8.3 Modbus Protocol Communication

8.3.1 Introduction

GM7 built-in communication supports Modbus, the Modicon product's communication protocol. It supports ASCII mode, using ASCII data and RTU mode using Hex data. Function code used in Modbus is supported by function block and especially function code 01, 02, 03, 04, 05, 06, 15, and 16. Refer to "Modicon Modbus Protocol Reference Guide"(http://www.modicon.com/techpubs/toc7.html).

8.3.2 Basic Specification

1) ASCII mode

(1) It communicates, using ASCII data.
(2) Each frame uses ': (colon : H3A)', for header, CRLF (Carriage Return-Line Feed : H0D H0A), for tail.
(3) It allows Max. 1 second interval between characters.
(4) It checks errors, using LRC.
(5) Frame structure (ASCII data)

Item	Header	Address	Function code	Data	LRC	Tail (CR LF)
Size	1 byte	2 bytes	2 bytes	n bytes	2 bytes	2 bytes

2) RTU mode

(1) It communicates, using hex data.
(2) There's no header and tail. It starts with address and finishes frame with CRC.
(3) It has at least 3.5 character times between two frames.
(4) It ignores the current frame when 1.5 character times elapse between characters.
(5) It checks errors, using 16 bit CRC.
(6) Frame structure (hex data).

Item	Address	Function code	Data	CRC
Size	1 byte	1 bytes	n bytes	2 bytes

REMARK

1) The size constituting 1 letter is 1 character. So 1 character is 8 bits that is 1 byte.

2) 1 character time means the time lapsed for sending 1 character.

Ex) Calculation of 1 character time at 1200 bps.

1200 bps means that it takes 1 sec to send 1200 bits. To send 1 bit, 1 sec/1200 bits = 0.83 ms. Therefore 1 character time is 0.83ms * 8 bits = 6.64ms.

3) 584, 984 A/B/X executes frame division, using intervals of more than 1 sec without LRC in processing internally.

3) Address area

(1) Setting range is available from 1 to 247, but GM7 supports from 0 to 31.

(2) Address 0 is used for broadcast address. Broadcast address is all slave device recognize and respond to like the self-address, which can't be supported by GM7.

4) Function code area

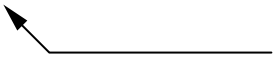
(1) GM7 supports only 01, 02, 03, 04, 05, 06, 15, and 16 among Modicon products' function codes.

(2) If the response format is confirm+(ACK), it uses the same function code.

(3) If the response format is confirm-(NCK), it returns as it sets the 8th bit of function code as 1.

Ex) If function code is 03, (we write here only function code part. Because only function codes are different.)

[Request]	0000 0011 (H03)
[Confirm+]	0000 0011 (H03)
[Confirm-]	**1**000 0011 (H83)



It returns as it sets the 8th bit of function code of request frame.

5) Data area

(1) It sends data, using ASCII data(ASCII mode) or hex (RTU mode).

(2) Data is changed according to each function code.

(3) Response frame uses data area as response data or error code.

6) LRC Check/CRC Check area

(1) LRC (Longitudinal Redundancy Check) : It works in ASCII mode. It takes 2' complement from sum of frame except header or tail to change into ASCII code,

(2) CRC (Cyclical Redundancy Check): It works in RTU mode. It uses 2-byte CRC check rules.

REMARK

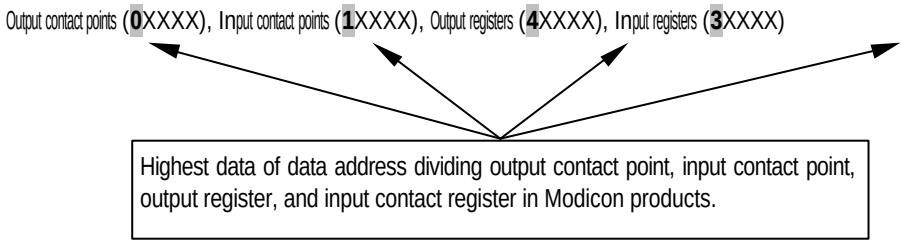
1) All numerical data can use hexadecimal, decimal, and binary type. If we convert decimal 7 and 10 into each type:
Hexadecimal : H07, H0A or 16#07, 16#0A
Decimal : 7, 10
Binary : 2#0111, 2#1010

7) Function code types and memory mapping

Code	Function code name	Modicon PLC Data address	GLOFA-mapping	Remark
01	Read Coil Status	0XXXX(bit-output)	%MX0~%MX9999	Read bits
02	Read Input Status	1XXXX(bit-input)	%MX0~%MX9999	Read bits
03	Read Holding Registers	4XXXX(word-output)	%MW0~%MW9999	Read words
04	Read Input Registers	3XXXX(word-input)	%MW0~%MW9999	Read words
05	Force Single Coil	0XXXX(bit-output)	%MX0~%MX9999	Write bit
06	Preset Single Register	4XXXX(word-output)	%MW0~%MW9999	Write word
15	Force Multiple Coils	0XXXX(bit-output)	%MX0~%MX9999	Write bits
16	Preset Multiple Registers	4XXXX(word-output)	%MW0~%MW9999	Write words

8) Modbus addressing rules

GM7 base unit starts its address from 0 and matches with 1 of Modicon products' data address. So GM7's address, n matches n+1 of Modicon products' address. Also, GM7 base unit has continuous M area without any division of output contact points (0XXXX), input contact points (1XXXX), output registers (4XXXX), input registers (3XXXX). This means that the output contact point 1 (0001) of Modicon products is marked as communication address 0 and the input contact point 1 (0001) of Modicon products is marked as communication address 0 in GM7.



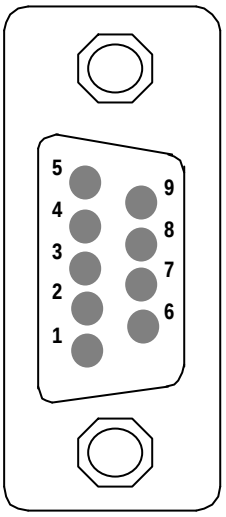
9) The size of using data

As for data size, GM7 base unit supports 128 bytes in ASCII mode and 256 bytes in RTU mode. The maximum size of the Modicon products is different from each other kind. So refer to "Modicon Modbus Protocol Reference Guide."

REMARK

1) GM7 base unit doesn't have any division between input and output area like Modicon PLC, when it supports Modbus protocol communication. It uses only M area. So the user must be caution to in set input and output area in M area for Modbus protocol communication.

