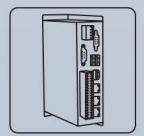
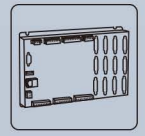


Vision Motion Controller

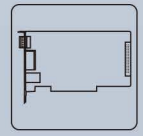
VPLC516E



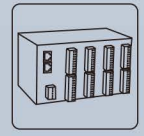
Vision Motion Controller



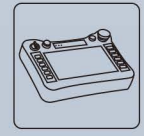
Motion Controller



Motion Control Card



Expansion Module



HMI



Foreword

Zmotion[®]

The motion controller provides rich interface, and it has excellent motion control performance, which can meet the expansion requirements of various projects.

This manual is copyrighted by Shenzhen Technology Co., Ltd., without the written permission of the Zmotion Technology, no person shall reproduce, translate and copy any content in this manual. The above-mentioned actions will constitute an infringement of the copyright of the company's manual, and Zmotion will investigate legal responsibility according to law.

For details about the ZMC controller software and the introduction and routine of each command, please refer to the ZBASIC software manual.

Information contained in this manual is only for reference. Due to improvements in design and functions and other aspects, Zmotion Technology reserves the final interpretation! Subject to change without notice!

Pay attention to safety when debugging the machine!

Please be sure to design an effective safety protection device in the machine, and add an error handling program in the software, otherwise Zmotion has no obligation or responsibility for the loss caused.

In order to ensure the safe, normal and effective use of the product, please be sure to read this product manual carefully before installing and using the product.



Safety Statement

- This chapter describes the safety precautions required for the correct use of this product. Before using this product, please read the instructions for use and correctly understand the relevant information on safety precautions.
- This product should be used in an environment that meets the design specifications, otherwise it may cause equipment damage or personal injury, and malfunctions or component damage caused by failure to comply with relevant regulations are not within the scope of product quality assurance.
- Zmotion will not take any legal responsibility for personal safety accidents and property losses caused by failure to comply with the contents of this manual or illegal operation of products.

Safety Level Definition

According to the level, it can be divided into " **Danger** " and " **Caution** ". Failure to operate as required may result in moderate injury, minor injury or equipment damage.

Please keep this guide in a safe place for reading when needed, and be sure to hand this manual to the end user.

Install	
 Danger	<ul style="list-style-type: none">◆ When the controller is disassembled, all external power supplies used by the system should be disconnected before operation, otherwise it may cause misoperation or damage to the equipment.◆ It is forbidden to use in the following places: places with dust, oil fume, conductive dust, corrosive gas and flammable gas; places exposed to high temperature, condensation, wind and rain; places with vibration and shock. Electric shock, fire and misuse can cause product damage and deterioration.
 Notice	<ul style="list-style-type: none">◆ Avoid metal shavings and wire ends falling into the hardware circuit board during installation.◆ After installation, ensure that there are no foreign objects on the hardware circuit board.◆ When installing, make it tightly and firmly with the mounting frame.

	<ul style="list-style-type: none"> ◆ Improper installation of the controller may result in misoperation, failure and fire.
Wiring	
 <p>Danger</p>	<ul style="list-style-type: none"> ◆ The specifications and installation methods of the external wiring of the equipment shall comply with the requirements of local power distribution regulations. ◆ When wiring, all external power supplies used by the system should be disconnected before operation. ◆ When powering on and running after the wiring work is completed, the terminals attached to the product must be installed. ◆ Cable terminals should be well insulated to ensure that the insulation distance between cables will not be reduced after the cables are installed on the terminal block.
 <p>Notice</p>	<ul style="list-style-type: none"> ◆ Avoid metal shavings and wire ends falling into the hardware circuit board during installation. ◆ The cable connection should be carried out correctly on the basis of confirming the type of the connected interface. ◆ It should be confirmed that the cables pressed into the terminals are in good contact. ◆ Do not bundle the control wires and communication cables with the main circuit or power supply wires, etc., and the distance between the wires should be more than 100 mm, otherwise noise may cause malfunction. ◆ If the controller is not installed properly, it may cause electric shock or equipment failure or malfunction.

Content

Chapter I Production Information.....	3
1.1. Product Information.....	3
1.2. Function Features.....	3
1.3. System Frame	5
1.4. Hardware Installment.....	5
Chapter II Product Specification	8
2.1. Basic Specification.....	8
2.2. Order Information.....	9
2.3. Interface Definition.....	9
2.4. Work Environment	10
Chapter III Wiring & Communication	12
3.1. Power Input, CAN Communication Interface	12
3.1.1. Power Specification.....	12
3.1.2. CAN Communication Specification & Wiring	12
3.1.3. Basic Usage Method	14
3.2. IN Digital Input & High-Speed Latch Port.....	15
3.2.1. Digital Input Specification & Wiring	17
3.2.2. Basic Usage Method	18
3.3. OUT Digital Output & PWM & Hardware Comparison Output & Single-ended Pulse 19	
3.3.2. Digital Output Specification & Wiring.....	20
3.3.3. Basic Usage Method	22
3.4. COM Serial Port	23
3.4.1. RS232/RS485 Communication Interface Specification & Wiring	24
3.4.2. Basic Usage Method	25
3.5. LAN Ethernet	26

3.6.	EtherCAT Bus Interface	27
3.7.	HDMI Interface	29
3.8.	USB Interface.....	30
3.9.	MPG Handwheel Interface	31
3.9.1.	MPG Handwheel Interface Specification & Wiring.....	32
3.9.2.	Basic Usage Method	34
Chapter IV Expansion Module		35
4.1.	CAN Bus Expansion.....	35
4.1.1.	CAN Bus Expansion Wiring.....	35
4.1.2.	CAN Bus Expansion Resource Mapping	37
4.2.	EtherCAT Bus Expansion	41
4.2.1.	EtherCAT Bus Expansion Wiring.....	41
4.2.2.	EtherCAT Bus Expansion Resource Mapping	43
Chapter V Program & Applications.....		45
5.1.	ZDevelop Software Usage.....	45
5.2.	PC Upper-Computer Program Application	50
Chapter VI Run and Maintain		53
6.1.	Regular Inspection and Maintenance	53
6.2.	Common Problems.....	54

Chapter I Production Information

1.1. Product Information

VPLC516E is a kind of bus type vision motion controller. The controller itself supports 16-axis at most, but can be extended to 32 axes to achieve all kinds of complex motion control, such as, electronica cam, linear, circular, continuous trajectory processing, robot, etc., and machine vision applications, positioning, measurement, detection, and identification.

VPLC516E motion controller can be applied in 3C electronics, lithium, printing and packaging equipment, medical equipment, robot, semiconductor, and laser.

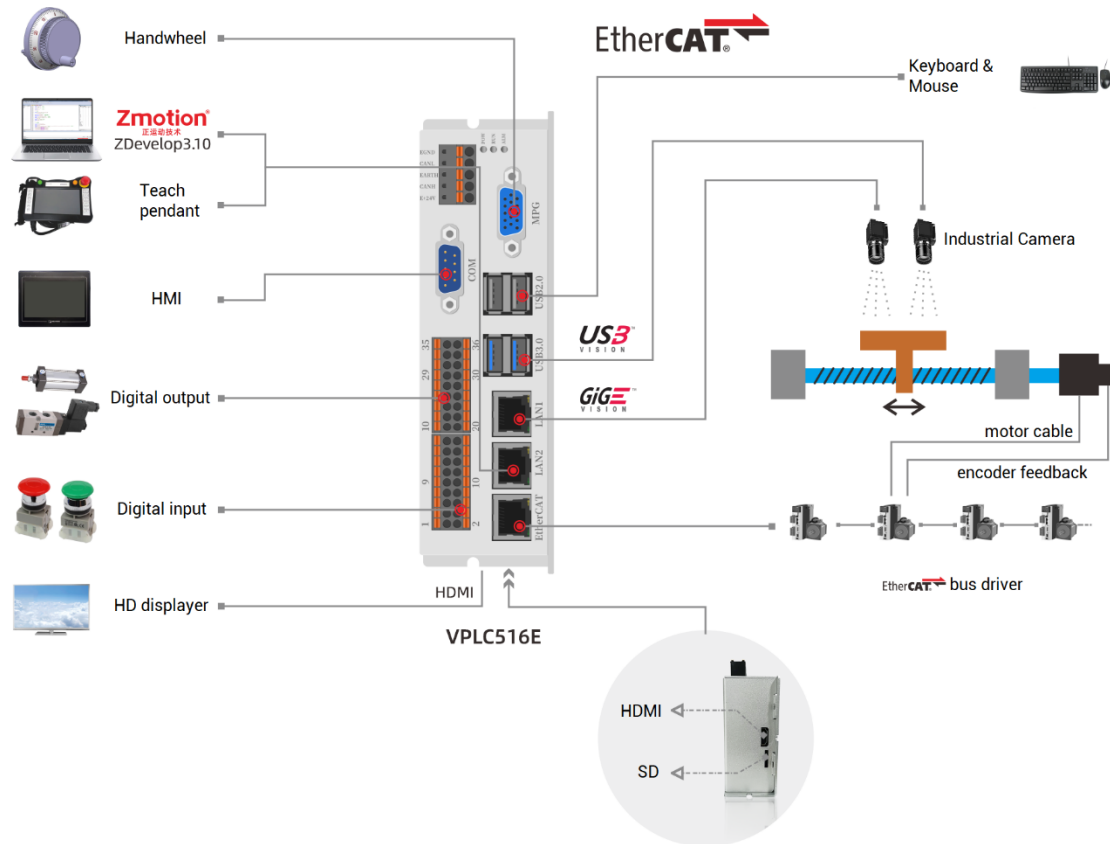
1.2. Function Features

- ◆ Support 16 axes motion control, which can be extended to 32 axes.
- ◆ Pulse output mode: pulse / direction.
- ◆ AXIS interface supports encoder position measurement, which can be configured as handwheel input mode.
- ◆ Specialized handwheel input interface.
- ◆ Maximum pulse output frequency of each axis is 500KHz.
- ◆ IO can be expanded through CAN and EtherCAT bus, 4096 isolated inputs and 4096 isolated outputs can be extended at most.
- ◆ There are 16 NPN general inputs, 2 are high-speed inputs, 14 are low-speed inputs, and high-speed inputs can be configured as latch signal.
- ◆ There are 16 NPN general outputs, 12 are high-speed outputs, 4 are low-speed outputs, and max output current can reach 300mA, which can drive some solenoid

valve directly.

- ◆ The maximum output current of general digital outputs can reach 300mA, which can directly drive some kinds of solenoid valves.
- ◆ Interfaces: 4 USB (2 are USB3.0), 1 COM (RS485/RS232), 2 1000M Ethernet, 1 100M EtherCAT.
- ◆ Support point to point, linear interpolation, circular interpolation, continuous interpolation, robot.
- ◆ Support electronic cam, electronic gear, position latch, synchronous follow, virtual axis, and other functions.
- ◆ Support hardware comparison output (HW_PSWITCH2), hardware timer, precision output when in motion.
- ◆ Support pulse closed loop, pitch compensation, etc.
- ◆ Support multi-task and multi-file programming in Basic.
- ◆ A variety of program encryption methods to protect the intellectual property rights of customers.
- ◆ Support power failure detection and power failure storage.

