User Manual

PMC007CxSxPx Series

Miniature Integrated Stepper Motor Controller

1. Version Control

1） Update Records

Date	Author	Version	Remark
2014-10-19	huangcheng	V0.1.0	Initial
2014-11-25	Liur	V0.1.1	Fix typo
2014-12-08	huangcheng	V0.1.2	Add system control object. Signal interface J1 adds BOOT and RESET signal description. Merge external emergency stop enabled and external emergency stop trigger mode object into an object. Modify the screenshot tool and add the relevant functional instructions.
2015-5-6	huangcheng	V0.1.3	PDO function supplement
2016-6-15	wentao	V0.1.4	Add senseless stall detection
2016-9-22	Liur	V0.1.5	Add function description for close loop
2016-11-7	huangcheng	V0.1.6	Add external stop3
2016-11-25	wentao	V0.1.7	Close loop function supplement
2017-10-8	liur	V0.1.8	Add pvtb mode, update feature & function
2018-6-15	jiawei	V0.1.9	Migrated from pmc007cxsx
2018-7-23	hc	V0.2.0	Modify instruction
2018-09-27	hc	V0.2.1	1、 Add analog input function 2、 Add step notify related objects
2018-11-30	Tanlu	V0.2.2	Supplement description for Ext port
2018-12-3	huangcheng	V0.2.3	Add sensor type parameter description
2019-02-25	huangcheng	V0.2.4	1、Functional description of adding PV/PP mode 2、Adding analog location function description
2019-05-12	liur	V0.2.5	1. Add absolute encoder support; 2. Add sensorness stall detection; 3. Add smooth mode; 4. Extend to 48V supply range.
2019-8-9	huangcheng	V0.2.6	1. Add driver mode settings 2. Add power down behavior settings

Catalog

1 Introduction

1.1 Statement of intellectual property right

PMC007CxSxP series controller has been applied for the following national patent:

• Controller scheme and method have been applied for the protection of the invention patent.

• Controller circuit has been applied for the protection of utility model patent.

• Controller appearance has been applied for the protection of appearance patent protection.

PMC007CxSxP series controller has embedded firmware code, it would be considered as a violation of intellectual property protection act and regulations that any behavior of trying to destroy the function of firmware code protection. If this behavior acquires the software or other achievements of intellectual property protection without authorization of CQPUSI, CQPUSI has the right to stop such behavior by filing a lawsuit according to the act.

1.2 Disclaimer

The using method of the device and other content in the description of this manual is only used to provide convenience for you, and may be update in future version. To ensure the application conforms to the technical specifications is the responsibility of your own. CQPUSI does not make any form of statement or guarantee to the information, which include but not limited to usage, quality, performance, merchantability or applicability of specific purpose. CQPUSI is not responsible for these information and the consequences result caused by such information. If the CQPUSI device is used for life support and/or life safety applications, all risks are borne by the buyer. The buyer agrees to protect the CQPUSI from legal liability and compensation for any injury, claim, lawsuit or loss caused by the application.

2 Overview

2.1 General Description

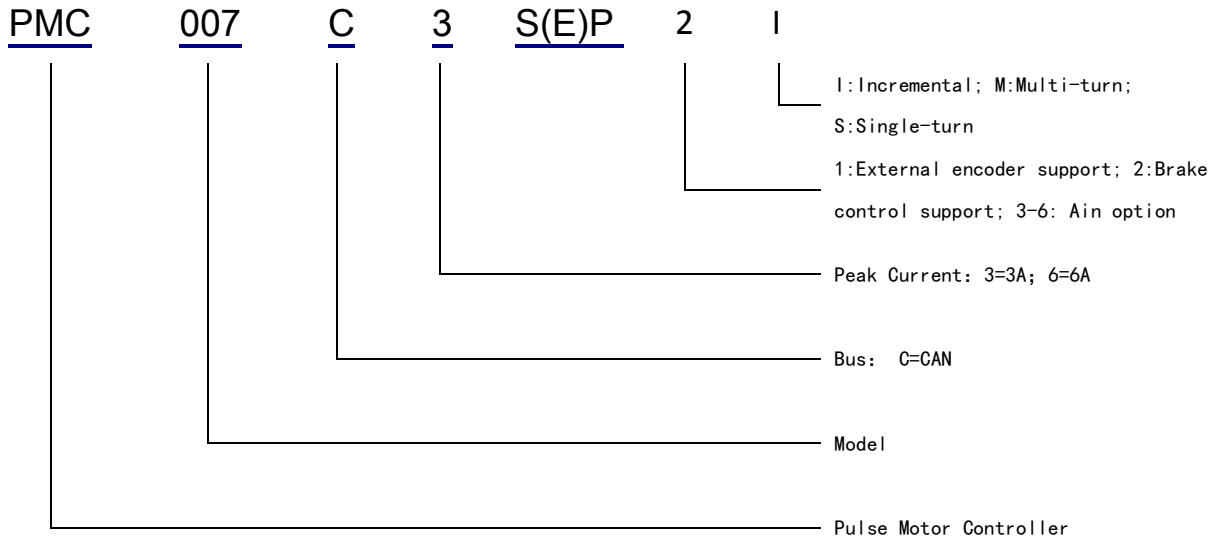
PMC007CxSxPx2 is a kind of miniature integrated stepper motor microstepping controller, which can be directly installed in the rear of 42/57/86 series stepper motor. The series controller provides a variety of models which can be chosen based on bus control of CAN and different current value. It is easy to achieve industrial control network of as many as 120 nodes, which can achieve closed-loop control based on encoder according to the requirements of user. PMC007CxSxPx adopts industrial standard CANOPEN DS301 control protocol, which not only greatly simplifies the complexity of the upper layer control system, but also maximally reserves flexibility of control, and is suitable for all kinds of high precision, wide range of industry using.

2.2 Features

- ✓ Wide range of 12-48V single voltage supply
- ✓ Output current 0.3A ~ 6A. Adjustable phase current by commands
- ✓ Automatic control of S curve acceleration and deceleration
- ✓ Support Position/Velocity/PP/PV/PVT/SP/Analog Position/Analog velocity etc. motion mode
- ✓ Support 0/2/4/8/16/32/64/128/256 microstepping resolution
- ✓ Suitable for 4/6/8 lines of 2 phase stepper motor
- ✓ Solenoid brake control function
- ✓ Support sensorless stall detection
- ✓ Support 200-2000CPR incremental encoder;
- ✓ Support SSI/BISS multi-turn absolute encoder;
- ✓ Burning and off-line automatic execution of custom program
- ✓ Support dragging graphic programming
- ✓ Support LUA script language programming
- ✓ Miniature size 42mmx42mmx18mm
- ✓ Precision aluminum shell, conducive to the protection and heat dissipation
- ✓ Automatic over-temperature, over-current, under-voltage and overvoltage protection

2.3 Production & Ordering Information

In order to serve you quicker and better, please provide the product number in following format when ordering PMC007CxSxPx:



Remark:

E: Closed-loop type.

P: Enhancement type.

Please be sure to contact the sales staff to confirm whether the required model is in a normal state of supply before placing an order.

3 Connector Description

3.1 Terminal port location

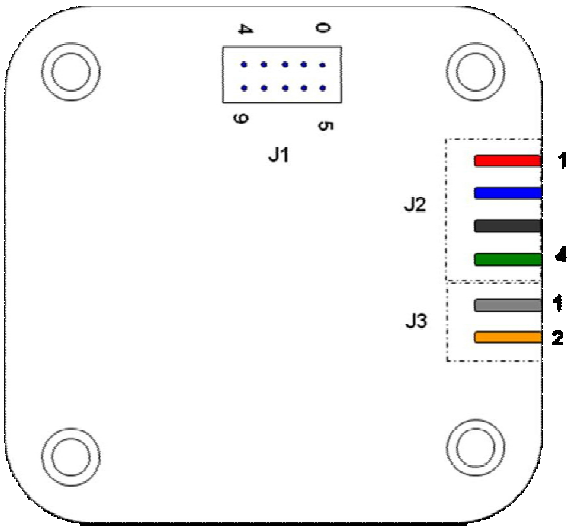


Figure 3-1

3.2 Motor connection J2

Pin no:	1	2	3	4
Designator:	M10	M11	M20	M21

Description:

M10, M11: the stepper motor phase A

M20, M21: the stepper motor phase B

WARNING: Incorrect connection of power or phase will permanently damage the controller! (Corresponding to red, blue and black, green line order in closed-loop type).

3.3 Power connection J3

Pin no:	1	2
Designator:	GND	VCC

Description:

VCC: Supply voltage, 12~48VDC.

GND: Supply voltage ground.

Remark:

1. When the current exceeds 3A, It is recommended to connect an electrolytic capacitor at least 1000uF near the J3 interface.

2. Hot plug is prohibited, which may damage the controller permanently!

3.4 Signal connection J1

Pin no:	0	1	2	3	4
Designator:	GND	Coil+	Coil-	CANH	CANL
Pin no:	5	6	7	8	9
Designator:	DVDD	AIN/EXT2	EXT1	GPIO8	FSET

Description:

DVDD: Controller voltage output(+5V). Maximum current is 100mA.

GND: Controller digital ground.

EXT1: External limit switch signal input 1, 0~24V.

AIN: Analog input for adjusting speed, 0~3.3V, (optional4-20mA).

GPIO8: Digital input and output,0~3.3V.

FSET: Factory reset input,0~3.3V. Low level effectiveness.

CANH: Connect to the CAN transceiver module.

CANL: Connect to the CAN transceiver module.

Coil+: Solenoid valve/brake positive control terminal, its voltage is equal to the voltage of power supply VCC; or encoder interface;

Coil-: Solenoid valve/brake negative control terminal; or encoder interface;

WARNING: The voltage of all signal ports must be between -0.3V~+5.3V, otherwise, it may cause permanent damage to the controller.

3.5 CAN Network Operation

Using the CAN bus connection can reach a maximum 5000 meters transmission

distance. It provides a network scheme that uses CAN bus to connect multiple PMC007CxSxPx controllers in the figure 3-2, which are compatible with CAN2.0A and CAN2.0B two technical specifications, and can be connected to 127 nodes.

Note: it is recommended to use the CAN bus specified 120 ohm shielded twisted pair, and the ends of the twisted pair are required to connect a 120 ohm termination resistor. In addition, The PTA2C in the diagram is a USB-CAN conversion module provided by the third party.

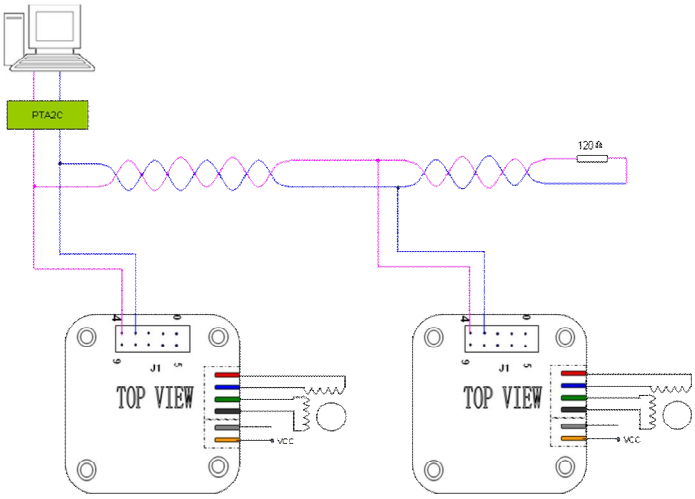


Figure 3-2

PMC007CxSxPx supports standard DS301 CANopen protocol, CQPUSI provides a dedicated debugging tool software PUSICAN for debugging PMC007CxSxPx network. The tool software currently supports a variety of USB2CAN modules which has been mainstream brands on the market.

3.6 Limit switch connection

There is a dedicated pin Ext1 in PMC007CxSxPx controller, which are used to connect the external limit(zero point, home position) switch. The trigger mode of each pin can be selected by the instruction. Default trigger mode is falling edge trigger. At this time, the level of the EXT1 is from low level to high level because the Ext1 pin integrates a fixed drop-down resistor and a reverse buffer as shown below Figure 3-3.

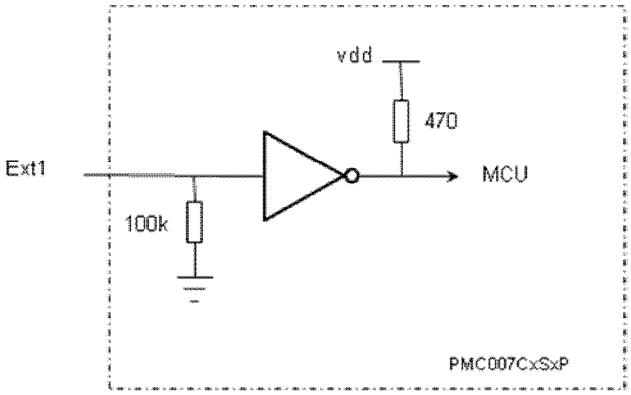


Figure 3-3

The input level range of EXT1 is 0~24V. When input voltage exceeds 3V, it is considered as a high level. The input port can be directly connected to the 5~24V PNP output type sensor, as shown in figure 3-4 left. For NPN or sensors which is open-collector output type, A 1Kohm pull-up resistor is required between the power supply and the Ext1 pin, as shown in figure 3-4 right.

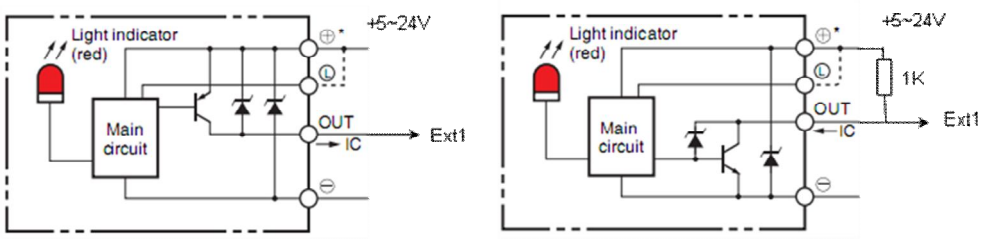


Figure 3-4

When using U-slot optocoupler, the transmitter terminal can be directly connected to GPIO8 and GND, and the collector of the receiver is connected to DVDD, Emitter is connected to EXT1, as shown in Figure 3-5 below.

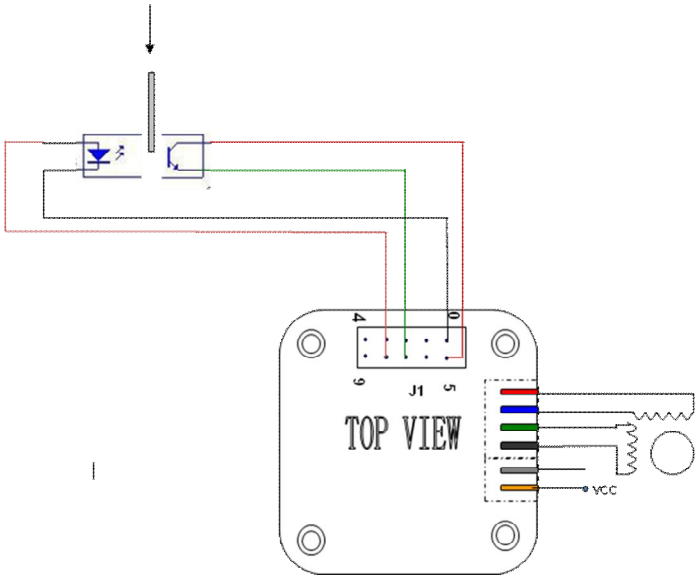


Figure 3-5

3.7 Multiple limit switch connection

The AIN and GPIO8 pin of the PMC007CxSxPx controller can also be multiplexed into two other limit switch inputs, but its acceptance voltage should not exceed 5V. When using 24V DC three wire NPN open-collector output type proximity switch, such as OMRON E2EC/ X□C□or E2E-X□D1S series. The connection is shown in the following figure and AIN and GPIO8 pin needs to be configured to pull up enable. Since the input port of the controller can only accept the 5V voltage range, Therefore, the 24V DC three line NPN open-collector proximity switch or 24V PNP type proximity switch cannot be connected to the controller.

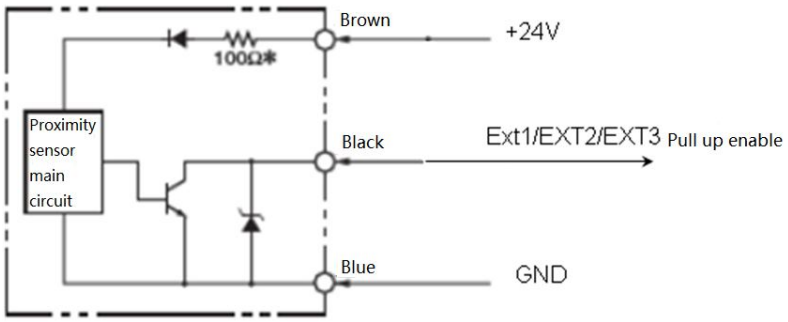


Figure 3-6

If the normal U-slot optocoupler is connected, the GPIO8 port can be used to drive the emitter LED. Set Ext2 port to pulldown enable if falling edge trigger mode is chose as shown Figure 3-7 left, set Ext2 port to pullup enable if rising edge trigger mode is chose as shown Figure 3-7 right.

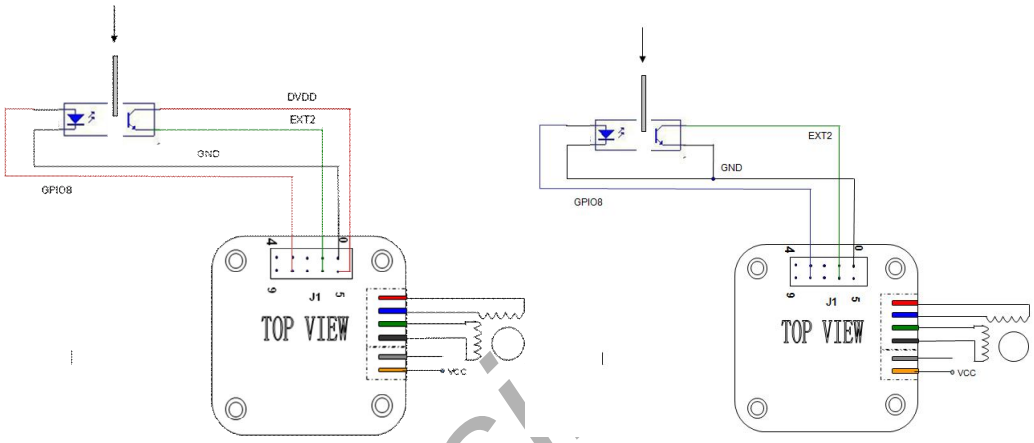


Figure 3-7

If the PNP or NPN proximity sensors with internal circuits are used, the care should be taken for choosing trigger mode. For figure 3-8 left PNP sensor, set Ext2 port to falling edge trigger mode and pulldown enable. For giure 3-8 right NPN sensor, set Ext2 port to rising edge trigger mode and pullup enable.

