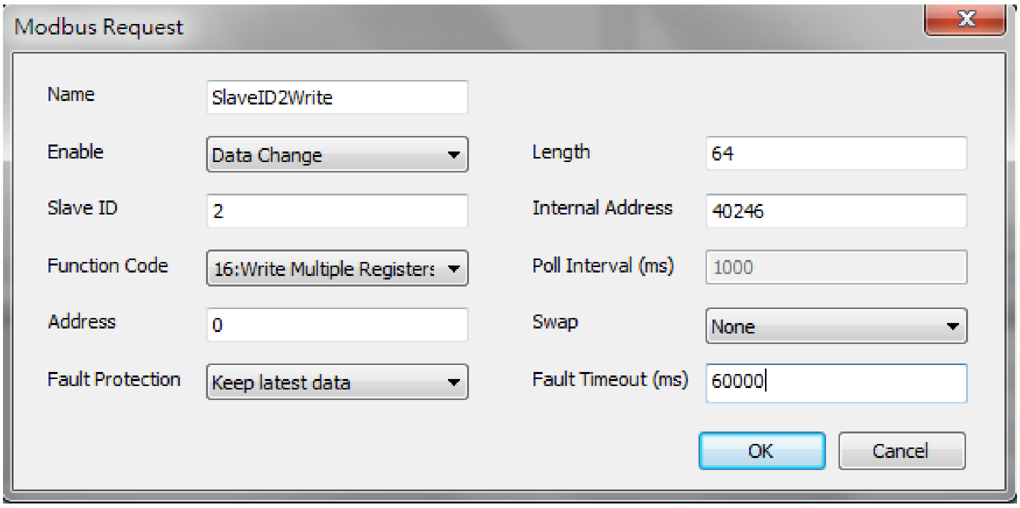
Name	SlaveID1Read		
Enable	Cyclic	Length	64
Slave ID	1	Internal Address	2
Function Code	03:Read Holding Registers	Poll Interval (ms)	1000
Address	0	Swap	None
Fault Protection	Clear data to zero	Fault Timeout (ms)	60000

3. Add Modbus write command for server (slave) ID 2 with a data length of 64 registers (register addresses 0 to 63). This data is stored in internal memory address **40246**, the start address of PROFIBUS page 2 output module.



Modbus Request

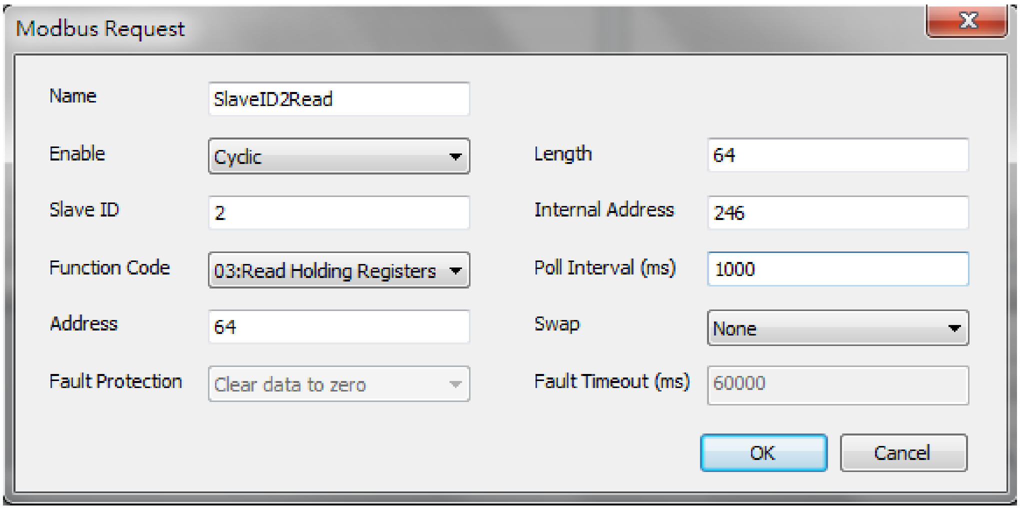
Enable	Data Change	Length	64
Slave ID	2	Internal Address	40246
Function Code	16:Write Multiple Registers	Poll Interval (ms)	1000
Address	0	Swap	None



Modbus Request

Name	SlaveID2Write		
Enable	Data Change	Length	64
Slave ID	2	Internal Address	40246
Function Code	16:Write Multiple Register	Poll Interval (ms)	1000
Address	0	Swap	None
Fault Protection	Keep latest data	Fault Timeout (ms)	60000

4. Add Modbus read command for server (slave) ID 2 with a data length of 64 registers (register addresses 64 to 127). This data is stored in internal memory address **246**, the start address of PROFIBUS page 2 input module address. The read polling time is set at 1,000 ms.

