

插板式模组应用

插板式 EtherCAT 系列从站

AN01010101 1.00 Date:2025/10/22

类别	内容
关键词	插板式EtherCAT模组、Codesys
摘要	本文介绍了使用Codesys对插板式电机驱动模块的应用示例

插板式模组应用

插板式 EtherCAT 系列从站

修订历史

版本	日期	原因
V1.00	2025/10/16	•创建文档

插板式模组应用

插板式 EtherCAT 系列从站

目 录

1. 适用范围	1
2. 原理概述	2
2.1 CiA402 简介	2
2.2 CiA402 状态机	2
2.2.1 状态	2
2.2.2 条件	3
2.3 CiA402 控制字	4
2.3.1 位定义	4
2.3.2 指令编码	5
2.4 CiA402 状态字	5
2.4.1 位定义	5
2.4.2 状态编码	6
2.5 CiA402 运行模式	6
2.5.1 到达和跟随判断	7
2.5.2 CSP 周期同步位置模式	9
2.5.3 PP 轮廓位置模式	10
2.5.4 PV 轮廓速度模式	11
2.5.5 HM 归零模式	12
2.6 CiA402 其他说明	13
2.6.1 CiA402 数字信号	13
3. 开发环境	14
4. 实现过程	15
5. 免责声明	31

插板式模组应用

插板式 EtherCAT 系列从站

1. 适用范围

本文档适用于使用 Codesys 搭配插板式 EtherCAT 电机驱动模组的应用。

产品概述

为满足个性化大型系统的控制需求,致远电子推出了插板式模组。该系统采用 EtherCAT 总线,尺寸小巧,采用 40 脚标准排针接口。用户按需制作分线底板,从站板插在底板上通过 EtherCAT 网络级联,最大支持 255 个节点。

- **以不变应万变**, 只需最少的工作量, 按需制作分线底板; 电机驱动、数字量、模拟量、编码器等多种模组选择组合。
- **小体积大系统**, 在最小的体积下集成最多的从站, 实现大型系统的控制。
- **高精度快布局**, 基于 EtherCAT, 实现高精度分布控制, 以及即插即用快速布局。

产品应用

- ◆ 物流装备
- ◆ 电子制造
- ◆ 新能源
- ◆ 纺织
- ◆ 包装

插板式模组应用

插板式 EtherCAT 系列从站

2. 原理概述

2.1 CiA402 简介

CiA402 是一种主流的电机控制通讯框架协议，使用主从模式，主站控制多台从站（电机）的位置、速度或转矩等。CiA402 最初只在 CANopen 运行，目前已扩展到 Modbus RTU、Modbus TCP 和 EtherCAT 等实时总线。但不管任何总线，只要按 CiA402 流程操作寄存器，就能控制电机。

驱动器上电后，会尝试通过网口连接 EtherCAT 主站，或通过调试口连接 Modbus RTU 主站。如果已连接上 EtherCAT 主站，则先自动进入 EtherCAT 的 Operational 状态，之后按 CiA402 运行电机协议。如果未连接上 EtherCAT 主站，但已连接上 Modbus RTU 主站，则直接运行 CiA402。

2.2 CiA402 状态机

主站连通驱动器后，驱动器按图 2.1 运行 CiA402 工作流程。CiA402 定义了不同地址的对象，例如最重要的控制字和状态字（在 EtherCAT 中的地址分别是 6040h 和 6041h）。主站通过 SDO 或 PDO 写控制字，就能触发不同的跳转条件（例如 T2~T16），让驱动器在不同的状态中切换，同时主站通过读状态字，就能掌握驱动器的基本状态，例如是否有故障。在此基础上，还有了不同功能的对象，例如控制位置的目标位置对象（607Ah）。

2.2.1 状态

在 Start 到 Ready To Switch On 状态，只需接通控制电源、保持通讯即可，无需接通驱动电机的主电源，但电机是未通电使能的。Switched On 状态表示控制电源和主电源都已接通，但电机未能通电使能。Operation Enabled 状态表示控制电源和主电源都已接通，电机已通电上锁。

插板式模组应用

插板式 EtherCAT 系列从站

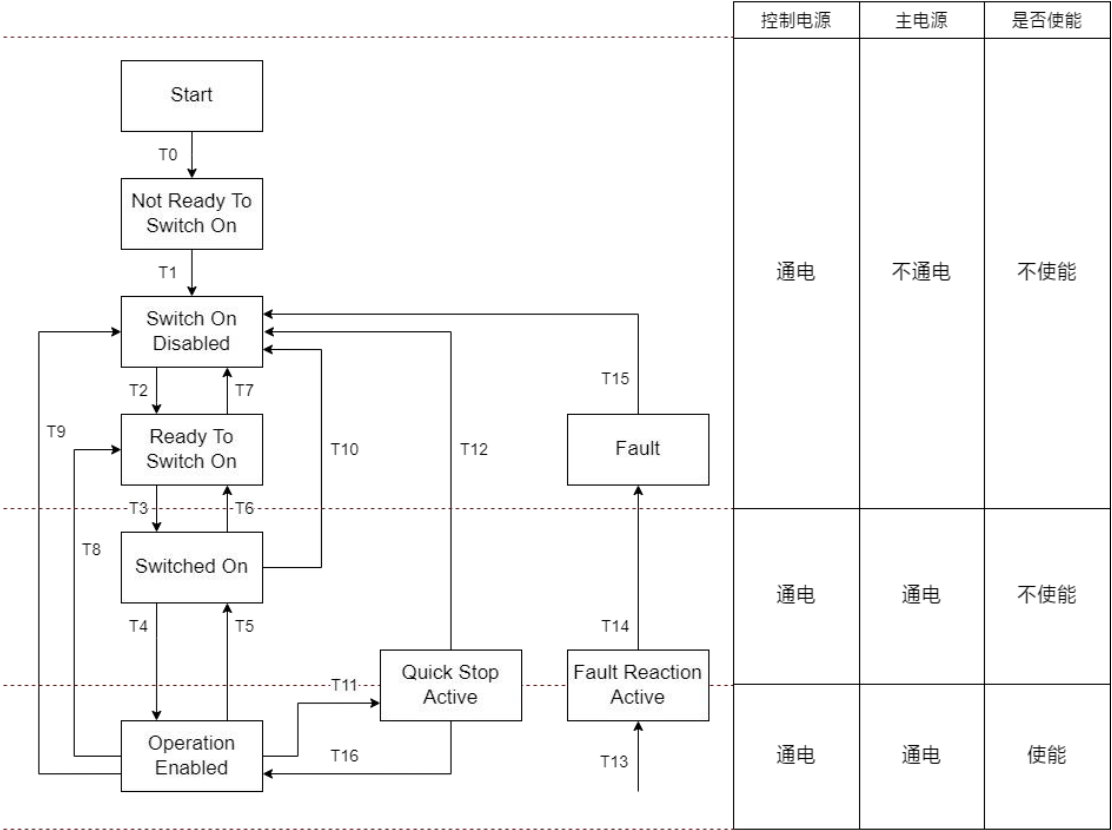


图 2.1 CiA402 工作流程

表 2.1 CiA402 状态

状态	驱动器动作
Not ready to switch on	控制电源已供电，开始初始化参数
Switch on disabled	初始化参数完毕
Ready to switch on	等待主电源供电
Switched on	控制电源和主电源都已供电，驱动器准备好，等待使能
Operation enabled	已被使能，电机可以运行
Quick stop active	电机快速减速停止
Fault reaction active	错误发生但未被处理，电机已停机
Fault	错误已被处理，等待切换到 Switch on disabled 状态

2.2.2 条件

图 2.1 各个跳转条件的说明如表 2.2 所示。控制字用于发送总线指令，状态字的 1~9 位表示跳转后的状态。另外本驱动器不含 STO 使能控制脚，表中 STO 固定为有效。

插板式模组应用

插板式 EtherCAT 系列从站

表 2.2 CiA402 跳转条件

条件	说明	控制字 6040h	状态字 6041h (Bit1~9)
T0	控制电源已供电	自动跳转	0x0000
T1	参数已成功初始化	自动跳转	0x0250
T2	总线发 Shutdown 指令，且 STO 有效	0x0006	0x0231
T3	总线发 Switch On 或 Enable Operation 指令	0x0007	0x0233
T4	自动跳转（T3 已发送 Enable Operation）或 总线发 Enable Operation 指令	0x000F	0x0237/0x1237
T5	总线发 Disable Operation 指令	0x0007	0x0233
T6	总线发 Shutdown 指令	0x0006	0x0231
T7	总线发送 Disable Voltage 或 Quick Stop 指令，或 STO 无效	0x0000 或 0x0002 或自动 跳转	0x0250
T8	总线发送 Shutdown 命令	0x0006	0x0231
T9	总线发 Disable Voltage 指令或 STO 无效	0x0000 或自动跳转	0x0250
T10	总线发送 Disable Voltage 或 Quick Stop 指令或 STO 无效	0x0000 或 0x0002	0x0250
T11	总线发送 Quick Stop 指令	0x0002	0x217
T12	总线发送 Disable Voltage 指令或 STO 无效	0x0000 或自动跳转	0x0250
T13	发生 2 等级故障	自动跳转	0x021F
T14	发生 3 或 4 等级的故障级 或 Fault reaction active 完成退出	自动跳转	0x0218
T15	指令 Fault Reset 指令或 STO 无效	0x0080 或自动跳转	0x0250
T16	总线发送 Enable Operation 指令	0x000F	0x0237/0x1237

2.3 CiA402 控制字

2.3.1 位定义

控制字 Controlword 的地址位于 6040h，主站通过它给从站发出不同的指令组合，各个位的定义如表 2.3 所示。其中 Bit 8 Halt 为 1 时驱动器减速暂停，如果电机正在图 2.1 的 Operation enabled 状态运行，此位为 1 时电机将按当前模式下的减速度进行减速停止，并停

