

### 3 Object Dictionary

---

#### 3.1 Object Type

Type Name	Description
WAS	single value, such as an UNSIGNED8, UNSIGNED16, or INTEGER32. Multiple data field objects,
ARRAY	where all variables in the field are of the same data type. UNSIGNED16 array, etc., Sub-index 0 belongs to UNSIGNED8, so it is not an array data multiple data field
RECORD	object, and the variables in the field are all different data types. Sub-index 0 belongs to UNSIGNED8, not RECORD data.

#### 3.2 Data Types

Type Name	Description
UNSIGNED8	Unsigned 8-bit data.
UNSIGNED16	Unsigned 16-bit data
UNSIGNED32	Unsigned 32-bit data.
INTEGER32	Signed 32-bit data

#### 3.3 Access Rights

Permission Name	Description
RO	Read-only
RW	Readable and writable

### 3.4 Object dictionary list

Object Dictionary	Object Type	Name	Data type access permissions
0x2000	WAS	Digital Input	UNSIGNED16 RO
0x2010	WAS	Analog Input 1	UNSIGNED16 RO
0x2011	WAS	Analog Input 2	UNSIGNED16 RO
0x2020	WAS	Encoder Input	INTEGER32 RO
0x2030	WAS	Counter 1	UNSIGNED32 RO
0x2031	WAS	Counter 2	UNSIGNED32 RO
0x2300	WAS	Digital Output	UNSIGNED16 RW
0x2310	WAS	Analog Output 1	UNSIGNED16 RW
0x2311	WAS	Analog Output 2	UNSIGNED16 RW
0x2320	WAS	PWM1 Frequency	UNSIGNED32 RW
0x2321	WAS	PWM2 Frequency	UNSIGNED32 RW
0x2330	WAS	PWM1 Duty Cycle	UNSIGNED16 RW
0x2331	WAS	PWM2 duty cycle	UNSIGNED16 RW
0x2400	WAS	clear counter, encoder value	UNSIGNED16 RW
0x2401	WAS	Set the polarity of PWM and counter	UNSIGNED16 RW
0x2402	RECORD	Serial port settings	UNSIGNED32 RW
0x2600	RECORD	Serial communication data (send)	UNSIGNED32 RW
0x2601	RECORD	Serial communication data (receive)	UNSIGNED32 RW
0x3000	RECORD	Set the serial port Modbus RTU protocol Coils and holding registers in	UNSIGNED8 RW
0x3001	RECORD		UNSIGNED8 RW
0x3002	RECORD		UNSIGNED8 RW
0x3003	RECORD		UNSIGNED8 RW
0x3004	RECORD		UNSIGNED8 RW
0x3005	RECORD		UNSIGNED8 RW
0x3006	RECORD		UNSIGNED8 RW
0x3007	RECORD		UNSIGNED8 RW
0x3020	RECORD	Set the serial port Modbus RTU protocol Discrete inputs, coils, inputs Registers, Holding Registers	UNSIGNED8 RW
0x3021	RECORD		UNSIGNED8 RW
0x3022	RECORD		UNSIGNED8 RW
0x3023	RECORD		UNSIGNED8 RW
0x3024	RECORD		UNSIGNED8 RW
0x3025	RECORD		UNSIGNED8 RW
0x3026	RECORD		UNSIGNED8 RW
0x3027	RECORD		UNSIGNED8 RW

## 3.5 Object Data

### 3.5.1 Switching Input (0x2000)

There are 16 switch input ports on the IO card. The status of each port corresponds to the corresponding 0x2000 object.

When the switch is on, the corresponding bit is set to 0, and when the switch is off, the corresponding bit is set to 1.

Index	0x2000
name	Switch input, corresponding to input ports X01 ~ X16
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RO
PDO Mapping	yes
The	0 ~ 0xFFFF
	0

correspondence between each bit of the value range default value object and the input port:

BIT	15	14	13	12	11	10	9	8
Input terminal	IN15	IN14	IN13	IN12	IN11	IN10	IN9	IN8
BIT	7	6	5	4	3	2	1	0
Input terminal	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0

### 3.5.2 Analog Input (0x2010, 0x2011)

The IO card provides 2 analog inputs, each with a conversion accuracy of 12 bits.

Index	0x2010
Name	Analog input 1
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RO
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	<p>Input voltage range: 0 ~ 10V</p> <p>The conversion relationship between voltage and analog value is:</p> $V = AD \cdot 10 / 4096$ <p>(AD is the analog value, V is the voltage value)</p>

Index	0x2011
name	Analog input 2

Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RO
PDO Mapping	yes
Value Range	0 ~ 0x0FFF
Default Value	0
Remarks	<p>Input voltage range: 0 ~ 10V</p> <p>The conversion relationship between voltage and analog value is:</p> $V = AD \cdot 10 / 4096$ <p>(AD is the analog value, V is the voltage value)</p>

**3.5.3 Encoder Input (0x2020)**

Index	0x2040
Name	Encoder input
Object Type	WAS
Data Type	INTEGER32
Access Rights	RO
PDO mapping	yes
value range	- 2147483648 ~ 2147483647
default value	0

**3.5.4 Counter (0x2030, 0x2031)**

Index	0x2030
Name	Counter input 1
Object Type	WAS
Data Type	INTEGER32
Access Rights	RO
PDO mapping	yes
value range	- 2147483648 ~ 2147483647
default value	0

Index	0x2031
Name	Counter input 2
Object Type	WAS
Data Type	INTEGER32

access permission	RO
PDO mapping	yes
value range	- 2147483648 ~ 2147483647
default value	0

### 3.5.5 Switching Output (0x2300)

There are 16 switch output ports on the IO card. Each bit of the object 0x2300 corresponds to the corresponding port.

When the corresponding bit is set to 0, the switch is connected; when the corresponding bit is set to 1, the switch is disconnected.

Index	0x2300
Name	Switch output, corresponding to output ports Y01 ~ Y16
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RW
The	yes
	0 ~ 0xFFFF
	0xFFFF

correspondence between each bit of the PDO mapping value range default value object and the output port:

BIT	15	14	13	12	11	10	9	8
Output terminal	OUT15	OUT14	OUT13	OUT12	OUT11	OUT10	OUT9	OUT8
BIT	7	6	5	4	3	2	1	0
Output terminal	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0

### 3.5.6 Analog output (0x2310, 0x2311)

The IO card provides 2 analog outputs, each with a conversion accuracy of 12 bits.