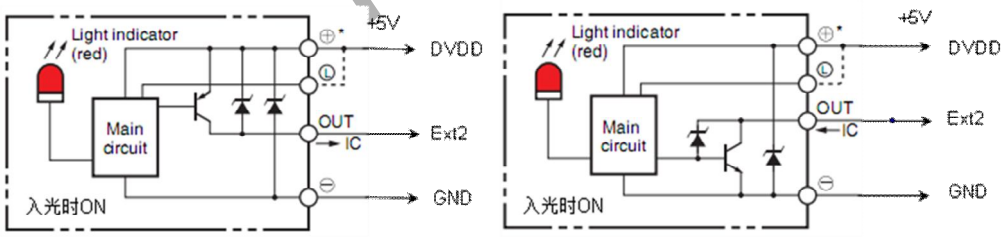


Figure 3-8

3.8 Mechanical switch connection

When using mechanical button switch or relay contact as the limit, for EXT1, the connection mode is as following figure left, using the falling edge trigger mode. For EXT2, the connection mode is as following figure right, enabling internal pull-up resistance, using falling edge trigger mode, as shown in figure 3-10 below.

Figure 3-10

3.9 Analog Adjusting Speed

PMC007CxSxPx controller can use analog adjusting speed control function in offline mode. In this application, the AIN pin is used as the analog input port, as shown in Figure 3-6 below. It can also be directly connected to the external input voltage, whose range is 0~3.3V. When using PLC or other industrial control equipment to output 4 ~ 20mA analog control, special comments are needed to distinguish versions.

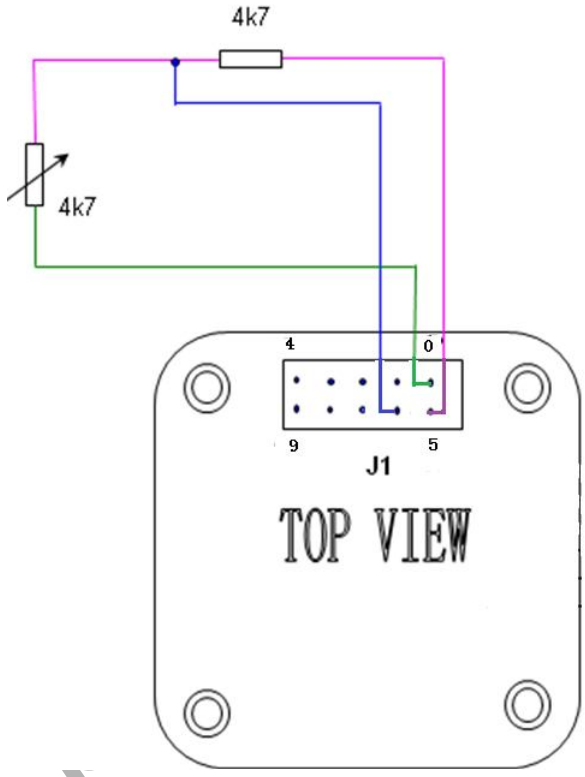


Figure 3-6

3.10 Solenoid valve / brake connection

PMC007CxSxPx controller supports direct control of inductive loads such as solenoid valves, electromagnets, electromagnetic brakes, and DC motors. As shown in following figure, connect the load to the Coil+ and Coil- pins on the controller J1. The output voltage is the same as the input voltage of the controller, and the maximum output current is 800mA. In order to reduce the working temperature of the load coil, the controller supports the PWM dynamic voltage adjustment function, and the user can modify the output voltage in real time by command

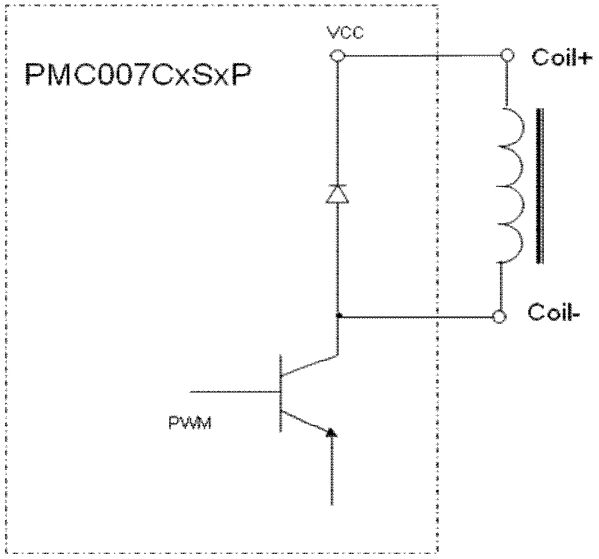


Figure 3-7

3.11 Factory Reset

When the PMC007CxSxPx controller performs a problematic user defined program, or when users accidentally overwrite the controller baud rate, the communication interface may lose response. In this case, if it is still unable to response after repower up, you can use this function to restore factory configuration. Connect the FSET of J1 to GND at least 20ms, and then repower up. The controller is automatically restored to the factory configuration, including the motor parameters, but the user's custom program will be reserved for debugging analysis.

4 Specific application description

4.1 Multi-axis Interpolation

The PMC007 controller can be configured as PVT motion mode. In this mode, the controller uses PUSIROBOT unique cubic spline interpolation optimization algorithm. In the same time coordinate, control the position and speed of the multi axis accurately, so that the end mechanism can realize the trajectory of the line , arc and complex curve, this is an important feature in robot arm or multi-axis application as shown in Figure 4-2 below.

The operation method of PVT mode is described in detail in 5.11.

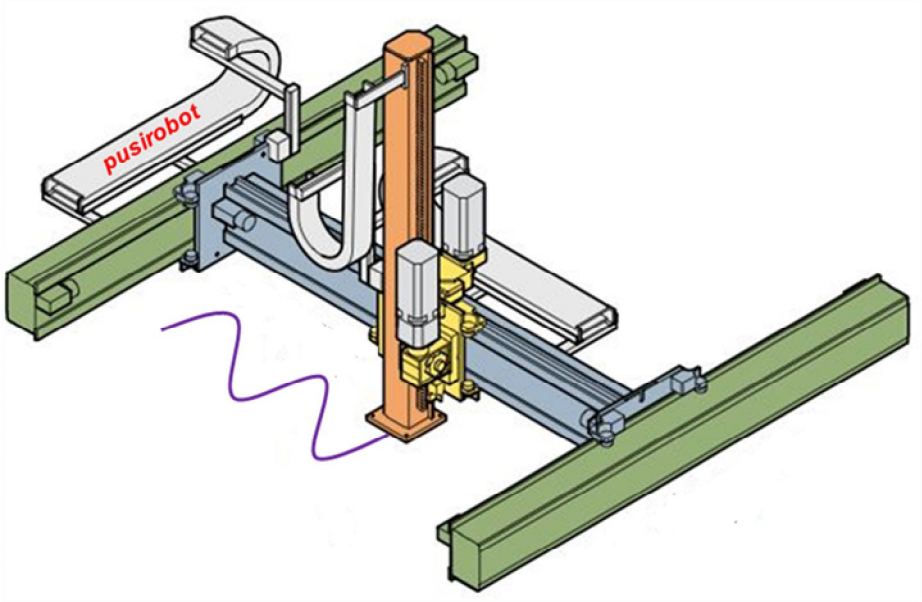


Figure 4-2

In the application of literary creation, such as floating ball matrix, clock matrix, umbrella matrix, etc. In order to present a holistic view, it is necessary to synchronize hundreds of axes. The PMC007 controller uses a special optimization algorithm, which can greatly reduce the bus load and improve the real-time response.

4.2 Drive mode

PMC007CxSxPx supports both high-speed mode and low-speed mute mode. The controller is in high-speed mode by default. When the user is sensitive to noise and works at low speed, low speed mute mode can be selected. In high speed mode, the power saving switch can be turned on to reduce the heat generation of the motor. The parameters of driver mode are set by debugging tool for the first time, and the setting method is described in 7.3.3.2.

5 CANopen communication

5.1 CANopen introduction

CAL provides all network management services and message protocols, but it does not define the contents of objects or the kind of objects being communicated(it defines how, not what). This is where CANopen enters the picture. CANopen is built on top of CAL, using a subset of CAL services and communication protocols, and providing an implementation of a distributed control system. It does this in such a way that nodes can range from simple to complex in their functionality without compromising the interoperability between the nodes in the network.

Central concept in CANopen is the Device Object Dictionary(OD), a concept used in other fieldbus systems as well(Profibus, Interbus-S). CANopen communication through the object dictionary (OD) can access all the parameters of the driver. Note that the Object Dictionary of CAL, it is an implementation aspect of CANopen. The

Object Dictionary which PMC007 is supported is shown in Appendix 1.

CANopen communication model defines the following messages (communication objects);

Abbreviation	Full name	Description
SDO	Service Data Object	Used for non-time critical data, such as parameters.
PDO	Process Data Object	Used to transfer time critical data(Setting values, Control word, status information, etc.)
SYNC	Synchronization Message	Used to synchronize CAN nodes.
EMCY	Emergency Message	Used to transport alarm event of a driver.
NMT	Network Management	Used for CANopen network management.
Heartbeat	Error Control Protocol	Used for monitoring the life status of all nodes.

5.2 CAN frame structure

CAN Data is transmitted between the host (controller) and the bus node through the data frame. The following table is the structure of the data frame.

Header	Arbitration domain		Control domain	Data domain	Check field	Response domain	Tail frame
	COB-ID(communication object identifier)	RTR(remote request)					
1bit	11 or 29 bits	1bit	6bits	0~8byte	16bits	2bits	7bits

When the transceiver is realized through the software, except to the COB-ID and data domain, the other bits in the frame structure are completed by the hardware of CAN transceiver controller. So Users only need to limit the COB-ID and data domain.

Note: This drive uses standard frame format, COB-ID is 11 bits. Remote frame is not supported temporarily.

The distribution of COB-ID is as follows:

Function Code				NODE ID (Node address)						
10	9	8	7	6	5	4	3	2	1	0

The controller parameters are accessed by the SDO read and write objects. For the driver, whose status information is needed to report to the main station in real time can be achieved by configuring the PDO.

5.3 CAN communication configuration

PMC007 factory default settings: node ID is 5, the baud rate is 125Kbit/s. The user can modify the settings by supporting the CANOPEN master debugging tool.

5.3.1 Node ID

Object name	Node ID
SDO ID	0x2002
Object type	U8,rw
Range	1-127
Storage type	ROM
Default value	5

5.3.2 Baud rate

Object name	Baud rate
SDO ID	0x2003
Object type	U8,rw
Range	0,1,2,3,4,5,6,7,8
Storage type	ROM
Default value	5

Relationship between each index and the baud rate is as follows:

0: 20Kbit/s
1: 25Kbit/s
2: 50Kbit/s
3: 100Kbit/s
4: 125Kbit/s
5: 250Kbit/s
6: 500Kbit/s
7: 800Kbit/s
8: 1000Kbit/s

5.3.3 Group ID

Object name	Group ID
SDO ID	0x2006
Object type	U8,rw
Range	1-127
Storage type	ROM
Default value	0

In a CANOpen network, if the PVT motion synchronous start and stop of two or more nodes is needed, this object should be configured.

5.4 System information acquisition

5.4.1 Device node name

Object name	Device node name
SDO ID	0x1008
Object type	string,ro
Range	-

Storage type	ROM
Default value	-

5.4.2 Hardware version

Object name	Hardware version
SDO ID	0x1009
Object type	string,ro
Range	-
Storage type	ROM
Default value	-

5.4.3 Software version

Object name	Software version
SDO ID	0x100A
Object type	string,ro
Range	-
Storage type	ROM
Default value	-

5.4.4 System control

Object name	System control
SDO ID	0x2007
Object type	U8,ro
Range	1, 2, 3
Storage type	RAM
Default value	-

System control values are defined as follows:

1: Jump to bootloader

2: Save Object Dictionary parameters

3: Reset factory settings

Note: the Storage type in the Object Dictionary which is ROM parameter is temporarily stored in memory after written by SDO. If you need to keep it permanently, you need to perform power down save operation for the Object Dictionary parameter.

5.5 Motor control parameters

5.5.1 Error status

Object name	Driving state
SDO ID	0x6000
Object type	U8,rw
Range	bit
Storage type	RAM
Default value	0

Driver state are defined as follows:

Bit0: TSD, over temperature shutdown

Bit1: AERR, coil A error

Bit2: BERR, coil B error

Bit3: AOC, A over current

Bit4: BOC, B over current

Bit5: UVLO, low voltage fault

Write 1 to the corresponding bit, the corresponding error state will be cleared.

5.5.2 Controller status

Object name	Controller status
SDO ID	0x6001
Object type	U8,rw
Range	bit
Storage type	RAM
Default value	0

Controller status is defined as follows:

Bit0: External stop 1

Bit1: External stop 2

Bit2: Stall state

Bit3: busy state

Bit4: External stop 3

Bit5: The FIFO of PVT Mode 3 is empty

Bit6: FIFO Lower bound of PVT Mode 3

Bit7: FIFO upper limit of PVT mode 3

Each bit except for busy state can be written 1 to clear the response state.

5.5.3 Rotation direction

Object name	Rotation direction
SDO ID	0x6002
Object type	U8,rw
Range	0,1
Storage type	RAM
Default value	0

The value of the rotation direction is defined as follows:

0: forward

1: backward

5.5.4 Maximum speed

Object name	Maximum speed (pps)
SDO ID	0x6003

Object type	U32,rw
Range	-200000 ~ +200000
Storage type	RAM
Default value	0

Note: the speed is a signed variable. Positive represents that the direction is 1, and negative represents that the direction is 0. So in the displacement mode it is recommended to set the speed firstly , and then set the direction.

5.5.5 Relative displacement command

Object name	Relative displacement command
SDO ID	0x6004
Object type	U32,rw
Range	0x0-0xFFFFFFFF
Storage type	RAM
Default value	0

Write step number, then the controller will control the motor rotate a given number of steps which is calculated based on the current microstepping settings at the setting direction, speed and acceleration.

When the controller is in the busy state, the step command will be ignored. When the error state and the other bits in the controller status are valid, you need to clear up before you start the step command.

In closed-loop mode, the input unit is 1/4 of the encoder resolution. For example, If CPR=500, the motor would rotate a circle when input 2000.

In the closed loop mode of absolute encoder, the input unit is the same counting unit of encoder, for example, the accuracy is 12 bits, so the motor rotates one turn when the input is 4096.

5.5.6 Absolute displacement command

Object name	Absolute displacement command
SDO ID	0x601c
Object name	U32,rw
Range	Incremental or single cycle absolute value encoder:-2^31 ~ (2^31-1) Multi-circle absolute value encoder:2^(r+11)~(2^(r+11)1), r is the encoder precision, for example, r=12 represents 12-bit encoder.
Storage type	RAM
Default value	0

Absolute displacement command gives the target position, then the controller will automatically calculate the direction and the required step number, and control the motor rotate to the target position at the setting speed and acceleration.

In the open loop mode, the number of steps is calculated in the current subdivision setting.

In closed-loop mode, the input unit is 1/4 of the encoder resolution. For example, If CPR=500, the motor would rotate a circle when input 2000.

In the closed loop mode of absolute encoder, the input unit is the same counting unit of encoder, for example, the accuracy is 12 bits, so the motor rotates one turn when the input is 4096.

5.5.7 Stop stepping command

Object name	Stop stepping command
SDO ID	0x6020
Object type	U8,rw
Range	0
Storage type	RAM
Default value	0

The command immediately terminates the motor running, regardless of the current mode is location mode or speed mode.

5.5.8 Operation mode

Object name	Operation mode
SDO ID	0x6005
Object type	U8,rw
Range	0,1, 2
Storage type	RAM
Default value	0

The value of the motor operation mode is defined as follows:

0: Position mode

1: Speed mode

2: PVT mode

3: Encoder following mode (special version of firmware)

4: PP (Profile Position) mode (including analog positioning)

5: PV(Profile Velocity) mode

When the operation mode is switched from the speed mode to the position mode, the motor will stopping at the setting deceleration.

5.5.9 Start speed

Object name	Start speed(Unit: pps)
SDO ID	0x6006
Object type	U16,rw

Range	0-0xFFFF
Storage type	ROM
Default value	400

5.5.10 Stop speed

Object name	Stop speed(Unit: pps)
SDO ID	0x6007
Object type	U16,rw
Range	0-0xFFFF
Storage type	ROM
Default value	0

5.5.11 Acceleration coefficient

Object name	Acceleration coefficient
SDO ID	0x6008
Object type	U8,rw
Range	0-8
Storage type	ROM
Default value	0

5.5.12 Deceleration coefficient

Object name	Deceleration coefficient
SDO ID	0x6009
Object type	U8,rw
Range	0-8
Storage type	ROM
Default value	0

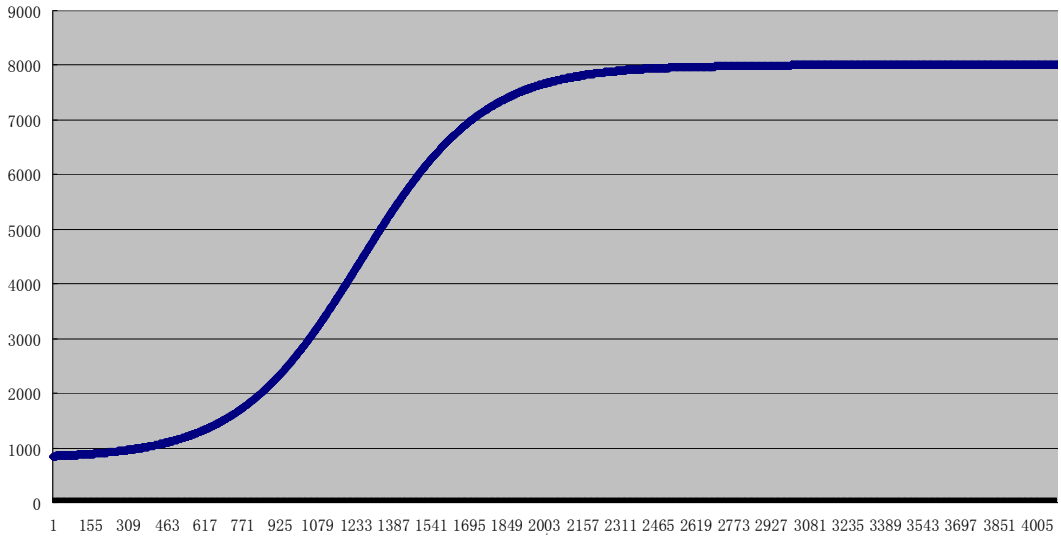


Figure 5-1

The PMC007CxSxPx controller uses S curve acceleration and deceleration. As shown

in Figure 5-1, Start speed, stop speed, acceleration and deceleration can be configured separately. There are a total of 8 gears for acceleration and deceleration. The relationship between each gear and the corresponding acceleration value is shown in the following table.

