

常问问题 • 06/2014

TP900 Comfort 与 Allen-Bradley Control Logix5500 通讯

Comfort Panels

目录

1 概述	3
2 测试软件及硬件	3
3 组态	4
3.1 PLC 硬件组态	4
3.2 PLC 的控制器变量和程序变量	5
3.3 TP900 Comfort 读取控制器变量和程序变量	7
3.3.1 创建连接	7
3.3.2 创建变量	9
3.3.3 运行结果	11
3.4 读取 PLC 整数变量的某一位的值	11
3.5 字符型变量	12
3.6 数组变量	14
3.7 自定义数据类型	15

1 概述

西门子精智面板（ Comfort Panels）可以使用“ Allen-Bradley EtherNet/IP” 驱动程序通过以太网与 Allen-Bradley 的 PLC 通讯：

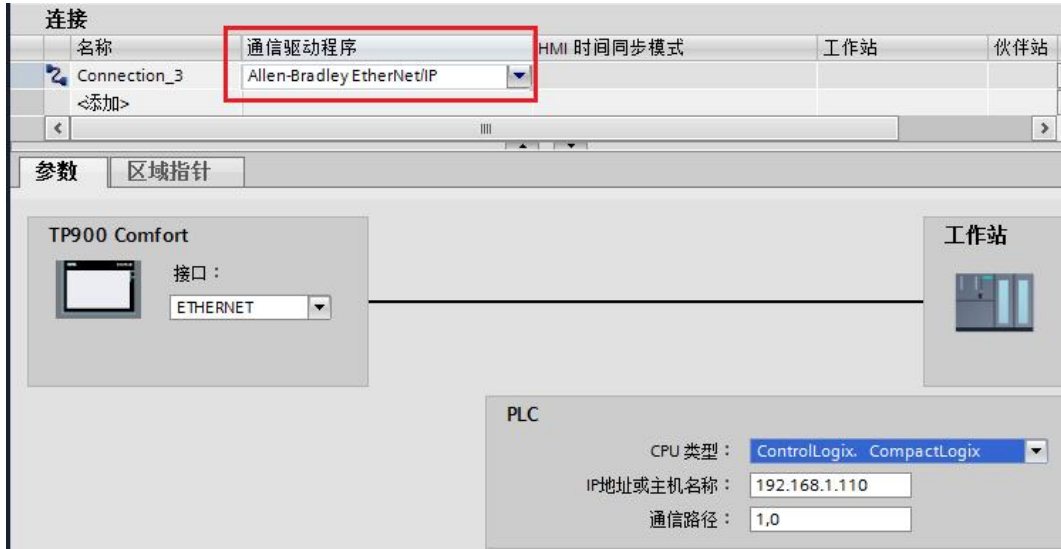


图 1

此驱动程序支持的 PLC 包括 ControlLogix5500、CompactLogix5300 系列 PLC、SLC500 系列 PLC、MicroLogix 系列 PLC。

下面以 TP900 Comfort 和 ControlLogix5563 为例介绍一下组态过程。

2 测试软件及硬件



图 2

2.1 硬件

ControlLogix5563 带 4 个 IO 模块（DI、DO、AI、AO）、1 个 ControlNet 网络模块、1 个以太网模块。

西门子精智面板：TP900 Comfort（订货号：6AV2124-0JC01-0AX0）。

2.2 软件

操作系统：Win7 SP1；

PLC 编程软件：RSLogix5000 Professional V19.01.00；

RSLinx：RSLinx V2.57；

TP900 Comfort 组态软件：WinCC Professional V12 SP1（使用 WinCC Comfort 以上版本）。

3 组态

3.1 PLC 硬件组态

按照实际设备情况，组态如下：

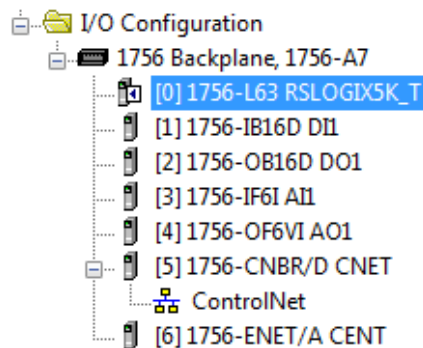


图 3

其中，CPU 版本为 19:

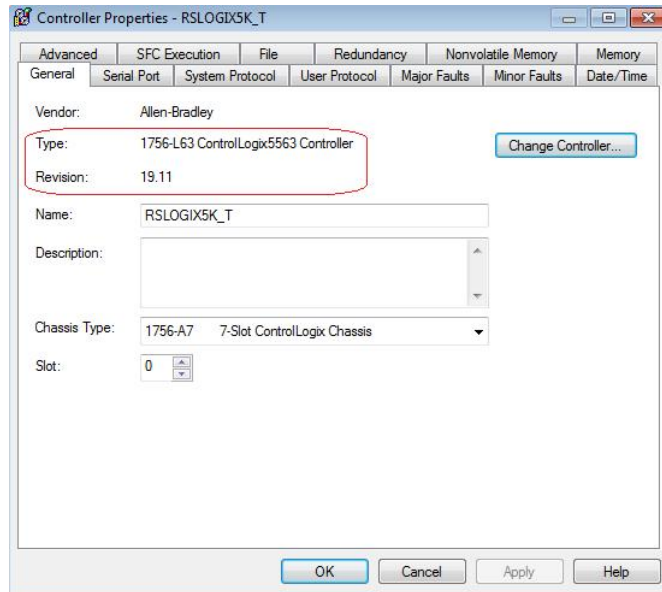


图 4

以太网模块地址 192.168.1.110:

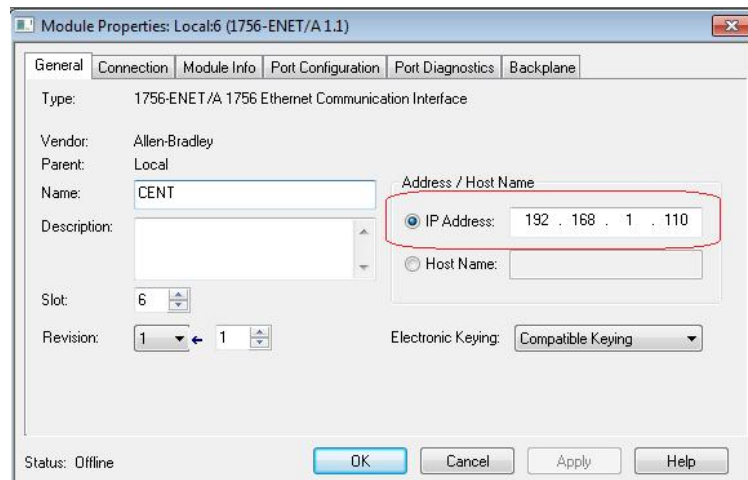


图 5

3.2 PLC 的控制器变量和程序变量

PLC 程序结构如下:

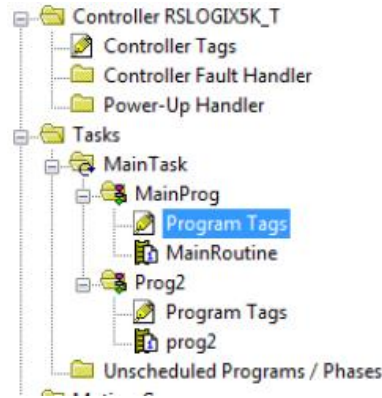


图 6

在“ Controller Tags” 下创建控制器变量，控制器变量可以在任何一个程序中使用。

Scope: **RSLOGIX5K_T** Show: All Tags

Name	Value	Force Mask	Style	Data Type
+ DIM1	{...}	{...}	Decimal	DINT[2]
+ DIM2	{...}	{...}	Decimal	DINT[2,2]
+ DIM3	{...}	{...}	Decimal	DINT[2,2,2]
+ Local:1:C	{...}	{...}		AB:1756_DI_DC_Diag:C:0
+ Local:1:I	{...}	{...}		AB:1756_DI_DC_Diag:I:0
+ Local:2:C	{...}	{...}		AB:1756_DO_DC_Diag:C:0
+ Local:2:I	{...}	{...}		AB:1756_DO_DC_Diag:I:0
+ Local:2:O	{...}	{...}		AB:1756_DO:0:0
+ Local:3:C	{...}	{...}		AB:1756_AI6_Float:C:0
+ Local:3:I	{...}	{...}		AB:1756_AI6_Float:I:0
+ Local:4:C	{...}	{...}		AB:1756_AO6_Float:C:0
+ Local:4:I	{...}	{...}		AB:1756_AO6_Float:I:0
+ Local:4:O	{...}	{...}		AB:1756_AO6_Float:O:0
+ my_dt1	{...}	{...}		DT1
+ str1	''	{...}		STRING
+ str2	''	{...}		mystring
+ tag_di1	0		Decimal	BOOL
+ tag_dint1	0		Decimal	DINT

