

通过扩展脚本与 OMRON 温控器通讯 说明书

普洛菲斯国际贸易(上海)有限公司 技术热线: 021-6361-5008

目录

内容	一页 码
1. 概述	3
2. 画面建立	4
3. 地址和 D 脚本说明	7
4. GP、PLC 和画面编辑软件版本	20
5. 画面复制	20
6. 注意事项	22
<附录>	
建立和编辑 D 脚本	23
关于 LS 区	24
确认地址	25

注意: 在您的系统中使用本例时, 在操作前请检查。

1. 概述

通过下面的步骤,您可以将 OMRON 温控器与三菱 PLC 同时连接在 GP 上,GP 作 为终端站点显示信息。



通过扩展串口脚本协议,生成与温控器通讯的驱动。与 PLC 通讯方式一样,GP 发送请求命令,然后处理温控器返回的响应数据(参考图表 1)

在 GP 系列中,通讯脚本是和画面处理分开执行的(参考图表 2)



图表 2 扩展 SIO 脚本通讯略图

(注意) 本示例在 GP(GLC)2000 系列上运行。在触摸屏上设定数值时可能需 要时间。(→6. 注意事项)

2. 画面建立

<标准画面>



- B1001 温控器画面
- A:开始/停止按键(位开关) 按下开始键,和温控器开始通讯。 按下停止键,和温控器终止通讯。
- B: 通讯状态显示灯(灯部件) 每次通讯, 灯会变亮/灭。
- C: 显示温控器的数据(数据显示部件) 显示读取数据(当前值,设定值)。
- **D**: 上/下开关设置温控器数值(字开关) 可以在触摸屏上直接设置数据值。
- E: 窗口显示开关(位开关) 按下该开关,弹出一个窗口。
- F: 窗口显示(窗口显示部件) 调用窗口并显示。
- A: 位开关



C: 数值显示

————————————————————————————————————
一般设置 显示格式 形状/颜色 报警设置
描述 字地址 () [158012] [] [] [] [] [] [] [] [] [] [] [] [] [
浏览 (8)
确定 取消 帮助 (2)

B: 指示灯



C: 数值显示

數值显示器设置[ND]	_007]	X
一般设置 显示格式	形状/颜色 报警设置	
数据显示格式 ▼ 符号 +/- 16 位 32 位 ○ Dec ○ Rex ○ BCD ○ BC ○ Oct 显示风格	数字位数 4 ÷ 小数点位置 1 ÷ co CD 工業的大小 工業の 工業の 及第 工業の 及第 位数 4 ÷ 小数点位置 1 ÷	
● 石砂 月 ○ 左段 月	▼ 李州前 □ 委員士 888.8	
、 2110 ,	▲ 北面小	
រ រិង	确定 取消 帮助(H)	

D: 字开关

E: 位开关

F: 窗口显示

窗口部件设置[WI_001]	
- 般设置 描述 「役置窗口画面号 窗口号 1001 →	窗口显示地址 窗口控制地址 ℃ LS7020 位0 ON/OFF 位1 显示顺序
1001 224x122 1002 320x81	
	取消 帮助(出)

3. 地址和D脚本说明

本示例运行需要使用扩展脚本,脚本说明如下。(→建立和编辑D脚本) LS区中用到的地址如下。(→关于LS区) (注意:通讯脚本中使用的地址只是LS 区地址)

使用的地址	详细说明
字地址	
LS7000	开始/停止
LS7001	通讯错误
LS7020	通讯错误窗口显示
LS7021	通讯命令窗口显示
LS7050	写入数据的中断标记
LS7100	BCC 操作的字节数
LS7101	BCC 操作结果存储地址
LS7110~LS7149	BCC 操作字符存储地址
LS7200	发送字节数
LS7210~LS7249	发送命令字符存储地址
LS7300	发送命令存储地址(S-tag 显示)
LS7400	接收字节数
LS7410~7449	接收命令字符存储地址
LS7500	接收命令存储地址(S-tag 显示)
LS8000~LS8002	接收数据存储地址
LS8012~LS8013	温控器当前值显示地址
LS8014~LS8015	温控器设置值显示地址
LS8100	温控器设置值写入地址
LS8110~LS8117	写入的数值经数据/字符转换后的存储地址

