No. DOC1102114

Technical Data

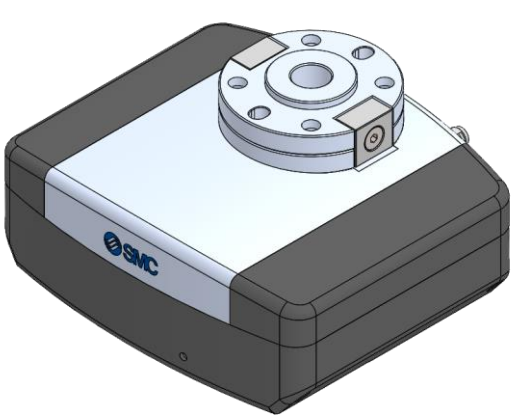
PRODUCT NAME

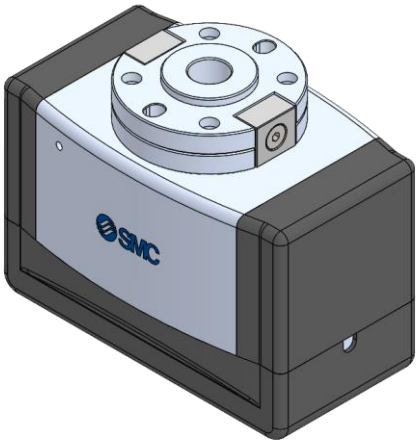
Electric Gripper for Collaborative Robots

MODEL / Series / Product Number

LEHR series

-Serial Communication (Modbus)





SMC Corporation

Contents

Precautions and Disclaimer

When using the serial communication (Modbus) of the LEHR series of electric grippers ("LEHR Serial Communication") based on this document, the following "Disclaimer" shall apply.
Please check and accept the following before using LEHR serial communication.

Precautions

Read the "Safety Instructions", " Common Precautions", "Specific Product Precautions" and "Specifications" described in the manual for the equipment when using this LEHR serial communication.

Disclaimer

SMC cannot take any responsibility for any damage incurred using LEHR serial communication.

1.Outline

This manual explains how to use the LEHR electric gripper when communicating via Modbus RTU over an RS485 transmission line. LEHR can perform the following operations via Modbus RTU communication.

1 Motion instruction for registered motion data.

2 Motion Data Edit.

3 Readout of Position, Velocity and Thrust data.

2. Communication

2.1 Communication

Item	Methods and conditions
Electrical Specifications	RS485
Communication standard	Modbus RTU
Communication method	2-wire and half-duplex communication
Communication Speed(bps)	19200
Bit length	8bit
Stop bit	1 bit
Parity	None
Endian	Big-Endian

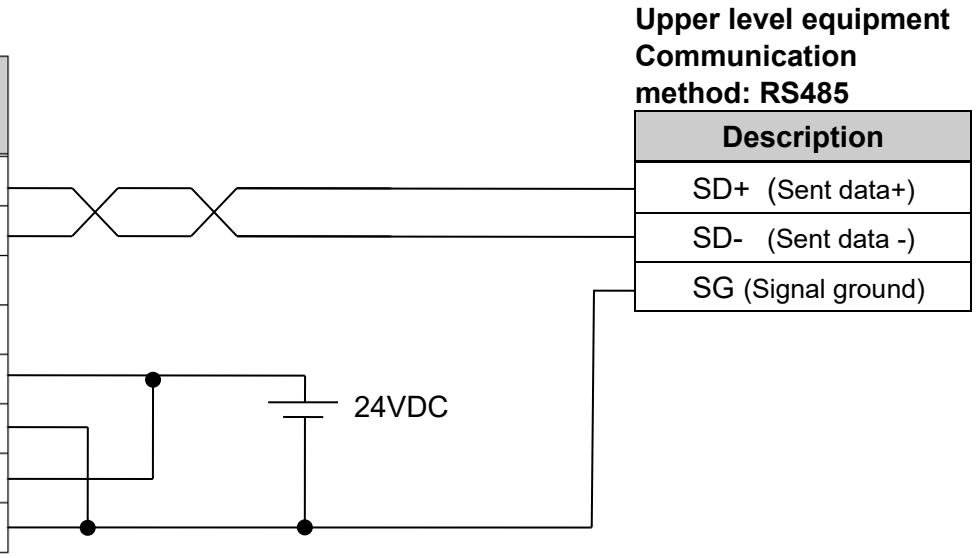
2. 2 LEHR RS485 wiring diagrams (for LEHR*-*-011: Universal Robot)

The following is an example of controller wiring in which one LEHR is connected to a higher-level communication device.

