

ZDevelop 使用手册

Version 3.10.04

版 权 说 明



本手册版权归深圳市正运动技术有限公司所有，未经正运动技术公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。

涉及 ZMC 运动控制器软件每个指令和函数的介绍和范例，请参阅正运动技术公司 ZBasic/ZPLC/ZHMI 编程手册。

本手册中的信息资料仅供参考。如涉及产品升级，内容需要更改，恕不另行通知！正运动技术公司保留对本资料的最终解释权！如需获取更多详情请登陆正运动技术公司网站。



调试机器要注意安全！请务必在机器中设计有效的安全保护装置，并在软件中加入出错处理程序，否则所造成的损失，正运动技术公司没有义务或责任对此负责。

目 录

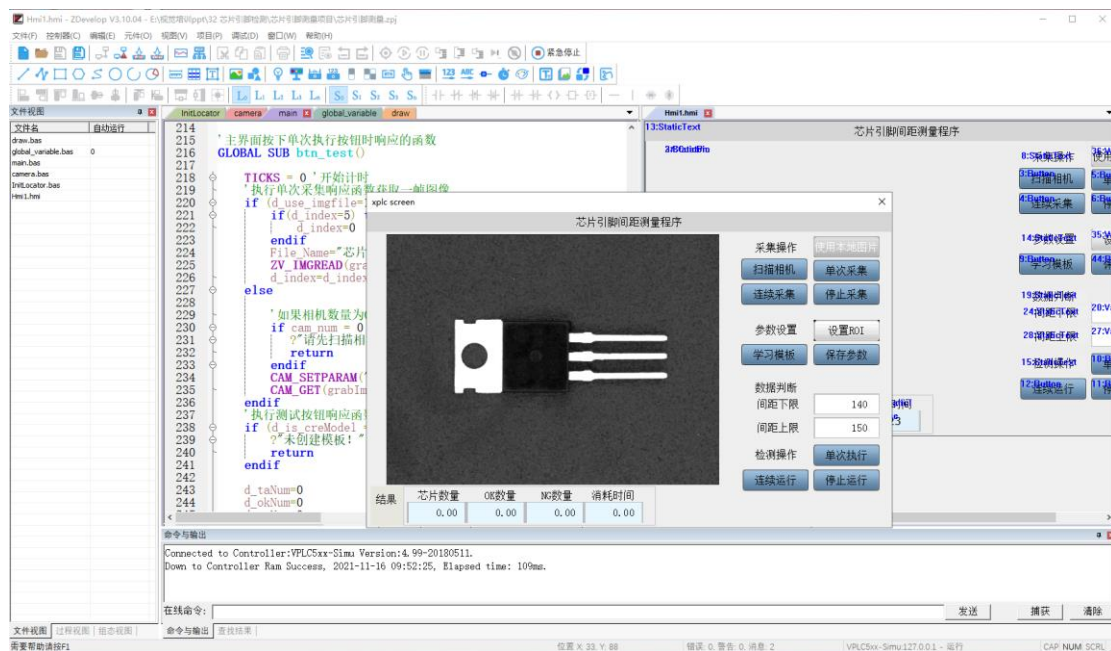
第一章 ZDevelop 简介	1
1.1 系统要求.....	1
1.2 ZDevelop 使用步骤	2
1.3 连接到运动控制器.....	6
1.4 连接到仿真器.....	10
1.5 使用帮助.....	11
1.6 语言切换.....	14
第二章 控制器操作.....	15
2.1 控制器状态.....	15
2.2 控制器复位.....	19
2.3 控制器锁定/解锁.....	19
2.4 在线命令/输出.....	20
2.5 固件升级.....	27
2.6 修改 IP 地址.....	29
2.7 系统时间设置.....	29
第三章 ZDevelop 项目	30
3.1 项目文件.....	30
3.2 项目设置.....	31
3.3 程序文件.....	31
3.4 ZAR 文件	31
3.5 库文件.....	31
第四章 程序编辑.....	31
4.1 快捷编辑.....	31
4.2 Basic 编辑	36
4.3 梯形图编辑.....	38
4.4 HMI 组态编辑	46
4.5 右键菜单.....	52
4.6 书签功能.....	54
4.7 文件视图.....	55
4.8 SUB 过程视图	55
4.9 组态视图.....	56
第五章 程序下载运行.....	56
5.1 程序下载 (RAM/ROM)	56
5.2 程序自动运行.....	57
5.3 ZAR 下载.....	57
5.4 编译 Lib.....	58
5.5 控制器程序比较.....	60
第六章 控制器窗口.....	60
6.1 文件/过程/组态窗口.....	60
6.2 命令与输出窗口.....	60
6.3 查找结果窗口.....	61
6.4 轴参数窗口.....	61
6.5 手动运动窗口.....	65

6.6 示波器窗口.....	65
6.7 输入 I/O 窗口.....	74
6.8 输出 I/O 窗口.....	76
6.9 任务/监视窗口.....	77
6.10 寄存器窗口.....	77
6.11 图像窗口.....	79
6.12 AD/DA 窗口.....	80
6.13 帮助窗口.....	81
6.14 注释窗口.....	81
6.15 交叉参数表.....	82
6.16 寄存器使用列表.....	83
6.17 字体视图.....	83
6.18 自定义视图.....	84
第七章 程序调试.....	85
7.1 进入调试.....	85
7.2 调试工具栏.....	88
7.3 任务状态窗口.....	88
7.4 监视窗口.....	89
7.5 断点.....	90
7.6 故障诊断.....	92
7.7 总线状态诊断.....	93
7.8 状态栏.....	94
第八章 常见问题.....	96

第一章 ZDevelop 简介

ZDevelop 是正运动推出的集成机器视觉+运动控制功能的一站式绿色免安装开发软件，支持 ZBasic、ZPLC 梯形图、ZHMI 组态、ZVision 机器视觉等二次开发，并可混合编程、实时仿真、在线跟踪以及诊断与调试，快速实现智能装备的视觉定位、测量、识别、检测和复杂的运动控制等系统的开发，ZDevelop 软件开发界面见下图所示。

用户可通过串口或网口连接 PC 与控制器，使用 ZDevelop 软件编写的程序可以直接下载到正运动控制器里脱机运行，也可以在 PC 平台仿真运行。



ZDevelop 软件支持三种编程方式：Basic、PLC 梯形图、HMI 组态。需要注意的是使用 Basic 语言编程开发时，可以多个 Basic 任务运行；如果使用 PLC 梯形图或 HMI 组态编程开发则只支持一个 PLC 任务或一个 HMI 任务运行。Basic 任务、PLC 梯形图任务和 HMI 组态任务之间可以多任务运行。

ZDevelop 软件支持在线仿真调试，自带仿真器 ZMC Simulator 和组态程序仿真工具 xplc screen。

1.1 系统要求

要使用 ZDevelop 软件，硬件要求如下：

项目	最小要求	推荐使用
----	------	------

CPU	Pentium 级别处理器，主频 450MHz	Pentium 级别处理器，主频 1GHz
内存大小	64MB	256MB
硬盘剩余空间	20MB	100MB
操作系统	Windows 98, Windows xp	Windows xp or win7
显示器	800x600/256 彩色	1024x768/24 位真彩色
通讯接口	RS232 串行口	RS232 串行口/USB/以太网口(可以通过 HUB 转接)

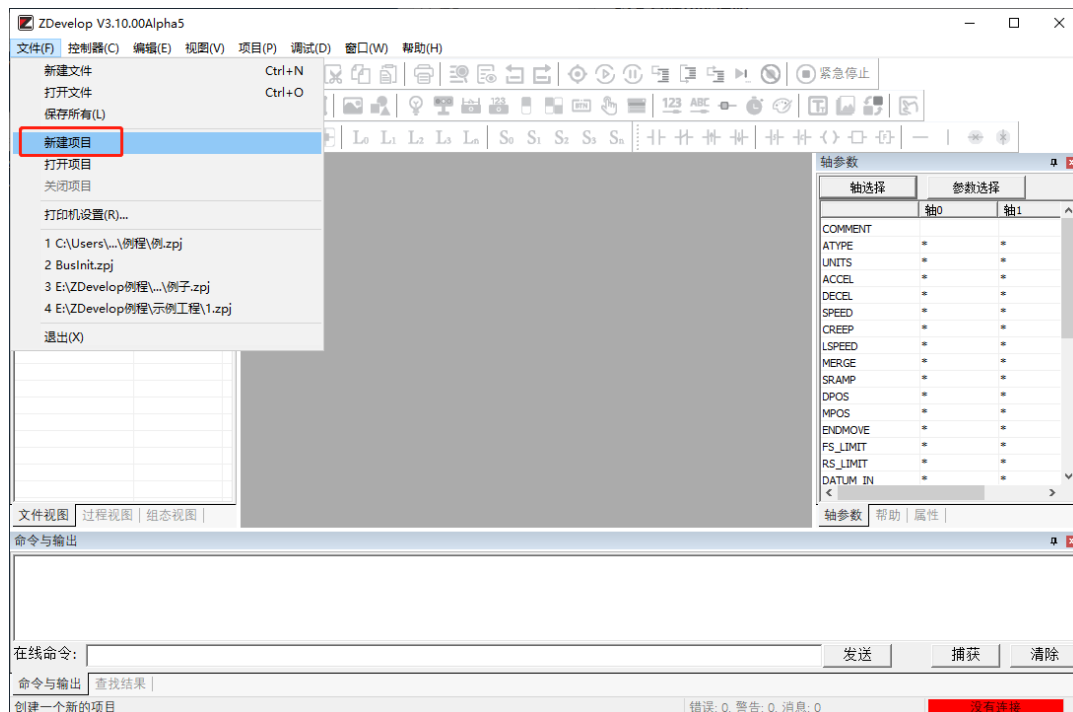
1.2 ZDevelop 使用步骤

在电脑里新建一个文件夹用来保存即将要建立的工程。打开 ZDevelop 编程软件，当前说明例程的 ZDevelop 软件版本为 V3.10，更新软件版本请前往正运动官方网站下载，网址：

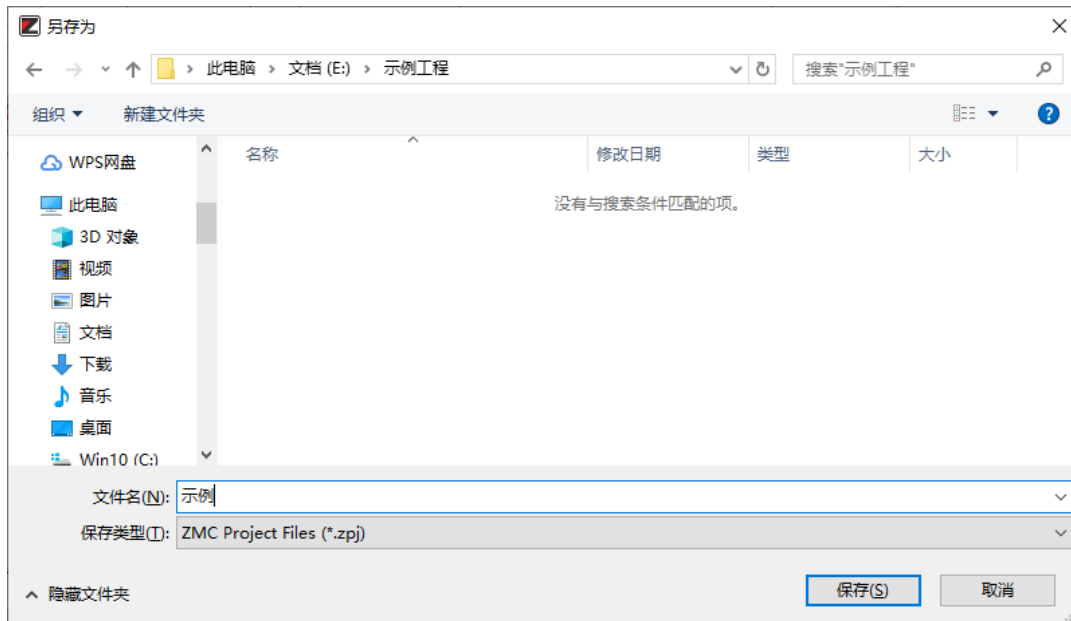
www.zmotion.com.cn

基本流程：新建项目→新建文件→文件添加自动运行任务号→编辑程序→连接控制器→下载程序到控制器运行。

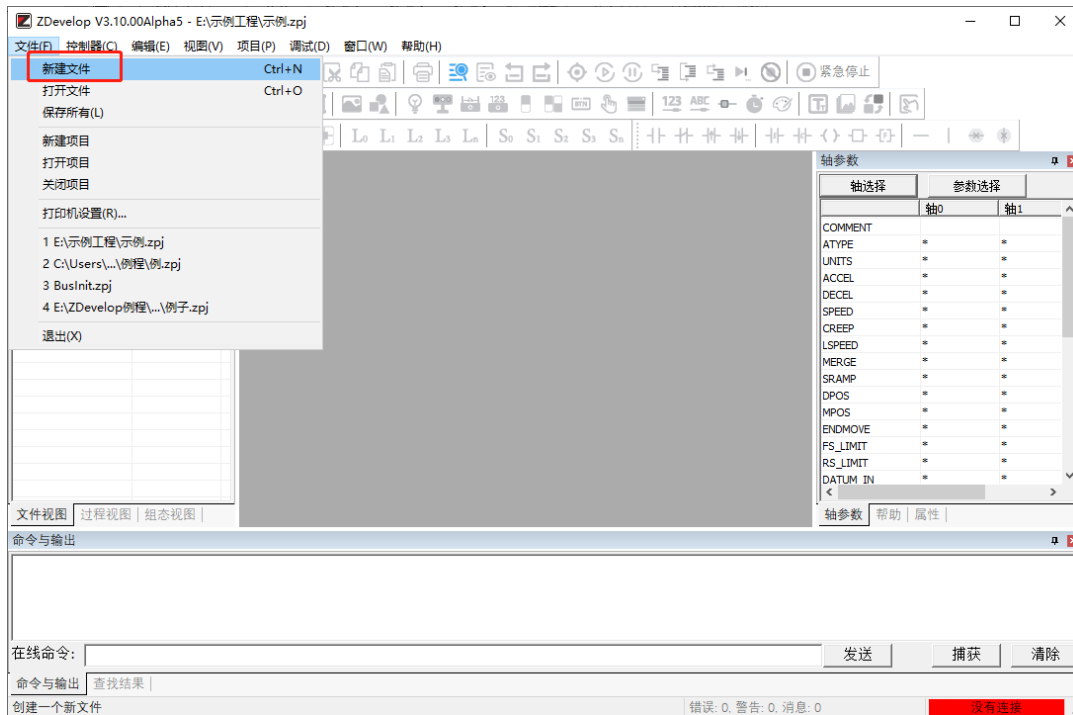
1. 新建项目：菜单栏“文件”→“新建项目”。



点击“新建项目”后弹出“另存为”界面，选择一个文件夹打开，输入文件名后保存项目，后缀为“.zpj”。



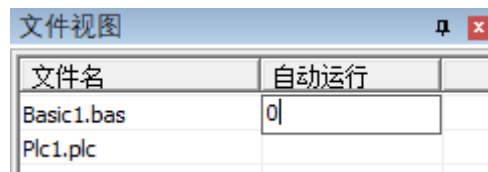
2. 新建文件：菜单栏“文件”→“新建文件”。



点击“新建文件”后，出现下图所示的弹窗，支持 Basic/PLC/Hmi 混合编程，这里选择新建的文件类型为 Basic 后确认。



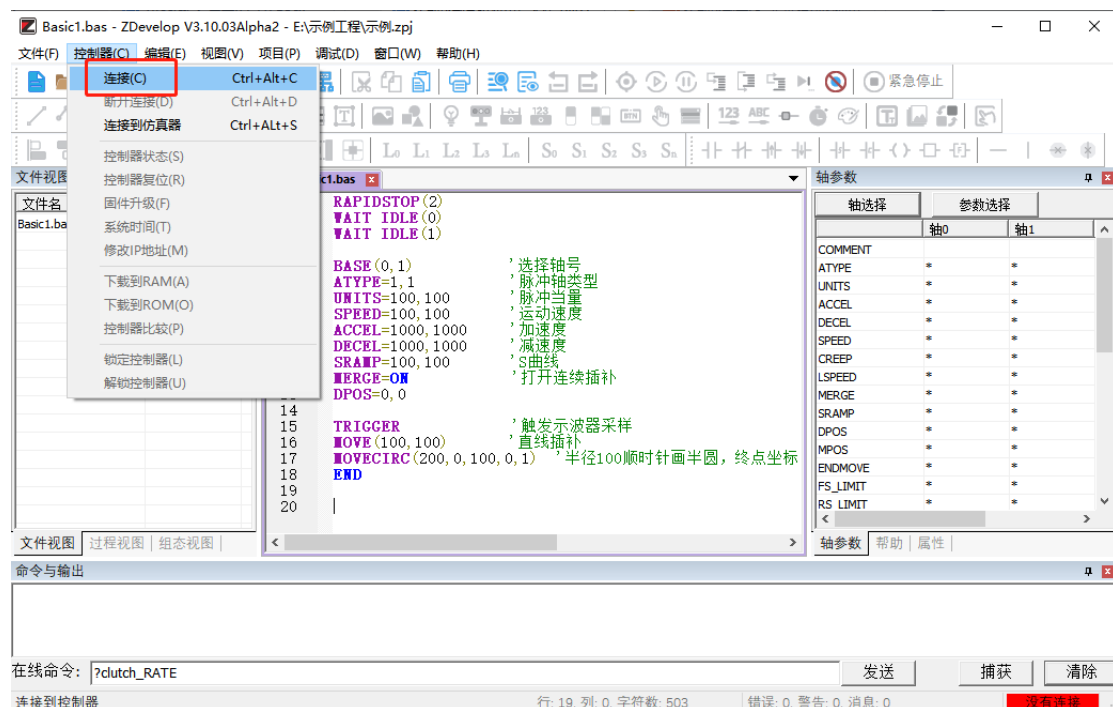
3. 设置文件自动运行：如下图，双击文件右边自动运行的位置，输入任务号“0”。



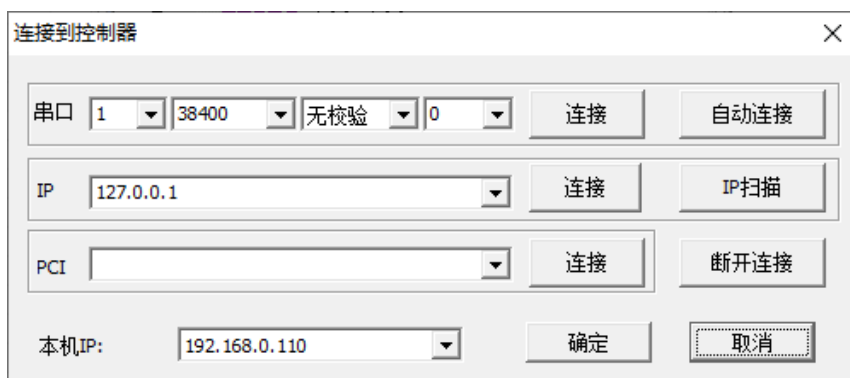
4. 编辑程序：程序编写完成，点击保存文件，新建的 Basic 文件会自动保存到项目 zpj 所在的文件夹下。

5. 连接到控制器：在程序输入窗口编辑好程序，点击“控制器”→“连接”。

没有控制器是可选择连接到仿真器仿真运行，点击“连接”→“连接到仿真器”，便可成功连接到仿真器，并弹出仿真器连接成功提示。



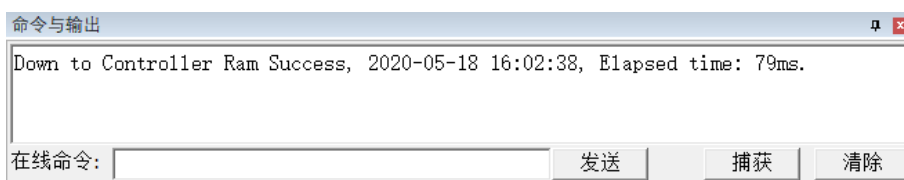
点击“连接”弹出“连接到控制器”窗口，可选择串口连接或网口连接，选择匹配的串口参数或网口 IP 地址后，点击连接即可。连接成功命令与输出窗口打印信息：Connected to Controller:ZMC432 Version:4.64-20170623.



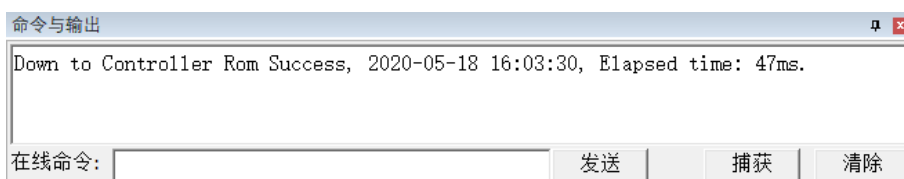
串口连接和网口连接的详细方法参见下节[连接到运动控制器](#)。

6. 下载程序：点击菜单栏按钮“下载到 RAM”或按钮“下载到 ROM”，下载成功命令和输出窗口会有提示，同时程序下载到控制器并自动运行。

成功下载到 RAM：



成功下载到 ROM：



RAM 下载掉电后程序不保存，ROM 下载掉电后程序保存。下载到 ROM 的程序下次连接上控制器之后程序会自动按照任务号运行。

注意事项：

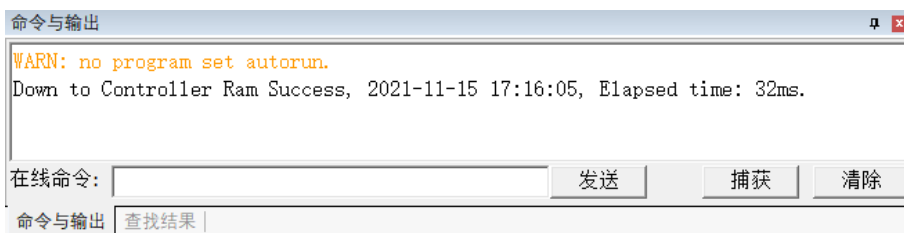
1. 打开工程项目时，选择打开项目 zpj 文件，若只打开其中的 Bas 文件，程序无法下载到控制器。

2. ZMCO 系列部分型号不支持下载到 RAM。

3. 不建立项目的时候，只有 Bas 文件无法下载到控制器。

4. 自动运行的数字 0 表示任务编号，以任务 0 运行程序，任务编号不具备优先级。

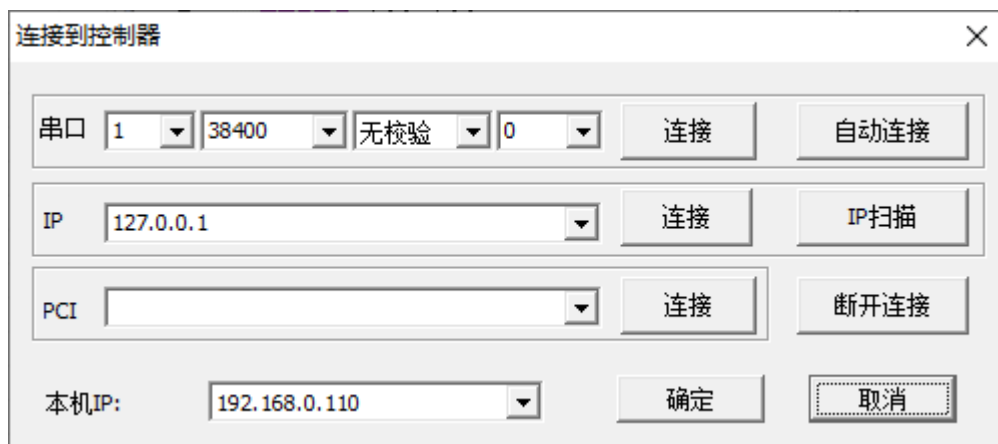
5. 若整个工程项目内的文件都不设置任务编号，下载到控制器时，系统提示如下信息
WARN: no program set autorun.



1.3 连接到运动控制器

通过“控制器”-“连接”菜单，可以连接到控制器。

ZDevelop 支持串口和以太网口连接到控制器。



串口参数：串口编号，波特率，是否校验，串口 ID（填 0 即可）。

IP：控制器 IP 地址。

串口连接

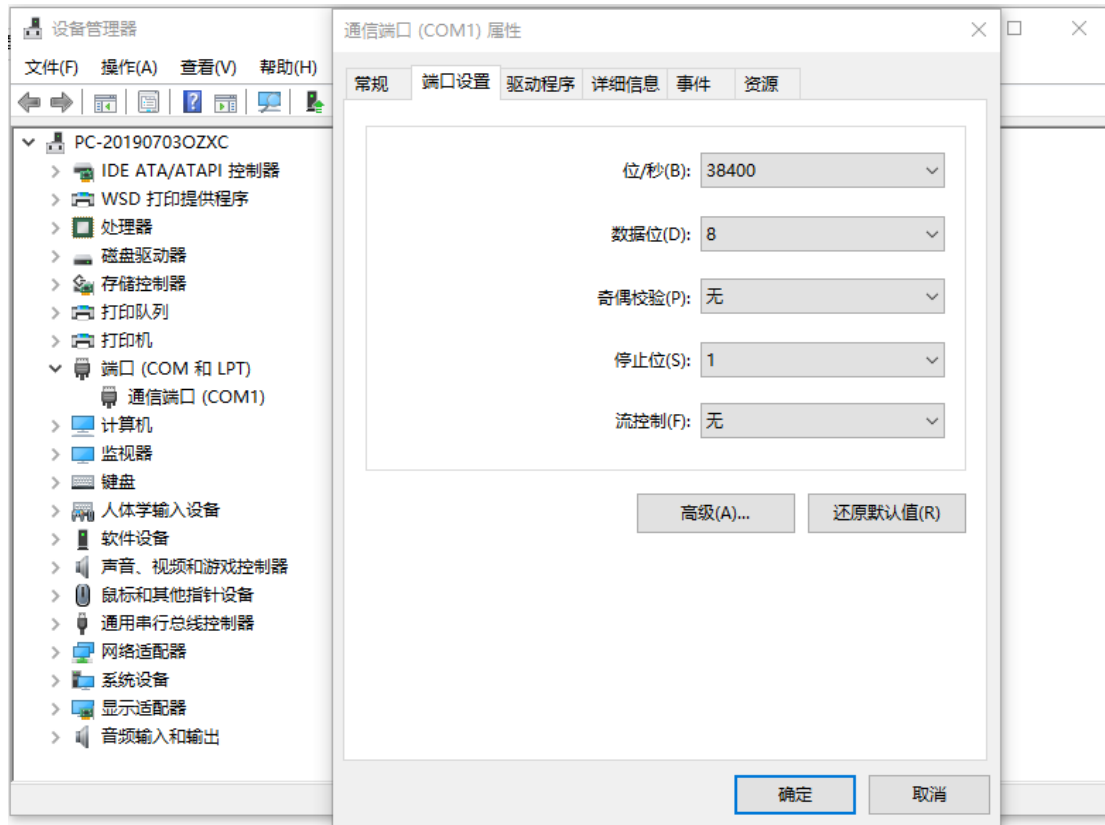
当串口列表下拉选择时，会自动列出本计算机上可用的串口号，选择需要连接的串口编号、设置波特率、校验位、停止位之后，点击连接，连接是否成功会在软件输出窗口自动打印出相应信息。

USB 连接会自动生成虚拟串口，选择串口号来连接即可。

若连接失败，按下面方法依次排查：

1. 查看串口连接线是否为交叉线。
2. “连接到控制器”里的 COM 口编号、参数是否选择正确。

打开电脑“设备管理器”-“端口”-“通信端口 (COM)”-“端口设置”，查看 COM 口设置是否正确，控制器串口默认参数 波特率 38400，数据位 8，停止位 1，校验位无。



在“端口设置”-“高级”选项中可更改 COM 端口号，通过下拉列表选择。



3. 当通过串口连接到控制器时，对应的控制器串口必须配置为 MODBUS 从协议模式（缺省模式），断电重启即可恢复。

4. COM 口是否已被其他程序占用，如串口调试助手等。
5. PC 端是否有足够的串口硬件。
6. 更换串口线/电脑测试。

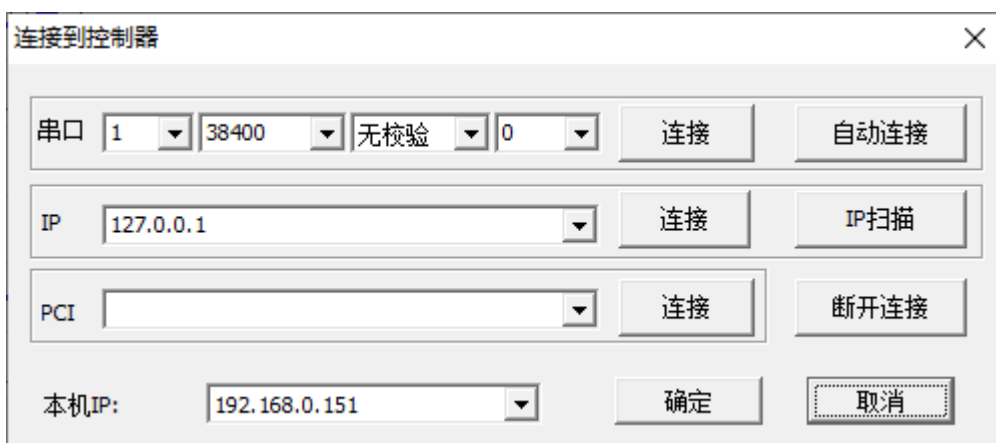
网口连接

IP 地址列表下拉选择时，会自动查找当前局域网可用的控制器 IP 地址。

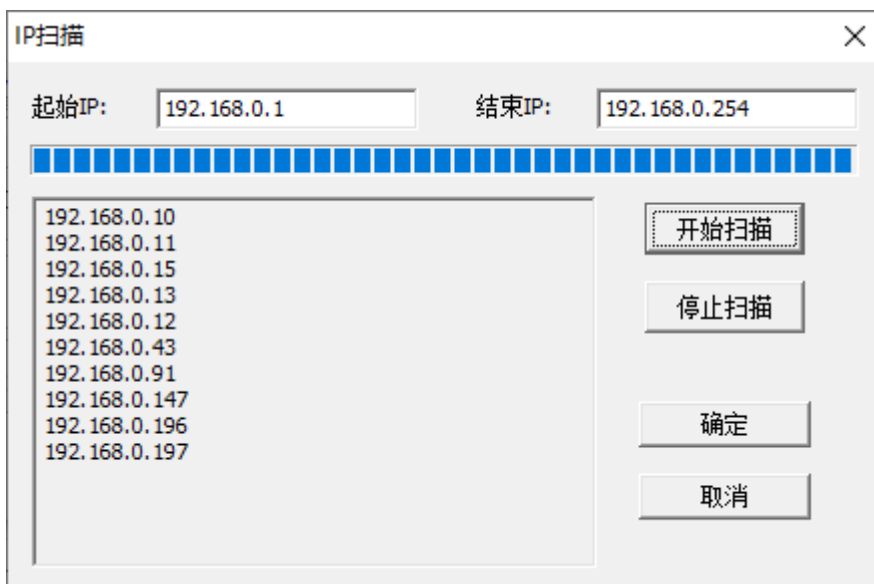
控制器出厂的缺省 IP 地址为 192.168.0.11，“连接到控制器”窗口能显示出本机 IP 地址，电脑需要设置 IP 地址与控制器 IP 处于同一网段才能连接，即四段的前三段要相同，最后一段不同才可通讯。

其中 VPLC5 系列控制器包含两个网口，LAN1 网口默认地址 192.168.0.11，LAN2 网口默认地址 192.168.1.11。

如下图通过“连接到控制器”窗口，可以快速查看本机 IP，请注意设置有线网卡与无线网卡各自的 IP。



同一个网络有多个控制器的时候，可以采取 IP 扫描来查看。



PC 端 IP 地址修改方法：

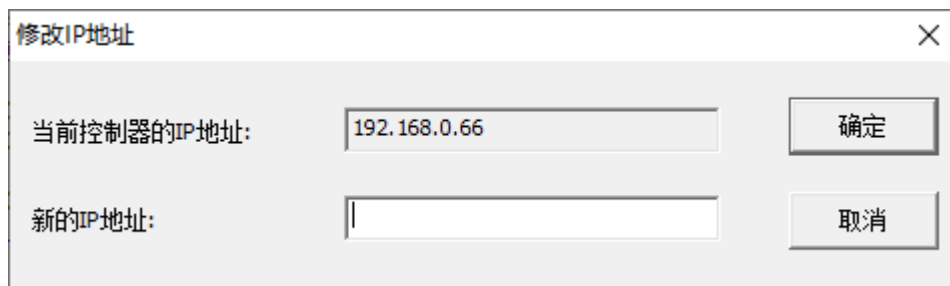
查看电脑本地 IP 协议版本 4 地址是否为 192.168.0.xxx，前三段与控制器一致，最后一段不能一样，控制器出厂默认 IP 192.168.0.11。如果 IP 地址的第三段不一样，则需要把对应的子网掩码改为 0。设置好之后再软件连接。



控制器 IP 地址修改方法：

如果控制器 IP 被修改，不处于 192.168.0.XXX 这个网段，此时只能先通过串口连接控制器，获取控制器 IP 地址，然后修改本机 IP 或控制器 IP 使二者处于同一网段。

修改控制器 IP 地址方法有多种，可点击菜单栏“控制器”-“修改 IP 地址”，弹出如下窗口，此时会显示当前控制器 IP，在窗口可直接输入新的 IP 地址。



或在 ZDevelop 菜单栏“控制器”-“控制器状态”查看或在线命令获取控制器 IP 地址，控制器 IP 用指令 IP_ADDRESS 修改。



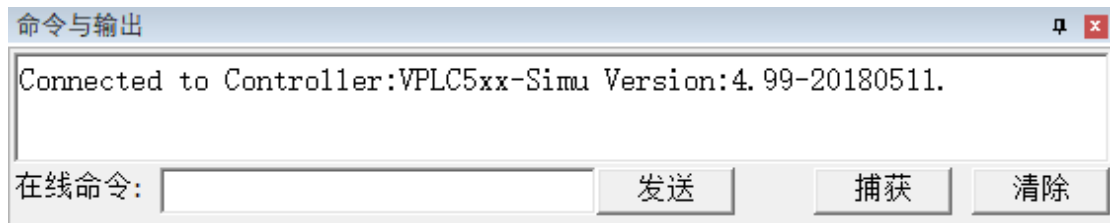
修改 IP 后，控制器与 ZDevelop 的连接会断开，此时再次选择新设置的 IP 地址连接即可。