10) Map of wiring



Male Type

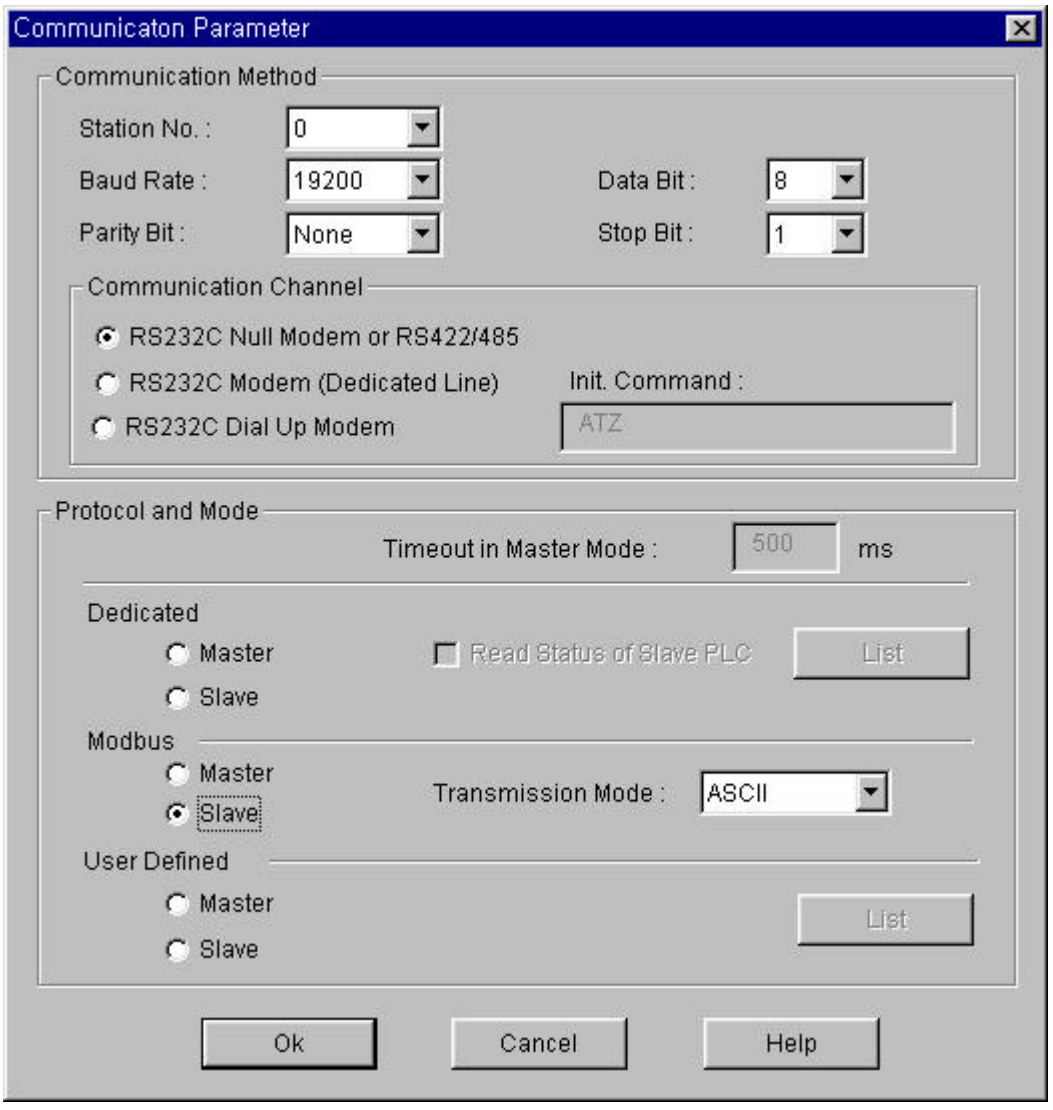
GM7 base unit	Connecting no. and direction	Quantum (9PIN)	
Pin no.		Pin no.	Signal
1		1	CD
2		2	RXD
3		3	TXD
4		4	DTR
5		5	SG
6		6	DSR
7		7	RTS
8		8	CTS
9		9	

8.3.3 Parameters Setting

1) Setting communication parameter

(1) Open a new project file at GMWIN.
- GM7 should be selected in PLC types.
- Open a new project file for each of the master and the slave.

(2) Select a communication parameter at GMWIN and double click to open the following window.

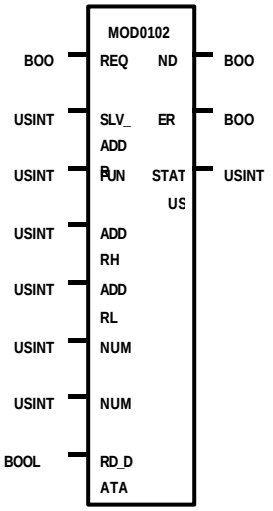


(3) Set the contents as follows.

Item	Setting contents
Station No.	Set a number between 0 to 31 (Don't assign no. 0 as broadcasting station lest it may be a cause for mistaken operation)
Baud Rate	Set one from 1200, 2400, 4800, 9600, 19200, 38400, or 57600 bps.
Data Bit	Set 7 or 8. ASCII mode: Set as 7 bits. RTU mode: Set as 8 bits.
Parity Bit	Set as one of None, Even, or Odd.
Stop Bit	Set 1 or 2 bit(s). When parity bit is set: Set as 1 bit. When parity bit isn't set: Set as 2 bits.
Communication Channel	● RS232C Null Modem or RS422/485 : It's a communication channel for the communication, using GM7 base unit's built-in communication and Cnet I/F module (G7L-CUEC). ● RS232C Modem (Dedicated Line) : It's to be selected for the communication, using an dedicated modem with Cnet I/F module (G7L-CUEB). ● RS232C Dial Up Modem : It's to be selected for the general communication connecting through the telephone line by dial up modem and Cnet I/F module (G7L-CUEB). Footnote) Using Cnet I/F module (G7L-CUEB) supporting RS232C, RS232C dedicated or dial-up modem communication can be done, but not through Cnet I/F module (G7L-CUEC) supporting RS422/485.
Time out in Master Mode	● It's the time waiting a responding frame since the master GM7 base unit sends a request frame. ● The default value is 500ms. ● It must be set in consideration of the max. periodical time for sending/receiving of the master PLC. ● If it's set smaller than the max. send/receive periodical time, it may cause communication error.
Modbus Master/ Slave	If it is set as the master, it's the subject in the communication system. If it's set as the slave, it only responds to the request frame of the master.
Transmission Mode	Select ASCII mode or RTU mode.

8.3.4 Function Block

1) MOD0102

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 01 and 02. ADDRH : High address of the starting addresses to be read from the slave station. ADDRL : Low address of the starting addresses to be read from the slave station. NUMH : High address of the data size of the starting address to be read from the slave station. NUML : Low address of the data size of the starting address to be read from the slave station. **Output** RD_DATA : A variable name to save the data that is read(The number of array is to be declared as same as or bigger than data size.). NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function
This is a function block that can execute either function code 01 or 02 for reading bits in Modbus protocol communication. Function code 01 reads Coil Status data and function 02 reads Input Status data.

(2) Error
It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) Example of the program
① It's supposed that GM7 base unit is the master and it reads Coil Status of the station no. 17, a Modicon product.

- The master reads status of the Coil 00020 ~ 00056 of the slave station no. 17. The Coil of the slave station is supposed to be as follows and the data that are read is saved in any array variable RD_DBD of the 40 sized BOOL type.

Coil	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40
Status	X	X	X	1	1	0	1	1	0	0	0	0	1	1	1	0	1	0	1	1
Hex	1				B				0				E				B			
Coil	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20
Status	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	1	1	0	1
Hex	2				6				B				C				D			

<Data status of the Modicon product' s Coil 00020-00059>

- The status of Coil 57, 58, 59 are redundancy.
- Data is sent starting from the low bit by byte unit. If the deficient bit of a byte is filled with 0. An example of sending the above data is as follows.

  Ex1) CD 6B B2 0E 1B

Function block input	Input value	
REQ	Enter the input condition to operate.	
SLV_ADDR	16#11 or 17	: The slave station number.
FUNC	16#01 or 1	: Enter ' 1 ' as the Coil Status is being read.
ADDRH	16#00 or 0	: High byte of the starting addresses to be read from the slave station.
ADDRL	16#13 or 19	: High byte of the starting addresses to be read from the slave station. - Read the no. 19 to read starting from the Coil 00020 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0	: High byte of which total data size to be read is expressed in hex.
NUML	16#25 or 37	: Low byte of which total data size to be read is expressed in hex. - Example is to be read 00020 ~ 00056, of which the total data size is 37. 37 are H0025 in hex. So H00 is input for NUHH and H25 for NUHL.

