BRUSHLESS DC MOTOR DRIVER

BLR400A
User manual

Giden Electronics

Всё для машиностроения и автоматизации

www.giden.ru

sale@giden.ru

+7 (495) 225-54-52

1. Brief introduction

BLR400A is designed with high-performance and low-cost which used for 69W~400W brushless DC motors. It supports Modbus communication protocol, and also provides more flexible choices for users among actual practice.

1.1 Features

- Support Modbus communication protocol, suitable for touch screen or PC
- Built-in potentiometer RV speed setting and external analog speed setting
- Input voltage 15~48VDC
- Working temperature -20℃~+60℃
- Closed loop control to ensure uniform speed when loading
- Works for 69W to 400W BLDC motor
- 10 seconds waiting time for locked-rotor

2. Electrical performance and environmental indicators

2.1 Electrical parameters

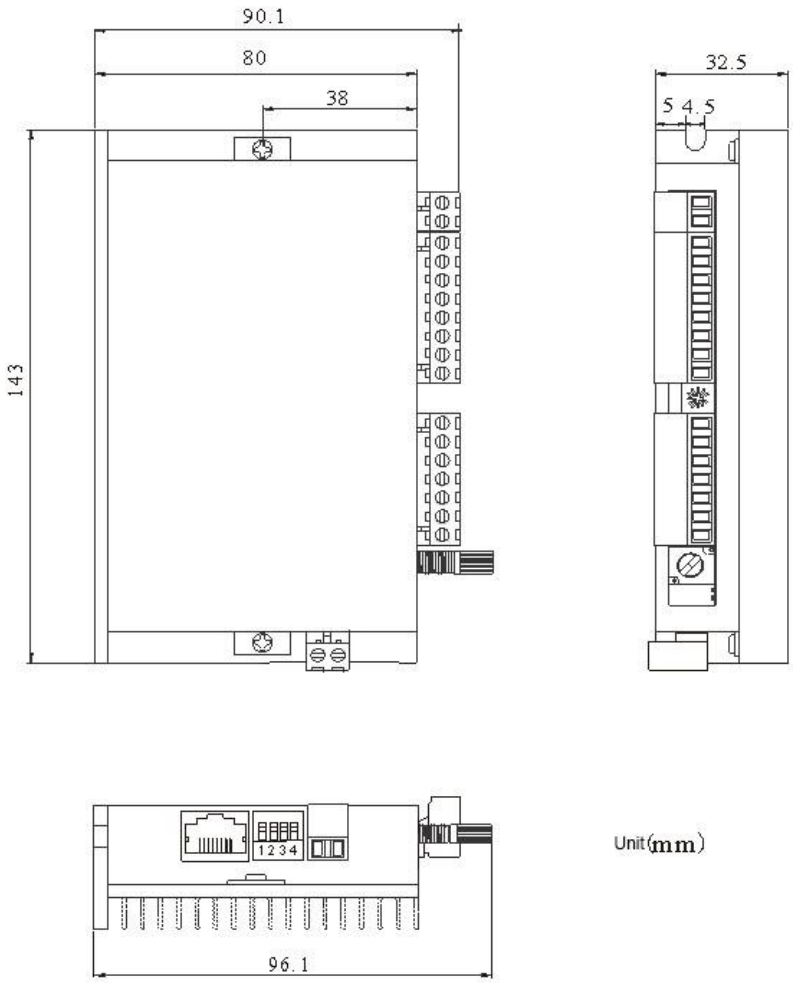
Parameter	Min.	Typical	Max
Input Voltage DC (V)	15	48	62
Peak Current (A)	2	-	20.6
Rotate Speed(rpm)	60[1]	-	-

[1] **Under closed loop, the minimum speed of the motor is 100 only if the pole number is set on the driver correctly.**