The case of LEHR*-*-011 (for Universal Robot) is shown here.

LEHR*-011

Description	Pin No.
RS485+	1
RS485-	2
-	3
-	4
24VDC	5
0 V	6
24VDC	7
0 V	8



2.3 Communication frame

2. 3. 1 Query

A message sent by the master is called a query. The communication frame of a query is as follows

Slave Address	Function	Register Address	Data	CRC Check
1 Byte	1 Byte	2 Bytes	N Byte	2 Bytes

■ **Slave Address**

Set the slave address of the LEHR.

The default ID is "1".

Function

Function codes and functions that can be used with this controller are shown below.

Code (Hex)	Modbus Name	Function
03h	Read holding register	Read motor information, parameters, and motion data
05h	Single coil writing	Execution of Commands
10h	Multiple Holding Register Writing	Writing parameters and motion data

■ **Data**

The data corresponds to each Function code, and the data length varies depending on the Function code.

■ **CRC Check**

16-bit CRC Check code for communication data.

When sending, the CRC data is calculated and added to the transmission frame before transmission.

When receiving, the CRC of the received data is calculated and compared with the CRC data added to the reception frame. If the CRC's do not match, no response (no response) is given.

2. 3. 2 Response

The slave responds to the query.

The message structure of the response is the same as for a query as follows.

Slave Address	Function	Register Address	Data	CRC Check
1 Byte	1 Byte	2 Bytes	N Bytes	2 Bytes

■ **Normal response**

The query is received successfully from the master, the slave executes the process successfully and returns a query corresponding to the function code.

■ **No response**

There are cases where the master device does not respond to queries, even though it has received them.

The following are possible causes of no response.

-Communication error: Ensure that the communication settings such as baud rate and parity match those of the master.

- Incorrect slave address: If a non-existent slave address is specified, no response is returned.

- Incorrect function Code: No response is given to function codes other than 0x03, 0x05, and 0x10.

-Incorrect CRC Check: If the CRC check data appended to the query is incorrect, there will be no response.

■ **Exception response**

The following exception response is returned when an irregularity occurs in the message frame of a query from the master.

Slave Address	Function	Register Address	Data	CRC Check
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

In the case of an exception response, all bytes from the 6th byte onwards will be set to 1.

The causes of an exception response include the following.

- Incorrect data in the query message frame from the master.

- Function code 0x03, 0x05, or 0x10 is incorrectly set or the register address is incorrect.

3. Unit conversion

The data to be written and read by Modbus communication in LEHR are as follows.

Position: Internal motor rotation system unit [pulse].
Speed: Internal motor rotation system unit [100pulse/s].
Gripping force: Output [%] when the rated value of the internal motor is 100

When handling with the upper device side, set the input value converted to the unit using the following formula.
Decimals will be rounded down.

■ Position

Position [pulse] = Position [mm/s] / 3.333 * 1000

■Speed

Speed [100pulse/s] = Speed [mm/s] / 3.333 * 10

■Gripping force

Gripping force[%] = Gripping force[N] * 0.3125 + 1.25

4. Holding register

4.1 Content

The following is an overview of the registers that can be accessed via Modbus RTU communication.

Description	Address (HEX)	Content
Motor information area	0102-0116	Information such as the current position can be read using function code 0x03.
Motion data area	0280 to 033F	Motion data can be read with function code 0x03 and written using 0x10. Motion data consists of position data, speed data, acceleration data, pushing torque data, torque limit data and motion mode. These data can be set first, and then the motion started with the execution.
Execution command area	0580-0599	Various commands can be executed using function code 0x05.

