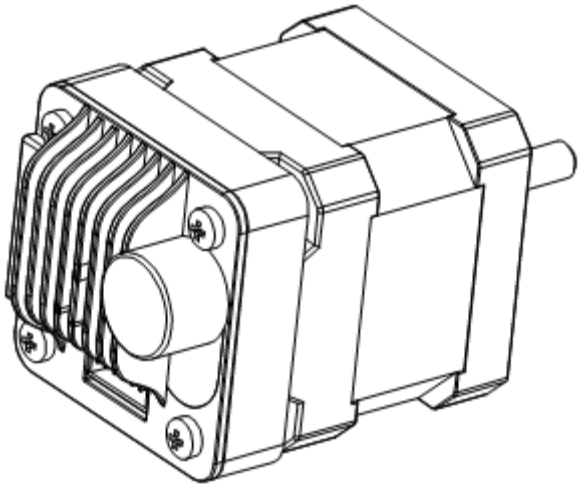
PUSIROBOT

CQPUSI ROBOT CONTROL SYSTEM

User Manual

PMC006B4

Miniature Integrated Stepper Motor Controller



1. Version Control

1） Update Records

Date	Author	Version	Remarks
2010-10-2	liur	V0.1.0	Initial
2011-02-23	tangxi	V0.1.1	Add RS485 feature
2011-04-05	robert	V0.1.2	Change baud setting
2012-5-27	liur	V0.1.10	Re-define port function
2012-6-22	jacky	V0.1.11	Add auto half current
2013-11-3	tangxi	V0.1.12	Add external stop
2014-04-15	jacky	V0.1.13	Modify auto accelerate
2014-05-06	robert	V0.1.14	Add self test
2014-07-16	liur	V0.1.15	Modify current setting
2014-10-23	robert	V0.1.18	Modify layout & port
2014-11-7	liur	V0.1.19	Modify input port trigger
2014-11-20	liur	V0.1.20	Add IO operating instruction.
2014-11-21	liur	V0.1.21	Add user defined instruction.
2015-01-03	Liur	V0.1.22	Add description for debug tool software.
2015-01-23	jacky	V0.1.23	Change ext configure and IO arrange
2015-6-22	Lijiwen	V0.1.24	Modify description due to software
2015-8-26	Jacky	V0.1.25	Append opt description
2016-2-17	huangcheng	V0.1.30	Software Arch update
2016-03-15	Liur	V0.1.31	Supplement description for acc/dec
2016-05-12	Liur	V0.1.32	Add PMC006B3
2016-06-02	huangcheng	V0.1.33	Add analog speed regulation
2017-06-18	huangcheng	V0.1.34	Add stall detection
2017-10-17	liur	V0.1.35	Update feature description

Catalog

1 Introduction

1.1 Statement of intellectual property right

PMC006B4 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.

Since PMC006B4 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. 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

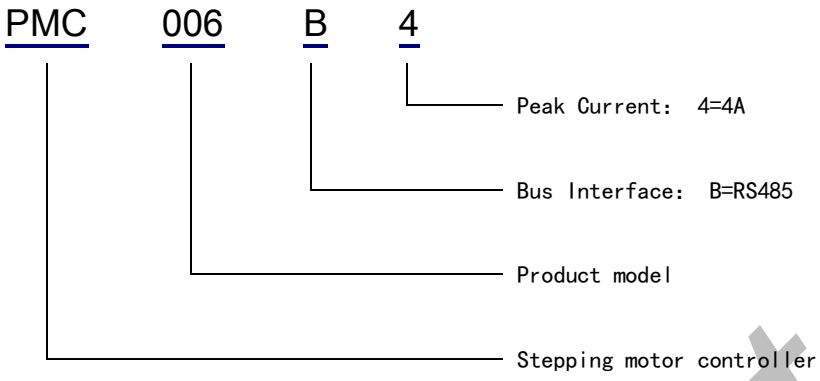
PMC006B4 is a kind of miniature integrated stepper motor microstepping controller, which can be directly installed in the rear of 42/57 etc series stepper motor. The series controller is controlled by RS485 bus. It is easy to achieve industrial control network of as many as 120 nodes by using PMC006B4 controller, which would have excellent noise suppression and excellent motion smoothness. PMC006B4 provides a simple and rich set of control commands, which not only greatly simplifies the complexity of the upper layer control system, but also maximally reserves flexibility of control, and is especially suitable for various industrial applications that require low vibration, high precision and wide voltage range.

2.2 Features

- ✓ Wide range of 8-36V single voltage supply
- ✓ Output current 0.3A ~ 3A, adjustable phase current by commands
- ✓ Automatic control of S curve acceleration and deceleration
- ✓ Ultra-low noise and low vibration control algorithm
- ✓ 2 external switches input ports for configurable emergency stop
- ✓ Support 0/2/4/8/16/32/64/128 microstepping resolution
- ✓ Support 4/6/8 lines of 2 phase stepper motor
- ✓ 10 GPIO ports
- ✓ Support speed regulation by analog quantity
- ✓ Stall detection without sensor
- ✓ Burning and off-line automatic execution of custom program
- ✓ Automatic current compensation and speed compensation
- ✓ Configurable instruction of automatic current attenuation function
- ✓ Miniature size 42mm*42mm*18mm
- ✓ Precision aluminum shell, conducive to the protection and heat dissipation
- ✓ Free disassembly, online firmware upgrades
- ✓ Control routines and the underlying driver based on VC++

2.3 Production & Ordering Information

In order to serve you quicker and better, please provide the PMC006B4 product model number in following format when ordering.



3 Connector Description

3.1 Terminal port location

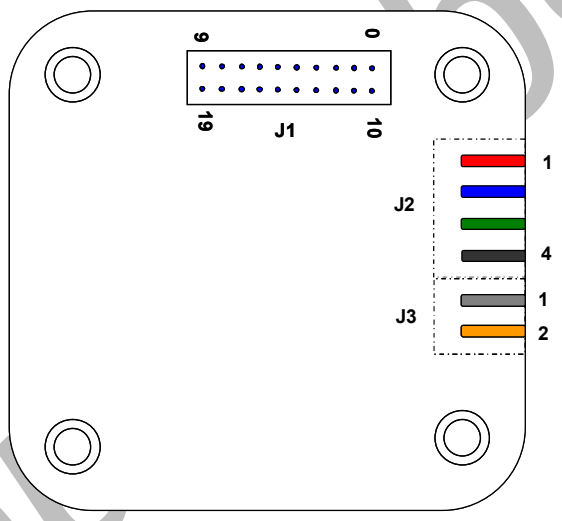


Figure 3-1

3.2 Motor connection J2

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

Description:

M10, M11: Connect to the stepper motor phase A

M20, M21: Connect to the stepper motor phase B

WARNING: Incorrect connection of power or phase will permanently damage the controller!