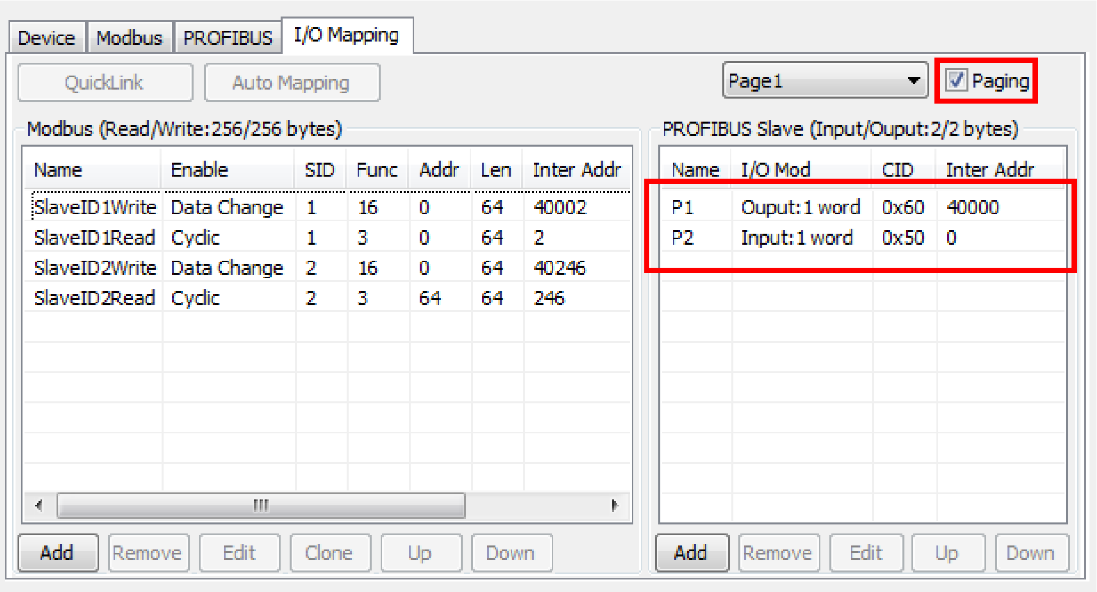


Modbus Request

Name	SlaveID2Read		
Enable	Cyclic	Length	64
Slave ID	2	Internal Address	246
Function Code	03:Read Holding Registers	Poll Interval (ms)	1000
Address	64	Swap	None
Fault Protection	Clear data to zero	Fault Timeout (ms)	60000

3.2 Configure the PROFIBUS Module

Activate the Paging function by ticking the Paging checkbox. A **Output 1 word** and **Input 1 word** module will be added automatically to the table, as shown in the figure below. **Output 1 word** is for changing the page according to the value that the PLC program filled in. **Input 1 word** is to show the page status.



Page1 ▾ ☑ Paging

Modbus (Read/Write:256/256 bytes)

Name	Enable	SID	Func	Addr	Len	Inter Addr
SlaveID1Write	Data Change	1	16	0	64	40002
SlaveID1Read	Cyclic	1	3	0	64	2
SlaveID2Write	Data Change	2	16	0	64	40246
SlaveID2Read	Cyclic	2	3	64	64	246

PROFIBUS Slave (Input/Ouput:2/2 bytes)

