

EtherCAT terminal board expands modbus function testing report

2020.05.23

Shenzhen Han's Robot Co., Ltd.

1 ,Testing purposes

EtherCAT End plate support modbus Functions can be extended for terminal access modbus

The capabilities of the device, for this function, use EtherCAT Master station and conform modbus

Agreement slave carry out testing.

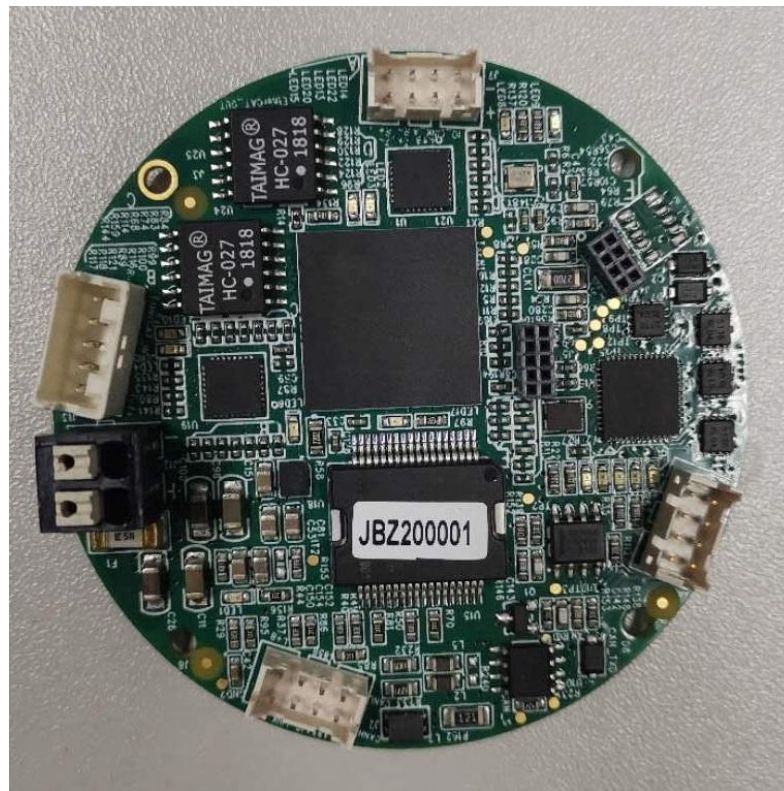
Currently,The first step is to test modbus Whether the communication function has been realized,

And whether it complies with the standard agreement rules.

2 ,Test object

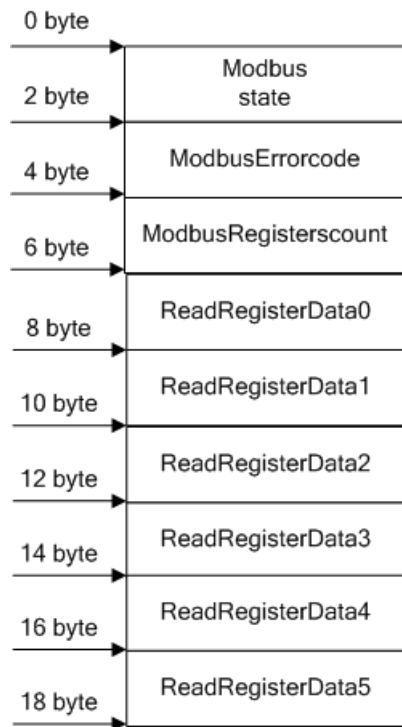
EtherCAT End plate (support modbus Features)

,As shown below:

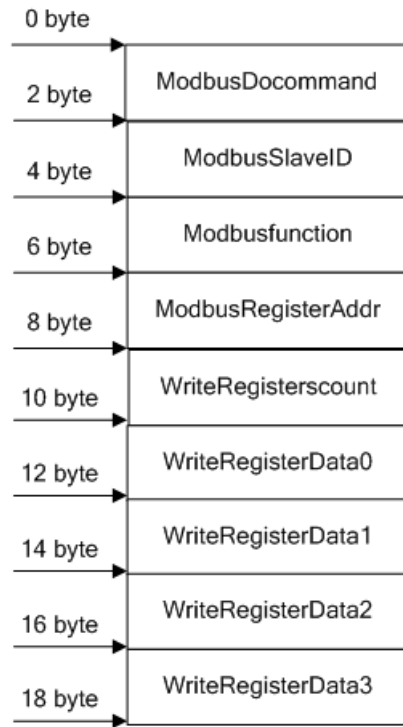


3 ,Glossary(PDOs)

Modbus Read PDO



Modbus write PDO



(1) Modbus state :