Gear	Acceleration and deceleration value (PPS^2)
0	Acceleration and deceleration cannot be enable
1	77440
2	48410
3	27170
4	21510
5	14080
6	10460
7	6915
8	5210

5.5.13 Microstepping

Object name	Microstepping
SDO ID	0x600A
Object type	U16,rw
Range	0,2,4,8,16,32,64,128,256
Storage type	ROM
Default value	0

5.5.14 Maximum phase current

Object name	Maximum phase current
SDO ID	0x600B
Object type	U16,rw
Range	0-6000
Storage type	ROM
Default value	0

5.5.15 Motor position

Object name	Motor position
SDO ID	0x600C
Object type	S32,rw
Range	-2147483648-2147483647
Storage type	RAM
Default value	0

When a stepping order is issued, the controller automatically records the

current position which is represented by an signed integer according to the given number of steps. A positive value indicates clockwise rotation, and a negative value indicates a counterclockwise rotation.

In open-loop mode, current position value is calculated by the number of steps, so when users need to change the microstepping, shall read the position information firstly and then change the microstepping, in order to avoid the position conversion error. In closed-loop mode, the 1/4 of the encoder resolution is the unit.

In open-loop mode, when the controller power down, the position information is automatically cleared.

In incremental closed-loop mode, when the controller is powered off, the position where the power is saved can be selected, and the position value of the last power loss will be loaded to the object the next time the power is turned on again.

In the single cycle absolute value closed loop mode, the position information is automatically cleared when the controller is powered off.

In a multi-turn absolute closed-loop mode, the controller reads the encoder position to this object in real time after the controller is powered on.

5.5.16 Calibration zero (absolute value encoder closed loop)

Object Name	Calibration zero
SDO ID	0x6034
SDO ID	S32,rw
Object type	Single cycle absolute value encoder: -2^31 ~ (2^31-1) Multi-loop absolute value encoder: -2^(r+11) ~ (2^(r+11))-1), r is encoder accuracy, such as r ≥ 12 for 12-bit encoder
Range	RAM
Storage type	0

The closed-loop controller of absolute encoder supports this object. When the user writes a value of motor position (0x600C object), the value of calibration zero will be calculated automatically, and the motor position read by the user = encoder position will be calibrated zero.

5.5.17 Encoder position (absolute value encoder closed loop)

Object Name	Calibration zero
SDO ID	0x6035
SDO ID	S32,rw
Object type	Single cycle absolute value encoder: -2^31 ~ (2^31-1) Multi-loop absolute value encoder: -2^(r+11) ~ (2^(r+11))-1), r is encoder accuracy, such as r ≥ 12 for 12-bit encoder
Range	RAM

Storage type	0

The controller reads the actual position value of the encoder.

5.5.18 Current reduction

Object name	Current reduction coefficient
SDO ID	0x600D
Object type	U8,rw
Range	0-3
Storage type	ROM
Default value	0

5.5.19 Motor enable

Object name	Motor enable
SDO ID	0x600E
Object type	U8,rw
Range	0,1
Storage type	RAM
Default value	1

The value of the motor is defined as follows:

0: Offline

1: Motor enable

After setting the offline, controller immediately release the control of motor and the current step command is terminated, phase current is reduced to 0. All subsequent step command issued by host computer cannot be processed, until the user reset motor enable.

5.5.20 Stall set（Open-loop）

Object name	After set stall, whether motor stop or not.
SDO ID	0x601b
Object type	Record
Range	0-1
Storage type	ROM
Default value	0

When set to 1, the power opportunity stops after blocking turn, and the motor does not stop when it is 0.

5.5.21 Stall parameters（Open-loop）

Object name	Set stall detection parameters
SDO ID	0x6017

Object type	U16,rw
Range	bit
Storage type	ROM
Default value	0

The values of detection parameters are defined as follows:

Bit0~6: Blocking threshold, with symbol number;

Bit8~15 : Reservation;

PMC007CxSxPx controller uses the reverse EMF of two-phase winding to realize sensorless blocking detection. Its accuracy is affected by many factors, such as current, subdivision, voltage, motor parameters and so on, especially the motor speed and phase inductance. The range of blocking threshold is usually set between-10 and 10.

5.5.22 Real time Speed（close loop only）

Object name	Read real time motor speed
SDO ID	0x6030
Object type	U16,ro
Range	-300000 ~ +300000
Storage type	ROM
Default value	0

Real-time speed is a signed variable, positive time represents direction 1, negative time represents direction 0.

5.6 External emergency stop

The PMC007CxSxPx controller provides a special limit switch input port EXT1, which can be used for emergency stop or home position search function.

When the emergency stop is enabled, if the corresponding input pin detects effective trigger edge, controller immediately lock the motor and stop responding to any step command. User can read the status of controller, and check which one input pin trigger the emergency stop. The controller will continue to respond to the new step command, only after the user clears the corresponding status bit.

Object name	External emergency stop
SDO ID	0x600F
Object type	Record
Storage type	ROM
Number of parameters	2

Subindex 0x01：External emergency stop enable

Object type	U8,rw
Range	bit

Default value	0

It is represented by 1bit that each external emergency stop enable. 0 indicates the prohibition, and 1 indicates enabling. Its definition is as follows:

bit0: External emergency stop 1 enable settings

bit1: External emergency stop 2 enable settings

bit4: External emergency stop 3 enable settings

Subindex 0x02: The trigger mode of external emergency stop

Object type	U8,rw
Range	bit
Default value	0

The trigger mode of each external emergency stop is represented by 1bit. 0 indicates falling edge trigger, and 1 indicates rising edge trigger. Its definition is as follows:

bit0: The trigger mode of external emergency stop 1

bit1: The trigger mode of external emergency stop 2

bit4: The trigger mode of external emergency stop 3

Subindex 0x03: Sensor type

Object type	U8,rw
Range	0-1
Default value	0

Its definition is as follows:

0: when the trigger mode is configured as the rising edge, the controller is configured as the internal pull-down resistance,When configured as a falling edge, the controller is configured with an internal pull-up resistor; typically used for an NPN type of sensor;

1: When the trigger mode is configured as the rising edge, the controller is configured as the internal pull-up resistance, when the trigger mode is configured as the falling edge, the controller is configured with the internal pull-down resistance, typically used for an PNP type of sensor;

The trigger delay of external emergency stop can be modified by 0x601A object. The controller delays the setting time after the detected edge signal, and then detects whether its level state is correct or not, then triggers the motor to stop urgently, otherwise the motor will continue to rotate.

Object name	EXT1/EXT2/EXT3 stabilize delay (ms)
SDO ID	0x601a
Object type	Record
Range	0~200
Storage type	ROM
Default value	100

5.7 General IO port

The PMC007CxSxPx controller provides 7 general purpose IO (GPIO) ports, 2 external emergency stop input (EXT) ports and 2 encoder input (ENC) ports.

5.7.1 General IO port set

Object name	General IO port set
SDO ID	0x6011
Object type	Record
Storage type	ROM
Number of parameters	2

Subindex 0x01: the direction of IO port

Object type	U16,rw
Range	bit
Default value	0

The direction of each IO port is represented by 1bit. 0 represents input, and 1 represents output. The meaning of each bit is as follow:

Bit0: GPIO1

Bit1: GPIO2

Bit2: GPIO3

Bit3: GPIO4

Bit4: GPIO5

Bit5: GPIO6

Bit6: GPIO7

Bit7: EXT1

Bit8: EXT2

Bit9: EXT3/ENC1

Bit10: ENC2

Bit11: GPIO8

Among them, the direction of emergency stop input port and encoder input port is fixed as input port, which cannot be configured.

Note: GPIO0~GPIO7 does not lead to the controller interface. It is only used for off-line programming.

Subindex 0x02: IO port configuration

Object type	U32,rw
Range	0-0x3fffff
Default value	0

Each port is configured by 2 bits. If the IO port is configured as a input port, the meaning of the value is as follows:

0: FLOATING

1: IPU

2: IPD

3: AIN

If the IO port is configured as a output port, the meaning of the value is as follows:

0: OD

1: PP

The definition of the IO port configuration is defined as follows:

Bit1-0: GPIO1

Bit3-2: GPIO2

Bit5-4: GPIO3

Bit7-6: GPIO4

Bit9-8: GPIO5

Bit11-10: GPIO6

Bit13-12: GPIO7

Bit15-14: EXT1

Bit17-16: EXT2

Bit19-18: EXT3/ENC1

Bit21-20: ENC2

Bit23-22: GPIO8

5.7.2 General IO port value

Object name	General IO port value
SDO ID	0x6012
Object type	U16,rw
Range	bit
Storage type	RAM
Default value	0

The value of each IO port is represented by 1bit, 0 indicates a high level, 1 indicates a low level, and writing value to the port is not valid for the input port. The meaning of each bit is as follows:

Bit0: GPIO1 value

Bit1: GPIO2 value

Bit2: GPIO3 value

Bit3: GPIO4 value

Bit4: GPIO5 value

Bit5: GPIO6 value

Bit6: GPIO7 value

Bit7: EXT1 value

Bit8: EXT2 value

Bit9: EXT3/ENC1 value

Bit10: ENC2 value

Bit11: GPIO8 value

5.8 Offline programming

5.8.1 Offline programming parameter 1

Object name	Offline programming parameter 1
SDO ID	0x6018
Object type	Record
Storage type	ROM
Number of parameters	2

Subindex 0x01: Number of offline programming command

Object type	U8,rw
Range	0-100
Default value	0

Subindex 0x02: Offline automatic operation enable

Object type	U8,rw
Range	0,1
Default value	0

The values of Offline automatic operation are defined as follows:

0: Disable offline automatic operation

1: Enable offline automatic operation

5.8.2 Offline programming parameter 2

Object name	Offline programming parameter 2
SDO ID	0x6019
Object type	Record
Storage type	RAM
Number of parameters	5

Subindex 0x01: Off-line program pointer

Object type	U8,rw
Range	0-100
Default value	0

Subindex 0x02: offline command

Object type	U32,rw
Range	-
Default value	-

For details about the definition of offline command, please refer to User-defined program section.

Subindex 0x03: Save offline command

Object type	U8,rw
Range	0,1
Default value	0

If 1 is written into, all offline command will be saved.

Subindex 0x04: GPIO mask

Object type	U16,rw
Range	bit
Default value	0

Subindex 0x05: run command

Object type	U16,rw
Range	0,1
Default value	0

If 1 is written into, run the command which the program pointer is pointing to.

5.9 Closed-loop control

PMC007CxSxPx supports 200-2000CPR incremental photoelectric encoder and uses PID to realize closed loop control. The following is a detailed description of the closed-loop parameters.

5.9.1 Encoder resolution

Object name	Encoder resolution
SDO ID	0x6021
Object type	U16,rw
Range	Incremental encoder closed loop:200,400,500,600,800,1000,1200,1600,2000 Absolute encoder closed loop:12
Storage type	ROM
Default value	1000,12

Note: After changing the encoder resolution, the power of controller must be reinstalled.

5.9.2 KP parameter

Object name	KP parameter
SDO ID	0x6023
Object type	U8,rw
Range	1-255

Storage type	ROM
Default value	48

This parameter affects the transient response characteristic of the system.

5.9.3 KI parameter

Object name	KI parameter
SDO ID	0x6024
Object type	U8,rw
Range	1-255
Storage type	ROM
Default value	8

This parameter affects the cumulative error characteristics of the system.

5.9.4 KD parameter

Object name	KD parameter
SDO ID	0x6025
Object type	U8,rw
Range	1-255
Storage type	ROM
Default value	8

This parameter affects the transient response characteristic of the system.

5.9.5 Pre-filtering parameter

Object name	Pre-filtering parameter
SDO ID	0x6026
Object type	U8,rw
Range	1-128
Storage type	ROM
Default value	8

This parameter affects the speed characteristics of the system. When speed or microstepping is high, it is recommended to use larger parameter values.

5.9.6 Post-filtering parameter

Object name	Post-filtering parameter
SDO ID	0x6027
Object type	U16,rw
Range	1-255
Storage type	ROM
Default value	8

This parameter is reserved for the time being.

5.9.7 Stall length parameter

Object name	Stall length parameter
SDO ID	0x6028
Object type	U16,rw
Range	1-255
Storage type	ROM
Default value	64

The threshold value of stall is judged by the current subdivision unit.

5.9.8 Torque ring enable

Object name	Torque ring enable
SDO ID	0x6029
Object type	U8,rw
Range	0-1
Storage type	ROM
Default value	1

When the moment ring is not enabled, the PID parameter does not take effect, and the controller works in the position loop mode.

5.9.9 Autosave when power is off enable

Object name	Autosave when power is off enable
SDO ID	0x602A
Object type	U8,rw
Range	0-1
Storage type	ROM
Default value	0

If this parameter is enabled, the controller would automatically detect the power off and write the current position into the EEPROM.

5.10 Synchronous position Motion mode

Object name	SP motion control register
SDO ID	0x601D
Object type	Record
Storage type	RAM
Number of parameters	2

In SP motion mode, firstly set the absolute position and speed of the specified node, then make multiple axes move at the same time by SP synchronous boot instruction.

5.10.1 SP speed

Subindex 0x01:SP speed

Object type	S32,rw
Range	-2147483648-2147483647
Default value	0

5.10.2 SP position

Subindex 0x02:SP position

Object type	S32,rw
Range	-2147483648-2147483647
Default value	0

5.11 PVT motion mode

Object name	PVT motion object
SDO ID	0x6010
Object type	Record
Storage type	RAM
Number of parameters	19

PMC007 supports three PVT control modes, each of which is suitable for different application scenarios.

Mode 1 is a single motion mode. When the controller executes the PVT sequence data written by the host computer, the PVT motion is finished.

Mode 2 is a circular motion mode. PVT motion will be end after dedicated cycle times which is assigned by host.

Mode 3 is a FIFO control mode. The host computer writes the PVT sequence to the controller continuously, the controller takes out PVT data to perform PVT motion.

In addition, the PMC007 support group ID setting, which is used to synchronize two or more nodes start and stop PVT running in the same network. For details about the use process of PVT motion pattern, please refer to the script example of the PUSICAN tool.

5.11.1 PVT Control

Subindex 0x01:PVT control operation

Object type	U8,rw
Range	0-3
Default value	0

0：Stop PVT motion.

1：Start PVT motion.

2：Write the PVT position, speed, and time object data into the queue;

3：Clear all PVT data in the queue.

5.11.2 PVT operation mode

Subindex 0x02:PVT operation mode

Object type	U8,rw
Range	0-2
Default value	0

0：PVT mode 1;

1：PVT mode 2;

2：PVT mode 3;

5.11.3 Max PVT points

Subindex 0x03: Max PVT points

Object type	U16,rw
Range	0-1000
Default value	0

5.11.4 PVT pointer

Subindex 0x04:Current PVT pointer

Object type	U16,r
Range	0-1000
Default value	0

5.11.5 PVT mode 1 parameter

1. PVT mode 1 start index
   Subindex 0x05:PVT mode 1 start index

Object type	U16,rw
Range	0-1000
Default value	0

2. PVT mode 1 end index
3. Suubindex 0x06: PVT mode 1 end index

Object type	U16,rw
Range	0-1000
Default value	0

5.11.6 PVT mode 2 parameter

1. PVT mode 2 start index at the acceleration stage

Subindex 0x07:PVT mode 2 start index at the acceleration stage

Object type	U16,rw
Range	0-1000
Default value	0

2. PVT mode 2 end index at the acceleration stage

Subindex 0x08:PVT mode 2 end index at the acceleration stage

Object type	U16,rw
Range	0-1000
Default value	0

3. PVT mode 2 start index at the cycle stage

Subindex 0x09:PVT mode 2 start index at the cycle stage

Object type	U16,rw
Range	0-1000
Default value	0

4. PVT mode 2 end index at the cycle stage

Subindex 0x0A:PVT mode 2 end index at the cycle stage

Object type	U16,rw
Range	0-1000
Default value	0

5. PVT mode 2 cycle times at the cycle stage

Subindex 0x0B:PVT mode 2 cycle times

Object type	U16,rw
Range	0-65535
Default value	0

6. PVT mode 2 start index at the deceleration stage

Subindex 0x0C:PVT mode 2 start index at the deceleration stage

Object type	U16,rw
Range	0-1000
Default value	0

7. PVT mode 2 end index at the deceleration stage

Subindex 0x0D:PVT mode 2 end index at the deceleration stage

Object type	U16,rw
Range	0-1000
Default value	0

5.11.7 PVT mode 3 parameter

1. PVT mode3 FIFO depth

   Subindex 0x0E:PVT mode3 FIFO depth

Object type	U16,r
Range	0-1000
Default value	0

2. PVT mode3 FIFO lower limit

   Subindex 0x0F:PVT mode3 FIFO lower limit

Object type	U16,rw
Range	0-1000
Default value	0

In PVT mode 3, once the FIFO depth is less than the set value of this object, and the FIFO lower limit of the controller state object would be set.

3. PVT mode 3 FIFO upper limit

   Subindex 0x10:PVT mode 3 FIFO upper limit

Object type	U16,rw
Range	0-1000
Default value	0

In PVT mode 3, once the FIFO depth is more than the set value of this object, and the FIFO upper limit of the controller state object would be set.

5.11.8 PVT position

Subindex 0x11:PVT position

Object type	S32,rw
Range	-2147483648-2147483647
Default value	0

An absolute position which the current PVT point is expected to move.

5.11.9 PVT speed

Subindex 0x12:PVT speed

Object type	S32,rw
Range	-2147483648-2147483647
Default value	0

The current PVT point expected speed of motion, unit PPS.

5.11.10 PVT time

Subindex 0x13:PVT time

Object type	S32,rw

Range	0-2147483647
Default value	0

The time is from the last PVT point to the current PVT point, unit ms.