使用的地址	详细
位地址	
LS700000	开始位
LS700001	停止位
LS700004	通讯状态确认位
LS700100	发送错误(电缆没有连接)
LS700101	接收错误
LS700104	通讯超时
LS705000	数据写入中断位
临时地址	
t0030	发送完成标志
t0032	接收超时计数
t0033	接收数据偏移
t0035	BCC 操作循环次数
t0036	BCC 操作循环偏移
t0037	发送命令循环次数
t0038	发送命令循环偏移
t0039	发送命令循环偏移
t0040	接收字节数
t0045	接收命令循环次数
t0046	接收命令循环偏移
t0047	接收命令循环偏移

临时地址只能用于D脚本内部操作

扩展串口协议脚本的说明

// 结束

[c:EXT_SIO_CTRL00]=1	// 清除发送缓冲
[c:EXT_SIO_CTRL01]=1	// 清除接收缓冲
[c:EXT_SIO_CTRL02]=1	// 错误清除

// 此脚本用于发送/接收从温控器读到的当前值		
[t:0030]=0	// 发送完成标志清除	
Call F2_R_SV1_S	// 调用发送功能	
[t:0033]=4	// 设置接收地址偏移	
if([t:0030]==1){	// 命令是否发送	
Call F3_RCV	// 调用接收功能	
}endif		

//	// 此脚本用于发送/接收从温控器读到的设定值		
[t	:0030]=0	//	发送完成标志清除
С	all F2_R_SV1_S	//	调用发送功能
[t	:0033]=4	//	设置接收数据的地址偏移值
if	([t:0030]==1){	//	命令是否已发送
С	all F3_RCV	//	调用接收功能
}€	endif		

// 此脚本用于发送/接收写入到温控器的当前值	Ĺ
[t:0030]=0	// 发送完成标志清除
Call F2_W_SV1_S	// 调用发送功能
[t:0033]=0	// 设置接收数据地址偏移
if([t:0030]==1){	// 是否发送命令
Call F3_RCV	// 调用接收功能
}endif	
if(([w:LS7010]==0)and([w:LS7011]==0)){	// 温控器没有通讯错误时
clear([b:LS705000])	// 释放写入保留区
}endif	

脚本功能: F1_BCC

// 获取databuf0的数据长度
// 减少开始位
// 取出数据
// 循环偏移
// BCC操作初始化
// 操作时间循环
// XOR操作
// 偏移相加
// databuf1初始化
// 操作结果到databuf1

// 创建发送命令读温控器当前值	
loop(){	
if([s:EXT_SIO_STAT00]==1){	// 发送缓冲是否准备好
break	// 跳出循环
}endif	
}endloop	
11 牛虎追掠盟通过会会	
// 主风通招奋也以即文 	// databuff()初始化
_stroat(databulo,)	// Ualabuio/初知化 // 设罢会太正裕
_stroot(databulo, 0x02)	// 反直仰ぐ开知 // 沿罢畄元巳矶
	//
_strcat(databuf0, "000")	
_strcat(databuf0, "0101C00000000001")	// 连接到读命令
_strcat(databuf0, 0x03)	// 连接到末尾
Call F1_BCC	// 调用BCC操作功能
_strcat(databuf0, databuf1)	// 连接BCC
_strlen([w:LS7200], databuf0)	// 获得发送字节数
memset([w:LS7210],0,50)	// 发送命令初始化
_dlcopy([w:LS7210], databuf0, 0, [w:LS7200])	// 确认发送命令
Call Send_Byte2Word	// 接受命令分段显示(调试)
IO_WRITE_EX([p:EXT_SIO], databuf0, [w:LS7200])	// 输出到扩展串口
[t:0030]=1	// 发送完成标志为ON