Means Modbus Status, the end board feedbacks the current Modbus State to this

PDO , The meaning of each value is as follows:

Value	status
0	Idle : Connected and in idle state, waiting for instructions at this time
1	Busy : Executing instruction
2	Disconnect :versus modbus Disconnect

(2) Modbus Error Code :

Means modbus Error code, if modbus If an error occurs during communication, it will

through this PDO To feedback the current error code, the meaning of each value is as follows:

Value	status
0	Success
1	illegal Function
2	illegal Data Address
3	illegal Data value
4	Slave Device Failure
224(0xE0)	Invalid SlaveID
225(0xE1)	Invalid Function
226(0xE2)	Modbus Response TimedOut
227(0xE3)	Invalid CRC

(3) Modbus Registers count :

Indicates the number of registers to be read and written, up to which can be read 6 The number of registers.

(4) Read Register Data 0~5 :

Represents the read data register 0~5 , Readable 6 Register data.

(5) Modbus Do Command :

Indicates the execution command of the end board. When all the parameters of a certain command are set,

To use Do command Instruct the end board to execute instructions.

Do command From 1 Becomes 0 Time, that is, the falling edge, the end board

make.

(6) Modbus Slave ID :

Means conformity Modbus Technical specification of the slave device ID , Here slave ID

for 1 ;

(7) Modbus Function :

Means Modbus The function code of communication, the standard function code is defined as follows:

function code (16 Base)	name	effect
The following is modbus Defined function code		
0x01 (Not implemented) Read coil status		Read discrete output bits
0x02 (Not implemented) Read input status		Read discrete input bits
0x03	Read holding register	
0x04	Read input register	read 16 Bit register, often used to read the whole Number or floating point
0x05 (Not implemented) Write a single coil		Write data to make the coil On/OFF
0x06	Write a single register	16 Write data in bit format
0x10	Write multiple registers	16 Write data in bit format

(8) Write Registers Addr :

Indicates the address of the register to be written, the length is 2byte .

(9) Write Registers count :

Represents the number of registers that need to be written, the maximum number of registers that can be written is 4 One.

(10) Write Register Data 0~3 :

It indicates the value that needs to be sent to the register.

