ZDevelop User Manual

Version 3.10.04

Copyright statement

ZMC controller software involved in details as well as the introduction and routines of each instruction, please refer to 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!

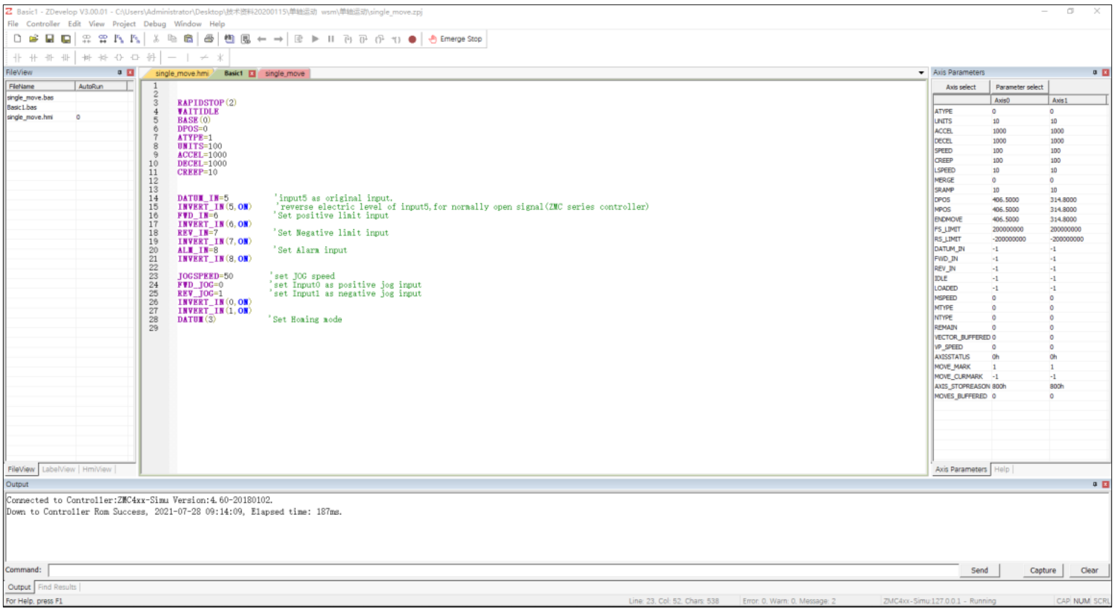
⚠ Pay attention to safety when debug the machine! Be sure to design effective safety devices in the machine and add error handling procedures in software. Zmotion has no obligation or responsibility for the loss.

Contents

Chapter I ZDevelop Overview

ZDevelop is a green development software that integrates machine vision and motion control functions developed by ZMOTION. It supports ZBaisc, ZPLC, ZHMI and ZVISION machine vision secondary development. Also, there are many functions, it can hybrid program, achieve simulation, track online, diagnose and debug. What's more, vision positioning, measurement, identification and detection of intellectual equipment can be realized rapidly, and complex motion control systems can be developed. ZDevelop software development page refers to below.

Users can connect PC and controller through serial port or net port, program written by ZDevelop software can be downloaded into ZMOTION motion controller to run offline or simulate on PC.


Basic1 - ZDevelop V3.00.01 - C:\Users\Administrator\Desktop\技术资料20200115\单轴运动 wsm\单轴运动\single_move.zpj
File Controller Edit View Project Debug Window Help
Emerge Stop
FileView
single_move.hmi
Basic1
single_move
Axis Parameters
Axis select
Parameter select
FileView LabelView HmiView
Axis Parameters Help
Output
Command:
Send
Capture
Clear
Output Find Results
For Help, press F1
Line: 23, Col: 52, Chars: 538
Error: 0, Warn: 0, Message: 2
ZMC4xx-Simu:127.0.0.1 - Running
CAP NUM SCRL

FileName	AutoRun
single_move.bas	
Basic1.bas	
single_move.hmi	0

```
RAPIDSTOP(2)
WAITIDLE
BASE(0)
DPOS=0
ATYPE=1
UNITS=100
ACCEL=1000
DECEL=1000
CREEP=10


DATUM_IN=5            'input5 as original input.
INVERT_IN(5,ON)        'reverse electric level of input5,for normally open signal(ZMC series controller)
FWD_IN=6              'Set positive limit input
INVERT_IN(6,ON)
REV_IN=7              'Set Negative limit input
INVERT_IN(7,ON)
ALM_IN=8              'Set Alarm input
INVERT_IN(8,ON)

JOGSPEED=50          'set JOG speed
FWD_JOG=0            'set Input0 as positive jog input
REV_JOG=1            'set Input1 as negative jog input
INVERT_IN(0,ON)
INVERT_IN(1,ON)
DATUM(3)             'Set Homing mode

```

	Axis0	Axis1
ATYPE	0	0
UNITS	10	10
ACCEL	1000	1000
DECEL	1000	1000
SPEED	100	100
CREEP	100	100
LSPEED	10	10
MERGE	0	0
SRAMP	10	10
DPOS	406.5000	314.8000
MPOS	406.5000	314.8000
ENDMOVE	406.5000	314.8000
FS_LIMIT	200000000	200000000
RS_LIMIT	-200000000	-200000000
DATUM_IN	-1	-1
FWD_IN	-1	-1
REV_IN	-1	-1
IDLE	-1	-1
LOADED	-1	-1
MSPEED	0	0
MTYPE	0	0
NTYPE	0	0
REMAIN	0	0
VECTOR_BUFFERED	0	0
VP_SPEED	0	0
AXISSTATUS	0h	0h
MOVE_MARK	1	1
MOVE_CURMARK	-1	-1
AXIS_STOPREASON	800h	800h
MOVES_BUFFERED	0	0

```
Connected to Controller:ZMC4xx-Simu Version:4.60-20180102.
Down to Controller Rom Success, 2021-07-28 09:14:09, Elapsed time: 187ms.
```

ZDevelop software supports 3 programming methods, Basic, PLC ladder of diagram and HMI configuration. Please note when using Basic language, it can run multiple Basic tasks. For PLC or HMI, it only supports one PLC task or one HMI task. Multi-task can be achieved among Basic, PLC and HMI.

ZDevelop software supports online simulation debug, and it has simulator "ZMC Simulator" and configuration program simulation tool "xplc screen".

1.1 System Requirements

To operate ZDevelop, minimum PC hardware configuration required as below:

Items	Minimum	Recommended

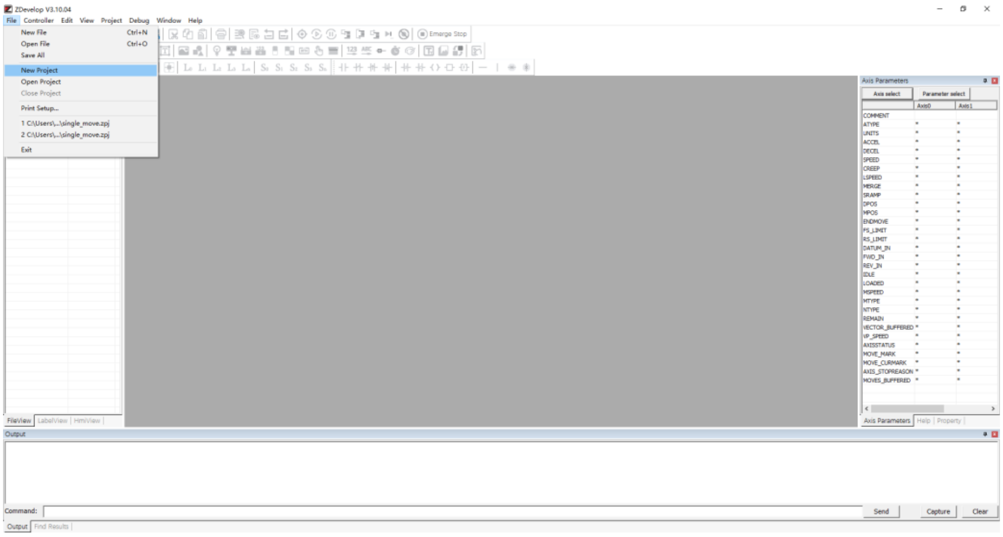
CPU	Pentium level, dominant frequency:450Mhz	Pentium level, dominant frequency:1GHz
Memory	64 MB	256MB
Hardware Disk	20MB	100MB
Operation System	Windows 98, Windows xp	Windows xp or win7.
Display	800x600/256 true color	1024x768/24 bits true color
Communication	RS232	RS232/USB/Ethernet (converted by HUB)

1.2 ZDevelop Usage Steps

Build a new folder in computer to save item that is going to be built. Open ZDevelop program software. Here take Zdevelop version V3.10 as example, please visit ZMOTION website for updating software. Website: zmotionglobal.com

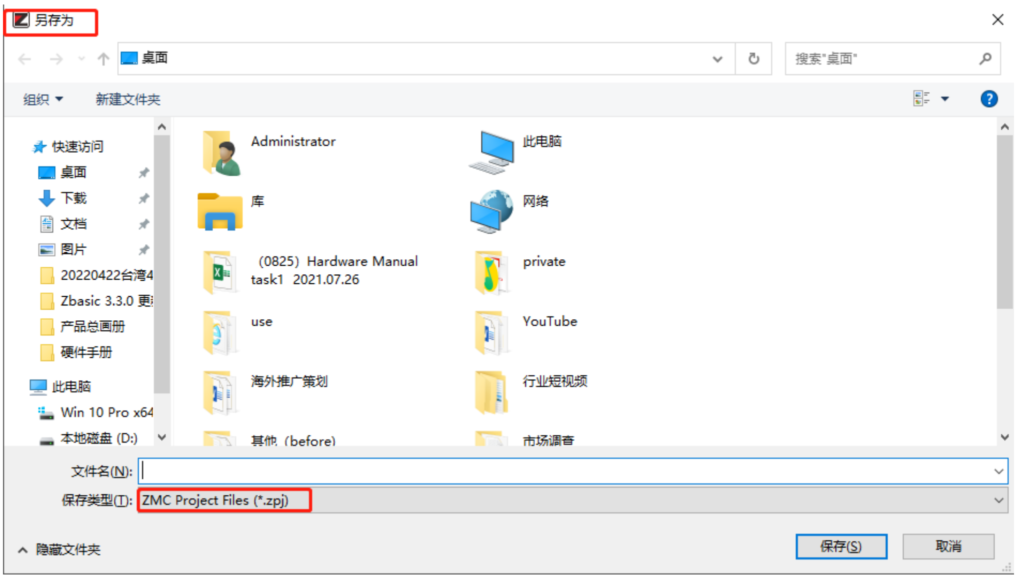
Basic process: build a new project → build a new file → add automatic run task NO. in file → edit the program → connect to controller → download the program into controller.

1. New build item: “File” in “Menu” → “New Project”.



	Axis0	Axis1
COMMENT		
ATYPE	*	*
UNITS	*	*
ACCEL	*	*
DECEL	*	*
SPEED	*	*
CREEP	*	*
LSPEED	*	*
MERGE	*	*
SRAMP	*	*
DPOS	*	*
MPOS	*	*
ENDMOVE	*	*
FS_LIMIT	*	*
RS_LIMIT	*	*
DATUM_IN	*	*
FWD_IN	*	*
REV_IN	*	*
IDLE	*	*
LOADED	*	*
MSPEED	*	*
MTYPE	*	*
NTYPE	*	*
REMAIN	*	*
VECTOR_BUFFERED	*	*
VP_SPEED	*	*
AXISSTATUS	*	*
MOVE_MARK	*	*
MOVE_CURMARK	*	*
AXIS_STOPREASON	*	*
MOVES_BUFFERED	*	*

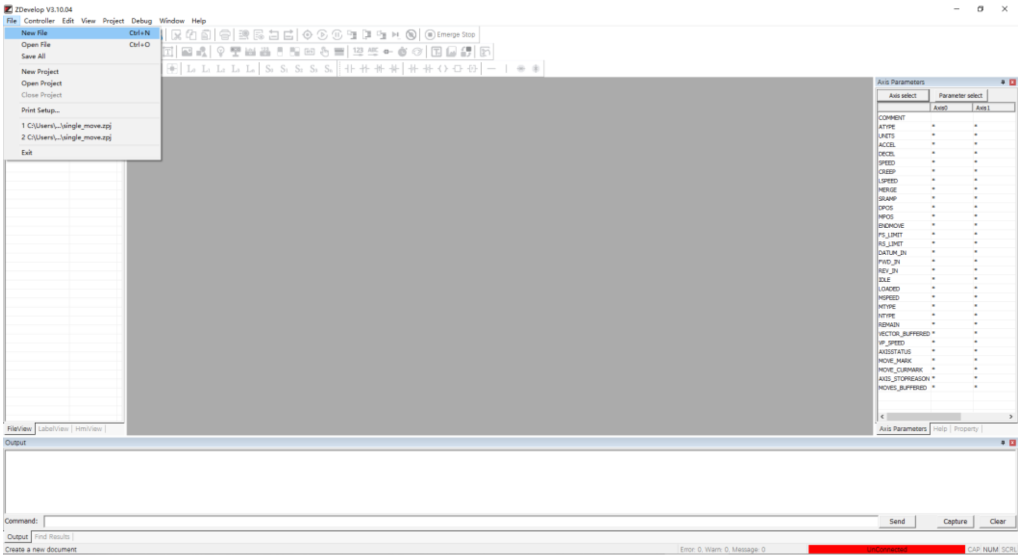
Click “New Project”, then “Save as…” will be jumped, select one folder and open it. Input folder’s name and save the project, pay attention to the suffix should be “.zpj”.



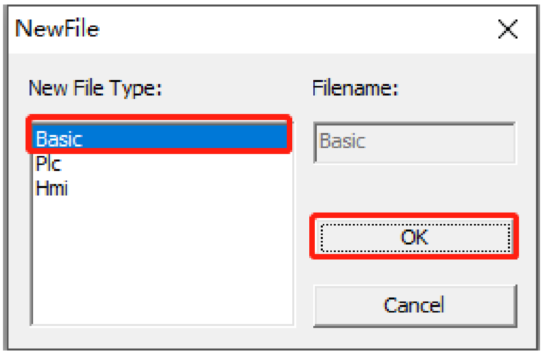
文件名(N):

保存类型(T): ZMC Project Files (*.zpj)

2. New build a file: “File” → “New File”.

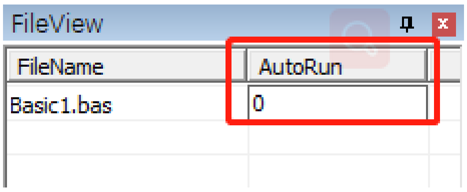


After clicking “New Project”, below jumping window will appear, which supports Basic/PLC/Hmi hybrid programming. Here selects the “Basic” file type and click “OK”.



Filename: Basic

3. Set file as automatic operation: please see the below picture, double click the right position “AutoRun” of “File”, and input task number is “0”.

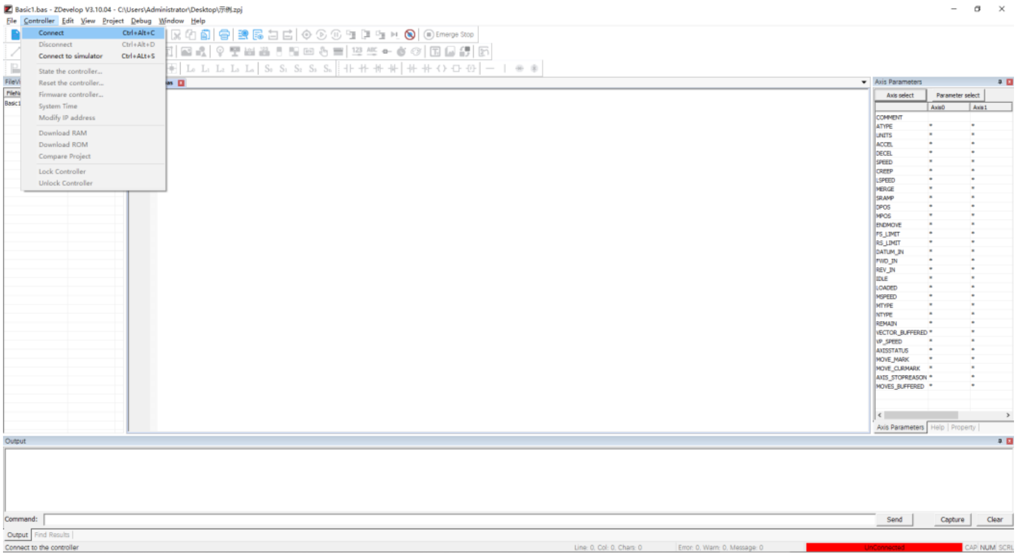


FileName	AutoRun
Basic1.bas	0

4. Program the procedure: when procedure is programmed, click “save” the file. New built Basic file will be saved automatically into the file in Project zpj.

5. Connect to Controller: program the procedure well in the input window, click “Controller” - “Connect”.

If there is no “Controller”, select connect to simulation, click “Connect” – “Connect to Simulator”. In this way, it can be connected to simulator, and there is hint showing simulator is connected successfully.



	Axis0	Axis1
COMMENT		
ATYPE	*	*
UNITS	*	*
ACCEL	*	*
DECEL	*	*
SPEED	*	*
CREEP	*	*
LSPEED	*	*
MERGE	*	*
SRAMP	*	*
DPOS	*	*
MPOS	*	*
ENDMOVE	*	*
FS_LIMIT	*	*
RS_LIMIT	*	*
DATUM_IN	*	*
FWD_IN	*	*
REV_IN	*	*
IDLE	*	*
LOADED	*	*
MSPEED	*	*
MTYPE	*	*
NTYPE	*	*
REMAIN	*	*
VECTOR_BUFFERED	*	*
VP_SPEED	*	*
AXISSTATUS	*	*
MOVE_MARK	*	*
MOVE_CURMARK	*	*
AXIS_STOPREASON	*	*
MOVES_BUFFERED	*	*

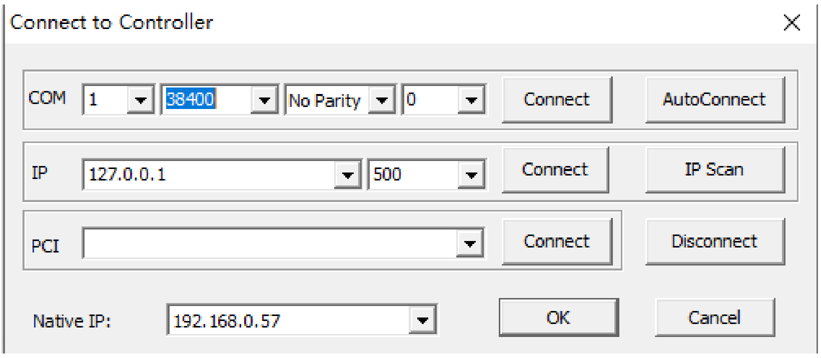
Command:

Connect to the controller

Line: 0, Col: 0, Chars: 0

Error: 0, Warn: 0, Message: 0

Click “Connect”, then “Connect to Controller” window will jump. And select serial port parameters or net port IP address, click “Connect”. When it is connected well, print information in Command and Output window: Connect to Controller: ZMC432 Version: 4.64-20170623.



COM 1 38400 No Parity 0

IP 127.0.0.1 500

PCI

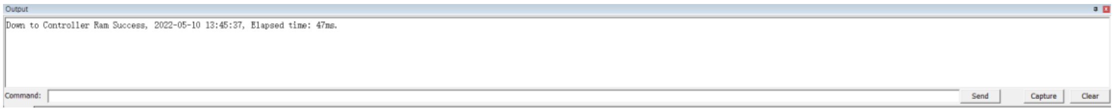
Native IP: 192.168.0.57

For the detailed method of serial port connection and network port connection, please refer to the next Link ZDevelop to Controller.

6. Download Program: click “Download RAM” or “Download ROM”. When it is downloaded

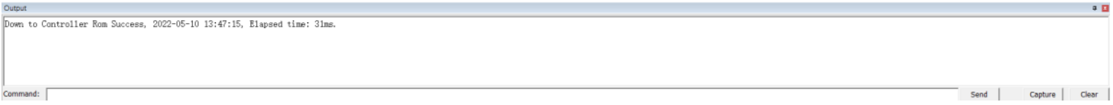
successfully, Command and Output window will give a hint. Program is downloaded into controller and will run automatically.

✓ Download RAM:



```
Down to Controller Ram Success, 2022-05-10 13:45:37, Elapsed time: 47ms.
```

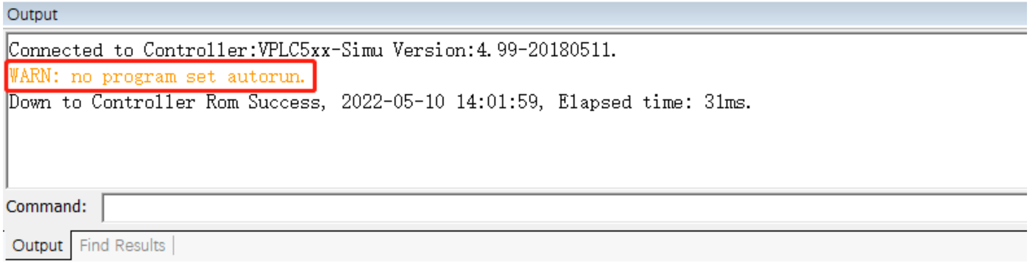
✓ Download ROM:



```
Down to Controller Rom Success, 2022-05-10 13:47:15, Elapsed time: 31ms.
```

Precautions:

- When open the project item, select the item zpj file. If only the Bas file is opened, program can’t be downloaded into controller.
- ZMC00x series controller don’t support Download RAM.
- When project is not built, only Bas file can’t be downloaded into controller.
- AutoRun 0 means the task number, task number 0 runs the procedure. Task number doesn’t have priority.
- If all files of whole project are not set the task number, when downloading into controller, system will give the indication: WARN: no program set autorun.

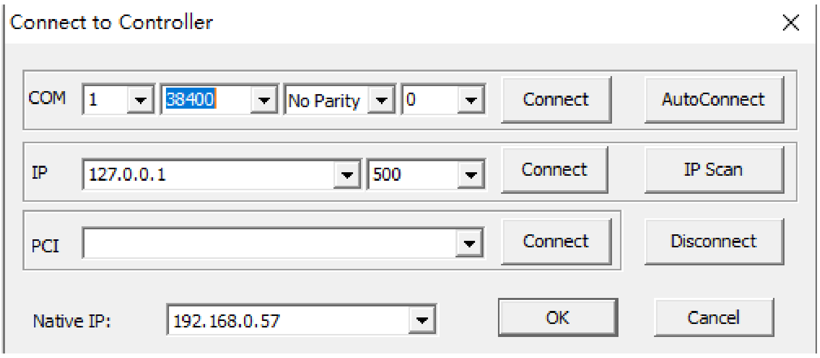


```
Connected to Controller:VPLC5xx-Simu Version:4.99-20180511.
WARN: no program set autorun.
Down to Controller Rom Success, 2022-05-10 14:01:59, Elapsed time: 31ms.
```

1.3 Link ZDevelop to Controller

Through “Controller” – “Connect”, it can connect with controller.

ZDevelop supports serial port and Ethernet connecting to controller.



Connect to Controller

COM 1 38400 No Parity 0

IP 127.0.0.1 500

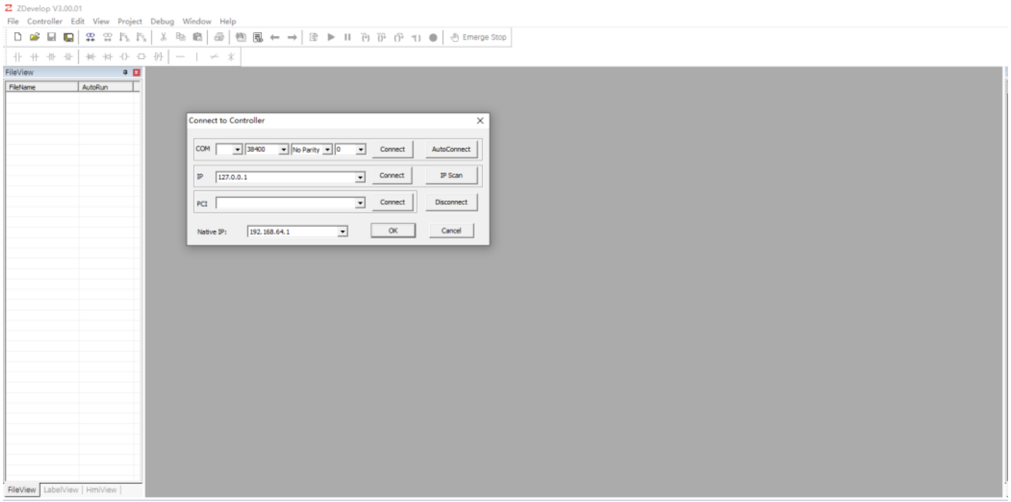
PCI

Native IP: 192.168.0.57

Through Serial Port:

Click menu: Controller > Connect, enter link configuration page, set serial parameters: com number, baud rate, parity bit, stop bit, then click "Connect" to link controller.

On the other hand, we can also link controller through USB: link USB port on PC side, and link serial port on controller side, in this way, we have to install related drive file on PC, to generate a virtual serial port mapped to USB connection, so we configure the virtual serial port same as actual.


Connect to Controller

COM: 38400, No Parity, 0

IP: 127.0.0.1

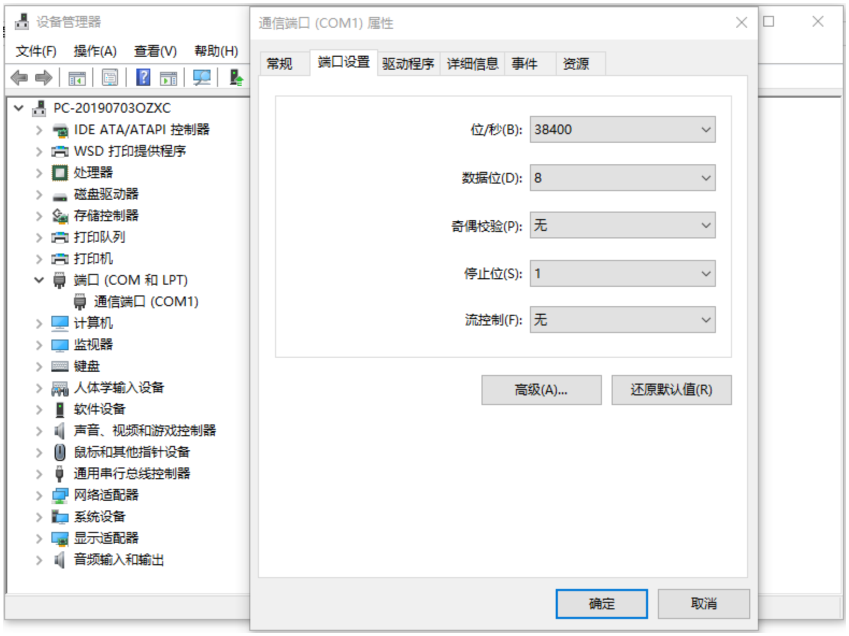
PCI:

Native IP: 192.168.64.1

If fail to link controller through serial port, possible reasons and solutions are as follow:

1.serial connection wire is not cross type;

2.COM configuration is not matched to PC Com, try to adjust the PC Com parameters as entering "device manage" page.


通信端口 (COM1) 属性

位/秒(B): 38400

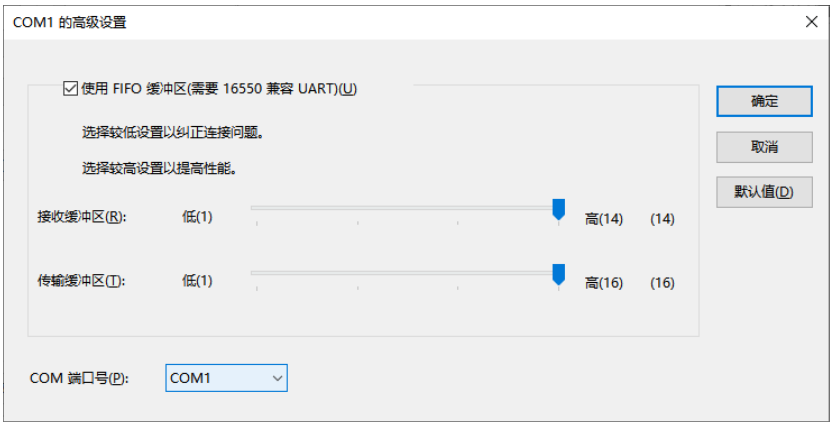
数据位(D): 8

奇偶校验(P): 无

停止位(S): 1

流控制(F): 无

We can also change Com number by entering "Advanced Setting" page of COM.



COM1 的高级设置

☑使用 FIFO 缓冲区(需要 16550 兼容 UART)(U)

选择较低设置以纠正连接问题。

选择较高设置以提高性能。

接收缓冲区(R): 低(1) 高(14) (14)

传输缓冲区(T): 低(1) 高(16) (16)

COM 端口号(P): COM1

3.When linked to controller, controller should be configured as MODBUS slave mode (default mode), and it will recover through power off – power on.

4.COM port on PC side was occupied by other application, like, serial port debug assistance.

5.Serial ports are not enough in PC.

6.Serial wire or PC is wrong.

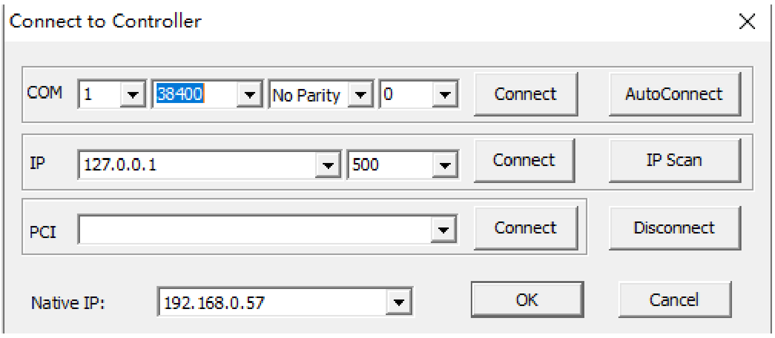
Through Ethernet:

When select IP in IP address list, it will find available controller IP address in current local ethernet range.

The default IP address is 192.168.0.11, “Controller” – “Connect” window can show this IP address, and it can be connected only when PC IP address is in same net segment as controller IP, which means the front 3 segments should be the same, the last segment is different.

VPLC5XX series controllers include 2 net ports, LAN1 and LAN2. (default address of LAN1: 192.168.0.11, default address of LAN2: 192.168.1.11)

Please see below, “Connect to controller” window, native IP can be checked quickly. Note: wired network card and wireless network card should be set separate IP.



Connect to Controller

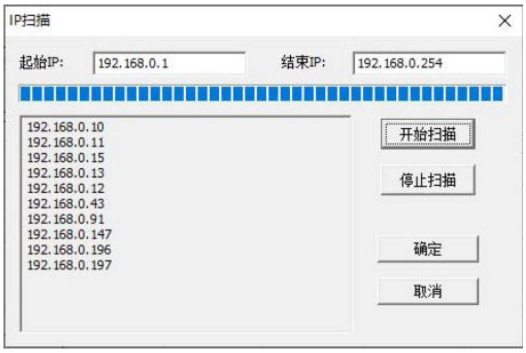
COM 1 38400 No Parity 0

IP 127.0.0.1 500

PCI

Native IP: 192.168.0.57

When there are multiple controllers in same network, it can be checked through IP scanning.



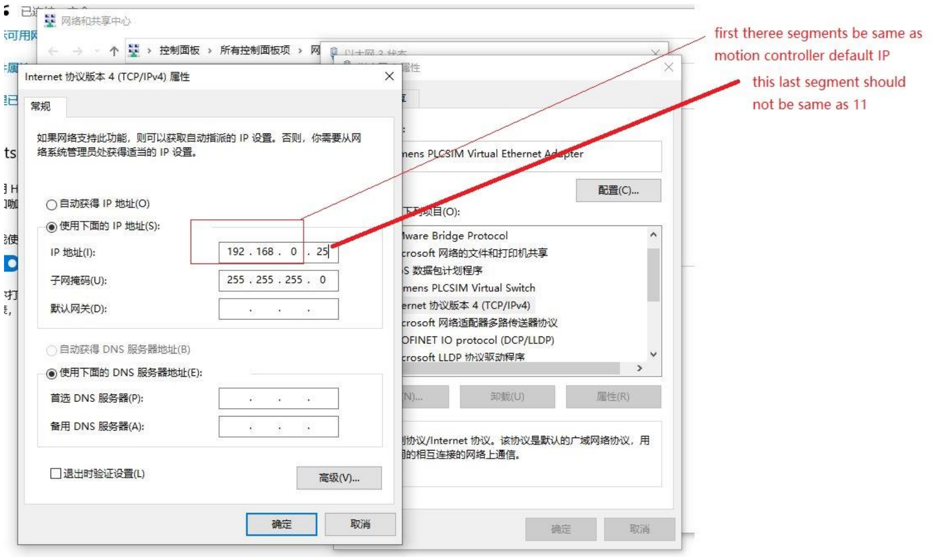
IP扫描

起始IP: 192.168.0.1 结束IP: 192.168.0.254

```
192.168.0.10
192.168.0.11
192.168.0.15
192.168.0.13
192.168.0.12
192.168.0.43
192.168.0.91
192.168.0.147
192.168.0.196
192.168.0.197
```

Modify IP address on PC side:

Check whether the computer's local IP protocol version 4 address is 192.168.0.xxx, the first three segments are consistent with the controller, and the last cannot be the same. The factory default IP of the controller is 192.168.0.11. If the third segment of the IP address is different, the corresponding subnet mask needs to be as 0. After setting, connect the software again.


