

埃夫特智能装备股份有限公司

EFORT 机器人 C30 系统 现场总线说明手册

版本号：v1.2

修订历史

版本	日期（年/月/日）	原因（创建/修订）	创建或修订人
V1.0	20190611	创建	朱桂林
V1.1	20191122	补充	蒋中慧
V1.2	20191211	更新	汪万军

目录

1 现场总线数据监控.....	1
1.1 总线数据查看.....	1
1.2 设置总线数据输出.....	2
2 Profibus_DP 功能.....	3
2.1 综述.....	3
2.2 PLC 到 ROBOX.....	3
2.3 ROBOX 到 PLC.....	5
2.4 SIEMENTS 配置.....	8
2.4.1 硬件组态.....	8
2.4.2 地址和通信数据配置.....	11
2.4.3 变量监控.....	12
2.4.4 工程下载.....	13
2.5 应用示例.....	14
2.5.1 修改 DP 地址.....	16
3 Modbus-tcp 功能.....	17
3.1 综述.....	17
3.2 系统层协议内容.....	17
3.3 示教器层协议内容.....	18
3.4 应用举例.....	19

1 现场总线数据监控

现场总线监控主要是用来查看总线通讯中的数据，同时可以设置数据的输出值。

目前主要包括 modbus 和 profibus 的通讯协议数据监控，其中 **modbus** 的输入输出各自包括 64 个 bool、6 个 int、24 个 float 类型的数据；**profibus** 的输入输出各自包括 64 个 bool、6 个 int、24 个 float 类型的数据。

1.1 总线数据查看

步骤	图片	描述
1. 打开现场总线监控界面。		① 点击状态栏下“监控”按钮。 ② 点击下拉菜单中“现场总线”。
2. 选中需要查看的变量。		① 在界面中选择需要查看的通讯协议的数据。这里以 modbus 为例。 ② 选择“名称”下面加（减）号可以展开（收起）数据列表。

3. 查看变量数据



- ① 展开数据中可以查看以下数据
第1列是数据名称；
第2列数据类型；
第3列是数据的值；
第4列是对应的示教器程序中的变量。

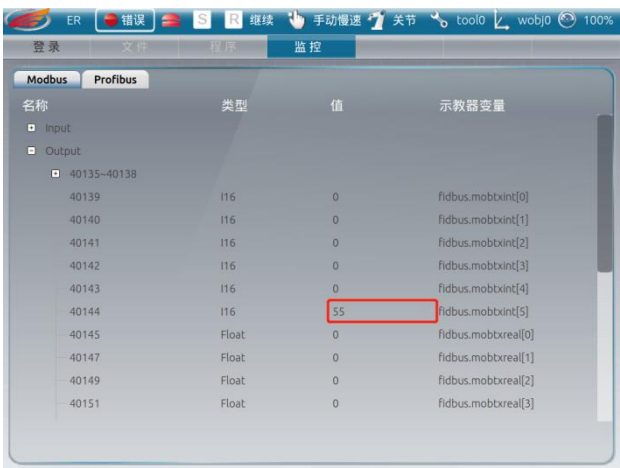
1.2设置总线数据输出

步骤

图片

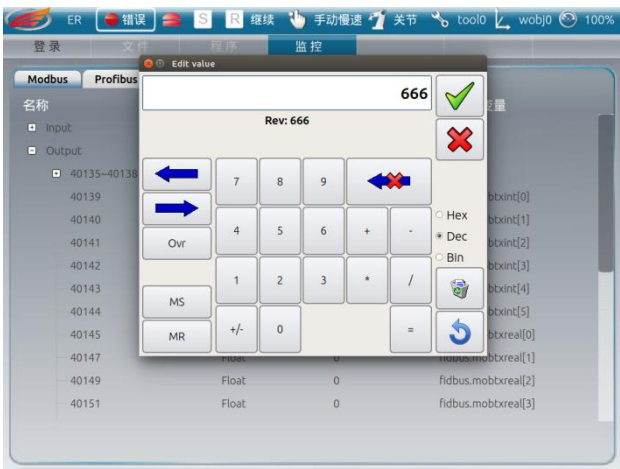
描述

1. 打开数据监控列表的输出部分



- ① 按 1.1 中方式打开数据监控列表的输出部分。
② 点击需要输出的变量的“值”这一列，则弹出输出框。以一个 real 变量为例。

2. 设置输出数据的值



- ① 设置输出的数据。
② 点击“√”则数据输出；点击“×”取消输出。

2. 输出数据



① 数据输出如图所示。

2 Profibus_DP 功能

2.1 综述

PROFIBUS DP 功能支持机器人做从站（slave），其地址为 2，包括 6 个 input block 和 6 个 output block，每个 block 包含 32byte，数据类型为 unsigned int（2 byte）或者是 float（4 byte）。数据的存储模式为大端模式，所以当将 unsigned int 拆分成 byte 类型时，注意数据的高低位转换。

2.2 PLC 到 ROBOX

PLC 到 ROBOX，PLC 作为发送端，机器人作为接受端。接收数据主要包括系统变量和 TPU 变量。其中 Bus_Get[0]-Bus_Get[191]为数据接收的 192 BYTE。

其中 fidbus.pfbrxbool[63]、fidbus.pfbrxint[5] 及 fidbus.pfbrxreal[23]为示教器上的 module 变量，通过从 plc 获取状态，来控制程序的运行。

物理地址	寄存器序号	RDE 变量	单位	子单 位	备注
接收数据					
Bus_Get[0]-[1]	R(2451)	/	UINT	BOOL	Bit0: 上/下伺服（脉冲） Bit1: 运行（脉冲） Bit2: 停止（脉冲） Bit3: 清除报警（脉冲） Bit4: 加载程序（脉冲） Bit5: 重新开始（程序回到第一行）（脉冲） Bit6: Plc 报警（高电平）
Bus_Get[2]-[3]	R(2452)	/	UINT	Bool	系统预留 BOOL 变量，

					用户不可以使用
Bus_Get[4]-[5]	R(2453)	/	UINT	UINT	运行速度（全局）
Bus_Get[6]-[7]	R(2454)	/	UINT	UINT	加载目标程序号，例： 首先设置目标程序号为2，然后给 Bus_Get[0]-[1]的Bit4高电平触发信号，完成程序加载（在程序运行过程中不可加载）
Bus_Get[8]-[19]	R(2455)-R(2460)	/	UINT	UINT	系统预留 UINT 类型变量， 用户不可以使用
Bus_Get[20]-[21]	R(2461)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbxbool[0]-[15]
Bus_Get[22]-[23]	R(2462)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbxbool[16]-[31]
Bus_Get[24]-[25]	R(2463)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbxbool[32]-[47]
Bus_Get[26]-[27]	R(2464)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbxbool[48]-[63]
Bus_Get[28]-[29]	R(2365)	/	UINT	UINT	对应 fidbus.pfbxint[0]
Bus_Get[30]-[31]	R(2366)	/	UINT	UINT	对应 fidbus.pfbxint[1]
Bus_Get[32]-[33]	R(2367)	/	UINT	UINT	对应 fidbus.pfbxint[2]
Bus_Get[34]-[35]	R(2368)	/	UINT	UINT	对应 fidbus.pfbxint[3]
Bus_Get[36]-[37]	R(2369)	/	UINT	UINT	对应 fidbus.pfbxint[4]
Bus_Get[38]-[39]	R(2370)	/	UINT	UINT	对应 fidbus.pfbxint[5]
Bus_Get[40]-[87]	RR(2451)-R(2462)	/	FLOAT	FLOAT	系统预留 FLOAT 类型变量， 用户不可以使用
Bus_Get[88]-[91]	RR(2463)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[0]
Bus_Get[92]-[95]	RR(2464)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[1]
Bus_Get[96]-[99]	RR(2465)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[2]
Bus_Get[100]-[103]	RR(2466)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[3]
Bus_Get[104]-[107]	RR(2467)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[4]
Bus_Get[108]-[111]	RR(2468)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[5]
Bus_Get[112]-[115]	RR(2469)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[6]

			T		
Bus_Get[116]-[119]	RR(2470)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[7]
Bus_Get[120]-[123]	RR(2471)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[8]
Bus_Get[124]-[127]	RR(2472)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[9]
Bus_Get[128]-[131]	RR(2473)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[10]
Bus_Get[132]-[135]	RR(2474)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[11]
Bus_Get[136]-[139]	RR(2475)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[12]
Bus_Get[140]-[143]	RR(2476)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[13]
Bus_Get[144]-[147]	RR(2477)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[14]
Bus_Get[148]-[151]	RR(2478)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[15]
Bus_Get[152]-[155]	RR(2479)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[16]
Bus_Get[156]-[159]	RR(2480)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[17]
Bus_Get[160]-[163]	RR(2481)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[18]
Bus_Get[164]-[167]	RR(2482)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[19]
Bus_Get[168]-[171]	RR(2483)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[20]
Bus_Get[172]-[175]	RR(2484)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[21]
Bus_Get[176]-[179]	RR(2485)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[22]
Bus_Get[180]-[183]	RR(2486)	/	FLOAT	FLOAT	对应 fidbus.pfbxreal[23]
Bus_Get[184]-[191]	/	/	/	/	未定义

