

FC6A 型

MICROSmart

通信手册



安全上的重要注意事项

- 在安装、接线、操作、维护和检查 FC6A 型 MICROSmart 前，请仔细阅读本《FC6A 型 MICROSmart 通信手册》以确保操作正确。
- 所有 FC6A 型 MICROSmart 都是在 IDEC 严格的质量管理系统下制造的，但如果是在使用过程中万一发生因为 FC6A 型 MICROSmart 的故障而可能导致的重大事故或者损害时，请用户务必在控制系统中做好备份或故障保护准备。
- 对于 FC6A 型 MICROSmart 型外部设备的未经授权访问等，请在网络系统侧采取措施。对于未经授权的访问等直接或间接造成的损失、损害或其他费用，我们不承担任何责任，敬请谅解。
- 在本手册中，安全事项按其重要性的顺序加以分类。



警告 警告提示用于强调操作不当会导致严重的人身伤亡。

- FC6A 型 MICROSmart 不适用于对可靠性和安全性要求较高的用途。请勿在这类用途中使用。
- 如果在非上述应用（对精确度和功能的稳定性要求较高）中使用 FC6A 型 MICROSmart，则必须对包含 FC6A 型 MICROSmart 的系统采取适当措施，如故障保护机制、冗余机制等。以下介绍了具体示例。
 - 必须在 FC6A 型 MICROSmart 的外部设置紧急停止和联锁电路。
 - 如果 FC6A 型 MICROSmart 输出电路中的继电器或晶体管发生故障，则输出会始终保持开启或关闭状态。为避免因输出信号而引起的严重事故，请在 FC6A 型 MICROSmart 外设置监控电路。
 - FC6A 型 MICROSmart 自我诊断功能可检测内部电路或程序错误，在检测到错误后会中止程序并关闭输出。为避免在关闭输出时损坏包含 FC6A 型 MICROSmart 的系统，请设置回路。
- 在安装、拆卸、接线、维护以及检查 FC6A 型 MICROSmart 前，请务必关闭 FC6A 型 MICROSmart 的电源。如果不关闭电源，可能导致破损、触电或火灾危险。
- 需要采用特殊的专门技术来安装、接线、编程和操作 FC6A 型 MICROSmart。没有这些专门技术的人员不得使用 FC6A 型 MICROSmart。
- 请按本用户手册所描述的操作步骤安装 FC6A 型 MICROSmart。安装不当将导致 FC6A 型 MICROSmart 脱落、故障或指令错误。



注意 在疏忽会导致人身伤害或设备损坏的地方会有注意提示。

- FC6A 型 MICROSmart 是为安装在机柜中设计的。请勿将 FC6A 型 MICROSmart 安装在机柜的外部。
- 请在《FC6A 型 MICROSmart 用户手册》所描述的环境中安装 FC6A 型 MICROSmart。如果在使用 FC6A 型 MICROSmart 时，FC6A 型 MICROSmart 周围的环境为高温、高湿度、有结露或腐蚀性气体，且摇摆和震动剧烈，如果在高温、高湿度、有结露、有腐蚀性气体、剧烈摇摆或剧烈震动的场所使用 FC6A 型 MICROSmart，则会导致触电、火灾或故障的发生。则会导致触电、火灾或故障发生。
- 使用 FC6A 型 MICROSmart 的环境是“污染等级 2”。请在污染等级为 2（按照 IEC 60664-1）的环境中使用 FC6A 型 MICROSmart。
- 要避免在移动和运输 FC6A 型 MICROSmart 的过程中将 FC6A 型 MICROSmart 跌落，否则会造成 FC6A 型 MICROSmart 损坏或出现故障。
- 接线用导线的尺寸必须适用于所采用的电压和电流。必须按照规定的紧固力矩将接线螺钉拧紧。
- 防止金属碎片和电缆片段落入 FC6A 型 MICROSmart 机架内部。安装和接线时，请在 FC6A 型 MICROSmart 上盖上面罩。若有碎屑进入，可能会导致火灾、损坏或故障。
- 使用额定值的电源。电源使用不当会导致火灾。
- 在 FC6A 型 MICROSmart 外部的电源线上使用符合 IEC 60127 的保险丝。这是销往欧洲的装有 FC6A 型 MICROSmart 的设备所必需的。
- 请在输出电路上使用经 IEC60127 认可的保险丝。这是销往欧洲的装有 FC6A 型 MICROSmart 的设备所必需的。
- 使用经欧盟认可的断路器。这是销往欧洲的装有 FC6A 型 MICROSmart 的设备所必需的。
- 在启动和停止 FC6A 型 MICROSmart 前，或操作 FC6A 型 MICROSmart 强行打开或关闭输出时，请确保安全。FC6A 型 MICROSmart 操作不当会导致机器损坏或意外事故。
- 请勿将地线与 FC6A 型 MICROSmart 直接连接。请使用 M4 或更大的螺钉为装有 FC6A 型 MICROSmart 的机柜提供保护性地。这是销往欧洲的装有 FC6A 型 MICROSmart 的设备所必需的。
- 请勿擅自分解、修理或改装 FC6A 型 MICROSmart。
- FC6A 型 MICROSmart 中包含电子元件和电池。因此，丢弃废弃的 FC6A 型 MICROSmart 产品时，请按照国家及当地的相关法规来进行处理。

关于本手册

本手册介绍 FC6A 型 MICROSmart 的各种通信功能。

第 1 章：前言

关于带有通信接口的 FC6A 型 MICROSmart 的常规信息。

第 2 章：设备地址

基本指令或高级指令中使用的输入输出、内部继电器、寄存器、定时器和计数器等各种设备的分配描述，以及用于通信功能的特殊内部继电器和特殊数据寄存器的分配详情。

第 3 章：通信设置

FC6A 型 MICROSmart 通信功能、如何配置这些功能及其使用示例。

第 4 章～第 8 章：

各种通信功能，例如维护通信、用户通信、Modbus 通信、数据连接通信和 J1939 通信。

第 9 章：Bluetooth 通信

关于 FC6A 型 MICROSmart 的 Bluetooth 通信以及将 FC6A 型 MICROSmart 与其他 Bluetooth 通信设备连接的方法。

第 10 章：FTP 服务器 / 客户端功能

可向 Plus CPU 模块中插入的 SD 卡写入或导出文件的 FTP 服务器功能以及可在 Plus CPU 模块中插入的 SD 卡和 FTP 服务器之间复制或移动文件的 FTP 客户端功能的描述。

第 11 章：PING 发送功能

用于向指定的远程主机发送 PING 数据包以确认是否可在网际协议（IP）层上进行通信的 PING 指令的描述。

第 12 章：EMAIL 发送功能

用于发送事先注册的电子邮件的 EMAIL 指令的描述。

第 13 章：网络服务器

FC6A 型 MICROSmart 中的网络服务器功能的描述。

第 14 章：通信监控

可监控与 FC6A 型 MICROSmart 相连的通信设备之间通信的通信监控功能的描述。

第 15 章：BACnet/IP

FC6A 型 MICROSmart 中的 BACnet 通信（BACnet/IP）功能的描述。

第 16 章：EtherNet/IP 通信

FC6A 型 MICROSmart 中的 EtherNet 通信功能的描述。

第 17 章：MQTT 通信

FC6A 型 MICROSmart 中的 MQTT 通信功能的描述。

第 18 章：MC 协议通信

FC6A 型 MICROSmart 中的 MC 协议通信功能的描述。

索引

依据关键字首字母顺序排序。

相关手册

可结合本手册一同参考使用。

手册名称	说明
FC6A 型 MICROSmart 用户手册	介绍产品规格、安装和接线说明、基本编程操作和特殊功能说明、设备和指令表以及 FC6A 型 MICROSmart 产品的故障排除步骤。
FC6A 型 MICROSmart 通信手册（本手册）	介绍 FC6A 型 MICROSmart 的通信相关规格、功能的说明、设置方法及使用示例。
FC6A 型 MICROSmart PID 模块用户手册	介绍 PID 模块的规格和功能。
梯形图编程手册	介绍使用梯形图编程的基本操作、监控方法、设备和指令列表以及每条指令的详细信息。

我们的网站随时免费提供最新的产品手册 PDF。请从我们的网站下载最新的产品手册 PDF。

版本简历

2015年12月	第1版
2016年2月	第2版
2016年4月	第3版
2017年3月	第4版
2017年8月	第5版
2018年3月	第6版
2018年6月	第7版
2018年11月	第8版
2019年5月	第9版
2019年12月	第10版
2020年4月	第11版
2020年7月	第12版
2020年12月	第13版
2021年3月	第14版
2021年7月	第15版
2022年2月	第16版
2022年9月	第17版
2023年11月	第18版

注意

- 本手册的所有权利均归 IDEC 公司所有。未经许可不得复制、转载、出售转让或出租。
- 本手册的内容如有更改，恕不另行通知。
- 关于产品的内容力求做到全，如有疑问或错误等发现之处，请联系购买的销售店或本公司的营业所、办事处。

商标

- WindLDR 和 MICROSmart 是 IDEC 公司在日本的注册商标。
- MELSEC 是三菱电机公司的商标。
- 本文提及的其他公司和产品名称可能是其各自所有者的商标或注册商标。

有关法规以及相关规定

以下是适用于本产品的各国法规以及规定。

欧洲法规、规定

本产品适用于如下欧洲指令

- 低电压指令
- EMC 指令
- RoHS 指令
- RE 指令（仅 FC6A-PC4）

为了对应上述指令，本产品基于如下国际规定以及欧洲规定去进行设计以及测试。

- IEC/EN 61131-2:2007
- EN50581:2012
- EN301 489-1 V2.1.1& EN301 489-17 V2.1.1（仅 FC6A-PC4）

北美法规、规定

本产品已通过 UL 取得以下认证。

- UL508^{*1}
- UL61010-1^{*1}
- UL61010-2-201^{*1}
- CSA C22.2 No.142^{*1}
- CSA C22.2 No.61010-1^{*1}
- CSA C22.2 No.61010-2-201^{*1}
- ANSI/ISA 12.12.01
- CAN/CSA C22.2 No.213

*1 FC6A 型 MICROSmart 内不支持部分机型。有关适用标准的详情，请咨询本公司。

船舶规定

本产品已取得以下船级社认证。

（FC6A-C16R1DE、FC6A-C16P1DE 和 FC6A-C16K1DE 未通过认证。）

- ABS（美国船级社）
- DNV GL（DNV GL 船级社）
- LR（劳埃德船级社）
- NK（日本海事协会）

* 未取得用于船桥及甲板的认证。

适用标准及 EU 指令的详情，请咨询您所购买的经销店或查看本公司网站。

重要声明

在任何情况下，IDEC 株式会社都不对由于使用或应用 IDEC PLC 组件而间接或直接导致的损坏负责（无论是单独使用，还是与其他设备结合使用）。
所有使用这些组件的人员都要自行承担选择适用于其应用程序的组件以及选择适用于这些组件的应用程序的责任（无论是单独使用，还是与其他设备结合使用）。
本手册中的所有图表和示例仅起说明作用。这些图表和示例并不保证其适用于任何特殊应用软件。在安装前，最终用户需承担测试和认可所有程序的责任。

本手册中使用的名称及简称

项目	说明
FC6A型	CPU模块、扩展模块、增设扩展模块、HMI模块、盒基本模块、盒的总称
CPU模块	All-in-One CPU模块、CAN J1939 All-in-One CPU模块、Plus CPU模块的总称
All-in-One CPU模块	FC6A-C*****E CPU 模块的总称
16-I/O型	输入输出端子总计 16 点 All-in-One CPU 模块的的总称 (FC6A-C16****)
24-I/O型	输入输出端子总计24点的All-in-One CPU模块的总称 (FC6A-C24****)
CAN J1939 All-in-One CPU模块	FC6A-C40***EJ CPU模块的总称
Plus CPU模块	FC6A-D****CEE CPU模块的总称
Plus 16-I/O型	输入输出端子总计 16 点的 Plus CPU 模块的总称 (FC6A-D16*****)
Plus 32-I/O型	输入输出端子总计 32 点的 Plus CPU 模块的总称 (FC6A-D32*****)
40-I/O型	输入输出端子总计 40 点的 CPU 模块的总称 (FC6A-C40****)
AC电源类型	电源规格为交流电的CPU模块的总称 (FC6A-C****AE、FC6A-C****AEJ)
DC电源类型	直流24V电源型、直流12V电源型的CPU模块的总称
24V DC电源类型	这是电源规格为直流24V的CPU模块的统称 (FC6A-C****CE、FC6A-C****CEJ、FC6A-D****CEE)
12V DC电源类型	这是电源规格为直流12V的CPU模块的统称 (FC6A-C****DE、FC6A-C****DEJ)
继电器输出类型	输出为继电器输出的 CPU 模块的统称 (FC6A-C**R**E、FC6A-C**R**E*)
晶体管输出类型	晶体管沉型输出类型、晶体管保护源型输出类型的CPU模块的统称
晶体管沉型输出类型	这是其输出为晶体管沉型输出的 CPU 模块的总称 (FC6A-C**K**E、FC6A-C**K**E*、FC6A-D**K*CEE)
晶体管保护源型输出类型	这是其输出为晶体管保护源型输出的 CPU 模块的总称 (FC6A-C**P**E、FC6A-C**P**E*、FC6A-D**P*CEE)
扩展模块	I/O模块、通信模块、PID模块的总称
I/O模块	数字I/O模块、模拟I/O模块的总称
数字I/O模块	数字输入模块、数字输出模块、数字混合I/O模块的总称
数字输入模块	带输入端子的数字 I/O 模块的总称 (FC6A-N****)
数字输出模块	带输出端子的数字 I/O 模块的总称 (FC6A-R***、FC6A-T****)
数字混合I/O模块	带有输入和输出端子的数字 I/O 模块的总称 (FC6A-M*****)
模拟I/O模块	模拟量输入模块、模拟量输出模块、模拟量混合I/O模块的总称
模拟量输入模块	带输入端子的模拟 I/O 模块的总称 (FC6A-J***、FC6A-J4CN*、FC6A-J4CH**、FC6A-J8CU*)
模拟量输出模块	带输出端子的模拟I/O 模块的总称 (FC6A-K***)
模拟量混合I/O模块	带有输入和输出端子的模拟I/O模块的总称 (FC6A-L03CN*、FC6A-L06A*)
通信模块	串行通信模块的简称
串行通信模块	FC6A-SIF52、FC6A-SIF524的总称

项目	说明
PID模块	FC6A-F2M*、FC6A-F2MR* 的总称
增设扩展模块	一体型、组合型主机、组合型从机的总称
一体型	FC6A-EXM2、FC6A-EXM24的总称
组合型主机	FC6A-EXM1M
组合型从机	FC6A-EXM1S、FC6A-EXM1S4 的总称
HMI模块	FC6A-PH1
盒基本模块	FC6A-HPH1
盒	I/O盒、通信盒的总称
I/O盒	数字I/O盒、模拟I/O盒的总称
数字I/O盒	扩展数字输入或数字输出的 I/O 盒的总称 (FC6A-PN4、FC6A-PT*4)
模拟I/O盒	扩展模拟输入或输出的 I/O 盒的总称 (FC6A-PJ2A、FC6A-P*2**)
通信盒	RS232C通信盒、RS485通信盒、Bluetooth通信盒的总称
RS232C通信盒	FC6A-PC1
RS485通信盒	FC6A-PC3
Bluetooth通信盒	FC6A-PC4
WindLDR	FC6A 型的梯形图编程软件
USB电缆	USB维护电缆 (HG9Z-XCM42)、USB-mini B 端口延长电缆 (HG9Z-XCE21) 的总称
用户程序	梯形图编程软件 WindLDR 设置的功能设置以及梯形图程序的统合数据
功能设置	FC6A 型的各项设置内容。 设置选项卡、模块构成编辑器的设置内容
梯形图程序	主程序、子程序、用户定义宏的总称
主程序	第一行是梯形图程序进入点的程序。进入点是梯形图程序执行的起点它是在梯形图程序编辑器的 [主程序] 选项卡上创建的。
子程序	以下程序。 <ul style="list-style-type: none"> 主程序内的从 LABEL 指令到 LRET 指令的程序 用 WindLDR 的子程序功能创建的程序 (梯形图程序编辑器的选项卡中显示为 [# 子程序] (# : 子程序编号))
用户定义宏	WindLDR 用户定义宏功能创建的程序。 (在梯形图程序编辑器的选项卡中, 用 [# 用户定义宏] 表示 (# : 用户定义宏的编号))
源设备	作为计算对象的设备 (为执行高级指令数据的存储位)
目标设备	它是一个存储操作结果的设备
定时器指令	TML指令、TIM指令、TMH指令、TMS指令的总称
断开延时定时器指令	TMLO指令、TIMO指令、TMHO指令、TMSO指令的总称
计数器指令	CNT指令、CDP指令、CUD指令的总称
双字计数器指令	CNTD指令、CDPD指令、CUDD指令的总称
移位寄存器指令	SFR指令、SFRN指令的总称
计数器比较指令	CC=指令、CC>=指令的总称
比较指令	CMP= 指令、CMP<> 指令、CMP< 指令、CMP> 指令、CMP<= 指令、CMP>= 指令的总称
脉冲输出指令	PULS指令、PWM指令、RAMP指令、RAMPL指令、ARAMP指令、ABS指令、JOG指令的总称
双/示教定时器指令	DTIM指令、DTML指令、DTMH指令、DTMS指令、TTIM指令的总称

WindLDR 的名称

本手册中使用的名称	WindLDR 操作步骤
功能设置	“设置”选项卡 > “功能设置”组
监控	选择“联机”>“监控”>“启动监控器”。
PLC 状态	选择“联机”>“PLC”>“状态”。
通信设置	选择“联机”>“通信”>“设置”。
Modbus 主机请求列表	在“设置”选项卡的“功能设置”中单击“通信端口”，并在所显示的“功能设置”对话框内，从“通信端口”的“通信模式”中选择“Modbus RTU 主机”或“Modbus TCP 客户端”
应用程序按钮	显示在菜单栏左侧的按钮。单击即可显示“新建”、“保存”、“另存为”等菜单或最近使用的项目、及“WindLDR 选项”、“退出 WindLDR”按钮

目录

安全上的重要注意事项	前言 -1
关于本手册	前言 -2
本手册中使用的名称及简称	前言 -5

第1章：	前言	
	概述	1-1
	通信功能	1-9
	维护通信	1-9
	用户通信	1-10
	Modbus 通信	1-11
	数据连接通信	1-11
	MC 协议通信	1-12
	BACnet 通信	1-12
	EtherNet/IP 通信	1-13
	MQTT 通信	1-14
	J1939 通信	1-14
第2章：	设备地址	
	设备地址	2-1
	特殊内部继电器	2-4
	特殊数据寄存器	2-19
第3章：	通信设置	
	设置列表	3-1
	通信端口的设置	3-2
	网络设置	3-3
	网络管理	3-11
	连接设置	3-14
	远程主机列表	3-21
第4章：	维护通信	
	通过 USB 端口的维护通信	4-3
	端口 1 的维护通信	4-4
	端口 1 (Bluetooth) 的维护通信	4-6
	通过以太网端口 1 及 2 的维护通信	4-7
	端口 2 至 33 的维护通信	4-13
	端口 1 ~ 3 (Bluetooth) 的维护通信	4-16
	HMI- 以太网端口的维护通信	4-17
第5章：	用户通信指令	
	TXD (发送)	5-2
	RXD (接收)	5-10
	ETXD (以太网用户通信发送)	5-23
	ERXD (以太网用户通信接收)	5-23
	通过串行通信进行的用户通信	5-24
	通过串行通信 (Bluetooth) 进行的用户通信	5-39
	通过以太网通信进行的用户通信	5-40
	用户通信错误	5-52
	ASCII 字符代码表	5-53
	示例程序 - 用户通信 TXD	5-54
	示例程序 - 用户通信 RXD	5-56
第6章：	Modbus 通信	
	概述	6-1
	通过 RS232C/RS485 进行的 Modbus RTU 通信	6-2

	Modbus RTU 主机通信	6-3
	Modbus RTU 从机通信	6-9
	通信格式	6-15
	通过以太网通信进行的 Modbus TCP 通信	6-21
	Modbus TCP 客户端	6-22
	Modbus TCP 服务器	6-27
	Modbus RTU Pass-Through 功能	6-31
第 7 章:	数据连接通信	
	数据连接系统安装	7-2
	使用其他 PLC 的数据连接	7-10
第 8 章:	J1939 通信	
	通过 CAN 进行 J1939 通信的概要	8-1
	设置 J1939 通信	8-7
第 9 章:	Bluetooth 通信	
	概要	9-1
	Bluetooth 通信盒的设置	9-3
	与 Bluetooth 通信的设置和通信	9-8
第 10 章:	FTP 服务器 / 客户端功能	
	FTP 服务器功能	10-1
	FTP 客户端功能	10-8
第 11 章:	PING 发送功能	
	PING (Ping)	11-1
第 12 章:	EMAIL 发送功能	
	概述	12-1
	EMAIL 指令 (发送电子邮件)	12-1
	电子邮件地址簿	12-11
	电子邮件编辑器	12-13
	附件编辑器	12-16
	附加日志数据设置	12-19
第 13 章:	Web 服务器	
	概述	13-1
	支持机型及通信端口	13-1
	操作环境	13-2
	基本规格	13-3
	Plus CPU 模块 Web 服务器	13-5
	“证书和私钥的批量设置”向导	13-11
	导入认证机构的根证书	13-16
	HMI 模块的 Web 服务器设置	13-19
	用户帐户设置	13-23
	系统网页	13-26
	用户网页	13-30
	自定义网页	13-44
	CGI	13-56
	SD 记忆卡读取 (文件下载)	13-58
	限制事项	13-58
第 14 章:	通信监控	
	概要	14-1
	通信监控的开始和停止	14-1

第 15 章:	<i>BACnet/IP</i>	
	概要	15-1
	BACnet/IP 中使用的通信端口	15-1
	BACnet 规格	15-2
	BACnet	15-3
	功能	15-5
	BACnet/IP 的动作	15-11
	BACnet/IP 的设置	15-13
	对象	15-33
	主要属性	15-41
第 16 章:	<i>EtherNet/IP 通信</i>	
	概要	16-1
	EtherNet/IP 中使用的通信端口	16-2
	EtherNet/IP 规格	16-3
	关于 EtherNet/IP	16-4
	功能	16-5
	EtherNet/IP 通信的动作	16-13
	EtherNet/IP 通信的设置	16-18
	“EtherNet/IP 设置”对话框	16-19
	EtherNet/IP 通信的设置流程	16-32
	EtherNet/IP 的设置示例	16-35
	对象	16-41
第 17 章:	<i>MQTT 通信</i>	
	概要	17-1
	支持机型以及通信端口	17-2
	基本规格	17-3
	支持的 MQTT 代理和云服务	17-3
	MQTT 通信的设置	17-5
	“MQTT 设置”对话框	17-6
	MQTT 设置	17-6
	连接至一个通用的 MQTT 代理	17-7
	连接至 AWS IoT Core	17-12
	使用 SAS 连接至 Azure IoT Hub	17-17
	使用 X.509 证书连接至 Azure IoT Hub	17-21
	通过 DPS 连接至 Azure IoT Hub	17-25
	发布	17-32
	订阅	17-37
	发布标题	17-49
	订阅标题	17-57
第 18 章:	<i>MC 协议通信</i>	
	概述	18-1
	通过以太网通讯进行的 MC 协议通讯	18-2
索引		
	产品保修说明	前言 -1

1: 前言

简介

本章将对配备通信接口的 FC6A 型的概要进行介绍。

概述

FC6A 型可使用各种通信接口进行多种通信。

All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块

是：可用作通信端口

否：无法用作通信端口

—：CPU 模块中未配备

通信 / 功能		端口						盒插槽			扩展模块	
		串行 端口 1	USB 端口	以太网 端口 1	以太网 端口 2	HMI- 以太网 端口	CAN 端口	1	2	3	通信模块	
RS232C 通信 / RS485 通信	维护通信	是 *1	是	是	—	是 *2	否	是 *4	是 *4	—	是	
	用户通信		否	否		否						
	Modbus RTU 通信 (主机 / 从机)											
	数据连接通信 (主站 / 从站)											
Bluetooth 通信	用户通信	否	否	—	是 *2	否	—	—	—	否		
	维护通信 (服务器)											
	用户通信 (服务器 / 客户端)											
EtherNet 通信	用户通信 (UDP)	否	否	—	否	否	—	—	—	否		
	Modbus TCP 通信 (服务器 / 客户端)				是 *2							
	MC 协议通信 (客户端)				否						否	否
	BACnet/IP 通信											
	EtherNet/IP 通信											
	MQTT 通信				是 *2							
	FTP 服务器 / 客户端功能											
	Web 服务器功能											
	PING 发送											
	CAN 通信				J1939 通信						否	否
是 *2												

*1 仅 All-in-One CPU 模块可使用串行端口 1 进行各种通信。

*2 可在 CPU 模块中连接 HMI 模块，增加 HMI- 以太网端口进行通信。

*3 仅 CAN J1939 All-in-One CPU 模块型可使用 CAN 端口进行 J1939 通信。

*4 可将通信盒连接到盒插槽上进行通信。

1: 前言

Plus CPU 模块

是：可用作通信端口

否：无法用作通信端口

—：CPU 模块中未配备

通信 / 功能		端口						盒插槽			扩展模块		
		串行 端口 1	USB 端口	以太网 端口 1	以太网 端口 2	HMI- 以太网 端口	CAN 端口	1	2	3	通信模块		
RS232C 通信 / RS485 通信	维护通信		是	是	是	是*1					是		
	用户通信												
	Modbus RTU 通信 (主机 / 从机)			否	否	否		是*2	是*2	是*3			
	数据连接通信 (主站 / 从站)												
Bluetooth 通信	用户通信												
EtherNet 通信	维护通信 (服务器)	—	否	是	是	是*1	—	否	否	否	否		
	用户通信 (服务器 / 客户端)												
	用户通信 (UDP)												
	Modbus TCP 通信 (服务器 / 客户端)												
	MC 协议通信 (客户端)			否									
	BACnet/IP 通信				是	否							
	EtherNet/IP 通信				否	是							
	MQTT 通信			是	否	是*1							
	FTP 服务器 / 客户端功能											否	×
	Web 服务器功能												
	PING 发送											是	是*1
EMAIL 发送	否	否											
CAN 通信			J1939 通信	否	否								

*1 可在 CPU 模块中连接 HMI 模块，增加 HMI- 以太网端口进行通信。

*2 可在 CPU 模块中连接盒基本模块，在盒插槽上连接通信盒进行通信。

*3 可在 CPU 模块中连接 HMI 模块，在盒插槽上连接通信盒进行通信。

通信接口

各通信接口的概要及规格如下所示。

USB 端口

可使用本端口与 PC 进行连接，并进行维护通信。

通信类型	USB2.0 Full speed, CDC class
通信功能	可与 PC 进行维护通信，或通过 USB 电源下载程序
连接器	USB mini-B
与内部电路隔离	不隔离

串行端口 1

可使用本端口与支持 PC 或可编程显示器、打印机等的 RS232C/RS485 通信的外部设备进行通信。可进行维护通信、用户通信、Modbus RTU 通信（主机 / 从机）、数据连接通信（主站 / 从站）。仅 All-in-One CPU 模块配备有串行端口 1。

通信类型	选择 EIA RS232C 或 RS485 的软件
最大通信速度	115,200bps
通信功能	维护通信、用户通信、Modbus RTU 通信（主站 / 从站）、数据连接通信（主站 / 从站）
连接器	RJ45
电缆	RS232C: 多芯屏蔽电缆 RS485: 屏蔽双绞电缆
最大电缆长度	RS232C: 5m RS485: 200m
与内部电路绝隔离	不隔离

串行通信模块

串行通信模块在连接到 CPU 模块的基本扩展侧或通信扩展侧时可使用。可使用本端口与支持 PC 或可编程显示器、打印机等的 RS232C/RS485 通信的外部设备进行通信。可进行维护通信、用户通信、Modbus RTU 通信（主机 / 从机）、数据连接通信（主站 / 从站）。

型号	FC6A-SIF52、FC6A-SIF524	
点数	2	
电气特性	EIA RS232C/EIA RS485*1	
最大通信速度	115,200bps	
通信功能	维护通信、用户通信、Modbus RTU 通信（主机 / 从机）、数据连接通信（主站 / 从站）	
最大电缆长度	15m (EIA RS232C)/1,200m (EIA RS485)	
与内部电路隔离	通信端口和内部电路之间	光电耦合器
	端口之间	变压器
电缆	推荐电缆	RS232C: 多芯屏蔽电缆 24AWG
		RS485: 屏蔽双绞电缆 24AWG

*1 在 WindLDR 的“功能设置”中进行指定。无法设置为“数据位：7 位”且“奇偶校验：无”。

1: 前言

RS232C 通信盒及 RS485 通信盒

RS232C 通信盒及 RS485 通信盒在连接到以下盒插槽时可使用。

- All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的盒插槽 1 及 2
- Plus CPU 模块上扩展的盒基本模块盒插槽 1 及 2
- Plus CPU 模块上扩展的 HMI 模块盒插槽 3

可使用本端口与支持 PC 或可编程显示器、打印机等 RS232C/RS485 通信的外部设备进行通信。可进行维护通信、用户通信、Modbus RTU 通信（主机 / 从机）、数据连接通信（主站 / 从站）。

型号	FC6A-PC1	FC6A-PC3
电气特性	EIA RS232C	EIA RS485
最快通信速度	115,200bps	115,200bps
通信功能	维护通信、用户通信、Modbus RTU 通信（主机 / 从机）、数据连接通信（主站 / 从站）	
最大电缆长度	5m	200m
与内部电路隔离	不隔离	不隔离
电缆	推荐电缆	多芯屏蔽电缆：24AWG
		屏蔽双绞电缆：24AWG

Bluetooth 通信盒

Bluetooth 通信盒在连接到以下盒插槽时可使用。

- All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的盒插槽 1 及 2
- Plus CPU 模块上扩展的盒基本模块盒插槽 1 及 2
- Plus CPU 模块上扩展的 HMI 模块盒插槽 3

可使用 Bluetooth 与 Windows PC 或智能手机、条形码读取器等支持 Bluetooth 通信的外部设备进行通信。可进行维护通信、用户通信。

型号	FC6A-PC4
电气特性	Bluetooth ver.2.1 + EDR
轮廓	SPP (Serial Port Profile) iAP (iPod Accessory Protocol)
通信功能	维护通信、用户通信
传输距离	10m (Class 2)

以太网端口 1 及 2

该端口支持以下通信和功能。此端口具有多个连接，并且每个连接可以同时使用不同的通信协议。有关详细信息，请参见“第 3 章连接设置”。

是：可用作通信端口

否：无法用作通信端口

—：CPU 模块中未配备

通信 / 功能		All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块		Plus CPU 模块			
		以太网端口 1	以太网端口 2	以太网端口 1	以太网端口 2		
EtherNet 通信	维护通信（服务器）	是	—	是	是		
	用户通信（服务器 / 客户端）						
	用户通信（UDP）					否	
	Modbus TCP 通信（服务器 / 客户端）	是		否	是		
	MC 协议通信（客户端）						
	BACnet/IP 通信	否		—	否	否	
	EtherNet/IP 通信				否	是	
	MQTT 通信				是	否	否
	FTP 服务器 / 客户端功能						
	Web 服务器功能						
	PING 发送	是		是			
EMAIL 发送	否	否					

通信类型	符合 IEEE802.3 标准
通信速度	10BASE-T、100BASE-TX
连接数	All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块：8 个 Plus CPU 模块：16 个
通信功能	维护通信（服务器）、用户通信（服务器 / 客户端）、用户通信（UDP）、Modbus TCP 通信（服务器 / 客户端）MC 协议通信（客户端）
Web 服务器功能	是
Web 数据存储位置	FROM
Web 数据最大尺寸 ^{*1}	5.0MB
EMAIL 发送功能	是
连接器	RJ45
电缆	CAT 5 以上的 STP
最大电缆长度	100m
与内部电路隔离	脉冲变压器隔离

*1 在 Web 数据最大大小内，用户可使用区域因使用系统网页和网页编辑器条件的不同，存在以下差异。

		系统网页	
		使用	不使用
网页编辑器	使用	2.5MB	3.0MB
	不使用	4.5MB	5.0MB

1: 前言

CAN 端口

CAN J1939 All-in-One CPU 模块可使用本端口进行 J1939 通信。

通信类型	CAN 总线通信
通信速度	250kbps
通信功能	J1939 通信
连接器	FC6A-PMTE05PN02
电缆	SAE-J1939-11 : 双绞线屏蔽电缆
	SAE-J1939-15 : 无双绞线屏蔽电缆
最大电缆长度	SAE-J1939-11 : 40m、短线最大 1m
	SAE-J1939-15 : 40m、短线最大 3m
终端电阻	120Ω (0.5W 以上)
与内部电路隔离	电源部: 变压器隔离 信号部: 电耦隔离, 光电耦合器隔离

HMI- 以太网端口

仅在连接 CPU 模块和 HMI 模块时可使用 HMI- 以太网端口。
可使用本端口与支持 PC 或可编程显示器等的以太网通信的外部设备进行通信。
本端口拥有可在以太网通信中使用的 8 个连接。各连接可设置维护通信 (服务器)。
此外, 本端口支持服务器功能、EMAIL 发送功能。

通信类型	符合 IEEE802.3 标准
通信速度	10BASE-T、100BASE-TX
连接数	8 个
通信模式	维护通信
Web 服务器功能	是
Web 数据存储位置	FROM
Web 数据最大尺寸 *1	5.0MB
EMAIL 发送功能	是
连接器	RJ45
电缆	CAT 5 以上的 STP
最大电缆长度	100m
与内部电路隔离	脉冲变压器隔离

*1 在 Web 数据最大大小内, 用户可使用区域因使用系统网页和网页编辑器条件的不同, 存在以下差异。

		系统网页	
		使用	不使用
网页编辑器	使用	2.5MB	3.0MB
	不使用	4.5MB	5.0MB



注意

- 将 FC6A 型连接至网络时, 必须对网络采取充分的安全措施, 如防止非法访问等。务必向您的网络管理员或互联网服务供应商咨询。
对于以太网通信过程中的安全性引起的损害或问题, IDEC 概不负责。
- 采用防火墙等适当的措施限制访问 FC6A 型的 IP 地址和端口。

CPU 模块和通信接口一览

可在各 CPU 模块上装配或扩展的通信接口如下所示。

有关各模块的通信接口位置的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“产品规格”。

型号	USB 端口	串行端口 1	通信模块端口 4 ~ 33 ^{*1}	通信盒	以太网 端口 1	以太网 端口 2	CAN 端口	HMI- 以太网 端口 ^{*4}
FC6A-C16***E FC6A-C24***E	1	1	最大 6	最大 1 ^{*2}	1	—	—	最大 1
FC6A-C40***E		—		最大 2 ^{*2}			1	
FC6A-C40***EJ			—	最大 30			最大 3 ^{*3}	
FC6A-D***CEE		—					—	

*1 串行通信模块在连接到 CPU 模块的基本扩展侧或通信扩展侧时可使用。

从靠近 CPU 模块一侧起依序分配端口 4、5.....。

*2 在连接到 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的盒插槽 1 及 2 时可使用通信盒。

*3 通信盒在连接到 Plus CPU 模块上连接的盒基本模块盒插槽 1 及 2、连接到 Plus CPU 模块上连接的 HMI 模块盒插槽 3 时可使用。

*4 HMI- 以太网端口可在 HMI 模块中使用。

通信端口的分配

All-in-One CPU 模块、CAN J1939 All-in-One CPU 模块及 Plus CPU 模块可与目标设备进行串行通信。各 CPU 模块可通过标配的接口或增加接口进行串行通信。可进行串行通信的接口如下所示。

- All-in-One CPU 模块 : 串行端口 1、盒插槽 1 及 2、串行通信模块的端口
- CAN J1939 All-in-One CPU 模块 : 盒插槽 1 及 2、串行通信模块的端口
- Plus CPU 模块 : 盒插槽 1 ~ 3、串行通信模块的端口

要将各接口用作通信端口，需要根据通信目标设备的通信规格设置通信格式。

请在“功能设置”的“通信端口”中设置通信端口。

各通信接口与通信端口编号的对应关系如下所示。

有关设置的详情，请参见第 3-2 页上的“通信端口的设置”。

■ CPU 模块及 HMI 模块的端口、盒插槽 1 ~ 3

- : CPU 模块中未配备
- 否 : 无法用作串行通信端口

CPU 模块		端口						盒插槽		
		串行端口 1	USB 端口	以太网端口 1	以太网端口 2	HMI-以太网端口	CAN 端口	1	2	3
All-in-One CPU 模块	16-I/O 型	端口 1	否	否	—	否*1	—	端口 2 *2*5	—	—
	24-I/O 型								端口 3 *2*5	
	40-I/O 型									
CAN J1939 All-in-One CPU 模块	40-I/O 型	—	—	—	—	—	—	—	—	
Plus CPU 模块	Plus 16-I/O 型	—	—	—	否	—	—	端口 1 *3*5	端口 2 *3*5	端口 3 *4*5
	Plus 32-I/O 型								—	

*1 CPU 模块上连接 HMI 模块，以增加了 HMI-Ethernet 端口时
 *2 盒插槽上连接了通信盒时
 *3 CPU 模块上连接盒基本模块，且盒插槽上连接了通信盒时
 *4 CPU 模块上连接 HMI 模块，且盒插槽上连接了通信盒时
 *5 无法设置为“数据位：7 位”且“奇偶校验：无”。

注释：有关串行端口 1、盒插槽 1 及 2 的位置、串行端口 1、通信模块、通信盒的配线详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“产品规格”。

■ 通信模块的端口

CPU 模块		通信模块					
		第 1 台	第 2 台	第 3 台	第 4 台	...	第 15 台
All-in-One CPU 模块	16-I/O 型	端口 4、5	端口 6、7	端口 8、9	*1	*1	*1
	24-I/O 型						
	40-I/O 型						
CAN J1939 All-in-One CPU 模块	40-I/O 型	—	—	—	—	—	—
Plus CPU 模块	Plus 16-I/O 型	—	—	—	端口 10、11	...	端口 32、33
	Plus 32-I/O 型						

*1 All-in-One CPU 模块、CAN J1939 All-in-One CPU 模块上无法连接 4 台以上的通信模块。

通信功能

FC6A 型支持维护通信、用户通信、Modbus 通信、数据连接通信、J1939 通信以及 BACnet 通信。

本节将对各通信功能的概要和连接示例进行介绍。

维护通信

借助 FC6A 型维护通信，您可以通过 PC 上安装的 PLC 编程软件 WindLDR 检查 FC6A 型的运行状态和 I/O 状态、监控和更改设备值以及下载和上传用户程序。有关维护通信的详情，请参见第 4-1 页上的“维护通信”。

支持的端口 *1

USB 端口	串行端口 1	以太网端口 1、2 和 HMI- 以太网端口 *2	通信盒和通信模块	CAN 端口
是	是	是	是	否

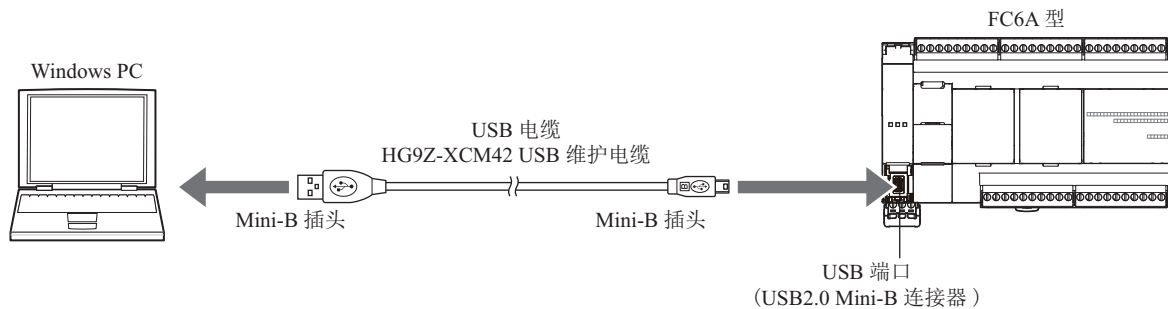
*1 根据使用的端口不同，可使用的维护通信的通信方式存在限制。有关限制的详情，请参见第 4-1 页上的“维护通信”。

*2 HMI- 以太网端口仅可使用维护通信。

注释：连接 HMI 模块时，可使用 HMI- 以太网端口进行维护通信。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

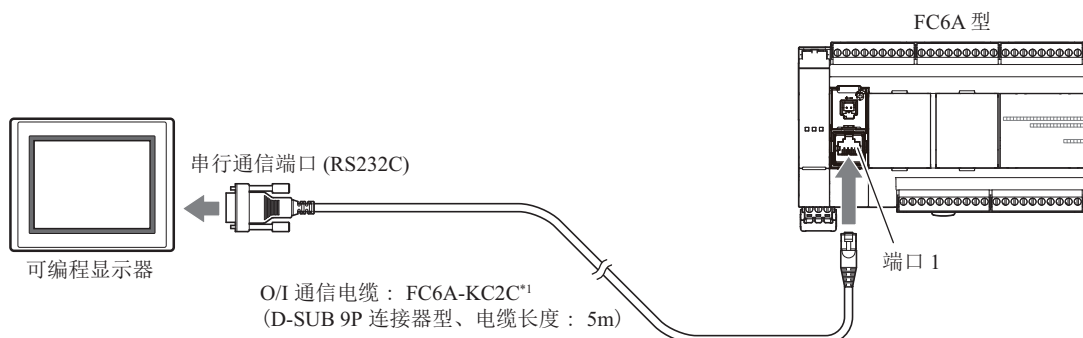
■ 使用 USB 端口的 1:1 维护通信

此例展示了通过 USB 连接的 FC6A 型和 PC 的 1:1 维护通信系统。采用的是 USB 维护电缆 (HG9Z-XCM42)。



■ 与采用串行端口 1 的 IDEC 可编程显示器进行 1:1 维护通信

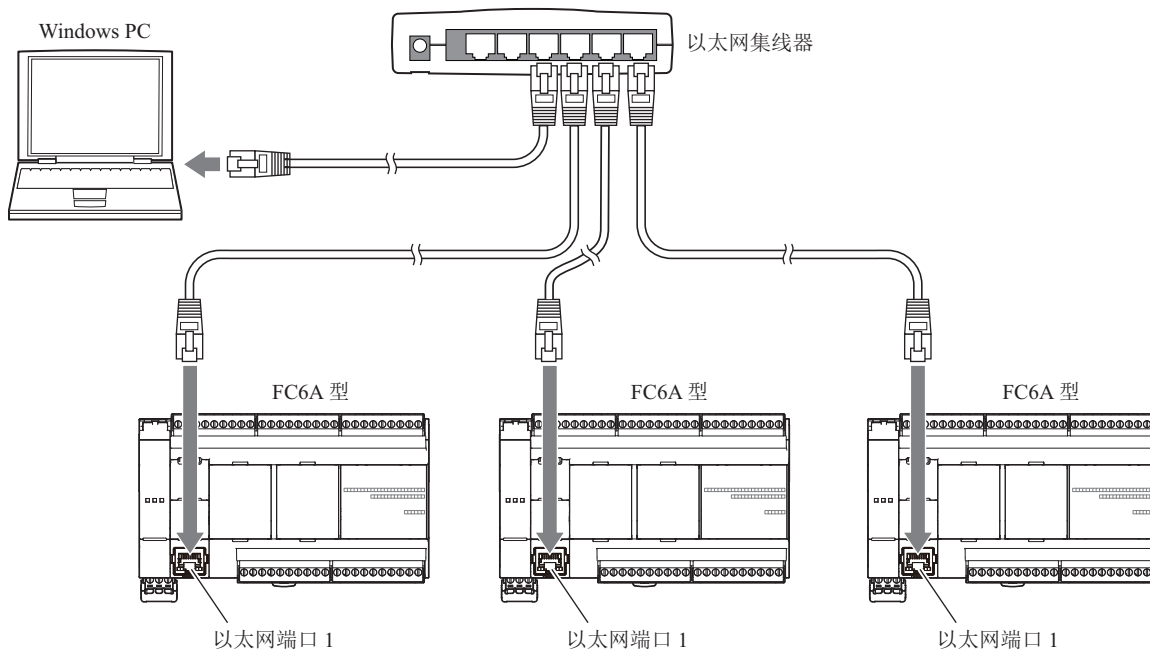
通过 FC6A 型与可编程显示器进行维护通信，使用可编程显示器监控及更改 FC6A 型的设备值的示例。连接 FC6A 型的端口 1 与 IDEC 生产的可编程显示器。



*1 有关 O/I 通信电缆的详情，请参见《FC6A 型 MICROSmart 用户手册》附录“各种电缆”。

■ 使用以太网端口的 1:N 维护通信

此例展示了通过以太网连接的 FC6A 型和 PC。以太网电缆连接到 FC6A 型以太网端口 1，FC6A 型通过以太网集线器连接到 PC。



用户通信

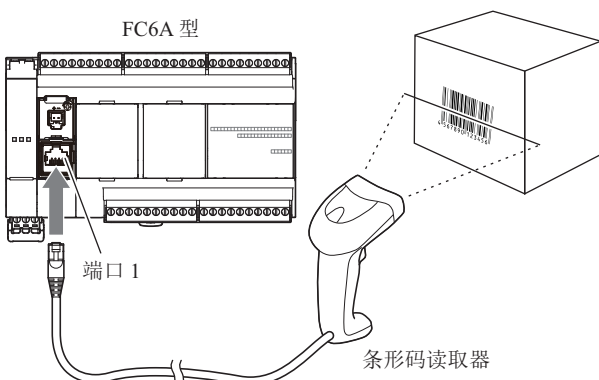
借助 FC6A 型用户通信，您可以控制 PC、打印机和条形码读取器等外部设备。有关用户通信的详情，请参见第 5-1 页上的“用户通信指令”。

支持的端口

USB 端口	串行端口 1	以太网端口 1 和 2	通信盒和通信模块	CAN 端口
否	是	是	是	否

■ 使用端口 1 的用户通信

此例展示了 FC6A 型接收条形码读取器读取的数据的系统。连接 FC6A 型的端口 1 与条形码读取器。



Modbus 通信

FC6A 型符合 Modbus RTU 协议，可用作 Modbus 通信的主机或从机。通过 Modbus 通信，可监控和更改逆变器与温度控制器的数据。

此外，以太网端口 1 及 2 支持 Modbus TCP 通信协议。

有关 Modbus 通信的详情，请参见第 6-1 页上的“Modbus 通信”。

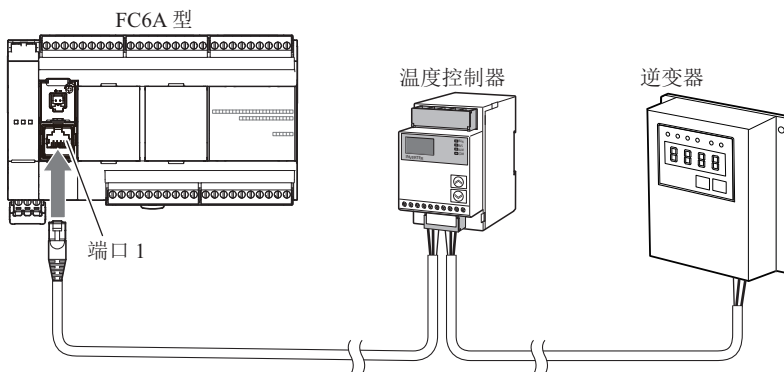
支持的端口

USB 端口	串行端口 1	以太网端口 1 和 2	通信盒*1 和通信模块	CAN 端口
否	是	是	是	否

*1 仅支持 RS232C 通信盒及 RS485 通信盒。

■ 使用串行端口 1 的 Modbus 通信

此例展示了 FC6A 型与支持 Modbus RTU 的温度控制器及逆变器进行通信的系统。连接 FC6A 型的端口 1 与温度控制器。



数据连接通信

FC6A 型支持数据连接通信，可使用端口 1 至 3 在 CPU 模块间共享数据。此外，FC6A 型与 FC5A 系列或 FC4A 系列的 CPU 模块皆可共享数据。可通过 WindLDR 进行设置，进行最多 32 台的分布式控制。

有关 Modbus 通信的详情，请参见第 7-1 页上的“数据连接通信”。

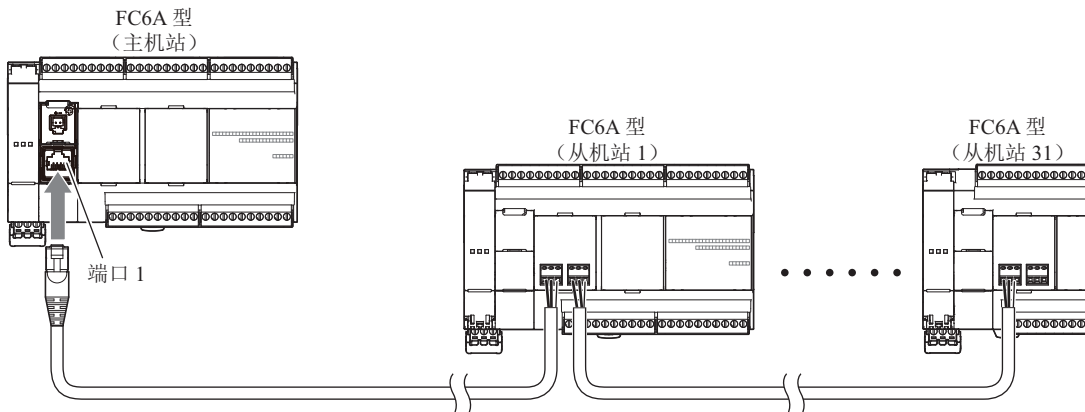
支持的端口

USB 端口	串行端口 1	以太网端口 1 和 2	通信盒*1 和通信模块	CAN 端口
否	是	否	是	否

*1 仅支持 RS232C 通信盒及 RS485 通信盒。

■ 使用串行端口 1 的数据连接通信

将 FC6A 型作为主站，与多个 CPU 模块进行通信的示例。连接 FC6A 型的端口 1 与作为从站的 CPU 模块。

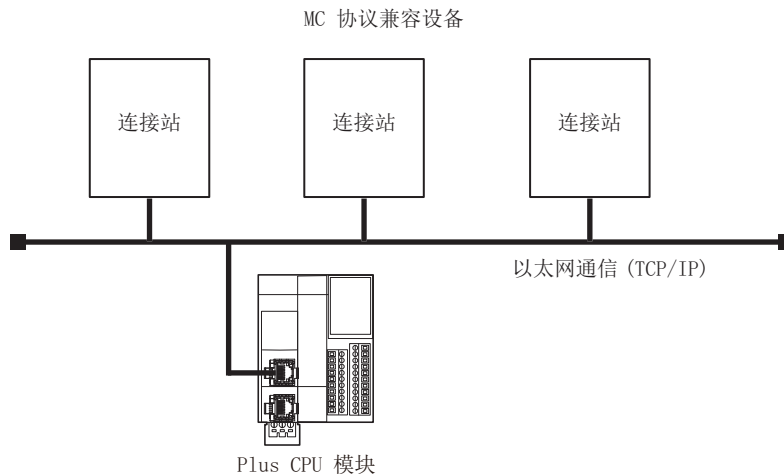


MC 协议通信

Plus CPU 模块可使用以太网端口 1 和 2，与其他支持 MC 协议通信的设备进行通信。
有关 MC 协议通信的详情，请参见第 18-1 页上的“MC 协议通信”。

支持的端口

USB 端口	串行端口 1	以太网端口 1	以太网端口 2	通信盒和通信模块	CAN 端口
否	否	是	是	否	否



BACnet 通信

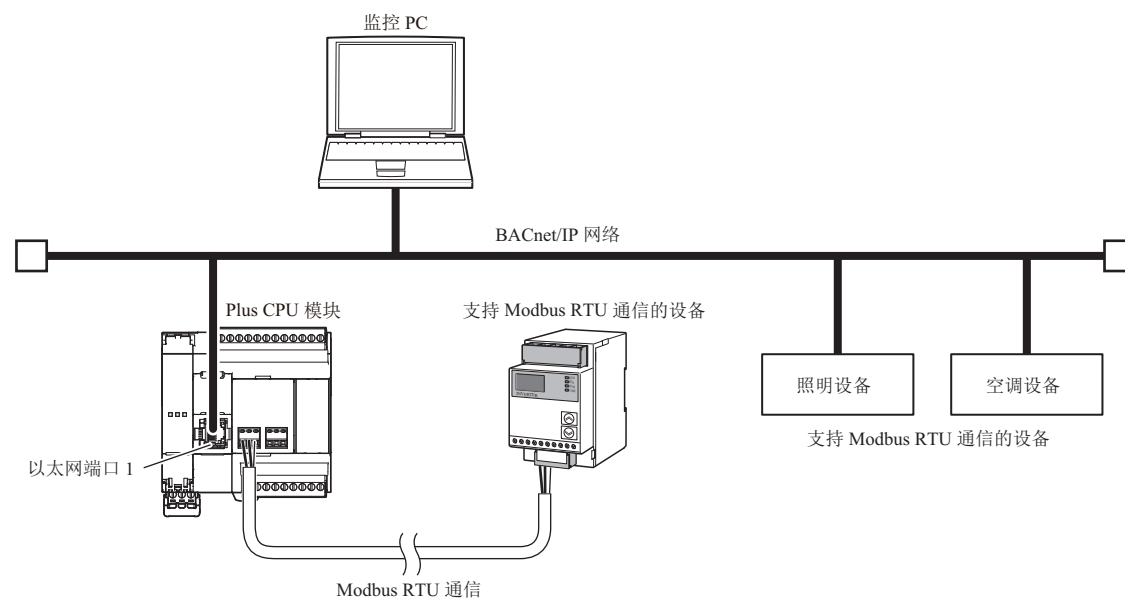
Plus CPU 模块可使用以太网端口 1 连接 BACnet/IP 网络，与其他支持 BACnet 通信的设备进行通信。有关 BACnet 通信的详情，请参见第 15-1 页上的“BACnet/IP”。

支持的端口

USB 端口	串行端口 1	以太网端口 1	以太网端口 2	通信盒和通信模块	CAN 端口
否	否	是	否	否	否

■ 使用以太网端口 1 的 BACnet 通信

Plus CPU 模块会收集支持 Modbus RTU 通信设备的信息，与 BACnet/IP 网络上连接的支持 BACnet 通信的设备进行通信，并公开其信息的示例。



EtherNet/IP 通信

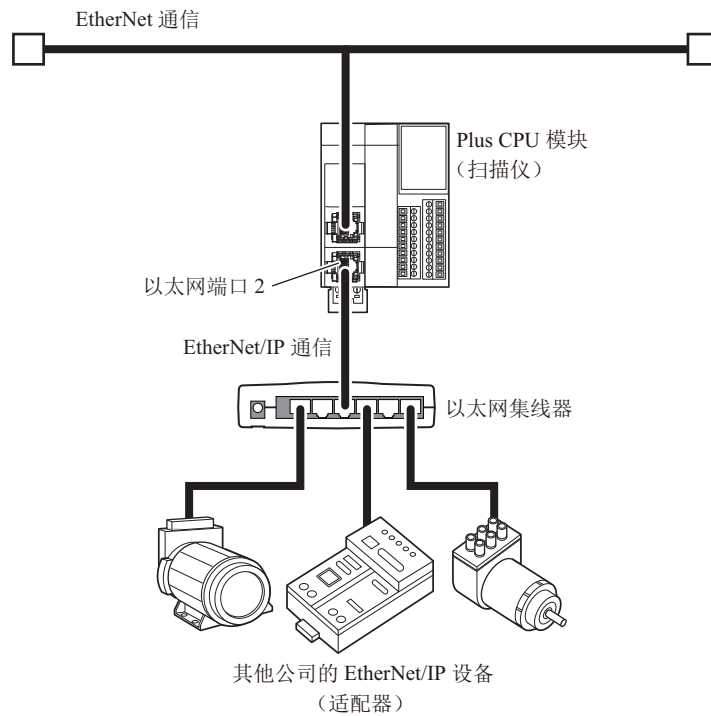
Plus CPU 模块可使用以太网端口 2 连接现有的以太网，与其他支持 EtherNet/IP 通信的设备进行通信。此外，由于 EtherNet/IP 通信采用了标准的以太网技术，可以混用各类支持以太网的设备来构建网络。有关 EtherNet/IP 通信的详情，请参见第 16-1 页上的“16: EtherNet/IP 通信”。

支持的端口

USB 端口	串行端口 1	以太网端口 1	以太网端口 2	通信盒和通信模块	CAN 端口
否	否	否	是	否	否

■ 使用以太网端口 2 的 EtherNet/IP 通信

Plus CPU 模块与支持 EtherNet/IP 通信的设备通信，控制各设备的示例。



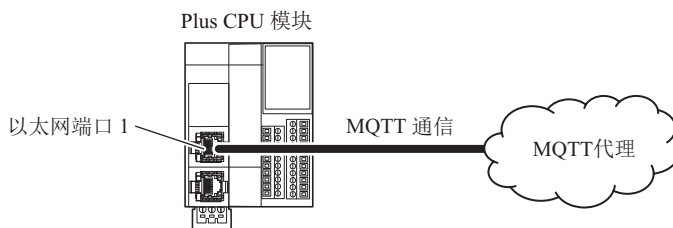
MQTT 通信

Plus CPU 模块可使用以太网端口 1，作为 MQTT 通信中的客户端（发布者和订阅者）与代理通信。有关 MQTT 通信的详情，请参见第 17-1 页上的“17: MQTT 通信”

支持的端口

USB 端口	串行端口 1	以太网端口 1	以太网端口 2	通信盒和通信模块	CAN 端口
否	否	是	否	否	否

■ 使用以太网端口 1 的 MQTT 通信



J1939 通信

CAN J1939 All-in-One CPU 模块可使用 CAN 端口连接 J1939 通信网络，与其他支持 J1939 通信的设备进行通信。可收发符合 SAEJ1939 标准的信息。

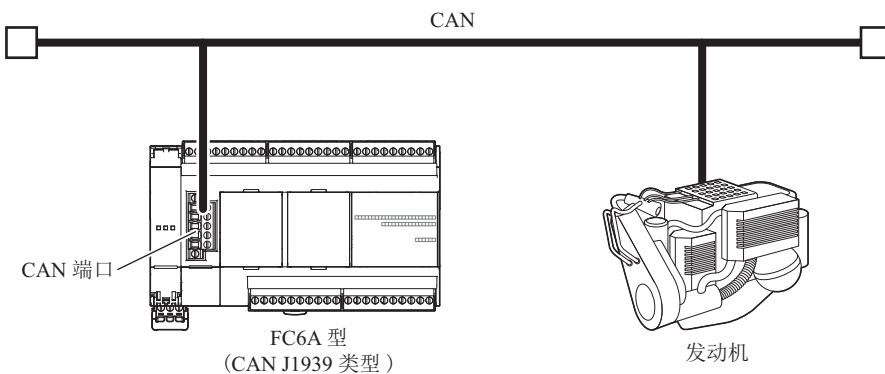
有关 J1939 通信的详情，请参见第 8-1 页上的“J1939 通信”。

支持的端口

USB 端口	串行端口 1	以太网端口 1 和 2	通信盒	CAN 端口
否	否	否	否	是

■ 使用 CAN 端口的 J1939 通信

FC6A 型与支持 J1939 通信的发动机进行通信的示例。连接 CAN J1939 All-in-One CPU 模块的 CAN 端口与发动机。



2: 设备地址

本章将对以基本指令或高级指令使用的输入输出、内部继电器、寄存器、定时器、计数器等各种设备的分配、特殊内部继电器及特殊数据寄存器的分配详情进行介绍。

请作为各设备的参考使用。

注释: 输入及操作 FC6A 型的用户程序时, 需要具备专业知识。

请充分理解本手册内容及程序后, 再使用 FC6A 型。

设备地址

All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块

设备	符号	单位	范围 (点数)		
			16-I/O 型	24-I/O 型	40-I/O 型
输入 *1	I	位	I0 - I10 (9 点)	I0 - I15 (14 点)	I0 - I27 (24 点)
扩展输入 *1	I	位	I30 - I187 (128 点) I190 - I507*2 (256 点) I630 - I633*4 (4 点)	I30 - I307 (224 点) I310 - I627*3 (256 点) I630 - I633*4 (4 点)	I30 - I307 (224 点) I310 - I627*3 (256 点) I630 - I637*4 (8 点)
输出 *1	Q	位	Q0 - Q6 (7 点)	Q0 - Q11 (10 点)	Q0 - Q17 (16 点)
扩展输出 *1	Q	位	Q30 - Q187 (128 点) Q190 - Q507*2 (256 点) Q630 - Q633*4 (4 点)	Q30 - Q307 (224 点) Q310 - Q627*3 (256 点) Q630 - Q633*4 (4 点)	Q30 - Q307 (224 点) Q310 - Q627*3 (256 点) Q630 - Q637*4 (8 点)
内部继电器 *1	M	位	M0 - M7997 (6,400 点) M10000 - M17497 (6,000 点)		
特殊内部继电器 *1	M	位	M8000 - M8317 (256 点)		
移位寄存器	R	位	R0 - R255 (256 点)		
定时器	T	位 / 字	T0 - T1023 (1,024 点)		
计数器	C	位 / 字	C0 - C511 (512 点)		
数据寄存器	D	位 / 字	D0000 - D7999 (8,000 点) D10000 - D55999 (46,000 点)		
特殊数据寄存器	D	位 / 字	D8000 - D8499 (500 点)		

*1 设备地址的下 1 位是八进制数 (0 ~ 7)。

*2 I190 ~ I507 及 Q190 ~ Q507 为可用于使用增设扩展模块一体型连接扩展模块 (增设扩展侧) 时的设备。

*3 I310 ~ I627 及 Q310 ~ Q627 为可用于使用增设扩展模块一体型连接扩展模块 (增设扩展侧) 时的设备。

*4 I630 ~ I637 及 Q630 ~ Q637 为可用于连接 I/O 盒时的设备。

2: 设备地址

Plus CPU 模块

设备	符号	单位	范围 (点数)	
			Plus 16-I/O 型	Plus 32-I/O 型
输入 *1	I	位	I0 - I17 (8 点)	I0 - I17 (16 点)
扩展输入 *1	I	位	I30 - I307 (224 点) I310 - I627*2 (256 点) I630 - I643*3 (12 点) I1000 - I10597*4 (2,016 点)	
输出 *1	Q	位	Q0 - Q7 (8 点)	Q0 - Q17 (16 点)
扩展输出 *1	Q	位	Q30 - Q307 (224 点) Q310 - Q627*2 (256 点) Q630 - Q643*3 (12 点) Q1000 - Q10597*4 (2,016 点)	
内部继电器 *1	M	位	M0 - M7997 (6,400 点) M10000 - M21247 (9,000 点)	
特殊内部继电器 *1	M	位	M8000 - M9997 (1,600 点)	
移位寄存器	R	位	R0 - R255 (256 点)	
定时器	T	位 / 字	T0 - T1999 (2,000 点)	
计数器	C	位 / 字	C0 - C511 (512 点)	
数据寄存器	D	位 / 字	D0000 - D7999 (8,000 点) D10000 - D61999 (52,000 点)	
	非保持型数据寄存器	D	位 *7 / 字	D70000 - D269999*5 (200,000 点)
特殊数据寄存器	D	位 / 字	D8000 - D8899 (900 点)	
间接寄存器 *6	P	2 个字	P0 - P15 (16 点)	

*1 设备地址的下 1 位是八进制数 (0 ~ 7)。

*2 I310 ~ I627 及 Q310 ~ Q627 为可用于使用增设扩展模块连接扩展模块 (增设扩展侧) 时的设备。(节点 0)

*3 I630 ~ I643 及 Q630 ~ Q643 为可用于连接 I/O 盒时的设备。

*4 I1000 ~ I10597 及 Q1000 ~ Q10597 为可用于使用增设扩展模块组合型主机 / 组合型从机及增设扩展模块一体型连接扩展模块时的设备。(节点 1 ~ 10)

*5 D70000 ~ D269999 无法指定保持。停止 → 运行过程中虽然会保持, 但通电时将复位为 0。

*6 可使用的数据类型仅为 L (长整)。

*7 仅 SCRPT 指令执行的脚本内和 UMACRO 指令使用的参数设备可使用。

- **输入 (I)、扩展输入 (I)**
将来自外部设备的 ON/OFF 信息输入至 FC6A 型的设备。
- **输出 (Q)、扩展输出 (Q)**
将来自 FC6A 型的 ON/OFF 信息输出至外部设备的设备。
- **内部继电器 (M)**
在 FC6A 型内部使用的位单位的设备。
- **特殊内部继电器 (M)**
在 FC6A 型内部使用的位单位的设备。各位分配有特殊的功能。
- **移位寄存器 (R)**
以 SFR 指令或 SFRN 指令使用的位单位的设备。根据脉冲输入对数据的位串进行移位。
- **定时器 (T)**
在 FC6A 型内部使用的定时器。包括定时器位 (符号: T、单位: 位)、定时器预置值 (符号: TP、单位: 字)、定时器当前值 (符号: TC、单位: 字) 3 种设备。
可用作接通延时定时器、断开延时定时器。有关定时器 (T) 的详情, 请参见《梯形图编程手册》第 3 章中的“使用定时器或计数器作为源设备”。
- **计数器 (C)**
在 FC6A 型内部使用的计数器。包括计数器位 (符号: C、单位: 位)、计数器预置值 (符号: CP、单位: 字)、计数器当前值 (符号: CC、单位: 字) 3 种设备。可用作加计数器、加/减切换计数器。有关计数器 (C) 的详情, 请参见《梯形图编程手册》第 3 章中的“使用定时器或计数器作为源设备”。
- **数据寄存器 (D)**
在 FC6A 型内部用于写入数值数据的字单位的设备。可同时用作位单位的设备。
- **特殊数据寄存器 (D)**
在 FC6A 型内部用于写入数值数据的字单位的设备。各数据寄存器分配有特殊的功能。可同时用作位单位的设备。
- **间接寄存器 (P)**
用于存储间接值的 2 字单位设备, 在指定间接设备地址时使用。有关详情, 请参见《梯形图编程手册》第 3 章中的“间接指定”。

注释:

- 内部继电器 (M0000 ~ M7997、M10000 ~ M21247) 和特殊内部继电器 (M8000 ~ M9997) 的设备符号同样为“M”, 但设备特性有所不同。特殊内部继电器的各位分配有特殊的功能。
- 数据寄存器 (D0000 ~ D7999、D10000 ~ D61999、D70000 ~ D269999) 和特殊数据寄存器 (D8000 ~ D8899) 的设备符号同样为“D”, 但设备特性有所不同。各特殊数据寄存器分配有特殊的功能。

2: 设备地址

特殊内部继电器

特殊内部继电器设备地址



警告 请勿改写特殊内部继电器设备地址中写为保留的区域内的数据。否则系统可能无法正常工作。

注释: 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明	停止时	电源关闭	读 / 写	
M8000	开始控制	保持	保持	读 / 写	
M8001	1 秒时钟复位	清除	清除	读 / 写	
M8002	所有输出关闭	清除	清除	读 / 写	
M8003	进位 (Cy) 或借位 (Bw)	清除	清除	读 / 写	
M8004	用户程序执行错误	清除	清除	读 / 写	
M8005	通信错误	保持	清除	读 / 写	
M8006	通信禁止标记 (数据连接主机时)	保持	保持	读 / 写	
M8007	初始化标记 (数据连接主机时) / 通信停止标记 (数据连接从机时)	清除	清除	读 / 写	
M8010	状态 LED 执行	执行	清除	读 / 写	
M8011 M8012	— 保留 —	—	—	—	
M8013	日历 / 时钟数据写入 / 调整错误标记	执行	清除	读 / 写	
M8014	日历 / 时钟数据读取错误标记	执行	清除	读 / 写	
M8015	— 保留 —	—	—	—	
M8016	日历数据写入标记	执行	清除	读 / 写	
M8017	时钟数据写入标记	执行	清除	读 / 写	
M8020	日历 / 时钟数据写入标记	执行	清除	读 / 写	
M8021	时钟数据调整标记	执行	清除	读 / 写	
M8022	用户通信接收指令取消标记 (端口 1)	清除	清除	读 / 写	
M8023	用户通信接收指令取消标记 (端口 2)	清除	清除	读 / 写	
M8024	BMOV/WSFT 执行标记	保持	保持	读 / 写	
M8025	停止时维持输出	保持	清除	读 / 写	
M8026	用户通信接收指令取消标记 (端口 3)	清除	清除	读 / 写	
M8027	高速计数器 (组 1/10)	计数方向标记	保持	清除	读 / 写
M8030		比较一致输出复位	清除	清除	读 / 写
M8031		门输入	保持	清除	读 / 写
M8032		复位输入	保持	清除	读 / 写
M8033	用户通信接收指令取消标记 (端口 4)	清除	清除	读 / 写	
M8034	高速计数器 (组 3/13)	比较一致输出复位	清除	清除	读 / 写
M8035		门输入	保持	清除	读 / 写
M8036		复位输入	保持	清除	读 / 写
M8037	— 保留 —	—	—	—	
M8040	高速计数器 (组 4/14)	比较一致输出复位	清除	清除	读 / 写
M8041		门输入	保持	清除	读 / 写
M8042		复位输入	保持	清除	读 / 写
M8043	高速计数器 (组 5/16)	计数方向标记	保持	清除	读 / 写
M8044		比较一致输出复位	清除	清除	读 / 写
M8045		门输入	保持	清除	读 / 写
M8046		复位输入	保持	清除	读 / 写

设备地址	说明	停止时	电源关闭	读 / 写	
M8047 M8050	— 保留 —	—	—	—	
M8051	高速计数器 (组 2/I1)	比较一致输出复位	清除	清除	读 / 写
M8052		门输入	保持	清除	读 / 写
M8053		复位输入	保持	清除	读 / 写
M8054		比较打开状态	保持	清除	读
M8055		上溢出	保持	清除	读
M8056	— 保留 —	—	—	—	
M8057	高速计数器 (组 6/I7)	比较一致输出复位	清除	清除	读 / 写
M8060		门输入	保持	清除	读 / 写
M8061		复位输入	保持	清除	读 / 写
M8062		比较打开状态	保持	清除	读
M8063		上溢出	保持	清除	读
M8064- M8067	— 保留 —	—	—	—	
M8070	SD 记忆卡固定状态	保持	清除	读	
M8071	SD 记忆卡写入标记	保持	清除	读	
M8072	SD 记忆卡固定解除	执行	清除	读 / 写	
M8073	功能开关状态	执行	清除	读	
M8074	电池电压测量标记	执行	清除	读 / 写	
M8075- M8077	— 保留 —	—	—	—	
M8080	数据连接从机 1 通信完成继电器 (数据连接主机时)	执行	清除	读	
M8081	数据连接从机 2 通信完成继电器	执行	清除	读	
M8082	数据连接从机 3 通信完成继电器	执行	清除	读	
M8083	数据连接从机 4 通信完成继电器	执行	清除	读	
M8084	数据连接从机 5 通信完成继电器	执行	清除	读	
M8085	数据连接从机 6 通信完成继电器	执行	清除	读	
M8086	数据连接从机 7 通信完成继电器	执行	清除	读	
M8087	数据连接从机 8 通信完成继电器	执行	清除	读	
M8090	数据连接从机 9 通信完成继电器	执行	清除	读	
M8091	数据连接从机 10 通信完成继电器	执行	清除	读	
M8092	数据连接从机 11 通信完成继电器	执行	清除	读	
M8093	数据连接从机 12 通信完成继电器	执行	清除	读	
M8094	数据连接从机 13 通信完成继电器	执行	清除	读	
M8095	数据连接从机 14 通信完成继电器	执行	清除	读	
M8096	数据连接从机 15 通信完成继电器	执行	清除	读	
M8097	数据连接从机 16 通信完成继电器	执行	清除	读	
M8100	数据连接从机 17 通信完成继电器	执行	清除	读	
M8101	数据连接从机 18 通信完成继电器	执行	清除	读	
M8102	数据连接从机 19 通信完成继电器	执行	清除	读	
M8103	数据连接从机 20 通信完成继电器	执行	清除	读	
M8104	数据连接从机 21 通信完成继电器	执行	清除	读	
M8105	数据连接从机 22 通信完成继电器	执行	清除	读	
M8106	数据连接从机 23 通信完成继电器	执行	清除	读	
M8107	数据连接从机 24 通信完成继电器	执行	清除	读	
M8110	数据连接从机 25 通信完成继电器	执行	清除	读	
M8111	数据连接从机 26 通信完成继电器	执行	清除	读	
M8112	数据连接从机 27 通信完成继电器	执行	清除	读	
M8113	数据连接从机 28 通信完成继电器	执行	清除	读	

