USER MANUAL
PIO002xx series

Catalog

1 Introduction

1.1 Statement of intellectual property right

PIO002xx 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 PIO002xx 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

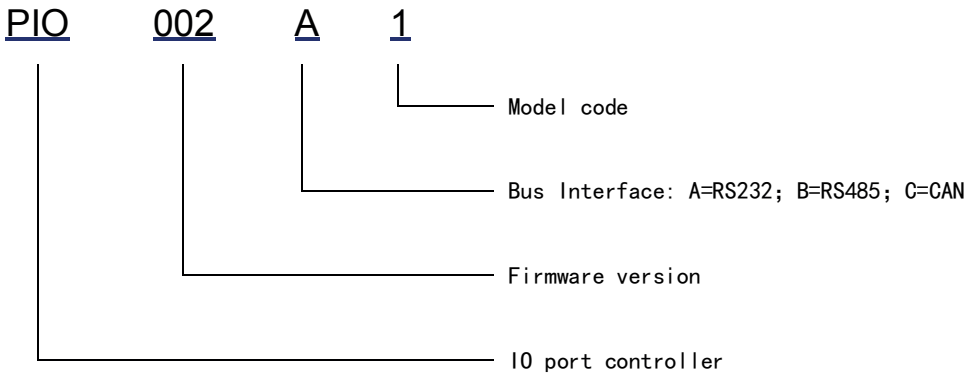
PIO002xx is a kind of programmable general purpose IO controller, which can be stacked installation with PMC006xx stepper motor controller directly. The series controller provides a variety of method of connection and control with general IO interface, encoder, DC motor and solenoid valve. The PIO002xx controller can be used to extend the 12 general-purpose IO ports and 14 channels for large current output ports. The PIO002xx provides simple, rich control command set, which can be connected with host computer through the UART/RS485/CAN, and which can automatically run the user defined program in offline mode.

2.2 Features

- ✓ 9-36V wide range of single voltage supply
- ✓ 12 real-time configurable digital ports
- ✓ 2 analog ports
- ✓ 4 PWM large current drive ports which can be configured in real-time
- ✓ 10 general large current drive ports which can be configured in real-time
- ✓ Five digital tubes display which can be configurable by command in real time
- ✓ Encoder connection is supported
- ✓ Open-loop control function of step motor
- ✓ Saving and reading user data is supported
- ✓ Miniature size: 44mmx44mmx15.5mm
- ✓ Firmware upgrade online without teardown
- ✓ Precision aluminum shell, conducive to the protection and heat dissipation
- ✓ User defined programming which can be configured running automatically in offline mode
- ✓ 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 product number in following format when ordering PIO002xx:



3 Connector Description

3.1 Terminal port location

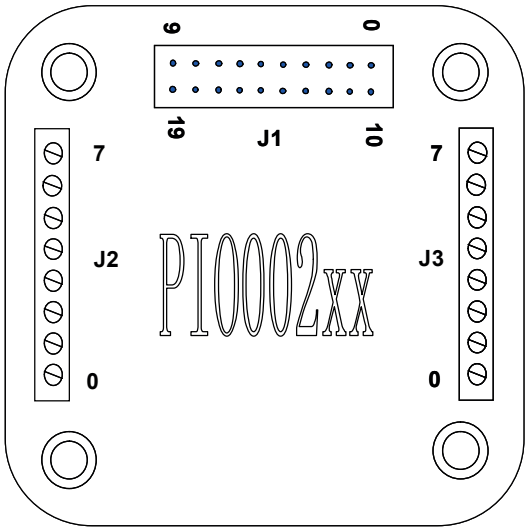


Figure 3-1

3.2 Large current output port J2

Pin:	0	1	2	3	4	5	6	7
Designator:	DRV0	DRV1	DRV2	DRV3	DRV4	DRV5	DRV6	VCC

Description:

VCC: Supply voltage, 8~36V;

DRV0~DRV3: PWM adjustable large current port, and DRV2 is reused for step motor pulse output;

DRV4~DRV6: General large current drive port, and DRV4 is reused for step motor direction output, and DRV5 is reused for step motor enable control.

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

3.3 Large current output port J3

Pin:	0	1	2	3	4	5	6	7
Designator:	GND	DRV13	DRV12	DRV11	DRV10	DRV9	DRV8	DRV7

Description:

GND: DC power cathode;

DRV7~DRV13: general large current drive port.

3.4 Signal ports J1

Pin:	0	1	2	3	4	5	6	7	8	9
Designator:	AN0	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	AN1	RXD	TXD
Pin:	10	11	12	13	14	15	16	17	18	19
Designator:	DVDD	GND	GPIO6	GPIO7	GPIO8	NC	FSET	GPIO10	GPIO11	GPIO12

Description:

GPIO0~12: general digital input/output ports 0-12

AN00~1: analog input ports

NC: reserved bit, which did not connect to any peripherals;

DVDD: controller +5V output

GND: digital ground of controller

GPIO6: being reused as EXT2

GPIO7: being reused as EXT1

FSET: factory reset pin, which did not connect to any peripherals

TXD: RS232/RS485 bus transmitting signal or connecting CAN transceiver modules

RXD: RS232/RS485 bus receiving signal or connecting CAN transceiver modules

Warning: ALL the signal ports voltages must be between -0.3V~+5.3V except TXD and RXD.

3.5 Standalone Operation

The figure below shows the connection way of the PIO002A1 controller in standalone operation. The device default address of each PIO002xx controller is 0xff. The standalone connection way must be used for the first time to burn controller device address. J1 port of this controller uses 20p 1.0 Pitch SHD connector, users can design their own plug connector, and can also modify according to the PIO002xx controller distribution cable. The GND of the RS232 cable should be connected to the GND of the controller in the following diagram.

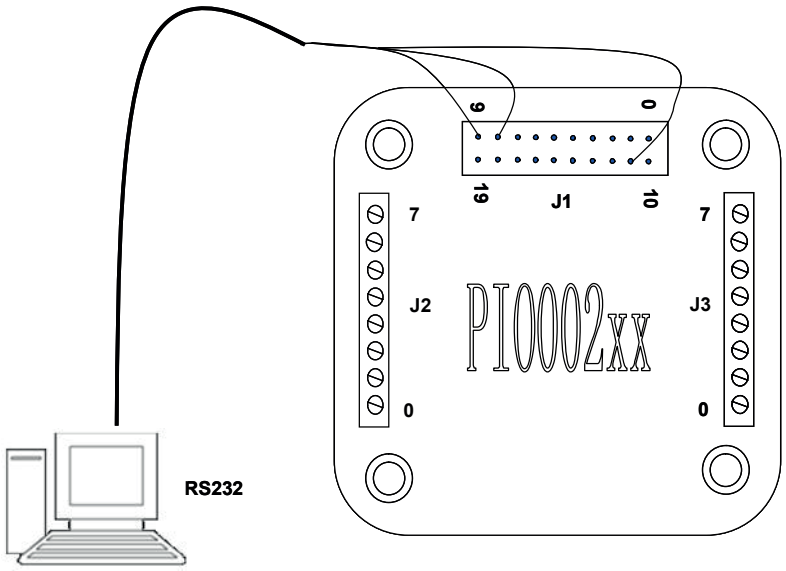


Figure 3-2

WARNING: Live plug is forbidden. Otherwise, the controller will be damaged permanently.