2.3 ROBOX 到 PLC

ROBOX 到 PLC，机器人作为发送端，PLC 作为接受端。发送数据主要包括系统变量和 TPU 变量。其中 Bus_Set[0]-Bus_Set[191]为数据发送的 192 BYTE。

其中 fidbus.pfbtxbool[63] 、fidbus.pfbtxint[5] 及 fidbus.pfbtxreal[23]为示教器上的 module 变量，向 plc 发送状态，来反馈程序的运行状态。

物理地址	寄存器序号	RD E 变 量	单位	子单 位	备注
发送数据					
Bus_Set[0]-[1]	R(2401)	/	UINT	BOOL	Bit0: 手动状态 Bit1: 自动状态 Bit2: 远程状态 Bit3: 伺服状态 Bit4: 报警状态 Bit5: 急停状态 Bit6: 程序运行状态 Bit7: 安全位置 1 状态 Bit8: 安全位置 2 状态 Bit9: 安全位置 3 状态 Bit10: 安全位置 4 状态
Bus_Set[2]-[3]	R(2402)	/	UINT	Bool	系统预留 BOOL 类型变量，用户不可用
Bus_Set[4]-[5]	R(2403)	/	UINT	UINT	运行速度（全局）
Bus_Set[6]-[7]	R(2404)	/	UINT	UINT	报警代码
Bus_Set[8]-[9]	R(2405)	/	UINT	UINT	报警代码
Bus_Set[10]-[11]	R(2406)	/	UINT	UINT	程序号（用于反馈加载目标程序是否完成，例：在 Plc 端加载程序号为 2 的程序，如果加载完成，则程序号反馈为 2，否则为其他值）
Bus_Set[12]-[19]	R(2407)-R(2410)	/	UINT	UINT	系统预留 UINT 类型变量，用户不可用
Bus_Set[20]-[21]	R(2411)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbtxbool[0]-[15]
Bus_Set[22]-[23]	R(2412)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbtxbool[16]-[31]
Bus_Set[24]-[25]	R(2413)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbtxbool[32]-[47]
Bus_Set[26]-[27]	R(2414)	/	UINT	BOOL	Bit0~Bit15 对应 fidbus.pfbtxbool[48]-[63]
Bus_Set[28]-[29]	R(2415)	/	UINT	UINT	对应 fidbus.pfbtxint[0]
Bus_Set[30]-[31]	R(2416)	/	UINT	UINT	对应 fidbus.pfbtxint[1]
Bus_Set[32]-[33]	R(2417)	/	UINT	UINT	对应 fidbus.pfbtxint[2]

Bus_Set[34]-[35]	R(2418)	/	UINT	UINT	对应 fidbus.pfbtxint[3]
Bus_Set[36]-[37]	R(2419)	/	UINT	UINT	对应 fidbus.pfbtxint[4]
Bus_Set[38]-[39]	R(2420)	/	UINT	UINT	对应 fidbus.pfbtxint[5]
Bus_Set[40]-[63]	RR(2401)-RR(2406)	/	FLOAT	FLOAT	J1~J6 关节角度值
Bus_Set[64]-[87]	RR(2407)-RR(2412)	/	FLOAT	FLOAT	系统预留 FLOAT 类型变量,用户不可用
Bus_Set[88]-[91]	RR(2413)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[0]
Bus_Set[92]-[95]	RR(2414)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[1]
Bus_Set[96]-[99]	RR(2415)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[2]
Bus_Set[100]-[103]	RR(2416)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[3]
Bus_Set[104]-[107]	RR(2417)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[4]
Bus_Set[108]-[111]	RR(2418)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[5]
Bus_Set[112]-[115]	RR(2419)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[6]
Bus_Set[116]-[119]	RR(2420)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[7]
Bus_Set[120]-[123]	RR(2421)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[8]
Bus_Set[124]-[127]	RR(2422)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[9]
Bus_Set[128]-[131]	RR(2423)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[10]
Bus_Set[132]-[135]	RR(2424)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[11]
Bus_Set[136]-[139]	RR(2425)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[12]
Bus_Set[140]-[143]	RR(2426)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[13]
Bus_Set[144]-[147]	RR(2427)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[14]
Bus_Set[148]-[151]	RR(2428)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[15]
Bus_Set[152]-[155]	RR(2429)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[16]
Bus_Set[156]-[159]	RR(2430)	/	FLOAT	FLOAT	对应 fidbus.pfbtxreal[17]

Bus_Set[160]-[163]	RR(2431)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[18]
Bus_Set[164]-[167]	RR(2432)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[19]
Bus_Set[168]-[171]	RR(2433)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[20]
Bus_Get[172]-[175]	RR(2434)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[21]
Bus_Get[176]-[179]	RR(2435)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[22]
Bus_Get[180]-[183]	RR(2436)	/	FLOA T	FLOA T	对应 fidbus.pfbtxreal[23]
Bus_Set[184]-[191]	/	/	/	/	未定义

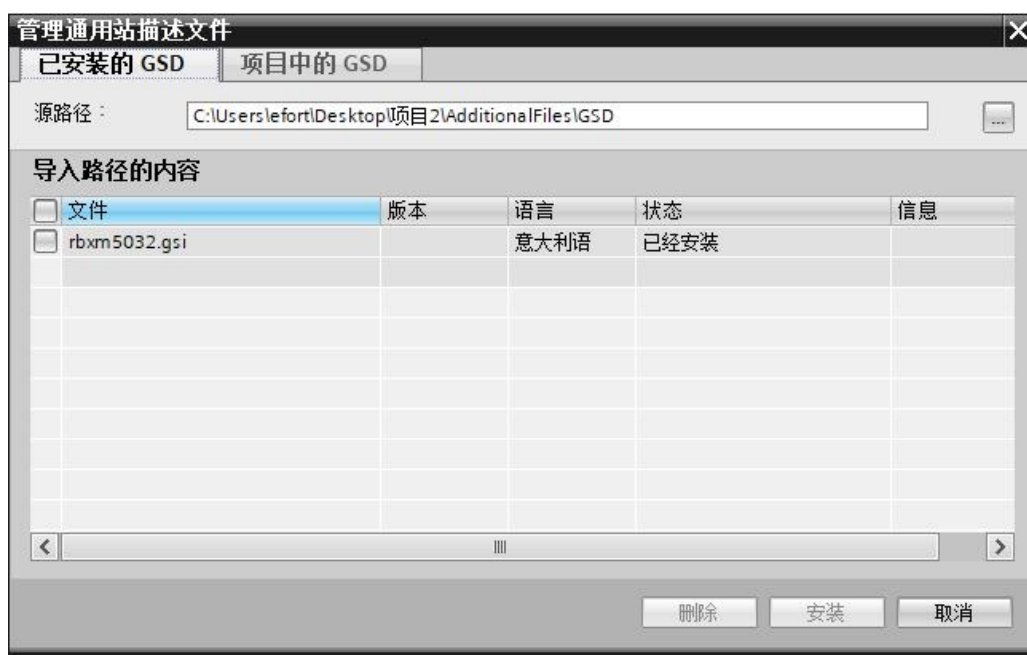
2.4 SIEMENTS 配置

2.4.1 硬件组态

在博途软件中新建项目，本案例选取西门子 S7-1200PLC，其型号为 CPU 1215C DC/DC/DC，PROFIBUS-DP 接口型号为 CM 1243-5，在本案例中，PLC 作为 PROFIBUS DP-MASTER。

