table of Contents

Catalog

1 communication specifications

Table 1-1 Description of performance parameters

	name	Description
EtherCAT Communication function	**Physical layer**	100BASE-TX
	Communication connector	RJ45 ×2
	Network Architecture	Concatenation
	transfer speed	2 ×100Mbps (full duplex)
	Maximum data length	1484 bytes
	SyncManager	SM0: MailBox output SM1: MailBox input SM2: Periodic output SM3 : Periodic data input
	FMMU (Bus Memory Management Unit)	FMMU0: Periodic data input area FMMU1: Periodic data output area FMMU2 : MailBox status area
	Application layer protocol	COE : CANOpen Over EtherCAT
	Synchronous mode	DC sync mode (SYNC0)
	Communication object	SDO: Service Data Object (non-periodic data) PDO: Process Data Object (Periodic Data) EMCY : Emergency
	Application layer specification	IEC61800-7 CIA402 Driver Profile
Supported CIA402 operating modes		Contour position mode Contour speed mode Contour torque mode Periodic sync position mode Cycle synchronous speed mode Periodic synchronous torque mode Zero return mode

2 wiring

2.1 Interface Information

The servo driver uses the dual RJ45 terminal as the EtherCAT protocol communication port, and the terminal interface is as shown below. The first interface (the upper interface) is the input interface, and the second interface (the lower interface) is the output interface.

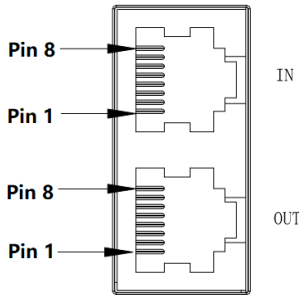


Figure 1.2 Pin Definition

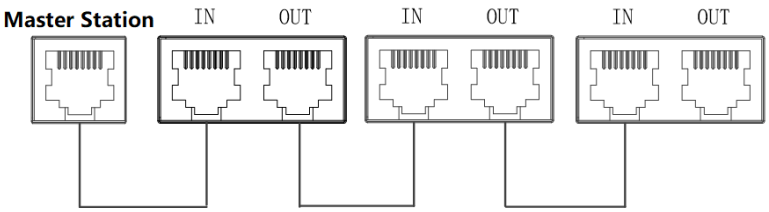
Table 2-1 Communication Signal Connector Pin Definitions

Pin number	name	Features
1	T X +	Receiver signal
2	T X -	Receiver signal
3	RX+	Send signal
4	--	--
5	--	--
6	R X -	Send signal
7	--	--
8	--	--
	shell	shield

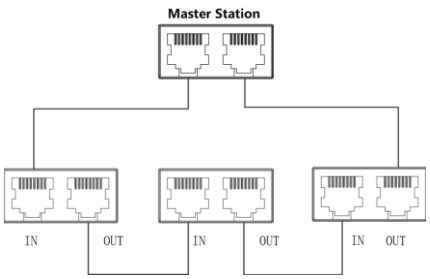
2.2 Topological connection

The SD700-EtherCAT servo drive topology communication connection is flexible and basically unlimited. The connection topology is as follows.

Linear connection:



Ring redundant connection:



2.3 communication cable

The EtherCAT communication cable uses an Ethernet Category 5 (100BASE-TX) network cable or a high-strength shielded network cable. currently using When using the SD700-EtherCAT servo drive, a shielded network cable is also required to enhance the anti-jamming capability.

2.4 EMC standard

The SD700-EtherCAT servo drive implements the IEC/EN61800-3:2004 (Adjustable speed electrical power drive systems-part3: EMC requirements and specific test methods) standard and the GB/t12668.3 national standards.

3 EtherCAT communication network settings

The EtherCAT communication network setting steps are as follows:

1. Import the XML file; (refer to the main site of the field application, determine the import method)
2. EtherCAT mode parameter setting; (refer to section 3.1)
3. EtherCAT communication parameter setting; (refer to Section 4.4)
4. Start the remote node. (You can observe section 3.2 to determine the current state)

3.1 Parameter setting of EtherCAT mode

In order to enable the servo driver to access the EthrCAT fieldbus network, it is necessary to set the relevant function code of the servo driver.

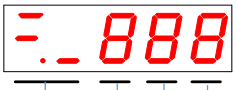
Table 3-3 System Settings Function Code Table

Pn000	**Function selection basic switch 0**	■	**Mailing address: 0x0000**
Factory default: 0000	**Setting range: 0000 ~ 001B**	**Unit: N/A**	**Control mode:**

Control mode selection	
0	Speed control mode
1	Analog speed mode
2	Torque mode
3	Internal speed
4	Internal speed <-> analog speed
5	Internal speed <-> position mode
6	Internal speed <-> torque mode
7	Position mode <-> analog speed
8	Position mode <-> torque mode
9	Torque mode <-> analog speed
10	Analog speed <-> zero speed mode
11	Position mode <-> pulse inhibit mode
16	**EtherCAT mode**

Note: When set to EtherCAT model, the control mode selection is invalid and the control mode is selected by the master station.

3.2 EtherCAT status monitoring



Control mode	
0	Reserved
1	Contour position mode (pp)
3	Contour speed mode (pv)
4	Contour torque mode (pt)
6	Zero return mode (hm)
8	Cycle sync position mode (csp)
9	Cycle Synchronous Speed Mode (csv)
10	Periodic synchronous torque mode (cst)

EtherCAT communication status	
1	Initialization status (I)
2	Pre-run status
4	Safe operating state
8	Operating status

Port connection status

Port 1 (In) connection status

Port 2 (Out) connection status

Status	Explanation	Status	Explanation
	On when the control power is ON, off when it is OFF		On when the main circuit power is ON, off when it is OFF
	Speed control : light up when speed is consistent output Position control: On when positioning is completed Torque control : always on		Lights when the rotation checkout (/TGON) output is on
			Speed control: light up when speed command is input Position control: lighted when the position command is being input
	On when servo is OFF, off when servo is ON		Torque control: light up when torque command is input Position control: light up when pulse clear signal is

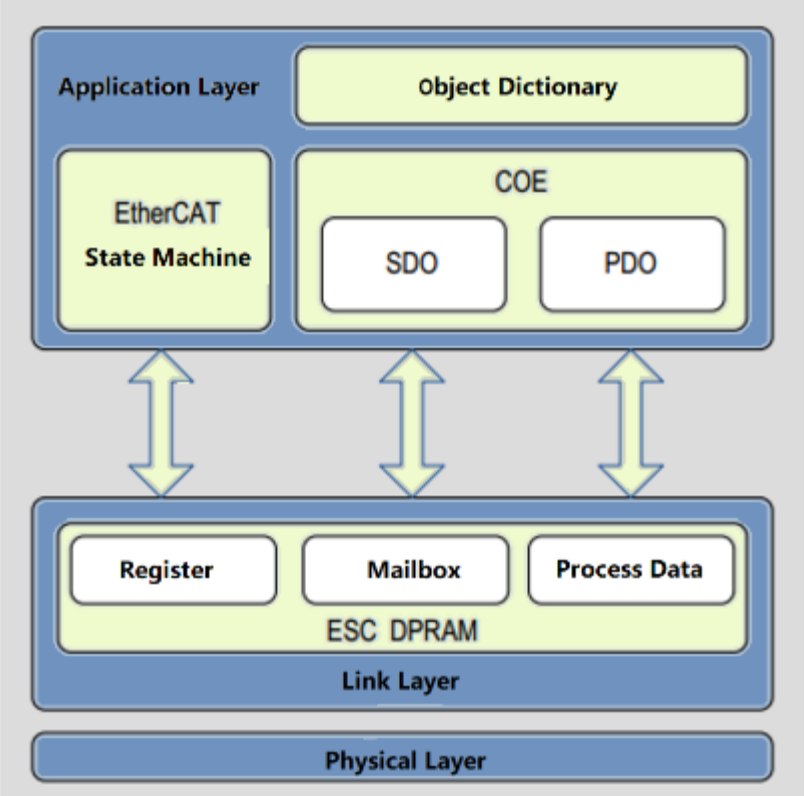
4. EtherCAT Communication Foundation

4.1 EtherCAT communication specification

Project		Specs
Communication protocol		IEC 61158 Type 12, IEC 61800-7 CiA 402 Drive Profile
Application layer	SDO	SDO request SDO response
	PDO	Variable PDO mapping
	CiA402	Contour position mode (pp) Contour velocity mode (pv) Contour torque mode (pt) Origin reset mode (hm) Synchronous periodic position mode (csp) Synchronous periodic velocity mode (csv) Synchronous periodic torque mode (cst)
Physical layer	Transport protocol	100BASE-TX (IEEE802.3)
	Maximum distance	100M
	Interface	RJ45 * 2 (INT、OUT)

4.2 Communication structure

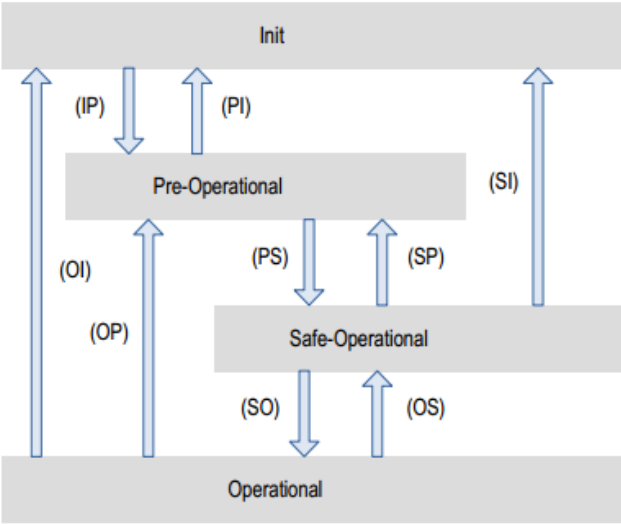
There are a variety of application layer protocols for using EtherCAT communication, but the SD700 servo driver described in this manual uses the IEC 61800-7 (CiA402)-CANopen motion control sub-protocol. The figure below shows the EtherCAT communication structure based on the CANopen application layer.



In the structure diagram, the application layer object dictionary contains communication parameters, service data objects (SDO), and process data objects (PDO). The PDO process data objects contain real-time data in the servo operation for periodic reading and writing. SDO mailbox communication is configured and accessed by aperiodic configuration of some communication parameter objects.

4.3 EtherCAT state machine

The following is a block diagram of the EtherCAT state transition:



The SD700-EtherCAT servo drive supports four state transitions and is responsible for coordinating the state transitions of the master and slave from initialization to runtime. The following table shows the relevant operations performed inside the initialization state to the running state servo.

State or state transition	Internal related operations
Initialization (I)	There is no communication on the application side, the master can only read and write the ESC register.
Initialize to pre-run conversion Init to Pre-Op (IP)	Master configuration slave station register Configure mailbox channel parameters Configuring distributed clock (DC) related registers The master writes a status control register to request a pre-run status
Pre-run (P)	Application layer mailbox data communication
Pre-run to convert to safe operation Pre-Op to Safe-Op (PS)	Master uses mailbox initialization process data mapping SM channel used by the master station configuration process data communication Master station configuration FMMU The master writes a status control register to request a safe operating state
Safe operation (S)	Application layer supports mailbox data communication There is process data communication, but only input data is allowed to be read, and no output signal is generated.
Safe running to run conversion Safe-Op to Op(SO)	The master sends valid output data The master writes a status control register to request a run status
Operating status (O)	Input and output are all valid Still using mailbox communication

4.4 Process Data

Real-time process data (PDO), following the producer - consumer model. The PDO can be divided into RPDO (Receive PDO) and TPDO (Transmit PDO). The slave receives the master command via RPDO and sends its own status information to the master via TPDO.

4.4.1 PDO Mapping Parameters

The PDO mapping is used to establish a mapping relationship between the object dictionary and the PDO. In the SD700-EtherCAT driver, 1600h~1603h is RPDO, and 1A00h~1A03h is TPDO. The following table shows information about the PDO mapping and mapping objects of the product, and the mapping objects can be changed.

PDO	index	Maximum number of maps	Longest mapped byte	Default mapping object
RPDO	1600h	8 Ge	32	6040h (control word) 607Ah (target location) 60FFh (target speed) 6071h (target torque) 6060h (operating mode)
	1601h	8 Ge	32	6040h (control word) 607Ah (target location)
	1602h	8 Ge	32	6040h (control word) 60FFh (target speed)
	1603h	8 Ge	32	6040h (control word) 6071h (target torque)
TPDO	1A00h	8 Ge	32	6041h (status word) 6064h (actual location) 606Ch (actual speed) 6077h (actual torque)
	1A01h	8 Ge	32	6041h (status word) 6064h (actual location)
	1A02h	8 Ge	32	6041h (status word) 6064h (actual location) 606Ch (actual speed)
	1A03h	8 Ge	32	6041h (status word) 6064h (actual location) 6077h (actual torque)

Note: It is recommended that the total number of RPDO and TPDO mapping bytes does not exceed 52 bytes, which may affect the servo performance.

4.4.2 PDO Synchronization Management Assignment Settings

In periodic data communication, the process data may include multiple PDO mapping objects. The CoE protocol uses the data objects 0x1C10~0x1C2F to define a list of PDO mapping objects of the corresponding synchronous management channels. Multiple PDOs may be mapped in different sub-indexes of 0x1C10~0x1C2F. In the SD700-EtherCAT servo drive, supports only. 1 th RPDO and. 1 th TPDO allocated in the following table:

Index	Subindex	Mapping assignment	Default map assignment
0x1C12	01h	Select one of 1600h~1603h as the RPDO application	1600h
0x1C13	01h	Select one of 1A00h~1A03h as the TPDO application	1A00h

4.4.3 PDO Configuration

The PDO mapping parameters (e.g., 1600h) contain pointers to the process data of the PDO that the PDO needs to send or receive, including the index of the mapping object, the sub-index, and the length of the object. The mapping parameter sub-index 0 records the number N of mapping objects of the PDO (for example, the maximum value of the SD700-EtherCAT servo N is 8), and one or more objects can be mapped at the same time. Sub-indexes 1 to 8 are mapping contents (mapping objects). The mapping parameter content is defined as follows.

Number of digits	31	.....	16	15	.....	8	7	.....	0
meaning	index			Subindex			Object length		

The index and the sub-index together determine the position of the object in the object dictionary. The length of the object indicates the specific bit length of the object, namely:

Object length	Bit length
08h	8 digits
10h	16 bits
20h	32 -bit

For example, the mapping parameter of object 6040h-00 is 60400010h.

The configuration process of PDO is as follows:

1. Cancel PDO. The 1C12h (or 1C13h) of 00h subindex write 0;
2. Clear the original mapping content. Write 0 to the 00h subindex of the mapping parameter (such as 1600h-00) to clear the original mapping content;
3. Write the PDO mapping content. Write the mapping parameter sub-index 1~N (N max is 8) according to the above definition;
4. Write the total number of PDO mapping objects. Step 3 The number of map N write mapping parameter subindex 0 (e.g. 1600h-00 write N);
5. Enable PDO. The 1C12h (or 1C13h) of 00h subindex write 1.

4.5 Mailbox Data SDO

The mailbox data SDO is used to transmit aperiodic data, such as the configuration of communication parameters, the configuration of servo drive operating parameters, and the like. In the SD700-EtherCAT servo drive, SDO requests and SDO are currently supported.

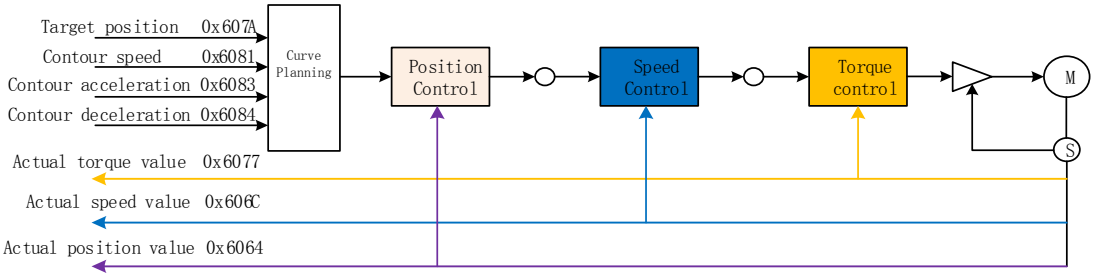
4.6 distributed clock

The distributed clock can make different servo devices use the same system clock, which can ensure that different servos receive instructions at the same time and execute instructions at the same time, which can achieve absolute time synchronization. The slave device can generate a synchronization signal based on the synchronization system time. The SD700-EtherCAT servo drive only supports DC synchronous mode. The synchronization period is controlled by SYNC0 and the period is 125us or an integer multiple of 250us.

5. Control mode

5.1 contour position mode (pp)

In the contour position mode, the master station sends a dictionary of related objects such as the required target position (absolute or relative), the speed of the position curve, acceleration and deceleration to the servo drive, and the servo driver generates a target curve based on the received related data and commands. instruction.



Related object dictionary

Control word 6040h		
Bit	**name**	**description**
0	Servo ready (Switch On)	0: invalid; 1: valid
1	Turn on the main circuit (Enable Voltage)	0: invalid; 1: valid
2	Fast stop (Quick Stop)	0: valid; 1: invalid
3	Servo operation (Enable Operation)	0: invalid; 1: valid
4	New target location (New Set-Point)	The rising edge triggers a new target position
5	Update Now (Change Set)	0: Not updated immediately; 1: Updated immediately
6	Absolute position command / relative position command (Abs/Rel)	0: The target position is an absolute position command 1: Target position is relative position command

Status word 6041h		
Bit	**name**	**Description**
10	Reaching the target (Target Reached)	0: The target location has not arrived 1: Target location arrives
12	Target location update (Set Point Acknowledge)	0: update target location 1: Cannot update the target location
13	Follow error (Following error)	0: No position deviation is too large 1: The position deviation is too large
15	Origin back to zero is completed (Home Find)	0: Zero return to origin is not completed 1: Home zero return is completed

index	Subindex	name	Read and write	type of data	unit	Predetermined area
0x603F	00	error code	RO	UINT 16	-	0~65535

0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10
0x6061	00	Mode display	RO	INT 8	-	0~10
0x6062	00	Position command	RO	INT32	Command unit	-
0x6063	00	Position feedback	RO	INT 32	Encoder unit	-
0x6064	00	Position feedback	RO	INT 32	Command unit	-
0x6065	00	Position deviation excessive threshold	RW	UINT 32	Command unit	$0\sim(2^{32}-1)$
0x6067	00	Location arrival threshold	RW	UINT 32	Command unit	$0\sim(2^{32}-1)$
0x6068	00	Location arrival window	RW	UINT 16	2ms	0~65535
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x6077	00	Actual torque	RO	UINT 16	1%	-
0x607A	00	target location	RW	INT 32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x6081	00	Contour speed	RW	UINT 32	Command unit /s	$0\sim(2^{32}-1)$
0x6083	00	Acceleration	RW	UINT 32	Command unit / S^2	$0\sim(2^{32}-1)$
0x6084	00	decrease speed	RW	UINT 32	Command unit / S^2	$0\sim(2^{32}-1)$
0x6091	01	Motor resolution	RW	UINT 32	-	$0\sim(2^{32}-1)$
	02	Axis resolution	RW	UINT 32	-	$1\sim(2^{32}-1)$
0x60E0	00	Forward torque limit	RW	UINT 16	0.1%	0~65535

0x60E1	00	Negative torque limit	RW	UINT 16	0.1%	0~65535
0x60F4	00	Position deviation	RO	INT 32	Encoder unit	-
0x60FC	00	Position command	RO	INT 32	Encoder unit	-

Curve planning - not immediately updated

1 The main station sends the relevant information of the position command to the slave station according to the need (acceleration time 6083h, deceleration time 6084h, contour speed 6081h, target position 607A);

② master station 6040h of bit4 set to 1 , the detected slave station 6040h of bit4 after the rising edge of the signal, for receiving the displacement new processing instruction.

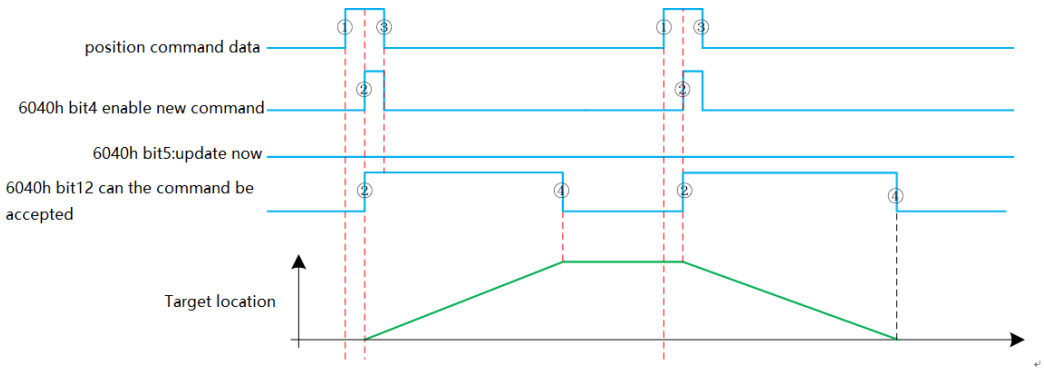
First, the station determines 6040h of bit5 whether 0, not 0 no correlation processing instruction information;

Next, the slave determines 6040h of bit5 is 0 , and 6041h of bit12 to 0 , from the station 6041h of bit12 is set to 1 , while executing the instruction ① related information. At this point, the slave is unable to continue to receive the new displacement command state.

③ the master station detects the status word 6041h of bit12 is set to 1 , the master station can release the displacement instruction data, the control word and 6040h of bit4 a 1 is set 0 .

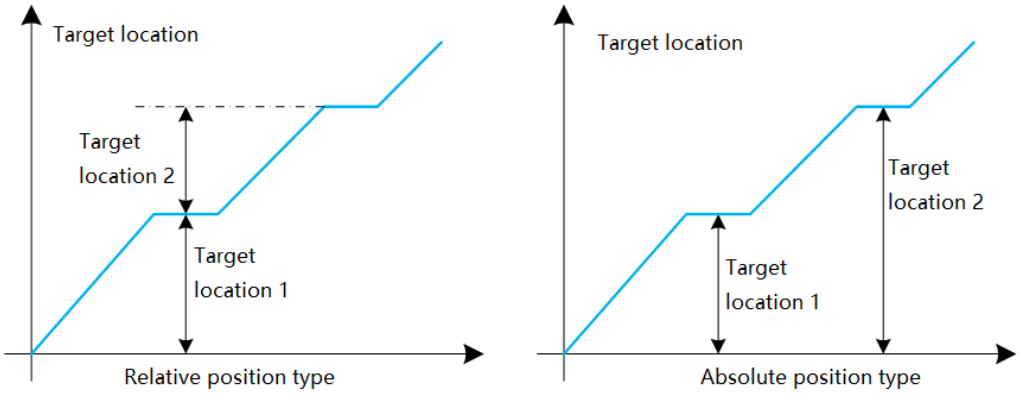
④ station detects a control word from 6040h to bit4 is 0 , positioning is completed after the current segment, the 6041h of bit12 is set 0 , indicating that the slave may receive a new shift command.

In the non-immediate update mode, the servo does not process the new displacement command while the current segment is running. The servo can receive and process the new displacement command only after the current segment positioning is completed.



Non-immediate update timing diagram

Control word 6040h of bit6 is 0, the absolute position of the control word 6040h to bit6 is. 1, the relative position.



Location type

Curve planning - update now

1 The main station sends the relevant information of the position command to the slave station according to the need (acceleration time 6083h, deceleration time 6084h, contour speed 6081h, target displacement 607A);

② master station 6040h of bit4 set to 1 , the detected slave station 6040h of bit4 after the rising edge of the signal, for receiving the displacement new processing instruction.

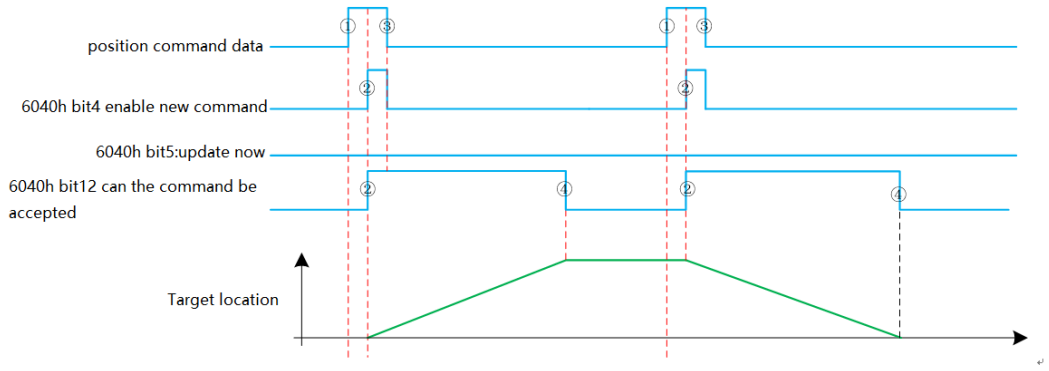
First, the station determines 6040h of bit5 whether one, not one is not related to the information processing instruction;

Next, the slave determines 6040h of bit5 is 1 , and 6041h of bit12 to 0 , from the station 6041h of bit12 is set to 1 , while executing the instruction ① related information. At this point, the slave is unable to continue to receive the new displacement command state.

③ the master station detects the status word 6041h of bit12 is set to 1 , the master station can release the displacement instruction data, the control word and 6040h of bit4 a 1 is set 0 .

④ detected from the station 6040h of bit4 a 1 is set to 0 , the then 6041h of bit12 is set to 0 . Indicates that the slave can receive a new displacement command.

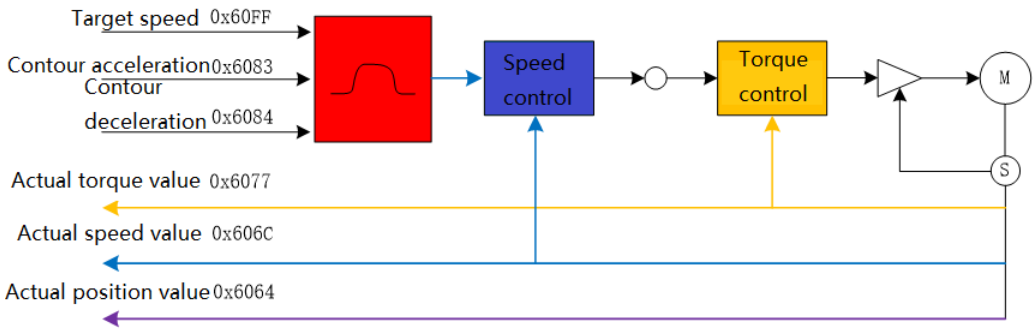
During the Immediate update mode, the current segment is running, is detected from the station to the 6040h of bit4 rising, while 604h of bit12 to 0, the servo displacement may receive a new command process.



Immediate update timing diagram

5.2 Contour speed mode (pv)

In the contour speed mode, the master sends the required target speed, acceleration time, and deceleration time to the servo drive, and the servo drive performs speed and torque adjustment.



Related object dictionary

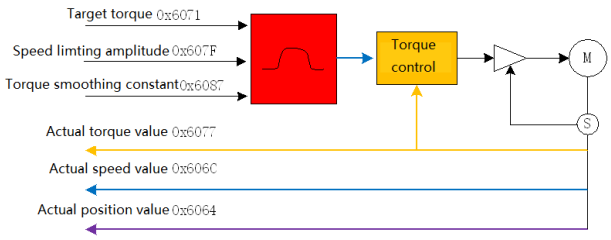
index	Subindex	name	Read and write	type of data	unit	Predetermined area
0x603F	00	error code	RO	UINT 16	-	-
0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10

0x6061	00	Mode display	RO	INT 8	-	0~10
0x6063	00	Position feedback	RO	INT32	Encoder unit	-
0x6064	00	Position feedback	RO	INT32	Command unit	-
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x6077	00	Actual torque	RO	UINT 16	1%	-
0x60E0	00	Forward torque limit	RW	UINT 16	0.1%	0~65535
0x60E1	00	Negative torque limit	RW	UINT 16	0.1%	0~65535
0x60FF	00	Target speed	RW	INT 32	Command unit / S	-2^{31}~$(2^{31}-1)$
0x607F	00	Maximum contour speed	RW	UINT 32	0.1rpm	0~$(2^{32}-1)$
0x6083	00	Acceleration	RW	UINT 32	Command unit / S^2	0 ~ $(2^{32}-1)$
0x6084	00	decrease speed	RW	UINT 32	Command unit / S^2	0 ~ $(2^{32}-1)$

Note: The speed limit value is determined by the smaller of 0x607F and the maximum motor speed.

5.3 Profile Torque Mode (pt)

In the contour torque mode, the primary station transmits the target torque command 6071h, the torque ramp constant 6087h, and the speed limit value 607Fh to the servo drive, which is executed internally by the servo driver. When the speed reaches the limit value of 607Fh, it will enter the speed regulation phase.



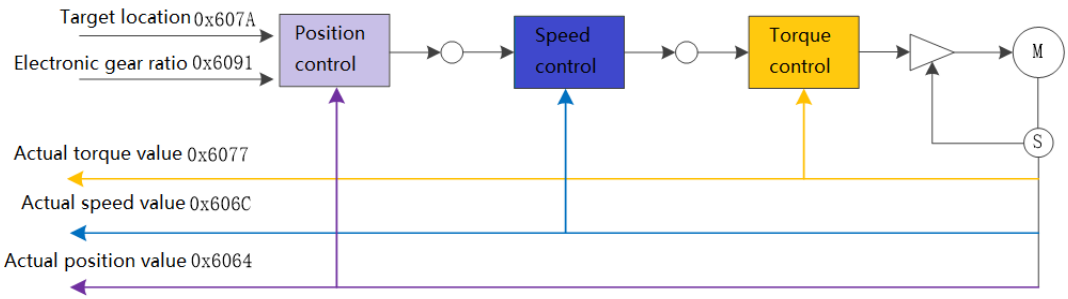
Related object dictionary

index	Subindex	Name	Read and write	type of data	unit	Predetermined area
0x603F	0x00	error code	RO	UINT 16	-	0~65535
0x6040	0x00	Control word	RW	UINT 16	-	0~65535

0x6041	0x00	Status word	RO	UINT 16	-	0~65535
0x6060	0x00	Operating mode	RW	INT 8	-	0~10
0x6061	0x00	Mode display	RO	INT 8	-	0~10
0x606C	0x00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x6071	0x00	Target torque	RW	INT 16	0.1%	-3000~3000
0x6074	0x00	Torque command	RO	INT 16	1%	-
0x6077	0x00	Actual torque	RO	UINT 16	1%	-
0x607F	0x00	Maximum contour speed	RW	UINT 32	0.1rpm	0-50000
0x6087	0x00	Torque ramp time	RW	UINT 32	Ms	0-(2^{32}-1)
0x60E0	00	Forward torque limit	RW	UINT 16	0.1%	0~65535
0x60E1	00	Negative torque limit	RW	UINT 16	0.1%	0~65535

5.4 cycle synchronous position mode (csp)

In the cyclic synchronous position mode, the primary station transmits the planned target position 607Ah to the servo drive in a periodically synchronized manner according to the value of the target speed 60FFh. Position, speed and torque control are performed inside the servo drive.



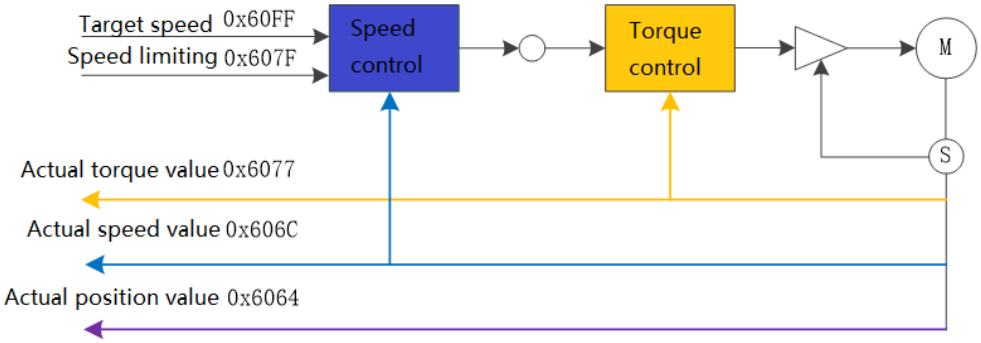
Related object dictionary

index	Subindex	Name	Read and write	type of data	unit	Predetermined area

0x603F	00	error code	RO	UINT 16	-	0~65535
0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10
0x6061	00	Mode display	RO	INT 8	-	0~10
0x6062	00	Position command	RO	INT 32	Command unit	-
0x6063	00	Position feedback	RO	INT 32	Encoder unit	-
0x6064	00	Actual position feedback	RO	INT 32	Command unit	-
0x6065	00	Position deviation excessive threshold	RW	UINT 32	Command unit	$0\sim(2^{32}-1)$
0x6067	00	Location arrival threshold	RW	UINT 32	Command unit	$0\sim(2^{32}-1)$
0x6068	00	Location arrival window time	RW	UINT 16	2ms	0~65535
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x607A	00	target location	RW	INT 32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x6091	01	Motor resolution	RW	UINT 32	-	$0\sim(2^{32}-1)$
	02	Axis resolution	RW	UINT 32	-	$1\sim(2^{32}-1)$
0x60B0	00	Position offset	RW	INT 32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x60F4	00	Position deviation	RO	INT 32	Encoder unit	-
0x60FC	00	Position command	RO	INT 32	Encoder unit	-

5.5 cycle synchronous speed mode (csv)

In the cycle synchronous speed mode, the master station sends the calculated target speed 60FFh periodically to the servo drive, and the speed and torque adjustment are internally adjusted by the servo.