Internet 协议版本 4 (TCP/IPv4) 属性

常规

如果网络支持此功能，则可以获取自动指派的 IP 设置。否则，你需要从网络系统管理员处获得适当的 IP 设置。

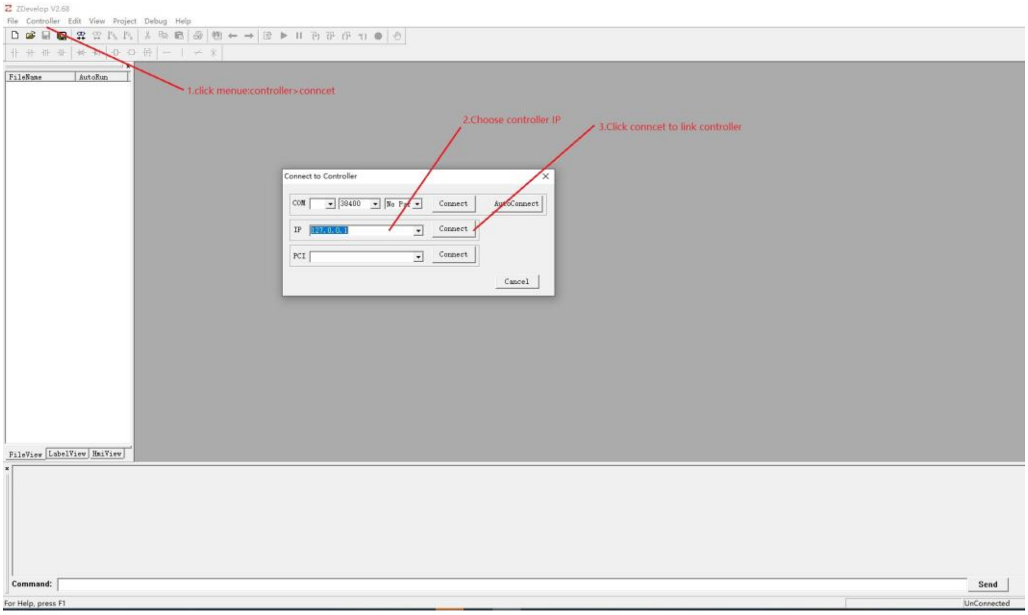
○ 自动获得 IP 地址(O)
● 使用下面的 IP 地址(S):
IP 地址(I): 192 . 168 . 0 . 25
子网掩码(U): 255 . 255 . 255 . 0
默认网关(D): . . .

○ 自动获得 DNS 服务器地址(B)
● 使用下面的 DNS 服务器地址(E):
首选 DNS 服务器(P): . . .
备用 DNS 服务器(A): . . .

☐ 退出时验证设置(L)

first theree segments be same as motion controller default IP

this last segment should not be same as 11
网络和共享中心
控制面板 › 所有控制面板项 › 网
高级(V)...
确定
取消
nens PLCSIM Virtual Ethernet Adapter
配置(C)...
下列项目(O):
lware Bridge Protocol
crosoft 网络的文件和打印机共享
S 数据包计划程序
mens PLCSIM Virtual Switch
ernet 协议版本 4 (TCP/IPv4)
crosoft 网络适配器多路传送器协议
OFINET IO protocol (DCP/LLDP)
crosoft LLDP 协议驱动程序
卸载(U)
属性(R)
协议/Internet 协议。该协议是默认的广域网络协议，用
的相互连接的网络上通信。


1.click menue:controller>conncet

2.Choose controller IP

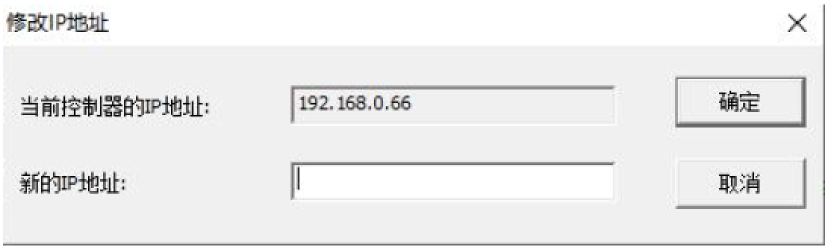
3.Click conncet to link controller

Connect to Controller
COM 38400 No Par
IP 127.0.0.1
PCI

Modify IP address on controller:

If the controller IP is modified and is not in the network segment 192.168.0.XXX, you can only connect the controller through the serial port first, obtain the controller IP address, and then modify the local IP or the controller IP so that the two are on the same network part.

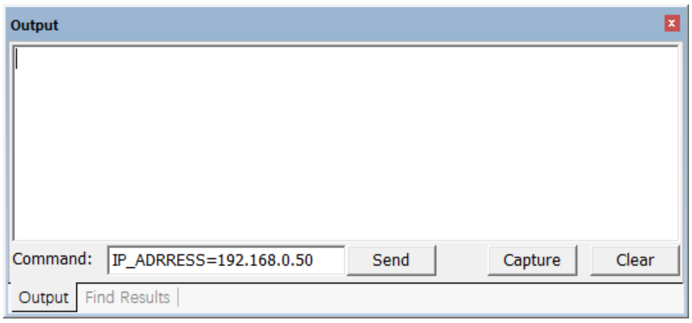
There are many ways to modify the IP address of the controller. You can click "Controller" - "Modify IP Address" in the menu bar, and the following window will pop up. At this time, the current controller IP will be displayed, and you can directly enter the new IP address in the window.


修改IP地址

当前控制器的IP地址: 192.168.0.66

新的IP地址:

Or in “Controller”-“State the controller’ to see, or through online command, then use IP_ADDRESS to modify IP.


Output

Command: IP_ADRRESS=192.168.0.50

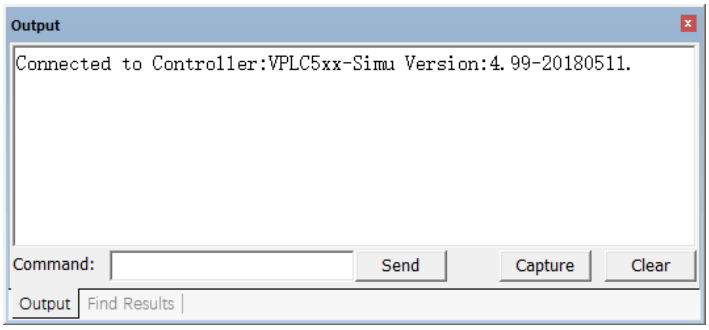
After modifying, controller and ZDevelop will disconnect, now, select new IP address to connect.

1.4 Link ZDevelop to simulator

Online simulation is valid in ZDevelop, it enables project simulation or program debug when hardware is not available.

Click menu: Controller > Connect to simulator, to open link simulator, also it can be connected through IP:127.0.0.1.

After successful connection, followed information will be printed:

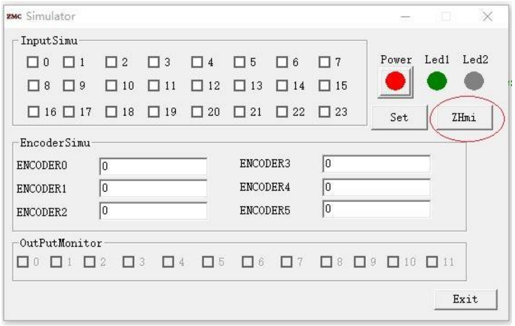

Output

```
Connected to Controller:VPLC5xx-Simu Version:4.99-20180511.
```

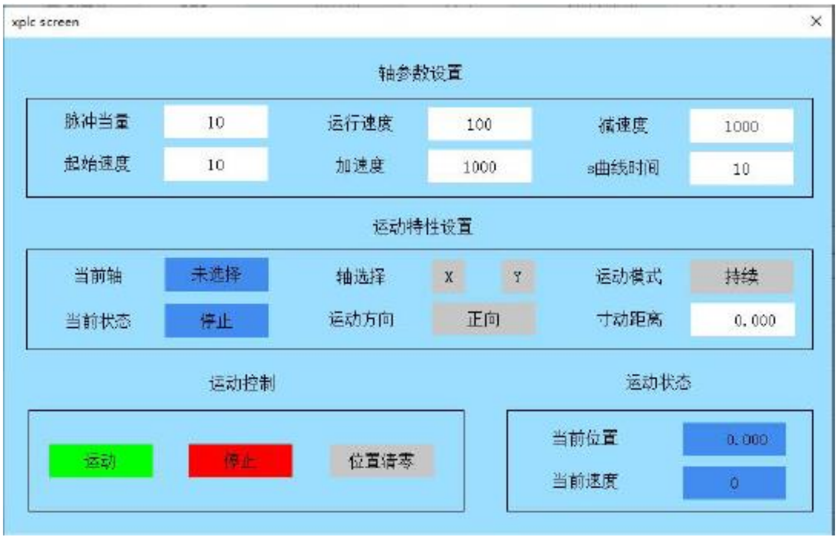
Command:

Please note: old versions don’t support simulation visual instructions, so update latest version.

If project contains HMI file, click “Zhmi” button to pop HMI interface.



One certain HMI example project is below:



xplc screen

轴参数设置

脉冲当量 10 运行速度 100 减速度 1000

起始速度 10 加速度 1000 s曲线时间 10

运动特性设置

当前轴 未选择 轴选择 X Y 运动模式 持续

当前状态 停止 运动方向 正向 寸动距离 0.000

运动控制

运动 停止 位置清零

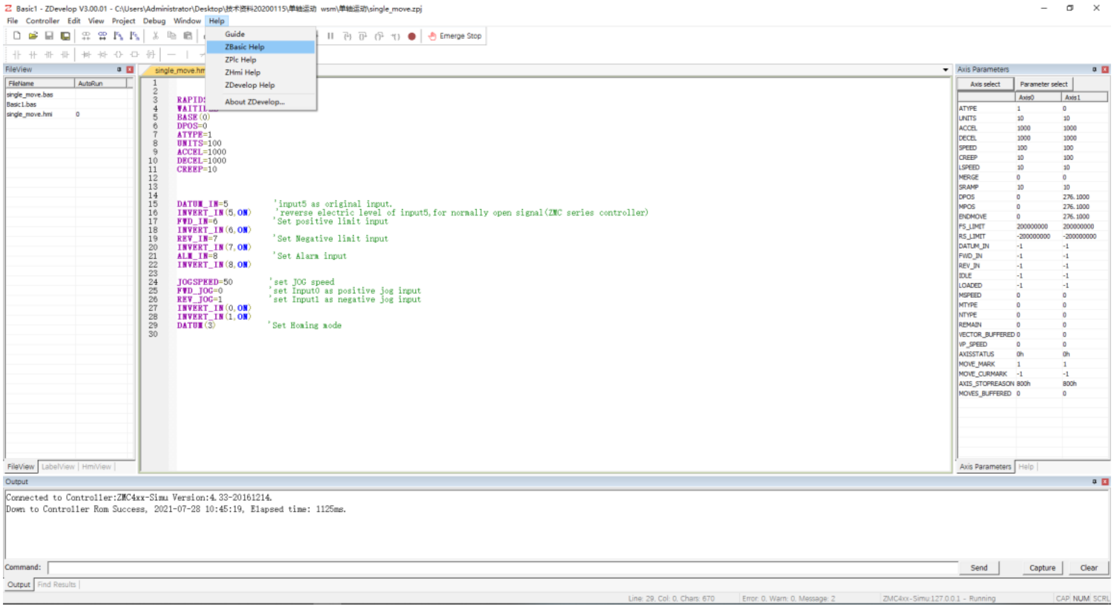
运动状态

当前位置 0.000

当前速度 0

1.5 Help

Click "Help" on the menu bar, open the controller usage documentation, ZBasic grammar help document, ZPlC grammar help document, ZHmi grammar help document, ZVision visual syntax help document, and ZDevelop development environment help document. In the document, you can view the description of all related commands and the introduction of some functions.



Basic1 - ZDevelop V3.00.01 - C:\Users\Administrator\Desktop\技术资料20200115\单轴运动 wsm\单轴运动\single_move.zpj

```


RAPID
WAITIL
BASE(0)
DPOS=0
ATYPE=1
UNITS=100
ACCEL=1000
DECEL=1000
CREEP=10



DATUM_IN=5          'input5 as original input.
INVERT_IN(5,ON)      'reverse electric level of input5,for normally open signal(ZMC series controller)
FWD_IN=6            'Set positive limit input
INVERT_IN(6,ON)
REV_IN=7            'Set Negative limit input
INVERT_IN(7,ON)
ALM_IN=8            'Set Alarm input
INVERT_IN(8,ON)

JOGSPEED=50        'set JOG speed
FWD_JOG=0          'set Input0 as positive jog input
REV_JOG=1          'set Input1 as negative jog input
INVERT_IN(0,ON)
INVERT_IN(1,ON)
DATUM(3)           'Set Homing mode

```

	Axis0	Axis1
ATYPE	1	0
UNITS	10	10
ACCEL	1000	1000
DECEL	1000	1000
SPEED	100	100
CREEP	10	100
LSPEED	10	10
MERGE	0	0
SRAMP	10	10
DPOS	0	276.1000
MPOS	0	276.1000
ENDMOVE	0	276.1000
FS_LIMIT	200000000	200000000
RS_LIMIT	-200000000	-200000000
DATUM_IN	-1	-1
FWD_IN	-1	-1
REV_IN	-1	-1
IDLE	-1	-1
LOADED	-1	-1
MSPEED	0	0
MTYPE	0	0
NTYPE	0	0
REMAIN	0	0
VECTOR_BUFFERED	0	0
VP_SPEED	0	0
AXISSTATUS	0h	0h
MOVE_MARK	1	1
MOVE_CURMARK	-1	-1
AXIS_STOPREASON	800h	800h
MOVES_BUFFERED	0	0

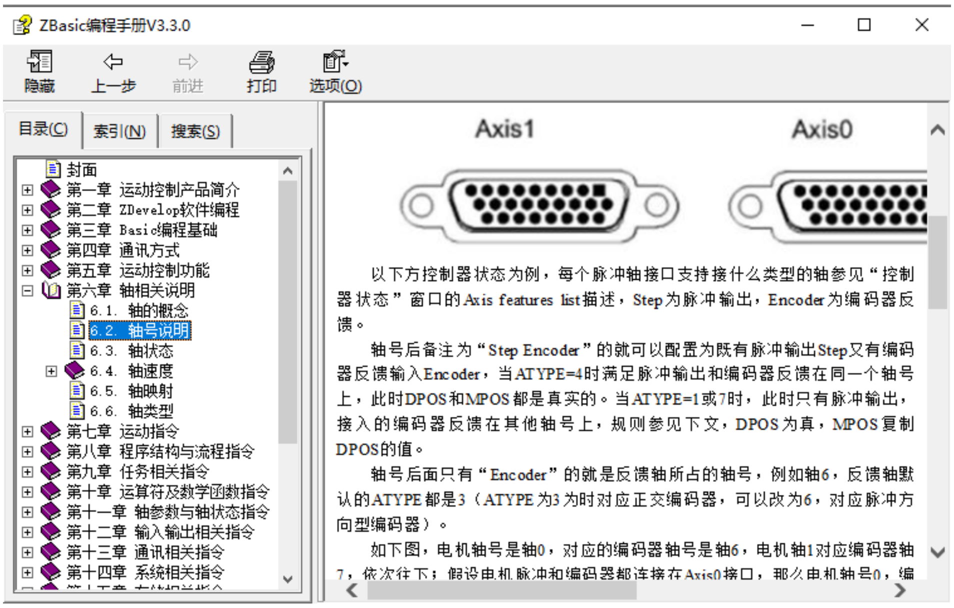
```
Connected to Controller:ZMC4xx-Simu Version:4.33-20161214.
Down to Controller Rom Success, 2021-07-28 10:45:19, Elapsed time: 1125ms.
```

Line: 29, Col: 0, Chars: 670 Error: 0, Warn: 0, Message: 2 ZMC4xx-Simu:127.0.0.1 - Running CAP NUM SCRL

In the Basic program, select an instruction and then press the F1 key to quickly open the instruction detailed description page in the help document in the help view window, as shown in the figure below. The F1 shortcut key is also supported under PLC and HMI programming.

In the help interface, hold down Ctrl and scroll the mouse to zoom in and out of the displayed content of the help document. The HOME key jumps to the beginning of the current line, the END key jumps to the end of the current line, pageup and pagedown are page up and down.

ZBasic Help:



封面
第一章 运动控制产品简介
第二章 ZDevelop软件编程
第三章 Basic编程基础
第四章 通讯方式
第五章 运动控制功能
第六章 轴相关说明
6.1. 轴的概念
6.2. 轴号说明
6.3. 轴状态
6.4. 轴速度
6.5. 轴映射
6.6. 轴类型
第七章 运动指令
第八章 程序结构与流程指令
第九章 任务相关指令
第十章 运算符及数学函数指令
第十一章 轴参数与轴状态指令
第十二章 输入输出相关指令
第十三章 通讯相关指令
第十四章 系统相关指令

以下方控制器状态为例，每个脉冲轴接口支持接什么类型的轴参见“控制器状态”窗口的Axis features list描述，Step为脉冲输出，Encoder为编码器反馈。

轴号后备注为“Step Encoder”的就可以配置为既有脉冲输出Step又有编码器反馈输入Encoder，当ATYPE=4时满足脉冲输出和编码器反馈在同一个轴号上，此时DPOS和MPOS都是真实的。当ATYPE=1或7时，此时只有脉冲输出，接入的编码器反馈在其他轴号上，规则参见下文，DPOS为真，MPOS复制DPOS的值。

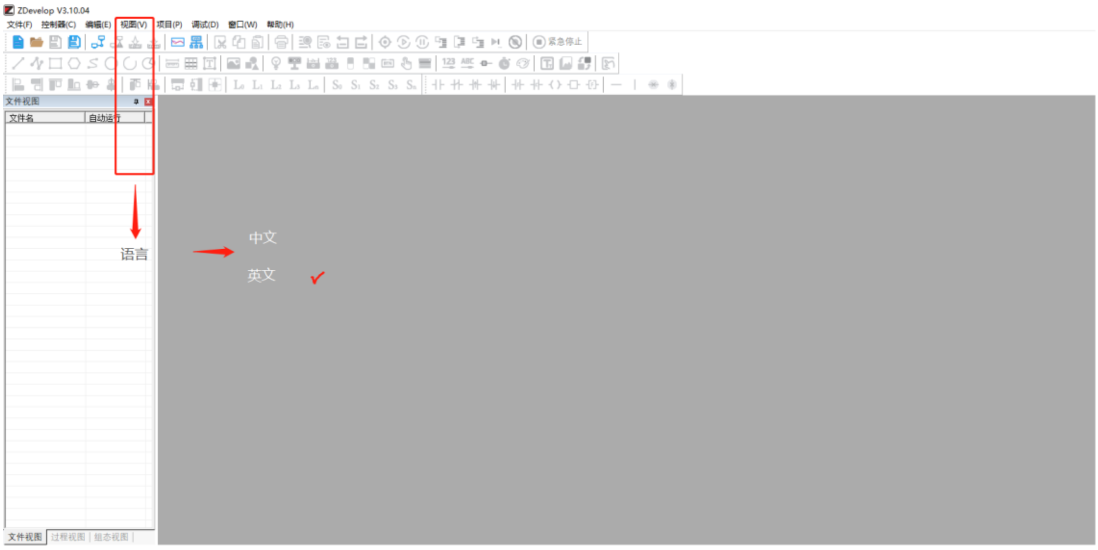
轴号后面只有“Encoder”的就是反馈轴所占的轴号，例如轴6，反馈轴默认的ATYPE都是3（ATYPE为3为时对应正交编码器，可以改为6，对应脉冲方向型编码器）。

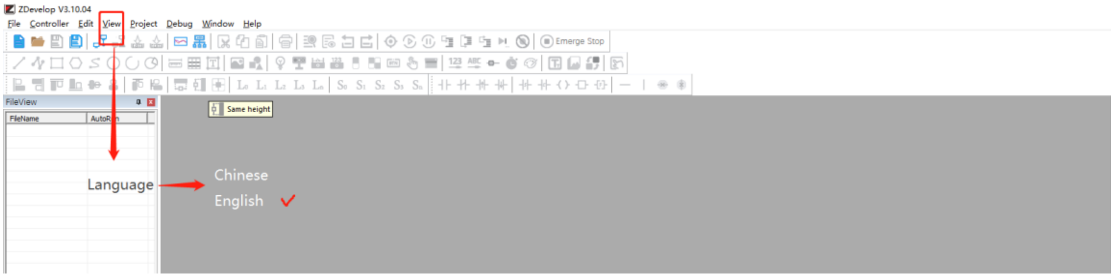
如下图，电机轴号是轴0，对应的编码器轴号是轴6，电机轴1对应编码器轴7，依次往下；假设电机脉冲和编码器都连接在Axis0接口，那么电机轴号0，编

(but there is no English help document in software, it can be checked here).

1.6 Language Switch

Now, ZDevelop supports Chinese and English, through “视图”-“语言”, then restart.


ZDevelop V3.10.04
文件(F) 控制器(C) 编辑(E) 视图(V) 项目(P) 调试(D) 窗口(W) 帮助(H)
文件视图
文件名
自动运行
语言
中文
英文 ✓



Chapter II Controller Operation

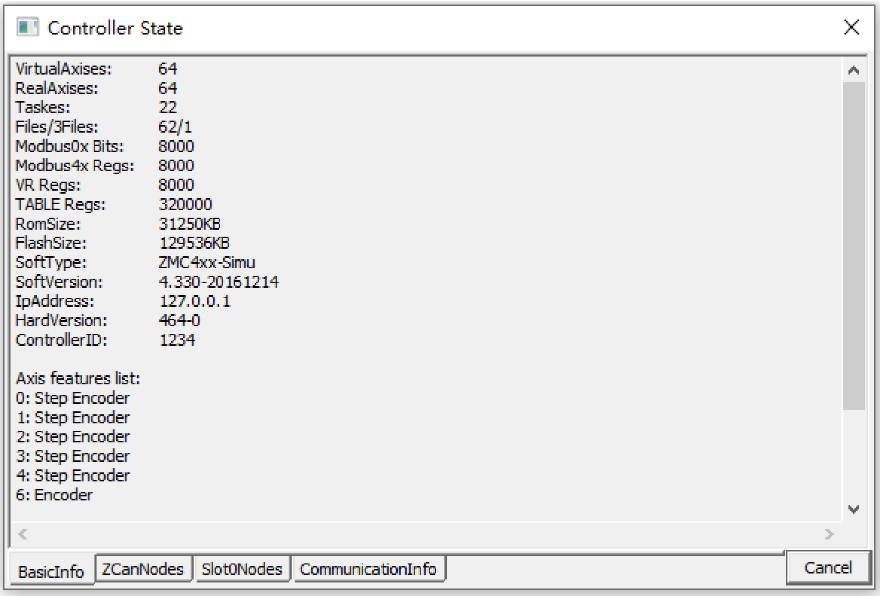
Controller status or configuration can be accessed through ZDevelop if it was built with controller.

2.1 Controller Status

Click menu controller > State the controller to view controller status, as below:

"Controller State" shows controller status information, including controller basic information, ZCan node status, slot node status and communication configuration. Node state can show the axis number of connected device, start IO number, etc.

Basic information includes the max virtual axes, max motor axes, task, file, space of each register, procedure volume, controller model, software version NO. and time, IP address, hardware version NO., controller hardware ID and configured type and map on each axis, etc.



```
VirtualAxises:     64
RealAxises:        64
Taskes:            22
Files/3Files:      62/1
Modbus0x Bits:     8000
Modbus4x Regs:     8000
VR Regs:           8000
TABLE Regs:        320000
RomSize:           31250KB
FlashSize:         129536KB
SoftType:          ZMC4xx-Simu
SoftVersion:       4.330-20161214
IpAddress:         127.0.0.1
HardVersion:       464-0
ControllerID:      1234

Axis features list:
0: Step Encoder
1: Step Encoder
2: Step Encoder
3: Step Encoder
4: Step Encoder
6: Encoder
```

VirtualAxises: max virtual axes

RealAxises: max actual axes

Taskes: max tasks

Files/3Files: max files/max .Z3P file

Modbus0x Bits: max modbus bits

Modbus4x Regs: max modbus words

VR Regs: max VR number

TABLE Regs: max Table number

RomSize: Rom size

FlashSize: Flash size

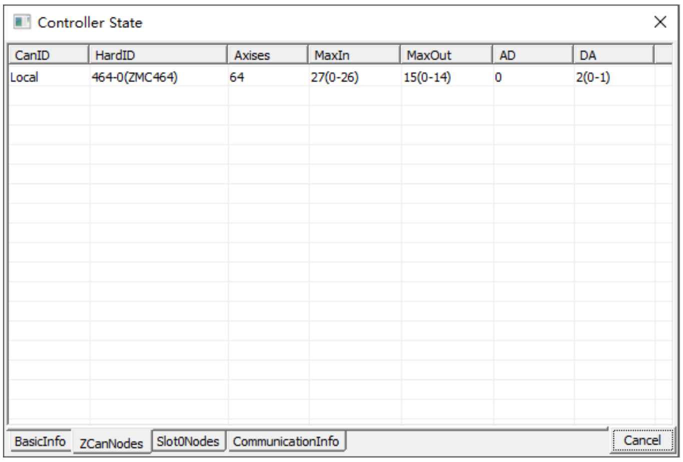
SoftType: software type

SoftVersion: system version and firmware version

IpAddress: controller IP address

HardVersion: hardware version
ControllerID: controller unique ID
Axis features list: axes type list

ZCan Node:


Controller State

CanID	HardID	Axises	MaxIn	MaxOut	AD	DA
Local	464-0(ZMC464)	64	27(0-26)	15(0-14)	0	2(0-1)

The first line of LOCAL node shows that hardware is VPLC516E controller, the first line corresponds to the local resources of the controller, supports 64 axes, a total of 27 digital inputs, numbered IN(0)-IN(26), and a total of 15 digital outputs , numbered OP(0)-OP(26), without AD, 2 DA, numbered DA(0)-DA(1).

Lines 2 to N display information about the extension modules of the ZCan node.

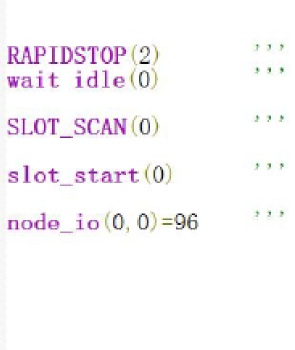
For example, the hardware of CanID node 2 is ZAIO0802, no digital input, no digital output, AD has 8, numbered AIN(24)-AIN(31), DA has 2, numbered AOUT(12)-AOUT(13).

The hardware of CanID node 4 is ECI0016PA, 8 digital inputs, numbered IN(80)-IN(87), 8 digital outputs, numbered OP(80)-OP(87), AD has 12, numbered AIN(40)-AIN(51), DA has 2, numbered AOUT(20)-AOUT(21).

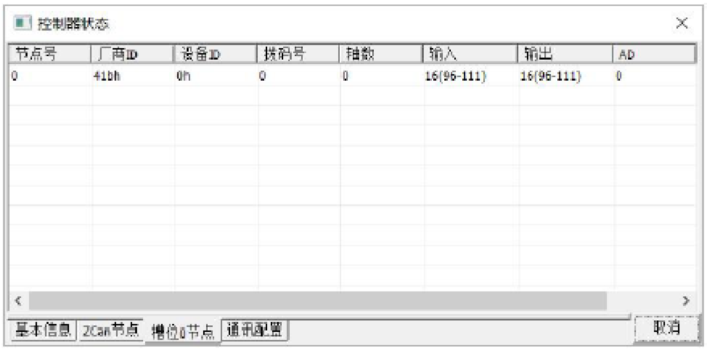
Slot 0 node:

Slot 0 is for EtherCAT in EtherCAT controller, for RTEX in RTEX controller;

If EtherCAT and RTEX both are supported in controller, then Slot 0 is for EtherCAT, Slot 1 is for RTEX.



```
RAPIDSTOP(2)          '''
wait idle(0)          '''

SLOT_SCAN(0)          '''

slot_start(0)         '''

node_io(0,0)=96       '''
```


控制器状态

节点号	厂商ID	设备ID	拨码号	轴数	输入	输出	AD
0	41bh	0h	0	0	16(96-111)	16(96-111)	0

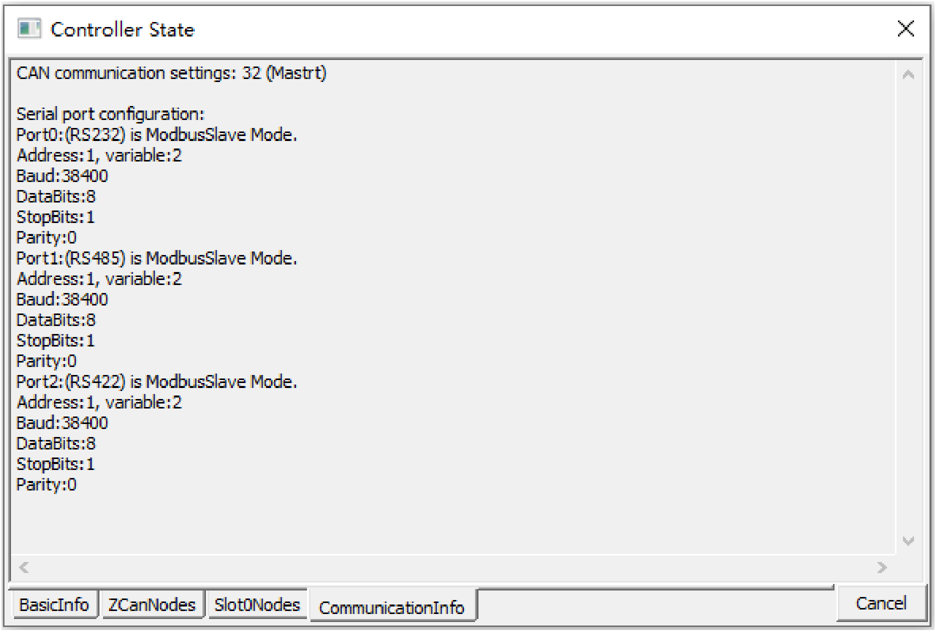
The left is the most basic EtherCAT initialization procedure.

Right shows that Lines 2 to N include the extension modules of the ZCan node.

See below, The hardware manufacturer ID of EtherCAT node 0 is 41bh (manufacturer ID of ZMOTION), the device ID of hardware manufacturer is 0h (product ID0 of ZMOTION, the corresponding product is EIO1616), dial number 0, no axis expansion, digital quantity 16 inputs, numbered IN(96)-IN(111), 16 digital outputs, numbered OP(96)-OP(111), no AD, no DA.

Controller communication configuration:

In Communication page, information of CAN, RS232, RS485 and RS422 is shown:



```
CAN communication settings: 32 (Mastrt)

Serial port configuration:
Port0:(RS232) is ModbusSlave Mode.
Address:1, variable:2
Baud:38400
DataBits:8
StopBits:1
Parity:0
Port1:(RS485) is ModbusSlave Mode.
Address:1, variable:2
Baud:38400
DataBits:8
StopBits:1
Parity:0
Port2:(RS422) is ModbusSlave Mode.
Address:1, variable:2
Baud:38400
DataBits:8
StopBits:1
Parity:0
```

CAN Communication Setting：

When set CANIO_ADDRESS=32, CANIO_ENABLE=1, it means controller is as ZCAN master, communication ratio is 500K, Can is enabled.(see details in ZBasic).

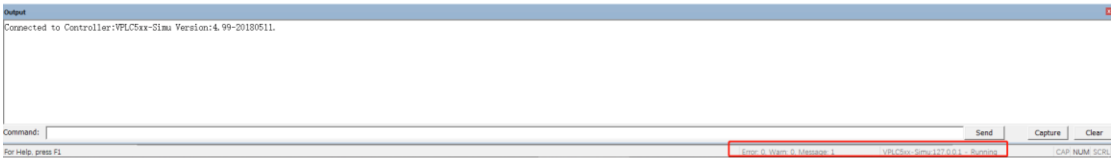
If need to modify communication setting, just set value of CANIO_ADDRESS, CANIO_ENABLE etc.

Port0: RS232, as slave, address is 1, VR and Modbus storage are independent;

Port0: RS485, as slave, address is 1, VR and Modbus storage are independent;

See details by instruction: SETCOM in ZDevelop.

After connecting the controller, you can check the model, IP address and running status of the currently connected controller by printing the information in the status bar below. You can also view the specific content of errors, warnings, and messages through the "Command and Output" window.

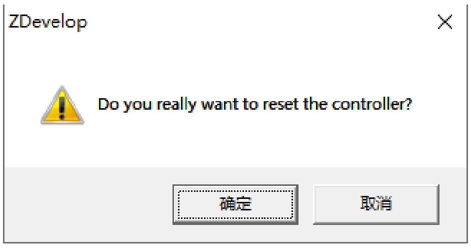


```
Connected to Controller:VPLC5xx-Simu Version:4.99-20180511.
```

Error: 0, Warn: 0, Message: 1   VPLC5xx-Simu:127.0.0.1 - Running

2.2 Controller Reset

After connecting controller, click menu bar “Controller” – “Reset the controller” to open below window, then click “确定”. Controller will restart after power on, then reconnect to controller.


ZDevelop

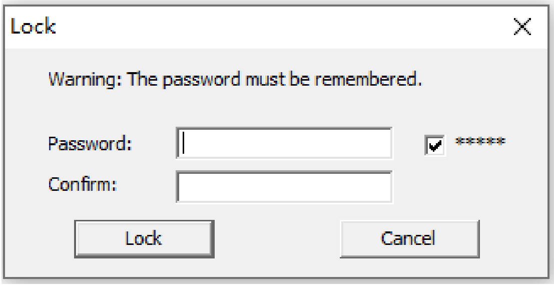
Do you really want to reset the controller?

2.3 Lock or Unlock Controller

Controller lock is used to protect code in controller, under lock status, project can not be loaded into controller, code in controller also can’t be accessed, but zar file is allowed to be loaded into controller, enable developer to capsulate code in zar file through ZDevelop and send it to customer, then customer upgrade controller by loading zar file into controller through independent tool: Zardown.exe(for customer use).

Irreversible algorithm is used for encryption, password can’t be discovered once forgotten.

Click menu **controller>lock controller** to set the controller lock password, after lock is done, we can’t debug controller anymore unless it is unlocked again by type into right password:

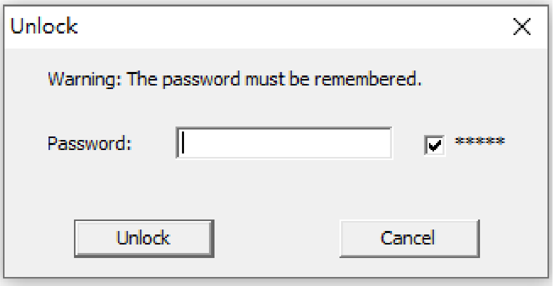

Lock

Warning: The password must be remembered.