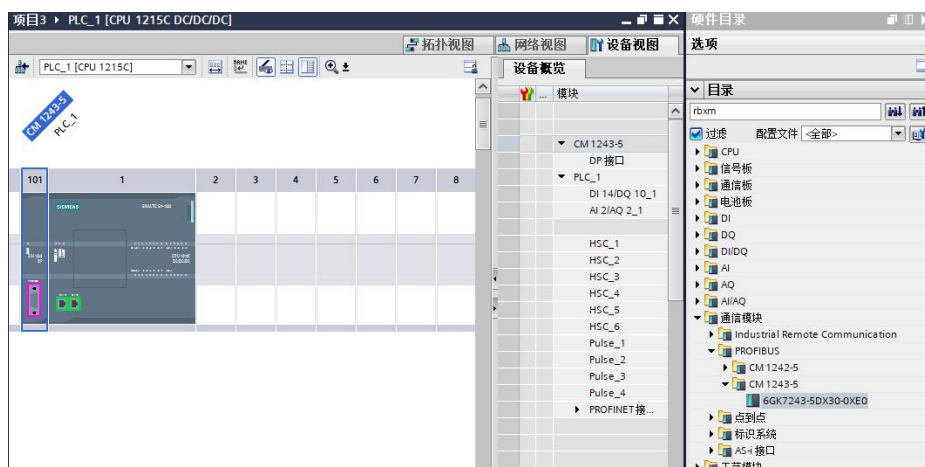
第一步：添加 GSD 文件：

在博途软件里，依次点击“选项”、“管理通用站描述文件”，如下图：



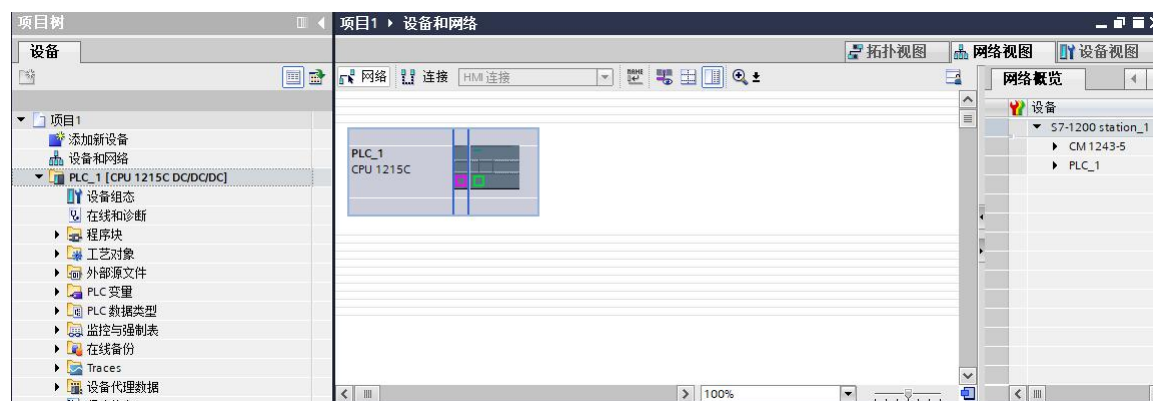
选取 rbxm502.gsi 文件，并点击安装，安装因电脑而已，可能需要一点时间。

第二步：双击“设备组态”，并将视图切换成设备视图，如下图：



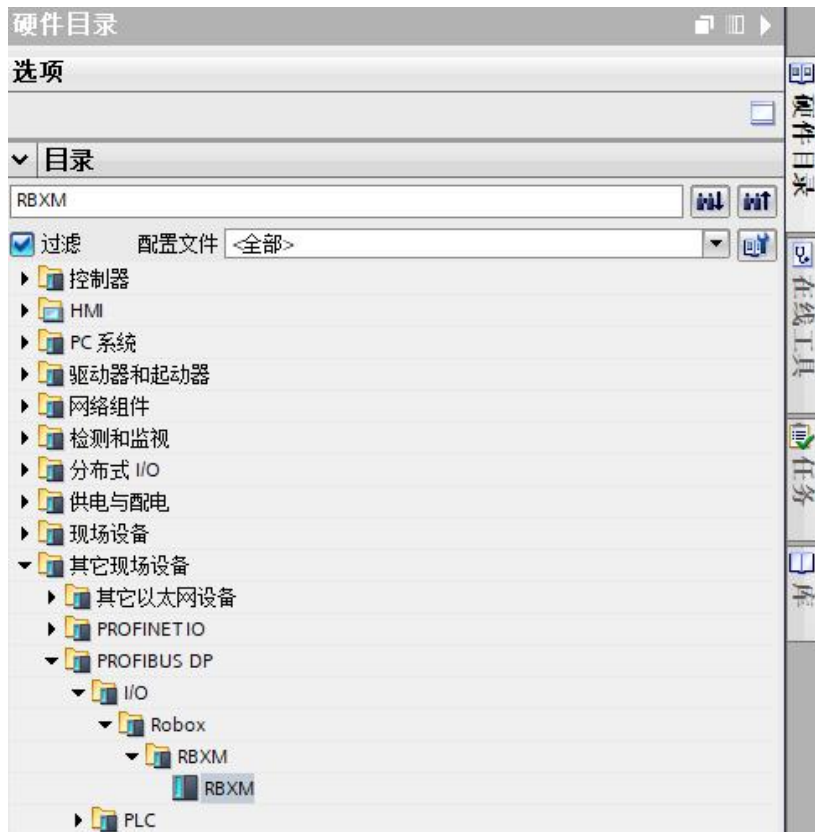
在右侧“硬件目录”、“通讯模块”下选取 CM 1243-5 通讯模块，双击加载到工程中。