插板式模组应用

插板式 EtherCAT 系列从站

留在 Operation enabled 状态。Bit 7 Fault reset 用于错误复位，如果出现了错误，此位从 0 变为 1 将清除当前的错误，并跳转到图 2.1 的 Switch on disabled 状态，等待再次运行。Bit 15~9 是预留位。Bit 3~0 是组合指令。

表 2.3 控制字位定义

15~9	8	7	6~4	3	2	1	0
N / A	Halt	Fault reset	Operation mode specific	Enable operation	Quick Stop	Enable voltage	Switch on

Bit6~4 是运行模式控制位，仅在 PP 轮廓位置和 HM 归零模式使用，如表 2.4 所示。

表 2.4 运行模式控制位定义

位	PP 轮廓位置	HM 归零
4	上升沿时更新目标位置	上升沿时开始归零
5	0: 到达当前的目标位置后，再往新的目标位置运行 1: 立刻往新的目标位置运行	---
6	0: 目标位置为绝对位置 1: 目标位置为相对位置	---

2.3.2 指令编码

控制字的 Bit 3~0 以及 Bit 7 组成命令代码，实现图 2.1 的跳转条件，如表 2.5 所示。

表 2.5 指令编码

指令名	位组合					跳转条件
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	T2,T6,T8
Switch on	0	0	1	1	1	T3
Switch on + enable operation	0	1	1	1	1	T3 + T4
Disable voltage	0	X	X	0	X	T7,T9,T10,T12
Quick stop	0	X	0	1	X	T7,T10,T11
Disable operation	0	0	1	1	1	T5
Enable operation	0	1	1	1	1	T4,T16
Fault reset	0->1	X	X	X	X	T15

2.4 CiA402 状态字

2.4.1 位定义

状态字 Statusword 的地址位于 6041h，主站通过读取此对象获悉从站的当前状态，各个位的定义如表 2.6 所示。

插板式模组应用

插板式 EtherCAT 系列从站

表 2.6 状态字位定义

位号	名称	说明
0	Ready to switch on	准备启动
1	Switch on	启动，主电源需接通
2	Operation enabled	允许启动，电机已通电上锁
3	Fault	错误
4	Voltage enabled	为 1 表示已接通主电源
5	Quick stop	为 0 表示正在快速停止
6	Switch on disabled	未启动
7	Warning	为 1 表示发生了错误等级为 0 的错误，即发出警告
8	N / A	本产品不使用
9	Remote	为 1 表示控制字的指令已被执行
10	Target reached	为 1 表示已达到目标值，位置、速度、转矩等
11	Internal limit active	本产品不使用
12~15	N / A	本产品不使用

2.4.2 状态编码

状态字的 Bit 0~3 以及 Bit5~6 组成状态编码，每组编码对应图 2.1 的一种状态，如表 2.7 所示。主站查询这些编码器，就能知道从站的当前状态。

表 2.7 状态编码

状态位组合	CiA402 状态
xxxx xxxx x0xx 0000b	Not ready to switch on
xxxx xxxx x1xx 0000b	Switch on disabled
xxxx xxxx x01x 0001b	Ready to switch on
xxxx xxxx x01x 0011b	Switched on
xxxx xxxx x01x 0111b	Operation enabled
xxxx xxxx x00x 0111b	Quick stop active
xxxx xxxx x0xx 1111b	Fault reaction active
xxxx xxxx x0xx 1000b	Fault

2.5 CiA402 运行模式

在进入 Operation enabled 状态后，驱动器将根据运行模式字 Modes of operation（地址 6060h）按不同的模式运行，如表 2.8 所示。本产品支持+8 周期同步位置模式（简称 CSP），+3 轮廓速度模式（PV），+1 轮廓位置模式（PP），+6 归零模式（HM）。

插板式模组应用

插板式 EtherCAT 系列从站

表 2.8 运行模式字

数值	定义
-128 ~ -1	厂家自定义
0	不执行任务
+1	Profile position mode PP 轮廓位置模式
+2	Velocity mode V 速度模式
+3	Profile velocity mode PV 轮廓速度模式
+4	Profile torque mode PT 轮廓转矩模式
+5	保留
+6	Homing mode HM 归零模式
+7	Interpolated position mode IP 插补位置模式
+8	Cyclic sync position mode CSP 周期同步位置模式
+9	Cyclic sync velocity mode CSV 周期同步速度模式
+10	Cyclic sync torque mode CST 周期同步转矩模式
+11 ~ +127	保留

2.5.1 到达和跟随判断

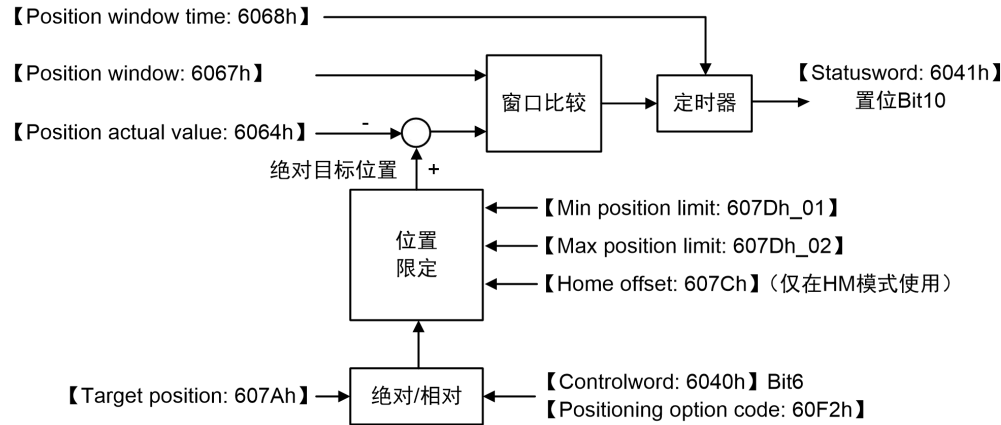
1. 位置到达和跟随

图 2.2 显示了在位置控制相关的模式下，位置到达的判断原理。【Target position: 607Ah】是目标位置，可以是绝对位置，如果控制字【Controlword: 6040h】的 Bit6 为 0 表示绝对位置，如果为 1 且【Positioning option code: 60F2h】为 0，相对上一个目标位置运行，如果【Positioning option code: 60F2h】为 2，相对当前实际位置运行。如果实际位置【Position actual value: 6064h】在设定的位置窗口【Position window: 6067h】范围内，且持续时间长于位置窗口时间【Position window time: 6068h】，就被判断为已到达目标位置，状态字的 Bit 10 被置 1。【Min position limit: 607Dh_01】、【Max position limit: 607Dh_02】用于限定最终的绝对目标位置的范围，【Home offset: 607Ch】是 HM 归零模式使用的偏移量。

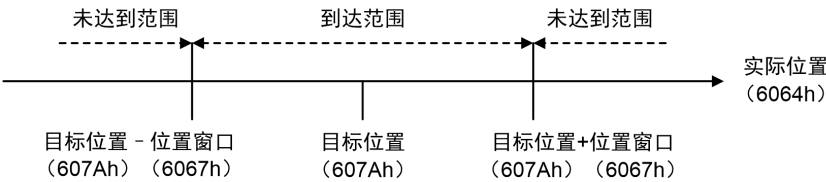
在 CSP 周期同步位置模式下不支持位置到达的判断，仅支持绝对位置控制。

插板式模组应用

插板式 EtherCAT 系列从站



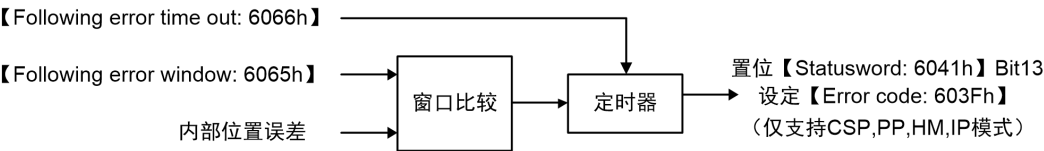
(a) 位置到达框图



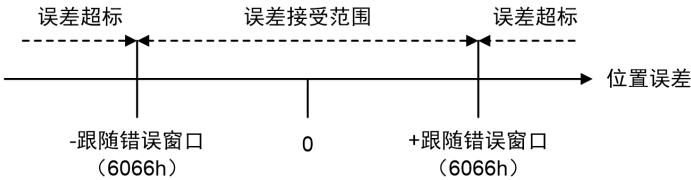
(b) 位置到达范围

图 2.2 位置到达判断

图 2.3 显示了 CSP 周期同步位置、PP 轮廓位置和 HM 归零等模式下位置跟随错误的判断原理。当驱动器内部位置误差大于跟随错误窗口【Following error window: 6065h】，且持续时间长于跟随错误超时实际【Following error time out: 6066h】，则就被判断为跟随错误，并且设定错误码到【Error code: 603Fh】。



