# i5系列 用户手册 V2.0



成都乐创自动化技术股份有限公司

修订记录

日期	修订版本	描述	
2018-08-28	V1.0	第一个发布版本。	
2019-01-07	V1.1	增加了 7.6, 7.7,7.8,7.9 四个小节,修改了 8.2 节中 对 SYSTEM 任务的描述;	
		根据 Assistant 合并到 MultiProg 后对相应的内容进 行了修改,适用于 MultiProg V5.51_1707;	
		修改了14.2中的异常代码描述;	
		根据安装程序的修改,修订了第5章;	
		对 M3 共享变量的读写属性进行了调整,修订了 14.1, 增加了 14.3。	
2019-6-19	V1.2	第6章,第7章部分截图及描述;	
		增加了 MC 部分的错误代码;	
		7.3.2节表格中增加了新三个功能块;	
		6.9.4.4 截图和描述, 6.9.48 节, 7.2.1.6 节描述;	
		第6章,第7章,资源类型描述,截图替换;	
		附表 3:地址错误更改;	
		删除 5.2.2 WinpCap 安装;	
		增加 9.2.6 逻辑分析器;第 6、7 章部分截图,描述更改;删除断点功能。	
2019-09-06	V1.3	全面错误修订。	
2019-12-19	V1.4	补充了 15.2 错误代码;	
		修改了 13.1 增加了系统加载阶段错误报警灯的详细说明;	
		修正了 4.6 操作步骤对应的章节号对应错误的问题;	
2020-03-21	V1.6	章节 6.8.4.8 增加螺距补偿,反向间隙补偿功能说明;	

2020-09-15	V2.0	1,	修改连接到控制器的方法描述,由手动配置通信 IP 地址再连接更改为通过扫描控制器自动连接
		2,	修改编辑 PLC 工程的步骤,简化使用轴变量的方式
		3,	修改手动配置控制器类型及版本的说明
		4,	更新任务优先级相关的描述

# 目录

1.	前言		7
	1.1.	阅读对象	7
	1.2.	适用产品	7
2.	注意事工	页	7
	2.1.	图形符号和含义	7
	2.2.	敬 <u>牛</u> 言口	7
	2.3.	注意	8
3.	术语		8
	3.1.	MULTIPROG 相关	8
	3.1.1.	MULTIPROG	8
	3.1.2.	MULTIPROG 工程	8
	3.1.3.	eCLR	8
	3.1.4.	POU	8
	3.1.5.	全局变量	8
	3.1.6.	局部变量	9
	3.2.	EtherCAT 相关	9
	3.2.1.	EtherCAT	9
	3.2.2.	PDO	9
	3.2.3.	COE 参数	9
	3.2.4.	ESI 文件	9
4.	系统概要	म् स्र	9
	4.1.	硬件组成	9
	4.2.	软件组成1	0
	4.3.	软件特点1	0
	4.4.	主要规格1	1
	4.5.	查看版本1	2
	4.5.1.	MULTIPROG 版本1	2
	4.5.2.	固件版本信息1	2

	4.6.	使用步骤	. 13
	4.6.1.	步骤的概要	. 13
	4.6.2.	详细步骤	. 14
	4.7.	快速使用示例	. 17
	4.7.1.	调试环境准备	. 17
	4.7.2.	创建 PLC 工程	. 17
	4.7.3.	连接到控制器	. 19
	4.7.4.	EtherCat 从站配置	. 19
	4.7.5.	配置轴参数	. 21
	4.7.6.	使用功能块编写控制逻辑	. 21
	4.7.7.	调试试运行	. 25
5.	软件安徽	装	. 26
	5.1.	PC 软件的安装	. 26
	5.1.1.	安装须知	. 26
	5.1.2.	安装需求	. 26
	г <b>э</b>	软件完装	. 26
	5.2.	111 又衣	-
	5.2.	WIT 文役 MultiProg Express 安装	. 26
	5.2. 5.2.1. 5.3.	WIT 实夜 MultiProg Express 安装 PC 软件的卸载	. 26 . 28
	5.2. 5.2.1. 5.3. 5.3.1.	MultiProg Express 安装 PC 软件的卸载 卸载 MultiProgExpress 打开控制面板	. 26 . 28 . 28
6.	5.2. 5.2.1. 5.3. 5.3.1. 使用 MI	WIT 文化 MultiProg Express 安装 PC 软件的卸载 卸载 MultiProgExpress 打开控制面板 JLTIPROG 创建 PLC 工程	. 26 . 28 . 28 . 30
6.	5.2. 5.3. 5.3.1. 使用 MI 6.1.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> </ul>	26 28 28 30 30
6.	5.2. 5.2.1. 5.3. 5.3.1. 使用 MI 6.1. 6.1.1.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置(Configuration)</li> </ul>	. 26 . 28 . 28 . 30 . 30 . 31
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li></ul>	26 28 30 30 31 32
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置 (Configuration)</li></ul>	26 28 28 30 30 31 31 32
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置 (Configuration)</li></ul>	26 28 30 30 31 32 35 36
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置 (Configuration)</li> <li>资源 (Resource)</li> <li>控制器</li> <li>设置控制器类型及版本</li> <li>任务 (Task)</li> </ul>	. 26 . 28 . 28 . 30 . 30 . 31 . 31 . 32 . 35 . 36 . 37
6.	5.2. 5.2.1. 5.3. 使用 M 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5. 6.1.6.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置 (Configuration)</li> <li>资源 (Resource)</li> <li>控制器</li> <li>设置控制器类型及版本</li> <li>在务 (Task)</li></ul>	. 26 . 28 . 30 . 30 . 31 . 32 . 35 . 36 . 37 . 42
6.	5.2. 5.2.1. 5.3. 使用 M 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5. 6.1.6. 6.1.7.	<ul> <li>MultiProg Express 安装</li> <li>PC 软件的卸载</li> <li>卸载 MultiProgExpress 打开控制面板</li> <li>JLTIPROG 创建 PLC 工程</li> <li>MULTIPROG 的工程组织基本概念</li> <li>配置 (Configuration)</li> <li>资源 (Resource)</li> <li>控制器</li> <li>设置控制器类型及版本</li></ul>	26 28 30 30 31 32 35 36 37 42 48
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5. 6.1.6. 6.1.7.	WILLIPROG burgess 安装	. 26 . 28 . 30 . 30 . 31 . 32 . 35 . 36 . 37 . 42 . 48 . 50
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5. 6.1.6. 6.1.7. 6.2. 6.2.1.	WILLIPROG Express 安装	. 26 . 28 . 30 . 30 . 31 . 32 . 35 . 36 . 37 . 42 . 48 . 50 . 50
6.	5.2. 5.2.1. 5.3. 使用 MI 6.1. 6.1.1. 6.1.2. 6.1.3. 6.1.4. 6.1.5. 6.1.6. 6.1.7. 6.2. 6.2.1. 6.2.2.	WILLIPROG Express 安装	. 26 . 28 . 30 . 30 . 31 . 32 . 35 . 36 . 37 . 42 . 48 . 50 . 50 . 51

	6.2.4.	消息窗口	. 53
	6.2.5.	编辑向导窗口	. 54
	6.2.6.	子窗口的模式	. 54
	6.2.7.	子窗口的大小调整	. 55
6.	3.	新建工程	. 56
	6.3.1.	.使用工程向导创建和配置新工程	. 56
	6.3.2.	使用工程模版创建新工程	. 63
6.	4.	添加从站	. 64
	6.4.1.	导入 ESI 文件	. 64
	6.4.2.	配置 MULTIPROG 与 i5 的连接	. 66
	6.4.3.	自动扫描从站	. 69
	6.4.4.	手动添加从站	. 70
6.	5.	EtherCAT 相关参数配置	. 72
	6.5.1.	从站公共参数配置	. 73
	6.5.2.	伺服从站参数配置	. 74
6.	6.	PDO 与 IO 变量的映射	. 80
6.	7.	变更从站	. 82
	6.7.1.	重新自动扫描从站	. 82
	6.7.2.	手动修改从站配置	. 84
6.	8.	轴参数配置	. 85
	6.8.1.	轴参数概要	. 86
	6.8.2.	轴的设定步骤	. 88
	6.8.3.	轴变量	. 96
	6.8.4.	用户程序中轴的指定方法	. 96
6.	9.	轴组参数配置	. 97
	6.9.1.	轴组概要	. 97
	6.9.2.	轴组参数概要	. 97
	6.9.3.	轴组的设定步骤	. 98
	6.9.4.	轴组变量	100
	6.9.5.	用户程序中轴组的指定方法	101
7.	控制程序	序的编写	101
7.	1.	支持的编程语言	101

	7.1.1.	编程语言列表	101
	7.1.2.	指令表(IL)编程语言	102
	7.1.3.	梯形图(LD)编程语言	103
	7.1.4.	功能块图(FBD)编程语言	106
	7.1.5.	结构化文本( <b>ST</b> )编程语言	109
	7.1.6.	顺序功能图( <b>SFC</b> )编程语言	110
	7.2.	运动控制功能	114
	7.2.1.	单轴位置控制	114
	7.2.2.	单轴速度控制	118
	7.2.3.	单轴控制的通用功能	119
	7.2.4.	轴组运动控制	125
	7.2.5.	多轴协调控制的通用功能	129
	7.2.6.	原点复位	130
	7.2.7.	其他功能	138
	7.3.	运动控制功能块库的使用	140
	7.3.1.	运动控制功能块库的添加	140
	7.3.2.	运动控制功能块列表	143
	7.3.3.	PLC 程序添加功能块	145
	7.3.4.	功能块实例化	146
	7.4.	MC 单轴运动实例	148
	7.4.1.	PLC 工程建立	148
	7.4.2.	单轴 DEMO 控制逻辑及操作方法	151
	7.5.	MC 轴组运动实例	154
	7.5.1.	PLC 工程建立	154
	7.5.2.	轴组 DEMO 控制逻辑及操作方法	156
	7.6.	POU 单元调用方法举例	160
	7.7.	多任务、多程序组合应用举例	162
	7.8.	封装及调用用户库举例	164
	7.9.	工程输出应用举例	171
8.	任务设计	+	175
	8.1.	任务设计步骤概要	175
	8.2.	任务类型	175

8.3.	任务优先级	177
8.4.	将程序分配到任务	177
8.4.1.	增加删除任务	177
8.4.2.	向任务分配程序	177
9. 调试与	运行	179
9.1.	步骤概要	179
9.2.	连接控制器	179
9.3.	联机调试	181
9.3.1.	调试模式	181
9.3.2.	强制	181
9.3.3.	覆盖	182
9.3.4.	交叉参考窗口	182
9.3.5.	变量监视窗口	183
9.3.6.	逻辑分析器	183
9.4.	正式运行	185
9.4.1.	下装引导工程实现上电启动运行的 PLC 程序	185
9.4.2.	删除上电启动运行的 PLC 程序	186
10.	连接人机界面	187
10.1.	与人机界面的连接步骤概要	187
10.2.	连接配置	187
10.2.1.	Modbus RTU 串口通信	187
10.2.2.	Modbus TCP 网口通信	187
10.3.	MODBUS 地址	187
11.	与 PC 组合控制	188
11.1.	与 PC 组合控制步骤概要	188
11.2.	连接配置	189
11.3.	函数库的 API 定义	189
11.3.1.	函数库的 API 接口描述	190
12.	系统维护	197
12.1.	固件升级	197
12.2.	设置 i5 的系统时间	197
12.3.	设置 i5 的 IP 地址	198

12.4.	系统日志1		
13.	故障的确认和处理	199	
13.1.	启动过程故障		
13.1.1.	动作状态显示 LED	199	
13.2.	调试过程故障	202	
13.2.1.	MULTIPROG 无法与 i5 连接显示超时	202	
13.2.2.	通过 MULTIPROG 查看错误信息	203	
13.2.3.	查看错误信息	203	
13.3.	运行过程故障	203	
14.	安全设置	203	
14.1.	安全设置概述	203	
14.2.	加密设置操作步骤	203	
15.	附录	204	
15.1.	附录 1: MODBUS 地址表	204	
15.2.	附录 2: 异常代码一览表	206	
15.2.1.	系统管理相关	206	
15.2.2.	参数操作相关	207	
15.2.3.	MODBUS 相关	209	
15.2.4.	参数文件解析相关	210	
15.2.5.	PLC 内核相关	210	
15.2.6.	运动控制相关	212	
15.3.	附录 3: M3 变量地址表	222	

## 1. 前言

## 1.1. 阅读对象

本手册提供给以下阅读对象:

使用 i5 系列控制器进行控制系统编程开发的研发技术人员。

使用人员应当具备以下基础:

具有电工专业知识的人员(合格的电气工程师或者具有同等知识的人员),能理解并按 手册描述安装和连接设备;

编程语言的阅读对象为理解国际标准 IEC61131-3 的内容的人员,能理解并按手册描述使用基于 IEC61131-3 标准规定的编程语言完成控制程序编写和调试;

## 1.2. 适用产品

本手册适用于 i5 系列控制器(以下简称 i5)。

## 2. 注意事项

## 2.1. 图形符号和含义

本手册中以下列标示和图形符号表示安全使用 i5 的注意事项。这里表示的注意事项记载了与安全相关的重要内容。请务必遵守。

标示和含义如下所示。

标示	名称	含义
<b>A</b>	注意	表示"一般注意事项",若不按此进行操
<u> </u>		作,可能导致部分功能不能正常使用。
	敬生 言 口	表示 " 严重事项 ",若不按此进行操作,
		可能导致系统严重错误,极端情况下可能引
		起控制系统和设备的损坏。

## 2.2. 警告

本手册各章节有以下格式的警告信息,请务必遵守。

	此处描述警告信息内容
警告	

## 2.3. 注意

本手册各章节有以下格式的注意信息,请使用相关功能时仔细阅读并按照注意事项进行 操作。



## 3. 术语

## 3.1. MULTIPROG 相关

## 3.1.1. MULTIPROG

MULTIPROG 是菲尼克斯软件公司针对中大型控制应用场合开发的通用 PLC 编程系统, 被广泛应用于机械制造、汽车和过程自动化行业。

该工具适用于 Windows 7、8、10 视窗操作系统。其工程结构完全符合 IEC61131-3 标准,支持标准定义的 5 种编程语言,且允许用户自定义库和数据结构。是一个编写调试控制程序的集成开发环境。

## 3.1.2. MULTIPROG 工程

MULTIPROG 工程是在 MULTIPROG 中基于 IEC61131-3 标准的编程语言编写的 PLC 和 MC (运动控制)控制程序和配置参数的组织单元。每个工程能单独保存,编译制作,下装并独 立运行。

## 3.1.3. eCLR

运行于 i5 的 PLC 内核,负责对 PLC 程序进行加载,控制其运行,实现 PLC 任务的调度, 并实现与 MULTIPROG 的通信。

#### 3.1.4. POU

根据 IEC 61131-3 标准的定义,是程序组织单元,英文缩写为 POU,是 PLC 程序的语言 元素。它们是包含了程序代码的小的、独立的软件单位。POU 的名称在工程内必须是唯一的。

每种 POU 都由两个不同部分组成:变量声明部分和代码本体部分。两者都被命名为"工作单"。在声明部分,声明了所有局部变量。一个 POU 的指令或代码部分包含指令,该指令是用所希望的编程语言编写的。

## 3.1.5. 全局变量

一个工程中所有 POU 都可访问的变量。i5 系统的全局变量包括以下几种主要类型:

- i5的输入变量,简称为I变量
- i5的输出变量,简称为Q变量
- PC通过MODBUS可访问的共享变量,简称为共享M变量
- PLC内核的系统变量,简称为PLC系统M变量
- i5的系统变量,简称为i5系统M变量

## 3.1.6. 局部变量

仅在一个 POU 内部的程序代码可访问的变量。

## 3.2. EtherCAT 相关

## 3.2.1. EtherCAT

一种由德国倍福开发的实时以太网工业现场总线,符合 EtherCAT 标准的主站和从站可 以通过网线连接构成可扩展的控制系统。

#### 3.2.2. PDO

Process Data Object 的缩写,即过程数据对象,EtherCAT 每个通讯周期会将其在主站和 从站之间进行同步,用于实现主站对从站的周期性控制。

其中 RPDO 为主站到从站的命令参数, TPDO 为从站返回主站的状态参数。

## 3.2.3. COE 参数

CANopen Over EtherCAT 的缩写,由 CANopen 标准定义的一组操作从站参数的数据对象。 主站通过 COE 参数读写实现对从站的控制功能。

对轴控制的 COE 参数,遵循 Cia 402 标准。

对 IO 控制的 COE 参数定义,遵循 Cia 401 标准。

## 3.2.4. ESI 文件

EtherCAT Slave Information 的缩写,一种由 EtherCAT 标准定义的,按 XML 格式描述 EtherCAT 从站配置信息的文件。主站根据其描述内容识别从站并对从站进行配置。

## 4. 系统概要

4.1. 硬件组成

要按照本手册进行控制系统的开发和调试,需要以下部分组成一个最小系统。

● 一台i5控制器

● EtherCAT从站若干,包括EtherCAT接口伺服,或者EtherCAT接口的轴模块,或者 EtherCAT接口的IO从站模块

- 一台安装MULTIPROG软件的PC
- 连接PC和i5的网线,以及连接i5和从站的网线

(可选)支持MOBUS TCP或者MODBUS RTU协议和接口的HMI,HMI的编程下载线,以 及与i5的连接线。

以上部分通过以下示意图连接形成一个开发环境。



图 4.1

4.2. 软件组成

序号	名称	描述	运行环境
1	MULTIPROG	使用基于 IEC61131-3 标准规定的编 程语言进行控制程序编写,编译, 调试,下装,以及对 EtherCAT 从站 参数进行配置,轴和轴组参数的配 置,i5 软件升级的集成开发环境。	安装运行 Windows7/8/10 操 作系统的 PC
2	i5 APIs	一组应用编程接口,包括接口函数的 DLL,接口函数声明头文件,支持 PC 软件通过 API 与 i5 交换数据,实现联合控制。	安装运行 Windows7/8/10 操 作系统的 PC,支持 Windows 标准 DLL 加载及接口函数调 用的编程环境
3	i5 Firmware	运行于 i5 控制器嵌入式系统环境下 的,能执行 PLC 控制程序,并通过 EtherCAT 控制从站,以及通过以太 网与 PC、人机界面进行交互的程序	i5 控制器

## 4.3. 软件特点

## ● 符合国际标准 IEC61131-3 的编程语言规格

支持符合 IEC61131-3 标准的编程语言规格。支持依照 PLCopen 标准的运动控制指令和 依照 IEC 标准的各种指令组 (POU)。

● 符合PLCopen标准的运动控制功能块及其扩展支持

提供符合 PLCopen 标准的运动控制功能块,并可根据应用的特殊需求提供定制化的扩展 功能块,用户也可以自定义功能块,实现对控制工艺逻辑的封装。

#### • 整合时序控制与运动控制

控制器兼具时序控制和运动控制。在相同控制周期内执行时序控制、运动控制及 I/O 刷新。

控制周期与 EtherCAT 的过程数据通信周期一致。因此,可在固定周期内实现波动较少的高精度时序控制和运动控制。

#### ● 支持多任务

可为多个任务分配执行用户程序,并可分配任务优先级,分别指定执行条件和执行顺序, 通过对其进行组合,根据应用程序灵活控制。

#### ● 多种工作模式

(1) i5 独立控制:通过 MultiProg 编写调试 PLC 控制程序,可编译下载后脱机运行实现 独立控制;(2)与 PC 组合控制: PC 完成前置数据处理,通过调用 API 发送数据和控制命令 到 i5,启动 i5 的实时控制逻辑,并可读取 i5 的控制逻辑执行状态,根据状态启动下一轮组 合控制;(3)多控制器组网使用:根据被控制设备对象的规模,可按照工位分别用多个 i5 控制,并通过网络连接实现各控制器之间的数据和控制命令的发送和接收,实现更大规模的 生产线设备控制。

#### ● 丰富的安全功能

备有操作角色和权限设置功能;数据备份恢复功能;每个项目单独设定保护功能;防范 未授权的软件复制功能。

#### • 监控控制系统的各种状态

控制器监控包括安装的 I5 系列控制器、标准 EtherCAT 从站在内的所有控制器的事件。 发生异常时的处理方法在 MultiProg 或连接的人机界面中显示。发生的事件保存为记录。

#### ● 自动化软件 MultiProg

MultiProg 是通过一个软件涵盖控制器、周边设备及 EtherCAT 的设备的整合开发环境。 向不同的设备提供统一的操作性。支持从设计到调试、模拟、启动、开始运行后的变更等所 有工序。

## 4.4. 主要规格

项目			15-1213	15-1215	
	程序容量	大小		20MB	
		数量	POU定义数	3,000	
伯田			POU实例数	5,000	
細性	空量容量 /	带保持	大小	256KB	
		属性	变量数	10,000	
		无保持	大小	4MB	

		属性	变量数	90,000	
	数据类型	数据类型	数	2,000	
	控制轴数	控制轴最	大数量	32轴	8轴
		单轴控制最大数量		单轴32轴	单轴8轴
		线性插补控制最大数量		每轴组4轴	
テカ体制		圆弧插补控制轴数		每轴组2轴	
运动控制	最大轴组数		32组		
	运动控制周期		与 EtherCAT 通信的过程数据通信周期相同		
	位置单位		脉冲、mm、μm、nm、degree、inch		
	超调		0.00, 0.01 $\sim$ 200.00%		

## 4.5. 查看版本

乐创自动化技术股份有限公司将会不定期发布 i5 以及相关 PC 软件的升级,增加功能并 解决发现的问题,可以通过售后支持获得版本更新,若要获知当前使用的软件版本,可通过 以下操作进行。

## 4.5.1. MULTIPROG 版本

PC 软件安装完成后, 启动 MULTIPROG, 在主界面选择菜单【?】->【信息】, 打开对话框, 可获得 MULTIPROG 版本信息。

关于 MULTIPROG Express Release	
₩. <mark>8</mark> :	MULTIPROG Express Release 5.51 Build 396 许可证编号: 001 - 00002 - M - 01000 Copyright(c)2015 by PHOENIX CONTACT Software GmbH Langenbruch 6 D-32657 Lemgo Germany 电话: +49 5261 9373-0 传真: +49 5261 9373-726 网址: http://www.pheenixcontact=software.com 电子邮件: support-pcs@phoenixcontact.com
安装的软件包: LeetroXWM Core Leetro Autor MULTIPROG Core PHOENIX CON	复制信息 Mation Corp., Ltd 2.0.0 IACT Software GmbH 5.51.396
包详细信息: 该版本进行了测试: Windows 7 SP 1	A
Windows 8.1 Windows 10	-
	确定

图 4.2

## 4.5.2. 固件版本信息

固件版本由运行于 i5 的若干固件模块的版本组成。 获取固件版本信息的方法如下: PC 软件安装完成后,启动 MULTIPROG,打开 MULTIPROG 工程,选择菜单【联机】->【版本 信息】,显示所连接控制器的详细固件版本信息。

版本信息		X
刷新控制器类型Lee	tro_I5_1213#	反本1.8.2目标版本 Leetro_i5_1213_V1.8.2 ▼ 控制器升级
名称	版本	编译时间
LOG	1.4.1.2663	2019/11/20 16:2:52
HardWare	0.0.0.1213	2018/1/1 0:0:0
Uboot	1.3.0.0	2019/7/8 7:37:44
OS Image	1.5.0.3208	2020/7/30 15:4:34
FPGA	1.2.0.102	2018/3/2 12:6:23
OS Init	1.5.0.3365	2020/8/28 11:12:38
ModuleManager	1.8.0.3370	2020/8/28 15:9:10
Sys Performance	1.2.1.2237	2019/5/30 13:54:33
MasterIO	1.4.2.2840	2020/6/17 18:26:12
Security	1.3.0.2423	2019/7/25 12:59:55
Param	1.8.2.3390	2020/9/7 14:5:27
PCF	1.3.0.2407	2019/7/24 13:39:44
Modbus	1.8.0.3301	2020/8/13 13:56:20
mc	1.8.2.3418	2020/9/11 15:49:52
ControllerIO	1.7.0.2991	2020/7/21 16:6:59
PlcUnitTest	1.8.0.3295	2020/8/11 17:25:41
LtCom	1.8.0.3300	2020/8/13 13:55:35
ECLR	1.8.2.3386	2020/9/4 12:42:0
Sampling	1.4.2.2747	2019/12/12 16:11:17
SYS_Main	1.7.0.3052	2020/7/3 18:31:50

图 4.3

4.6. 使用步骤

## 4.6.1. 步骤的概要

按照以下流程使用 i5 控制器。

# *STEP* **1**. 软件设计

根据控制工艺要求规划设计用到的轴, I/O 等从站,以及控制程序的任务和逻辑。

STEP1-1 I/O 设计

STEP1-2 任务设计

STEP1-3 程序的设计

# STEP 2. 建立硬件调试环境

搭建由 i5 控制器和必要的从站单元及 IO 元件构成的开发调试环境。

# STEP 3.软件设定与编程调试

使用 MULTIPROG 制作 STEP1 设计的控制系统,配置从站,分配变量,编制任务和 程序,进行调试。 STEP3-1 配置从站

JILF J-I 出直/八川

STEP3-2 控制器设定

STEP3-3 编程
STEP3-4 调试

# STEP 4.安装与硬件的设定

在设备中安装控制器及各种从站单元,根据需要进行硬件的设定。

STEP 5.接线

进行网线和 1/0 的接线。

# STEP 6. 通过实际设备进行的动作确认与运行

连接至实际系统,下载工程。使用实际设备进行动作确认,开始运行。

## 4.6.2. 详细步骤

STEP 1.软件设计		
步骤	内容	参考
STEP1-1	• 外部 I/O 与从站单元的构成设计	
I/O 设计	• 外部设备控制的周期	
	• 程序处理的内容	
STEP1-2	• 任务构成	
任务设计	• 任务与程序的关系	
	• 各任务的周期	
	• 从站的控制周期	
	• 任务间变量的同步互斥方法	
STEP1-3		
程序的设计		
POU 的设计	• 程序	
	• 函数/功能块	
	• 各算法语言的选择	
变量的设计	• 区分 POU 通用的全局变量和仅各 POU 使用的	
	局部变量	
	• 设计用于访问从站的设备变量的变量名称	
	• 设计变量的名称、保持等各种属性	
	• 设计变量的数据类型	

# STEP 2. 建立硬件调试环境

步骤	内容	参考
	1. 准备一台调试用 PC;	《i5-系列硬件手册》
	2. 准备一台 i5 控制器;	
	3. 准备若干根据 STEP1 设计用到的包含轴,或者	
	I/O的从站单元;	
	4. 准备其它类型必要的从站单元;	
	5. 按照硬件手册说明进行连接;	

# STEP 3.软件设定与编程调试

步骤	内容	参考

STEP3-1		
配置从站		
1)建立工程	1. 使用 Multiprog 新建工程;或者打开已有工程进	《i5 系列用户手册》6.3
	行修改;或者基于模板创建工程	节
2)添加从站	1. 若从未在 MultiProg 中添加使用过某种类型的从	《i5 系列用户手册》6.4-
	站,则先导入从站的 ESI 文件;否则直接进行第 2	6.5 节
	步;	
	2. 连接从站到主站并自动扫描,或者手动添加从站	
	设备信息到设备树;	
	3. 对 PDO 设置进行修改或采用缺省的 PDO 设置;	
	对控制周期以及 DC 模式进行设置;	

3)将设备变量分配至	1. 通过建立 PDO 与全局变量的映射将 PLC 可访问	《i5-系列用户手册》6.6
I/O 端口	的全局变量分配至从站的 I/O 端口;	廿

4)"轴"和"轴组"的	1. 根据 Step1 中对软件的设计规划,增加轴或者轴	《i5-系列用户手册》6.6-
建立和参数配置并与	组,并将轴与从站建立映射关系;	6.7 节
伺服/轴模块从站建	2. 设置轴或者轴组的参数;	
立映射		

STEP3-2		
控制器设定		
1) 对控制器通讯参数	1. 在【视图】->【通信参数】页面对控制器的 IP 地	《i5 系列用户手册》12.3
设定	址通讯参数进行设置;若设置的 IP 地址与 i5 当前	节
	的 IP 地址不一致,则会用设置的地址修改 i5 的当	
	前 IP,修改会在下装后进行,但在重新上电启动后	
	生效;修改 IP 地址下装并重新上电启动生效后,将	

	导致 MULTIPROG 与 i5 连接超时中断,必须修改	
	MULTIPROG 的工程中的 IP 地址为修改后的 IP 地址	
	才能恢复与 i5 的连接;	
	2. 对 MODBUS 参数进行设置;设置参数将影响到	
	与 HMI 的连接,以及与基于 API 调用的 PC 端应用	
	程序的连接;	
2) 对控制器的系统时	1. 查看控制器的时钟是否正确, 若不正确, 可对其	《i5-系列用户手册》12.2
钟进行设定	进行设置; 若不需修改, 则忽略本步骤;	廿

STEP3-3		
编程		
1)导入运动控制功能	1. 将运动控制功能块库导入到 MultiProg 的工程;	《i5-系列用户手册》
块库		7.3.1 节
2)POU 的算法的建立	1. 根据控制逻辑采用 5 种编程语言之一编写控制	《i5-系列用户手册》第7
	程序,实现控制算法和工艺;	章
	2. 根据需要可编写自定义功能块,实现对可复用逻	
	辑的封装;	
	3. 根据需要可编写自定义函数,实现对可复用算法	
	的封装;	
3)任务的设定	1. 根据软件设计规划将 POU 设定到任务;	《i5-系列用户手册》第8
	2. 设定任务的优先级;	章
	3. 设定任务的周期或者执行条件;	

STEP3-4	1.	编译工程项目并修改编译错误;	《i5-系列用户手册》第9
调试	2.	下装工程,试运行;	<b></b> 章
	3.	调试修改控制逻辑;	

STEP 4.安装与硬	件设定	
步骤	内容	参考
	<ol> <li>1. 安装 i5 控制器到设备;</li> <li>2. 根据需要安装从站单元到设备;</li> </ol>	

# STEP 5.接线

步骤	内容	参考
	1. 连接各从站单元端口至控制设备;	《i5-系列硬件手册》
	2. 用网线连接 i5 控制器和各从站单元;	
	3. 用网线连接 i5 控制器和 PC;	
	4. 若有 HMI 则用网线连接 HMI(基于 MODBUS-	

	TCP)或者用串口线连接 HMI(基于 MODBUS-	
	RPU)	
5.	将电源连接到各从站单元;	
6.	将电源连接到 i5;	
7.	上电并检查 i5 面板指示灯显示是否正确;	

## STEP 6. 通过实际设备进行的动作确认与运行

步骤	内容	参考
	1. 检查 i5 控制器,各从站单元和设备的连接;	《i5-系列用户手册》第
	2. 下装调试完成的工程项目,并在设备上试运行;	9.3 节
	3. 动作确认正确后通过 MultiProg 将工程项目设	
	定为作为引导工程永久驻留;	
	4. 重新对设备开机进行操作确认;	

## 4.7. 快速使用示例

以下以 i5 连接控制一台松下 A5B 伺服实现单轴运动为例,说明使用的过程。

## 4.7.1. 调试环境准备

- 1) 按照第5章描述安装软件到开发用的 PC;
- 2) 准备一台 i5 控制器,一台松下 A5B 伺服,按照硬件手册连接。
  i5 的 ETHERCAT 端口通过网线连接至 A5B 的 X2A 端口;
  i5 的 ETHERNET 端口通过网线连接至 PC 的网口;
  i5 的电源端子通过电源线连接至 24V 电源;
  检查连接后 i5 上电;

## 4.7.2. 创建 PLC 工程

1) PC 上启动 MULTIPROG, 进入主界面;



图 4.4

2) 在 MULTIPROG 的主界面,选择菜单【文件】->【新建工程】,在弹出的窗口中选中【I5\_1213 工程】,单击【确定】按钮;



图 4.5

3) 进入工程主界面。选择菜单【文件】->【工程另存为】,在弹出的窗口中指定要保存的工程文件名,如 MyFirstProject,单击【保存按钮】。

💶 工程另存/压缩	沩				×
保存在(II):	🍌 新建文件夹	•	<b>G</b>	• ⊡ 对	
Ca.	名称	*		修改日期	i
最近访问的位置		没有与搜索条件匹配的	项。		
桌面					
<mark>篇</mark> 库					
() 计算机					
(1) 网络					
	•	III			Þ
	文件名(M):	MyFirstProject.mwe		-	保存(S)
	保存类型(T):			•	取消
压缩选项					
☑ 压缩用户I □ 压缩用件I	库(U) 库(W)	□压缩前端代码(F)			
□ 压缩回[]]	哖(w) 布局(P)				

图 4.6

## 4.7.3. 连接到控制器

在 MULTIPROG 的主界面,选择菜单【联机】->【扫描控制器】,在弹出的窗口中双击需要被 连接的控制器。



图 4.7

## 4.7.4. EtherCat 从站配置

4) 在 MULTIPROG 的主界面,选择菜单【视图】->【EtherCAT】,在弹出的窗口中左侧鼠标 右键单击【配置/资源】,在弹出菜单中选择【Scan EtherCAT Network】;

Master		
Unit Name	配置/资源	
Cycle Time [us]	1000	-
	Master   Gerena Unin Name Cycle Time (xn) Source MAC address	Matter Greenal Unit Name 赴唐/贞厚 Cycle Time (ad) 1000 Source MAC address

图 4.8

5) 【EtherCAT】根据物理连接顺序列举出所有的从站

Project Explorer	Device Editor			
* i RCBL/9578	General PDO Mapping Address Station Address	Variables	Advanced Options   Distributed Clock   Init Commands   1001   \$	
	Information			
	Name		51	
	Description		MADHT1505BA1	
	Vendor		Panasonic Corporation, Appliances Company (0x0000066F)	1
	Product Code		0x515050A1 (1364218017)	
	Revision Number		0x00010000 (65536)	
	ESI File		C:\ProgramData\PHOENIX CONTACT Software\MULTIPROG Express A58_V0_22.xml	٨
	Identification Value		Not Used	
	Ports			
	A		配置/党理	
	D		Net Available	
	8		Not Connected	
	c	0	Not Available	
				1

图 4.9

6)关闭【EtherCAT】窗口,在 MULTIPROG 主界面选择菜单【变量映射】,弹出以下窗口。 鼠标左键单击右侧列表栏的第一行选中,然后拖动右侧列表栏的卷滚条至最下面,按住 键盘上"SHIFT"键同时鼠标左键单击列表中最后一行,选中所有行。鼠标左键单击中间 【New】,即可将 PDO 资源映射到变量。

Variables Mapping			
N語ノ供得/PLC_SYS_TICK_CN NEE/治療/PLC_TASS_FICK_CN NEE/治療/PLC_TASS_DEFINI NEE/注意/PLCMODE_LOADI NEE/注意/PLCMODE_STOP NEE/注意/PLCMODE_HAIT NEE/治療/PLC_TICKS_PER_SI NEE/治療/PLC_TICKS_PER_SI Sak Mapping	IT (UDINT) ED (INT) OOL) NG (BOOL) (BOOL) BOOL) BOOL) EC (UINT) C (UIDINT)	* E X New	REIL/95/W     IN: SLTransmit PDO mapping LError code (UII IN: SLTransmit PDO mapping 1.Statuwerd (UI IN: SLTransmit PDO mapping 1.Statuwerd (UI IN: SLTransmit PDO mapping 1.Followis atma IN: SLTransmit PDO mapping 1.Touch probe p IN: SLTransmit PDO mapping 1.Touch probe p 1. SLTransmit PDO mapping 1. Following atm 10
Resource '配置/资源 (IN)':	<default></default>		
Resource '配置/资源 (OUT)':	<default></default>		
Memory Mapping VO Channel Shared Memo	ory		
Input:	0		
<b>C</b> 1 - 1	4		

图 4.10

映射后,拖动左侧列表卷滚条至最下面,可以看到资源映射的内容,同时可以在全局变 量表中看到对应的变量与变量的绝对地址。

ariables Mapping					
* HCHE/2009/51_Transmit_PUU	_mapping_1_mode (Sin		IN: SL Transmit PDU mapping	1. I ouch probe st	
IN: S1.Transmit PDO ma	pping 1.Modes of open		IN: S1.Transmit PDO mapping	1.Touch probe p-	
* 配置/按照/S1_Transmit_PDO	_mapping_1_Posi (DIN1	222	IN: S1.Transmit PDO mapping	1.Following error	
IN: S1.Transmit PDO ma	pping 1.Position actual	4.4	IN: S1.Transmit PDO mapping	1.Digital inputs (	
▼ 配置/资源/S1_Transmit_PDO	_mapping_1_Touc (UIN	X	OUT: S1.Receive PDO mapping	OUT: S1.Receive PDO mapping 1.Controlword (	
IN: S1.Transmit PDO ma	pping 1.Touch probe st	New	OUT: \$1.Receive PDO mapping	1.Modes of ope	
* 配置/资源/S1_Transmit_PDO_mapping_1_001 (DINT			OUT: S1.Receive PDO mapping	1.Target positio	
IN: S1.Transmit PDO ma	pping 1.Touch probe p		OUT: S1.Receive PDO mapping 1.Touch pro		
4 [			·	ь.	
sk Mapping					
Resource 配置/资源 (IN)':	<default></default>				
-	<default></default>				
Resource "配置/贷跟 (OUT)":					
Resource "配置/资源 (OUT)": Memory Mapping					
Resource '환폐/영과 (OUT)': Memory Mapping @ I/O Channel					
Resource 配置/管線 (OUT): Memory Mapping I/O Channel Shared Memo	эry				
Resource 配置/高線 (OUT): Memory Mapping I/O Channel Shared Memo Input:	ary				

图 4.11

## 4.7.5. 配置轴参数

7) 选择菜单栏【视图】->【运动控制】运动控制配置界面



图 4.12

8) 鼠标右键单击左侧分栏中【轴设置】,在弹出菜单中选择【新建轴】,则将在【轴设置】 节点下增加【MC\_Axis001】节点。鼠标左键双击【MC\_Axis001】节点,打开【MC\_Axis001】 标签页窗口,确认【轴基本设置】组中的【输出控制】一栏的下拉列表选择为增加的从 站节点。

输设置	轴基本设置 单	· 接其资金、操作资金、随你资金、监守处回资金、供金升料资金、创助和EA的金金、	
10C_Axi#001 轴组设置	袖号 袖使能 袖英型 控制英型 輸出控制	1  +#9780304 - 	

图 4.13

## 4.7.6. 使用功能块编写控制逻辑

9) 在 MULTIPROG 右侧【编辑向导】窗口,选中"MC\_Reset",按住鼠标左键将其拖动到 中间的"代码 main"窗口空白的地方,放开鼠标左键,则弹出如下窗口,然后单击【确 定】按钮,关闭窗口。



图 4.14

添加完成后如下图所示:

The second second second						1984-9	
Augustali Augusta					- 19	COLUMN COL	
D blockbar Billers						AND AND	
I supr						St. 1 pheatherPes	
		TC Fee	1.1			■ W_SrpletOwerride ■ W_Srpleto	
		R. Jes	ant.			# W_S splits (Bovella) # W_Sec.	
			2000			# 95 Innedteralten	
		arecore.	2009			# W_AwerConsho	
						S. Swellindha	
			Arter			S. SowsLandel	
		1	Children I.			S. Jower	
		1.1				S. R. Anathermal Fels	tim city
						R. feathestforer	
						St. fon Bigit alby	1
						R.K.SeaParmeter	
						B. Japat	
						S. S. DelParation	
						. W. Speckronkberle	**
						R largeCastrol	Υ.
	1+0					S. Joshi Prate	4 put
P 10 10 10 10 10 10 10 10 10 10 10 10 10	& these					The first of ar sant of	



**10**) 使用与上一步相同的方式,添加 MC\_Power, MC\_MoveRelative 到代码中,添加完成 后,如下图所示。



图 4.16

11) 添加输出触点。在"代码 main"窗口中的空白处,单击鼠标右键弹出菜单栏,选择 触点。

2 A state 2 A state	04-2 04-2 04-3 04-4 04-4 04-4 04-4 04-4 04-4 04-4		Balls Iren Salls Iren Salls Iren Salls Iren		- 10 - 20 - 20 - 20 - 20 - 20 - 20 - 20 - 2	E. Jacobie E. Jacobie etc. etc. etc. etc. etc. etc. etc. etc	ing ing ing ing ing ing ing ing ing ing			1	All PLOPER, LID S.B. S.R. StableRCom S.R. SteableRCom S.R. Streaturer S.R. Streaturer
	0142 0142 0144 0144 0144 0144 0144 0144		Anno Anno Anno Anno Anno Anno Anno Anno			Ki der die Ki der die ets teneralister ere tity land tat	Ales a base a base a base a base a base a				88 18. JustiaEcon 18. StatisEcon 18. Steater 18. JustiaEcon 18. JustiaEcon
	Cherry OH-42 OH-42 OH-43 OH-44 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42 OH-42		Age - in Age - in Add for Add for Add for A			ette Staronallydere Gitte Land Lat	Alexandra Juna - Real or Local of a				AR IN. InshielDraw IN. Hornissee IN. Revelation IN. Revelation
(2) Here ↑ 4 (4) (3) Here (4) Here (5) Her	0442 0444 0444 0444 0442 0442 0442 0442		<ul> <li>App - App -</li></ul>			ette Innovalladete Elity Landini	Star				1187, Brahland om 1187, Billoweter 1187, Breedinsar 1187, Breedinsar
Λg         Holig           Λg         Holig </td <td>01/7 Cat-V D4-3 D4-4 D4-4 D4-2 D4-2 D4-2 D4-2 D4-2 D4-2</td> <td></td> <td>- Jakis Inn - Jakis Inn Yuk Sawi</td> <td></td> <td></td> <td>cia insendistera ener City bendist</td> <td>line of Rect of Longitudients</td> <td></td> <td></td> <td></td> <td>\$18C_BSCounter \$18C_Revelopment \$18C_Revelopment</td>	01/7 Cat-V D4-3 D4-4 D4-4 D4-2 D4-2 D4-2 D4-2 D4-2 D4-2		- Jakis Inn - Jakis Inn Yuk Sawi			cia insendistera ener City bendist	line of Rect of Longitudients				\$18C_BSCounter \$18C_Revelopment \$18C_Revelopment
<ul> <li>№ 18400</li> <li>○ 48-240</li> <li>&gt; € 898-26</li> <li>&gt; € 898-26</li> <li>&gt; € 898-26</li> <li>&gt; № 898-26</li>     &lt;</ul>	Call+V Dol+3 Dol+6 Dol+1 Dol+2 III Dol+2 III Dol Dol Dol	111	taile fran Via Band	F	- 00 - 20 - 10 - 10	annaladere ann covr bandint	Bar of Art of Pa Constituents				SINC Revelament
○ 田山田 ○ 田山田 ○ 田田田 ○ 田田田 ○ 田田田田 ○ 田田田田田 ○ 田田田田 ○ 田田田 ○ 田田 ○ 田 ○	054+3 054+6 054+6 054+2 04+2 05 05 05 05 05 05 05 05 05 05 05 05 05		Via Breed	F		city bendline	fire or other				1 1 LSC Read(b Sect
· · · · · · · · · · · · · · · · · · ·	04+4 04+4 04+1 04+2 15 15 16(5) 0-1	-F	Seed	F	- 1	cov bentine	Considered as				ATEL BALANTIN
X 행동/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/GG 아씨/G 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇	0444 0441 0442 19 19 19 19	E.	Seed	F	-	dentise.	200				THE Pritelit
(C)     (	04+1 DH+2 11 15 DH51 D-	[					Real Property lies				LIK Fritelbaart
Kenedowa	04+2 15 15 10-15						free little				S. Abert frigger
中 ジアフロル デオート デオート マント マント マント マント マント マント マント マント マント マン	11 15 10-				and loss						E M. Ceydurable
(1) 日本のの 「日本の」 「日本の日本の」 (1) 日本の日本の (1) 日本の日本の日本(1) 日本の日本(1) 日本の日本(1) 日本の日本(1) 日本の日本(1) 日本の日本(1) 日本の日本(1) 日本(1) 1) (1) (1) (1) (1) (1) (1) (1) (1) (	185) D-										S. Ceplishis
법: 日本(100) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	185) D-										*K.Seplandistel
<ul> <li>デアドルに</li> <li>12. (An eff.)</li> <li>13. (An eff.)</li> <li>14. (An eff.)</li> <li>14. (An eff.)</li> <li>14. (An eff.)</li> <li>15. (An eff.)</li> <li>16. (An eff.)</li> <li>16. (An eff.)</li> <li>16. (An eff.)</li> </ul>	D										S.C. SeptendertPor
15. martulas 40. dataveru 44. statutas activeras	D										K.Sepfieret
<ul> <li>O ISSNETO</li> <li>NE XRC280</li> <li>NC 8449862</li> </ul>	and a second										The septemented
04 2.0(20) 00 000002											E (aslester)
Int Barrisso	n										W Tran
10 6-016 b											IN Innediatelten
	. 95										S. Reendbanbate
11 600(4)	17										K. Beretteribe
N086(2)											* K. Rees (irila)
1786-843											S.C. Seeal india
111 000											K freelingsl
#23.2#	102										W Burglelauity
외문제사다											S. K. Junt
30.65 M											S.K. Sendlet unliften
A water	10. 10.0										S. Tandártunt Tal
( Presentation											S. Indariellerie
											*K. Leolarisisfs
											# M., ReefDigitalle
										1.1	K. Aradiat sendta
											R. featfitheter
											W. Incat
											BW fatharride
											SK. Tetferitins.
											S. 17. 17. 19
											St. Synchereitent
0											*K_lynaMereFelor
and a second sec											SK_lergesCentral
8 (5) 20 B											*