4.2 Motor Information Area

The status of the LEHR's motor is stored.

Various information can be read using function code 0x03.

Address (HEX)	Parameter name	Byte	Data type	Content
0102	Current position(Upper)	2	Integer	Displays the Current Position (Unit: 1 pulse)
0103	Current position (Lower)	2		
0104	Current speed (Upper)	2		Displays the current speed (Unit: 1 pulse/s)
0105	Current speed (Lower)	2		
0106	Current torque (Upper)	2		Displays the current thrust (Unit: %)
0107	Current torque (Lower)	2		
0108	Motor Status (Upper)	2	Bit	Displays the motor status (Refer to 4.2.1 Motor Status)
0109	Motor status (Lower)	2		
010A	Warning Status (Upper)	2		Displays the warning status (Refer to 4.2.2 Warning Status)
010B	Warning Status (Lower)	2		
010C to 0115	Unused	-		-
0116	Execution command status (Upper)	2		Displays the ON/OFF status of executed commands at once (Refer to 4.2.3 Execution Command Status)
0117	Execution command status (Lower)	2		

4.2.1 Motor Status (0x0108, 0x0109)

Various statuses including motor errors are as follows

Bit	Description	Detail	Classification
B0	Location deviation overflow	Alarm occurs when Position deviation exceeds the threshold value	Alarm
B1	Overvoltage alarm	Alarm occurs when input voltage exceeds 30 V	Alarm
B2	Unused	-	-
B3	INP	During in-position range with Servo on	-
B4	Servo OFF	Servo-off state	-
B5	During pushing operation	During pushing operation	-
B6	Over current alarm	Alarm occurs when excessive current such as regenerative current flows	Alarm
B7	Temperature abnormality alarm	Alarm occurs when internal temperature exceeds 110°C	Alarm
B8	Unused	-	-
B9	Low voltage alarm	Alarm occurs when the main power supply falls below 18 V	Alarm
B10 to B31	Unused	-	-

⚠ Caution

When the motion data is executed outside the in-position range, all bits of 0x0108 and 0x0109 will be set to 0.

■About Alarms

When an alarm occurs, the motor status bit corresponding to the alarm content is set, the alarm output is turned on, and the servo is turned off. After removing the cause of the alarm and clearing the alarm (0x0582), the servo on (0x0580) can be accepted.

4.2.2 Warning Status (0x010A, 0x010B)

The details of the warning are stored in the following bits of 0x010A and 0x010B.

Bit	Description	Details
B0	Pushdown warning	Target Position is reached before the pushing operation is completed.
B1	Unused	-
B2	Unused	-
B3	Unused	-
B4	Unused	-
B5	Execution Warning	A motion command was input in a state where operation was not possible.
B6	Temperature warning	Temperature inside the driver case exceeds the K61 setting.
B7	Overload Warning	Load factor exceeds K62 setpoint.
B9	Work Lost Warning	After completion of pushing, the torque fell below a certain value.
B10	Overload Warning	Overloading
B11 to B31	Unused	-

■About Warnings

The warning does not turn off the servo even if it occurs. It is released when the next operation is executed, or the stop command is accepted.

4.2.3 Execution command status (0x0116, 0x0117)

The on/off status of the execution command area is stored.

Bit	Content	Corresponding address (HEX)	Motion data execution conditions
B0	Servo ON/OFF	0580	1
B1	Stop	0581	0
B2	Alarm reset	0582	0
B3	Disabled	0583	-
B4	Move to Position 0	0584	0
B5	Disabled	0585	0
B6	Disabled	0586	-
B7	Software reset	0587	-
B8	Disabled	0588	0
B9	Save	0589	-
B10	Execute motion Data 1	058A	0
~	~	~	0
B25	Execute motion data 16.	0599	0
B26 to B31	Disabled	-	-

⚠Caution

Motion data can be executed only when the execution command status matches the "Motion Data Execution Conditions" in the table below.

4.3 Motion Data Area

Motion data can be read using function code 0x03 and written using 0x10.