1.4 连接到仿真器

ZDevelop 支持离线仿真，在无控制器情况下可以提前仿真与验证程序。

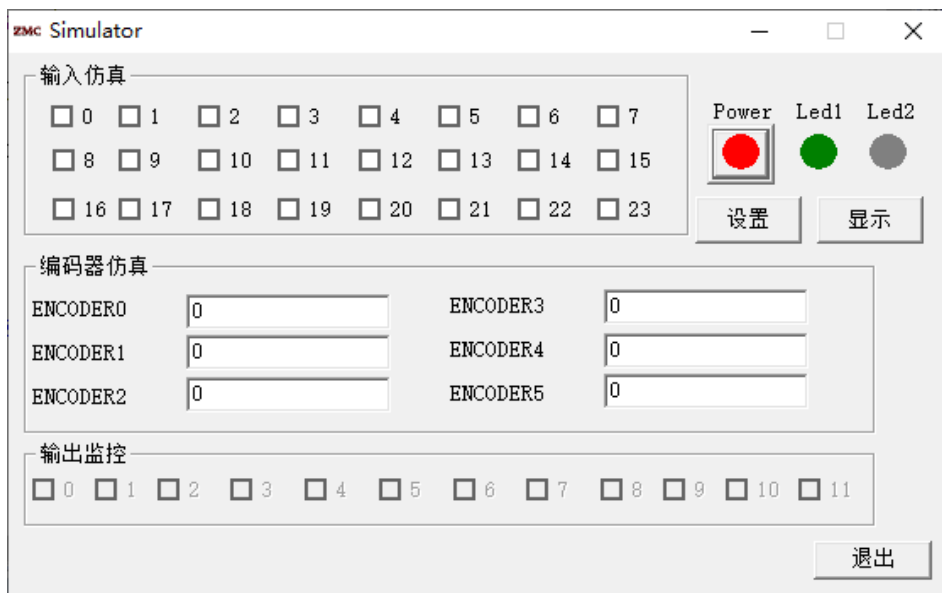
通过“控制器”-“连接到仿真器”菜单可以自动启动仿真器并连接，或当仿真器启动后可以通过 IP 地址“127.0.0.1”来连接。

成功连接到仿真器，打印如下提示信息：



注意：较早版本的仿真器不支持仿真视觉指令，请更新软件到最新版。

当程序包含 HMI 工程时，点击“显示”来实现 HMI 界面仿真。

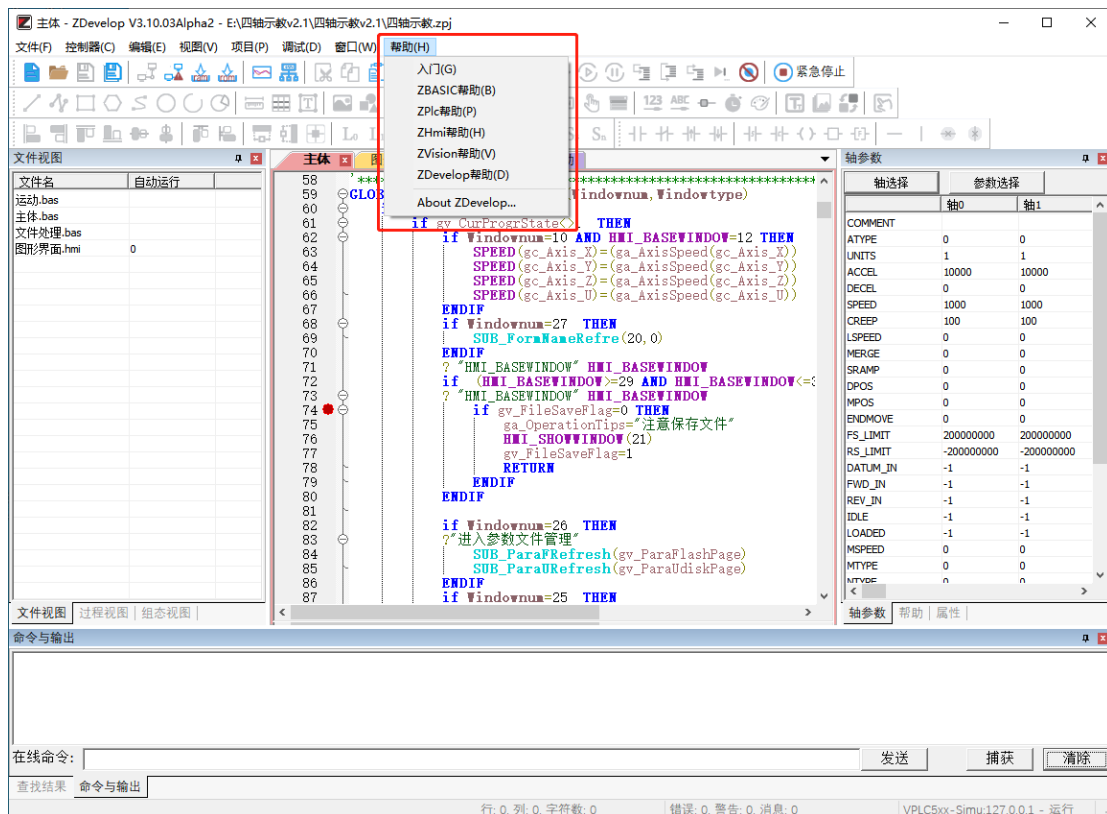


某个 HMI 示例工程的界面显示如下。



1.5 使用帮助

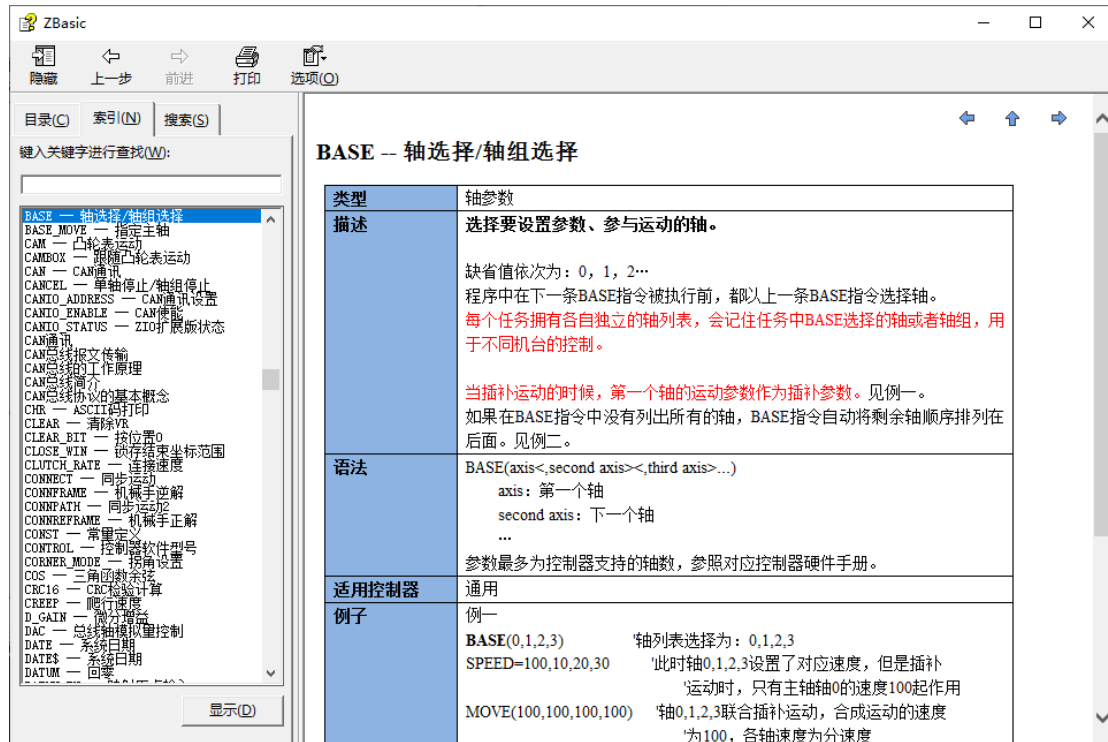
点击菜单栏”帮助”，可以打开控制器使用入门文档、ZBasic 语法帮助文档、ZPlc 语法帮助文档、ZHmi 语法帮助文档、ZVision 视觉语法帮助文档、ZDevelop 开发环境帮助文档。文档内可查看所有相关指令的说明，与部分功能的介绍。



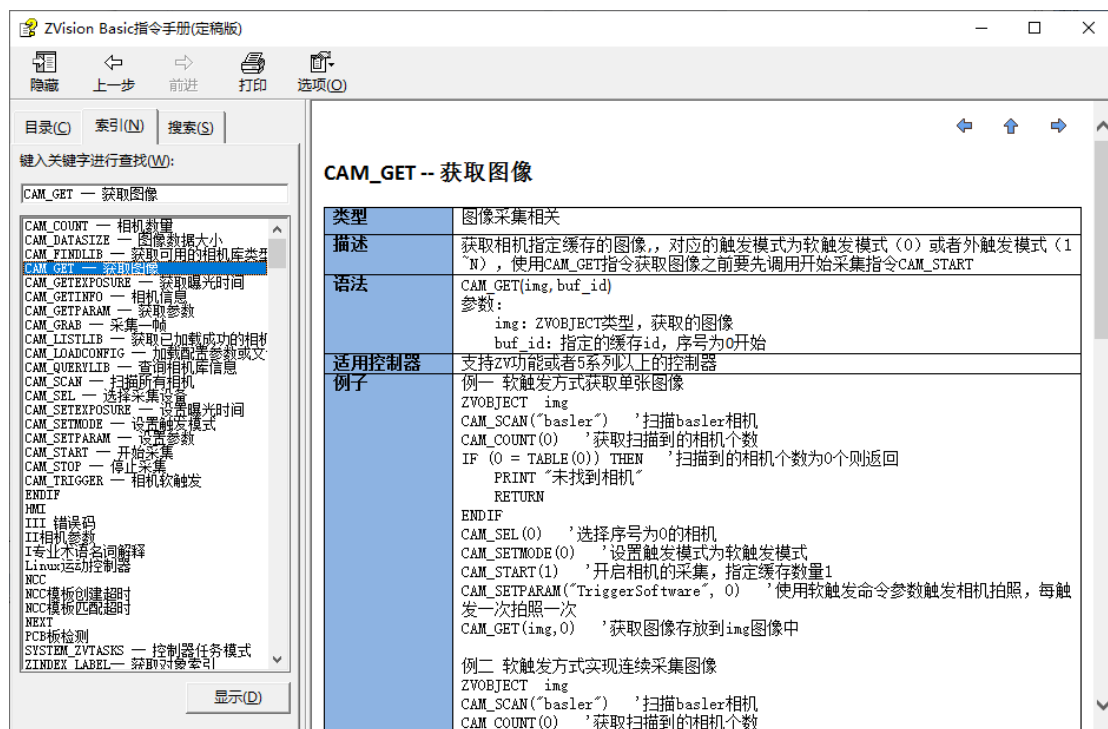
在 Basic 程序中，选中指令再按下 F1 键，可快速在帮助视图窗口打开帮助文档中的该指令详细说明页面，如下图。在 PLC 和 HMI 编程下也支持 F1 快捷键。

在帮助界面内，按住 Ctrl，滚动鼠标可对帮助文档显示内容进行放大缩小。HOME 键跳到当前行的行首，END 键跳到当前行的行尾，pageup 和 pagedown 为上下翻页。

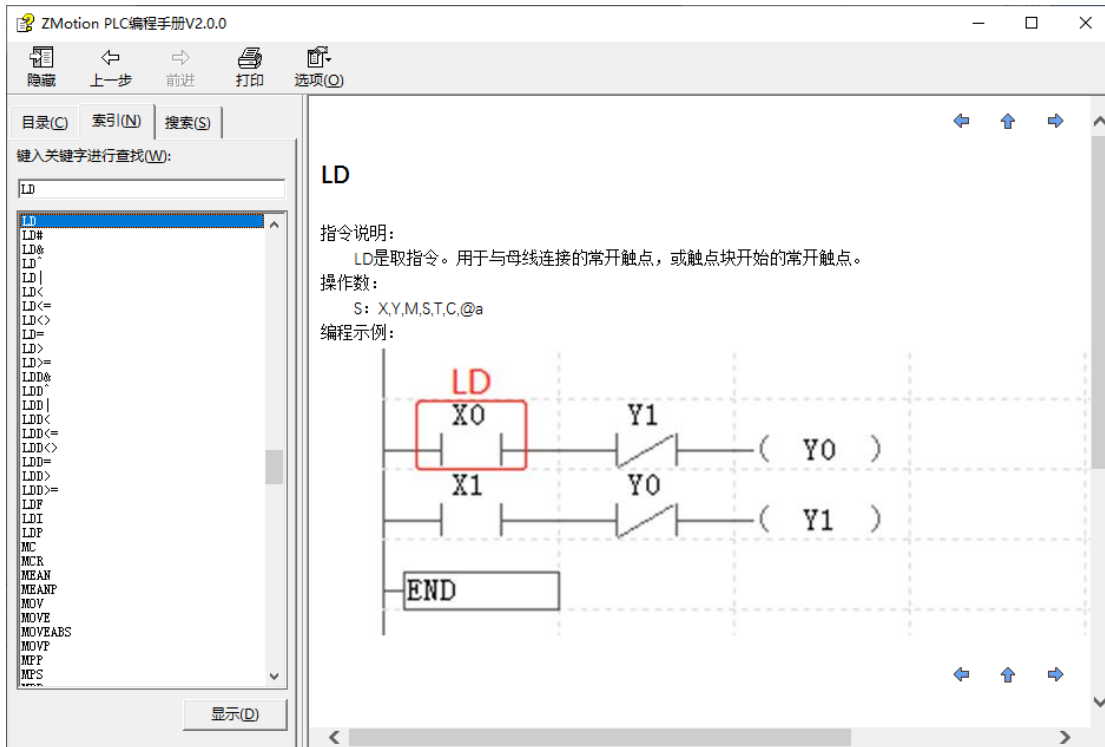
ZBasic 帮助：



ZVision 帮助：



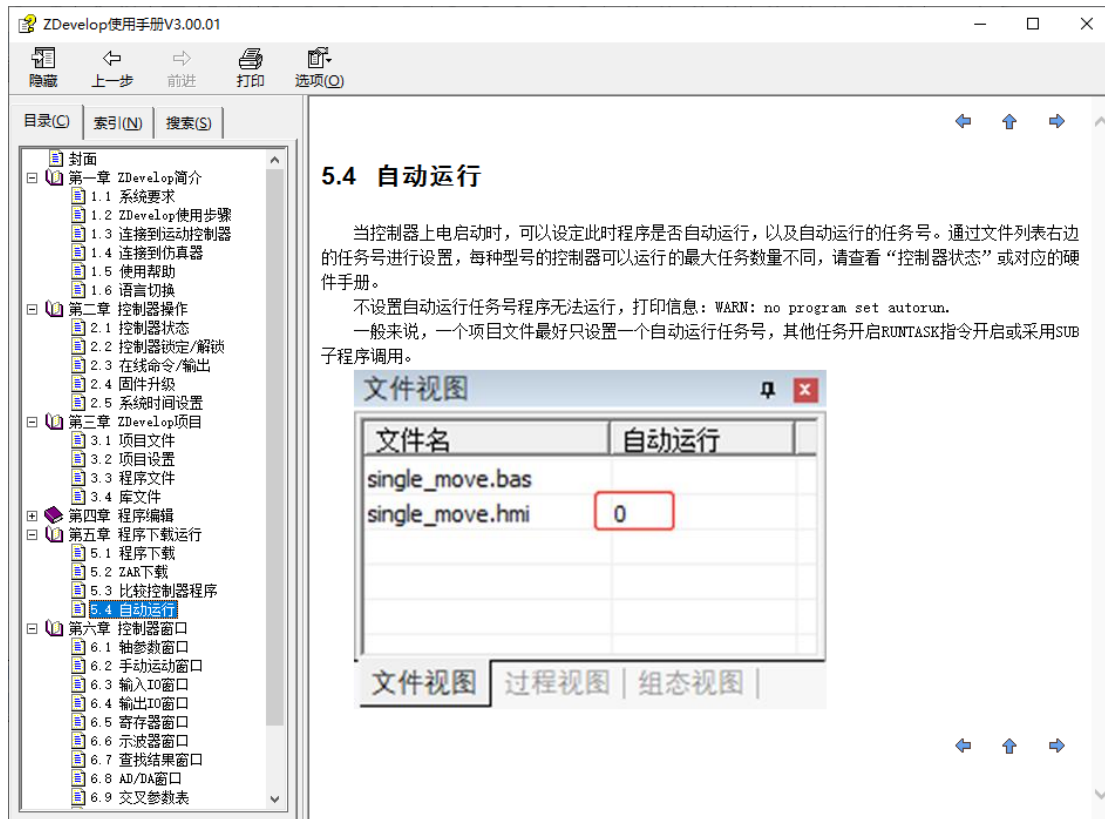
ZPLC 帮助:



ZHMI 帮助:

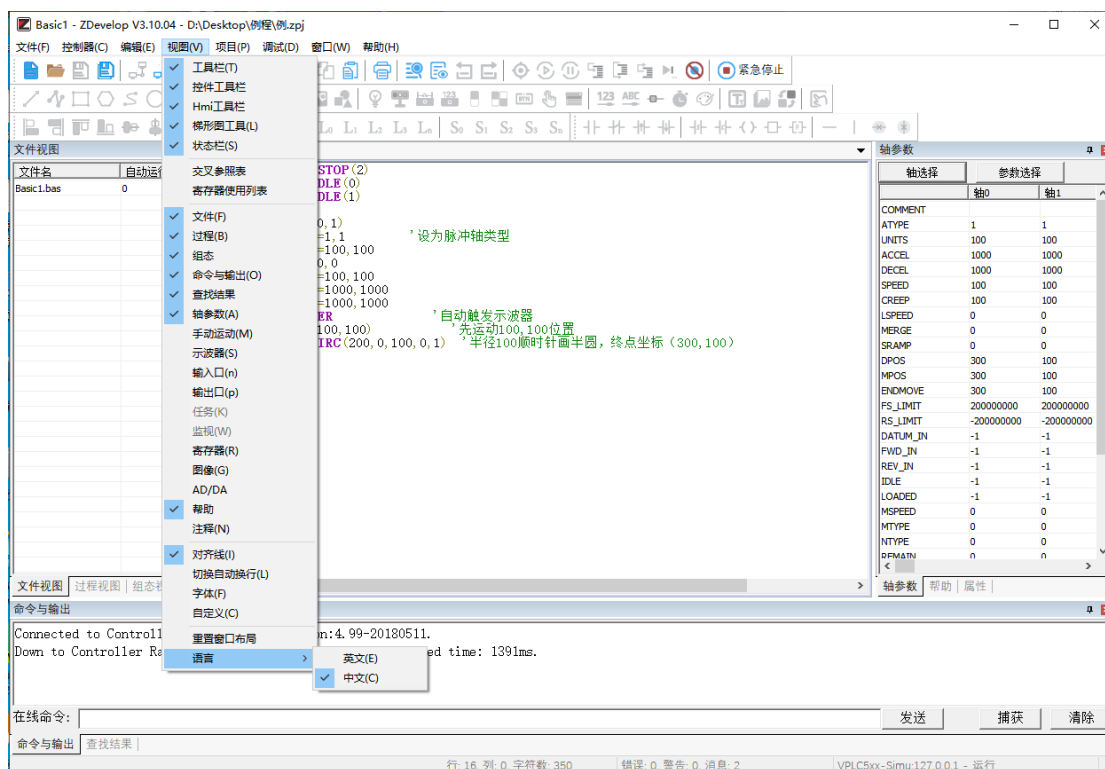


ZDevelop 帮助:

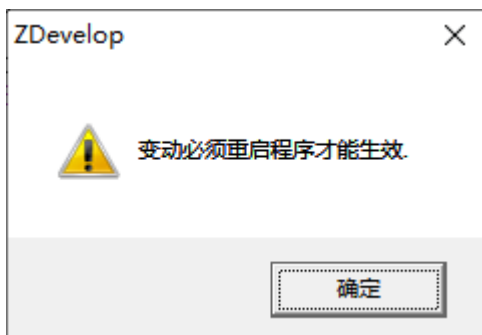


1.6 语言切换

当前 ZDevelop 版本支持中文和英文。通过“视图”-“语言”选择，选择后需要重启 ZDevelop 完成更换。



选择语言后点击“确定”，再重新启动 ZDevelop，语言切换成功。



第二章 控制器操作

ZDevelop 支持查看控制器状态和配置控制器。

控制器程序文件比较请参见“[程序下载运行](#)”章节。

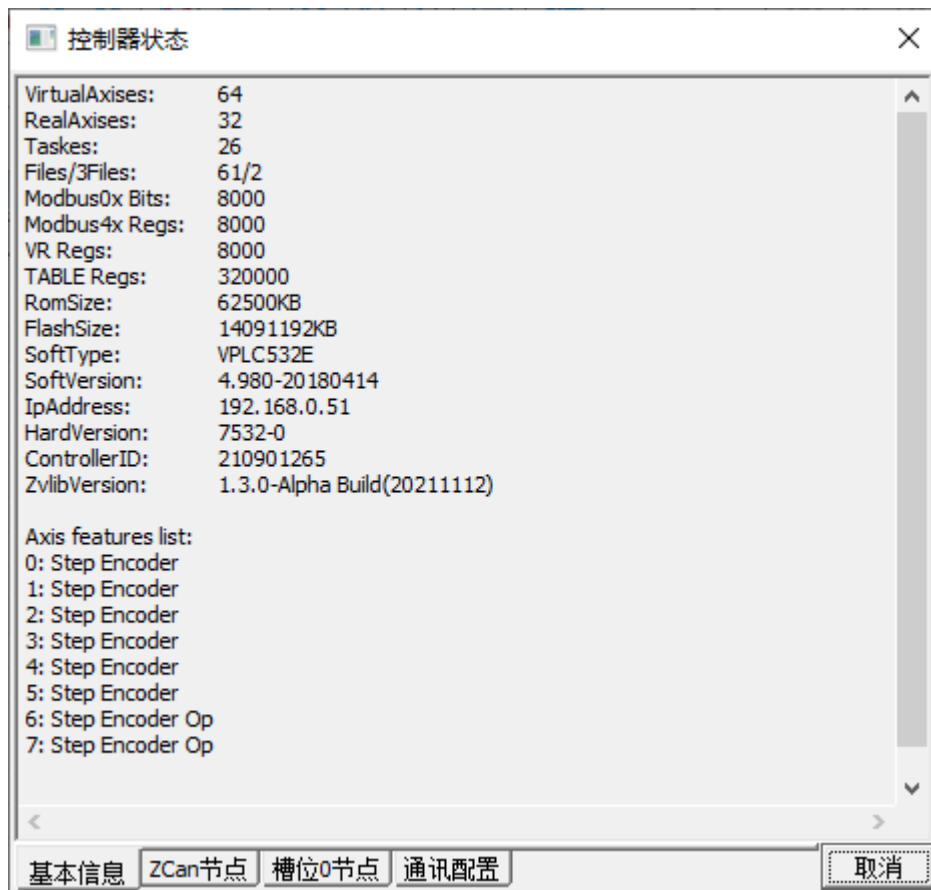
2.1 控制器状态

连接好控制器或仿真器后，通过“控制器”-“控制器状态”查看当前连接的控制器状

态。

“控制器状态”能显示出控制器状态信息，包括控制器基本信息、ZCan 节点状态、槽位节点状态、通讯配置。节点状态能显示连接的设备的轴数、起始 I/O 编号等信息。

基本信息包含最大虚轴数、最大电机轴数、任务数、文件数量、各类寄存器空间大小、程序容量大小、存储器大小、控制器的型号、软件版本号与时间、IP 地址、硬件版本号，控制器硬件 ID 以及各轴上可配置的类型及映射等。



控制器状态基本信息:

VirtualAxes: 支持最大虚拟轴数

RealAxes: 支持最大电机轴数

Tasks: 最大任务数

Files/3Files: 最大文件/三次文件数

Modbus0x Bits: Modbus 位寄存器用户可用空间大小

Modbus4x Regs: Modbus 字寄存器用户可用空间大小

VR Regs: VR 寄存器用户可用空间大小

TABLE Regs: TABLE 数组用户可用空间大小

RomSize: Rom 容量

FlashSize: Flash 容量

SoftType: 软件型号

SoftVersion: 系统软件版本+固件版本

IpAddress: 控制器 IP 地址

HardVersion: 硬件版本

ControllerID: 控制器唯一 ID

ZvlibVersion: 视觉库文件版本

Axis features list: 轴类型列表

ZCan 节点信息:

CanID	硬件ID	轴数	输入	输出	AD	DA	
Local	7516-0(VPLC516E)	16	28(0-27)	16(0-15)	0	0	
2	10(ZAIO0802)	0	0	0	8(24-31)	2(12-13)	
4	4016(ECI0016PA)	0	8(80-87)	8(80-87)	12(40-51)	2(20-21)	

第一行 LOCAL 节点，显示硬件为 VPLC516E 控制器，第一行对应控制器的本地资源，支持 16 轴，数字量输入一共 28 个，编号 IN(0)–IN(27)，数字量输出一共 16 个，编号 OP(0)–OP(15)，没有 AD 和 DA。

第二至 N 行显示 ZCan 节点的扩展模块的信息。

例如，CanID 节点 2 的硬件为 ZAIO0802，无数字量输入，无数字量输出，AD 有 8 个，编号 AIN(24)–AIN(31)，DA 有 2 个，编号 AOUT(12)–AOUT(13)。

CanID 节点 4 的硬件为 ECI0016PA，数字量输入 8 个，编号 IN(80)–IN(87)，数字量输出 8 个，编号 OP(80)–OP(87)，AD 有 12 个，编号 AIN(40)–AIN(51)，DA 有 2 个，编号 AOUT(20)–AOUT(21)。

槽位 0 节点信息:

如果只有 EtherCAT 总线，槽位 0 是 EtherCAT 总线。

如果只有 RTEX 总线，槽位 0 是 RTEX 总线。

如果是 EtherCAT+RTEX 双总线，则槽位 0 是 EtherCAT 总线，槽位 1 是 RTEX 总线。



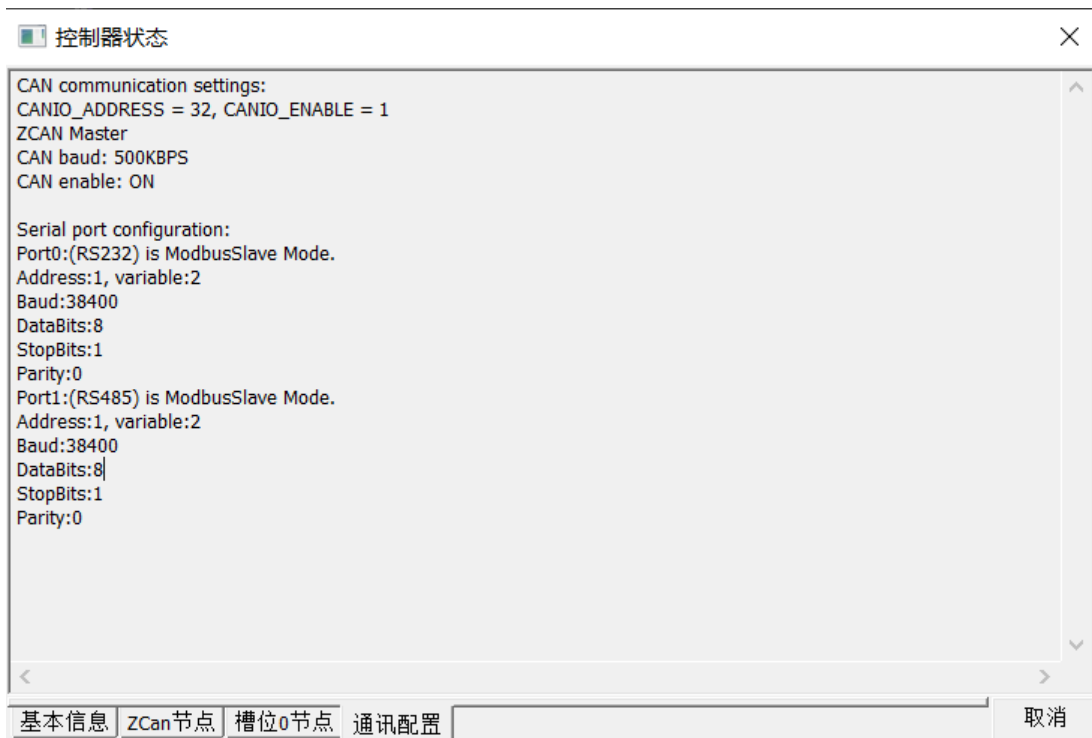
左边是一个最基本的 EtherCAT 初始化程序。

右边显示的第 1 至 N 行包含的是 EtherCAT 节点的扩展模块的信息。

如上图，EtherCAT 节点 0 的硬件厂家 ID 为 41bh(正运动技术的厂商 ID)，硬件厂家设备 ID 为 0h(正运动技术的产品 ID0，对应产品为 EI01616)，拨码号 0，无轴数扩展，数字量输入 16 个，编号 IN(96)-IN(111)，数字量输出 16 个，编号 OP(96)-OP(111)，无 AD，无 DA。

控制器通讯配置：

CAN 信息与 RS232/RS485/RS422 串口信息。



通讯设置显示内容如下：

此时 CAN 通讯的设置：CANIO_ADDRESS = 32，CANIO_ENABLE=1

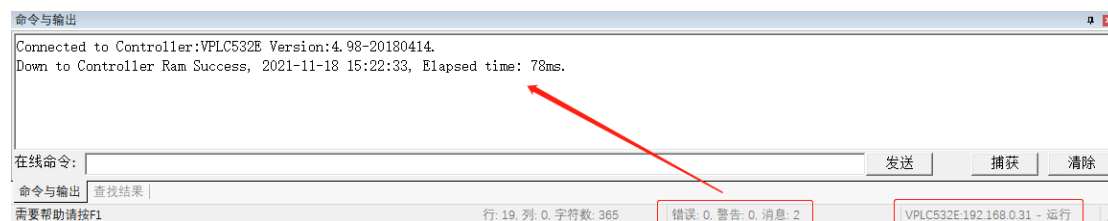
结合 CANIO_ADDRESS 与 CANIO_ENABLE 的信息可以知晓，此时控制器是 ZCAN Master 主站模式，CAN 总线通讯速率 500kbps，CAN 使能状态。

如需修改 CAN 通讯的设置，修改 CANIO_ADDRESS 与 CANIO_ENABLE 相关参数即可。

Port0 为 RS232，ModbusSlave 状态，地址 1，VR 与 MODBUS 寄存器是两片独立区间。

Port1 为 RS485, ModbusSlave 状态, 地址 1, VR 与 MODBUS 寄存器是两片独立区间。
更详细的解释请查看 SETCOM 指令相关的参数说明。

连接上控制器后通过下方状态栏打印信息查看当前连接的控制器型号、IP 地址、运行状态。也可通过“命令与输出”窗口查看错误、警告、消息的具体内容。



2.2 控制器复位

连接控制器后, 点击菜单栏“控制器”-“控制器复位”打开如下窗口, 点击确定之后控制器会重新上电启动一次, 需要再次连接控制器。



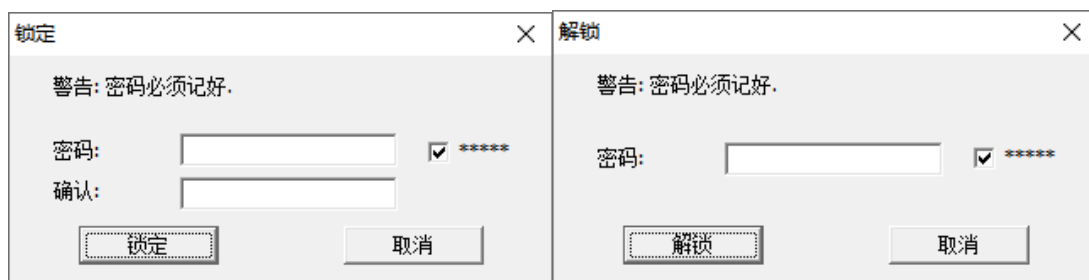
2.3 控制器锁定/解锁

控制器 LOCK 后将不能进行调试。控制器锁定用来保护控制器内下载的程序, 锁定之后上位机程序将无法下载到控制器, 但生成的 ZAR 文件仍可下载。密码可设置为数字、字母和特殊符号, 长度不超过 16 个字符。

密码采用不可逆算法加密, 一旦忘记, 将无法知晓。

解锁控制器输入密码即可解锁。

操作路径: 菜单栏“控制器”-“锁定控制器”/“解锁控制器”。



2.4 在线命令/输出

在线命令与输出窗口可以查询与输出控制器的各种参数、控制轴运动、打印程序运行结果、打印程序错误信息，软件开发人员在程序中给出的打印输出函数（由?、PRINT、WARN、ERROR、TRACE 等命令输出）。

注意问号使用英文符号，中文符号输入无效。ERRSWITCH 为 TRACE、WARN、ERROR 指令的控制开关，不同的参数值对应不同的输出效果：

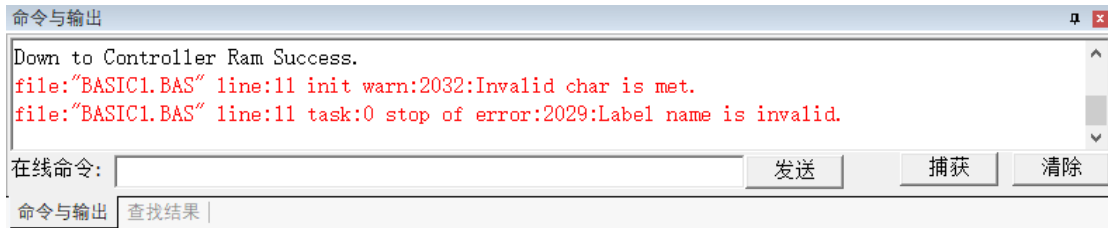
- 0: TRACE、WARN、ERROR 指令全部不输出
- 1: 只输出 ERROR 指令
- 2: 输出 WARN、ERROR 指令
- 3: TRACE、WARN、ERROR 指令全部输出