Index	0x2310
Name	Analog output 1
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	Output voltage range: 0 ~ 10V  The conversion relationship between voltage and analog value is:

	$DA = V \cdot 4096 / 10$ <p>(DA is the analog value, V is the voltage value)</p>
--	--

Index	0x2311
Name	Analog output 2
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0x0FFF
Default Value	0
Remarks	<p>Output voltage range: 0 ~ 10V</p> <p>The conversion relationship between voltage and analog value is:</p> $DA = V \cdot 4096 / 10$ <p>(DA is the analog value, V is the voltage value)</p>

### 3.5.7 PWM output frequency (0x2320, 0x2321)

Index	0x2320
name	PWM1 output frequency
Object Type	WAS
Data Type	UNSIGNED32
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFFFFFF
Default Value	0x3E80000(65536000)
Remarks	<p>The base clock frequency of PWM is 4M.</p> <p>The PWM output frequency is the value after the actual frequency is amplified 65536 times.</p> <p>The actual frequency range is 0 ~ 65535.9999, and their conversion relationship is:</p> $F_{pwm} = F_d \cdot 65536$ <p>(Fpwm is the PWM output frequency, Fd is the actual frequency)</p>

Index	0x2321
name	PWM2 output frequency

Object Type	WAS
Data Type	UNSIGNED32
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFFFFFF
Default Value	0x3E80000(65536000)
Remarks	<p>The base clock frequency of PWM is 4M.</p> <p>The PWM output frequency is the value after the actual frequency is amplified 65536 times.</p> <p>The actual frequency range is 0 ~ 65535.9999, and their conversion relationship is:</p> $F_{pwm} = F_d * 65536$ <p>(Fpwm is the PWM output frequency, Fd is the actual frequency)</p>

### 3.5.8 PWM Duty Cycle (0x2330, 0x2331)

Index	0x2330
Name	PWM1 Duty Cycle
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0x8000
Remarks	<p>The PWM duty cycle is the value after the actual duty cycle is magnified 65536 times.</p> <p>The actual duty cycle range is 0 ~ 0.9999, and their conversion relationship is:</p> $P_{pwm} = P_d * 65536$ <p>(Ppwm is the PWM duty cycle, Pd is the actual duty cycle)</p>

Index	0x2331
Name	PWM2 Duty Cycle
Object Type	WAS
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0x8000
Remarks	<p>The PWM duty cycle is the value after the actual duty cycle is magnified 65536 times.</p> <p>The actual duty cycle range is 0 ~ 0.9999, and their conversion relationship is:</p>

	$P_{pwm} = P_d * 65536$ <p>(Ppwm is the PWM duty cycle, Pd is the actual duty cycle)</p>
--	--

### 3.5.9 Clear the encoder and counter values (0x2400)

Index	0x2400														
Name	Clear the encoder and counter values														
Object Type	WAS														
Data Type	UNSIGNED16														
Access Rights	RW														
PDO Mapping	no														
Value Range	0 ~ 0xFFFF														
Default Value	0														
Remarks	<table><tr><td>...</td><td>Bit 9</td><td>Bit 8</td><td>...</td><td>Bit 0</td></tr><tr><td>RES</td><td>CLRC2</td><td>CLRC1</td><td>RES</td><td>CLRE</td></tr></table>					...	Bit 9	Bit 8	...	Bit 0	RES	CLRC2	CLRC1	RES	CLRE
	...	Bit 9	Bit 8	...	Bit 0										
	RES	CLRC2	CLRC1	RES	CLRE										
RES: reserved, meaningless															
CLRE: When it changes from 0 to 1, the encoder value is cleared															
CLRC1: When it changes from 0 to 1, the value of counter 1 is cleared.															
CLRC2: When it changes from 0 to 1, the value of register 2 is cleared															

### 3.5.10 Set PWM, counter effective polarity and counter direction (0x2401)

Index	0x2401						
Name	Set the effective polarity of PWM and counter						
Object Type	WAS						
Data Type	UNSIGNED16						
Access Rights	RW						
PDO Mapping	no						
Value Range	0 ~ 0xFFFF						
Default Value	0xFFFF						
Remarks	...		Bit 13	Bit 12	...	Bit 9	Bit 8
	RES	DCNT2 DCNT1	RES SCNT2	SCNT1			
	bit7 – bit2					Bit 1	Bit 0
	RES					SPWM2 SPWM1	
	RES: reserved, meaningless						
SPWM1: Set the effective polarity of PWM1,							
SPWM2: Set the effective polarity of PWM2,							



	<p>SCNT1: Set the effective polarity of counter 1,</p> <p>SCNT2: Set the effective polarity of counter 2.</p> <p>The value of each bit of the effective polarity is 0, which is low level effective, and 1, which is High level is effective</p> <p>DCNT1: Set the counting direction of counter 1, 1 is positive counting</p> <p>DCNT2: Set the counting direction of counter 2, 1 is positive counting</p>
--	--

### 3.5.11 Serial port settings (0x2402)

Use this object to set the serial port baud rate, serial port communication protocol, and Modbus communication timeout.

Time, polling interval.

Index	0x2402
Name	Serial port settings
Object Type	RECORD
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	no

subindex	0
Describe	Number of object members
data type access	UNSIGNED8
permissions	RO
PDO mapping	no
value range	4
default value	4

subindex	1
Describe	Modbus communication timeout
data type access	UNSIGNED16
permissions	RW
PDO Mapping	no
Value Range	2 ~ 65535
Default Value	100
Remarks	The timeout unit is 0.1 milliseconds

subindex	2
describe	Serial port baud rate

Data type	UNSIGNED32
access permissions	RW
PDO mapping	no
value range	9600~19200~38400~57600~115200
default value	115200

subindex	3
describe	Modbus cycle
Data type	UNSIGNED16
access permissions	RW
PDO Mapping	no
Value Range	1~65535
Default Value	2
Remarks	The cycle unit is milliseconds

subindex	4
Describe	Serial communication protocol
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	0~0xFF
Default Value	0
Remarks	0: Modbus RTU protocol Others: User-defined protocols