Password: ☑ *****

Confirm:

The same we can enter controller unlock page:


Unlock

Warning: The password must be remembered.

Password: ☑ *****

2.4 Command / Output

Command and Output can check controller parameters control axis motion, print procedure running result, output procedure errors, etc. print and output function in program is given by software developer (through ?, PRINT, WARN, ERROR, TRACE, etc).

Note: they should be in English format. ERRSWITCH is used to control output of TRACE, WARN and ERROR, different values indicate different output modes:

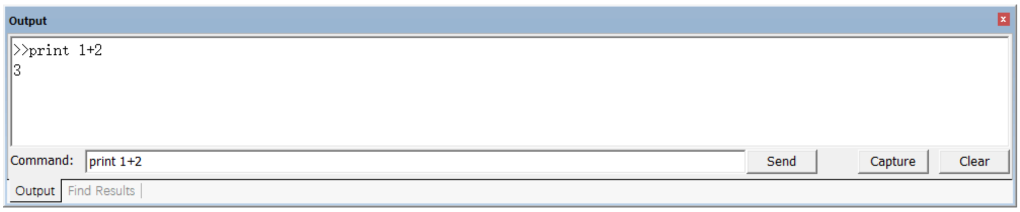
0: Disable TRACE, WARN and ERROR;

1: Enable ERROR;

2: Enable WARN and ERROR;

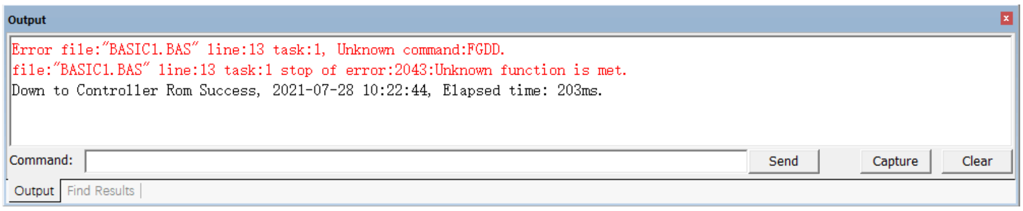
3: Enable TRACE, WARN and ERROR.

Online command and output window is below: >>: commands input from Command box; Capture: Save all Output box information in PC; Clear: Clear all information in Output box.



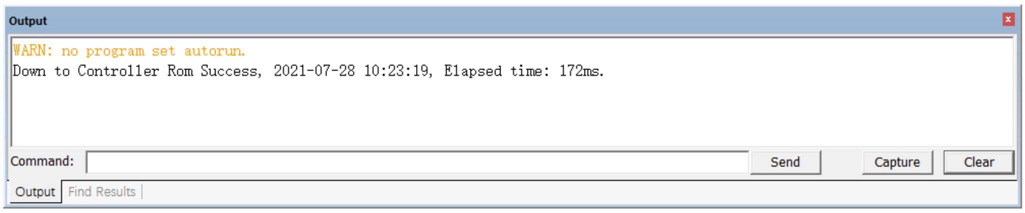
```
>>print 1+2
3
```

ERROR information shown in red color as follow:



```
Error file:"BASIC1.BAS" line:13 task:1, Unknown command:FGDD.
file:"BASIC1.BAS" line:13 task:1 stop of error:2043:Unknown function is met.
Down to Controller Rom Success, 2021-07-28 10:22:44, Elapsed time: 203ms.
```

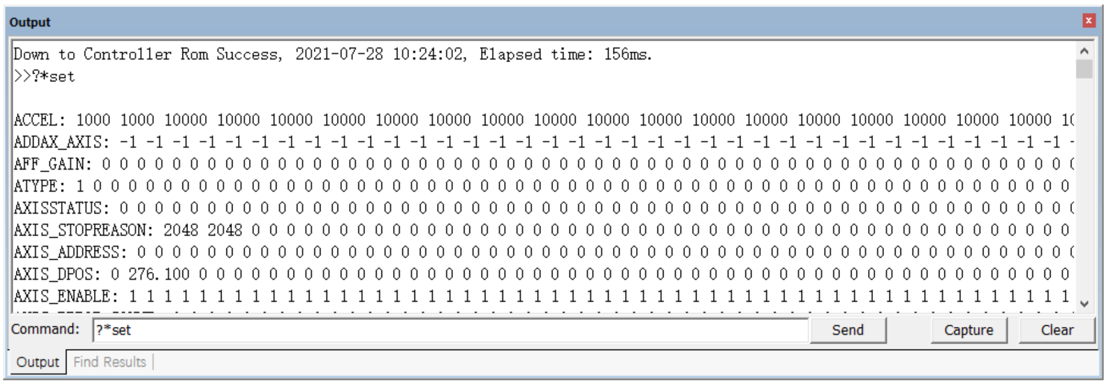
WARN information shown in yellow color as follow:



```
WARN: no program set autorun.
Down to Controller Rom Success, 2021-07-28 10:23:19, Elapsed time: 172ms.
```

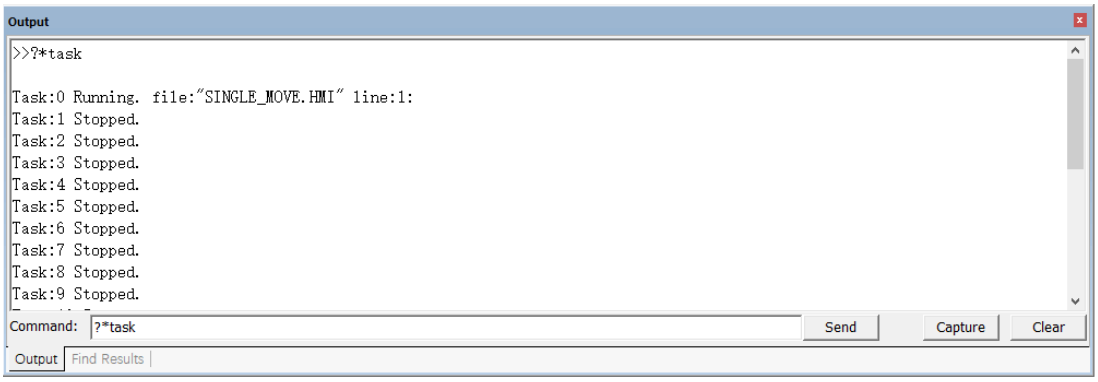
Common used print & check instructions:

?*SET: print axes and system parameters.



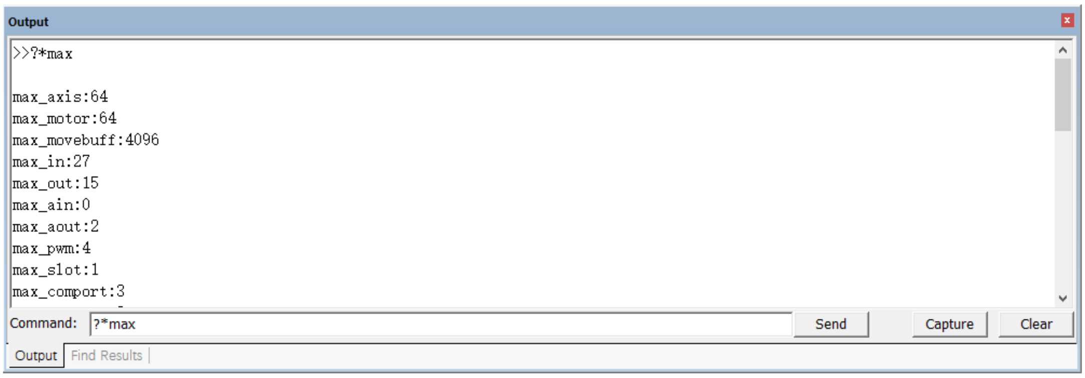
```
Down to Controller Rom Success, 2021-07-28 10:24:02, Elapsed time: 156ms.
>>?*set

ACCEL: 1000 1000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 1(
ADDAX_AXIS: -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -
AFF_GAIN: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (
ATYPE: 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AXISSTATUS: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (
AXIS_STOPREASON: 2048 2048 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AXIS_ADDRESS: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (
AXIS_DPOS: 0 276.100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AXIS_ENABLE: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

?*TASK: print task status, including error task number and lines;



```
>>?*task

Task:0 Running. file:"SINGLE_MOVE.HMI" line:1:
Task:1 Stopped.
Task:2 Stopped.
Task:3 Stopped.
Task:4 Stopped.
Task:5 Stopped.
Task:6 Stopped.
Task:7 Stopped.
Task:8 Stopped.
Task:9 Stopped.
```

?*MAX: print all controller parameters;



```
>>?*max

max_axis:64
max_motor:64
max_movebuff:4096
max_in:27
max_out:15
max_ain:0
max_aout:2
max_pwm:4
max_slot:1
max_comport:3
```

max_axis: 64   max axes, including virtual axes.

max_motor:64   max motor axes

max_movebuff:4096   max motion buffer of each axis or each axis group

max_in:27   inputs and max inputs after extended

max_out:15   outputs and max outputs after extended

max_ain:0   analog inputs and max analog input after extended

max_aout:2   analog outputs and max analog outputs after extended

max_pwm:4   max PWM channels

max_slot:1   max Bus channels

max_comport:3   max serial ports

max_ethport: max Ethernet ports

max_ethcustom: max self-defined Ethernet ports

max_ethiport: max Ethernet ports for communication between Zmotion controllers.

max_flashnum: max FLASH blocks;

max_flashsize: space of each FLASH block;

max_nand: max NandFlash space;

max_nandremain: remaining NandFlash space;

max_pswitch: max soft position comparers;

max_file: max files of system

max_3file:max z3p file of system;

max_task: max tasks of controller

max_timer: max timers

max_loopnest: max internal recycles and selections

max_callstack: max stack levels called by SUB process

max_local of one sub: local variables of SUB process

max_vr: max VR number;

max_table: max TABLE number;

max_modbusbit: max MODBUS_BIT number;

max_modbusreg: max MODBUS_REG number;

max_var: max variables; (including global and file variables)

max_array: max array number; (including global and file array)

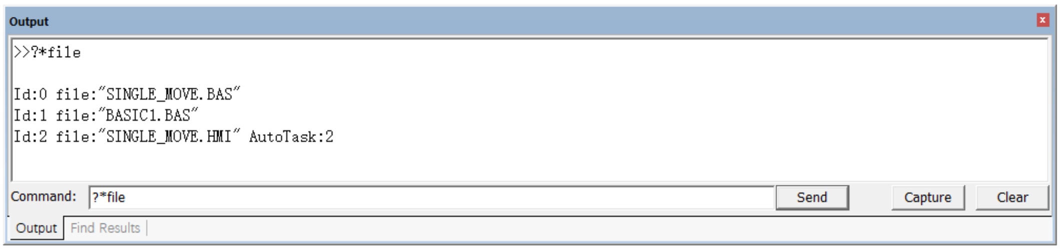
max_arrayspace: total array space;

max_sub:max SUB numbers;

max_edgescan: max falling or rising edge scanning threads;

max_lablelength: max char length of array and variable name;
max_hmi: max HMI numbers and HMI screen size;
function support: motion functions supported;

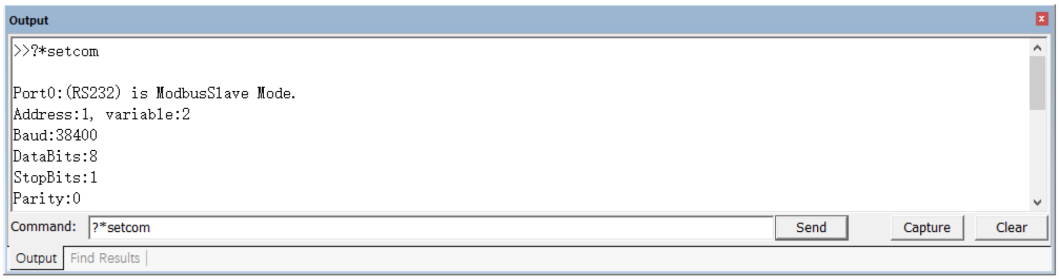
?*FILE:print file tasks information.



```
>>?*file

Id:0 file:"SINGLE_MOVE.BAS"
Id:1 file:"BASIC1.BAS"
Id:2 file:"SINGLE_MOVE.HMI" AutoTask:2
```

Command: ?*file

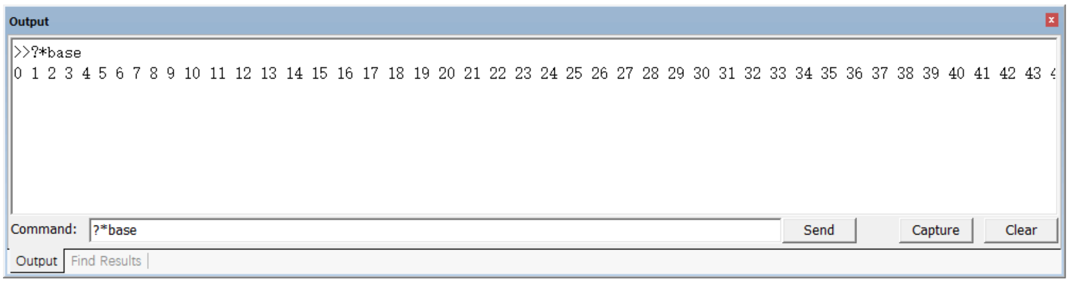
?*SETCOM: print serial ports information



```
>>?*setcom

Port0:(RS232) is ModbusSlave Mode.
Address:1, variable:2
Baud:38400
DataBits:8
StopBits:1
Parity:0
```

Command: ?*setcom

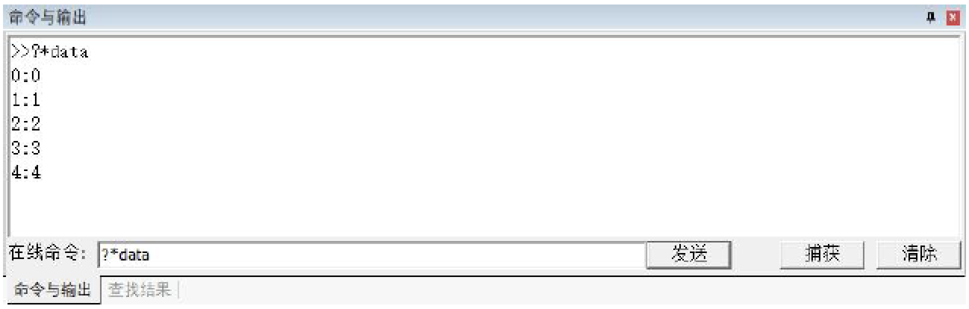
?*BASE: print BASE axes list of present task(valid in firmware version above 140123）



```
>>?*base
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 4
```

Command: ?*base

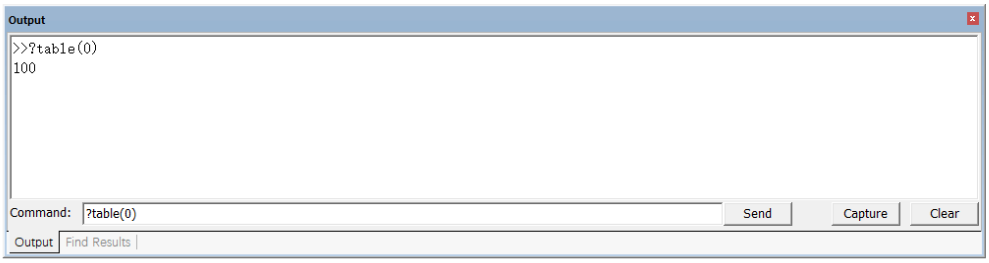
?*array: print all elements of array, if array length is too large, only partial elements can be shown;



```
>>?*data
0:0
1:1
2:2
3:3
4:4
```

在线命令: ?*data

?*array(n):print one array element.

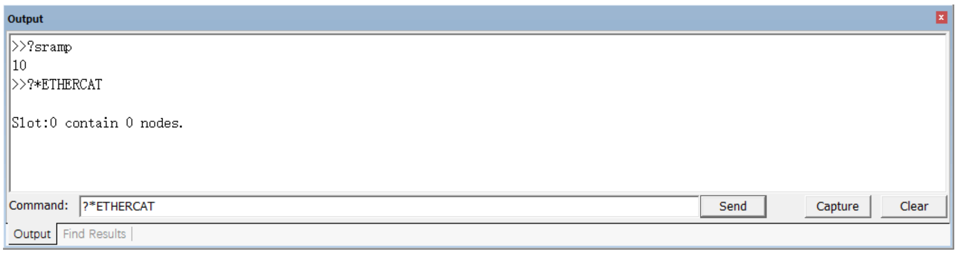


```
>>?table(0)
100
```

Command: ?table(0)

?*parameter name: print this parameter of all axes.

?*ETHERCAT: print EtherCAT connection status.



```
>>?sramp
10
>>?*ETHERCAT

Slot:0 contain 0 nodes.
```

Command: ?*ETHERCAT

Slot 0 contain 1 nodes:devices connected to slot 0

Lostcount 0-0:lost packets

Node:NODE serial number of connected devices;

Status:devive connection status,see ZBasic instruction:NODE_STATUS for reference;

Manid:manufacturer ID;

Productid:device ID;

Axises:max axes of device;

AL Status:outputs status of device;

Node_profile:profile setting of device;

Bindaxis:mapped controller axis number;

Drive_profile:PDO setting

Controlword:control word

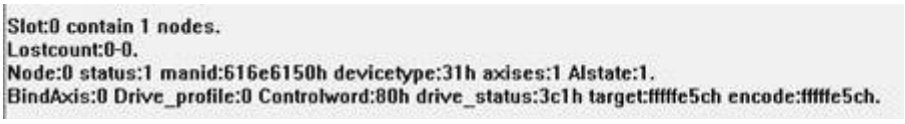
Drive_status:device status,see ZBasic instruction:DRIVE_STATUS for reference;

Drive_mode:device control mode;

Target:motor position;

Encode:encoder position.

?*RTEX: print RTEX connection status.



```
Slot:0 contain 1 nodes.
Lostcount:0-0.
Node:0 status:1 manid:616e6150h devicetype:31h axises:1 Alstate:1.
BindAxis:0 Drive_profile:0 Controlword:80h drive_status:3c1h target:fffffe5ch encode:fffffe5ch.
```

Slot 0 contain 1 nodes:devices connected to slot 0

Lostcount 0-0:lost packets

Node:NODE serial number of connected devices;

Status:devive connection status,see ZBasic instruction:NODE_STATUS for reference;

Manid:manufacturer ID;

Devicetype:device ID;

Axises:max axes of device;

AL Status:outputs status of device;

Bindaxis:mapped controller axis number;

Drive_profile:PDO setting

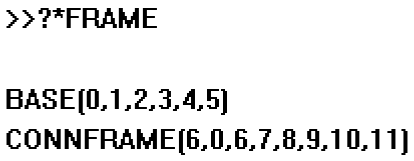
Controlword:control word

Drive_status:device status,see ZBasic instruction:DRIVE_STATUS for reference;

Target:motor position;

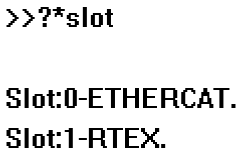
Encode:encoder position.

?*FRAME:print robot arm parameters,valid in firmware 161022 or above



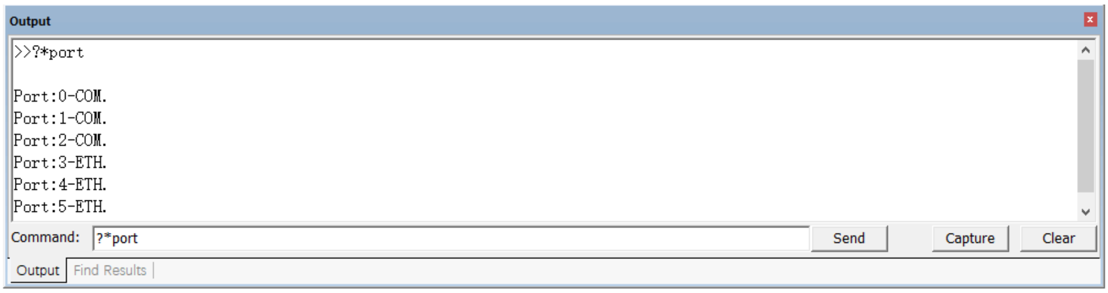
```
>>?*FRAME

BASE(0,1,2,3,4,5)
CONNFRAME(6,0,6,7,8,9,10,11)
```

?*SLOT:print slot information.



```
>>?*slot

Slot:0-ETHERCAT.
Slot:1-RTEX.
```

?*PORT: print all ports information.



```
>>?*port

Port:0-COM.
Port:1-COM.
Port:2-COM.
Port:3-ETH.
Port:4-ETH.
Port:5-ETH.
```

Command: ?*port

COM: serial port;
ETH: Ethernet port;
ECUSTOM: self-defined net port;
CONNECT: net port of controller link.

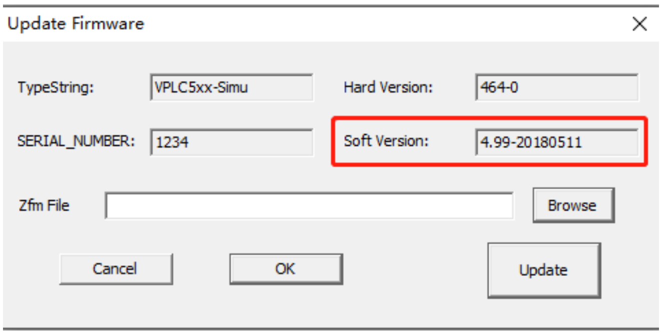
2.5 Firmware Upgrade

The firmware upgrade is used to update the existing firmware version of the controller. If the current firmware version cannot meet the program running requirements, or some command functions are not supported, the firmware needs to be upgraded.

There are 2 ways to upgrade controller firmware, by independent tool: zfirmdown, or by ZDevelop directly.

By ZDevelop:

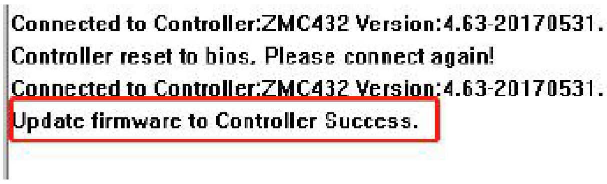
Prepare firmware file first, click menu: Controller > Firmware controller, to enter firmware upgrade window.



TypeString: VPLC5xx-Simu
Hard Version: 464-0
SERIAL_NUMBER: 1234
Soft Version: 4.99-20180511
Zfm File

Click “Browse” to select firmware file, click “Update”, will pop window: Enter ZBIOS?, click “Confirm”, then will pop another new window: Connect to controller? Link to controller again. after link with controller is built again, it will enter firmware upgrade window again, which is in ZBIOS status, typestring item is shown as “model-ZBIOS”.

click “update”, process bar will come, firmware upgrade finished after it reach 100%, output box will print message: Update firmware to controller Success.

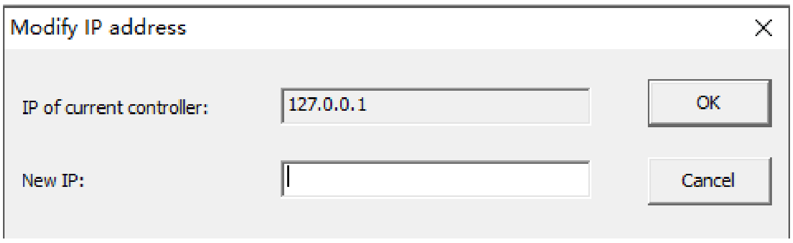


```
Connected to Controller:ZMC432 Version:4.63-20170531.
Controller reset to bios. Please connect again!
Connected to Controller:ZMC432 Version:4.63-20170531.
Update firmware to Controller Success.
```

Then, reconnect to controller and check the firmware version number of controller state.

2.6 Modify IP Address

Menu bar “Controller” – “Modify IP Address”, below window will appear, now, it shows current controller IP, then input new IP address directly.



Modify IP address

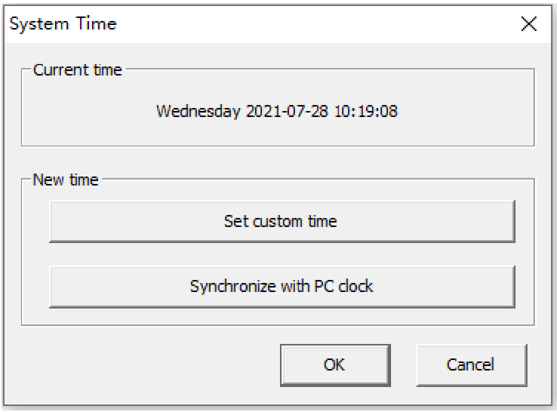
IP of current controller: 127.0.0.1

New IP:

After modifying, controller and ZDevelop will disconnect, which means they need to make connection again.

2.7 System Time Setting

Controller time can be set by manual or synchronize with PC system.



System Time

Current time

Wednesday 2021-07-28 10:19:08

New time

Chapter III ZDevelop Project

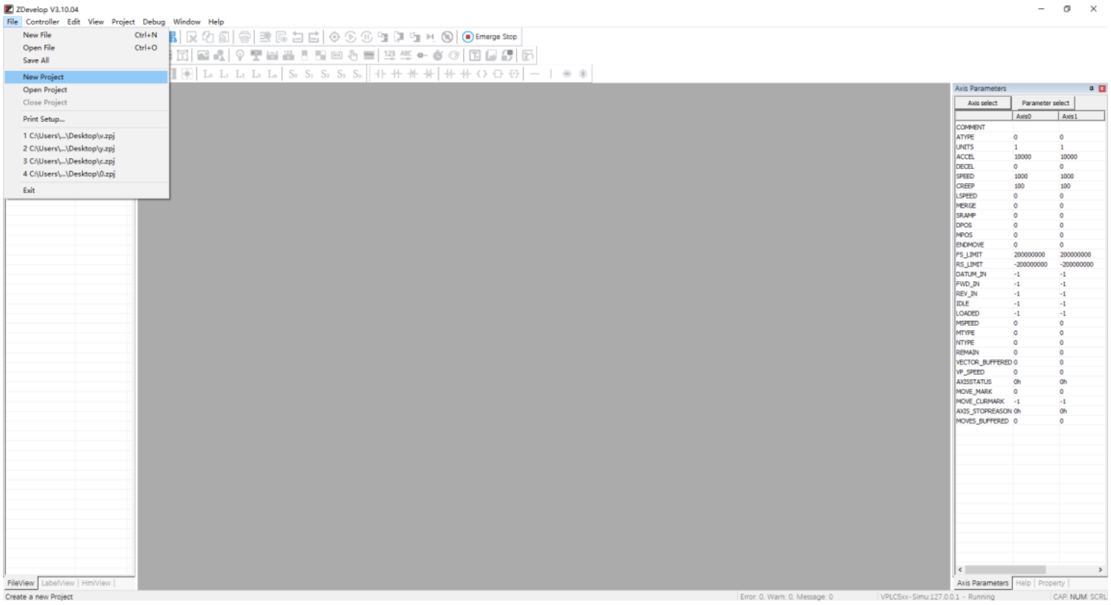
Project is necessary file which is operated in Zmotion controller, we need to create it in Develop, then load it into motion controller. One project contains all logic needed for application operation.

3.1 Project File

File name of project file with suffix ".zpj", file in program must be in the same folder as project file (namely, bas/plc/hmi file included by zpj file are saved in same folder).

Click "File", and select "open project", added project file (bas/plc/hmi file in this zpj) will open automatically, or drag zpj file to ZDevelop so that open directly.

If only open bas/plc/hmi file, but relative projects are not opened, then program can't be downloaded, which means it can't run.



	Axis0	Axis1
COMMENT		
ATYPE	0	0
UNITS	1	1
ACCEL	10000	10000
DECEL	0	0
SPEED	1000	1000
CREEP	100	100
LSPEED	0	0
MERGE	0	0
SRAMP	0	0
DPOS	0	0
MPOS	0	0
ENDMOVE	0	0
FS_LIMIT	200000000	200000000
RS_LIMIT	-200000000	-200000000
DATUM_IN	-1	-1
FWD_IN	-1	-1
REV_IN	-1	-1
IDLE	-1	-1
LOADED	-1	-1
MSPEED	0	0
MTYPE	0	0
NTYPE	0	0
REMAIN	0	0
VECTOR_BUFFERED	0	0
VP_SPEED	0	0
AXISSTATUS	0h	0h
MOVE_MARK	0	0
MOVE_CURMARK	-1	-1
AXIS_STOPREASON	0h	0h
MOVES_BUFFERED	0	0

Create a new Project

Error: 0, Warn: 0, Message: 0

VPLC5xx-Simu:127.0.0.1 - Running

3.2 Project Configuration

Reserved functions

3.3 Program File

Zbasic file: programmed in Zbasic, which is the most important language for Zmotion products, including almost all main product function concepts, as most recommended language when try to develop application through Zmotion controllers, file suffix is ".bas".

ZPlc file: programed in ZPlc, a ladder diagram language developed by Zmotion, file suffix is ".plc".

ZHmi file: programed in ZHmi, which is developed for Zmotion HMI configuration, file suffix is ".hmi"

3.4 ZAR File

ZAR file is a kind of encryption file, the suffix is .zar. After project item generated ZAR file, code can't be found. Now, you can download ZAR file into controller, the methods of downloading ZAR file refer to chapter IV.

3.5 Library File

Library file is generated from "compile as Lib" and is saved for program protection, program won't be modified. Library file's name with suffix ".zlb", the methods of compiling Lib file refer to Chapter IV.

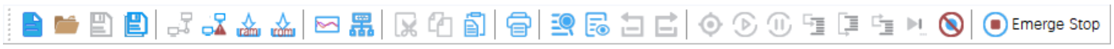
Chapter IV Program Edit

ZDevelop integrates program edit and program debug, which means it is easy to use.

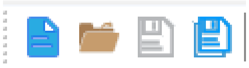
4.1 Quick Edit

ZDevelop provides a very convenient program editting environment, many shortcut buttons can be used.

1. Tool bar:



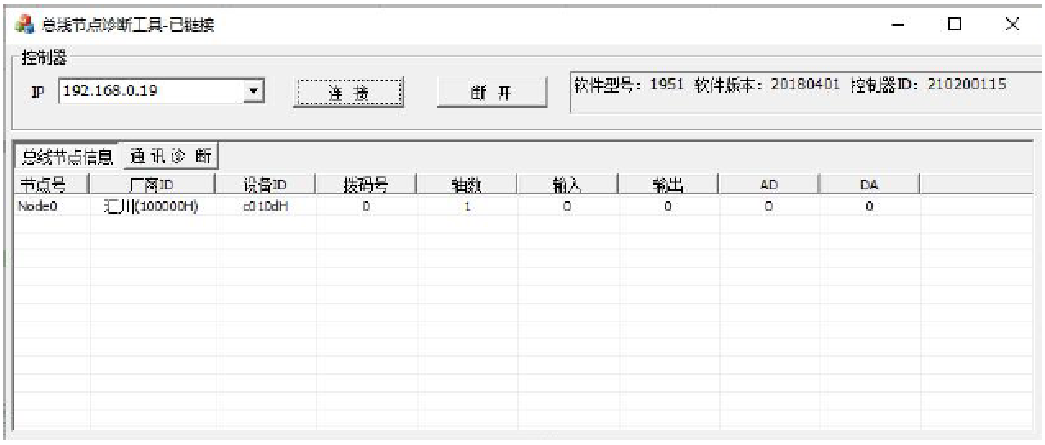
When the mouse is placed on the button, the name of each button is automatically prompted.



New: new build a file, basic/plc/hmi can be selected. (Ctrl + N)
Open: open file, basic/plc/hmi can be selected. (Ctrl + O)
Save: save the modification of current file. (Ctrl + S)
Save All: save modification of all files.

Connect: connect to PC. (Ctrl + Alt + C)
Disconnect: disconnect to PC. (Ctrl + Alt + D)
Down Ram: download the current program into controller, don’t support power failure storage.
Down Rom: download the current program into controller, support power failure storage.

Toggle Scope Window: open oscilloscope
EtherError_Tool: entrance of bus node diagnose tool, open below window, current node information and communication diagnose can be checked.



总线节点诊断工具-已链接

控制器

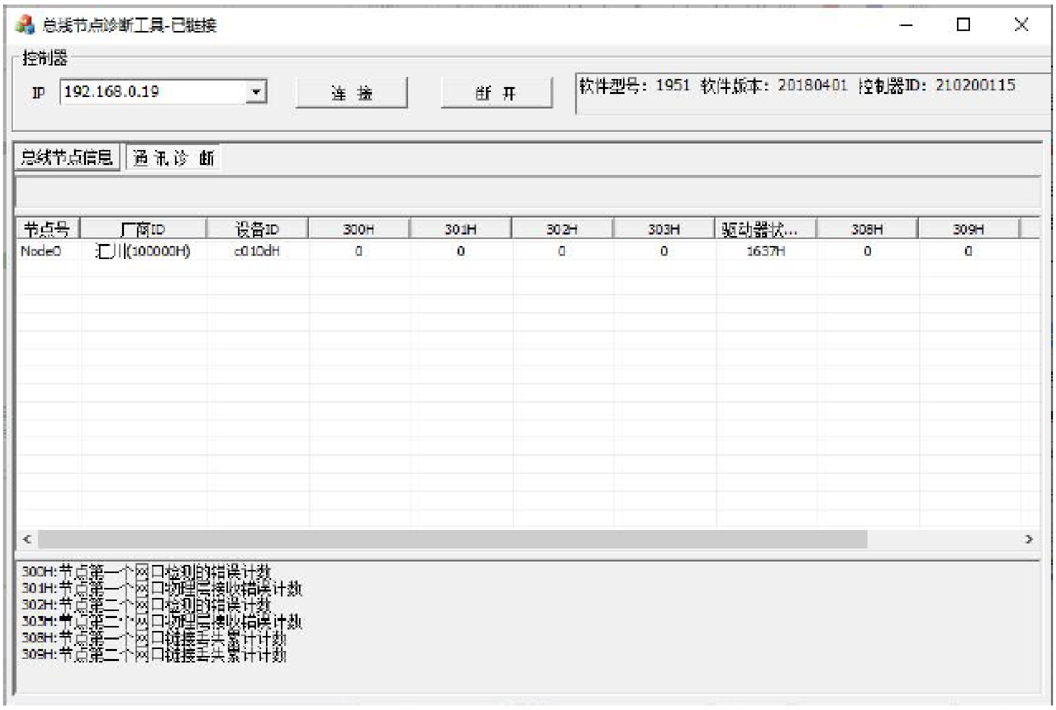
IP 192.168.0.19

连 接　断 开

软件型号：1951 软件版本：20180401 控制器ID：210200115

总线节点信息　通 讯 诊 断

节点号	厂商ID	设备ID	拨码号	轴数	输入	输出	AD	DA
Node0	汇川(100000H)	c010dH	0	1	0	0	0	0


总线节点诊断工具-已链接

控制器

IP 192.168.0.19 连 接 断 开

软件型号：1951 软件版本：20180401 控制器ID：210200115

总线节点信息 通 讯 诊 断

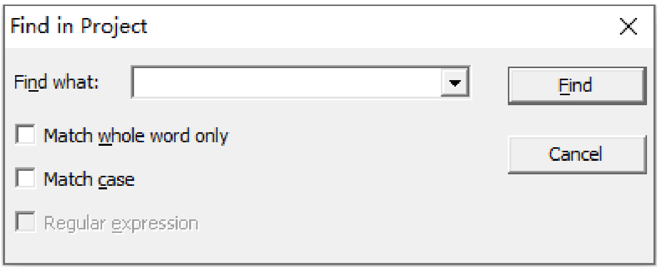
节点号	厂商ID	设备ID	300H	301H	302H	303H	驱动器状...	308H	309H
Node0	汇川(100000H)	c010dH	0	0	0	0	1637H	0	0

300H:节点第一个网口检测的错误计数
301H:节点第一个网口物理层接收错误计数
302H:节点第二个网口检测的错误计数
303H:节点第二个网口物理层接收错误计数
308H:节点第一个网口链接丢失累计计数
309H:节点第二个网口链接丢失累计计数

Cut (Ctrl + X) / Copy (Ctrl + C) / Paste (Ctrl + V) / Print (Ctrl + P) program

"Print" is only valid in Basic program and PLC statement list program, it doesn't support HMI and PLC.

Search (Shift + F4): search in whole project, and the result will show in "Find" window.


Find in Project

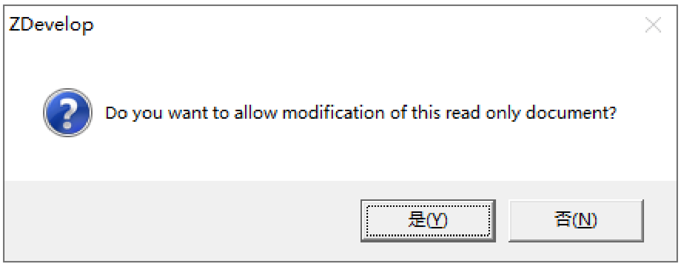
Find what:

☐ Match whole word only

☐ Match case

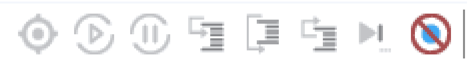
☐ Regular expression

RO (read only): under this status, program can't be modified. After starting RO function, the mark will become bule color. If you still input, below hint will appear. (HMI doesn't have RO function)


ZDevelop

Do you want to allow modification of this read only document?

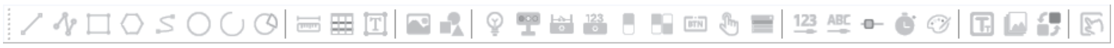
Go back: it will go to former modification place.
Go forward: it will go to next modification place.



Debug tool: it is valid in debug state, please see Program Debug.

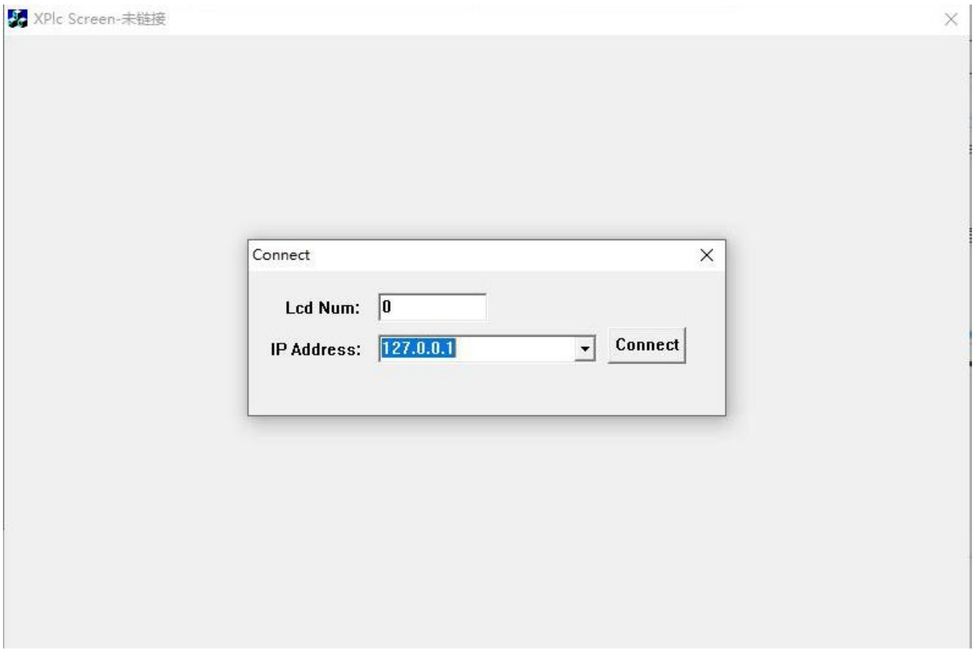
Emerge stop: emerge stop program and motions of all axes.

2. Control tool bar:



The HMI component shortcut button is only valid under HMI programming. The last hand button is the button to open the xplcscreen touch screen simulation tool.

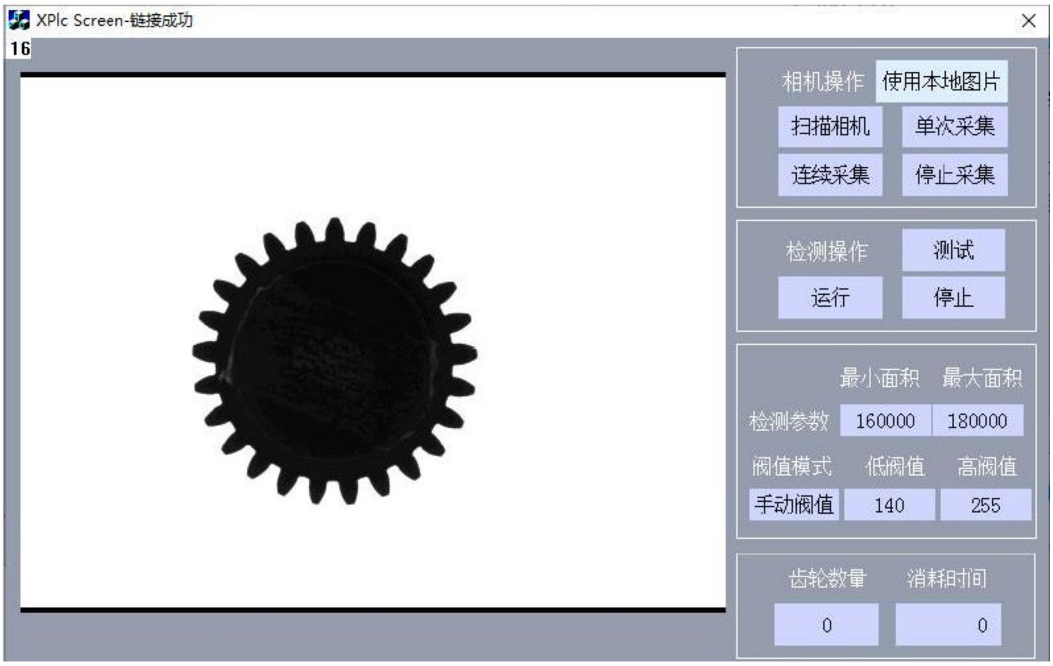
First download the program to the controller or simulator, and then click the hand button, the following window will pop up, select the IP of the controller in the small window and connect it, and the HMI configuration interface that has been downloaded to the controller/simulator will display.



Lcd Num: 0

IP Address: 127.0.0.1

After connecting successfully:

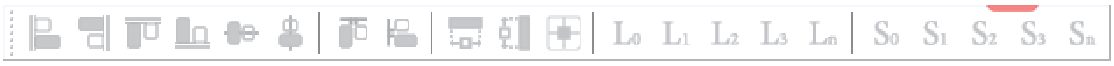


	最小面积	最大面积
检测参数	160000	180000

阈值模式	低阈值	高阈值
手动阈值	140	255

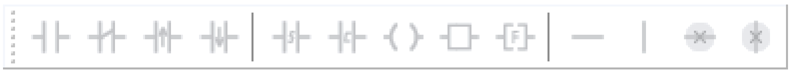
齿轮数量	消耗时间
0	0

3. **HMI tool bar:**



The left part is HMI component alignment button, the right part is HMI language switch and status switch button.

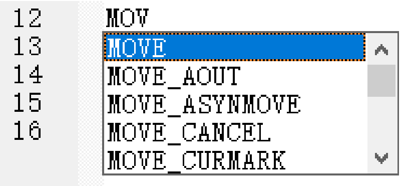
4. **Ladder of Diagram tool:**



PLC ladder diagram shortcut input button.

4.2 ZBasic

There are quick tips when type instruction:

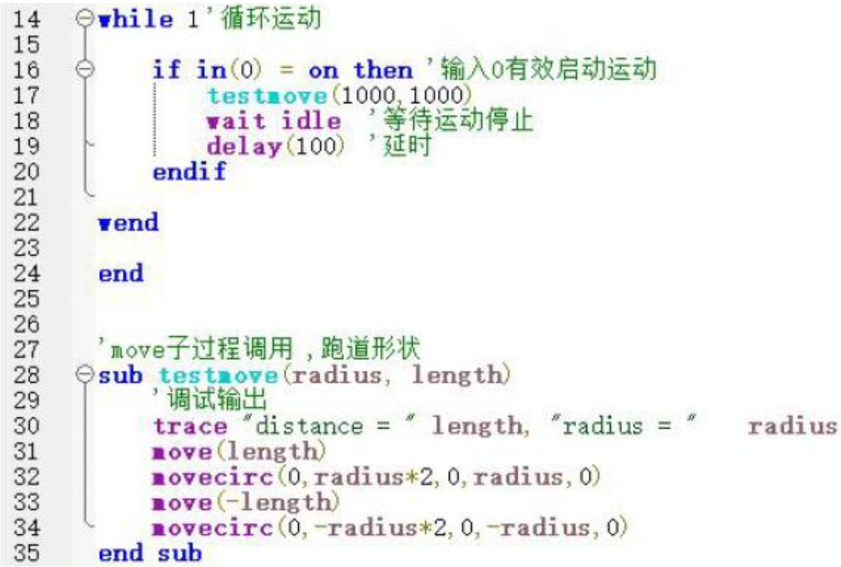