图 4.17

然后双击触点,弹出触点属性对话框,单击确定按钮,关闭对话框。

de She (an)	中心英国	
2001 CO		确定
C001	◎ 開卧(L) ○ 王時(C)	取消
数据类型(D):	問部貢重祖(¥):	
BOOL •	Pefault -	帮助(H)
用法(U):	全局变量组(A):	ñe -
VAR • RETAIN	□ 📁 物理硬件	
初值(I):	白 — 配置	
	E- M.W.	
L/O 読計 (S):	NewGroup	
	System Variables	
篇( <b>律</b> (F),		
PDD EOPC 目標欄(X)	团 显示工作单的所有变量(W)	
初值作为财认值.		
A		
89 C) / 35 PB		
and the second		

图 4.18

- 12) 添加电源导轨线。鼠标右键单击触点 C001,弹出菜单,选择左电源导轨线。
- 13) 连线。按照程序逻辑,将功能块与触点进行连接,鼠标移到引脚的引线上的小圆点
  - 上,然后按住鼠标左键,拖动至另一个小圆点,形成连线。连接完成后如下图:



图 4.19

14) 给功能块引脚赋值。鼠标双击功能块 Axis 引脚,弹出【变量属性】窗口,如下图。 在名称下拉菜单中选择 MC\_Axis001,单击确定按钮关闭窗口。

变量属性		23
名称(N): MC_Axis001 数据类型(D): AXIS_REF → 用法(U): VAR_GLOBAL → RETAIN 初值(I): (AxisNo := 1) I/O地址(S):	定义范围 ③ 局部(L) 全局(G) 局部変里组(V): ② Default ④ 局変量组(A): ● 物理循件 ● 配置 ④ KutherCAT Variables ■ NewGroup	<u>确定</u> 取消 帮助(H)
描述(E): Axis creat by system □ PDD □ OPC □ 陰蔵(X) □ 初值作为默认值.	<ul> <li>☑ 显示工作单的所有变型(₩)</li> <li>☑ 技组名排序</li> </ul>	



双击功能块 MC\_MoveRelative 的 Distance 引脚,弹出【变量属性】窗口,如下图。修改变量名称为 dis,并设置初值为 10000.0。单击确定按钮关闭窗口。

变量属性		23
名称(N): dis	定义范围 ◎ 局部(L)   全局(G) 局部变量组(V): ② Default 全局变量组(A): □ ● 物理硬件 □ ● 敬彊 □ ● 愛源 EtherCAT Variables NewGroup System Variables	确定 取消 帮助(H)
描述(E):		
■ PDD ■ OPC □ 隐藏(X) □ 初值作为默认值.	<ul> <li>☑ 显示工作单的所有变里(₩)</li> <li>□ 按组名排序</li> </ul>	

图 4.21

使用相同的方法给 MC\_MoveRelative 功能块的 Distance、Velocity、Acceleration、 Deceleration 引脚赋值,并设置为不同的名称。结果如下。



## 4.7.7. 调试试运行

回到 MULTIPROG 主窗口,单击工具栏中的【制作】按钮,或者选择菜单【建立】->【制作】。在消息窗口中确认编译没有错误。

		_
	main	-
	正在生成IEC代码	
T	正在收集被资源'资源'所使用的POU···	_
	正在为资源'资源'生成IEC代码···	
	正在为资源、资源、创建实例树	=
	正在生成专用目标代码,用于配置 配置	-
	正在生成专用目标代码,用于资源 资源	
御殿	│ 💡 □个错误,1个警告	-
197	④ ▶ \ 建立 \ 错误 \ 警告 \ 信息 \ PLC出错 \ 打印 /	_



2) 在 MULTIPROG 主窗口,单击工具栏中的【工程控制】按钮,或者选择菜单【联机】->【工程控制】,打开如下窗口。依次单击【下装】按钮,【冷启】按钮。

资源	- • 💌
状态: 运行	
停止(S)	冷启(0)
复位(R)	──暖启(\\')
[错误(E) □	热启(T)
下装(D)	
更多(M)	信息(1)
关闭(C)	帮助(H)

图 4.24

3) 在 MULTIPROG 主窗口,单击工具栏中的【调试开/关】按钮,切换到调试状态下。鼠标 左键双击"COOO"变量,弹出窗口,确保"值"一栏的单选按钮置于"TRUE",单击【覆 盖】按钮,即可切换变量 COOO 的值为 TRUE,依次启动 PLC 程序中的 MC\_Reset 功能块, MC\_PowerOn 功能块和 MC\_MoveRelative 功能块,将伺服切换到 PowerOn 状态,并启 动一个相对运动,。

调试:资源		X
强制/覆盖		断点
C000		设置(S)
值		复位(R)
TRUE	FALSE	全部复位(A)
强制(0) 夏位强制(E) 夏位强制(E) 夏位强制(A) 夏位强制(A) (L)	覆盖(W)	値的显示 ● 标准(D) ● 十进制(M) ● 十进制(N) ● 二进制(Y) 実数値 寛度(I): 精度(P): 3 7 ▼ IEEE 格式化(M)
关闭(C)	信息印	帮助(H)

图 4.25

4) 再次鼠标左键双击"COOO"变量,弹出窗口,确保"值"一栏的单选按钮置于"FALSE", 单击【覆盖】按钮,即可切换变量 COOO 的值为 FALSE,通过 PLC 程序中的 Reset 功能 块,和 PowerOn 功能块,将伺服切换到 PowerOff 状态。

## 5. 软件安装

## 5.1. PC 软件的安装

## 5.1.1. 安装须知

- 安装本软件需要管理员权限,否则部分安装文件不能被写入将会造成安装失败。
- 安装过程中,文件会被复制到安装目录,中途取消安装可能会有部分文件遗留在安装目录。
- 本软件在安装过程中可能需要重启,才能完成安装,所以请在安装本软件前,关闭 其他应用程序,并根据安装提示进行安装。

## 5.1.2. 安装需求

## 5.1.2.1. 支持的操作系统

MULTIPROG 软件安装在以下操作系统: Windows 7(32 位或 64 位 ) Windows 10(32 位或 64 位 )

## 5.1.2.2. 安装的应用软件

以下软件将会被安装

名称	版本
MULTIPROG	Ver.5.51.x
WinPcap	Ver.4.1.x
.NET Framework	Ver.3.5SP1
	Ver.4.5.2
Vc++ Redistributable Package	2005
	2009
	2010
	2015

## 5.2. 软件安装

## 5.2.1. MultiProg Express 安装

- 1) 管理员身份运行 MultiProg Express\_5.51\_xxxx.exe
- 2) 仔细阅读许可证协议,如果接受许可协议中的条款,单击【我接受】开始安装





3) 选择安装选项, 单击【开始安装】安装。



图 5.2

4) 等待安装完成。在安装过程中可能需要重启,请在重启后重复 1-5 步骤继续完成安装。





5) 单击【完成】完成安装。



## 5.3. PC 软件的卸载

# 5.3.1. 卸载 MultiProgExpress 打开控制面板

- 1) 打开控制面板
- 2) 选择 MultiProg Express,然后右键选择卸载按钮。
- 3) 单击【卸载】开始卸载。





4) 等待卸载完成,单击完成按钮退出卸载程序



图 5.6

## 6. 使用 MULTIPROG 创建 PLC 工程

## 6.1. MULTIPROG 的工程组织基本概念

IEC61131-3 标准着眼于清晰明了地描述一个复杂的控制问题,并尽可能地通过模块化的组织方式来减少代码的重复编写。IEC61131-3 标准的软件模型采用层次结构来表示,每层的元素和各层的相互关系如图所示。 (FU 有时也称作 FC,是功能的简称)



图 6.1

图 6.1 IEC61131-3 软件模型结构示意图

MULTIPROG 在工程组织结构上完全遵循了 IEC61131-3 标准定义的层次结构和模块 化的思想。打开 MULTIPROG,打开已经建立的工程或者使用工程设置向导新建一个工程, 就可以看到屏幕左侧的工程树窗口中列出了当前工程所用到的所有节点信息, 如图所示。



## 图 6.2工程树窗口



#### 图 6.3工程树窗口

工程树中的"库"、"数据类型"、"逻辑 POU"和"物理硬件"这 4 个节点在使用工程设置 向导建立工程之后就必定会出现,但是根据所建立的工程的复杂程度,"库"和"数据类型"节 点可以为空节点。单击"物理硬件"节点前面的加号或者点击工程树窗口下面的"硬件"选项卡, 就可以展开物理硬件的配置(Configuration)。在使用中文语言设置,并且未改动默认的建 立名称的情况下,可以看到"配置(Configuration)"、"资源(Resource)"、"任务(Task)"、 "Global\_Variables"和"IO\_Configuration"这几个节点,如图所示。其中"任务"节点是可以为 空的,其余节点在工程建立之后则必定存在。



图 6.4 物理硬件节点展开示意图

## 6.1.1. 配置(Configuration)

配置是一种结构元素,对应于 IEC61131-3 标准中最外层的框架,用于定义针对特定应用的控制系统的特性,其含义是确定了解决某种特定的控制问题的系统框架,包括硬件设备、处理器资源、I/O 通道、存储和通信地址等。

通常,一个配置就对应于一台实际的控制器。当需要处理的控制问题较为复杂时,例如 一条生产线的自动化,则需要几台控制器的相互配合,此时可以定义多个配置,每个配置都 是相互分离的个体,不过可以通过通信接口实现信息的交互。

#### 6.1.1.1. 增加配置



该功能仅在 MultiProg Professional 版本提供。Express 版本只支持工程树中存在一个配置。

当需要增加配置时,可以在"物理硬件"或者"配置:eCLR"节点右击,在弹出的开始菜单 中选择"插入…"->"配置",如图所示。在弹出的对话框中填入配置名称,并选择好相应的 CPU 类型,点击"确定"按钮即可,如图所示:(目前暂不支持在一个项目中添加多个配置)



图 6.5 右键菜单插入配置

名称(N):	类型	确定
Config2	◎ 配置(C)	Hn cak
程序类型(M):	○ 资源 (R)	40.75
	────────────────────────────────────	帮助(H)
PLC类型(Y):	○ 描述(D)	
eCLR	● 変里(V)	
	□□□□ OFB头例	
	模式:	
	◎ 插入(I)	
그 세하스는 여 것 수 년 (22)	◎ 追加(A)	

#### 图 6.6 配置插入对话框

## 6.1.1.2. 配置的其他操作

如需要删除或者修改配置的定义,同样可以在需要操作的配置节点上使用右键菜单,选择相应的功能即可。

## 6.1.2. 资源(Resource)

资源位于 IEC61131-3 软件模型的第二层,为运行具体的程序提供支持。资源反映了控制器的物理结构,为程序和 PLC 的物理输入输出通道提供一个接口。资源是标准定义的一个语言元素,是作为执行 IEC 程序的虚拟界面。一个资源可以加载、启动和执行多个相互独立的程序。

资源需要通过资源名称来加以区别,一个资源相当于一个 CPU,所以资源也可以简单 理解为控制器里的微处理器单元。 在使用工程向导建立工程的情况下,默认选择为"Leetro\_I5\_1213",这个 CPU 类型代表的就是 MULTIPROG 运行在 i5 系统上的 PLC。

## 6.1.2.1. 修改资源的属性

右击"资源"节点上,右键菜单中选择"属性",如图所示。



图 6.7 修改资源的属性

在弹出窗口的"PLC/处理器"选项卡中的"处理器类型"下拉框中选择其他种类额 CPU, 如图所示。

'资源'	×
名称	PLC/处理器 属性 安全性
i	
	处理器类型(R):
	Leetro_I5_1213
	Leetro_I5_1213
	eLLK_Simulation
	确定 取消 应用 (A) 帮助

图 6.8 更改处理器类型

"eCLR\_Simulation" CPU 类型为 PC 仿真, 可用于调试 PLC 逻辑。

#### 6.1.2.2. 设置资源

对于资源的设置主要是定义通信接口的类型和参数,以及在线监视时的数据刷新速率。 右击"资源"节点,在弹出的快捷菜单中选择"设置"菜单,如图所示,其弹出的窗口如图所示。
● 物理硬件 ● ● 配置: eCLR ● ● 医第 : Lettro_I5_12 ● ● 任务: CUCLI ● □ 程序: em Global_Variable: ■ ID_Configuratio	<ul> <li>插入(1)</li> <li>耐除(D)</li> <li>前切(C)</li> <li>気利(Q)</li> <li>私防(P)</li> <li>从外部交量创建全局、 从全局交量中更新分 创建Bootproject</li> <li>属性(R)</li> <li>设置(S)</li> </ul>	Pelete Ctrl+X Ctrl+C Ctrl+V 時变量(G) 部变量(S)
● 工程 图 硬件 图 6.9 对	资源进行认	}習

eetro_I5_1213的	资源设置 🛛
通信	
类型:	TCP/IP
参数:	192.168.1.88 ·p41100
版本	
建立设置:	eCLR EtherCAT V1.8.4
更新建立设置()	Build settings行为:
○ 自动更新	if (A)
<ul> <li>● 更新前し</li> </ul>	Bia(B)
⑦ 不更新(	N)
0 1 22010	
在线更新	
时间间隔	10 臺秒 (范围: 060000)
编译器选项	
□ 堆桟桧3	Ŧ
☑ 新细计5	- 見給奋
网络讨伴的	
C STICK	(-1) 200 ()
	确定(0) 取消(C) 帮助(H)

图 6.10 资源的设置窗口

"建立设置"是用于定义将 IEC 代码编译生成下载到控制器里执行的机器码的一整套 库定义的文件,不同的版本生成的机器码可能不同,所以一般需要选择最新版的"建立设置" 文件。

"在线更新"的"时间间隔"指的是在使用 MULTIPROG 的 Debug 模式时,控制器内部 变量的值在 MULTIPROG 界面上显示的更新周期。

# 6.1.2.3. 全局变量

工程树中的"全局变量"节点是每个资源(Resource)中惟一存在的。双击该节点即可打 开全局变量工作单,如图所示。

名称	类型	用法	描述	地址	初值	保持	P	0	TB	隐藏	初	默认的…	冗余变量
PLC_SYS_TICK	DINT	VAR_GL		%MD1.0									
PLC_TASK_DE	INT	VAR_GL		%MW1.4									
PLCMODE_ON	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.2016.0									
PLCMODE_LO	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.2017.0									
PLCMODE_STOP	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.6.0									
PLCMODE_RUN	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.7.0									
PLCMODE_HALT	BOOL	VAR_GLOBAL	TRUE : current PLC mode is	%MX1.8.0									
PLC_TICKS_PE	DINT	VAR_GL		%MD1.2000									
PLC_MAX_ERR	DINT	VAR_GL		%MD1.2004									
PLC_ERRORS	DINT	VAR_GL		%MD1.2008									
PLC_TASK_AV	INT	VAR_GL		%MW1.2012		1		1					
PLC_SYSTASK	INT	VAR_GL		%MW1.2016									
PLCDEBUG_FO	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.2018.0									
PLCDEBUG_BP	BOOL	VAR_GL	TRUE : one or more	%MX1.2019.0									
PLCDEBUG_P	BOOL	VAR_GL	TRUE : current PLC mode is	%MX1.2020.0									
Vin1	DINT	VAR_GL											
Vin2	DINT	VAR_GL											
Vout	DINT	VAR_GL						V					
Trig_Enable	BOOL	VAR_GL											
Trig_State	BOOL	VAR_GL											
E Global_V													

### 图 6.11 全局变量工作单

全局变量工作单可以简单理解为全局变量表,在其中声明的每一个变量对于其所在的资 源来说都是唯一存在的。同时,对于在其所属的资源中存在的任何一个程序,全局变量都是 可以被访问的,这样就方便了同一资源中的不同程序交换数据和共享信息。

全局变量工作单中比较常用的是"名称"、"类型"、"地址"和"初值"4个属性。

有效地址格式包含:

- 百分比号 "%"
- 位置前缀"M"(针对存储器变量)、"I"(物理输入), "Q"(物理 输出)
- 尺寸前缀"X"(BOOL 1 位), "B"(字节 8 位), "W"(字 16 位),
   "D"(双字 32 位), "L"(长字 64 位),
- 存储区符号 0、1、2 或 3, 后加点号,
- 该字节的存储器地址,后加点号(BOOL 类型),
- 比特位位置(BOOL 类型)。

总结为: %MXMemoryAreaSign.ByteAddress.BitPosition

**%MB**MemoryAreaSign.ByteAddress

**%IX.** ByteAddress. BitPosition

%IB. ByteAddress

%QX. ByteAddress. BitPosition

%QB. ByteAddress

6.1.3. 控制器

PLC 工程需要按照设定的控制器(处理器)类型进行编译、并下装到对应的控制器中。MultiProg 软件已适配多种类型的控制器,用户可根据实际所使用的硬件,

在工程中选择相应的控制器类型及版本进行开发。设定的控制器与实际连接的控制器不一致(包括版本不一致,类型不一致)会导致程序无法下装。

### 6.1.4. 设置控制器类型及版本

具体操作方式如下:

方式 1: 通过扫描连接控制器,自动设定控制器类型。在菜单栏单击【联机】-【扫描控制器】打开扫描控制器窗口



图 6.12

双击控制器图标,可以连接到相应的控制器,同时打开控制控制对话框,并将工程 设置成与所连接控制器相同的类型和版本

- 方式 2:
  - 手动选择控制器类型。 在工程树->【硬件】选项卡中,右键单击【资源】,选择 【属性】,弹出资源属性窗口。在下拉菜单中选择相应的处理器类型。

<ul> <li>● 物理硬件*</li> <li>● 資置: eCLR*</li> <li>● 資源: Leetro_I5_1213*</li> <li>● (Interview: Leetro_I5_1213*)</li> <li>● (Interview: Leetro_I5_12</li></ul>	'资源'     ×       名称     PLC/处理器   属性   安全性
	处理器类型(R): <u>Leetro_I5_1213</u> ▼ 确定 取消 应用(A) 帮助

图 6.13

手动选择控制器版本。在工程树->【硬件】选项卡中,右键单击【资源】,选择【设置】,弹出资源设置窗口。在【建立设置】中选择相应的版本。

Leetro_I5_1213的资	源设置
通信	
类型:	TCP/IP 🔹
参数:	192.168.1.88 -p41100
# <b>*</b> +	
- 版4	
建立设置:	eCLR EtherCAT V2.0.0
更新建立设置(Bui	ld settings行为:
◎ 自动更新(/	Α)
◙ 更新前询问	ק(B)
◎ 不更新(N)	
在线更新	
时间间隔:	10 毫秒 (范围: 060000)
编译器选项	
□ 堆栈检查	
☑ 数组边界树	
- 经过优化的	
	确定(0) 取消(C) 帮助(H)

图 6.14

#### 6.1.4.1. 控制器类型

目前常用型号有 i5-1213, i5-1215, MPC-8801, 规格请参考章节 4.4。 具体型号差异请参考硬件手册或咨询经销商。

#### 6.1.4.2. 控制器版本

因软件版本迭代更新,形成不同的软件版本。用户在使用过程中,需注意选择与实际控制器版本相匹配的版本进行编译、下装操作,否则会导致 PLC 程序无法正常运行。

用户也可通过 MultiProg 对控制器进行升级,将控制器升级到指定的软件版本。具体操 作方式请参考章节 12.1。

#### 6.1.5. 任务(Task)

任务处于 IEC61131-3 标准软件模型的第三层,其用于管理程序组织单元(POU)在运行期的程序实例。可以把任务理解为一个容器,其中装载的就是 POU(POU 的概念将在下一章阐述)的各种元素,作用是规定了其中装载的 POU 元素何时处于运行状态,何时处于非活动状态。

#### 6.1.5.1. 任务类型

为了能更清晰地区分不同功能的任务,方便编程人员使用,MULTIPROG 定义了 4 种 任务类型,分别是 Default、Cyclic、Event 和 System,这些任务的优先级依次升高。

**Default** 任务:相当于传统 PLC 中运行的任务(或者程序),从控制器上电启动之后 便一直循环往复地运行,可以被其他三种任务打断。

**Cyclic** 任务:顾名思义是周期性的任务,该类型任务具有一个重要属性"运行间隔",每 隔这个间隔时间,任务即被投入运行,可以被 Event 和 System 任务打断。

**Event** 任务: 该类型任务是基于事件驱动的,即在控制器正常运行时该任务处于非激活状态,一旦对应的事件被触发,在情况合适的条件下,该任务就会立刻被激活投入运行。该类型的任务一般用于处理比较紧急的事件。每个任务都需要绑定到一个事件号,每个事件号对应一种特定事件,支持多个 Event 任务绑定到同一个事件号。

System 任务: 该类型的任务也是事件驱动的, 与 Event 任务不同的是, 触发 System 任务运行的时间都是非常紧急的事件, 一般都是控制器运行期的异常(Exception)。System 任务在创建时需要指定触发该程序的事件。System 任务可以抢占任何其他类型任务的运行。



图给出了 Default 任务的运行模式, 图表示了 Cyclic 任务和 Event 任务、System 任务的运行模式及相互之间的优先级关系。



## 6.1.5.2. 修改任务属性

鼠标右键点击工程树中"Tasks"节点下属的某个具体任务的节点,在快捷菜单中选择"属性...",如图所示。

<ul> <li>→ 物理硬件</li> <li>→ 配置: eCLR</li> <li>→ 逆 资源: Leetro_I5_1213*</li> <li>→ □ Tasks</li> </ul>					
白一一一 代务 一 一 1 利 一 ① Global J 一 ① IO_Confi	插入(I) 删除(D) 参 剪切(C) 量 复制(O) 通 粘贴(P) 量 属性(R) ⑦ 设置(S)	Delete Ctrl+X Ctrl+C Ctrl+V			
《學工程 翻 褐色					

图 6.17 修改任务属性

在弹出窗口中选择"类型"标签页,在"任务类型"下拉框中即可选择任务的类型,如图所示。

'MCTask'				<b>-</b> ×
名称 类型				
住务类型:				_
DEFAULT				-
	确定	取消	应用 (A)	帮助

图 6.18 任务属性窗口

### 6.1.5.3. 设置任务参数

任务的类型不同,其需要设置的参数也不一样,下面分别进行说明。在需要更改参数的 任务节点上点击右键,在弹出菜单中选择"设置..."选项,如图所示。

・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	tro_I5_1213* 活入(1) 删除(2) 予 質切(2) ● 質切(2) ● 行助(2) ● 行助(2) ■ 属性(2) ⑦ 设置(2)	Delete Ctrl+X Ctrl+C Ctrl+V
(学工程) 提供		

图 6.19 更改任务参数

### Default 任务:

如图所示为 Default 任务的参数设置窗口,其中只有一个参数需要设置。"监视定时时间"为监视定时器的倒数计时时间,当使能了监视定时器之后,监视定时器就会从"监视定时时间"开始倒计时,当计时为零时,监视定时器就会向运行的任务发出询问,若运行的任务没有响应,则监视定时器即会认为任务运行出错,此时就会产生一个运行期异常,对应的 System 任务就会被激活。

提示:当控制器的负载较重时,可以适当增大"监视定时时间"或者关闭监视定时器,防止系统运行出错停止。

'P_ADDER'的任务设	置	X
事件(E):		•
优先权(P):		
监视定时时间(W):	100	臺秒 📝 使能监视定时器(N
	确定	取消

图 6.20 任务参数设置窗口

## Cyclic 任务:

图所示为 Cyclic 任务的参数设置对话框,其中需要设置的参数有"间隔"、"优先权"和 "监视定时时间"3 个。"间隔"就是任务的运行周期,"优先权"就是任务的优先级,只是这个 优先级是在所有的 Cyclic 任务的范围内定义的。默认设置下的优先级范围是 0~31。

'任务'的任务设置		8
间隔(1):	100	毫秒
优先权(P):	4	
监视定时时间(W):	1000	毫秒 📝 使能监视定时器(N
	确定	取消

### 图 6.21 Cyclic 任务参数设置窗口

### Event 任务:

图所示为 Event 任务的参数设置窗口,和 Cyclic 任务不同, Event 任务主要关注的该 任务需要绑定的事件号,默认的事件号的范围是 0~5,共 6 个事件,事件的内容可以自行 定义(详情方法请联系乐创技术人员)。Event 任务的优先权的范围为所有 Event 任务内。 默认设置下的优先级范围也是 0~31。

'P_Event'的任务设置	Ē	X
事件(E):	事件0	•
优先权(P):	0	
监视定时时间(W):	100	臺秒 🛛 使能监视定时器(N
	确定	取消

图 6.22 Event 任务参数设置窗口

### System 任务:

System 任务的参数设置窗口见图。因为 System 任务具有最高的优先级,并且一旦相 对应的事件被触发,该任务会立即得到执行,所以"优先权"和"监视定时时间"不可配置。

'P_Sys'的任务设置		X
系统事件(S):	暖启动	•
优先权(E):		
监视定时时间(₩):		
	确定	取消

图 6.23 System 任务参数设置窗口

### 6.1.5.4. 增加删除任务

一个资源可以支持不止一个任务的运行,即在宏观上,所有这些任务可以同时运行。添加任务可以通过在"Tasks"节点上点击鼠标右键,在快捷菜单中选择"插入…"->"任务"实现,如图所示。

□ 🥮 物理硬件   □ 🧊 配置 : eCLR	
亩	
百· 插入①	任务(工)
IO_Configuration*	
⑦ 工程 Ⅲ 硬件	

图 6.24 添加任务

在弹出窗口中填写添加任务的名称并选择任务的类型,点击"确定",如图所示。

插入		X
名称(N):	类型	确定
T_anr	○ 配置(C)	町当
程序类型(M):	<ul> <li>资源(R)</li> <li>(R)</li> </ul>	41/19
· · · · · · · · · · · · · · · · · · ·	● 程劳(1) ● 程序实例(P)	帮助(H)
任务类型(Y):	○ 描述(D)	
CYCLIC •	<ul> <li>○ 受重(V)</li> <li>○ FB实例</li> </ul>	
	模式:	
	◎ 插入(I)	
□排除在编译之外(X)	◎追加(A)	

图 6.25 编辑所添加任务的属性

注意:任务的名称长度是有限制的,不能超过7个字符,并且前6个字符中必须有1个 是不同的。

当任务不再需要时,可以在工程树节点上点击右键,在弹出的快捷菜单中选择"删除"项 将任务删除。

## 6.1.6. 程序组织单元 (POU)

程序组织单元(Programming Organization Unit, POU)是 IEC61131-3 标准中最重要的 IEC 基本编程单元,其中包括了程序(Program)、功能块(Function Block)和功能(Function) 3 个组成元素。

程序: 是所有可编程语言元素和结构的逻辑组合,是程序组织单元的最大形式。同时程序也是 IEC 逻辑的执行主体,是和具体任务相联系的。

功能块:其是具有面向对象特性的基本算法单元,也就是说一个功能块中包含有自己私 有的变量和内存区域。例如 PID 算法功能块,其内部有用于保存当前时刻输入值和前几个 周期输入值的变量。

功能: 是一种具有过程特性的基本算法单元,即这个算法单元没有自己私有的变量和内存区域,只关注算法的实现,运算结果只和当前输入值有关。例如计算正弦的 sin 函数。

【注意】从概念上来说,功能块和功能的区别在于有没有私有的数据,反应到结果上,则是算法的结果输出是否具有确定惟一性。功能块的输出值不仅和当前的输入值有关,也和功能块内部的状态及前面保存的结果有关,所以对于不同的输入值可能会产生不同的输出结果。而对于功能来说,相同的输入值必定具有相同的输出值。

程序、功能块和功能在层次和隶属关系上也是有不同的。正如图 IEC61131-3 软件模型 结构图所展示的,程序可以调用功能或者功能块。同样功能块也可以调用功能,但是功能不 能反过来调用功能块和程序。

#### 6.1.6.1. 程序、功能块和功能的节点组成

程序、功能块和功能在工程树中的组成是相同的,都具有变量工作单、代码工作单和说明工作单,如图所示。程序节点的图标是一个绿色的矩形" I "。功能块的节点图标是一个绿色的矩形,两侧分别有两条并行的短横线,矩形的上方还有一条长横线" I ",这表明了在接口上,功能块可以有多个输入和多个输出。功能的节点图标是一个绿色矩形,左侧有两条短横线,右侧只有一条短横线"I

图 6.26", 这表明在接口上, 功能可以有多个输入但是只有一个输出。



图 6.27 POU 元素的工程树组成

代码工作单用于编写实现控制逻辑的算法,除功能不能使用顺序功能图表(SFC)外, 支持 IEC61131-3 标准定义的全部 5 种编程语言。不同的编程语言,工程树中对应的节点 图标也不相同,图中是功能块图(FBD)对应的图标。双击 POU,即可打开代码工作单,如 图所示。



图 6.28 功能块图语言的代码工作单

变量工作单展示了一个程序、功能块或者功能中使用到或者引用的变量。首先打开一个 POU 的代码工作单,然后点击变量工作单点即可打开变量工作单,如图所示,点击即可打开 程序 POU "M1"的变量工作单。

•• MULTIPROG Express - test2 - [代码:M1*]	-	-		the state	-		
● 文件(F) 編輯(E) 视图(V) 工程(P) 建立(B) 对象(Q)	布局(山)	联机(N)附加(X)	窗口(W)?				
📁 🖬 🔍 🔍 🗁 🎦 📰 🗔 🍝 🎽 int 👭	中間	HH₩	g = 1	* 🌮 🎽 🖀 🌾	s 🖉		
□-「● 库 ↓ ■ MS_PLCopen_LIB							
日本語 数据类型 一語 Axis_Type* 二語 逻辑POI							· ·
FC1*			•			AXIS1-	MC_Power_1 MC_Power Axis Axis AXIS1
	III -						Enable Status
							Error -•·
							ErrorID
		•					· ·

图 6.29 变量工作单

POU 元素的变量工作单和全局变量工作单的内容是一样的,区别在于全局变量工作单中的"用法"一列只有"VAR\_GLOBAL"一项,而这里有"VAR"和"VAR\_EXTERNAL"可选。 "VAR"代表变量为局部变量,而"VAR\_EXTERNAL"表示变量是外部的,此处只是对其的引用,一般来说使用这个用法的变量就是全局变量工作单中的变量,对其进行的操作等同于对 全局变量的操作。"VAR\_EXTERNAL\_PG"现在已停止使用。

注意:当使用"VAR"标示变量时,变量名称可以和全局变量工作单中的变量同名,但是这样表示的是两个变量,彼此没有联系。

#### 6.1.6.2. 添加删除程序、功能块和功能

添加 POU 元素的操作过程是大体类似的,下面以添加程序为例说明步骤。 右键点击"逻辑 POU"节点,在弹出的快捷菜单中选择"插入…"->"程序",如图所示。

	OG Expre	ess - test	2 - [代码	:M1*]					-
■ 文件(E)	编辑(E)	视图(⊻)	工程(2)	建立(B)	联机(N)	附加(X	) <u>2</u>		
💋 🔲 🔍	۹ 🔽	ا 🗢 🛠		s 🗧	lack ded	45 16	HI II	-k}-	<b>•</b>
□	S_PLCopen 类型 xis_Type*	LIB							
	插入 1 1 1 1 1 1 1 1 1 1 1 1 1	D (P) (C (R)	trl+V	<ul> <li>2 程序</li> <li>3 功能</li> </ul>	(P) (E) 块(B)				

图 6.30 添加程序

在弹出窗口中填入程序的名称,选择相应的编程语言,点击"确定",如图所示。

<b>5</b> 称(N):		72-
fvProg1		WHI AL
类型	语言	取消
<ul> <li>健序(G)</li> <li>功能(U)</li> <li>功能快(B)</li> <li>动作</li> <li>特換</li> <li>SFC步</li> <li>工作单</li> </ul>	<ul> <li>IL</li> <li>ST</li> <li>SFC</li> <li>FBD</li> <li>LD</li> <li>FFILD</li> <li>MSFC</li> <li>VAR</li> <li>教振奕型(Y)</li> <li>福达(C)</li> </ul>	帮助(E) □ 使用保留(R) 模式 ① 插入(1) ③ 插入(3)
返回值的数据类型(P):		0 200
		¥
PLC类型(Y):	处理器	类型(0):

图 6.31 插入程序弹出对话框

删除 POU 元素也很简单,在不需要使用的 POU 元素的节点上右击,在快捷菜单中选择"删除"即可。在执行此步骤之前需要先关闭该节点有关的工作单。

#### 6.1.6.3. 功能块的实例化

使用功能块的代码工作单编写完一段算法并在变量工作单中声明了相关变量,仅仅是对 这个功能块的一个定义,其并不能直接在控制器中运行。要使其能真正运行,就涉及到一个 实例化的概念。功能块实例化之后就产生一个功能块实例,它是这个功能块所定义的各种私 有数据、内存区域,并经过封装的集合,是一个数据实体。

简单来说,完成一个功能块相当于声明了一种变量类型,而对一个功能块进行实例化就 是生成了这种变量类型的一个实际变量。

由上面的解释可以得知,功能块的实例化是通过程序对功能块的调用实现的。在 MULTIPROG 中进行实例化功能块的步骤很简单,以 FBD 语言的程序说明,如图所示。



图 6.32 在 FBD 语言的程序代码工作单中实例化功能块

在 MULTIPROG 右侧的"编辑向导"窗口中,通过下拉框"组"选择"功能块",在任意一个功能块图标(以"TON"为例)上点住鼠标左键不放,将图标拖拽至程序的代码工作单范围内。此时松开鼠标左键,会弹出一个对话框窗口,如图所示。

安量属性		X
名称(N): TON_1	定义范围 ◎ 局部(L)	确定 取消
TON	<ul> <li>Pefault ▼</li> <li>全局变量组(A):</li> <li>□● 物理硬件</li> </ul>	帮助(H)
初值(I): I/O地址(S):	白	
描述(E):		
□ PDD □ OPC □ 隐藏(X) □ 初值作为默认值.	☑ 显示工作单的所有变量 (₩)	

## 图 6.33实例化功能块对话框窗口

在"名称"一栏可以看到有一个默认的名字"TON\_1",用于表明功能块"TON"的一个实例。 同时通过"定义范围"Tab 页中的"局部变量组(V):"可以知道,功能块实例其实质是一个变量。 注意:功能块实例的名称可以自己定义,不一定要包含有功能块名称,但是不能和功能 块重名。

点击图中所示对话框的"确定"按钮,在程序的代码工作单中就出现了功能块的一个实例,如图所示。在功能块实例"TON\_1"图标的顶部可以看到"TON"字符,这用于表示其是通过哪个功能块实例化得到的。



图 6.34 功能块"TON" 实例化完成

和功能块的情况类似,程序也只是定义了一种数据的结构和类型,同样不能在实际控制器中运行。要使其能实际运行,同样需要一个实例化的过程。

程序的实例化是和任务相关联的,而任务又是在资源中运行的,所以程序实例化的过程 可以简单理解为程序所定义的数据、算法指令、数据空间在实际的微处理器上的分配过程。

在 MULTIPROG 中对程序进行实例化就是在任务中插入程序实例。在工程树中右击需要运行某个程序的任务节点,选择快捷菜单的"插入..."->"程序实例"条目,如图所示。



图 6.35 添加程序实例

在弹出窗口里命名程序实例,并选择"程序类型",如图所示。"程序类型"就是指当前工程树中所有定义的程序节点。

ίλ	14	
程序实例(N):	类型	确定
程序类型(M):	● 配置 (C) ● 资源 (R)	取消
adder1	□ 仕分(I) ● 程序实例(P)	帮助(H)
adder1 Main	○ 描述 (D)	
	○ 受重 (V) ◎ FB实例	
	模式:	
	◎插入(I)	
□排除在编译之外(X)	◎追加(A)	

### 图 6.36程序实例化对话框

注意:程序的实例化和功能块略有不同,程序实例和程序可以重名。程序的实例是通过 任务区分的,在同一个任务中不能有同名的程序实例。

# 6.1.7. 用户自定义数据类型

IEC61131-3 标准为了方便各种不同行业和领域的编程人员使用,定义了丰富的数据类型,分为基本数据类型、一般数据类型和衍生数据类型 3 类。

# 6.1.7.1. 基本数据类型

基本数据类型是在标准中预先定义好的标准化的数据类型,包括数据关键字、数据宽度、 数据表示范围和约定的初始值,详细信息如表所示。

数据类型	关键字	数据宽度(bit)	表示范围	约定初始值
布尔	BOOL	1	0和1	0
8 位长度位串	BYTE	8	0~16#FF	
16 位长度位串	WORD	16	0~16#FFFF	
32 位长度位串	DWORD	32	0~16#FFFF_FFF	
64 位长度位串	LWORD	64	0~16#FFFFFFFF_FFFFFFF	
短整型	SINT	8	-128~+127	0
整形	INT	16	-32768~+32767	0
双整形	DINT	32	-2 <sup>31</sup> ~+2 <sup>31</sup> -1	0
长整形	LINT	64	-2 <sup>63</sup> ~+2 <sup>63</sup> -1	0
无符号短整型	USINT	8	0~+255	0
无符号整形	UINT	16	0~+65535	0
无符号双整形	UDINT	32	0~+2 <sup>32</sup> -1	0
无符号长整形	ULINT	64	0~+2 <sup>64</sup> -1	0
实数	REAL	32	1.5e <sup>-45</sup> ~3.4e <sup>38</sup>	0.0
长实数	LREAL	64	5.0e-324~1.7e308	0.0
持续时间	TIME			T#0s
日期	DAY			D#0001-01-01
时刻	TOD			TOD#00:00:00

表 3.1 基本数据类型一览表

日期和时刻	DT			DT#0001-
				0101-00:00:00
变长单字字节	STRING	80	在运行期决定	空串
字符串				

#### 6.1.7.2. 一般数据类型

一般数据类型用"ANY"进行标识,采用分层的结构。该数据类型不对用户的 POU 开放, 所以在编程时无法使用。

#### 6.1.7.3. 衍生数据类型

衍生数据类型是在基本数据类型的基础上由用户建立的数据类型,也称导出数据类型, 就是本标题中所说的用户自定义数据类型。

在 MULTIPROG 中插入用户自定义的数据类型步骤如下:工程树中右击"数据类型",选择快捷菜单的"插入…"->"数据类型"条目,如图所示。



图 6.37 插入自定义的数据类型

在弹出对话框中填入名称,点击"确定",如图所示。

<b>呂称(N):</b>		确定
Jser1 类型		取消
<ul> <li>程序(G)</li> <li>功能(U)</li> <li>功能块(B)</li> <li>动作</li> <li>转换</li> <li>SFC步</li> <li>① 工作単</li> </ul>	IL ST SFC FBD LD FFLD MSFC VAR ● 数据类型(Y) 描述(C)	<b>帮助(H)</b> <ul> <li>一使用保留(R)</li> <li>模式</li> <li>● 插入(I)</li> <li>● 追加(A)</li> </ul>
反回值的数据类型(P):		
		<b>•</b>
LC奕型(Y):	处理器组	类型(0):

图 6.38 数据类型插入对话框

注意:此处的"名称"只是数据类型工作单的名称,与数据类型没有关系。

在数据类型工作单中使用字符语言进行编辑,如下所示。

ТҮРЕ	
AI_Signal: (Single,Differ);	枚举数据类型
END_TYPE	
ТҮРЕ	
AnaSig:INT(016000);	子范围数据类型
END_TYPE	
ТҮРЕ	
AI:ARRAY[1100] OF AnaSig;	数组数据类型
END_TYPE	
ТҮРЕ	
STUDENT:	
STRUCT	
NAME: STRING;	
STU_ID: UDINT;	
CLASS: USINT;	结构体数据类型
DATE_OF_ENROLL: DATE;	
HEIGHT: REAL;	
WEIGH: REAL;	
END_STRUCT;	
END_TYPE	

图 6.37 数据类型定义工作单

按照 IEC61131-3 标准的规定,所有的衍生数据类型的定义都使用"TYPE"关键字开头,以

"END\_TYPE"关键字结束。MULTIPROG 支持 4 种基本的自定义数据类型,分别是枚举、子范围、数组和结构体,其对应的示例如上图所示。

如果这些类型不能满足要求, MULTIPROG 还支持多维数组和数组和结构体的相互嵌套定义。

# 6.2. MULTIPROG 窗口及通用操作

# 6.2.1. 主窗口

MULTIPROG 主窗口包含多个子窗口,其布局如下:



图 6.39

图 6.38 MULTIPROG 主窗口

各子窗口可以通过"视图"菜单显示和隐藏,即单击一次"视图"菜单对应的子菜单项,则对应子窗口显示,再次单击该子菜单,则对应子窗口关闭。

# 6.2.2. 工程树窗口

工程树窗口用于管理 PLC 工程的所有资源,包括所有的 POU 对象的组织方式,运动控制功能块库,数据类型,以及硬件资源,任务配置等。



图 6.40

# 6.2.3. 代码及变量窗口

每个 PLC 程序逻辑 POU 对应一个程序窗口,一个局部变量窗口;

在工程树窗口中,切换到"工程"标签页,展开逻辑 POU,鼠标双击逻辑 POU 下的 POU 项,可以打开对应的程序窗口;



图 6.41

选中 POU 项,选择主窗口菜单【视图】->【打开变量工作单】,可以打开该 POU 对应的局部变量窗口;

	名称 🛆	类型	用法	描述	地址		-
1	🗆 Default						
2	add_dec	REAL	VAR_EXT				Ξ
3	axis1	BOOL	VAR				
4	axis10	BOOL	VAR				
5	axis11	BOOL	VAR				
6	axis12	BOOL	VAR				
7	axis13	BOOL	VAR				
8	axis14	BOOL	VAR				
9	axis15	BOOL	VAR				
10	axis16	BOOL	VAR				
11	axis2	BOOL	VAR				
12	axis3	BOOL	VAR				-
13	axis4	BOOL	VAR				
14	axis5	BOOL	VAR				
15	axis6	BOOL	VAR				
16	axis7	BOOL	VAR				
17	axis8	BOOL	VAR				
18	axis9	BOOL	VAR				
19	Cyclic10_done	BOOL	VAR				
20	Cyclic11_done	BOOL	VAR				-
21	Cyclic12_done	BOOL	VAR				
22	Cyclic13_done	BOOL	VAR				
23	Cyclic14_done	BOOL	VAR				
24	Cyclic15_done	BOOL	VAR				
25	Cyclic16_done	BOOL	VAR				Ŧ
•						•	

