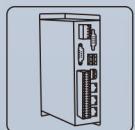
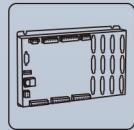


# 立式总线扩展模块

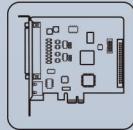
ZMIO310



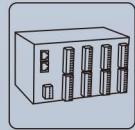
机器视觉运动  
控制一体机



运动控制器



运动控制卡



IO扩展模块



人机界面

## 声明

感谢您选择正运动公司的产品。在使用之前,请务必仔细阅读该手册,以便您能够正确、安全地使用本产品。本公司不对因使用本产品而造成的任何直接或间接损失承担责任。

本手册版权归深圳市正运动技术有限公司所有。未经本公司书面许可,严禁以任何形式翻印、翻译和抄袭本手册的任何内容。

本手册中的信息仅供参考。由于改进设计等原因,正运动公司保留对本资料的最终解释权!内容如有更改,恕不另行通知!

## 安全注意事项

为防止因错误使用本产品而可能带来的伤害和损害,对务必遵守的事项做出以下说明。



### 危 险

不要在有水的地方,存在腐蚀性、易燃性气体的环境内和靠近可燃性物质的地方使用。

安装或拆卸时,请确保产品处于断电状态。

电缆应切实接好,通电外露部位须通过绝缘物做到绝缘。

接线工作必须由专业人员来操作。

可能造成触电、火灾、损坏等



### 注 意

产品应安装在规定的环境范围内。

确保产品硬件电路板上没有异物。

安装后产品与安装架之间应紧密牢固。

产品安装后与周边部件之间应至少留出 2-3cm 以便通风和更换。

绝不可自行拆解、改造、修理。

可能造成损坏、误操作等

## 目录

第一章 产品信息 .....	1
1.1 产品简介 .....	1
1.2 模块介绍 .....	2
1.3 产品规格 .....	2
1.4 铭牌及型号 .....	3
1.5 供电需求 .....	3
1.6 应用环境 .....	4
1.7 硬件安装 .....	5
第二章 耦合器模块 .....	6
2.1 ZMI0310-ECAT 通讯模块 .....	6
接口定义 .....	6
性能规格 .....	6
用户端子 .....	7
EtherCAT 总线接口 .....	7
故障指示与处理对策 .....	8
2.2 ZMI0310-CAN 通讯模块 .....	9
接口定义 .....	9
性能规格 .....	9
用户端子 .....	10
CAN 总线接口 .....	10
拨码开关 .....	11
故障指示与处理对策 .....	11
2.3 耦合器子模块 (16 点位/32 点位) 扩展能力 .....	12
第三章 扩展子模块 .....	13
3.1 ZMI0310-1616N .....	13
接口定义 .....	13
用户端子 .....	13
故障指示与处理对策 .....	14
3.2 ZMI0310-1616P .....	15

接口定义 .....	15
用户端子 .....	15
故障指示与处理对策 .....	16
3.3 ZMI0310-32DI .....	17
接口定义 .....	17
用户端子 .....	17
故障指示与处理对策 .....	18
3.4 ZMI0310-32DO .....	19
接口定义 .....	19
用户端子 .....	19
故障指示与处理对策 .....	20
3.5 ZMI0310-32DOP .....	21
接口定义 .....	21
用户端子 .....	21
故障指示与处理对策 .....	22
3.6 ZMI0310-4AD .....	23
接口定义 .....	23
用户端子 .....	23
故障指示与处理对策 .....	24
3.7 ZMI0310-4DA .....	24
接口定义 .....	24
用户端子 .....	25
故障指示与处理对策 .....	25
3.8 子模块性能规格 .....	26
数字量输入 .....	26
数字量输出 (NPN) .....	26
数字量输出 (PNP) .....	26
模拟量输入 .....	27
模拟量输出 .....	27
3.9 子模块接线 .....	28

数字量输入 (NPN) .....	28
数字量输入 (PNP) .....	28
数字量输出 (NPN) .....	29
数字量输出 (PNP) .....	29
模拟量输入 (电压型) .....	30
模拟量输入 (电流型) .....	30
模拟量输出 (电压型) .....	31
模拟量输出 (电流型) .....	31
第四章 使用说明 .....	32
4.1 功耗计算示例 .....	32
4.2 IO 起始编号设置 .....	32
EtherCAT 总线扩展 .....	32
CAN 总线扩展 .....	33
4.3 数字量及模拟量读写 .....	34
4.4 子模块配置功能 .....	35
写数据字典 .....	35
读数据字典 .....	35
第五章 数据字典说明 .....	37
5.1 数据字典概览 .....	37
5.2 本地后级扩展地址说明 .....	38
子模块地址分配 .....	38
模块对应的数据字典 .....	39
5.3 数据字典详细说明 .....	39
格式说明 .....	39
索引: 5000h .....	39
索引: 5001h .....	40
索引: 6000h .....	43
索引: 7000h .....	44
索引: 6001h .....	45
索引: 7001h .....	47

---

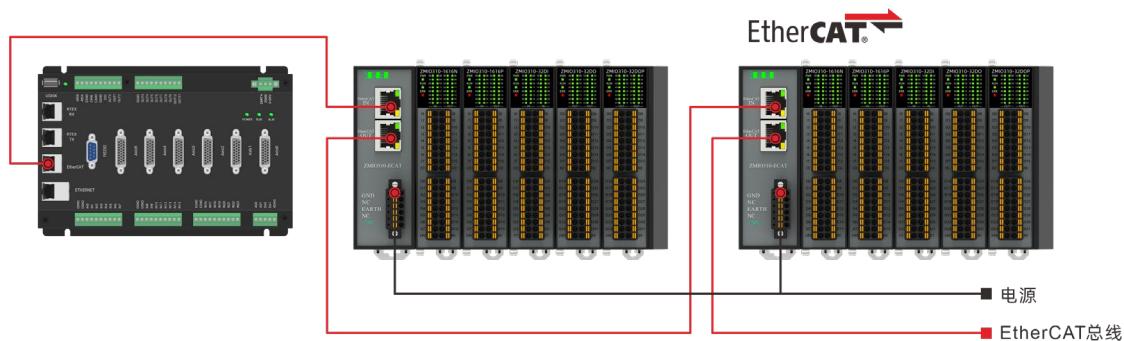
索引: 6002h .....	50
索引: 7002h .....	51
第六章 编程应用 .....	53
6.1 RTSys 软件使用 .....	53
6.2 上位机编程应用 .....	56
第七章 运行维护 .....	59
7.1 定期检查与维护 .....	59
7.2 故障排查 .....	59
第八章 售后服务 .....	61
附录 .....	62
更新记录 .....	62

# 第一章 产品信息

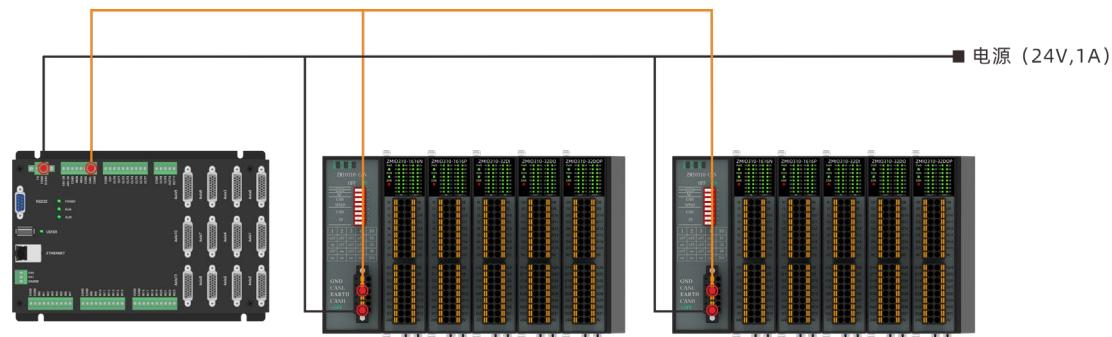
## 1.1 产品简介

ZMI0310 系列扩展模块是立式总线扩展模块，可支持 EtherCAT 和 CAN 两种总线方式扩展数字量 IO。当 IO、AD、DA 等资源不够的时候，需要耦合器模块（ZMI0310-ECAT、ZMI0310-CAN 通讯模块）搭配其他扩展子模块进行扩展，子模块输入输出口数量、类型可选，子模块包括 ZMI0310-1616N、ZMI0310-1616P、ZMI0310-32DI、ZMI0310-32DO、ZMI0310-32DOP，ZMI0310-4AD，ZMI0310-4DA。

EtherCAT 总线扩展系统框图如下图所示：



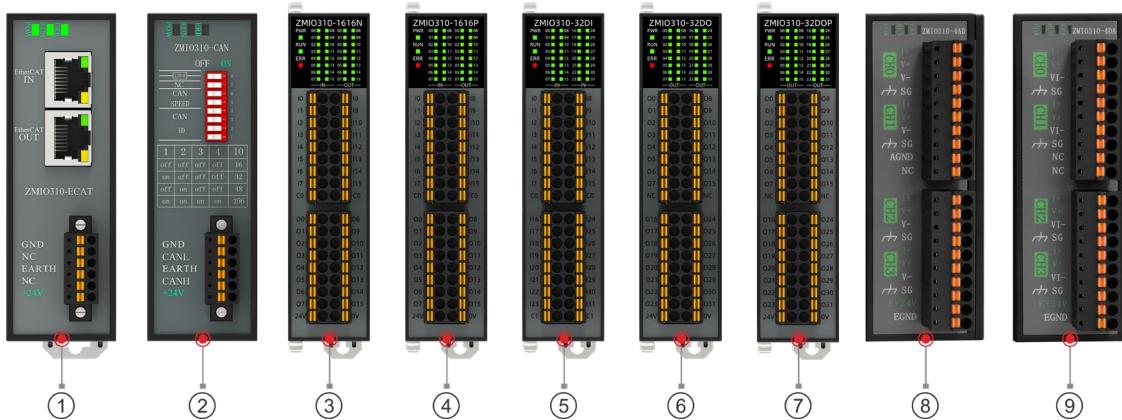
CAN 总线扩展系统框图如下图所示：



- 数字量输入输出均带信号指示灯，方便查看 IO 状态。
- 模拟量分辨率 16 位，可选电压型或电流型，且提供多种量程。
- 单个耦合器最多可扩展 512 路输入或 512 路输出，需要扩展更多，请选择多个耦合器。
- 单个耦合器最多可扩展 32 路 AD 或 32 路 DA，需要扩展更多，请选择多个耦合器。
- 扩展接线方便快捷。

本手册主要描述 ZMI0310 系列扩展模块的规格、特性及使用方法等，便于参考。使用前请详细阅读此手册，以便更清楚地掌握产品的特性，更安全地使用本产品。

## 1.2 模块介绍



序号	接口	说明
①	ZMIO310-ECAT	耦合器模块，ECAT 通讯模块
②	ZMIO310-CAN	耦合器模块，CAN 通讯模块
③	ZMIO310-1616N	扩展子模块，16 IN(NPN/PNP)、16 OUT(NPN)
④	ZMIO310-1616P	扩展子模块，16 IN(NPN/PNP)、16 OUT(PNP)
⑤	ZMIO310-32DI	扩展子模块，32 IN(NPN/PNP)
⑥	ZMIO310-32DO	扩展子模块，32 OUT(NPN)
⑦	ZMIO310-32DOP	扩展子模块，32 OUT(PNP)
⑧	ZMIO310-4AD	扩展子模块，4 路 AD，16bit
⑨	ZMIO310-4DA	扩展子模块，4 路 DA，16bit

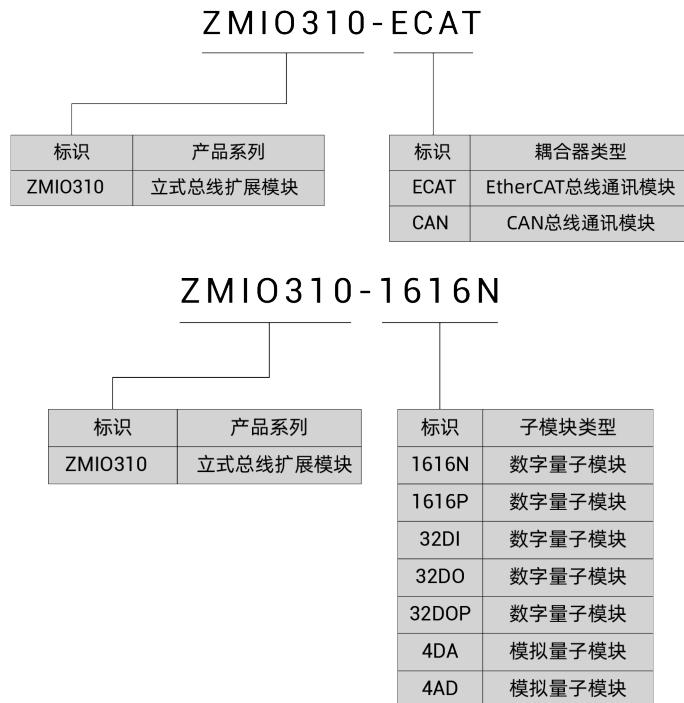
## 1.3 产品规格

型号	ZMIO310-4AD	ZMIO310-4DA	ZMIO310-ECAT	ZMIO310-CAN
AD 输入	4	0	—	—
DA 输出	0	4	—	—
模拟量分辨率	16bit	16bit	—	—
通讯协议	—	—	EtherCAT 总线通讯	CAN 总线通讯
外形尺寸(mm)	32*95*93			

型号	ZMIO310-1616N	ZMIO310-1616P	ZMIO310-32DI	ZMIO310-32DO	ZMIO310-32DOP
数字输入	16	16	32	0	0
数字输入类型	NPN/PNP	NPN/PNP	NPN/PNP	—	—
数字输出	16	16	0	32	32
数字输出类型	NPN	PNP	—	NPN	PNP
外形尺寸(mm)	24*95*93				

## 1.4 铭牌及型号

铭牌信息：耦合器(上)+扩展子模块(下)



产品选型：

耦合器	内部输入口	内部输出口	通讯接口	功能描述
ZMIO310-ECAT	—	—	EtherCAT	EtherCAT 总线通讯模块
ZMIO310-CAN	—	—	CAN	CAN 总线通讯模块

扩展子模块	内部输入口	内部输出口	AD/DA	IO 类型	功能描述
ZMIO310-1616N	16	16	—	IN: NPN/PNP OUT: NPN	输入/输出模块
ZMIO310-1616P	16	16	—	IN: NPN/PNP OUT: PNP	输入/输出模块
ZMIO310-32DI	32	0	—	IN: NPN/PNP	输入模块
ZMIO310-32DO	0	32	—	OUT: NPN	输出模块
ZMIO310-32DOP	0	32	—	OUT: PNP	输出模块
ZMIO310-4AD	—	—	4	—	AD模块 (16Bit)
ZMIO310-4DA	—	—	4	—	DA模块 (16Bit)

## 1.5 供电需求

此扩展模块采用双电源供电，即耦合器模块 ZMIO310-ECAT 或 ZMIO310-CAN 采用一个电源，扩展子模块采用另一个电源（在电源功率足够的情况下，子模块可使用同一个电源供电，但不可与耦合器模块使用同一个电源）。

型号/项目	电源电压	启动电流	工作电流
ZMI0310-ECAT	耦合器主电源 建议 DC24V, 最大 18V-36V  扩展子模块电源 建议 DC24V, 最大 18V-36V	1A	0.5A
ZMI0310-CAN		0.2A	0.1A
ZMI0310-1616N		0.2A	0.1A
ZMI0310-1616P		0.6A	0.3A
ZMI0310-32DI		0.2A	0.1A
ZMI0310-32DO		0.2A	0.1A
ZMI0310-32DOP		1A	0.5A
ZMI0310-4AD		0.2A	0.1A
ZMI0310-4DA		0.2A	0.1A

## 1.6 应用环境

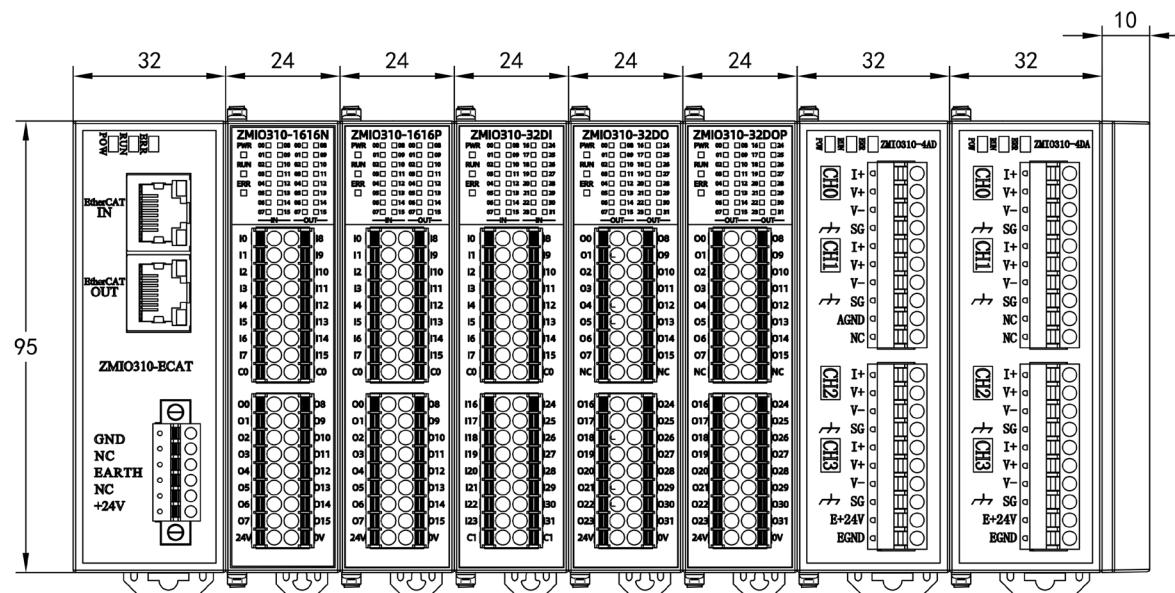
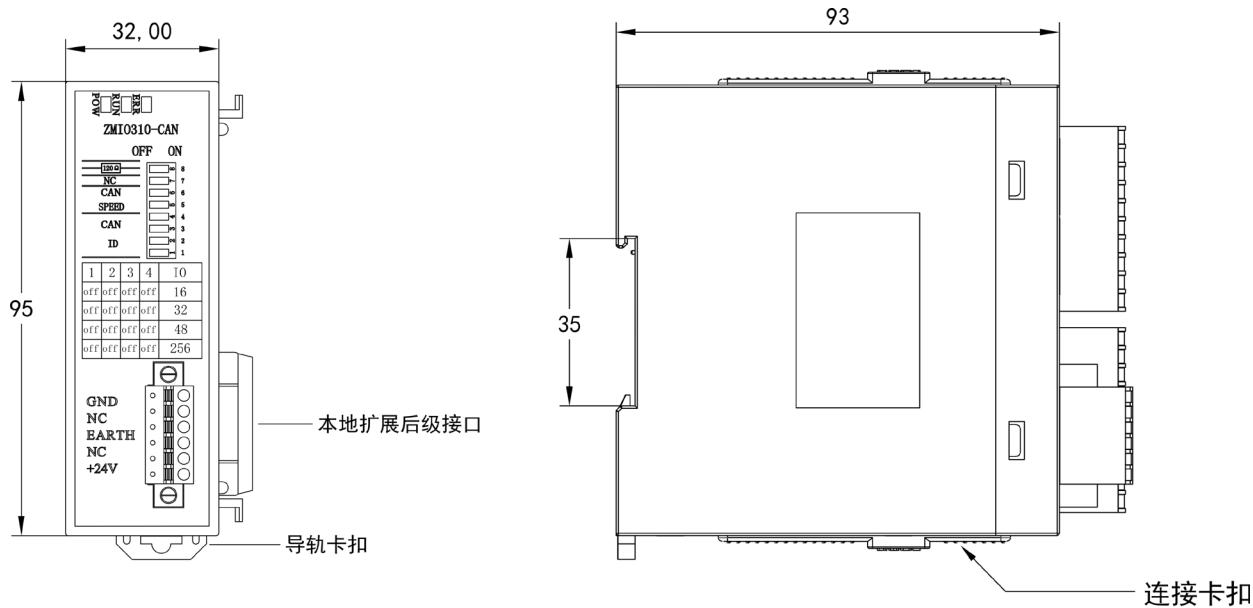
项目	参数
工作温度	-10°C ~ 55°C
工作相对湿度	10%~95%非凝结
储存温度	-40°C ~ 80°C (不冻结)
储存湿度	90%RH 以下(不结露)
振动	频率
	3.5mm(直接安装) (<9Hz)
	1g(直接安装) (>9Hz)
	3 轴向
冲击(碰撞)	15g, 11ms, 半正弦波, 3 轴向
防护等级	IP20