// 创建发送命令读温控器设定值	
loop(){	
if([s:EXT_SIO_STAT00]==1){	// 发送缓冲是否准备好
break	// 跳出循环
}endif	
}endloop	
// 设置温控器的通讯命令	
_strset(databuf0, "")	// databuf0初始化
_strset(databuf0, 0x02)	// 设置命令开始
_strcat(databuf0, "01")	// 设置号码
_strcat(databuf0, "000")	
_strcat(databuf0, "0101C10003000001")	// 连接读命令
_strcat(databuf0, 0x03)	// 连接末尾
Call F1_BCC	// 调用BCC操作功能
_strcat(databuf0, databuf1)	// 连接BCC
_strlen([w:LS7200], databuf0)	// 获得发送字节数
memset([w:LS7210],0,50)	// 发送命令初始化
_dlcopy([w:LS7210], databuf0, 0, [w:LS7200])	// 确认发送命令
Call Send_Byte2Word	Ⅱ 接受命令分段显示(调试)
O_WRITE_EX([p:EXT_SIO], databuf0, [w:LS7200]]) // 输出到扩展串口
[t:0030]=1	// 发送完成标志为ON

// 创建发送命令写温控器的设定值 loop(){ if([s:EXT SIO STAT00]==1){ // 发送缓冲是否准备好 // 跳出循环 break }endif }endloop // 创建和温控器的通讯命令 _strset(databuf0, "") // databuf0初始化 _strset(databuf0, 0x02) // 设置开始命令 _strcat(databuf0, "01") // 设置号码 _strcat(databuf0, "000") // 设置0 strcat(databuf0, "0102C10003000001") // 连接写入命令 Call F2_W_Bin2ASC // 创建写入数据 _strset(databuf1, "") // databuf1初始化 ldcopy(databuf1, [w:LS8114], 4) // 在缓冲中存储写入数据 _strcat(databuf0, databuf1) // 连接数据 strset(databuf1, "") // databuf1初始化 // 在缓冲中存储写入数据 ldcopy(databuf1, [w:LS8110], 4) _strcat(databuf0, databuf1) // 连接数据 // 连接开始 _strcat(databuf0, 0x03) Call F1_BCC // BCC操作功能处理 _strcat(databuf0, databuf1) // 连接BCC strlen([w:LS7200], databuf0) // 获得发送字节数 // 发送命令初始化 memset([w:LS7210],0,50) _dlcopy([w:LS7210], databuf0, 0, [w:LS7200]) // 确认发送命令 Call Send Byte2Word // 接受命令分段显示(调试) IO_WRITE_EX([p:EXT_SIO], databuf0, [w:LS7200]) // 输出到扩展串口 [t:0030]=1 // 发送完成标志为ON

脚本函数: F2_W_Bin2ASC

// 此脚本用于 Bin 到 ASCII 字符转换	
//处理第 8 位 [w:LS8120]=([w:LS8100]&0xF000)>>12 if(([w:LS8120]>=0x00) and ([w:LS8120]<=0x09)){ [w:LS8110]=[w:LS8120]+0x30	// 位标记和位移动 // 数值举例 // 加十六进制 30
}endif if(([w:LS8120]>=0xA) and ([w:LS8120]<=0xF)){ [w:LS8110]=[w:LS8120]+0x37	// 字符举例 // 加十六进制 37
//处理第 7 位	
[w:LS8121]=([w:LS8100]&0x0F00)>>8 if(([w:LS8121]>=0x00) and ([w:LS8121]<=0x09)){ [w:LS8111]=[w:LS8121]+0x30	
if(([w:LS8121]>=0xA) and ([w:LS8121]<=0xF)){ [w:LS8111]=[w:LS8121]+0x37 }endif	
w:LS8122]=([w:LS8100]&0x00F0)>>4	
if(([w:LS8122]>=0x00) and ([w:LS8122]<=0x09)){ [w:LS8112]=[w:LS8122]+0x30 }endif	
if(([w:LS8122]>=0xA) and ([w:LS8122]<=0xF)){ [w:LS8112]=[w:LS8122]+0x37	
//处理第 5 位	
[w:LS8123]=([w:LS8100]&0x000F)	
if(([w:LS8123]>=0x00) and ([w:LS8123]<=0x09)){ [w:LS8113]=[w:LS8123]+0x30 }endif	
if(([w:LS8123]>=0xA) and ([w:LS8123]<=0xF)){ [w:LS8113]=[w:LS8123]+0x37	
}enair	

脚本函数: F2_W_Bin2ASC(继续)

