Modbus

User Manual

MOONS'
moving in better ways

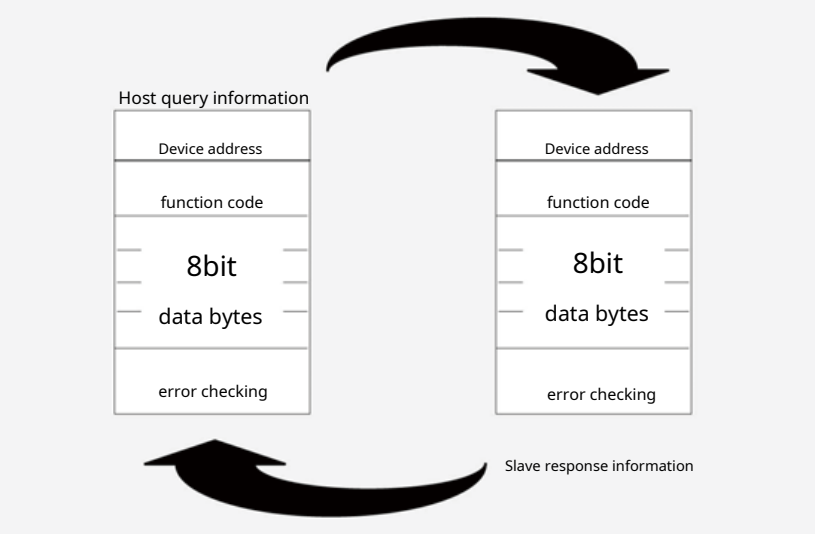
Shanghai Anpu Mingzhi Automation Equipment Co., Ltd.

Version:1.0
3/1/2022

Table of contents

1 ModbusIntroduction to Communication Protocol

ModbusThe communication protocol was firstModiconcompany development,is a master-slave communication mode,That is, only the host can initiate the request,The slave device responds by providing the requested data to the master or by performing the operation requested in the query,Bus protocol widely used in industrial field. Master-slave query - the response mechanism is shown in the figure below.



The host can access the slave device by specifying the node address,It is also possible to access all slave devices by sending a broadcast message;The slave device only responds to its own inquiries,will not respond to broadcast queries.

ModbusThe communication protocol is an application layer message transmission protocol,includeRTU,ASCIIandTCP,standardModbusThe protocol physical layer interface includesRS232,RS485and ethernet.

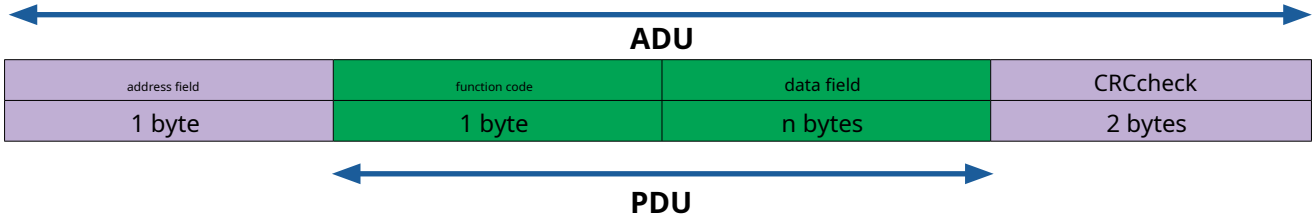
Modbus RTUandModbus ASCIIBoth are serial transmission.in,Modbus RTUAdopt binary representation and compact data structure,higher communication efficiency,Wider application.andModbus ASCIIuseASCIIcode transmission,And use special characters as the start and end identifiers of its bytes,Its transmission efficiency is much lower thanModbus RTUprotocol.

Modbus TCPvia Industrial EthernetTCP/IPnetwork transmissionModbuscommunication.ModbusData transfer provides connection over Ethernet TCP/IP Real-time communication between client and server on the network.

2 Modbusdata frame structure

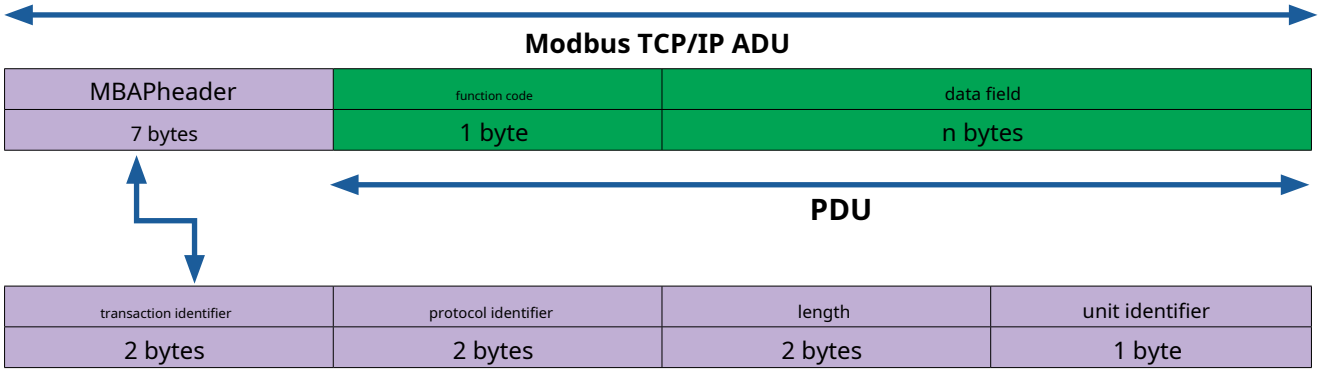
2.1 Modbus RTUdata frame structure

Modbus RTUprotocol through the slave device address(or broadcast),Define the function code of the requested operation,data to be sent andCRCChecksum is put into host query to build query message;The response message of the slave device is also usedModbus RTUmessage structure,Include slave device address, Function code of the requested operation,data to be sent andCRCcheck;If an error occurs while receiving the message,or the slave device cannot perform the requested operation,The slave device will send an exception message in response.Modbus RTUThe data frame structure is as follows:



2.2 Modbus TCPdata frame structure

Modbus TCPagreement is inModbus RTUjoin the agreementMBAP(Modbus Application Protocol Header)header,becauseTCPis a reliable connection based service,so inModbus TCPnot in the agreementCRCcheck,allModbus TCP ADUBoth sending and receiving are usingTCPTransmission Control Protocol,port number is502.Modbus TCPThe data frame structure is as follows:



area	describe	Client computer	server
transaction identifier	ModbusRequest/Response Office ID code	client startup	The server copies from the received request
protocol identifier	0expressModbusprotocol	client startup	The server copies from the received request
length	Includes unit identifier and data fields bytes	client startup(ask)	server(response)start up
unit identifier	Slave device address	client startup	The server copies from the received request

transaction identifier:pairing for transaction processing,ModbusThe server replicates the requested transaction identifier in the response. protocol identifier:ModbusThe protocol identifier is0x0000. length:including the unit identifier andPDU bytes,The unit is bytes.

unit identifier:serverIPaddress identification code,by the requestModbusClient Settings,The server must return the same value in the response .

Remark:

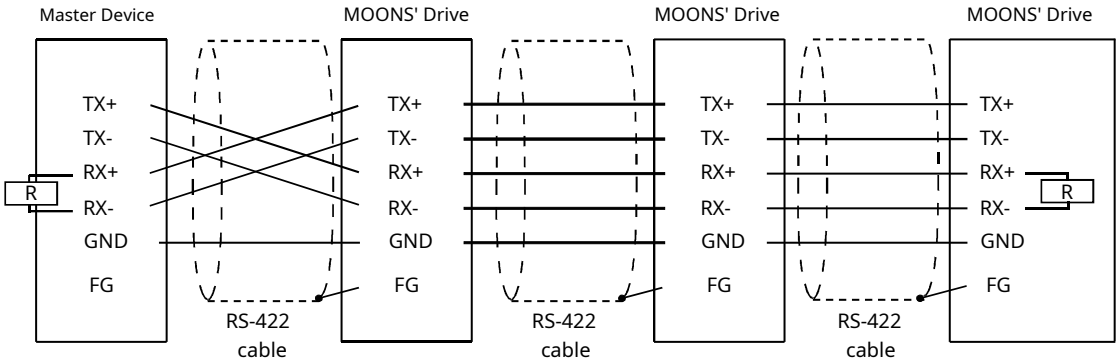
ADU:Application Data Unit **PDU:Protocol Data Unit**

3 Modbuswiring

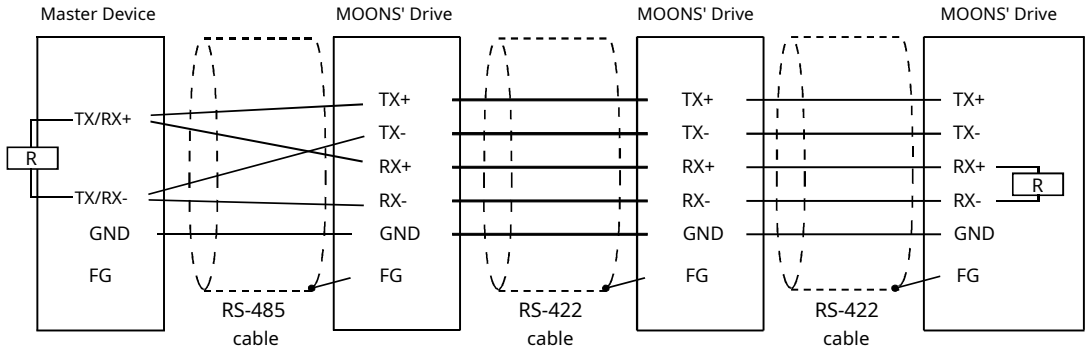
3.1 Modbus RTUwiring

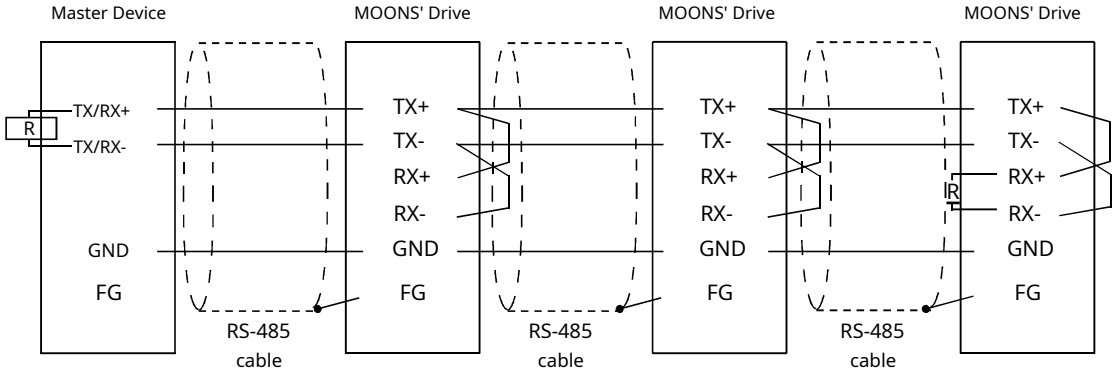
Modbus RTUprotocol usageRS-232orRS-485physical layer,useRS-485The physical layer can be configured1~32slave device address, constituteRS-422/RS-485Network topology,Usually connected in parallel at the end of the physical connection120Ωterminating resistor.Modbus RTUSupport full-duplex and half-duplex connection,Usually we recommend using full-duplex wiring to build a communication network.

RS422The full-duplex four-wire wiring method is shown in the figure:



RS485The half-duplex two-wire wiring method is shown in the figure:





Remark:**R**express**120Ω**terminating resistor

3.2 Modbus TCPwiring

supportModbus TCPCommunication protocol products useRJ45Connector,meets the100BASE-TX(100Mbps),standard can be used 5above class,Straight-through network cable connection.

4 Modbusconfigure

4.1 Modbus RTUconfigure

fromMOONS'website(www.moons.com.cn)Download the debugging software for the corresponding product,Configuration via softwareModbus RTUCommunication parameters.

4.1.1Host communication parameter configuration

1.baud rate:Consistent with the slave device

2.data bits:8bit data bit

3.stop bit:1bit stop bit

4.Check Digit:No check digit

4.1.2Slave deviceMOONS'Drive communication parameter configuration

1.Slave device address

in the same network,Each slave device has a unique address,Only slave devices that meet the address requirements will respond to commands from the master device.Modbus address"0"is the broadcast address,cannot be used as a slave address,Modbus RTUThe address range of the slave device under the communication protocol is1~32.Install the driver for the communication address setting rotary switch,Then the address is set by the rotary switch;If the address setting rotary switch is not installed on the drive,Then set it through the configuration software,or bySCLinstructionDAmake settings.