Related object dictionary

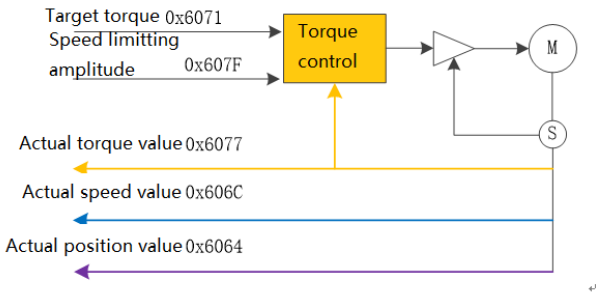
index	Subindex	Name	Read and write	type of data	unit	Predetermined area
0x603F	00	error code	RO	UINT 16	-	0~65535
0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10
0x6061	00	Mode display	RO	INT 8	-	0~10
0x6063	00	Position feedback	RO	INT 32	Encoder unit	-
0x6064	00	Actual position feedback	RO	INT 32	Command unit	
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x6077	00	Actual torque	RO	INT 16	1%	-
0x607F	00	Maximum speed	RW	UINT 32	0.1r/min	0~50000
0x60E0	00	Forward torque limit	RW	UINT 16	0.1%	0~65535
0x60E1	00	Negative torque limit	RW	UINT 16	0.1%	0~65535
0x6091	01	Motor resolution	RW	UINT 32	-	0~(2^{32}-1)

	02	Axis resolution	RW	UINT 32	-	1~(2^{32}-1)
0x60FF	00	Target speed	RW	INT 32	Command unit /s	-2^{31}~(2^{31}-1)

Note: The speed limit value is determined by the smaller of 0x607F and the maximum motor speed.

5.6 cycle synchronous torque mode (cst)

In the cyclic synchronous torque mode, the master station periodically synchronizes the calculated target torque 6071h to the servo drive, and the torque adjustment is performed internally by the servo. When the speed reaches the limit value, it enters the speed regulation phase.



Related object dictionary

index	Subindex	Name	Read and write	type of data	unit	Predetermined area
0x603F	00	error code	RO	UINT 16	-	0~65535
0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10
0x6061	00	Mode display	RO	INT 8	-	0~10
0x6064	00	Actual position feedback	RO	INT 32	Command unit	-
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-
0x6071	00	Target torque	RW	INT 16	0.1%	-3000~3000
0x6077	00	Actual torque	RO	INT 16	1%	-
0x607F	00	Maximum speed	RW	UINT 32	0.1r/min	0~(2^{32}-1)

0x60E0	00	Forward torque limit	RW	UINT 16	0.1%	0~65535
0x60E1	00	Negative torque limit	RW	UINT 16	0.1%	0~65535

Note: The speed limit value is determined by the smaller of 0x607F and the maximum motor speed.

5.7 zero return mode (hm)

The zero return mode is used to find the mechanical origin and locate the positional relationship between the mechanical origin and the mechanical zero.

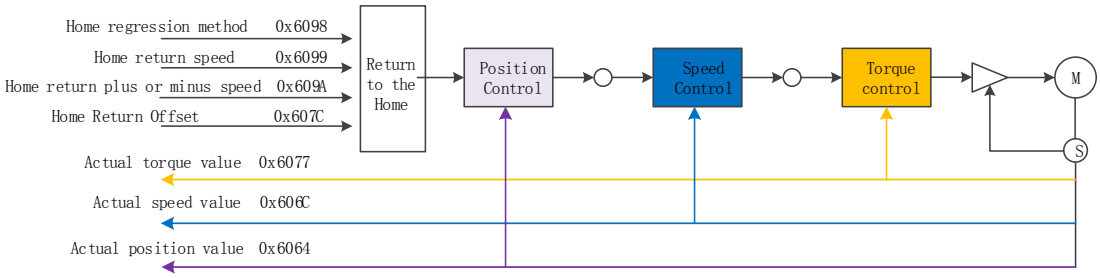
Mechanical origin: A fixed position on the machine that corresponds to a certain origin signal switch.

Mechanical origin = mechanical zero + 607C (origin offset)

Mechanical zero point: Absolute 0 position on the machine.

After the servo drive returns to zero at the origin, the motor stops at the machine origin, and the positional relationship between the machine origin and the mechanical zero is adjusted by setting the value of the object dictionary 0x607C.

Zero return control block diagram



Related object dictionary

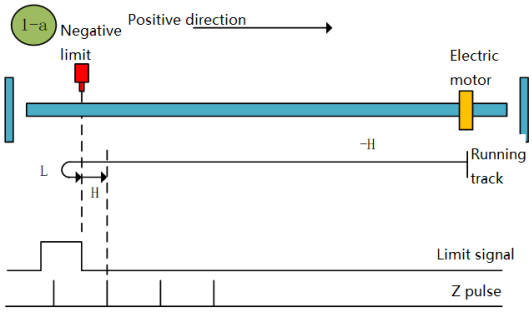
index	Subindex	name	Read and write	type of data	unit	Predetermined area
0x603F	00	error code	RO	UINT 16	-	0~65535
0x6040	00	Control word	RW	UINT 16	-	0~65535
0x6041	00	Status word	RO	UINT 16	-	0~65535
0x6060	00	Operating mode	RW	INT 8	-	0~10
0x6061	00	Mode display	RO	INT 8	-	0~10
0x6062	00	Command position	RO	INT 32	Command unit	-
0x6064	00	Actual position feedback	RO	INT 32	Command unit	-
0x606C	00	Actual speed feedback	RO	INT 32	Command unit /s	-

0x6067	00	Position arrival threshold	RO	UINT 32	Command unit	-
0x6068	00	Location arrival window	RW	UINT 16	2ms	-
0x6077	00	Actual torque	RO	INT 16	1%	-
0x6098	00	Origin return method	RW	INT8	-	1~35
0x6099	01	High speed search deceleration point	RW	UINT 32	Command unit /s	0~(2^{32}-1)
	02	Search origin low speed	RW	UINT 32	Command unit /s	1~(2^{32}-1)
0x609A	00	Acceleration	RW	UINT 32	Command unit / s^2	0~(2^{32}-1)
0x60F4	00	Position deviation	RO	INT 32	Encoder unit	-

Introduction to the zero return method:

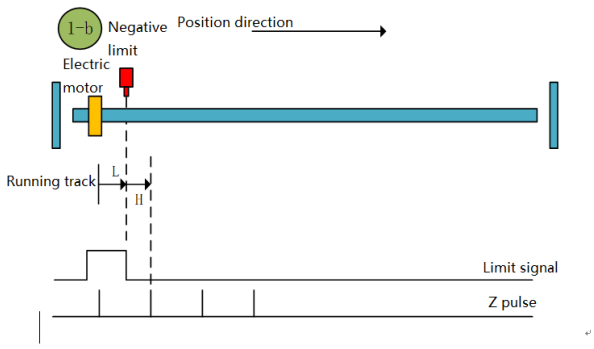
Zero return mode 1 (6098 00h=1)

a Start zero return → Reverse high speed to find negative limit → Hit negative limit rising edge → Deceleration to 0 → Forward low speed to find negative limit falling edge → Forward to find Z pulse



Zero return method 1-a

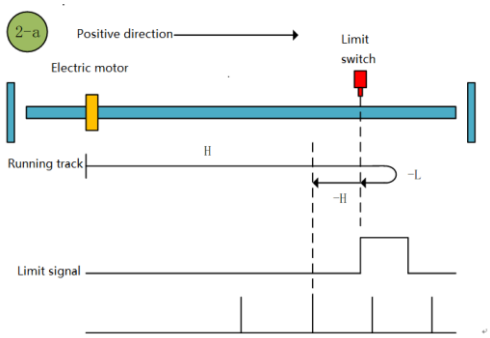
b Start origin return → Negative limit valid → Positive low speed find negative limit falling edge → Forward find Z pulse



Zero return method 1-b

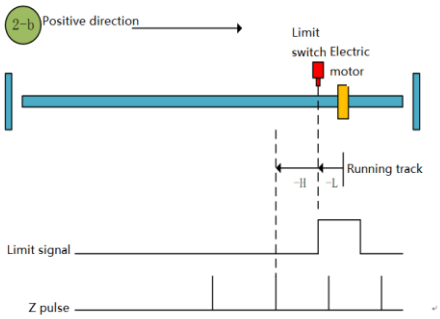
Zero return mode 2 (6098 00h = 2)

a Start zero return → Forward high speed correction limit → Hit the positive limit rising edge → Decelerate to 0 → Reverse low speed correction limit falling edge → Reverse find Z pulse



Zero return method 2-a

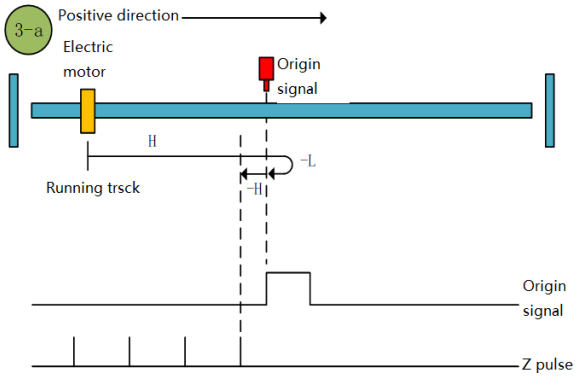
b Start the origin return → positive limit valid → reverse low speed correction limit falling edge → reverse find Z pulse



Origin return mode 2-b

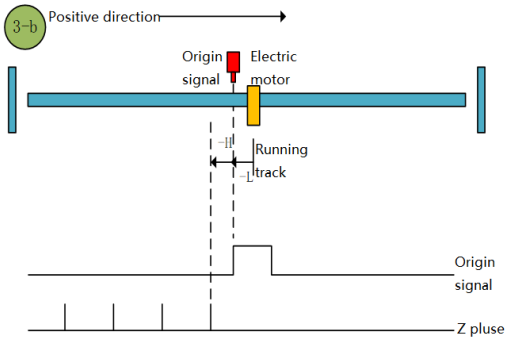
Zero return mode 3 (6098 00h = 3)

a. Start origin return to zero → Origin signal is OFF → Forward high speed finds the origin signal rising edge → Deceleration to 0 → Reverse low speed finds the origin signal falling edge → Reverse finds Z pulse



Origin return mode 3-a

b. Start origin return to zero → origin signal ON → reverse low speed to find the origin falling edge → reverse find Z pulse

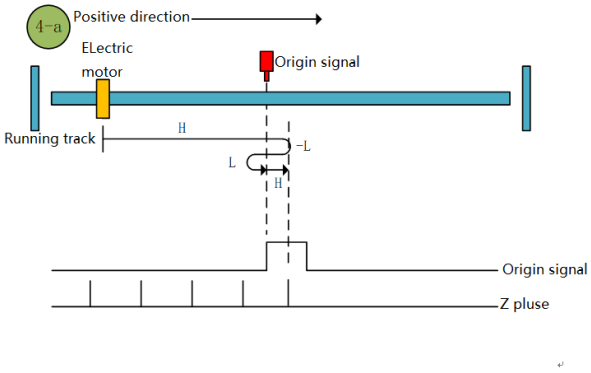


Origin return mode 3-b

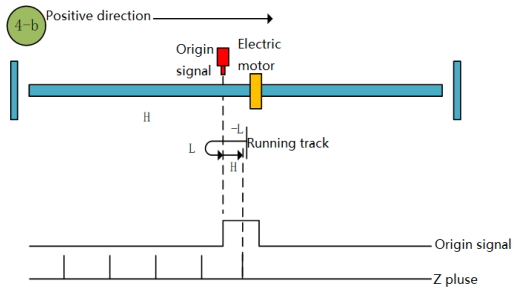
Zero return mode 4 (6098 00h = 4)

a. Start zero return → Origin signal OFF → Forward high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Forward to find Z pulse.

b. Start the origin return → origin signal ON → reverse low speed to find the origin falling edge → positive low speed to find the origin rising edge → forward find Z pulse



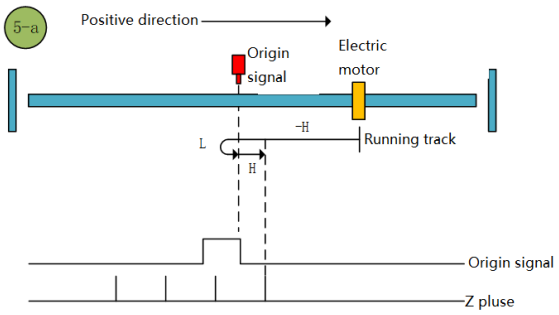
Origin return mode 4-a



Origin return mode 4-b

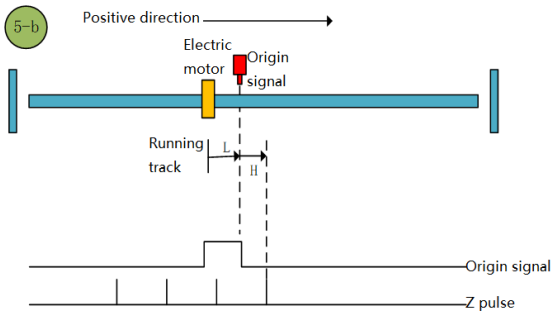
Zero return mode 5 (6098 00h = 5)

a. Start origin return to zero → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Forward to find Z pulse



Zero return method 5-a

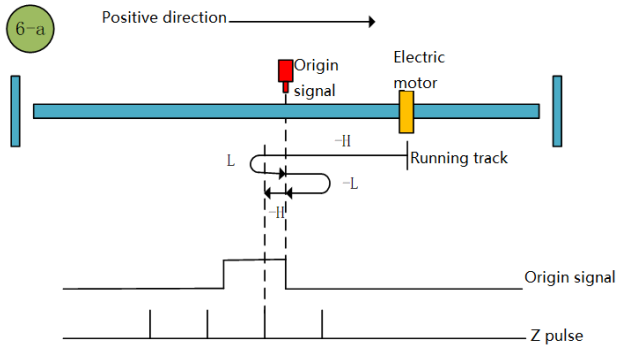
b. Start origin return to zero → origin signal ON → forward low speed to find the origin falling edge → forward find Z pulse



Origin return mode 5-b

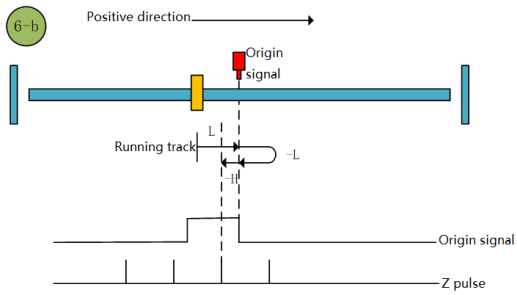
Origin return mode 6 (6098 00h = 6)

a. Start OPR → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Reverse low speed to find the origin rising edge → Reverse to find Z pulse



Origin return mode 6-a

b. Start the origin return → the origin signal ON → the forward low speed to find the origin falling edge → the reverse low speed to find the origin rising edge → reverse to find the Z pulse



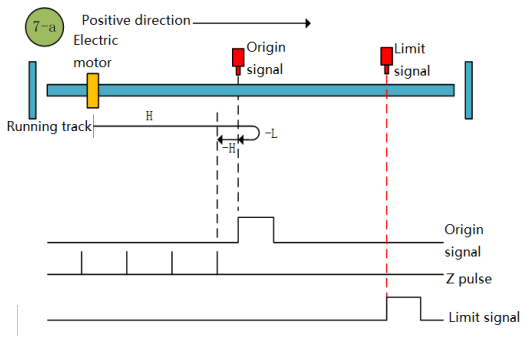
Origin return mode 6-b

Zero return mode 7 (6098 00h = 7)

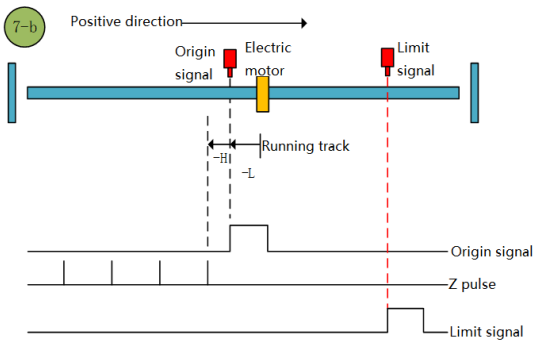
a. Start zero return → Origin signal OFF → Forward high speed to find the origin rising edge → Deceleration to 0 → Reverse low speed to find the origin falling edge → Reverse to find Z pulse

b. Start zero return → origin signal ON → reverse low speed to find the origin falling edge → reverse find Z pulse

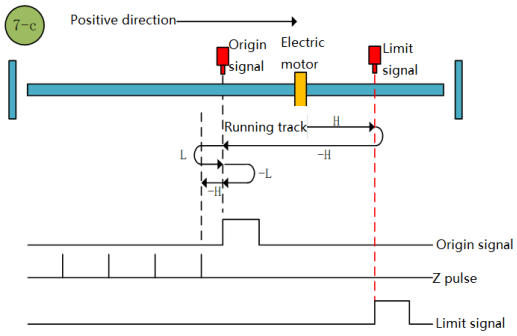
c. Start zero return → origin OFF → forward high speed to find the origin rising edge → hit the positive limit → reverse high speed to find the origin falling edge → decelerate to 0 → forward low speed to find the origin rising edge → reverse low speed to find the origin Find Z pulse along → reverse



Zero return method 7-a



Origin return mode 7-b



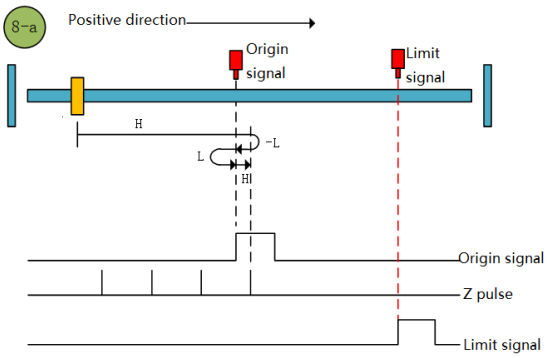
Origin return mode 7-c

Origin return mode 8 (6098 00h = 8)

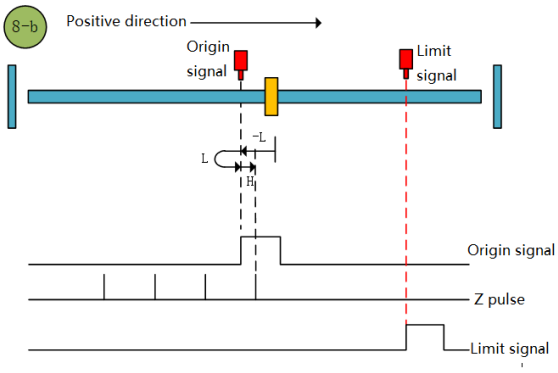
a. begins zeroing origin point signal → OFF → positive → rising speed seek decelerate to the origin 0 → → reverse slow to find the falling edge of positive origin to find the origin of the rising edge of the low-speed forward looking → Z pulse

b. Start zero return → origin signal ON → reverse low speed to find the origin falling edge → positive low speed to find the origin rising edge → forward find Z pulse

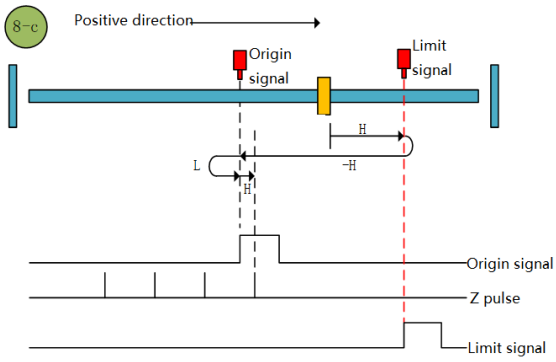
c. Start zero return → origin OFF → forward high speed to find the origin rising edge → hit the positive limit → reverse high speed to find the origin falling edge → decelerate to 0 → forward low speed to find the origin rising edge → forward find Z pulse



Origin return mode 8-a



Origin return mode 8-b



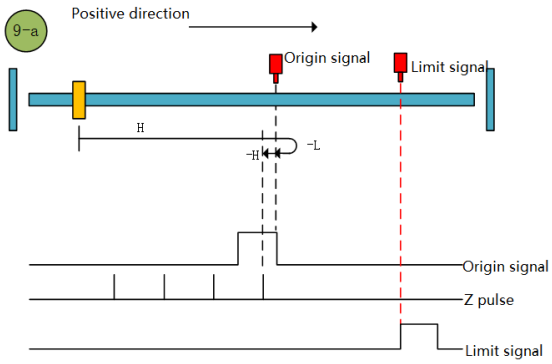
Origin return mode 8-c

Zero return mode 9 (6098 00h = 9)

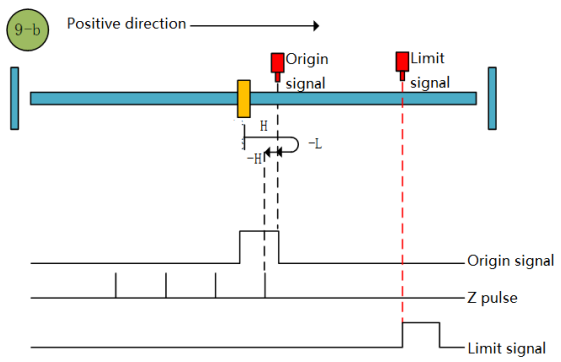
a. Start zero return → Origin signal OFF → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge → Reverse to find Z pulse

b. Start zero return → origin signal ON → forward high speed to find the origin falling edge → decelerate to 0 → reverse low speed to find the origin rising edge → reverse find Z pulse

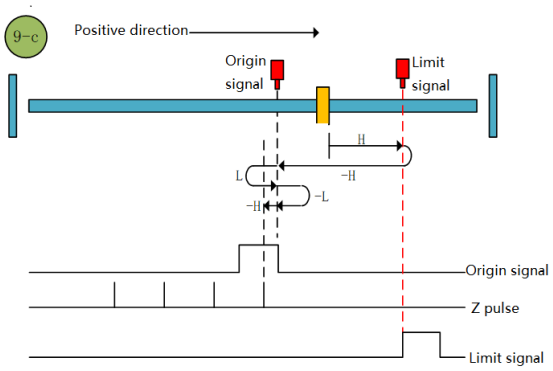
c. Start origin return → origin OFF → forward high speed to find the origin falling edge → hit the positive limit → reverse high speed to find the origin rising edge → decelerate to 0 → forward low speed to find the origin falling edge → reverse low speed to find the origin rising edge →Reverse looking for Z pulse



Zero return method 9-a



Zero return method 9-b



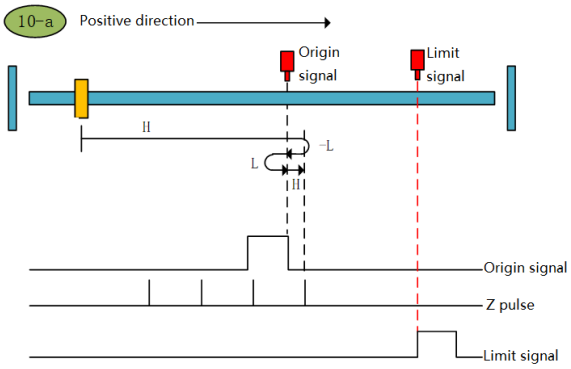
Zero return method 9-c

Zero return mode of origin 10 (6098 00h = 10)

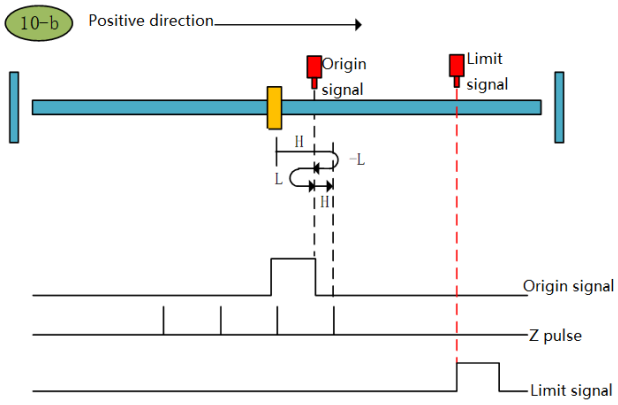
a. Start origin return → Origin signal OFF → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge → Forward low speed to find the origin falling edge → Forward to find Z pulse

b. Start origin return → origin signal ON → forward high speed to find the origin falling edge → decelerate to 0 → reverse low speed to find the origin rising edge → positive low speed to find the origin falling edge → forward find Z pulse

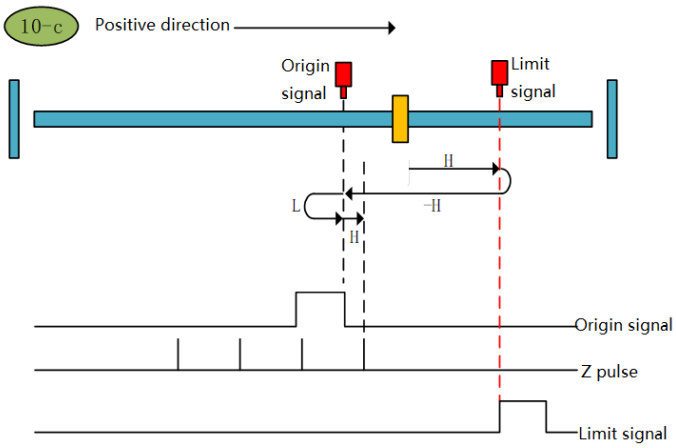
c. Start zero return → origin OFF → forward high speed to find the origin falling edge → hit the positive limit → reverse high speed to find the origin rising edge → decelerate to 0 → forward low speed to find the origin falling edge → forward find Z pulse



Zero return method 10-a



Zero return method 10-b



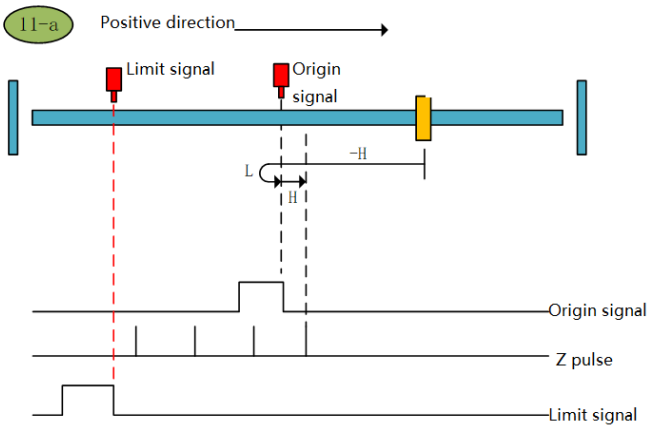
Zero return method of origin 0 -c

Zero return mode of origin (6098 00h = 11)

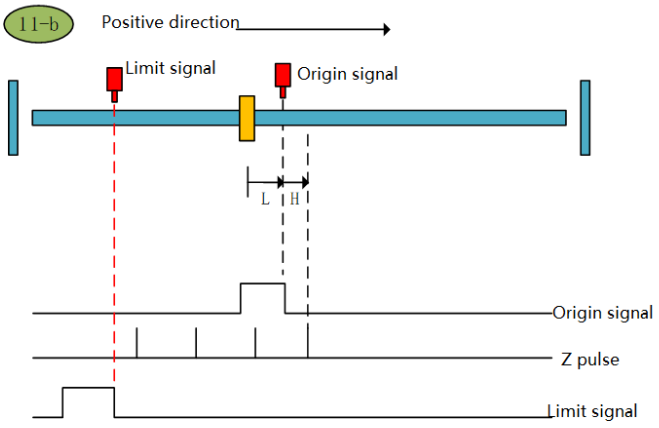
a. Origin return to zero → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Forward to find Z pulse

b. Origin return to zero → Origin signal ON → Forward low speed to find the origin falling edge → Forward to find Z pulse

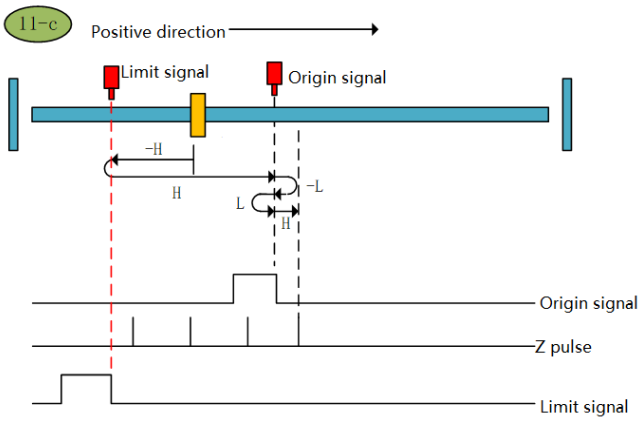
c. Origin return to zero → Origin signal OFF → Reverse high speed to find the origin rising edge → Hit the negative limit → Forward high speed to find the origin signal falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge → Forward to find Z pulse



Origin return mode 11-a



Origin return mode 11-b



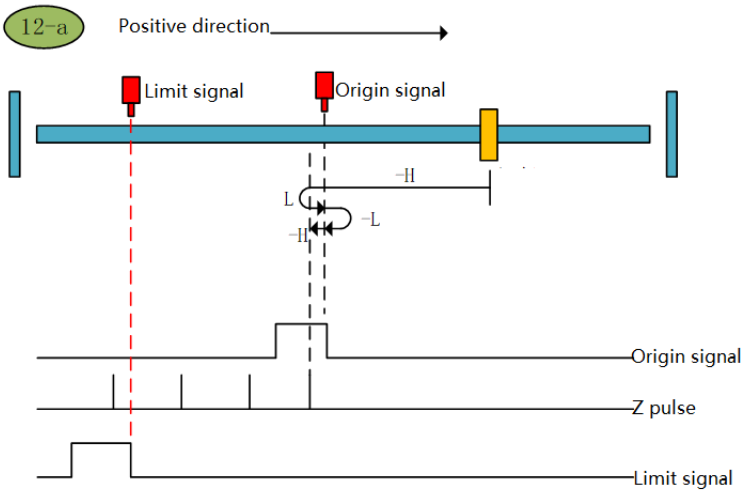
Origin return mode 11-c

Zero return mode of origin 12 (6098 00h = 12)

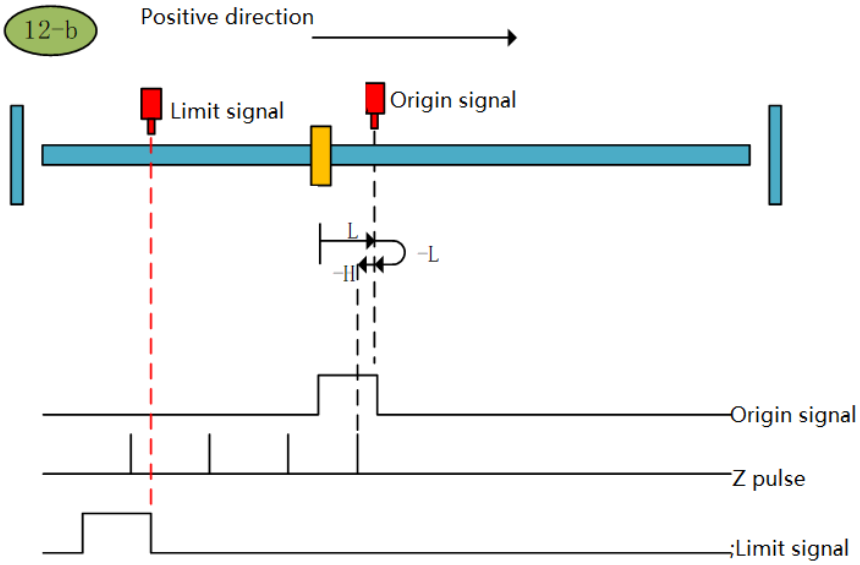
a. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Reverse low speed to find the origin rising edge → Reverse to find the Z pulse

b. Origin return start → Origin signal ON → Forward low speed find origin falling edge → Reverse low speed find origin rising edge → Reverse find Z pulse