图 7

在 MainProg 和 Prog2 下的“ Program Tags” 中分别创建各自的程序变量，程序变量只能在各自的程序中使用。

Scope: **MainProg** Show: All Tags

Name	Value	Force Mask	Style	Data Type
prog_di1	0		Decimal	BOOL
+ prog_dint1	0		Decimal	DINT

图 8

Scope: Prog2		Show: All Tags				
Name	Value	Force Mask	Style	Data Type		
prog_di1	0		Decimal	BOOL		
+ prog_dint1	0		Decimal	DINT		

图 9

3.3 TP900 Comfort 读取控制器变量和程序变量

3.3.1 创建连接

在 WinCC Professional V12 中创建 TP900 Comfort 项目，并创建“ Allen-Bradley EtherNet/IP”连接：

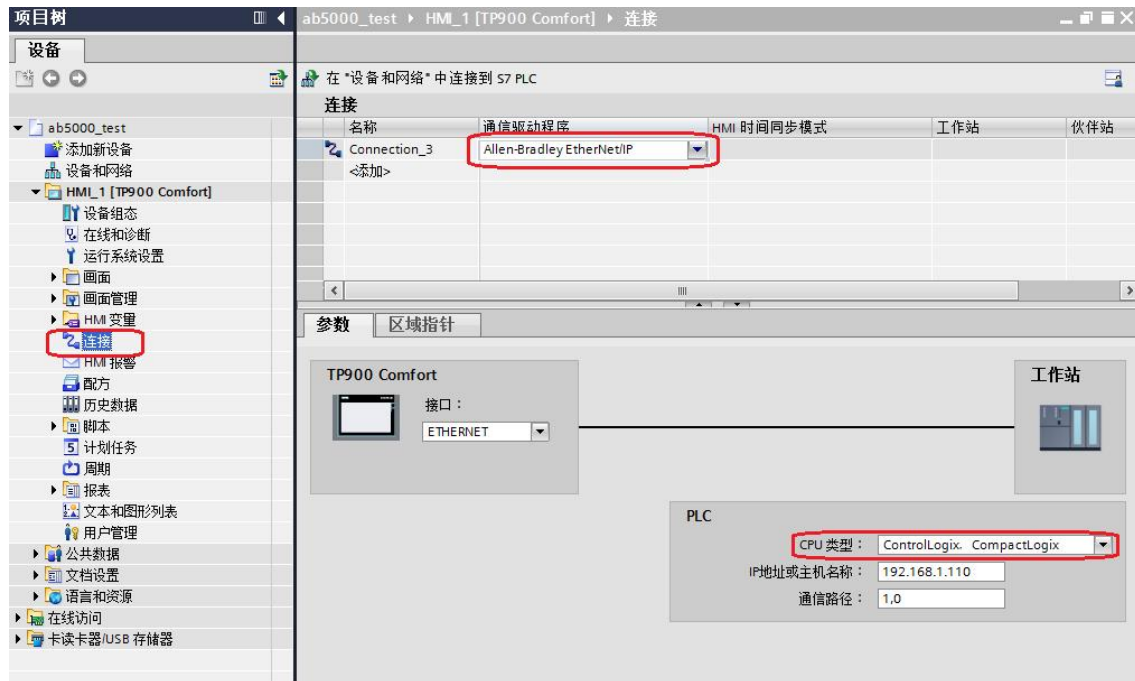


图 10

其中，CPU 类型选择“ ControlLogix、CompactLogix”。

IP 地址输入 PLC 以太网模块的地址。

通讯路径中，1 代表背板连接，0 代表 CPU 所在机架中插槽。

关于“ 通讯路径” 的说明：

- 1、 连接本地机架上的 CPU（CPU 和以太网模块在同一机架），通讯路径为“ 1， x”， x 代表 CPU 所在插槽，和 IO 组态对应即可。

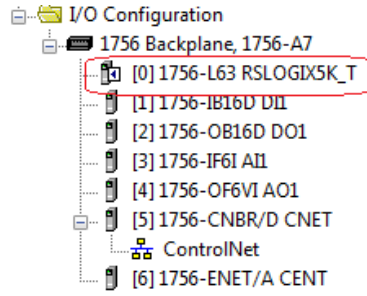


图 11

- 2、 连接**远程机架**上的 CPU（CPU 和以太网模块不在同一机架，但本地机架和远程机架通过网络相连），以如下系统为例：

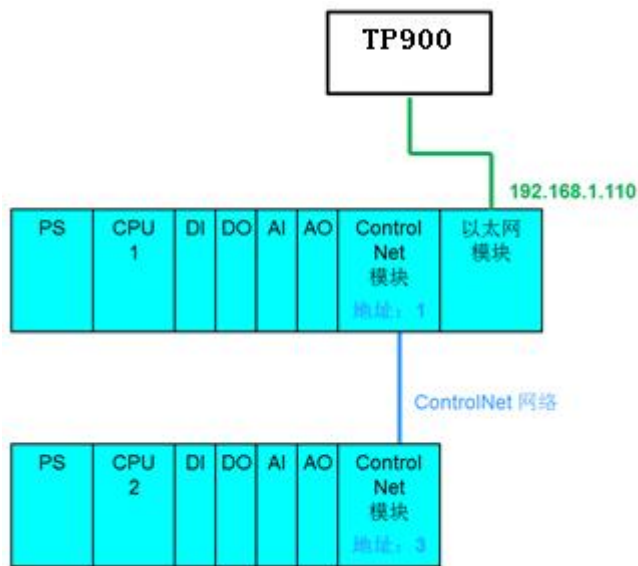


图 12

TP900 Comfort 访问远程机架上的 CPU2 的通讯路径写法：



图 13

其中：

1：背板连接

