

# DM500-Modbus Manual

---

V1.1

Introduction .....	2
Modbus Macro Operation .....	3
1.1 Write operation .....	3
1.1.1 #1200 (send data buffer start address) .....	3
1.1.2 #1201 (serial port number) .....	3
1.1.3 #1202 (slave ID) .....	3
1.1.4 #1203 (start address of write data) .....	3
1.1.5 #1204 (write data byte length) .....	3
1.1.6 #1205 (configuration macro) .....	3
1.1.7 #1206 (write operation status return value) .....	4
1.1.8 #1209 (trigger write operation) .....	4
1.1.9 Examples 1 (0FH) .....	4
1.1.10 Examples 2 (10H) .....	5
1.2 Read operation .....	7
1.2.1 #1210 (receive data buffer start address) .....	7
1.2.2 #1211 (serial port number) .....	7
1.2.3 #1212 (slave ID) .....	7
1.2.4 #1213 (start address of read data) .....	7
1.2.5 #1214 (read data byte length) .....	7
1.2.6 #1215 (configuration macro) .....	7
1.2.7 #1216 (read operation status return value) .....	8
1.2.8 #1219 (Execute read operation macro address) .....	8
1.2.9 Example1 (01H) .....	9
1.2.10 Example 2(02H) .....	11
1.2.11 Example 3 (03H) .....	13
1.2.12 Example 4 (04H) .....	15
Access route .....	17

# Introduction

The DM500 controller supports the modbus RTU communication protocol;

- Interface: serial port;
- Level: RS232
- The default baud rate is 9600;
- Master-slave mode: default host mode
- Support function codes: 01H, 02H, 03H, 04H, 0FH, 10H;

If you need to cascade multiple modbus slave devices, you need to use a 232-485 adapter board.

# Modbus Macro Operation

## 1.1 Write operation

### 1.1.1 #1200 (send data buffer start address)

Send data buffer start macro address;

For example: #1200 =1230

#1230 = 10 (data buffer byte 1)

#1231 = 12 (data buffer byte 2)

#1232 = 20 (data buffer byte 3)

.....

### 1.1.2 #1201 (serial port number)

Serial communication port number, DM500 has only one serial port, the default is 0;

### 1.1.3 #1202 (slave ID)

Slave device ID number;

### 1.1.4 #1203 (start address of write data)

Slave register start address;

### 1.1.5 #1204 (write data byte length)

0FH: #1204 indicates the number of coils

10H: #1204 indicates the number of bytes

### 1.1.6 #1205 (configuration macro)

Unsigned 32 bit quantity

Features	Bit	description
type of data	[7:0]	0x00: byte    0x01: Bit
Read and write mode	[15:8]	0x00: R/W    0x01: Read only
Communication mode	[31:16]	0x0000: RTU    0x0001: ASCII

Write operation function code allocation table (RTU mode)

function code	[7:0]	[15:8]	[31:16]
0FH	0x01	0x00	0x0000
10H	0x00	0x00	0x0000

### 1.1.7 #1206 (write operation status return value)

Exception response code list:

return value	description
0x00	normal
0x01	Invalid or unsupported function code
0x02	Invalid or unsupported address
0x03	Invalid or unsupported data
0x04	Action execution failed
0x05	Action execution (may take a long time)
0x06	The device is busy and cannot perform the action temporarily.
0x08	File data verification error
0x0A	Invalid gateway path
0x0B	Target device is not responding
0xE0	Transmission error or illegal modbus data frame
0xFF	time out
0xe1	Undefined action

Read #1206 when #1209 changes from 1 to 0;

### 1.1.8 #1209 (trigger write operation)

Read and write

#1209 = 1 : Perform a write operation;

If #1209 == 0, the write action is completed;

### 1.1.9 Examples 1 (0FH)

G04 P0

#1200 = 1230 (send data buffer address)

#1201 = 0 (communication port 0)

#1202 = 1 (slave ID)

#1203 = 6 (slave start address)

#1204 = 12 (number of coils)

#1205 = 1 (communication mode: 0FH, RTU)

#1230 = 3 (byte 1)

#1231 = 4 (byte 2)

#1209 = 1 (trigger sending)

WHILE [#1209 NE 0] DO4 (waiting for completion)

G04P0

END4

The data frame is as follows:

01 0F 00 06 00 0C 02 03 04 E4 E5

domain name	Hex
ID	01
Features	0F
Starting address Hi	00
Starting address Lo	06
Output quantity Hi	00
Output quantity Li	0C
Number of bytes	02
Output value Hi	03
Output value Lo	04
CRC Hi	E4
CRC Lo	E5

