# ZLAC8015DD SERVO DRIVER (SPECIAL FOR HUB SERVO MOTOR)

# CANopen COMMUNICATION ROUTINE

| Version | Description | Date |
|---|---|---|
| V1.0 | - | 2020-03-14 |

# Catalog

# 1.OUTLINE

This manual only gives a brief introduction to the most commonly used related concepts and precautions in the use of ZLAC8015D, so that users can understand the normal use of ZLAC8015D series products in the shortest time.

### Communication Standard followed by ZLAC8015D

- CAN 2.0A Standard
- CANopen Standard protocol DS 301 V4.02
- CANopen Standard protocol DS 402 V2.01

### Services supported by ZLAC8015D

- SDO Service
- PDO Service: each slave station could configure up to 4 TxPDO and 4 RxPDO
- NMT Slave Service
- Device Monitor: supports Heartbeat Message

### The following description takes the driver address as 1, the baud rate as 500K.

The driver address can be set to 0-127. ZLAC8015D has 7 optional baud rates: 25kHz, 50kHz, 100kHz, 125kHz, 250kHz, 500kHz, 1MHz. Baud rate could be set through software, its default value is 500kHz.

# 2. CiA301 DESCRIPTION

## 2.1 COMMUNICATION OBJECT IDENTIFIER(COB-ID)

The communication object identifier (COB-ID) is used to specify the priority of the communication object and to identify the communication object. In CANopen protocol, it is stipulated that 11 arbitration bits are divided into the high 4 bits of the Function Code and the low 7 bits of the node address (Node-ID), which is called COB-ID (Communication Object Identifier).

| CANopen predefined master / slave connection set | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Function Code | | | | Node-ID | | | | | | |

※Node-ID range is 1~127 (0 is not allowed to be used).

| Object | Function Code | Node-ID | COB-ID | Object Dictionary Index |
|---|---|---|---|---|
| NMT(Network Management) | 0000 | 0 | 0x000 | - |
| Sync Message | 0001 | 0 | 0x080 | 0x1005 |
| Emergency Stop | 0001 | 1-127 | 0x081-0x0FF | 0x1014 |
| TPDO1 | 0011 | 1-127 | 0x181-0x1FF | 0x1800 |
| RPDO1 | 0100 | 1-127 | 0x201-0x27F | 0x1400 |
| TPDO2 | 0101 | 1-127 | 0x281-0x2FF | 0x1801 |

| | | | | |
|---|---|---|---|---|
| RPDO2 | 0110 | 1-127 | 0x301-0x47F | 0x1401 |
| TPDO3 | 0111 | 1-127 | 0x381-0x3FF | 0x1802 |
| RPDO3 | 1000 | 1-127 | 0x401-0x47F | 0x1402 |
| TPDO4 | 1001 | 1-127 | 0x481-0x4FF | 0x1803 |
| RPDO4 | 1010 | 1-127 | 0x501-0x57F | 0x1403 |
| RSDO Server Send | 1100 | 1-127 | 0x581-0x5FF | 0x1200 |
| TSDO Client Respond | 1011 | 1-127 | 0x601-0x67F | 0x1200 |
| NMT Error Control | 1110 | 1-127 | 0x701-0x77F | 0x1016-0x1017 |

## 2.2 OBJECT DICTIONARY(OD)

Object Dictionary is the core concept of CANopen. Every CANopen device in the network has an object dictionary. Object Dictionary is an ordered collection of data objects. These objects describe all the communication and device parameters of the device, and their positions are determined in Object Dictionary by a 16-bit index and an 8-bit subindex.

**Object Dictionary supported by ZLAC8015D**

| Index Range | Description |
|---|---|
| 0x1000-0x1A03 | CiA301 Communication Object Sub-Protocol Area |
| 0x2003-0x2030 | Factory Custom Area |
| 0x603F-0x60FF | CiA402 Motion Control Sub-Protocol Area |

※*See Appendix A for details*

## 2.3 SERVICE DATA OBJECT(SDO)

SDO is used to access the Object Dictionary of a device. The visitor is called a Client, and the CANopen device whose object dictionary is accessed and provides the requested service is called a Server. The client's CAN message and the server's response CAN message always contain 8 bytes of data (although not all data bytes are necessarily meaningful). A client's request must have a response from the server.

SDO Message Format

| Byte0 | Byte1:2 | Byte3 | Byte4:7 |
|---|---|---|---|
| SDO Command | Object Index | Object Sub-Index | Data |

Command Word Description

| Command | Function | Type | Data Length |
|---|---|---|---|
| 2F | Setting | M->S Request | 1 byte |
| 2B | Setting | M->S Request | 2 byte |
| 27 | Setting | M->S Request | 3 byte |
| 23 | Setting | M->S Request | 4 byte |
| 60 | Setting feedback | S->M Confirm | |
| 40 | Read | M->S Request | 0 byte |
| 4F | Read feedback | S->M Answer | 1 byte |

| | | | |
|---|---|---|---|
| 4B | Read feedback | S->M Answer | 2 byte |
| 47 | Read feedback | S->M Answer | 3 byte |
| 43 | Read feedback | S->M Answer | 4 byte |
| 80 | Error feedback | S->M Answer | 4 byte |

**※Eg: write data 0x2064 into the object dictionary with index 0x60FF and sub-index 3 through SDO message.**

| COB-ID | Byte7 | Byte6 | Byte5 | Byte4 | Byte3 | Byte2 | Byte1 | Byte0 |
|---|---|---|---|---|---|---|---|---|
| **Client→Server** | | | | | | | | |
| 0x601 | 2F | FF | 60 | 03 | 64 | 20 | 00 | 00 |
| **Server→Client** | | | | | | | | |
| 0x581 | 60 | FF | 60 | 03 | 00 | 00 | 00 | 00 |

## 2.4. NETWORK MANAGEMENT(NMT)

NMT provides network management service. This service is implemented using a master-slave communication mode (so there is only one NMT master node). Only the NMT master node can transmit NMT module control messages, all slave nodes must support the NMT module control service, and the NMT module control does not need to response. The message format is as follows:

| NMT master node → NMT slave node | | |
|---|---|---|
| **COB-ID** | **Byte 1** | **Byte 0** |
| 0x000 | Command | Node-ID |

When Node-ID=0, all NMT slave nodes are addressed. The corresponding relationship between the value of the command word and the service is as follows:

| Command | NMT Service |
|---|---|
| 0x01 | Start command (Put the node into operating state) |
| 0x02 | Stop command (Put the node into stop state) |
| 0x80 | Pre-operation command (Put the node into pre-operation state) |
| 0x81 | Reset node application layer (Put the node into application layer reset state) |
| 0x82 | Reset node communication (Put the node into communication reset state) |

The node status is described in the following table:

| Status | Description |
|---|---|
| Initialization | Initialize CAN controller after the node is powered on. |
| Application layer reset | Node application reset |
| Communication reset | Node CANopen communication reset |
| Pre-operational state | Node CANopen communication is in operation;<br>Could communicate with SDO and NMT. |
| Operating state | After the node receives the start command sent by NMT master station, the node's PDO communication is activated. |
| Stop state | After the node receives the stop command from NMT master station, the node's PDO communication is prohibited. |

## 2.5. HEARTBEAT(NMT ERROR CONTROL)

NMT error control is mainly used to detect whether the device in the network is online and the state of the device, including node/life protection and heartbeat. Currently, ZLAC8015D only supports the heartbeat method.

A node can be configured to generate periodic messages which are called Heartbeat messages. The heartbeat mode uses a producer (slave)-consumer (master) model. CANopen device can send heartbeat messages according to the period set by the producer's heartbeat interval object 0x1017, the unit is ms. A node in the network that always has a consumer heartbeat function monitors the producer according to the consumer time set by object 0x1016. Once the producer's heartbeat of the corresponding node is not received within the consumer heartbeat time range, the node is considered to be offline ( Or there is a fault).

The CANopen slave station sends the heartbeat message according to the heartbeat production time (ms) filled in 1017h in its object dictionary, and CANopen master station (NMT master station) will check according to the heartbeat consumption time filled in its 1016h. Assuming that more than a few heartbeat consumption times, CANopen master (NMT master) has not received the heartbeat message from the slave station, it is considered that the slave station is offline or damaged.

The slave sends heartbeat messages every 0x1017 time to monitor the master of the slave (or other slaves) . If the heartbeat message is not received within the consumer time, the slave is considered to be offline. 0x1017×2 should be smaller than the consumer time of the master of the slave (or other slaves), otherwise it is easy to report that the slave is offline incorrectly.