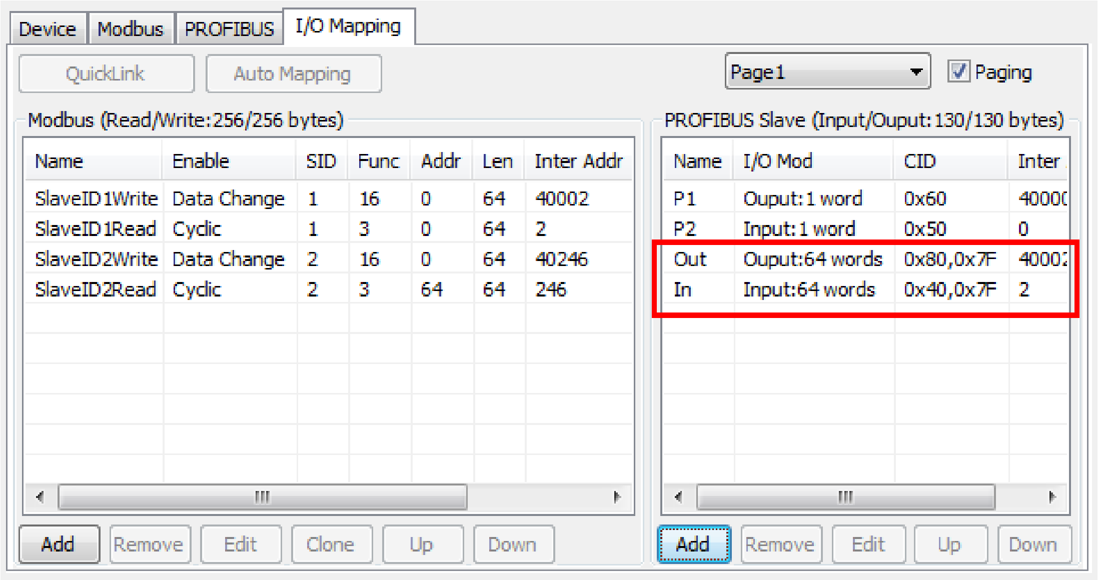
Name	I/O Mod	CID	Inter Addr
P1	Ouput:1 word	0x60	40000
P2	Input:1 word	0x50	0

Description:

1. Output 1 word: The PROFIBUS master controls which page is for input data or output data. Put the output data page number in the first byte. Put the input data page number in the second byte. For instance, 0x0101 stands for writing the output to page 1 and shows the MGate 4101 changed the input page to 1.
2. Input 1 word: The value in the first byte is the current output page in the MGate 4101. The value in the second byte is the page number of the input data in the MGate 4101.

   For instance, 0x0101 stands for output data changes to page 1, and the input data is from page 1 in MGate 4101.

Add “64 Word Output Module” and “64 Word Input Module” for Modbus data.



Modbus (Read/Write:256/256 bytes)

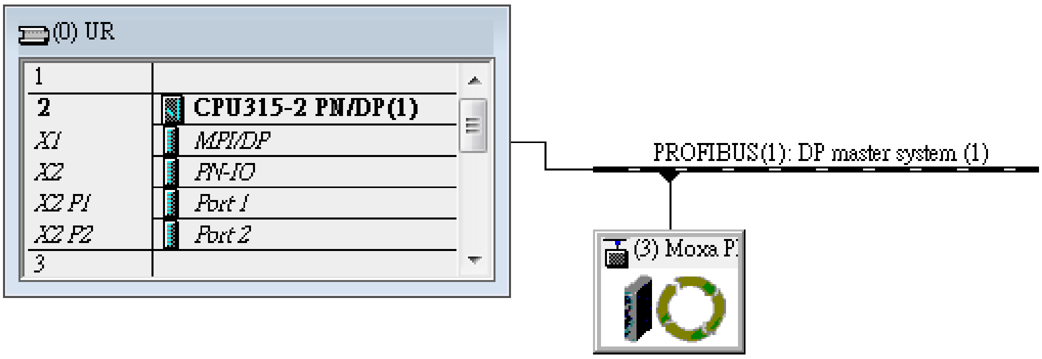
Name	Enable	SID	Func	Addr	Len	Inter Addr
SlaveID1Write	Data Change	1	16	0	64	40002
SlaveID1Read	Cyclic	1	3	0	64	2
SlaveID2Write	Data Change	2	16	0	64	40246
SlaveID2Read	Cyclic	2	3	64	64	246

PROFIBUS Slave (Input/Ouput:130/130 bytes)