### 3.5.12 Serial port data (send) (0x2602)

Index	0x2602
Name	Serial port data
Object Type	ARRAY
Data Type	UNSIGNED32
Access Rights	RW
PDO Mapping	yes

subindex	0
describe	Number of data
type of data	UNSIGNED8

access permission	RO
PDO mapping	no
value range	8
default value	8

subindex	1 ~ 8																																																							
Describe	Data value sent via the serial port																																																							
data type	UNSIGNED32																																																							
access permissions	RW																																																							
PDO Mapping	yes																																																							
Value Range	0 ~ 0xFFFFFFFF																																																							
Default Value	0																																																							
Remarks	<p>When 0x2402.4 is set to Modbus protocol, its value is written to the slave station</p> <p>The value of the coil or holding register. Set by 0x3000 ~ 0x3007 object</p> <p>The corresponding 0x2602 object attributes are:</p> <table><tr><td>Correspondence</td><td>Correspondence</td></tr><tr><td>0x2602.1 &lt; - &gt; 0x3000 0x2602.5 &lt; - &gt; 0x3004</td><td></td></tr><tr><td>0x2602.2 &lt; - &gt; 0x3001 0x2602.6 &lt; - &gt; 0x3005</td><td></td></tr><tr><td>0x2602.3 &lt; - &gt; 0x3002 0x2602.7 &lt; - &gt; 0x3006</td><td></td></tr><tr><td>0x2602.4 &lt; - &gt; 0x3003</td><td>0x2602.8 &lt; - &gt; 0x3007</td></tr></table> <p>When 0x2402.4 is set to user-defined protocol, Data 1 ~ 32 are used for</p> <p>Transmit the data that needs to be sent through the serial port:</p> <table><tr><td>bit</td><td>0 – 7</td><td>8 - 15</td><td>16 - 23</td><td>24 - 31</td></tr><tr><td>0x2602.1 Data 1</td><td></td><td>Data 2</td><td>Data 3</td><td>Data 4</td></tr><tr><td>0x2602.2 Data 5</td><td></td><td>Data 6</td><td>Data 7</td><td>Data 8</td></tr><tr><td>0x2602.3 Data 9</td><td></td><td>Data 10</td><td>Data 11</td><td>Data 12</td></tr><tr><td>0x2602.4 Data 13</td><td></td><td>Data 14</td><td>Data 15</td><td>Data 16</td></tr><tr><td>0x2602.5 Data 17</td><td></td><td>Data 18</td><td>Data 19</td><td>Data 20</td></tr><tr><td>0x2602.6 Data 21</td><td></td><td>Data 22</td><td>Data 23</td><td>Data 24</td></tr><tr><td>0x2602.7 Data 25</td><td></td><td>Data 26</td><td>Data 27</td><td>Data 28</td></tr><tr><td>0x2602.8 Data 29</td><td></td><td>Data 30</td><td>Data 31</td><td>Data 32</td></tr></table>	Correspondence	Correspondence	0x2602.1 < - > 0x3000 0x2602.5 < - > 0x3004		0x2602.2 < - > 0x3001 0x2602.6 < - > 0x3005		0x2602.3 < - > 0x3002 0x2602.7 < - > 0x3006		0x2602.4 < - > 0x3003	0x2602.8 < - > 0x3007	bit	0 – 7	8 - 15	16 - 23	24 - 31	0x2602.1 Data 1		Data 2	Data 3	Data 4	0x2602.2 Data 5		Data 6	Data 7	Data 8	0x2602.3 Data 9		Data 10	Data 11	Data 12	0x2602.4 Data 13		Data 14	Data 15	Data 16	0x2602.5 Data 17		Data 18	Data 19	Data 20	0x2602.6 Data 21		Data 22	Data 23	Data 24	0x2602.7 Data 25		Data 26	Data 27	Data 28	0x2602.8 Data 29		Data 30	Data 31	Data 32
Correspondence	Correspondence																																																							
0x2602.1 < - > 0x3000 0x2602.5 < - > 0x3004																																																								
0x2602.2 < - > 0x3001 0x2602.6 < - > 0x3005																																																								
0x2602.3 < - > 0x3002 0x2602.7 < - > 0x3006																																																								
0x2602.4 < - > 0x3003	0x2602.8 < - > 0x3007																																																							
bit	0 – 7	8 - 15	16 - 23	24 - 31																																																				
0x2602.1 Data 1		Data 2	Data 3	Data 4																																																				
0x2602.2 Data 5		Data 6	Data 7	Data 8																																																				
0x2602.3 Data 9		Data 10	Data 11	Data 12																																																				
0x2602.4 Data 13		Data 14	Data 15	Data 16																																																				
0x2602.5 Data 17		Data 18	Data 19	Data 20																																																				
0x2602.6 Data 21		Data 22	Data 23	Data 24																																																				
0x2602.7 Data 25		Data 26	Data 27	Data 28																																																				
0x2602.8 Data 29		Data 30	Data 31	Data 32																																																				

### 3.5.13 Serial port data (receive) (0x2603)

Index	0x2603
Name	Serial port data
object type	ARRAY

Data type	UNSIGNED32
access permissions	RW
PDO Mapping	yes

subindex	0
Describe	Number of data
data type	UNSIGNED8
access permissions	RO
PDO mapping	no
value range	8
default value	8

subindex	1 ~ 8																																								
Describe	Data value received by the serial port																																								
data type	UNSIGNED32																																								
access permissions	RW																																								
PDO Mapping	yes																																								
Value Range	0 ~ 0xFFFFFFFF																																								
Default Value	0																																								
Remarks	<p>When 0x2402.4 is set to Modbus protocol, it is the coil read back from the slave.</p> <p>Discrete input, input register, and holding register values. The corresponding 0x2603 object attributes are set by 0x3020 ~ 0x3027 objects:</p> <table><tr><td>Correspondence</td><td>Correspondence</td></tr><tr><td>0x2603.1 &lt; - &gt; 0x3020 0x2603.5 &lt; - &gt; 0x3024</td><td></td></tr><tr><td>0x2603.2 &lt; - &gt; 0x3021 0x2603.6 &lt; - &gt; 0x3025</td><td></td></tr><tr><td>0x2603.3 &lt; - &gt; 0x3022 0x2603.7 &lt; - &gt; 0x3026</td><td></td></tr><tr><td>0x2603.4 &lt; - &gt; 0x3023</td><td>0x2603.8 &lt; - &gt; 0x3027</td></tr></table> <p>When 0x2402.4 is set to a user-defined protocol, 0x2601.2 is set to other values.</p> <p>When it is changed to 0xFF, the IO card updates the data received by the serial port to data 1~32.</p> <p>The number of updates is indicated by 0x2601.4:</p> <table><tr><td>bit</td><td>0 – 7</td><td>8 - 15</td><td>16 - 23</td><td>24 - 31</td></tr><tr><td>0x2603.1 Data 1</td><td></td><td>Data 2</td><td>Data 3</td><td>Data 4</td></tr><tr><td>0x2603.2 Data 5</td><td></td><td>Data 6</td><td>Data 7</td><td>Data 8</td></tr><tr><td>0x2603.3 Data 9</td><td></td><td>Data 10</td><td>Data 11</td><td>Data 12</td></tr><tr><td>0x2603.4 Data 13</td><td></td><td>Data 14</td><td>Data 15</td><td>Data 16</td></tr><tr><td>0x2603.5 Data 17</td><td></td><td>Data 18</td><td>Data 19</td><td>Data 20</td></tr></table>	Correspondence	Correspondence	0x2603.1 < - > 0x3020 0x2603.5 < - > 0x3024		0x2603.2 < - > 0x3021 0x2603.6 < - > 0x3025		0x2603.3 < - > 0x3022 0x2603.7 < - > 0x3026		0x2603.4 < - > 0x3023	0x2603.8 < - > 0x3027	bit	0 – 7	8 - 15	16 - 23	24 - 31	0x2603.1 Data 1		Data 2	Data 3	Data 4	0x2603.2 Data 5		Data 6	Data 7	Data 8	0x2603.3 Data 9		Data 10	Data 11	Data 12	0x2603.4 Data 13		Data 14	Data 15	Data 16	0x2603.5 Data 17		Data 18	Data 19	Data 20
Correspondence	Correspondence																																								
0x2603.1 < - > 0x3020 0x2603.5 < - > 0x3024																																									
0x2603.2 < - > 0x3021 0x2603.6 < - > 0x3025																																									
0x2603.3 < - > 0x3022 0x2603.7 < - > 0x3026																																									
0x2603.4 < - > 0x3023	0x2603.8 < - > 0x3027																																								
bit	0 – 7	8 - 15	16 - 23	24 - 31																																					
0x2603.1 Data 1		Data 2	Data 3	Data 4																																					
0x2603.2 Data 5		Data 6	Data 7	Data 8																																					
0x2603.3 Data 9		Data 10	Data 11	Data 12																																					
0x2603.4 Data 13		Data 14	Data 15	Data 16																																					
0x2603.5 Data 17		Data 18	Data 19	Data 20																																					