## 1.7 硬件安装

ZMI0310 系列扩展模块安装步骤如下：

- 请使用 35mm 的标准 DIN 导轨。
- 打开 ECAT 通讯模块的导轨卡扣，将 ECAT 通讯模块嵌入在 DIN 导轨上。
- 压合 ECAT 通讯模块的导轨卡扣，将 ECAT 通讯模块固定在 DIN 导轨上。

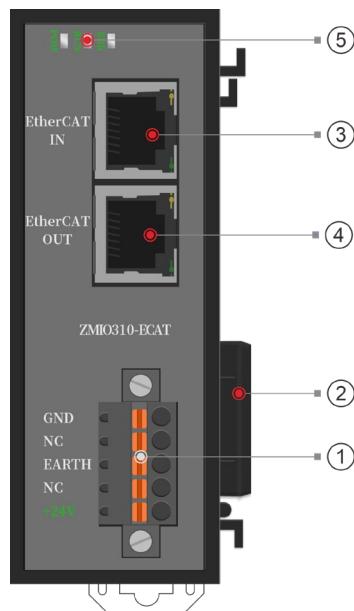
模块尺寸图：(单位：mm)



## 第二章 耦合器模块

### 2.1 ZMIO310-ECAT 通讯模块

#### 接口定义



序号	接口	说明
①	用户端子	电源端子, 接 DC24V 电源
②	本地扩展后级接口	连接扩展子模块, 不支持热插拔
③	EtherCAT IN	EtherCAT 输入口, 用于连接 EtherCAT 主站或上一级 EtherCAT 从站
④	EtherCAT OUT	EtherCAT 输出口, 用于连接 EtherCAT 从站
⑤	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

注意: EtherCAT IN 和 EtherCAT OUT 不可混用。

#### 性能规格

项目	规格
电源电压	24VDC
通讯协议	EtherCAT 工业实时总线协议
支持服务	CoE (PDO、SDO)、固件升级
通讯周期	250 μs、500 μs、1ms、2ms、4ms 等
最高通讯速度	以太网 100Mbps
网口	标准以太网 RJ45 接口
传输媒介	超五类屏蔽双绞线

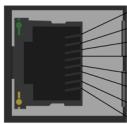
传输距离	两节点小于 100 米
后续子模块扩展能力	ZMI0310-32DI/DO/DOP 最多可扩展 16 个子模块(可以混合扩展) ZMI0310-1616N/P 最多可扩展 8 个子模块(可以混合扩展) ZMI0310-4AD/DA 最多可扩展 8 个子模块(可以混合扩展) 扩展子模块的总个数在 16 个以内即可 (一个 ZMI0310-1616N/1616P 等于同时使用两个子模块) 实际数量还要以各模块功耗进行限定 例: 已经扩展 10 个 ZMI0310-32DI/DO/DOP 子模块的情况下, 最多可以再接入 3 个 ZMI0310-1616N/P 子模块
后续子模块最多 IO 数	512 路进或 512 出
后续子模块最多 AIO 数	32 通道 AD 或 32 通道 DA
自身功耗	1.6W
提供给后级内部功耗	8.4W

## 用 户 端 子

主电源端子	名称	类型	功能
GND	GND	输入	主电源地
NC	NC	-	预留
EARTH	EARTH	接地	安规地/屏蔽层
NC	NC	-	预留
+24V	+24V	输入	主电源 24V 输入

## EtherCAT 总 线 接 口

### 规 格

引脚定义			项目	说明
	引脚	信号	说明	
	1	TX+	发送信号 (+)	
	2	TX-	发送信号 (-)	
	3	RX+	接收信号 (+)	
	4	NC	预留	
	5	NC	预留	
	6	RX-	接收信号 (-)	
	7	NC	预留	
	8	NC	预留	

### 接 线

1. 连接 EtherCAT 总线驱动器或其他从站设备时, 可通过一根超五类屏蔽网线与后级设备的 EtherCAT IN 口连接, 还可再通过该从站设备的 EtherCAT OUT 口继续连接后级从站设备的 EtherCAT IN 口实现多级扩展;
2. 网口灯闪烁情况:

状态	常亮	闪烁
LED 灯		
绿灯	建立十兆通讯	正在数据收发
黄灯	建立百兆通讯	正在数据收发

## 注意 →

- 请使用超五类屏蔽网线，尤其是环境恶劣的场合，以提升信号的抗干扰性；
- 现场布线要注意和强电之间的距离，建议 30cm 以上；
- 注意控制器的接地要良好，机壳的接地要接在标准的厂房地桩上。

## 使 用 方 法

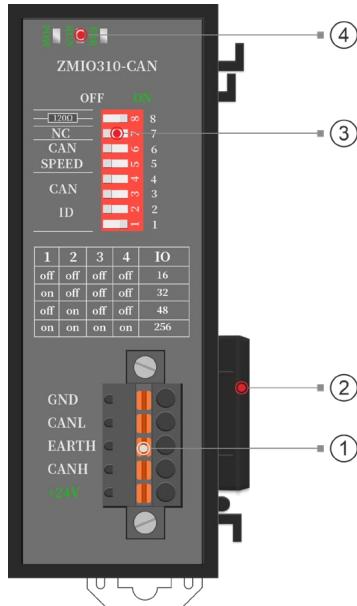
1. 请按照以上接线说明正确完成电源、EtherCAT 总线模块的接线；
2. 正确接线后上电并通过网口/串口将控制器连接到 RTSys；
3. EtherCAT 总线连接驱动器设备：
  - a) 采用“SLOT\_SCAN”指令扫描总线上的槽位号；
  - b) 采用“AXIS\_ADDRESS”指令映射轴号；
  - c) 采用“SLOT\_START”指令开启总线或采用“SLOT\_STOP”指令关闭总线；
  - d) 完成后如本地脉冲轴进行配置和操作；
4. EtherCAT 总线连接扩展模块：
  - a) 采用“SLOT\_SCAN”指令扫描总线上的槽位号；
  - b) 采用“AXIS\_ADDRESS”指令映射轴号，采用“NODE\_IO/NODE\_AIO”指令映射 IO 编号；
  - c) 采用“SLOT\_START”指令开启总线或采用“SLOT\_STOP”指令关闭总线；
  - d) 以上完成后可如本地 IO 和轴一般进行操作；
5. 通过“RTSys>控制器>控制器状态>槽位 0 节点”界面直观查看槽位号节点信息；
6. 以上指令详解以及其他相关指令请查看“RTSys>帮助文档>RTBasic 帮助文档”。

## 故 障 指 示 与 处 理 对 策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	慢速交替闪烁	EtherCAT 主站与耦合器模块之间通讯断开		检查水晶头是否松动； 检查网线是否损坏； 重启电源。
亮	快速交替闪烁	模块预扫描的子模块和实际接续子模块不完全相符		检查接续的子模块是否丢失、 出现故障； 检查是否发生热插拔； 重启电源。

## 2.2 ZMI0310-CAN 通讯模块

### 接 口 定 义



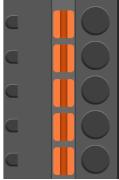
序号	接口	说明
①	用户端子	电源端子, 接 DC24V 电源
	CAN 总线接口	接控制器做从站或接子模块来扩展资源
②	本地扩展后级接口	连接扩展子模块, 不支持热插拔
③	拨码开关	8 个拨码, 可选择 CAN 地址、CAN 速度、控制 CAN 120 Ω 终端电阻的导通
④	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

### 性 能 规 格

项目	规格
电源电压	24VDC
通讯接口	CAN总线接口
可连接数	最多可连接16个CAN从站模块
传输距离	小于30米
地址设置	拨码开关设置
后续子模块扩展能力	ZMI0310-32DI/DO/DOP 最多可扩展 3 个子模块(可以混合扩展) ZMI0310-1616N/P 最多可扩展 6 个子模块 (可以混合扩展) ZMI0310-4AD/DA 最多可扩展 3 个子模块 (可以混合扩展) 扩展子模块的总个数在 16 个以内即可, IN 或 OUT 都不超过 96 个 (一个ZMI0310-1616N/1616P等于同时使用两个子模块) 实际数量还要以各模块功耗进行限定 例: 已经扩展 3 个 ZMI0310-32DI 和 3 个 ZMI0310-32DO 子模块的情况下, 最多可以再接入 3 个 ZMI0310-4AD 子模块和 3 个 ZMI0310-4DA 子模块

自身功耗	0.6W
提供给后级内部功耗	7.9W
支持服务	固件升级
注意：新版本升级了模块程序，增加了子模块数量超限耦合器会告警功能。	

## 用 户 端 子

主电源/CAN 端子	名称	类型	功能
	GND	输入	主电源地
	CANL	输入/输出	CAN 差分数据-
	EARTH	接地	安规地/屏蔽层
	CANH	输入/输出	CAN 差分数据+
	+24V	输入	主电源 24V 输入

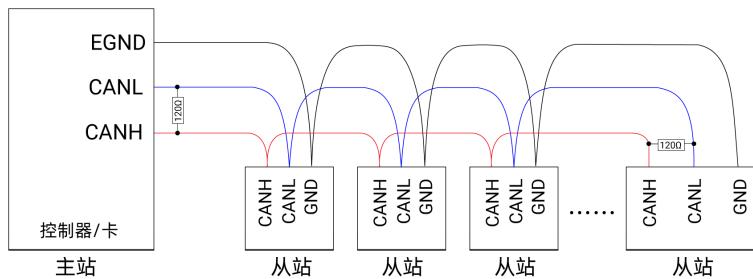
## C A N 总 线 接 口

接口定义见“用户端子”章节

## 规 格

CAN	说明
通讯速率	$\leq 1\text{Mbps}$
终端电阻	$120\Omega$
布线结构	菊花链结构
可扩展节点数	$\leq 16$ 个
接线长度	建议 $<30\text{m}$ (500kbps)
通讯隔离	√

## 接 线



## 注 意 …>

- 如上为菊花链布线结构，不可采用星型结构，节点之间的距离越短越好；
- 请在 CAN 总线最两端接口各并接一个  $120\Omega$  的终端电阻，匹配电路阻抗，保证通讯稳定性；
- 请务必连接 CAN 总线上各个节点的公共端达到共零效果，也防止 CAN 芯片烧坏；
- 请使用双绞屏蔽线，尤其是环境恶劣的场合，务必使屏蔽层充分接地（机壳）；

- 现场布线要注意和强电之间的距离，建议 30cm 以上；
- 要注意整个线路上的接地要良好，机壳的接地要接在标准的厂房地桩上。

## 使 用 方 法

1. 请按照以上接线说明正确完成电源、CAN 总线模块的接线；
2. 正确接线后上电并通过网口/串口将控制器连接到 RTSys；
3. 配置控制器 CAN 主站：
  - a) 采用“CANIO\_ADDRESS”指令设置主站“地址”和“速率”；
  - b) 采用“CANIO\_ENABLE”指令设置使能或禁止 CAN 主站功能；
  - c) 可通过“RTSys>控制器>控制器状态>通讯配置”界面直观查看参数；
  - d) 可通过“RTSys>控制器>控制器状态>ZCan 节点”界面直观查看总线节点参数；
4. 正确匹配 CAN 从站模块的“速率”和“地址”并完成资源映射操作；
5. 设置完成后重启所有站点即可建立通讯正常使用，通讯成功“控制器状态>CAN 节点”界面则会显示扩展模块的信息；若从站模块“ERR”灯亮起则表示通讯建立失败；
6. 注意 CAN 总线上每个节点的“速率”设置必须一致，“地址”设置和资源映射不能够产生冲突，否则会通讯建立失败或者通讯错乱；
7. 以上指令详解以及其他相关指令请查看“RTSys>帮助文档>RTBasic 帮助文档”。

## 拨 码 开 关



CAN 扩展板一般带 8 位拨码开关用于通讯配置和资源映射，拨 ON 生效，拨码含义如下：

- 1-4: CAN 模块地址 ID，组合值为 0-15 (4 位二进制转 10 进制)；
- 5-6: CAN 模块通讯速率，组合值为 0-3 (2 位二进制转 10 进制)，可选四种不同的速度；
- 7: 预留；
- 8: 120 欧电阻，拨 ON 表示 CANL 和 CANH 间接入一个 120 欧电阻。

扩展 IO 地址映射和 CAN 通讯速度的详细配置方法，参见第四章节说明。

## 故 障 指 示 与 处 理 对 策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确； 检查总线两端是否接入 120 Ω 电阻；

				检查是否多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
--	--	--	--	--

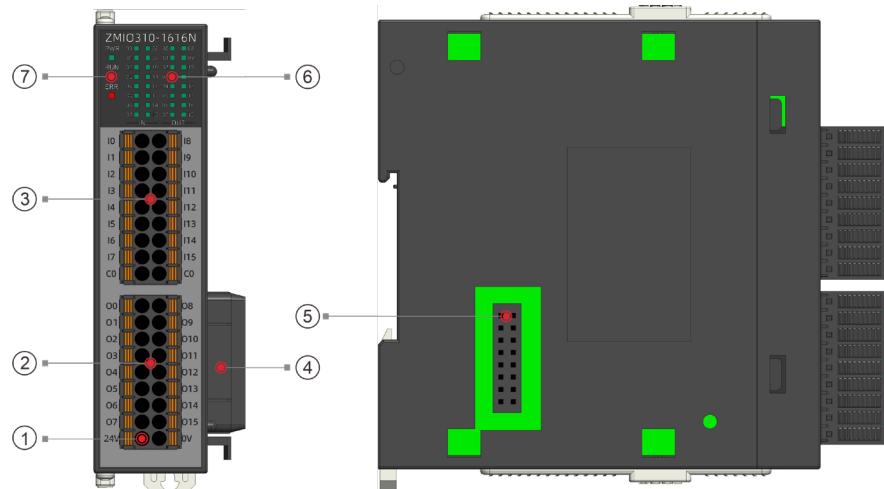
## 2.3 耦合器子模块 (16 点位/32 点位) 扩展能力

耦合器	子模块型号	子模块最大个数
ZMI0310-ECAT	所有子模块组合 (1616N/1616P两种子模块占2个子模块个数)	16
	16DI/16DO/16DOP/32DI/32DO/32DOP	16
	1616N/1616P/4AD/4DA	8
ZMI0310-ZCAN	所有子模块组合 (IN数量和 OUT 数量均不可超过96)	16
	16DI/16DO/16DOP/1616N/1616P	6
	32DI/32DO/32DOP/4AD/4DA	3

## 第三章 扩展子模块

### 3.1 ZMI0310-1616N

#### 接口定义



序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	数字输出口	OUT0-OUT15
③	数字输入口	IN0-IN15
④	本地扩展后级接口	连接扩展子模块, 不支持热插拔
⑤	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑥	IO 信号指示灯	分别对应各路输入/输出信号指示
⑦	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

#### 用户端子

端子	名称	类型	功能
I0	I8	NPN/PNP 型 低速输入	数字输入 0
I1	I9		数字输入 1
I2	I10		数字输入 2
I3	I11		数字输入 3
I4	I12		数字输入 4
I5	I13		数字输入 5
I6	I14		数字输入 6
I7	I15		数字输入 7
CO	I8		数字输入 8

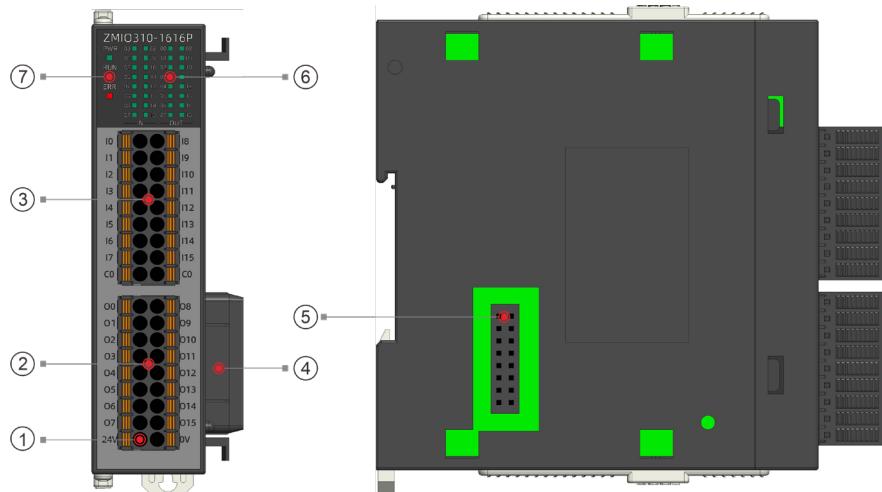
I9		数字输入 9
I10		数字输入 10
I11		数字输入 11
I12		数字输入 12
I13		数字输入 13
I14		数字输入 14
I15		数字输入 15
C0	COM0; IN(0-15) 公共端, 决定其输入类型 COM0 接 24V 为 NPN; COM0 接 0V 为 PNP	
00	NPN 型 低速输出	数字输出 0
01		数字输出 1
02		数字输出 2
03		数字输出 3
04		数字输出 4
05		数字输出 5
06		数字输出 6
07		数字输出 7
08		数字输出 8
09		数字输出 9
0A		数字输出 10
0B		数字输出 11
0C		数字输出 12
0D		数字输出 13
0E		数字输出 14
0F		数字输出 15
24V	24V 电源输入+	
0V	24V 电源输入-, 数字输出公共端	