2: 设备地址

设备地址	说明		停止时	电源关闭	读 / 写
M8114	数据连接从机 29 通信完成继电器		执行	清除	读
M8115	数据连接从机 30 通信完成继电器		执行	清除	读
M8116	数据连接从机 31 通信完成继电器		执行	清除	读
M8117	数据连接全部从机通信完成继电器		执行	清除	读
M8120	初始化脉冲		清除	清除	读
M8121	1 秒时钟		执行	清除	读
M8122	100 毫秒时钟		执行	清除	读
M8123	10 毫秒时钟		执行	清除	读
M8124	定时器 / 计数器预置值更改		保持	清除	读
M8125	动作中输出		清除	清除	读
M8126	程序运行中下载完成后打开 1 次扫描		清除	清除	读
M8127	— 保留 —		—	—	—
M8130	高速计数器 (组 1/10)	复位状态	保持	清除	读
M8131		比较打开状态	保持	清除	读
M8132	— 保留 —		—	—	—
M8133	高速计数器 (组 3/13)	比较打开状态	保持	清除	读
M8134	高速计数器 (组 4/14)	比较打开状态	保持	清除	读
M8135	高速计数器 (组 5/16)	复位状态	保持	清除	读
M8136		比较打开状态	保持	清除	读
M8137	中断输入 I0 状态 (组 1/10)	(ON: 已允许, OFF: 已禁止)	清除	清除	读
M8140	中断输入 I1 状态 (组 2/11)		清除	清除	读
M8141	中断输入 I3 状态 (组 3/13)		清除	清除	读
M8142	中断输入 I4 状态 (组 4/14)		清除	清除	读
M8143	中断输入 I6 状态 (组 5/16)		清除	清除	读
M8144	定时器中断状态		清除	清除	读
M8145	用户通信接收指令取消标记 (端口 5)		清除	清除	读 / 写
M8146	用户通信接收指令取消标记 (端口 6)		清除	清除	读 / 写
M8147	用户通信接收指令取消标记 (端口 7)		清除	清除	读 / 写
M8150	比较结果 1		保持	清除	读
M8151	比较结果 2		保持	清除	读
M8152	比较结果 3		保持	清除	读
M8153	捕捉输入开 / 关状态	组 1/10	保持	清除	读
M8154		组 2/11	保持	清除	读
M8155		组 3/13	保持	清除	读
M8156		组 4/14	保持	清除	读
M8157		组 5/16	保持	清除	读
M8160		组 6/17	保持	清除	读
M8161	高速计数器 (组 1/10)	上溢出	保持	清除	读
M8162		下溢出	保持	清除	读
M8163	高速计数器 (组 5/16)	上溢出	保持	清除	读
M8164		下溢出	保持	清除	读
M8165	高速计数器 (组 3/13)	上溢出	保持	清除	读
M8166	高速计数器 (组 4/14)	上溢出	保持	清除	读
M8167	中断输入 I7 状态 (组 6/17)	(ON: 已允许, OFF: 已禁止)	保持	清除	读
M8170	用户通信接收指令取消标记 (端口 8)		清除	清除	读 / 写
M8171	— 保留 —		—	—	—
M8172	晶体管源型输出过电流检测	组 1	执行	清除	读
M8173		组 2	执行	清除	读
M8174		组 3	执行	清除	读
M8175		组 4	执行	清除	读

设备地址	说明	停止时	电源关闭	读/写
M8176	用户通信接收指令取消标记 (端口 9)	清除	清除	读/写
M8177- M8183	— 保留 —	—	—	—
M8184	HMI 模块 网络设置更改触发器	执行	清除	读/写
M8185	在夏时制时间段	执行	清除	读
M8186	以太网端口 1 自动 Ping 执行中	执行	清除	读
M8187	以太网端口 1 自动 Ping 停止标记	执行	清除	读/写
M8190	CPU 模块 以太网端口 1 网络设置更改触发器	执行	清除	读/写
M8191	SNTP 获取标记	执行	清除	读/写
M8192	中断输入 I0 边沿	ON: 上升沿 OFF: 下降沿	清除	读
M8193	中断输入 I3 边沿		清除	读
M8194	中断输入 I4 边沿		清除	读
M8195	中断输入 I6 边沿		清除	读
M8196	中断输入 I7 边沿		清除	读
M8197	中断输入 I1 边沿		清除	读
M8200	用户通信接收指令取消标记	连接 1	清除	读/写
M8201		连接 2	清除	读/写
M8202		连接 3	清除	读/写
M8203		连接 4	清除	读/写
M8204		连接 5	清除	读/写
M8205		连接 6	清除	读/写
M8206		连接 7	清除	读/写
M8207		连接 8	清除	读/写
M8210	— 保留 —	—	—	—
M8211	初始化 HMI 模块电子邮件发送服务器设置	执行	清除	读/写
M8212	连接状态 (ON: 已接通, OFF: 未接通)	连接 1	执行	读
M8213		连接 2	执行	读
M8214		连接 3	执行	读
M8215		连接 4	执行	读
M8216		连接 5	执行	读
M8217		连接 6	执行	读
M8220		连接 7	执行	读
M8221		连接 8	执行	读
M8222	断开用户通信连接	连接 1	执行	读/写
M8223		连接 2	执行	读/写
M8224		连接 3	执行	读/写
M8225		连接 4	执行	读/写
M8226		连接 5	执行	读/写
M8227		连接 6	执行	读/写
M8230		连接 7	执行	读/写
M8231		连接 8	执行	读/写
M8232	HMI 模块连接信息参考 连接状态	执行	清除	读
M8233- M8247	— 保留 —	—	—	—
M8250	从 SD 记忆卡下载执行标记	执行	清除	读/写
M8251	向 SD 记忆卡上传执行标记	执行	清除	读/写
M8252	SD 记忆卡下载执行中	执行	清除	读
M8253	SD 记忆卡上传执行中	执行	清除	读
M8254	SD 记忆卡下载 / 上传执行完成输出	执行	清除	读
M8255	SD 记忆卡下载 / 上传执行错误输出	执行	清除	读

2: 设备地址

设备地址	说明		停止时	电源关闭	读 / 写
M8256 M8257	— 保留 —		—	—	—
M8260	写入配方执行标记		执行	清除	读 / 写
M8261	读取配方执行标记		执行	清除	读 / 写
M8262	写入配方执行中		执行	清除	读 / 写
M8263	读取配方执行中		执行	清除	读 / 写
M8264	配方频道执行完成输出		执行	清除	读 / 写
M8265	执行配方错误输出		执行	清除	读 / 写
M8266	配方块执行完成输出		执行	清除	读 / 写
M8267	配方内存 (ROM- 区域) 读取限制		执行	清除	读
M8270	— 保留 —		—	—	—
M8271	从 SD 记忆卡进行 MQTT 基本设置下载的执行标记		执行	清除	读 / 写
M8272	从 SD 记忆卡进行 MQTT 基本设置下载的执行完成输出		执行	清除	读
M8273	从 SD 记忆卡进行 MQTT 基本设置下载的错误输出		执行	清除	读
M8274- M8297	— 保留 —		—	—	—
M8300	J1939 通信许可标记		清除	清除	读 / 写
M8301	J1939 联机状态		清除	清除	读
M8302	J1939 本机地址确认状态		清除	清除	读
M8303	J1939 通信错误输出		清除	清除	读
M8304	发生 J1939 通信总线关闭输出		清除	清除	读
M8305- M8310	— 保留 —		—	—	—
M8311	ESC+ 键输入 (上)	ESC+ 键输入 (⬆)	清除	清除	读
M8312	ESC+ 键输入 (下)	ESC+ 键输入 (⬇)	清除	清除	读
M8313	ESC+ 键输入 (左)	ESC+ 键输入 (⬅)	清除	清除	读
M8314	ESC+ 键输入 (右)	ESC+ 键输入 (➡)	清除	清除	读
M8315- M8319	— 保留 —		—	—	—
M8320	初始化增设扩展模块组合型主机 / 从机		执行	清除	读 / 写
M8321- M8330	— 保留 —		—	—	—
M8331	以太网端口 2 自动 Ping 执行中		执行	清除	读
M8332	以太网端口 2 自动 Ping 停止标记		执行	清除	读 / 写
M8333	CPU 模块 以太网端口 2 网络设置更改触发器		执行	清除	读 / 写
M8334	用户通信接收指令取消标记	连接 9	清除	清除	读 / 写
M8335		连接 10	清除	清除	读 / 写
M8336		连接 11	清除	清除	读 / 写
M8337		连接 12	清除	清除	读 / 写
M8340		连接 13	清除	清除	读 / 写
M8341		连接 14	清除	清除	读 / 写
M8342		连接 15	清除	清除	读 / 写
M8343		连接 16	清除	清除	读 / 写
M8344	以太网端口 1 初始化电子邮件发送服务器设置		执行	清除	读 / 写
M8345	连接状态 (ON: 已接通, OFF: 未接通)	连接 9	执行	清除	读
M8346		连接 10	执行	清除	读
M8347		连接 11	执行	清除	读
M8350		连接 12	执行	清除	读
M8351		连接 13	执行	清除	读
M8352		连接 14	执行	清除	读

设备地址	说明	停止时	电源关闭	读 / 写	
M8353	连接状态 (ON: 已接通, OFF: 未接通)	连接 15	执行	清除	读
M8354		连接 16	执行	清除	读
M8355	断开用户通信连接	连接 9	执行	清除	读 / 写
M8356		连接 10	执行	清除	读 / 写
M8357		连接 11	执行	清除	读 / 写
M8360		连接 12	执行	清除	读 / 写
M8361		连接 13	执行	清除	读 / 写
M8362		连接 14	执行	清除	读 / 写
M8363		连接 15	执行	清除	读 / 写
M8364		连接 16	执行	清除	读 / 写
M8365	用户通信接收指令取消标记 (端口 10)	清除	清除	清除	读 / 写
M8366	用户通信接收指令取消标记 (端口 11)	清除	清除	清除	读 / 写
M8367	用户通信接收指令取消标记 (端口 12)	清除	清除	清除	读 / 写
M8370	用户通信接收指令取消标记 (端口 13)	清除	清除	清除	读 / 写
M8371	用户通信接收指令取消标记 (端口 14)	清除	清除	清除	读 / 写
M8372	用户通信接收指令取消标记 (端口 15)	清除	清除	清除	读 / 写
M8373	用户通信接收指令取消标记 (端口 16)	清除	清除	清除	读 / 写
M8374	用户通信接收指令取消标记 (端口 17)	清除	清除	清除	读 / 写
M8375	用户通信接收指令取消标记 (端口 18)	清除	清除	清除	读 / 写
M8376	用户通信接收指令取消标记 (端口 19)	清除	清除	清除	读 / 写
M8377	用户通信接收指令取消标记 (端口 20)	清除	清除	清除	读 / 写
M8380	用户通信接收指令取消标记 (端口 21)	清除	清除	清除	读 / 写
M8381	用户通信接收指令取消标记 (端口 22)	清除	清除	清除	读 / 写
M8382	用户通信接收指令取消标记 (端口 23)	清除	清除	清除	读 / 写
M8383	用户通信接收指令取消标记 (端口 24)	清除	清除	清除	读 / 写
M8384	用户通信接收指令取消标记 (端口 25)	清除	清除	清除	读 / 写
M8385	用户通信接收指令取消标记 (端口 26)	清除	清除	清除	读 / 写
M8386	用户通信接收指令取消标记 (端口 27)	清除	清除	清除	读 / 写
M8387	用户通信接收指令取消标记 (端口 28)	清除	清除	清除	读 / 写
M8390	用户通信接收指令取消标记 (端口 29)	清除	清除	清除	读 / 写
M8391	用户通信接收指令取消标记 (端口 30)	清除	清除	清除	读 / 写
M8392	用户通信接收指令取消标记 (端口 31)	清除	清除	清除	读 / 写
M8393	用户通信接收指令取消标记 (端口 32)	清除	清除	清除	读 / 写
M8394	用户通信接收指令取消标记 (端口 33)	清除	清除	清除	读 / 写
M8395- M8400	— 保留 —	—	—	—	—
M8401	从 SD 记忆卡进行用于服务器功能的文件下载执行标记	执行	清除	清除	读 / 写
M8402	从 SD 记忆卡进行用于服务器功能的文件下载执行完成输出	执行	清除	清除	读
M8403	从 SD 记忆卡进行用于服务器功能的文件下载错误输出	执行	清除	清除	读
M8404	Plus CPU 模块 Web 服务器启动状态	执行	清除	清除	读
M8405- M8447	— 保留 —	—	—	—	—
M8450	BACnet 通信许可	执行	清除	清除	读 / 写
M8451- M8457	— 保留 —	—	—	—	—
M8460	EtherNet/IP 通信许可	清除	清除	清除	读 / 写
M8461- M8597	— 保留 —	—	—	—	—

2: 设备地址

设备地址	说明	停止时	电源关闭	读 / 写	
M8600	高速计数器 (组 3/I3)	复位状态	保持	清除	读
M8601		下溢出	保持	清除	读
M8602		计数方向标记	保持	清除	读
M8603- M8997	— 保留 —	—	—	—	

特殊内部继电器补充说明

■ M8000: 开始控制

控制 FC6A 型的运行 / 停止状态。打开 M8000 时进入运行状态，关闭则进入停止状态。请参见《梯形图编程手册》第 1 章中的“启动 / 停止操作”。但是，功能开关、停止输入及复位输入皆优先于开始控制。虽然 M8000 会在电源关闭时保持其状态，但当关闭时间超过备份时间而使保持数据丢失时，将按照“功能设置”的“在保持数据错误时的运行 / 停止指定”中设置的内容（运行指定 / 停止指定）进行动作。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“在保持数据错误时的运行 / 停止指定”。

■ M8001: 1 秒时钟复位

M8001 打开期间，M8121（1 秒时钟）将始终关闭。

■ M8002: 所有输出均关闭

M8002 打开期间，所有输出将关闭。以梯形图程序创建的自保持电路也将关闭。

■ M8003: 进位 (Cy)/ 借位 (Bw)

执行高级指令过程中产生进位 (Cy) 或借位 (Bw) 时，M8003 将打开。有关详情，请参见《梯形图编程手册》第 3 章中的“进位和借位”。

■ M8004: 用户程序执行错误

执行用户程序过程中发生错误时，M8004 将打开。有关用户程序执行错误的详情，请参见《FC6A 型 MICROSmart 用户手册》第 13 章中的“用户程序执行错误代码”。

■ M8005: 通信错误

在数据连接通信时发生错误时，M8005 将打开。即使错误解除也将保持。

■ M8006: 通信禁止标记（数据连接主机时）

数据连接通信时、M8006 打开期间，将停止通信。

■ M8007: 初始化标记（数据连接主机时） / 通信停止标记（数据连接从机时）

数据连接主机时：在运行状态下将该标记从关闭设为打开时，为确认连接状态，数据连接将仅初始化 1 次。用于构成数据连接的从机电源接通时间晚于主机时。

数据连接从机时：来自主机的通信经过 10s 以上切断时，该标记将打开。如果可正常接收，则关闭。

■ M8010: 状态 LED 执行

M8010 打开期间，状态 LED [STAT] 亮起。关闭期间，状态 LED [STAT] 熄灭。

■ M8013: 日历 / 时钟数据写入 / 调整错误标记

时钟写入或时钟调整处理无法正常执行时，将打开。处理正常完成后将关闭。

■ M8014: 日历 / 时钟数据读取错误标记

从内部时钟向特殊数据寄存器（D8008 ~ D8021）的日历数据及时钟数据读取失败时，将打开。读取正常完成后将关闭。

■ M8016: 日历数据写入标记

在向日历数据（只写）的特殊数据寄存器（D8015 ~ D8018）中写入数据后，将 M8016 从关闭设为打开时，将向内置时钟写入日历数据（年、月、日、星期）。

■ M8017: 时钟数据写入标记

在向时钟数据（只写）的特殊数据寄存器（D8019 ~ D8021）中写入数据后，将 M8017 从关闭设为打开时，将向内置时钟写入时钟数据（时、分、秒）。

■ M8020: 日历 / 时钟数据写入标记

在向日历 / 时钟数据（只写）的特殊数据寄存器（D8015 ~ D8021）写入数据后，将 M8020 从关闭设为打开时，将向内置时钟写入日历数据（年、月、日、星期）及时钟数据（时、分、秒）。

■ M8021: 时钟数据调整标记

将 M8021 从关闭设为打开时，将对内置时钟的秒数进行调整。

- 秒数在 0 ~ 29 之间将 M8021 从关闭设为打开时，将秒数设为 0。

- 秒数在 30 ~ 59 之间将 M8021 从关闭设为打开时，分钟数将加 1 且将秒数设为 0。

■ M8022: 用户通信接收指令取消标记（端口 1）

M8022 打开期间，将取消正在端口 1 中执行的用户通信（接收指令）。

2: 设备地址

■ M8023: 用户通信接收指令取消标记 (端口 2)

M8023 打开期间, 将取消正在端口 2 中执行的用户通信 (接收指令)。

■ M8024: BMOV/WSFT 执行标记

执行 WSFT (字移位) 指令、BMOV (块传送) 指令过程中, M8024 将打开; 指令执行完成 (正常结束) 后将关闭。

■ M8025: 停止时维持输出

在 FC6A 型运行过程中打开 M8025, 停止时输出将保持运行时的状态。再次运行时, M8025 将自动关闭。

■ M8026: 用户通信接收指令取消标记 (端口 3)

M8026 打开期间, 将取消正在端口 3 中执行的用户通信 (接收指令)。

■ M8033、M8145 ~ M8147、M8170、M8176、M8365 ~ M8394: 用户通信接收指令取消标记 (端口 4 ~ 33)

这些标记打开期间, 将取消正在相应端口中执行的用户通信 (接收指令)。

M8033 = 用户通信接收指令取消标记 (端口 4)

M8145 = 用户通信接收指令取消标记 (端口 5)

M8146 = 用户通信接收指令取消标记 (端口 6)

M8147 = 用户通信接收指令取消标记 (端口 7)

M8170 = 用户通信接收指令取消标记 (端口 8)

M8176 = 用户通信接收指令取消标记 (端口 9)

M8365 = 用户通信接收指令取消标记 (端口 10)

M8366 = 用户通信接收指令取消标记 (端口 11)

M8367 = 用户通信接收指令取消标记 (端口 12)

M8370 = 用户通信接收指令取消标记 (端口 13)

M8371 = 用户通信接收指令取消标记 (端口 14)

M8372 = 用户通信接收指令取消标记 (端口 15)

M8373 = 用户通信接收指令取消标记 (端口 16)

M8374 = 用户通信接收指令取消标记 (端口 17)

M8375 = 用户通信接收指令取消标记 (端口 18)

M8376 = 用户通信接收指令取消标记 (端口 19)

M8377 = 用户通信接收指令取消标记 (端口 20)

M8380 = 用户通信接收指令取消标记 (端口 21)

M8381 = 用户通信接收指令取消标记 (端口 22)

M8382 = 用户通信接收指令取消标记 (端口 23)

M8383 = 用户通信接收指令取消标记 (端口 24)

M8384 = 用户通信接收指令取消标记 (端口 25)

M8385 = 用户通信接收指令取消标记 (端口 26)

M8386 = 用户通信接收指令取消标记 (端口 27)

M8387 = 用户通信接收指令取消标记 (端口 28)

M8390 = 用户通信接收指令取消标记 (端口 29)

M8391 = 用户通信接收指令取消标记 (端口 30)

M8392 = 用户通信接收指令取消标记 (端口 31)

M8393 = 用户通信接收指令取消标记 (端口 32)

M8394 = 用户通信接收指令取消标记 (端口 33)

■ M8027 ~ M8032、M8034 ~ M8036、M8040 ~ M8046、M8051 ~ M8055、M8057 ~ M8063、M8130、M8131、M8133 ~ M8136、M8161 ~ M8167、M8600 ~ M8602: 用于高速计数器的特殊内部继电器

用于高速计数器的特殊内部继电器。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“高速计数器”。

M8027 ~ M8032、M8130、M8131、M8161、M8162 = 高速计数器 (组 1/I0)

M8034 ~ M8036、M8133、M8165、M8600 ~ M8602 = 高速计数器 (组 3/I3)

M8040 ~ M8042、M8134、M8166 = 高速计数器 (组 4/I4)

M8043 ~ M8046、M8135、M8136、M8163、M8164 = 高速计数器 (组 5/I6)

M8051 ~ M8055 = 高速计数器 (组 2/I1)

M8057 ~ M8063 = 高速计数器 (组 6/I7)

■ M8070: SD 记忆卡固定状态

在 FC6A 型中插入 SD 记忆卡, 且 SD 记忆卡被识别并处于可使用状态时, 将打开。未插入 SD 记忆卡或未被识别时, 将关闭。

■ M8071: SD 记忆卡写入标记

访问 SD 记忆卡过程中，将打开。访问完成后将关闭。

■ M8072: SD 记忆卡固定解除

将 M8072 从关闭设为打开时，将停止访问 SD 记忆卡。为了将已停止访问的 SD 记忆卡设为可访问，请重新插入 SD 记忆卡。

■ M8073: 功能开关状态

表示位于 CPU 模块前面的功能开关的状态。

功能开关为 1 时，将打开。功能开关为 0 时，将关闭。

■ M8074: 电池电压测量标记

表示备份用电池的电池电压测量状态。

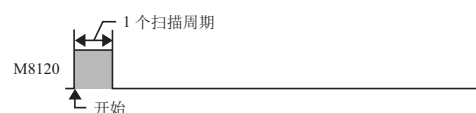
将 M8074 从关闭设为打开时，将开始测量电池电压；测量完成后将关闭。

■ M8080 ~ M8117: 数据连接通信完成继电器

用于数据连接通信的特殊内部继电器。有关详情，请参见 7-1 页上的“数据连接通信”。

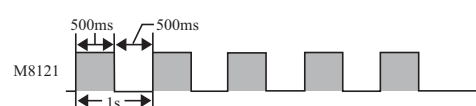
■ M8120: 初始化脉冲

仅打开运行（RUN）开始时的 1 次扫描。



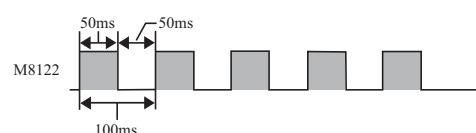
■ M8121: 1 秒时钟

M8001 关闭期间，M8121 将以 1s 为周期重复打开和关闭（占空比 1: 1）。



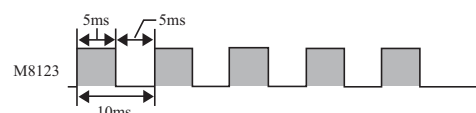
■ M8122: 100 毫秒时钟

M8122 以 100ms 为周期重复打开和关闭（占空比 1: 1）。



■ M8123: 10 毫秒时钟

M8123 以 10ms 为周期重复打开和关闭（占空比 1: 1）。



■ M8124: 定时器 / 计数器预置值更改

更改定时器 / 计数器的设置后，将打开。传送用户程序时或清除更改数据时，将关闭。

■ M8125: 运行中输出

处于运行状态期间，将打开。

■ M8126: 程序运行中下载完成后打开 1 次扫描

在运行状态下更改用户程序时（程序运行中下载）、下载完成后、开始运行更改后的用户程序时，将仅打开 1 次扫描。

2: 设备地址

■ M8137 ~ M8143、M8167: 中断输入状态

启用相应的用户中断时, 将打开。禁用用户中断时, 将关闭。

M8137 = 中断输入 I0 状态

M8140 = 中断输入 I1 状态

M8141 = 中断输入 I3 状态

M8142 = 中断输入 I4 状态

M8143 = 中断输入 I6 状态

M8167 = 中断输入 I7 状态

■ M8144: 定时器中断状态

启用定时器中断时, 将打开。禁用定时器中断时, 将关闭。

■ M8150 ~ M8152: 比较结果

M8150 ~ M8152 将根据 CMP= (比较等于) 指令、ICMP>= (间隔比较大于或等于) 指令的比较结果打开。有关详情, 请参见《梯形图编程手册》第 6 章中的“特殊内部继电器 M8150、M8151 和 M8152 用于 ICMP>=”。

■ M8153 ~ M8160: 捕捉输入开 / 关状态

在 1 次扫描期间检测捕捉输入中指定的输入接点上升沿 / 下降沿输入时, 将读取输入接点的状态。可检测的边沿为 1 个扫描中 1 次。

M8153 = 组 1/I0 的状态

M8154 = 组 2/I1 的状态

M8155 = 组 3/I3 的状态

M8156 = 组 4/I4 的状态

M8157 = 组 5/I6 的状态

M8160 = 组 6/I7 的状态

■ M8172 ~ M8175: 晶体管源型输出过电流检测

在 CPU 模块的晶体管保护源型输出中产生过电流输出时, 将打开特殊内部继电器 (M8172 ~ M8175)。并将输出 4 点作为 1 个组, 分配下一个特殊内部继电器。在任意一个特殊内部继电器中产生过电流输出时, 将打开。

即使过电流输出解除, 这些特殊内部继电器也不会恢复为关闭。如需恢复关闭状态, 请通过梯形图程序编程使其关闭。

M8172 = 组 1 (Q0 ~ Q3) 的状态

M8173 = 组 2 (Q4 ~ Q7) 的状态

M8174 = 组 3 (Q10 ~ Q13) 的状态

M8175 = 组 4 (Q14 ~ Q17) 的状态

■ M8184: HMI 模块 网络设置更改触发器

将 M8184 从关闭设为打开时, D8437 ~ D8456 中存储的值将设置为 HMI 模块的 IP 地址。

仅在 D8437 ~ D8456 的值有所更改时, 无法进行设置。有关 HMI 模块 网络设置更改的详情, 请参见第 3-6 页上的“由 HMI 模块的特殊数据寄存器进行的网络设置”。

■ M8185: 在夏时制时间段

夏时制功能启用时、在夏时制时间段时将打开。在夏时制时间段之外时将关闭。

夏时制功能禁用时将关闭。

■ M8186: 以太网端口 1 自动 Ping 执行中

以太网端口 1 的自动 Ping 运行过程中将打开。自动 Ping 停止时将关闭。有关自动 Ping 的详情, 请参见第 3-23 页上的“自动 Ping 功能”。

■ M8187: 以太网端口 1 自动 Ping 停止标记

M8187 打开期间, 将停止以太网端口 1 的自动 Ping。M8187 关闭期间, 则执行自动 Ping。此时, 无论上次的结束状态如何, 自动 Ping 都将从远程主机列表中指定的远程主机编号的最小编号开始执行。

■ M8190: CPU 模块 以太网端口 1 网络设置更改触发器

将 M8190 从关闭设为打开时, D8303 ~ D8323 中存储的值将设置为 CPU 模块以太网端口 1 的 IP 设置 /DNS 设置。

仅在 D8303 ~ D8323 的值有所更改时, 无法进行设置。有关 CPU 模块以太网端口 1 的 IP 设置 /DNS 设置更改的详情, 请参见第 3-4 页上的“由特殊数据寄存器进行的网络设置”。

■ **M8191: SNTP 获取标记**

将 M8191 从关闭设为打开时, 将从 SNTP 服务器获取时间信息。

■ **M8192 ~ M8197: 中断输入边沿**

在中断输入的上升沿发生中断时, 将打开。在中断输入的下降沿发生中断时, 将关闭。

M8192 = 中断输入 I0 边沿

M8193 = 中断输入 I3 边沿

M8194 = 中断输入 I4 边沿

M8195 = 中断输入 I6 边沿

M8196 = 中断输入 I7 边沿

M8197 = 中断输入 I1 边沿

■ **M8200 ~ M8207、M8334 ~ M8343: 用户通信接收指令取消标记**

将 M8200 ~ M8207 或 M8334 ~ M8343 从关闭设为打开时, 将中断正在执行的用户通信接收指令。

M8200 = 正在客户端连接 1 中执行的用户通信接收指令

M8201 = 正在客户端连接 2 中执行的用户通信接收指令

M8202 = 正在客户端连接 3 中执行的用户通信接收指令

M8203 = 正在客户端连接 4 中执行的用户通信接收指令

M8204 = 正在客户端连接 5 中执行的用户通信接收指令

M8205 = 正在客户端连接 6 中执行的用户通信接收指令

M8206 = 正在客户端连接 7 中执行的用户通信接收指令

M8207 = 正在客户端连接 8 中执行的用户通信接收指令

M8334 = 正在客户端连接 9 中执行的用户通信接收指令

M8335 = 正在客户端连接 10 中执行的用户通信接收指令

M8336 = 正在客户端连接 11 中执行的用户通信接收指令

M8337 = 正在客户端连接 12 中执行的用户通信接收指令

M8340 = 正在客户端连接 13 中执行的用户通信接收指令

M8341 = 正在客户端连接 14 中执行的用户通信接收指令

M8342 = 正在客户端连接 15 中执行的用户通信接收指令

M8343 = 正在客户端连接 16 中执行的用户通信接收指令

■ **M8211: 初始化 HMI 模块电子邮件发送服务器设置**

将 M8211 从关闭设为打开时, 将初始化 HMI- 以太网端口的电子邮件发送服务器设置。

■ **M8212 ~ M8221、M8345 ~ M8354: 连接状态**

通过维护通信服务器、用户通信服务器 / 客户端、Modbus TCP 服务器 / 客户端连接网络设备期间, 连接状态将打开。未连接期间, 将关闭。

M8212 = 连接 1

M8213 = 连接 2

M8214 = 连接 3

M8215 = 连接 4

M8216 = 连接 5

M8217 = 连接 6

M8220 = 连接 7

M8221 = 连接 8

M8345 = 连接 9

M8346 = 连接 10

M8347 = 连接 11

M8350 = 连接 12

M8351 = 连接 13

M8352 = 连接 14

M8353 = 连接 15

M8354 = 连接 16

2: 设备地址

■ M8222 ~ M8231、M8355 ~ M8364: 断开用户通信连接

通过用户通信进行连接时，如果将 M8222 ~ M8231 从关闭设为打开，将断开相应的连接。

M8222 = 连接 1
M8223 = 连接 2
M8224 = 连接 3
M8225 = 连接 4
M8226 = 连接 5
M8227 = 连接 6
M8230 = 连接 7
M8231 = 连接 8
M8355 = 连接 9
M8356 = 连接 10
M8357 = 连接 11
M8360 = 连接 12
M8361 = 连接 13
M8362 = 连接 14
M8363 = 连接 15
M8364 = 连接 16

仅会在使用用户通信客户端期间启用。

■ M8232: HMI 模块连接信息参考 连接状态

在 D8429 中指定的连接编号将在有连接时打开。在无连接时，将关闭。

■ M8250: 从 SD 记忆卡下载执行标记

将 M8250 从关闭设为打开时，将从 SD 记忆卡中下载 ZLD 文件。要下载的文件为以 autoexec.ini 文件指定的 ZLD 文件。

■ M8251: 向 SD 记忆卡上传执行标记

将 M8251 从关闭设为打开时，将向 SD 记忆卡中上传 ZLD 文件。
以通过 autoexec.ini 文件指定的文件名创建 ZLD 文件。

■ M8252: SD 记忆卡下载执行中

开始从 SD 记忆卡下载时将打开，完成后将关闭。

■ M8253: SD 记忆卡上传执行中

开始向 SD 记忆卡中上传时将打开，完成后将关闭。

■ M8254: SD 记忆卡下载 / 上传执行完成输出

开始从 SD 记忆卡下载或向 SD 记忆卡上传时 M8254 将关闭，下载 / 上传完成后则打开。

■ M8255: SD 记忆卡下载 / 上传执行错误输出

从 SD 记忆卡下载或向 SD 记忆卡上传完成时，将进行更新。D8255（下载、上传执行状态）为 0 以外数值时，将打开。

■ M8260: 写入配方执行标记

从关闭设为打开时，对 D8260（配方块编号）中指定的所有频道进行配方写入。

■ M8261: 读取配方执行标记

从关闭设为打开时，对 D8260（配方块编号）中指定的所有频道进行配方读出。

■ M8262: 写入配方执行中

在开始写入配方处理时将打开，完成后将关闭。在开始读取配方处理时也将关闭。

■ M8263: 读取配方执行中

在开始读取配方处理时将打开，完成后将关闭。在开始写入配方处理时也将关闭。

■ M8264: 配方频道执行完成输出

在开始读写配方时将关闭，读写配方完成后则打开。

■ M8265: 执行配方错误输出

执行配方完成时，如果 D8264（配方执行状态）为 0 以外数值，将打开。有关配方的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

- **M8266: 配方块执行完成输出**
配方块读写开始时关闭, 配方块读写完成时打开。
- **M8267: 配方内存 (ROM- 区域) 读取限制**
向内存 (ROM- 区域) 完成配方块读取时打开。M8267 打开时, 无法将配方读取到内存 (ROM- 区域 1) 和内存 (ROM- 区域 2) 中。读取配方时, 请关闭 M8267。
- **M8271: 从 SD 记忆卡进行 MQTT 基本设置下载的执行标记**
将 M8271 从关闭设为打开时, 下载 MQTT 基本设置用文件。有关 MQTT 基本设置用文件的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。
- **M8272: 从 SD 记忆卡进行 MQTT 基本设置下载的执行完成输出**
在开始下载 MQTT 基本设置用文件时关闭, 结束后打开。
- **M8273: 从 SD 记忆卡进行 MQTT 基本设置下载的错误输出**
在开始下载 MQTT 基本设置用文件的处理时关闭。在处理过程中发生错误时打开。
- **M8300 ~ M8304: J1939 通信**
用于 J1939 通信的特殊数据寄存器。有关详情, 请参见第 8-1 页上的“J1939 通信”。
- **M8311 ~ M8314: 键输入状态**
同时按下 HMI 模块的 ESC 键和方向键期间, 将打开。未按下键时, 将关闭。
M8311 = ESC 键+上 (⬆) 键
M8312 = ESC 键+下 (⬇) 键
M8313 = ESC 键+左 (⬅) 键
M8314 = ESC 键+右 (➡) 键
- **M8320: 初始化增设扩展模块组合型主机 / 从机**
将 M8320 设为打开时, 将初始化增设扩展模块组合型主机 / 从机及增设扩展模块组合型从机上所连接的扩展模块。初始化结束后, 将自动恢复关闭, 且增设扩展模块组合型从机重新开始 I/O 刷新。
- **M8331: 以太网端口 2 自动 Ping 执行中**
以太网端口 2 的自动 Ping 运行过程中将打开。自动 Ping 停止时将关闭。有关自动 Ping 的详情, 请参见第 3-23 页上的“自动 Ping 功能”。
- **M8332: 以太网端口 2 自动 Ping 停止标记**
M8332 打开期间, 将停止以太网端口 2 的自动 Ping。M8332 关闭期间, 则执行自动 Ping。此时, 无论上次的结束状态如何, 自动 Ping 都将从远程主机列表中指定的远程主机编号的最小编号开始执行。
- **M8333: CPU 模块 以太网端口 2 网络设置更改触发器**
将 M8333 从关闭设为打开时, D8630 ~ D8650 中存储的值将设置为 CPU 模块以太网端口 2 的 IP 设置 /DNS 设置。仅在 D8630 ~ D8650 的值有所更改时, 无法进行设置。有关 CPU 模块以太网端口 2 的 IP 设置 /DNS 设置更改的详情, 请参见第 3-4 页上的“由特殊数据寄存器进行的网络设置”。
- **M8344: 以太网端口 1 初始化电子邮件发送服务器设置**
将 M8344 从关闭设为打开时, 将初始化以太网端口 1 的电子邮件发送服务器设置。
- **M8401: 从 SD 记忆卡进行用于服务器功能的文件下载执行标记**
将 M8401 打开时, 将下载服务器功能文件。有关服务器功能文件的详细信息, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“SD 记忆卡”。
- **M8402: 从 SD 记忆卡进行用于服务器功能的文件下载执行完成输出**
当服务器功能文件开始下载时关闭, 完成服务器功能文件的下载后打开。
- **M8403: 从 SD 记忆卡进行用于服务器功能的文件下载错误输出**
当服务器功能文件开始下载时关闭, 如果在处理过程中发生错误, 则打开。

2: 设备地址

■ M8404: Plus CPU 模块 Web 服务器启动状态

当 Plus CPU 模块 Web 服务器开启完成时打开。在以下情况下关闭。

- Plus CPU 模块通电时
- 开始下载用户程序时
- 开始下载系统软件时
- 开始下载服务器功能用文件时

注释: 更改 D8303 (CPU 模块以太网端口 1 IP 设置 / DNS 设置切换) 也不会关闭 M8404。

■ M8450: BACnet 通信许可

M8450 打开期间, 进行 BACnet 通信。

■ M8460: EtherNet/IP 通信许可

启用 / 禁用 EtherNet/IP 通信。

OFF: 禁用 EtherNet/IP 通信

ON: 启用 EtherNet/IP 通信

特殊数据寄存器

特殊数据寄存器设备地址



警告 请勿改写特殊数据寄存器一览中写为保留的区域内的数据。否则系统可能无法正常工作。

注释: 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明	刷新时间	读 / 写
D8000	输入点数	I/O 初始化时	读
D8001	输出点数	I/O 初始化时	读
D8002	CPU 模块类型信息	通电时	读
D8003 D8004	— 保留 —	—	—
D8005	通用错误代码	发生错误时	读 / 写
D8006	用户程序执行错误代码	发生错误时	读
D8007	— 保留 —	—	—
D8008	日历 / 时钟 当前值 (只读)	年	每 500ms
D8009		月	每 500ms
D8010		日	每 500ms
D8011		星期	每 500ms
D8012		小时	每 500ms
D8013		分钟	每 500ms
D8014		秒	每 500ms
D8015	日历 / 时钟 预置值 (只写)	年	—
D8016		月	—
D8017		日	—
D8018		星期	—
D8019		小时	—
D8020		分钟	—
D8021		秒	—
D8022	扫描时间数据	固定扫描时间预置值	—
D8023		扫描时间 (当前值)	每次扫描
D8024		扫描时间 (最大值)	出现时
D8025		扫描时间 (最小值)	出现时
D8026	通信模式信息 (端口 1 ~ 3)	每次扫描	读
D8027 D8028	— 保留 —	—	—
D8029	系统软件版本	通电时	读
D8030	通信盒信息	通电时	读
D8031	选项连接信息	通电时	读
D8032	中断输入跳转目标标签编号 (I1)	—	读 / 写
D8033	中断输入跳转目标标签编号 (I3)	—	读 / 写
D8034	中断输入跳转目标标签编号 (I4)	—	读 / 写
D8035	中断输入跳转目标标签编号 (I6)	—	读 / 写
D8036	定时器中断跳转目标标签编号	—	读 / 写
D8037	输入输出模块连接台数	I/O 初始化时	读
D8038 D8039	— 保留 —	—	—

2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8040	从机编号 (端口 4)	—	读 / 写
D8041	从机编号 (端口 5)	—	读 / 写
D8042	从机编号 (端口 6)	—	读 / 写
D8043	从机编号 (端口 7)	—	读 / 写
D8044	从机编号 (端口 8)	—	读 / 写
D8045	从机编号 (端口 9)	—	读 / 写
D8046- D8051	— 保留 —	—	—
D8052	J1939 通信错误代码	每次扫描	读 / 写
D8053- D8055	— 保留 —	—	—
D8056	电池电压	—	读
D8057	模拟量 (AI0)	每次扫描	读
D8058	内置模拟量输入 (AI1)	每次扫描	读
D8059	模拟量输入状态 AI0	每次扫描	读
D8060	模拟量输入状态 AI1	每次扫描	读
D8061- D8066	— 保留 —	—	—
D8067	背光亮起时间	—	读 / 写
D8068	— 保留 —	—	—
D8069	从机 1 通信状态 / 错误 (数据连接主机模式时) 从机通信状态 / 错误 (数据连接从机模式时)	发生错误时	读
D8070	从机 2 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8071	从机 3 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8072	从机 4 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8073	从机 5 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8074	从机 6 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8075	从机 7 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8076	从机 8 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8077	从机 9 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8078	从机 10 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8079	从机 11 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8080	从机 12 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8081	从机 13 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8082	从机 14 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8083	从机 15 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8084	从机 16 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8085	从机 17 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8086	从机 18 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8087	从机 19 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8088	从机 20 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8089	从机 21 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8090	从机 22 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8091	从机 23 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8092	从机 24 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8093	从机 25 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8094	从机 26 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8095	从机 27 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8096	从机 28 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8097	从机 29 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读

设备地址	说明		刷新时间	读/写	
D8098	从机 30 通信状态 / 错误 (数据连接主机模式时)		发生错误时	读	
D8099	从机 31 通信状态 / 错误 (数据连接主机模式时)		发生错误时	读	
D8100	从机编号 (端口 1)		—	读/写	
D8101	— 保留 —		—	—	
D8102	从机编号 (端口 2)		—	读/写	
D8103	从机编号 (端口 3)		—	读/写	
D8104	控制信号状态 (端口 1 ~ 5)		每次扫描	读	
D8105	RS232C DSR 控制信号状态 (端口 1 ~ 5)		每次扫描	读/写	
D8106	RS232C DTR 输出控制信号选项 (端口 1 ~ 5)		每次扫描	读/写	
D8107- D8119	— 保留 —		—	—	
D8120	HMI 模块信息	类型 ID/ 状态	—	读	
D8121		系统软件版本	—	读	
D8122	盒插槽 1 信息	类型 ID/ 状态	—	读	
D8123		系统软件版本	—	读	
D8124	盒插槽 2 信息	类型 ID/ 状态	—	读	
D8125		系统软件版本	—	读	
D8126	盒插槽 3 信息	类型 ID/ 状态	—	读	
D8127		系统软件版本	—	读	
D8128- D8169	— 保留 —		—	—	
D8170	模拟 I/O 盒输入输出 AI2/AQ2		每次扫描	读	
D8171	模拟 I/O 盒输入输出 AI3/AQ3		每次扫描	读	
D8172	模拟 I/O 盒状态 AI2/AQ2		每次扫描	读	
D8173	模拟 I/O 盒状态 AI3/AQ3		每次扫描	读	
D8174	模拟 I/O 盒输入输出 AI4/AQ4		每次扫描	读	
D8175	模拟 I/O 盒输入输出 AI5/AQ5		每次扫描	读	
D8176	模拟 I/O 盒状态 AI4/AQ4		每次扫描	读	
D8177	模拟 I/O 盒状态 AI5/AQ5		每次扫描	读	
D8178	模拟 I/O 盒输入输出 AI6/AQ6		每次扫描	读	
D8179	模拟 I/O 盒输入输出 AI7/AQ7		每次扫描	读	
D8180	模拟 I/O 盒状态 AI6/AQ6		每次扫描	读	
D8181	模拟 I/O 盒状态 AI7/AQ7		每次扫描	读	
D8182- D8191	— 保留 —		—	—	
D8192	高速计数器 (组 2/I1)	高位字	当前值 / 频率测量 (I1) 当前值	每次扫描	读
D8193		低位字		每次扫描	读
D8194		高位字	预置值	—	读/写
D8195		低位字		—	读/写
D8196		高位字	复位值	—	读/写
D8197		低位字		—	读/写
D8198	高速计数器 (组 6/I7)	高位字	当前值 / 频率测量 (I7) 当前值	每次扫描	读
D8199		低位字		每次扫描	读
D8200		高位字	预置值	—	读/写
D8201		低位字		—	读/写
D8202		高位字	复位值	—	读/写
D8203		低位字		—	读/写
D8204	控制信号状态 (端口 6 ~ 9)		每次扫描	读	
D8205	RS232C DSR 控制信号状态 (端口 6 ~ 9)		每次扫描	读/写	
D8206	RS232C DTR 输出控制信号选项 (端口 6 ~ 9)		每次扫描	读/写	

2: 设备地址

设备地址	说明		刷新时间	读 / 写	
D8207- D8209	— 保留 —		—	—	
D8210	高速计数器 (组 1/I0)	高位字	当前值 / 频率测量 (I0) 当前值	每次扫描	读
D8211		低位字		每次扫描	读
D8212		高位字	预置值	—	读 / 写
D8213		低位字		—	读 / 写
D8214	中断输入跳转目标标签编号 (I7)		—	读 / 写	
D8215	中断输入跳转目标标签编号 (I0)		—	读 / 写	
D8216	高速计数器 (组 1/I0)	高位字	复位值	—	读 / 写
D8217		低位字		—	读 / 写
D8218	高速计数器 (组 3/I3)	高位字	当前值 / 频率测量 (I3) 当前值	每次扫描	读
D8219		低位字		每次扫描	读
D8220		高位字	预置值	—	读 / 写
D8221		低位字		—	读 / 写
D8222	高速计数器 (组 4/I4)	高位字	当前值 / 频率测量 (I4) 当前值	每次扫描	读
D8223		低位字		每次扫描	读
D8224		高位字	预置值	—	读 / 写
D8225		低位字		—	读 / 写
D8226	高速计数器 (组 5/I6)	高位字	当前值 / 频率测量 (I6) 当前值	每次扫描	读
D8227		低位字		每次扫描	读
D8228		高位字	预置值	—	读 / 写
D8229		低位字		—	读 / 写
D8230 D8231	— 保留 —		—	—	
D8232	高速计数器 (组 5/I6)	高位字	复位值	—	读 / 写
D8233		低位字		—	读 / 写
D8234	高速计数器 (组 3/I3)	高位字	复位值	—	读 / 写
D8235		低位字		—	读 / 写
D8236	高速计数器 (组 4/I4)	高位字	复位值	—	读 / 写
D8237		低位字		—	读 / 写
D8238	— 保留 —		—	—	
D8239	绝对位置管理状态		每次扫描	读	
D8240	绝对位置计数 器 1	高位字	绝对位置	每次扫描	读
D8241		低位字		每次扫描	读
D8242	绝对位置计数 器 2	高位字	绝对位置	每次扫描	读
D8243		低位字		每次扫描	读
D8244	绝对位置计数 器 3	高位字	绝对位置	每次扫描	读
D8245		低位字		每次扫描	读
D8246	绝对位置计数 器 4	高位字	绝对位置	每次扫描	读
D8247		低位字		每次扫描	读
D8248 D8249	— 保留 —		—	—	
D8250	SD 记忆卡容量显示		每次扫描	读	
D8251	SD 记忆卡可用空间显示		每次扫描	读	
D8252 D8253	— 保留 —		—	—	
D8254	SD 记忆卡 下载 / 上传执行信息		处理完成时	读	
D8255	SD 记忆卡 下载 / 上传执行状态		处理完成时	读	
D8256	从 SD 记忆卡进行 MQTT 基本设置下载的 执行信息		处理完成时	读	
D8257	从 SD 记忆卡进行 MQTT 基本设置下载的 执行错误信息		处理完成时	读	

设备地址	说明		刷新时间	读 / 写
D8258- D8259	— 保留 —		—	—
D8260	配方块编号		—	读 / 写
D8261	执行配方块编号		执行配方完成时	读
D8262	执行配方频道编号		执行配方完成时	读
D8263	执行配方动作		执行配方完成时	读
D8264	执行配方状态		执行配方完成时	读
D8265	执行配方错误信息		执行配方完成时	读
D8266	配方内存 (ROM- 区域 1) 读取次数		执行配方完成时	读
D8267	配方内存 (ROM- 区域 2) 读取次数		执行配方完成时	读
D8268	连接 1 的远程主机号 (1 ~ 255)		—	读 / 写
D8269	连接 2 的远程主机号 (1 ~ 255)		—	读 / 写
D8270	连接 3 的远程主机号 (1 ~ 255)		—	读 / 写
D8271	连接 4 的远程主机号 (1 ~ 255)		—	读 / 写
D8272	连接 5 的远程主机号 (1 ~ 255)		—	读 / 写
D8273	连接 6 的远程主机号 (1 ~ 255)		—	读 / 写
D8274	连接 7 的远程主机号 (1 ~ 255)		—	读 / 写
D8275	连接 8 的远程主机号 (1 ~ 255)		—	读 / 写
D8276 D8277	— 保留 —		—	—
D8278	通信模式信息	连接 1 ~ 4	—	读
D8279		连接 5 ~ 8	—	读
D8280- D8283	— 保留 —		—	—
D8284	通信模式信息 (HMI 连接)	HMI 连接 1 ~ 4	—	读
D8285		HMI 连接 5 ~ 8	—	读
D8286- D8302	— 保留 —		—	—
D8303	CPU 模块以太网端口 1 IP 设置 / DNS 设置切换		—	读 / 写
D8304	CPU 模块以太网端口 1 IP 地址 (只写)		—	写
D8305			—	写
D8306			—	写
D8307			—	写
D8308			—	写
D8309	CPU 模块以太网端口 1 子网掩码 (只写)		—	写
D8310			—	写
D8311			—	写
D8312	CPU 模块以太网端口 1 默认网关 (只写)		—	写
D8313			—	写
D8314			—	写
D8315			—	写
D8316	CPU 模块以太网端口 1 首选 DNS 服务器 (只写)		—	写
D8317			—	写
D8318			—	写
D8319			—	写
D8320	CPU 模块以太网端口 1 备用 DNS 服务器 (只写)		—	写
D8321			—	写
D8322			—	写
D8323			—	写

2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8324	CPU 模块以太网端口 1 MAC 地址 (只读当前值)	每 1 秒	读
D8325		每 1 秒	读
D8326		每 1 秒	读
D8327		每 1 秒	读
D8328		每 1 秒	读
D8329		每 1 秒	读
D8330	CPU 模块以太网端口 1 IP 地址 (只读当前值)	每 1 秒	读
D8331		每 1 秒	读
D8332		每 1 秒	读
D8333		每 1 秒	读
D8334	CPU 模块以太网端口 1 子网掩码 (只读当前值)	每 1 秒	读
D8335		每 1 秒	读
D8336		每 1 秒	读
D8337		每 1 秒	读
D8338	CPU 模块以太网端口 1 默认网关 (只读当前值)	每 1 秒	读
D8339		每 1 秒	读
D8340		每 1 秒	读
D8341		每 1 秒	读
D8342	CPU 模块以太网端口 1 首选 DNS 服务器 (只读当前值)	每 1 秒	读
D8343		每 1 秒	读
D8344		每 1 秒	读
D8345		每 1 秒	读
D8346	CPU 模块以太网端口 1 备用 DNS 服务器 (只读当前值)	每 1 秒	读
D8347		每 1 秒	读
D8348		每 1 秒	读
D8349		每 1 秒	读
D8350	连接 1 连接的 IP 地址	每 1 秒	读
D8351		每 1 秒	读
D8352		每 1 秒	读
D8353	连接 2 连接的 IP 地址	每 1 秒	读
D8354		每 1 秒	读
D8355		每 1 秒	读
D8356		每 1 秒	读
D8357	连接 3 连接的 IP 地址	每 1 秒	读
D8358		每 1 秒	读
D8359		每 1 秒	读
D8360		每 1 秒	读
D8361	连接 4 连接的 IP 地址	每 1 秒	读
D8362		每 1 秒	读
D8363		每 1 秒	读
D8364		每 1 秒	读
D8365	连接 5 连接的 IP 地址	每 1 秒	读
D8366		每 1 秒	读
D8367		每 1 秒	读
D8368		每 1 秒	读
D8369	连接 6 连接的 IP 地址	每 1 秒	读
D8370		每 1 秒	读
D8371		每 1 秒	读
D8372		每 1 秒	读
D8373	连接 6 连接的 IP 地址	每 1 秒	读

设备地址	说明	刷新时间	读 / 写	
D8374	连接 7 连接的 IP 地址	每 1 秒	读	
D8375		每 1 秒	读	
D8376		每 1 秒	读	
D8377		每 1 秒	读	
D8378	连接 8 连接的 IP 地址	每 1 秒	读	
D8379		每 1 秒	读	
D8380		每 1 秒	读	
D8381		每 1 秒	读	
D8382	HMI 模块 MAC 地址 (只读当前值)	每 1 秒	读	
D8383		每 1 秒	读	
D8384		每 1 秒	读	
D8385		每 1 秒	读	
D8386		每 1 秒	读	
D8387	HMI 模块 IP 地址 (只读当前值)	每 1 秒	读	
D8388		每 1 秒	读	
D8389		每 1 秒	读	
D8390		每 1 秒	读	
D8391	HMI 模块子网掩码 (只读当前值)	每 1 秒	读	
D8392		每 1 秒	读	
D8393		每 1 秒	读	
D8394		每 1 秒	读	
D8395	HMI 模块默认网关 (只读当前值)	每 1 秒	读	
D8396		每 1 秒	读	
D8397		每 1 秒	读	
D8398		每 1 秒	读	
D8399	HMI 模块首选 DNS 服务器 (只读当前值)	每 1 秒	读	
D8400		每 1 秒	读	
D8401		每 1 秒	读	
D8402		每 1 秒	读	
D8403	HMI 模块备用 DNS 服务器 (只读当前值)	每 1 秒	读	
D8404		每 1 秒	读	
D8405		每 1 秒	读	
D8406		每 1 秒	读	
D8407		每 1 秒	读	
D8408- D8412	— 保留 —	—	—	
D8413	时区偏移量	—	读 / 写	
D8414	SNTP 运行状态	—	读	
D8415	SNTP 访问经过时间	—	读	
D8416- D8428	— 保留 —	—	—	
D8429	HMI 模块连接信息参考	连接编号	—	
D8430		连接 IP 地址	每 1 秒	读 / 写
D8431			每 1 秒	读
D8432			每 1 秒	读
D8433			每 1 秒	读
D8434	连接端口编号	每 1 秒	读	
D8435 D8436	— 保留 —	—	—	

2: 设备地址

设备地址	说明	刷新时间	读 / 写	
D8437	HMI 模块 IP 地址 (只写)	—	写	
D8438		—	写	
D8439		—	写	
D8440		—	写	
D8441	HMI 模块 子网掩码 (只写)	—	写	
D8442		—	写	
D8443		—	写	
D8444		—	写	
D8445	HMI 模块 默认网关 (只写)	—	写	
D8446		—	写	
D8447		—	写	
D8448		—	写	
D8449	HMI 模块 首选 DNS 服务器 (只写)	—	写	
D8450		—	写	
D8451		—	写	
D8452		—	写	
D8453	HMI 模块 备用 DNS 服务器 (只写)	—	写	
D8454		—	写	
D8455		—	写	
D8456		—	写	
D8457	EMAIL 指令详细错误信息 (HMI- 以太网端口)	—	读	
D8458- D8469	— 保留 —	—	—	
D8470	扩展模块 槽 1 信息	类型 ID/ 状态	—	读
D8471		系统软件版本 / 位置信息	—	读
D8472	扩展模块 槽 2 信息	类型 ID/ 状态	—	读
D8473		系统软件版本 / 位置信息	—	读
D8474	扩展模块 槽 3 信息	类型 ID/ 状态	—	读
D8475		系统软件版本 / 位置信息	—	读
D8476	扩展模块 槽 4 信息	类型 ID/ 状态	—	读
D8477		系统软件版本 / 位置信息	—	读
D8478	扩展模块 槽 5 信息	类型 ID/ 状态	—	读
D8479		系统软件版本 / 位置信息	—	读
D8480	扩展模块 槽 6 信息	类型 ID/ 状态	—	读
D8481		系统软件版本 / 位置信息	—	读
D8482	扩展模块 槽 7 信息	类型 ID/ 状态	—	读
D8483		系统软件版本 / 位置信息	—	读
D8484	扩展模块 槽 8 信息	类型 ID/ 状态	—	读
D8485		系统软件版本 / 位置信息	—	读
D8486	扩展模块 槽 9 信息	类型 ID/ 状态	—	读
D8487		系统软件版本 / 位置信息	—	读
D8488	扩展模块 槽 10 信息	类型 ID/ 状态	—	读
D8489		系统软件版本 / 位置信息	—	读
D8490	扩展模块 槽 11 信息	类型 ID/ 状态	—	读
D8491		系统软件版本 / 位置信息	—	读
D8492	扩展模块 槽 12 信息	类型 ID/ 状态	—	读
D8493		系统软件版本 / 位置信息	—	读
D8494	扩展模块 槽 13 信息	类型 ID/ 状态	—	读
D8495		系统软件版本 / 位置信息	—	读

设备地址	说明	刷新时间	读 / 写	
D8496	扩展模块 槽 14 信息	类型 ID/ 状态	—	读
D8497		系统软件版本 / 位置信息	—	读
D8498	扩展模块 槽 15 信息	类型 ID/ 状态	—	读
D8499		系统软件版本 / 位置信息	—	读
D8500	扩展模块 槽 16 信息	类型 ID/ 状态	—	读
D8501		系统软件版本 / 位置信息	—	读
D8502	扩展模块 槽 17 信息	类型 ID/ 状态	—	读
D8503		系统软件版本 / 位置信息	—	读
D8504	扩展模块 槽 18 信息	类型 ID/ 状态	—	读
D8505		系统软件版本 / 位置信息	—	读
D8506	扩展模块 槽 19 信息	类型 ID/ 状态	—	读
D8507		系统软件版本 / 位置信息	—	读
D8508	扩展模块 槽 20 信息	类型 ID/ 状态	—	读
D8509		系统软件版本 / 位置信息	—	读
D8510	扩展模块 槽 21 信息	类型 ID/ 状态	—	读
D8511		系统软件版本 / 位置信息	—	读
D8512	扩展模块 槽 22 信息	类型 ID/ 状态	—	读
D8513		系统软件版本 / 位置信息	—	读
D8514	扩展模块 槽 23 信息	类型 ID/ 状态	—	读
D8515		系统软件版本 / 位置信息	—	读
D8516	扩展模块 槽 24 信息	类型 ID/ 状态	—	读
D8517		系统软件版本 / 位置信息	—	读
D8518	扩展模块 槽 25 信息	类型 ID/ 状态	—	读
D8519		系统软件版本 / 位置信息	—	读
D8520	扩展模块 槽 26 信息	类型 ID/ 状态	—	读
D8521		系统软件版本 / 位置信息	—	读
D8522	扩展模块 槽 27 信息	类型 ID/ 状态	—	读
D8523		系统软件版本 / 位置信息	—	读
D8524	扩展模块 槽 28 信息	类型 ID/ 状态	—	读
D8525		系统软件版本 / 位置信息	—	读
D8526	扩展模块 槽 29 信息	类型 ID/ 状态	—	读
D8527		系统软件版本 / 位置信息	—	读
D8528	扩展模块 槽 30 信息	类型 ID/ 状态	—	读
D8529		系统软件版本 / 位置信息	—	读
D8530	扩展模块 槽 31 信息	类型 ID/ 状态	—	读
D8531		系统软件版本 / 位置信息	—	读
D8532	扩展模块 槽 32 信息	类型 ID/ 状态	—	读
D8533		系统软件版本 / 位置信息	—	读
D8534	扩展模块 槽 33 信息	类型 ID/ 状态	—	读
D8535		系统软件版本 / 位置信息	—	读
D8536	扩展模块 槽 34 信息	类型 ID/ 状态	—	读
D8537		系统软件版本 / 位置信息	—	读
D8538	扩展模块 槽 35 信息	类型 ID/ 状态	—	读
D8539		系统软件版本 / 位置信息	—	读
D8530	扩展模块 槽 36 信息	类型 ID/ 状态	—	读
D8541		系统软件版本 / 位置信息	—	读
D8542	扩展模块 槽 37 信息	类型 ID/ 状态	—	读
D8543		系统软件版本 / 位置信息	—	读
D8544	扩展模块 槽 38 信息	类型 ID/ 状态	—	读
D8545		系统软件版本 / 位置信息	—	读

2: 设备地址

设备地址	说明	刷新时间	读 / 写	
D8546	扩展模块 槽 39 信息	类型 ID/ 状态	—	读
D8547		系统软件版本 / 位置信息	—	读
D8548	扩展模块 槽 40 信息	类型 ID/ 状态	—	读
D8549		系统软件版本 / 位置信息	—	读
D8550	扩展模块 槽 41 信息	类型 ID/ 状态	—	读
D8551		系统软件版本 / 位置信息	—	读
D8552	扩展模块 槽 42 信息	类型 ID/ 状态	—	读
D8553		系统软件版本 / 位置信息	—	读
D8554	扩展模块 槽 43 信息	类型 ID/ 状态	—	读
D8555		系统软件版本 / 位置信息	—	读
D8556	扩展模块 槽 44 信息	类型 ID/ 状态	—	读
D8557		系统软件版本 / 位置信息	—	读
D8558	扩展模块 槽 45 信息	类型 ID/ 状态	—	读
D8559		系统软件版本 / 位置信息	—	读
D8560	扩展模块 槽 46 信息	类型 ID/ 状态	—	读
D8561		系统软件版本 / 位置信息	—	读
D8562	扩展模块 槽 47 信息	类型 ID/ 状态	—	读
D8563		系统软件版本 / 位置信息	—	读
D8564	扩展模块 槽 48 信息	类型 ID/ 状态	—	读
D8565		系统软件版本 / 位置信息	—	读
D8566	扩展模块 槽 49 信息	类型 ID/ 状态	—	读
D8567		系统软件版本 / 位置信息	—	读
D8568	扩展模块 槽 50 信息	类型 ID/ 状态	—	读
D8569		系统软件版本 / 位置信息	—	读
D8570	扩展模块 槽 51 信息	类型 ID/ 状态	—	读
D8571		系统软件版本 / 位置信息	—	读
D8572	扩展模块 槽 52 信息	类型 ID/ 状态	—	读
D8573		系统软件版本 / 位置信息	—	读
D8574	扩展模块 槽 53 信息	类型 ID/ 状态	—	读
D8575		系统软件版本 / 位置信息	—	读
D8576	扩展模块 槽 54 信息	类型 ID/ 状态	—	读
D8577		系统软件版本 / 位置信息	—	读
D8578	扩展模块 槽 55 信息	类型 ID/ 状态	—	读
D8579		系统软件版本 / 位置信息	—	读
D8580	扩展模块 槽 56 信息	类型 ID/ 状态	—	读
D8581		系统软件版本 / 位置信息	—	读
D8582	扩展模块 槽 57 信息	类型 ID/ 状态	—	读
D8583		系统软件版本 / 位置信息	—	读
D8584	扩展模块 槽 58 信息	类型 ID/ 状态	—	读
D8585		系统软件版本 / 位置信息	—	读
D8586	扩展模块 槽 59 信息	类型 ID/ 状态	—	读
D8587		系统软件版本 / 位置信息	—	读
D8588	扩展模块 槽 60 信息	类型 ID/ 状态	—	读
D8589		系统软件版本 / 位置信息	—	读
D8590	扩展模块 槽 61 信息	类型 ID/ 状态	—	读
D8591		系统软件版本 / 位置信息	—	读
D8592	扩展模块 槽 62 信息	类型 ID/ 状态	—	读
D8593		系统软件版本 / 位置信息	—	读
D8594	扩展模块 槽 63 信息	类型 ID/ 状态	—	读
D8595		系统软件版本 / 位置信息	—	读

设备地址	说明	刷新时间	读/写	
D8596	增设扩展模块组合型主机槽信息	类型 ID/ 状态	—	读
D8597		系统软件版本 / 增设扩展模块组合型从机的连接台数	—	读
D8598	增设扩展模块组合型从机 (节点 1) 槽信息	类型 ID/ 状态	—	读
D8599		系统软件版本 / 扩展模块的连接信息	—	读
D8600	增设扩展模块组合型从机 (节点 2) 槽信息	类型 ID/ 状态	—	读
D8601		系统软件版本 / 扩展模块的连接信息	—	读
D8602	增设扩展模块组合型从机 (节点 3) 槽信息	类型 ID/ 状态	—	读
D8603		系统软件版本 / 扩展模块的连接信息	—	读
D8604	增设扩展模块组合型从机 (节点 4) 槽信息	类型 ID/ 状态	—	读
D8605		系统软件版本 / 扩展模块的连接信息	—	读
D8606	增设扩展模块组合型从机 (节点 5) 槽信息	类型 ID/ 状态	—	读
D8607		系统软件版本 / 扩展模块的连接信息	—	读
D8608	增设扩展模块组合型从机 (节点 6) 槽信息	类型 ID/ 状态	—	读
D8609		系统软件版本 / 扩展模块的连接信息	—	读
D8610	增设扩展模块组合型从机 (节点 7) 槽信息	类型 ID/ 状态	—	读
D8611		系统软件版本 / 扩展模块的连接信息	—	读
D8612	增设扩展模块组合型从机 (节点 8) 槽信息	类型 ID/ 状态	—	读
D8613		系统软件版本 / 扩展模块的连接信息	—	读
D8614	增设扩展模块组合型从机 (节点 9) 槽信息	类型 ID/ 状态	—	读
D8615		系统软件版本 / 扩展模块的连接信息	—	读
D8616	增设扩展模块组合型从机 (节点 10) 槽信息	类型 ID/ 状态	—	读
D8617		系统软件版本 / 扩展模块的连接信息	—	读
D8618	增设扩展模块组合型 I/O 刷新时间 当前值	每次扫描	读	
D8619	增设扩展模块组合型 I/O 刷新时间 最大值	每次扫描	读	
D8620- D8629	— 保留 —	—	—	
D8630	CPU 模块以太网端口 2 IP 设置 / DNS 设置切换	—	读 / 写	
D8631	CPU 模块 以太网端口 2 IP 地址 (只写)	—	写	
D8632		—	写	
D8633		—	写	
D8634		—	写	
D8635	CPU 模块 以太网端口 2 子网掩码 (只写)	—	写	
D8636		—	写	
D8637		—	写	
D8638		—	写	
D8639	CPU 模块 以太网端口 2 默认网关 (只写)	—	写	
D8640		—	写	
D8641		—	写	
D8642		—	写	
D8643	CPU 模块 以太网端口 2 首选 DNS 服务器 (只写)	—	写	
D8644		—	写	
D8645		—	写	
D8646		—	写	
D8647	CPU 模块 以太网端口 2 备用 DNS 服务器 (只写)	—	写	
D8648		—	写	
D8649		—	写	
D8650		—	写	

2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8651	CPU 模块 以太网端口 2 MAC 地址 (只读当前值)	每 1 秒	读
D8652		每 1 秒	读
D8653		每 1 秒	读
D8654		每 1 秒	读
D8655		每 1 秒	读
D8656		每 1 秒	读
D8657	CPU 模块 以太网端口 2 IP 地址 (只读当前值)	每 1 秒	读
D8658		每 1 秒	读
D8659		每 1 秒	读
D8660		每 1 秒	读
D8661	CPU 模块 以太网端口 2 子网掩码 (只读当前值)	每 1 秒	读
D8662		每 1 秒	读
D8663		每 1 秒	读
D8664		每 1 秒	读
D8665	CPU 模块 以太网端口 2 默认网关 (只读当前值)	每 1 秒	读
D8666		每 1 秒	读
D8667		每 1 秒	读
D8668		每 1 秒	读
D8669	CPU 模块 以太网端口 2 首选 DNS 服务器 (只读当前值)	每 1 秒	读
D8670		每 1 秒	读
D8671		每 1 秒	读
D8672		每 1 秒	读
D8673	CPU 模块 以太网端口 2 备用 DNS 服务器 (只读当前值)	每 1 秒	读
D8674		每 1 秒	读
D8675		每 1 秒	读
D8676		每 1 秒	读
D8677	连接 9 连接的 IP 地址	每 1 秒	读
D8678		每 1 秒	读
D8679		每 1 秒	读
D8680	连接 10 连接的 IP 地址	每 1 秒	读
D8681		每 1 秒	读
D8682		每 1 秒	读
D8683		每 1 秒	读
D8684	连接 11 连接的 IP 地址	每 1 秒	读
D8685		每 1 秒	读
D8686		每 1 秒	读
D8687		每 1 秒	读
D8688	连接 12 连接的 IP 地址	每 1 秒	读
D8689		每 1 秒	读
D8690		每 1 秒	读
D8691		每 1 秒	读
D8692	连接 13 连接的 IP 地址	每 1 秒	读
D8693		每 1 秒	读
D8694		每 1 秒	读
D8695		每 1 秒	读
D8696	连接 14 连接的 IP 地址	每 1 秒	读
D8697		每 1 秒	读
D8698		每 1 秒	读
D8699		每 1 秒	读
D8700		每 1 秒	读

设备地址	说明	刷新时间	读/写
D8701	连接 15 连接的 IP 地址	每 1 秒	读
D8702		每 1 秒	读
D8703		每 1 秒	读
D8704		每 1 秒	读
D8705	连接 16 连接的 IP 地址	每 1 秒	读
D8706		每 1 秒	读
D8707		每 1 秒	读
D8708		每 1 秒	读
D8709- D8716	— 保留 —	—	—
D8717	控制信号状态 (端口 10 ~ 13)	每次扫描	读
D8718	RS232C DSR 控制信号状态 (端口 10 ~ 13)	每次扫描	读/写
D8719	RS232C DTR 输出控制信号选项 (端口 10 ~ 13)	每次扫描	读/写
D8720	控制信号状态 (端口 14 ~ 17)	每次扫描	读
D8721	RS232C DSR 控制信号状态 (端口 14 ~ 17)	每次扫描	读/写
D8722	RS232C DTR 输出控制信号选项 (端口 14 ~ 17)	每次扫描	读/写
D8723	控制信号状态 (端口 18 ~ 21)	每次扫描	读
D8724	RS232C DSR 控制信号状态 (端口 18 ~ 21)	每次扫描	读/写
D8725	RS232C DTR 输出控制信号选项 (端口 18 ~ 21)	每次扫描	读/写
D8726	控制信号状态 (端口 22 ~ 25)	每次扫描	读
D8727	RS232C DSR 控制信号状态 (端口 22 ~ 25)	每次扫描	读/写
D8728	RS232C DTR 输出控制信号选项 (端口 22 ~ 25)	每次扫描	读/写
D8729	控制信号状态 (端口 26 ~ 29)	每次扫描	读
D8730	RS232C DSR 控制信号状态 (端口 26 ~ 29)	每次扫描	读/写
D8731	RS232C DTR 输出控制信号选项 (端口 26 ~ 29)	每次扫描	读/写
D8732	控制信号状态 (端口 30 ~ 33)	每次扫描	读
D8733	RS232C DSR 控制信号状态 (端口 30 ~ 33)	每次扫描	读/写
D8734	RS232C DTR 输出控制信号选项 (端口 30 ~ 33)	每次扫描	读/写
D8735	从机编号 (端口 10)	—	读/写
D8736	从机编号 (端口 11)	—	读/写
D8737	从机编号 (端口 12)	—	读/写
D8738	从机编号 (端口 13)	—	读/写
D8739	从机编号 (端口 14)	—	读/写
D8740	从机编号 (端口 15)	—	读/写
D8741	从机编号 (端口 16)	—	读/写
D8742	从机编号 (端口 17)	—	读/写
D8743	从机编号 (端口 18)	—	读/写
D8744	从机编号 (端口 19)	—	读/写
D8745	从机编号 (端口 20)	—	读/写
D8746	从机编号 (端口 21)	—	读/写
D8747	从机编号 (端口 22)	—	读/写
D8748	从机编号 (端口 23)	—	读/写
D8749	从机编号 (端口 24)	—	读/写
D8750	从机编号 (端口 25)	—	读/写
D8751	从机编号 (端口 26)	—	读/写
D8752	从机编号 (端口 27)	—	读/写
D8753	从机编号 (端口 28)	—	读/写
D8754	从机编号 (端口 29)	—	读/写
D8755	从机编号 (端口 30)	—	读/写
D8756	从机编号 (端口 31)	—	读/写

2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8757	从机编号 (端口 32)	—	读 / 写
D8758	从机编号 (端口 33)	—	读 / 写
D8759	EMAIL 指令详细错误信息 (以太网端口 1)	—	读
D8760	通信模式信息	连接 9 ~ 12	读
D8761		连接 13 ~ 16	读
D8762- D8773	— 保留 —	—	—
D8774	连接 9 的远程主机号 (1 ~ 255)	—	读 / 写
D8775	连接 10 的远程主机号 (1 ~ 255)	—	读 / 写
D8776	连接 11 的远程主机号 (1 ~ 255)	—	读 / 写
D8777	连接 12 的远程主机号 (1 ~ 255)	—	读 / 写
D8778	连接 13 的远程主机号 (1 ~ 255)	—	读 / 写
D8779	连接 14 的远程主机号 (1 ~ 255)	—	读 / 写
D8780	连接 15 的远程主机号 (1 ~ 255)	—	读 / 写
D8781	连接 16 的远程主机号 (1 ~ 255)	—	读 / 写
D8782	BACnet 运行状态	—	读
D8783	BACnet 错误信息	—	读
D8784- D8789	— 保留 —	—	—
D8790	EtherNet/IP 运行状态	—	读
D8791	EtherNet/IP 错误信息	—	读
D8792- D8819	— 保留 —	—	—
D8820	从 SD 记忆卡下载用于服务器功能文件执行信息	执行完成时	
D8821	从 SD 记忆卡下载用于服务器功能文件执行错误信息	执行完成时	
D8822	Web 服务器当前连接数 (Plus CPU 模块)	约每 1 秒	
D8823- D8899	— 保留 —	—	—