	<b>0x2603.6</b> Data 21	Data 22	Data 23	Data 24
	<b>0x2603.7</b> Data 25	Data 26	Data 27	Data 28
	<b>0x2603.8</b> Data 29	Data 30	Data 31	Data 32

### 3.5.14 Custom protocol (send) (0x2600)

When 0x2402.4 is set to a custom protocol, the serial port data can be passed to the IO through the 0x2602 object Card, and then sent through the serial port of the IO card.

To use the serial port, you must first send data to 0x2602.1~8, and then

Set the number of valid data (BYTE) to 0x2600.4, and then modify the ID in 0x2600.3. The IO card passes the judgment.

After the ID is different, the data in 0x2602.1~8 will be saved in the buffer area, so each time the data is sent,

The ID in 0x2600.3 must be different from the previous one. After all data is sent to the IO card, send 0x2600.1

Change the value from other values to 0xFF, and IO sends the data in the buffer through the serial port. 0x2600.2 indicates that the data is sent Status code.

Index	0x2600
Name	Serial port data
Object Type	RECORD
Data Type	UNSIGNED32
Access Rights	RW
PDO Mapping	yes

subindex	0
Describe	Number of members
data type	UNSIGNED8
access permissions	RO
PDO Mapping	no
Value Range	4
Default Value	4

subindex	1
describe	Send Enable
Data type	UNSIGNED16
access permissions	RW
PDO mapping	yes
value range	0 ~ 0xFF
default value	0

Remark	<p>Enable serial port data transmission. When the data to be sent is transmitted to the IO card</p> <p>After that, when the value changes from other values to 0xFF, the IO card can transfer the cached data through the serial port.</p> <p>Send via the port.</p>
--------	--

subindex	2
Describe	Data sending status
data type access	UNSIGNED16
permissions	RW
PDO Mapping	yes
Value Range	0 ~ 0xFF
Default Value	0
Remarks	<p>When bit0 is set to 1, the data length in the serial port buffer exceeds the maximum value (256)</p> <p>When bit1 is set to 1, it indicates that the serial port is sending data.</p>

subindex	3
Describe	Serial port data ID in 0x2602.1~8
data type access	UNSIGNED16
permissions	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	<p>When the ID is different from the last time and subindex4 is not 0, 0x2602.1~8</p> <p>The data is saved in the buffer.</p>

subindex	4
Describe	The number of serial port data in 0x2602.1~8, in byte
data type access	UNSIGNED16
permissions	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	<p>0x2602.1~8 The valid number of transmitted data (bytes) cannot exceed</p> <p>These data will be cached in the cache area, which can cache</p> <p>The total data length is 256 bytes.</p>

**3.5.15 Custom Protocol (Receive) (0x2601)**

When 0x2402.4 is set to a custom protocol, the serial port receives data and saves the data to the buffer area.

And declare the number of data in the buffer in 0x2601.1 (maximum 256).

When it is changed to 0xFF, the serial port will update the data in the buffer to 0x2603.1~8 and indicate at 0x2601.4

The valid number of data, in 0x2601.3, indicates the index of this data.

The index of the data is incremented by 1.

Index	0x2601
Name	Serial port data
Object Type	RECORD
Data Type	UNSIGNED32
Access Rights	RW
PDO Mapping	yes

subindex	0
Describe	Number of members
data type access	UNSIGNED8
permissions	RO
PDO mapping	no
value range	4
default value	4

subindex	1
Describe	The number of data received by the serial port in the buffer area
data type access	UNSIGNED16
permissions	RO
PDO Mapping	yes
Value Range	0 ~ 256
Default Value	0
Remarks	When subindex2 is changed from other values to 0xFF, the IO card will  If the data is updated to 0x2603.1~8, the number of data not updated is saved in  subindex1

subindex	2
	Update the data in 0x2603.1~8
Describing Data Types	UNSIGNED16
access permission	RW

PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	When subindex2 is changed from other values to 0xFF, the IO card will  The data is updated to 0x2603.1~8, and the number of updates is indicated by subindex4.

subindex	3
Describe	Data ID in 0x2603.1~8
data type access	UNSIGNED16
permissions	RW
PDO Mapping	yes
Value Range	0 ~ 0xFFFF
Default Value	0
Remarks	When the ID is different from the last time, it indicates that the data in 0x2603.1~8 has been changed.  The number of updated data is determined by subindex 4.

subindex	4
Describe	0x2603.1~8 The number of valid data (in bytes)
data type access	UNSIGNED16
permissions	RW
PDO Mapping	yes
Value Range	0 ~ 32
Default Value	0
Remarks	

### 3.5.16 Modbus register setting (write) (0x3000 ~ 0x3007)

The IO card has 8 objects for setting Modbus write register communication. Each object can set a slave

Station object.