//处理第 4 位	
[w:LS8124]=([w:LS8101]&0xF000)>>12	// 位标记和位移动
if(([w:LS8124]>=0x00) and ([w:LS8124]<=0x09)){	// 数值举例
[w:LS8114]=[w:LS8124]+0x30	// 加十六进制30
}endif	
if(([w:LS8124]>=0xA) and ([w:LS8124]<=0xF)){	// 字符举例
[w:LS8114]=[w:LS8124]+0x37	// 加十六进制37
}endif	
//处理第 3 位	
[w:LS8125]=([w:LS8101]&0x0F00)>>8	
if(([w:LS8125]>=0x00) and ([w:LS8125]<=0x09)){	
[w:LS8115]=[w:LS8125]+0x30	
}endif	
if(([w:LS8125]>=0xA) and ([w:LS8125]<=0xF)){	
[w:LS8115]=[w:LS8125]+0x37	
11 处理第 2 位	
[w:LS8126]=([w:LS8101]&0x00F0)>>4	
$If(([W:LS8126] >= 0X00) and ([W:LS8126] <= 0X09)){$	
[W.L30110]-[W.L30120]+0X30 lendif	
$if(([w:1 S8126]>=0x\Delta) and ([w:1 S8126]<=0xE))$	
[w: 1 S8116] = [w: 1 S8126] + 0x37	
endif	
//处理第1位	
[w:LS8127]=([w:LS8101]&0x000F)	
if(([w:LS8127]>=0x00) and ([w:LS8127]<=0x09)){	
[w:LS8117]=[w:LS8127]+0x30	
}endif	
if(([w:LS8127]>=0xA) and ([w:LS8127]<=0xF)){	
[w:LS8117]=[w:LS8127]+0x37	
}endif	

脚本函数: Send_Byte2Word

// 逐步发送字节数据到字地址 LS7300	
[t:0037]=20 // 设定循环次数 [t:0038]=0 // 初始化 [t:0039]=0 // 初始化	
loop([t:0037]){	
[w:LS7350]#[t:0039]=[w:LS7210]#[t:0038]<	<8 // 偶字节上移
[t:0038]=[t:0038]+1	// 增加偏移值
[w:LS7300]#[t:0039]=[w:LS7350]#[t:0039] [v	w:LS7210]#[t:0038]
	// 高低位加
[t:0038]=[t:0038]+1	// 增加偏移值
[t:0039]=[t:0039]+1	// 增加偏移值
}endloop	

// 此脚本用于反向接收发送命令	
[t:0032]=10000	// 接收超时时间设定(100ms×100)
	, , , , , , , , , , , , , , , , , , ,
if([s:EXT_SIO_STAT02]==1){	// 接收数据是否存在
if([s:EXT_SIO_STAT03]==1){	// 接收错误是否存在
[c:EXT_SIO_CTRL01]=1	// 接收缓冲清除
return	// 处理过程结束
}else{	
_wait(3)	// 等待接数据
_strset(databuf2, "")	// databuf2初始化
[t:0040]=[r:EXT_SIO_RECV]	
IO_READ_EX([p:EXT_SIO], dates the second sec	atabuf2, [t:0040]) // 获得接收数据
_strlen([w:LS7400], databuf2)	// 接收字节数
memset([w:LS7410],0,20)	// 接收地址区初始化
_dlcopy([w:LS7410], databuf2,	0, [w:LS7400]) // 确认接收命令
Call RCV_Byte2Word	// 调试
// 温控器结束代码处理	
_strset(databut3, ^m)	// Catabut3 初始化
_stimid(databul3, databul2, 5,	Z) // 从数据缓冲取出数据 buf2) // 结束供和
	buis) // 细來代码
パ 血圧曲 や広 1 (時文)生 strset(databuf3 "")	// databuf3初始化
strmid(databuf3, databuf2, 11	.4) // 从数据缓冲取出数据
hexasc2bin([w:LS7011], data	buf3)
// 温控器数据处理	,
_strset(databuf3, "")	// databuf3初始化
_strmid(databuf3, databuf2, 15	5,8) // 从数据缓冲取出数据
Call F4_DataCopy	// 复制数据到 LS
return	
}endif	
}endif	
if([t:0032]==0){	// 接收是否超时
[c:EXT_SIO_CTRL01]=1	// 接收缓冲清除
set([b:LS700104])	// 接收超时标志
return	// 功能处理结束
}endif	
[t:0032]=[t:0032]-100	// 超时时间减少
_wait(1)	// 100ms等待
}endloop	