3.3 Power connection J3

Pin no:	1	2
Designator:	GND	VCC

Description:

VCC: Supply voltage, 8~36V

GND: Supply voltage ground

WARNING: Charged plug is prohibited, which may damage the controller permanently!

3.4 Signal connection J1

Pin no:	0	1	2	3	4	5	6	7	8	9
Designator:	DVDD	DVDD	GND	GPI2	GPI1	NC	EXT1	FSET	RXD	TXD
Pin no:	10	11	12	13	14	15	16	17	18	19
Designator:	GPI4	GPI5	GND485	EXT2	GPI3	GPO1	GPO2	GPO3	GPO4	GPO5

Description:

GPI1~5: Generic inputs 1-5

GPO1~5: Generic outputs 1-5

NC: Null. Reserved bit, not connected to any of the peripherals

DVDD: Voltage output(+5V)

GND: Digital ground

GND485: RS485 ground

EXT1: External limit switch signal input 1 (open loop)

EXT2: External limit switch signal input 2 (open loop)

TXD: RS232/RS485 bus transmitting signals or connected CAN transceiver module

RXD: RS232/RS485 bus receiving signals or connected CAN transceiver module

FSET: Restore factory settings, Active low level

WARNING: The voltage of all signal ports must be between -0.3V~+5.3V in addition to TXD and RXD.

3.4.1 Connector assembly

The J1 port of the PMC006B4 controller is a precise small spacing connector. When plugged in, it is necessary to ensure appropriate strength and parallel angles. Incorrect plugging may cause the pin to be deformed or broken, as shown in the following right figure.

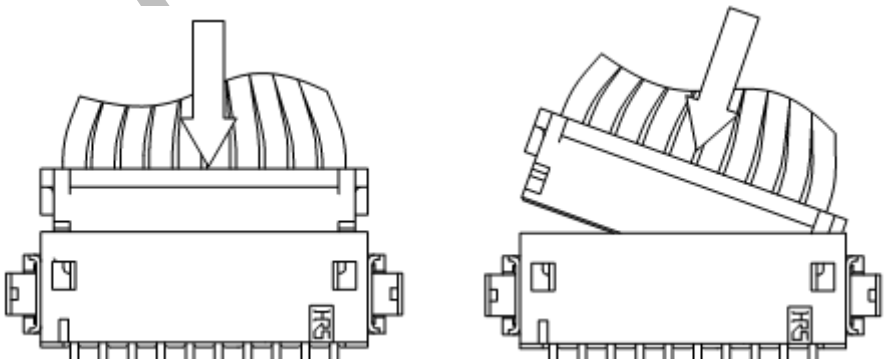


Figure 3-2

WARNING: Charged plug is prohibited, which may damage the controller permanently!

3.5 RS485 Network Operation

It provides a network scheme which uses RS485 bus to connect multiple PMC006B4 controllers in the Figure 3-3. Maximum communication distance of the scheme is 1200 meters. If the transfer distance is over 50 meters when using a pair of twisted pair to connect all the nodes, both ends of the network should be terminated with 120Ω terminating resistors to prevent signal reflection and overshoot. Meanwhile, the host RS485 and the controller of each node must be common-grounded.

Warning: The upper and lower limit of the signal threshold of RS-485 is ±200mV. That is, when A-B>200mV, the bus state should be expressed as "1"; when A-B<-200mV, the bus state should be expressed as "0". However, when the A-B is between ±200mV, the bus state is not determined. So, in the actual network, it is recommended that the user set up and drop resistance in the B, A line, to avoid this uncertainty.

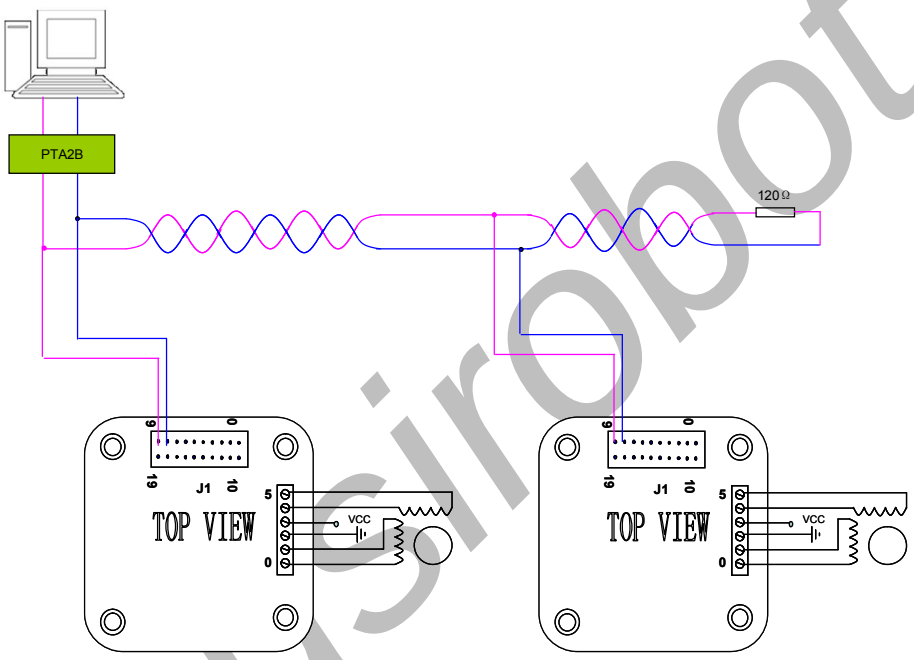


Figure 3-3

For detailed understanding of module PTA2B used to achieve RS232 to RS485 conversion, please contact the sales staff.

3.6 Emergency Switch Connection

The default mode of PMC006B4 controller is open loop operation mode. The Ext1 and Ext2 pins are used to connect the external limit switches. The trigger mode of each pin can be selected by the instruction in real-time. Default trigger mode is falling edge trigger. The user can also open or close any of the limit switches by the instruction.

There are two kinds of trigger mode, which are the rising edge trigger and falling edge trigger. When falling edge trigger is selected, the internal pull-up resistor is enabled automatically. The input pins can be directly connected to the collector of optical coupler as shown in Figure 3-5 left; when rising edge trigger

is selected, the internal pull-up resistor is not enabled. The external pull-down resistor is needed to pull the clamp output voltage to low reliably such as Figure 3-5 right.