2.2 Environmental indicators

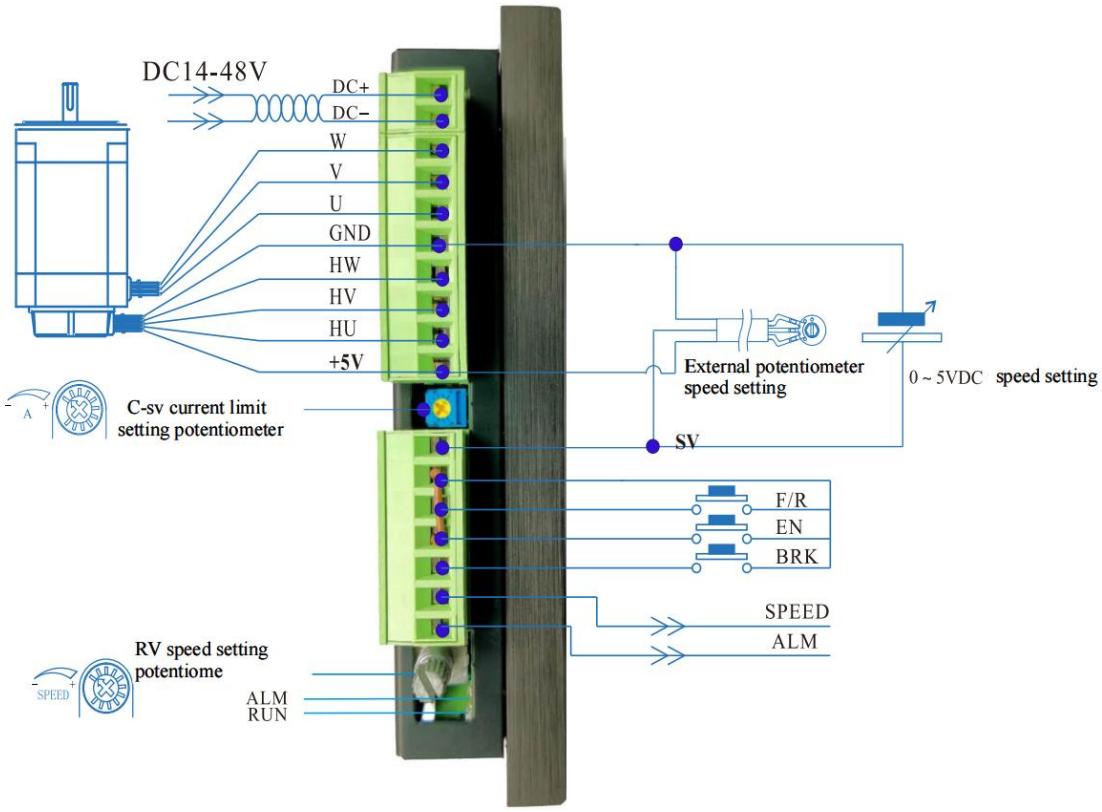
Heat Sinking Method	Natural cooling or forced cooling
Occasions	avoid dust, oil and corrosive gases
Operating Temperature	-20℃~+60℃
Storage Temperature	-30℃~+70℃

3. Dimension (Unit: mm)



4. Interface and wiring diagram

4.1 Driver interface



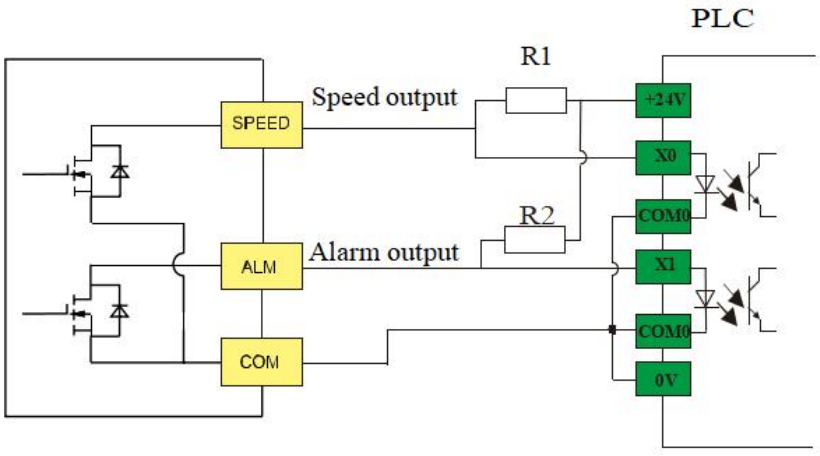
4.2 Port signal description

Signal	Terminal	Function
Power Connection	DC+	Power supply positive
	DC-	Power supply negative
Motor Connection	W	Motor line W phase
	V	Motor line V phase
	U	Motor line U phase
Hall Signal	GND	Hall sensor signal ground electrode
	HW	Hall sensor signal HW
	HV	Hall sensor signal HV
	HU	Hall sensor signal HU
	+5V	Hall sensor signal power supply+
Control Signal	SV	① External potentiometer ② External analog signal input
	COM	Common port
	F/R	F/R disconnected with COM, motor rotates CW, otherwise motor rotates CCW
	EN	EN and COM disconnected, motor stops slowly, motor runs.
	BRK	BRK and COM disconnect, the motor runs, otherwise motor brakes to stop
Output Signal	SPEED	Refer to 4.3. According to the motor running speed, the pulse signal of the corresponding frequency is output, and the speed of the motor can be calculated using this frequency. The formula is: N(rpm)= (F/P)×10 F: output pulse frequency (Hz); P: motor pole pair number; N: motor speed Example: Motor with 4 pairs of poles F=500HZ N(RPM)=(500/4)×10=1250
	ALM	Refer to 4.3. Normally high-impedance output, low-level output when a fault occurs