## 故障指示与处理对策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 120 Ω 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。
			主站/从站固件异常	主站/从站均需升级固件

### 3.2 ZMI0310-1616P

#### 接 口 定 义



序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	数字输出口	OUT0-OUT15
③	数字输入口	IN0-IN15
④	本地扩展后级接口	连接扩展子模块, 不支持热插拔
⑤	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑥	IO 信号指示灯	分别对应各路输入/输出信号指示
⑦	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯 RUN 指示灯: 绿色, 正常运行时亮灯 ERR 指示灯: 红色, 运行错误时亮灯

#### 用 户 端 子

端子	名称	类型	功能
	I0	NPN/PNP 型 低速输入	数字输入 0
	I1		数字输入 1
	I2		数字输入 2
	I3		数字输入 3
	I4		数字输入 4
	I5		数字输入 5
	I6		数字输入 6
	I7		数字输入 7
	I8		数字输入 8
	I9		数字输入 9
	I10		数字输入 10
	I11		数字输入 11

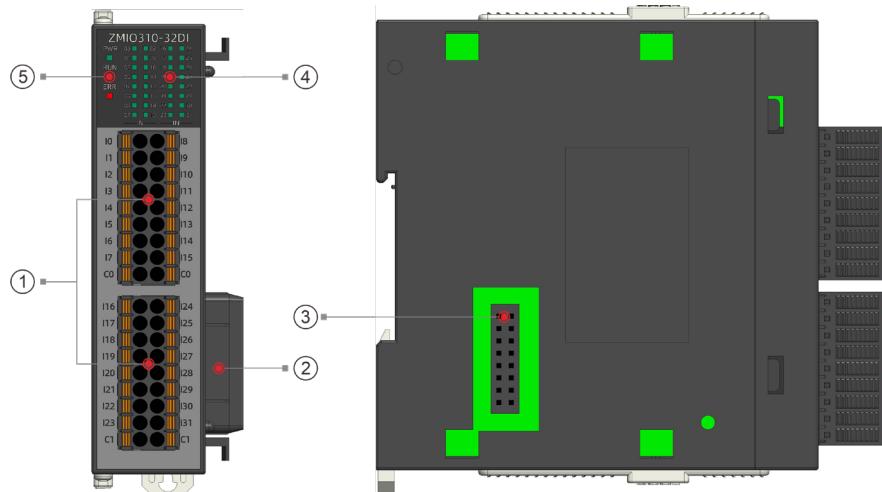
I12		数字输入 12
I13		数字输入 13
I14		数字输入 14
I15		数字输入 15
C0	COM0; IN(0-15) 公共端, 决定其输入类型 COM0 接 24V 为 NPN; COM0 接 0V 为 PNP	
00	PNP 型 低速输出	数字输出 0
01		数字输出 1
02		数字输出 2
03		数字输出 3
04		数字输出 4
05		数字输出 5
06		数字输出 6
07		数字输出 7
08		数字输出 8
09		数字输出 9
0A		数字输出 10
0B		数字输出 11
0C		数字输出 12
0D		数字输出 13
0E		数字输出 14
0F		数字输出 15
24V	24V 电源输入+	
0V	24V 电源输入-, 数字输出公共端	

## 故障指示与处理对策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 120Ω 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符 主站/从站固件异常	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。 主站/从站均需升级固件

### 3.3 ZMI0310-32DI

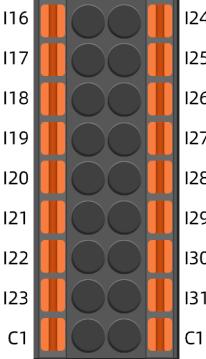
#### 接 口 定 义



序号	接口	说明
①	数字输入口	IN0-IN31
②	本地扩展后级接口	连接扩展子模块, 不支持热插拔
③	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
④	IO 信号指示灯	分别对应各路输入信号指示
⑤	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯 RUN 指示灯: 绿色, 正常运行时亮灯 ERR 指示灯: 红色, 运行错误时亮灯

#### 用 户 端 子

端子	名称	类型	功能 1
	I0	NPN/PNP 型 低速输入	数字输入 0
	I1		数字输入 1
	I2		数字输入 2
	I3		数字输入 3
	I4		数字输入 4
	I5		数字输入 5
	I6		数字输入 6
	I7		数字输入 7
	I8		数字输入 8
	I9		数字输入 9
	I10		数字输入 10
	I11		数字输入 11
	I12		数字输入 12
	I13		数字输入 13

	I14		数字输入 14
	I15		数字输入 15
	C0	IN(0-15) 公共端, 决定其输入类型 COM0 接 24V 为 NPN; COM0 接 0V 为 PNP	
 NPN/PNP 型 低速输入	I16	NPN/PNP 型 低速输入	数字输入 16
	I17		数字输入 17
	I18		数字输入 18
	I19		数字输入 19
	I20		数字输入 20
	I21		数字输入 21
	I22		数字输入 22
	I23		数字输入 23
	I24		数字输入 24
	I25		数字输入 25
	I26		数字输入 26
	I27		数字输入 27
	I28		数字输入 28
	I29		数字输入 29
	I30		数字输入 30
	I31		数字输入 31
	C1	IN(16-31) 公共端, 决定其输入类型 COM1 接 24V 为 NPN; COM1 接 0V 为 PNP	
	C1		

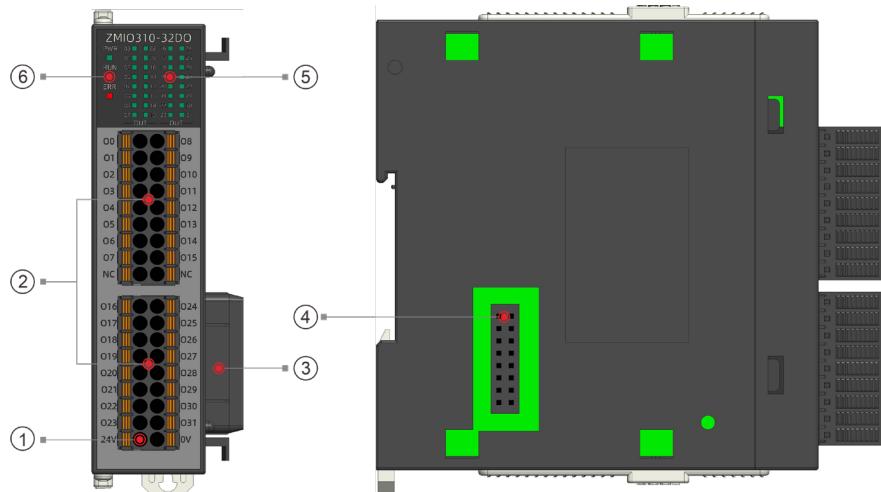
说明: 该模块本身不带有 24V/0V 电源输入端子, 不需要单独供电。其输入口类型可通过用户端子上的 C0/C1 端接入耦合器模块的+24V/GND 或其他子模块的 24V/0V 来切换。

## 故 障 指 示 与 处 理 对 策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 120Ω 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。
			主站/从站固件异常	主站/从站均需升级固件

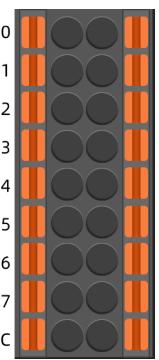
### 3.4 ZMI0310-32DO

#### 接 口 定 义

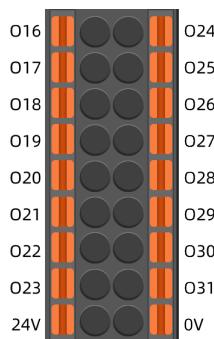


序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	数字输出口	OUT0-OUT31
③	本地扩展后级接口	连接扩展子模块, 不支持热插拔
④	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑤	IO 信号指示灯	分别对应各路输出信号指示
⑥	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

#### 用 户 端 子

端子	名称	类型	功能 1
	00	NPN 型 低速输出	数字输出 0
	01		数字输出 1
	02		数字输出 2
	03		数字输出 3
	04		数字输出 4
	05		数字输出 5
	06		数字输出 6
	07		数字输出 7
	08		数字输出 8
	09		数字输出 9
	010		数字输出 10
	011		数字输出 11
	012		数字输出 12

	013		数字输出 13
	014		数字输出 14
	NC	预留	
	016		数字输出 16
	017		数字输出 17
	018		数字输出 18
	019		数字输出 19
	020		数字输出 20
	021		数字输出 21
	022		数字输出 22
	023		数字输出 23
	024		数字输出 24
	025		数字输出 25
	026		数字输出 26
	027		数字输出 27
	028		数字输出 28
	029		数字输出 29
	030		数字输出 30
	031		数字输出 31
	24V	24V 电源输入+	
	0V	24V 电源输入-, 数字输出公共端	

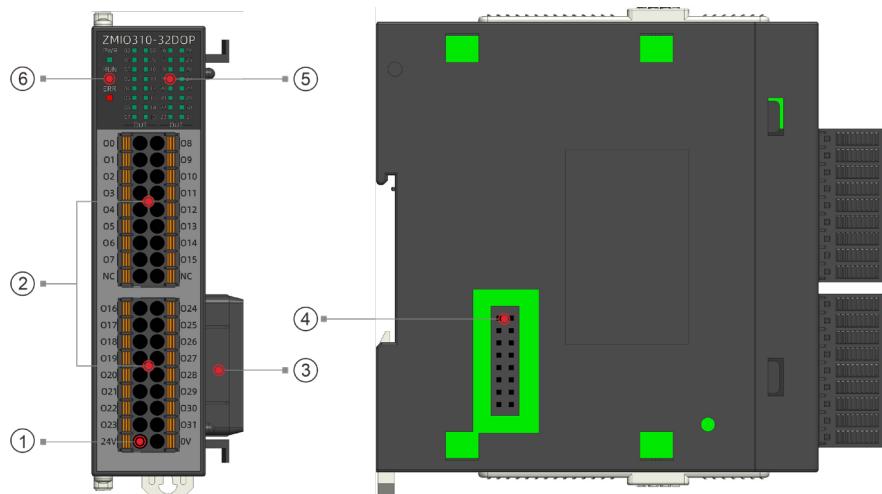
NPN 型  
低速输出

## 故障指示与处理对策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 120Ω 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符 主站/从站固件异常	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。 主站/从站均需升级固件

### 3.5 ZMI0310-32DOP

#### 接 口 定 义

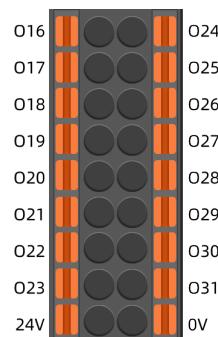


序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	数字输出口	OUT0-OUT31
③	本地扩展后级接口	连接扩展子模块, 不支持热插拔
④	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑤	IO 信号指示灯	分别对应各路输出信号指示
⑥	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

#### 用 户 端 子

端子	名称	类型	功能 1
	00	PNP 型 低速输出	数字输出 0
	01		数字输出 1
	02		数字输出 2
	03		数字输出 3
	04		数字输出 4
	05		数字输出 5
	06		数字输出 6
	07		数字输出 7
	08		数字输出 8
	09		数字输出 9
	010		数字输出 10
	011		数字输出 11
	012		数字输出 12

	013		数字输出 13
	014		数字输出 14
	NC	预留	
	016		数字输出 16
	017		数字输出 17
	018		数字输出 18
	019		数字输出 19
	020		数字输出 20
	021		数字输出 21
	022		数字输出 22
	023		数字输出 23
	024		数字输出 24
	025		数字输出 25
	026		数字输出 26
	027		数字输出 27
	028		数字输出 28
	029		数字输出 29
	030		数字输出 30
	031		数字输出 31
	24V	24V 电源输入+	
	0V	24V 电源输入-, 数字输出公共端	

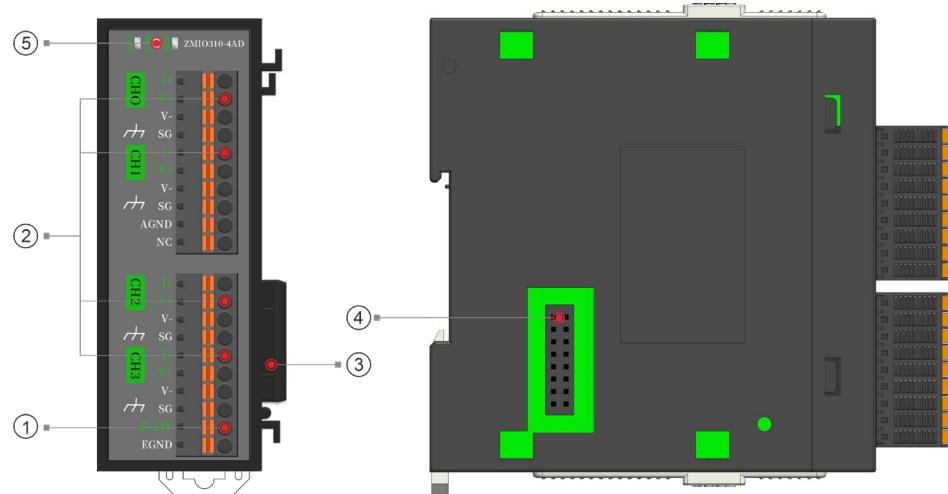
PNP 型  
低速输出

## 故障指示与处理对策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 120Ω 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符 主站/从站固件异常	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。 主站/从站均需升级固件

### 3.6 ZMI0310-4AD

#### 接 口 定 义



序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	模拟量输入口	4 通道模拟量输入
③	本地扩展后级接口	连接扩展子模块, 不支持热插拔
④	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑤	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯 RUN 指示灯: 绿色, 正常运行时亮灯 ERR 指示灯: 红色, 运行错误时亮灯

#### 用 户 端 子

端子	名称	类型	功能 1
I+ V+ V- SG I+ V+ V- SG AGND NC	I+	模拟量输入	通道 0 电流输入正极
	V+		通道 0 电压输入正极
	V-		通道 0 电压输入负极
	SG		安规地/屏蔽层
	I+		通道 1 电流输入正极
	V+		通道 1 电压输入正极
	V-		通道 1 电压输入负极
	SG		安规地/屏蔽层
	AGND		模拟量公共电流输入负极
	NC		预留
	I+	模拟量输入	通道 2 电流输入正极
	V+		通道 2 电压输入正极
	V-		通道 2 电压输入负极
	SG		安规地/屏蔽层

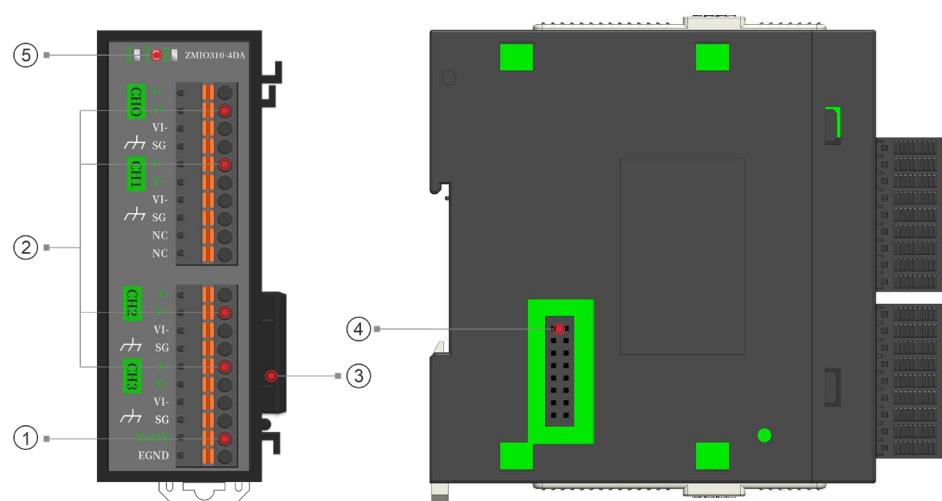
I+	I+	通道 3 电流输入正极
V+	V+	通道 3 电压输入正极
V-	V-	通道 3 电压输入负极
SG	SG	安规地/屏蔽层
I+	E+24V	/ 24V 电源输入+
V+		
V-		
SG		
E+24V		24V 电源输入-
EGND		

## 故障指示与处理对策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 $120\Omega$ 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符 主站/从站固件异常	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。 主站/从站均需升级固件

## 3.7 ZMI0310-4DA

### 接口定义



序号	接口	说明
①	IO 电源	电源端子, 接 DC24V 电源
②	模拟量输出口	4 通道模拟量输出
③	本地扩展后级接口	连接扩展子模块, 不支持热插拔

④	本地扩展前级接口	连接耦合器模块或扩展子模块, 不支持热插拔
⑤	状态指示灯	POW 指示灯: 绿色, 电源接通时亮灯
		RUN 指示灯: 绿色, 正常运行时亮灯
		ERR 指示灯: 红色, 运行错误时亮灯

## 用 户 端 子

端子	名称	类型	功能 1
I+ V+ VI- SG I+ V+ VI- SG NC NC	I+	模拟量输出	通道 0 电流输出正极
	V+		通道 0 电压输出正极
	VI-		通道 0 电压/电流输出负极
	SG		安规地/屏蔽层
	I+		通道 1 电流输出正极
	V+		通道 1 电压输出正极
	VI-		通道 1 电压/电流输出负极
	SG		安规地/屏蔽层
	NC		预留
	NC		预留
I+ V+ VI- SG I+ V+ VI- SG E+24V EGND	I+	模拟量输出	通道 2 电流输出正极
	V+		通道 2 电压输出正极
	VI-		通道 2 电压/电流输出负极
	SG		安规地/屏蔽层
	I+		通道 3 电流输出正极
	V+		通道 3 电压输出正极
	VI-		通道 3 电压/电流输出负极
	SG		安规地/屏蔽层
	E+24V		24V 电源输入+
	EGND		24V 电源输入-