(a) 位置跟随错误框图



(b) 位置跟随错误范围

图 2.3 位置跟随错误判断

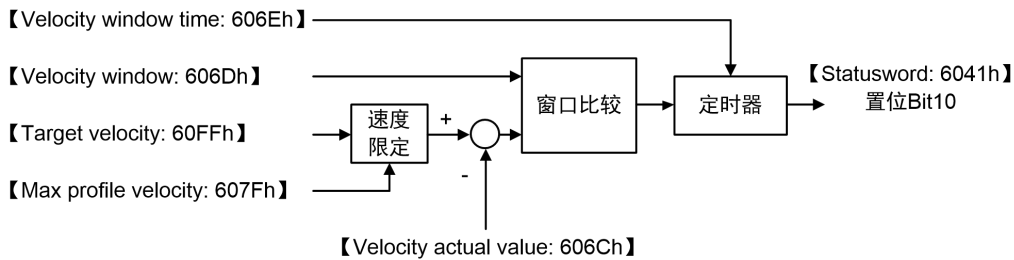
2. 速度到达和零速

图 2.4 显示了在 PV 轮廓速度模式下判断目标速度是否到达的原理。目标速度【Target

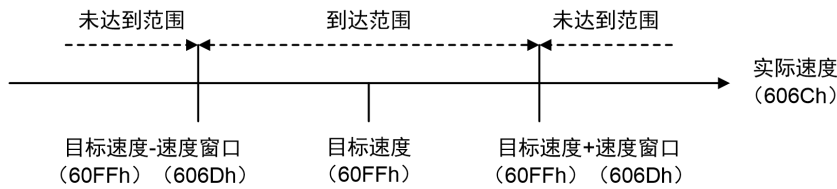
插板式模组应用

插板式 EtherCAT 系列从站

velocity: 60FFh】被限定在【Max profile velocity: 607Fh】的范围内。当目标速度和实际速度【Velocity actual value: 606Ch】的差值，在速度窗口【Velocity window: 606Dh】的范围内，且持续时间长于【Velocity window time: 606Eh】，就被判断为速度到达，状态字的 Bit10 被置位。



(a) 速度到达框图



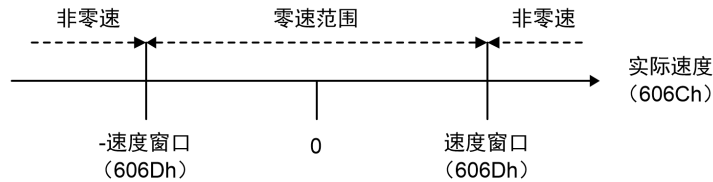
(b) 速度到达范围

图 2.4 速度到达判断

图 2.5 显示了在 PV 轮廓速度模式下零速度的判断原理。当实际速度【Velocity actual value: 606Ch】在速度窗口【Velocity window: 606Dh】的范围内，且持续时间长于【Velocity window time: 606Eh】，就被判断为零速度，状态字的 Bit12 被置位。



(a) 零速判断框图



(b) 零速判断范围

图 2.5 零速判断

2.5.2 CSP 周期同步位置模式

插板式模组应用

插板式 EtherCAT 系列从站

在 CSP 周期同步位置模式下，主站通过 PDO 定时更新目标位置。主站根据需求实现的加减速和匀速过程，精确计算好每个周期的目标位置，从站不负责加减速的处理。本驱动器 CSP 模式仅支持绝对运动，不支持相对运动。

1. SDO 初始化

先按下面的流程，通过 SDO 初始化必要的对象。其中带*的对象可使用默认值，不进行初始化。

- 1) 【Mode of operations: 6060h】设定为周期同步位置模式(0x08)。
- 2) 【Following error window: 6065h】*设定最大的位置误差，单位 PUU。
- 3) 【Following error time out: 6066h】*设定最大误差的允许时间，单位 ms。
- 4) 【Position window: 6067h】*设定目标位置到达范围，单位步脉冲。
- 5) 【Position window time: 6068h】*设定目标位置到达范围的允许时间，单位 ms。
- 6) 【Min position limit: 607Dh_01】*设定目标位置最小限定值，默认-0x7FFFFFFF。
- 7) 【Max position limit: 607Dh_02】*设定目标位置最大限定值，默认 0x7FFFFFFF。

其中【Following error window: 6065h】设定最大允许的误差，即当前给定位置 and 实际位置的差值，如果误差一直大于此值，并且持续时间长于【Following error time out: 6066h】，就报错停机，先后进入 Fault reaction active 和 Fault 状态，【Statusword: 6041h】状态寄存器的 Fault 位置 1，【Error code: 603Fh】显示对应的错误码。往【Controlword: 6040h】写入 0x0080，触发 T15，返回 Switch on disabled 状态，可按上面流程重新开始执行。

2. PDO 控制

初始化完成后，再使用如下 PDO 控制电机，如表 2.9 所示。【Target position: 607Ah】，设定目标位置【Controlword: 6040h】依次写入 0x06->0x07->0x0F，进入 Operation enabled 状态。电机将运行到目标位置。

注：以整步角度 1.8°，256 细分为例：如果【Target position: 607Ah】设置为 51200((360/1.8 (整步角度)) * 256 (细分))，即电机转动一圈。细分数可以在 2000h 对象组中修改。

表 2.9 CSP 模式 PDO

方向	对象	名称	数据类型	单位	备注
RxPDO 驱动接收	【Controlword: 6040h】	控制字	UINT16	---	必选
	【Target position: 607Ah】	目标位置	INT32	微步	必选
TxPDO 驱动发送	【Statusword: 6041h】	状态字	UINT16	---	必选
	【Error code: 603Fh】	错误码	UINT16	---	可选
	【Position actual value: 6064h】	实际位置	INT32	微步	必选
	【Digital inputs: 60FDh】	数字信号输入	UINT32	---	可选

2.5.3 PP 轮廓位置模式

在 PP 轮廓位置位置模式下，主站通过 PDO 更新目标位置，驱动器自行实现加减速和匀速过程，最终到达目标位置。PP 模式支持绝对运动和相对运动。

1. SDO 初始化

先按下面的流程，通过 SDO 初始化必要的对象。其中带*的对象可使用默认值，不进行

插板式模组应用

插板式 EtherCAT 系列从站

初始化。

- 1) 【Mode of operations: 6060h】设定为轮廓位置模式模式(0x01)。
- 2) 【Following error window: 6065h】*设定最大的位置误差，单位 PUU。
- 3) 【Following error time out: 6066h】*设定最大误差的允许时间，单位 ms。
- 4) 【Position window: 6067h】*设定目标位置到达范围，单位步脉冲。
- 5) 【Position window time: 6068h】*设定目标位置到达范围的允许时间，单位 ms。
- 6) 【Profile jerk use: 60A4】*设定 Jerk 参数，仅使能 S 曲线时有效，单位 RPM/s/s。
- 7) 【Max profile velocity: 607Fh】设定最高限定速度，单位 0.1RPM。
- 8) 【Positioning option code: 60F2】*设定相对位置的方式，【Controlword: 6040h】的 Bit6 为 1 时表示相对运动，本对象有效，否则无效；本对象默认为 0 表示相对前一个目标位置运行，为 1 表示相对当前位置运行。(暂不支持相对运动)
- 9) 【Min position limit: 607Dh_01】*设定目标位置最小限定值，默认-0x7FFFFFFF。
- 10) 【Max position limit: 607Dh_02】*设定目标位置最大限定值，默认 0x7FFFFFFF。

2. PDO 控制

初始化完成后，再使用如下 PDO 控制电机，如表 2.10 所示。【Controlword: 6040h】的 Bit6~4 需按表 2.4 设定，Bit6 和 Bit5 设定绝对运动和相对运动，【Target position: 607Ah】设定目标位置。【Profile velocity: 6081h】设定轮廓速度，【Profile acceleration: 6083h】设定加速度，【Profile deceleration: 6084h】设定减速度。最后【Controlword: 6040h】依次写入 0x06->0x07->0x0F，进入 Operation enabled 状态。

随后电机将以设定的加速度加速到目标速度，最终以设定的减速度停在目标位置。当【Controlword: 6040h】的 Bit4 出现 0->1，则按新的配置运行到新的目标位置；往整个字写入 0x10F 电机减速停止。

表 2.10 PP 模式 PDO

方向	对象	名称	数据类型	单位	备注
RxPDO 驱动接收	【Controlword: 6040h】	控制字	UINT16	---	必选
	【Target position: 607Ah】	目标位置	INT32	微步	必选
	【Profile velocity: 6081h】	轮廓速度	UINT32	0.1RPM	必选
	【Profile acceleration: 6083h】	加速度	UINT32	RPM/s	必选
	【Profile deceleration: 6084h】	减速度	UINT32	RPM/s	必选
TxPDO 驱动发送	【Statusword: 6041h】	状态字	UINT16	---	必选
	【Error code: 603Fh】	错误码	UINT16	---	必选
	【Position actual value: 6064h】	实际位置	INT32	微步	必选
	【Velocity actual value: 606C】	实际速度	INT32	0.1RPM	必选

2.5.4 PV 轮廓速度模式

在 PV 轮廓速度模式下，让电机匀速转动。

插板式模组应用

插板式 EtherCAT 系列从站

1. SDO 初始化

先按下面的流程，通过 SDO 初始化必要的对象。其中带*的对象可使用默认值，不进行初始化。

- 1) 【Mode of operations: 6060h】设定为轮廓速度模式(0x03)。
- 2) 【Velocity window: 606Dh】*设定目标速度到达范围，单位 0.1RPM。
- 3) 【Velocity window time: 606Eh】*设定目标速度到达范围的允许时间，单位 ms。
- 4) 【Profile jerk use: 60A4】*设定 Jerk 参数，仅使能 S 曲线时有效，单位 RPM/s/s。
- 5) 【Max profile velocity: 607Fh】*设定最高限定速度，单位 0.1RPM。

2. PDO 控制

初始化完成后，再使用如下 PDO 控制电机，如表 2.11 所示。【Target velocity: 60FFh】设定不同的目标速度，【Profile acceleration: 6083h】设定加速度，【Profile deceleration: 6084h】设定减速度。最后【Controlword: 6040h】控制字，依次写入 0x06->0x07->0x0F，进入 Operation enabled 状态。电机将以设定的加速度加速到目标速度，【Target velocity: 60FFh】目标速度，设定为 0 时，电机以该减速度减速停止。