特殊数据寄存器补充说明

■ D8000: 输入点数

写入 FC6A 型的输入点数。
写入 CPU 模块和扩展模块的输入点数合计。

■ D8001: 输出点数

写入 FC6A 型的输出点数。
写入 CPU 模块和扩展模块的输出点数合计值。

■ D8002: CPU 模块类型信息

写入 CPU 模块的类型信息。
0 (00h): 16-I/O 型
1 (01h): 24-I/O 型
2 (02h): 40-I/O 型 (All-in-One CPU 模块)
18 (12h): 40-I/O 型 (CAN J1939 All-in-One CPU 模块)
32 (20h): Plus 16-I/O 型
33 (21h): Plus 32-I/O 型

■ D8005: 通用错误代码

写入 FC6A 型的通用错误代码。如果出现通用错误, 则会开启与该错误相对应的位。通过将“1”写入 D8005 的最高有效位, 可清除通用错误和用户程序执行错误。有关通用错误代码的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 13 章中的“通用错误代码”。

■ D8006: 用户程序执行错误代码

FC6A 型写入用户程序执行错误信息。发生用户程序执行错误时, 将写入与错误内容对应的错误代码。有关用户程序执行错误的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 13 章中的“用户程序执行错误代码”。

■ D8008 ~ D8021: 日历 / 时钟数据

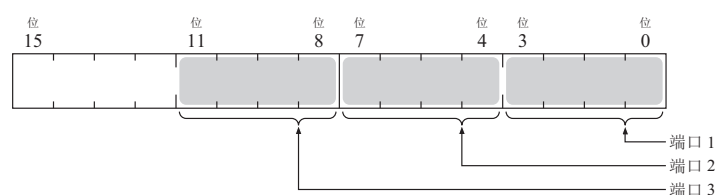
D8008 到 D8021 用于读取来自内部时钟的日历 / 时钟数据, 也用于将日历 / 时钟数据写入内部时钟。有关日历 / 时钟数据的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“时钟功能”。

■ D8022 ~ D8025: 扫描时间数据

D8022 到 D8025 都是特殊数据寄存器, 用于检查扫描时间和设置常量扫描时间。有关扫描时间的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“固定扫描时间”。

■ D8026: 通信模式信息 (端口 1 ~ 3)

表示端口 1 ~ 3 的通信模式。
设备内各通信端口的分配 (位分配) 如下所示。



0 (0000): 维护通信
1 (0001): 用户通信
2 (0010): Modbus RTU 主机
3 (0011): Modbus RTU 从机
4 (0100): 数据连接通信

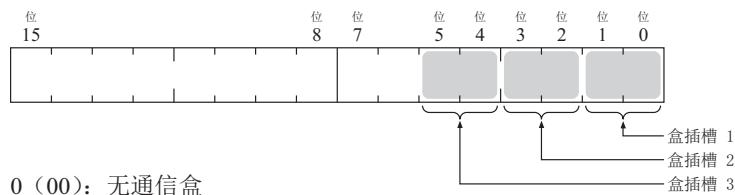
■ D8029: 系统版本信息

写入 CPU 模块的系统软件的版本号。

2: 设备地址

■ D8030: 通信盒信息

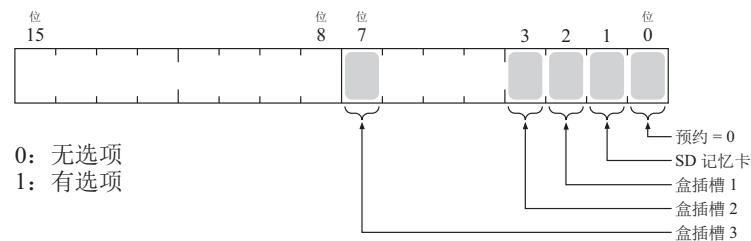
表示端口 2 及端口 3 上的通信盒连接状况。
设备内各通信端口的分配（位分配）如下所示。



- 0 (00): 无通信盒
- 1 (01): RS232C 通信盒连接
- 2 (10): RS485 通信盒连接
- 3 (11): Bluetooth 通信盒连接

■ D8031: 选项连接信息

写入选项连接信息。
设备内的分配（位分配）如下所示。



- 0: 无选项
- 1: 有选项

■ D8032 ~ D8035、D8214、D8215: 中断输入跳转目标标签编号

写入中断输入的跳转目标标签编号。如需使用中断输入，请写入与中断输入中分配的特殊数据寄存器相对应的标签编号。有关中断输入的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“中断输入”。

- D8032 = I1
- D8033 = I3
- D8034 = I4
- D8035 = I6
- D8214 = I7
- D8215 = I0

■ D8036: 定时器中断跳转目标标签编号

写入发生定时器中断时的跳转目标标签编号。如需使用定时器中断，请存储相应的标签编号。
有关定时器中断的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“定时器中断”。

■ D8037: 扩展模块连接台数

写入 CPU 模块所连接的扩展模块（I/O 模块、PID 模块及通信模块）的台数。

■ D8052: J1939 通信错误代码

J1939 通信中发生错误时，将写入错误代码。有关 J1939 通信错误代码的详情，请参见 8-5 页上的“J1939 通信错误代码（D8052）”。

■ D8056: 电池电压

以 1mV 为单位写入备份用电池的电池电压测量结果。

- 0: 测量错误

■ D8057: 模拟量 (AI0)

将模拟量值转换为数字值进行写入。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“模拟量”。

■ D8058: 内置模拟量输入 (AI1)

将模拟量输入的模拟量输入值（0 ~ 10V DC）转换为数字值进行写入。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“内置模拟量输入”。

■ D8059: 模拟量输入状态 AI0

写入模拟量的模拟量输入状态。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“模拟量”。

■ D8060: 模拟量输入状态 A11

写入模拟量输入的模拟量输入状态。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“内置模拟量输入”。

■ D8067: 背光亮起时间

写入 HMI 模块的背光亮起时间。背光亮起时间可通过更改 D8067 的值，在 1 ~ 65,535s 范围内进行设置。将 D8067 的值设为 0s 时，背光将始终亮起。背光亮起时间可在 HMI 模块的环境设置中进行更改。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章中的“设置 LCD 背光亮起时间”。

■ D8069 ~ D8099: 从机 (1 ~ 31) 通信状态 / 错误

用于数据连接通信的特殊数据寄存器。有关详情，请参见 7-4 页上的“主机站”。

■ D8040 ~ D8045、D8100、D8102、D8103、D8735 ~ D8758: 从机编号 (端口 1 ~ 33)

端口 1 ~ 33 的通信模式为维护通信、Modbus RTU 从机或数据连接通信时，将写入从机编号。通过在“功能设置”中更改相应设备的值，可更改从机编号。

D8100 = 端口 1 从机编号
D8102 = 端口 2 从机编号
D8103 = 端口 3 从机编号

D8040 = 端口 4 从机编号
D8041 = 端口 5 从机编号
D8042 = 端口 6 从机编号
D8043 = 端口 7 从机编号
D8044 = 端口 8 从机编号
D8045 = 端口 9 从机编号
D8735 = 端口 10 从机编号
D8736 = 端口 11 从机编号
D8737 = 端口 12 从机编号
D8738 = 端口 13 从机编号
D8739 = 端口 14 从机编号
D8740 = 端口 15 从机编号
D8741 = 端口 16 从机编号

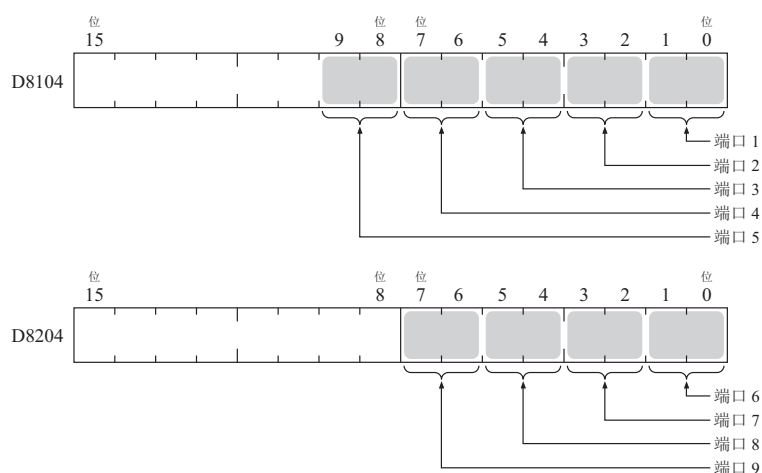
D8742 = 端口 17 从机编号
D8743 = 端口 18 从机编号
D8744 = 端口 19 从机编号
D8745 = 端口 20 从机编号
D8746 = 端口 21 从机编号
D8747 = 端口 22 从机编号
D8748 = 端口 23 从机编号
D8749 = 端口 24 从机编号
D8750 = 端口 25 从机编号
D8751 = 端口 26 从机编号
D8752 = 端口 27 从机编号
D8753 = 端口 28 从机编号
D8754 = 端口 29 从机编号
D8755 = 端口 30 从机编号
D8756 = 端口 31 从机编号
D8757 = 端口 32 从机编号
D8758 = 端口 33 从机编号

有关各通信模式的详情，请参见本手册中的以下内容。

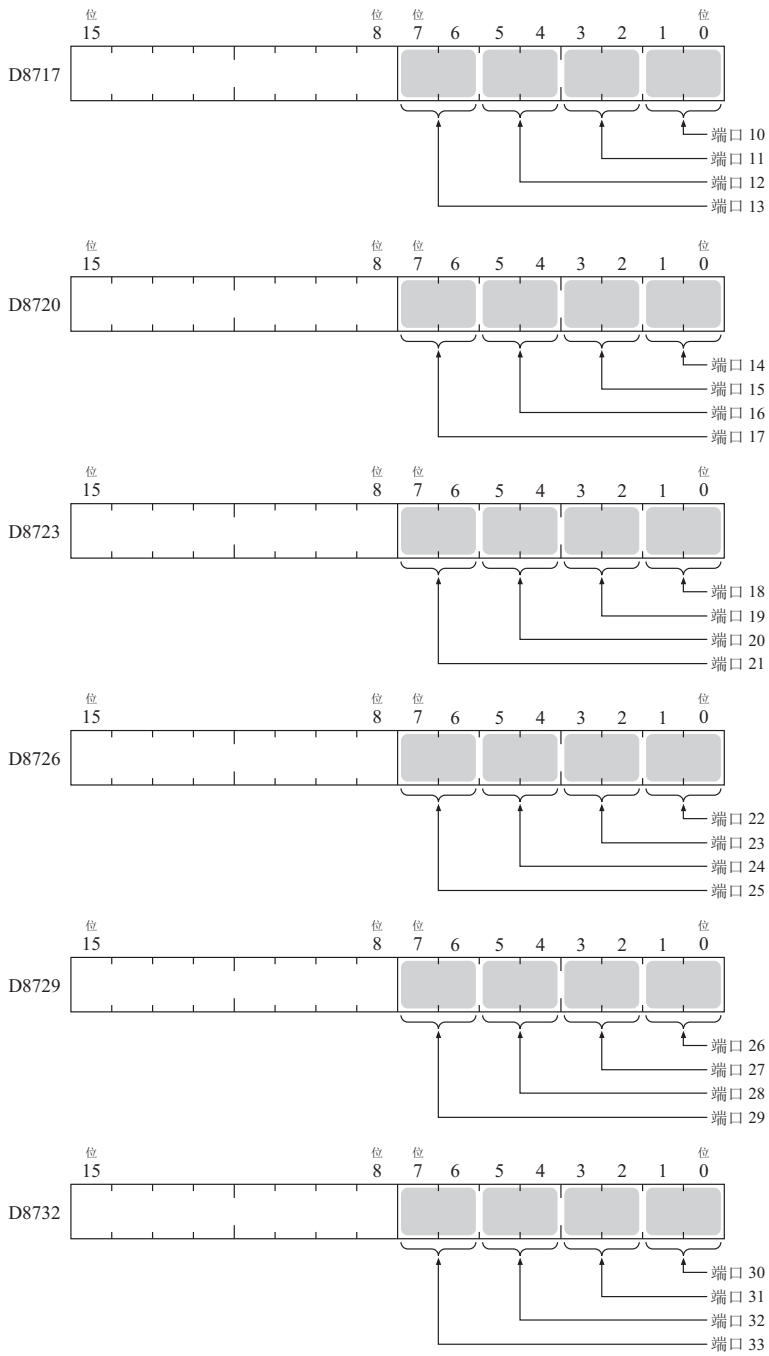
- 维护通信：4-1 页上的“维护通信”
- Modbus RTU 从机：6-1 页上的“Modbus 通信”
- 数据连接通信：7-1 页上的“数据连接通信”

■ D8104、D8204、D8717、D8720、D8723、D8726、D8729、D8732: 控制信号状态 (端口 1 ~ 33)

写入 DSR、DTR 的各控制线的信号状态。以停止中及运行中的 END 处理进行刷新。设备内各通信端口的分配 (位分配) 如下所示。



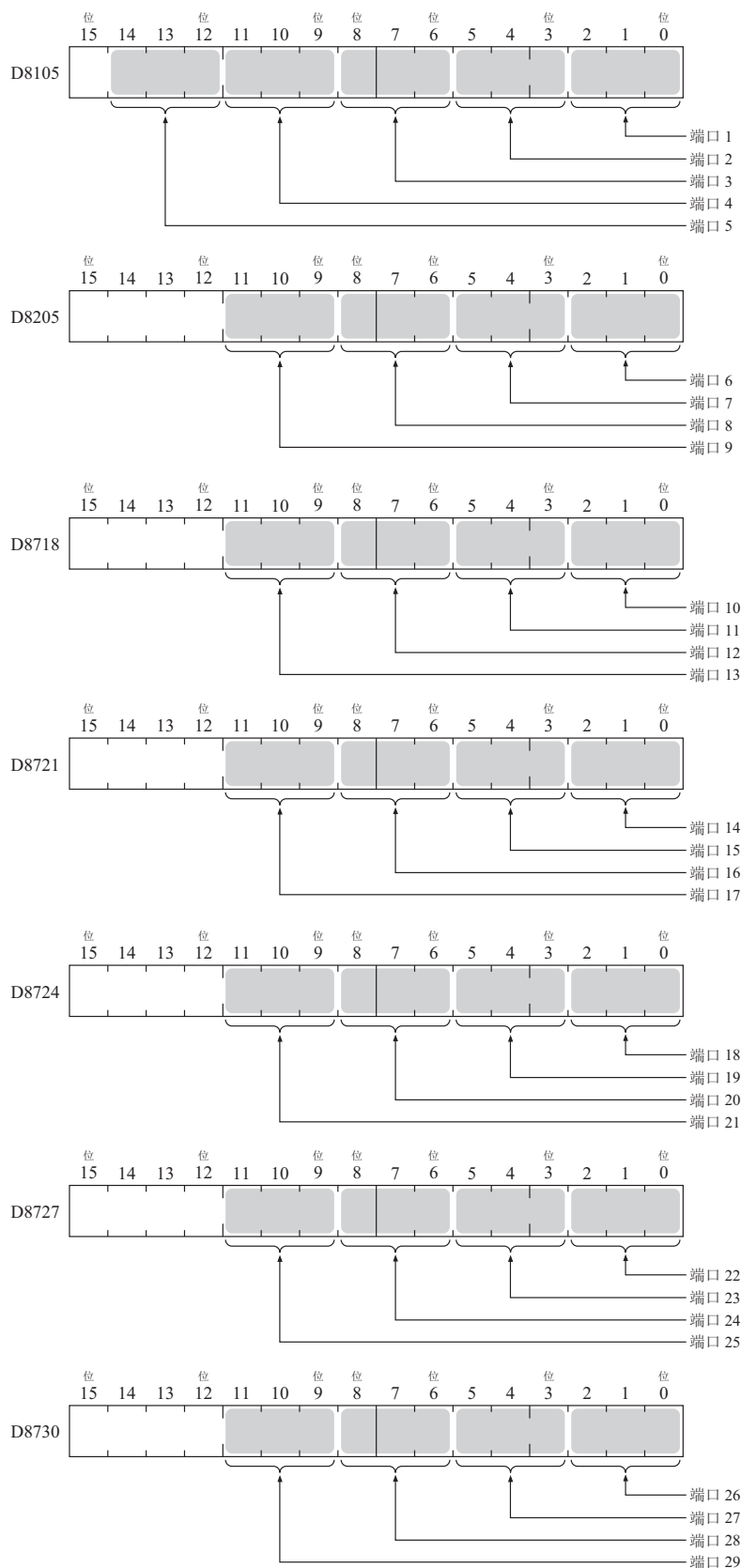
2: 设备地址



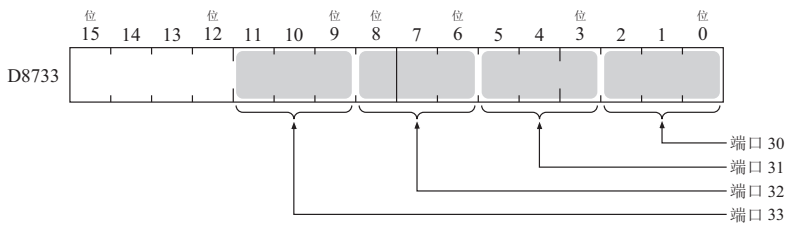
- 0 (00): DTR 和 DSR 皆为关闭。
- 1 (01): DTR 为关闭, DSR 为打开。
- 2 (10): DTR 为打开, DSR 为关闭。
- 3 (11): DTR 和 DSR 皆为打开。

■ D8105、D8205、D8718、D8721、D8724、D8727、D8730、D8733: RS232C DSR 控制信号状态 (端口 1 ~ 33)

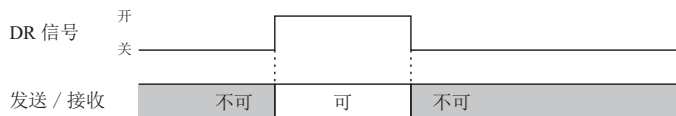
写入 DSR、DTR 的各控制线的信号状态。以停止中及运行中的 END 处理进行刷新。设备内各通信端口的分配 (位分配) 如下所示。



2: 设备地址



- 0 (000): FC6A 型的收发控制中不会使用 DSR 信号的状态。
如无需进行 DSR 信号控制, 请在该状态下使用。
- 1 (001): DSR 信号打开时, FC6A 型可进行收发。



- 2 (010): DSR 信号关闭时, FC6A 型可进行收发。

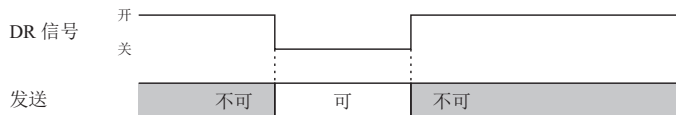


- 3 (011): DSR 信号打开时, 可进行发送 (始终可发送)。



这通常称为“Busy 控制”, 用于处理速度较慢的设备 (打印机等) 的发送控制。
(如果以连接设备立场来看, 则为输入数据的限制。)

- 4 (100): DSR 信号关闭时, 可进行发送。

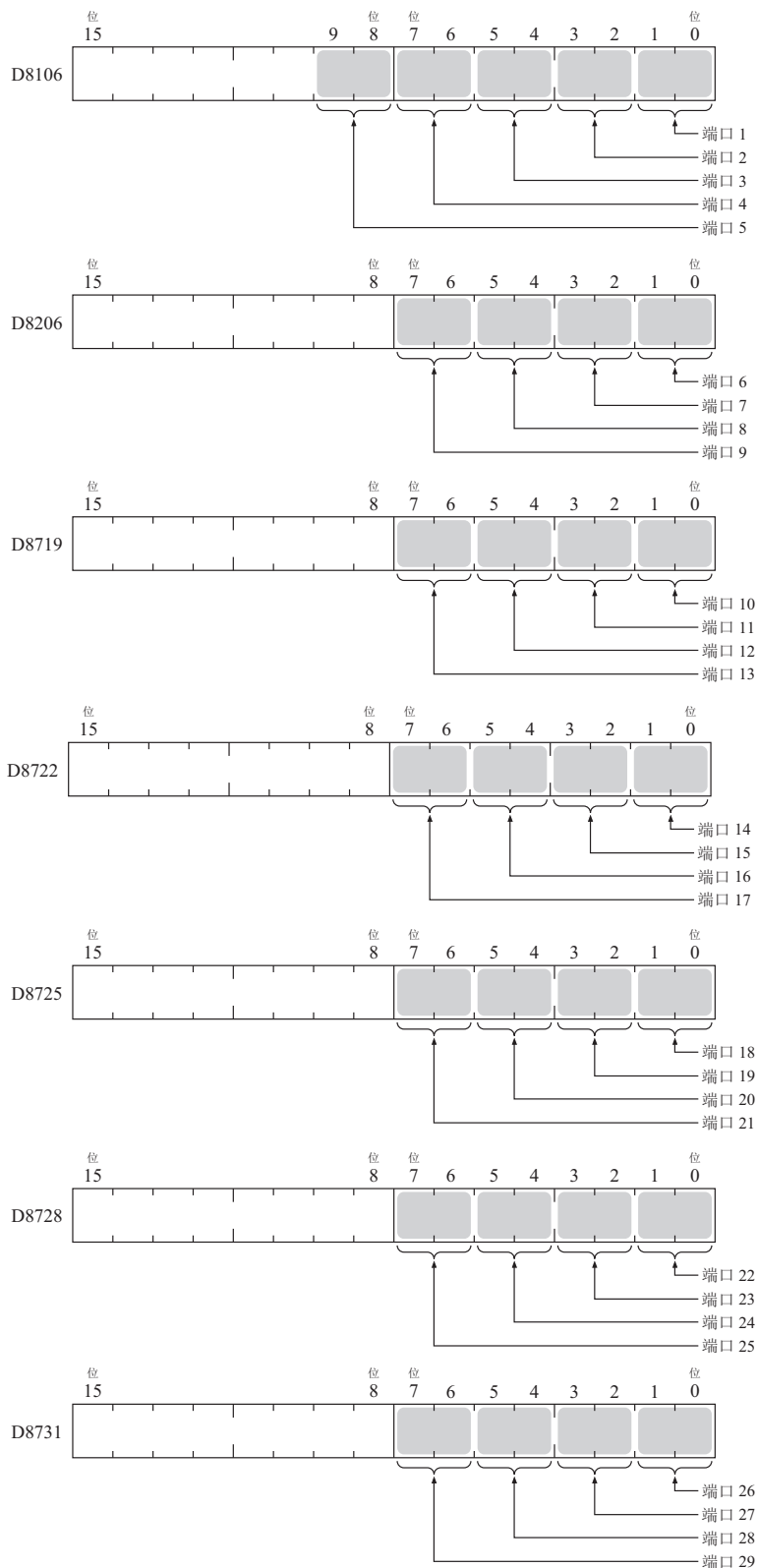


- 5 以上: 进行与预置值“000”相同的动作。

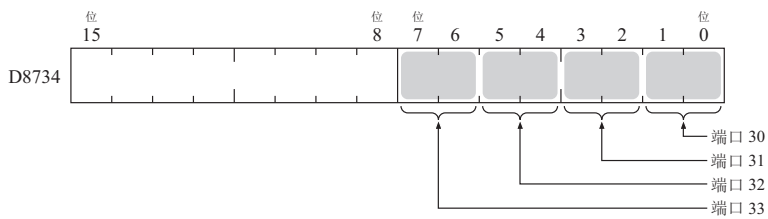
■ D8106、D8206、D8719、D8722、D8725、D8728、D8731、D8734: RS232C DTR 输出控制信号选项（端口 1 ~ 33）

用于在目标设备中表示 FC6A 型的控制状态或收发状态时。该控制线为从 FC6A 型到目标设备的输出信号。仅在用户通信时启用。

设备内各通信端口的分配（位分配）如下所示。



2: 设备地址



- 0 (00): 在 FC6A 型为运行状态时打开，为停止状态时关闭。运行中将始终打开，不受数据收发的影响。需要显示运行状态时进行设置。



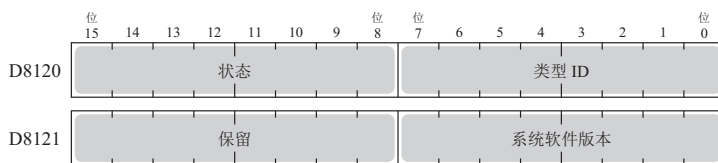
- 1 (01): 始终关闭。
- 2 (10): 对接收数据进行流程控制时进行设置。可从目标设备接收数据时，将打开。不可接收时，将关闭。



- 3 (11): 进行与预置值“0”相同的动作。

■ D8120、D8121: HMI 模块信息

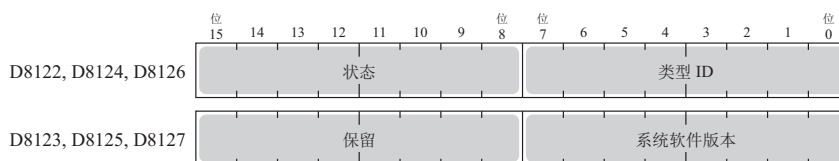
写入 HMI 模块的类型信息。
信息的分配（位分配）如下所示。



将在系统软件版本中显示向设备内写入的软件版本。
有关类型 ID 和状态的详情，请参见第 2-52 页上的“类型 ID、状态一览”。

■ D8122 ~ D8127: 盒插槽信息

写入盒插槽 1 ~ 3 的信息。
信息的分配（位分配）如下所示。



- D8122、D8123 = 盒插槽 1 信息
- D8124、D8125 = 盒插槽 2 信息
- D8126、D8127 = 盒插槽 3 信息

有关类型 ID 和状态的详情，请参见第 2-52 页上的“类型 ID、状态一览”。

■ D8170、D8171、D8174、D8175、D8178、D8179: 模拟 I/O 盒输入输出

写入模拟 I/O 盒的模拟量输入输出值。

模拟量输入型时: 将模拟 I/O 盒中输入的模拟量值转换为数字值进行写入。

模拟量输出型时: 将存储为数字值的数字值转换为模拟量值从模拟 I/O 盒进行输出。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

D8170 = AI2/AQ2

D8171 = AI3/AQ3

D8174 = AI4/AQ4

D8175 = AI5/AQ5

D8178 = AI6/AQ6

D8179 = AI7/AQ7

■ D8172、D8173、D8176、D8177、D8180、D8181: 模拟 I/O 盒状态

写入模拟 I/O 盒的模拟状态。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

D8172 = AI2/AQ2

D8173 = AI3/AQ3

D8176 = AI4/AQ4

D8177 = AI5/AQ5

D8180 = AI6/AQ6

D8181 = AI7/AQ7

■ D8192 ~ D8203、D8210 ~ D8213、D8216 ~ D8229、D8232 ~ D8237: 高速计数器

这些特殊数据寄存器用于高速计数器功能和频率测量功能。

有关高速计数器的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“高速计数器”。

D8210 ~ D8213、D8216、D8217 = 高速计数器 (组 1/10)

D8218 ~ D8221、D8234、D8235 = 高速计数器 (组 3/13)

D8222 ~ D8225、D8236、D8237 = 高速计数器 (组 4/14)

D8226 ~ D8229、D8232、D8233 = 高速计数器 (组 5/16)

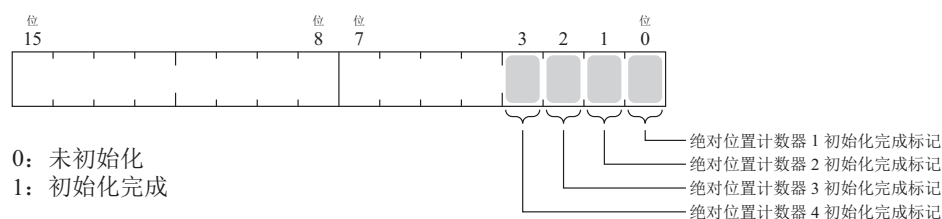
D8192 ~ D8197 = 高速计数器 (组 2/11)

D8198 ~ D8203 = 高速计数器 (组 6/17)

■ D8239: 绝对位置管理状态

表示绝对位置管理的状态。

设备内的绝对位置计数器初始化完成标记的分配 (位分配) 如下所示。有关绝对位置管理状态的详情, 请参见《梯形图编程手册》第 18 章中的“ABS (绝对位置设置)”。



■ D8240 ~ D8247: 绝对位置计数器 1 ~ 4

根据脉冲输出写入绝对位置。绝对位置可以 ABS 指令进行初始化。

以脉冲输出指令 (RAMP/ ARAMP/ ZRN/ JOG) 设置为有方向控制时, 将根据输出结果加减计算绝对位置。

有关绝对位置计数器的详

情, 请参见《梯形图编程手册》第 18 章中的“ABS (绝对位置设置)”。

D8240、D8241 = 绝对位置计数器 1

D8242、D8243 = 绝对位置计数器 2

D8244、D8245 = 绝对位置计数器 3

D8246、D8247 = 绝对位置计数器 4

■ D8250: SD 记忆卡容量显示

以 MB 为单位显示已识别 SD、SDHC (最大 32GB) 对应的 SD 记忆卡容量。

未插入 SD 记忆卡或记忆卡未被识别时, 将变为 0。

2: 设备地址

■ D8251: SD 记忆卡可用空间显示

以 MB 为单位显示已识别 SD 记忆卡的可用空间。
未插入 SD 记忆卡或记忆卡未被识别时, 将变为 0。

■ D8254: SD 记忆卡 下载 / 上传执行信息

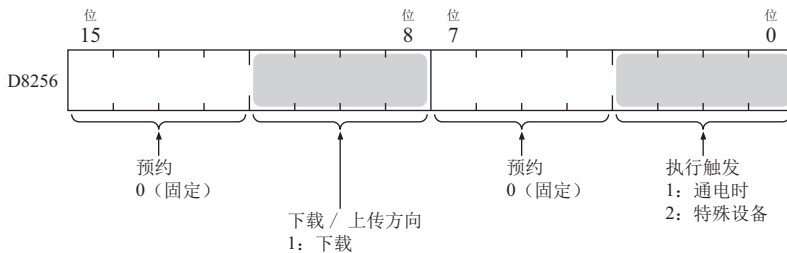
已执行的 SD 记忆卡 下载 / 上传的信息。
有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

■ D8255: SD 记忆卡 下载 / 上传执行状态

已执行的 SD 记忆卡 下载 / 上传的状态。
有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

■ D8256: 从 SD 记忆卡进行 MQTT 基本设置下载的执行信息

此为执行的 MQTT 基本设置用文件的下载信息。设备内的分配 (位分配) 如下所示。



■ D8257: 从 SD 记忆卡进行 MQTT 基本设置下载的执行错误信息

此为执行的 MQTT 基本设置用文件的下载错误信息。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》中的“下载 MQTT 基本设置”。

■ D8260: 配方块编号

使用特殊内部继电器进行读写的配方块编号。
已指定块编号的所有频道皆为目标。

■ D8261: 执行配方块编号

已执行配方的配方块编号。
在开始配方块时进行刷新, 完成后则维持状态。

■ D8262: 执行配方频道编号

已执行配方的频道编号。
在开始频道处理时进行刷新, 完成后则维持状态。

■ D8263: 执行配方动作

已执行配方动作的相关信息。
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

■ D8264: 执行配方状态

已执行配方的状态。
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

■ D8265: 执行配方错误信息

已执行配方的错误信息。
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

■ D8266: 配方内存 (ROM- 区域 1) 读取次数

将设备值作为配方数据读取到内存 (ROM- 区域 1) 中的次数以 100 单位进行存储。

■ D8267: 配方内存 (ROM- 区域 2) 读取次数

将设备值作为配方数据读取到内存 (ROM- 区域 2) 中的次数以 100 单位进行存储。

■ D8268 ~ D8275、D8774 ~ D8781: 远程主机号 1 ~ 255 (连接 1 ~ 16)

在设置用户通信客户端时更改通信目标的特殊数据寄存器。有关详情, 第 5-43 页上的“切换远程主机号”。

■ D8278、D8279、D8760、D8761: 通信模式信息 (连接 1 ~ 16)

D8278 = 表示连接 1 ~ 4 的通信模式。

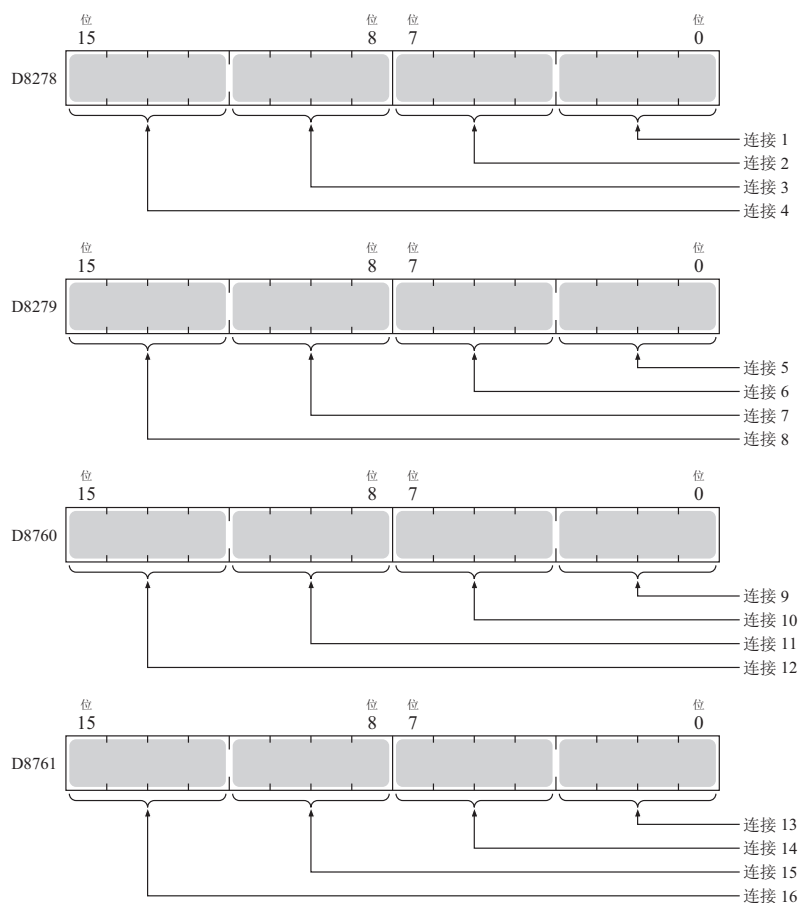
D8279 = 表示连接 5 ~ 8 的通信模式。

D8760 = 表示连接 9 ~ 12 的通信模式。

D8761 = 表示连接 13 ~ 16 的通信模式。

设备内各连接的分配 (位分配) 如下所示。

从连接数的最低位开始, 以 4 位为单位分配连接。



连接 (4 位) 如下所示。

- 客户端连接 (最高有效位 = 0)
 - 0000: 未使用
 - 0001: 用户通信
 - 0010: Modbus TCP 客户端
 - 0011: MC 协议客户端
 - 0100: 用户通信 UDP
- 服务器连接 (最高有效位 = 1)
 - 1000: 维护通信
 - 1001: 用户通信
 - 1010: Modbus TCP 服务器

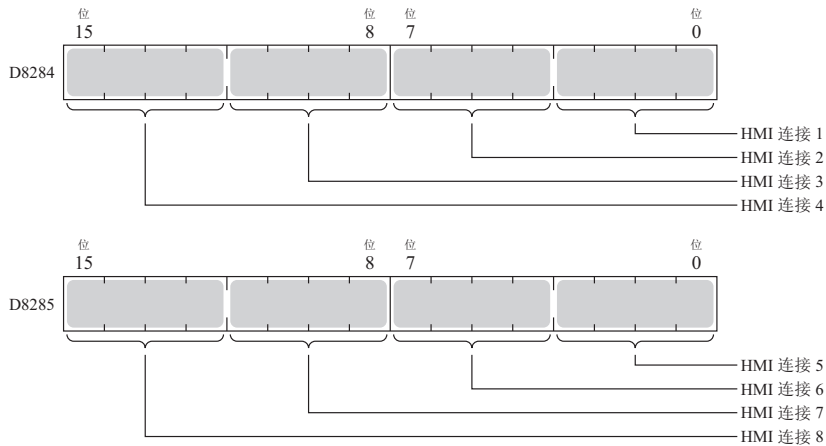
2: 设备地址

■ D8284、D8285: 通信模式信息 (HMI 连接) (HMI 连接 1 ~ 8)

D8284: 表示 HMI 连接 1 ~ 4 的通信模式。

D8285: 表示 HMI 连接 5 ~ 8 的通信模式。

设备内各连接的分配 (位分配) 如下所示。



- 客户端连接 (最高有效位 = 0)
 - 0000: 未使用
- 服务器连接 (最高有效位 = 1)
 - 1000: 维护通信

■ D8303: CPU 模块以太网端口 1 IP 设置 / DNS 设置切换

通过向 D8303 写入下页表中的预置值, 并将 M8190 切换到 ON, 可更改以太网端口 1 的 IP 设置 / DNS 设置。

如需使用本功能, 请在 WindLDR 的“功能设置”中启用以下项目。

- All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块
 - “功能设置”的“网络设置”中“启用 D8630 (IP 设置 / DNS 设置切换)”
- Plus CPU 模块
 - “功能设置”的“以太网端口 1”中“启用 D8630 (IP 设置 / DNS 设置切换)”

预置值的含义如下所示。

预置值	IP 设置 / DNS 设置
0	依照功能设置。
1	启用 DHCP。
2	依照特殊数据寄存器 (D8304 ~ D8323) 的设置。

■ D8304 ~ D8307: CPU 模块以太网端口 1 IP 地址 (只写)

用于写入 CPU 模块的 IP 地址。

IP 地址: 设置为 aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8304 = aaa、D8305 = bbb、D8306 = ccc、D8307 = ddd

■ D8308 ~ D8311: CPU 模块以太网端口 1 子网掩码 (只写)

用于写入 CPU 模块的子网掩码。

子网掩码: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8308 = aaa、D8309 = bbb、D8310 = ccc、D8311 = ddd

■ D8312 ~ D8315: CPU 模块以太网端口 1 默认网关 (只写)

用于写入 CPU 模块的默认网关。

默认网关: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8312 = aaa、D8313 = bbb、D8314 = ccc、D8315 = ddd

■ D8316 ~ D8319: CPU 模块以太网端口 1 首选 DNS 服务器 (只写)

用于写入 CPU 模块的首选 DNS 服务器。

首选 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8316 = aaa、D8317 = bbb、D8318 = ccc、D8319 = ddd

■ **D8320 ~ D8323: CPU 模块以太网端口 1 备用 DNS 服务器 (只写)**

用于写入 CPU 模块的备用 DNS 服务器。

备用 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8320 = aaa、D8321 = bbb、D8322 = ccc、D8323 = ddd

■ **D8324 ~ D8329: CPU 模块以太网端口 1 MAC 地址 (只读当前值)**

CPU 模块的 MAC 地址将如下所示写入到各特殊数据寄存器中。

例) MAC 地址: AA-BB-CC-DD-EE-FF

D8324=AA、D8325=BB、D8326=CC、D8327=DD、D8328=EE、D8329=FF

■ **D8330 ~ D8333: CPU 模块以太网端口 1 IP 地址 (只读当前值)**

CPU 模块的 IP 地址将如下所示写入到各特殊数据寄存器中。

例) 本机 IP 地址: aaa.bbb.ccc.ddd

D8330=aaa、D8331=bbb、D8332=ccc、D8333=ddd

■ **D8334 ~ D8337: CPU 模块以太网端口 1 子网掩码 (只读当前值)**

CPU 模块的子网掩码将如下所示写入到各特殊数据寄存器中。

例) 子网掩码: aaa.bbb.ccc.ddd

D8334=aaa、D8335=bbb、D8336=ccc、D8337=ddd

■ **D8338 ~ D8341: CPU 模块以太网端口 1 默认网关 (只读当前值)**

CPU 模块的默认网关将如下所示写入到各特殊数据寄存器中。

例) 默认网关: aaa.bbb.ccc.ddd

D8338=aaa、D8339=bbb、D8340=ccc、D8341=ddd

■ **D8342 ~ D8345: CPU 模块以太网端口 1 首选 DNS 服务器 (只读当前值)**

CPU 模块的首选 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) 首选 DNS 服务器: aaa.bbb.ccc.ddd

D8342=aaa、D8343=bbb、D8344=ccc、D8345=ddd

■ **D8346 ~ D8349: CPU 模块以太网端口 1 备用 DNS 服务器 (只读当前值)**

CPU 模块的备用 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) 备用 DNS 服务器: aaa.bbb.ccc.ddd

D8346=aaa、D8347=bbb、D8348=ccc、D8349=ddd

■ **D8350 ~ D8381、D8677 ~ D8708: 已接通从机的 IP 地址**

如下所示写入正在访问连接的目标设备的 IP 地址。

连接 1 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8350=aaa、D8351=bbb、D8352=ccc、D8353=ddd

连接 2 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8354=aaa、D8355=bbb、D8356=ccc、D8357=ddd

连接 3 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8358=aaa、D8359=bbb、D8360=ccc、D8361=ddd

连接 4 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8362=aaa、D8363=bbb、D8364=ccc、D8365=ddd

连接 5 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8366=aaa、D8367=bbb、D8368=ccc、D8369=ddd

连接 6 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8370=aaa、D8371=bbb、D8372=ccc、D8373=ddd

连接 7 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8374=aaa、D8375=bbb、D8376=ccc、D8377=ddd

连接 8 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8378=aaa、D8379=bbb、D8380=ccc、D8381=ddd

连接 9 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8677=aaa、D8678=bbb、D8679=ccc、D8680=ddd

连接 10 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8681=aaa、D8682=bbb、D8683=ccc、D8684=ddd

连接 11 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8685=aaa、D8686=bbb、D8687=ccc、D8688=ddd

连接 12 连接的 IP 地址: aaa.bbb.ccc.ddd 时

2: 设备地址

D8689=aaa、D8690=bbb、D8691=ccc、D8692=ddd
连接 13 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8693=aaa、D8694=bbb、D8695=ccc、D8696=ddd
连接 14 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8697=aaa、D8698=bbb、D8699=ccc、D8700=ddd
连接 15 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8701=aaa、D8702=bbb、D8703=ccc、D8704=ddd
连接 16 连接的 IP 地址: aaa.bbb.ccc.ddd 时

D8705=aaa、D8706=bbb、D8707=ccc、D8708=ddd

■ D8382 ~ D8387: HMI 模块 MAC 地址 (只读当前值)

MAC 地址将如下所示写入到各特殊数据寄存器中。

例) MAC 地址: AA-BB-CC-DD-EE-FF

D8382=AA、D8383=BB、D8384=CC、D8385=DD、D8386=EE、D8387=FF

■ D8388 ~ D8391: HMI 模块 IP 地址 (只读当前值)

HMI 模块的 IP 地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块 IP 地址: aaa.bbb.ccc.ddd

D8388=aaa、D8389=bbb、D8390=ccc、D8391=ddd

■ D8392 ~ D8395: HMI 模块子网掩码 (只读当前值)

HMI 模块的子网掩码的值将如下所示写入到各特殊数据寄存器中。

例) HMI 模块子网掩码: aaa.bbb.ccc.ddd

D8392=aaa、D8393=bbb、D8394=ccc、D8395=ddd

■ D8396 ~ D8399: HMI 模块默认网关 (只读当前值)

HMI 模块的默认网关的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块默认网关: aaa.bbb.ccc.ddd

D8396=aaa、D8397=bbb、D8398=ccc、D8399=ddd

■ D8400 ~ D8403: HMI 模块首选 DNS 服务器 (只读当前值)

HMI 模块的首选 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块首选 DNS 服务器: aaa.bbb.ccc.ddd

D8400=aaa、D8401=bbb、D8402=ccc、D8403=ddd

■ D8404 ~ D8407: HMI 模块备用 DNS 服务器 (只读当前值)

HMI 模块的备用 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块备用 DNS 服务器: aaa.bbb.ccc.ddd

D8404=aaa、D8405=bbb、D8406=ccc、D8407=ddd

■ D8413: 时区偏移量

可以 15 分为单位调整功能设置中设置的时区。

有关详情, 请参见第 3-11 页上的“SNTP 设置”。

■ D8414: SNTP 运行状态

写入通过操作 M8191 (SNTP 时间获取标记) 获取时间信息、或自动获取时间信息时的动作状态。

有关详情, 请参见第 3-11 页上的“SNTP 设置”。

■ D8415: SNTP 访问经过时间

以分为单位写入自最后从 SNTP 服务器获取时间信息起的经过时间。

有关详情, 请参见第 3-11 页上的“SNTP 设置”。

■ D8429: HMI 模块连接信息参考连接编号

将已指定连接编号的连接信息反映到 D8430 ~ D8434 中。写入 0 时, 将向 D8430 ~ D8434 中写入 0。即使指定了不存在的连接编号, 也会向 D8430 ~ D8434 中写入 0。

■ D8430 ~ D8433: HMI 模块连接信息参考连接 IP 地址

正在访问连接的终端的 IP 地址将如下所示写入到各特殊数据寄存器中。

例) 读取目标的 IP 地址: aaa.bbb.ccc.ddd

D8430=aaa、D8431=bbb、D8432=ccc、D8433=ddd

■ **D8434: HMI 模块连接信息参考 连接端口编号**

写入正在访问连接的终端的端口编号。

■ **D8437 ~ D8440: HMI 模块 IP 地址 (只写)**

用于写入 HMI 模块的 IP 地址。

HMI 模块 IP 地址: 设置为 aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8437 = aaa、D8438 = bbb、D8439 = ccc、D8440 = ddd

■ **D8441 ~ D8444: HMI 模块子网掩码 (只写)**

用于写入 HMI 模块的子网掩码。

HMI 模块子网掩码: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8441 = aaa、D8442 = bbb、D8443 = ccc、D8444 = ddd

■ **D8445 ~ D8448: HMI 模块默认网关 (只写)**

用于写入 HMI 模块的默认网关。

HMI 模块默认网关: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8445 = aaa、D8446 = bbb、D8447 = ccc、D8448 = ddd

■ **D8449 ~ D8452: HMI 模块首选 DNS 服务器 (只写)**

用于写入 HMI 模块的首选 DNS 服务器。

HMI 模块首选 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8449 = aaa、D8450 = bbb、D8451 = ccc、D8452 = ddd

■ **D8453 ~ D8456: HMI 模块备用 DNS 服务器 (只写)**

用于写入 HMI 模块的备用 DNS 服务器。

HMI 模块备用 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8453 = aaa、D8454 = bbb、D8455 = ccc、D8456 = ddd

■ **D8457: EMAIL 指令详细错误信息 (HMI- 以太网端口)**

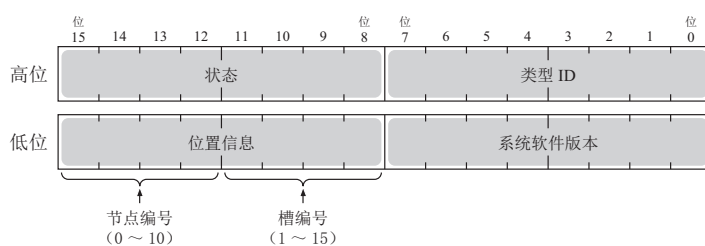
写入 EMAIL 指令的详细错误信息。

有关详情, 请参见第 12-3 页上的“确认 EMAIL 指令的错误详情”。

■ **D8470 ~ D8595: 扩展模块槽信息**

写入扩展模块的类型信息。

信息的分配 (位分配) 如下所示。



高位	低位	
D8470、D8471		= 扩展模块槽 1 信息
D8472、D8473		= 扩展模块槽 2 信息
D8474、D8475		= 扩展模块槽 3 信息
D8476、D8477		= 扩展模块槽 4 信息
D8478、D8479		= 扩展模块槽 5 信息
D8480、D8481		= 扩展模块槽 6 信息
D8482、D8483		= 扩展模块槽 7 信息
D8484、D8485		= 扩展模块槽 8 信息
D8486、D8487		= 扩展模块槽 9 信息
D8488、D8489		= 扩展模块槽 10 信息
D8490、D8491		= 扩展模块槽 11 信息
D8492、D8493		= 扩展模块槽 12 信息
D8494、D8495		= 扩展模块槽 13 信息
D8496、D8497		= 扩展模块槽 14 信息

2: 设备地址

高位	低位	
D8498、	D8499	=扩展模块槽 15 信息
D8500、	D8501	=扩展模块槽 16 信息
D8502、	D8503	=扩展模块槽 17 信息
D8504、	D8505	=扩展模块槽 18 信息
D8506、	D8507	=扩展模块槽 19 信息
D8508、	D8509	=扩展模块槽 20 信息
D8510、	D8511	=扩展模块槽 21 信息
D8512、	D8513	=扩展模块槽 22 信息
D8514、	D8515	=扩展模块槽 23 信息
D8516、	D8517	=扩展模块槽 24 信息
D8518、	D8519	=扩展模块槽 25 信息
D8520、	D8521	=扩展模块槽 26 信息
D8522、	D8523	=扩展模块槽 27 信息
D8524、	D8525	=扩展模块槽 28 信息
D8526、	D8527	=扩展模块槽 29 信息
D8528、	D8529	=扩展模块槽 30 信息
D8530、	D8531	=扩展模块槽 31 信息
D8532、	D8533	=扩展模块槽 32 信息
D8534、	D8535	=扩展模块槽 33 信息
D8536、	D8537	=扩展模块槽 34 信息
D8538、	D8539	=扩展模块槽 35 信息
D8540、	D8541	=扩展模块槽 36 信息
D8542、	D8543	=扩展模块槽 37 信息
D8544、	D8545	=扩展模块槽 38 信息
D8546、	D8547	=扩展模块槽 39 信息
D8548、	D8549	=扩展模块槽 40 信息
D8550、	D8551	=扩展模块槽 41 信息
D8552、	D8553	=扩展模块槽 42 信息
D8554、	D8555	=扩展模块槽 43 信息
D8556、	D8557	=扩展模块槽 44 信息
D8558、	D8559	=扩展模块槽 45 信息
D8560、	D8561	=扩展模块槽 46 信息
D8562、	D8563	=扩展模块槽 47 信息
D8564、	D8565	=扩展模块槽 48 信息
D8566、	D8567	=扩展模块槽 49 信息
D8568、	D8569	=扩展模块槽 50 信息
D8570、	D8571	=扩展模块槽 51 信息
D8572、	D8573	=扩展模块槽 52 信息
D8574、	D8575	=扩展模块槽 53 信息
D8576、	D8577	=扩展模块槽 54 信息
D8578、	D8579	=扩展模块槽 55 信息
D8580、	D8581	=扩展模块槽 56 信息
D8582、	D8583	=扩展模块槽 57 信息
D8584、	D8585	=扩展模块槽 58 信息
D8586、	D8587	=扩展模块槽 59 信息
D8588、	D8589	=扩展模块槽 60 信息
D8590、	D8591	=扩展模块槽 61 信息
D8592、	D8593	=扩展模块槽 62 信息
D8594、	D8595	=扩展模块槽 63 信息

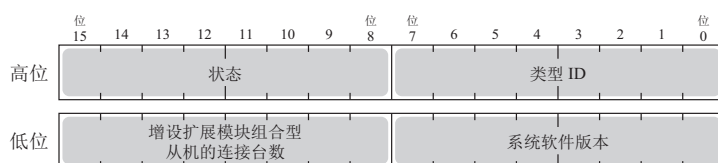
将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-52 页上的“类型 ID、状态一览”。

■ D8596、D8597：增设扩展模块组合型主机槽信息

写入增设扩展模块组合型主机的类型信息。

信息的分配（位分配）如下所示。



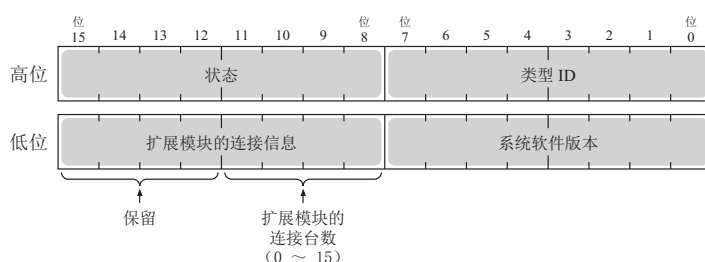
将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-52 页上的“类型 ID、状态一览”。

■ D8598 ~ D8617：增设扩展模块组合型从机槽信息

写入增设扩展模块组合型从机的类型信息。

信息的分配（位分配）如下所示。



高位 低位

- D8598、D8599 = 增设扩展模块组合型从机（第 1 台）槽信息
- D8600、D8601 = 增设扩展模块组合型从机（第 2 台）槽信息
- D8602、D8603 = 增设扩展模块组合型从机（第 3 台）槽信息
- D8604、D8605 = 增设扩展模块组合型从机（第 4 台）槽信息
- D8606、D8607 = 增设扩展模块组合型从机（第 5 台）槽信息
- D8608、D8609 = 增设扩展模块组合型从机（第 6 台）槽信息
- D8610、D8611 = 增设扩展模块组合型从机（第 7 台）槽信息
- D8612、D8613 = 增设扩展模块组合型从机（第 8 台）槽信息
- D8614、D8615 = 增设扩展模块组合型从机（第 9 台）槽信息
- D8616、D8617 = 增设扩展模块组合型从机（第 10 台）槽信息

将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-52 页上的“类型 ID、状态一览”。

■ D8618：增设扩展模块组合型 I/O 刷新时间 当前值

连接增设扩展模块组合型以后的扩展模块的 I/O 刷新时间的当前值会以 1ms 为单位进行存储。

■ D8619：增设扩展模块组合型 I/O 刷新时间 最大值

连接增设扩展模块组合型以后的扩展模块的 I/O 刷新时间的最大值会以 1ms 为单位进行存储。

■ D8630：CPU 模块以太网端口 2 IP 设置 / DNS 设置切换

通过向 D8630 写入下表中的预置值，并将 M8333 切换到 ON，可更改以太网端口 2 的 IP 设置 / DNS 设置。

如需使用本功能，请在 WindLDR 的“功能设置”中启用“以太网端口 2”的“启用 D8630（IP 设置 / DNS 设置切换）”。

预置值的含义如下所示。

预置值	IP 设置 / DNS 设置
0	依照功能设置。
1	启用 DHCP。
2	依照特殊数据寄存器（D8631 ~ D8650）的设置。

2: 设备地址

- **D8631 ~ D8634: CPU 模块 以太网端口 2 IP 地址 (只写)**
用于写入 Plus CPU 模块以太网端口 2 的 IP 地址。
IP 地址: 设置为 aaa.bbb.ccc.ddd 时, 将进行如下写入。
D8631 = aaa、D8632 = bbb、D8633 = ccc、D8634 = ddd
- **D8635 ~ D8638: CPU 模块 以太网端口 2 子网掩码 (只写)**
用于写入 Plus CPU 模块以太网端口 2 的子网掩码。
子网掩码: aaa.bbb.ccc.ddd 时, 将进行如下写入。
D8635 = aaa、D8636 = bbb、D8637 = ccc、D8638 = ddd
- **D8639 ~ D8642: CPU 模块 以太网端口 2 默认网关 (只写)**
用于写入 Plus CPU 模块以太网端口 2 的默认网关。
默认网关: aaa.bbb.ccc.ddd 时, 将进行如下写入。
D8639 = aaa、D8640 = bbb、D8641 = ccc、D8642 = ddd
- **D8643 ~ D8646: CPU 模块 以太网端口 2 首选 DNS 服务器 (只写)**
用于写入 Plus CPU 模块以太网端口 2 的首选 DNS 服务器。
首选 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。
D8643 = aaa、D8644 = bbb、D8645 = ccc、D8646 = ddd
- **D8647 ~ D8650: CPU 模块 以太网端口 2 备用 DNS 服务器 (只写)**
用于写入 Plus CPU 模块以太网端口 2 的备用 DNS 服务器。
备用 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。
D8647 = aaa、D8648 = bbb、D8649 = ccc、D8650 = ddd
- **D8651 ~ D8656: CPU 模块 以太网端口 2 MAC 地址 (只读当前值)**
Plus CPU 模块以太网端口 2 的 MAC 地址将如下所示写入到各特殊数据寄存器中。
例) MAC 地址: AA-BB-CC-DD-EE-FF
D8651 = AA、D8652 = BB、D8653 = CC、D8654 = DD、D8655 = EE、D8656 = FF
- **D8657 ~ D8660: CPU 模块 以太网端口 2 IP 地址 (只读当前值)**
Plus CPU 模块以太网端口 2 的 IP 地址将如下所示写入到各特殊数据寄存器中。
例) 本机 IP 地址: aaa.bbb.ccc.ddd
D8657 = aaa、D8658 = bbb、D8659 = ccc、D8660 = ddd
- **D8661 ~ D8664: CPU 模块 以太网端口 2 子网掩码 (只读当前值)**
Plus CPU 模块以太网端口 2 的子网掩码值将如下所示写入到各特殊数据寄存器中。
例) 子网掩码: aaa.bbb.ccc.ddd
D8661 = aaa、D8662 = bbb、D8663 = ccc、D8664 = ddd
- **D8665 ~ D8668: CPU 模块 以太网端口 2 默认网关 (只读当前值)**
Plus CPU 模块以太网端口 2 的默认网关地址将如下所示写入到各特殊数据寄存器中。
例) 默认网关: aaa.bbb.ccc.ddd
D8665 = aaa、D8666 = bbb、D8667 = ccc、D8668 = ddd
- **D8669 ~ D8672: CPU 模块 以太网端口 2 首选 DNS 服务器 (只读当前值)**
Plus CPU 模块以太网端口 2 的首选 DNS 服务器地址将如下所示写入到各特殊数据寄存器中。
例) 首选 DNS 服务器: aaa.bbb.ccc.ddd
D8669 = aaa、D8670 = bbb、D8671 = ccc、D8672 = ddd
- **D8673 ~ D8676: CPU 模块 以太网端口 2 备用 DNS 服务器 (只读当前值)**
Plus CPU 模块以太网端口 2 的备用 DNS 服务器地址将如下所示写入到各特殊数据寄存器中。
例) 备用 DNS 服务器: aaa.bbb.ccc.ddd
D8673 = aaa、D8674 = bbb、D8675 = ccc、D8676 = ddd

■ **D8759: EMAIL 指令详细错误信息 (以太网端口 1)**

写入 EMAIL 指令 (以太网端口 1) 的详细错误信息。

有关详情, 请参见第 12-3 页上的“确认 EMAIL 指令的错误详情”。

■ **D8782: BACnet 运行状态**

写入 BACnet 通信的运行状态。

有关详情, 请参见第 15-11 页上的“BACnet/IP 使用的特殊设备”。

■ **D8783: BACnet 错误信息**

写入 BACnet 通信中最后发生的错误信息。

有关详情, 请参见第 15-11 页上的“BACnet/IP 使用的特殊设备”。

■ **D8790: EtherNet/IP 运行状态**

写入 EtherNet/IP 通信的动作状态。

有关详情, 请参见第 16-13 页上的“EtherNet/IP 通信使用的特殊设备”。

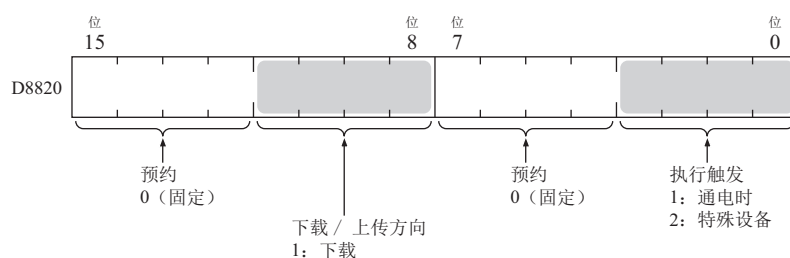
■ **D8791: EtherNet/IP 错误信息**

写入 EtherNet/IP 通信中最后发生的错误信息。

有关详情, 请参见第 16-13 页上的“EtherNet/IP 通信使用的特殊设备”。

■ **D8820: 从 SD 记忆卡下载用于服务器功能文件执行信息**

执行的服务器功能文件的下载信息。设备内的分配 (位分配) 如下所示。



■ **D8821: 从 SD 记忆卡下载用于服务器功能文件执行错误信息**

这是服务器功能文件的执行错误信息。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

■ **D8822: Web 服务器当前连接数 (Plus CPU 模块)**

这是 Web 服务器功能的当前连接数。仅支持 Plus CPU 模块。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

2: 设备地址

类型 ID、状态一览

■ 类型 ID

扩展模块、HMI 模块

类型 ID		型号
十六进制	二进制	
0x00	0000 0000	FC6A-N16B1、FC6A-N16B4、FC6A-N16B3
0x01	0000 0001	FC6A-R161、FC6A-R164、FC6A-T16K1、FC6A-T16K4、F6A-T16P1、FC6A-T16P4、F6A-T16K3、FCA-T16P3
0x02	0000 0010	FC6A-N32B3
0x03	0000 0011	FC6A-T32K3、FC6A-T32P3
0x04	0000 0100	FC6A-N08B1、FC6A-N08B4、FC6A-N08A11、FC6A-N08A14
0x05	0000 0101	FC6A-R081、FC6A-R084、FC6A-T08K1、FC6A-T08K4、FC6A-T08P1、FC6A-T08P4
0x06	0000 0110	FC6A-M08BR1、FC6A-M08BR4
0x07	0000 0111	FC6A-M24BR1、FC6A-M24BR4
0x18	0001 1000	FC6A-PH1
0x19	0001 1001	FC6A-EXM2、FC6A-EXM24
0x1A	0001 1010	FC6A-EXM1S、FC6A-EXM1S4
0x20	0010 0000	FC6A-J2C1、FC6A-J2C4
0x21	0010 0001	FC6A-J4A1、FC6A-J4A4
0x22	0010 0010	FC6A-J8A1、FC6A-J8A4
0x23	0010 0011	FC6A-K2A1、FC6A-K2A4
0x24	0010 0100	FC6A-K4A1、FC6A-K4A4
0x25	0010 0101	FC6A-L06A1、FC6A-L06A4
0x26	0010 0110	FC6A-L03CN1、FC6A-L03CN4
0x27	0010 0111	FC6A-J4CN1、FC6A-J4CN4
0x28	0010 1000	FC6A-J8CU1、FC6A-J8CU4
0x29	0010 1001	FC6A-F2M1、FC6A-F2M4
0x2A	0010 1010	FC6A-F2MR1、FC6A-F2MR4
0x2B	0010 1011	FC6A-J4CH1Y、FC6A-J4CH4Y
0x2C	0010 1100	FC6A-EXM1M
0x2E	0010 1110	FC6A-SIF52、FC6A-SIF524
0xFF	1111 1111	未连接

盒

类型 ID		型号
十六进制	二进制	
0x00	0000 0000	FC6A-PJ2A
0x01	0000 0001	FC6A-PK2AV
0x02	0000 0010	FC6A-PK2AW
0x03	0000 0011	FC6A-PJ2CP
0x06	0000 0110	FC6A-PC1
0x07	0000 0111	FC6A-PC3
0x09	0000 1001	FC6A-PTS4、FC6A-PTK4
0x0A	0000 1010	FC6A-PN4
0x0C	0000 1100	FC6A-PC4
0xFF	1111 1111	未连接

■ 状态

状态		说明
十六进制	二进制	
0x00	0000 0000	正常
0x81	1000 0001	通信错误（在扩展模块、HMI 模块、盒与 CPU 模块之间进行通信时，会发生异常。）
0x82	1000 0010	检测未知设备（已连接 FC6A 型以外的设备。）
0x83	1000 0011	设备配置错误（未连接设备或已连接与用户程序配置设备不同的设备。）
0x84	1000 0100	设备写入错误（设备的动作设置失败。）
0x85	1000 0101	系统更新错误（系统更新失败。）
0x86	1000 0110	增设扩展模块组合型主机通信错误（在增设扩展模块组合型主机与增设扩展模块组合型从机之间进行通信时，会发生异常。）

3: 通信设置

简介

本章将对使用 FC6A 型所具备的通信功能的设置方法及使用示例进行介绍。

对于通信端口和网络设置等需要事先设置的功能，请先在 WindLDR 上的“功能设置”对话框等中进行设置，然后将用户程序下载到 FC6A 型中。

有关 CAN J1939 通信的设置方法，请参见第 8-1 页上的“J1939 通信”。

设置列表

功能名称	概述	参考	设置位置
通信端口	设置符合通信设备的通信功能及端口。	3-2	功能设置
网络设置	设置用于将 FC6A 型连接至网络的信息。	3-3	
网络管理	设置 SNTP 设置和 PING 指令的超时时间。	3-11	
连接设置	可为每个连接配置以太网通信的通信模式和参数，以便 FC6A 型可通过以太网与其他网络设备进行通信。	3-14	
远程主机列表	可在远程主机列表中注册和管理网络中与 FC6A 型通信的远程主机设备。	3-21	远程主机列表
自动 Ping 功能	设置监控远程主机的网络连接状态的自动 Ping 功能。	3-23	

3: 通信设置

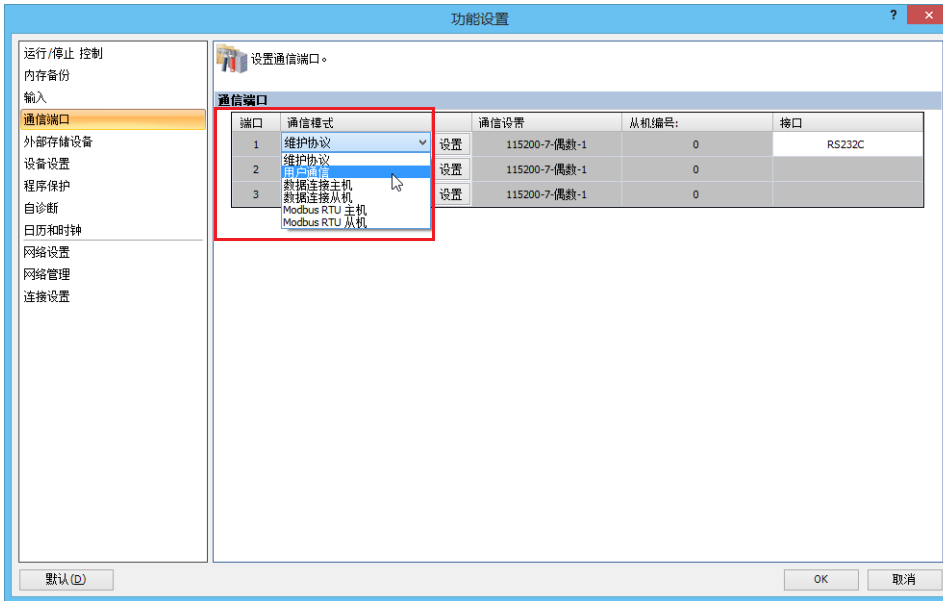
通信端口的设置

将对使用通信端口 1 ~ 33，与通信设备进行通信时的通信端口设置进行介绍。

设置 WindLDR

根据外围设备的通信规格配置通信格式。

1. 从 WindLDR 菜单栏中选择“配置”>“通信端口”。
出现“功能设置”对话框。
2. 单击要使用端口的“通信模式”，选择要使用的通信模式。
将显示与该通信模式相对应的配置对话框。



3. 在配置对话框中，根据目的地设备的通信格式更改设置。
下面显示了一个用户通信示例。



4. 单击“确定”。
配置即完成。

网络设置

针对使用以太网端口 1、以太网端口 2、HMI-以太网端口，将 FC6A 型连接到网络的设置进行介绍。

概述

为了将 FC6A 型连接到以太网的网络，可在 FC6A 型的网络设置中设置的以太网端口和设置项目如下所示。

CPU 模块	以太网端口	设置项目								
		IP 设置	DNS 设置	SNTP 设置	Ping 设置	电子邮件设置	Web 服务器设置	BACnet 设置	FTP 设置	EtherNet/IP 设置
All-in-One CPU 模块	以太网端口 1	是	是	是	是	—	—	—	—	—
Plus CPU 模块	以太网端口 1	是	是	是	是	是	是	是	是	—
	以太网端口 2	是	是	—	是	—	—	—	—	是
HMI 模块	HMI-Ethernet 端口	是	是	—	—	是	是	—	—	—

网络设置

IP 设置

FC6A 型的 IP 地址、子网掩码、默认网关的指定方法，可从以下 3 种进行选择。

指定方法	说明
自动获取 IP 地址 (DHCP)	FC6A 型会在用户程序下载完成时或 FC6A 型的电源启动时，自动从 DHCP 服务器获取 IP 地址、子网掩码、默认网关。但前提是设置有 FC6A 型的网络上必须存在 DHCP 服务器。
使用特殊数据寄存器设置 IP 地址	使用特殊数据寄存器，设置以太网端口 1、以太网端口 2、HMI-以太网端口的 IP 地址、子网掩码、默认网关。上述值会在 FC6A 型的电源启动时及特殊内部继电器 M8190、M8333 或 M8184 从 OFF 变为 ON 的上升沿时，反映至 FC6A 型。
使用以下 IP 地址	指定固定的 IP 地址、子网掩码、默认网关。此处指定的值会在用户程序下载完成时，反映至 FC6A 型。

注释:

- FC6A 型的默认 IP 地址、子网掩码、默认网关为“192.168.1.5”、“255.255.255.0”、“0.0.0.0”。
- FC6A 型的 IP 地址、子网掩码、默认网关可使用 HMI 模块进行更改。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

DNS 设置

DNS 服务器地址的指定方法，可从以下 3 种进行选择。

指定方法	说明
自动获取 DNS 服务器地址 (DHCP)	FC6A 型会在用户程序下载完成时或 FC6A 型的电源启动时，自动获取 DNS 服务器的地址。但前提是设置有 FC6A 型的网络上必须存在 DHCP 服务器。
使用特殊数据寄存器设置 DNS 服务器地址	使用特殊数据寄存器，设置以太网端口 1、以太网端口 2、HMI-以太网端口首选 DNS 服务器地址及备用 DNS 服务器地址。上述值会在 FC6A 型的电源启动时及特殊内部继电器 M8190、M8333 或 M8184 从 OFF 变为 ON 的上升沿时，反映至 FC6A 型。
使用以下 DNS 服务器地址	指定固定的首选 DNS 服务器地址及备用 DNS 服务器地址。此处指定的值会在用户程序下载完成时，反映至 FC6A 型。

注释:

- 无法从首选 DNS 服务器获取 IP 地址时，请访问备用 DNS 服务器。
- DNS 服务器的地址可使用 HMI 模块进行更改。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

3: 通信设置

由特殊数据寄存器进行的网络设置

如果使用特殊数据寄存器来设置 IP 地址、DNS 服务器的地址，在 FC6A 型的电源启动时及特殊内部继电器 M8190、M8333 从 OFF 变为 ON 的上升沿时，将设置存储在特殊数据寄存器中的值。

例)

- 将 M8190 (CPU 模块以太网端口 1 网络设置更改触发器) 从 OFF 变为 ON 时，将存储在 D8304 ~ D8323 中的值设置为以太网端口 1 的 IP 地址、DNS 服务器的地址。
- 将 M8333 (CPU 模块以太网端口 2 网络设置更改触发器) 从 OFF 变为 ON 时，将存储在 D8631 ~ D8650 中的值设置为以太网端口 2 的 IP 地址、DNS 服务器的地址。

特殊数据寄存器	说明	读 / 写
D8304-D8307	设置在以太网端口 1 中的 IP 地址的预置值。 例) IP 地址为 192.168.0.1 时 D8304=192、D8305=168、D8306=0、D8307=1	读 / 写
D8308-D8311	设置在以太网端口 1 中的子网掩码的预置值。 例) 子网掩码为 255.255.255.0 时 D8308=255、D8309=255、D8310=255、D8311=0	读 / 写
D8312-D8315	设置在以太网端口 1 中的默认网关的预置值。 例) 默认网关为 192.168.0.24 时 D8312=192、D8313=168、D8314=0、D8315=24	读 / 写
D8316-D8319	以太网端口 1 所使用的首选 DNS 服务器地址的预置值。 例) 首选 DNS 服务器地址为 192.168.0.100 时 D8316=192、D8317=168、D8318=0、D8319=100	读 / 写
D8320-D8323	以太网端口 1 所使用的备用 DNS 服务器地址的预置值。 例) 备用 DNS 服务器地址为 192.168.0.101 时 D8320=192、D8321=168、D8322=0、D8323=101	读 / 写
D8631-D8634	设置在以太网端口 2 中的 IP 地址的预置值。 例) IP 地址为 192.168.1.1 时 D8631=192、D8632=168、D8633=1、D8634=1	读 / 写
D8635-D8638	设置在以太网端口 2 中的子网掩码的预置值。 例) 子网掩码为 255.255.255.0 时 D8635=255、D8636=255、D8637=255、D8638=0	读 / 写
D8639-D8642	设置在以太网端口 2 中的默认网关的预置值。 例) 默认网关为 192.168.1.24 时 D8639=192、D8640=168、D8641=1、D8642=24	读 / 写
D8643-D8646	以太网端口 2 所使用的首选 DNS 服务器地址的预置值。 例) 首选 DNS 服务器地址为 192.168.1.100 时 D8643=192、D8644=168、D8645=1、D8646=100	读 / 写
D8647-D8650	以太网端口 2 所使用的备用 DNS 服务器地址的预置值。 例) 备用 DNS 服务器地址为 192.168.1.101 时 D8647=192、D8648=168、D8649=1、D8650=101	读 / 写