第三步：双击“设备组态”，并将视图切换成网络视图，如下图：

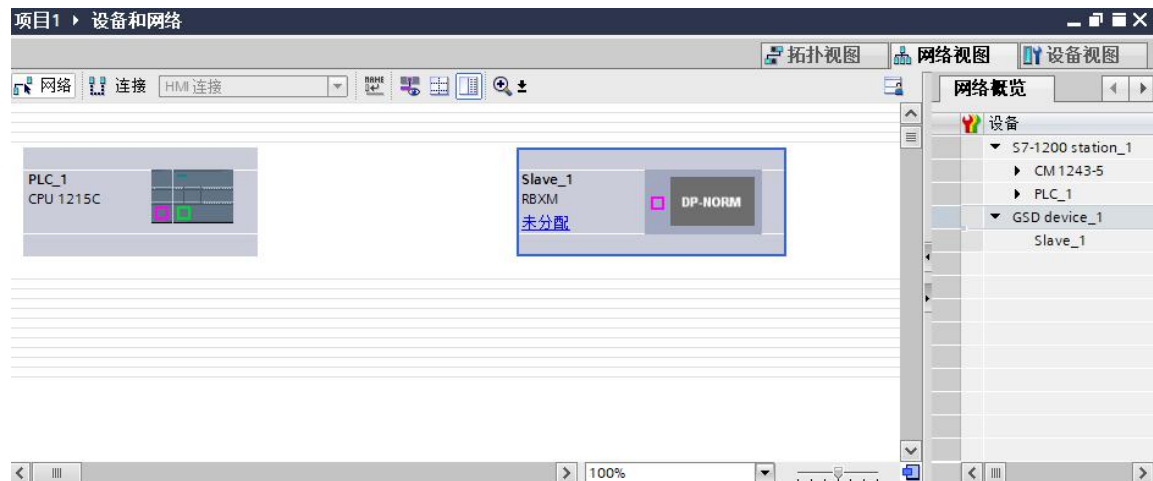


第四步：在右侧“硬件目录”功能栏里的搜索栏里输入 rbxm502，出现如下搜索

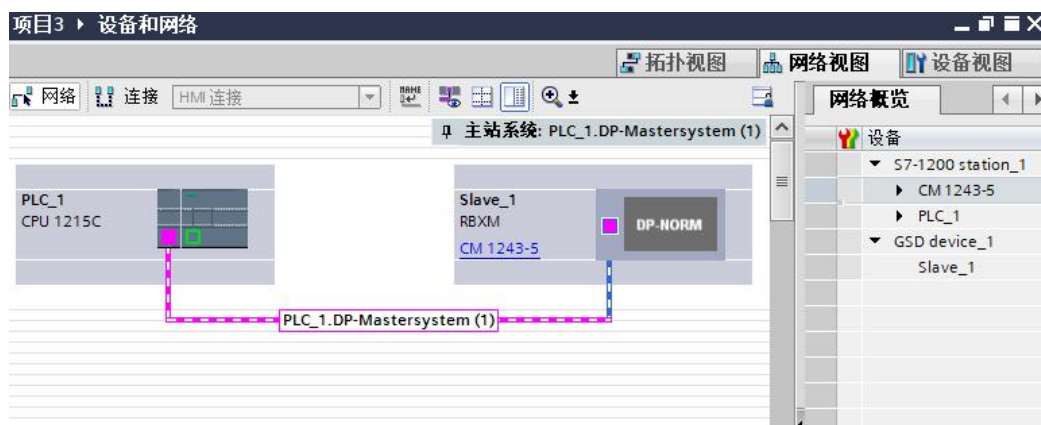
结果：



第五步：展开 **RBXM**,并按住鼠标左键，将其拖拽到网络视图中，如下图：



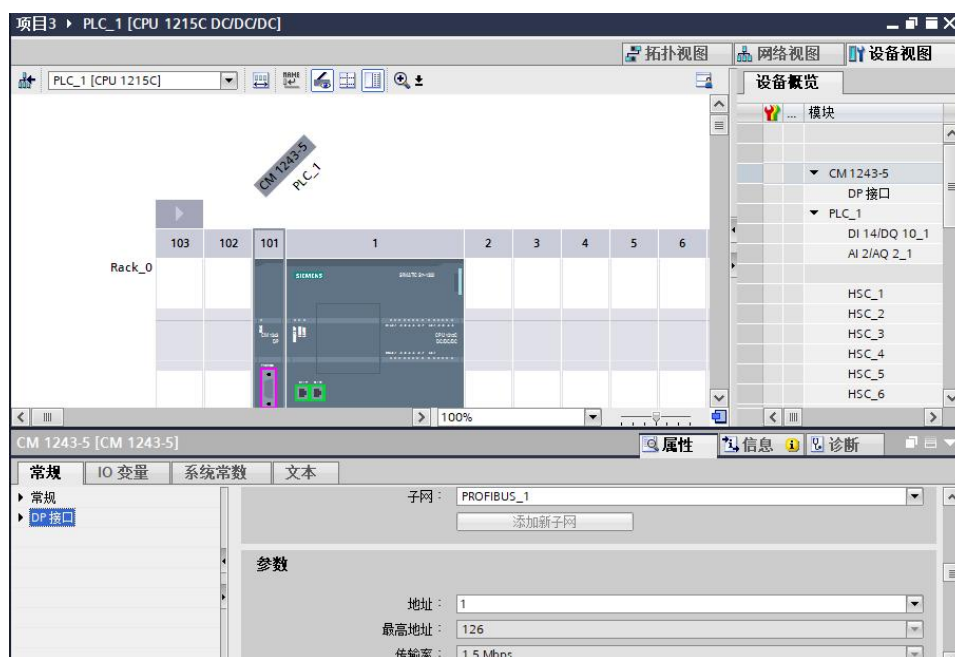
第六步：按住鼠标左键选中 **PLC_1** 的 **DP** 口，并与 **slave_1** 的 **DP** 口相连，如下图所示：



做到这里，PLC 与机器人的硬件组态就已经结束。

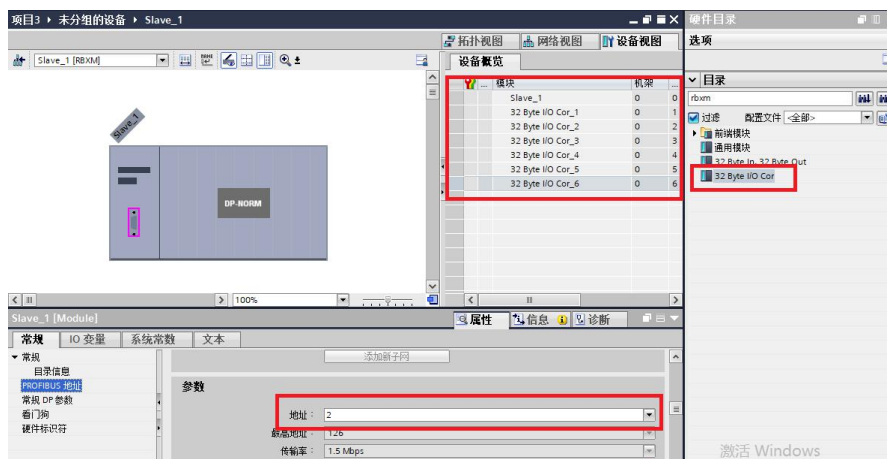
2.4.2 地址和通信数据配置

第一步：设置主站地址，在网络视图中双击 **DP** 接口进入设备视图中，双击 **DP-MASTER** 接口，出现如下图参数设置：



设置主站地址，本案例 PLC 作为主站，设置主站地址为 1。

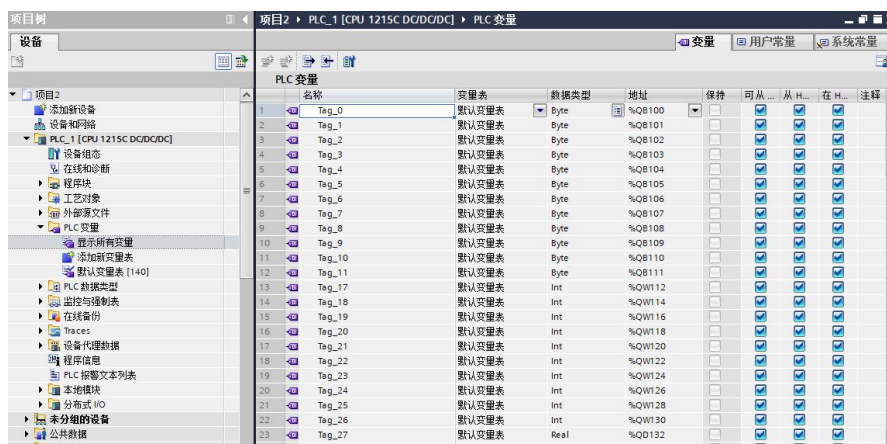
第二步：配置从站参数，在网络视图中双击从站模块进入设备视图中，双击 **DP-SLAVE** 接口，出现如下图参数设置：



在硬件目录下，选取 32 Byte I/O Cor（默认选取此模块，否则会导致通讯异常），拖动到右侧模块中，需要配置 6 个 BLOCK；设置从站地址，本案例 ROBOX RP 1 作为主站，设置从站地址为 2。

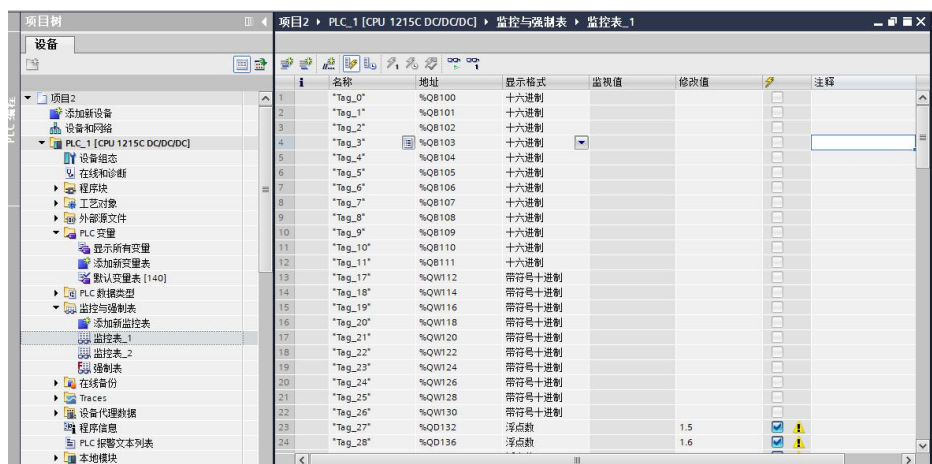
2.4.3 变量监控

第一步：在右侧 PLC 变量，配置变量表，如下图所示：



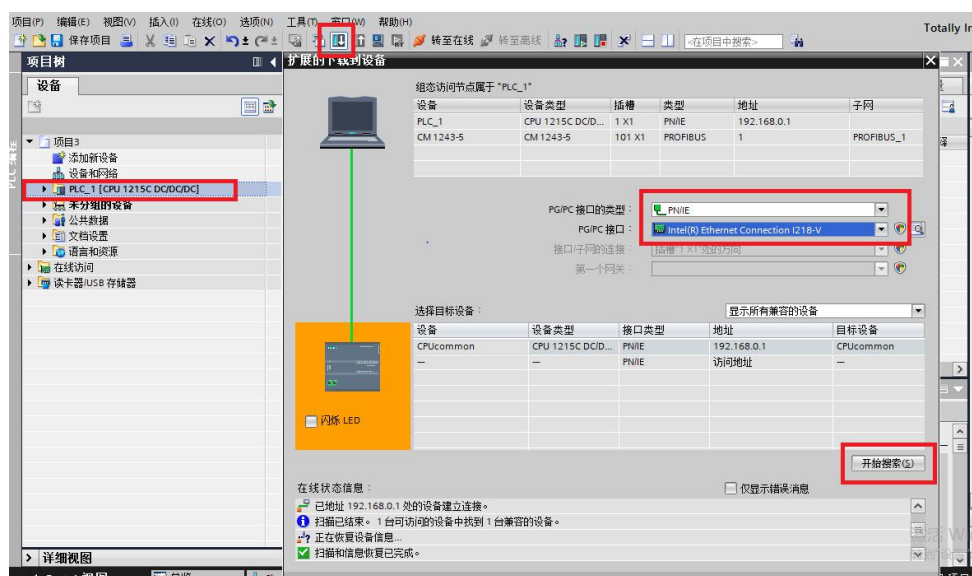
注意：由于 Robox 控制器端配置的数据类型是 UINT 和 Float 类型，所以对应的 Siemens PLC 端应该也配置成相应类型，如果使用 BYTE 类型（例如 QB100 和 QB101 为两个 byte，对应 Robox 控制器的一个 Uint，如果 QB100 的值为 0，QB101 的值为 32，则 Robox 端接收的数据为 32），请注意高低位转换问题。