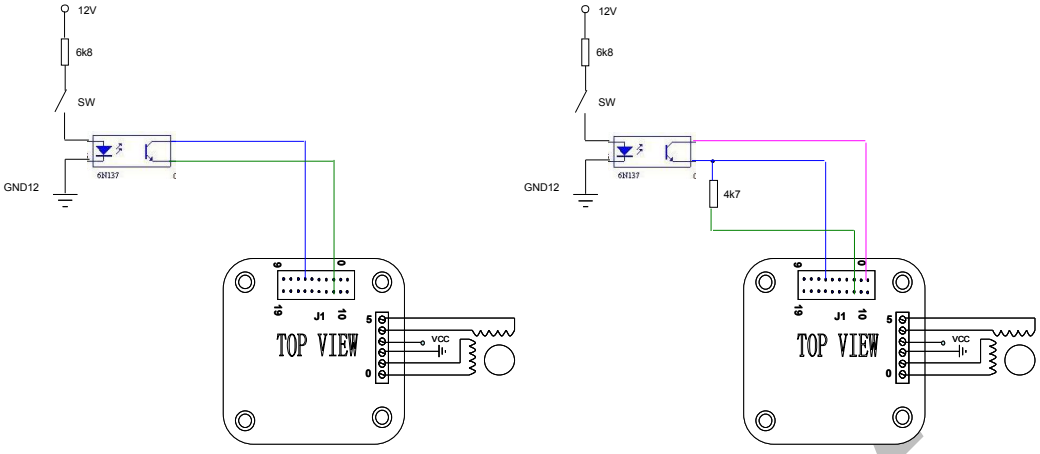


Figure 3-5

Above illustrated connection is suit for the case in which the optical emitter is normally open, however, in some of scenarios such as ram pump driver. The optical emitter is normally closed. When the block moves to a certain position, receiver is disconnected. So, when using the falling edge trigger, input pin can be connected directly to the emitter of opticalcoupler as Figure 3-6 left; when using the rising edge trigger, input pin can be connected directly to the collector of opticalcoupler as Figure 3-6 right.

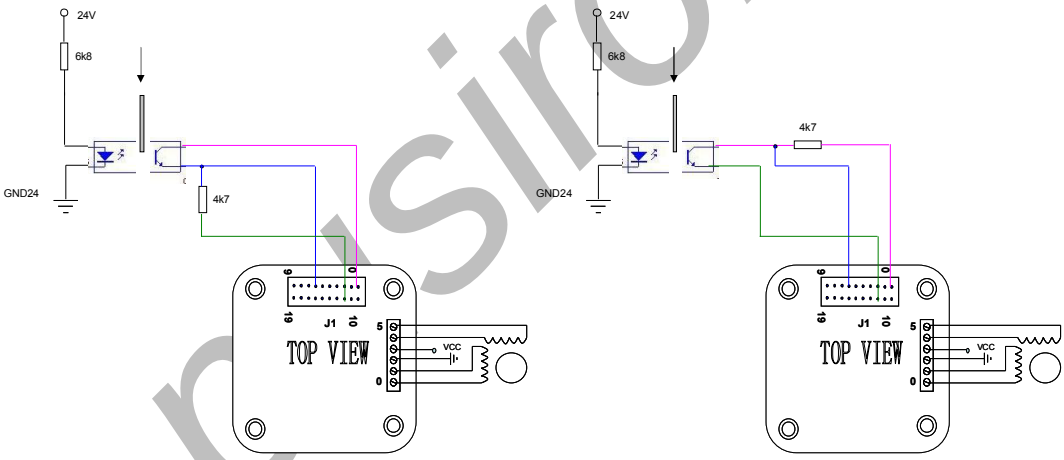


Figure 3-6

3.6.1 Proximity Switch Connection

The above connections are both connected with the opposite-type photoelectric sensor. The PMC006B4 controller can also be connected with the 24V DC three wire system NPN open-collector output type proximity switch, such as OMRON E2EC/ X□C □or E2E-X□D1S series. The connection method is shown in the following diagram. 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.

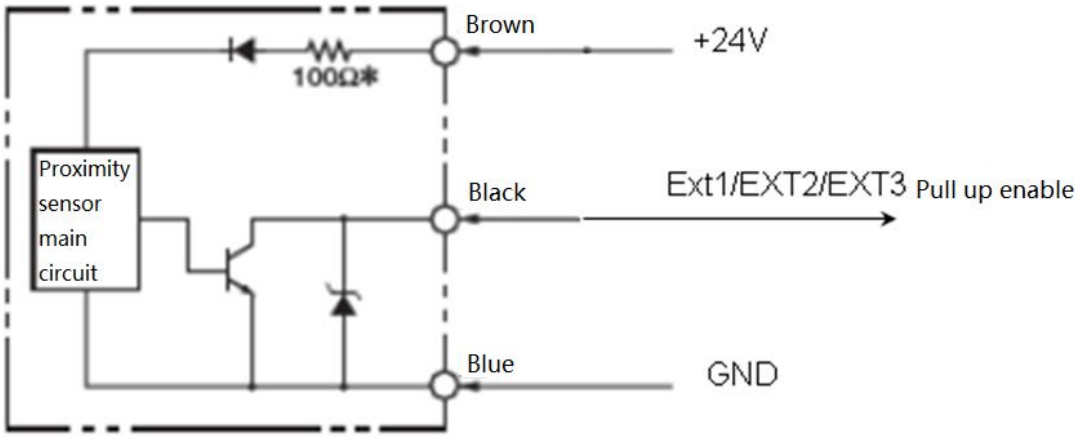


Figure 3-6

3.7 Analog Adjusting Speed Connection

The function of analog adjusting speed connection speed can be used when the PMC006B4 controller is in offline mode. In this case, GP11 pin is used as analog input port, as shown in the Figure 3-7. Please refer to chapter 4.3 for detailed operation.