表 2.11 PV 模式 PDO

方向	对象	名称	数据类型	单位	备注
RxPDO 驱动接收	【Controlword: 6040h】	控制字	UINT16	---	必选
	【Target velocity: 60FFh】	目标速度	INT32	0.1RPM	必选
	【Profile acceleration: 6083h】	加速度	UINT32	RPM/s	可选
	【Profile deceleration: 6084h】	减速度	UINT32	RPM/s	可选
TxPDO 驱动发送	【Statusword: 6041h】	状态字	UINT16	---	必选
	【Error code: 603Fh】	错误码	UINT16	---	可选
	【Position actual value: 6064h】	实际位置	INT32	微步	必选
	【Velocity actual value: 606Ch】	实际速度	INT32	0.1RPM	可选

2.5.5 HM 归零模式

在 HM 归零模式下，让电机找到零点。

1. SDO 初始化

先按下面的流程，通过 SDO 初始化必要的对象。其中带*的对象可使用默认值，不进行初始化。

- 1) 【Mode of operations: 6060h】设定为归零模式(0x06)。
- 2) 【Following error window: 6065h】*设定最大的位置误差，单位 PUU。
- 3) 【Following error time out: 6066h】*设定最大误差的允许时间，单位 ms。
- 4) 【Position window: 6067h】*设定目标位置到达范围，单位步脉冲。
- 5) 【Position window time: 6068h】*设定目标位置到达范围的允许时间，单位 ms。
- 6) 【Controlword: 6040h】依次写入 0x06->0x07->0x0F，进入 Operation enabled 状态。

插板式模组应用

插板式 EtherCAT 系列从站

2. PDO 控制

初始化完成后，再使用如下 PDO 控制电机，如表 2.12 所示。设定【Homing speed switch: 6099h_01】寻找归零开关的最高速度，【Homing speed zero: 6099h_02】离开归零开关的最高速度，【Homing acceleration: 609Ah】归零的加减速。【Controlword: 6040h】的 Bit4 需按表 2.4 设定，出现 0->1 变化开始运动，1->0 暂停运动。【Homing method: 6098h】设定归零模式，17 使用正限位开关归零，18 使用负限位开关归零，其他设定不支持。

表 2.12 HM 模式 PDO

方向	对象	名称	数据类型	单位	备注
RxPDO 驱动接收	【Controlword: 6040h】	控制字	UINT16	---	必选
	【Homing method: 6098h】	归零模式	INT8	---	可选
	【Homing speed switch: 6099h_01】	寻找归零开关的最高速度	UINT32	0.1RPM	可选
	【Homing speed zero: 6099h_02】	离开归零开关的最高速度	UINT32	0.1RPM	可选
	【Homing acceleration: 609Ah】	归零的加减速	UINT32	RPM/s	可选
	【Homeing offset: 607Ch】	零点偏移量	INT32	微步	可选
TxPDO 驱动发送	【Statusword: 6041h】	状态字	UINT16	---	必选
	【Error code: 603Fh】	错误码	UINT16	---	可选
	【Position actual value: 6064h】	实际位置	INT32	微步	必选
	【Velocity actual value: 606C】	实际速度	INT32	0.1RPM	可选

2.6 CiA402 其他说明

2.6.1 CiA402 数字信号

【Digital inputs: 60FDh】反馈输入数字信号的状态，为 0 表示无效，为 1 表示有效。Bit 0 为 DI 1，Bit 1 为 DI 2，Bit 2 为 Latch，Bit 3~31 保留。当配置使能限位功能后，DI 2 和 DI 1 能分别触发正负限位的故障。

插板式模组应用

插板式 EtherCAT 系列从站

3. 开发环境

(1) 1 个 ZCPB-80600 耦合器模组、1 个 ZPWB-240302 电源模组、4 个 ZMD-E1-050-ECT 步进电机驱动模组。1 个 ZIOB-MB8 分线底板用于连接 6 块模组。

(2) 安装有 CODESYS 软件的 PC 一台。

(3) 本文档使用的是 CODESYS V3.5 SP20 版本进行的演示。

注意:使用 CODESYS SP18 以上版本的 CODESYS Control Win V3 环境需要安装 Npcap 驱动,Npcap 驱动软件包含在 Wireshark 设置中。若使用 CODESYS Control RTE V3 则不需要 WinPCap 或 Npcap。详细描述见 CODESYS 官方帮助:

https://content.helpme-codesys.com/zh-CHS/CODESYS%20EtherCAT/_ecat_general.html

插板式模组应用

插板式 EtherCAT 系列从站

4. 实现过程

(1) 需要先在 PC 端安装 SP18 以上版本的 CODESYS。本用例使用的版本为 SP20，如下图 4.1 所示。



图 4.1 软件版本

(2) 将 1 个 ZCPB-80600 耦合器、1 个 ZPWB-240302 电源模组、4 个 ZMDE1050ECAT 步进电机模组，从左到右分别插入 ZIOB-MB8 分线底板。分别在 EtherCAT 24V 端子和系统 24V 端子接上两路独立的 24V 电源，对于简单测试也可共用一组 24V 电源，底板的大地 EARTH 可以悬空。

用网线连接 ZCPB-80600 耦合器的“IN”口和 PC 的以太网口。硬件环境如下图 4.2 所示。

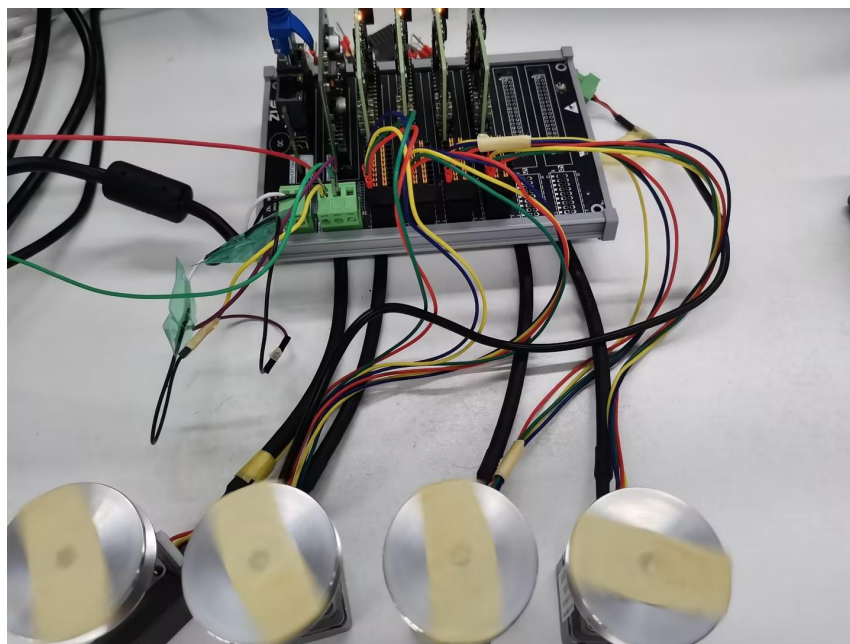


图 4.2 硬件环境

插板式模组应用

插板式 EtherCAT 系列从站

(3) 打开 CODESYS 软件，新建一个标准工程，运行环境选择 CODESYS Control Win V3 x64，PLC 语法选择结构化文本(ST)，如下图 4.3 所示。

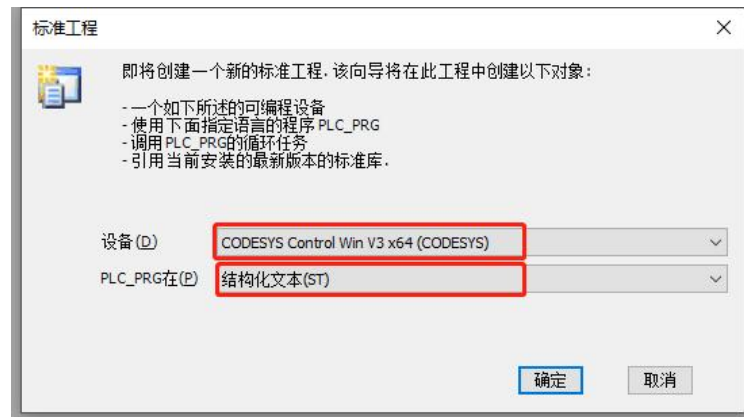


图 4.3 新建一个标准工程

(4) 点击菜单栏“工具”->“设备存储库”->“安装”添加从站描述文件 (xml)，如下图 4.4 所示。

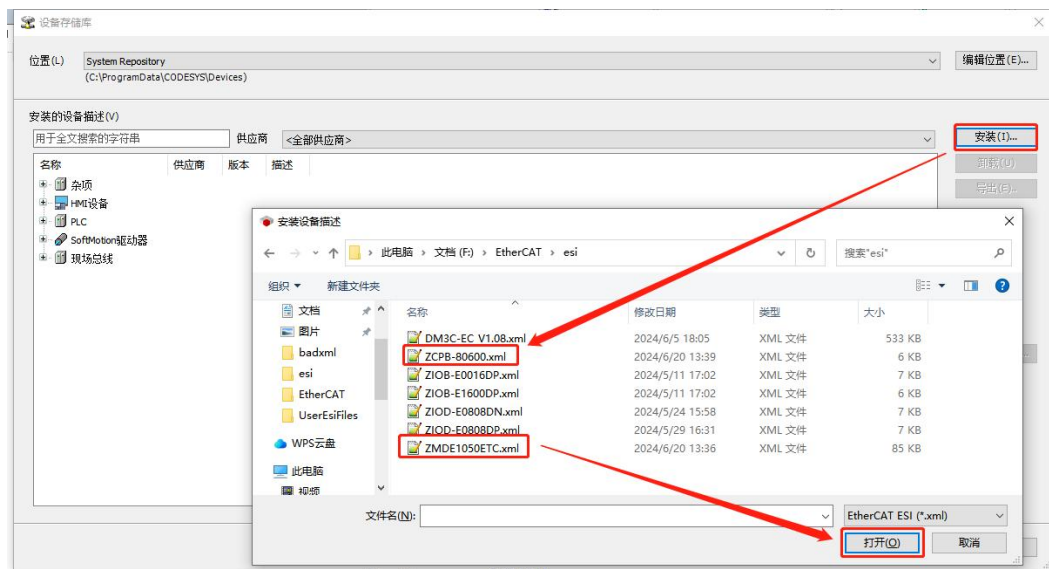


图 4.4 添加从站描述文件

(5) 右击设备栏的“Device(CODESYS Control Win V3 x64)”->“添加设备”，选择 EtherCAT->主站->EtherCAT Master 后点击添加，如下图 4.5 所示。