5.12 PV/PVT Synchronous start and stop

PMC007 can achieve two or more nodes PVT synchronous start/stop in a network by NMT extended instructions of standard CANopen.

Standard NMT format

COB-ID	Byte0	Byte1
0x000	CS	Node-ID

The extended NMT instructions add new definitions to Byte0 and Byte1 without affecting the standard protocol.

Byte0 is defined as follows:

Command	Function
10	Start PV motion
11	Start PVT motion
12	Stop PVT motion

Byte1 is group ID. The corresponding command operation is performed only when the group ID received by the controller matches its own group ID.

5.13 PP motion mode

The Profile Position Mode will be entered if motion mode 4 is chose, In this mode the trapezoid acceleration is adopted, a set of new parameters can be issued from host, after the control bit is set, the driver will smoothly switch from last set of parameters to the new set of parameters. The description of control bit is refer to section 5.13.3, and the description of control object is refer to section 5.13.1 and 5.13.2.

5.13.1 PP mode parameter 1

PP mode parameter 1 can be power down preserved.

Object Name	PP mode parameter 1
SDO ID	0x602d
Object type	Record
Storage type	ROM
Number of Parameters	4

5.13.1.1 Acceleration

Sub-index 0x01: acceleration, pps/s

Object type	U32,rw
Range	>150

Default	32000

5.13.1.2 Deceleration

Sub-index 0x02: deceleration, pps/s

Object type	U32,rw
Range	>150
Default	32000

5.13.1.3 Start speed

Sub-index 0x03: start speed

Object type	U32,rw
Range	>150
Default	600

5.13.1.4 Stop speed

Sub-index 0x04: stop speed

Object type	U32,rw
Range	>150
Default	600

5.13.2 PP mode parameter 2

PP mode parameter 2 is stored in RAM and cannot be power down preserved.

Object Name	PP mode parameter 2
SDO ID	0x602e
Object type	Record
Storage type	ROM
Number of parameters	4

5.13.2.1 Control word

Sbu-index 0x01: Control word

Object type	U16,rw
Rang	0-0xFFFF
Default	0

The function description for Control word object (602e,1):

- Bit 4: Start task. Start the task when bit value switch from 0 to 1.
- Bit 5: The task trigged by Bit4 will be immediately executed if this bit is set 1. The task will be executed after last task completed if this bit is set 0.

☐ Bit 6: The target position (602e,4) is absolute position when this bit is set 0, the target position is relative position if this bit is set 1.

☐ Bit 8 (Halt): This bit is for PV motion mode, motor will be accelerated to target speed in preferred slope if this bit change from 1 to 0. Motor will be decelerated to zero if this bit change from 0 to 1.

☐ Bit 9: Motor will change speed after the first target point arrived if this bit is set 1.

5.13.2.2 Status word

Sub-index 0x02: status word

Object type	U16,rw
Range	0-0xFFFF
Default	0

The function of status word(602e,2) is following:

- Bit 10 : This bit will be set 1 after the final target is arrived.
- Bit 12 : This bit is the ACK bit that will be set after receive new target potion.

4.14.2.3 Running Speed

Subindex 0x03:Running Speed

Object type	U32,rw
Range	-300000- -150,150-300000
Default value	32000

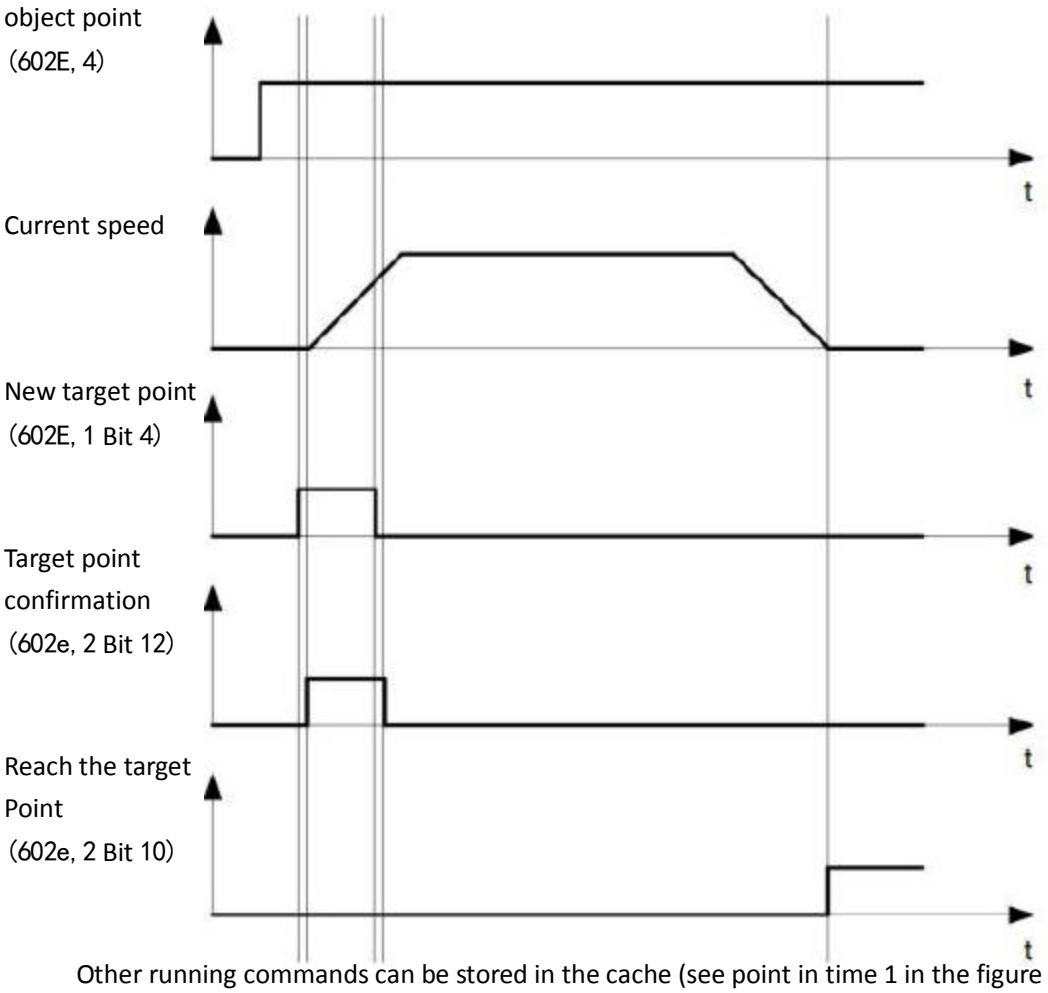
4.14.2.4 Target Location

Subindex 0x04:Target Location

Object type	S32,rw
Range	-2^31~2^31
Default value	0

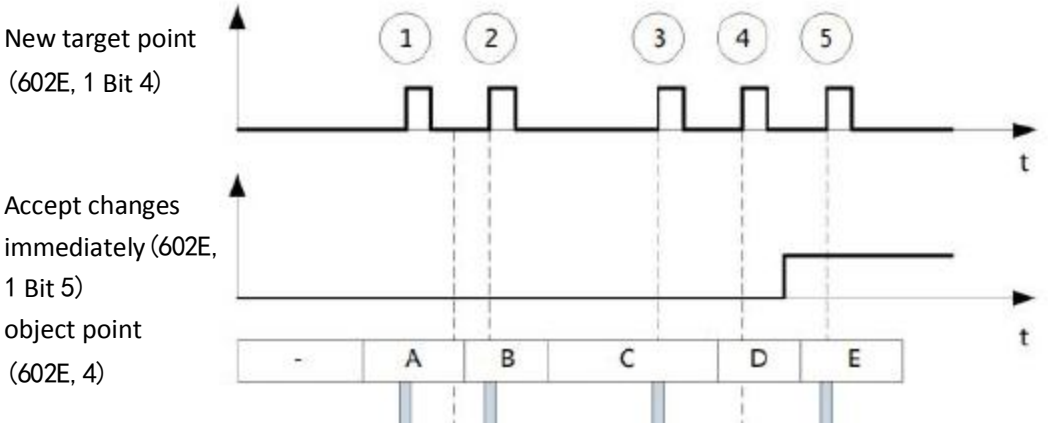
5.13.3 PP mode working timing

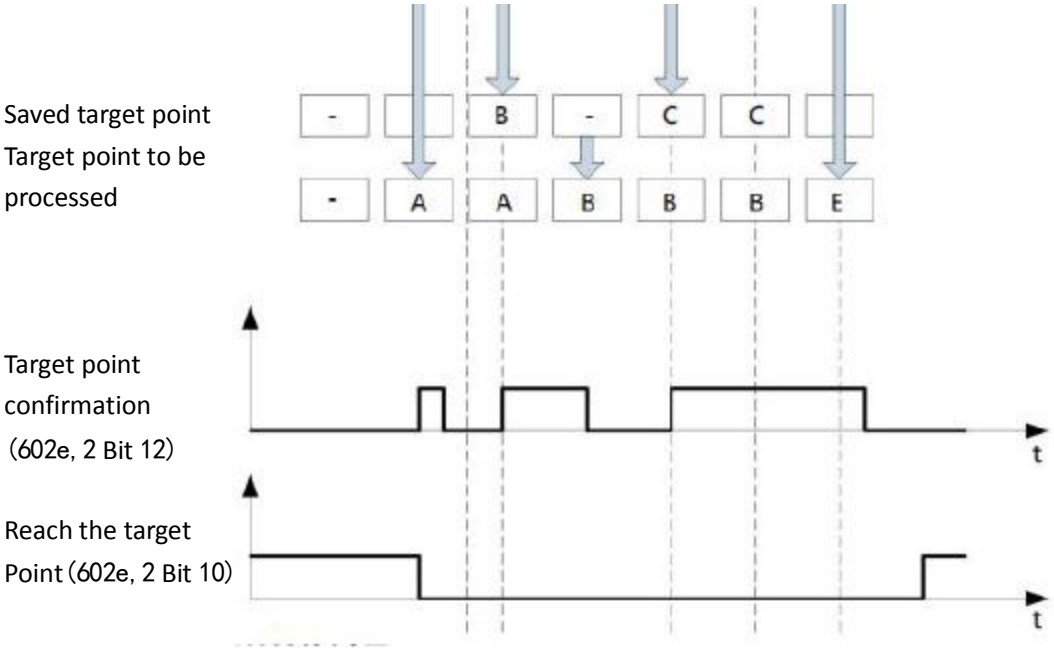
A new target position is set in the target position object (602e,4). Next, the bit 4 in the control word object (602e,1) is set for trigger the operation command. If the target location is valid, the controller will reply through the bit 12 in the object status word to locate the start of the operation. When the location is reached, the bit 10 in the status word will immediately set to a "1".



Other running commands can be stored in the cache (see point in time 1 in the figure below), and bit 12 in the status word object (602e, 2 sets the target point response) will be set to "0". During the motion to the target position, a second target position can be sent to the controller to prepare for it. At this point, you can reset all parameters, such as velocity, acceleration, deceleration, etc. (point in time 2). If the cache is idle again, the next point in time can enter the queue (point in time 3).

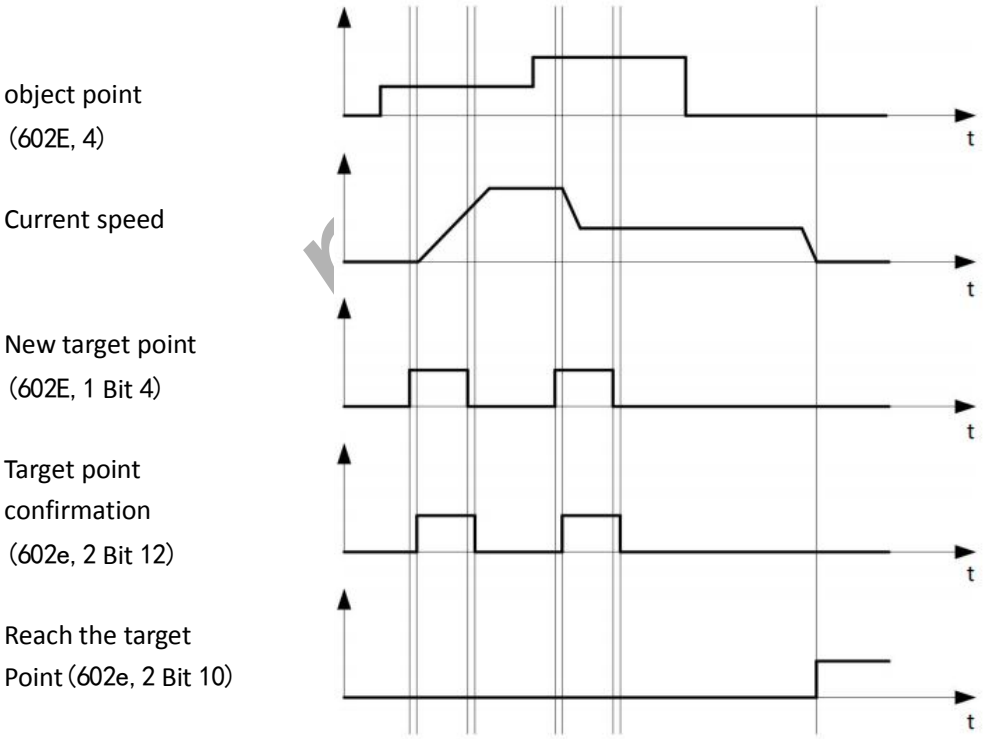
If the cache is full, the new target point will be ignored (point in time 4). If bit 5 in the control word object (602e, 1 bit: "change the target point now") is set, the controller will not use cache when working, and the new running command will be executed directly (point in time 5).





The conversion process of the second target position:

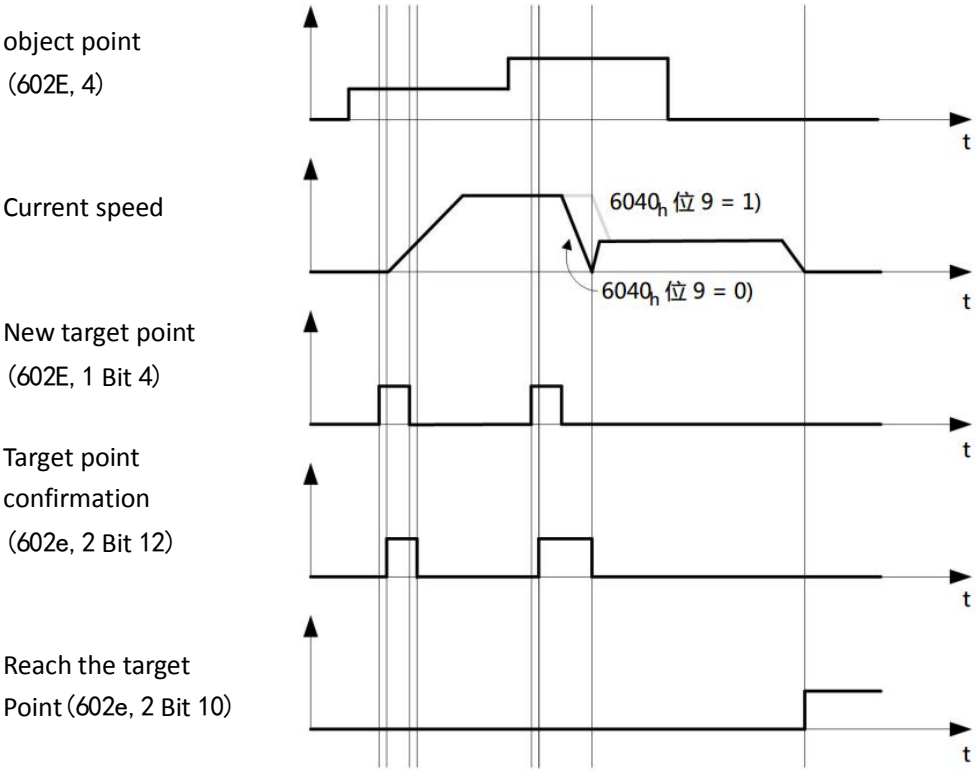
The following figure shows the conversion process of the second target position when moving to the first target position. In this figure, the bit 5 of the control word object (602e, 1) is set to "1" and the new target value will be accepted immediately.



The method of moving to the target position:

If the bit 9 in the control word object (602e,1) is a "0", it will first fully travel to the current target position. In this example, the final speed of the first target position is equal to zero. If bit 9 is set to "1", the final speed will be maintained until the target position is reached, and then the

newly set motion parameters will take effect.



5.14 PV Mode

The working mode is set to 5 into (Profile Velocity Mode) PP mode, which adopts ladder acceleration and deceleration, and shares the starting speed, stop speed, acceleration, deceleration and running speed parameters with PP mode.

When the value of bit 8 (Halt) of the control word changes from "1" to "0", the motor will accelerate to the target speed with a preset starting speed in slope. When the value of the bit changes from "0" to "1", the motor slows down and stops moving. In the process of motion, a new running speed can be sent out, and the controller will smooth over to the newly set speed.

5.15 Analog positioning

PMC007C2 has an analog signal input port, and the internal 12-bit ADC, can be configured into analog positioning mode through software. First configure the analog positioning related parameters, and finally turn on the analog positioning enable. The following quart describes the analog related objects in detail.

Object name	Set analog positioning
SDO ID	0x602f
Object type	Record
Storage type	ROM
Default value	6

5.15.1 Enable analog positioning

Subindex 0x01: enable analog positioning,1 open, 0 closed

Object type	U8, rw
Range	0、1
Default value	0

5.15.2 Analog initial AD code

Subindex 0x02: Analog quantity start AD code, corresponding to the minimum value of the analog position

Object type	U16, rw
Range	0、1
Default value	0

5.15.3 Analog adjustment interval

Subindex 0x03: Analog adjustment interval, Unit ms

The controller checks the analog input value at this time, and if the difference between the AD input value and the last input value is greater than the threshold value, the position will be adjusted once.

Object type	U16, rw
Range	0-65535
Default value	100

5.15.4 Analog regulating trigger value

Subindex 0x04: The analog quantity adjusts the trigger value, and when the difference between the acquired AD code and the last acquired AD code is converted to a position greater than this value, the controller will adjust the position once.

Object type	U16, rw
Range	0-65535
Default value	30

5.15.5 Minimum value of analog position

Subindex 0x05: Minimum value of analog position. Absolute position corresponding to the analog start AD code

Object type	S32, rw
Range	-2^31~2^31
Default value	0

5.15.6 Maximum value of analog position

Subindex 0x06: Analog position minimum.Absolute position corresponding to the analog start AD code

Object type	S32, rw
Range	-2^31~2^31
Default value	64000

5.16 Brake control

PMC007 supports brake control, and the output duty cycle can be adjusted by software, to avoid the serious problem of long-term brake heating.