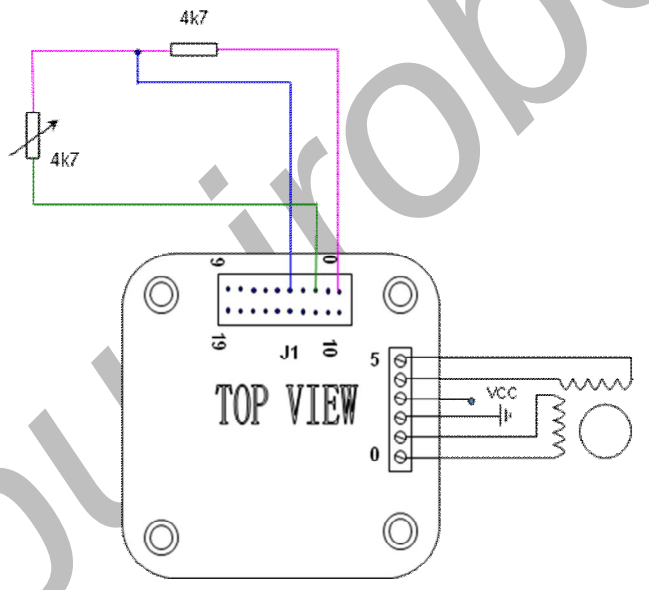


Figure 3-7

3.8 Factory Reset

When the PMC006B4 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 Instruction Set

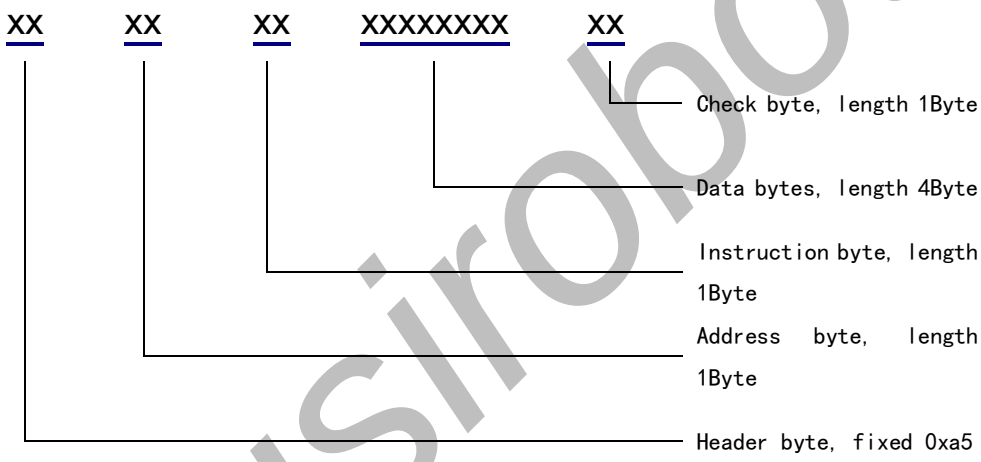
The PMC006B4 controller uses a set of simplified instruction to regulate and simplify the operation of the host computer to the controller. The controller receives the operation instructions from the host computer, and returns the ACK to the host computer, and returns the data or the state of user's needs.

Communication mode between host computer and the PMC006B4 controller is half duplex. The same time the information can only be transmitted from the host computer to the controller or the controller to the host computer. PMC006B4 controllers can't communicate with each other.

4.1 Instruction Structure

The communication format between PMC006B4 and the host computer is transparent transmission format based on RS232 string, and the frame is transmitted in sequence of bytes, and each frame is fixed to 8-byte length.

Instruction Structure::



The length of command frame which is transmit to controller by host controller and the acknowledgement frame which is responded to host computer by controller is fixed 8 bytes (without any spaces or separators). If the controller sends out an instruction frame which has valid address and is verified correctly, the PMC006B4 controller returns an answer frame (even if the instruction is invalid).

Both 8 bytes of the command frame and the response frame must be transmitted before the bus can be released, otherwise the host computer or the slave machine will wait for a timeout.

When the address and the check byte in the instruction frame that the host computer sends out is not correct, it will not receive any response frame. At this time the host computer will continue to send an effective instruction frame after its own timeout mechanism is effect, which will not cause the bus conflict or hang up.

4.1.1 Address Byte

When the host computer sends out the instruction, the byte value range is 0x1~0x78. In the network connection mode, the controller address of each node must be unique。 The user can modify the controller address in a single mode (the default address is 0xff); when the PMC006B4 controller responds to instruction, the byte is fixed to 0x7a.

4.1.2 Instruction Byte

When the host computer sends a command, the byte content is the corresponding instruction. See details the 4.2 instruction summary table.

When the PMC006B4 controller feeds back instruction, the byte is the controller's address.

4.1.3 Data Byte

When the host computer sends a command, the four bytes are the required data of corresponding instructions. For the instructions without data, the 4 bytes can be any value.

When the PMC006B4 controller feeds back instruction, the four bytes is the returned state or data. For data without feedback, the content of 4 bytes should be ignored.

Both PC and controller, the data type of 4 bytes is defined as a signed integer (ANSC C: signed long) and the low bit is first to send or receive in the transmission, such as data 0x12345678, the transmission order is 0x78 - > 0x56 ->0x34 ->0x12.

4.1.4 CRC Byte

The check byte is defined as the sum of the previous 7 bytes (discarding carry); in the instruction frame, the host computer calculates and transmits. In the acknowledgement frame, the PMC006B4 controller calculates and transmits. The frame whose check byte is not correct will be discarded without processing.

4.2 Instruction Set Summary

Instruction	Description	Data range	Note
s(0x73)	Set rotation steps	1~0x7fffffff	
m(0x6d)	Set microstepping	0/2/4/8/16/32/64/128	Can be saved after power-down
p(0x70)	Reserved	Reserved	
t(0x74)	Save all parameters	----	
c(0x63)	Read current position	----	
d(0x64)	Set direction of rotation	0: backward, 1: forward	
v(0x76)	Set maximum speed	1~16000 PPS (step/second)	

w(0x77)	Set device address	1~120	Can be saved after power-down
h(0x68)	Reserved	Reserved	
r(0x72)	Read acc para	----	
e(0x65)	Set peak phase current	400~4000 (Unit: mA)	Can be saved after power-down
f(0x66)	Set the external emergency stop	0/1/2/3	
n(0x6e)	Read phase current setting	----	
k(0x6b)	Read microstepping setting	----	
q(0x71)	Read speed setting	----	
o(0x6f)	Set deceleration coefficient	0-5	Can be saved after power-down
a(0x61)	Set the automatic current decay enable	0/1	Can be saved after power-down
g(0x67)	Set offline enable	0/1	
i(0x69)	Set current position	0-0x7fffffff	
j(0x6a)	Read status 1 of controller	Bit[3:0]定义见指令详解	
l(0x6c)	Clean flag of ext_stop2	----	
b(0x62)	Clean flag of ext_stop1	----	
x(0x78)	Read IO port	0~0x3ff	
y(0x79)	Write IO port	0~0xff	
z(0x5a)	Try run the current custom program	----	
u(0x75)	Set acceleration coefficient	0-5	Can be saved after power-down
I(0x49)	Abort the current step command	0	
J(0x4a)	Set ext_stop trigger mode	0~3	Can be saved after power-down
K(0x4b)	Read ext_stop trigger mode	----	
F(0x46)	Set offline execution	0/1	Can be saved after power-down
R(0x52)	Read firmware version	0	
L(0x4c)	Set start speed(pps)	65-3000	Can be saved after power-down
M(0x4d)	Set current	100-1200	Can be saved after