网络设置的确认

可通过特殊数据寄存器确认 FC6A 型以太网端口 1 及以太网端口 2 所使用的目前的 IP 地址、子网掩码、默认网关以及 DNS 服务器地址。

特殊数据寄存器	说明	读 / 写
D8330-D8333	以太网端口 1 中使用的 IP 地址的当前值。 例) IP 地址为 192.168.0.1 时 D8330=192、D8331=168、D8332=0、D8333=1	读
D8334-D8337	以太网端口 1 中使用的子网掩码的当前值。 例) 子网掩码为 255.255.255.0 时 D8334=255、D8335=255、D8336=255、D8337=0	读
D8338-D8341	以太网端口 1 中使用的默认网关的当前值。 例) 默认网关为 192.168.0.24 时 D8338=192、D8339=168、D8340=0、D8341=24	读
D8342-D8345	以太网端口 1 所使用的首选 DNS 服务器地址的当前值。 例) 首选 DNS 服务器地址为 192.168.0.100 时 D8342=192、D8343=168、D8344=0、D8345=100	读
D8346-D8349	以太网端口 1 所使用的备用 DNS 服务器地址的当前值。 例) 备用 DNS 服务器地址为 192.168.0.101 时 D8346=192、D8347=168、D8348=0、D8349=101	读
D8657-D8660	以太网端口 2 中使用的 IP 地址的当前值。 例) IP 地址为 192.168.1.1 时 D8657=192、D8658=168、D8659=1、D8660=1	读
D8661-D8664	以太网端口 2 中使用的子网掩码的当前值。 例) 子网掩码为 255.255.255.0 时 D8661=255、D8662=255、D8663=255、D8664=0	读
D8665-D8668	以太网端口 2 中使用的默认网关的当前值。 例) 默认网关为 192.168.1.24 时 D8665=192、D8666=168、D8667=1、D8668=24	读
D8669-D8672	以太网端口 2 所使用的首选 DNS 服务器地址的当前值。 例) 首选 DNS 服务器地址为 192.168.1.100 时 D8669=192、D8670=168、D8671=1、D8672=100	读
D8673-D8676	以太网端口 2 所使用的备用 DNS 服务器地址的当前值。 例) 备用 DNS 服务器地址为 192.168.1.101 时 D8673=192、D8674=168、D8675=1、D8676=101	读

3: 通信设置

由 HMI 模块的特殊数据寄存器进行的网络设置

如果使用特殊数据寄存器来设置 IP 地址、DNS 服务器的地址，在 FC6A 型的电源启动时及特殊内部继电器 M8184 从 OFF 变为 ON 的上升沿时，将设置存储在特殊数据寄存器中的值。

例)

将 M8184 (HMI 模块网络设置更改触发器) 从 OFF 变为 ON 时，将存储在 D8437 ~ D8456 中的值设置为 HMI 以太网端口的 IP 地址、DNS 服务器的地址。

特殊数据寄存器	说明	读 / 写
D8437-D8440	HMI 模块 IP 地址的预置值。 例) IP 地址为 192.168.0.1 时 D8437=192、D8438=168、D8439=0、D8440=1	读 / 写
D8441-D8444	HMI 模块子网掩码的预置值。 例) 子网掩码为 255.255.255.0 时 D8441=255、D8442=255、D8443=255、D8444=0	读 / 写
D8445-D8448	HMI 模块默认网关的预置值。 例) 默认网关为 192.168.0.24 时 D8445=192、D8446=168、D8447=0、D8448=24	读 / 写
D8449-D8452	HMI 模块使用的首选 DNS 服务器地址的预置值。 例) 首选 DNS 服务器地址为 192.168.0.100 时 D8449=192、D8450=168、D8451=0、D8452=100	读 / 写
D8453-D8456	HMI 模块使用的备用 DNS 服务器地址的预置值。 例) 备用 DNS 服务器地址为 192.168.0.101 时 D8453=192、D8454=168、D8455=0、D8456=101	读 / 写

HMI 模块网络设置的确认

可通过特殊数据寄存器确认目前的 HMI 模块的 IP 地址、子网掩码、默认网关以及 DNS 服务器地址。

特殊数据寄存器	说明	读 / 写
D8388-D8391	HMI 模块 IP 地址的当前值。 例) IP 地址为 192.168.0.1 时 D8388=192、D8389=168、D8390=0、D8391=1	读
D8392-D8395	HMI 模块子网掩码的当前值。 例) 子网掩码为 255.255.255.0 时 D8392=255、D8393=255、D8394=255、D8395=0	读
D8396-D8399	HMI 模块默认网关的当前值。 例) 默认网关为 192.168.0.24 时 D8396=192、D8397=168、D8398=0、D8399=24	读
D8400-D8403	HMI 模块使用的首选 DNS 服务器地址的当前值。 例) 首选 DNS 服务器地址为 192.168.0.100 时 D8400=192、D8401=168、D8402=0、D8403=100	读
D8404-D8407	HMI 模块使用的备用 DNS 服务器地址的当前值。 例) 备用 DNS 服务器地址为 192.168.0.101 时 D8404=192、D8405=168、D8406=0、D8407=101	读

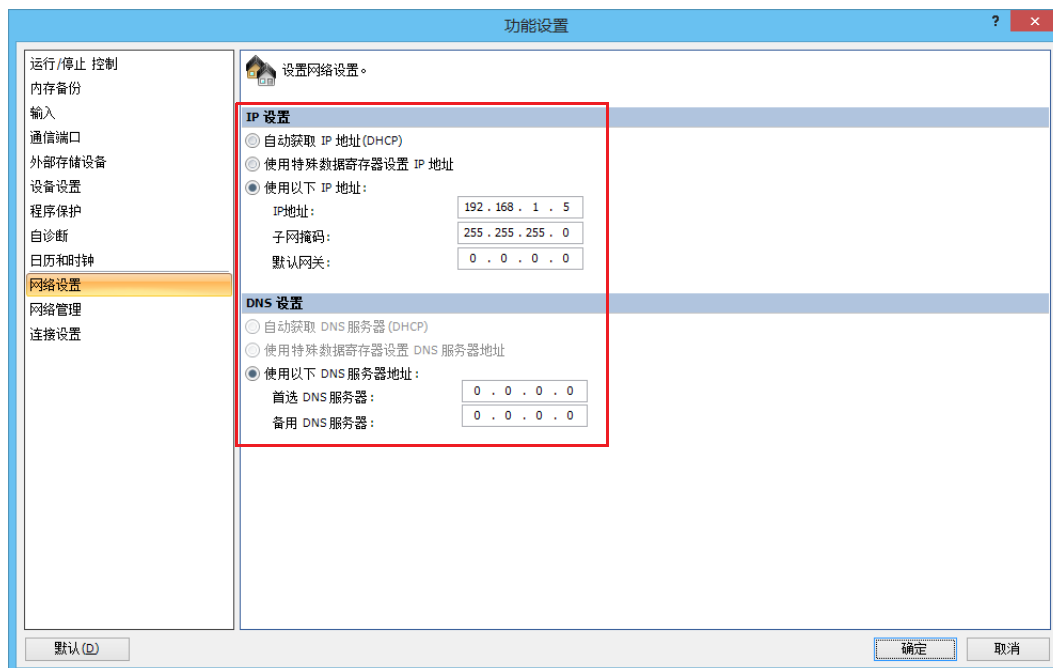
设置 WindLDR

1. 在“设置”选项卡的“功能设置”中单击以下项目。

- All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块
“网络设置”
- Plus CPU 模块
设置以太网端口 1 时为“以太网端口 1”
设置以太网端口 2 时为“以太网端口 2”

出现“功能设置”对话框。

2. 分别对“IP 设置”、“DNS 设置”进行设置。



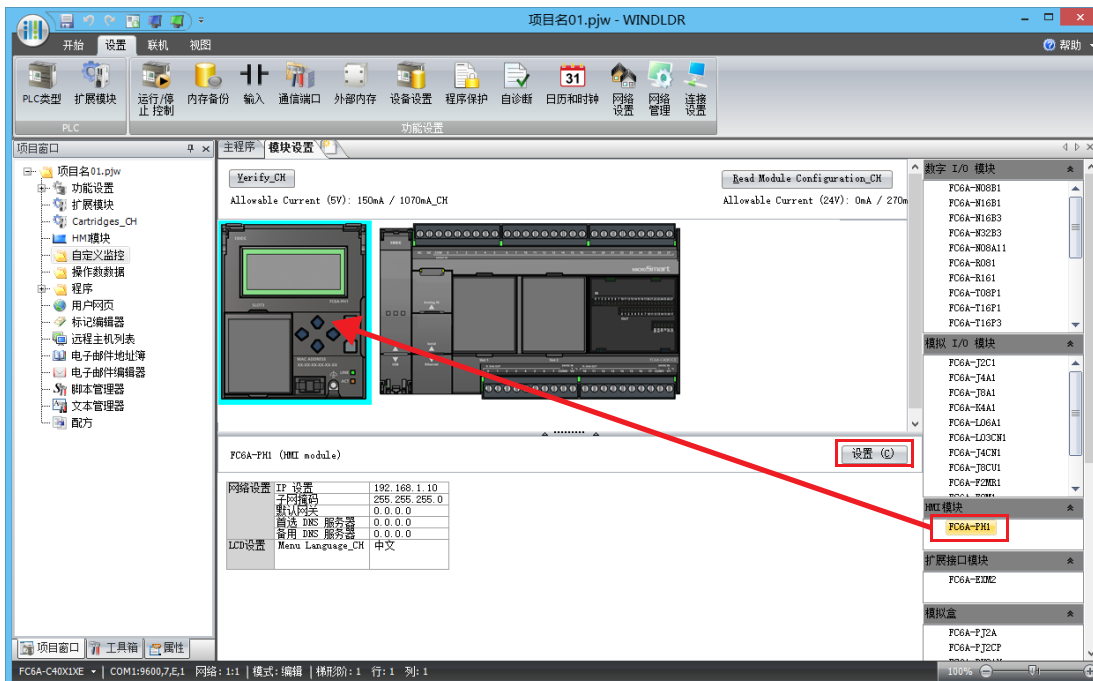
3. 单击“确定”。

4. 可通过模块构成编辑器进行 HMI 模块的网络设置。
在“设置”选项卡的“PLC”中单击“扩展模块”。

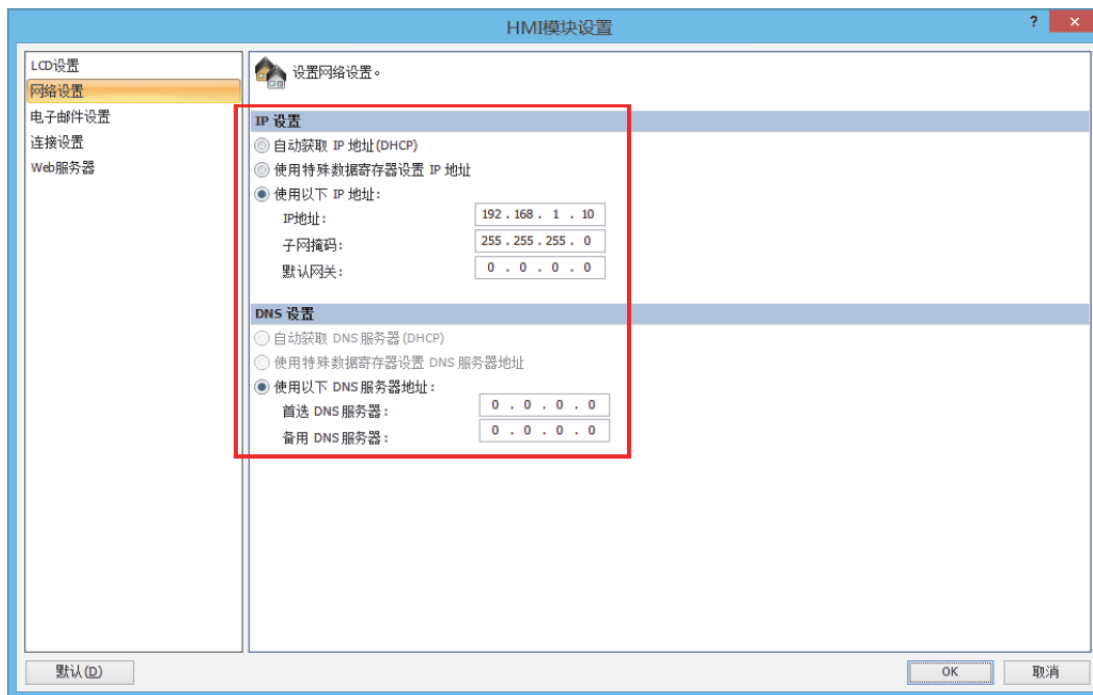
5. 单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮。
将显示 HMI 模块的设置对话框。

注释：在项目窗口中双击“HMI 模块”，也可显示 HMI 模块的设置对话框。

3: 通信设置



6. 单击“网络设置”选项卡，分别对“IP 设置”、“DNS 设置”进行设置。



配置即完成。

IP 设置 / DNS 设置切换

使用特殊数据寄存器（D8303、D8630）和特殊内部继电器（M8190、M8333），可切换 CPU 模块的以太网端口 1 或以太网端口 2 的 IP 设置和 DNS 设置。切换步骤如下所示。

设置 WindLDR

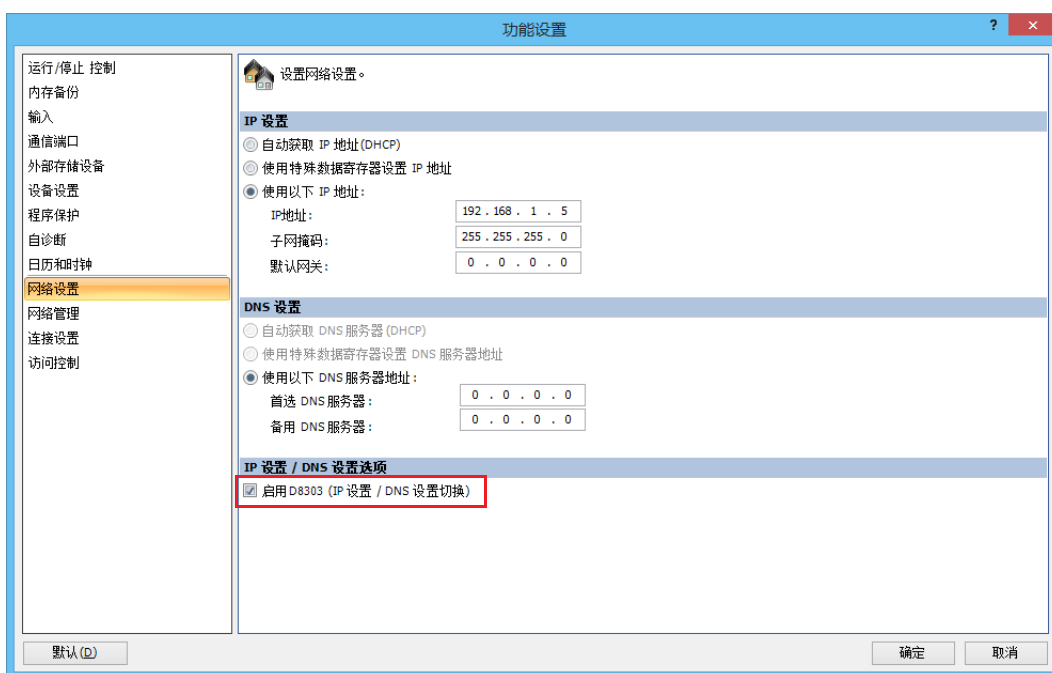
1. 在“设置”选项卡的“功能设置”中单击以下项目。

- All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块
“网络设置”

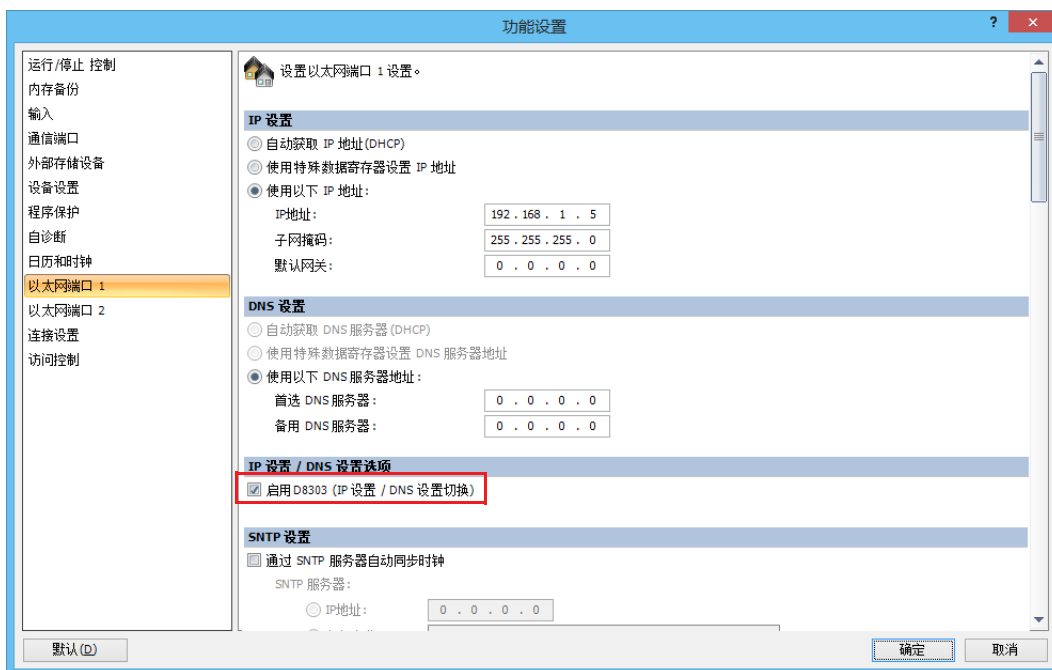
- Plus CPU 模块
设置以太网端口 1 时为“以太网端口 1”
设置以太网端口 2 时为“以太网端口 2”

出现“功能设置”对话框。

2. 分别对“IP 设置”、“DNS 设置”进行设置。
 3. 如需设置以太网端口 1，选中“启用 D8303（IP 设置 / DNS 设置切换）”复选框。
- All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块



- Plus CPU 模块



3: 通信设置

4. 如需设置以太网端口 2，选中“启用 D8630（IP 设置 / DNS 设置切换）”复选框。
- Plus CPU 模块



5. 单击“确定”。

配置即完成。

下载用户程序后，IP 设置和 DNS 设置将依照 D8303 或 D8630 的预置值进行切换。D8303 和 D8630 的预置值含义如下所示。

D8303 或 D8630 中存储值的含义

预置值	IP 设置 / DNS 设置
0	依照功能设置。
1	启用 DHCP。
2	依照特殊数据寄存器（D8304 ~ D8323 或 D8631 ~ D8650）的设置。

注释：

- 将 D8303 或 D8630 的值设为 2（依照特殊数据寄存器的设置），并将 M8190 或 M8333 由 OFF 切换到 ON 或重新接通电源后，与功能设置无关，可强制使用特殊数据寄存器来设置“IP 设置”、“DNS 设置”。
- 将 D8303 或 D8630 的值设为 1（启用 DHCP），并将 M8190 或 M8333 由 OFF 切换到 ON 或重新接通电源后，与功能设置无关，可强制进行自动获取（DHCP）设置。
- 要恢复启用功能设置时，请将 D8303 或 D8630 的值设为 0，并将 M8190 或 M8333 由 OFF 切换到 ON 或重新接通电源。

网络管理

SNTTP 设置

本节将对从网络上的 SNTTP 服务器获取当前时间（UTC）调整 FC6A 型的内置时钟的设置进行介绍。

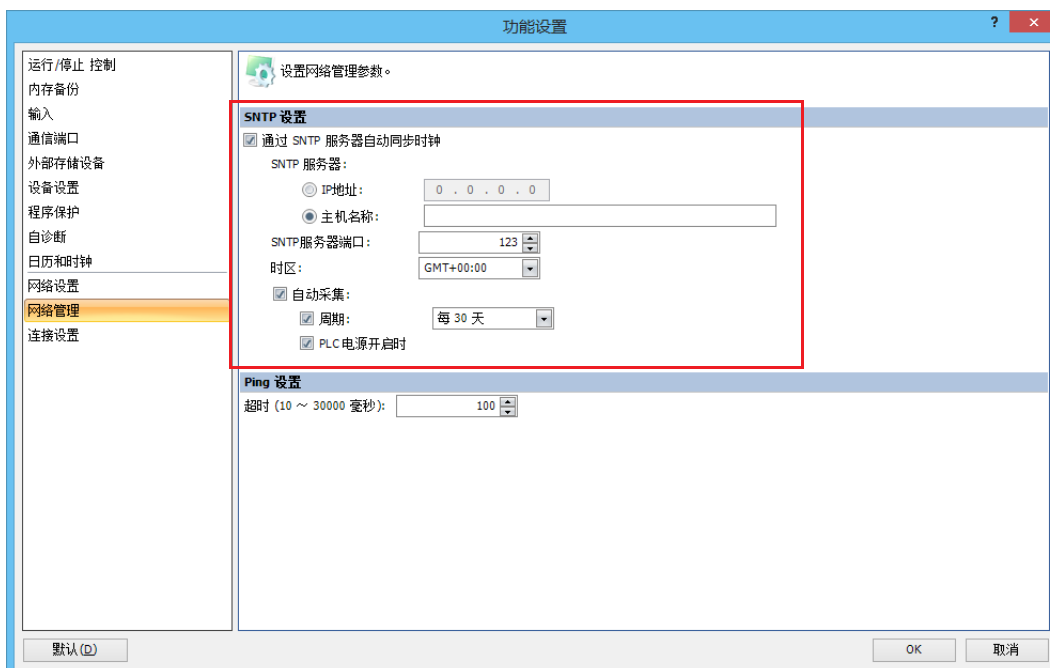
概述

FC6A 型可从网络上的 SNTTP 服务器自动或以设置的间隔获取当前时间（UTC），并根据时区的设置调整内置时钟。如果启用夏时制设置，在夏时制期间中，将夏时制校正反映至通过 SNTTP 服务器所获取的当前时间（UTC）。

设置 WindLDR

设置获取当前时间的 SNTTP 服务器和当前时间的获取方法。

- 在“设置”选项卡的“功能设置”中单击以下项目。
 - All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块
“网络管理”
 - Plus CPU 模块
“以太网端口 1”
- 选中“过 SNTTP 服务器自动同步时钟”复选框。



- 设置 SNTTP 服务器、时区、当前时间的获取方法。

项目	预置值
SNTTP 服务器（IP 地址）	指定获取当前时间的 SNTTP 服务器的 IP 地址。 输入形式为“xxx.xxx.xxx.xxx”。“xxx”中输入 0 至 255 的值。
SNTTP 服务器（主机名称）	指定获取当前时间的 SNTTP 服务器的主机名称。 主机名称最多可输入 40 个半角英文数字。
SNTTP 服务器端口	指定 SNTTP 服务器的端口编号（0 至 65,535）。
时区	选择以 SNTTP 服务器获取的标准时间相对应的地区时间带之差。 可在“使用日历和时钟中的设置” ^{*1} 或“UTC-12:00”至“UTC+14:00”范围内设置。

*1 使用“功能设置”对话框的“日历和时钟”标签中设置的时区。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“时区”。

3: 通信设置

注释: 可以使用时区偏移量 (D8413) 以 15 分钟为单位调整时区。有关详情, 请参见第 3-12 页上的“时区的调整 (D8413)”。

- 如需自动获取当前时间, 请选中“自动采集”复选框。
发生保持数据错误时, 或下载用户程序自动获取复选框从未选中更改为选中时, 将自动从 SNTP 服务器获取当前时间。
如需在上述以外时间点定期获取当前时间, 请选中“周期”复选框, 从“每 10 分钟”、“每小时”、“每天”、“1 周”、“每 30 天”中进行选择。
此外, 打开 FC6A 型的电源后, 如需从 SNTP 服务器获取当前时间, 请选中“PLC 电源开启时”复选框。
如果未选中“自动采集”复选框, 特殊内部继电器 (M8191) 从 OFF 置于 ON 时, 将从 SNTP 服务器获取当前时间。

- 单击“确定”。

配置即完成。

时区的调整 (D8413)

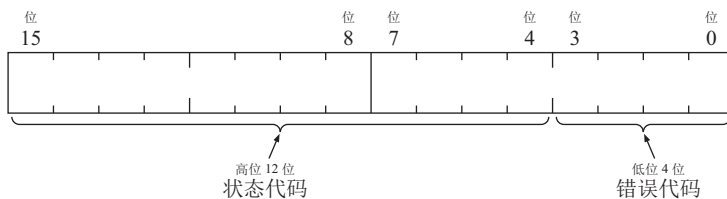
选择的时区 (UTC-12:00 至 UTC+14:00) 请使用特殊数据寄存器 D8413 (时区偏移量) 的值, 以 15 分钟为单位进行调整。

例如, 选择了“UTC+09:00”时, 如果 D8413 中存储 +1, 则时间前进 15 分钟, 为“UTC+09:15”。如果 D8413 中存储 -2, 则时间倒退 30 分钟, 为“UTC+08:30”。

运行状态的确认 (D8414)

特殊数据寄存器 D8414 (SNTP 运行状态) 中存储有 SNTP 服务器的运行状态。运行状态表示运行的状态 (状态代码) 和错误内容 (错误代码)。

运行的状态 (状态代码) 存储在 D8414 的高位 12 位中, 错误内容 (错误代码) 存储在 D8414 的低位 4 位中。



状态代码	运行的状态	状态说明
0 (000000000000)	空操作	未访问至 SNTP 服务器的状态
32 (000001000000)	等待响应	访问至 SNTP 服务器, 等待来自 SNTP 服务器的响应的状态
64 (000001000000)	成功获取时间信息	正常接收到来自 SNTP 服务器的响应的状态

错误代码	错误内容
0 (0000)	正常
2 (0010)	超时错误
3 (0011)	未能 DNS 解决已设置的 SNTP 服务器的 IP 地址
9 (1001)	接收到不正确的数据

在任意时间点获取当前时间 (M8191)

在特殊内部继电器 M8191 (SNTP 获取标记) 从 OFF 变为 ON 的上升沿时, 将从 SNTP 服务器获取当前时间。

获取当前时间后的经过时间 (D8415)

特殊数据寄存器 D8415 (SNTP 访问后经过时间) 中以分钟为单位存储有最后从 SNTP 服务器获取当前时间到当前为止的经过时间 (0 至 65,535)。由于可存储的值为 0 至 65,535, 为 65,535 分钟 ÷ 60 分钟 ÷ 24 小时 = 约 45 日, 因此最大可测量 45 日。

例如, 最后从 SNTP 服务器获取当前时间的日期时间为 1 月 1 日 12:00 时, 如果在 1 月 1 日 15:00 确认 D8415 的值, 则经过了 3 小时 = 180 分钟, 因此将存储“180”。

D8415 将在成功获取当前时间时复位为 0, 并开始计算经过时间。如果未成功从 SNTP 服务器获取当前时间, D8415 的值将不会更新。

PING 设置

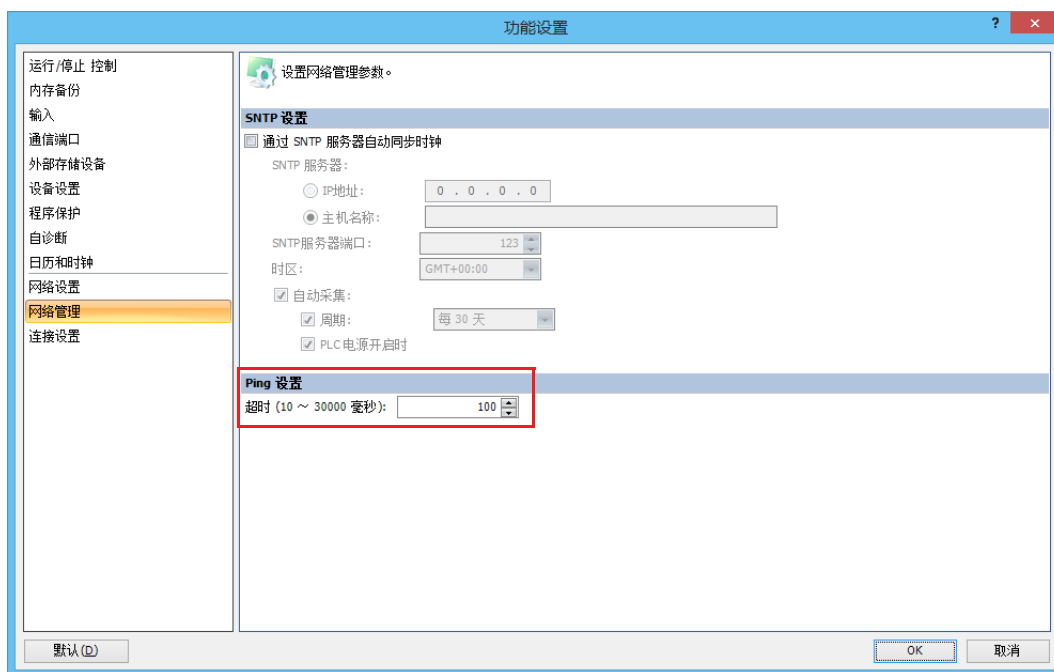
将对执行 PING 指令或自动 Ping 功能时的各 Ping 超时时间的设置进行介绍。

概述

通过 PING 指令或自动 Ping 功能执行 Ping 时，可向指定的远程主机发送 Ping 数据组，并确认是否可通过 IP LEVEL 进行通信。设置发送该 Ping 后到超时为止的时间。

设置 WindLDR

- 在“设置”选项卡的“功能设置”中单击以下项目。
 - All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块
“网络管理”
 - Plus CPU 模块
设置以太网端口 1 时为“以太网端口 1”
设置以太网端口 2 时为“以太网端口 2”
- 在“超时（10 ~ 30000 毫秒）”中将 PING 指令的超时时间设置为 10 至 30,000ms 之间的数值。
默认为 100ms。



- 单击“确定”。

配置即完成。

3: 通信设置

连接设置

将对使用 FC6A 型的以太网端口 1、以太网端口 2、HMI- 以太网端口的 TCP/IP 通信中的客户端 / 服务器通信的设置进行介绍。

应用

All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块允许在以太网端口 1 上最多配置八个连接。Plus CPU 模块最多可以配置 16 个到以太网端口 1 和以太网端口 2 的连接。请参阅下表，了解可以为每个连接设置的通信。

使用 HMI 模块时，最多可以配置八个与 HMI 以太网端口的连接。但是，只能为该连接设置维护通信。

使用**功能设置**对话框设置通信所需的连接。

如果将连接用作服务器，则可以为连接配置按 IP 地址过滤。只有具有特定 IP 地址的外部设备才能与 FC6A 型通信，并且可以限制来自未指定客户端的访问。

功能说明

各连接的通信模式可从维护通信服务器、用户通信服务器、用户通信客户端、Modbus TCP 服务器、Modbus TCP 客户端 MC 协议客户端和未使用 6 个选项中进行选择，而 Plus CPU 模块会从加上用户通信（UDP）的 8 个选项中进行选择。All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多可设置 8 个连接、Plus CPU 模块最多可设置 16 个连接。

若要限制访问 FC6A 型，可使用 IP 地址过滤功能。通过指定可访问 FC6A 型的 IP 地址，可防止匿名访问。

可在以太网端口中使用的通信模式

各以太网端口支持的通信模式如下所示。

通信模式	All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块		HMI 模块
	以太网端口 1	以太网端口 1	以太网端口 2	HMI- 以太网端口
维护通信服务器（默认）	X	X	X	X
用户通信服务器	X	X	X	—
用户通信客户端	X	X	X	—
用户通信（UDP）	—	X	X	—
Modbus TCP 服务器	X	X	X	—
Modbus TCP 客户端	X	X	X	—
MC 协议客户端	—	X	X	—
未使用	X	X	X	X

连接数

CPU 模块	个数
All-in-One CPU 模块	最大 16
Plus CPU 模块	最大 16 ^{*1}
HMI 模块	最大 8

*1 使用 Modbus TCP 客户端时，在一个连接设置中最多可以注册 255 个通信对象。最多可以同时设置 16 个连接设置。在每个连接设置中注册的通信对象中，可以在保持连接的状态下将通信对象与总共 128 个通信对象进行切换。有关详细信息，请参见“第 6 章 Modbus TCP 客户端”。使用 MC 协议时，在一个连接设置中最多可以注册 255 个通信对象。每次通信对象更改时，MC 协议通信都会断开连接并建立连接。

通信模式的概要

各通信模式的概要如下所示。

通信模式	概要	参考
维护通信服务器（默认）	可通过 WindLDR 监控或更改设备，并下载或上传用户程序。	4-7
用户通信服务器	可按照 ETXD（以太网用户通信发送）指令及 ERXD（以太网用户通信接收）指令与客户端设备进行通信。	5-47
用户通信客户端	可按照 ETXD（以太网用户通信发送）指令及 ERXD（以太网用户通信接收）指令与服务器设备进行通信。	5-43
用户通信（UDP）*1	可按照 ETXD（以太网用户通信发送）指令及 ERXD（以太网用户通信接收）指令与 UDP 通信设备进行通信。	5-50
Modbus TCP 服务器	将 FC6A 型设置为 Modbus TCP 服务器时，可通过支持 Modbus TCP 客户端的设备监控或更改 FC6A 型的数据。	6-27
Modbus TCP 客户端	可通过 Modbus TCP 客户端监控或更改支持 Modbus TCP 服务器的设备的数据。	6-22
MC 协议客户端	可通过 MC 协议客户端监控或更改支持 MC 协议的兼容设备的数据。	18-2
未使用	不使用连接。由于会停止内部控制处理，因此其他连接的响应性将提高。	—

*1 仅可选择 Plus CPU 模块。

连接状态和连接的 IP 地址

可使用特殊内部继电器 M8212 至 M8231 和 M8345 至 M8364 确认与远程主机的连接状态。当与远程主机的连接建立后，相应的特殊内部继电器开启。当连接断开时，相应的特殊内部继电器关闭。可使用特殊数据寄存器 D8350 至 D8381 确认远程主机的 IP 地址。

注释：R/W 是读 / 写的简称。当为 R/W 时，可以读取和写入。当为 R 时，只能读取。当为 W 时，只能写入。

特殊内部继电器

设备地址	说明	详情	读 / 写
M8212	连接 1 的状态	当与远程主机的连接建立后，特殊内部继电器将会开启。如果未建立连接，继电器将会关闭。	读
M8213	连接 2 的状态		
M8214	连接 3 的状态		
M8215	连接 4 的状态		
M8216	连接 5 的状态		
M8217	连接 6 的状态		
M8220	连接 7 的状态		
M8221	连接 8 的状态		
M8345 ^{*1}	连接 9 的状态		
M8346 ^{*1}	连接 10 的状态		
M8347 ^{*1}	连接 11 的状态		
M8350 ^{*1}	连接 12 的状态		
M8351 ^{*1}	连接 13 的状态		
M8352 ^{*1}	连接 14 的状态		
M8353 ^{*1}	连接 15 的状态		
M8354 ^{*1}	连接 16 的状态		

3: 通信设置

设备地址	说明	详情	读 / 写
M8222	连接 1 的切断标记	从 OFF 变为 ON 时，会强制切断与服务器的连接，并停止通信。从 ON 变为 OFF 时，会确立与服务器的连接，并重新开始通信。 仅在用户通信客户端有效。	读 / 写
M8223	连接 2 的切断标记		
M8224	连接 3 的切断标记		
M8225	连接 4 的切断标记		
M8226	连接 5 的切断标记		
M8227	连接 6 的切断标记		
M8230	连接 7 的切断标记		
M8231	连接 8 的切断标记		
M8355 ^{*1}	连接 9 的切断标记		
M8356 ^{*1}	连接 10 的切断标记		
M8357 ^{*1}	连接 11 的切断标记		
M8360 ^{*1}	连接 12 的切断标记		
M8361 ^{*1}	连接 13 的切断标记		
M8362 ^{*1}	连接 14 的切断标记		
M8363 ^{*1}	连接 15 的切断标记		
M8364 ^{*1}	连接 16 的切断标记		

*1 仅可使用 Plus CPU 模块。

连接 IP 地址

特殊数据寄存器	说明		读 / 写
D8350-D8353	连接 1 连接的 IP 地址	外部设备与各连接之间的通信已确立时，所连接的外部设备的 IP 地址将如下所示存储到各特殊数据寄存器中。 例) IP 地址: aaa.bbb.ccc.ddd、 D8350=aaa、D8351=bbb、D8352=ccc、D8353=ddd	读
D8354-D8357	连接 2 连接的 IP 地址		
D8358-D8361	连接 3 连接的 IP 地址		
D8362-D8365	连接 4 连接的 IP 地址		
D8366-D8369	连接 5 连接的 IP 地址		
D8370-D8373	连接 6 连接的 IP 地址		
D8374-D8377	连接 7 连接的 IP 地址		
D8378-D8381	连接 8 连接的 IP 地址		
D8677-D8680 ^{*1}	连接 9 连接的 IP 地址		
D8681-D8684 ^{*1}	连接 10 连接的 IP 地址		
D8685-D8688 ^{*1}	连接 11 连接的 IP 地址		
D8689-D8692 ^{*1}	连接 12 连接的 IP 地址		
D8693-D8696 ^{*1}	连接 13 连接的 IP 地址		
D8697-D8700 ^{*1}	连接 14 连接的 IP 地址		
D8701-D8704 ^{*1}	连接 15 连接的 IP 地址		
D8705-D8708 ^{*1}	连接 16 连接的 IP 地址		

*1 仅可使用 Plus CPU 模块。

HMI 模块中的连接状态信息及连接 IP 地址

D8429 中指定的 HMI 模块的连接编号连接状态，存储在特殊内部继电器、特殊数据寄存器中。

与 HMI 模块外部设备的连接状态，可通过特殊内部继电器 M8232 进行确认。

与外部设备建立连接时变为 ON，未建立连接时则变为 OFF。

连接目标的 IP 地址可通过特殊数据寄存器 D8430 至 D8433 进行确认。

指定读取目标连接编号

特殊内部继电器	说明		读 / 写
D8429	HMI 模块 读取目标连接编号	指定编号的连接信息将反映到 D8430 至 D8434、M8232 中。 指定 0 时，所有目标设备将设为 0。 如果指定了不存在的连接编号，则进行与指定 0 时相同的动作。	读 / 写

状态信息

特殊内部继电器	说明		读 / 写
M8232	HMI 模块 连接状态	在 D8429 中指定的连接，将在与外部设备连接时打开，未与外部设备连接时关闭。	读

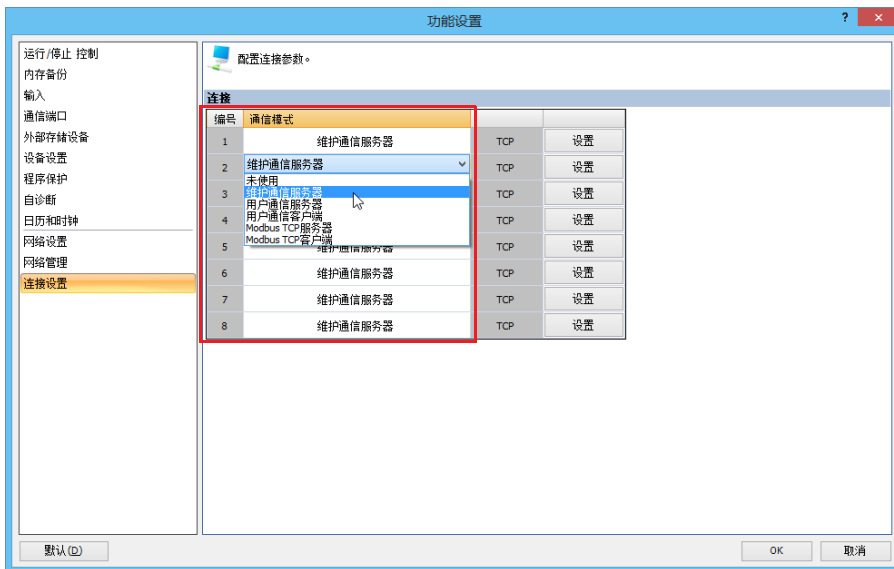
连接 IP 地址

特殊内部继电器	说明		读 / 写
D8430-D8433	HMI 连接 连接 IP 地址	在外部设备与 D8429 中指定的连接之间确立通信时，所连接的外部设备 IP 地址将如下所示存储到各特殊数据寄存器中。 例) IP 地址: aaa.bbb.ccc.ddd、 D8430=aaa、D8431=bbb、D8432=ccc、D8433=ddd	读

3: 通信设置

设置 WindLDR

1. 从 WindLDR 菜单栏中选择“配置”>“连接设置”。
出现“功能设置”对话框。
2. 单击要使用端口的“通信模式”，选择要使用的连接的通信模式。



设置为可用作客户端、服务器的 All-in-One CPU 模块最多分配 8 个连接、为 Plus CPU 模块分配 16 个连接的通信模式。每个连接可设置为不同的端口号。将不用的连接设置为“未使用”。
将显示支持通信模式的对话框。

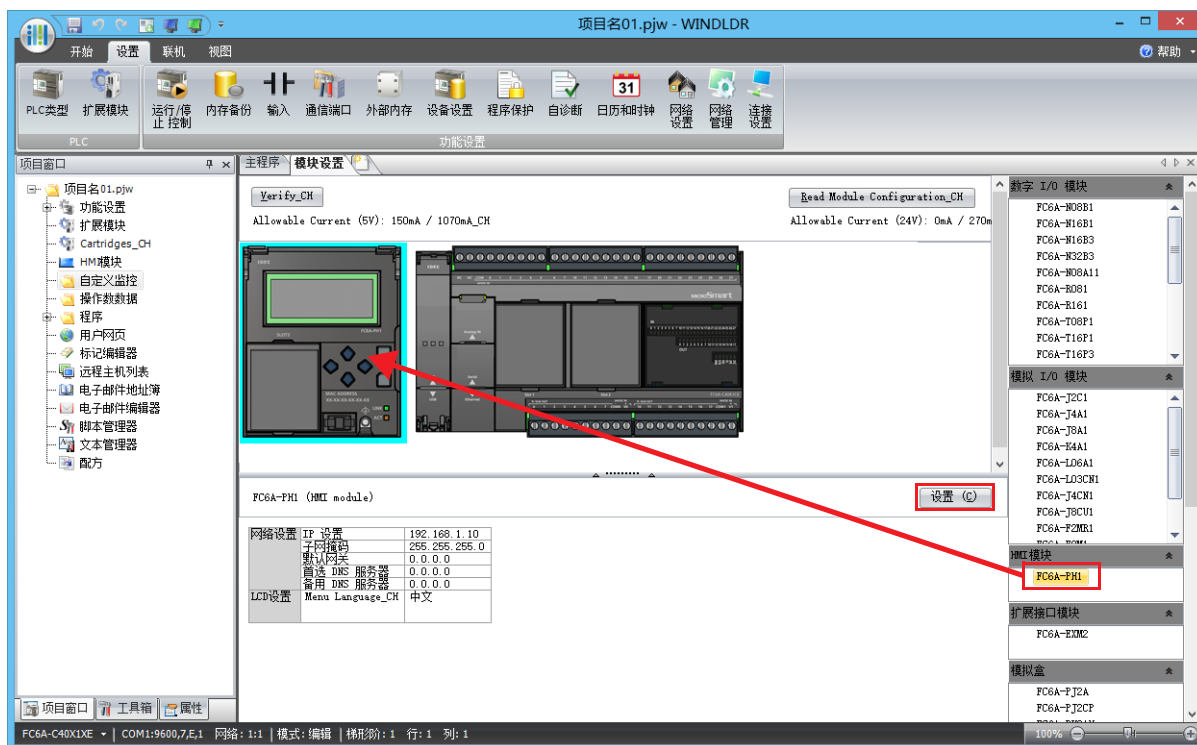
3. 根据通信设备的通信格式，指定各参数。



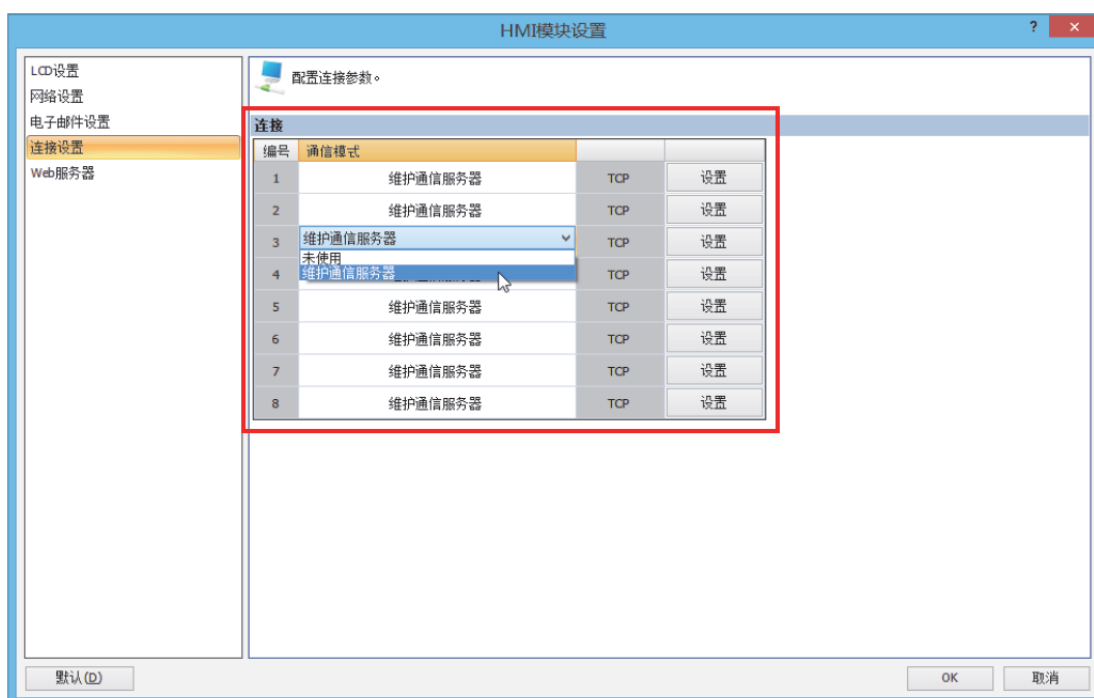
4. 单击“确定”。
5. 可通过模块构成编辑器进行 HMI 模块的连接设置。
在“设置”选项卡的“PLC”中单击“扩展模块”。

- 单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮。
将显示 HMI 模块的设置对话框。

注释：在项目窗口中双击“HMI 模块”，也可显示“HMI 模块的设置”对话框。



- 单击“连接设置”选项卡。
- 单击要使用连接编号的“通信模式”，选择通信模式。



3: 通信设置

9. 根据通信设备的通信格式，设置各参数。



10. 单击“确定”。

配置即完成。

远程主机列表

将对注册 / 管理网络上的通信设备（远程主机）的远程主机列表的设置进行介绍。

概述

FC6A 型在使用以下以太网通信功能时，将通信设备作为远程主机，需要注册到远程主机列表中。

- PING 指令
- ETXD/ERXD 指令（以太网用户通信）
- Modbus TCP 客户端
- MC 协议客户端

功能说明

远程主机由“IP 地址”或“主机名”及“端口号”组成。

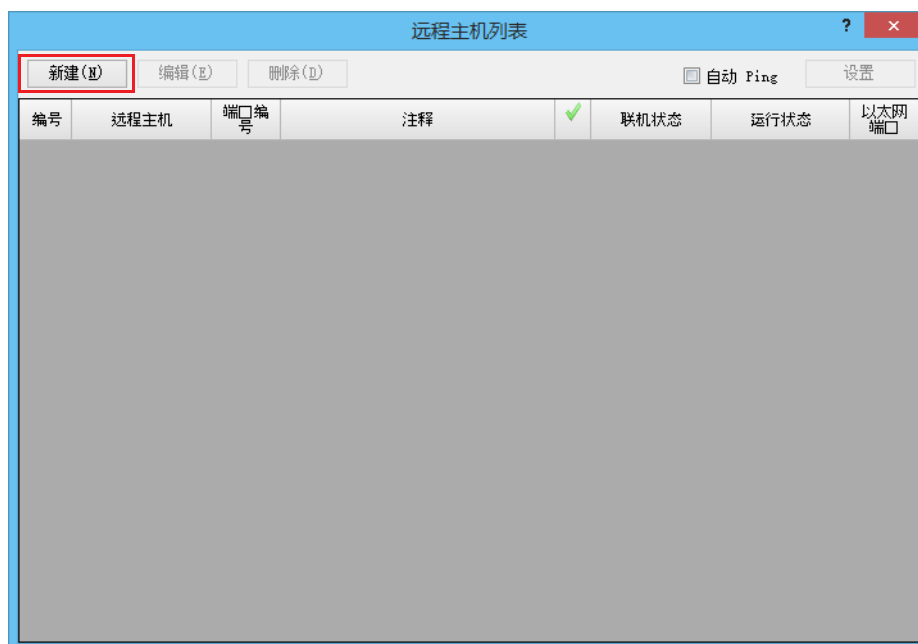
当用 IP 地址指定远程主机时，FC6A 型与具有指定 IP 地址和相应端口号的远程主机建立连接，然后启动通信。

当用主机名指定远程主机时，FC6A 型试图使用 DNS 服务器从指定的主机名中获取 IP 地址。如果成功获取 IP 地址，FC6A 型与具有指定 IP 地址和相应端口号的远程主机建立连接，然后启动通信。

有关 DNS 服务器设置的详情，请参见“功能设置”中的网络设置。

设置 WindLDR

1. 在“视图”选项卡的“工作区”中单击“项目窗口”。
将在画面左侧显示项目窗口。
2. 双击“远程主机列表”。
将显示“远程主机列表”对话框。
3. 单击“新建”按钮。
将显示“远程主机”对话框。



3: 通信设置

4. 输入到“远程主机”对话框的各项中，并单击“添加”按钮。

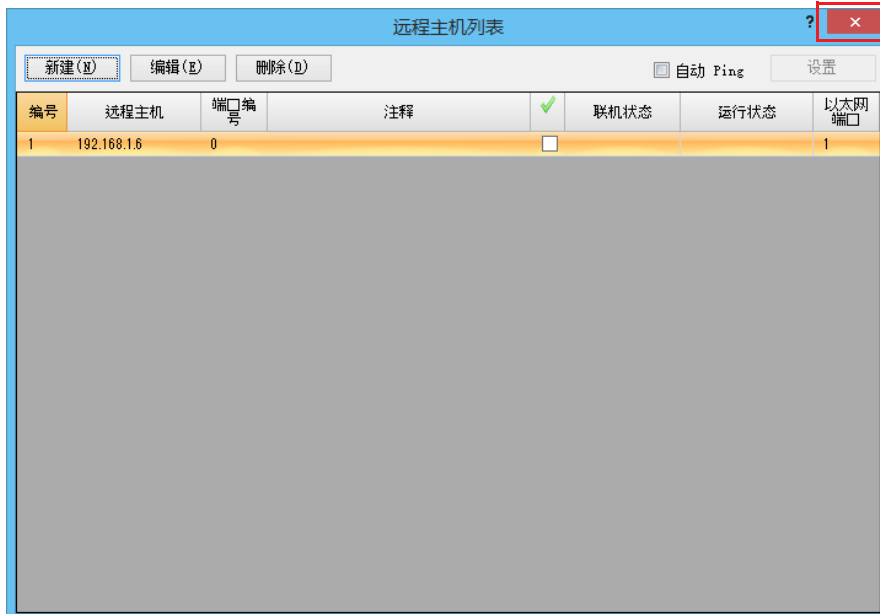


远程主机由以下项目构成。

项目	预置值
IP 地址	以 IP 地址设置远程主机。针对 FC6A 型设置的 IP 地址和其端口编号确立连接，并进行通信。
主机名称	按主机名称设置远程主机。最多可输入 40 个半角英文数字。
数据寄存器	在数据寄存器（4 字）中设置远程主机的 IP 地址。
端口	设置开启远程主机时的端口编号。此端口编号是 TCP 通信的端口编号，与 FC6A 型的 USB 端口、端口 1 至 3 的编号不同。
注释	该注释可说明远程主机。通信不受输入内容及长度的影响。
以太网端口	选择与远程主机通信的以太网端口。

注册远程主机，并反映到远程主机列表中。接着，要注册远程主机时，请重复进行相同操作。

- 如需针对设置的远程主机周期性执行 Ping，请设置自动 Ping。有关详情，请参见第 3-23 页上的“自动 Ping 功能”。
- 所有远程主机注册完成后，单击“关闭”按钮。



7. 要删除已注册的远程主机时，应打开“远程主机列表”对话框，选择要删除的远程主机后单击“删除”按钮。

配置即完成。

注释：从远程主机列表中删除远程主机后，之后的所有远程主机的远程主机编号均会发生变化。因此，参照这些远程主机列表编号的 PING 指令、Modbus TCP 客户端或用户通信客户端的动作会受到影响。

自动 Ping 功能

本节将介绍针对指定的远程主机周期性执行 Ping 的自动 Ping 功能。

功能说明

针对指定的远程主机周期性执行 Ping 的功能称作自动 Ping 功能。使用该功能，可通过向指定的远程主机发送数据组，确认可否与主机进行通信。自动 Ping 功能的结果可以与 Modbus TCP 和 MC 协议客户端的请求处理进行联动。有关与 Modbus TCP 客户端链接的详细信息，请参见“第 6 章 Modbus TCP 客户端”。有关与 MC 协议客户端链接的详细信息，请参阅“第 18 章通过以太网通信进行 MC 协议通信”。

远程主机通过远程主机列表进行指定。

自动 Ping 启用时，将在 FC6A 型的电源为 ON 或更改用户程序以启用自动 Ping 后，开始执行自动 Ping，并从指定的最小远程主机编号开始按顺序执行 Ping。FC6A 的运行、停止状态或梯形图程序不会受到影响。

在以太网端口 1 上，当 M8186（以太网端口 1 自动 Ping 运行中）和 M8187（以太网端口 1 自动 Ping 停止标记）处于 ON 时，自动 Ping 停止。当 M8187 为 OFF 时，将执行自动 Ping。

对于以太网端口 2，当 M8331（以太网端口 2 的自动 Ping 命令运行中）为 ON 并且 M8332（以太网端口 2 的自动 Ping 停止标记）为 ON 时，自动 Ping 命令将停止。当 M8332 为 OFF 时，自动 Ping 开始执行。以太网端口 2 只能与 PlusCPU 模块一起使用。

设置 WindLDR

1. 在“远程主机列表”对话框中选中“自动 Ping”复选框，然后单击“设置”按钮。
出现“自动 Ping 设置”对话框。



3: 通信设置

2. 指定自动 Ping 功能的各项目，然后单击“确定”按钮。



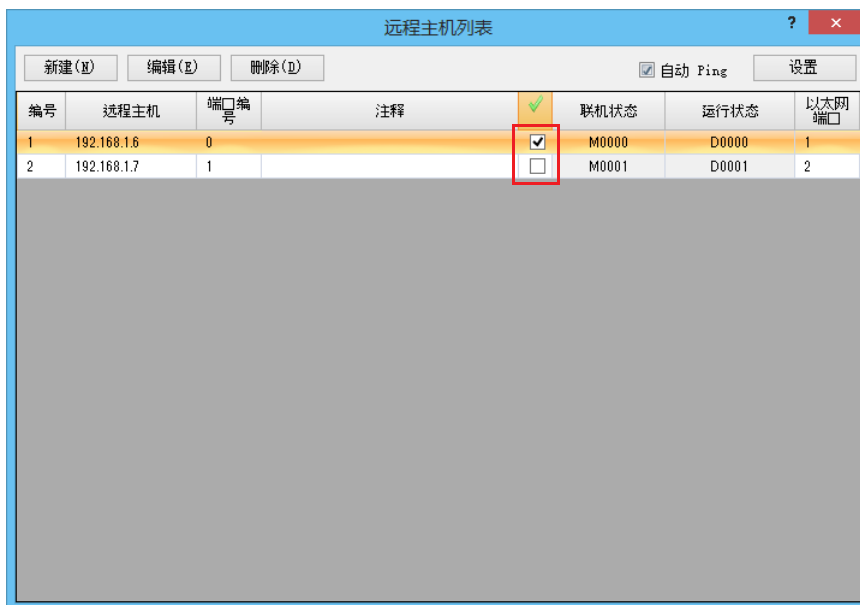
自动 Ping 功能由以下项目构成。

项目	预置值
联机状态*1	设置一个内部继电器来存储自动 Ping 功能发送的 Ping 结果。从设置的内部继电器开始，将按照远程主机编号的顺序自动分配内部继电器。
运行状态	指定存储自动 Ping 功能发送的 Ping 运行状态的数据寄存器。输入数据寄存器后，将在右侧显示占用的数据寄存器的范围。 运行状态表示运行的状态（状态代码）和错误内容（错误代码）。状态代码存储在高位 12 位中，错误代码存储在低位 4 位中。 有关各状态代码的内容的详情，请参见第 3-26 页上的“运行状态”。 PING 指令的动作则不会受到影响。
重试	指定自动 Ping 功能发送的 Ping 的重试次数（0 至 5 次）。
Ping 间隔	以 10ms 为单位指定自动 Ping 功能发送的各 Ping 的发送间隔（0 至 60,000ms）。
自动 Ping 间隔	以 100ms 为单位指定执行自动 Ping 后，到下次执行自动 Ping 为止的间隔（100 至 6,000,000ms）。

*1 当自动 Ping 功能开始执行时，[联机状态] 将自动关闭。此外，在发送 Ping 之后，[联机状态] 会在收到响应时打开，而在无法收到响应时自动关闭。自动 Ping 停止之前，[联机状态] 保持 ON / OFF 状态。特殊内部继电器的状态和自动 Ping 功能使用的 [联机状态] 如下。

与远程主机通信的以太网端口	联机状态的状态
以太网端口 1	当自动 Ping 功能开始执行时，[联机状态] 将自动关闭。此外，在发送 Ping 之后，[联机状态] 会在收到响应时打开，而在无法收到响应时自动关闭。自动 Ping 停止之前，[联机状态] 保持 ON / OFF 状态。特殊内部继电器的状态和自动 Ping 功能使用的 [联机状态] 如下。
以太网端口 2 (仅 Plus CPU 模块)	当执行以太网端口 2 自动 Ping (M8331 为 ON) 时，当将以太网端口 2 自动 Ping 停止标记 (M8332) 从 OFF 设置为 ON 并停止自动 Ping 时，将保持先前的 ON / OFF 状态。 停止自动 Ping (M8331 为 OFF) 时，如果将以太网端口 2 自动 Ping 停止标记 (M8332) 从 ON 更改为 OFF，则 [在线状态] 为 OFF，并且保持 OFF 状态。

- 在“远程主机列表”对话框中，选中执行自动 Ping 时发送 Ping 的远程主机的复选框。



- 单击“关闭”。
配置即完成。

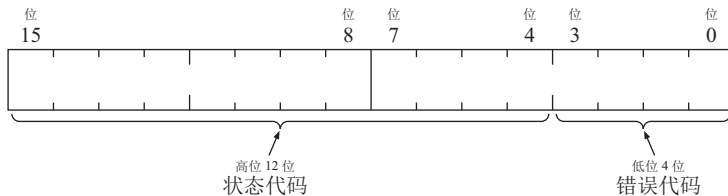
3: 通信设置

运行状态

运行状态表示运行的状态（状态代码）和错误内容（错误代码）。
 运行的状态（状态代码）存储在 D8414 的高位 12 位中，错误内容（错误代码）存储在 D8414 的低位 4 位中。
 运行状态的数据寄存器的值除以 16，其余数即为错误代码。

[运行状态的数据寄存器的值为“66”时]

$66 \div 16 = 4$ 余 2，
 则错误代码为“2”。



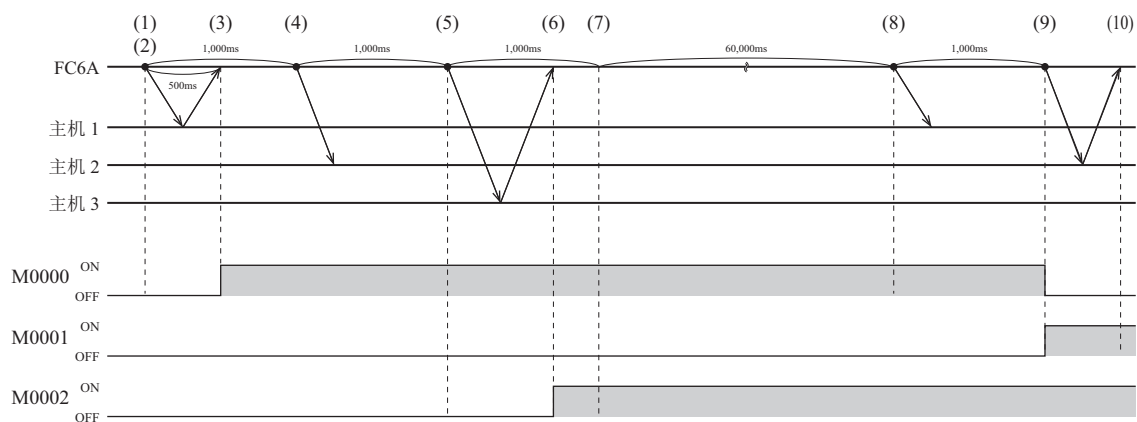
状态代码	运行的状态
16 (00000010000)	执行 Ping 后、发送数据组前的状态
32 (00000100000)	完成数据组发送处理后、等待来自主机的响应的状态
64 (00001000000)	正常接收数据组的响应、或发生超时错误，可执行下一 Ping 的状态

错误代码	错误内容
0 (0000)	正常
2 (0010)	超时错误
3 (0011)	未能 DNS 解决地址的主机名称

自动 Ping 动作示例 1

本节将对按以下条件对 3 个远程主机执行自动 Ping 时的动作进行介绍。

设置项目	设置内容	
Ping 设置	超时	1,000ms
	连接状态	M0000
自动 Ping 设置	重试	0 次
	Ping 间隔	1,000ms
	执行自动 Ping 的间隔	60,000ms



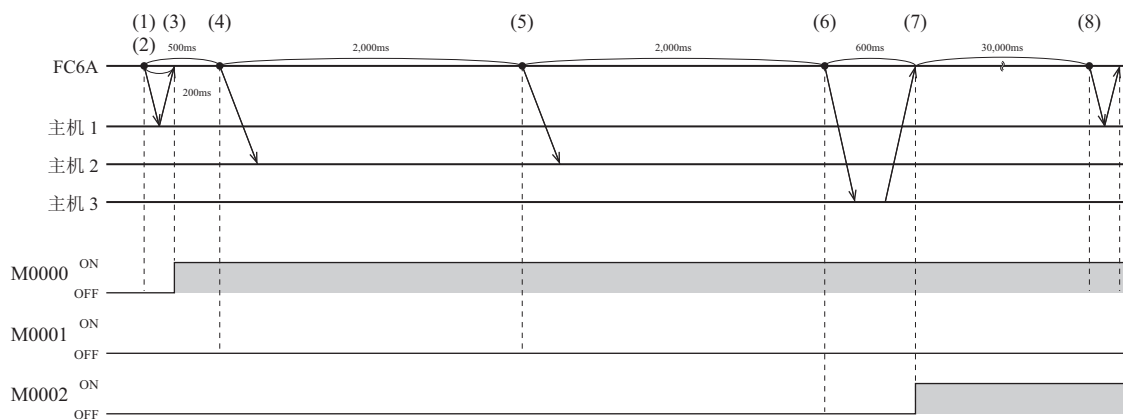
- (1) : 开始自动 Ping 动作
- (2) 至 (3) : 在超时时间内有来自主机 1 的响应
- (4) : 在超时时间内没有来自主机 2 的响应
- (5) 至 (6) : 在超时时间内有来自主机 3 的响应

- (7) 至 (8) : 执行自动 Ping 的间隔
 (8) : 在超时时间内没有来自主机 1 的响应
 (9) 至 (10) : 在超时时间内有来自主机 2 的响应

自动 Ping 动作示例 2

本节将对按以下条件对 3 个远程主机执行自动 Ping 时的动作进行介绍。

设置项目	设置内容
Ping 设置	超时 2,000ms
自动 Ping 设置	连接状态 M0000
	重试 1 次
	Ping 间隔 500ms
	执行自动 Ping 的间隔 30,000ms



- (1) : 开始自动 Ping 动作
 (2) 至 (3) : 在超时时间内有来自主机 1 的响应
 (4) 至 (5) : 在超时时间内没有来自主机 2 的响应，重试后仍无响应
 超时时间长于 Ping 间隔时，Ping 间隔没有等待时间。
 (6) : 来自主机 3 的响应延迟，但在超时时间内有响应
 (7) 至 (8) : 执行自动 Ping 的间隔

4: 维护通信

介绍

本章描述的是 FC6A 型维护通信功能。

维护通信是一种通信协议，专用于 IDEC 可编程控制器。当 WindLDR 或 IDEC 操作员界面与 FC6A 型通信时使用该协议。

FC6A 型通过各通信端口支持维护通信，可为各种系统配置选择最佳通信方法。

维护通信功能

与 FC6A 型进行维护通信时，可使用下列功能：

功能	说明
下载用户程序	可将 WindLDR 中创建的用户程序下载到 FC6A 型。 请参见《FC6A 型 MICROSmart 用户手册》第 4 章“下载程序”。
上传用户程序	可将 FC6A 型中存储的用户程序上传到 WindLDR。
监控 / 更改设备值	使用 WindLDR 可监控 FC6A 型的用户程序和设备值，并可对设备值进行更改。 请参见《FC6A 型 MICROSmart 用户手册》第 4 章“监控操作”。
下载系统软件	可将系统软件下载到 FC6A 型。 请参见《FC6A 型 MICROSmart 用户手册》附录“升级 FC6A 型系统软件”。

注释：要使用维护通信，请参见《FC6A 型 MICROSmart 用户手册》第 4 章“启动 WindLDR”，然后进行设置。

维护通信所使用的通信端口

支持的机型和端口以及槽如下所示。

端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
USB 端口	是	是	是	是	是	是
端口 1	是	是	是	—	是 *2	是 *2
端口 2	是 *1	是 *1	是 *1	是 *1	是 *2	是 *2
端口 3	—	—	是 *1	是 *1	是 *3	是 *3
端口 4 至 33	是（端口 4 至 9）*4				是（端口 4 至 33）*5	
以太网端口 1	是	是	是	是	是	是
以太网端口 2	—	—	—	—	是	是
HMI- 以太网端口 *6	是	是	是	是	是	是

*1 连接了通信盒时

*2 连接盒基本模块、连接了通信盒时

*3 连接 HMI 模块、连接了通信盒时

*4 All-in-One CPU 模块及 CAN J1939 All-in-One CPU 模块中最多可连接 3 台通信模块，以扩展端口 4 ~ 9。

*5 Plus CPU 模块中最多可连接 15 台通信模块，以扩展端口 4 ~ 33。

*6 CPU 模块中可连接 HMI 模块，以扩展 HMI- 以太网端口。

支持各通信端口的维护通信的通信方法如下所示。

通信方法		USB 端口	端口 1	端口 2 及 3	端口 4 至 33	以太网端口 1	以太网端口 2	HMI- 以太网端口
维护通信	下载系统软件	是	是 *1	否	否	是	是	是
	下载 / 上传用户程序	是	是 *1	否	否	是	是	是
	运行中下载	是	是 *1	否	否	是	是	是
	监控 / 更改设备	是	是	是	是	是	是	是

*1 通信盒时无法支持。

4: 维护通信

支持各通信端口以及槽的维护通信功能如下所示。

All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块

通信端口	标准 / 扩展	通信设置
USB 端口	标准	无
端口 1	标准 *1	功能设置。请参见第 4-4 页上的“端口 1 的维护通信”。
端口 2	扩展 *2	使用 RS232C/RS485 通信盒时： 功能设置。请参见第 4-13 页上的“端口 2 至 33 的维护通信”。
端口 3		使用 Bluetooth 通信盒时： 功能设置。请参见第 4-16 页上的“端口 1 ~ 3 (Bluetooth) 的维护通信”。
端口 4 至 9	扩展 *3	功能设置。请参见第 4-13 页上的“端口 2 至 33 的维护通信”。
以太网端口 1	标准	功能设置。请参见第 4-7 页上的“通过以太网端口 1 及 2 的维护通信”。
以太网端口 2	—	—
HMI- 以太网端口	标准 *4	功能设置。请参见第 4-17 页上的“HMI- 以太网端口的维护通信”。

*1 CAN J1939 All-in-One CPU 模块中无端口 1。

*2 CPU 模块中可连接通信盒，以扩展端口。

*3 可使用通信模块扩展端口。

*4 可使用 HMI 模块扩展端口。

Plus CPU 模块

通信端口	标准 / 扩展	通信设置
USB 端口	标准	无
端口 1	扩展 *1	使用 RS232C/RS485 通信盒时： 功能设置。请参见第 4-4 页上的“端口 1 的维护通信”。
端口 2		使用 Bluetooth 通信盒时： 功能设置。请参见第 4-6 页上的“端口 1 (Bluetooth) 的维护通信”。
端口 3	扩展 *2	使用 RS232C/RS485 通信盒时： 功能设置。请参见第 4-13 页上的“端口 2 至 33 的维护通信”。
端口 4 至 33	扩展 *3	使用 Bluetooth 通信盒时： 功能设置。请参见第 4-16 页上的“端口 1 ~ 3 (Bluetooth) 的维护通信”。
以太网端口 1	标准	功能设置。请参见第 4-13 页上的“端口 2 至 33 的维护通信”。
以太网端口 2		功能设置。请参见第 4-7 页上的“通过以太网端口 1 及 2 的维护通信”。
HMI- 以太网端口	扩展 *4	功能设置。请参见第 4-17 页上的“HMI- 以太网端口的维护通信”。

*1 盒基本模块中可连接通信盒，以扩展端口。

*2 HMI 模块中可连接通信盒，以扩展端口。

*3 可使用通信模块扩展端口。

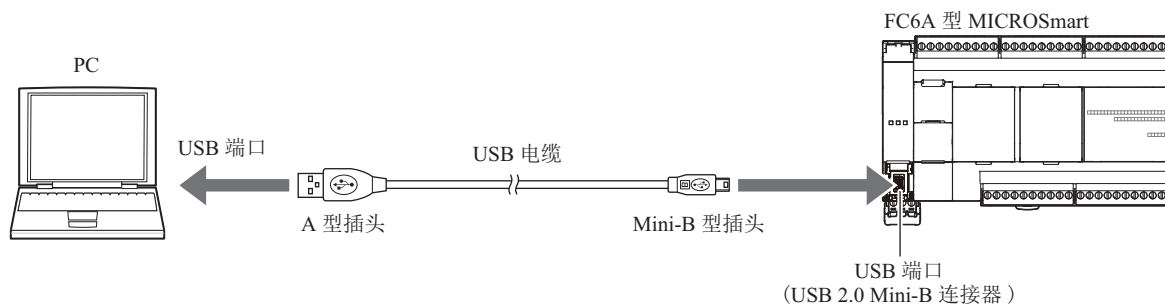
*4 可使用 HMI 模块扩展端口。

通过 USB 端口的维护通信

使用 USB 端口可将 FC6A 型连接至装有 WindLDR 的 PC，监控并更改设备值，下载和上传用户程序，以及下载系统软件。

使用 USB 电缆（建议的电缆为：HG9Z-XCM42）来连接 PC 和 FC6A 型。

FC6A 型的 USB 端口



维护通信规格

项目	规格 / 功能
电缆	建议的电缆：USB 维护电缆（HG9Z-XCM42）
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序 下载系统软件 运行中下载

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值：第 4 章“监控操作”
- 下载 / 上传用户程序：第 4 章“下载程序”
- 下载系统软件：附录“升级 FC6A 型系统软件”
- 运行中下载：第 5 章“联机编辑”