Object name	Brake control
SDO ID	0x6016
Object type	U8, rw
Range	0~100
Storage type	RAM
Default value	0

5.17 Analogue input

PMC007C2 supports 0~24V voltage analog input, 12-bit ADC.

Object name	Analogue input
SDO ID	0x602b
Object type	U16, rw
Range	0~4095
Storage type	RAM
Default value	0

5.18 Step notification

The PMC007CxSxPx controller can set step notification in position mode or speed mode, that is, when the motor motion reaches a set position in one step, the controller can report step to position notification through TPDO to support two step notification location points.

Object name	Analogue input
SDO ID	0x602C
Object type	Record
Storage type	RAM
Default value	3

Subindex 0x01: step notification status

Object type	U8, rw

Range	bit
Default value	0

Each IO port direction is represented by 1bit, 0 is the input, 1 is the output, the meaning of you is as follows:

Bit0: step notification point 1 is valid;

Bit1: step notification point 2 is valid;

The object can be mapped to the TPDO, object value will be automatically reported to the host computer when the value of the object changes.

Subindex 0x02: step notification position 1 (absolute position) setting

Object type	S32,rw
Range	-2147483647 ~ +2147483647
Default value	0

Subindex 0x03: step notification position 2 (absolute position) setting

Object type	S32,rw
Range	-2147483647 ~ +2147483647
Default value	0

5.19 Power loss behavior

PMC007CxSxPx can detect the power loss of the system and set the corresponding power loss behavior. The following is a detailed description of the power loss behavior settings related objects.

Object name	Power down behavior setting
SDO ID	0x6031
Object type	Record
Storage type	RAM
Default valueNumber of parameters	3

5.19.1 Power-down behavior control word

Sub-index 0x01: Power-down behavior control switch

Object type	U16,rw
range	0-65535
Default value	0

Bit0: detects power loss and goes offline

Bit1: brake lock

The switch is turned on when the corresponding value is 1 and the switch is turned off for 0.

5.19.2 Power-down off-line voltage

Sub-index 0x02: Offline threshold voltage, unit mv

Object type	U16,rw
range	0-65535
Default value	0

When the offline switch is turned on, the motor is offline when the power supply voltage is detected to be below this voltage.

5.19.3 Switching voltage

Sub-index 0x03: Voltage limit of electric lock gate, unit mv

Object type	U16,rw
range	0-65535
Default value	0

When the power-down brake switch is opened, the brake brake is detected when the voltage of the power supply is lower than the voltage.

6 User-defined program

PMC007CxSxPx can be configured into offline mode. In this mode, controller automatically execute custom user code after powered on, the code is compiled and in advance burned to the EEPROM through CQPUSI tool software.

When the PMC007CxSxPx controller works in offline mode, the CAN communication interface is still responsive to the user's online instruction.

The maximum number of instructions that PMC007CxSxPx controller supports for user is 100.

Please refer to the "controller offline programming guide" for details about detailed example of user defined program.

7 Introduction of the Debug Tool

Users can use the CQPUSI tool software Tool Debug to debug command, IO port setting detection, set control parameters of motor, custom programming.

7.1 Installation preparation

Tool software PUSICAN requires CAN adapter (USB2CAN or PCI2CAN) support. Currently, the tool software has supported a wide variety of common USB2CAN adapter on the market. If you need to support other types of adapter, please contact with sales staffs.

7.2 Software installation

7.2.1 Driver installation

Install the driver of adapter, please follow the instructions on the user manual of adapter.

7.2.2 Tool software installation

Debug tool PUSICAN is green free installation software. After download, extract it into a special folder. Double-click the pusican.exe, then the software is running.

7.3 Software instructions

7.3.1 Preparation for using

Connect the PMC007CxSxPx and CAN adapter to the computer by way shown in Figure 3-22. Then power PMC007CxSxPx up. After power up normally, the LED lights will be flashing at 2.5Hz frequency.

7.3.2 Main interface

Double click on the desktop PUSICAN shortcut icon to enter the main interface. As shown in Figure 7-1:

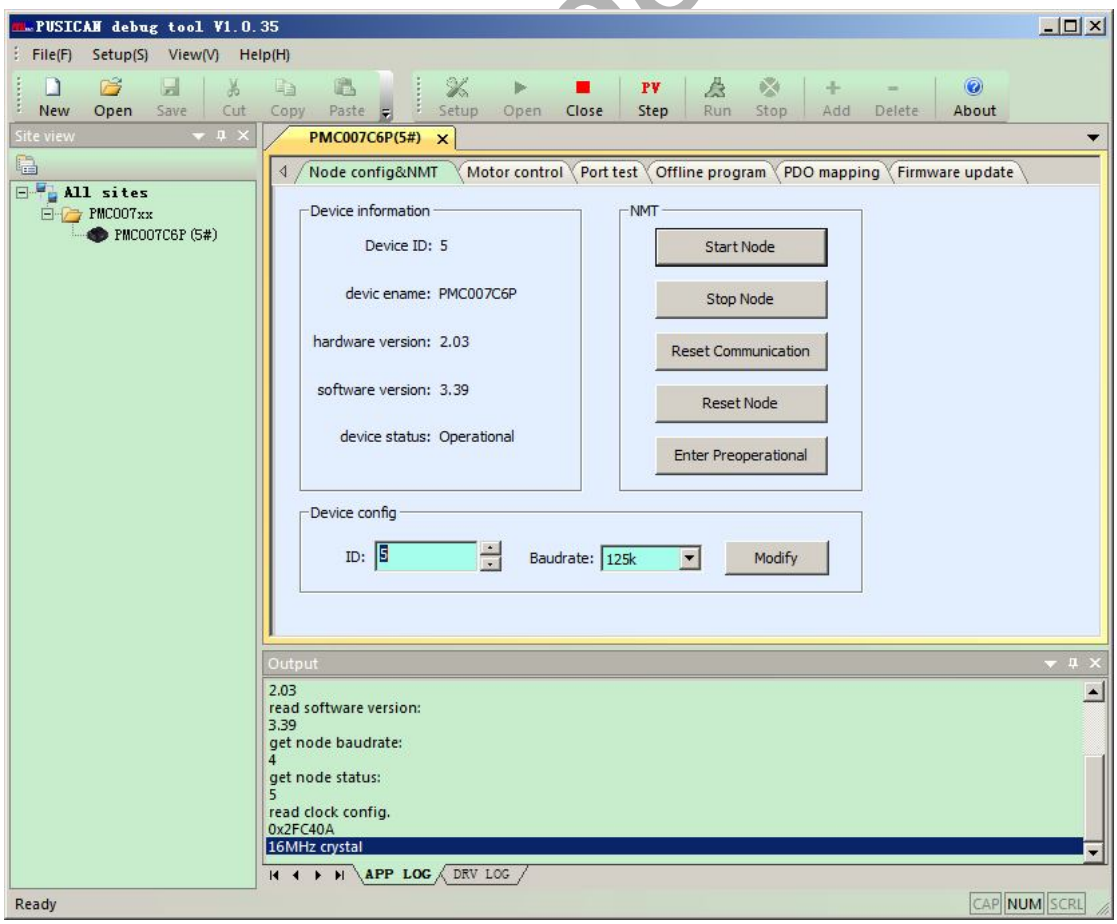


Figure 7-1

In the main interface, click "Settings" icon to select the adapter and baud rate. PMC007CxSxPx controller factory default baud rate is 125kHz. Click "open" icon, debugging tools will boot the adapter scanning site, and all online activity equipment are listed in the left side tree list. Double-click on operation site and the right workspace will display the control interface of the device.

Start node: Make PMC007CxSxPx into the operational state;

Stop node: Make PMC007CxSxPx into stopped state. The node will not respond to any SDO command;

Reset communication: The communication parameters take effect immediately by this operation, after SDO modifies the communication parameters;

Reset node: Notify the node reenter power on reset process;

Pre operation state: In this state, the node waits for the network command of the main station, receives the configuration request of the main station, so it can receive and send all messages except PDO.

7.3.3 Motor motion control interface

7.3.3.1 motor control

In the main interface, click "motor drive settings" into the motor motion control interface. As shown in Figure 7-2:

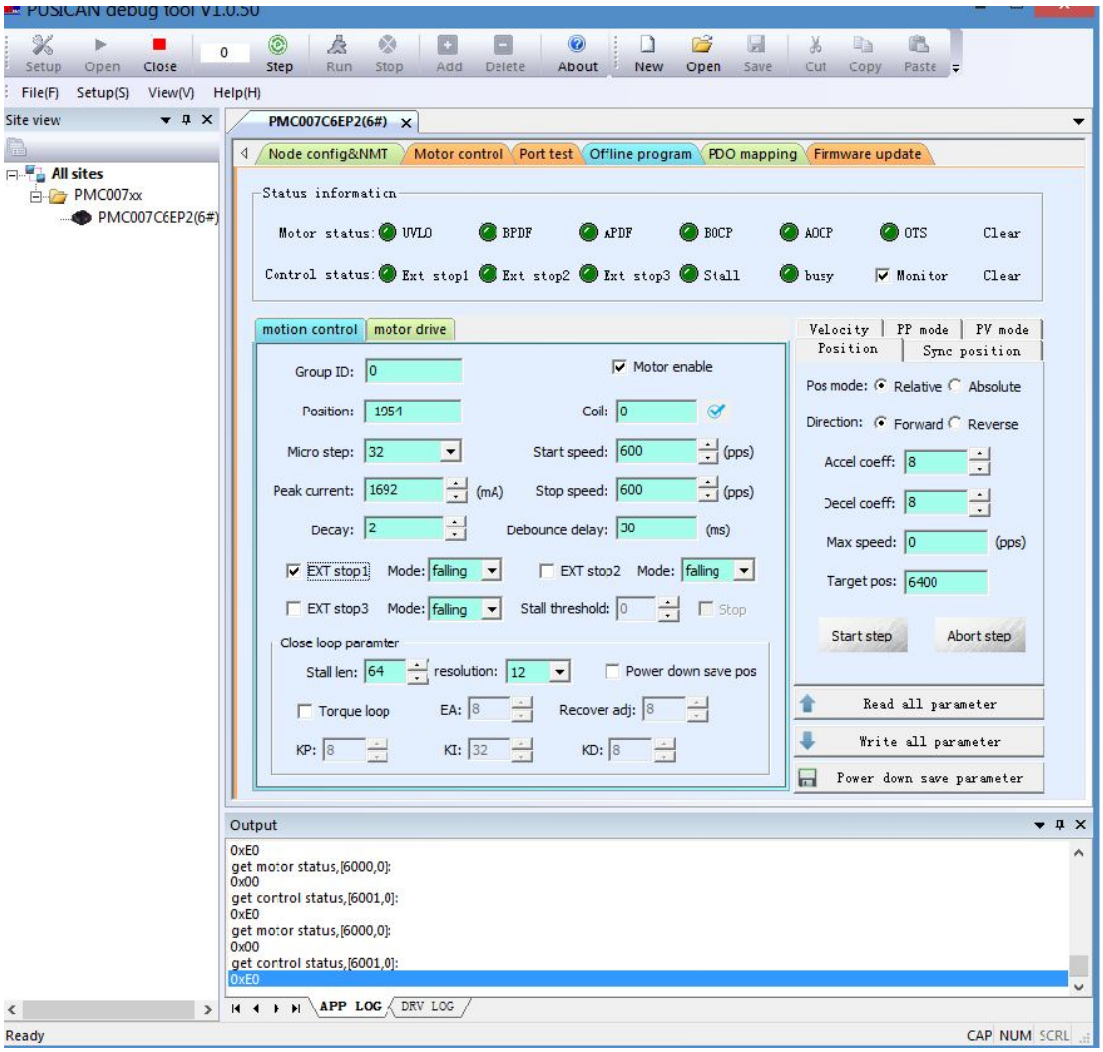


Figure 7-2

Before modifying the parameters, it is recommended to first click on "read all the parameters", update the current display value from the node. After set the control parameters, click "write all parameters" to write the setting parameters to the controller and make it effect. Click "power down to save all parameters" to write the parameter to the FLASH of the device in order to save permanently.

When an error occurs in the motor status, it must be cleared before motor can be started again.

7.3.3.2 Drive mode

Select the Motor driver property page under Motor driver Settings, as shown in figure 7-3.

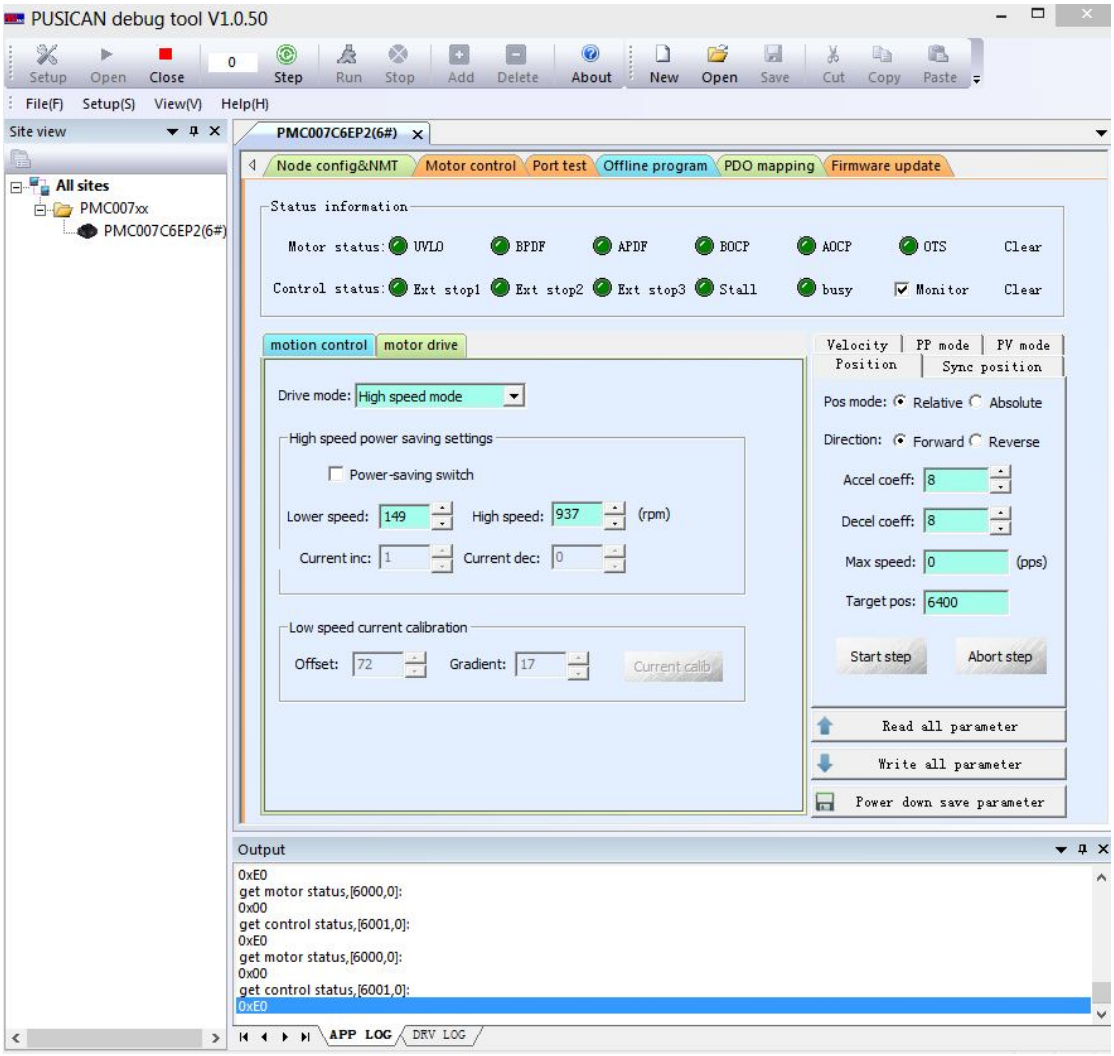


Figure 7-3

7.3.4 Port configure

In the main interface, click on "port test" into the port configure interface, As shown in Figure 7-.

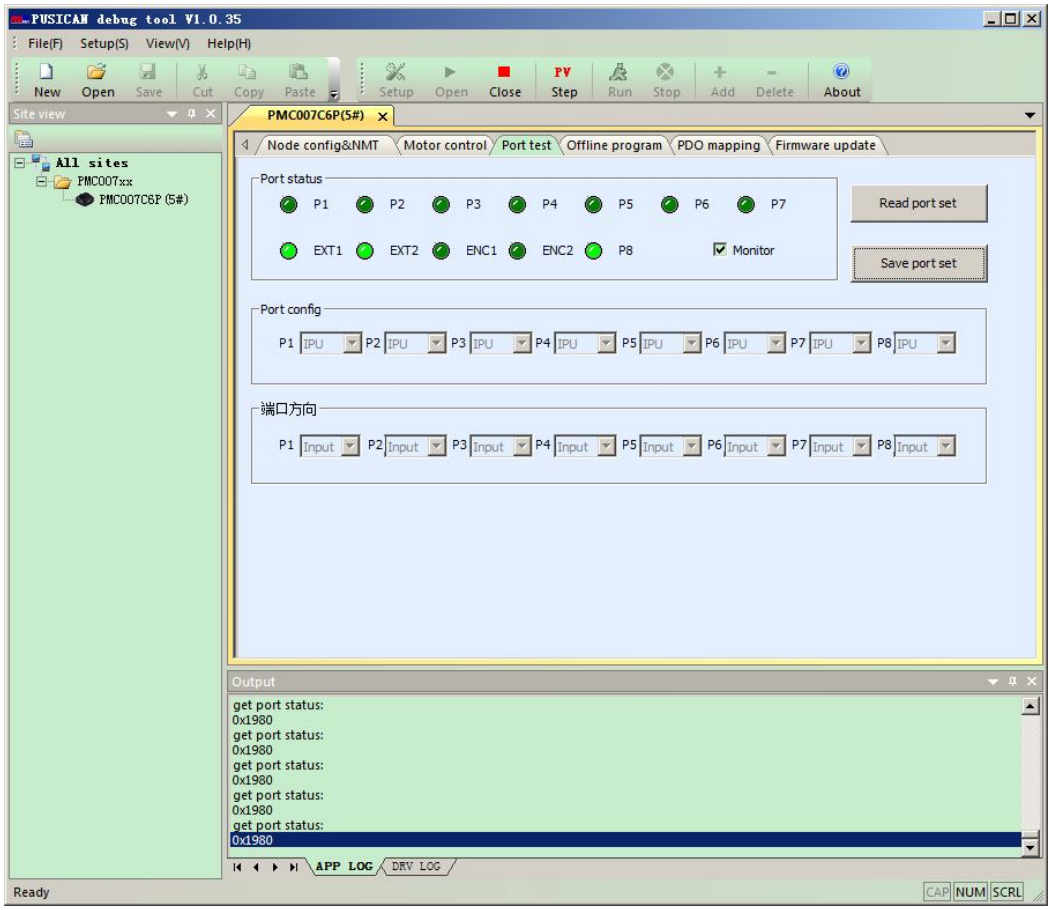


Figure 7-4

After entering the interface, the software will monitor the port state by default. Turn off the port monitor, and then can set the port direction, and change the port configuration.