The heartbeat message format is as follows:

| Heartbeat producer→Consumer | |
|---|---|
| **COB-ID** | **Byte 0** |
| 0x700+Node-ID | State |

The status description is as follows:

| State | Description |
|---|---|
| 0x00 | Boot-up |
| 0x04 | Stop state |
| 0x05 | Operating state |
| 0x7F | Pre-operation state |

*※ZLAC8015D is a heartbeat producer.*

## 2.6. PROCESS DATA OBJECT(PDO)

PDO belongs to process data, which is used to transmit real-time data, that is, one-way transmission. It does not require the receiving node to respond with CAN messages to confirm. It belongs to the "production and consumption" model in terms of communication. The length of PDO can be less than 8 bytes, and the transmission speed is relatively fast. Each PDO information includes sending PDO (TxPDO) and receiving PDO (RxPDO) information, and its transmission mode is defined in the PDO communication parameter index. All PDO transmission data must be mapped to the corresponding index area through the

object dictionary.

ZLAC8015D supports 4 groups of sending and receiving PDO, the related objects are as follows:

| Object | COB-ID | Communication Object | Mapping Parameter |
|---|---|---|---|
| TPDO0 | 0x181-0x1FF<br>（0x180+ Node-ID） | 0x1800 | 0x1A00 |
| RPDO0 | 0x201-0x27F<br>（0x200+ Node-ID） | 0x1400 | 0x1600 |
| TPDO1 | 0x281-0x2FF<br>（0x280+ Node-ID） | 0x1801 | 0x1A01 |
| RPDO1 | 0x301-0x37F<br>（0x300+ Node-ID） | 0x1401 | 0x1601 |
| TPDO2 | 0x381-0x3FF<br>（0x380+ Node-ID） | 0x1802 | 0x1A02 |
| RPDO2 | 0x401-0x47F<br>（0x400+ Node-ID） | 0x1402 | 0x1602 |
| TPDO3 | 0x481-0x4FF<br>（0x480+ Node-ID） | 0x1803 | 0x1A03 |
| RPDO3 | 0x501-0x57F<br>（0x500+ Node-ID） | 0x1403 | 0x1603 |

### 2.6.1 PDO COMMUNICATION PARAMETERS

**1. PDO CAN identifier**

The PDO CAN identifier is the COB-ID of the PDO, which contains control bits and identification data, to determine the bus priority of PDO. The COB-ID is located in the sub-index 01 of the communication parameters (RPDO: 1400h~1403h, TPDO: 1800h~1803h), and its highest bit determines whether the PDO is valid.

**2. PDO transmission type**

The PDO transmission type is located on the sub-index 02 of the communication parameters (RPDO: 1400h~1403h, TPDO: 1800h~1803h).

**Asynchronous Transmission**: the transmission is triggered by events, including data change, periodic event timer.

**Synchronous Transmission**: it's related to synchronization frames in the network.

For sub-index 02 of communication parameters (RPDO: 1400h~1403h, TPDO: 1800h~1803h), different values represent different transmission types, it defines the method of triggering TPDO transmission or processing the received RPDO. The specific correspondence is as follows:

| Transmission Type | Synchronous Transmission | | Asynchronous Transmission |
|---|---|---|---|
| | Cyclic | Acyclic | |
| 0 | | √ | |
| 1-240 | √ | | |
| 241-254 | √ | | |
| 254, 255 | | | √ |

Note:

1) When TPDO transmission type is 0, if the data of the mapping object changes, and a synchronization frame is received,

the TPDO is sent.

2) When TPDO transmission type is 1~240, and the synchronization frames with corresponding number is received, the TPDO is sent .

3) When TPDO transmission type is 254 or 255, and the mapping data changes, or the event timer arrives, the TPDO is sent.

4) When RPDO transmission type is 0~240, as long as a synchronization frame is received, the latest data of RPDO will be updated to the application. When RPDO transmission type is 254 or 255, the received data will be directly updated to the application.

*※Note: ZLAC8015D only supports 254/255 transmission method at present.*

**3. Prohibition Time**

The prohibition time is set for TPDO, and it's stored in the sub-index 03 of the communication parameter (1800h~1803h) to prevent the CAN network from being continuously occupied by the higher priority PDO. The unit of this parameter is 100us. After setting the value, the transmission interval of the same TPDO shall not be less than the time corresponding to this parameter.

Eg: if TPDO2 prohibition time is 300, the transmission interval of TPDO will not be less than 30ms.

*Recommendation: when objects that change frequently (such as position feedback, speed feedback, etc.) are configured to TPDO, and TPDO transmission type is asynchronous, it is recommended to set a certain prohibition time.*

**4. Event timer**

For asynchronous transmission TPDO (transmission type is 254 or 255), we define the Event Timer, which is located on the sub-index 05 of the communication parameter (1800h~1803h), and the unit is 500us. Event Timer can also be regarded as a trigger event, which will also trigger the corresponding TPDO transmission. If other events such as data changes occur during the timer running period, TPDO will also trigger and the event counter will be reset immediately.

### 2.6.2 PDO Mapping Parameters

PDO mapping parameter contains a pointer to the process data corresponding to the PDO that needs to be sent or received to point to PDO, including index, sub-index and the length of the mapping object. The data length of each PDO can be up to 8 bytes, and one or more objects can be mapped at the same time. Among them, sub-index 0 records the number of objects specifically mapped by the PDO, and sub-index 1~8 are the mapping content. The mapping parameter content is defined as follows:

**Eg: map 0x6040 to 0x1A00 01:**

| | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
|---|---|---|---|---|---|---|---|---|
| | **2B** | **00** | **1A** | **01** | **10** | **00** | **40** | **60** |
| Define | 2B | Mapping Area | | Sub-index No. | Mapping object length | | Mapped object | |

**Eg: configure 0x606C 03 as TPDO0, and use event trigger (254) or timer trigger (255) as the transmission mode.**

| Master Station（COB-ID:0x601） | Slave Station (COB-ID:0x581） | Function Description |
|---|---|---|
| 2F 00 1A 00 00 00 00 00 | 60 00 1A 00 00 00 00 00 | Clear TPD01 mapping |
| 23 00 1A 01 20 00 6C 60 | 60 00 1A 01 00 00 00 00 | Map 0x606C 03 to 0x1A00 01 |
| 23 00 18 01 81 01 00 00 | 60 00 18 01 00 00 00 00 | Set TPDO0 COB-ID is 181 |
| 2F 00 18 02 FE 00 00 00 | 60 00 18 02 00 00 00 00 | Set TPDO0 transmission mode to event trigger |

| | | |
|---|---|---|
| 2F 00 18 03 88 13 00 00 | 60 00 18 03 00 00 00 00 | Set prohibition time is 500ms |
| 2F 00 1A 00 01 00 00 00 | 60 00 1A 00 00 00 00 00 | Start TPDO0 mapping |
| 2F 10 20 00 02 00 00 00 | 60 10 20 00 00 00 00 00 | Save parameters to EEPROM |

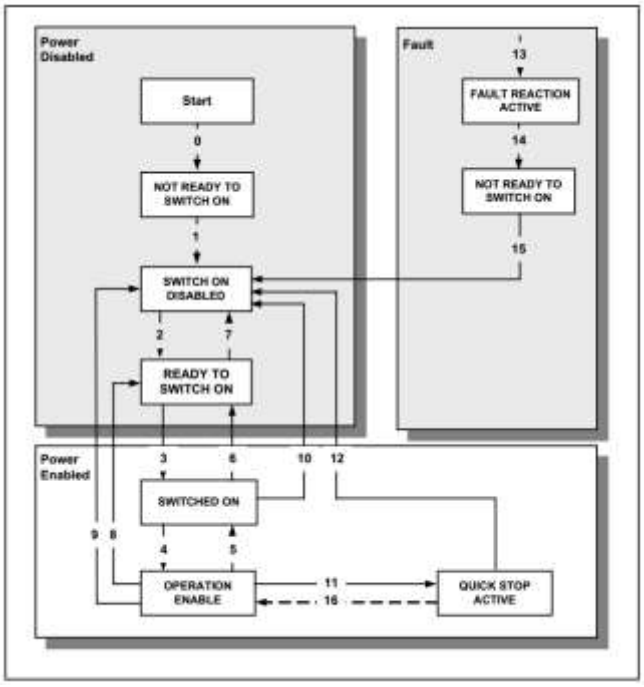
| Master Station（COB-ID:0x601） | Slave Station (COB-ID:0x581） | Function Description |
|---|---|---|
| 2F 00 1A 00 00 00 00 00 | 60 00 1A 00 00 00 00 00 | Clear TPDO1mapping |
| 23 00 1A 01 20 00 6C 60 | 60 00 1A 01 00 00 00 00 | Map 0x606C 03 to 0x1A00 01 |
| 23 00 18 01 81 01 00 00 | 60 00 18 01 00 00 00 00 | Set TPDO0 COB-ID is 181 |
| 2F 00 18 02 FF 00 00 00 | 60 00 18 02 00 00 00 00 | Set TPDO0 transmission mode to timer trigger |
| 2F 00 18 05 E8 03 00 00 | 60 00 18 05 00 00 00 00 | Set the timer time to 500ms |
| 2F 00 1A 00 01 00 00 00 | 60 00 1A 00 00 00 00 00 | Start TPDO0 mapping |
| 2F 10 20 00 02 00 00 00 | 60 10 20 00 00 00 00 00 | Save parameters to EEPROM |

# 3. CiA402 DESCRIPTION

## 3.1. CiA402 STATE MACHINE

CiA402 protocol defines the standard state machine of motion control device, as well as various operating modes and their definitions in object dictionary.