```
MOV
MOVE
MOVE_AOUT
MOVE_ASYNMOVE
MOVE_CANCEL
MOVE_CURMARK
```

It provides grammar highlight function.

Instructions, variables parameters, comment and others show in different colors.

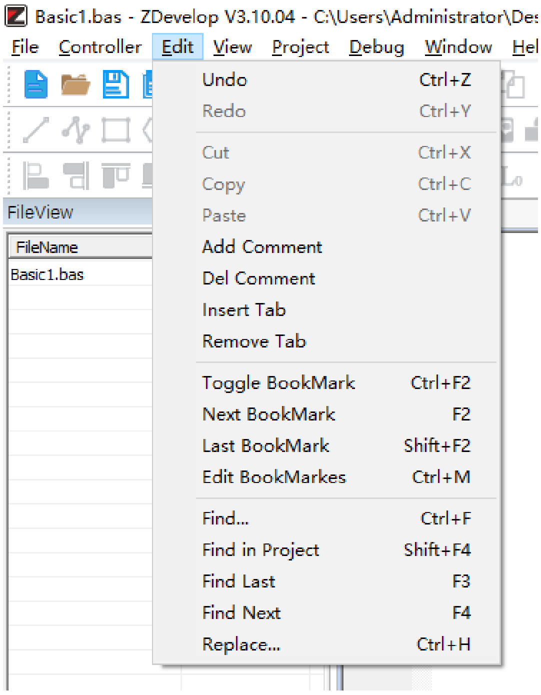
The line below the IF command is a snap line, which can be canceled by using the menu bar "View" - "Indent Line". The "Auto Linefeed" function can make the program display automatically wrap according to the size of the current program editing window.



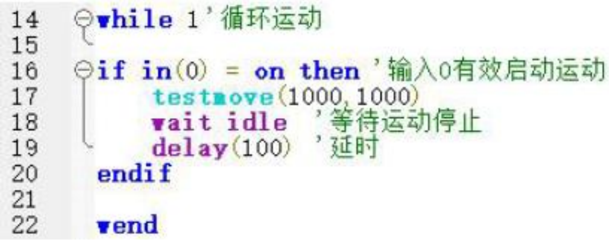
```
while 1 '循环运动

    if in(0) = on then '输入0有效启动运动
        testmove(1000,1000)
        wait idle  '等待运动停止
        delay(100)  '延时
    endif

wend

end


'move子过程调用 ，跑道形状
sub testmove(radius, length)
    '调试输出
    trace "distance = " length, "radius = "    radius
    move(length)
    movecirc(0,radius*2,0,radius,0)
    move(-length)
    movecirc(0,-radius*2,0,-radius,0)
end sub
```

"Add Comment" makes program block into comment part quickly, "Del Comment" makes comment into program.

Operation: select one certain program in Basic, click "Edit" – "Add Comment", then selected program will become comment. Also, select one certain comment, click "Edit" – "Del comment", then selected comment will become program.

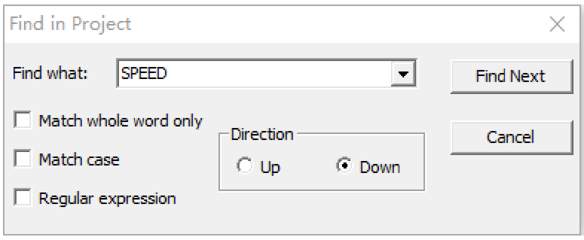


Menu "Edit" – "Insert Tab" / "Del Tab" function is used to control the program indent. Below is the "Del Tab" operation result for whole IF cycle command, operate once, and indent is fixed.

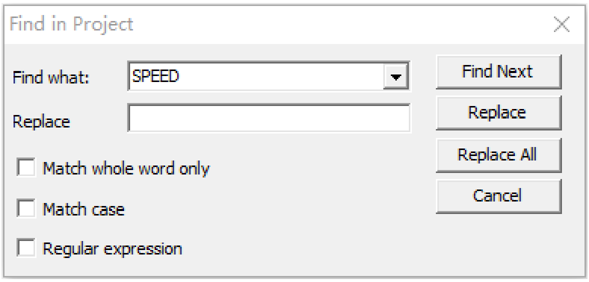


```
while 1 '循环运动

if in(0) = on then '输入0有效启动运动
    testmove(1000,1000)
    wait idle  '等待运动停止
    delay(100)  '延时
endif

wend
```

Menu “Edit” – “Find”, find in project.



Menu “Edit” – “Replace”, find and replace in project.



4.3 ZPLC

In ZPLC, there are 2 programming modes: ladder diagram and instruction list, these two modes can switch to each other freely.

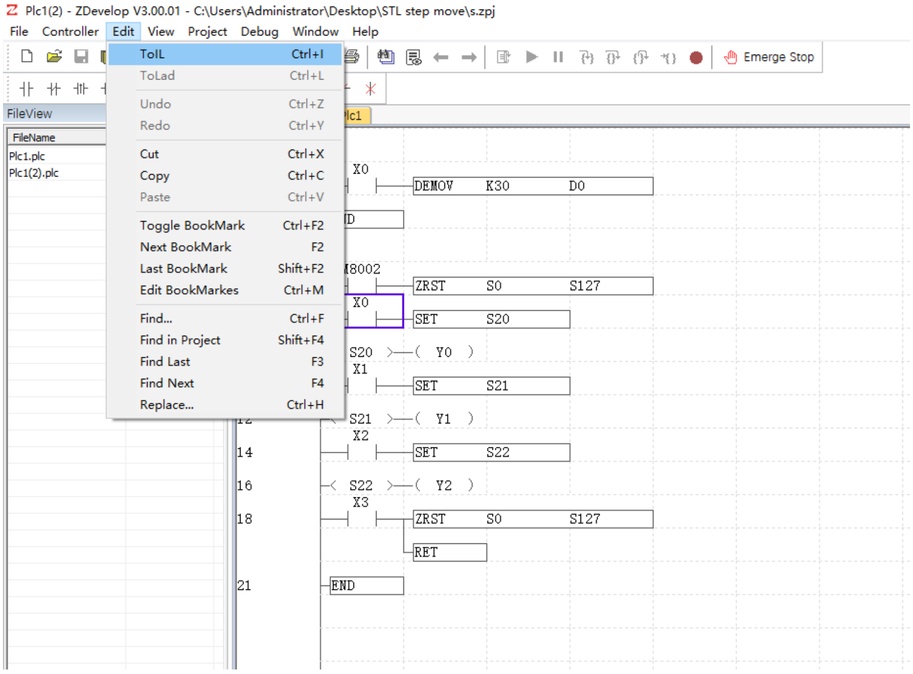
Note: no distinction between capital and lower letters when programming in instruction list mode.

Menu “Edit” – “TolL” / “ToLad”

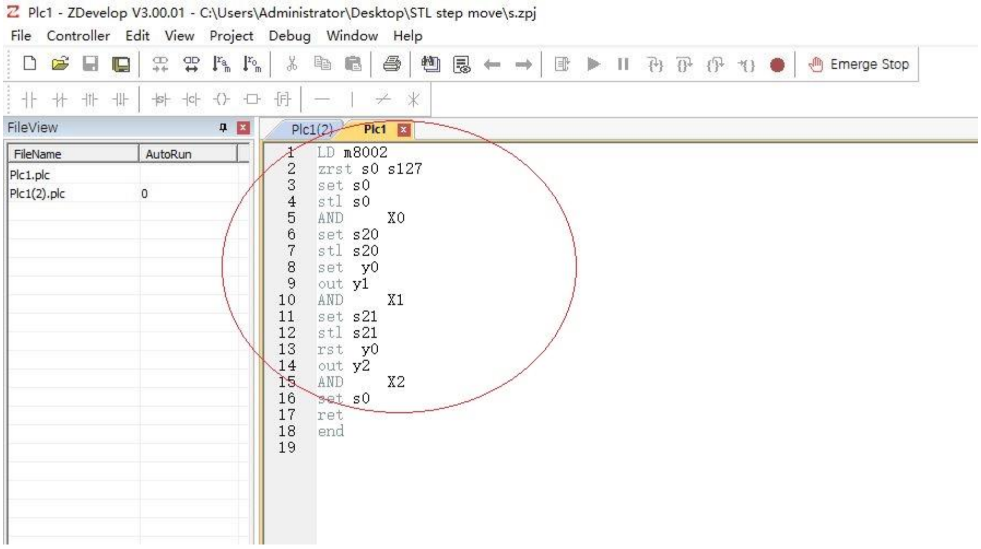
TolL: convert to instruction list

ToLad: convert to ladder diagram program

- **Ladder diagram:** it is convenient and east to observe, it needs programmers are familiar with PLC instructions



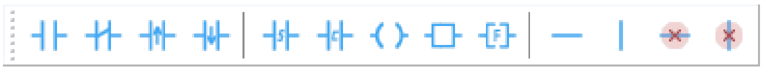
- **Instruction list:**

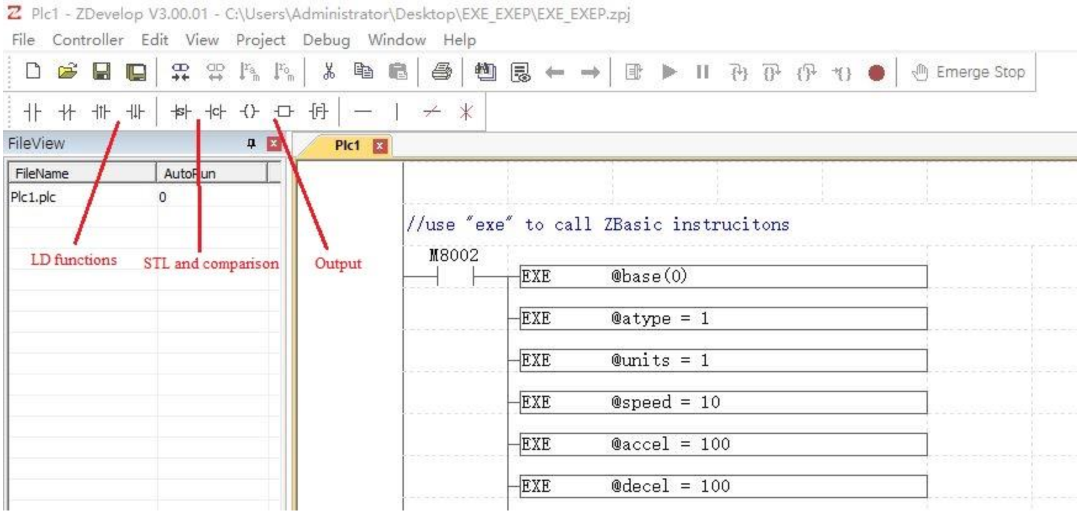

Plc1 - ZDevelop V3.00.01 - C:\Users\Administrator\Desktop\STL step move\s.zpj
File Controller Edit View Project Debug Window Help
Emerge Stop
FileView
FileName AutoRun
Plc1.plc
Plc1(2).plc 0
Plc1(2) Plc1

```
LD m8002
zrst s0 s127
set s0
stl s0
AND     X0
set s20
stl s20
set  y0
out y1
AND     X1
set s21
stl s21
rst  y0
out y2
AND     X2
set s0
ret
end
```

Program Edit:

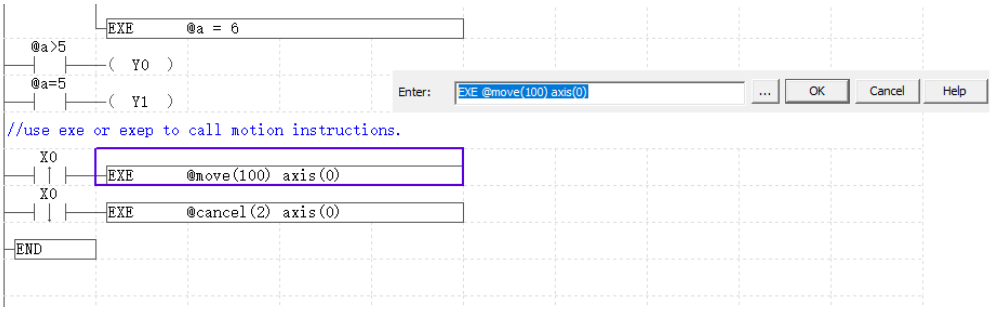
Click shortcut labels to insert relevant functions:




Plc1 - ZDevelop V3.00.01 - C:\Users\Administrator\Desktop\EXE_EXEP\EXE_EXEP.zpj
File Controller Edit View Project Debug Window Help
Emerge Stop
FileView
FileName AutoRun
Plc1.plc 0
Plc1
LD functions
STL and comparison
Output
M8002

```
//use "exe" to call ZBasic instrucitons
M8002
EXE     @base(0)
EXE     @atype = 1
EXE     @units = 1
EXE     @speed = 10
EXE     @accel = 100
EXE     @decel = 100
```

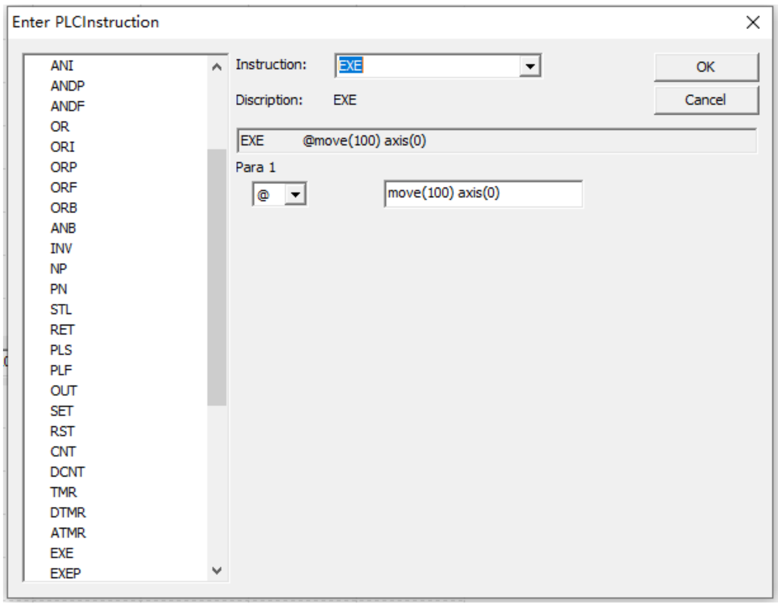
Double-click in the corresponding cell or directly enter the command, you can also achieve shortcut input. You can enter commands or modify parameters. Click "OK" to save the modification results, click "Cancel" to close the shortcut input box, and click "Help" to open the help document.



```
EXE     @a = 6
@a>5
( Y0 )
@a=5
( Y1 )
//use exe or exep to call motion instructions.
X0
EXE     @move(100) axis(0)
X0
EXE     @cancel(2) axis(0)
END
```

Enter: EXE @move(100) axis(0)

Click "…" label in quick insert box to open PLC instructions input box, which enables instruction quick selection and parameter setting.



Enter PLCInstruction

Instruction: EXE

Discription: EXE

EXE @move(100) axis(0)

Para 1

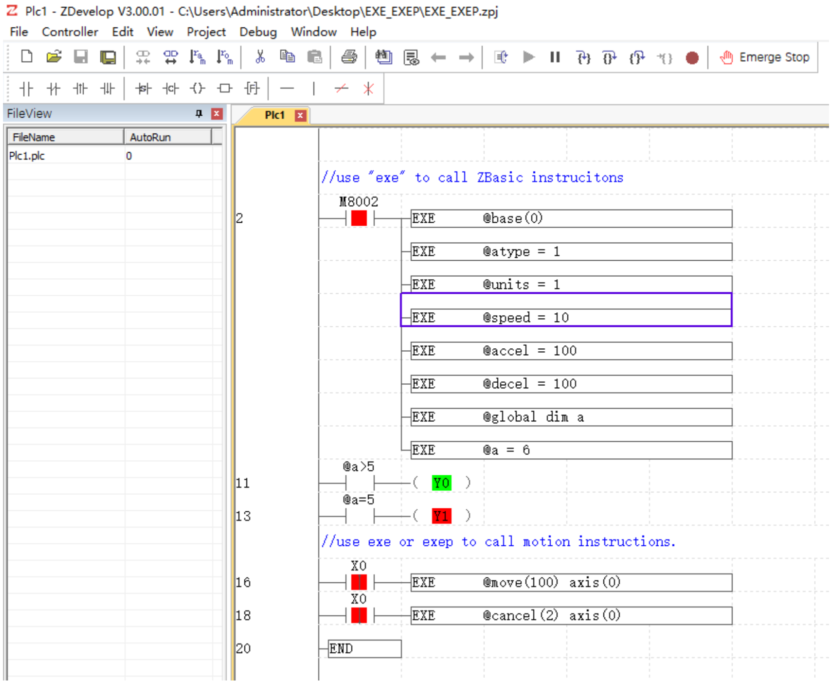
@ move(100) axis(0)

Remember to add END in the end, or it can't run.

Use "EXE @" command to call Basic in PLC, PLC and Basic have some shared variables, please see PLC program manuals for details.

Debug

After entering debug mode, some signs will appear to indicate status: green-on, red-break, register present value also will show above.



Plc1 - ZDevelop V3.00.01 - C:\Users\Administrator\Desktop\EXE_EXEP\EXE_EXEP.zpj

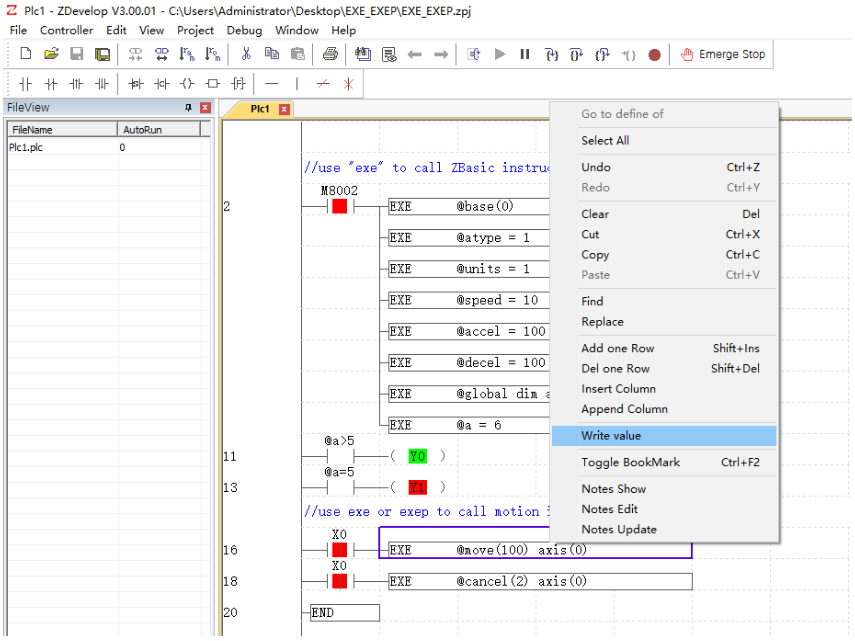
FileName	AutoRun
Plc1.plc	0

```
   //use "exe" to call ZBasic instrucitons
      M8002
2  ─┤🟥├──EXE   @base(0)
         EXE   @atype = 1
         EXE   @units = 1
         EXE   @speed = 10
         EXE   @accel = 100
         EXE   @decel = 100
         EXE   @global dim a
         EXE   @a = 6
      @a>5
11 ─┤ ├──( 🟩Y0 )
      @a=5
13 ─┤ ├──( 🟥Y1 )
   //use exe or exep to call motion instructions.
      X0
16 ─┤🟥├──EXE   @move(100) axis(0)
      X0
18 ─┤🟥├──EXE   @cancel(2) axis(0)
20 END
```

Note: interruption debug is not supported in ladder diagram mode, but it is supported under instruction list mode.

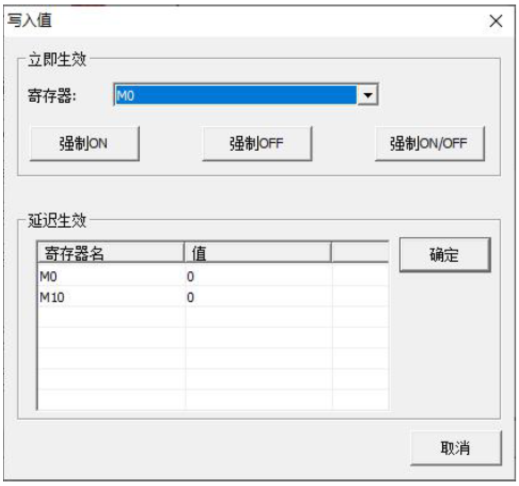
Write Value

In debug mode, select target cell and right click to enter menu: write data in this page, we can edit M(bit value) or D(word value).