- Results

Variable	Value to save	Variable	Value to save	Variable	Value to save	Variable	Value to save
RD_DB0 [0]	1	RD_DB0 [10]	0	RD_DB0 [20]	1	RD_DB0 [30]	0
RD_DB0 [1]	0	RD_DB0 [11]	1	RD_DB0 [21]	1	RD_DB0 [31]	0
RD_DB0 [2]	1	RD_DB0 [12]	0	RD_DB0 [22]	0	RD_DB0 [32]	1
RD_DB0 [3]	1	RD_DB0 [13]	1	RD_DB0 [23]	1	RD_DB0 [33]	1
RD_DB0 [4]	0	RD_DB0 [14]	1	RD_DB0 [24]	0	RD_DB0 [34]	0
RD_DB0 [5]	0	RD_DB0 [15]	0	RD_DB0 [25]	1	RD_DB0 [35]	1
RD_DB0 [6]	1	RD_DB0 [16]	0	RD_DB0 [26]	1	RD_DB0 [36]	1
RD_DB0 [7]	1	RD_DB0 [17]	1	RD_DB0 [27]	1	RD_DB0 [37]	X
RD_DB0 [8]	1	RD_DB0 [18]	0	RD_DB0 [28]	0	RD_DB0 [38]	X
RD_DB0 [9]	1	RD_DB0 [19]	0	RD_DB0 [29]	0	RD_DB0 [39]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DB0[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

② It's supposed that GM7 base unit is the master and it reads Input Status of the station no. 17, a Modicon product.

- The master reads status of the Input 10197 ~ 10218 of the slave station no. 17. The Input of the slave station is supposed to be as follows and the data that are previously read is saved in any array variable RD_DB1 of the 25 sized BOOL type.

Input	1022C	1021B	1021A	10219	10218	10217	10216	10215
Status	X	X	1	1	0	1	0	1
Hex	3				5			
Input	10214	10213	10212	10211	10210	1020F	1020E	1020D
Status	1	1	0	1	1	0	1	1
Hex	D				B			
Input	1020C	1020B	1020A	10209	10208	10207	10206	10205
Status	0	1	0	1	1	0	1	1
Hex	A				C			

- Input coil 10219, 10220 are redundancy.
- Data is sent starting from the low bit by byte unit. If the deficient bit of a byte is filled with 0. An example of sending the above data is as follows.

Ex1) AC DB 35

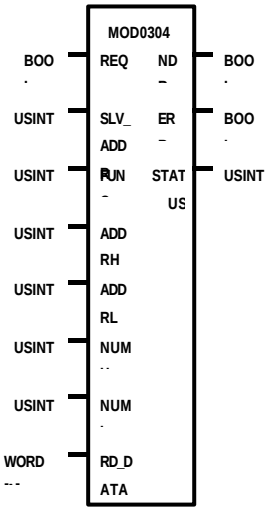
Function block input	Input value	
REQ	Enter the input condition to operate.	
SLV_ADDR	16#11 or 17	: The slave station number.
FUNC	16#02 or 2	: Enter '2' as the Input Status is being read.
ADDRH	16#00 or 0	: High byte of the starting addresses to be read from the slave station.
ADDRL	16#C4 or 196	: High byte of the starting addresses to be read from the slave station. - Read the no. 196 to read starting from the input coil 10197 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0	: High byte of which total data size to be read is expressed in hex.
NUML	16#16 or 22	: Low byte of which total data size to be read is expressed in hex. - Example is to read 10197-10218, of which the total data size is 22. 22 are H0016 in hex. So H00 is input for NUHH and H16 for NUHL.

- Results

Variable	Value to save	Variable	Value to save	Variable	Value to save	Variable	Value to save
RD_DB1 [0]	0	RD_DB1 [6]	0	RD_DB1 [12]	1	RD_DB1 [18]	1
RD_DB1 [1]	0	RD_DB1 [7]	1	RD_DB1 [13]	0	RD_DB1 [19]	0
RD_DB1 [2]	1	RD_DB1 [8]	1	RD_DB1 [14]	1	RD_DB1 [20]	1
RD_DB1 [3]	1	RD_DB1 [9]	1	RD_DB1 [15]	1	RD_DB1 [21]	1
RD_DB1 [4]	0	RD_DB1 [10]	0	RD_DB1 [16]	1	RD_DB1 [22]	X
RD_DB1 [5]	1	RD_DB1 [11]	1	RD_DB1 [17]	0	RD_DB1 [23]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DB1[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

2) MOD0304

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 03 and 04. ADDRH : High address of the starting addresses to be read from the slave station. ADDRL : Low address of the starting addresses to be read from the slave station. NUMH : High address of the data size of the starting address to be read from the slave station. NUML : Low address of the data size of the starting address to be read from the slave station. **Output** RD_DATA : A variable name to save the data that is read(The number of array is to be declared as same as or bigger in data size.). NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 03 or 04 for reading words in Modbus protocol communication. Function code 03 reads Holding Registers and function 04 reads Input Registers.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) Example of the program

① It's supposed that GM7 base unit is the master and it reads Holding Registers of the station no. 17, a Modicon product.

- The master reads the Holding Registers 40108 ~ 40110 of the slave station no. 17. The status of the Holding Registers of the slave station is supposed to be as follows and the previously read data are saved in any array variable RD_DWO of the 40 sized WORD type.

Holding Registers	40110	40109	40108
Register status	H0064	H0000	H022B

- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) 02 2B 00 00 00 64

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#03 or 3 : Enter '3' as the Holding Registers is being read.
ADDRH	16#00 or 0 : High byte of the starting addresses to be read from the slave station.
ADDRL	16#6B or 107 : High byte of the starting addresses to be read from the slave station. - Read the no. 107 to read starting from the output holding Registers 40108 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte of which total data size to be read is expressed in hex.
NUML	16#03 or 3 : Low byte of which total data size to be read is expressed in hex. - Example is to read 40108 ~ 40110, of which the total data size is 3. 3 are H0003 in hex. So H00 is input for NUHH and H03 for NUHL.

- Result

Variable	Value to save
RD_DW0 [0]	H002B or 555
RD_DW0 [1]	H0000 or 0
RD_DW0 [2]	H0064 or 100
RD_DW0 [3]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DW0[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

② It's supposed that GM7 base unit is the master and it reads output coil data of the station no. 17, a Modicon product.

- The master reads status of the input registers 30009 of the slave station no. 17. The input coil of the slave station is supposed to be as follows and the data that are read is saved in any array variable RD_DW1 of the 2-sized WORD type.

Input Register	30009
Register status	H000A

- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) 00 0A

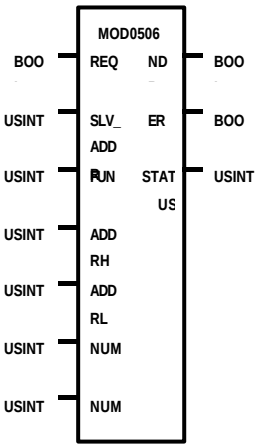
Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#04 or 4 : Enter '4' as the Input Registers is being read.
ADDRH	16#00 or 0 : High byte of the starting addresses to be read from the slave station.
ADDRL	16#08 or 8 : High byte of the starting addresses to be read from the slave station. - Read the no. 8 to read starting from the input Registers 30009 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte of which total data size to be read is expressed in hex.
NUML	16#01 or 1 : Low byte of which total data size to be read is expressed in hex. - Example is to read only 30009, of which the total data size is 1. 1 is H0001 in hex. So H00 is input for NUHH and H01 for NUHL.

- Results

Variable	Value to save
RD_DW1 [0]	H000A or 10
RD_DW1 [1]	X

- The variable to which saves the previously read data must be array type. The size of array type must be same as or bigger than the size of the data of read. If it's smaller, the error code is marked in STATUS.
- The previously read data is saved from the array variable, RD_DW1[0].
- The remnant part of an array variable is redundancy, after the variable is filled with the previously read data.

3) MOD0506

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 05 and 06. ADDRH : High address of the starting addresses to write on the slave station. ADDRL : Low address of the starting addresses to write on the slave station. NUMH : High address of the data size of the starting address to write on the slave station. NUML : Low address of the data size of the starting address to write on the slave station. **Output** NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 05 or 06 for writing 1 bit (function code 05) and writing 1 word (function code 06) in Modbus protocol communication. Function code 05 does 1 bit data writing on the Output Coil. If the Input NUMH is set as 255 (or HFF), it writes 1 on the output coil. If the Input NUMH is set as 0 (or H00), it writes 0 on the output coil. And function 06 does 1 word data writing on the Output Holding Register.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) An example of the program

① It's supposed that GM7 base unit is the master and it writes 1 bit data on the Coil of the station no. 17, a Modicon product.