	compensation factor		power-down
N(0x4e)	Set speed mode enable	0/1	
O(0x4f)	Read status 2 of controller		
P(0x50)	Set speed compensation factor	300-2000	Can be saved after power-down
Q(0x51)	Automatic current attenuation coefficient	1-4	Can be saved after power-down
T(0x54)	Read/write stall length	0~0x3f (More than 0x3f for reading)	
S(0x53)	Set stop speed	65-3000	Can be saved after power-down
X(0x58)	Read stall position	----	
Y(0x59)	Read/write stall configuration register	0~0x7 (More than 0x7 for reading) Write operation: Bit0 : stall detection enable; Bit1 : stop motor after stall; Bit2: GP05 is pulled up after stall;	
Z(0x5a)	Read/write stall trigger level	0~0x1ff (more than 0x1ff for reading)	

4.3 Instruction Details

The software package includes examples of communication interface DLL based on vcc++ and related examples. The user program can operate the PMC006B4 controller by directly calling the interface function of DLL without need to understand the specific transmission mode and feedback information of each instruction. The following is a detailed explanation of the using method of a part of instruction.

4.3.1 Step Instruction

The step instruction 0x73 can control the stepper motor to rotate specified number of steps according to the setting direction, speed, acceleration. The number of steps is calculated according to the current microstepping settings.

If the controller is busy, stepping commands will be ignored. So, you should first check the controller's status bits through the 0x6A command. For a detailed explanation, please refer to the description of controller status.

Host computer send: 0xa5 0x01 0x73 0x1f 0x01 0x00 0x00 0x39

Controller ACK: 0xa5 0x7a 0x01 0x03 0x00 0x00 0x00 0x23

Description: Send a command to the controller whose address is 0x01 to rotate 287.

4.3.2 Set micro-stepping resolution

PMC006B4 controller supports maximum 128 microstepping resolution。It would not cause loss of accuracy when the controller is converted from a low microstepping to a high microstepping. However, it would cause loss of accuracy when the controller is converted from a high microstepping to a low microstepping, because the motor rotor is in a middle position.

When the acceleration value is set high, it may cause the motor out of step in a low microstepping. In this condition, it's better to increase the microstepping.

The setting value of microstepping will be stored after the controller is powered off. And also the final configuration is enabled automatically when the controller is powered on next time

Host computer send: 0xa5 0x02 0x6d 0x04 0x00 0x00 0x00 0x18

Controller ACK: 0xa5 0x7a 0x02 0x04 0x00 0x00 0x00 0x25

Description: Set the microstepping resolution of the controller whose address is 0x02 to quarter.

4.3.3 Set acceleration and deceleration

PMC006B4 controller supports the function of S curve automatic acceleration and deceleration without additional module supports. After the acceleration/deceleration/start speed/stop speed is set by instruction, the controller calculates the acceleration curve in real time until the maximum speed is caught. Then the controller calculates the deceleration curve to control the motor slow down, as shown in figure 4-1.

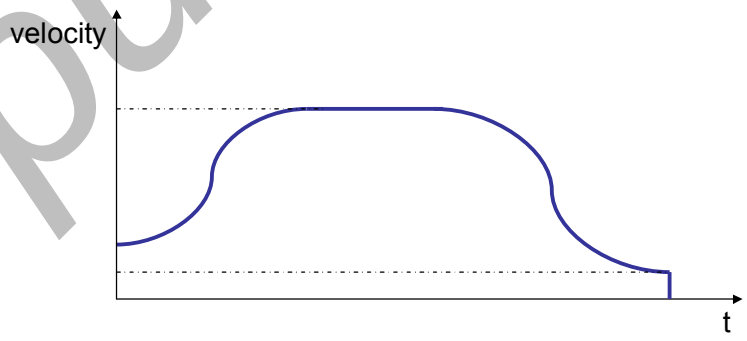


Figure 4-1

Acceleration coefficient and deceleration coefficient need to be separately configured, but they both have the same dimension. The corresponding relationship between the coefficient and the acceleration value is as follows.

Coefficient	Acceleration PPS^2
0	1700

1	3400
2	6800
3	13600
4	27200
5	54400

When the start command is started, the controller controls acceleration and deceleration adaptively, in order to make the motor run the specified number of steps at the expected speed. Therefore, when the step numbers are few and the setting velocity is high, the actual acceleration and deceleration coefficient may increase automatically.

When the step numbers are less than 256, the controller would not use the method of S curve automatic acceleration and deceleration to control, but directly control the motor decelerate to the stop speed.

In most applications, the deceleration coefficient is set to be relatively small, and the acceleration coefficient is increased appropriately. Thus, the performance of motion is smoother than the application which acceleration and deceleration is symmetry.

4.3.4 Read motor position

Whenever a rotation command is issued, according to the number of steps of the command the controller automatically records the current position which is represented by a signed integer. A positive value indicates clockwise position, while a negative value indicates the anti clockwise position.

Because the value of current position is calculated by the number of steps, when a user needs to change the microstepping resolution, you should first read the current location information, and then reset the position by the 0x69 instruction to avoid error of conversion location.

When the controller is powered off, the position information is automatically cleared.

4.3.5 Set peak phase current

A current closed-loop control circuit is built in PMC006B4 controller. Current setting range is 400~4000mA, the smallest step unit is 150mA. It is recommended that the setting value of current does not exceed 3000mA.

Note：When the motor works at a very low speed, it is required to pay attention to the heating of the motor and the controller. Usually current only needs to set a smaller value in this condition.

4.3.6 Set the external emergency stop

Two limit switch input ports EXT1 and EXT2 are provided in PMC006B4 controller, which can be used as an emergency stop function in open loop mode. The user can set any one or two to enable, the data in the command can be set to 0~3, 0 means that

the two are not enabled, 1 means EXT1 is enabled, 2means EXT2 is enabled, 3 means the EXT1 and EXT2 is both enabled.