```
   //use "exe" to call ZBasic instru
   M8002
2  EXE   @base(0)
   EXE   @atype = 1
   EXE   @units = 1
   EXE   @speed = 10
   EXE   @accel = 100
   EXE   @decel = 100
   EXE   @global dim a
   EXE   @a = 6
   @a>5
11 ( Y0 )
   @a=5
13 ( Y1 )
   //use exe or exep to call motion i
   X0
16 EXE   @move(100) axis(0)
   X0
18 EXE   @cancel(2) axis(0)
20 END
```

Write value window is below: it only shows the current selected register.



寄存器: M0

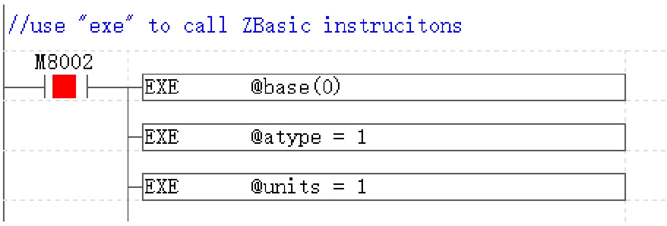
寄存器名	值
M0	0
M10	0

Add comments

There are three ways to add comments, as follow:

1. by instruction "//"

This is usually used to explain function block in the front of the code, and it consumes one line.

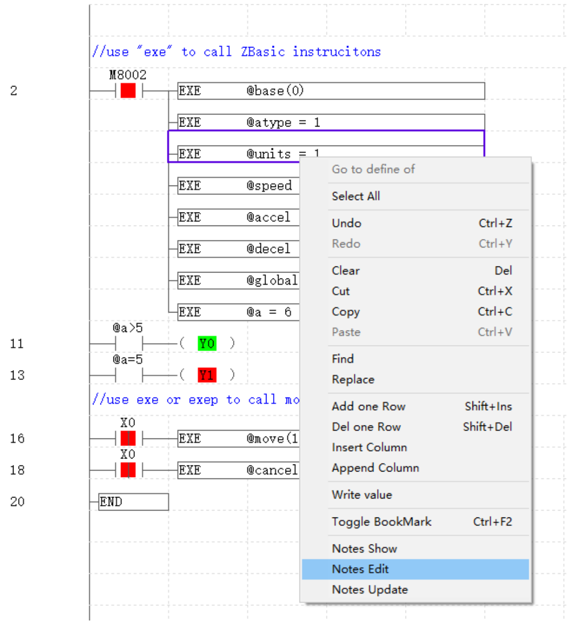


```
//use "exe" to call ZBasic instrucitons
M8002
EXE   @base(0)
EXE   @atype = 1
EXE   @units = 1
```

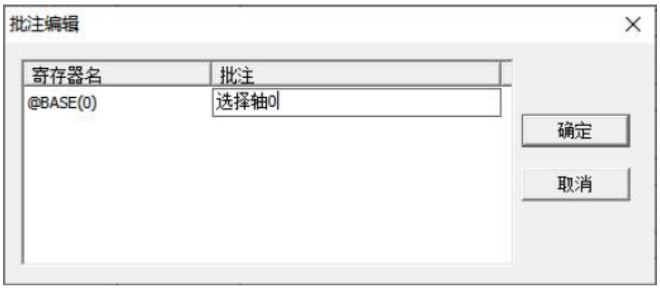
2. By right click menu function: notes edit

This function is used to explain soft elements under ladder diagram program.

Note edit method: select target soft element, right click to enter "note edit" page.



When there is comment editing window, double click "left button" below the comment, then input comment, click OK.



寄存器名	批注
@BASE(0)	选择轴0

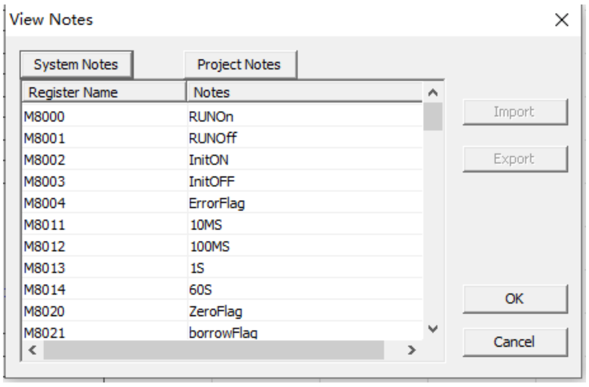
Comments can be hided or displayed by disable or enable right click menu item: notes show.

3. By view note window

Click menu: View> notes to enter view note window, it is usually used to add comments of PLC elements.

"system notes" is used for special relay M and special register D, quick add comments to all elements.

"project notes" is used for input register definition and comments add, project notes can be imported or exported.

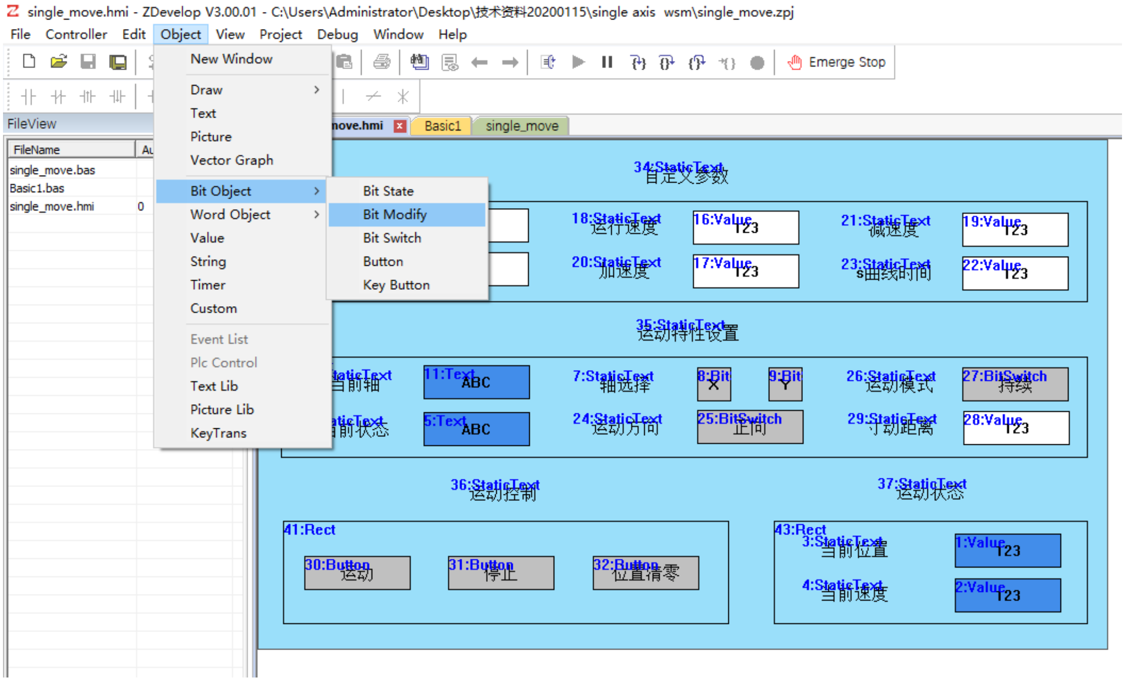


Register Name	Notes
M8000	RUNOn
M8001	RUNOff
M8002	InitON
M8003	InitOFF
M8004	ErrorFlag
M8011	10MS
M8012	100MS
M8013	1S
M8014	60S
M8020	ZeroFlag
M8021	borrowFlag

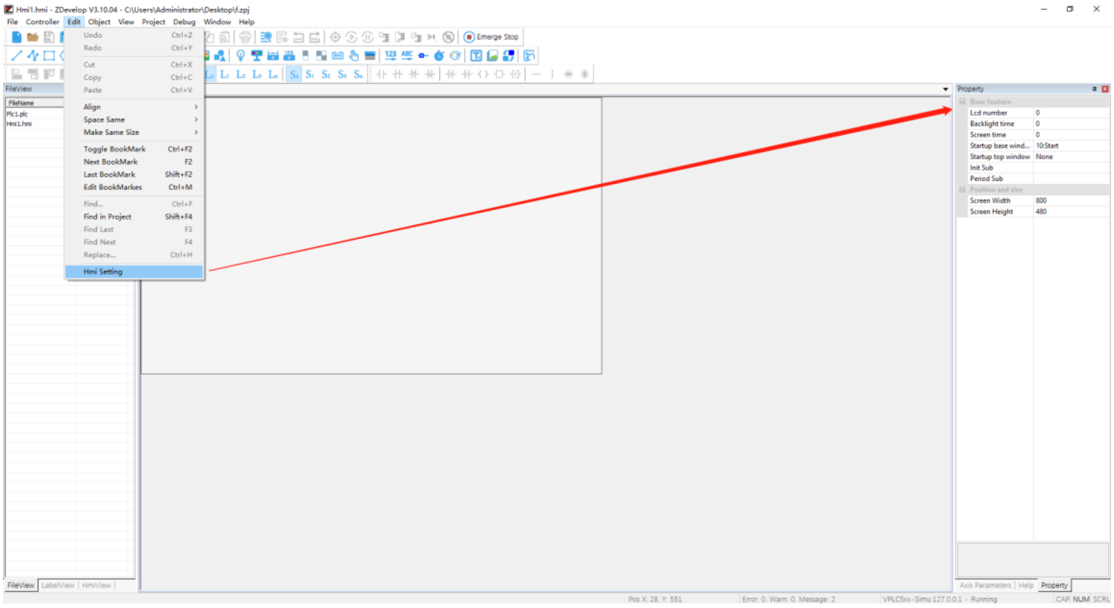
4.4 ZHMI

HMI programming is used to customize the display on the display screen, and touch to perform related operations. Open the configuration "component" to quickly select the input object and improve the programming efficiency.

Common functions of elements: call sub functions, open / close configuration window, bind register, show text / graphic, etc. Different functions have differences, please see “ZHMI Program Manual” for details.



Menu “Edit” – “Hmi Setting”, it must set if run HMI file, Hmi system configuration is in right window, please see it.



Property

Base feature

Lcd number	0
Backlight time	0
Screen time	0
Startup base wind...	10:Start
Startup top window	None
Init Sub	
Period Sub	

Position and size

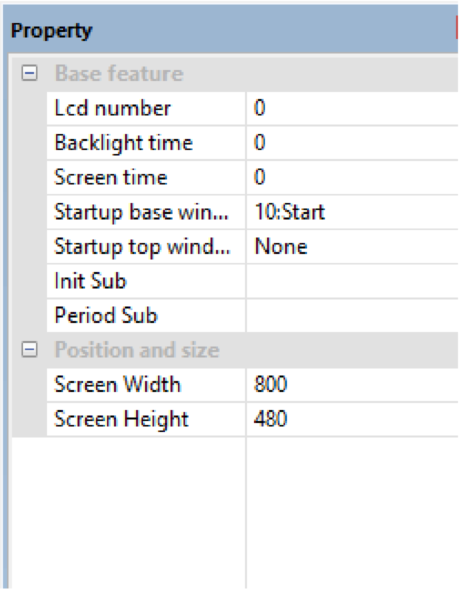
Screen Width	800
Screen Height	480

Resolution is set according to display screen size, normally, it is 800*480 for 7-inch screen.

Initialization function is only called once after power on, it is defined in Basic, and function must be global function (GLOBAL).

The initialization function or periodic function is not required to be set. The way the HMI calls the Basic function also includes the drawing function or refresh function of the custom component, and the action “call sub” of the component.

In the configuration view, the right-click menu "Property" can quickly edit the window display. The pop-up property window is related to the position of the mouse. Place the mouse on the component and right-click the property window as shown in the left figure. Place the mouse on the blank space and right-click. The Property window is shown on the right.



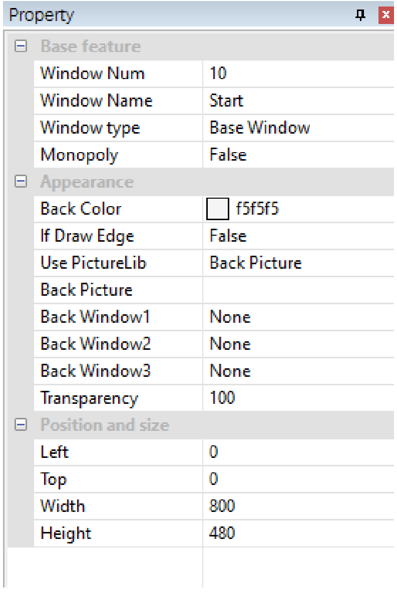
Property

Base feature

Lcd number	0
Backlight time	0
Screen time	0
Startup base win...	10:Start
Startup top wind...	None
Init Sub	
Period Sub	

Position and size

Screen Width	800
Screen Height	480



Property

Base feature

Window Num	10
Window Name	Start
Window type	Base Window
Monopoly	False

Appearance

Back Color	□ f5f5f5
If Draw Edge	False
Use PictureLib	Back Picture
Back Picture	
Back Window1	None
Back Window2	None
Back Window3	None
Transparency	100

Position and size

Left	0
Top	0
Width	800
Height	480

Component right click “Property” background right click “Property”

Property description of Bit status switch (BitSwitch):

Property	Function	Description
Object number	/	/
Object name	/	/
Layer	Select component display layer	TopLayer: on the surface, it shows the most external layer, and covers below components. MidLayer: the middle layer BottomLayer: the bottom layer (default)
IfValid	Select component displays or not	Default is Ture, when select False, the component does not show and no functions
Valid Control	Control component display or not through register	Default is False, True means it is controlled by register.
Safe time (ms)	The leaset button time	Unit is ms.
Device NO.	Device number	Default is local
Regtype	Select register type	Select from the list
Regnumber	Select register number	It does not show when register value is 0, it is used when not 0.
Use PictureLib	BackPicture / Back Picture	Select from picture library or backgroud

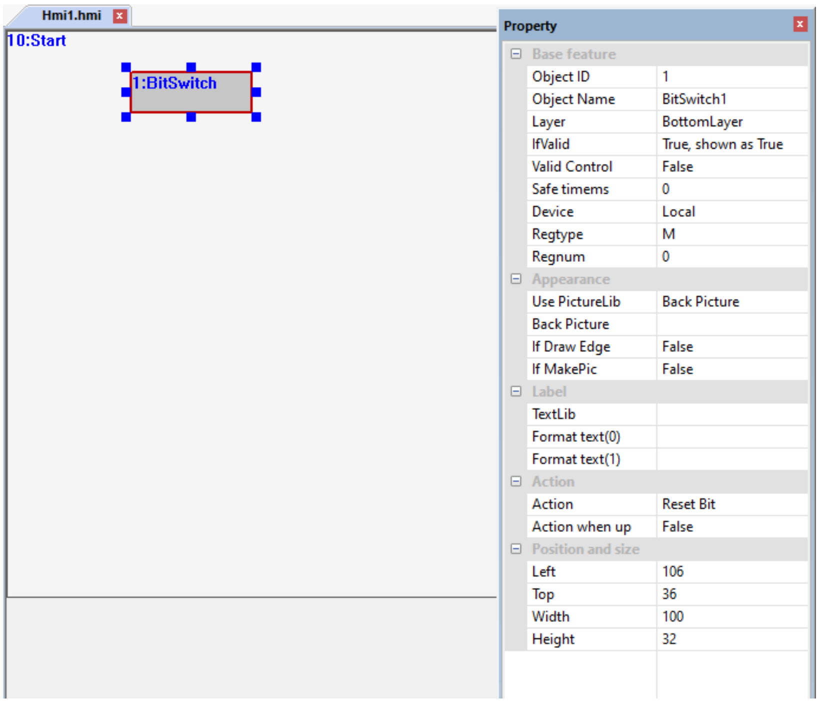
Background picture	Select background picture	After select background picture for picture source, then add
If Draw edge	Select if draw edge	Default is False
If MakePic	Element into picture	Default is False
Text Lib	The name of text library	If empty, indicates the use of a text label
Format text 0 / 1	Open the Format Text Settings window to set the text to be displayed by the component	Default is text 0, button it, it will become text 1
Action	Motion to be executed when button	
Action when up	Select execution action when press or release.	Default False is the action when pressed, True is the action when released
Action operation window	Select window numer be operated	Select from the window list
Follow main window close	Sub window close together with main window	Default is False
Action data	Write assigned value for register after pressing	/
Action function name	SUB function to be called after pressing	Select global SUB function of Basic
Virtual button code	Select virtual button codes	Default is not to select
Top	Vertical starting position	Not to execeed vertical resolution
Left	Horizontal starting position	Not to execeed Horizonical resolution
Width	The width of element	/
Height	The height of element	/

Steps for running new-built HMI file

1. Build HMI file, add autom run task number.
2. Open “HMI setting” window, set screen resolution, start dispalyed window, etc.
3. Edit HMI file, build a new window, add component, and open relative “Property” window and element function.
4. Build new Basic file, edit Basic subfucntion content to be called by HMI component.
5. Add initialization function and period function in “HMI Setting”, and add subfunction name in component motion place of Basic program that is going to be called.
6. Connect to controller or simulator, download the program. If there is no real object, connect to xplc screen for simulation.

HMI Program Example: bind with register

1. Build a new HMI file, and add auto run task number.
2. Add status switch component in start IO window.
3. Open component “Property” window and select register type and number, “Action” select “Reset Bit”, which means assign binded register as 0.
4. Download program into controller or simulator, press the component in main page for checking effects.



Property

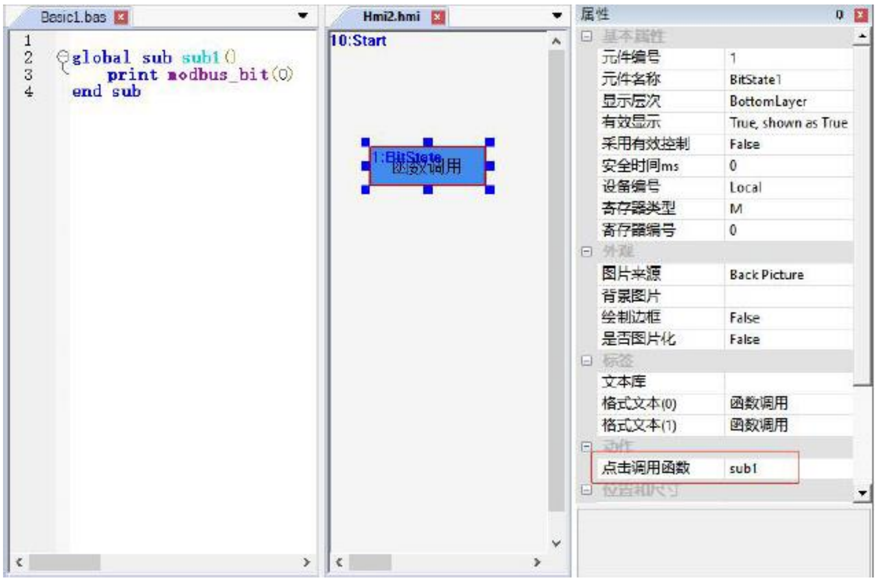
Base feature	
Object ID	1
Object Name	BitSwitch1
Layer	BottomLayer
IfValid	True, shown as True
Valid Control	False
Safe timems	0
Device	Local
Regtype	M
Regnum	0
Appearance	
Use PictureLib	Back Picture
Back Picture	
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	
Format text(1)	
Action	
Action	Reset Bit
Action when up	False
Position and size	
Left	106
Top	36
Width	100
Height	32

Achieved effect:

When press the object, MODBUS_BIT(0) = 0, if select release action, which means first to press it, then release it, and MODBUS(0) still equals to 0.

HMI Program Example: Call SUB function

1. Build new HMI file, and add auto run task number.
2. Add status display object in Bit switch window.
3. New build Basic file, no need to set auto run task number, and edit global SUB subfucntion to be called in Basic.
4. Open object “Property” window, and select editted SUB subfunction name in “click calling function”.
5. Download program into controller or simulator, press the component in main page for checking effects.



```
global sub sub1()
    print modbus_bit(0)
end sub
```

属性

基本属性	
元件编号	1
元件名称	BitState1
显示层次	BottomLayer
有效显示	True, shown as True
采用有效控制	False
安全时间ms	0
设备编号	Local
寄存器类型	M
寄存器编号	0
外观	
图片来源	Back Picture
背景图片	
绘制边框	False
是否图片化	False
标签	
文本库	
格式文本(0)	函数调用
格式文本(1)	函数调用
动作	
点击调用函数	sub1
位置和尺寸	

Achieved effect:

When object is pressed, call SUB subfunction of Basic, call function once when press once.

More HMI routines can be downloaded from ZMOTION webiste.

4.5 Right Click Menu

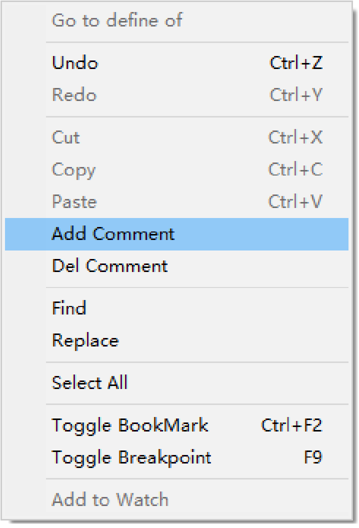
Right click menu functions differ in different program modes: ZBasic, ZPLC and ZHMI. There are some important functions need to keep in mind, which will increase programming efficiency.

Go to define of—select ZBasic keyword, enter right click menu to find source of keyword.

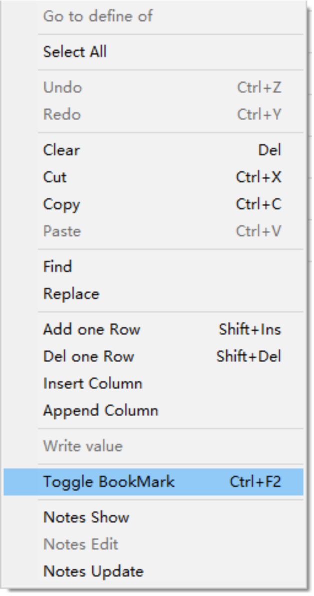
Toggle breakpoint—add breakpoint in program, help to debug;

Add to watch—add variables or values to watch window;

Find/Replace—find or replace letters.



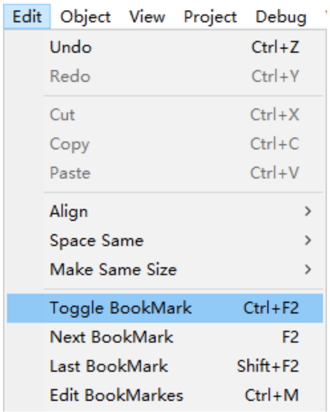
Right Click in ZBasic



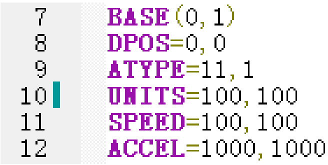
Right Click in ZPLC

4.6 Bookmark

In Edit menu, we can select bookmark functions: Toggle bookmark, next bookmark, last bookmark, edit bookmarks.

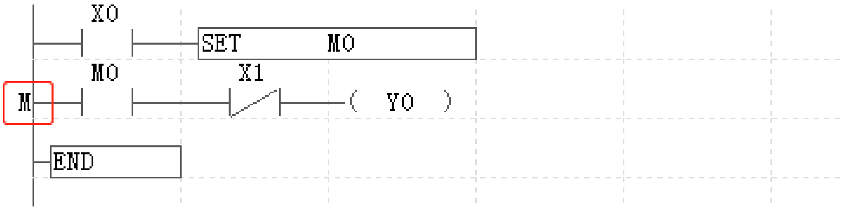


In ZBasic, green label will appear in front of code line if bookmark added:



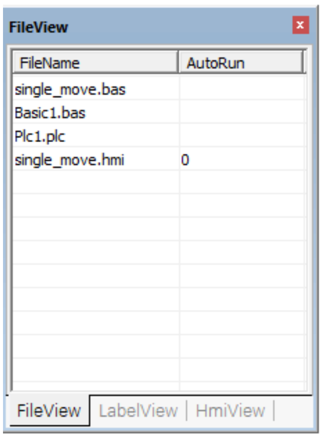
```
BASE(0,1)
DPOS=0,0
ATYPE=11,1
UNITS=100,100
SPEED=100,100
ACCEL=1000,1000
```

In ZPLC, letter “M” will appear in front of code line if bookmark added:



4.7 File View

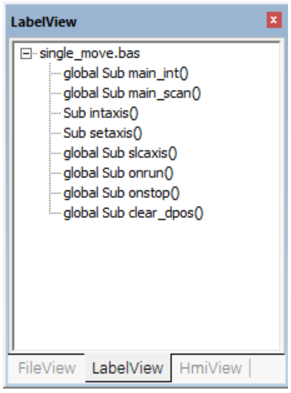
On left side of ZDevelop, it is FileVIew page, file numbers, type, and task number are all shown here. Double click file item to open and edit file, right click file item to open setting window for more configuration.



FileName	AutoRun
single_move.bas	
Basic1.bas	
Plc1.plc	
single_move.hmi	0

4.8 SUB View

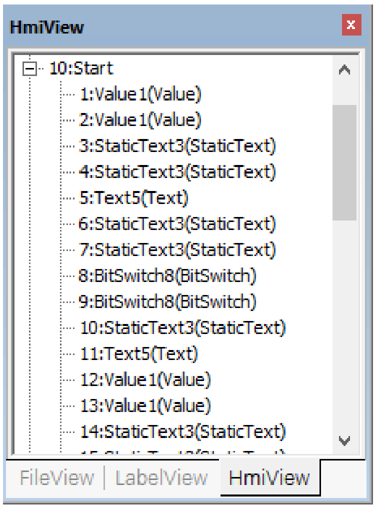
On left side of ZDevelop, click “Sub View” to check all Sub functions in project, double click sub item to jump to sub definition line.



```
single_move.bas
    global Sub main_int()
    global Sub main_scan()
    Sub intaxis()
    Sub setaxis()
    global Sub slcaxis()
    global Sub onrun()
    global Sub onstop()
    global Sub clear_dpos()
```

4.9 HMI View

On left side of ZDevelop, click “HMI View” to check all windows and elements in HMI file. Double click item to open property setting window: Feature



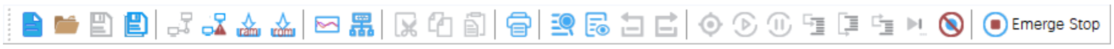
HmiView

```
10:Start
    1:Value1(Value)
    2:Value1(Value)
    3:StaticText3(StaticText)
    4:StaticText3(StaticText)
    5:Text5(Text)
    6:StaticText3(StaticText)
    7:StaticText3(StaticText)
    8:BitSwitch8(BitSwitch)
    9:BitSwitch8(BitSwitch)
    10:StaticText3(StaticText)
    11:Text5(Text)
    12:Value1(Value)
    13:Value1(Value)
    14:StaticText3(StaticText)
```

Chapter V Run Program

5.1 Load Program (RAM/ROM)

In ZDevelop, program must be downloaded into controller, and must build project, then it can download.

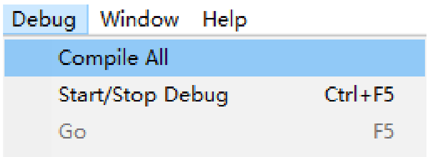


Download RAM: load project into ram, run immediately, it will lose after power off.

Download ROM: load project into rom, run immediately, it supports power failure storage.

⚠ this function is not supported in ZMC00x series.

If project size is too large, click menu: Debug>Compile all, to compile code first, then load project into controller, in this way, project load time will be shortened.

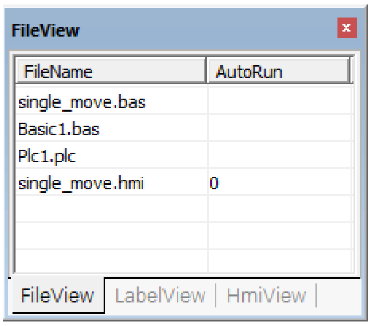


When download, you can open oscilloscope to get motion curve and other changes.

5.2 Auto Run

In a project, at least one auto run number should be set for one file. After load project into controller, project files with auto run number will run controller automatically, if auto run number was not set in any project file, project can not be executed in controller, error information will come: WARN: no program set autorun.

Set project file autorun number in “File View” page. Usually only one autorun number is set, other tasks can be called be instruction: Runtask, Run. Maximum tasks supported in controller is shown in “controller state” page in ZDevelop.

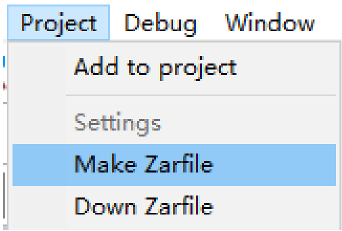


FileName	AutoRun
single_move.bas	
Basic1.bas	
Plc1.plc	
single_move.hmi	0

5.3 Load ZAR

In ZDevelop, developer can convert project into encrypted ZAR file, which can’t be opened and edited again, and Zar file also can be bound with Controller ID (controller unique sequence number, which can be checked through “Controller status” or SERIAL_NUMBER) to let this Zar file only be valid after loaded into unique controller whose ID matches its bound controller.

Through menu bar “Project”, select “Make Zarfile”.

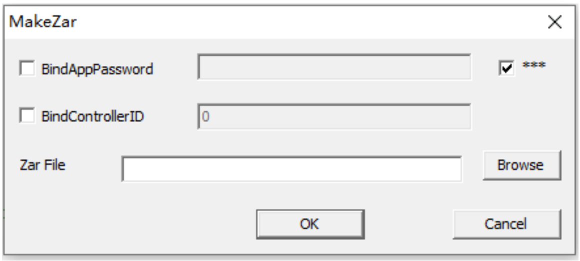


Make Zarfile:

After debugged program, you can download ZAR file after setting password, which needs to use APP_PASS instruction to write password into program for correction.

If select it is bound with controller ID, it will correct controller ID automatically when download ZAR program, it will be downloaded successfully only when their ID are consistent.

End user downloads independent tool software “ZarDown”. Usage steps are same as ZDevelop.



Download Zarfile:

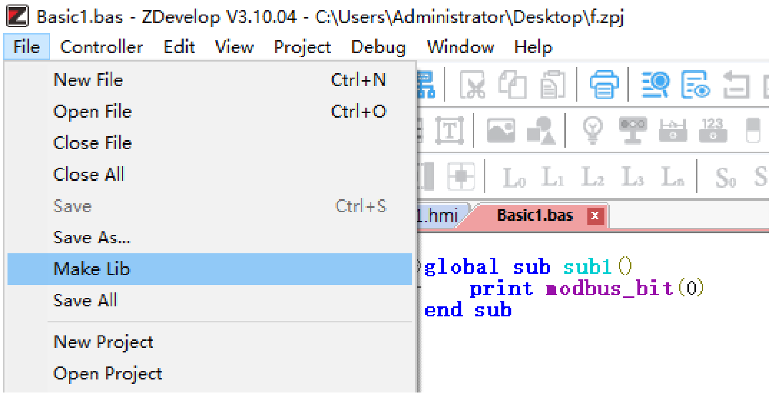
Click “Download Zarfile”, then select from system disk.

Or use FILE command “LOAD_ZAR” function to upload ZAR file in U disk.

5.4 Compile Lib

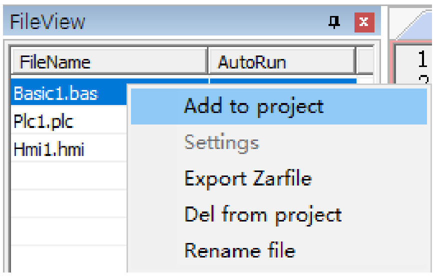
This function belongs to program encryption, same as ZAR Download. It can compile a program file as one library file to save, which is convenient to protect program. File name of library file with suffix “. zlb”, library file only shows global SUB definition.

After debugged, click menu “File” – “Compile Lib”, it will compile all project files as one library file.

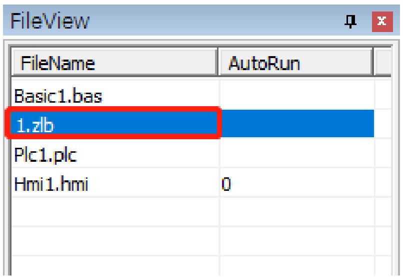


```
global sub sub1()
    print modbus_bit(0)
end sub
```

After compile, add Lib into project again, click right button of mouse in left project window of software, please see the left. Then Lib file is added, see the right picture.

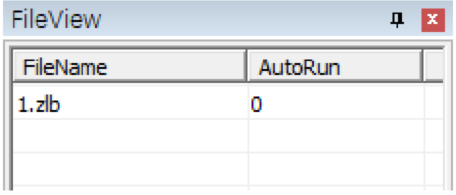


FileName	AutoRun
Basic1.bas	
Plc1.plc	
Hmi1.hmi	

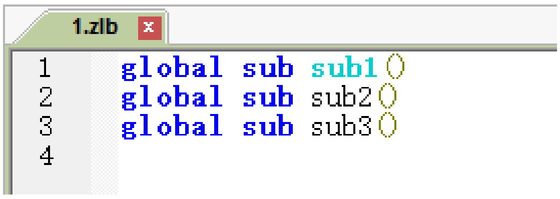


FileName	AutoRun
Basic1.bas	
1.zlb	
Plc1.plc	
Hmi1.hmi	0

Then, delete the original Lib file, set auto run task number is 0, redownload the Lib file into controller.



FileName	AutoRun
1.zlb	0



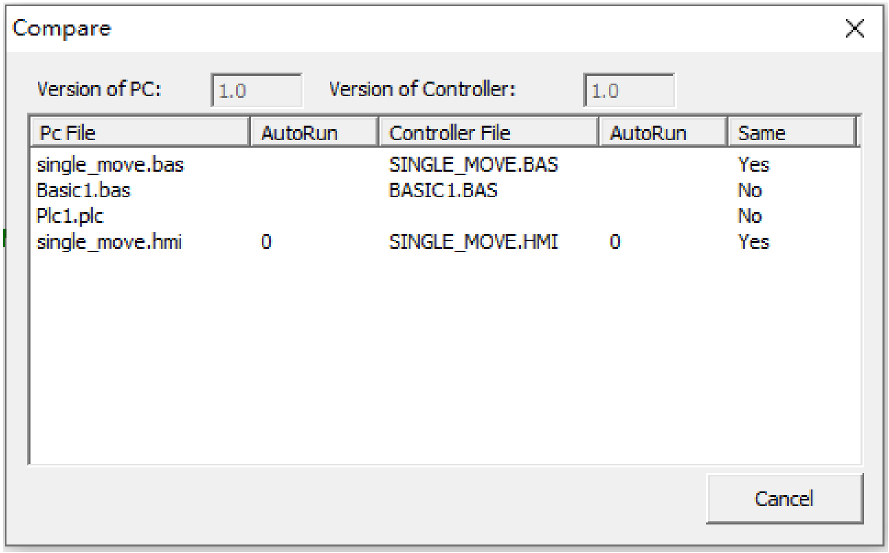
```
global sub sub1()
global sub sub2()
global sub sub3()
```

Open the program in Lib format at this time, you can see the declaration of the program body definition, only the global definition can be viewed, the local variable definition and the specific process of the SUB sub-function cannot be seen.

5.5 Compare Project

Click menu: Controller > Compare Project, to compare current program projects between PC and Controller.

If file is same, it will show “Yes”, if not, then show “No”.



Version of PC: 1.0   Version of Controller: 1.0

Pc File	AutoRun	Controller File	AutoRun	Same
single_move.bas		SINGLE_MOVE.BAS		Yes
Basic1.bas		BASIC1.BAS		No
Plc1.plc				No
single_move.hmi	0	SINGLE_MOVE.HMI	0	Yes

Note: for enryption purpose, code in controller can not be uploaded.

Chapter VI Controller Windows

6.1 File/Process/Configuration Window

File window, process window, and configuration window refer to Chapter IV.

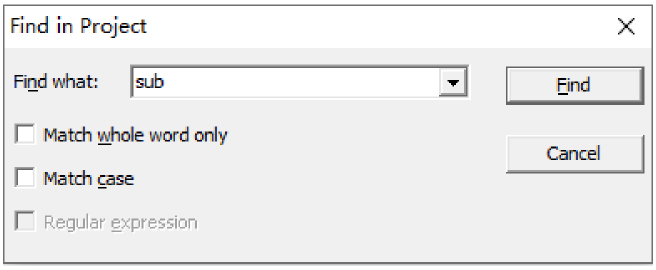
6.2 Command and Output Window

Command and Output window refers to Chapter II.

6.3 Find Result Window

"Find" window is used to quickly find matched keywords, the result searched from whole project file will show in "result window".

Find method: menu "Edit" – "Find in Project", open below window:



Find in Project

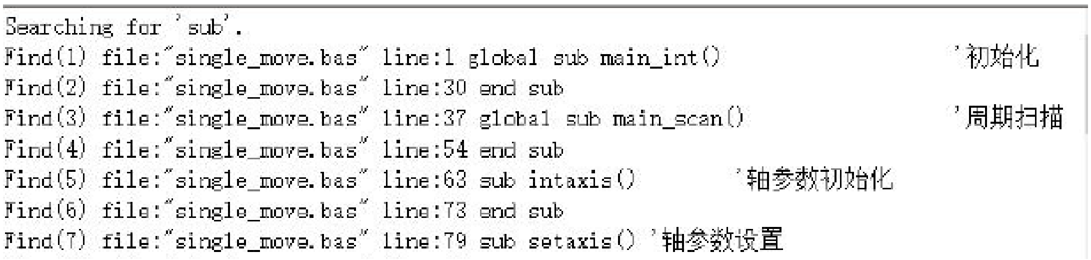
Find what: sub

□ Match whole word only

□ Match case

□ Regular expression

Researched result is below, it shows filename, line number and content, click relative line, program position can be located.

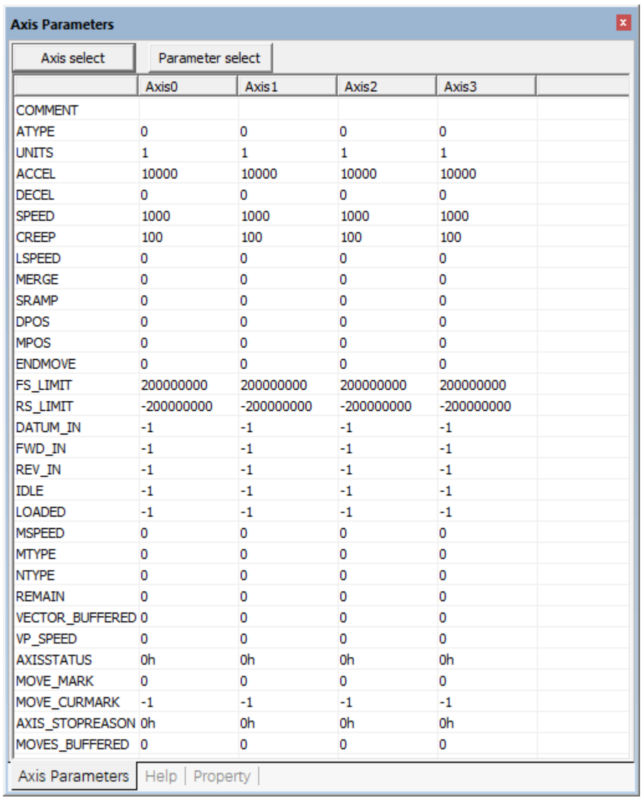


