IR42, IT42 Integrated Stepper Motor User Manual

Shenzhen Rtelligent Technology Co., Ltd

www.rtelligentglobal.com

Revision History

Date	Version	Description
2024.09.11	V4.0	Version 4 product update

1. Feature

- Driver, control, motor integrated intelligent stepper motor.
- Working voltage: 12~24VDC
- Pulse control mode: pulse + direction, double pulse, orthogonal pulse
- Communication control mode: RS485 / Modbus RTU
- Communication setting:
  - 5-bit DIP switches: 31 axes address
  - 2-bit DIP switched: 4 speed baud rate
- Running direction setting: 1-bit DIP switch to set the motor running direction
- Maximum phase current output: 1.5A/phase (sinusoidal peak)
- Digital IO port:
  - 3 photoelectric isolated input digital signal input: 5V or 24V single-ended input, common anode connection method.

Note: Signal level 5V and 24V have different product models, please refer to the naming rules for details.

2. Naming Rule

IT	42	AM	06	-	D	08	24	-	5V	-	M
①	②	③	④		⑤	⑥	⑦		⑧		⑨

No.	Part Name	Description
①	Series name	IR: Integral open loop series IT: Integral closed loop series
②	Motor base	Base: 20, 28, 42, 57, 60, 86 Unit: mm
③	Motor version	AM, CM version
④	Motor torque	06: 0.6NM
⑤	Motor shaft type	D: Single flat K: Keyed G: Plain shaft
⑥	Shaft diameter	Unit: mm
⑦	Shaft length	Unit: mm
⑧	IO signal level	5V、24V
⑨	Communication interface	M: Modbus - RS485 C: Can – CanOpen E: EtherCAT

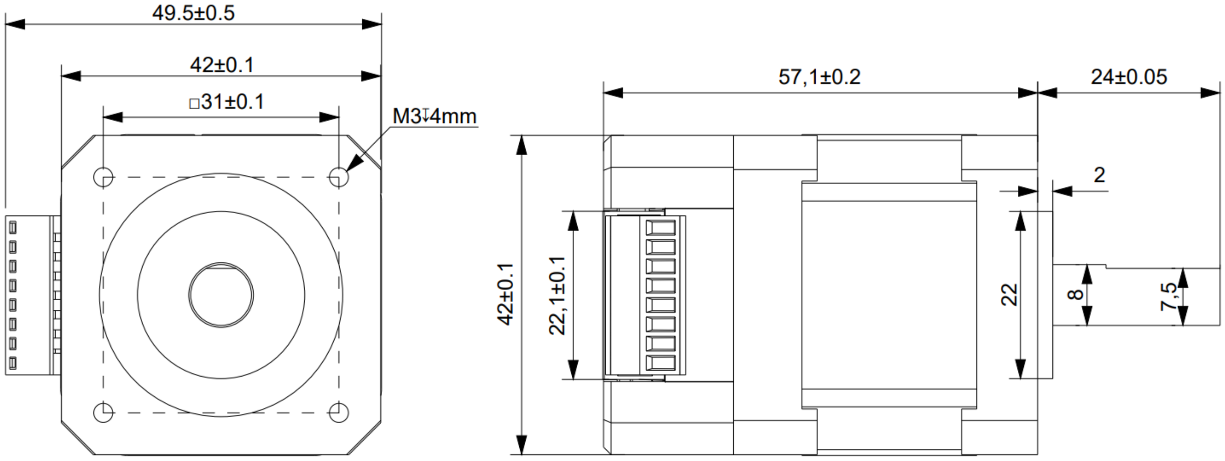
3. Installation

3.1. Installation Requirement

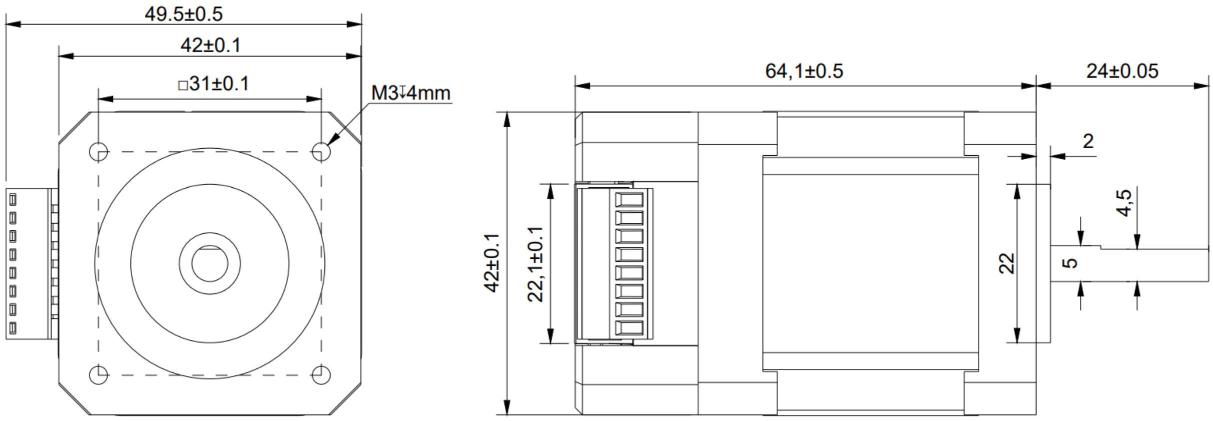
The integrated stepper motor contains a precision position detection device and excessive axial hammering of the motor shaft should be avoided.

3.2. Installation Dimension

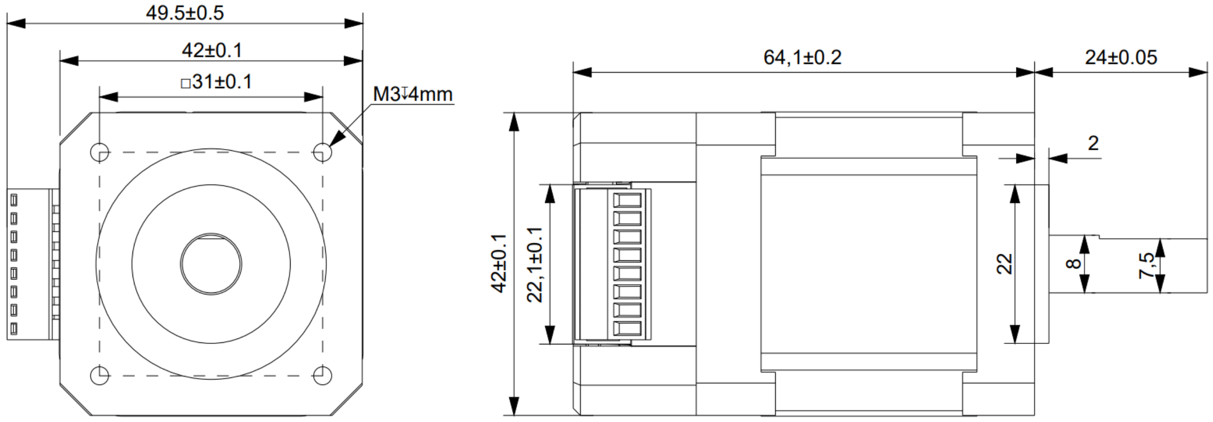
3.2.1. Ix42AM04-D0824-xxV-x



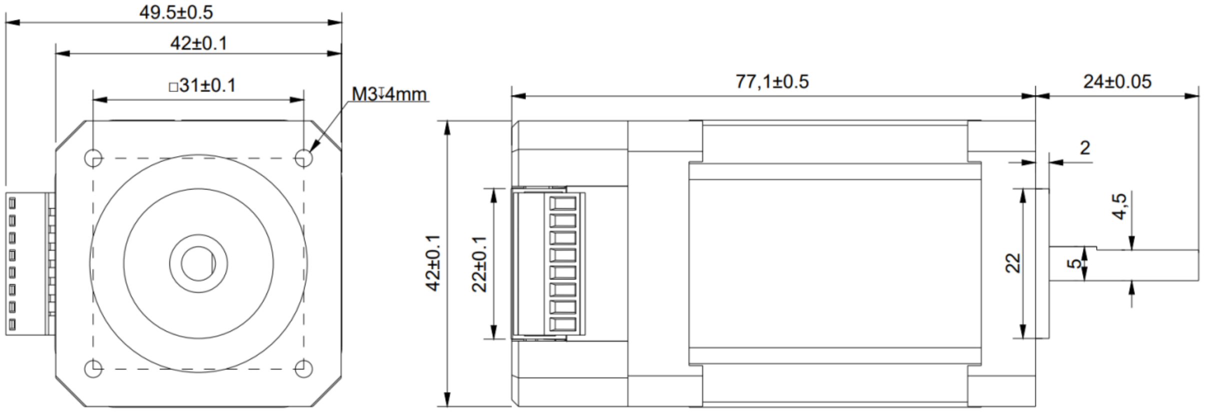
3.2.2. Ix42AM06-D0524-xxV-x



3.2.3. Ix42AM06-D0824-xxV-x

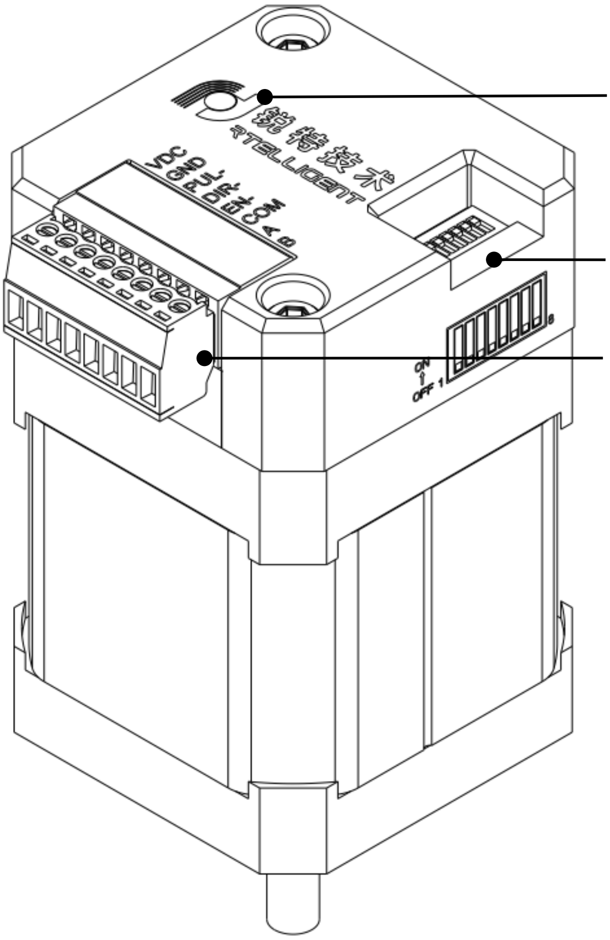


3.2.4. Ix42AM08-D0524-xxV-x



4. Port, Wiring & Setting

4.1. Port Function Description



① Indicator

② DIP switches

③ Power supply,
Control signal,
RS485 communication

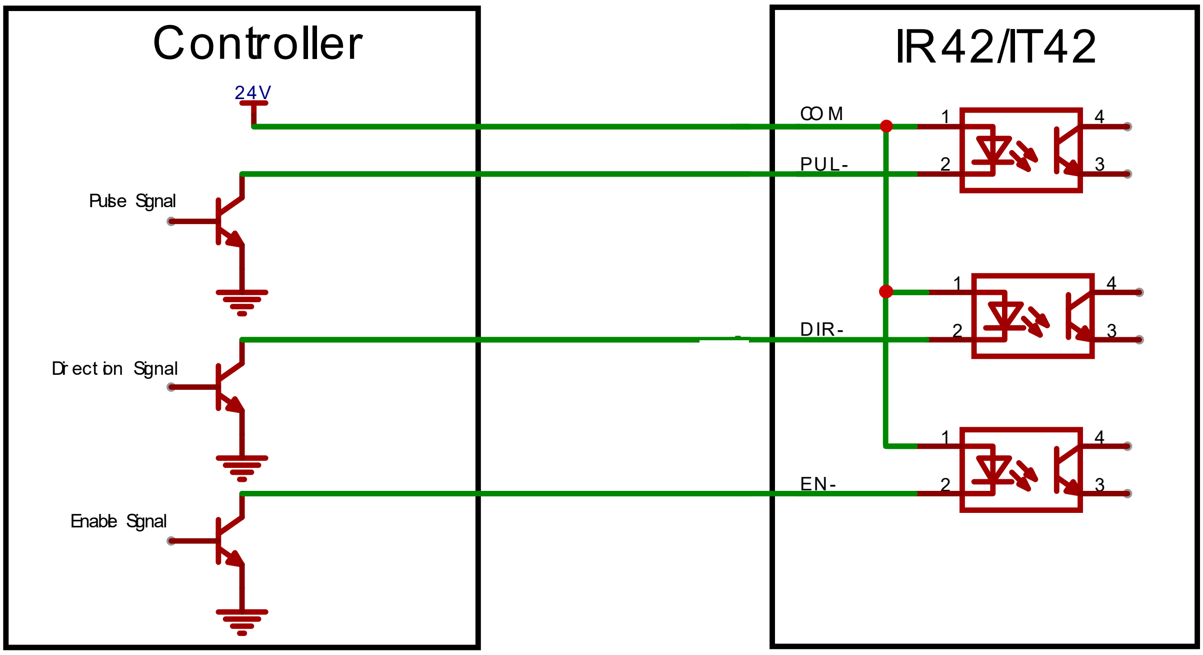
Number	Part Name	Description
①	Status indicator	LED lights are used to display the operating status and fault status of the driver
②	DIP switches	Used to set slave station address, baud rate and motor running direction
③	Power supply, control signal, RS485 communication interface	Power supply, pulse, direction, enable signal and RS485 communication signal input interface