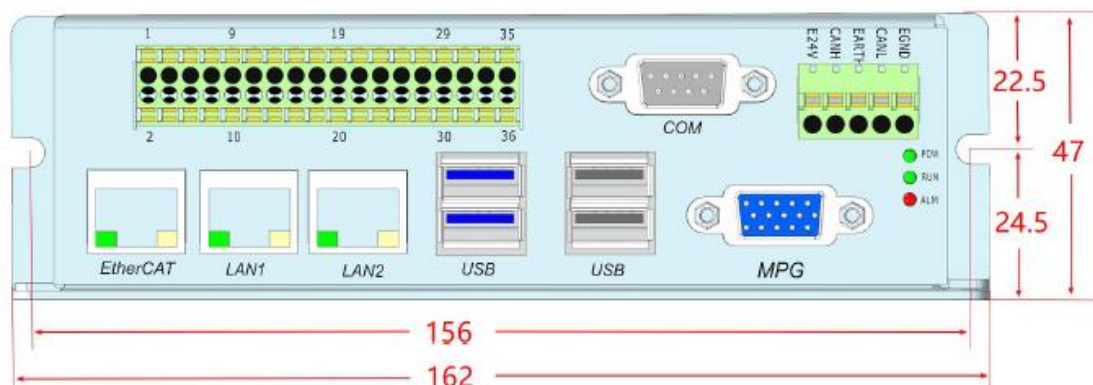
1.3. System Frame

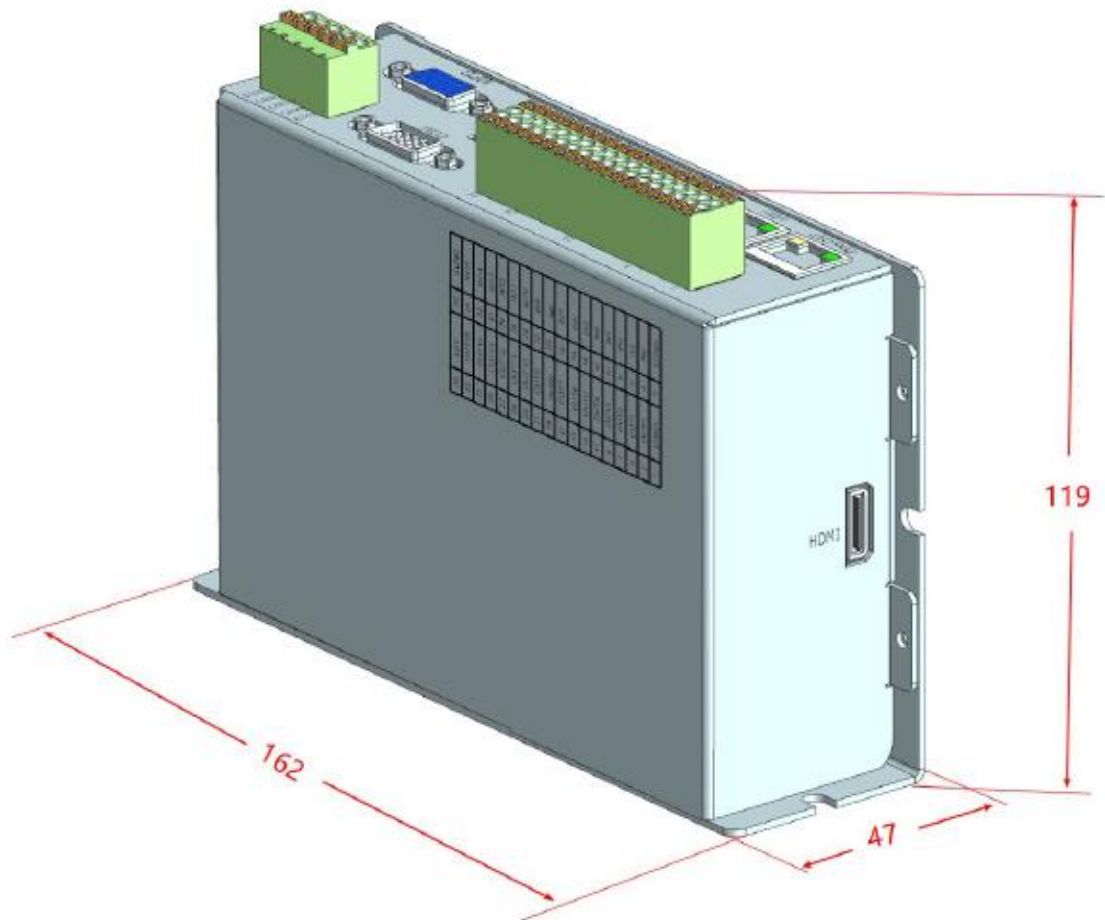
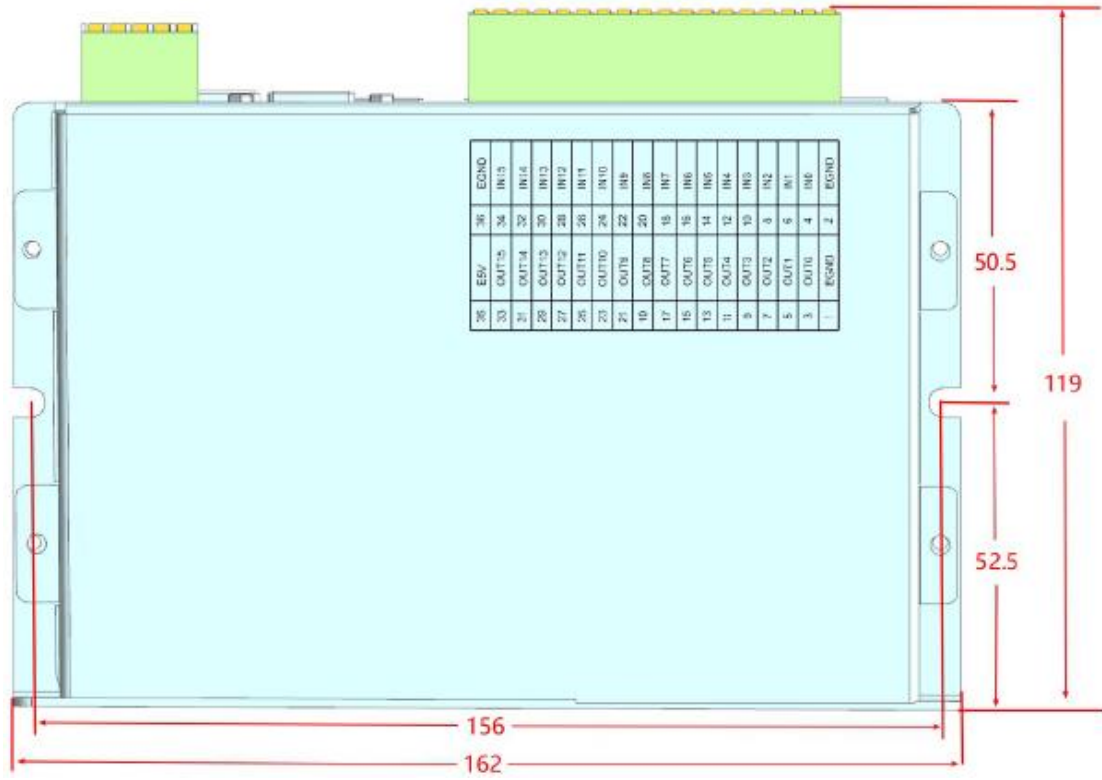


1.4. Hardware Installment

Fix the controller to the mounting surface through two M5 screws. When installing, please pay attention to the installation position. Please face the front of the controller (the actual installation surface of the operator) to the operator and make it perpendicular to the wall.

(Unit: mm, Mounting Hole Diameter 4.5mm)







**Installation
attention**

- Non-professionals are strictly prohibited to operate. Specifically, professionals who had been trained related electrical equipment, or who master electrical knowledge.
- Please be sure to read the product instruction manual and safety precautions carefully before installation.
- Before installation, please ensure that the product is powered off.
- Do not disassemble the module, otherwise the machine may be damaged.
- In order to facilitate ventilation and controller replacement, 2-3cm should be left between the upper and lower parts of the controller and the installation environment and surrounding components.
- Considering the convenient operation and maintenance of the controller, please **do not** install the controller in the following places:
 - a) places where the surrounding ambient temperature exceeds the range of -10°C-55°C
 - b) places where the ambient humidity exceeds the range of 10%-95% (non-condensing)
 - c) places with corrosive gases and flammable gases
 - d) places with many conductive powders such as dust and iron powder, oil mist, salt, and organic solvents
 - e) places with direct sunlight.

Chapter II Product Specification

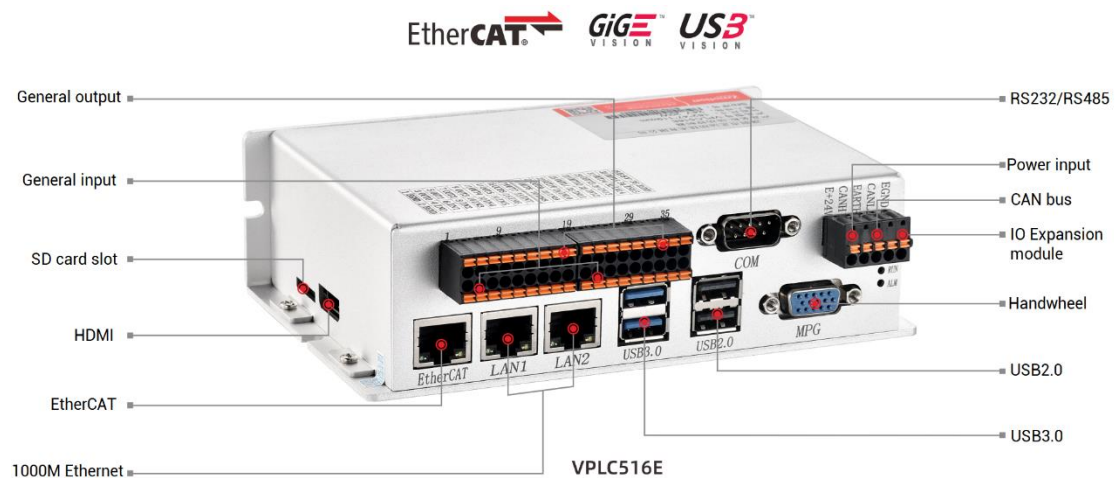
2.1. Basic Specification

Item	Description
Model	VPLC516E
Basic Axes	16
Max Extended Axes	32
Type of Basic Axes	Pulse axis / encoder axis / ECAT bus axis
Digital IO	16 inputs and 16 outputs.
Max Extended IO	4096 inputs and 4096 outputs.
PWM	2
AD/DA	0
Max Extended AD/DA	512 ADs, 512 DAs.
Pulse Bit	64
Encoder Bit	64
Speed and Acceleration Bit	64
Highest Pulse Frequency	500KHz
Motion Buffer of Each Axis	4096
Array Space	1280000
Program Space	64MByte
Flash Space	512MB
Power Supply Input	24V DC input
Communication Interfaces	RS232, RS485, USB, ETHERNET, CAN, EtherCAT
Dimension	162*47*119mm

2.2. Order Information

Model	Description
VPLC516E-4	4 axes, point to point, linear, circular, electronic cam, continuous trajectory motion, robot command.
VPLC516E-6	6 axes, point to point, linear, circular, electronic cam, continuous trajectory motion, robot command.
VPLC516E-8	8 axes, point to point, linear, circular, electronic cam, continuous trajectory motion, robot command.
VPLC516E	16 axes, point to point, linear, circular, electronic cam, continuous trajectory motion, robot command.

2.3. Interface Definition



→ Interface Description

Mark	Interface	Number	Description
POW	The led that indicates the current state.	1	Power state: it lights when power is conducted.
RUN		1	Run state: it lights when runs normally
ALM		1	Error state: it lights when runs incorrectly
RS232	RS232 (port 0) serial port	1	Use MODBUS_RTU protocol

RS485	RS485 (port 1) serial port	1	Use MODBUS_RTU protocol
EtherCAT	EtherCAT bus interface	1	EtherCAT bus interface, connect to EtherCAT bus drive and EtherCAT bus expansion module
ETHERNET	Network port	2	Use MODBUS_TCP protocol, expand the number of network ports through the interchanger, and the number of net port channels can be checked through "?*port" command, default IP address is 192.168.0.11
E+24V	Main power supply	1	24V DC power, it supplies the power for controller.
E5V	5V power supply output	1	It is used to provide PWM or used for common anode when single-ended axis expands.
USB	USB3.0/2.0	4	Compatible with USB2.0 and USB1.0 interfaces, they can be connected to external equipment, such as, camera, mouse, keyboard, U disk.
CAN	CAN bus interface	1	Connect to CAN expansion modules and CAN equipment of other standards.
IN	Digital IO input port	16	NPN type, the power is supplied by internal 24V power supply.
OUT	Digital IO output port	16	NPN Leakage type, the power is supplied by internal 24V power supply.
Axis	Pulse axis interface	16	Include differential pulse output and differential encoder input.
MPG	Handwheel	1	5-24V handwheel signa input

2.4. Work Environment

Item	Parameters
Work Temperature	-10℃-55℃
Work relative Humidity	10%-95% non-condensing

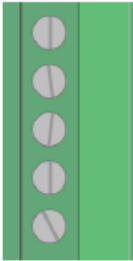
Storage Temperature		-40°C ~ 80°C (not frozen)
Storage Humidity		Below 90%RH (no frost)
vibration	Frequency	5-150Hz
	Displacement	3.5mm(directly install)(<9Hz)
	Acceleration	1g(directly install)(>9Hz)
	Direction	3 axial direction
Shock (collide)		15g, 11ms, half sinusoid, 3 axial direction
Degree of Protection		IP20

Chapter III Wiring & Communication

3.1. Power Input, CAN Communication Interface

The power supply input adopts a 5Pin screw-type pluggable wiring terminal, and the interval (means the gap distance between two ports) should be 3.81mm. This 5Pin terminal is shared by controller power and CAN communication.

→ **Terminal Definition:**

Terminal	Name	Type	Function
	EGND	Input	Power ground
	CANL	Input/output	CAN differential data -
	EARTH	Earthing	Protect, shield
	CANH	Input/output	CAN differential data +
	E+24V	Input	Power 24V input

3.1.1. Power Specification

→ **Specification**

Item	Description
Voltage	DC24V (-5%~5%)
The current to open	≤1A
The current to work	≤1.2A
Anti-reverse connection	YES
Overcurrent Protection	YES

3.1.2. CAN Communication Specification & Wiring

The CAN interface of the controller adopts the standard CAN communication

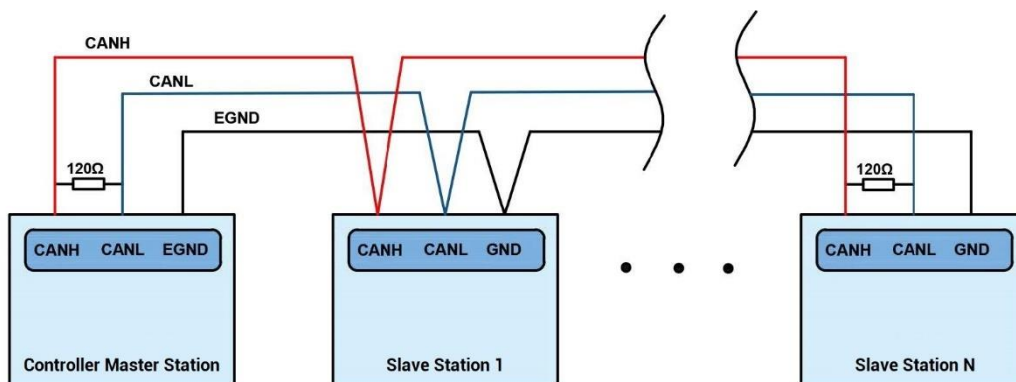
protocol, which mainly includes three ports, CANL, CANH and the public end. And it can connect CAN expansion modules and other standard CAN devices.

→ Specification

Item	CAN
Maximum Communication Rate (bps)	1M
Terminal Resistor	120Ω
Topological Structure	Daisy Chain Topology
The number of nodes can be extended	Up to 16
Communication Distance	The longer communication distance is, the lower communication rate is, and maximum of 100m is recommended.

→ Wiring Reference

Connect the CANL and CANH of the standard CAN module to the CANL and CANH of the other side correspondingly. And public ends of the CAN bus communication both parties are connected together. In CAN bus left and right sides, connect a 120Ω resistor respectively (please see below graphic).



→ Wiring Notes:

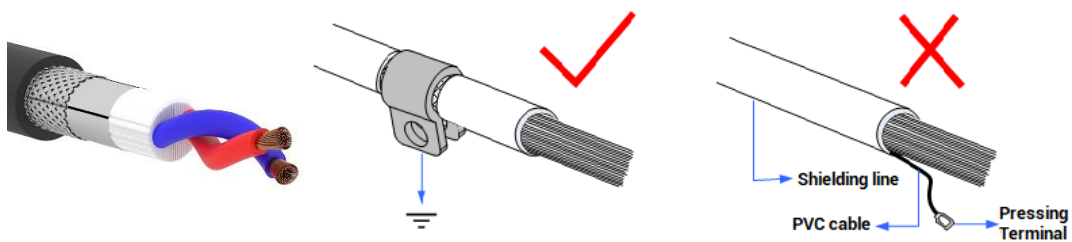
- As above, the daisy chain topology is used for wiring (the star topology structure cannot be used). When the use environment is ideal and there are no many nodes,

the branch structure also can be used.