Name	I/O Mod	CID	Inter
P1	Ouput:1 word	0x60	4000
P2	Input:1 word	0x50	0
Out	Ouput:64 words	0x80,0x7F	4000
In	Input:64 words	0x40,0x7F	2

4 Siemens PLC Setting

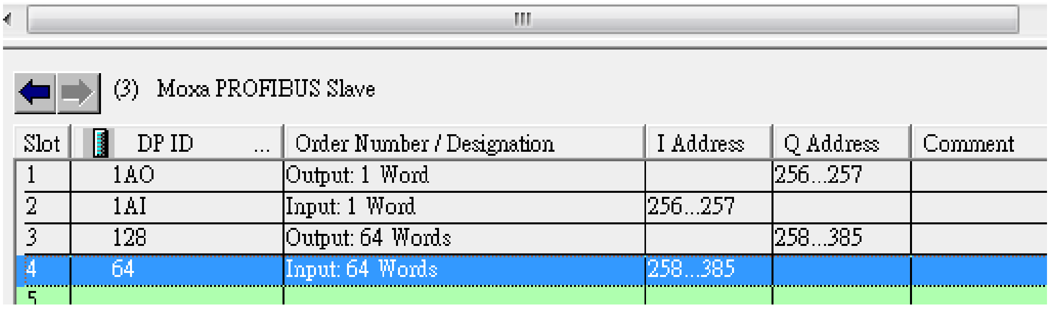
1. In Siemens Step 7 software, insert "1 Word Output", "1 Word Input", "64 Word Output", and "64 Word Input" module in the **HW Config**. It should be the same as the IO nodule configuration in the MGate 4101.



(0) UR

1	
2	CPU315-2 PN/DP(1)
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	

The figure below shows the I Address and Q Address for the PLC program to access the data of the input and output words.



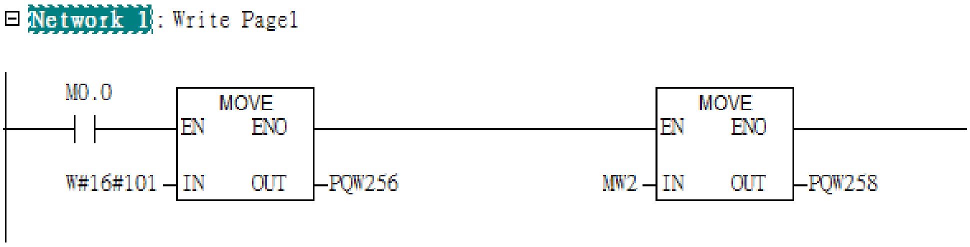
(3) Moxa PROFIBUS Slave

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	1AO	Output: 1 Word		256...257	
2	1AI	Input: 1 Word	256...257		
3	128	Output: 64 Words		258...385	
4	64	Input: 64 Words	258...385		
5					

2. Add the following PLC program in **OB1**:

   a. Write Page 1:

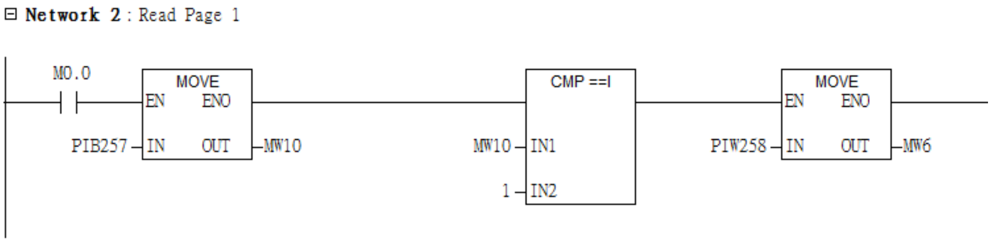
   If **M0.0** is **True**, write value 0x0101 into **PQW256** (Q Address, Module Output 1 Word). The output value will write to page 1 and will also ask that the MGate 4101 input should change to page 1. Also, move the **MW2** value to **PQW258** (Q Address, Module Output 64 Word). At the Modbus side, the data will write to server (slave) ID 1.