在线命令与输出窗口如下所示，“>>”表示 ZDevelop 在线命令输入的指令，在线命令输入“print 1+2”窗口会打印计算结果。

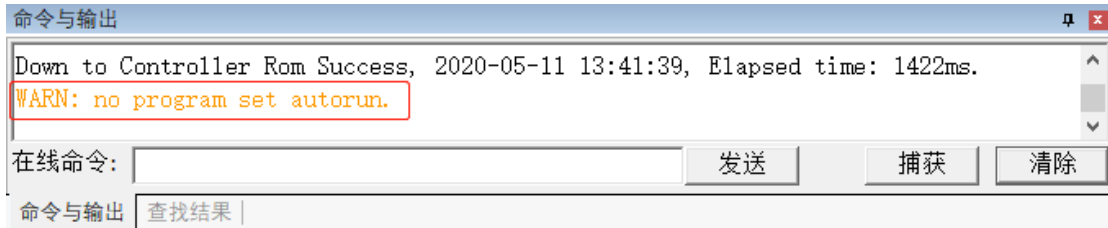
连接了控制器或仿真器就可以使用此功能，不受程序运行状态的限制。“清除”按钮用以清空“命令与输出”窗口所有内容。“捕获”开启后弹出“另存为”窗口，将接下来命令与输出窗口输出的所有内容保存到 PC 上，捕获开启后，原“捕获”按钮变为“捕获中”，直到再次按下“捕获中”停止捕获信息。



ERROR 错误信息以红色字体提示，如下图。

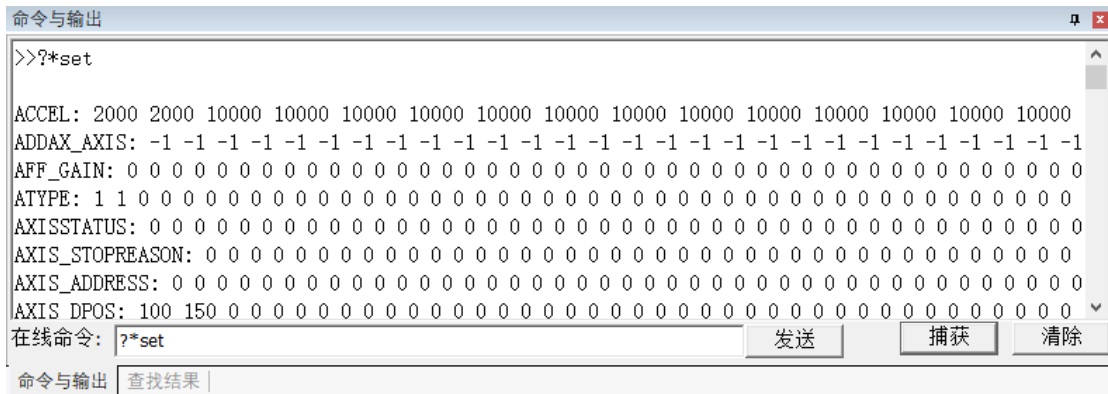


WARN 警告信息以橙色字体提示，如下图所示：



常用的打印查看命令有：

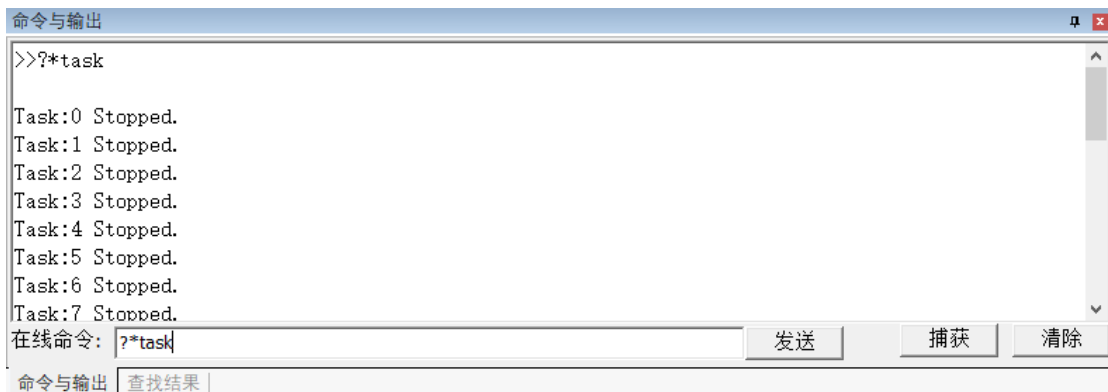
?*SET：打印所有轴参数值与系统参数值



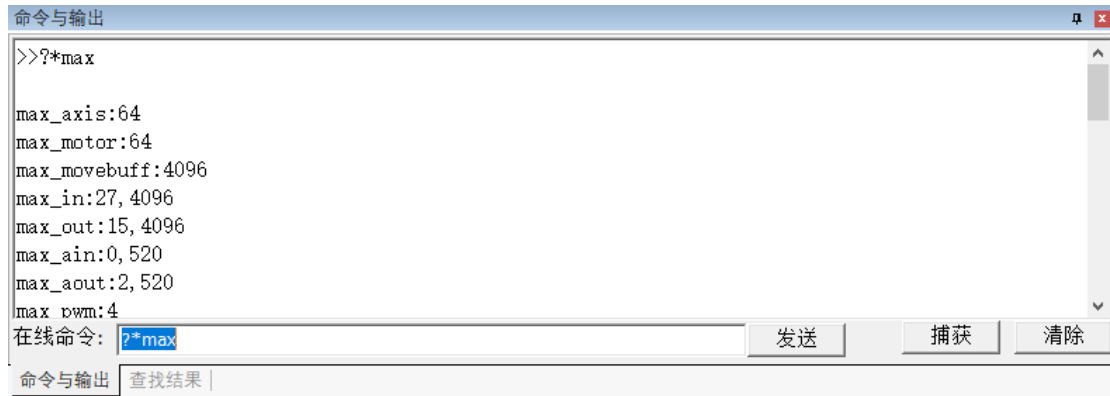
?*TASK：打印任务信息

任务正常时只打印任务状态

任务出错时还会打印出错误任务号，具体错误行



?*MAX：打印所有规格参数

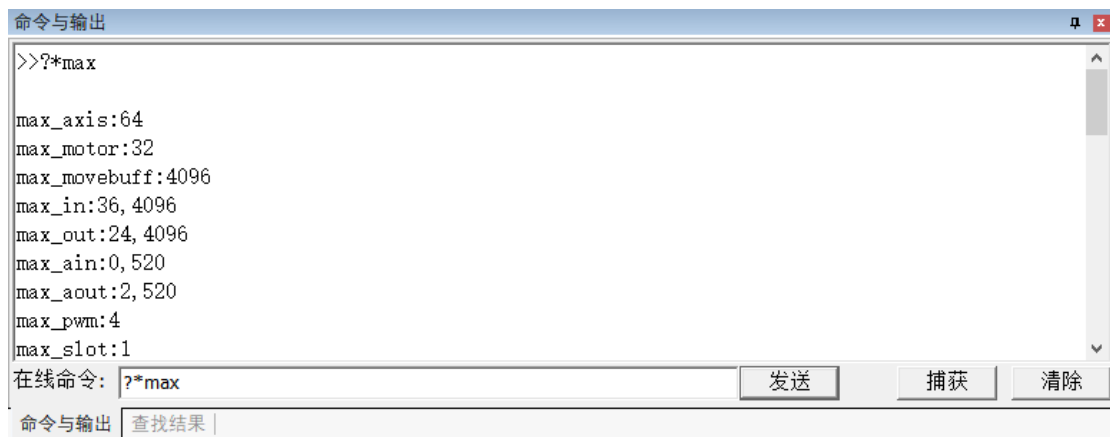


```

命令与输出
>>?*max

max_axis:64
max_motor:64
max_movebuff:4096
max_in:27, 4096
max_out:15, 4096
max_ain:0, 520
max_aout:2, 520
max_pwm:4
在线命令: ?*max  发送  捕获  清除
命令与输出  查找结果

```



```

命令与输出
>>?*max

max_axis:64
max_motor:32
max_movebuff:4096
max_in:36, 4096
max_out:24, 4096
max_ain:0, 520
max_aout:2, 520
max_pwm:4
max_slot:1
在线命令: ?*max  发送  捕获  清除
命令与输出  查找结果

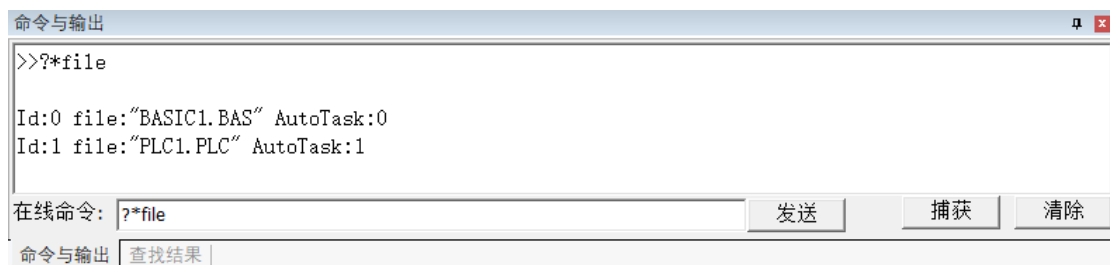
```

- max_axis:64 所有轴的最大轴数
- max_motor:32 可控的最大电机轴数
- max_movebuff:4096 每个轴或者轴组的最大运动缓冲
- max_in:27, 4096 控制器自带 IN 输入个数，最多支持 IN 输入个数
- max_out:15, 4096 控制器自带 OUT 输出个数，最多支持 OUT 输出个数
- max_ain:0, 520 控制器自带模拟量输入个数，最多支持模拟量输入个数
- max_aout:2, 520 控制器自带模拟量输出个数，最多支持模拟量输出个数
- max_pwm:4 PWM 输出个数
- max_slot:1 总线个数
- max_comport:2 串口个数
- max_ethport:8 与 PC、API 函数的网口通讯连接
- max_ethcustom:6 自定义网口通讯的连接
- max_ethiport:1 正运动控制器互联互通的网口通讯连接
- max_flashnum:9999 FLASH 块数
- max_flashsize:20480 每个 FLASH 空间大小
- max_nand:14091192KB NandFlash 存储总共的数量空间
- max_nandremain:8357548KB NandFlash 存储剩余可用的数量空间
- max_hwout:4, 100 硬件比较输出口

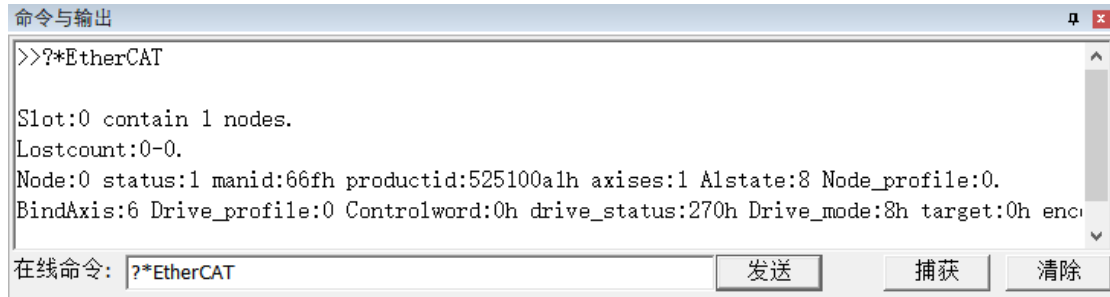
max_pswitch:64 软件位置比较输出的最多个数
 max_file:61 系统最多支持的文件数
 max_3file:2 系统最多支持的三次文件数
 max_task:26 任务数
 max_timer:1024 定时器个数
 max_loopnest:8 内部循环或者选择的次数
 max_callstack:10 子程序调用的堆栈层数
 max_local of one sub:32 SUB 的局部变量数
 max_vr:8000 VR 寄存器空间个数
 max_table:320000 TABLE 数组空间个数
 max_modbusbit:8000 MODBUS_BIT 位寄存器空间大小
 max_modbusreg:8000 MODBUS_REG 字寄存器空间大小
 max_var:20480 最多支持变量个数(含全局变量与文件变量)
 max_array:4096 最多支持数组个数(含全局数组与文件数组)
 max_arrayspace:2560000 所有数组总共的空间大小
 max_sub:2048 最多支持 SUB 子程序的个数
 max_edgescan:1024 最多可支持的上升沿/下降沿扫描个数
 max_lablelength:25 数组与变量等自定义字符的最大长度
 max_hmi:2, x:1920 y:1080 支持 2 个远端 HMI, 最大尺寸为 1920*1080
 max_zvlatch:4 视觉图像锁存通道个数
 max_zvtask:3 视觉任务数
 SERVO_PERIOD:1000 min:500 max:4000 系统周期, 最小 500 微秒, 最大 4 毫秒
 function support:Coder Cam MultiMove Circ Merge Frame NcGcode Zvision 支持

的功能

?*FILE: 打印程序文件信息



?*SETCOM: 打印当前串口的配置信息



```
命令与输出
>>?*EtherCAT

Slot:0 contain 1 nodes.
Lostcount:0-0.
Node:0 status:1 manid:66fh productid:525100alh axes:1 Alstate:8 Node_profile:0.
BindAxis:6 Drive_profile:0 Controlword:0h drive_status:270h Drive_mode:8h target:0h enc:

在线命令: ?*EtherCAT  发送  捕获  清除
```

Slot 0 contain 1 nodes: 0 槽位口共连接了 1 个设备

Lostcount 0-0: 丢包数

Node: 设备连接 NODE 编号

Status: 设备连接状态, 参考 NODE_STATUS

Manid: 厂商 ID

Productid: 设备 ID

Axes: 设备总轴数

AL Status: 设备 OP 状态

Node_profile: 设备 Profile 设置

Bindaxis: 映射到控制器轴号

Drive_profile: 设备收发 PDO 设置

Controlword: 控制字

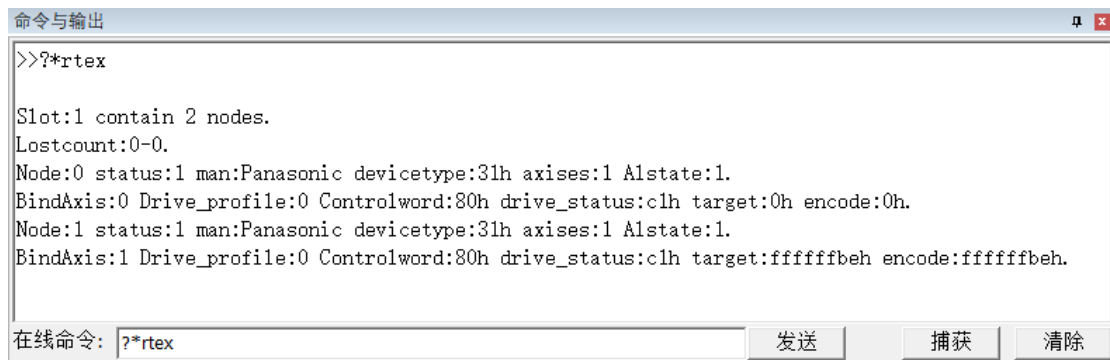
Drive_status: 设备当前状态, 参考 DRIVE_STATUS

Drive_mode: 设备控制模式

Target: 电机位置

Encoder: 编码器位置

?*RTEX: 打印 Rtex 总线连接设置状态



```
命令与输出
>>?*rtex

Slot:1 contain 2 nodes.
Lostcount:0-0.
Node:0 status:1 man:Panasonic devicetype:3lh axes:1 Alstate:1.
BindAxis:0 Drive_profile:0 Controlword:80h drive_status:c1h target:0h encode:0h.
Node:1 status:1 man:Panasonic devicetype:3lh axes:1 Alstate:1.
BindAxis:1 Drive_profile:0 Controlword:80h drive_status:c1h target:fffffbh encode:fffffbh.

在线命令: ?*rtex  发送  捕获  清除
```

Slot 0 contain 1 nodes: 0 槽位口共连接了 1 个设备

Lostcount 0-0: 丢包数

Node: 设备连接 NODE 编号

Status: 设备连接状态, 参考 NODE_STATUS

Manid: 厂商 ID

Devicetype: 设备 ID

Axes: 设备总轴数

ALState: 设备 OP 状态

Bindaxis: 映射到控制器轴号

Drive_profile: 设备收发 PDO 设置

Controlword: 控制字

Drive_status: 设备当前状态, 参考 DRIVE_STATUS

Target: 电机位置

Encode: 编码器位置

?*FRAME: 打印机械手参数, 需要 161022 及以上固件支持

>>?*FRAME

BASE[0,1,2,3,4,5]

CONNFRAME[6,0,6,7,8,9,10,11]

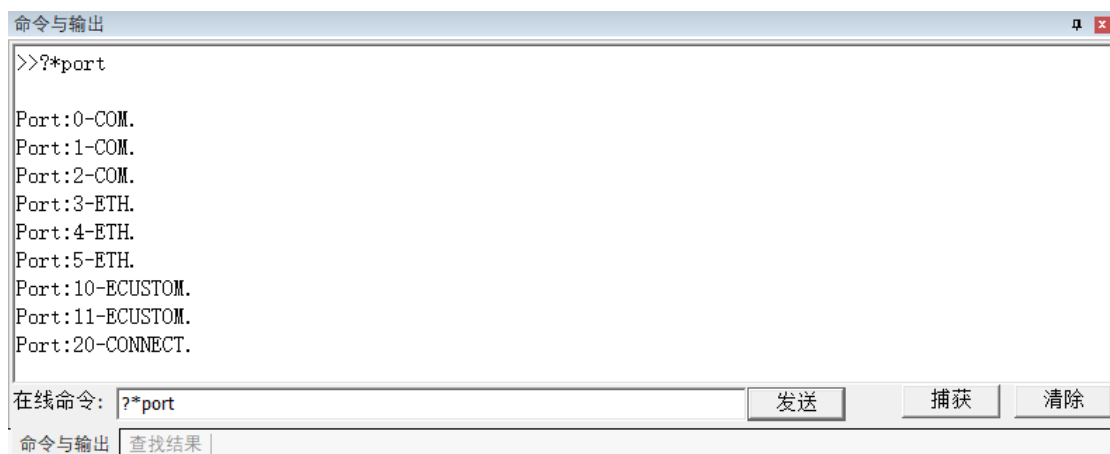
?*SLOT: 打印出控制器槽位口信息 (下图所示为双总线控制器, 包含一个 RTEX 口和一个 EtherCAT 口)

>>?*slot

Slot:0-ETHERCAT.

Slot:1-RTEX.

?*PORT: 打印所有 PORT 通讯口



COM: 串口通道

ETH: 网口通道

ECUSTOM: 自定义网口通道

CONNECT: 控制器互联通道

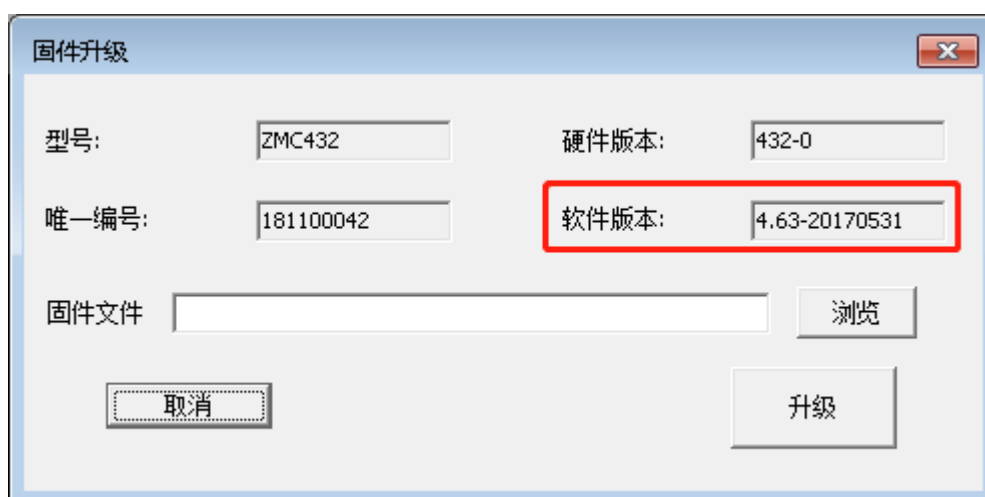
2.5 固件升级

固件升级用于对控制器现有固件版本更新，当前固件版本若无法满足程序运行要求，或部分指令功能不支持，就需要对固件进行升级。

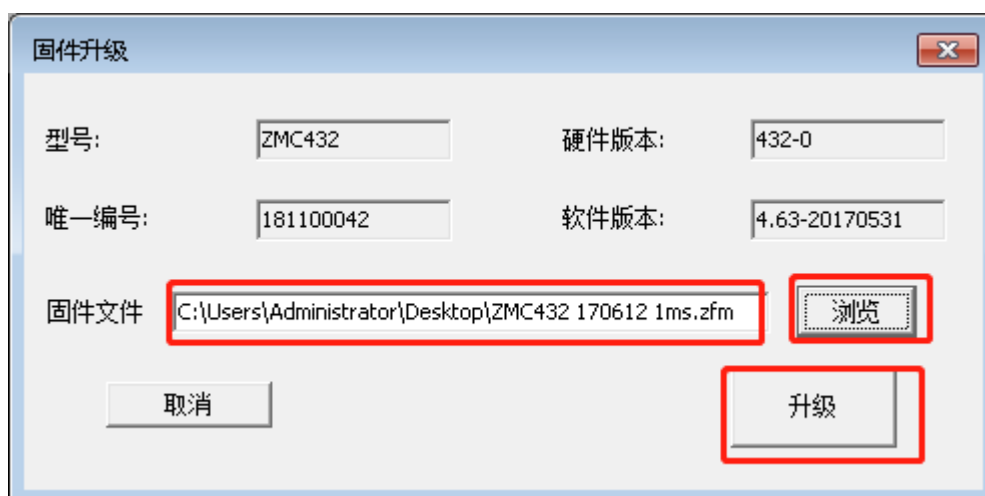
固件升级有两种方法，一种是使用 ZDevelop 软件自带的固件升级功能，另一种是采用 zfirmdown 工具软件下载 zfm 固件包，进行固件升级，两种升级操作方法类似，以下为 ZDevelop 软件固件升级操作指引。

先下载好需要升级的固件，点击 ZDevelop 软件菜单栏“控制器”-“固件升级”后，在弹出的窗口里点击“浏览”选择目标固件进行固件升级操作。

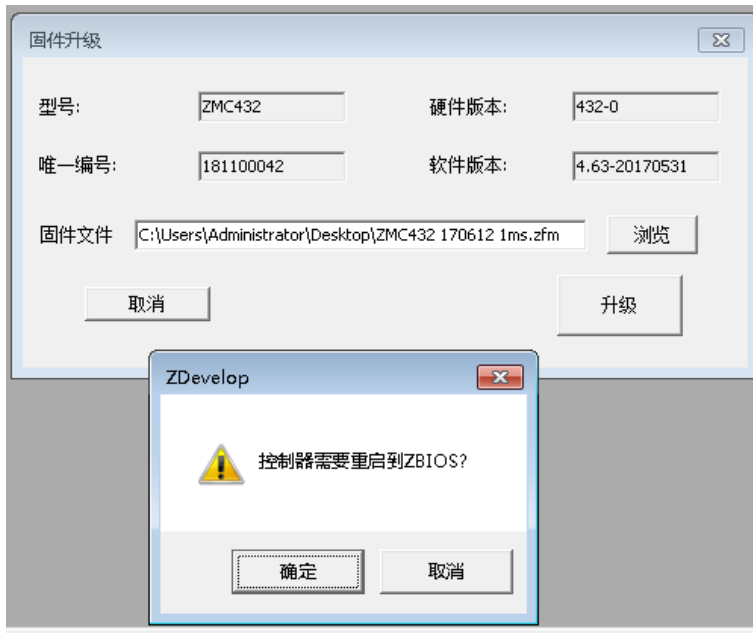
操作过程参见以下示例。



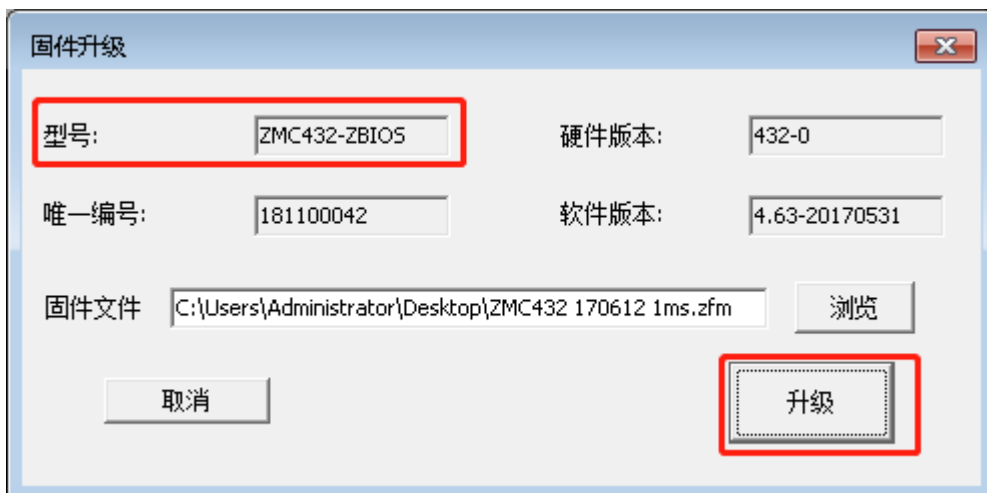
选择已保存的固件文件“ZMC432 170612 1ms.zfm”，点击“升级”。注意升级的固件版本需要与控制器硬件型号一致，否则会报错。



确认要升级之后会提示重启到 ZBIOS 的对话框，此时控制器需要重新连接，点击“确认”后会弹出“连接到控制器”窗口，选择合适的连接方式重新连接（串口或网口）。



控制器再次连接成功后，弹出“固件升级”界面，系统进入 ZBIOS 状态，显示型号为“ZMC432-ZBIOS”。再次点击“升级”。



弹出以下界面，正在下载文件。



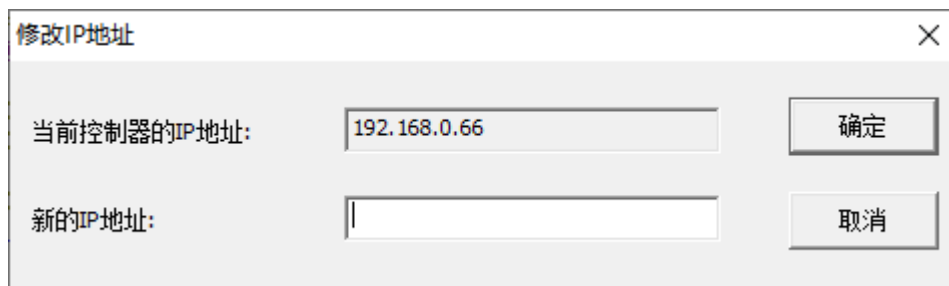
进度条满格后，“固件升级”界面消失，命令与输出窗口显示如下信息，表示固件升级成功。


```
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.
```

之后重新连接控制器、查看控制器状态的固件版本号。

2.6 修改 IP 地址

菜单栏“控制器”-“修改 IP 地址”，弹出如下窗口，此时会显示当前控制器 IP，在窗口可直接输入新的 IP 地址。



修改 IP 后，控制器与 ZDevelop 的连接会断开，此时再次选择新设置的 IP 地址连接即可。

2.7 系统时间设置

用户可选择自定义时间和同步 PC 时钟。



第三章 ZDevelop 项目

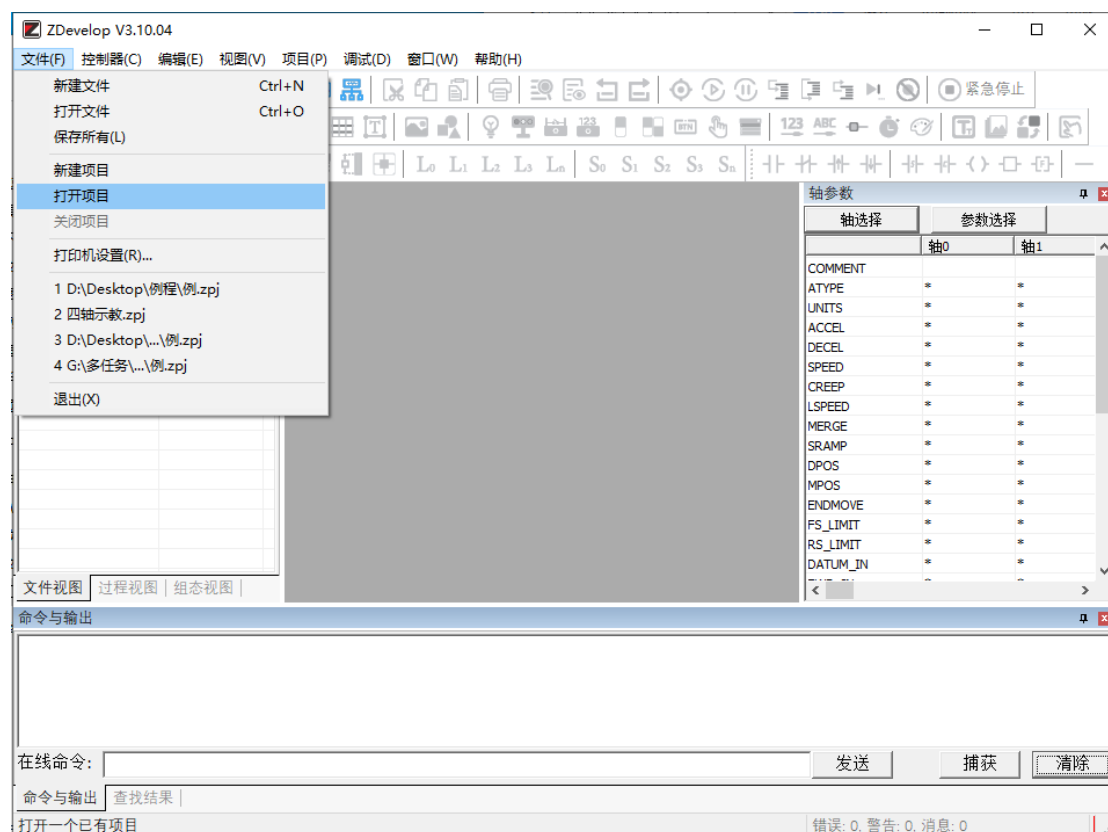
使用 ZDevelop 必须理解“项目”的概念。为便捷的进行应用设计开发，通过建立一个文件夹，里面包含该应用相关的各个程序，这样的集合体称之为“项目”，一个项目通过一个项目文件内含一个或多个文件来管理。

3.1 项目文件

项目文件的文件名后缀为“.zpj”，项目里面程序文件必须与项目文件位于同一个文件夹（即将zpj文件包含的bas/plc/hmi文件保存在同一文件夹内）。

打开文件时选择打开项目，已增加到项目的文件，即该zpj文件下的bas/plc/hmi文件会自动打开，或者拖动zpj文件到ZDevelop直接打开。

只打开 bas/plc/hmi 文件，不打开对应的项目，程序无法下载，也无法运行。



3.2 项目设置

预留功能。

3.3 程序文件

Basic 程序文件的文件名后缀为 “.bas”。

PLC 文件的后缀为 “.plc”。

HMI 组态文件的后缀为 “.hmi”。

3.4 ZAR 文件

ZAR 文件是一种加密文件，文件后缀为 .zar，项目文件生成 ZAR 文件之后，看不到任何代码，可以将 ZAR 文件下载到控制器运行，[下载 ZAR 文件](#)的方法参见第五章说明。

3.5 库文件

库文件靠“编译为 Lib”功能生成后保存，便于程序保密或防止程序被修改，库文件的文件名后缀为 “.zlb”，[编译 Lib](#)文件的方法参见第五章说明。

第四章 程序编辑

ZDevelop 集程序编辑和程序调试于一体，简单易用。

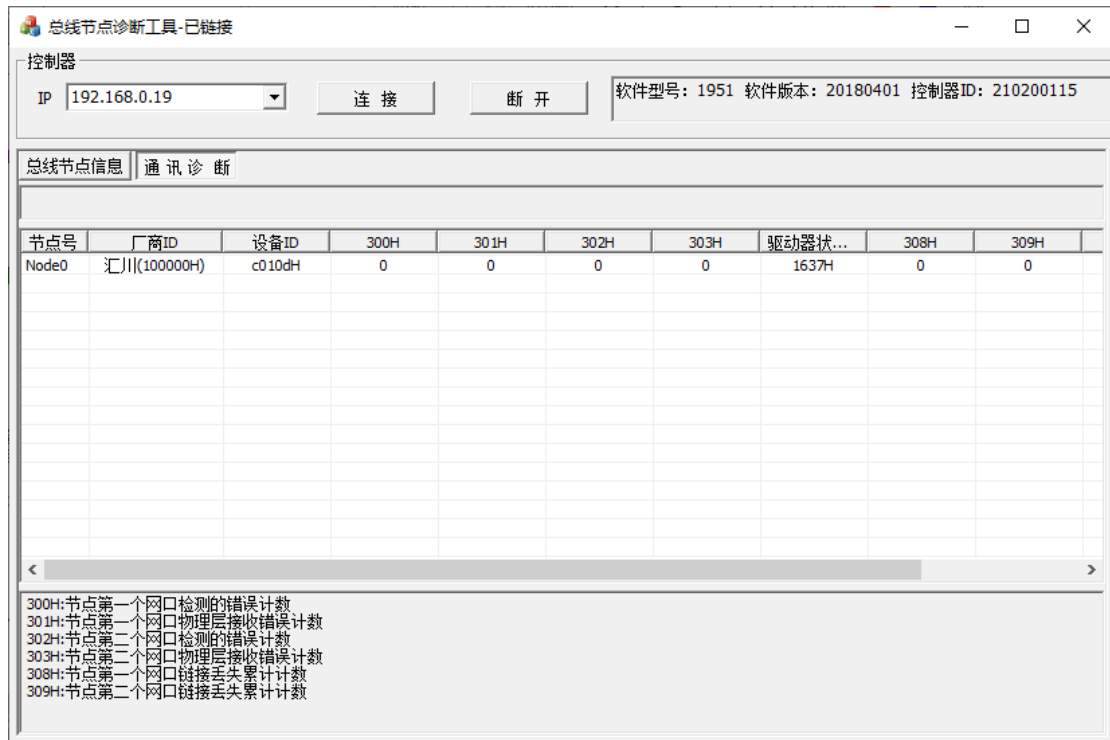
4.1 快捷编辑

ZDevelop 提供非常方便的程序编辑环境，多个快捷键供使用。

1. 工具栏：



鼠标放置于按钮上，自动提示各个按钮名称。



剪切/复制/粘贴/打印程序

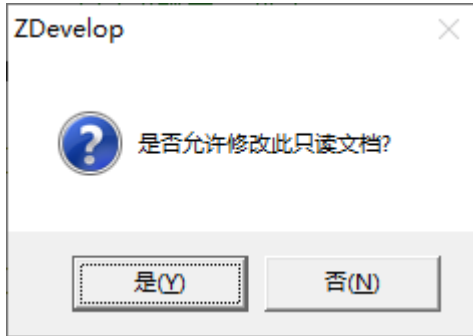
打印程序支持打印 Basic 程序和 PLC 语句表程序，HMI 和 PLC 梯形图不支持