Index	0x3000 ~ 0x3007
Name	Modbus Register Settings
Object Type	RECORD
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	no

subindex	0
describe	Number of members



Data type	UNSIGNED8
access permissions	RO
PDO mapping	no
value range	7
default value	7

subindex	1									
describe	Modbus status value									
Data type	UNSIGNED16									
access permissions	RO									
PDO Mapping	yes									
Value Range	0 ~ 0xFFFF									
Default Value	0									
Remarks	...		Bit6	Bit5	Bit4		...	Bit1	Bit0	
	RES		TYPE		SID		BIT	RES	TO	RES
	When the corresponding bit is set to 1:									
	RES: reserved bit, meaningless									
	TO: Communication timeout									
	BIT: The number of valid bits for setting Modbus register is incorrect									
	SID: The Modbus slave ID is incorrect.									
	TYPE: The type of Modbus register is incorrect.									

subindex	2
Describe	Set Modbus register address
data type	UNSIGNED16
access permissions	RW
PDO mapping	no
value range	0 ~ 0xFFFF
default value	0

subindex	3
	Set the number of significant digits of the Modbus register
Describing Data Types	UNSIGNED8

access permission	RW
PDO Mapping	no
Value Range	0 ~ 32
Default Value	0
Remarks	<p>0: Invalid value, this value must be rewritten when used</p> <p>When the Modbus register type is set to coil, the value range is 1 ~ 32, indicating the number of coils to be controlled.</p> <p>When the Modbus register type is set to a holding register, only the value 16 or 32, indicating the number of bits of the holding register to be controlled</p>

subindex	4
Describe	Set the Modbus slave ID
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	1 ~ 247
Default Value	0
Remarks	0: Invalid value, this value must be rewritten when used

subindex	5
Describe	Set the Modbus register type
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	1~4
Default Value	0
Remarks	<p>0: Invalid value, this value must be rewritten when used</p> <p>1: Coil</p> <p>4: Holding register</p>

subindex	6
Describe	Enable
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value range	0~1

Default	0
Value Notes	0: Do not write to this register. 1: Write the set slave register regularly

subindex	7
Describe	Communication error count
data type	UNSIGNED16
access permissions	RW
PDO Mapping	no
Value Range	0~65535
Default Value	0
Remarks	

### 3.5.17 Modbus register setting (read) (0x3020 ~ 0x3027)

The IO card has 8 objects used to set up Modbus read register communication. Each object can read a slave

Station object value.

Index	0x3020 ~ 0x3027
Name	Modbus Register Settings
Object Type	RECORD
Data Type	UNSIGNED16
Access Rights	RW
PDO Mapping	no

subindex	0
Describe	Modbus register setting quantity
data type	UNSIGNED8
access permissions	RO
PDO mapping	no
value range	7
default value	7

subindex	1
Describe	Modbus status value
data type	UNSIGNED16
access permissions	RO
PDO Mapping	yes

Value Range	0 ~ 0xFFFF						
Default Value	0						
Remarks	...	Bit6	Bit5	Bit4	...	Bit1	Bit0
	RES TYPE	SID		BIT	RES TO		RES
<p>When the corresponding bit is set to 1:</p> <p>RES: reserved bit, meaningless</p> <p>TO: Communication timeout</p> <p>BIT: The number of valid bits for setting Modbus register is incorrect</p> <p>SID: The Modbus slave ID is incorrect.</p> <p>TYPE: The type of Modbus register is incorrect.</p>							

subindex	2
describe	Set Modbus register address
Data type	UNSIGNED16
access permissions	RW
PDO mapping	no
value range	0 ~ 0xFFFF
default value	0

subindex	3
Describe	Set the number of significant digits of the Modbus register
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	0 ~ 32
Default Value	0
Remarks	<p>0: Indicates an invalid value. This value must be rewritten when used.</p> <p>When the Modbus register type is set to discrete input or coil, the value</p> <p>The range is 1 ~ 32, indicating the number of discrete inputs or coils to be read.</p> <p>When setting the Modbus register type to input register or holding register</p> <p>When the value is 16 or 32, it indicates the input register or holding register</p> <p><small>Number of digits</small></p>

subindex	4
	Set the Modbus slave ID
Describing Data Types	UNSIGNED8

access permission	RW
PDO Mapping	no
Value Range	1 ~ 247
Default Value	0
Remarks	0: Invalid value, this value must be rewritten when used

subindex	5
describe	Set the Modbus register type
Data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	0 ~ 4
Default Value	0
Remarks	0: Invalid value, this value must be rewritten when used  1: Coil  2: Discrete input  3: Input register  4: Holding register

subindex	6
Describe	Enable
data type	UNSIGNED8
access permissions	RW
PDO Mapping	no
Value Range	0~1
Default Value	0
Remarks	0: Do not read the register set for this function.  1: Read the register set for this operation

subindex	7
Describe	Communication error count
data type	UNSIGNED16
access permissions	RW
PDO mapping	no
value range	0~65535
default value	0

Remark	
--------	--

**3.5.18 Firmware Date (0x4000)**

Index	0x4000
Name	Firmware Date
Object Type	WAS
Data Type	UNSIGNED32
Access Rights	RO
PDO Mapping	no
Value Range	0 ~ 0xFFFFFFFF
Default Value	0x20191107
Remarks	Year, month, day in hexadecimal characters

**3.5.19 Serial Number (0x4001)**

Index	0x4001
Name	serial number
Object Type	ARRAY
Data Type	UNSIGNED32
Access Rights	RO
PDO Mapping	no

subindex	0
Describe	Number of data
data type	UNSIGNED8
access permissions	RO
PDO mapping	no
value range	2
default value	2

subindex	1 ~ 2
Describe	Data value sent via the serial port
data type	UNSIGNED32
access permissions	RO
PDO Mapping	no
Value Range	0 ~ 0xFFFFFFFF
Default Value	0

Remark	Each IO card has a unique serial number, which is 8 bytes and is stored in two 32-bit data.
--------	---

## 4 Quick Test Tools

---

The EcatIO\_Test.exe program provides all functions for quickly testing an IO card.