4: 维护通信

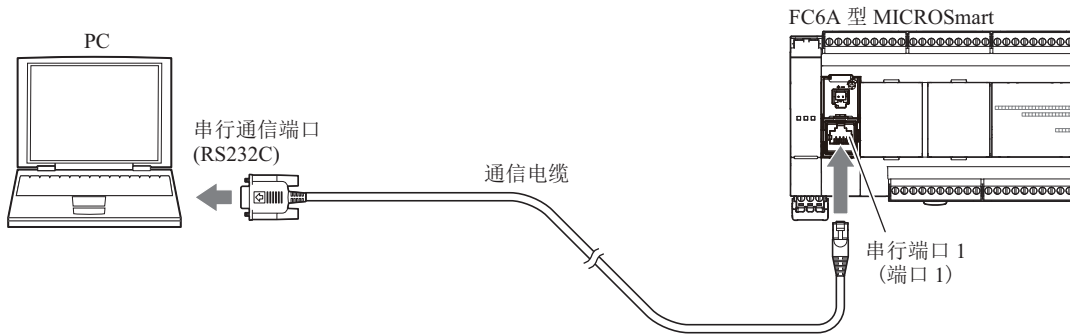
端口 1 的维护通信

可将 FC6A 型与安装有 WindLDR 的 PC 或 IDEC 生产的可编程显示器等进行连接，以监控或更改设备值、下载或上传用户程序、下载系统软件。

使用 All-in-One CPU 模块的串行端口 1，或通过使用 Plus CPU 模块上扩展的盒基本模块的盒插槽 1 所连接的通信盒，即可在端口 1 中进行维护通信。

有关通信电缆的详细信息，请参见《FC6A 型 MICROSmart 用户手册》附录“各种电缆”。

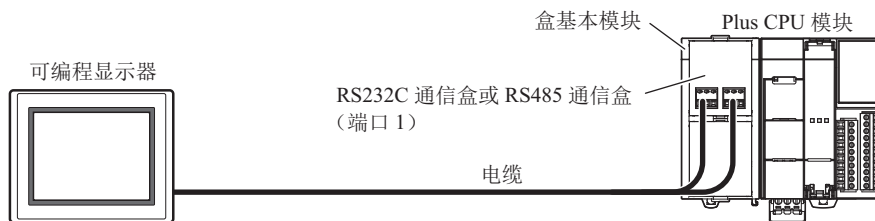
All-in-One CPU 模块的端口 1



维护通信规格

项目	规格 / 功能
电缆	O/I 通信电缆 (FC6A-KC2C)
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序 下载系统软件 运行中下载

Plus CPU 模块的端口 1



维护通信规格

项目	规格 / 功能
电缆	RS232C 通信盒: 多芯屏蔽电缆 RS485 通信盒: 屏蔽双绞电缆
维护通信功能	监控 / 更改设备值

注释:

使用 RS232C 通信盒及 RS485 通信盒的维护通信不支持以下功能。

- 下载 / 上传用户程序
- 下载系统软件
- 运行中下载

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值: 第 4 章“监控操作”

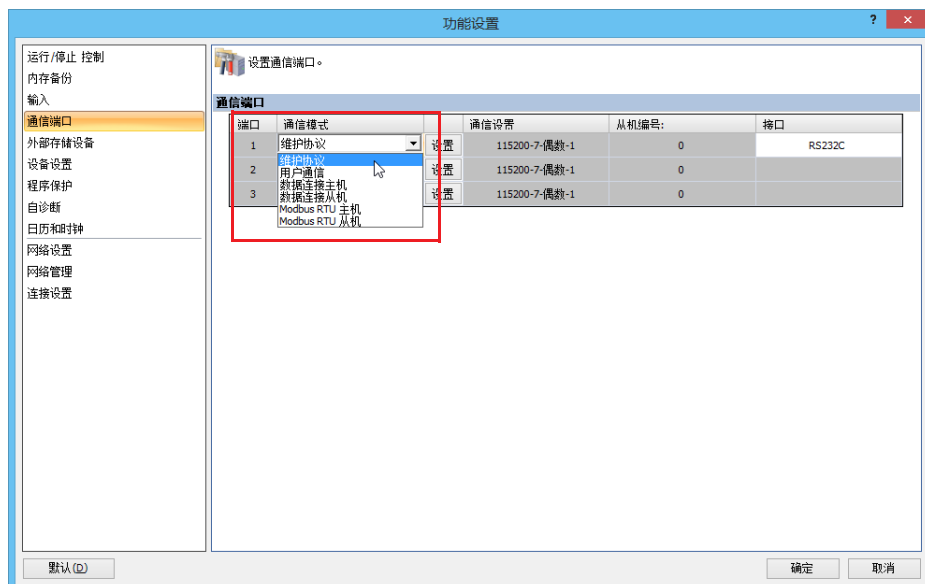
设置 WindLDR

配置维护通信的设置。

设置步骤

1. 从 WindLDR 菜单栏选择“设置”>“通信端口”。
出现“功能设置”对话框。

2. 单击端口 1 的“通信模式”，选择“维护协议”。
出现“维护协议（端口 1）”对话框。



3. 配置参数以匹配 PC 或可编程显示器的通信设置。



通信速度 (bps): 115200bps (1200、2400、4800、9600、19200、38400、57600、115200)
 数据长度: 7 (7 或 8)
 奇偶校验: 偶数 (无、偶数、奇数)
 停止位: 1 (1 或 2)
 接收超时 (毫秒): 500 (10 到 2550)
 从机编号: 0 (0 到 31)

从机编号可由常量或数据寄存器指定。

类型	详情
常量	在 0 到 31 的范围内进行设置
数据寄存器	在以下特殊数据寄存器中存储从机编号 0 到 31 端口 1: D8100 端口 2: D8102 端口 3: D8103 端口 4 至 9 ^{*1} : D8040 至 D8045 端口 10 至 33 ^{*1*2} : D8735 至 D8758

*1 使用通信模块时

*2 仅 Plus CPU 模块

* 不在括号中的值为默认值。

* 通过端口 1 下载系统软件时，请将“数据长度”设为“8”。

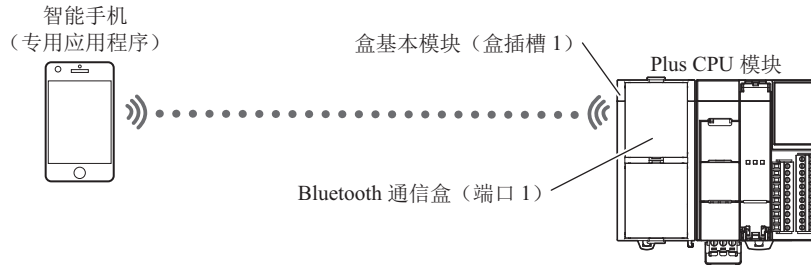
4. 单击“确定”。
为扩展通信端口配置维护通信的操作现已完成。

4: 维护通信

端口 1 (Bluetooth) 的维护通信

通过将 Bluetooth 通信盒连接到 Plus CPU 模块上扩展的盒基本模块的盒插槽 1，可与装配有 Bluetooth 的 PC 或智能手机进行连接，以监控或更改设备值。

Plus CPU 模块的 Bluetooth



维护通信规格

项目	规格 / 功能
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序

注释:

- 使用 Bluetooth 通信盒的维护通信不支持以下功能。
 - 下载系统软件
 - 运行中下载
- 如果下载可更改 Bluetooth 通信盒设置的用户程序，则在下载结束时可能会导致无法进行 Bluetooth 通信而发生下载错误。

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值：第 4 章“监控操作”

设置 WindLDR

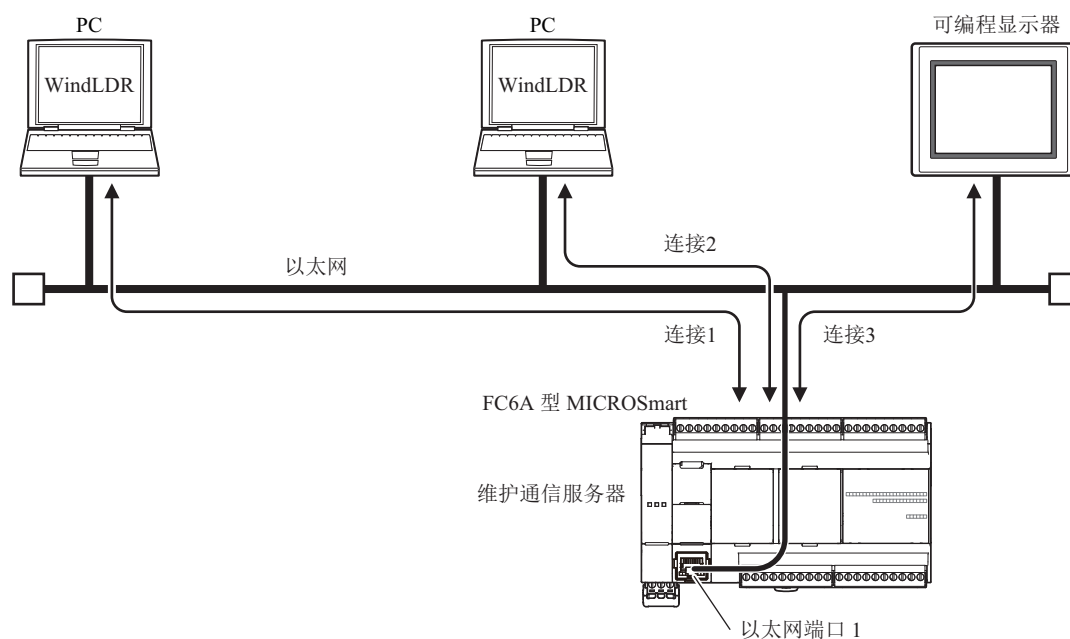
有关详情，请参见第 9-1 页上的“Bluetooth 通信”。

通过以太网端口 1 及 2 的维护通信

可使用 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的以太网端口 1、Plus CPU 模块的以太网端口 1 及 2，与 PC 或可编程显示器等网络设备进行通信。网络中的外部设备可监控或更改设备值，以及下载或上传用户程序。

可为 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配维护通信服务器或 Modbus TCP 通信等各种不同的通信功能，且能同时使用维护通信服务器功能以及其他通信功能。

FC6A 型的以太网端口 1 及 2



维护通信规格

项目	规格 / 功能
电缆	以太网电缆 (Cat 5. STP)
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序 下载系统软件 运行中下载

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值：第 4 章“监控操作”
- 下载 / 上传用户程序：第 4 章“下载程序”
- 下载系统软件：附录“升级 FC6A 型系统软件”
- 运行中下载：第 5 章“联机编辑”

4: 维护通信

设置 WindLDR

本节描述了为以太网端口 1 及 2 配置维护通信服务器以及通过以太网与 FC6A 型进行通信的步骤。

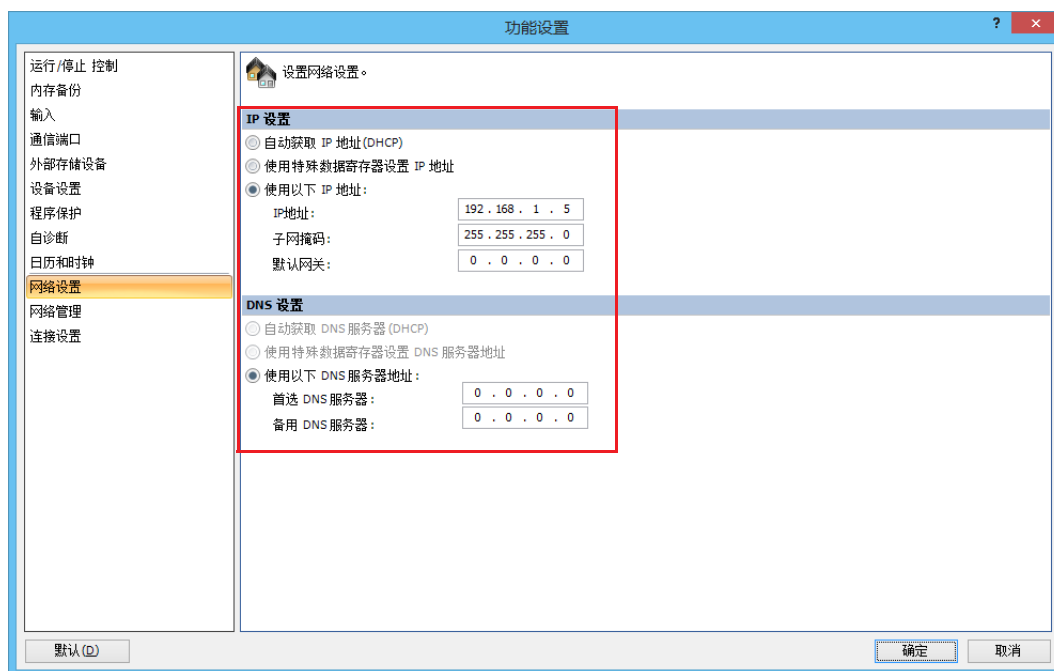
配置维护通信服务器

1. 在“设置”选项卡的“功能设置”中单击以下项目。

- All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块
“网络设置”
- Plus CPU 模块
设置以太网端口 1 时为“以太网端口 1”
设置以太网端口 2 时为“以太网端口 2”

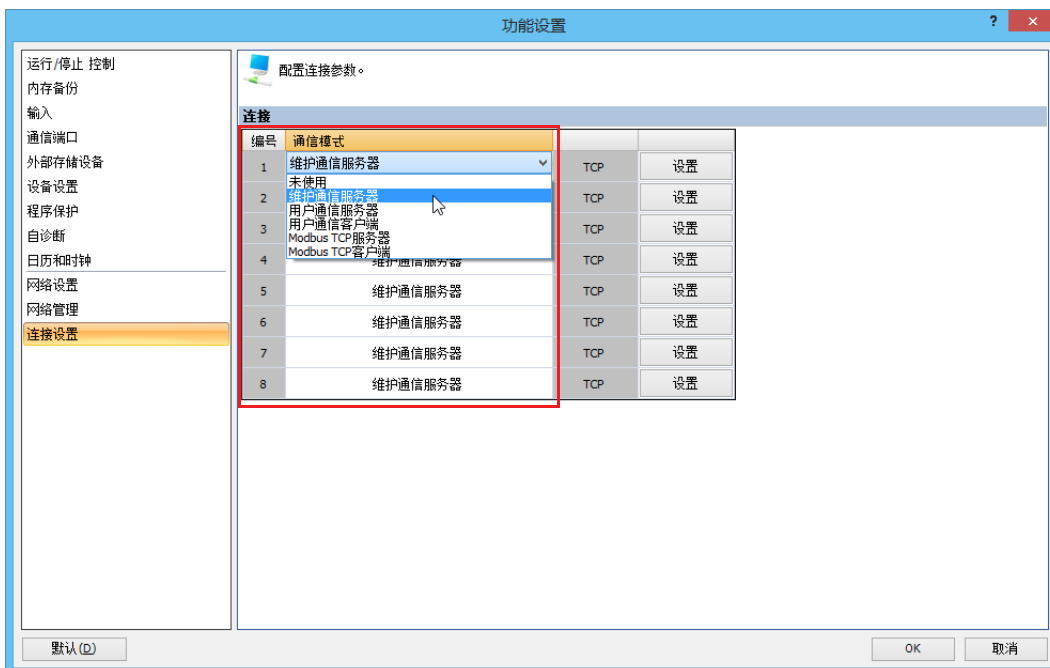
出现“功能设置”对话框。

2. 输入 IP 地址、子网掩码和默认网关。



3. 单击“连接设置”。

4. 单击要使用连接的“通信模式”，选择“维护通信服务器”。
出现“维护通信服务器”对话框。



5. 设置参数以匹配 PC 或可编程显示器的通信设置。



本地主机端口号: 2101 (FC6A 型用于维护通信服务器的端口号)

接收超时 (毫秒): 2000 (100 到 25500ms)

允许 IP 地址访问: 禁用 (启用该选项可只允许具有输入的 IP 地址的设备进行访问。)

启用通过 Modbus RTU (端口 1) 的直通:

可将端口 1 用作 Pass-Through 端口。此时，请将端口 1 的“通信模式”设为“Modbus RTU 主机”。请将用作 Pass-Through 端口的端口的本地主机端口号指定为与其他服务器连接的端口号不同的编号。

注释: 可同时连接至 FC6A 型的客户端数为每个连接一个客户端。如果连接 1 到连接 8 都设置为维护通信服务器，那么八个客户端可同时连接至 FC6A 型。

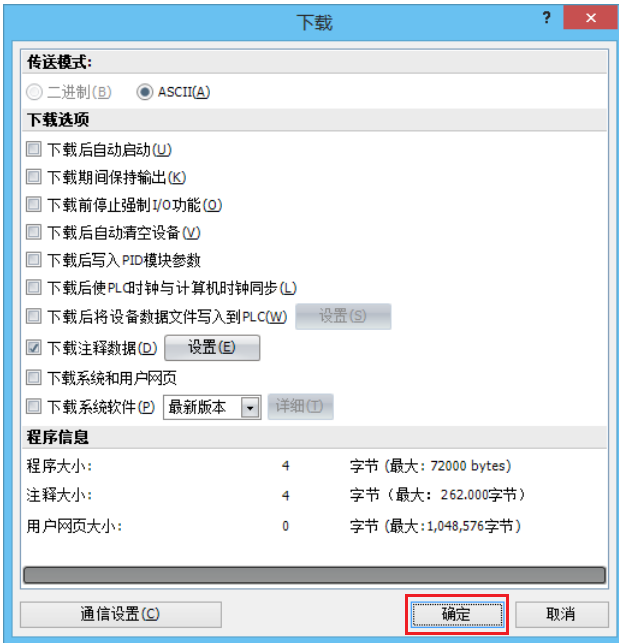
6. 单击“确定”。
完成维护通信设置。

4: 维护通信

通过 USB 端口下载用户程序和确认 IP 地址

开始以太网通信前，配置功能设置并通过 USB 将用户程序下载到 FC6A 型。

7. 使用 USB 电缆来连接 PC 和 FC6A 型。
8. 从 WindLDR 菜单栏选择“联机”>“传送”>“下载”。
出现“下载”对话框。
9. 单击“确定”。
用户程序下载至 FC6A 型。



10. 成功下载用户程序后，转到监控模式以检查 FC6A 型的状态。从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”。
11. 从 WindLDR 菜单栏选择“联机”>“监控”>“批处理”。
出现“多点监控”对话框。
12. 确认步骤 2 中输入的 IP 地址是否正确显示在 D8330 到 D8333 中。

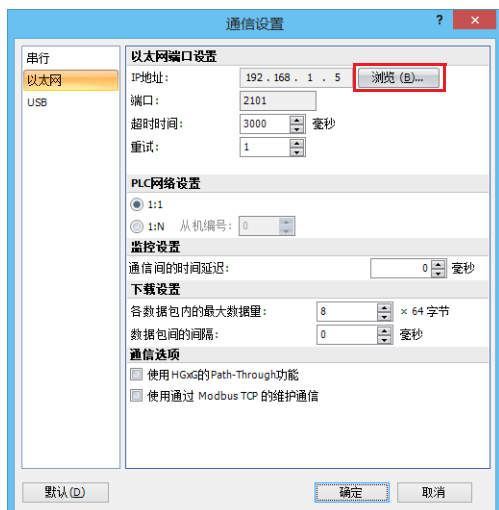
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D8330	192	168	1	5	255	255	255	0	0	0
D8340	0	0	0	0	0	0	0	0	0	0
D8350	0	0	0	0	0	0	0	0	0	0
D8360	0	0	0	0	0	0	0	0	0	0
D8370	0	0	0	0	0	0	0	0	0	0
D8380	0	0	0	0	0	0	0	0	0	0
D8390	0	0	0	0	0	0	0	0	0	0
D8400	0	0	0	0	0	0	0	0	0	0
D8410	0	0	0	0	0	0	0	0	0	0
D8420	0	0	0	0	0	0	0	0	0	0
D8430	0	0	0	0	0	0	0	0	0	0
D8440	0	0	0	0	0	0	0	0	0	0
D8450	0	0	0	0	0	0	0	0	0	0
D8460	0	0	0	0	0	0	0	0	0	0
D8470	33023	0	33023	0	33023	0	33023	0	33023	0
D8480	33023	0	33023	0	33023	0	33023	0	33023	0
D8490	33023	0	33023	0	33023	0	33023	0	33023	0

至此，通过 USB 端口进行的用户程序下载和 IP 地址确认完成。

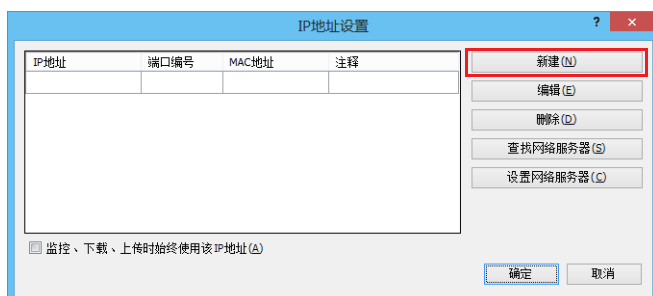
通过以太网端口 1 及 2 监控 FC6A 型

使用 WindLDR 通过以太网监控 FC6A 型。

- 从 WindLDR 菜单栏选择“联机”>“通信”>“设置”。
出现“通信设置”对话框。
- 选择“以太网”选项卡，然后单击“浏览”。
出现“IP 地址设置”对话框。



- 单击“新建”。
出现“输入 IP 地址”对话框。



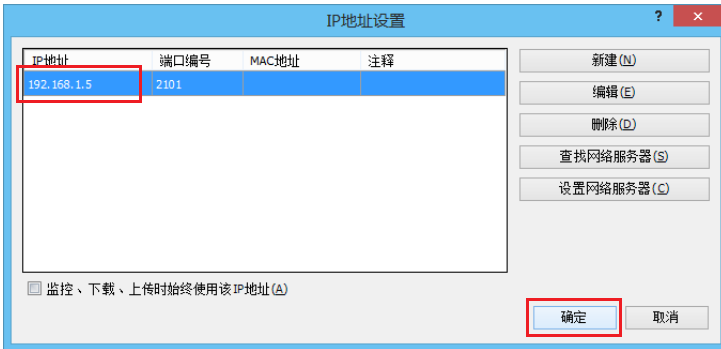
- 输入在步骤 2 中输入的 IP 地址，然后单击“确定”。



- 从 WindLDR 菜单栏选择“联机”>“监控”>“监控”。
出现“IP 地址设置”对话框。

4: 维护通信

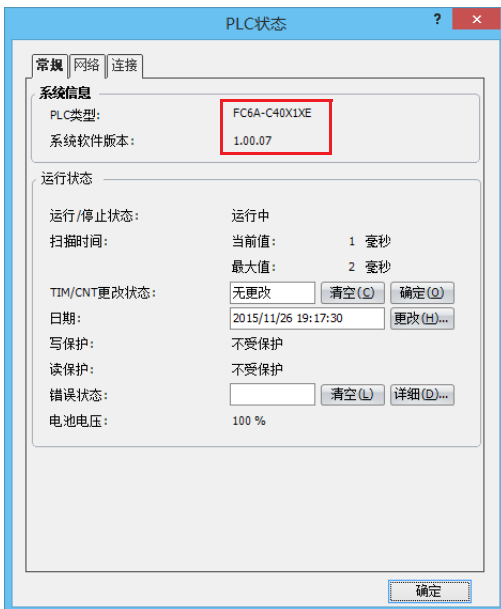
18. 选择您输入的 IP 地址，然后单击“确定”。



19. 从 WindLDR 菜单栏选择“联机”>“PLC”>“状态”。

出现“PLC 状态”对话框。

20. 检查 FC6A 型模块类型和系统软件版本是否正确显示。



为以太网配置维护通信服务器初始设置的操作现已完成。您可以通过以太网下载和上传用户程序，以及监控和更改设备值。

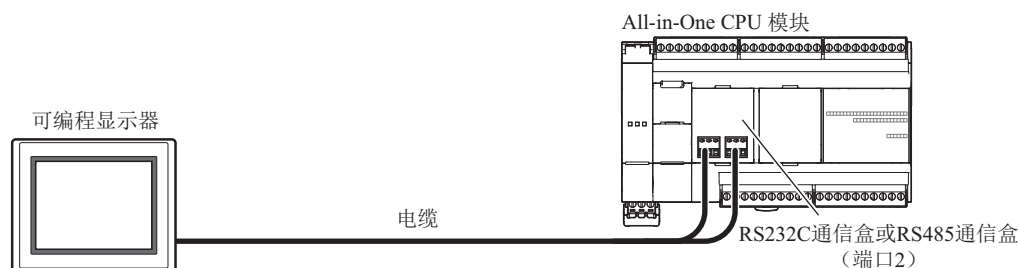
端口 2 至 33 的维护通信

通过对 FC6A 型进行以下扩展，可与装配有 RS232C 或 RS485 端口的 PC 或可编程显示器进行连接，以监控或更改设备值。

- 将 RS232C 通信盒或 RS485 通信盒连接到 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的盒插槽
- 将 RS232C 通信盒或 RS485 通信盒连接到 Plus CPU 模块上扩展的盒基本模块的盒插槽 2
- 将 RS232C 通信盒或 RS485 通信盒连接到 Plus CPU 模块上扩展的 HMI 模块的盒插槽
- 将通信模块连接到 FC6A 型

有关通信电缆的详情，请参见《FC6A 型 MICROSmart 用户手册》附录“各种电缆”。

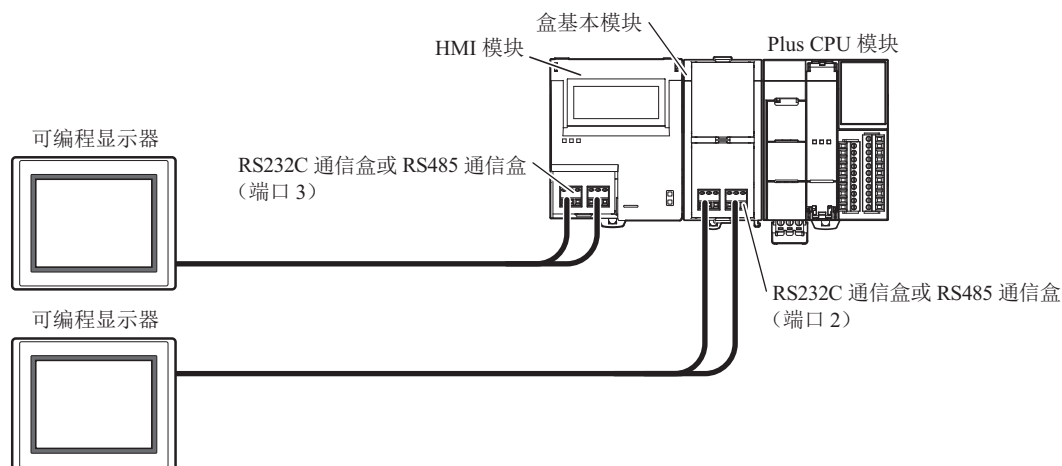
All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的端口 2 及 3



维护通信规格

项目	规格 / 功能
电缆	RS232C 通信盒：多芯屏蔽电缆 RS485 通信盒：屏蔽双绞电缆
维护通信功能	监控 / 更改设备值

Plus CPU 模块的端口 2 及 3

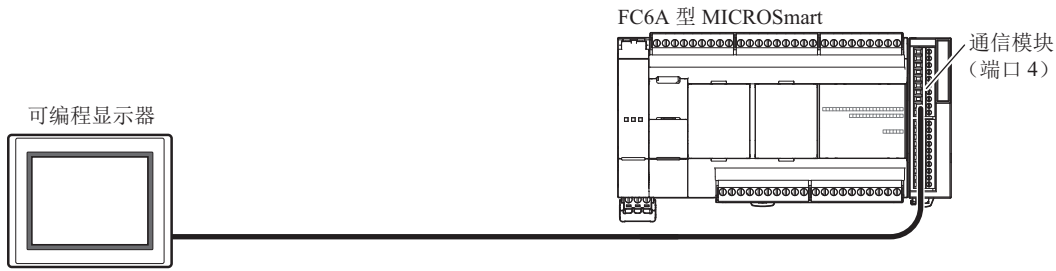


维护通信规格

项目	规格 / 功能
电缆	RS232C 通信盒：多芯屏蔽电缆 RS485 通信盒：屏蔽双绞电缆
维护通信功能	监控 / 更改设备值

4: 维护通信

FC6A 型的端口 4 ~ 33



All-in-One CPU 模块及 CAN J1939 All-in-One CPU 模块最多可连接 3 台通信模块，以扩展端口 4 ~ 9。
Plus CPU 模块最多可连接 15 台通信模块，以扩展端口 4 ~ 33。有关通信模块的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“通信模块”。

维护通信规格

项目	规格 / 功能
电缆	RS232C 通信盒: 多芯屏蔽电缆 RS485 通信盒: 屏蔽双绞电缆
维护通信功能	监控 / 更改设备值

维护通信的各功能操作的详情

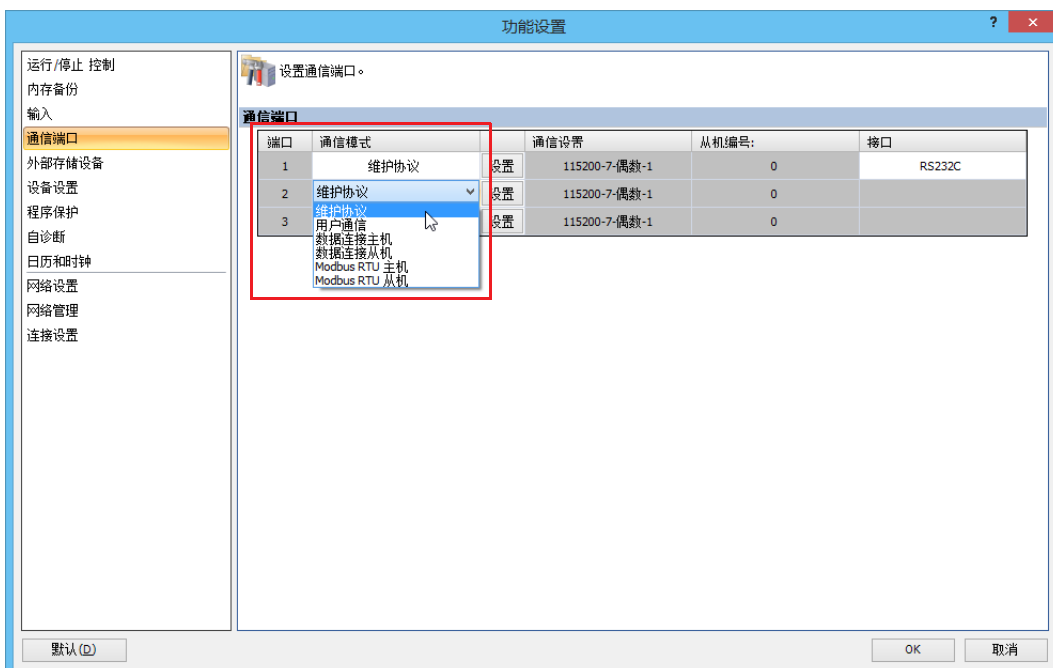
请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值: 第 4 章“监控操作”

设置 WindLDR

进行用于实施维护通信的各项设置。

1. 从 WindLDR 菜单栏中，选择“设置”>“功能设置”>“通信端口”。
出现“功能设置”对话框。
2. 单击端口 2 至 33 的“通信模式”，选择“维护协议”。
出现“维护协议（端口 n）”对话框。



3. 根据 PC 或可编程显示器的通信格式，设置各参数。



通信速度 (bps) : 115200bps (1200、2400、4800、9600、19200、38400、57600、115200)
 数据长度 : 7 (7 或 8)
 奇偶校验 : 偶数 (无、偶数、奇数)
 停止位 : 1 (1 或 2)
 接收超时 (毫秒) : 500 (10 至 2550)
 从机编号 : 0 (0 至 31)

从机编号可由常量或数据寄存器指定。

类型	详情
常量	在 0 至 31 的范围内进行设置
数据寄存器	在以下特殊数据寄存器中存储从机编号 0 至 31 端口 1: D8100 端口 2: D8102 端口 3: D8103 端口 4 至 9*1: D8040 至 D8045 端口 10 至 33*1*2: D8735 至 D8758

*1 使用通信模块时

*2 仅 Plus CPU 模块

* 不在括号中的值为默认值。

4. 单击“确定”按钮。

至此，维护通信的设置完成。

4: 维护通信

端口 1 ~ 3 (Bluetooth) 的维护通信

FC6A 型可与装配有 Bluetooth 的 PC 或智能手机进行连接，以监控或更改设备值。

通过将 Bluetooth 通信盒连接到盒插槽 1、2、(Plus CPU 模块时还需连接到盒插槽 3)，即可进行 Bluetooth 通信。

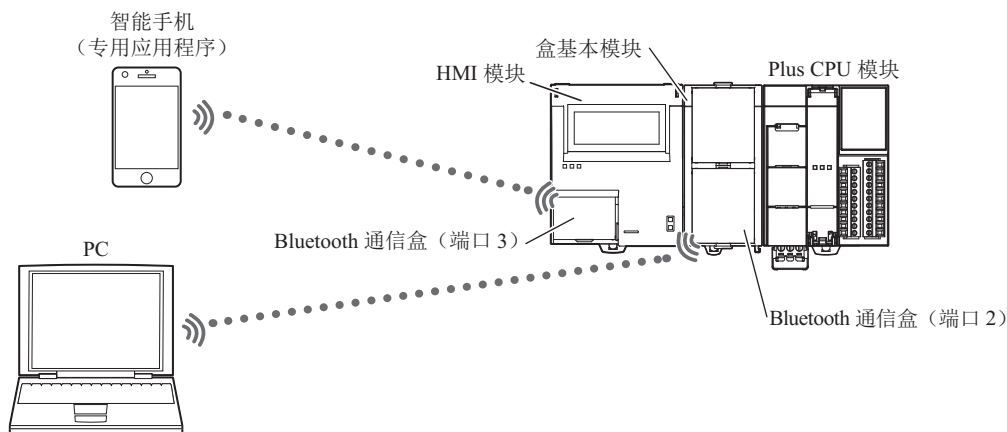
All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块的 Bluetooth



维护通信规格

项目	规格 / 功能
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序

Plus CPU 模块的 Bluetooth



维护通信规格

项目	规格 / 功能
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序

注释:

- 使用 Bluetooth 通信盒的维护通信不支持以下功能。
 - 下载系统软件
 - 运行中下载
- 如果下载可更改 Bluetooth 通信盒设置的用户程序，则在下载结束时可能会导致无法进行 Bluetooth 通信而发生下载错误。

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值: 第 4 章 “监控操作”

设置 WindLDR

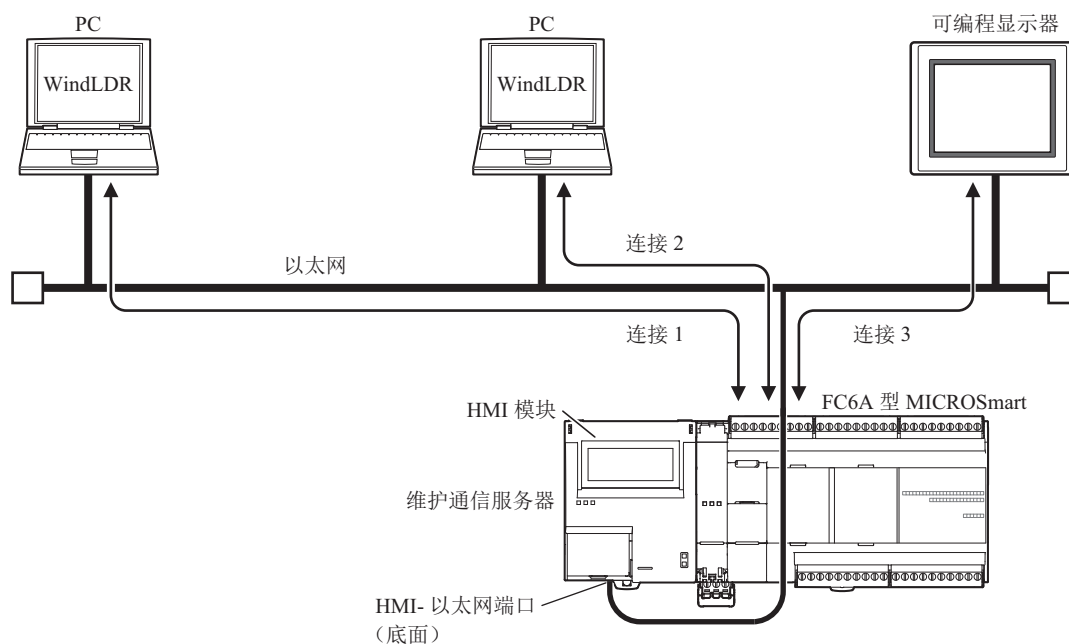
有关详情，请参见第 9-1 页上的“Bluetooth 通信”。

HMI- 以太网端口的维护通信

使用连接在 CPU 模块中 HMI 模块的 HMI- 以太网端口，可与 PC 或可编程显示器等的网络设备进行维护通信。可通过连接到网络的外部设备监控或更改 FC6A 型的设备值，并下载或上传用户程序。

FC6A 型通过使用 HMI- 以太网端口，不仅可实现标配于 All-in-One CPU 模块的以太网端口 1 所拥有的连接（最多 8 个）、标配于 Plus CPU 模块的以太网端口 1 及 2 所拥有的连接（最多 16 个），还可扩展最多 8 个连接。可向通过 HMI- 以太网端口所扩展的连接（最大 8 个）分配维护通信服务器。

HMI 模块的 HMI- 以太网端口 1



维护通信规格

项目	规格 / 功能
电缆	以太网电缆 (Cat 5. STP)
维护通信功能	监控 / 更改设备值 下载 / 上传用户程序 下载系统软件 运行中下载

维护通信的各功能操作的详情

请参见《FC6A 型 MICROSmart 用户手册》的以下内容。

- 监控 / 更改设备值：第 4 章“监控操作”
- 下载 / 上传用户程序：第 4 章“下载程序”
- 下载系统软件：附录“升级 FC6A 型系统软件”
- 运行中下载：第 5 章“联机编辑”

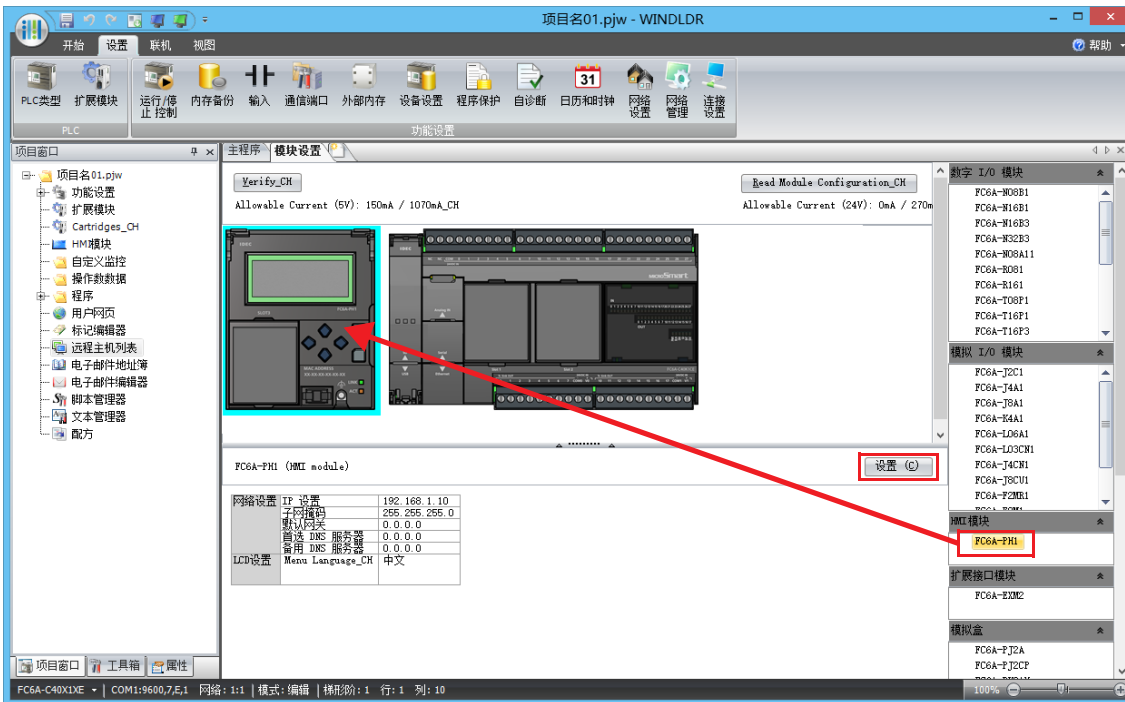
4: 维护通信

设置 WindLDR

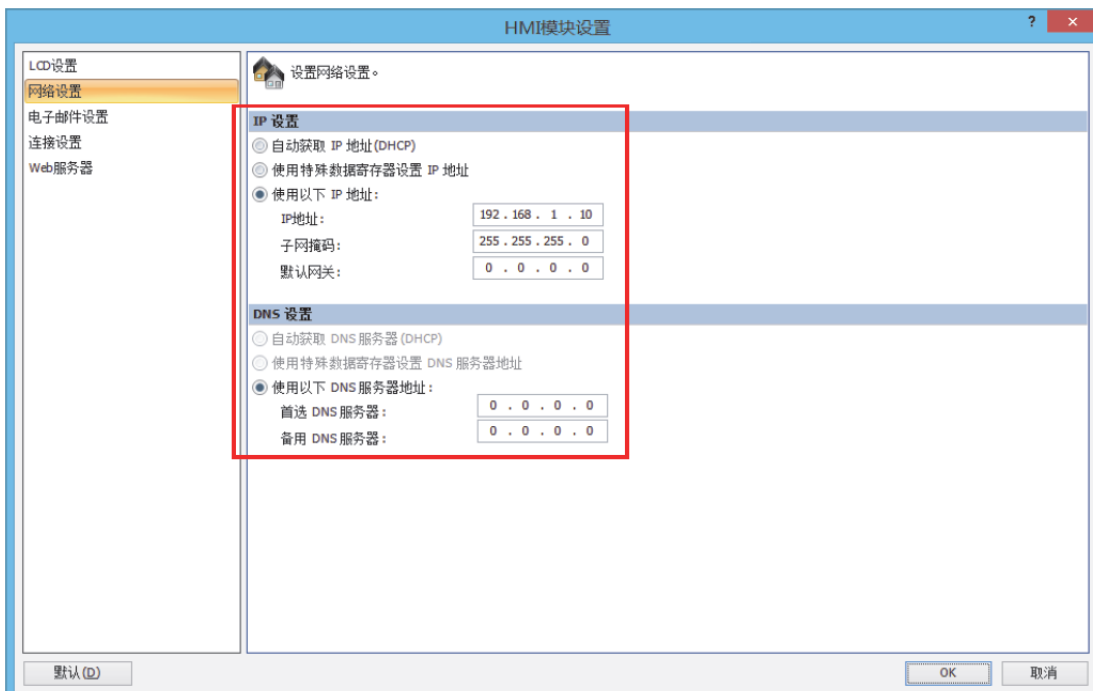
配置维护通信的设置。

1. 可通过模块构成编辑器进行 HMI 模块的网络设置及连接设置。
在“设置”选项卡的“PLC”中单击“扩展模块”。
2. 单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮。
将显示 HMI 模块的设置对话框。

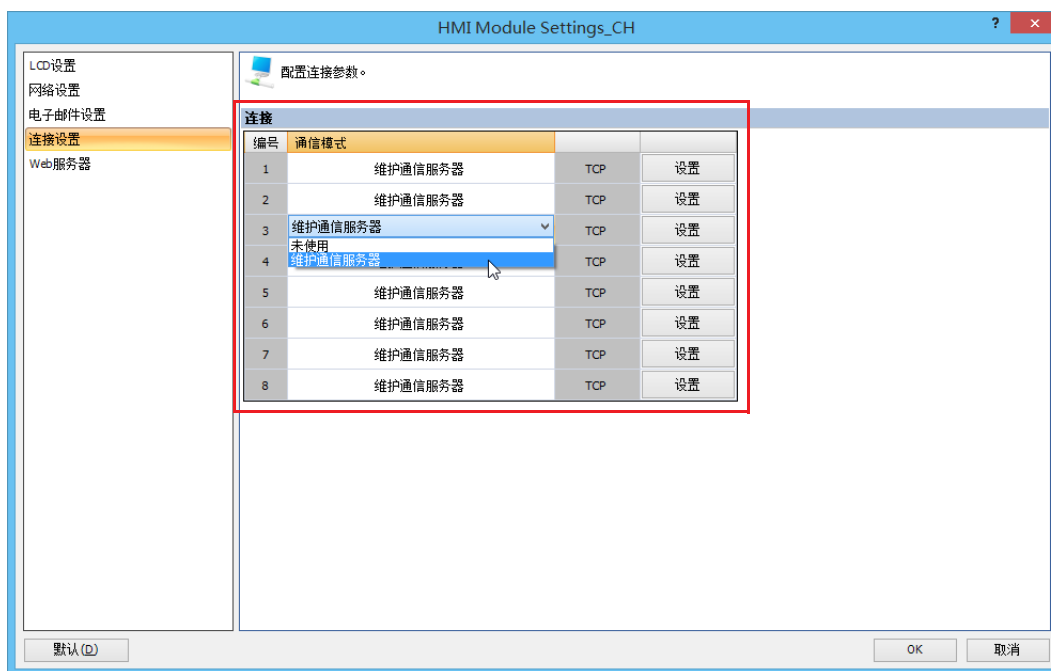
注释: 在项目窗口中双击“HMI 模块”，也可显示 HMI 模块的设置对话框。



3. 单击“网络设置”选项卡，分别对“IP 设置”、“DNS 设置”进行设置。



4. 单击“连接设置”。
5. 单击要使用连接编号的“通信模式”，选择通信模式。



6. 设置参数以匹配 PC 或可编程显示器的通信设置。



- 本地主机端口号： 2101 (FC6A 型用于维护通信服务器的端口号)
- 接收超时 (毫秒)： 2000 (100 到 25500ms)
- 允许 IP 地址访问： 禁用 (启用该选项可只允许具有输入的 IP 地址的设备进行访问。)

注释：可同时连接至 FC6A 型的客户端数为每个连接一个客户端。如果连接 1 到连接 8 都设置为维护通信服务器，那么八个客户端可同时连接至 FC6A 型。

7. 单击“确定”。
- 完成维护通信设置。

5: 用户通信指令

简介

本章对将设置数据转换为符合 FC6A 型所连接外部设备的数据类型后进行收发的用户通信进行介绍。

执行用户通信时，应使用用户通信指令。

用户通信指令因使用的通信接口所使用的用户通信指令而异。

- 以串行通信（RS232C 或 RS485）的方式与使用端口 1 至 33 的外部设备进行用户通信时
以串行通信（Bluetooth）的方式与使用端口 1 至 3 的外部设备进行用户通信时
第 5-2 页上的“TXD（发送）”
第 5-10 页上的“RXD（接收）”
- 以太网通信的方式与使用以太网端口 1 及 2 的外部设备进行用户通信时
第 5-23 页上的“ETXD（以太网用户通信发送）”
第 5-23 页上的“ERXD（以太网用户通信接收）”

用户通信中使用的通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
端口 1	是	是	是	—	是 ^{*2}	是 ^{*2}
端口 2	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*2}	是 ^{*2}
端口 3	—	—	是 ^{*1}	是 ^{*1}	是 ^{*3}	是 ^{*3}
端口 4 至 33	是（端口 4 至 9） ^{*4}			—	是（端口 4 至 33） ^{*5}	
以太网端口 1	是	是	是	是	是	是
以太网端口 2	—	—	—	—	是	是
HMI- 以太网端口	—	—	—	—	—	—

*1 连接了通信盒时

*2 连接盒基本模块、连接了通信盒时

*3 连接 HMI 模块、连接了通信盒时

*4 All-in-One CPU 模块及 CAN J1939 All-in-One CPU 模块中最多可连接 3 台通信模块，以扩展端口 4 ~ 9。

*5 Plus CPU 模块中最多可连接 15 台通信模块，以扩展端口 4 ~ 33。

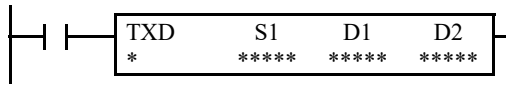
设置用户通信

各通信端口支持的用户通信的设置如下所示。

- 使用端口 1 至 33，与外部设备进行串行通信（RS232C/RS485）
- 使用端口 1 至 3，与外部设备进行串行通信（Bluetooth）
第 5-24 页上的“通过串行通信进行的用户通信”
- 与通过以太网端口 1 及 2 连接的外部设备的以太网通信
第 5-40 页上的“通过以太网通信进行的用户通信”

5: 用户通信指令

TXD（发送）



使用端口 1 至 33，将发送数据转为已设置的数据类型，并发送到外部设备。送至配备 RS232C/RS485 端口的远程终端。

有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	发送数据	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	发送完成输出	—	X	X ^{*1}	—	—	—	—	—	—	—
D2（目标 2）	发送状态寄存器	—	—	—	—	—	—	X ^{*2}	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

*1 不能将特殊内部继电器指定为 D1。

*2 无法使用特殊数据寄存器。

由设备 S1 指定的数据最多发送 1,536 字节。

发送完成后，将打开由设备 D1 指定的输出或内部继电器。

目标 2 占据两个 D2 指定的设备开始的连续数据寄存器。发送状态数据寄存器存储发送状态和错误代码。下一个数据寄存器存储已发送数据的字节计数。不能将同一数据寄存器用作 TXD1 ~ TXD33 指令的发送状态寄存器及 RXD1 ~ RXD33 指令的接收状态寄存器。

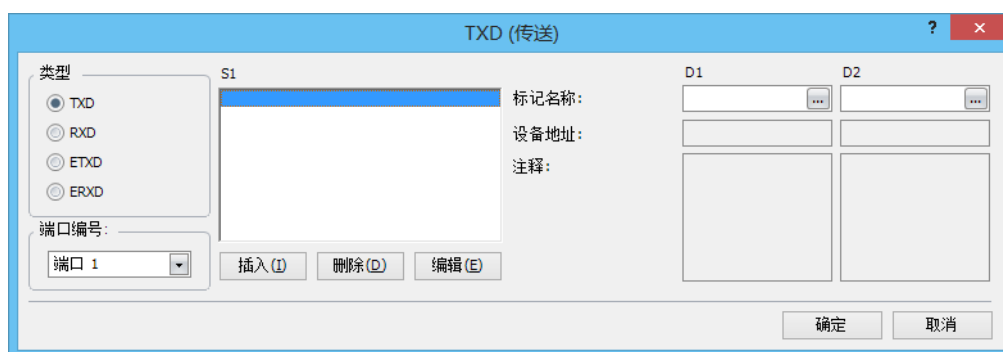
注释：

- 有关串行通信的用户通信的详情，请参见第 5-24 页上的“通过串行通信进行的用户通信”。
- 有关 Bluetooth 通信的用户通信的详情，请参见第 5-39 页上的“通过串行通信（Bluetooth）进行的用户通信”。

编程 TXD 指令的注意事项

- FC6A 型有五个格式设置区域，用于执行 TXD1 ~ TXD33 指令，所以可以同时执行各五个 TXD1 ~ TXD33 指令。如果同时打开五个以上 TXD 指令输入，设备 D2 指定的发送状态数据寄存器中将出现错误代码，表示无法执行过多的 TXD 指令。
- 如果在执行 1 个 TXD 指令时打开另一个 TXD 指令输入，则在前一个 TXD 指令执行完毕后，其后的 TXD 指令要执行 2 次扫描。
- 由于 TXD 指令是在已打开输入时在每次扫描中执行的，所以，应当使用来自 SOTU 或 SOTD 指令的脉冲输入。
- TXD 指令及 RXD 指令中存储发送 / 接收运行状态和发送 / 接收数据字节数的数据寄存器无法重复。
- TXD 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见《梯形图编程手册》第 3 章中的“用户程序执行错误”。

WindLDR 中的用户通信发送指令对话框



发送指令对话框中的选项和设备

类型	TXD	发送指令
	RXD	接收指令
端口编号	端口 1 ~ 33	从端口 1 (TXD1) ~ 33 (TXD33) 发送用户通信
S1	源 1	在此区域中输入要发送的数据 发送数据可以是常量值 (字符或十六进制)、数据寄存器或 BCC。
D1	目标 1	发送完成输出可以是输出或内部继电器。
D2	目标 2	发送状态寄存器可以是数据寄存器。 下一个数据寄存器存储已发送数据的字节计数。

5: 用户通信指令

发送数据

源设备 S1 使用常量值或数据寄存器指定发送数据。还可以自动计算 BCC 代码并将其添加到发送数据。1 个 TXD 指令最多可以发送 1,500 字节的数据。

S1 (源 1)

发送数据	转换类型	发送位数 (字节)	重复	BCC 计算	当前地址
常量 (字符)	不转换	1	—	—	—
常量 (十六进制)					
数据 寄存器	A: 二进制数 → ASCII 码 B: BCD 码 → ASCII 码 -: 不转换	1 ~ 4 1 ~ 5 1 ~ 2	1 ~ 99	—	—
BCC	A: 二进制数 → ASCII 码 -: 不转换	1 ~ 2	—	X: XOR A: ADD C: Add-2comp M: Modbus ASCII M: Modbus RTU	1 ~ 15

注释: WindLDR 梯形图上 TXD 指令的 S1 将显示发送数据的字节总数。

指定常量作为 S1

将常量值指定为源设备 S1 时, 发送 1 个字节的数据无需转换。有效发送数据值取决于在“通信参数”对话框中选择的数据位数。可以通过“设置”>“通信端口”, 然后选择端口 1~33 列表框中的“用户协议”, 再单击“设置”按钮设置这些数据位。当默认选择 7 个数据位时, 将发送 00h ~ 7Fh。当选择 8 个数据位时, 将发送 00h ~ FFh。在源数据中输入字符或十六进制符号作为常量值。

常量 (字符)

计算机键盘上的所有可用字符都可以输入。1 个字符记做 1 个字节。

常量 (十六进制)

使用此选项可输入任一 ASCII 字符的十六进制代码。还可以使用此选项输入 ASCII 控制代码 NUL (00h) ~ US (1Fh)。

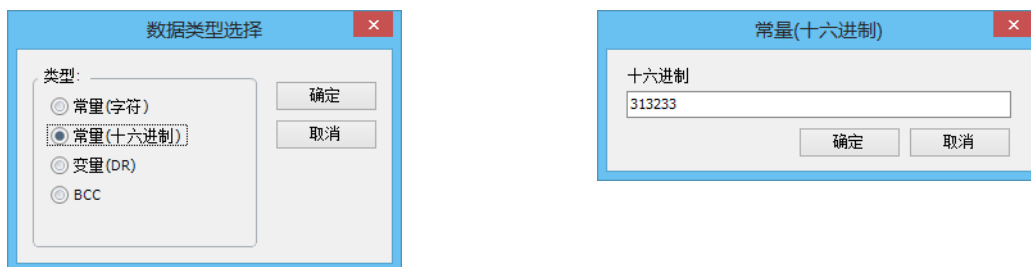
示例:

以下示例显示两种输入 3 字节 ASCII 数据“1” (31h)、 “2” (32h)、 “3” (33h) 的方法。

(1) 常量 (字符)



(2) 常量 (十六进制)



指定数据寄存器作为 S1

当数据寄存器被指定为源设备 S1 时，必须还要指定转换类型和发送位数。转换存储在指定数据寄存器中的数据，发送结果数据的指定数量的数字。可用转换类型包括：二进制数→ASCII 码、BCD 码→ASCII 码和不转换。

指定重复时，将发送以指定的数据寄存器开始并与重复次数一样多的数据寄存器数据。最多重复 99 次。

转换类型

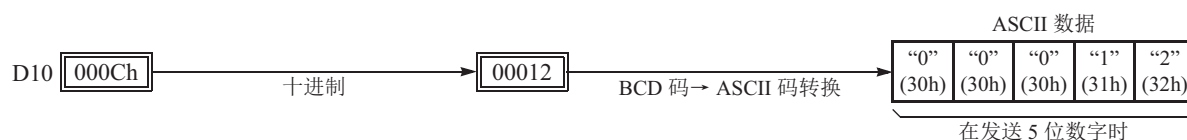
根据如下所述的指定转换类型转换发送数据：

示例：D10 存储 000Ch (12)

(1) 二进制数→ASCII 码转换



(2) BCD → BCD 码



(3) 不转换

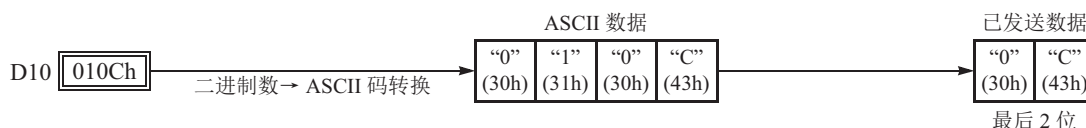


发送位数 (字节)

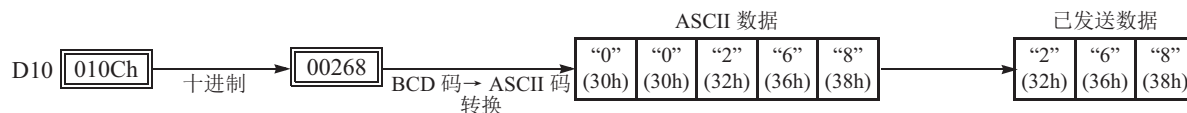
转换后，将提取出指定位数的发送数据。所需位数取决于所选转换类型。

示例：D10 存储 010Ch (268)

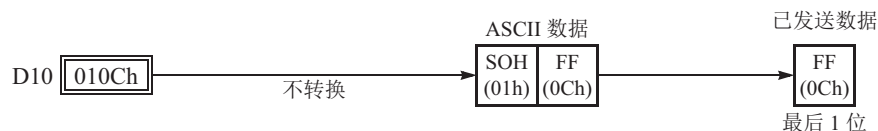
(1) 二进制数→ASCII 码转换，发送位数 = 2



(2) BCD 码→ASCII 码转换，发送位数 = 3



(3) 不转换，发送位数 = 1



5: 用户通信指令

重复次数

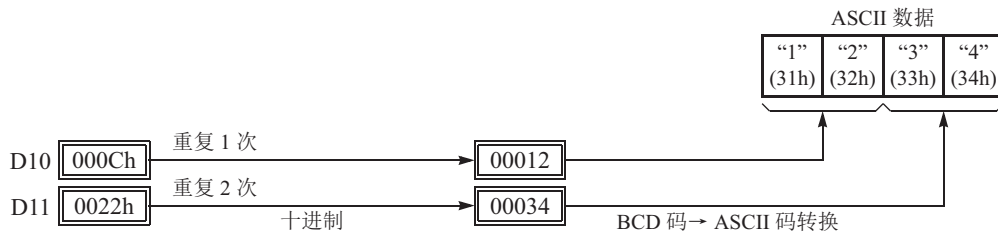
当数据寄存器指定为重复时，将使用与重复次数同样多的连续数据寄存器发送相同转换类型和发送位数的数据。

示例：

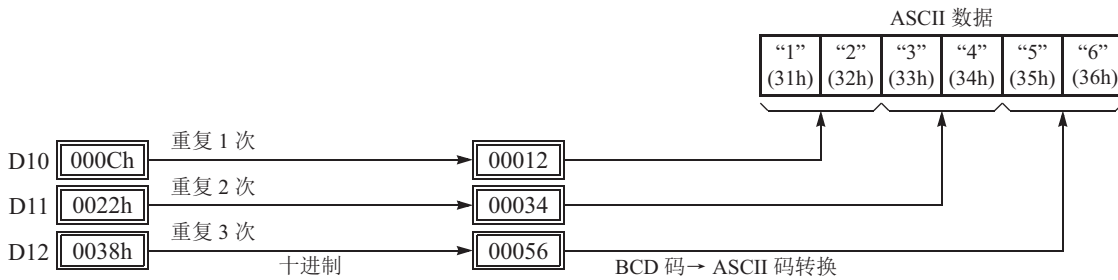
D10	000Ch	数据寄存器编号：D10
D11	0022h	发送位数： 2
D12	0038h	转换类型： BCD 码→ ASCII 码

以 D10 开始的数据寄存器数据将进行 BCD 码→ ASCII 码转换，并按照指定重复次数发送。

(1) 重复次数 = 2

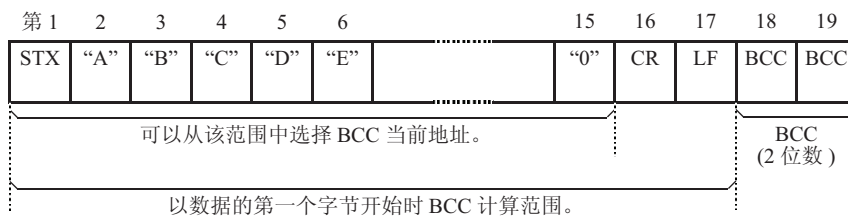


(2) 重复次数 = 3



BCC (块校验字符)

可以将块校验字符添加至发送数据。可以从第 1 个字节~第 15 个字节中选择 BCC 计算的起始位置。BCC 可以是 1 或 2 位数。

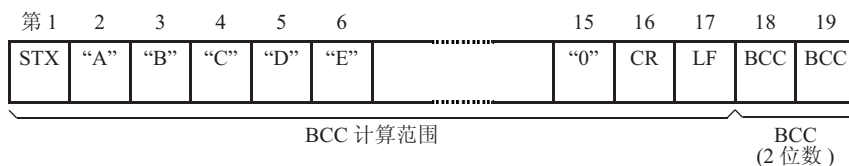


BCC 当前地址

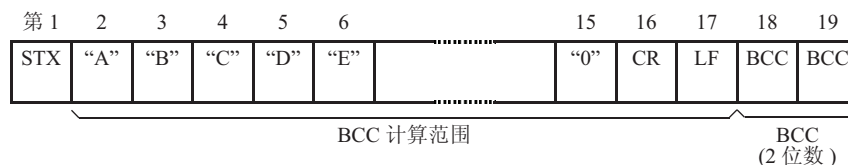
BCC 当前地址可从第 1 位~第 15 字节中选择，为从指定位置开始到 BCC 发送数据之前的位的范围计算 BCC。

示例：发送数据包包含 17 个字节外加 2 位 BCC 数字。

(1) 当前地址 = 1



(2) 当前地址 = 2

**BCC 计算公式**

可以从 XOR (异或)、ADD (加)、ADD-2comp、Modbus ASCII 或 Modbus RTU 运算中选择 BCC 计算公式。

示例：发送数据的转换结果包括 41h、42h、43h 和 44h。

ASCII 数据			
"A"	"B"	"C"	"D"
(41h)	(42h)	(43h)	(44h)

(1) BCC 计算公式 = XOR

计算结果 = $41h \oplus 42h \oplus 43h \oplus 44h = 04h$

(2) BCC 计算公式 = ADD

计算结果 = $41h + 42h + 43h + 44h = 10Ah \rightarrow 0Ah$ (只有最后 1 或 2 位数可用作 BCC。)

(3) BCC 计算公式 = ADD-2comp

计算结果 = FEh, F6h (2 位数没有转换)

(4) BCC 计算公式 = Modbus ASCII

计算结果 = 88 (ASCII)

(5) BCC 计算公式 = Modbus RTU

计算结果 = 85h 0Fh (二进制)

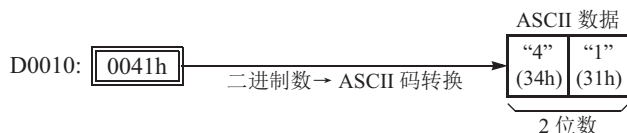
5: 用户通信指令

转换类型

BCC 计算结果根据如下所述的指定转换类型进行转换或不转换：

示例：BCC 计算结果为 0041h。

(1) 二进制数 → ASCII 码转换



注释：在 WindLDR 上，Modbus ASCII 默认为二进制数 → ASCII 码转换。

(2) 不转换

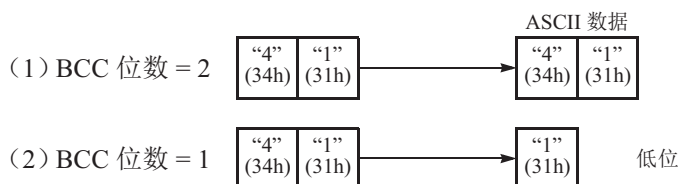


注释：在 WindLDR 上，Modbus RTU 默认为不转换。

BCC 位数（字节）

可以从 1 或 2 中选择 BCC 代码的数字位数（字节）。

示例：



注释：在 WindLDR 上，Modbus ASCII 和 Modbus RTU 默认为 2 位数字。

D1（目标 1）

将内部继电器或输出设置为发送完成输出。

TXD 指令的启动输入打开，且在结束发送预处理 → 发送处理的一系列处理时，发送完成输出打开。

D2（目标 2）

设置存储发送运行状态及发送字节数的数据寄存器。

发送运行状态存储到 D2+0，发送字节数存储到 D2+1 中。

D2+0（发送状态寄存器）

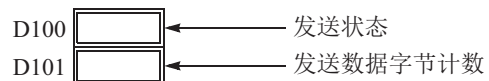
发送状态代码	状态	说明
16	准备发送	从打开 TXD 指令的起始输入至将发送数据存储在内部发送缓冲区内。
32	正在发送数据	从通过 END 处理启用数据发送至完成全部数据发送
48	数据发送完成	从完成全部数据发送至完成 TXD 指令的 END 处理
64	发送指令完成	全部发送操作已完成，可以执行下一个发送

如果出现与上述不同的发送状态代码，则可能出现发送指令错误。有关详情，请参见第 5-52 页上的“用户通信错误”。

D2+1（发送字节数）

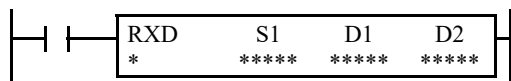
D2 中设置的数据寄存器的下一数据寄存器（D2+1）中将存储发送字节数。发送字节数中还包含 BCC 数据。
为发送状态指定的设备的数据寄存器用于存储 TXD 指令发送的数据的字节计数。当发送数据中有 BCC 时，发送数据字节计数中还包括 BCC 的字节计数。

示例：数据寄存器 D100 已指定为发送状态的设备。



5: 用户通信指令

RXD (接收)



当输入打开时，将根据 S1 指定的接收格式，转换通过端口 1 ~ 33 从 RS232C/RS485 远程终端接收的数据并将其存储至数据寄存器中。

有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	接收格式	—	—	—	—	—	—	X	—	X	—
D1 (目标 1)	接收完成输出	—	X	X ^{*1}	—	—	—	—	—	—	—
D2 (目标 2)	接收状态	—	—	—	—	—	—	X ^{*2}	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

*1 不能将特殊内部继电器指定为 D1。

*2 无法使用特殊数据寄存器。

由设备 S1 指定的接收格式最多为 1,536 字节。

数据接收完成后，将打开由设备 D1 指定的输出或内部继电器。

目标 2 占据两个 D2 指定的设备开始的连续数据寄存器。接收状态数据寄存器存储数据接收状态和错误代码。下一个数据寄存器存储已接收数据的字节计数。不能将同一数据寄存器用作 TXD1 ~ TXD33 指令的发送状态寄存器及 RXD1 ~ RXD33 指令的接收状态寄存器。

注释:

- 有关串行通信的用户通信的详情，请参见第 5-24 页上的“通过串行通信进行的用户通信”。
- 有关 Bluetooth 通信的用户通信的详情，请参见第 5-39 页上的“通过串行通信 (Bluetooth) 进行的用户通信”。

编写 RXD 指令的注意事项

- FC6A 型最多可执行各五个 RXD1 ~ RXD33 指令，这些指令同时具有 1 个起始分隔符。如果没有在 RXD1 ~ RXD33 指令中编入起始分隔符，FC6A 型一次只能执行 1 个 RXD1 ~ RXD33 指令。如果在执行 1 个没有起始分隔符的 RXD1 ~ RXD33 指令时打开另一个 RXD1 ~ RXD33 指令的起始输入，将出现用户通信错误。
- 因为在输入打开时，每次扫描都要执行 RXD 指令，所以应该使用 SOTU 或 SOTD 指令中的脉冲输入。
- 一旦打开 RXD 指令输入时，即使已关闭输入，仍会启动 RXD 并准备好接收输入通信。当 RXD 完成数据接收时，如果关闭输入至 RXD，则 RXD 将停用。或者，如果输入已打开，则 RXD 会准备好接收另一个通信。有特殊内部继电器可以停用所有正在等待输入通信的 RXD 指令。有关用户通信接收指令取消标志，请参见第 5-22 页上的“用户通信接收指令取消标志”。
- RXD 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见《梯形图编程手册》第 3 章中的“用户程序执行错误”。

WindLDR 中的用户通信接收指令对话框



接收指令对话框中的选项和设备

类型	TXD	发送指令
	RXD	接收指令
端口编号	端口 1 ~ 33	接收至端口 1 (RXD1) ~ 33 (RXD33) 的用户通信
S1	源 1	在此区域中输入接收格式。 接收格式可以包括起始分隔符、存储输入数据的数据寄存器、结束分隔符、BCC 和跳过。
D1	目标 1	接收完成输出可以是输出继电器或内部继电器。
D2	目标 2	接收状态寄存器可以是数据寄存器。 下一个数据寄存器存储已接收数据的字节计数。

接收格式

由源设备 S1 指定的接收格式将指定存储接收数据的数据寄存器，存储数据的数据位数，数据转换类型和重复次数。接收格式中包括起始分隔符和结束分隔符以区别有效输入通信。当需要已接收数据中的某些字符时，可以使用“跳过”来忽略指定数量的字符。还可以将 BCC 代码附加至接收格式以确认接收数据。1 个 RXD 指令最多可以接收 1,500 字节的数据。

S1(源 1)

接收格式	接收位数 (字节)	转换类型	重复	BCC 计算	计算起始 位置	跳过 字节	分隔符
数据寄存器	1 ~ 4 1 ~ 5 1 ~ 2	A: ASCII 码 → 二进制数 B: ASCII 码 → BCD 码 -: 不转换	1 ~ 99	—	—	—	Hex ASCII
起始分隔符	—	不转换	—	—	—	—	
结束分隔符	—	不转换	—	—	—	—	
常量	—	不转换	—	—	—	—	
BCC	1 ~ 2	A: 二进制数 → ASCII 码 -: 不转换	—	X: XOR A: ADD C: Add-2comp M: Modbus ASCII M: Modbus RTU	1 ~ 15	—	
跳过	—	—	—	—	—	1 ~ 99	

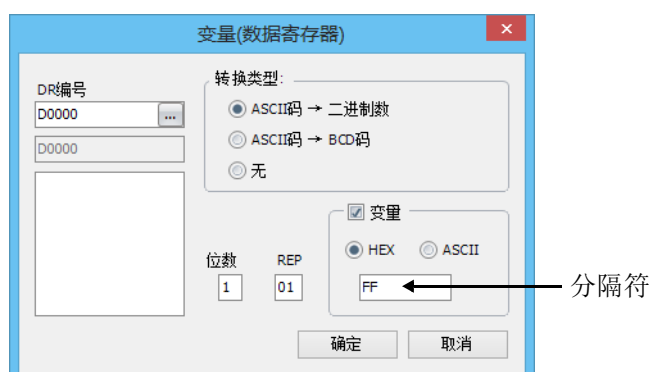
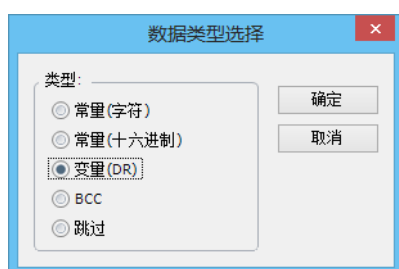
注释：WindLDR 梯形图上 RXD 指令的 S1 将显示接收格式的字节总数。

指定数据寄存器作为 S1

当指定数据寄存器为源设备 S1 时，还必须指定接收位数和转换类型。按指定接收位数将接收数据分为块，再按指定转换类型进行转换，然后存储至指定数据寄存器。可用转换类型包括 ASCII 码 → 二进制数、ASCII 码 → BCD 码和不转换。