```
Searching for 'sub'.
Find(1) file:"single_move.bas" line:1 global sub main_int()                  '初始化
Find(2) file:"single_move.bas" line:30 end sub
Find(3) file:"single_move.bas" line:37 global sub main_scan()                '周期扫描
Find(4) file:"single_move.bas" line:54 end sub
Find(5) file:"single_move.bas" line:63 sub intaxis()        '轴参数初始化
Find(6) file:"single_move.bas" line:73 end sub
Find(7) file:"single_move.bas" line:79 sub setaxis() '轴参数设置
```

6.4 Axis Parameter Window

Click menu **View>Axis Parameter** to open axis parameters page.

In this page, we can select axes and parameters need to check. Axis parameters include speed, acceleration, Atype, etc, all parameter explanations are mentioned in ZBasic in details, please check in ZBasic. Below changeable parameters can be modified directly through double-click, but it is invalid for read only parameter.



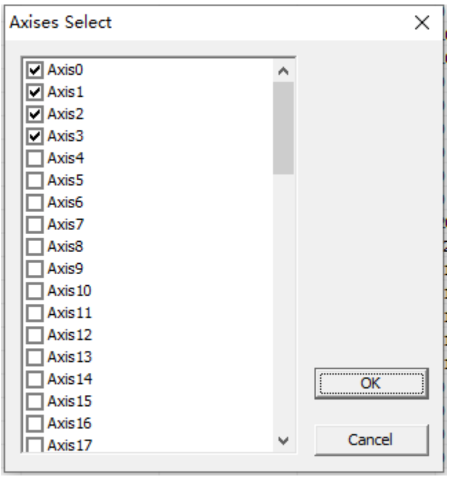
	Axis0	Axis1	Axis2	Axis3
COMMENT				
ATYPE	0	0	0	0
UNITS	1	1	1	1
ACCEL	10000	10000	10000	10000
DECEL	0	0	0	0
SPEED	1000	1000	1000	1000
CREEP	100	100	100	100
LSPEED	0	0	0	0
MERGE	0	0	0	0
SRAMP	0	0	0	0
DPOS	0	0	0	0
MPOS	0	0	0	0
ENDMOVE	0	0	0	0
FS_LIMIT	200000000	200000000	200000000	200000000
RS_LIMIT	-200000000	-200000000	-200000000	-200000000
DATUM_IN	-1	-1	-1	-1
FWD_IN	-1	-1	-1	-1
REV_IN	-1	-1	-1	-1
IDLE	-1	-1	-1	-1
LOADED	-1	-1	-1	-1
MSPEED	0	0	0	0
MTYPE	0	0	0	0
NTYPE	0	0	0	0
REMAIN	0	0	0	0
VECTOR_BUFFERED	0	0	0	0
VP_SPEED	0	0	0	0
AXISSTATUS	0h	0h	0h	0h
MOVE_MARK	0	0	0	0
MOVE_CURMARK	-1	-1	-1	-1
AXIS_STOPREASON	0h	0h	0h	0h
MOVES_BUFFERED	0	0	0	0

Parameter	Description
ATYPE	Axis type, 0-virtual axis, 1- pulse output, 3- orthogonal encoder input, 65-EtherCAT CSP mode
UNITS	Pulse equivalent means the pulse numbers are sent in one unit, which supports 5-bit decimal precision.
ACCEL/DECEL	Axis acceleration/deceleration, unit is units/s/s, when multi-axis moves, acceleration of axis group interpolation is master axis combined vector acceleration/ master axis combined vector deceleration. When DECEL is not set, it is ACCEL by default.
SPEED	Axis speed, unit is units/s. when multi axes move, it is the main axis vector speed as axis group interpolation.
CREEP	The creep speed when axis homing, it is used for origin searching, unit is units/s.
LSPEED	Axis starting speed, also it is stopping speed, default is 0, unit is units/s. when in multi-axis motion, it is joint vector speed of axis-group interpolation. It can be set not as 0 for achieving efficiency, but can’t be oversize.
MERGE	When it is ON, in the front and back buffer motions are connected together, speed keeps, which is used for continuous interpolation.
SRAMP	S curve setting when in acceleration and deceleration motion, unit is ms. It is time for axis-group to compound vector curve in multi axes motion.
DPOS	Expected coordinate position of axis, or target position, unit is units.
MPOS	Measured feedback position of axis, or actual position, unit is units.

ENDMOVE	The end target absolute position of current motion, unit is units.
FS_LIMIT	Axis forward software position limit. unit is units. If axis motion exceeds the boundary position, it will stop and report error “FSOFT”.
RS_LIMIT	Axis reverse software position limit. unit is units. If axis motion exceeds the boundary position, it will stop and report error “RSOFT”.
DATUM_IN	Input number related to origin switch position, -1 means invalid.
FWD_IN/REV_IN	Input number related to forward/reverse hardware position limit switch, -1 means invalid. When controller position limit signal took effect, axis will stop immediately, stop deceleration is FAST_DEC. Normally, FAST_DEC is set to be 10 multiples of DECEL.
IDLE	Axis current motion status judgement, it will return 0 when in motion, it will return -1 when motion ends, only for read.
LOADED	When there is no motion instructions to be buffered in motion buffer, it will return TRUE, or will return FALSE, only for read.
MSPEED	Axis actual speed form measured feedback, unit is units/s. MSPEED is differential value from MPOS, only for read.
MTYPE/NTYPE	Current running motion instruction type MTYPE/the first buffered motion instruction type NTYPE. When in joint interpolation, it is motion instruction type for slave axis to return to vector master axis. Only for read.
REMAIN	Return uncompleted distance of axis current motion MTYPE, unit is units. Only for read.
VECTOR_BUFFEREED	Return uncompleted distance of axis current motion and buffer motion, unit is units. It is compound vector distance for multi-axis interpolation, unit is units.
VP_SPEED	Return planned speed of current motion, unit is units/s. when in multi-axis motion, axis-group returns to compound speed of interpolation motion, not component velocity. If it is not master axis, it returns to related compound vector speed of axis and relative component velocity, the same effect as MSPEED, only for read.
AXISSTATUS	Check all bits’ status, such as, forward/reverse hardware position limit, forward/reverse software position limit, axis running status, etc. only for read.
MOVE_MARK	MARK label number of motion instruction, this is written into motion buffer together with motion instructions. Each motion command is called, MOVE_MARK will add one automatically. If need to force to assign MOVE_MARK, it needs to be set once before motion. It can pause in different MARK boundaries through MOVE_PAUSE.
MOVE_CURMARK	Return MOVE_MARK label number of current axis which is running.

More parameters can be referred through Axis parameter and axis status instruction in

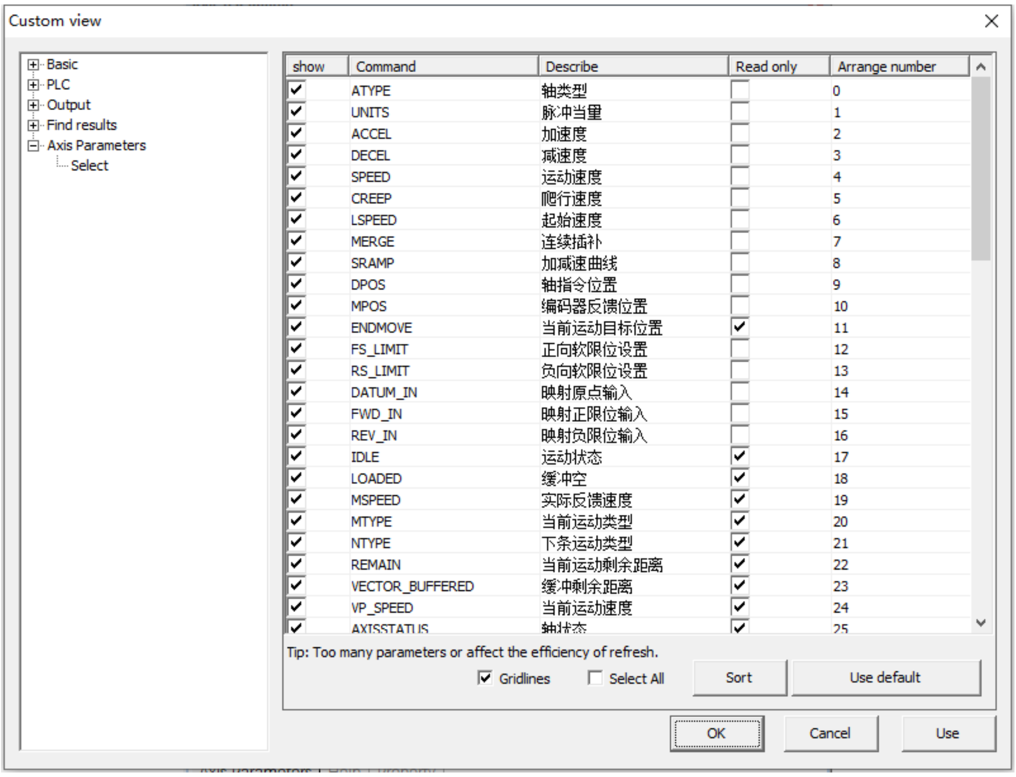
Click “Axis Select”, axis can be monitored.



Axises Select

☑ Axis0
☑ Axis1
☑ Axis2
☑ Axis3
□ Axis4
□ Axis5
□ Axis6
□ Axis7
□ Axis8
□ Axis9
□ Axis10
□ Axis11
□ Axis12
□ Axis13
□ Axis14
□ Axis15
□ Axis16
□ Axis17

Click “Parameter Select”, below window will appear, axis parameter can be customized for monitor, but it is not recommended to set too many, avoid affecting refresh efficiency.



Custom view

show	Command	Describe	Read only	Arrange number
☑	ATYPE	轴类型	□	0
☑	UNITS	脉冲当量	□	1
☑	ACCEL	加速度	□	2
☑	DECEL	减速度	□	3
☑	SPEED	运动速度	□	4
☑	CREEP	爬行速度	□	5
☑	LSPEED	起始速度	□	6
☑	MERGE	连续插补	□	7
☑	SRAMP	加减速曲线	□	8
☑	DPOS	轴指令位置	□	9
☑	MPOS	编码器反馈位置	□	10
☑	ENDMOVE	当前运动目标位置	☑	11
☑	FS_LIMIT	正向软限位设置	□	12
☑	RS_LIMIT	负向软限位设置	□	13
☑	DATUM_IN	映射原点输入	□	14
☑	FWD_IN	映射正限位输入	□	15
☑	REV_IN	映射负限位输入	□	16
☑	IDLE	运动状态	☑	17
☑	LOADED	缓冲空	☑	18
☑	MSPEED	实际反馈速度	☑	19
☑	MTYPE	当前运动类型	☑	20
☑	NTYPE	下条运动类型	☑	21
☑	REMAIN	当前运动剩余距离	☑	22
☑	VECTOR_BUFFERED	缓冲剩余距离	☑	23
☑	VP_SPEED	当前运动速度	☑	24
☑	AXISSTATUS	轴状态	☑	25

Tip: Too many parameters or affect the efficiency of refresh.

☑ Gridlines □ Select All

6.5 Manual Motion Window

Click menu **View**>**Manual** to enter Manual page, in this page, user can operate connected motor by manual. User is able to select axis, set axis parameters: Atype, Units, Accel, Decel, Speed.

Left parameters can be modified real-time, press “LeftMove” / “RightMove” (not loosen), motor will continue to move in left / right, the button is released, the motion will stop.

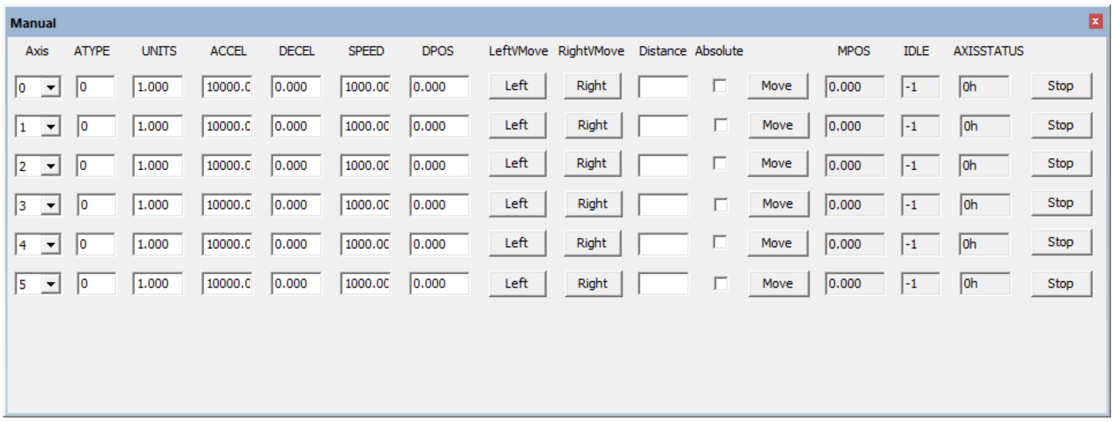
“DPOS” shows current DPOS motion distance (unit is units).

When fill “Distance”, select “Absolute”, the motor will move to absolute distance. When “Absolute” is not selected, motor moves in relative distance.

Axis number can be selected from the list.

“MPOS” / “IDLE” / “AXISSTATUS” these 3 parameters are only for read, they can’t be modified.

“Stop” is pressed, axis motion will stop immediately.

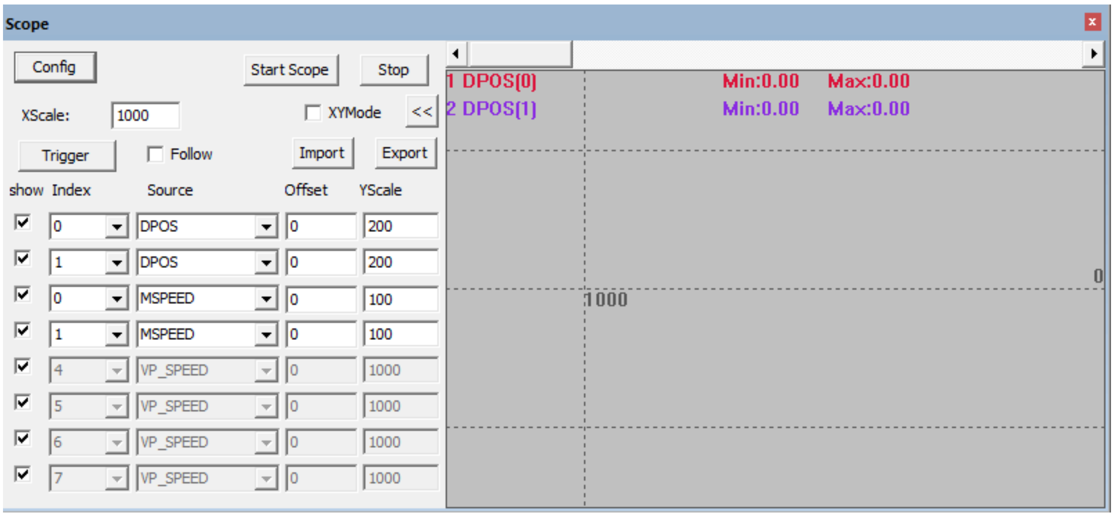


Manual

Axis	ATYPE	UNITS	ACCEL	DECEL	SPEED	DPOS	LeftVMove	RightVMove	Distance	Absolute		MPOS	IDLE	AXISSTATUS	
0	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop
1	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop
2	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop
3	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop
4	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop
5	0	1.000	10000.0	0.000	1000.00	0.000	Left	Right		□	Move	0.000	-1	0h	Stop

6.6 Oscilloscope Window

Oscilloscope is extremely important of program debugging and running. It is used to transfer signals that can’t be seen by naked eyes into graphics, so it is convenient to analyze change processes of all kinds of signals. Oscilloscope shows controller internal data in graph, it can display different signals, like, axis parameter, axis status, etc. “View” – “Scope”, then oscilloscope window can be opened, or click the shortcut button .



Scope

Config Start Scope Stop

XScale: 1000 □ XYMode <<

Trigger □ Follow Import Export

show	Index	Source	Offset	YScale
☑	0	DPOS	0	200
☑	1	DPOS	0	200
☑	0	MSPEED	0	100
☑	1	MSPEED	0	100
☑	4	VP_SPEED	0	1000
☑	5	VP_SPEED	0	1000
☑	6	VP_SPEED	0	1000
☑	7	VP_SPEED	0	1000

The oscilloscope must be triggered to successfully sample. Turn on the oscilloscope, click "Start Scope" after setting the relevant parameters. Also, it could be manually triggered sampling, or add the "TRIGGER" instruction to the program to automatically trigger the oscilloscope sampling.

Oscilloscope basic setting button function:

Config: open oscilloscope configuration window, set relevant parameters

Start Scope: start oscilloscope (but not to trigger oscilloscope capture)

Stop: stop oscilloscope capture

XY mode: when checked, switch to the XY plane to display the interpolated composite track of the two axes.

<<: press to hide the channel name and peak value, and only display the channel number.

Trigger: manually trigger oscilloscope capture button (use TIGGER instruction to trigger automatically)

Follow: After the following is enabled, the horizontal axis automatically moves to the real-time sampling point and follows the waveform display.

Show: choose whether the present channel curve shows or not

Index: select the data source number that needs to be captured, such as, axis number, digital IO number, analog IO number, TABLE number, VR number, MODUBUS number, etc.

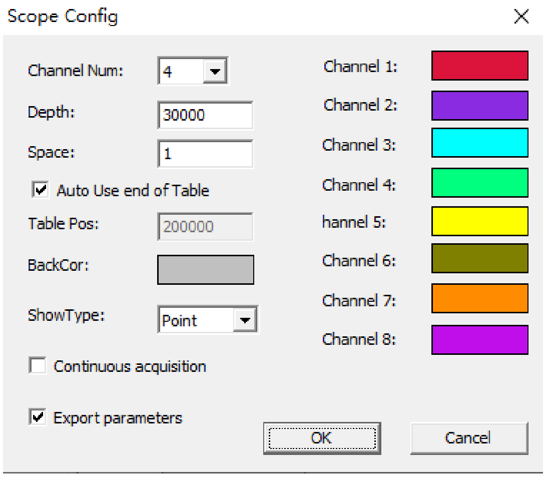
Source: select captured data type

Offset: Set the offset of the vertical axis of the waveform.

Xscale: The scale of one grid on the vertical axis.

Yscale: The scale of one grid on the horizontal axis.

If sets oscilloscope parameters, like, axis number, data source and start oscilloscope configuration window, it should stop firstly, then to do configure.



Scope Config

Channel Num: 4

Depth: 30000

Space: 1

☑ Auto Use end of Table

Table Pos: 200000

BackCor:

ShowType: Point

□ Continuous acquisition

☑ Export parameters

Chanel Num: channel total numbers to be sampled.

Depth: total sampling numbers, the depth is bigger, the sampling ranger is bigger.

Space: sampling time space. The unit is system cycle, which is related to the firmware version of the controller. Generally, default is 1ms, and see it through SERVO_PERIOD instruction. Usually, the smaller the space, the more accurate the sampled data, and the larger the amount of data per unit time.

TABLE Pos: Set the position where the captured data is stored. Generally, the space at the end of the TABLE data is automatically used by default, and you can also customize the configuration, but be careful not to overlap with the TABLE data area used by the program when setting.

BackCor/Channel Color: set background and the color related to each channel's waveform.

ShowType: there are two curve types to be chosen, point and line. Abnormal data is easily found out by Line.

Continuous acquisition: when the continuous acquisition is not enabled, the sampling will stop when the sampling depth is reached. After the continuous acquisition is enabled, the oscilloscope will continue to sample.
Export parameters: export oscilloscope parameters.

Import and export data:

Import： data can be imported when oscilloscope stops, and if it is imported successfully, sampling waveform can reappear.
The method of importing sampling data: click “Import”, import data file that should be the former file type exported from oscilloscope, then open it.

Export: export parameters, including oscilloscope parameter configuration, data type of each channel and each sampling point’s data.
The method of exporting data: click “Export parameters” in “Scope Config”, then start scope. After sampling, click “Export”, and select folder to store oscilloscope data, the exported data is text file.

Oscilloscope Sampling Method:

➢ Open project, connect to controller or simulator, then open the oscilloscope window (note: first, connect to controller or simulator, then operate the oscilloscope window).
➢ Click “Scope Config” in oscilloscope window, select sampling channel numbers, sampling depth, sampling space, sampling data TABLE stored position (generally the end position of TABLE array will be used automatically) and sampling type, etc. Then, click “OK” for saving this configuration.
➢ Select sampling Index and Source, click “Start Scope”.
➢ Download program into controller, the program should include TRIGGER, the trigger oscilloscope sampling automatically instruction. Now, it starts sampling and shows different data source’s waveforms. It can adjust Show Scale and Wave Offset, this is for observing different waveforms.

If the waveform accuracy is not high or the display is incomplete, click the "Stop" button and then open the "Scope Config", and adjust the sampling space and sampling depth, and perform the above sampling process again.
If the sampling time is long, start “Continuous acquisition” function.

Precautions for Using Oscilloscope:

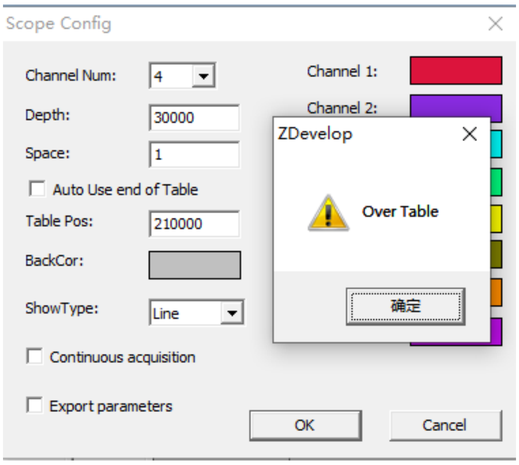
Oscilloscope sampling time calculation:

✧ For example, depth: 1000, space: 5
✧ If system cycle is SERVO_PERIOD=1000, which means 1ms trajectory planning cycle. Space 5 means sampling one data point per 5ms. Total sampling data number is 10000, so sampling time length is 50s.

TABLE data end storage space calculation:

✧ Set the position where the captured data is stored. Generally, the space at the end of the TABLE data is automatically used by default, now starting space address is calculated automatically according to captured data space.
✧ Calculation method: captured data space = channel numbers * depth

- For example, if TABLE space of controller is 320000, there are 4 sampling channels, depth is 30000, each sampling point occupies one TABLE, so it will occupy 4*30000=120000 TABLE positions. 320000-120000=200000, which means starting position of TABLE is 200000.
- The position for storing data also can be self-defined, if according to above channel number and depth, TABLE starting space can't exceed 200000 when it is self-defined, otherwise, it can't be configured, please see below picture:



Scope Config

Channel Num: 4

Depth: 30000

Space: 1

□ Auto Use end of Table

Table Pos: 210000

BackCor:

ShowType: Line

□ Continuous acquisition

□ Export parameters

Channel 1:

Channel 2:

ZDevelop

Over Table

- The space occupied by the oscilloscope sample data should not overlap with the TABLE data area used by the program.
- Controller TABLE space can be read through TSIZE instruction, check "Controller Status" or input ?*max to print and check.

Continuous acquisition:

- When continuous acquisition is not selected, the oscilloscope automatically stops sampling when the sampling depth is reached.
- First select "Continuous acquisition" in "Scope Config", then start oscilloscope, it will continue to sampling after triggered, and sampling even if it reached the depth. It will stop until press "Stop" button manually.
- All waveforms and captured data from continuous acquisition can be exported.

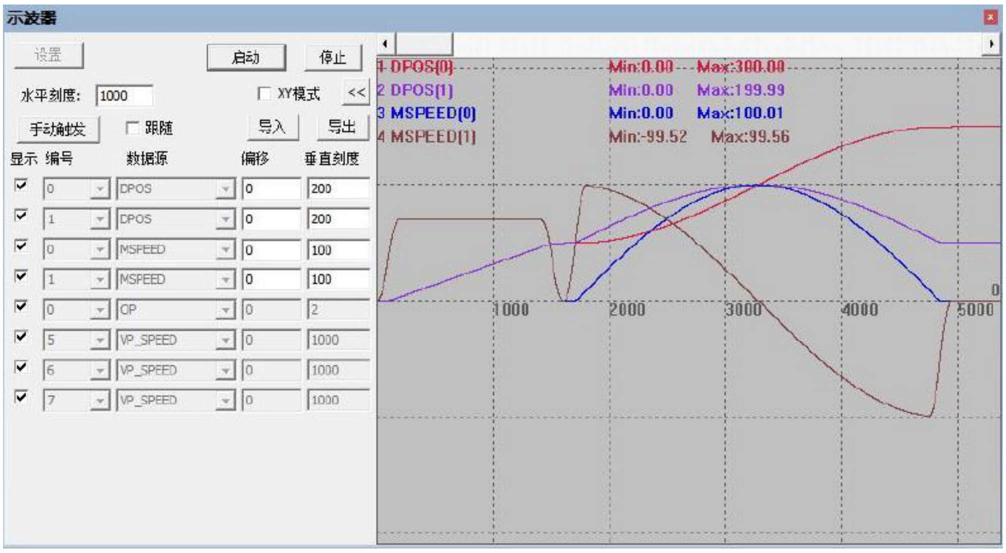
Oscilloscope Usage Routine:

Example 1: Continuous trajectory look-ahead application

```
RAPIDSTOP(2)
WAIT IDLE(0)
WAIT IDLE(1)

BASE(0,1)
DPOS=0,0
ATYPE=1,1
UNITS=100,100
SPEED=100,100
ACCEL=1000,1000
DECEL=1000,1000
SRAMP=100,100
MERGE=ON
```

```
CORNER_MODE=2                                 'start corner deceleration
DECEL_ANGLE = 15 * (PI/180)                   'set angle of starting deceleration
STOP_ANGLE = 45 * (PI/180)                    'set angle of ending deceleration
FORCE_SPEED=100 TRIGGER                       'trigger oscilloscope automatically
MOVE(100,100)
MOVECIRC(200,0,100,0,1)    'Radius 100 draw a semi-circle clockwise, end coordinates (300,100)
```

Speed and position curve of sampling axis 0 and axis 1:

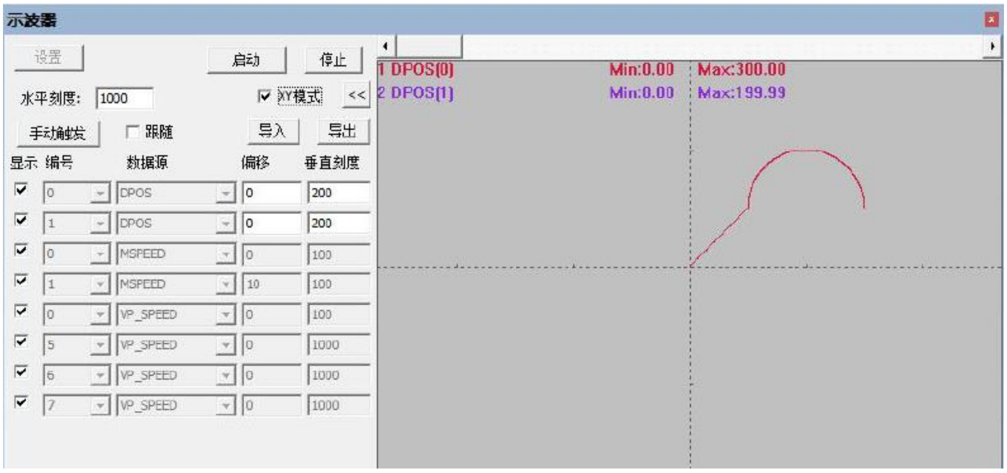


水平刻度: 1000 □ XY模式 □ 跟随

显示	编号	数据源	偏移	垂直刻度
☑	0	DPOS	0	200
☑	1	DPOS	0	200
☑	0	MSPEED	0	100
☑	1	MSPEED	0	100
☑	0	OP	0	2
☑	5	VP_SPEED	0	1000
☑	6	VP_SPEED	0	1000
☑	7	VP_SPEED	0	1000

1 DPOS(0) Min:0.00 Max:300.00
2 DPOS(1) Min:0.00 Max:199.99
3 MSPEED(0) Min:0.00 Max:100.01
4 MSPEED(1) Min:-99.52 Max:99.56

Two-axis interpolation synthetic trajectory in XY mode:

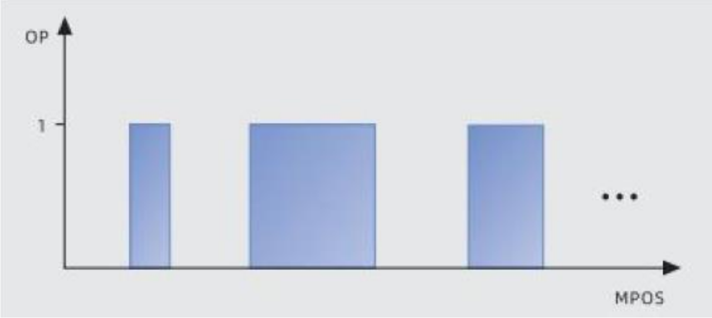


水平刻度: 1000 ☑ XY模式 □ 跟随

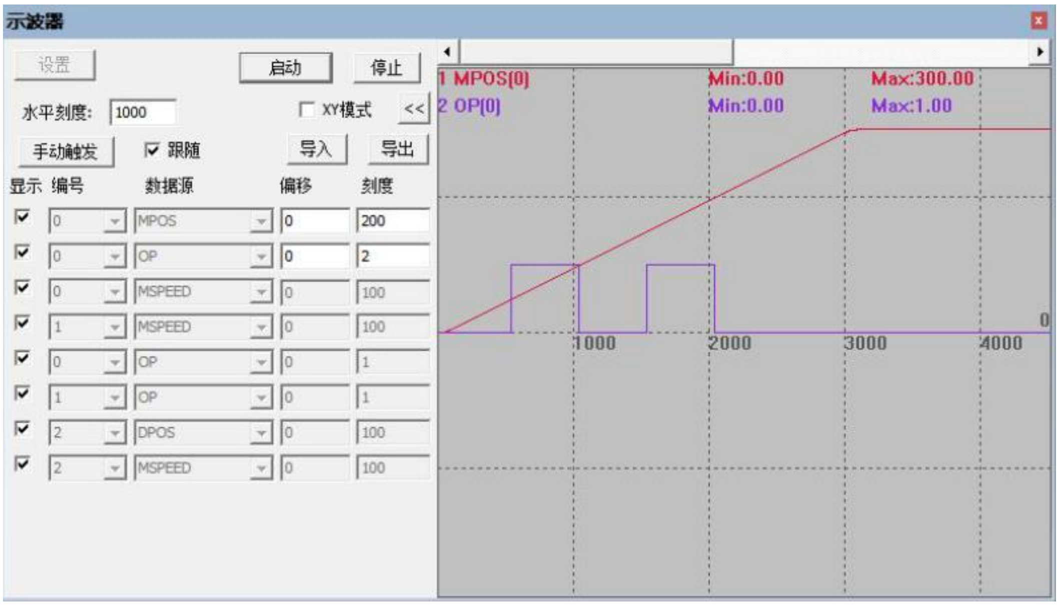
显示	编号	数据源	偏移	垂直刻度
☑	0	DPOS	0	200
☑	1	DPOS	0	200
☑	0	MSPEED	0	100
☑	1	MSPEED	10	100
☑	0	VP_SPEED	0	100
☑	5	VP_SPEED	0	1000
☑	6	VP_SPEED	0	1000
☑	7	VP_SPEED	0	1000

1 DPOS(0) Min:0.00 Max:300.00
2 DPOS(1) Min:0.00 Max:199.99

Example 2: PSO position synchronization output, output OP signal when arriving comparison point



```
RAPIDSTOP(2)
WAIT IDLE(0)

BASE(0)