② The master writes 1 on the Coil 00173 of the slave station no. 17, a Modicon product.

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#05 or 5 : Enter '5' as writes bit by bit on the Coil.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#AC or 172 : High byte of the starting addresses to write on the slave station. - Write on the no. 172 to write on, starting from the output coil 00173 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#FF or 255 : The data to be written on the slave station. - '0' → H00 - '1' → HFF
NUML	16#00 or 0 : The data to be written on the slave station. - As the example writes 1, enter HFF for HUMH and H00 for NUML

- Result : The Coil 00173 turns ON. (In case of GM7 base unit, 1 is saved on the related M area.)

Coil	00173
Status	1

③ It's supposed that GM7 base unit is the master and it writes on 1 words Holding Registers of the station no. 17, a Modicon product.

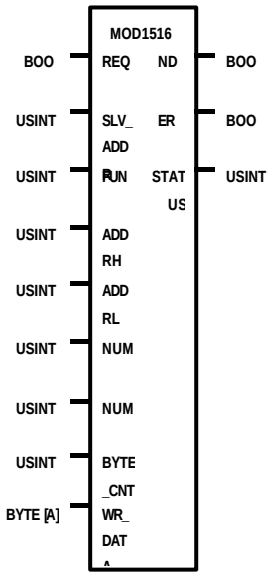
- An example of writing 3 on Holding Register 4002 of the station no. 17.

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#06 or 6 : Enter '6' as 1 word is being written on the Holding Register.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#01 or 1 : High byte of the starting addresses to write on the slave station. - Write on no. 1 to write on, starting from the Holding Register 40002 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#03 or 3 : High byte when the data to be written on is expressed in hex. - As the example writes 3, of which hex is H0003. So H00 is input for NUHH and H03 for NUHL.

- Result : The Holding Register 40002 is saved on H0003. (In case of GM7 base unit, H0003 is saved on a related M area.)

Holding Register	40002
Register status	H0003

4) MOD1516

Function block	Description
	**Input** REQ : Execute function block when it's 1(rising edge). SLV_ADDR : Input the number of the slave station. FUNC : Input the function code. It supports function code 15 and 16. ADDRH : High address of the starting addresses to write on the slave station. ADDRL : Low address of the starting addresses to write on the slave station. NUMH : High address of the data size of the starting address to write on the slave station. NUML : Low address of the data size of the starting address to write on the slave station. BYTE_CNT : The size of data to be written on the slave station. WR_DATA : Variable name to save the data to be written(The number of array is to be declared as same as or bigger in data size.). **Output** NDR : If it ends without error, output 1 and keep the value till the call for the next function block. ERR : If an error occurs, output 1 and keep the value till the call for the next function block. STATUS : When an error occurs, output an error code.

(1) Function

This is a function block that can execute either function code 15 or 16 for writing 1 bit (function code 15) and writing 1 word (function code 16) in Modbus protocol communication. Function code 15 does 1 bit by 1 bit data writing on each Coil in a sequence Coils. And function 16 does 1 word by 1 word data writing on sequence of the Holding Registers.

(2) Error

It outputs error codes to output STATUS. Refer to "Error codes" for the detailed.

(3) An example of the program

① It's supposed that GM7 base unit is the master and it writes bits continually on the output coil of the station no. 17, a Modicon product.

- The master writes continual 10 bits, 01110011011 on the Coils 00020 of the slave station no. 17 1 bit by 1 bit. The data that is to be written are saved in any array variable WR_DB0 of the 2 sized BYTE type.

Variable	Value to save
WR_DB0 [0]	2#11001101 또는 16#CD
WR_DB0 [1]	2#10000001 또는 16#81

- The size of BYTE_CNT is the same as when the data to be written are converted by byte. The above data are 10 by 1 bit. They can't be filled by 1 byte. So they must be filled from the low bit, using 2 bytes. And 0 fills the remnant 6 bits. Therefore the size of BYTE_CNT is 2.
- If it is supposed that data of 1000 0001 1100 1101 are saved in the array variable, WR_DB0, the data are sent as 10 bits (01 1100 1101) at the bottom plus 6 bit of 0 at the top. For the size of the data is set as 10 bits to send and they are sent by bytes, the deficient 6 bits are filled with 0.
- Data is sent starting from the low bit by byte unit. An example of sending the above data is as follows.

  Ex1) CD 01

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#0F or 15 : Enter '15' as bits are continually written on the output coils.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#13 or 19 : Low byte of the starting addresses to write on the slave station. - Write on No. 19 to write on, starting from the output holding Registers 00020 in accordance with the previous no. 8) "Modbus addressing rules." And the highest data of the data address doesn't need to be input. Because it's automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#0A or 10 : Low byte when the data to be written on is expressed in hex. - For the data of the example are 10 bits continued from 00020, their size is 10, which is H000A in hex. So H00 is input for NUHH and H0A for NUHL.
BYTE_CNT	16#02 or 2 : The size of the data to be written on the slave station, when they are converted by bytes. - For the data of the example are continual 10 bits, which is converted to 2 bytes. Therefore input H02 for BYTE_CNT.

- Result
  From the 2 bytes (16 bits) sent, only the low 10 bits are valid as set for its size.

Coil	00029	00028	00027	00026	00025	00024	00023	00022	00021	00020
Status	0	1	1	1	0	0	1	1	0	1

② It's supposed that GM7 base unit is the master and it writes word data continually on the Holding Registers of the station no. 17, a Modicon product.

- The master writes 000A and 0102 on the Holding Registers 40002 of the slave station no. 17. The data that is to be written are saved in any array variable WR_DB1 of the 4 sized BYTE type.

Variable	Value to save
WR_DB1 [0]	2#00001010 or 16#0A
WR_DB1 [1]	2#00000000 or 16#00
WR_DB1 [2]	2#00000010 or 16#02
WR_DB1 [3]	2#00000001 or 16#01

- The size of BYTE_CNT is the same as when the data to be written are converted by byte. The above data are 2 words that need 4 bytes. Therefore the size of BYTE_CNT is 4.
- Data is sent starting from the low word by byte unit. An example of sending the above data is as follows.

  Ex1) 00 0A 01 02

Function block input	Input value
REQ	Enter the input condition to operate.
SLV_ADDR	16#11 or 17 : The slave station number.
FUNC	16#10 or 16 : Enter ‘16’ as words are continually written on the Holding Registers.
ADDRH	16#00 or 0 : High byte of the starting addresses to write on the slave station.
ADDRL	16#01 or 1 : Low byte of the starting addresses to write on the slave station. - Write on no. 1 to write on, starting from the output holding Registers 40002 in accordance with the previous no. 8) “Modbus addressing rules.” And the highest data of the data address doesn’t need to be input. Because it’s automatically processed by the input value of the input FUNC of the function block.
NUMH	16#00 or 0 : High byte when the data to be written on is expressed in hex.
NUML	16#02 or 2 : Low byte when the data to be written on is expressed in hex. - For the data of the example are 2-word data continued from 40002, their size is 2, which is H0002 in hex. So H00 is input for NUHH and H02 for NUHL.
BYTE_CNT	16#04 or 4 : The size of the data to be written on the slave station, when they are converted by bytes. - For the data of the example are continual 2word data, which is converted to 4 bytes. Therefore input H04 for BYTE_CNT.