Motion data consists of Position data, Velocity data, Gripping Torque data, Positioning Torque data and Motion Mode. These data are set first, and then the motion can be started by the execution command.

Register address (HEX)	Description	Input range		Content
		Setting value	Reference value	
0280	Position data1 (Upper)	0 to 15001 [Unit: pulse].	0 to 50 [Unit: mm].	Set Position data
0281	Position data1 (Lower)			
...	...			
029E	Position data 16 (Upper)			
029F	Position data 16 (Lower)			
02A0	Speed data 1 (Upper)	<Positioning> 15-300 [Unit: 100pulse/s]. <Grip> 1500 to 9000 [Unit: pulse/s].	<Positioning> 5 to 100 [Unit: mm/s]. <Grip> 5 to 30 [Unit: mm/s].	Set speed data
02A1	Speed data 1 (Lower)			
...	...			
02BE	Speed data 16 (Upper)			
02BF	Speed data 16 (Lower)			
02E0	Gripping torque data 1 (Upper)	20-45 [Unit: %]	60 to 140 [Unit: N].	Set torque for gripping operation
02E1	Gripping torque data 1 (Lower)			
...	...			
02FE	Gripping torque data16 (Upper)			
02FF	Gripping torque data16 (Lower)			
0300	Positioning torque data 1 (Upper)	20-54 [Unit: %]	60 to 169 [Unit: N].	Set torque for positioning operation
0301	Positioning torque data 1 (Lower)			
...	...			
031E	Positioning torque data 16 (Upper)			
031F	Positioning torque data 16 (Lower)			
0320	Operation mode 1 (Upper)	-	-	Set operation mode 0: Positioning operation 2: Gripping motion
0321	Operating mode 1 (Lower)			
...	...			
033E	Operation mode 16 (Upper)			
033F	Operating mode 16 (Lower)			

⚠ Caution

Please note that changes to motion data will not be saved unless the save (0x0589) command is executed. Also note that if motion data is changed, the SMC plug-in software for collaborative robots may not operate correctly.

4.4 Execution command area

The following command can be executed by re-writing the register using function code 0x05.

Register address (HEX)	Description	Content
0580	Servo ON/OFF	Switches servo ON/OFF
0581	Stop	Stops operation
0582	Alarm reset	Reset alarm
0583	Disabled	Starts origin detection operation.
0584	Move to Position 0	Move to Position 0 pulse
0585	Disabled	Reset the current Position to 0 pulse
0586	Disabled	Sets the current Position to the target Position
0587	Software reset*.	Same as turning the power back on.
0588	Disabled	Sets the current Position as the origin.
0589	Save*.	Saves changes to motion data
058A	Execute motion data 1	Executes motion data 1 to 16 and Restarts the stopped operation.
058B	Execute motion data 2	
058C	Execute motion data 3	
058D	Execute motion data 4.	
058E	Execute motion data 5.	
058F	Execute motion data 6.	
0590	Execute motion data 7.	
0591	Execute motion data 8.	
0592	Execute motion data 9.	
0593	Execute motion data 10.	
0594	Execute motion data 11.	
0595	Execute motion data 12.	
0596	Execute motion data 13.	
0597	Execute motion data 14.	
0598	Execute motion data 15.	
0599	Execute motion data 16.	

⚠ Caution

Software reset (0x0587) and Save (0x0589) commands must be performed when operation is stopped.

After sending a command, wait more than 10 msec before sending the next command.

5. Example of operation

An example of the actual operation is shown below.

5.1 Servo-on

1.Execute Servo-ON (0x580).

- **Query**

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (U)**	05	Servo-on (0x0580)
	Starting address (L)	80	
	Write value (U)	FF	ON : 0xFF00 OFF : 0x0000
	Write value (L)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

- **Response**

The same content as the query is returned.