2.baud rate

The master and slave devices must set the same baud rate.Install the driver for the baud rate setting DIP switch,Then the baud rate is set by the DIP switch;If the baud rate setting DIP switch is not installed on the drive,Then set it through the configuration software,or bySCLinstructionBRmake settings,BRThe corresponding relationship between the command parameter value and the baud rate is as follows:

1:9600bps
2:19200bps
3:38400bps
4:57600bps
5:115200bps

3.Power-on working mode

Configure the working mode of the drive after power-on.passSCLinstructionPMmake settings,Modbus RTUin communication modePMThe corresponding relationship between the command parameter value and the working mode is as follows:

8:The driver is in the enabled operating mode after power-on

9:The driver automatically executes after power-onQThe working mode of the

program 10:The driver is in a non-enabled working mode after power-on

4.Protocol

configure32Bit data high and low byte order andRS-485Communication full-duplex/half-duplex connection.Setup via configuration software,or bySCLinstructionPRmake settings,PRThe corresponding relationship between the command parameter value and the communication protocol is as follows:

5:Half-duplex connection,Big Endian 133:
Half-duplex connection,Little Endian 261:
Full duplex connection,Big Endian 389:full
duplex connection,Little Endian Note:

Big Endianexpress32data high16Bit data row at the low address end of memory,Low16Bit data rows are placed at the high address end of memory

Little Endianexpress32low data16Bit data row at the low address end of memory,high16Bit data rows are placed at the high address end of memory

4.2 Modbus TCPconfigure

fromMOONS'website(www.moons.com.cn)Download the debugging software for the corresponding product,Configuration via softwareModbus TCPCommunication parameters.

4.2.1Client communication parameter configuration

1.Client computerIPaddress:on the same network segment as the serverIPaddress

2.The port number:useModbus TCPThe port number of the communication server,fixed as502

4.2.2serverMOONS'Drive communication parameter configuration

1.serverIPaddress

in the same network,Each slave device has a uniqueIPaddress,Only the server that meets the address requirements will respond to the client's command.InstallIP Address selection switch driver,butIPThe address is set by the switch;If not installed on the driveIPaddress selector switch,Then set it through the configuration software,The configuration software allows you to change the switch corresponding to each gearIPaddress.

2.Power-on working mode

Configure the working mode of the drive after power-on.passSCLinstructionPMmake settings,Modbus TCPin communication modePMThe corresponding relationship between the command parameter value and the working mode is as follows:

8:The driver is in the enabled operating mode after power-on

9:The driver automatically executes after power-onQThe working mode of the

program 10:The driver is in a non-enabled working mode after power-on

3.Protocol

configure32Bit data high and low byte order.Setup via configuration software,or bySCLinstructionPRmake settings,PRThe corresponding relationship between the command parameter value and the data encoding is as follows:

5:Big Endian

133:Little Endian

Note:

Big Endianexpress32data high16Bit data row at the low address end of memory,Low16Bit data rows are placed at the high address end of memory

Little Endianexpress32low data16Bit data row at the low address end of memory,high16Bit data rows are placed at the high address end of memory

5 ModbusSupported function codes

MOONS'drive supportedModbusThe function code is as follows:

0x03:read holding register

0x06:write a single register

0x10:write multiple registers

5.1function code0x03:read holding register

Read single or multiple holding registers,read at most50registers,Broadcast commands are not supported.

example:Read the slave address as1drive status,The address of this register is40002,Suppose the value of the register is0x0009.

Host sends data:Command Message(Master)				Slave returns data:Response Message(Slave)		
Function	data	number of bytes		Function	data	number of bytes
Slave Address Slave address	01H	1		Slave Address Slave address	01H	1
Function Code function code	03H	1		Function Code function code	03H	1
Starting Data Address data start address (register40002)	00H(High) 01H(Low)	2		Number of Data (In Byte) number of data	04H	1
Number of Data (In word) number of data	00(High) 01(Low)	2		Content of Starting Data Address 40002 initial address40002The data	00H(High) 09H(Low)	2
CRC Check Low CRCcheck low byte	D5H	1		CRC Check Low CRCcheck low byte	78H	1
CRC Check High CRCcheck high byte	CAH	1		CRC Check High CRCcheck high byte	42H	1

Host sends:01 03 00 01 00 01 D5 CA Return

from the station:01 03 02 00 09 78 42

The format of the data returned by the exception is:01 83 XX CRC_L

CRC_H in,XX = 01H:Does not support reading function codes03H XX =

02H:invalid register XX = 03H:illegal data area XX = 11H:Register does

not support reading

Host sendsModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 01	00 01	D5 CA
Modbus TCP	00 00 00 00 00 06 01	none	03	00 01	00 01	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	number of bytes	data content	CRCcheck
Modbus RTU	none	01	03	02	00 09	78 42
Modbus TCP	00 00 00 00 00 05 01	none	03	02	00 09	none

5.2function code0x06:write a single register

Write a single holding register,When using broadcast commands,All slave devices on the bus perform writes to the same registers.

example:The address of the slave device is11The drive writes the motor running speed,The address of this register is40030,Suppose the motor speed is set to be 12.5rps,then write data bits12.5 x 240 = 3000,convert to16base as12CH.

Host sends data:Command Message(Master)				Slave returns data:Response Message(slave)		
Function	data	number of bytes		Function	data	number of bytes
Slave Address Slave address	0BH	1		Slave Address Slave address	0BH	1
Function Code function code	06H	1		Function Code function code	06H	1
Starting Data Address data start address (register40030)	00H(High) 1DH(Low)	2		Starting Data Address data start address (register40030)	00H(High) 1DH(Low)	2
Content of Data data content	01(High) 2C(Low)	2		Content of Data data content	01(High) 2C(Low)	2
CRC Check Low CRCcheck low byte	19H	1		CRC Check Low CRCcheck low byte	19H	1
CRC Check High CRCcheck high byte	2BH	1		CRC Check High CRCcheck high byte	2BH	1

Host sends:0B 06 00 1D 01 2C 19 2B Return

from the station:0B 06 00 1D 01 2C 19 2B

The format of the data returned by the exception is:0B 86 XX

CRC_L CRC_H in,XX = 01H:Writing function codes is not supported

06H XX = 02H:invalid register XX = 03H:illegal data area XX = 12H:

Register does not support writing XX = 13H:Set value out of range

Host sendsModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	0B	06	00 1D	01 2C	19 2B
Modbus TCP	00 00 00 00 00 06 0B	none	06	00 1D	01 2C	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	0B	06	00 1D	01 2C	19 2B
Modbus TCP	00 00 00 00 00 06 0B	none	06	00 1D	01 2C	none

5.3function code0x10:write multiple registers

write single or multiple holding registers,Write at most50registers;When using broadcast commands,All slaves on the bus perform writes to the same registers.

example:The address of the slave station is10The drive writes the target distance,The address of this register is40031and40032,Suppose the set target distance is 30000,convert to16base as7530H.byBig Endiancode transmission.

Host sends data:Command Message(Master)		
Function	data	number of bytes
Slave Address Slave address	0AH	1
Function Code function code	10H	1
Starting Data Address data start address (register40031)	00H(High) 1EH(Low)	2
Number of Data (In word) number of data	00H(High) 02H(Low)	2
Number of Data (In byte) number of data	04H	1
Content of First Data Address The data content of the first address	00H(High) 00H(Low)	2
Content of Second Data Address The data content of the second address	75H(High) 30H(Low)	2
CRC Check Low CRCcheck low byte	70H	1
CRC Check High CRCcheck high byte	8FH	1

Slave returns data:Response Message(slave)		
Function	data	number of bytes
Slave Address Slave address	0AH	1
Function Code function code	10H	1
Starting Data Address data start address (register40031)	00H(High) 1EH(Low)	2
Number of Data (In word) number of data	00(High) 02(Low)	2
CRC Check Low CRCcheck low byte	20H	1
CRC Check High CRCcheck high byte	B5H	1

Host sends:0A 10 00 1E 00 02 04 00 00 75 30 70 8F Return from the station:0A 10 00 1E 00 02 20 B5

The format of the data returned by the exception is:0A 90 XX

CRC_L CRC_H in,XX = 01H:Writing function codes is not supported

10H XX = 02H:invalid register XX = 03H:illegal data area XX = 12H:

Register does not support writing XX = 13H:Set value out of range

Host sendsModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	0A	10	00 1E	00 02	04	00 00 75 30	70 8F
Modbus TCP	00 00 00 00 00 0B 0A	none	10	00 1E	00 02	04	00 00 75 30	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register ground site	Number of registers quantity	CRCcheck
Modbus RTU	none	0A	10	00 1E	00 02	20 B5
Modbus TCP	00 00 00 00 00 06 0A	none	10	00 1E	00 02	none

6 SCLInstruction code table

6.1opcode

Modbusregister in register table40125is defined as the opcode register,Towards40125register is written to the corresponding opcode,That is, the action that executes the corresponding opcode,The supported opcodes are as follows:

SCL Command Encoding Table							
Function	SCL	Opcode	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Alarm Reset	AX	0xBA	×	×	×	×	×
Start Jogging	CJ	0x96	×	×	×	×	×
Stop Jogging	SJ	0xD8	×	×	×	×	×
Encoder Function[1*]	EF	0xD6	0,1,2 or 6	×	×	×	×
Encoder Position	EP	0x98	Position	Position	×	×	×
Feed to Length with Speed Change[2*]	FC	0x6D	I/O Point	Condition	×	×	×
Feed to Double Sensor[1*]	FD	0x69	I/O Point 1	Condition 1	I/O Point 2	Condition 2	×
Follow Encoder[1*]	FE	0xCC	I/O Point	Condition	×	×	×
Feed to Length	FL	0x66	×	×	×	×	×
Feed to Sensor with Mask Distance	FM	0x6A	I/O Point	Condition	×	×	×
Feed and Set Output	FO	0x68	I/O Point	Condition	×	×	×
Feed to Position	FP	0x67	×	×	×	×	×
Feed to Sensor	FS	0x6B	I/O Point	Condition	×	×	×
Feed to Sensor with Safety Distance	FY	0x6C	I/O Point	Condition	×	×	×
Jog Disable[1*]	JD	0xA3	×	×	×	×	×
Jog Enable[1*]	JE	0xA2	×	×	×	×	×
Motor Disable	MD	0x9E	×	×	×	×	×
Motor Enable	ME	0x9F	×	×	×	×	×
Seek Home	SH	0x6E	I/O Point	Condition	×	×	×
Set Position	SP	0xA5	Position	Position	×	×	×
Filter Input[1*]	FI	0xC0	I/O Point	Filter Time	×	×	×
Filter Select Inputs[1*]	FX	0xD3	×	×	×	×	×
Step Filter Freq[1*]	SF	0x06	Frequency	×	×	×	×
Analog Deadband[1*]	AD	0xD2	0.001 V	×	×	×	×
Alarm Reset Input[1*]	AI	0x46	Function ('1'..'3')	I/O Point	×	×	×
Alarm Output[1*]	AO	0x47	Function ('1'..'3')	I/O Point	×	×	×
Analog Scaling[1*]	AS	0xD1	×	×	×	×	×
Define Limits[1*]	DL	0x42	1..3	×	×	×	×

Full Closed-loop Control Switch	XM	0x54	0..1	×	×	×	×
Set Output	SO	0x8B	I/O Point	Condition	×	×	×
Wait for Input	WI	0x70	×	×	×	×	×
Queue Load & Execute	QX	0x78	1..12	×	×	×	×
Wait Time	WT	0x6F	0.01 sec	×	×	×	×
Find Home[3*]	FH	0xDB	- 4..35	×	×	×	×
Stop Move & Kill Buffer, Max Decel	SK	0xE1	×	×	×	×	×
Stop Move & Kill Buffer, Normal Decel	SKD	0xE2	×	×	×	×	×

Remark:in the form "×" symbol means not to use

[1*]:**M3**Series products do not support this opcode [2*]:Only applies to**STF-D**,**M3**series products [3*]:Only applies to**M3**series products

example:existMOONS'productSCLin the instruction "FL"The command indicates the execution of relative position control,existModbusinside,to register40125write "0x66"(which isFLin the code tableOpcode)i.e. perform relative position control.

Detailed opcode functions,Please refer toHOST COMMAND REFERENCEmanual.

6.2 I/Oinstruction code

Encoding of digital input/output ports and states,The specific codes are as follows.

character	hex	describe
'0'	0x30	EncoderZPhase signal
'1'	0x31	Digital input/output ports1
'2'	0x32	Digital input/output ports2
'3'	0x33	Digital input/output ports3
'4'	0x34	Digital input/output ports4
'5'	0x35	Digital input/output ports5
'6'	0x36	Digital input/output ports6
'7'	0x37	Digital output port7
'8'	0x38	Digital output port8
'9'	0x39	Digital output port9
':'	0x3A	Digital output port10
';'	0x3B	Digital output port11
'<'	0x3C	Digital output port12
'L'	0x4C	Low level (optical coupler on)
'H'	0x48	High level (optical coupler disconnected)
'R'	0x52	Signal rising edge
'F'	0x46	Signal falling edge

example:existMOONS'productSCLin the instruction "FS1F",existModbusinside,to register40125write "0x6B",40126 write "0x31",40127write "0x46"i.e. perform the same control.

7 Modbusroutine

MOONS'Different product series,ModbusRegister addresses vary;in use,The following routine can change the register address according to the product series.

7.1Position Control Routine

The slave device address is1,set acceleration,deceleration,speed and target position,correspondMOONS' SCLThe instructions are as follows:

SCLinstruction	set value	unit	register address	Hex format register address	write register device value	illustrate
AC	100	Rps/sec	40028	00 1B	600	The preset acceleration is100,need to register40028 write600 (0x0258)
DE	100	Rps/sec	40029	00 1C	600	The preset deceleration is100,need to register40029 write600 (0x0258)
VE	1	Rps	40030	00 1D	240	The preset speed is1,need to register40030write 240 (0x00F0)
DI	200000	Counts	40031, 40032	00 1E, 00 1F	200000	The preset target position is200000,need to40031 and 40032register write200000 (0x00030D40)

but:

Notice:See the following message before,Please read the remarks first**1)**The conversion relationship in and the message format of Appendix I.