4.2. Power Supply, IO & RS485 Signal

Mark	Name	Description
VDC	Power supply positive	The working voltage range of the product is 12~24VDC, the polarity of the power supply must not be reversed and the voltage must not exceed 28V, otherwise the product will be damaged.
GND	Power supply negative	
PUL-	Pulse signal interface	Users need to know clearly whether the signal level is 5V or 24V, otherwise the IO ports may be damaged. ◆ Note: The signal level of 5V and 24V are different, please refer to the naming rules for details.
DIR-	Direction signal interface	
EN-	Enable signal interface	
COM	Common anode input common terminal	
A	RS485 A	RS485 communication interface, non-isolated, with GND as reference ground.
B	RS485 B	

4.2.1. IO Signal NPN single-ended 24V Common Anode Connection

The IO signals of IR42/IT42 series products only support single-ended common anode connection, and the wiring diagram is as follows:



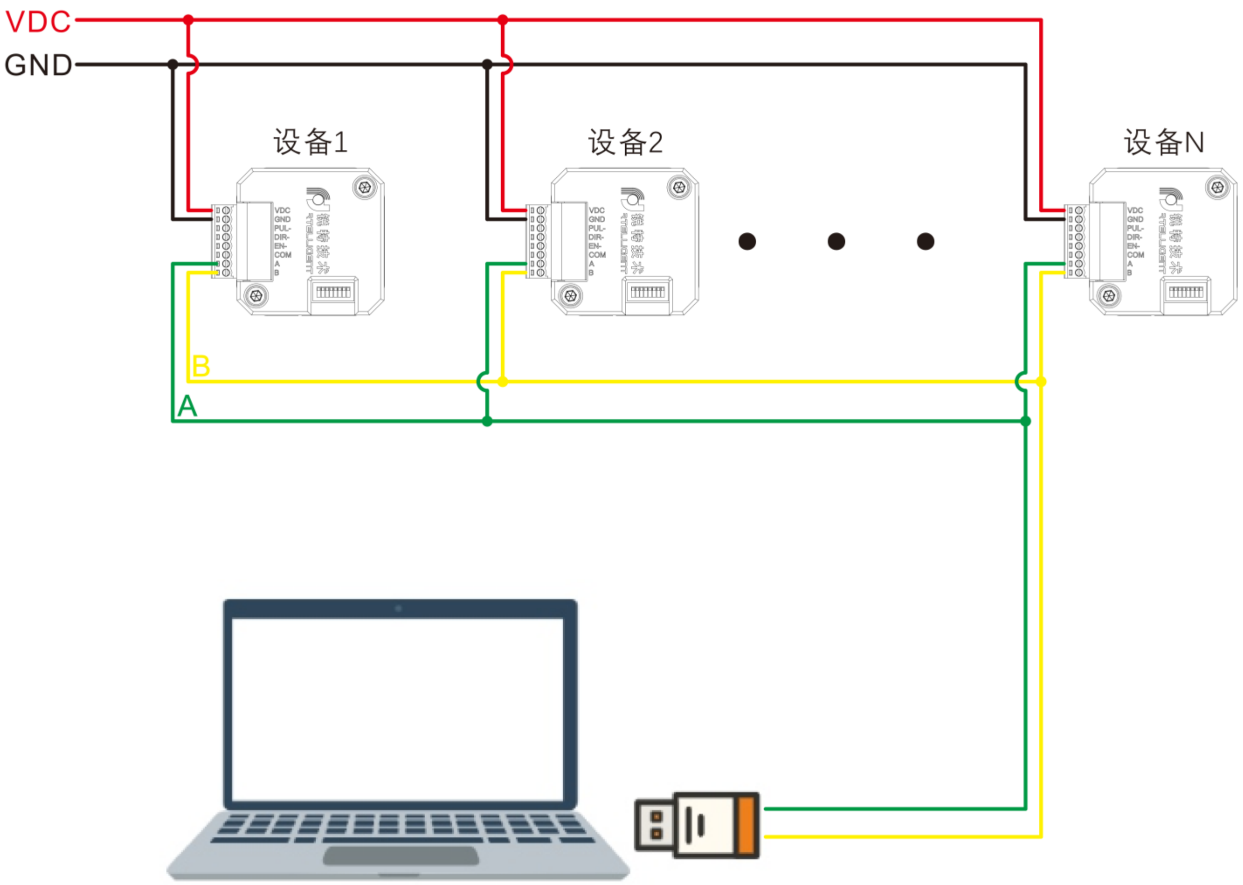
4.3. RS485 Communication DIP Switches Setting

4.3.1. Communication Format

- Baud rate: same as slave
- Data bit: 8-bit
- Stop bit: 1-bit
- Check bit: no check bit

4.3.2. RS485 Network Connection

It is recommended to use the daisy chain form polarity RS485 network connection as follows:



4.3.3. Termination Matching Resistor

The end can choose whether to install 120 ohm termination resistor or not depending on the situation, usually not needed for short distance.

4.3.4. Slave Address [SW1~SW5]

Each slave in the same network has a unique address.

Slave ID	SW1	SW2	SW3	SW4	SW5
Default	ON	ON	ON	ON	ON
1	OFF	ON	ON	ON	ON
2	ON	OFF	ON	ON	ON
3	OFF	OFF	ON	ON	ON
4	ON	ON	OFF	ON	ON
……	……	……	……	……	……
30	ON	OFF	OFF	OFF	OFF
31	OFF	OFF	OFF	OFF	OFF

ON = 0，OFF = 1

Slave address = SW1 + SW2 × 2 + SW3 × 4 + SW4 × 8 + SW5 × 16

4.3.5. Baud Rate [SW6~SW7]

Master and slave must be set to the same baud rate.

BDR	SW6	SW7
9600	ON	ON
19200	OFF	ON
38400	ON	OFF
115200	OFF	OFF

4.3.6. Direction Setting [SW8]

The direction of motor operation can be changed by setting the ON or OFF state of SW8 without changing the driver related settings. The factory default is OFF.

4.4. Status and Alarm Indicators

- The integrated motor contains an integrated red and green indicator for status and errors.
- When the motor is in the enabled state, the green indicator flashes slowly, when the green light is on for a long time the motor is non-enabled.
- If the red LED flashes, an error occurs, the error is indicated by a combination of red and green flashing lights as follows:

LED status		Driver status
🟢	Steady green light	Driver is not enabled
🟢 🟢	Flashing green light	Driver works fine
🟢 🔴	1 green, 1 red	Driver overcurrent
🟢 🔴 🔴	1 green, 2 red	Driver input power supply overvoltage
🟢 🔴 🔴 🔴	1 green, 3 red	Driver internal voltage error
🟢 🔴 🔴 🔴 🔴	1 green, 4 red	Encoder out of tolerance alarm
🟢 🔴 🔴 🔴 🔴 🔴 🔴	1 green, 6 red	Parameter check error
🟢 🔴 🔴 🔴 🔴 🔴 🔴 🔴	1 green, 7 red	Motor phase loss

5. Communication Protocol

5.1. Modbus/RTU Configuration

◆ The serial communication parameters are as follows:

(1) Baud rate: same as slave
(2) Data bits: 8-bit
(3) Stop bit: 1-bit
(4) Check bit: None

◆ IT/IR series products currently support the following Modbus function codes:

(1) 0x03: Read holding register
(2) 0x06: Write single register
(3) 0x10: Write multiple registers

◆ MODBUS registers start with 0, while in touch screen and PLC, register addresses are typically represented as 400x type, starting with 1.

So: PLC address=MODBUS address + 1

◆ Register operation type:

(1) R: Rad only
(2) W: Write only
(3) R/W: Read or write

◆ Data type:

(1) SHORT —— 16bit
(2) LONG —— 32bit

◆ MODBUS defaults to a 16 bit register. Two consecutive registers form a 32-bit data, with low 16-bit data at the beginning and high 1-bit data at the end.

5.2. Register Explanation

5.2.1. Driver Flag Register [0～1]

5.2.1.1. Alarm Flag Register [0]

Defined all alarm flags for the driver. MODBUS address: 0

15				11	10	9	8
Reserve							ECDE1
	R-0				R-0		

7	6	5	4	3	2	1	0
POSE	MPE	MEM	OT	UV	OV	OC	IVE
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

BIT	Name	Description
9～15	Reserve	**Read always returns 0**
8	ECDE1	**Encoder failure** 0: Encoder signal is normal 1: Encoder signal is abnormal
7	POSE	**Tracking error alarm** No tracking error alarm 1: Tracking error alarm occurs, the motor cannot follow the encoder normally ◆ The possible impacts of a tracking error alarm are as follows: ① Position deviation alarm threshold ② Encoder cable ③ Motor power cable ④ Check whether the setting of parameters such as speed and acceleration is reasonable
6	MPE	**Motor phase loss alarm** No alarm 1: Alarm occurs ◆ When a phase loss alarm occurs, the driver cannot detect the current of the motor winding normally ① Need to check motor wiring and motor type
5	MEM	**Parameter check error** 0: Parameter check is correct 1: Parameter check error

4	OT	**Over temperature alarm sign** 0: Driver temperature is normal 1: The temperature of the internal components of the driver is too high
3	UV	**Undervoltage alarm flag** 0: No undervoltage alarm 1: Driver experiencing undervoltage
2	OV	**Overvoltage alarm sign** 0: No overvoltage alarm 1: Driver overvoltage occurs ◆ Drive overvoltage occurs, check the following: ① Check the input power supply ② Check the pump voltage during motor deceleration
1	OC	**Overcurrent alarm flag** 0: No overcurrent alarm 1: The driver has triggered an overcurrent alarm ◆ The driver has experienced an overcurrent alarm, and the possible reasons are as follows ① Short circuit occurs in the motor winding ② The current set by the driver is too large, causing the motor to burn out ③ Internal components of the driver are damaged
0	IVE	**Internal voltage error alarm flag** 0: No internal voltage error 1: Internal voltage error ◆ The internal voltage error alarm of the driver is usually caused by damage to the internal components of the driver

5.2.1.2. Driver Status Register [1]

Defined some status flags inside the driver. MODBUS Address: 1

15	11	10	9	8
Reserve	TC	POW	NL	PL

R-0

7	6	5	4	3	2	1	0
CLAMP	ARRSPD	RDY	HOME	MOV	INPOS	ALM	ENA
R-0	R-0	R-0	R-1	R-0	R-0	R-0	R-1

BIT	Name	Description
12～15	Reserve	**Read always returns 0**
11	TC	**Torque reaching state** 0: The torque has not reached the set value 1: The torque reaches the set value
10	STALL	**Stall detection** 0: During open-loop operation, the motor runs normally 1: During open-loop operation, the motor experiences stalling
9	NL	**Negative limit effective state** 0: Not in the negative limit position 1: In the negative limit position
8	PL	**Positive limit effective state** 0: Not in the positive limit position 1: In the positive limit position
7	CLAMP	**Motor mechanical brake status** 0: The brake is not opened, mechanically locking the motor shaft 1: The brake has been opened and the motor can run
6	ARRSPD	**Whether the motor is running to the set speed** 0: Speed not reached 1: The speed has reached ◆ In the internal pulse command mode, it is used to indicate whether the motor has reached the set speed
5	RDY	**Driver ready sign** 0: Not ready 1: Ready ◆ Usually, when the driver is in the enabled state, it is in the ready state. But during the transition from disable to enable of the motor, it takes 100ms to enter the ready state. In addition, automatic parameter recognition and current step testing during power on can cause the motor to be in an unprepared state.
4	HOME	**Homing sign** 0: Homing not completed 1: Homing completed
3	MOV	**Motor running sign** 0: Motor stopped state 1: The motor is running ◆ When the motor is in operation, it cannot respond to new motion commands and can only respond to stop commands.