第二步：在右侧监控与强制表，配置需要监控的变量，如下图所示：

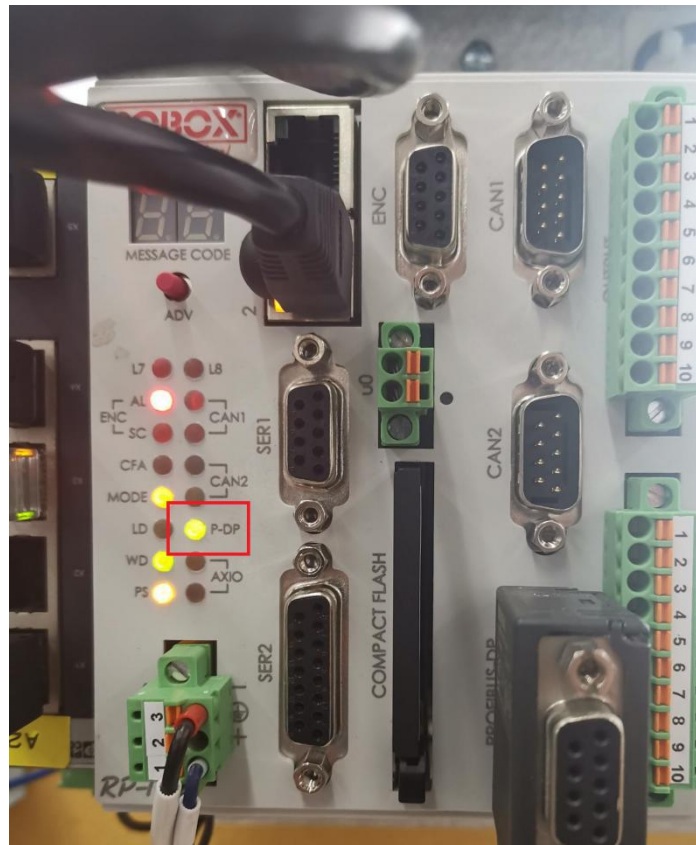


2.4.4 工程下载

第一步：将配置好的项目下载到 PLC 中，如下图：



下载完成后，即可与 ROBOX 控制器进行通讯，这时可以看到 ROBOX 控制器上的 P-DP 灯常亮，证明通讯建立完成，注意要使用 Siemens 标准的通讯线缆，否则会出现通讯异常现象。



2.5 应用示例

PLC 输出变量如下：

*系统状态 1"	%QB100	无符号十进制	0	0	
*系统状态 2"	%QB101	二进制	2#0000_0011	2#0000_0011	
*speed"	%QW102	无符号十进制	70	70	
*系统预留 INT 1"	%QW104	无符号十进制	0	0	
*系统预留 INT 2"	%QW106	无符号十进制	0	0	
*系统预留 INT 3"	%QW108	无符号十进制	0	0	
*系统预留 INT 4"	%QW110	无符号十进制	0	0	
*TPU 预留 BOOL 1"	%QW112	无符号十进制	256	256	
*TPU 预留 BOOL 2"	%QW114	无符号十进制	0	0	
*TPU 预留 BOOL 3"	%QW116	无符号十进制	65535	65535	
*TPU 预留 BOOL 4"	%QW118	无符号十进制	65535	65535	
*TPU 预留 INT 1"	%QW120	无符号十进制	1	1	
*TPU 预留 INT 2"	%QW122	无符号十进制	2	2	
*TPU 预留 INT 3"	%QW124	无符号十进制	3	3	
*TPU 预留 INT 4"	%QW126	无符号十进制	4	4	
*TPU 预留 INT 5"	%QW128	无符号十进制	5	5	
*TPU 预留 INT 6"	%QW130	无符号十进制	6	6	
*系统预留 FLOAT 1"	%QD132	浮点数	0.0	0.0	
*系统预留 FLOAT 2"	%QD136	浮点数	0.0	0.0	
*系统预留 FLOAT 3"	%QD140	浮点数	0.0	0.0	
*系统预留 FLOAT 4"	%QD144	浮点数	0.0	0.0	
*系统预留 FLOAT 5"	%QD148	浮点数	0.0	0.0	
*系统预留 FLOAT 6"	%QD152	浮点数	0.0	0.0	
*TPU 预留 FLOAT 1"	%QD156	浮点数	1.1	1.1	

*TPU 预留 FLOAT 1"	%QD156	浮点数	1.1	1.1	☒	
*TPU 预留 FLOAT 2"	%QD160	浮点数	1.2	1.2	☒	
*TPU 预留 FLOAT 3"	%QD164	浮点数	1.33	1.33	☒	
*TPU 预留 FLOAT 4"	%QD168	浮点数	1.444	1.444	☒	
*TPU 预留 FLOAT 5"	%QD172	浮点数	1.5555	1.5555	☒	
*TPU 预留 FLOAT 6"	%QD176	浮点数	1.666	1.666	☒	
*TPU 预留 FLOAT 7"	%QD180	浮点数	1.777	1.777	☒	
*TPU 预留 FLOAT 8"	%QD184	浮点数	1.8888	1.8888	☒	
*TPU 预留 FLOAT 9"	%QD188	浮点数	1.999	1.999	☒	
*TPU 预留 FLOAT 10"	%QD192	浮点数	2.1111	2.1111	☒	

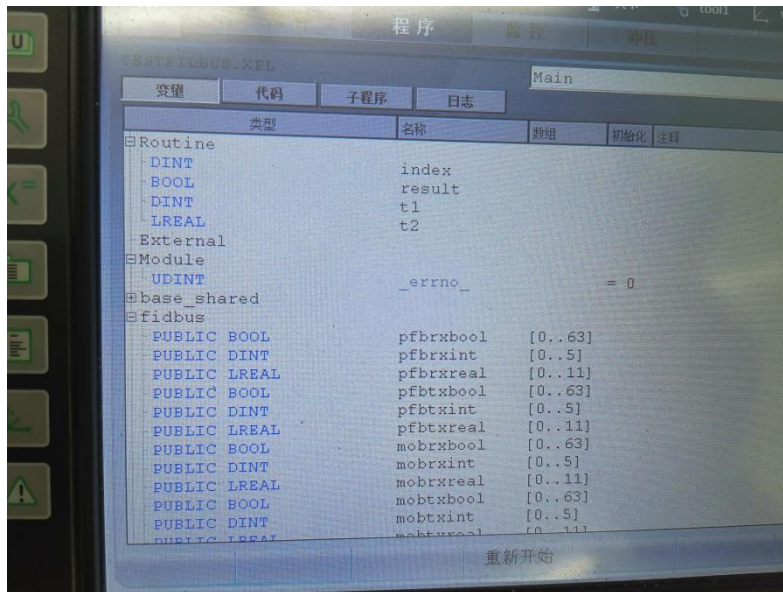
PLC 输入变量如下:

*系统状态 1(1)"	%B100	无符号十进制	0		☐	
*系统状态 2(1)"	%B101	二进制	2#0000_1010		☐	
*运行速度"	%IW102	无符号十进制	20		☐	
*系统预留 INT 1(1)"	%IW104	无符号十进制	0		☐	
*系统预留 INT 2(1)"	%IW106	无符号十进制	0		☐	
*系统预留 INT 3(1)"	%IW108	无符号十进制	0		☐	
*系统预留 INT 4(1)"	%IW110	无符号十进制	0		☐	
*TPU 预留 BOOL 1(1)"	%IW112	无符号十进制	65535		☐	
*TPU 预留 BOOL 2(1)"	%IW114	无符号十进制	65535		☐	
*TPU 预留 BOOL 3(1)"	%IW116	无符号十进制	65535		☐	
*TPU 预留 BOOL 4(1)"	%IW118	无符号十进制	65535		☐	
*TPU 预留 INT 1(1)"	%IW120	无符号十进制	1		☐	
*TPU 预留 INT 2(1)"	%IW122	无符号十进制	2		☐	
*TPU 预留 INT 3(1)"	%IW124	无符号十进制	3		☐	
*TPU 预留 INT 4(1)"	%IW126	无符号十进制	4		☐	
*TPU 预留 INT 5(1)"	%IW128	无符号十进制	5		☐	
*TPU 预留 INT 6(1)"	%IW130	无符号十进制	6		☐	
*关节角 1"	%ID132	浮点数	15.80901		☐	
*关节角 2"	%ID136	浮点数	29.19197		☐	
*关节角 3"	%ID140	浮点数	-69.931		☐	
*关节角 4"	%ID144	浮点数	-16.02797		☐	
*关节角 5"	%ID148	浮点数	0.0		☐	
*关节角 6"	%ID152	浮点数	0.0		☐	