When the function of emergency stop is enabled, if an effective low level is detected at the relative input pin, the motor is locked by controller and stops responding to the command. The user can check the 6th, 7th bit of status byte by the 0x6a command, to see which one input pin is triggered. When the user uses the 0x62, the 0x6c command to clear the status flag bits of corresponding emergency stop, the controller will continue to respond to a new step command.

4.3.7 Set automatic current decay

Use this command to set whether or not phase current is automatically attenuated when the controller does not receive any step instruction. The default attenuation amplitude is 50%, the user can set the attenuation coefficient through the 0x4d instruction, the value is bigger, and the attenuation amplitude is higher. When the phase current is above 2.5A, the proposed configuration is more than 2.

4.3.8 Enter speed mode

Use this command to set the controller to enter the speed mode in which the motor will continue to run at the setting speed. In speed mode, can use this command with 0 parameters or use the 0x49 command to exit the speed mode. Notice the difference between these two instructions: Set the speed mode to 0 will make the motor gradually stopped, and the use of the 0x49 command will immediately stop the motor.

4.3.9 Set start speed and stop speed

Use 0x4c and 0x53 commands to set the motor start speed and stop speed, the default value is 600pps, and the minimum value is 65pps.

Note: When any one of these two values is higher than the setting value of maximum speed, the step command will not work.

4.3.10 Set current compensation factor

According to the parameter set by 0x4d,Controller makes a real-time compensation for c. The value of parameter is related to the phase inductance and internal resistance of motor. The default value is 300, which is suitable for most of the 42-stepping motor whose current value is between 0.9A-2A. For most of the 57 motors, the recommended value is 200.

4.3.11 Set speed compensation factor

In order to get the optimal motion smoothness, the controller optimizes current waveform according to the parameter which is set by 0x50, and the value of parameter is related to mutual inductance and sectional area of magnetic flux of the motor.

The default value is 1250, which is suitable for most of the 42 stepping motor whose current value is between 0.9A-2A. For most of the 57 motors, the recommended value is 512.

4.3.12 Set freerun enable

Using the command can set the offline enabled or disabled. When the function is enabled, the controller releases the control of motor immediately, the current step instruction is terminated, and the phase current is reduced to 0. All subsequent step instructions send by host computer will not be processed, until the user use 0x67 instruction to set offline disabled.

The difference between 0x67 and 0x49 is: The 0x49 only can stop the current step instruction, and the phase current is not 0. The 0x49 will not set status flag, dose not to prevent the execution of the next step instruction.

4.3.13 Analog adjusting speed

In user's offline program, user can enable the function of analogue adjusting speed. The GPI1 pin is used as an analog input port, and range of input voltage is 0 to 2.5V. User can set 2.5V corresponding to the maximum speed in offline program.

4.3.14 Stall detection

PMC006B4 uses BEMF to realize stall detection without sensor. The reliability of the detection results is closely related to the inductance parameters and rotational speed of the motor. First, using the 0x5a, the 0x54 instruction sets the trigger level and the stall length respectively, and then uses the 0x59 instruction to set up the stall configuration register. The current stall position is automatically recorded and the user can read through the 0x58 instruction, after stalled. When use the 0x70 command to clear up the stop sign bit, the GPO5 high level output will be cleared at the same time. If the stall sign bit is not manually cleared, the next step command will also be automatically cleared

4.3.15 GPIO port operation

5 general input ports GPI1~5 and 5 general output ports GPO1~5 are provided in PCM006B4 controller, which can be read and assigned by the command 0x78 and 0x79. These two commands are to be operated on a three-byte. The order from high to low of the 12 ports in the byte is: {EXT2, EXT1, GPI5, GPI4, GPO5, GPO4, GPO3, GPO2, GPO1, GPI3, GPI2, GPI1}. The port output command 0x79 only takes effect on the GPO bit, and is invalid for the GPI and EXT.

Host computer send: 0xa5 0x02 0x79 0x08 0x00 0x00 0x00 0x28

Controller ACK: 0xa5 0x7a 0x02 0x08 0x00 0x00 0x00 0x29

Description: The +5V is output at port GPO1, the 0V is output at GPO2~5.

4.3.16 Query controller status

The controller's status has two registers, which can be read by the command 0x6a and 0x4f. Respectively, 0x6a command is used to read the state 1 of the controller. There are a total of 5 bits in the status, the definition is as follows:

Bit	Designator	Description
0	STATUS	0: IDLE; 1: Busy
1	EXT1	0: No emergency stop; 1: emergency stop
2	EXT2	0: No emergency stop; 1: emergency stop
3	AUTO_DEC	0: auto decay disable; 1: auto decay
4	STALL	0: no stall; 1: stalled

6 bits of the status register 2 are defined as follows:

Bit	Designator	Description
0	FREE_RUN	0: Offline disable; 1: Offline enable
1	EXT1_EN	0: Ext_stop disable; 1: Ext_stop enable
2	EXT2_EN	0: Ext_stop disable; 1: Ext_stop enable
3	SPEED_MODE	0: displacement mode; 1: speed mode
4	DIR	0: backward; 1: forward
5	OFFLINE_AUTO	0: Not automatically run in offline mode; 1: automatically run in offline mode

4.3.17 Optimization of performance

The PMC006B4 controller is designed to provide excellent motion performance over the speed range of 65~16000pps. The higher the microstepping resolution is configured (the highest microstepping is 128), the better the performance of motion is and lower the level of noise is. In most application, PMC006xx in the 16 microstepping can provide a very good performance of motion, especially the small motor whose current is below 2A and the parameter don't need to be adjusted basically. For more than 57 or special type of motor, may need to adjust the parameters to obtain the best performance.

4.3.18 Parameters save

All parameters that can be saved when power down are indicated in the summary table in the 4.2 section. These parameters can be saved in EEPROM by sending 0x74 commands to controller. Because all the non-lost memory has a limited erase life, it is recommended that the user use the command only in the case of the real need.

It will take effect immediately after the address of controller is modified. So, when modify the address of controller by using the debug tool, user should click "scan" button to retrieve the site number after write the parameters of controller. Then, click "save all parameter" button to save address to EEPROM.

4.4 User-defined program

PMC006B4 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. Please refer to the "controller offline programming guide" for details about operation method and paradigm.

When the PMC006B4 controller works in offline mode, the UART (or CAN or RS485) communication interface is still responsive to the user's online instruction.

The maximum number of instructions that PMC006B4 controller supports for user is 100.

4.4.1 User Instruction Set Summary

PMC006B4 controller supports the following user defined instructions, these commands are provided by the CQPUSI tool software to interact with the controller automatically, and users do not need to write their own programs, only need to operate the command in the "custom programming interface".