搜索：在整个项目中搜索，搜索结果显示在“查找结果”窗口



只读：该状态下无法修改程序。开启只读功能后，图标变为蓝色，强制输入会弹出如下提示是否解除只读（HMI 不参与只读功能）

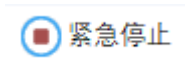


上一个地方：光标定位到上一处修改的地方

下一个地方：光标定位到下一处修改的地方




调试工具，调试状态下有效，详情参见[程序调试](#)章节



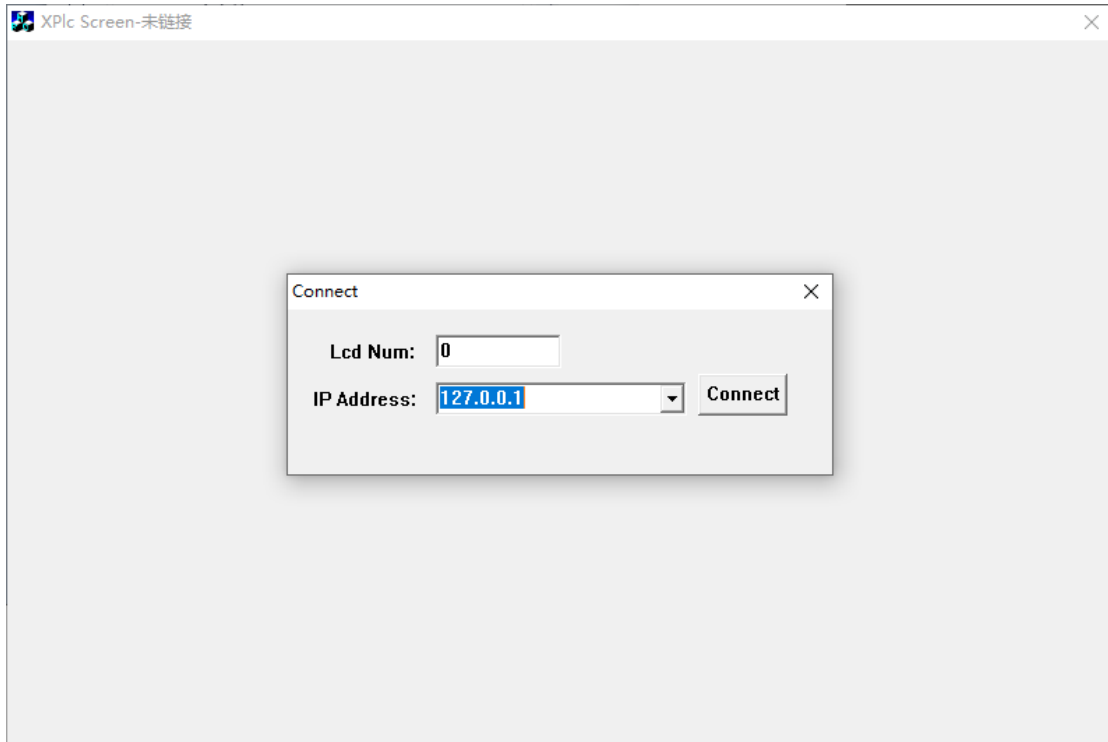
紧急停止：紧急停止程序和所有轴的运动

2. 控件工具栏：



HMI 元件快捷添加按钮，仅在 HMI 编程下有效，最后的手型按钮  为打开 xplc screen 触摸屏仿真工具的按钮。

先将程序下载到控制器或仿真器之后再点击手型按钮，弹出下图窗口，在小窗中选择控制器的 IP 后连接，即可显示出已下载到控制器/仿真器的 HMI 组态界面。



连接成功后显示的界面：



3. HMI 工具栏：



左半部分为 HMI 元件对齐按钮，右半部分为 HMI 语言切换和状态切换。

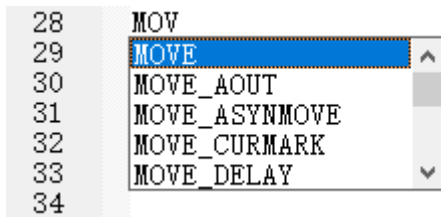
4. 梯形图工具：



PLC 梯形图快捷输入按钮。

4.2 Basic 编辑

指令输入具有快速提示输入功能。



提供了语法高亮功能。指令、变量参数、注释等分别采用不同颜色显示。

IF 指令下方的线条为对齐线，可使用菜单栏“视图”-“对齐线”取消。“切换自动换行”功能可以让程序显示跟随当前程序编辑窗口的大小自动换行。

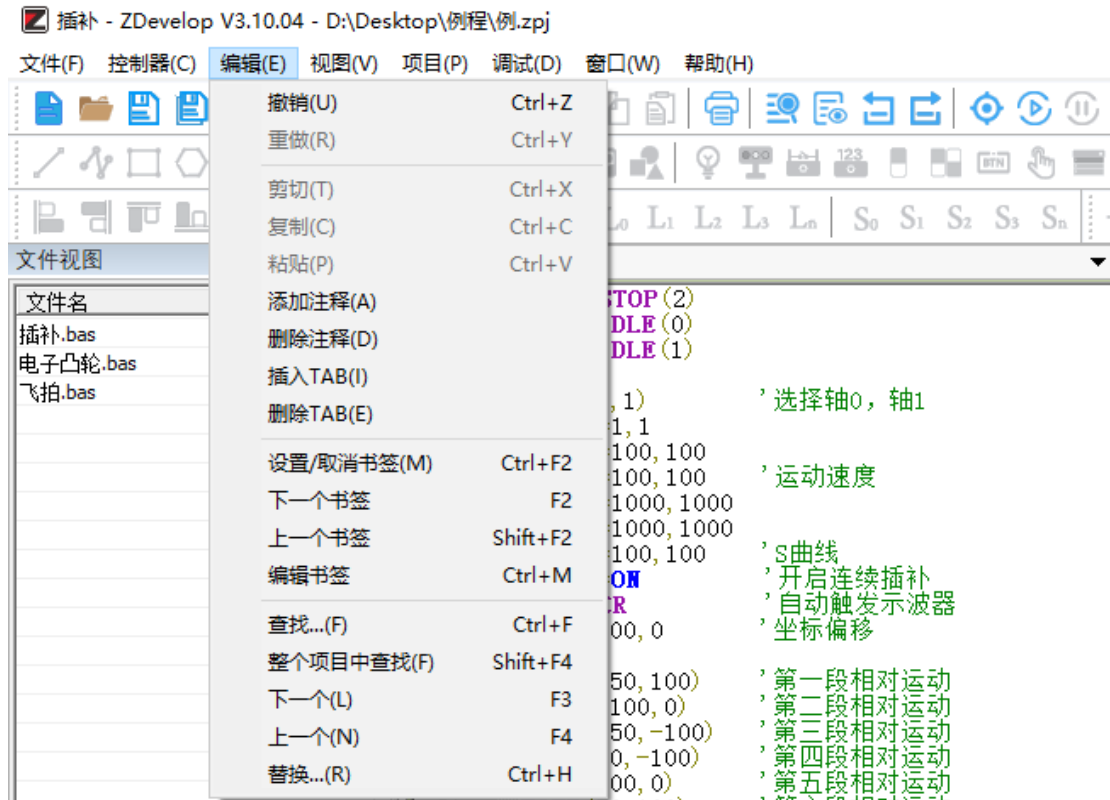
```

14  while 1 '循环运动
15
16      if in(0) = on then '输入0有效启动运动
17          testmove(1000,1000)
18          wait idle '等待运动停止
19          delay(100) '延时
20      endif
21
22  vend
23
24  end
25
26  'move子过程调用,跑道形状
27  sub testmove(radius, length)
28      '调试输出
29      trace "distance = " length, "radius = " radius
30      move(length)
31      movecirc(0, radius*2, 0, radius, 0)
32      move(-length)
33      movecirc(0, -radius*2, 0, -radius, 0)
34  end sub
35

```

“添加注释”可快速将程序块变为注释部分，“删除注释”是将注释变为程序。

操作方法：在 Basic 程序中选中某段程序，点击菜单栏“编辑”-“添加注释”后，选中的程序就变为了注释；选中某段注释，点击菜单栏“编辑”-“删除注释”后，选中的注释就变为了程序。



菜单栏“编辑”-“插入 TAB”/“删除 TAB”功能用于控制程序缩进量，对整个 IF 循环语句执行“删除 TAB”操作后的效果如下，单次操作，缩进量是固定的。

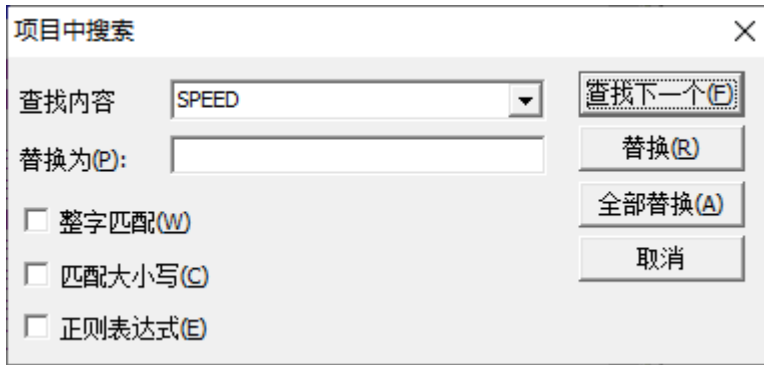
```

14  while 1 ' 循环运动
15
16  if in(0) = on then ' 输入0有效启动运动
17      testmove(1000, 1000)
18      wait idle ' 等待运动停止
19      delay(100) ' 延时
20  endif
21
22  wend
  
```

菜单栏“编辑”-“查找”，在项目中查找。



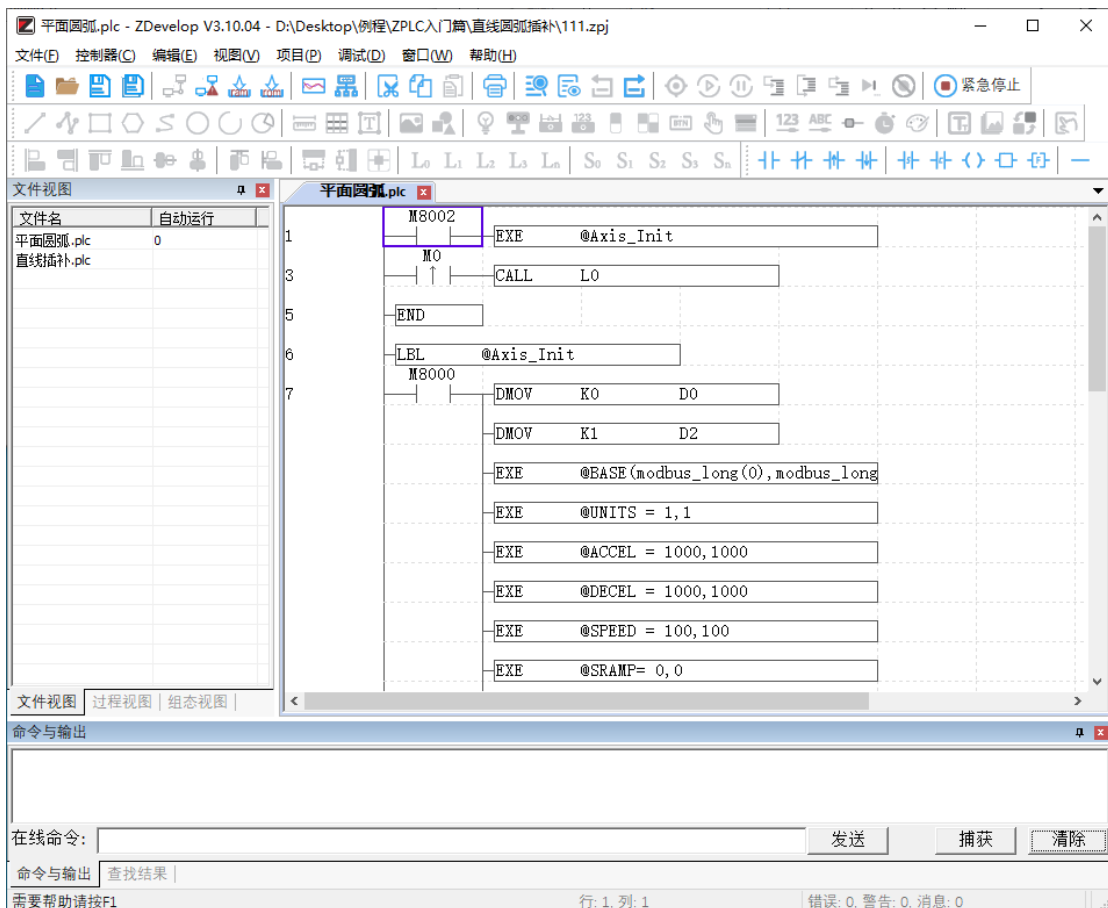
菜单栏“编辑”-“替换”，在项目中查找并替换。



4.3 梯形图编辑

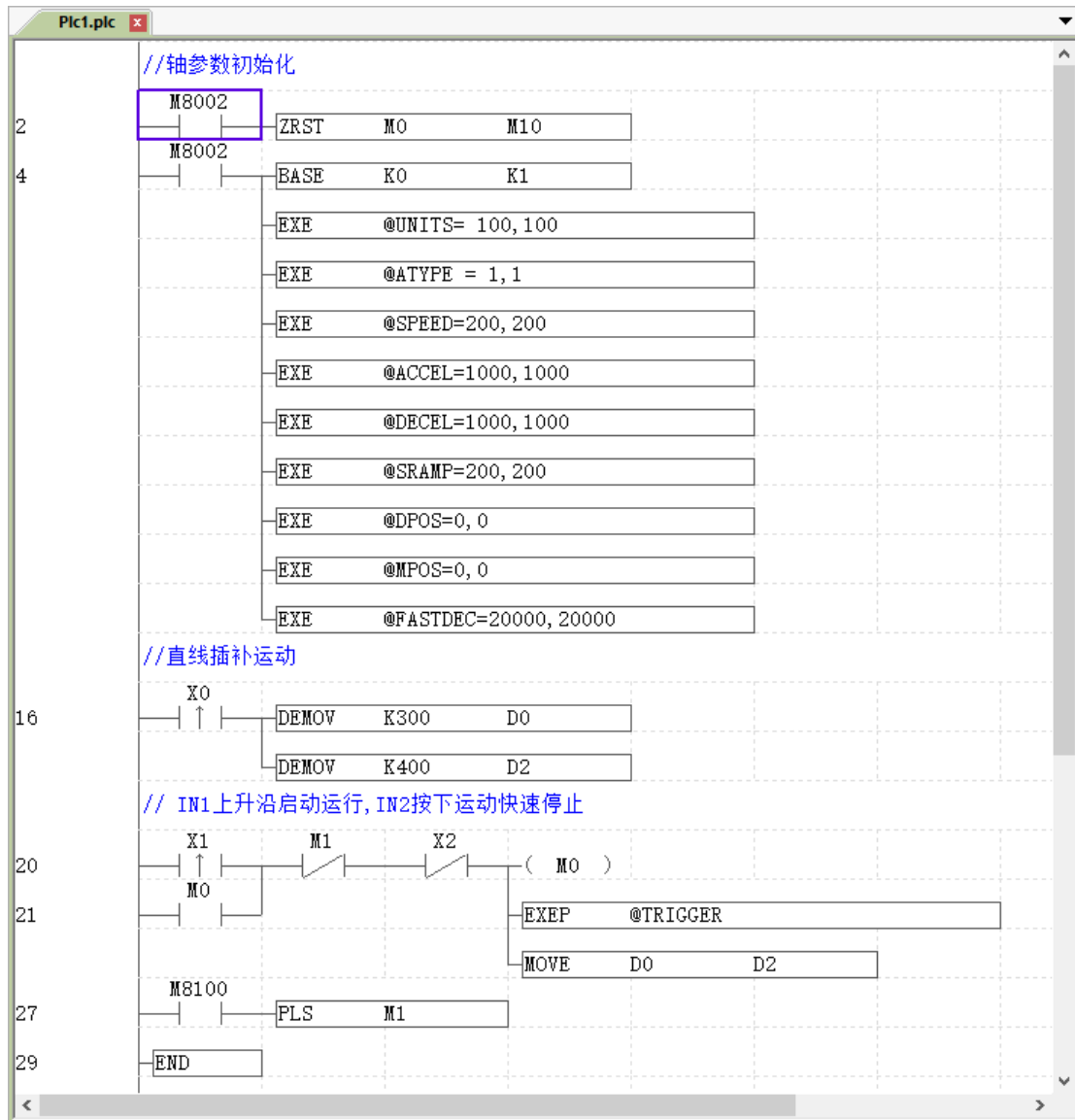
ZDevelop 提供 PLC 梯形图与语句表两种编程方式，且梯形图和语句表可相互转换两种模式下均可编辑程序，指令不分大小写。

通过菜单栏的“编辑”-“转换成语句表”“转换成梯形图”互相转换。

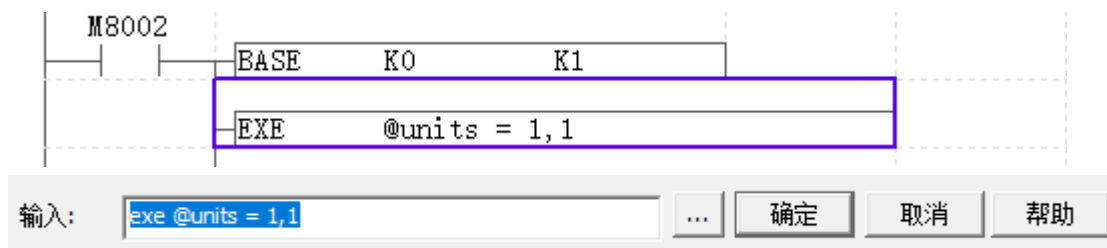


梯形图编程较为直观方便，语句表编程需要对 PLC 指令较为熟悉。

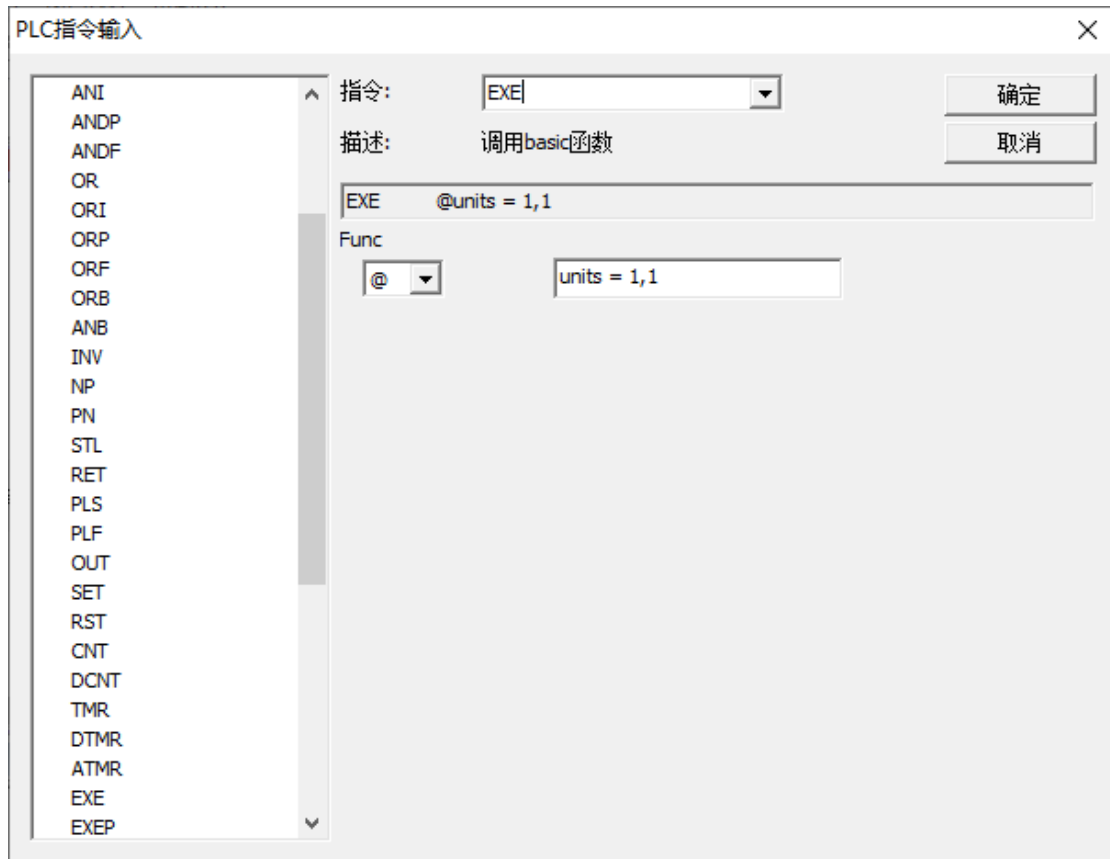
语句表编程界面如下所示：



在对应的单元格双击或直接输入命令，也可以实现快捷输入。可输入指令或修改参数，点击“确认”保存修改结果，点击“取消”关闭快捷输入框，点击“帮助”打开帮助文档。



三个点按钮表示打开 PLC 指令输入框，可选择指令和指令操作数，如下图。

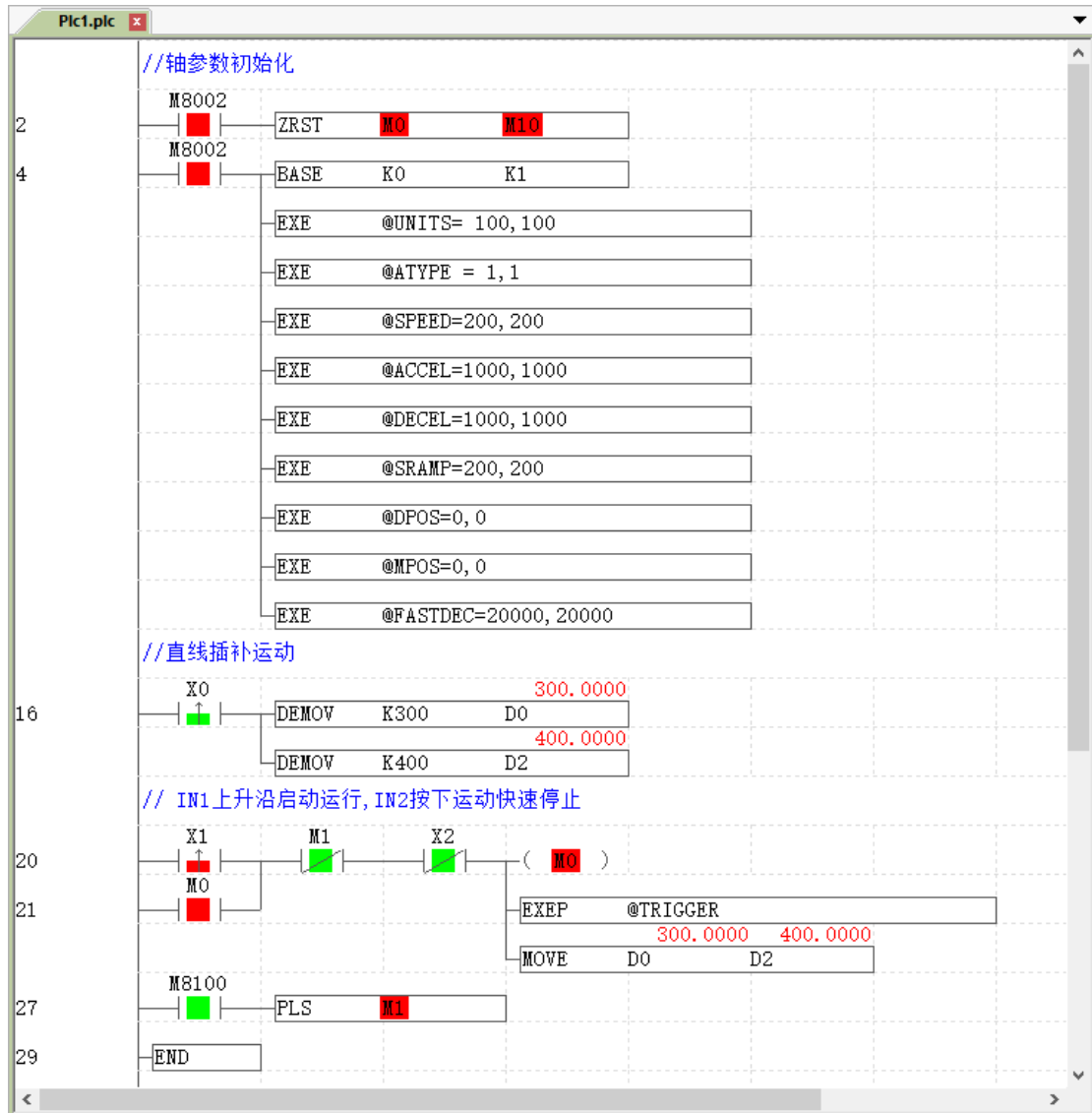


程序编辑完成在末尾加上 END，否则无法运行。

PLC 中使用 EXE @指令调用 Basic 的命令执行，PLC 和 Basic 还有一些共享的变量，详情参见 PLC 编程手册。

调试

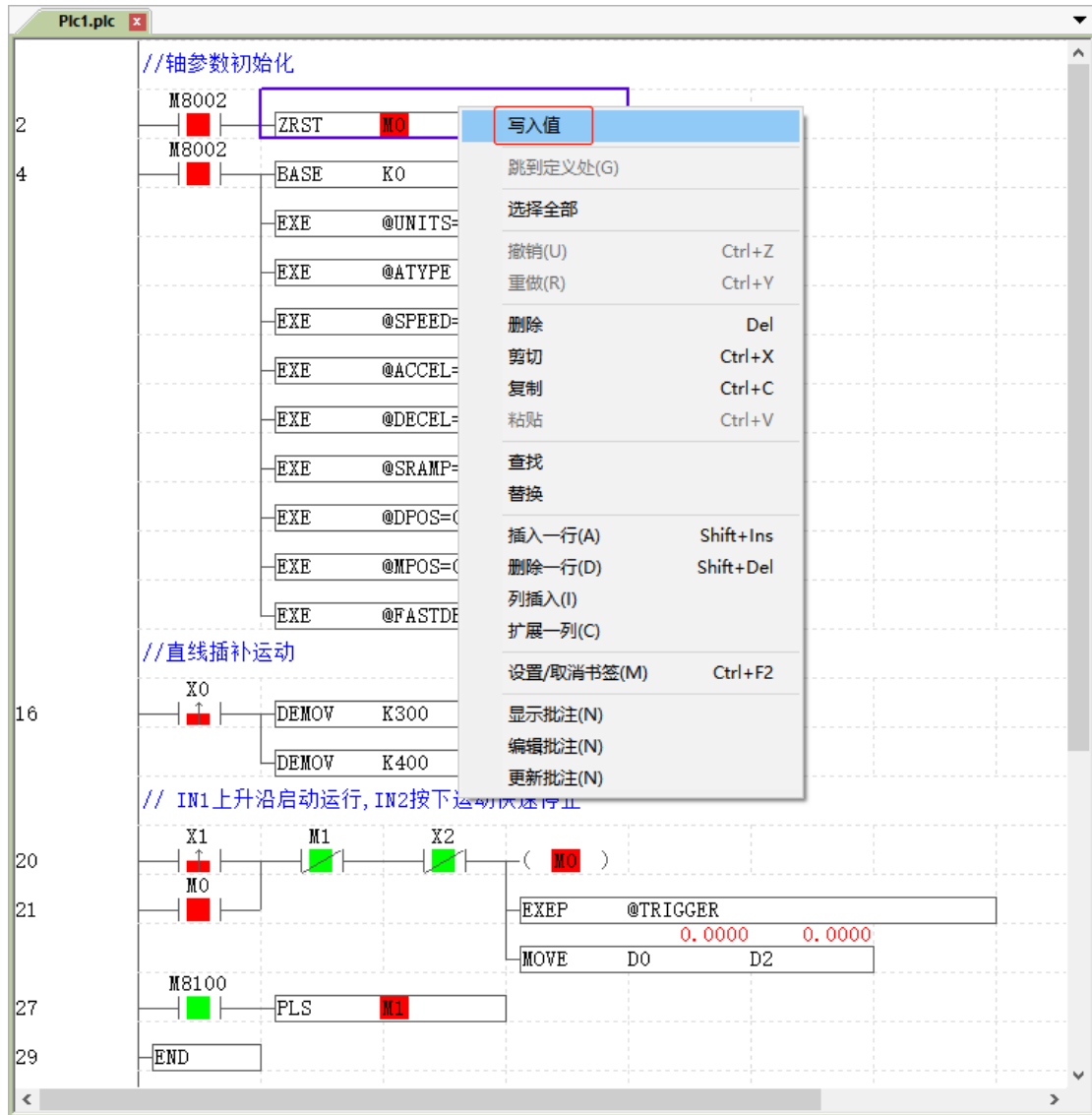
进入调试模式，会出现红绿色提示，绿色表示导通，红色表示关断。调试模式下，在寄存器上方也会显示寄存器的当前值。



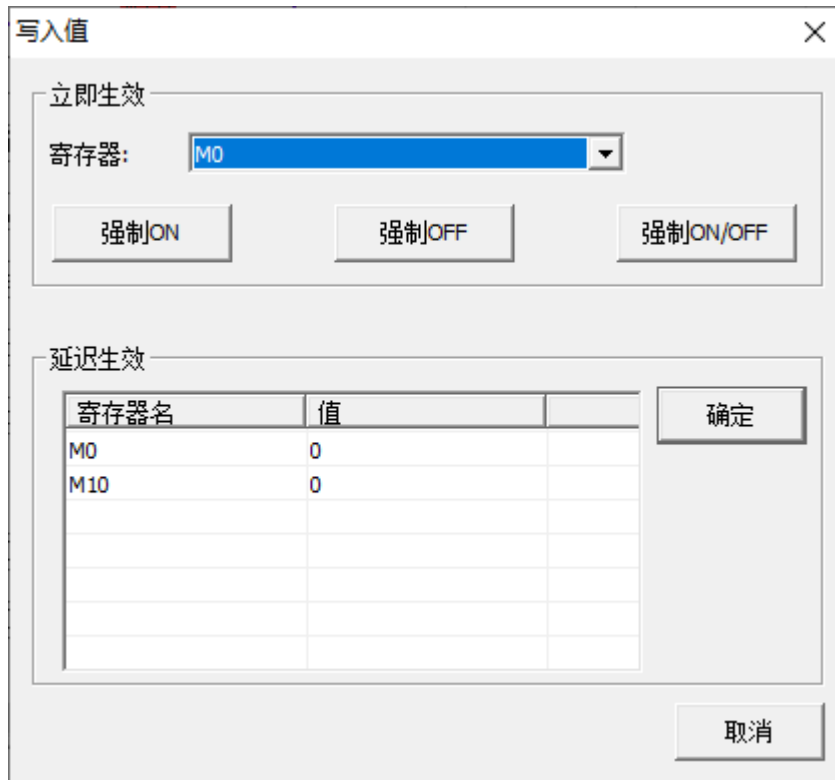
PLC 梯形图模式下不支持断点调试，指令表模式下支持断点调试。

写入值

调试状态下，选中目标窗格，点击右键打开写入值窗口，快速对位变量 M 或字节变量 D 的值进行编辑修改。下图示例给 M0 写入数据 ON 或 OFF。



写入值窗口如下，仅显示当前窗格选中的寄存器。

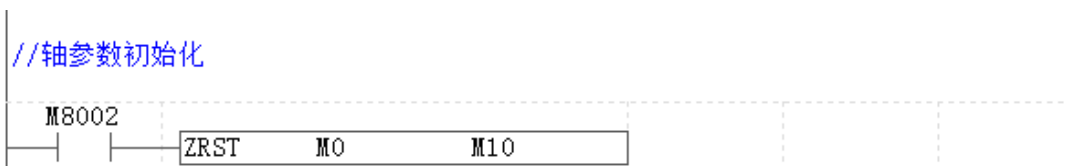


注释

PLC 梯形图添加注释方式有三种，使用 PLC 注释指令、PLC 右键菜单“编辑批注”、菜单栏“视图”-“注释”窗口。

1. 注释指令“//”

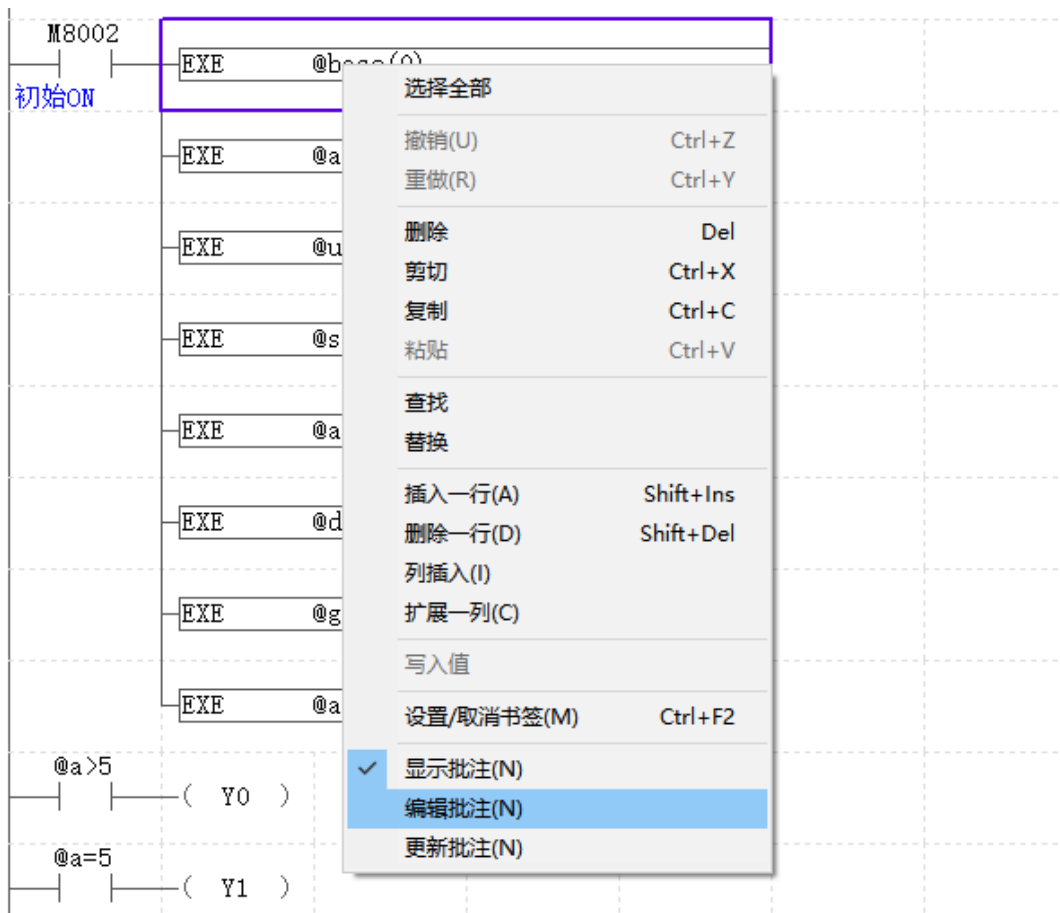
使用该指令注释占一行，一般在某个功能模块前添加注释，补充说明该功能的作用等相关信息。