- Result

Holding Registers	40003	40002
Registers status	H0102	H000A

5) Error code

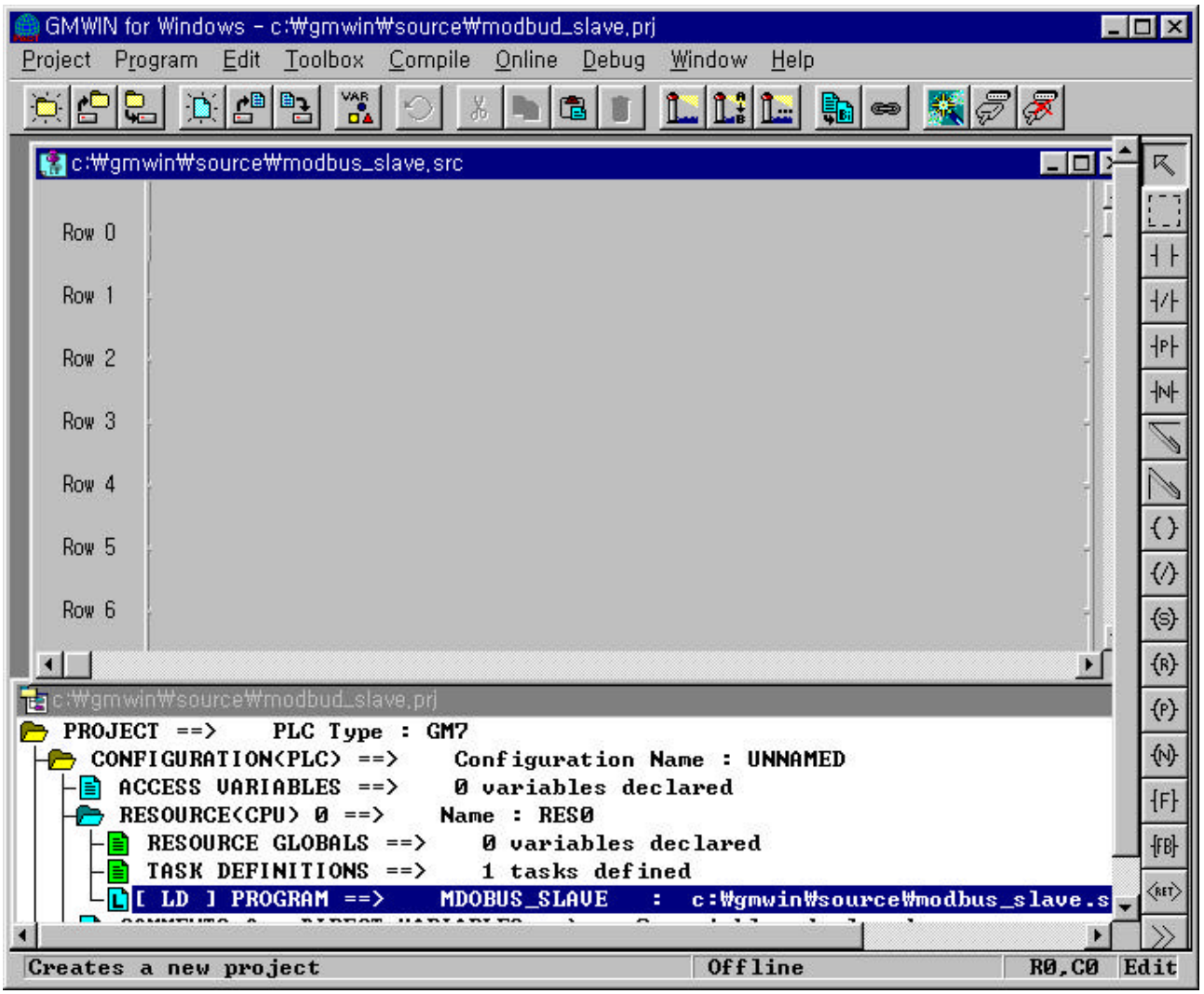
CODE	Error type	Meaning
01	Illegal Function	Error in inputting function code in function block.
02	Illegal Address	Error of exceeding the area limit of reading/writing on the slave station.
03	Illegal Data Value	Error when the data value to be read from or write on the slave station isn't allowed.
04	Slave Device Failure	Error status of the slave station.
05	Acknowledge	It's a responding code of the slave station for the master station to prevent the master stations time-out error, when request command processing takes time. The master station marks an error code and waits for a certain time without making any second request.
06	Slave Device Busy	Error when request command processing takes too much time. The master should request again.
07	Time Out	Error when exceeds the time limit of the communication parameter as it communicates.
08	Number Error	Errors when data is 0 or more than 256 bytes, when the data size is bigger than the array size, or when Number and BYTE_CNT are different from each other.
09	Parameter Error	Error of setting parameters (mode, master/ slave)
10	Station Error	Error when the station number of itself and the station number set by the input parameter of the function block are the same.

8.3.5 Example of Use

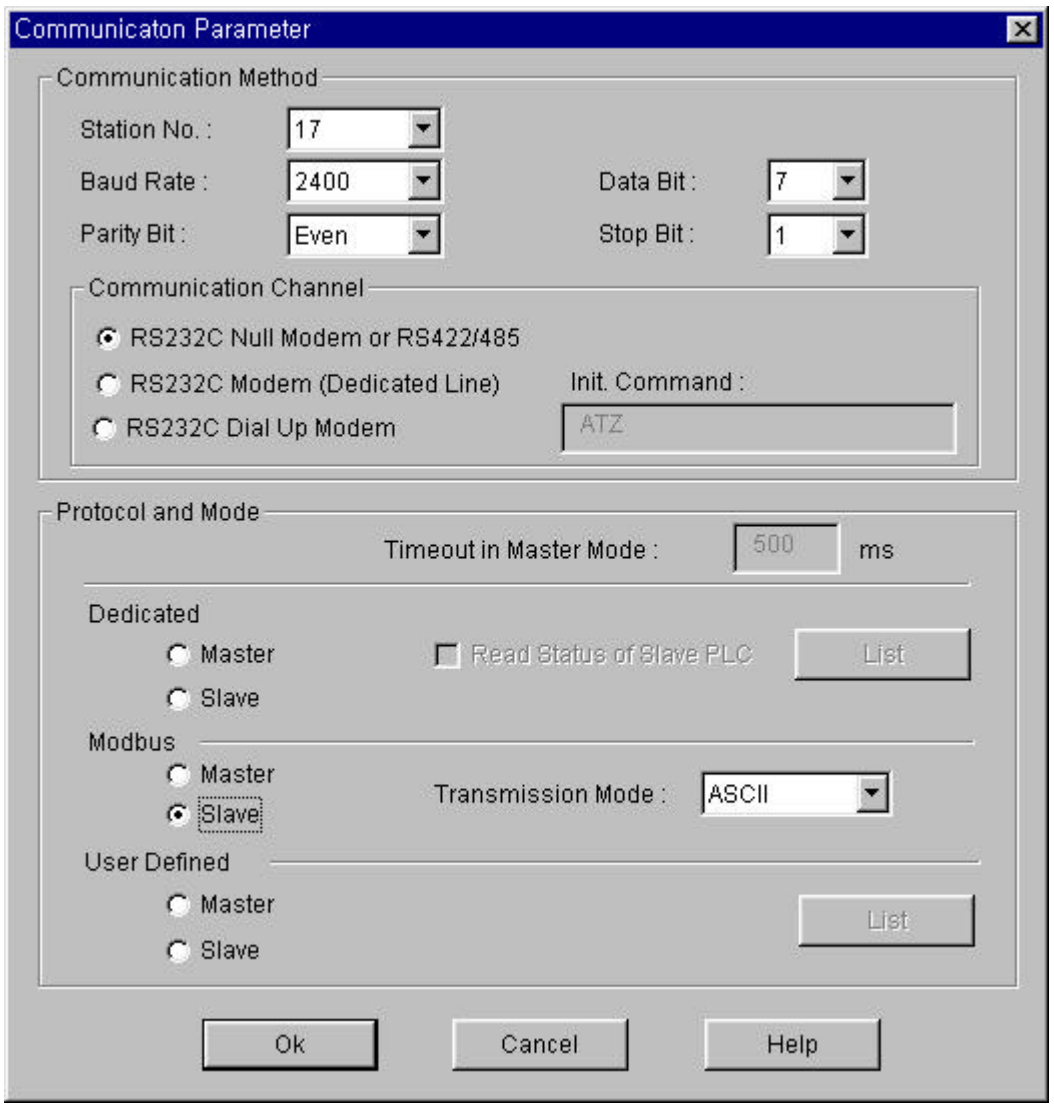
- According to the setting of device supporting Modbus protocol, setting of GM7 basic unit is changed, but this example explains Modbus protocol communication among GM7 units.
- The slave station program: This outputs the received data saved in M area through the output coil.
- The master station program : It saves 16#FF (or 255) at %MW0 (It is coincided with %MX0 ~ MX15 or %MB0 ~ %MB1) in function block MOD0506 (function code 06), then reads %MX0 through MOD0102 (function code 01), and again saves 0 at %MX0 ~ %MX9 using function block MOD1516 (function code 15), then reads %MW0 through Mod0304.
- The cable used in this example is same with that used for the dedicated protocol communication between GM7's.