Network 1: Write Page1

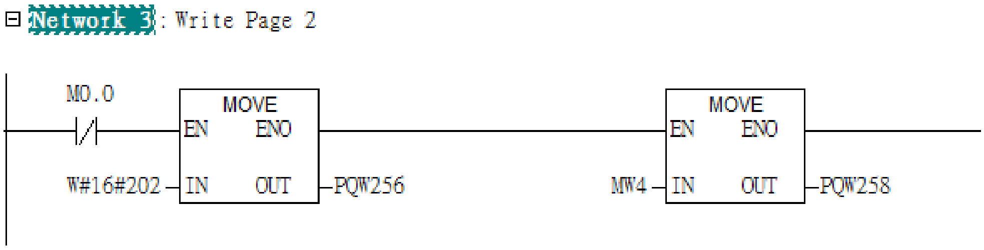
b. **Read Page 1:**

If **M0.0** is **True**, check whether **PIB257** (the second byte of Module Input 1 word) equals to 1 or not. If yes, move the PIW**258** value to **MW6**. This value means the Input Module was read from page 1. So **MW6** is the value of the first two bytes of Module Input 64 Word, which also stands for the value of **Modbus server (slave) ID 1 register address 64**.



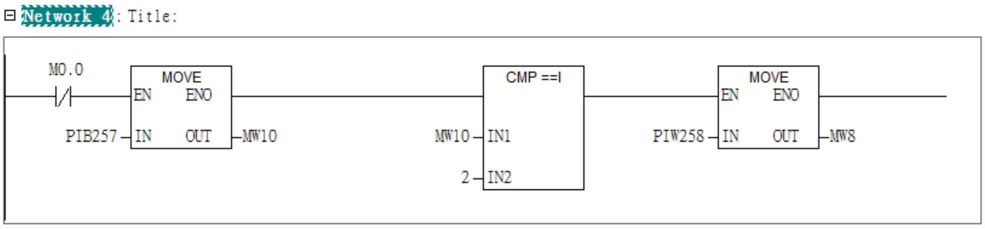
c. **Write Page 2:**

If **M0.0** is **false**, write value 0x0202 into **PQW256** (Q Address, Module Output 1 Word). The output value will write to page 2 and will also ask that the MGate 4101 input change to page 2. Also move the **MW4** value to **PQW258** (Q Address, Module Output 64 Word). At the Modbus side, the data will write to server (slave) ID 2.



d. **Read Page 2:**

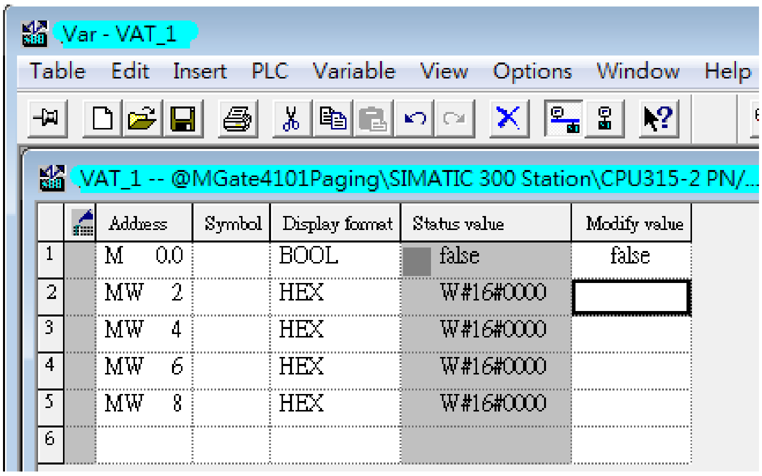
If **M0.0** is **False**, check whether **PIB257** (the second byte of Module Input 1 Word) equals to 2 or not. If yes, move the **PIW258** value to **MW8**. This value means that the Input Module was read from page 2. So **MW8** is the value of the first two bytes of Module Input 64 Word, which also stands for the value of **Modbus server (slave) ID 2 register address 64**.



5 Communication Test

1. Create a Variable Table

Add MW2 to MW8 in the variable table for monitoring, then enter **On-Line** mode.