## 故 障 指 示 与 处 理 对 策

状态指示灯			原因	解决方法
POW	RUN	ERR		
亮	亮	亮	CAN 通讯出现异常	检查 CAN 总线端子接线是否正确; 检查总线两端是否接入 $120\Omega$ 电阻; 检查是否有多个 CAN 通讯模块使用相同的硬件 ID; 检查子模块数量是否超过最大模块数量。
亮	灭	灭	模块预扫描的子模块和实际接续子模块不完全相符	检查接续的子模块是否丢失、出现故障; 检查是否发生热插拔; 重启电源。
			主站/从站固件异常	主站/从站均需升级固件

### 3.8 子模块性能规格

#### 数 字 量 输入

项目	规格
电源电压	24V DC
输入类型	数字量输入
IO供电电压输入方式	NPN/PNP NPN (IO 公共端接+24V) PNP (IO 公共端接EGND)
输入电流 (典型)	NPN型 (-4.8mA)； PNP型 (+4.8mA)
输入阻抗	4.7K Ω
ON时电压	PNP型>7.2V； NPN型<14.5V
OFF时电压	PNP型<6.8V； NPN型>14.7V
内部功耗	0.3W
自身功耗	1.9W
隔离方式	光耦隔离
输入频率	<5kHz
输入动作显示	输入为ON状态时，输入指示灯亮

#### 数 字 量 输出 ( N P N )

项目	规格
电源电压	24V DC
输出类型	数字量输出
输出方式	NPN 型，有输出时为低电平
输出过流保护	最大 300mA，最大跳闸电流 600mA
OFF 时最大漏电流	25 μ A
ON 时响应时间	12 μ s
OFF 时响应时间	80 μ s
内部功耗	0.3W
自身功耗	1.3W
隔离方式	光耦隔离
输出频率	<8kHz
输出动作显示	输出为 ON 状态时，输出指示灯亮
支持服务	固件升级

#### 数 字 量 输出 ( P N P )

项目	规格
电源电压	24V DC
输出类型	数字量输出

输出方式	PNP 型, 有输出时为高电平
输出过流保护	最大 300mA, 最大跳闸电流 600mA
OFF 时最大漏电流	25 $\mu$ A
ON 时响应时间	12 $\mu$ s
OFF 时响应时间	60 $\mu$ s
内部功耗	0.3W
自身功耗	1.3W
隔离方式	光耦隔离
输出频率	<8kHz
输出动作显示	输出为 ON 状态时, 输出指示灯亮
支持服务	固件升级

## 模 拟 量 输入

项目	规格
电源电压	24V DC
输入通道	4
电压输入阻抗	>1M $\Omega$
电压输入量程	双极性-5~5V、-10~10V, 单极性0~5V、0~10V
电流输入量程	0~20mA, 4~20mA
分辨率	16位
采样时间	1ms/通道
精度 (常温25°C)	电压±0.1%, 电流±0.1% (全量程)
精度 (环境温度0~55°C)	电压±0.3%, 电流±0.8%
内部功耗	0.4W
自身功耗	0.7W
隔离方式	光耦隔离
支持服务	固件升级

## 模 拟 量 输出

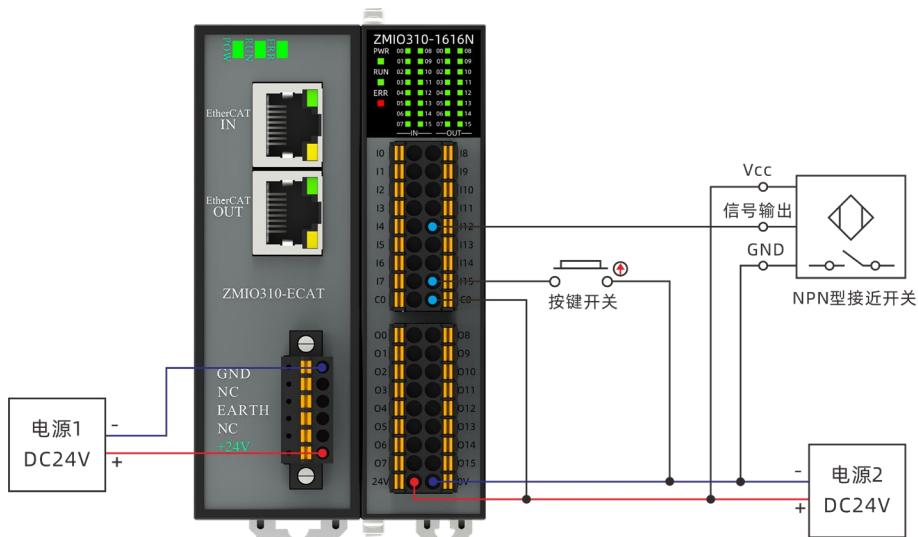
项目	规格
电源电压	24V DC
输出通道	4
电压输出负载	>10K $\Omega$
电压输出范围	双极性-5~5V、-10~10V, 单极性0~5V、0~10V
电流输出范围	0~20mA, 4~20mA
分辨率	16位
转换时间	1ms/通道
精度 (常温25°C)	电压±0.1%, 电流±0.1% (全量程)

精度 (环境温度0~55°C)	电压±0.3%, 电流±0.8%
内部功耗	0.3W
自身功耗	0.9W
隔离方式	光耦隔离
支持服务	固件升级

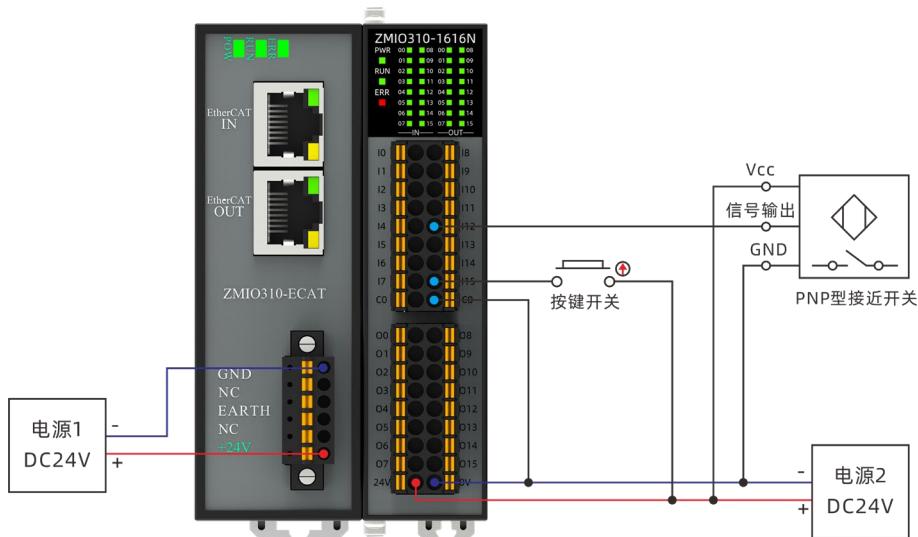
### 3.9 子模块接线

ZMIO310-32DI 模块本身不带有 24V/0V 端子。其输入口类型可通过用户端子上的 C0/C1 端接入耦合器模块的+24V/GND 或其他子模块的 24V/0V 来切换。

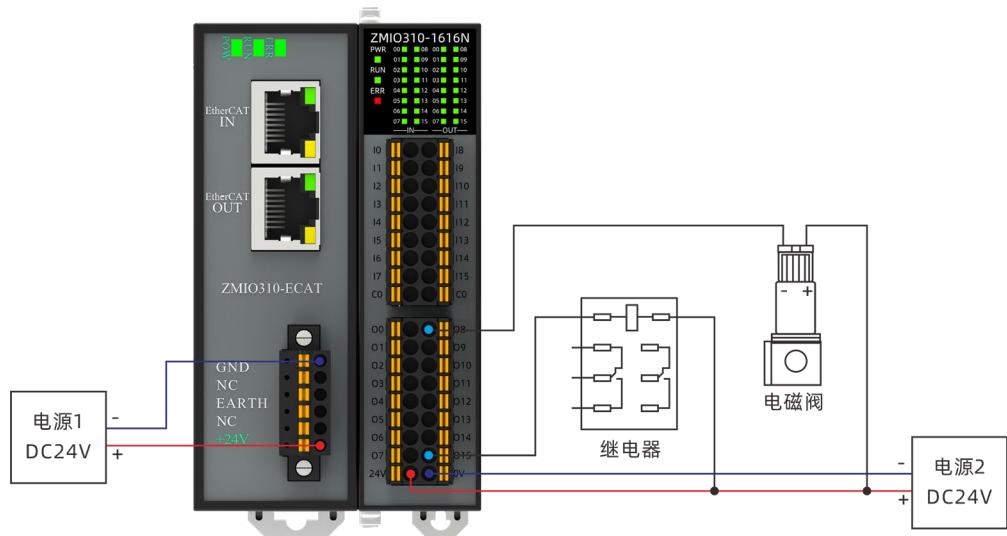
#### 数字量输入 (NPN)



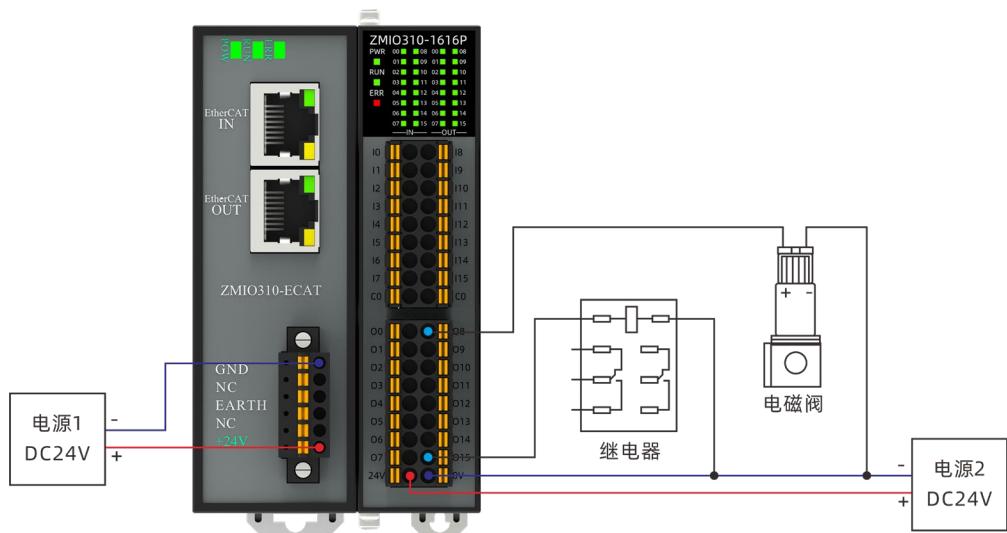
#### 数字量输入 (PNP)



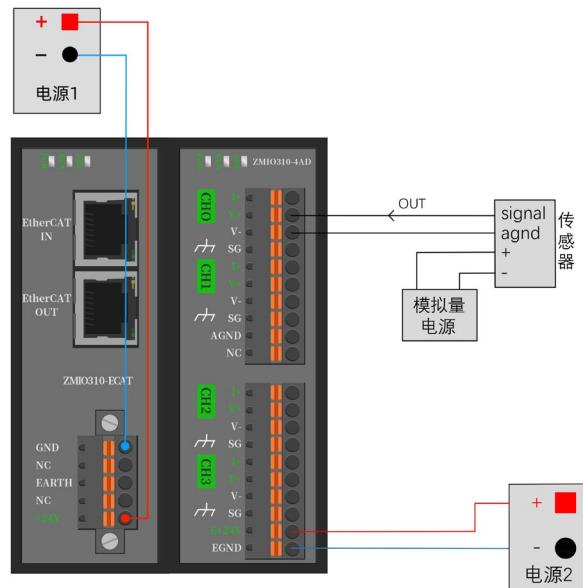
## 数 字 量 输出 ( N P N )



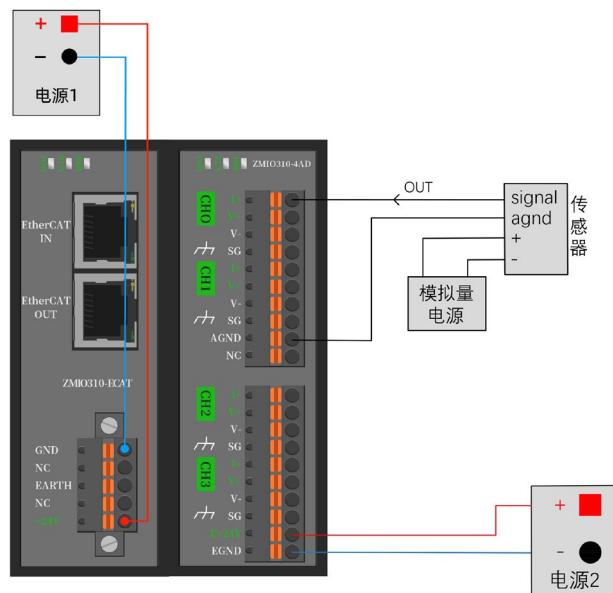
## 数 字 量 输出 ( P N P )



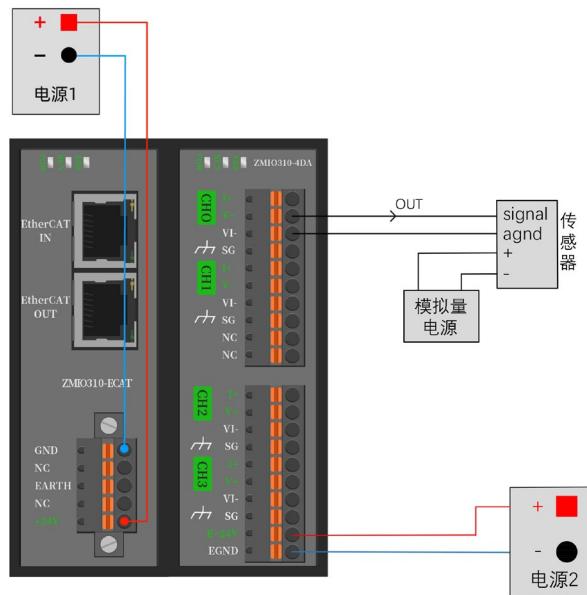
## 模拟量输入 (电压型)



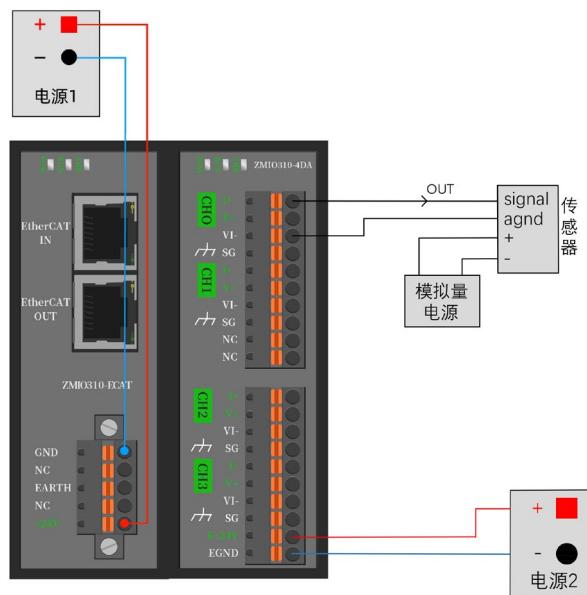
## 模拟量输入 (电流型)



## 模拟量输出 (电压型)



## 模拟量输出 (电流型)



## 第四章 使用说明

### 4.1 功耗计算示例

示例的耦合器模块选用 ZMI0310-ECAT 通讯模块为例。

ZMI0310-ECAT 通讯模块外部供电 DC24V, 内部耦合器向各模块供电 5V, 最大输出 2A 电流。除去自身的内部功耗 1.6W, 还可以向各扩展子模块提供 8.4W。可参考下图各模块系统侧功耗。

ZMI0310-ECAT	ZMI0310-1616N	ZMI0310-1616P	ZMI0310-32DI	ZMI0310-4AD	ZMI0310-4DA	...
ECAT通讯模块	输入/输出模块	输入/输出模块	输入模块	AD模块	DA模块	...
1.6W	0.3W	0.3W	0.3W	0.4W	0.3W	...