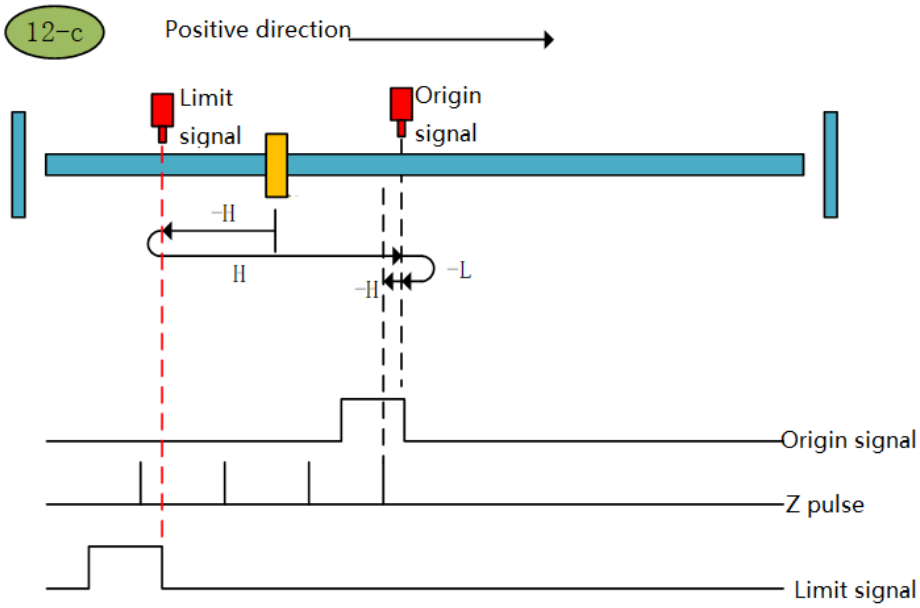
c. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Hit the negative limit → Forward high speed to find the origin signal falling edge → Decelerate to 0 → Reverse low speed find the origin rising edge → Reverse find Z pulse



Zero return method 12-a



Zero return method 12-b



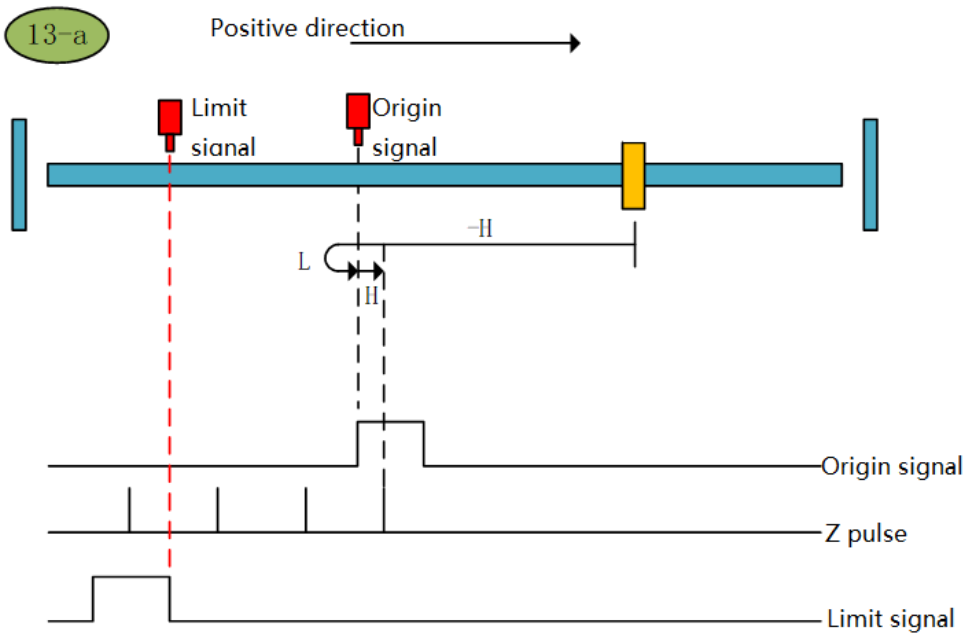
Zero return method 12-c

Zero return method of origin (6098 00h = 13)

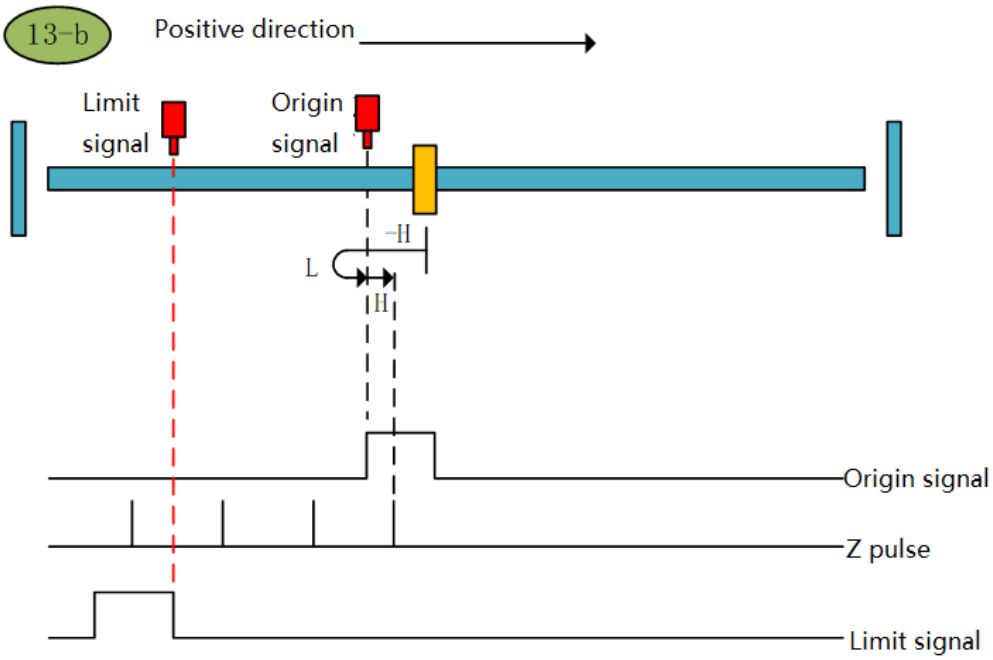
a. Origin return to zero → Origin signal OFF → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Forward to find Z pulse

b. Origin return to zero → Origin signal ON → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Forward to find Z pulse

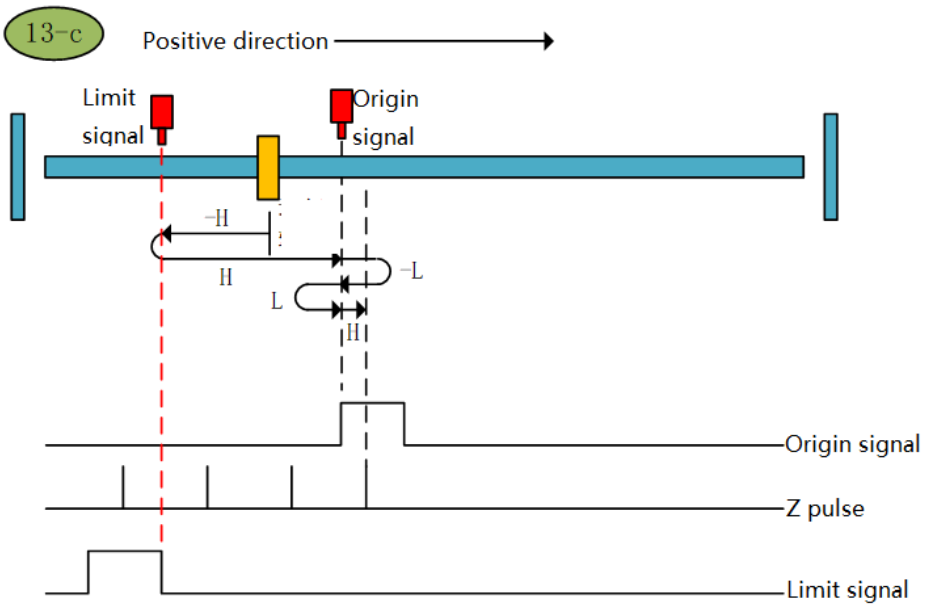
c. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Hit the negative limit → Forward high speed to find the origin signal rising edge → Decelerate to 0 → Reverse low speed to find the origin signal falling edge → Forward low speed Origin signal rising edge → positive looking Z pulse



Origin return mode 13-a



Origin return mode 13-b



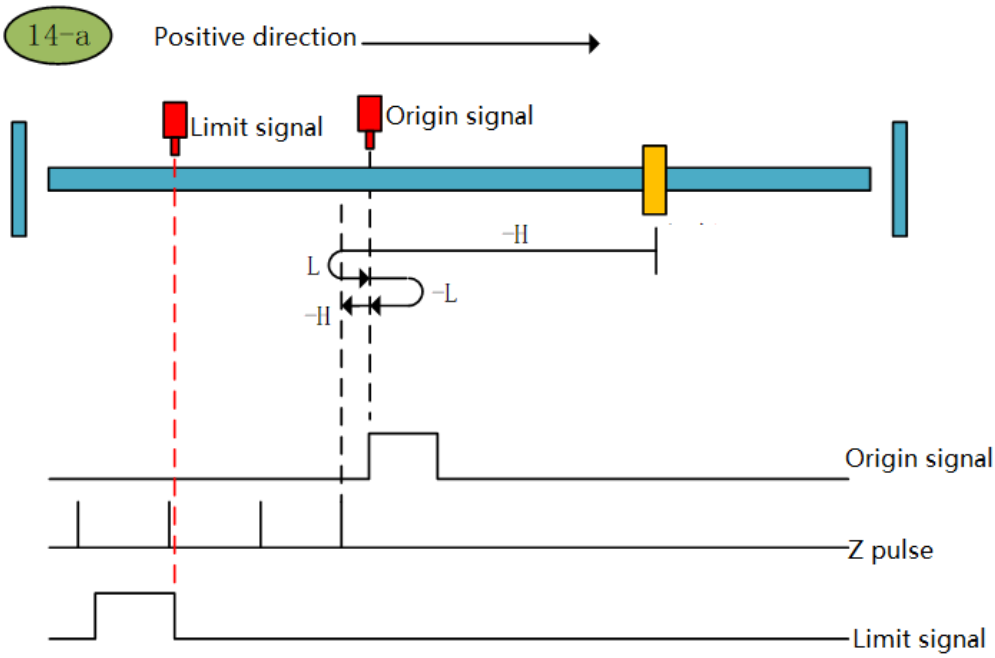
Origin return mode 13-c

Zero return method of origin (6098 00h = 14)

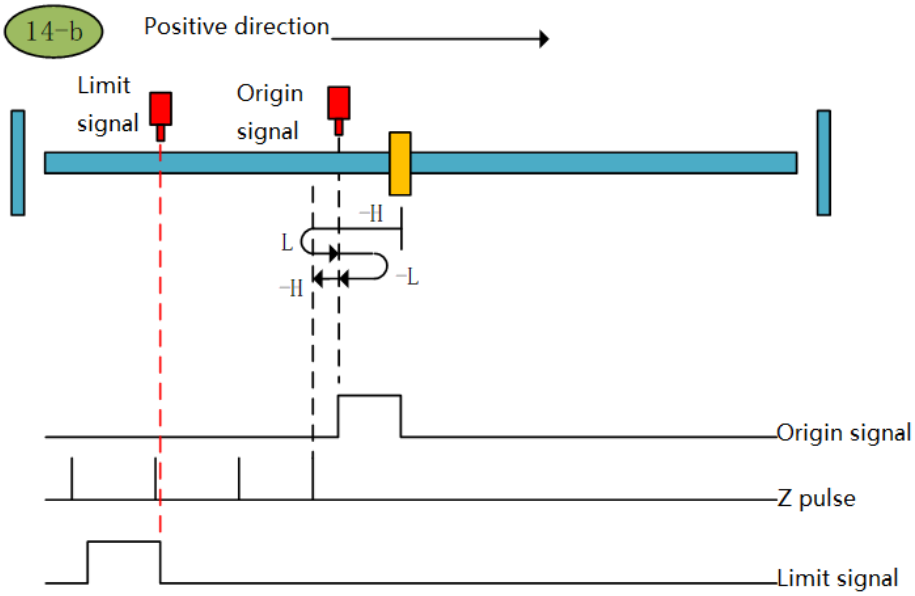
a. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Reverse low speed to find the origin falling edge → Reverse to find the Z pulse

b. Origin return start → Origin signal ON → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Reverse low speed to find the origin falling edge → Reverse to find Z pulse

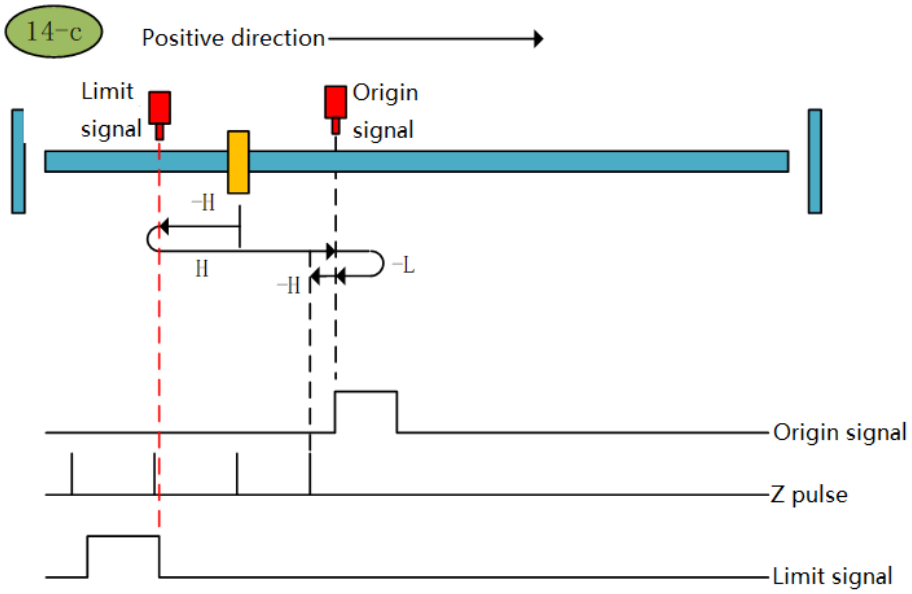
c. Origin return to zero → Origin signal OFF → Reverse high speed to find the origin falling edge → Hit the negative limit → Forward high speed to find the origin signal rising edge → Deceleration to 0 → Reverse low speed to find the origin signal falling edge → Reverse looking Z pulse



Origin return mode 14-a



Origin return mode 14-b



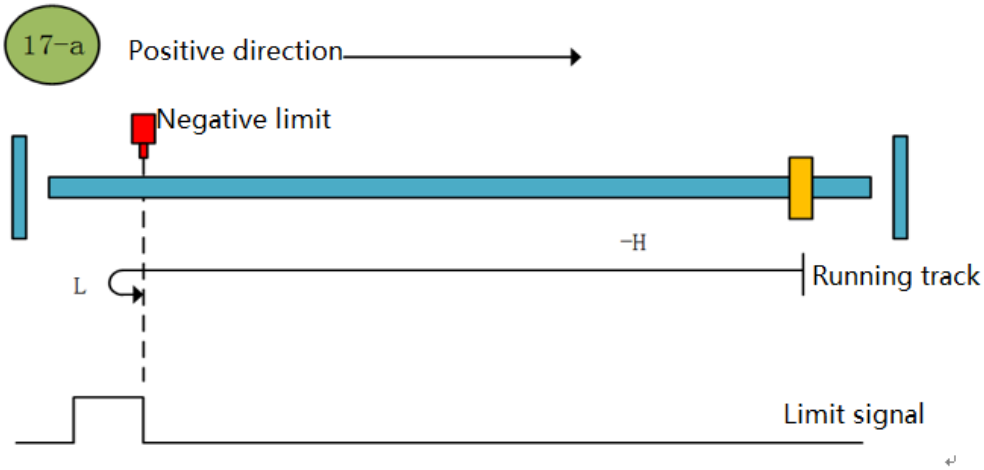
Origin return mode 14-c

Home zero return mode 15 (6098 00h = 15): Reserved.

Home zero return mode 16 (6098 00h = 16): Reserved.

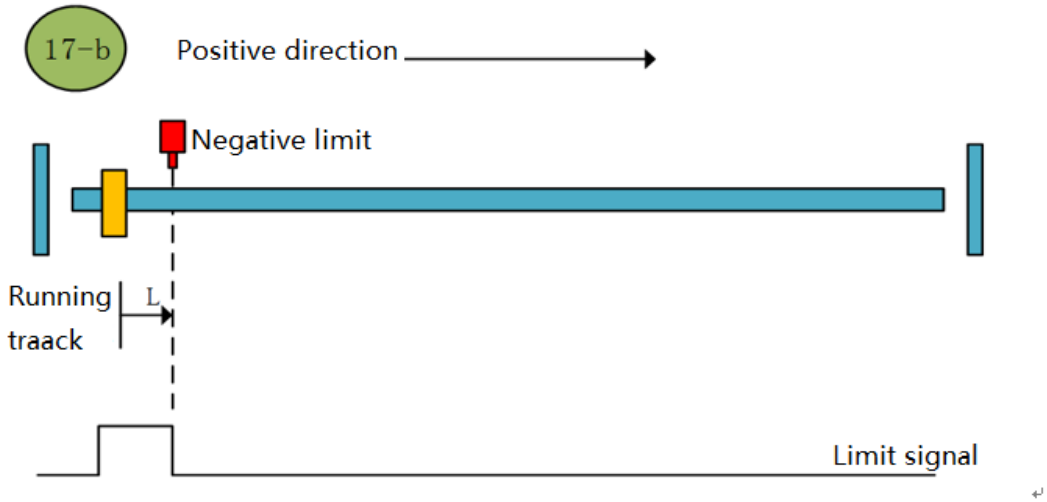
Zero return mode 17 (6098 00h = 17)

a. Start the origin return → reverse high speed to find the negative limit → hit the negative limit rising edge → decelerate to 0 → forward low speed to find the negative limit after the falling edge



Origin return mode 17-a

b. Start origin return → Negative limit is valid → Positive low speed finds negative limit and stops after falling

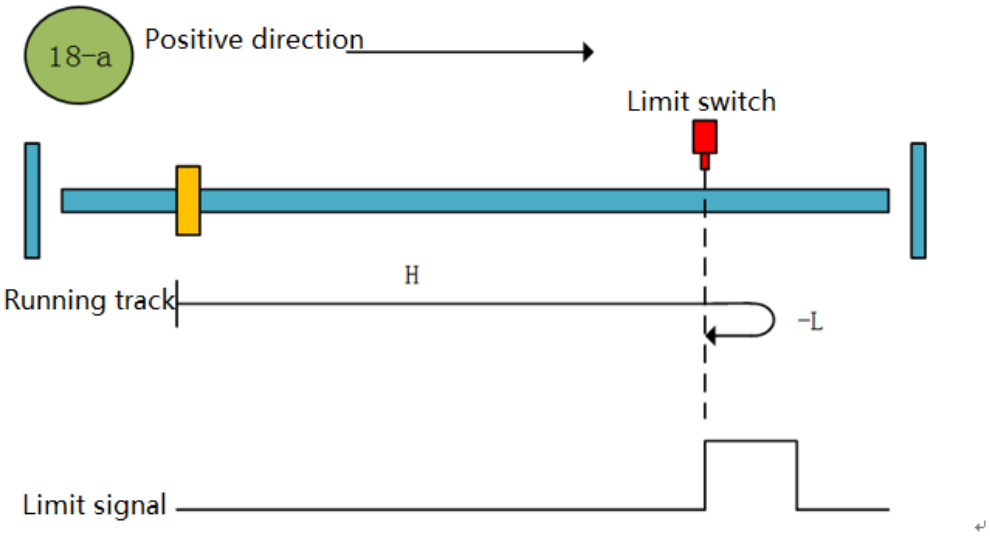


Origin return mode 17-b

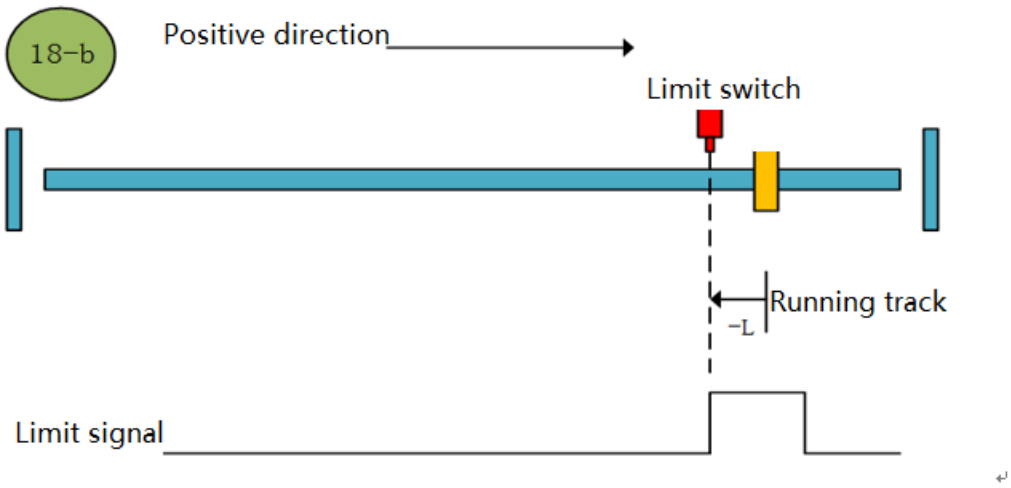
Zero return method of origin (6098 00h = 18)

a. Start the origin return → positive high speed correction limit → hit the positive limit rising edge → decelerate to 0 → reverse low speed to find the positive limit falling edge and stop

b. Start origin return → positive limit valid → reverse low speed correction limit



Zero return method 18-a

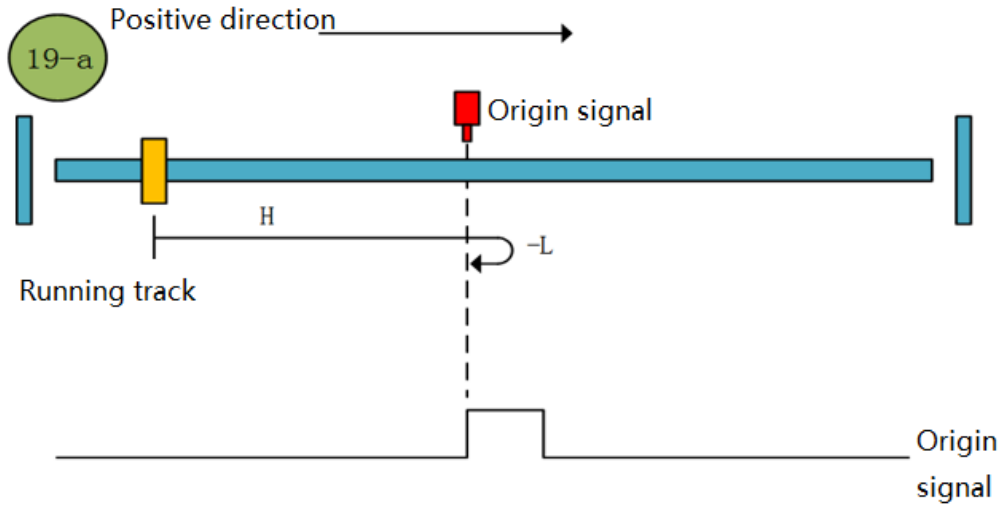


Zero return method 18-b

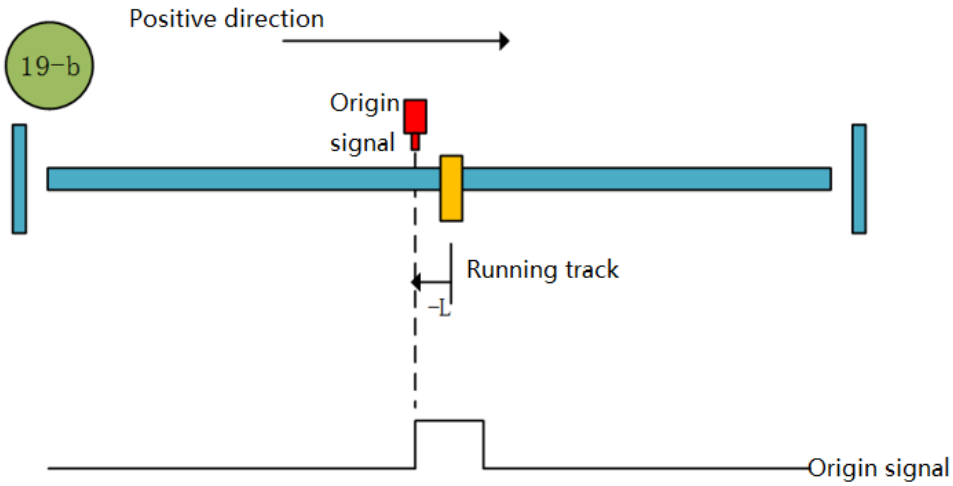
Zero return mode of origin 19 (6098 00h = 19)

a. Start the origin return → positive high speed correction limit → hit the positive limit rising edge → decelerate to 0 → reverse low speed to find the positive limit falling edge and stop

b. Start origin return → positive limit valid → reverse low speed correction limit



Zero return method 19-a

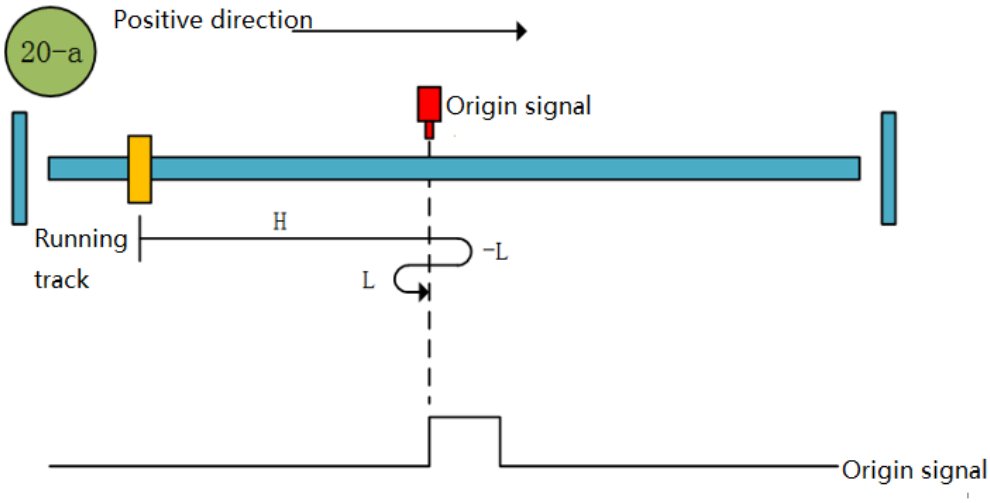


Zero return method 19-b

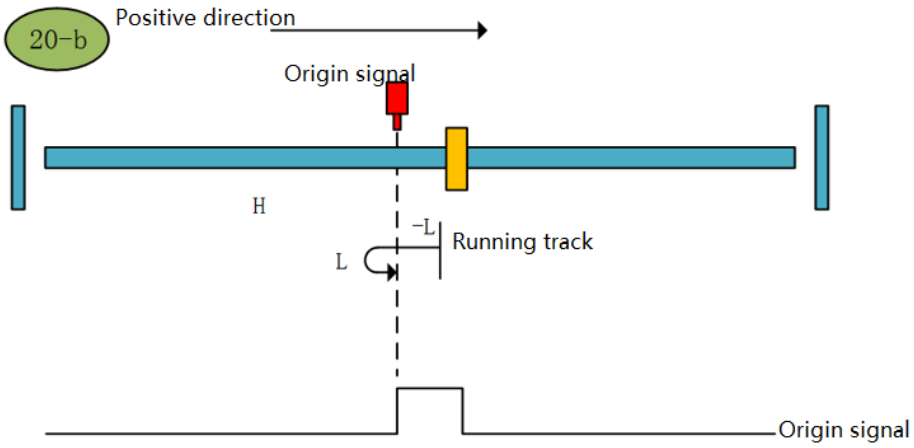
Zero return method of origin (6098 00h = 20)

a. Start OPR → Origin signal OFF → Forward high speed to find the origin rising edge → Deceleration to 0 → Reverse low speed to find the origin falling edge → Forward low speed to find the origin rising edge and stop

b. Start origin return → origin signal ON → reverse low speed to find the origin falling edge → positive low speed to find the original point rising edge and stop



Zero return method 20-a

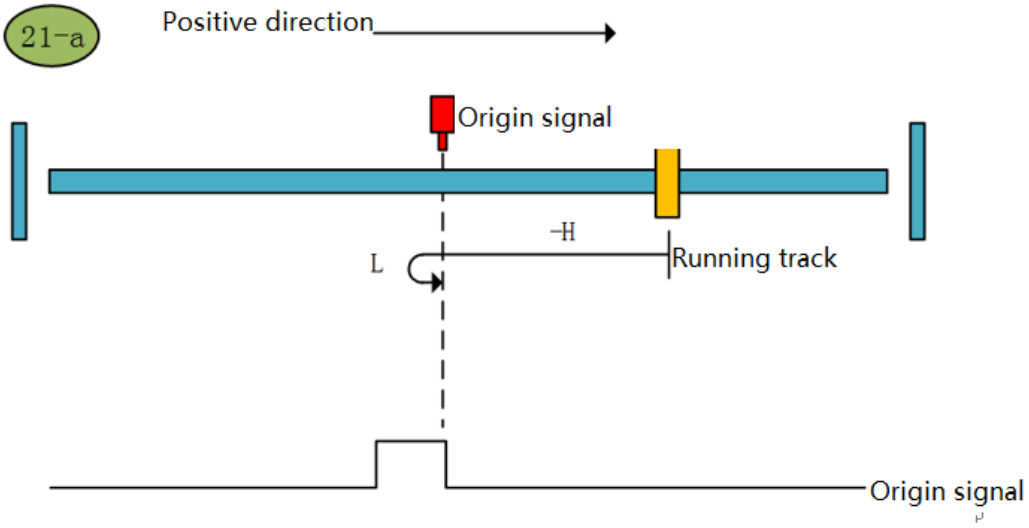


Origin return mode 20-b

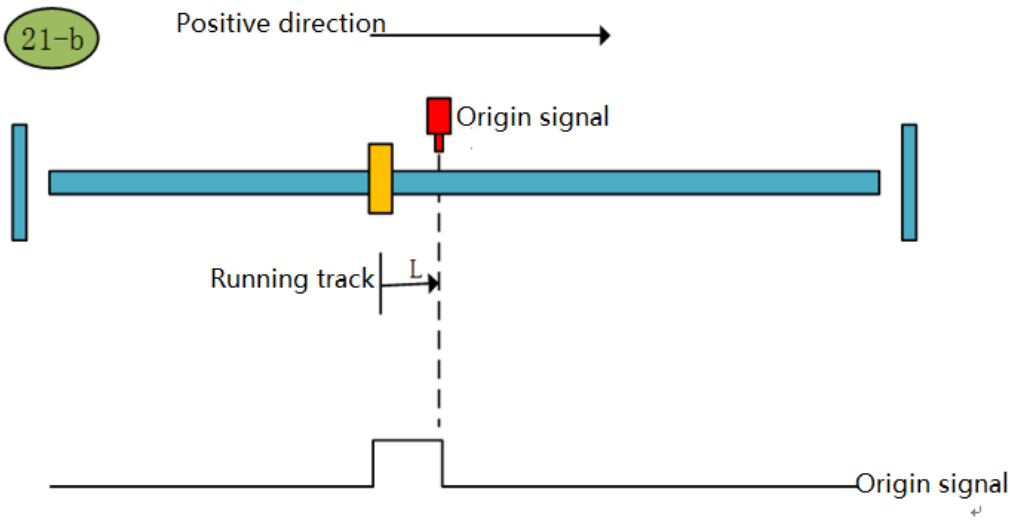
Zero return method of origin (6098 00h = 21)

a. Start OPR → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge and stop

b. Start OPR → Origin signal ON → Stop at low speed and find the falling edge of the origin