2	INPOS	**Motor positioning completion sign in closed-loop mode** 0: Positioning not completed 1: Positioning Completed
1	ALM	**Driver alarm flag** 0: Driver without alarm 1: The driver has triggered an alarm, please check the register REG_LMCODE (address 0)
0	ENA	**Driver enable flag** 0: Driver not enabled 1: The driver has been enabled ◆ The default driver is already enabled when powered on

5.2.2. Input/output status register [2～7]

5.2.2.1. Input Port Value Register [2]

Used to indicate the value of the current input port. Due to the optoelectronic isolation of the input port, for ease of understanding, the article uses whether the optocoupler is conducting to indicate the status of the input port. MODBUS Address: 2

15	8
Reserve	
R-0	

7	6	5	4	3	2	1	0
Reserve		IN6	IN5	IN4	IN3	IN2	IN1
R-0		R-0	R-0	R-0	R-0	R-0	R-0

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	IN6	**Level status of input port IN6** 0: Input port 6 is not conducting 1: Input port 6 is conducting
4	IN5	**Level status of input port IN5** 0: Input port 5 is not conducting 1: Input port 5 is conducting
3	IN4	**Level status of input port IN4** 0: Input port 4 is not conducting 1: Input port 4 is conducting

2	IN3	**Level status of input port IN3** 0: Input port 3 is not conducting 1: Input port 3 is conducting
1	IN2	**Level status of input port IN2** 0: Input port 2 is not conducting 1: Input port 2 is conducting
0	IN1	**Level status of input port IN1** 0: Input port 1 is not conducting 1: Input port 1 is conducting

5.2.2.2. Current Output Port Value [3]

Output port value register. MODBUS Address: 3

15 ~ 8
Reserve
R-0

7 ~ 4	3	2	1	0
Reserve	OUT4	OUT3	OUT2	OUT1
R-0	R-0	R-0	R-0	R-0

BIT	Name	Description
4～15	Reserve	**Read always returns 0**
3	OUT4	**Level status of output port 4 (used by other products)** 0: Output port 4 is not conducting 1: Output port 4 is conducting
2	OUT3	**Level status of output port 3 (used by other products)** 0: Output port 3 is not conducting 1: Output port 3 is conducting
1	OUT2	**Level status of output port 2** 0: Output port 2 is not conducting 1: Output port 2 is conducting
0	OUT1	**Level status of output port 1** 0: Output port 1 is not conducting 1: Output port 1 is conducting

5.2.2.3. Input Port Conduction Edge Latch Register [4]

Every time the port changes from off state to on state, the driver will latch this change. MODBUS Address: 4

15	8
Reserve	
R-0	

7	6	5	4	3	2	1	0
Reserve		IN6	IN5	IN4	IN3	IN2	IN1
R-0		R-0	R-0	R-0	R-0	R-0	R-0

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	IN6	**Input port IN6 conduction edge latch flag** 0: Input port 6 has not experienced any conduction edge 1: Input port 6 has a conductive edge
4	IN5	**Input port IN5 conduction edge latch flag** 0: Input port 5 has not experienced any conduction edge 1: Input port 5 has a conductive edge
3	IN4	**Input port IN4 conduction edge latch flag** 0: Input port 4 has not experienced any conduction edge 1: Input port 4 has a conductive edge
2	IN3	**Input port IN3 conduction edge latch flag** 0: Input port 3 has not experienced any conduction edge 1: Input port 3 has a conductive edge
1	IN2	**Input port IN2 conduction edge latch flag** 0: Input port 2 has not experienced any conduction edge 1: Input port 2 has a conductive edge
0	IN1	**Input port IN1 conduction edge latch flag** 0: Input port 1 has not experienced any conduction edge 1: Input port 1 has a conductive edge

5.2.2.4. Input Port Shutdown Edge Latch Register [5]

Every time the port changes from on to off, the driver will latch this change. MODBUS Address: 5

15	8
Reserve	
R-0	

7	6	5	4	3	2	1	0
Reserve		IN6	IN5	IN4	IN3	IN2	IN1
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	IN6	**Input port IN6 shutdown edge latch flag** 0: Input port 6 did not experience any pass/fail edges 1: Input port 6 experienced a shutdown edge
4	IN5	**Input port IN5 shutdown edge latch flag** 0: Input port 5 did not experience any pass/fail edges 1: Input port 5 experienced a shutdown edge
3	IN4	**Input port IN4 shutdown edge latch flag** 0: Input port 4 did not experience any pass/fail edges 1: Input port 4 experienced a shutdown edge
2	IN3	**Input port IN3 shutdown edge latch flag** 0: Input port 3 did not experience any pass/fail edges 1: Input port 3 experienced a shutdown edge
1	IN2	**Input port IN2 shutdown edge latch flag** 0: Input port 2 did not experience any pass/fail edges 1: Input port 2 experienced a shutdown edge
0	IN1	**Input port IN1 shutdown edge latch flag** 0: Input port 1 did not experience any pass/fail edges 1: Input port 1 experienced a shutdown edge

5.2.2.5. Input Port Conduction Edge Clearing Register [6]

Used to clear the conduction edge flag of the latch. MODBUS Address: 6

15	8
Reserve	
R-0	

7	6	5	4	3	2	1	0
Reserve		IN6	IN5	IN4	IN3	IN2	IN1
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	IN6	**Clear the conduction edge latch status flag of IN6** 0: No effect 1: Clear the conduction edge latch flag of IN6 port
4	IN5	**Clear the conduction edge latch status flag of IN5** 0: No effect 1: Clear the conduction edge latch flag of IN5 port
3	IN4	**Clear the conduction edge latch status flag of IN4** 0: No effect 1: Clear the conduction edge latch flag of IN4 port
2	IN3	**Clear the conduction edge latch status flag of IN3** 0: No effect 1: Clear the conduction edge latch flag of IN3 port
1	IN2	**Clear the conduction edge latch status flag of IN2** 0: No effect 1: Clear the conduction edge latch flag of IN2 port
0	IN1	**Clear the conduction edge latch status flag of IN1** 0: No effect 1: Clear the conduction edge latch flag of IN1 port

5.2.2.6. Input Port Shutdown Edge Clearing Register [7]

Used to clear the off edge flag of the latch. MODBUS Address: 7

15	8
Reserve	
R-0	

7	6	5	4	3	2	1	0
Reserve		IN6	IN5	IN4	IN3	IN2	IN1
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	IN6	**Clear the off edge latch status flag of IN6** 0: No effect 1: Clear the shutdown edge latch flag of IN5 port
4	IN5	**Clear the off edge latch status flag of IN5** 0: No effect 1: Clear the shutdown edge latch flag of IN5 port
3	IN4	**Clear the off edge latch status flag of IN4** 0: No effect 1: Clear the shutdown edge latch flag of IN4 port
2	IN3	**Clear the shutdown edge latch status flag of IN3** 0: No effect 1: Clear the shutdown edge latch flag of IN3 port
1	IN2	**Clear the latch status flag of IN2's turn off edge** 0: No effect 1: Clear the shutdown edge latch flag of IN2 port
0	IN1	**Clear the off edge latch status flag of IN1** 0: No effect 1: Clear the shutdown edge latch flag of IN1 port

5.2.3.Current position and speed related registers of the motor [8～16]

MODBUS address	Attribute	Default value	Range	Description
8	R	0	[0,65535]	Internal pulse mode, current absolute position, low 16-bit
9	R	0	[0,65535]	Internal pulse mode, current absolute position, high 16-bit
10	R	0	[-3000,3000]	Current command speed, signed 16-bit data Unit: RPM
11	R	-	[0,100]	Current bus voltage value Unit: mV
12	R	0	[0,65535]	Closed-loop mode, motor tracking error, low 16-bits Unit: Encoder Resolution
13	R	0	[0,65535]	Closed-loop mode, motor tracking error, high 16-bits
14	R	0	[0,65535]	External pulse counter, low 16-bit
15	R	0	[0,65535]	External pulse counter, high 16-bit
16	R/W	0	[0,1]	Clear external pulse counter Writing 0 has no effect, reading always returns 0 Writing 1 will clear the external pulse counter, and the values of registers 14 and 15 will become 0. Then this register will become 0.

5.2.4. Driver control mode setting [17～23]

MODBUS address	Attribute	Default value	Range	Description
17	R/W	0	[0,1]	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
18	R/W	0	[0,6]	0: Waiting state The driver will restore the bit waiting state after processing any control instructions received. 1: Fixed length forward rotation In relative position mode, the motor runs forward according to the 70-74 register parameters. In absolute position mode, the operating status is determined based on the current position and the

				absolute position set at 70-74. 2: Fixed length reversal In relative position mode, the motor operates in reverse according to the 70-74 register parameters. In absolute position mode, the operating status is determined based on the current position and the absolute position set at 70-74. 3: Speed mode, continuous forward rotation The motor accelerates forward according to registers 75 and 77 4: Speed mode, continuous reverse The motor accelerates in reverse according to registers 75 and 77 6: Slow down and stop Position mode, the motor decelerates and stops according to the 71 register Speed mode, the motor decelerates and stops according to the 76 register 9: Trigger homing This register only works when the value of the internal pulse mode register 20 is 0
19	R/W	0	[0,2]	External pulse command mode setting register 0: PUL+DIR mode: PUL is the pulse input port, DIR is the direction input port 1: Reserved 2: CW+CCW mode: PUL is the CW input signal, DIR is the CCW input signal 3: Orthogonal input mode:
20	R/W	0	[0,5]	Preset application selection for internal pulse mode 0: Communication control: responds to instructions from register 18 1: Reserved, do not use 2: Preset IO control mode 1 start stop + direction 3: Preset IO control Mode 2: forward rotation + reverse rotation 4: Preset IO control mode 3: speed table 5: Preset IO control mode 4: internal position table
21	R/W	0	[0,1]	Motor type setting register 0: Two phase stepper motor 1: Three phase stepper motor

				2: Five phase stepper motor ◆ The IR42/IT42 series is a two-phase motor, please do not make any other settings
22	R/W	0	[0,2]	Motor operation mode setting register 0: Open loop operation 1: Servo operation
23	R	0	[0,1]	Motor direction reverse setting register 0: Default running direction 1: Reverse the direction of motor operation ◆ Due to the direction of the IR42/IT42 series being set through SW5, this register is used to display the current settings

5.2.5. Open Loop Operating Parameters [24～29]

MODBUS address	Attribute	Default value	Range	Description
24	R	4000	[200,65535]	Display the pulse/revolution of driver runtime Unit: Pulse/rev
25	R	1500	[0,6000]	Display the maximum operating current of the driver Unit: mA
26	R/W	50	[0,100]	Standby current percentage Set the percentage of current relative to operating current when the driver enters standby mode in open-loop operation mode. Unit:%
27	R/W	500	[10,65535]	Standby time setting Set the time after the pulse stops for a certain period of time when the driver enters standby mode during open-loop operation. Unit: ms
28	R/W	512	[1,1024]	Pulse command filter Used for smoothing pulse commands (including internal and external pulses), filtering time = set value * 50us
29	Reserve	-	-	-

5.2.6. Motor & Current Loop Parameters [30～39]

MODBUS address	Attribute	Default value	Range	Description
30	R/W	0	[0,1]	Auto-PI enable function Driver built-in parameter recognition and gain optimization algorithm. Usually, good results can be achieved. If the customer needs optimization, this feature can be disabled. 0: Do not use auto-PI function 1: Use the auto-PI function
31	R	-	[100,65535]	Automatically recognized resistance value Read the motor winding resistance value automatically recognized by the driver. Unit: mOhm
32	R	-	[1,65535]	Automatically recognized inductance value Read the motor winding inductance value automatically recognized by the driver. Unit: mH
33	R/W	1000	[100,10000]	User set resistance value When the automatic PI function is cancelled, the resistance value set by the user takes effect. Unit: mOhm
34	R/W	1	[1,10]	User set inductance value When the automatic PI function is disabled, the inductance value set by the user takes effect. Unit: mH
35	R/W	200	[0,1000]	Motor torque constant The parameters are only valid when the motor control mode is servo mode two
36	R/W	1000	[200,10000]	Proportional gain of current loop KP in the current loop PI algorithm. When the auto-PI function is enabled, ILOOPKP is automatically generated. When the auto-PI function is disabled, users can modify ILOOPKP.