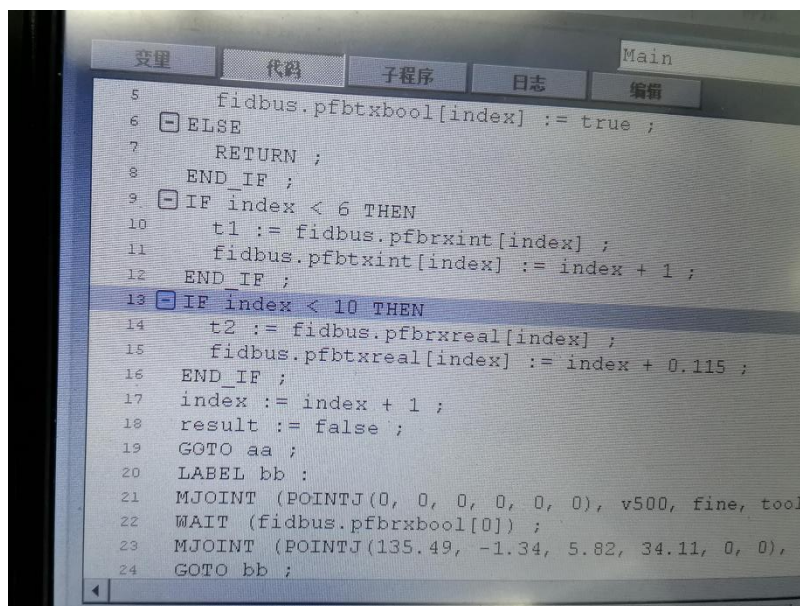
*笛卡尔 1"	%ID156	浮点数	448.0689		☐	
*笛卡尔 2"	%ID160	浮点数	126.8668		☐	
*笛卡尔 3"	%ID164	浮点数	617.4863		☐	
*笛卡尔 4"	%ID168	浮点数	-0.2189619		☐	
*笛卡尔 5"	%ID172	浮点数	0.0		☐	
*笛卡尔 6"	%ID176	浮点数	0.0		☐	
*关节速度 1"	%ID180	浮点数	0.0		☐	
*关节速度 2"	%ID184	浮点数	0.0		☐	
*关节速度 3"	%ID188	浮点数	0.0		☐	
*关节速度 4"	%ID192	浮点数	0.0		☐	
*关节速度 5"	%ID196	浮点数	0.0		☐	
*关节速度 6"	%ID200	浮点数	0.0		☐	
*关节力矩 1"	%ID204	浮点数	0.0		☐	
*关节力矩 2"	%ID208	浮点数	0.0		☐	
*关节力矩 3"	%ID212	浮点数	0.0		☐	
*关节力矩 4"	%ID216	浮点数	0.0		☐	
*关节力矩 5"	%ID220	浮点数	0.0		☐	
*关节力矩 6"	%ID224	浮点数	0.0		☐	
*系统预留 FLOAT 1(1)"	%ID228	浮点数	0.0		☐	
*系统预留 FLOAT 2(1)"	%ID232	浮点数	0.0		☐	
*系统预留 FLOAT 3(1)"	%ID236	浮点数	0.0		☐	
*系统预留 FLOAT 4(1)"	%ID240	浮点数	0.0		☐	
*系统预留 FLOAT 5(1)"	%ID244	浮点数	0.0		☐	
*系统预留 FLOAT 6(1)"	%ID248	浮点数	0.0		☐	

*TPU 预留 FLOAT 1(1)"	%ID252	浮点数	0.115		☐	
*TPU 预留 FLOAT 2(1)"	%ID256	浮点数	1.115		☐	
*TPU 预留 FLOAT 3(1)"	%ID260	浮点数	2.115		☐	
*TPU 预留 FLOAT 4(1)"	%ID264	浮点数	3.115		☐	
*TPU 预留 FLOAT 5(1)"	%ID268	浮点数	4.115		☐	
*TPU 预留 FLOAT 6(1)"	%ID272	浮点数	5.115		☐	
*TPU 预留 FLOAT 7(1)"	%ID276	浮点数	6.115		☐	
*TPU 预留 FLOAT 8(1)"	%ID280	浮点数	7.115		☐	
*TPU 预留 FLOAT 9(1)"	%ID284	浮点数	8.115		☐	
*TPU 预留 FLOAT 10(1)"	%ID288	浮点数	9.115		☐	

在 TPU 端，新建一个 RPL 程序，在变量界面，可以看到 fidbus，其中的变量对应于第 2 章和第 3 章中的输入输出变量，也对应于上图所示的 TPU 预留变脸，两者是一一对应关系。

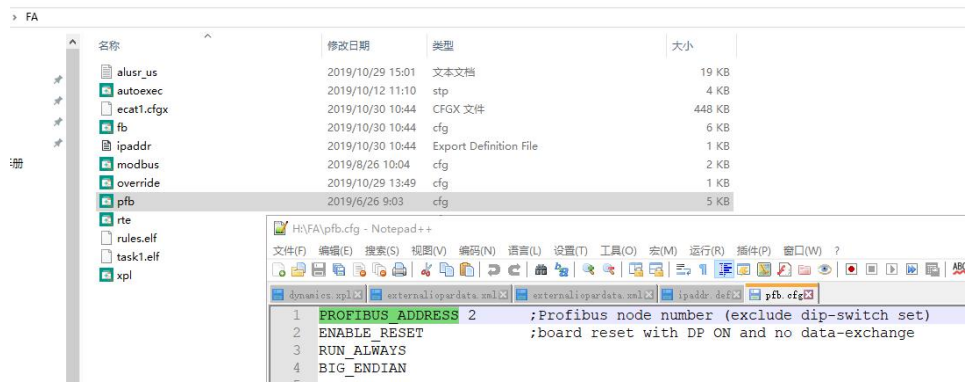


在 RPL 程序中，应用实例如下：



2.5.1 修改 DP 地址

将机器人回到零点，关闭电源，将控制器中的卡拔出，通过读卡器读取其中的数据，找到 FA/pfb.cfg 文件，打开此文件，修改 PROFIBUS_ADDRESS 值为从站地址。



3 Modbus-tcp 功能

3.1 综述

Modbus tcp 功能只支持机器人做从站 (slave), IP 地址即为控制器设置地址, 端口号固定为 502。协议内容分为两部分: 系统层协议、示教器层协议内容。

3.2 系统层协议内容

系统协议内容, 这部分由开发人员进行配置及维护, 主要是读取及设置机器人状态及运动参数等。用户只能通过固定的地址对固定的变量进行读写, 其中 40001~40066 为机器人发送给 PLC 的数据, 40101 为机器人从 PLC 接受的数据。具体的地址与变量映射关系如表 2.1 所示。

物理地址	类型	备注
机器人发送数据		
40001	I16	Bit0: 手动状态 Bit1: 自动状态 Bit2: 远程状态 Bit3: 伺服状态 Bit4: 报警状态 Bit5: 急停状态 Bit6: 程序运行状态 Bit7: 安全位置 1 状态 Bit8: 安全位置 2 状态 Bit9: 安全位置 3 状态 Bit10: 安全位置 4 状态
40002	I16	系统预留 BOOL 变量, 用户不可用
40003	I16	机器人运行速度(全局)
40004	I16	报警代码 1

40005	l16	报警代码 2
40006	l16	程序号（用于反馈加载目标程序是否完成，例：在 Plc 端加载程序号为 2 的程序，如果加载完成，则程序号反馈为 2，否则为其他值）
40007-40010	l16	系统预留 Int 变量， 用户不可用
40011 ~ 40021	float	J1~J6 关节角度值
40023 ~ 40033	float	系统预留 Float 变量， 用户不可用
机器人接收数据		
40101	l16	Bit0: 上/下伺服（脉冲） Bit1: 运行（脉冲） Bit2: 停止（脉冲） Bit3: 清除报警（脉冲） Bit4: 加载程序（脉冲） Bit5: 重新开始（程序回到第一行）（脉冲） Bit6: Plc 报警（高电平）
40102	l16	系统预留 BOOL 变量， 用户不可用
40103	l16	设置机器人速度（上升沿）
40104	l16	加载目标程序号，例：首先设置目标程序号为 2，然后给 40101 的 Bit4 高电平触发信号，完成程序加载（在程序运行过程中不可加载）
40105 ~ 401110	l16	系统预留 Int 变量， 用户不可用
40111 ~ 40133	float	系统预留 Float 变量， 用户不可用

表 2.1 系统协议内容

3.3 示教器层协议内容

示教器层协议内容，该部分有终端用户自行配置及维护，**该部分工程只有在使用机器人语言编程中可以使用，不支持在工艺包中使用**。原始协议中接收端和发送端各留有 64 个 BOOL, 6 个 int, 23 个 float 数据接口，终端用户可以通过示教器编写程序读写主站 PLC 的数据。具体的地址与变量映射关系如表 3.1 所示。