1) Setting and the program of the slave station

(1) Open a new project file and a new program for the slave station.



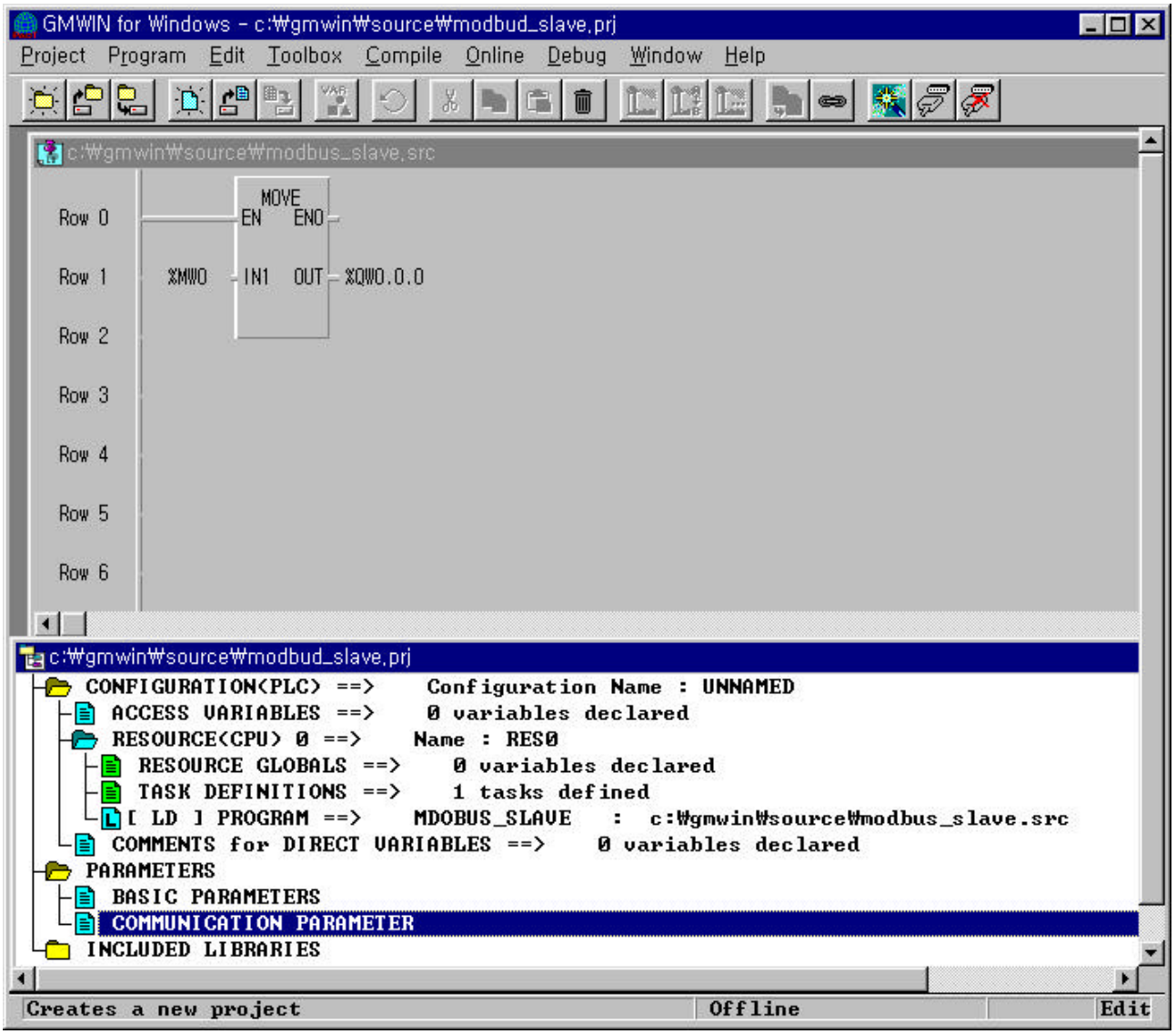
(2) Select communication parameter in GMWIN parameter and double-click on it, the window of communication parameter opens.



- Set parameters as the following table.

Communication Method						Protocol and Mode	
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication Channel	Modbus	Transmission Mode
17	2400	7	Even	1	RS232C Null Modem or RS422/485	Slave	ASCII

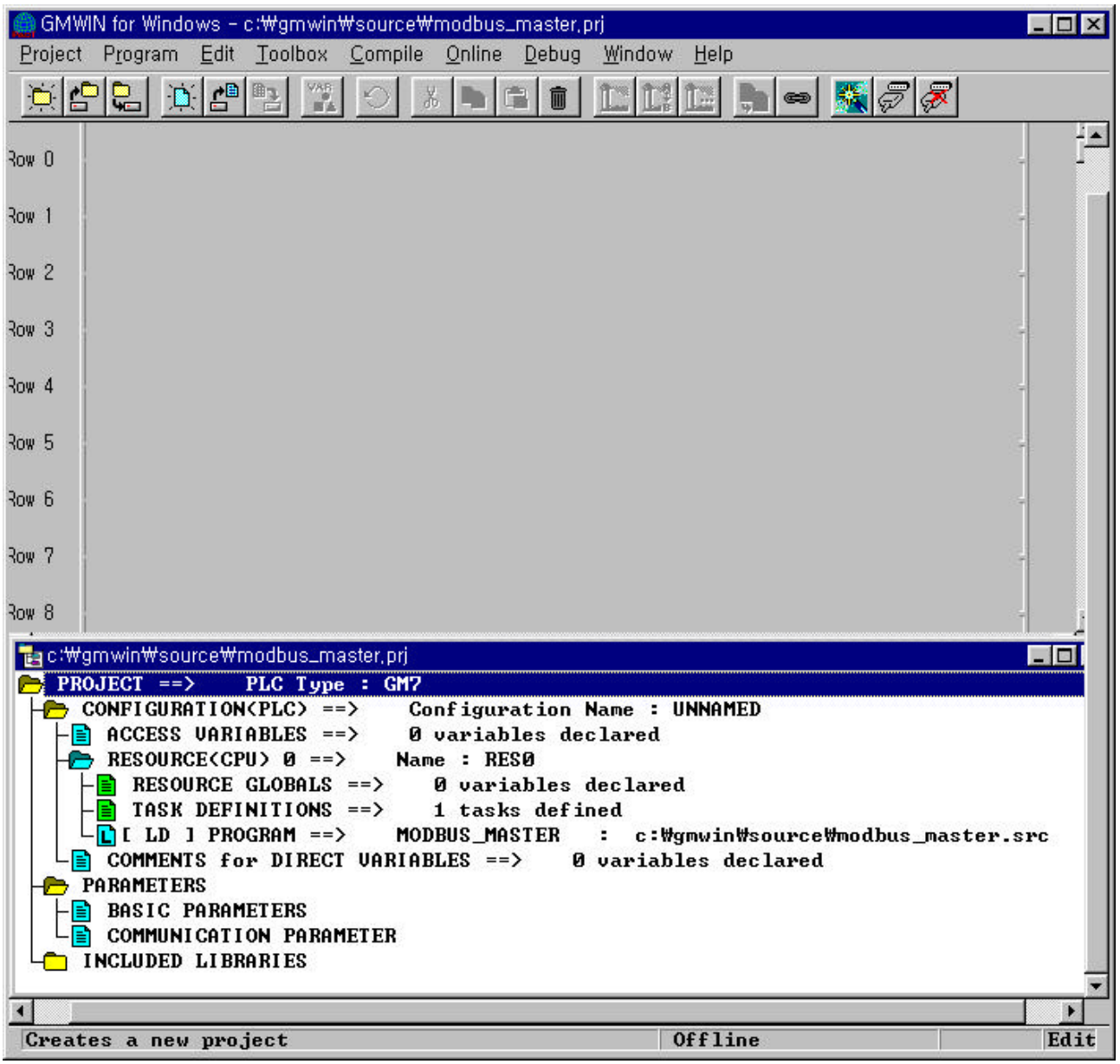
(3) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.