- Please connect a 120Ω terminal resistor in parallel to each end of the CAN bus for matching the circuit impedance and ensuring communication stability.
- Please be sure to connect the public ends of each node on the CAN bus to prevent the CAN chip from burning out.
- Please use STP (Shielded Twisted Pair), especially in bad environments, and make sure the shielding layer is fully grounded.
- When on-site wiring, pay attention to make the distance between strong current and weak current, it is recommended for the distance to be more than 20cm.
- It should be noted that the equipment grounding (chassis) on the entire line must be good, and the grounding of the chassis should be connected to the standard factory ground pile.

→ Cable Requirements:

Shielded Twisted Pair, and the shielded cable is grounded.



3.1.3. Basic Usage Method

- (1) Please follow the above wiring instructions to wiring correctly.
- (2) After powered on, please use EtherNET or RS232 to connect to ZDevelop.
- (3) Please use the "CANIO_ADDRESS" command to set the master's "address" and

"speed" according to the needs, and use the "CANIO_ENABLE" command to enable or disable the internal CAN master function, or through "ZDevelop/Controller/State the Controller/Communication Info" to view the CAN status intuitively, and refer to the "ZBasic Programming Manual" for details.

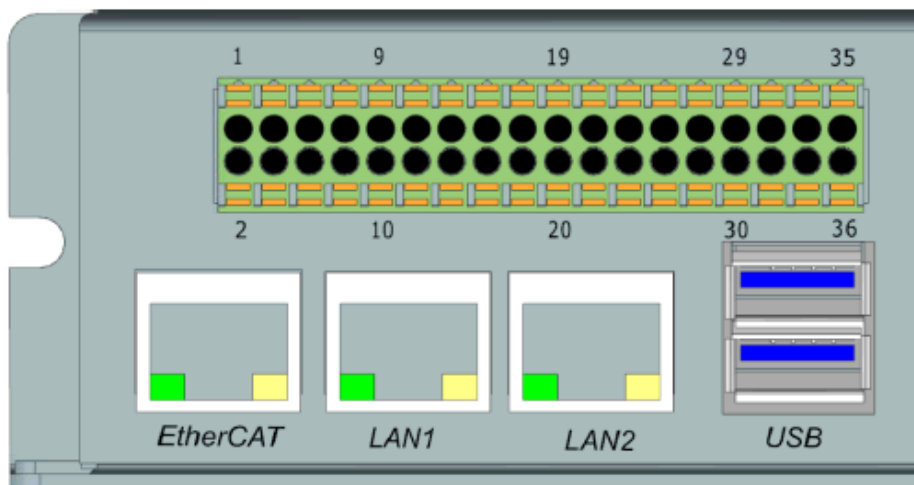
```
CAN communication settings:  
CANIO_ADDRESS = 32, CANIO_ENABLE = 1  
ZCAN Master  
CAN baud: 500KBPS  
CAN enable: ON  
  
Serial port configuration:  
Port0:(RS232) is ModbusSlave Mode.  
Address:1, variable:2 delay:400ms  
Baud:38400  
DataBits:8  
StopBits:1  
Parity:0  
Port1:(RS485) is ModbusSlave Mode.  
Address:1, variable:2 delay:400ms  
Baud:38400  
DataBits:8  
StopBits:1  
Parity:0
```

- (4) Correctly set the "address" and "speed" of the slave station expansion module according to the manual of the slave station.
- (5) After all the settings are completed, restart the power supply of all stations to establish communication.
- (6) Note that the "speed" settings of each node on the CAN bus must be consistent, and the "address" settings cannot cause conflicts, otherwise the "ALM" alarm light will be on, and the communication establishment will fail or the communication will be disordered.

3.2. IN Digital Input & High-Speed Latch Port

The digital input adopts screw-type pluggable terminals, and the gap distance between terminals should be 3.81mm. In addition, the high-speed latch function is

integrated in digital input signals.



→ Terminal Definition

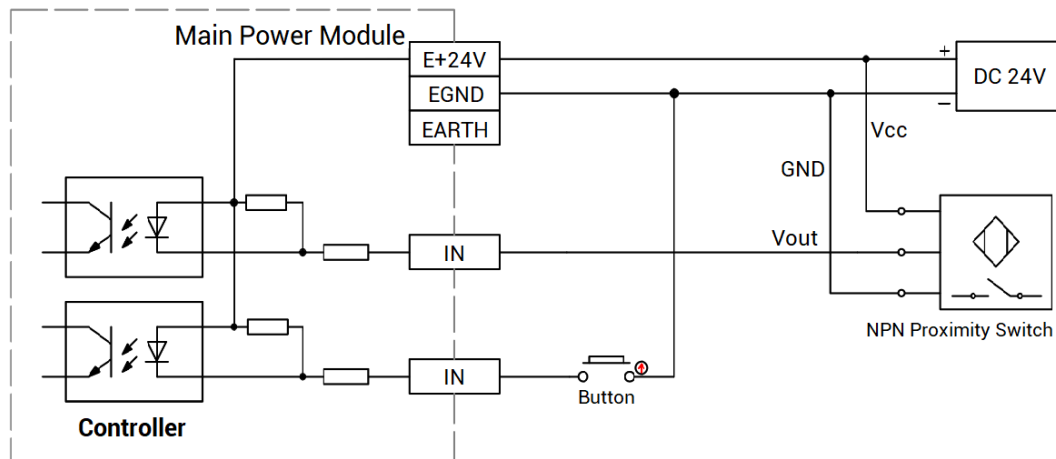
PIN	Name	Type	Function 1	Function 2	Function 3
2	EGND	/	IO public end	/	/
4	IN0	NPN type, high-speed input	Input 0	High-speed latch A	EA1
6	IN1		Input 1	High-speed latch B	EB1
8	IN2	NPN type, low-speed input	Input 2	/	EZ1
10	IN3		Input 3	/	/
12	IN4		Input 4	/	/
14	IN5		Input 5	/	/
16	IN6		Input 6	/	/
18	IN7		Input 7	/	/
20	IN8		Input 8	/	/
22	IN9		Input 9	/	/
24	IN10	NPN type, low-speed input	Input 10	/	/
26	IN11		Input 11	/	/
28	IN12		Input 12	/	/
30	IN13		Input 13	/	/
32	IN14		Input 14	/	/
34	IN15		Input 15	/	/
36	EGND	/	IO public end	/	/

3.2.1. Digital Input Specification & Wiring

→ Specification

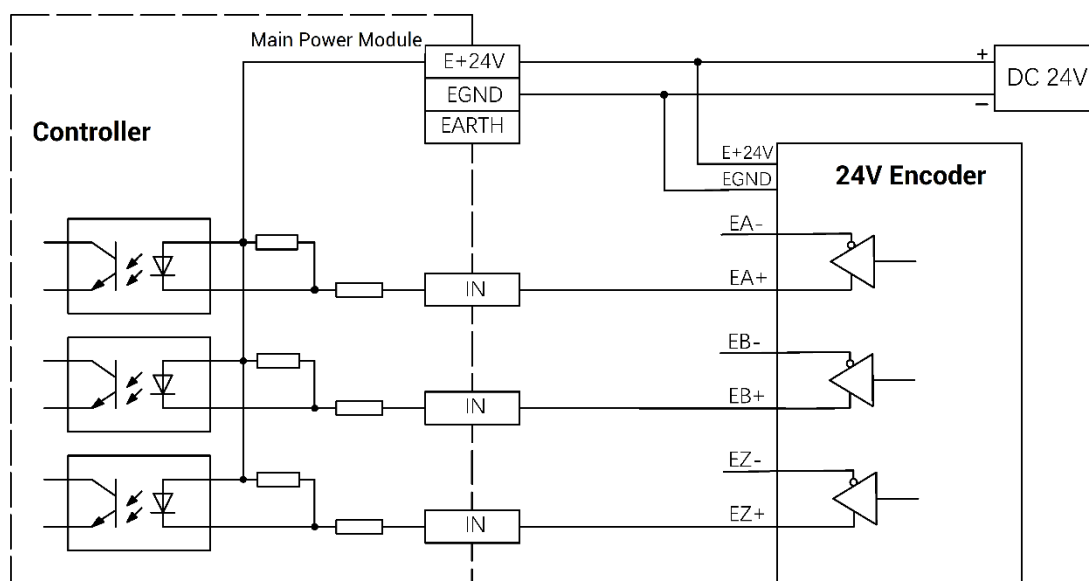
Item	High-Speed Input (IN0-1)	Low-Speed Input (IN2-15)
Input mode	NPN leakage type, the input is triggered when there is low-electric level	
Frequency	< 100kHz	< 5kHz
Impedance	3.3KΩ	4.7KΩ
Voltage level	DC24V	DC24V
The voltage to open	<15V	<14.5V
The voltage to close	>15.1V	>14.7V
Minimal current	-2.3mA (negative)	-1.8mA (negative)
Max current	-7.5mA (negative)	-6mA (negative)
Isolation mode	optoelectronic isolation	
<p>Note: the above parameters are standard values when the voltage of controller power supply (E+24V port) is 24V.</p>		

→ Wiring Reference



→ Encoder Wiring

Here, use IN0-2 to connect to encoder. When wiring is done, configure IN as encoder input signal through $ATYPE(1) = 3$, and IN0 is EA1, IN1 is EB1 and IN2 is EZ1, corresponding encoder axis No. is 1.



→ Wiring Note:

- The wiring principle of high-speed digital input IN (0-1) and low-speed digital input IN (2-15) is shown in the figure above. The external signal source can be an optocoupler, a key switch or a sensor, etc., all can be connected as long as the requirements on output of electric level can be achieved.
- For the public end, please connect the "EGND" port on the power supply to the "COM" terminal of the external input device. If the signal area power supply of the external device and the power supply of the controller are in the same power supply system, this connection also can be omitted.

3.2.2. Basic Usage Method

- (1) Please follow the above wiring instructions to wiring correctly.
- (2) After powered on, please select ETHERNET or RS232 to connect to ZDevelop.
- (3) State values of relative input ports can be read directly through "IN" command, also, it can be read through "ZDevelop/View/In". Please refer to "ZBasic" for details.

0	●	●	rev_in(0)
1	●	●	rev_in(1)
2	●	●	rev_in(2)
3	●	●	fwd_in(0)
4	●	●	fwd_in(1)
5	●	●	fwd_in(2)
6	●	●	alm_in(0)
7	●	●	alm_in(1)
8	●	●	alm_in(2)
9	●	●	
10	●	●	

- (4) Latch function can be set and triggered through "REGIST" instruction, in software, use REG_INPUTS to configure. Please refer to "ZBasic" for details.

3.3. OUT Digital Output & PWM & Hardware Comparison Output & Single-ended Pulse

The digital output adopts screw-type pluggable terminals with a spacing of 3.81 mm, and the PWM and hardware comparison output functions are integrated in digital output signal.

→ Terminal Definition

PIN	Name	Type	Function 1	Function 2	Function 3
1	EGND	/	E5V power ground / IO public end	/	/
3	OUT0	NPN leakage, high-speed output	Output 0	PWM 0	Hardware comparison output 0
5	OUT1		Output 1	PWM 1	Hardware comparison output 1
7	OUT2	NPN leakage, low-speed output	Output 2	/	
9	OUT3		Output 3	/	
11	OUT4		Output 4	/	
13	OUT5		Output 5	/	
15	OUT6	NPN leakage,	Output 6	/	
17	OUT7		Output 7	/	

19	OUT8	high-speed output	Output 8	DIR3	
21	OUT9		Output 9	PUL3	
23	OUT10		Output 10	DIR2	
25	OUT11		Output 11	PUL2	
27	OUT12		Output 12	DIR1	
29	OUT13		Output 13	PUL1	
31	OUT14		Output 14	DIR0	
33	OUT15		Output 15	PUL0	
35	E5V	/	5V power output, max is 300mA	/	

Note:

1. The E5V power output port is used for PWM or common anode wiring of single-ended axis. It is not recommended for other purposes due to lower power.
2. OUT8-15 have the function of axis 0-3, when ATYPE = 0, they are general outputs.
3. OUT0-3 have the functions of PWM and hardware comparison output.

3.3.2. Digital Output Specification & Wiring

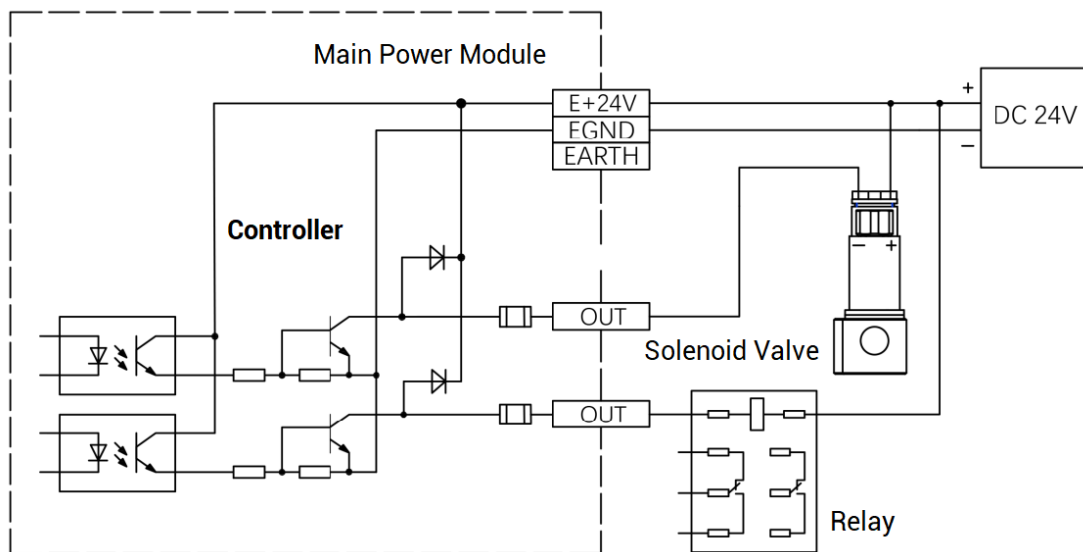
→ Specification

Item	High Speed Output (OUT0-1, OUT6-15)	Low Speed Output (OUT2-5)
Output mode	NPN leakage type, it is 0V when outputs.	
Frequency	< 400kHz	< 8kHz
Voltage level	DC24V	DC24V
Max output current	+300mA	+300mA
Max leakage current when off	25μA	25μA
Respond time to conduct	1μs (resistive load typical value)	12μs
Respond time to close	3μs	80μs
Overcurrent protection	Support	Support
Isolation method	optoelectronic isolation	

Note:

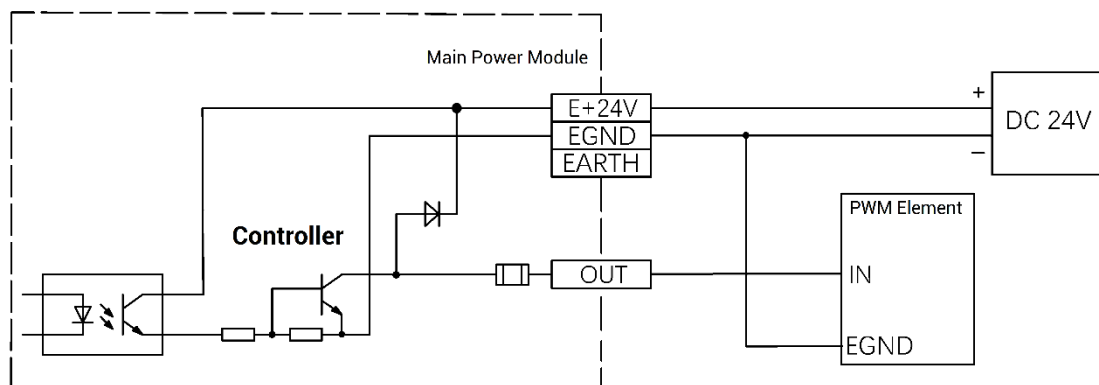
- ✧ The times in the form are typical based on the resistive load, and may change when the load circuit changes.
- ✧ Due to the leak-type output, the shutdown of the output will be obviously affected by the external load circuit, and the output frequency should not be set too high in the application. For high-speed output, it is recommended to set below 400KHz, for low-speed output, it is recommended to set below 8KHz.