Origin return mode 21-a

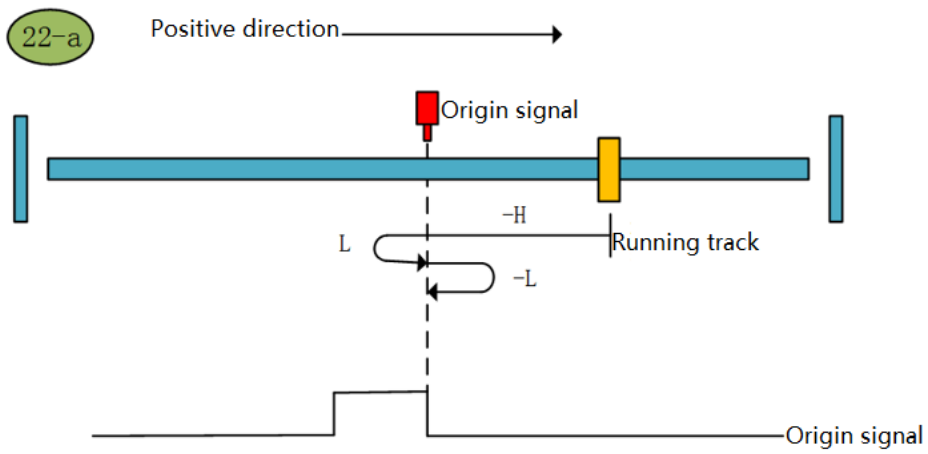


Origin return mode 21-b

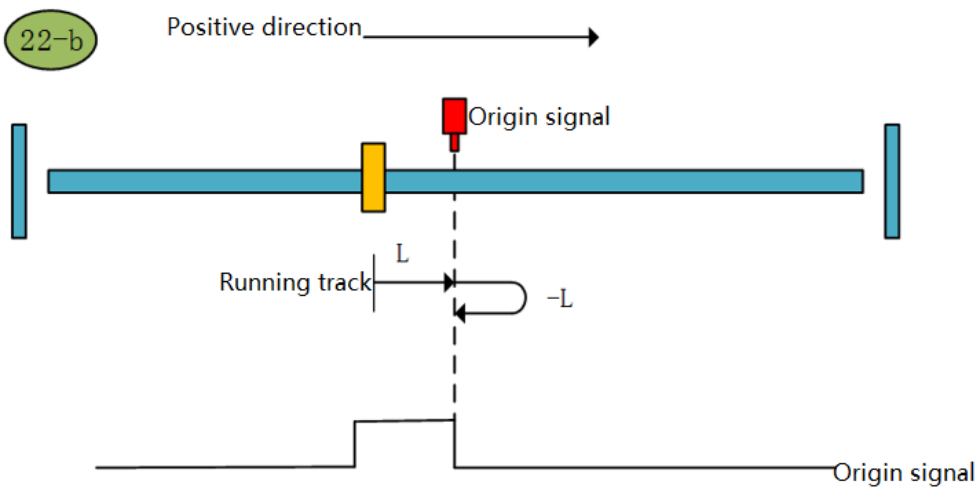
Origin return mode 22 (6098 00h = 22)

a Start OPR → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Reverse low speed to find the origin rising edge and stop

b Start the origin return → the origin signal ON → the forward low speed finds the origin falling edge → the reverse low speed finds the origin rising edge and stops



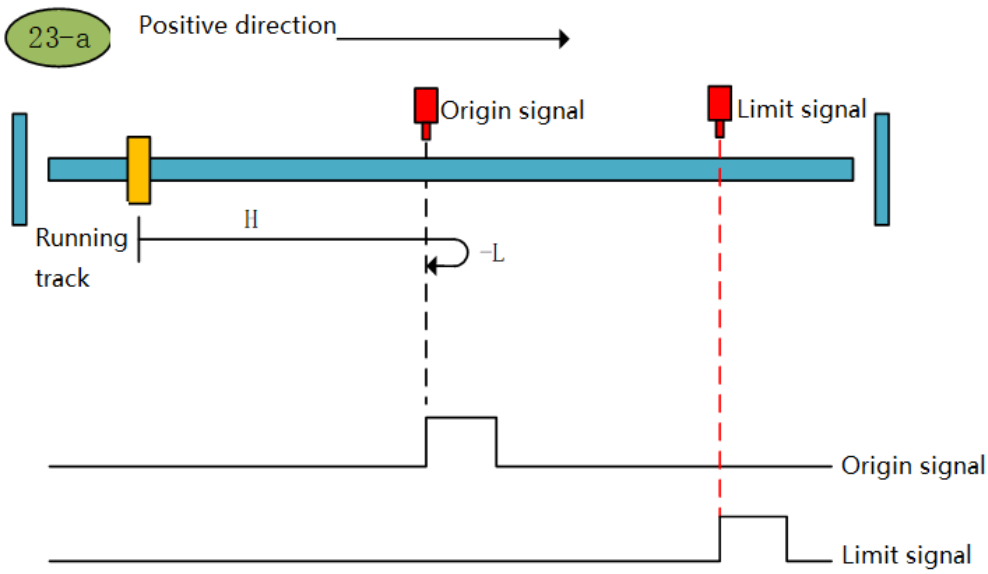
Origin return mode 22-a



Origin return mode 22-b

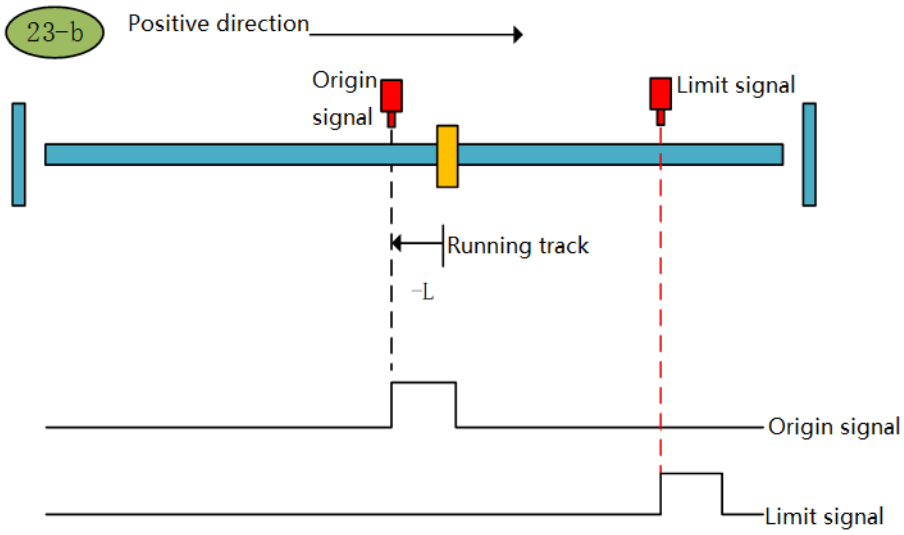
Zero return mode 23 (6098 00h = 23)

a Start OPR → Origin signal OFF → Forward high speed to find the origin rising edge → Deceleration to 0 → Reverse low speed to find the origin falling edge and stop



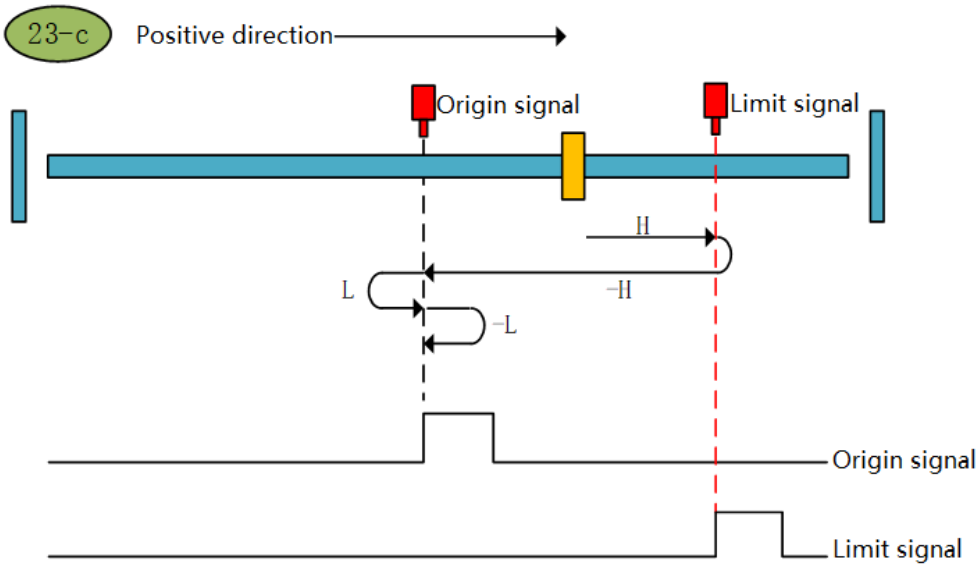
Origin return mode 23-a

b Start the origin return → origin signal ON → reverse low speed to find the original point after the falling edge



Origin return mode 23-b

c Start origin return → origin OFF → forward high speed to find the origin rising edge → hit the positive limit → reverse high speed to find the origin falling edge → decelerate to 0 → forward low speed to find the origin rising edge → reverse low speed to find the origin after the falling edge Downtime



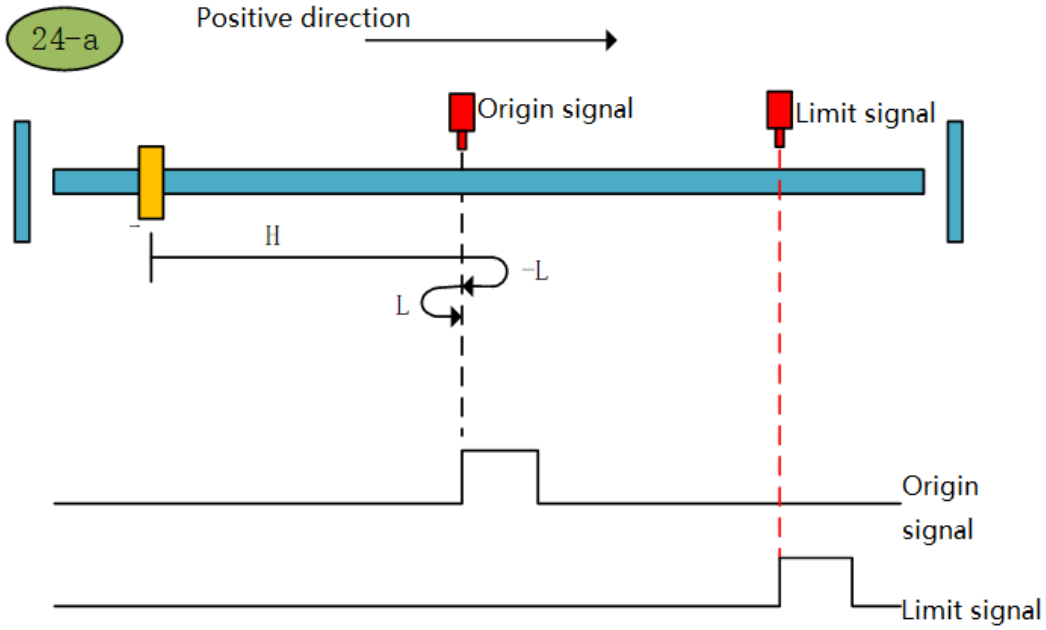
Origin return mode 23-c

Origin return mode 24 (6098 00h = 24)

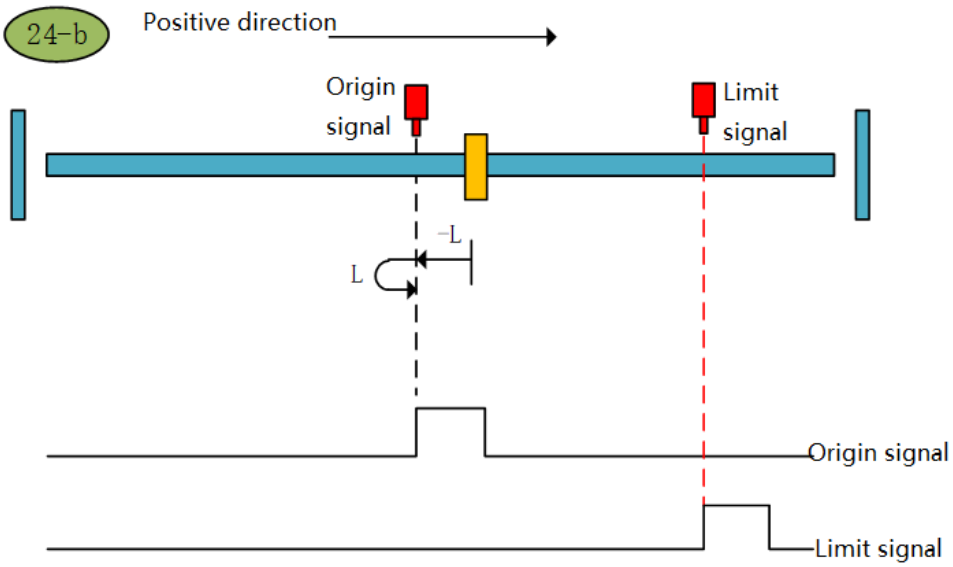
a. Start OPR → Origin signal OFF → Forward high speed to find the origin rising edge → Deceleration to 0 → Reverse low speed to find the origin falling edge → Forward low speed to find the origin rising edge and stop

b. Start origin return → origin signal ON → reverse low speed to find the origin falling edge → positive low speed to find the original point rising edge and stop

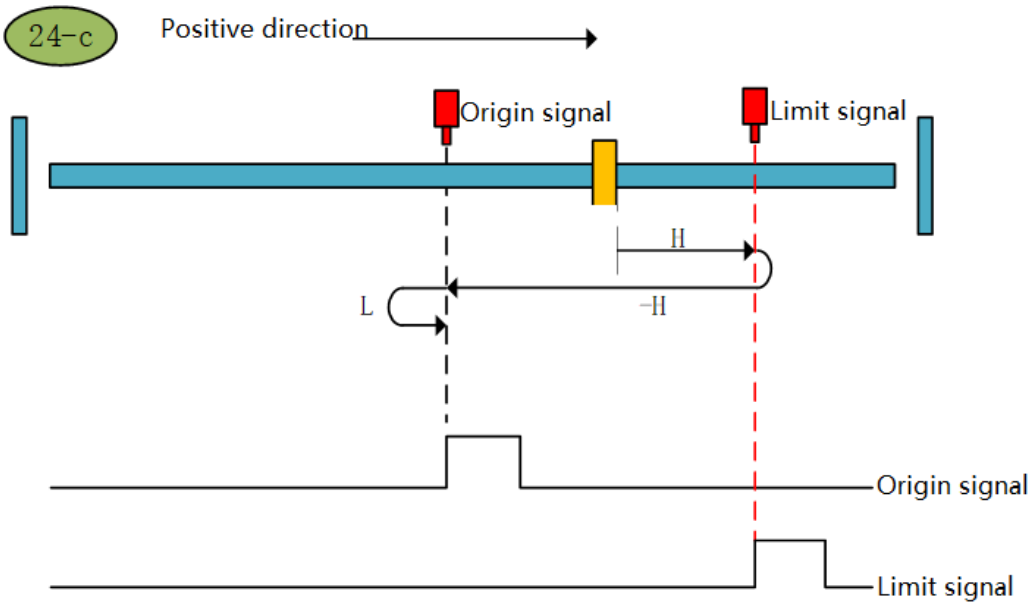
c. Start origin return → origin OFF → forward high speed to find the original rising edge → hit the positive limit → reverse high speed to find the origin falling edge → decelerate to 0 → forward low speed to find the original rising edge and stop



Origin return mode 24-a



Origin return mode 24-b



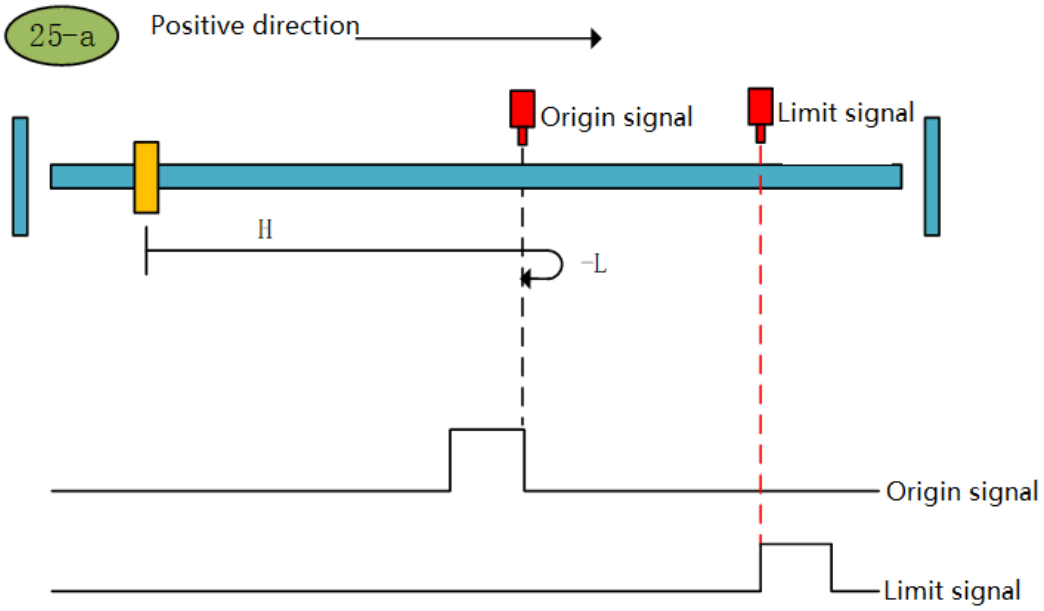
Origin return mode 24-c

Zero return mode of origin 25 (6098 00h = 25)

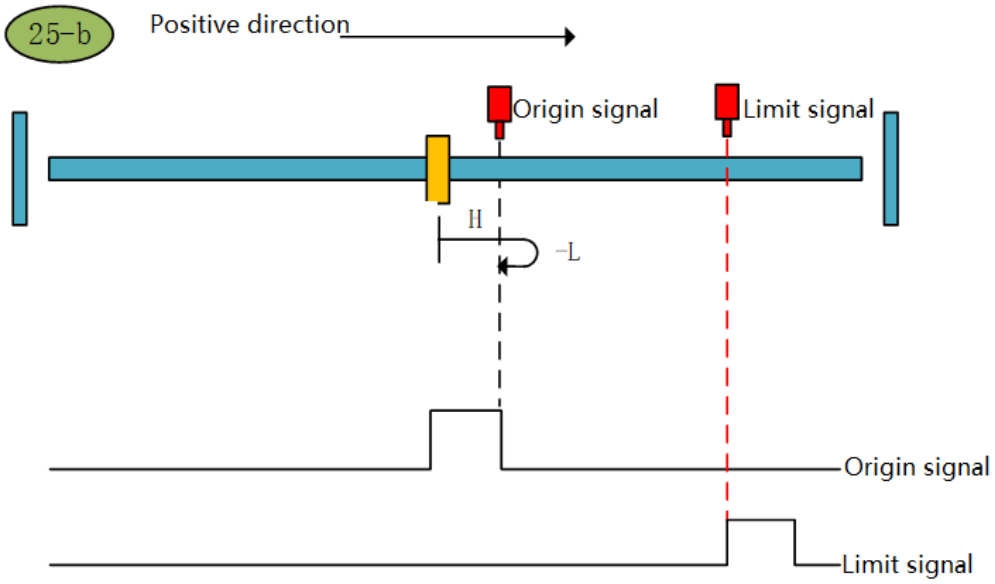
a. Start OPR → Origin signal OFF → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the original rising edge and stop

b. Start OPR → Origin signal ON → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the original rising edge and stop

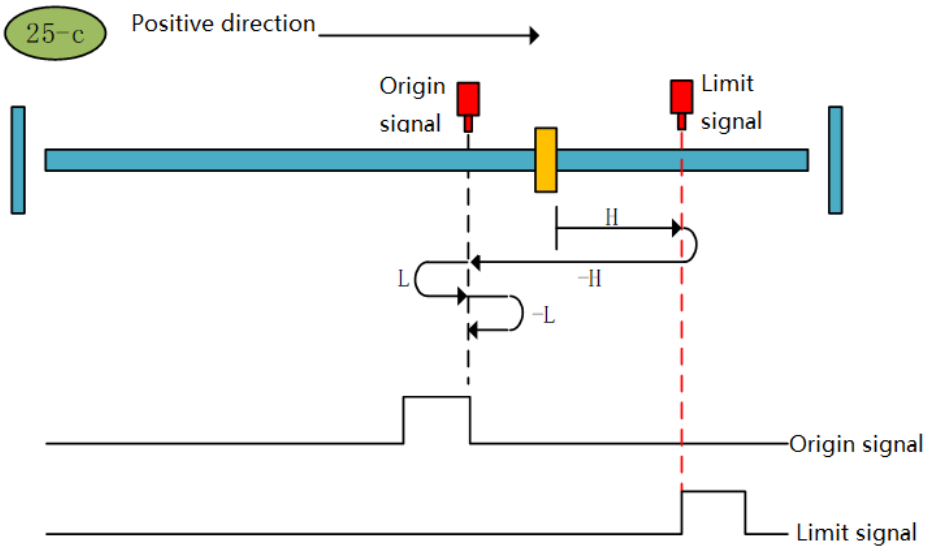
c. Start origin return → origin OFF → forward high speed to find the origin falling edge → hit the positive limit → reverse high speed to find the origin rising edge → decelerate to 0 → forward low speed to find the origin falling edge → reverse low speed to find the origin rising edge and stop



Origin return to zero 25-a



Origin return mode 25-b



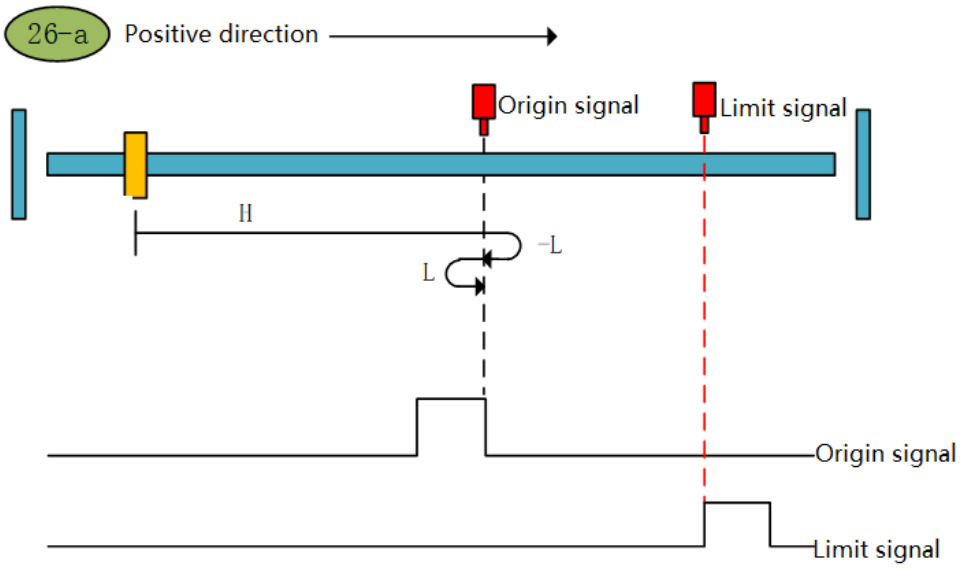
Origin return mode 25-c

Origin return mode 26 (6098 00h = 26)

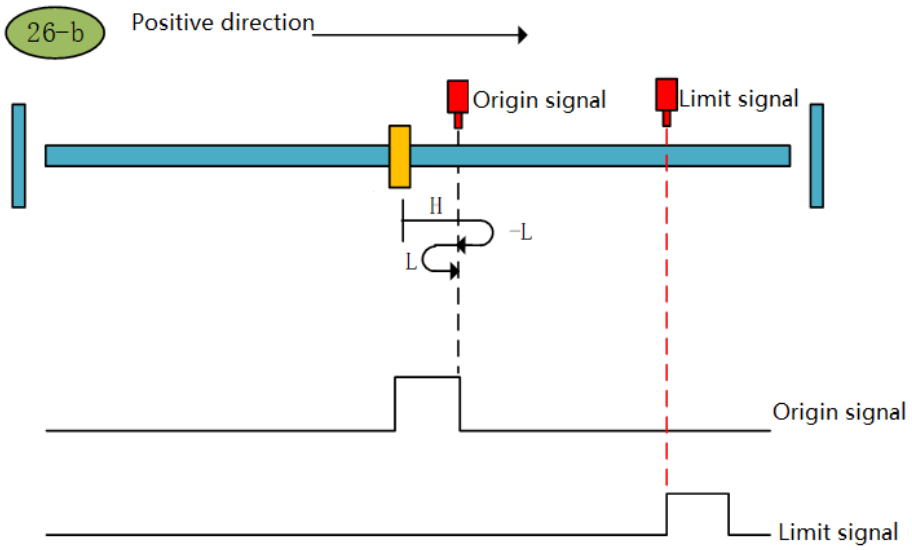
a. Start OPR → Origin signal OFF → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge → Forward low speed to find the origin falling edge and stop

b. Start OPR → Origin signal ON → Forward high speed to find the origin falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge → Forward low speed to find the origin falling edge and stop

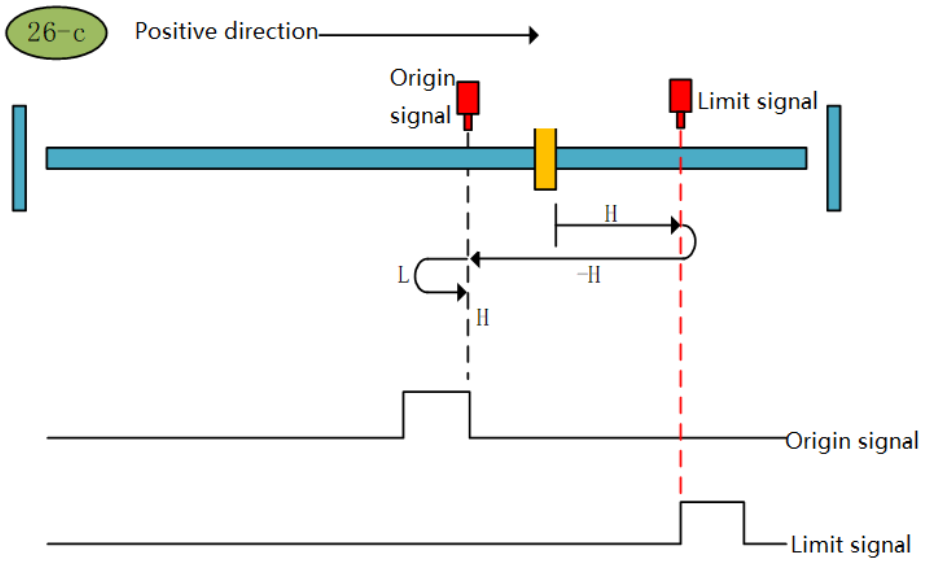
c. Start origin return → origin OFF → forward high speed to find the origin falling edge → hit the positive limit → reverse high speed to find the origin rising edge → decelerate to 0 → forward low speed to find the origin falling edge and stop



Origin return mode 26-a



Origin return mode 26-b



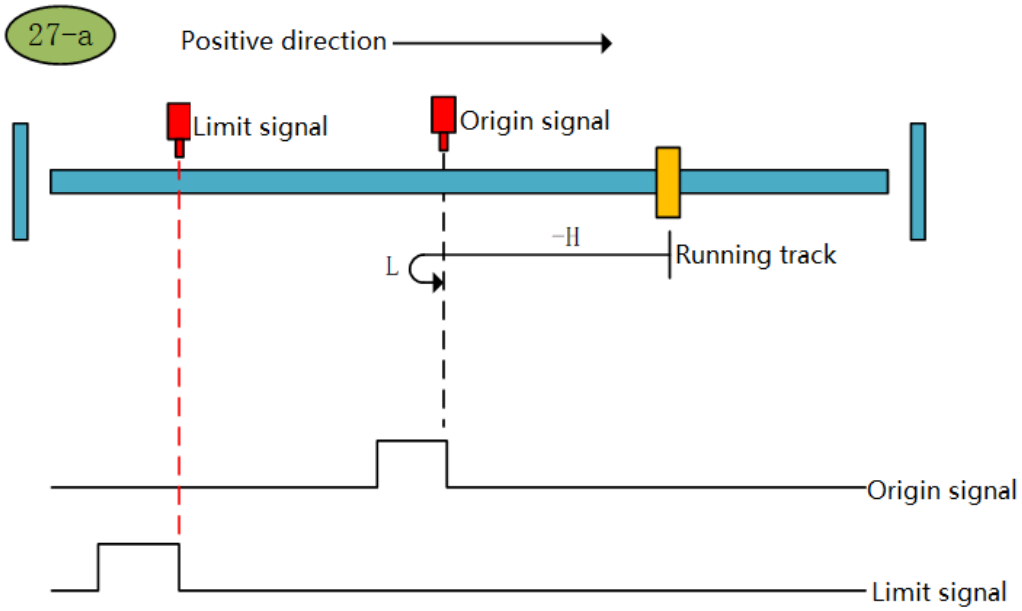
Origin return mode 26-c

Zero return mode 27 (6098 00h = 27)

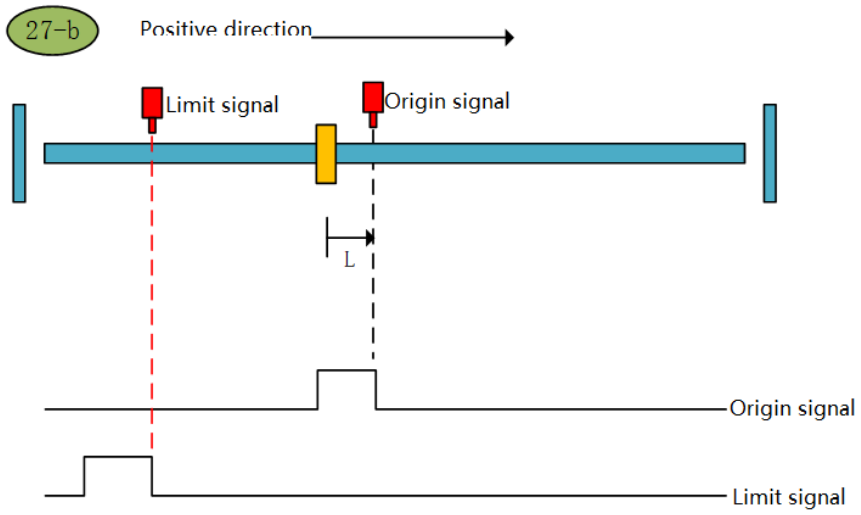
a. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge and stop

b. Origin return start → origin signal ON → forward low speed to find the origin after the falling edge

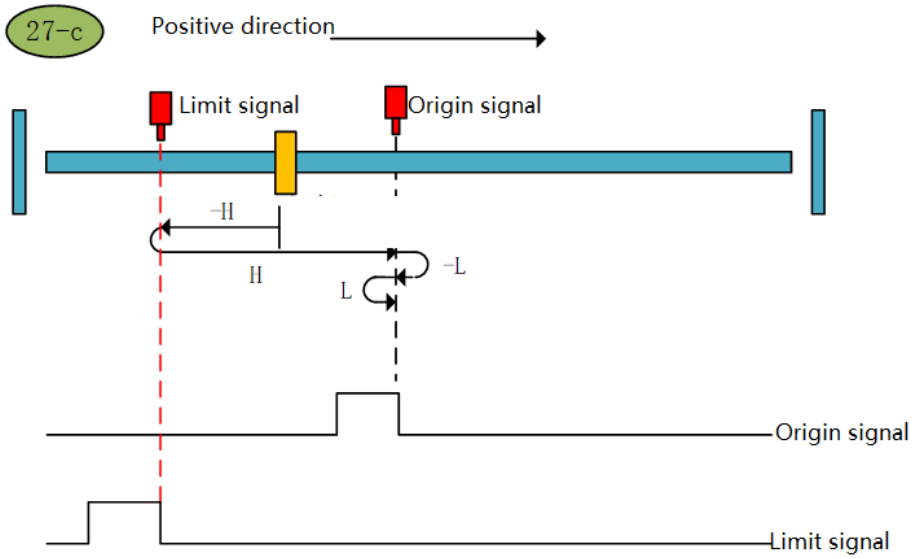
c. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Hit the negative limit → Forward high speed to find the origin signal falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge and stop