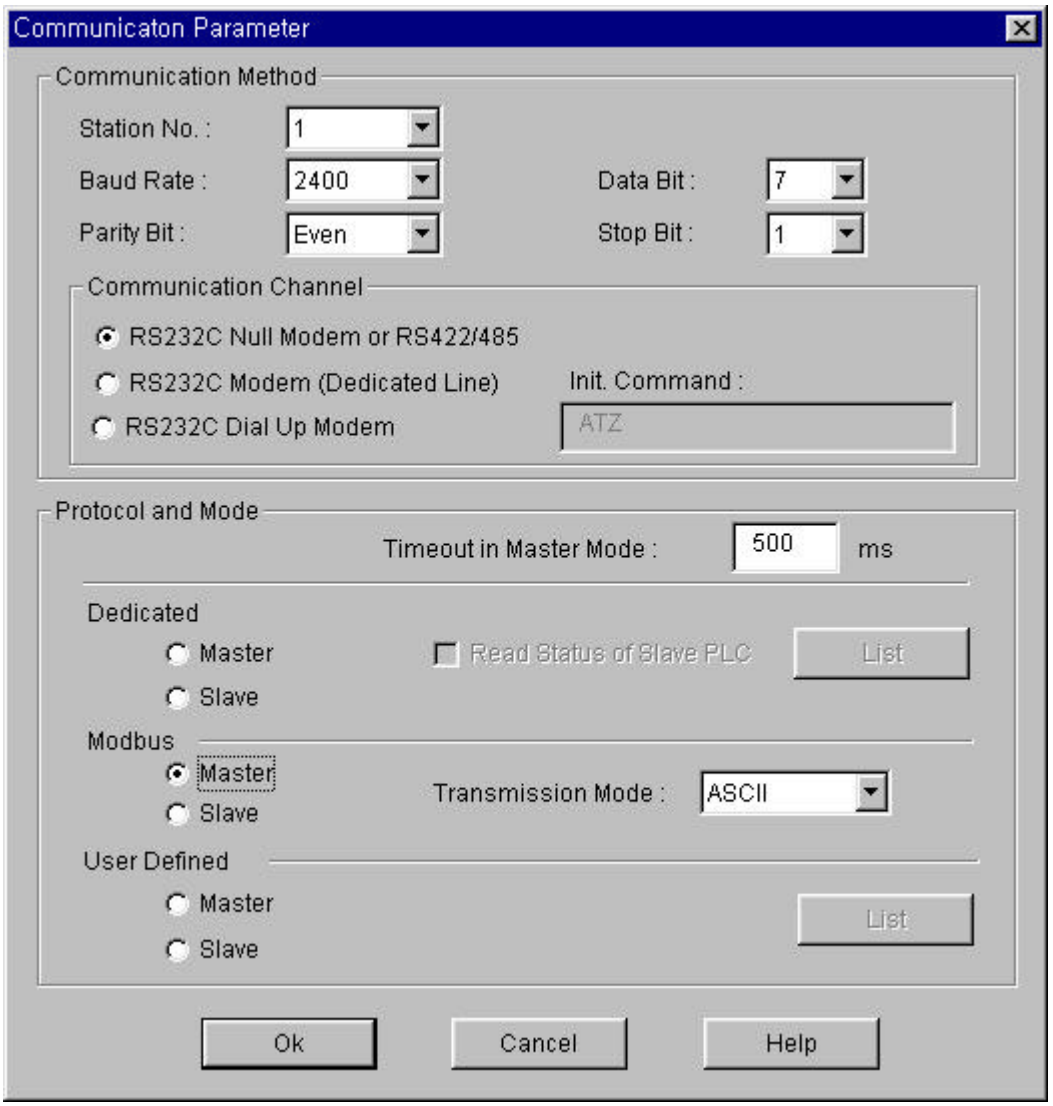
- The program of the slave is to output the data at M area to the output contact coil.

2) Setting and the program of the master station

(1) Make a new project file and a new program for the master station.



(2) If you double click the 'communication parameter' window in GMWIN, you can see the following window of the 'communication parameter.'