4.3 ALM, SPEED, and PLC Wiring

The internal structure of ALM and SPEED of this drive is as follows, which is related to the PLC wiring and the type of PLC input. For example, input modules such as AX40/741/42/50/60 of Mitsubishi A series PLC and QX40/41/42 of Q series only support source mode, and users should connect according to the characteristics of this input type.

Output interface circuit of brushless driver



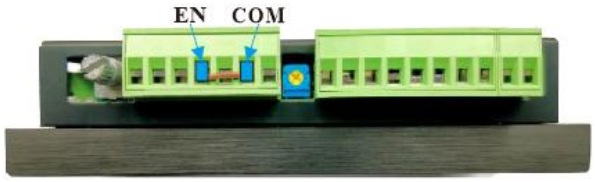
Notice: R1,R2=1KΩ 12V
R1,R2=2KΩ 24V

5. Function setting

5.1 Start and stop

When EN and COM terminal are connected, the motor runs; Otherwise, the motor stops slowly.

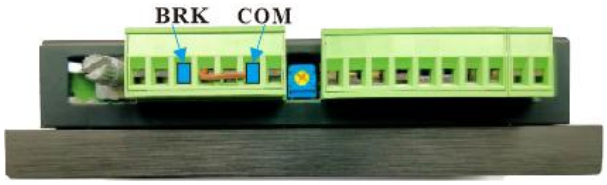
By connecting a switch between COM and EN or using PLC can control the motor start and stop.



5.2 Brake

When connecting the BRK and COM, the motor stops quickly. Otherwise, the motor runs.

By connecting a switch between COM and BRK or using PLC could control run and rapid stop.



The difference between EN and BRK:

EN control is natural stop; BRK control is rapid stop. For example, it takes 592ms for the motor to completely stop after EN is disconnected, and it only takes 12ms after BRK is connected.

5.3 Direction control

F/R and COM disconnect in default, when power is on, motor will start to run CW.

F/R and COM disconnect, motor runs clockwise.

F/R and COM connect, motor runs anticlockwise.

5.4 Closed loop work

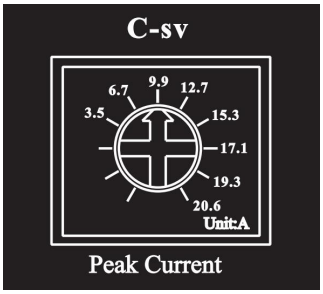
The driver always runs in the speed closed loop state.

5.5 Current limit value adjustment

Current limit value adjustment(C-sv) can limit the peak value of the motor phase current when the load suddenly increases. The output current will be limited to the set value to reduce the motor speed and protect the motor not damaged.

The setting range is 3.5 ~ 20.6A.

Set the peak current according to the scale in the right figure.



Since the error between the set peak output current and the actual peak output current is about ±10%. For safety, please appropriately reduce the peak output current.

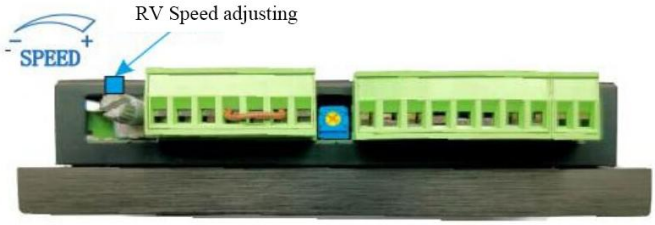
5.6 Stalling torque holding

When the load suddenly rises or hits an obstacle, the output current of the motor is limited to the set value of the current-limiting potentiometer to protect the motor and the driver. Due to the effective holding force, the force drop and unreasonable collision are prevented. If the reason for the restriction of the rotor is solved within 10 seconds, the motor can rotate again. If not, the motor will stop and report 8 alarms, and should reset the motor (User cannot be restarted by removing obstacles alone).