插板式模组应用

插板式 EtherCAT 系列从站

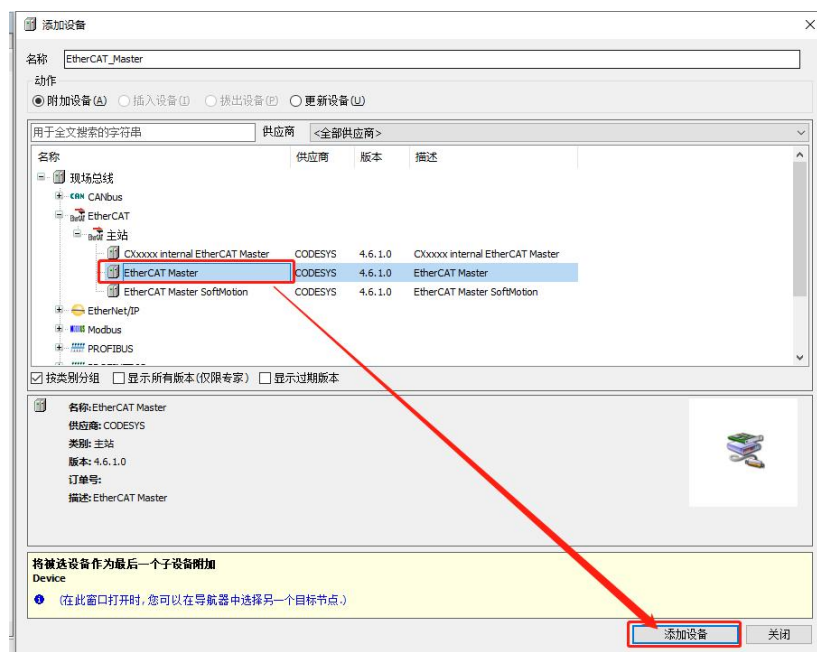


图 4.5 添加设备

(6) 右击任务栏“CODESYS Control Win Systray”图标，点击“Start PLC”启动软 PLC。然后设备栏双击“Device(CODESYS Control Win V3 x64)”，在“通信设置”选项选择设备，随后弹出“设备用户登录”界面，如图 4.6 所示。设置好用户名和密码完成网关设备激活，如下图 4.7 所示。

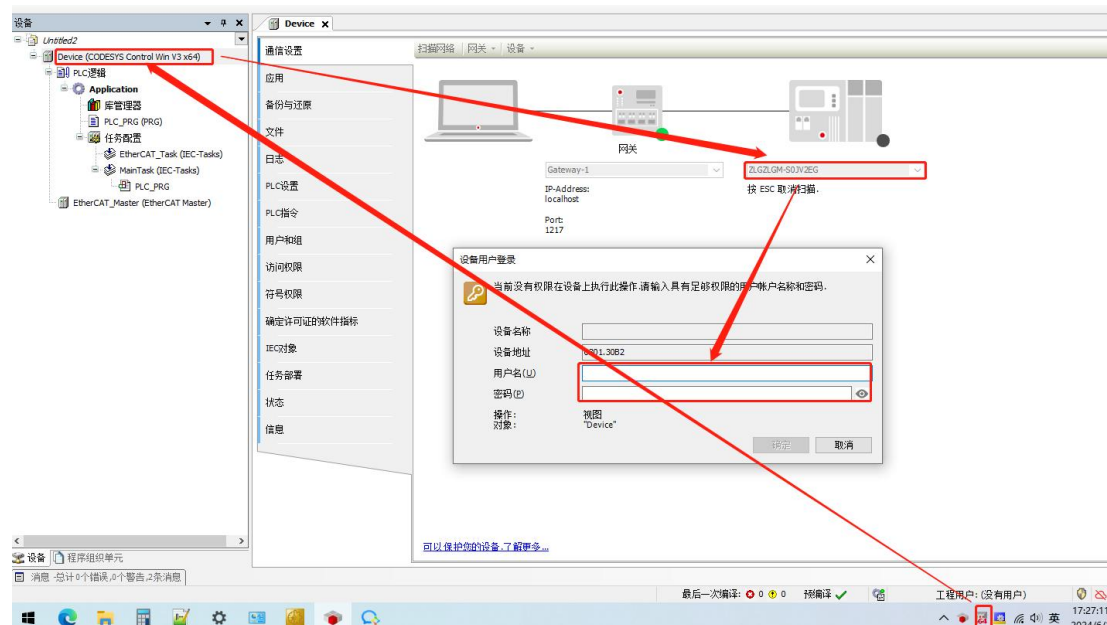


图 4.6 通讯设置

插板式模组应用

插板式 EtherCAT 系列从站

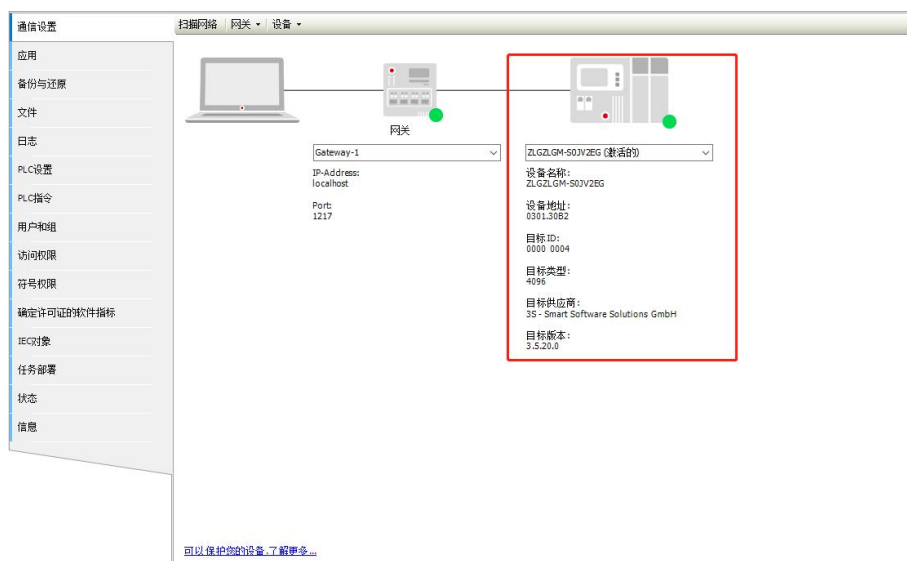


图 4.7 激活的网关

(7) 双击设备栏“EtherCAT_Master”，在“通用”选项下选择对应的网卡。在“源地址(MAC)”项，点击“选择”，选择对应的网卡，如下图 4.8 所示。

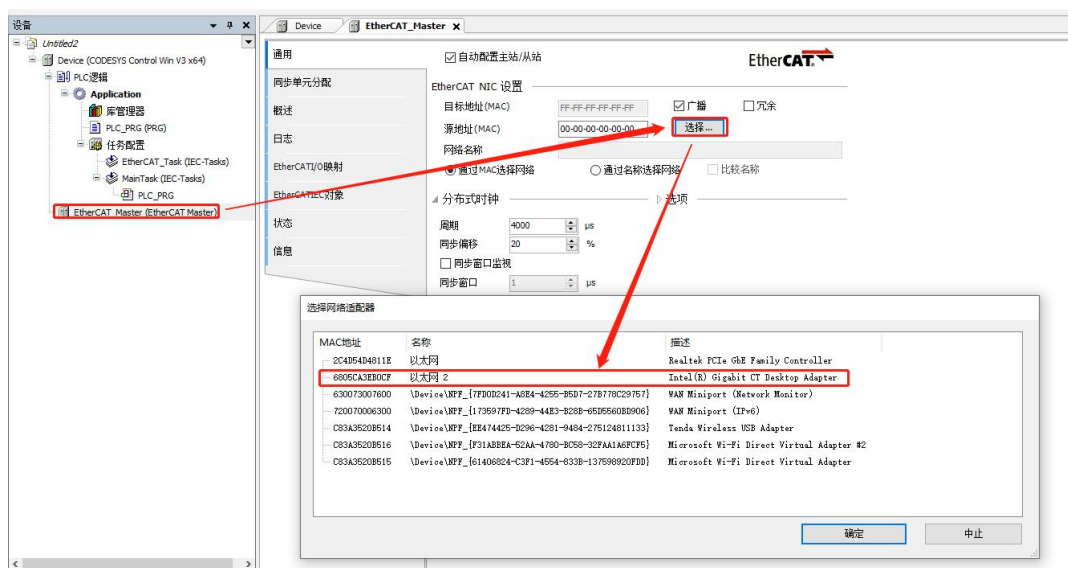


图 4.8 选择网卡

(8) 点击菜单栏上的“在线配置模式”按钮进入在线配置模式，如图 4.9 所示。再右击设备栏“EtherCAT_Master”，选择扫描设备，选择“ZPBC_80600”后点击“复制到工程”，如下图 4.10 所示。