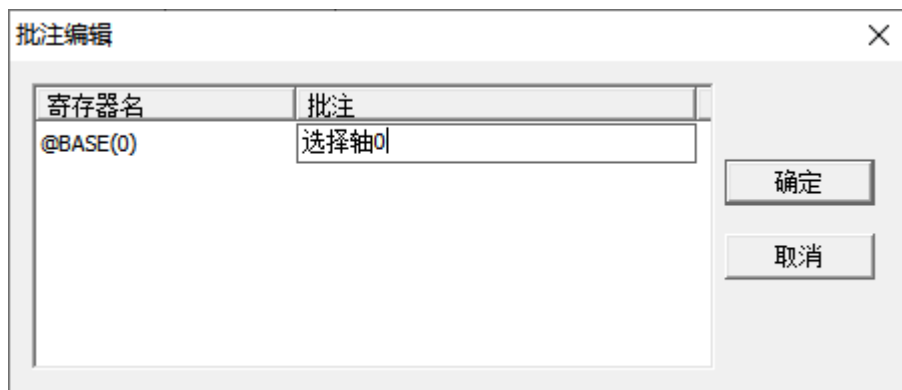
2. 编辑批注

此功能在梯形图界面下可对软元件进行注释。

批注方法：选中要批注的软元件，点击右键选择“批注编辑”。



弹出批注编辑窗口之后，在批注下方双击左键，输入注释内容，点击确定。



完成操作成功显示批注，批注程序默认显示，不希望显示批注时不勾选“显示批注”这一项即可隐藏批注。



3. 注释窗口

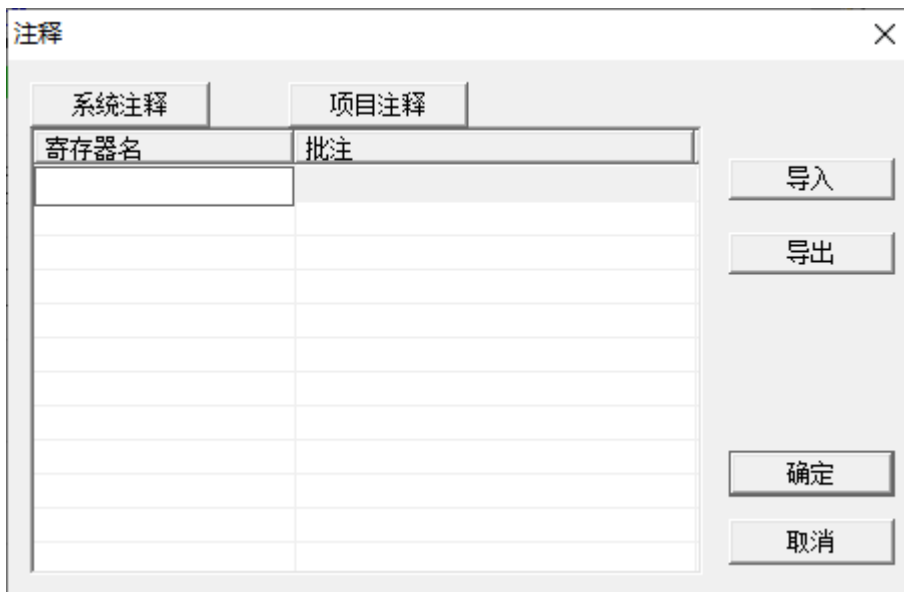
菜单栏“视图”-“注释”打开注释窗口用于对 PLC 元件进行批注。

其中“系统注释”主要针对 PLC 特殊继电器 M 和特殊寄存器 D，可快速对所有特殊元件

进行注释。



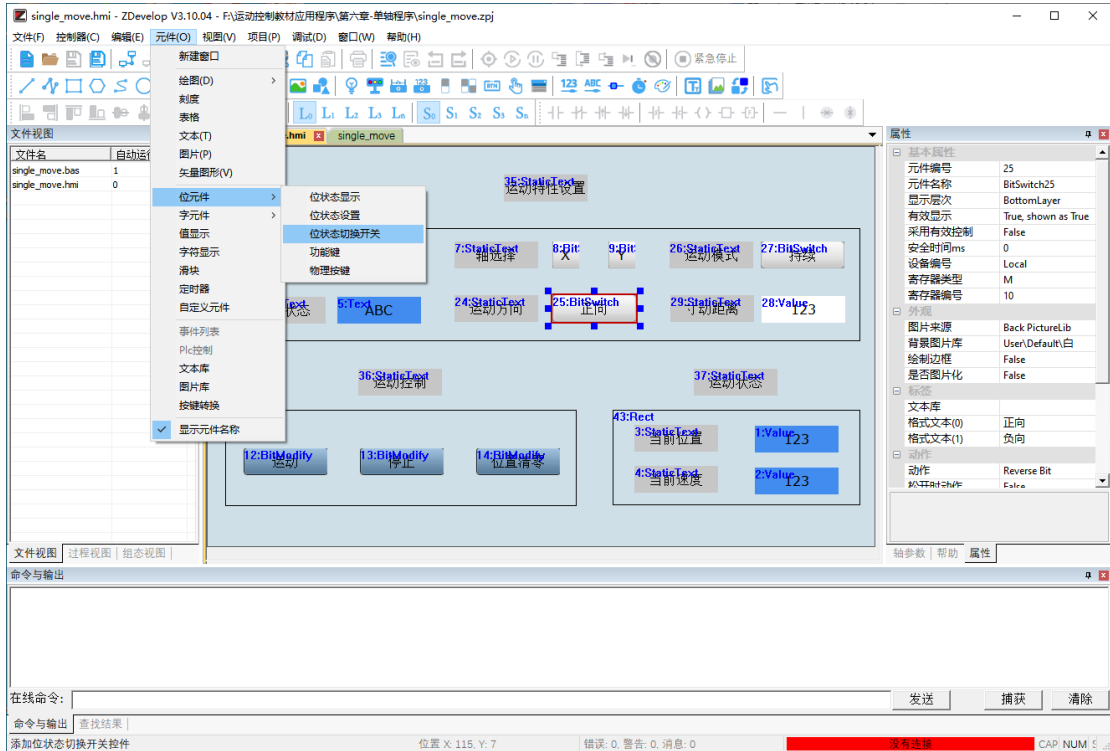
“项目注释”自定义输入寄存器名和注释，与编辑批注使用类似，支持将此窗口的项目注释导入导出。



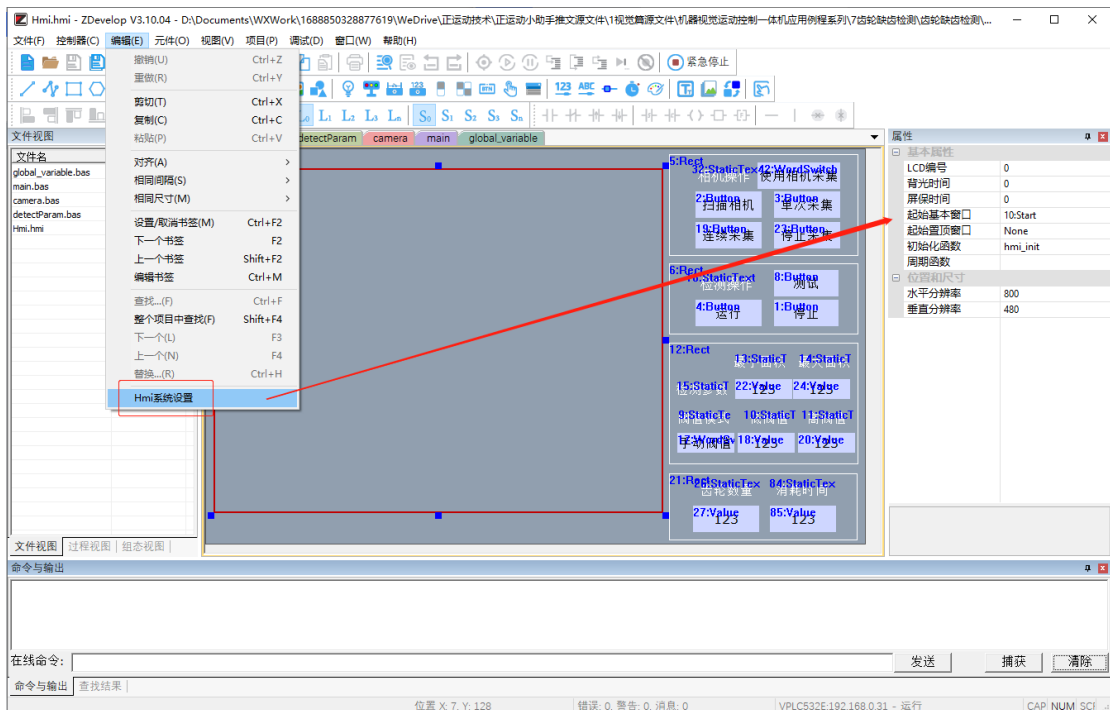
4.4 HMI 组态编辑

HMI 编程用于在显示屏自定义显示，触摸执行相关操作。打开组态“元件”可快速选择输入对象，提高编程效率。

元件常用的功能为：调用 Basic 子函数，打开/关闭组态窗口，绑定寄存器，显示文本/图片等，不同的元件功能有所区别，详细用法参见《ZHMI 编程手册》。



菜单栏“编辑”-“Hmi 系统设置”，要运行 HMI 文件必须要进行此设置，Hmi 系统设置如右侧窗口所示。



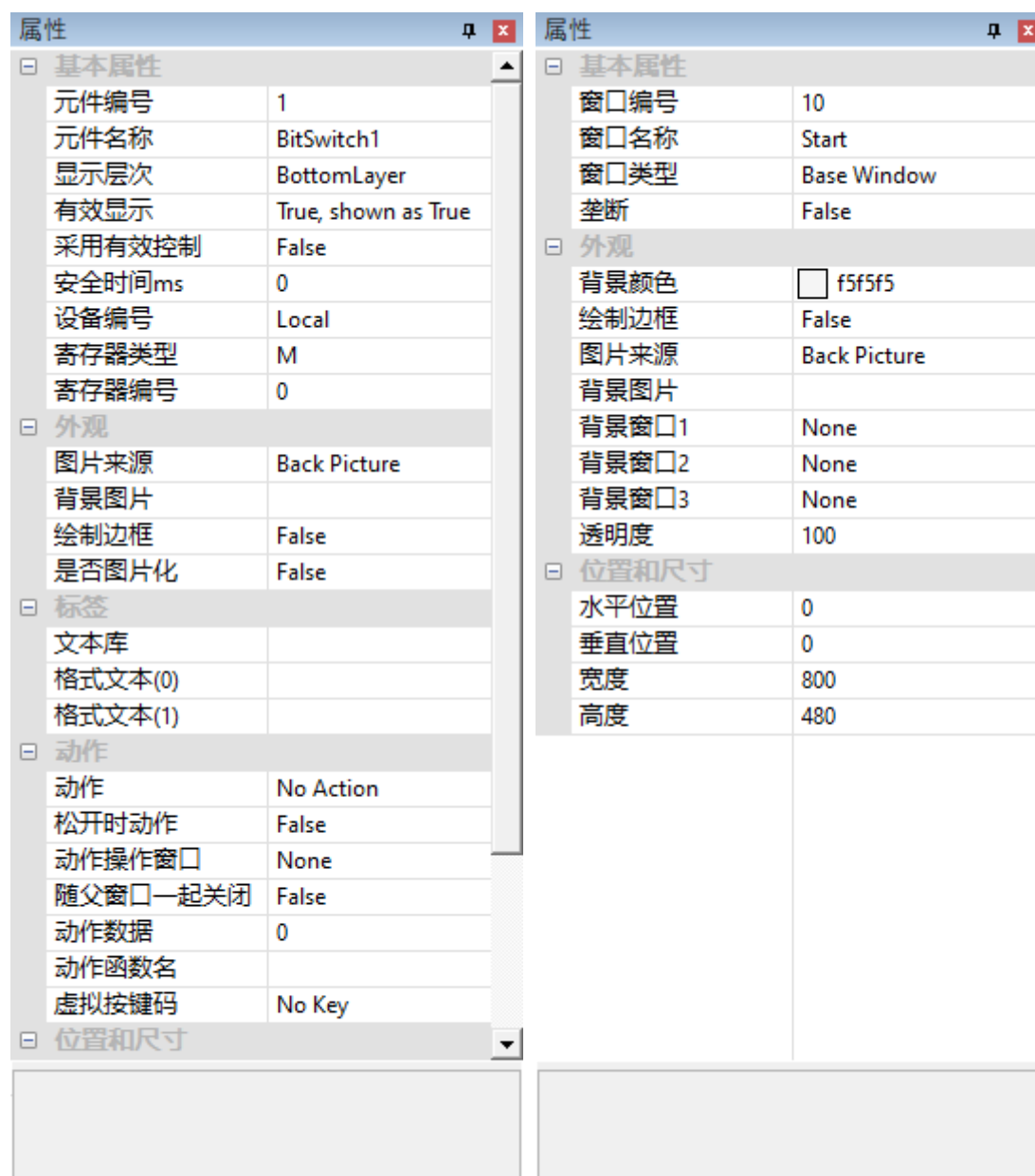
分辨率根据显示屏尺寸设置，常见的 7 寸屏尺寸 800*480。

初始化函数为上电后只调用一次的函数，在 Basic 文件中定义，函数的定义必须是全局的(GLOBAL)。

周期函数为不断扫描的函数，在 Basic 文件中定义，函数的定义必须是全局的(GLOBAL)。

初始化函数或周期函数不是必须设置的，HMI 调用 Basic 函数的方式还有自定义元件的绘图函数或刷新函数，元件的动作 call sub 等。

在组态视图中，右键菜单“属性”可快速编辑窗口显示情况，弹出的属性窗口与鼠标放置的位置有关，鼠标放于元件上右键属性窗口如左图所示，鼠标放于背景空白处右键属性窗口如右图所示。



元件右键“属性”功能窗口

背景右键“属性”功能窗口

位状态切换开关 BitSwitch 的属性说明：

属性	功能	说明
元件编号	/	/
元件名称	/	/
显示层次	选择元件显示层次	TopLayer: 表层，显示在最外层，覆

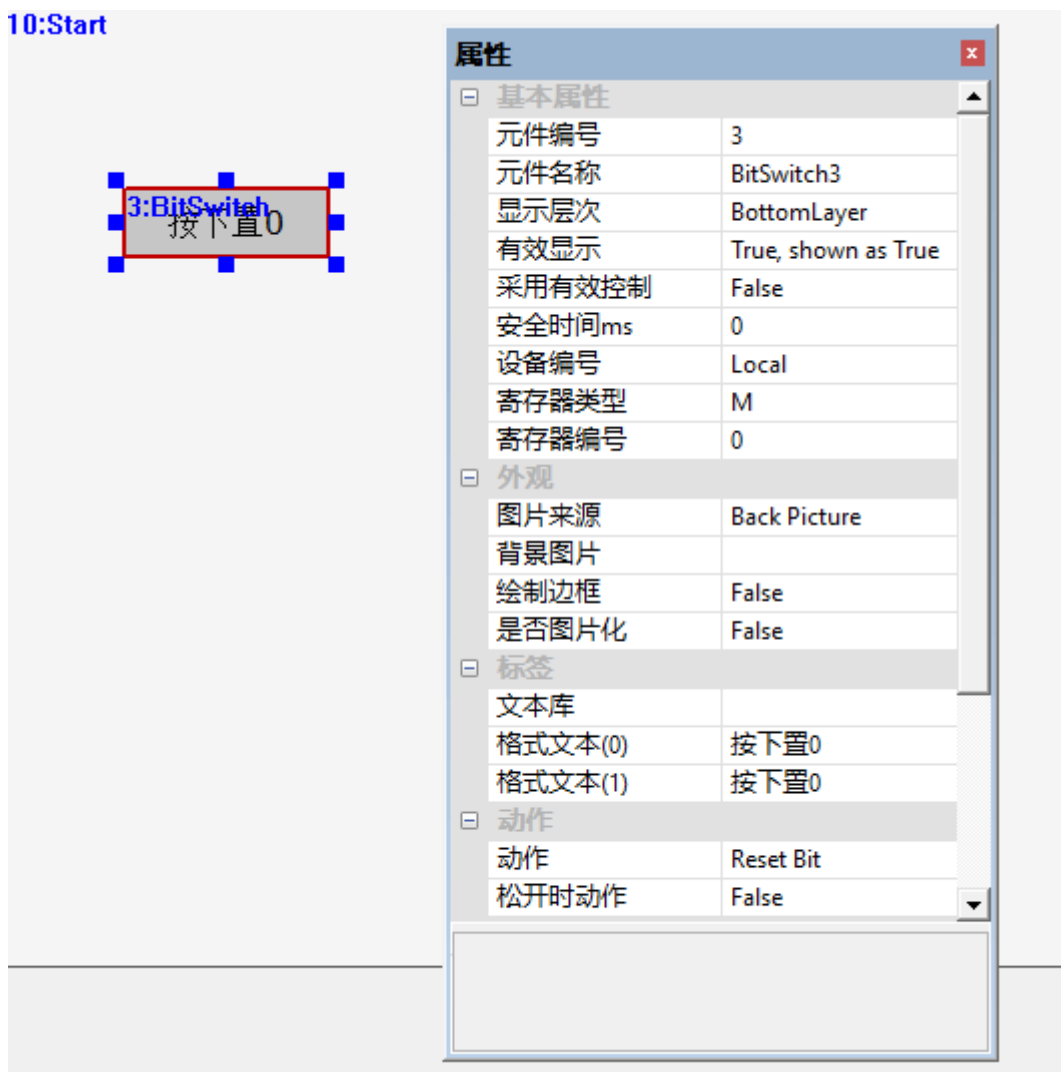
		盖底下元件 MidLayer: 中间层 BottomLayer: 底层(默认)
有效显示	选择元件是否显示	默认 True, 选择 False 时, 元件不显示且无功能作用
采用有效控制	通过寄存器控制元件是否显示	默认 False, 选择 True 通过寄存器控制元件是否显示
安全时间 ms	最少按键时间	单位 ms
设备编号	设备编号	默认 local
寄存器类型	选择寄存器类型	多种寄存器下拉列表选择
寄存器编号	选择寄存器编号	寄存器值为 0 时不显示, 非 0 时使用
图片来源	Back PictureLib 或 Back Picture	图片库或背景图片中选择
背景图片	背景图片选择	在图片来源先选择背景图片后添加
绘制边框	选择是否绘制边框	默认 False
是否图片化	元件变为图片的形式	默认 False
文本库	文本库的名称	不设置文本库显示格式文本
格式文本 0/1	打开格式文本设置窗口设置元件要显示的文本	默认显示文本 0, 按下时显示文本 1
动作	按键执行时的动作	参见“动作”章节描述
松开时动作	选择按下时或松开时执行动作	默认 False 为按下执行动作, True 为松开时动作
动作操作窗口	选择需要操作的窗口编号	下拉列表选择已有窗口
随父窗口一起关闭	子窗口随父窗口一起关闭	默认 False
动作数据	按键动作后给寄存器写入指定值	/
动作函数名	按键动作后要调用的 SUB 函数	下拉列表选择 Basic 已有全局 SUB 函数
虚拟按键码	选择虚拟按键码	默认不选择
水平位置	元件的水平起始位置	不要超出水平分辨率
垂直位置	元件的垂直起始位置	不要超出垂直分辨率
宽度	元件的宽度	/
高度	元件的高度	/

新建 HMI 文件运行步骤:

1. 建立 HMI 文件，添加自动运行任务号。
2. 打开“HMI 系统设置”窗口，设置屏幕分辨率，起始显示的窗口等。
3. 编辑 HMI 文件，新建窗口，添加元件，打开对应“属性”设置窗口和元件的功能。
4. 新建 Basic 文件，编辑 HMI 元件要调用的 Basic 子函数内容。
5. 在“HMI 系统设置”窗口添加初始化函数和周期函数，在要调用 Basic 程序的元件动作处添加子函数名称。
6. 连接控制器或仿真器，下载程序运行。无实物屏连接到 xplc screen 仿真。

HMI 编程案例：绑定寄存器

1. 新建 HMI 文件，并添加自动运行任务号。
2. 在起始 10 号窗口添加位状态切换开关元件。
3. 打开元件“属性”窗口选择寄存器类型和编号，“动作”选择“Reset Bit”，表示给绑定的寄存器赋值 0。
4. 下载程序到控制器或仿真器，在组态界面按下主界面的元件查看效果。

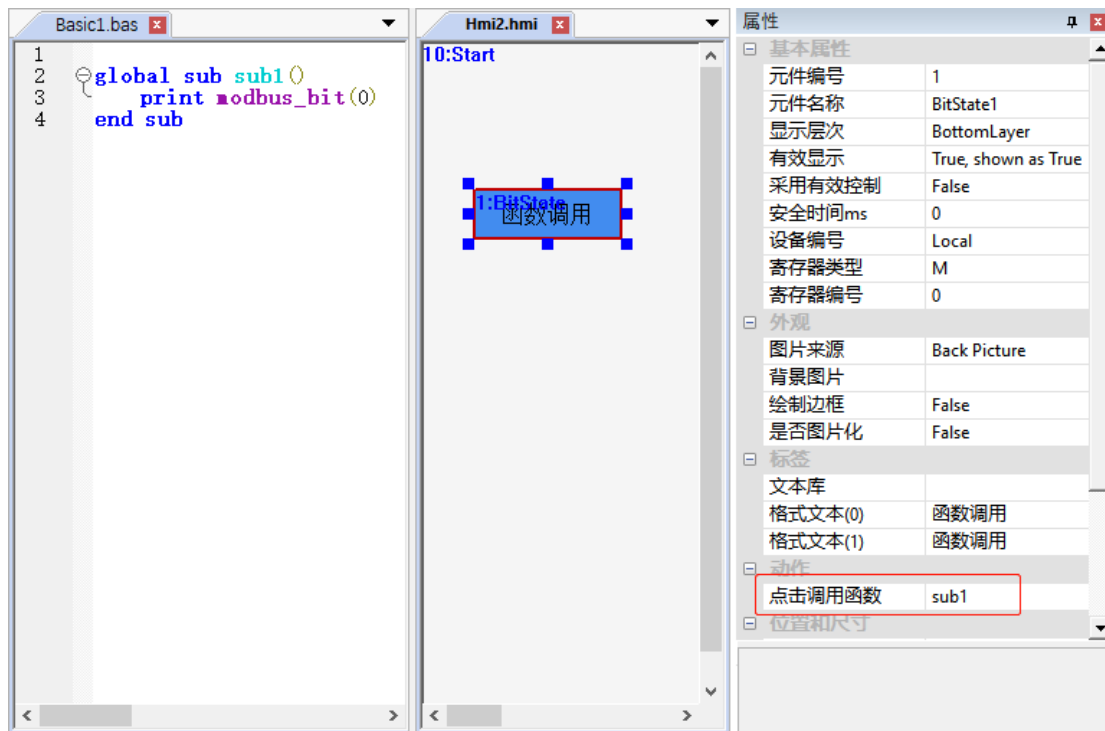


实现效果:

按下元件时, MODBUS_BIT(0)=0, 如果选择了松开时动作, 即按下元件再松开后, MODBUS_BIT(0)=0, MODBUS_BIT(0)的值为保持为0。

HMI 编程案例: 调用 SUB 函数

1. 新建 HMI 文件, 并添加自动运行任务号。
2. 在起始 10 号窗口添加位状态显示元件。
3. 新建 Basic 文件, 无需设置自动运行任务号, 在 Basic 里编辑好 HMI 要调用的全局 SUB 子函数。
4. 打开元件“属性”窗口, 在“点击调用函数”处选择上一步编辑好的 SUB 子函数名称。
5. 下载程序到控制器或仿真器, 在组态界面按下主界面的元件查看效果。



实现效果:

当元件被按下时, 调用 Basic 的 SUB 子函数执行, 每按下一次调用一次函数。

更多 HMI 例程在正运动官网下载。

4.5 右键菜单

通过右键菜单, 可以做各种特殊编辑操作, 也可以方便的跳转到变量或 SUB 的定义处, 还可以直接增加断点或添加到监视列表。以下分别为 Basic 文件 (左图) 与 PLC 文件 (右图) 的右键功能。

HMI 右键窗口参见上节内容。

跳到定义处(G) modbus_reg	
撤销	Ctrl+Z
重做	Ctrl+Y
剪切	Ctrl+X
复制	Ctrl+C
粘贴	Ctrl+V
添加注释(A)	
删除注释(D)	
查找	
替换	
选择所有	
设置/取消书签	Ctrl+F2
设置/取消断点	F9
增加到监视(W)	

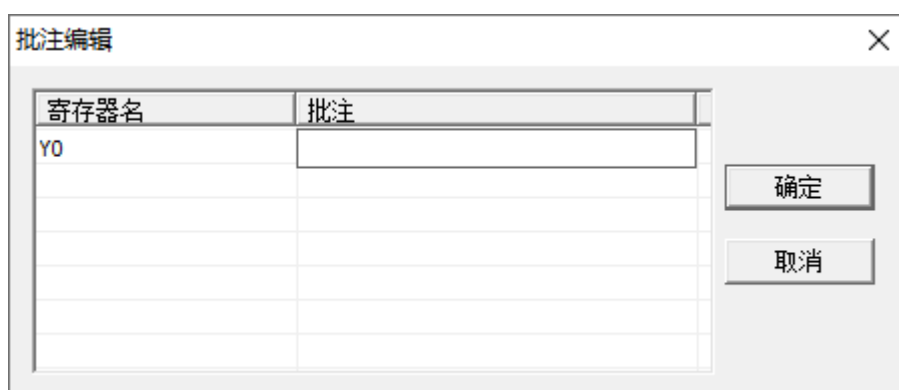
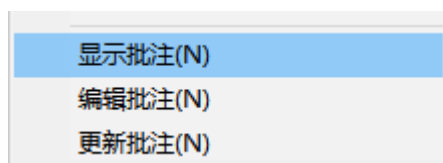
Basic 文件中的右键功能

写入值	
跳到定义处(G)	
选择全部	
撤销(U)	Ctrl+Z
重做(R)	Ctrl+Y
删除	Del
剪切	Ctrl+X
复制	Ctrl+C
粘贴	Ctrl+V
查找	
替换	
插入一行(A)	Shift+Ins
删除一行(D)	Shift+Del
列插入(I)	
扩展一列(C)	
设置/取消书签(M)	Ctrl+F2
显示批注(N)	
编辑批注(N)	
更新批注(N)	

PLC 文件中的右键功能

批注功能

可以对 PLC 寄存器进行批注，在 PLC 编程区域的右键菜单中可以选择是否显示批注。可以从其他项目文件中导入其批注信息。



查找/替换功能

菜单栏“编辑”-“替换”可以快速替换 M/D/T/C/S 的变量，例如将指令 LD M0 替换为 LDI M0。查找替换也可以通过在 Basic/PLC 程序编辑区点右键打开。



4.6 书签功能

通过“编辑”菜单可以设置/取消书签，或是查找上一个下一个书签。



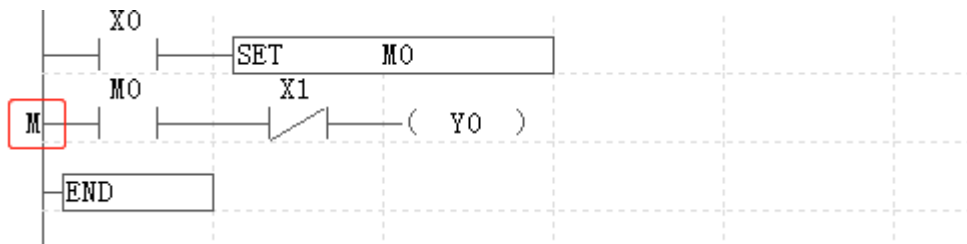
Basic 设置书签后在语句前方显示绿色竖线，如下图。

```

7  BASE(0,1)
8  DPOS=0,0
9  ATYPE=11,1
10 UNITS=100,100
11 SPEED=100,100
12 ACCEL=1000,1000

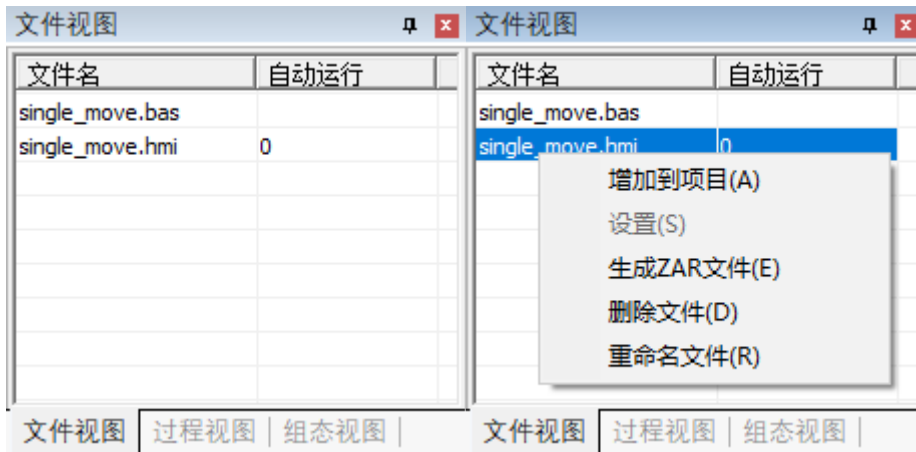
```

PLC 设置书签后在梯形图前方显示字母“M”，如下图。



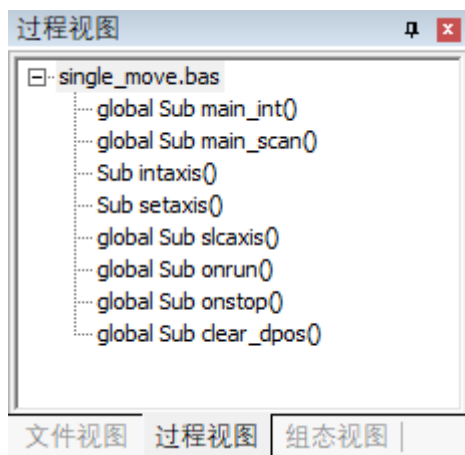
4.7 文件视图

选择软件左侧的“文件视图”，可以查看当前项目中包含的文件数量、类型和文件的任务号，双击文件可以打开文件内容并进行编辑，在文件视图上点鼠标右键，弹出文件设置窗口。



4.8 SUB 过程视图

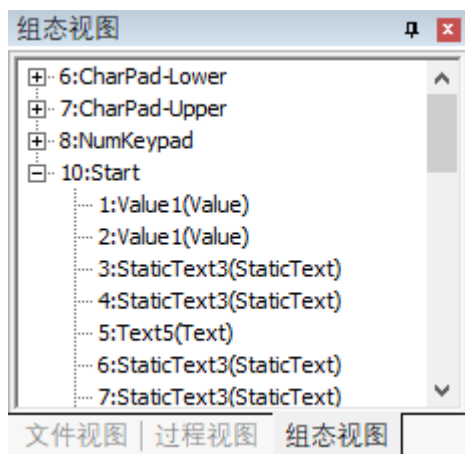
点击“过程视图”可以查看所有 Basic 文件中含有的 SUB 函数列表。
鼠标双击对应的 SUB，可以快速跳转到程序对应的定义文件和行号。



4.9 组态视图

点击“组态视图”可以查看 Hmi 文件中含有的所有窗口及每个窗口中包含的元件。

鼠标双击对应的元件可弹出元件“属性”编辑窗口。



第五章 程序下载运行

5.1 程序下载 (RAM/ROM)

程序必须下载到控制器上才能够运行，必须建立“项目”后才能下载。

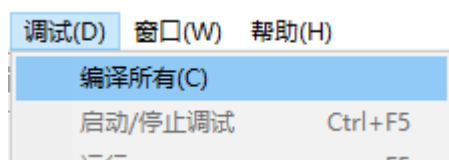


下载到 RAM: 程序项目下载到控制器的 RAM 上，下载成功立刻运行，掉电后当前下载的项目会丢失。

下载到 ROM: 程序项目下载到控制器的 FLASH 上, 下载成功立刻运行, 掉电后当前下载的项目保持。

⚠程序报错 error 时无法成功下载, ZMC0 系列部分型号不支持下载到 RAM。

文件较大的时候建议先点“编译所有”再下载到控制器, 这样下载速度会比直接下载到控制器快很多。小文件下载过程此操作可以省略。



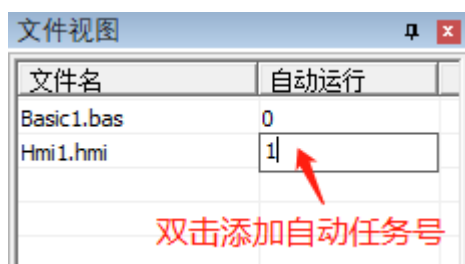
下载运行时, 可打开示波器采样运动曲线或其他数据的变化。

5.2 程序自动运行

当控制器上电启动时, 可以设定此时程序是否自动运行, 以及自动运行的任务号。通过文件列表右边的任务号进行设置, 每种型号的控制器的最大任务数量不同, 请查看“控制器状态”或对应的硬件手册。