### 4.2 IO 起始编号设置

#### EtherCAT 总线扩展

若耦合器使用 ECAT 通讯模块时, 数字量输入/输出模块 IO 起始编号通过总线指令 NODE\_IO 来设置, AD 模块和 DA 模块的 AIO 起始编号通过总线指令 NODE\_AIO 来设置。

数字量:

NODE_IO			
使用语法	NODE_IO(slot, node)=iobase		
参数列表	slot	控制器总线槽位号	缺省0
	node	设备编号	从0开始
	iobase	IO起始编号	扩展的输入、输出起始编号一致
使用示例	NODE_IO(0, 0)=32 ‘节点0扩展的IO起始编号为32		
注意事项	NODE_IO指令只能将IO起始编号设为8的倍数, 如0、8、16…等等, 若IO起始编号为30, 实际将设为24。 先通过查看控制器状态, 明确控制器本地IO的最大值, 再使用NODE_IO避开重 复。若扩展的IO编号与本地IO重合, 二者将同时起作用, 不建议这样做。 IN要有更新才变, 没有更新的区域就是之前的值, 例如当起始地址 16, 偏移到 32, 前面 16~31 区域没有实际 IO 对应, 区域内 IN 状态就不会被更新。		

模拟量:

NODE_AIO			
使用语法	NODE_AIO(slot, node)=aiobase		
参数列表	slot	控制器总线槽位号	缺省 0
	node	设备编号	从 0 开始
	aiobase	AIO 起始编号	扩展的 AD、DA 起始编号一致
使用示例	NODE_AIO(0, 0)=32 ‘节点 0 扩展的 AIO 起始编号为 32		
注意事项	先通过查看控制器状态, 明确控制器本地 AIO 的最大值, 再使用 NODE_AIO 避开重 复。若扩展的 AIO 编号与本地 AIO 重合, 二者将同时起作用, 不建议这样 做。		

## CAN 总线扩展

若耦合器使用 CAN 通讯模块时, 数字量输入/输出模块的 IO 起始编号、AD 模块和 DA 模块的 AIO 起始编号通过拨码开关设置, 拨码开关 1-4 用于配置 IO 地址, 5-6 用于配置 CAN 通讯速度。

### 数字量 IO 起始编号设置:

拨码 1-4 来设置地址组合值, 参考已有 IO 编号, 控制器再根据地址组合值来设置对应的扩展模块的 IO、AIO 起始编号, 拨码每位 OFF 时对应值 0, ON 时对应值 1, 地址组合值=拨码  $4 \times 8 +$  拨码  $3 \times 4 +$  拨码  $2 \times 2 +$  拨码 1;

如控制器本身包含 28 个 IN, 16 个 OP, 那么第一个扩展板设置的起始地址应超过最大值 28, 按下图规则应将拨码设置为组合值 1(二进制组合值 0001, 从右往左对应拨码 1-4, 此时拨码 1 置 ON, 其他置 OFF), 扩展板上的 IO 编号=扩展板编号值+起始 IO 编号值, 其中, 29-31 空缺出来的 IO 编号舍去不用。后续的扩展板则依次按 IO 点数继续确认拨码设置。

数字量起始 IO 映射编号从 16 开始, 按 16 的倍数递增。

数字 IO 映射编号如下表:

拨码 4	拨码 3	拨码 2	拨码 1	地址 ID	起始 IO 编号	结束 IO 编号
0	0	0	0	0	16	31
0	0	0	1	1	32	47
0	0	1	0	2	48	63
0	0	1	1	3	64	79
0	1	0	0	4	80	95
0	1	0	1	5	96	111
0	1	1	0	6	112	127
0	1	1	1	7	128	143
1	0	0	0	8	144	159
1	0	0	1	9	160	175
1	0	1	0	10	176	191
1	0	1	1	11	192	207
1	1	0	0	12	208	223
1	1	0	1	13	224	239
1	1	1	0	14	240	255
1	1	1	1	15	256	271

### 模拟量 IO 起始编号设置:

模拟量 AD 起始 IO 映射编号从 8 开始, 按 8 的倍数递增。模拟量 DA 起始 IO 映射编号从 4 开始, 按 4 的倍数递增。

模拟量 IO 映射编号如下表, 1-4 位拨码状态与对应地址 ID 可参考上表:

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

#### 通讯速度设置：

拨码 5-6 选择 CAN 总线通讯速度，速度组合值=拨码 6×2+拨码 5×1，组合值范围 0-3。

对应的速度如下所示：

拨码 6	拨码 5	拨码 5-6 组合值	CAN 通讯速度
0	0	0	500Kbps (缺省值)
0	1	1	250Kbps
1	0	2	125Kbps
1	1	3	1Mbps

注意：请根据实际运用场合选择 CAN 通讯速度，考虑因素有传输距离、延迟时间、电磁干扰等。建议波特率默认用 500Kbps。

## 4.3 数字量及模拟量读写

扩展的 I0、AD 和 DA 可以通过使用输入输出相关指令 IN、OP、AIN、AOUT 来操作。

相关内容如下：

类型	具体型号	相关指令	权限	相关视图
输入 IN	ZMI0310-1616N ZMI0310-1616P ZMI0310-32DI	IN	只读	输入口视图

输出OUT	ZMI0310-1616N ZMI0310-1616P ZMI0310-32DO ZMI0310-32DOP	OP	读/写	输出口视图
模拟量AD	ZMI0310-4AD	AIN	只读	AD/DA视图
模拟量DA	ZMI0310-4DA	AOUT	读/写	AD/DA视图

查询及设置 AD, DA 模块量程可用 CANIO\_INFO(can\_id, 17, 扩展子模块地址) 指令;

指令详情请参阅《RTBasic 编程手册》。

## 4.4 子模块配置功能

配置功能仅耦合器使用 ECAT 通讯模块进行扩展的才有效，都需要使用总线指令 SDO\_WRITE 来写入 SDO 消息实现，且使用 SDO\_READ 指令来读取 SDO 消息。

### 写数据字典

SDO_WRITE		
使用语法	SDO_WRITE(slot, node, index, subindex, type, value)	
参数列表	slot	总线槽位号
	node	设备编号
	index	数据字典编号
	subindex	子编号
	type	数据类型
	value	数据值

### 读数据字典

SDO_READ		
使用语法	SDO_READ(slot, node, index, subindex, type, tablenum)	
参数列表	slot	总线槽位号
	node	设备编号
	index	数据字典编号
	subindex	子编号
	type	数据类型
	tablenum	读取的数据存储的TABLE位置

Type 数据类型：根据数据字典描述的数据类型填入 type 值。

type 值	对应数据类型
1	boolean
2	integer 8
3	integer 16

---

4	integer 32
5	unsigned 8
6	unsigned 16
7	unsigned 32

## 第五章 数据字典说明

数据字典仅 ECAT 通讯模块才能创建, CAN 通讯模块无数据字典。

### 5.1 数据字典概览

下表为 ZMI0310-ECAT 通讯模块所有字典概览:

注: 扩展子模块地址按接入耦合器的顺序, 从 0 依次编号, 例如, 耦合器连接的第一个扩展子模块, 地址为 0; 第二个扩展子模块, 地址为 1; 依此类推。

索引	子索引	描述
5000h  (5001+扩展子模块地址)h 例: (5001+a)h=500bh	-	设置设备工作模式
	00h	工作模式类别
	01h	掉电后恢复初始状态, 或掉电保持状态
	02h	预留
(6000+10*扩展子模块地址)h 例: (6000+a*10)h=60a0h	-	控制字典
	00h	配置的类型数目
	01h	配置DA/AD模块量程类型, 或获取DA/AD模块量程类型
	02h	配置AD模块通道开关
	03h	预留
	04h	预留
(7000+10*扩展子模块地址)h 例: (7000+a*10)h=70a0h	-	状态字典
	00h	ZMI0310-1616N、ZMI0310-1616P模块状态字典子索引数目
	01h	获取ZMI0310-1616N、ZMI0310-1616P模块通道的输入状态值
(6001+10*扩展子模块地址)h 例: (6001+a*10)h=60a1h	-	控制字典
	00h	ZMI0310-1616N、ZMI0310-1616P模块控制字典子索引数目
	01h	配置ZMI0310-1616N、ZMI0310-1616P模块通道的输出状态
	-	状态字典
	00h	AD模块状态字典子索引数目
(7001+10*扩展子模块地址)h 例: (7001+a*10)h=70a1h	01h	获取AD模块通道0的输入状态值
	02h	获取AD模块通道1的输入状态值
	03h	获取AD模块通道2的输入状态值
	04h	获取AD模块通道3的输入状态值
	-	控制字典
	00h	DA模块控制字典子索引数目
(6002+10*扩展子模块地址)h	01h	配置DA模块通道0的输出值
	02h	配置DA模块通道1的输出值
	03h	配置DA模块通道2的输出值
	04h	配置DA模块通道3的输出值
	-	状态字典

例: (6002+a*10) h=60a2h	00h	ZMI0310-32DI模块状态字典子索引数目
	01h	获取ZMI0310-32DI模块CH0-15通道的输入状态值
	02h	获取ZMI0310-32DI模块CH16-31通道的输入状态值
(7002+10*扩展子模块地址) h 例: (7002+a*10) h=70a2h	-	控制字典
	00h	ZMI0310-32DO、ZMI0310-32DOP模块控制字典子索引数目
	01h	配置 ZMI0310-32DO、ZMI0310-32DOP模块CH0-15通道的输出状态值
	02h	配置 ZMI0310-32DO、ZMI0310-32DOP模块CH16-31通道的输出状态值

## 5.2 本地后级扩展地址说明

耦合器模块上电后，会扫描本地后级扩展接口，为扫描出的每一个扩展子模块分配一个扩展地址。

扩展案例：

如 ECAT 通讯模块 (ZMI0310-ECAT) 的本地后级接口依次接入 7 个扩展子模块 (2 个 ZMI0310-1616N 子模块、2 个 ZMI0310-32DI 子模块、1 个 ZMI0310-32DO 子模块、1 个 ZMI0310-4AD 模块和 1 个 ZMI0310-4DA 模块) 的情况。且槽位号和设备号都默认为 0。

### 子模块地址分配

后续数据字典章节中指令例子按照如下子模块地址。

ECAT 通讯模块上电开始扫描并分配地址，子模块分配的地址从 0 开始，根据接入的顺序来分配。按 16 进制计算，第一块输入模块的扩展地址为 0，DA 模块的扩展地址为 6。

模块顺序	型号名称	分配地址
ECAT耦合器	ZMI0310-ECAT	/
第一个子模块	ZMI0310-1616N	0
第二个子模块	ZMI0310-1616N	1
第三个子模块	ZMI0310-32DI	2
第四个子模块	ZMI0310-32DI	3
第五个子模块	ZMI0310-32DO	4
第六个子模块	ZMI0310-4AD	5
第七个子模块	ZMI0310-4ADA	6

- 此扩展地址用于数据字典的创建，地址分配不受扩展子模块的类型影响。
- CAN 通讯模块仅分配地址用于内部控制。

## 模块对应的字典

地址	备注	字典描述	
		规则	字典
-	-	-	5000h
0	第1个ZMI0310-1616N	输入口状态字典: (6000+10*扩展子模块地址)h	6000h 7000h
1	第2个ZMI0310-1616N	输出口控制字典: (7000+10*扩展子模块地址)h	6010h 7010h
2	第1个ZMI0310-32DI	状态字典: (6002+10*扩展子模块地址)h	6022h
3	第2个ZMI0310-32DI		6032h
4	第1个ZMI0310-32DO	控制字典: (7002+10*扩展子模块地址)h	7042h
5	第1个ZMI0310-4AD	状态字典: (6001+10*扩展子模块地址)h	6051h
		控制字典: (5001+扩展子模块地址)h	5006h
6	第1个ZMI0310-4DA	控制字典: (7001+10*扩展子模块地址)h	7061h
		控制字典: (5001+扩展子模块地址)h	5007h

以下按各个扩展子模块分别说明其数据字典的内容，及各个功能的具体配置方法。

该地址 0~6 对应的数据字典仅对应扩展模块接入时才会创建，其他模块接入时不会创建。

## 5.3 数据字典详细说明

本小节内容为 ZMI0310-ECAT 通讯模块所有字典详细说明。

### 格式说明

索引 (hex)	子索引 (hex)	对象名	缺省值	数据范围	数据类型	权限

索引 (hex)：对象的索引号，用一个四位十六进制数表示。

子索引 (hex)：对象的子索引号，用一个两位十六进制数表示。

对象名：对象的名称。对于子索引来说，是子索引的名称。

缺省值：默认设置的值。

数据范围：对于只读对象，是读取范围。对于读写对象，是设置范围。

数据类型：对象的数据类型。

权限：确定对象是只读还是读写的。

### 索引：5000h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限
5000h	00h	CONFIG_DATA	2	2	UNSIGNED8	只读
		•表示工作模式类别，即默认的子索引数目； •固定值为2.				

		CONFIG_1_INDENT	2	1或2	UNSIGNED16	读/写
	01h	•表示掉电后恢复初始状态或保持状态; •数据值的用途: 1: SLOT_START总线开启后, 通讯中断, DO/DOP模块和DA模块的输出状态恢复初始值; 2: 通讯中断, DO/DOP 模块和 DA 模块的输出状态保持当前状态。				
	02h	预留				

## 通讯中断后输出状态配置

使用指令	SDO_WRITE																
指令语法	<pre>SDO_WRITE(slot, node, index, subindex, type, value) slot: 总线槽位号, 缺省 0 node: 设备编号, 从 0 开始 index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表</pre> <table border="1"> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> <tr> <td>4</td> <td>integer 32</td> </tr> <tr> <td>5</td> <td>unsigned 8</td> </tr> <tr> <td>6</td> <td>unsigned 16</td> </tr> <tr> <td>7</td> <td>unsigned 32</td> </tr> </table> <p>value: 数据值</p>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于在主从站通讯中断后, DA/DO/DOP的输出状态是否保持, 共有两种模式设置, 缺省为模式2。 模式1: 通讯中断后, 输出状态为初始值(输出为0, DA为当前量程最小值); 模式2: 通讯中断后, 输出状态为断开时的状态。																
使用对象	数字量输出模块: ZMI0310-1616N、ZMI0310-1616P、ZMI0310-32DO、ZMI0310-32DOP 模拟量输出模块: ZMI0310-4DA <b>该功能对当前耦合器扩展的输出和 DA 有效, 其他耦合器扩展的输出和 DA 无效</b>																
数据字典	5000h																
使用示例	SDO_WRITE(0, 0, \$5000, 1, 6, 1)      '配置为模式1 SDO_WRITE(0, 0, \$5000, 1, 6, 2)      '配置为模式2																

索引: 5001h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限
(5001+扩展子模块地址)h	00h	CONFIG_DATA	2	4	UNSIGNED8	只读
	•默认的子索引数目; 固定值为2.					
	CONFIG_1_INDENT	2	2~7和10~15	UNSIGNED16	读/写	
	•配置模拟量模块量程类型, 或读取量程类型值; •数据值的含义:					
	AD	模块类型	数据值	量程类型		
		2	0~10V	DA	10	
		3	-10~10V		11	
		-10~10V				

		4	4~20mA		12	4~20mA																																					
		5	0~20mA		13	0~20mA																																					
		6	0~5V		14	0~5V																																					
		7	-5~5V		15	-5~5V																																					
		CONFIG_2_INDENT		15	0~15		UNSIGNED16 读/写																																				
		<ul style="list-style-type: none"> <li>配置AD模块通道数据；</li> <li>数据值模型，即对应AD模块的四个通道：</li> </ul>		<table border="1"> <thead> <tr> <th>AD通道</th><th>CH3</th><th>CH2</th><th>CH1</th><th>CH0</th></tr> </thead> <tbody> <tr> <td>值 (16进制)</td><td>8</td><td>4</td><td>2</td><td>1</td></tr> </tbody> </table>				AD通道	CH3	CH2	CH1	CH0	值 (16进制)	8	4	2	1																										
AD通道	CH3	CH2	CH1	CH0																																							
值 (16进制)	8	4	2	1																																							
		<ul style="list-style-type: none"> <li>每个数据值的含义：</li> </ul>		<table border="1"> <thead> <tr> <th>数据值</th><th>含义</th></tr> </thead> <tbody> <tr><td>0</td><td>全通道关闭</td></tr> <tr><td>1</td><td>通道0开启</td></tr> <tr><td>2</td><td>通道1开启</td></tr> <tr><td>3</td><td>通道0、1开启</td></tr> <tr><td>4</td><td>通道2开启</td></tr> <tr><td>5</td><td>通道0、2开启</td></tr> <tr><td>6</td><td>通道1、2开启</td></tr> <tr><td>7</td><td>通道0、1、2开启</td></tr> </tbody> </table>		数据值	含义	0	全通道关闭	1	通道0开启	2	通道1开启	3	通道0、1开启	4	通道2开启	5	通道0、2开启	6	通道1、2开启	7	通道0、1、2开启	<table border="1"> <thead> <tr> <th>数据值</th><th>含义</th></tr> </thead> <tbody> <tr><td>8</td><td>通道3开启</td></tr> <tr><td>9</td><td>通道0、3开启</td></tr> <tr><td>10</td><td>通道1、3开启</td></tr> <tr><td>11</td><td>通道0、1、3开启</td></tr> <tr><td>12</td><td>通道2、3开启</td></tr> <tr><td>13</td><td>通道0、2、3开启</td></tr> <tr><td>14</td><td>通道1、2、3开启</td></tr> <tr><td>15</td><td>全通道开启</td></tr> </tbody> </table>		数据值	含义	8	通道3开启	9	通道0、3开启	10	通道1、3开启	11	通道0、1、3开启	12	通道2、3开启	13	通道0、2、3开启	14	通道1、2、3开启	15	全通道开启
数据值	含义																																										
0	全通道关闭																																										
1	通道0开启																																										
2	通道1开启																																										
3	通道0、1开启																																										
4	通道2开启																																										
5	通道0、2开启																																										
6	通道1、2开启																																										
7	通道0、1、2开启																																										
数据值	含义																																										
8	通道3开启																																										
9	通道0、3开启																																										
10	通道1、3开启																																										
11	通道0、1、3开启																																										
12	通道2、3开启																																										
13	通道0、2、3开启																																										
14	通道1、2、3开启																																										
15	全通道开启																																										
	02h	03h 预留																																									
		04h 预留																																									

## 通道使能配置

使用指令	SDO_WRITE																
指令语法	<p>SDO_WRITE(slot, node, index, subindex, type, value)</p> <p>slot: 总线槽位号, 缺省0</p> <p>node: 设备编号, 从0开始</p> <p>index: 数据字典编号</p> <p>subindex: 子编号</p> <p>type: 数据类型, 参考 Type 数据类型表</p> <table border="1"> <thead> <tr> <th>type 值</th><th>对应数据类型</th></tr> </thead> <tbody> <tr><td>1</td><td>boolean</td></tr> <tr><td>2</td><td>integer 8</td></tr> <tr><td>3</td><td>integer 16</td></tr> <tr><td>4</td><td>integer 32</td></tr> <tr><td>5</td><td>unsigned 8</td></tr> <tr><td>6</td><td>unsigned 16</td></tr> <tr><td>7</td><td>unsigned 32</td></tr> </tbody> </table> <p>value: 数据值</p>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	<p>此功能用于控制AD模块的输入通道是否开启使能, 共有16种通道使能方式设置, 缺省为全通道开启使能</p> <p>一个四位二进制数代表四个通道的使能状态, 每一位为一个通道, 最低位为通道0, 最高位为通道3</p>																
使用对象	模拟量输入模块: ZMI0310-4AD																

	该功能对当前的 AD 模块有效, 其耦合器扩展的其他 AD 模块无效
数据字典	5006h: (5001+对应的AD模块的扩展地址编号) h
使用示例	' 5006h为AD模块控制字典 SDO_WRITE(0, 0, \$5006, 2, 6, 0)      ' 全通道关闭 SDO_WRITE(0, 0, \$5006, 2, 6, 3)      ' 通道0、1开启 SDO_WRITE(0, 0, \$5006, 2, 6, 12)     ' 通道2、3开启 SDO_WRITE(0, 0, \$5006, 2, 6, 15)     ' 全通道开启

## 量程切换配置

使用指令	SDO_WRITE																
指令语法	SDO_WRITE(slot, node, index, subindex, type, value) slot: 总线槽位号, 缺省 0 node: 设备编号, 从 0 开始 index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表																
	<table border="1"> <thead> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> <tr> <td>4</td> <td>integer 32</td> </tr> <tr> <td>5</td> <td>unsigned 8</td> </tr> <tr> <td>6</td> <td>unsigned 16</td> </tr> <tr> <td>7</td> <td>unsigned 32</td> </tr> </tbody> </table> value: 数据值	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于切换AD模块的输入通道（或DA模块的输出通道）的量程, 两种模块缺省值都为0~10V的量程 量程与扩展子模块的类型编号相对应, 请参考下表																
使用对象	模拟量输入模块: ZMI0310-4AD 模拟量输出模块: ZMI0310-4DA																
数据字典	5006h: (5001+对应的AD模块的扩展地址编号) h 5007h: (5001+对应的DA模块的扩展地址编号) h																
使用示例	' 5007h为DA模块控制字典, 5006h为AD模块控制字典 SDO_WRITE(0, 0, \$5006, 1, 6, 3)      ' AD模块切换成-10~10V量程 SDO_WRITE(0, 0, \$5006, 1, 6, 5)      ' AD模块切换成0~20mA量程 SDO_WRITE(0, 0, \$5007, 1, 6, 11)     ' DA模块切换成-10~10V量程 SDO_WRITE(0, 0, \$5007, 1, 6, 13)     ' DA模块切换成0~20mA量程																