37	R/W	200	[0,2000]	Current loop integral gain KI in the current loop PI algorithm. When the auto-PI function is enabled, ILOOPKI is automatically generated. When the automatic PI function is disabled, users can modify ILOOPKI
38	R/W	256	[0,1024]	KC in Current Loop PI Algorithm
39	R/W	0	[0,1]	Current step test Writing 0 is no effect, reading always returns 0. Writing 1 will initiate the current loop step test. At this point, the current of the motor winding will first be zero and then increase to 1000mA. Users can view step response through NTConfigurator, manually adjust ILOOPKP and ILOOPKI, and optimize motor response.

5.2.7. Closed Loop Motor Parameters [40～49]

MODBUS address	Attribute	Default value	Range	Description
40	R/W	4000	[256,65535]	Encoder resolution The driver is capable of receiving input signals from orthogonal encoders and performing 4th harmonic processing. Encoder resolution = number of encoder lines * 4
41	R/W	2000	[100,65535]	Positioning completion detection mode 0: Immediate mode, at any time as long as the position error is less than the value set in register 42 and continues for the time set in register 43, it is considered to be in place 1: The command stop mode only judges the error after the motor motion instruction stops for more than the time set in register 44.
42	R/W	10	[1,65535]	Positioning completion accuracy Unit: Encoder resolution

43	R/W	50	[1,65535]	Positioning completion duration Set the duration after the motor enters the completion accuracy. Duration = Set value * 50us
44	R/W	100	[1,65535]	The time when the positioning is completed and the detection begins After the driver stops receiving pulses for a set period of time, it begins to determine whether the positioning is complete. Set time = set value * 50us
45	R/W	4000	[0,65535]	Out of tolerance alarm threshold, low 16-bit
46	R/W	0	[0,65535]	Out of tolerance alarm threshold, high 16-bit When both registers 45 and 46 are 0, cancel the out of tolerance alarm.
47	R/W	0	[0,1]	Encoder signal A, B exchange It is possible to match motors from other manufacturers without changing the wiring of the motor and encoder.
48	R	0	[0,65535]	Encoder single-turn count value
49	-	-	-	-

5.2.8. Output Port Parameters [50～59]

IR42/IT42 products do not have output ports, and corresponding setting registers are reserved here for compatibility with other products.

15	8
Reserve	
R/W-0	

7	6	5	4	3	0
Reserve			Polarity	Function	
R-0			R/W-0	R/W-0	

BIT	Name	Description
5～15	Reserve	**Read always returns 0**
4	Polarity	**Output port polarity** 0: Normally close 1: Normally open (default value)
0～3	Function	**Output port function selection** 0: Normal output, user controlled 1: Alarm output, default value for OUT1 2: Brake signal output 3: In place signal output 4: Speed reaches output, OUT2 default value 5: Homing completed output 6: Driver ready for output 7: Motor stop state output 8:: Positive limit output 9: Negative limit output 10: Power indicator output 11: Torque reaches output Other: The input port is not functional and only serves as a regular input port

Homing completed output:

MODBUS address	Attribute	Default value	Range	Description
50	R/W	1	[0,11]	Output port 1 setting register (other products)
51	R/W	4	[0,11]	Output port 2 setting register (other products)
52	R/W	1	[0,11]	Output port 3 setting register (other products)
53	R/W	4	[0,11]	Output port 4 setting register (other products)

5.2.9. Input Port Parameters [60～69]

The IR42/IT42 product includes 3 input ports, each with the same setting method.

15	8
Reserve	
R/W-0	

7	6	5	4	0
Reserve		Polarity	Function	
R-0		R/W-0	R/W-0	

BIT	Name	Description
6～15	Reserve	**Read always returns 0**
5	Polarity	**Effective voltage level of input port** 0: Normally close 1: Normally open (default value)
0～4	Function	**Input port function selection** 0: Pulse Inputs 1: Direction input 2: Orthogonal encoder A-phase input 3: Orthogonal encoder B-phase input 4: Motor offline 5: Clear fault 6: Emergency stop 7: Jog forward / start stop 8: Jog reverse / direction 9: Positive limit input 10: Negative limit input 11: Zero point signal 12: Start homing 13: Reverse the direction of motor operation 14: Multi-segment speed control 0 15: Multi-segment speed control 1

		16: Multi-segment speed control 2 17: Multi-segment speed control 3 18: Multi-segment position control 0 19: Multi-segment position control 1 20: Multi-segment position control 2 21: Multi-segment position control 3 22：USER1 23：USER2 24：USER3 25：USER4 26：USER5 27：USER6 28：USER7 29：USER8 30：USER9 31：USER10 Other: The input port is not functional and only serves as a regular input port

MODBUS address	Attribute	Default value	Range	Description
60	R/W	0	[0,31]	Input port 1 setting register
61	R/W	1	[0,31]	Input port 2 setting register
62	R/W	4	[0,31]	Input port 3 setting register
63	R/W	7	[0,31]	Input port 4 setting register
64	R/W	12	[0,31]	Input port 5 setting register
65	R/W	11	[0,31]	Input port 6 setting register

5.2.10. PP & PV Mode Motion Parameters [70～79]

MODBU address	Attribute	Default value	Range	Description
70	R/W	200	[10,1000]	Acceleration during point-to-point motion, unit: R/S ^ 2
71	R/W	200	[10,1000]	Deceleration during point-to-point motion, unit: R/S ^ 2
72	R/W	300	[0,3000]	Maximum speed during point-to-point movement, unit: RPM
73	R/W	2000	[-16777216,16777216]	Pulse command during point-to-point movement, unit: number of pulses P73 is low 16-bit data, P74 is high 16-bit data
74				
75	R/W	60	[10,1000]	Jog acceleration, unit: R/S ^ 2
76	R/W	60	[10,1000]	Jog deceleration, unit: R/S ^ 2
77	R/W	300	[0,3000]	Jog speed, unit: RPM
78	R/W	0	[0,1]	Absolute motion (ABS) / relative motion (INC) motion settings
79	W	0	1	Write 1 to clear position counter (registers 8, 9)

- The 73 and 74 registers form a 32-bit signed register.
- In incremental operation mode, the absolute values of 73 and 74 represent the distance traveled, and register 18 is used to write 1 or 2 to control whether the motor runs in the forward or reverse direction.
- In absolute position mode, the signed data consisting of 73 and 74 represents the target position, and the motor runs to the set distance by writing 1 through 18.

5.2.11. Homing Parameters [80～89]

MODBUS address	Attribute	Default value	Range	Description
80	R/W	200	[1,3000]	Homing high speed Unit: RPM
81	R/W	40	[1,3000]	Homing low speed Unit: RPM
82	R/W	100	[0,1000]	Homing acceleration
83	R/W	0	[-32768,32767]	Zero point offset
84	R/W	0	[0,3]	Zero point offset mode setting
85	R/W	0	[17，35]	Homing mode selection
86	R/W	0	[0,1000]	Emergency stop deceleration
87	R/W	50	[0,1000]	Stop time of homing process Unit: ms ◆ During the homing process, the driver will detect the status of the sensor and make corresponding deceleration stop actions. After the motor stops, after a set time, it will further act according to the homing method
88	R/W	5000	[0,65535]	Homing timeout time Unit: ms ◆ After triggering the homing, if the set timeout period is exceeded and the homing is still not completed, the motor will slow down and stop. Waiting for the user's next trigger.
89	R/W	0	[0,1]	Enable automatic homing upon power on 0: It is prohibited to reset to homing after powering on 1: After the driver is powered on, it starts homing

5.2.12. Auxiliary Function Register [90～99]

MODBUS address	Attribute	Default value	Range	Description
90	R/W	0	[0,1]	Writing 0 is invalid, reading returns 0 Write 1 to save the current parameters
91	R/W	0	[0,1]	Writing 0 is invalid, reading returns 0 Write 1 to restore factory settings
92	-	-	-	Manufacturer reserves use, user prohibits writing data
93	R	-	-	Driver ID number
94	R	-	-	Driver version number
95	R	-	-	-

5.2.13. Shaft Locking Time [105,106]

When the stepper motor is initially powered on, the position is at a random position, and the initial position between the stator and rotor of the motor needs to be determined by locking the shaft. Locking the shaft through a current ramp can reduce the vibration of the motor during power on and enable.

MODBUS address	Attribute	Default value	Range	Description
105	R/W	16000	[0,65535]	Lock shaft ramp time; unit: 62.5us
106	R/W	4000	[0,65535]	Lock shaft duration; unit: 62.5us

5.2.14. Current & Pulse/revolution Parameters [110~114]

MODBUS address	Attribute	Default value	Range	Description
110	R/W	1500	[0,1500]	Unit mA; The maximum current of IR42/IT42 is 1500 When in open-loop mode, this register is used to set the current for open-loop operation mode. In closed-loop mode, this register is used to set the maximum current for closed-loop operation mode.
111	R/W	4000	[200,65535]	Set the pulse required for one rotation of the motor
112	Reserve			
113	R/W	30	[0,100]	Current compensation percentage
114	R/W	30	[0,100]	Current loop gain compensation percentage

5.2.15. Phase Loss Detection [115,116]

Phase loss refers to an abnormality in the winding of a stepper motor, such as a broken wire in the motor winding

The IR42/IT42 series products include the connection between the motor winding and the driver circuit board, which monitors the motor winding status through a current detection circuit to ensure that the corresponding current is fed back normally. When a phase loss alarm occurs, the driver shuts off power output and the motor is in a free state.

In some cases, it is impossible to detect phase loss, such as when the driver is powered on for the first time, the current reference value of the B-phase winding is zero. If the B-phase winding breaks, the current detected by the driver is also zero, and the driver cannot give an alarm signal at this time. If a command is sent to make the motor run for more than 2 full steps, the driver can determine the phase loss state of the B-phase winding

MODBU address	Attribute	Default value	Range	Description
115	R/W	0	[0,1]	Enable phase loss detection function 0: Disable phase loss detection function 1: Enable phase loss detection function
116	R/W	50	[0,1500]	Phase loss detection current threshold: mA

5.2.16. Stall Detection [117~119]

The IR42/IT42 series products can operate in open-loop mode, and the driver is equipped with a sensorless stall detection function. Stall refers to the inability of a stepper motor to rotate normally due to external reasons, which can cause the stepper motor to overload, even burn out, and damage mechanical components of the equipment.