2. Check the Servo-off (Bit 4) of the Motor status (0x0108-0x0109) and confirm that the Servo is turned on.

- **Query**

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read out
DATA	**Starting address (U)**	01	Motor Status Start address (0x0108)
	Starting address (L)	08	
	Number of registers (U)	00	Number of registers read out 2 (0x0108-0x0109)
	Number of registers (L)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

- **Response**

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read out
DATA	**Number of data bytes**	04	Twice the number of registers to be read out
	Readout value 1 (U)	00	Value of address (0x0108)
	Readout value 1 (L)	00	
	Readout value 2 (U)	00	Value of address (0x0109)
	Readout value 2 (L)	04	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

5.2 Positioning operation

1. Set Position to Motion Data 4.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (U)**	02	Position data 4 (0x0286)	
	Starting address (L)	86		
	Number of registers (U)	00	Number of registers to be written: 2	
	Number of registers (L)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (U)	00	Address (0x0286)	20 [mm] = 6000 [pulse] 0d6000 = 0x00001770
	Write value 1 (L)	00		
	Write value2 (U)	17	Address (0x0287)	
	Write value 2 (L)	70		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	02	Same value as query
	Starting address (Lower)	86	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

2. Set the speed to Motion Data 4.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (U)**	02	Speed data 4 (0x02A6)	
	Starting address (L)	A6		
	Number of registers (U)	00	Number of registers to be written: 2	
	Number of registers (L)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x02A6)	100 [mm/s] = 300 [100pulse/s]. 0d300=0x0000012C
	Write value 1 (Lower)	00		
	Write value2 (Upper)	75	Address (0x02A7)	
	Write value 2 (Lower)	33		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	02	Same value as query
	Starting address (Lower)	A6	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

3. Set positioning torque data to Motion Data 4.

▪ Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	03	Speed data 4 (0x0306)	
	Starting address (Lower)	06		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x0306)	207 [N] = 54 [%]. 0d54 = 0x00000036
	Write value 1 (Lower)	00		
	Write value2 (Upper)	00	Address (0x0307)	
	Write value 2 (Lower)	36		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

▪ Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	03	Same value as query
	Starting address (Lower)	06	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

4. Set the operation mode to Motion Data 4.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	03	Operation mode (40x0326)	
	Starting address (Lower)	26		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x0320)	0: Positioning operation = 0x00000000
	Write value 1 (Lower)	00		
	Write value2 (Upper)	00	Address (0x0321)	
	Write value 2 (Lower)	00		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	03	Same value as query
	Starting address (Lower)	26	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

5. Save the Motion data.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Save (0x0589)
	Starting address (Lower)	89	
	Write value (Upper)	FF	ON : 0xFF00
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

6. Reset the DATA of the save command.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	save (0x0589)
	Starting address (Lower)	89	
	Write value (Upper)	00	OFF : 0x0000
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

7. Execute Motion Data 4.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Motion data 4 execution (0x058D)
	Starting address (Lower)	8D	
	Write value (Upper)	FF	ON: 0xFF00
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

8. To execute the next operation, reset the DATA of motion data 4 when the operation starts.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Motion data 3 execution (0x058D)
	Starting address (Lower)	8C	
	Write value (Upper)	00	OFF : 0x0000
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

9. Check the in-position (Bit 3) of the motor status to confirm that positioning is complete.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read
DATA	**Starting address (Upper)**	01	Start address of motor status (0x0108)
	Starting address (Lower)	08	
	Number of registers (Upper)	00	Number of registers to be read: 2 (0x0108-0x0109)
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read
DATA	**Number of data bytes**	04	Twice the number of registers to be read
	Readout value 1 (Upper)	00	Value of address (0x0108)
	Readout value 1 (Lower)	00	
	Readout value 2 (Upper)	00	Value of address (0x0109)
	Readout value 2 (Lower)	04	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

5.3 Gripping operation

1. Set Position to Motion Data 3.

・**Query**

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	02	Position data 4 (0x0284)	
	Starting address (Lower)	84		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x0284)	50 [mm] =15001 [pulse]. 0d15001 = 0x00003A99
	Write value 1 (Lower)	00		
	Write value2 (Upper)	3A	Address (0x0285)	
	Write value 2 (Lower)	99		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・**Response**

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	02	Same value as query
	Starting address (Lower)	84	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