### 1.1.10 Examples 2 (10H)

G04 P0

#1200 = 1230 (send data buffer address)

#1201 = 0 (communication port 0)

#1202 = 1 (slave ID)

#1203 = 5 (slave start address)

#1204 = 4 (number of bytes)

#1205 = 0 (communication mode: 10H, RTU)

```

#1230 = 7          (byte 1)
#1231 = 8          (byte 2)
#1232 = 9          (byte 3)
#1233 = 10         (byte 4)
#1209 = 1          (trigger sending)
WHILE [#1209 NE 0] DO4  (waiting for completion)
G04P0
END4

```

The data frame is as follows:

01 10 00 05 00 02 04 08 07 0A 09 46 97

domain name	Hex
ID	01
Features	10
Starting address Hi	00
Starting address Lo	05
Number of registers Hi	00
Number of registers Li	02
Number of bytes	04
Register value Hi	08
Register value Lo	07
Register value Hi	0A
Register value Lo	09
CRC Hi	46
CRC Lo	97

## 1.2 Read operation

### 1.2.1 #1210 (receive data buffer start address)

Send data buffer start macro address;

E.g: #1200 = 1240

For example: #1210 = 1240

#1240 (receive buffer byte 1)

#1241 (receive buffer byte 2)

#1242 (receive buffer byte 3)

.....

### 1.2.2 #1211 (serial port number)

(same as #1201)

### 1.2.3 #1212 (slave ID)

(same as #1202)

### 1.2.4 #1213 (start address of read data)

Slave register start address;

### 1.2.5 #1214 (read data byte length)

01H, 02H: #1204 indicates the number of coils or discrete quantities

03H, 04H: #1204 indicates the number of bytes (number of registers \* 2)

### 1.2.6 #1215 (configuration macro)

Unsigned 32 bit quantity

Features	Bit	description
type of data	[7:0]	0x00: byte    0x01: Bit
Read and write mode	[15:8]	0x00: R/W    0x01: Read only
Communication mode	[31:16]	0x0000: RTU    0x0001: ASCII



Read operation function code allocation table (RTU mode)

function code	[7:0]	[15:8]	[31:16]
01H	0x01	0x00	0x0000
02H	0x01	0x01	0x0000
03H	0x00	0x00	0x0000
04H	0x00	0x01	0x0000

### 1.2.7 #1216 (read operation status return value)

Exception response code list:

return value	description
0x00	normal
0x01	Invalid or unsupported function code
0x02	Invalid or unsupported address
0x03	Invalid or unsupported data
0x04	Action execution failed
0x05	Action execution (may take a long time)
0x06	The device is busy and cannot perform the action temporarily.
0x08	File data verification error
0x0A	Invalid gateway path
0x0B	Target device is not responding
0xE0	Transmission error or illegal modbus data frame
0xFF	time out
0xe1	Undefined action

Read #1216 when #1219 changes from 1 to 0;

### 1.2.8 #1219 (Execute read operation macro address)

Read and write

#1219 = 1 : Perform a write operation;

If #1219 == 0, the write action is completed;

### 1.2.9 Example1 (01H)

Example of requesting to read discrete output 20-38:

G04 P0

#1210 = 1240 (send data buffer address)

#1211 = 0 (communication port 0)

#1212 = 1 (slave station number)

#1213 = 19 (slave coil start address 0x13)

#1214 = 19 (number of coils 0x13)

#1215 = 1 (communication mode 01H, RTU)

#1219 = 1 (trigger sending)

WHILE [#1219 NE 0] DO4 (waiting for completion)

G04P0

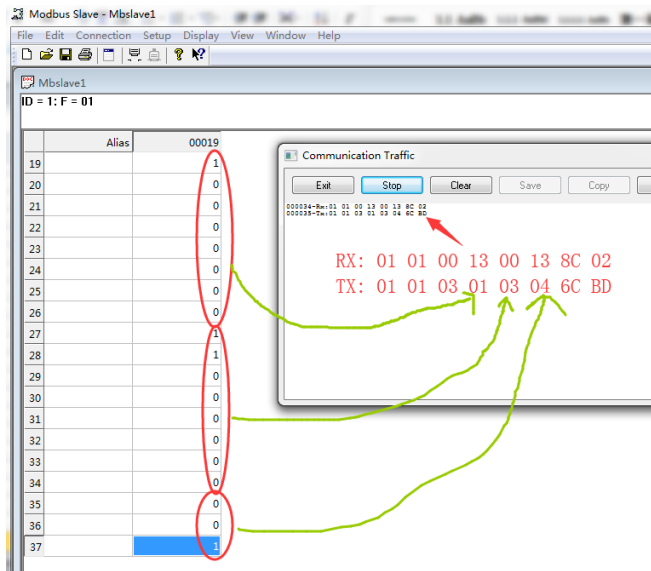
END4

The data frame is as follows:

RX: 01 01 00 13 00 13 8C 02

TX: 01 01 03 01 03 04 6C BD

The DM500 communicates with the computer virtual slave:



request	
domain name	HEX
ID	01
Features	01
Starting address Hi	00
Starting address Lo	13
Output quantity Hi	00
Output quantity Li	13
CRC Hi	8c
CRC Lo	02

response	
domain name	HEX
ID	01
Features	01
Number of bytes	03
Output Status 27-20 [#1240]	01
Output Status 35-28 [#1241]	03
Output Status 38-36 [#1242]	04
CRC Hi	6C
CRC Lo	BD

### 1.2.10 Example 2(02H)

Request to read an instance of discrete input 197-216:

G04 P0

#1210 = 1240 (send data buffer address)  
#1211 = 0 (communication port 0)  
#1212 = 1 (slave station number)  
#1213 = 196 (slave discrete start address)  
#1214 = 20 (discrete number)  
#1215 = 257 (communication mode 02H, RTU)  
#1219 = 1 (trigger sending)  
WHILE [#1219 NE 0] DO4 (waiting for completion)

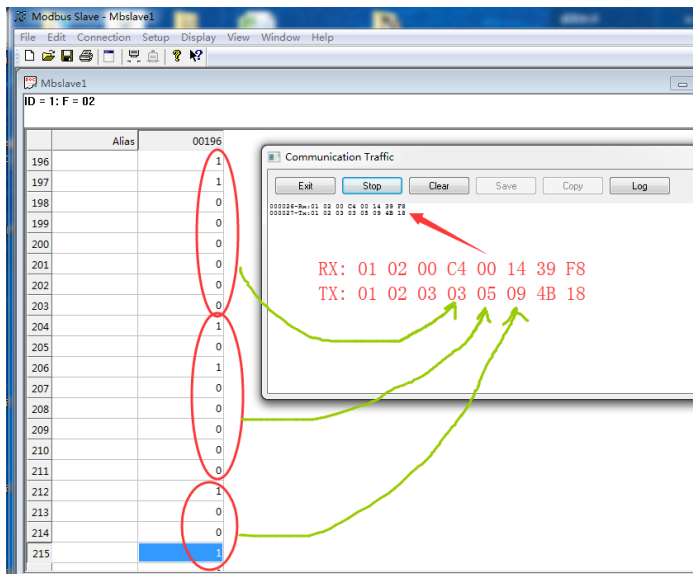
G04P0

The END4 data frame is as follows:

RX: 01 02 00 C4 00 14 39 F8

TX: 01 02 03 03 05 09 4B 18

The DM500 communicates with the computer virtual slave:



request	
domain name	HEX
ID	01
Features	02
Starting address Hi	00
Starting address Lo	C4
Output quantity Hi	00
Output quantity Li	14
CRC Hi	39
CRC Lo	F8

response	
domain name	HEX
ID	01
Features	02
Number of bytes	03
Output Status 204-197 [#1240]	03
Output Status 212-205 [#1241]	05
Output Status 216-213 [#1242]	09
CRC Hi	4B
CRC Lo	18

### 1.2.11 Example 3 (03H)

An example of reading registers 108-110:

G04 P0

#1210 = 1240 (send data buffer address)

#1211 = 0 (communication port 0)

#1212 = 1 (slave station number)

#1213 = 107 (slave register start address)

#1214 = 6 (number of bytes, number of registers = (#1214/2))

#1215 = 0 (communication mode 03H, RTU)

#1219 = 1 (trigger sending)

WHILE [#1219 NE 0] DO4 (waiting for completion)

G04P0

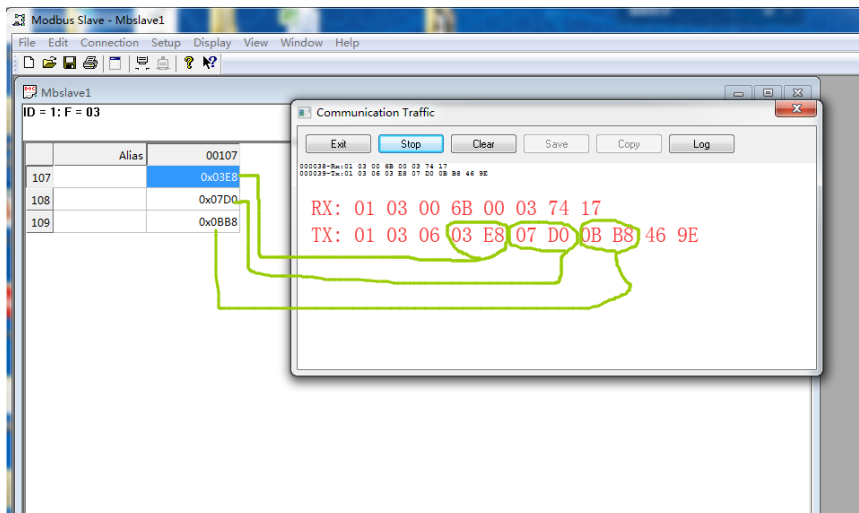
END4

The data frame is as follows:

RX: 01 03 00 6B 00 03 74 17

TX: 01 03 06 03 E8 07 D0 0B B8 46 9E

The DM500 communicates with the computer virtual slave:



request	
domain name	HEX
ID	01
Features	03
Starting address Hi	00
Starting address Lo	6B
Number of registers Hi	00
Number of registers Li	03
CRC Hi	74
CRC Lo	17

response	
domain name	HEX
ID	01
Features	03
Number of bytes	06
Register Value Hi(108) [#1241]	03
Register Value Lo(108) [#1240]	E8
Register Value Hi(109) [#1243]	07
Register Value Lo(109) [#1242]	D0
Register Value Hi(109) [#1245]	0B
Register Value Lo(109) [#1244]	B8
CRC Hi	46
CRC Lo	9E

### 1.2.12 Example 4 (04H)

Request to read an example of input register 9:

G04 P0

#1210 = 1240 (send data buffer address)

#1211 = 0 (communication port 0)

#1212 = 1 (slave station number)

#1213 = 08 (slave register start address)

#1214 = 2 (number of bytes, number of registers = (#1214/2))

#1215 = 256 (communication mode 04H, RTU)

#1219 = 1 (trigger sending)

WHILE [#1219 NE 0] DO4 (waiting for completion)

G04P0

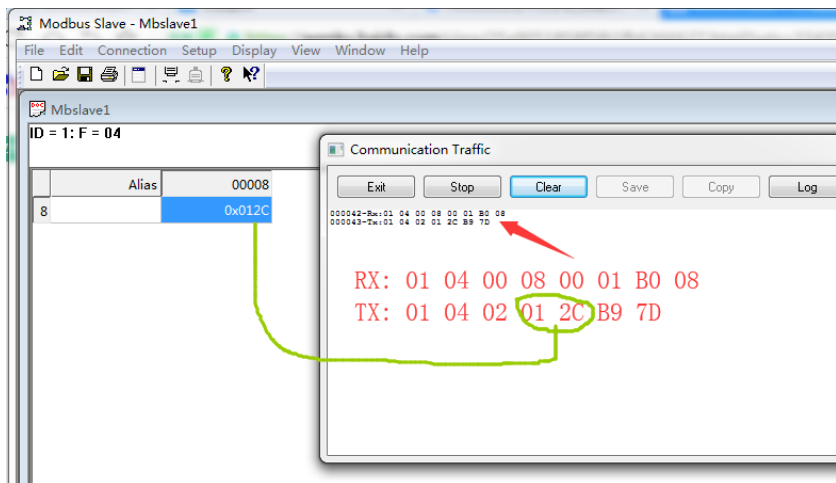
END4

The data frame is as follows:

RX: 01 04 00 08 00 01 B0 08

TX: 01 04 02 01 2C B9 7D

The DM500 communicates with the computer virtual slave:





request	
domain name	HEX
ID	01
Features	04
Starting address Hi	00
Starting address Lo	08
Number of registers Hi	00
Number of registers Li	01
CRC Hi	B0
CRC Lo	08

response	
domain name	16 进制
ID	01
Features	04
Number of bytes	02
Input register value Hi(9) [#1241]	01
Input register value Lo(9) [#1240]	2C
CRC Hi	B9
CRC Lo	7D

# Access route

To purchase the modbus version of the DM500, please click on the following unique link to purchase, other links of the DM500 do not have Modbus protocol function by default.

Note whether MODBUS version is required before purchase.

Alibaba Express:

[https://www.aliexpress.com/store/product/DM500-4axis-Handwheel-Motion-Controller-For-Engraving-Machine-cnc-controlor-500Khz-support-modbus/4427122\\_32959222068.html?spm=2114.12010611.8148356.1.33f83289TIVb55](https://www.aliexpress.com/store/product/DM500-4axis-Handwheel-Motion-Controller-For-Engraving-Machine-cnc-controlor-500Khz-support-modbus/4427122_32959222068.html?spm=2114.12010611.8148356.1.33f83289TIVb55)