3.6 RS485 Network Operation

It provides a network scheme which uses RS485 bus to connect multiple PIO002xx controllers and PMC006xx 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 pull-up and pull-down resistance in the A, B line, to avoid this uncertainty.

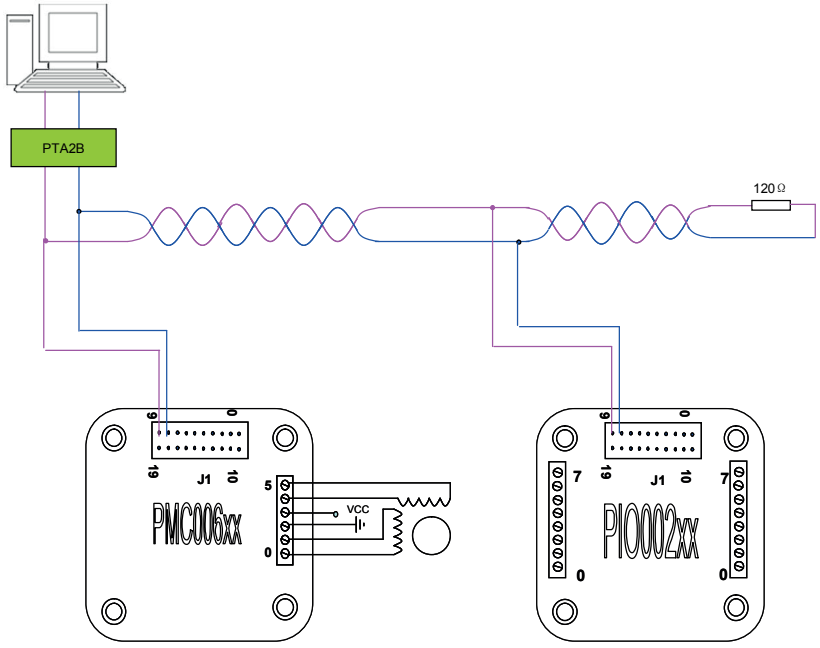


Figure 3-3

For more details about the module PTA2B used to achieve RS232 to RS485 conversion, please contact the sales staff.

3.7 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 PIO002xx controllers and PMC006xx controllers in the figure 3-3, which are compatible with CAN2.0A and CAN2.0B two technical specifications, and can be connected up to 100 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. Meanwhile, the host CAN converter PTA2C and CAN transceiver cable PTA2C of each node must be common-grounded.

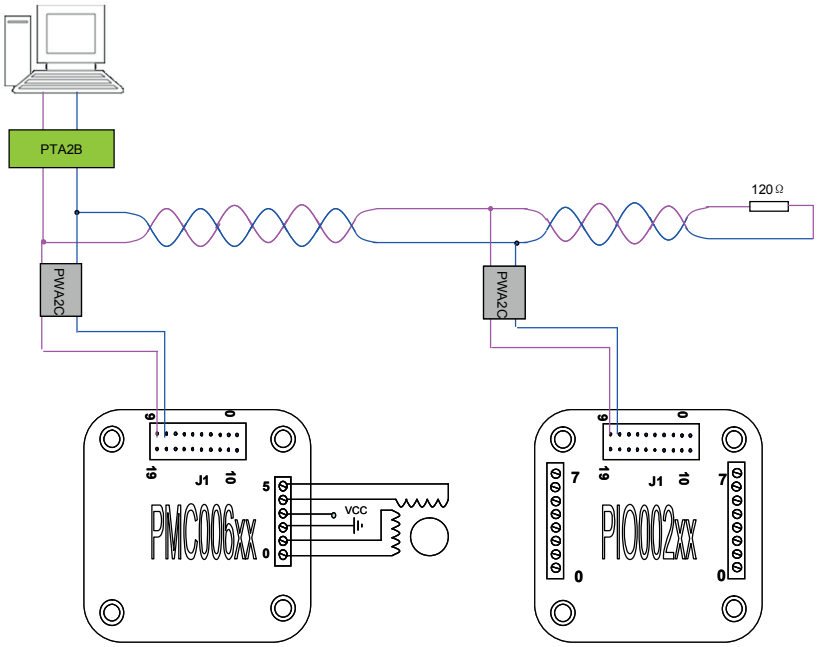


Figure 3-4

About RS232 to CAN module PTA2C and CAN transceiver cable PWA2C details, please contact the sales staff.

3.8 Emergency Switch Connection

When enable the step motor control function, the pin Ext1 and Ext2 can be used to connect the external limit switch. The trigger mode of each pin can be chosen by command in real time, and the default mode is falling edge trigger. The user also can control any one of limit switches to be opened or closed by command.

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 is not enabled. In order to make the input port clamp to low level reliably when there is no trigger, it need to connect an external pull-down resistor, as shown in figure 3-5 right.

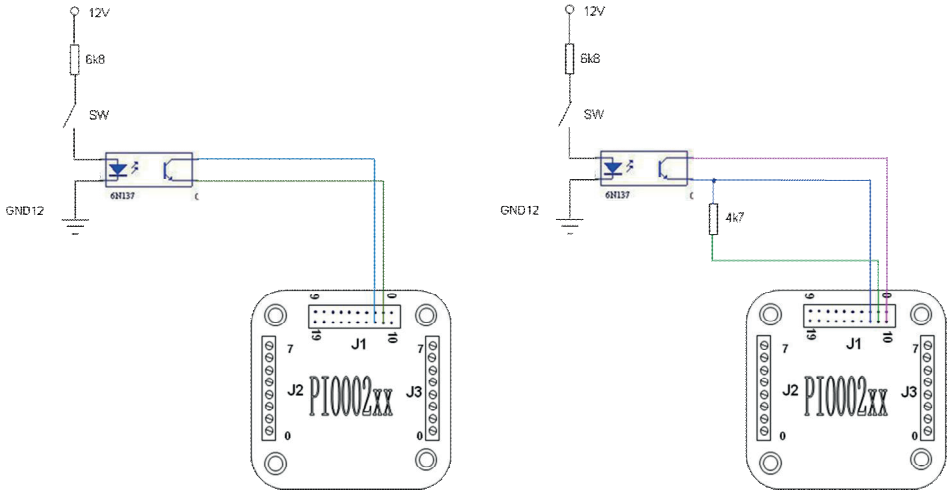


Figure 3-5

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.

