

# **Modbus TCP Protocol For CWT-MB series**

## 1. Read digital input

### Fox example, read 8 channels of Di, slave id=1

Master sends: 00 00 00 00 00 06 01 02 00 00 00 08

RTU responds: 00 00 00 00 00 04 01 02 01 02

the explanation of master request command:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	06H	
slave id	1	01H	slave id = 1
function code	1	02H	Read State
start address	2	0000H	Read register start from 10001
number of points	2	0008H	Read 8 registers (10001-10008) corresponds to Di0 to Di7

RTU responds:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	04H	
slave id	1	01H	slave id = 1
function code	1	02H	Read State
number of bytes	1	01H	
data	1	02H	02H = 00000010 (BIN) corresponds to status of Di0 to Di7 Mean Di1 is close

## 2. Write digital output

### Fox example, write multiple Do (slave id=1):

Master sends 《DO-0、 1 close, DO2-7 open》: 00 00 00 00 00 08 01 0F 00 00 00 08 01 03

Module responds 《DO-0、 1 close, DO2-7 open》: 00 00 00 00 00 06 01 0F 00 00 00 08

=====

Master sends 《DO:0- 7 close》: 00 00 00 00 00 08 01 0F 00 00 00 08 01 FF

Module responds 《DO:0- 7 close》: 00 00 00 00 00 06 01 0F 00 00 00 08

=====

Master sends 《DO:0- 7 open》: 00 00 00 00 00 08 01 0F 00 00 00 08 01 00

Module responds 《DO:0- 7 open》: 00 00 00 00 00 06 01 0F 00 00 00 08

=====

Master sends 《DO:0-12 close》: 00 00 00 00 00 09 01 0F 00 00 00 0C 02 FF FF

Module responds 《DO:0-12 close》: 00 00 00 00 00 06 01 0F 00 00 00 0C

Master sends 《DO:0-12 open》: 00 00 00 00 00 09 01 0F 00 00 00 0C 02 00 00

Module responds 《DO:0-12 open》: 00 00 00 00 00 06 01 0F 00 00 00 0C

the explanation of master send command:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	08H	
slave id	1	01H	slave id =1
function code	1	0FH	write multiple coil
start address	2	0000H	write register start from 00001
number of points	2	0008H	write 8 registers (00001-00008) corresponds to Do0 to Do7
number of bytes	1	01H	write 1 byte
data	1	03H	03H = 0 0 0 0 0 0 1 1 (bin) Mean close Do0, Do1, and open Do2-Do7

RTU responds:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	08H	
slave id	1	01H	slave id =1
function code	1	0FH	write multiple coil
start address	2	0000H	write register start from0001
number of points	2	0008H	write 8 registers (00001-00008) corresponds to Do0 to Do7

### Write single DO

Master sends 《DO-0 close》: 00 00 00 00 00 06 01 05 00 00 FF 00

Module responds 《DO-0 close》: 00 00 00 00 00 06 01 05 00 00 FF 00

Master sends 《DO-0 open》: 00 00 00 00 00 06 01 05 00 00 00 00

Module responds 《DO-0 open》: 00 00 00 00 00 06 01 05 00 00 00 00

=====

Master sends 《DO-1 close》: 00 00 00 00 00 06 01 05 00 01 FF 00

Module responds 《DO-1 close》: 00 00 00 00 00 06 01 05 00 01 FF 00

Master sends 《DO-1 open》: 00 00 00 00 00 06 01 05 00 01 00 00

Module responds 《DO-1 open》: 00 00 00 00 00 06 01 05 00 01 00 00

=====

Master sends 《DO-2 close》: 00 00 00 00 00 06 01 05 00 02 FF 00

Module responds 《DO-2 close》: 00 00 00 00 00 06 01 05 00 02 FF 00

Master sends 《DO-2 open》: 00 00 00 00 00 06 01 05 00 02 00 00

Module responds 《DO-2 open》: 00 00 00 00 00 06 01 05 00 02 00 00

Master sends 《DO-3 close》: 00 00 00 00 00 06 01 05 00 03 FF 00  
Module responds 《DO-3 close》: 00 00 00 00 00 06 01 05 00 03 FF 00  
Master sends 《DO-3 open》: 00 00 00 00 00 06 01 05 00 03 00 00  
Module responds 《DO-3 open》: 00 00 00 00 00 06 01 05 00 03 00 00  
=====

Master sends 《DO-4 close》: 00 00 00 00 00 06 01 05 00 04 FF 00  
Module responds 《DO-4 close》: 00 00 00 00 00 06 01 05 00 04 FF 00  
Master sends 《DO-4 open》: 00 00 00 00 00 06 01 05 00 04 00 00  
Module responds 《DO-4 open》: 00 00 00 00 00 06 01 05 00 04 00 00  
=====

Master sends 《DO-5 close》: 00 00 00 00 00 06 01 05 00 05 FF 00  
Module responds 《DO-5 close》: 00 00 00 00 00 06 01 05 00 05 FF 00  
Master sends 《DO-5 open》: 00 00 00 00 00 06 01 05 00 05 00 00  
Module responds 《DO-5 open》: 00 00 00 00 00 06 01 05 00 05 00 00  
=====

Master sends 《DO-6 close》: 00 00 00 00 00 06 01 05 00 06 FF 00  
Module responds 《DO-6 close》: 00 00 00 00 00 06 01 05 00 06 FF 00  
Master sends 《DO-6 open》: 00 00 00 00 00 06 01 05 00 06 00 00  
Module responds 《DO-6 open》: 00 00 00 00 00 06 01 05 00 06 00 00  
=====