The acceleration of the pre-written planning curve (40028)=600(0x0258),deceleration(40029)=600(0x0258),speed (40030)=240(0x00F0),target location(40031,40032)=200000(0x00030D40).NoticePRdifferent modes,That32The difference between the high and low bits of the input data of the bit register, Its message is as follows:

likePR=5,which isBig Endianin mode: Host sends

Modbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 1B	00 05	0A	02 58 02 58 00 F0 00 03 0D 40	CD 83
Modbus TCP	00 00 00 00 00 11 01	none	10	00 1B	00 05	0A	02 58 02 58 00 F0 00 03 0D 40	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	00 1B	00 05	70 0D
Modbus TCP	00 00 00 00 00 06 01	none	10	00 1B	00 05	none

likePR=133,which isLittle Endianin mode: Host sends

Modbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 1B	00 05	0A	02 58 02 58 00 F0 0D 40 00 03	7B 9A
Modbus TCP	00 00 00 00 00 11 01	none	10	00 1B	00 05	0A	02 58 02 58 00 F0 0D 40 00 03	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	00 1B	00 05	70 0D
Modbus TCP	00 00 00 00 00 06 01	none	10	00 1B	00 05	none

Write Command Opcode Register (40125)data0x0066 (FL),That is to execute the relative position control host

to sendModbus RTU/TCPThe message is as follows://**perform relative motion,SCLinstructionFL**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 66	C8 38
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 66	none

Write Command Opcode Register (40125)data0x0067 (FP),That is, the absolute position control host

sendsModbus RTU/TCPThe message is as follows://**perform absolute motion,SCLinstructionFP**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 67	09 F8
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 67	none

Write Command Opcode Register (40125)data0x00E1(SK),i.e. execute stop control

Host sendsModbus RTU/TCPThe message is as follows://**stop at maximum deceleration,SCLinstructionSK**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 E1	88 5A
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 E1	none

Read the target position register (40031,40032),Its message is as follows:

Host sendsModbus RTU/TCPThe message is as follows://**read register40031,40032**//

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 1E	00 02	A4 0D
Modbus TCP	00 00 00 00 00 06 01	none	03	00 1E	00 02	none

Remark:

1. **Modbus**When the message is read and written,Pay attention to the conversion relationship between register addresses,as register**40125**converted to**0x007C**, which is **40125-40000-1=124(0x007C)**

2. **PR=5**pattern and**PR=133**Mode difference**:**

   in order to place the target**DI=200000**write to destination register**(40032 ,40031)**,immediately**32**Bit register address write **200000 (0x030D40)**

   - exist**PR=5**,which is**Big Endian**in mode,means write**32**bit data is high**16**Bit data row at the low address end of memory,Low **16**Bit data rows are placed at the high address end of memory
   - exist**PR=133**,which is**Little Endian**in mode,means write**32**bit data is low**16**Bit data row at the low address end of memory, high**16**Bit data rows are placed at the high address end of memory

3.speed,add**/**Deceleration register parameter setting value unit

- The unit of the set value of the speed register parameter is$\frac{1}{240}$**rps**
- add**/**The unit of deceleration register parameter setting value is$\frac{1}{6}$**rps/s**

7.2Speed Control Routine

The slave device address is1,set acceleration,deceleration and speed,correspondMOONS' SCLThe instructions are as follows:

SCLinstruction	set value	unit	register address	Hex format register address	write register device value	illustrate
JA	100	Rps/sec	40047	00 2E	600	The preset acceleration is100,need to register40047 write600 (0x0258)
JL	100	Rps/sec	40048	00 2F	600	The preset deceleration is100,need to register40048 write600 (0x0258)
JS	10	Rps	40049	00 30	2400	The preset speed is10,need to register40049write 2400 (0x0960)

but:

The acceleration of the pre-written planning curve (40047)data0x0258,deceleration(40048)data0x0258,speed(40049)data0x0960 Host

sendsModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 2E	00 03	06	02 58 02 58 09 60	20 23
Modbus TCP	00 00 00 00 00 0D 01	none	10	00 2E	00 03	06	02 58 02 58 09 60	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	00 2E	00 03	E0 01
Modbus TCP	00 00 00 00 00 06 01	none	10	00 2E	00 03	none

Write Command Opcode Register (40125)data0x0096 (CJ),i.e. execute startJogcontrol

Host sendsModbus RTU/TCPThe message is as follows://**pair register40125write00 96**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 96	C8 7C
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 96	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 96	C8 7C
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 96	none

Write Command Opcode Register (40125)data0x00D8(SJ),i.e. execution stopsJogcontrol

Host sendsModbus RTU/TCPThe message is as follows://**pair register40125write00 D8**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 D8	48 48
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 D8	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 7C	00 D8	48 48
Modbus TCP	00 00 00 00 00 06 01	none	06	00 7C	00 D8	none

7.3Home Control Routine

The slave device address is1,electronic gearEG=10000 Pulses/r,Set the back-to-origin acceleration/deceleration,first gear speed,Second gear speed and origin offset,and torque limit using hardware limit homing method,correspondMOONS' SCLThe instructions are as follows:

SCLinstruction	set value	unit	register address	Hex format register address	write register device value	illustrate
HA1	20	Rps/sec	40357,40358	01 64,01 65	120	The preset acceleration/deceleration is20,need to register 40357,40358write120 (0x00000078)
HL1	0	Rps/sec	40359,40360	01 66,01 67	0	reserved
HV1	5	Rps	40361, 40362	01 68,01 69	1200	The preset first gear speed is5,need to register 40361, 40362write1200 (0x000004B0)
HV2	1	Pulses	40363,40364	01 6A, 01 6B	240	The preset second gear speed is1,need to register 40363,40364write240 (0x000000F0)
HO	5000	Pulses	40365,40366	01 6C, 01 6D	5000	The default back-to-origin offset is5000,need to register 40365,40366write5000 (0x00001388)
HC	70	%	40279,40280	01 16, 01 17	700	The torque limit of the preset hardware limit return-to-origin method is: 70%,need to register40279,40280write700 (0x000002BC)

but:

Pre-written acceleration/deceleration for homing (40357, 40358)data0x00000078,first speed (40361, 40362)data0x000004B0, second speed (40363, 40364)data0x000000F0,Origin offset (40365, 40366)data0x00001388.NoticePRdifferent modes,That32The difference between the high and low bits of the input data of the bit register,Its message is as follows:

likePR=5,which isBig Endianin mode: Host sends

Modbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	01 64	00 0A	14	00 00 00 78 00 00 00 00 00 00 04 B0 00 00 00 F0 00 00 13 88	66 26
Modbus TCP	00 00 00 00 00 1B 01	none	10	01 64	00 0A	14	00 00 00 78 00 00 00 00 00 00 04 B0 00 00 00 F0 00 00 13 88	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	01 64	00 0A	00 2D
Modbus TCP	00 00 00 00 00 06 01	none	10	01 64	00 0A	none

likePR=133,which isLittle Endianin mode: Host sends

Modbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	01 64	00 0A	14	00 78 00 00 00 00 00 00 04 B0 00 00 00 F0 00 00 13 88 00 00	96 68
Modbus TCP	00 00 00 00 00 1B 01	none	10	01 64	00 0A	14	00 78 00 00 00 00 00 00 04 B0 00 00 00 F0 00 00 13 88 00 00	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	01 64	00 0A	00 2D
Modbus TCP	00 00 00 00 00 06 01	none	10	01 64	00 0A	none

Write Command Opcode Register (40125)data0x00DB(FH),Write parameter register (40126)data0x0001,select the1Back-to-origin method to perform back-to-origin control

Host sendsModbus RTU/TCPThe message is as follows://**pair register40125write00 DB,register40126write00 01**//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 7C	00 02	04	00 DB 00 01	45 25
Modbus TCP	00 00 00 00 00 0D 01	none	10	00 7C	00 02	04	00 DB 00 01	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	00 7C	00 02	80 10
Modbus TCP	00 00 00 00 00 06 01	none	10	00 7C	00 02	none

When selecting the -1 ~ -4Return to origin method,It is necessary to set the maximum torque of the motor during the origin return process,Set by the torque limit of the hardware limit homing method,100%corresponds to1times the rated torque of the motor.

The maximum torque (40279, 40280)data0x000002BC,according to the maximum70%the rated torque of the motor to perform homing,Its message is as follows:

likePR=5,which isBig Endianin mode:

Host sendsModbus RTU/TCPThe message is as follows://**pair register40279,40280write000002BC **//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	01 16	00 02	04	00 00 02 BC	7F C8
Modbus TCP	00 00 00 00 00 0B 01	none	10	01 16	00 02	04	00 00 02 BC	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	01 16	00 02	A1 F0
Modbus TCP	00 00 00 00 00 06 01	none	10	01 16	00 02	none

likePR=133,which isLittle Endianin mode:

Host sendsModbus RTU/TCPThe message is as follows://**pair register40279,40280write02BC0000 **//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	01 16	00 02	04	02 BC 00 00	BF 45
Modbus TCP	00 00 00 00 00 0B 01	none	10	01 16	00 02	04	02 BC 00 00	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	CRCcheck
Modbus RTU	none	01	10	01 16	00 02	A1 F0
Modbus TCP	00 00 00 00 00 06 01	none	10	01 16	00 02	none

Note:

1.When the origin return method selects the manufacturer-defined No.**-1**~**-4**When planting the back-to-origin method,After the motor returns to the mechanical origin,Continue to move an origin offset distance,After exercise,The current position of the motor is**0**.

2.When the origin return method is selected**CiA402**defined in**1**~**35**When planting the back-to-origin method,After the motor returns to the mechanical origin,stop exercising,The current position of the motor is the value of the origin offset.

3.The origin sensor and limit sensor signal input ports are in**Luna**software digital**I/O**interface to set.

7.4internalQProgram Control Routines

7.4.1 MOONS'Inside the driveQprogram routine

Line	Label	Cmd	Param1	Param2	Comment
1		WT	2		延时2秒
2		RX	1	0	1号用户寄存器赋值0，初始化设置为0
3		RX	2	11	2号用户寄存器赋值11，设定判断条件1
4		RX	3	12	3号用户寄存器赋值12，设定判断条件2
5		EP	0		编码器位置清零
6		SP	0		指令位置清零
7	Label3	CR	1	2	比较用户寄存器1和2的值
8		QJ	E	#Label1	判断两个寄存器的值是否相等，若相等，则跳转到Label1，若不相等，则向下执行
9		CR	1	3	比较用户寄存器1和3的值
10		QJ	E	#Label2	判断两个寄存器的值是否相等，若相等，则跳转到Label2，若不相等，则向下执行
11		QG	#Label3		跳转到Label3
12	Label1	RM	4	A	将4号用户寄存器的值传送到A寄存器，A寄存器的值为位置控制模式下的加速度
13		RM	5	B	将5号用户寄存器的值传送到B寄存器，B寄存器的值为位置控制模式下的减速度
14		RM	6	V	将6号用户寄存器的值传送到V寄存器，V寄存器的值为位置控制模式下的速度
15		RM	7	D	将7号用户寄存器的值传送到D寄存器，D寄存器的值为位置控制模式下的目标位置/距离
16		FP			执行绝对运动
17		QG	#Label3		跳转到Label3
18	Label2	RM	4	A	将4号用户寄存器的值传送到A寄存器，A寄存器的值为位置控制模式下的加速度
19		RM	5	B	将5号用户寄存器的值传送到B寄存器，B寄存器的值为位置控制模式下的减速度
20		RM	6	V	将6号用户寄存器的值传送到V寄存器，V寄存器的值为位置控制模式下的速度
21		RM	8	D	将8号用户寄存器的值传送到D寄存器，D寄存器的值为位置控制模式下的目标位置/距离
22		FP			执行绝对运动
23		QG	#Label3		跳转到Label3

7.4.2 QProgram variable comparison table

Function	map users register	register address	preset curve parameter	unit	write user register device value	illustrate
acceleration	4	4006,40068	100	Rps/sec	600	set value = acceleration *6
deceleration	5	40069,40070	100	Rps/sec	600	set value = deceleration *6
speed	6	40071,40072	1	Rps	240	set value = speed *240
first target location	7	40073,40074	200000	Counts	200000	1:1relation
second target location	8	40075,40076	- 200000	Counts	- 200000	1:1relation
opcode	1	40061,40062				judge,implement

7.4.3 MOONS'compatibleModbusdrive settings

Other settings are as above,But need to apply driver internalQprogramming function,needs to be setPM=9(or in the configuration interface→Control mode selectionSCL/Q,Check bothModbus→QIn the program interface, check the power-on automatic executionQprogram,then download to the drive),That is, after the drive is powered on,automatically from the1block start executionQprogram.