Origin return mode 27-a



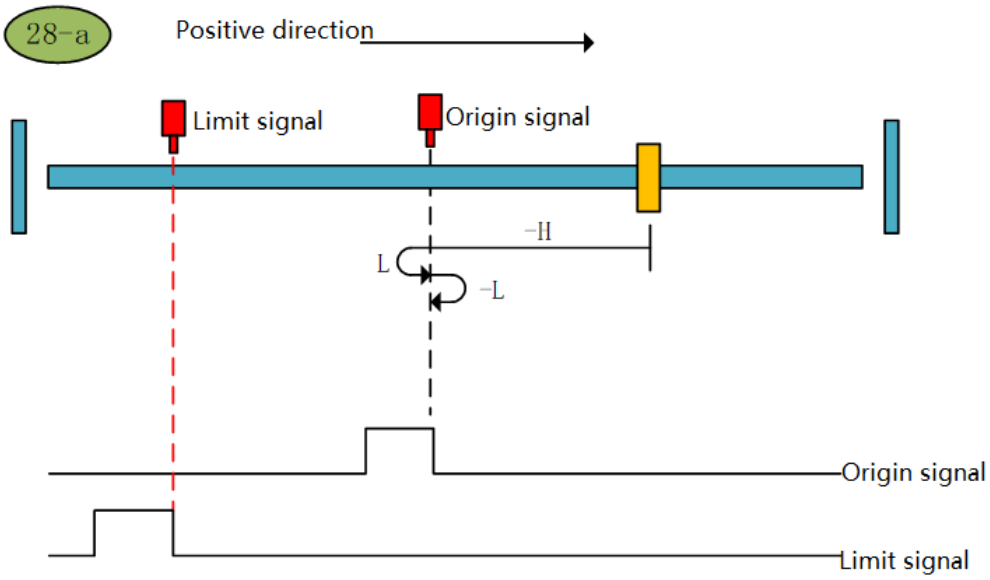
Origin return mode 27-b



Origin return mode 27-c

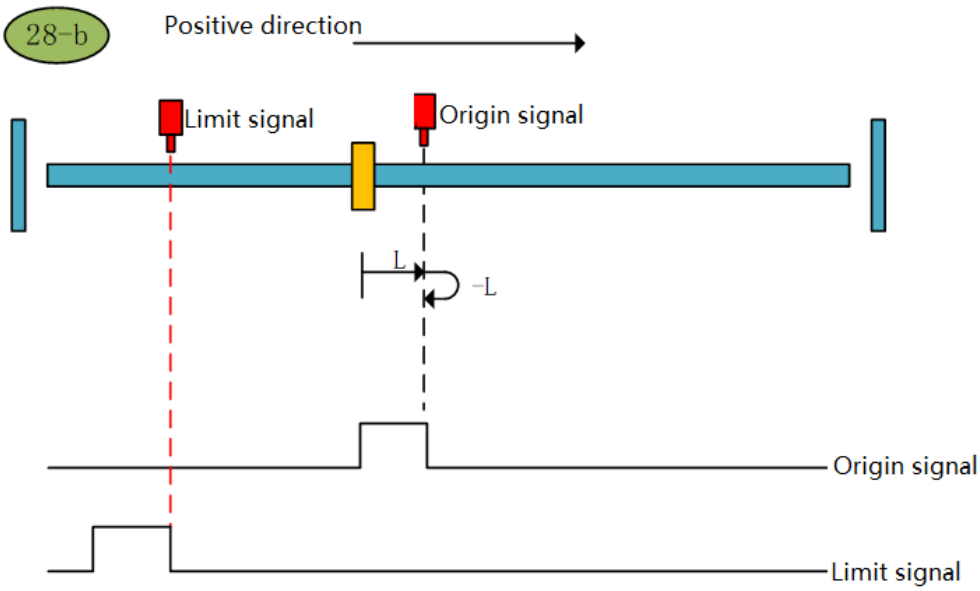
Zero return mode 28 (6098 00h = 28)

a. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Deceleration to 0 → Forward low speed to find the origin falling edge → Reverse low speed to find the origin rising edge and stop



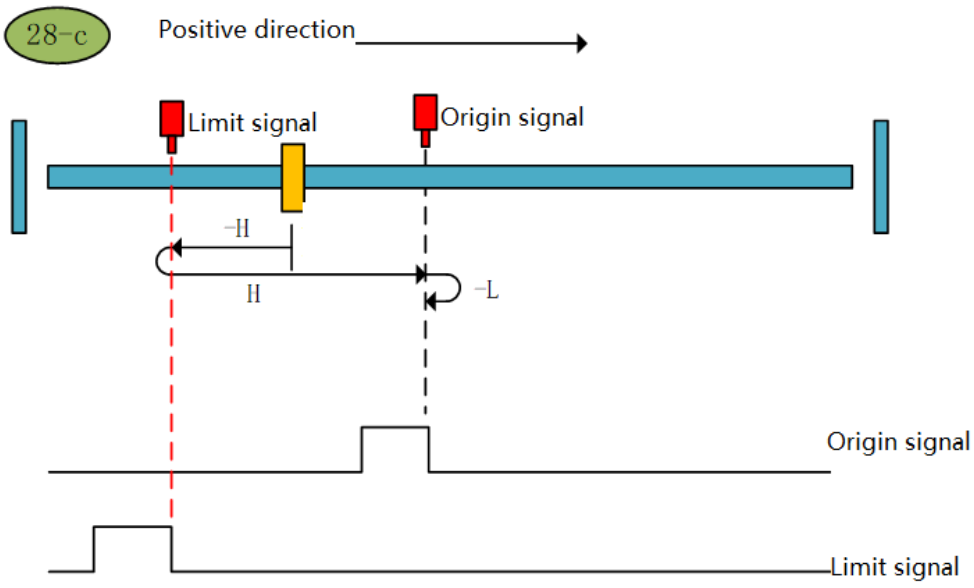
Origin return mode 28-a

b. Origin return start → Origin signal ON → Forward low speed to find the origin falling edge → Reverse low speed to find the origin rising edge and stop



Origin return mode 28-b

c. Origin return start → Origin signal OFF → Reverse high speed to find the origin rising edge → Hit the negative limit → Forward high speed to find the origin signal falling edge → Deceleration to 0 → Reverse low speed to find the origin rising edge and stop



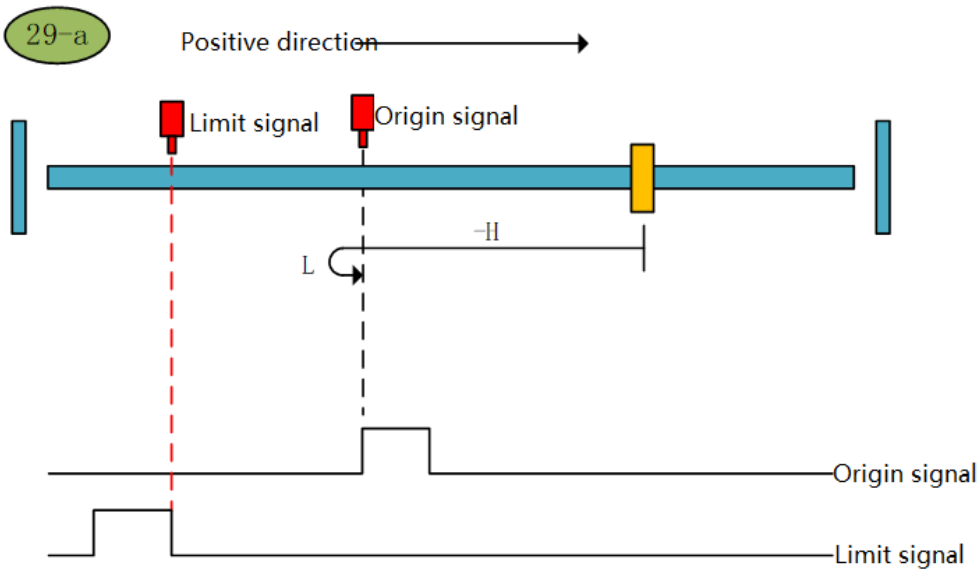
Origin return mode 28-c

Zero return mode 29 (6098 00h = 29)

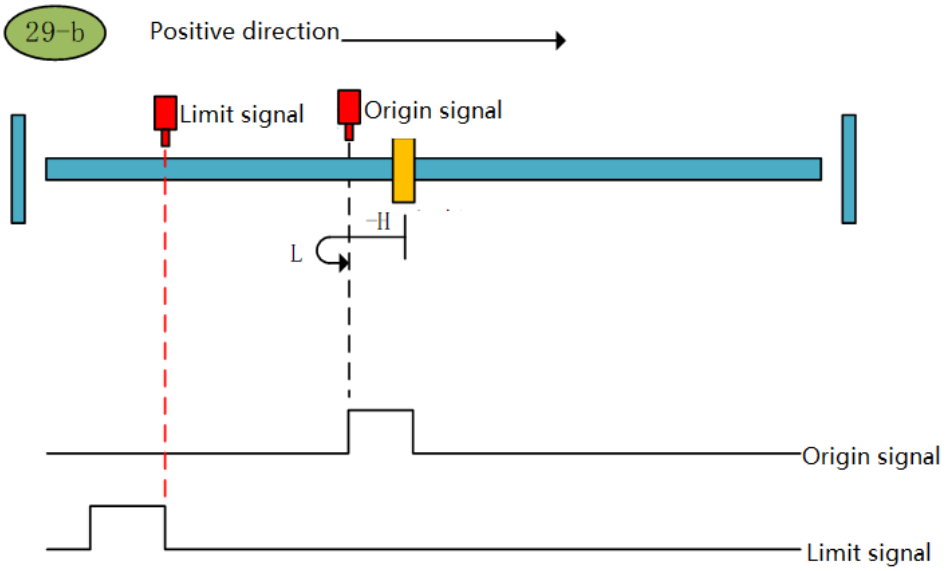
a. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge and stop

b. Origin return start → Origin signal ON → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge and stop

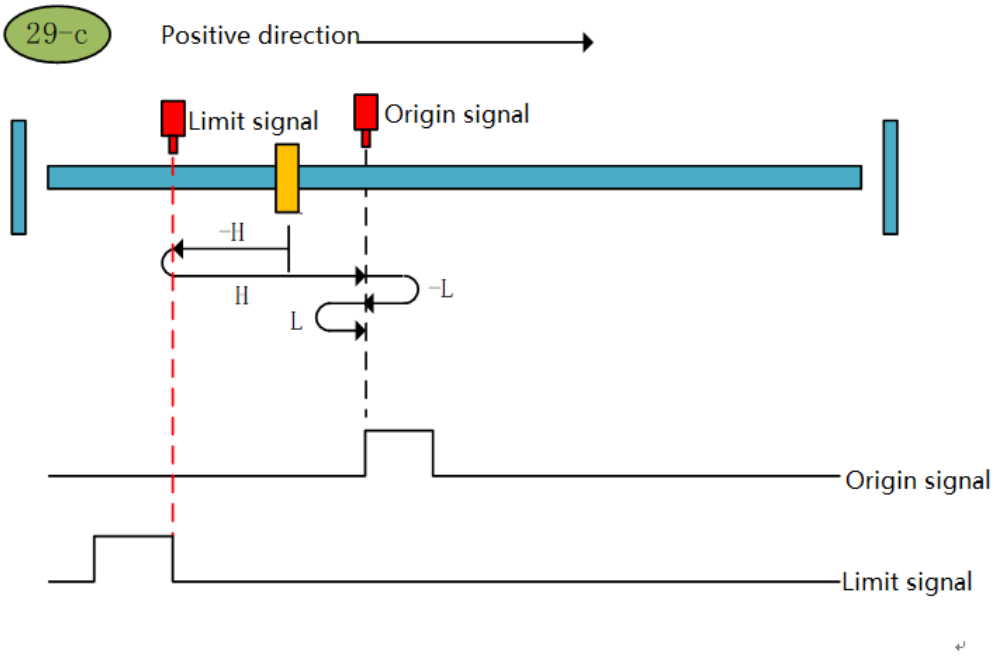
c. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Hit the negative limit → Forward high speed to find the origin signal rising edge → Decelerate to 0 → Reverse low speed to find the origin signal falling edge → Forward low speed to find the origin signal Stop after rising edge



Zero return method 29-a



Origin return mode 29-b



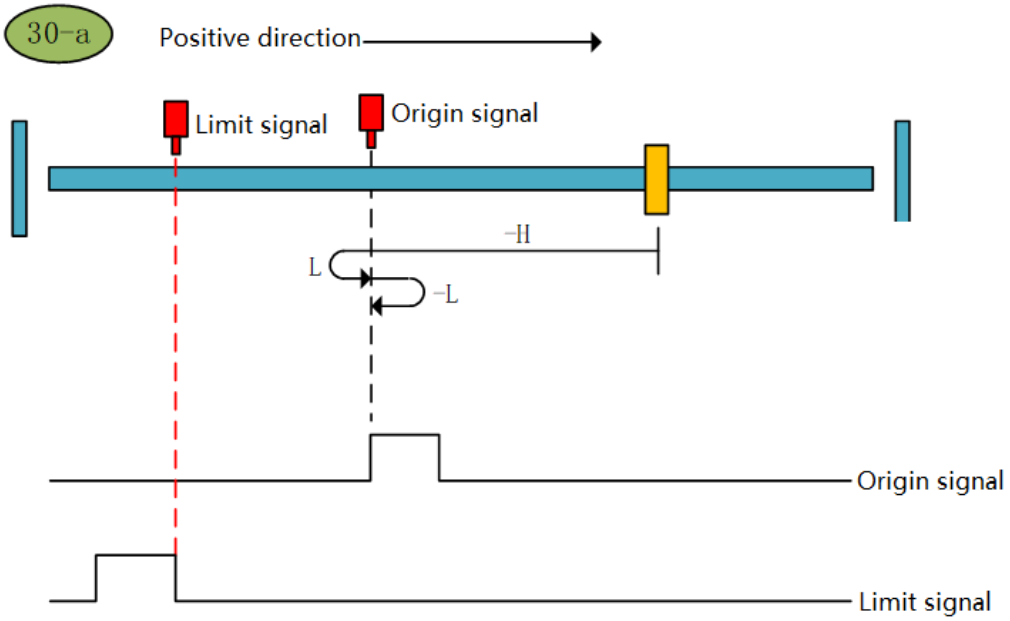
Origin return mode 29-c

Zero return mode 30 (6098 00h = 30)

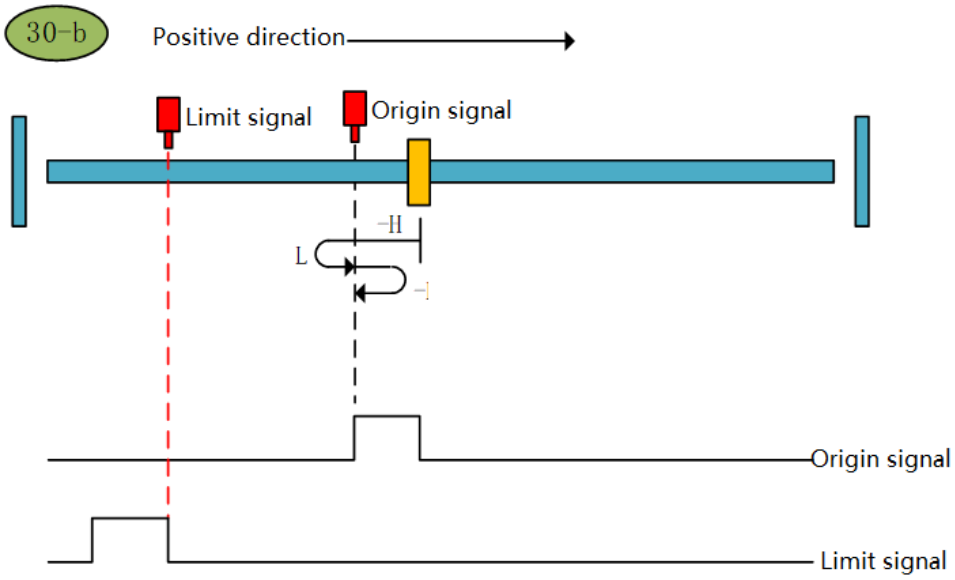
a. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Reverse low speed to find the origin falling edge and stop

b. Origin return start → Origin signal ON → Reverse high speed to find the origin falling edge → Deceleration to 0 → Forward low speed to find the origin rising edge → Reverse low speed to find the origin falling edge and stop

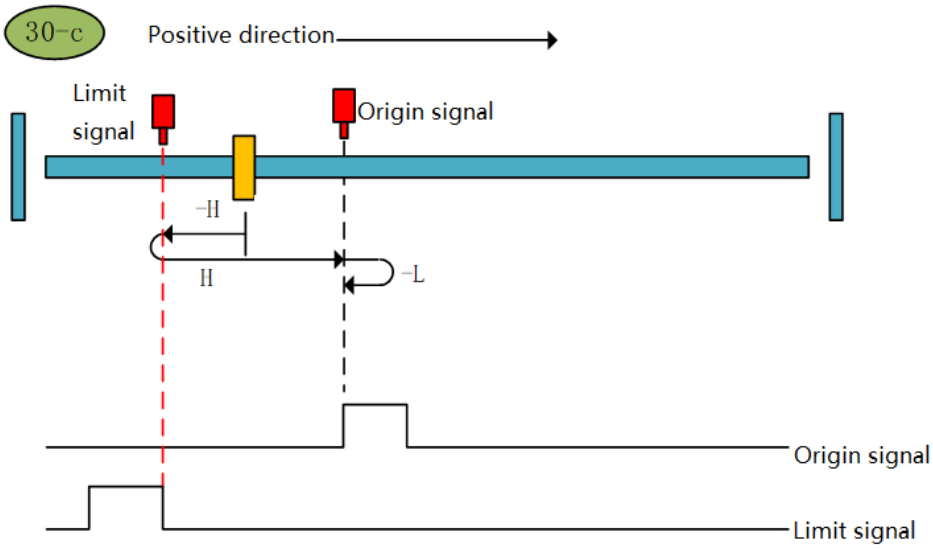
c. Origin return start → Origin signal OFF → Reverse high speed to find the origin falling edge → Hit the negative limit → Forward high speed to find the origin signal rising edge → Decelerate to 0 → Reverse low speed to find the origin signal falling edge and stop



Zero return method 30-a



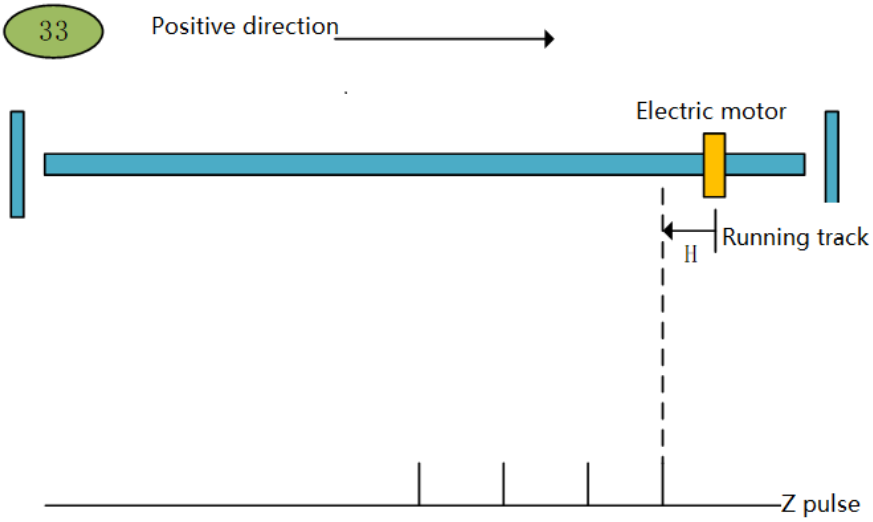
Zero return method 30-b



Zero return method 30-c

Zero return mode 31 (6098 00h = 31): Reserved.
Home zero return mode 32 (6098 00h = 32): Reserved.
Zero return mode of origin 33 (6098 00h = 33)

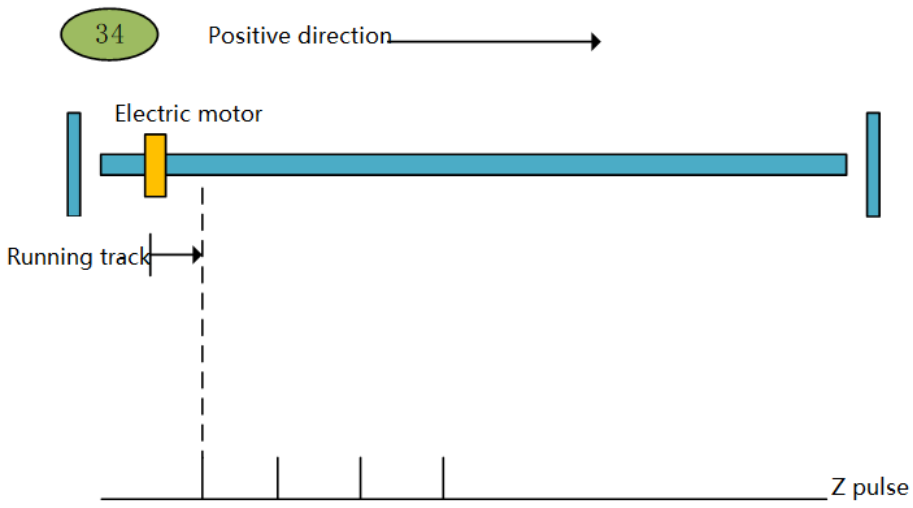
Starting zero return → negative looking for the first Z pulse



Origin return mode 33

Zero return mode of origin 34 (6098 00h = 34)

Origin return to zero → forward to find the first Z pulse



Origin return mode 34

6. Object Dictionary object dictionary

6.1 object attribute description

Glossary

" **Index** ": Specify the position of each object in the object dictionary, expressed in hexadecimal (h).

" **Subindex** ": Multiple objects under the same index, each object is offset under this class.

" **Data Type** ": See Table 5-1 for details.

Table 5-1 Description of data types

type of data	Numerical range	Data length	DS301 value
Int8	-128 ~ 127	1 byte	2
Uint8	0 ~ 255	1 byte	5
Int16	-32768 ~ +32767	2 bytes	3
Uint16	0 ~ 65535	2 bytes	6
Int32	-2147483648 ~ +2147483647	4 bytes	4
Uint32	0 ~ 4294967295	4 bytes	7
String	ASCII	-	9

" **Read and write type** ": See Table 4-1 for details.

Table 5-2 Description of read and write types

Read and write type	Description
RW	Read and write
WO	just write
RO	Read only
CONST	Constant, read only

" **Object structure** ": See Table 3-3 for details.

Table 5-3 Description of object structure

Object structure	Description
VAR	A single simple value containing the data types in Table 3-1
ARR	Multiple data field objects, the variables in the fields are all the same data shape
REC	Multiple data field objects, the variables in the fields are different data shapes

6.2 1000h group object list

index	Subindex	name	Accessibility	PDO mapping	type of data	unit	range	Factory default
1000h	00	Drive type	RO	NO	UINT32	-	-	0x00020192
100 1 h	00	error register	RO	NO	UINT8	-	-	-
100 8 h	00	Driver name	RO	NO	-	-	-	Servo Device
100 A h	00	Software version	RO	NO	-	-	-	3761
10 18 h	00	Maximum son Quote number	RO	NO	UINT8	-	-	4
	01	Vendor ID	RO	NO	UINT32	-	-	0x850104
	02	Device code	RO	NO	UINT32	-	-	0x26483052
	03	Revision	RO	NO	UINT32	-	-	0x39613708
	04	serial number	RO	NO	UINT32	-	-	0x20181123
1 600 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	5
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60400010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x607A0020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0x60FF0020
	04	Mapping object 4	RW	NO	UINT32		0~(2^{32}-1)	0x60710010
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0x60600008
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 601 h	00	Mapping object	RW	NO	UINT8	-	0~8	2

		Number						
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60400010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60FF0020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 602 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	2
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60400010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60710020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 603 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	7
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60400010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x607A0020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0x60FF0020

	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0x60600008
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0x60B80010
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0x60E00010
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0x60E10010
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 A00 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	5
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60410010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60640020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0x606C0020
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0x60770010
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0x60610008
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 A01 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	2
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60410010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60640020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0

	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 A02 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	3
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60410010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60640020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0x606C0020
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 A03 h	00	Mapping object Number	RW	NO	UINT8	-	0~8	3
	01	Mapping object 1	RW	NO	UINT32	-	0~(2^{32}-1)	0x60410010
	02	Mapping object 2	RW	NO	UINT32	-	0~(2^{32}-1)	0x60640020
	03	Mapping object 3	RW	NO	UINT32	-	0~(2^{32}-1)	0x60770010
	04	Mapping object 4	RW	NO	UINT32	-	0~(2^{32}-1)	0
	05	Mapping object 5	RW	NO	UINT32	-	0~(2^{32}-1)	0
	06	Mapping object 6	RW	NO	UINT32	-	0~(2^{32}-1)	0
	07	Mapping object 7	RW	NO	UINT32	-	0~(2^{32}-1)	0
	08	Mapping object 8	RW	NO	UINT32	-	0~(2^{32}-1)	0
1 C00 h	00	Maximum subindex	RO	NO	UINT8	-	-	4
	01	SM0 type	RO	NO	UINT8	-	-	1

	02	SM1 type	RO	NO	UINT8	-	-	2
	03	SM2 type	RO	NO	UINT8	-	-	3
	04	SM3 type	RO	NO	UINT8	-	-	4
1 C12 h	00	RPDO allocation Maximum subindex	RW	NO	UINT8	-	0~1	1
	01	RPDO allocation Object index	RW	YES	UINT16	-	0~65535	0x1601
1 C13 h	00	TPDO allocation Maximum subindex	RW	NO	UINT8	-	0~1	1
	01	TPDO allocation Object index	RW	YES	UINT16	-	0~65535	0x1A00
1 C32 h	00	Synchronous output Largest child Number of indexes	RO	NO	UINT8	-	-	32
	01	Synchronization type	RW	NO	UINT16	-	-	2
	02	Cycle Time	RO	NO	UINT32	Ns	-	0
	04	Supported synchronization Types of	RO	NO	UINT16	-	-	4
	05	Minimum period	RO	NO	UINT32	Ns	-	125000
	06	Calculation and replication time	RO	NO	UINT32	Ns	-	0
	08	Get loop time	RW	NO	UINT16	-	-	0
	09	delay	RO	NO	UINT32	Ns	-	0
	0A	SYN0 time	RW	NO	UINT32	Ns	-	-
	0B	Lost number	RO	NO	UINT16	-	-	0
	0C	Loop out count	RO	NO	UINT16	-	-	0
	20	Synchronization error	RO	NO	BOOL	-	-	FALSE
1 C33 h	00	Synchronous input Largest child Number of indexes	RO	NO	UINT8	-	-	32

	01	Synchronization type	RW	NO	UINT16	-	-	2
	02	Cycle Time	RO	NO	UINT32	Ns	-	0
	04	Supported synchronization Types of	RO	NO	UINT16	-	-	4
	05	Minimum period	RO	NO	UINT32	Ns	-	125000
	06	Calculation and replication time	RO	NO	UINT32	Ns	-	0
	08	Get loop time	RW	NO	UINT16	-	-	0
	09	delay	RO	NO	UINT32	Ns	-	0
	0A	SYN0 time	RW	NO	UINT32	Ns	-	-
	0B	Lost number	RO	NO	UINT16	-	-	0
	0C	Loop out count	RO	NO	UINT16	-	-	0
	20	Synchronization error	RO	NO	BOOL	-	-	0

6.3 6000h group object list

index	Subindex	name	Accessibility	PDO mapping	type of data	unit	range	Factory default
603F h	00	error code	RO	TPDO	UINT16	-	0~65535	0
6040 h	00	Control word	RW	RPDO	UINT16	-	0~65535	0
6041 h	00	Status word	RO	TPDO	UINT16	-	0~65535	0
6060 h	00	Control mode	RO	RPDO	INT8	-	0~10	8
6061 h	00	Mode display	RO	TPDO	INT8	-	0~10	-
6062 h	00	Position command	RO	TPDO	INT32	Command unit	-	-
6063 h	00	Position feedback	RO	TPDO	INT32	Encoder unit	-	-
6064h	00	Position feedback	RO	TPDO	INT32	Command unit	-	-
6065h	00	Position deviation	RW	RPDO	UINT32	Command unit	-	0xA0000000

		Excessive threshold						
6067h	00	Location arrival Threshold	RW	RPDO	UINT32	Command unit	0~(2^{32}-1)	100
6068h	00	Location arrival Window time	RW	RPDO	UINT16	2ms	0~65535	0
606Bh	00	Actual speed instruction	RO	TPDO	UINT32	r/min	-	-
606Ch	00	Actual speed	RO	TPDO	INT32	Command unit /s	-	-
606Dh	00	Speed arrival Threshold	RW	RPDO	UINT16	r/min	0~65535	100
606Eh	00	Speed arrival Window time	RW	RPDO	UINT16	Ms	0~65535	0
606Fh	00	Zero speed threshold	RW	RPDO	UINT16	r/min	0~65535	10
6070h	00	0 -speed time window	RW	RPDO	UINT16	Ms	0~65535	0
6071h	00	Target torque	RW	RPDO	INT16	0.1%	-3000~3000	0
6074h	00	Torque command	RO	TPDO	INT16	1%	-300~300	0
6075h	00	Motor rating Current	RO	TPDO	UINT32	mA	0~(2^{32}-1)	-
6076h	00	Motor rating Torque	RO	TPDO	UINT32	mNm	0~(2^{32}-1)	-
6077h	00	Actual torque	RO	TPDO	UINT16	1%	-300~300	0
607Ah	00	target location	RW	RPDO	INT32	Command unit	-2^{31}~(2^{31}-1)	0
607Ch	00	Origin offset	RW	RPDO	INT32	Encoder unit	-2^{31}~(2^{31}-1)	0
607Dh	00	Sub-index number	RO	NO	UINT8	-	-	2
	01	Minimum position limit	RW	RPDO	INT32	Command unit	-2^{31}~(2^{31}-1)	-2^{31}
	02	Maximum position limit	RW	RPDO	INT32	Command unit	-2^{31}~(2^{31}-1)	2^{31}-1

607Eh	00	Command polarity	RW	RPDO	UINT8	-	0~255	0
607Fh	00	Maximum profile speed	RW	RPDO	UINT32	0.1r/min	0~(2^{32}-1)	50000
6080h	00	Maximum motor speed	RW	RPDO	UINT32	1r/min	0~(2^{32}-1)	6000
6081h	00	Contour speed	RW	RPDO	UINT32	Command unit /s	0~(2^{32}-1)	10000000
6083h	00	Contour acceleration degree	RW	RPDO	UINT32	Command unit /s^2	0~(2^{32}-1)	200
6084h	00	Contour deceleration degree	RW	RPDO	UINT32	Command unit /s^2	0~(2^{32}-1)	200
6086h	00	Running curve select	RW	RPDO	INT16	-	-2^{15}~(2^{15}-1)	0
6087h	00	Torque ramp time	RW	RPDO	UINT32	MS	0~(2^{32}-1)	0
6091h	00	Subindex Number	RO	NO	UINT8	-	-	2
	01	Motor Resolution	RW	RPDO	UINT32	-	1~(2^{32}-1)	1
	02	Load shaft Resolution	RW	RPDO	UINT32	-	1~(2^{32}-1)	1
6098h	00	Origin return the way	RW	RPDO	INT8	-	1~35	1
6099h	00	Subindex Number	RO	NO	UINT8	-	-	2
	01	High speed search Deceleration point	RW	RPDO	UINT32	Command unit /s	0~(2^{32}-1)	279620266
	02	Low speed search origin	RW	RPDO	UINT32	Command unit /s	0~(2^{32}-1)	5592405
609Ah	00	Return to zero Acceleration	RW	RPDO	UINT32	Command unit /s^2	0~(2^{32}-1)	16000