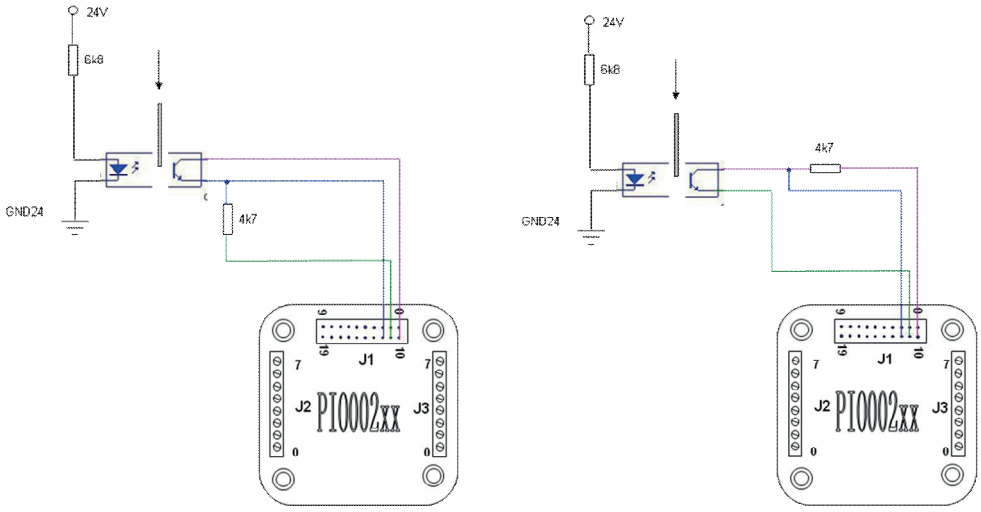


Figure 3-6

3.9 General digital IO port connection

There are 13 general digital IO ports in the PI0002xx controller. Each port can be configured as input or output by command in real time. with the input and output direction. When the port is configured as output, the output level can be configured in real time. If +5V output of the controller and the GND5 are used as external logic power, the port can be directly connected to the external logic power in the case of not exceeding the rated voltage (-0.5V~+5.5V). If the external logic uses different power, it is recommended using a opticalcoupler to connect, as shown in the following figure.

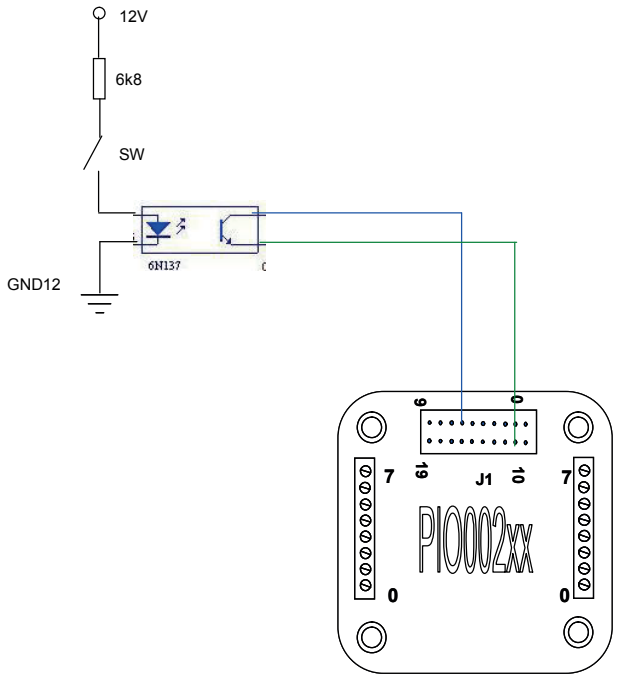


Figure 3-5

3.10 Analog IO port connection

There is a total of 2 analog input ports in the PIO002xx controller, which can be connected with two analog input signals. User can select any one of them by command in real time to perform analog-to-digital conversion. The PIO002xx controller uses an internal 2.56V voltage as reference source, and the sampling frequency of the internal ADC is 125kHz. In the default state, the pull-up resistor of two analog input port is enabled. Only when the ADC conversion is started, the pull-up resistor of corresponding port is forbidden.

3.11 Digital tube connection

PIO002xx controller supports 5-bit digital tube display function. The function can enable and control the display of the data by command in real time. When set digital tube display function enabled, a total of 12 pins will be occupied, which includes AN0, GPIO0~5 and GPIO7/8/10/11/12. These pins cannot be configured for other purposes, otherwise it will cause disorder of display. The PIO002xx controller only supports 7 segment cathode 5-bit digital tube, and can realize the 5-bit digit display through multi bit mosaicking. The following is the connection method of 7 segment common cathode 5-bit digital tube.