7.3.5 Offline programming interface

In the main interface, click on "offline programming" into the Offline programming interface, As shown in 7-5.

After entering the interface, the software will send command to close offline automatic operation, click on “read command from Buffer”, then read all offline programming command of the device, and display them on the interface.

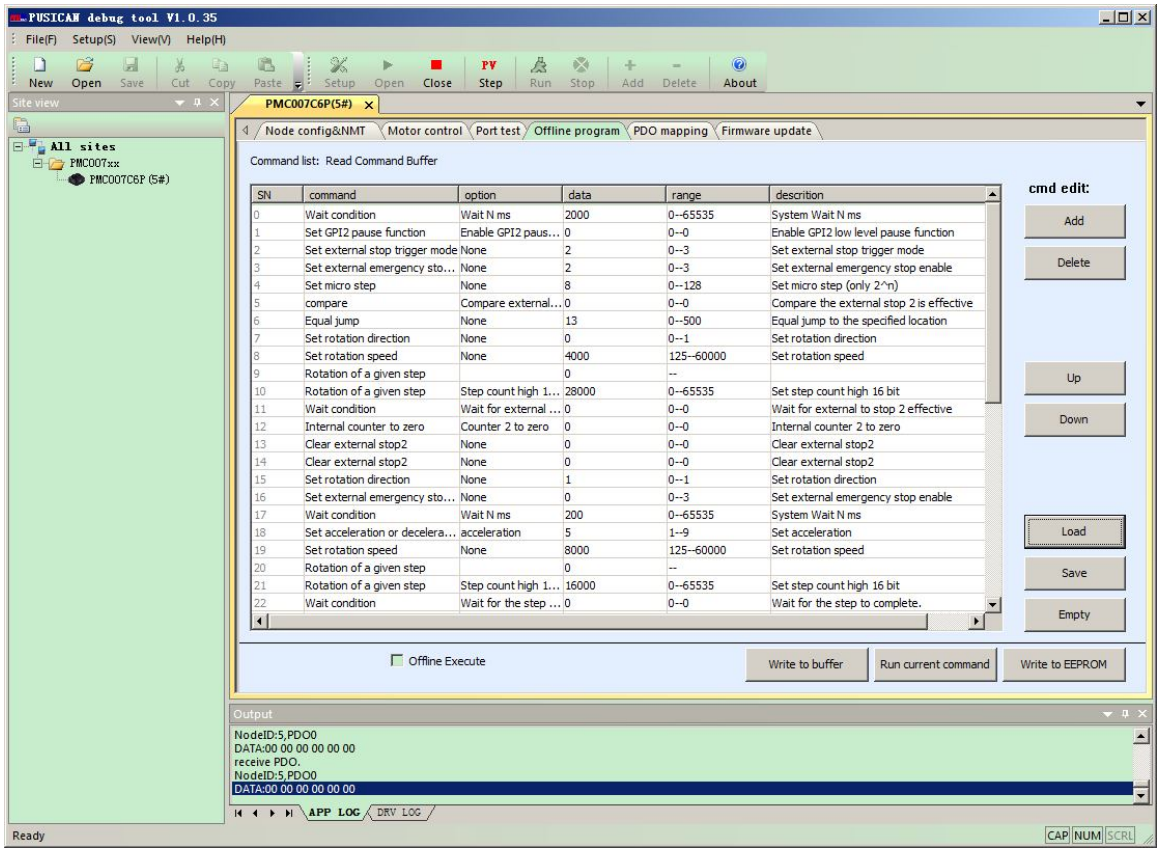


Figure 7-5

Users can complete the edit of offline command through the "add", "delete", "move", "down" button on the right according to the actual demand. Commands can be saved as a local disk file, and the offline command file on the local disk can also be loaded into interface to display.

Once editing is completed when users need to do online debugging, first press the "write cmd to the buffer" button to download the program to on-chip memory of PMC007xx Controller, and then press the "try run current command" button, and then device will run the current selected command. After confirmation, press the "Write to Flash" button to burn all program to non-volatile Flash memory.

If you select the "offline exec", PMC007CxSxPx controller will automatically run the offline program. Controller will read offline program from Flash and automatically run next time power is on.

7.3.6 PDO mapping

In the main interface, click "PDO map" into this interface. after entering the interface, the software will automatically read the current mapping object from the device and display them in the interface, as shown in Figure 7-6.

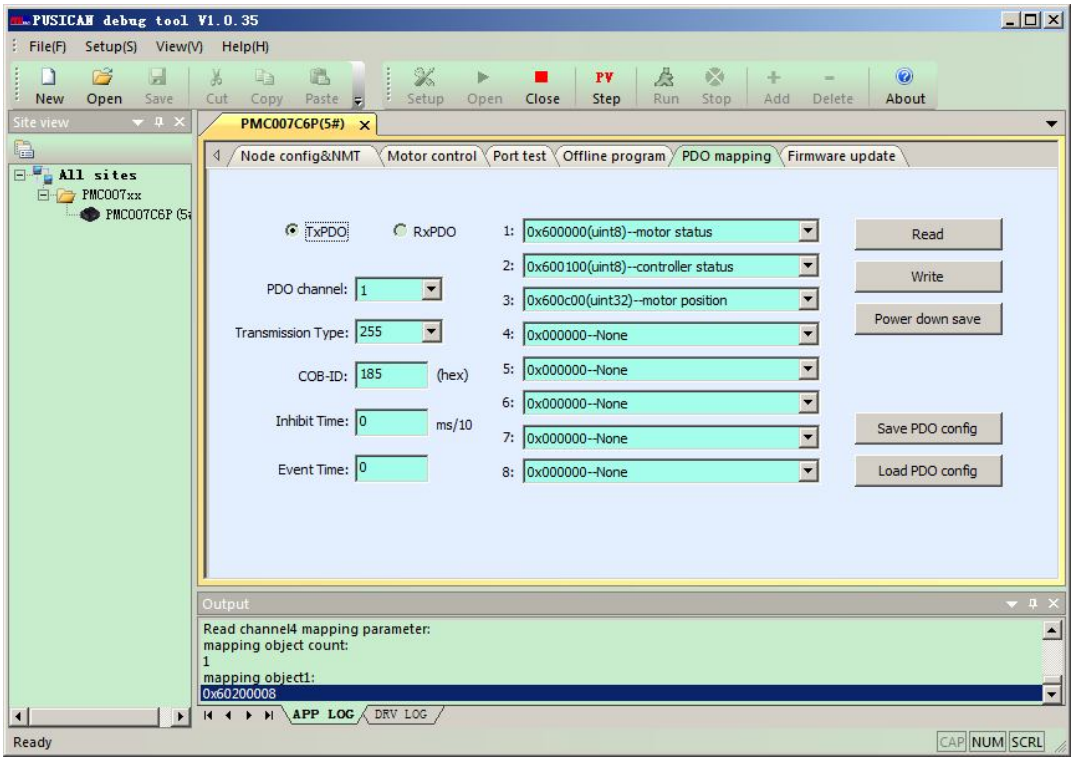


Figure 7-6

PMC007CxSxPx supports sending and receiving PDO of 4 channels, and each channel can map up to 8 objects.

For more details about related parameters configuration, please refer to 2.1 chapter of Appendix 2.

7.3.7 Firmware upgrade

Firmware upgrade of PMC007CxSxPx can be carried out in bootloader mode through the CAN bus. In "firmware upgrade" interface, click "into the bootloader/ application", PMC007CxSxPx will enter bootloader mode. The node ID and baud rate are set in the application model.

After entering the bootloader mode, LED lights will be double flash. In the "application path" column select the upgrade file, click the "upgrade" button to start upgrading, as shown in Figure 7-7. After firmware upgrade is completed, tool will prompt a dialog box, then you can click on "into the bootloader/ application" (or repower on the controller), and then the controller will switch to the normal application mode.

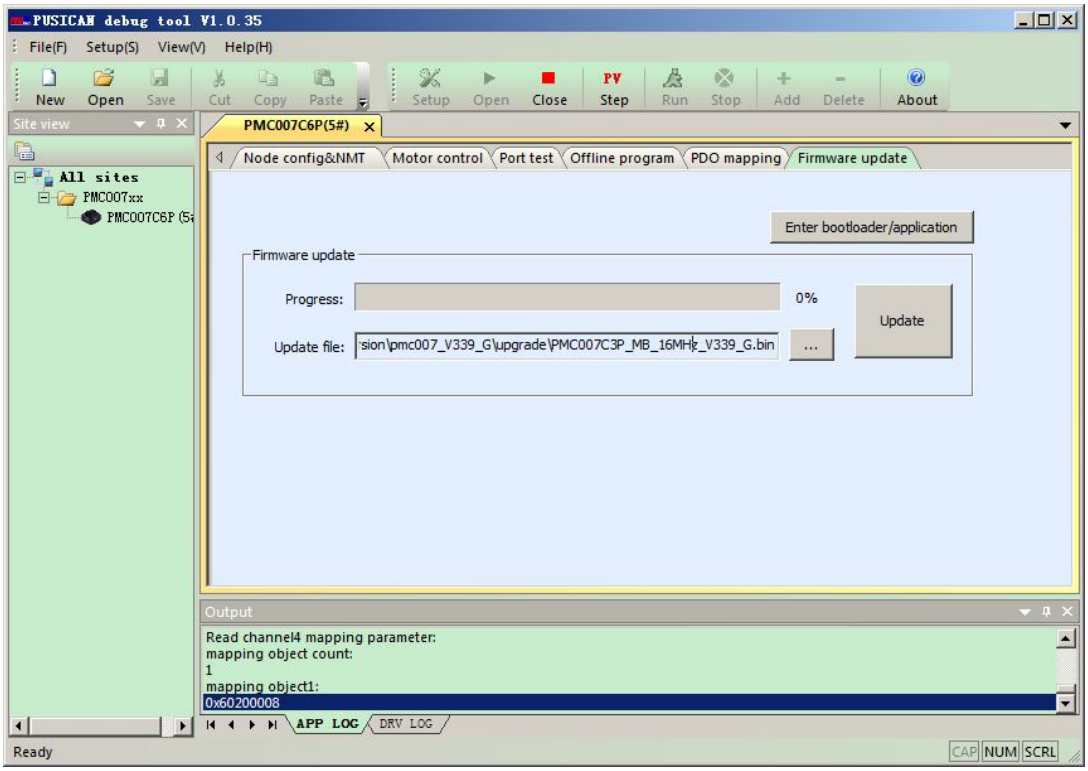


Figure 7-7

7.3.8 Support graphic programming

PUSICAN debugging tool software supports graphic programming. User can add a flow item through the interface, set the related motion parameters, then a simple motion would be completed.

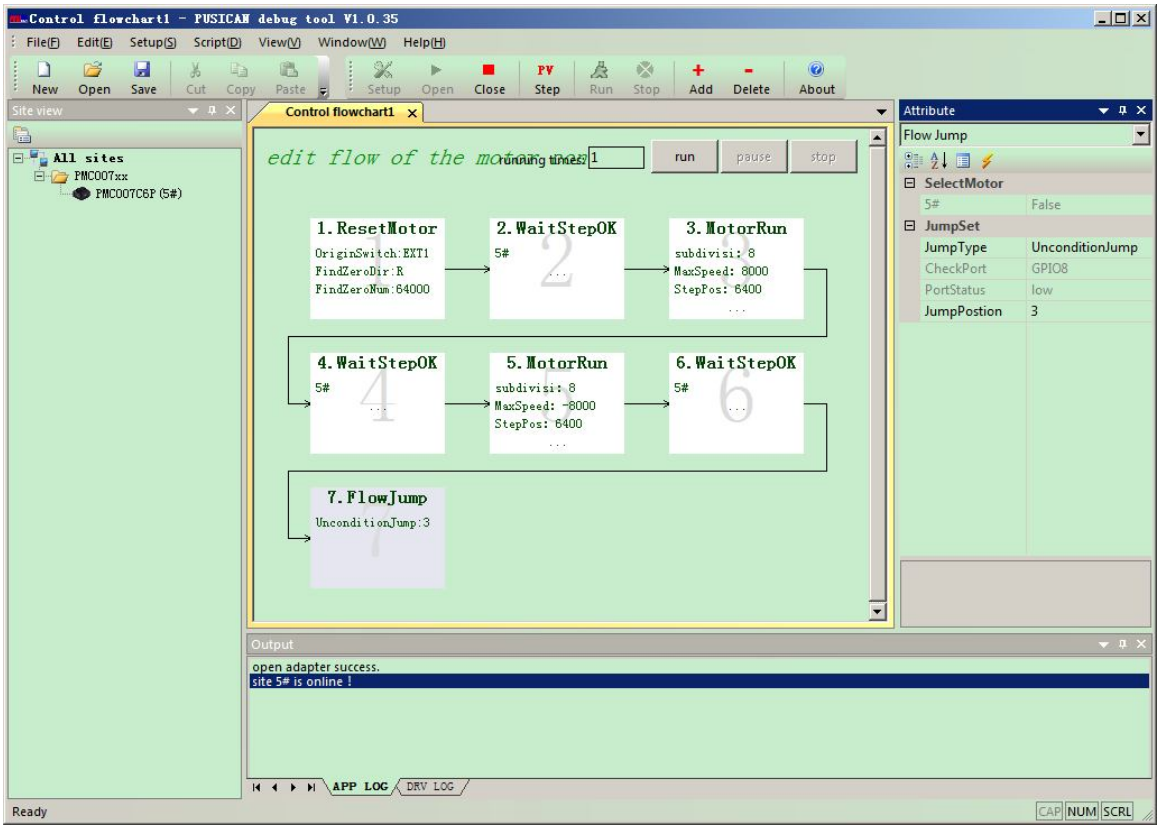


Figure 7-8

7.3.9 Support scripting language

PUSICAN debugging tool software supports LUA scripting language, and has built SDO CANOPEN operating function which user can call directly in the script program. User can create or open a script file through click the "new" and "open" icon on the top left. Once the script was written completely, you can control program execution through click the "run", "stop" icon on the top right. As shown in Figure 7-9.

The syntax of the LUA scripting language is similar to C language. In application scenarios without special requirements of UI interface, the user can complete the complex control loop task with the powerful function of LUA script easily, and no need to develop the CANOPEN master control program in the host computer.

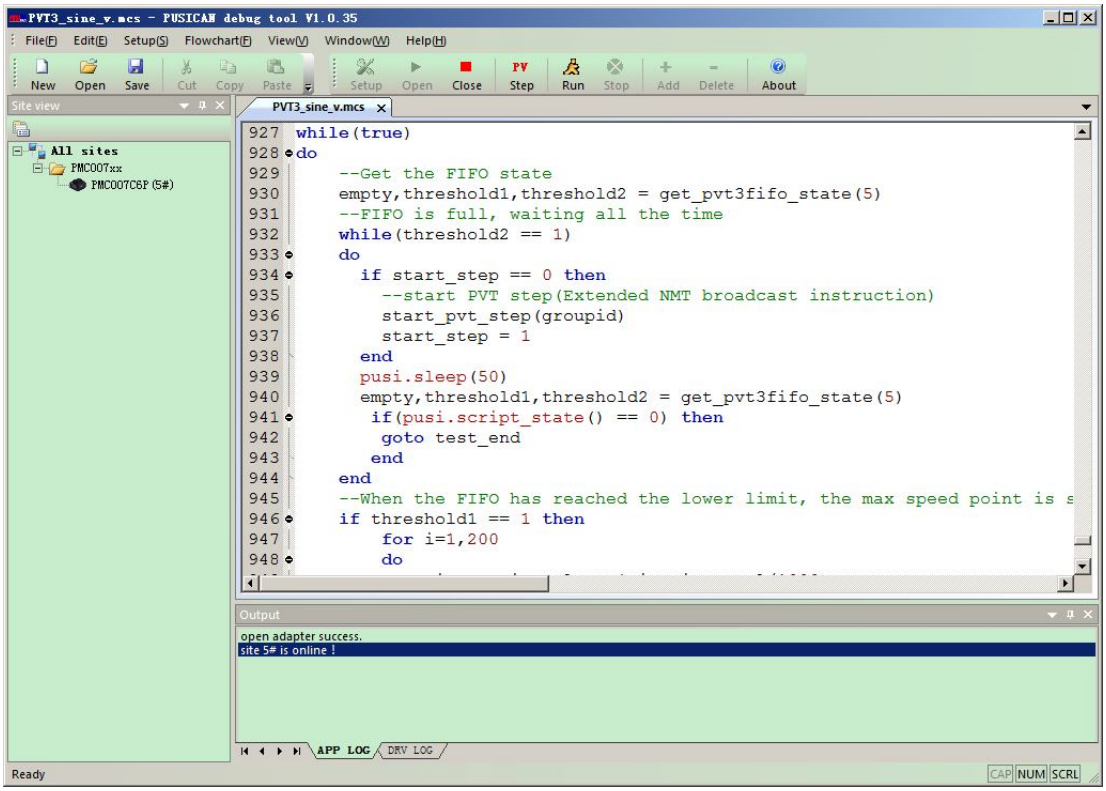
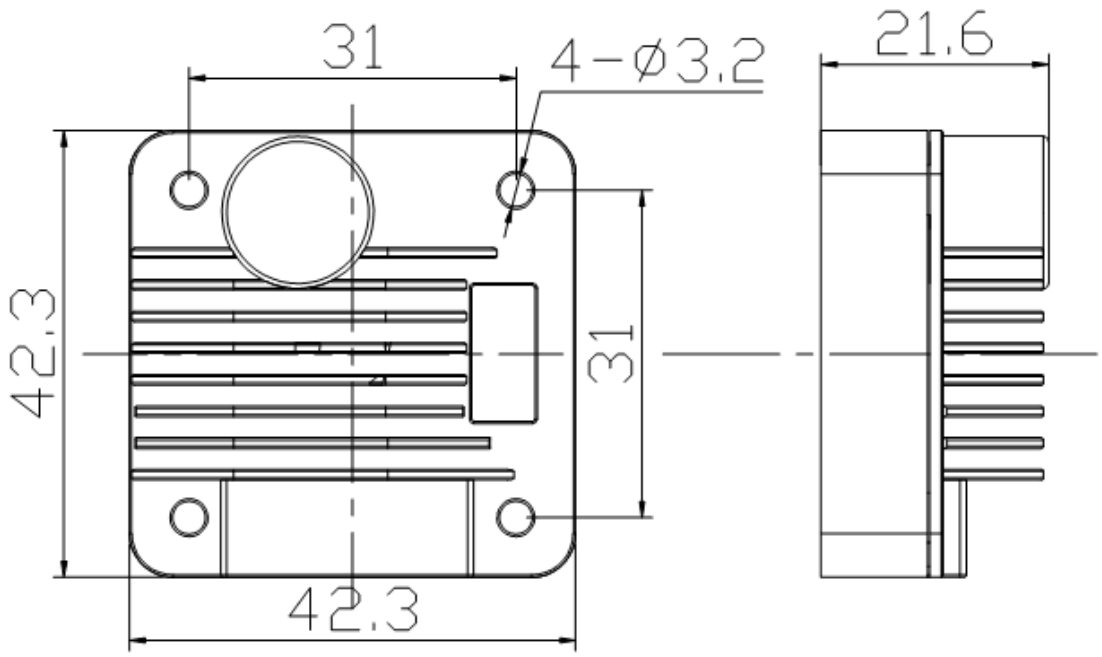