2. Set the speed to Motion Data 3.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	02	Speed data 3 (0x02A4)	
	Starting address (Lower)	A4		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x02A4)	30 [mm/s] =90 [100pulse]. 0d9000 = 0x0000005A
	Write value 1 (Lower)	00		
	Write value2 (Upper)	23	Address (0x02A5)	
	Write value 2 (Lower)	28		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	02	Same value as query
	Starting address (Lower)	A4	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

3. Set positioning torque data to Motion Data 3.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	02	Gripping torque data 3 (0x02E4)	
	Starting address (Lower)	E4		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x0306)	140 [N] = 45 [%]. 0d45 = 0x0000002D
	Write value 1 (Lower)	00		
	Write value2(Upper)	00	Address (0x0307)	
	Write value 2 (Lower)	2D		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	02	Same value as query
	Starting address (Lower)	E4	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

4. Set the operation mode to Motion Data 3.

・Query

Description		Value (HEX)	Content	
Address		01	Slave ID	
Function		10	Holding Register Write	
DATA	**Starting address (Upper)**	03	Operation mode 3 (0x0324)	
	Starting address (Lower)	24		
	Number of registers (Upper)	00	Number of registers to be written: 2	
	Number of registers (Lower)	02		
	Number of data bytes	04	Twice the number of registers to be written	
	Write value 1 (Upper)	00	Address (0x0324)	2: Gripping operation = 0x00000002
	Write value 1 (Lower)	00		
	Write value2 (Upper)	00	Address (0x0325)	
	Write value 2 (Lower)	02		
CRC16 (L)		Calculated value	CRC16 calculation results	
CRC16 (U)		Calculated value		

・Response

Description		Value (HEX)	Content
Address		01	Same value as query
Function		10	Same value as query
DATA	**Starting address (Upper)**	03	Same value as query
	Starting address (Lower)	24	
	Number of registers (Upper)	00	Same value as query
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

5. Save the Motion data.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Save (0x0589)
	Starting address (Lower)	89	
	Write value (Upper)	FF	ON : 0xFF00
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

6. Reset the DATA of the Save command.

Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Save (0x0589)
	Starting address (Lower)	89	
	Write value (Upper)	00	OFF : 0x0000
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

7. Execute Motion Data 3.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Motion data 4 execution (0x058C)
	Starting address (Lower)	8C	
	Write value (Upper)	FF	ON : 0xFF00
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

8. To execute the next operation, reset the DATA of Motion Data 3 when the operation starts.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		05	Single coil write
DATA	**Starting address (Upper)**	05	Motion data 3 execution (0x058C)
	Starting address (Lower)	8C	
	Write value (Upper)	00	OFF : 0x0000
	Write value (Lower)	00	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

・Response

The same content as the query is returned.

9. Check the in-position (Bit 3) of the Motor status to confirm that positioning is complete.

・Query

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read
DATA	**Starting address (Upper)**	01	Motor Status Start address (0x0108)
	Starting address (Lower)	08	
	Number of registers (Upper)	00	Number of registers to be read: 2 (0x0108-0x0109)
	Number of registers (Lower)	02	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

Response

Description		Value (HEX)	Content
Address		01	Slave ID
Function		03	Holding register read
DATA	**Number of data bytes**	04	Twice the number of registers to be read
	Readout value 1 (Upper)	00	Value of address (0x0108)
	Readout value 1 (Lower)	00	
	Readout value 2 (Upper)	00	Value of address (0x0109)
	Readout value 2 (Lower)	04	
CRC16 (L)		Calculated value	CRC16 calculation results
CRC16 (U)		Calculated value	

6.Troubleshooting

6.1 When Modbus RTU communication is not established

Please refer to "2.3.2 Response".

6. 2 When motion data is not executed

If the operation does not start, the possible causes are as follows:

- Motion data is not set correctly
- The target Position is set to the same value as the current Position.
- Positioning torque is set outside the range and cannot be driven.
- Operation mode is not set.
- Motion data execution command is still set.
  (Refer to "4.2.3 Execution command status")
- The Servo is turned off.

Revision history
November 2024 : Frist edition October 2024 : Indemnification July 2025 Corrected unit conversions

SMC Corporation

URL https://www.smcworld.com