→ **Wiring Reference**



→ **PWM Wiring**

Please select OUT ports that support PWM, OUT0 or OUT1.

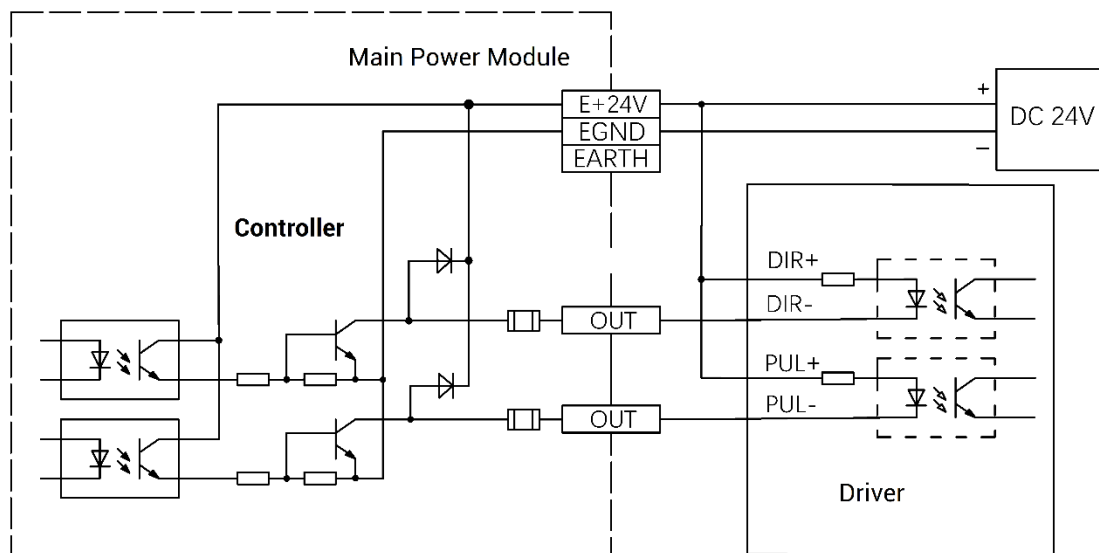


→ **Pulse Wiring**

Here, use OUT8 and OUT9 to connect to driver, when wiring is done, when OUT8 and

OUT9 are configured through ATYPE (3) = 1. OUT 8 is DIR3, OUT9 is PUL3, and corresponding pulse driver axis No. is 3.

E24V or E5V can be used according to specific driver.

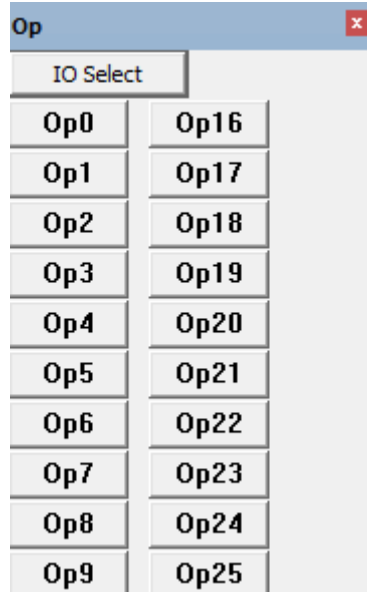


→ Wiring Note:

- The wiring principle of high-speed digital output OUT (0-1, 6-15) and low-speed digital output OUT (2-5) is shown in the figure above. The external signal receiving end can be an optocoupler or a relay or solenoid valve, all can be connected as long as the input current does not exceed 300mA.
- For the connection of the public end, please connect the "EGND" port on the IO terminal to the negative pole of the DC power supply of the external input device. If the DC power supply of the external device and the controller power supply are in the same power supply system, this connection can also be omitted.

3.3.3. Basic Usage Method

- (1) Please follow the above wiring instructions to wiring correctly.
- (2) After powered on, please use ETHERNET or RS232 to connect to ZDevelop.
- (3) Open or close output port directly through "OP" command, also, it can be opened or closed through "ZDevelop/View/Op". Please refer to "ZBasic" for details.



- (4) The PWM function, set the frequency and duty cycle through "PWM_FREQ" and "PWM_DUTY". Please refer to ZBasic for details.
- (5) Hardware comparison output can be set and opened through "HW_PSWITCH2". Please refer to ZBasic for details.

3.4. COM Serial Port

RS232 and RS485 integrate in one standard DB9 male socket, which support MODBUS_RTU protocol and custom communication.

→COM Interface Definition

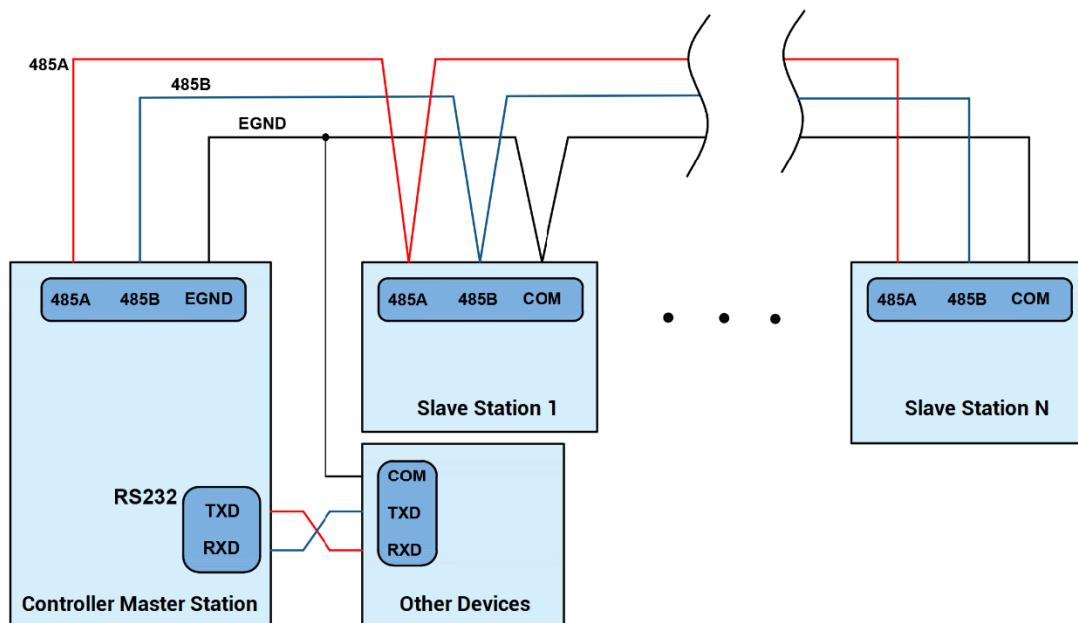
Terminal	PIN	Name	Type	Function
	1,6,8,9	NC	Spare	Reserved
	2	232RXD	Input	RS232 (port 0) signal receive
	3	232TXD	Output	RS232 (port 0) signal send
	4	485A/+	Input/output	RS485 (port 1) signal A/+
	5	EGND	Output	communication public end
	7	485B/-	Input/output	RS485 (port 1) signal B/-

3.4.1. RS232/RS485 Communication Interface Specification & Wiring

→ **Specification**

Item	RS232 (port0)	RS485 (port1)
Maximum Communication Rate (bps)	115200	115200
Terminal Resistor	No	No
Topology Structure	Connect correspondingly (1 to 1)	Daisy chain structure
The number of nodes can be extended	1	127
Communication Distance	The Longer communication distance is, the lower communication rate is, maximum 10m is recommended.	The Longer communication distance is, the lower communication rate is, maximum 100m is recommended.

→ **Wiring Reference**



→Wiring Notes:

- The wiring of RS232 (port0) is as above, it needs to cross-wiring for sending and receiving signals, and it is recommended to use a double-female head cross line when connecting to a computer.
- The wiring of RS485 (port1) is above, the daisy chain topology is used (the star topology structure cannot be used). When the use environment is ideal and there are no many nodes, the branch structure also can be used.
- Please be sure to connect the public ends of each communication node on RS485 to prevent the RS485 communication chip from burning out.
- When there are multiple slave stations, please connect a 120Ω terminal resistor in parallel to each end of the CAN bus for matching the circuit impedance and ensuring communication stability.
- Each node branch cable distance on RS485 wiring circuit needs to be less than 3m.
- Please use STP (Shielded Twisted Pair), especially in bad environments, and make sure the shielding layer is fully grounded.
- When on-site wiring, pay attention to make the distance between strong current and weak current, it is recommended for the distance to be more than 20cm.
- It should be noted that the equipment grounding (chassis) on the entire line must be good, and the grounding of the chassis should be connected to the standard factory ground pile.

3.4.2. Basic Usage Method

1. Please follow the above wiring instructions to wiring correctly.
2. After powered on, please use any one interface among the three interfaces ETHERNET, RS232 (there is default parameter, which can be connected directly) and RS485 (there is default parameter, which can be connected directly, but for hardware, adapter head is needed) to connect to ZDevelop.

3. Please use the "ADDRESS" and "SETCOM" commands to set and view the protocol station number and configured parameters, see "ZBasic Programming Manual" for details.
4. According to their respectively instructions, correctly set the relevant parameters of the third-party equipment to match the parameters of each node.
5. When all is configured, it can start to do communicating.
6. Communication data of RS232 / RS485 can be directly viewed through "ZDevelop / Controller / State the Controller / CommunicationInfo".

```

CAN communication settings:
CANIO_ADDRESS = 32, CANIO_ENABLE = 1
ZCAN Master
CAN baud: 500KBPS
CAN enable: ON

Serial port configuration:
Port0:(RS232) is ModbusSlave Mode.
Address:1, variable:2 delay:400ms
Baud:38400
DataBits:8
StopBits:1
Parity:0
Port1:(RS485) is ModbusSlave Mode.
Address:1, variable:2 delay:400ms
Baud:38400
DataBits:8
StopBits:1
Parity:0
    
```

3.5. LAN Ethernet

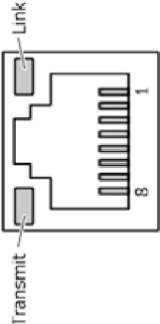
There are 2 1000M ethernet on VPLC516E board, and standard RJ45 interface is used. It supports MODBUS_TCP protocol and custom communication.

Ethernet factory default IP address is LAN1: 192.168.0.11, LAN2: 192.168.1.11, etc.

Ethernet supports EtherCAT bus interface configuration.

→ Interface Definition

Ethernet	PIN	100BASE Signal	1000BASE Signal
	1	TX+	TRD0+
	2	TX-	TRD0-

	3	RX+	TRD1+
	4	NC	TRD2+
	5	NC	TRD2-
	6	RX-	TRD1-
	7	NC	TRD3+
	8	NC	TRD3-
<p>Note:</p> <ol style="list-style-type: none"> There are 2 LED lights on RJ45, which represent ethernet Link and data transmission (Transmit). When ethernet is connected normally, Link led is ON in green. When data transmission is in 100M, Transit led is shrinking in green. When it is 1000M, Transit led is shrinking in orange. 			

→ Specification

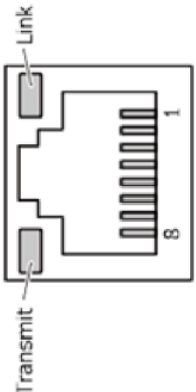
Item	USB3.0
Ethernet type	1000BASE-T/100BASE-TX/10BASE-T
Transmission Speed	1000Mbps/100Mbps/10Mbps
Max cable distance	100m/segment
Ethernet card type	Intel® Ethernet Controller
<p>Note: when transmission speed is 1000Mbps, the ethernet cable should be above CAT 5e at least.</p>	

3.6. EtherCAT Bus Interface

VPLCE516E motion controller has a 100M EtherCAT communication interface, and it supports EtherCAT bus protocol. In addition, EtherCAT driver or EtherCAT expansion module can be connected.

→ Interface Definition

ECAT	PIN	Signal
	1	TX+
	2	TX-

	3	RX+
	4	NC
	5	NC
	6	RX-
	7	NC
	8	NC

Note:

1. There are 2 LED lights on RJ45, which represent ethernet Link and data transmission (Transmit). When ethernet is connected normally, Link led is ON in green. When there is data transmission, Transmit led is shrinking in yellow.