DPOS=0
MPOS=0
ATYPE=1
UNITS=100
SPEED=100
ACCEL=1000
DECEL=1000
OP(0,OFF)
TABLE(0,50,100,150,200)                         'coordinate of comparison point
HW_PSWITCH2(2)                                  'stop and delete incomplete comparison points
HW_PSWITCH2(1, 0, 1, 0, 3,1)                    'compare 4 points, operate output 0
TRIGGER                                         'trigger oscilloscope automatically
MOVE(300)
```



水平刻度: 1000

□ XY模式

☑ 跟随

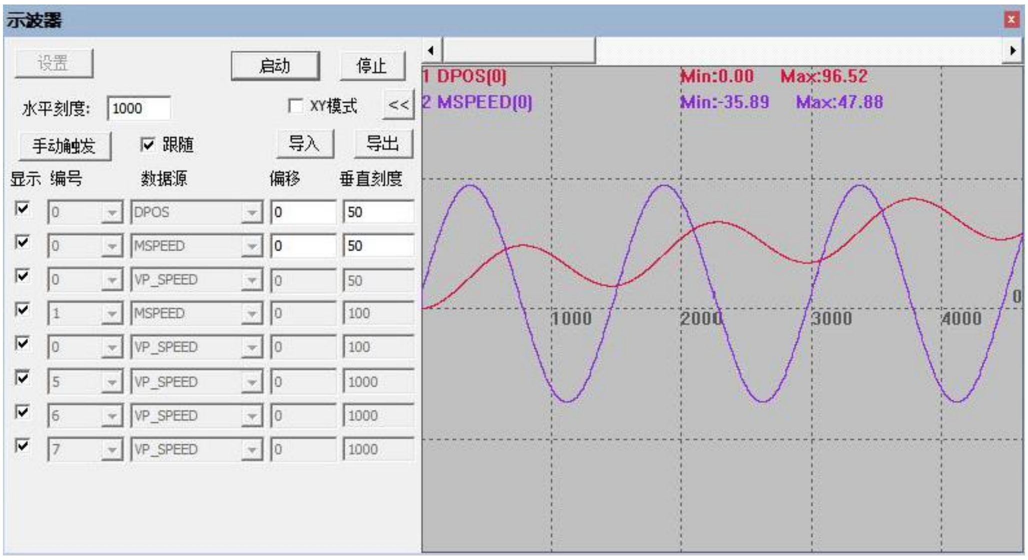
显示	编号	数据源	偏移	刻度
☑	0	MPOS	0	200
☑	0	OP	0	2
☑	0	MSPEED	0	100
☑	1	MSPEED	0	100
☑	0	OP	0	1
☑	1	OP	0	1
☑	2	DPOS	0	100
☑	2	MSPEED	0	100

Example 3: Electronic Cam Application

```
RAPIDSTOP(2)
WAIT IDLE(0)
```

```
BASE(0)                                        ′select axis 0
ATYPE=1                                        ′pulse directional step or servo
DPOS = 0
UNITS = 100                                    ′pulse equivalent
SPEED = 200
ACCEL = 2000
DECEL = 2000

′Calculate TABLE data
DIM deg, rad, x, stepdeg
stepdeg = 2          ′use this to modify line number, line is more, speed is more stable
FOR deg = 0 TO 360 STEP stepdeg
    rad = deg * 2 * PI/360                     ′convert to radian
    X = deg * 25 + 10000 * (1-COS (rad))       ′calculate offset of each small segment
    TABLE (deg/stepdeg, X)                     ′store TABEL
    TRACE deg/stepdeg, X
NEXT deg

TRIGGER                                        ′trigger oscilloscope sampling
WHILE 1                                        ′cycle motion
CAM (0, 360/stepdeg, 0.1, 300)                 ′the virtual follow length is 300
WAIT UNTIL IDLE                                ′wait until motion stops
END
```

Motion trajectory: total time of each cam instruction = distance / speed = 300/200 = 1.5s



水平刻度: 1000

□ XY模式

☑ 跟随

显示	编号	数据源	偏移	垂直刻度
☑	0	DPOS	0	50
☑	0	MSPEED	0	50
☑	0	VP_SPEED	0	50
☑	1	MSPEED	0	100
☑	0	VP_SPEED	0	100
☑	5	VP_SPEED	0	1000
☑	6	VP_SPEED	0	1000
☑	7	VP_SPEED	0	1000

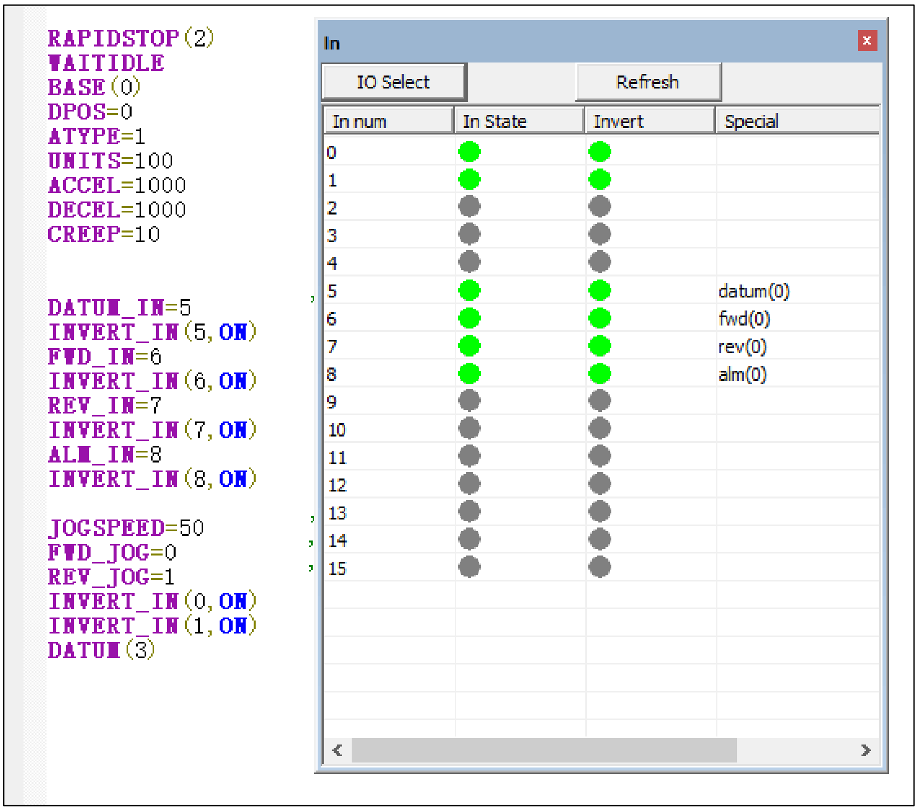
6.7 Input IO Window

Click menu **View>In** to enter input state page, in this page, click “IO select” button to select Inputs need to check, click “refresh” button to view special functions attached to inputs.

"Special" is to show some special signal inputs: homing, position limit, alarm, etc.

In some situation, we will use ZBasic instruction: INVERT_IN to reverse input signal. (it needs to reverse signal after ZMC Series special input definition, but no need for ECI Series, because ZMC Series is valid in OFF.) normally, input state is green when signal comes; after setting "INVERT_IN", invert state turn to green, input state is gray when normally open signal comes. Check "INVERT_IN" in ZBasic for details.

For example: JOG motion

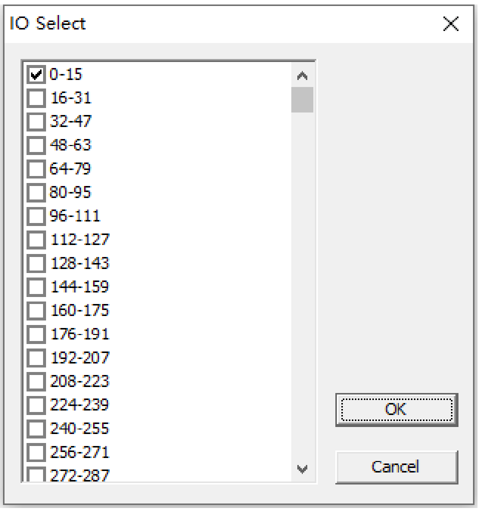


```
RAPIDSTOP(2)
WAITIDLE
BASE(0)
DPOS=0
ATYPE=1
UNITS=100
ACCEL=1000
DECEL=1000
CREEP=10

DATUM_IN=5
INVERT_IN(5,ON)
FWD_IN=6
INVERT_IN(6,ON)
REV_IN=7
INVERT_IN(7,ON)
ALM_IN=8
INVERT_IN(8,ON)

JOGSPEED=50
FWD_JOG=0
REV_JOG=1
INVERT_IN(0,ON)
INVERT_IN(1,ON)
DATUM(3)
```

In num	In State	Invert	Special
0	🟢	🟢	
1	🟢	🟢	
2	⚫	⚫	
3	⚫	⚫	
4	⚫	⚫	
5	🟢	🟢	datum(0)
6	🟢	🟢	fwd(0)
7	🟢	🟢	rev(0)
8	🟢	🟢	alm(0)
9	⚫	⚫	
10	⚫	⚫	
11	⚫	⚫	
12	⚫	⚫	
13	⚫	⚫	
14	⚫	⚫	
15	⚫	⚫	

IN0, IN1, IN5, IN6, IN7 and IN8 all defined to inverse input signals (ZMC Series motion controller default OFF is valid, so it will inverse, ECI won't), only IN5 has input, others have no input.

Click "IO Select", it will show input IN, one group has 16 inputs, see below. Click "Refresh", it will refresh each IO status and special function definitions.



☑ 0-15
☐ 16-31
☐ 32-47
☐ 48-63
☐ 64-79
☐ 80-95
☐ 96-111
☐ 112-127
☐ 128-143
☐ 144-159
☐ 160-175
☐ 176-191
☐ 192-207
☐ 208-223
☐ 224-239
☐ 240-255
☐ 256-271
☐ 272-287

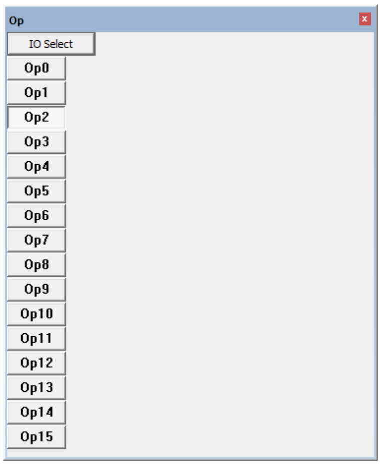
6.8 Output IO Window

Click menu **View>Op** to open output status page.

Press the button, OP can be operated, please see below, OP 1 and OP 2 are opened, others are still closed.

When link with simulator, output IO status cam be checked through output and monitor window, but it only shows number 0-11 status.

“IO Select” is used to select output to be displayed, each group has 16 outputs. The output status can be switched through clicking output directly.



6.9 Task / Monitor Window

Task and monitor windows belong to program debug, these two windows can be used to view task running situation and adding monitor project, Task and Monitor window can be referred to Chapter VII.

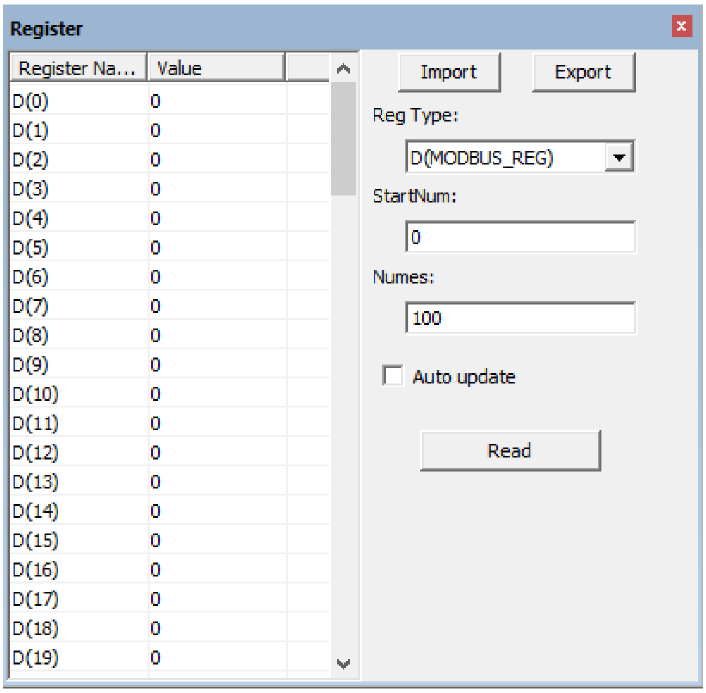
6.10 Register Window

Click menu **View>Register** to open register page, in this page, different kinds of register can be viewed directly, including modbus, VR, table, input, output etc. Note: this function only available in controllers that support PLC function.

How to use: Select the type, starting number, and number of the register to be read, and then click "Read" to display the data in the window. Be careful not to read more than the register range, otherwise it will prompt an error. The refresh function is convenient to automatically collect and display the change of register value in real time, otherwise, you need to click again to read the value change of the register. Click "Import"/"Export" to quickly upload/download register data.

Export data is convenient for customers to export some registers that they care about to text to save.

Importing data allows customers to directly change the saved data of the concerned part into the controller.



Register Na...	Value
D(0)	0
D(1)	0
D(2)	0
D(3)	0
D(4)	0
D(5)	0
D(6)	0
D(7)	0
D(8)	0
D(9)	0
D(10)	0
D(11)	0
D(12)	0
D(13)	0
D(14)	0
D(15)	0
D(16)	0
D(17)	0
D(18)	0
D(19)	0

Reg Type: D(MODBUS_REG)

StartNum: 0

Numes: 100

□ Auto update

Register Type:

X(IN): input relay, receive external signals, related to IN;

Y(OP): output relay, drive external load, related to OP;

S: state relay, control work steps;

M: auxiliary relay, can’t drive external load, related to MODBUS_BIT;

D(MODBUS_REG):16 bits integer register, Modbus area data, related to MODBUS_REG

D(MODBUS_LONG):32 bits integer register, Modbus area data, related to MODBUS_LONG

D(MODBUS_IEEE):32 bits float register, Modbus area data, related to MODBUS_IEEE

D(MODBUS_STRING):1 byte character string register, Modbus area data, related to MODBUS_STRING

AIN: analog input

AOUT: analog output

DT(TABLE):32 bits float register, related to TABLE;

V: index register,16 bits;

Z: index register,16 bits;

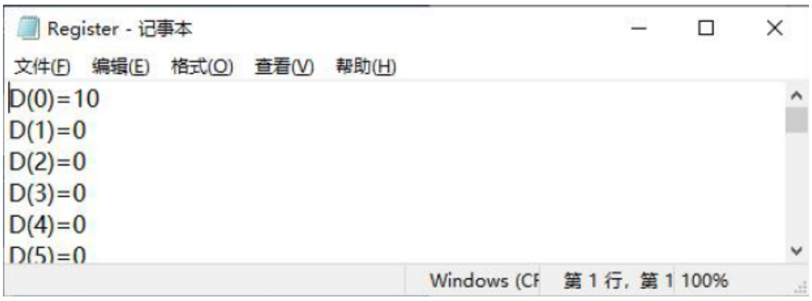
T: time, unit is ms;

C: Counter;

VR: retentive register,32 bits float;

VR_INT: retentive register,32 bits integer.

Exported data example:

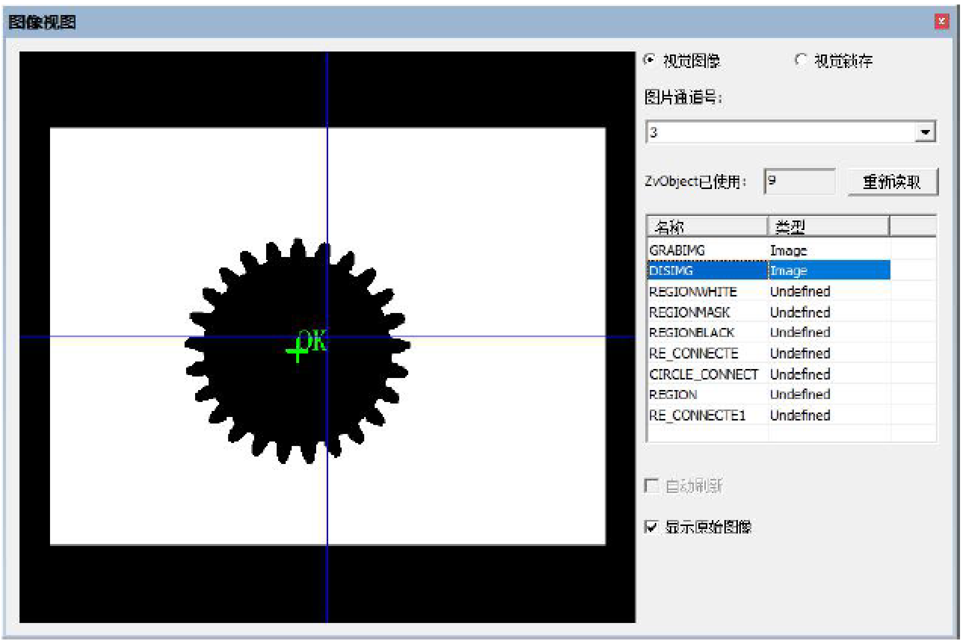

Register - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Windows (CF 第 1 行，第 1 100%

```
D(0)=10
D(1)=0
D(2)=0
D(3)=0
D(4)=0
D(5)=0
```

6.11 Graphic Window

Click “View” – “Register”, the graphic window can be opened. Use this window to view visual images in a machine vision development environment. The image window can only be used normally after the image is captured. It can be switched between the display of all visual images and the display of visual latched channel images.

Visual image can select all the pictures under the current project to display, including the collected images and processed images.



● 视觉图像 ○ 视觉锁存

图片通道号: 3

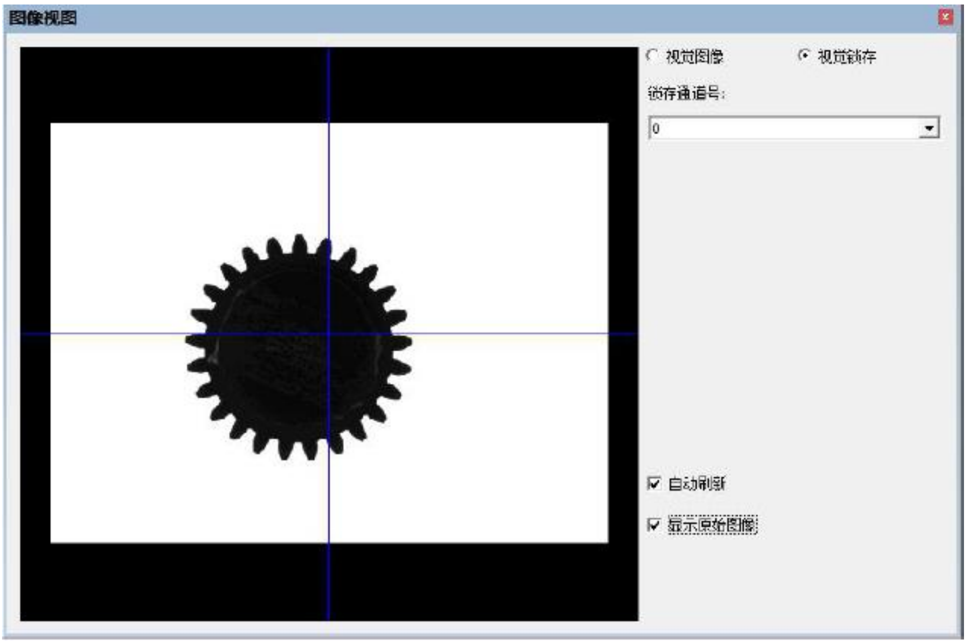
ZvObject已使用: 9 [重新读取]

名称	类型
GRABIMG	Image
DISIMG	Image
REGIONWHITE	Undefined
REGIONMASK	Undefined
REGIONBLACK	Undefined
RE_CONNECTE	Undefined
CIRCLE_CONNECT	Undefined
REGION	Undefined
RE_CONNECTE1	Undefined

□ 自动刷新

☑ 显示原始图像

Before displaying the image of the visual latch, you need to select the latch channel number first, and then display the image of the current latch channel. When the current latch channel is empty, there is no image.



○ 视觉图像 ● 视觉锁存

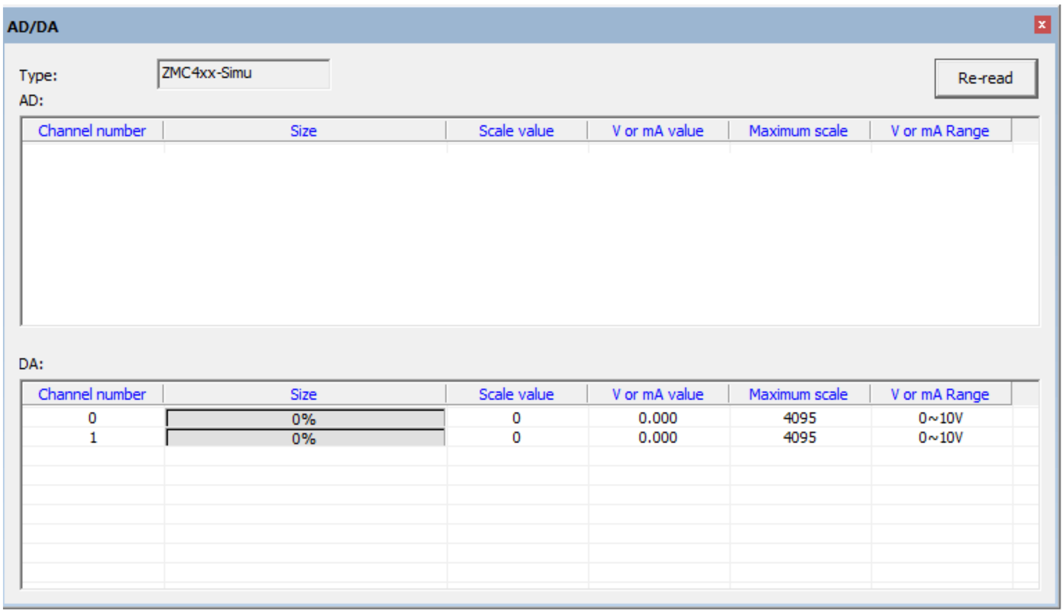
锁存通道号: 0

☑ 自动刷新

☑ 显示原始图像

6.12 AD/DA Window

Click menu **View>AD/DA** to view Ads or DAs status of controller.



Type: ZMC4xx-Simu

AD:

Channel number	Size	Scale value	V or mA value	Maximum scale	V or mA Range

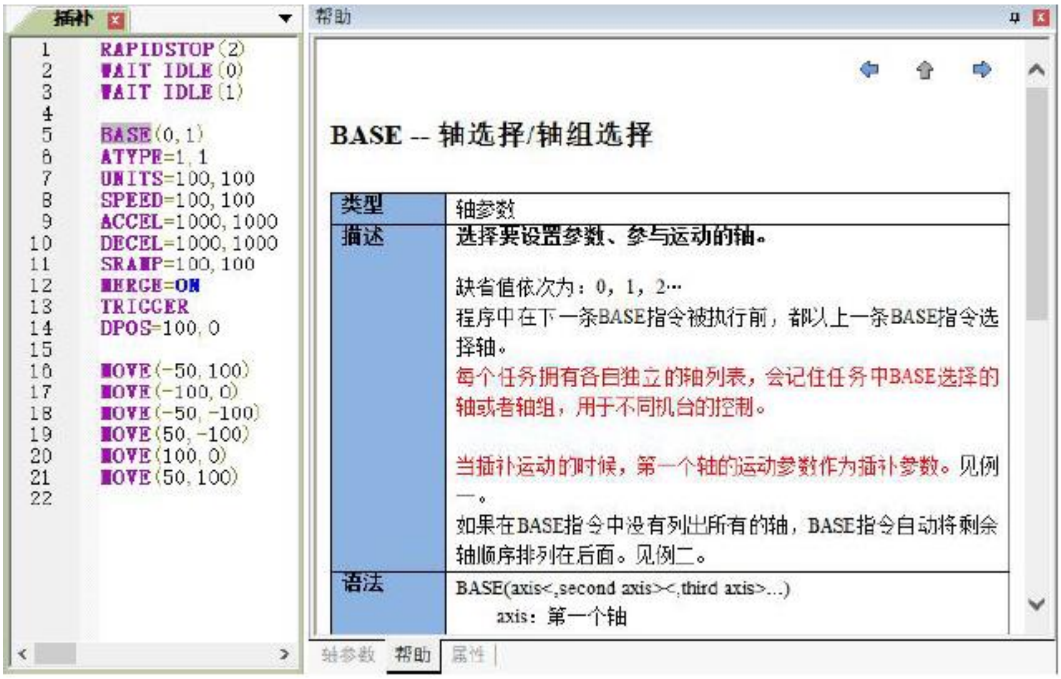
DA:

Channel number	Size	Scale value	V or mA value	Maximum scale	V or mA Range
0	0%	0	0.000	4095	0~10V
1	0%	0	0.000	4095	0~10V

6.13 Help Window

Click “View” – “Help”, the window opens.

Select instructions you want to check, press F1 or click the instruction, help documents will be opened quickly.



```
RAPIDSTOP(2)
WAIT IDLE(0)
WAIT IDLE(1)

BASE(0,1)
ATYPE=1,1
UNITS=100,100
SPEED=100,100
ACCEL=1000,1000
DECEL=1000,1000
SRAMP=100,100
MERGE=ON
TRIGGER
DPOS=100,0

MOVE(-50,100)
MOVE(-100,0)
MOVE(-50,-100)
MOVE(50,-100)
MOVE(100,0)
MOVE(50,100)
```

BASE -- 轴选择/轴组选择

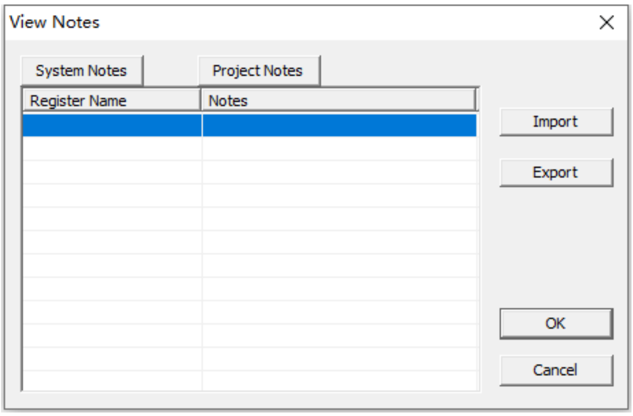
类型	轴参数
描述	**选择要设置参数、参与运动的轴。** 缺省值依次为：0，1，2… 程序中在下一条BASE指令被执行前，都以上一条BASE指令选择轴。 每个任务拥有各自独立的轴列表，会记住任务中BASE选择的轴或者轴组，用于不同机台的控制。 当插补运动的时候，第一个轴的运动参数作为插补参数。见例一。 如果在BASE指令中没有列出所有的轴，BASE指令自动将剩余轴顺序排列在后面。见例二。
语法	BASE(axis<,second axis><,third axis>...) axis：第一个轴

6.14 Notes Window

Click “View” – “Notes”.

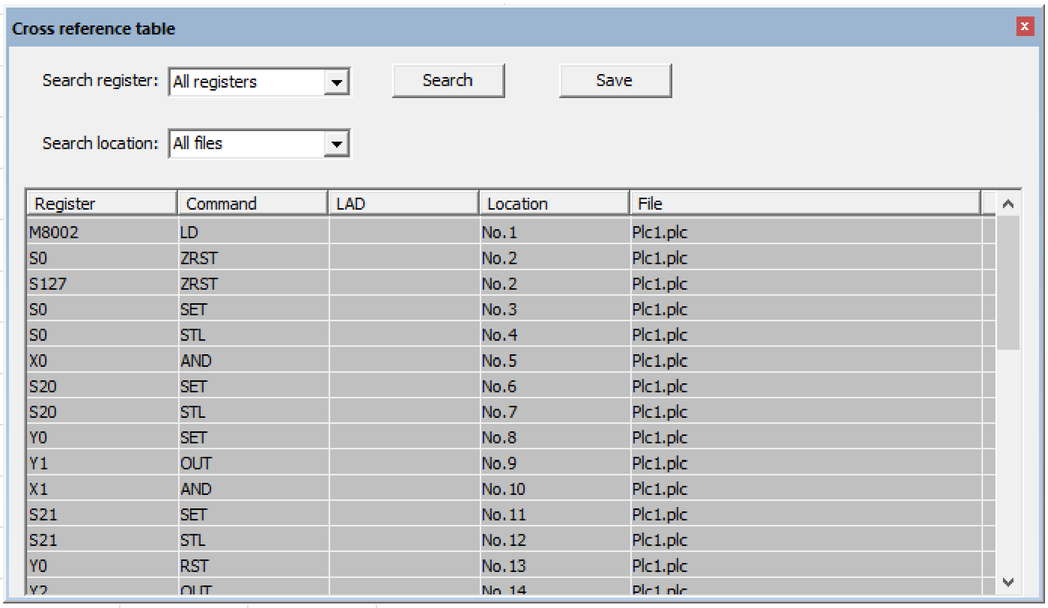
System Notes: check the function of register that has been noted by system.

Project Notes: add register and relative note content.



6.15 Cross Reference Table

Click menu **View>Cross Reference Table** to view used registers information, double click register label to skip to its position in program. This table is suitable for PLC programming.



Search register: All registers

Search location: All files

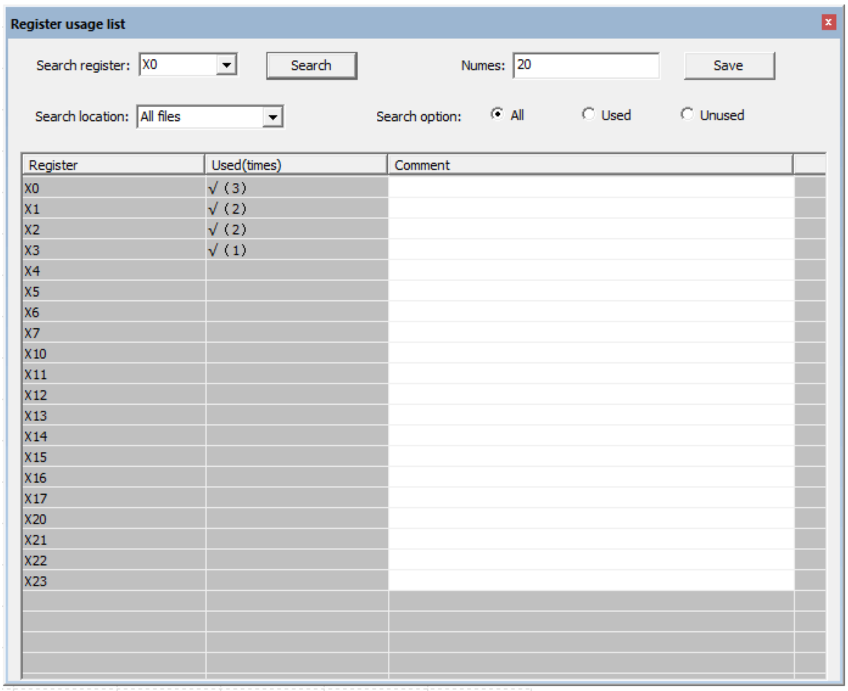
Register	Command	LAD	Location	File
M8002	LD		No.1	Plc1.plc
S0	ZRST		No.2	Plc1.plc
S127	ZRST		No.2	Plc1.plc
S0	SET		No.3	Plc1.plc
S0	STL		No.4	Plc1.plc
X0	AND		No.5	Plc1.plc
S20	SET		No.6	Plc1.plc
S20	STL		No.7	Plc1.plc
Y0	SET		No.8	Plc1.plc
Y1	OUT		No.9	Plc1.plc
X1	AND		No.10	Plc1.plc
S21	SET		No.11	Plc1.plc
S21	STL		No.12	Plc1.plc
Y0	RST		No.13	Plc1.plc
Y2	OUT		No.14	Plc1.plc

6.16 Register Usage List

Click menu **View>Register Usage List** to view registers use status.

Double left click gray area of label, **Cross Reference Table** of this label will pop;

Double left click blank area of label to add comments.



Register usage list

Search register: X0 Search Numes: 20 Save

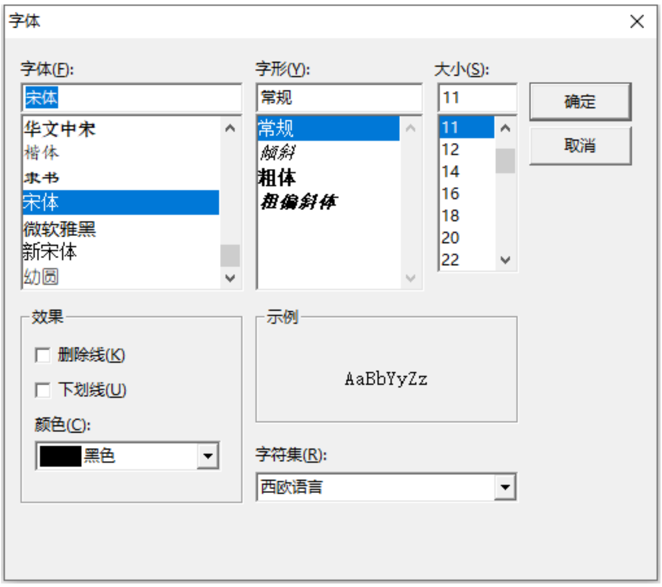
Search location: All files Search option: ● All ○ Used ○ Unused

Register	Used(times)	Comment
X0	√（3）	
X1	√（2）	
X2	√（2）	
X3	√（1）	
X4		
X5		
X6		
X7		
X10		
X11		
X12		
X13		
X14		
X15		
X16		
X17		
X20		
X21		
X22		
X23		

6.17 Fond View

Click “View” – “Fond”.

User can modify default fond, size, color, etc.



字体

字体(F): 宋体

华文中宋
楷体
隶书
宋体
微软雅黑
新宋体
幼圆

字形(Y): 常规

常规
倾斜
粗体
粗偏斜体

大小(S): 11

11
12
14
16
18
20
22

效果

□ 删除线(K)

□ 下划线(U)

颜色(C): ■ 黑色

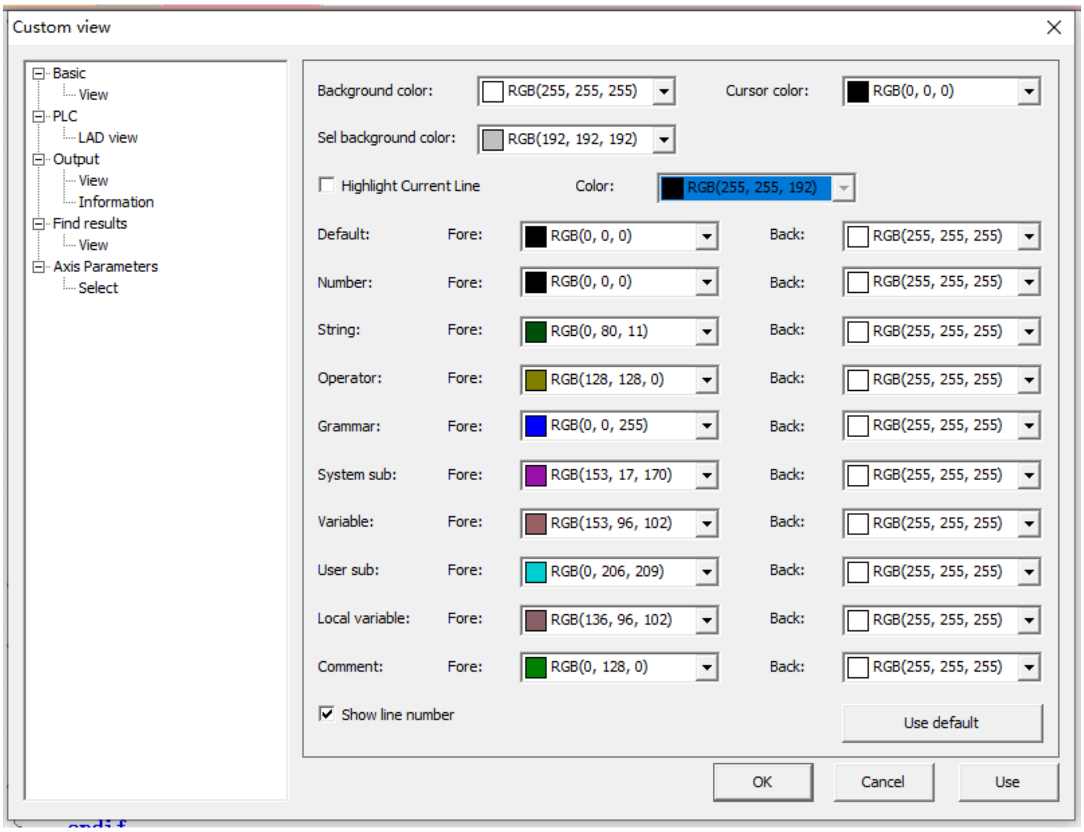
示例

AaBbYyZz

字符集(R): 西欧语言

6.18 Custom View

Click menu **View>Custom** to set ZDevelop display style, also displayed parameters can be set here.



Custom view

Background color: RGB(255, 255, 255)   Cursor color: RGB(0, 0, 0)

Sel background color: RGB(192, 192, 192)

□ Highlight Current Line   Color: RGB(255, 255, 192)

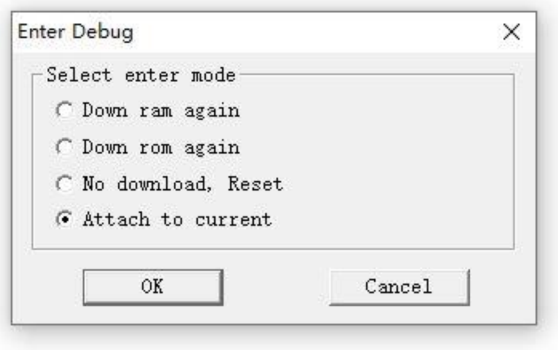
Default:	Fore:	RGB(0, 0, 0)	Back:	RGB(255, 255, 255)
Number:	Fore:	RGB(0, 0, 0)	Back:	RGB(255, 255, 255)
String:	Fore:	RGB(0, 80, 11)	Back:	RGB(255, 255, 255)
Operator:	Fore:	RGB(128, 128, 0)	Back:	RGB(255, 255, 255)
Grammar:	Fore:	RGB(0, 0, 255)	Back:	RGB(255, 255, 255)
System sub:	Fore:	RGB(153, 17, 170)	Back:	RGB(255, 255, 255)
Variable:	Fore:	RGB(153, 96, 102)	Back:	RGB(255, 255, 255)
User sub:	Fore:	RGB(0, 206, 209)	Back:	RGB(255, 255, 255)
Local variable:	Fore:	RGB(136, 96, 102)	Back:	RGB(255, 255, 255)
Comment:	Fore:	RGB(0, 128, 0)	Back:	RGB(255, 255, 255)

☑ Show line number

Chapter VII Debug

7.1 Enter Debug

Click menu **Debug>Start/Stop Debug** to enter debug mode, choose right mode to start.



Enter Debug

Select enter mode

○ Down ram again

○ Down rom again

○ No download, Reset

● Attach to current

Down ram again: load project to ram again and start to debug, not saved when power off.

Down rom again: load project to rom again and start to debug, support power failure storage.

No download, Reset: restart controller again and run program that has been downloaded, open task window to show current operation status.

Attach to current: debug current operating project, this program is not downloaded, only open task window to show current operation status.

When errors come in operating project, ZDevelop will show error information, double-click the error information, it will locate program position, which means take it for reference to help debugging. If there is no error information showing, we can also view errors by typing in command:?*task, or open “Troubleshooting” window,.

Take below routine as example:

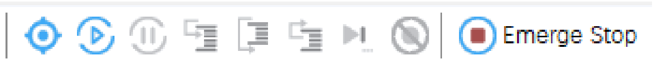
```
RAPIDSTOP(2)
WAIT IDLE(0)
WAIT IDLE(1)                ′select axis 0, axis 1