State machine describes the state of the device and the possible control sequences of driver. The status of each step represents a specific internal or external behavior, and the status of the device also determines which commands could be received.




Driver state machine

※ The corresponding state description of the state machine is as follows:

| State Name | Description |
|---|---|

| | |
|---|---|
| NOT READY TO SWITCH ON | It only supplies power to the driver chip, the driver is initializing and self-checking, the driver function is not enabled, and this state is an internal state. |
| SWITCH ON DISABLED | After the driver is initialized, the driver parameters are established and could be modified. This state does not supply power to the motor. This state is the lowest state that user can operate. This state is also the state that use will contact after the driver is powered on. |
| READY TO SWITCH ON | The driver parameters could be modified, the driver function is not enabled, waiting to enter SWITCH ON state. |
| SWITCH ON | Provide high voltage to the driver, the power amplifier is ready, the driver parameters could be modified, and the driver function is not enabled. |
| OPERATION ENABLE | If no fault is detected, the driver function is enabled, and the motor is powered on. The driver parameters could be modified. According to BP [N] parameter, it is determined whether the brake will be automatically released in this state. |
| QUICK STOP ACTIVE | The driver parameters could be modified, the emergency stop function is enabled, the driver function is enabled, and the motor is powered on. |
| FAULT REACTION ACTIVE | The driver parameters could be modified, the driver has a fault, the fault response function is enabled, and the drive function is disabled. This state cannot be entered manually. The driver will enter this state automatically when a fault occurs. |

Driver state machine is controlled by bit0-bit3, bit7 of the control word (object 6040h). The specific description is as follows:

※ Control word switching state:

| **Command** | **Control Word** | | | | | **State Switch** |
|---|---|---|---|---|---|---|
| | Bit7 | Bit3 | Bit2 | Bit1 | Bit0 | |
| Shutdown | 0 | X | 1 | 1 | 0 | 2,6,8 |
| Switchon | 0 | 0 | 1 | 1 | 1 | 3 |
| Switchon +Enable operation | 0 | 1 | 1 | 1 | 1 | 3+4 |
| Disable voltage | 0 | X | X | 0 | X | 7,9,10,12 |
| Quick stop | 0 | X | 0 | 1 | X | 7,10,11 |
| Disable Operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable Operation | 0 | 1 | 1 | 1 | 1 | 4,16 |
| Fault reset | ↑ | X | X | X | X | 15 |
| Bit marked as X is invalid. | | | | | | |

Each state in state machine could be displayed through bit0-bit3, bit5, bit6 of status word (object 6041h), the specific description is as follows:

※ Status word switching state

| **State** | **State word** | | | | | |
|---|---|---|---|---|---|---|
| | Bit6 | Bit5 | Bit3 | Bit2 | Bit1 | Bit0 |
| NOT READY TO SWITCH ON | 0 | X | 0 | 0 | 0 | 0 |
| SWITCH ON DISABLED | 1 | X | 0 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| READY TO SWITCH ON | 0 | 1 | 0 | 0 | 0 | 1 |
| SWITCH ON | 0 | 1 | 0 | 0 | 1 | 1 |
| OPERATION ENABLE | 0 | 1 | 0 | 1 | 1 | 1 |
| QUICK STOP ACTIVE | 0 | 0 | 0 | 1 | 1 | 1 |
| FAULT REACTION ACTIVE | 0 | X | 1 | 1 | 1 | 1 |
| NOT READY TO SWITCH ON | 0 | X | 1 | 0 | 0 | 0 |
| Bit marked as X is invalid. | | | | | | |

## 3.2. CONTROL WORD AND STATUS WORD

The start and stop control commands and state description of the driver are mainly realized through the control word 6040h and the status word 6041h. Therefore, the skilled use of the control word and status word is very necessary. The following table briefly describes the definition of the control word and status word.

| Control Word | Common Command | Function Description |
|---|---|---|
| 6040h | 00 | Initialization step 0: At this time, the low 4-bit status of 6041 is 0000, motor is released; |
| | 06 | Initialization step 1: At this time, the low 4-bit status of 6041 is 0001, motor is released; |
| | 07 | Initialization step 2: At this time, the low 4-bit status of 6041 is 0011, motor is enabled; |
| | 0F | Initialization step 3: At this time, the low 4-bit status of 6041 is 0111, motor is enabled; |
| | 0F | Start command in Profile Velocity Mode (6061 = 3); |
| | 0F->1F | Start command in Profile Torque Mode (6061 = 4);<br>Absolute motion start command in position mode (6061 = 1); |
| | 4F->5F | Relative motion start command in position mode (6061 = 1); |

| Status Word | Bit Definition | Function Description |
|---|---|---|
| 6041h | Bit0~Bit3 | 6040=0： xxxx xxxx xxxx 0000<br>6040=6： xxxx xxxx xxxx 0001<br>6040=7： xxxx xxxx xxxx 0011<br>6040=F： xxxx xxxx xxxx 0111 |
| | Bit7 | 0: driver is normal;<br>1: driver alarms; |
| | Bit8 | 0: torque is not completed;<br>1: torque has been completed; |
| | Bit11 | 0: the status of Bit4 at 6040h is 0;<br>1: the status of Bit4 at 6040h is 1; |
| | Bit13 | 0: motor release;<br>1: motor is enabled; |
| | Bit14 | 0: motor is stopped;<br>1: motor is running; |
| | Bit15 | 0: The motion is not in position in position mode;<br>1: The motion is in position in position mode. |

Eg: Initialize the driver after power-on. After initialization, it enters the normal operation state. This

operation is generally performed after power-on.

| Master station | Slave station | Status word of slave station |
|---|---|---|
| 00: 01 00 | NMT initialization | NMT initialization |
| 601:<br>2B 40 60 00 00 00 00 00 | 581:<br>60 40 60 00 00 00 00 00 | 6041:<br>xxxx xxxx xxxx 0000 |
| 601:<br>2B 40 60 00 06 00 00 00 | 581:<br>60 40 60 00 00 00 00 00 | 6041:<br>xxxx xxxx xxxx 0001 |
| 601:<br>2B 40 60 00 07 00 00 00 | 581:<br>60 40 60 00 00 00 00 00 | 6041:<br>xxxx xxxx xxxx 0011 |
| 601:<br>2B 40 60 00 0F 00 00 00 | 581:<br>60 40 60 00 00 00 00 00 | 6041:<br>xxxx xxxx xxxx 0111 |

CANopen sets the operation mode of the driver through the object 6060h (Mode of Operation) and reflects the current operation mode status of the driver through the object 6061h (Mode of operation display). ZLAC8015D series driver currently supports 3 operation modes: Profile Position Mode, Profile Velocity Mode, and Profile Torque Mode.

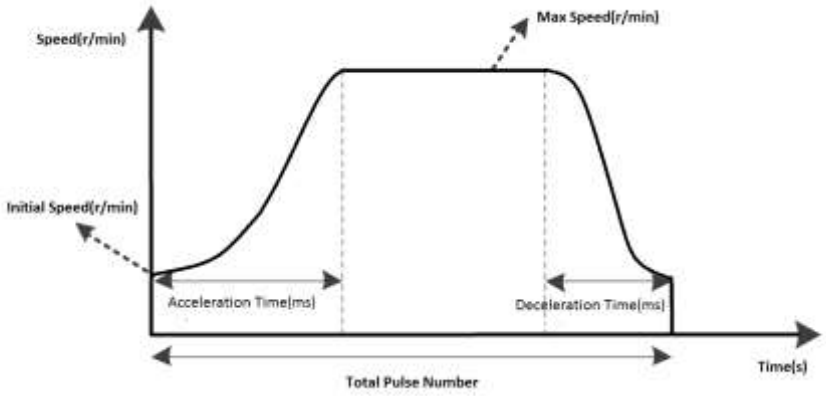
※ Driver Operation Mode:

| Index | Sub-Index | Name | Type | Attribute | PDO Mapping | Parameter Range | Default |
|---|---|---|---|---|---|---|---|
| 6060h | 00 | Modes_of_operation | I8 | RW | NO | 0: undefined<br>1: Profile Position Mode<br>3: Profile Velocity Mode<br>4: Profile Torque Mode | 0 |

## 3.3. PROFILE POSITION MODE

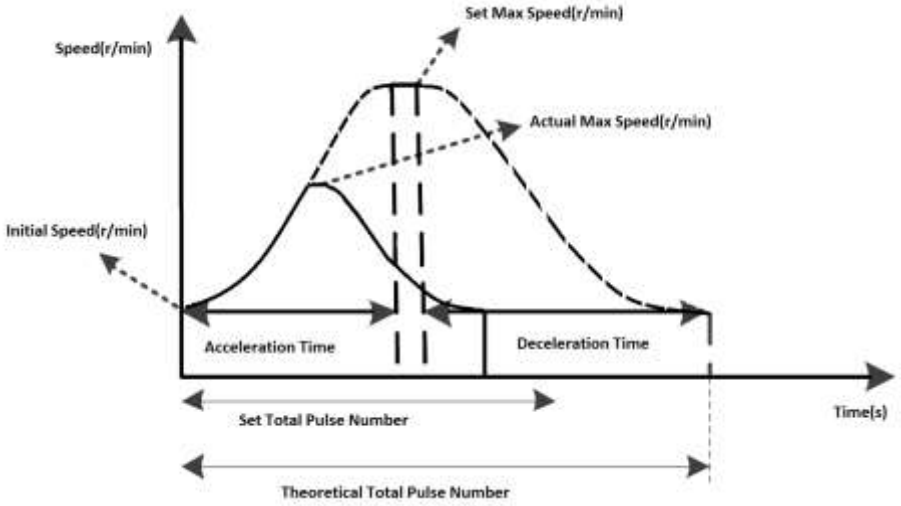
### 3.3.1 Profile Position Mode Description

Profile position mode is realized by an S-shaped acceleration and deceleration curve. User could set several parameters such as initial speed, maximum speed, acceleration time, deceleration time, and total pulse number through the bus to achieve precise position control. The S-shaped acceleration / deceleration curve is shown in the figure below.