60B0h	00	Position offset	RW	RPDO	INT32	Command unit	-2^{31} ~ (2^{31}-1)	0
60B8h	00	Probe mode	RW	RPDO	UINT16	-	0~65535	0
60B9h	00	Probe status	RO	TPDO	UINT16	-	0~65535	0
60BAh	00	Probe 1 on Rising position	RO	TPDO	INT32	Command unit	-2^{31} ~ (2^{31}-1)	0
60BBh	00	Under probe 1 Falling edge position	RO	TPDO	INT32	Command unit	-2^{31} ~ (2^{31}-1)	0
60BCh	00	Probe 1 on Rising position	RO	TPDO	INT32	Command unit	-2^{31} ~ (2^{31}-1)	0
60BDh	00	Under probe 1 Falling edge position	RO	TPDO	INT32	Command unit	-2^{31} ~ (2^{31}-1)	0
60E0h	00	Forward torque limit	RW	RPDO	UINT16	0.1%	0~3000	3000
60E1h	00	Reverse torque limit	RW	RPDO	UINT16	0.1%	0~3000	3000
60F4h	00	Position deviation	RO	TPDO	INT32	Command unit	-	-
60FCh	00	Position command	RO	TPDO	INT32	Encoder unit	-	-
60FDh	00	DI status	RO	TPDO	UNT32	-	-	-
60FEh	00	Number of subindexes	RO	NO	UINT8	-	-	2
	01	Physical output	RO	TPDO	UINT32	-	0~(2^{32}-1)	0
	02	-	-	-	-	-	-	-
60FFh	00	Target speed	RW	RPDO	INT32	Command unit /s	-	-
6052h	00	Supported Drive mode	RO	NO	UINT32	-	-	0x3AD

6.4 Object Dictionary Details

6.4.1 Detailed Description of 1000h Group Objects

Object 1000h : Device Type					
Index	1000h				
Name	Equipment type				
Object structure	VAR	**type of data**	Uint32	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	0x00020 192

Describe the CoE device type:

BIT	name	description
0~15	Device subprotocol	402 (0x192): device sub-protocol
16~23	Types of	02: Servo drive
25~31	mode	Factory customization

Object 1001h : Error Register					
Index	1001h				
Name	Error register				
Object structure	VAR	**type of data**	Uint8	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	0

Bitwise to contain error messages

BIT	meaning	BIT	meaning
0	conventional	4	Communication
1	Current	5	Sub protocol
2	Voltage	6	Reserved
3	temperature	7	Vendor definition

Object 1008h : Manufacturer Device Name					
Index	1008h				
Name	Manufacturer equipment name				
Object structure	-	**type of data**	UINT8	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory setting**	Servo Device

Describe the manufacturer's device name

Object 100Ah : Software Version (Software Version)					
Index	100Ah				
Name	Software version				
Object structure	-	**type of data**	-	**data range**	-

Can map	NO	**Accessibility**	RO	**Factory default**	3761
Describe the software version number					

Object 1018h : ID Object (Identity Object)					
Index	1018h				
Name	ID object				
Object structure	REC	**type of data**	-	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	-

Subindex	00h				
Name	Child index number (number of entries)				
Object structure	-	**type of data**	UINT8	**data range**	4
Can map	NO	**Accessibility**	R O	**Factory setting**	4

Subindex	01h				
Name	Vendor ID (Vendor-ID)				
Object structure	-	**type of data**	UINT 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	0x850104
Uniform distribution by ETG manufacturers					

Subindex	02 h				
name	Device code (Product Code)				
Object structure	-	**type of data**	UINT 32	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory setting**	0x26483052

The device code corresponds to the product series and product model of the electronic tag, and the corresponding relationship is as follows:

Bit	0~15	16~31
meaning	Product number	Product Series

Subindex	03h				
name	Revision number (Revision Number)				
Object structure	-	**type of data**	UINT 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	0x37613708

Indicates the upgrade record of the driver software

Bit	0~15	16~31
meaning	Minor revision	Main revision

Subindex	04h				
name	Serial number (Serial number)				
Object structure	-	**type of data**	UINT32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Indicates the software update time (manufacturer's own definition)					

Object 1600h : RPDO1 mapping object (1st Receive PDO Mapping)					
index	1600h				
name	RPDO1 mapping object				
Object structure	REC	**type of data**	UINT32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	RPDO1 map number (RPDO1 Mapping Numbers)				
Object structure	-	**type of data**	UINT8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	4
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	UINT32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60400010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	UINT32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x607A0020

Subindex	03h
name	The third mapping object (Mapping entry (3))

Object structure	-	**type of data**	UINT32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60FF0020

Subindex	04h				
name	The fourth mapping object (Mapping entry (4))				
Object structure	-	**type of data**	UINT32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60710010

Subindex	05h~08h				
name	Of 5-8 mappings objects (Mapping entry (5) ~ (8))				
Object structure	-	**type of data**	UINT32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1601h : RPDO2 mapping object (2nd Receive PDO Mapping)					
index	1601h				
name	RPDO1 mapping object				
Object structure	REC	**type of data**	UINT32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	RPDO2 map number (RPDO2 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	2
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60400010

Subindex	02h
name	The second mapping object (Mapping entry (2))

Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x607A0020

Subindex	03h~08h				
name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1602h : RPDO3 mapping object (3rd Receive PDO Mapping)					
index	1602h				
name	RPDO1 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	RPDO3 map number (RPDO3 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	2
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60400010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60FF0020

Subindex	03h~08h
name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))

Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1603h : RPDO4 mapping object (4th Receive PDO Mapping)					
index	1603h				
name	RPDO4 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	RPDO4 map number (RPDO4 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	2
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60400010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60710020

Subindex	03h~08h				
name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1A00h : TPDO1 mapping object (1st Transmit PDO Mapping)					
index	1A00h				
name	TPDO1 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	TPDO1 map number (TPDO1 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	4
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60410010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60640020

Subindex	03h				
name	The third mapping object (Mapping entry (3))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x606C0020

Subindex	04h				
name	The fourth mapping object (Mapping entry (4))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60770010

Subindex	05h~08h

name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1A01h : TPDO2 mapping object (2nd Transmit PDO Mapping)					
index	1A01h				
name	TPDO2 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	TPDO2 map number (TPDO2 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	3
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60410010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60640020

Subindex	03h				
name	The third mapping object (Mapping entry (3))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60780010

Subindex	04h~08h

name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1A02h : TPDO3 mapping object (3rd Transmit PDO Mapping)					
index	1A02h				
name	TPDO3 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	TPDO3 map number (TPDO3 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	3
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60410010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60640020
Subindex	03h				
name	The third mapping object (Mapping entry (3))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x606C0020

Subindex	04h~08h				
name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5

Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1A03h : TPDO4 mapping object (4th Transmit PDO Mapping)					
index	1A03h				
name	TPDO4 mapping object				
Object structure	REC	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	RW	**Factory default**	-
Function description: This object can only be modified if the PDO is invalid.					

Subindex	00h				
name	TPDO3 map number (TPDO3 Mapping Numbers)				
Object structure	-	**type of data**	Uint 8	**data range**	0~8
Can map	NO	**Accessibility**	R W	**Factory setting**	3
Function description: When writing 0 , other sub-index mapping objects are invalid					

Subindex	01h				
name	The first mapping object (Mapping entry (1))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60410010

Subindex	02h				
name	The second mapping object (Mapping entry (2))				
Object structure	-	**type of data**	Uint 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60640020
Sub index	03h				
name	The third mapping object (Mapping entry (3))				
Object structure	-	**type of data**	UINT 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	0x60770020

Sub index	04h~08h				
name	Of 3 to 8 mappings objects (Mapping entry (3) ~ (8))				
Object structure	-	**type of data**	UINT 32	**data range**	0~ 429496729 5
Can map	NO	**Accessibility**	R W	**Factory setting**	-

Object 1C00h : Sync manager type

index	1C00h				
name	Synchronous management type				
Object structure	REC	**type of data**	-	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	-

Sub index	00h				
name	Synchronize management of the largest sub-index number				
Object structure	-	**type of data**	UINT 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	4

Sub index	01h				
name	SM0 communication type (Communication Type SM0)				
Object structure	-	**type of data**	UINT 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	1
SM0 communication type: receiving mailbox					

Subindex	02h				
name	SM1 communication type (Communication Type SM1)				
Object structure	-	**type of data**	Uint 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2
SM1 communication type: send mailbox					
Subindex	03h				
name	SM2 communication type (Communication Type SM2)				
Object structure	-	**type of data**	Uint 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	3
SM2 communication type: process data output					

Subindex	04h				
name	SM3 communication type (Communication Type SM3)				
Object structure	-	**type of data**	Uint 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	4
SM3 communication type: process data input					

Object 1C12h : Synchronous Management 2_RPDO Assignment (Sync Manager 2 RPDO Assignment)	
index	1C12h

name	Synchronous management 2_RPDO allocation				
Object structure	ARR	type of data	Uint 16	data range	-
Can map	NO	Accessibility	RW	Factory default	-
Set the object index of the RPDO allocation					

Subindex	00h				
name	Synchronize the maximum sub-index number assigned by 2_RPDO				
Object structure	-	type of data	Uint 8	data range	-
Can map	NO	Accessibility	R W	Factory setting	1

Subindex	01h				
name	RPDO assigned object index (Index of RPDO Assignment)				
Object structure	-	type of data	Uint 8	data range	-
Can map	NO	Accessibility	R W	Factory setting	1
Set the RPDO allocation index: 1. Must be configurable in pre-run state 2. If you use the TwinCAT software to directly select the RPDO allocation, otherwise: a. 1C12-00h write value 0 b. 1C12-01h writes the pre-used RPDOx (1600h~1603h) and configures the RPDOx mapping object (such as 1600h) c. 1C12-00h write value 1					

Object 1C13h : Synchronous Management 2_TPDO Assignment (Sync Manager 2 TPDO Assignment)					
index	1C13h				
name	Synchronous management 2_TPDO allocation				
Object structure	ARR	type of data	Uint 16	data range	-
Can map	NO	Accessibility	RW	Factory default	-
Set the object index of the TPDO allocation					

Subindex	00h				
name	Synchronous management of the maximum sub-index number assigned by 2_TPDO				
Object structure	-	type of data	Uint 8	data range	-
Can map	NO	Accessibility	R W	Factory setting	1

Subindex	01h
name	TPDO assigned object index (Index of TPDO Assignment)

Object structure	-	**type of data**	Uint 8	**data range**	-
Can map	NO	**Accessibility**	R W	**Factory setting**	1
Set the TPDO allocation index: 1. Must be configurable in pre-run state 2. If you use TwinCAT software to directly select the allocation of TPDO, otherwise: a. 1C13-00h write value 0 b. 1C13-01h writes the pre-used TPDOx (1A00h~1A03h) and configures the TPDOx mapping object (such as 1A00h) c. 1C13-00h write value 1					

Object 1C32h : synchronization management 2_ synchronization parameter output (Sync Output Manager Paramater 2)					
index	1C32h				
name	Synchronous management 2_sync output parameters				
Object structure	REC	**type of data**	-	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	-
Describe the output parameters of SM2					

Sub index	00h				
name	Maximum and maximum sub index number of the synchronization management 2 synchronization parameter				
Object structure	-	**type of data**	UINT 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	32

Subindex	01h				
name	Type of synchronization (Synchronization Type)				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2
2 indicates that the synchronization type of SM2 is distributed clock synchronization 0 mode (DC SYNC0 Mode)					

Subindex	02h				
name	Cycle time (Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	0
Reflects the period of DC SYNC0					

Subindex	04h

name	Synchronization Types supported				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	4
Reflect distribution clock type 4 indicates support for distributed clock 0 mode (DC SYNC0)					

Subindex	05h				
name	The minimum cycle time (Minimum Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	125000
Reflects the minimum synchronization period supported by the slave (ns)					

Subindex	06h				
name	Calculation and copy time (Calc and Copy Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Reflects the time (ns) that the microprocessor copies data from synchronization management to the local					

Subindex	08h				
name	Acquisition cycle time (Get Cycle Time)				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-

Subindex	09h				
name	Delay Time (Delay Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Unit (ns)					

Subindex	0Ah				
name	SYNC0 Cycle Time (SYNC0 Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
When the clock mode is distributed, the value of the ESC register 09A0h is set (ns)					

Subindex	0Bh				
name	Number of synchronization events lost (SM-Event Missed)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-

Subindex	0Ch				
name	Loop count exceeded (Cycle Time Too Small)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Setting period is too small					

Subindex	20h				
name	Sync Error (SYNC Error)				
Object structure	-	**type of data**	BOOL	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
TURE: Synchronous activation with no errors FALSE : Synchronization is not activated or no synchronization error has occurred.					

Object 1C33h : synchronization management 2_ sync input parameters (Sync Manager Paramater INPUT 2)					
index	1C33h				
name	Synchronous management 2_synchronous input parameters				
Object structure	REC	**type of data**	-	**data range**	-
Can map	NO	**Accessibility**	RO	**Factory default**	-
Describe the input parameters of SM2					

Subindex	00h				
name	Maximum and maximum subindex number of the synchronization management 2 synchronization parameter				
Object structure	-	**type of data**	Uint 8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	32

Subindex	01h				
name	Type of synchronization (Synchronization Type)				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2

2 indicates that the synchronization type of SM2 is distributed clock synchronization 0 mode (DC SYNC0 Mode)

Subindex	02h				
name	Cycle time (Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	0
Reflects the period of DC SYNC0					

Subindex	04h				
name	Synchronization Types supported				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	4
Reflect distribution clock type 4 indicates support for distributed clock 0 mode (DC SYNC0)					

Subindex	05h				
name	The minimum cycle time (Minimum Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	125000
Reflects the minimum synchronization period supported by the slave (ns)					

Subindex	06h				
name	Calculation and copy time (Calc and Copy Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Reflects the time (ns) that the microprocessor copies data from synchronization management to the local					

Subindex	08h				
name	Acquisition cycle time (Get Cycle Time)				
Object structure	-	**type of data**	Uint 16	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-

Subindex	09h				
name	Delay Time (Delay Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-

Unit (ns)

Subindex	0Ah				
name	SYNC0 Cycle Time (SYNC0 Cycle Time)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
When the clock mode is distributed, the value of the ESC register 09A0h is set (ns)					

Subindex	0Bh				
name	Number of synchronization events lost (SM-Event Missed)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-

Subindex	0Ch				
name	Loop count exceeded (Cycle Time Too Small)				
Object structure	-	**type of data**	Uint 32	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
Setting period is too small					

Subindex	20h				
name	Sync Error (SYNC Error)				
Object structure	-	**type of data**	BOOL	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	-
TURE: Synchronous activation with no errors FALSE : Synchronization is not activated or no synchronization error has occurred.					

6.4.2 Detailed description of the 6000h group object

Object 603Fh : Error Code (Error Code)					
index	603Fh				
name	error code				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	NO	**Accessibility**	RO	**Factory default**	-
The most recent error in the fault code drive. See the fault list for details.					

Object 6040h : Control Word (Control Word)	
index	6040h

name	Control word				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	0

Control word definition:

Bit	meaning	description
0	Servo ready	0- invalid ; 1- valid
1	Turn on the main loop	0- invalid; 1- valid
2	Rapid shutdown	0- valid; 1- invalid
3	Servo operation	0- invalid ; 1- valid
4~6	Related to mode	-
7	Fault reset	Bit7 rising edge is valid Bit7 remains at 1 , other control commands are invalid.
8	time out	
9~10	NA	Reserved
11~15	Factory customization	Reserved

Bits 4~6 are defined:

Bit	Operating mode			
	Contour position	Contour speed	Contour torque	Zero return mode
4	New location Rising edge trigger	Reserved	Reserved	Zero return
5	0: Not updated immediately 1 : Update now	Reserved	Reserved	Reserved
6	0: absolute position 1 : Relative position	Reserved	Reserved	Reserved

Object 6041h : status word (the Status Word)					
index	6041h				
name	Status word				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	0

Reflect the servo status:

Bit	name	Bit definition
0	Servo no fault	1- valid; 0- invalid
1	Waiting to turn on the servo enable	1- valid; 0- invalid
2	Servo operation	1- valid; 0- invalid
3	malfunction	0: no fault 1 : There is a fault
4	Turn on the main loop	1- valid; 0- invalid
5	Rapid shutdown	0- valid; 1- invalid
6	Servo is not operational	1- valid; 0- invalid
7	caveat	0 : no warning; 1 : warning
8	Factory customization	
9	remote control	0 - Non- CAN open mode. 1-CANopen remote control mode.
10	Target arrival	0 - The target position or speed has not arrived. 1- Target position or speed arrives.
11	Software internal location is out of limits	The 0- position command or feedback did not reach the software internal position limit. 1- position command or feedback reaches the software internal position limit
12	Zero speed signal	Speed mode: 0 -non-zero speed 1- zero speed Position mode: 0- positioning completed 1- positioning is not completed
13		Origin zero return failure flag: 0: zero error does not occur 1: Error occurs when returning to zero (zero return mode, zero return timeout)
14	NA	Reserved
15	Zero return to origin	0- Home return is not performed or not completed. 1- The zero point of the origin has been completed and the reference point has been found.

Object 6060h : mode selection (Modes of Operation)	
index	6060h
name	Mode selection

Object structure	VAR	**type of data**	Int8	**data range**	0~10
Can map	RPDO	**Accessibility**	RW	**Factory default**	8

Select the servo operation mode:

Set value	Description
0	NA
1	Contour position mode (pp)
2	NA
3	Contour speed mode (pv)
4	Contour torque mode (pt)
5	NA
6	Zero return mode (hm)
7	NA
8	Cycle sync position mode (csp)
9	Cycle Synchronous Speed Mode (csv)
10	Periodic synchronous torque mode (cst)

Object 6061h : Modes of Operation Display					
index	6061h				
name	Operating mode display				
Object structure	VAR	**type of data**	Int8	**data range**	0~10
Can map	TPDO	**Accessibility**	RO	**Factory default**	0

Select the servo operation mode:

Set value	Description
0	NA
1	Contour position mode (pp)
2	NA
3	Contour speed mode (pv)
4	Contour torque mode (pt)
5	NA
6	Zero return mode (hm)
7	NA
8	Cycle sync position mode (csp)
9	Cycle Synchronous Speed Mode (csv)
10	Periodic synchronous torque mode (cst)

Object 6062h : Position Demand Value

index	6062h				
name	Position command				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~(2^{31}-1)
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Reflect real-time position command (unit: command unit)					

Object 6063h : Position Actual Value					
index	6063h				
name	Motor position feedback				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~(2^{31}-1)
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Reflect real-time position feedback (unit: encoder unit)					

Object 6064h : Position Actual Value					
index	6064h				
name	Position feedback				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~(2^{31}-1)
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Reflect real-time position feedback (unit: command unit)					

Object 6065h : Position deviation excessive threshold (Following error Window)					
index	6065h				
name	Position deviation excessive threshold				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~(2^{31}-1)
Can map	TPDO	**Accessibility**	RW	**Factory default**	-
Set the position deviation too large threshold (command unit). When the difference between the user position command 6062h and the user position feedback 6064h exceeds ±6065h , a positional deviation excessive fault occurs. When 6065h is set to (2^{32}-1), the servo does not perform excessive position deviation monitoring.					

Object 6067h : Positional arrival threshold (Position Window)					
index	6067h				
name	**Location arrival threshold**				
Object structure	VAR	**type of data**	Uint32	**data range**	0~(2^{32}-1)

Can map	RPDO	**Accessibility**	RW	**Factory default**	-
Set the threshold for the arrival of the position (unit: command unit). The difference between the user position command 6062h and the user actual position feedback 6064h is within ±6067h , and when the time reaches 6068h , the position is considered to arrive, and in the contour position mode, the bit 10 of the status word 6041h is =1 .					

Object 6068h : Motor Position Time Window (Position Window Time)					
index	6068h				
name	Motor position arrival window time				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
Set the time window for the decision position to arrive (unit: 2ms). The difference between the user position command 6062h and the user actual position feedback 6064h is within ±6067h , and when the time reaches 6068h , the position is considered to arrive, and in the contour position mode, the bit 10 of the status word 6041h is =1 .					

Object 606Bh : User's actual speed command (Velocity Demand Value)					
index	6068h				
name	User actual speed command				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~$(2^{31}-1)$
Can map	TPDO	**Accessibility**	RO	**Factory default**	0
Reflects the user's actual speed command (unit: 1rpm). In the position class mode, a speed command corresponding to the output of the position adjuster is reflected; In the speed mode, the input command of the speed regulator is reflected.					

Object 606Ch : User's actual speed feedback (Velocity Actual Value)					
index	606Ch				
name	User actual speed feedback				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~$(2^{31}-1)$
Can map	TPDO	**Accessibility**	RO	**Factory default**	0
Reflects the user's actual speed feedback value (unit: command unit / s).					

Object 606Dh : Speed reaches the threshold speed (Velocity Window)	
index	606Dh
name	Speed reaches threshold speed

Object structure	VAR	**type of data**	Uint16	**data range**	0~3000
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
Set the threshold for speed arrival (unit: 1 rpm). Target speed 60FFh user actual speed 606Ch is a difference ±606Dh less, and the time to reach 606Eh time, that the speed reaches the speed profile mode, the status word 6041h of Bit10 = . 1. Otherwise the status word 6061h of Bit10 = 0 .					

Object 606Eh : Velocity Window Time					
index	606Eh				
name	Speed arrival time window				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
Set the time window in which the judgment speed reaches the valid (unit: ms). Target speed 60FFh user actual speed 606Ch is a difference ±606Dh less, and the time to reach 606Eh time, that the speed reaches the speed profile mode, the status word 6041h of Bit10 = . 1. Otherwise the status word 6061h of Bit10 = 0 .					

Object 606Fh : zero speed threshold (Velocity Threshold)					
index	606Fh				
name	Zero speed threshold				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	10
Set the threshold (unit: 1 rpm) for determining whether the user speed is 0. User feedback speed 606Ch in ± 606Fh within, and the time to reach 6070h set value indicates that the user rate is 0 , case status word 6041h of bit12 = 1; do not satisfy any of the conditions in both, speed is not considered that the user is 0 , this when the status word 6041h of bit1 2 = 0 .					

Object 6070h : zero speed the time window (Zero the Velocity the Threshold Time)					
index	6070h				
name	Zero speed time window				
Object structure	VAR	**type of data**	Uint16	**data range**	0~65535
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
Set the time window (unit: 2 ms) used to determine if the user's speed is zero.					

User feedback speed 606Ch in ± 606Fh within, and the time to reach 6070h set value indicates that the user rate is 0 , case status word 6041h of bit12 = 1; do not satisfy any of the conditions in both, speed is not considered that the user is 0 , this when the status word 6041h of bit1 2 = 0 .

Object 6071h : target torque (Target torque)					
index	6071h				
name	Target torque				
Object structure	VAR	**type of data**	Int16	**data range**	**-3000~3000**
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
Only used for contour torque mode, torque command target value (unit: 0.1%).					

Object 6074h : a torque command (the Torque Demand the Value)					
index	6074h				
name	Torque command				
Object structure	VAR	**type of data**	Int16	**data range**	-
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Displays the current torque command (unit: 1%).					

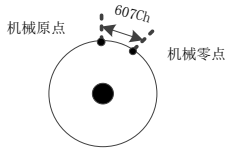
Object 6075h : nominal motor current (Motor Rated Current)					
index	6075h				
name	Motor rated current				
Object structure	VAR	**type of data**	Uint32	**data range**	-
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Rated current on the motor nameplate (unit: mA). All current-related parameter values are related to this parameter.					

Object 6076h : Motor rated torque (Motor Rated Torque)					
index	6076h				
name	Motor rated torque				
Object structure	VAR	**type of data**	Uint32	**data range**	$0\sim(2^{32}-1)$
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Rated torque on the motor nameplate (unit: m Nm). All torque related parameter values are related to this parameter.					

Object 6077h : motor feedback torque (T orque Actual the Value)	
index	6077h
name	Motor feedback torque

Object structure	VAR	**type of data**	Int16	**data range**	-
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
The instantaneous torque output of the response servo motor (unit: 1%).					

Object 607Ah : target position (Target Position)					
index	607Ah				
name	target location				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~$(2^{31}-1)$
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Set the servo target position (unit: command unit) in the contour position mode and cycle synchronization position mode .					

Object 607Ch : Home Offset (Hmoe Offset)					
index	607Ch				
name	Origin offset				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~$(2^{31}-1)$
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
In the position class control mode, the mechanical zero deviates from the physical position of the motor origin (unit: encoder unit). Mechanical origin = mechanical zero + 607Ch (origin offset). When set to 0, the origin is unbiased. 					

Object 607Dh : Software absolute position limit (Software Limit position)					
index	607Dh				
name	Software absolute position limit				
Object structure	VAR	**type of data**	Int32	**data range**	-
Can map	RPDO	**Accessibility**	RW	**Factory default**	-
Set the minimum and maximum values for the absolute position limit of the software. Minimum absolute position limit = (607D-1h) Maximum absolute position limit = (607D-2h) Software absolute position limit setting: 1. When both (607D-1h) and (607D-2h) are set to the default values, the software limit does not take effect.					

2. AL.931 warning will occur when the minimum absolute position limit (607D-1h) is greater than the maximum absolute position limit (607D-2h).

3. When the position command or position feedback reaches the software limit value, in the position mode, the servo will operate with the position limit as the target position, stop at the position limit, and prompt the overtravel warning AL.950. Entering the reverse command will cause the motor to exit the position overrun condition.

Subindex	00h				
name	Number of sub-indexes for absolute position restrictions				
Object structure	-	**type of data**	Uint8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2

Subindex	01h				
name	Minimum software absolute position limit (Min Position Limit)				
Object structure	-	**type of data**	Uint8	**data range**	-2^{31}~$(2^{31}-1)$
Can map	RPDO	**Accessibility**	R W	**Factory setting**	-2^{31}

Subindex	02h				
name	Maximum software absolute position limit (Max Position Limit)				
Object structure	-	**type of data**	Uint8	**data range**	-2^{31}~$(2^{31}-1)$
Can map	RPDO	**Accessibility**	R W	**Factory setting**	$2^{31}-1$

Object 607Eh : Command Polarity (Polarity)					
index	607Eh				
name	Command polarity				
Object structure	VAR	**type of data**	Uint8	**data range**	0-255
Can map	RPDO	**Accessibility**	RW	**Factory default**	0

Reflects the polarity of the position command or speed command.

B it	7	6	5	4~0
meaning	Position command polarity	Speed command polarity	Torque command polarity	Reserved

Bit7=1, indicating the standard position mode, the position command ×(-1), the motor is reversed. In the contour position mode and the cycle synchronous position mode, the position command and the target position are inverted.

Bit6=1 means that the speed command (60FFh) ×(-1) will be reversed in the speed mode .

Object 607Fh : Max Profile Velocity	
index	607Fh
name	Maximum contour speed

Object structure	VAR	**type of data**	Uint32	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	50000
Maximum user speed (unit: 0.1rpm). The set value takes effect when the slave speed command changes.					

Object 6080h : Max Motor Velocity					
index	6080h				
name	Maximum motor speed				
Object structure	VAR	**type of data**	**Uint32**	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	6000
The maximum permissible operating speed of the motor can be obtained from the instruction manual of the servo motor (unit: rpm)					

Object 6081h : Profile Velocity					
index	6081h				
name	Contour speed				
Object structure	VAR	**type of data**	**Uint32**	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	10000000
In the contour position mode, the running speed of the uniform speed section reached after the acceleration section is completed (unit: command unit / s). $$\text{motor speed} = \frac{6081\text{h} \times \frac{6091\text{h}-1}{6091\text{h}-2}}{\text{Encoder resolution}} \times 60$$					

Object 6083h : Acceleration profile (Profile Acceleration)					
index	6083h				
name	Contour acceleration				
Object structure	VAR	**type of data**	**Uint32**	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	200
Lower profile mode, from the motor 0rpm accelerated to the speed reference acceleration (unit: Command / $S^{2)}$. After the slave receives the segment shift command, the set value takes effect.					

Object 6084h : Profile Deceleration	
index	6084h

name	Profile deceleration				
Object structure	VAR	**type of data**	**Uint32**	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	200
The speed profile mode, from the motor speed is decelerated to the current 0rpm deceleration (unit: Command / S $^{2)}$. After the slave receives the segment shift command, the set value takes effect.					

Object 6086h : Motor Run Curve Type					
index	6086h				
name	Motor running curve type				
Object structure	VAR	**type of data**	**Int16**	**data range**	-2^{15}~(2^{15}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
The type of curve for the motor position command or speed command. 0- linear					

Object 6087h : torque ramp (the Torque S Lope)					
index	6087h				
name	Torque ramp				
Object structure	VAR	**type of data**	**UInt32**	**data range**	0~(2^{32}-1)
Can map	RPDO	**Accessibility**	RW	**Factory default**	0
In the contour torque mode, it indicates the time (in ms) that the torque given amount has risen from 0% of the rated torque to 100% .					

Object 6091h : gear ratio (Gear Ratio)					
index	6091h				
name	Gear ratio				
Object structure	VAR	**type of data**	**UInt32**	**data range**	-
Can map	RPDO	**Accessibility**	RW	**Factory default**	-
The position factor is used to establish the proportional relationship between the load displacement specified by the user and the motor displacement: Motor displacement (motor unit) = load displacement (user unit) × position factor The position factor setting is related to the mechanical reduction ratio, mechanical size related parameters, and motor resolution. The calculation method is as follows:					

$$\text{Position factor} = \frac{\text{Motor resolution} \times \text{gear ratio}}{\text{Load feed}}$$

Subindex	00h				
name	**Number of sub-indexes**				
Object structure	VAR	**type of data**	Uint8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2

Subindex	01h				
name	**Motor resolution (Motor Revolutions)**				
Object structure	VAR	**type of data**	Uint32	**data range**	$1\sim(2^{32}-1)$
Can map	RPDO	**Accessibility**	R W	**Factory setting**	1

Subindex	02h				
name	**Resolution shaft (Shaft Revolutions)**				
Object structure	VAR	**type of data**	Uint32	**data range**	$1\sim(2^{32}-1)$
Can map	RPDO	**Accessibility**	R W	**Factory setting**	1

Object 6098h : zero modes (Homing Method,)					
index	6098h				
name	**Zero return mode**				
Object structure	VAR	**type of data**	Int8	**data range**	1~35
Can map	RPDO	**Accessibility**	RW	**Factory default**	1

Select the origin return method:

value	Description
1	Regression with negative limit switch and Z pulse signal
2	Regression with positive limit switch and Z pulse signal
3,4	Regression with positive origin switch and Z pulse signal
5,6	Regression to negative origin switch and Z pulse signal
7~14	Regression with the origin switch and Z pulse signal
15~16	Reserved
17~30	Regression without reference to the Z pulse signal
31~32	Reserved
33~34	Regression without reference to the Z pulse signal
35	Reset with current position

Object 6099h : Homing Speeds

index	6099h				
name	**Zero return speed**				
Object structure	ARR	**type of data**	-	**data range**	-
Can map	YES	**Accessibility**	RW	**Factory default**	-
In the zero return mode, the two speed value settings are included: 6099-1h search deceleration point signal speed (unit: command unit / s); 6099-2h Search for the origin signal speed (unit: command unit / s).					

Subindex	00h				
name	**Number of sub-indexes**				
Object structure	VAR	**type of data**	Uint8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2

Subindex	01h				
name	Search deceleration point signal speed (Speed During Search for Switch)				
Object structure	VAR	**type of data**	Uint32	**data range**	**0 ~(2^{32}-1)**
Can map	RPDO	**Accessibility**	R W	**Factory setting**	27962026

Subindex	02h				
name	Search for the speed of the origin signal (Speed During Search for Zero)				
Object structure	VAR	**type of data**	Uint32	**data range**	**0 ~(2^{32}-1)**
Can map	RPDO	**Accessibility**	R W	**Factory setting**	5592405

Object 609Ah : Homing Acceleration					
index	609Ah				
name	**Zero return acceleration**				
Object structure	ARR	**type of data**	**Uint32**	**data range**	**0 ~(2^{32}-1)**
Can map	RPDO	**Accessibility**	RW	**Factory default**	16000
Set the acceleration in the home return mode. The object dictionary unit is defined as the position instruction increment per second (command unit / s^2) .					

Object 60B0h : position offset (the Position offset)					
index	60B0h				
name	**Position offset**				
Object structure	ARR	**type of data**	**Int32**	**data range**	**-2^{31}~(2^{31}-1)**

Can map	RPDO	Accessibility	RW	Factory default	0
Used to synchronize the position command offset amount in the cycle position mode, servo target position = 607Ah + 60B0h .					

Object 60B8h : Probe Function (Touch Probe Function)					
index	60B8h				
name	Probe function				
Object structure	VAR	type of data	Uint16	data range	0
Can map	RPDO	Accessibility	RW	Factory default	0

I.e. the probe feature position latch function, can latch the external DI signal or the electrical machine Z position information of the signals change. This servo supports two probe functions to latch 4 position information. Probe 1 can be selected DI44 or Z signal as a probe signal, the probe 2 can select DI45 or Z signal as a probe signal.