5：本地机架上网络模块插槽（例子 controlnet 模块位于 5 槽）

- 2: 网络连接
- 3: 远程模块网络地址（远程 controlnet 模块地址为 3）
- 1: 远程机架的背板连接
- 0: 远程机架 CPU 槽号

这样 TP900 Comfort 就可以通过路由的方式访问远程机架上的 CPU2:

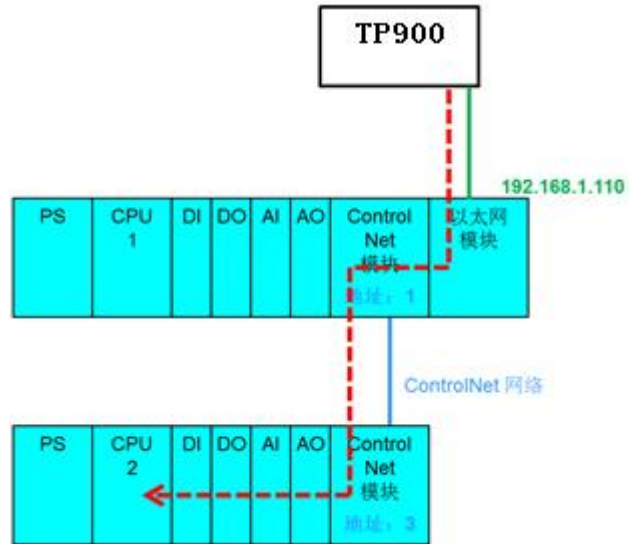


图 14

3.3.2 创建变量

在 WinCC Professional 连接下创建变量。

ab5000_test ▶ HM_1 [TP900 Comfort] ▶ HM 变量 ▶ Default tag table [16]

Default tag table

名称	数据类型	连接	地址
DIM1_1	Dint	Connection_3	DIM1[1]
DIM2_1	Dint	Connection_3	DIM2[1,1]
DIM3_1	Dint	Connection_3	DIM3[1,1,1]
mydt_1	Dint	Connection_3	my_dt1.NUM
mydt_2	String	Connection_3	my_dt1.STR
plc_di1	Bool	Connection_3	tag_di1
plc_dint1	Dint	Connection_3	tag_dint1
plc_dint1_0	Bool	Connection_3	tag_dint1/0
plc_str1	String	Connection_3	str1
plc_str2	String	Connection_3	str2.DATA
prog1_di1	Bool	Connection_3	MainProg:prog_di1
prog1_dint1	Dint	Connection_3	MainProg:prog_dint1
prog2_di1	Bool	Connection_3	Prog2:prog_di1
prog2_dint1	Dint	Connection_3	Prog2:prog_dint1
REMOT_AAA	Dint	Connection_3	aaa[0]