7.4.4 Modbusread/write operations

1)likePR=133,which isLittle Endianmodel

Pre-read user registers1 (40061,40062),2 (40063,40064)and3(40065,40066)The data in the host sends

Modbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 3C	00 06	05 C4
Modbus TCP	00 00 00 00 00 06 01	none	03	00 3C	00 06	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	number of bytes	data content	CRCcheck
Modbus RTU	none	01	03	0C	00 00 00 00 00 0B 00 00 00 0C 00 00	E9 B3
Modbus TCP	00 00 00 00 00 0F 01	none	03	0C	00 00 00 00 00 0B 00 00 00 0C 00 00	none

Pre-Write User Registers4(40067,40068)data600,user register5(40069,40070)data600(0x00000258), user register6(40071,40072)data240 (0x000000F0),user register7(40073,40074)data 200000 (0x00030D40),user register8(40075,40076)data-200000 (0xFFFCF2C0)

Host sendsModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 42	00 0A	14	02 58 00 00 02 58 00 00 00 F0 00 00 0D 40 00 03 F2 C0 FF FC	DC FC
Modbus TCP	00 00 00 00 00 1B 01	none	10	00 42	00 0A	14	02 58 00 00 02 58 00 00 00 F0 00 00 0D 40 00 03 F2 C0 FF FC	none

Pre-Write User Registers1 (40061,40062)data11 (0x000B),Inside the driveQprocedural judgment,If register1data inside = register2internal data,Then run to the target position as200000

Host sendsModbus RTU/TCPThe message is as follows://**write40061,40062**//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 3C	00 02	04	00 0B 00 00	81 2C
Modbus TCP	00 00 00 00 00 0B 01	none	10	00 3C	00 02	04	00 0B 00 00	none

or://**write40061**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 3C	00 0B	08 01
Modbus TCP	00 00 00 00 00 06 01	none	06	00 3C	00 0B	none

Pre-Write User Registers1 (40061,40062)data12 (0x0000C),Inside the driveQprocedural judgment,If register1data inside = register3internal data,Then run to the target position as -200000

Host sendsModbus RTU/TCPThe message is as follows://**write40061,40062**//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 3C	00 02	04	00 0C 00 00	30ED
Modbus TCP	00 00 00 00 00 0B 01	none	10	00 3C	00 02	04	00 0C 00 00	none

or://**write40061**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 3C	00 0C	49 C3
Modbus TCP	00 00 00 00 00 06 01	none	06	00 3C	00 0C	none

Pre-read user registers4(40067,40068)The data,Qprogram can set the user register4Write the contents of the acceleration register to theAhost sendModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 42	00 02	64 1F
Modbus TCP	00 00 00 00 00 06 01	none	03	00 42	00 02	none

Pre-read user registers4(40069,40070)The data,Qprogram can set the user register5The contents of the deceleration register are written toBhost sendModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 44	00 02	84 1E
Modbus TCP	00 00 00 00 00 06 01	none	03	00 44	00 02	none

Pre-read user registers6(40071,40072)The data,Qprogram can set the user register6The contents of the write to the speed registerVhost send Modbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 46	00 02	25 DE
Modbus TCP	00 00 00 00 00 06 01	none	03	00 46	00 02	none

Pre-read user registers7(40073,40074)The data,Qprogram can set the user register7The contents of the write to the location registerDhost send Modbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 48	00 02	44 1D
Modbus TCP	00 00 00 00 00 06 01	none	03	00 48	00 02	none

Pre-read user registers8(40075,40076)The data,Qprogram can set the user register8The contents of the write to the location registerDhost send Modbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 4A	00 02	E5 DD
Modbus TCP	00 00 00 00 00 06 01	none	03	00 4A	00 02	none

Read the status register (40002),Can display the operating status of the drive,For example, if the feedback information is0x4001,expressQProgram is running and the drive is enabled,For details, see "Host Command Referencemanual",Its message is as follows:

Host sendsModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 01	00 01	D5 CA
Modbus TCP	00 00 00 00 00 06 01	none	03	00 01	00 01	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	number of bytes	data content	CRCcheck
Modbus RTU	none	01	03	02	40 01	48 44
Modbus TCP	00 00 00 00 00 05 01	none	03	02	40 01	none

readQThe line number of the program currently executing (40018)

Host sendsModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 11	00 01	D4 0F
Modbus TCP	00 00 00 00 00 06 01	none	03	00 11	00 01	none

Read the acceleration of the planned curve (40028),deceleration(40029),speed(40030),target location(40031,40032) Host

sendsModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 1B	00 05	F5 CE
Modbus TCP	00 00 00 00 00 06 01	none	03	00 1B	00 05	none

2)likePR=5,which isBig Endianmodel

Pre-read user registers1 (40061,40062),2 (40063,40064)and3(40065,40066)The data in the host sends

Modbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	register address	Number of registers	CRCcheck
Modbus RTU	none	01	03	00 3C	00 06	05 C4
Modbus TCP	00 00 00 00 00 06 01	none	03	00 3C	00 06	none

Return from the station deviceModbus RTU/TCPThe message is as follows:

Modbusagreement type	MBAPheader	address code	function code	number of bytes	data content	CRCcheck
Modbus RTU	none	01	03	0C	00 00 00 00 00 00 00 0B 00 00 00 0C	36 B4
Modbus TCP	00 00 00 00 00 0F 01	none	03	0C	00 00 00 00 00 00 00 0B 00 00 00 0C	none

Pre-Write User Registers4(40067,40068)data600,user register5(40069,40070)data600 (0x00000258) , user register6(40071,40072)data240 (0x000000F0),user register7(40073,40074)data 200000 (0x00030D40),user register8(40075,40076)data-200000 (0xFFFCF2C0)

Host sendsModbus RTU/TCPThe message is as follows:

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 42	00 0A	14	00 00 02 58 00 00 02 58 00 00 00 F0 00 03 0D 40 FF FC F2 C0	BF 30
Modbus TCP	00 00 00 00 00 1B 01	none	10	00 42	00 0A	14	00 00 02 58 00 00 02 58 00 00 00 F0 00 03 0D 40 FF FC F2 C0	none

Pre-Write User Registers1 (40061,40062)data11 (0x000B),Inside the driveQprocedural judgment,If register1data inside = register2internal data,Then run to the target position as200000

Host sendsModbus RTU/TCPThe message is as follows://**write40061,40062**//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 3C	00 02	04	00 00 00 0B	B1 29
Modbus TCP	00 00 00 00 00 0B 01	none	10	00 3C	00 02	04	00 00 00 0B	none

or://**write40061**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 3C	00 0B	08 01
Modbus TCP	00 00 00 00 00 06 01	none	06	00 3C	00 0B	none

Pre-Write User Registers1 (40061,40062)data12 (0x0000C),Inside the driveQprocedural judgment,If register1data inside = register3internal data,Then run to the target position as -200000

Host sendsModbus RTU/TCPThe message is as follows://**write40061,40062**//

Modbusprotocol type	MBAPheader	address code	function code	register address	register quantity	number of bytes	data content	CRCcheck
Modbus RTU	none	01	10	00 3C	00 02	04	00 00 00 0C	F0 EB
Modbus TCP	00 00 00 00 00 0B 01	none	10	00 3C	00 02	04	00 00 00 0C	none

or://**write40061**//

Modbusagreement type	MBAPheader	address code	function code	register address	data content	CRCcheck
Modbus RTU	none	01	06	00 3C	00 0C	49 C3
Modbus TCP	00 00 00 00 00 06 01	none	06	00 3C	00 0C	none

appendix1Function code message format

function code0x03read holding register:

query message:

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	03
Starting Address Hi	00
Starting Address Lo	6B
No. of Points Hi	00
No. of Roints Lo	03
Error Check (LRC or CRC)	—

response message:

RESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	03
Byte Count	06
Data Hi(Register 40108)	02
Data Lo(Register 40108)	2B
Data Hi(Register 40109)	00
Data Lo(Register 40109)	00
Data Hi(Register 40110)	00
Data Lo(Register 40110)	64
Error Check (LRC or CRC)	—

function code0x06write a single register:

query message:

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

response message:

RESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

function code0x10write multiple registers

query message:

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data Hi	00
Data Lo	0A
Data Hi	01
Data Lo	02
Error Check (LRC or CRC)	—

response message:

PESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi No. of	00
Registers Lo Error Check	02
(LRC or CRC)	—

appendix2 ModbusException response and code

query message:

QUERY

Byte	Contents	Example
1	Slave Address	0A
2	Function	01
3	Starting Address Hi	04
4	Starting Address Lo	A1
5	No. of Coils Hi	00
6	No. of Coils Lo	01
7	LRC	4F

Abnormal response message:

EXCEPTION RESPONSE

Byte	Contents	Example
1	Slave Address	0A
2	Function	81
3	Exception Code	02
4	LRC	73

In the above example,Slave device address10(0AH),Function code for reading coil status01,The address where the host accesses the coil1245(04A1H),The number of read coils is1indivual(0001H).

If this coil address does not exist in the slave device,i.e. with exception code02Return an exception response to the host,Indicates that this register address is illegal.

MOONS'Drive Abnormal Response Code Table:

code (Hex)	name	meaning
01	Unsupported function code	The slave device does not support this function code
02	invalid register	The number of one-time access registers exceeds the range (the maximum number of accesses in the step system is125,The maximum number of accesses to the servo system is50) , register address greater than40200or less than40001
03	illegal data area	The number of registers accessed is0
11	Register does not support reading	The accessed register address does not support reading
12	Register does not support writing	The accessed register address does not support writing
13	Set value out of range	The written value exceeds the register setting range

appendix3 CRCcheck

Cyclic Redundancy CheckCRCDistrict is2byte,contains one16bit binary data.Calculated by the sending deviceCRCvalue,and attach the calculated value to the message,When the receiving device receives the information,recalculateCRCvalue,and compare the calculated value with the receivedCRCCompare the actual value in the area,If the two are not the same,produces an error.

CRCAt the beginning, put the register16All bits are set to "1",then put adjacent2indivual8Bit-byte data into the current register,only each character8bit data is used to generateCRC,start bit,Stop bits and parity bits are not added toCRCmiddle.

produceCRCperiod,Every8XOR the bit data with the value in the register,The result is shifted one bit to the right (towardLSB direction),and use "0”fill inMSB,detectLSB,likeLSBfor"1"XOR with the preset fixed value,likeLSBfor"0”no XOR operation.

Repeat the above process,up to displacement8Second-rate,complete the8after shift,Next8bit data,XOR with the current value of this register,After all information has been processed,The final value in the register isCRCvalue.

produceCRCthe process of:

1.Bundle16bitCRCregister set toFFFFH.

2.First8bit data withCRCregister low8XOR the bits,put the result inCRCregister.

3. CRCshift register to the right1bit,MSBzero fill,an examinationLSB.

4. (likeLSBfor0):repeat3,move right1bit.

(likeLSBfor1):CRCregister withA001HXOR operation

5.repeat3and4until complete8secondary shift,Finish8bit byte handling.

6.repeat2to5step,process the next8bit data,until all bytes have been processed.

7. CRCThe final value of the register isCRCvalue.

8.BundleCRCWhen the value is put into the message,high8bit and low8bits should be placed separately.BundleCRCvalue into message,in sending information16 bitCRC value,send low first8bit,escort high8bit.

likeCRCvalue is1241(0001 0010 0100 0001):

Addr	Func	Data Count	Data	Data	Data	Data	CRC Lo	CRC Hi
							41	12

example:

all possibleCRCvalue,Load in two columns,a column in16bitCRCheight of8bit area,for(0-256of)CRCvalue

another class is low8bit area,forCRClow value of.

obtained in this wayCRCIt performs faster than calculating a new character for each new character in the bufferCRCvalue method.