Profile position mode acceleration/deceleration curve

When the total pulse number set by user is small, the motor may need to decelerate before it accelerates to the maximum speed (that is, the motor does not accelerate to the maximum speed set by the user during actual operation). The speed curve is shown in the figure below. The solid line in the figure shows the actual running curve of the motor, and the dotted line is the curve that motor needs to run, to accelerate to the set maximum speed. The theoretical total pulse number is the minimum total pulse number calculated according to the user-set parameters (initial speed, maximum speed, acceleration time, deceleration time). When the total pulse number set by the user is less than the theoretical total pulse number, the motor will run as shown by the solid line in the figure.




Profile position mode acceleration/deceleration curve (not accelerated to the set maximum speed)

**Control word and status word**

The control word in Profile position mode is controlled by bit4 -bit6, bit8:

| Byte | Name | Function Description |
|---|---|---|
| Bit4 | New set-point | 0: No assumed target position;<br>1: Assumed target position; |
| Bit5 | Change set immediately | 0: Complete the current position and then start the next position;<br>1: Interrupt the current position and start the next position; |

| | | |
|---|---|---|
| Bit6 | ABS/REL | 0: The target position is an absolute value;<br>1: The target position is a relative value; |
| Bit8 | Halt | 0: Terminate current position;<br>1: Set deceleration to stop; |
| Note: According to the above table , the control word of absolute position motion command is 0x0F-> 0x1F, and the control word of relative position motion command is 0x4F-> 0x5F. | | |

※ Bit10, Bit12, Bit15 of status word display the driver status

| Byte | Name | Function Description |
|---|---|---|
| Bit10 | Target reached | 0: Halt = 0 target position has not been reached;<br>Halt = 1 motor is decelerating;<br>1: Halt = 0 reach the target position;<br>Halt = 1 motor speed is 0; |
| Bit12 | Set-point acknowledge | 0: The target position is pending;<br>1: The target position has taken effect; |
| Bit15 | Pend | 0: Not in position;<br>1: In position. |

※ Related Object Dictionary Content

| Index | Sub-Index | Name | Description | Type | Attribute | PDO mapping | Defaults |
|---|---|---|---|---|---|---|---|
| 6040h | 00 | Controlword | Controlword | U16 | RW | YES | 0 |
| 6060h | 00 | Modes_of_operation | 0: Undefined;<br>1: Profile position mode;<br>3: Profile velocity mode;<br>4: Profile torque mode; | I8 | RW | YES | 0 |
| 607Ah | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Target_position (left) | Range of total pulse number in pofile position mode operation:<br>relatively:<br>-0x7FFFFFFF~0x7FFFFFFF<br>absolute:<br>-0x3FFFFFFF~0x3FFFFFFF | I32 | RW | YES | 0 |
| | 02 | Target_position (right) | Range of total pulse number in profile position mode operation:<br>relatively:<br>-0x7FFFFFFF~0x7FFFFFFF<br>absolute:<br>-0x3FFFFFFF~0x3FFFFFFF | I32 | RW | YES | 0 |
| 6081h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_velocity (left) | Maximum speed in profile position mode; | U32 | RW | YES | 120r/min |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Range: 1-1000r/min; | | | | |
| | 02 | Profile_velocity (right) | Maximum speed in position mode;<br>Range: 1-1000r/min; | U32 | RW | YES | 120r/min |
| 6083h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_acceleration (left) | Acceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| | 02 | Profile_acceleration (right) | Acceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| 6084h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_deceleration (left) | Deceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| | 02 | Profile_deceleration (right) | Deceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |

### 3.3.2 Profile Position Mode Configuration Routine

Make the motor run relatively based on the parameters (acceleration time 100ms, deceleration time 100ms, maximum speed 60r / min, total pulse number 3200).

**Profile position mode initialization:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2F 60 60 00 01 00 00 00 | 60 60 60 00 00 00 00 00 | Set profile position mode |
| 23 83 60 01 64 00 00 00 | 60 83 60 01 00 00 00 00 | Set the left motor<br>Acceleration time 100ms |
| 23 83 60 02 64 00 00 00 | 60 83 60 01 00 00 00 00 | Set the right motor<br>Acceleration time 100ms |
| 23 84 60 01 64 00 00 00 | 60 84 60 01 00 00 00 00 | Set the left motor<br>Deceleration time 100ms |
| 23 84 60 02 64 00 00 00 | 60 84 60 01 00 00 00 00 | Set the right motor<br>Deceleration time 100ms |
| 23 81 60 01 3C 00 00 00 | 60 81 60 01 00 00 00 00 | Set the left motor<br>Maximum speed 60r/min |
| 23 81 60 02 3C 00 00 00 | 60 81 60 01 00 00 00 00 | Set the right motor<br>Maximum speed 60r/min |
| 2B 40 60 00 06 00 00 00 | 60 40 60 00 00 00 00 00 | Enable motor |
| 2B 40 60 00 07 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | |

**Relative position control of the left motor:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 01 00 7D 00 00 | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position 32000 |

| | | |
|---|---|---|
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 01 00 83 FF FF | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position-32000 |
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

### Relative position control of the right motor:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 02 00 7D 00 00 | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position 32000 |
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 02 00 83 FF FF | 60 7A 60 01 00 00 00 00 | Set the right motor<br>Target position-32000 |
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

### Absolute position control of the left motor:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 01 00 7D 00 00 | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position 32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 01 00 83 FF FF | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position-32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

### Absolute position control of the right motor:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 02 00 7D 00 00 | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position 32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 02 00 83 FF FF | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position-32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |

| | | |
|---|---|---|
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

**Relative position control of both motors:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 01 00 7D 00 00 | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position 32000 |
| 23 7A 60 02 00 7D 00 00 | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position 32000 |
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 01 00 83 FF FF | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position-32000 |
| 23 7A 60 02 00 83 FF FF | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position-32000 |
| 2B 40 60 00 4F 00 00 00 | 60 40 60 00 00 00 00 00 | Start relative movement |
| 2B 40 60 00 5F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motors |

**Absolute position control of both motors:**

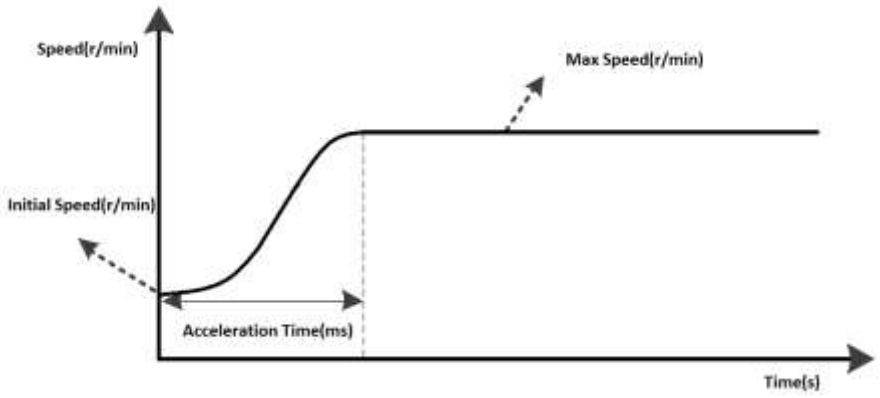
| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 7A 60 01 00 7D 00 00 | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position 32000 |
| 23 7A 60 02 00 7D 00 00 | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position 32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 23 7A 60 01 00 83 FF FF | 60 7A 60 01 00 00 00 00 | Set the left motor<br>Target position-32000 |
| 23 7A 60 02 00 83 FF FF | 60 7A 60 02 00 00 00 00 | Set the right motor<br>Target position-32000 |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Start absolute movement |
| 2B 40 60 00 1F 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motors |

※Note: In position control, the switch of control word will control both motors at the same time, so when controlling a single motor, the target position of the other motor should be given 0

## 3.4. PROFILE VELOCITY MODE

### 3.4.1 Profile Velocity Mode Description

The acceleration curve in Profile velocity mode is shown in the figure below. Differ from Profile position mode, Profile velocity mode only needs to set 3 parameters: initial speed (address 200800h), target speed (address 608100h), acceleration time (address 608300h). After the motor accelerates to the maximum speed according to set 3 parameters , it will run uniformly at the maximum speed.