注意: 相同数据字典, 子索引不一致时, 两条指令之间需要延时。如下图程序:

```

SDO_WRITE(0, 0, $5007, 1, 6, 4)  ' AD 模块切换成 4~20mA 量程
DELAY(10)  ' 都是设置数据字典$5007所以要加延时
SDO_WRITE(0, 0, $5007, 2, 6, 15)  ' 全通道开启

' 设置数据字典$5007和$5008不是同一个数据字典, 所以可以不加延时
' 模拟量输出的量程设置(模拟量输出没有通道使能的概念)
SDO_WRITE(0, 0, $5008, 1, 6, 12)  ' AD 模块切换成 4~20mA 量程

```

类型编号说明:

类型编号	类型名称	模块类型	对应量程
1	ZM1111	输入模块	—
2	ZM1112	AD模块	0~10V

3	ZM1113	AD模块	-10~10V
4	ZM1114	AD模块	4~20mA
5	ZM1115	AD模块	0~20mA
6	ZM1116	AD模块	0~5V
7	ZM1117	AD模块	-5~5V
9	ZM2111	输出模块	-
10	ZM2112	DA模块	0~10V
11	ZM2113	DA模块	-10~10V
12	ZM2114	DA模块	4~20mA
13	ZM2115	DA模块	0~20mA
14	ZM2116	DA模块	0~5V
15	ZM2117	DA模块	-5~5V

- 该功能对当前的 AD 模块（或 DA 模块）有效，其耦合器扩展的其他 AD 模块（或 DA 模块）无效。
- 一块 AD 模块（或 DA 模块）不可使用多量程，其所有通道使用同一量程，不同块可以进行多量程搭配。请根据实际使用情况切换量程。
- 类型编号 1 和 9 是输入模块和输出模块固定使用的，不可使用。
- AD 模块仅可使用类型编号 2~7，DA 模块仅可使用类型编号 10~15。
- 使用 AD 模块时，输入电压（或电流）超过量程范围，将稳定在最大值。

索引：6000h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限				
(6000+10*扩展子模块地址)h	00h	IN_GENERIC	1	1	UNSIGNED8	只读				
		•默认的子索引数目； 固定值为1。								
01h	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	只读				
		•获取ZMIO310-1616N、ZMIO310-1616P模块通道的输入状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输入状态； 例：获取到的输入值为0x0FF0，通道状态表示内容如下：								
		通道	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		bit(2进制)	0	0	0	0	1	1	1	1
		16进制	0			F				
		通道	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		bit(2进制)	1	1	1	1	0	0	0	0
		16进制	F			0				

获取 ZMIO310-1616N/1616P 模块的通道输入状态值

使用指令	SDO_READ
指令语法	SDO_READ(slot, node, index, subindex, type, tablenum) slot: 总线槽位号, 缺省 0 node: 设备编号, 从 0 开始 index: 数据字典编号

	<p>subindex：子编号 type：数据类型，参考 Type 数据类型表</p> <table border="1"> <thead> <tr> <th>type 值</th><th>对应数据类型</th></tr> </thead> <tbody> <tr><td>1</td><td>boolean</td></tr> <tr><td>2</td><td>integer 8</td></tr> <tr><td>3</td><td>integer 16</td></tr> <tr><td>4</td><td>integer 32</td></tr> <tr><td>5</td><td>unsigned 8</td></tr> <tr><td>6</td><td>unsigned 16</td></tr> <tr><td>7</td><td>unsigned 32</td></tr> </tbody> </table> <p>tablenum：读取的数据存储的TABLE位置</p>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于读取ZMI0310-1616N/1616P模块的通道输入状态值； 其中一个ZMI0310-1616N/1616P模块有16个输入通道，即通道0~15； 四位16进制数转为十六位二进制数；每一位表示每个通道的输入状态。																
使用对象	ZMI0310-1616N、ZMI0310-1616P																
数据字典	6000h、6010h： (6000 +对应的ZMI0310-1616N/1616P模块的扩展编号*10)h																
使用示例	<p>’6000h、6010h为ZMI0310-1616N/1616P模块状态字典 ’读取到的数据存储到TABLE位置，需要使用TABLE命令获取内容： SDO_READ (0, 0, \$6000, 1, 6, 100) ’获取第一个ZMI0310-1616N模块的16个通道输入状态 ? TABLE(100) ’打印第一个ZMI0310-1616N模块通道状态 SDO_READ (0, 0, \$6010, 1, 6, 100) ’或取第二个ZMI0310-1616N模块的16个通道输入状态 ? TABLE(100) ’打印第二个ZMI0310-1616N模块通道状态</p>																

索引：7000h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限																																																						
	00h	IN_GENERIC	1	1	UNSIGNED8	只读																																																						
•默认的子索引数目； 固定值为1。																																																												
	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	读/写																																																						
•获取ZMI0310-1616N、ZMI0310-1616P模块通道的输出状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输出状态； 例：获取到的输出值为0x0FF0，通道状态表示内容如下：																																																												
<table border="1"> <thead> <tr> <th>通道</th><th>CH15</th><th>CH14</th><th>CH13</th><th>CH12</th><th>CH11</th><th>CH10</th><th>CH9</th><th>CH8</th></tr> </thead> <tbody> <tr> <td>bit(2进制)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr> <td>16进制</td><td colspan="3">0</td><td colspan="5">F</td></tr> <tr> <th>通道</th><th>CH7</th><th>CH6</th><th>CH5</th><th>CH4</th><th>CH3</th><th>CH2</th><th>CH1</th><th>CH0</th></tr> <tr> <td>bit(2进制)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>16进制</td><td colspan="3">F</td><td colspan="5">0</td></tr> </tbody> </table>							通道	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	bit(2进制)	0	0	0	0	1	1	1	1	16进制	0			F					通道	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0	bit(2进制)	1	1	1	1	0	0	0	0	16进制	F			0				
通道	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8																																																				
bit(2进制)	0	0	0	0	1	1	1	1																																																				
16进制	0			F																																																								
通道	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0																																																				
bit(2进制)	1	1	1	1	0	0	0	0																																																				
16进制	F			0																																																								

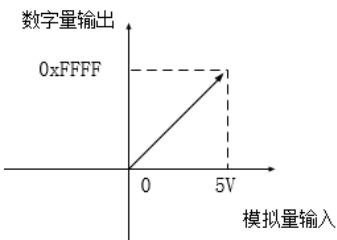
### 配置 ZMI0310-1616N/1616P 模块的通道输入状态值

使用指令	SDO_WRITE
指令语法	SDO_WRITE(slot, node, index, subindex, type, value) slot：总线槽位号, 缺省 0 node：设备编号, 从 0 开始

	<p>index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表</p> <table border="1"> <thead> <tr> <th>type 值</th><th>对应数据类型</th></tr> </thead> <tbody> <tr> <td>1</td><td>boolean</td></tr> <tr> <td>2</td><td>integer 8</td></tr> <tr> <td>3</td><td>integer 16</td></tr> <tr> <td>4</td><td>integer 32</td></tr> <tr> <td>5</td><td>unsigned 8</td></tr> <tr> <td>6</td><td>unsigned 16</td></tr> <tr> <td>7</td><td>unsigned 32</td></tr> </tbody> </table> <p>value: 数据值</p>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于配置 ZMI0310-1616N/1616P模块的通道输入状态值; 其中一个ZMI0310-1616N/1616P模块有16个输出通道, 即通道0~15; 四位16进制数转为十六位二进制数; 每一位表示每个通道的输入状态。																
使用对象	ZMI0310-1616N、ZMI0310-1616P																
数据字典	7000h、7010h: (7000 +对应的ZMI0310-1616N/1616P模块的扩展编号*10)h																
使用示例	<ul style="list-style-type: none"> <li>7000h、7010h为ZMI0310-1616N/1616P模块控制字典</li> <li>配置第一个ZMI0310-1616N模块高8位的输出通道状态为ON, 低8位的通道状态为OFF SDO_WRITE(0, 0, \$7000, 1, 6, 0xFF00)</li> <li>配置第二个ZMI0310-1616N模块高8位的输出通道状态为OFF, 低8位的通道状态为ON SDO_WRITE(0, 0, \$7010, 1, 6, 0x00FF)</li> </ul>																

**注意:** 若使用 SDO 配置 DO 输出, 只能在 EtherCAT START 前有效, EtherCAT START 后控制器会自动启用 PDO 指令配置 DO 通道输出, PDO 实时性较高, 会覆盖 SDO 配置 DO 通道数据, PDO 命令对应 Basic 指令 OP, OP 具体使用方法请参考《RTBasic 帮助手册》。

## 索引 : 6001h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限	
	00h	IN_GENERIC	4	4	UNSIGNED8	只读	
•默认的子索引数目; 固定值为4。							
	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	只读	
•获取AD通道0的输入状态值; •数据值的含义: 0x0000~0xFFFF表示模拟量的刻度值, 由AD获取到的模拟量转换得到; 例: 若当前AD模块的量程为0~5V, 其表示内容如下:							
							
		<p>AD 转换得到的数字量用 Y 表示; AD 获取到的模拟量值用 X 表示; 由图所示, 可计算出 AD 输入值: <math>(Y-0) / (0xFFFF-0) = (X-0) / (5V-0)</math>, 即</p>					

		$Y = 0xFFFF * X / 5V$ <p>X 值是当前 AD 模块获取到的已知条件, PC 端通过 AD 转换得到 Y 数值, 同样在 PC 端得到 Y 值, 可以通过计算逆推出 X 的值。</p>					
		IN_GEN_INT2	0	0x0000~0xFFFF	UNSIGNED16	只读	
<ul style="list-style-type: none"> <li>• 获取AD通道1的输入状态值;</li> <li>• 数据值的含义:</li> </ul>		<p>0x0000~0xFFFF 表示模拟量的刻度值, 由AD获取到的模拟量转换得到; 例: 若当前AD模块的量程为0~5V, 其表示内容如下:</p>					
02h		<p>AD 转换得到的数字量用 Y 表示; AD 获取到的模拟量值用 X 表示; 由图所示, 可计算出 AD 输入值: <math>(Y-0) / (0xFFFF-0) = [X-(5V)] / [5-(-5V)]</math>, 即</p>					
03h		IN_GEN_INT3	0	0x0000~0xFFFF	UNSIGNED16	只读	
<ul style="list-style-type: none"> <li>• 获取AD通道2的输入状态值;</li> <li>• 数据值的含义:</li> </ul>		<p>0x0000~0xFFFF 表示模拟量的刻度值, 由AD获取到的模拟量转换得到; 例: 若当前AD模块的量程为0~20mA, 其表示内容如下:</p>					
04h		<p>AD 转换得到的数字量用 Y 表示; AD 获取到的模拟量值用 X 表示; 由图所示, 可计算出 AD 输入值: <math>(Y-0) / (0xFFFF-0) = (X-0) / (20mA-0)</math>, 即</p> $Y = 0xFFFF * X / 20mA$ <p>X 值是当前 AD 模块获取到的已知条件, PC 端通过 AD 转换得到 Y 数值, 同样在 PC 端得到 Y 值, 可以通过计算逆推出 X 的值。</p>					
<ul style="list-style-type: none"> <li>• 获取AD通道3的输入状态值;</li> <li>• 数据值的含义:</li> </ul>		<p>0x0000~0xFFFF 表示模拟量的刻度值, 由AD获取到的模拟量转换得到; 例: 若当前AD模块的量程为0~20mA, 其表示内容如下:</p>					
04h		<p>AD 转换得到的数字量用 Y 表示; AD 获取到的模拟量值用 X 表示;</p>					

		<p>由图所示, 可计算出 AD 输入值: <math>(Y-0) / (0xFFFF-0) = (X-4mA) / (20mA-4mA)</math> 即  <math display="block">Y = 0xFFFF * (X-4mA) / (20mA-4mA)</math> <p>X 值是当前 AD 模块获取到的已知条件, PC 端通过 AD 转换得到 Y 数值, 同样在 PC 端得到 Y 值, 可以通过计算逆推出 X 的值。</p> </p>
--	--	---

## 获取 AD 模块的通道输入状态值

使用指令	SDO_READ																
指令语法	<p>SDO_READ(slot, node, index, subindex, type, tablenum)  slot: 总线槽位号, 缺省 0  node: 设备编号, 从 0 开始  index: 数据字典编号  subindex: 子编号  type: 数据类型, 参考 Type 数据类型表</p> <table border="1" data-bbox="643 685 1151 1055"> <thead> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> <tr> <td>4</td> <td>integer 32</td> </tr> <tr> <td>5</td> <td>unsigned 8</td> </tr> <tr> <td>6</td> <td>unsigned 16</td> </tr> <tr> <td>7</td> <td>unsigned 32</td> </tr> </tbody> </table> <p>tablenum: 读取的数据存储的TABLE位置</p>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于读取AD模块的通道输入状态值; 其中一个AD模块有4个通道, 分别为通道0, 通道1, 通道2, 通道3;																
使用对象	模拟量输入模块: ZMI0310-4AD																
数据字典	6051h: (6001+对应的AD模块的扩展编号*10)h																
使用示例	<p>’6051h为AD模块状态字典  ’读取到的数据存储到TABLE位置, 需要使用TABLE命令获取内容;  SDO_READ (0, 0, \$6051, 1, 6, 100) ’获取通道0的输入状态  ? TABLE(100) ’打印通道0的读取内容  SDO_READ (0, 0, \$6051, 2, 6, 100) ’获取通道1的输入状态  ? TABLE(100) ’打印通道1的读取内容  SDO_READ (0, 0, \$6051, 3, 6, 100) ’获取通道2的输入状态  ? TABLE(100) ’打印通道2的读取内容  SDO_READ (0, 0, \$6051, 4, 6, 100) ’获取通道3的输入状态  ? TABLE(100) ’打印通道3的读取内容</p>																

索引 : 7001h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限
(7001+10*扩展子模块地址)h	00h	IN_GENERIC	4	4	UNSIGNED8	只读
		•默认的子索引数目; 固定值为4。				
	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	读/写
		•获取DA通道0的输出状态值; •数据值的含义: 0x0000~0xFFFF表示模拟量的刻度值, PC端输入的刻度值通过DA转换输出模拟量;				

		<p>例：若当前DA模块的量程为0~10V，其表示内容如下：</p> <p>DA 转换得到的模拟量用 Y 表示； PC 端输入的数字量，即刻度值用 X 表示； 由图所示，可计算出 DA 输出值：<math>(Y-0)/(10V-0) = (X-0)/(0xFFFF-0)</math>，即 <math>Y = 10*X/0xFFFF</math></p> <p>X 值是当前 PC 端输入的已知条件，X 值再通过 DA 转换得到 Y 数值；同样也可先预设 DA 的输出值，再通过计算逆推出 X 的值。</p>					
	02h	<table border="1"> <tr> <td>IN_GEN_INT2</td><td>0</td><td>0x0000~0xFFFF</td><td>UNSIGNED16</td><td>读/写</td></tr> </table> <ul style="list-style-type: none"> <li>• 获取DA通道1的输出状态值；</li> <li>• 数据值的含义： 0x0000~0xFFFF表示模拟量的刻度值，PC端输入的刻度值通过DA转换输出模拟量；</li> </ul> <p>例：若当前DA模块的量程为-10~10V，其表示内容如下：</p> <p>DA 转换得到的模拟量用 Y 表示； PC 端输入的数字量，即刻度值用 X 表示； 由图所示，可计算出 DA 输出值：<math>[Y-(-10V)]/[10V-(-10V)] = (X-0)/(0xFFFF-0)</math>，即 <math>Y = [10V-(-10V)]*X/0xFFFF-10V</math></p> <p>X 值是当前 PC 端输入的已知条件，X 值再通过 DA 转换得到 Y 数值；同样也可先预设 DA 的输出值，再通过计算逆推出 X 的值。</p>	IN_GEN_INT2	0	0x0000~0xFFFF	UNSIGNED16	读/写
IN_GEN_INT2	0	0x0000~0xFFFF	UNSIGNED16	读/写			
	03h	<table border="1"> <tr> <td>IN_GEN_INT3</td><td>0</td><td>0x0000~0xFFFF</td><td>UNSIGNED16</td><td>读/写</td></tr> </table> <ul style="list-style-type: none"> <li>• 获取DA通道2的输出状态值；</li> <li>• 数据值的含义： 0x0000~0xFFFF表示模拟量的刻度值，PC端输入的刻度值通过DA转换输出模拟量；</li> </ul> <p>例：若当前DA模块的量程为0~20mA，其表示内容如下：</p> <p>DA 转换得到的模拟量用 Y 表示； PC 端输入的数字量，即刻度值用 X 表示； 由图所示，可计算出 DA 输出值：<math>(Y-0)/(20mA-0) = (X-0)/(0xFFFF-0)</math>，即 <math>Y = 20mA *X/0xFFFF</math></p> <p>X 值是当前 PC 端输入的已知条件，X 值再通过 DA 转换得到 Y 数值；同样也</p>	IN_GEN_INT3	0	0x0000~0xFFFF	UNSIGNED16	读/写
IN_GEN_INT3	0	0x0000~0xFFFF	UNSIGNED16	读/写			

可先预设 DA 的输出值，再通过计算逆推出 X 的值。				
		IN_GEN_INT4	0	0x0000~0xFFFF
• 获取DA通道3的输出状态值； • 数据值的含义： 0x0000~0xFFFF表示模拟量的刻度值，PC端输入的刻度值通过DA转换输出模拟量； 例：若当前DA模块的量程为4~20mA，其表示内容如下：				
DA 转换得到的模拟量用 Y 表示； PC 端输入的数字量，即刻度值用 X 表示； 由图所示，可计算出 DA 输出值： $(Y-4mA) / (20mA-4mA) = (X-0) / (0xFFFF-0)$ ， 即 $Y = (20mA-4mA) * X / 0xFFFF + 4mA$ X 值是当前 PC 端输入的已知条件，X 值再通过 DA 转换得到 Y 数值；同样也可先预设 DA 的输出值，再通过计算逆推出 X 的值。				