The IR42/IT42 series products collect real-time current feedback from the motor, estimate the load torque changes of the motor through an observer, and then determine whether the motor is experiencing stalling. Once motor stalling is detected, corresponding protective measures can be taken.

Using sensorless technology for locked rotor detection, the accuracy of the detection depends on the recognition of motor parameters and the correct setting of relevant observer parameters. The default parameter recognition of the driver is usually sufficient for this task. When a misjudgment occurs in practical testing applications, users can turn off this function.

When a stall condition is detected, even if the controller continues to send commands to the driver, the driver will stop running, but the motor will remain in full current state and lock the shaft. At the same time, the red alarm indicator light will remain on, and the alarm output port will output an alarm signal. When the user stops sending pulses, the driver will automatically clear the alarm state and respond to subsequent command signals.

The related registers are as follows:

MODBUS address	Attribute	Default value	Range	Description
117	R/W	0	[0,1000]	Stall detection enable 0--- Disable the stall detection function 0--- Enable the stall detection function
118	R/W	180	[0,3000]	Starting speed for stall detection: RPM
119	R/W	2	[0,65535]	Stall detection position error alarm threshold

5.2.17. Low-speed Resonance [120~129]

The IR42/IT42 series products provide a harmonic injection algorithm to effectively reduce the low-speed resonance of two-phase stepper motors when operating in open-loop mode.

Usually, the first resonance point speed V1 of a two-phase stepper motor is around 60 RPM, and the second resonance point speed V2 is around 120 RPM. The specific resonance speed of a motor is directly related to its inertia, current, etc., and requires testing to discover the resonance point under different conditions.

The speed relationship between the two resonance points is V2 = 2 * V1. To debug the resonance, the following steps need to be completed:

The first step is to identify two resonance points and determine the velocity range of the resonance region. External pulses or internal pulses can be used to compare and identify the first resonance point, and then maintain the current speed.

Step two, adjust the amplitude and phase of the first resonance point to significantly reduce the resonance of the motor.

Step three, run the motor at the second resonance point and maintain the current speed.

Step four, adjust the amplitude and phase of the second resonance point to significantly reduce the resonance of the motor.

MODBUS address	Attribute	Default value	Range	Description
120	R/W	0	[0,1500]	The amplitude of the first resonance point
121		0	[0,1024]	Phase of the first resonance point
122		0		Reserve
123		0	[0,1500]	The amplitude of the second resonance point
124		0	[0,1024]	The amplitude of the second resonance point
125		0		Reserve
126		0	[0,600]	The initial velocity of the first vibration point
127		0	[0,600]	The speed of the first vibration point has ended
128		0	[0,600]	Starting speed of the second vibration point
129		0	[0,600]	The speed of the second vibration point ends

5.2.18. Closed Loop Control Parameters [130~159]

MODBUS address	Attribute	Default value	Range	Description
130	R/W	300	[1,5000]	First order speed filtering bandwidth
131		900	[1,5000]	Second order speed filtering bandwidth
132		2000	[1,5000]	Second order speed filtering bandwidth
133~149				Reserve
150		3500	[500,20000]	Position loop Kp
151		0	[0,65535]	Position loop Kd
152		40	[10,500]	Speed loop Kp
153		100	[0,600]	Speed loop KI
154		0	[0,65535]	Acceleration feedforward
155		3000	[0,5000]	Position loop output filtering bandwidth
156		3000	[0,5000]	Position loop differential output filtering bandwidth
157		500	[0,5000]	Speed loop output filtering bandwidth
158		80	[0,100]	Proportion of first-order speed feedback
159		Reserve	Reserve	Reserve

By adjusting the parameters of the position loop and speed loop, it is possible to achieve rigid adjustment of position tracking

5.2.19. Speed Table Parameters Setting [205～220]

MODBUS address	Attribute	Default value	Range	Description
205	R/W	0	[-3000,3000]	Speed table 1, unit: RPM
206	R/W	20		Speed table 2, unit: RPM
207	R/W	30		Speed table 3, unit: RPM
208	R/W	50		Speed table 4, unit: RPM
209	R/W	60		Speed table 5, unit: RPM
210	R/W	80		Speed table 6, unit: RPM
211	R/W	100		Speed table 7, unit: RPM
212	R/W	150		Speed table 8, unit: RPM
213	R/W	200		Speed table 9, unit: RPM
214	R/W	250		Speed table 10, unit: RPM
215	R/W	300		Speed table 11, unit: RPM
216	R/W	400		Speed table 12, unit: RPM
217	R/W	500		Speed table 13, unit: RPM
218	R/W	600		Speed table 14, unit: RPM
219	R/W	700		Speed table 15, unit: RPM
220	R/W	800		Speed table 16, unit: RPM

6. Application Instructions for Motion Control

6.1. Pulse Control Mode

IR42/IT42 series products use pulse + direction control by default when they leave the factory. The related register settings are as follows:

Step	Register address	Unit	Value	Parameter description
1	17	---	1	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
2	19	---	0	External pulse command mode setting register 0: PUL+DIR mode: PUL is the pulse input port, DIR is the direction input port 1: Reserved 2: CW+CCW mode: PUL is the CW input signal, DIR is the CCW input signal 3: Orthogonal input mode
3	110	mA	1500mA	\The current of T60S series drivers has been comprehensively optimized by default. Users can further optimize based on this.
4	111	Pulse/rev	4000	Default control using 4000 pulses per revolution

For details, see section 4.3.2

6.2. Communication Control Mode

In this mode, users can perform homing, positioning, and speed control movements of the motor through communication

Parameter settings for communication control mode:

Step	Register address	Unit	Value	Parameter Description
1	17	---	0	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
2	20	---	0	Preset application selection for internal pulse mode 0: Communication control: responds to instructions from

				register 18 1: Reserved, do not use 2: Preset IO control mode 1: start stop + direction 3: Preset IO control mode 2: forward rotation + reverse rotation 4: Preset IO control mode 3: speed table 5: Preset IO control mode 4: position table
3	18	---		0: Waiting state. The driver will restore the bit waiting state after processing any control commands received. 1: Fixed length forward rotation. In relative position mode, the motor runs forward according to the 70-74 register parameters. In absolute position mode, the operating status is determined based on the current position and the absolute position set at 70-74. 2: Fixed length reverse. In relative position mode, the motor operates in reverse according to the 70-74 register parameters. In absolute position mode, the operating status is determined based on the current position and the absolute position set at 70-74. 3: Speed mode, continuous forward rotation. The motor accelerates forward according to registers 75 and 77 4: Speed mode, continuous reverse. The motor accelerates in reverse according to registers 75 and 77 6: Slow down and stop. Position mode, the motor decelerates and stops according to the 71 register Speed mode, the motor decelerates and stops according to the 76 register 9: Trigger homing ◆ This register only works when the value of the internal pulse mode register 20 is 0

6.2.1. PP Position Control Mode

The IR42/IT42 series products have the function of communication control motor operation with specified pulse stroke. The specific modes and parameters that need to be set are as follows (register addresses are decimal numbers unless otherwise specified or indicated):

(1) Set the value of register address 20 (preset application selection in internal pulse mode) to 0 (communication control, responding to commands at register address 18);

(2) Set the functions of digital input and output ports according to application needs and actual wiring terminals;

(3) Set motion parameters:

Address	Unit	Parameter Description
70	R/S^2	Acceleration of position mode
71	R/S^2	Deceleration of position mode
72	RPM	Speed of position mode
73	Command Pulse	Number of command pulses for position mode, low 16-bit register
74	Command Pulse	Number of command pulses for position mode, high 16-bit register
78	R/S^2	Set position operation mode: 0: Incremental 1: Absolute
79	-	Write 1 and clear the position counter (registers 8 and 9)

(4) Communication given operation command: start point-to-point movement by writing values 1 (fixed length forward rotation) and 2 (fixed length reverse rotation) to register 18 (for detailed commands on this register, please refer to register 18 in "Driver Control Mode Settings [17-23]");

(5) During operation, if shutdown is required, the value 6 can be written to register 18 (deceleration stop, deceleration is set to the value in register 71)

◆ Attention: Changing the acceleration (register 70), deceleration (register 71), and speed (register 72) during the operation of the motor will not immediately respond to these set values by the driver. It is necessary to trigger them by writing the corresponding commands again to register 18.

6.2.2. PV Speed Control Mode

The IR42/IT42 series products have the function of controlling the motor's jog operation through communication. The specific modes and parameters that need to be set are as follows (register addresses are decimal numbers unless otherwise specified or indicated):

(1) Set the value of register address 20 (preset application selection in internal pulse mode) to 0 (communication control, responding to instructions at register address 18);

(2) Set the functions of digital input and output ports according to application needs and actual wiring terminals;

(3) Set motion parameters:

Address	Unit	Parameter Description
75	R/S^2	Acceleration of Jog motion
76	R/S^2	Deceleration of jog motion
77	RPM	Speed of jog motion

(4) Communication given operation command: start point-to-point movement by writing values 3 (continuous forward rotation) and 4 (continuous reverse rotation) to register 18 (for detailed instructions on this register, please refer to register 18 in "Driver Control Mode Settings [17-23]");

(5) During operation, if shutdown is required, the value 6 can be written to register 18 (deceleration stop, deceleration is set to register 76).

◆ Attention:

(1) The motor is in operation and only responds to stop commands (deceleration stop or emergency stop). If it is necessary to change the direction of motor operation through instructions, a stop command needs to be sent to wait for the motor to stop before sending a start signal in the other direction.

(2) During the operation of the motor, the speed can be changed (register 77), and the driver will immediately respond, that is, the motor will immediately run at the set speed value without the need to stop and restart before responding.

6.2.3. Homing

The IR42/IT42 series closed-loop stepper driver products have three working modes: power on automatic homing, communication triggered homing, and IO triggered homing. They support the 17~30 and 35 homing methods defined in CANOPEN

6.3. Preset IO Control Mode 1: Start Stop+Direction

The IR42/IT42 series closed-loop stepper driver products can control the operation of the motor using two IN ports through this mode. One IN terminal is used to control the start/stop of the motor, and the other IN terminal is used to control the direction of operation of the motor. The specific settings are as follows:

Step	Register address	Unit	Value	Parameter Description
1	17	---	0	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
2	20	---	2	Preset application selection for internal pulse mode 0: Communication control: responds to instructions from register 18 1: Reserved, do not use 2: Preset IO control mode 1: start stop + direction 3: Preset IO control mode 2: forward rotation + reverse rotation 4: Preset IO control mode 3: speed table 5: Preset IO control mode 4: position table
3	60	---	39	PUL- port (IN1) is set to "7-jog forward rotation / start stop" and normally open; Set to 7 when normally closed.
	61	---	40	DIR- port (IN2) is set to "8-jog reverse / direction" and normally open; Set to 8 when normally closed.
4	75	R/S^2	60	Acceleration of jog motion
	76	R/S^2	60	Deceleration of jog motion
5	205	RPM	100	Speed of motor operation

- The acceleration (register 75), deceleration (register 76), and speed (register 205) can be dynamically changed during the operation of the motor, and the driver will immediately respond to these set values.

- The direction signal can be switched during the operation of the motor. At this time, the motor will slow down and stop at the deceleration set in register 76, and then accelerate in the opposite direction to the set speed.