Profile Velocity Mode acceleration curve

**Control word and Status word**

※Control word in profile velocity mode is controlled by bit8

| Byte | Name | Function Description |
|---|---|---|
| Bit8 | Halt | 0: Implement motion;<br>1: Stop motion. |

※ Bit10 and Bit12 of status word display driver status

| Byte | Name | Function Description |
|---|---|---|
| Bit10 | Target reached | 0: Halt = 0 target speed has not been reached;<br>Halt = 1 motor is decelerating;<br>1: Halt = 0 reach target speed;<br>Halt = 1 motor speed is 0; |
| Bit12 | Speed | 0: Speed is not 0;<br>1: Speed is 0. |

※ Related Object Dictionary Content

| Index | Sub-Index | Name | Description | Type | Attribute | PDO mapping | Defaults |
|---|---|---|---|---|---|---|---|
| 200Fh | 00 | Synchronous/asynchronous control flag | 0: Asynchronous control<br>1: Synchronization control | U16 | RW | YES | 0 |
| 6040h | 00 | Controlword | Control word | U16 | RW | YES | 0 |
| 6060h | 00 | Modes_of_operation | 0: Undefined;<br>1: Profile position mode;<br>3: Profile Velocity mode;<br>4: Profile torque mode; | I8 | RW | YES | 0 |
| 6083h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_acceleration (left) | Acceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| | 02 | Profile_acceleration (right) | Acceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| 6084h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 01 | Profile_deceleration (left) | Deceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |
| | 02 | Profile_deceleration (right) | Deceleration time;<br>Range: 0-32767ms;; | U32 | RW | YES | 500ms |
| 60FFh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Target_velocity (left) | Target speed in profile velocity mode;<br>range: -1000-1000r/min; | I32 | RW | YES | 0 |
| | 02 | Target_velocity (right) | Target speed in profile velocity mode;<br>range: -1000-1000r/min; | I32 | RW | YES | 0 |
| | 03 | Left and right target speed combination<br>Low 16 is the left motor<br>High 16 is the right motor | The current movement speed of the motor,<br>unit is 0.1r/min | U32(I16+I16) | RO | YES | 0 |

### 3.4.2 Profile Velocity Mode Configuration Routine

Make the motor run based on the parameters (acceleration time 100ms, deceleration time 100ms, maximum speed 60r/min).

※ Assume that the slave station number of the driver is 1. CANopen instruction control is described in the following table:

**Profile velocity mode asynchronous control initialization:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2F 0F 20 00 00 00 00 00 | 60 0F 20 00 00 00 00 00 | Set asynchronous control |
| 2F 60 60 00 03 00 00 00 | 60 60 60 00 00 00 00 00 | Set profile velocity mode |
| 23 83 60 01 64 00 00 00 | 60 83 60 01 00 00 00 00 | Set the left motor<br>Acceleration time 100ms |
| 23 83 60 02 64 00 00 00 | 60 83 60 02 00 00 00 00 | Set the right motor<br>Acceleration time 100ms |
| 23 84 60 01 64 00 00 00 | 60 84 60 01 00 00 00 00 | Set the left motor<br>Deceleration time 100ms |
| 23 84 60 02 64 00 00 00 | 60 84 60 02 00 00 00 00 | Set the right motor<br>Deceleration time 100ms |
| 2B 40 60 00 06 00 00 00 | 60 40 60 00 00 00 00 00 | Enable motor |
| 2B 40 60 00 07 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | |

**Velocity control of the left motor:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 FF 60 01 00 64 00 00 | 60 FF 60 01 00 00 00 00 | Set the left motor<br>Target speed 100rpm |
| 23 FF 60 01 9C FF FF FF | 60 FF 60 01 00 00 00 00 | Set the left motor<br>Target speed -100rpm |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

**Velocity control of the right motor:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 FF 60 02 00 64 00 00 | 60 FF 60 02 00 00 00 00 | Set the right motor<br>Target target speed 100rpm |
| 23 FF 60 02 9C FF FF FF | 60 FF 60 02 00 00 00 00 | Set the right motor<br>Target speed -100rpm |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

**Profile velcity mode synchronization control initialization:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2F 0F 20 00 01 00 00 00 | 60 0F 20 00 00 00 00 00 | Set synchronization control |
| 2F 60 60 00 03 00 00 00 | 60 60 60 00 00 00 00 00 | Set profile velocity mode |
| 23 83 60 01 64 00 00 00 | 60 83 60 01 00 00 00 00 | Set the left motor<br>Acceleration time 100ms |
| 23 83 60 02 64 00 00 00 | 60 83 60 02 00 00 00 00 | Set the right motor<br>Acceleration time 100ms |
| 23 84 60 01 64 00 00 00 | 60 84 60 01 00 00 00 00 | Set the left motor<br>Deceleration time 100ms |
| 23 84 60 02 64 00 00 00 | 60 84 60 02 00 00 00 00 | Set the right motor<br>Deceleration time 100ms |
| 2B 40 60 00 06 00 00 00 | 60 40 60 00 00 00 00 00 | Enable motor |
| 2B 40 60 00 07 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | |

**Synchronous velocity control:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 23 FF 60 03 00 64 00 00 | 60 FF 60 03 00 00 00 00 | Set synchronization control<br>Target speed 100rpm |
| 23 FF 60 03 9C FF 9C FF | 60 FF 60 03 00 00 00 00 | Set synchronization control<br>Target speed -100rpm |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motors |

*※Note: Synchronous target speed: low 16 is the left motor, high 16 is the right motor.*

# 3.5. PROFILE TORQUE MODE

## 3.5.1 Profile Torque Mode Description

In Profile torque mode, the value of operation mode object 6060h needs to be set to 4. When the operation mode status reads the register of object 6061h as 4, the relevant operations of HM operation mode can be performed. The objects involved in this mode are as follows:

**Control word and Status word**

※ Control word in profile torque mode is controlled by bit8

| Type | Name | Function Description |
|---|---|---|

| | | |
|---|---|---|
| Bit8 | Halt | 0: Implement motion;<br>1: Stop motion. |

※ Bit10 and Bit12 of status word display driver status

| Type | Name | Function Description |
|---|---|---|
| Bit8 | Torque attained | 0: torque has not been attained;<br>1: attain torque; |
| Bit10 | Target reached | 0: Halt = 0 target torque has not been reached;<br>Halt = 1 motor is acceleration;<br>1: Halt = 0 reach target torque;<br>Halt = 1 motor speed is 0; |

※ Related Object Dictionary Content

| Index | Sub-Index | Name | Description | Type | Attribute | PDO mapping | Defaults |
|---|---|---|---|---|---|---|---|
| 200Fh | 00 | Synchronous/asynchronous control flag | 0: Asynchronous control<br>1: Synchronization control | U16 | RW | YES | 0 |
| 6040h | 00 | Controlword | Control word | U16 | RW | YES | 0 |
| 6060h | 00 | Modes_of_operation | 0: Undefined;<br>1: Profile position mode;<br>3: Profile Velocity mode;<br>4: Profile torque mode; | I8 | RW | YES | 0 |
| 6071h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RW | NO | 2 |
| | 01 | Target_torque (left) | Unit: mA<br>Range: -30000~30000; | I16 | RW | YES | 0 |
| | 02 | Target_torque (right) | Unit: mA<br>Range: -30000~30000; | I16 | RW | YES | 0 |
| | 03 | Left and right target torque combination<br>Low 16 is the left motor<br>High 16 is the right motor | Unit: mA<br>Range: -30000~30000; | U32(I16+I16) | RW | YES | 0 |
| 6087h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Torque_slope (left) | Current/1000/second;<br>Unit: mA/s; | U32 | RW | YES | 300ms |
| | 02 | Torque_slope (right) | Current/1000/second;<br>Unit: mA/s; | U32 | RW | YES | 300ms |

### 3.5.2 Profile Torque Mode Configuration Routine

Complete torque work, target torque is 100mA.