Notice:The function internally swaps**CRC**medium high**/**low byte,return**CRC**value,its bytes are swapped. therefore,returned by this

functionCRCvalue,can be sent directly in the message. routine:

function take2independent variable:

unsigned char *puchMsg ;to generateCRCvalue,Point the pointer to the buffer containing the binary data

unsigned short usDataLen ;the number of bytes in the buffer. The function returnsCRCas a type"unsigned

short". CRCgenerated function

```
unsigned short CRC16(puchMsg, usDataLen) unsigned char *puchMsg ; /*
Calculated by the number of bytes of informationCRC*/ unsigned short
usDataLen ; /* quantity of bytes in message */ {

unsigned char uchCRCHi = 0xFF ; /*Initialize high byte */
unsigned char uchCRCLo = 0xFF ; /*Initialize low byte */
unsigned uIndex ; /*BundleCRCsurface*/ while
(usDataLen––) /*via data buffer */ {

uIndex = uchCRCHi ^ *puchMsgg++ ; /*calculate
CRC*/ uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex] ;
uchCRCLo = auchCRCLo[uIndex] ;
}
return (uchCRCHi << 8 | uchCRCLo) ;
```

appendix4 Modbus/RTU16bitCRCverification routine

```
usingSystem;
usingSystem.Collections.Generic;
usingSystem.Text;

namespaceModbus
{
    public static classUtility {

        private static readonly ushort[] m_CrcTable = {

            0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241, 0XC601, 0X06C0, 0X0780, 0XC741, 0X0500,
            0XC5C1, 0XC481, 0X0440, 0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40, 0X0A00, 0XCAC1,
            0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841, 0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81,
            0X1A40, 0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41, 0X1400, 0XD4C1, 0XD581, 0X1540,
            0XD701, 0X17C0, 0X1680, 0XD641, 0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040, 0XF001,
            0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240, 0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0,
            0X3480, 0XF441, 0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41, 0XFA01, 0X3AC0, 0X3B80,
            0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840, 0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
            0XEE01, 0X2EC0, 0X2F80,0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40, 0XE401, 0X24C0, 0X2580, 0XE541, 0X2700,
            0XE7C1, 0XE681, 0X2640, 0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041, 0XA001, 0X60C0,
            0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240, 0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480,
            0XA441, 0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41, 0XAA01, 0X6AC0, 0X6B80, 0XAB41,
            0X6900, 0XA9C1, 0XA881, 0X6840, 0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41, 0XBE01,
            0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40, 0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1,
            0XB681, 0X7640,0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441, 0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01,
            0X6FC0, 0X6E80, 0XAE41, 0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840, 0X7800, 0XB8C1,
            0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41, 0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81,
            0X7C40, 0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,0XA781, 0X6740, 0XA501, 0X65C0,
            0X6480, 0XA441, 0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41, 0XAA01, 0X6AC0, 0X6B80,
            0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840, 0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
            0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40, 0XB401, 0X74C0, 0X7580, 0XB541, 0X7700,
            0XB7C1, 0XB681, 0X7640,
```

```
    0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
    0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
    0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
    0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
    0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
    0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
    0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
    0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
    0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040
};
/// <summary>
///Calculate Longitudinal Redundancy Check. ///
</summary>
/// <param name="data">The data used in LRC</param> ///
<returns>LRC value</returns> public static byteCalculateLrc(
byte[] data) {

    if(data ==null) {

      throw newArgumentNullException("data");
    }
    byte lrc = 0;
    foreach(byte bindata) {

      lrc += b;
    }
    lrc = (byte)((lrc^0xFF) + 1);
    returnlrc;
}

/// <summary>
///Calculate Cyclical Redundancy Check ///
</summary>
/// <param name="data">The data used in CRC</param> ///
<returns>CRC value</returns>
```

```
    public static byte[] CalculateCrc(byte[] data) {


      if(data ==null) {


        throw newArgumentNullException("data");
      }
      ushortcrc =ushort.MaxValue;
      foreach(bytebindata) {


        bytetableIndex = (byte)(crc^b);
        crc >>= 8;
        crc ^= m_CrcTable[tableIndex];
      }
      returnBitConverter.GetBytes(crc);
    }
  }
}
```

The following is the calling method:

```
byte[]_Data =new byte[] { 0x31, 0x32};
byte[]_Crc = Modbus.Utility.CalculateCrc(_Data); //
result: _Crc = { 0x95, 0xf5 }
```

M2series				
Register	**Access**	**Data Type**	**Description**	**SCL Register**
40154	R/W	SHORT	Step Mode (SZ) pulse mode	
40155	R/W	SHORT	Position Fault (PF) Position Error Alarm Threshold	
40156	R/W	SHORT	Dynamic Position Error Count (PL) Dynamic following error threshold	
40157	R/W	SHORT	In-Position Counts (PD) Static position error range	
40158	R/W	SHORT	In-Position Timing (PE) Static position error duration	
40159	R/W	SHORT	Pulse Complete Timing (TT) Pulse input completion detection time	
40160	R/W	SHORT	Analog Velocity Gain (AG) Analog speed scaling	
40161	R/W	SHORT	Analog Torque Gain (AN) Analog torque scaling	
40162	R/W	SHORT	Analog Offset 1 (AV1) Analog input1Offset	
40163	R/W	SHORT	Analog Offset 2 (AV2) Analog input2Offset	
40164	R/W	SHORT	Analog Type (AS) Analog input type	
40165	R/W	SHORT	Analog Deadband 1 (AD1) Analog input1dead zone	
40166	R/W	SHORT	Analog Deadband 2 (AD2) Analog input2dead zone	
40167	R/W	SHORT	Analog Deadband (AD) Differential analog input deadband	
40168	R/W	SHORT	Analog Function (FA) Analog function	
40169	R/W	SHORT	Servo Enable (SI) Enable input pin function	
40170	R/W	SHORT	Alarm Reset (AI) Alarm clear input pin function	
40171	R/W	SHORT	Define Limits Input (DL) Define limit sensor input function	
40172	R/W	SHORT	Motion Input X7,X8,X9,X10Input pin function definition	
40173	R/W	SHORT	Alarm Output (AO) Alarm output pin function definition	
40174	R/W	SHORT	Brake Output (BO) Motor brake output pin function definition	
40175	R/W	SHORT	Motion Output (MO) Y3,Y4,Y5,Y6Output pin function setting	
40176	Read Only	SHORT	Reserved	

M3/MBDVseries					
Register	**Access**	**Data Type**	**Units**	**Description**	**SCL Register**
40001..002	Read Only	LONG	— —	Alarm Code (AL) Alarm code (main code)	f
40003..004	Read Only	LONG	— —	Status Code (SC) status code	s
40005	Read Only	SHORT	— —	Digital Output Status (IO) Digital output port status	y
40006	Read Only	SHORT	— —	Digital Input Status (IS) Digital input port status	i
40007..008	Read Only	LONG	pulses	Immediate Absolute Position (IP) Reference position	l
40009..010	Read Only	LONG	pulses	Secondary Encoder Position (EQ) Second encoder position	
40011..012	Read Only	LONG	pulses	Encoder Position (EP) Motor encoder position	e
40013..014	Read Only	LONG	pulses	internal use	
40015	R/W	SHORT		Reserved	
40016	Read Only	SHORT	rev	Encoder Multi-turn Data Absolute encoder multi-turn data	
40017	Read Only	SHORT	1/240rps	Immediate Actual Velocity (IV) Instantaneous actual speed	v
40018	Read Only	SHORT	1/240rps	Immediate Target Velocity (IV1) Instantaneous command speed	w
40019	Read Only	SHORT	0.1°C	Immediate Drive Temperature (IT) Instantaneous drive temperature	t
40020	Read Only	SHORT	0.1°C	Immediate DSP Temperature (IT1) InstantaneousDSPtemperature	
40021	Read Only	SHORT	0.1°C	Immediate Encoder Temperature (IT2) (Only for M3)Instantaneous motor temperature	
40022	Read Only	SHORT	0.1V	Immediate DC_Bus Voltage (IU) InstantaneousDCbus voltage	u
40023..024	Read Only	LONG	pulses	Immediate Position Error (IX) Instantaneous position error	x
40025	R/W	SHORT		Reserved	
40026	Read Only	SHORT	mv	Analog Input 1 (IA1) Analog input1Voltage	j
40027	Read Only	SHORT	mv	Analog Input 2 (IA2) Analog input2Voltage	k
40028	R/W	SHORT	mv	Analog Output 1 (OA1) Analog output1Voltage	T
40029	R/W	SHORT	mv	Analog Output 2 (OA2) Analog output2Voltage	W (capital) Capital
40030	Read Only	SHORT	— —	Q Program Line Number QThe line number of the program currently executing	b

M3/MBDVseries					
Register	**Access**	**Data Type**	**Units**	**Description**	**SCL Register**
40031	Read Only	SHORT	0.1%	Immediate Current Command (IC) Instantaneous actual current	c
40032	Read Only	SHORT	0.1%	Q Current (IQ) InstantaneousQshaft current	q
40033..034	Read Only	LONG	pulses	Relative Distance (ID) relative position	d
40035..036	Read Only	LONG	pulses	Sensor Position Sensor location	g
40037	Read Only	SHORT	— —	Condition Code Compare status codes	h
40038	Read Only	SHORT	— —	Control Mode control mode	m
40039	Read Only	SHORT	— —	Velocity Move State Current state of motion in speed mode	n
40040	Read Only	SHORT	— —	Point-to-Point Move State Current state of motion in peer-to-peer mode	o
40041	Read Only	SHORT	— —	Q Segment Number QThe current segment number of the program	p
40042	Read Only	SHORT	— —	Model Number Drive model code	
40043	Read Only	SHORT	— —	Sub Model Drive model subcode	
40044	Read Only	SHORT	— —	DSP Firmware Version DSPFirmware version number	
40045	Read Only	SHORT	— —	FPGA Firmware Version NO FPGAFirmware version number1	
40046	Read Only	SHORT	— —	FPGA Firmware Version LA FPGAFirmware version number2	
40047..048	R/W	LONG	pulses	Input Counter input count	I (capital) Capital
40049..050	R/W	LONG	pulses	Pulse Counter (Only for M3) Pulse input count	S (capital) Capital
40051	R/W	SHORT	— —	internal use	
40052..053	Read Only	LONG	s	Power Up Seconds Drive runtime	
40054..055	Read Only	LONG	times	Power On Times Drive boot count	
40056	Read Only	SHORT	— —	Encoder Firmware Version Encoder firmware version number	
40057	R/W	SHORT	— —	internal use	
40058	Read Only	SHORT	— —	internal use	
40059	Read	SHORT	— —	internal use	
40060	R/W	SHORT	1%	internal use	

M3/MBDVseries					
Register	**Access**	**Data Type**	**Units**	**Description**	**SCL Register**
40061	R/W	SHORT	pulses	internal use	
40062	R/W	SHORT	ms	internal use	
40063	R/W	SHORT	— —	internal use	
40064	R/W	SHORT	— —	internal use	
40065	R/W	SHORT	0~3000	Torque Limit Dynamic CW CWDirectional dynamic torque limit	Y
40066	R/W	SHORT	0~3000	Torque Limit Dynamic CCW CCWDirectional dynamic torque limit	Z(capital) (Capital)
40067..68	Read Only	LONG	— —	Alarm Code Alarm code (auxiliary code)	r
40069	Read Only	SHORT	— —	Alarm Buffer 0 Drive error code record8	
40070	Read Only	SHORT	— —	Alarm Buffer 1 Drive error code record1	
40071	Read Only	SHORT	— —	Alarm Buffer 2 Drive error code record2	
40072	Read Only	SHORT	— —	Alarm Buffer 3 Drive error code record3	
40073	Read Only	SHORT	— —	Alarm Buffer 4 Drive error code record4	
40074	Read Only	SHORT	— —	Alarm Buffer 5 Drive error code record5	
40075	Read Only	SHORT	— —	Alarm Buffer 6 Drive error code record6	
40076	Read Only	SHORT	— —	Alarm Buffer 7 Drive error code record7	
40077..78	Read Only	LONG	s	Alarm Buffer 8 Drive error code record8Generation time	
40079..80	Read Only	LONG	s	Alarm Buffer 9 Drive error code record1Generation time	
40081..82	Read Only	LONG	s	Alarm Buffer 10 Drive error code record2Generation time	
40083..84	Read Only	LONG	s	Alarm Buffer 11 Drive error code record3Generation time	
40085..86	Read Only	LONG	s	Alarm Buffer 12 Drive error code record4Generation time	
40087..88	Read Only	LONG	s	Alarm Buffer 13 Drive error code record5Generation time	
40089..90	Read Only	LONG	s	Alarm Buffer 14 Drive error code record6Generation time	
40091..92	Read Only	LONG	s	Alarm Buffer 15 Drive error code record7Generation time	
40093..94	Read Only	LONG	— —	Alarm Buffer 16	

M3/MBDVseries					
Register	**Access**	**Data Type**	**Units**	**Description**	**SCL Register**
40095..96	Read Only	LONG	— —	Alarm Buffer 17	
40097..98	Read Only	LONG	— —	Alarm Buffer 18	
40099..100	Read	LONG	— —	Alarm Buffer 19	
40101..102	Read Only	LONG	— —	Alarm Buffer 20	
40103..104	Read Only	LONG	— —	Alarm Buffer 21	
40105..106	Read Only	LONG	— —	Alarm Buffer 22	
40107..108	Read Only	LONG	— —	Alarm Buffer 23	
40109..110	Read Only	LONG	— —	Alarm Buffer 24 Drive error code record8generate time error value	
40111..112	Read Only	LONG	— —	Alarm Buffer 25 Drive error code record1generate time error value	
40113..114	Read Only	LONG	— —	Alarm Buffer 26 Drive error code record2generate time error value	
40115..116	Read Only	LONG	— —	Alarm Buffer 27 Drive error code record3generate time error value	
40117..118	Read Only	LONG	— —	Alarm Buffer 28 Drive error code record4generate time error value	
40119..120	Read Only	LONG	— —	Alarm Buffer 29 Drive error code record5generate time error value	
40121..122	Read Only	LONG	— —	Alarm Buffer 30 Drive error code record6generate time error value	
40123..124	Read Only	LONG	— —	Alarm Buffer 31 Drive error code record7generate time error value	
40125	R/W	SHORT	— —	Command Opcode	
40126	R/W	SHORT	— —	Parameter 1	
40127	R/W	SHORT	— —	Parameter 2	
40128	R/W	SHORT	— —	Parameter 3	
40129	R/W	SHORT	— —	Parameter 4	
40130	R/W	SHORT	— —	Parameter 5	
40131..132	Read Only	LONG	— —	Accumulator accumulator	0
40133..134	R/W	LONG	— —	User Register 1 User-defined register1	1
40135..136	R/W	LONG	— —	User Register 2 User-defined register2	2
40137..138	R/W	LONG	— —	User Register 3 User-defined register3	3