### 4.1 Drivers

To use this program to test the IO card, you must first install the driver. The driver installation program is in the accompanying CD, or you can get it from the CD-ROM information downloaded from the official website. The driver installation program is named IMCdrv\_Ins.exe. Windows 7 and Windows 10 systems require administrator privileges to install the driver, so you need to right-click the exe and select "Run as Administrator" to run the installation program. When you run the installation program, a dialog box like "Figure 4-1 Driver Installation Program" appears:



Figure 4-1 Driver installation program

Click the [Install] button to install directly. After a few seconds, the driver installation is completed, as shown in "Figure 4-2 Driver

The "Automatic Installation Completed Dialog Box" is shown as follows:



Figure 4-2 Driver installation completed dialog box

### 4.2 Test Tool Description

The interface of the test tool program startup is shown in the figure below:



EcatIO卡测试工具

设备

网卡选择: Intel(R) 82574L Gigabit Network Connection 搜索从站 IO卡名称:

硬件版本: 00000 软件版本: 0000 日期: 0000.00.00 序列号: 0000000000000000

IO测试 串口(自定义协议)测试 串口(Modbus协议)测试 设置

输入端	输入值	输入端	输入值	输出端	输出值	输出端	输出值	模拟量	模拟值	模拟值(换算)	计数器	计数值
IN0:	0	IN8:	0	OUT0:	0	OUT8:	0	AD1:	0	0.000	CNT1:	0
IN1:	0	IN9:	0	OUT1:	0	OUT9:	0	AD2:	0	0.000	CNT2:	0
IN2:	0	IN10:	0	OUT2:	0	OUT10:	0	DA1:	0	0.000	ENC:	0
IN3:	0	IN11:	0	OUT3:	0	OUT11:	0	DA2:	0	0.000		
IN4:	0	IN12:	0	OUT4:	0	OUT12:	0	PWM1freq:	0	0.000		
IN5:	0	IN13:	0	OUT5:	0	OUT13:	0	PWM2freq:	0	0.000		
IN6:	0	IN14:	0	OUT6:	0	OUT14:	0	PWM1prop:	0	0.000%		
IN7:	0	IN15:	0	OUT7:	0	OUT15:	0	PWM2prop:	0	0.000%		

All input and output monitoring

模拟量输出

模拟量输出1: 5 V 输出 模拟量输出2: 5 V 输出

Test analog output

PWM输出

频率1: 1000.0 Hz/s 频率2: 1000.0 Hz/s

占空比1: 50 % 占空比2: 50 %

Testing PWM Output

开关量输出

OUT0 OUT1 OUT2 OUT3 OUT4 OUT5 OUT6 OUT7 全1

OUT8 OUT9 OUT10 OUT11 OUT12 OUT13 OUT14 OUT15 全0

Test switch output

清零计数器值

一键测试

DA输出: 每 2 秒增加电压 1 V

开关量输出: 每隔 0.5 秒改变一次状态

## 5 twincat test

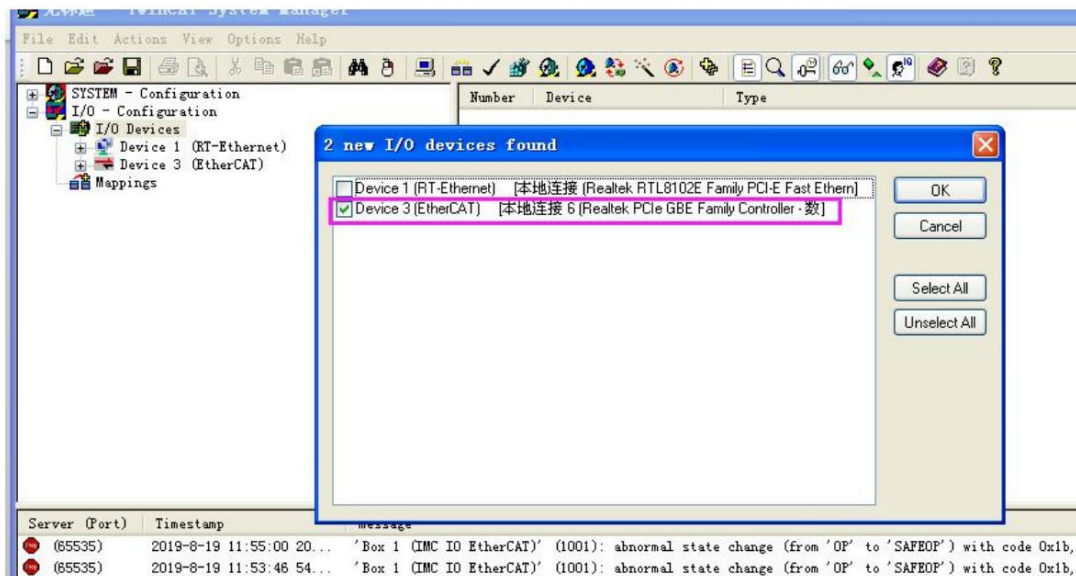
First, before testing, please put the xml into the "Io\EtherCAT" file in the twincat directory.

folder; then open twincat and enter system manager (note: the version of twincat is v2.11.2301, the test methods for other versions are similar).



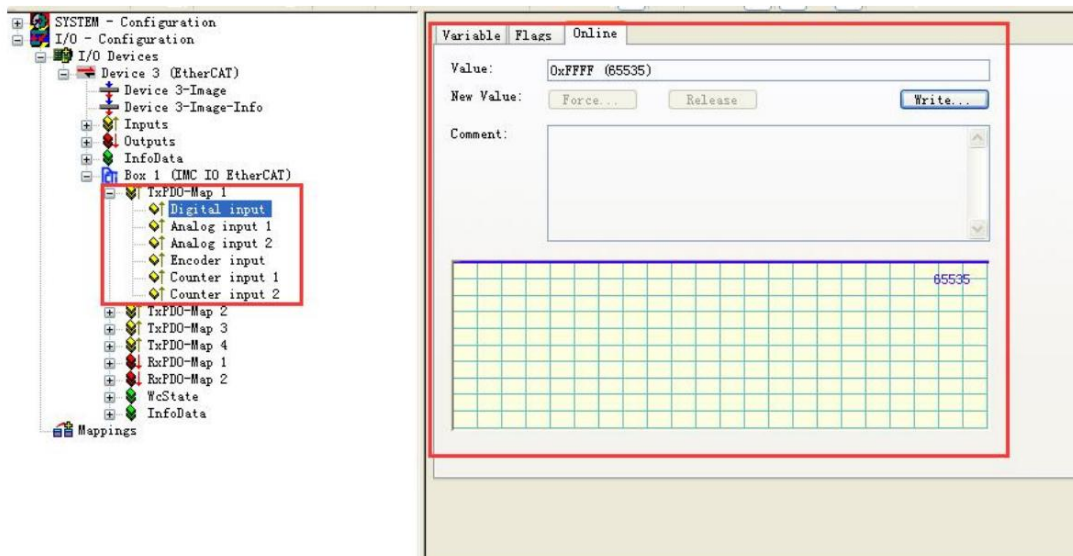
Right-click I/O Devices, then click Scan Devices to find the corresponding network device.

Select the correct connection and click OK to run.