※ Assume that the slave station number of the driver is 1. CANopen instruction control is described in the following table:

**Profile torque mode asynchronous control initialization:**

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2F 0F 20 00 00 00 00 00 | 60 0F 20 00 00 00 00 00 | Set asynchronous control |

| | | |
|---|---|---|
| 2F 60 60 00 04 00 00 00 | 60 60 60 00 00 00 00 00 | Set profile torque mode |
| 23 87 60 01 64 00 00 00 | 60 87 60 01 00 00 00 00 | Set the left motor<br>Torque slope 100mA/s |
| 23 87 60 02 64 00 00 00 | 60 87 60 02 00 00 00 00 | Set the right motor<br>Torque slope 100mA/s |
| 2B 40 60 00 06 00 00 00 | 60 40 60 00 00 00 00 00 | Enable motor |
| 2B 40 60 00 07 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | |

### Torque control of the left motor:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2B 71 60 01 E8 03 00 00 | 60 71 60 01 00 00 00 00 | Set the left motor<br>Target torque 1000mA/s |
| 2B 71 60 01 18 FC FF FF | 60 71 60 01 00 00 00 00 | Set the left motor<br>Target torque -1000mA/s |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

### Torque control of the right motor:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2B 71 60 02 E8 03 00 00 | 60 71 60 02 00 00 00 00 | Set the right motor<br>Target torque 1000mA/s |
| 2B 71 60 02 18 FC FF FF | 60 71 60 02 00 00 00 00 | Set the right motor<br>Target torque -1000mA/s |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motor |

※Note: The left/right target torque is 16-bit data, the SDO write command word is 0x2B, the synchronization target torque is 32-bit data, and the SDO write command word is 0x23.

### Profile torque mode synchronization control initialization:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2F 0F 20 00 01 00 00 00 | 60 0F 20 00 00 00 00 00 | Set synchronization control |
| 2F 60 60 00 04 00 00 00 | 60 60 60 00 00 00 00 00 | Set profile torque mode |
| 23 87 60 01 64 00 00 00 | 60 87 60 01 00 00 00 00 | Set the left motor<br>Torque slope 100mA/s |
| 23 87 60 02 64 00 00 00 | 60 87 60 02 00 00 00 00 | Set the right motor<br>Torque slope 100mA/s |
| 2B 40 60 00 06 00 00 00 | 60 40 60 00 00 00 00 00 | Enable motor |
| 2B 40 60 00 07 00 00 00 | 60 40 60 00 00 00 00 00 | |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | |

### Synchronous torque control:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2B 71 60 03 E8 03 E8 03 | 60 71 60 03 00 00 00 00 | Set synchronization control<br>Target torque 1000mA/s |

| | | |
|---|---|---|
| 2B 71 60 03 18 FC 18 FC | 60 71 60 03 00 00 00 00 | Set synchronization control<br>Target torque -1000mA/s |
| 2B 40 60 00 00 00 00 00 | 60 40 60 00 00 00 00 00 | Stop motors |

※*Note: Synchronous target speed: low 16 is the left motor, high 16 is the right motor.*

## 3.6. EMERGENCY STOP INSTRUCTION

The emergency stop can be controlled by an external input signal, or it could be achieved by switching the state of the control word through communication.

※ Related Object Dictionary Content

| Index | Sub-Index | Name | Description | Type | Attribute | PDO mapping | Defaults |
|---|---|---|---|---|---|---|---|
| 2003h | 00 | Input signal status | 2 input signal level status<br>Bit0-Bit1: X0-X1 input level status; | U16 | RO | YES | 0 |
| 2030h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 8 |
| | 01 | Input terminal effective level | Bit0: Input terminal X0 control bit;<br>Bit1: Input terminal X1 control bit;<br>Bit2-Bit15: Reserved;<br>0: Default;<br>1: Level inversion;<br>The driver defaults to the input terminal level rising edge or high level active; | U16 | RW | YES | 0 |
| | 02 | Function selection of input terminal X0 terminal | 0: Undefined;<br>1-6: NC;<br>9: Emergency stop signal; | U16 | RW | YES | 9 |
| | 03 | Function selection of input terminal X1 terminal | | U16 | RW | YES | 0 |
| | 06 | Output terminal effective level | Bit0: Control bit of output terminal Y0;<br>Bit1: Control bit of output terminal Y1;<br>0: Default;<br>1: Level inversion;<br>The driver defaults to the input terminal level rising edge or high level active; | U16 | RW | YES | 0 |
| | 07 | Function selection of output terminal Y0 | Brake state:<br>0: open | U16 | RW | YES | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | terminal | 1: Close | | | | |
| | 08 | Function selection of output terminal Y1 terminal | Brake state:<br>0: open<br>1: Close | U16 | RW | YES | 0 |

Emergency Stop Instruction:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2B 40 60 00 0F 01 00 00 | 60 40 60 00 00 00 00 00 | The motor stops and remains enabled |
| 2B 40 60 00 0F 00 00 00 | 60 40 60 00 00 00 00 00 | Motor enable (release emergency stop state) |

## 3.7 ERROR AND CLEAR

ZLAC8015D supports protections such as overvoltage and overcurrent. All fault information can be obtained by reading object 0x603F.

| 0x603F | Function Desription |
|---|---|
| 0x0000 0000h | No error |
| 0x0000 0001h | Over-voltage |
| 0x0000 0002h | Under-voltage |
| 0x0000 0100h | EEPROM read and write error |
| 0x0004 0004h | Over-current |
| 0x0008 0008h | Overload |
| 0x0010 0010h | Current out of tolerance |
| 0x0020 0020h | Encoder out of tolerance |
| 0x0040 0040h | Velocity out of tolerance |
| 0x0080 0080h | Reference voltage error |
| 0x0200 0200h | Hall error |

*※Note: No error/over-voltage/under-voltage/EEPROM read and write error are common faults are error for both motors. Other errors are divided into left and right drives. The high 16 bits of the error code are left driver, and the low 16 bits are right driver.*

Clear error:

| Master Station (COB-ID:0x601) | Slave Station (COB-ID:0x581) | Function Description |
|---|---|---|
| 2B 40 60 00 80 00 00 00 | 60 40 60 00 00 00 00 00 | Clear error |

# APPENDIX A. OBJECT DICTIONARY

ZLAC8015D series bus type servo driver parameter register includes 3 parts: 1000h-1FFFh register defined by CIA301, 2000h-2FFFh register defined by the manufacturer and 6000h-6FFFh register defined by CIA402.

1000h-1FFFh register is CANopen related basic communication parameter defined by CIA301, including SDO, PDO, and mapping register.

2000h-2FFFh register is a manufacturer-defined register content, subdivision and current modification could be implemented within this group of parameters.

6000h-6FFFh register is motion parameter related to motion control defined by CIA402, including profile position mode, profile velocity mode, profile torque mode, other operation mode registers, and related motion parameter registers.