The function of probe 1 and probe 2:

Bit	description	Range
0	Probe 1 enable	0--- Probe 1 is not enabled 1--- Probe 1 enabled
1	Probe 1 trigger mode	0--- single trigger 1--- Continuous trigger
2	Probe 1 trigger signal selection	0---DI44 input signal 1---Z signal
3	NA	
4	Probe 1 rising edge, falling edge selection	0--- falling edge latch 1--- rising edge latch
5-7	NA	
8	Probe 2 enable	0--- Probe 2 is not enabled 1--- Probe 2 enabled
9	Probe 2 trigger mode	0--- single trigger 1--- Continuous trigger
10	Probe 2 trigger signal selection	0---DI45 input signal 1---Z signal
11	NA	
12	Probe 2 rising edge, falling edge selection	0--- falling edge latch 1--- rising edge latch
13-15	NA	

Object 60B9h : Probe Status (Touch Probe Status)

index	60B9h				
name	Probe status				
Object structure	VAR	type of data	Uint16	data range	0
Can map	TPDO	Accessibility	RO	Factory default	0

The state of probe 1 and probe 2.

Bit	description	Remarks
0	0--- Probe 1 is not enabled 1--- Probe 1 enabled	
1	0 — Probe 1 rising edge latch is not executed 1 — Probe 1 rising edge latch has been executed	
2	0 — Probe 1 falling edge latch is not executed 1 — Probe 1 falling edge latch has been executed	
3~5	NA	
6	0 — DI44 input signal 1 — Z signal	
7	0 — DI44 is low 1 — DI44 is high	
8	0--- Probe 2 is not enabled 1--- Probe 2 enabled	
9	0 — Probe 2 rising edge latch is not executed 1 — Probe 2 rising edge latch has been executed	
10	0 — Probe 2 falling edge latch is not executed 1 — Probe 2 falling edge latch has been executed	
11~13	NA	
14	0 — DI45 input signal 1 — Z signal	
15	0 — DI45 is low 1 — DI45 is high	

Object 60BAh : Probe 1 rising edge position feedback (Touch Probe Pos1 Pos Value)	
index	60Bah
name	**Probe 1 rising edge position feedback**

Object structure	VAR	**type of data**	**Int32**	**data range**	**-**
Can map	TPDO	**Accessibility**	RO	**Factory default**	**-**
Displays the rising edge of the probe 1 signal and the position feedback (command unit).					

Object 60BBh : Probe 1 Falling Edge Position Feedback (Touch Probe Pos1 Neg Value)					
index	60BBh				
name	**Probe 1 falling edge position feedback**				
Object structure	VAR	**type of data**	**Int32**	**data range**	**-**
Can map	TPDO	**Accessibility**	RO	**Factory default**	**-**
Displays the falling edge of the probe 1 signal and the position feedback (command unit).					

Object 60BCh : Probe 2 rising edge position feedback (Touch Probe Pos1 Pos Value)					
index	60BCh				
name	**Probe 2 rising edge position feedback**				
Object structure	VAR	**type of data**	**Int32**	**data range**	**-**
Can map	TPDO	**Accessibility**	RO	**Factory default**	**-**
Displays the rising edge of the probe 2 signal and the position feedback (command unit).					

Object 60BDh : Probe 1 Falling Edge Position Feedback (Touch Probe Pos2 Neg Value)					
index	60BDh				
name	**Probe 2 falling edge position feedback**				
Object structure	VAR	**type of data**	**Int32**	**data range**	**-**
Can map	TPDO	**Accessibility**	RO	**Factory default**	**-**
Displays the falling edge of the probe 2 signal and the position feedback (command unit).					

Object 60E0h : Forward torque limit value (Positive Torque limit)					
index	**60E0h**				
name	Forward torque limit				
Object structure	VAR	**type of data**	**Uint16**	**data range**	**0~3000**
Can map	RPDO	**Accessibility**	RW	**Factory default**	3000
Limit the maximum forward torque (in 0.1%).					

Object 60E1h : Negative to the torque limit value (Negtive Torque limit)

index	60E1h				
name	Negative torque limit				
Object structure	VAR	type of data	Uint16	data range	0~3000
Can map	RPDO	Accessibility	RW	Factory default	3000
Limit the maximum negative torque (in 0.1%).					

Object 60F4h : positional deviation (Following Error Actual Value)					
index	60F4h				
name	Position deviation				
Object structure	VAR	type of data	Int32	data range	-2^{31}~$(2^{31}-1)$
Can map	TPDO	Accessibility	RO	Factory default	-
Display position deviation (command unit).					

Object 60FCh : position command (the Position the Value Demand)					
index	60FCh				
name	Position command				
Object structure	VAR	type of data	Int32	data range	-2^{31}~$(2^{31}-1)$
Can map	TPDO	Accessibility	RO	Factory default	-
Motor real-time position command Position command (6062h) × electronic gear ratio (609 1 h) = motor position command (60FCh)					

Object 60FDh : Digital Input					
index	60FDh				
name	Digital input				
Object structure	VAR	type of data	Uint32	data range	0~0xFFFFFFFF
Can map	TPDO	Accessibility	RO	Factory default	0

Reflects the current DI terminal logic of the drive, 0 means invalid, 1 means valid

The DI signals indicated by each of the following are as follows:

Bit	31~16	15~ 5	4	3	2	1	0
description	Factory customization	Reserved	DI 45 level	DI 44 level	Origin switch	Forward overtravel switch	Reverse overtravel switch

Object 60FEh : Digital Output (Digital Out put)	
index	60FEh

name	Digital output				
Object structure	ARR	**type of data**	Uint32	**data range**	-
Can map	TPDO	**Accessibility**	RO	**Factory default**	-
Reflects the current D O terminal logic of the drive , 0 means invalid, 1 means valid					

Subindex	00h				
name	**Number of sub-indexes**				
Object structure	VAR	**type of data**	Uint8	**data range**	-
Can map	NO	**Accessibility**	R O	**Factory setting**	2

Sub index	01h				
name	**Physical output**				
Object structure	VAR	**type of data**	Uint32	**data range**	0~0xFFFFFFFF
Can map	TPDO	**Accessibility**	R O	**Factory setting**	0

The D O signals indicated by each of the following are as follows:

Bit	description
0	Brake output
1~15	NA
16~31	Factory customization

Subindex	02h				
name	Physical output enable				
Object structure	VAR	**type of data**	Uint32	**data range**	0~0xFFFFFFFF
Can map	TPDO	**Accessibility**	R O	**Factory setting**	0
invalid					

Object 60FFh : Target Velocity					
index	**60FFh**				
name	Target speed				
Object structure	VAR	**type of data**	Int32	**data range**	-2^{31}~$(2^{31}-1)$
Can map	RPDO	**Accessibility**	RW	**Factory default**	-
Set the speed command (in unit: command unit / s) in the contour speed mode and synchronous cycle speed mode .					

Object 6052h : Support servo operation mode (Supported Drive Modes)

index	6052h				
name	Support servo operation mode				
Object structure	VAR	type of data	Uint32	data range	0~(2^{32}-1)
Can map	TPDO	Accessibility	RO	Factory default	-

The servo operation mode supported by the drive, 0 means not supported, 1 means support.

Bit	Described above	value
0	Contour position mode	1
1	Frequency control mode	0
2	Contour speed mode	1
3	Contour torque mode	1
4	NA	0
5	Zero return mode	1
6	Interpolation position mode	0
7	Cycle sync position mode (csp)	1
8	Cycle Synchronous Speed Mode (csv)	1
9	Periodic synchronous torque mode (cst)	1
1 0 ~ 31	Manufacturers custom defined	Pre- stay

7 Troubleshooting information

7.1 fault code

Fault display	Fault name	error code	Auxiliary code
Er.020	Parameter and check exception	0x6000	0x00000020
Er.021	Parameter formatting exception	0x6001	0x00000021
Er.022	System and check exception	0x600 2	0x00000022
Er. 023	XML file not burned	0x6003	0x00000023
Er.030	Main circuit detection part is abnormal	0x600 4	0x00000030
Er.040	Abnormal parameter setting	0x600 5	0x00000040
Er.041	Distribution pulse output setting is abnormal	0x600 6	0x00000041
Er.042	Abnormal combination of parameters	0x600 7	0x00000042
Er.044	Semi-closed loop / full closed loop parameter setting exception	0x600 8	0x00000044
Er.050	Drive does not match motor capacity	0x600 9	0x00000050
Er.051	Product does not support alarm	0x600 A	0x00000051
Er.080	Encoder unit pulse distance setting is abnormal	0x600B	0x00000080
Er.08A	Position sensor resolution setting is abnormal	0x600 C	0x0000008A

Er.0B0	Servo on command invalid alarm	0x600 D	0x000000B0
Er.100	Overcurrent (OC)	0x600 E	0x00000100
Er.300	Regeneration failure	0x600 F	0x00000300
Er.320	Regenerative overload	0x6010	0x00000320
Er.330	Main circuit power supply wiring error	0x601 1	0x00000330
Er.400	Main circuit overvoltage (OV)	0x601 2	0x00000400
Er.410	Main circuit under voltage	0x601 3	0x00000410
Er.42A	Converter exception	0x601 4	0x0000042A
Er.450	Main circuit capacitor overvoltage	0x601 5	0x00000450
Er.510	Over speed (OS)	0x601 6	0x00000510
Er.511	Crossover pulse output overspeed	0x601 7	0x00000511
Er.520	Vibration alarm	0x601 8	0x00000520
Er.521	Automatic tuning alarm	0x601 9	0x00000521
Er.550	The highest speed setting is abnormal	0x601 A	0x00000550
Er.710	Overload (instantaneous maximum load)	0x601 B	0x00000710
Fault display	**Fault name**	**error code**	**Auxiliary code**
Er.720	Overload (continuous maximum load)	0x601 C	0x00000720
Er.730	DB overload 1	0x601 D	0x00000730
Er.731	DB overload 2	0x601 E	0x00000731
Er.740	Inrush current limiting resistor overload	0x601 F	0x00000740
Er.7A0	Heat sink overheating	0x6020	0x000007A0
Er.7AA	Control the substrate temperature abnormality	0x602 1	0x000007AA
Er.7AB	Servo unit built-in fan stop	0x602 2	0x000007AB
Er.810	Encoder backup alarm	0x602 3	0x00000810
Er.820	Encoder and number verification alarm	0x602 4	0x00000820
Er.830	Encoder battery alarm	0x602 5	0x00000830
Er.840	Encoder data is abnormal	0x602 6	0x00000840
Er.850	Encoder overspeed	0x602 7	0x00000850
Er.860	Encoder overheating	0x602 8	0x00000860
Er.870	Encoder write error	0x60 78	0x00000870
Er.900	CAN main station dropped (life factor)	0x607 A	0x00000900
Er.901	CAN main station dropped (consumer time)	0x607 B	0x00000901
Er.902	CAN master station off monitoring line configuration conflict	0x607 C	0x00000902
Er.910	Critical point number error	0x606 D	0x00000910
Er.911	Spindle data error	0x606 E	0x00000911
Er.912	Spindle speed error	0x606 F	0x00000912
Er.913	Slave speed error	0x6070	0x00000913
Er.914	Line type error	0x607 1	0x00000914
Er.915	System operation error	0x607 2	0x00000915
Er.B10	Speed command A/D exception	0x603 1	0x00000B10

Er.B11	Speed command A/D conversion data is abnormal	0x603 2	0x00000B11
Er.B20	Torque command A/D exception	0x603 3	0x00000B20
Er.B31	Current detection fault 1 (U phase)	0x603 4	0x00000B31
Er.B32	Current detection fault 2 (V phase)	0x603 5	0x00000B32
Er.B33	Current detection fault 3 (current detector)	0x603 6	0x00000B33
Er.B40	Reserved	0x603 7	0x00000B40
Er.BE0	Reserved	0x603 8	0x00000BE0
Fault display	**Fault name**	**error code**	**Auxiliary code**
Er.BF0	System alarm 0	0x603 9	0x00000BF0
Er.BF1	System alarm 1	0x603 A	0x00000BF1
Er.BF2	System alarm 2	0x603 B	0x00000BF2
Er.BF3	System alarm 3	0x603 C	0x00000BF3
Er.BF4	Hardware overcurrent	0x603 D	0x00000BF4
Er.C10	Prevent out of control detection	0x603 E	0x00000C10
Er.C20	Phase error detection	0x603 F	0x00000C20
Er.C21	Hall sensor is abnormal	0x60 4 0	0x00000C21
Er.C22	Inconsistent phase information	0x6041	0x00000C22
Er.C50	Magnetic pole monitoring failed	0x604 2	0x00000C50
Er.C51	Magnetic pole monitoring stopped	0x604 3	0x00000C51
Er.C52	Magnetic pole monitoring is not finished	0x604 4	0x00000C52
Er.C53	Magnetic pole monitoring overtravel	0x604 5	0x00000C53
Er.C54	Magnetic pole monitoring failure 2	0x604 6	0x00000C54
Er.C80	Encoder clear exception	0x604 7	0x00000C80
Er.C90	Encoder communication failure	0x604 8	0x00000C90
Er.C91	Encoder communication position data acceleration abnormal	0x604 9	0x00000C91
Er.C92	Encoder communication timer is abnormal	0x604 A	0x00000C92
Er.CA0	Encoder parameter exception	0x604 B	0x00000CA0
Er.CB0	Encoder loopback check exception	0x604 C	0x00000CB0
Er.CC0	Inconsistent upper and lower limits of the number of revolutions	0x604 D	0x00000CC0
Er.CF1	Reserved	0x604 E	0x00000CF1
Er.CF2	Reserved	0x604 F	0x00000CF2
Er.D00	Position deviation is too large	0x6050	0x00000D00
Er.D01	Excessive position deviation when servo ON	0x605 1	0x00000D01
Er.D02	Excessive positional deviation caused by speed limit when servo is ON	0x605 2	0x00000D02
Er.D10	Motor - load position deviation is too large	0x605 3	0x00000D10
Er.D30	Location data overflow	0x605 4	0x00000D30
Er.EB9	EtherCAT initialization exception	0x606 C	0x00000EB9
Er.EC6	Reserved	0x607 3	0x00000EC6

Er.F10	Phase loss when the main circuit selects three-phase power input	0x607 7	0x00000F10

7.2 warning code

Fault display	Fault name	error code	Auxiliary code
AL.900	Position deviation is too large	0x900	0x700
AL.901	Excessive position deviation when servo ON	0x901	0x701
AL.910	Overload warning	0x910	0x710
AL.911	Vibration warning	0x911	0x711
AL.920	Regenerative overload warning	0x920	0x720
AL.921	DB overload warning	0x921	0x7 21
AL.930	Battery undervoltage warning	0x930	0x7 3 0
AL.931	The software limit 607Dh is abnormally set.	0x931	0x7 31
AL.940	The origin is offset outside the soft limit	0x940	0x7 4 0
AL.941	Parameter change warning that requires power failure	0x941	0x7 41
AL.942	EtherCAT control mode setting is abnormal	0x942	0x7 42
AL.950	Position exceeds the soft limit	0x950	0x7 50
AL.971	Undervoltage warning	0x971	0x700
AL.9A0	Overtravel warning	0x9A0	0x7 A0
AL .9B0	SYNC0 sync frame is lost	0x9B0	0x7 B0
AL.9F0	Distributed clock cycle setting exception (125us integer multiple)	0x9F0	0x7 F0

7.3 Communication failure recovery method

For the fault of the servo drive itself, please refer to the corresponding manual. This section only describes the processing method of the EtherCAT communication part.

1) Er.023: Unburned xml configuration file

the reason	Treatment
Unburned configuration file	Burn configuration file
Drive failure	Replace the servo drive

2) Er.EB9: EtherCAT initialization exception

the reason	Treatment
Unburned configuration file	Burn configuration file
Drive failure	Replace the servo drive
Unburned FPGA code	Burn FPGA code

3) AL.931: The software limit 607D is set abnormally.

the reason	Treatment
The minimum limit value of 607Dh is greater than the maximum limit value.	Change 607Dh maximum limit value or minimum limit value

4) AL.940: The origin is offset outside the soft limit

the reason	Treatment
607Ch is set larger than the 607Dh limit value	Change the value of 607Dh or 607Ch

5) AL.942: EtherCAT control mode setting is abnormal

the reason	Treatment
Object dictionary 6060h is set incorrectly	Modify the setting value of 6060h

6) AL.950: Position exceeds the soft limit

the reason	Treatment
Target position 607Ah or actual position 6064h exceeds the limit of 607Dh	Change the value of 607Dh or 607Ah

7) AL.9b0: SYNC0 sync frame is lost

the reason	Treatment
Slave receives an exception during synchronous communication	1. Use shielded twisted pair 2. View the wiring status through the digital display of the keyboard
The master sends an exception when synchronizing communication	Replace the master station equipment

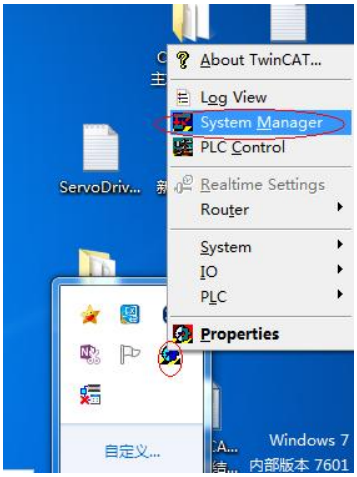
8) AL.9F0: Distribution clock cycle setting is abnormal

the reason	Treatment
The synchronization period is not an integer multiple of 125us	Modify the synchronization period setting to an integer multiple of 125us

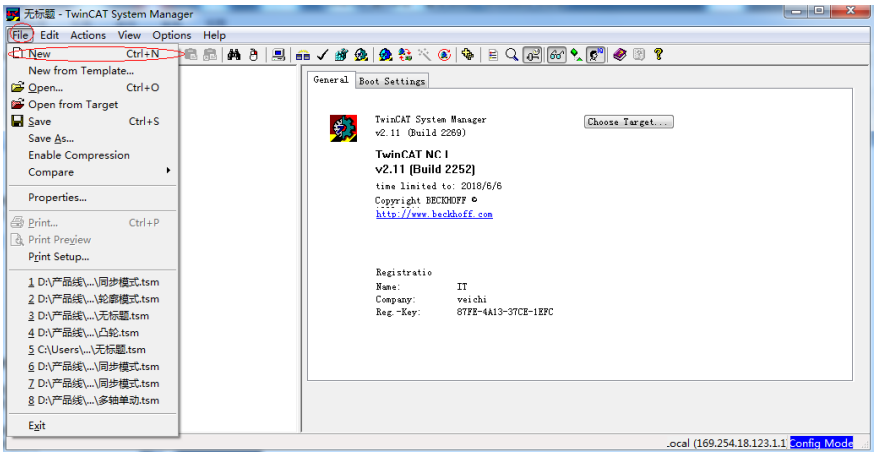
8 with the Beckhoff main station operation example

The following example is a system configuration using Beckhoff TwinCAT 2.110.2252 software with Beckhoff controller CX9020. Please install the TWinCAT software correctly before configuring the software.

1. Firstly connect the servo driver correctly and supply power, and set the first digit of function code Pn000 to 1.
2. Copy the VEICHI XML file to the folder where TwinCAT is installed (the path is usually: C: \TwinCAT\IO\EtherCAT);
3. Restart TwinCAT.
4. Open the TwinCAT System Manager as shown below and start the configuration process.

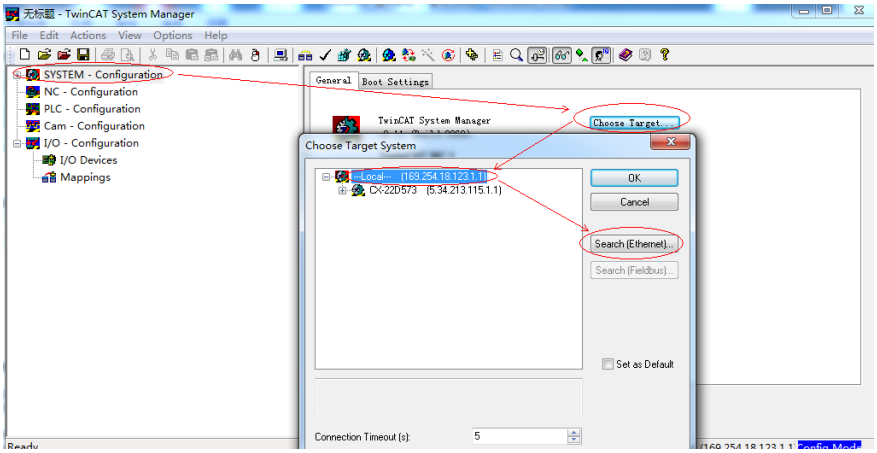


5. New project [" FILE " → " New "]

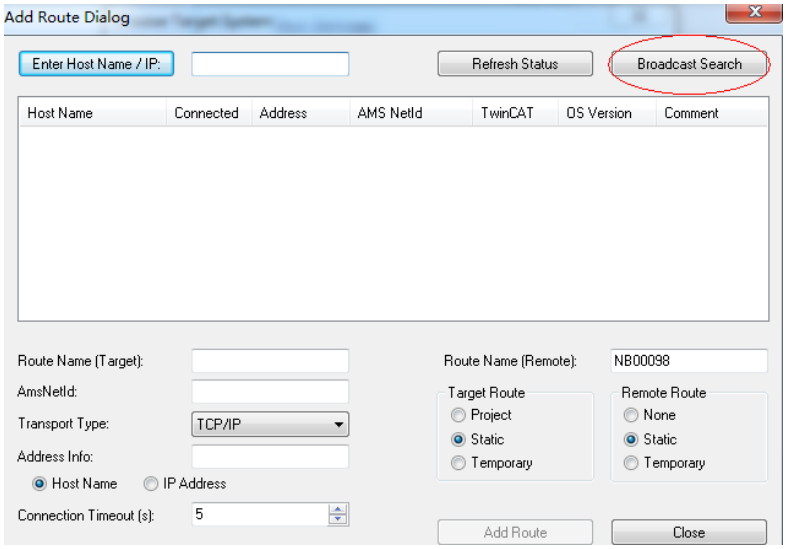


6. Connect the CX9020 and the servo drive and supply power.

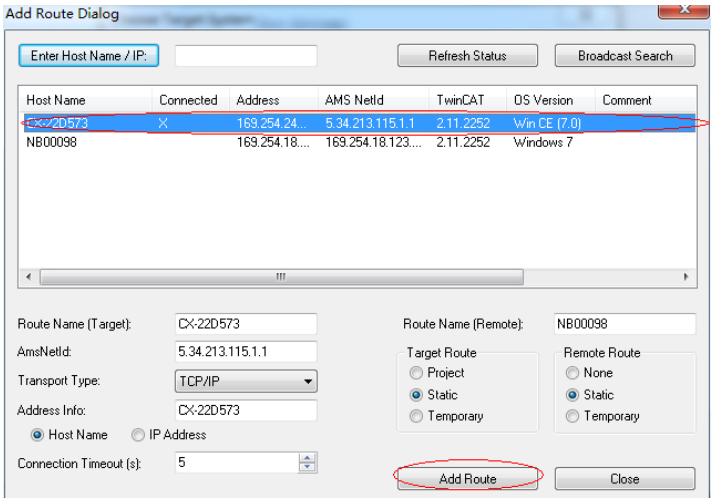
7. Start searching for the CX9020 controller.



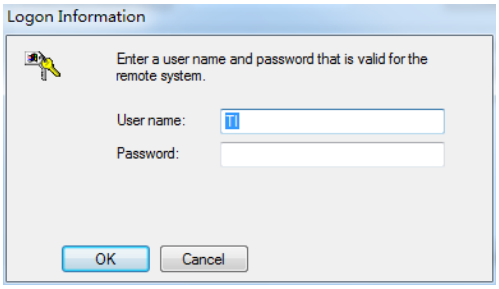
8. Click on " Broadcast Search "



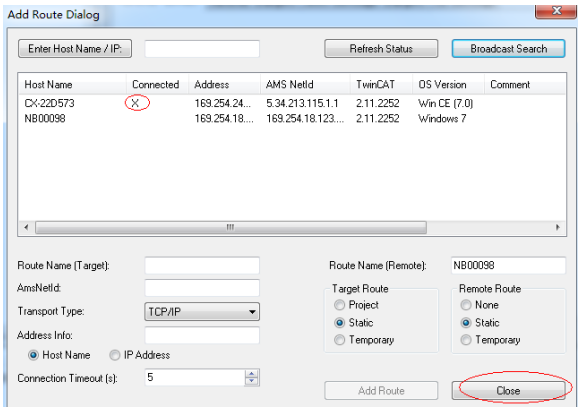
9. Select as shown below and click on “ Add Route ”



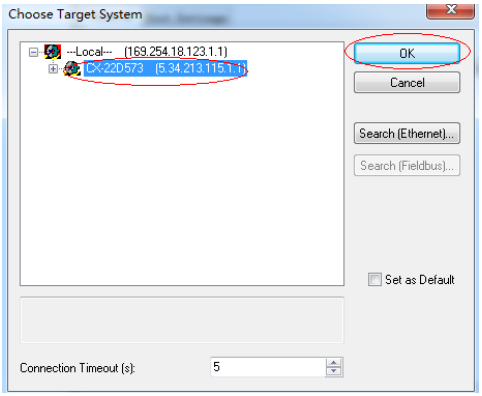
10.Click " OK "



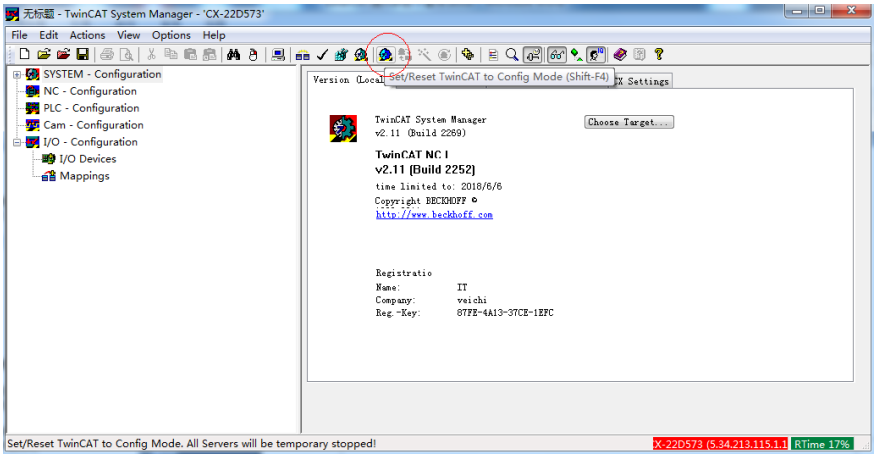
11. The " X " mark appears to indicate that it is connected, click " Close "



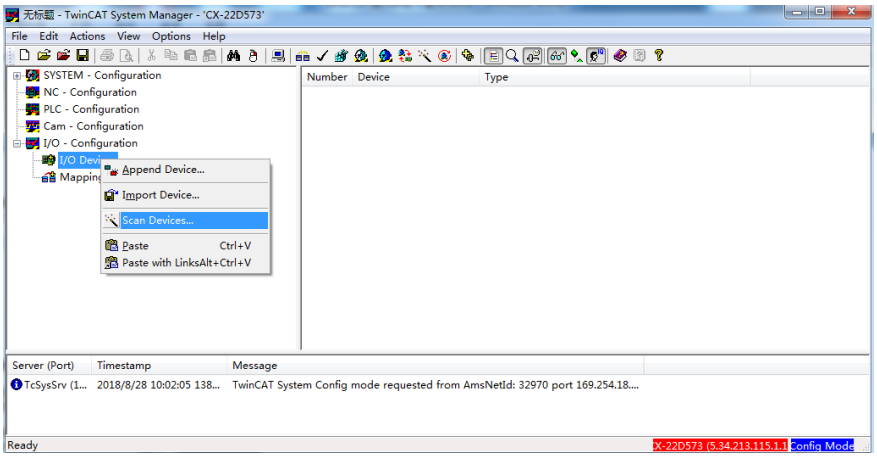
12. Select as shown below



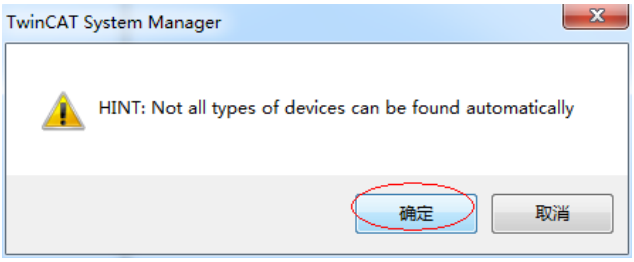
13. Put TwinCAT in configuration mode, click on the image below



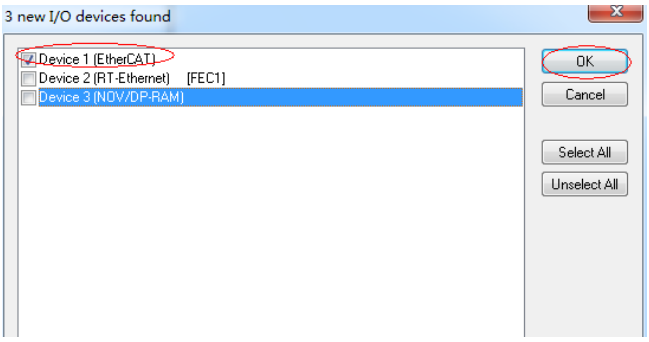
14. Right click on " I/O Device " to search for the servo device



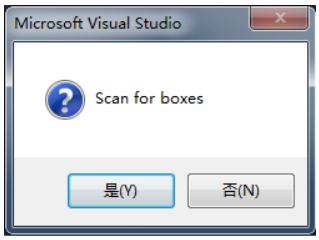
15. Click "OK"



16. Select " EtherCAT "

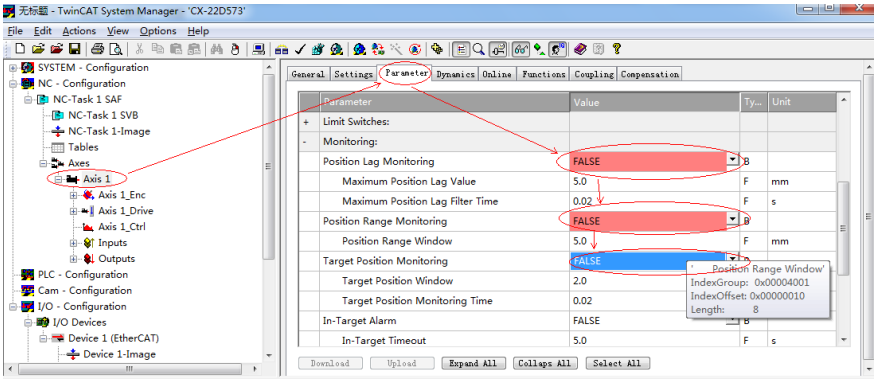


17. Select "Yes" to enable EtherCAT device detection.



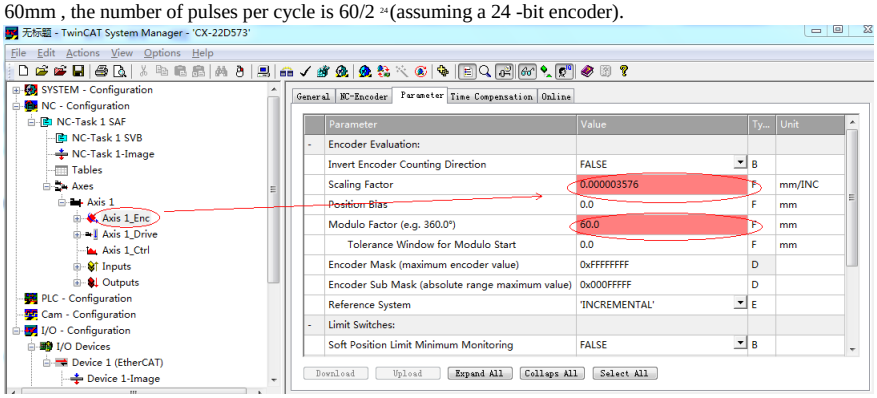
18. When the following window pops up, click No

19. Turn off the monitoring error function

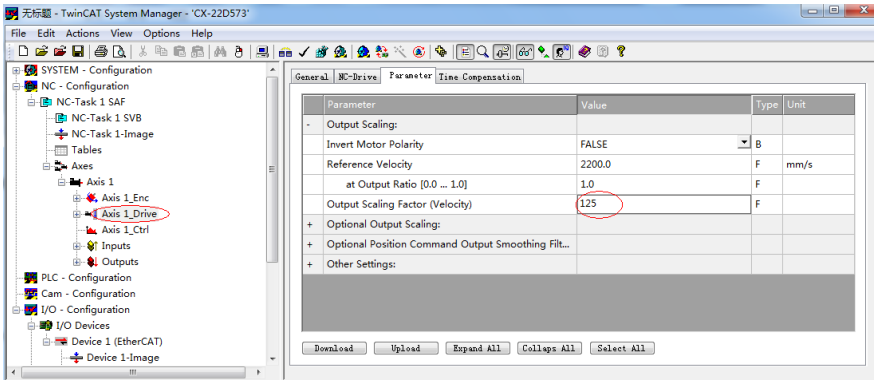


20. Set the pulse coefficient

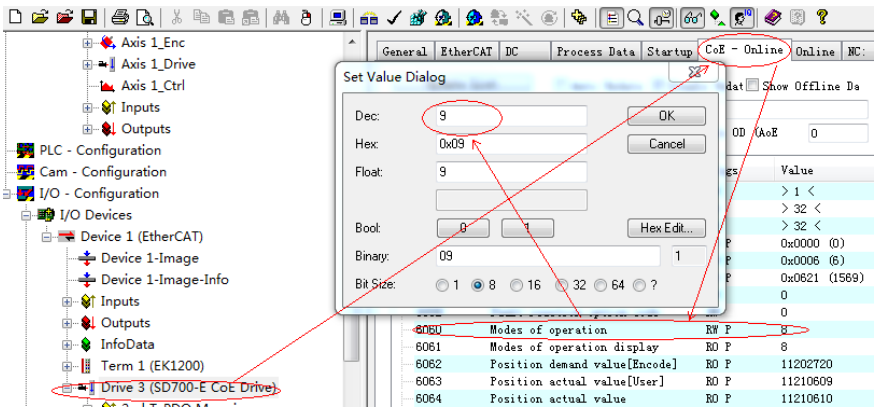
If the setting is 60mm , the number of pulses per cycle is 60/2 ^{24}(assuming a 24 -bit encoder).