6. Speed setting methods and settings

6.1 Via built-in potentiometer

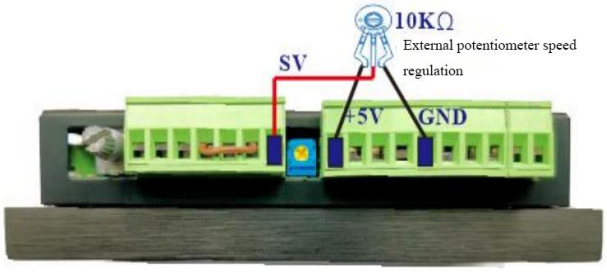
Motor speed is increase when RV knobs is rotated clockwise, when anticlockwise, motor speed decreases.



⚠ If user use other speed modes, RV should be rotated anticlockwise to limit position.

6.2 Via external potentiometer

Use a suitable potentiometer with a resistance value of 10kΩ; When connect external potentiometer, the middle terminal connects to SV, the other two terminals connect to +5V and GND

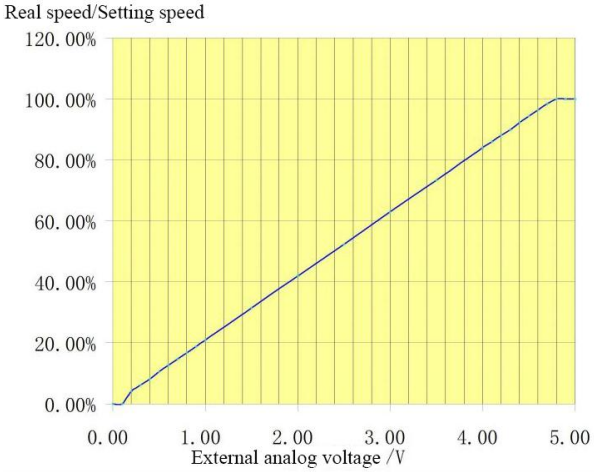


⚠
① RV should be rotated anticlockwise to limit position.
② Notice the connection sequence of the potentiometer.

6.3 Via external analog signal 0-5V

When use external analog signal, the built-in potentiometer RV must be OFF (RV should be rotated anticlockwise to limit position).

External analog speed control linearity test curve:



7. Status indicator & trouble shooting

7.1 Status indicator

When appearing locked-rotor, Hall fault, over-temperature and over voltage, driver will give an alarm signal and stop working.

Alarm indicator	Status description
Red light flashes twice	Driver components over temperature (over 85℃)
Red light flashes 3 times	Over voltage
Red light flashes 4 times	Under voltage
Red light flashes 5 times	Hall fault
Red light flashes 8 times	Motor locked-rotor or driving circuit abnormal
Red light flashes 21-22 times	I2C communication physical layer error
Red light flashes 23 times	The memory chip cannot write
Red light flashes 24 times	The memory chip cannot read
Red light flashes 30-42 times	Peripheral circuit of master control chip abnormal

7.2 Trouble shooting

When any of abnormal indicator in above table occurs, a reset command should be issued to driver to clear the alarm signal off.

*The reset command refers to one of the following commands :

- Rotate built-in potentiometer anticlockwise to limit position.
- Adjust external analog signal to 0.
- Open EN with COM, then reconnect.
- Open BRK with COM, then Re-connect.
- Turn off power and restart.

If the alarm signal still exists, then proceed as below table:

LED display	Issue handling
Red light flashes twice	Power on again after the driver cools down
Red light flashes 3 times	Check to make sure power supply voltage is normal
Red light flashes 4 times	Check to make sure power supply voltage is normal
Red light flashes 5 times	Check motor connecting wire is stable and ensure motor is not damaged
Red light flashes 8 times	Check if the load is too heavy and the motor is damaged, if not, please change another same type driver to test
Red light flashes 21-22 times	Depot repair
Red light flashes 23 times	Depot repair
Red light flashes 24 times	Depot repair
Red light flashes 30-42 times	Depot repair

7.3 Alarm signal output

If any of alarm in 6.2 table happens, ALM terminal would output low level; if driver is normal, ALM would output high level.

8. Modbus Communication

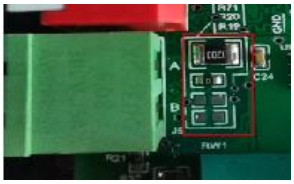
8.1 DIP switch function

The function of SW1~SW3 is to set the slave address. The setting method is shown in the following table. SW4 is to connect a terminal resistance of 120Ω in parallel at the 485 terminal, and it means to disconnect the terminal resistance in the OFF state.