图 6.42

一个 PLC 工程各 POU 共用一个全局变量窗口,通过选择主窗口菜单【视图】->【全局变 量声明】,可以打开全局变量窗口。

	名称	类型	用法	描述	推址	初位	保持	PDD	-
1	System Variables								=
2	PLC_SYS_TICK_CNT	UDINT	VAR_G		%MD1.0				
3	PLC_TASK_DEFINED	INT	VAR_G		%MW1.4				Î
4	PLCMODE_ON	BOOL	VAR_G	TRUE : current PLC mode is ON	%MX1.2016.0				Ť.
5	PLCMODE_LOADING	BOOL	VAR_G	TRUE : current PLC mode is LOADING	%MX1.2017.0				Î
6	PLCMODE_STOP	BOOL	VAR_G	TRUE : current PLC mode is STOP	%MX1.6.0				Ī
7	PLCMODE_RUN	BOOL	VAR_G	TRUE : current PLC mode is RUN	%MX1.7.0				I
8	PLCMODE_HALT	BOOL	VAR_G	TRUE : current PLC mode is HALT	%MX1.8.0				Ī
9	PLC_TICKS_PER_SEC	UINT	VAR_G		%MW1.2000				Ι
10	PLC_MAX_ERRORS	UDINT	VAR_G		%MD1.2004				Ī
11	PLC_ERRORS	UDINT	VAR_G		%MD1.2008				Ι
12	PLC_TASK_AVAILABLE	INT	VAR_G		%MW1.2012				I
13	PLC_SYSTASK_AVAILABLE	INT	VAR_G		%MW1.2016				Ι
14	PLCDEBUG_FORCE	BOOL	VAR_G	TRUE : current PLC mode is POWER on	%MX1.2018.0				I
15	PLCDEBUG_POWERFLOW	BOOL	VAR_G	TRUE : current PLC mode is POWER on	%MX1.2020.0				Ι
16	PLCDEBUG_BPSET	BOOL	VAR_G	TRUE : one or more	%MX1.2019.0				I
17	exe_rel	BOOL	VAR_G			1			Ι
18	reset_delay	TIME	VAR_G			TIME#0.1s			Ι
19	powerOn_delay	TIME	VAR_G			TIME#50ms			Ι
20	Distance	REAL	VAR_G			1000000.0			I
21	vel	REAL	VAR_G			500000.0			Ι
22	add_dec	REAL 🚽	VAR_G			10000.0			Ι
23	Distance1	REAL	VAR_G			-1000000.0			Ι
24	proup_pos1	GrpPosition	VAR_G			[4(-1000000.0)]			Ι
25	proup_pos	GrpPosition	VAR_G			[4(100000.0)]			Ι
26	jerk	REAL	VAR_G			50000.0			I
27	Cyclic2_done	BOOL	VAR_G						Ι
28	Cyclic3_done	BOOL	VAR_G						I
29	Cyclic4_done	BOOL	VAR_G						Ī
30	Cyclic5 done	BOOL	VAR G						<b>–</b>
I <								•	

图 6.43

# 6.2.4. 消息窗口

显示所有编程调试阶段的输出信息。



图 6.44

## 6.2.5. 编辑向导窗口

用于向程序中添加功能块,分类列出了所有功能块,用鼠标按住左键将其拖动到程序窗 口,即可完成向程序添加对应的功能块。

:编辑向导		4	▼ 🛛
组:			
<全部FU和FB>			-
名称	描述		
= LREAL_TO_UDINT	将LREAL转换至UDINT		
= LREAL_TO_UINT	将LREAL转换至UINT		
LREAL_TO_USINT	将LREAL转换至USINT		
= LREAL_TO_WORD	将LREAL转换至WORD		
-∎-LT	小于: <		
The string	小于STRING: <		
= MAX	最大值		
- MAX_DINT	DINT型数据的最大值		
= MAX_INT	INT型数据的最大值		
= MAX_LINT			
= MAX_LREAL	LREAL型数据的最大值		
= MAX_REAL	REAL型数据的最大值		=
= MAX_SINT	SINT型数据的最大值		
= MAX_STRING	字符串的最大值		
The MC_AbortTrigger			
≢ MC_GrpDisable			
≢ MC_GrpEnable			
≢ MC_GrpImmediateStop			
= MC_GrpReadActPos			
= MC_GrpReset			
The MC_GrpSetOverride			
= MC_GrpStop			
# MC_GrpSyncMoveAbs			
= MC_Home			
MC_ImmediateStop			
# MC_MoveAbsolute			
MC_MoveCircAbs			Ŧ

图 6.45

# 6.2.6. 子窗口的模式

对于带有红色 X 和蓝色三角符号的子窗口,可以停靠在主界面的窗口内,如下图样式:

		axisI_2	-
		axis1_3	
		axis4group5_8	
	1	axis4group1	
		axis4group1_4	
		axis1_4	
		axis1_8	
		A8	
		axis2group9_16	
		Main_Test17_32	
		axis2group1_8	
		axis1_16	
		正在生成IBC代码	
		正在收集被资源/ 资源/ 所使用的POU···	
		正在为资源"资源"生成IEC代码···	
		正在为资源"资源"创建实例树	
		正在生成专用目标代码,用于配置 配置	=
		正在生成专用目标代码,用于资源 资源	
日初	8	0个错误, 1个警告	
100			Ŧ
3		III (建立 Λ 错误 Λ 警告 Λ 信息 Λ PLC出错 Λ 打印 /	

#### 图 6.46

## 图 6.47 MULTIPROG停靠式子窗口

也可以通过鼠标在子窗口带红色 X 按钮的标题栏双击切换到浮动窗口,或者按住鼠标左 键拖动,将其改变为浮动窗口,浮动窗口样式如下图所示:

消息窗口	
axis1_2	*
axis1_3	
axis4group5_8	
axis4group1	
axis4group1_4	
axis1_4	
axis1_8	
A8	
axis2group9_16	
Main_Test17_32	
axis2group1_8	=
axis1_16	
正在生成IBC代码	
正在收集被资源'资源'所使用的POU···	<b>•</b>
▲▶\建立 〈 错误 〉 警告 〉 信息 〉 PLC出错 〉 打印 /	

### 图 6.48 MULTIPROG浮动式子窗口

再次在浮动窗口标题栏双击或者按住鼠标左键拖动窗口到主窗口边沿,可以将其改变为 停靠子窗口。

拖动过程中,若系统判断到合适的停靠位置,则会自动停靠。若要避免自动停靠,在拖动过程中按住"CTRL"键。

# 6.2.7. 子窗口的大小调整

对于子窗口,无论是浮动状态还是停靠状态,均可以将鼠标移动到窗口边沿,按住鼠标 左键拖动以调整窗口到任意大小。

也可以鼠标单击子窗口标题栏中三角形按钮,将该子窗口在子窗口区域最大化和各子窗 口大小均匀分布状态之间切换。

子窗口最大化:



图 6.50

## 6.3. 新建工程

新建一个工程,可以通过工程向导或者一个工程模板来进行;

相对于工程向导创建工程来说,工程模板创建的步骤更简单,且创建后包含了工程向导 创建后需要手动添加的内容。

用工程向导创建工程后,必须手工配置以下内容,而用工程模板创建工程后这些内容已 经配置好了。

### ● 删除缺省的IO配置;

- 功能块库的导入;
- 功能块所引用的数据结构, 如轴的数据结构定义;

#### 6.3.1. . 使用工程向导创建和配置新工程

#### 6.3.1.1. 创建工程

点击 MULTIPROG 主界面中选择菜单"文件"->"新建工程...",打开新建工程对话框, 在对话框模板区中选择"工程向导"选项,如图所示,点击"确定"进入下一步。



图 6.51 新建工程对话框

以下描述工程向导创建工程的过程。

在向导窗口的"工程名称"框中输入工程名称,例如"My\_first\_Project",如图所示。根据工程的命名规则,工程的名称和路径一定不能含有空格或特殊字符,"工程路径"输入框指明了工程保存的路径,初始状态下为默认路径,用户可以自行指定。

完成后点击"确定"按钮。



工程向导(第1步,共6步)	Image: State of the state of t
Project  Cubranies  Cubranies  Cubranies  Cubranies  Cubranies  Cubranie  C	工程向导将帮助您创建新的工程。 你可以在任何时候按'后退'来更改所作的选择。 工程名称 ON): My_first_Project 工程路径 (P): C:\Users\Public\Documents\MULTIPROG\Projects
	< 上一步 (B) 下一步 (A) > 取消 帮助 帮助

#### 图 6.52 设置工程名称和保存路径

工程向导第二步对话框如图所示。将第一个 POU 取名为"Main",编程语言选择"梯形图 (LD)",或者根据自己熟悉的编程语言选择一种。

点击"下一步"。

🕞 😋 Project	请选择初始程序POU(程序组织单元)的名称和语言。
Libraries	POU名称(M):
E	Main
E Physical Hardware	语言
🗄 📾 MyConfiguration	◎ 指令表 (IL)
i Mynesource i ∭ Tasks	◎ 结构化文本 (ST)
Globals	◎ 顺序功能图 (SFC)
IU_Config	○ 功能块图 (PBD)
12	
	◎ 四定省式物形图 (1110)

### 图 6.53设定POU名称和编程语言

工程向导第三步用于确定配置的名称和类型,对话框如图所示。在"名称"输入框中填入 配置的名称,这里保持默认的"配置"。在"类型"列表框中选择 PLC 类型为"eCLR",点击"下一 步"。 【注意】该步骤只在 Professional 版本中配置, Express 版本无该步骤。

程向导(第3步,共6步)	<u> </u>
□	置的名称和类型。 所连接PLC的特性。
Logical POUs 一	M): 配置 T): @CLR
	(〈 上一步 03 ) [下一步 03 ) 〉 取消 】 「 帮助

图 6.54 确定配置的名称和类型

工程向导第四步用于选择所要使用的资源,对话框如图所示。"名称"保持默认的"资 源","类型"列表框中选择与控制器硬件相对应的类型,点击"下一步"继续。

【注意】不同的类型选择代表不同的硬件平台,因为在工程制作时,系统会根据资源的 类型生成只能在特定硬件下运行的代码。

工程向导(第4步,共6步)		8
Project Libraries Data Types Logical POUs B-O MyProgram Physical Hardware D-MyConfiguration D-MyConfiguration Tasks Globals ID_Config	请选择资源名称 资源描述了PLC 资源 名称 (0): 类型 (7):	和资源类型。 处理器类型的特性。 资源 [Leetro_I5_1213 、
		< 上一步 (B) 下一步 (B) > 取消 帮助

图 6.55 确定资源的名称和类型

向导第五步用于指定任务的名称和类型,对话框如图所示。这里仍旧保持默认的名称"任务",类型选择为"CYCLIC",点击"下一步"。有关任务类型的具体含义将在下一章说明。

Project     Libraries     Data Tunes	请选择任务的:	名称和类型,你的预定义POU将以这种任务类型运行。
Logical POUs B-O MyPogram Prysical Hardware MyConfiguration Tasks Globals IO_Config	任务 名称(M): 类型(T):	任务 [CYCLIC
		< 上一步 (0) ) 下一步 (0) > □ 取消 □ 帮助

图 6.56 任务名称和类型选择

在最后一步,向导的弹出对话框对前面的设置进行了小结,如图所示。

工程向导(第6步,共6步)		<b>X</b>
Project     Libraries     Data Types	工程描述 工程名称: 工程路径:	My_first_Project C:\Users\Public\Documents\MULTIPROG\Projec
Logical PUUs     Logical PuUs     L MyProgram     Physical Hardware     MyConfiguration     MyResource	POV名称: POV语言: 配置名称:	Main 秭形图 (UD) 配罢
Globals	PIC类型: 资源名称: 处理器类型:	eUR 资源 Leetro_I5_1213
	任务名称: 任务类型:	任务 CYCLIC
		< 上一步 (B) 完成 取消 帮助 (100 年) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (100 #) (1

图 6.57 工程向导小结对话框

如果没有错误,点击"完成",弹出如下警告信息对话框:

MULTIPRO	DG Express
<u> </u>	该工程可能含有其它语言(远东)的字符。这些字符会导致错误。您还要打开这个工程吗?
	□对于这个工程,不要再问了(A)。
	是(Y) 否(N)

图 6.58

忽略该警告信息,点击"是"。

就可以在工程树窗口中看到新生成的工程树。如图所示。其中的"工程"页中"逻辑 POU" 节点是有关于算法实现的部分,而"硬件"页中"物理硬件"则是和实际的控制器的类型和设置 相关联。

□	
_ <∲ 工程 _ 硬件	

图 6.59新生成的工程树-工程

● ● 物理硬件 ● ● 配置 : eCLR ● ● 资源 : Leetro_I5_1213* ● ● ● 任务 : CYCLIC ● ● 日务 : main Global_Variables ■ IO_Configuration*
(学工程

图 6.60新生成的工程树-硬件

## 6.3.1.2. 导入运动控制功能块固件库

使用工程向导创建的工程,若要在程序中使用运动控制功能块库,还必须进行如下操作,导入包含运动控制功能块的固件库。否则,只能使用 IO 从站和对应的逻辑控制;

- 导入运动控制功能固件库的步骤如下:
- 1) 打开"工程树窗口"。



# 图 6.61

2) 鼠标右键单击【库】。在弹出菜单中选择【插入】->【固件库】。弹出如下对话框:

🔹 包含库			×
查找范围(I):	🕌 FW_LIB 👻 🥝 (	• 🖅 对	
<u>(</u>	名称	修改日期	1
最近访问的位置	BIT_UTIL CIFSYS	2018/8/20 16:48	2
	MS_PLCopen_LIB	2018/8/20 16:48	
桌面		2018/8/20 16:48	1
<mark>库</mark>	- Ficcilos	2010/0/20 10.40	1
<b>《</b> 】 计算机			
<b>(</b> ) 网络			
		- (m+ii 4	*
	文件4 (0). 文件类型(T): 固件库 (*.fw1)	· 回摘(	i

图 6.62

3) 鼠标双击【MS\_PLCopen\_LIB】打开该文件夹,单击【包括】按钮;即可将固件库导入到 工程中。



图 6.63

### 6.3.1.3. 添加数据类型

运动控制功能块会用到一些参数,是用 IEC61131-3 的数据结构定义的,要在程序中引用 轴,则需要定义数据结构,以下描述如何自行添加数据结构:

 在工程树窗口中,鼠标右键单击【数据类型】,在弹出的菜单中选择【插入】->【数据 类型】。



图 6.64

2) 在【名称】栏输入包含数据类型的工作单名称,如"MyData",单击【确定】按钮,即可添加一个空的工作单,用于输入数据结构的定义。

盾入 名称(N):		<mark>■×</mark> 确定
MyData 类型	 语言	取消
<ul> <li>○程序(G)</li> <li>○功能(U)</li> <li>○功能抉(B)</li> </ul>	IL ST SFC	帮助(H)
<ul> <li>→ 加加</li> <li>→ 加作</li> <li>→ 转換</li> <li>→ 500 μ</li> </ul>	FBD LD FFLD	□使用保留(R)
◎ 工作单	● <sup>VAR</sup> ● <b>数据类型(Y)</b> ● 描述(C)	<b>模式</b> ◎ 插入(I) ◎ 追加(A)
返回值的数据类型(P):		
		V
PLC类型(Y):	处理器类	类型(0):
<独立>	☞ (<独立>	~

图 6.65

3) 输入数据结构定义 正确输入描述轴等数据结构的代码,即可在编程中引用该数据结构。 重新打开工程时,会自动在程序结构中添加系统默认的数据结构。

## 6.3.2. 使用工程模版创建新工程

点击 MULTIPROG 主界面中选择菜单【文件】文件->【新建工程...】,打开新建工程对 话框,在对话框模板区中选择"工程模版"选项,如图所示,点击【确定】即可完成工程创建。

常规 Leetro	确定
※ 正程向号 I5_1213工 I5_1215工 MPC_8801工 提	取消

图 6.66新建工程对话框

用工程模板创建的工程,已经包含了 6.3.1 描述的所有内容。

创建成功后,建议先在 MULTIPROG 主界面中选择【文件】->【工程另存为】菜单,将工程进行保存。

若没有进行保存,当打开另一个工程,或者退出 MULTIPROG 时,会提示保存:



图 6.67

单击【是】按钮,选择一个保存文件夹,和保存的工程名称,即可将模板生成的工程进行保存。



# 6.4. 添加从站

### 6.4.1. 导入 ESI 文件

ESI 文件由 EtherCAT 从站供应商提供,每个不同型号的从站均应当有对应的 ESI 文件。 MultiProg 必须导入 ESI 文件,才能自动扫描识别 EtherCAT 总线上连接的从站;或者手动 添加对应的从站到设备树;

ESI 导入方法如下:

先打开"视图"->"Ethercat",然后鼠标点击选中 Ethercat 窗口, 配置 ESI 的"Ethercat" 选项才会在菜单栏中出现。



图 6.68 1) 启动 MultiProg,单击菜单"EtherCAT"->"ESI Manager",打开如下对话框:

e 🖉	SI N	Mana	ger							
ESI	File	s								
S	elec	t an	ESI file which should be de	leted or expo	rted or add new ESI files	5.				
	۱.	۵	Baumueller Nuernberg Gmb	umueller Nuernberg GmbH						
	×.	BECK	Beckhoff Automation GmbH	1 & Co. KG						
	×	<b>@</b>	Copley Controls Corp.							
	×	L	Leetro Auto Corp.							
	×	Р	Panasonic Corporation, App	liances Compa	ny					
	×	UIPA	VIPA GmbH	VIPA GmbH						
	►	7	Yaskawa Electric Corporation							
						Number of ESI files: 80				
						Number of devices: 2041				
				<b>D</b> 1 -						
	A	dd F	ie Add Folder	Delete	Export	Close				

图 6.69

2) 在 ESI Manager 对话框中,单击"Add File",在弹出的打开文件对话框中定位到要添加的 ESI 文件所在的文件夹,并选中要添加的 ESI 文件,单击"打开"按钮。

BIQ · 新建文件央						18 ·		
第7 🏭		名称	*	修改日期	关型	大小		
國 桌面	100	Reckhol	F EK'yyyyy yml	2015/11/18 22-54	XMI WH	175 88		
3 最近访问的位置		Beckhol	FEI 1 yyy yml	2017/1/12 16:17	YMI OIE	2 627 KB		
		Beckhof	f EL2xxx.xml	2017/1/25 15:56	XML 文件	2.656 KB		
二 年		Beckhof	f ELSxxx.xml	2015/9/14 22:54	XML 文件	5.054 KB		
PPTV825	E	DMD57	OVE V1.0.xml	2019/1/2 14:10	XML文件	178 KB		
Subversion		E20H1 V	1.0.5.xml	2018/11/20 17:27	XML文件	115 KB		
H 1050		E20P1 V	1.5.0.xml	2018/11/5 13:14	XML文件	110 KB		
三原比		E28H1 V	/1.0.3.xml	2018/10/16 13:56	XML文件	120 KB		
		Leetro A	A1E_Ver1.6.aml	2019/3/5 9:35	XML 文件	121 KB		
h est		Panasor	ic_MINAS-A5B_V0_22.xml	2018/10/24 9:05	XML文件	1,689 KB		
		Panasor	ic_MINAS-A6BE_V1_3.xml	2018/10/24 9:05	XML文件	661 KB		
₩ 计算机		Panasor	ic_MINAS-A68F_V1_3.xml	2018/10/24 9:05	XML文件	679 KB		
System (C:)								
👝 Software (D:)	*							
\$488(N): 1ee	rt 6 vml				Supported Files (* ym)	rin)		

图 6.70

3) 添加文件进度条显示添加进度,完成后会在 ESI manager 窗口中显示新添加的设备列表 信息。可单击列表栏每行公司名称左侧的三角符号展开列表,在列表下一级显示具体的 ESI 文件列表项;继续单击 ESI 文件列表项左侧的三角符号,则展开 ESI 文件中描述的设 备名,设备描述信息以及版本信息。



图 6.71

# 6.4.2. 配置 MULTIPROG 与 i5 的连接

安装 MULTIPROG 的 PC 与 i5 通过网络连接,必须在 MULTIPROG 工程中配置要连接的 i5 的 IP 地址,才能通过 MULTIPROG 与 i5 连接实现自动扫描从站操作。

有自动配置和手动配置方法两种方法。

#### 6.4.2.1. 自动配置

选择菜单【联机】->【扫描控制器】,在弹出的窗口中双击需要被连接的控制器。



图 6.72

#### 6.4.2.2. 手动配置

- 1) 启动 MULTIPROG。在主界面切换到硬件标签页。依次展开"物理硬件", "配置", "资源"。
- 2) 右键单击"资源",在弹出菜单中选择"设置..."。



图 6.73

3) 在"资源设置"窗口中,修改"通信"-> "类型"为"TCP/IP"。在"通信"-> "参数" 对应的文本框中输入 i5 的 IP 地址, IP 地址后面空格后-p41100 保持不变。

eetro_I5_1213#	的资源设置
通信	
类型:	TCP/IP 🔹
参数:	192.168.1.88 -p41100
版本	
建立设置:	eCLR EtherCAT V2.0.0
更新建立设击 ◎ 自动更 ◎ 更新前 ◎ 不更新	Build settings}f;73: 新(A) 询问(B) (N)
在线更新 时间间隔:	10 毫秒 范围: 0.60000)
编译器选项 □ 堆栈检 ☑ 数组边 □ 经过优	查 界检查 化的发展码
	确定(0) 取消(C) 帮助(H)

图 6.74

单击"确定"按钮关闭对话框,即完成连接设置。

若配置正确,连接成功,则通过主界面菜单"联机"->"工程控制..."打开工程控制面板显示如下左边状态;若配置的连接参数与 i5 参数不一致,则出现如下右侧状态,即处于通信连接超时状态,此时无法进行联机操作。

资源 🔤	- • 💌
状态: 运行	
停止(S)	冷启(0)
复位(R)	(₩)
	热启(T)
下装(D)	
更多(M)	信息()
关闭(C)	帮助(H)

(U) (() (H)

图 6.75

资源	- • 💌
状态: 超时!	
停止(S)	冷启(0)
复位(R)	暖启(W)
错误(E)	热启(T)
下装(D)	
更多(M)	信息の
关闭(C)	帮助(H)

图 6.76



# 6.4.3. 自动扫描从站

将要添加的从站设备通过网线连接到主站,并将主站和从站上电进入运行状态。在导入 从站设备的 ESI 文件后,即可按照如下步骤扫描添加从站信息到 MultiProg 的设备树。

1) 启动 MultiProg, 单击菜单"视图"->"EtherCAT", 打开如下对话框:

EtherCAT Configurator			
Project Explorer	Device Editor		<b>_</b>
⊎ 配置/资源	Master		
	General		
	Unit Name	配置/资源	
	Cycle Time [us]	1000 -	
	Source MAC address	00-0A-35-11-22-33	
			H .

图 6.78

2) 右键单击"配置/资源",在弹出的如下菜单中单击"Scan EtherCAT Network"。



进度条显示完成后,窗口左侧设备树中显示扫描到的设备。
图 6.80

# 6.4.4. 手动添加从站

在要添加的从站设备没有连接到主站的情况下,在导入从站设备的 ESI 文件后,也可以 手动添加从站。可按照如下步骤手动添加从站信息到 MultiProg 的设备树。

```
1) 启动 MultiProg, 单击菜单"视图"->"EtherCAT", 打开如下对话框:
```

EtherCAT Configurator			×
Project Explorer	Device Editor		
Project Explorer 业 配置/资源	Device Editor Master General Unit Name Cycle Time [us] Source MAC address	配置/资源 1000 00-0A-35-11-22-33	L. L

图 6.81

2) 若右键单击"配置/资源",在弹出的如下菜单中单击"Insert Slave"。则将在所有设备列 表的第一个位置插入新的设备。



### 图 6.82

3) 若右键单击"配置/资源"下设备树中的选中设备,在弹出的如下菜单中单击"Insert Slave After"。则将在选中设备的后面位置插入新的设备。



图 6.83

4) 若右键单击"配置/资源"下的的最后一个设备,在弹出的如下菜单中单击"Append Slave"。 则将在设备列表的最后位置插入新的设备。



图 6.84

在单击菜单后,弹出以下对话框,在"Slaves"列表中会列出已经通过 ESI 文件导入的所有 设备。选择要添加的设备,并在"Number of Slaves"栏输入设备数量,单击"OK",则将添加 指定数量的从站设备到设备树。

Д Ins	ert Eth	nerCAT Slave aft	ter '配置/资源'		
Filter					
Sea	rch				
Ver	ndors		(ALL VENDORS)	•	
Sho	w Hid	den Slaves			
Conn	ection	1			
Cor	nnect a	at	Port B, MII	-	
Slava	-				
Sele	∘ ectas	pecific slave fro	m the list and adjust the numb	er of slaves.	
•	٩	Baumueller Nue	ernberg GmbH		
•	BECK	Beckhoff Autom	nation GmbH & Co. KG		
•	((	Copley Controls	s Corp.		
•	L	Leetro Auto Cor	rp.		
•	Р	Panasonic Corp	oration, Appliances Company		
•	UIPA	VIPA GmbH			
•	7	Yaskawa Electri	c Corporation		
Nur	mber o	of Slaves	1 💌	OK	Cancel

图 6.85

# 6.5. EtherCAT 相关参数配置

在通过自动扫描或者手动添加设备到设备树后,则可按照以下描述配置 EtherCAT 相关参数。

# 6.5.1. 从站公共参数配置

启动 MultiProg, 单击菜单"视图"->"EtherCAT", 打开如下对话框:

EtherCAT Configurator			8
Project Explorer	Device Editor		
▼ 😃 配置/资源	Master Process Data Image	Advanced Options Distributed Clocks	
Slave_1001 [Leetro A1E] (1001) Slave_1002 [Leetro A1E] (1002)	General		
	Unit Name	配置/资源	
	Cycle Time [us]	-	
	Source MAC address	00-0A-35-11-22-33	
		=	
		L	
( <u>L</u>		1	

图 6.86

在"EtherCAT"对话框的设备树中单击选中"配置/资源",在右侧各个标签页对应所有 从站的公共参数配置。

#### 6.5.1.1. 控制周期

单击"Master"标签,切换到该页,通过"Cycle Time[us]"下拉列表可选择控制周期, 该控制周期也是 EtherCAT 通讯周期,以微妙为单位。

i5-1213 支持的最小控制周期为 500 微妙,常用值为 1000 微妙。

当控制从站数目越多,该周期应当设置越大,避免在设定的控制周期内无法完成通讯。

#### 6.5.1.2. DC 模式设置

单击 "Distributed Clocks"标签,切换到该页, i5 只支持 "Bus Shift"模式,因此务必使 "Clock Adjustment"栏的 "Bus Shift"单选按钮被选中。

EtherCAT Configurator		×
Project Explorer	Device Editor	
▼ U 配置/资源 U LT1001 (1001) U LT1002 (1002)	Master         Process Data Image         Advanced Options         Distributed Clocks           Reference Clock         Introduction         Introduction         Introduction           Name         LT1001 (1001)         Introduction         Introduction	
	Clock Adjustment Master Shift (EtherCAT Master Time controlled by Reference Clock)	
	Bus Shift (Keterence Clock controlled by EtherCA1 Master Time)     External Mode (Reference Clock controlled by External Sync Device)	
	Options	
	Sync Window Monitoring	
Classic View Flat View		•

图 6.87

### 6.5.2. 伺服从站参数配置

### 6.5.2.1. 从站名称和地址

在"EtherCAT"对话框的设备树中选中要配置的从站,在右侧各个标签页对应该设备的 属性,其中"General"标签页显示和设置从站的地址,名称。名称的修改将影响到与 PLC 变量 映射后的变量名,其将作为变量名的一部分。

可通过"Information"栏的"Name"项的文本框修改从站名称。

"Address"栏的"Station Address"项为从站的地址。一般采用扫描后的缺省地址即可。



缺省的设备名称通常比较长,建议修改成一个短名称, 以避免在映射到变量后导致变量名过长,而被系统自动 截断变量名称,使得变量名的可读性变差。

EtherCAT Configurator		
▼ 및 配置/资源	General PDO Mapping Variable	Advanced Options Distributed Clock Init Commands CoE Object-Dictionary
<ul> <li>Slave_1001 [Leetro A1E] (100)</li> <li>Slave_1002 [Leetro A1E] (100)</li> </ul>	Address Station Address	1001 🚍
	Name	Slave_1001 [Leetro A1E]
	Description	Leetro CiA402 Drive
	Vendor	Leetro Automation Corp., Ltd (0x000005E7)
	Product Code	0x12343052 (305410130)
	Revision Number	0x10101070 (269488240)
	ESI File	C:\ProgramData\PHOENIX CONTACT Software\MULTIPROG Express\5_50_10209 \Leetro_A1E_Ver1.4_20180719.xml
	Identification Value	Not Used
	Ports	
	A	配置/资源
	D	Not Available
	в	Slave_1002 [Leetro A1E] (1002)
	c (	Not Available

图 6.88

### 6.5.2.2. PDO 参数

PDO 参数用于 i5 对从站进行周期性控制,基于运动控制和 IO 控制的需要,必须进行 PDO 配置,并将 PDO 参数与全局变量进行映射。

对于 IO 模块, PDO 参数为数字量或者模拟量输入/输出。

映射到全局变量后,PLC程序可通过映射后的全局变量访问实现数字量或者模拟量输入输出的控制。

therCAT Configurator								
Project Explorer	Device Ec	ditor	_					
▼ 』 配置/资源	Genera	PDO Mapping	Variables	Advanced Options	Distribu	ited Clock	Init Commands	CoE Obj
Slave_1010 [E20P1] (1010)	Inputs				Output	ts		
Slave_1011 [E20H1] (1011)	•	Digital Input1	L	0x1A00	•	🔽 Digita	al Output1	
		Name 1	index	Bit Length		Name	Index	Bit Le
		DIO	x6000:01	1	11	DO0	0x7000:01	1
		DI1 0	x6000:02	1		DO1	0x7000:02	1
		DI2 0	x6000:03	1		DO2	0x7000:03	1
		DI3 0	x6000:04	1		DO3	0x7000:04	1
		DI4 0	x6000:05	1		DO4	0x7000:05	1
		DI5 0	x6000:06	1		DO5	0x7000:06	1
		DI6 0	x6000:07	1		DO6	0x7000:07	1
		DI7 0	x6000:08	1		DO7	0x7000:08	1
	-	Analog Input	1	0x1A01	•	📝 Analo	g Output1	
		Name 1	index	Bit Length		Name	Index	Bit Le
		AIO 0	x6020:01	16		AO0	0x7010:01	16
		AI1 0	x6020:02	16		A01	0x7010:02	16
		AI2 0	x6020:03	16		AO2	0x7010:03	16
		AI3 0	x6020:04	16		AO3	0x7010:04	16
< <u> </u>			bbA	Delete	dit	Up	Down	Load PDC
4	11		7 tota	20.000				+

图 6.89

对于伺服, PDO 参数为 Cia 402 标准定义的可配置为 PDO 参数的 COE 参数数据对象。



i5 要实现运动控制功能,则带轴功能的从站必须能够配置以下 COE 参数为 PDO。添加从站后,若缺省 PDO 参数列表中缺少下表中一个或多个参数,则需要通过下一节PDO 配置描述的方法将下表中的 PDO 参数配置完整。

PDO 类别	Index	Subindex	英文名称	中文名称
TPDO	0x6041	0	Status Word	状态字
	0x6064	0	Actual Position	实际位置
	0x606C	0	Actual Velocity	实际速度
	0x6061	0	Mode of Operation Display	显示当前工作模式
	0x603F	0	Error Code	错误代码
	0x60FD	0	Digit Inputs	数字量输入状态
RPDO	0x6040	0	Control Word	控制字
	0x607A	0	Target Position	目标位置
	0x60FF	0	Target Velocity	目标速度
	0x6060	0	Mode of Operation	设置工作模式

#### 6.5.2.3. PDO 配置

"PDO Mapping"标签页的内容为 PDO 参数的显示和配置修改。

置/资源	Genera	PDO Mapping	Variables Ad	vanced Options	Distribut	ed Cloo	k Init Commands	CoE Object-Di	ctionary
Slave_1001 [Leetro A1E	Inputs					Outpu	s		
Slave_1002 [Leetro A1E	-	Ist Transmit PD	O mapping		0x1A00	•	Ist Receive PDC	O mapping	0x1600
		Name	Index	Bit Length			Name	Index	Bit Length
		Status Word	0x6041:00	16			Control Word	0x6040:00	16
		ActualPosition	0x6064:00	32			TargetPosition	0x607A:00	32
		ActualVelocity	0x606C:00	32			TargetVelocity	0x60FF:00	32
		ModeOfOperationE	0x6061:00	8			ModeOfOperation	0x6060:00	8
		Dummy Byte2	0x37FF:00	8			Dummy Byte1	0x37FE:00	8
	•	2nd Transmit Pl	DO mapping (e	cluded by 0x1A0	0x1A01		ErrorCode	0x603F:00	16
		Name	Index	Bit Length			DigitInputs	0x60FD:00	32
		Status Word	0x6041:00	16		-	2nd Receive PD	O mapping (exclu	uded by 0x1600) 0x1601
		ActualPosition	0x6064:00	32			Name	Index	Bit Length
	•	3rd Transmit PE	O mapping (ex	cluded by 0x1A00	0x1A02		Control Word	0x6040:00	16
		Name	Index	Bit Length			TargetPosition	0x607A:00	32
		Status Word	0x6041:00	16		-	3rd Receive PD	O mapping (exclu	ided by 0x1600) 0x1602
		ActualPosition	0x6064:00	32			Name	Index	Bit Length
							Control Word	0x6040:00	16
							TargetVelocity	0x60FF:00	32
Ⅲ Þ iew Flat View				Add Delete	e Eo	dit	Up Down	n	Load PDO information

图 6.90

Inputs 栏为 TPDO,即从站发给主站的 PDO,也就是 i5 主站要从从站读取的状态类数据 对象;

Outputs 栏为 RPDO,即主站发给从站的 PDO,也就是 i5 主站要发给从站的命令类数据 对象;

Inputs 栏列出的标有 0x1A00, 0x1A01, ... 的各栏为从站 ESI 文件预定义的分组 TPDO 数 据对象,可对其进行如下操作:

- 1) 鼠标单击各栏复选框,选中栏的数据对象将进入 I 变量区,由系统自动统一编址。
- 2) 鼠标单击选中各复选框栏的标题,即复选框标题文件背景加深状态,此时单击底部的 Edit 可打开如下对话框:在该对话框可对 TPDO 对象进行增加,删除,编辑修改,以及 改变在 I 变量中的排列顺序等操作;修改完成后单击"OK"使修改生效并关闭对话框;单 击"Cancel"放弃修改并关闭对话框;

【注意】若从站 ESI 文件定义中不允许对 PDO 配置进行修改,则该对话框底部的操作按钮将变灰无效。

					×			
General			Optional					
Name	TPDO1			Exclude:				
Index	0x1A00		Dec Hex	✓ 1A01 ✓ 1A01				
Flags	Di							
Mandatory	(							
Fixed Content	: 0	🔵 RxPdo						
Entries		Index	Bit Lenath	Comment	*			
Entries Name Status Word		Index 0x6041:00	Bit Length 16	Comment object 0x604:	*			
Entries Name Status Word ActualPosition		Index 0x6041:00 0x6064:00	Bit Length 16 32	Comment object 0x604: object 0x6064	*			
Entries Name Status Word ActualPosition ActualVelocity		Index 0x6041:00 0x6064:00 0x606C:00	Bit Length 16 32 32	Comment object 0x604: object 0x6064 object 0x6064	*			
Entries Name Status Word ActualPosition ActualVelocity ModeOfOperationD	isplay	Index 0x6041:00 0x6064:00 0x606C:00 0x6061:00	Bit Length           16           32           32           8	Comment object 0x604: object 0x6064 object 0x6066 object 0x6065	* III *			
Entries Name Status Word ActualPosition ActualVelocity ModeOfOperationD Add	<b>Delete</b>	Index 0x6041:00 0x6064:00 0x606C:00 0x6061:00 Edit	Bit Length 16 32 32 8 Up	Comment object 0x604: object 0x6064 object 0x6066 object 0x6065 Down	* III *			

图 6.91

Outputs 栏列出的标有 0x1600, 0x1601, ... 的各栏为从站 ESI 文件预定义的分组 RPDO 数据对象,可对其进行如下操作:

- 1) 鼠标单击各栏复选框,选中栏的数据对象将进入Q变量区,由系统自动统一编址。
- 2) 鼠标单击选中各复选框栏的标题,即复选框标题文件背景加深状态,此时单击底部的 Edit 可打开如下对话框:在该对话框可对 RPDO 对象进行增加,删除,编辑修改,以及 改变在 Q 变量中的排列顺序等操作;修改完成后单击"OK"使修改生效并关闭对话框; 单击"Cancel"放弃修改并关闭对话框;



🟓 Edit PDO					23		
General			Optional				
Name	RPDO1		Exclude:				
Index	0x1600		Dec Hex	✓ 1601 ✓ 1602			
Flags	Dii	rection					
Mandatory							
Fixed Content	. (	🕽 RxPdo					
Virtual PDO							
Entries Name		Index	Bit Length	Comment	*		
Entries Name Control Word		Index 0x6040:00	Bit Length 16	Comment object 0x604(	*		
Entries Name Control Word TargetPosition		Index 0x6040:00 0x607A:00	Bit Length 16 32	Comment object 0x604( object 0x607,	- m		
Entries Name Control Word TargetPosition TargetVelocity		Index 0x6040:00 0x607A:00 0x60FF:00	Bit Length 16 32 32	Comment object 0x604( object 0x607, object 0x60Ff	- m		
Entries Name Control Word TargetPosition TargetVelocity ModeOfOperation		Index 0x6040:00 0x607A:00 0x60FF:00 0x6060:00	Bit Length 16 32 32 8	Comment object 0x604( object 0x607, object 0x60Ff object 0x606(	* H		
Entries Name Control Word TargetPosition TargetVelocity ModeOfOperation Add	Delete	Index 0x6040:00 0x607A:00 0x60FF:00 0x6060:00 Edit	Bit Length 16 32 32 8 Up	Comment object 0x604( object 0x607, object 0x60Ff object 0x606( Down	* H +		

图 6.93

"Variables"标签页的内容为左侧选中从站的全部 RPDO 和 TPDO 参数的列表。通过该 页可以进行以下操作:

- 1) 对 PDO 的名称进行修改,名称的长短将影响到 PDO 映射到 PLC 变量的名称,若名称过程,导致映射到 PLC 变量的名称过长,则 PLC 变量名称将会被截断,使得可读性变差;
- 2) 对排列顺序进行修改,排列顺序会作为与 PLC 变量映射后的 I 变量或者 Q 变量中的变量 顺序;
- 3) 增加 PDO 对象;
- 4) 删除 PDO 对象;

EtherCAT	Configurator											
Project	t Explorer	Device Edi	tor									-
▼ <u>↓</u>	配置/资源	General	PDO Mapping	Variables	Advanced Options	Distributed Clock	Init Comma	ands Co	E Objec	t-Dictio	onary	
	LT1001 (1001)	Variables	;									
	₫ L11002 (1002)	I	lame			Datatype	Group Info	Offset	*	Size		
		L	.T1001.TPDO1.Stat	tus Word		UINT	[Default]	IN :	0.0	2.0		
		L	.T1001.TPDO1.Act	ualPosition		DINT	[Default]	IN :	2.0	4.0		
		L	.T1001.TPDO1.Act	ualVelocity		DINT	[Default]	IN :	6.0	4.0		
		L	.T1001.TPDO1.Mo	deOfOperati	onDisplay	SINT	[Default]	IN :	10.0	1.0		
		L	.T1001.TPDO1.Dur	mmy Byte2		USINT	[Default]	IN :	11.0	1.0		
		L	.T1001.RPDO1.Cor	ntrol Word		UINT	[Default]	OUT :	0.0	2.0		
		L	.T1001.RPDO1.Tan	getPosition		DINT	[Default]	OUT :	2.0	4.0		
		L	.T1001.RPDO1.Tar	getVelocity		DINT	[Default]	OUT :	6.0	4.0		
		L	.T1001.RPDO1.Mo	deOfOperat	ion	SINT	[Default]	OUT :	10.0	1.0		
		L	.T1001.RPDO1.Dur	mmy Byte1		USINT	[Default]	OUT :	11.0	1.0		
		L	.T1001.RPDO1.Erro	orCode		UINT	[Default]	OUT :	12.0	2.0		
		L	.T1001.RPDO1.Dig	itInputs		UDINT	[Default]	OUT :	14.0	4.0		
		Edit Varia	able									
		Mov	e Up Move D	own			New		Edit		Delete	
Class	ic View Flat View	New	Alias Edit Al	lias Dele	ete Alias							•

图 6.94

#### 6.5.2.4. DC 配置

"Distributed Clock"标签页的内容为左侧选中从站的 DC 配置选项。

从"Operation Mode"选项的下拉列表中可以选择从站工作在 FreeRUN 模式,或者工作 在 "DC-Synchron"模式;



<ul> <li>■ 配置/资源</li> <li>■ LT1001 (1001)</li> <li>■ LT1002 (1002)</li> <li>General PDO Mapping Variables Advanced Options Distributed Clock Init Commands CoE Object-Dictionary</li> <li>Distributed Clock</li> <li>Operation Mode DC-Synchron •</li> <li>Sync Unit Cycle (us) 1000</li> <li>Overwrite Mode</li> <li>Sync Unit 5</li> <li>Sync Unit 0</li> <li>Cycle Time</li> <li>User defined 1000</li> <li>Shift Time (us)</li> <li>Sync Unit 1</li> </ul>
LT1001 (1001) Distributed Clock Operation Mode DC-Synchron Sync Unit Cycle (us) Overwrite Mode Sync Units Sync Units Sync Units Sync Unit 0 Cycle Time Sync Unit Cycle Sync Unit 1
Cycle Time Sync Unit Cycle Sync 0 Cycle User defined Shift Time (us)

图 6.96

### 6.6. PDO 与 IO 变量的映射

对每个从站进行 PDO 配置后,需要将 PDO 对象映射到 PLC 程序可访问的全局变量,以 便在 PLC 程序中通过映射的全局变量访问从站的数据。

将 PDO 与 IO 变量映射的步骤如下:

1) 在 MultiProg 的主界面中单击选择菜单"变量映射",弹出如下对话框

anabics mapping		_				
配置/资源/PLC_SYS_TICK_CN	T (UDINT)	<u>~</u>		▼	记置/资源	
配置/资源/PLC_TASK_DEFINE	D (INT)				IN: LT1001.TPDO1.Status Word (UINT)	
配置/资源/PLCMODE_ON (BC	DOL)		< <		IN: LT1001.TPDO1.ActualPosition (DINT)	
配置/资源/PLCMODE_LOADI	NG (BOOL)		X		IN: LT1001.TPDO1.ActualVelocity (DINT)	
配置/资源/PLCMODE_STOP (	BOOL)	=			IN: LT1001.TPDO1.ModeOfOperationDisplay (SINT)	1
配置/资源/PLCMODE_RUN (B	iool)		New		IN: LT1001.TPDO1.Dummy Byte2 (USINT)	
配置/资源/PLCMODE_HALT (	BOOL)				IN: LT1002.TPDO1.Status Word (UINT)	l
配置/资源/PLC_TICKS_PER_SE	C (UINT)				IN: LT1002.TPDO1.ActualPosition (DINT)	
配置/资源/PLC_MAX_ERRORS				IN: LT1002.TPDO1.ActualVelocity (DINT)		
配置/资源/PLC_ERRORS (UDI				IN: LT1002.TPDO1.ModeOfOperationDisplay (SINT)		
配置/资源/PLC_TASK_AVAILA	BLE (INT)				IN: LT1002.TPDO1.Dummy Byte2 (USINT)	
配置/资源/PLC_SYSTASK_AVA	AILABLE (INT)				OUT: LT1001.RPDO1.Control Word (UINT)	
配置/资源/PLCDEBUG_FORCE	(BOOL)				OUT: LT1001.RPDO1.TargetPosition (DINT)	
配置/资源/PLCDEBUG_BPSET	(BOOL)	*		٠		
ask Mapping Resource '配置/资源 (IN)':	<default></default>					
Resource '配置/资源 (OUT)':	<default></default>					
Memory Mapping						
<ul> <li>I/O Channel</li> </ul>						
Shared Memory	iry					
Input:	0					

图 6.97

- 2) 鼠标单击选中右侧 PDO; 按住 CTRL 键同时多次单击可多选;单击选中第一个,然后按 住 SHIFT 同时单击最后一个,可选择从第一个单击选中到最后一个单击选中的所有对象。
- 3) 单击"New",即可将右侧选中的 PDO 映射到全局变量,映射后的全局,在左侧列表中列 出。左侧列表前面列出的是预先定义的一些系统全局变量,若要查看 PDO 映射的全局 变量,则向下拖动左侧列表栏的卷滚条。



图 6.98

- 4) 映射完成后,在 Task Mapping 中选择将 IO 的输入\输出绑定绑定到 PLC 任务,默认选择 为 default,输入\输出将按照 4ms 周期性进行 IO 读取和输出。
- 5) 单击窗口右上角的 X 关闭窗口
- 6) 也可以单击选择菜单"视图"->"全局变量声明",打开全局变量详细列表窗口查看系统变量和 PDO 映射的全局变量的详细信息。