不设置自动运行任务号程序下载后无法运行, 打印信息: WARN: no program set autorun. 一般在下载程序之前设置好自动运行任务号。

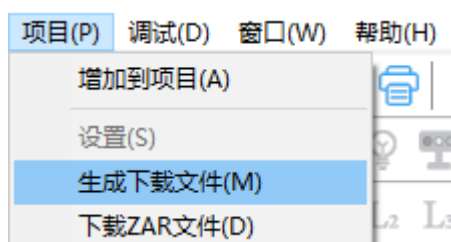
一般来说, 一个项目文件最好只设置一个自动运行任务号, 其他任务使用 RUNTASK 指令或 RUN 指令开启。



5.3 ZAR 下载

通过生成专门的 ZAR 加密下载文件, 可以实现独立的程序下载, 这样可以把下载文件传给终端客户而不用担心程序泄密, 文件后缀 .zar。还支持绑定控制器 ID (控制器 ID 是出厂时控制器的唯一序列号, 控制器 ID 可以在“控制器状态”里查看, 也可通过指令 SERIAL_NUMBER 查看), 绑定后该 ZAR 文件仅供此控制器使用。

通过菜单栏“项目”选择 ZAR 文件操作。



生成 ZAR 文件：

调试完成程序，可选择设置密码后再下载 ZAR 文件，需要使用 APP_PASS 指令将密码写入程序中校验。

若勾选绑定控制器 ID，下载 ZAR 程序时自动校验控制器 ID，ID 一致 ZAR 程序才能成功下载到控制器。

终端客户下载使用独立的工具软件“ZarDown”。使用步骤与 ZDevelop 相似。



下载 ZAR 文件：

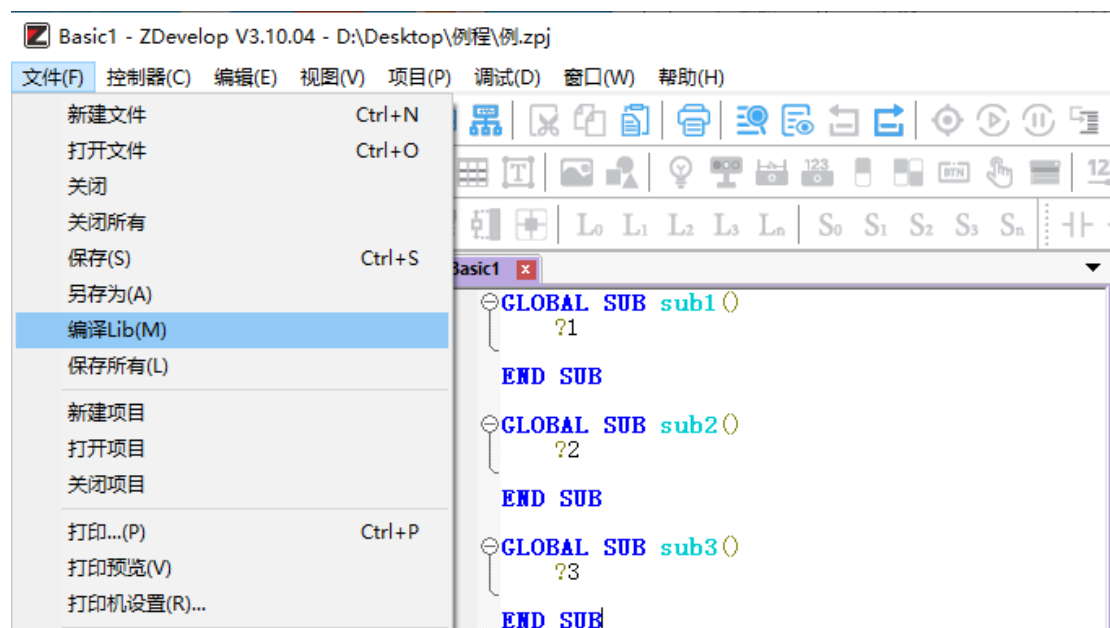
点击“下载 ZAR 文件”，弹出“打开”窗口，从系统盘里选择 ZAR 文件即可。

或者使用 FILE 指令的“LOAD_ZAR”功能加载 U 盘里的 ZAR 文件执行。

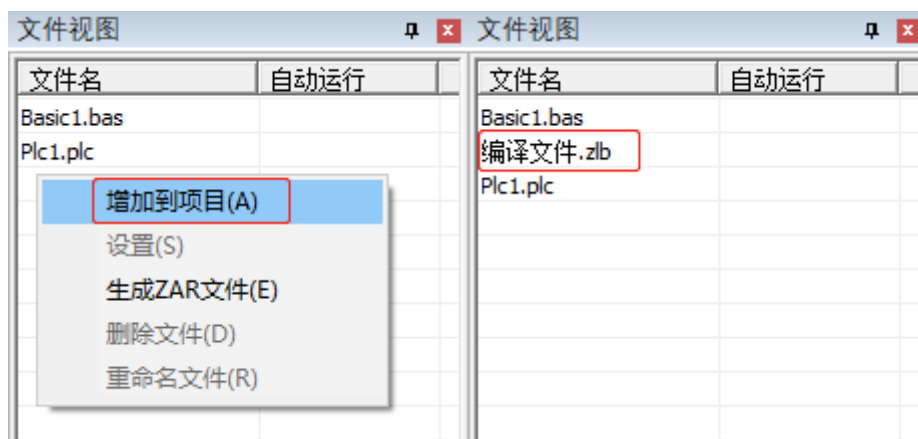
5.4 编译 Lib

此功能同 ZAR 下载一样属于程序加密的范畴，“编译为 Lib”功能可以把一个程序文件编译为一个库文件后保存，便于程序保密或防止修改，库文件的文件名后缀为“.z1b”，库文件仅能显示全局 SUB 定义。

程序调试完成，菜单栏“文件”-“编译 Lib”将所有项目文件编译成一个库文件。



编译后将 Lib 重新添加到项目中，在软件左侧的项目窗口点击鼠标右键，如下左图，选择 Lib 文件添加后如右图所示。



然后将生成 Lib 文件的源文件删除，设置自动运行任务号 0，将 Lib 文件重新下载到控制器即可。

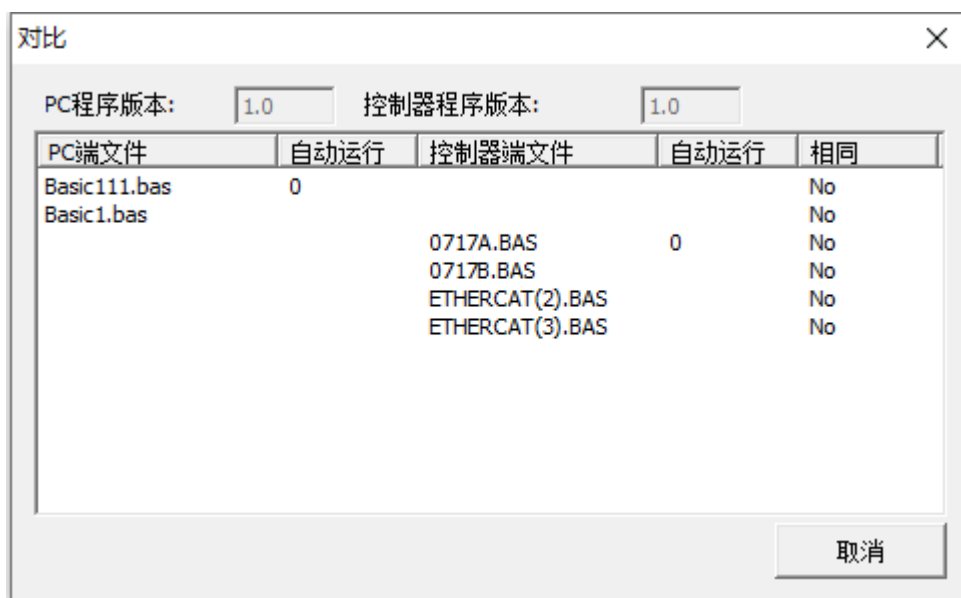


打开此时 Lib 格式的程序可以看到程序主体定义的声明，只有全局定义可以查看，但无法看到局部变量定义和 SUB 子函数具体过程，适合多人合作开发时子程序的保密。

5.5 控制器程序比较

通过“控制器”-“控制器比较”，可以比较当前项目程序和控制器程序是否一致。

左上方为软件当前程序，右下方的为控制器内保存的程序，程序相同显示“YES”，程序不同显示“NO”。



⚠ 为程序保密，控制器不支持程序上传。

第六章 控制器窗口

6.1 文件/过程/组态窗口

[文件窗口](#)、[过程窗口](#)、[组态窗口](#)参见第四章说明。

6.2 命令与输出窗口

[命令与输出](#)窗口的作用参见第二章说明。

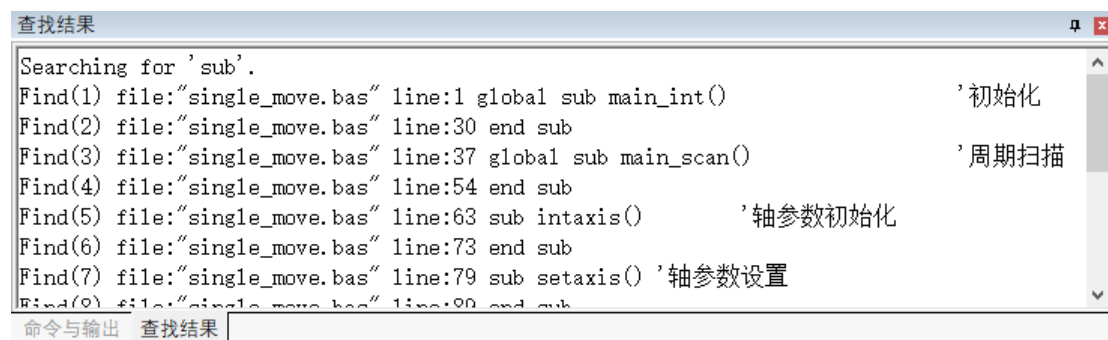
6.3 查找结果窗口

查找结果窗口用于快速查找匹配关键字,从整个项目文件中搜索匹配到的结果显示在查找结果窗口。

查找方法: 菜单栏“编辑”-“整个项目中查找”打开如下项目中搜索窗口。



搜索到的结果如下图所示,显示匹配结果所在的文件名、行号和内容,点击对应行能定位到程序位置。



6.4 轴参数窗口

参数窗口通过菜单栏“视图”-“轴参数”打开。

轴参数窗口可以监控运动控制中常见的参数。下方可变参数可以双击后直接修改,只读参数不支持修改,参数指令含义可查看 ZBasic 帮助。

轴选择	参数选择					
	轴0	轴1	轴2	轴3	轴4	轴5
ATYPE	1	1	1	1	1	1
UNITS	1	1	1	1	1	1
ACCEL	10000	10000	10000	10000	10000	10000
DECEL	10000	10000	10000	10000	10000	10000
SPEED	1000	1000	1000	1000	1000	1000
CREEP	100	100	100	100	100	100
LSPEED	0	0	0	0	0	0
MERGE	0	0	0	0	0	0
SRAMP	0	0	0	0	0	0
DPOS	0	0	0	0	0	0
MPOS	0	0	0	0	0	0
ENDMOVE	0	0	0	0	0	0
FS_LIMIT	200000000	200000000	200000000	200000000	200000000	200000000
RS_LIMIT	-200000000	-200000000	-200000000	-200000000	-200000000	-200000000
DATUM_IN	-1	-1	-1	-1	-1	-1
FWD_IN	-1	-1	-1	-1	-1	-1
REV_IN	-1	-1	-1	-1	-1	-1
IDLE	-1	-1	-1	-1	-1	-1
LOADED	-1	-1	-1	-1	-1	-1
MSPEED	0	0	0	0	0	0
MTYPE	0	0	0	0	0	0
NTYPE	0	0	0	0	0	0
REMAIN	0	0	0	0	0	0
VECTOR_BUFFERED	0	0	0	0	0	0
VP_SPEED	0	0	0	0	0	0
AXISSTATUS	0h	0h	0h	0h	0h	0h
MOVE_MARK	0	0	0	0	0	0
MOVE_CURMARK	-1	-1	-1	-1	-1	-1
AXIS_STOPREASON	0h	0h	0h	0h	0h	0h
MOVES_BUFFERED	0	0	0	0	0	0

参数解释如下：

ATYPE：轴类型设置，例如值设置为 0 表示虚轴，1 为脉冲输出，3 为正交编码输入，65 为 EtherCAT CSP 模式。

UNITS：脉冲当量，指定每单位发送的脉冲数，支持 5 位小数精度。

ACCEL/DECEL：轴加速度/轴减速度，单位为 units/s/s。当多轴运动时，轴组插补运动的加速度为主轴合成矢量加速度/主轴合成矢量减速度，不设置 DECEL 时，默认等于 ACCEL 的值。

SPEED：轴速度，单位为 units/s。当多轴运动时，作为轴组插补运动时主轴的合成矢量速度。

CREEP：轴回零时爬行速度，用于原点搜寻，单位为 units/s。

LSPEED：轴起始速度，同时用于停止速度，缺省 0，单位为 units/s。多轴运动时为轴组插补运动的合成矢量起始速度。当追求效率时，此值可设置非 0 值，但不可设置过大。

MERGE：设置 ON 时，前后缓冲的运动连接到一起而不减速，用于连续插补。

SRAMP：加减速过程 S 曲线设置，单位为 ms。多轴运动时，为轴组合成矢量曲线的时间。

DPOS：轴的期望坐标位置，或称目标位置。单位是 units。

MPOS：轴的测量反馈位置，或称实际位置。单位是 units。

ENDMOVE: 轴当前运动的最终目标绝对位置。单位是 units。

FS_LIMIT: 轴正向软限位位置，单位是 units。轴运动越界会停止并报 FSOFT。

RS_LIMIT: 轴负向软限位位置，单位是 units。轴运动越界会停止并报 RSOFT。

DATUM_IN: 原点开关对应的输入口编号，-1 无效。

FWD_IN/REV_IN: 正向/反向硬件限位开关对应的输入点编号，-1 无效。控制器限位信号生效后，会立即停止轴，停止减速度为 FAST_DEC。FAST_DEC 一般设置为 10 倍的 DECEL。

IDLE: 轴当前运动状态判断，运动中返回 0，运动结束返回-1，只读。

LOADED: 运动缓冲区没有缓冲的运动指令时返回 TURE，否则 FALSE，只读。

MSPEED: 轴的测量反馈的实际速度，单位是 units/s。MSPEED 为 MPOS 微分出来的数值，只读。

MTYPE/NTYPE: 当前正在进行运动指令类型 MTYPE/被缓冲的第一条运动指令类型 NTYPE。当插补联动时，对从轴总是返回合成矢量主轴的运动指令类型，只读。

REMAIN: 返回轴当前运动 MTYPE 还未完成的距离，单位是 units，只读。

VECTOR_BUFFERED: 返回轴当前当前运动和缓冲运动还未完成的距离，单位是 units。对多轴插补是合成矢量距离，单位是 units。

VP_SPEED: 返回轴当前运动的规划速度，单位为 units/s。当多轴运动时，轴组返回的是插补运动的合速度，不是主轴的分速度。非主轴返回的是该轴对应合成矢量速度相应的分速度，与 MSPEED 效果一致，只读。

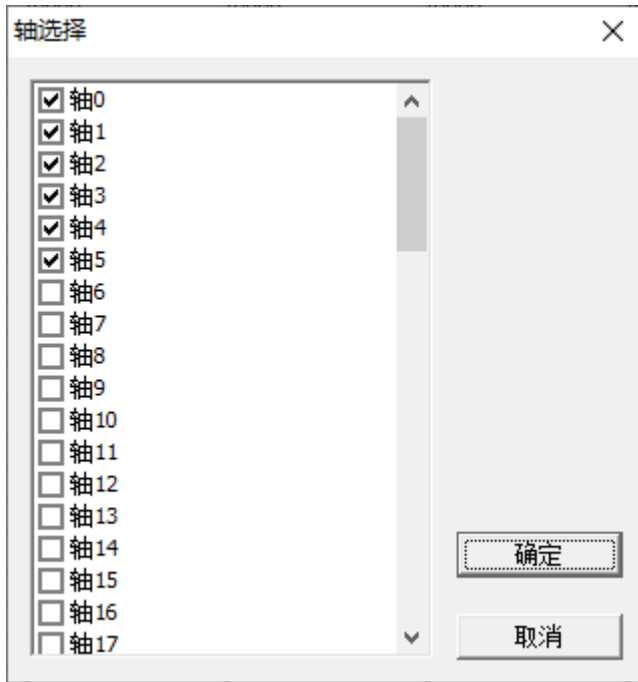
AXISSTATUS: 查看轴各种位的状态，例如：正反向硬限位，正反向软限位，轴运行状态等，只读。

MOVE_MARK: 运动指令的 MARK 标号，这个标号会和运动指令一起写入运动缓冲。每调用一条运动指令，MOVE_MARK 会自动加一。如果要强制指定 MOVE_MARK，需要每次运动前都设定一次。通过 MOVE_PAUSE 指令可以在 MARK 不同的边界处暂停。

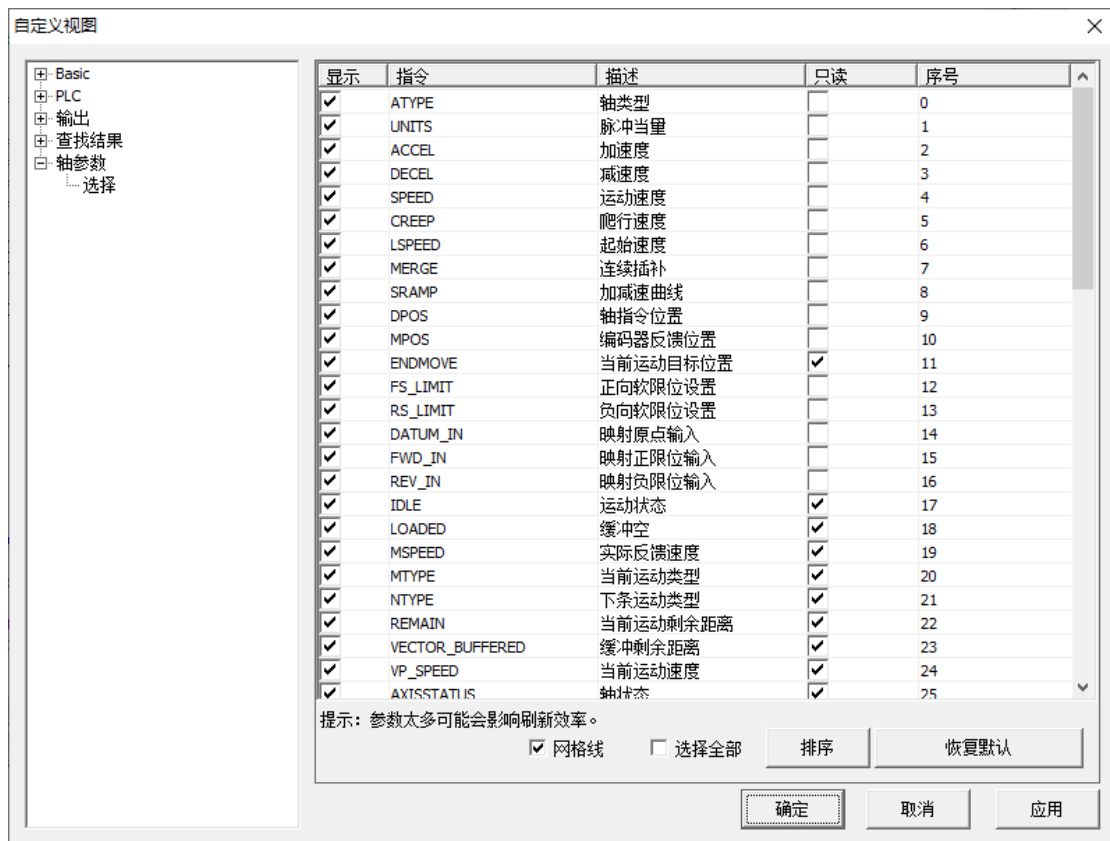
MOVE_CURMARK: 返回当前轴正在运动指令的 MOVE_MARK 标号。

更多参数参见正运动技术《ZMotion Basic 编程手册》中轴参数与轴状态指令。

点击“轴选择”可以选取需要监控的轴。



“参数选择”弹出窗口如下图所以，可以自定义需要监控显示的轴参数，参数不宜过多，否则会影响刷新效率。



6.5 手动运动窗口

手动运动窗口通过菜单栏“视图”-“手动运动”打开。


“手动运动”可通过手动操作电机，左侧轴相关的参数可实时修改，按住“左转”/“右转”按钮不放，电机持续左或右运动，松开按钮停止运动，“指令位置”显示当前 DPOS 运动距离（单位为 units）。填写“距离”参数，点击“运动”，勾选“绝对”时，电机运动到绝对距离参数位置；不勾选“绝对”时，点击“运动”，电机按相对距离参数运动。

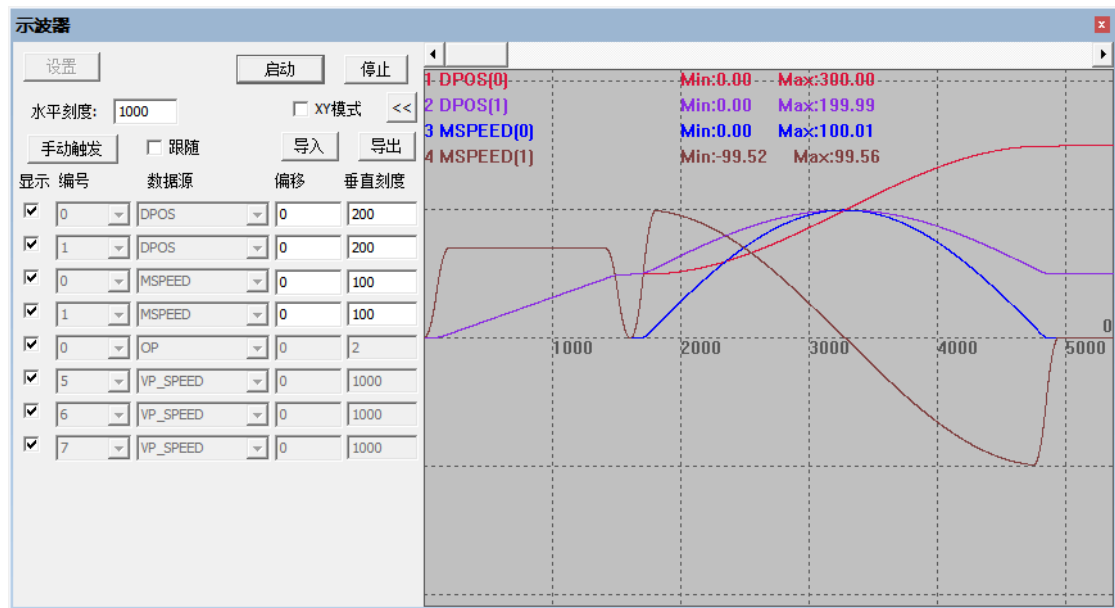
轴号可以在下拉列表中选择。“反馈位置”/“运动状态”/“轴状态”这三个参数为只读，不可修改。按下“停止”按钮轴运动立刻停下。



6.6 示波器窗口

示波器属于程序调试与运行中极其重要的一个部分，用于把肉眼看不到的信号转换成图形，便于研究各种信号变化过程。示波器利用控制器内部处理的数据，把数据显示成图形，利用示波器可以显示各种不同的信号，如轴参数、轴状态等。在“视图”→“示波器”中，

打开示波器窗口或点击快捷按钮 。



示波器必须先启动后触发才能成功采样。打开示波器，设置好相关参数之后点击“启动”，可手动触发采样，也可在程序里加入“TRIGGER”指令自动触发示波器采样。

示波器基础设置按钮功能：

设置：打开示波器设置窗口，设置示波器相关参数。

启动：启动示波器(但不触发示波器采样)。

停止：停止示波器采样。

XY 模式：勾选时切换成 XY 平面显示两个轴的插补合成轨迹。

<<：按下隐藏通道名称和峰值，只显示通道编号。

手动触发：手动触发示波器采样按钮（自动触发使用 TRIGGER 指令）。

跟随：开启跟随后，横轴自动移动到实时采样处，跟随波形显示。

显示：选择当前通道曲线是否显示。

编号：选择需要采集的数据源编号，如：轴号、数字量 IO 编号、模拟量 IO 编号、TABLE 编号、VR 编号、MODBUS 编号等。

数据源：选择采集的数据类型，下拉菜单选择，多种类型参数可选。

偏移：波形纵轴偏移量设置。

垂直刻度：纵轴一格的刻度。

水平刻度：横轴一格的刻度。

若要设置示波器参数，如轴编号、数据源以及启动示波器设置窗口，要先停止示波器再设置。

除了窗口上方进度条可以控制曲线的显示，鼠标按住左键或右键也能拖动曲线。

示波器设置窗口：

点击“设置”按钮，弹出如下所示“示波器设置”窗口。



通道数：要采样的通道总数。

深度：总共采样的数据次数，深度越大采样范围越大。

间隔：采样时间间隔，单位为系统周期，与控制器固件版本有关，一般默认 1ms，指令 SERVO_PERIOD 查看。一般来说，间隔越小，采样数据越准确，单位时间内数据量越大。

TABLE 位置：设置抓取数据存放的位置，一般默认自动使用 TABLE 数据末尾空间，也可以自定义配置，但是设置时注意不要与程序使用的 TABLE 数据区域重合。

背景颜色/通道颜色：设置背景与每个通道波形对应的颜色。

显示类型：点和线段两种曲线类型可选。线段更容易发现异常的数据。

连续采集：不开启连续采集时，到达采样深度后便停止采样，开启了连续采集之后示波器会持续采样。

导出参数：需要导出示波器数据时勾选。

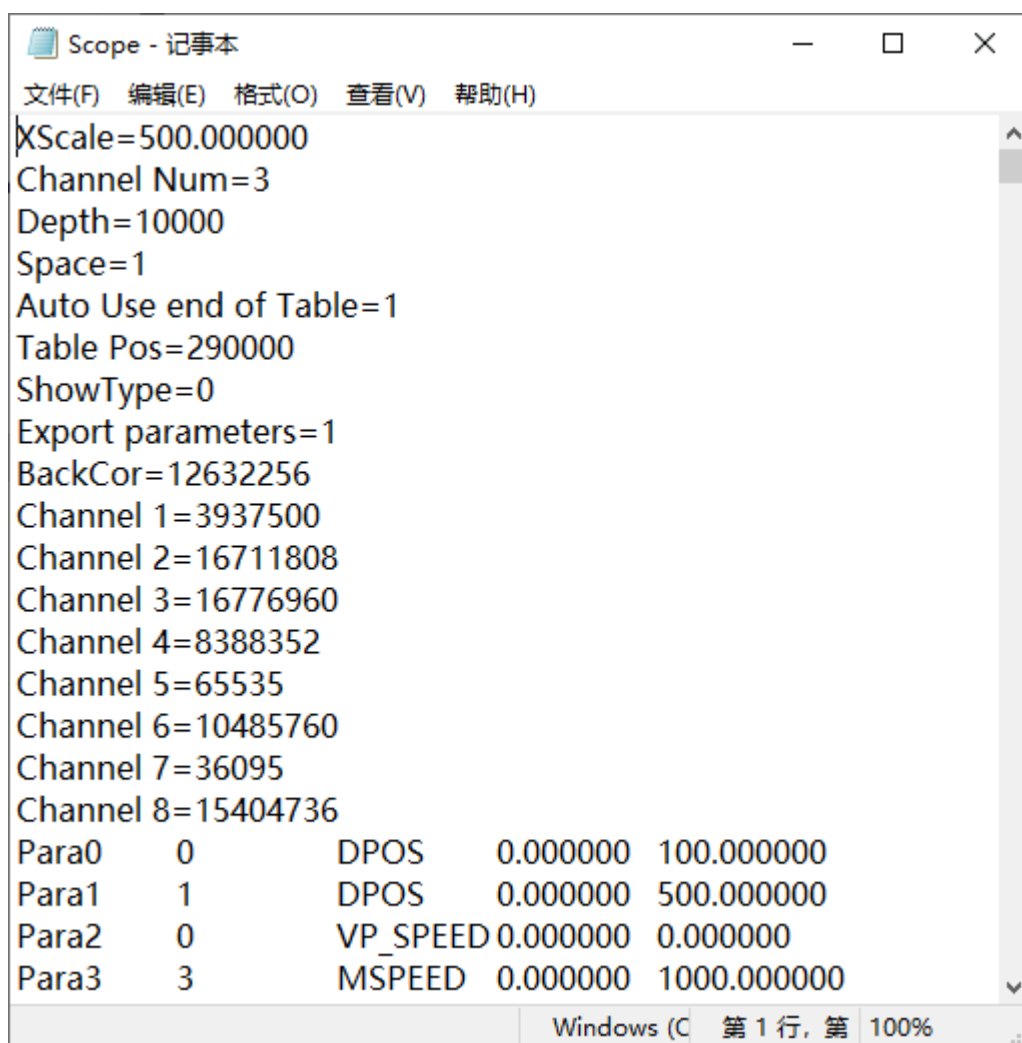
示波器数据导入导出：

导入：示波器必须在停止状态下才能导入数据，导入成功能将采样波形复现出来。

导入采样数据方法：点击“导入”，选择导入的数据文件为之前从示波器导出的文件类型后打开即可。

导出：导出参数包括示波器参数设置情况，以及各个通道的数据类型和每个采样点数据。

导出采样数据方法：先在设置里勾选“导出参数”，启动示波器采样，采样完成后点击“导出”，选择文件夹保存示波器数据，导出数据为文本文件。



示波器采样方法:

1. 打开工程项目，连接控制器或仿真器，再打开示波器窗口（操作示波器窗口之前需要连接到控制器或仿真器才可以操作）。

2. 在示波器窗口点击“设置”，选择采样通道数、采样深度、采样间隔、采样数据 TABLE 存储位置（一般来说自动使用 TABLE 数组末尾空间即可）和采样类型等，设置完成确认保存当前设置。

3. 再选择采样数据编号和数据源，点击“启动”按钮。

4. 将程序下载到控制器运行，程序里需要包含 TRIGGER 自动触发示波器采样指令，此时示波器开始采样，显示出不同数据源的波形。可调整显示刻度和波形偏移，便于观察不同波形。

5. 若波形精度不高或显示不完整，可点击“停止”按钮后再打开“设置”，调整好采样间隔和采样深度后重新执行上述采样过程。

若需要采样的时间较长，开启“连续采集”功能，此时采样时间与深度无关。

TRIGGER 指令可以非常简洁与灵活的辅助波形查看与问题查找，示波器功能对查找问

题非常有用，可以针对性的用好，进一步提高调试效率。

示波器使用注意事项：

示波器采样时间计算：

例如深度：10000，间隔：5

如果系统周期 SERVO_PERIOD=1000，也就是 1ms 轨迹规划周期，间隔 5 表示每 5ms 采集一个数据点，一共采集 10000 次数据，采集时间长度为 50s。

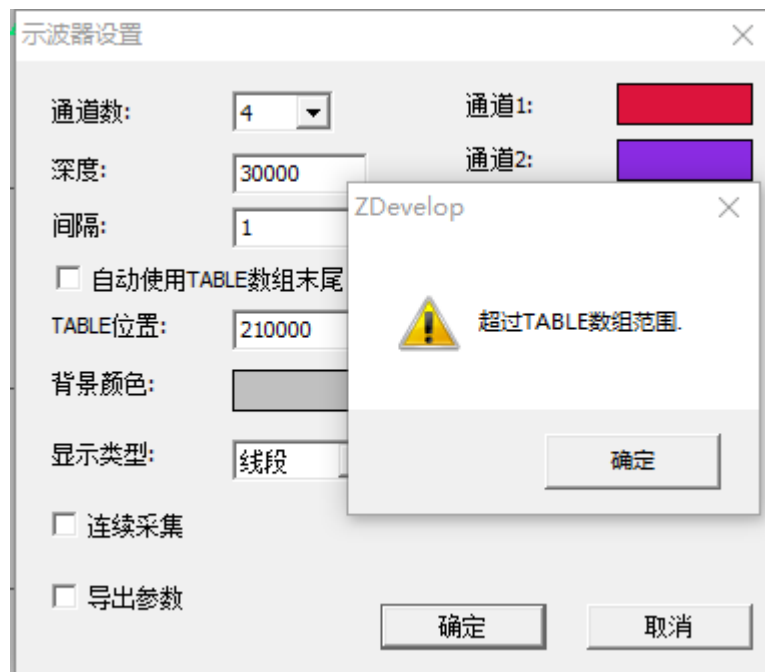
TABLE 数据末尾存储空间计算：

设置抓取数据存放的位置，一般默认自动使用 TABLE 数据末尾空间，此时根据采样数据占用空间大小自动计算起始空间地址。

计算方法：采样数据占用空间大小=通道数*深度

例：若控制器的 TABLE 空间大小为 320000，采样 4 个通道，深度为 30000，每个采样点占用一个 TABLE，所以会占用 $4*30000 = 120000$ 个 TABLE 位置， $320000-120000=200000$ ，此时 TABLE 的起始位置为 200000。