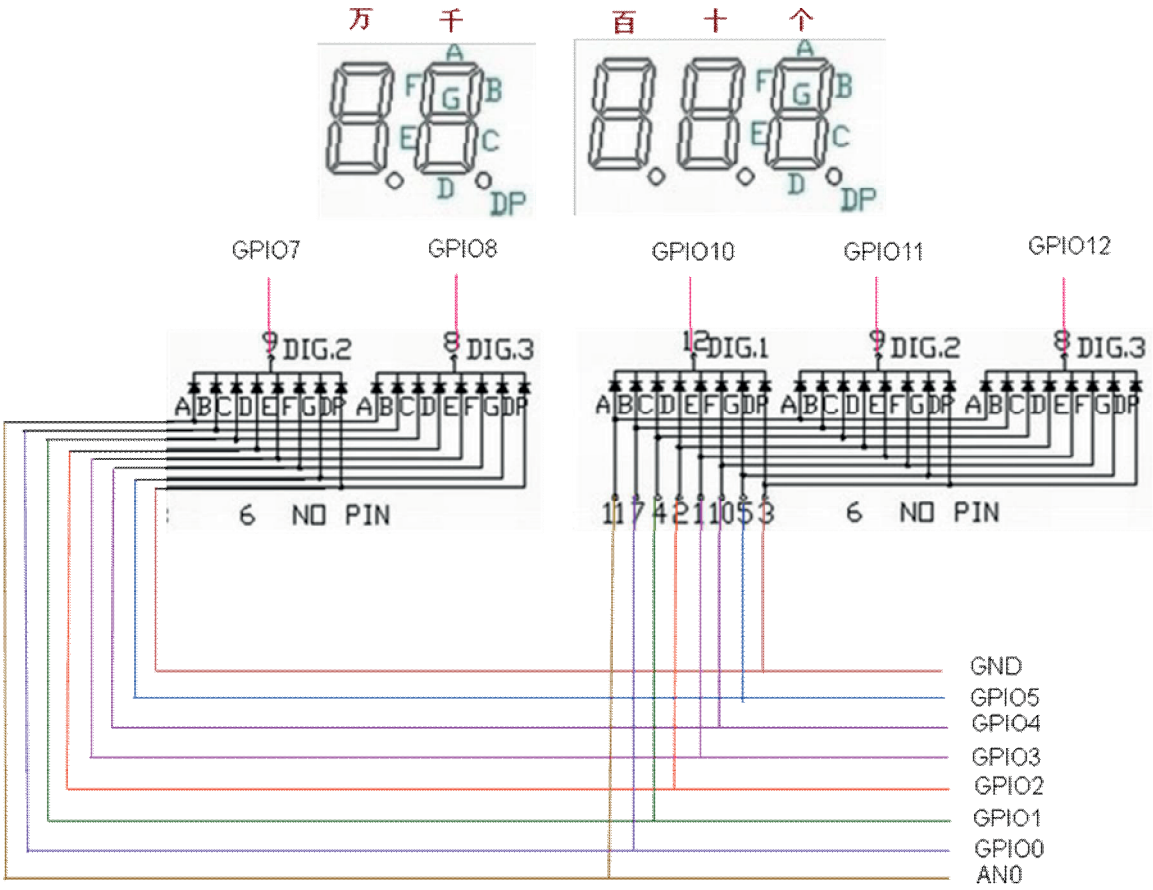


Figure 3-6

3.12 PWM large current drive port connection

PI0002xx can control 4 PWM adjustable large current drive port, which can be used to drive the DC motor within 36V, 500mA or other types of DC load. The switch magnitude of current of each port can be adjusted by command. The current regulation mode is open loop. The PI0002xx controller adjusts the load current by adjusting the PWM duty ratio of internal signal. The frequency of PWM is fixed to 31.25khz. PI0002xx controller output using open collector mode, and there is a fast diode used to connect the power supply and output inside. In this way, the device can be protected from being damaged by the inductive electromotive force of the external load. The schematic diagram of the load connection is as follows:

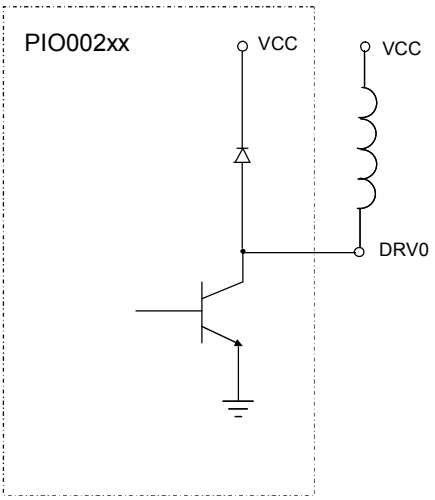


Figure 3-6

3.13 General large current drive port connection

PIO002xx can control 10 general large current drive ports, which can be used to drive the DC solenoid valve within 36V, 500mA or other types of DC load. The switch of each port can be adjusted by command. But the magnitude of current is determined by load condition, cannot be adjusted by command. PIO002xx controller output using open collector mode, and there is a fast diode used to connect the power supply and output inside. In this way, the device can be protected from being damaged by the inductive electromotive force of the external load. The connection mode is the same as Figure 3-10.

3.14 Rotary encoder connection

PIO002xx supports the connection of rotary encoder. The following figure shows connection mode of open collector output type encoder and the controller. The 5V power supply which is output by the controller and digital ground is respectively connected to the power supply and the ground of the encoder. GPIO0 is connected with the encoder B phase, and GPIO8 is connected with encoder A phase. After enabling the encoder function, the internal pull-up resistor of GPIO8 and GPIO0 is also automatically enabled, which can simplify the connection of external line.

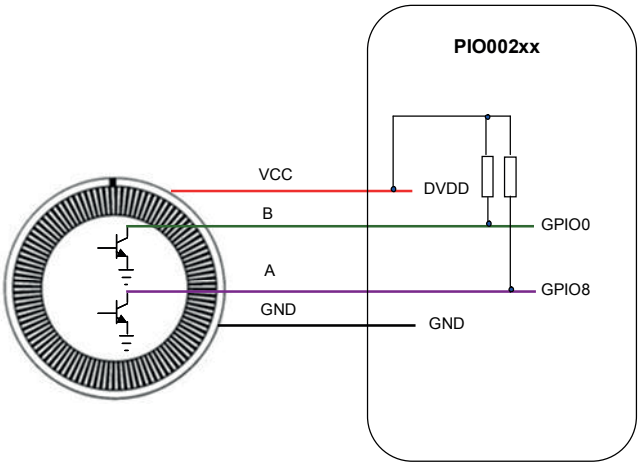


Figure 3-7

3.15 stepper motor driver connection

PIO002xx can control the stepper motor driver in open loop method. The following figure shows the connection method of PMD006 stepper motor driver and PIO002xx controller. After the stepper motor control function is enabled, DRV2 and DRV4 will be respectively used as the pulse and direction output function, while the DRV3 port also must remain suspended, and cannot be used for other purposes. For more details, please refer to 4.3 chapter.

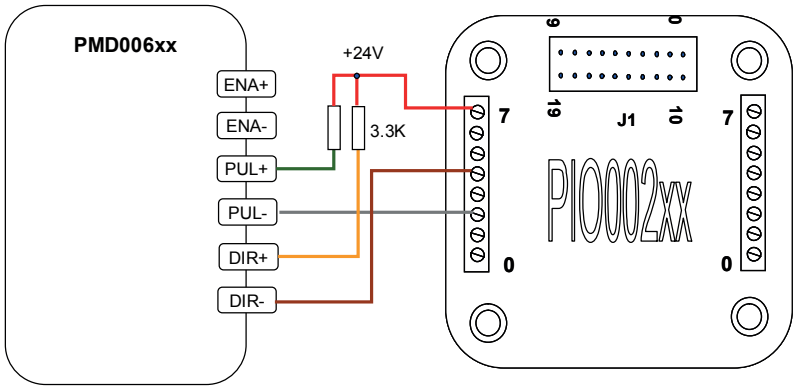


Figure 3-8

3.16 DC motor connection

PIO002xx can directly support the brushless DC motor, and realize the speed adjusting function. If you need to add the reversing function, you can achieve this with the help of two small solid relay, as shown in Figure 3-9.

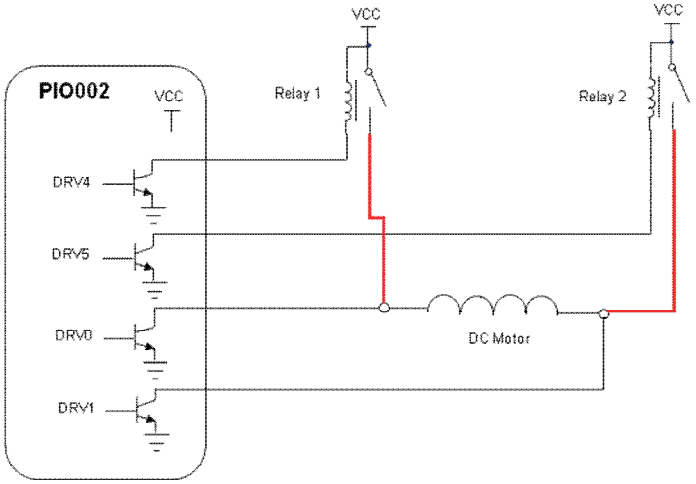


Figure 3-9

3.17 Analogue Adjusting Speed Connection

When the stepper motor drive is enabled, PIO002xx can connect with an external adjustable resistor to realize adjusting speed function, as shown in figure 3-10. When set analog adjusting speed enable in offline program, GPI1 pin will be used as analog input port to realize adjusting speed of motor dynamically.