MULTIP	ROG Expr	ess - LTSen	18072	5_2SlaveTry	- (Global	Variable	+配置.受测							0	9 8
11 文件(E)	(REE) 1	REW IN		Z(B) BEELIN	PETEROXO	2									- e ×
			0		-	-			1						
11#I	保存	航大	-	工程时	編載回导	AR	交叉着考 出现面口 安伊工	9.W	新	调成开/关 工程控					
lard.	8 🖗 🖡						名祥		2	用法	模述		地址	初值	保持 ~
网络	10 10 10	NS_PLCopen	LID		1	BSy	tem Variables			•	1				
15.05	57	AXIS			2	PLC S	SYS_TICK_CNT	UDINT		VAR_GLOBAL			%MD1.0	6 - S	13
646	6 📁 🖉	编POU			3	PLC_1	ASK_DEFINED	INT		VAR_GLOBAL	-		%MW1.4	1	10
rite		Test2*			4	PLCM	ODE_ON	BOOL		VAR_GLOBAL	TRUE	: current PLC mod	%MX1.2016.0		13
并行		Untitled*			5	PLCM	ODE_LOADING	BOOL		VAR_GLOBAL	TRUE	current PLC mod	%MX1 2017.0		13
					6	PLCM	ODE_STOP	BOOL		VAR_GLOBAL	TRUE	: current PLC mod	%MX1.6.0		<u>[11]</u>
右线器					7	PLCM	ODE_RUN	BOOL		VAR_GLOBAL	TRUE	current PLC mod.	%MX1.7.0		13
14					8	PLCM	ODE_HALT	BOOL		VAR_GLOBAL	TRUE	: current PLC mod	%MX1.8.0		13
左侧电					9	PLC_1	ICKS_PER_SEC	UINT		VAR_GLOBAL	_		%MW1,2000		
					10	PLC_I	MAX_ERRORS	UDINT		VAR_GLOBAL	-		%MD1.2004		13.11
右列电					11	PLC_E	ERRORS	UDINT		VAR_GLOBAL	_		%MD1.2008		<u>E</u>
10					12	PLC_1	ASK_AVAILABLE	INT		VAR_GLOBAL			%MW1.2012		13
转占类型					13	PLC_S	SYSTASK_AVAILABLE	INT		VAR_GLOBAL			%MW1.2016		1
					14	PLCD	EBUG_FORCE	BOOL		VAR_GLOBAL	TRUE	current PLC mod.	%MX1.2018.0		
					15	PLCD	EBUG_BPSET	BOOL		VAR_GLOBAL	TRUE	one or more	%MX1.2019.0		13
资制72					16	PLCD	EBUG POWERFLOW	BOOL		VAR_GLOBAL	TRUE	: current PLC mod	%MX1.2020.0		
					1/	Eth	erCAT Variables	1.4140					-	1 4100	100
URIAFP					18	0000		AUS		VAR_GLOBAL	-			(AUS	
					19	V001		AXIS		VAR_GLOBAL	_		-	(AXIS	10 11
2					20	1002		AND		VAR_GLOBAL	-		-	( AVID	
CIU SP					21	VOUS		MOOD	-	VAR GLOBAL			-	( MUD	
					22	VU16		NURD		VAR_GLOBAL			A/14/02 4000	-	83
MLASP					23	Toott	reakruum.	UCHUT		VAR GLOBAL			%MD3.1000		100
					24	I T100	1 TRDO1 Status Mord	LIMT		VAR GLOBAL	-		96840		
CHARADIT					26	1 T100	1 TPDO1 ActualPosition	DINT		VAR GLOBAL			96102		10 10
					27	1 1 100	1 TPDO1 Actual/elecity	DINT		VAR GLOBAL	-		\$105		10
					28	1 T100	1 TPDO1 ModeOfOnerationD	SINT		VAR GLOBAL	-		%UB10		100
					29	1 T100	1 TPDO1 Dummy Bute?	USINT		VAR GLOBAL	-		94811		10
					30	LT100	2 TPDO1 Status Word	LIINT		VAR GLOBAL			\$4W18		1 191
					31	LT100	2 TPDO1 ActualPosition	DINT		VAR GLOBAL	-		%ID20	-	100
					32	LT100	2 TPDO1 ActualVelocity	DINT		VAR GLOBAL	-		%ID24		191
					33	LT100	2 TPDO1 ModeOfOperationDi	SINT		VAR GLOBAL	-		%B28	-	PI
					34	LT100	2 TPDO1 Dummy Byte2	USINT	_	VAR GLOBAL	-		%1829		10
					35	LT100	1 RPDO1 Control Word	UINT		VAR GLOBAL			%QW0		P
					26	117100	TODON TANANDANSIAN	PHAT		VAD CLORAL	_		\$* 0 M		1 10 1
	1	-			_							1			
	@ #8	<b>m</b> #0			8 f	【码:Test	2 [] Global_V								
0												E S & V = 8	/ POU	(T-12.00	1
1											111			-	
												-			
-												1			
EAPA	■☆ / (11)	人警告人	信息入口	PLC出错入#	TED /							• A	10	_	,
ARD CO. H	NIØF1					_			_		and Loop of				D: >2GE
Contraction of the later	and the														

图 6.99

#### 6.7. 变更从站

当一个工程项目的从站设备发生变化,则需要对 MultiProg 的从站配置信息进行修改。

### 6.7.1. 重新自动扫描从站

若要通过重新自动扫描从站修改从站配置,必须确保从站通过网线正确连接到 i5,且 i5 与安装 MultiProg 的 PC 通过网线连接,连接方法见硬件手册。i5 和从站应当处于上电运行状态。

修改方法和步骤如下:

1) 单击选择菜单"视图"->"EtherCAT",打开 EtherCAT 配置窗口。鼠标右键单击左侧列表中 "配置/资源"项。

roject Explorer	Device Editor	
▲ 配置/资源 ④ Slave_1001 [Leetro A1E] (1001) ④ Slave_1002 [Leetro A1E] (1002)	Master Process Data Image General Unit Name Cycle Time [us] Source MAC address	Advanced Options Distributed Clocks 配置/资源 1000   O0-0A-35-11-22-33

图 6.100

2) 在弹出的菜单中选择"Scan EtherCAT Network"菜单项。



图 6.101

3) 在弹出的对话框中,系统提示重新扫描将删除设备树中已经存在的从站,是否继续。单击"是",系统开始重新扫描并重建设备树。



图 6.102

进度条显示完成后,窗口左侧设备树中显示扫描到的设备。若没有扫描到设备,请对照 硬件手册检查 i5 和从站的连接,以及设备是否处于上电运行状态。

oject Explorer	Device Editor		
↓ 配置/资源	Master Process Data Image	Advanced Options Distributed Clocks	
Slave_1001 [Leetro A1E] (1001)	Control		
Slave_1002 [Leetro A1E] (1002)	Unit Name	ETER ANYE	
	Curle Time [ur]	10直/资源	
	Cycle Time [ds]	1000	•
	Source WAC address	00-0A-35-11-22-33	

图 6.103

然后按照 6.3 和 6.4 节重新配置 EtherCAT 相关参数,并进行 PDO 与 IO 变量的映射。

重新扫描重建设备树后,原设备树中的从站设备的 PDO 建立的全局 IO 变量的映射关系 自动取消,但全局变量仍在全局变量表中,但地址栏已经为空。新扫描到的从站设备的 PDO 映射后的全局变量名若与之前的一致,则系统会自动在变量名后增加"\_001"后缀加以区分, 避免同名。

MULTIP	ROG Express - LTSen	vo180725	_2SlaveTry -	Global_V	ariables:配置.资源]						6 X
🔝 文件(E)	编辑(E) 视图(V) 工程	副(P)建立	(B) 布局(L)	联机( <u>N</u> )	附加(X) 窗口(W) <u>?</u>						_ # ×
₩ 打开工	保存 放大	- 编小	工程树	編輯向导		● 支重 制作	<b>演</b> 试开/关 工程控				
lacs	□ 10 K R C	TTR			名称	类型	用法	描述	地址	初值	保持 ^
网络	□ 10 数据类型	مس		22	V016	WORD	VAR_GLOBAL				
4.6	AXIS			23	EcatBreakNum	UDINT	VAR_GLOBAL		%MD3.1000		
右触点	😑 🧊 逻辑POV			24	TestRetain	UDINT	VAR_GLOBAL		%MD3.1004		
r‼r	Test2*			25	LT1001_TPDO1_Status_Word	UINT	VAR_GLOBAL				
并行	onci ciedo			26	LT1001_TPDO1_ActualPosition	DINT	VAR_GLOBAL				
- IKO				27	LT1001_TPDO1_ActualVelocity	DINT	VAR_GLOBAL				
右线圈				28	LT1001_IPDO1_ModeOfOperationDi	SINT	VAR_GLOBAL				
<b>H</b>				29	LT1001_TPDO1_Dummy_Byte2	USINI	VAR_GLOBAL				
左侧电				30	LT1002_TPDO1_Status_Word	DINT	VAR_GLOBAL				
- IH				32	LT1002_TPDO1_ActualValacity	DINT	VAR_GLOBAL				
右侧电				32	LT1002_TPDO1_Actual/elocity	SINT	VAR_GLOBAL				
4				34	1 T1002_TPDO1_NiddeoloperationBr	LISINT	VAR GLOBAL				
触点类型				35	LT1001 RPD01 Control Word	LINT	VAR GLOBAL				
				36	LT1001 RPD01 TargetPosition	DINT	VAR GLOBAL				
(日本)(pp				37	LT1001 RPDO1 TargetVelocity	DINT	VAR GLOBAL				
2 mjr i				38	LT1001 RPDO1 ModeOfOperation	SINT	VAR GLOBAL				
+TI+A PP				39	LT1001 RPDO1 Dummy Byte1	USINT	VAR GLOBAL				
6/JJ#11				40	LT1001_RPDO1_ErrorCode	UINT	VAR_GLOBAL				E [
<b></b>				41	LT1001_RPDO1_DigitInputs	UDINT	VAR_GLOBAL				
创建步				42	LT1002_RPDO1_Control_Word	UINT	VAR_GLOBAL				
母				43	LT1002_RPDO1_TargetPosition	DINT	VAR_GLOBAL				
插入SF				44	LT1002_RPDO1_TargetVelocity	DINT	VAR_GLOBAL				
				45	LT1002_RPDO1_ModeOfOperation	SINT	VAR_GLOBAL				
创建动作				46	LT1002_RPDO1_Dummy_Byte1	USINT	VAR_GLOBAL				
				47	LT1002_RPDO1_ErrorCode	UINT	VAR_GLOBAL				
				48	LT1002_RPDO1_DigitInputs	UDINT	VAR_GLOBAL				
				49	LT1001_TPDO1_Status_Word_001	UNI	VAR_GLOBAL		%1000		
				50	LT1001_TPDO1_ActualPositio_001	DINT	VAR_GLOBAL		%ID2		
				51	LT1001_TPDO1_ActualVelocit_001	PINT	VAR_GLOBAL		%ID6		
				52	LT1001_TPDO1_WodeOrOperati_001	LISINT	VAR GLOBAL		%IB11		
				53	LT1002 TRDO1 Statue Word 001	LINT	VAR_GLOBAL		%IW/18		
				55	T1002_TPDO1_Status_V010_001	DINT	VAR_GLOBAL		%ID20		
				56	LT1002 TPDO1 Actual/elocit 001	DINT	VAR GLOBAL		%ID24		
				67	LT1002 TDDO1 MadeOfOperati 001	CINIT	WAD CLOBAL		0/1000		
						III					•
	◇学工程 11 硬件			■ 代	码:Test2 [] Global_V						
帮助信息 · 请	转F1										D: ≥2GB
TRANSPORT PROVIDENT											0.0200

图 6.104

为避免重新扫描到的从站设备的 PDO 映射后的全局变量名与扫描前已经存在的全局变量名相近,不便于区分,可删除不再使用的全局变量名。

删除全局变量的方法如下:

- 在全局变量详细信息列表中,单击选中要删除的全局变量。若要多选不连续的多个变量, 按住"CTRL"键同时鼠标多次单击要删除的变量;若要多选连续的多个变量,单击连续区域的第一个变量,按住"SHIFT"键的同时鼠标单击连续区域的最后一个变量,则从第一个 到最后一个变量之间的所有变量被选中。
- 2) 鼠标右键单击选中的变量,在弹出菜单中选择"删除变量/组",即可删除选中的变量。

36	01001		ROOL	VAR_GLO	
37	8 Eth-	***********	and the second second	The second s	
38	S1004	插入交量(A) 台建交量值(B)	Ctrl+Shift+1	VAR GLO	
39	S1004	创建变量集(匹)		VAR GLO	
40	S1004	進加完量(N)		VAR_GLO	
41	S1004	加入组印	Ctrl+G	VAR GLO	
42	S1004	影除变像/组(0)	Ctrl+Shift+D	VAR GLO	
43	S1004	借止/律能管量(5)	Ctrl+Shift+E	VAR_GLO	
44	S1004	-		VAR GLO	
45	S1004	\$\$\$\$0(D)	CBI+X	VAR GLO	
46	S1004	F 周期(G) 9 枯渇(P)	Ctrl+C	VAR_GLO	
47	S1004		Ctrl+V	VAR GLO	
48	S1004	过洋器设置(图		VAR_GLO	
49	S1004	过续器(E)		VAR_GLO	
50	S1004	御序(Q)	•	VAR_GLO	
51	S1004	TILMAN		VAR_GLO	
52	S1004	1111000000		VAR_GLO	
53	S1004	找出所有引用		VAR_GLO	
		置行	· ·	A Martin Theorem 1	
20		只謂(編)			
THOT	00 BL_Y	Wine and			
0.00		SELECTION STATES	-		

图 6.105

### 6.7.2. 手动修改从站配置

therCAT Configurator		
▼ ■ 配置/资源 ■ Slave 1001 [Lettro A1E] (1001)	General PDO Mapping Variables	Advanced Options Distributed Clock Init Commands CoE Object-Dictionary
Slave_1002 [Leetro A1E] (1002)	Address Station Address Information	1001 😨
	Name	Slave_1001 [Leetro A1E]
	Description	Leetro CiA402 Drive
	Vendor	Leetro Automation Corp., Ltd (0x000005E7)
	Product Code	0x12343052 (305410130)
	Revision Number	0x10101070 (269488240)
	ESI File	C:\ProgramData\PHOENIX CONTACT Software\MULTIPROG Express\5_50_10209 \Leetro_A1E_Ver1.4_20180719.xml
	Identification Value 🧅	Not Used
	Ports	
	А 🌒	配置/资源
	D 🧼	Not Available
	в 🥥	Slave_1002 [Leetro A1E] (1002)
	c 🌒	Not Available
•		•

1) 启动 MultiProg, 单击菜单"视图"->"EtherCAT", 打开如下对话框:

图 6.106

2) 选中左侧设备树中要删除的从站设备。右键单击该选中设备,在弹出对话框中选择 "Remove Slave"菜单项,即可删除该设备配置。



图 6.107

然后按照 6.2.3 节描述手动添加从站,并按照 6.3 和 6.4 节描述配置参数,完成 PDO 与 IO 变量的映射。

#### 6.8. 轴参数配置

在运动控制系统中,将运动控制的对象称为"轴"。轴中含有通过 EtherCAT 连接的实际 伺服驱动器, EtherCAT 接口的步进电机驱动器和 MC 功能模块内部的虚拟轴。

MC 功能模块的轴中包括下表中的种类。

轴种类	内容
伺服轴	使用EtherCAT从站伺服驱动器的轴。分配为实际伺服驱动器加以使用。

	将1台伺服电机作为1根轴使用。
虚拟伺服轴	MC功能模块内的虚拟轴。不使用实际伺服驱动器。作为同步控制的主轴
	等使用。

#### MC 功能模块的轴中含有以下相关要素

构成要素	内容
轴参数	为MC功能模块控制的各轴设定最大速度、启动速度、原点返回等轴
	动作。
轴变量	轴变量是由MC功能模块控制的包含各轴的轴号等参数的结构体变
	豊。
用户程序中轴的	在用户程序中,使用运动控制指令执行运动控制。将执行单轴控制
指定方法	的运动控制指令称为"轴指令"。
	在指令的输入输出变量结构体Axis_REF中,设定该结构体中的成员
	轴号变量的参数值指定轴指令控制哪个轴。

# 6.8.1. 轴参数概要

分类	参数名称				
轴基本设置	轴号				
	轴使能				
	轴类型				
	控制类型				
	输出控制				
单位换算设置	显示单位				
	电机转一周指令脉冲数				
	电机转一周工作行程				
操作设置	最大速度				
	启动速度				
	最大点进速度				
	最大加速度				
	最大减速度				
	跟随误差检测使能				
	最大允许跟随误差				
	定位范围				
	定位检查时间				
	位置偏差报警时间				
	立即停止输入停止方法				
	限位停止输入立即停止方法				
	正限位				
	负限位				
	立即停止				
限位设置	软件正限位				
	正软件限位				

# ● 轴参数一览表

	软件负限位
	负软件限位
原点返回设置	回原点模式
	回原点速度
	近点速度
	原点返回加速度
位置计数设置	计数模式
	模最大位置设计值
	编码器类型
	模最小位置设计值
伺服驱动设置	电机最大速度
	转矩偏移量
	速度偏移量
	位置偏移量
	加加速度倍率因子
	加速度倍率因子
	速度倍率因子
	反向间隙补偿
	反向间隙值
	反向间隙补偿速度
	螺距误差补偿
	螺距误差补偿间隔
	螺距误差表

# • 使用轴时必须设定的项目

为通过 MLUTIPROG 创建的轴设定以下项目后可使用轴

分类	参数名称	设定内容
轴基本设置	轴号	自定义轴号,由阿拉伯数字表示
	轴使能	选择[1#使用的轴]
	轴类型	选择需控制的轴的种类
	选择[0#开环]	
	输出控制	选择轴中分配的EtherCAT 从站设备的节点地址。 轴种类为[虚拟轴]时无法选择。

# ● 通过MLUTIPROG试运行伺服驱动器时须设定的项目

分类	参数名称	设定内容
轴基本设置	轴号	自定义轴号,由阿拉伯数字表示
	轴使能	选择[1#使用的轴]
	轴类型	选择需控制的轴的种类
	控制类型	选择[0#开环]
	输出控制	选择轴中分配的EtherCAT 从站设备的节点地址。
		轴种类为[虚拟轴]时无法选择。
单位换算设置	显示单位	选择[脉冲]、[毫米]等需显示的单位

电机转一周指令脉冲数	根据编码器的分辨率,设定电机转1圈的脉冲数。
电机转一周工作行程	根据机械的规格,设定电机转1圈的工件移动量。

### 6.8.2. 轴的设定步骤

### 6.8.2.1. 打开运动控制设置界面

- 1) 选择菜单【视图】->【运动控制】,打开运动控制设置界面
- 2) 鼠标右键单击左侧树状列表中的列表项[轴设置],在弹出菜单中单击[新建轴],完成后 鼠标左键双击 MC\_Axis001,便能对新建的轴进行基本设置。



图 6.108

### 6.8.2.2. 轴基本设置

运	动	空制		6				
	4	<b>轴设置</b> MC_Axis000 MC_Axis002 轴组设置	抽基本设置         抽号         抽使能         抽类型         控制类型         输出控制	単位換算设置 1 1#使用 0#伺服 0#开环 1001#S	: 操作设置 的轴 ▼ au ▼ lave_1001	)限位设置) · [Leetro A1	原点返回设置	
L	4							÷

图 6.109

### ● 轴号

设定逻辑轴号,数值范围为1到32,不能与其他轴重复,该逻辑轴号在PLC程序中与轴 参数数据结构中的轴号对应,以便功能块可以通过轴号对[输出控制]项设定对应的从站上的 轴进行控制。

● 轴使能

使能轴或不使能轴,如果为不使能,在运动控制功能块中使用该轴将会产生错误提示。

#### ● 轴类型

当前支持伺服轴及虚拟轴。

● 控制类型

未使用,当前值默认值为开环,修改后不会生效。

# ● 输出控制

设定将轴号映射到 EtherCAT 从站(若一个从站只有一个轴,大多数 EtherCAT 伺服为此种模式),或从站的轴(若一个从站含多个轴)。



必须将轴号映射到从站上的轴或者虚拟轴,否则对工程 编译将会显示出错信息

运动控制		×
运动控制 《 轴设置 MC_Axis001 MC_Axis003 MC_Axis004 4 轴组设置 MC_Group001	曲基本设置] 単位換算设置] 操作设置] 限位设置] 原点返回设置] 位置计数设置] 作 記示单位 記机转一圈脉冲数 10000 脉冲 記机转一周工作行程 10000 脉冲	■ 3月辰弘区志力〕
4		+

图 6.110

PLC 程序中所有运动指令的位置单位称为"指令单位",控制器主站与伺服从站(伺服 驱动器)之间使用脉冲单位的位置。指令单位与脉冲单位之间通过主站电子齿轮比(或称脉 冲当量)进行换算。

此功能的使用,可以设定每单位的指令位置对应的电机脉冲数。



具体设置方法是:

在轴的单位换算设置一栏中的"电机转一周指令脉冲数"后面的文本框里输入齿轮比的 分子,在"电机转一周的工作行程"后面的文本框里输入齿轮比的分母,则

> 主站电子齿轮比 = 电机转 1 周的脉冲数 电机转 1 周轴的工作行程

工作行程指工作台移动的位移。工作行程单位,即指令单位,由显示单位选项指定。主 站电子齿轮参数可以包含机械减速比。 在运动指令中位置相关的单位均为指令单位,在 MultiProg 中显示的位置相关的单位为 脉冲单位。



6.8.2.4. 操作设置

运动控制			_		_	_	-
运动控制  4 铀设置 MC_Axis001 MC_Axis002 MC_Axis003 MC_Axis004 4 铀组设置 MC_Group001	MCAxis001 抽基本设置 单位排 最大速度 最大转矩 启动速度 最大点进速度 最大点进速度 最大流氓差检测使能 最大允许跟随误差 到位力断模式 定位检查时间 位置偏差报警时间 立即停止输入停止力 正限位 众限位	▲算设置)操作设置 3000 100 1000000 1000000 00世使能 ▼ 10000 1=根根据目标位置折 100 10 5法(0±立即停止 5法(0±立即停止 0±无效 ▼ 0=无效 ▼	8 <u>限位设置</u> 旅中/s 成:0.1% 脉:中/s 脉:中/s 脉:中/s 指令单位 指令单位 指。	原点返回设置	位置计数设置	何服逐步改变置)	
▲ 铀设置 MC_Axis001 MC_Axis003 MC_Axis003 MC_Axis004 ▲ 铀组设置 MC_Group001	WCAxis001 141 基本设置 单位排 最大速度 最大转矩 启动速度 最大点进速度 最大加速度 最大城速度 跟随误差检测使能	▲算设置)操作设置 4E+08 3000 100 100 1000000 1000000 0=使能 ▼	<u>8</u> ] 限位设置) 脉:中/s x0.1% 脉:中/s 脉:中/s <sup>=</sup> 脉:中/s 脉:中/s 脉:中/s	原点返回设置	位置计数设置	何服驱动设置	
	最大允许跟随误差 到位判断模式 定位范围 定位检查时间 位置偏差报警时间 立即停止输入停止7 限位停止输入停止7 正限位 负限位 立即停止	10000 1#根据目标位置并 100 10 10 5法 0=立即停止 5法 0=立即停止 0=无效 ▼ 0=无效 ▼ 0=无效 ▼	指令单位 <u>谢</u> 斯 • 指令单位 == == •				

图 6.111

#### ● 最大速度

设定了轴的速度上限,运动控制功能块中设置的速度参数若超过这个参数值,则实际被 限制到此速度值。若低于这个参数值,则为功能块设置的速度值。

#### ● 最大转矩

设定了轴的转矩上限,运动控制功能块中设置的转矩参数若超过这个参数值,则实际被限制到此限定值。注意:转矩上限设置过小,可能引起轴的加减速时间加长,无法及时响应动作指令,也可能造成减速距离过长,引起碰撞。

#### ● 启动速度

设置轴的启动速度,轴的运动将从此速度开始按照功能块设定的加速度完成升速过程。

#### ● 最大点进速度(未使用)

#### ● 最大加速度

设置轴的加速度上限,运动控制功能块中设置的加速度参数若超过这个参数值,则实际 被限制到此加速度值。若低于这个参数值,则为功能块设置的加速度值。

#### ● 最大减速度

设置轴的减速度上限,运动控制功能块中设置的减速度参数若超过这个参数值,则实际 被限制到此减速度值。若低于这个参数值,则为功能块设置的减速度值。

#### 跟随误差检测使能(未使用)

#### ● 最大允许跟随误差

设置轴的最大允许跟随误差,若电机反馈与伺服当前位置的差的绝对值超过设置值,将 会产生报警。

### ● 到位判断模式

设置判断到位的模式,可选"根据目标位置判断"或"根据当前位置判断"。

- 位置偏差报警时间(未使用)
- 立即停止输入停止方法(未使用)
- 限位停止输入停止方法(未使用)
- 正限位(未使用)
- 负限位(未使用)
- 立即停止(未使用)

#### 6.8.2.5. 限位设置

运动控制	
▲ 袖设置 MC_Axis001 MC_Axis002 MC_Axis003 MC_Axis004 ▲ 袖徂设置 MC_Group001	抽基本设置         単位換算设置         操作设置         限位设置         原点返回设置         位置计数设置         伺服服运动           软件正限位         0=无效
•	III >

图 6.112

【描述各参数的意义】

● 软件正限位

使能或取消使能 软件正限位功能

● 正软件限位

设置软件正限位值,若被使能,在运动控制中设置绝对位置值大于该设定值,功能块将 产生错误。

● 软件负限位

使能或取消使能 软件正限位功能

● 负软件限位

设置软件负限位值,若被使能,在运动控制中设置绝对位置值小于该设定值,功能块将 产生错误。

### 6.8.2.6. 原点返回设置

Motor	Control Settir	ng		
4	油设置	轴基本设置 单位排	與算设置 操作设置	問,限位设置)原点返回设置(位置计数设置)何服驱动设置
	MC_Axis001 MC_Axis002	回原点模式	1#: -Ve LS & In	ndex Fulse 🔻
\$	曲组设置	回原点速度	10000	脉冲/s
		近点速度	2000	)航心中/s
		原点返回加速度	30000	βῆκ)-ἡ/s²

#### 图 6.113

【描述各参数的意义】

### ● 回原点模式

设置伺服回原点功能的回原点方式。具体信息参考第7.2.6节

● 回原点速度

设置伺服回原点时的速度,在启动运动控制的回原点功能后,将以该速度进行回零点。

#### ● 近点速度

当回原过程中接受到近点信号,伺服速度将按照该设定值运行。

### ● 原点返回加速度

设置回原点过程中的加速度。

# 6.8.2.7. 位置计数设置

运动控制	
▲ 轴设置 MC_Axis00 MC_Axis00 轴组设置	₩基本设置 单位换算设置 操作设置 限位设置 原点返回设置 位置计数设置 作 编码器类型 ● 增重式编码器 ●绝对编式码器 计数模式 ● 线性模式 ● 循环模式
•	

图 6.114

【描述各参数的意义】

- 计数模式(未使用)
- 编码器类型(未使用)

#### 6.8.2.8. 伺服驱动设置



⊿ 轴设置	MCAxis001					
MC_Axis001 轴组设罟	轴基本设置单位排	與算设置 操作设置	1 限位设置	原点返回设置	位置计数设置	伺服驱动设置
MAL WILL	电机最大速度	3000	转/min			
	转矩偏移里	0	N•m			
	速度偏移重	0	脉冲∕≤			
	位置偏移重	0	脉冲			
	加加速度倍率因子	100	%			
	加速度倍率因子	100	%			
	速度倍率因子	100	%			
	反向间隙补偿	🔘 不启用 🔍 启用				
	反向间隙值	0	脉冲			
	反向间隙补偿速度	0	脉冲/≤			
	螺距误差补偿	🔘 不启用 🔍 启用				
	螺距误差补偿间隔	0	脉冲			
	序号 误差(脉	)(中)				
	0 1.2					
	1 0.2					
	2 -1					
	2 -0					
	3 -2					
	4 0					

图 6.115

【描述各参数的意义】

#### ● 电机最大速度

该值为电机的最大运行速度

- 转矩偏移量(未使用)
- 速度偏移量(未使用)
- 位置偏移量(未使用)
- 加加速度倍率因子(未使用)
- 加速度倍率因子(未使用)
- 速度倍率因子

实际速度=设定目标速度\*速度倍率因子/100; 速度倍率因子数值范围为0~200;

反向间隙补偿
 启用后使能反向间隙功能,反向间隙功能如下图:



图 6.116

反向间隙值

● 反向间隙补偿速度

#### ● 螺距误差补偿

启用后使能螺距误差补偿功能,螺距补偿功能如下图:



#### ● 螺距误差补偿间隔

螺距误差按照等间隔的方式进行补偿,螺距误差表中相邻两个补偿点之间的距离

#### ● 螺距误差表

对整个工作行程(软件限位的负极限到正极限)进行等间隔(螺距误差补偿间隔) 划分,在每个补偿点设置设补偿值,补偿点之间的补偿值按照相邻两个补偿点进行线性 计算;

误差表的第一个补偿值对应位置为软件限位的负极限位置。 螺距误差表最多允许**100**个元素。

#### 6.8.3. 轴变量

i5 中的轴变量由数据类型定义如下,用户可以自己在工程中按如下格式添加数据类型, 也可以通过选择包含轴数据类型的工程模板建立工程后直接使用。

工程模板中包含的轴变量数据类型定义如下



图 6.118

### 6.8.4. 用户程序中轴的指定方法

双击功能块的 Axis 引脚,弹出变量属性窗口,单击名称下拉菜单,选择在 6.8.2 中创建的轴。

变量属性		8
名称(N): MC_AxisOO1  数据类型(D): AXIS_REF  用法(U): VAR_GLOBAL  KETAIN  初值(1): ( AxisNo := 1 ) I/O地址(S): 描述(E): Axis creat by system	定义范围 ◎ 局部(L) 全局(G) 局部変量组(V): ② Default ▼ 全局変量组(A): □ ● 物理硬件 ◎ 武器 ○ ○ 常器 ○ ○ System Variables	确定 取消 帮助(H)
□ PDD □ OPC □ 隐藏(X) □ 初值作为默认值.	☑ 显示工作单的所有变量 (₩) □ 按组名排序	

图 6.119

### 6.9. 轴组参数配置

# 6.9.1. 轴组概要

轴组由多个轴构成,运行直线插补和圆弧插补等含多个轴的指令时,使用轴组。 在 MLUTIPROG 中使用功能块可启动轴组的运动控制指令或获取轴组的状态。 轴组规格如下:

项目	规格
轴组数	最多 32 组
构成轴数	最多4轴/轴组

#### 轴组中含有以下相关要素。

构成要素	内容
轴组参数	为MC功能模块控制的各轴组设定最大插补速度和最大插补加减速度等
	轴组动作。
轴组变量	轴组变量是由MC功能模块控制的各轴组的部分轴组参数、指令插补速
	度及异常信息等监控信息组成的变量。
用户程序中轴组	在用户程序中,使用运动控制指令执行运动控制。将执行多轴协调控
的指定方法	制的运动控制指令称为"轴组指令"。
	在指令的输入输出变量结构体Axis_GROUP_REF中,设定该结构体中的
	轴组号、轴的数量、各轴的轴号三个变量的参数值指定轴指令控制哪
	个轴。

# 6.9.2. 轴组参数概要

### ● 轴组参数一览表

分类	参数名称
轴组基本设置	轴组号
	轴使能
	构成
	升降速模式
	轴 AO、A1、A2、A3

轴组操作设置	最大插补速度
	最大插补加速度
	最大插补减速度

### ● 使用轴组时必须设定的项目

为通过 MLUTIPROG 创建的轴组设定以下项目后可使用轴组。

分类	参数名称	设定内容
轴组基本设置	轴组号	为轴组设定一个编号,由阿拉伯数字表示
	轴使用	选择[1#使用的轴]
	构成	选择需控制的轴构成
	升降速模式	选择轴组的升降速模式
	轴 AO、A1、A2、A3	选择分配至轴组的轴,最多可分配4轴

### 6.9.3. 轴组的设定步骤

### 6.9.3.1. 打开运动控制设置界面

- 1) 选择菜单【视图】->【运动控制】,打开运动控制设置界面
- 2) 鼠标右键单击左侧树状列表中的列表项[轴组设置],在弹出菜单中单击[新建轴组],完成后鼠标左键双击 MC\_ Group001,便能对新建的轴组进行基本设置。



图 6.120

### 6.9.3.2. 轴组基本设置

运动控制		
▲ 抽设置 MC_Axis001 MC_Axis002 MC_Axis004 ▲ 抽组设置 MC_Group001	抽組基本设置】       抽組操作设置】         抽組令       1         抽組使用       1#使用的轴组 ▼         构成       0#2轴 ▼         升降速模式       0#1型 ▼         轴A0       1#MC_Axis001 ▼         轴A1       2#MC_Axis002 ▼	
•	III	Þ

图 6.121

#### ● 轴组号

设置轴组的轴组号,注意与其他轴组不能重复。

### ● 轴使能

是否使能该轴组,若为使能,在调用运动控制块轴组相关功能块时将产生错误提示。

#### ● 构成

设置轴组的构成数量,最少为2轴,最多为4轴

### ● 升降速模式

设置轴组的插补升降速模式

# ● 轴A0...轴A4

将轴组中的逻辑轴号与配置的轴进行映射。

#### 6.9.3.3. 轴组操作设置

运动控制				<b>×</b>
▲ 轴设置	袖组基本设置)轴	祖操作设置		
MC_Axis001 MC_Axis002	最大插补速度	4E+08		
MC_Axis003	最大插补加速度	1000000		
MC_Axis004	最大插补减速度	1000000		
▲ 細組设立 MC Group001				
sc_oroapoor				
				4

图 6.122

#### ● 最大插补速度

设置轴组的最大插补速度值,当运动控制功能中轴组的功能块插补速度值超过该设定值 时,将按照该设定值运行,否则则按照功能块设定值运行。

#### ● 最大插补加速度

设置轴组的最大插补加速度值,当运动控制功能中轴组的功能块插补加速度值超过该设 定值时,将按照该设定值运行,否则则按照功能块设定值运行。

#### ● 最大插补减速度

设置轴组的最大插补减速度值,当运动控制功能中轴组的功能块插补减速度值超过该设 定值时,将按照该设定值运行,否则则按照功能块设定值运行。

### 6.9.4. 轴组变量

i5 中的轴组变量由数据类型定义如下,用户可以自己在工程中按如下格式添加数据类型, 也可以通过选择包含轴组数据类型的工程模板建立工程后直接使用。 工程模板中包含的轴组变量数据类型定义如下



图 6.123

# 6.9.5. 用户程序中轴组的指定方法

双击功能块的 AxesGroup 引脚,弹出变量属性窗口,单击名称下拉菜单,选择在 6.9.3 中创建的轴组。

变量属性		23
名称(N): MC_Group001 ↓ 数据类型(D): AXIS_GROUP_REF ↓ 用法(U): VAR_GLOBAL ↓ RETAIN 初值(I): (AxisGroupNo:=1) I/O地址(S):	定义范围 ● 局部(L) 全局(G) 局部变量组(V): ■ Default 全局变量组(A): ◆ 物理硬件 ● ● 翻置 ● ● ◎ 配置 ● ● ○ ◎ 配置 ● ● ◎ NU ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	确定 取消 帮助(H)
描述(E): [SYS]Axis group creat by system		
■ PDD ■ OPC □ 隐癒(X) □ 初值作为默认值.	☑ 显示工作单的所有变重(₩) □ 按组名排序	

图 6.124

# 7. 控制程序的编写

# 7.1. 支持的编程语言

### 7.1.1. 编程语言列表

IEC61131-3 标准定义了 5 种编程语言,下表列出了这 5 种编程语言。

序号	中文名称	英文名称	特点
1	指令表	Instruction	指令表编程语言是一种低层语言,具有容易记忆、
		List,简称 IL	便于操作的特点,同时其占用系统资源少,比较适
			合在功能不复杂的小型控制系统中使用
2	梯形图	Ladder	是 PLC 首先采用的编程语言,也是 PLC 最普遍采
		Diagram,简称	用的编程语言。梯形图编程语言是从继电器控制系
		LD	统原理图的基础上演变而来的,与继电器控制系统
			梯形图的基本思想是一致的,只是在使用符号和表

			达方式上有一定区别。其容易被工厂电气人员掌
			握,特别适用于开关量逻辑控制。
3	功能块图	Function Block	是用逻辑功能符号组成的功能块来表达命令的图
		Diagram,简称	形语言,与数字电路中逻辑图一样,它极易表现条
		FBD	件与结果之间的逻辑功能。这种编程方法是根据信
			息流将各种功能块加以组合,是一种逐步发展起来
			的新式的编程语言,正在受到各种可编程控制器厂
			家的重视。由于具有图形符号、图形连接,且操作
			方便,因此,该编程语言将被广泛采用
4	结构化文本	Structured	大量采用抽象性语句来描述复杂控制系统地功能,
		Text,简称 ST	是一种高层语言。更适用于复杂算法的实现。
5	顺序功能图	Sequence	针对顺序控制系统的控制条件和过程的一套表示
		Function	逻辑控制功能的方法。由于其结构严密,流程清晰,
		Chart , 简 称	简单易学,有利于设计人员和其他专业人员的交流
		SFC	沟通

下面将根据这 5 种语言的特点,结合实际的例子说明在 MULTIPROG 中使用这些语言进行编程。

### 7.1.2. 指令表 (IL) 编程语言

指令表(Instruction List,简称 IL)编程语言起源于德国的"Anweisungsliste(AWL)", 主要在欧洲和日本流行。其基本原理是基于单累加器操作的执行模式,即每行代码只执行一 次操作。

指令表编程语言是一种低层语言,具有容易记忆、便于操作的特点,同时其占用系统资源少,比较适合在功能不复杂的小型控制系统中使用。

示例:实现四则运算 A+B×(C+D)。

- 使用工程建立向导新建一个工程,工程名称为"LanguageSample",POU 名称为 "MAIN",语言选择"FBD",配置选择"Leetro\_I5\_1213",资源选择"eCLR EtherCAT V2.0.0",任务选择"Cyclic"。
- 2) 在工程树 POU 节点中插入功能块,名称为"ILSample",语言选择为"IL"。
- 3) 双击打开代码工作单,在其中编写如图所示代码。

1	LD D
2	ADD C
3	MUL B
4	ADD A
5	ST X
6	



# 图 7.2 IL代码实现

双击打开变量工作单,在其中定义如图所示的变量。

名称	类型	用法
🗆 Default		
A	INT	VAR_INPUT
В	INT	VAR_INPUT
C	INT	VAR_INPUT
D	INT	VAR_INPUT
X	INT	VAR_OUTPUT

图 7.3 定义变量

- 4) 点击编译工作单按钮。
- 5) 双击打开程序"MAIN"的代码工作单,在右侧的编辑向导中选择"LanguageSample",将

ILSample 功能块拖拽至代码工作单,如图所示。



图 7.4 ILSample 功能块实例

### 7.1.3. 梯形图 (LD) 编程语言

梯形图(LD-Ladder Diagram)语言是 PLC 首先采用的编程语言,也是 PLC 最普遍 采用的编程语言。梯形图编程语言是从继电器控制系统原理图的基础上演变而来的,与继电 器控制系统梯形图的基本思想是一致的,只是在使用符号和表达方式上有一定区别。其容易 被工厂电气人员掌握,特别适用于开关量逻辑控制。

示例: 电机的启动、保持和停止控制

- 1) 使用指令表编程语言示例中所建立的工程,在工程树 POU 节点中插入功能块 "LDSample",语言选择"LD"。
- 2) 打开代码工作单,点击快捷工具栏中的"网络"功能,在代码工作单中插入 LD 网络,如图所示。

	ÌХ	:件(E)	编辑(E)	视图(⊻)	工程(P)	建立( <u>B</u> )	对象( <u>O</u> )	布	局(L)	联机( <u>N</u>	) 附加(X)	窗口(W)	2		
¢		] 🔍	۹ 🗾	۲		🔁	₩ĸ¥ <b>₩</b> ₩	切	<del>  0</del>	11.11	₩ ÷	· 물 몸	<b>4</b> 8	🛗 🏅	<b>a</b>
		库 数据 逻辑I II M F F	S_PLCoper 类型 xis_Type OU Dsample* AIN* BI* C1*	n_LIB ∗				C			cooo 1 1		C001	· · ·	

图 7.5 插入 LD 网络

3) 双击"C000"触点,在弹出窗口中将其名称修改为"START",用法选择为 "VAR\_INPUT",如图所示,点击"确定"。

腋点/线圈属性		X
名称(N): START ▼	定义范围 ⑥ 局部(L)	确定 取消
数据类型(D): BOOL ▼	局部变重组(V):	帮助(H)
用法(U): VAR_INPUT ▼ □ RETAIN 初值(I):	全局变望组(A): □-● 物理硬件 □-● 配置 □-● 認置 □-● Suntan Variables	
I/o地址(S): 描述(E):	Jysten variables	
■ PDD ■ OPC ■ 隐藏(X) □ 初值作为默认值.	☑ 显示工作单的所有变量(♥)	
<ul> <li>● 触点(I)</li> <li>类型(Y):</li> <li>⑦ 线圈(0)</li> </ul>	: -   - •	

图 7.6 修改触点名称和用法

4) 双击"C001"线圈,在弹出窗口中将其名称改为"RUN",用法选择为 "VAR\_OUTPUT",如图所示,点击"确定"。

触点/线圈属性		X
名称(N): RUN ↓ 數据类型(D): BOOL ↓ 用法(U): VAR_OUTPUT ↓ RETAIN 初值(I): I/O地址(S): 描述(E):	定义范围 ◎ 局部(L) 全局(G) 局部変量組(V): 全局変量組(V): 全局変量組(A): 小理硬件 → 物理硬件 → 部環 ○ 局部 System Variables	确定 取消 帮助(H)
<ul> <li>PDD □OPC □ 稳辙 (X)</li> <li>初值作为默认值.</li> <li>触点/线圈</li> <li>● 线圈 (0) 类型 (Y)</li> </ul>	<ul> <li>✓ 显示工作单的所有交量 (♥)</li> <li>: [-(-)▼</li> </ul>	

图 7.7 修改线圈名称和用法

5) 单击代码工作单中的"START"触点,然后选择快捷工具栏中的"并行",在其下面插入 一个并行的触点"C002",并将该触点改名为"RUN",如图所示。



图 4.7 插入并行触点

6) 点击"RUN"线圈,选择快捷工具栏中的"左触点",在左边插入一个触点"C003",如图 所示。



图 7.9 图 4.8 插入左触点
7) 双击触点"C004",在弹出窗口中将其名称改为"STOP",用法为"VAR\_INPUT",并选 择类型为常闭触点,如图所示。

触点/线圈属性		23
名称(N): STOP ◆ 數据类型(D): BOOL ◆ 用法(U): VAR_INFUT ◆ RETAIN 初值(I):	定义范围 ◎局部(L) 全局(C) 局部変量組(V): 一 Default ◆ 全局変量組(A): ● 物理硬件 ● 認置 ● 変源 ● 変源 ● System Variables	确定 取消 帮助(H)
H / OBLAI (3) · 描述(E) : □ PDD □ OPC □ 隐輩(X) □ 初值作为默认值.	☑ 显示工作单的所有交里 (¥)	
触点/线圈 ◎ 触点(T) ◎ 线圈(0) 类型(Y)	: <mark>-   /  -        </mark>	

图 7.10

- 8) 点击编译工作单。
- 9) 打开"MAIN"代码工作单,将"LDSample"功能块拖拽至工作单中,如图所示。



图 7.11 LDSample 功能块实例

# 7.1.4. 功能块图 (FBD) 编程语言

功能块图(Function Block Diagram,简称 FBD)编程语言实际上是用逻辑功能符号组成 的功能块来表达命令的图形语言,与数字电路中逻辑图一样,它极易表现条件与结果之间的 逻辑功能。这种编程方法是根据信息流将各种功能块加以组合,是一种逐步发展起来的新式 的编程语言,正在受到各种可编程控制器厂家的重视。由于具有图形符号、图形连接,且操 作方便,因此,该编程语言将被广泛采用。