## 配置 DA 模块的通道输出值

使用指令	SDO_WRITE																
指令语法	SDO_WRITE(slot, node, index, subindex, type, value) slot: 总线槽位号, 缺省 0 node: 设备编号, 从 0 开始 index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表																
	<table border="1"> <thead> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> <tr> <td>4</td> <td>integer 32</td> </tr> <tr> <td>5</td> <td>unsigned 8</td> </tr> <tr> <td>6</td> <td>unsigned 16</td> </tr> <tr> <td>7</td> <td>unsigned 32</td> </tr> </tbody> </table> value: 数据值	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于配置DA模块的通道输出值； 其中一个DA模块有4个通道，分别为通道0，通道1，通道2，通道3。																
使用对象	模拟量输出模块：ZMI0310-4DA																
数据字典	7061h: (7001+对应的DA模块的扩展编号*10)h																
使用示例	' 7061h 为 DA 模块控制字典 SDO_WRITE(0, 0, \$7061, 1, 6, 65535) ' 配置通道0的输出值为0xFFFF SDO_WRITE(0, 0, \$7061, 2, 6, 65535) ' 配置通道1的输出值为0xFFFF SDO_WRITE(0, 0, \$7061, 3, 6, 65535) ' 配置通道2的输出值为0xFFFF SDO_WRITE(0, 0, \$7061, 4, 6, 65535) ' 配置通道3的输出值为0xFFFF																

注意：若使用 SDO 指令配置 DA 通道输出，只能在 EtherCAT START 前有效，EtherCAT START 后控制器会自动启用 PDO 指令配置 DA 通道输出，PDO 实时性较高，会覆盖 SDO 配置 DA 通道数据，PDO 命令对应 Basic 指令 AOUT，AOUT 具体使用方法请参考《RTBasic 帮助手册》。

索引：6002h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限				
(6002+10*扩展子模块地址)h	00h	IN_GENERIC	1	1	UNSIGNED8	只读				
		•默认的子索引数目； 固定值为1。								
	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	只读				
		•获取ZMIO310-32DI模块CH0~15通道的输入状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输入状态； 例：获取到的输入值为0x0FF0，通道状态表示内容如下：								
		通道	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		bit(2进制)	0	0	0	0	1	1	1	1
		16进制	0				F			
		通道	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		bit(2进制)	1	1	1	1	0	0	0	0
		16进制	F				0			
	02h	IN_GEN_INT2	0	0x0000~0xFFFF	UNSIGNED16	只读				
		•获取ZMIO310-32DI模块CH16~31通道的输入状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输入状态； 例：获取到的输入值为0x0FF0，通道状态表示内容如下：								
		通道	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
		bit(2进制)	0	0	0	0	1	1	1	1
		16进制	0				F			
		通道	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
		bit(2进制)	1	1	1	1	0	0	0	0
		16进制	F				0			

### 获取 ZMIO310-32DI 模块的通道输入状态值

使用指令	SDO_READ								
指令语法	SDO_READ(slot, node, index, subindex, type, tablenum) slot: 总线槽位号, 缺省0 node: 设备编号, 从0开始 index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表 <table border="1" style="margin-left: 20px;"> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> </table>	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16
type 值	对应数据类型								
1	boolean								
2	integer 8								
3	integer 16								

		4	integer 32	
		5	unsigned 8	
		6	unsigned 16	
		7	unsigned 32	

tablenum: 读取的数据存储的TABLE位置

功能说明	此功能用于读取ZMI0310-32DI模块的通道输入状态值； 其中一个ZMI0310-32DI模块有32个输入通道，即通道0~31； 四位16进制数转为十六位二进制数；每一位表示每个通道的输入状态。			
使用对象	ZMI0310-32DI			
数据字典	6022h、6032h： (6002 +对应的ZMI0310-132DI模块的扩展编号\*10)h			
使用示例	'6022h、6032h为ZMI0310-32DI模块状态字典 '读取到的数据存储到TABLE位置，需要使用TABLE命令获取内容： SDO\_READ (0, 0, \$6022, 1, 6, 100) '获取第一个ZMI0310-32DI模块的0-15通道输入状态 ? TABLE(100) '打印第一个ZMI0310-32DI模块通道状态 SDO\_READ (0, 0, \$6022, 2, 6, 100) '获取第一个ZMI0310-32DI模块的16-31通道输入状态 ? TABLE(100) '打印第一个ZMI0310-32DI模块通道状态 SDO\_READ (0, 0, \$6032, 1, 6, 100) '或取第二个ZMI0310-32DI模块的0-15通道输入状态 ? TABLE(100) '打印第二个ZMI0310-32DI模块通道状态 SDO\_READ (0, 0, \$6032, 2, 6, 100) '或取第二个ZMI0310-32DI模块的16-31通道输入状态 ? TABLE(100) '打印第二个ZMI0310-32DI模块通道状态			

## 索引：7002h

索引	子索引	对象名称/描述	缺省	数据范围	数据类型	权限				
	00h	IN_GENERIC	1	1	UNSIGNED8	只读				
		•默认的子索引数目； 固定值为1。								
	01h	IN_GEN_INT1	0	0x0000~0xFFFF	UNSIGNED16	读/写				
		•获取ZMI0310-32DO、ZMI0310-32DOP模块CH0-15通道的输出状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输出状态； 例：获取到的输出值为0x0FF0，通道状态表示内容如下：								
		通道	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		bit(2进制)	0	0	0	0	1	1	1	1
		16进制	0				F			
		通道	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		bit(2进制)	1	1	1	1	0	0	0	0
		16进制	F				0			
	02h	IN_GEN_INT2	0	0x0000~0xFFFF	UNSIGNED16	读/写				
		•获取ZMI0310-32DO、ZMI0310-32DOP模块CH16-31通道的输出状态值； •数据值的用途： 4位16进制数转为16位2进制数；每1位表示每个通道的输出状态； 例：获取到的输出值为0x0FF0，通道状态表示内容如下：								
		通道	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
		bit(2进制)	0	0	0	0	1	1	1	1
		16进制	0				F			
		通道	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16

		bit(2进制)	1	1	1	1	0	0	0	0	
		16进制		F			0				

### 配置 ZMIO310-32DO/32DOP 模块的通道输入状态值

使用指令	SDO_WRITE																
指令语法	SDO_WRITE(slot, node, index, subindex, type, value) slot: 总线槽位号, 缺省 0 node: 设备编号, 从 0 开始 index: 数据字典编号 subindex: 子编号 type: 数据类型, 参考 Type 数据类型表 <table border="1" data-bbox="643 617 1151 988"> <thead> <tr> <th>type 值</th> <th>对应数据类型</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>boolean</td> </tr> <tr> <td>2</td> <td>integer 8</td> </tr> <tr> <td>3</td> <td>integer 16</td> </tr> <tr> <td>4</td> <td>integer 32</td> </tr> <tr> <td>5</td> <td>unsigned 8</td> </tr> <tr> <td>6</td> <td>unsigned 16</td> </tr> <tr> <td>7</td> <td>unsigned 32</td> </tr> </tbody> </table> value: 数据值	type 值	对应数据类型	1	boolean	2	integer 8	3	integer 16	4	integer 32	5	unsigned 8	6	unsigned 16	7	unsigned 32
type 值	对应数据类型																
1	boolean																
2	integer 8																
3	integer 16																
4	integer 32																
5	unsigned 8																
6	unsigned 16																
7	unsigned 32																
功能说明	此功能用于配置 ZMIO310-1616N/1616P 模块的通道输入状态值; 其中一个 ZMIO310-1616N/1616P 模块有 16 个输出通道, 即通道 0~15; 四位 16 进制数转为十六位二进制数; 每一位表示每个通道的输入状态。																
使用对象	ZMIO310-32DO、ZMIO310-32DOP																
数据字典	7042h: (7002 + 对应的 ZMIO310-32DO/32DOP 模块的扩展编号 * 10) h																
使用示例	' 7042h 为 ZMIO310-32DO/32DOP 模块控制字典 ' 配置第一个 ZMIO310-32DO/32DOP 模块的 CH0-15 通道的高 8 位的输出通道状态为 ON, 低 8 位的通道状态为 OFF SDO_WRITE(0, 0, \$7042, 1, 6, 0xFF00) ' 配置第一个 ZMIO310-32DO/32DOP 模块的 CH16-31 通道的高 8 位的输出通道状态为 ON, 低 8 位的通道状态为 OFF SDO_WRITE(0, 0, \$7042, 2, 6, 0xFF00)																

**注意:** 若使用 SDO 配置 DO 输出, 只能在 EtherCAT START 前有效, EtherCAT START 后控制器会自动启用 PDO 指令配置 DO 通道输出, PDO 实时性较高, 会覆盖 SDO 配置 DO 通道数据, PDO 命令对应 Basic 指令 OP, OP 具体使用方法请参考《RTBasic 帮助手册》。

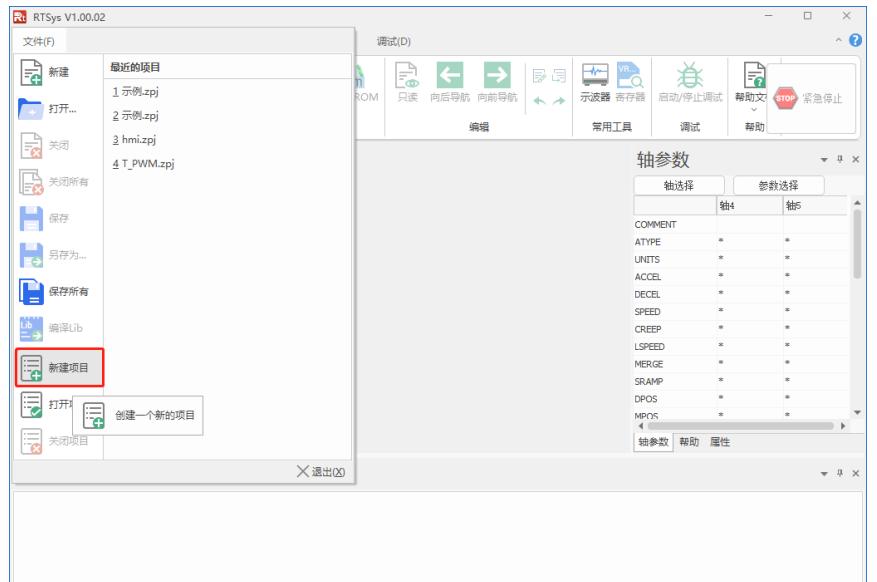
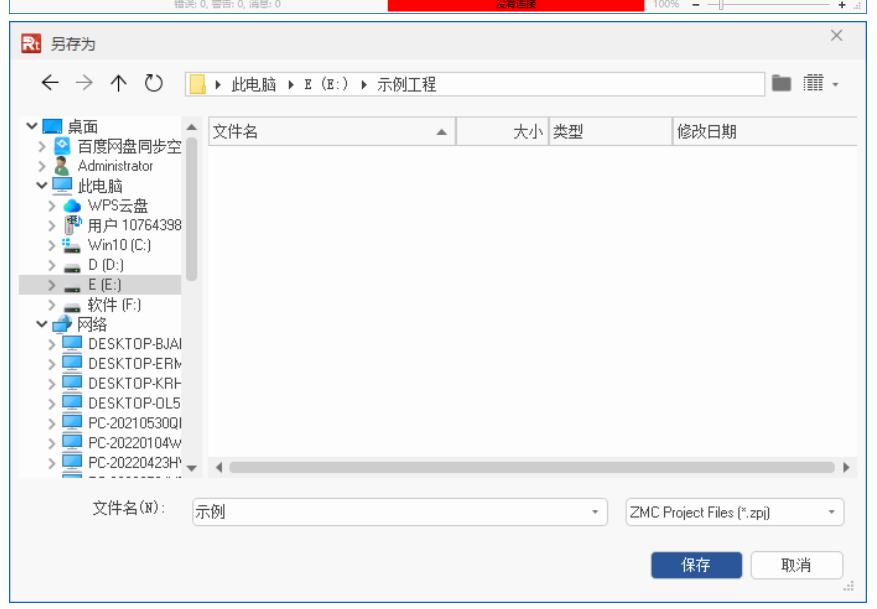
## 第六章 编程应用

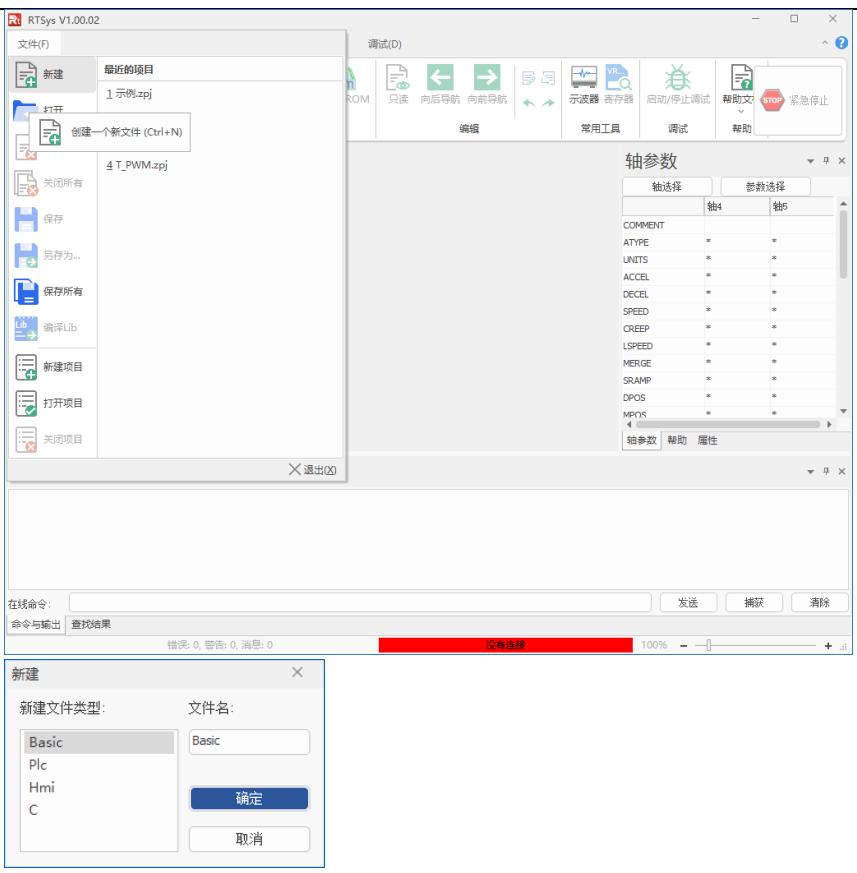
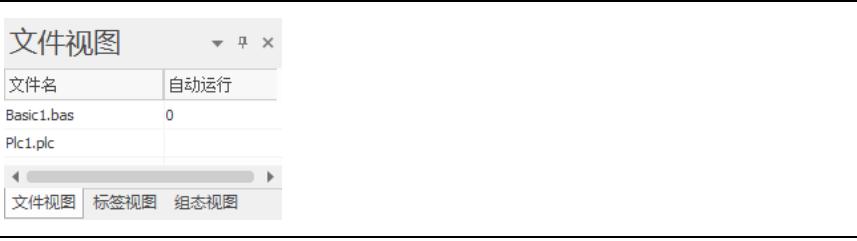
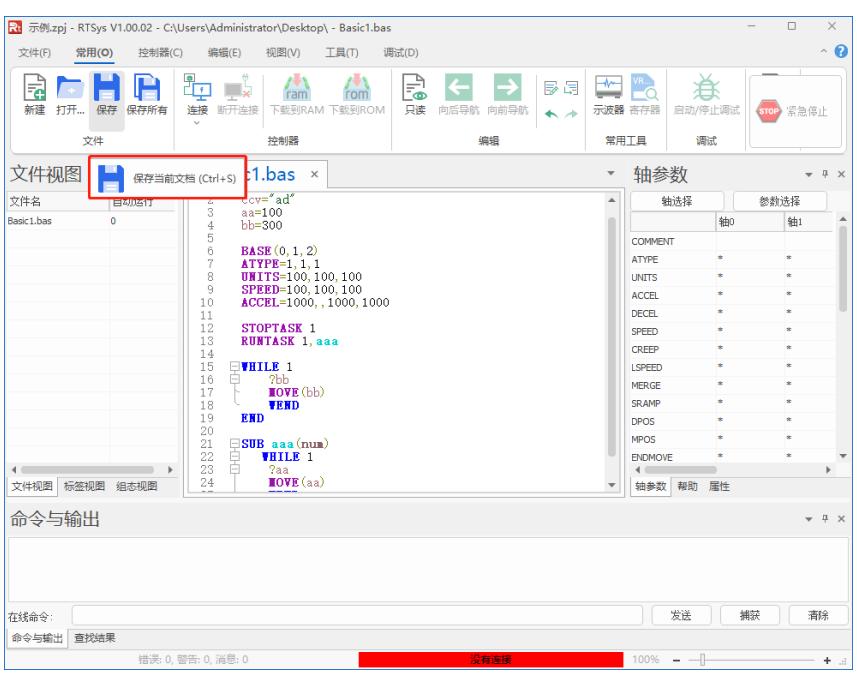
### 6.1 RTSys 软件使用

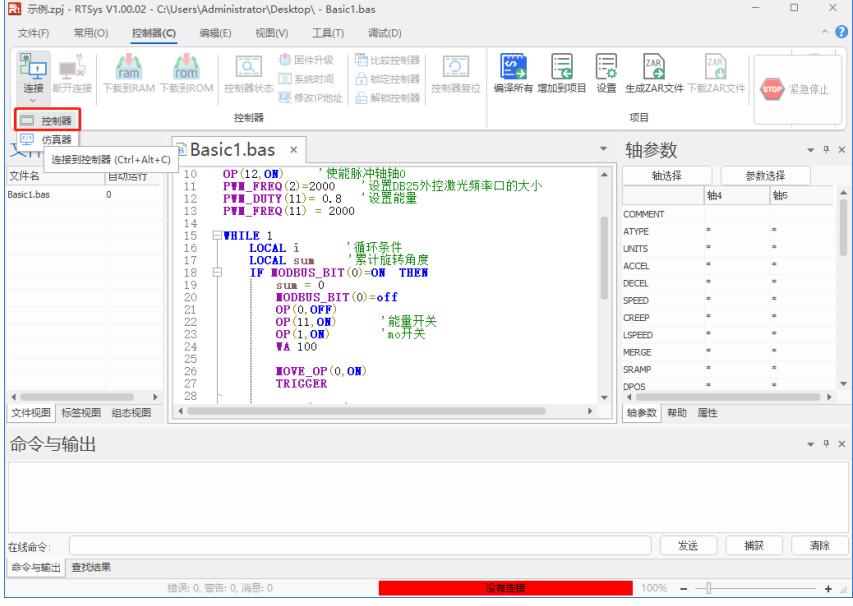
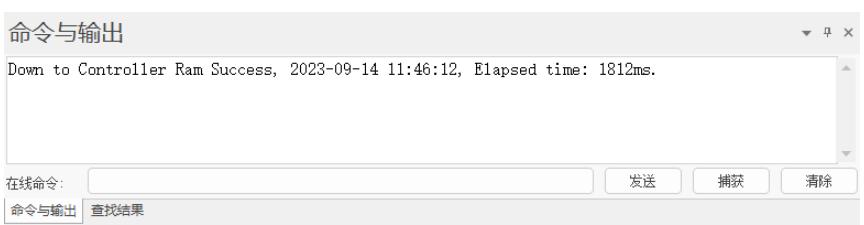
RTSys 是正运动技术 ZMotion 系列扩展模块的 PC 端程序开发调试与诊断软件，通过它用户能够很容易的对控制器进行程序编辑与配置，快速开发应用程序、实时诊断系统运行参数以及对扩展模块正在运行的程序进行实时调试，支持中英双语环境。