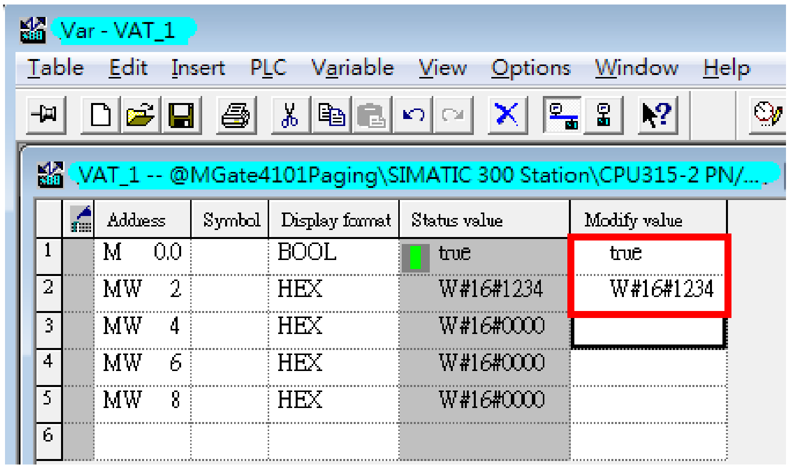


VAT_1 -- @MGate4101Paging\SIMATIC 300 Station\CPU315-2 PN/...

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	false	false
2	MW 2		HEX	W#16#0000	
3	MW 4		HEX	W#16#0000	
4	MW 6		HEX	W#16#0000	
5	MW 8		HEX	W#16#0000	
6					

2. Write value to Page 1

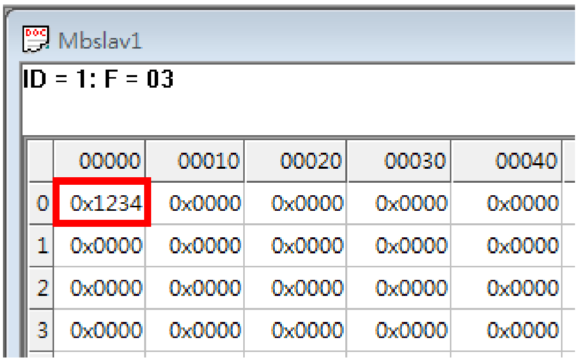
Set **M0.0** = true, **MW2** = 0x1234



VAT_1 -- @MGate4101Paging\SIMATIC 300 Station\CPU315-2 PN/...

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	🟩 true	true
2	MW 2		HEX	W#16#1234	W#16#1234
3	MW 4		HEX	W#16#0000	
4	MW 6		HEX	W#16#0000	
5	MW 8		HEX	W#16#0000	
6					

After changing the value, you can check that the value becomes 0x1234 in register 0 of Modbus server (slave) ID 1.

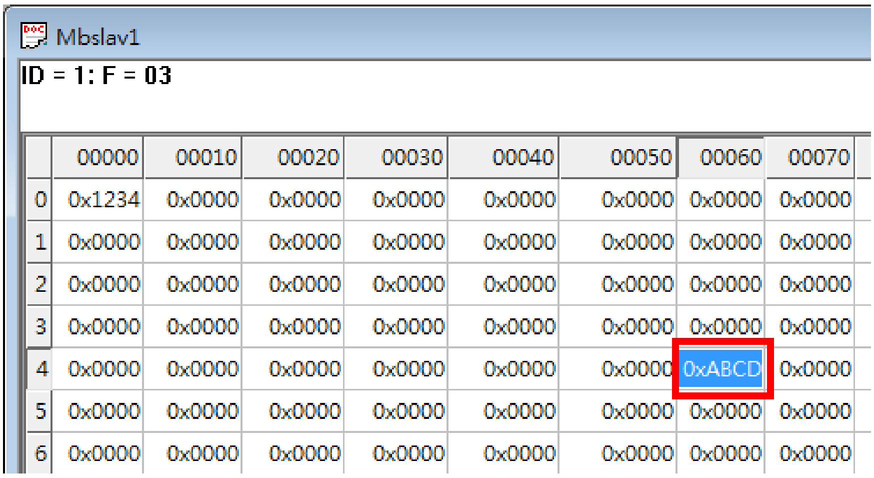


ID = 1: F = 03

	00000	00010	00020	00030	00040
0	0x1234	0x0000	0x0000	0x0000	0x0000
1	0x0000	0x0000	0x0000	0x0000	0x0000
2	0x0000	0x0000	0x0000	0x0000	0x0000
3	0x0000	0x0000	0x0000	0x0000	0x0000