示例:方波发生器,占空比 50%,周期可调。

- 1) 使用指令表编程语言示例中所建立的工程,在工程树 POU 节点中插入功能块 "FBDSample",语言选择"FBD"。
- 双击打开"FBDSample"的代码工作单,从右侧的"编辑向导"窗口中选择"功能块",把 "TON"功能块拖拽至其中,在弹出窗口中将其命名为"TON\_Front",点击"确定"。使用 同样的方法实例化第二个"TON"功能块,取名为"TON\_Back"。
- 3) 将鼠标移动至"TON\_Back"的"Q"输出端,如图所示。此时按住鼠标左键不放,将鼠标 移动到

"TON\_Front"的"IN"输入端,待连接线如图所示变为绿色后,松开左键。



图 7.12 鼠标悬停于"TON\_Back"的"Q"输出端



图 7.13 鼠标悬停于"TON\_Front"的"IN"输入端

- 4) 使用同样的方法连接"TON\_Front"的"Q"和"TON\_Back"的"IN"。
- 5) 双击"TON\_Front"功能块,在弹出窗口中的"形式参数"Tab 页中"IN"条目勾选"取了反 的",如图所示。点击确定,此时代码工作单中的"TON\_Front"的"IN"输入端就增加了 一个小圆圈,如图所示。

沵(N):		定义范围				确定
N_Front	-	⑧局部(L)		全局(G	)	The sets
据类型(D):		局部变量组(Ⅴ)	:			4X/H
N	*	Default			•	· 帮助(H)
去(U):						
R	RETAIN	📝 显示工作单的	的所有变	肈(₩)		
±(E):						
		归胞/ 归胞厌				
		man 2 (G): 12				
√式参数(Ⅲ):	40 <del>10</del> 46 III		60 24-20	. p.ia, rjata	吟涛/方	
/式参数(M): 名称	数据类型	取了反	的边沿	隐藏	隐藏值	
《式参数(M): 名称 到 IN ■ PT	数据类型 BOOL	取了反	的边沿	隐藏	隐藏值	
《式参数(M): 名称 到 IN 到 PT □ 00	数据类型 BOOL TIME BOOL	取了反	的边沿	隐藏	隐藏值	
《式参数()(1): 名称 到 IN 到 PT 日 Q ● ET	数据类型 BOOL TIME BOOL TIME	取了反 マ 	的边沿	· 隐藏	隐藏值	
<ul> <li>(机):</li> <li>名称</li> <li>IN</li> <li>PT</li> <li>Q</li> <li>ET</li> </ul>	教祖类型 BOOL TIME BOOL TIME	取了反 マ 	的边沿	隐藏 ————————————————————————————————————	隐藏值	
<ul> <li>/式参数(M):</li> <li>名称</li> <li>IN</li> <li>PI</li> <li>Q</li> <li>EI</li> </ul>	数据类型 BOOL TIME BOOL TIME	取了反 マ	的边沿	· 隐藏	隐藏值	
3式参数(M): 名称 到 IN 到 PT 到 Q ■ Q ET	数据类型 BOOL TIME BOOL TIME	取了反 マ 	的边沿	· 隐藏 - · · · · · · · · · · · · · · · · · · ·	隐藏值	
<ul> <li>(肌):</li> <li>名称</li> <li>11</li> <li>15</li> <li>16</li> <li>17</li> <li>19</li> <li>18</li> <li>18</li> </ul>	数据类型 BOOL TIME BOOL TIME	取了反	的	· 隐藏 □ □ □	隐藏值	(0187/2

图 7.14 对功能块(功能)的引脚进行取反



图 7.15 功能块的输入端已经取反

6) 在"TON\_Front"的输入端"PT"上点击右键,选择快捷菜单中的"变量"项,如图所示。在弹出窗口中的名称栏填入"HalfCycle\_Time",用法选择"VAR\_INPUT",点击"确定"。



图 7.16在代码工作单中声明变量

7) 连接"TON\_Back"的"PT"输入端和变量"HalfCycle\_Time",如图所示。



图 7.17 两个功能块的引脚连接同一个变量

- 8) 再插入一个变量"WaveOut",类型选择为"BOOL",用法为"VAR\_OUTPUT"。
- 9) 最后连接"WaveOut"到"TON\_front"的"Q"和"TON\_Back"的"IN"的连接线上,如图所示。



图 7.18连接输出变量

- 10) 点击编译工作单。
- 11) 打开"MAIN"的代码工作单,从右侧的"编译向导"窗口中将"FBDSample"拖拽至其中, 如图所示。



图 7.19 图 4.18 FBDSample 功能块实例

# 7.1.5. 结构化文本 (ST) 编程语言

结构化文本(Structured Text,简称 ST)是 IEC61131-3 标准中新定义的一种文本化 编程语言,其大量采用抽象性语句来描述复杂控制系统地功能,是一种高层语言。

结构化文本采用高度压缩的表达形式,具有和其他高级编程语言相同的程序流程控制能力,编写的代码结构清晰,适用于解决复杂的控制问题,特别适合已经熟悉像 C、 VB 这类高级语言的编程人员使用。但是由于 ST 语言与底层硬件结构无关,所以其需 要编译为机器代码才能执行,运行效率比其他底层语言低。

示例:求自然数的累加和阶乘。

1) 使用指令表编程语言示例中所建立的工程,在工程树 POU 节点中插入功能块 "STSample",语言选择"ST"。

名称	类型	用法	描述
Default			
SUM	INT	VAR_OUTPUT	
FACTORIAL	INT	VAR_OUTPUT	
BEGIN_NUM	INT	VAR_INPUT	
END_NUM	INT	VAR_INPUT	
1	INT	VAR	-

2) 在"STSample"的变量工作单中输入如图所示的变量。

图 7.20 功能块"STSample"的变量声明

3) 在"STSample"的代码工作单输入如图所示代码。

1	SUM:=0; FACTORIAL:=1;
2	FOR I:= BEGIN_NUM TO END_NUM BY 1 DO
3	SUM := SUM + I;
4	FACTORIAL := FACTORIAL * I;
5	END_FOR;

图 7.21 功能块"STSample"的代码实现

- 4) 点击编译工作单。
- 5) 打开"MAIN"的代码工作单,从右侧的"编译向导"窗口中将"STSample"拖拽至其中, 如图所示。





## 7.1.6. 顺序功能图 (SFC) 编程语言

顺序功能图(Sequence Function Chart,简称 SFC)最早是由法国国家自动化促进会

(Graphe de Commande Etape-Transistion, GRADCET)提出的,针对顺序控制系统的控

制条件和过程的一套表示逻辑控制功能的方法。由于其结构严密,流程清晰,简单易学,有 利于设计人员和其他专业人员的交流沟通,所以自公布不久,就成为许多国家和国际标准, 如 IEC 60848、GB/T26988.6-1993。

但在 IEC61131-3 标准中,顺序功能图并没有作为一种单独的编程语言被定义,而是作为编程语言的公共元素。在顺序功能图中,描述顺序控制系统的过程、功能和特性时采用了 文字叙述和图形符号相结合的方法。其即可以被理解成文本类语言,也可以当成图形类语言 使用,按照通常的归类,其还是属于图形类编程语言。

虽然顺序功能图能对控制问题进行清晰的描述,但是其是文本与图形相结合的语言,而 且 SFC 具有固有的逻辑控制框架,语言代码体积较大,占用的存储空间较多,运行期间所 需的硬件资源较多,不适合解决简单问题和使用于小型控制器。 示例:交通灯控制程序。

交通灯有一个控制开关,当开关合上后,交通灯进入自动切换状态。自动状态下,首先南北方向红灯点亮 13s,同时东西方向的绿灯点亮 8s,然后东西绿灯闪烁 3s,东西绿灯熄 灭,黄灯点亮 2s。接着自动切换,南北红灯熄灭,东西黄灯熄灭,东西方向红灯点亮 15s,南北绿灯点亮 10s,接着南北绿灯闪烁 3s,之后南北绿灯熄灭,黄灯点亮 2s。

1) 右键点击工程树窗口中的"逻辑 POU"节点,在弹出菜单中选择插入"功能块"。

2) 在弹出对话框中添加名称"TrafficLight",语选择"SFC",点击确定,如图所示。

		UH LE
类型	语言	取消
<ul> <li>程序(G)</li> <li>功能(U)</li> <li>可能快(B)</li> <li>动作</li> <li>转換</li> <li>SFC歩</li> <li>工作単</li> </ul>	<ul> <li>□ IL</li> <li>□ ST</li> <li>□ SFC</li> <li>□ FBD</li> <li>□ LD</li> <li>□ FFLD</li> <li>□ MSFC</li> <li>□ VAR</li> <li>□ 数据类型(Y)</li> <li>□ 描述(C)</li> </ul>	帮助(H) ● 使用保留(R) 模式 ● 插入(I) ● 追加(A)

图 7.23 插入功能块

 打开"TrafficLight"的变量工作单,在其中添加 7 个"BOOL"类型变量,一个为 "VAR\_INPUT",名称为"启动"。其余为"VAR\_OUTPUT",名称分别为"南北红"、"南北 绿"、"南北黄"、"东西红"、"东西绿"、"东西黄",如图所示。

_			
	名称	类型	用法:
	Default		
	启动	BOOL	VAR_INPUT
	南北红	BOOL	VAR_OUTPUT
	南北绿	BOOL	VAR_OUTPUT
	南北黄	BOOL	VAR_OUTPUT
	东西红	BOOL	VAR_OUTPUT
	东西绿	BOOL	VAR_OUTPUT
	东西黄	BOOL	VAR_OUTPUT

图 7.24 声明功能块"TrafficLight"的变量

- 4) 打开"TrafficLight"的代码工作单,在其中编辑 SFC 网络。
- 5) 点击快捷工具栏中的"创建步转换序列"按钮插入一个最简 SFC 网络到工作单中,如 图所示。



图 7.25 插入最简 SFC 网络

"创建步转换序列"右边一个按钮是"创建动作"按钮,在选中动作或者步的情况下,可以 增加一个动作条目。

"创建步转换序列"右边第二个按钮是"插入并行/选择分支"按钮。在选中转换的情况下 点击,则会在该转换下添加一个 2 条支路的并行结构。如果在选中步的情况下点击,则 会在该步下添加 2 条支路的选择结构。

"创建步转换序列"右边第三个按钮是"插入 SFC 分支"按钮,点击该按钮可以手动绘制 SFC 分支。

6) 绘制如图所示的 SFC 网络。



图 7.26 交通灯 SFC 网络

7) 双击延时 8s 转换,打开图形工作单,在其中编写 FBD 程序,如图所示。



图 7.27 延时 8s FBD 语言实现

"东西绿灯.x"表示步的活动状态。"延时 8s"是转换的名称,同时也表示转换的布尔结果。 8) 双击延时 3s 转换,打开代码编辑单,在其中编写 ST 程序。

对于其他转换,可以采用 LD、ST、IL、FBD 四种语言中的任何一种实现,这里不再演

TON\_2(IN:=东西绿闪.x,PT:=T#3s);

延时 3s:=TON\_2.Q;

示。

9) 双击打开动作"闪烁东西绿灯",使用 FBD 语言编写程序,如图所示。



图 7.28 动作"闪烁东西绿灯"的实现

对于动作"闪烁南北绿灯"也可以采用类似的方法,此处不再详述。

- 10) 在完成所有代码后,编译功能块"TrafficLight"。
- 11) 打开 SFC\_1 图形代码工作单,在其中加入刚才编译通过的 TrafficLight 功能块。在 输入输出端连接相应的全局变量,如图所示。

TrafficLight_1			
1	Traffi	icLight	
START	启动	南北红	—NSR
		南北绿	—NSG
		南北黄	—NSY
		东西红	—EWR
		东西绿	—EWG
		东西黄	—EWY
			ļ.

图 7.29 交通灯程序实现

12) 右击工程树中"资源: eCLR\_Simulation"节点,选择"设置"菜单,在如图所示的弹出对 话框中选择通信类型为"仿真 1",建立设置为"eCLR 2.2 Simulation (Core: 2.8.1)"。

100 100		
通信		
类型:	仿真 1	•
恭教:		
版本		
建立设置:	eCLR 2.2 Simulation (Core: 2.8.1)	•]
<ul> <li>回 更新前i</li> <li>回 不更新前i</li> </ul>	() (4) 時间(8) N)	
在线更新		
です。 在线更新 时间间隔:	10 茶砂 (范国: 0.60000)	
6 中立30 在线更新 时间间隔: 编译器选项	10 養砂 (范围: 0.60000)	
在线更新 时间间隔: 编译器选项 一堆栈检	10 爱州 (武王) (3.6000)	
<ul> <li>中式加新</li> <li>时间间隔:</li> <li>備講:</li> <li>満译器迭项</li> <li>単状检:</li> <li>マ 教組辺:</li> </ul>	10 爱护 (范围: 0.60000) 更 转级查	

- **13)** 点击确定后,制作整个工程。点击工程控制按钮。在工程控制对话框中点击"下装"按钮。
- 14) 点击"冷启"按钮后关闭工程控制对话框,并点击"调试"按钮,既可以在线观察交通灯 的运行状态。
- 7.2. 运动控制功能

# 7.2.1. 单轴位置控制

## 7.2.1.1. 动作概要

MC 功能模块的单轴控制功能包括指定动作曲线的控制和同步控制。 在需指定动作曲线的控制中,可通过位置控制、速度控制2种控制模式进行指定。

# 7.2.1.2. 绝对值定位

指定以原点为基准的绝对坐标目标位置的定位。



图 7.31

功能的详情请参阅"leetro i5-1213 控制器指令手册"的 MC\_MoveAbsolu(绝对位置运动) 指令。

#### 7.2.1.3. 相对值定位

指定与当前位置之间的距离的定位。



功能的详情请参阅"leetro i5 控制器指令手册"的相对位置运动指令 MC\_MoveRelative。

#### 7.2.1.4. 周期同步定位

周期同步定位是按任务周期向轴输出指定目标位置的功能。目标位置以绝对位置指定。 用于需按用户创建的任意轨迹动作的场合等。



# 图 7.33

功能的详情请参阅"leetro I5 控制器指令基准手册"的 MC\_SyncMoveAbsolute(周期同步 绝对位置运动)指令。

# 7.2.1.5. 停止

停止轴动作的功能包括通过限位输入信号和用户程序的运动控制指令停止的功能以及发生异常等后停止。

#### 限位停止输入停止方法(正方向极限输入或负方向极限输入)

根据伺服驱动器的输入信号状态,在 MC 功能模块中执行立即停止处理。

#### 通过运动控制指令停止

需停止单轴动作时,应使用 MC\_Stop(减速停止)指令或 MC\_ImmediateStop(立即停止)指令。

# ● MC\_Stop( 减速停止) 指令

对于非周期同步运动,按指定的减速度减速停止。对于周期同步运动,轴运动将立即停止。

如果减速度指定为"0",则立即停止运动。

通过本指令执行减速停止时,不会响应其他运动指令。但在减速停止过程中,可通过 MC\_ImmediateStop 指令立即停止减速过程。

# ● MC\_ImmediateStop( 立即停止) 指令

可对单轴控制功能执行立即停止。

功能的详情请参阅"leetro I5 系列控制器指令手册"的 MC\_Stop(强制停止)指令、 MC\_ImmediateStop(立即停止)指令。

# 发生异常后等停止

#### ● 单轴动作中发生异常后停止

单轴动作中发生异常时,根据异常内容立即停止或减速停止。

# ● 发生导致伺服OFF的异常

发生导致伺服 OFF 的异常时立即停止。伺服 OFF 时伺服驱动器的动作取决于伺服驱动器的设定。

# 停止方法



# 停止的优先顺序

停止的优先顺序如下表所示。停止过程中如果发生优先顺序较高的停止,则切换至优先 顺序较高的停止方法。

停止方法	优先顺序 (数字越大优先度越高)	
立即停止	2	
减速停止	1	

# 7.2.1.6. 超调

通过 MC\_SetOverride(设定超调值)指令,设定启动中的轴动作相应的超调值的功能。 超调值是目标速度相应的比率,可在 0 ~ 200%的范围内设定。 如果相对于目标速度设定为 0%,则以速度"0"停止,保持动作状态。 启用超调时,导入设定的超调值。 变更后的超调值超过最高速度时,则变为各轴的"最高速度"。

# ● 对MC\_MoveAbsolute(绝对值定位)指令的超调



图 7.34 功能的详情请参阅"leetro I5 系列控制器指令手册"的 MC\_SetOverride (速度控制)指令。

# 7.2.2. 单轴速度控制

# 7.2.2.1. 速度控制

速度控制是指以指定速度连续移动轴的功能。

也可指定加速度、减速度及跃度。

需停止轴时,应使用 MC\_Stop、MC\_ImmdiateStop 指令或启动其他运动指令。

目标速度指定为"0"时,轴不会移动,但轴处于正在动作的状态。

速度控制过程中,如果多重启动其他运动控制指令,则按指令的降速加速度减速停止后 切换到其它运动指令的动作。



图 7.35

功能的详情请参阅"leetro I5 系列控制器指令手册"的 MC\_MoveVelocity(速度控制)指令。

#### 7.2.2.2. 周期同步速度控制

将伺服驱动器的控制模式作为速度控制模式,按控制周期输出指令速度的功能。 需停止轴时,应使用 MC\_Stop、MC\_ImmdiateStop 指令或启动其他动作指令。 目标速度指定为"0"时,轴不会移动,但轴处于正在动作的状态。

伺服驱动器中执行速度控制循环以接受指令,因此即使因外部干扰等导致比指令速度慢, 也不会发生速度变动以消除位置偏差。

功能的详情请参阅"leetro I5 系列控制器指令手册"的 MC\_SyncMoveVelocity (周期同步 速度控制)指令。

MC\_SyncMoveVelocity命令



图 7.36

# 7.2.3. 单轴控制的通用功能

# 7.2.3.1. 位置

位置的种类

MC功能块中包含有以下2种位置。

位置类型	定义
指令位置	用于控制轴的由MC功能块输出的位置
反馈位置或实际位置	从伺服驱动器或计数器输入的实际位置

指令位置与反馈位置的关系如下图所示。



图 7.37

指令单位与脉冲单位之间通过主站电子齿轮比(脉冲当量)进行转换。公式如下: 脉冲单位的位置 = 指令单位的位置 × 主站电子齿轮比

指令位置与反馈位置的以下项目相同

项目	指令位置	反馈位置
计数模式	线性模式	与指令位置的计数模式相同
位置的单位	从脉冲、毫米、微米、纳米 、度、	与指令位置的单位相同
	英寸中设定任意一个	
软件限位	设定软件可操作的范围	与指令位置的范围相同
当前位置变更	将当前位置变更为任意位置(通过	同时设定与指令位置相同的值
	MC_SetPosition实现)	
原点复位	具有原点复位前和复位后的状态	与指令位置的原点状态相同

#### 位置相关的轴参数

参数名称	功 能	设定范围	初始值
定位检查时间	保留。以毫秒为单	0~10000	0
	位,设定定位完成		
	的检查时间。		
跟随误差检查使能	检查当前跟随误差	0, 1	1
	是否超过最大允许		
	跟随差		
最大允许跟随误差	设置最大跟随误差	大于0的32位浮点数	100
位置偏差报警时间	保留。设定超过最	16位无符号数	2
	大跟随误差的允许		
	时间,单位为毫秒		
软件正限位	设定正方向侧的软	$0 \sim 2147483647$	2147483647
	件限制值。(单位:		
	指令单位)		
软件负限位	设定负方向侧的软	-2147483647~0	-2147483647
	件限制值。(单位:		
	指令单位)		

# 轴动作的目标位置指定

通过运动控制指令的输入变量"Position(目标位置)"、"Distance(移动距离)"指定实际 定位动作的位置和距离。

# 位置的监控

可通过指令MC\_ReadActualPosition读取轴的反馈位置(指令单位),通过 MC\_ReadParameter读取轴的指令位置(指令单位);或在MultiProg中监控脉冲单位的如下 变量:

变量名称	数据类型	名称	功能
0x6064h_00h	I32	当前实际位置	反馈位置的当前值,脉冲单位

# 7.2.3.2. 速度

## 速度的种类

MC功能模块使用的轴的速度分为以下2种。

速度种类	定义
指令速度	用于控制轴的由MC 功能模块输出的速度。
反馈速度或实际速	根据通过伺服驱动器和计数器输入的反馈位置,在MC 功能模块内
度	计算出的速度。

#### 速度的单位

速度的单位为"指令单位/s"。

指令单位与脉冲单位之间通过脉冲当量进行转换。公式如下:

脉冲单位的位置 = 指令单位的位置 × 电机转一周指令脉冲数 / 电机转一周工作行程

#### 速度相关的轴参数

参数名称	功能	设定范围	初始值
最大速度	限制轴的最大速度,指	大于零的32位浮点数	4E+08
	令单位		
最大启动速度	设置轴的启动速度,指	大于零的32位浮点数	100
	令单位		
最大点进速度	<b>保留</b> ,指令单位	大于零的32位浮点数	1000000

# 轴动作的目标速度指定

实际的定位动作的速度通过运动控制指令的输入变量"Velocity(目标速度)"指定。

# 速度监控

可通过MCIF\_ReadActualVelocity 指令读取轴反馈速度(指令单位),通过 MC\_ReadParameter读取轴的指令速度(指令单位);或通过MultiProg程序读取轴指令速度 和反馈速度(脉冲单位)。

变量名称	数据类型	名称	功能
0x60FFh_00h	I32	当前目标速度	指令速度的当前值,脉冲单位。
			向正方向移动带正号,向负方向移动带
			负号。
0x606Ch_00h	I32	当前实际速度	反馈速度的当前值,脉冲单位。
			向正方向移动带正号,向负方向移动带
			负号。

## 7.2.3.3. 加速与减速

## 加速度与减速度的单位

加速度和减速度的单位为"指令单位/s²"。

指令单位是通过位置的脉冲当量进行单位转换后的值。

111111111111111111111111111111111111111

参数名称	功能	设定范围	初始值
最大加速度	设定轴的最大加速度	大于0的32位浮点数	1000000
最大减速度	设定轴的最大减速度	大于0的32位浮点数	1000000

#### 轴动作的加速度和减速度的指定

通过运动控制指令的输入变量"Acceleration(加速度)"、"Deceleration (减速度)"指定实际定位动作的加速度和减速度。

#### 加减速动作示例

对于速度模式运动,如MC\_MoveVelocity,指令执行过程中,加减速动作示例如下图 所示。



对于位置模式运动,如MC\_MoveAbsolute,当指定的移动量较小或加减速度较低时,可能无法达到目标速度。

#### 7.2.3.4. 运动控制指令的重启

下面对单轴动作中变更指令输入变量并重启进行说明。

指令的重启针对的是启动参数为Execute的指令。这类指令执行过程中,使输入变量 Execute由TRUE->FASLE->TRUE变化时,即指令执行过程中再次输入上升沿信号,称为指 令的重启。对于支持重启的指令,将按变更后输入变量的值进行动作。目前单轴指令支持 的重启操作的有: MC\_MoveVelocity、MC\_Stop、MC\_WriteParameter。

重启运动指令可变更输入变量。

如果指令不支持重启,则丢弃重启操作,指令按原参数执行。

如下图所示,指令 MC\_MoveVelocity 执行重启的时序。



图 7.39

功能的详情请参阅"leetro I5 系列控制器指令基准手册"。

# 7.2.3.5. 运动控制指令的多重启动(缓存模式)

多重启动在 PLCopen 技术规格书中定义为功能块的缓存模式,指轴在某个指令控制的动作中,又启动其他运动指令。

使用多重启动后,可连续、逐次执行多个运动控制指令。

在 MC 功能模块中多重启动使用以下用语。

用语		含义
本手册	PLCopen	
当前指令	以前的 FB	执行多重启动前动作的运动控制指令
缓存指令	下一个 FB	轴动作中启动并处于执行等待状态的运控控制指令

指定运动控制指令的输入变量 "BufferMode(选择缓存模式)" 后可选择以下多重启动 模式。

上述模式的主要差异在于多重启动指令的执行开始时间和中继速度。

多重启动模式	动作说明
中断 (Aborting)	中断当前指令,执行多重启动的运动控制指令
等待(Buffered)	当前指令的动作正常结束后执行缓存指令。

等待时,新启动的指令(缓存指令)不立即执行,而是存储在缓冲内,等待当前指令正常执行结束后再启动。各轴可缓存 1 条指令。

中断时,先停止当前指令,之后启动新指令。

使用注意事项



等待

执行中的当前指令的动作完成前,等待执行多重启动的指令。当前指令的动作正常结束后执行缓存指令。

# 中断

默认模式。

中断执行中的指令,等待当前指令结束后,启动多重启动的指令。 可通过单轴控制和周期同步控制的运动控制指令多重启动中断。

如下图所示,多重启动两个 MC\_MoveAbsolute 指令,缓存模式设置为中断(Aborting)的时序图。



图 7.40

## 7.2.4. 轴组运动控制

在使用轴组指令前,事先设定轴组,可进行多轴的插补控制。

# 7.2.4.1. 动作概要

轴组运动控制是为了控制被控对象的轨迹,将多个相关轴进行组合、实现插补运动的 功能。

在 MC 功能模块中,将执行插补运动的轴组合设定为轴组。

通过用户程序将各轴设为伺服 ON 后, 启用实际执行插补运动的轴组。

轴组控制以轴组所属的轴的插补运动为目的,因此无法对轴组中的轴启动单轴动作指 令。轴组所属的轴发生异常时,轴组中所有轴均停止运动。

MC 功能模块中可执行 2 轴~ 4 轴的直线插补和 2 轴的圆弧插补(圆弧插补还未正式发 布)。

## 启用/禁用轴组

需启用轴组时,应通过 MC\_GrpEnable 指令指定需启用的轴组并使能。如果在禁用状态下启动其它轴组指令,则指令会发生异常并无法启动。使能轴组后,不能对轴组中的各轴执行单轴运动指令。轴必须处于停止状态才能调用启用轴组指令。

需禁用处于启用状态下的轴组时,应通过 MC\_GrpDisable 指令指定需禁用的轴组。禁用后,可使用单轴运动控制指令启动轴运动。

如果在轴组动作中启动 MC\_GrpDisable 指令,则 MC\_GrpDisable 发生异常。必须等待轴组运动停止后调用禁用轴组指令。



功能的详情请参阅"leetro I5 系列控制器指令基准手册"中的 MC\_GrpEnable 指令和 MC\_GrpDisable 指令。

# 轴组位置获取

通过 MC\_GrpReadActPos(获取轴组位置)指令获取轴组中各轴当前的反馈位置。

#### 轴组错误复位

轴组中各个从站轴(伺服)发生异常时,通过调用 MC\_GrpReset(轴组错误复位)指 令解除异常。 轴组错误复位的详情请参阅"leetro I5 系列控制器指令基准手册"中的 MC\_ GrpReset (获取轴组位置)指令。

# 7.2.4.2. 直线插补

可执行绝对值定位和相对值定位,可指定最大插补速度、最大插补加速度、最大插补 减速度。

MC 功能模块包括以下2 种直线插补指令。

- MC\_MoveLinAbs: 绝对位置直线插补
- MC\_MoveLinRel: 相对位移直线插补

控制2轴(X、Y)由A 点至B 点的直线插补如下图所示。



直线插补的详情请参阅"leetro I5 系列控制器指令基准手册"。

# 7.2.4.3. 圆弧插补

圆弧插补使用轴组中 2 轴,在二维平面中进行圆弧运动定位。 可执行绝对位置定位和相对位移定位,可指定圆弧最大插补速度、最大插补加速度、 最大插补减速度。

#### 7.2.4.4. 轴组周期同步位置控制

该功能周期性输出为轴组中各轴指定的目标位置,应以绝对位置指定出目标位置。

轴组周期同步位置控制功能的详情请参阅"leetro I5 系列控制器指令基准手册"中的 MC\_GrpSyncMoveAbs 指令。

#### 7.2.4.5. 插补运动中的停止

停止轴组插补运动的方式包括调用轴组停止指令、异常引起的停止。

#### 通过运动控制指令停止

需停止轴组的动作时,应使用 MC\_GrpStop(轴组强制停止)指令 MC\_GrpImmediateStop(轴组立即停止)指令。

#### ● MC\_GrpStop 指令

可对轴组中执行的直线插补和圆弧插补指定减速度和跃度,使其在指定轨迹上减速停止。

如果减速度指定为"0",则将立即停止的指令发送至伺服驱动器。 通过本指令执行减速停止时,无法响应其他运动指令。

# MC\_GrpImmediateStop 指令 可立即停止轴组中的所有轴运动。 也可对执行 MC\_GrpStop 指令后处于减速停止状态的轴组执行。

#### 发生异常等后停止

● 轴组动作中发生异常后停止

轴组运动过程中,轴组中的组成轴遇到运动方向的限位信号,将立即停止轴组运动。 异常停止执行过程类似于执行 MC\_GrpInmediateStop 指令。

#### 7.2.4.6. 多轴协调控制中的超调

在轴组中执行多轴协调控制时,可通过 MC\_GrpSetOverride(设定轴组超调值)指令 对启动中的插补动作设定超调值。

超调值是插补目标速度相应的比率,可在0~200%的范围内设定。

如果相对于插补目标速度设定为0%,则以速度"0"停止,保持动作状态。

变更后的超调值超过最高插补速度时,则执行轴组参数的"最高插补速度";小于0 按0执行。

功能的详情请参阅"leetro I5 系列控制器指令基准手册"中的 MC\_GrpSetOverride 指令。

#### 7.2.5. 多轴协调控制的通用功能

# 7.2.5.1. 多轴协调控制中的速度

多轴协调控制时以轨迹上的插补速度指定速度。单位与单轴时相同,使用"指令单位/s"。

#### 速度相关的轴参数

参数名称	功能	设定范围	初始值
最高插补速度	设定轨迹的最高插补速度。	大于0的32	80000
	通过轴组运动指令指定了超过最高	位浮点数	
	插补速度的插补目标速度时,以最		
	高插补速度进行动作。		

#### 设定轴动作的目标速度

实际的运动的插补速度通过运动控制指令的输入变量"Velocity(目标速度)"指定。

#### 7.2.5.2. 插补运动中的加速和减速

插补运动时以指令设定的插补加速度和插补减速度控制升降速。轴组加速度单位与单轴时相同,使用"指令单位/s<sup>2</sup>"。

#### 插补加速度和插补减速度相关的轴参数

参数名称	功能	设定范围	初始值
最大插补加速度	设定轨迹的最大插补加速度。单	大于0的32	10000
	位: 指令单位/s <sup>2</sup>	位浮点数	
最大插补减速度	设定轨迹的最大插补减速度。单	大于0的32	10000
	位: 指令单位/s <sup>2</sup>	位浮点数	

#### 设置轴组动作的插补加速度和插补减速度

通过轴组运动控制指令的输入变量"Acceleration(升速加速度)"、"Deceleration (降速加速度)"指定实际定位动作的插补加速度和插补减速度。

# 7.2.5.3. 多轴协调控制的运动控制指令的重启

重启直线插补或圆弧插补指令后,指令不响应重启操作,仍执行原来的运动。

#### 7.2.5.4. 多轴协调控制的运动控制指令的多重启动(缓存模式)

与轴动作相同,在轴组中也可多重启动多轴协调控制。 如果在插补运动中使用多重启动,则可执行多个直线和圆弧连续的轨迹控制。 指定运动控制指令的输入变量"BufferMode" 后可选择与单轴时相同的多重启动模

式。

轴动作的指令和轴组动作的指令无法相互多重启动。 轴组运动多重启动时序与单轴运动相同。

#### 中断

指令的默认模式。 中断执行中的指令,停止后执行多重启动的新指令。 单轴运动控制指令和轴组运动控制指令无法相互执行中断。 如果对单轴运动中的轴启动轴组动作,轴组指令会发生异常并停止。 如果在轴组动作中,对轴组中的某轴启动单轴运动指令,则单轴运动指令会发生异常 并停止。

#### 等待

执行中的动作完成前,等待执行多重启动的指令。 当前指令的动作正常结束后执行缓存指令。

# 7.2.6. 原点复位

MC\_Home 指令启动回原点运动(或称回零运动)及设置回远点结束后系统所处的坐标位置。

回原点运动的参数,包括回原点模式、回原点速度、近点速度、原点返回加速度、电机最大速度。 通过【轴设置】->【原点返回设置】 设置原点模式、回原点速度、近点速度、原点返回加速度这4个参数。在【轴设置】->【伺服驱动设置】设置 "电机最大速度"。

回零方式一共有 37 种,注意:回零方式一定不能设置成 0,否则将导致电机无法正常 执行回零操作。不同的伺服,轴模块等产品支持的回零方式有所不同,请对照伺服说明手 册,选择正确的回零模式。

回零方式号	说明
0	No homing method assigned
1	-Ve LS & Index Pulse
2	+Ve LS & Index Pulse
3	+Ve HS & Index Pulse direction reversal
4	+Ve HS & Index Pulse no direction change
5	-Ve HS & Index Pulse direction reversal
6	-Ve HS & Index Pulse no direction change
7	On +Ve HS –Index Pulse
8	On +Ve HS +Index Pulse
9	After +Ve HS reverse +Index Pulse
10	After +Ve HS +Index Pulse
11	On -Ve HS –Index Pulse
12	On -Ve HS +Index Pulse
13	After -Ve HS reverse +Index Pulse
14	On -Ve HS +Index Pulse
15	保留
16	保留

# 37 种回零方式

17	Same as 1 without Index Pulse
18	Same as 2 without Index Pulse
19	Same as 3 without Index Pulse
20	Same as 4 without Index Pulse
21	Same as 5 without Index Pulse
22	Same as 6 without Index Pulse
23	Same as 7 without Index Pulse
24	Same as 8 without Index Pulse
25	Same as 9 without Index Pulse
26	Same as 10 without Index Pulse
27	Same as 11 without Index Pulse
28	Same as 12 without Index Pulse
29	Same as 13 without Index Pulse
30	Same as 14 without Index Pulse
31	保留
32	保留
33	On Index Pulse +Ve direction
34	On Index Pulse -Ve direction
35	Current positon = home
36	保留
37	Current positon = home

备注:

+Ve: positive 方向

-Ve: negative 方向

LS: Limit switch

HS: Home switch

Index Pulse: 编码器 Z 脉冲

方法1

·此方法是,如果未激活负限位开关,初始化动作方向是负方向。

(图示为低电平状态下非激活状态。)

·原点检出位置是负限位信号为非激活后的在正方向侧位置的最初的索引脉冲检出位置。



图 7.42

#### ・方法2

- ·此方法是,如果未激活正限位开关,初始化动作方向是正方向。 (图示为低电平状态下非激活状态)
- ·原点检出位置是正限位信号为非激活后的在负方向侧位置的最初的索引脉冲检出位置。 (请参照下图)



图 7.43

方法3,4

- ・此方法是,基于启动时的原点开关的状态初始化动作方向变化。
   ・原点检出位置是原点开关的状态变化后的负方向侧,或者负方向侧最初的索引脉冲检出位置。 (请参照下图)



图 7.44

・方法 5,6

·此方法是,基于启动时的原点开关的状态初始化动作方向变化。

•原点检出位置是原点开关的状态变化后的负方向侧,或者正方向侧最初的索引脉冲检出位置。 (请参照下图)





・方法 7, 8, 9, 10

- ·此方法是,使用原点开关盒索引脉冲。
- ·方法 7, 8 的初始动作方向是原点开关如果在动作开始时已经激活,则为负方向。
- ·方法 9, 10 的初始化动作方向是原点开关如果在动作开始时已经激活,则为正方向。
- ·原点检出位置是,原点开关的上升沿或者下降沿附近的索引脉冲。

(请参照下图)



图	7.46
---	------

・方法11,12,13,14

- ·此方法是,使用原点开关和索引脉冲。
- ·方法 11, 12 的初始化动作方向是原点开关如果在动作开始时已经激活,则为正方向。
- ·方法 13, 14 的初始化动作方向是原点开关如果在动作开始时已经激活,则为负方向。
- ·原点检出位置是,原点开关的上升沿或者下降沿附近的索引脉冲。
- (请参照下图)



图 7.47

- 方法 17
  - ・此方法是和方法1相似。
     不同的是,原点检出位置不是索引脉冲,而是限位开关变化的位置。
     (请参照下图)



图 7.48

・方法 18

・此方法和方法2相似。
 不同的是,原点检出位置不是索引脉冲,而是限位开关变化的位置。
 (请参照下图)



图 7.49

- ・方法 19,20
- ・此方法和方法 3,4 相似。
   不同的是,原点检出位置不是索引脉冲,而是原点开关变化的位置。
   (请参照下图)



图 7.50

・方法 21, 22

・此方法和方法 5, 6 相似。
 不同的是,原点检出位置不是索引脉冲,而是原点开关变化的位置。
 (请参照下图)



图 7.51

・方法23,24,25,26

・此方法和方法 7, 8, 9, 10 相似。
 不同的是,原点检出位置不是索引脉冲,而是原点开关变化的位置。
 (请参照下图)



图 7.52

・方法 27, 28, 29, 30

・此方法是和方法 11, 12, 13, 14 相似。
 不同的是,原点检出位置不是索引脉冲,而是原点开关变化的位置。
 (请参照下图)



图 7.53

- ・方法 33,34
  - 此方法是只使用索引脉冲。

·在图中所示方向动作后检出索引脉冲作为原点检出位置。



图 7.54

・方法 35, 37

・在执行伺服驱动器的坐标系的设定(位置信息的设定)时使用。
 在原点复位启动时的点,以此位置为基准初始化(预置)下述的对象。

6062h(Position demand value) = 6064h(Position actual value) = 607Ch(Home offset)

6063h(Position actual internal value) = 60FCh(Position demand internal value) = 0

(注)607Ch(Home offset)被加算到6062h以及6064h中。

· PDS 状态不是操作使能状态,也可以执行。

·方法 35 和 37 是相同的功能,但是新规设计时基于 ETG 规格,请使用方法 37。



图 7.55

# 7.2.7. 其他功能

#### 7.2.7.1. 当前位置变更功能

将伺服轴的指令当前位置变更为任意值,将反馈当前位置变更为与指令当前位置保持 位置偏差的值的功能。

使用MC\_SetPosition(变更当前位置)指令,指定需变更的当前位置。 可在轴动作中变更当前位置。但不能中停止运动指令执行中变更当前位置。 在轴运动中变更当前位置,只改变轴坐标位置,不变更移动的距离。

#### ● 轴停止状态下执行时的时序图



图 7.56

MC\_SetPosition(变更当前位置)指令的详情请参阅"leetro I5系列控制器指令手册"。

# 7.2.7.2. 锁定功能

当伺服驱动器接收到有效的触发信号时,自动记录触发时轴的位置,也称为锁存轴当前位置,通过MC\_TouchProbe(启用锁定)指令可反馈出该锁存位置。

锁存信号参数通过MC\_TouchProbe指令设置。触发信号是一个结构体变量,包括:设备选择(驱动器锁存、控制器锁存)、锁存器ID(每个轴有两个锁存器)、触发时间模式(单次锁存、连续锁存)、触发源(输入口、Index脉冲)、触发信号电平(上升沿触发、下降沿触发)。

当MC\_TouchProbe指令处于激活状态但需要终止锁定功能时,应使用MC\_AbortTrigger (禁用外部锁定)指令。

锁定功能的详情请参阅"leetro I5系列控制器指令手册"中的MC\_TouchProbe指令和 MC\_AbortTrigger指令。

#### 7.2.7.3. 软件限制功能

除硬件的限位输入信号输入以外,还可设置软件限位。

软件限位的监控范围由【轴设置】->【限位设置】中"正软件限位"和"负软件限 制"设定。



图 7.57

#### 软件限制功能相关的轴参数

参数名称	功能	设定范围	初始值
软件正限位	设定正方向的软件限制值。	32位浮点数,	2147483647
	单位为指令单位。	负数、正数、0	
软件负限位	设定负方向的软件限制值。		-2147483647
	单位为指令单位。		

根据轴的动作状态和运动控制指令,软件限制在以下2种情况下启动。

# ■ 启动运动指令时

·当前位置在软件限制范围内时,目标位置超过软件限制范围时指令会发生异常。 ·当前位置超过软件限制范围时,可向软件限制范围内移动。此时的目标位置需在软件 限制范围内。

#### ■ 轴动作时

定位动作中、连续动作中、多轴协调动作中的轴动作时,指令位置从在范围内变为超过范围时轴会发生异常。

# 7.2.7.4. 位置偏差监控功能

轴的指令位置与反馈位置之差为位置偏差。MC 功能模块按运动控制周期监控位置偏差。位置偏差值超过轴参数的"最大允许跟随误差"的时长大于 "位置偏差报警时间"时,指令会输出错误。通过原点复位执行接触动作时,位置偏差监控功能无效。

# 7.3. 运动控制功能块库的使用

# 7.3.1. 运动控制功能块库的添加

1) 打开"工程树窗口"。

● 章 数据类型 ● 章 逻辑POV Untitled*	
(学工程 ) 硬件	

图 7.58

2) 鼠标右键单击【库】。在弹出菜单中选择【插入】->【固件库】。弹出如下对话框:

💶 包含库			×
查找范围(I):	🕌 FW_LIB 👻 🧿 🕻	ۇ 📂 🛄 ד	
Ca.	名称	修改日期	1
最近访问的位置		2018/8/20 16:48	2
	MS_PLCopen_LIB	2018/8/20 16:48	
桌面		2018/8/20 16:48	1
「」 库	ProConUS	2018/8/20 16:48	1
して して して して して して して して して して			
<b>(</b> ) 网络			
	<	·	•
	文件名 00): 六(小水川) (万川) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7) - (7	▼ 包括(C)	
	又扞奕型(I):	取消	

图 7.59

3) 鼠标双击【MS\_PLCopen\_LIB】打开该文件夹,单击【包括】按钮;即可将固件库导入到 工程中。
| 🔹 包含库                                  |                     |                                 |     |                 | ×        |
|----------------------------------------|---------------------|---------------------------------|-----|-----------------|----------|
| 查找范围(I):                               | 🌗 MS_PLCopen_       | LIB                             | - 0 | • 🖅 🥙 🏂         |          |
| <u>G</u>                               | 名称                  | *                               |     | 修改日期            | 1        |
| 最近访问的位置                                | MS_PLCope           | en_LIB.FWL                      |     | 2018/6/28 16:16 | F        |
| 桌面                                     |                     |                                 |     |                 |          |
| <mark>」</mark> 库                       |                     |                                 |     |                 |          |
| (人)                                    |                     |                                 |     |                 |          |
| rr <i>⇔</i> ₩6                         |                     |                                 |     |                 |          |
| 「「「「」」の「「」」の「「」」の「「」」の「「」」の「「」」の「「」」の「 |                     |                                 |     |                 |          |
| MAN                                    |                     |                                 |     |                 |          |
|                                        |                     |                                 |     |                 |          |
|                                        |                     |                                 |     |                 |          |
|                                        |                     |                                 |     |                 |          |
|                                        |                     |                                 |     |                 | *        |
|                                        | 又任名(M):<br>文件米刑(m): | ms_rLLopen_LLB.FmL<br>田仲庄(4 年刊) |     |                 | ж<br>(U) |
|                                        | 大厅关望(1).            | 四十年(*.1*1)                      |     | • 402           | H.       |

图 7.60

4) 在右侧的[编辑向导]窗口的"组"一栏的下拉列表中选择【MS\_PLCopen\_LIB】或【全部 FU 和 FB】,在下拉列表下面便会列出所有的 MC\_前缀的运动控制功能块。

MULTI	PROG Exp	ess - Par	aServoFr	omTemp1806	522 - [代码:m	ain]						<b>1</b>
■ 文件(E)	编辑(E)	規图(⊻) □	E程(P) 建	立(B) 対象(C	) 布局(L) 联	机(N)附加(X) 窗口(M	D 2					- 8 ×
₩ 打开工	保存	① 放大	○ 缩小	工程树	编辑向导	消息 交叉参考 监袖	- 🤘 🧐 18日 安重工 3		( ) 週试开/关 工程控			
🗉 🥥 库				bazal						1	- :编辑向导	7 🔻 🖬
	MS_PLCopen P#F#1	LIB		50122	· ·						祖:	
St 12	5≫-22 Axis Tvoe			11.11							<pre> </pre>	
🚽 🥥 逻辑	POV			右軸占							名称	描述
I I	main			Th	· ·						MC_AbortTrigger	
				并行	l .						MC_GrpDisable	
				11/1							T MC_GrpEnable	
				右线圈							TWC GrpReadictPos	
				L							MC GrpReset	
				左侧电							I MC_GrpSet0verride	
				14	· ·						MC_GrpStop	
				右侧电	l .						MC_GrpSyncMoveAbs	
				-8-								
				触点类型	· ·						MC MoveAbsolute	
											MC_MoveCircAbs	
				Tablan							MC_MoveCircRel	
				复中Jrr	· ·						MC_MoveLinAbs	
				+TItA PP							MC MoveBelative	
				ыл <u>ж</u> тт							MC_MoveVelocity	
				÷	· ·						MC_Power	
				创建步	l .						MC_ReadActualPosition	
				母							KC_ReadActualVelocity	
				插入SF	- ·						MC_ReadAxisInfo	
				8	l .						TMC ReadDigitalInput	
				创建动作							MC_ReadMotionState	
											MC_ReadParameter	
											MC_KeadStatus	
											MC SetOverride	
											MC_SetPosition	
											at MC_Stop	
											MC_SyncMoveAbsolute	
					· ·						MC_Synchovevelocity	
											MC_WriteDigitalOutput	
											MC_WriteParameter	
					1 • İ							
] @ ±# [	11 硬件					n 🔲 Axis Type						
				· .	an Idelinas							
	建立 (谱	呈)题去	入信申入	PIC出错 》:	ŧTED /		<u>★ ■ </u> <u>★ ■ </u>	▶ <b>▽ <u>変量</u> / / / / / / / / / / / / / / / / / / /</b>	POU/工作单	访问 I/O地址		
			A mark P								86,125	D: >2GB

图 7.61

# 7.3.2. 运动控制功能块列表

下表为 i5 运动控制功能块,也称为运动控制指令的种类和说明。

种类	指令	PLCopen 标准	功能说明
	MC_Power	是	该功能块控制伺服驱动器使能或禁止.
	MC_Home	是	该功能块控制一个轴回原点的操作。回原点 的模式、速度等参数可以由轴参数来设定。
	MC_Stop	是	该功能块用于轴减速停止。
轴指令	MC_MoveAbsolute	是	该功能块指定轴的绝对坐标的目标位置,进 行定位。
	MC_MoveRelative	是	指定轴自指令当前位置起的移动距离,进行 定位。
	MC_MoveVelocity	是	以指定速度参数进行运动。
	MC_TorqueControl	是	以指定转矩参数进行运动。
	MC_SetPosition	是	设置轴的当前指令位置。

	MC_SetOverride	是	设置轴的倍率因子。
	MC_ReadParameter	是	读取轴参数。
	MC_WriteParameter	是	修改轴参数。
	MC_ReadActualPosition	是	读取轴的实际位置。
	MC_ReadActualVelocity	是	读取轴的实际速度。
	MC_ReadStatus	是	读取轴状态。
	MC_ReadMotionState	是	读取轴运动状态。
	MC_ReadAxisInfo	是	读取轴信息。
	MC_ReadAxisError	是	读取轴错误信息。
	MC_Reset	是	轴错误复位。
	MC_TouchProbe	是	轴位置锁存。
	MC_AbortTrigger	是	取消轴位置锁存。
	MC_ImmediateStop	扩展指令	立即停止。
	MC_SyncMoveVelocity	扩展指令	单轴周期同步速度模式运动。
	MC_SyncMoveAbsolute	扩展指令	单轴周期同步位置模式运动。
	MC_GrpEnable	是	启用轴组。
	MC_GrpDisable	是	取消轴组。
	MC_GrpReset	是	轴组错误复位。
	MC_GrpSetOverride	是	设置轴组的倍率因子。
	MC_GrpReadActPos	是	读取轴组中各轴的实际位置。
	MC_GrpStop	是	减速停止轴组运动。
	MC_GrpImmediateStop	扩展指令	立即停止轴组运动。
	MC_MoveLinAbs	是	指定绝对坐标的目标位置,进行直线插补。
轴组指 令	MC_MoveLinRel	是	指定相对位置,进行直线插补。
	MC_GrpSyncMoveAbsolute	扩展指令	轴组周期同步位置模式运动。
	LMC_MoveLinear	扩展指令	轴组直线插补,插补中可以动态改变目标位 置(绝对)
	MC_MoveCircAbs	是	绝对圆弧插补(二维)
	MC_MoveCircRel	是	相对圆弧插补(二维)
	MC_ReadDigitalInput	是	读取主站输入口状态。
10 华人	MC_WriteDigitalOutput	是	保留。
し日文	LMC_WriteBit	扩展指令	设置主站输出口状态/设置从站输出口状态 (系统任务下)(按位设置)。

	LMC_WriteByte	扩展指令	设置主站输出口状态/设置从站输出口状态 (系统任务下)(按字节设置)。
高速计	LMC_EnableHSCounter	扩展指令	该功能块控制高速计数器使能或禁止.
数器指 令	LMC_HSCounter	扩展指令	读取高速计数器的计数值。
扩展轴	LMC_WriteObject	扩展指令	修改轴参数,参数类型为整形。
指令	LMC_ReadObject	扩展指令	读取轴参数,参数类型为整形。

# 7.3.3. PLC 程序添加功能块

1) 光标移动到想要添加的功能块上,按住鼠标左键,将其拖动至工作区域的某个位置。



图 7.62

2) 然后松开鼠标。



图 7.63

3) 鼠标左键单击代码的空白区域的某个位置,点击[网络],向代码中添加一个网络。移动

ń 网络 MC\_Power\_1 MC\_Power 右触点 Axis \_\_\_\_ Axis 001 并行 C001 C002 Status Enable 右线圈 Valid 左侧电. Error ErrorID 右侧电. 触点类型 复制FP 切换FP 创建步 插入SF. 创建动作 1 • Ė ■ 代码: Unt.