SW3	SW2	SW1	Server Address
0	0	0	Broadcast communication mode which can only write data
0	0	1	0x01
0	1	0	0x02
0	1	1	0x03
1	0	0	0x04
1	0	1	0x05
1	1	0	0x06
1	1	1	0x07

If users need to expand the address range, users can change the hardware as follows:

- Remove R26, R71
- R19, R20 welded with 0Ω resistance
- Connect the terminal resistance 120Ω to the plug-in terminal according to the site conditions



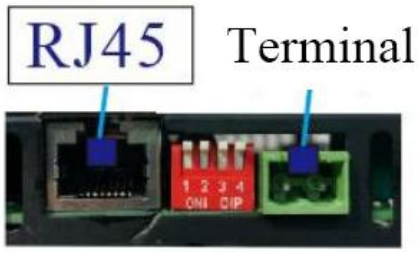
Changed address:

SW4	SW3	SW2	SW1	Server Address
0	0	0	0	Broadcast communication mode which can only write data
0	0	0	1	0x01
0	0	1	0	0x02
0	0	1	1	0x03
0	1	0	0	0x04
0	1	0	1	0x05
0	1	1	0	0x06
0	1	1	1	0x07
1	0	0	0	0x08
1	0	0	1	0x09
1	0	1	0	0x0A
1	0	1	1	0x0B
1	1	0	0	0x0C
1	1	0	1	0x0D
1	1	1	0	0x0E
1	1	1	1	0x0F

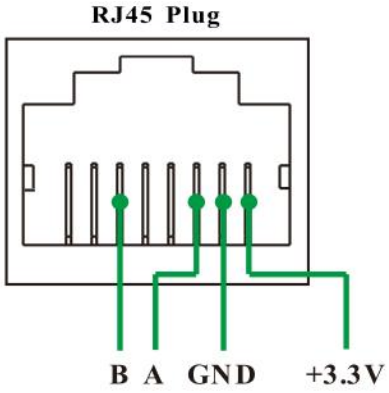
8.2 Communication interface and wiring method

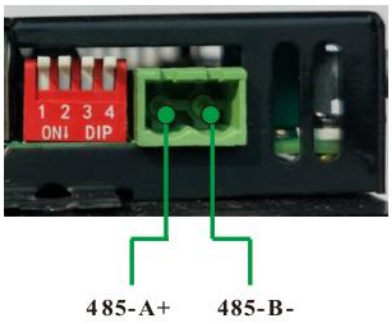
This drive has 2 kinds of interfaces for wiring,

one is RJ45, the other is pluggable terminal



The "2-wire" electrical interface in EIA / tia-485 standard is adopted in the BLR400A,RJ45 interface require: A、B、GND connecting. The detail please take the Section 7.3.2 in GB / T 19582.2-2008 for reference. Another connecting mode is just use for debugging the customer programming. Considering the industry practice, there's 3.3V voltage on the RJ45Plug, Users can decide whether to connect 3.3V or not according to the actual situation.





8.3 Address of all holding registers of the server

The register address on the server read by client:

Server address (1byte)	Function code (1byte)	Holding register access initial address (2byte)	Access data size (2byte)	CRC check (2byte)	Function description
0xnn	0x03	0x0056	0x0001	CRC check	Read the set speed of the driver (unit: rpm)
0xnn	0x03	0x005F	0x0001	CRC check	Read motor feedback speed (unit: rpm)
0xnn	0x03	0x0066	0x0001	CRC check	Read driver whether enable, 0=enable state 1=disable state
0xnn	0x03	0x006A	0x0001	CRC check	Read driver whether brake, 0=brake 1=not brake
0xnn	0x03	0x006D	0x0001	CRC check	Read driver rotate direction, 1=forward rotation 0=reverse rotation
0xnn	0x03	0x0076	0x0001	CRC check	Read the alarm code, refer to section 7. of this manual. For examples: 2=NTC1 detects that the PCB temperature over 85 ℃ 3=voltage over 63V
0xnn	0x03	0x0086	0x0001	CRC check	Read the set number of poles
0xnn	0x03	0x008A	0x0001	CRC check	Read acceleration time setting value(unit:100ms)
0xnn	0x03	0x0092	0x0001	CRC check	Read the maximum speed of analog speed regulation (unit: rpm)
0xnn	0x03	0x0096	0x0002	CRC check	Read the current limit value (unit: A), and note that the current limit value is a floating-point number, with the high 16 bits in 0x0096 and the low 16 bits in 0x0097
0xnn	0x03	0x00B6	0x0001	CRC check	Read RS-485 connection status
0xnn	0x03	0x00BB	0x0001	CRC check	The driver program version is not available to users
0xnn	0x03	0x00C8	0x0002	CRC check	Read the value of power supply voltage (unit: V). Note that the voltage is a floating-point number. The high 16 bits are placed in 0x00C8, and the low 16 bits are placed in 0x00C9
0xnn	0x03	0x00D2	0x0002	CRC check	Read the temperature value (unit: ℃). Note that the temperature is a floating-point number. The high 16 bits are placed in 0x00D2, and the low 16 bits are placed in 0x00D3