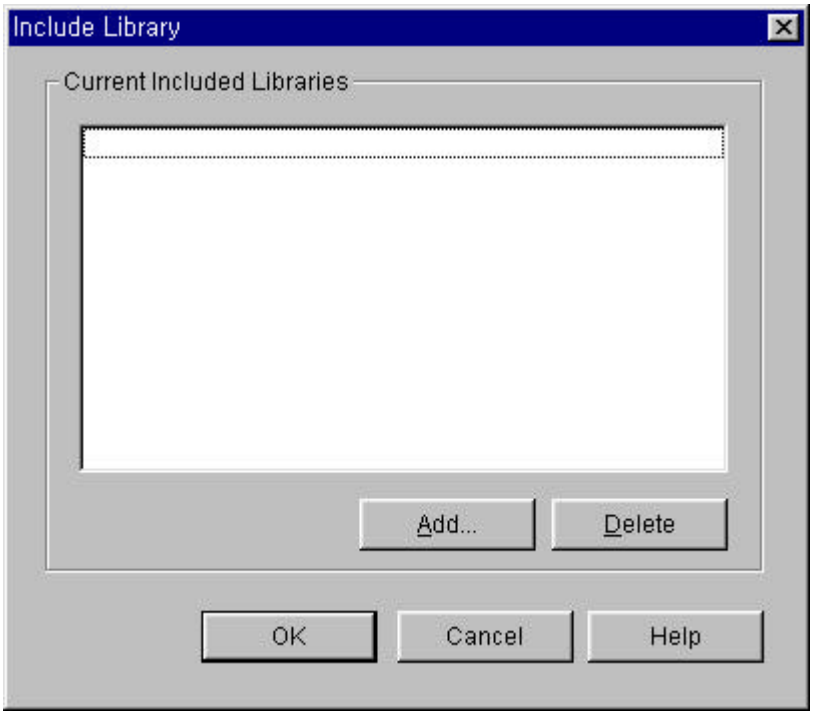


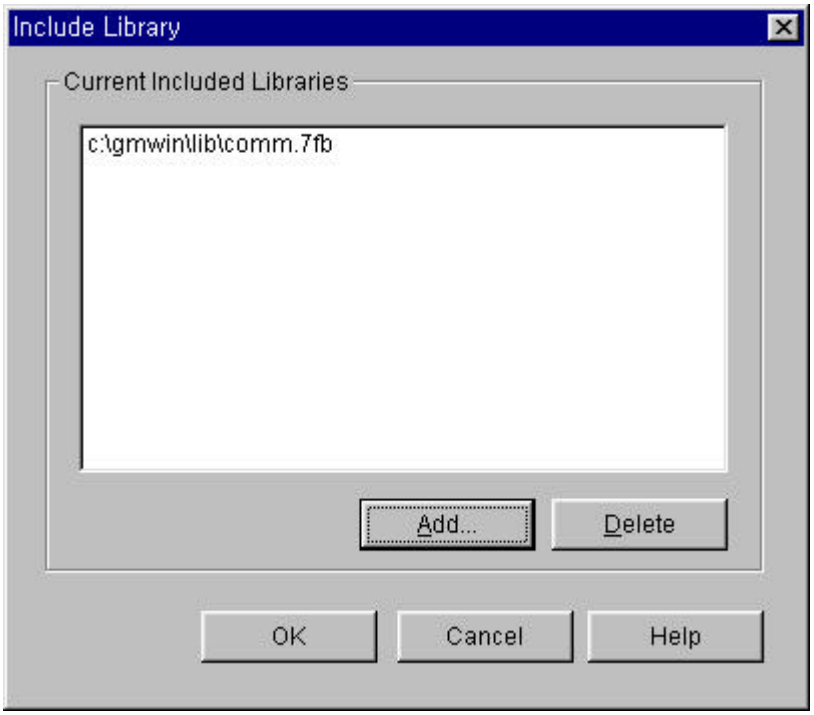
- Set parameters as the following table.

Communication Method						Protocol and Mode	
Station No.	Baud Rate	Data Bit	Parity Bit	Stop Bit	Communication Channel	Modbus	Transmission Mode
1	2400	7	Even	1	RS232C Null Modem or RS422/485	Master	ASCII

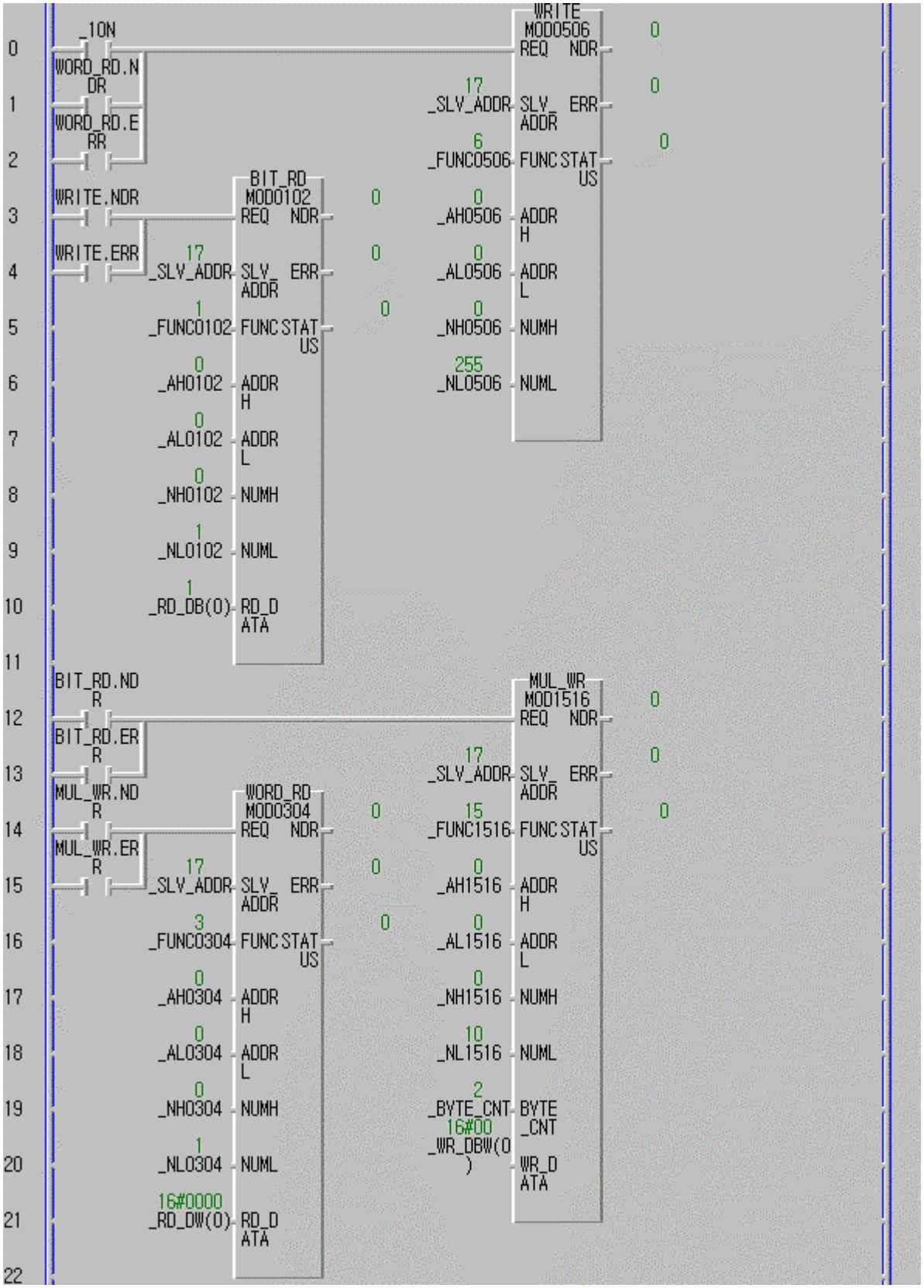
(3) Set up a program like the following figure and download to the slave station GM7. For the detailed program setting and downloading, refer to GMWIN manual.

- Function block is used in the program. Before using function block, double click "Current included Libraries" to open the following window. Click "Add" to add COMM.7FB and click "OK."





- Framing program



- It saves 16#FF (or 255) at %MW0 (It is coincided with %MX0 ~ %MX15 or %MB0 ~ %MB1) in function block MOD0506 (function code 06), then reads %MX0 through MOD0102 (function code 01), and again saves 0 at %MX0 ~ %MX9 using function block MOD1516 (function code 15), then reads %MW0 through Mod0304.
- Then you will see that 8 LEDs of output contact point operate on/off continually.
- The above picture is the monitored scene of the program operation. Therefore the values appeared in _RD_DB, _RD_DW, array variables are not the initial ones, but the resulted value after executing reading.
- Variables like Instanse name.NDR, Instanse name.ERR, Instanse name.STATUS are automatically generated when an instanse variable of function block is declared.
- _1ON flag is a flag that is on for 1 scan.
- Each function block input REQ conditions to make separate function block output as its input.
- The size of _BYTE_CNT must be the same when it is converted into bytes.
- Error occurs when the size of array variable is smaller than the data to be read or to be written.

- Table of variables

Variable	Variable type	Initial value	Variable	Variable type	Initial value
_SLV_ADDR	USINT	17(H11)	_NH0102	USINT	0(H00)
_FUNC0102	USINT	1(H01)	_NH0304	USINT	0(H00)
_FUNC0304	USINT	3(H03)	_NH0506	USINT	0(H00)
_FUNC0506	USINT	6(H06)	_NH1516	USINT	0(H00)
_FUNC1516	USINT	15(H0F)	_NL0102	USINT	1(H01)
_AH0102	USINT	0(H00)	_NL0304	USINT	255(HFF)
_AH0304	USINT	0(H00)	_NL0506	USINT	1(H01)
_AH0506	USINT	0(H00)	_NL1516	USINT	10(H0A)
_AH1516	USINT	0(H00)	_RD_DB	BOOL-typed ARRAY [40]	{0,0,0}
_AL0102	USINT	0(H00)	_RD_DW	WORD-typed ARRAY [4]	{0,0,0,0}
_AL0304	USINT	0(H00)	_WR_DBW	BYTE-typed ARRAY [4]	{0,0,0,0}
_AL0506	USINT	0(H00)	_BYTE_CNT	USINT	2(H02)
_AL1516	USINT	0(H00)			