Under the above register settings, connecting 24V to the COM port and controlling the status of PUL-, DIR-, EN- can control the operation of the motor:

PUL- level	DIR- level	EN- level	Motor status
/	/	0V	Offline
0V	24V	24V	Forward
0V	0V		Reverse
24V	24V		Not run
24V	0V		Not run

6.4. Preset IO Control Mode 2: Forward Rotation+Reverse Rotation

The IR42/IT42 series products can control the operation of the motor using two IN ports through this mode. One IN terminal is used to control the start/stop of the motor, and the other IN terminal is used to control the direction of operation of the motor. The specific settings are as follows:

Step	Register address	Unit	Value	Parameter Description
1	17	---	0	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
2	20	---	3	Preset application selection for internal pulse mode 0: Communication control: responds to instructions from register 18 1: Reserved, do not use 2: Preset IO control mode 1: start stop + direction 3: Preset IO control mode 2: forward rotation + reverse rotation 4: Preset IO control mode 3: speed table 5: Preset IO control mode 4: position table
3	60	---	39	PUL- port (IN1) is set to "7- jog forward/start stop" and normally open; Set to 7 when normally closed.
	61	---	40	DIR- port (IN2) is set to "8- jog reversel/direction" and normally open; Set to 8 when normally closed.
4	75	R/S^2	60	Acceleration of jog motion
	76	R/S^2	60	Deceleration of jog motion
5	205	RPM	100	Speed of motor operation

- The acceleration (register 75), deceleration (register 76), and speed (register 205) can be dynamically changed during the operation of the motor, and the driver will immediately respond to these set values.
- The direction signal can be switched during the operation of the motor. At this time, the motor will slow down and stop at the deceleration set in register 76, and then accelerate in the opposite direction to the set speed.

Under the above register settings, connecting 24V to the COM port and controlling the status of PUL-, DIR-, EN- can control the operation of the motor:

PUL- Level	DIR- Level	EN- Level	Motor status
/	/	0V	Offline
0V	24V	24V	Forward
24V	0V		Reverse
24V	24V		Not run
0V	0V		Not run

6.5. Preset IO Control Mode 3: Speed Table Mode

The IR42/IT42 series products can control up to 8 speeds through up to 3 IOs. Usually, the first gear speed is set to 0, indicating that the motor stops when there is no signal input. The relevant register settings are as follows:

Step	Register address	Unit	Value	Parameter Description
1	17	---	0	Command mode setting register, set the source of pulse commands for the driver 0: Internal pulse command 1: External pulse command
2	20	---	4	Preset application selection for internal pulse mode 0: Communication control: responds to instructions from register 18 1: Reserved, do not use 2: Preset IO control mode 1: start stop + direction 3: Preset IO control mode 2: forward rotation + reverse rotation 4: Preset IO control mode 3: speed table 5: Preset IO control mode 4: position table
3	60	---	46	PUL- port (IN1) is set to "14- Multi segment speed 0" and normally open;

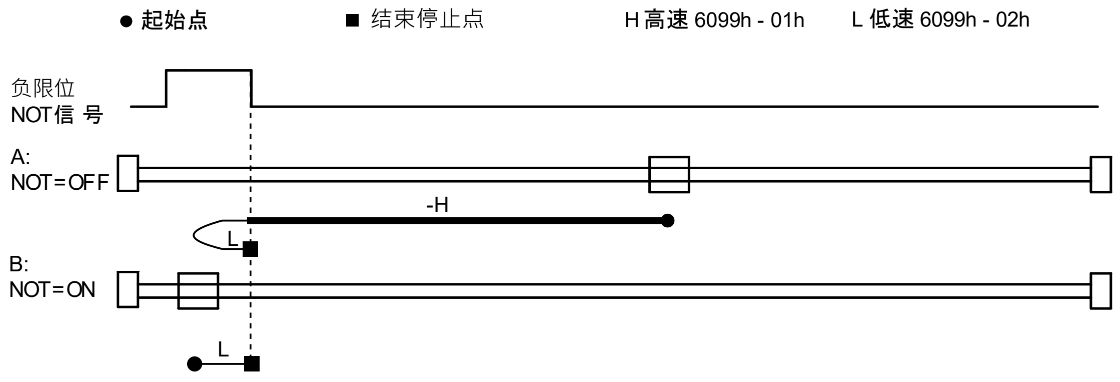
				Set to 14 when normally closed.
	61		47	DIR- port (IN2) is set to "15- Multi-segment Speed 1" and normally open; Set to 15 when normally closed.
	62		48	EN- port (IN3) is set to "16- Multi-segment Speed 2" and normally open; Set to 16 when normally closed.
4	75	R/S^2	60	Acceleration of jog motion
	76	R/S^2	60	Deceleration of jog motion
5	205	RPM	0	Speed table 1
	206		200	Speed table 2
	207		-200	Speed table 3
	208		300	Speed table 4
	209		-300	Speed table 5
	210		400	Speed table 6
	211		-400	Speed table 7
	212		500	Speed table 8

- The value of the speedometer register is signed 16-bit data, which can be used to achieve forward and reverse operation of the motor by setting the positive and negative values of the speed.

6.6. Homing Method

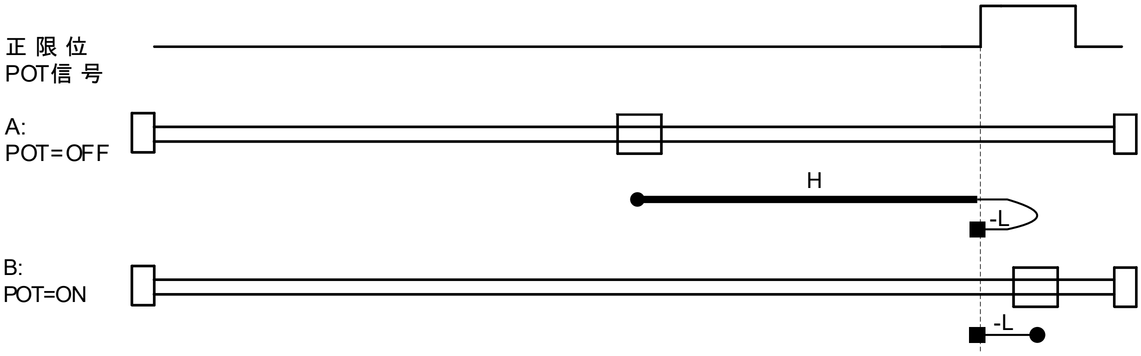
The IR42/IT42 series closed-loop stepper driver products have three working modes: power on automatic homing, communication triggered homing, and IO triggered homing. They support the 17~30 and 35 homing modes defined in CANOPEN. The specific process of the homing method is described below.