M3/MBDVseries					
Register	**Access**	**Data Type**	**Units**	**Description**	**SCL Register**
40139..140	R/W	LONG	— —	User Register 4 User-defined register4	4
40141..142	R/W	LONG	— —	User Register 5 User-defined register5	5
40143..144	R/W	LONG	— —	User Register 6 User-defined register6	6
40145..146	R/W	LONG	— —	User Register 7 User-defined register7	7
40147..148	R/W	LONG	— —	User Register 8 User-defined register8	8
40149..150	R/W	LONG	— —	User Register 9 User-defined register9	9
40151..152	R/W	LONG	— —	User Register 10 User-defined register10	:
40153..154	R/W	LONG	— —	User Register 11 User-defined register11	;
40155..156	R/W	LONG	— —	User Register 12 User-defined register12	<
40157..158	R/W	LONG	— —	User Register 13 User-defined register13	=
40159..160	R/W	LONG	— —	User Register 14 User-defined register14	>
40161..162	R/W	LONG	— —	User Register 15 User-defined register15	?
40163..164	R/W	LONG	— —	User Register 16 User-defined register16	@
40165..166	R/W	LONG	— —	User Register 17 User-defined register17	[
40167..168	R/W	LONG	— —	User Register 18 User-defined register18	\
40169..170	R/W	LONG	— —	User Register 19 User-defined register19	]
40171..172	R/W	LONG	— —	User Register 20 User-defined register20	^
40173..174	R/W	LONG	— —	User Register 21 User-defined registertwenty one	_
40175..176	R/W	LONG	— —	User Register 22 User-defined registertwenty two	`

M3/MBDVseries-**P0**Group**(PID)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40177..178	R/W	LONG	— —	0~2	Tuning Mode Selection (UM) Parameter tuning mode	
40179..180	R/W	LONG	— —	0~10	Load Type (LY) load type	
40181..182	R/W	LONG	— —	0~100	Inertia Ratio (NR) Load inertia ratio	
40183..184	R/W	LONG	— —	1~20	1st Mechanical Stiffness Level (KG) first rigidity class	
40185..186	R/W	LONG	— —	1~20	2nd Mechanical Stiffness Level (KX) Second rigidity level	
40187..188	R/W	LONG	0.1Hz	0~20000	1st Position Loop Gain (KP) first position loop gain	
40189..190	R/W	LONG	ms	0~30000	1st Position Loop Integral Time Constant (KI) 1st position loop integral time constant	
40191..192	R/W	LONG	ms	0~30000	1st Position Loop Derivative Time Constant (KD) 1st position loop differential time constant	
40193..194	R/W	LONG	0.1Hz	0~40000	1st Position Loop Derivative Filter (KE) First position loop differential filter frequency	
40195..196	R/W	LONG	0.01%	- 30000~30000	Velocity Feedforward Gain (KL) Speed feed forward gain	
40197..198	R/W	LONG	0.1Hz	0~40000	Velocity Feedforward Filter (KR) Velocity feedforward filter frequency	
40199..200	R/W	LONG	0.01%	- 30000~30000	1st Velocity Command Gain (KF) First command speed gain	
40201..202	R/W	LONG	0.1Hz	0~30000	1st Velocity Loop Gain (VP) first speed loop gain	
40203..204	R/W	LONG	ms	0~30000	1st Velocity Loop Integral Time Constant (VI) 1st speed loop integral time constant	
40205..206	R/W	LONG	0.01%	0~20000	Acceleration Feedforward Gain (KK) Acceleration feedforward gain	
40207..208	R/W	LONG	0.1Hz	0~40000	Acceleration Feedforward Filter (KT) Acceleration feedforward filter frequency	
40209..210	R/W	LONG	0.1Hz	0~40000	1st Torque Command Filter (KC) First command torque filter frequency	
40211..212	R/W	LONG	0.1Hz	0~20000	2nd Position Loop Gain (UP) Second position loop gain	
40213..214	R/W	LONG	ms	0~30000	2nd Position Loop Integral Time Constant (UI) Second position loop integral time constant	
40215..216	R/W	LONG	ms	0~30000	2nd Position Loop Derivative Time Constant (UD) Second position loop differential time constant	

M3/MBDVseries-**P0**Group**(PID)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40217..218	R/W	LONG	0.1Hz	0~40000	2nd Position Loop Derivative Filter (UE) The second position loop differential filter frequency	
40219..220	R/W	LONG	0.01%	- 30000~30000	2nd Velocity Command Gain (UF) Second command speed gain	
40221..222	R/W	LONG	0.1Hz	0~30000	2nd Velocity Loop Gain (UV) Second speed loop gain	
40223..224	R/W	LONG	ms	0~30000	2nd Velocity Loop Integral Time Constant (UG) Second speed loop integral time constant	
40225..226	R/W	LONG	0.1Hz	0~40000	2nd Torque Command Filter (UC) Second command torque filter frequency	
40227..228	R/W	LONG	0.1Hz	0~20000	Full Closed-loop Position Loop Gain (XP) Fully closed loop - position loop gain	
40229..230	R/W	LONG	ms	0~30000	Full Closed-loop Position Loop Integral Time Constant (XI) Fully closed loop-position loop integral time constant	
40231..232	R/W	LONG	ms	0~32767	Full Closed-loop Position Loop Derivative Time Constant (XD) Fully closed loop - differential time constant of position loop	
40233..234	R/W	LONG	0.1Hz	0~40000	Full Closed-loop Position Loop Derivative Filter (XE) Full closed loop-position loop differential filter frequency	
40235..236	R/W	LONG	0.01%	- 30000~30000	Full Closed-loop Velocity Command Gain (XF) Full closed loop - command speed gain	
40237..238	R/W	LONG	0.1Hz	0~30000	Full Closed-loop Velocity Loop Gain (XV) Fully closed loop - speed loop gain	
40239..240	R/W	LONG	ms	0~30000	Full Closed-loop Velocity Loop Integral Time Constant (XG) Full closed loop - speed loop integral time constant	
40241..242	R/W	LONG	0.1Hz	0~40000	Full Closed-loop Torque Command Filter (XC) Full closed loop - command torque filter frequency	
40243..244	R/W	LONG	— —	0~4	Automatic Gain Switching Method (SD) Gain switching condition selection	O (capital) Capital
40245..246	R/W	LONG	pulses	0~2147483647	Use Position Error as the Condition (PN) Gain Switching Condition - Position	
40247..248	R/W	LONG	1/240rps	0~24000	Use Actual Speed as the Condition (VN) Gain Switching Condition - Speed	

M3/MBDVseries-**P0**Group**(PID)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40249..250	R/W	LONG	0.1%	0~3000	Use Actual Torque as the Condition (TN) Gain Switching Condition - Torque	
40251..252	R/W	LONG	ms	0~10000	Gain Switching Waiting Time 1 (SE1) Second gain switch to first gain delay time	
40253..254	R/W	LONG	ms	0~10000	Gain Switching Waiting Time 2 (SE2) 1st gain to 2nd gain delay time	
40255..256	R/W	LONG	— —	0~3	Velocity Feedback Filter (LR) Velocity Feedback Filter	
40257..258	R/W	LONG	— —	0~1	Self-adapting Filter Switch (AE) Adaptive Filter Switch	
40259..260	R/W	LONG			Reserved	

M3/MBDVseries-**P1**Group**(Configuration)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40261..262	R/W	LONG	— —		Reserved	
40263..264	R/W	LONG	— —	1,2,7,11,15,21	Main Control Mode (CM) master control mode	
40265..266	R/W	LONG	— —	1,2,7,11,15,21	Secondary Control Mode (CN) second control mode	
40267..268	R/W	LONG	— —	8~10	Operation Mode When Power-up (PM) Power-on working mode	
40269..270	R/W	LONG	— —	1~2	Speed Control Clamp Mode (JM) Speed Control Clamp Mode	
40271..272	R/W	LONG	— —	0~1	Full Closed-loop Control Switch (XM) Fully closed loop mode switch	
40273..274	R/W	LONG	0.1%	- 3000~3000	Torque Command of Internal Torque Mode (GC) Command torque in internal torque mode	G
40275..276	R/W	LONG	0.1%	0~3000	1st Torque Limit (CC) first torque limit	
40277..278	R/W	LONG	0.1%	0~3000	Target Value of Torque Arrival (CV) Judging that the torque reaches the target value	
40279..280	R/W	LONG	0.1%	0~3000	Torque Limit of Hardstop Homing (HC) Torque limit of hardware limit homing mode	
40281..282	R/W	LONG	ms	0~30000	Current Foldback Continuous Time (CL) Torque overload duration	
40283..284	R/W	LONG	— —	0~5	Torque Limit Method (LD) Torque limit method	
40285..286	R/W	LONG	— —	0~1	Rotational Direction Setup (RN) Motor rotation direction selection	
40287..288	R/W	LONG			Reserved	
40289..290	R/W	LONG	— —	1~511	Communication Protocol (PR) Protocol	
40291..292	R/W	LONG	ms	0~20	Transmit Delay (TD) response delay	
40293..294	R/W	LONG	— —	1~5	RS-485 Baud Rate (BR) RS-485Communication baud rate	
40295..296	R/W	LONG	— —	0~32	RS-485 Address (DA) RS-485mailing address	
40297..298	R/W	LONG	— —	1~127	Node ID (CO) CANopen/IPCommunication node address	
40299..300	R/W	LONG	— —	0~7	CANopen Baud Rate CANopenCommunication baud rate	
40301..302	R/W	LONG	Ω	10~32000	Regeneration Resistor Value (ZR) Regenerative absorption resistor resistance	

M3/MBDVseries-**P1**Group**(Configuration)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40303..304	R/W	LONG	W	1~32000	Regeneration Resistor Wattage (ZW) Regenerating Absorption Resistor Power	
40305..306	R/W	LONG	ms	0~8000	Regeneration Resistor Time Constant (ZT) Regeneration absorption time constant	
40307..308	R/W	LONG	— —	0~1	Keypad Setting Lock (PK) Key setting lock	
40309..310	R/W	LONG	— —	0~20	Default Display (DD) leddefault display item	
40311..312	R/W	LONG	— —	0~4294967295	Alarm Mask (MA) Alarm masking	
40313..314	R/W	LONG	0.1%	0~3000	2nd Torque Limit (CX) Second torque limit	
40315..316	R/W	LONG	0.1%	0~3000	3rd Torque Limit (CY) Third torque limit	
40317..318	R/W	LONG	0.1%	0~3000	4th Torque Limit (CZ) Fourth torque limit	
40319..320	R/W	LONG	ms	0~30000	Motor Stall Protection Time (HT) Motor stall protection time	
40321..322	R/W	LONG	— —	0~5	Dynamic Brake Sequence when Servo Off (YV) The action of the dynamic brake when it is disabled	
40323..324	R/W	LONG	— —	0~3	Dynamic Brake Sequence when Fault Occurs (YR) The action of the dynamic brake when an error is reported	
40325..326	R/W	LONG	ms	0~30000	Dynamic Brake Action Time during Deceleration of Servo Off (YM) Dynamic braking during deceleration longest action time	
40327..328	R/W	LONG	ms	0~30000	Dynamic Brake Action Time during Deceleration when Fault Occurs (YN) Dynamic braking during the deceleration of the error is the most long action time	
40329..330	R/W	LONG	— —	0~1	Main Power Phase Lost Detecting (OT) Drive main circuit power input phase loss detection is ON close	
40331..332	R/W	LONG	0.1%	0~3000	Current Ramp Limit (RT) Driver Output Current Transient Limits	
40333..334	R/W	LONG	0.1V	200~800	Dumping Cicuit Working Voltage (DW) (Only for MBDV) Operating voltage point of bleeder circuit	

M3/MBDVseries-**P2**Group**(Trajectory)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40335..336	R/W	LONG	1/6(rps/s)	1~30000	Max Brake Deceleration (AM) Servo brake deceleration	
40337..338	R/W	LONG	1/240rps	0~24000	Max Velocity (VM) Maximum speed	M
40339..340	R/W	LONG	1/6(rps/s)	1~30000	Jog Accel (JA) Internal Velocity Mode Acceleration	K (capital) Capital
40341..342	R/W	LONG	1/6(rps/s)	1~30000	Jog Decel (JL) Internal speed mode deceleration	L
40343..344	R/W	LONG	1/240rps	- 24000~24000	Jog Velocity (JS) Internal speed mode target speed	J
40345..346	R/W	LONG	1/6(rps/s)	1~30000	Point-to-Point Accel (AC) Acceleration in internal peer-to-peer mode	A
40347..348	R/W	LONG	1/6(rps/s)	1~30000	Point-to-Point Decel (DE) Deceleration in internal peer-to-peer mode	B
40349..350	R/W	LONG	1/240rps	0~24000	Point-to-Point Velocity (VE) Speed in internal peer-to-peer mode	V (capital) Capital
40351..352	R/W	LONG	pulses	- 2147483647~ 2147483647	Point-to-Point Distance (DI) Distance (position) in internal peer-to-peer mode	D
40353..354	R/W	LONG	pulses	- 2147483647~ 2147483647	Point-to-Point Change Distance (DC) Pitch in internal point-to-point mode	C (capital) Capital
40355..356	R/W	LONG	1/240rps	0~24000	Point-to-Point Change Velocity (VC) Speed down in internal peer-to-peer mode	U (capital) Capital
40357..358	R/W	LONG	1/6(rps/s)	1~30000	Homing Accel /Decel (HA1) Return to origin acceleration/deceleration	
40359..360	R/W	LONG			Reserved	
40361..362	R/W	LONG	1/240rps	0~24000	Homing Velocity 1 (HV1) The first speed of returning to the origin	
40363..364	R/W	LONG	1/240rps	0~24000	Homing Velocity 2 (HV2) The second speed of returning to the origin	
40365..366	R/W	LONG	pulses	- 2147483647~ 2147483647	Homing Offset (HO) Return to origin offset	
40367..368	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 1 (JC1) Multi-speed control: the first1gear speed	
40369..370	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 2 (JC2) Multi-speed control: the first2gear speed	
40371..372	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 3 (JC3) Multi-speed control: the first3gear speed	
40373..374	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 4 (JC4) Multi-speed control: the first4gear speed	