The function of analog adjusting speed can be used when the PMC006xx controller is in offline mode. In this case, GPI1 pin is used as analog input port. For more details, please refer to chapter 4.3.

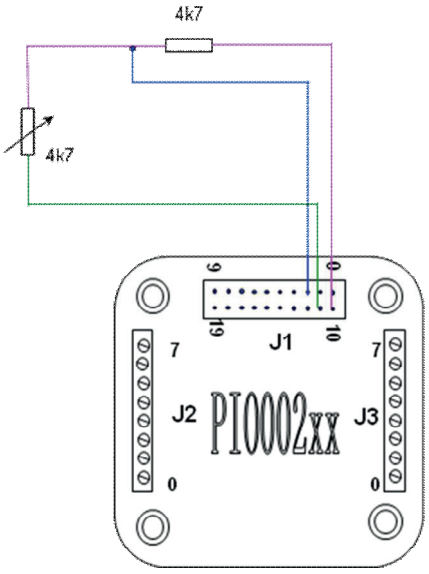


Figure 3-10

3.18 Factory Reset

When the PIO002xx controller performs a problematic user defined program, or when the user accidentally overwrites 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 a variety of power down storage parameters, but the user's custom program will be reserved for debugging analysis.

4 Instruction Set

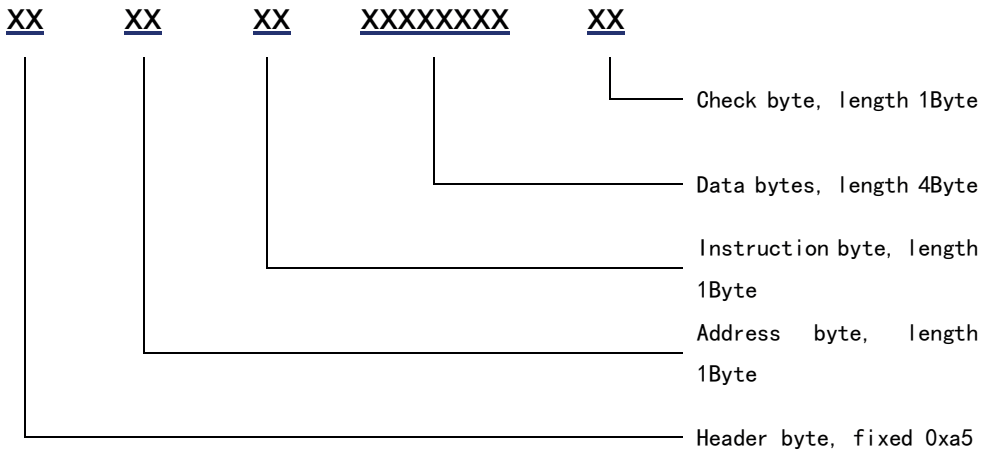
The PIO002xx controller uses a set of simplified instruction to regulate and simplify the operation of the host computer to the controller. The controller receives the operation instructions from the host computer, and returns the ACK to the host computer, and returns the data or the state of user's needs.

Communication mode between host computer and the PIO002xx 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. PIO002xx controllers can't communicate with each other.

4.1 Instruction Structure

The communication format between PIO002xx 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 PIO002xx 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 or 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 (even for different types of controllers). The user can modify the controller address in a single mode (the default address is 0xff); when the PIO002xx 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. For more details, please refer to Instruction Set Summary in 4.2 section.

When the PIO002xx 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 PIO002xx 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 the host computer and controller, the data type of 4 bytes is defined as a signed integer (ANSC C: signed long) and the low byte 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 PIO002xx 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
R(0x52)	Read firmware version information	----	
w(0x77)	Configure the controller's address	0-120	③
t(0x74)	Save all parameters when power down	----	①
W(0x57)	Read Status Register 1	----	①
X(0x58)	Read Status Register 2	----	①
U(0x55)	Configure the direction of digital port	0x0-0x1fff	
Q(0x51)	Read the direction of digital port	----	
y(0x79)	Configure the value of digital port	0x0-0x1fff	
x(0x78)	Read the value of digital port	----	
K(0x4b)	Start analog input conversion	0~3	①
P(0x50)	Stop continuous analog input conversion	----	①
L(0x4c)	Read analog conversion value	0x0-0xffff	
M(0x4d)	Enable general high current output port control	0x0-0x3fff	Valid only for common ports
N(0x4e)	Adjustable current output port PWM0~3 control	0x0-0x300ff	①

V(0x56)	Read the general high current output port	----	①
O(0x4f)	Read large current output port PWM	----	
F(0x46)	Set offline execution automatically	0/1	③
S(0x53)	Set the digital tube display function	0/1	
T(0x54)	Display the specified number in the digital tube	0-99999	①
a(0x61)	Set user data address	0x0-0xff	③
b(0x62)	Read data address of current user	----	
c(0x63)	Write user data	0x0-0xff	③
d(0x64)	Read user data	----	
e(0x65)	Set stepper motor control enabled	0/1	①, ③
f(0x66)	Read and write the max speed of stepper motor	65-40000pps	①
g(0x67)	Read and write the acceleration coefficient of stepper motor	0-5	①, ③
h(0x68)	Read and write the deceleration coefficient of stepper motor.	0-5	① , ③
i(0x69)	Set direction of rotation	0/1	
j(0x6a)	Set rotation steps	0-0x7fffffff	①
n(0x6e)	Read and write start speed of stepper motor	1-16000pps	①, ③
o(0x6f)	Read and write stop speed of stepper motor	1-16000pps	① , ③
p(0x70)	Read current position of stepper motor	----	
u(0x75)	Set current position stepper motor	0-0x7fffffff	②
q(0x71)	Enter or exit speed mode	0/1	
r(0x72)	Read and write external trigger mode	0-3	①, ③
s(0x73)	Enable the read and write of external trigger	0-3	①, ③
Y(0x59)	Clear external stop status bit	1-3	②
k(0x6b)	Set encoder enable	0/1	①, ③

m(0x6d)	Read encoder position	----	
l(0x6c)	Reset encoder position	0	

Remark:

① : Compared with the previous firmware 0x03bb version, the instruction is modified;

② : Compared with the previous firmware 0x03bb version, the instruction is added;

③ : The parameter can be saved when power down.