6.6.1. Method 17



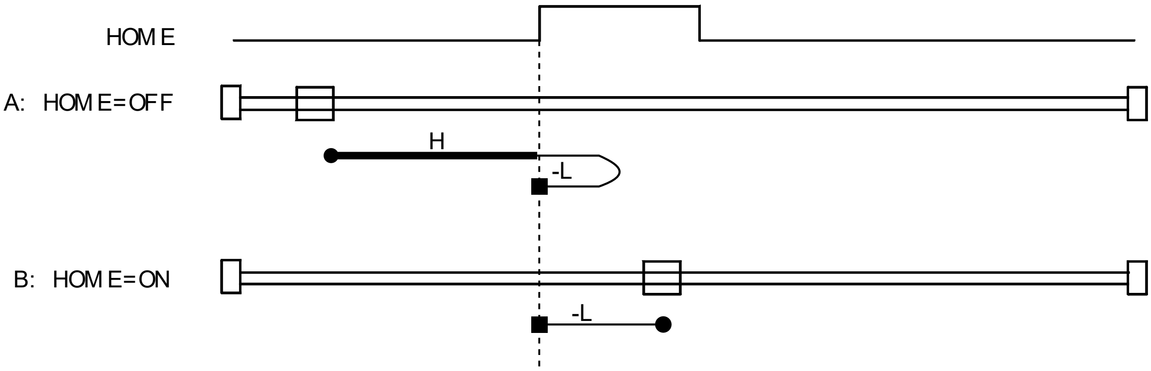
6.6.2. Method 18

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



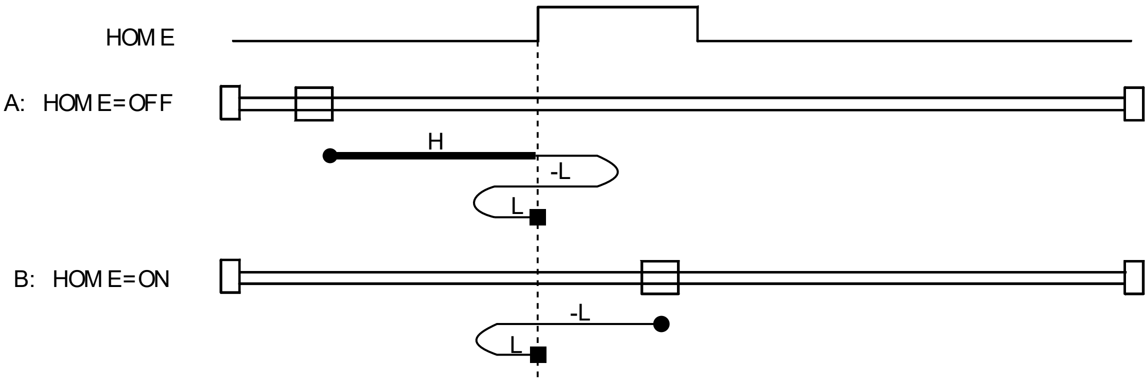
6.6.3. Method 19

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



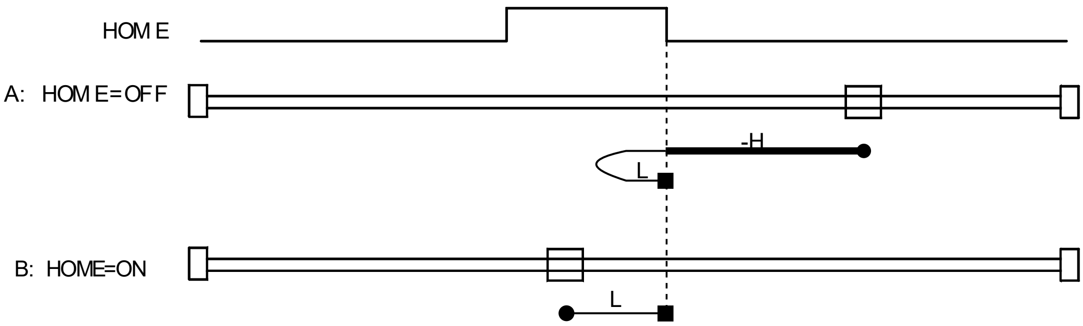
6.6.4. Method 20

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



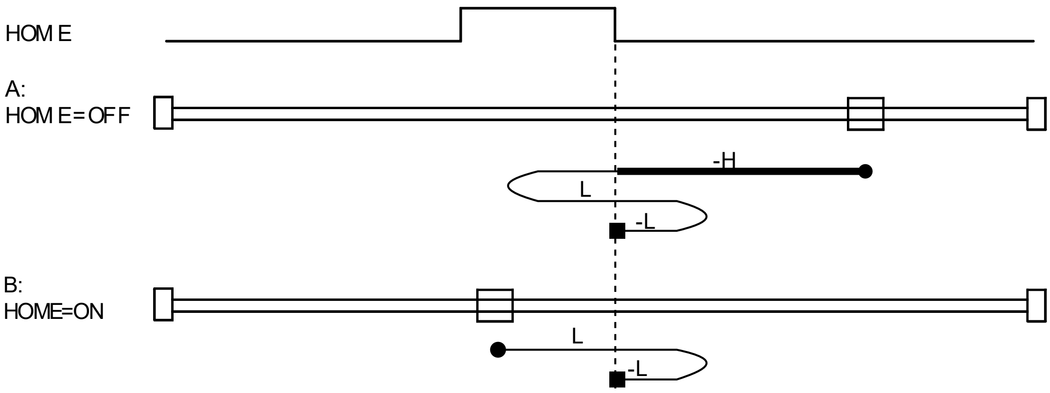
6.6.5. Method 21

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



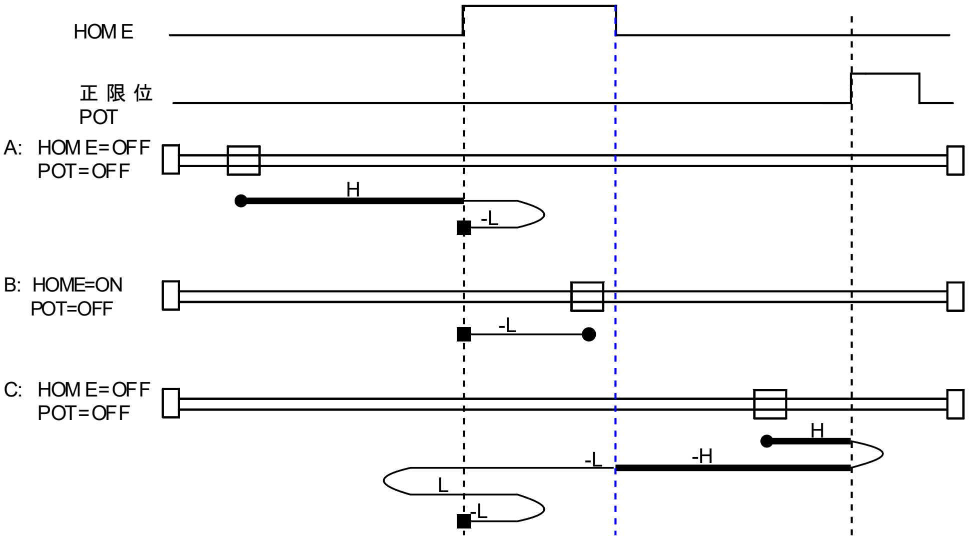
6.6.6. Method 22

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



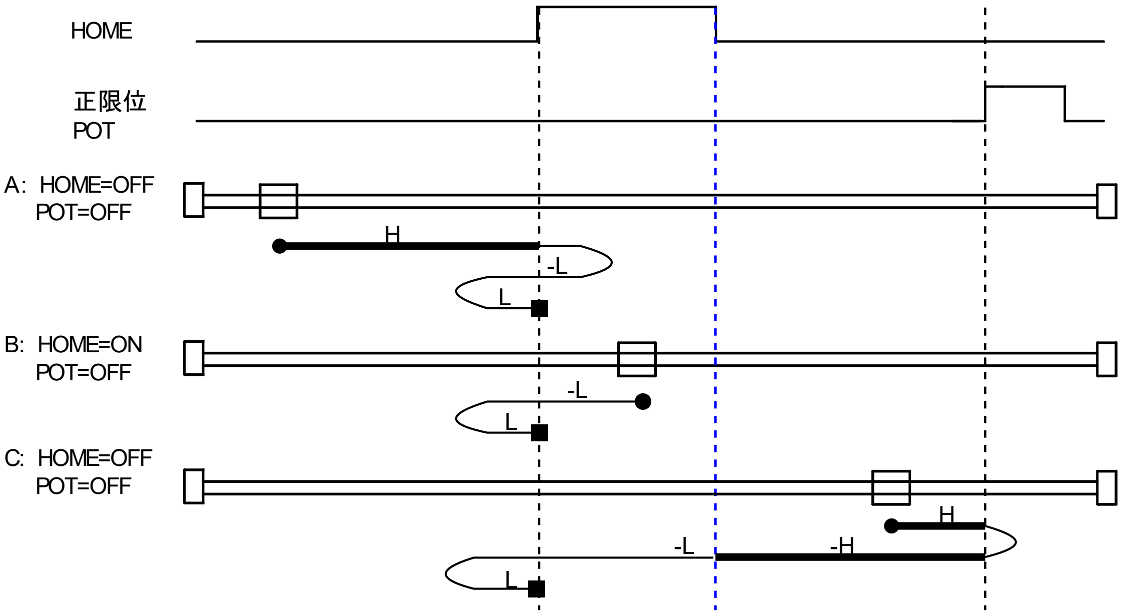
6.6.7. Method 23

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



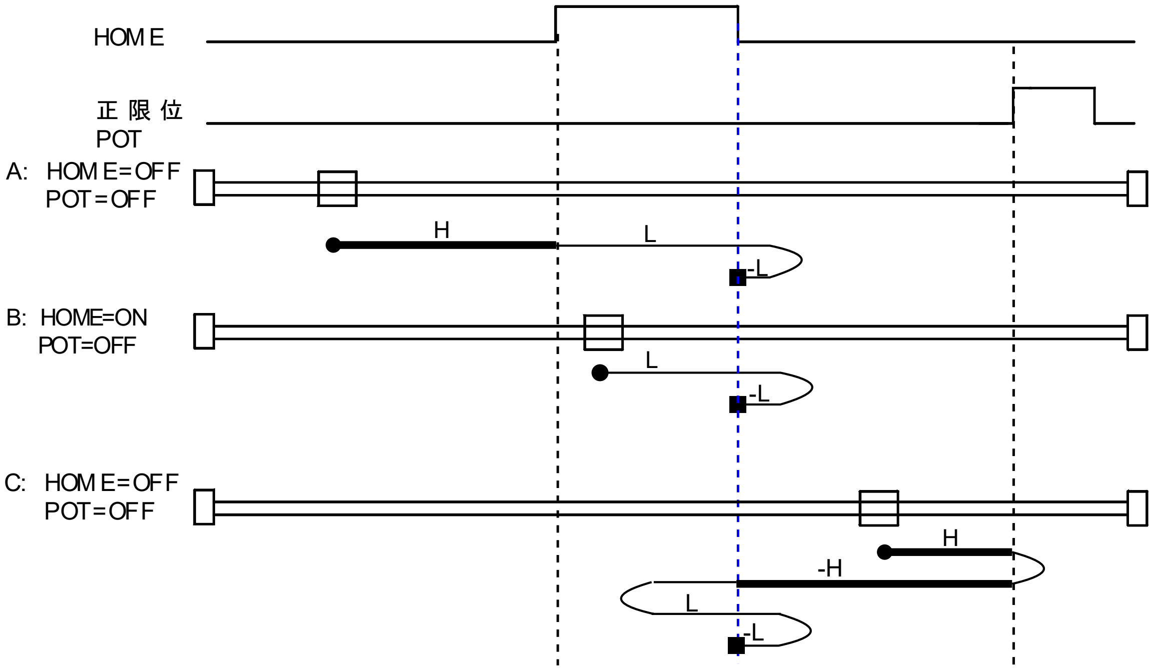
6.6.8. Method 24

● 起始点 ■ 结束停止点 H 高速 6099h - 01h L 低速 6099h - 02h

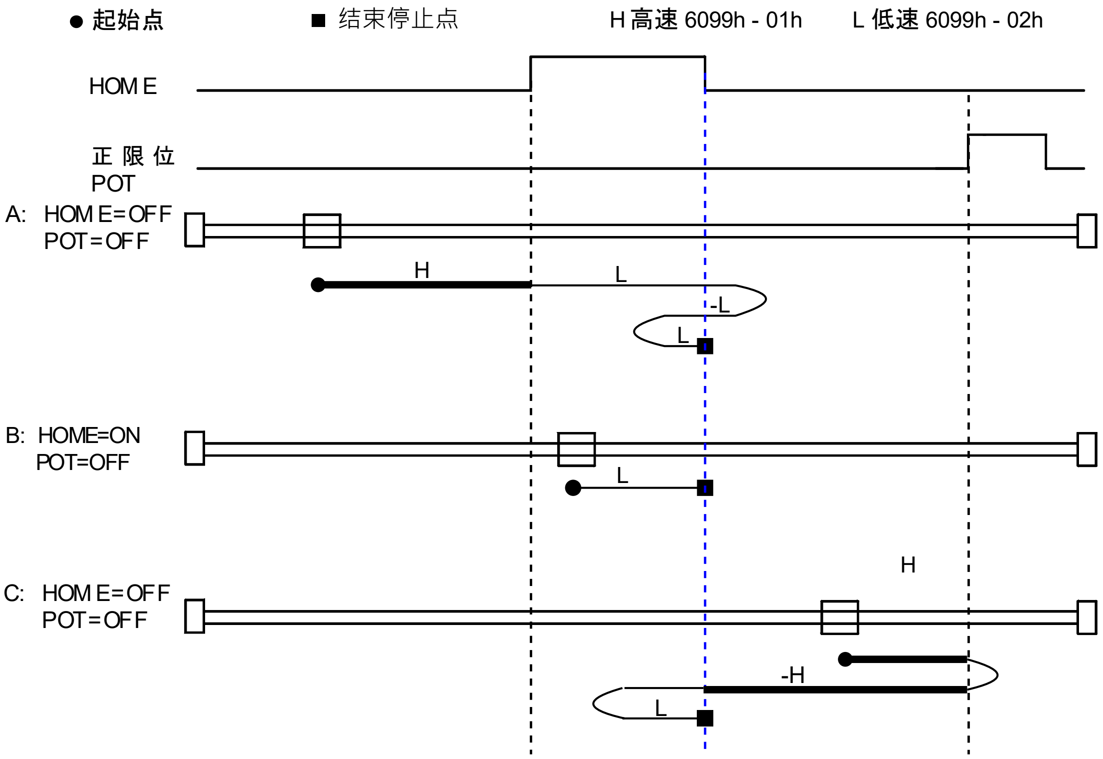


6.6.9. Method 25

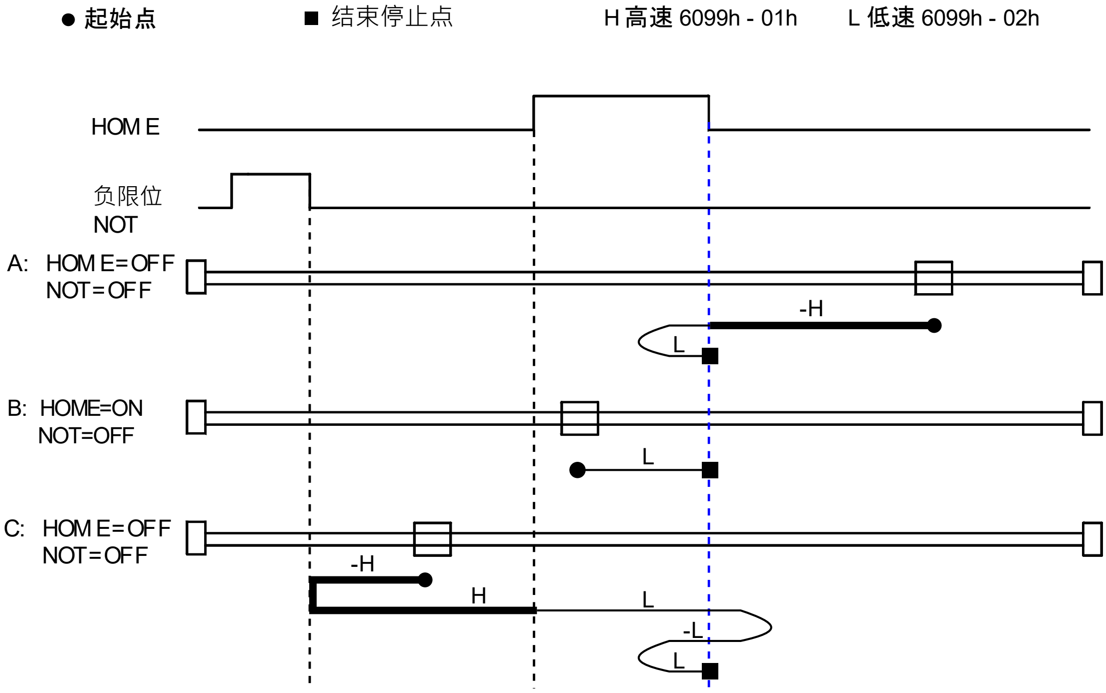
● 起始点 ■ 结束停止点 H 高速 6099h - 01h L 低速 6099h - 02h