### 5.1 Input Test

The input part is in the TxPDO-Map 1 part, as shown in the figure, where the corresponding Value part in Online can read the real-time value:



### 6.1.1 Switch input test The

specific real-time value of the input will be displayed in TxPDO-Map 1 in Box 1. After connecting the switch, you can see the value change of Digital input 1 in the figure above when the switch is turned on and off. For a simple test, you can first connect 24V+ to the COM port, and then use 24V- to touch the input port to see the input change. (You can see the real-time value of the input in Online in Digital input 1);

### 6.1.2 Analog input test In

TxPDO-Map 1, there are two analog inputs, Analog input 1 and Analog input 2. The test steps for both are the same, so only one is selected for detailed testing. The range of analog input is 0~10V, and accordingly, the volts will be linearly converted to values of 0~0xFFFFh. During the test, the DA0 port can be directly connected to the DA0 and DA1 ports. When DA0 is changed, the change of analog input can be seen (the change of its value can be seen in the Online of Analog input 1 or Analog input 2);

### 5.1.3 Encoder test In TxPDO-

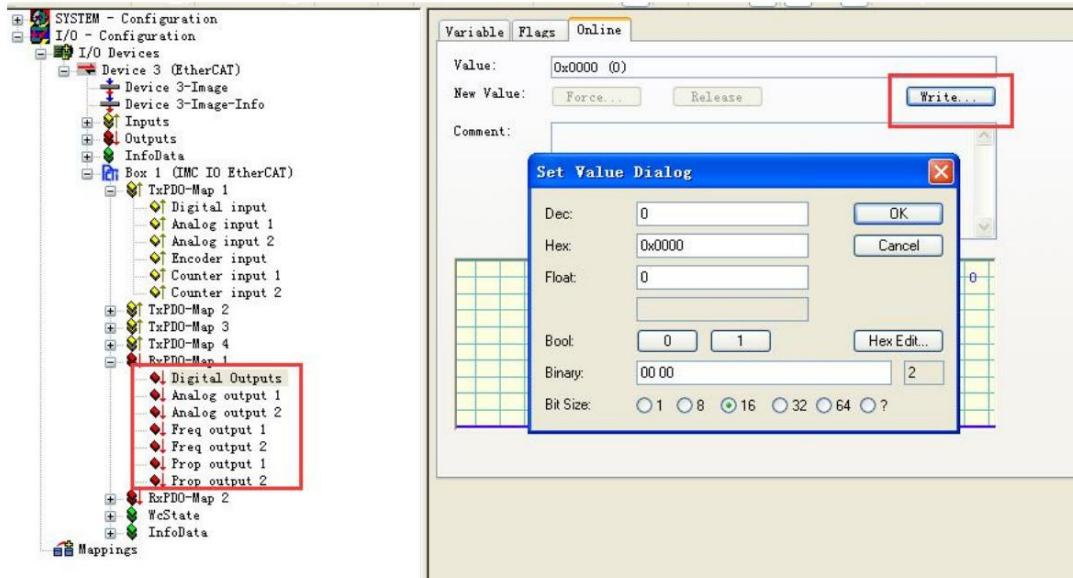
Map 1, you can see the Encoder input. This object is the real-time value of the encoder. After connecting to the handwheel and other devices, you can see the corresponding reading changes by shaking the handwheel (in Online in Encoder input);

### 5.1.4 Counter test In TxPDO-

Map 1, you can see Counter input1 and Counter input2, which are the real-time values of Counter 1 and Counter 2. After connecting the corresponding 5V switch to one of the input ports, repeatedly disconnect and connect the switch to see the real-time changes in the counter value. If you just want to do a simple test, connect the 5VOUT of the encoder to C+, and touch GND with C-, and release it. Repeat this process to see the reading change (you can read the change in Online of Counter input 1 or Counter input 2).

## 5.2 Output Test

In RxPDO-Map1, some outputs can be operated. In the Online of the corresponding items (such as switch output), the set value can be written. Click Write, as shown in the figure:



### 5.2.1 Switching output test

The real-time displayed output value is in RxPDO-Map 1. Since the output port is readable and writable in RxPDO-Map 1, the output port can also be controlled in RxPDO-Map 1. When testing, you can connect 24V- (the ground of the power supply) to OGND, then open a certain output port, and use a small relay to connect 24V+ and the output port. When the output changes, you can observe the value changes in twincat and the changes in the relay;

### 5.2.2 Analog output test

In RxPDO-Map 1, you can set the analog output, the range is from 0~10V, the corresponding value is 0~0x0FFFh, write the value in the Values of Online in Analog output to change it, when testing, you can use a multimeter to connect one end to DA and the other end to GND to measure the voltage. Or directly connect DA to AD, and you can read the value from the analog input;

### 5.2.3 PWM Test

In RxPDO-Map 1, you can see Freq output 1/2 and Prop output 1/2, which correspond to the PWM frequency and PWM duty cycle. After setting, connect the PWM output port to an oscilloscope to see the waveform changes (for example, write 655360 to Freq output and 32768 to Prop output to see a waveform with a frequency of 10 and a duty cycle of 50%).

## 5.3 Other tests

There is a serial port on the IO card, which can be connected to devices with modbus protocol or non-standard protocol (if it is non-standard, it needs to be pre-configured, the default is standard modbus protocol).

The configuration in can be done through addresses 0x3000~0x3007 and 0x3020~0x3027, as shown in the figure:

+ 3000:0	set modbus send Reg 1	RW	> 6 <
+ 3001:0	set modbus send Reg 2	RW	> 6 <
+ 3002:0	set modbus send Reg 3	RW	> 6 <
+ 3003:0	set modbus send Reg 4	RW	> 6 <
+ 3004:0	set modbus send Reg 5	RW	> 6 <
+ 3005:0	set modbus send Reg 6	RW	> 6 <
+ 3006:0	set modbus send Reg 7	RW	> 6 <
+ 3007:0	set modbus send Reg 8	RW	> 6 <
+ 3020:0	set modbus receive Reg 1	RW	> 6 <
+ 3021:0	set modbus receive Reg 2	RW	> 6 <
+ 3022:0	set modbus receive Reg 3	RW	> 6 <
+ 3023:0	set modbus receive Reg 4	RW	> 6 <
+ 3024:0	set modbus receive Reg 5	RW	> 6 <
+ 3025:0	set modbus receive Reg 6	RW	> 6 <
+ 3026:0	set modbus receive Reg 7	RW	> 6 <
+ 3027:0	set modbus receive Reg 8	RW	> 6 <