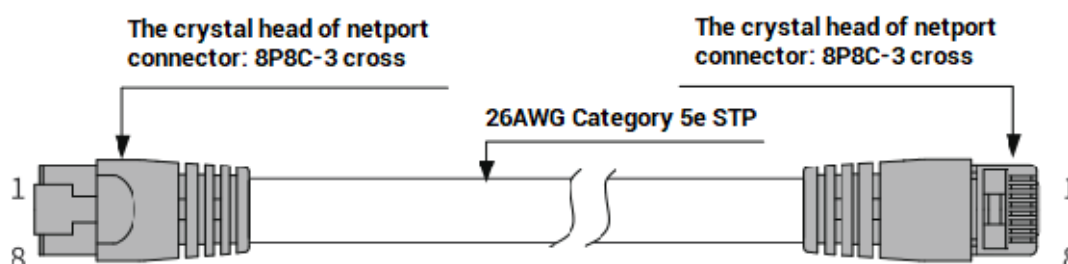
→ Specification

Item	Specification
Communication protocol	EtherCAT protocol
Valid service	CoE(PDO, SDO), FoE
Synchronization method	IO adopts input and output synchronization / DC-distributed clock
Physical level	100BASE-TX
Duplex mode	Full duplex
Topology	linear topology
Transfer media	Cable
Transfer distance	It is less than 100M between 2 nodes
Process data	Maximum 1486 bytes of one single frame
Synchronization shaking of two slave stations	<1us
Refresh	1000 digital input and output about is 30us

→ Communication Cable Requirements

Both ETHERNET communication interface and EtherCAT communication interface adopt standard Ethernet RJ45 interface.

The network cable adopts Category 5e STP, and the crystal head has a metal shell to reduce interference and to prevent information from being eavesdropped. As shown below:



Item	Specification
Cable type	Flexible crossover cable, Category 5e
traverse	twisted pair
Line pairs	4
Isolation	cross skeleton
Connector	Crystal head with iron shell
Cable material	PVC
Cable length	Less than 100m

Use RJ45 network cable connection method:

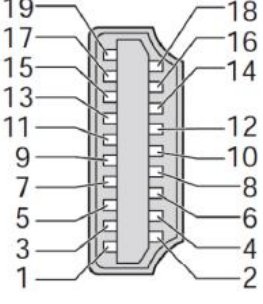
- When installing, hold the crystal head that is with the cable and insert it into the RJ45 interface until it makes a "click" sound (kada).
- In order to ensure the stability of communication, please fix the cables with cable ties.
- When disassembling, press the tail mechanism of the crystal head, and pull out the connector and the module in a horizontal direction.

Please use tube-type pre-insulated terminals and cables with appropriate wire diameters to connect the user terminals.

3.7. HDMI Interface

There is one standard HDMI high-definition multimedia display interface, which can connect to the touch screen.

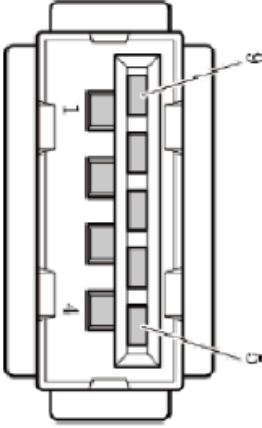
→ Interface Definition

HDMI	PIN	Signal	PIN	Signal
	1	TMDS DATA 2+	11	TMDS CLOCK SHIELD
	2	TMDS DATA 2 SHIELD	12	TMDS CLOCK -
	3	TMDS DATA 2-	13	CEC
	4	TMDS DATA 1+	14	N.C.
	5	TMDS DATA 1 SHIELD	15	DDC CLOCK
	6	TMDS DATA 1-	16	DDC DATA
	7	TMDS DATA 0+	17	GND
	8	TMDS DATA 0 SHIELD	18	+5V PWR
	9	TMDS DATA 0-	19	HOT PLUG DETECT
	10	TMDS CLOCK+		

3.8. USB Interface

The VPLC516E motion controller provides 4 USB communication interfaces, USB3.0 interface can be compatible with below USB2.0, which can connect to camera, keyboard, mouse, U disk, etc.

→ **Interface Definition**

USB3.0	PIN No.	Signal
	1	VCC5
	2	DATA-
	3	DATA+
	4	GND
	5	SSRX-
	6	SSRX+
	7	GND
	8	SSTX-
	9	SSTX+

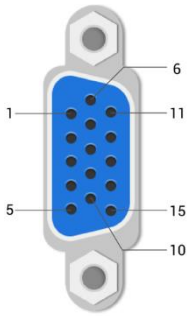
→ Specification

Item	USB3.0
Highest Communication Ratio	5.0Gbps
Max Output Current of VCC (5V)	500mA
Whether Isolates	No

3.9. MPG Handwheel Interface

This product provides a special interface for the local handwheel encoder axis, which is a double-row standard DB15 female seat.

→ Interface Definition

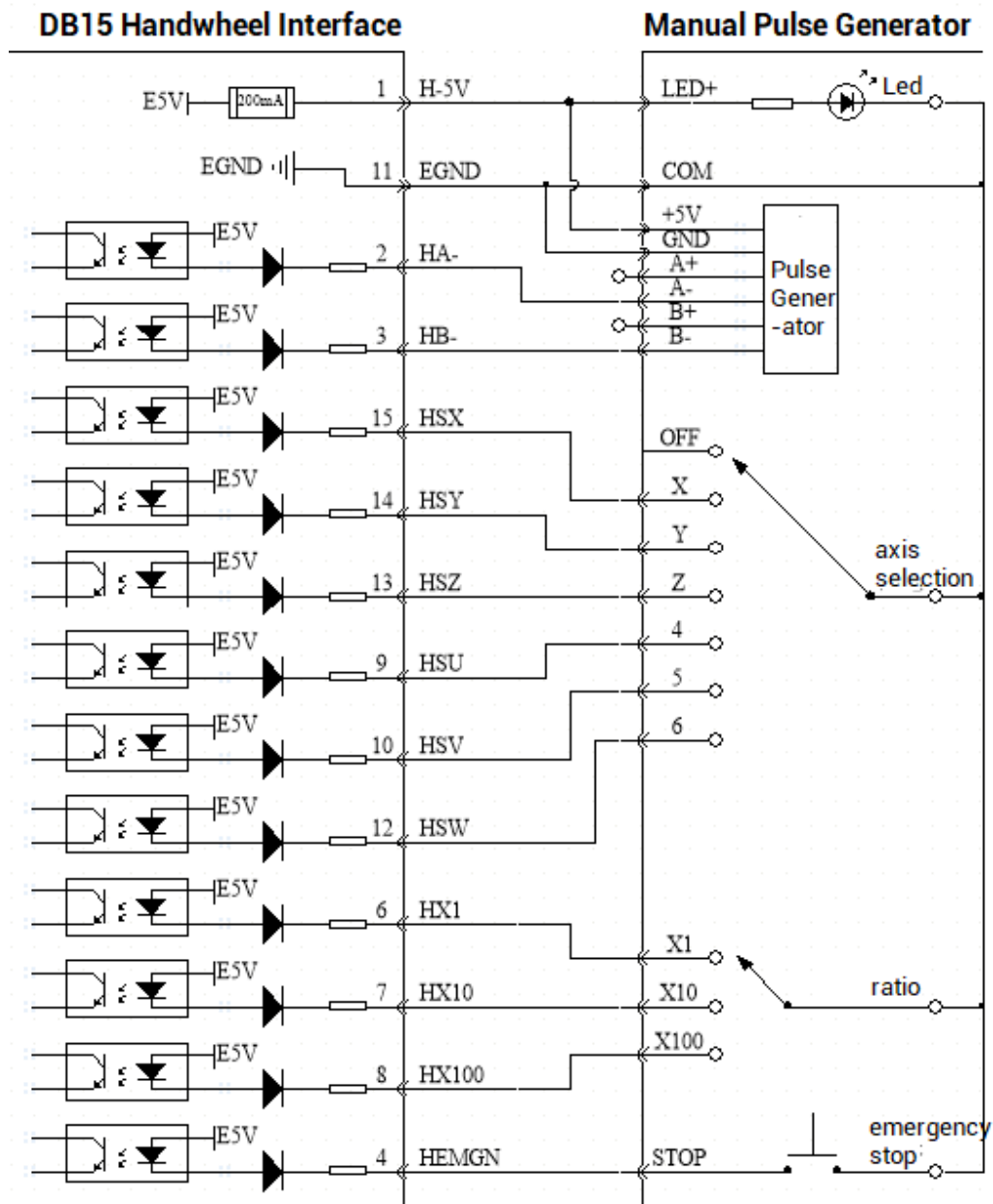
Interface	PIN	Signal	Description
	1	H-5V	Positive pole of 5V power supply for output, which supplied power for handwheel
	2	HA-	Encoder phase A signal (IN16)
	3	HB-	Encoder phase B signal (IN17)
	4	HEMGN	Emergency stop signal (IN27)
	5	NC	Reserved
	6	HX1	Select ration X1 (IN18)
	7	HX10	Select ration X10 (IN19)
	8	HX100	Select ration X100 (IN20)
	9	HSU	Axis selection 3 (IN24)
	10	HSV	Axis selection 4 (IN25)
	11	EGND	External power ground
	12	HSW	Axis selection 5 (IN26)
	13	HSZ	Axis selection 2 (IN23)
	14	HSY	Axis selection 1 (IN22)
	15	HSX	Axis selection 0 (IN21)

3.9.1. MPG Handwheel Interface Specification & Wiring

→ Specification

Item	IN (16-27)
Mode (input)	NPN leakage it is triggered when low electric inputs.
Frequency (input)	<5kHz is recommended
Impedance (input)	510Ω
Max voltage (input)	24V
The current to open (input)	<2.8
The current to close (input)	>2.9A
Min current (input)	-1.8mA
Max current (input)	-5.5mA
Isolation	optoelectronic isolation
5V power supply (H-5V, EGND) max output current	100mA

→ Wiring Reference



→ Wiring Note

- The wiring principle of handwheel encoder axis interface is shown above, please connect carefully due to diversified handwheel designs.
- Please use STP, especially in bad environments, and make sure the shielding layer is fully grounded.

3.9.2. Basic Usage Method

- (1) Refer to above handwheel wiring graphic, correctly connect the handwheel and controller.
- (2) After powered on, please select ETHERNET or RS232 to connect to ZDevelop.
- (3) Configure axis No., if there is no default AXIS axis No. (axis 10, 11, 12, 13 are recommended) of controller handwheel interface, remapped must be done. Followings are processes.

```

BASE(target axis number)      'the axis No. to be remapped
ATYPE(target axis number) = 0  'set axis type as 0
BASE(8)                        'handwheel interface initial axis No. is 8 (invalid)
ATYPE(8) = 0                   'set initial type of handwheel interface as 0
AXIS_ADDRESS(target axis number)=(-1<<16) + 8
                                'bind initial axis 8 to target axis No.

ATYPE(target axis number) = 6
                                'set this new axis interface as required axis type, such as 3 or 6

```

- (4) Configure IO: assign axis selection (HSX, HSY, HSZ, HSU) and ratio (HX1, HX10, HX100) and emergency stop (HEMGN) functions as required. These signals are essentially digital input signals with fixed numbers but no fixed functions. It needs ZDevelop development (the axis selection is the connected axis of "connect" synchronization motion, and the ratio is the "connect" ratio).
- (5) When completed above steps, it can start to use handwheel.

BASIC Routine Reference:

```

ATYPE(4) = 0                    'restore axis type of axis 10
ATYPE(8) = 0                    'restore default handwheel axis type
AXIS_ADDRESS(4) = (-1<<16)+ 8  'map the address of MPG manual pulse axis to axis 4
ATYPE(4) = 3                    'set manual pulse axis as quadrature encoder type
UNITS(4) = 1                    'set the unit as pulse for pulse amount of manual pulse axis
CONNECT(100,4) AXIS(0)
'axis 0 connects to manual pulse axis at the synchronous ration of 100

```

Chapter IV Expansion Module

The controller can expand digital IO, analog IO, pulse axis and other resources through CAN bus or EtherCAT bus expansion modules. For Zmotion, ZIO series CAN expansion module, EIO series EtherCAT expansion modules, or ZMIO310 series vertical bus expansion modules can be used. For details, please refer to corresponding user manual.

4.1. CAN Bus Expansion

ZIO series expansion modules or ZMIO310-CAN coupler with submodule can be selected.

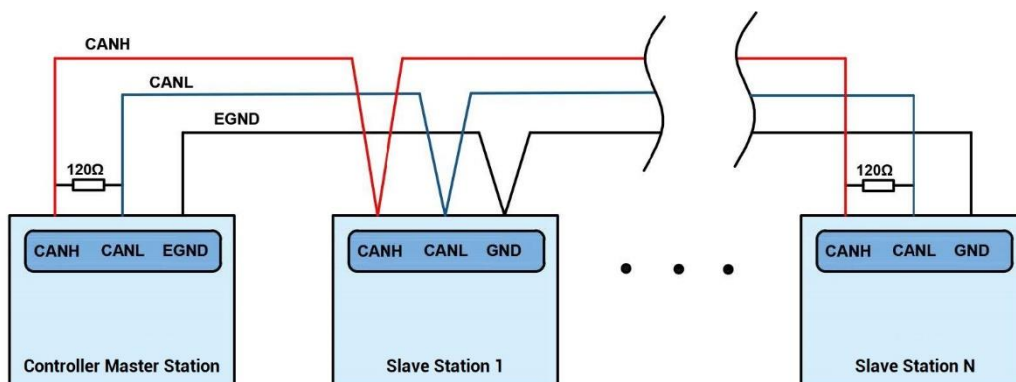
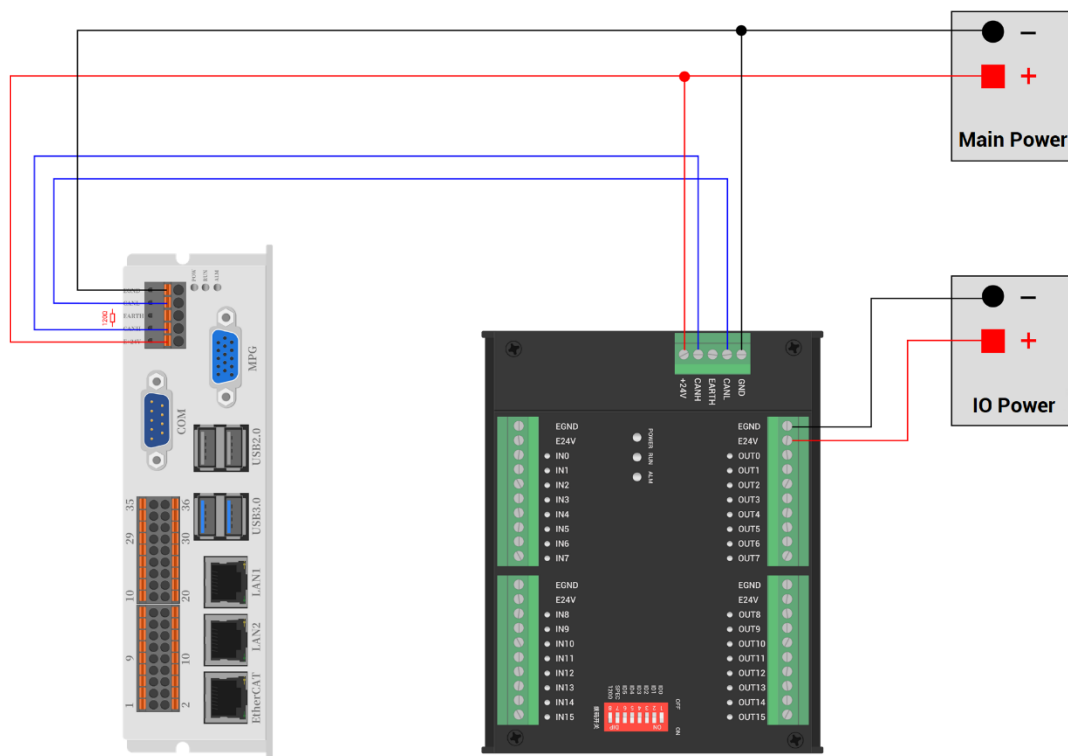
4.1.1. CAN Bus Expansion Wiring

The ZIO expansion module is powered by the dual power supply. Except the main power supply, an additional IO power supply is required to supply independent power for IO. Both the main power supply and the IO power supply use 24V DC power supply. For ZAIO, it only needs to connect to the main power supply.