3. **Read value from Page 1**

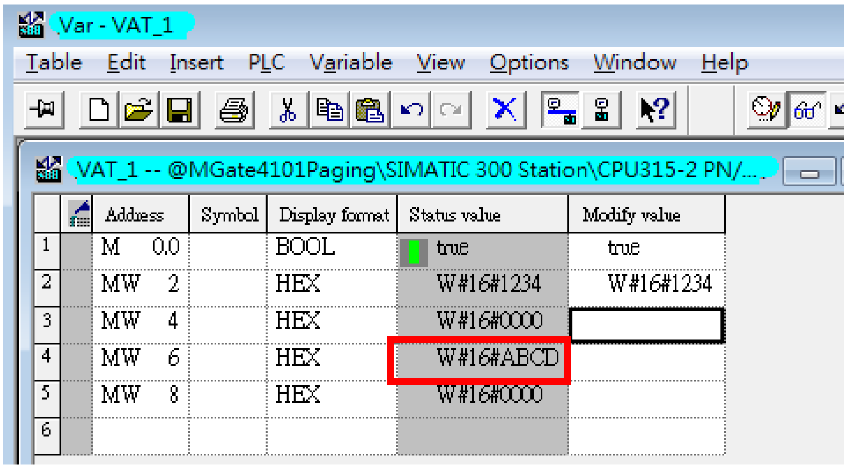
   Change the value of register 64 of Modbus server (slave) ID 1 to 0xABCD.



ID = 1: F = 03

	00000	00010	00020	00030	00040	00050	00060	00070
0	0x1234	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
1	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
2	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
3	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
4	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0xABCD	0x0000
5	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
6	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000

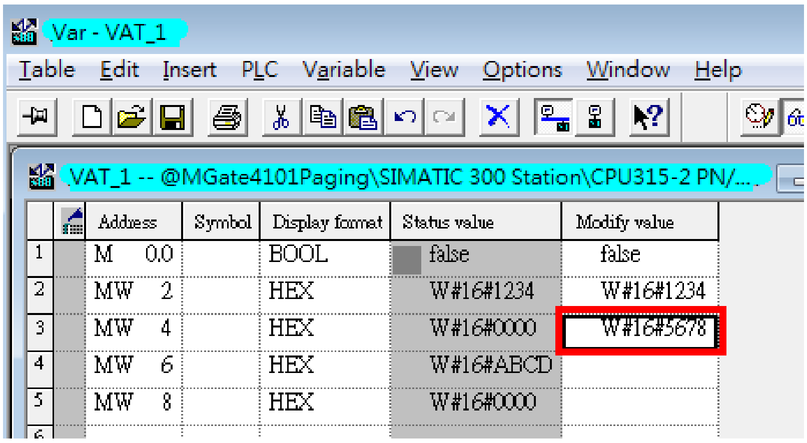
   You can see the MW**6** variable becomes 0xABCD in the following figure.


Var - VAT_1
Table Edit Insert PLC Variable View Options Window Help
VAT_1 -- @MGate4101Paging\SIMATIC 300 Station\CPU315-2 PN/...

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	🟩 true	true
2	MW 2		HEX	W#16#1234	W#16#1234
3	MW 4		HEX	W#16#0000	
4	MW 6		HEX	W#16#ABCD	
5	MW 8		HEX	W#16#0000	
6					

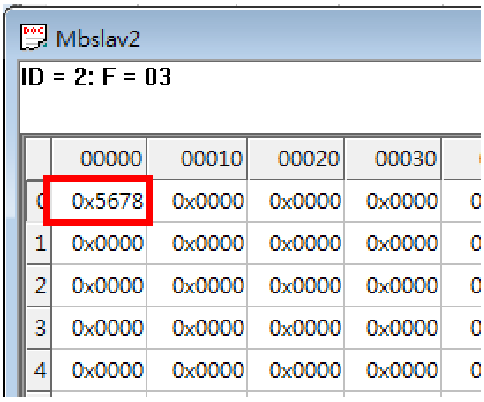
4. **Write value to Page 2**

   Set **M0.0** = false, **MW4** = 0x5678


Var - VAT_1
Table Edit Insert PLC Variable View Options Window Help
VAT_1 -- @MGate4101Paging\SIMATIC 300 Station\CPU315-2 PN/...