ATYPE=1,1
UNITS=100,100
SPEED=100,100                ′motion speed
ACCEL=1000,1000
DECEL=1000,1000
SRAMP=100,100               ′S curve
MERGE=ON                     ′open continuous interpolation
TRIGGER                      ′trigger oscilloscope automatically
DPOS=100,0                   ′coordinate offset

MOVE(-50,100)               ′relative motion 1
MOVE(-100,0)                ′relative motion 2
MOVE(-50,-100)          ′relative motion 3
MOVE(50,-100)           ′relative motion 4
MOVE(100,0)             ′relative motion 5
MOVE(50,100)            ′relative motion 6
END
```

Command and output window prints line 16 of interpolation file appears grammar error, error is reported, which means program can’t be downloaded into controller. After solving the problem, redownload into program.

Also, error information can be checked through **“Troubleshooting” window**.

7.2 Debug Tool

Debug tool bar is only valid in debug mode.



Reset: execute from the start

Run (F5): continue to run or pause

Step into (F11): run into program

Step over (F10): run to next program

Step out (Shift + F11): jump out SUB subprogram operation

Run to (Ctrl + F10): run to assigned line

Toggle breakpoint (F9): set breakpoint, first click to set, second click to cancel.

Emerge Stop: stop program and motion of all axes compulsively.

Note:

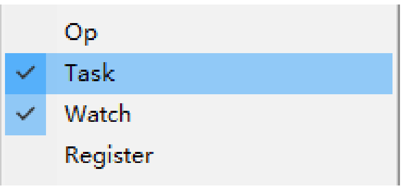
1.in debug mode, when procedure in ZDevelop is not same as in Controller, debug indicating line will offset. To avoid errors, do upgrade controller procedure if there is modification in ZDevelop procedure.

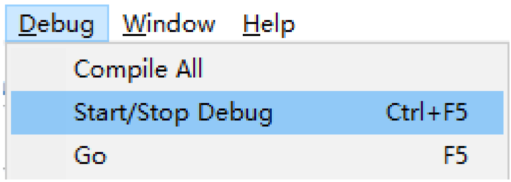
2.When click “pause”, motion in buffers will not stop.

7.3 Task Window

Click menu **View>Task** to open task window. Or click “Debug” – “Start/Stop Debug” to call task and watch window.

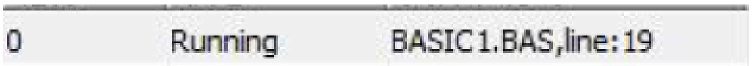
Please note task and watch window only can be opened under debug mode.





Controller supports multi-task running, and the details can be checked when program is in motion. Below example is single-task, the max tasks to be supported can be viewed in “Controller Status” – “Tasked”.

Controller supports multi-task running, after entering debug mode, “View” – “Task” can check each task’s specific status. Below example is single task, the task number is 0, status is Running, and executed file is Basic1.Bas, executed line number is line:19.



0 Running BASIC1.BAS,line:19

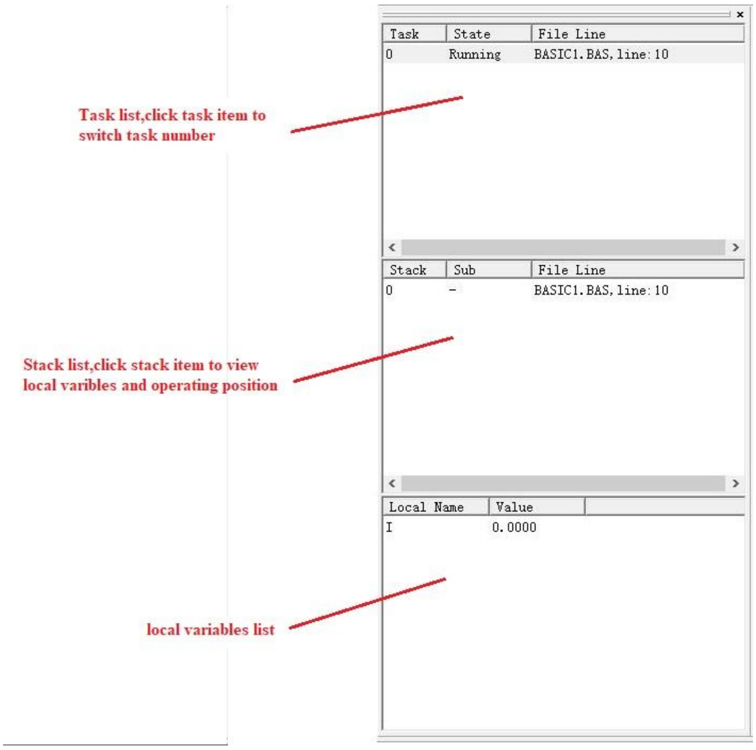
Task window consists of three areas:

Task tracing: view all tasks status in operating project.

Stack tracing: when SUB process is called, its status and local variables will be saved in stack. We can view these parameters in stack then.

Local variables: all variables defined by “Local” or input parameters through calling SUB process can be viewed here.

(note: the same task has different local variables of different stakes, even though their names are the same; stake’s layer is limited, normally, stake called by subprogram is 8 levels, be careful to use recursion structure.)



Task	State	File Line
0	Running	BASIC1.BAS, line: 10

Stack	Sub	File Line
0	-	BASIC1.BAS, line: 10

Local Name	Value
I	0.0000

Task list,click task item to switch task number
Stack list,click stack item to view local varibles and operating position
local variables list

7.4 Watch Window

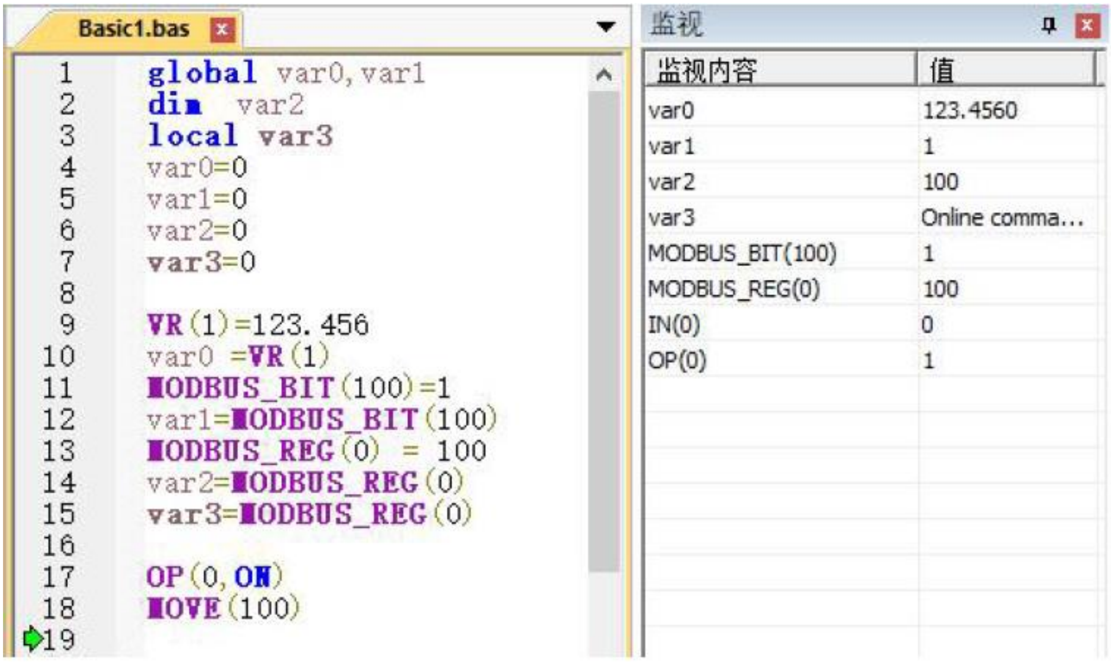
Click **View>Watch** to open Watch window, in watch window, we can add global or modular variables, registers, IOs, valid expressions, etc. Then, trace the values.

Two ways to add variables to watch window:

1.Select variables, registers, expression in code, open right click menu to add them to watch window.

2.In watch window, double click blank areas under “Watch Name” item to add or modify variables need to watch.

Note: Local variables can’t be traced in Watch Window, need to trace in task window. Please see below, var3 is local variable, Watch window can’t get the value.



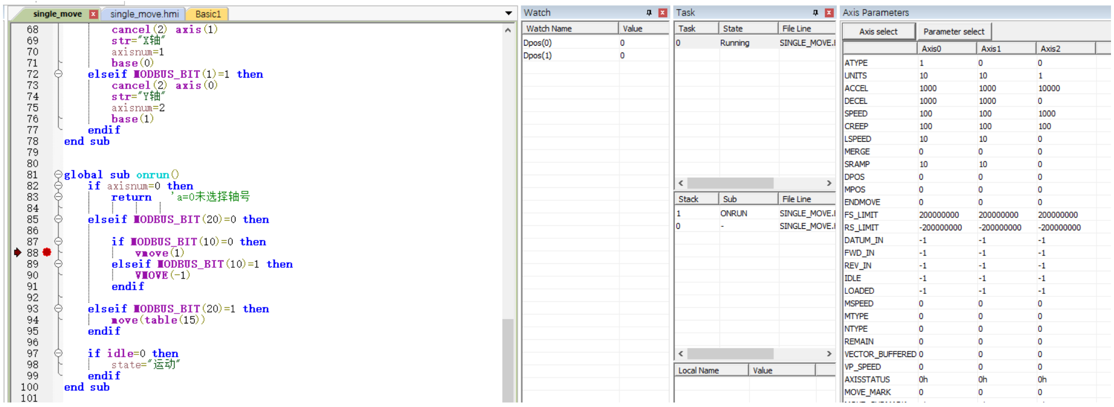
```
global var0,var1
dim  var2
local var3
var0=0
var1=0
var2=0
var3=0

VR(1)=123.456
var0 =VR(1)
MODBUS_BIT(100)=1
var1=MODBUS_BIT(100)
MODBUS_REG(0) = 100
var2=MODBUS_REG(0)
var3=MODBUS_REG(0)

OP(0,ON)
MOVE(100)
```

监视内容	值
var0	123.4560
var1	1
var2	100
var3	Online comma...
MODBUS_BIT(100)	1
MODBUS_REG(0)	100
IN(0)	0
OP(0)	1

7.5 Breakpoint

We can add breakpoint in program, to find out execution of every program step execution on registers, variables, array etc. F9 is shortcut for breakpoint add and delete. Or "Debug" – "Toggle/Edit Breakpoint", several breakpoints can be added. "Debug" – "Clear All breakpoints" is used to clear all breakpoints in project file once.

After adding breakpoints, program will stop in breakpoint position. Now, instruction in breakpoint line has not been executed, it won't influence scanned program.



```
        cancel(2) axis(1)
        str="X轴"
        axisnum=1
        base(0)
    elseif MODBUS_BIT(1)=1 then
        cancel(2) axis(0)
        str="Y轴"
        axisnum=2
        base(1)
    endif
end sub


global sub onrun()
    if axisnum=0 then
        return   'a=0未选择轴号

    elseif MODBUS_BIT(20)=0 then

        if MODBUS_BIT(10)=0 then
            vmove(1)
        elseif MODBUS_BIT(10)=1 then
            VMOVE(-1)
        endif

    elseif MODBUS_BIT(20)=1 then
        move(table(15))
    endif

    if idle=0 then
        state="运动"
    endif
end sub
```

Watch Name	Value
Dpos(0)	0
Dpos(1)	0

Task	State	File Line
0	Running	SINGLE_MOVE.I

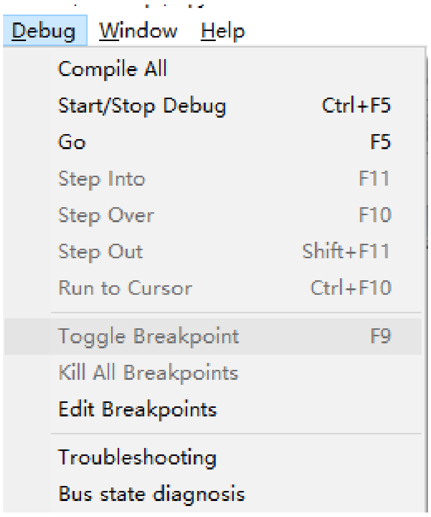
Stack	Sub	File Line
1	ONRUN	SINGLE_MOVE.I
0	-	SINGLE_MOVE.I

Local Name	Value

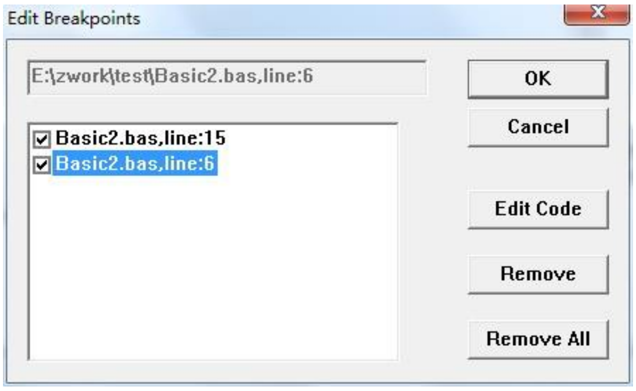
	Axis0	Axis1	Axis2
ATYPE	1	0	0
UNITS	10	10	1
ACCEL	1000	1000	10000
DECEL	1000	1000	0
SPEED	100	100	1000
CREEP	100	100	100
LSPEED	10	10	0
MERGE	0	0	0
SRAMP	10	10	0
DPOS	0	0	0
MPOS	0	0	0
ENDMOVE	0	0	0
FS_LIMIT	200000000	200000000	200000000
RS_LIMIT	-200000000	-200000000	-200000000
DATUM_IN	-1	-1	-1
FWD_IN	-1	-1	-1
REV_IN	-1	-1	-1
IDLE	-1	-1	-1
LOADED	-1	-1	-1
MSPEED	0	0	0
MTYPE	0	0	0
NTYPE	0	0	0
REMAIN	0	0	0
VECTOR_BUFFERED	0	0	0
VP_SPEED	0	0	0
AXISSTATUS	0h	0h	0h
MOVE_MARK	0	0	0

When program stops at breakpoint, we can execute forward by step by clicking F11, each click program will execute one step forward.

If breakpoint is set in the cycle, then next run to breakpoint position again, program still will stop.



Click menu **Debug>Edit Breakpoints** to enter Edit Breakpoint page, here user can edit all breakpoint, such as, to remove one or multi breakpoints, double click breakpoint label to skip to its position in program.



E:\zwork\test\Basic2.bas,line:6

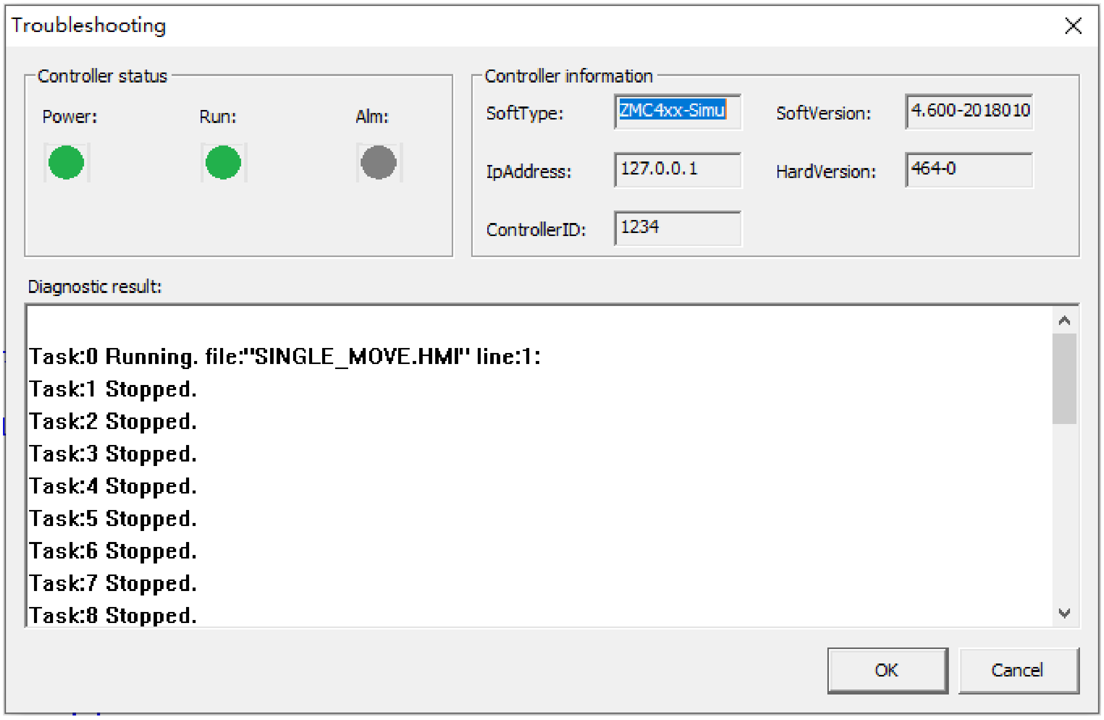
☑ Basic2.bas,line:15
☑ Basic2.bas,line:6

When debug is finished, please clear all breakpoints before load program into controller again, or errors will come: Warn file:"Basic1.BAS" line:11 task:0, Paused.

7.6 Troubleshooting

Click menu **Debug>Troubleshooting** to view check errors in tasks. User can check controller firmware or hardware version and date, task operation status, errors information, where the program stops etc.

Run and Alarm led can be set by manual here also, current connected controller can be found rapidly in multi-controller.

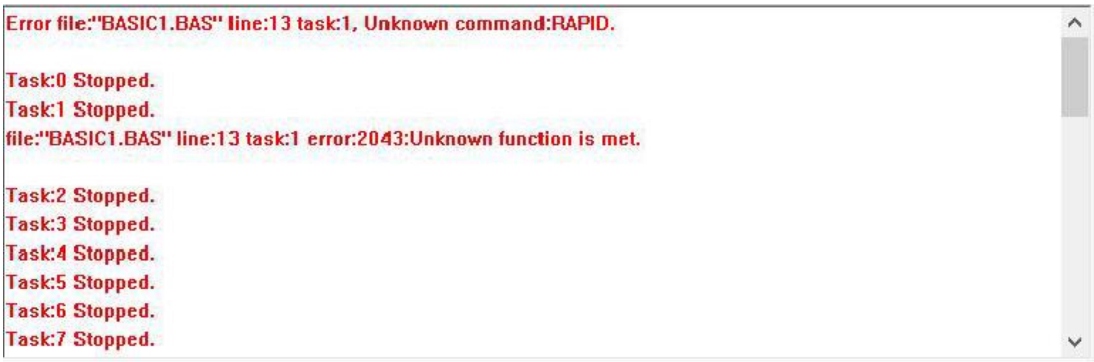


SoftType: ZMC4xx-Simu
SoftVersion: 4.600-2018010
IpAddress: 127.0.0.1
HardVersion: 464-0
ControllerID: 1234

Diagnostic result:

```
Task:0 Running. file:"SINGLE_MOVE.HMI" line:1:
Task:1 Stopped.
Task:2 Stopped.
Task:3 Stopped.
Task:4 Stopped.
Task:5 Stopped.
Task:6 Stopped.
Task:7 Stopped.
Task:8 Stopped.
```

Error information:

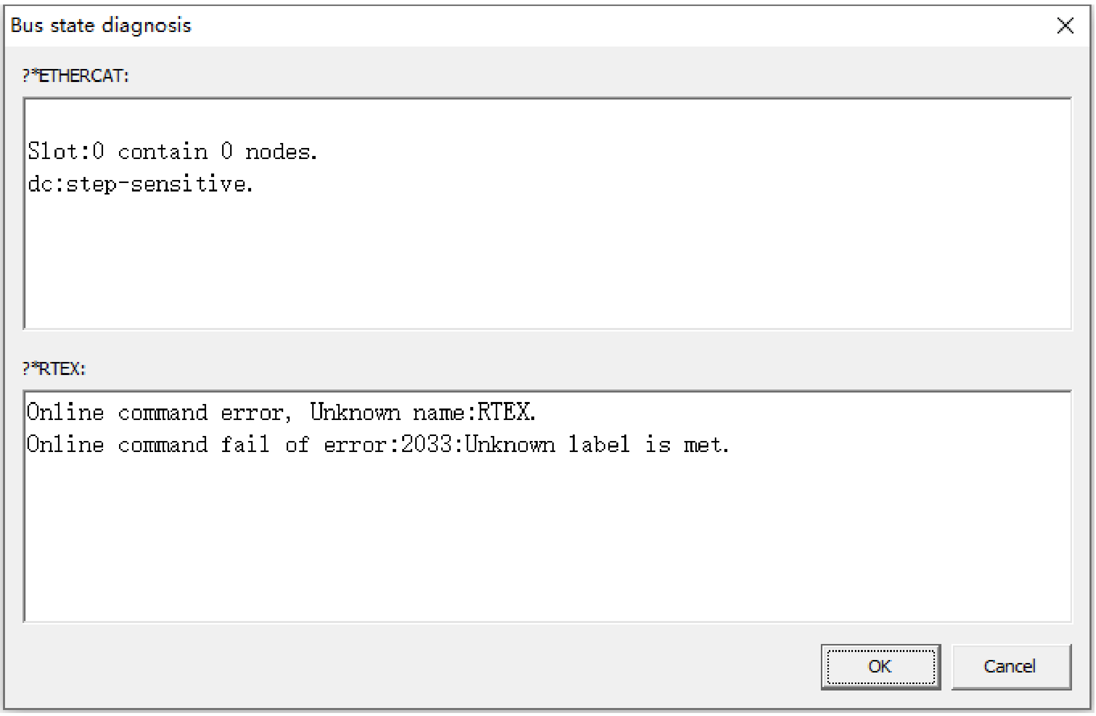


```
Error file:"BASIC1.BAS" line:13 task:1, Unknown command:RAPID.

Task:0 Stopped.
Task:1 Stopped.
file:"BASIC1.BAS" line:13 task:1 error:2043:Unknown function is met.

Task:2 Stopped.
Task:3 Stopped.
Task:4 Stopped.
Task:5 Stopped.
Task:6 Stopped.
Task:7 Stopped.
```

7.7 Bus State Diagnosis

“Debug” – “Bus State Diagnosis” window can see the current bus supported by controller and device information of all nodes connected by bus.

Print information definition refers to “?*ETHERCAT” for help. This window’s data result same as ?*ETHERCAT and ?*RTEX.



```
Slot:0 contain 0 nodes.
dc:step-sensitive.
```

```
Online command error, Unknown name:RTEX.
Online command fail of error:2033:Unknown label is met.
```

7.8 Status Bar

Status bar shows 3 parts, which can be used to show cursor positioning, printed information counting, controller mode/IP/status.

1. **Cursor positioning:**

show line and row information of cursor place (such as, Basic, line 23, row 1, 437 characters numbers indicate there are 16 lines program in the front of cursor, cursor is in behind the first character of this line, there are 437 from first line to cursor position; PLC, ROW:30, Col:2, which means cursor is in the second grid position of line 30).

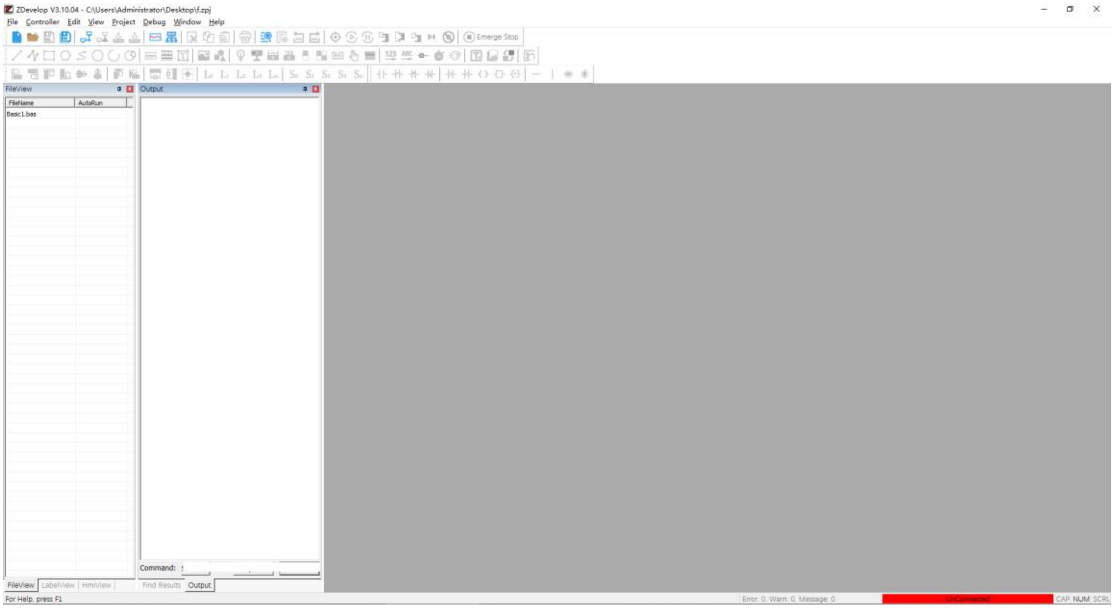
2. **Count printed information:**

Count error information of current instructions and output window, alarm, the number of printed information.

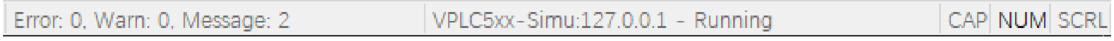
3. **Show controller mode/IP/status:**

Show current connected controller mode/IP/running status, when there is ALM or ERROR, a hint will appear in right corner in red background.

If there is no controller connected, red fond “Unconnected” in right corner.

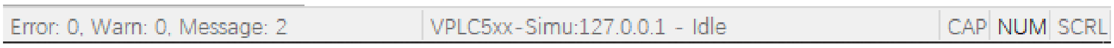

ZDevelop V3.10.04 - C:\Users\Administrator\Desktop\f.zpj
File Controller Edit View Project Debug Window Help
Emerge Stop
FileView
FileName
AutoRun
Basic1.bas
Output
Command:
FileView LabelView HmiView
Find Results Output
For Help, press F1
Error: 0, Warn: 0, Message: 0
UnConnected
CAP NUM SCRL

Controller runs normally, there show “Running” status



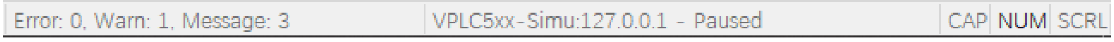
Error: 0, Warn: 0, Message: 2 | VPLC5xx-Simu:127.0.0.1 - Running | CAP NUM SCRL

Controller doesn’t run or emerge stop, there shows “idle”



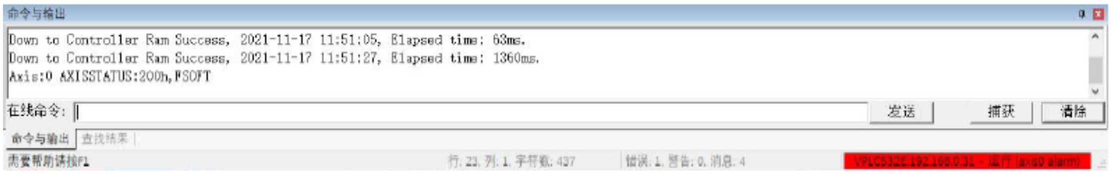
Error: 0, Warn: 0, Message: 2 | VPLC5xx-Simu:127.0.0.1 - Idle | CAP NUM SCRL

Click “Debug” – “Pause”, program scanning pause:



Error: 0, Warn: 1, Message: 3 | VPLC5xx-Simu:127.0.0.1 - Paused | CAP NUM SCRL

There is “Red” hint in motion:



```
Down to Controller Ram Success, 2021-11-17 11:51:05, Elapsed time: 63ms.
Down to Controller Ram Success, 2021-11-17 11:51:27, Elapsed time: 1360ms.
Axis:0 AXISSTATUS:200h,FSOFT
```

行: 23, 列: 1, 字符数: 437 | 错误: 1, 警告: 0, 消息: 4 | VPLC532E:192.168.0.31 - 运行 (axis0 alarm)

We can see from the above, axis 0 axis alarm, customer can check axis0 axisstatus or check printed information.

Chapter VIII Common Errors

When errors come in operating project, ZDevelop will show error information, take it for reference to help debugging, we can also view errors by typing in command:?*task in ZDevelop command box, double click error information to reach related code lines.

Error	Reasons or Solutions
Download button is gray	1.project not built or no files in project.
	2.not connected with controller.
	3.some controllers only support ROM download, RAM is not valid.
WARN: no program set autorun.	1.no file with auto run task number.
Error:5002, Operate Failed!	1.there are more than 1 HMI file or task in project, delete redundant HMI files.
Controller alarm Led is on	1.input "?*task" to check error information again
	2.check if axis status values are 0 through AXISSTATUS
	3.check print information in command and output window.
Manual operation did not work.	1.check axis parameter: AXISSTATUS, to view if any error happens, especially position limit or alarm.
	2.see ATYPE if relative axis is 0, if it is set as encoder.
	3.check axis UNITS value, if it is too small or just is 0.
	4.check motor wiring
Manual operation only works in one direction	1.check axis parameter: AXISSTATUS, to view if any error happens, especially position limit or alarm.
	2.check if the drive double pulse setting same as controller, view controller setting through INVERT_STEP。
	3.check motor wiring, especially direction wire.