To prevent interference, separate the IO power supply from the main power supply.

Please select the expansion module according to the requirements, and select IO mapping or axis mapping according to the resources of the expansion module.

Wiring reference of connection between ZIO expansion module and control card and standard wiring of CAN bus are shown as below:



→ **Wiring Note:**

- ✧ VPLCE516E controller uses the single power, and ZIO expansion module uses dual-power. When using, main power supply of expansion module and main power supply of controller can share one power. When they use different power supplies, controller power EGND needs to connect to expansion module power GND, otherwise CAN may be burnt out.
- ✧ When connecting multiple ZIO expansion modules on the CAN bus, a 120-ohm resistor needs to be connected in parallel between the CANL and CANH terminals, for the ZIO expansion module that is with 8-digit dialing codes, the terminal resistor can be realized by dialing the code (DIP).

4.1.2. CAN Bus Expansion Resource Mapping



The ZCAN expansion module generally has an 8-bit DIP switch, dial ON to take effect, and the meaning of the DIP is as follows:

1-4: they are used for ZCAN expansion module IO address mapping, the corresponding value is 0-15.

5-6: CAN communication speed, corresponding value is 0-3, four different speeds are optional.

7: reserved.

8: 120 ohm resistor, dial ON means a 120 ohm resistor is connected between CANL and CANH.

The IO numbers of the entire control system cannot be repeated, and existed numbers must be avoided when mapping resources. And the DIP switch must be dialed before power-on, if re-dial after power-on, it is invalid. It needs to be powered on again to take effect.

Dial 1-4 to select the CAN address, and the controller sets the IO number range of the corresponding expansion module according to the CAN DIP address. When each is dialed as OFF, the corresponding value is 0, when it is ON, it corresponds to a value of 1, and the address combination value = dial 4 \times 8 + dial code 3 \times 4 + dial code 2 \times 2 + dial code 1.

Dial code 5-6 to select CAN bus communication speed, speed combination value=dial code 6 \times 2 + dial code 5 \times 1, the combined value range is 0-3.

The corresponding speeds are as follows:

DIP 5-6 combination value	CANIO_ADDRESS high 8-bit value	CAN communication speed
0	0 (corresponds to decimal 128)	500KBPS (default value)
1	1 (corresponds to decimal 256)	250KBPS
2	2 (corresponding to decimal 512)	125KBPS
3	3 (corresponding to decimal 768)	1MBPS

The controller side sets the CAN communication speed through the CANIO_ADDRESS

command. There are also four speed parameters that can be selected. The communication speed must be consistent with the communication speed of the expansion module that corresponds to the combination value, then they can communicate with each other.

The factory default communication speed is 500 KBPS on both sides, there is no need to set this, unless you need to change the speed.

The CANIO_ADDRESS command is a system parameter, and it can set the master-slave end of CAN communication. The default value of the controller is 32, that is, CANIO_ADDRESS=32 is the master end, and the slave end is set between 0-31.

The CAN communication configuration can be viewed in the "State the Controller" window.

→ IO Mapping:

the CAN expansion module uses bit1-4 of the DIP switch. According to the number of currently included IO points (the largest number in IN and OP must include IO point in the axis interface), use the bit 1-4 to set the ID, so as to determine the number range of IO to be expanded.

If the controller itself contains 28 INs and 16 OPs, then the starting address set by the first extended board should exceed the maximum value of 28. According to below rule, the dial code should be set to the combination value 1 (binary combination value 0001, from right to left, dial code 1-4, at this time dial 1 is set to ON, and the others are set to OFF), the IO number on the expansion board = the expansion board number value + the initial IO number value, among them, the IOs that are vacant from 29-31 Numbers are not used. Subsequent extended boards continue to confirm the dial settings according to the IO points in turn.

The initial digital IO mapping number starts from 16 and increases in multiples of 16. The distribution of digital IO numbers corresponding to different dial IDs is as follows:

DIP 1-4 combination value	Starting IO number	End IO number
0	16	31
1	32	47
2	48	63
3	64	79
4	80	95
5	96	111

6	112	127
7	128	143
8	144	159
9	160	175
10	176	191
11	192	207
12	208	223
13	224	239
14	240	255
15	256	271

The initial IO mapping number of the analog AD starts from 8 and increases in multiples of 8. The initial IO mapping number of the analog DA starts from 4 and increases in multiples of 4. The allocation of digital IO numbers corresponding to different dial code IDs is as follows:

DIP 1-4 combination value	Starting AD number	End AD number	Starting DA number	End DA number
0	8	15	4	7
1	16	23	8	11
2	24	31	12	15
3	32	39	16	19
4	40	47	20	23
5	48	55	24	27
6	56	63	28	31
7	64	71	32	35
8	72	79	36	39
9	80	87	40	43
10	88	95	44	47
11	96	103	48	51
12	104	111	52	55
13	112	119	56	59
14	120	127	60	63
15	128	135	64	67

→ **Axis Mapping:**

When the CAN bus expansion mode is used to expand the pulse axis, ZIO16082M can be selected to expand two pulse axes. These two pulse axes need to be mapped and bound with the axis No., then access.

Extended axes need to perform axis mapping operations, using the AXIS_ADDRESS command to map, and the mapping rules are as follows:

AXIS_ADDRESS(axis No.)=(32*0)+ID

'the local axis interface of the expansion module AXIS 0

AXIS_ADDRESS(axis No.)=(32*1)+ID

'the local axis interface of the expansion module AXIS 1

The ID is the combined value of the DIP bit1-4 of the expansion module. After the mapping is completed and the axis parameters such as ATYPE are set, the expansion axis can be used.

Example:

ATYPE(6)=0 'set as virtual axis

AXIS_ADDRESS(6)=1+(32*0)

'ZCAN expansion module ID 1 axis 0 is mapped to axis 6

ATYPE(6)=8 'ZCAN extended axis type, pulse direction stepping or servo

UNITS(6)=1000 'pulse equivalent 1000

SPEED(6)=100 'speed 100units/s

ACCEL(6)=1000 'acceleration 1000units/s^2

MOVE(100) AXIS(6) 'extended axis movement 100units

Extended resource viewing:

According to the CAN connection, after the power is turned on, and the wiring resistance dial code is set correctly, the power indication led (POWER) and the running indication led (RUN), the IO power indication led (IO POWER) are on, and the alarm indication led (ALM) is off. At the same time, the "Controller" - "State the controller" - "ZCanNodes" in the ZDevelop software displays the expansion module information and the extended IO number range.

The dial ID and the corresponding resource number when connecting multiple expansion modules are as follows:

Local	432-0(ZMC432)	32	30(0-29)	18(0-17)	0	2(0-1)
1	48(ZIO1632)	0	16(32-47)	32(32-63)	0	0
3	26(ZIO16082)	2	16(64-79)	8(64-71)	0	0
4	10(ZAIO0802)	0	0	0	8(40-47)	2(20-21)

ALMRM indicator light is on, please check whether the wiring, resistor and dial setting are correct, and whether the CANIO_ADDRESS command of the controller is set as the master end (32), and whether the CAN communication speed is consistent.

4.2. EtherCAT Bus Expansion

The EIO expansion modules and ZMIO310-ECAT are expansion modules used by the EtherCAT bus controller. For example, EIO series can expand the resources of digital IO and pulse axis. When the resources of the controller are insufficient, the EtherCAT bus controller can be connected to multiple EIO expansion modules for expansion, you can view the maximum number of IO expansion points and the maximum number of expansion axes of the controller, and in this way, it supports IO remote expansion.

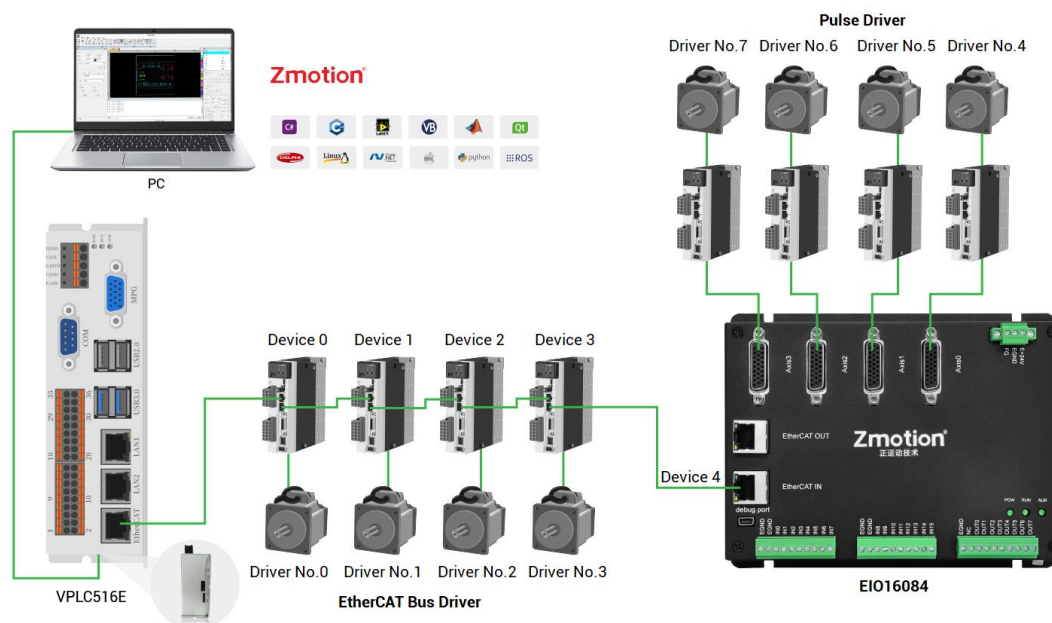
4.2.1. EtherCAT Bus Expansion Wiring

After the expansion wiring is completed, each EIO expansion module does not need to develop again. It only needs to manually configure the unique IO address and axis address in the EtherCAT master controller, and it can be accessed after the configuration is completed.

The IO address number is set through the bus command NODE_IO, and the program on the controller can access the resources on the expansion module only through the IO number. The configuration of the axis address uses the AXIS_ADDRESS command to map axis number, and when the binding is completed, specify the axis number through the BASE or AXIS command.

When wiring, pay attention that EtherCAT IN is connected to the upper-level module, and EtherCAT OUT is connected to the lower-level module. The IN and OUT ports cannot be mixed.

EIO expansion module wiring reference example (take ZMC408SCAN as an example):



Involved number concepts in above figure are as follows: the bus-related command parameters will use the following numbers:

Slot number (slot):

The slot number refers to the number of the bus interface on the controller, and the slot number of the EtherCAT bus is 0.

Device number (node):

The device number refers to the number of all devices connected to a slot. It starts from 0 and is automatically numbered according to the connection sequence of the devices on the bus. You can view the total number of devices connected to the bus through the `NODE_COUNT(slot)` command.

Drive number:

The controller will automatically identify the drive on the slot, and the number starts from 0, and the number is automatically numbered according to the connection sequence of the drive on the bus.

The drive number is different from the device number. Only the drive device number on the slot is assigned, and other devices are ignored. The drive number will be used when mapping the axis number.

4.2.2. EtherCAT Bus Expansion Resource Mapping

→ IO Mapping:

The program on the controller can access the resources on the expansion module only through the IO number. The IO number of the EtherCAT bus expansion module is set through the bus command NODE_IO, and the input and output are configured at the same time.

When IO mapping, first check the maximum IO number of the controller itself (including the external IO interface and the interface in the pulse axis), and then use the command to set.

If the extended IO coincides with the IO number of the controller itself, the two will work at the same time, so the mapped number of the IO mapping must not be repeated in the entire control system.

IO mapping syntax:

NODE_IO(slot, node) = iobase

slot: slot number, 0-default

node: device number, starting from 0

iobase : mapping the IO start number, the setting result will only be a multiple of 8

Example:

NODE_IO(0,0)=32 'set the IO start number of slot 0 interface device 0 to 32

If device 0 is EIO16084, after configuration according to the above syntax, the IO numbers corresponding to input IN0-15 are 32-47 in turn, the general input port numbers in the axis interface are 48-55, and the drive alarm inputs of axes AXIS 0-3 are 48-51 respectively. The IO numbers corresponding to the output OUT0-7 are 32-39 in sequence, the general output port numbers in the axis interface are 40-47, and the drive enable outputs of the axes AXIS 0-3 are 40-43 respectively.

0	41bh	1918h	0	4	24(32-55)	16(32-47)	0

→ AXIS Mapping:

Before using the axis of the expansion module, you need to use the AXIS_ADDRESS

command to map the axis number, and the axis mapping also needs to pay attention to the axis number of the entire system cannot be repeated. The mapping syntax of the EIO series extended axis is the same as that of the bus driver.