数据存放的位置也可以自定义配置，若按上面的通道数和深度，起始 TABLE 空间自定义时不能超过 200000，否则无法设置，如下图。



示波器采样数据占用的空间不要与程序使用的 TABLE 数据区域重合。

控制器 TABLE 空间大小可使用 TSIZE 指令读取、在“控制器状态”窗口查看或在线命令?*max 打印查看。

连续采集功能：

不选择连续采集时，到达采样深度后示波器自动停止采样。

在示波器“设置”里勾选连续采集，再开启示波器，示波器触发采样后会持续采样，到达采样深度后仍继续采样，忽略设置的采样深度，直到按下停止才会停止采样。

连续采集的所有波形采样数据均能导出。

示波器使用例程：

例一：连续轨迹前瞻应用

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 '启动拐角减速

DECEL_ANGLE = 15 * (PI/180) '设置开始减速角度

STOP_ANGLE = 45 * (PI/180) '设置结束减速角度

FORCE_SPEED=100 '等比减速时起作用

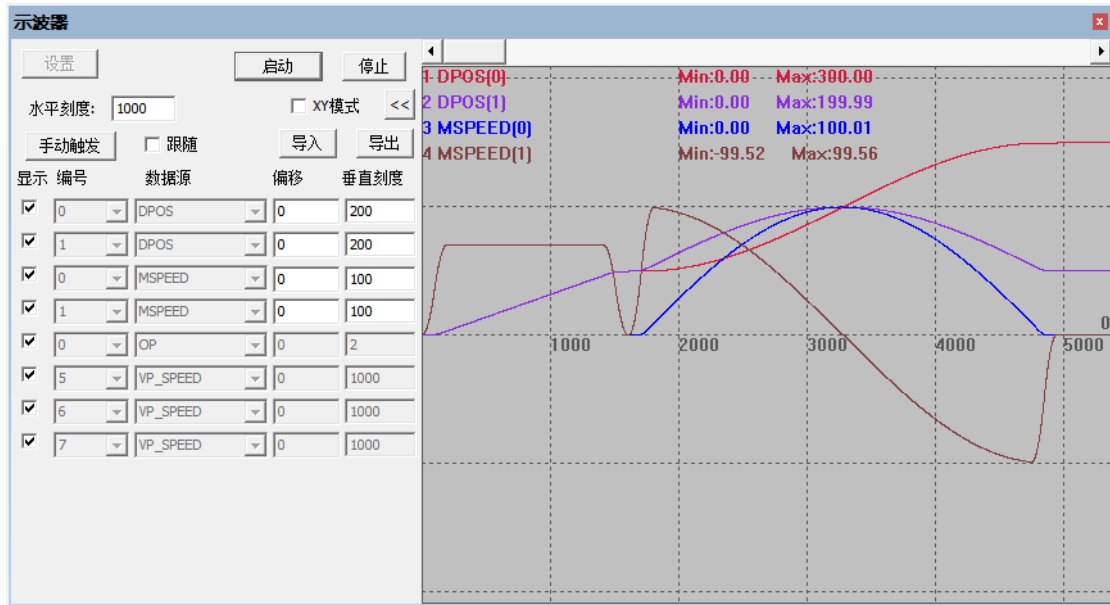
TRIGGER '自动触发示波器

MOVE (100, 100)

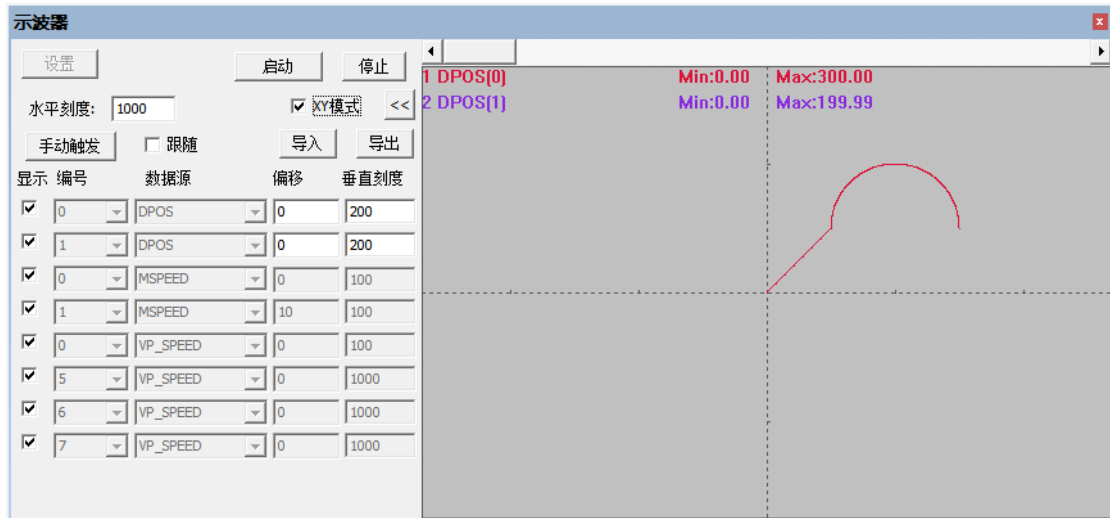
MOVECIRC (200, 0, 100, 0, 1) '半径 100 顺时针画半圆，终点坐标 (300, 100)

END

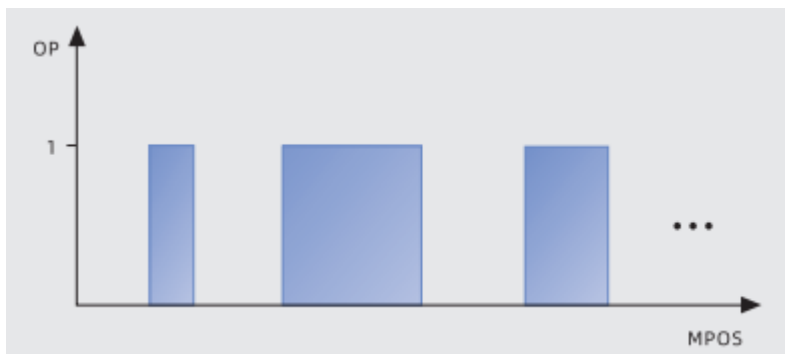
示波器采样轴 0 和轴 1 的速度和位置曲线：



XY 模式下的二轴插补合成轨迹:



例二：PSO 位置同步输出, 到达比较点输出 OP 信号。



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) ' 比较点坐标

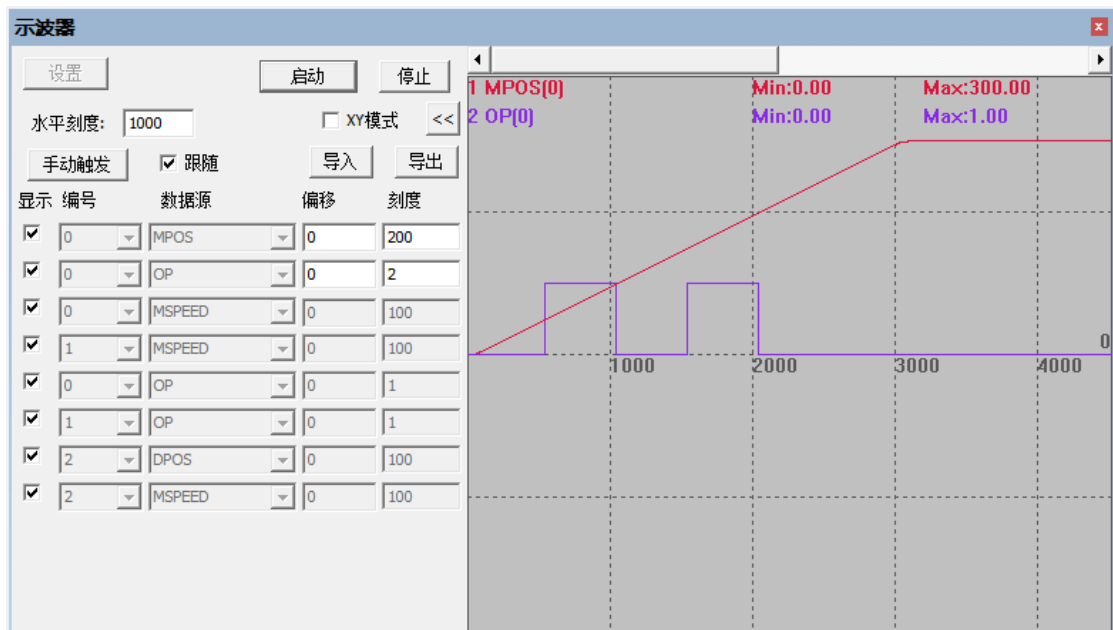
HW_PSWITCH2 (2) ' 停止并删除没有完成的比较点

HW_PSWITCH2 (1, 0, 1, 0, 3, 1) ' 比较 4 个点, 操作输出口 0

TRIGGER ' 自动触发示波器采样

MOVE (300)

END



例三：电子凸轮的应用

RAPIDSTOP (2)

WAIT IDLE (0)

BASE (0) ' 选择第 0 轴

ATYPE=1 ' 脉冲方式步进或伺服

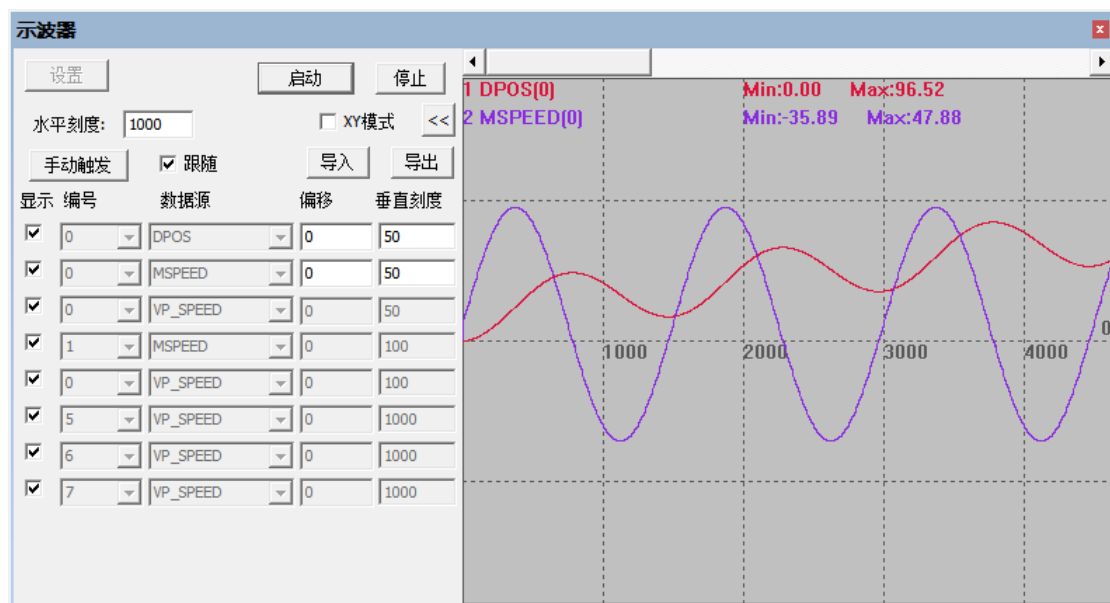
```

DPOS = 0
UNITS = 100      ' 脉冲当量
SPEED = 200
ACCEL = 2000
DECEL = 2000
' 计算 TABLE 的数据
DIM deg, rad, x, stepdeg
stepdeg = 2      ' 可以通过这个来修改段数，段数越多速度越平稳
FOR deg=0 TO 360 STEP stepdeg
    rad = deg * 2 * PI/360          ' 转换为弧度
    X = deg * 25 + 10000 * (1-COS(rad)) ' 计算每小段位移
    TABLE(deg/stepdeg, X)         ' 存储 TABLE
    TRACE deg/stepdeg, X
NEXT deg

TRIGGER          ' 触发示波器采样
WHILE 1          ' 循环运动
    CAM(0, 360/stepdeg, 0.1, 300) ' 虚拟跟踪总长度 300
    WAIT UNTIL IDLE ' 等待运动停止
WEND
END

```

运动轨迹：每个凸轮指令运动总时间=distance/speed=300/200=1.5s



6.7 输入 IO 窗口

通过菜单栏“视图”-“输入口”打开。

控制器检测到输入口状态有变化，窗口的输入状态发生相应改变。

程序中使用 INVERT_IN 指令设置使某个输入口输入反转（ZMC 系列的特殊输入定义后需要信号反转，因为 ZMC 系列是 OFF 有效，ECI 系列不需要反转）。设置了输入反转之后，“输入反转”一栏绿灯常量，无反转则是灰色，此时输入口有输入时，“输入状态”显示灰色，输入口无输入显示绿色。

“特殊输入”用来显示原点、限位、报警等特殊信号提示。

例：JOG 运动

RAPIDSTOP (2)

WAITIDLE (0)

BASE (0)

DPOS=0

ATYPE=1

UNITS=100

ACCEL=1000

DECEL=1000

CREEP=10

DATUM_IN=5 ' 输入 IN5 作为原点开关

INVERT_IN (5, ON) ' 反转 IN5 电平信号，常开信号进行反转 (ZMC 控制器)

FWD_IN=6 ' 设置正向限位开关

INVERT_IN (6, ON) ' 反转信号

REV_IN=7 ' 设置负向限位开关

INVERT_IN (7, ON) ' 反转信号

ALM_IN=8 ' 设置报警信号

INVERT_IN (8, ON) ' 反转信号

JOGSPEED=50 ' JOG 速度 50

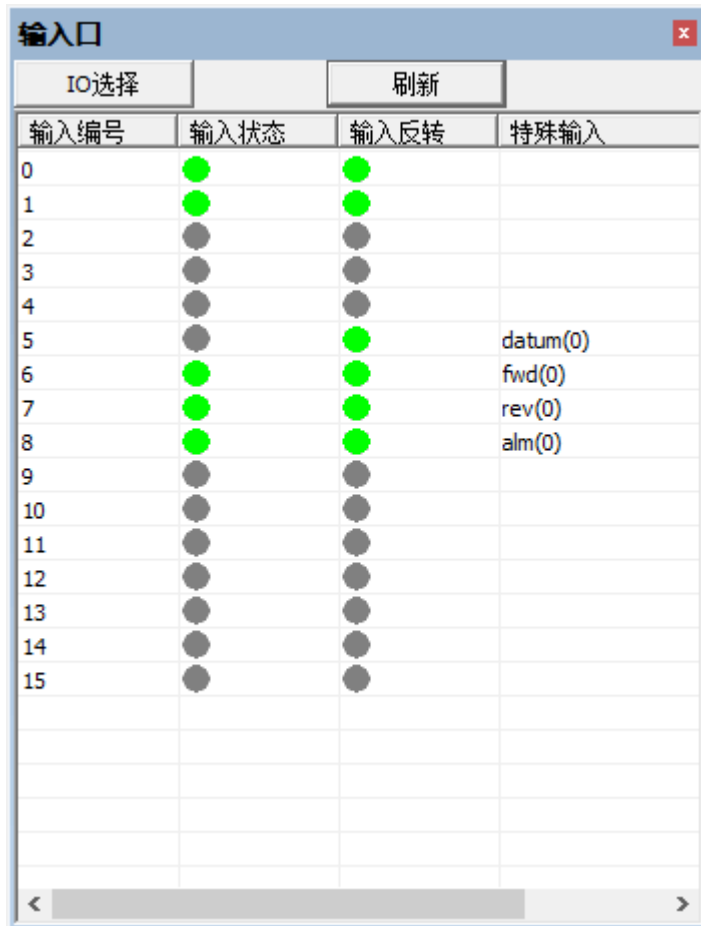
FWD_JOG=0 ’输入 IN0 作为正向 JOG 开关

REV_JOG=1 ’输入 IN1 作为负向 JOG 开关

INVERT_IN(0, ON) ’反转信号

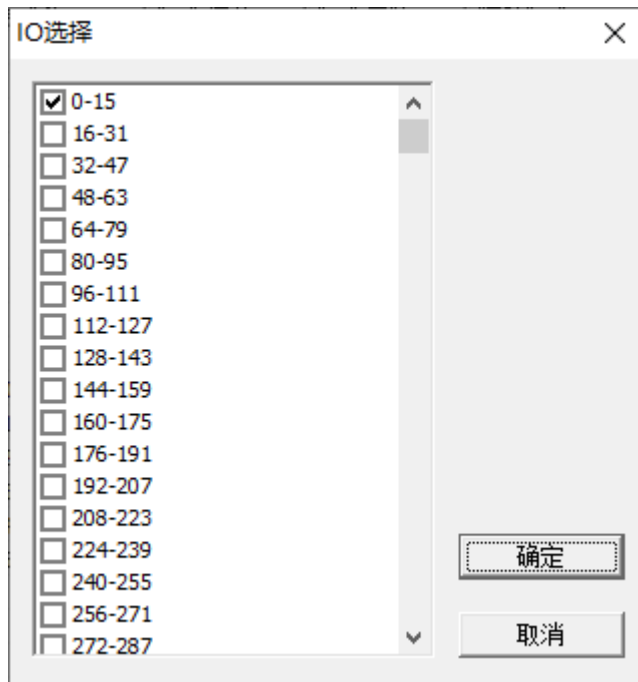
INVERT_IN(1, ON)

DATUM(3) ’回零模式



上图中，输入 IN0、IN1、IN5、IN6、IN7、IN8 均定义输入反转（ZMC 系列控制器默认 OFF 有效，故进行信号反转，ECI 系列与之相反），只有 IN5 有输入，其他口均无输入。

点击“IO 选择”可自定义显示输入 IO 口，每 16 个输入一组，如下图所示。“刷新”刷新各个输入 IO 的状态及特殊功能定义。



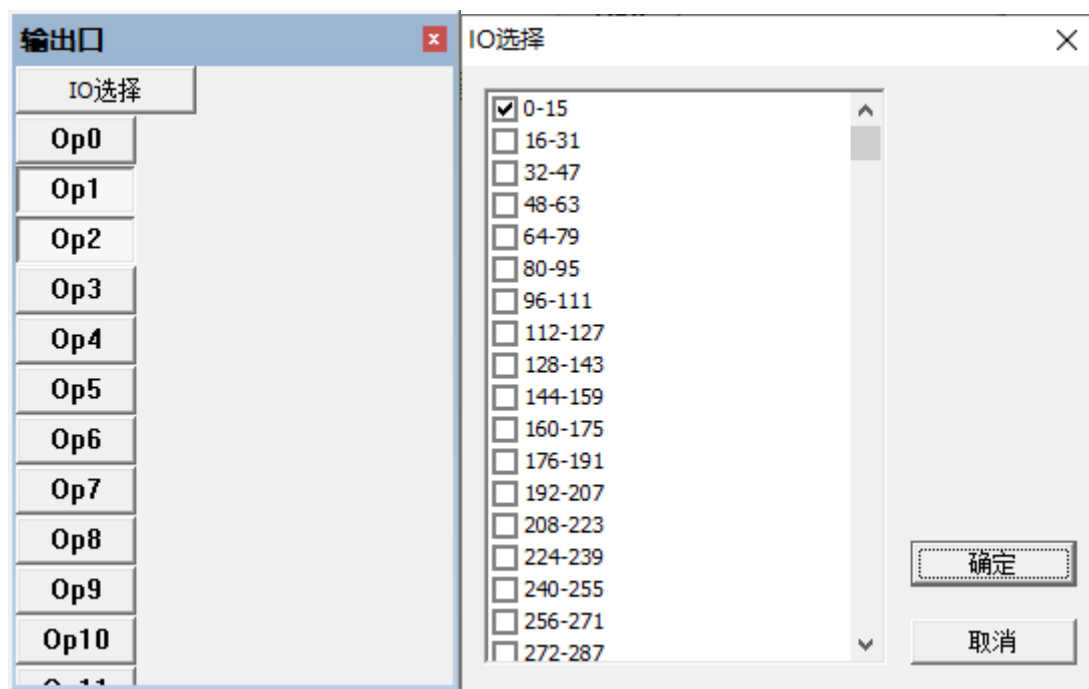
6.8 输出 IO 窗口

通过“视图”-“输出口”打开。

按下按钮能操作 OP 口输出，如下图，OP1 口和 OP2 口开启，其他口关闭。

连接到仿真器时，输出 IO 口状态在仿真器窗口输出监控处也能查看，不过仅能显示编号 0-11 的状态。

“IO 选择”选择要显示的输出，每 16 个输出一组。直接点击输出口可以切换其输出状态。



6.9 任务/监视窗口

任务与监视窗口属于程序调试的部分，在调试时打开这两个窗口查看任务的运行情况和增加监视项目，[任务](#)和[监视](#)窗口的用法参见第七章的说明。

6.10 寄存器窗口

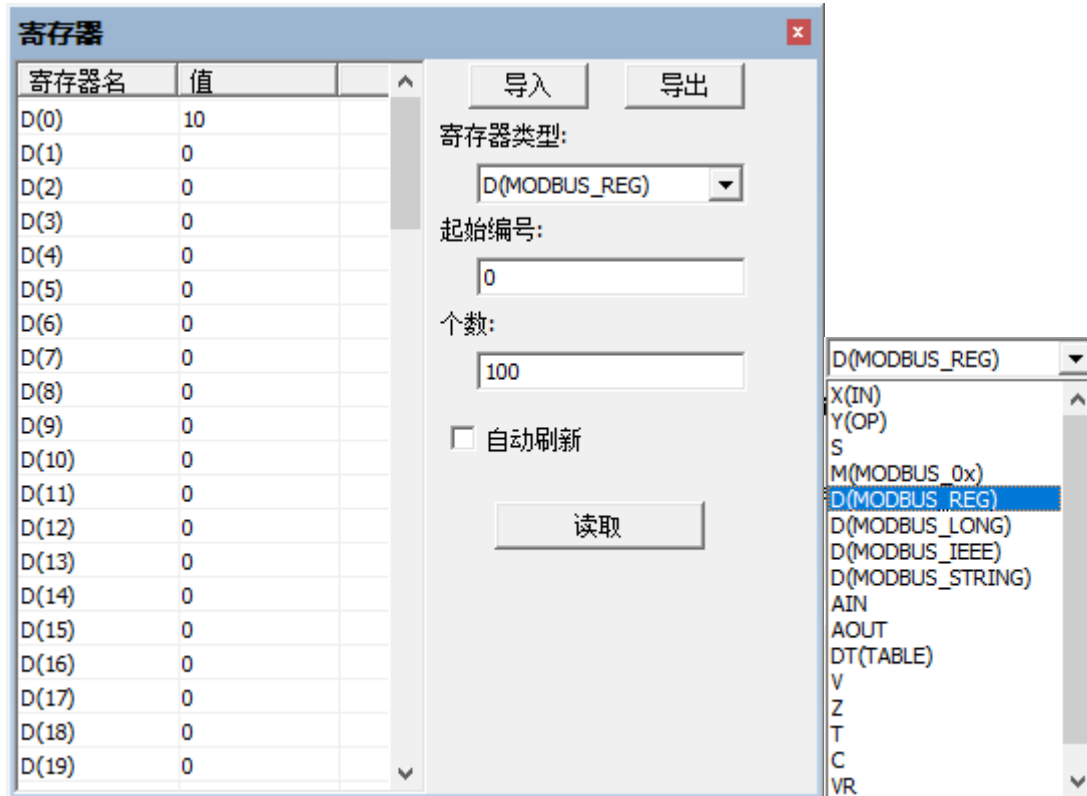
通过菜单栏“视图”-“寄存器”打开。

通过这个窗口可以批量查看控制器寄存器的数值，可以选择查看不同类型的寄存器（支持 PLC 功能的控制器才支持此功能）。

使用方法：选择要读取的寄存器的类型、起始编号、个数后点击“读取”即可在窗口显示出数据，注意读取个数不要超出寄存器范围，否则会提示错误，勾选自动刷新功能便于自动实时采集寄存器值变化并显示出来，否则需要再次点击读取才能获取的寄存器的值变化情况。点击“导入”/“导出”可快速上传/下载寄存器数据。

导出数据方便客户把自己关注的部分寄存器导出到文本保存。

导入数据方便客户把关注部分的已保存的数据直接更改到控制器内部。



寄存器类型:

X(IN): 输入继电器, 由外部开关信号驱动, 对应 IN

Y(OP): 输出继电器, 能直接驱动外部负载, 对应 OP

S: 状态继电器, 用于对工序步进控制

M: 辅助继电器, 不能直接驱动外部负载, 对应 MODBUS_BIT

D(MODBUS_REG): 16 位整型数据寄存器, MODBUS 区域数据, 对应 MODBUS_REG

D(MODBUS_LONG): 32 位整型数据寄存器, MODBUS 区域数据, 对应 MODBUS_LONG

D(MODBUS_IEEE): 32 位浮点型数据寄存器, MODBUS 区域数据, 对应 MODBUS_IEEE

D(MODBUS_STRING): : 1 字节字符串数据寄存器, MODBUS 区域数据, 对应 MODBUS_STRING

AIN: 模拟量输入

AOUT: 模拟量输出

DT(TABLE): 浮点寄存器, 长度 32 位, 对应 TABLE

V: 变址寄存器, 长度 16 位。

Z: 变址寄存器, , 长度 16 位。

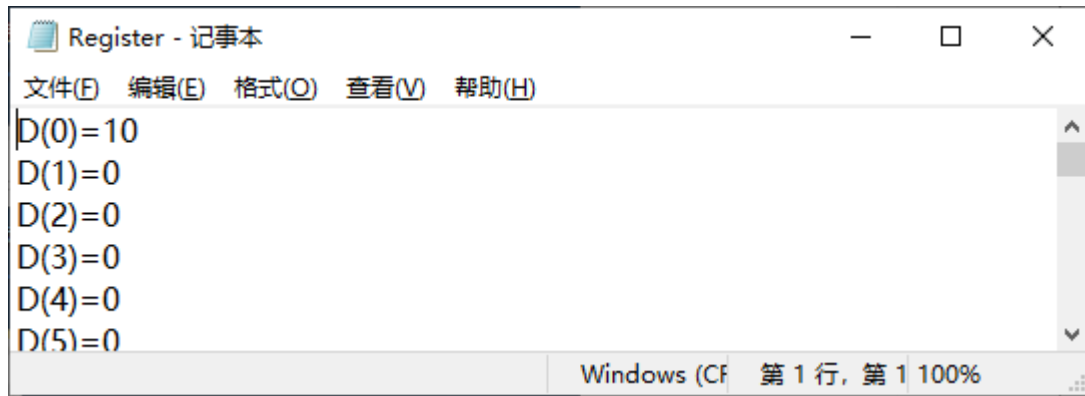
T: 定时器, 单位为 1ms。

C: 计数器。

VR: 掉电保存寄存器, 32 位浮点型。

VR_INT: 掉电保存寄存器, 32 位整型。

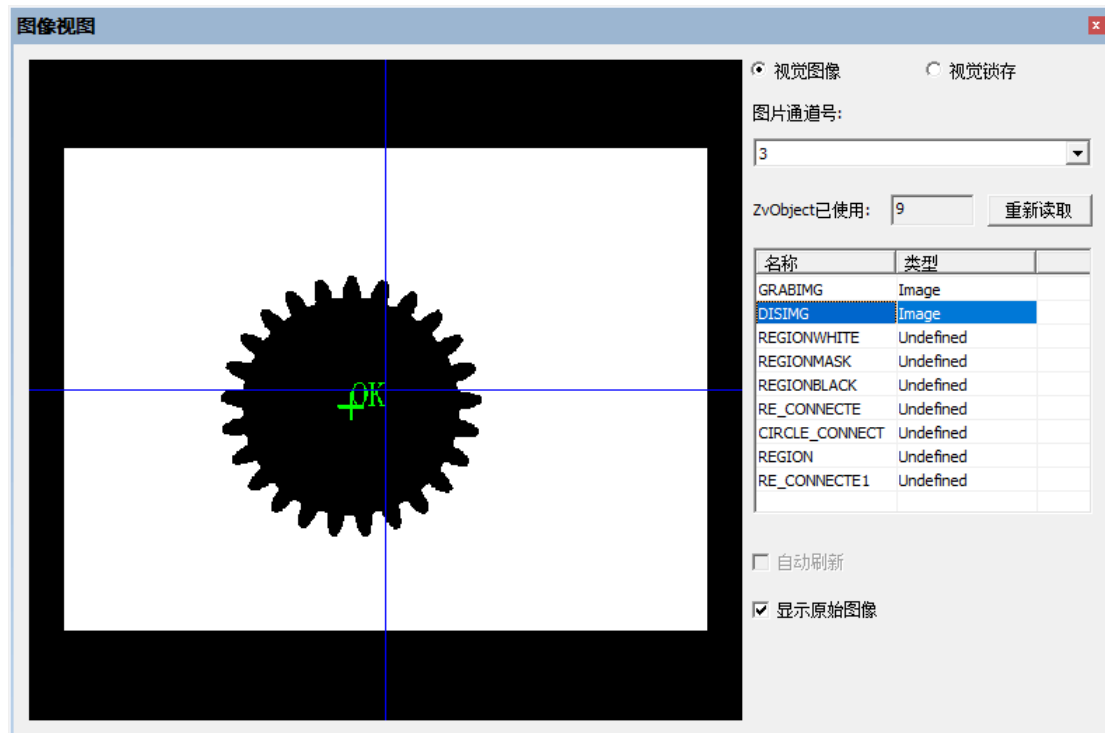
导出数据示例:



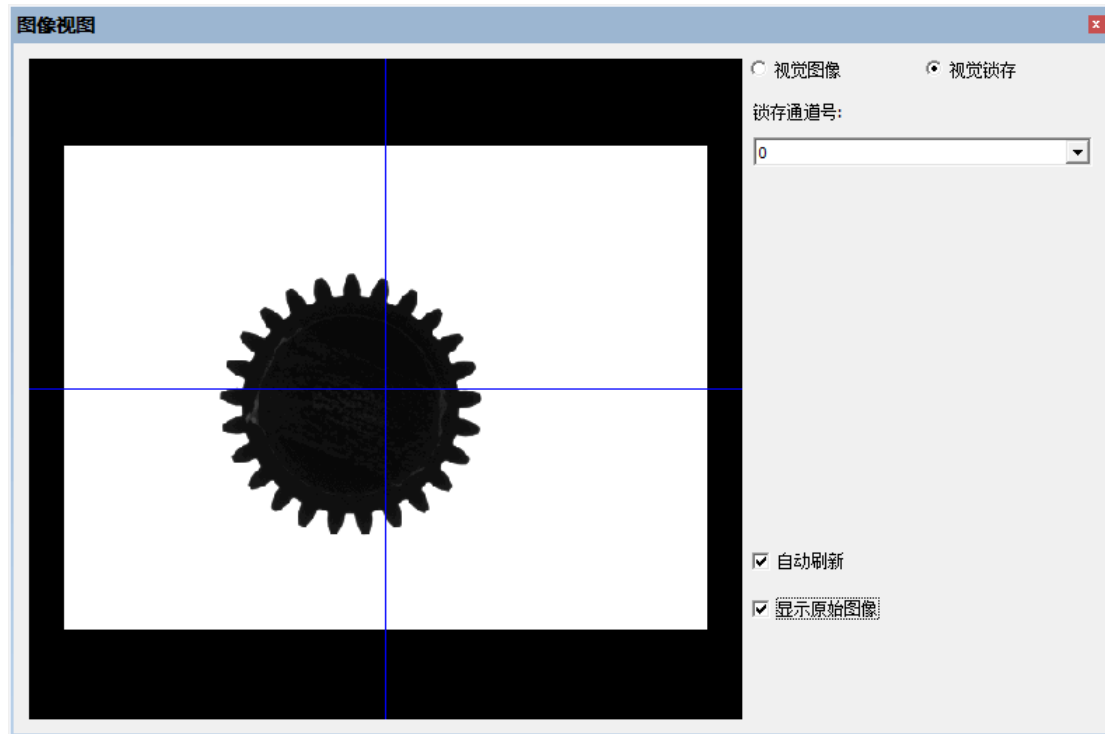
6.11 图像窗口

通过菜单栏“视图”-“寄存器”打开。在机器视觉开发环境下使用此窗口查看视觉图像。必须先采集到图片之后图像窗口才能正常使用，可在全部视觉图像显示和视觉锁存通道图像显示之间切换。

视觉图像可选择当前项目下的所有图片显示，包括采集到的图像和处理的图像。



视觉锁存的图像显示前需要先选择锁存通道号，再显示当前锁存通道的图像，当前锁存通道为空时，没有图像。



6.12 AD/DA 窗口

通过菜单栏“视图”-“AD/DA”打开。

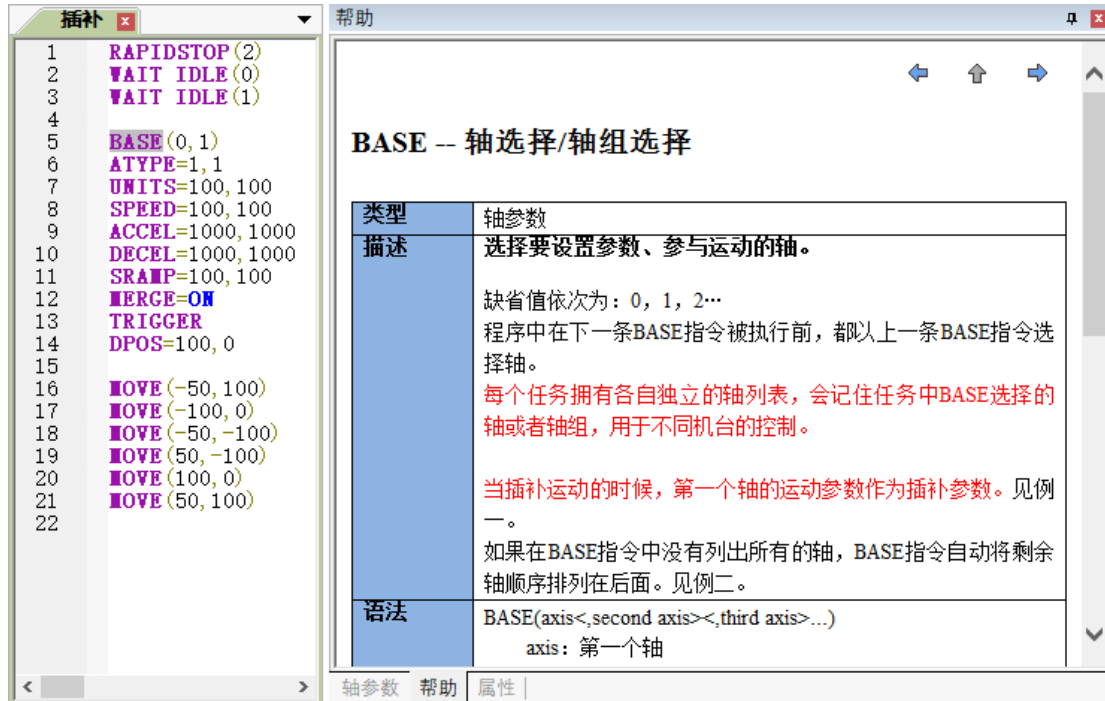
方便查看 AIN 和 AOUT 通道情况。

AD/DA						
控制器类型:	ZMC306					重新读取
AD:						
通道号	大小	刻度值	电压值(V)	最大刻度值	电压范围(V)	
0	2%	90	0.110	0	0.000~5.000	
1	2%	94	0.115	0	0.000~5.000	
2	2%	88	0.107	0	0.000~5.000	
3	2%	89	0.109	0	0.000~5.000	
DA:						
通道号	大小	刻度值	电压值(V)	最大刻度值	电压范围(V)	
0	0%	19	0.024	4095	0.000~5.000	
1	0%	0	0.000	4095	0.000~5.000	