插板式模组应用

插板式 EtherCAT 系列从站

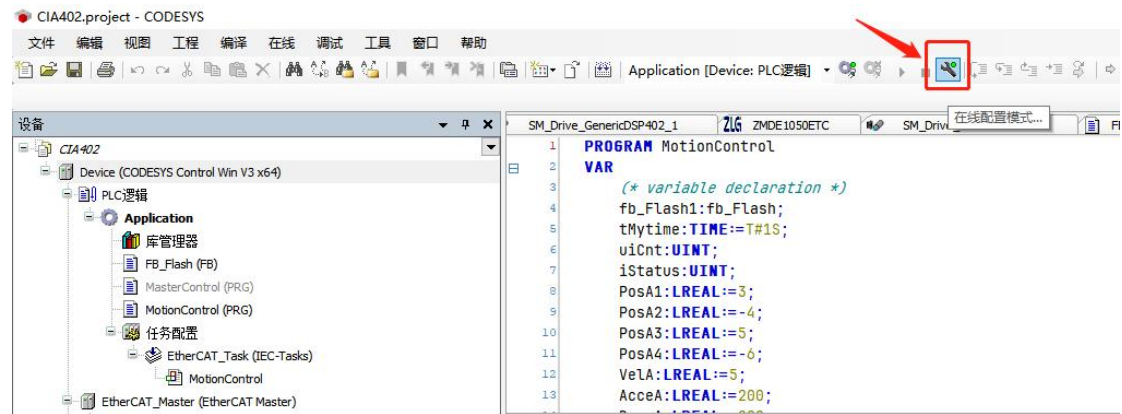


图 4.9 在线配置模式

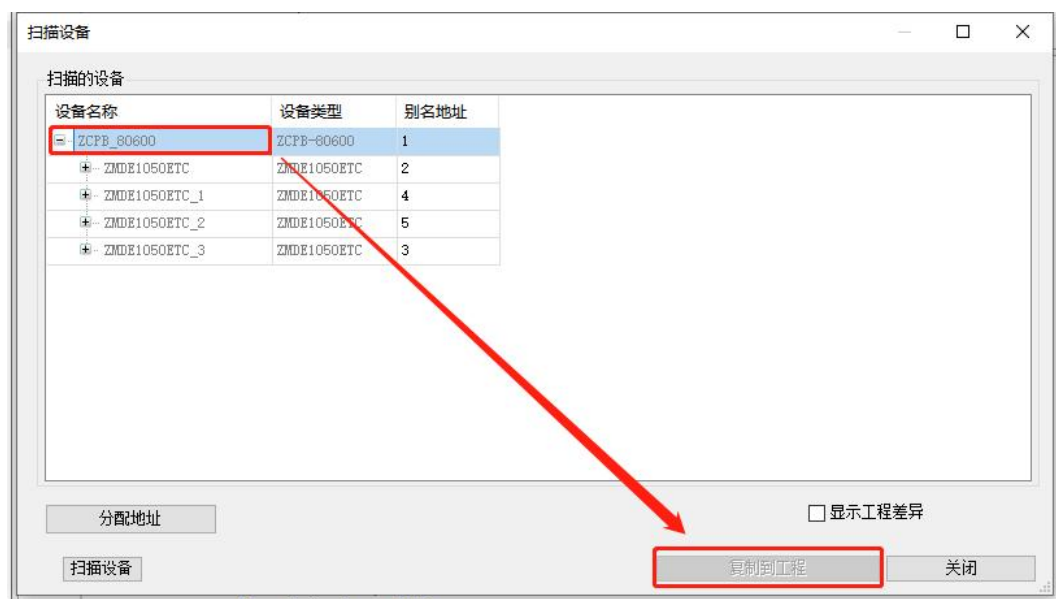


图 4.10 复制到工程

(9) 完成上一步后，可看到主站下添加了一个“ZCPB_80600”耦合器站点，该站点的“X2”端口下添加了 4 个“ZMDE1050ETC”站点。在每个“ZMDE1050ETC”站点下右击选择“添加 SoftMotion CiA402 轴”，如下图 4.11 所示。

插板式模组应用

插板式 EtherCAT 系列从站

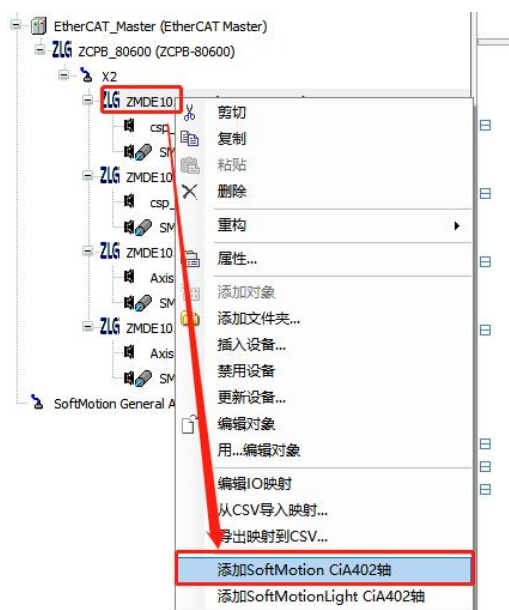


图 4.11 为站点添加 CiA402 轴

(10) 由于本测试使用的 PC 主机未授权 CODESYS 运行时环境，故 EtherCAT 总线只能运行于 FreeRun 或 SM 同步模式。分别双击打开每个“ZMDE1050ETC”站点的配置界面，在“通用”选项界面下勾选“专家设置”，然后在“分布式时钟”栏下选择“SM-Synchron”，如下图 4.12 所示。

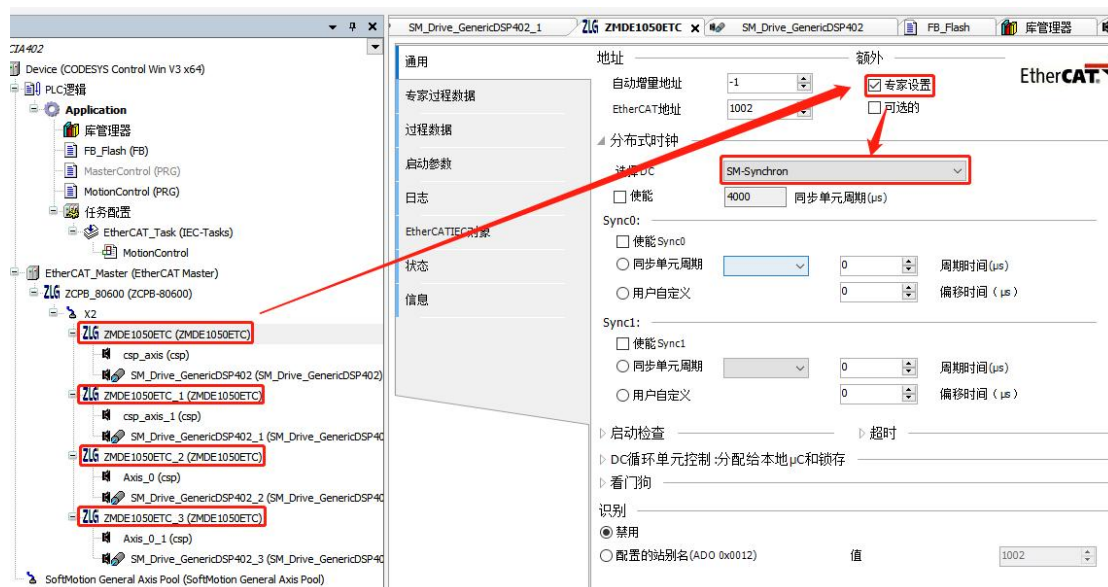


图 4.12 同步模式设置

(11) 完成站点配置与运动控制轴的添加后，开始编写 PLC 程序。找到右侧“设备”框里的“Application”，右击选择“添加对象”->“POU...”，添加功能块类型的 POU，并命名为 FB_Flash。双击打开 PLC 程序编辑界面，如图 4.13 所示。将下面的代码分别添加到变量声明区和函数调用区。

插板式模组应用

插板式 EtherCAT 系列从站

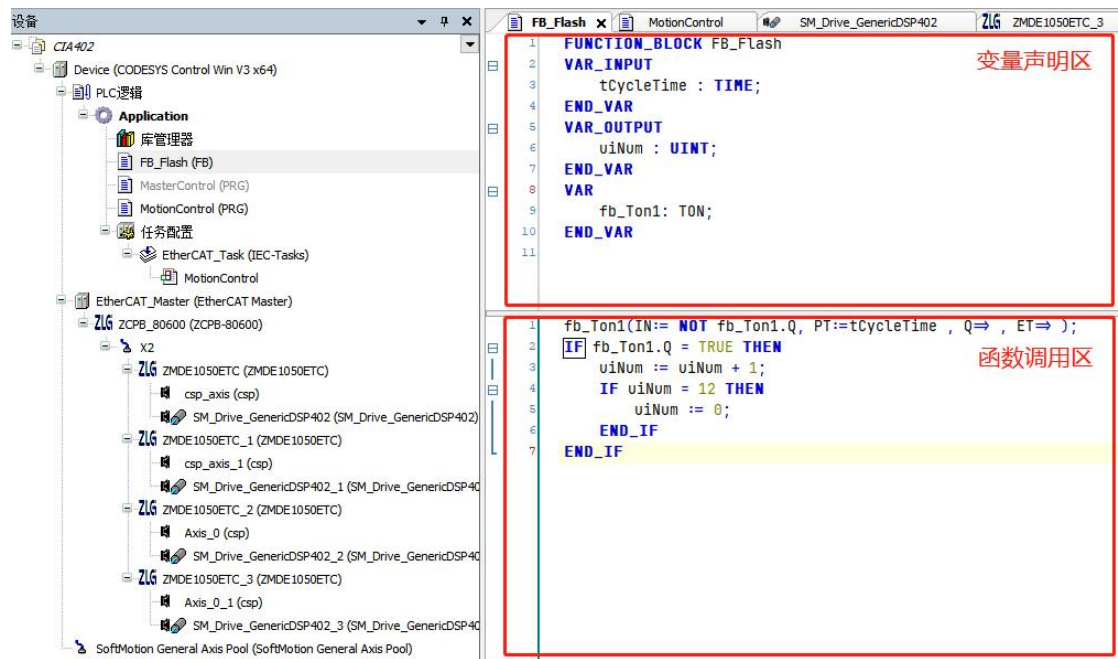


图 4.13 PLC 程序编辑界面

(* 变量声明部分代码 *)

```
FUNCTION_BLOCK FB_Flash

VAR_INPUT

    tCycleTime : TIME;

END_VAR

VAR_OUTPUT

    uiNum : UINT;

END_VAR

VAR

    fb_Ton1: TON;

END_VAR

(* 变量声明区部分代码结束 *)
```