4.3 Instruction Details

The software package provided by PUSI includes the communication interface DLL based on vcc++ and related examples. The user's program can operate the 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 The configuration of digital port direction

The configuration command of digital port direction, which is 0x55, is used to set the direction of digital ports. The valid bit-width of data is 12, which is from 0 to 12, respectively corresponding to GPIO0~GPIO12. When the bit level is set to 1, the direction of corresponding port is output, and when the bit level is set to 0, the direction of corresponding port is input. When the port direction is set as input, the internal pull-up is automatically enabled, and the input level is default high. In this case, the port will output current when it is pulled down by the external. When the port direction is set as output, there is no enabled internal pull-up and pull-down, and the output level is default low, and the port will absorb the external current. If output level is set high, it will output current.

Note: ①The direction of analog ports is always input, and is not affected by the command;

Host computer send: 0xa5 0x01 0x55 0x07 0x00 0x00 0x00 0x02

Controller ACK: 0xa5 0x7a 0x01 0x07 0x00 0x00 0x00 0x27

Description: set the digital ports GPIO0~GPIO3 of the controller whose address is 0x01 as output.

4.3.2 The configuration of digital port value

The configuration command of digital port value, which is 0x79, is used to set the value of digital ports. The valid bit-width of data is 12, which is from 0 to 12, respectively corresponding to GPIO0~GPIO12. When the bit level is set to 1, the corresponding port outputs high level, and when the bit level is set to 0, the corresponding port outputs low level. This command is valid only for the ports whose port direction is output.

Host computer send:0xa5 0x02 0x79 0x03 0x00 0x00 0x00 0x23

Controller ACK: 0xa5 0x7a 0x02 0x03 0x00 0x00 0x00 0x24

Description: Digital ports GPIO0~GPIO1 of the controller whose address is 0x02 output high level.

4.3.3 Boot analog input port

Any one of the two analog inputs of PIO002xx controller can be configured to perform analog-to-digital conversion. When starting the analog input through the 0x4b command. It indicates AN1 is analog input when bit0 of the data is 1, and it indicates AN0 is analog input when bit0 of the data is 0. It represents continuous conversion when bit1 of the data is 1, and It represents single conversion when bit1 of the data is 0. Since the time of a single conversion is microsecond, the host computer can read the result of the conversion without waiting. 0x50 command can be used to stop continuous analog conversion.

4.3.4 General high current port enabled control

Using the 0x4d command, user can open and close any of the DRV4~DRV13 ports. When the command is sent, the corresponding bit in the data controls the corresponding port, and the low four bits of the data bits are reserved bits, which cannot be used. For example: if the data 0x10 is sent, DRV4 will be opened; if the data 0x20 is sent, DRV5 will be opened.

Note: ①The DRV0~DRV3 ports are not controlled by 0x4d command; ②When the stepper motor control is enabled, DRV4 is no longer controlled by 0x4d command.

4.3.5 Adjustable high current port PWM control

DRV0~DRV3 ports of the PIO002xx are large current output ports whose PWM can be adjusted. Each port can be individually adjusted, and the scope of adjustment is 0-255. When set to 0, no current output, and when set to 255, output 100% load current. DRV0~DRV3 ports are controlled by 0x4e command. The third byte of data represents the port number, whose range is 0~3. The valid bits of PWM only take up the low 8 bits of data.

Note: ①DRV0~DRV3 ports are not controlled by the 0x4d command. When power is on, the PWM value of the four ports is 0, that is, there is no output; ②When the stepper motor control is enabled, DRV1~2 is no longer controlled by the 0x4e command.

4.3.6 Digital tube display control

The digital tube display function of PIO002xx controller will occupy 12 general IO ports, which can be opened and closed by using the 0x53 command. Once the function is opened, the relevant GPIO ports immediately switch to digital tube function, and at this time the data can be displayed by the 0x54 command in real time. After the function is closed, the GPIO return to the default state.

4.3.7 User data access

PI0002xx controller supports user data access function, the user can achieve data access of maximum 255 bytes through the 0x61~0x64 command. Every time before using the 0x63 command to write data, firstly, set the current data address through the 0x61 command. The controller will not automatically carry out data accumulation.

4.3.8 Encoder position read

When the function of the encoder is enabled by 0x6b command, the encoder position can be read by 0x6d in real time. The location information is represented by a signed 32-bit integer, and can be cleared by 0x6c command.

4.3.9 Analogue 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. In this application stepping motor driving must be enabled.

4.3.10 Stepper motor control

PI0002xx controller can connect with an external stepper motor driver module, and the highest frequency of output pulse is 40KHz, supporting S curve automatic acceleration and deceleration function. 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.

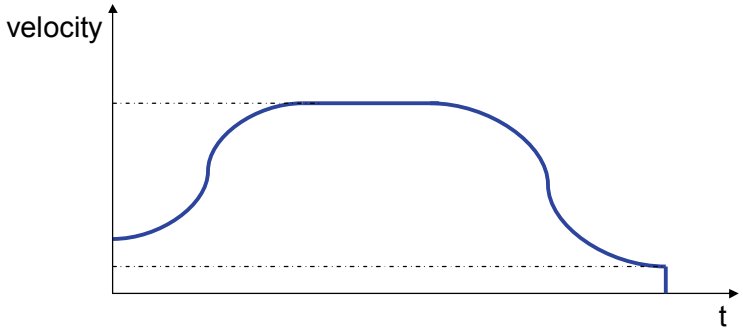


Figure 4-1

When the 0x6e and 0x6f instructions are used to read and write start speed and stop speed, it represents reading that the data is 0, and it represents the writing value that the data is 1~16000.

When the 0x67 and 0x68 instructions are used to read and write acceleration and deceleration coefficient, the value 0~5 is valid writing data. it represents reading when the data is greater than 10.

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 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.

4.3.11 External trigger control

PIO002xx controller supports 2 external trigger stop inputs, respectively, which multiplex with GPIO6 and GPIO7. User can read and write the corresponding trigger enabled and trigger mode through the 0x72~0x73 command. The valid data range of writing is 0~3. When the data is greater than 3, indicate reading. The 0x59 command is used to clear the external stop status bit, and writing 1 to the corresponding bit represents the cleanup. These three commands are using for operation to the two BIT bits, BIT0 corresponding to the external trigger 1, BIT1 corresponding to the external trigger 2.

Trigger mode: 0 represents the falling edge trigger, and 1 represents the rising edge trigger.

4.3.12 Query controller status

The PIO002xx controller's status is represented by two registers, which can be read by the command 0x57 and 0x58. 0x57 command is used to read the status register 1 of the controller. There is a total of 3 bits in the status byte, 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

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

Bit	Designator	Description
0	SM_EN	0: step control is not enabled; 1: step control is enabled.
1	EXT1_EN	0: disable emergency stop;1: enable emergency stop
2	EXT2_EN	0: disable emergency stop;1: enable emergency stop
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
6	ENC_EN	0: disable encoder; 1: enable encoder
7	ENC_CFG	0: configure encoder inverting;1: configure encoder forward
8	LED_EN	0: The digital tube is not enabled; 1: Digital tube enable

4.4 User-defined program

PIO002xx can be configured into offline mode. In this mode, controller automatically execute custom user code after power-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 and example.