The register address on the server written by client:

Server address (1byte)	Function code (1byte)	Holding register access initial address (2byte)	Access data size (2byte)	CRC check (2byte)	Function description
0xnn	0x06	0x00B6	-	CRC check	Write RS-485 connected state (the function of setting this parameter is deciding to follow which one when the driver receives both speed setting signals : built-in signal and signal from client); Write 0=485 control disable, client can only read holding register, but cannot write other holding register except 0x00B6; Write 1=485 control enable, client can read and write register.
0xnn	0x06	0x0040	-	CRC check	Post-back function setting, which would affect communication rate, but be convenient to adjust machine. If write 0, then data would be post back. For an example: 0x0040 wrote 0, every time the PLC sending 06 function code command to driver, the driver would execute and post back data at once. 0x0040 write 1, driver would only execute but not post back.
0xnn	0x06	0x0056	-	CRC check	Set the motor speed (unit: rpm), the value range of this register is 0~60000
0xnn	0x06	0x0066	-	CRC check	Write 0: enable state; write 1: disable state
0xnn	0x06	0x006A	-	CRC check	Write 0: brake state; write 1: not brake state
0xnn	0x06	0x006D	-	CRC check	Write 1: forward rotation; write 0: reverse rotation
0xnn	0x06	0x0076	-	CRC check	Write non-zero, register keep the original value; Write 0: reset alarm
0xnn	0x06	0x0086	-	CRC check	Motor pole number setting, the value range of this register is 0~255, writing other values is invalid
0xnn	0x06	0x008A	-	CRC check	Acceleration time setting (unit: 100ms; range: 0~150) For an example: the time is set as 12, the given rotate speed of driver accelerating from 0 to 3000 needs 12s. Notice : the actual rotate speed and given speed are out of sync.

0xnn	0x06	0x0092	-	CRC check	The maximum speed of analog speed regulation (unit: rpm), the value range of this register is 0~20000, writing other values is invalid. This parameter is defined as the actual speed of the motor when SV pin inputs 5V. If SV pin inputs 1V, then the speed of the motor is the value of this parameter / 5. In addition, this parameter has no effect on the speed range of the client (touch screen and other devices)
0xnn	0x06	0x00BC	-	CRC check	Write 1: save each setting parameter once
0xnn	0x06	0x00CC	-	CRC check	Restore factory settings Write 1: restore factory settings. The specific operation is to restore all parameters (excluding restoring factory settings) related to writing in this table to default values, and then restore all saved values to default values. For example, if the current saved value of motor pole number (0x0086) is 2, it is restored to default value 4. After the recovery, the 0x06 hold register is reset to 0 Write 0: do not restore factory settings or restore factory settings successfully

In addition to the address of the register, the user also needs to know how the drive stores data and calls data. The following table shows the relevant detail：

Holding register access initial address (2byte)	Function description	Whether to save the value in the memory chip after writing 1 to register 0X00BC	When B6 holding register = 1, whether to call the value stored in the memory chip to set the server	When B6 holding register = 0, whether to call the value stored in the memory chip to set the serve	Default value
0x0040	Set whether posting back data	Yes	No	No	0
0x0056	Set the motor rotate speed (unit: rpm), the value range : 0~20000	No	No	No	0
0x0066	Write 1: enable; write 0: disable	No	No	No	1
0x006A	Write 1: brake; write 0: not brake	No	No	No	0
0x006D	Write 0: forward rotation; write 1: reverse rotation	Yes	Yes	No	0
0x0076	Write non-zero, register keep the original value; Write 0: the value of register =1, reset alarm	No	No	No	1