Figure 7-9

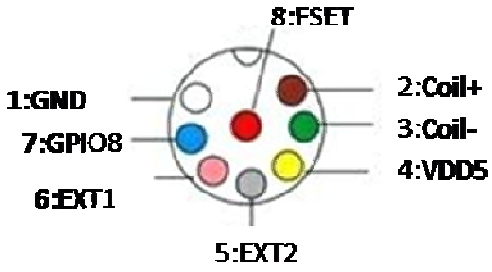
8 Electrical Characteristics

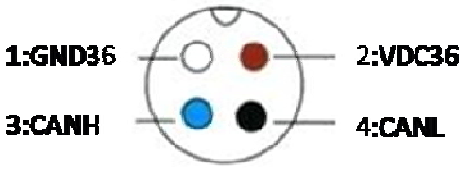
Parameter	Condition	Min	Typical	Max	Unit
Supply Power Voltage	Normal 25℃	12	24	48	V
Operation Temperature	24V DC	-20(M, I) -40(S)		75	℃
IO maximum current	source/sink current	0	10	20	mA
Output current	Normal 25℃	0.4	4	6	A
IO low Voltage	12V DC	-0.5		1.0	V
IO High Voltage	12V DC	3.0		5.5	V

9 Dimensions



For PMC007CxSxPx IP64 Integrated motor with M12 connectors, the pinout is following:





10 Appendix 1 PMC007xx Object dictionary table

Index	Sub index	Object type	Name	Type	Attr.	PDO	Storage type
1000h	--	VAR	Device type	UINT32	RO	NO	ROM
1001h	--	VAR	Error registor	UINT8	RO	Optional	RAM
1002h	--	VAR	manufacturer status register	UINT32	RO	Optional	RAM
1003h	--	ARRAY	pre-defined error field	--	--	--	RAM
	0h		number of errors	UINT8		NO	
	1h-7h		standard error field			Optional	
1005h	--	VAR	COB-ID SYNC	UINT32	RW	NO	ROM
1006h	--	VAR	communication cycle period	UINT32	RW	NO	ROM
1007h	--	VAR	synchronous window length	UINT32	RW	Optional	ROM
1008h	--	VAR	manufacturer device name	Visible String	con st	NO	ROM
1009h	--	VAR	manufacturer hardware version	Visible String	con st	NO	ROM
100ah	--	VAR	manufacturer software version	Visible String	con st	NO	ROM
1014h	--	VAR	COB-ID Emergency message	UINT32	RO	NO	ROM
1015h	--	VAR	Inhibit Time EMCY	UINT16	RW	NO	ROM
1016h	--	ARRAY	Consumer Heartbeat Time	--	--	--	ROM
	0h		number entries	UINT8	RO	NO	
	1h-3h		Consumer Heartbeat Time	UINT32	RW	NO	
1017h	--	VAR	Producer Heartbeat Time	UINT16	RW	NO	ROM
1018h	--	RECORD	Identity Object	--	--	--	ROM
	0h		number of entries	UINT8	RO	NO	
	1h		Vendor ID	UINT32	RO	NO	
	2h		Product code	UINT32	RO	NO	
	3h		Revision number	UINT32	RO	NO	
	4h		Serial number	UINT32	RO	NO	
1200h	--	RECORD	Server SDO parameter	--	--	--	ROM
	0h		number of entries	UINT8	RO	NO	
	1h		COB-ID Client->Server (rx)	UINT32	RO	NO	
	2h		COB-ID Server ->	UINT32	RO	NO	

			Client (tx)				
	3h		Node-ID of the SDO client	UINT32	RW	NO	
1280h	--	RECORD	Client SDO parameter	--	--	--	RAM
	0h		number of entries	UINT8	RO	NO	
	1h		COB-ID Client->Server (tx)	UINT32	RW	NO	
	2h		COB-ID Server -> Client (rx)	UINT32	RW	NO	
	3h		Node-ID of the SDO server	UINT32	RW	NO	
1400h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1401h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1402h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1403h	--	RECORD	receive PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		compatibility entry	UINT8	RW	NO	

	5h		event timer	UINT16	RW	NO	
1600h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1601h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1602h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1603h	--	RECORD	receive PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the nth application object to be mapped	UINT32	RW	NO	
1800h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1801h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	

	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1802h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1803h	--	RECORD	transmit PDO parameter	--	--	--	ROM
	0h		largest sub-index supported	UINT8	RO	NO	
	1h		COB-ID used by PDO	UINT32	RW	NO	
	2h		transmission type	UINT8	RW	NO	
	3h		inhibit time	UINT16	RW	NO	
	4h		reserved	UINT8	RW	NO	
	5h		event timer	UINT16	RW	NO	
1a00h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
1a01h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
1a02h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the	UINT32	RW	NO	

			n-th application object to be mapped				
1a03h	--	RECORD	transmit PDO mapping	--	--	--	ROM
	0h		number of mapped application objects in PDO	UINT8	RO	NO	
	1h-8h		PDO mapping for the n-th application object to be mapped	UINT32	RW	NO	
2002h	--	VAR	Node ID	UINT8	RW	NO	ROM
2003h	--	VAR	Baud rate	UINT8	RW	NO	ROM
2006h	--	VAR	Group ID	UINT8	RW	NO	ROM
2007h	--	VAR	System control	UINT8	RW	NO	RAM
6000h	--	VAR	Error state	UINT8	RW	Optional	RAM
6001h	--	VAR	Controller status	UINT8	RW	Optional	RAM
6002h	--	VAR	Rotation direction	UINT8	RW	NO	RAM
6003h	--	VAR	Max speed	UINT32	RW	NO	RAM
6004h	--	VAR	Step command	UINT32	RW	NO	RAM
6005h	--	VAR	Operation mode	UINT8	RW	NO	RAM
6006h	--	VAR	Start speed	UINT16	RW	NO	ROM
6007h	--	VAR	Stop speed	UINT16	RW	NO	ROM
6008h	--	VAR	Acceleration coefficient	UINT8	RW	NO	ROM
6009h	--	VAR	Deceleration coefficient	UINT8	RW	NO	ROM
600ah	--	VAR	Microstepping	UINT16	RW	NO	ROM
600bh	--	VAR	Max phase current	UINT16	RW	NO	ROM
600ch	--	VAR	Motor position	UINT32	RW	Optional	RAM
600dh	--	VAR	Current attenuation	UINT8	RW	NO	ROM
600eh	--	VAR	Motor enable	UINT8	RW	NO	RAM
600fh	--	RECORD	External emergency stop	UINT8	RW	NO	ROM
	0h		The number of parameters	UINT8	RO	NO	ROM
	1h		External emergency stop enable	UINT8	RW	NO	RAM
	2h		Trigger mode of external emergency stop	UINT8	RW	NO	RAM
6010h	--	RECORD	PVT step	UINT8	RW	NO	ROM
	0h		The number of	UINT8	RO	NO	ROM

			parameters				
	1h		PVT operation control	UINT8	RW	NO	RAM
	2h		PVT mode control	UINT8	RW	NO	RAM
	3h		Max PVT point number	UINT16	RW	NO	RAM
	4h		PVT pointer	UINT16	RO	NO	ROM
	5h		Start index of PVT mode 1	UINT16	RW	NO	RAM
	6h		End index of PVT mode 1	UINT16	RW	NO	RAM
	7h		Start index of PVT model 2 acceleration stage	UINT16	RW	NO	RAM
	8h		End index of PVT model 2 acceleration stage	UINT16	RW	NO	RAM
	9h		Start index of PVT model 2 cycle stage	UINT16	RW	NO	RAM
	Ah		End index of PVT model 2 cycle stage	UINT16	RW	NO	RAM
	Bh		The times of PVT model 2 cycle stage	UINT16	RW	NO	RAM
	Ch		Start index of PVT model 2 deceleration stage	UINT16	RW	NO	RAM
	Dh		End index of PVT model 2 deceleration stage	UINT16	RW	NO	RAM
	Eh		FIFO depth of PVT mode 3	UINT16	RW	NO	RAM
	Fh		FIFO lower limit of PVT mode 3	UINT16	RW	NO	RAM
	10h		FIFO upper limit of PVT mode 3	UINT16	RW	NO	RAM
	11h		PVT position	INT32	RW	NO	RAM
	12h		PVT speed	INT32	RW	NO	RAM
	13h		PVT time	INT32	RW	NO	RAM
6011h	--	RECORD	GPIO parameter	--	--	--	ROM
	0h		The number of GPIO parameters	UINT8	RO	NO	
	1h		GPIO direction	UINT16	RW	NO	
	2h		GPIO configuration	UINT32	RW	NO	
6012h	--	VAR	GPIO value	UINT16	RW	Optional	RAM

6013h	--	RECORD	OCP parameter(ROM)	--	--	--	
	0h		OCPNumber of parameters	UINT8	RO	NO	
	1h		OCT1	UINT8	RW	NO	
	2h		OCD1	UINT8	RW	NO	
6016h	--	VAR	Brake control	UINT8	RW	NO	RAM
6018h	--	RECORD	Off-line programming parameter 1	--	--	--	ROM
	0h		Number of Off-line programming parameter 1	UINT8	RO	NO	
	1h		Total number of offline programming command	UINT8	RW	NO	
	2h		Offline operation enable automatically	UINT8	RW	NO	
6019h	--	RECORD	Off-line programming parameter 2	--	--	--	RAM
	0h		Number of Off-line programming parameter 2	UINT8	RO	NO	
	1h		Off-line parameter pointer	UINT8	RW	NO	
	2h		Off-line command	UINT32	RW	NO	
	3h		Offline command preservation	UINT8	RW	NO	
	4h		Run current command	UINT8	RW	NO	
601ah	--	VAR	Jitter delay of external emergency stop	UINT16	RW	NO	ROM
601bh	--	VAR	Locked-Rotor configuration	UINT8	RW	NO	ROM
601ch	--	VAR	Absolute position step	INT32	RW	NO	RAM
601dh	--	RECORD	PV step	--	--	--	RAM
	0h		The number of PV step parameter	UINT8	RO	NO	
	1h		PV speed	INT32	RW	NO	
	2h		PV position	INT32	RW	NO	
6020h	--	VAR	Termination step	UINT8	RW	NO	RAM
6021h	--	VAR	Encoder CPR	UINT16	RW	NO	ROM
6022h	--	VAR	Position saving value power-down	INT32	RO	NO	ROM
6023h	--	VAR	Closed-loop parameter KP	UINT8	RW	NO	ROM

6024h	--	VAR	Closed-loop parameter KI	UINT8	RW	NO	ROM
6025h	--	VAR	Closed-loop parameter KD	UINT8	RW	NO	ROM
6026h	--	VAR	Closed-loop Pre-filter parameter	INT8	RW	NO	ROM
6027h	--	VAR	Closed-loop post-filter parameter	INT16	RW	NO	ROM
6028h	--	VAR	Closed-loop stall length	INT16	RW	NO	ROM
6029h	--	VAR	Enable closed-loop torque loop	UINT8	RW	NO	ROM
602Ah	--	VAR	Enable saving automatically when power is off	UINT8	RW	NO	ROM
602Bh	--	VAR	Analogue input	UINT16	RW	Optional	RAM
602Ch	--	RECORD	Step notification	--	--	--	RAM
	0h		Number of parameters	UINT8	RO	NO	
	1h		Step notification status	UINT8	RW	Optional	
	2h		Step notification position 1	INT32	RW	Optional	
	3h		Step notification position 2	INT32	RW	Optional	
602Dh	--	RECORD	PP/PVmode parameters 1	--	--	--	ROM
	0h		Number of parameters	UINT8	RO	NO	
	1h		Accelerated speed	UINT32	RW	Optional	
	2h		Dccelerated speed	UINT32	RW	Optional	
	3h		Initial speed	UINT32	RW	Optional	
	4h		Stop speed	UINT32	RW	Optional	
602Eh	--	RECORD	PP/PVmode parameters 2	--	--	--	RAM
	0h		Number of parameters	UINT8	RO	NO	
	1h		control word	UINT16	RW	Optional	
	2h		status word	UINT16	RW	Optional	
	3h		running speed	INT32	RW	Optional	
	4h		target location	INT32	RW	Optional	
602Fh	--	RECORD	Analog location parameters	--	--	--	ROM
	0h		Number of parameters	UINT8	RO	NO	
	1h		Enable analog positioning	UINT8	RW	Optional	

	2h		Analog initial AD code	UINT16	RW	Optional	
	3h		Analog adjustment interval time	UINT16	RW	Optional	
	4h		Analog regulating trigger value	UINT16	RW	Optional	
	5h		Minimum value of analog position	INT32	RW	Optional	
	6h		Maximum value of analog position	INT32	RW	Optional	
6030h	--	VAR	real-time speed	INT32	RW	Optional	RAM
6031h	--	RECORD	power-down behavior control	--	--	--	ROM
	0h		Number of parameters	UINT8	RO	NO	
	1h		Power-down behavior control word	UINT16	RW	Optional	
	2h		Power off motor enabling threshold	UINT16	RW	Optional	
	3h		Brake lock threshold	UINT16	RW	Optional	
6034h	--	VAR	Calibration zero position	INT32	RW	NO	ROM
6035h	--	VAR	Encoder position	INT32	RW	Optional	RAM

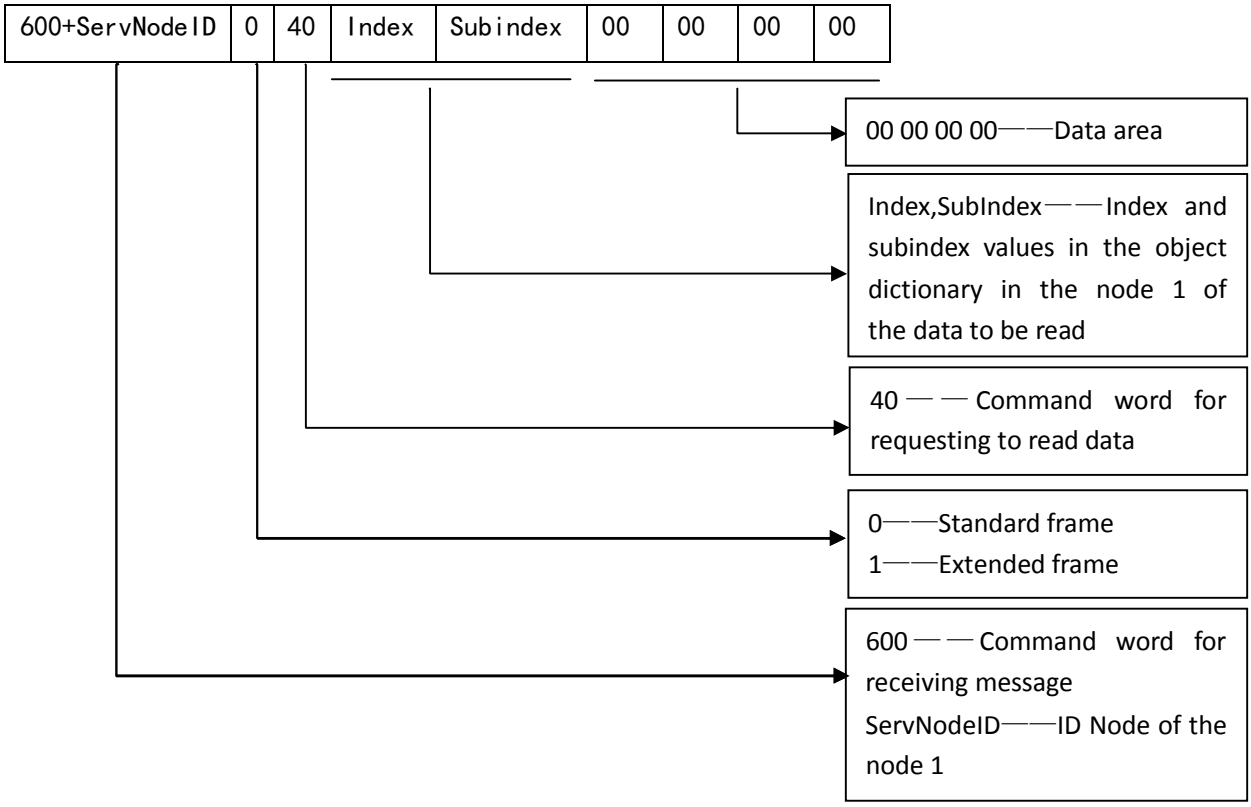
11 Appendix 2 CANOPEN Communication example

11.1 SDO Reading and writing example

11.1.1 SDO Read

11.1.1.1 Data frame format

Master send:



Slave response:

When the data length is 1 byte								
580+ServNodeID	0	4F	Index	Subindex	d0	0	0	0
When the data length is 2 bytes								
580+ServNodeID	0	4B	Index	Subindex	d0	d1	0	0
When the data length is 3 bytes								
580+ServNodeID	0	47	Index	Subindex	d0	d1	d2	0
When the data length is 4 bytes								
580+ServNodeID	0	43	Index	Subindex	d0	d1	d2	d3

11.1.1.2 SDO Read example

Master send: 605 40 01 60 00 00 00 00 00

Slave response: 585 4F 01 60 00 08 00 00 00

The master initiated a read request to the device whose node ID is 5. The index and subindex of the request are 0x6001 and 0x00 respectively, which corresponds to controller status parameter in the PMC007 Object Dictionary. The slave response 4F indicates that the parameter length is one byte, the data is 0x08 and the device is in busy state.