Command	Function	Options	Data range
ROT	Rotate a given number of steps	0	1~0x7fffffff
MIC	Set microstepping	0	0/2/4/8/16/32/64/128
DIR	Set direction of rotation	0	0：反向，1：正向
EXTEN	Set ext_stop enable	0	0/1
FREE	Set freerun enable	0	0/1
CLR2	clean ext_stop2 flag	0	----
CLR1	clean ext_stop1 flag	0	----
VSET	Set speed(pps)	0	0-6000PPS
ACC	Set Acceleration	0	3-6
TRIG	Set external trigger mode	0	0-3
CNTI	Internal counter plus1	0	----
CNTC	Internal counter reset	0	----
JMP	Unconditional jump	0	0-50
JNE	Unequal jump	0	0-50
JEQ	Equal jump	0	0-50
WAIT	Waiting condition	1-7	0/1
OUT	Port output	1-5	0/1
CMP	Comparison	1-6	0~65536
RESET_EN	Set GPI1 reset enable	0-1	0-50
PAUSE_EN	Set GPI2 pause enable	0-1	0

4.4.2 User command description

The detail information for some of command is described as following.

4.4.2.1 CNTI，CNTC command

These two instructions are used to add and reset the internal counter, and the internal counter can be used as a function of the cycle count in the user's custom program. The value of the counter can be used as a comparison condition in the CMP command.

4.4.2.2 JMP command

Unconditional jump instruction, the program jumps to the specified location.

4.4.2.3 JNE，JEQ command

Conditional jump instruction. Based on the flag which is generated by CMP instruction, jump to the specified position. If the flag bit is 1, the program will jump to the specified position by the JEQ instruction; if the flag is 0, the program will jump to the specified position by the JEQ instruction.

4.4.2.4 WAIT command

Pause the program execution. Execute the next instruction until the condition of the option is satisfied. A total of 9 options can be selected, please refer to the CQPUSI tool software for "custom programming" interface settings for details.

Note: when using the ROT command to launch the motor rotation, the next instruction is executed immediately and don't wait until the rotation command is completed. So generally, there should be a WAIT instruction following the rotation command.

4.4.2.5 OUT command

Output value to GP01~5. The instruction can only output a port value at a time.

4.4.2.6 CMP command

Compare the value of option with the setting value. Option can be the value of the internal counter, or any one input port or an external stop status or all input ports as a bus data comparison, a total of 9 options can be selected. After comparing the internal flag will be set, if the result of comparison is equal, the flag is set to 1, otherwise set to 0.

4.4.2.7 RESET_EN and PAUSE_EN command

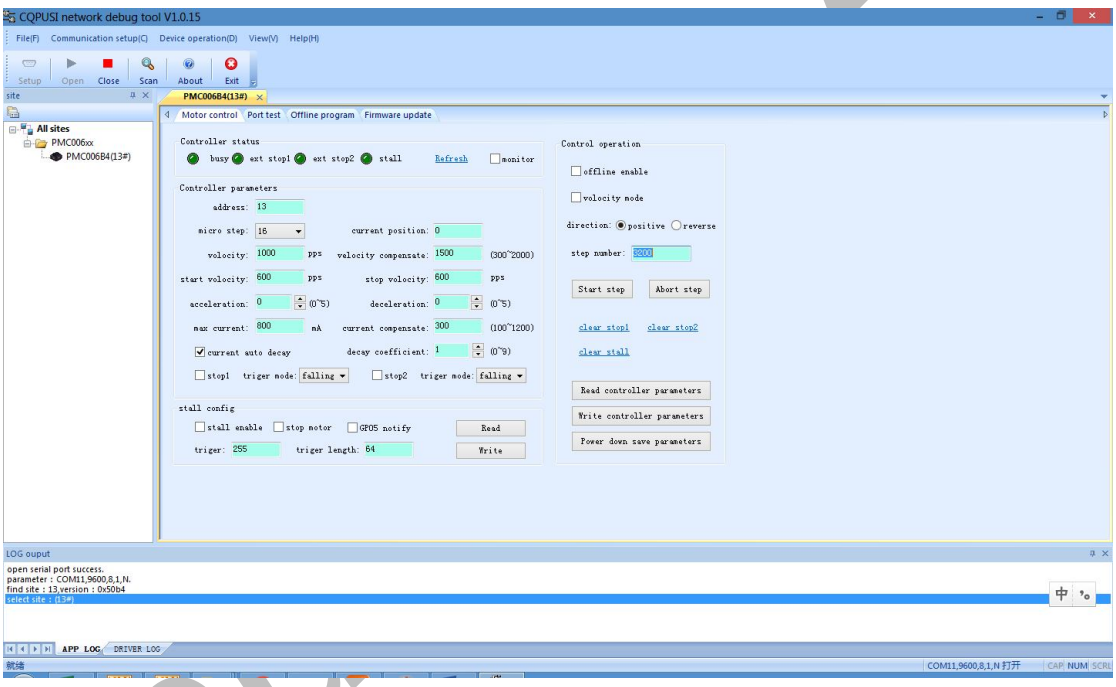
The controller can choose to bind GPI1 as the external reset stop key input, GPI2 as external pause/start key input. These functions can only be enabled by the offline program and only need to be done once in the program to take effect globally,

so the user should try to put these two statements in the beginning of the offline program. When the external suspension / launch function is enabled, the low-level pulse on the GPI2 will alternately start or pause the execution of the offline program, but the rotation command which has been issued will not be stopped. When the external reset stop function is enabled, the low-level pulse on the GPI1 will immediately stop all operation instructions, including the rotation command being executed, and put the program pointer to the set position.