21. Set the speed output coefficient (used in speed mode)

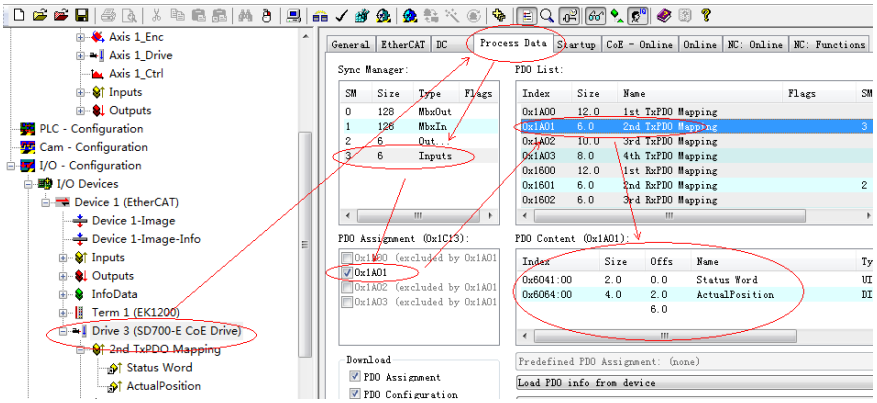


22. Set the relevant parameters through SDO (for example, change the control mode to synchronous cycle speed mode)

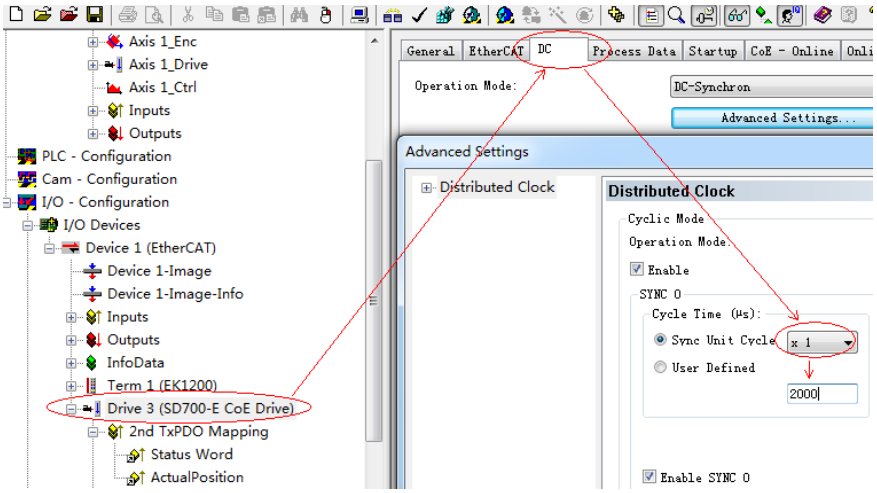


23. Configure PDO

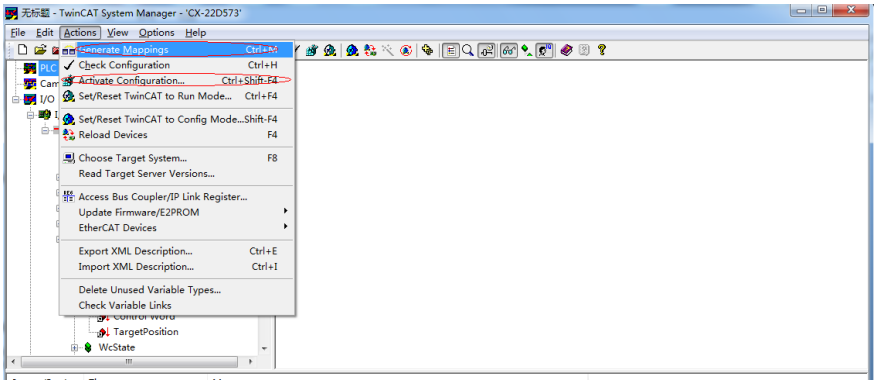
PDO mapping parameter defaults to 1601h and 1A01h , can arbitrarily change the RPDO and TPDO mapping parameters, but can only check 1 Ge RPDO and 1 Ge TPDO . The contents of the mapping object can be added and deleted, but it is recommended that the total number of bytes of the mapping object does not exceed 52 bytes.



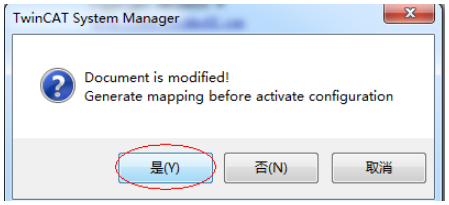
24. Distributed clock cycle setting (set cycle time according to specific conditions, default 2ms)



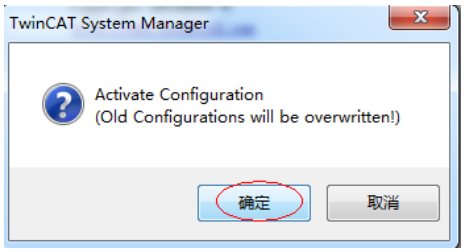
25. Activation configuration



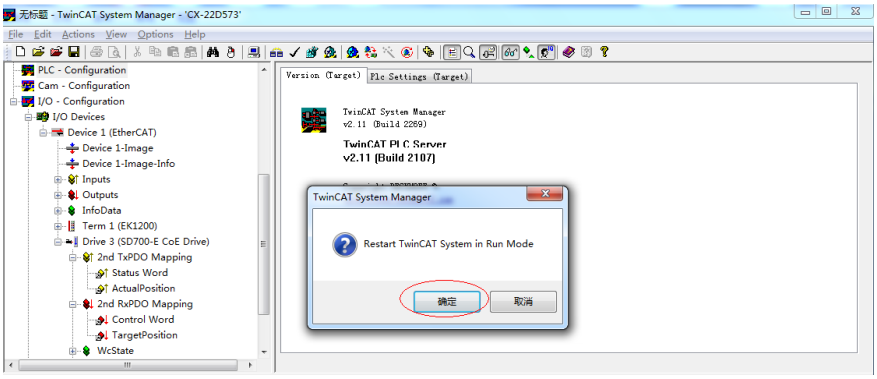
26. Click "Yes"



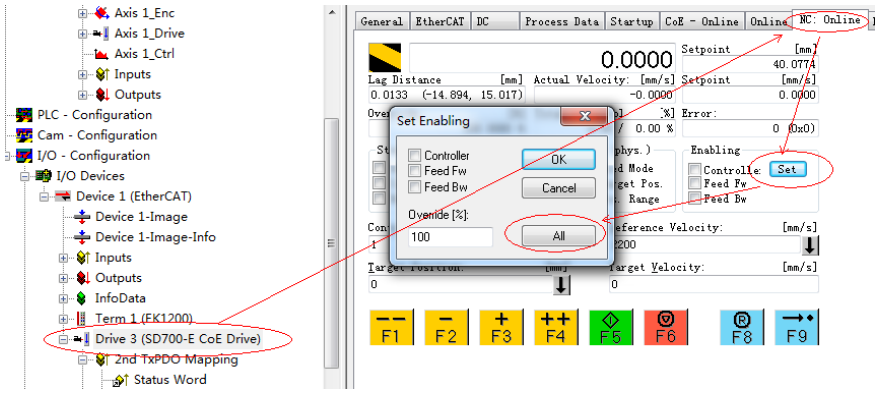
27. Click "OK"



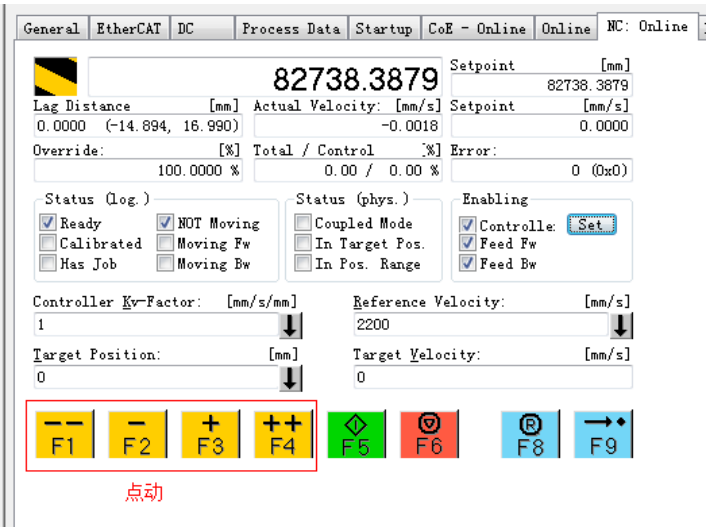
28. When the window below appears, click OK to make TwinCAT running.



29. Enable Servo Drive



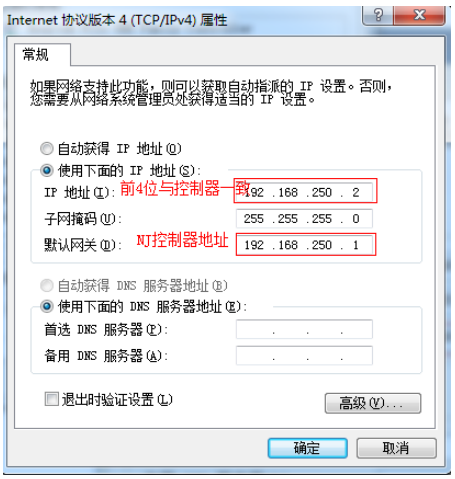
30. Jog operation



9 . With the OMRON main station operation example

9.1 Preparation

1. Install sysmac studio software, it is recommended to install version 1.09 and above.
2. Place the device description file of Flextronics " VEICHI_SD700-E_Rev4_XML_ET1100_20190228.xml " and above into the specified path : OMRON\Sysmac Studio\IO DeviceProfiles\EsiFiles\UserEsiFiles
For the first time, change the file to this path and you need to restart the sysmac studio software .
3. Set the network connection properties of the computer :
To connect the computer to the NJ controller via Etherenet , you need to set the TCP/IP properties of the computer :



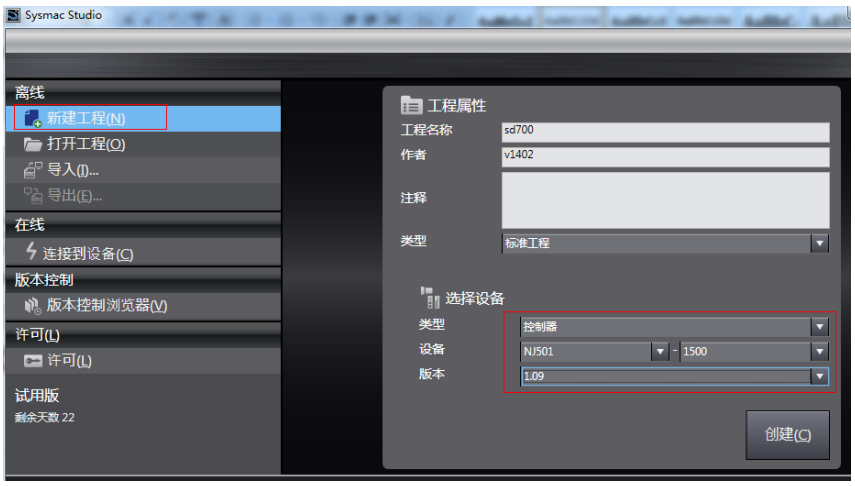
4. Confirm that the servo drive ARM software version is " 3762 " and above.

5. Since the OMRON master station cannot automatically assign the station address to the slave station, the slave station needs to manually set the node address. The SD700 sets the site alias , that is, the communication station number, through **Pn080** . It is recommended to set the station number according to the actual physical connection order in order to manage the configuration.

9.2 Omron background software configuration

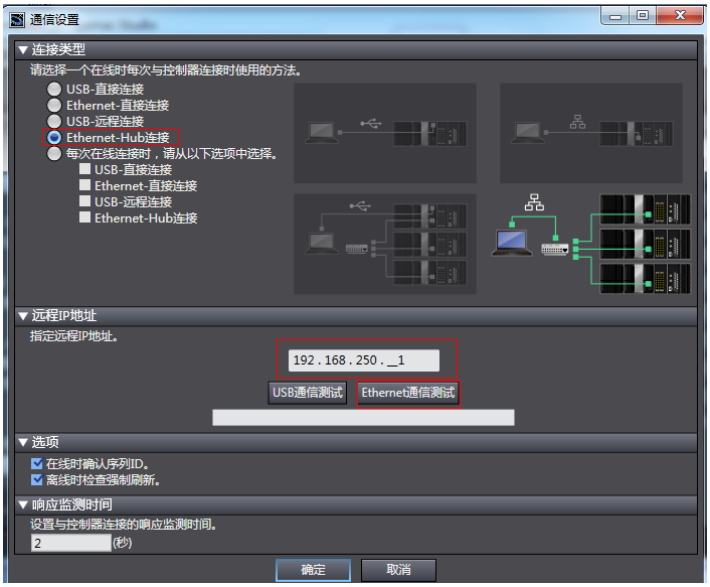
9.2 . 1 New construction

Select the controller type and software version, it is recommended that the software select version 1.09 or higher.



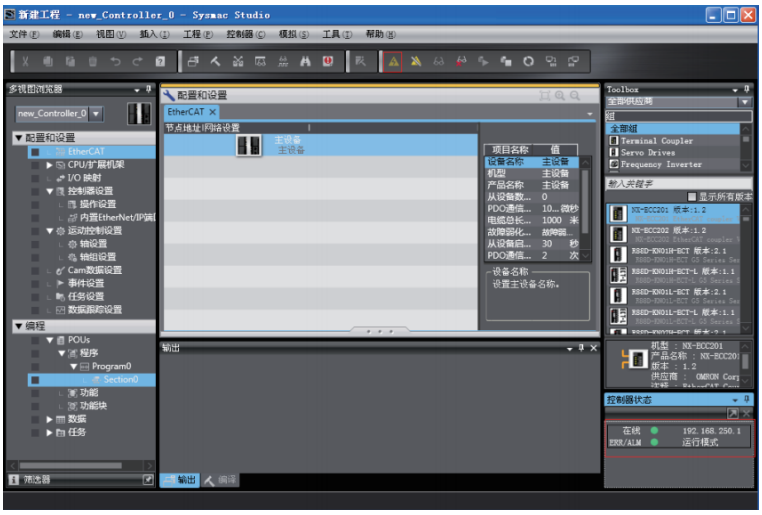
9.2.2 Communication Settings

After entering the main interface, click the menu bar "Controller" - "Communication Settings", select the Ethernet-Hub connection, type the controller network address, and click " Ethernet Communication Test". If there are no problems, the test will be successful.



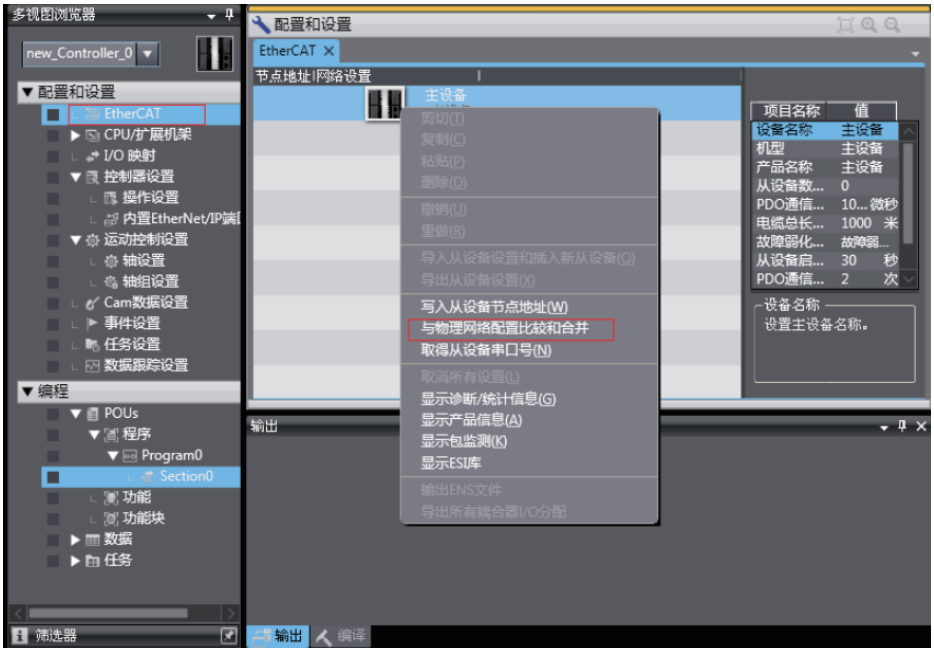
9.2.3 Scanning equipment

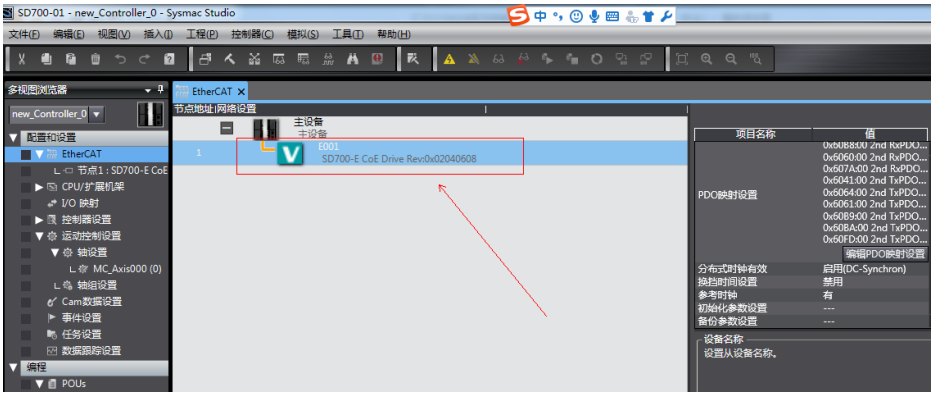
Switch the controller to "online" mode.



Scan the slave and add

In the left menu bar "Configuration and Settings", click " EtherCAT ", right-click "Master Device", and select "Compare with physical network configuration." The controller then automatically scans all slaves in the network. After scanning, click “Apply Physical Network Configuration”, then the slave is added. On the main page you can see the added SD700 slaves. As shown below.

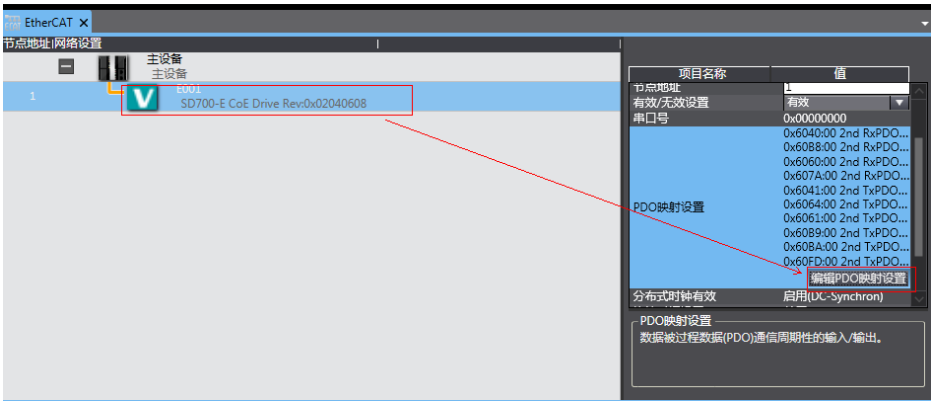


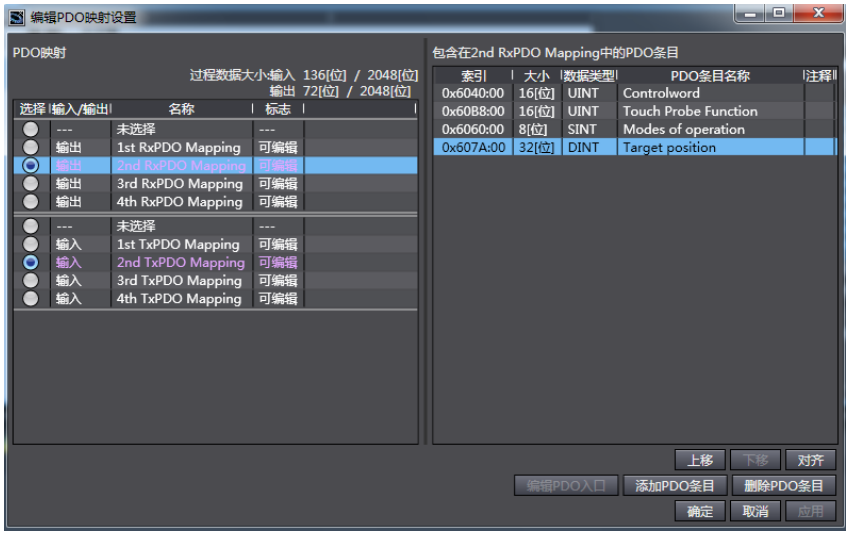


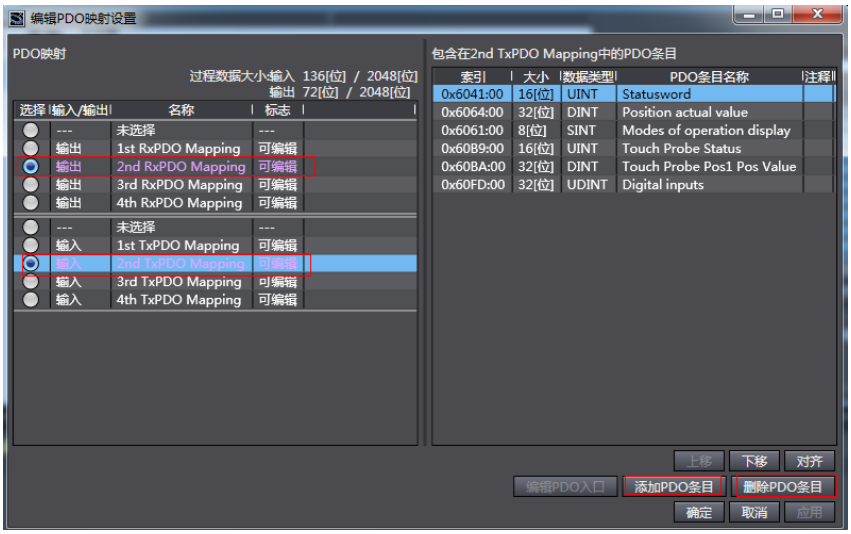
9.2.4 Parameter Configuration

Switch the controller to the offline mode and configure the PDO , DC clock, axis parameters, and so on.

Click on the scanned SD700 slave and click on "Compile PDO Mapping Configuration" on the right . After opening the can see the default choice TPDO and RPDO can change the input and output, change after, and can be for PDO content mapping that is added or deleted .



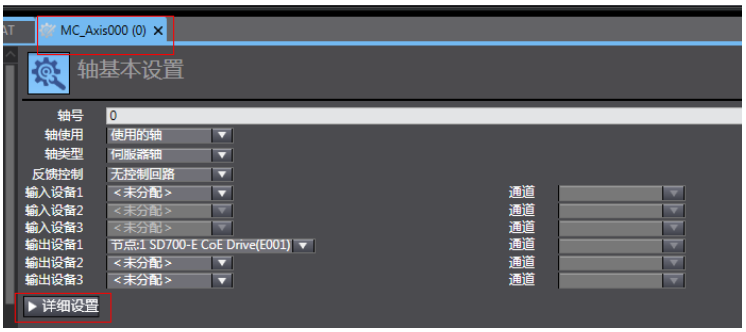




Among them, the object dictionary 0x6060 and the object dictionary 0x60B8 must be mapped, otherwise the primary station will report an error.

Right-click on "Motion Control Settings" - "Axis Settings" to add "Motion Control Axis". Then make the basic settings of the axis.

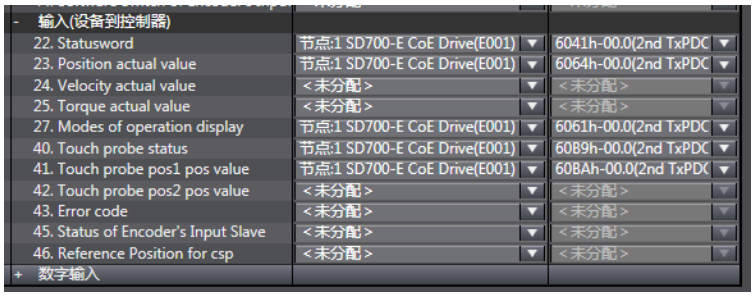
Set the basic settings of the axis as shown below, click "Detailed Settings"



After clicking on the detailed settings, you can expand the "Output", "Input" and "Digital Input". It is recommended to assign the corresponding content as shown in the figure below.

Output (controller to device):

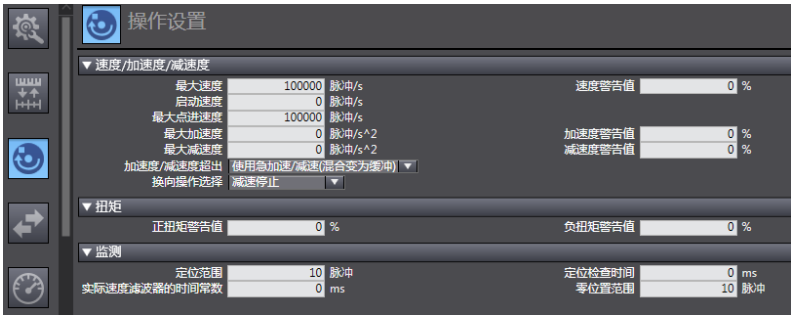
Input (device to controller) :



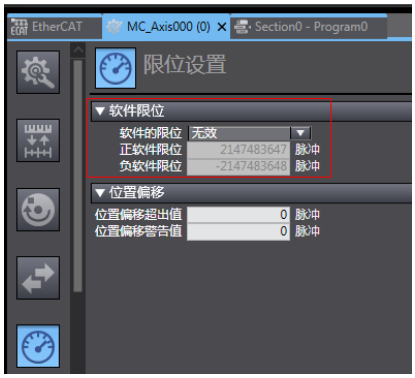
Digital input:

The actual motor resolution setting "motor revolution 1 the number of turns of the command pulse" (such as 23 is registration motor revolution is selected 8388608 pulses). The working stroke of the motor can be kept as default by one week. The electronic gear ratio is no longer set inside the servo.

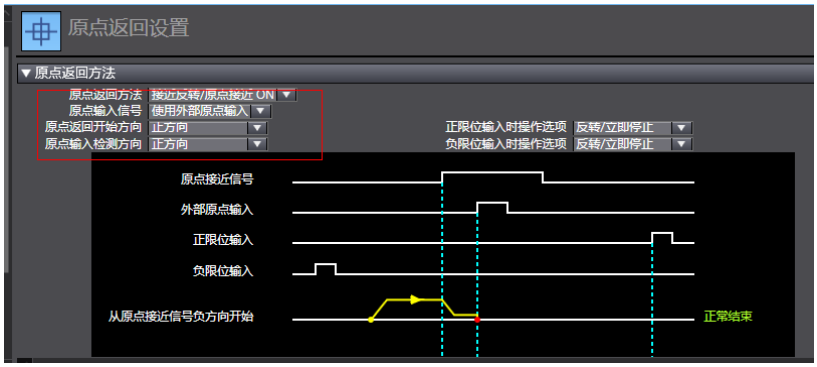
Set the speed, acceleration, deceleration, etc. according to the conversion of the unit. The acceleration or deceleration is 0 , indicating that the running curve is planned with the maximum acceleration or deceleration.



The limit function can be selected according to the actual situation.



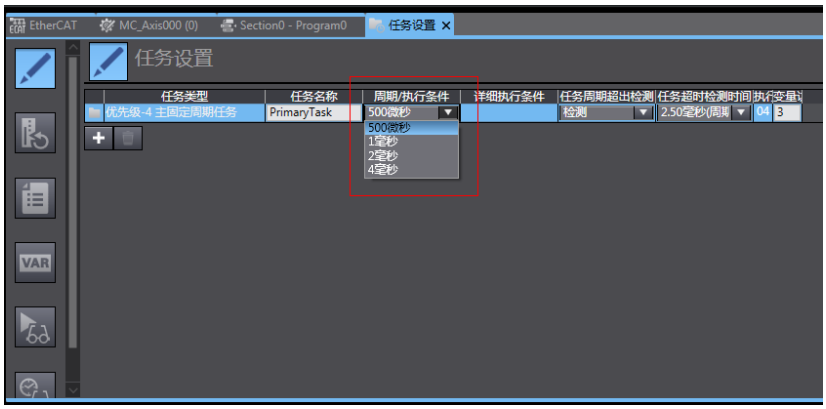
Use the following figure to set the OPR method and method.



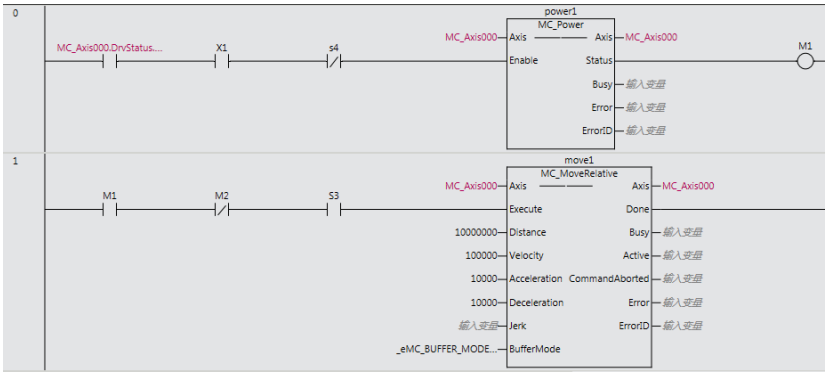
Setting the origin return method requires attention, the cooperation between the servo and the host computer function, refer to the following table.

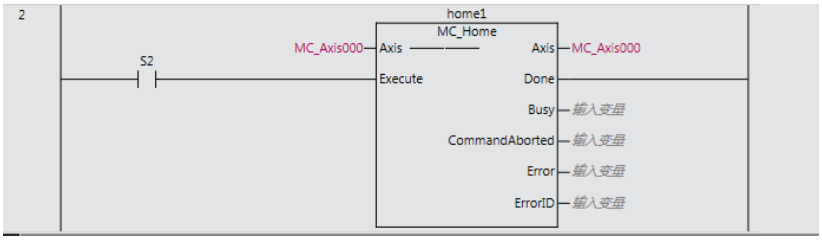
NJ software description	Servo response function	Terminal configuration
Origin approach signal	Origin switch	DI38-DI41 , DI46
External origin input	Probe 1 or probe 2	DI44 or DI45
Z phase signal input	Motor encoder Z- phase signal	-
Positive limit input	Positive limit	DI38-DI41 , DI46
Negative limit input	Negative limit	DI38-DI41 , DI46

The distributed clock cycle can be set by the following figure



After the configuration is completed, the servo operation can be controlled by the PLC program. The following is a programming example, which can be referred to.





After all configuration and programming is completed, switch to the online state and execute the download to the controller to compare the difference between the current program and the program in the controller. After that, you can run the program and click the corresponding switch in the PLC program to drive the servo.