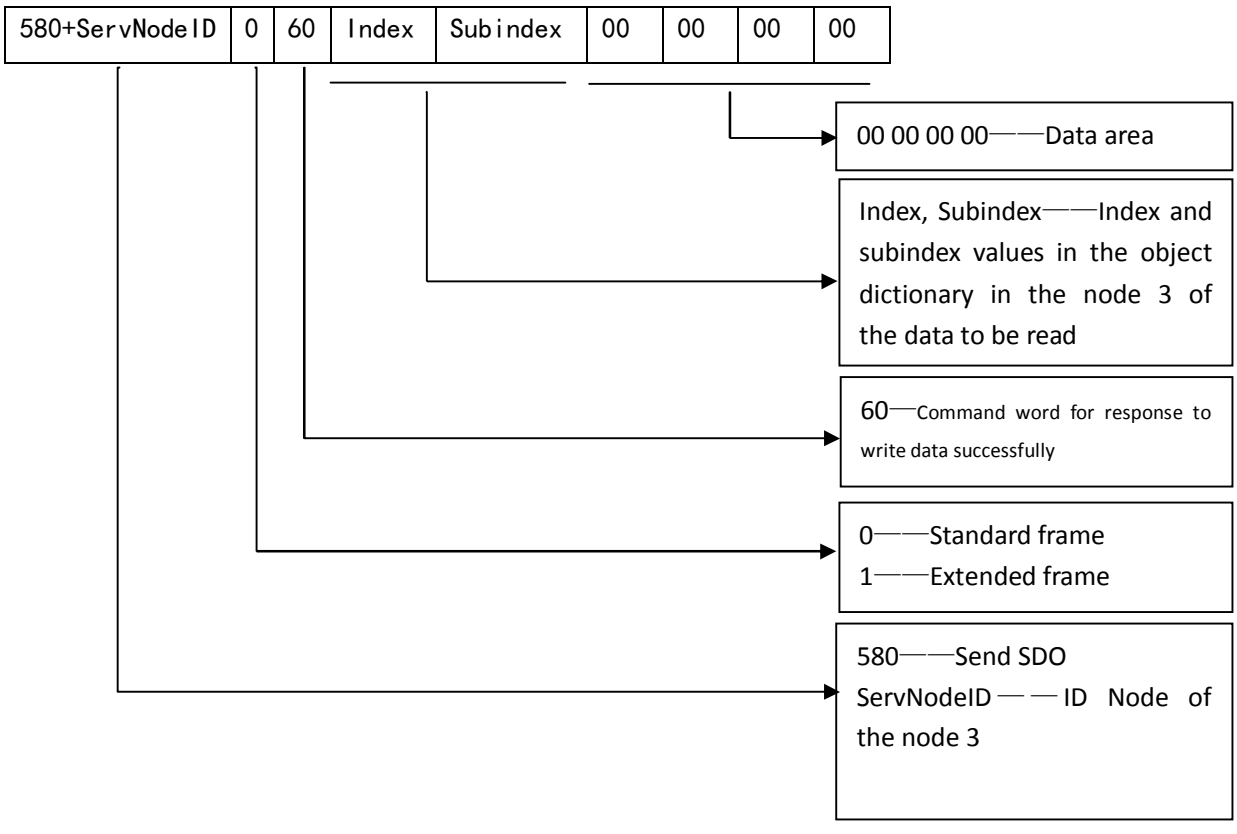
11.1.2 SDO Write in

11.1.2.1 Data frame format

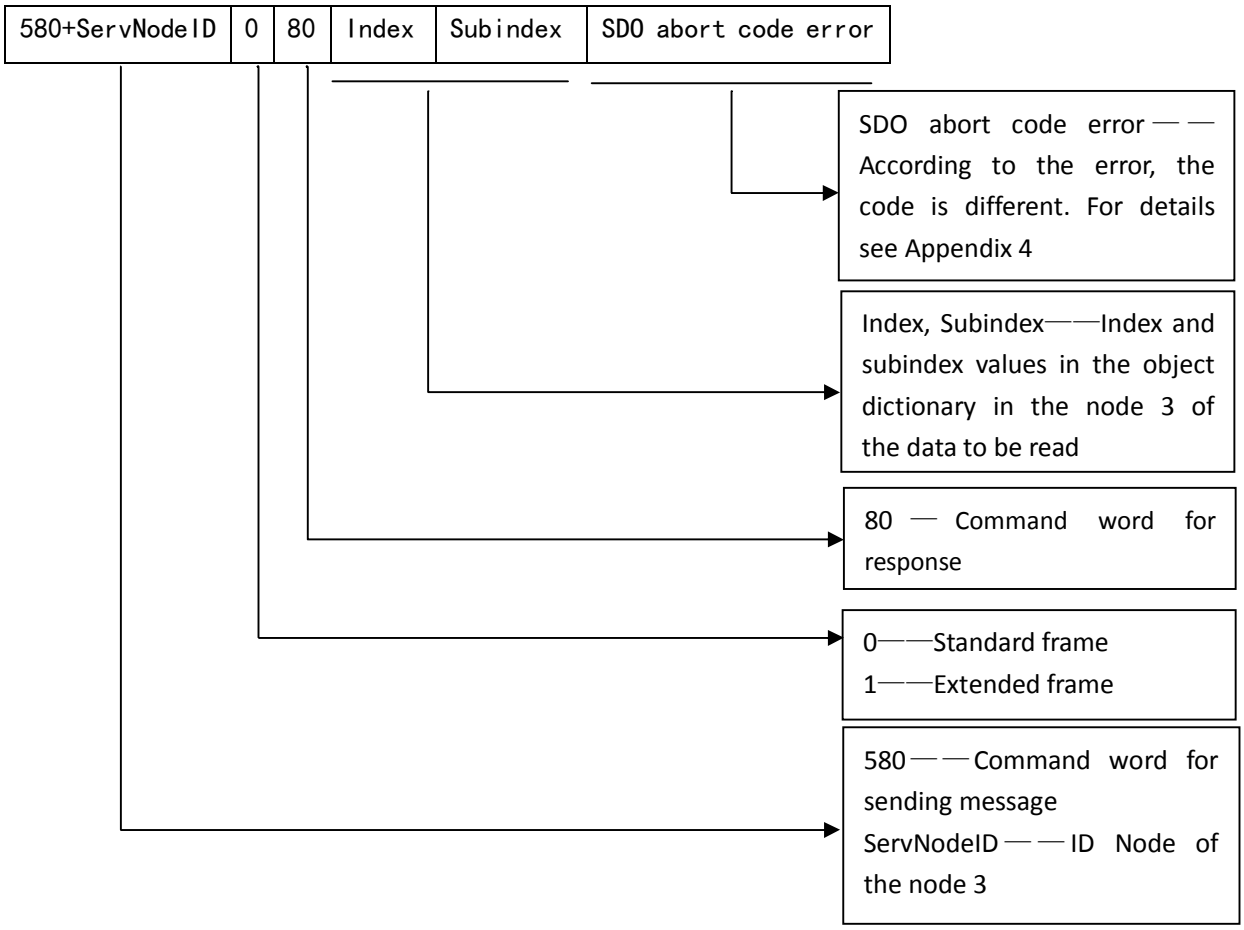
Master send:

When the data length is 1 byte								
600+ServNodeID	0	2F	Index	Subindex	d0	0	0	0
When the data length is 2 byte								
600+ServNodeID	0	2B	Index	Subindex	d0	d1	0	0
When the data length is 3 byte								
600+ServNodeID	0	27	Index	Subindex	d0	d1	d2	0
When the data length is 4 byte								
600+ServNodeID	0	23	Index	Subindex	d0	d1	d2	d3

Correct response from the slave station:



Error response from the slave station:



Note: Abort code error SDO returns the corresponding parameters according to the specific error. The specific parameters are shown in Appendix 4.

11.1.2.2 SDO Write example

Master send: 605 2F 03 20 00 07 00 00 00

Slave response: 585 60 03 20 00 00 00 00 00

The master initiated a write request to the device whose node ID is 5. The index and subindex of the request are 0x2003 and 0x00 respectively, which corresponds to the baud rate setting parameter in the PMC007 Object Dictionary. and the write data is 7, which indicates the baud rate is set to 800Kbit/s. The slave response 60 indicates the data is written successfully.

Master send: 605 23 04 60 00 80 0C 00 00

Slave response: 585 80 04 60 00 22 00 00 08

The master initiated a write request to the device whose node ID is 5. The index and subindex of the request are 0x6004 and 0x00 respectively, which corresponds to the step command parameter in the PMC007 Object Dictionary. and the write data is C80(3200), which indicates that the motor performs 3200 steps. The slave response 60 indicates the data is written unsuccessfully, and error code is 0x08000022. See Appendix 4 we can know that the error code indicates that the data cannot be transferred or saved to the application due to the current device status. Check whether the controller status parameter is that the external stop is enabled and whether there is an error in the error state.

12 Appendix 3 PDO configuration example

12.1 PDO overview

PDO communication is based on the Producer/Consumer model, which is mainly used to transfer real-time data. The node which generated data puts the data with its own node ID on the bus, and nodes which need the data can be configured to receive the data sent by the node. The transmission of PDO is triggered by the event, which can represent a change in a PDO variable and can also be a time of expiration or a specific message to be received. Process data is transmitted directly in an CAN message without a protocol header file. The length of a PDO is between 0 and 8 bytes.

PDOs is included in the mapping parameter and communication parameter. PMC007xx supports 4 PDOs.

12.1.1 The structure PDO——Mapping parameter

A PDO in the Object Dictionary consists of adjacent items. The mapping parameters define the connection of these items. A mapping parameter defines a data source through an index, a subindex, and a number of bits.

For example:

Index	*Sub-index*	*Object Data*	*Description*
0x1A00	0	4	Number of mapped entries
	1	0x20000310	The entry at index 0x2000, sub-index 3, with a length of 16 bit, is mapped to bytes 0 and 1 within the CAN message.
	2	0x20000108	The entry at index 0x2000, sub-index 1, with a length of 8 bit, is mapped to byte 2 within the CAN message.
	...	...	

Table 1: Example for mapping parameters for the first TPDO

A CAN message has not more than 8 bytes. This means that there can send 8 object items at most when there is only one PDO used.

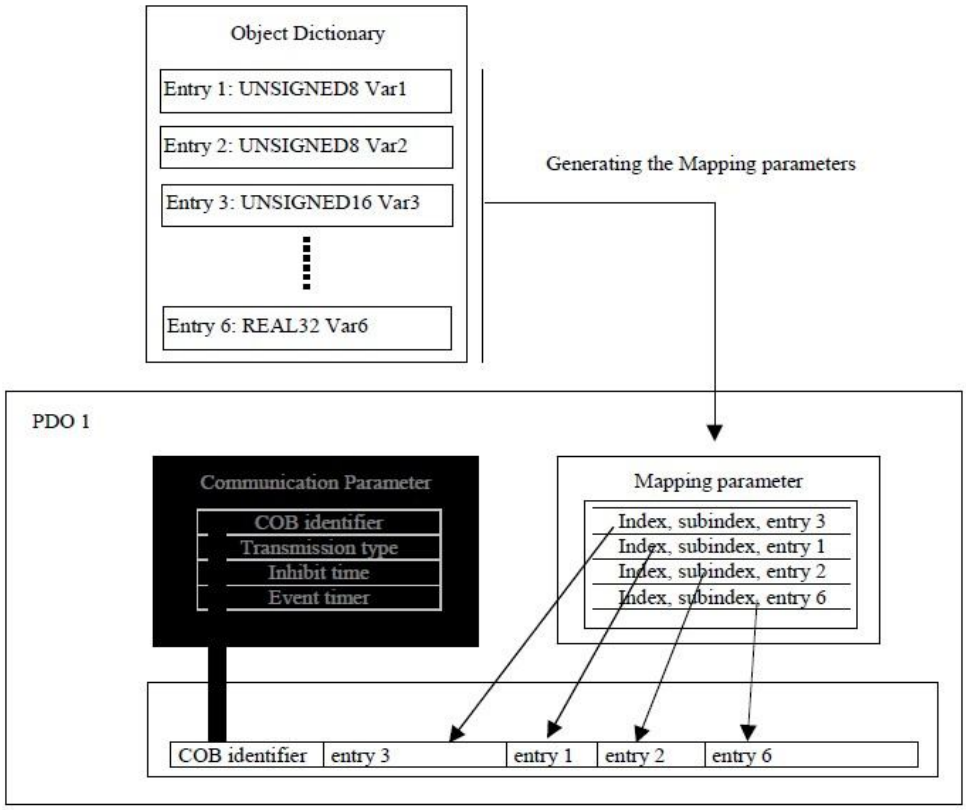


Figure 3: Mapping of Object Dictionary entries into a PDO

12.1.2 The structure PDO——Communication parameter

In order to transmit a PDO, the communication parameter defines the nature of the transport and the CAN identifier.

Index	Sub-index	Object Data	Description
1800h	0	Number on entries	
	1	COB-ID	CAN identifier for the PDO
	2	Transmission Type	transmission type of the PDO
	3	Inhibit Time	minimum inhibit time for a TPDO
	4	reserved	reserved
	5	Event Time	maximum time between two TPDOs

Table 4: Communication parameter for the first TPDO

PDO communication parameter is an item in the Object Dictionary.

(RPDOs: index 0x1400–0x15FF, TPDOs: 0x1800–0x19FF)

If allowed, the communication parameter can be modified by the CAN with the help of the data service object.

12.1.2.1 COB-ID(CAN identifier. Subindex 1)

COB-ID as proof of identity, the priority of PDO is before the bus access. For

every CAN message, there is only one sender (producer). However, it allows multiple recipients (consumers) for the existing message.

Bit	*31*	*30*	*29*	*28 – 11*	*10 - 0*
11-bit-ID	0/1	0/1	0	00000000000000000	11-bit identifier
29-bit-ID	0/1	0/1	1	29-bit identifier	

Table 5: Structure of a COB-ID for PDOs

The thirtieth bit is 0, which indicates that a remote transmission request (RTR) is allowed for this PDO.

PDO COB-ID distribution:

PDO1(send)	181H-1FFH
PDO1(receive)	201H-27FH
PDO2(send)	281H-2FFH
PDO2(receive)	301H-37FH
PDO3(send)	381H-3FFH
PDO3(receive)	401H-47FH
PDO4(send)	481H-4FFH
PDO4(receive)	501H-57FH

12.1.3 PDO Trigger mode

Sending of PDO can be triggered by the following methods:

1) Event trigger.
2) Time trigger.
3) Single query.
4) Synchronization.

When only use event to trigger the sending of PDO, once the event process is changed, the PDO is sent. It may bring very serious consequences, that is, when the frequency of a process data change is very high, the PDO is sent uninterruptedly , that will cause the message of other nodes is not sent out, which seriously affect the efficiency of the bus.

CANopen uses the " Inhibit time " mechanism to solve this problem. Inhibit time is a configurable time period in units of 100 us. The same PDO sends at least this time interval, so it can determine the maximum transmission frequency of an event triggered PDO.

Generally, the sending of PDO can be triggered by a combination of any of the trigger mode. But the most common way is to combine the Event trigger and Time trigger. In the

case of single event trigger, since process data did not change for a long time (such as temperature variables), the PDO have not been triggered for a long time. It will affect the nodes just joined the network. So if plus time triggered mode, PDO is forced to send again within the stipulated time. For example, for a PDO, the inhibition time is configured as 5, event timer is configured as 250, so the PDO can be sent when process data changes. The minimal interval of sending is 5ms, on the other hand, no matter whether there is no change in the data, the PDO will be sent every 250ms.

Configuration of PDO trigger mode is realized through setting subindex 2 in the Object Dictionary of PDO communication parameter. The range of the subindex is 0-255 The following lists the different values for different trigger modes.

0: PDO is sent after SYNC is received, but not cycle.

1-240: PDO is sent periodically after SYNC is received . The value is the number of SYNC between two send of PDO .

255：Event trigger.

12.2 PDO Configuration example

PMC007xx supports PDO mapping by SDO configuration. To configure the GPIO value to be TPDO1 as an example, the SDO is sent as:

Set the communication COB-ID to be 187, that is, the device whose node ID is 7 receives the PDO

Master send: 605 23 00 18 01 87 01 00 00

Event trigger is set

Master send: 605 2F 00 18 02 FF 00 00 00

Set Inhibit time as 5ms

Master send: 605 2B 00 18 03 32 00 00 00

Set Event time as 1000ms

Master send:605 2B 00 18 05 E8 03 00 00

Set the number of map entries to be 1

Master send:605 2F 00 1A 00 01 00 00 00

Set the mapping parameters to map 0x6012 to TPDO1

Master send:605 23 00 1A 01 10 00 12 60

After the configuration is completed, PMC007 will send PDO message every 1s. When the of value GPIO port is changed, PMC007 will also issue the message.

187 03 00

The message indicates that GPIO1 and GPIO2 are both high level.

13 Appendix 4 SDO abort code error

Abort code	Code function description
05030000	No alternation of trigger bits
05040000	SDO protocol timeout
05040001	Illegal or unknown Client/Server command word
05040002	Invalid block size (only Transfer Block mode)
05040003	Invalid serial number (only Transfer Block mode)
05030004	CRC error (only Transfer Block mode)
05030005	Out of memory
06010000	Access is not supported for the Object.
06010001	Try to read write-only objects
06010002	Try to write read-only objects
06020000	Object does not exist in the Object Dictionary
06040041	Object cannot be mapped to PDO
06040042	The number and length of the mapped object exceeds the PDO length
06040043	General parameters are not compatible
06040047	General equipment is not compatible
06060000	Hardware error causes the object access failure
06060010	Data type does not match, and service parameter length does not match
06060012	Data type does not match, the service parameter is too large
06060013	Data type does not match, the service parameter is too small
06090011	The subindex does not exist
06090030	Beyond the range of the parameter values (when write access)
06090031	Parameter value is written too large
06090032	Parameter value is written too small
06090036	The maximum value is less than the minimum value
08000000	General error
08000020	Data cannot be transferred or saved to applications
08000021	Due to local control, data cannot be transferred or saved to applications
08000022	Due to the current device status, data cannot be transferred or saved to applications
08000023	The dynamic condition of Object dictionary generates error or Object Dictionary does not exist