Axis mapping syntax:

$AXIS_ADDRESS(\text{axis number}) = (\text{slot number} \ll 16) + \text{driver number} + 1$

Example:

$AXIS_ADDRESS(0) = (0 \ll 16) + 0 + 1$

'the first drive on the EtherCAT bus, drive number 0, bound as axis 0

$AXIS_ADDRESS(1) = (0 \ll 16) + 1 + 1$

'the second drive on the EtherCAT bus, drive number 1, bound as axis 1

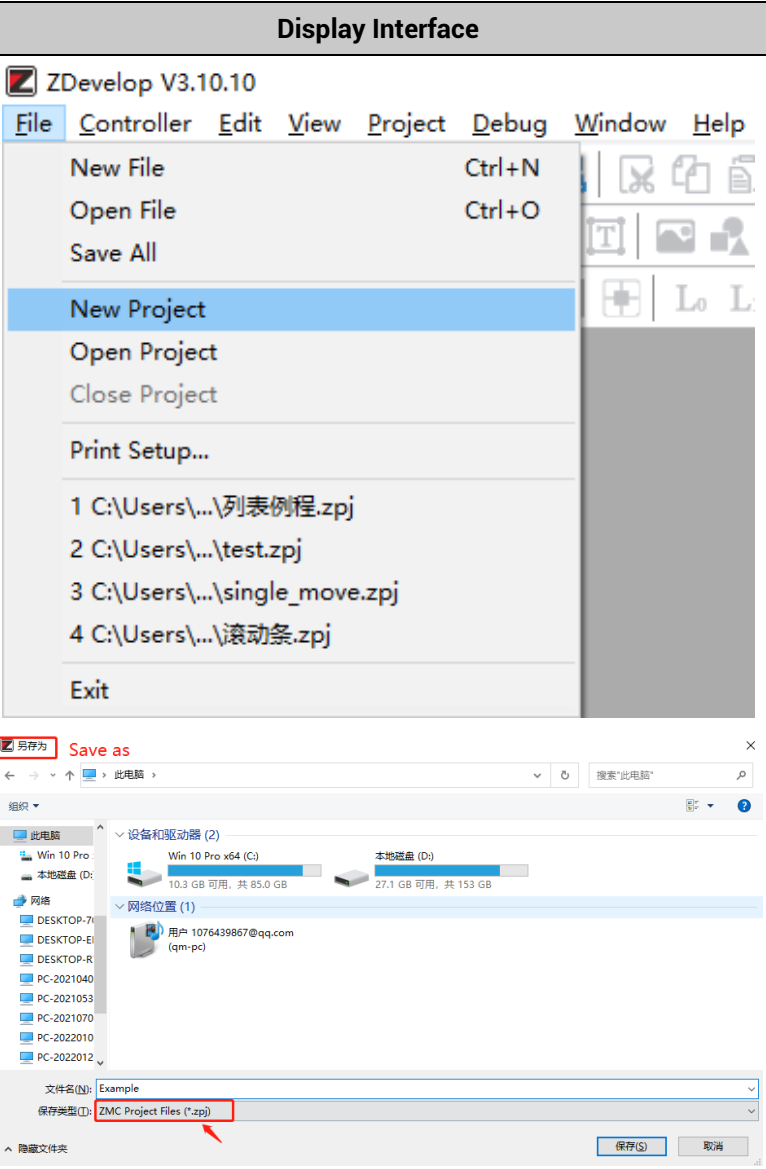
If the first node is EIO16084, and EIO16084 is connected to drive, then driver 0 here is the first pulse driver connected to EIO16084, otherwise it is the EtherCAT driver.

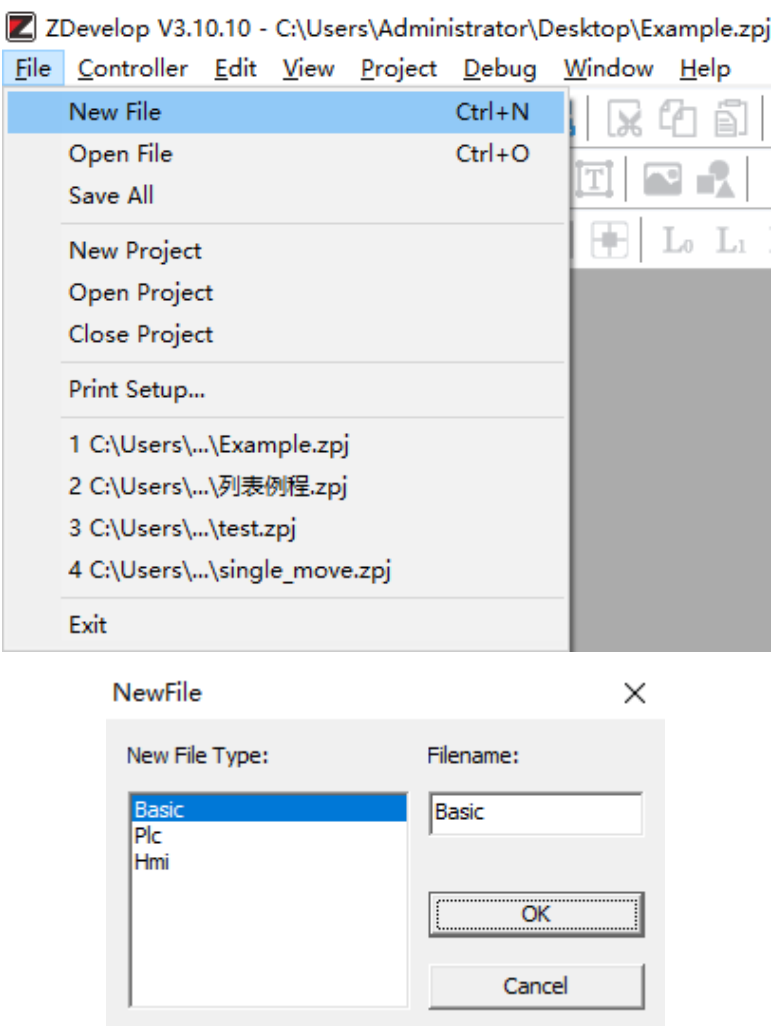
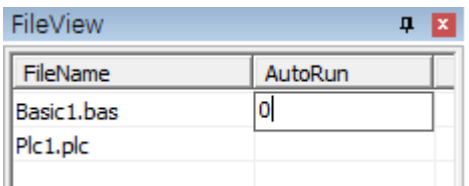
Chapter V Program & Applications

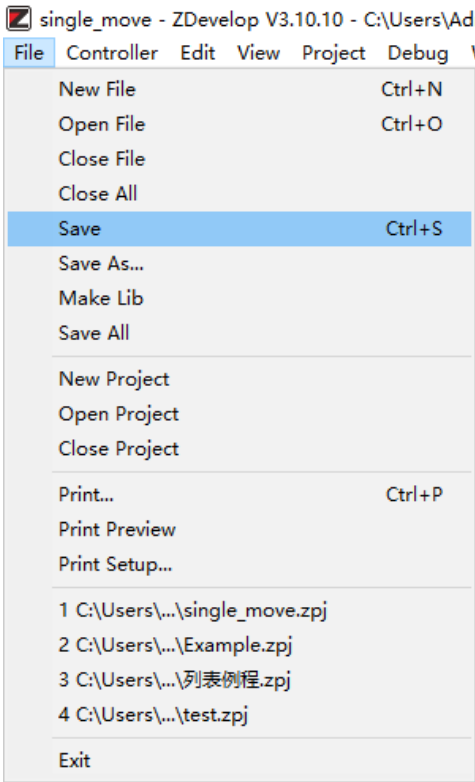
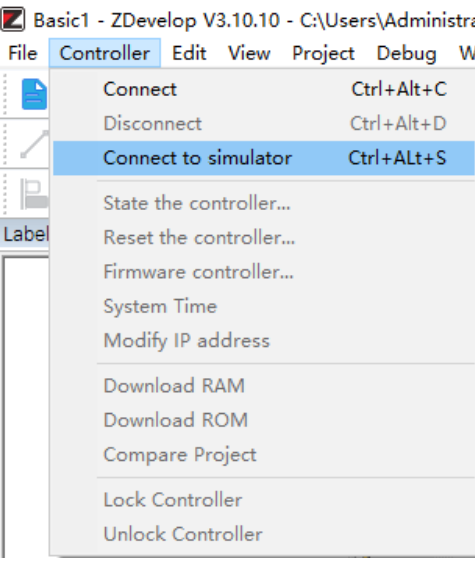
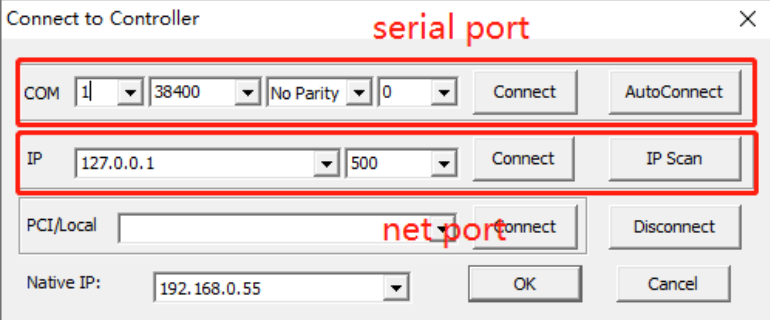
5.1. ZDevelop Software Usage

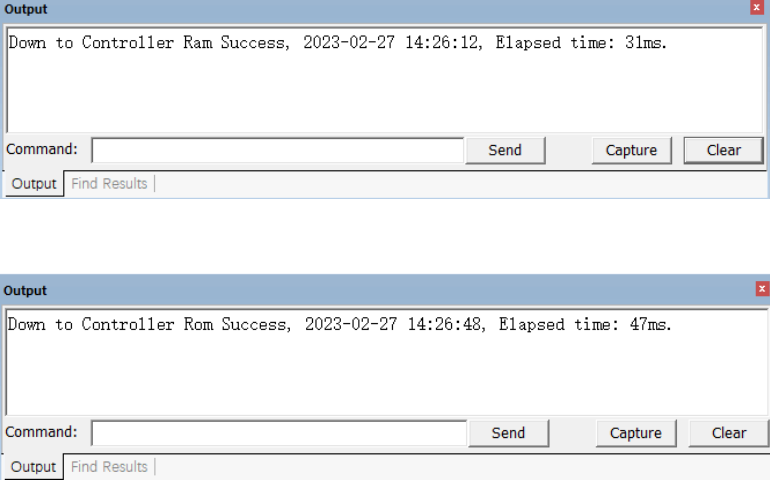
ZDevelop is a PC-side program development, debugging and diagnostic software for the ZMoiton series motion controllers of Zmotion Technology. Through it, users can easily edit and configure the controller program, quickly develop applications, diagnose system operating parameters in real time, and watch the motion controller. The running program is debugged in real time and supports Chinese and English bilingual environments.

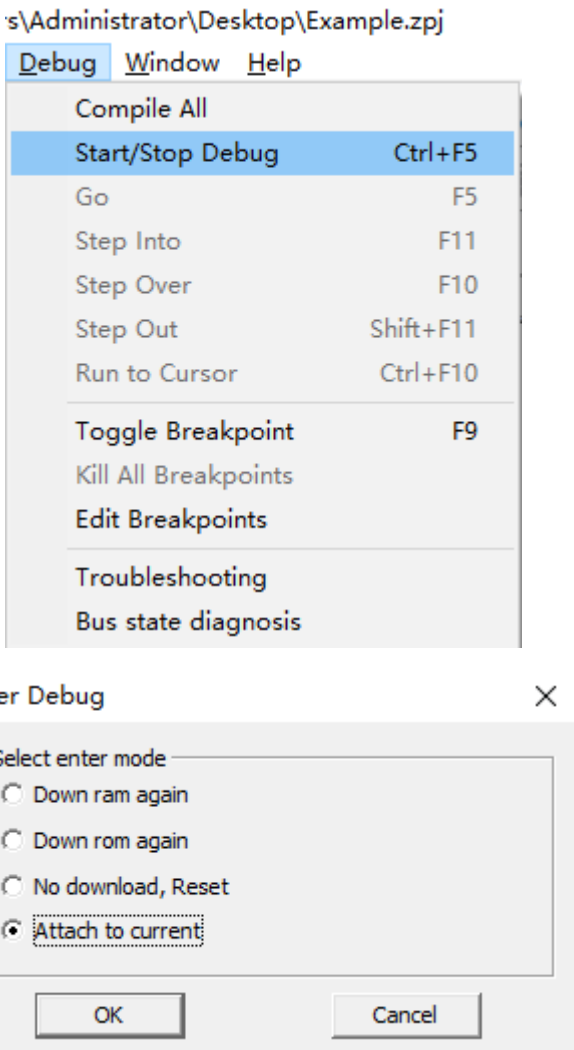
ZBasic, ZPLC and ZHMI can run multi-tasks, and ZBasic can run multi-tasks, and can be mixed with ZPLC and ZHMI.