When the PIO002xx controller works in offline mode, the UART\RS485\CAN communication interface is still responsive to the user's online instruction.

The maximum number of instructions that PIO002xx controller supports for user is 100.

4.4.1 User Instruction Set Summary

PIO002xx controller supports the following user defined instructions. These commands are automatically interacting with the controller through the tool software provided by PUSI. 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
WPORT	Set port value	1-13	0/1
DIR	Set the direction of digital port	1-13	0: input;1: output
CFG_AN	Configure analog input port	0	0x0-0x1
START_AN	Start analog input conversion	0	0/1
DRV_EN	Enable general high current port	1-10	0/1
PWM_SET	Set the PWM value of the large current port	1-4	0x0-0xff
CNTI	Internal counter plus 1	0	----
CNTC	Internal counter reset	0	----
JMP	Unconditional jump	0	0-100
JNE	Unequal jump	0	0-100
JEQ	Equal jump	0	0-100
WAIT	Waiting condition	1-15	0/1
CMP	Comparison	1-15	0~65535
DT_EN	Enable digital tube control	0	0/1
DT_DATA	Digital tube display data	1-2	0-65535
STOP_EN	Enable AN1 reset stop function	0	0-1
PAUSE_EN	Enable AN0 start/pause	0	0-1
SM_SPEED	Set the max speed of stepper motor	0	0-40000
ACCEL	Set acceleration of stepping motor	0	0-5
DECEL	Set deceleration of stepping motor	0	0-5
START_SPEED	Set the start speed of the stepper motor	0	0-16000
END_SPEED	Set the stop speed of the stepper motor	0	0-16000
ROT_DIR	Set the direction of rotation of the stepper motor	0	0/1
ROT_STEP	Set the rotation steps of stepper motor	0	0-65535
V_MODE	Set speed mode	0	0/1

ENCODER_EN	Set encoder enabled	0	0/1
CL_ENCODER	Clear the position of encoder	0	----
EXT_TRIG_EN	Enable external trigger control	0	0-3
EXT_TRIG_CFG	Set external trigger mode	0	0-3
EXT_TRIG_CLR	Clear external trigger flag bit	0	0-3

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 JNE 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 16 options can be selected, please refer to the PUSI tool software for "custom programming" interface settings for details.

4.4.2.5 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. 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.

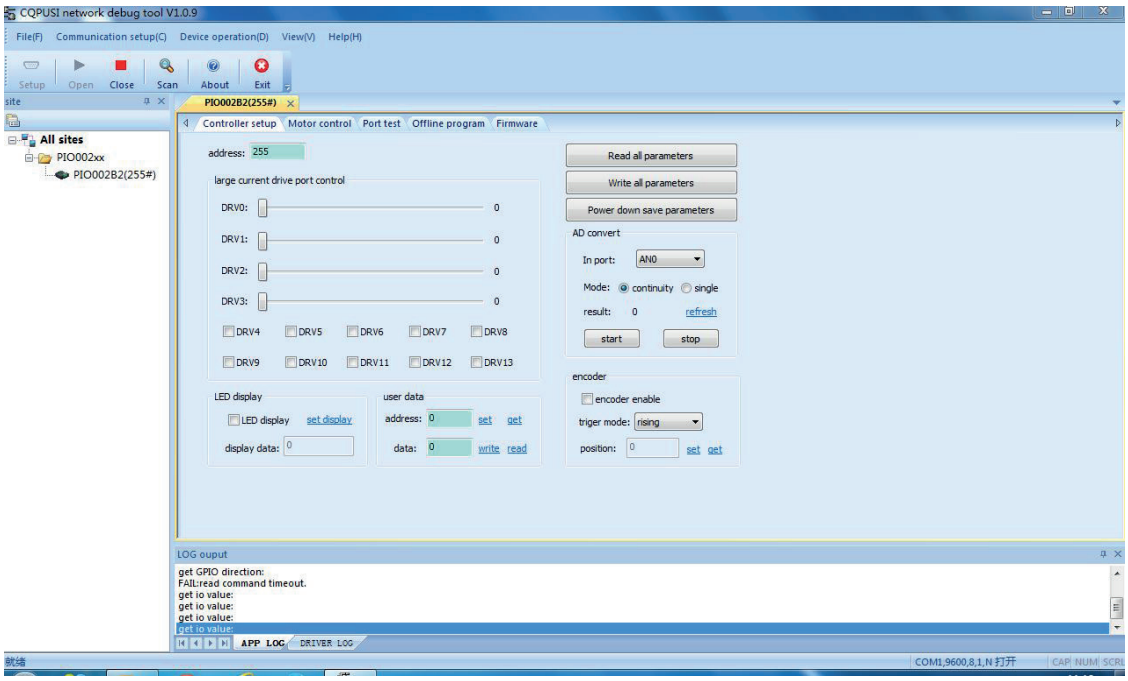
4.4.2.6 STOP_EN and PAUSE_EN command

The controller can choose to bind AN1 as the external reset stop key input, AN0 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 AN0 will alternately start or pause the execution of the offline program. When the external reset stop function is enabled, the low level pulse on the AN1 will immediately stop all operation instructions, and put the program pointer to the set position.

5 Introduction to Debug Tool

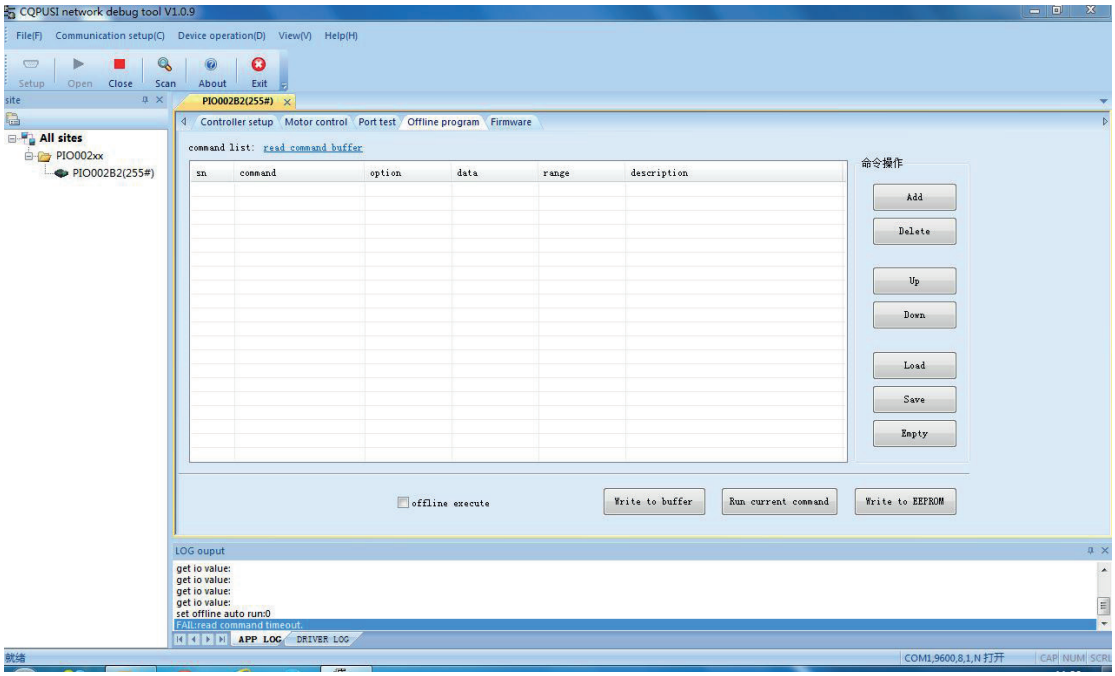
Users can use the PUSI tool software Tool Debug to debug command, IO port setting detection, custom programming.