4 , Test date

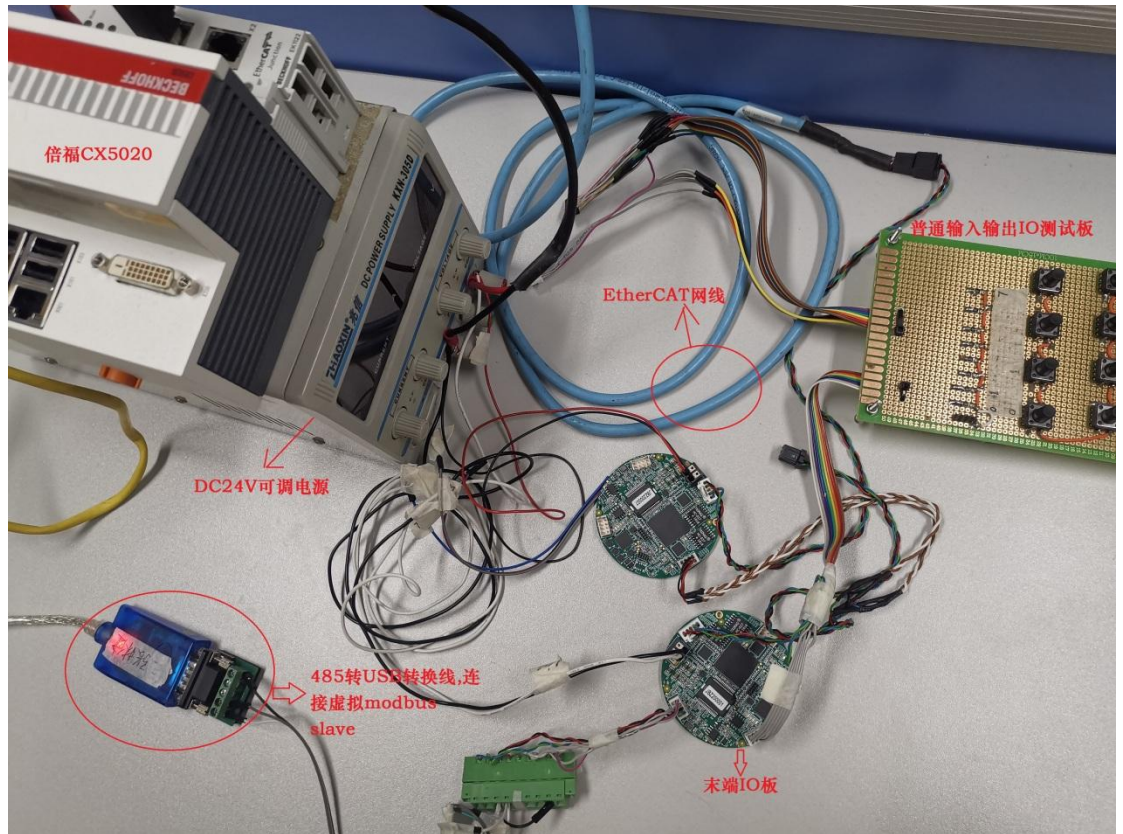
2020.05.21~2020.05.22

5 ,test environment

- (1) EtherCAT Master station equipment: Beckhoff CX5020 One, EK1122 One;
- (2) modbus Slave Equipment: Virtual slave, which is modbus slave software;
- (3) EtherCAT End plate modbus expanding board;
- (4) other: 24V power supply, twinCAT2 , Several power cords;

6 ,Testing process

a) Test platform construction



b) TwinCAT2 test program

(1) Write multiple registers function (0x10):

```
(*write multiple/single registers*)
IF Input_State =0 AND Input_ErrorCode = 0 THEN
    Output_SlaveID :=1;
    Output_Function :=16;
    Output_RegAddr :=1;
    Output_RegCnt :=4;
    Output_RegData0 :=1;
    Output_RegData1 :=2;
    Output_RegData2 :=3;
    Output_RegData3 :=4;
END_IF;
```

(2) Read multiple registers function (0x03):

```
IF Input_State =0 AND Input_ErrorCode = 0 THEN
    Output_SlaveID :=1;
    Output_Function :=3;
    Output_RegAddr :=1;
    Output_RegCnt :=6;
END_IF;
```

(3) Trigger read/write function command:

```
(*Do command*)
Output_DoCommand:=1;

Delay_Count:=Delay_Count+1;
IF Delay_Count >= 100 THEN
    Output_DoCommand:=0;
    Delay_Count:=0;
    DigitalOutput:=1;
END_IF;
```

7 ,Test Results

(1) Write multiple registers function (0x10):

①TwinCAT2 Issue a write register instruction to 4 Write any number in each register:

Name		Online	Type	Size	>Add...	In...	Us...
DigitalOutputs	X	0x0001 (1)	UINT	2.0	39.0	Output	0
ModbusDoCommand	X	0x0000 (0)	UINT	2.0	41.0	Output	0
ModbusSlaveID	X	0x0001 (1)	UINT	2.0	43.0	Output	0
ModbusFunction	X	0x0010 (16)	UINT	2.0	45.0	Output	0
ModbusRegisterAddr	X	0x0001 (1)	UINT	2.0	47.0	Output	0
ModbusRegisterCount	X	0x0004 (4)	UINT	2.0	49.0	Output	0
ModbusRegisterData0	X	0x0001 (1)	UINT	2.0	51.0	Output	0
ModbusRegisterData1	X	0x0002 (2)	UINT	2.0	53.0	Output	0
ModbusRegisterData2	X	0x0003 (3)	UINT	2.0	55.0	Output	0
ModbusRegisterData3	X	0x0004 (4)	UINT	2.0	57.0	Output	0

②modbus slave Receive data and the results are as follows:

	Alias	00000
0		0
1		1
2		2
3		3
4		4
5		0
6		0
7		0
8		0
9		0

Virtual Modbus The slave station can receive PDO Issued modbus instruction;

(2) Read multiple registers function (0x03):

①TwinCAT2 Read 6 Register instructions:

Name		Online	Type	Size	>Add...	In...	Us...
DigitalOutputs	X	0x0002 (2)	UINT	2.0	39.0	Output	0
ModbusDoCommand	X	0x0000 (0)	UINT	2.0	41.0	Output	0
ModbusSlaveID	X	0x0001 (1)	UINT	2.0	43.0	Output	0
ModbusFunction	X	0x0003 (3)	UINT	2.0	45.0	Output	0
ModbusRegisterAddr	X	0x0001 (1)	UINT	2.0	47.0	Output	0
ModbusRegisterCount	X	0x0006 (6)	UINT	2.0	49.0	Output	0
ModbusRegisterData0	X	0x0000 (0)	UINT	2.0	51.0	Output	0
ModbusRegisterData1	X	0x0000 (0)	UINT	2.0	53.0	Output	0
ModbusRegisterData2	X	0x0000 (0)	UINT	2.0	55.0	Output	0
ModbusRegisterData3	X	0x0000 (0)	UINT	2.0	57.0	Output	0