指定重复时，已接收数据被分割、转换并存储至与重复次数相同数量的数据寄存器中（以指定数据寄存器开始）。最多重复 99 次。

当指定数据寄存器为源设备 S1 时，在数据寄存器指定的结束接收通信中可以包含分隔符。



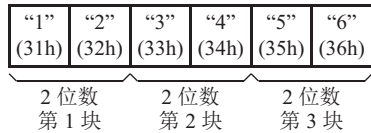
5: 用户通信指令

接收位数

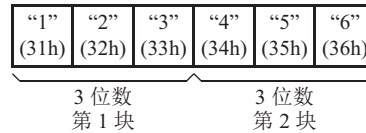
在按如下所述进行转换之前，要按指定接收位数分割已接收数据。

示例：6 字节已接收数据被分成不同的接收位数。（还要指定重复。）

(1) 接收位数 = 2



(2) 接收位数 = 3

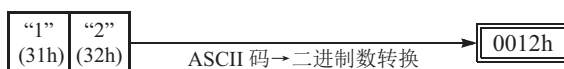


转换类型

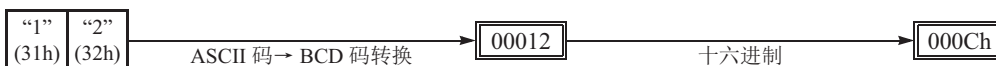
然后根据如下所述的指定转换类型转换指定接收位数的数据块：

示例：接收数据被分为 2 位数字块。

(1) ASCII 码→二进制数转换



(2) ASCII 码→BCD 码转换



(3) 不转换

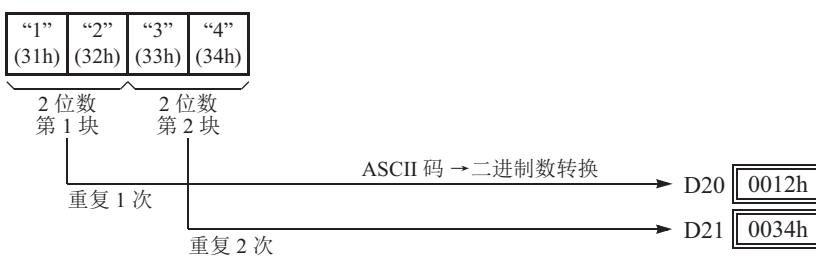


重复次数

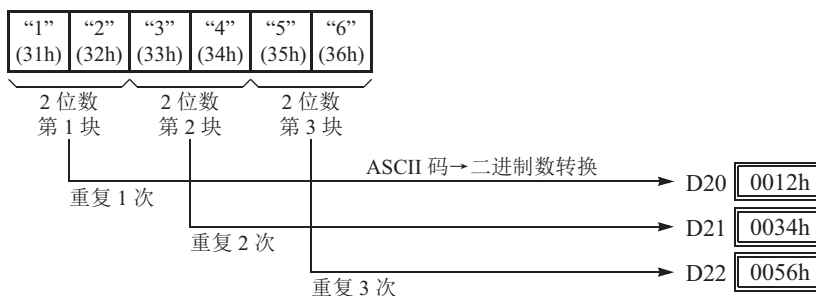
当数据寄存器指定为重复时，将按指定方式分割和转换接收数据，并且将转换的数据存储至与重复次数相同的连续数据寄存器中。

示例：6 字节接收数据被分为 2 位数字块，并进行 ASCII 码→二进制数，然后存储至以 D20 开始的数据寄存器中。

(1) 重复次数 = 2



(2) 重复次数 = 3



分隔符

在数据寄存器接收格式中的分隔符可以被指定。使用分隔符，可以接收输入数据的变量长度，并将其存储到数据寄存器。

分隔符	如何将输入数据被存储到数据寄存器
指定	输入数据将存储到数据寄存器直至处理完所有数据的接收位数、转换类型和重复或接收到指定的分隔符。
无分隔符	输入数据将存储到数据寄存器直至处理完所有数据的接收位数、转换类型和重复。

注释：数据寄存器的分隔符仅限使用于 RXD 指令的接收格式。

指定常量作为起始分隔符

可以在 RXD 指令的接收格式的第 1 个字节处编入起始分隔符；尽管也可以执行没有起始分隔符的 RXD 指令，但是 FC6A 型将识别出有效通信的开始部分。

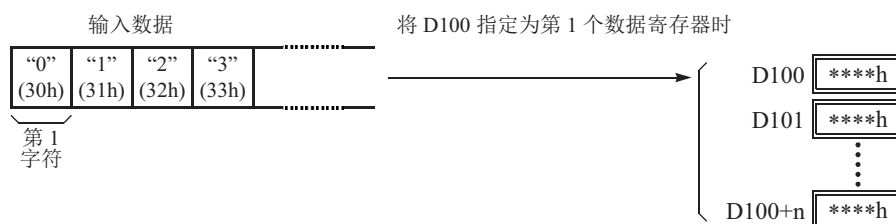
当在源设备 S1 的第 1 个字节处指定常量值时，将从作为起始分隔符的单字节数据处开始处理接收数据。

最多同时执行五个带不同起始分隔符的 RXD1 ~ RXD3 指令。当输入数据的第 1 个字节与 RXD 指令的起始分隔符匹配时，将根据 RXD 指令中指定的接收格式处理和存储接收数据。如果输入数据的第 1 个字节与已执行的任一 RXD 指令的起始分隔符都不匹配，FC6A 型将丢弃输入数据并等待下一个通信。

在执行没有起始分隔符的 RXD 指令时，将根据接收格式连续处理输入数据。一次只能执行 1 个没有起始分隔符的 RXD1 或 RXD3 指令。如果同时开始输入两个或多个没有起始分隔符的 RXD 指令，将执行地址最小的指令，并打开相应的完成输出。

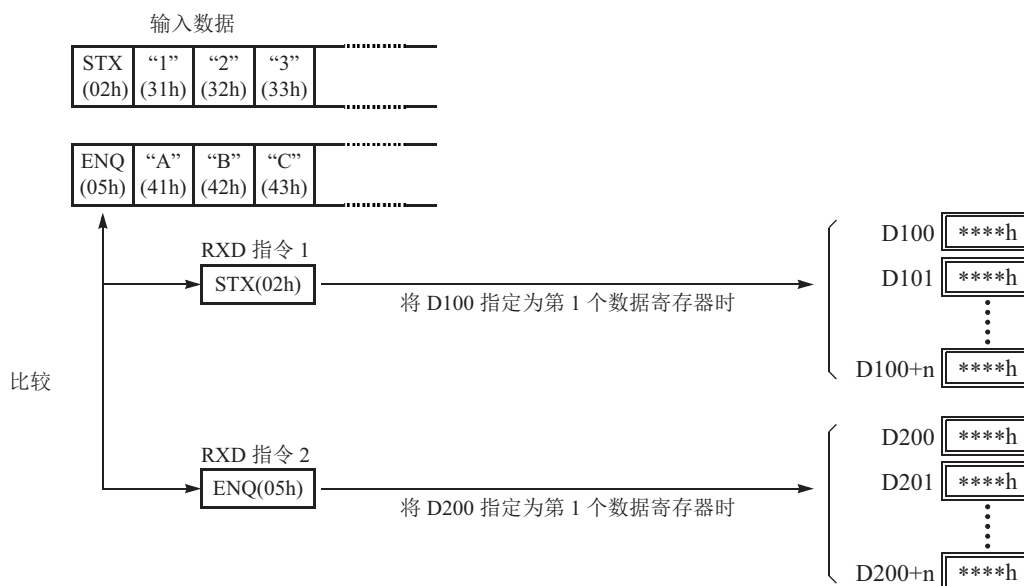
示例：

(1) 当执行没有起始分隔符的 RXD 指令时



根据接收格式分割、转换和存储输入数据至数据寄存器。

(2) 当执行带有起始分隔符 STX (02h) 和 ENQ (05h) 的 RXD 指令时



根据接收格式分割、转换和存储输入数据至数据寄存器。
起始分隔符未存储到数据寄存器。

5: 用户通信指令

多字节起始分隔符

可以在 RXD 指令的接收格式的第 1 个字节处编入起始分隔符；即使没有起始分隔符也可以执行 RXD 指令，FC6A 型将识别出有效通信的开始部分。从接收格式的第 1 字节中被认定为多字节起始分隔符可以是最多为 5 个连续的常量值（字符或十六进制）。

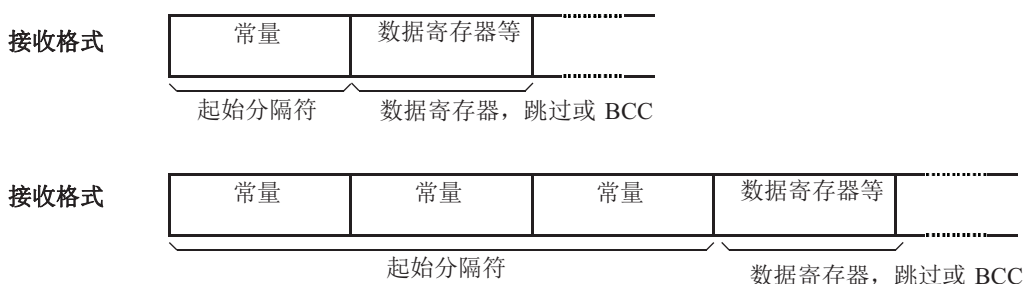
如果执行了 RXD 指令的起始分隔符并执行另一个 RXD 指令的同一个起始分隔符，用户通信错误代码 5 将被存储在第二个 RXD 指令的接收状态所指定的数据寄存器。当发生错误时，取消后一个 RXD 指令的执行，并保持前一个 RXD 指令的执行。

如果指定了多字节起始分隔符，并且输入数据与整个多字节起始分隔符不匹配时，接收的数据将被丢弃。接收了第 1 个字节后，将启动定时器并监控在指定了多字节起始分隔符的输入数据的间隔时间。如果接收第 1 个字节后，在指定的接收超时值的期间内没有接收到数据时，将发生接收超时，并且用户通信错误代码 11 将被存储在状态数据寄存器。

示例：多字节起始分隔符

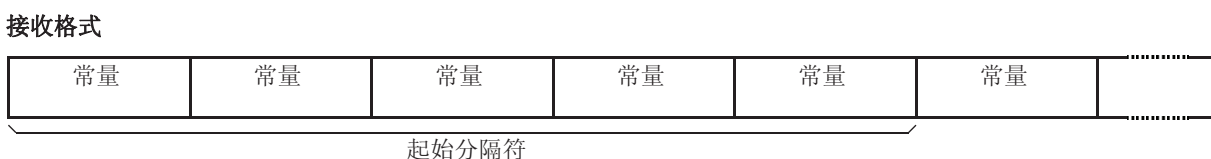
多字节起始分隔符是由接收格式的构造决定的。下例将展示如何确定多字节起始分隔符。

• 数据寄存器、跳过或 BCC 紧接着常量



注释：即使处于接收格式的起始的头 5 字节内，紧跟在数据寄存器，跳过或 BCC 后的常量不作为起始分隔符。

• 5 个以上的常量将从第一个字节指定



注释：既不是起始分隔符也不是结束分隔符的常量被指定为用于验证的常量。请参见第 5-17 页上的“用于确认的常量”。

示例：起始分隔符复写错误

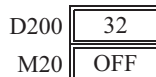
当打开输入 I0 时，执行第一个 RXD 指令并将状态代码 32 存储到接收状态 D200，显示的 RXD 指令等待接收数据。当打开输入 I1 时，执行另一个 RXD 指令，但是这两个 RXD 指令拥有同样的起始分隔符，第二个 RXD 指令将不被执行，并将用户通信错误代码 5 存储到接收状态 D300 中。



接收格式 S1



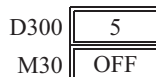
通信端口： 端口 1
接收完成输出： M20
接收状态寄存器： D200
接收数据字节计数器： D201



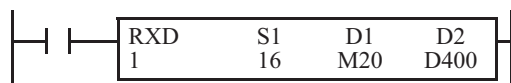
接收格式 S1



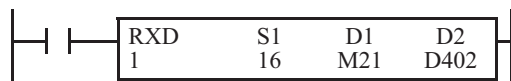
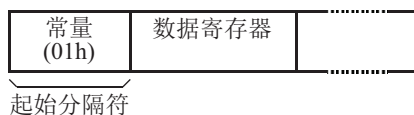
通信端口： 端口 1
接收完成输出： M30
接收状态寄存器： D300
接收数据字节计数器： D301



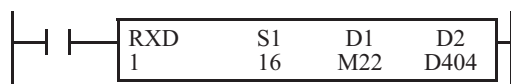
注释: 同时执行两个或多个带有多个字节起始分隔符的 RXD 指令时, 这些 RXD 指令的起始分隔符必须与其他指令的起始分隔符不同。如果 RXD 指令的多个字节起始分隔符的长度不同, 则应用其中最短的长度来检查重复的起始分隔符。以下任何 2 个 RXD 接收指令都被当作相同。



接收格式 S1



接收格式 S1



接收格式 S1

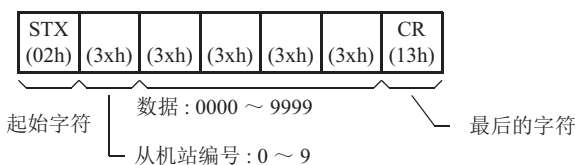


示例: 使用多字节起始分隔符

以下示例将说明使用多字节起始分隔符优于单字节起始分隔符。RXD 指令处理来自主机站的输入数据。输入数据传送到多个从机站 (0 ~ 9), 并且本地从机站编号为 1。因此, 当输入数据从从机站 1 传送时, 只能接收来自主机站的输入数据。

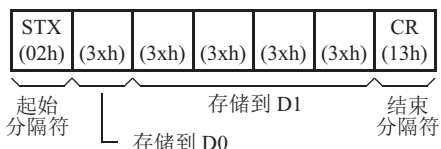
• 输入数据

输入数据由起始分隔符 STX 构成, 从机站编号可以是 0 ~ 9, 数据 0000 ~ 9999, 和结束分隔符 CR。



• 单字节起始分隔符

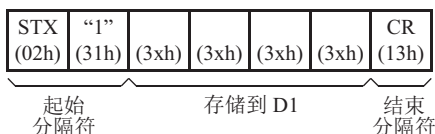
只有第一个字节才可以成为起始分隔符。输入数据的第二个字节为从机站编号, 必须存储到数据寄存器 D0, 并且附加的梯形图程序必须确认输入通信的从机站编号是否为 1。只有当从机站编号为 1 时, 接收的数据存储到对本地 PLC 有效的 D1 中。



5: 用户通信指令

• 多字节起始分隔符

前 2 个字节可以设置为多字节起始分隔符。当输入数据的前 2 个字节与起始分隔符相匹配，输入数据按照接收格式处理。因此，只处理传送到从机站 1 的输入数据。附加的梯形图程序不需要检查从机站编号。



指定常量作为结束分隔符

可以在 RXD 指令的接收格式的末尾写入结束分隔符；尽管也可以执行没有结束分隔符的 RXD 指令，但是 FC6A 型会识别出有效通信的结束部分。

当在源设备 S1 的末尾指定常量值时，将从作为结束分隔符的单或多字节数据处结束处理接收数据。结束分隔符可以是 00h ~ 7Fh。当选择 8 位数据时，结束分隔符可以是 00h ~ FFh。在源数据中输入字符或十六进制符号作为常量值。当在一个用户程序中重复使用同一个 RXD 指令时，请为每个 RXD 指令指定不同的结束分隔符。

如果输入数据中的字符与结束分隔符匹配，则 RXD 指令在该处结束接收数据，并按指定开始继续接收。即使与字符匹配的结束分隔符的位置比预期的早，RXD 指令也可以在那里结束接收数据。

如果 RXD 指令的接收格式中有 BCC 代码，则结束分隔符可以定位在紧接 BCC 代码之前或之后。如果在 BCC 和结束分隔符之间指定数据寄存器或跳过，则无法确保接收正确。

当执行没有结束分隔符的 RXD 指令时，数据接收将在收到接收格式的指定数据字节（如数据寄存器和跳过）时结束。另外，当输入数据字符之间的时间间隔超过在“通信设置”对话框中指定的接收超时时间值时，无论 RXD 是否有结束分隔符，数据接收都会结束。当接收到输入通信的第 1 个字符时启动字符间隔定时器，当接收到下一个字符时，定时器重新启动。若在预定时间内没有接收到字符，将出现超时，且 RXD 结束数据接收操作。

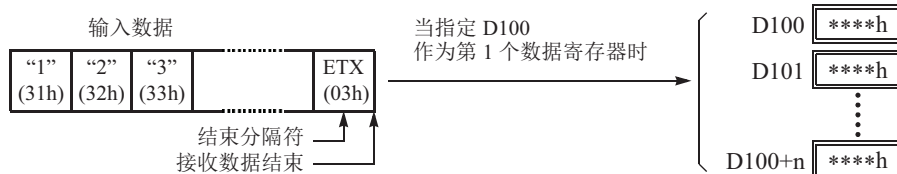
示例:

(1) 在执行没有结束分隔符的 RXD 指令时



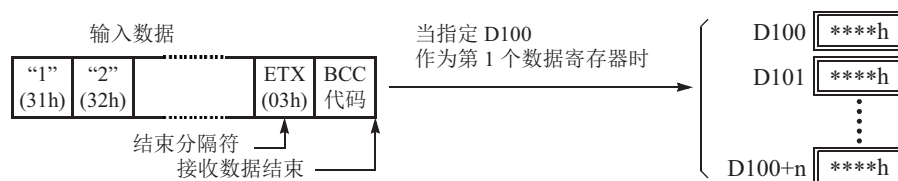
根据接收格式分割、转换和存储输入数据至数据寄存器。
当收到 RXD 中编写的总字符数时接收操作完成。

(2) 在执行没有结束分隔符 ETX (03h) 和没有 BCC 的 RXD 指令时



根据接收格式分割、转换和存储输入数据至数据寄存器。
结束分隔符未存储至数据寄存器。
将丢弃所有结束分隔符后的数据。

(3) 在执行有结束分隔符 ETX (03h) 和单字节 BCC 的 RXD 指令时



根据接收格式分割、转换和存储输入数据至数据寄存器。

结束分隔符和 BCC 代码未存储至数据寄存器。

在接收到结束分隔符后，FC6A 型 MICROSmart 仅接收单字节 BCC 代码。

用于确认的常量

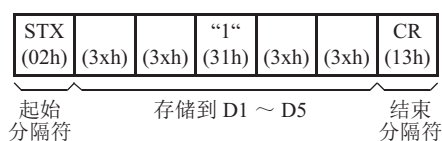
除起始和结束分隔符之外的常量可以配置为接收格式，从而通过常量（字符或十六进制值）来检查输入数据。可以根据需要，配置尽可能多的常量来用于检查。而常量的数量可任意设置，确认结果被存储在 RXD 指令的接收状态中。

示例：确认编辑的常量

以下示例将展示使用常量来确认的优点。输入数据的正中包含一个常量 "1"，并且常量值需要确认输入数据是否有效。

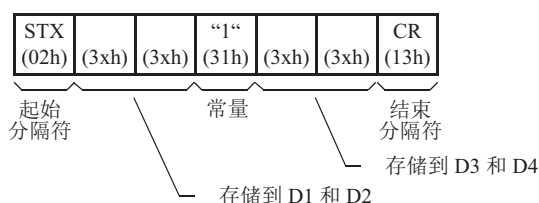
• 使用数据寄存器

输入数据包括常量值需要保存到数据寄存器。即使常量值不是预期值，当 RXD 指令完全接收输入数据，接收状态为 64，意味着完成了 RXD 指令并且没有发生错误。附加梯形图程序需要确认在输入数据中的常量值是否正确。



• 使用用于确认的常量

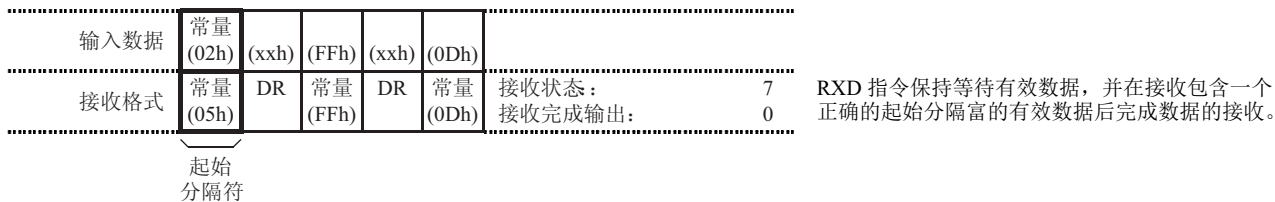
在接收格式中指定用于确认在输入数据的常量值的常量。当 RXD 指令完成接收输入数据时，如果常量值不是预期值，接收状态为 74，意味着 RXD 指令已完成但发生用户通信错误代码 5。附加的梯形图程序不需要确认接收数据中的常量值是否正确。



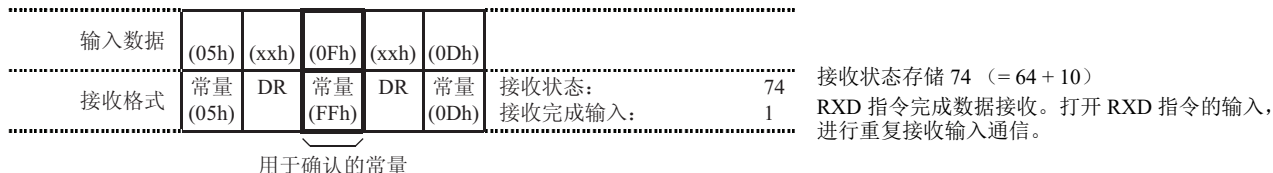
注释：在接收格式中设置字符或十六进制值的常量，并且输入数据与接收格式中的常量不匹配时，用户通信错误代码将被存储到接收状态。常量是否作为起始分隔符使用或是用于确认的常量使用将决定接受状态中的错误代码。如果作为起始分隔符使用，用户通信错误代码 7 将存储到接受状态中，并且 RXD 指令保持等待有效输入数据。如果作为确认的常量使用时，接受状态为 74，RXD 指令将完成执行。打开 RXD 指令的输入，进行重复接收输入通信。

5: 用户通信指令

- 输入数据的起始分隔符与接收格式不匹配

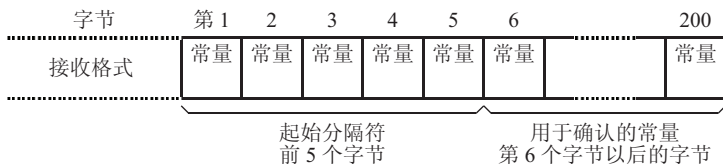


- 用于确认输入数据的常量与接收格式不匹配

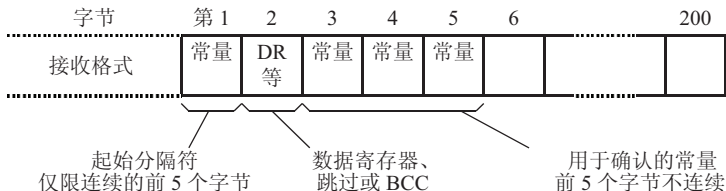


注释: 设置在接收格式起始位置上的常量有不同功能，如下所示：

- 5 个以上的常量被设置在接收格式的起始位置



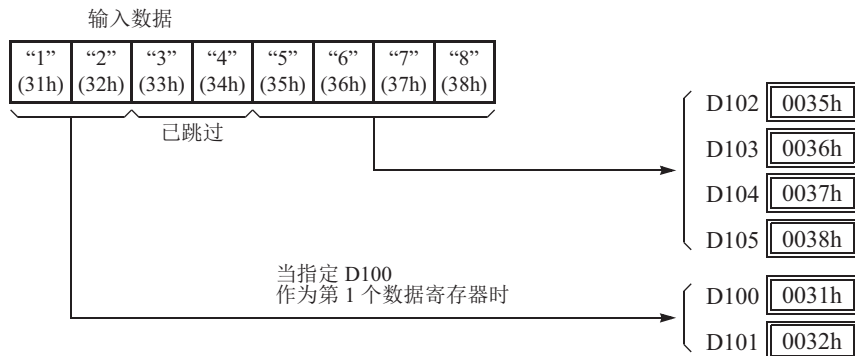
- 常量以外（数据寄存器、跳过或 BCC）被设置在接收格式的前 5 个字节



跳过

当在接收格式中指定“跳过”时，将跳过输入数据中指定数量的数字，这些数字不会存储至数据寄存器中。最多可以连续跳过 99 位（字节）字符。

示例: 当执行带有跳过第三字节前两位数字的 RXD 指令时



BCC（块校验字符）

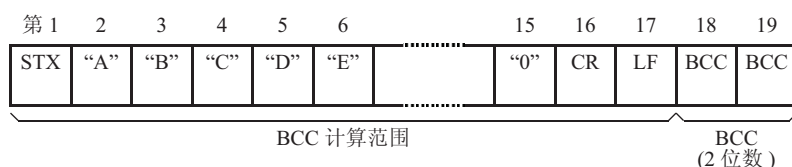
FC6A 型有自动 BCC 计算功能，可检测输入数据中的通信错误。如果在 RXD 指令的接收格式中指定 BCC 代码，FC6A 型将通过 BCC 前的起始位置计算指定起始位置的 BCC 值。然后，将计算结果与已接收输入数据中的 BCC 代码进行比较。可以从第 1 个字节～第 15 个字节中指定 BCC 当前地址。BCC 可以是 1 或 2 位数。当 RXD 指令中未使用结束分隔符时，BCC 代码必须定位在源 1 设备指定的接收格式末尾。当使用结束分隔符时，BCC 代码必须紧接在结束分隔符之前或之后。FC6A 型根据接收格式读取输入数据中指定数量的 BCC 位数，以计算和比较已接收 BCC 代码和 BCC 计算结果。

BCC 当前地址

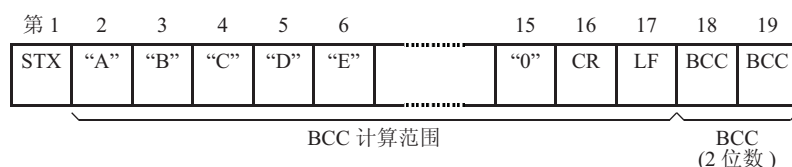
BCC 当前地址可从第 1 位～第 15 位中选择，为从指定位置开始到 BCC 发送数据之前的位的范围计算 BCC。

示例：已接收数据包含 17 个字节外加 2 位 BCC 数字。

(1) 当前地址 = 1



(2) 当前地址 = 2



BCC 计算公式

可以从 XOR（异或）、ADD（加）、ADD-2comp、Modbus ASCII 或 Modbus RTU 操作中选择 BCC 计算公式。

示例：输入数据包括 41h、42h、43h 和 44h。

(1) BCC 计算公式 = XOR

$$\text{计算结果} = 41\text{h} \oplus 42\text{h} \oplus 43\text{h} \oplus 44\text{h} = 04\text{h}$$

(2) BCC 计算公式 = ADD

$$\text{计算结果} = 41\text{h} + 42\text{h} + 43\text{h} + 44\text{h} = 10\text{Ah} \rightarrow 0\text{Ah} \quad (\text{只有最后 1 或 2 位数可用作 BCC。})$$

(3) BCC 计算公式 = ADD-2comp

$$\text{计算结果} = \text{FEh}, \text{F6h} \quad (2 \text{ 位数没有转换})$$

(4) BCC 计算公式 = Modbus ASCII

$$\text{计算结果} = 88 \quad (\text{ASCII})$$

(5) BCC 计算公式 = Modbus RTU

$$\text{计算结果} = 85\text{h} 0\text{Fh} \quad (\text{二进制})$$

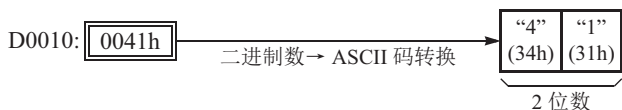
5: 用户通信指令

转换类型

可以转换 BCC 计算结果或不根据如下所述的指定转换类型进行转换:

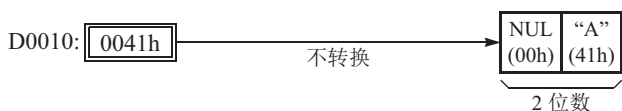
示例: BCC 计算结果为 0041h。

(1) 二进制数 → ASCII 码转换



注释: 在 WindLDR 上, Modbus ASCII 默认为二进制数 → ASCII 码转换。

(2) 不转换

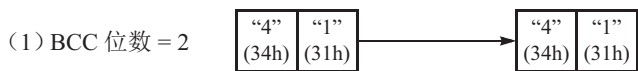


注释: 在 WindLDR 上, Modbus RTU 默认为不转换。

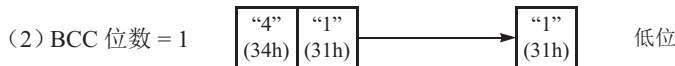
BCC 位数 (字节)

可以从 1 或 2 中选择 BCC 代码的数字位数 (字节)。

示例:



注释: WindLDR 上, Modbus ASCII 和 Modbus RTU 默认设置为 2 位数字。

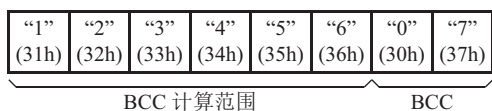


比较 BCC 代码

FC6A 型将 BCC 计算结果与已接收数据中的 BCC 代码进行比较, 以检查输入通信中是否有由于外部噪音或其他原因导致的错误。如果在比较中发现不一致, 则错误代码将存储至指定为 RXD 指令接收状态的数据寄存器中。有关用户通信错误代码, 请参见第 5-52 页上的“用户通信错误”。

示例 1: 使用 XOR 格式计算第 1 字节~第 6 字节的 BCC, 然后进行二进制数 → ASCII 码转换, 再与添加至输入数据第七和第八字节的 BCC 代码进行比较。

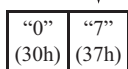
输入数据



BCC 计算结果

$$31h \oplus 32h \oplus 33h \oplus 34h \oplus 35h \oplus 36h = 07h$$

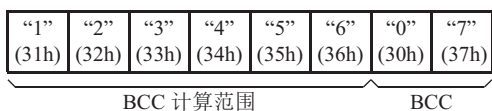
二进制数 → ASCII 码转换



比较结果一致表示数据接收正确

示例 2: 使用 ADD 格式计算第 1 字节~第 6 字节的 BCC, 然后进行二进制数 → ASCII 码转换, 再与添加至输入数据第 7 和第 8 字节的 BCC 代码进行比较。

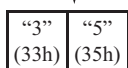
输入数据



BCC 计算结果

$$31h + 32h + 33h + 34h + 35h + 36h = 135h$$

二进制数 → ASCII 码转换



比较结果不一致。

错误代码 9 将存储至接收状态数据寄存器中。

D1 (目标 1)

将内部继电器或输出设置为接收完成输出。

RXD 指令的启动输入打开，且在结束接收预处理→数据接收→接收数据展开的一系列处理时，接收完成输出打开。

接收数据完成条件

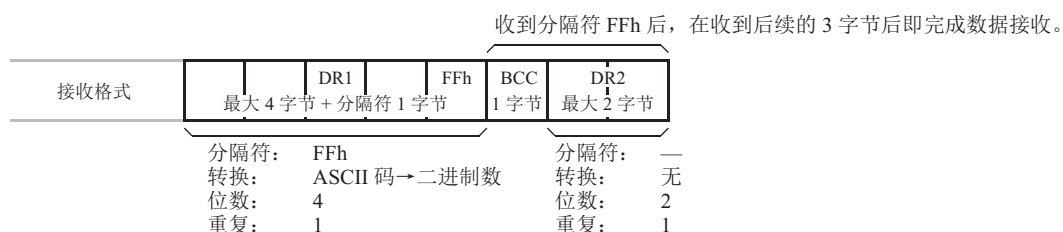
在开始接收数据后，可以以三种方式完成 RXD 指令：

结束分隔符	分隔符	完成接收数据的条件
有	有或无	当收到指定字节数的数据（位数 x 重复）或收到结束分隔符时。如果在结束分隔符之后紧接着 BCC，将在结束数据接收之前收到 BCC。
无	有	收到 RXD 指令中指定的最后一个常量（包括分隔符）后，在收到后续字节数的数据时即完成数据接收。
无	无	当收到指定字节数的数据（位数 x 重复）时。

注释：发生接收超时，数据接收将随意停止。

当满足上述三个条件之一时，数据接收完成。要终止 RXD 指令，请使用用户通信接收指令取消标志的特殊内部继电器。请参见第 5-22 页上的“用户通信接收指令取消标志”。

示例：RXD 指令不含结束分隔符，并为数据寄存器编程了接收格式的分隔符。

**D2 (目标 2)**

设置存储接收运行状态及接收字节数的数据寄存器。

接收运行状态存储到 D2+0，接收字节数存储到 D2+1 中。

D2+0 (接收状态)

D2 中设置的数据寄存器中将存储接收运行状态。接收运行状态包括接收运行状态和错误信息。

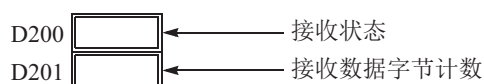
接收状态代码	状态	说明
16	准备数据接收	从开始输入 RXD 指令以读取接收格式，至通过 END 处理启用 RXD 指令
32	正在接收数据	从通过 END 处理启用 RXD 指令，至接收输入数据
48	数据接收完成	从接收输入数据，至根据接收格式转换接收数据并将其存储至数据寄存器中
64	接收指令完成	全部数据接收操作已完成，可以执行下一个数据接收
128	用户通信接收指令取消标志已启动	使用用户通信接收指令取消标志的特殊内部继电器取消 RXD 指令，例如 M8022、M8023、M8026、M8033、M8145 至 M8147、M8170、M8176、或 M8365 至 M8394

如果出现与上述不同的接收状态代码，则可能出现接收指令错误。请参见第 5-52 页上的“用户通信错误”。

D2+1 (接收字节数)

为接收状态指定的设备旁边的数据寄存器用于存储 RXD 指令接收的数据的字节计数。当接收数据中有起始分隔符、结束分隔符和 BCC 时，这些代码的字节计数也会包括在接收数据字节计数中。

示例：数据寄存器 D200 已指定为接收状态的设备。



5: 用户通信指令

用户通信接收指令取消标志

用户通信接收指令的接收预处理已完成，如果在接收中（状态代码 32）的状态下打开用户通信接收指令取消，则会取消执行相应端口对应的所有接收指令。接收数据等待的状态较长，希望取消执行接收指令时启用。如需激活已取消的接收指令，请在关闭用户通信接收指令取消标记后，重新打开接收指令的输入条件。用户通信接收指令取消标记在各通信端口中作为特殊内部继电器进行如下分配。

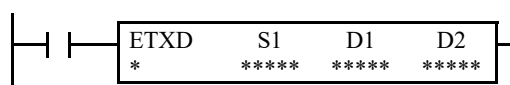
设备地址	说明	CPU 停止	电源关闭	R/W
M8022	使用通信接收指令取消标志（端口 1）	清除	清除	R/W
M8023	使用通信接收指令取消标志（端口 2）	清除	清除	R/W
M8026	使用通信接收指令取消标志（端口 3）	清除	清除	R/W
M8033	使用通信接收指令取消标志（端口 4）*1	清除	清除	R/W
M8145	使用通信接收指令取消标志（端口 5）*1	清除	清除	R/W
M8146	使用通信接收指令取消标志（端口 6）*1	清除	清除	R/W
M8147	使用通信接收指令取消标志（端口 7）*1	清除	清除	R/W
M8170	使用通信接收指令取消标志（端口 8）*1	清除	清除	R/W
M8176	使用通信接收指令取消标志（端口 9）*1	清除	清除	R/W
M8365	使用通信接收指令取消标志（端口 10）*1*2	清除	清除	R/W
M8366	使用通信接收指令取消标志（端口 11）*1*2	清除	清除	R/W
M8367	使用通信接收指令取消标志（端口 12）*1*2	清除	清除	R/W
M8370	使用通信接收指令取消标志（端口 13）*1*2	清除	清除	R/W
M8371	使用通信接收指令取消标志（端口 14）*1*2	清除	清除	R/W
M8372	使用通信接收指令取消标志（端口 15）*1*2	清除	清除	R/W
M8373	使用通信接收指令取消标志（端口 16）*1*2	清除	清除	R/W
M8374	使用通信接收指令取消标志（端口 17）*1*2	清除	清除	R/W
M8375	使用通信接收指令取消标志（端口 18）*1*2	清除	清除	R/W
M8376	使用通信接收指令取消标志（端口 19）*1*2	清除	清除	R/W
M8377	使用通信接收指令取消标志（端口 20）*1*2	清除	清除	R/W
M8380	使用通信接收指令取消标志（端口 21）*1*2	清除	清除	R/W
M8381	使用通信接收指令取消标志（端口 22）*1*2	清除	清除	R/W
M8382	使用通信接收指令取消标志（端口 23）*1*2	清除	清除	R/W
M8383	使用通信接收指令取消标志（端口 24）*1*2	清除	清除	R/W
M8384	使用通信接收指令取消标志（端口 25）*1*2	清除	清除	R/W
M8385	使用通信接收指令取消标志（端口 26）*1*2	清除	清除	R/W
M8386	使用通信接收指令取消标志（端口 27）*1*2	清除	清除	R/W
M8387	使用通信接收指令取消标志（端口 28）*1*2	清除	清除	R/W
M8390	使用通信接收指令取消标志（端口 29）*1*2	清除	清除	R/W
M8391	使用通信接收指令取消标志（端口 30）*1*2	清除	清除	R/W
M8392	使用通信接收指令取消标志（端口 31）*1*2	清除	清除	R/W
M8393	使用通信接收指令取消标志（端口 32）*1*2	清除	清除	R/W
M8394	使用通信接收指令取消标志（端口 33）*1*2	清除	清除	R/W

*1 使用通信模块时

*2 仅 Plus CPU 模块

注释：“R/W”是读取/写入的缩写。R/W表示可以读取和写入。R表示只能读取。W表示只能写入。

ETXD（以太网用户通信发送）



ETXD 指令将发送数据转换为指定的数据类型，并将其发送到通过以太网端口 1 及 2 连接的外部设备上。

输入打开后，S1 指定的发送数据将发送到与指定连接相连的设备上。

发送完成后，D1 指定的设备将打开。发送状态（发送状态和错误代码）会存储到 D2 指定的设备上。

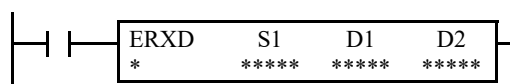
发送数据的字节数会存储到 D2+1。

除了连接设置以外，ETXD 与 TXD 指令的其他设置都相同。有关 TXD 指令的详情，请参见第 5-2 页上的“TXD（发送）”。

注释：

- 有关用户通信客户端和用户通信服务器规格的详情以及以太网用户通信的详情，请参见第 5-40 页上的“通过以太网通信进行的用户通信”。
- ETXD 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见《梯形图编程手册》第 3 章中的“用户程序执行错误”。

ERXD（以太网用户通信接收）



ERXD 指令接收通过以太网端口 1 及 2 连接的外部设备发出的数据，并将接收的数据转换为指定的格式，然后将转换的数据存储到数据寄存器中。

输入打开后，系统将从与指定连接相连的设备接收 S1 指定的接收数据。

所有数据接收完毕后，D1 指定的设备将打开。接收状态（接收状态和错误代码）会存储到 D2 指定的设备上。

接收数据的字节数会存储到 D2+1。

在接收传入数据的过程中，如果打开了用户通信接收指令取消标记（M8200 ~ M8207 和 M8334 ~ M8343），则系统将取消执行对应连接的所有有效接收指令。

除了连接设置和用户通信接收指令取消标志的分配以外，ERXD 和 RXD 指令的其他设置都相同。有关 RXD 指令的详情，请参见第 5-10 页上的“RXD（接收）”。

注释：

- 有关用户通信客户端和用户通信服务器规格的详情以及用户通信（UDP）的详情，请参见第 5-40 页上的“通过以太网通信进行的用户通信”。
- ERXD 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见《梯形图编程手册》第 3 章中的“用户程序执行错误”。

5: 用户通信指令

通过串行通信进行的用户通信

以串行通信的方式进行用户通信时，只要使用 TXD（发送）指令及 RXD（接收）指令，即可与端口 1 至 33 所连接的打印机、条形码读取器等外部设备进行数据的收发。

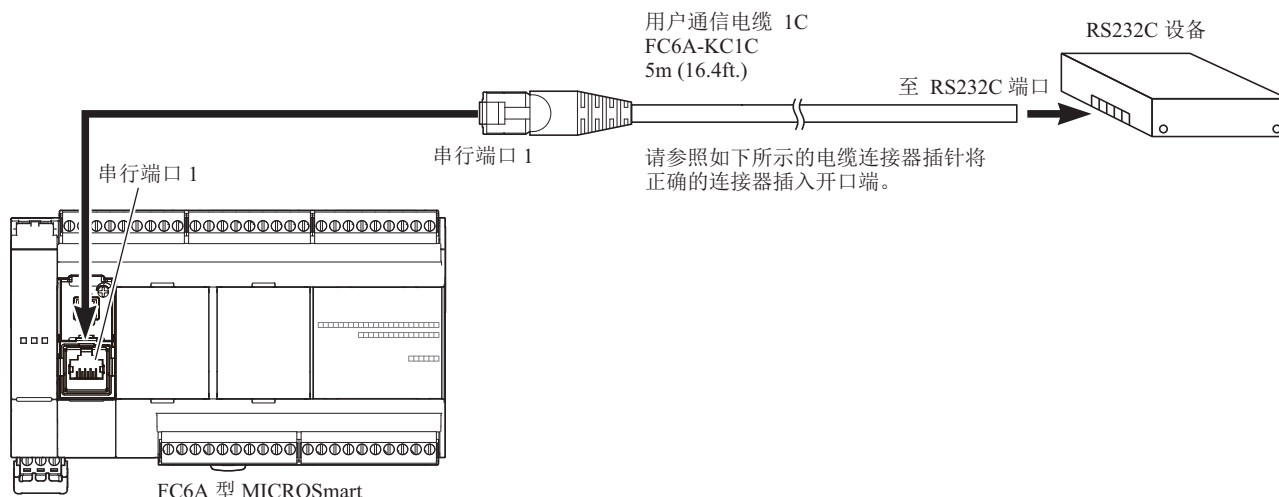
用户通信概述

通过在 FC6A 型扩展通信端口上安装通信盒，FC6A 型可同时与两台外部设备通信。

使用 RS485 通信盒时，FC6A 型最多可使用用户通信与 31 台 RS485 设备通信。

可以编辑用户通信发送和接收指令与通信设备的通信协议一致。可以参考下述用户通信模式说明以确认使用用户通信模式进行通信的可行性。

将 RS232C 设备连接到 FC6A 型的端口 1。



用户通信模式规格

类型	RS232C 用户通信	RS485 用户通信
通信端口	端口 1 至 33	端口 1 至 33
连接设备数	每个端口 1 个	最多 31 个
标准	EIA RS232C	EIA RS485
通信速度	1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200bps (默认: 115,200)	
数据长度	7 或 8 位 (默认: 7)	
奇偶校验	无、奇数、偶数 (默认: 偶数)	
停止位	1 或 2 位 (默认: 1)	
接收超时时间	10 ~ 2,540ms (增量为 10ms) 或无 (当选择 2,550ms 时禁用接收超时。) 在使用 RXD 指令时接收超时会有一些影响。	
通信方式	起止同步系统半双工	
最大电缆长度	FC6A-PC1: 5m FC6A-SIF52: 15m	FC6A-PC3: 200m FC6A-SIF52: 1,200m
最大发送数据量	1,500 字节	
最大接收数据量	1,500 字节	
BCC 计算	XOR、ADD、ADD-2comp*、Modbus ASCII*、Modbus RTU* (* 有关计算示例, 请参见第 5-58 页上的“BCC 计算示例”。)	

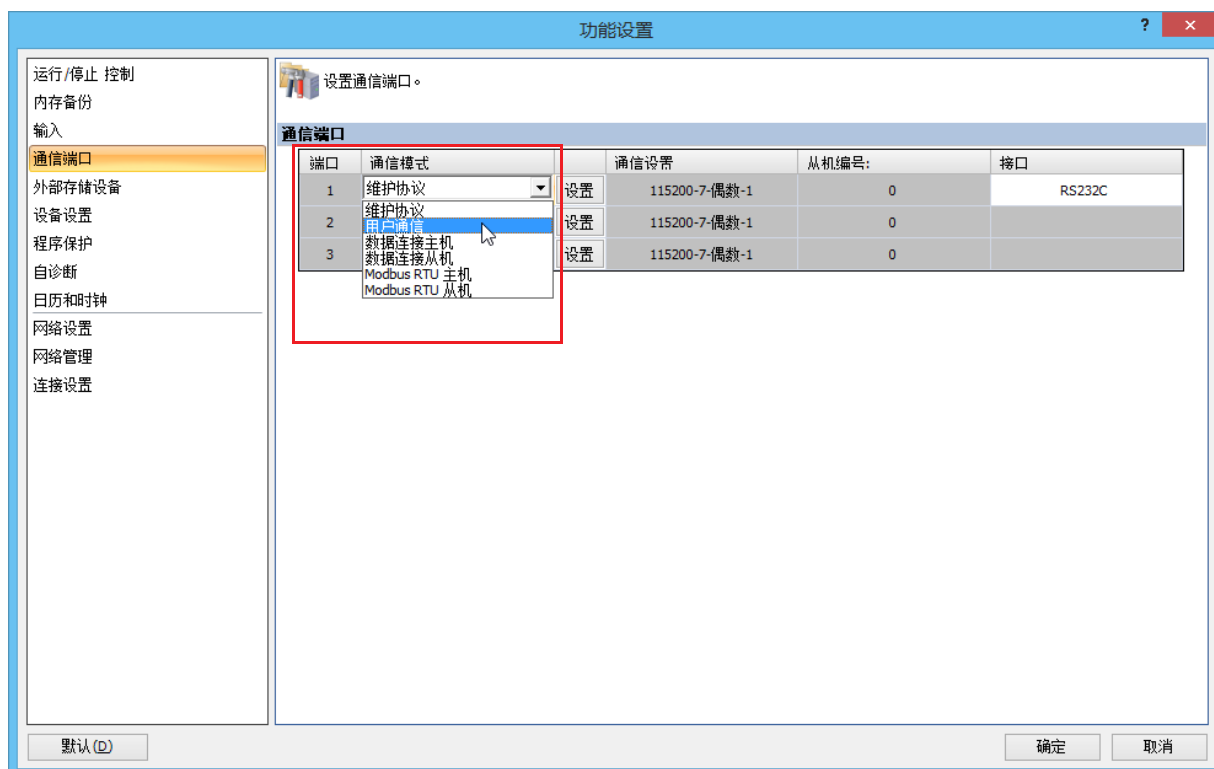
设置 WindLDR

当使用用户通信功能与外部 RS232C 或 RS485 设备进行通信时，请设置 FC6A 型的通信参数与外部设备的一致。

注释：因为功能设置中的通信设置与用户程序相关，所以在更改任何设置后，必须下载用户程序至 FC6A 型。

1. 在 WindLDR 菜单栏中选择“设置”>“通信端口”。

此时出现通信端口的“功能设置”对话框。



2. 在“端口1”~“端口3”的“通信模式”下拉列表中，选择“用户通信”。（在更改以前的设置时，请单击“设置”按钮。）

出现“用户通信（端口1）”对话框。



在“接收超时时间”框中选择“2550ms”时，将禁用接收超时功能。

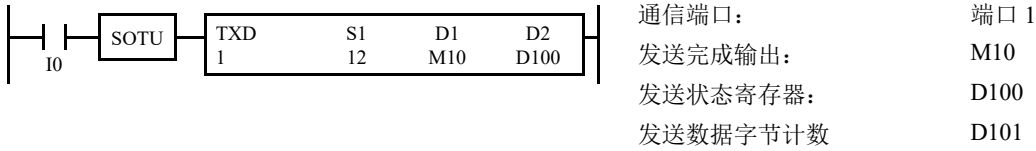
3. 选择与通信设备相同的通信设置。
4. 单击“确定”按钮。

5: 用户通信指令

使用 WindLDR 编程 TXD 指令

以下示例说明如何使用 WindLDR 编写 TXD 指令，包括起始分隔符、BCC 和结束分隔符。

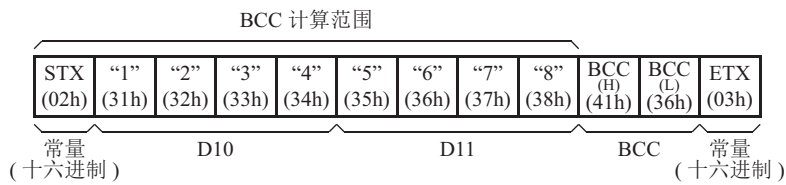
TXD 示例程序:



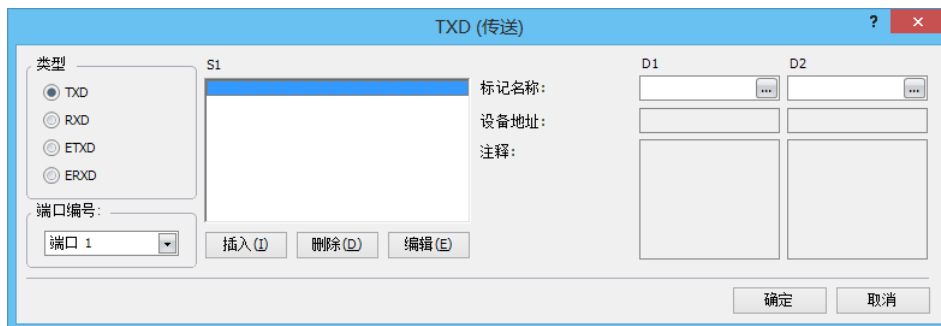
数据寄存器内容:

D10 04D2h = 1234
 D11 162Eh = 5678

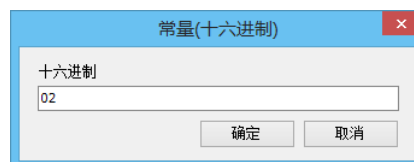
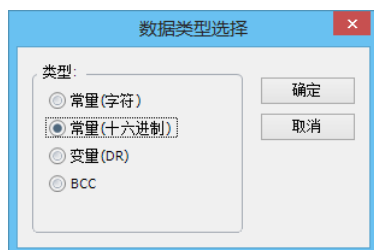
发送数据示例:



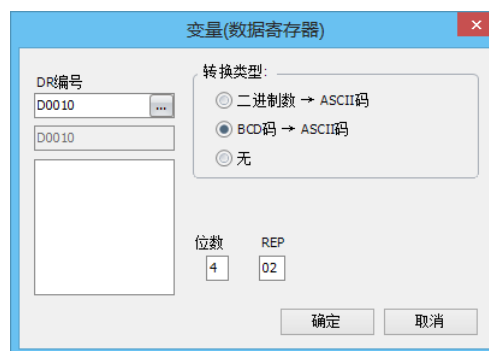
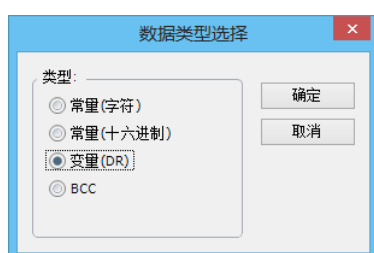
1. 开始编写 TXD 指令。移动光标至要插入 TXD 指令和类型“TXD”的位置。还可以单击菜单栏中的“用户通信”图标，并在编程编辑区单击插入 TXD 指令的位置插入 TXD 指令。出现发送指令对话框。



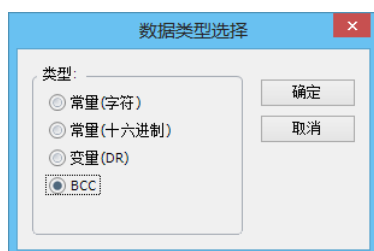
2. 确认已在类型框中选中“TXD”，并在端口框中选择“端口 1”。然后单击“插入”。将出现“数据类型选择”对话框。可以使用此对话框编写源设备 S1。
3. 选中类型框中的“常量（十六进制）”后单击“确定”。然后，在“常量（十六进制）”对话框中，输入“02”编写起始分隔符 STX（02h）程序。完成后，单击“确定”。



4. 当“发送指令”对话框再次出现时，重复以上步骤。在“数据类型选择”对话框中，单击“变量（DR）”后单击“确定”。然后，在“变量（数据寄存器）”对话框中，在 DR 编号框中输入“D0010”，然后单击“BCD 码→ ASCII 码”以选择 BCD 码→ ASCII 码转换。输入“4”至位数框（4 位数）中，然后输入“02”至 REP 框（重复 2 次）中。完成后，单击“确定”。

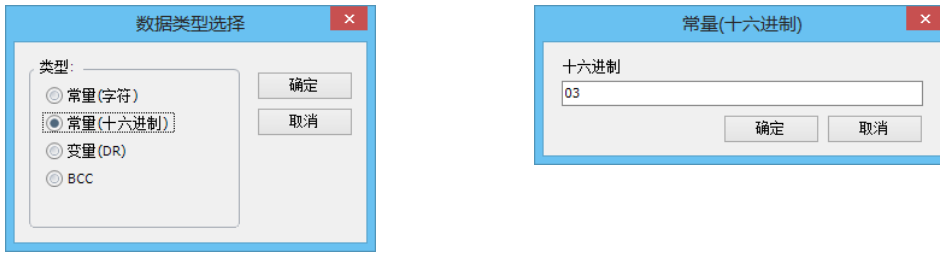


5. 再次在“数据类型选择”对话框中，单击“BCC”后单击“确定”。然后，在“BCC”对话框中，在当前地址框中输入“1”，选择“ADD”作为计算方法，然后单击“二进制数→ ASCII 码”作为转换类型，再单击“2”作为位数。完成后，单击“确定”。

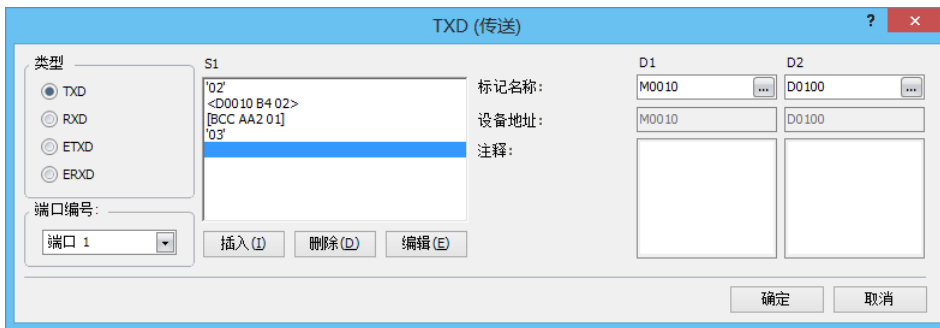


5: 用户通信指令

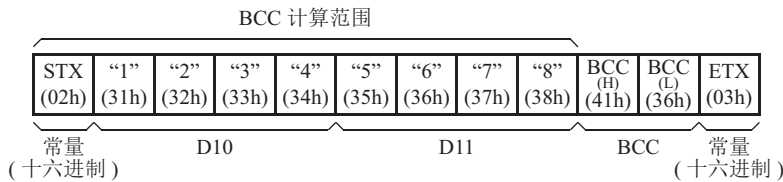
6. 再次在“数据类型选择”对话框中，单击“常量（十六进制）”后单击“确定”。然后，在“常量（十六进制）”对话框中，输入“03”以编写结束分隔符 ETX（03h）程序。完成后，单击“确定”。



7. 在“TXD（传送）”对话框中，在目标 D1 框中输入“M0010”，然后在目标 D2 框中输入“D0100”。完成后，单击“确定”。



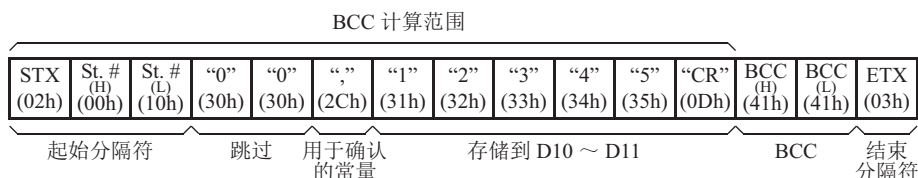
编写 TXD1 指令完成后，按如下所示指定发送数据：



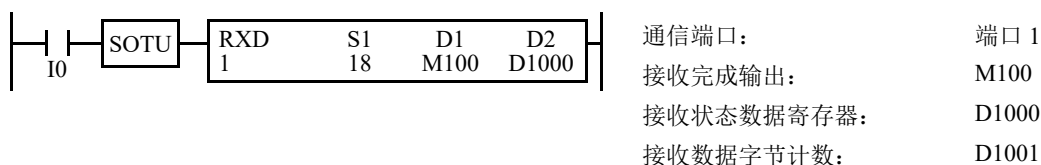
使用 WindLDR 编写 RXD 指令

以下示例说明如何使用 WindLDR 编写 RXD 指令，包括起始分隔符、跳过、BCC 和结束分隔符。转换的数据被存储至数据寄存器 D10 和 D11。内部继电器 M100 用作目标 D1 以接收完成输出。数据寄存器 D1000 用作接收状态的目标 D2，数据寄存器 D1001 用于存储接收数据字节计数。

接收数据示例



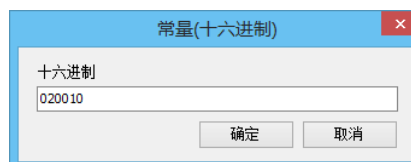
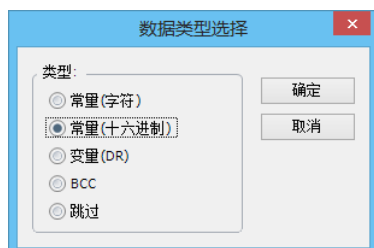
RXD 示例程序：



1. 开始编写 RXD 指令。移动光标至要插入 RXD 指令和类型“RXD”的位置。还可以单击菜单栏中的“用户通信”图标，并在程序编辑区单击插入 RXD 指令的位置，然后当“发送”对话框出现时，插入 RXD 指令。单击“RXD”可更改该对话框为“接收”对话框。出现接收指令对话框。

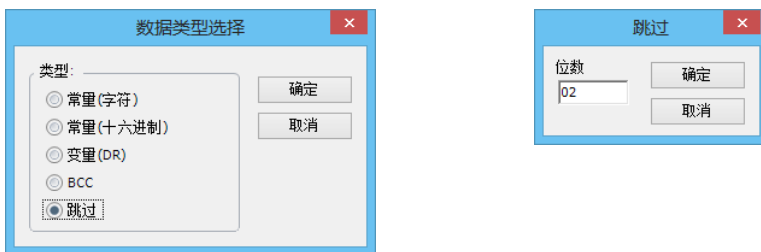


2. 确认已在类型框中选中“RXD”，然后选择端口框中的“端口 1”。然后单击“插入”。将出现“数据类型选择”对话框。可以使用此对话框编写源设备 S1。
3. 单击类型框中的“常量（十六进制）”，然后单击“确定”。然后，在“常量（十六进制）”对话框中，输入“020010”编写起始分隔符 STX（02h），站号 H（00h）和站号 L（10h）。完成后，单击“确定”。



5: 用户通信指令

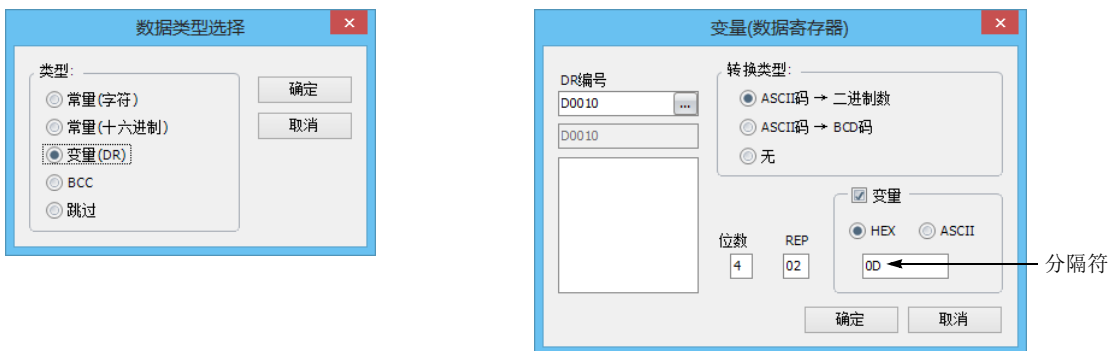
- 当接收指令对话框再次出现时，重复以上步骤。在“数据类型选择”对话框中，单击“跳过”后单击“确定”。然后，在“跳过”对话框中，输入“02”至位数框内，然后单击“确定”。



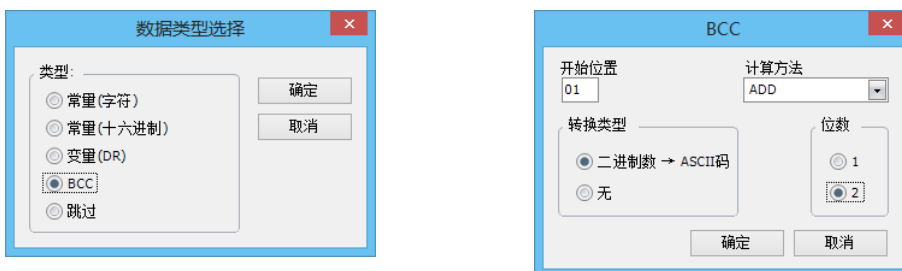
- 再次返回“数据类型选择”对话框，单击“常量（字符）”后单击“确定”。然后在“常量（字符）”对话框中，输入验证作为常量的“（2Ch）”。完成后，单击“确定”。



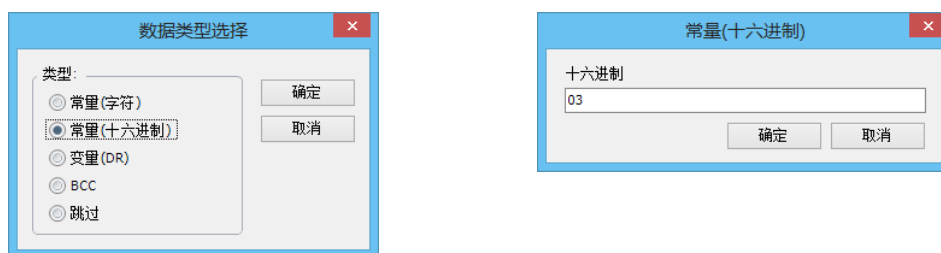
- 再次返回“数据类型选择”对话框，单击“变量（DR）”，然后单击“确定”。然后，在“变量（数据寄存器）”对话框中，输入“D0010”至寄存器编号框中，然后单击“ASCII码→二进制数”后 ASCII 码数将转换为二进制数。输入“4”至位数框（4 位数）中，然后输入“02”至 REP 框（重复 2 次）中。选中“变量”后选择“HEX”，并输入“0D”指定分隔符。完成后，单击“确定”。



- 再次返回“数据类型选择”对话框中，单击“BCC”，然后单击“确定”。然后，在“BCC”对话框中，在当前地址框中输入“01”，选择“ADD”作为计算方法，然后单击“二进制数→ASCII码”作为转换类型，再单击“2”作为位数。完成后，单击“确定”。



8. 再次返回“数据类型选择”对话框，单击“常量（十六进制）”，然后单击“确定”。然后，在“常量（十六进制）”对话框中，输入“03”以编写结束分隔符 ETX（03h）程序。完成后，单击“确定”。



9. 在“RXD（接收）”对话框中，输入“M0100”至目标 D1 框中，然后输入“D1000”至目标 D2 框中。完成后，单击“确定”。



编写 RXD1 指令完成后，按如下所示存储接收数据：

$$\begin{aligned} D10 & \boxed{1234h} = 4660 \\ D11 & \boxed{0005h} = 5 \end{aligned}$$

5: 用户通信指令

RS232C 线控信号

当 FC6A 型处于用户通信模式时，可以使用特殊数据寄存器启用或禁用端口 1 ~ 33 的 DSR 和 DTR 控制信号选项。

RS485 通信无法使用控制信号选项。

RS232C 通信盒（FC6A-PC1）的 RS 信号是始终处于 ON 的输出信号。

在维护通信模式状态下，DSR 无影响和 DTR 保持打开。

RS232C 线控信号的特殊数据寄存器

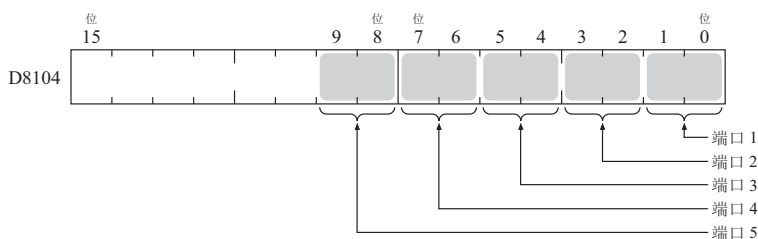
特殊数据寄存器 D8104 ~ D8734 分配给 RS232C 线控信号。

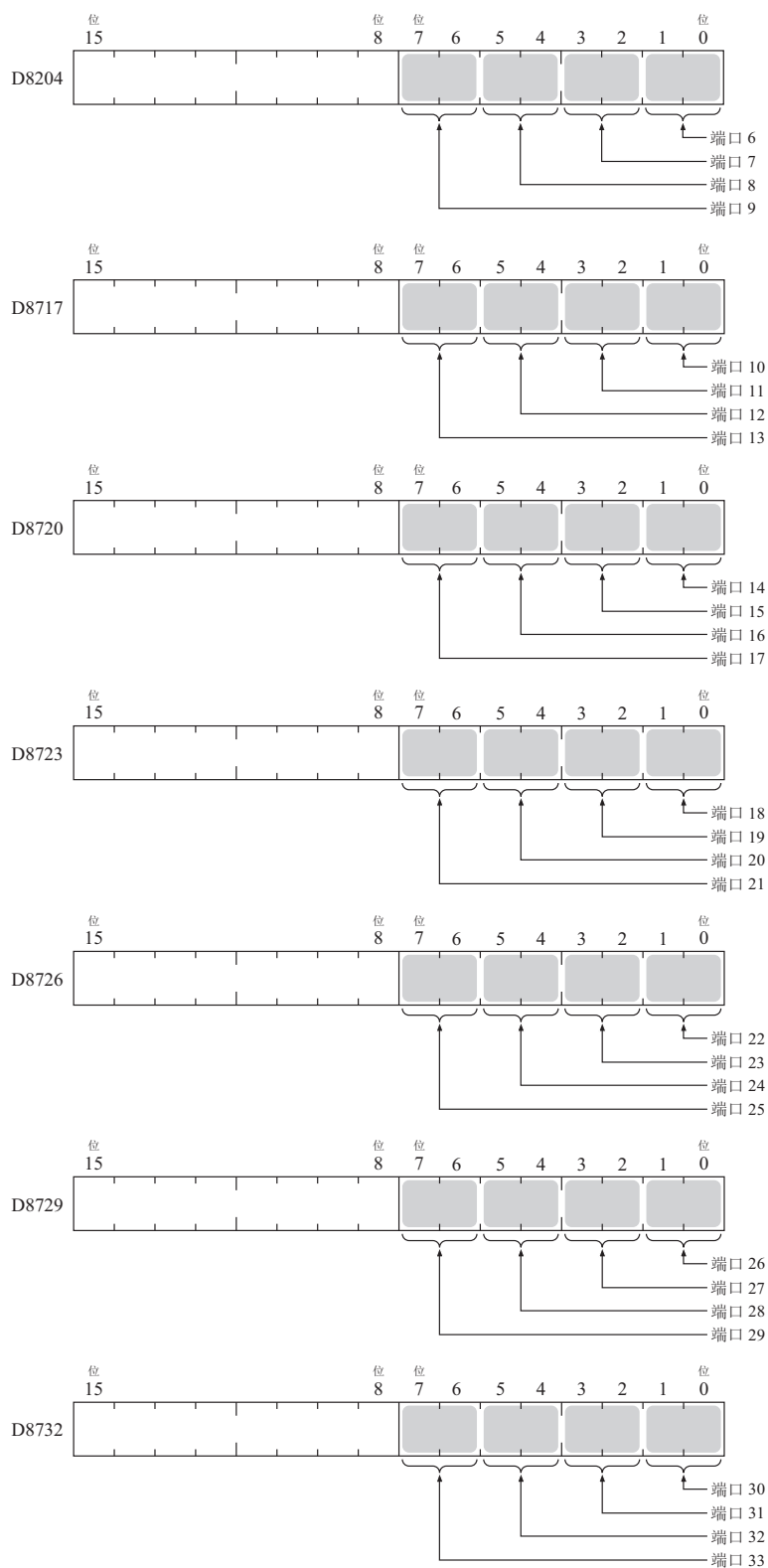
DR 编号	数据寄存器功能	DR 值已更新	读 / 写
D8104	控制信号状态（端口 1 ~ 5）	每次扫描	读
D8105	DSR 输入控制信号选项（端口 1 ~ 5）	每次扫描	读 / 写
D8106	DTR 输出控制信号选项（端口 1 ~ 5）	每次扫描	读 / 写
D8204	控制信号状态（端口 6 ~ 9）	每次扫描	读
D8205	DSR 输入控制信号选项（端口 6 ~ 9）	每次扫描	读 / 写
D8206	DTR 输出控制信号选项（端口 6 ~ 9）	每次扫描	读 / 写
D8717	控制信号状态（端口 10 ~ 13）	每次扫描	读
D8718	DSR 输入控制信号选项（端口 10 ~ 13）	每次扫描	读 / 写
D8719	DTR 输出控制信号选项（端口 10 ~ 13）	每次扫描	读 / 写
D8720	控制信号状态（端口 14 ~ 17）	每次扫描	读
D8721	DSR 输入控制信号选项（端口 14 ~ 17）	每次扫描	读 / 写
D8722	DTR 输出控制信号选项（端口 14 ~ 17）	每次扫描	读 / 写
D8723	控制信号状态（端口 18 ~ 21）	每次扫描	读
D8724	DSR 输入控制信号选项（端口 18 ~ 21）	每次扫描	读 / 写
D8725	DTR 输出控制信号选项（端口 18 ~ 21）	每次扫描	读 / 写
D8726	控制信号状态（端口 22 ~ 25）	每次扫描	读
D8727	DSR 输入控制信号选项（端口 22 ~ 25）	每次扫描	读 / 写
D8728	DTR 输出控制信号选项（端口 22 ~ 25）	每次扫描	读 / 写
D8729	控制信号状态（端口 26 ~ 29）	每次扫描	读
D8730	DSR 输入控制信号选项（端口 26 ~ 29）	每次扫描	读 / 写
D8731	DTR 输出控制信号选项（端口 26 ~ 29）	每次扫描	读 / 写
D8732	控制信号状态（端口 30 ~ 33）	每次扫描	读
D8733	DSR 输入控制信号选项（端口 30 ~ 33）	每次扫描	读 / 写
D8734	DTR 输出控制信号选项（端口 30 ~ 33）	每次扫描	读 / 写

控制信号状态

将控制线（DSR、DTR）的信号状态存储到 D8104、D8204、D8717、D8720、D8723、D8726、D8729、D8732。

DSR 信号及 DTR 信号的状态可存储到通信端口对应的以上设备。设备值以停止中、运行中的 END 处理进行刷新。设备内各通信端口的分配（位分配）如下所示。





2 位二进制值	DSR	DTR	说明
00	关	关	DSR 和 DTR 都关闭
01	关	开	DSR 打开
10	开	关	DTR 打开
11	开	开	DSR 和 DTR 都打开

5: 用户通信指令

RUN 和 STOP 模式下的 DSR 控制信号状态

扩展通信模式	3 位二进制值	DSR(输入) 状态	
		RUN 模式	STOP 模式
用户通信模式	000 (默认)	无影响	无影响 (TXD/RXD 禁用)
	001	开: 启用 TXD/RXD 关: 禁用 TXD/RXD	无影响 (TXD/RXD 禁用)
	010	开: 禁用 TXD/RXD 关: 启用 TXD/RXD	无影响 (TXD/RXD 禁用)
	011	开: 启用 TXD 关: 禁用 TXD	无影响 (TXD/RXD 禁用)
	100	开: 禁用 TXD 关: 启用 TXD	无影响 (TXD/RXD 禁用)
	>=101	无影响	无影响 (TXD/RXD 禁用)
维护模式	—	无影响	无影响

RUN 和 STOP 模式下的 DTR 控制信号状态

扩展通信模式	2 位二进制值	DTR(输出) 状态	
		RUN 模式	STOP 模式
用户通信模式	00 (默认)	开	关
	01	关	关
	10	RXD 已启: 开 RXD 已禁用: 关	关
	11	开	关
维护模式	—	开	开