工作区域中的功能块,使功能块的 Enable 和 Status 引脚与网络线重合,形成网络回路。

图 7.64

# 7.3.4. 功能块实例化

 右击[数据类型],在弹出的菜单栏中将光标定位到[插入],然后点击[数据类型],创建一 个包含轴信息的结构体。若用工程模板创建的工程,则已经包含了轴的数据类型。



2) 点击功能块的引脚 Axis

名称(N):       定义范围         夏制PP       V000          切換PP.       数据类型(D):          助提类型(D):        局部安量组(V):         局部安量组(V):           内读P.       REAL           用法(U):        RETAIN          创建步.       VAR        RETAIN         初值(I):            描入SF.            创建动作       I/O地址(S):           描述(E):		组: (1) 名
	9定 双消 助(H)	

3) 修改名称,数据类型,初值,完成该引脚的设置。

变量属性		23
名称(N): MyAxis 数据类型(D): AXIS_REF 用法(U): VAR VAR I(D): (AxisNo := 1) I/O地址(S): 描述(E):	定义范围 ● 局部(L)  ● 全局(G) 局部变量组(V): ■ Default  ▼ 全局变量组(A): □	确定 取消 帮助(H)
■ PDD ■ OPC □ 隐藏(X) □ 初值作为默认值.	☑ 显示工作单的所有变里(₩)	

图 7.67

4) 输入触点 C001 和输出引脚 C002 实例化,如果输入触点和输出引脚不实例化,将不能 完成工程的编译。

001 · PowEn ● ewEn ● ewEn	MC_Power_1 MyAxis Axis Bhable Status Valid Error BrrorD C C C C C C C C C C C C C	· · · · · · · · · · · · · · · · · · ·
PDD OPC	<ul> <li>□ 隐藏(X)</li> <li>☑ 显示工作单的所有变量(W)</li> <li>□ 初值作为默认值.</li> </ul>	
触点/线圈 ◎ 触点(T) ◎ 线圈(0)	类型(Y): -( )- ▼	

图 7.68

# 7.4. MC 单轴运动实例

注:此处的单轴仅指未使用轴组,并不是指系统只有一个伺服轴。

# 7.4.1. PLC 工程建立

以下实例模拟生产线上贴标机给商品贴标签的过程,轴1代表贴标机的标签轴、轴2代表配合贴标机的水平传送带轴,实现步骤如下:

1) 使用工程模版创建新工程。



图 7.69

# 2) 添加伺服从站

ct Explorer	De	vice E	ditor								
■ 配置/资源	G	ienera	I PDO Mapping Variab	les Advan	ced Options	Distributed	Clock	Init Commands C	oE Object-l	Dictionary	
Slave_1001 [Leetro A1E] (1001)	ь	nputs					Outpu	ts			
Slave_1002 [Leetro A1E] (1002)			✓ 1st Transmit PDO mag	oping		0x1A00	i 🔽	✓ 1st Receive PD	eive PDO mapping 0x16		
			Name	Index	Bit Length			Name	Index	Bit Length	
			Status Word	0x6041:00	16			Control Word	0x6040:00	16	
			ActualPosition	0x6064:00	32			TargetPosition	0x607A:00	32	
			ActualVelocity	0x606C:00	32			TargetVelocity	0x60FF:00	32	
			ModeOfOperationDisplay	0x6061:00	8			ModeOfOperation	0x6060:00	8	
			Dummy Byte2	0x37FF:00	8			Dummy Byte1	0x37FE:00	8	
			ErrorCode	0x603F:00	16		•	2nd Receive Pl	DO mapping	(excluded by 0x1600)	0x1601
			DigitInputs	0x60FD:00	32			Name	Index	Bit Length	
		•	2nd Transmit PDO ma	pping (exclu	ded by 0x1A00	) 0x1A01		Control Word	0x6040:00	16	
			Name	Index	Bit Length			TargetPosition	0x607A:00	32	
			Status Word	0x6041:00	16		-	3rd Receive PE	O mapping	(excluded by 0x1600)	0x1602
			ActualPosition	0x6064:00	32			Name	Index	Bit Length	
		-	3rd Transmit PDO ma	pping (exclu	ded by 0x1A00)	0x1A02		Control Word	0x6040:00	16	
			Name	Index	Bit Length			TargetVelocity	0x60FF:00	32	
			Status Word	0x6041:00	16						
			ActualPosition	0x6064:00	32						



3) PDO 与 IO 变量映射



图 7.71

- 4) 制作工程,检查操作是否正确
- 5) 配置运动控制参数

本報设置 MC_Axis001 MC_Axis002 報組设置         抽屉         1           報告 和快能         1世使用的抽 ~           報告         1           日本日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	MotorControl Setting
	* 轴设置 M_Axia001 M_Axia002 轴组设置         抽墨本设置 单位换算设置 操作设置 原点返回设置 [位置计数设置] 何朋驱动设置           轴母         1           抽皮         1           抽皮型         06月服油 ~           抽皮型         06月服油 ~           輸出控制         08未分配 ~

图 7.72 注:实际配置在此不一一描述,请参考章节 6.8, 6.9。

6) 根据工艺流程,控制需求编写 PLC 程序



运动功能块具体使用方法参考指令手册,更多详细帮助信息可参考 PLC 帮助(按"F4"打开)。 7) 在编程窗口点击"制作"按钮,编译工程。

8) 编译后点击"工程控制对话框"按钮,再选择"下装",程序下装后点击"冷启动"。

# 7.4.2. 单轴 DEMO 控制逻辑及操作方法

操作方法:控制器上电或启动以后,先置位"First\_check",并保持为1,然后置位"Device\_start", 伺服轴按照控制流程循环执行贴标动作。需要停止演示时,复位"Device\_start"即可。 控制逻辑

每一次下装工程或重启控制器以后必须先**置位布尔变量"First\_check"并保持该变量为1**, 这一步是为了保证下装或者重启以后,复位可能出现的控制器和伺服错误信息。若复位指令 执行成功,会自动激活轴使能,周期信号会以2S一次频率输出,模拟传感器信号。



图 7.74

传送带轴在设备启动后会按照变量"Velocity"指定的速度保持匀速运动。标签轴在周期信号 出发后,延迟一定时间后开始以和传送带相同的速度转动固定距离后停止,等待下一次触发, 模拟贴标动作。



图 7.75

复位"Device\_start",伺服轴停止并关闭轴轴使能。



注: 具体 DEMO 可在 MLUPROG 软件的菜单"文件"、"打开工程\解压缩工程"目录下找

到。

# 7.5. MC 轴组运动实例

以下实例模拟立体仓库中堆垛机取托盘的过程,X 轴代表堆垛机的水平移动轴、Y 轴代 表堆垛机的垂直移动轴,Z 轴代表堆垛机的货叉伸缩轴;实现步骤如下:

# 7.5.1. PLC 工程建立

1) 使用工程模版创建新工程。



图 7.77

2) 添加伺服从站

----

In	enera puts	Variab	ales Advan	cea Options I	Distributed	I CIOCK	Init Commands C	or Uplect-L	Jictionary	
In	puts								,	
						Outpu	ts			
1	•	💽 1st Transmit PDO map	oping		0x1A00	•	Ist Receive PD	O mapping		0x1600
		Name	Index	Bit Length			Name	Index	Bit Length	
		Status Word	0x6041:00	16			Control Word	0x6040:00	16	
		ActualPosition	0x6064:00	32			TargetPosition	0x607A:00	32	
		ActualVelocity	0x606C:00	32			TargetVelocity	0x60FF:00	32	
		ModeOfOperationDisplay	0x6061:00	8			ModeOfOperation	0x6060:00	8	
		Dummy Byte2	0x37FF:00	8			Dummy Byte1	0x37FE:00	8	
		ErrorCode	0x603F:00	16			2nd Receive Pl	DO mapping	(excluded by 0x1600)	0x1601
		DigitInputs	0x60FD:00	32			Name	Index	Bit Length	
	-	2nd Transmit PDO ma	pping (exclu	ded by 0x1A00)	0x1A01		Control Word	0x6040:00	16	
		Name	Index	Bit Length			TargetPosition	0x607A:00	32	
		Status Word	0x6041:00	16		•	3rd Receive PD	O mapping	(excluded by 0x1600)	0x1602
		ActualPosition	0x6064:00	32			Name	Index	Bit Length	
	-	3rd Transmit PDO ma	pping (exclue	ded by 0x1A00)	0x1A02		Control Word	0x6040:00	16	
		Name	Index	Bit Length			TargetVelocity	0x60FF:00	32	
		Status Word	0x6041:00	16		11				
		ActualPosition	0x6064:00	32						
		•	ActualPosition ActualVelocity ModeOfOperationDisplay Dummy Byte2 ErrorCode DigitInputs 2 20 20 20 20 20 20 20 20 20 20 20 20 20	ActualPosition     0x6064.00       ActualVelocity     0x606C.00       ModeOfOperationDisplay     0x6061.00       Dummy Byte2     0x37F.00       ErrorCode     0x609F.00       DigitInputs     0x609F.00       2nd Transmit PDO mapping (exclut       Name     Index       Status Word     0x6041.00       ActualPosition     0x6041.00       ActualPosition     0x6041.00       ActualPosition     0x6041.00	ActualPosition     0x6064:00     32       ActualVelocity     0x606C:00     32       ModeOfOperationDisplay     0x6051:00     8       Dummy Byte2     0x37F:00     8       ErrorCode     0x60F:00     32       2     2nd Transmit PDO mapping (excluster by 0x1A00)       Name     Index     8it Length       Status Word     0x6064:00     32       3rd Transmit PDO mapping (excluster by 0x1A00)       Name     Index     8it Length       Status Word     0x6064:00     32       Status Word     0x6064:00     32	ActualPosition     0x606400     32       ActualVelocity     0x606C00     32       ModeOfOperationDisplay     0x605100     8       Dummy Byte2     0x37F500     8       ErrorCode     0x60FD00     32     16       DigitInputs     0x60FD00     32     1       2nd Transmit PDO mapping (excluster by 0x1A00)     0x1A02       Xame     Index     8it Length       Status Word     0x6064:00     32       3rd Transmit PDO mapping (excluster by 0x1A00)     0x1A02       Name     Index     8it Length       Status Word     0x6064:00     32       Status Word     0x6064:00     32       Ox6064:00     16     16       ActualPosition     0x6064:00     32	ActualPosition         0x6064:00         32	ActualPosition       0x6064.00       32       Image: Status Weight of Status Word         ActualPosition       0x6065.00       32       Image: Status Word         Digitinputs       0x6067.00       32       Image: Status Word         Status Word       0x6064.00       32       Image: Status Word         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x1A00)       0x1A00         Name       Index       Bit Length       Image: Status Word       0x6041.00         Status Word       0x6041.00       16       Image: Status Word       0x6041.00         Name       Index       Bit Length       Image: Status Word       0x6064.00       32         Name       Index       Bit Length       Image: Status Word       0x6064.00       32       Image: Status Word       0x6064.00       32       Image: Status Word       0x6064.00       32       Image: Status Word       Image: Status Word       Image: Status Word       0x6064.00       32       Image: Status Word       Imagee: Status Wo	ActualPosition       0x6064:00       32       Image: Position       0x607A:00         ActualVelocity       0x606C:00       32       Image: Position       Target: Position       0x607A:00         ModeOtOperationDisplay       0x6061:00       8       Image: Position       0x607A:00         Dummy Byte2       0x37F:00       8       Image: Position       0x607Depration       0x607F:00         Dummy Byte2       0x37F:00       8       Image: Position       0x507F:00       2       Image: Position       0x507F:00         Digitinputs       0x60FD:00       32       Image: Position       0x607D:00       Image: Position       0x607D:00         Name       Index       Bit Length       Image: Position       0x607D:00       Image: Position       0x607D:00         Name       Index       Bit Length       Image: Position       0x607D:00       Image: Position       0x607D:00         Name       Index       Bit Length       Image: Position       0x60FD:00       Image: Position       0x60FD:00         Name       Index       Bit Length       Image: Position       0x60FD:00       Image: Position       0x60FD:00         Status Word       0x60F1:00       16       Image: Position       0x60FD:00       Image: Position	ActualPosition       0x6664:00       32       I         ActualVelocity       0x666:00       32       I         ModeOtOperationDisplay       0x606:00       8       I         Dummy Byte2       0x37F:00       8       I         ErrorCode       0x605:00       16       I         V       2nd Transmit PDO mapping (excluded by 0x1A00)       0x1A01         Name       Index       Bit Length         Status Word       0x606:00       32         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x4002         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x402         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x402         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x402         V       3rd Transmit PDO mapping (excluded by 0x1A00)       0x402         Name       Index       Bit Length         Status Word       0x606:00       32         Name       Index       Bit Length         Status Word       0x606:00       32         Name       Index       Bit Length         Status Word       0x606:00       32         Name       Index       Bit Length

图 7.78

# 3) PDO 与 IO 变量映射

EtherCAT Variables Mapping
Variables Mapping

配置/资源/PLC_SYS_TICK_CNT (UDINT)	*	*	配置/资源	
配置/资源/PLC_TASK_DEFINED (INT)			IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLCMODE_ON (BOOL)		< <	IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCMODE_LOADING (BOOL)		×	IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCMODE_STOP (BOOL)		^	IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCMODE_RUN (BOOL)	=	New	IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCMODE_HALT (BOOL)			IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLC_TICKS_PER_SEC (UINT)			IN: Slave_1001 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLC_MAX_ERRORS (UDINT)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLC_ERRORS (UDINT)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLC_TASK_AVAILABLE (INT)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLC_SYSTASK_AVAILABLE (INT)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappi
配置/资源/PLCDEBUG_FORCE (BOOL)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCDEBUG_BPSET (BOOL)			IN: Slave_1002 [Leetro A1E].1st Transmit PDC	mappir
配置/资源/PLCDEBUG_POWERFLOW (BOOL)	-	-		Þ
Mapping Resource '配置/资源 (IN)':  Resource '配置/资源 (QUT)':  				
mory wapping				
Charact Memory				
Shared Memory				
Input: 0				

ß

8

图 7.79

# 4) 制作工程,检查操作是否正确

5) 配置运动控制参数

|--|

▲ 轴设置	釉基本设置  单位换算设置  操作设置  限位设置  原点返回设置  位置计数设置  伺服驱动设置
MC_Axis001	
MC_Axis002	
MC_Axis003	袖徒能 1+使用的轴 ▼
▲ 轴组设置	抽类型 □====================================
MC Group001	
MC Group002	
	輸出控制 0#未分配 ▼

图 7.80

注:实际配置在此不一一描述,请参考章节6.8,6.9。

6) 根据工艺流程和控制需求编写 PLC 程序



运动功能块具体使用方法参考指令手册,更多详细帮助信息可参考 PLC 帮助(按"F4"打开)。 7) 在编程窗口点击"制作"按钮,编译工程。

8) 编译后点击"工程控制对话框"按钮,再选择"下装",程序下装后点击"冷启动"。 注:具体 DEMO 可在 MLUPROG 软件的菜单"文件"、"打开工程\解压缩工程"目录下找 到。

# 7.5.2. 轴组 DEMO 控制逻辑及操作方法

操作方法:控制器上电或启动以后,先置位"First\_check",并保持为1,然后置位"Device\_start", 伺服轴按照控制流程运行完一个动作周期后停止,复位"Device\_start"后再将其置位(生成 上升沿),伺服轴又会再运行一个动作周期。需要停止演示时,复位"First\_check"即可。 控制逻辑

每一次下装工程或重启控制器以后必须先**置位布尔变量"First\_check"并保持该变量为1**, 这一步是为了保证下装或者重启以后,复位可能出现的控制器和伺服错误信息。若复位指令 执行成功,会自动激活轴使能与启用轴组,轴组有X轴和Y轴组成。本段代码如下所示



然后给变量 "AXIS\_X" (X 轴位置) 和 "AXIS\_Y" (Y 轴位置) 赋值,也可以直接使用初始值 演示。接着置位变量 "Device\_start",轴组运动指令被上升沿激活,X 轴与 Y 轴开始同步运 动,模拟堆垛机沿直线运动到货物存放位置。



到达货物存放位置后,首先关闭轴组(释放轴组后,Y轴才能单独运动),Y轴和Z轴会模 拟堆垛机货叉取托盘过程,代码如下



货物从货架取出后,轴组会在下一个 PLC 周期内再次被激活,然后 x 轴和 Y 轴模拟堆垛机带着货物沿直线返回出发位置,至此一个动作周期完成。代码如下

图 7.85

# 7.6. POU 单元调用方法举例

软件支持 3 种 POU 单元,程序(Program)、功能块(Function Block)和功能(Function), 总的原则: "程序"可以调用所有"功能块"和"功能",但是不能调用另一个"程序"; "功能块"只能调用除它自身以外的其他"功能块"和所有"功能";"功能"只能调用除 自身以外的其它"功能";在原则内,"功能块"和"功能"能够重复调用,以下面的工程 举例



在该工程中,main 是程序 POU,FB1 和 FB2 是功能块 POU,FC1 和 FC2 是功能 POU。 首先打开 main 程序 POU,允许的调用如下。



然后打开 FB1 功能块 POU, 允许的调用下。



图 7.88

下面这种情况属于错误调用,因为 FB1 调用了自身,属于递归调用范畴,在 PLC 工程内不允 许递归调用。



最后打开 FC1 功能 POU,允许的调用如下。

Incertainee chicas constraine (1969) et 1										
∎ 文件①	编辑(E)视图(V) I	[程(P)建立(B)对象	(O) 布局(L) 野	机(N) 附	肋(X) 窗口(W) <u>?</u>					
<b>ジ</b> 打开エー	■ ① 保存 放大		编辑向导			新作 调试开/关 工程控				
₩○ 和 和 和 和 和 和 和 和 和 和 和 和 和	□ □ □ □ □ □ □ □ □ □ □ □ □ □	n_IIB			FC2 EN ENO C2 +	周用正确				
e #IKP										

图 7.90

下面这种属于错误调用,因为 FC1 调用了自身,工程内不允许递归调用。



图 7.91

# 7.7. 多任务、多程序组合应用举例

工程中可以使用多个任务和多个程序灵活组合,实现不同的项目需求,以下面的工程为例讲 解正确的组合方法、优先级等问题。

首先需要明确,程序 POU 必须插入到"任务"中才能执行,而"任务"中也只能插入程序, 不能插入其他类型的 POU;

工程如下, main\_1、main\_2、main\_3 是程序 POU, 任务 1 是 DEFAULT (默认循环) 任务, 任 务 2 和任务 3 都是 CYCLIC (周期循环) 任务。分别在任务 1、2、3 中插入程序 main\_1、 main\_2、main\_3;



图 7.92

任务 2, 任务 3, 是 CYCLE 任务, 任务优先级、循环执行周期可根据用户自行设定。任务优 先级数据越小, 优先级越高。

任务1是 DEFAULT 任务,任务优先级为16,循环执行周期为4ms。

CYCLE 任务、DEFAULT 任务均由系统周期性调度执行。优先级高的任务优先抢占系统资源(CPU 运算能力),优先执行,执行完毕后释放系统资源。

程序的执行顺序与优先级设置有关,分为两种情况;

情况 1: 任务 2 与任务 3 的优先级不同,比如任务 2 优先级为"0",任务 3 优先级为"1",即任务 2 优先级高于任务 3,则在 PLC 启动时,按照调度规则,优先执行任务 2,其次为任务 3,再然后是任务 1。

情况 2: 任务 2 与任务 3 的优先级相同,比如任务 2 和任务 3 优先级都为"0", 即任务 2 任务 3 优先级相同,则在 PLC 启动时,按照调度规则,任务 2、任务 3 平均分配系 统资源(CPU 运算能力)优先启动执行,再然后是任务 1。

关于优先级的详细信息,请参考本文8.3节。

程序与任务的组合可以是多种多样的,工程任务中插入的程序也可以是如下形式

<b>》</b> 打开工	↓	○ 缩小 工程树	- ** 編輯向导	■ 消息	交叉参考 监视	● ◎ ◎ ① ② ① ①	<b>全日</b> 変単	制作	<b> </b> {
	□··· ● 物理硬件 □·· ● 配置 : eCL □·· ● 资置 : : : : : : : : : : : : : : : : : :	R ARM_LE_GCC3*							
<b>米</b> (1) 网络		ks 任务1 : DEFAULT □ main_3 : main_3 任冬2 : CYCLTC							
右触点		口 main_2 : main_2 任务3 : CYCLIC □ main_1 : main_1	2						
并行 10 方线圈	61.0 10_0	bal_Variables Configuration							
石或四 左侧电									
右侧电									•

#### 7.8. 封装及调用用户库举例

'库'子树含有您的工程中所包含的所有库。它们既可以是用户自定义库,也可以是固件 库:

**固件库**是那些包含了由 PLC 制造商所提供的 POU 的库。固件库的文件扩展名为\*.fwl。 用户库是"普通的"工程,其 POU 将被重复使用。作为用户库而包括进来的工程中的程序、功 能块、功能和用户自定义数据类型都可以在当前工程中重复使用。用户库的扩展名为\*.mwe。 在前面的章节中已经详细讲解了如何添加使用固件库,此节专门讲解如何导入用户库。 在添加用户库前,我们应该先有一个作为用户库工程,可以新建以下工程,工程名为 USER\_LIABRARY。工程包含一个程序 POU (main),一个功能块 POU (FB1),一个功能 POU (FC1);在 POU 中编写好代码以后,编译保存。工程默认使用工程模板创建。(此处仅是 演示,POU 内没有编写具体代码)

有两点需要重点注意:

1、一定要保证作为用户库的工程能编译制作通过,这是工程能作为用户库工程的前提条件, 每次用户库工程有更改都应该记住重新编译保存; 接着我们再次使用工程模板新建第2个工程作为调用工程,工程名为"DE1"。在第2个工程中使用第1个工程作为用户库,导入方式与导入固件库类似。在工程树中,右键点击'库'文件夹,并从上下文菜单选择'插入 > 用户库...',选中对应工程,点击'包括库'对话框打开。



图 7.94

包含用户库以后我们可以在库目录下看到它,如下图所示



图 7.95

然后我们需要进行一个额外的操作, 删除红圈处的数据类型文件 "Axis\_TYPE", 因为两个工

程都是用工程模板建立,所以都包含了相同的自定义数据类型文件,若不删除一个编译时会 提示重复定义数据类型。用户库在调用工程中是不能更改的,所以删除工程的自定义数据类 型文件是唯一选择。

另外还要注意的一点:程序也是用户库中的可调用 POU,可以直接插入到任务中执行。但是注意不要和调用工程中的 POU 同名,否则用户库中的程序 POU 会被隐藏。



#### 图 7.96

调用用户库中的 POU 和调用其它 POU 的方法一致,直接用鼠标左键点住,拖动到代码工作 单内即可,双击用户库中 POU 单元,可以通过查看模式打开 POU 的代码工作单和变量工作 单;



图 7.97

(图中 FB1 定义了形式参数, FB2 没有定义, 所以调用图形有区别)

#### 用户库的多重包含

软件支持用户库的多重包含,如用户库工程 2 包含使用了用户库工程 1,最终工程 3 包含了 用户库工程 2 是可行调用方式;即使用了用户库的工程本身也可以作为用户库。注意此时必 须满足的条件是工程1也作为用户库导入到工程 3 中;

# 注意各个 POU 不要重名;

不建议在两个工程中互相包含为用户库,极易陷入递归调用的陷阱。



图 7.98

#### 用户库的加密和隐藏

该方法适用于所有 POU 和自定义数据类型文件的加密,不管是用户库工程还是调用工程,以用户库工程举例。

打开用户库工程,若我们要隐藏 POU 单元 "FB1"则右键点击 "FB1",在打开的菜单中选择 "属性",在属性菜单中选择 "安全性"把 "读保护"、 "写保护"都勾选上,点击 "应用"再点击 "确定"。

MULTI	PROG Express	s - LIABRAR	Y1											
文件(E)	编辑(E)视图	(1) 工程(12)	建立(B)	联机( <u>N</u> )	附加(X) 2	2								
<b>ジ</b> 打开エー	保存	放大	<b>〇</b> 缩小	工程树		消息	交叉参考	山田	◎ 変重工	♀ 受重	<b>〕</b> 制	) Iff	🦻 📶 调试开/关 工程控	
网络 右触点 并石线通电。 夏谢呼P 切换FP.		NS_PLCopen_1 大学 Axis_Type FOU FOU FD1* FB1* FB2		'FB1' 名称 权际	类型   ((R): 写保护 读保护	PIC/处理	器   属性 确定 [	安全性             取消	) ( 应用 (A)		23 FBD			

图 7.99

接下来我们还需要设置密码,在菜单栏中选择"文件">"输入密码",设置密码以后点击

"激活密码"。

MULTIPROG Express - LIABRARY1
文件(E)编辑(E)视图(V) 工程(P)建立(B)联机(N)附加(X) 2
所容       新S_FLCopen_LIB         数据失型       文Axis_Type         逻辑POU       原本is         逻辑POU       原本is         了FB:*       FD2         方线理       FD3         在設理       第万2         如此点类型       数活密码(k)         数活密码(k)       数活密码(k)         動力       数活密码(k)

图 7.100

密码激活以后如下图所示;我们可以看到 POU 单元 "FB1"前面出现了一个钥匙符号,此时再双击 "FB1"无法打开它的工作单。另外一种激活密码的方式是在设置密码时点击"确定"编译制作后先关闭工程,然后重新打开。

MULT	PROG Ex	press - Li	IABRAR	Y1											
文件(E)	编辑(E)	视图(⊻)	工程(P)	建立( <u>B</u> )	联机( <u>N</u> )	附加(X) ?									
<b>ジ</b> 打开エー	. 保存	<b>④</b> 放:	t	● 缩小	工程树			交叉参考	山口	● 変重工	<b>9</b> 受重	前	<b>7</b> 调试开/关	 工程控	
四日本 日本 日		库 数据类型 定和is 逻辑POU I main N FB2	LCopen_I ! _Type 31	ΤB											

图 7.101

若想要重新编辑"FB1"则需要输入正确的密码,"文件">"输入密码"。

MULTIP	ROG Express - LIABRAF	RY1						
文件(E)	扁損(E) 视图(V) 工程(P	)建立( <u>B</u> )联机( <u>N</u> )	附加(X) ?					
<b>ジ</b> 打开エー	保存 放大	Q □ 缩小 □ 工程树		消息 交叉参考	- 🚺 😸 监视窗口 変重工	<b>全国</b> 変重	計 制作	🦻 💭 调试开/关 工程控
网络 五触点 并行 五线圈	□ □ □ □ □ □ □ □ □ □ □ □ □ □	LIB		工程注册(login) 密码(P): 编辑密	码 (E)	确定		

图 7.102

# 安全设置读/写保护和文件输入密码两个步骤缺一不可,缺少任何一个环节都实现不了加 密效果。

注。若没有勾选"读保护"和"写保护",只是添加了文件密码,那么各个 POU 的代码和变量工作单都可以被编辑,仅仅是 POU 的"属性">"安全性"被锁定,无法勾选。

#### 取消密码操作

若不想再使用密码,在输入密码以后,再次选择"文件">"输入密码",然后点击"撤销 密码",确认即可。密码撤销后,工程不再处于加密状态。

大 缩小 🗋		<u>消息 [交叉参考]</u> 监视窗口	1 役里上…	受望 :	制作	调试开/天	上柱控.
Copen_LIB XARY1 htg:###4	ſ	工程密码			×)		
UGA文型 習輯POV !		密码(P):(至少5个字符) 		确定			
		密码确认(C):		42.75			
	取消密码	▶ 撤消密码(R)	激活密码	码(A)			

## 图 7.103

按照上述方法把用户库工程中所有 POU 单元和数据类型文件都设置为加密模式,编译后保存。然后打开调用工程,可以看到加密后的用户库在调用工程中的状态。

用户库中的 POU 和数据类型定义文件都已经显示为加密状态,无法再查看 POU 中的内容 (查看模式不再有效)。加密的 POU 不能再从目录下直接拖动到代码单调用,只能从右侧 的"编辑向导"中选择对应的用户库,然后拖动调用;

	MULTIPROG Express - LEAD DEMO1 - [代始祖年1*]											
■ 文件(E)	编辑(E) 视图(V) 工程(P) 建立(B) 联机(N) 附加(X)	2										
<b>ジ</b> 打开工		消息 交叉参考	· 监视窗口 变量工	2 安堂 制	) 🧳 🚚 作 调试开/关 工程控							
間の	● ● 床 ● ● 床 ● ● 除 S_PLCopen_LIB ● ● ILABRARY1 ● ● ● 教提类型 ■ ● ● 逻辑FOU ■ ● 200						:編輯向导 组: <liabrary1> 名称 描述 書 FB1</liabrary1>					
右触点	■ FB1 FB2 ● 新展共画 · · · · · · · · · 无法	直接拖动	调用		从编辑向	导中调用	■ FB2					
	□											
右侧电												
1												

若想要隐藏用户库中的部分 POU,保证这一部分 POU 不能在调用工程中被使用,一种灵活的方式是把这一部分 POU 设置为隐藏。重新打开用户库工程,因为工程出于加密状态,所以首先选择"文件">"输入密码",解锁工程以后,选择要隐藏的 POU 单元,比如"FB2", 右键点击在打开的菜单中选择"属性">"属性",勾选"隐藏"然后点击"应用"、"确定",编译后保存。

	OG Express - LIABRARY1		
文件(E) 編	辑(E)视图(V) 工程(P)建立	(B)联机(N)附加(X)?	
₩ 打开工	日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日		
四路 石融点 并行 石线理 。  石 肥地  百 肥地  百 號  百 號  百 號  百 號  百 號  百 形  	日 ● ● ● ● ● ● ● ● ● ● ● ● ●		YB2'     X       名称 [ 类型] PL/处理器 属性 [ 安全性 ]       只读 (R)       「現蔵 (B)       確定<取消

# 图 7.105

再次打开调用工程,可以发现在"编辑向导"中只能调用 POU 单元"FB1",已经看不到"FB2"。

3 MULTIPROG Express - LEAD DEMO1 - [代码理序1]													
■ 文件(E)	编辑(E) 视	週(⊻) 工程(6	2)建立(	B) 联机(N) 附加(2	2								
<b>じ</b> 打开エー・・	保存	④ ( 放大 ≤	<b>3</b> 渝小	12日本 編輯向导	消息	交叉参考	□ 「□ 监视窗口 変量工····	2 変重	(1) 20 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)				
一一一日本語の一番目的である。		KS_FLCopen_L1 IABRAR1 新聞大型 「加速ない」 「「「「「「」」 「「」」 「「」」 「「」」 「「」」 「」 「	IB s_Type n					- - - - - - - - -	。 (又	【有 FB1	可调用		: 編編局号 组: (LIABRARYI) 名称 描述 書FB1
图 7	.106												

# 7.9. 工程输出应用举例

当需要把工程输出给另一方使用时,有两种方式可以实现工程输出。原文件直接拷贝和原文件压缩后拷贝。在输出过程中,因为库(固件库、用户库)的引用位置变更需要接收用户进行一些操作,才能正确使用。

## 原文件直接拷贝

用工程模板新建一个工程,保存工程并将工程命名为"DE1",完整的工程文件应包含以下两部分。



图 7.107

打开工程"DE1",在此例中,工程内同时包含了固件库和用户库(用户库已经加密)。如下图所示

	MULTIPROG Express - DE1 - [代码:main_2*]												
■ 文件(E)	编辑( <u>E</u> )	视图(⊻) Ⅰ	[程(P) 建立	之( <u>B)</u> 联机	(N) 附加(X)	2							
<b>6</b> 打开工	保存	⊕ 放大	<ul><li>○ 缩小</li></ul>	工程树		消息	交叉参考	山口	◎ 変量工	♀ 変単	新作	● 週试开/关 工程控	<b>.</b>
マン 対 対 が 石 健 に 一 井 行 石 近 男 名 一 井 行 一 二 一 井 行 一 二 一 一 一 一 一 一 一 一 一 一 一 一 一		3 MS_PLCop4 1 LABRARY 3 JAR 3 JAR	Axis_Type U main FB1 FB2	──	固件库 户库								
						冬	7.108						

右键点击固件库和用户库,选择"属性",看到两个库的路径 固件库路径



用户库路径

	ROG Express - DE1 - [代码:main_2*]										
□ 文件(F) 編辑(E) 视图(V) 工程(P) 建立(B) 联机(N) 附加(X) <u>?</u>											
<b>ジ</b> 打井エー	保存放大 缩小 工程	★ 編輯向导 消息 交		制作 调试开/关 工程控							
	□ □ □ □ □ □ □ □ □ □ □ □ □ □	'LIABRARY1' 类型  版本   库类型 ◎ 用户库	<b>访问</b> □受保护								
石鴫点 并行 石线圈	□ ■ main_2* ■ FB2* ■ FB2* ■ FB2* ■ FD2* ■ FD2*	○ 固件库 库路径: C:\Users\&dministra	ator\Desktop\LIABRARY1.mwe								
左侧电 右侧电 触点类型			<b>确定 取消</b> 应用 (A) .								