(* 函数调用区部分代码 *)

```
fb_Ton1(IN:= NOT fb_Ton1.Q, PT:=tCycleTime , Q=> , ET=> );

IF fb_Ton1.Q = TRUE THEN

    uiNum := uiNum + 1;

    IF uiNum = 12 THEN

        uiNum := 0;

    END_IF

END_IF
```

插板式模组应用

插板式 EtherCAT 系列从站

(12) 找到右侧“设备”框里的“Application”，右击选择“添加对象”->“POU...”，添加程序类型的 POU, 并命名为 MotionControl。再用鼠标将刚才添加的 POU 拖动到下方的“任务配置”->“EtherCAT_Task(IEC-Task)”主任务下。双击打开 PLC 程序编辑界面，将下面的变量声明部分代码添加到“MotionControl”的上半窗口，即变量声明窗口，再将下面主程序部分代码添加到下半窗口，即程序调用窗口，如图 4.14 所示。

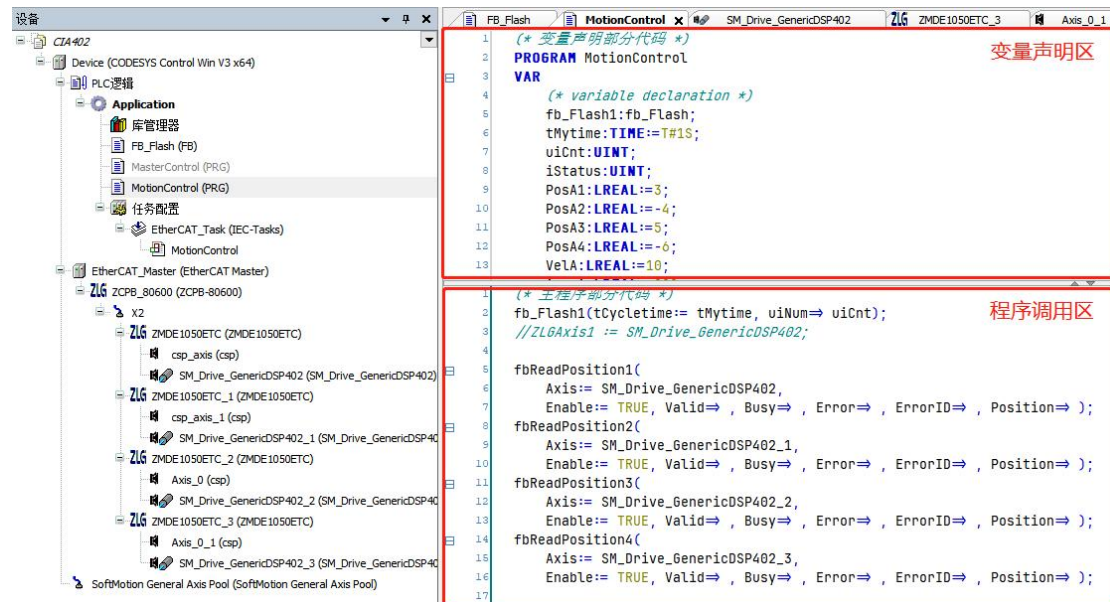


图 4.14 代码放置位置

```
(* 变量声明部分代码 *)  
PROGRAM MotionControl  
VAR  
    (* variable declaration *)  
    fb_Flash1:fb_Flash;  
    tMytime:TIME:=T#1S;  
    uiCnt:UINT;  
    iStatus:UINT;  
    PosA1:LREAL:=3;  
    PosA2:LREAL:=-4;  
    PosA3:LREAL:=5;  
    PosA4:LREAL:=-6;  
    VelA:LREAL:=10;  
    AcceA:LREAL:=200;  
    DeceA:LREAL:=200;  
    JerkA:LREAL:=500;
```

插板式模组应用

插板式 EtherCAT 系列从站

```
PosB1:LREAL:=-3;
PosB2:LREAL:=4;
PosB3:LREAL:=-5;
PosB4:LREAL:=6;
VelB:LREAL:=10;
AcceB:LREAL:=200;
DeceB:LREAL:=200;
JerkB:LREAL:=500;
bPower:BOOL;
bStop: BOOL;
bReset:BOOL;
bMoveAbsA:BOOL;
bMoveAbsB:BOOL;
//ZLGAxis1:AXIS_REF_SM3;

(* function block declaration *)
fbPower1:MC_Power;
fbPower2:MC_Power;
fbPower3:MC_Power;
fbPower4:MC_Power;
fbStop1:MC_Stop;
fbStop2:MC_Stop;
fbStop3:MC_Stop;
fbStop4:MC_Stop;
fbReset1:MC_Reset;
fbReset2:MC_Reset;
fbReset3:MC_Reset;
fbReset4:MC_Reset;
fbSetpos1:MC_SetPosition;
fbMoveAbsA1:MC_MoveAbsolute;
fbMoveAbsA2:MC_MoveAbsolute;
fbMoveAbsA3:MC_MoveAbsolute;
fbMoveAbsA4:MC_MoveAbsolute;
fbMoveAbsB1:MC_MoveAbsolute;
fbMoveAbsB2:MC_MoveAbsolute;
```

插板式模组应用

插板式 EtherCAT 系列从站

```
fbMoveAbsB3:MC_MoveAbsolute;

fbMoveAbsB4:MC_MoveAbsolute;

fbReadPosition1 : MC_ReadActualPosition;

fbReadPosition2 : MC_ReadActualPosition;

fbReadPosition3 : MC_ReadActualPosition;

fbReadPosition4 : MC_ReadActualPosition;

END_VAR

(* 变量声明部分代码结束 *)
```

```
(* 主程序部分代码 *)

fb_Flash1(tCycletime:= tMytime, uiNum=> uiCnt);

fbReadPosition1(
    Axis:= SM_Drive_GenericDSP402,
    Enable:= TRUE, Valid=> , Busy=> , Error=> , ErrorID=> , Position=> );
fbReadPosition2(
    Axis:= SM_Drive_GenericDSP402_1,
    Enable:= TRUE, Valid=> , Busy=> , Error=> , ErrorID=> , Position=> );
fbReadPosition3(
    Axis:= SM_Drive_GenericDSP402_2,
    Enable:= TRUE, Valid=> , Busy=> , Error=> , ErrorID=> , Position=> );
fbReadPosition4(
    Axis:= SM_Drive_GenericDSP402_3,
    Enable:= TRUE, Valid=> , Busy=> , Error=> , ErrorID=> , Position=> );

CASE iStatus OF
    STATE_POWER: //enable all axis
        fbPower1(
            Axis:= SM_Drive_GenericDSP402,
            Enable:= TRUE,
            bRegulatorOn:= TRUE,
            bDriveStart:= TRUE,
            Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> ,
            ErrorID=> );
```

插板式模组应用

插板式 EtherCAT 系列从站

```
fbPower2(  
    Axis:= SM_Drive_GenericDSP402_1,  
    Enable:= TRUE,  
    bRegulatorOn:= TRUE,  
    bDriveStart:= TRUE,  
    Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> ,  
    ErrorID=> );  
  
fbPower3(  
    Axis:= SM_Drive_GenericDSP402_2,  
    Enable:= TRUE,  
    bRegulatorOn:= TRUE,  
    bDriveStart:= TRUE,  
    Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> ,  
    ErrorID=> );  
  
fbPower4(  
    Axis:= SM_Drive_GenericDSP402_3,  
    Enable:= TRUE,  
    bRegulatorOn:= TRUE,  
    bDriveStart:= TRUE,  
    Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> ,  
    ErrorID=> );  
  
IF fbPower1.Status AND fbPower2.Status AND fbPower3.Status AND fbPower4.Status AND  
uiCnt>3 THEN  
    iStatus := STATE_MOVE_POSABS1;  
END_IF  
  
STATE_MOVE_POSABS1: //move to positon A  
    bMoveAbsA := TRUE;  
    bMoveAbsB := FALSE;  
    IF fbReadPosition1.Valid=TRUE AND fbReadPosition1.Position=PosA1 AND  
    fbReadPosition2.Valid=TRUE AND fbReadPosition2.Position=PosA2 AND  
    fbReadPosition3.Valid=TRUE AND fbReadPosition3.Position=PosA3 AND  
    fbReadPosition4.Valid=TRUE AND fbReadPosition4.Position=PosA4 THEN  
        iStatus := STATE_MOVE_POSABS2;  
    END_IF  
  
    IF fbMoveAbsA1.Error OR fbMoveAbsA2.Error OR fbMoveAbsA3.Error OR  
    fbMoveAbsA4.Error THEN  
        iStatus := STATE_STOP;
```