Master sends 《DO-7 close》: 00 00 00 00 00 06 01 05 00 07 FF 00  
Module responds 《DO-7 close》: 00 00 00 00 00 06 01 05 00 07 FF 00  
Master sends 《DO-7 open》: 00 00 00 00 00 06 01 05 00 07 00 00  
Module responds 《DO-7 open》: 00 00 00 00 00 06 01 05 00 07 00 00  
=====

Master sends 《DO-8 close》: 00 00 00 00 00 06 01 05 00 08 FF 00  
Module responds 《DO-8 close》: 00 00 00 00 00 06 01 05 00 08 FF 00  
Master sends 《DO-8 open》: 00 00 00 00 00 06 01 05 00 08 00 00  
Module responds 《DO-8 open》: 00 00 00 00 00 06 01 05 00 08 00 00  
=====

Master sends 《DO-9 close》: 00 00 00 00 00 06 01 05 00 09 FF 00  
Module responds 《DO-9 close》: 00 00 00 00 00 06 01 05 00 09 FF 00  
Master sends 《DO-9 open》: 00 00 00 00 00 06 01 05 00 09 00 00  
Module responds 《DO-9 open》: 00 00 00 00 00 06 01 05 00 09 00 00  
=====

Master sends 《DO-10 close》: 00 00 00 00 00 06 01 05 00 0A FF 00  
Module responds 《DO-10 close》: 00 00 00 00 00 06 01 05 00 0A FF 00  
Master sends 《DO-10 open》: 00 00 00 00 00 06 01 05 00 0A 00 00  
Module responds 《DO-10 open》: 00 00 00 00 00 06 01 05 00 0A 00 00  
=====

Master sends 《DO-11 close》: 00 00 00 00 00 06 01 05 00 0B FF 00

Module responds 《DO-11 close》: 00 00 00 00 00 06 01 05 00 0B FF 00

Master sends 《DO-11 open》: 00 00 00 00 00 06 01 05 00 0B 00 00

Module responds 《DO-11 open》: 00 00 00 00 00 06 01 05 00 0B 00 00

### 3. Read analog input

**Fox example, read 8 channels of Ai, slave id=1:**

Master sends: 00 00 00 00 00 06 01 03 00 32 00 08

Module responds: 00 00 00 00 00 13 01 03 10 11 A1 12 A2 13 A3 14 A4 15 A5 16 A6 17 A7 18 A8

the explanation of master request command:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	06H	
slave id	1	01H	Slave id =1
function code	1	03H	Read Hold
start address	2	0032H	read register start from 40051
number of points	2	0008H	Read 8 registers (40051-40058) corresponds to Ai0 to Ai7

RTU responds:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1	13H	
slave id	1	01H	slave id = 1
function code	1	03H	Read Hold
number of bytes	1	10H	16 bytes
data	16	11A1H 12A2H	Channel 0=4513      Channel 4=5541
		13A3H 14A4H	Channel 1=4770      Channel 5=5789
		15A5H 16A6H	Channel 2=5027      Channel 6=6055
		17A7H 18A8H	Channel 3=5284      Channel 7=6312

### 4. Write analog output

**Fox example, write multiple AO**

AO0=10000, AO1=20000, AO2=30000, AO3=40000 <DEC>

AO0= 2710, AO1= 4E20, AO2= 7530, AO3= 9C40 <Hex>

Master sends: 00 00 00 00 00 0F 01 10 00 00 00 04 08 27 10 4E 20 75 30 9C 40

Module responds: 00 00 00 00 00 06 01 10 00 00 00 04

the explanation of master request command:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1		
slave id	1	01H	Slave id =1
function code	1	10H	Write multiple hold registers
start address	2	0000H	register start from 40001
number of points	2	0004H	write 4 registers (40001-40004) corresponds to AO0 to AO3
number of bytes	1	08H	Register quantity*2
AO0 value	2	2710H	
AO0 value	2	4E20H	
AO0 value	2	7530H	
AO0 value	2	9C40H	
CRC CHECK	2	189FH	

RTU responds:

Message description	Number of bytes	Message	Explanation
Protocol header	5	0 0 0 0 0	
following bytes	1		
slave id	1	01H	slave id = 1
function code	1	10H	Write multiple hold registers
start address	2	0000H	register start from 40001
number of points	2	0004H	write 4 registers (40001-40004) corresponds to AO0 to AO3

## Write single AO

Master sends 《AO-0 Write 10000》: 00 00 00 00 00 06 01 06 00 00 27 10

Module responds 《AO-0 Write 10000》: 00 00 00 00 00 06 01 06 00 00 27 10

Master sends 《AO-1 Write 10000》: 00 00 00 00 00 06 01 06 00 01 27 10

Module responds 《AO-1 Write 10000》: 00 00 00 00 00 06 01 06 00 01 27 10

Master sends 《AO-2 Write 10000》: 00 00 00 00 00 06 01 06 00 02 27 10

Module responds 《AO-2 Write 10000》: 00 00 00 00 00 06 01 06 00 02 27 10

Master sends 《AO-3 Write 10000》: 00 00 00 00 00 06 01 06 00 03 27 10

Module responds 《AO-3 Write 10000》: 00 00 00 00 00 06 01 06 00 03 27 10