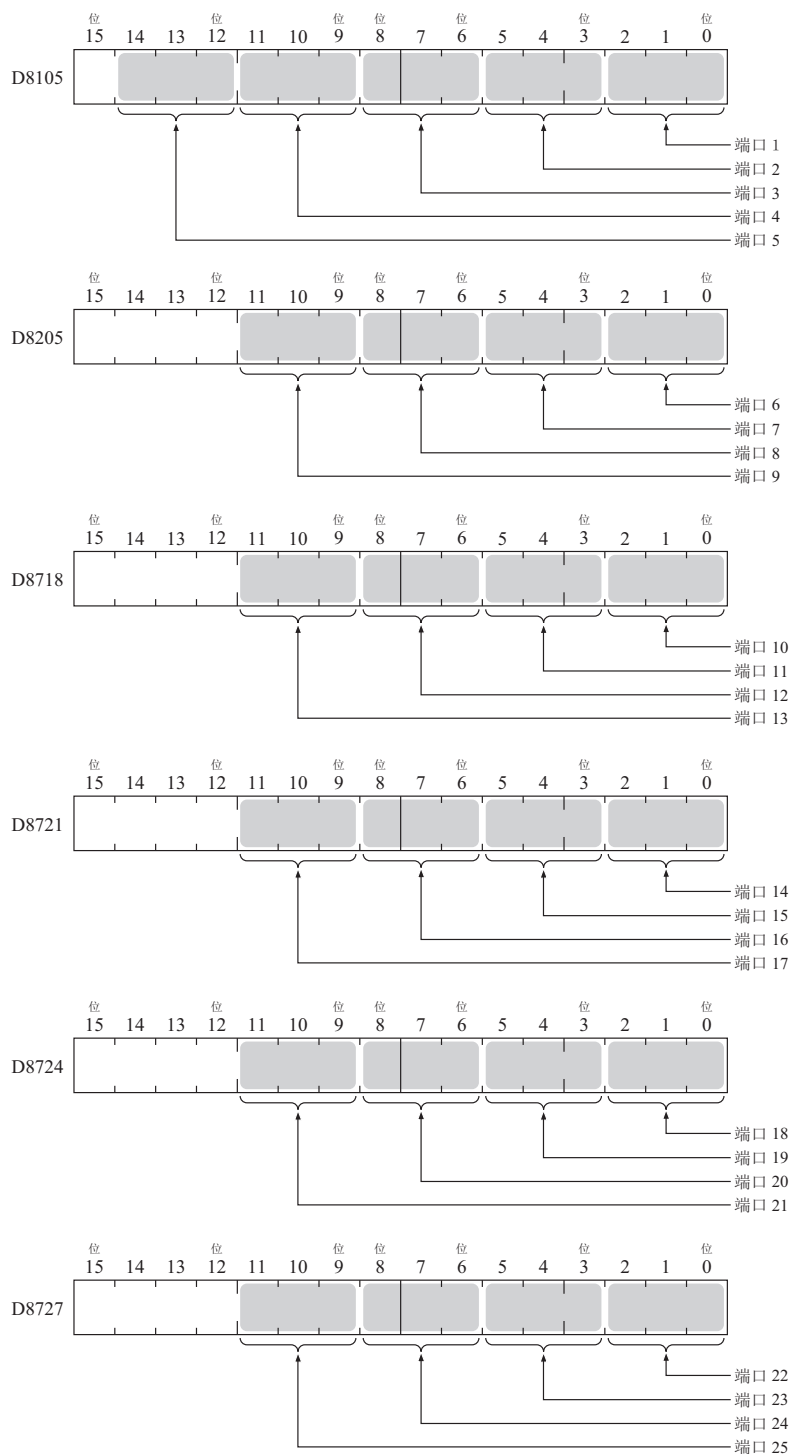
DSR 输入控制信号选项

外部设备使用 DSR 信号，向 FC6A 型传达外部设备可接收或是否发送有效数据等状态。

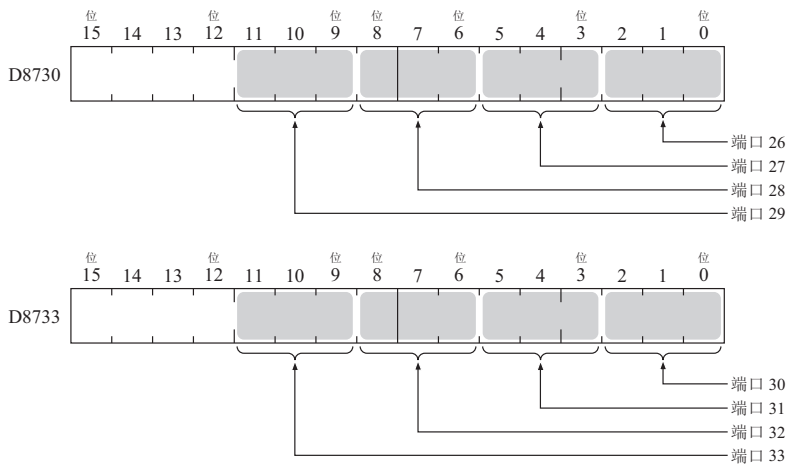
FC6A 型根据外部设备的 DSR 信号状态决定收发。

通过在 D8105、D8205、D8718、D8721、D8724、D8727、D8730、D8733 中设置值，可控制 DSR 信号。仅在用户通信时启用。

设备内各通信端口的分配（位分配）如下所示。



5: 用户通信指令



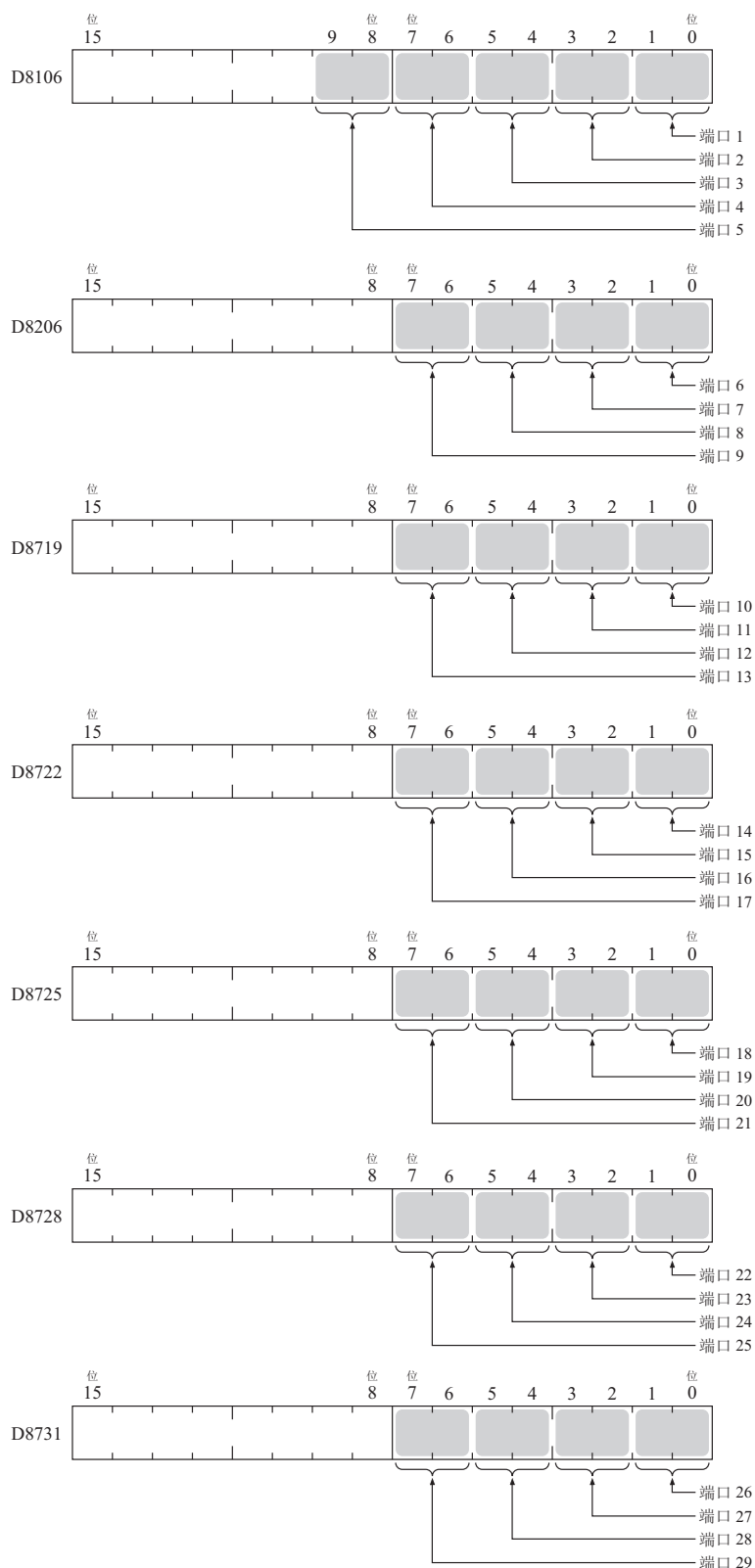
3 位二进制值	说明			
000	DSR 不可用于数据流控制。如无需进行 DSR 信号控制，请在该状态下使用。			
001	<p>当 DSR 打开时，FC6A 型可以发送和接收数据。</p> <p>DSR 信号 开 关</p> <p>发送/接收</p> <table border="1"> <tr> <td>不可能</td> <td>可能</td> <td>不可能</td> </tr> </table>	不可能	可能	不可能
不可能	可能	不可能		
010	<p>当 DSR 关闭时，FC6A 型可以发送和接收数据。</p> <p>DSR 信号 开 关</p> <p>发送/接收</p> <table border="1"> <tr> <td>不可能</td> <td>可能</td> <td>不可能</td> </tr> </table>	不可能	可能	不可能
不可能	可能	不可能		
011	<p>当 DSR 打开时，FC6A 型可以发送数据。此功能通常称为“忙控制”，用于控制向处理速度较慢的远程终端（如打印机）的发送。当远程终端忙时，将限制向远程终端的数据输入。</p> <p>DSR 信号 开 关</p> <p>发送</p> <table border="1"> <tr> <td>不可能</td> <td>可能</td> <td>不可能</td> </tr> </table>	不可能	可能	不可能
不可能	可能	不可能		
100	<p>当 DSR 关闭时，FC6A 型可以发送数据。</p> <p>DSR 信号 开 关</p> <p>发送</p> <table border="1"> <tr> <td>不可能</td> <td>可能</td> <td>不可能</td> </tr> </table>	不可能	可能	不可能
不可能	可能	不可能		
≥ 101	进行与值“000”相同的动作。DSR 未用作数据流控制。			

DTR 输出控制信号选项

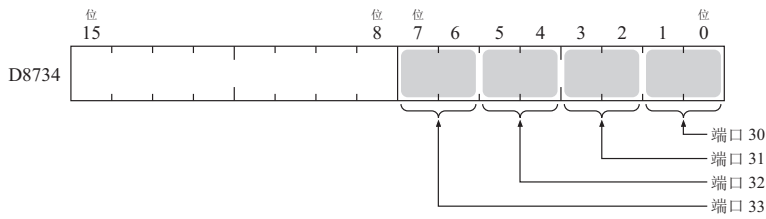
FC6A 型使用 DTR 信号，向外部设备传达 FC6A 型的控制状态及收发状态。

通过在 D8106、D8206、D8719、D8722、D8725、D8728、D8731、D8734 中设置值，可控制 DTR 信号。仅在用户通信时启用。

设备内各通信端口的分配（位分配）如下所示。



5: 用户通信指令



2 位二进制值	说明								
00	<p>当运行 FC6A 型时，无论 FC6A 型正在发送数据，还是在接收数据，DTR 都打开。当停止 FC6A 型时，DTR 保持关闭。使用此选项可指示 FC6A 型的操作状态。</p> <p>FC6A型</p> <table border="1"> <tr> <td>停止</td> <td>运行</td> <td>停止</td> </tr> </table> <p>DTR 信号</p> <p>开</p> <table border="1"> <tr> <td>关</td> <td>开</td> <td>关</td> </tr> </table>	停止	运行	停止	关	开	关		
停止	运行	停止							
关	开	关							
01	<p>无论 FC6A 型正在运行，还是已停止，DTR 保持关闭。</p> <p>FC6A型</p> <table border="1"> <tr> <td>停止</td> <td>运行</td> <td>停止</td> </tr> </table> <p>DTR 信号</p> <p>开</p> <table border="1"> <tr> <td>关</td> <td>关</td> <td>关</td> </tr> </table>	停止	运行	停止	关	关	关		
停止	运行	停止							
关	关	关							
10	<p>当 FC6A 型可以接收数据时，将打开 DTR。当 FC6A 型不能接收数据时，DTR 保持关闭。当需要控制接收数据流时要使用此选项。</p> <p>接收</p> <table border="1"> <tr> <td>不可能</td> <td>可能</td> <td>不可能</td> </tr> </table> <p>DTR 信号</p> <p>开</p> <table border="1"> <tr> <td>关</td> <td>开</td> <td>关</td> </tr> </table>	不可能	可能	不可能	关	开	关		
不可能	可能	不可能							
关	开	关							
11	进行与值“00”相同的动作。								

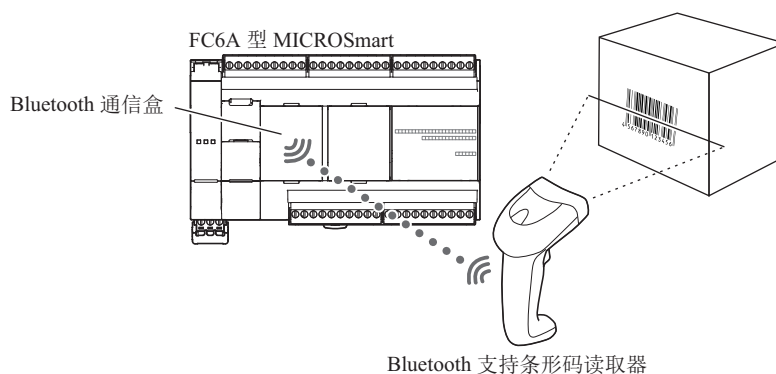
通过串行通信（Bluetooth）进行的用户通信

通过将 Bluetooth 通信盒连接到 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块的盒插槽、连接 Plus CPU 模块的盒基本模块或 HMI 模块的盒插槽，即可以 Bluetooth 通信的方式使用用户通信。

以 Bluetooth 通信的方式进行用户通信时，只要使用 TXD（用户通信发送）指令及 RXD（用户通信接收）指令，即可与连接 Bluetooth 的条形码读取器等外部设备进行数据的收发。

接收条形码读取器所读取数据的示例

将 Bluetooth 通信盒连接到 FC6A 型的盒插槽 1，与条形码读取器进行通信。



有关通信规格及 WindLDR 设置的详情，请参见第 9-1 页上的“Bluetooth 通信”。

5: 用户通信指令

通过以太网通信进行的用户通信

本节介绍以太网用户通信。以太网用户通信基于 TCP/UDP 协议。FC6A 型可作用户通信客户端 / 服务器。通过以太网用户通信指令（ETXD 和 ERXD 指令），FC6A 型可与网络中的设备交换数据。

除端口号和用户通信接收指令取消标志的分配外，以太网用户通信指令（ETXD 和 ERXD 指令）与 TXD 和 RXD 指令相同。有关 TXD 和 RXD 指令的详情，请参见第 5-2 页上的“TXD（发送）”和第 5-10 页上的“RXD（接收）”。

以太网用户通信概述

FC6A 型可用作以太网用户通信（TCP）的客户端 / 服务器。Plus CPU 模块也可作用户通信（UDP）。可以同时用于维护通信服务器、Modbus TCP 服务器和 Modbus TCP 客户端。

当使用 FC6A 型用户通信客户端时，FC6A 型可使用服务器设备协议访问服务器设备并与之通信。可为 All-in-One CPU 模块 / CAN J1939 CPU 模块所拥有的 8 个连接、Plus CPU 模块所拥有的 16 个连接分配各自不同的通信。

在第 5-41 页上的“将 FC6A 型作用户通信客户端”中介绍用户通信客户端的功能和设置。在第 5-47 页上的“用户通信服务器”中介绍用户通信服务器的功能和设置。

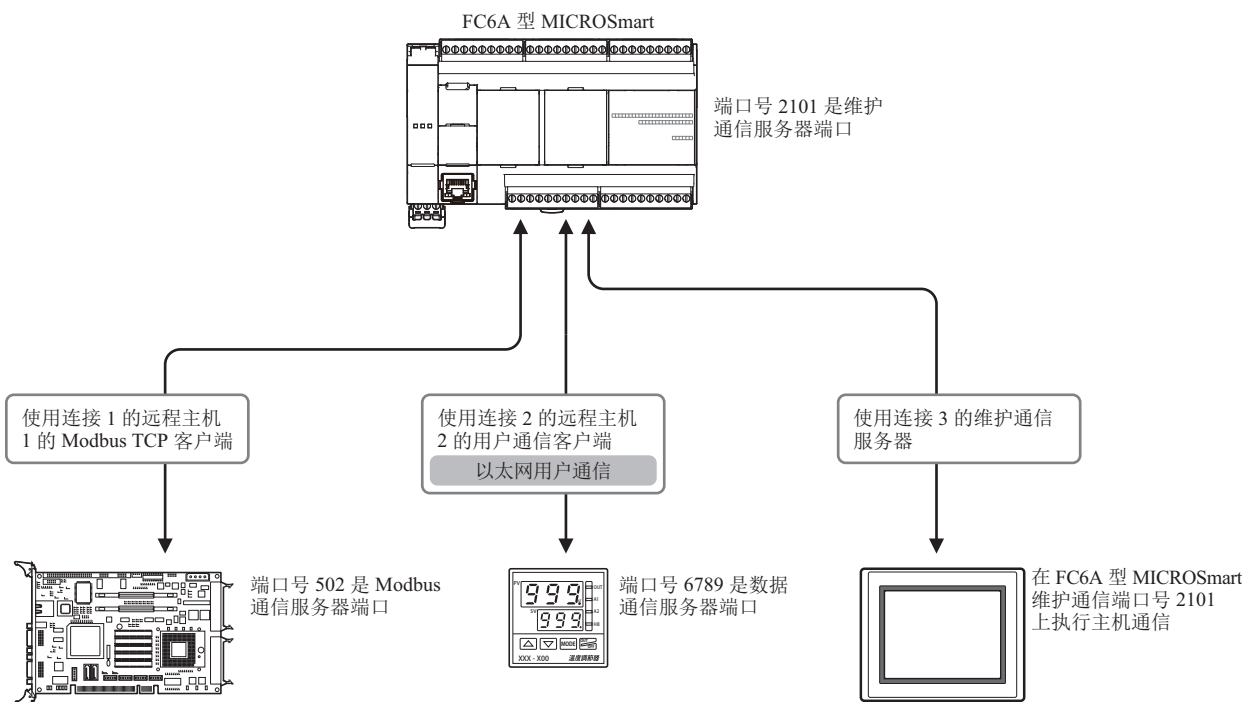
FC6A 型支持 TCP/UDP 协议。

FC6A 型可以使用 ETXD（以太网用户通信发送）指令和 ERXD（以太网用户通信接收）指令在其与网络上的设备之间发送和接收数据。

FC6A 型可以用作以太网用户通信客户端和服务器。

在 FC6A 型拥有的八个连接中，每个连接都可以分配给不同类型的通信。以太网用户通信可同时使用维护通信服务器、Modbus TCP 服务器和 Modbus TCP 客户端。

使用三个连接的以太网通信示例



FC6A 型功能设置连接设置

连接	通信协议	其他设置
1	Modbus TCP 客户端	目标: 远程主机 1
2	用户通信客户端	目标: 远程主机 2
3	维护通信服务器	端口号: 2101

远程主机表

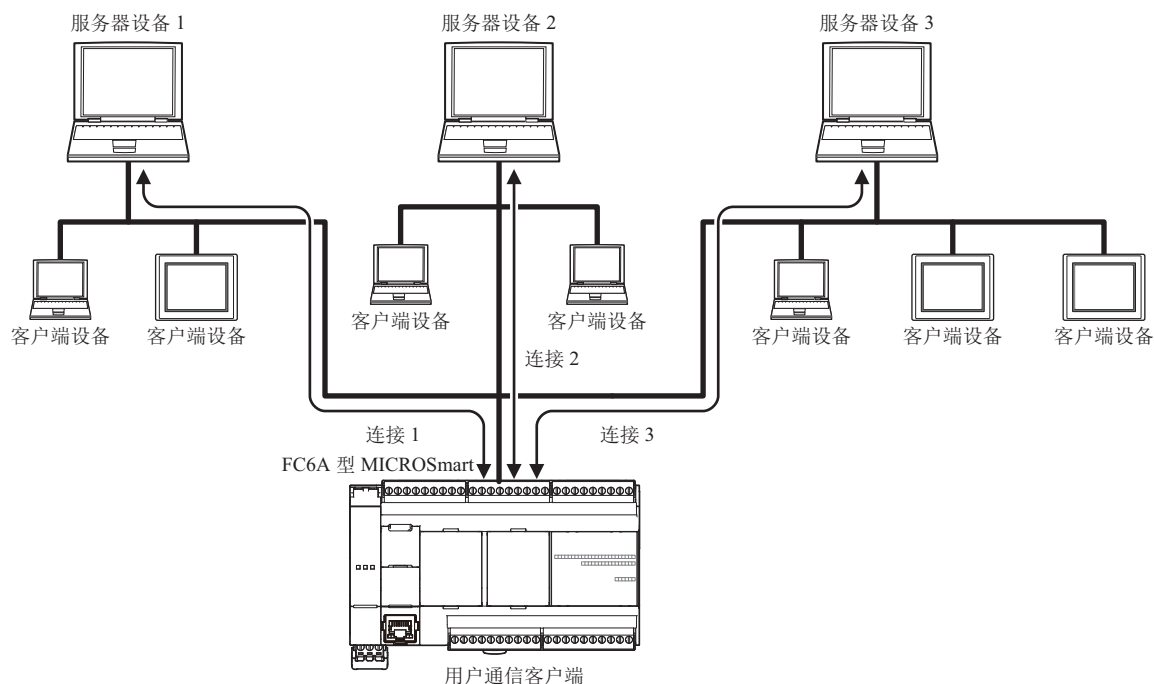
远程主机号	IP 地址	端口号
1	192.168.0.12	502
2	192.168.0.13	6789

将 FC6A 型用作用户通信客户端

通过网络将 FC6A 型连接到服务器设备，并使用以太网用户通信指令与服务器设备进行通信。

通过将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到用户通信客户端，可同时连接到不同的服务器设备并与它们进行通信。

将三个连接分配给用户通信客户端时

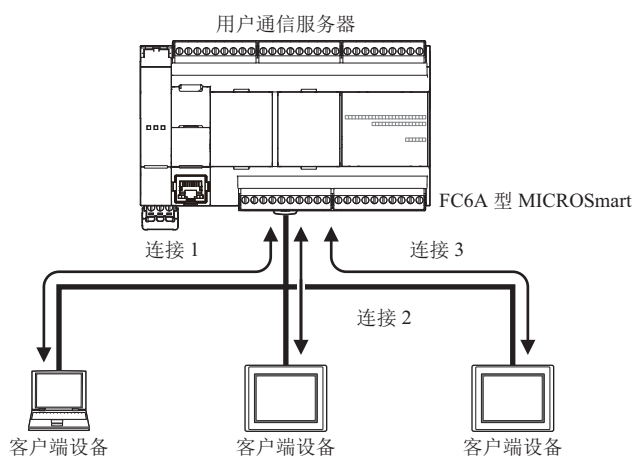


将 FC6A 型用作用户通信服务器

客户端设备连接到 FC6A 型，FC6A 型使用以太网用户通信指令与客户端设备进行通信。

通过将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到用户通信服务器，FC6A 型最多可同时连接 8 台客户端设备或最多可同时连接 16 台客户端设备并与它们进行通信。

将三个连接分配给用户通信服务器时



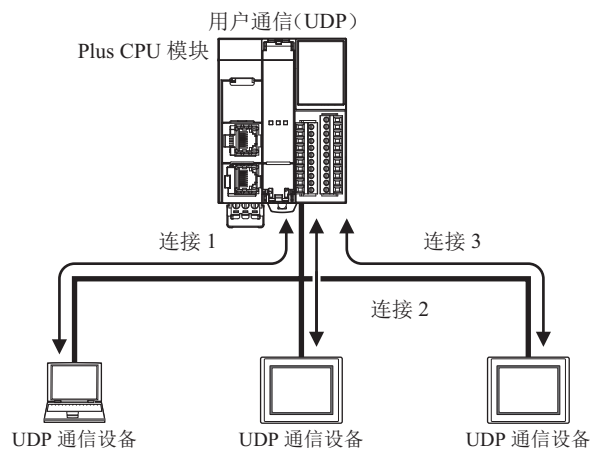
5: 用户通信指令

将 FC6A 型用作用户通信 (UDP) 时

将 UDP 通信设备连接到 Plus CPU 模块，并使用以太网用户通信指令与 UDP 通信设备进行通信。

通过将 Plus CPU 模块最多 16 个连接分配到用户通信 (UDP)，可在 Plus CPU 模块中最多同时连接 16 台 UDP 通信设备并与它们进行通信。

将三个连接分配给用户通信 (UDP) 时



用户通信客户端

当客户端连接设为用户通信客户端时，FC6A 型根据在 ETXD 和 ERXD 指令（用于客户端连接）中所作的设置与指定的服务器设备进行通信。可在“功能设置”对话框的连接设置选项卡中设置远程主机号及进行其他通信设置。

规格（用户通信客户端）

项目	用户通信客户端
远程主机号	1 ~ 255 根据特殊数据寄存器（D8268 ~ D8275、D8774 ~ D8781）切换
建立连接	<ul style="list-style-type: none"> 当执行 ETXD/ERXD 指令时 当 FC6A 型启动运行时*1
断开连接	<ul style="list-style-type: none"> 当 FC6A 型停止运行时 当特殊内部继电器（M8222 至 M8231 和 M8355 至 M8364）开启时
FC6A 型可以同时通信的远程主机数	每个用户通信客户端对应一台远程主机*2
接收超时时间	100 ~ 25,500ms（以 100ms 为递增单位）*3

*1 可在“功能设置”的“连接设置”中，选择启用/禁用。

*2 通过将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到用户通信客户端，FC6A 型可同时与 8 台或可同时与 16 台不同的服务器设备进行通信。

*3 不在括号中的数值为默认值。

切换远程主机号

设置用户通信客户端时，可使用特殊数据寄存器（D8268 ~ D8275、D8774 ~ D8781）更改通信目标。对特殊数据寄存器指定希望切换的通信目标远程主机号。指定了 1 ~ 255 以外的数值时，将变为功能设置中设置的远程主机号。

特殊数据寄存器	说明
D8268	连接 1 的远程主机号（1 ~ 255）
D8269	连接 2 的远程主机号（1 ~ 255）
D8270	连接 3 的远程主机号（1 ~ 255）
D8271	连接 4 的远程主机号（1 ~ 255）
D8272	连接 5 的远程主机号（1 ~ 255）
D8273	连接 6 的远程主机号（1 ~ 255）
D8274	连接 7 的远程主机号（1 ~ 255）
D8275	连接 8 的远程主机号（1 ~ 255）
D8774*1	连接 9 的远程主机号（1 ~ 255）
D8775*1	连接 10 的远程主机号（1 ~ 255）
D8776*1	连接 11 的远程主机号（1 ~ 255）
D8777*1	连接 12 的远程主机号（1 ~ 255）
D8778*1	连接 13 的远程主机号（1 ~ 255）
D8779*1	连接 14 的远程主机号（1 ~ 255）
D8780*1	连接 15 的远程主机号（1 ~ 255）
D8781*1	连接 16 的远程主机号（1 ~ 255）

*1 仅 Plus CPU 模块

5: 用户通信指令

建立 / 断开用户通信客户端连接

当设置用户通信客户端时，根据 TCP 协议建立连接。当执行 ETXD/ERXD 指令或当 FC6A 型启动运行时建立连接（见上述部分）*1。建立连接后，将保持连接开启状况，直到 FC6A 型停止运行或分配到连接的特殊内部继电器（M8222 至 M8231 及 M8355 至 M8364）开启为止。

特殊内部继电器	说明	运行
M8222	连接 1 断开	当继电器开启后，相应的连接断开。开启后，也无法建立连接。
M8223	连接 2 断开	
M8224	连接 3 断开	
M8225	连接 4 断开	
M8226	连接 5 断开	
M8227	连接 6 断开	
M8230	连接 7 断开	
M8231	连接 8 断开	
M8355 ^{*2}	连接 9 断开	
M8356 ^{*2}	连接 10 断开	
M8357 ^{*2}	连接 11 断开	
M8360 ^{*2}	连接 12 断开	
M8361 ^{*2}	连接 13 断开	
M8362 ^{*2}	连接 14 断开	
M8363 ^{*2}	连接 15 断开	
M8364 ^{*2}	连接 16 断开	

*1 可在“功能设置”的“连接设置”中，选择启用 / 禁用。

*2 仅 Plus CPU 模块

用户通信客户端接收指令（ERXD）取消标记

各客户端连接所用的用户通信接收指令取消标记的分配如下表所示。有关用户通信接收指令取消标记的详情，请参见第 5-22 页上的“用户通信接收指令取消标志”。

特殊内部继电器	说明
M8200	用户通信接收指令取消标记（连接 1）
M8201	用户通信接收指令取消标记（连接 2）
M8202	用户通信接收指令取消标记（连接 3）
M8203	用户通信接收指令取消标记（连接 4）
M8204	用户通信接收指令取消标记（连接 5）
M8205	用户通信接收指令取消标记（连接 6）
M8206	用户通信接收指令取消标记（连接 7）
M8207	用户通信接收指令取消标记（连接 8）
M8334 ^{*1}	用户通信接收指令取消标记（连接 9）
M8335 ^{*1}	用户通信接收指令取消标记（连接 10）
M8336 ^{*1}	用户通信接收指令取消标记（连接 11）
M8337 ^{*1}	用户通信接收指令取消标记（连接 12）
M8340 ^{*1}	用户通信接收指令取消标记（连接 13）
M8341 ^{*1}	用户通信接收指令取消标记（连接 14）
M8342 ^{*1}	用户通信接收指令取消标记（连接 15）
M8343 ^{*1}	用户通信接收指令取消标记（连接 16）

*1 仅 Plus CPU 模块

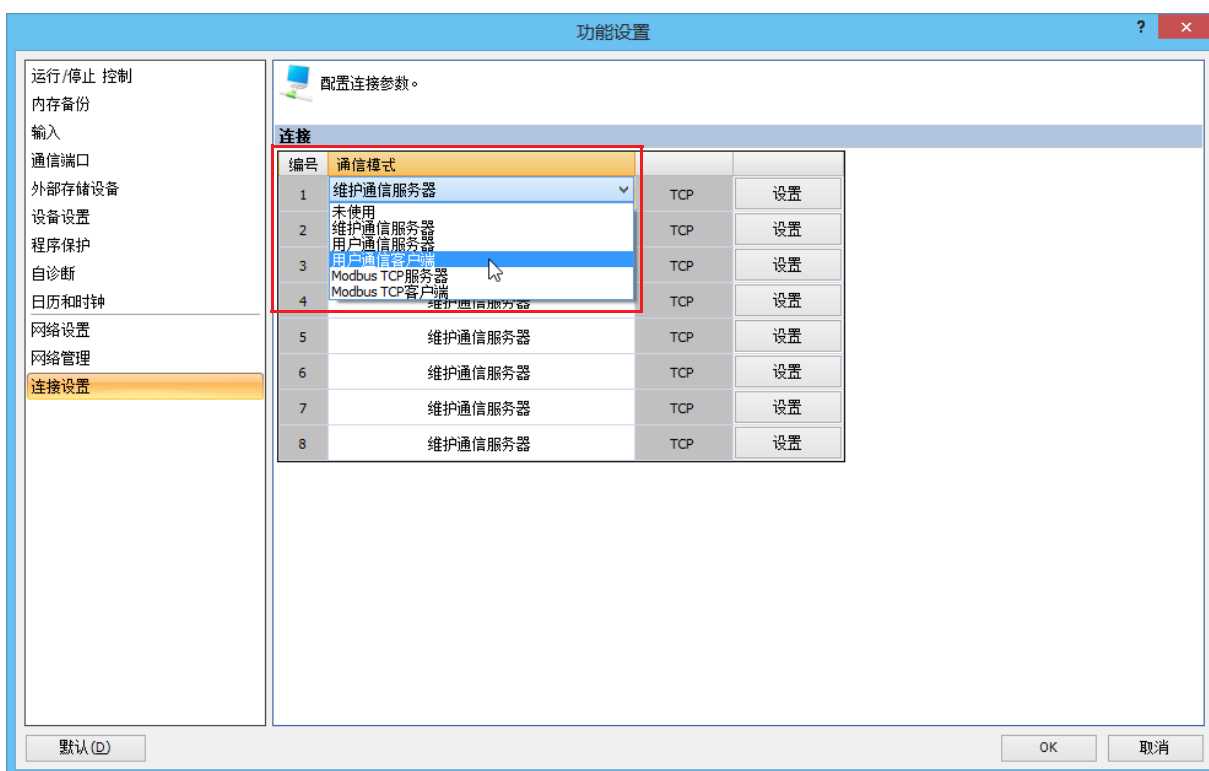
设置 WindLDR（用户通信客户端）

若要使用用户通信客户端，可在“功能设置”对话框中设定用户客户端通信设置，然后将用户程序下载到 FC6A 型。

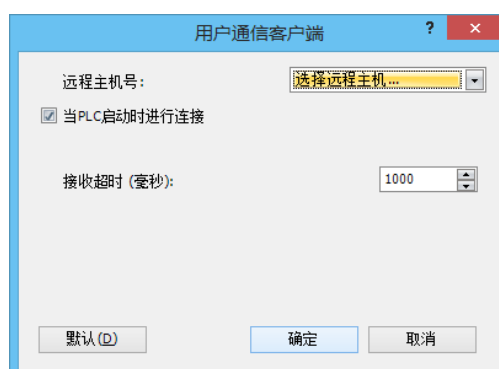
1. 从 WindLDR 菜单栏选择“设置”，然后点击“连接设置”。

出现“功能设置”对话框。

2. 选择“用户通信客户端”作为客户端连接 1 的通信模式。



出现“用户通信客户端”对话框。

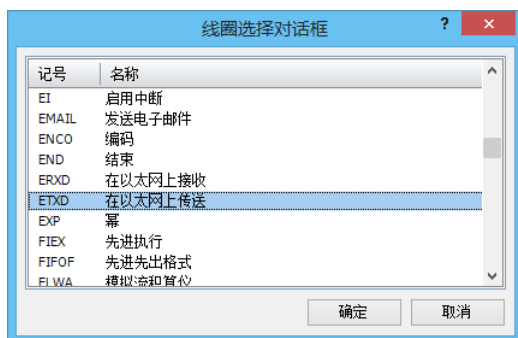


设置远程主机号和接收超时。如果想要在 FC6A 型启动运行时建立连接，请选择“当 PLC 启动时进行连接”。单击“确定”按钮关闭对话框。

5: 用户通信指令

3. 编辑用户程序。

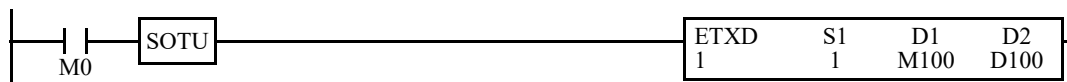
如果要在梯形图编辑器中插入以太网用户通信指令，可在“线圈选择对话框”中选择以太网用户通信指令（ETXD 或 ERXD 指令）。



出现“ETXD（在以太网上传送）”对话框。



选择“ETXD”（在以太网上传送）或“ERXD”（在以太网上接收）作为指令类型发送或接收数据。从 1 至 8 中选择客户端连接，并指定 S1、D1 和 D2。单击“确定”按钮关闭对话框。



4. 下载用户程序。

完成用户通信客户端的设置。

以太网用户通信指令（ETXD 和 ERXD 指令）的规格与 TXD 和 RXD 指令相同。有关 TXD 和 RXD 指令的详情，请参见第 5-2 页上的“TXD（发送）”和第 5-10 页上的“RXD（接收）”。

用户通信服务器

当服务器连接设为用户通信服务器时，客户端设备可以访问 FC6A 型并与之通信。FC6A 型根据在 ETXD 和 ERXD 指令（用于服务器连接）中所作设置与客户端设备进行通信。可在“功能设置”对话框的“连接设置”选项卡中设置本地主机号及进行其他通信设置。

规格（用户通信服务器）

项目	用户通信服务器
本地主机端口号	2102 ~ 2117 (可在 0 和 65535 之间进行更改)
可与 FC6A 型同时通信的客户端数	每一用户通信服务器对应一个客户端 ^{*1}
接收超时时间	100 ~ 25,500ms (以 100ms 为递增单位)

*1 通过将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到用户通信服务器，可同时将 FC6A 型最多 8 台客户端设备或最多 16 台的客户端设备连接到 FC6A 型并与它们进行通信。

用户通信服务器接收指令 (ERXD) 取消标记

各服务器连接所用的用户通信接收指令取消标志的分配如下表所示。有关用户通信接收指令取消标志的详情，请参见第 5-22 页上的“用户通信接收指令取消标志”。

设备地址	说明
M8200	用户通信接收指令取消标记 (连接 1)
M8201	用户通信接收指令取消标记 (连接 2)
M8202	用户通信接收指令取消标记 (连接 3)
M8203	用户通信接收指令取消标记 (连接 4)
M8204	用户通信接收指令取消标记 (连接 5)
M8205	用户通信接收指令取消标记 (连接 6)
M8206	用户通信接收指令取消标记 (连接 7)
M8207	用户通信接收指令取消标记 (连接 8)
M8334 ^{*1}	用户通信接收指令取消标记 (连接 9)
M8335 ^{*1}	用户通信接收指令取消标记 (连接 10)
M8336 ^{*1}	用户通信接收指令取消标记 (连接 11)
M8337 ^{*1}	用户通信接收指令取消标记 (连接 12)
M8340 ^{*1}	用户通信接收指令取消标记 (连接 13)
M8341 ^{*1}	用户通信接收指令取消标记 (连接 14)
M8342 ^{*1}	用户通信接收指令取消标记 (连接 15)
M8343 ^{*1}	用户通信接收指令取消标记 (连接 16)

*1 仅 Plus CPU 模块

5: 用户通信指令

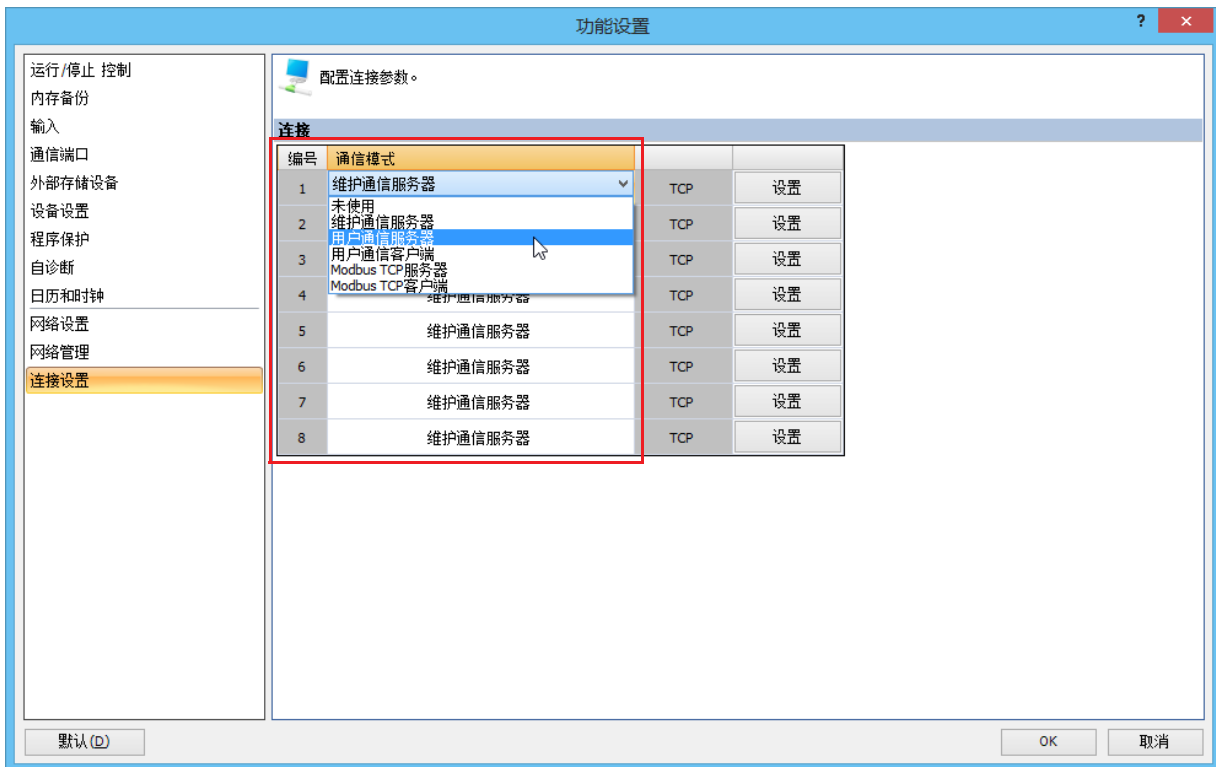
设置 WindLDR（用户通信服务器）

若要使用用户通信服务器，可在“功能设置”对话框中设定用户通信服务器设置，然后将用户程序下载到 FC6A 型。

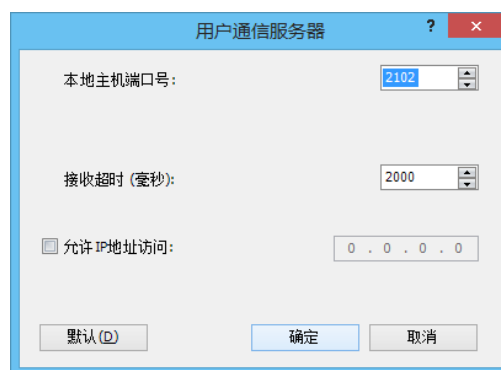
1. 从 WindLDR 菜单栏选择“设置”，然后点击“连接设置”。

出现“功能设置”对话框。

2. 选择“用户通信服务器”作为服务器连接 1 的通信模式。



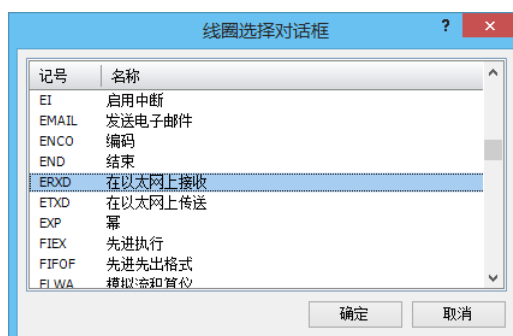
出现“用户通信服务器”对话框。



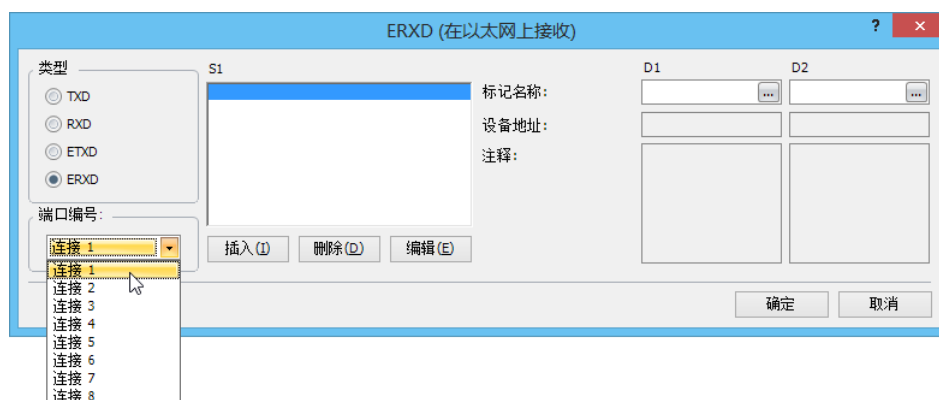
设置本地主机端口号和接收超时。如果想要限制使用 IP 地址访问，请设置允许的 IP 地址。

3. 编辑用户程序。

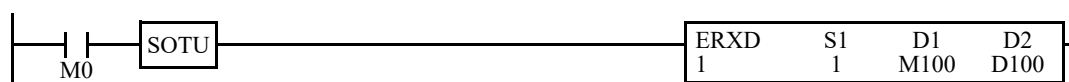
如果要在梯形图编辑器中插入以太网用户通信指令，可在“线圈选择对话框”中选择以太网用户通信指令（ETXD 或 ERXD 指令）。



出现“ERXD（在以太网上接收）”对话框。



选择“ETXD”（在以太网上发送）和“ERXD”（在以太网上接收）作为指令类型发送或接收数据。从 1 至 16 中选择服务器连接，并指定 S1、D1 和 D2。单击“确定”按钮关闭对话框。



4. 下载用户程序。

完成用户通信服务器的设置。

以太网用户通信指令（ETXD 和 ERXD 指令）的规格与 TXD 和 RXD 指令相同。有关 TXD 和 RXD 指令的详情，请参见第 5-2 页上的“TXD（发送）”和第 5-10 页上的“RXD（接收）”。

5: 用户通信指令

用户通信 (UDP)

Plus CPU 模块通过用户通信服务器, 根据相应连接编号所设置的 ETXD (以太网用户通信发送) 指令及 ERXD (以太网用户通信接收) 指令与 UDP 通信设备进行通信。

用户通信服务器端口号的设置及通信设置可在“设置”选项卡的“功能设置”中设置“连接设置”。

规格

项目	用户通信服务器
远程主机号	1 ~ 255
本地主机端口号	0 ~ 65535
允许 IP 地址访问	0.0.0.0 ~ 255.255.255.255

分配 ERXD (以太网用户通信接收) 指令的接收取消标记

表示各连接的接收取消标记的分配。

用户通信接收指令的接收预处理已完成, 如果在接收中 (状态代码 32) 的状态下打开用户通信接收指令取消标记, 则会取消执行相应连接对应的所有接收指令。接收数据等待的状态较长, 希望取消执行接收指令时启用。

如需激活已取消的接收指令, 请在关闭用户通信接收指令取消标记后, 重新打开接收指令的输入条件。

用户通信接收指令取消标记在各连接中作为特殊内部继电器进行如下分配。

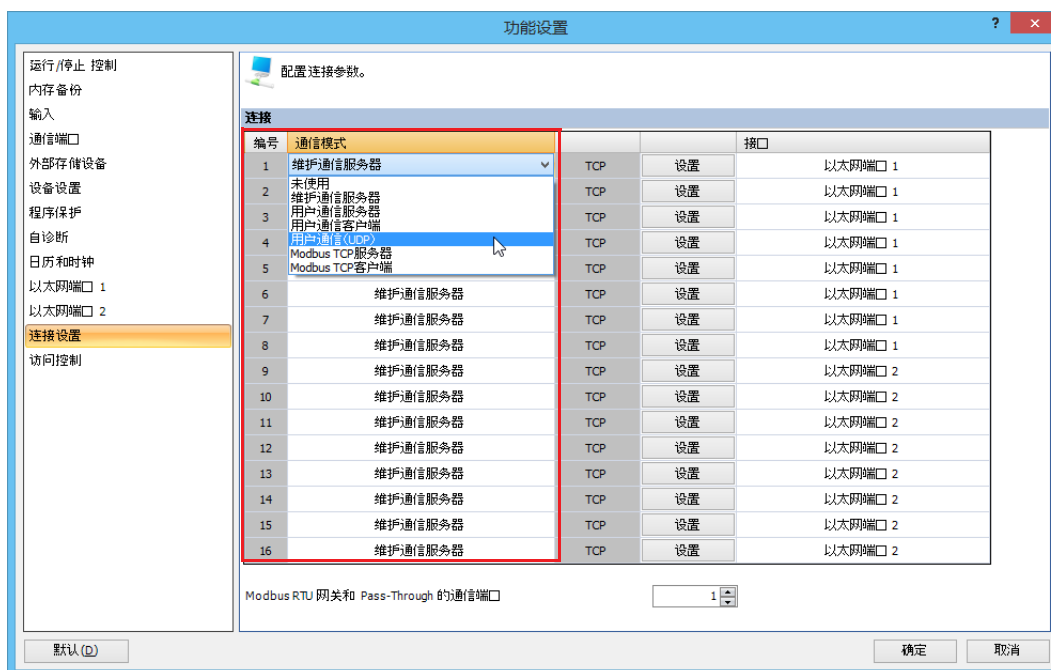
特殊内部继电器	说明
M8200	用户通信接收指令取消标记 (连接 1)
M8201	用户通信接收指令取消标记 (连接 2)
M8202	用户通信接收指令取消标记 (连接 3)
M8203	用户通信接收指令取消标记 (连接 4)
M8204	用户通信接收指令取消标记 (连接 5)
M8205	用户通信接收指令取消标记 (连接 6)
M8206	用户通信接收指令取消标记 (连接 7)
M8207	用户通信接收指令取消标记 (连接 8)
M8334	用户通信接收指令取消标记 (连接 9)
M8335	用户通信接收指令取消标记 (连接 10)
M8336	用户通信接收指令取消标记 (连接 11)
M8337	用户通信接收指令取消标记 (连接 12)
M8340	用户通信接收指令取消标记 (连接 13)
M8341	用户通信接收指令取消标记 (连接 14)
M8342	用户通信接收指令取消标记 (连接 15)
M8343	用户通信接收指令取消标记 (连接 16)

设置 WindLDR

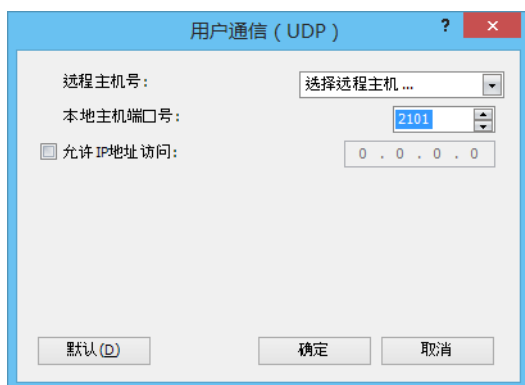
将 Plus CPU 模块设置为用户通信（UDP）。

在下例中，将对使用 WindLDR 在连接 1 中设置用户通信（UDP）的步骤进行介绍。

1. 从 WindLDR 菜单栏中，选择“设置”>“功能设置”>“连接设置”。
出现“功能设置”对话框。
2. 单击连接编号 1 的“通信模式”，选择“用户通信（UDP）”。
将显示“用户通信（UDP）”对话框。



3. 设置“远程主机号”、“本地主机端口号”。如果启用限制 IP 地址访问，应选中“允许 IP 地址访问”复选框，设置允许访问的 IP 地址。
有关远程主机号的详情，请参见第 3-21 页上的“远程主机列表”。



4. 单击“确定”按钮。

至此，用户通信（UDP）的设置完成。

接着，使用已设置连接的对应用户通信指令创建用户程序，下载到 Plus CPU 模块。

5: 用户通信指令

用户通信错误

当发生用户通信错误时，错误代码将存储到 TXD 指令中指定为发送状态或 RXD 指令中指定为接收状态的数据寄存器中。当出现多个错误时，最后的错误代码会覆盖所有以前的错误，并被存储至状态数据寄存器中。

状态数据寄存器中还包含发送 / 接收状态代码。要从状态数据寄存器中提取用户通信错误代码，请该值除以 16。余数即为用户通信错误代码。有关发送状态的详情，请参见第 5-8 页上的“D2 (目标 2)”；有关接收状态的详情，请参见第 5-21 页上的“D2 (目标 2)”。

要更正错误，请参照以下所述的错误原因更正用户程序。

用户通信错误代码

用户通信错误代码	错误原因	发送 / 接收完成输出
1	同时开始输入至 5 个以上 TXD 指令。	打开梯形图前 5 个 TXD 指令的发送完成输出。
2	发送目标忙超时	发送完成输出持续运行。
3	同时打开启动输入超过 5 个以上带有起始分隔符的 RXD 指令。	在梯形图的前 5 个 RXD 指令中，如果起始分隔符与接收数据的第 1 字节匹配，则 RXD 指令的接收完成输出将继续。
4	在执行没有起始分隔符的 RXD 指令时，执行另一个有或没有起始分隔符的 RXD 指令。	稍小地址处的 RXD 指令的接收完成输出继续。
5	在执行没有起始分隔符的 RXD 指令时，执行另一个相同的起始分隔符的 RXD 指令。	不影响接收完成输出。
7	已接收数据的第 1 个字节与指定起始分隔符不匹配。	不影响接收完成输出。 如果随后收到带有匹配起始分隔符的输入数据，则接收完成输出继续。
8	当在接收格式中指定 ASCII 码→二进制数或 ASCII 码→BCD 码转换时，将接收 0~9 和 A~F 以外的代码。(在转换时这些代码被视作 0。)	接收完成输出继续。
9	从 RXD 指令计算出的 BCC 与添加至接收数据的 BCC 不匹配。	接收完成输出继续。
10	RXD 指令中指定的包含常量的结束分隔符代码与接收的常量不匹配。	接收完成输出继续。
11	字符间的接收超时时间 (收到数据的 1 个字节后，下一个字节没有在接收超时值指定的时间内收到。)	接收完成输出继续。
12	数据溢出错误 (接收处理完成之前，接收下一个数据。)	接收完成输出关闭。
13	数据格式错误 (起始位或停止位检测错误)	不影响完成输出。
14	奇偶错误 (找到奇偶校验错误。)	不影响完成输出。
15	即使端口设置或连接设置未设置为用户通信模式，系统也会使用用户通信指令。或者，在未连接通信盒或通信模块的情况下，在配置为用户通信模式的端口上使用了用户通信指令。	不影响完成输出。

ASCII 字符代码表

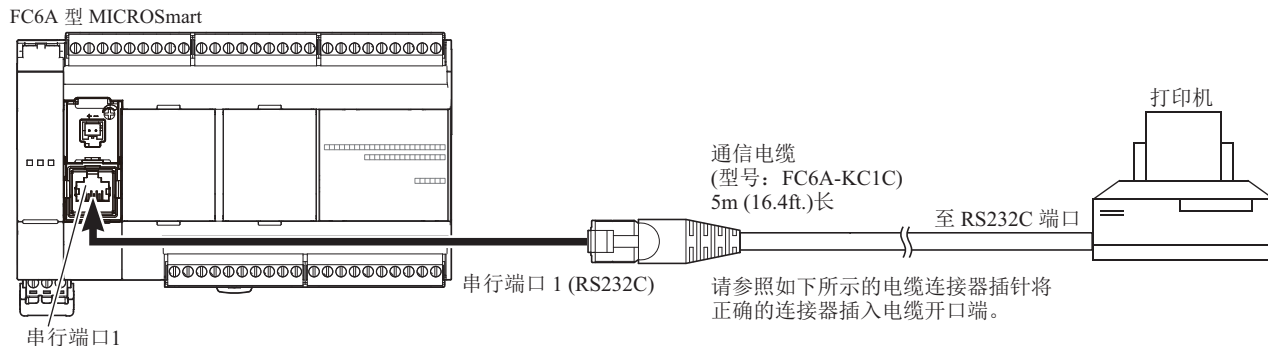
低位	高位															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	N _U L	D _L E	SP	0	@	P	`	P								
十进制	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	S _O H	D _C 1	!	1	A	Q	a	q								
十进制	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	S _T X	D _C 2	”	2	B	R	b	r								
十进制	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	E _T X	D _C 3	#	3	C	S	c	s								
十进制	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	E _O T	D _C 4	\$	4	D	T	d	t								
十进制	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	E _N Q	N _A K	%	5	E	U	e	u								
十进制	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	A _C K	S _Y N	&	6	F	V	f	v								
十进制	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	B _E L	E _T B	,	7	G	W	g	w								
十进制	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	BS	C _A N	(8	H	X	h	x								
十进制	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	HT	EM)	9	I	Y	i	y								
十进制	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	LF	S _U B	*	:	J	Z	j	z								
十进制	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	VT	E _S C	+	;	K	[k	{								
十进制	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	FF	FS	,	<	L	\	l									
十进制	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	CR	GS	-	=	M]	m	}								
十进制	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	SO	RS	.	>	N	^	n	~								
十进制	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	SI	US	/	?	O	_	o									
十进制	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

5: 用户通信指令

示例程序 - 用户通信 TXD

此示例说明使用用户通信 TXD1（发送）指令发送数据至打印机的程序。

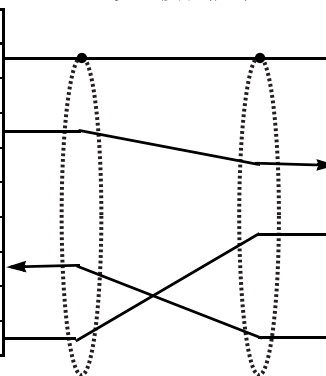
系统设置



RJ45 连接器插针

说明	颜色	插针编号
屏蔽	—	盖
RXD 接收数据	白色 / 橙色	1
TXD 发送数据	橙色	2
DTR 数据终端就绪	白色 / 绿色	3
A	蓝色	4
B	白色 / 蓝色	5
DSR 数据设置就绪	绿色	6
NC 没有连接	白色 / 褐色	7
SG 信号接地	褐色	8

电缆连接和插针



D-sub 9 针连接器插针

插针编号	说明
屏蔽	
1	NC 没有连接
2	NC 没有连接
3	DATA 接收数据
4	NC 没有连接
5	SG 信号接地
6	NC 没有连接
7	NC 没有连接
8	BUSY 忙信号
9	NC 没有连接

BUSY 终端的名称因打印机而异，如 DTR。该终端的功能是无论打印机是否已准备好打印数据都将信号发送至远程设备。因为该信号的操作根据打印机的不同而不同，所以请在连接电缆前确认操作。

注意 请勿将任何电线连接至 NC (没有连接) 针脚; 否则 FC6A 型和打印机可能无法正常工作并受到损坏。

操作说明

每分钟都会打印计数器 C2 的数据和寄存器 D30。打印输出示例显示在右侧。

编写特殊数据寄存器

特殊数据寄存器 D8105 用于监控 BUSY 信号和控制打印数据的发送。

特殊 DR	值	说明
D8105	3 (011)	当 DSR 打开 (不忙) 时, FC6A 型将发送数据。 当 DSR 关闭 (忙) 时, FC6A 型将停止发送数据。 如果关闭时间超过限制 (约 5 秒钟), 将出现发送忙超时错误, 且不再发送剩余数据。发送状态寄存器存储错误代码。 请参见第 5-52 页上的“用户通信错误代码”。

FC6A 型监控 DSR 信号以避免打印机的接收缓冲区溢出。有关 DSR 信号, 请参见第 5-35 页上的“DSR 输入控制信号选项”。

打印输出示例

```

--- PRINT TEST ---

11H 00M

CNT2...0050
D030...3854

--- PRINT TEST ---

11H 01M

CNT2...0110
D030...2124
    
```

在 WindLDR 功能设置中设置用户通信

此示例使用串行端口 1 (RS232C)，请使用 WindLDR 在“功能设置”中选择端口 1 的用户协议。请参见第 5-25 页上的“设置 WindLDR”。

设置通信设置

设置通信参数与打印机的参数匹配。请参见第 5-25 页上的“设置 WindLDR”。有关打印机通信设置的详细信息，请参阅打印机用户手册。示例如下：

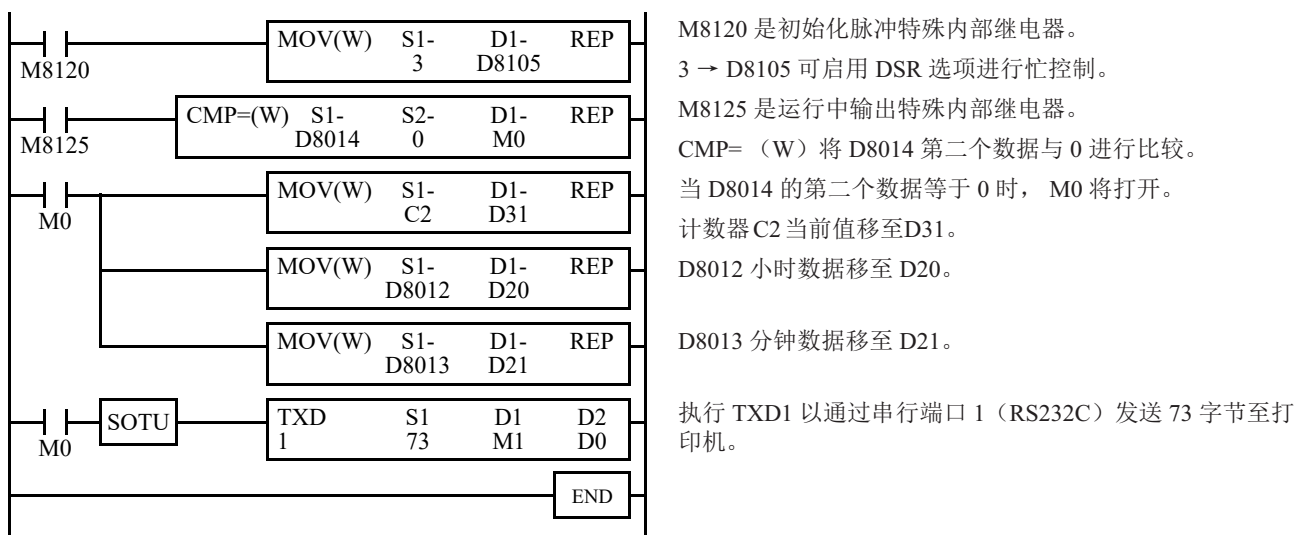
通信设置：

通信速度： 9,600bps
 数据长度： 8
 奇偶校验： 无
 停止位： 1

注释：接收超时值用于用户通信模式下的 RXD 指令中。因为该示例仅使用 TXD 指令，所以接收超时值不受影响。

梯形图

使用 CMP= (比较等于) 指令将特殊数据寄存器 D8014 中的第二个数据与 0 进行比较。每次条件满足时，将执行 TXD1 指令以发送 C2 和 D30 数据至打印机。此示例程序中省略了计数器 C2 的计数电路。



发送指令 S1 的设置内容

SP	SP	SP	-	-	-	SP	P	R	I	N	T	SP	T
20h	20h	20h	2Dh	2Dh	2Dh	20h	50h	52h	49h	4Eh	54h	20h	54h
E	S	T	SP	-	-	-	CR	LF	CR	LF	SP	SP	SP
45h	53h	54h	20h	2Dh	2Dh	2Dh	0Dh	0Ah	0Dh	0Ah	20h	20h	20h
D20 转换: BCD → ASCII 位数: 2 REP: 01													
H	SP												
48h	20h												
D21 转换: BCD → ASCII 位数: 2 REP: 01													
M	CR	LF	CR	LF									
4Dh	0Dh	0Ah	0Dh	0Ah									
SP	SP	SP	C	N	T	2
20h	20h	20h	43h	4Eh	54h	32h	2Eh	2Eh	2Eh	2Eh	2Eh	2Eh	2Eh
D31 转换: BCD → ASCII 位数: 4 REP: 01													
CR	LF	SP	SP	SP	D	0	3	0
0Dh	0Ah	20h	20h	20h	44h	30h	33h	30h	2Eh	2Eh	2Eh	2Eh	2Eh
D30 转换: BCD → ASCII 位数: 4 REP: 01													
CR	LF	CR	LF										
0Dh	0Ah	0Dh	0Ah										

对 D20 小时数据进行 BCD 码 → ASCII 码转换，并发送 2 位数字。

对 D21 分钟数据进行 BCD 码 → ASCII 码转换，并发送 2 位数字。

对 D31 计数器 C2 数据进行 BCD 码 → ASCII 码转换，并发送 4 位数字。

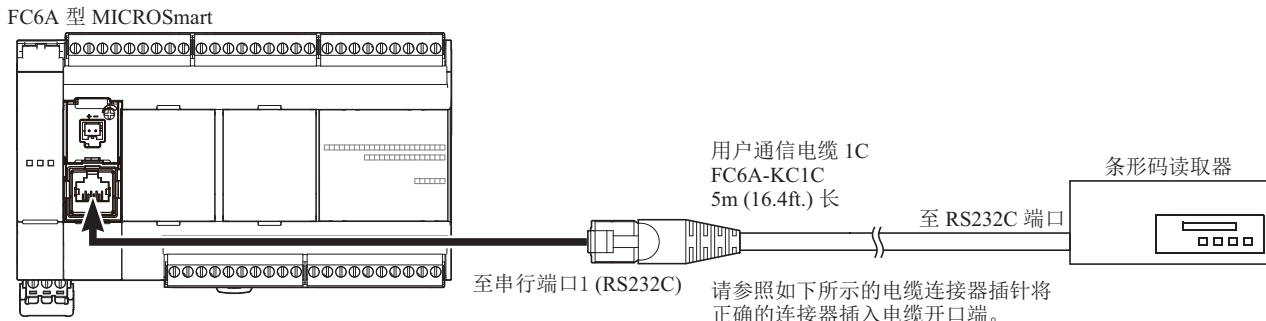
对 D30 数据进行 BCD 码 → ASCII 码转换，并发送 4 位数字。

5: 用户通信指令

示例程序 - 用户通信 RXD

该示例说明使用用户通信 RXD1（接收）指令接收来自配有 RS232C 端口的条形码读取器的数据的程序。

系统设置

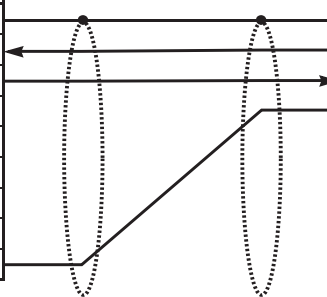


RJ45连接器插针

说明	颜色	插针编号
屏蔽	—	盖
RXD 接收数据	白色/橙色	1
TXD 发送数据	橙色	2
DTR 数据终端就绪	白色/绿色	3
A	蓝色	4
B	白色/蓝色	5
DSR 数据设置就绪	绿色	6
NC 没有连接	白色/褐色	7
SG 信号接地	褐色	8

D-sub 25针连接器插针

插针编号	说明
1	FG 帧接地
2	TXD1 发送数据
3	RXD1 接收数据
7	GND 接地



注意

请勿将任何电线连接至 NC (没有连接) 针脚; 否则 FC6A 型和条形码读取器可能无法正常工作并受到损坏。

操作说明

条形码读取器用于扫描 8 位数条形码。已扫描的数据通过串行端口 1 (RS232C) 被发送至 FC6A 型, 然后存储至寄存器。数据的上 8 位存储到寄存器 D20, 下 8 位存储到寄存器 D21。

在 WindLDR 功能设置中设置用户通信

此示例使用串行端口 1 (RS232C), 请使用 WindLDR 在功能设置中选择端口 1 的用户协议。请参见第 5-25 页上的“设置 WindLDR”。

设置通信设置

设置通信参数与条形码读取器的参数匹配。请参见第 5-25 页上的“设置 WindLDR”。有关条形码读取器通信设置的详细信息, 请参阅条形码读取器用户手册。示例如下:

通信设置:

通信速度: 9,600bps
 数据长度: 7
 奇偶校验: 偶数
 停止位: 1

设置条形码读取器

如下所示值为设置条形码读取器示例。有关实际设置，请参阅条形码读取器的用户手册。

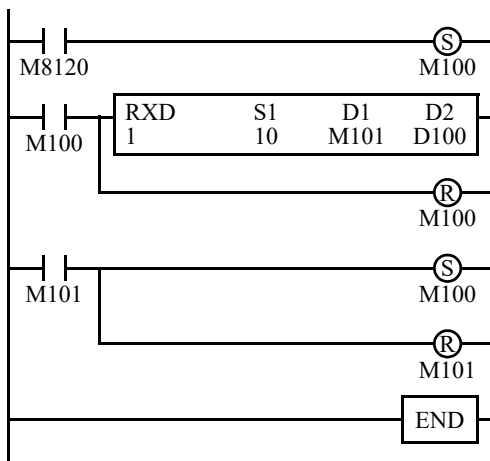
同步模式	自动			
读取模式	单读取或多读取			
通信设置	通信速度:	9,600bps	数据长度:	7
	奇偶校验:	偶数	停止位:	1
其他通信设置	页眉:	02h	结束符号:	03h
	数据回送:	否	BCR 数据输出:	是
	输出计时:	输出优先级 1	字符取消:	否
	数据输出过滤器:	否	主系列输入:	否
	子系列:	否		
比较预置模式	未使用			

设备地址

M100	输入以开始接收条形码数据
M101	条形码数据接收完成输出
M8120	初始化脉冲特殊内部继电器
D20	存储条形码数据 (高位 4 位数字)
D21	存储条形码数据 (低位 4 位数字)
D100	条形码数据接收状态寄存器
D101	接收数据字节计数寄存器

梯形图

当 FC6A 型开始操作时，将执行 RXD1 指令以等待输入数据。当数据接收完成时，将存储数据至寄存器 D20 和 D21。接收完成信号用于执行 RXD1 指令以等待另一个输入数据。



M8120 是初始化脉冲特殊内部继电器，用于设置 M100。

在 M100 的上升沿执行 RXD1 以准备接收数据。
即使 M100 已复位，RXD1 仍然等待输入数据。

数据接收完成时，M101 将打开，然后设置 M100 以执行 RXD1 来接收下一个输入数据。

RXD1 数据



5: 用户通信指令

BCC 计算示例

FC6A 型可以使用 ADD-2comp、Modbus ASCII 和 Modbus RTU 的三个新 BCC 计算公式发送指令 TXD1、TXD2 和 TXD3 以及接收指令 RXD1、RXD2 和 RXD3。按如下所示计算这些块校验字符。

ADD-2comp

增加从 BCC 当前地址至 BCC 前字节范围内的字符，然后逐位取反结果，并增加 1。

1. 在 BCC 当前地址至 BCC 之前紧接的字节之间增加字符。
2. 逐位切换结果，并增加 1（2 的求补）。
3. 根据指定转换类型（二进制数→ASCII 码转换或不转换）和指定 BCC 位数存储结果至 BCC 位置。

示例：二进制数→ASCII 码转换，2 位 BCC

当步骤 2 的结果为 175h 时，BCC 将包含 37h 和 35h。

Modbus ASCII — 计算 LRC（纵向冗余校验）

在 BCC 当前地址至 BCC 前字节范围内使用 LRC（纵向冗余校验）计算 BCC。

1. 在 BCC 当前地址至 BCC 之前紧接的字节之间转换 ASCII 字符（以两个字符为单位）生成 1 字节十六进制数据。
（示例：37h, 35h → 75h）
2. 增加步骤 1 的结果。
3. 逐位切换结果，并增加 1(2 的求补)。
4. 转换最低位的 1 字节数据为 ASCII 字符。（示例：75h → 37h, 35h）
5. 存储两位数至 BCC（LRC）位置。

如果 BCC 计算范围由字节个数为奇数，那么 BCC 计算产生不定值。Modbus 协议定义 BCC 计算范围为偶数字节数。

Modbus RTU — 计算 CRC-16（循环冗余校验和）

在 BCC 当前地址至 BCC 前字节范围内使用 CRC-16（循环冗余校验和）计算 BCC。

生成多项式为： $X^{16} + X^{15} + X^2 + 1$

1. 提取 FFFFh 异或 (XOR) 以及 BCC 当前地址的第一个 1 字节数据。
2. 逐位切换结果至右侧。当出现带进位时，提取 A001 的异或 (XOR)，然后进入步骤 3。否则，直接进入步骤 3。
3. 切换 8 次，重复步骤 2。
4. 提取结果的异或 (XOR) 以及下一个 1 字节数据。
5. 重复步骤 2 到步骤 4 直至 BCC 前紧接的字节。
6. 转换步骤 5 的结果的高位字节和低位字节，并将结果 CRC-16 存储至 BCC（CRC）位置。
（示例：1234h → 34h, 12h）

6: Modbus 通信

简介

本章介绍 FC6A 型的 Modbus 通信功能。

概述

Modbus 通信将 FC6A 型用作 Modbus RTU 通信的主机或从机、Modbus TCP 通信的客户端或服务器，并与外部设备进行通信。

Modbus 通信中使用的通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
端口 1	是	是	是	—	是 ^{*2}	是 ^{*2}
端口 2	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*2}	是 ^{*2}
端口 3	—	—	是 ^{*1}	是 ^{*1}	是 ^{*3}	是 ^{*3}
端口 4 至 33	是（端口 4 至 9） ^{*4}			—	是（端口 4 至 33） ^{*5}	
以太网端口 1	是	是	是	是	是	是
以太网端口 2	—	—	—	—	是	是
HMI- 以太网端口	—	—	—	—	—	—