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	▨ false	false
2	MW 2		HEX	W#16#1234	W#16#1234
3	MW 4		HEX	W#16#0000	W#16#5678
4	MW 6		HEX	W#16#ABCD	
5	MW 8		HEX	W#16#0000	

After changing the value, you can check that the value should be 0x5678 in register 0 of Modbus server (slave) ID 2.

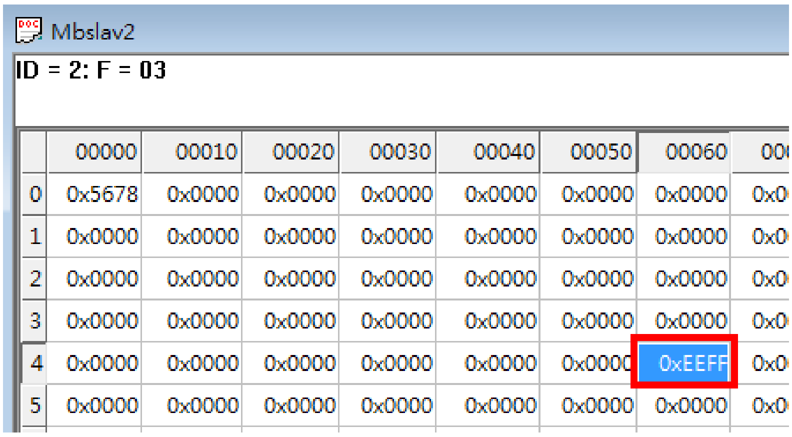


ID = 2: F = 03

	00000	00010	00020	00030	
0	0x5678	0x0000	0x0000	0x0000	0
1	0x0000	0x0000	0x0000	0x0000	0
2	0x0000	0x0000	0x0000	0x0000	0
3	0x0000	0x0000	0x0000	0x0000	0
4	0x0000	0x0000	0x0000	0x0000	0

5. **Read value from Page 2**

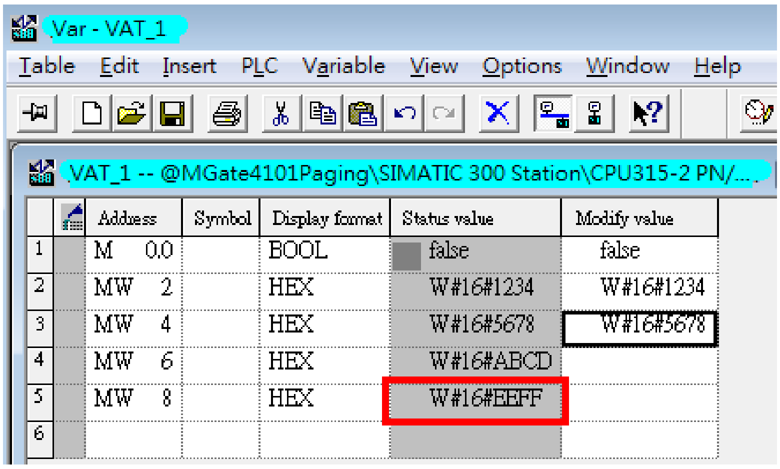
Change **Register 64 value** of Modbus server (slave) ID 2 to 0xEEFF.



ID = 2: F = 03

	00000	00010	00020	00030	00040	00050	00060	00(
0	0x5678	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0
1	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0
2	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0
3	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0
4	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0xEEFF	0x0
5	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0

You can see the **MW8** variable becomes 0xEEEF in the following figure.



VAT_1 -- @MGate4101Paging\SIMATIC 300 Station\CPU315-2 PN/...

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0		BOOL	false	false
2	MW 2		HEX	W#16#1234	W#16#1234
3	MW 4		HEX	W#16#5678	W#16#5678
4	MW 6		HEX	W#16#ABCD	
5	MW 8		HEX	W#16#EEFF	
6					