5 Introduction to Debug Tool

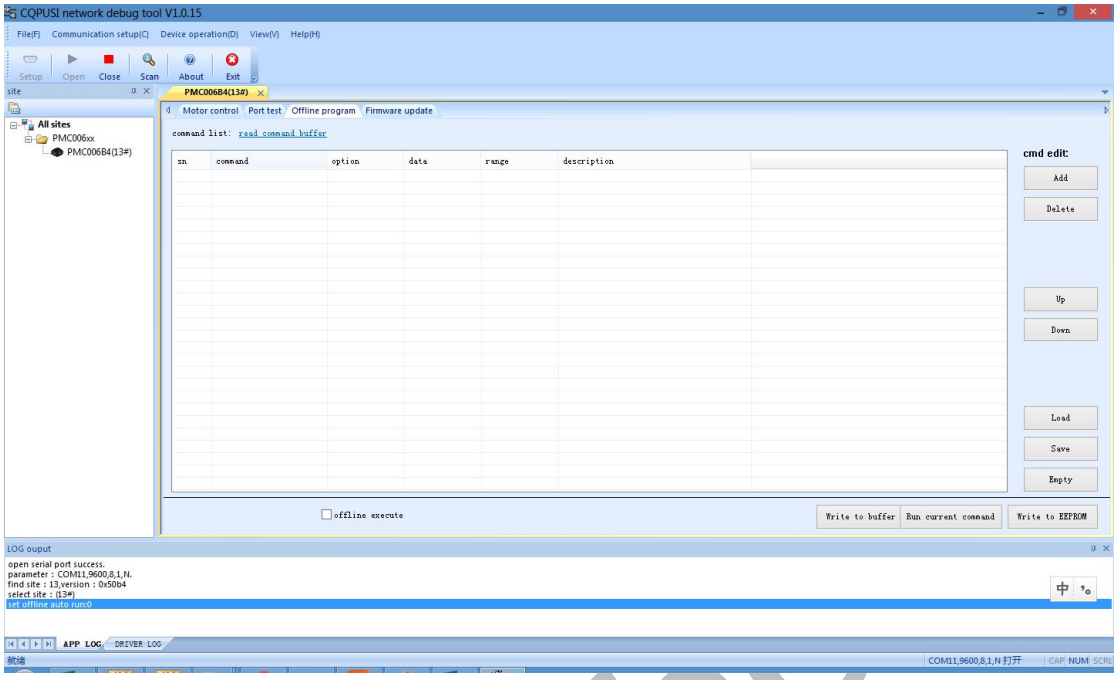
Users can use the CQPUSI tool software Tool Debug to set control parameters of motor, port detection, custom programming.

5.1.1 Main GUI



Firstly, according to System Settings, select the serial port number. And the port is configured to 9600, n, 8, 1 format, open the serial port. Then you can click the "Scan" button to scan the site. All sites connected to the bus will be display to the left of the window. User can double-click to select any of the sites to operate.

5.1.2 Custom Programming Interface



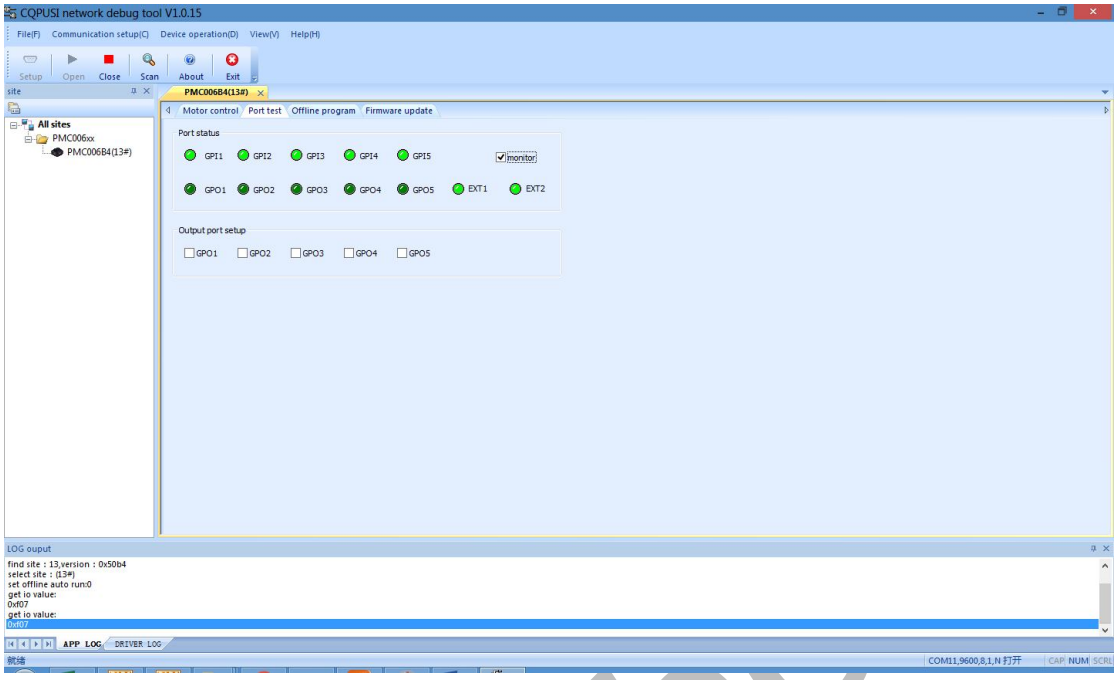
Click the "offline programming" button in the main interface to enter this interface. If there is already a user instruction in the PMC006B4 controller, click the "read command buffer", all instructions can be read and displayed automatically. Users can operate the instruction through the "insert", "delete", "up", "down" and other buttons.

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 PMC006xx Controller, and then press the "try run current command" button, and then debug instruction pointed by the cursor. After confirmation, press the "Write to EEPROM" button to burn all program to non-volatile memory. If you select the "offline exec", PMC006xx controller will automatically run the program which has been already burned next time power is on.

After successful commissioning instruction, you can press the "Save" button to save the instruction to your local disk, so that when the batch burn is needed, users can only need to click the "Open" button to read the program stored in disk, then press "write instruction to the buffer" and "Write to EEPROM" button to download the program to the controllers.

Note: Once the user enters the offline programming interface, it will be prohibited that user defined program is automatically executed even if there is no operation in the interface. Therefore, when user leaves this interface, need to click "off line execute" button to make it open. Otherwise, the user defined program will not be automatically run next time controller powers on.

5.1.3 Port test interface



The port test interface is used to debug the IO port of PMC006B4 in real time. After clicking the "state monitor" button, the indicator will display the current port state in real time. After the abolition of the “state monitor”, you can assign the level of output port.

5.1.4 Dynamic link library

CQPUSI tool software Tool Debug contains a dynamic link library for the communication interface, which can be used for the development of the host computer customized program. The DLL uses a very simple interface function, including the serial port initialization, sending and receiving of command and data, and other basic functions. In order to facilitate further debugging, the DLL also provides real-time LOG records which is stored as date + time file format. For details, please refer to the "dynamic library usage note" in the software package.

6 Electrical Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Supply Power Voltage	Normal 25℃	8	24	36	V
Operation Temperature	12V DC	-20		85	℃
IO maximum current	source/sink current	0		20	mA
Each phase output current	Normal 25℃	0	2.5	3.75	A
IO low Voltage	12V DC	-0.5		1.0	V
IO High Voltage	12V DC	3.0		5.5	V

7 Dimensions（Unit：mm）