图 15

数据类型：要和 PLC 中定义的数据类型对应。

变量地址：

控制器变量地址直接写 PLC 中定义的变量名称；

程序变量地址格式为“程序名：变量名”，例如，PLC 中 MainProg 的变量
“ prog_dint1 ”：

Scope: MainProg Show: All Tags

Name	Value	Force Mask	Style	Data Type
prog_di1	0		Decimal	BOOL
prog_dint1	0		Decimal	DINT

图 16

WinCC Professional 中对应的地址为“ MainProg: prog_dint1 ”。

3.3.3 运行结果

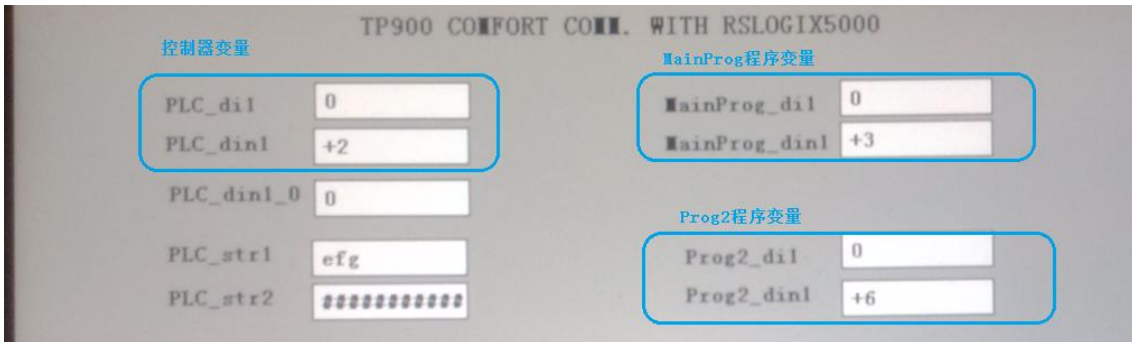


图 17

监视 PLC 中的变量：

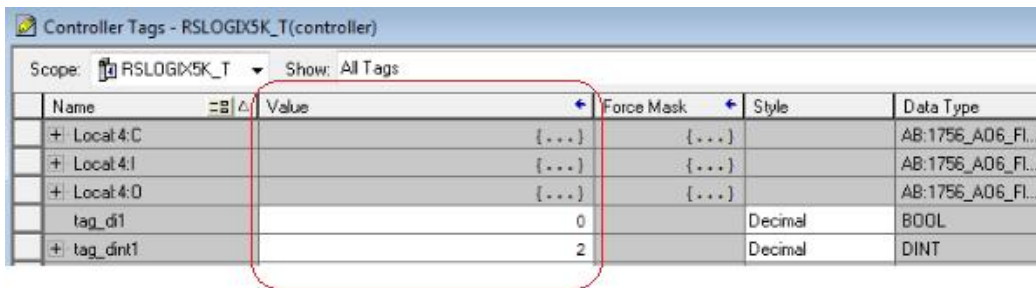


图 18



图 19

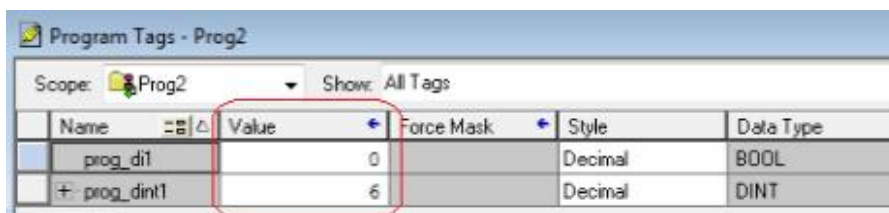


图 20

3.4 读取 PLC 整数变量的某一位的值

可以在 WinCC Professional 中创建 Bool 型变量读取整数变量中的某一位的值，例如可以创建如下变量读取控制器变量“ tag_dint1 ”的第 0 位数值：