5.1.1 Main GUI



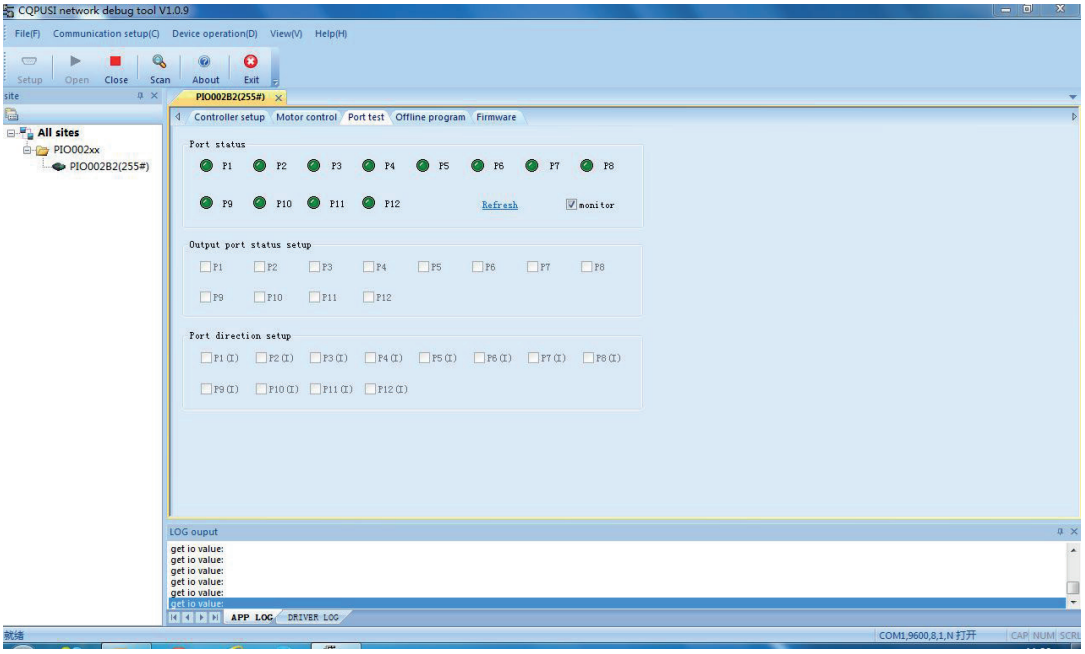
Firstly, according to System Settings, select the serial port number. And the port is configured to 9600, n, 8, 1 format, then open the serial port. User can click the “Scan” button to scan the site. User can double-click to select any of the sites which have been scanned to operate.

5.1.2 Custom Programming interface



Click the "offline programming" Tab in the main interface to enter this interface. If there is already a user instruction in the PIO002xx controller, all instructions will be read and displayed automatically in the list when the interface is activated. Users can operate the instruction through the "Add", "Delete", "Up", "Down" and other buttons. Once editing is completed, first press the "write to the buffer" button to download the program to on-chip memory of PIO002xx Controller, then press the "Run current command" button, and then user can debug the 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 execute", PIO002xx controller will automatically run the program which has been already burned next time power is on.

5.1.3 Port test interface



The Port test interface is used to debug the IO port of PIO002xx in real time. After clicking the "Port test" Tab, the indicator will display the current port state in real time. After abolition of the “state monitor”, You can click on the "Port direction" to configure the direction of the ports. When the direction of a port is configured to output, you can assign the level of this port.

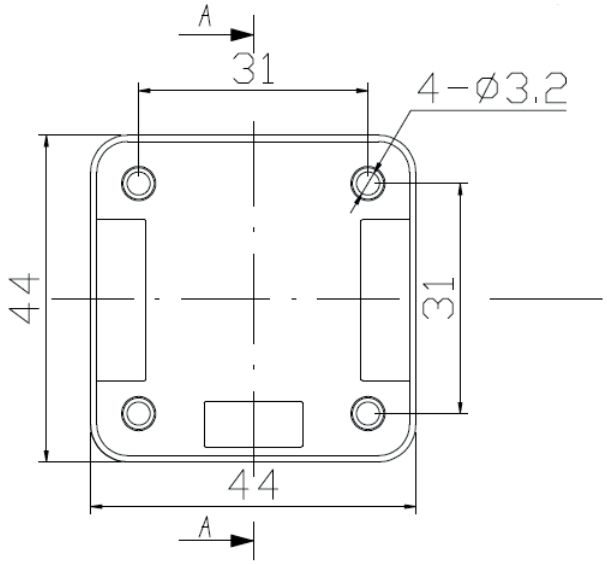
5.1.4 Dynamic link library

CQPUSI tool software Tool Debug contains a dynamic link library for packaging 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℃	9		36	V
Operation Temperature	12V DC	-10		55	℃
IO Maximum Current	Source/sink current	0		20	mA
Controller Maximum Current	Normal 25℃	0		3.5	A
IO Low voltage	12V DC	-0.5		1.0	V
IO High Voltage	12V DC	3.0		5.5	V
Maximum Current of Large Current Port	12V DC	0		500	mA

7 Dimensions（Unit：mm）



www.purelogic.ru

8 800 555-63-74 бесплатные звонки по РФ

Контакты

+7 (495) 505-63-74 - Москва
+7 (473) 204-51-56 - Воронеж
+7 (812) 425-17-35 - Санкт-Петербург
purelogic.ru

394033, Россия, г. Воронеж,
Ленинский пр-т, 160, офис 149

Пн-Чт: 8.00–17:00
Пт: 8.00–16.00
Перерыв: 12.30–13.30

info@purelogic.ru