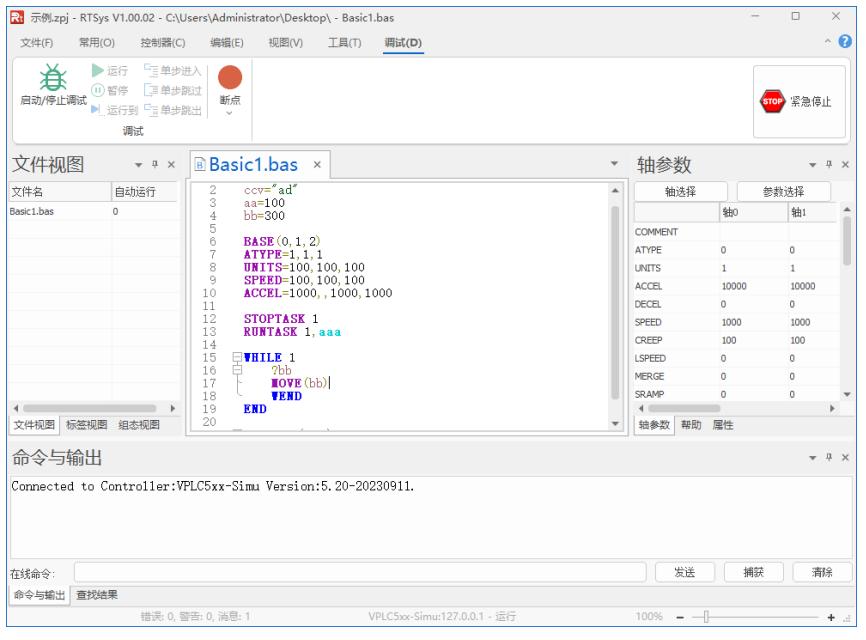
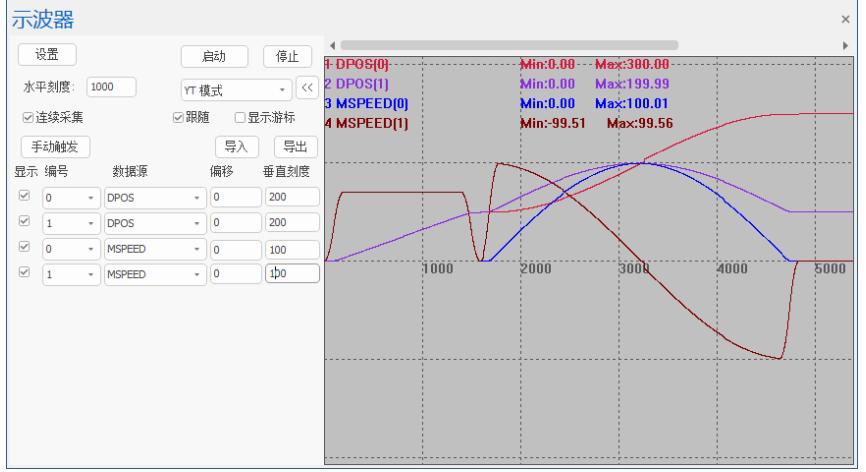
Basic、Plc、Hmi 和 C 语言之间可以多任务运行，其中 Basic 可以多任务号运行，可与 Plc、Hmi 与 C 混合编程。

更新软件版本请前往正运动网站下载，网址：[www.zmotion.com.cn](http://www.zmotion.com.cn)。

步骤	操作	显示界面
1	打开 RTSys 编程软件，菜单栏“文件”-“新建项目”弹出另存为界面，输入文件名后保存形式后缀为“.zpj”的项目文件。	 

2	菜单栏“文件”-“新建文件”，出现右图弹窗，选择新建的文件类型为 basic 后确认。支持 Basic/Plc/Hmi 混合编程。	
3	文件视图窗口双击文件右边自动运行的位置，输入任务号“0”。	
4	在程序输入窗口编辑好程序，点击保存文件，新建的 basic 文件会自动保存到项目 zpj 所在的文件下。保存所有即保存该项目下的所有文件。	

5	<p>点击“控制器”-“连接”-“控制器”，没有控制器时可选择连接到仿真器仿真运行，点击“连接”-“仿真器”。</p>	
	<p>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。</p>	
6	<p>点击菜单栏-“控制器”-“RAM/ROM”-“下载到 RAM”/“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。 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。</p>	<p>成功下载到 RAM:</p>  <p>成功下载到 ROM:</p> 

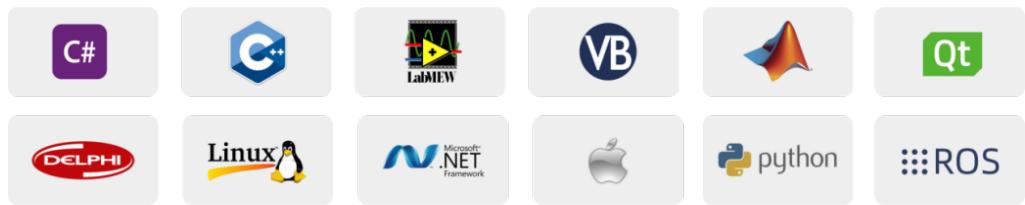
7	<p>点击菜单栏“调试”-“启动/停止调试”调用任务与监视窗口。因为之前下载过了，这里选择附加到当前程序即可。</p>	
8	<p>在菜单栏“工具”-“示波器”打开示波器窗口 示波器使用参见正运动小助手“快速入门   篇九：如何进行扩展模块示波器的应用”。</p>	

说明：

1. 打开工程项目时，需选择打开项目 zpj 文件，若只打开其中的 Bas 文件，程序无法下载到控制器。
2. 不建立项目的时候，只有 Bas 文件无法下载到控制器。
3. 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。
4. 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.**

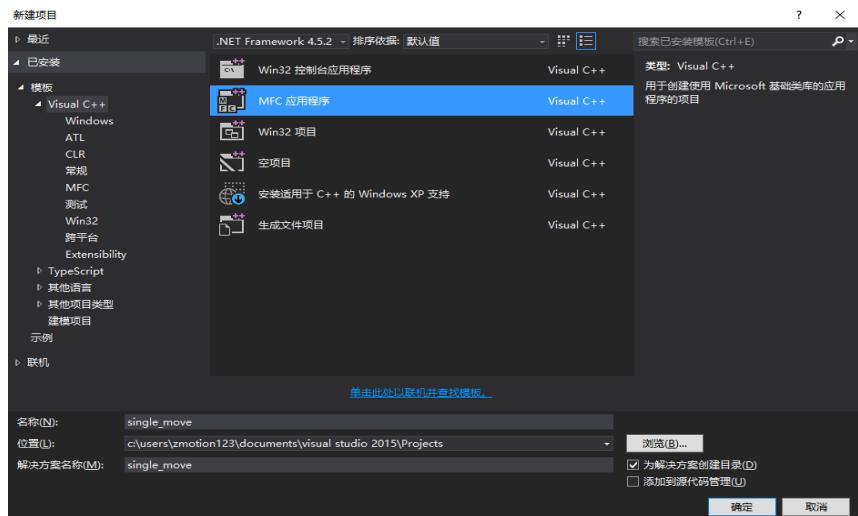
## 6.2 上位机编程应用

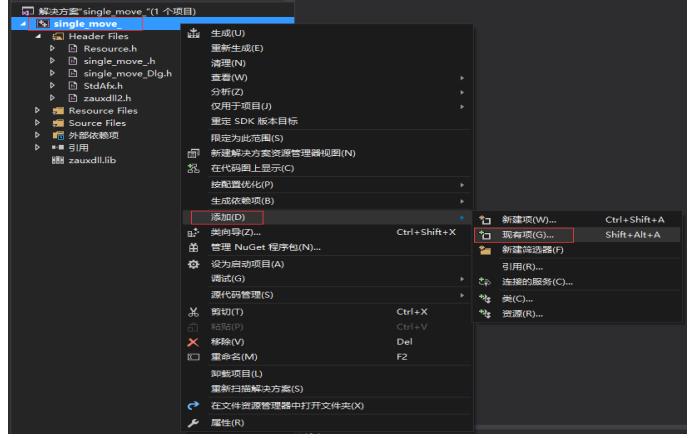
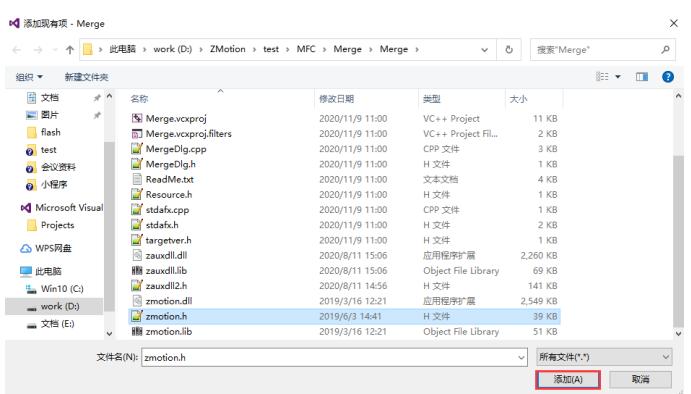
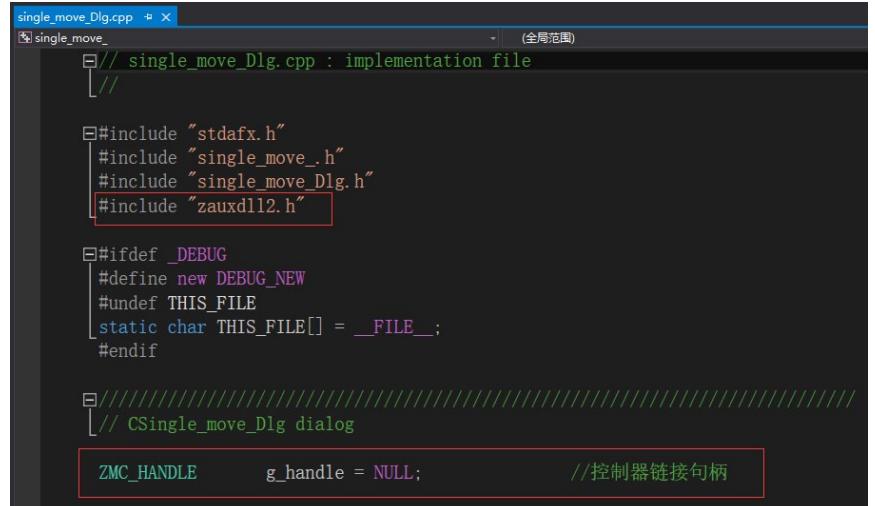
控制器支持 windows, linux, Mac, Android, wince 各种操作系统下的开发，提供 vc, c#, vb.net, labview 等各种环境的 dll 库，如下图。上位机软件编程参考《ZMotion PC 函数库编程手册》。



使用 PC 上位机软件开发的程序无法下载到控制器，通过 d11 动态库连接到控制器，开发时需要将 d11 库添加到头文件中并声明。

VS 中的 c++ 项目开发过程如下：

步骤	操作	显示界面
1	打开 VS，点击菜单“文件”→“新建”→“项目”，启动创建项目向导。	
2	选择开发语言为“Visual C++”和程序类型“MFC 应用程序”。	
3	下一步，选择类型为“基于对话框”，下一步或者完成。	

4	找到厂家提供的光盘资料里面的 C++ 函数库，路径如下(64 位库为例)。	
5	将上述路径下面的所有 DLL 相关库文件复制到新建的项目里面。	
6	<p>在项目中添加静态库和相关头文件。静态库： zauxdll.lib, zmotion.lib 相关头文件： zauxdll2.h, zmotion.h</p>	<p>1) 先右击头文件，接着依次选择：“添加”→“现有项”。</p>  <p>2) 在弹出的窗口中依次添加静态库和相关头文件。</p> 
7	声明相关的头文件和定义控制器连接句柄，至此项目新建完成。	

## 第七章 运行维护

设备正确的运行及维护有利于保证和延长设备本身的生命周期，防止设备性能劣化或降低设备失效的概率，按事先规定的计划或相应技术条件规定进行的技术管理。

### 7.1 定期检查与维护

工作环境等对设备有影响，所以，通常以 6 个月~1 年的检查周期为标准对其做定期检查，可以根据周围环境适当调整设备的检查周期，使其工作在规定的标准环境中。

检查项目	检查内容	检查标准
电源	测量电压是否为额定值	DC24V (-5%~+5%)
周围环境	环境温度是否在规定范围内(柜内安装时，柜内温度即环境温度)	-10°C~55°C
	环境湿度是否在规定范围内(柜内安装时，柜内湿度即环境湿度)	10%~95% 非凝结
	是否有阳光直射	应无
	有无水、油、化学品等的飞沫	应无
	有无粉尘、盐分、铁屑、污垢	应无
	有无腐蚀性气体	应无
	有无易燃、易爆性气体或物品	应无
	设备是否受到振动或冲击	应在耐振动、耐冲击的范围内
	散热性是否良好	应保持良好通风及散热
安装和接线状态	基本单元和扩展单元是否安装牢固	安装螺丝应上紧、无松动
	基本单元和扩展单元的联接电缆是否完全插好	联接电缆不能松动
	外部接线的螺丝是否松动	螺丝应上紧、无松动
	外部接线是否损坏	外部接线不能有任何外观异常

### 7.2 故障排查

常见问题	解决建议
输入口检测不到信号	1. 检查是否需要 IO 电源; 2. 检查信号电平是否与输入口匹配，排查公共端是否相连; 3. 检查输出口编号是否与操作的一致。
输出口操作无响应	1. 检查是否需要 IO 电源; 2. 检查输出口编号是否与操作的一致。
输入通道无电压电流信号	1. 检查是否需要 IO 电源; 2. 检查输出口编号是否与操作的一致。
POWER 灯亮，RUN 灯不亮	1. 检查供电电源功率是否充足，电压是否异常，调整好后重启控制器; 2. ERR 灯是否有规律的闪烁，如果有则可能是硬件故障。
RUN 灯亮，ERR 灯也亮	1. 程序运行错误，请查验 RTSys/ZDevelop 错误代码，检查应用程序; 2. 请检查接线，电阻以及拨码设置是否正确; 3. 控制器的 CANIO ADDRESS 指令是否设置为主端(32)，CAN 通

	讯速度是否一致。
CAN 扩展模块连接不上	<ol style="list-style-type: none"><li>1. 检查 CAN 接线和供电回路, 120 欧姆电阻是否有安装在两端;</li><li>2. 检查主从端配置, 通讯速度配置等;</li><li>3. 检查拨码开关, 是否有多个扩展模块采用同样的 ID;</li><li>4. 干扰严重的场合使用双绞屏蔽线, 使用双电源供电 (双电源扩展模块主电源和 IO 电源分开供电)。</li></ol>

## 第八章 售后服务

### 服务对象

本售后服务条款规定的服务内容适用于在中国市场上通过正运动技术及其授权的合法渠道购买的扩展模块、运动控制卡、扩展模块、人机界面等产品。

### 服务项目

#### 1. 保修期: 12 个月

在保修期内, 如果产品发生非人为故障, 我们为您提供保修服务。请客户联系商务人员并填写《维修申请表》(主要信息如: 产品型号、序列号、故障描述、特殊要求等), 寄到我们公司, 我们将在维修周期内完成维修并寄还给您。

保修期计算方法, 一般按条码管理扫描出库时间作为发货时间(如果客户能提供确切的发货时间证明, 也可以按照该时间作为发货时间)。

#### 2. 换货:

自产品发货之日起 3 个月内, 如果产品发生非人为故障, 我们可以为您更换同型号产品。

#### 3. 终身维护:

我们将为客户提供终身维护服务。在保修期内但不符合保修条件或超过保修期限的故障产品, 我们提供有偿维修服务, 在客户确认接受产品的维修费用后, 我们安排进行产品的维修。但对已经停产的产品, 或缺乏维修物料, 或损坏过于严重无维修价值的返回品则无法提供维修服务。

#### 4. 维修费用:

- 1) 保修期内的产品, 非人为原因引起的故障, 免费维修;
- 2) 超保修期或人为损坏产品收费标准, 我们将根据不同型号和损坏程度收取元件的成本费、人工费和运费; 具体的费用, 由对接的商务人员报价给您;
- 3) 运费: 保修范围内产品运费由我司负担单程, 非保修范围内的产品运费由客户负担;

#### 5. 不享受免费保修的情况:

- 1) 由于火灾、水灾、地震等不可抗力因素造成的产品故障;
- 2) 由于客户安装或者使用不当所导致的损坏;
- 3) 未经正运动技术授权的人员对产品进行了拆卸、维修或者改装造成的产品故障;
- 4) 非正运动技术直销或授权的合法渠道购买的产品;
- 5) 产品的编码撕毁、涂改或者其他原因造成的产品编码无法辨认;

## 附录

### 更新记录

产品型号: ZMI0310(32 点位)系列立式总线扩展模块			
更新日期	版本号	版本(更改)说明	更改人
2025/1/11	V2.0.0	1. 手册发布	xcx

电话

**0755-2606 6955**

传真

**0755-2606 6955**

网站

**www.zmotion.com.cn**

业务咨询专线

**400-089-8936**

技术支持专线

**400-089-8966**

业务咨询邮箱

**sales@zmotion.com.cn**

技术支持邮箱

**support@zmotion.com.cn**

地址

深圳市宝安区西乡洲石路阳光工业园A1栋5楼



正运动技术



正运动小助手

**深圳市正运动技术有限公司**  
Shenzhen Zmotion Technology Co., Ltd.

深圳正运动公司版权所有，相关规格如有变动，恕不另行通知