| Index | Sub-index | Name | Description | Type | Attributes | PDO mapping | Defaults |
|---|---|---|---|---|---|---|---|
| **CiA301 Basic Communication Parameter Group** | | | | | | | |
| 1000h | 00 | Device Type | This device supports CiA301, CiA402 protocol | U32 | RO | NO | 0X00040192 |
| 1001h | 00 | Error Register | Driver current error status | U8 | RO | NO | 0 |
| 1005h | 00 | COB ID SYNC | COB identifier of synchronization message | U32 | RW | NO | 0x80 |
| 1009h | 00 | Manufacturer Hardware Version | Hardware version | U16 | RO | NO | - |
| 100Ah | 00 | Manufacturer Software Version | Software version | U16 | RO | NO | - |
| 1014h | 00 | COB ID EMCY | EMNC Emergency Message COB | U32 | RW | NO | 0x80 |
| 1017h | 00 | Producer Heartbeat Time | Producer heartbeat interval, unit: ms | U16 | RW/S | NO | 0 |
| 1018h | 00 | Number of entries | Number of entries | U8 | RO | NO | 5 |
| | 01 | Vendor Id | Vendor ID | U32 | RO | NO | 0x0100 |
| | 02 | Product Code | Supplier Product Number | U32 | RO | NO | 0x0001 |
| 1200h | 00 | Number of entries | Number of entries | U8 | RO | NO | 2 |
| | 01 | COB ID Client to Server | COB ID Client to Server | U32 | RO | NO | 600h+Node-ID |
| | 02 | COB ID Server to Client | COB ID Server to Client | U32 | RO | NO | 580h+Node-ID |
| 1400h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | RPDO0 COB ID | U32 | RO | NO | 200+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1401h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | RPDO1 COB ID | U32 | RO | NO | 300+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1402h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | RPDO2 COB ID | U32 | RO | NO | 400+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1403h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | RPDO3 COB ID | U32 | RO | NO | 500+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1600h | 00 | Number of entries | Number of entries | U8 | RO | NO | 1 |
| | 01 | PDO Mapping Entry | Map to 6040h register | U32 | RW/S | NO | 60400010h |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1601h | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| 1602h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1603h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1800h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | TPDO0 COB ID | U32 | RO | NO | 180+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1801h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | TPDO1 COB ID | U32 | RO | NO | 280+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1802h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | TPDO2 COB ID | U32 | RO | NO | 380+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1803h | 00 | Number of entries | Number of sub-indexes | U8 | RO | NO | 5 |
| | 01 | COB ID | TPDO3 COB ID | U32 | RO | NO | 480+Node-ID |
| | 02 | Transmission Type | Transmission Type | U8 | RW/S | NO | FFh |
| | 03 | Inhibit Time | Inhibit Time | U16 | RW/S | NO | 0 |
| | 04 | Compatibility Entry | Compatibility Entry | U8 | RW | NO | 0 |
| | 05 | Event Timer | Event Timer | U16 | RW/S | NO | 0 |
| 1A00h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1A01h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1A02h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| 1A03h | 00 | Number of entries | Number of entries | U8 | RO | NO | 0 |
| | 01 | PDO Mapping Entry | Unmapped | U32 | RW/S | NO | - |
| | 02 | PDO Mapping Entry_2 | Unmapped | U32 | RW/S | NO | - |
| | 03 | PDO Mapping Entry_3 | Unmapped | U32 | RW/S | NO | - |
| | 04 | PDO Mapping Entry_4 | Unmapped | U32 | RW/S | NO | - |
| **Factory custom parameters** | | | | | | | |
| 2000h | 00 | Communication power-down protection time | Driver and host communication power-down time setting<br>Unit: ms<br>Range: 0-32767; | U16 | RW | YES | 1000 |
| 2001h | 00 | RS485 custom driver node number | When the external dial switch is 0, it can be set to 4~127;<br>When the external dial switch is 1-3, this bit is invalid | U16 | RW | YES | 4 |
| 2002h | 00 | RS485 custom communication baudrate | 0: 256000bps<br>1: 128000bps<br>2: 115200bps<br>3: 57600bps<br>4: 38400bps<br>5: 19200bps<br>6: 9600bps | U16 | RW | YES | 2 |
| 2003h | 00 | Input signal status | 2 input signal level status;<br>Bit0 – Bit1: X0 ~ X1 input level status; | U16 | RO | YES | 0 |
| 2004h | 00 | Output signal status | 2 output signal level status;<br>Bit0 ~ Bit1: Y0 ~ Y1 output status; | U16 | RO | YES | 0 |
| 2005h | 00 | Clear postion feedback | Clear position feedback in profile position mode<br>0: Invalid;<br>1: Cleared position feedback.<br>Not save<br>High 8 bits: (left)<br>Low 8 bits: (right) | U16 | RW | YES | 0 |
| 2006h | 00 | Clear the current position in absolute position mode | Clear the current position in absolute position mode<br>0: Invalid;<br>1: Clear the current position;<br>High 8 bits: (left)<br>Low 8 bits: (right) | U16 | RW | YES | 0 |
| 2007h | 00 | Shaft lock method | 0: Not enable, not lock the | U16 | RW | YES | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | shaft;<br>1: Not enable, lock the shaft; | | | | |
| 2008h | 00 | Motor max speed | Motor max working speed<br>Unit: r/min;<br>Range: 1-1000r/min; | U16 | RW | YES | 1000 |
| 2009h | 00 | Register parameter setting | 0: Invalid;<br>1: Restore factory settings; | U16 | RW | YES | 0 |
| 200Ah | 00 | CAN custom driver node number | When the external dial switch is set to 0, it can be set from 4 to 127; | U16 | RW | YES | 4 |
| 200Bh | 00 | CAN custom communication baudrate | 0: 1000 Kbit/s<br>1: 500 Kbit/s<br>2: 250 Kbit/s<br>3: 125 Kbit/s<br>4: 100 Kbit/s<br>5: 50 Kbit/s<br>6: 25 Kbit/s | U16 | RW | YES | 1 |
| 200Ch | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Motor pole pair number (left) | 4-64 | U16 | RW | YES | 15 |
| | 02 | Motor pole pair number (right) | 4-64 | U16 | RW | YES | 15 |
| 200Dh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Starting speed (left) | The initial speed at which the movement started;<br>Unit The initial speed at which the movement started;<br>Unit: r/min;<br>Range: 1-256r/ min; | U16 | RW | YES | 1r/min |
| | 01 | Starting speed (right) | The initial speed at which the movement started;<br>Unit The initial speed at which the movement started;<br>Unit: r/min;<br>Range: 1-256r/ min; | U16 | RW | YES | 1r/min |
| 200Eh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Encoder wire number setting (left) | 0-4096 | U16 | RW | YES | 1024 |
| | 02 | Encoder wire number setting (right) | 0-4096 | U16 | RW | YES | 1024 |
| 200Fh | 00 | Synchronous/asynchronous control flag | 0: Asynchronous control<br>1: Synchronous control | U16 | RW | YES | 0 |
| 2010h | 00 | Whether the parameter are updated to EEPROM | Whether the communication write function code value is | U16 | RW | YES | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | updated to EEPROM<br>0: Invalid;<br>1: Parameters with attribute RW / S are updated to EEPROM synchronously; | | | | |
| 2011h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Offset angle of motor and Hall (left) | Unit: 1°;<br>Range: -360-+360 | I16 | RW | YES | 0 |
| | 02 | Offset angle of motor and Hall (right) | Unit: 1°;<br>Range: -360-+360 | I16 | RW | YES | 0 |
| 2012h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Overload factor (left) | Range: 0-300,<br>Unit: % | U16 | RW | YES | 200 |
| | 02 | Overload factor (right) | Range: 0-300,<br>Unit: % | U16 | RW | YES | 200 |
| 2013h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Motor temperature protection threshold (left) | Unit: 0.1°C;<br>Range: 0-1200 | U16 | RW | YES | 800 |
| | 02 | Motor temperature protection threshold (right) | Unit: 0.1°C;<br>Range: 0-1200 | U16 | RW | YES | 800 |
| 2014h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Rated current (left) | Driver rated output current<br>Unit: 0.1A;<br>Range: 0-150 | U16 | RW | YES | 150 |
| | 02 | Rated current (right) | Driver rated output current<br>Unit: 0.1A;<br>Range: 0-150 | U16 | RW | YES | 150 |
| 2015h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Maximum current (left) | Maximum driver output current<br>Unit: 0.1A;<br>Range: 0-300 | U16 | RW | YES | 300 |
| | 02 | Maximum current (right) | Maximum driver output current<br>Unit: 0.1A;<br>Range: 0-300 | U16 | RW | YES | 300 |
| 2016h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Overload protection time (left) | Driver overload protection time<br>Unit: 10ms;<br>Range: 0-6553 | U16 | RW | YES | 300 |
| | 02 | Overload protection | Driver overload protection | U16 | RW | YES | 300 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | time (right) | time<br>Unit: 10ms;<br>Range: 0-6553 | | | | |
| 2017h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Out of tolerance alarm threshold (left) | Encoder tolerance threshold<br>Unit: *10counts;<br>Range: 1-6553 | U16 | RW | YES | 409 |
| | 02 | Out of tolerance alarm threshold (right) | Encoder tolerance threshold<br>Unit: *10counts;<br>Range: 1-6553 | U16 | RW | YES | 409 |
| 2018h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed smoothing coefficient | 0-30000 | U16 | RW | YES | 50 |
| | 02 | R speed smoothing coefficient | 0-30000 | U16 | RW | YES | 50 |
| 2019h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L current loop proportional coefficient | 0-30000 | U16 | RW | YES | 600 |
| | 02 | R current loop proportional coefficient | 0-30000 | U16 | RW | YES | 600 |
| 201Ah | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L current loop integral gain | 0-30000 | U16 | RW | YES | 300 |
| | 02 | R current loop integral gain | 0-30000 | U16 | RW | YES | 300 |
| 201Bh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L feedforward output smoothing coefficient | 0-30000 | U16 | RW | YES | 100 |
| | 02 | R feedforward output smoothing coefficient | 0-30000 | U16 | RW | YES | 100 |
| 201Ch | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L torque output smooth coefficient | 0-30000 | U16 | RW | YES | 100 |
| | 02 | R torque output smooth coefficient | 0-30000 | U16 | RW | YES | 100 |