②PDO received modbus The data of the feedback register is as follows:

Name	Online	Type	Size	>Add...	In...	Us...
DigitalInputs	X	0x0000 (0)	UINT	2.0	39.0	Input 0
AnalogInputCH1	X	0x0007 (7)	UINT	2.0	41.0	Input 0
AnalogInputCH2	X	0x000A (10)	UINT	2.0	43.0	Input 0
Temperature	X	0x0584 (1412)	UINT	2.0	45.0	Input 0
Vdc_Bus	X	0x0773 (1907)	UINT	2.0	47.0	Input 0
ModbusState	X	0x0000 (0)	UINT	2.0	49.0	Input 0
ModbusErrorCode	X	0x0000 (0)	UINT	2.0	51.0	Input 0
ModbusRegistersCount	X	0x0006 (6)	UINT	2.0	53.0	Input 0
ModbusRegistersData0	X	0x0001 (1)	UINT	2.0	55.0	Input 0
ModbusRegistersData1	X	0x0002 (2)	UINT	2.0	57.0	Input 0
ModbusRegistersData2	X	0x0003 (3)	UINT	2.0	59.0	Input 0
ModbusRegistersData3	X	0x0004 (4)	UINT	2.0	61.0	Input 0
ModbusRegistersData4	X	0x0000 (0)	UINT	2.0	63.0	Input 0
ModbusRegistersData5	X	0x0000 (0)	UINT	2.0	65.0	Input 0

The master station issues an instruction to read the register, PDO Can receive the value read up.

(3) Multiple EtherCAT End board connection test:

①TwinCAT2 Scanning two slave stations, the scan results are as follows:

The screenshot displays the TwinCAT2 configuration interface. The left pane shows a project tree with 'Device 1 (EtherCAT)' and 'Device 2 (EtherCAT)' selected. The right pane shows the 'General' tab for 'Device 1 (EtherCAT)' with fields for Name, Type, and Comment. Below this is a table of I/O data. The bottom pane shows the 'Process Data' tab for 'Device 2 (EtherCAT)' with a table of I/O data.

Number	Box Name	Address	Type	In Size	Out ...	E-Bus ...	Link
1	Term 1 (EKI122)	1001	EKI122			-220 1	
2	Box 2 (F2838x CPU1 EtherCAT Slave)	1002	F2838x CPU1 Eth...	38.0	30.0		

Name	Type	Size	>Add...	In...	Us...	Linked to
DigitalInputs	X	UINT	2.0	109.0	Input 0	DigitalInputs . Inpu...
AnalogInputCH1	X	UINT	2.0	111.0	Input 0	AnalogInCH1 . Inputs...
AnalogInputCH2	X	UINT	2.0	113.0	Input 0	AnalogInCH2 . Inputs...
Temperature	X	UINT	2.0	115.0	Input 0	Temperature . Inputs...
Vdc_Bus	X	UINT	2.0	117.0	Input 0	vdc_bus . Inputs . S...
ModbusState	X	UINT	2.0	119.0	Input 0	Input_State . Inputs...
ModbusErrorCode	X	UINT	2.0	121.0	Input 0	Input_ErrorCode . In...
ModbusRegistersCount	X	UINT	2.0	123.0	Input 0	Input_Regent . Inputs...
ModbusRegistersData0	X	UINT	2.0	125.0	Input 0	Input_RegData0 . Inp...
ModbusRegistersData1	X	UINT	2.0	127.0	Input 0	Input_RegData1 . Inp...
ModbusRegistersData2	X	UINT	2.0	129.0	Input 0	Input_RegData2 . Inp...
ModbusRegistersData3	X	UINT	2.0	131.0	Input 0	Input_RegData3 . Inp...
ModbusRegistersData4	X	UINT	2.0	133.0	Input 0	Input_RegData4 . Inp...
ModbusRegistersData5	X	UINT	2.0	135.0	Input 0	Input_RegData5 . Inp...
DataToMaster	UINT	4.0	137.0	Input 0		
ModeResponse	UINT	2.0	141.0	Input 0		
SpeedFosFbk	UINT	4.0	143.0	Input 0		
WcState	BOOL	0.1	1522.3	Input 0		
InputToggle	BOOL	0.1	1524.3	Input 0		

From the scan results, the master station can scan two slave stations, and the communication data feedback is positive often;

8 , The next test plan

(1) Stability test:

- ① Connect a single slave station, and all connected to the actual modbus slave Communication aging

Test to monitor the probability of errors;

- ② Connect multiple slave stations, and all are connected to the actual modbus slave Communication aging

Test to monitor the probability of errors;