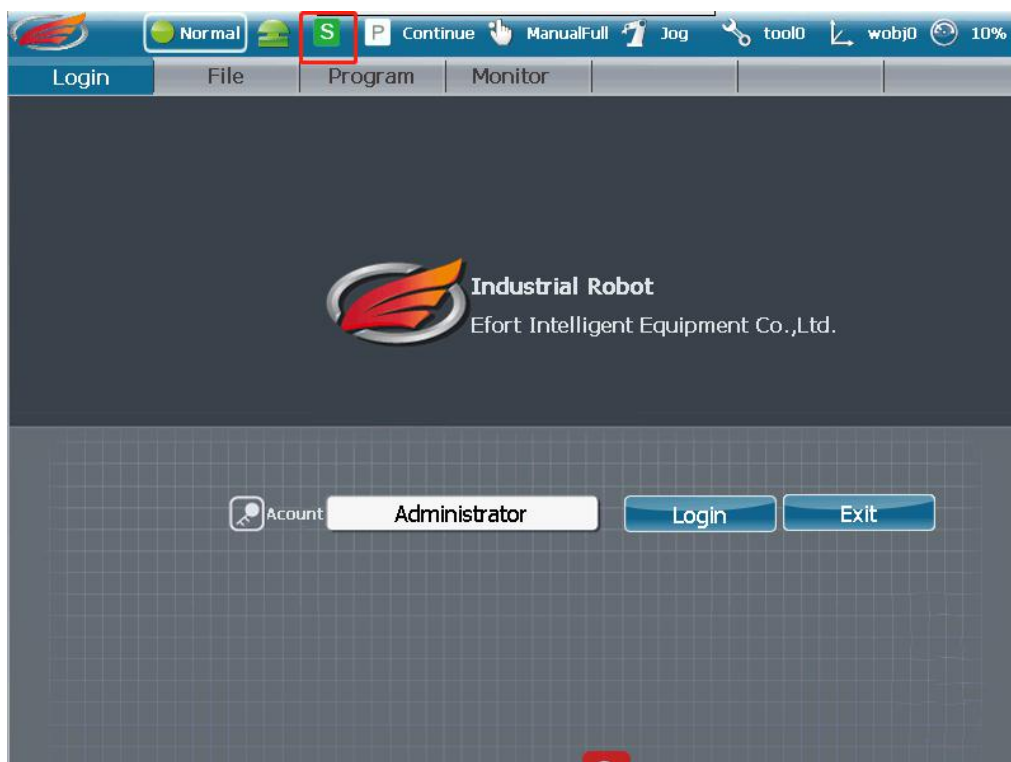
物理地址	类型	示教器对应变量
机器人发送数据		
40035	l16	fidbus.mobtxbool[0] ~ fidbus.mobtxbool[15] bit0: fidbus.mobtxbool[0] ... bit15: fidbus.mobtxbool[15]

40036	l16	fidbus.mobtxbool[16] ~ fidbus.mobtxbool[31] bit0: fidbus.mobtxbool[16] ... bit15: fidbus.mobtxbool[31]
40037	l16	fidbus.mobtxbool[32] ~ fidbus.mobtxbool[47] bit0: fidbus.mobtxbool[32] ... bit15: fidbus.mobtxbool[47]
40038	l16	fidbus.mobtxbool[48] ~ fidbus.mobtxbool[63] bit0: fidbus.mobtxbool[48] ... bit15: fidbus.mobtxbool[63]
40039 ~ 40044	l16	fidbus.mobtxint[0] ~ fidbus.mobtxint[5]
40045 ~ 40091	float	fidbus.mobtxreal[0] ~ fidbus.mobtxreal[23]
机器人接收数据		
40135	l16	fidbus.mobrxbool[0] ~ fidbus.mobrxbool[15] bit0: fidbus.mobrxbool[0] ... bit15: fidbus.mobrxbool[15]
40136	l16	fidbus.mobrxbool[16] ~ fidbus.mobrxbool[31] bit0: fidbus.mobrxbool[16] ... bit15: fidbus.mobrxbool[31]
40137	l16	fidbus.mobrxbool[32] ~ fidbus.mobrxbool[47] bit0: fidbus.mobrxbool[32] ... bit15: fidbus.mobrxbool[47]
40138	l16	fidbus.mobrxbool[48] ~ fidbus.mobrxbool[63] bit0: fidbus.mobrxbool[48] ... bit15: fidbus.mobrxbool[63]
40139 ~ 40144	l16	fidbus.mobrxint[0] ~ fidbus.mobrxint[5]
40145 ~ 40191	float	fidbus.mobrxfloat[0] ~ fidbus.mobrxint[23]

3.4应用举例

A. PLC 判断机器人是否上伺服:

通过示教器, 上伺服, 查看 PLC 中第 40001 号地址的 bit3 位;



Tx = 28; Err = 0; ID = 1; F = 03; SR = 1000ms

	Alias	4x0000	Alias	4x0020
1		0000 0000 0000 1001		-20773
2				17022
3				-1092
4				17537
5				21794
6				16758
7				0
8				0
9				0
10				0
11				0
12				0
13				0
14		16694		0
15		0		0

Write Single Register Binary

Slave ID:

1

Send

Address:

1

Cancel

Value

☐
☐
☐
☐
☐
☐
☐
☐
☒
☐
☐

Result

N/A

☐ Close dialog on "Response ok"

Use Function

☒ 06: Write single register
 ☐ 16: Write multiple registers

- B. PLC 判断机器人当前是否在运行程序:
运行程序后, 查看 PLC 中第 40001 号地址的 bit6 位;



Tx = 4592: Err = 0: ID = 1: F = 03: SR = 1000ms

	Alias	4x0000	Alias	4x0020
1		0000 0000 0100 1010		-20773
2		0		17022
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14		16694		0

Write Single Register Binary

Slave ID:

1

Send

Address:

1

Cancel

Value

☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐ ☐ ☒ ☐ ☒ ☐ ☒ ☐

Result

N/A

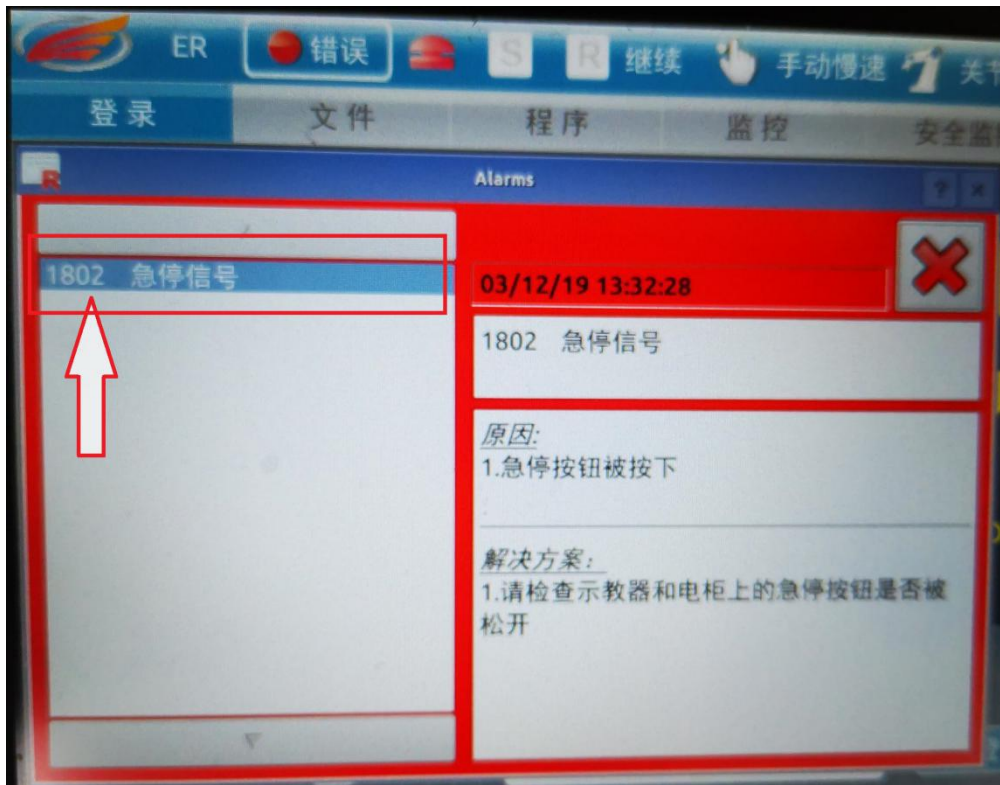
☐ Close dialog on "Response ok"

Use Function

☒ 06: Write single register
☐ 16: Write multiple registers

C. PLC 查看机器人报警代码:

报警后, 查看 PLC 中 40004 号地址的值:



	Alias	4x0000
1		0000 0000 0011 0001
2		0
3		10
4		1802
5		0

D. 查看机器人 1 轴位置:

注意位置值为 float 类型，占用 PLC 端两个物理地址，数据保存方式为 float CD AB 模式，查看 40011 和 4012 地址的值：

	Alias	4x0000	Alias	4x0020	Alias	4x0040	Alias
1		0000 0000 0011 0001		-85.49		0	
2		0		--		0	
3		10		0		0	
4		1802		0		0	
5		0		0		0	
6		0		0		--	
7		0		0		0	
8		0		0		--	
9		0		0		0	
10		0		0		--	
11		-98.0624		0		0	
12		--		0		--	
13		-3.22998					
14		--					
15		22.9642					
16		--					
17		0.0199402					
18		--					
19		-76.0699					
20		--					

Format	>	Signed	Alt+Shift+S
Read/write Definition...	F8	Unsigned	Alt+Shift+U
Cut	Ctrl+X	Hex	Alt+Shift+H
Copy	Ctrl+C	Binary	Alt+Shift+B
Paste	Ctrl+V	Long AB CD	
Select All	Ctrl+A	Long CD AB	
Foreground Color	Ctrl+F	Long BA DC	
Background Color	Ctrl+B	Long DC BA	
Font		Float AB CD	

- E. 当机器人按下'stop'键停止运行后，在 PLC 端运行机器人：
 在 40101 号地址的 bit1 写入 1(写入之前需该值必须是 0，上升沿有效)；



Tx = 1180: Err = 0: ID = 1: F = 03: SR = 1000ms

	Alias	4x0100	Alias	4x0120
1		0000 0000 0000 0000		0
2		0		0
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Write Single Register Binary

Slave ID: 1

Address: 101

Value

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☒

Result

N/A

☐ Close dialog on "Response ok"

Use Function

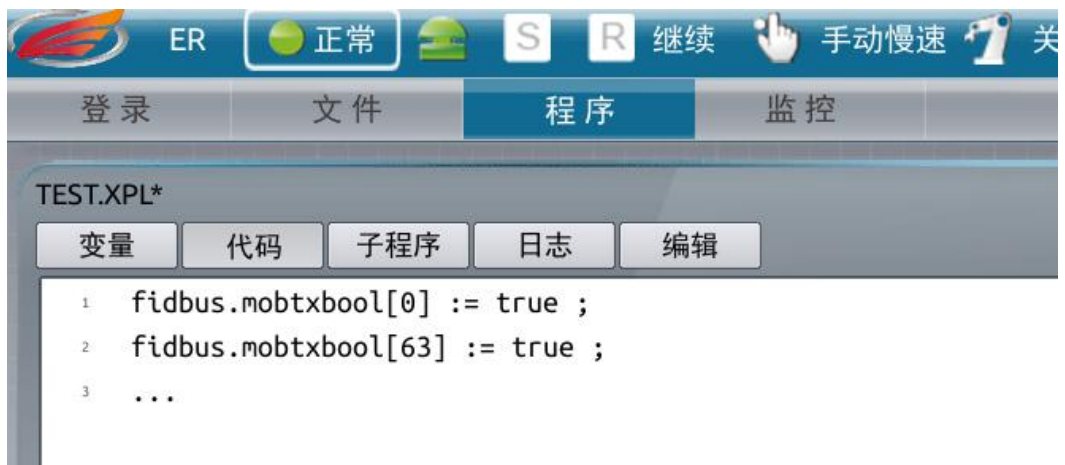
☒ 06: Write single register

Send

Cancel

F. 将示教器程序中的布尔变量传送到 PLC 中：

如将一个布尔值发送到 40035 地址中的 bit0,40038 地址中的 bit15。



	Alias	4x0000	Alias	4x0020
1		0000 0000 0000 0001		109.799
2		0		--
3		20		0
4		0		0
5		0		0
6		0		0
7		0		0
8		0		0
9		0		0
10		0		0
11		-69.2		0
12		--		0
13		-5.1		0
14		--		0
15		-36.7992		1
16		--		0
17		58.0991		0
18		--		32768
19		-32.999		0
20		--		0

G. 将 PLC 中的数据传送到示教器程序布尔变量中：

如将 PLC 中 40135 号地址中 bit3 值传送到示教器程序变量中：

Tx = 222; Err = 0; ID = 1; F = 03; SR = 1000ms

	Alias	4x0100	Alias	4x0120	Alias	4x0140	Ali
1		0000 0000 0000 0000		0		0	
2		0		--		0	
3		0		0		0	
4		0		--		0	
5		0		0		0	
6		0		--		--	
7		0		0		0	
8		0		--		--	
9		0		0		0	
10		0		--		--	
11		0		0		0	
12		--		--		--	
13		0		0		0	
14		--		--		--	
15		0		0000 0000 000...			
16		--		0			
17		0		0			
18		--		0			
19		0		0			
20		--		0			

Write Single Register Binary

Slave ID: 1

Address: 135

Value: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐ ☐

Result
Response ok
☐ Close dialog on "Response ok"

Use Function
☒ 06: Write single register
☐ 16: Write multiple registers

Send Cancel



H. 将示教器程序中的整型变量传送到 PLC 中:

如将一个整型发送到 40039 和 40044 地址上;



	Alias	4x0000	Alias	4x0020	Alias	4x0040
1		0000 0000 0000 0001		109.799		0
2		0		--		0
3		20		0		0
4		0		0		65535
5		0		0		0
6		0		0		--
7		0		0		0
8		0		0		--
9		0		0		0
10		0		0		--
11		-69.2		0		0
12		--		0		--
13		-5.1		0		0
14		--		0		--
15		-36.7992		1		0
16		--		0		--
17		58.0991		0		0
18		--		32768		--
19		-32.999		20		0
20		--		0		--

I. 将 PLC 中的整型数据传送到示教器程序变量中:

如将 40126 地址的值发送到示教器程序中;

	Alias	4x0100	Alias	4x0120	Alias	4x0140
1		0000 0000 0000 0000		0		0
2		0		--		0
3		0		0		0
4		0		--		0
5		0		0		0
6		0		--		--
7		0		0		0
8		0		--		--
9		0		0		0
10		0		--		
11		0		0		
12		--		--		
13		0		0		
14		--		--		
15		0		0000 0000 000...		
16		--		0		
17		0		0		
18		--		1000 0000 000...		
19		0		65535		
20		--		0		

Write Single Register

Slave ID:

Address:

Value:

Result: Response ok

☐ Close dialog on "Response ok"

Use Function:
☒ 06: Write single register
☐ 16: Write multiple registers



- J. 将示教器程序中的浮点型变量传送到 PLC 中：
 如将一个浮点型变量发送到 40045 和 40091 地址上；



	Alias	4x0000	Alias	4x0020	Alias	4x0040	Alias	4x0060	Alias	4x0080
1		0000 0000 0000 0001		109.799		0		0		0
2		0		--		0		--		--
3		20		0		0		0		0
4		0		0		65535		--		--
5		0		0		3.14159		0		0
6		0		0		--		--		--
7		0		0		0		0		0
8		0		0		--		--		--
9		0		0		0		0		0
10		0		0		--		--		--
11		-69.2		0		0		0		3.14159
12		--		0		--		--		--
13		-5.1		0		0		0		0
14		--		0		--		--		0
15		-36.7992		1		0		0		0
16		--		0		--		--		0
17		58.0991		0		0		0		0
18		--		32768		--		--		0
19		-32.999		20		0		0		0
20		--		0		--		--		0

K. 将 PLC 中的浮点型数据传送到示教器程序变量中：

如将一个浮点型数据发送到 40145 地址上

	Alias	4x0100	Alias	4x0120	Alias	4x0140	Alias	4x0160
1		0000 0000 0000 0000		0		0		0
2		0		--		0		--
3		0		0		0		0
4		0		--		0		--
5		0		0		1.34159		0
6		0		--		--		--
7		0		0		0		0
8		0		--		--		--
9		0		0		0		0
10		0		--		--		--
11		0		0		0		0
12		--		--		--		--
13		0		0		0		0
14		--		--		--		--
15		0		0000 0000 000...		0		0
16		--		0		--		--
17		0		0		0		0
18		--		0000 0000 000...		--		--
19		0		0		0		0
20		--		0		--		--

Write Float

Slave ID: 1 Send

Address: 145 Cancel

Value: 1.34159

☐ Result
☐ Response ok
☐ Close dialog on "Response ok"

ER

正常

S

R 继续

手动慢速

关节

tool0

wobj0

20%

登录

文件

程序

监控

TEST.XPL

robot

变量

代码

子程序

日志

Main

1 rxfloat := 3.15 ;

2 rxfloat := fidbus.mobrxreal[0] ;

在线监控

名称	值
rxfloat	1.34159004688263