| 201Dh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed proportional gain Kp | 0-30000 | U16 | RW | YES | 500 |
| | 02 | R speed proportional gain Kp | 0-30000 | U16 | RW | YES | 500 |
| 201Eh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed integral gain Ki | 0-30000 | U16 | RW | YES | 100 |
| | 02 | R speed integral gain Ki | 0-30000 | U16 | RW | YES | 100 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 201Fh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed feedforward gain Kf | 0-30000 | U16 | RW | YES | 1000 |
| | 02 | R speed feedforward gain Kf | 0-30000 | U16 | RW | YES | 1000 |
| 2020h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L position proportional gain Kp | 0-30000 | U16 | RW | YES | 50 |
| | 02 | R position proportional gain Kp | 0-30000 | U16 | RW | YES | 50 |
| 2021h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L position feedforward gain Kf | 0-30000 | U16 | RW | YES | 200 |
| | 02 | R position feedforward gain Kf | 0-30000 | U16 | RW | YES | 200 |
| 2022h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed observer coefficient 1 | 0-30000 | U16 | RW | YES | 1000 |
| | 02 | R speed observer coefficient 2 | 0-30000 | U16 | RW | YES | 1000 |
| 2023h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed observer coefficient 2 | 0-30000 | U16 | RW | YES | 750 |
| | 02 | R speed observer coefficient 2 | 0-30000 | U16 | RW | YES | 750 |
| 2024h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed observer coefficient 3 | 0-30000 | U16 | RW | YES | 350 |
| | 02 | R speed observer coefficient 3 | 0-30000 | U16 | RW | YES | 350 |
| 2025h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | L speed observer coefficient 4 | 0-30000 | U16 | RW | YES | 1000 |
| | 02 | R speed observer coefficient 4 | 0-30000 | U16 | RW | YES | 1000 |
| 2030h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 8 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 01 | Input terminal effective level | Bit0: Input terminal X0 control bit;<br>Bit1: Input terminal X1 control bit;<br>Bit2- Bit15: Reserved;<br>0: Default;<br>1: level inversion;<br>The driver's default input terminal level rising edge or high level is valid; | U16 | RW | YES | 0 |
| | 02 | Input terminal X0 terminal function selection | 0: undefined;<br>1-6：NC；<br>9: emergency stop signal; | U16 | RW | YES | 9 |
| | 03 | Input terminal X1 terminal function selection | | U16 | RW | YES | 0 |
| | 04 | NC | | U16 | RW | YES | 0 |
| | 05 | NC | | U16 | RW | YES | 0 |
| | 06 | Output terminal effective level | Bit0: Output terminal Y0 control bit;<br>Bit1: Output terminal Y1 control bit;<br>0: default;<br>1: level inversion;<br>The driver's default input terminal level rising edge or high level is valid; | U16 | RW | YES | 0 |
| | 07 | Output terminal Y0 terminal function selection | Brake state<br>0: Open<br>1: Close | U16 | RW | YES | 0 |
| | 08 | Output terminal Y1 terminal function selection | Brake state<br>0: Open<br>1: Close | U16 | RW | YES | 0 |
| 2031h | 00 | Software version | Factory default | U16 | RO | NO | - |
| 2032h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Motor temperature (left) | Unit: 0.1°C;<br>Range: 0-1200°C | U16 | RO | YES | 800 |
| | 02 | Motor temperature (right) | Unit: 0.1°C;<br>Range: 0-1200°C | U16 | RO | YES | 800 |
| 2033h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Motor status register (left) | The driver controls the motor motion state<br>0: The motor is stationary;<br>1: Motor is running; | U16 | RO | YES | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 02 | Motor status register (right) | The driver controls the motor motion state<br>0: The motor is stationary;<br>1: Motor is running; | U16 | RO | YES | 0 |
| 2034h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Hall input state (left) | 0-7;<br>If 0 or 7 appears, it is a Hall error | U16 | RO | YES | 0 |
| | 02 | Hall input state (right) | 0-7;<br>If 0 or 7 appears, it is a Hall error | U16 | RO | YES | 0 |
| 2035h | 00 | Bus voltage | Unit: 0.01V | U16 | RO | YES | 0 |
| **CiA 402 Parameter** | | | | | | | |
| 603Fh | 00 | Error_code | Factory-defined drive error conditions.<br>0000h: no errors;<br>FF01h: overvoltage;<br>FF02h: overcurrent;<br>FF04h: missing phase;<br>FF08h: Internal reference voltage error;<br>FF10h: EEPROM read and write errors; | U16 | RO | YES | 0 |
| 6040h | 00 | Controlword | Control word | U16 | RW | YES | 0 |
| 6041h | 00 | Statusword | Status word | U16 | RO | YES | 0 |
| 605Ah | 00 | Quick_stop_option_code | Driver processing method after quick stop command.<br>5: Stop normally , maintain quick stop status;<br>6: Decelerate to stop emergencely and maintain quick stop state;<br>7: Emergency stop, maintain quick stop state; | I16 | RW | NO | 5 |
| 605Bh | 00 | Shutdown_option_code | Driver processing method after the close command<br>0: invalid;<br>1: Stop normally, go to ready to switch on state; | I16 | RW | NO | 1 |
| 605Ch | 00 | Disable_operation_option_code | Driver processing mode after the disable operation command<br>0: Invalid;<br>1: Stop normally , switch to | I16 | RW | NO | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | switched on state; | | | | |
| 605Dh | 00 | Disable_operation_option_code | Driver processing mode after the disable operation command<br>0: Invalid;<br>1: Stop normally , switch to switched on state; | I16 | RW | NO | 1 |
| 6060h | 00 | Modes_of_operation | 0: undefined;<br>1: position mode;<br>3: speed mode;<br>6: Return to origin mode; | I8 | RW | YES | 0 |
| 6061h | 00 | Modes_of_operation_display | 0: undefined;<br>1: position mode;<br>3: speed mode;<br>6: Return to origin mode; | I8 | RO | YES | 0 |
| 6064h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Position_actual_value (left) | Actual position feedback, unit: count; | I32 | RO | YES | 0 |
| | 02 | Position_actual_value (right) | Actual position feedback, unit: count; | I32 | RO | YES | 0 |
| 606Ch | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Velocity_actual_value (left) | Current motor speed,<br>Unit: r / min | I32 | RO | YES | 0 |
| | 02 | Velocity_actual_value (right) | Current motor speed,<br>Unit: r / min | I32 | RO | YES | 0 |
| | 03 | Left motor and right motor speed actual value combination<br>Low 16 is the left motor,<br>High 16 is the right motor | Current motor speed,<br>Unit: r / min | U32(I16+I16) | RO | YES | 0 |
| 6071h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RW | NO | 2 |
| | 01 | Target torque (left) | Unit: mA<br>Range: -30000~30000; | I16 | RW | YES | 0 |
| | 02 | Target torque (right) | Unit: mA<br>Range: -30000~30000;; | I16 | RW | YES | 0 |
| | 03 | Left moroe and right motor target torque combination<br>Low 16 is the left motor,<br>High 16 is the right motor | Unit: mA<br>Range: -30000~30000; | U32(I16+I16) | RW | YES | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6077h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Torque actual value (left) | Unit: 0.1A<br>Range: -600~600; | I16 | RO | YES | 0 |
| | 02 | Torque actual value (right) | Unit: 0.1A<br>Range: -600~600; | I16 | RO | YES | 0 |
| | 03 | Left motor and right motor torque actual value combination<br>Low 16 is the left motor,<br>High 16 is the right motor | Unit: 0.1A<br>Range: -600~600; | U32(I16+I16) | RW | YES | 0 |
| 607Ah | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Target_position (left) | Total number of pulses in profile position mode;<br>Range:<br>Relative:<br>-0x7FFFFFFF~0x7FFFFFFF<br>Absolute:<br>-0x3FFFFFFF~0x3FFFFFFF | I32 | RW | YES | 0 |
| | 02 | Target_position (right) | Total number of pulses in profile position mode;<br>Range:<br>Relative:<br>-0x7FFFFFFF~0x7FFFFFFF<br>Absolute:<br>-0x3FFFFFFF~0x3FFFFFFF | I32 | RW | YES | 0 |
| 6081h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_velocity (left) | Max speed in profile position mode;<br>Range: 1-1000r/min; | U32 | RW | YES | 120r/min |
| | 02 | Profile_velocity (right) | Max speed in profile position mode;<br>Range: 1-1000r/min; | U32 | RW | YES | 120r/min |
| 6082h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | End_velocity (left) | Start / stop speed in profile position mode;<br>Range: 1-250r/min; | U32 | RW | YES | 1r/min |
| | 02 | End_velocity (right) | Start / stop speed in profile position mode;<br>Range: 1-250r/min; | U32 | RW | YES | 1r/min |
| 6083h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_acceleration (left) | Acceleration time;<br>Range: 0-32767ms; | U32 | RW | YES | 500ms |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 02 | Profile_acceleration (right) | Acceleration time; Range: 0-32767ms; | U32 | RW | YES | 500ms |
| 6084h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Profile_deceleration (left) | Deceleration time; Range: 32767ms; | U32 | RW | YES | 500ms |
| | 02 | Profile_deceleration (right) | Deceleration time; Range: 32767ms; | U32 | RW | YES | 500ms |
| 6085h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Quick_stop_deceleration (left) | Deceleration time; Range: 0-32767ms; | U32 | RW | YES | 10ms |
| | 02 | Quick_stop_deceleration (right) | Deceleration time; Range: 0-32767ms; | U32 | RW | YES | 10ms |
| 6087h | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Torque Slope (left) | Current/1000/second; Unit: mA/s; | U32 | RW | YES | 300ms |
| | 02 | Torque Slope (right) | Current/1000/second; Unit: mA/s; | U32 | RW | YES | 300ms |
| 60FFh | 00 | Number of sub-indexes | Number of sub-indexes | U16 | RO | NO | 2 |
| | 01 | Target_velocity (left) | Target speed in profile velocity mode; Range: -1000-1000r/min; | I32 | RW | YES | 0 |
| | 02 | Target_velocity (right) | Target speed in profile velocity mode; Range: -1000-1000r/min; | I32 | RW | YES | 0 |
| | 03 | Left and right target speed combination Low 16 is the left motor High 16 is the right moto | Target speed in profile velocity mode; Range: -1000-1000r/min; | U32(I16+I16) | RO | YES | 0 |

Note:

- U16 means unsigned 16 bits; I16 means signed 16 bits; U32 means unsigned 32 bits; I32 means signed 32 bits.