M3/MBDVseries-**P2**Group**(Trajectory)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40375..376	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 5 (JC5) Multi-speed control: the first5gear speed	
40377..378	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 6 (JC6) Multi-speed control: the first6gear speed	
40379..380	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 7 (JC7) Multi-speed control: the first7gear speed	
40381..382	R/W	LONG	1/240rps	- 24000~24000	Internal Velocity Control: Speed 8 (JC8) Multi-speed control: the first8gear speed	
40383..384	R/W	LONG	ms	0~125	Jerk Time (JT) jerk time	
40385..386	R/W	LONG	ms	0~1000	Jerk Filter (KJ) low pass smoothing filter	
40387..388	R/W	LONG	ms	0~125	Interpolation Filter (FF) (Only for M3) interpolation filter	
40389..390	R/W	LONG	1/240rps	0~24000	Velocity Limit of Torque Mode (VT) Speed limit in torque mode	
40391..392	R/W	LONG	1/240rps	0~24000	Dynamic Brake Velocity (DV) Dynamic braking action speed	
40393..394	R/W	LONG	— —	0~1	No COMM Detect Enable (ZE) (Only for MBDV) Bus communication interruption detection function selection	
40395..396	R/W	LONG	ms	0~10000	No COMM Detect Time (ZS) (Only for MBDV) Bus communication interruption alarm detection time	
40397..398	R/W	LONG	— —	1~16	No COMM Detect Action (ZA) (Only for MBDV) After bus communication is interrupted,Motor action selection	

M3/MBDVseries-**P3**Group**(Encoder & Step/Dir)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40399..400	R/W	LONG	— —	1~2147483647	Electronic Gear Ratio – Numerator (EN) (Only for M3) Electronic gear ratio numerator	
40401..402	R/W	LONG	— —	1~2147483647	Electronic Gear Ratio - Denominator (EU) (Only for M3) Electronic gear ratio denominator	
40403..404	R/W	LONG	0.1μs	0~32000	Pulse Input Noise Filter (SZ) (Only for M3) Pulse input filter width	
40405..406	R/W	LONG	— —	0~31	Pulse Input Setting (PT) (Only for M3) Pulse input setting	
40407..408	R/W	LONG	pulses	0~2147483647	Position Error Limit (PF) Position Error Alarm Threshold	
40409..410	R/W	LONG	pulses/ rev	200~131072	Command Pulses per Revolution (EG) Number of pulses required per revolution	R
40411..412	R/W	LONG	— —	0~1	Second Encoder Direction (PV) The direction of the second encoder	
40413..414	R/W	LONG			Reserved	
40415..416	R/W	LONG			Reserved	
40417..418	R/W	LONG	rev	1~100	Hybrid Deviation Clear Setting (XT) Hybrid deviation clearing setting in full closed loop mode	
40419..420	R/W	LONG	pulses	0~2147483647	Hybrid Deviation Fault Threshold (XO) Position Error Alarm Threshold in Full Closed Loop Mode	
40421..422	R/W	LONG	pulses/ rev	200~100000	Second Encoder Resolution (XR) Second encoder resolution	
40423..424	R/W	LONG	— —	0~256	Pulses Output Mode (PO) Pulse frequency division output mode	
40425..426	R/W	LONG	— —	0~13107200	Pulse Output Gear Ratio - Numerator (ON) Pulse divider output ratio numerator	
40427..428	R/W	LONG	— —	0~13107200	Pulse Output Gear Ratio -Denominator (OD) Pulse frequency division output ratio denominator	
40429..430	R/W	LONG	— —	0~3	Absolute Encoder Usage (ES) Absolute encoder usage mode	
40431..432	R/W	LONG	— —	0~1	Electronic Gearing Switch (PU) Electronic gear ratio switch	
40433..434	R/W	LONG	0.01Ω	350~10000	Dynamic Brake Resistance Ohms (DR)Dynamic Brake Resistor Value	
40435..436	R/W	LONG	W	100~30000	Dynamic Brake Resistance Power (DO)Dynamic braking resistor power	
40437..438	R/W	LONG	rev	0~32766	Absolute Encoder Multi-turn Data Upper Limit @ ES=4 (FV) Absolute encoder multi-turn data upper limit@ES=4	

M3/MBDVseries-**P3**Group**(Encoder & Step/Dir)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40439..440	R/W	LONG			Reserved	

M3/MBDVseries-**P4**Group**(Analog)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40441..442	R/W	LONG			Reserved	
40443..444	R/W	LONG	1/240rps	0~24000	Analog Input Velocity Gain (AG) Analog input speed scaling	
40445..446	R/W	LONG	0.1%	0~3000	Analog Input Torque Gain (AN) Analog input torque scaling	
40447..448	R/W	LONG	mv	- 10000~10000	Analog Input 1 Offset (AV1) Analog input1Offset	
40449..450	R/W	LONG	mv	- 10000~10000	Analog Input 2 Offset (AV2) Analog input2Offset	
40451..452	R/W	LONG	mv	0~255	Analog Input 1 Deadband (AD1) Analog input1dead zone	
40453..454	R/W	LONG	mv	0~255	Analog Input 2 Deadband (AD2) Analog input2dead zone	
40455..456	R/W	LONG	0.1Hz	1~20000	Analog Input 1 Filter (AF1) Analog input1low pass filter	
40457..458	R/W	LONG	0.1Hz	1~20000	Analog Input 2 Filter (AF2) Analog input2low pass filter	
40459..460	R/W	LONG	mv	- 10000~10000	Analog Input 1 Threshold (AT1) Analog input1trigger threshold	
40461..462	R/W	LONG	mv	- 10000~10000	Analog Input 2 Threshold (AT2) Analog input2trigger threshold	
40463..464	R/W	LONG	— —	0~1	Velocity Limit Setting of Torque Control (FA1) Speed limit source setting	
40465..466	R/W	LONG			Reserved	
40467..468	R/W	LONG			Reserved	
40469..470	R/W	LONG			Reserved	
40471..472	R/W	LONG			Reserved	
40473..474	R/W	LONG	— —	1~32000	Analog Output 1 Scale (OS1) Analog output1target	
40475..476	R/W	LONG	— —	1~32000	Analog Output 2 Scale (OS2) Analog output2target	
40477..478	R/W	LONG	— —	0~5	Analog Output 1 Function (XA1) Analog output1function definition	
40479..480	R/W	LONG	— —	0~5	Analog Output 2 Function (XA2) Analog output2function definition	
40481..482	R/W	LONG			Reserved	
40483..484	R/W	LONG			Reserved	
40485..486	R/W	LONG			Reserved	
40487..488	R/W	LONG			Reserved	
40489..490	R/W	LONG			Reserved	

M3/MBDVseries-**P5**Group**(I/O)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40491..492	R/W	LONG	— —	0~46	Digital Input 1 Function (MU1) Digital input port1Function	
40493..494	R/W	LONG	— —	0~46	Digital Input 2 Function (MU2) Digital input port2Function	
40495..496	R/W	LONG	— —	0~46	Digital Input 3 Function (MU3) Digital input port3Function	
40497..498	R/W	LONG	— —	0~46	Digital Input 4 Function (MU4) Digital input port4Function	
40499..500	R/W	LONG	— —	0~46	Digital Input 5 Function (MU5) Digital input port5Function	
40501..502	R/W	LONG	— —	0~46	Digital Input 6 Function (MU6) Digital input port6Function	
40503..504	R/W	LONG	— —	0~46	Digital Input 7 Function (MU7) Digital input port7Function	
40505..506	R/W	LONG	— —	0~46	Digital Input 8 Function (MU8) Digital input port8Function	
40507..508	R/W	LONG	— —	0~46	Digital Input 9 Function (MU9) Digital input port9Function	
40509..510	R/W	LONG	— —	0~46	Digital Input 10 Function (MUA) Digital input port10Function	
40511..512	R/W	LONG			Reserved	
40513..514	R/W	LONG			Reserved	
40515..516	R/W	LONG			Reserved	
40517..518	R/W	LONG			Reserved	
40519..520	R/W	LONG	— —	0~36	Digital Output 1 Function (MO1) Digital output port1Function	
40521..522	R/W	LONG	— —	0~36	Digital Output 2 Function (MO2) Digital output port2Function	
40523..524	R/W	LONG	— —	0~36	Digital Output 3 Function (MO3) Digital output port3Function	
40525..526	R/W	LONG	— —	0~36	Digital Output 4 Function (MO4) Digital output port4Function	
40527..528	R/W	LONG	— —	0~36	Digital Output 5 Function (MO5) Digital output port5Function	
40529..530	R/W	LONG	— —	0~36	Digital Output 6 Function (MO6) Digital output port6Function	
40531..532	R/W	LONG			Reserved	
40533..534	R/W	LONG			Reserved	
40535..536	R/W	LONG			Reserved	
40537..538	R/W	LONG			Reserved	

M3/MBDVseries-**P5**Group**(I/O)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40539..540	R/W	LONG	ms	0~32000	Move Command Waiting Time When Brake Release (BD) After the brake is released,exercise waiting time	
40541..542	R/W	LONG	ms	0~32000	Servo-off Brake Engage Waiting Time (BE) after brake,Motor disable wait delay	
40543..544	R/W	LONG			Reserved	
40545..546	R/W	LONG	— —	0~10	Home Sensor (HX) origin sensor	
40547..548	R/W	LONG	ms	0~8000	Digital Input 1 Filter (FI1) digital input1filter	
40549..550	R/W	LONG	ms	0~8000	Digital Input 2 Filter (FI2) digital input2filter	
40551..552	R/W	LONG	ms	0~8000	Digital Input 3 Filter (FI3) digital input3filter	
40553..554	R/W	LONG	ms	0~8000	Digital Input 4 Filter (FI4) digital input4filter	
40555..556	R/W	LONG	ms	0~8000	Digital Input 5 Filter (FI5) digital input5filter	
40557..558	R/W	LONG	ms	0~8000	Digital Input 6 Filter(FI6) digital input6filter	
40559..560	R/W	LONG	ms	0~8000	Digital Input 7 Filter (FI7) digital input7filter	
40561..562	R/W	LONG	ms	0~8000	Digital Input 8 Filter (FI8) digital input8filter	
40563..564	R/W	LONG	ms	0~8000	Digital Input 9 Filter (FI9) digital input9filter	
40565..566	R/W	LONG	ms	0~8000	Digital Input 10 Filter (FIA) digital input10filter	
40567..568	R/W	LONG	pulses	0~2147483647	Dynamic Follow Error Threshold (PL) Dynamic following error threshold	
40569..570	R/W	LONG	pulses	0~32000	In-position Output Threshold (PD) Positioning complete signal position error threshold	
40571..572	R/W	LONG	ms	0~32000	Time Constant of Motion Output Condition (PE) Motion judgment condition count time	
40573..574	R/W	LONG	ms	0~20000	Pulse Complete Timing (TT) Pulse input completion detection time	
40575..576	R/W	LONG	1/240rps	twenty four~480	Zero Speed Width (ZV) Zero speed judgment threshold	
40577..578	R/W	LONG	1/240rps	twenty four~24000	Speed Coincidence Width (VR) Speed consistent fluctuation range	
40579..580	R/W	LONG	1/240rps	0~24000	Target Value of AT-speed (VV) Judging that the speed reaches the target value	

M3/MBDVseries-**P5**Group**(I/O)**						
Register	**Access**	**Data Type**	**Units**	**Range**	**Description**	**SCL Register**
40581..582	R/W	LONG	0.1%	0~3000	Torque Arrival Width (TV) Torque reaches the fluctuation range	
40583..584	R/W	LONG	pulses	- 2147483647~ 2147483647	Near Target Position (DG) absolute arrival position	
40585..586	R/W	LONG	pulses	- 2147483647~ 2147483647	Positive Software Limit (LP) Positive soft limit	
40587..588	R/W	LONG	pulses	- 2147483647~ 2147483647	Negative Software Limit (LM) Reverse soft limit	
40589..590	R/W	LONG	— —	- 4~35	Homing Method (HE) Back to origin method	
40591..592	R/W	LONG	— —	1~8	Emergency Stop (EO) Emergency stop mode selection	
40593..594	R/W	LONG	— —	0~1	Zero Speed Clamp Function in Velocity Mode (MS) Zero-speed clamp function in speed mode	

illustrate:

MBDV-2X-520AC The two axes of the drive can be respectively configured as two different node addresses,sending the command pair**Modbus** When register operation,The following three methods are supported.