名称	数据类型	连接	地址
plc_dint1_0	Bool	Connection_3	tag_dint1/0

图 21

Scope: RSLOGIX5K_T Show: All Tags

Name	Value	Force Mask	Style	Data Type
+ Local3:I	{...}	{...}		AB:1756_AI6_Float:I:0
+ Local4:C	{...}	{...}		AB:1756_AO6_Float:C:0
+ Local4:I	{...}	{...}		AB:1756_AO6_Float:I:0
+ Local4:O	{...}	{...}		AB:1756_AO6_Float:O:0
+ my_dt1	{...}	{...}		DT1
+ str1	''	{...}		STRING
+ str2	''	{...}		mystring
tag_dt1	0		Decimal	BOOL
- tag_dint1	0		Decimal	DINT
tag_dint1.0	0		Decimal	BOOL
tag_dint1.1	0		Decimal	BOOL
tag_dint1.2	0		Decimal	BOOL
tag_dint1.3	0		Decimal	BOOL

图 22

3.5 字符型变量

3.5.1 “STRING” 变量

PLC 中定义“ STRING” 类型变量（长度 82）：

- str1	''	{...}		STRING
+ str1.LEN	0		Decimal	DINT
+ str1.DATA	{...}	{...}	ASCII	SINT[82]

图 23

WinCC Professional 创建对应的变量（地址为“ str1”）：

名称	数据类型	连接	地址
plc_str1	String	Connection_3	str1

图 24

3.5.2 自定义长度字符串变量：

在 RSLogix5000 中创建自定义长度字符串变量类型：

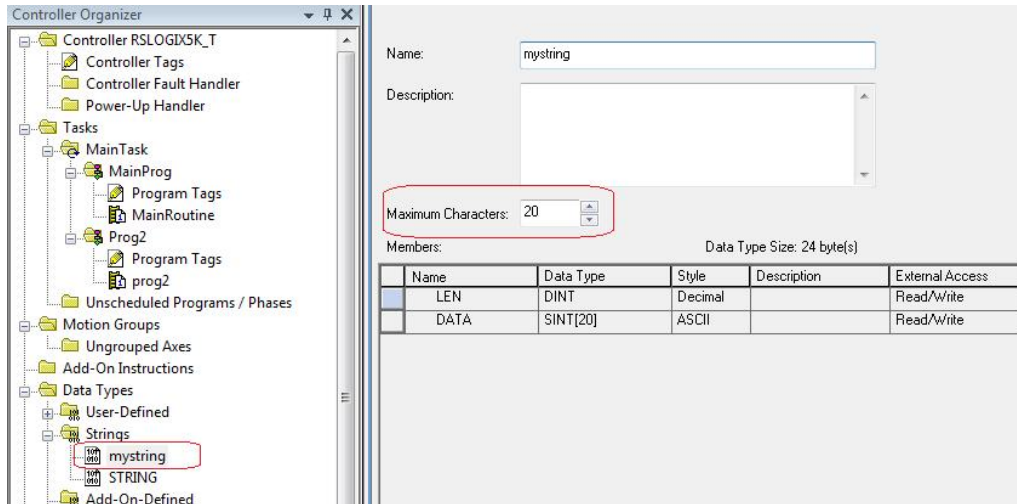


图 25

并在 PLC 中创建变量：

- str2	''	{...}		mystring
+ str2.LEN	0		Decimal	DINT
+ str2.DATA	{...}	{...}	ASCII	SINT[20]