固件库和用户库的路径属性告诉我们,工程文件本身并没有包含固件库文件和用户库文件, 工程内只是提供了一个路径调用库文件。因此,当我们把工程文件拷贝给另一方使用时,应 该把固件库文件和用户库文件一起提供给对方.当接收工程的用户在电脑上上打开工程后, 由于库路径并不会自动更新,所以应该先删除原有的固件库和用户库。然后重新添加这两个 库,使库路径指向用户电脑上正确库文件所在的位置。

■ 文件(E)	编辑(E) 视图(V)	工程(P) 建立	Z(B) 联机(N)	附加(X)	2				
<b>6</b> 打开工	日本 日	⊂ 缩小	2000 工程树 绯	<b>ジ</b> 議 向 导	消息 交叉参考	5 监视窗口 变重工	<b>全国</b> 変単	新作 调试开/关 工程控	
<b>油()</b> 网络	● 库 ● 数据类型 ● ● 逻辑POU ■ main_2 ■ FB2* ■ FC2*	*	删除固	件库、	,用户库后	,重新添加周	车 ·		
						· ·	· ·	· · ·	· ·
右线圈								· ·	
右侧电							•		
					图 7.1	.11			

如果不重新添加库直接编译时往往会提示如下信息

 · 编辑(E)视图(V)	工程(2)建立	之(B) 联机( <u>(</u>	N) 附加(X)	2				2		
保存 放大	缩小 RY1 居类型 PAxis_Type POU main FB1 FB2 open_LIB	<u></u>	<u>编辑向导</u>	<u>消息</u>	交叉要考 MULTIPROC	当 监视窗口 G Express 库山ABRAH	变里工 RY1不存在! 确定		制作	调词 一



也可能提示

中文件(E)	编辑 医视图仪 工程 图 建立 图 联机 图 附加	∞ 2					
初井工				明作 调试开/关 工程控	Z		
同語 石鹸点 井行 石製題 左別し 二 一 一 一 一 一 一 一 一 一 一 一 一 一	使 任 通 日本語本末日 日 日本語本末日 日 日本語本末日 日 日本語 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明白 明 明白 明白 明白 明白 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日					80 20 80 80 80 80 80 80 80 80 80 80 80 80 80	
松西类型	( <b>₽</b> ± <b>1</b> ) #4	(1) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) * (2) *					
10 😵 未声日 4 7	∯rtof MC_Cv+er' 1						

图 7.113

执行完上面步骤以后,接收工程的用户可以正常使用工程。 注.如果接收工程的用户重新编辑了用户库工程,同样需要删除然后重新添加库才能保证用 户库工程能被编译通过,用户库工程编译以后才能在调用工程中作为用户库使用。建议加密 用户库工程,避免错误操作。

#### 压缩文件操作

当要输出的工程含有较多的文件时(比如工程含有多个库文件),也可以选择把所有的文件 压缩为一个文件打包输出,打包后的文件后缀为".zwe"。

首先打开一个工程,选择"文件">"工程另存为/压缩为",在"保存类型"一栏下拉,选择"已压缩工程文件(\*.ZWE)",再把"压缩选项"中的内容全部勾选,选择存储目录后 点击"压缩"。如下图所示



压缩后得到文件如下图所示



图 7.115

当用户接收到该压缩工程文件后,双击即可解压该工程。解压过程中弹出的对话框全部选择 "是"或者"允许",解压完成后得到一个新工程,新工程自动链接了新的固件库和用户库, 无需用户重新连接。

# 8. 任务设计

# 8.1. 任务设计步骤概要

- 1) 规划设计每个程序 POU 的任务类型;
- 2) 规划设计每个程序 POU 的任务优先级;
- 3) 按设计将程序 POU 分配到任务;

# 8.2. 任务类型

为了能更清晰地区分不同功能的任务,方便编程人员使用,MULTIPROG 定义了 4 种 任务类型,分别是 Default、Cyclic、Event 和 System,这些任务的优先级依次升高。

Default 任务:相当于传统 PLC 中运行的任务(或者程序),从控制器上电启动之后

便一直循环往复地运行,可以被其他三种任务打断。

**Cyclic** 任务:顾名思义是周期性的任务,该类型任务具有一个重要属性"运行间隔",每 隔这个间隔时间,任务即被投入运行,可以被 Event 和 System 任务打断。

**Event** 任务: 该类型任务是基于事件驱动的,即在控制器正常运行时该任务处于非激活状态,一旦对应的事件被触发,在情况合适的条件下,该任务就会立刻被激活投入运行。该类型的任务一般用于处理比较紧急的事件。每个任务都需要绑定到一个事件号,每个时间号对应一种特定事件,支持多个 Event 任务绑定到同一个事件号。

System 任务: 该类型的任务也是事件驱动的, 与 Event 任务不同的是, 触发 System 任务运行的时间都是非常紧急的事件, 一般都是控制器运行期的异常(Exception)。System 任务也有对应的事件号, 不过不需要用户手动绑定, 因为这些事件号和对应的事件都是预先 定义好的。System 任务可以抢占任何其他类型任务的运行。

图 6.12 给出了 Default 任务的运行模式,图 6.13 表示了 Cyclic 任务和 Event 任务、 System 任务的运行模式及相互之间的优先级关系。





图 6.12 Default 任务运行模式示意图

图 6.13 其他三种任务的运行模式和优先级关系

## 8.3. 任务优先级

Cyclic 任务和 Event 任务可以设置 0~15 的优先级。任务按照优先级抢占执行。 Default 任务优先级低于 Cyclic 任务和 Event 任务,当然也低于 System 任务。 System 任务优先级高于 Cyclic 任务和 Event 任务。

根据以上任务优先级,将不同 POU 的控制逻辑添加到对应优先级的任务中。

## 8.4. 将程序分配到任务

# 8.4.1. 增加删除任务

一个资源可以支持不止一个任务的运行,即在宏观上,所有这些任务可以同时运行。添加任务可以通过在"Tasks"节点上点击鼠标右键,在快捷菜单中选择"插入…"->"任务"实现,如图所示。



图 8.3 添加任务

在弹出窗口中填写添加任务的名称并选择任务的类型,点击"确定",如图所示。

插入		23
名称(N):	类型	确定
T_anr	○ 配置(C)	 
程序类型(M):	<ul> <li>资源(R)</li> <li>(R)</li> </ul>	47.713
· · · · · · · · · · · · · · · · · · ·	● H分(I) ○ 程序实例(P)	帮助(H)
任务类型(Y):	○ 描述(D)	
CYCLIC •	<ul> <li>● 变重(V)</li> <li>● FB实例</li> </ul>	
	模式:	
	◎ 插入(I)	
□排除在编译之外(X)	◎追加(A)	

图 8.4 编辑所添加任务的属性

注意:任务的名称长度是有限制的,不能超过7个字符,并且前6个字符中必须有1个 是不同的;目前软件版本最多支持10个不同优先级的任务。

当任务不再需要时,可以在工程树节点上点击右键,在弹出的快捷菜单中选择"删除"项 将任务删除。

## 8.4.2. 向任务分配程序
1) 鼠标右键单击"Tasks"。





2) 在弹出菜单选择"插入"->"程序实例..."。

o -	CVCI TC	· ·	
	插入(1)		任务(1)
	删除( <u>D</u> )	Delete	程序实例(P)
*	剪切( <u>C</u> )	Ctrl+X	
∎ <sup>₽</sup>	复制( <u>O</u> )	Ctrl+C	
r <sub>e</sub>	粘贴(P)	Ctrl+V	
	属性( <u>R</u> )		
7	设置( <u>S</u> )		
		图	

3) 在弹出的窗口中,"程序类型"下拉列表选择程序类型,"程序实例"填写该类型程序 实例化的名称,然后单击"确定"即可向任务添加程序实例。

插入		<b>—</b>
程序实例(N): 程序类型(M): Demo1 ▼	类型 ◎ 配置(C) ◎ 资源(R) ◎ 任务(T) ◎ 程序实例(P) ◎ 描述(D) ◎ 变里(V)	确定 取消 帮助(H)
□ 排除在编译之外(X)	模式: ● 插入(I) ◎ 追加(A)	

图 8.7

【注意】必须在逻辑 POU 中已经编写了程序,才可以在任务中插入程序实例操作时,从 程序类型的下拉列表中选择逻辑 POU 中的程序,并将其实例化。



图 8.8

4) 鼠标右键单击选中的任务下的程序实例, 在弹出菜单中选择"删除"即可删除添加的程序 实例。

### 9. 调试与运行

#### 9.1. 步骤概要

控制程序编写后,可以按照以下步骤进行调试运行,以确保控制逻辑符合控制需要。

- 1) 按照第 6 章描述内容使用,创建了 PLC 程序,配置了从站,并确认已配置 MULTIPROG 与 i5 的连接。
- 2) 确认 i5 与从站连接并处于上电运行状态。
- 3) 编译制作通过并下装程序。
- 4) 冷启以使得 PLC 处于运行状态。
- 5) 点击工具栏中的"调试开/关"按钮,从编辑模式切换到调试模式。
- 6) 运用本章描述的调试手段进行 PLC 程序调试。

### 9.2. 连接控制器

可通过以下两种方式连接到控制器:

(1) 在线扫描 在菜单栏单击【联机】-【扫描控制器】打开扫描控制器窗口



图 9.1

窗口中将列举出所有可以被连接的控制器的基本信息(包括处理器类型、处理器版本、IP 地址)。

双击控制器图标,可以连接到相应的控制器,同时打开控制控制对话框,并将工程 设置成与所连接控制器相同的类型和版本

(2) 手动设置

a 手动选择控制器类型。在菜单栏单击【附加】-【选择控制器】打开选择控制器窗口,选择对应控制器的类型,及版本号,双击控制器图标,关闭选择控制器窗口



图 9.2

b 在工程树窗口点击左下角硬件选项卡,切换到物理硬件配置界面。在【资源】节点右键单击,弹出右键菜单,单击【设置】,弹出资源设置对话框。

<ul> <li>□ ● 物理硬件</li> <li>□ ● ■ 配置: eCLR</li> </ul>			
□····  资源 : Leetro_I5_1	1	臿入(1)	•
任务 : CYCL	r f	删除( <u>D</u> )	Delete
	• •	鹑切( <u>C</u> )	Ctrl+X
IO_Configuration	. 🗗 1	夏制( <u>O</u> )	Ctrl+C
_	re i	钻贴( <u>P</u> )	Ctrl+V
	]	从外部变量创建	全局变量(G)
		从全局变量中更	新外部变量(⊻)
	1	创建Bootproie	ct
		<b>冨性(R)</b>	
		まに <u>(()</u> 分置(S)	
3 工程 2 建築			



	3気漆(2重		24
通信			
8 <b>2</b>	TCP/IF	p.:	•
创教	192,16	81.88 p41100	
版本			
建立设置:	eCLR I	EtherCAT	•
<ul> <li>● 更新前</li> <li>○ 不更新</li> </ul>	御(司(D) [N]		
● 更新新 ◎ 不更新	御(司(B) [N]		
<ul> <li>更新前</li> <li>一 不更新</li> <li>一 不更新</li> <li>在线更新</li> <li>时间间隔</li> </ul>	朝(司(B) [N] 10	変形 (花園: 0. 60000)	
<ul> <li>● 更新第</li> <li>○ 不更新</li> <li>         在线更新         时间间隔     </li> <li>         は頃间隔         摘導器法项     </li> </ul>	翻(司(B) [N] 10	<b>雅</b> 谷 (石)路 0. 60000)	
<ul> <li>更新第</li> <li>不更新</li> <li>不更新</li> <li>市は更新</li> <li>时间间隔</li> <li>助间间隔</li> <li>端译器选项</li> <li>二 堆材松</li> </ul>	朝(司(D) (N) 10 変	<b>変</b> せ (石酸 0.0000)	
<ul> <li>更新新</li> <li>不更新</li> <li>不更新</li> <li>市場回帰苑</li> <li>由局间境苑</li> <li>論译器违项</li> <li>二 堆积检</li> <li>又 数组边</li> </ul>	翊(司(D) [N] 10 変 界检查	爱好 (23% 0.0000)	

图 9.4

c 在通信参数中,填写目标控制器的 IP 地址。点击【确定】保存通信参数、 并关闭窗口。

d 点击工具栏的图标 ┛ 打开工程控制对话窗口。

### 9.3. 联机调试

### 9.3.1. 调试模式

在线监视 PLC 的运行,能动态地观察各个变量的值及程序的运行状态。 1) 点击工程控制对话框中的"冷启"按钮,保证 PLC 处于运行状态。

- 2) 点击工具栏中的"调试开/关"按钮,从编辑模式切换到调试模式。
- 3) 图 9.5

### 9.3.2. 强制

将一个变量锁定在某个值上,在强制期间该变量都维持该值不变,直到复位强制。强制操作只对 I/O 变量有效。

1) 在调试模式下,双击变量,出现"调试:资源"对话框,如下图所示:

	Mot	or Start	设置(S)
直			
	TRUE	FALSE	全部复位(A)
强制	<b>9(0)</b> 夏位3 夏位强制列表(L)	量制(E) 覆盖(₩)	<ul> <li>●标准(D)</li> <li>●标准(D)</li> <li>● 市/进制(M)</li> <li>● 二进制(Y)</li> <li>○ 二进制(Y)</li> <li>&gt; 实数值</li> <li>⑦度(I): 精度(P):</li> <li>3</li> <li>7</li> <li>✓ IEEE 格式化(M)</li> </ul>



- 2) 选择 "TRUE", 然后点击"强制", 结果"Motor\_Start"变量将被强制为"TRUE";
- 3) 若要取消强制变量,则在"调试:资源"对话框中,点击"复位强制列表"即可取消。

### 9.3.3. 覆盖

由用户临时性地将一个值赋给一个变量。该值将一直保持到程序在下一个程序循环中用原值再次覆盖掉这个值的时候。

- 1) 在调试模式下,双击变量,出现"调试:资源"对话框,如上图所示。
- 2) 点击"覆盖",即可实现该操作。

### 9.3.4. 交叉参考窗口

用于调试和错误隔离

- 1) 点击工具栏中的'交叉参考窗口'图标,以打开交叉参考窗口。
- 将光标放于交叉参考窗口中,并用鼠标右键点击窗口背景,以打开其上下文菜单,如下 图所示。

	•	变量	$\sim$	POV/工作单	访 🗸	允许对接		类型	初	注释
Ŧ						隐藏				
						过滤器(I)				
						转到某位置 (G)				
						转到定义之处(10)				
						第一个参考(图)				
						上一个参考(2)	Ctrl+Shift+F12			
口 (昭						下一个参考(图)	Alt+F12			
大参し						最后一个参考 (L)				
: 22	<					建立交叉参考(B)				>

图 9.7

3) c,点击"建立交叉参考"菜单项,创建的交叉参考列表如图所示。

X	▶ 茨量	$\sim$	POV/工作单	访问	命令	I/0	全局路径	类型 初	注释 🔼
1	nActual_Time		配置.资源.Global_Var				配置 资源	TIME	
	Actual_Time		Untitled Untitled	写			配置 资源	TIME	
	Actual_Time		Untitled UntitledV				配置 资源	TIME	
	motor		配置.资源.Global_Var			%QXO. O	配置 资源	BOOL	
	notor 🏠		Untitled Untitled	写	-(S)-	%QXO. 0	配置 资源	BOOL	
	notor 🏠		Untitled Untitled	读	-   -	%QXO. O	配置 资源	BOOL	
	notor 🏠		Untitled Untitled	读	-   -	%QXO. 0	配置 资源	BOOL	
	notor 🏠		Untitled Untitled	写	- (R) -	%QXO. O	配置 资源	BOOL	
	<b>D</b> Motor		Untitled UntitledV			%QXO. O	配置 资源	BOOL	
<b>王</b>	notor_start 🏠		配置.资源.Global_Var			%IXO. O	配置 资源	BOOL	
X #	notor_start 🏠		Untitled Untitled	读	-   -	%IXO.0	配置 资源	BOOL	~
₩	<								>

图 9.8

- 4) 双击交叉参考窗口中的某个变量,将会打开使用了这个变量的工作单,并加亮此变量。
- 5) 另外,如果您在工作单中标记一个变量,交叉参考窗口中的相应变量也将被标记。
- 6) 过点击"交叉参考"图标来关掉交叉参考窗口。

#### 9.3.5. 变量监视窗口

在不打开使用了该变量的工作单情况下,就可以监视该变量当前值。 1) 点击"调试开/关"图标,将工作单切换到调试模式。

2) 点击菜单栏中"监视窗口"图标,打开变量监视窗口。

- 3) 图 9.9
- 4) 在工作单中,用鼠标右键点击某个变量,以打开其上下文菜单,并选择'添加到监视窗口',以便将此变量插入列表中。

例:将变量 Motor\_Start、Pressed、Actual\_Time 添加到变量监视窗口中,如下图。

变量	值	默认值	类型	实例
Motor_start	FALSE		BOOL	配置. 资源. Task. Untitled. Motor_start
Pressed	0		INT	配置. 资源. Task. Untitled. Pressed
Actual_Time	0.000		TIME	配置.资源.Task.Untitled.Actual_Time
21				
<b>●</b> \ <b>♥</b> atch 1	A Watch 2 A Watch 3	A Watch 4 A Watch 5 A	Watch 6 ∧	Watch 7 A Watch 8 A Watch 9 A Wat
	Motor_start Fressed Actual_Time	Motor_start Pressed Actual_Time 0.000 Actual_Time 0.000 Vatch 1 / Vatch 2 / Vatch 3 ,	Motor_start FALSE Pressed 0 Actual_Time 0.000	Motor_start     TALSE     DOIL       Pressed     0     INT       Actual_Time     0.000     TIME

图 9.10

现在可以使用 I/O 仿真程序来操作各触点,在监视窗口中观察变量值的变化。

#### 9.3.6. 逻辑分析器

### 9.3.6.1. 功能

逻辑分析器是一个功能强大的工具,用于记录 PLC 上的变量值(以连续模式或限定周 期模式),并且以曲线的形式图形化地显示它们,其 x 轴(时间)和 y 轴(变量值)上都有可调节 的刻度。

#### 9.3.6.2. 记录模式

逻辑分析器提供了两种记录模式:

1)限定周期的记录:值被记录了指定数目的周期。在记录过程中,不更新逻辑分析器的显示。值由运行期系统存储,并且在完成记录之后被上传。

2) 连续记录:这个模式对于 PLC 应用程序的长期诊断很有用。与限定周期模式相比,、 该曲线的显示在记录过程中不断更新(即:连续的)。记录过程可以由用户来停止,也会 在所使用的 PC 的存储器容量满了的时候停止。

#### 9.3.6.3. 如何使用

- 1) 使用条件: 在使用逻辑分析器之前, 要确保将已编译工程下装到 PLC。PLC 处于运 行状态。编程系统处于联机模式。
- 2) 打开逻辑分析器: 主界面单击"视图"->"逻辑分析器",随后可右键弹出页面打 开设置栏。或者在主界面单击"联机"->"逻辑分析器",也可以进入设置栏。如 下两图所示。







图 9.12

3) 添加变量:打开相应变量表,选中相应变量,右键点击"添加到逻辑分析器", 随后在上图设置栏内,点击"已连接变量..."对需要监控的变量进行勾选或清除。 (默认全勾选)如下两图所示。





图 9.13

4) 配置: 在上图设置栏内点击"触发器配置",进入下图界面。设置触发条件和记 录模式。在这里我们选择连续采样模式,同时将触发条件设置为变量等于自身, 让其在开始后能一直采集(详细设置请参考 MULTIPROG "PLC 帮助")

<ul> <li>● 触发器配置</li> <li>● 触发器配置:</li> <li>名称(M):</li> </ul>	☑ 连续记录(C)	8
LADATA LADATA	最大记录周期: 0 天 12 小时 28 分钟 58 秒 971	(约)
	触发条件 <ol> <li>2. 变量</li> <li>任务.LL.C002</li> <li>相等</li> <li>支量</li> <li>任务.LL.C002</li> </ol>	•
新建(W) 去除(V)	数据采集         ● 与任务同步(S):       任务         ◆         传输之前记录的最大周期数(2200)(N):       200	
	确定取消	

图 9.15

- 5)记录和保存:选择需要监控的变量(最多 10 个变量),点击设置栏内"开始记录", 进行记录。记录完毕后,请点击"停止记录"。如果需要,您可以将已经记录的 数据导出到一个 \*.csv 文件,以及/或者将曲线保存到一个文件,以便以后再次加 载和显示它们。要删除所有曲线,请选择"清除曲线"。
- 6)参考:详细操作流程(加载触发条件、设置 X 轴单位等)请参考 MULTIPROG "PLC 帮助",本文不作过多赘述。
- 9.4. 正式运行

#### 9.4.1. 下装引导工程实现上电启动运行的 PLC 程序

工程中的控制程序调试完成后,则可以将工程下装成为引导工程,在 i5 上电时自动加载运行,即正式运行时脱离 MULTIPROG 独立运行。

下装引导工程意味着:将工程(包括所有的编译数据)发送到所连接的目标系统上,并保存在永久存储器中。在发生电源故障或者 PLC 上电之后,会自动执行引导工程。

设置工程为引导工程的方法为:

1) 主界面单击"联机"->"工程控制..."菜单,打开工程控制面板对话框。

资源	
状态: 运行	
停止(S)	冷启(0)
复位(R)	暖启(W)
错误(E)斜耀	热启(T)
下装(D)	
更多(M)	信息()
关闭(C)	帮助(H)

图 9.16

2) 单击"更多..."按钮,打开"下装选项"对话框。单击"下装"按钮,则将当前工程(称为A工程)下装为引导工程;但关闭该窗口回到工程控制窗口,执行下装工程操作(称为B工程),则只是将B工程下装到 i5 内存(相比于作为引导工程下装速度更快),

下次上电仍加载运行 A 工程。

装选项	"窗口和"工程控制"窗口"下装"按钮执行下装,均将下装的工程设置为	引导工程。
Í	下装选项	
	选项	
	☑ 作为引导工程永久驻留(P)	
	🔲 包括源代码(S)	
	☑包括用户库(U)	
	☑包括页面布局(L)	
	☑ 包括后端代码(8)	
	□ 总是允许下装过程中违反实时性	
	引导工程	
	下装(D) 激活(A) 删除(E)	
	(美田の) (お助い)	

3) 若选中"作为引导工程永久驻留"复选框。单击"关闭"按钮关闭窗口,则以后通过"下

图 9.17

 再次打开"更多..."按钮,打开"下装选项"对话框。单击"激活"按钮,将引导工程 激活。

### 9.4.2. 删除上电启动运行的 PLC 程序

已设为上电启动运行的 PLC 程序,可以通过以下方法删除。

1) 主界面单击"联机"->"工程控制..."菜单,打开工程控制面板对话框。

资源	
状态: 运行	
停止(S)	冷启(0)
复位(B)	暖启(W)
错误EPA 耀	热启(T)
下装(D)	
更多(M)	信息(1)
关闭(C)	帮助(H)

图 9.18

2) 单击"更多..."按钮,打开"下装选项"对话框。单击"删除"按钮即可删除引导工程。 删除引导工程后,重新上电系统不再加载运行工程,必须通过"工程控制"窗口,执行下 装启动操作。

【注意】若"作为引导工程永久驻留"复选框被选中,则在删除引导工程后,无论在哪个窗 口再执行下装,则下装工程仍将作为引导工程。



图 9.19

### 10. 连接人机界面

#### 10.1. 与人机界面的连接步骤概要

i5 系列支持 Modbus RTU 及 Modbus TCP 两种 Moudbus 通信机制。

#### 10.2. 连接配置

#### 10.2.1. Modbus RTU 串口通信

本串口支持 RS-485 RS-232 两种方式。参考硬件手册,连接 HMI 的串口与 i5 控制器的串口。

i5 控制器 Modbus 串口通信参数可在【视图】->【通信参数】 中进行配置, 配置参数 随工程下装, 并重启后生效。

根据标准 Modbus RTU 协议即可与 i5 控制器进行通信。

#### 10.2.2. Modbus TCP 网口通信

将 i5 控制器 的 EtherNet 与 HMI 的网口直接连接或经过路由器连接。

i5 控制器 Modbus 网口通信端口可在【视图】->【通信参数】中进行配置,配置参数随 工程下装,并重启后生效。

根据标准 Modbus TCP 协议即可与 i5 控制器进行通信。

#### 10.3. MODBUS 地址

ModBus 对应的寄存器列表见附表 1。

### 11. 与 PC 组合控制

### 11.1. 与 PC 组合控制步骤概要

i5 主要按照控制程序实现对设备的实时控制,在控制过程中,若需要高密集计算的处理,如视觉处理,对内存和 CPU 的要求比较高,为提高处理效率,可以将这部分运算处理通过 PC 编程实现,将处理结果通过 API 接口调用和通信传输到 i5,影响 i5 的实时控制逻辑。

因 Windows 为非实时的操作系统,且基于标准以太网 TCP/IP 连接也非实时,因此应当 合理分配 PC 运算处理内容和 i5 的控制逻辑内容,避免频繁地通信交换数据,而影响运行效 率和 PLC 控制的实时性。

PC 的处理和 i5 的控制逻辑应当交替进行,PC 作为控制主体,通过共享变量将处理结果数据以及命令传递给 i5, i5 的 PLC 程序根据数据和命令字执行实时控制逻辑,并将状态更新到共享变量,PC 通过共享变量的访问读取查询 i5 的控制逻辑执行状态,进行下一个应用处理周期的运算处理。



实现 i5 与 PC 的组合控制步骤如下:

- 1) 根据总需求分解需要 PC 编程处理和 i5 编程处理的应用逻辑部分;
- 2) 根据需求分解,设计 PC 应用和 PLC 程序需要交换的数据和命令及状态,定义共享变量;
- 3) 使用 Visual C++或者 Visual C#, 编写 windows 应用程序, 实现 PC 处理的部分;
- 4) 调用适用于 i5 包含 API 的 DLL 与 i5 进行通信,对 MODBUS 和 PLC 共享的变量进行读 写;
- 5) 适用 IEC61131-3 标准规范的语言编写 PLC 控制程序,对 MODBUS 和 PLC 共享的变量进行读写。
- 6) 运行 Windows 应用程序的 PC 和运行 PLC 控制程序的 i5 联调。

### 11.2. 连接配置

PC 端通过 Modbus TCP 协议与 i5 通信进行数据交换,实现组合控制。

将 i5 控制器 的 EtherNet 与 PC 的网口直接连接或经过路由器连接。

i5 控制器 Modbus 网口通信端口可在【视图】->【通信参数】中进行配置,配置参数在 随工程下装,并重启后生效。

调用 API 即可与运行中的 i5 控制器 进行通信。

Modbus 通信 API 基本调用流程如下:

- 1) 调用 LTI5\_ NewLink 函数 获取连接句柄;
- 2) 调用 LTI5\_ ModbusConnect 函数建立与控制器的连接。
- 3) 执行步骤 2 无错误返回,可调用 LTI5\_ModbusWriteRegister 将数值写入控制器 Modbus 寄存器中,或者调用 LTI5\_ModbusReadInputRegisters 从控制器中读取 Modbus 寄存 器值。其他读写操作及函数使用见 API 接口函数说明。
- 4) 调用 LTI5\_ ModbusClose 可断开与控制器的连接。

数据采集 API 基本调用流程如下:

- 1) 调用 LTI5\_ SamplingInit初始化数据采集;
- 2) 调用LTI5\_SetSamlingRange 设置采集数据的范围
- 3) 调用LTI5\_SetSamplingTime 设置采集的时间参数
- 4) 调用LTI5\_SamplingStart 开始采集数据
- 5) 调用LTI5\_GetSamplingData 获取控制器采集到的数据,用户根据得到的数据进行解析
- 6) 调用LTI5\_SamplingStop停止采集数据

#### 11.3. 函数库的 API 定义

【DLL 接口函数的说明,函数名列表,详细说明见指令手册】

序号	接口函数
1	LTI5_LINK LTI5_NewLink();
2	void LTI5_DeleteLink(LTI5_LINK tLink);
3	void LTI5_SetDebug(LTI5_LINK tLink,BOOL blsDebug);
4	UINT32 LTI5_ModbusConnect(LTI5_LINK tLink,const CHAR8* strParm);
5	UINT32 LTI5_ModbusClose(LTI5_LINK tLink);
6	UINT32 LTI5_ModbusReadInputBits(LTI5_LINK tLink , UINT32 dwAddr, CHAR8 *pchValue)

7	UINT32 LTI5_ModbusReadCoils(LTI5_LINK tLink, UINT32 dwAddr, CHAR8 *pchValue, UINT32 dwBitNum)
8	UINT32 LTI5_ModbusWriteCoil(LTI5_LINK tLink , UINT32 dwAddr, CHAR8 chValue)
9	UINT32 LTI5_ModbusWriteCoils(LTI5_LINK tLink , UINT32 dwAddr, CHAR8 *pchValue, UINT32 dwBitNum)
10	UINT32 LTI5_ModbusReadInputRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32
	dwRegNum)
11	UINT32 LTI5_ModbusReadRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32
	dwRegNum)
12	UINT32 LTI5_ModbusWriteRegister(LTI5_LINK tLink , UINT32 dwAddr, UINT16 wValue)
13	UINT32 LTI5_ModbusWriteRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32
	dwRegNum)
14	UINT32 LTI5_ModbusWriteReadRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32
	dwRegNum)
15	UINT32 LTI5_SamplingInit(char *pAddr);
16	UINT32 LTI5_SamplingExit(void);
17	UINT32 LTI5_SamplingStart(void);
18	UINT32 LTI5_SamplingStop(void);
19	CHAR8* LTI5_GetSamplingData();
20	UINT32 LTI5_SetSamplingTime(UINT16 wTime_ms, UINT16 wTimePeriod, UINT16 wPreTime_ms);
21	UINT32 LTI5_SetSamplingRange(UINT16 wStartAddrl, UINT16 wNumberl, UINT16 wStartAddrQ, UINT16
	wNumberQ, UINT16 wStartAddrM, UINT16 wNumberM);

11.3.1.函数库的 API 接口描述

### IF-1. LTI5\_NewLink

语法	LTI5_LINK LTI5_NewLink();
输入	无
输出	无
返回值	连接对象的句柄 NULL:错误 非NULL:正确
描述	创建新连接对象
上级	
下级	

# IF-2. LTI5\_ DeleteLink

语法	void LTI5_DeleteLink(LTI5_LINK tLink);
输入	tLink - 由LTI5_NewLink()返回的句柄
输出	无
返回值	无
描述	删除连接句柄,释放内存
上级	

下级

### IF-3. LTI5\_SetDebug

语法	void LTI5_SetDebug(LTI5_LINK tLink,BOOL bIsDebug);
输入	tLink - 由LTI5_NewLink()返回的句柄 hlsDebug 调试标志 0. 非调试模式 1. 调试模式
输出	无
返回值	无
描述	设置调试模式 在调试模式下可打印输出通信数据帧
上级	
下级	

#### IF-4. LTI5\_ ModbusConnect

语法	UINT32 LTI5_ModbusConnect(LTI5_LINK tLink,const CHAR8* strParm);
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄 strParm-连接参数 格式为-ip{addr}:{port} 例如: -ip192.168.1.88:1502
输出	无
返回值	0: 正确 非0: 错误代码
描述	建立MODBUS连接
上级	
下级	

### IF-5. LTI5\_ ModbusClose

语法	UINT32 LTI5_ModbusClose(LTI5_LINK tLink);
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄
输出	无
返回值	0: 正确 非0: 错误代码
描述	关闭MODBUS连接,关闭后MODBUS读写接口不再可以使用
上级	
下级	

### IF-6. LTI5\_ModbusReadInputBits

### 语法 UINT32 LTI5\_ModbusReadInputBits(LTI5\_LINK tLink, UINT32 dwAddr, const CHAR8 \*pchValue)

输入	tLink - 由LTI5_NewLink() 输出参数返回的句柄 dwAddr – Modbus地址
输出	pchValue – 读回的值;
返回值	0: 正确 非0: 错误代码
描述	读取输入位变量
上级	
下级	

### IF-7. LTI5\_ModbusReadCoils

语法	UINT32 LTI5_ModbusReadCoils(LTI5_LINK tLink, UINT32 dwAddr, CHAR8 *pchValue, UINT32 dwBitNum)
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄 dwAddr – Modbus地址 dwBitNum – 要从dwAddr地址开始读取的连续线圈个数
输出	pchValue – 读回的值,从[0]单元开始依次存放读回的线圈状态,0或者1;
返回值	0: 正确 非0: 错误代码
描述	读取多个输出或者内部线圈值
上级	
下级	

## IF-8. LTI5\_ModbusWriteCoil

语法	UINT32 LTI5_ModbusWriteCoil(LTI5_LINK tLink , UINT32 dwAddr, CHAR8 chValue)
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄
	dwAddr – Modbus地址
	chValue – 要写到dwAddr地址的值
输出	无
返回值	<b>0:</b> 正确
	非0:错误代码
描述	写一个输出或者内部线圈值
上级	
下级	

### IF-9. LTI5\_ModbusWriteCoils

语法	UINT32 LTI5_ModbusWriteCoils(LTI5_LINK tLink , UINT32 dwAddr, const CHAR8 *pchValue, UINT32 dwBitNum)
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄

	dwAddr – Modbus地址		
	pchValue – 要写到dwAddr地址开始的值,从[0]单元开始依次存放要写的线圈状态值,0或者1;		
	dwBitNum -要写到dwAddr地址开始的连续线圈个数		
输出	无		
返回值	0: 正确		
	非0:错误代码		
描述	写多个输出或者内部线圈值		
上级			
下级			

### IF-10.LTI5\_ModbusReadInputRegisters

语法	UINT32 LTI5_ModbusReadInputRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32 dwRegNum)	
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄 dwAddr – Modbus地址 dwBitNum -要从dwAddr地址开始读取的连续寄存器个数	
输出	pwValue – 返回读取到的dwAddr地址开始的值,从[0]单元开始依次存放读取到的寄存器值;	
返回值	0: 正确 非0: 错误代码	
描述	读多个输入寄存器值	
上级		
下级		

# $IF-11.LTI5\_ModbusReadRegisters$

语法	UINT32 LTI5_ModbusReadRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue, UINT32 dwRegNum)
输入	tLink - 由LTI5_NewLink()输出参数返回的句柄 dwAddr – Modbus地址 dwBitNum -要从dwAddr地址开始读取的连续寄存器个数
输出	pwValue – 返回读取到的从dwAddr地址开始的值,从[0]单元开始依次存放读取到的寄存器值;
返回值	0: 正确 非0: 错误代码
描述	读多个输出或者内部寄存器值
上级	
下级	

### IF-12.LTI5\_ModbusWriteRegister

语法	UINT32 LTI5_ModbusWriteRegister(LTI5_LINK tLink	, UINT32 dwAddr, UINT16 wValue)
----	-------------------------------------------------	---------------------------------

输入	tLink - 由LTI5_NewLink()输出参数返回的句柄 dwAddr – Modbus地址 wValue – 要写到dwAddr地址的寄存器值;	
输出	无	
返回值	0: 正确 非0: 错误代码	
描述	写单个输出或者内部寄存器值	
上级		
下级		

### IF-13.LTI5\_ModbusWriteRegisters

语注	UINT32 LTI5_ModbusWriteRegisters(LTI5_LINK tLink , UINT32 dwAddr, const UINT16 *pwValue,	
ATA	UINT32 dwRegNum)	
	tLink - 由LTI5_NewLink()输出参数返回的句柄	
输入	dwAddr – Modbus地址	
	pwValue – 要写到dwAddr地址开始的值,从[0]单元开始依次存放要写的寄存器值;	
	dwBitNum -要写到dwAddr地址开始的连续寄存器个数	
输出	无	
返回值	0: 正确	
	非0: 错误代码	
描述	写多个输出或者内部寄存器值	
上级		
下级		

# ${\sf IF-14.LTI5\_ModbusWriteReadRegisters}$

海外	UINT32 LTI5_ModbusWriteReadRegisters(LTI5_LINK tLink , UINT32 dwAddr, UINT16 *pwValue,	
后公	UINT32 dwRegNum)	
	tLink - 由LTI5_NewLink()输出参数返回的句柄	
*4.)	dwAddr – Modbus地址	
和八	pwValue – 要写到dwAddr地址开始的值,从[0]单元开始依次存放要写的寄存器值;	
	dwRegNum -要读写的dwAddr地址开始的连续寄存器个数	
输出	pwValue – 读回的从dwAddr地址开始的值,从[0]单元开始依次存放读取到的寄存器值;	
返回值	0: 正确	
	非0:错误代码	
描述	读写多个输出或者内部寄存器值	
上级		
下级		

### IF-15.LTI5\_ SamplingInit

语法	UINT32 LTI5_SamplingInit(char *pAddr);	
输入	pAddr –IP地址 例如"192.168.1.88"	
输出	无	
返回值	0: 正确 非0: 错误代码	
描述	采样初始化	
上级		
下级		

### IF-16.LTI5\_ SamplingExit

语法	UINT32 LTI5_SamplingExit(void);
输入	无
输出	
返回值	<b>0:</b> 正确
	非0:错误代码
描述	退出采样
上级	
下级	

### IF-17.LTI5\_ SamplingStart

语法	UINT32 LTI5_SamplingStart(void);
输入	无
输出	无
返回值	0: 正确 非0: 错误代码
描述	开始采样
上级	
下级	

### IF-18.LTI5\_ SamplingStop

语法	UINT32 LTI5_SamplingStop(void);
输入	无
输出	无
返回值	0: 正确
	非0:错误代码
描述	停止采样
上级	

下级

IF-19.LTI5_	GetSamplingData
-------------	-----------------

语法	CHAR8* LTI5_GetSamplingData()										
输入	无										
输出	无										
返回值	NULL: 获取数据错误 非NULL: 数据										
	获取采样数据 获取的数据格式:										
	内容	数	数 据	数据	数 据	数据	0 0 0 0	数据	数据	数据	数据
		据	1	1	1	1		n	n	n	n
		组	索引	时间戳	长度			索引	时间戳	长度	
描述		数		(us)					(us)		
		n									
	长度	2	4	4	1	数据1		4	4	1	数据n
	(字节)					长度					长度
	数据内容 按	·照 N	/13区、1	[区、Q区数	据先后	进行排列					
上级											
下级											

### IF-20.LTI5\_SetSamplingTime

语法	UINT32 LTI5_SetSamplingTime(UINT16 wTime_ms, UINT16 wTimePeriod, UINT16 wPreTime_ms);
输入	wTime_ms-0:连续采样 非0-采样时长(毫秒) wTimePeriod –采样分度 分度为1时 1ms采样一次, 分度为2时 2ms 采样一次
	wPreTime_ms –预采样时间
输出	无
近回店	0: 正确
必回阻	非0:错误代码
描述	设置采样时长,采样间隔时间等
上级	
下级	

# IF-21.LTI5\_SetSamlingRange

语法	UINT32 LTI5_SetSamplingRange(UINT16 wStartAddrI, UINT16 wNumberI, UINT16 wStartAddrQ,
	UINT16 wNumberQ, UINT16 wStartAddrM, UINT16 wNumberM);
	wStartAddrI-采样I区域开始地址,
输入	wNumberI-采样I区域数据长度,
	wStartAddrQ-采样Q区域开始地址,

	wNumberQ-采样Q区域数据长度,							
	wStartAddrM-采样M3区域开始地址,							
	wNumberM-采样M3区域数据长度,							
	以上地址均对应为PLC 程序内变量分配的地址							
输出	无							
近回店	<b>0:</b> 正确							
必凹沮	非0:错误代码							
H# 3-2-	设置采样范围							
油坯	各区域总长度之和不得超多128字节							
上级								
下级								

# 12. 系统维护

### 12.1. 固件升级

1) 连接到控制器,然后打开菜单【联机】->【版本信息】

版本信息				2
刷新 控制器类型Lee	tro_I5_1213#	反本1.8.2目标版本 Leetro	_i5_1213_V1.8.2	• 控制器升级
名称	版本	编译时间		
LOG	1.4.1.2663	2019/11/20 16:2:52		
HardWare	0.0.0.1213	2018/1/1 0:0:0		
Uboot	1.3.0.0	2019/7/8 7:37:44		
OS Image	1.5.0.3208	2020/7/30 15:4:34		
FPGA	1.2.0.102	2018/3/2 12:6:23		
OS Init	1.5.0.3365	2020/8/28 11:12:38		
ModuleManager	1.8.0.3370	2020/8/28 15:9:10		
Sys Performance	1.2.1.2237	2019/5/30 13:54:33		
MasterIO	1.4.2.2840	2020/6/17 18:26:12		
Security	1.3.0.2423	2019/7/25 12:59:55		
Param	1.8.2.3390	2020/9/7 14:5:27		
PCF	1.3.0.2407	2019/7/24 13:39:44		
Modbus	1.8.0.3301	2020/8/13 13:56:20		
mc	1.8.2.3418	2020/9/11 15:49:52		
ControllerIO	1.7.0.2991	2020/7/21 16:6:59		
PlcUnitTest	1.8.0.3295	2020/8/11 17:25:41		
LtCom	1.8.0.3300	2020/8/13 13:55:35		
ECLR	1.8.2.3386	2020/9/4 12:42:0		
Sampling	1.4.2.2747	2019/12/12 16:11:17		
SYS_Main	1.7.0.3052	2020/7/3 18:31:50		

图 12.1

点击"控制器升级",开始升级固件(若控制器当前版本与目标版本相同,则无需升级)。 升级完成后,重启控制器完成升级。

## 12.2. 设置 i5 的系统时间

1) 打开菜单【联机】->【设置系统时钟】。

· 空 控制器设置		
时间与日期:	2019/3/12 13:59:01	更新日期和时间

图 12.2

在【时间与日期】栏修改时间后,单击【更新日期和时间】按钮,则控制器时间将被设定为 修改后的时间。

### 12.3. 设置 i5 的 IP 地址

1) 打开菜单【视图】->【通信参数】,单击 【TCP/IP 设置】选项卡,将 IP 地址设定为新 IP 地址。

CommunicationSetting								
TCP/IP设置 M	TCP/IF设置 Modbus 设置							
DHCP	● 不启用 ◎ 启用							
IP地址	192. 168. 1. 91							
子网掩码	255. 255. 255. 0							
默认网关	192. 168. 1. 1							
远程连接	◎ 不启用 ◎ 启用							

#### 图 12.3

使用 MultiProg 编译该工程;编译完成后在消息窗口中提示"当前 IP 与设置 IP 不一致,若下 装工程 控制器系统 IP 将被更改为:xxx.xxx.(新 IP);确认需要更改 IP 地址则忽略改信息。 使用 MultiProg 下装编译后的工程;下装后 IP 将会被设置为新的 IP。新的 IP 地址将在重新 上电启动后生效。因为 IP 地址改变,导致 MultiProg 与 i5 的连接中断,【工程控制面板】中 显示连接状态为"超时"。

通过 6.4.2 节方法将工程中的资源中的 IP 地址改为新设置的 IP 地址,即可恢复 MultiProg 与 i5 的连接。

### 12.4. 系统日志

系统日志记录了 i5 运行过程中的日志信息,可在日志页面查看。

日志信息包括日志产生的时间,错误代码,以及附加信息;日志错误代码的详细说明见 附录 2:异常代码一览表

日志查看步骤:

1) 【联机】->【日志】显示系统日志。

• 日志				
导出				清除 ☑ 去重
时间	美型	错误代码	重复次数	详细信息
2020/9/18 12:56:0 普	通日志	0136	0	[ECLR] EtherCAT DC mode and Cycle time is not set while ECLR starting, use default
2020/9/18 12:56:0 普	通日志	0109	0	[ECLR] EtherCAT cfg info; DcMode:1 CycleTime:1000
2020/9/18 12:56:2 普	通日志	0124	0	[ECLR] DC mode was set to: DcmMode_BusShift
2020/9/18 12:56:2 普	通日志	0127	0	[ECLR] MC PDO INFO: Size TPDO 70, RPDO 70
2020/9/18 12:56:2 普	通日志	0137	0	[ECLR] EtherCAT master was started for init
2020/9/18 12:56:2 普	通日志	011E	0	[ECLR] EtherCAT BUS state: Cable is linked, the num of slaves scanned is (1), configed is (1)
2020/9/18 12:56:7	般错误	013A	0	[ECLR] PLC cycle time is timeout, be set as: 1000 us, real time:1230 us
2020/9/18 12:56:3	般错误	1512	0	CMD=0x100c0000, FUN=0x15000309: 0, Homing error and end
2020/9/18 12:56:3	般错误	1527	0	CMD=0x0, FUN=0x12040202: 0, Get PDS status error
2020/9/18 12:56:3	般错误	1510	0	CMD=0x100c0000, FUN=0x14010203: 0, Slave is disabled
2020/9/18 12:56:3	般错误	1527	0	CMD=0x0, FUN=0x12040202: 0, Get PDS status error
2020/9/18 12:56:3	般错误	1558	0	CMD=0x100b0000, FUN=0x15000308: 0, Power On Failed



点击【清除】可清除当前日志;

勾选【去重】可去除重复产生的日志;

点击【导出】可将日志导出到文件中保存;



日志会在重新上电后被清除!