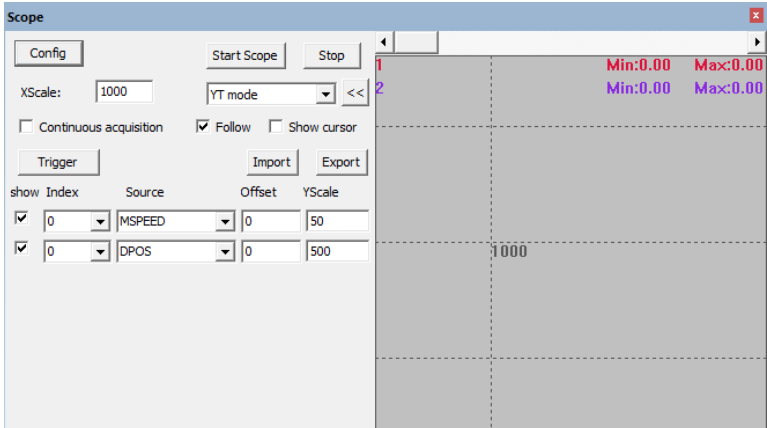
Step	Operations	Display Interface
1	Open ZDevelop, click "File" – "New Project", Save as window will pop up, then enter file name, save the project file with suffix "zpj".	 <p>The screenshot shows the ZDevelop V3.10.10 application window. The 'File' menu is open, and 'New Project' is highlighted. Below the menu, a list of existing project files is visible: '1 C:\Users\...\列表例程.zpj', '2 C:\Users\...\test.zpj', '3 C:\Users\...\single_move.zpj', and '4 C:\Users\...\滚动条.zpj'. A 'Save as' dialog box is also open, showing the file name 'Example' and the file type 'ZMC Project Files (*.zpj)' selected. The dialog box also shows the file explorer view with '本地磁盘 (D:)' selected.</p>

<p>2</p>	<p>Click "File" – "New File", select file type to build, here select Basic, click "OK".</p>	 <p>The screenshot shows the ZDevelop V3.10.10 application window. The 'File' menu is open, showing options like 'New File (Ctrl+N)', 'Open File (Ctrl+O)', 'Save All', 'New Project', 'Open Project', 'Close Project', 'Print Setup...', and a list of recent files. Below the menu, the 'NewFile' dialog box is displayed. It has a 'New File Type' list with 'Basic', 'Plc', and 'Hmi' options. The 'Basic' option is selected. The 'Filename' field contains the text 'Basic'. There are 'OK' and 'Cancel' buttons at the bottom of the dialog.</p>						
<p>3</p>	<p>Double click "AutoRun", enter task number 0.</p>	 <p>The screenshot shows a 'FileView' window with a table. The table has two columns: 'FileName' and 'AutoRun'. The first row contains 'Basic1.bas' and '0'. The second row contains 'Plc1.plc' and an empty field.</p> <table border="1" data-bbox="715 1288 1181 1422"> <thead> <tr> <th>FileName</th> <th>AutoRun</th> </tr> </thead> <tbody> <tr> <td>Basic1.bas</td> <td>0</td> </tr> <tr> <td>Plc1.plc</td> <td></td> </tr> </tbody> </table>	FileName	AutoRun	Basic1.bas	0	Plc1.plc	
FileName	AutoRun							
Basic1.bas	0							
Plc1.plc								

<p>4</p>	<p>Edit the program in program editing window, click "save", new built basic file will be saved under "zpj." project automatically. "Save all" means all files under this project will be saved.</p>	 <p>The screenshot shows the 'File' menu in ZDevelop V3.10.10. The 'Save' option is highlighted in blue, with the keyboard shortcut 'Ctrl+S' displayed to its right. Other menu items include 'New File', 'Open File', 'Close File', 'Close All', 'Save As...', 'Make Lib', 'Save All', 'New Project', 'Open Project', 'Close Project', 'Print...', 'Print Preview', 'Print Setup...', and a list of open files.</p>
<p>5</p>	<p>Click "controller – connect", if no controller, select connect to simulator.</p>	 <p>The screenshot shows the 'Controller' menu in ZDevelop V3.10.10. The 'Connect to simulator' option is highlighted in blue, with the keyboard shortcut 'Ctrl+ALT+S' displayed to its right. Other menu items include 'Connect', 'Disconnect', 'State the controller...', 'Reset the controller...', 'Firmware controller...', 'System Time', 'Modify IP address', 'Download RAM', 'Download ROM', 'Compare Project', 'Lock Controller', and 'Unlock Controller'.</p>
<p>Then, "connect to controller" window will pop up, you can select serial port or net port to connect, select matched serial port</p>	<p></p>	 <p>The screenshot shows the 'Connect to Controller' dialog box. It has two main sections. The top section, labeled 'serial port', contains fields for COM (1), Baud Rate (38400), Parity (No Parity), and Stop Bits (0), along with 'Connect' and 'AutoConnect' buttons. The bottom section, labeled 'net port', contains fields for IP (127.0.0.1) and Port (500), along with 'Connect' and 'IP Scan' buttons. Below these are fields for 'PCI/Local' and 'Native IP' (192.168.0.55), with 'Connect', 'Disconnect', 'OK', and 'Cancel' buttons.</p>

	<p>parameters or net port IP address, then click "connect".</p>	
<p>6</p>	<p>Click "Ram/Rom" – "download RAM / download ROM", if it is successful, there is print indication, at the same time, program is downloaded into controller and runs automatically. RAM: it will not save when power off. ROM: it will save data when power off, and when the program is connected to controller again, running according to task number.</p>	 <p>The image shows two screenshots of the 'Output' window. The top screenshot displays the message: 'Down to Controller Ram Success, 2023-02-27 14:26:12, Elapsed time: 31ms.' The bottom screenshot displays the message: 'Down to Controller Rom Success, 2023-02-27 14:26:48, Elapsed time: 47ms.' Both screenshots include a 'Command:' input field, 'Send', 'Capture', and 'Clear' buttons, and a 'Find Results' button.</p>

<p>7</p>	<p>Click “Debug” – “Start/Stop Debug” to call “Task” and “Watch” window, because it was downloaded before, here select “Attach the current”.</p>	
----------	--	---

<p>8</p>	<p>Click “View” – “Scope” to open oscilloscope.</p>	
----------	---	--

Note:

- When opening an project, choose to open the zpj file of the project. If only the Bas file is opened, the program cannot be downloaded to the controller.
- When the project is not created, only the Bas file cannot be downloaded to the controller.
- The number 0 in automatic operation represents the task number, and the program

runs with task 0, and the task number has no priority.

- If no task number is set for the files in the entire project, when downloading to the controller, the system prompts the following message WARN: no program set autorun

5.2. PC Upper-Computer Program Application

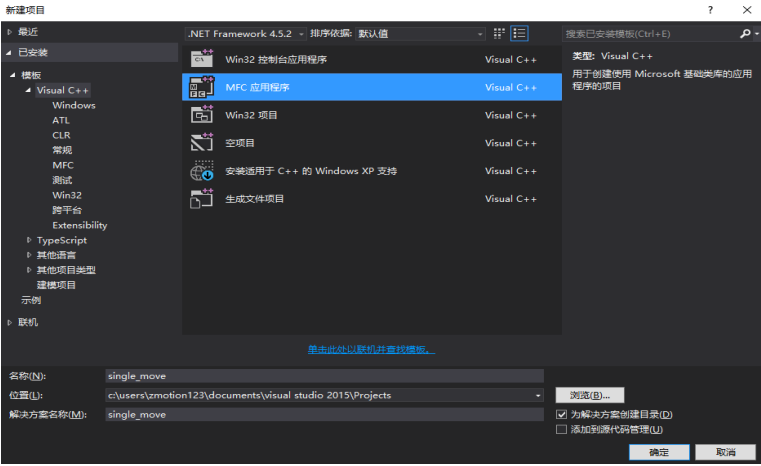

The controller supports development under various operating systems such as windows, linux, Mac, Android, and wince, and provides dll libraries in various environments such as vc, c#, vb.net, and labview, as shown in the figure below. PC software programming refers to "ZMotion PC Function Library Programming Manual".

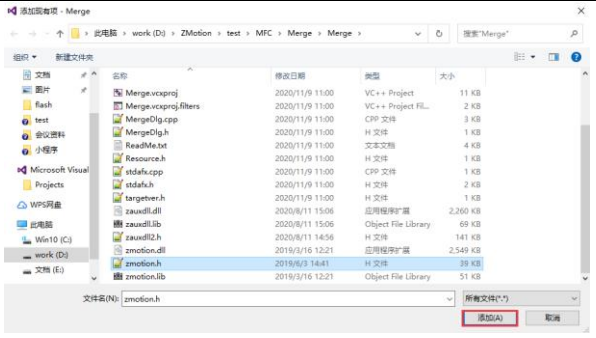
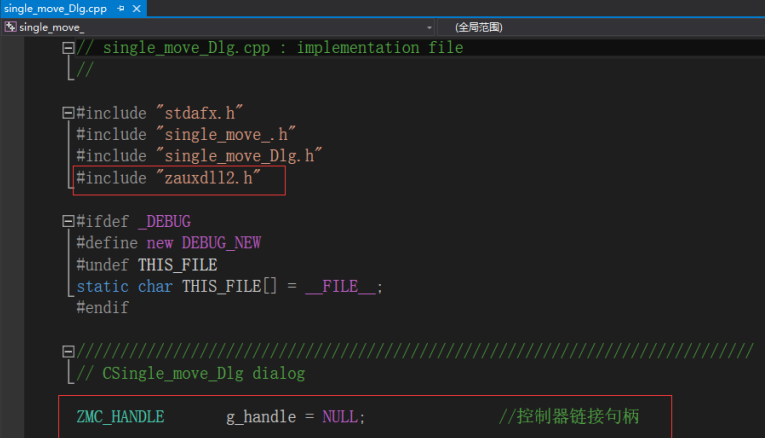


The program developed using the PC software cannot be downloaded to the controller, and it is connected to the controller through the dll dynamic library. The dll library needs to be added to the header file and declared during development.

The c++ project development process in VS is as follows:

Step	Operations	Display Interface
1	Open VS, click "File" – "New" – "Project".	<p>The screenshot shows the Microsoft Visual Studio interface. The 'File' menu is open, and the 'New' option is selected, which has opened a sub-menu. In this sub-menu, the 'Project...' option is highlighted with a red box. The keyboard shortcut 'Ctrl+Shift+N' is visible next to it. Other menu items like 'Open...', 'Close', and 'Save' are also visible.</p>

<p>2</p>	<p>Select development language as "Visual C++" and the select program type as "MFC application type".</p>	
<p>3</p>	<p>Select "Based on basic box", click "next" or "finish".</p>	
<p>4</p>	<p>Find C++ function library provided by manufacturer. Routine is below (64-bit library)</p>	
<p>5 Copy all DLL related library files under the above path to the newly created project.</p>		
<p>6</p>	<p>Add a static library and related header files to the project. Static library: zauxdll.lib,</p>	<p>1) Right-click the header file first, and then select: "Add" → "Existing"</p> 

	<p>zmotion.lib</p> <p>Related header files: zauxdll2.h, zmotion.h</p>	<p>Item".</p> <p>2) Add static libraries and related header files in sequence in the pop-up window.</p>	
<p>7</p>	<p>Declare the relevant header files and define the controller connection handle, so far the project is newly created.</p>		

Chapter VI Run and Maintain

The correct operation and maintenance of the device can not only guarantee and extend the life cycle of the equipment itself, but also take technical management measures according to the pre-specified plan or the corresponding technical conditions to prevent equipment performance degradation or reduce the probability of equipment failure.

6.1. Regular Inspection and Maintenance

The working environment has an impact on the device. Therefore, it is usually inspected regularly based on the inspection cycle of 6 months to 1 year. The inspection cycle of the device can be appropriately adjusted according to the surrounding environment to make it work within the specified standard environment.

Check item	Check content	Inspection standards
power supply	Check whether the voltage is rated	DC 24 V (-5%~5%)
surroundings	Whether the ambient temperature is within the specified range (when installed in the cabinet, the temperature inside the cabinet is the ambient temperature)	-10°C - 55 °C
	Whether the ambient humidity is within the specified range (when installed in the cabinet, the humidity in the cabinet is the ambient humidity)	5%-95% non-condensing
	Is there direct sunlight	No
	With or without droplets of water, oil, chemicals, etc.	No
	Whether there is dust, salt, iron filings, dirt	No
	Whether there is corrosive gas	No
	Whether there are flammable and explosive gases or articles	No

	Whether the device is subjected to vibration or shock	Should be within the range of vibration resistance and impact resistance
	Is the heat dissipation good	Keep good ventilation and heat dissipation
Installation and Wiring Status	Whether the basic unit and the expansion unit are installed firmly	The mounting screws should be tightened without loosening
	Whether the connecting cables of the basic unit and the expansion unit are fully inserted	The connection cable cannot be loosened
	Are the screws of the external wiring loose	Screws should be tightened without loosening
	Whether the cable is damaged, aged, cracked	The cable must not have any abnormal appearance

6.2. Common Problems

Problems	Suggestions
Motor does not rotate.	<ol style="list-style-type: none"> 1. Check whether the ATYPE of the controller is correct. 2. Check whether hardware position limit, software position limit, alarm signal work, and whether axis states are normal. 3. Check whether motor is enabled successfully. 4. Confirm whether pulse amount UNITS and speed values are suitable. If there is the encoder feedback, check whether MPOS changes. 5. Check whether pulse mode and pulse mode of drive are matched. 6. Check whether alarm is produced on motion controller station or drive station. 7. Check whether the wiring is correct. 8. Confirm whether controller sends pulses normally.
The position limit signal is	<ol style="list-style-type: none"> 1. Check whether the limit sensor is working normally,

invalid.	<p>and whether the "input" view can watch the signal change of the limit sensor.</p> <ol style="list-style-type: none"> 2. Check whether the mapping of the limit switch is correct. 3. Check whether the limit sensor is connected to the common terminal of the controller.
No signal comes to the input.	<ol style="list-style-type: none"> 1. Check whether the limit sensor is working normally, and whether the "input" view can watch the signal change of the limit sensor. 2. Check whether the mapping of the limit switch is correct. 3. Check whether the limit sensor is connected to the common terminal of the controller.
The output does not work.	<ol style="list-style-type: none"> 1. Check whether IO power is needed. 2. Check whether the output number matches the ID of the IO board.
POWER led is ON, RUN led is OFF.	<ol style="list-style-type: none"> 1. Check whether the power of the power supply is sufficient. At this time, it is best to supply power to the controller alone, and restart the controller after adjustment. 2. Check whether the ALM light flickers regularly (hardware problem).
RUN led is ON, ALM led is ON.	<ol style="list-style-type: none"> 1. Program running error, please check ZDevelop error code, and check application program.
Fail to connect controller to PC through serial port.	<ol style="list-style-type: none"> 1. Check whether the serial port parameters are modified by the running program, you can check all the current serial port configurations through ?*SETCOM. 2. Check whether the serial port parameters of the PC match the controller. 3. Open the device manager and check whether the serial driver of the PC is normal.
CAN expansion module cannot be connected.	<ol style="list-style-type: none"> 1. Check the CAN wiring and power supply circuit, whether the 120 ohm resistor is installed at both ends.

	<ol style="list-style-type: none"> 2. Check the master-slave configuration, communication speed configuration, etc. 3. Check the DIP switch to see if there are multiple expansion modules with the same ID. 4. Use twisted-pair cables, ground the shielding layer, and use dual power supplies for severe interference (the main power supply of the expansion module and the IO power supply are separately powered)
<p>Fail to connect controller to PC through net port.</p>	<ol style="list-style-type: none"> 1. Check IP address of PC, it needs to be at the same segment with controller IP address. 2. Check controller IP address, it can be checked and captured after connection through serial port. 3. When net port led is off, please check wiring. 4. Check whether controller power led POWER and running indicator led RUN are ON normally. 5. Check whether the cable is good quality, change one better cable to try again. 6. Check whether controller IP conflicts with other devices. 7. Check whether controller net port channel ETH are all occupied by other devices, disconnect to other devices, then try again. 8. When there are multiple net cards, don't use other net cards, or change one computer to connect again. 9. Check PC firewall setting. 10. Use "Packet Internet Groper" tool (Ping), check whether controller can be Ping, if it can't, please check physical interface or net cable. 11. Check IP address and MAC address through arp-a.