6.6.10. Method 26

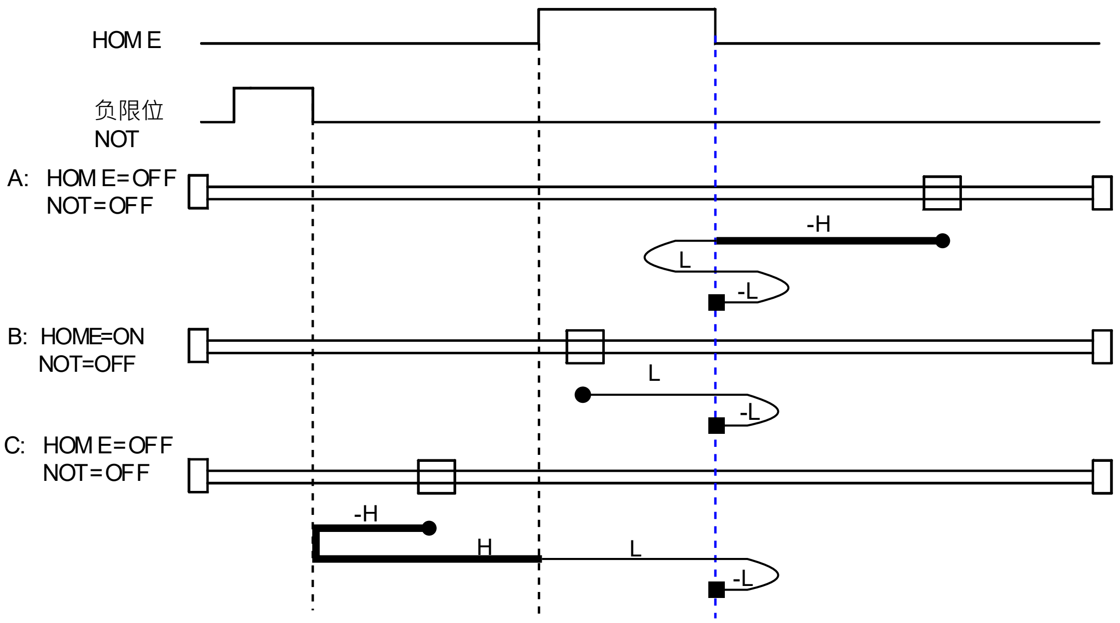


6.6.11. Method 27



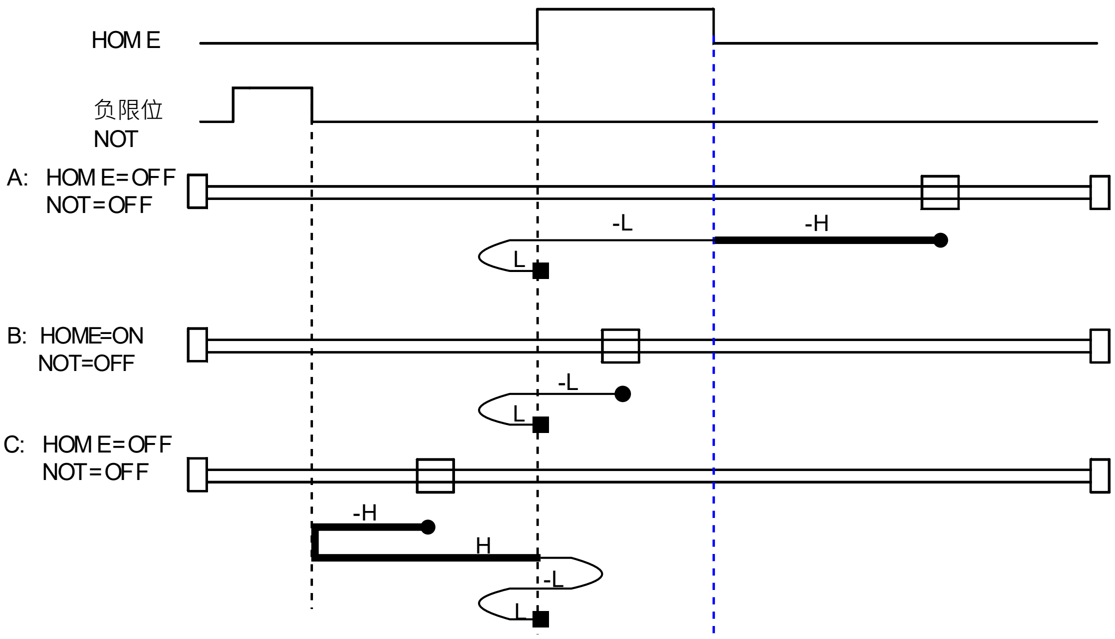
6.6.12. Method 28

● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h

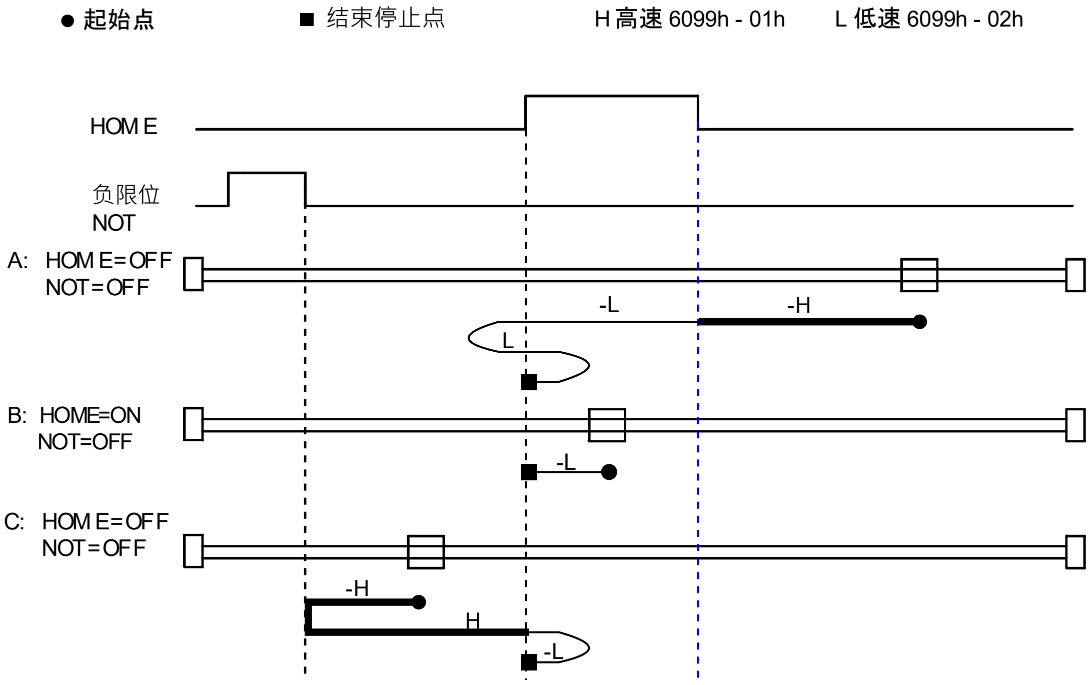


6.6.13. Method 29

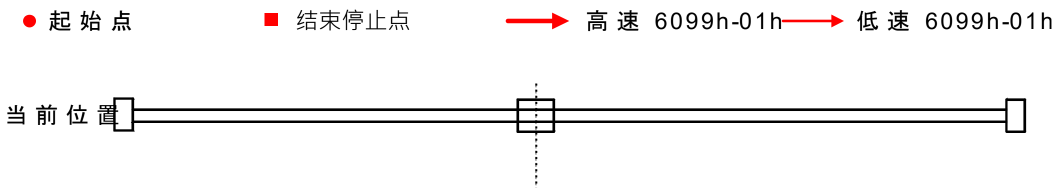
● 起始点　　■ 结束停止点　　H 高速 6099h - 01h　　L 低速 6099h - 02h



6.6.14. Method 30



6.6.15. Method 35



Appendix A Function Code Message Format

1. Function 03: Read and hold register

Query message:

QUERY Field Name	Example(Hex)
Slave address	01
Function code	03
Start address, high 8bit	00
Start address, low 8-bit	00
Data length, high 8bit	00
Data length, low 8-bit	05
CRC check, low 8-bit	85
CRC check, high 8-bit	C9

Response message:

RESPONSE Field Name	Example(Hex)
Slave address	01
Function code	03
Number of bytes returned	0A
Data, high (Register 40001)	00
Data, low (Register 40001)	00
Data, high (Register 40002)	00
Data, low (Register 40002)	01
Data, high (Register 40003)	00
Data, low (Register 40003)	00
Data, high (Register 40004)	00
Data, low (Register 40004)	03
Data, high (Register 40005)	FF
Data, low (Register 40005)	FF
CRC check, low 8-bit	C5
CRC check, high 8-bit	C6

2. Function 06: write to a single register

Query message: 01 06 00 12 00 00 29 CF

QUERY Field Name	Example(Hex)
Slave address	01
Function code	06
Address, high 8-bit	00
Address, low 8-bit	12
Data, high 8-bit	00
Data, low 8-bit	00
CRC check, low 8-bit	29
CRC check, high 8-bit	CF

Response message:

QUERY Field Name	Example(Hex)
Slave address	01
Function code	06
Address, high 8-bit	00
Address, low 8-bit	12
Data, high 8-bit	00
Data, low 8-bit	00
CRC check, low 8-bit	29
CRC check, high 8-bit	CF

3. Function 16(10 HEX): Write to multiple registers

Query message: 01 10 00 4B 00 04 08 00 64 00 64 02 58 01 F4 86 EC

QUERY Field Name	Example(Hex)
Slave address	01
Function code	10
Start address, high 8-bit	00
Start address, low 8-bit	4B
Data length, high 8-bit	00
Data length, low 8-bit	04
Number of bytes	08
Data high, (Register 40076)	00
Data, low (Register 40076)	64
Data, high (Register 40077)	00
Data, low (Register 40077)	64
Data, high (Register 40078)	02
Data, low (Register 40078)	58
Data, high (Register 40079)	01
Data, low (Register 40079)	F4
CRC check, low 8-bit	86
CRC check, high 8-bit	EC

Response message:

QUERY Field Name	Example(Hex)
Slave address	01
Function code	10
Start address, high 8-bit	00
Start address, low 8-bit	4B
Data length, high 8-bit	00
Data length, low 8-bit	04
CRC check, low 8-bit	B1
CRC check, high 8-bit	DC

Appendix B: Modbus/RTU Abnormal Response and Code

Response and code of NT60 when communication is abnormal

```
// exception code
#define ILLEGAL_FUNCTION 0x01
#define ILLEGAL_DATA_ADD 0x02
#define ILLEGAL_DATA_VAL  0x03
#define DEVICEFAIL            0x04
```

Appendix C: Modbus/RTU 16-bit CRC Check Routine

The CRC routine is written using the C language specification, making it easy for users to port to various platforms. The CRC_Checksum.c file contains two functions for calculating CRC.

Using computational CRC method:

```
    unsigned short CalcCRCbyAlgorithm(unsigned char* pDataBuffer, unsigned long
usDataLen)
    {
        /* Use the Modbus algorithm as detailed in the Watlow comms guide */

        const unsigned short POLYNOMIAL = 0xA001;
        unsigned short wCrc;
        int iByte, iBit;

        /* Initialize CRC */
        wCrc = 0xFFFF;

        for (iByte = 0; iByte < usDataLen; iByte++)
        {
            /* Exclusive-OR the byte with the CRC */
            wCrc ^= *(pDataBuffer + iByte);

            /* Loop through all 8 data bits */

            for (iBit = 0; iBit <= 7; iBit++)
            {
                /* If the LSB is 1, shift the CRC and XOR the polynomial mask with the CRC
*/

                /* Note - the bit test is performed before the rotation, so can't move the <<
here */
                if (wCrc & 0x0001)
                {
                    wCrc >>= 1;
                    wCrc ^= POLYNOMIAL;
                }
                else
                {
                    /* Just rotate it */
                    wCrc >>= 1;
                }
            }
        }

        return wCrc;
    }
```

CRC is calculated using table lookup:

```
/* Table Of CRC Values */
const unsigned short TABLE_CRC16[] =
{
    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,
    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
};

unsigned short CalcCRC_TAB(unsigned char* pDataBuffer, unsigned long usDataLen)
{
    unsigned char nTemp;
    unsigned short wCRCWord = 0xFFFF;

    while (usDataLen--)
    {
        nTemp = wCRCWord ^ *(pDataBuffer++);
        wCRCWord >>= 8;
        wCRCWord ^= TABLE_CRC16[nTemp];
    }

    return wCRCWord;
}
```