// 逐步接收字节数据到字地址 LS7500		
[t:0055]=20 // 循环次数设定		
[t:0056]=0 // 初始化		
[t:0057]=0 // 初始化		
loop([t:0055]){		
[w:LS7550]#[t:0057]=[w:LS7410]#[t	:0056]<<8 // 偶字节到更高列	
[t:0056]=[t:0056]+1	// 偏移量相加	
[w:LS7500]#[t:0057]=[w:LS7550]#[t:0057] [w:LS7410]#[t:0056]		
	// 高位和低位相加	
[t:0056]=[t:0056]+1	// 偏移量相加	
[t:0057]=[t:0057]+1	// 偏移量相加	
}endloop		
	_	

_hexasc2bin([w:LS8000], databuf3)	// 文字和数字转换
[w:LS8010]#[t:0033]=[w:LS8000]	// 逐步将当前值写入地址
[w:LS8100]=[w:LS8014]	// 复制设定值到写入地址

4. GP、PLC 和画面编辑软件版本

本例使用的 GP、PLC、画面编辑软件的版本号说明如下。请注意,如果你的 PLC 和触摸屏的型号与本例不同,需要对工程画面和设置作必要的调整。(→6.使用注意)

使用的GP和PLC

GP: GP2300

PLC: MELSEC A1SJH(通讯模块), 三菱电机

(协议: MITSUBISHI MELSEC-AnN(LINK))

画面编辑软件版本: GP-PRO/PBIII C-Package03(V7.23中文版)

5. 画面复制

如果您的工程文件已创建好,可以直接复制本例到您的工程中。请注意在复制时,地 址和画面号不能与工程中已有的地址和画面号重复。(→确认地址) 如果您使用的触摸屏型号与本例不同,需要对画面位置和设置进行适当的调整。

更改	X
描述:	确定
Verl.0	
_GP机型	
GP2000 Series 🔽	
GP2300	
_ 串行/扩展 SIO	
串行 I/F 切换	
设备/PLC机型(扩展串行I/F)	
Mitsubishi Electric Corp. 💌	
MITSUBISHI MELSEC-AnN(LINK)	
匚 转换地址	
└ 扩展SIO类型(串行I/F)	
Extend SIO Script	帮助 (丑)

在您的工程文件的"扩
展SIO型号"下选择
Extend SIO Script。
如果在 "串口 I/F 交
换"中选择"交换"则
用COM1 连接温控器。

🥦 工程程序。prw : 您的工程	文件 - 工程管理器 📃 🗖 🔀
工程(E) 画面/设置(S) 控制(C)	应用 (1) 帮助 (1)
%	画面复制(S) 转换图像(V) 压缩图像(C) 转换DXF(C)
	导入/导出文本画面(1) 转换地址(a) 转换调用画面号(1)
0	全局交叉参考(26) ▶
GP系统设置 工程 投稿	CF卡工具(1) 压缩工具(2) 重建工具(数据恢复/优化)(2)
🗋 新建 😪	文本索引表转换器 (2) 参输
 □ □<th> 協控 ① 根数 □ 模拟 ● 打印 </th>	 協控 ① 根数 □ 模拟 ● 打印
CP2300 IIISUBISHI TELSEC-ANN (LINK) Extend SIO Script	安更 管理器 Pro-face

在您工程文件里的工 程管理器[应用]下拉菜 单中,单击[画面复制] 命令。

选择		? 🛛
查找范围 (I): 📔) sample 🔹 🖨 🛗 📰 -	
nron_Thermo_	ExSIO_Script_Connection_CN.prw	
文件名 (20): □ □ 文件本刊 (71): □ □	ron_Thermo_ExSID_Script_Connection_CN 打开 (0)	
жн <u>х</u> шер: ры		
加速・		
显示机型:	GP2300	
设备/PLC机型:	MITSUBISHI MELSEC-AnN(LINK)	
扩展SIO类型:	Extend SID Script	