6.13 帮助窗口

通过菜单栏“视图”-“帮助”打开。

选择要查看的指令，按下 F1 或点击指令，快速打开指令帮助文档。



6.14 注释窗口

通过菜单栏“视图”-“注释”打开。

系统注释：查看系统已经注释的寄存器的作用，项目注释：自行添加寄存器和对应的批注内容。



6.15 交叉参数表

通过菜单栏“视图”-“交叉参数表”打开。

适用于 PLC 编程方式，“交叉参照表”查看已使用的寄存器的具体信息，双击快速跳转到寄存器在程序中的对应窗格。



6.16 寄存器使用列表

通过菜单栏“视图”-“寄存器使用列表”打开。

适用于 PLC 编程方式“寄存器使用列表”便于查看寄存器的使用情况，查看哪些寄存器用了，哪些没用，在后续的程序中好规划。

在左侧灰色区域双击鼠标左键，即可弹出该寄存器的“交叉参照表”视图。

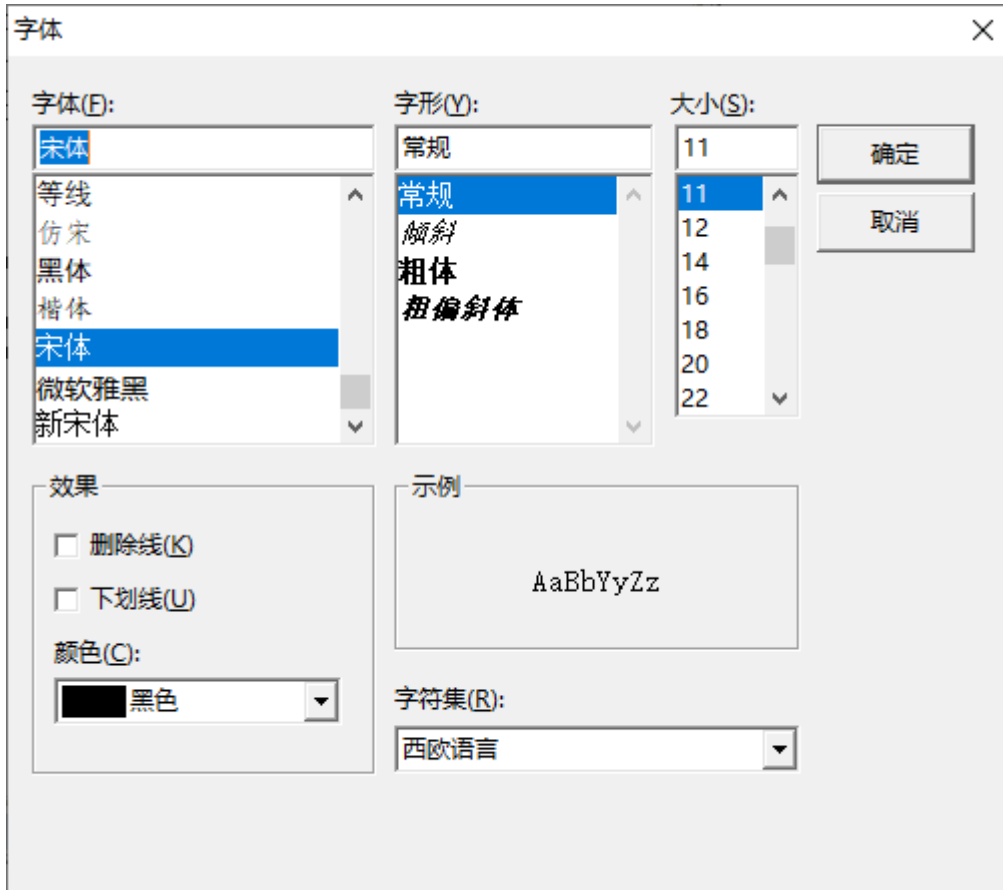
在右侧注释空白行双击鼠标左键，可对该寄存器进行注释。



6.17 字体视图

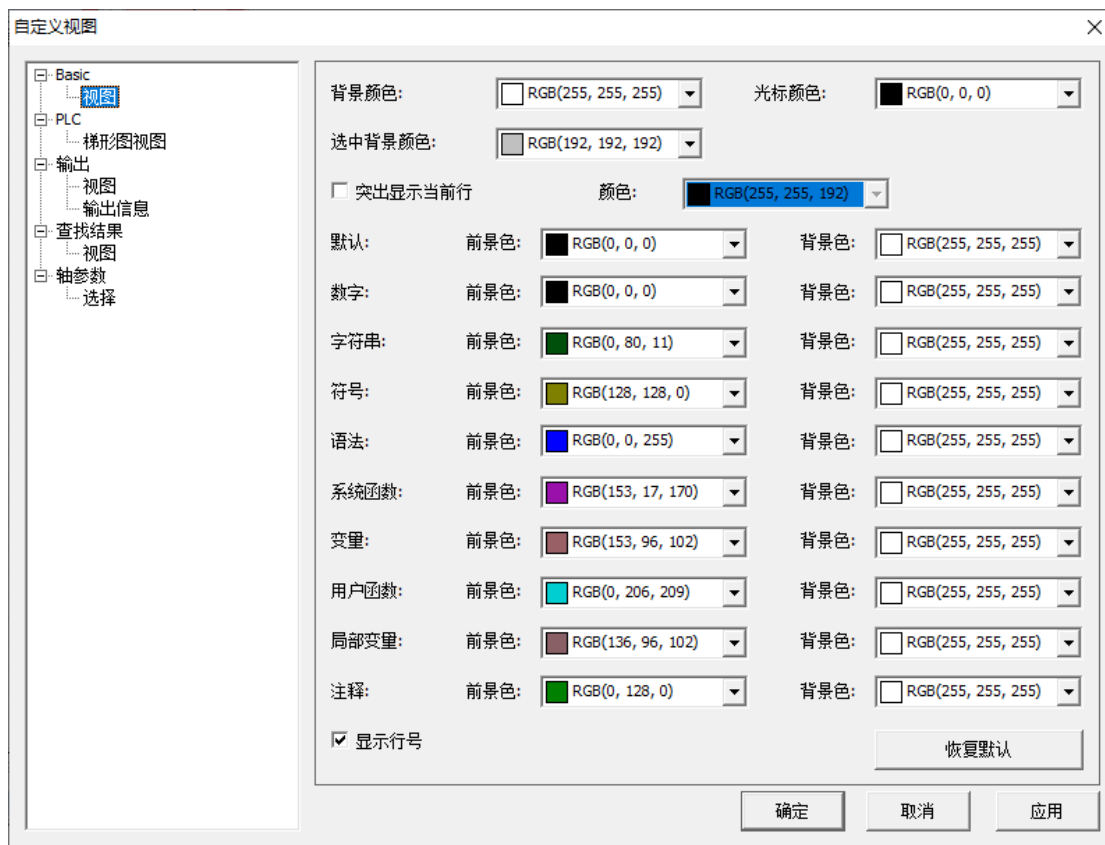
通过菜单栏“视图”-“字体”打开。

用户可更改默认的字體、大小和颜色等。



6.18 自定义视图

自定义视图用于对编程界面显示风格自定义设置，其中轴参数用以选择“轴参数”视图显示参数情况。

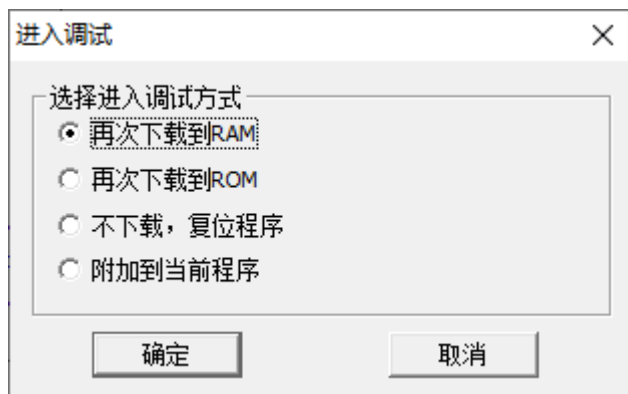


第七章 程序调试

7.1 进入调试

通过菜单栏“调试”-“启动/停止调试”进入程序调试状态。

调试功能可以追踪程序运行，从调试菜单进入调试模式，可以选择不同的进入方式。当需要查看当前控制器程序的运行状态时，请选择附加到当前程序。



再次下载到 RAM：表示程序再次下载到 RAM 运行，RAM 掉电不保存。

再次下载到 ROM：表示程序再次下载到 ROM 运行，ROM 掉电保存。

不下载，复位程序：表示不下载程序，重新运行之前下载的程序，并打开任务窗口显示目前的运行状态。

附加到当前程序：表示此时程序不下载，仅打开任务窗口显示目前的运行状态。

当程序运动出错后，ZDevelop 软件会显示出错信息，双击出错信息可以自动切换到程序出错位置，如果出错信息没有看到，可以通过命令行输入?*task 再次查看出错信息，或打开“故障诊断”窗口。

以如下程序为例说明：

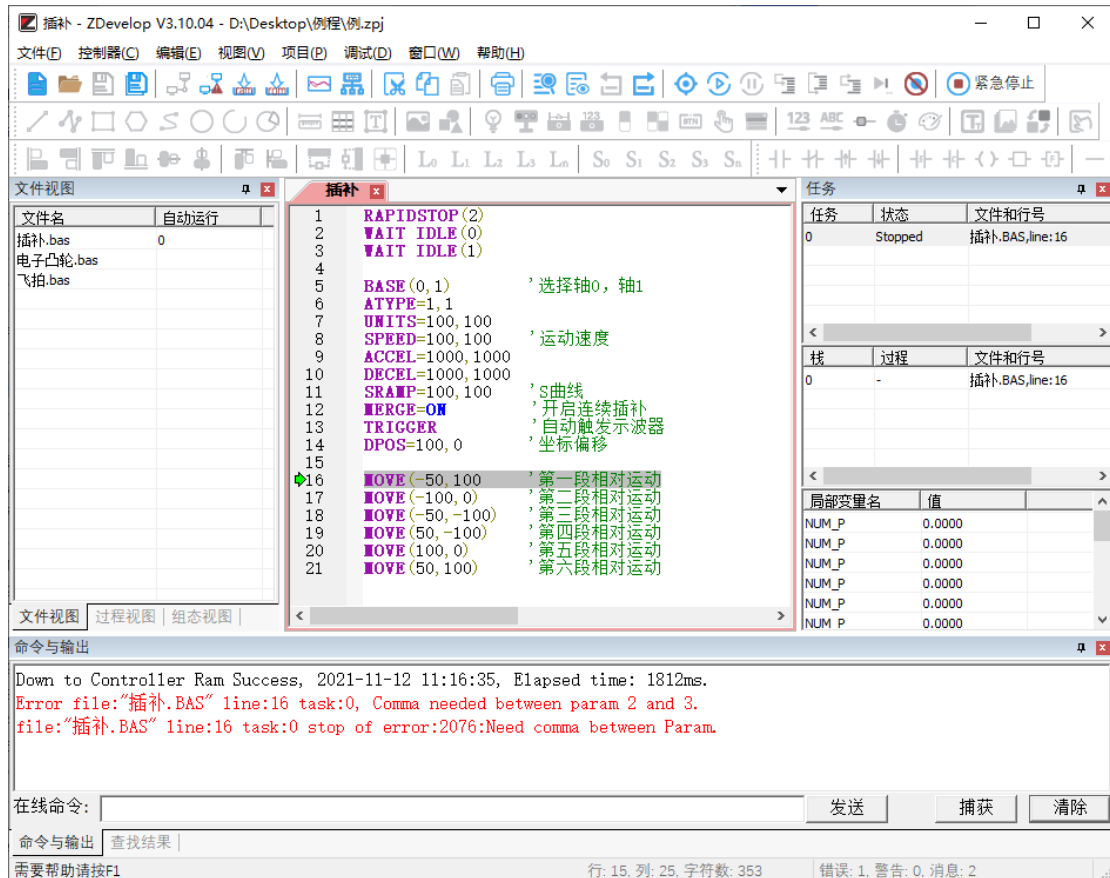
```
RAPIDSTOP (2)
WAIT IDLE (0)
WAIT IDLE (1)

BASE (0, 1)      ' 选择轴 0, 轴 1
ATYPE=1, 1
UNITS=100, 100
SPEED=100, 100  ' 运动速度
ACCEL=1000, 1000
DECEL=1000, 1000
SRAMP=100, 100  ' S 曲线
MERGE=ON        ' 开启连续插补
TRIGGER         ' 自动触发示波器
DPOS=100, 0     ' 坐标偏移

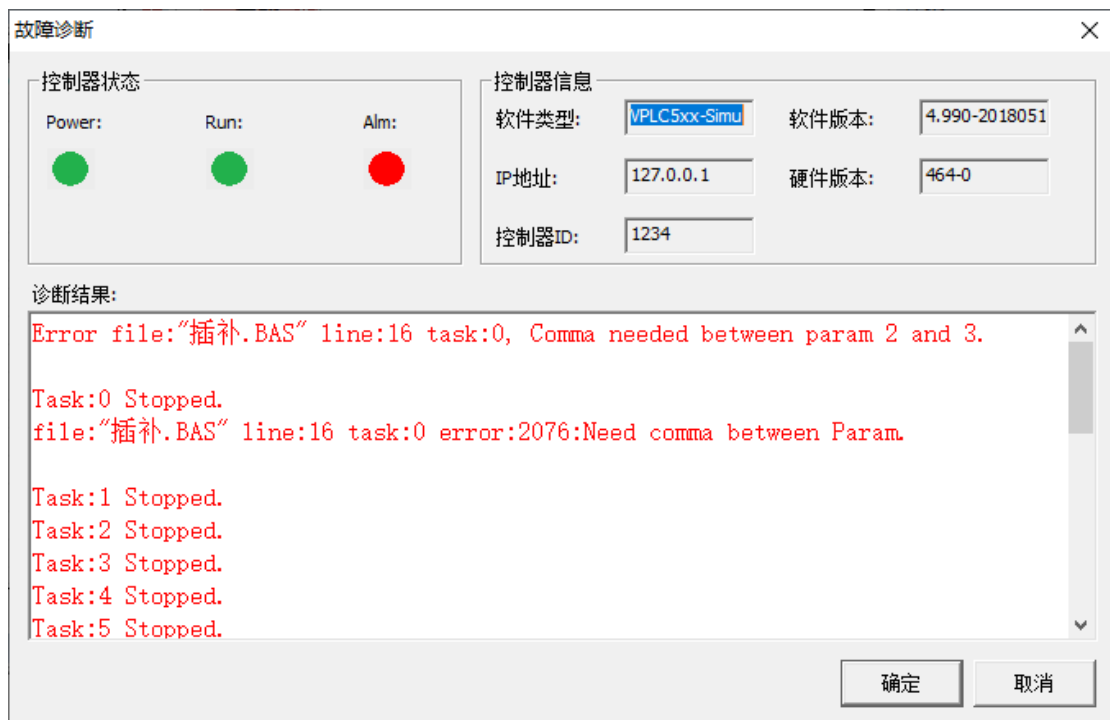
MOVE (-50, 100  ' 第一段相对运动
MOVE (-100, 0)  ' 第二段相对运动
MOVE (-50, -100) ' 第三段相对运动
MOVE (50, -100) ' 第四段相对运动
MOVE (100, 0)   ' 第五段相对运动
MOVE (50, 100)  ' 第六段相对运动

END
```

命令与输出窗口打印插补文件的第十六行语法错误，报 error 错误程序无法成功下载到控制器，解决问题之后再次点击下载程序。



故障诊断窗口也能查询程序出错的信息。



7.2 调试工具栏

开启调试时，调试工具栏有效。



复位：从起始处开始运行

运行：继续/暂停运行

暂停：

单步进入：运行到程序里面

单步跳过：运行到下一条程序

单步跳出：跳出 SUB 子程序运行

运行到：运行到指定行

增加或删除断点：按一下增加断点，在原位置再按一下删除断点



紧急停止：紧急停止程序和所有轴的运动

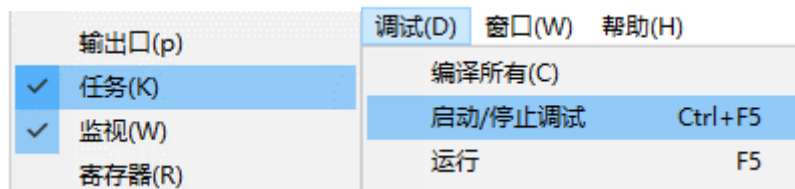
⚠ 当程序与控制器不一致或是对程序进行再修改后没有及时下载，会导致调试指定的行号产生偏移。

⚠ 程序暂停时当前已经提交的运动并不会暂停。

7.3 任务状态窗口

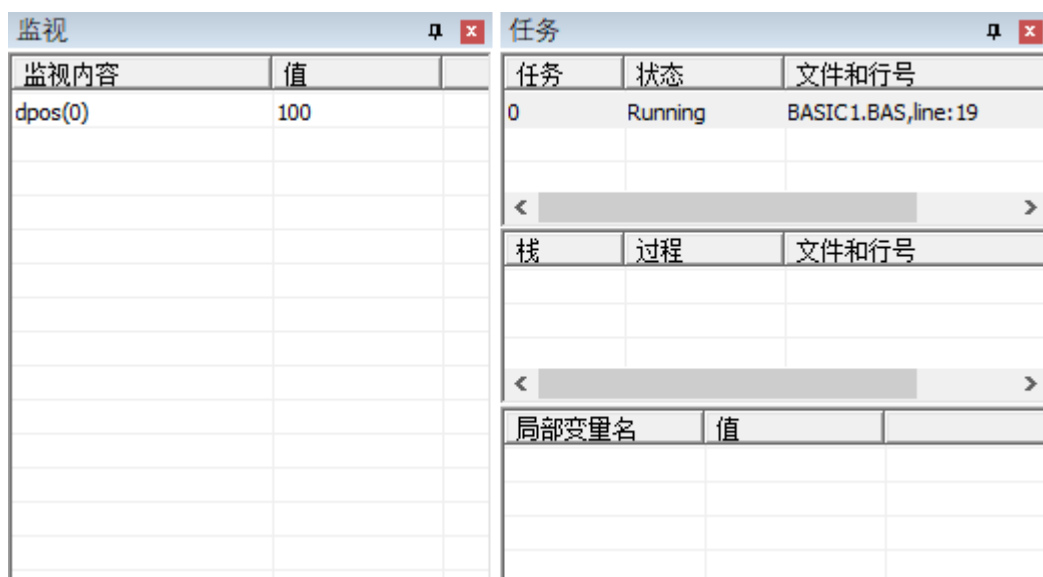
任务状态窗口可以在菜单栏“视图”-“任务”中开启关闭，或点击菜单栏“调试”-“启动/停止调试”调用任务与监视窗口。

注意只有在调试状态下才能打开任务与监控窗口。



控制器支持多任务运行，程序运行时可以查看各个任务的详细状态。下图示例为单个任务，最大支持任务数在“控制器状态”窗口查看 Taskes 参数。

控制器支持多任务运行，启动调试后进入调试状态，通过菜单栏“视图”-“任务”可以查看各个任务的详细状态。下图中示例为单个任务任务号 0，状态 Running 正在运行，执行的文件为 Basic1.BAS，执行的行号为 line:19。



栈：当程序调用 SUB 过程时，会自动将原来的状态和局部变量存储起来，称之为栈。

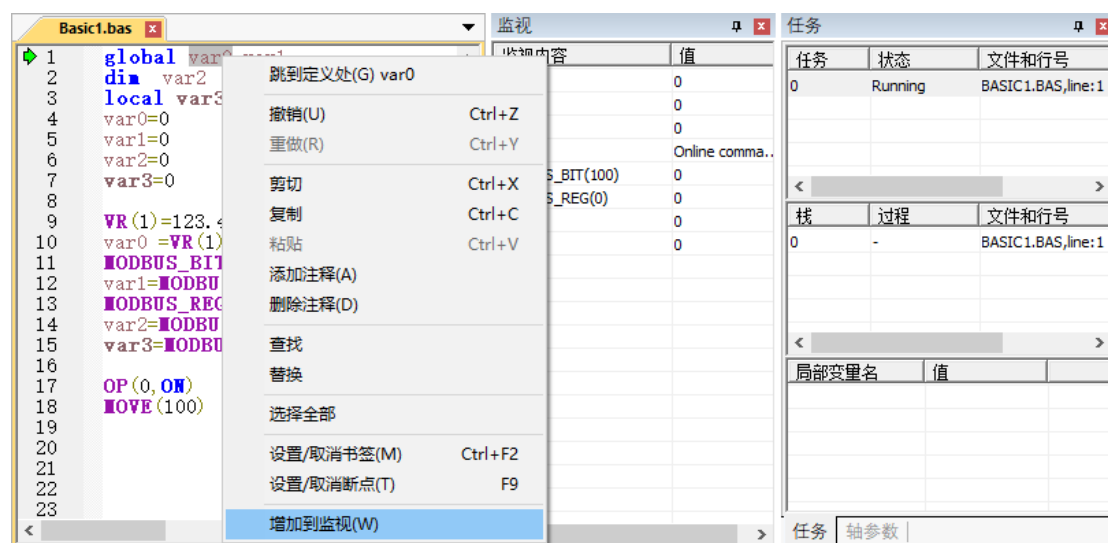
局部变量：包括 LOCAL 变量定义，以及 SUB 调用传入的参数。

⚠ 同一任务的不同栈的局部变量是不一样的，就算名称一样。

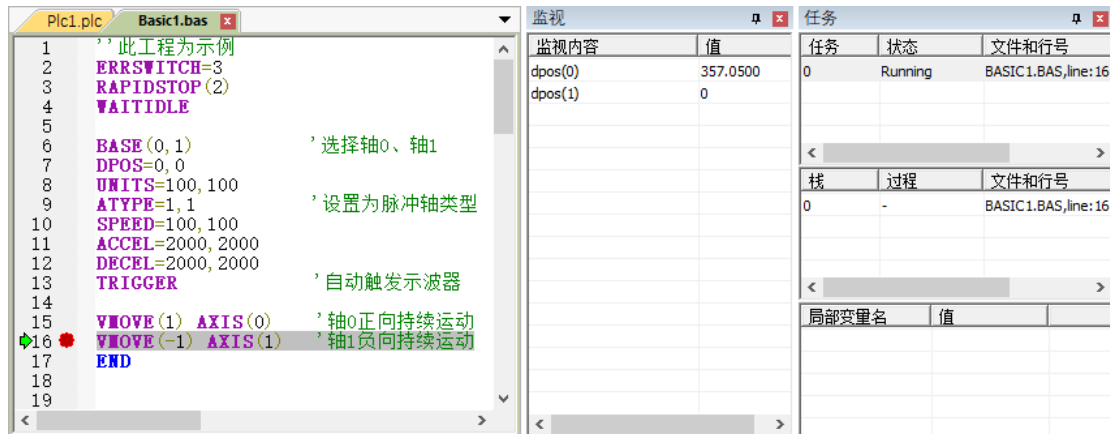
⚠ 栈层次有限，子程序调用的堆栈层数一般为 8 层，请留意递归的使用。

7.4 监视窗口

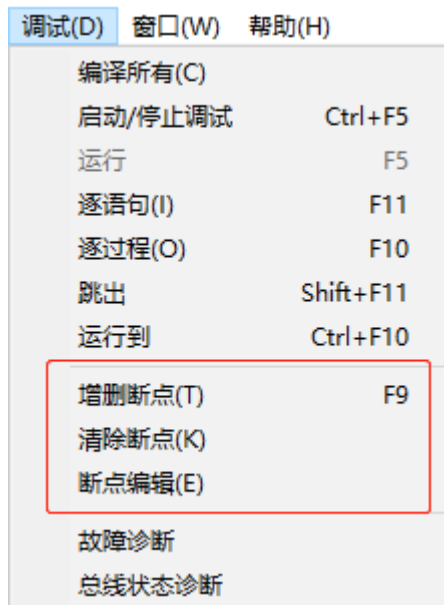
可以把全局变量和文件变量等有效表达式加入到监视窗口，程序运行时自动获取参数值显示出来。也可以在调试状态下，在程序编辑区域选择变量后点击右键“增加到监视”加入到监视内容，通过双击监视内容名称来修改或增加监视项。



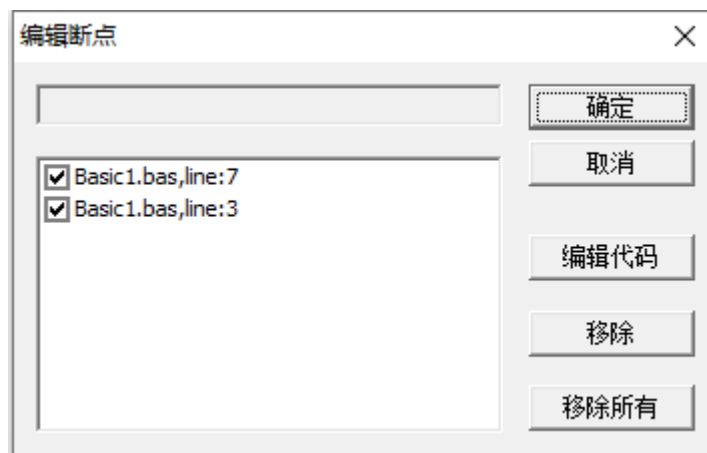
⚠ 不支持 LOCAL 局部变量的监视，需要通过任务窗口查看。如下图，因为 var3 是局部



程序停止在断点处后，就可以进行逐步调试，快捷键 F11，按一次程序向下执行一步。如果断点是设置在循环中，那么下次循环运行到断点处时还是会停止程序。



通过断点编辑窗口可以对当前所有的断点进行编辑。双击断点信息跳转到断点行，可以选择移除一个或多个断点，移除后点击确认生效。



程序调试完成后，需要清除所有断点才能下载到控制器运行，断点不清除就下载程序到控制器，命令与输出区域打印如下警告信息：Warn file:"Basic1.BAS" line:11 task:0,

Paused.

7.6 故障诊断

菜单栏“调试”-“故障诊断”，方便查看控制器软/硬件型号、日期、版本号、任务运行状态以及错误信息提示，方便客户现场查看故障以及程序停在哪里。

Run 灯、Alm 也可以手动开关，可以在众多控制器里快速找到当前连接的控制器。

任务运行状态：



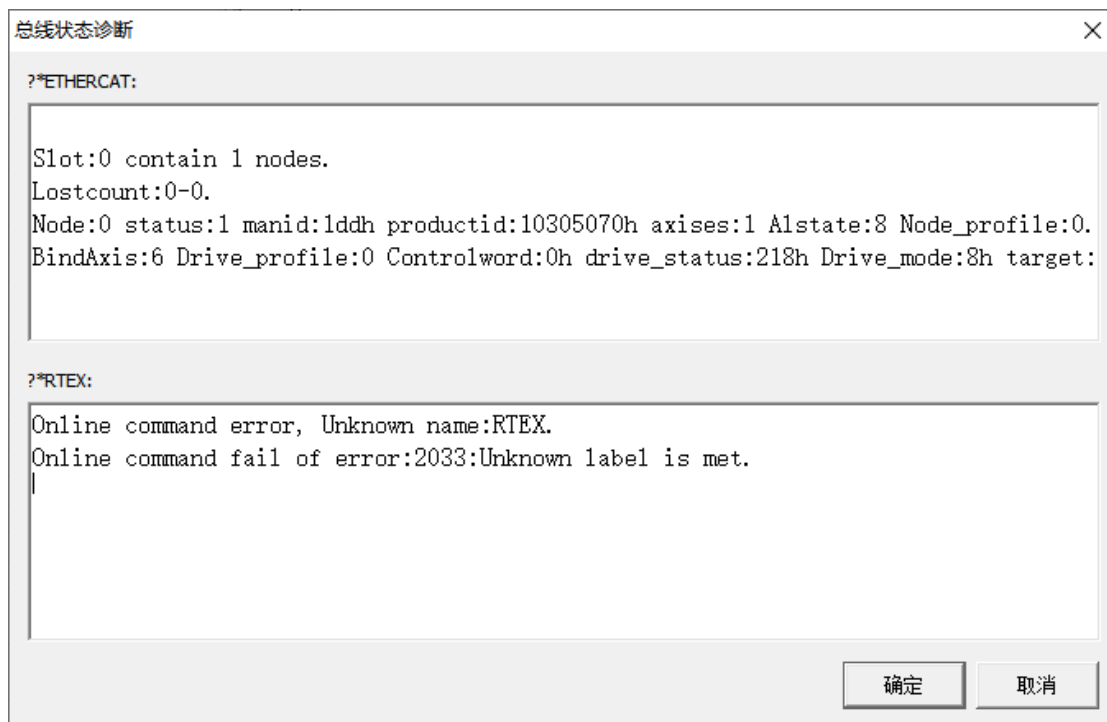
错误信息提示：



7.7 总线状态诊断

菜单栏“调试”-“总线状态诊断”窗口，方便查看控制器当前支持的总线，总线上连接的所有节点的设备信息。

打印信息含义参见?*ETHERCAT 指令帮助。该窗口与?*ETHERCAT 和?*RTEX 打印数据结果一致。



7.8 状态栏

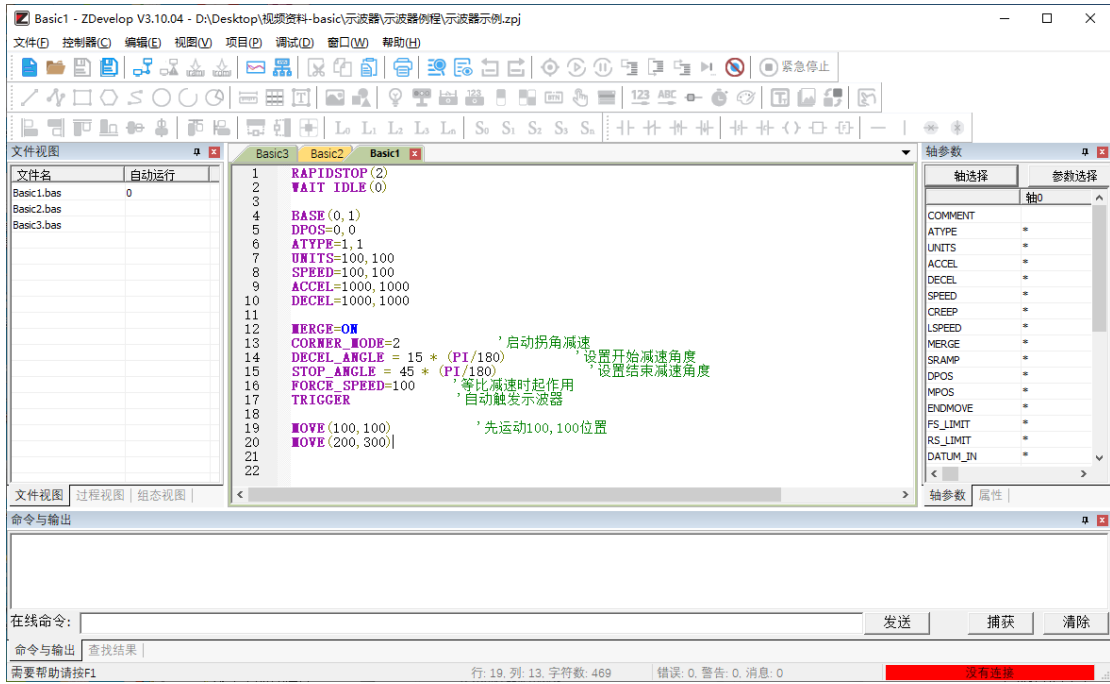
状态栏显示三部分内容，可用于光标定位、打印消息统计、控制器型号/IP/状态显示。

一：光标定位，显示光标处的行列等信息（例如 Basic，行 23，列 1，字符数 437 表示光标前程序有 13 行，光标在该行第一个字符后，首行到光标处字符总数 437；例如 PLC，ROW:30, Col:2 表示当前光标停留在第 30 行的第 2 个网格处）。

二：统计当前命令与输出窗口的错误信息、告警、打印信息的条数。

三：显示当前连接的控制器型号、控制器 IP 地址以及控制器运行状态，出现 ALM 或者 ERROR 右下角会提示哪个轴出错或者系统出错，并用红色背景突出显示。

没有连接到控制器显示红色字体“没有连接”：



控制器正常运行显示“运行”状态：



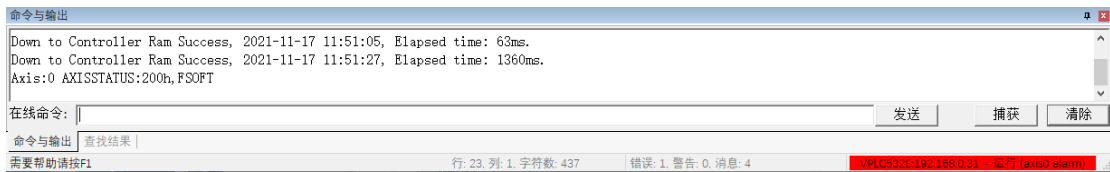
程序不运行或急停后显示“待机”状态：



按下调试菜单“暂停”按钮，程序暂停扫描：



运行过程中出现警报状态栏“红色闪烁”提示：



上图就可以看到轴 0 axis alarm，客户就可以针对性去检查 axis0 axisstatus 的状态或查看打印信息提示。

第八章 常见问题

程序运动出错后，ZDevelop 软件会显示出错信息，如果出错信息没有看到，可以通过命令行输入?*task 再次查看出错信息，双击出错信息可以自动切换到程序出错位置。

问题	可能原因
下载按钮是灰的	没有建立项目或没有打开项目 zpj 文件，建立项目后把程序文件加入项目。 没有连接到控制器。 部分控制器只支持下载到 ROM，不支持下载 RAM。
WARN: no program set autorun.	程序文件没有指定自动运行的任务号。
Error:5002, Operate Failed!	程序中有多个 hmi 任务，请去除多余的 hmi 任务，只留一个。
控制器告警灯亮	通过命令行输入?*task 再次查看出错信息。 查看轴状态 AXISSTATUS 的值是否非 0。 查看命令与输出窗口的打印信息提示。
手动运动时电机不动	查找轴参数里面的 AXISSTATUS，如果不为 0 确认对应的错误位，重点检查限位，软限位，告警位。 查看 ATYPE 看对应轴是否为 0，是否配置为编码器。 检查 UNITS 参数设置，是否配置为 0 或极小值。 检查电机接线。
手动运动电机只能一个方向动	查找轴参数里面的 AXISSTATUS，如果不为 0 确认对应的错误位，重点检查硬限位，软限位，告警位。 检查驱动器的双脉冲配置是否与控制器一致，控制器脉冲模式的配置参数为 INVERT_STEP。 检查电机接线，重点查方向线。