插板式模组应用

插板式 EtherCAT 系列从站

```
END_IF

STATE_MOVE_POSABS2:
    bMoveAbsA := FALSE;
    bMoveAbsB := TRUE;
    IF fbReadPosition1.Valid=TRUE AND fbReadPosition1.Position=PosB1 AND
    fbReadPosition2.Valid=TRUE AND fbReadPosition2.Position=PosB2 AND
    fbReadPosition3.Valid=TRUE AND fbReadPosition3.Position=PosB3 AND
    fbReadPosition4.Valid=TRUE AND fbReadPosition4.Position=PosB4 THEN
        iStatus := STATE_MOVE_POSABS1;
    END_IF
    IF fbMoveAbsB1.Error OR fbMoveAbsB2.Error OR fbMoveAbsB3.Error OR
fbMoveAbsB4.Error THEN
        iStatus := STATE_STOP;
    END_IF

STATE_STOP:
    (* when error happen, all axis will stop.
    Then all axis should reset manually(set the iStatus to STATE_RESET). *)
    bMoveAbsA := FALSE;
    bMoveAbsB := FALSE;
    bStop := TRUE;

STATE_RESET: //reset all axis
    bMoveAbsA := FALSE;
    bMoveAbsB := FALSE;
    bReset := TRUE;
    IF fbReset1.Done AND fbReset2.Done AND fbReset3.Done AND fbReset4.Done THEN
        iStatus := 0;
    END_IF
END_CASE

(* move to position A *)
fbMoveAbsA1(
    Axis:= SM_Drive_GenericDSP402,
    Execute:= bMoveAbsA,
```

插板式模组应用

插板式 EtherCAT 系列从站

```
Position:= PosA1,
    Velocity:= VelA,
    Acceleration:= AcceA,
    Deceleration:= DeceA,
    Jerk:= JerkA,
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,
    ErrorID=> );
fbMoveAbsA2(
    Axis:= SM_Drive_GenericDSP402_1,
    Execute:= bMoveAbsA,
    Position:= PosA2,
    Velocity:= VelA,
    Acceleration:= AcceA,
    Deceleration:= DeceA,
    Jerk:= JerkA,
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,
    ErrorID=> );
fbMoveAbsA3(
    Axis:= SM_Drive_GenericDSP402_2,
    Execute:= bMoveAbsA,
    Position:= PosA3,
    Velocity:= VelA,
    Acceleration:= AcceA,
    Deceleration:= DeceA,
    Jerk:= JerkA,
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,
    ErrorID=> );
fbMoveAbsA4(
    Axis:= SM_Drive_GenericDSP402_3,
    Execute:= bMoveAbsA,
    Position:= PosA4,
    Velocity:= VelA,
    Acceleration:= AcceA,
    Deceleration:= DeceA,
    Jerk:= JerkA,
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,
    ErrorID=> );
```

插板式模组应用

插板式 EtherCAT 系列从站

```
(* move to position B *)  
  
fbMoveAbsB1(  
    Axis:= SM_Drive_GenericDSP402,  
    Execute:= bMoveAbsB,  
    Position:= PosB1,  
    Velocity:= VelB,  
    Acceleration:= AcceB,  
    Deceleration:= DeceB,  
    Jerk:= JerkB,  
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,  
    ErrorID=> );  
  
fbMoveAbsB2(  
    Axis:= SM_Drive_GenericDSP402_1,  
    Execute:= bMoveAbsB,  
    Position:= PosB2,  
    Velocity:= VelB,  
    Acceleration:= AcceB,  
    Deceleration:= DeceB,  
    Jerk:= JerkB,  
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,  
    ErrorID=> );  
  
fbMoveAbsB3(  
    Axis:= SM_Drive_GenericDSP402_2,  
    Execute:= bMoveAbsB,  
    Position:= PosB3,  
    Velocity:= VelB,  
    Acceleration:= AcceB,  
    Deceleration:= DeceB,  
    Jerk:= JerkB,  
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,  
    ErrorID=> );  
  
fbMoveAbsB4(  
    Axis:= SM_Drive_GenericDSP402_3,  
    Execute:= bMoveAbsB,  
    Position:= PosB4,  
    Velocity:= VelB,  
    Acceleration:= AcceB,
```

插板式模组应用

插板式 EtherCAT 系列从站

```
Deceleration:= DeceB,  
    Jerk:= JerkB,  
    Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> ,  
    ErrorID=> );  
  
(* stop the axis *)  
fbStop1(  
    Axis:= SM_Drive_GenericDSP402,  
    Execute:= bStop,  
    Deceleration:= 100,  
    Jerk:= 500,  
    Done=> , Busy=> , Error=> , ErrorID=> );  
fbStop2(  
    Axis:= SM_Drive_GenericDSP402_1,  
    Execute:= bStop,  
    Deceleration:= 100,  
    Jerk:= 500,  
    Done=> , Busy=> , Error=> , ErrorID=> );  
fbStop3(  
    Axis:= SM_Drive_GenericDSP402_2,  
    Execute:= bStop,  
    Deceleration:= 100,  
    Jerk:= 500,  
    Done=> , Busy=> , Error=> , ErrorID=> );  
fbStop4(  
    Axis:= SM_Drive_GenericDSP402_3,  
    Execute:= bStop,  
    Deceleration:= 100,  
    Jerk:= 500,  
    Done=> , Busy=> , Error=> , ErrorID=> );  
  
(* reset the asix *)  
fbReset1(  
    Axis:= SM_Drive_GenericDSP402,  
    Execute:= bReset,  
    Done=> , Busy=> , Error=> , ErrorID=> );
```

插板式模组应用

插板式 EtherCAT 系列从站

```
fbReset2(
    Axis:= SM_Drive_GenericDSP402_1,
    Execute:= bReset,
    Done=> , Busy=> , Error=> , ErrorID=> );

fbReset3(
    Axis:= SM_Drive_GenericDSP402_2,
    Execute:= bReset,
    Done=> , Busy=> , Error=> , ErrorID=> );

fbReset4(
    Axis:= SM_Drive_GenericDSP402_3,
    Execute:= bReset,
    Done=> , Busy=> , Error=> , ErrorID=> );

(* 主程序代码部分结束 *)
```

(13) PLC 程序编写完成后，运行程序。点击菜单栏里的“登录”->“运行”按钮，如图 4.15 所示。若程序正确运行，则四个电机会进行往返运动。现象为四个电机同时启动，依次停转，又立马同时起转，依次停转，如此无限循环。

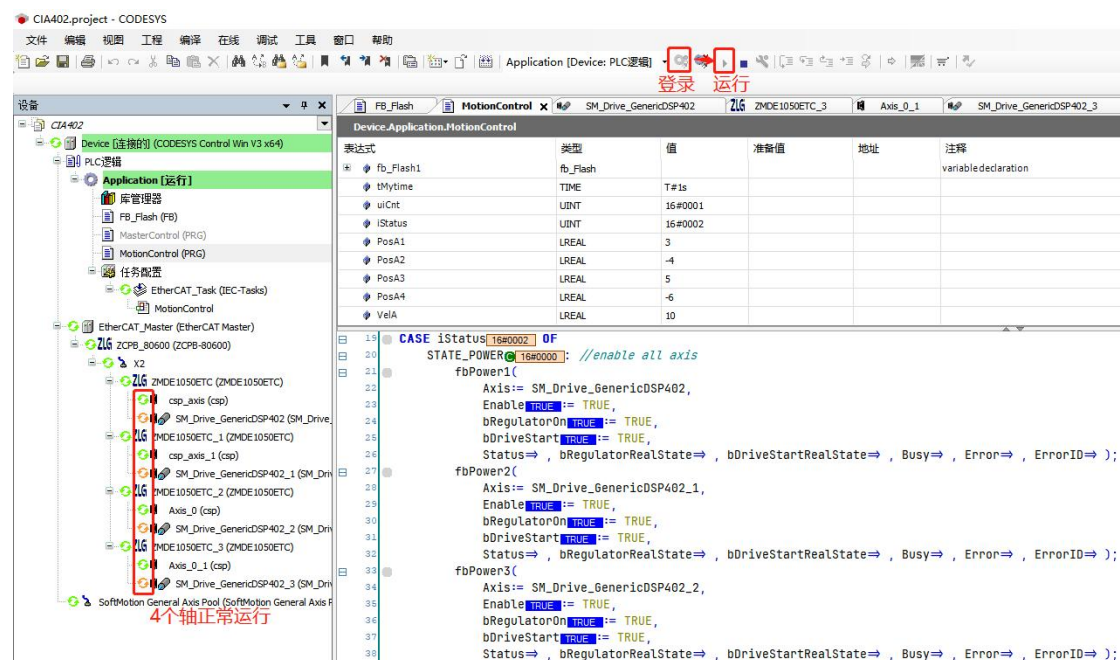


图 4.15 运行程序

插板式模组应用

插板式 EtherCAT 系列从站

5. 免责声明

本着为用户提供更好服务的原则，广州致远电子股份有限公司（下称“致远电子”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，致远电子不能完全保证该文档在任何时段的时效性与适用性。致远电子有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问致远电子官方网站或者与致远电子工作人员联系。感谢您的包容与支持！

诚信共赢，持续学习，客户为先，专业专注，只做第一

广州致远电子股份有限公司

更多详情请访问
www.zlg.cn

欢迎拨打全国服务热线
400-888-4005