0x0086	Motor pole number setting, the value range of this register is 0~255, writing other values is invalid	Yes	Yes	Yes	4
0x008A	Acceleration time setting (unit: 100ms; value range: 0~150), writing other values is invalid.	Yes	Yes	Yes	0
0x0092	The value range of this register is 0~20000. The definition of this parameter is the actual speed of the motor when SV pin inputs 5V. If SV pin inputs 1V, then the speed of the motor is the value of this parameter / 5. In addition, this parameter has no effect on the speed range of the client (touch screen and other devices)	Yes	Yes	Yes	4000
0x00B6	Write RS-485 connection status (the setting function of this parameter is to decide which one to listen to when the driver receives the internal speed control signal and the speed control signal from the client at the same time) Write 0 = 485 Control disable, at this time, the client can only read the holding register, cannot write other holding registers except for the address of 0x00b6 Write 1 = 485 Control enable. At this time, the client can read and write registers	No	No	No	0
0x00CC	Write 1: restore factory settings.	No	No	No	0

8.4 Communication steps for touch screen

User should set the touch screen according to the following parameters before communication

- Baud rate：9600
- Transmission mode: adopt RTU mode, and data mode is 8-bit asynchronous serial, 1 stop bit, no check bit
- Error check mode：CRC-16/Modbus

According to the register address, the corresponding connection channel is added to the touch screen program. The specific process is determined by the type of touch screen selected by the user.

8.5 Communication steps for PC and other devices

This part is applicable to the case that the client is a PC and other devices. If the user uses the touch screen to communicate with the driver, please ignore the below information.

Before communication, users need to have a certain understanding of the following two standards

- GB/T 19582.1-2008：《Industrial automation network specification based on Modbus Protocol Part 1: Modbus Application Protocol》
- GB/T 19582.2-2008：《Industrial automation network specification based on Modbus Protocol Part 2: Modbus Protocol Implementation Guide on serial link》

When communicating, user needs to write the program according to following steps:

① Determine the address of the drive

② According to the baud rate to send with a start bit, two stop bits of a single byte program

③ Write CRC verification program

④ The structure of ADU is determined according to the function, and the ADU is sent

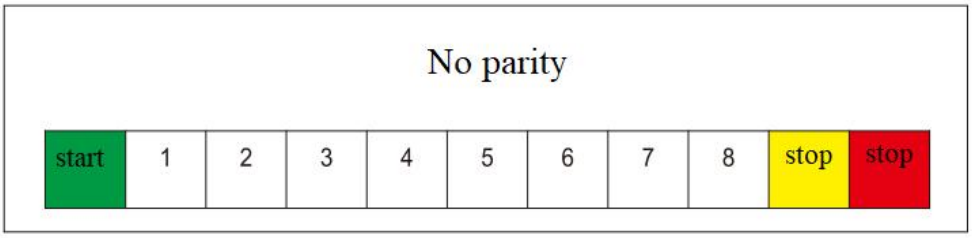
⑤ Analyze the data according to the ADU returned by the drive

The address of driver could refer to section 9.3 of this manual, the detailed description for ②~⑤ is followed as below:

● Write a single byte program with 1 start bit and 2 stop bits according to the baud rate

About the details of this portion, users can refer to part 6.5.1 of national standard GB/T 19582.2-2008.

*Notice: each character or byte is sent from left to right:（LSB）…（MSB）



In above picture, the starting signal of a single data packet for serial communication is represented by logical 0 data bit, and the two stopping signals of data packet are represented by logical 1 data bit.

● Write CRC verification program

The driver adopts CRC-16/Modbus algorithm model. Users can refer to Appendix B.2 of GB/T 19582.2-2008 for more details about this calibration method. Below is C source program for generating verified code which is for user's reference.

```
typedef unsigned char u8;
typedef unsigned int u16;
/**
  *@brief    generating check code
  *@param *ptr    array for storing information code, the first address of array is placed the first byte of ADU
  *@param length    bytes of check code removed by ADU
  *@retval u16   check code
  */
u16 getCRC16(u8 *ptr, u8 length)
{
   u8 i;
   u16 crc = 0xFFFF;

   if(length == 0)
      length = 1;
      while(length--) {
         crc ^= *ptr;
         for(i = 0; i < 8; i++)  {
            if(crc & 1)  {
               crc >>= 1;
               crc ^= 0xA001;
            }
            else
               crc >>= 1;
         }
         ptr++;
      }
   return(crc);
}
```

● Determine the structure of ADU according to function and send the ADU

This drive uses two function codes: 03 and 06, so users need to be familiar with two formats of ADU only. Request ADU and response ADU of 03 function code can refer to section 7.3 of GB/T 19582.1-2008. Request ADU and response ADU of 06 function code can refer to section 7.6 of GB/T 19582.1-2008. The common ADU is listed in below table for user's reference.