### 5.3.1 485 and modbus communications

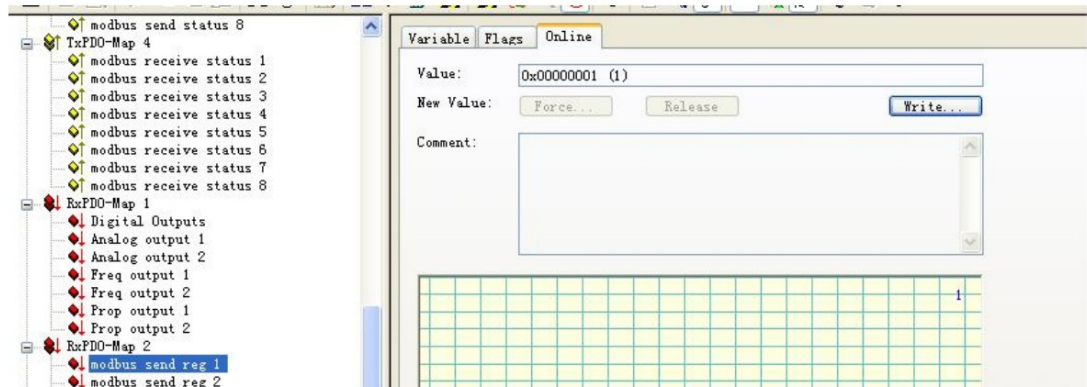
On the IO card, there is a 485 interface dedicated to serial communication, which can be used to communicate with other 485 serial devices. Before communication, you need to set the serial port, including the ID of the slave station, the register address, the number of valid bits of the register, and whether to enable the read and write of the address. If it is not enabled, the register of this address cannot be read and written (Note: All configurations must be written before the address read and write is enabled, and they cannot be configured after it is enabled). After configuration, you can control the register by writing 0x2600, or obtain the status of the register by reading 0x2601.

Example: Write a value of 1 to the coil register with the slave address of 1 and address 0. First, set the read and write enable to be closed, that is, 0x3000.7 is 0; secondly, write the address 0 of the register to 0x3000.3; write the effective number of bits of the register to 1 at 0x3000.4; write the ID of the slave to 0x3000.5, and the value is 1; write the type of the register to 0x3000.6 as coil (value is 1); write the read and write enable of the register to 0x3000.7 (value is 1); as shown in the figure:

+ 2600:0	Modbus send data reg	RW	> 8 <
+ 2601:0	Modbus receive data reg	RW	> 8 <
+ 3000:0	set modbus send Reg 1	RW	> 6 <
3000:01	op status	RW	0x0002 (2)
3000:02	reg address	RW	0x0001 (1)
3000:03	reg bit	RW	0x01 (1)
3000:04	slave id	RW	0x01 (1)
3000:05	reg type	RW	0x01 (1)
3000:06	run	RW	0x01 (1)
+ 3001:0	set modbus send Reg 2	RW	> 6 <
+ 3002:0	set modbus send Reg 3	RW	> 6 <

Finally, send the PDO mapping to 0x2600.1 (i.e. modbus send reg1 in RxPDO-Map 2)

Just write the value 1 in the , as shown in the figure:





## 6 Using iMC6 Series to Test IO Cards

If you have our company's iMC6 series motion control card, you can use this card to set and read the parameters of the IO card. The preparation is the same as connecting an ordinary IO card. First, put the xml into the xml folder of our iMC6 series software, then use our iMCAT-6xxE software to search, connect, and configure. Select all PDO mappings, as shown below (Note: If you want to conduct the following experiment, do not check "Map IMC parameters". This function maps PDO parameters to the registers of our control card, which will make the PDO write function invalid): Since all read and write function test methods are similar, only digital output and digital input are selected for testing, and other tests will not be repeated.

配置从站: (0) IMC IO MODEL with EtherCAT

设置别名:  (最多8个字符, 主要用于API函数)

此从站是驱动器设备

PDO映射设置: (注意: 列表中的【id】列, 用于对从站中对应的寄存器对象进行控制)

选择输入输出PDO映射

In/Out	id	sm	pdo	index	subindex	bitlen	映射IMC参数	IMC参数类型	默认值 (HEX)	描述
out	1	2	0x1600	0x2300	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Digital Outputs
out	2	2	0x1600	0x2310	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog output 1
out	3	2	0x1600	0x2311	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog output 2
out	4	2	0x1600	0x2320	0	32	<input type="checkbox"/>	由系统分配	0x00000000	Freq output 1
out	5	2	0x1600	0x2321	0	32	<input type="checkbox"/>	由系统分配	0x00000000	Freq output 2
out	6	2	0x1600	0x2330	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Prop output 1
out	7	2	0x1600	0x2331	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Prop output 2
out	8	2	0x1601	0x2600	1	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 1
out	9	2	0x1601	0x2600	2	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 2
out	10	2	0x1601	0x2600	3	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 3
out	11	2	0x1601	0x2600	4	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 4
out	12	2	0x1601	0x2600	5	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 5
out	13	2	0x1601	0x2600	6	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 6
out	14	2	0x1601	0x2600	7	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 7
out	15	2	0x1601	0x2600	8	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 8
in	1	3	0x1A00	0x2000	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Digital input 1
in	2	3	0x1A00	0x2010	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog input 1
in	3	3	0x1A00	0x2011	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog input 2

### 6.1 Testing digital output

Select the "Bus Input and Output" section in the Function Test module, select the corresponding slave station, and set its corresponding PDO mapping ID to 1, write the output value, and set it. You can see the changes in the indicator light on the IO card, as shown in the figure:

参数配置 零点 点动 点到点 直线插补 圆弧插补 龙门驱动 电子齿轮/手轮 开关量输入输出 间隙补偿 总线输入输出 SDO请求

从站Position: 0 : IMC IO MODEL with EtherCAT

寄存器id: 1 (只能输入从站配置过的输入输出ID)

寄存器值: 0000AAAA ☐ 十进制 ☒ 十六进制 ☐ 单精度

设置输出 获取输入

注意: 此处的功能需要根据您配置到控制卡中的从站配置来操作。

### 6.2 Testing digital input

Select the corresponding slave station and set its corresponding PDO mapping ID to 1. When the input switch state is on, read the input value and you can see the change of the register value, as shown in the figure:

参数配置

零点

点动

点到点

直线插补

圆弧插补

龙门驱动

电子齿轮/手轮

开关量输入输出

间隙补偿

总线输入输出

SDO请求

从站Position:

0 : IMC IO MODEL with EtherCAT

寄存器id:

1

(只能输入从站配置过的输入输出ID)

寄存器值:

00000001

☐ 十进制

☒ 十六进制

☐ 单精度

设置输出

获取输入

注意: 此处的功能需要根据您配置到控制卡中的从站配置来操作。