### 13. 故障的确认和处理

13.1. 启动过程故障

启动过程指 i5 从上电到进入运行状态的过程,该过程中,日志系统可能还未进入就绪状态,需要通过动作状态指示灯来获知启动过程是否完成以及发生的故障种类,为现场排除故障提供依据。

### 13.1.1.动作状态显示LED

### ● 控制器的动作状态显示LED示意图

在控制器的正面中央,有显示控制器动作状态的 LED。



图 13.1 动作状态显示 LED

### ● 控制器状态显示LED详细描述

LED 显示的内容如下所示 (O: 点亮 0N ◎: 闪烁 BLINK n ●: 熄灭 0FF) BLINK n: 循环闪烁 n 次,一次闪烁为亮 200 毫秒, 熄灭 200 毫秒, 闪烁 n 次后熄灭 1000 毫秒, 然后重复以上过程。

LED 名称	显示色	状态		内容		
		0	点亮	通电中。		
PWR	绿色	•	熄灭	未接通电源。或者电源电压处于容许电压范		
				围外。		
		0	点亮	启动中(电源接通后,正在切换至运行模式		
				或程序模式)		
		O	循环闪烁3次	运行模式下正常状态		
RUN	绿鱼		后熄灭1秒			
NON	>/K 🗅	ullet	熄灭	程序模式下运行停止中、控制器复位中或发		
				生了以下任一异常:		
				● 全部停止故障等级的控制器异常		
				● 控制器处理器异常		
ERROR	红色	0	点亮	通过自诊断检测到以下异常		
				● 全部停止故障等级的控制器异常		
				● 控制器处理器异常		
		O	循环闪烁 n 次	上电初始化过程中不同阶段出现错误(此时		
			后熄灭1秒	RUN 为 OFF,BUSY 为 ON);		
				运行过程中出现不同致命错误(此时 RUN		
				为 ON,BUSY 为 OFF)		
		•	熄灭	控制器正常运行中,处理器复位中。		
		0	点亮	系统上电启动过程中正在对外设进行初始		
BLISV	苦缶			化设置		
0031	Щ	O	闪烁	正在访问内置非易失性存储器。		
		ullet	熄灭	上述除外。		
SD PWR	绿色	0	点亮	正在向 SD 存储卡供电,可使用。		

		O	闪烁	正在执行备份、恢复等。
		•	熄灭	正在停止向 SD 存储卡供电、未安装 SD 存
				储卡或安装的 SD 存储卡的文件格式不支
				持。
	去色	O	闪烁	正在访问 SD 存储卡。
SD BOST	與巴	•	熄灭	未访问 SD 存储卡。

上电启动过程在 OS 加载成功,并加载应用程序,日志系统正常工作前主要通过 LED 显示启动过程以及出现的故障状态;

成功后 RUN 进入循环闪烁 3 次状态后,则通过日志可以获取更多错误信息;

启动	步骤说明	POWER LED	RUN LED	ERR LED	BUSY LED
顺序					
1	上电前	OFF	OFF	OFF	OFF
2	UBOOT 启动完成	ON	ON	ON	ON
3	OS 镜像加载失败,加载备份 OS 镜	ON	ON	ON	OFF
	像				
4	OS 镜像加载成功, Init.out 加载失	ON	OFF	ON	OFF
	败				
5	Init.out 加载 SYS_Main.out 调用	ON	OFF	ON	ON
	SYS_Run()失败				
6	SYS_Run()加载模块文件失败	ON	OFF	BLINK 1	ON
7	SYS_Run()判断已加载模块版本不	ON	OFF	BLINK 2	ON
	匹配				
8	SYS_Run()调用模块初始化失败	ON	OFF	BLINK 3	ON
9	创建 Task 失败	ON	OFF	BLINK 4	ON
10	MODBUS 通讯建立失败	ON	OFF	BLINK 5	ON
11	系统未连接从站进入运行状态	ON	BLINK 2	OFF	OFF
12	系统进入运行状态正常	ON	BLINK 3	OFF	OFF
13	系统 ECAT 周期处理时间超过设定	ON	ON	BLINK 1	ON
	值				
14	系统 PLC 周期处理时间超过设定	ON	ON	BLINK 2	ON
	值				

注:

- 1) BUSY LED 灯闪烁时,请勿关闭控制器的电源。在 BUSY LED 亮灯过程中,将控制器内的 用户程序及设定值备份到内置非易失性存储器中。若关闭电源,则无法备份数据。另外, 还会导致下次启动时全部停止故障等级的控制器异常,并且运行停止。
- 2) SD存储卡访问中(SD BUSY LED 闪烁),请勿关闭电源或拔出 SD 存储卡。否则可能会损 坏数据,导致使用该数据时无法正确动作。从通电中的控制器中取出 SD 存储卡时,请 先按下 SD 卡供电停止按钮,待 SD BUSY LED 和 SD PWR LED 熄灭后,拔出 SD 存储卡。

### 13.2. 调试过程故障

对 PLC 程序的调试过程中的关键错误导致的故障,可通过 MULTIPROG 获取。所有故障的错误代码和信息也可通过系统日志查看。

### 13.2.1. MULTIPROG 无法与 i5 连接显示超时

在 MULTIPROG 与 i5 联机状态下,点击 MULTIPROG 中的"联机"->"工程控制"菜单,弹出 如下对话:

资源	
状态: 超时!	
(S)	冷启(0)
复位(R)	(₩)
错误EC牌	热启(T)
下装(D)	
更多(M)	信息(1)
关闭(C)	帮助(H)

图 13.2

显示超时,无法进行联机操作,可能的原因及处理办法如下:

● 安装运行MULTIPROG的PC与i5的网线连接断开,或接触不良导致通讯中断 检查网线与 PC 和 i5 的网口是否连接,可重新插拔网线确认物理连接正确; 网线是否正确插入了 i5 上标注为 EtherNET 的端口,而非 EtherCAT 的端口;

### ● MULTIPROG中当前工程的IP设置与i5的IP不一致

通过鼠标右键单击工程中的"资源",在弹出的菜单中选择"设置",弹出如下对话框:

ARM_LE_GCC3的资	δ源设置 Σ
通信	
类型:	TCP/IP
<ul><li> 参数: </li></ul>	192 168 1 84 -p41100
版本	
建立设置:	eCLR EtherCAT 🔹
更新建立设置(B	uild settings沂方为: f(A) ŋ)词(B) J)
在线更新 时间间隔:	100 毫秒 范围: 0.60000)
编译器选项 □ 堆栈检查 ☑ 数组边界 □ 经过优化	[ ]检查 
	确定(0) 取消(C) 帮助(H)

在对话框的"通信","参数"一栏,修改 IP 地址为 i5 的 IP 地址,单击"确定"关闭对话框,则可以与 i5 恢复正常连接;

#### 13.2.2. 通过 MULTIPROG 查看错误信息

在 MULTIPROG 与 i5 联机状态下,下装 PLC 程序并启动 PLC 程序后,如果"工程控制窗 口"中的"错误"按钮闪烁,则鼠标单击"错误",即可切换到错误信息显示窗口,显示错误 代码及其错误信息。

#### 13.2.3. 查看错误信息

在与 i5 联机状态下,通过菜单【联机】->【日志】,可以查看系统产生的日志信息,包括日志产生时间,日志代码,附加信息。

#### 13.3. 运行过程故障

正式运行过程,一般不再与 MULTIPROG 连接,可按照第 11 章描述内容,配置 HMI 显示 日志信息,通过 HMI 查看错误信息日志。

### 14. 安全设置

#### 14.1. 安全设置概述

为保证用户 PLC 程序安全,保证在非法设备上不能运行非法拷贝的 PLC 程序,可设置 控制器密码将 PLC 程序进行加密。

### 14.2. 加密设置操作步骤

1, 打开【联机】->【设置控制密码】

叱 设置控制器密码	3		
輸入密码	***	]	
再次输入密码	***	1	

图 14.1

2, 输入需要设置的密码并确认后, 点击设置密码, 完成操作;



# 15. 附录

# 15.1. 附录 1: MODBUS 地址表

Modbus Map地 址	功能及说明	数据类型	初始值
0~9999	输出线圈	BOOL	
$10000^{\sim}29999$	输入线圈	BOOL	
30000~39999	输入寄存器	UDINT	
$40000^{\sim}65535$	保持寄存器	UDINT	
30000~30999	MultiProg I变量	UDINT	
	$(10^{\sim}I1999)$		
	两个I变量占用一个Modbus		
	寄存器,例如10占用30000		
	低8位, I1占用30000高8位		
$31000^{\sim}31999$	MultiProg Q变量	UDINT	
	$(Q0^{\sim}Q1999)$		
	两个Q变量占用一个Modbus		
	寄存器,例如Q0占用31000		
	低8位,Q1占用31000高8位		
$40000^{\sim}49999$	MultiProg M3变量	UDINT	
	$(M0^{M19999})$		
	两个M变量占用一个Modbus		
	寄存器,例如M0占用40000		
	M1占用40000高8位		
	M3 变量具体划分请参考附		
	录3		
138	T: 操作记录使能 F: 操作	BOOL	F
	记录禁止		
139	F->T: 翻到第1页	BOOL	F
140	F->T:翻到最后一页	BOOL	F
141	F->T: 前翻一页	BOOL	F
142	F->T: 后翻一页	BOOL	F
158	F->T: 添加一个账号信息	BOOL	F
159	F->T:更新一个账号	BOOL	F
160	F->T: 删除一个账号	BOOL	F
161	F->T: 切换登录账号	BOOL	F
168	T:运行 F:停止	BOOL	F
169	F->T: 单轴点动	BOOL	F
170	F->T: 单轴停止	BOOL	F
171	F->T: 单轴回零	BOOL	F
172	F->T: 回零停止	BOOL	F

178	T: 由eCLR内核设置为	BOOL	F
	TRUE,表示执行了PLC程序		
	下装		
	F: 由MODBUS客户端获取状		
	态后将其复位成FALSE		
20000	TRUE: PLC运行状态	BOOL	F
	FALSE: PLC停止状态		
20001	TRUE: PLC正在下装状态	BOOL	F
	FALSE: PLC下装完成状		
	态		
20010	TRUE: 控制器系统运行异	BOOL	F
	常,产生了运行错误,具体		
	错误见错误代码		
	FALSE: 控制器系统运行正		
	常		
20011	TRUE:用户程序运行中抛出	BOOL	F
	异常为		
$35000^{\sim}35005$	控制器信息>>序列号	UDINT	0
	(UDINT数组)		
35006~35009	控制器信息>>硬件版本	UDINT	0
	(UDINT数组)		
$35010^{\sim}3513$	控制器信息>>软件版本	UDINT	0
	(UDINT数组)		
35016	控制器信息>>主版本号	UDINT	0
35017	控制器信息>>次版本号	UDINT	0
35020	当前时间>>年	UDINT	0
35021	当前时间>>月	UDINT	0
35022	当前时间>>日	UDINT	0
35023	当前时间>>时	UDINT	0
35024	当前时间>>分	UDINT	0
35025	当前时间>>秒	UDINT	0
35026	系统IP>>类型	UDINT	0
	0 Ipv4 1Ipv6		
$35025^{\sim}35032$	系统IP>>IP地址	UDINT	0
	当IP类型为IPv4时前4个字		
	节有效,存放4字节16进制数		
	当IP类型为IPV6时16个字节		
	有效,存放16字节16进制数		
35033~35040	系统IP>>子网掩码	UDINT	0
	存放方式与IP地址相同		
$35480^{\sim}\overline{35919}$	模块信息(共20个模块)	UDINT	
	第一个模块地址范围为		
	35480~35501 模块详细信息		
	见地址35480~35501		

35480~35483	模块>>版本号	UDINT	0
	依次为主版本号,次版本		
	号,子版本号,Build号		
35484~35489	模块>>编译时间	UDINT	0
	依次为年,月,日,时,分,秒		
35490	模块>>编译模式	UDINT	0
	1 -Debug 2 -Release		
35491	模块名称长度字节数	UDINT	0
$35492^{\sim}35499$	模块名称	STRING	NULL
35500	全局变量数据大小	UDINT	0
35501	模块 ID	UDINT	0
$35920^{\sim}37159$	最近20条日志信息	UDINT	
	第一条日志地址范围为		
	35920~35981 日志详细信息		
	见地址35920~35981		
$35920^{\sim}35921$	日志>>错误代码	DWORD	0
35922	日志>>日志来源	UDINT	0
$35923^{\sim}35928$	日志>>时间	UDINT	0
	依次为年,月,日,时,分,秒		
35929	日志>>附加信息长度(字节)	UDINT	0
$35930^{\sim}35979$	日志>>附加信息	STRING	NULL
35980~35981	日志>>扩展日志码高32位ID	DWORD	0
50020~50025	系统时间设置	UDINT	0
	依次为年,月,日,时,分,秒		

# 15.2. 附录 2: 异常代码一览表

# 15.2.1.系统管理相关

代码	描述	处理方法
	(描述发生了什么异常)	(描述异常产生的原因,以及如何通过
		操作解决这个异常)
0x0000	无错误	
0x0401	内存分配失败	联系供应商
0x0402	参数文件读取失败	重新编译工程,下装
0x0403	参数文件写入失败	联系供应商
0x0404	参数数据类型错误	联系供应商
0x0501	切换工作目录失败	联系供应商
0x0502	卸载网口驱动失败	联系供应商

0x0503	读取系统初配置文件失败	联系供应商
0x0504	打开模块文件失败	联系供应商
0x0505	加载模块失败	联系供应商
0x0506	系统加载模块失败	联系供应商
0x0509	创建任务失败	联系供应商
0x050A	启动任务失败	联系供应商
0x050B	模块初始化失败	联系供应商
0x050C	创建信号量失败	联系供应商
0x050D	获取函数指针失败	联系供应商
		1)连接拟使用的从站
0x050F	未连接 EtherCat 从站	2)检查与从站的通信线路是否正常
0x0511	内存分配失败	联系供应商
0x0512	配置系统 IP 失败	联系供应商
0x0514	获取系统 IP 地址失败	联系供应商
0x0515	获取系统 Mask 地址失败	联系供应商
	设置参数模块回调函数失败(系	联系供应商
0x051A	统 IP 配置)	
	系统 IP 更改为{0},重新上电后	提示信息,确认IP修改
0x051B	生效	
0x0522	配置系统时间失败	联系供应商
0x0523	配置 RTC 时钟失败	联系供应商
0x0526	设置委托函数失败(系统时间)	联系供应商
0x0536	系统升级完成	提示信息,确认升级成功
0x0538	文件重命名错误	检查文件明是否合法
		1) 尝试重新升级
0x0539	升级包解析错误	2) 检查升级包是否完整
0x053A	获取文件校验码错误	检查升级包是否完整
0x053B	升级包打开错误	检查升级包是否正确
0x053C	设置委托函数失败(系统 IP 配置)	联系供应商
0x0541	Modbus 同步数据失败	联系供应商
	Modbus 连接回调函数设置失	联系供应商
0x0542	败	
	设置参数模块回调函数失败	联系供应商
0x0543	(Modbus 配置)	
0x0551	获取 FPGA 版本信息失败	联系供应商
0x0552	获取 FPGA 编译日期失败	联系供应商
0x0553	获取 OS 版本信息失败	联系供应商
0x0554	获取 OS 编译日期失败	联系供应商

# 15.2.2.参数操作相关

代码	描述	处理方法
	(描述发生了什么异常)	(描述异常产生的原因,以及如何通过
		操作解决这个异常)
0x0000	无错误	
0x0301	使用了未分配的内存	请与供应商联系
0x0302	内存分配失败	请与供应商联系
		1) 检查是否在PDO Mapping 中配置了
		该对象,并进行映射
0x0306	参数未定义	2) 检查该从站是否支持该参数
0x0307	复制附加信息出错	请与供应商联系
		检查EtherCAT 从站错误代码,并根据
0x0320	EtherCAT 产生未知错误	错误代码排除问题
		1) 检查通信连接线路,是否与工程相
		符合
0x0321	从站未进入 OP 状态	2) 检查从设备通信状态是否正确
	SDO 参数未定义(SubIndex 超出	检查参数配置是否正确
0x0322	范围)	
0x0323	试图修改只读的 SDO 参数	检查参数是否被配置为RXPDO
	SDO 参数类型错误,数据长度不	检查参数配置是否被配置为RXPDO
0x0324	匹配	
		1) 检查通信连接线路,是否与工程相
		符合
0x0325	从站无效状态	2) 检查从设备通信状态是否正确
		1) 检查通信连接线路,是否与工程相
	初始化命令未执行(从站未挂载	符合
0x0326	在总线上)	2) 检查从设备通信状态是否正确
0x0327	不支持邮箱协议	检查从设备是否支持邮箱协议
0x0340	打开参数配置文件失败	重新编译下装工程文件
0x0341	轴未映射	在轴设置中,映射从站
0x0342	轴的数量超出范围	将轴的数量减少到最大支持的轴以内
		将轴组的数量减少到最大支持的轴组以
0x0343	轴组的数量超出范围	内
		将从站的数量减少到最大支持的从站数
0x0344	从站的数量超出范围	量以内
0x034A	参数 Index 超出范围	检查参数是否正确
0x034B	参数 SubIndex 超出范围	检查参数是否正确
0x034C	没有映射轴	在轴设置中,映射从站
0x034D	没有配置 TPDO 对象	添加需要的TPDO对象
0x034E	没有配置 RPDO 对象	添加需要的RPDO对象
	参数配置文件与参数模块不兼	升级MultiProg 软件或控制器固件
0x034F	容	
0x0352	掉电保持参数保存失败	联系供应商

0x0353	掉电保持参数加载失败	联系供应商
	从内核获取掉电保持参数地址	联系供应商
0x0354	失败	
0x0370	试图修改只读参数	检查该参数是否被配置成RxPDO
0x0371	获取参数值错误	联系供应商
0x0372	设置参数值错误	联系供应商
0x0380	保存参数到 FRAM 错误	联系供应商
0x0381	从 FRAM 获取参数错误	联系供应商
0x0391	读取主站 IO 错误	联系供应商
0x0392	写主站 IO 错误	联系供应商
0x03A0	轴没有定义	在轴设置中添加拟使用的轴
0x03A1	轴组没有定义	在轴组设置中添加拟使用的轴组
0x03A2	从站没有定义	在轴设置中选择正确的从站
		1) 检查是否在PDO Mapping 中配置了
		该对象,并进行映射
0x03A5	参数 没有定义	2) 检查该从站是否支持该参数
0x03A6	参数 只读	检查该参数是否被配置成RxPD0
0x03A7	获取参数值错误	联系供应商
0x03A8	设置参数值错误	联系供应商
0x03A9	参数数据类型错误	联系供应商
0x03AA	从站没有映射到轴	在轴设置中选择正确的从站

# 15.2.3. MODBUS 相关

代码	描述	处理方法
	(描述发生了什么异常)	(描述异常产生的原因,以及如何通过
		操作解决这个异常)
		1) 重启控制器
0x0701	内存分配失败	2)请与供应商联系
0x0702	Modbus Map 写入失败	请与供应商联系
0x0703	Modbus Map 初始化错误	请与供应商联系
0x0704	Modbus 读取失败线圈错误	检查Modbus 通信参数是否正确
0x0705	Modbus 读取输入线圈错误	检查Modbus 通信参数是否正确
0x0706	Modbus 读取寄存器错误	检查Modbus 通信参数是否正确
0x0707	Modbus 读取驶入寄存器错误	检查Modbus 通信参数是否正确
0x0708	Modbus 写单个寄存器错误	检查Modbus 通信参数是否正确
0x0709	Modbus 写多个个寄存器错误	检查Modbus 通信参数是否正确
0x070A	Modbus 写单个线圈错误	检查Modbus 通信参数是否正确
0x070B	Modbus 写多个线圈错误	检查Modbus 通信参数是否正确
0x070C	Modbus 读写多个寄存器错误	检查Modbus 通信参数是否正确

0x0801	内存分配失败	请与供应商联系
0x0808	Modbus 初始化失败	请与供应商联系
0x0809	Modbus 请求记录错误	请与供应商联系
0x080A	委托函数执行错误	请与供应商联系
0x080C	试图调用未初始化的委托函数	请与供应商联系

# 15.2.4.参数文件解析相关

代码	描述	处理方法
	(描述发生了什么异常)	(描述异常产生的原因,以及如何通过
		操作解决这个异常)
		2) 重启控制器
0x0401	内存分配失败	3) 请与供应商联系
		1) 重新编译下装程序
0x0402	参数文件读取失败	2) 请与供应商联系
		1) 重新编译下装程序
0x0403	参数文件写入失败	2) 请与供应商联系
		1) 重新编译下装程序
0x0404	参数数据类型错误	2) 请与供应商联系

# 15.2.5.PLC 内核相关

代码	描述	处理方法
	(描述发生了什么异常)	(描述异常产生的原因,以及如何通过
		操作解决这个异常)
0x0101	内存动态分配失败	系统没有足够大的连续空闲内存。重启
		i5后若仍然出现,则请与供应商联系。
0x0102	从工程中解析参数配置文件失败	可能是工程下载出错或者内部存储异
		常;请重新下载工程再试,若仍然出
		现,则请与供应商联系。
0x0103	PD0没有配置	工程中没有配置PDO并映射PDO到PLC变
		量,请通过MULTIPROG配置并映射PDO。
0x0104	从工程文件中解析参数配置文件	内部存储异常,请重新下载工程再试,
	成功,但保存文件失败	若仍然出现,则请与供应商联系。
0x0105	EtherCAT周期内任务处理时间超	可能是PLC工程中的处理逻辑太复杂,
	过最大允许值	超过i5的处理能力;请用一个简单逻辑
		的PLC工程下装运行尝试,若仍然出现
		该错误,请与供应商联系;若不在出
		现,则需要简化PLC的工程逻辑,降低
		复杂性;

0x0106	EtherCAT实际周期超过了设定周	系统错误,请与供应商联系。
	期允许的偏差的最大值	
0x0107	EtherCAT实际周期小于了设定周	系统错误,请与供应商联系。
	期允许的偏差的最小值	
0x010A	当前下装的工程中设置的DC的模	重新上电启动再试,若仍然出现该错误
	式与i5内核当前的DC模式不一致	代码,请与供应商联系
0x010B	找不到其它模块的接口函数的入	模块没有全部正确加载,请升级i5再
	П	试; 若问题持续出现, 请与供应商联系
0x010C	加载系统配置文件错误	可能是内部存储异常,请上电重启再
		试,若仍有问题,请与供应商联系
0x010D	PLC内核初始化错误	请与供应商联系
0x010E	PLC内核初始化错误	请与供应商联系
0x010F	PLC内核初始化错误	请与供应商联系
0x0110	PLC内核初始化错误	请与供应商联系
0x0111	PLC内核初始化错误	请与供应商联系
0x0112	PLC内核初始化错误	请与供应商联系
0x0113	PLC内核初始化错误	请与供应商联系
0x0114	PLC内核初始化错误	请与供应商联系
0x0115	PLC内核初始化错误	请与供应商联系
0x0116	PLC内核初始化错误	请与供应商联系
0x0117	PLC工程中PDO没有配置	请通过MultiProg配置从站PDO,并映射
		为变量
0x0118	EtherCAT周期设置改变,但没有	重新上电后新设置的周期才能生效
	生效, PLC无法启动	
0x0119	EtherCAT周期设置为了控制器不	请将EtherCAT周期设置为控制器支持的
	支持的值	范围,并重新上电
0x011A	EtherCAT的DC模式设置为了控制	请将EtherCAT的DC模式设置为控制器支
	器不支持的模式	持的模式,并重新上电
0x011B	PLC内核错误	请与供应商联系
0x011C	EtherCAT协议栈错误	请与供应商联系
0x011D	EtherCAT总线上的从站和	请检查MultiProg中配置的从站是否与
	MultiProg中配置的从站不一致	总线上连接的从站数量和型号是一致
		的;或者从站扫描从站信息;
0x011F	【致命错误】RPDO配置的数据对	检查RPD0中配置的数据对象的地址是否
	象地址和偏移量超过了缓冲区范	有效
	围	
0x0120	【致命错误】TPDO配置的数据对	减少TPD0中配置的数据对象的地址是否
	象地址和偏移量超过了缓冲区范	有效
	围	
0x0121	【致命错误】掉电保存信息版本	本次启动加载的掉电保存数据版本与固
	不匹配	件版本不匹配,本次掉电保存数据失
		效,请确认当前掉电保存数据是否正
		确,从新的一次掉电开始这些数据将重

		新得到保存
0x0122	EtherCAT连接中断	检查EtherCAT连接电缆是否可靠
0x0124	日志信息: DC模式被设置为	确认是否是当前的所需的DC模式
0x0127	日志信息: PDO配置数据大小	用于确认配置的PD0数据大小
0x0128	EtherCAT初始化过程中对从站初	检查从站型号是否在控制器测试适配的
	始化命令收到错误的回应	范围,若没有请与供应商联系
0x012C	SD0数据对象操作错误	根据日志附加信息确定错误的数据对象
		的索引和子索引
0x012D	至少有一个从站在错误状态	检查从站是否处于报警状态,若从站处
		于报警状态,对从站重新上电
0x012E	从站错误	具体错误信息见错误日志附加信息
0x0130	PLC启动过程等待从站进入0p状态	检查从站是否处于报警状态,若从站处
	超时	于报警状态,对从站重新上电
0x0131	EtherCAT周期性过程通讯计数器	检查主站和从站连接是否可靠
	错误	
0x0132	【致命错误】PLC运行因错误停止	检查日志中的最近的错误信息确定具体
	无法启动	原因
0x0133	主站发送给从站的数值帧没有获	检查主站和从站连接是否可靠,以及通
	得从站回应	讯周期和DC模式设置是否在允许范围内
0x01F0	错误详细描述1	用于显示错误附加信息,通常显示其前
		面最近的一条错误信息的附加说明第1
		部分
0x01F1	错误详细描述2	用于显示错误附加信息,通常显示其前
		面最近的一条错误信息的附加说明第2
		部分

# 15.2.6.运动控制相关

代码	描述 (描述发生了什么异常)	处理方法 (描述异常产生的原因,以及如何通过 操作解决这个异常)
0x1200	读写配置参数或COE对象出错	参数、对象未在MultiProg中配置,映 射,或类型错误。 可通过MultiProg查看日志显示的详细错 误提示信息,或编译信息,再补充相应 参数。
0x1211	PDS状态转换出错	伺服使能过程中状态转换出错 重新使能功能块MC_Power,如果错误保 持,请与供应商联系。
0x1212	PDS不支持HM工作模式	伺服不支持HM(Homing mode)工作模式 不能使用MC_Home功能块,请与供应商联

		系。	
0x1213	PDS不支持csp工作模式	伺服不支持Cyclic位置控制模式(csp	
		mode)工作模式	
		不能使用绝对运动、相对运动功能块,	
		请与供应商联系。	
0x1214	PDS不支持csv工作模式	伺服不支持Cyclic速度控制模式(csv	
		mode)工作模式	
		不能使用速度运动功能块,请与供应商	
		联系。	
0x121B	未配置的COE对象	控制器接收到不支持的COE对象	
		通过MultiProg配置该对象,或请与供应	
		商联系。	
0x121C	伺服的错误码对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		603F对象到TxPD0,或未映射到全局变量	
0x121D	COE控制字对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6040对象到RxPD0,或未映射到全局变量	
0x121E	COE状态字对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6041对象到TxPD0,或未映射到全局变量	
0x121F	COE工作模式对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6041对象到RxPD0,或未映射到全局变量	
0x1220	COE显示工作模式对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6061对象到TxPD0,或未映射到全局变量	
0x1221	COE实际位置对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6064对象到TxPD0,或未映射到全局变量	
0x1222	COE跟随误差窗口对象未配置	该从站对象字典不支持6065对象。	
0x1223	COE跟随误差超时对象未配置	该从站对象字典不支持6066对象。	
0x1224	COE到位检查范围对象未配置	该从站对象字典不支持6067对象。	
0x1225	COE到位检查时间对象未配置	该从站对象字典不支持6068对象。	
0x1226	COE实际速度对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		606C对象到TxPD0,或未映射到全局变量	
0x1227	COE目标位置对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		607A对象到RxPDO,或未映射到全局量。	
0x1228	COE工作行程对象未配置	该从站对象字典不支持607B对象。	
0x1229	COE软限位对象未配置	该从站对象字典不支持607D对象。	
0x122A	COE原点偏移量对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		607C对象到RxPD0,或未映射到全局变	
		量。	
0x122C	COE pp模式最大速度对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		607F对象到RxPD0,或未映射到全局量。	
0x122D	COE最大电机速度对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		6080对象到RxPD0,或未映射到全局变	
		量。	
0x1231	COE回零模式对象未配置	该从站对象字典不支持6098对象。	
0x1232	COE回零速度对象未配置	该从站对象字典不支持6099对象。	
--------	-------------------	-----------------------------	--
0x1233	COE回零加速度对象未配置	该从站对象字典不支持609A对象。	
0x1234	COE位置偏移量对象未配置	该从站对象字典不支持60B0对象。	
0x1235	COE速度偏移量对象未配置	该从站对象字典不支持60B1对象。	
0x1236	COE扭矩偏移量对象未配置	该从站对象字典不支持60B2对象。	
0x1237	COE锁存功能对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		60B8对象到RxPD0,或未映射到全局变	
		量。	
0x1238	COE锁存状态对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		60B9对象到RxPDO,或未映射到全局变	
		皇。	
0x1239	COE锁存器1上升沿锁存值对象未配	未在MultiProg中配置EtherCAT伺服从站	
	置	60BA对象到TxPDO,或未映射到全局变	
		星。	
0x123A	COE锁存器1下降沿锁存值对象未配	未在MultiProg中配置EtherCAT伺服从站	
	置	60BB对象到TxPD0,或未映射到全局变	
		量。	
0x123B	COE锁存器2上升沿锁存值对象未配	未在MultiProg中配置EtherCAT伺服从站	
	置	60BC对象到TxPD0,或未映射到全局变	
		量。	
0x123C	COE锁存器2下降沿锁存值对象未配	未在MultiProg中配置EtherCAT伺服从站	
	置	60BD对象到TxPD0,或未映射到全局变	
		量。	
0x123D	COE最大升速加速度对象未配置	该从站对象字典不支持60C5对象。	
0x123E	COE最大降速加速度对象未配置	该从站对象字典不支持60C6对象。	
0x123F	COE数字IO对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		60FD对象到TxPD0,或未映射到全局变	
0x1240	COE目标速度对象未配置	未在MultiProg中配置EtherCAT伺服从站	
		60FF对象到RxPD0,或未映射到全局变	
		重。	
0x1242	状态字对象没有配置为PD0	未在MultiProg中配置EtherCAT伺服从站	
		6041对象到TxPD0,或禾映射到全局变	
0x1243	错误码对家木配直	木住MultiProg中配置EtherCAT何服从站	
		603F对象到TxPD0, 或木映射到全局受	
0.1044			
0x1244	错误码对家没有配直为PD0	本在MultiProg中配直EtherCAT何服从站	
		003F	
0 1045	<u> </u>		
UX1245	数子制八NI家沒有能直为PDU	不住MultiProg中能直出therUAI何服从站	
		UUFDAY家到IXFDU,以不哄射到生同受	
01040	· 一定位里对色机去和巴头ppo	里。	
UX124b	头孙'但直对家汉有能直为PDU	不住MultiProg中能直LtherUAI 何服从站	

		6064对象到TxPD0,或未映射到全局变
		量。
0x1247	该从站不支持MC_TouchProbe功能	该从站不支持MC_TouchProbe功能
0x1248	该从站不支持MC_AbortTrigger功	该从站不支持MC_AbortTrigger功能
	能	
0x1249	该从站不支持此对象	该从站对象字典不支持此对象
0x124A	伺服未使能	伺服未使能。
		检查伺服是否未使能或出错。
0x124B	最大输出转矩对象未配置	未在MultiProg中配置EtherCAT伺服从站
		6072对象到RxPD0,或未映射到全局量。
0x124D	编码器命令对象为未配置为PDO	未在MultiProg中配置EtherCAT伺服从站
		642B对象到TxPD0,或未映射到全局变
		量。
0x124E	编码器状态对象未配置为PD0	未在MultiProg中配置EtherCAT伺服从站
		6511对象到TxPD0,或未映射到全局变
		量。
0x124F	编码器当前位置对象未配置为PDO	未在MultiProg中配置EtherCAT伺服从站
		6020对象到TxPD0,或未映射到全局变
		量。 ————————————————————————————————————
0x1250	编码器预设位置对象未配置为PD0	未在MultiProg中配置EtherCAT伺服从站
		6010对象到TxPD0,或未映射到全局变
		量。
0x1251	编码器位置比较输出映射对象未配	未在MultiProg中配置EtherCAT伺服从站
	置为PD0	633A对象到TxPD0,或未映射到全局变
0x1252	编码器配置预置位置比较输出点对	未在MultiProg中配置EtherCAT伺服从站
	象禾配置为PD0	642D对象到TxPD0,或未映射到全局变
0x1300	创建RIL节点缓冲失败	控制器上电初始化分配内存失败。
		控制器重新上电,尤法消除请与供应简 1995
0.1001		联系。
0x1301	创建RIL轴缓冲失败	控制器上电初始化分配内存失败。
		控制器里新上电,尤法消除请与供应商
0.1000		联系。
0x1303	输出COE参数超时	将切能块操作的COE参数输出到伺服从站
		位值MultiProg中能直的何脉定省与连按
		的何 <u>你</u> 匹配, <u></u>
01904		
UX13U4	仅有能直的[1003]家	▲ 和 multiProg 中 能 直 的 PDU 劝 家。 相 捉 MultiProg 武 日 士 信 自 · 确 会 土 町 平
		□1K1/hmult1F10g以口芯信芯,佣化不能直的余粉ID 在MultiPro~中配罢岀DD0余
		hij gain, filmuitii tog T 印 且/风 DU 参 粉
0v1205	未配置的参数	メ ま在MultiProg由配置的对象
071000	小山且印罗双	/NFLmuitining「日山目山小」刻。

		根据MultiProg或日志信息,确定未配置
		的参数ID,在MultiProg中配置。
0x1307	从站没有连接	内核反馈没有找到指定的伺服从站。
		检查EtherCAT连接,重新上电控制器和
		从站。
0x1308	从站号没有配置	内核反馈没有伺服从站号错误。
		检查EtherCAT连接,重新上电控制器和
		从站。
0x1309	从站未知错误	内核反馈没有伺服从站号错误。
		检查EtherCAT连接,重新上电控制器和
		从站。
0x130A	轴没有与从站映射	轴设置中轴号没有与伺服从站完成映
		射。
		在轴设置中将伺服从站配置到指定的轴
		号。
0x130D	超出位置范围	MC_SetPosition中设置的位置超出范
		围。
		修改MC_SetPositon中的位置参数。
0x130E	读取RTL层命令节点缓冲失败	指令在控制器内部读缓冲失败。
		系统内部错误,请与供应商联系。
0x1400	单轴功能块操作轴组轴	MC_Stop或MC_ImmediateStop操作的轴属
		于轴组里的一根轴。
		要停止轴组轴的运动采用MC_GrpStop或
		$MC\_GrpImmediateStop_{\circ}$
0x1401	创建BLL节点缓冲失败	控制器上电初始化分配内存失败。
		控制器重新上电,无法消除请与供应商
		联系。
0x1402	创建BLL轴缓冲失败	控制器上电初始化分配内存失败。
		控制器重新上电,无法消除请与供应商
		联系。
0x1403	伺服未使能	伺服未使能。
		检查伺服是否未使能或出错。
0x1404	单轴插补已结束,剩余距离为0	在动态改变单轴目标位置时单轴运动已
		经结束又改变目标位置报1404错误。
0x1405	单轴改变目标位置,轴处于降速过	在动态改变单轴目标位置时,轴处于降
	程中反向运动前这个阶段不能再次	速过程中反向运动前这个阶段不能再次
	改变目标位置	改变目标位置。
		可以在降速结束后反向运动时再次修改
		目标位置。
0x1406	创建BLL轴组缓冲失败	控制器上电初始化分配内存失败。
		控制器重新上电,无法消除请与供应商
		联系。
0x1480	轴组插补已结束,剩余距离为0	在动态改变轴组目标位置时轴组运动已
		经结束又改变目标位置报1480错误。

0x1481	轴组改变目标位置,轴组处于降速	在动态改变轴组目标位置时,轴组处于
	过程中反向运动前这个阶段不能再	降速过程中反向运动前这个阶段不能再
	次改变目标位置	次改变目标位置。
		可以在降速结束后反向运动时再次修改
		目标位置。
0x1490	辅助点为圆心,无法构成圆弧	修改圆心坐标,使圆心到起点与终点的
		距离相等。
0x1491	辅助点为通过点,无法构成圆弧	修改通过点坐标,使其三点不在同一条
		直线上
0x1492	辅助点为半径,半径=0或太小	修改半径长度,使半径*2大于等于起点
		到终点的距离
0x1493	辅助点为半径,起点终点重合	不支持起点终点重合的半径模式圆弧,
		若要绘制整圆,采用其他两种模式。
0x1500	多个MC_Power功能块操作了相同轴	每个轴只能使用一个MC_Power功能块进
		行使能或禁用。
		在MultiProg中删除多余的MC_Power功能
		块或修改轴号。
0x1501	功能块主参数出错	PLC执行过程中执行过程中发生变化,轴
		号发生了变化。
		检查PLC程序,功能块在执行过程中是否
		修改了轴号参数。
0x1502	功能块序号出错	功能块序号超出有限范围。
		系统内部错误,请与供应商联系。
0x1504	写运行缓冲失败	指令在控制器内部写缓冲失败。
		系统内部错误,请与供应商联系。
0x1505	写进给缓冲失败	指令在控制器内部写缓冲失败。
		系统内部错误,请与供应商联系。
0x1506	写通信缓冲失败	指令在控制器内部写缓冲失败。
		系统内部错误,请与供应商联系。
0x1507	写轴组缓冲失败	指令在控制器内部写缓冲失败。
		系统内部错误,请与供应商联系。
0x1510	何服未使能	何服还未进入使能状态。
		调用MC_Power,使能伺服轴。
0x1512	伺服回零错误	何服电机回零过程中出现错误。
		何服重新上电。如果个能消除,请与供
0 1510	与坦同意供当	应同联系。 与四点地 只要过现点 业现件 况
0x1513	19版四令钳误	何版电机回零过程甲出现错误。
0 1 5 1 5	与职同委件识	四间状余。 与职力相同委员和古典报告。
Ux1515	19版四令钳误	
		[
0 1500		应用状系。 自动声声运动,在图示力量,业产处约
0x1520	拙运动有,仲关,个能后动运动	后列速度运动、位直运动时,当則的轴

		杏看指今手册。了解功能快的启动冬
		一旦有话《了加,了研究能获的加切东 件,修改PLC程序。
0x1521	周期同步运动的缓冲模式参数有误	启动周期同步运动功能块时,功能块的
UNICEI		<i>经冲参数设置有误。不能为笔待方式</i>
		极计多级设置自然, <b>不能力</b> 导的方式。 修改周期同生运动功能拉的浮油会粉为
		应因为为内少运动为能头的级科参数/ <b>为</b> 由艇主式
0 1500		
0x1522	后功MC_AbortIrigger天败	后初MU_AbortIr1gger时,操作的轴已处
		于停止领存状态中,即已启动于停止领
		仔切能,还木执行结束。
		检查PLC程序是否有多个停止锁存功能块
		同时启动执行。
0x1523	轴运动有冲突,不能启动回零	启动回零运动时,当前的轴状态不允许
		启动运动。
		查看指令手册,了解功能块的启动条
		件,修改PLC程序。
0x1524	复位启动失败	启动MC_Reset功能块时,轴处于运动过
		程中。
		修改PLC程序,等待运动停止后启动复位
		操作。
0x1525	设置位置失败	启动MC_SetPosition功能块时,当前的
		轴状态不允许功能块执行。
		查看指令手册,了解MC SetPosition功
0x1526	功能块轴号参数设置错误	扫描功能块时,未找到配置的轴号。
		检查轴设置中已配置了哪些轴,修改PLC
		程序轴号参数。
0x1527	控制器内部错误	控制器内部错误。
		请与供应商联系。
0x1528	创建工作节点失败	控制器系统初始化时分配内存失败。
		请与供应商联系。
0x1529	创建轴失败	控制器系统初始化时分配内存失败。
		请与供应商联系。
0x152A	创建轴组失败	控制器系统初始化时分配内存失败。
		请与供应商联系。
0x152B	功能块数量超过允许最大值	控制器最多允许PLC程序中使用5000个功
		能块.功能块数超个5000将报错。
		删除PLC程序中的多余功能块。
0x152F	轴组中轴的数量超过4	轴组功能快最大轴数为4. 超过该值将报
UNITED .		田子····································
		<sup>113</sup>   修改轴组功能快中的轴粉参数
0v159F	MC TouchProbaTh能也会粉供得	1943年初にハーリオースジス。 MC TouchProbeTh能性TrigonInput 任坊
UAT J2P	mo_10ucm100e切配ر多数相伏	mc_rouchi rouc为形式firger input 如构 休的LouolTupo会粉铀生信旦由亚识里雄
		PAULEVEIType多效概况后与电干仅且指
		修以PLU程序甲切能块参数。

0x1531	MC_MoveVelocity启动失败	启动MC_MoveVelocity功能块时,轴处于
		错误状态。
		查看指令手册, 了解功能块的启动条
		件,修改PLC程序。
0x1532	MC_Home启动失败	启动MC_Home功能块时,轴处于错误状
		态。
		查看指令手册, 了解功能块的启动条
		件,修改PLC程序。
0x1533	MC_Stop启动失败	启动MC_Stop功能块时,轴处于错误状
		念。
		查看指令手册, 了解功能块的启动条
		件,修改PLC程序。
0x1534	MC_GrpStop启动失败	启动MC_GrpStop功能块时,轴处于错误
		状态。
		查看指令手册, 了解功能块的启动条
		件,修改PLC程序。
0x1535	MC_GrpImmediateStop启动失败	启动MC_GrpImmediateStop功能块时,轴
		处于错误状态。
		查看指令手册,了解功能块的启动条
		件,修改PLC程序。
0x1536	轴组绝对运动或相对运动启动失败	启动轴组绝对运动或相对运动功能块
		时,轴处于错误状态。
		查看指令手册,了解轴组绝对运动或相
		对运动功能块的启动条件,修改PLC程
		序。
0x1537	MC_GrpSyncMoveAbs启动失败	启动MC_GrpSyncMoveAbs功能块时,轴处
		于错误状态。
		查看指令手册, 了解功能块的启动条
		件,修改PLC程序。
0x1538	MC_ImmediateStop启动失败	启动MC_ImmediateStop功能块时,轴处
		于错误状态。
		查看指令手册,了解功能块的启动条
		件,修改PLC程序。
0x1539	MC_WriteParameter启动失败	启动MC_WriteParameter功能块时,轴处
		于错误状态。
		查看指令手册,了解功能块的启动条
		件,修改PLC程序。
0x153A	创建或取消轴组失败	启动MC_GrpEnable、MC_GrpDisable功能
		块时,轴处于错误状态。
		查看指令手册,了解功能块的启动条
		件,修改PLC程序。
0x153B	MC_GrpReset启动失败	启动MC_GrpReset功能块时,轴处于错误
		状态。
		查看指令手册,了解功能块的启动条

		件,修改PLC程序。
0x153C	单轴运动的轴已被配置到轴组中	如果轴已被配置的轴组中,不能单独启
		动。
		修改功能块轴号参数。
0x153D	轴组之间有冲突	轴组中的某轴已被配置到其它轴组中。
		修改功能块轴号参数。
0x153E	启动轴组功能块时,未使能该轴组	轴组功能块启动前,必须先调用
		MC_GrpEnable使能轴组。
		修改PLC程序,补充MC_GrpEnable功能
		块。
0x1542	节点号超限	控制器系统内部错误。
		请与供应商联系。
0x1543	轴缓冲满	单轴运动功能块缓冲已存储了一条指令
		(满),不能存储新增的指令。
0x1544	轴组缓冲满	轴组运动功能块缓冲已存储了50条指令
		(满),不能存储新增的指令。
0x1545	轴组缓冲读错误	未能从轴组缓冲中取出指令。
		系统内部错误,请与供应商联系。
0x1547	伺服中6080 COE参数值为0	伺服驱动器中6080存储的是最大电机速
		度参数,伺服在hm(回零模式)、csv模
		式下,该参数必须大于0。
		在轴设置中配置该参数大于0,重新下载
		PLC程序。
0x1548	功能块轴组号参数设置错误	扫描轴组功能块时,未找到配置的轴组
		号。
		检查轴设置中已配置了哪些轴组,修改
		PLC程序轴组号参数。
0x1549	运动过程中轴组变成未使能状态	轴组运动中,调用了MC_GrpDisable功能
		块。
		轴组运动中,不能调用MC_GrpDisable功
		能块。
0x154A	功能块参数错误	功能块参数错误。
		查看指令手册,了解功能块输入参数范
		围,修改PLC程序。
0x154B	线段长度超限错误	输出到伺服的线段长度超过最大跟随误
		差。
		(1)检查轴设置中跟随误差是否过小;
		(2)系统内部出错,请与供应商联系。
0x154C	控制器内部创建缓冲失败	控制器上电初始化创建共享内存缓冲失
		败。
		控制器重新上电,无法消除请与供应商
		联系。
0x154D	1 轴组绝对运动或相对运动启动失败	当前正在执行轴组的周期同步运动,而
		新启动的运动与上一条指令间的衔接方

		式为缓冲模式,则新指令无法执行。
		修改轴组运动的衔接方式为中断模式。
0x154E	验证轴组数为0	未配置轴组,如果PLC程序中使用了轴组
		功能块,将出现此错误。
		配置拟使用的轴组。
0x154F	验证轴组号出错	使用了未配置的轴组号。
		(1) PLC程序中使用配置的轴组号:
		(2) 配置拟使用的轴组。
0x1550	轴类型错误,不支持的轴类型	轴类型选择错误,该版本只支持伺服轴
		和虚拟伺服轴。
		轴类型选择伺服轴或虚拟伺服轴。
0x1552	轴未使能为轴组	启动轴组运动前未使能轴组。
		在启动轴组运动前先启动MC_GrpEnable
		使能轴组。
0x1553	周期运动指令与其他运动冲突	轴在运动过程中不能使用周期运动指
		令。
0x1554	该从站对应的轴不支持此功能 该从站对应的轴不支持此功能	
0x1555	系统还未准备好,无法启动功能块 请与供应商联系。	
0x1556	功能块超时,目前用于功能块速度	请查看日志并记录日志内容与供应商联
	为0的超时报警	系
0x1557	Power off失败,伺服未进入OFF状	(1) 检查EtherCAT网络连接; (2) 请
	态	与供应商联系。
0x1558	Power on失败,伺服未进入ON状态	(1)检查EtherCAT网络连接; (2)请
		与供应商联系。
0x1564	电机一周工作行程为0错误	轴设置中电机一周工作行程不能为0。
		修改该参数大于0。
0x1565	PLC程序启动时设置伺服参数出错	MultiProg下载PLC程序或控制器启动PLC
		程序时,控制器会配置的伺服参数写入
		从站,如果写入出错将输出此错误。
		(1)检查EtherCAT网络连接; (2)请
		与供应商联系。
0x1567	PLC程序启动时读配置文件参数出	MultiProg下载PLC程序或控制器启动PLC
	错	程序时,控制器会将读取配置参数,如
		果读取出错将输出此错误。
		请与供应商联系。
0x1570	创建工作节点缓冲失败	首次扫描功能块创建工作节点时分配内
$\sim$		存失败。
0x157C		系统内部错误,请与供应商联系。
0x1580	复位伺服故障失败	MC_Reset复位伺服异常失败。
		检查伺服的异常号是否支持软件复位。
0x1581	使用了不支持的参数ID号	MC_WriteParameter, MC_ReadParameter
		中参数ID号错误。
		查看指令手册,了解控制器支持的ID
		号。

0x1582	只读参数不支持写操作	MC_WriteParameter写了只读参数。
		修改功能块参数号。
0x1590	虚拟轴无此功能	虚拟轴无MC_TouchProbe、
		MC_AbortTrigger功能。
0x1600	从站缺省错误	重启从站或对从站进行Reset操作。
0x1700	功能块输入参数结构体长度校验出	检查功能块对应的结构体长度或者数组
	错	长度是否正确。
0x1701	不支持的指令	不支持的指令。
0x1702	轴组参数错误	检查轴组号设置与PLC程序中是否一致。
0x1800	多个LMC_EnableHSCounter功能块	每个轴只能使用一个
	操作了相同 <u>ch</u>	LMC_EnableHSCounter功能块进行使能或
		禁用。
		在MultiProg中删除多余的
		LMC_EnableHSCounter功能块或修改轴
		号。
0x1801	高速计数器未使能	高速计数器还未进入使能状态。
		调用LMC_EnableHSCounter, 使能高速计
		数器。
0x1802	高速计数器功能关闭失败,高速计	请与供应商联系。
	数器未进入0FF状态	
0x1803	高速计数器 <u>ch</u> 错误	使用了未配置的高速计数器ch号。
		(1) PLC程序中使用配置的高速计数器
		CH号; (2) 配置拟使用的高速计数器。

## 15.3. 附录 3: M3 变量地址表

地址	功能及说明	数据类型
$0^{\sim}1999$	具有掉电保持功能的Modbus 可访问的变量	CHAR8
	将变量地址映射到该区间,则该变量具有掉电保持属	
	性,无需勾选保持属性。	
$2000^{\sim}19999$	不具有掉电保持功能的Modbus 可访问变量	CHAR8

$\triangle$	注意	<ol> <li>分配到 M3 地址的变量不能在编程界面勾选保持属性,否则在编译时将报错。</li> <li>PLC 工程在冷启后,具有保持属性的数据将被重置为初始值</li> </ol>
		3, M3 保持区域的初始值,在冷启后均将被设置为 0.