画面复制	
	目标文件
Omron_Thermo_ExSIO_S 选择	工程程序.prw
○ 全部 ○ 画面 从 1001 ÷	起始自 1001 :
到 1002 -	复制
画面类型:	
全部	郡毗公
○ 内部设置	
□ 内部设置	
□ 系统信息	
	◎ 復量 C 合并
◎ 復告 C 合开	CSV 数据传输
	□ 数据记录设置
	声音改査 (原葉
6 猎盖 C 合并	
	▼ 手提式/ST数据
© 覆盖 ○ 合并	▶ 全局功能键
▼ 扩展SIO脚本	▶ 全局振动
□ 安全设置	▶ 背景灯颜色变更
□ 任务计划	□ 全局互锁

画面复制 🛛 🔀
基本 一面 面已 豆 复制 :0 違 違 如 面 面 已 豆 复 制 :0 液 如 面 面 豆 豆 复 制 :0 液 如 四 面 面 豆 豆 复 制 :0 で 液 切 四 面 面 豆 豆 复 制 :0 で 液 切 四 面 面 豆 豆 复 制 :0 で 液 切 四 面 面 豆 豆 复 気 制 :0 で 液 切 四 面 面 豆 豆 复 気 制 :0 で 初 奶 切 切 思 し 記 波 製 制 :0 つ 数 据 ご む え 割 :0 つ 数 据 式 記 式 制 :0 こ の 数 据 記 豆 志 記 支 制 :0 こ の 数 据 記 三 支 制 :0 こ の 数 据 記 三 支 制 :0 こ の 数 据 こ の 数 据 こ こ の し 支 制 :0 こ の 数 据 こ こ の し 支 制 :0 こ の 数 据 :0 こ の 数 据 こ こ の 数 据 こ こ 気 制 :0 こ の 数 据 :0 こ の 、 の 本 来 む 豆 夏 制 :0 こ の 、 の 数 据 :0 こ の 、 の 数 居 こ 三 夏 制 :0 こ の 数 本 売 三 豆 夏 制 :0 こ の 、 の 本 売 三 豆 夏 制 :0 こ の こ こ 之 夏 気 制 :0 こ つ こ 之 豆 夏 制 :0 こ つ こ 之 之 こ こ 気 気 制 :0 こ つ 、 ひ に う 、 う 、 の こ つ 、 の こ つ 、 の こ つ 、 の こ つ 、 う 、 の 、 の こ こ 夏 制 : 0 つ 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の 、 の の こ つ こ こ し こ し こ つ い う い こ の こ つ こ つ こ つ い こ の こ の こ の こ つ こ つ い こ の こ の の こ の こ の こ の こ の こ の こ の こ の こ の こ の つ こ の こ の こ の こ の こ の つ こ つ い こ こ の つ こ こ の つ い こ の つ こ こ の こ の つ こ こ の つ つ こ つ こ つ こ つ つ こ つ こ つ つ こ つ こ つ つ こ つ つ こ つ つ つ こ つ つ つ こ つ つ つ つ つ つ つ つ つ つ つ つ つ
读取设置信息已经完成。 信息数据已映射到工程。
<u> </u>

6. 注意事项

如果您的触摸屏型号与本例中使用的触摸屏型号不一致(比本例屏的尺寸大),画面 设置是不同的,因此需要调整画面。可以使用本例的型号有: GP2000 系列和 GLC 系 列,采用的软件是 6.23 或 6.23 以上版本。

在屏上写入设定值时需要时间。

如果在"串口 I/F 交换"中做了与本例相同的设置,那么在COM1连接温控器,在 COM2连接PLC。

本例中,您可以使用 16 位的二进制数据(0~65535)。

如果使用了其它格式或负数数据,您需要在 D 脚本中进行相应的设置。

示例:使用 16 位二进制负数数据

Ŋ∂D─脚本函数 : F4_DataCopy		
文件 (P) 选项 (Q)		
函数名称: F4_DataCopy	显示工具箱 🔽	
公式:		函数:
_hexasc2bin[[w:LS8000], databuf3) // 文气 [w:LS8010]#(t:0033]=[w:LS8000] // 逐步 [w:LS8100]=[w:LS8014] // 复制	注和数字转换 步将当前值写入♯ 引设定值到写入♯	F0_INIT F1_BCC F1_R_FV1 F1_R_SV1 F1_W_SV1 F2_R_FV1_S F2_R_SV1_S F2_W_B1n2ASC F2_W_SV1_S F3_RCV RCV_Byte2Word V
数据长度: 32Bit	▼ 符号 +/- □	调用 新建
操作表达式正确		编辑删除
		复制粘贴
确定	帮助(H)	