Below example shows the debugging of the serial port assistant when register 0X40 value is 0, and parameters will be posted back. (if no need to post back data, then write 1 to register 0X40)

Read register	
Read motor feedback speed	Sent：**01 03 00 5F 00 01 B4 18** Receive：**01 03 02 02 48 B9 12** **(The motor speed is 584 rpm)**
Read alarm code	Sent：**01 03 00 76 00 01 65 D0** Receive：**01 03 02 00 00 B8 44** **(The alarm code is 0)**
Read pole pairs	Sent：**01 03 00 86 00 01 65 E3** Receive：**01 03 02 00 04 B9 87** **（ pole pair is 4）**
Read acceleration time	Sent：**01 03 00 8A 00 01 A5 E0** Receive：**01 03 02 00 00 B8 44** **(The acceleration time is 0)**
Read the maximum speed of analog speed regulation	Sent：**01 03 00 92 00 01 25 E7** Receive：**01 03 02 0B B8 BF 06** **（The maximum speed of analog speed regulation is 3000rpm）**
Read current limiting value	Sent：**01 03 00 96 00 02 24 27** Receive：**01 03 04 41 9F 39 5B 8D 8A** **（current limiting value is 19.9A）**
Unicast mode write register	
Write set speed 1100rpm	Sent：**01 06 00 56 04 4C 6A EF** (When 485 control is enabled) receive:**01 06 00 56 04 4C 6A EF** (When 485 control is disabled) receive:**01 86 FF 02 20**
Write enable	Sent：**01 06 00 66 00 01 A8 15** (when 485 control is enabled) receive：**01 06 00 66 00 01 A8 15** (when 485 control is disabled) receive：**01 86 FF 02 20**
Write forward	Sent：**01 06 00 6D 00 01 D9 D7** (when 485 control is enabled) receive：**01 06 00 6D 00 01 D9 D7** (when 485 control is disabled) receive：**01 86 FF 02 20**
Write pole pair 2	Sent：**01 06 00 86 00 02 E9 E2** (when 485 control is enabled) receive：**01 06 00 86 00 02 E9 E2** (when 485 control is disabled) receive：**01 86 FF 02 20**
Write acceleration time 9 seconds	Sent：**01 06 00 8A 00 5A 28 1B** (when 485 control is enabled) receive：**01 06 00 8A 00 5A 28 1B** (when 485 control is disabled) receive：**01 86 FF 02 20**
Write the maximum speed of analog speed regulation 500rpm	Sent：**01 06 00 92 01 F4 28 30** (when 485 control is enabled) receive：**01 06 00 92 01 F4 28 30** (when 485 control is disabled) receive：**01 86 FF 02 20**

Write 485 Control enable	Sent：**01 06 00 B6 00 01 A9 EC** Receive：**01 06 00 B6 00 01 A9 EC**
Broadcast mode write register	
Write pole pair 2	Sent：**00 06 00 86 00 02 E8 33** Receive：No response
Write set speed 1100rpm	Sent：**00 06 00 56 04 4C 6B 3E** Receive：No response
Write 485 Control enable	Sent：**00 06 00 B6 00 01 A8 3D** Receive：No response

● Parse data according to the ADU returned by driver

Users should perform CRC verification on the returned ADU firstly, then parse the data. Data in ADU can be stored in two ways: one is stored in 16 bits register (the data is a 16 bit short integer at this time), the other is to put data in two registers with consecutive addresses (the data is a 32-bit single precision floating point number at this time). Due to limited space, this manual only gives the C source program for reading data in the second case as follows :

```
/**
* @brief        Get floating point number
* @param  Address     8-bit character array address, the first byte of ADU put in the first address
* @retval float   floating point number which can be read
*/
float getFloat(unsigned char *Address)
{
    unsigned char
    *floatAddress=Address+3; int
    temp=0;
    temp+=(int)(*(floatAddress))<<24;
    temp+=(int)(*(floatAddress+1))<<1
    6;
    temp+=(int)(*(floatAddress+2))<<8;
    temp+=(int)(*(floatAddress+3));
    return *(float *)&temp;
}
```