The first:Use different node addresses to operate the registers of the two axes respectively,ModbusRegister address see above table.

the second:use axis1The node addresses of the two axes operate on the registers,axis1ofModbusRegister address see above table,axis 2of Modbusregister address = axis1ofModbusregister address +1000

the third:use axis1The node addresses of the two axes operate on the registers,axis1and axis2the same oneModbusThe addresses of the registers are interleaved according to the data type,The starting address is42001.E.g,The first register for both axes is32number of digitsAlarm Code,The second register is32 number of digitsStatus Code,The third register is16number of digitsDigital Output Status,

but:

42001..002:axis1of40001..002register

42003..004:axis2of40001..002register

42005..006:axis1of40003..004register

42007..008:axis2of40003..004register

42009: axis1of40005register

42010: axis2of40005register

- For register data type is16number of

digits 42000+2n-1:axis1of40000+nregister

42000+2n:axis2of40000+nregister

- For register data type is32number

of digits (42000+2n-1)..(42000+2n): axis1of(40000+n)..(40000+n+1)register

(42000+2n+1)..(42000+2n+2):axis2of(40000+n)..(40000+n+1)register n

range:1 ⩽ n ⩽ 593

Servo system**M3**,**MBDV**Drive Alarm Code(main code)surface:

register	bit	illustrate		bit	illustrate
40001 40002	0	*Position error overrun*		16	Drive main circuit power input phase loss
	1	Reverse prohibit limit		17	*Safe torque off(**STO**)*
	2	Positive prohibit limit		18	reserved
	3	*over temperature*		19	*Motor speed exceeds limit*
	4	*internal error*		20	Drive undervoltage
	5	*Supply voltage out of range*		twenty one	*emergency stop*
	6	reserved		twenty two	*Second encoder not connected*
	7	*Drive overcurrent*		twenty three	*Full closed loop hybrid deviation overrun*
	8	reserved		twenty four	Absolute encoder battery undervoltage
	9	*Motor encoder not connected*		25	Absolute position lost
	10	Communication exception		26	Absolute position overflow
	11	reserved		27	***RS485***or***CANopen***Bus communication interrupted
	12	vent failure		28	*Absolute encoder multi-turn error*
	13	Motor overload protection		29	*Abnormal motor action protection*
	14	reserved		30	***EtherCAT**communication error*
	15	Unusual start alarm		31	Back-to-origin parameter configuration error

Servo system**M3**,**MBDV**Drive Alarm Code(Secondary code)surface:

register **(40001..02)** bit	illustrate	register **(40067..68)** bit	illustrate
3	*over temperature*	***5***	*Drive processor overtemperature*
		6	*Drive power module over temperature*
		7	*Motor overtemperature*
4	*internal error*	***8***	*Parameter read failed*
		9	*Internal voltage error*
		10	*reserved function,keep as "**0**"*
		11	*reserved function,keep as "**0**"*
		12	***FPGA**mistake*
		13	*Failed to save parameters*
		14	*Motor encoder communication error*
5	*Supply voltage out of range*	***15***	*Drive overvoltage*
		16	*Drive low voltage*
7	*overcurrent*	***2***	*Low side overcurrent*
		3	*High-end overcurrent*
		4	*Reading overcurrent*
15	Unusual startup warning	17	calledQblock is empty
		18	The motor is commanded to run when it is not enabled
		19	I/OSignal function multiplexing
29	*Abnormal motor action protection*	***twenty four***	*Motor stall protection*
		25	*Motor anti-collision protection*

DC brushless system**BLD**Driver Alarm Code Table:

register	bit	illustrate
40001	0	reserved
	1	reserved
	2	reserved
	3	*drive over temperature*
	4	*Drive internal voltage error*
	5	*Drive overvoltage*
	6	*Drive low voltage*
	7	*Drive overcurrent*
	8	*Winding open circuit*
	9	*Hall signal error*
	10	Communication exception
	11	Failed to save parameters
	12	de-enable
	13	Motor overload status
	14	*memory error*
	15	reserved

Remark:The italic and bold font alarm items in the above table represent the drive reporting a fault,When a fault alarm occurs, the motor is disabled.

2.status code

Status code is used to indicate the current working status of the drive,The user can know the specific status information by querying the status register.Each bit of the status code register represents different status information,when a bit is set1When the drive is in the state defined by this bit,For the definition of each bit, please refer to the following table.

Stepper and Servo**M2**,**MDX**Drive Status Code Table:

register	bit	illustrate
40002	0	Enable
	1	sampling(The software oscilloscope function is enabled)
	2	Drive reports failure
	3	Movement in place
	4	in motion
	5	jogging running
	6	decelerating
	7	waiting for input signal(e.g. executeWIinstruction)
	8	parameter saving
	9	drive warning
	10	back to origin
	11	waiting time(e.g. executeWT,WDinstruction)
	12	internal use
	13	Encoder detection
	14	Qprogram is running
	15	initialization(step system),Servo ready(Servo system)

Servo system**M3**,**MBDV**Drive Status Code Table:

register	bit	illustrate		bit	illustrate
40003 40004	0	Servo enable		16	CSPfollow
	1	sampling(LunaThe software oscilloscope function is enabled)		17	same speed
	2	Drive reports failure		18	zero speed
	3	Movement in place		19	Torque arrives
	4	in motion		20	same torque
	5	jogging running		twenty one	The second group gain is working
	6	decelerating		twenty two	The second control mode is working
	7	waiting for input signal(e.g. executeWIinstruction)		twenty three	speed to reach
	8	parameter saving		twenty four	back to origin complete
	9	drive warning		25	reserved
	10	back to origin		26	reserved
	11	waiting time(e.g. executeWT,WDinstruction)		27	reserved
	12	internal use		28	reserved
	13	Encoder detection		29	reserved
	14	Qprogram is running		30	reserved
	15	Servo ready		31	reserved

DC brushless system**BLD**Drive Status Code Table:

register	bit	illustrate
40002	0	Enable
	1	reserved
	2	Drive reports failure
	3	reserved
	4	reserved
	5	jogging running
	6	decelerating
	7	reserved
	8	reserved
	9	drive warning
	10	reserved
	11	reserved
	12	internal use
	13	reserved
	14	reserved
	15	reserved

Remark:When the drive has a fault alarm,Drive fault and warning states are set simultaneously**1**.

appendix7supportModbus/RTUprotocolMOONS'Drive model

series	model	Firmware version
TSMseries	TSM11Q-xxx	1.05Eafter
	TSM17Q-xxx	1.05Aafter
	TSM23Q-xxx	1.05Aafter
	TSM24Q-xxx	1.05Aafter
SSMseries	SSM17Q-xxx	1.05Aafter
	SSM23Q-xxx	1.05Aafter
	SSM24Q-xxx	1.05Aafter
TXMseries	TXM24Q-xxx	1.05Aafter
SSseries	SS03-Qx	1.06Aafter
	SS05-Qx	1.06Aafter
	SS10-Qx	1.06Aafter
RSseries	RS03-Qx	1.06Aafter
	RS06-Qx	1.06Aafter
STMseries	STM11Q-xxx	1.20Gafter
	STM17Q-xxx	1.06Gafter
	STM23Q-xxx	1.06Gafter
	STM24QF-xxx	1.06Eafter
SWMseries	SWM24QF-xxx	1.06Fafter
STseries	MSST5-Q-xx	1.06Dafter
	MSST10-Q-xx	1.06Dafter
STBseries	MSSTB05-R	1.05Lafter
	MSSTB10-R	1.05Lafter
STFseries	STF03-R	1.00Cafter
	STF05-R	1.00Cafter
	STF06-R	1.00Cafter
	STF10-R	1.00Cafter
	STF05-RH	
	STF10-RH	
STACseries	MSSTAC5-Q-xx-2V	1.06Bafter
M2series	M2DV-XXXXR	1.00Cafter
	M2DC-XXXXR	1.00Rafter
	M2DC-XXXXR-H	1.01Dafter
M3series	M3DV-XXXXRX	1.00Gafter
MBDVseries	MBDV-520AC	1.00Bafter
	MBDV-2X-520AC	1.00Bafter
MDXseries	MDXXXXXXXR	1.07Dafter
BLDCseries	BLD05-R	1.11Qafter
	BLD10-R	1.01Pafter

appendix8supportModbus/TCPprotocolMOONS'Drive model

series	model	Firmware version
TSMseries	TSM23XXG-D/IP	
	TSM23X3B-D/IP	2.00Eafter
	TSM34Q-XDG	1.06Dafter
	TSM34IP-XDG	1.06Dafter
TXMseries	TXM24Q-3EG	
	TXM24IP-3EG	
	TXM24X3B-IP/IPE	2.00Eafter
	TXM34Q-XDG	1.06Dafter
	TXM34IP-XDG	1.06Dafter
STMseries	STM23Q-xEx	1.07Gafter
SSDCseries	SSDC03-D/IP	1.01Bafter
	SSDC06-D/IP	1.01Bafter
	SSDC10-D/IP	1.01Bafter
STFseries	STF03-D/IP	1.00Cafter
	STF05-D/IP	1.00Cafter
	STF06-D/IP	1.00Cafter
	STF10-D/IP	1.00Cafter
M2series	M2DV-XXXXE	1.00Qafter
	M2DV-XXXXD	1.00Qafter
	M2DV-XXXXIP	1.00Qafter
	M2DC-XXXXD	1.00Rafter
	M2DC-XXXXIP	1.00Rafter
	M2DC-XXXXD-H	1.01Dafter
	M2DC-XXXXIP-H	1.01Dafter
MDXseries	MDXXXXXXXD	1.07Eafter
	MDXXXXXXXIP	1.07Eafter

connectMOONS'

客户咨询中心
400-820-9661

MOONS headquarters

No. 168, Mingjia Road, Minbei Industrial Zone, Minhang District, Shanghai

Postcode: 201107
Tel: +86 (0)21 52634688
Fax: +86 (0)21 52634098

MOONS' International Trade

4th Floor, Building 30, No. 69, Guijing Road, Caohejing Emerging Technology Development Zone, Shanghai

Postcode: 200233
Tel: +86 (0)21 64952755
Fax: +86 (0)21 64951993

Domestic office

Shenzhen

Room 2209, Shenzhen Kerry Center, No. 2008, Renmin South Road, Luohu District,

Shenzhen Postcode: 518001
Tel: +86 (0)755 25472080
Fax: +86 (0)755 25472081

Beijing

Room 816, Block B, China Electronics Building, No. 3 Danling Street, Haidian District, Beijing

Postcode: 100080
Tel: +86 (0)10 58753312
Fax: +86 (0)10 58752279

Nanjing

Room 1101/1102, 11th Floor, Building 2, New Town Development Center, No. 126 Tianyuan Middle Road,

Jiangning District, Nanjing

Postcode: 211106
Tel: +86 (0)25 52785841
Fax: +86 (0)25 52785485

Qingdao

Room 1012, Zhuoyue Building, No. 16 Fengcheng Road, Shibei District, Qingdao

Postcode: 266000
Tel: +86 (0)532 80969935
Fax: +86 (0)532 80919938

Wuhan

Room 3001, Shimao Building, No. 686 Jiefang Avenue, Jianghan District,

Wuhan Postcode: 430022
Tel: +86 (0)27 85448742
Fax: +86 (0)27 85448355

Chengdu

Room 1917, Weston Federal Building, No. 19, Section 4, Renmin South Road, Wuhou District,

Chengdu Zip code: 610041
Tel: +86 (0)28 85268102
Fax: +86 (0)28 85268103

Xi'an

Room 1006, Block D, Wangzuo International City, No. 1 Tangyan Road, Xi'an

Postcode: 710065
Tel: +86 (0)29 81870400
Fax: +86 (0)29 81870340

Ningbo

Room 309, Tower B, Taifu Plaza, No. 565, Jingjia Road, Jiangdong District, Ningbo City, Zhejiang Province

Postcode: 315040
Tel: +86 (0)574 87052739
Fax: +86 (0)574 87052365

Guangzhou

Room 06, 40th Floor, Tower B, Yew Chung Plaza, No. 9 Linhe West Road, Tianhe District,

Guangzhou Postcode: 510610
Tel: +86 (0)20 38010153
Fax: +86 (0)20 38103661

North American Company

MOONS'INDUSTRIES (AMERICA), INC.
1113 North Prospect Avenue, Itasca, IL 60143 USA
Tel: +1 630 8335940
Fax: +1 630 8335946

APPLIED MOTION PRODUCTS, INC.
404 Westridge Dr. Watsonville, CA 95076, USA
Tel: +1 831 7616555
Fax: +1 831 7616544

LIN ENGINEERING, INC.
16245 Vineyard Blvd., Morgan Hill, CA 95037
Tel: +1 408 9190200
Fax: +1 408 9190201

European companies

MOONS'INDUSTRIES (EUROPE) SRL
Via Torri Bianche n.1 20871 Vimercate(MB) Italy
Tel: +39 039 6260521
Fax: +39 039 9631409

Southeast Asian companies

MOONS'INDUSTRIES (SOUTH-EAST ASIA) PTE. LTD.
33 Ubi Avenue 3 #08-23 Vertex Singapore 408868
Tel: +65 66341198
Fax: +65 66341138

Japanese company

MOONS'INDUSTRIES JAPAN CO., LTD.
222-0033

Kanagawa Prefecture Yokohama City, Konkita-ku, Shin-Yokohama 2-chome 12-12

Shin-Yokohama Mitsunobu 6F 601

Phone number: +81 (0)45 4755788
ァックス: +81 (0)45 4755787