图 26

在 WinCC Professional 中创建对应的变量：

名称	数据类型	连接	地址
plc_str2	String	Connection_3	str2.DATA

图 27

3.5.3 TP900 Comfort 不支持自定义长度的字符串变量

运行效果：

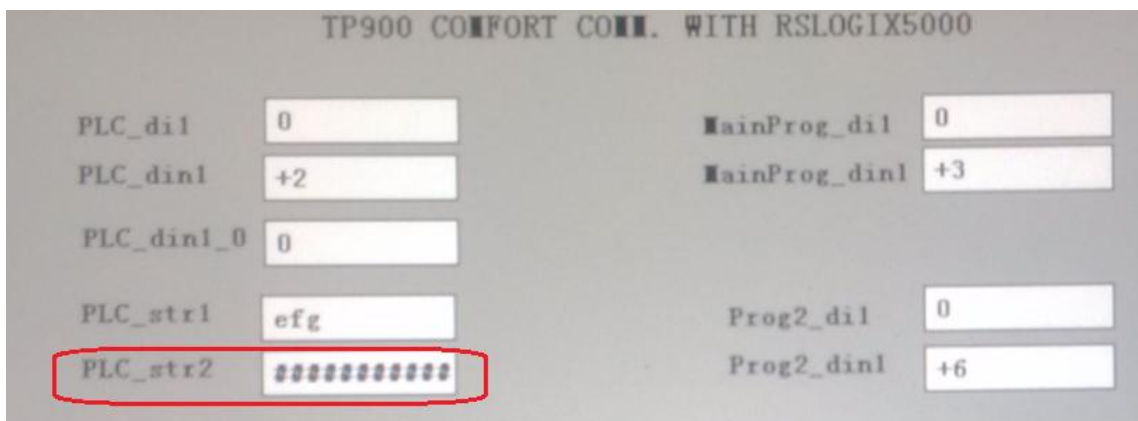


图 28

PLC 变量监视:

Name	Value	Force Mask	Style	Data Type
Local 4:C	{...}	{...}		AB:1756_A06_FL...
Local 4:I	{...}	{...}		AB:1756_A06_FL...
Local 4:O	{...}	{...}		AB:1756_A06_FL...
tag_d1	0		Decimal	BOOL
tag_dint1	2		Decimal	DINT
str1	'efg000000000000000'	{...}		STRING
str1.LEN	10		Decimal	DINT
str1.DATA	{...}	{...}	ASCII	SINT[82]
str2	''	{...}		mystring
str2.LEN	0		Decimal	DINT
str2.DATA	{...}	{...}	ASCII	SINT[20]

图 29

可以看到，TP900 Comfort 可以正常读取 PLC 中“ STRING”类型的变量，但 TP900 Comfort 不能读取 PLC 中自定义长度的字符串变量。

3.6 数组变量

在 PLC 中定义一维、二维、三维变量:

Name	Value	Force Mask	Style	Data Type
Local 4:C	{...}	{...}		AB:1756_A06_FL...
Local 4:I	{...}	{...}		AB:1756_A06_FL...
Local 4:O	{...}	{...}		AB:1756_A06_FL...
tag_d1	0		Decimal	BOOL
tag_dint1	2		Decimal	DINT
str1	'efg000000000000000'	{...}		STRING
str1.LEN	10		Decimal	DINT
str1.DATA	{...}	{...}	ASCII	SINT[82]
str2	''	{...}		mystring
str2.LEN	0		Decimal	DINT
str2.DATA	{...}	{...}	ASCII	SINT[20]
DIM1	{...}	{...}	Decimal	DINT[2]
DIM1[0]	0		Decimal	DINT
DIM1[1]	1		Decimal	DINT
DIM2	{...}	{...}	Decimal	DINT[2,2]
DIM2[0,0]	0		Decimal	DINT
DIM2[0,1]	0		Decimal	DINT
DIM2[1,0]	0		Decimal	DINT
DIM2[1,1]	2		Decimal	DINT
DIM3	{...}	{...}	Decimal	DINT[2,2,2]
DIM3[0,0,0]	0		Decimal	DINT
DIM3[0,0,1]	0		Decimal	DINT
DIM3[0,1,0]	0		Decimal	DINT
DIM3[0,1,1]	0		Decimal	DINT
DIM3[1,0,0]	0		Decimal	DINT
DIM3[1,0,1]	0		Decimal	DINT
DIM3[1,1,0]	0		Decimal	DINT
DIM3[1,1,1]	3		Decimal	DINT

图 30

WinCC Professional 变量对应的地址写法:

Default tag table			
名称 ▲	数据类型	连接	地址
DIM1_1	DInt	Connection_3	DIM1[1]
DIM2_1	DInt	Connection_3	DIM2[1,1]
DIM3_1	DInt	Connection_3	DIM3[1,1,1]

图 31

3.7 自定义数据类型

PLC 中创建自定义数据类型:

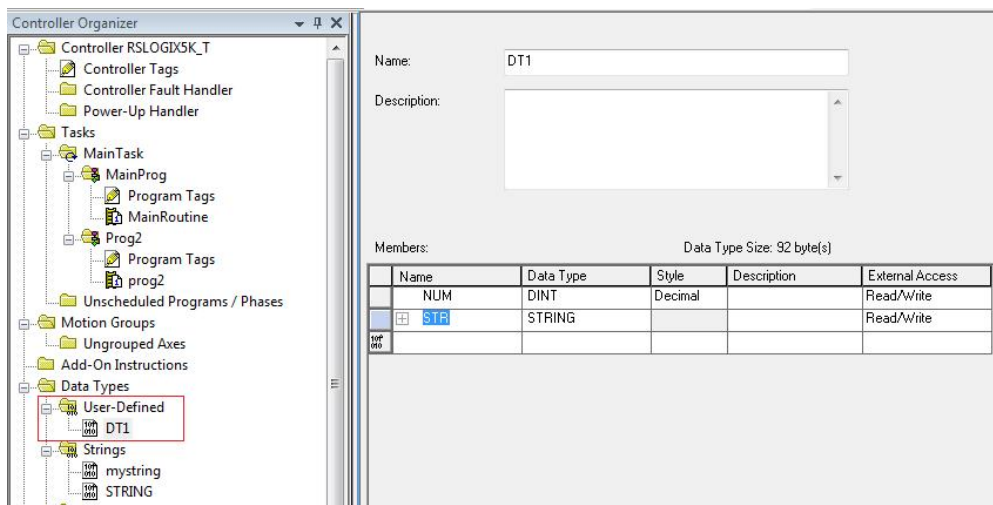


图 32

并创建 PLC 变量:

- my_dt1	{...}	{...}	DT1
+ my_dt1.NUM	0	Decimal	DINT
+ my_dt1.STR	' '	{...}	STRING

图 33

WinCC Professional 对应变量的地址:

名称 ▲	数据类型	连接	地址
mydt_1	DInt	Connection_3	my_dt1.NUM
mydt_2	String	Connection_3	my_dt1.STR

图 34

运行结果:

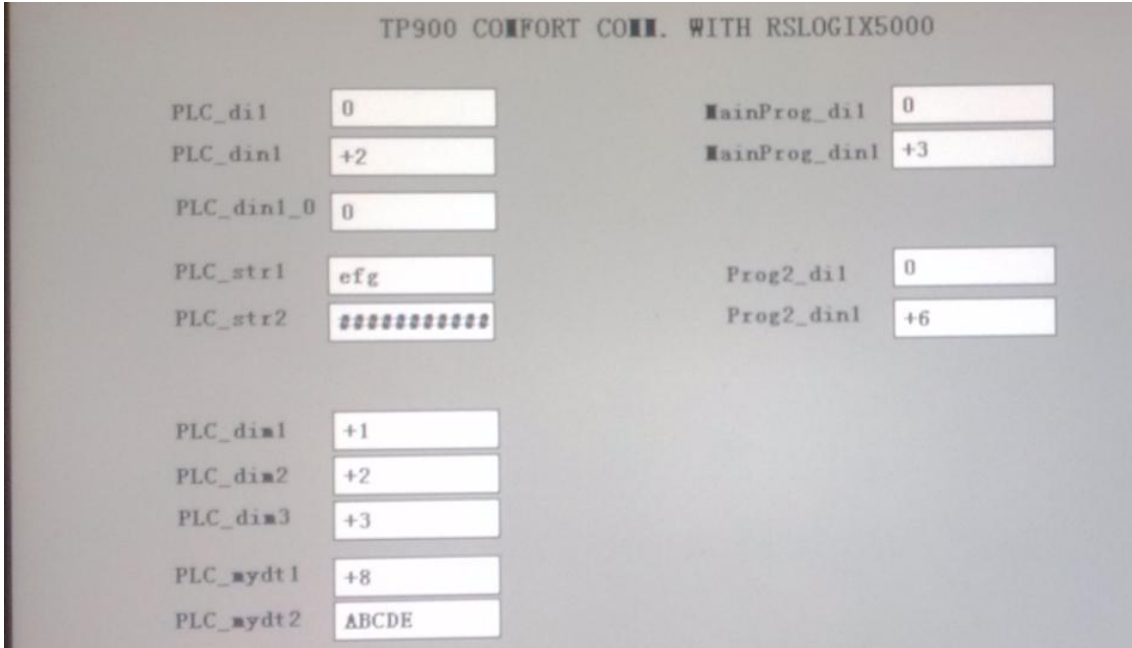


图 35