<附录> 建立和编辑 D 脚本

建立新的D脚本



在画面编辑屏幕的 菜单栏里,点击[特 殊]下拉菜单中的[扩 展的SIO脚本]。

小 D-脚本编辑器 : Iain			×
文件 (2) 选项 (0)			
描述: Main	显示工具箱 🔽		
公式:(主函数)			函数: RO TNTT
 // 此脚本用于和 OMRON 温挖器 // Start process if[[b:LS700000]==1]{ Call F0_INIT loop[]{ // End process if[[b:LS7000001]==1]{ (b:LS700001]==1]{ 	的通讯 // 当通讯开关为 ON 时 // 扩展 SIO 初始化 // 循环 // 当"停止"按键触发时		F0_R11 F1_BCC F1_R_FV1 F1_R_SV1 F1_R_SV1 F2_R_FV1_S F2_R_SV1_S F2_W_BinZASC F2_W_SV1_S F2_W_SV1_S F2_R_SV1 S_RCV F4_DataCopy RCV_Byte2Word Send_Byte2Word
break //) }endif	"通讯33来 以循环中既出		Ver
<	>		1
<u> </u>	数据长度: 16 位 💌 符号 +/-	Г	
操作表达式正确		^	调用新建
			编辑册除
J		~	复制 粘贴
确定 取消	帮助(H)		

弹出D脚本编辑器窗口。 通过工具栏进 行地址输入和SIO操作 等。单击[确定]进行保 存。



LS 区是 GP 的一个内存区域,用来控制 GP 操作。 LS 区的构成如下表:

用户区是 GP 内部使用的一块地址区域,不能分配给 PLC。这个区域用作 GP 对部件和 Tags 的内部处理,不能被 PLC 控制。

确认地址

以下部分介绍当前工程中哪些地址已被部件、D 脚本等使用,画面号也可通过同样的方法确定。

🐕 Omron_Thermo_ExSIO_Sc	ript_Connection_CN.prv : 🔳 🗖 🔀
工程(E) 画面/设置(S) 控制(C)	应用 (11) 帮助 (11)
	画面复制(S) 转换图像(Y) 压缩图像(C) 转换DXF(X) 导入/导出文本画面(I) 转换地址(A) 转换调用画面号(L)
	全局交叉参考(R) ▶ 列表(L)
CP系統设置 エモ	Cr卡工具 (1) 圧縮工具 (2) 重建工具 (数据恢复/优化) (2) 文本索引表特换器 (2) 協控 (2) 授加 (2) (3) (3) (4) (4) (4) (5) (5) (5) (6) (7) (7) (7) (7) (7) (7) (7) (7
GP2300 IITSUBISHI ELSEC-ANN(LINK) EXtend SIO Script	変更 管理器 Pro-face
全局交叉参考列表	

在您工程的工程管理器[应用]
菜单中选择[全局交叉参考]-
[列表]命令

全局交叉参考列	表			
位地址 字地址	1			
地址	画面	<u>^</u>	地址	
LS700000	扩展SIO脚本, B1001			
LS700001				
LS700004 IS700100			● 己用 心	
LS700101	报警消息,扩展SIO脚本, V1001		こ所有(4)	
LS700102	报警消息		· MH W	
LS700103	报警消息		[
LS700104	报警消息, V1001		打开画面	
LS702000	B1001	<u>></u>	转换地址	
关闭帮助 (<u>H</u>)				

然后会弹出"全局交叉参考列 表"窗口,列表中将显示已使 用的地址和画面号。双击某个 地址或点击窗口右侧的[打开 画面],输入您要查找的地 址,然后使用该地址的画面将 自动打开。