*1 连接了 RS232C 通信盒或 RS485 通信盒时

*2 连接盒基本模块、连接了 RS232C 通信盒或 RS485 通信盒时

*3 连接 HMI 模块、连接了 RS232C 通信盒或 RS485 通信盒时

*4 All-in-One CPU 模块及 CAN J1939 All-in-One CPU 模块中最多可连接 3 台通信模块，以扩展端口 4 ~ 9。

*5 Plus CPU 模块中最多可连接 15 台通信模块，以扩展端口 4 ~ 33。

设置 Modbus 通信

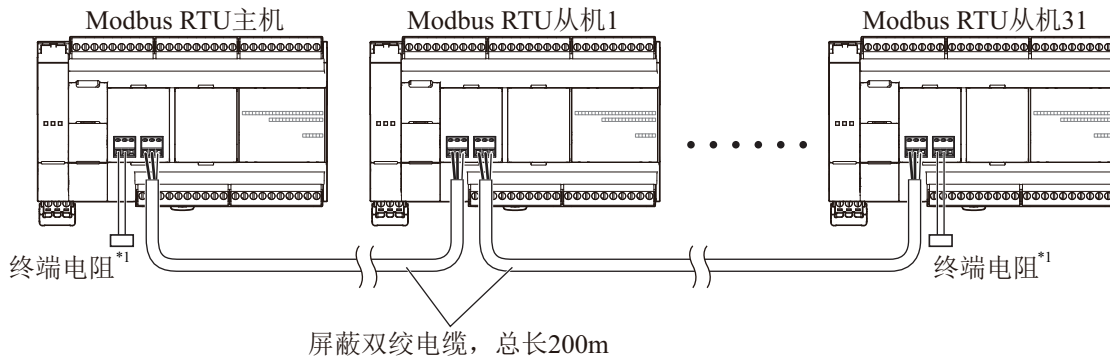
各通信端口支持的 Modbus 通信的设置如下所示。

- 使用端口 1 至 33，与外部设备进行串行通信（RS232C/RS485）
第 6-2 页上的“通过 RS232C/RS485 进行的 Modbus RTU 通信”
- 与通过以太网端口 1 及 2 连接的外部设备的以太网通信
第 6-21 页上的“通过以太网通信进行的 Modbus TCP 通信”

通过 RS232C/RS485 进行的 Modbus RTU 通信

FC6A 型支持 Modbus RTU 协议，可用作 Modbus RTU 主机和 Modbus RTU 从机。将 FC6A 型配置为 Modbus RTU 主机时，它可以监控和更改 Modbus RTU 从机设备的数据。将 FC6A 型配置为 Modbus RTU 从机时，它可以从 Modbus RTU 主机设备监控器和更改 FC6A 型的设备数据。

有关 Modbus RTU 主机功能及其设置方式，请参见第 6-3 页上的“Modbus RTU 主机通信”。有关 Modbus RTU 从机功能及其设置方式，请参见第 6-9 页上的“Modbus RTU 从机通信”。



*1 通信规范不稳定时，请在两端插入符合特性阻抗的终端电阻。请使用额定值 1/2W 以上的电阻。

Modbus RTU 主机通信

将 FC6A 型配置为 Modbus RTU 主机时，它可向 Modbus RTU 从机发送通信请求以读取 / 写入数据。各项通信请求会根据配置的请求列表发送到 Modbus 从机。

Modbus RTU 从机站的 Modbus RTU 主机通信设置和请求列表可使用 WinLDR“功能设置”进行设置。进行从机站通信与用户程序执行异步的，并且按照请求列表中指定的请求编号顺序在 END 处理中处理通信数据。当指定了请求执行内部继电器时，只有当相应的请求执行内部继电器打开时才执行请求。当没有指定请求执行内部继电器时，连续执行所有请求。

Modbus RTU 主机通信规格

项目	说明
通信速度 (bps)	9,600, 19,200, 38,400, 57,600, 115,200
数据长度	8 位 (固定)
停止位	1, 2 位
奇偶校验	无、奇数、偶数
从机编号	1 ~ 247 (0: 广播从机编号) *1
从机最大编号	通过 RS232C 进行的 Modbus RTU 通信: 1 台 通过 RS485 进行的 Modbus RTU 通信: 31 台
电缆总延长	通过 RS232C 进行的 Modbus 通信 FC6A-PC1: 5m FC6A-SIF52: 15m 通过 RS485 进行的 Modbus 通信 FC6A-PC3: 200m FC6A-SIF52: 1,200m
接收超时 *2	10 ~ 2,550ms (以 10ms 为增量)
字符间的超时	10ms
传输等待时间	0 ~ 5,000ms (以 1ms 为增量)
重试次数	0 ~ 10

*1 指定从机编号 0 时，通信请求将成为广播。所有 Modbus RTU 从机都会收到广播通信请求。Modbus RTU 从机不会应答广播通信。广播用来向所有 Modbus RTU 从机写入相同的数据。

*2 在接收从机的一个响应帧之前，指定时间。

Modbus RTU 主机通信启动和停止

在 Modbus RTU 主机请求列表中指定了请求执行设备时，系统会分配与请求数量一样多的内部继电器或数据寄存器的位来执行 Modbus RTU 主机通信。内部继电器或数据寄存器的位按照请求的顺序进行分配。例如，当内部继电器 M0 指定为请求执行内部继电器时，M0 分配到 1 号请求，M1 分配到 2 号请求，依次类推。要执行一个请求，打开相应的请求执行内部继电器。

当通信完成时，请求执行内部继电器自动关闭。当需连续发送请求时，使用 SET 或 OUT 指令保持相应的请求执行内部继电器打开。

当没有指定请求执行内部继电器时，连续执行请求列表中设置的所有请求。

通信完成和通信错误

当读取或写入过程成功完成后或发生通信错误时，Modbus RTU 通信结束。当重复出现通信失败的次数超过指定的重试次数，或者主机站在指定的接收超时期间没有收到响应时，系统就会发生通信错误。发生通信错误时，系统将取消该请求并处理下一个请求。指定错误状态数据寄存器后，可以确认每项请求的通信状态。

注释： Modbus 主机每次扫描可处理最多一个 Modbus 请求。

6: MODBUS通信

通信错误数据

通过“功能设置”在“请求列表”中配置“错误状态”后，可以确认每项请求的错误数据。

所有通信请求使用单个数据寄存器	各通信请求的错误数据
未确认	可以确认整个请求列表中各项请求的错误数据、远程主机号（高位字节）和错误代码（低位字节）。根据请求数量保留数据寄存器以存储错误数据。当发生请求错误时，错误数据将存储到相应的数据寄存器中。
已确认	所有请求共享单个数据寄存器。当发生请求错误时，错误数据将存储到数据寄存器中并将覆盖旧错误数据。
位分配	
远程主机号 (高位字节)	1 ~ 255
错误代码 (低位字节)	00h: 正常完成 01h: 功能代码错误（不支持的功能代码） 02h: 访问目标错误（地址超出范围，地址 + 设备量超出范围） 03h: 设备数错误，1位写入数据错误（不支持1位写入的指定设备数量） 12h: 帧长度错误（所发送请求的帧长度超出范围） 13h: BCC 错误（BCC 不匹配） 14h: 从机编号错误（收到的从机编号无效） 16h: 超时错误（超时）

各个请求的通信错误数据

可确认整个请求列表中每个请求的错误数据。要确认各个请求的错误数据，从“功能设置”中选择使用“请求列表”中的“错误状态”，并输入第一个数据寄存器编号。当未选择所有通信请求均“使用单个数据记录器”时，从数据寄存器编号开始，将保留与请求数量一样多的数据寄存器用于保存错误数据。当一个请求发生错误时，错误代码存储到相应的数据寄存器。

当选择了使用一个数据记录器来保存所有通信请求时，所有请求将共用同一个数据寄存器。当请求出现错误时，数据寄存器中将保存错误代码，原值将被覆盖。

Modbus RTU 主机上的请求数

下表显示了请求列表中可编程的请求数量：

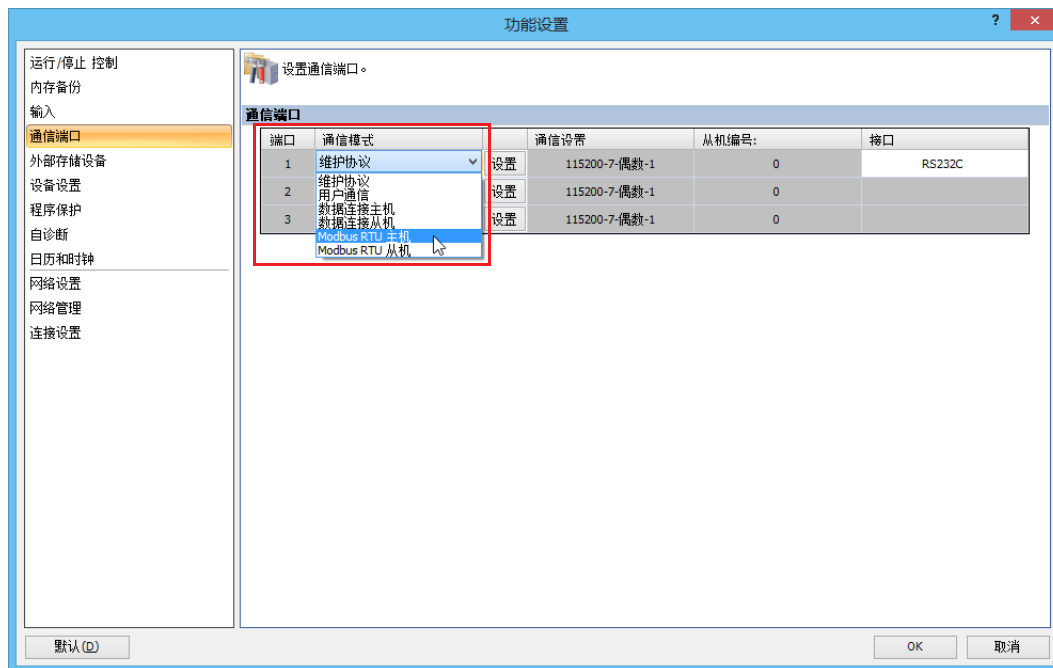
端口	端口 1 ~ 端口 33
请求的编号	255

注释：每个请求均需要 8 字节的用户程序区。

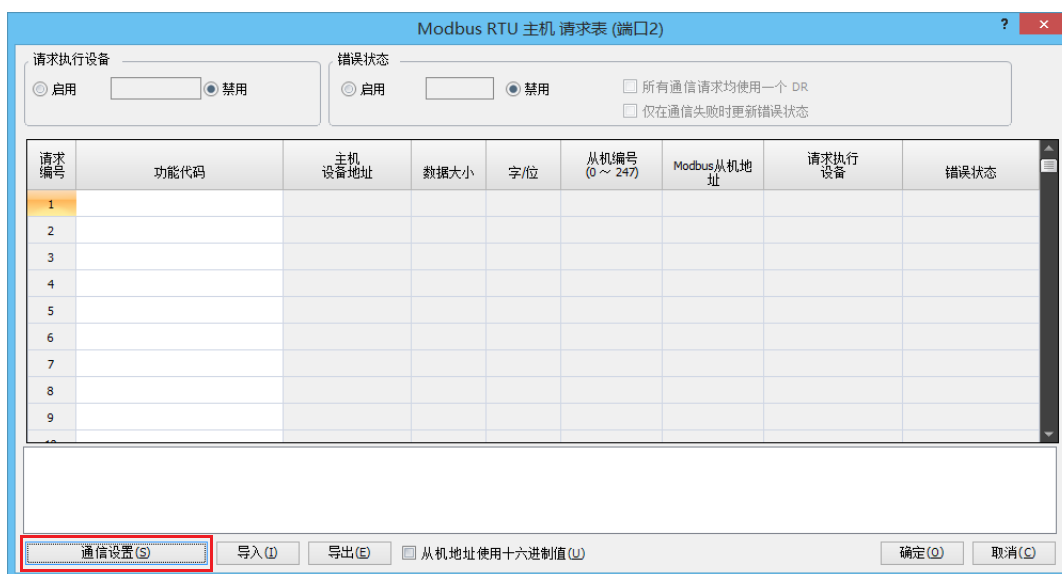
使用 WindLDR 设置 Modbus RTU 主机

使用 WindLDR 设置 Modbus 主机通信用于 Modbus RTU 模式。由于这些设置与用户程序相关，所以必须在更改设置后将用户程序下载到 FC6A 型。

1. 在 WindLDR 菜单栏中选择“设置”>“通信端口”。
此时出现通信端口的“功能设置”对话框。
2. 单击所使用端口的“通信模式”，选择“Modbus RTU 主机”。



3. 单击端口 2 的“设置”按钮。显示“Modbus RTU 主机请求列表”对话框。



6: MODBUS通信

4. 单击“通信设置”按钮。出现“通信设置”对话框。若有必要，请更改设置。

通信速度 (bps)	9600, 19200, 38400, 57600, 115200
奇偶校验	无、奇数、偶数
停止位	1 或 2
重试次数	0 ~ 10
接收超时	1 ~ 255 (×10ms)
传输等待时间	0 ~ 5000 (ms)

5. 单击“确定”按钮返回 Modbus RTU 主机请求列表。在“功能代码”下指定请求。一个请求列表中最多可输入 255 个请求。

如有必要，请选择使用“请求执行设备”和“错误状态”数据寄存器。当使用“请求执行设备”和“错误状态”数据寄存器时，输入设备的第一个编号。

编辑请求列表的注释

请求执行内部继电器和错误状态数据寄存器按照请求编号顺序分配。当删除一个请求或更改请求的顺序时，更改了请求与请求执行内部继电器和错误状态数据寄存器的关系。如果在用户程序中使用了内部继电器或数据寄存器，必须相应地更改设备编号。在完成更改后，重新下载用户程序。

6. 当完成编辑“主机请求列表”时，单击“确认”按钮保存更改。

7. 将用户程序下载到 FC6A 型。

现在，设置 Modbus 主机完成。有关参数和有效值的详细信息如下。

功能代码

FC6A 型的 Modbus RTU 支持下表列出的八个功能代码。受支持的功能代码和有效的从机地址因要进行通信的各台 Modbus 从机设备而异。请根据 Modbus 从机设备的规格配置功能代码。

功能代码	数据大小	从机地址	作为 Modbus 从机的 FC6A 型
01 读取线圈状态	1 ~ 2,000 位	000001 - 065535	读取 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的位设备状态。
02 读取输入状态	1 ~ 2,000 位	100001 - 165535	读取 I (输入)、T (定时器接点) 或 C (计数器接点) 的位设备状态。
03 读取保持寄存器	1 ~ 125 字	400001 - 465535	读取 D (数据寄存器)、T (定时器预置值) 或 C (计数器预置值) 的字设备数据。
04 读取输入寄存器	1 ~ 125 字	300001 - 365535	读取 T (定时器当前值) 或 C (计数器当前值) 的字设备数据。
05 强制单线圈	1 位	000001 - 065535	更改 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的位设备状态。
06 预置单寄存器	1 字	400001 - 465535	更改 D (数据寄存器) 的字设备数据。
15 强制多线圈	1 ~ 1,968 位	000001 - 065535	更改 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的多位设备状态。
16 预置多位数据寄存器	1 ~ 123 字	400001 - 465535	更改 D (数据寄存器) 的多字设备。

主机设备地址

当选择功能代码 01、02、03 或 04 从 Modbus 从机读取数据时，指定第一个数据寄存器或内部继电器编号存储从 Modbus 从机接收的数据。当选择功能代码 05、06、15 或 16 向 Modbus 从机写入数据时，指定第一个数据寄存器或内部继电器编号存储写入 Modbus 从机的数据。可按照主机设备地址指定数据寄存器和内部继电器。

数据大小和字 / 位

指定要读取或写入的数据量。有效数据大小取决于功能代码。当选择了功能代码 01、02、05 或 15 时，以位指定数据大小。当选择了功能代码 03、04、06 或 16 时，以字指定数据大小。有关有效数据大小，请参阅上表。

从机编号

指定从机编号 0 ~ 247，相同的从机编号可重复指定给不同的请求编号，可以是 1 ~ 255。在 Modbus 通信中，从机编号 0 用于广播从机编号。广播用来向所有 Modbus 从机写入相同的数据。

从机地址

指定 Modbus 从机的数据内存地址。有效从机地址范围取决于功能代码。有关有效从机地址，请参阅上表。内存地址的分配因各台 Modbus 从机设备而异。请参阅各台 Modbus 从机设备的手册。

请求执行设备

要使用请求执行设备，在 Modbus RTU 主机请求列表中单击“使用”无线按钮并指定第一个内部继电器。在表中自动列出用于执行的内部继电器。要执行一个请求，打开相应的请求执行内部继电器。

数据寄存器也可以指定为请求执行设备。当第一个数据寄存器被指定为请求执行设备时，数据寄存器的位与从第一个数据寄存器的最低位分配的请求数量一样多。数据寄存器的位将被分配为在请求列表中自动一览显示的执行继电器。

当没有指定请求执行设备时，连续执行请求列表中设置的所有请求。

错误状态数据寄存器

要使用错误状态数据寄存器，在 Modbus RTU 主机请求列表中单击“使用”无线按钮并指定第一个数据寄存器。在表中自动列出用于存储错误状态的数据寄存器。当“所有通信请求使用一个数据寄存器”被选定时，所有请求将共用第一个数据寄存器。

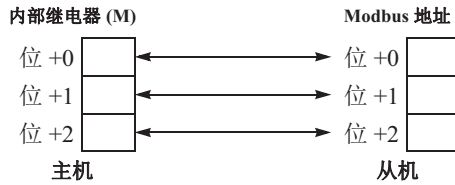
6: MODBUS通信

处理请求

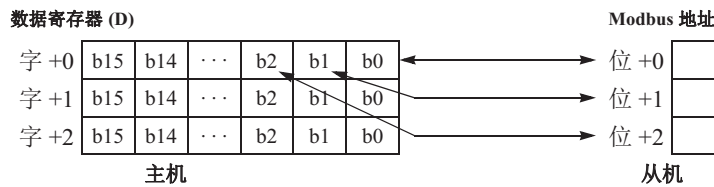
如下所示，Modbus 通信数据在主机和从机之间处理。

从机位数据（功能代码 01、02、05 和 15）

- 主机设备地址：内部继电器

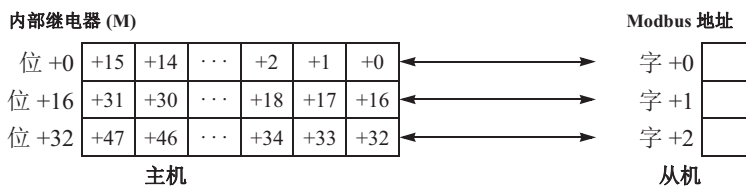


- 主机设备地址：数据寄存器

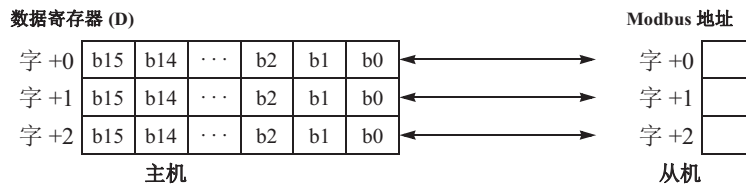


从机字数据（功能代码 03、04、06 和 16）

- 主机设备地址：内部继电器



- 主机设备地址：数据寄存器



Modbus RTU 从机通信

通过在 WindLDR 的“功能设置”端口 1、端口 2 和端口 3 选择“Modbus RTU 从机”，可配置 Modbus RTU 从机通信。当 Modbus RTU 从机从 Modbus RTU 主机接收请求时，Modbus RTU 从机根据请求读取或写入数据。在用户程序的 END 处理时处理请求。

Modbus RTU 从机不会应答 Modbus RTU 主机的广播请求。

Modbus RTU 从机通信规格

项目	说明	
通信速度 (bps)	9,600, 19,200, 38,400, 57,600, 115,200	
数据长度	8 位 (固定)	
停止位	1, 2 位	
奇偶校验	无、奇数、偶数	
从机编号	常量	1 ~ 247
	数据寄存器	在 1 ~ 247 的范围内设置特殊数据寄存器的值 端口 1: D8100 端口 2: D8102 端口 3: D8103 端口 4 至 9 ^{*2} : D8040 至 D8045 端口 10 至 33 ^{*2*3} : D8735 至 D8758
字符间的超时 ^{*1}	1.5 字符以上 ^{*4}	
帧间的超时 ^{*1}	3.5 字符以上 ^{*5}	

*1 当发生超时，FC6A 型丢弃接收的数据并等待下一个有效通信的第一个帧。

*2 使用通信模块时

*3 仅 Plus CPU 模块

*4 当通信速度设为 19,200bps 或更高速时，字符之间的超时至少要达到 0.75ms。

*5 进行 119,200bps 或更高速度的通信时，帧与帧之间的超时最少必须在 1.75ms。

Modbus RTU 从机的从机地址变换

Modbus 设备名称	Modbus 地址变换 (十进制) ^{*1}	通信帧地址 ^{*2}	FC6A 型设备 ^{*3}	适用功能代码
线圈 (000000 以上)	000001 - 000516	0000 - 0203	Q0 - Q643	1, 5, 15
	000701 - 000956	02BC - 03BB	R000 - R255	
	001001 - 003048	03E8 - 0BE7	M0000 - M2557	
	003049 - 007400	0BE8 - 1CE7	M2560 - M7997	
	009001 - 009256	2328 - 2427	M8000 - M8317	
	009257 - 010600	2428 - 2967	M8320 - M9997	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
	017001 - 020000	4268 - 4E1F	M17500 - M21247	
	030001 - 030480	7530 - 770F	Q1000 - Q1597	
	030481 - 030960	7710 - 78EF	Q2000 - Q2597	
	030961 - 031440	78F0 - 7ACF	Q3000 - Q3597	
	031441 - 031920	7AD0 - 7CAF	Q4000 - Q4597	
	031921 - 032400	7CB0 - 7E8F	Q5000 - Q5597	
	032401 - 032880	7E90 - 806F	Q6000 - Q6597	
	032881 - 033360	8070 - 824F	Q7000 - Q7597	
	033361 - 033840	8250 - 842F	Q8000 - Q8597	
033841 - 034320	8430 - 860F	Q9000 - Q9597		
034321 - 034800	8610 - 87EF	Q10000 - Q10597		

6: MODBUS通信

Modbus 设备名称	Modbus 地址变换 (十进制) *1	通信帧地址 *2	FC6A 型设备 *3	适用功能代码
输入继电器 (100000 以上)	100001 - 100516	0000 - 0203	I0 - I643	2
	101001 - 101256	03E8 - 04E7	T000 - T255 (定时器接点)	
	101501 - 101756	05DC - 06DB	C000 - C255 (计数器接点)	
	102001 - 102768	07D0 - 0ACF	T256 - T1023 (定时器接点)	
	102769 - 103744	0AD0 - 0E9F	T1024 - T1999 (定时器接点)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (计数器接点)	
	130001 - 130480	7530 - 770F	I1000 - I1597	
	130481 - 130960	7710 - 78EF	I2000 - I2597	
	130961 - 131440	78F0 - 7ACF	I3000 - I3597	
	131441 - 131920	7AD0 - 7CAF	I4000 - I4597	
	131921 - 132400	7CB0 - 7E8F	I5000 - I5597	
	132401 - 132880	7E90 - 806F	I6000 - I6597	
	132881 - 133360	8070 - 824F	I7000 - I7597	
	133361 - 133840	8250 - 842F	I8000 - I8597	
133841 - 134320	8430 - 860F	I9000 - I9597		
134321 - 134800	8610 - 87EF	I10000 - I10597		
输入寄存器 (300000 以上)	300001 - 300256	0000 - 00FF	T000 - T255 (定时器当前值)	4
	300501 - 300756	01F4 - 02F3	C000 - C255 (计数器当前值)	
	302001 - 302768	07D0 - 0ACF	T256 - T1023 (定时器当前值)	
	302769 - 303744	0AD0 - 0E9F	T1024 - T1999 (定时器当前值)	
	304001 - 304256	0FA0 - 109F	C256 - C511 (计数器当前值)	
保持寄存器 (400000 以上)	400001 - 408000	0000 - 1F3F	D0000 - D7999	3, 6, 16
	408001 - 408500	1F40 - 2133	D8000 - D8499	
	408501 - 408900	2134 - 22C3	D8500 - D8899	
	409001 - 409256	2328 - 2427	T000 - T255 (定时器预置值)	3
	409501 - 409756	251C - 261B	C000 - C255 (计数器预置值)	
	410001 - 456000	2710 - DABF	D10000 - D55999	3, 6, 16
	456001 - 462000	DAC0 - F22F	D56000 - D61999	
	462001 - 462768	F230 - F52F	T256 - T1023 (定时器预置值)	3
	462769 - 463744	F530 - F8FF	T1024 - T1999 (定时器预置值)	
464001 - 464256	FA00 - FAFF	C256 - C511 (计数器预置值)		

*1 地址通常用于 Modbus 通信。在第 6-11 页上的“计算 FC6A 型设备的 Modbus 地址”中，通过 FC6A 型的设备显示计算从机地址的方法。

*2 这些 4 位数地址用于通信帧。要计算通信帧中使用的地址，提取 Modbus 地址的低 5 位数，从该值中减去 1，并将结果转换为十六进制。

*3 在所用 FC6A 型类型的设备范围内进行访问。

计算 FC6A 型设备的 Modbus 地址

FC6A 型设备		计算 Modbus 地址	计算示例
I, Q, M	M <u>XXXX</u> X ②: 八进制 ①: 十进制	$(① - ④) \times 8 + ② + ⑤$ ④: 最小地址 ⑤: 偏移量	示例: M325 $(32 - 0) \times 8 + 5 + 1001 = 1262$ Modbus 地址: 1262 $1262 - 1 = 1261 = 04ED$ 通信从机地址: 04ED
R, T, C, D	D <u>XXXXX</u> ③: 十进制	$(③ - ④) + ⑤$ ④: 最小地址 ⑤: 偏移量	示例: D756 $(756 - 0) + 400001 = 400757$ Modbus 地址: 400757 提取低 5 位数 → 757 $757 - 1 = 756 = 02F4$ 通信从机地址: 02F4

Modbus 设备名称	FC6A 型设备	最小地址 ④	偏移量 ⑤
线圈	Q0 - Q643	0	1
	R000 - R255	0	701
	M0000 - M2557	0	1001
	M2560 - M7997	256	3049
	M8000 - M8317	800	9001
	M8320 - M9997	832	9257
	M10000 - M17497	1000	11001
	M17500 - M21247	1750	17001
	Q1000 - Q1597	100	30001
	Q2000 - Q2597	200	30481
	Q3000 - Q3597	300	30961
	Q4000 - Q4597	400	31441
	Q5000 - Q5597	500	31921
	Q6000 - Q6597	600	32401
	Q7000 - Q7597	700	32881
	Q8000 - Q8597	800	33361
	Q9000 - Q9597	900	33841
Q10000 - Q10597	1000	34321	
输入继电器	I0 - I643	0	100001
	T000 - T255 (定时器接点)	0	101001
	C000 - C255 (计数器接点)	0	101501
	T256 - T1023 (定时器接点)	256	102001
	T1024 - T1999 (定时器接点)	1024	102769
	C256 - C511 (计数器接点)	256	104001
	I1000 - I1597	100	130001
	I2000 - I2597	200	130481
	I3000 - I3597	300	130961
	I4000 - I4597	400	131441
	I5000 - I5597	500	131921
	I6000 - I6597	600	132401
	I7000 - I7597	700	132881
	I8000 - I8597	800	133361
I9000 - I9597	900	133841	
I10000 - I10597	1000	134321	

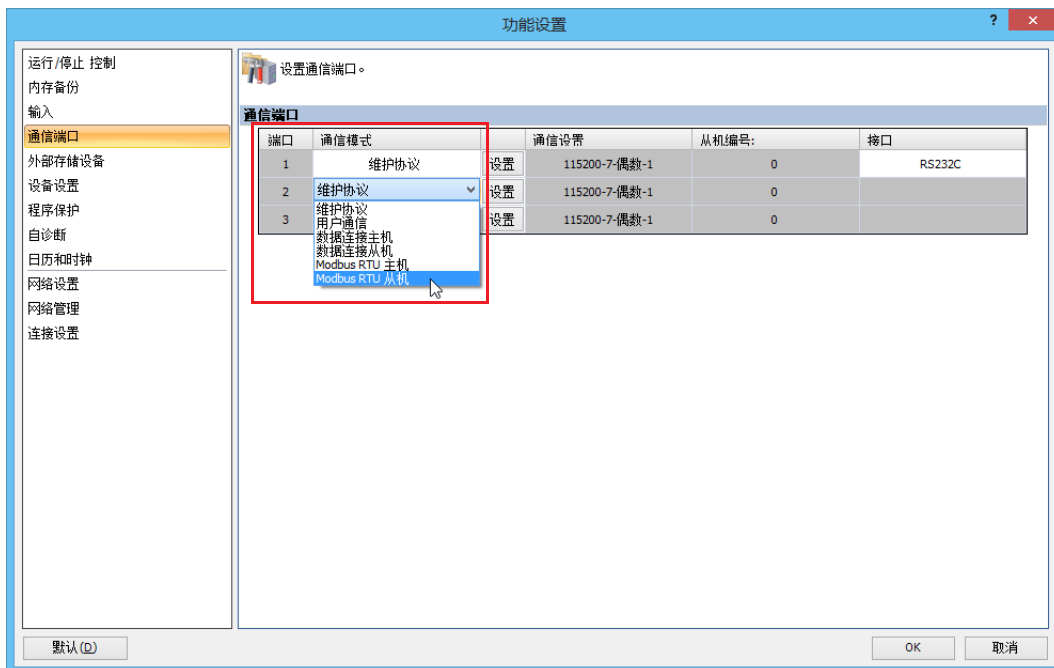
6: MODBUS通信

Modbus 设备名称	FC6A 型设备	最小地址 ④	偏移量⑤
输入寄存器	T000 - T255 (定时器当前值)	0	300001
	C000 - C255 (计数器当前值)	0	300501
	T256 - T1023 (定时器当前值)	256	302001
	T1024 - T1999 (定时器当前值)	1024	302769
	C256 - C511 (计数器当前值)	256	304001
保持寄存器	D0000 - D7999	0	400001
	D8000 - D8499	8000	408001
	D8500 - D8899	8500	408501
	T000 - T255 (定时器预置值)	0	409001
	C000 - C255 (计数器预置值)	0	409501
	D10000 - D55999	10000	410001
	D56000 - D61999	56000	456001
	T256 - T1023 (定时器预置值)	256	462001
	T1024 - T1999 (定时器预置值)	256	462769
	C256 - C511 (计数器预置值)	256	464001

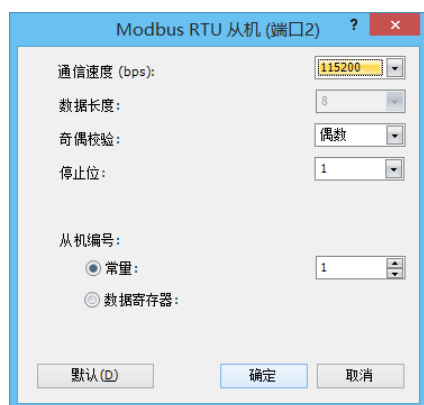
使用 WindLDR 设置 Modbus 从机

使用 WindLDR 设置 Modbus 从机通信用于 Modbus RTU。由于这些设置与用户程序相关，所以必须在更改设置后将用户程序下载到 FC6A 型。

1. 在 WindLDR 菜单栏中选择“设置”>“功能设置”>“通信端口”。
此时出现通信端口的“功能设置”对话框。
2. 单击所使用端口的“通信模式”，选择“Modbus RTU 从机”。



3. 单击“设置”按钮。出现“Modbus RTU 从机”对话框。若有必要，请更改设置。



通信速度 (bps)	9600 19200 38400 57600 115200	
数据长度	8	
奇偶校验	无、奇数、偶数	
停止位	1 或 2	
从机编号	常量	1 ~ 247
	数据寄存器	使用 D8100 (端口 1)、D8102 (端口 2)、D8103 (端口 3)、D8040 至 D8045 (端口 4 至 9 ^{*1})、或 D8735 至 D8758 (端口 10 至 33 ^{*1*2}) 的值。

*1 使用通信模块时

*2 仅 Plus CPU 模块

4. 单击“确定”按钮保存更改。
5. 将用户程序下载到 FC6A 型。
现在，设置 Modbus 从机完成。有关参数和有效值的详细信息如下。

6: MODBUS通信

通信协议

这一节描述用于 Modbus RTU 通信的通信帧格式。

RTU 模式的通信格式

Modbus RTU 主机请求

空闲 3.5 字符	从机编号	功能代码	数据	CRC	空闲 3.5 字符
	1 字节	1 字节		2 字节	

Modbus RTU 从机的 ACK 应答

空闲 3.5 字符	从机编号	功能代码	数据	CRC	空闲 3.5 字符
	1 字节	1 字节		2 字节	

Modbus RTU 从机的 NAK 应答

空闲 3.5 字符	从机编号	功能代码 + 80H	错误代码	CRC	空闲 3.5 字符
	1 字节	1 字节	1 字节	2 字节	

注释：空闲指通信线上没有数据通过。

通信帧格式

Modbus RTU 模式需要帧之间最小 3.5 字符长的空闲时间确定一个帧的开始。FC6A 型 Modbus 主机在 5ms 的空闲间隔发送请求，您可以在“功能设置”对话框中更改该设置。

从机编号

FC6A 型可指定从机编号 1 ~ 247，在使用 RS232C 的 1: 1 通信中，必须在主机和 FC6A 型中设置相同的从机编号。

从机编号 0 保留用于广播从机编号，并用于向所有 Modbus RTU 从机写入相同的数据。在这种情况下，Modbus RTU 从机不会向主机发送应答。

Modbus RTU 通信 NG 应答错误代码

NAK 应答中将存储下列错误代码之一。

- 01h: 功能代码错误（不支持的功能代码）
- 02h: 访问目标错误（地址超出范围，地址 + 设备量超出范围）
- 03h: 设备量错误，1 位写入数据错误

CRC

Modbus RTU 模式使用 CRC 校验代码。

• Modbus RTU 模式 — 计算 CRC-16（循环冗余校验和）

在从机编号至 BCC 之前紧接的字节范围内，使用 CRC-16 计算 BCC。生成多项式为： $X^{16} + X^{15} + X^2 + 1$ 。

1. 提取 FFFFh 异或（XOR）以及从机编号的第一个 1 字节数据。
2. 将结果右移 1 位。当出现带进位时，提取 A001 的异或（XOR），然后进入步骤 3。否则，直接进入步骤 3。
3. 切换 8 次，重复步骤 2。
4. 提取结果的异或（XOR）以及下一个 1 字节数据。
5. 重复步骤 2 到步骤 4 直至 BCC 前紧接的字节。
6. 转换步骤 5 的结果的高位字节和低位字节，并将结果 CRC-16 存储至 BCC（CRC）位置。（示例：1234h → 34h, 12h）

通信格式

这一节描述从机编号到校验代码之前各个功能代码的通信格式。

功能代码 01（读取线圈状态）和功能代码 02（读取输入状态）

功能代码 01 读取 Q（输出）、R（移位寄存器）或 M（内部继电器）的位设备状态。可读出 1 ~ 2,000 连续位。

功能代码 02 读取 I（输入）、T（定时器接点）或 C（计数器接点）的位设备状态。可读出 1 ~ 2,000 连续位。

通信帧

Modbus RTU 主机请求

从机编号	功能代码	地址	位数
xxh	01h / 02h	xxxxh	xxxxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	数据量	第一个 8 位	第二个 8 位	最后一个 8 位
xxh	01h / 02h	xxh	xxh	xxh	xxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	81h / 82h	xxh

通信示例

用途	读取输出 Q10 开始的 15 位。 $Q10 \rightarrow (1 - 0) \times 8 + 0 + 1 = 9$ Modbus 地址: 9 $9 - 1 = 8 = 8h$ 通信帧地址: 0008h
条件	从机 8 号 Q10 ~ Q27 二进制数据: 1234h

• Modbus RTU 模式

Modbus RTU 主机请求	08 01 0008 0010 (CRC)
Modbus RTU 从机的 ACK 应答	08 01 02 34 12 (CRC)
Modbus RTU 从机的 NAK 应答	08 81 xx (CRC)

6: MODBUS通信

功能代码 03（读取保持寄存器）和功能代码 04（读取输入寄存器）

功能代码 03 读取 D（数据寄存器）、T（定时器预置值）或 C（计数器预置值）的字设备数据。可读出 1 ~ 125 连续位。

功能代码 04 读取 T（定时器当前值）或 C（计数器当前值）的字设备数据。可读出 1 ~ 125 连续位。

通信帧

Modbus RTU 主机请求

从机编号	功能代码	地址	字数
xxh	03h / 04h	xxxxh	xxxxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	数据量	第一个高位字节	第一个低位字节	最后一个低位字节
xxh	03h / 04h	xxh	xxh	xxh	xxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	83h / 84h	xxh

通信示例

用途	读取从数据寄存器 D1710 开始的 2 个字。 $D1710 \rightarrow (1710 - 0) + 400001 = 401711$ Modbus 地址: 401711 提取低 5 位数 $\rightarrow 1711$ $1711 - 1 = 1710 = 6AEh$ 通信帧地址: 06AEh
条件	从机 8 号 D1710 数据: 1234h D1711 数据: 5678h

• Modbus RTU 模式

Modbus RTU 主机请求	08 03 06AE 0002 (CRC)
Modbus RTU 从机的 ACK 应答	08 03 04 12 34 56 78 (CRC)
Modbus RTU 从机的 NAK 应答	08 83 xx (CRC)

功能代码 05（强制单线圈）

功能代码 05 更改 Q（输出）、R（移位寄存器）或 M（内部继电器）的位设备状态。

通信帧**Modbus RTU 主机请求**

从机编号	功能代码	地址	关：0000h 开：FF00h
xxh	05h	xxxxh	xxxxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	地址	关：0000h 开：FF00h
xxh	05h	xxxxh	xxxxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	85h	xxh

通信示例

用途	强制内部继电器 M1320 打开。 M1320 → (132 - 0) × 8 + 0 + 1001 = 2057 Modbus 地址：2057 2057 - 1 = 2056 = 808h 通信帧地址：0808h
条件	从机 8 号

• **Modbus RTU 模式**

Modbus RTU 主机请求	08 05 0808 FF00 (CRC)
Modbus RTU 从机的 ACK 应答	08 05 0808 FF00 (CRC)
Modbus RTU 从机的 NAK 应答	08 85 xx (CRC)

6: MODBUS通信

功能代码 06（强制单寄存器）

功能代码 06 更改 D（数据寄存器）的字设备数据。

通信帧

Modbus RTU 主机请求

从机编号	功能代码	地址	新数据
xxh	06h	xxxxh	xxxxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	地址	确认数据
xxh	06h	xxxxh	xxxxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	86h	xxh

通信示例

用途	将 8000 写入数据寄存器 D1708。 $D1708 \rightarrow (1708 - 0) + 400001 = 401709$ Modbus 地址: 401709 提取低 5 位数 $\rightarrow 1709$ $1709 - 1 = 1708 = 6ACh$ 通信帧地址: 06ACh
条件	从机 8 号

• Modbus RTU 模式

Modbus RTU 主机请求	08 06 06AC 1F40 (CRC)
Modbus RTU 从机的 ACK 应答	08 06 06AC 1F40 (CRC)
Modbus RTU 从机的 NAK 应答	08 86 xx (CRC)

功能代码 15（强制多线圈）

功能代码 15 更改 Q（输出）、R（移位寄存器）或 M（内部继电器）的位设备状态。可更改 1 ~ 1,968 连续位。

通信帧

Modbus RTU 主机请求

从机编号	功能代码	地址	位数	数据量	第一个 8 位	第二个 8 位	最后一个 8 位
xxh	0Fh	xxxxh	xxxxh	xxh	xxh	xxh	xxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	地址	位数
xxh	0Fh	xxxxh	xxxxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	8Fh	xxh

通信示例

用途	将下列位状态写入内部继电器 M605 ~ M624。							
	M610 (开)	M611 (关)	M612 (开)	M613 (开)	M614 (关)	M605 (开)	M606 (开)	M607 (开)
条件	M620 (关)	M621 (关)	M622 (关)	M623 (关)	M624 (关)	M615 (关)	M616 (开)	M617 (关)
	M605 (LSB) ~ M614 (MSB) 二进制数据: 6B M615 (LSB) ~ M624 (MSB) 二进制数据: 02 $M605 \rightarrow (60 - 0) \times 8 + 5 + 1001 = 1486$ Modbus 地址: 1486 $1486 - 1 = 1485 = 5CDh$ 通信帧地址: 05CDh							

• Modbus RTU 模式

Modbus RTU 主机请求	08 0F 05CD 0010 02 6B 02 (CRC)
Modbus RTU 从机的 ACK 应答	08 0F 05CD 0010 (CRC)
Modbus RTU 从机的 NAK 应答	08 8F xx (CRC)

6: MODBUS通信

功能代码 16（预置多寄存器）

功能代码 16 更改 D（数据寄存器）的字设备数据。可更改 1 ~ 123 连续字。

通信帧

Modbus RTU 主机请求

从机编号	功能代码	地址	字数	数据量	第一个高位字节	第一个低位字节	最后一个低位字节
xxh	10h	xxxxh	xxxxh	xxh	xxh	xxh	xxh

Modbus RTU 从机的 ACK 应答

从机编号	功能代码	地址	字数
xxh	10h	xxxxh	xxxxh

Modbus RTU 从机的 NAK 应答

从机编号	功能代码	错误代码
xxh	90h	xxh

通信示例

用途	<p>将下列数据写入四个数据寄存器 D1708 ~ D1711。</p> <p>D1708 D1709 D1710 D1711 (1234h) (5678h) (ABCDh) (EF01h)</p> <p>D1708 → (1708 - 0) + 400001 = 401709 Modbus 地址: 401709 提取低 5 位数 → 1709 1709 - 1 = 1708 = 6ACh 通信帧地址: 06ACh</p>
条件	从机 8 号

• Modbus RTU 模式

Modbus RTU 主机请求	08 10 06AC 0004 08 12 34 56 78 AB CD EF 01 (CRC)
Modbus RTU 从机的 ACK 应答	08 10 06AC 0004 (CRC)
Modbus RTU 从机的 NAK 应答	08 90 xx (CRC)

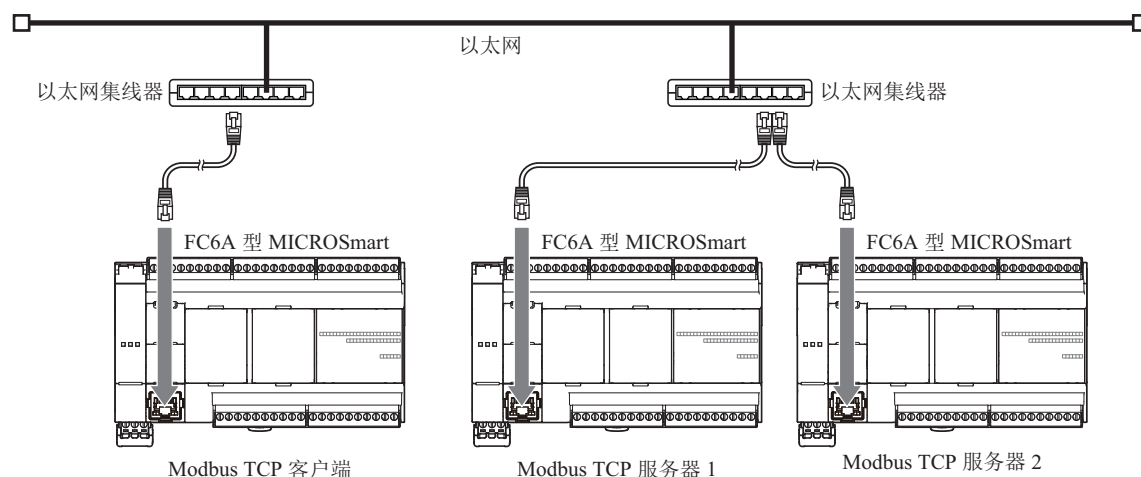
通过以太网通信进行的 Modbus TCP 通信

FC6A 型支持 Modbus TCP 客户端和 Modbus TCP 服务器。在 FC6A 型上连接以太网端口 1、以太网端口 2（仅 Plus CPU 模块），以使 FC6A 型能够与符合 Modbus TCP 规格的设备进行通信。

将 FC6A 型配置为 Modbus TCP 客户端时，它可以监控和更改支持 Modbus TCP 服务器的网络设备的数据内存。可将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到 Modbus TCP 客户端。每个连接可以与各自多台（1 ~ 255 台）的 Modbus TCP 服务器设备进行通信。

将 FC6A 型配置为 Modbus TCP 服务器时，它可以从 Modbus TCP 客户端设备监控器和更改 FC6A 型的设备数据。可将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块最多 8 个连接、Plus CPU 模块最多 16 个连接分配到 Modbus TCP 服务器。

有关 Modbus TCP 客户端功能及其设置方式，请参见第 6-22 页上的“Modbus TCP 客户端”。有关 Modbus TCP 服务器功能及其设置方式，请参见第 6-27 页上的“Modbus TCP 服务器”。



6: MODBUS通信

Modbus TCP 客户端

将 FC6A 型配置为 Modbus TCP 客户端时，它可向 Modbus TCP 服务器发送通信请求以读取 / 写入数据。各项通信请求会根据配置的请求列表发送到 Modbus 服务器。

可以使用 WindLDR“功能设置”编制 Modbus TCP 客户端通信设置和 Modbus TCP 服务器请求列表。FC6A 型根据这些设置与 Modbus TCP 服务器进行通信。

Modbus TCP 服务器的通信与用户程序执行同步进行，按照在请求列表中指定的请求编号的顺序以 END 处理方式处理通信数据。在 Plus CPU 模块的 Modbus TCP 客户端通信中，最初的 128 请求不会切断与 Modbus TCP 服务器的连接，进行 Modbus TCP 通信。129 请求以后会在每次 Modbus TCP 通信时进行与 Modbus TCP 服务器连接的连接与切断。

Modbus TCP 客户端规格

参数	Modbus TCP 客户端
从机编号	0 ~ 255
服务器最大数量	8 (每个连接一台服务器)
接收超时 *1	100 ~ 25,500ms (以 100ms 为递增单位)
传输等待时间	0 ~ 5,000ms (0ms)

*1 指定接收服务器响应帧之前的时段。

请求执行设备	开始、停止
启用	在 Modbus TCP 客户端请求列表中指定了请求执行设备时，系统会分配内部继电器或数据寄存器位 (与请求数量一样多) 来执行 Modbus TCP 客户端通信。例如，当将内部继电器 M0 指定为请求执行设备时，M0 被分配为 1 号请求，M1 为 2 号请求，以此类推。执行请求时，应开启相应的请求执行设备。 当通信完成后，请求执行设备自动关闭。如果需要持续发送请求，可使用 SET 或 OUT 指令将相应的请求执行设备保持开启状况。 如果未指定请求执行设备，将持续执行请求列表中编制的所有请求。
禁用	内部继电器或数据寄存器的位按照请求的顺序进行分配。
自动 Ping 连动	通过自动 Ping 连动的联机状态的 ON/OFF，可控制是否发送来自 Modbus TCP 客户端的请求。通过将不联网的远程主机请求设为发送对象外，可防止发生不需要的超时时间等问题。 有关自动 Ping 连动的详情，请参见第 3-23 页上的“自动 Ping 功能”。

组合请求执行设置的本体动作如下所示。

请求执行设置		自动 Ping 连动	
		不使用	使用
请求执行设备	不使用	始终发送所有请求。	仅联机状态开启时发送相应请求，关闭时不会发送相应请求。
	使用	仅请求执行设备开启时发送相应请求，关闭时不会发送相应请求。	仅联机状态和请求执行设备均开启时发送相应请求，除此以外不会发送相应请求。

通信完成和通信错误

当读取或写入过程成功完成后或发生通信错误时，Modbus 通信结束。如果通信故障重复三次，则发生通信错误。发生通信错误时，系统将取消该请求并处理下一个请求。如果指定错误状态数据寄存器，则可确认每项请求的通信状态。

通信错误数据

当通过功能设置在请求列表中设置错误状态后，便可确认每项请求的错误数据。

所有通信请求使用单个数据寄存器	各通信请求的错误数据
未确认	可以确认整个请求列表中各项请求的错误数据、远程主机号 (高位字节) 和错误代码 (低位字节)。根据请求数量保留数据寄存器以存储错误数据。当发生请求错误时，错误数据将存储到相应的数据寄存器中。

所有通信请求使用单个数据寄存器	各通信请求的错误数据
已确认	所有请求共享单个数据寄存器。当发生请求错误时，错误数据将存储到数据寄存器中并将覆盖旧错误数据。
位分配	
远程主机号 (高位字节)	1 ~ 255
错误代码 (低位字节)	00h: 正常完成 01h: 功能代码错误 (不支持的功能代码) 02h: 访问目标错误 (地址超出范围, 地址 + 设备量超出范围) 03h: 设备数错误, 1 位写入数据错误 (不支持 1 位写入的指定设备数量) 12h: 帧长度错误 (所发送请求的帧长度超出范围) 13h: BCC 错误 (BCC 不匹配) 14h: 从机编号错误 (收到的从机编号无效) 16h: 超时错误 (超时)

Modbus TCP 通信请求列表

可在 Modbus TCP 客户端请求列表中最多设置 255 项请求。

注释:

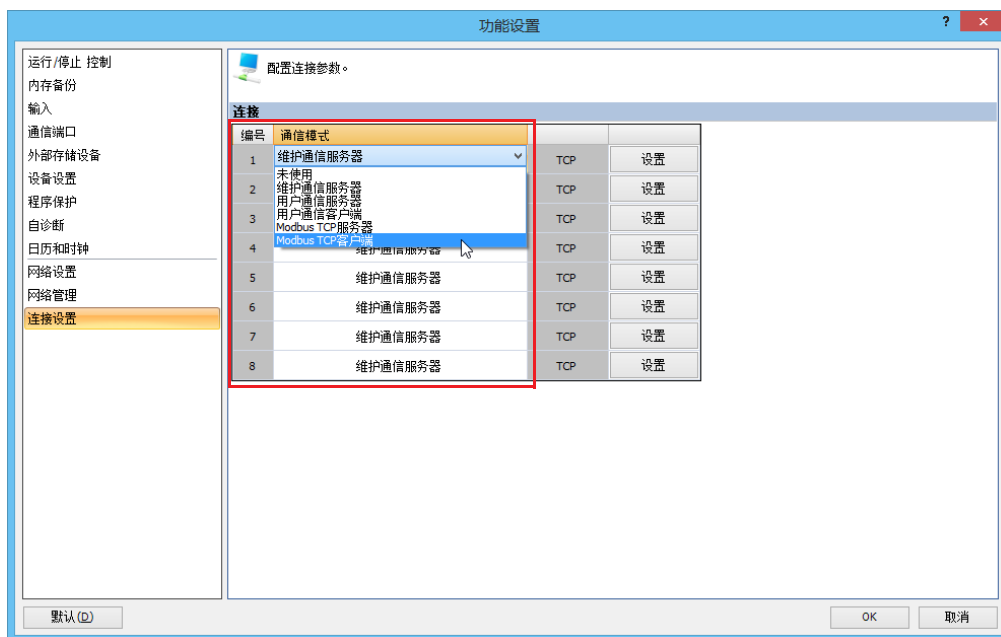
- 每项请求需要 10 个字节的用户程序区。
- 按照请求编号的顺序分配请求执行设备和错误状态数据寄存器。删除某项请求或更改请求顺序时，针对请求执行设备和错误状态数据寄存器的请求关系也将随之改变。如果在用户程序中使用分配的内部继电器或数据寄存器，则这些设备地址必须进行相应的更新。

6: MODBUS通信

设置 WindLDR（Modbus TCP 客户端）

若要使用 Modbus TCP 客户端，可在“功能设置”对话框中设置 Modbus TCP 客户端，然后将用户程序下载到 FC6A 型。

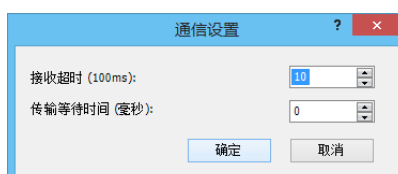
1. 从 WindLDR 菜单栏中选择“设置”>“连接设置”。
出现“功能设置”对话框。
2. 选择“Modbus TCP 客户端”作为连接 1 的通信模式。



出现“Modbus TCP 客户端”对话框。



3. 单击“通信设置”按钮。
出现“通信设置”对话框。设置接收超时时间和传输等待时间。
单击“确定”。



4. 设置 Modbus TCP 客户端的请求。

可在一份请求列表中最多输入 255 项请求。用远程主机号指定 Modbus TCP 服务器。设置完所有请求后，单击“确定”按钮关闭对话框。

如有必要，请选择使用“请求执行设备”和“错误状态”数据寄存器。当使用“请求执行设备”和“错误状态”数据寄存器时，输入设备的第一个编号。



5. 将用户程序下载到 FC6A 型。

至此，完成 Modbus TCP 客户端的编程。参数和有效值的详情如下。

功能代码

FC6A 型的 Modbus TCP 支持下表列出的八个功能代码。受支持的功能代码和有效的从机地址因要进行通信的各台 Modbus 服务器设备而异。请根据 Modbus 服务器设备的规格配置功能代码。

功能代码	数据大小	从机地址	作为 Modbus 从机的 FC6A 型
01 读取线圈状态	1 ~ 2,000 位	000001 - 065535	读取 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的位设备状态。
02 读取输入状态	1 ~ 2,000 位	100001 - 165535	读取 I (输入)、T (定时器接点) 或 C (计数器触点) 的位设备状态。
03 读取保持寄存器	1 ~ 125 字	400001 - 465535	读取 D (数据寄存器)、T (定时器预置值) 或 C (计数器预置值) 的字设备数据。
04 读取输入寄存器	1 ~ 125 字	300001 - 365535	读取 T (定时器当前值) 或 C (计数器当前值) 的字设备数据。
05 强制单线圈	1 位	000001 - 065535	更改 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的位设备状态。
06 预置单寄存器	1 字	400001 - 465535	更改 D (数据寄存器) 的字设备数据。
15 强制多线圈	1 ~ 1,968 位	000001 - 065535	更改 Q (输出)、R (移位寄存器) 或 M (内部继电器) 的多位设备状态。
16 预置多位数据寄存器	1 ~ 123 字	400001 - 465535	更改 D (数据寄存器) 的多字设备。

主机设备地址

当选择功能代码 01、02、03 或 04 从 Modbus 服务器读取数据时，指定第一个数据寄存器或内部继电器编号存储从 Modbus 从机接收的数据。当选择功能代码 05、06、15 或 16 向 Modbus 服务器写入数据时，指定第一个数据寄存器或内部继电器编号存储写入 Modbus 服务器的数据。可按照主机设备地址指定数据寄存器和内部继电器。

数据大小和字 / 位

指定要读取或写入的数据量。有效数据大小取决于功能代码。当选择了功能代码 01、02、05 或 15 时，以位指定数据大小。当选择了功能代码 03、04、06 或 16 时，以字指定数据大小。有关有效数据大小，请参见第 6-25 页上的“功能代码”。

远程主机号

指定在“远程主机列表”对话框中设置的远程主机号。在“远程主机列表”对话框中，设置每台远程主机的 IP 地址和端口号。Modbus TCP 通信的默认端口号为 502。如果 Modbus TCP 服务器使用不同的端口号，则应在“远程主机列表”对话框中设置该端口号。

从机编号

指定从机号 0 至 255。可以为 1 至 255 的不同请求编号重复指定相同的从机号。Modbus TCP 服务器未参照从机编号时，请设置任意从机编号。

从机地址

指定 Modbus 服务器的数据内存地址。有效从机地址范围取决于功能代码。有关有效从机地址，请参阅上表。内存地址的分配因各台 Modbus 服务器设备而异。请参阅各台 Modbus 服务器设备的手册。

请求执行设备

要使用请求执行设备，请在 Modbus TCP 客户端请求列表中单击“使用”单选按钮，并指定第一个内部继电器或数据寄存器。用于执行请求的内部继电器或数据寄存器的位自动列于表中。执行请求时，应开启相应的请求执行设备。

当没有指定请求执行设备时，连续执行请求列表中设置的所有请求。

错误状态数据寄存器

要使用错误状态数据寄存器，在 Modbus TCP 客户端请求列表中单击“使用”单选按钮并指定第一个数据寄存器。在表中自动列出用于存储错误状态的数据寄存器。当“所有通信请求使用一个数据寄存器”被选定时，所有请求将共用第一个数据寄存器。

Modbus TCP 服务器

将 FC6A 型配置为 Modbus TCP 服务器时，Modbus TCP 客户端设备可以读取 FC6A 型的数据或写入数据到 FC6A 型。当 FC6A 型从 Modbus TCP 客户端设备接收有效请求时，将根据接收到的请求读取或写入数据。从 Modbus TCP 客户端接收到的通信数据以用户程序的 END 处理方式进行处理。

Modbus TCP 服务器规格

参数	Modbus TCP 服务器
从机编号	忽略
响应时间	1.5ms
可以同时访问的客户端数 ^{*1}	All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块： 8（每个连接一个客户端） Plus CPU 模块：16（每个连接一个客户端）
端口号	502（可在 0 和 65535 之间进行更改）
支持的功能代码	01 读取线圈状态 02 读取输入状态 03 读取保持寄存器 04 读取输入寄存器 05 强制单线圈 06 预置单寄存器 15 强制多线圈 16 预置多位数据寄存器

*1 将 All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块 8 个连接、Plus CPU 模块 16 个连接全部设为 Modbus TCP 服务器时的个数。

Modbus TCP 服务器的从机地址变换

Modbus TCP 客户端可以访问 Modbus 服务器的 Modbus 设备（线圈、输入继电器、输入寄存器和保持寄存器）以读取或写入 FC6A 型的设备数据（I、Q、M、R、T、C 和 D）。请参阅下表来配置 Modbus TCP 客户端。

Modbus 设备名称	从机地址	通信从机地址 ^{*1}	FC6A 型设备 ^{*2}	适用功能代码
线圈 (000000 以上)	000001 - 000516	0000 - 0203	Q0 - Q643	1, 5, 15
	000701 - 000956	02BC - 03BB	R000 - R255	
	001001 - 003048	03E8 - 0BE7	M0000 - M2557	
	003049 - 007400	0BE8 - 1CE7	M2560 - M7997	
	009001 - 009256	2328 - 2427	M8000 - M8317	
	009257 - 010600	2428 - 2967	M8320 - M9997	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
	017001 - 020000	4268 - 4E1F	M17500 - M21247	
	030001 - 030480	7530 - 770F	Q1000 - Q1597	
	030481 - 030960	7710 - 78EF	Q2000 - Q2597	
	030961 - 031440	78F0 - 7ACF	Q3000 - Q3597	
	031441 - 031920	7AD0 - 7CAF	Q4000 - Q4597	
	031921 - 032400	7CB0 - 7E8F	Q5000 - Q5597	
	032401 - 032880	7E90 - 806F	Q6000 - Q6597	
	032881 - 033360	8070 - 824F	Q7000 - Q7597	
	033361 - 033840	8250 - 842F	Q8000 - Q8597	
	033841 - 034320	8430 - 860F	Q9000 - Q9597	
034321 - 034800	8610 - 87EF	Q10000 - Q10597		

6: MODBUS通信

Modbus 设备名称	从机地址	通信从机地址 ^{*1}	FC6A 型设备 ^{*2}	适用功能代码
输入继电器 (100000 以上)	100001 - 100516	0000 - 0203	I0 - I643	2
	101001 - 101256	03E8 - 04E7	T000 - T255 (定时器接点)	
	101501 - 101756	05DC - 06DB	C000 - C255 (计数器接点)	
	102001 - 102768	07D0 - 0ACF	T256 - T1023 (定时器接点)	
	102769 - 103744	0AD0 - 0E9F	T1024 - T1999 (定时器接点)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (计数器接点)	
	130001 - 130480	7530 - 770F	I1000 - I1597	
	130481 - 130960	7710 - 78EF	I2000 - I2597	
	130961 - 131440	78F0 - 7ACF	I3000 - I3597	
	131441 - 131920	7AD0 - 7CAF	I4000 - I4597	
	131921 - 132400	7CB0 - 7E8F	I5000 - I5597	
	132401 - 132880	7E90 - 806F	I6000 - I6597	
	132881 - 133360	8070 - 824F	I7000 - I7597	
	133361 - 133840	8250 - 842F	I8000 - I8597	
133841 - 134320	8430 - 860F	I9000 - I9597		
134321 - 134800	8610 - 87EF	I10000 - I10597		
输入寄存器 (300000 以上)	300001 - 300256	0000 - 00FF	T000 - T255 (定时器当前值)	4
	300501 - 300756	01F4 - 02F3	C000 - C255 (计数器当前值)	
	302001 - 302768	07D0 - 0ACF	T256 - T1023 (定时器当前值)	
	302769 - 303744	0AD0 - 0E9F	T1024 - T1999 (定时器当前值)	
	304001 - 304256	0FA0 - 109F	C256 - C511 (计数器当前值)	
保持寄存器 (400000 以上)	400001 - 408000	0000 - 1F3F	D0000 - D7999	3, 6, 16
	408001 - 408500	1F40 - 2133	D8000 - D8499	
	408501 - 408900	2134 - 22C3	D8500 - D8899	
	409001 - 409256	2328 - 2427	T000 - T255 (定时器预置值)	3
	409501 - 409756	251C - 261B	C000 - C255 (计数器预置值)	
	410001 - 456000	2710 - DABF	D10000 - D55999	3, 6, 16
	456001 - 462000	DAC0 - F22F	D56000 - D61999	
	462001 - 462768	F230 - F52F	T256 - T1023 (定时器预置值)	3
	462769 - 463744	F530 - F8FF	T1024 - T1999 (定时器预置值)	
464001 - 464256	FA00 - FAFB	C256 - C511 (计数器预置值)		

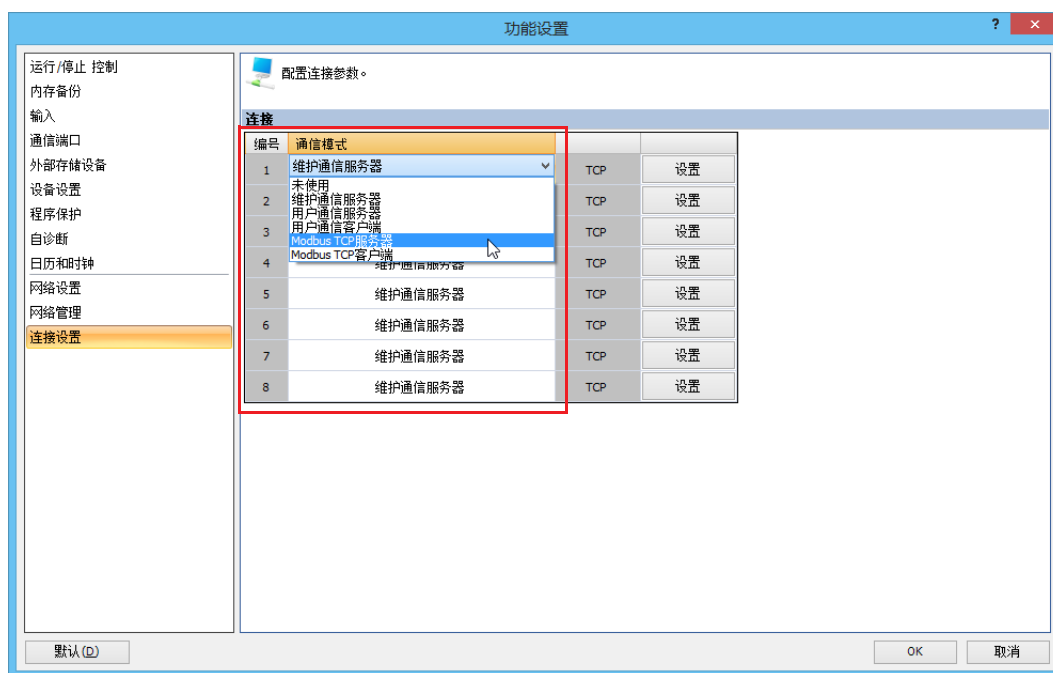
*1 通信从机地址为通信帧地址部中的 4 位数值。将从机地址的后 5 位减去 1 后的值，以十六进制进行存储。有关地址的计算方法的详情，请参见第 6-9 页上的“Modbus RTU 从机通信”。

*2 在所用 FC6A 型类型的设备范围内进行访问。

设置 WindLDR（Modbus TCP 服务器）

若要使用 Modbus TCP 服务器，可在“功能设置”对话框中设置 Modbus TCP 服务器，然后将用户程序下载到 FC6A 型。

1. 从 WindLDR 菜单栏中选择“设置”>“连接设置”。
出现“功能设置”对话框。
2. 选择“Modbus TCP 服务器”作为连接 1 的通信模式。



出现“Modbus TCP 服务器”对话框。

3. 设置参数，然后单“确定”按钮。



注释：有关每个参数的详情，请参见下一页。

4. 将用户程序下载到 FC6A 型。
至此，完成 Modbus TCP 服务器的编程。

6: MODBUS通信

Modbus TCP 服务器通信设置

本地主机端口号

将本地主机端口号配置为 0 到 65,535 之间。多个连接可以使用相同的本地主机端口号。

如果多个连接使用相同的端口号，则与连接数一样多的 Modbus TCP 客户端可通过相同的端口号与 FC6A 型进行通信。

允许 IP 地址访问

通过设置允许的 IP 地址，只有使用指定 IP 地址的设备才能与 FC6A 型建立连接并与 Modbus TCP 服务器进行通信。如果多个连接中配置了相同的本地主机端口号，则所有允许的 IP 地址设置均有效。如果未配置允许的 IP 地址的连接使用相同的本地主机端口号，则允许通过该端口进行任意访问。

例 1) 如果两个连接使用相同的本地端口号，但这两个连接均未配置允许的 IP 地址，则接受从具有任意 IP 地址的总共两个客户端进行访问。

例 2) 如果两个连接使用相同的本地端口号，且 192.168.1.101 和 192.168.1.102 配置为允许的 IP 地址，则接受从 IP 地址为 192.168.1.101 和 192.168.1.102 的总共两个客户端进行访问。

例 3) 如果连接 1 和连接 2 使用相同的本地端口号，并且连接 1 配置了允许的 IP 地址 192.168.1.101，而连接 2 未配置该允许的 IP 地址，则接受从具有任意 IP 地址的总共两个客户端进行访问。

Modbus TCP 通信格式

本节介绍 Modbus TCP 客户端和服务器通信所用的通信格式。以 Modbus TCP 首部启动 Modbus TCP 通信格式，然后是 RTU 模式通信格式，两端及 CRC 不带 3.5 个闲置字符，如下所示。

Modbus TCP 通信格式

交易 ID	协议 ID	信息长度 (字节)	单位 ID	功能代码	数据		
2 字节	2 字节	2 字节	1 字节	1 字节	N 字节		
Modbus TCP 首部							
RTU 模式 通信格式		闲置的 3.5 个字符	从机号	功能代码	数据	CRC	闲置的 3.5 个字符
			1 字节	1 字节	N 字节	2 字节	

交易 ID

Modbus TCP 服务器（从机）返回客户端（主机）发送的请求 ID 并保持不变。当收到返回的请求 ID 时，客户端可以确认要对哪一请求进行回复。如果无需确认，则指定 0 为交易 ID。

协议 ID

制定 0 以确定 Modbus TCP 协议。

信息长度

以字节为单位指定以下信息的长度。

单位 ID

识别设备的 ID。它存储 Modbus TCP 服务器从机编号。如果收到的请求的设备 ID 不是 0，则 FC6A 型 Modbus TCP 服务器将接受和处理请求。如果设备 ID 是 0，则收到的请求按广播通信进行处理，且不会向 Modbus TCP 客户端返回响应。

功能代码

指定功能代码，如 01（读取线圈状态）和 02（读取输入状态）。

数据

指定所需的各功能数据。

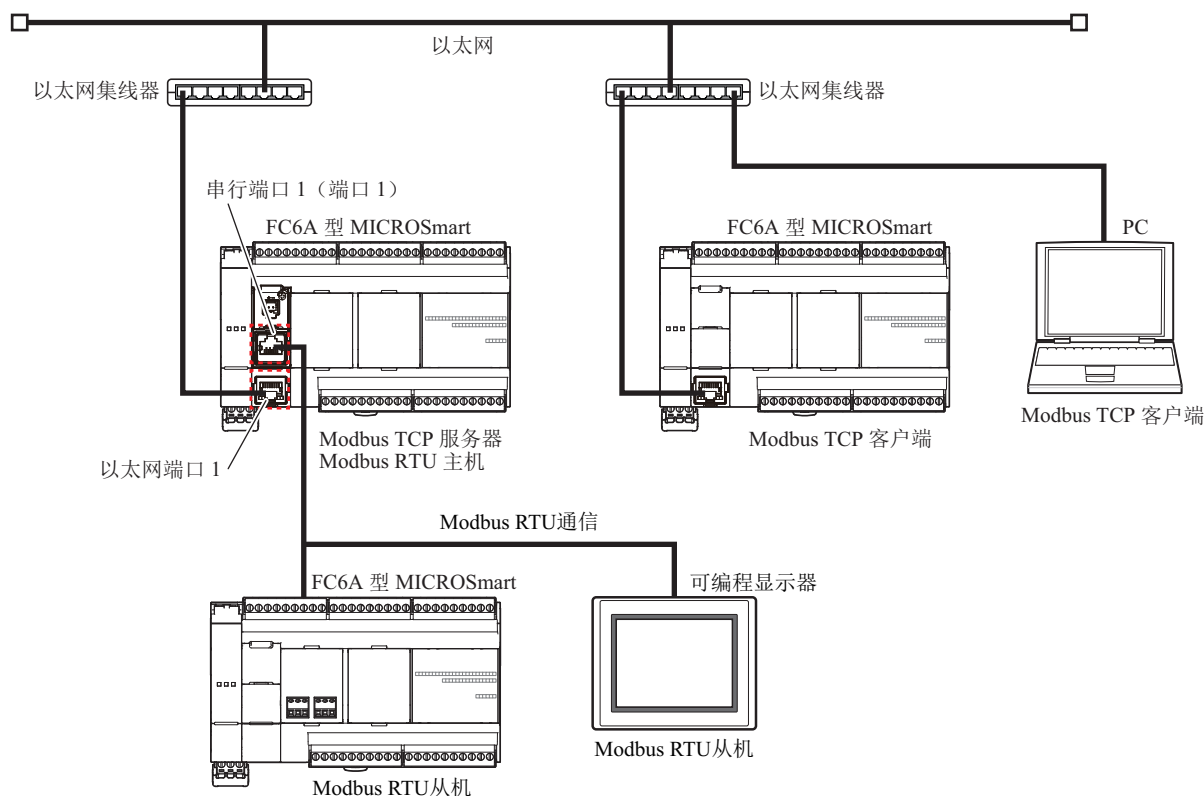
Modbus RTU Pass-Through 功能

Modbus RTU Pass-Through 功能是连接 Modbus TCP 网络的 Modbus TCP 客户端访问连接 Modbus RTU 网络的 Modbus RTU 从机的一种功能。Modbus RTU Pass-Through 功能可在 All-in-One CPU 模块及 Plus CPU 模块中使用。无法在 CAN J1939 All-in-One CPU 模块中使用。

All-in-One CPU 模块时，可通过连接以太网端口 1 的 Modbus TCP 客户端设备，读取或写入连接 FC6A 型串行端口 1 的 Modbus RTU 从机设备的设备信息。

Modbus RTU Pass-Through 功能仅在 All-in-One CPU 模块的以太网端口 1、串行端口 1 之间支持。

Plus CPU 模块时，可通过连接以太网端口 1 及 2 的 Modbus TCP 客户端设备，选择连接 Plus CPU 模块端口 1 ~ 33 的 Modbus RTU 从机设备，读取或写入设备信息。Modbus RTU Pass-Through 功能在 Plus CPU 模块以太网端口 1 及端口 2 和端口 1 ~ 33 之间支持。



6: MODBUS通信

Modbus RTU Pass-Through 功能的 Modbus TCP 通信格式

Modbus RTU Pass-Through 功能可从 Modbus TCP 服务器所接收的 Modbus TCP 通信帧删除 Modbus TCP 标题，并在“单元 ID”、“功能代码”、“数据”的帧中附加 CRC，将帧发送到单元 ID 所指定的 Modbus RTU 从机中。Modbus RTU Pass-Through 功能的 Modbus TCP 通信格式如下所示。

Modbus TCP 通信格式

交易 ID	协议 ID	信息长度(字节)	单元 ID	功能代码	数据
2 字节	2 字节	2 字节	1 字节	1 字节	N 字节

Modbus TCP 首部

RTU 通信格式

“Idle”	站号	功能代码	数据	CRC	“Idle”
3.5 个字符	1 字节	1 字节	N 字节	2 字节	3.5 个字符

交易 ID

Modbus TCP 服务器可直接返回来自客户端的交易 ID。客户端设备可确认是哪个交易的响应。若不特别确认时则输入 0。

协议 ID

将表示 Modbus TCP 协议的编号变为 0。

信息长度

以字节为单位表示以后连续的信息长度。

单元 ID

本 ID 用于识别设备。存储 Modbus RTU 从机的从机编号。将单元 ID 以后通过 Modbus RTU 网络。在单元 ID 中指定 255 时，不通过 Modbus RTU 从机，而是由 Modbus TCP 服务器（本站）进行处理

功能代码

读取及写入等功能的编号。

数据

各处理所需的数据。

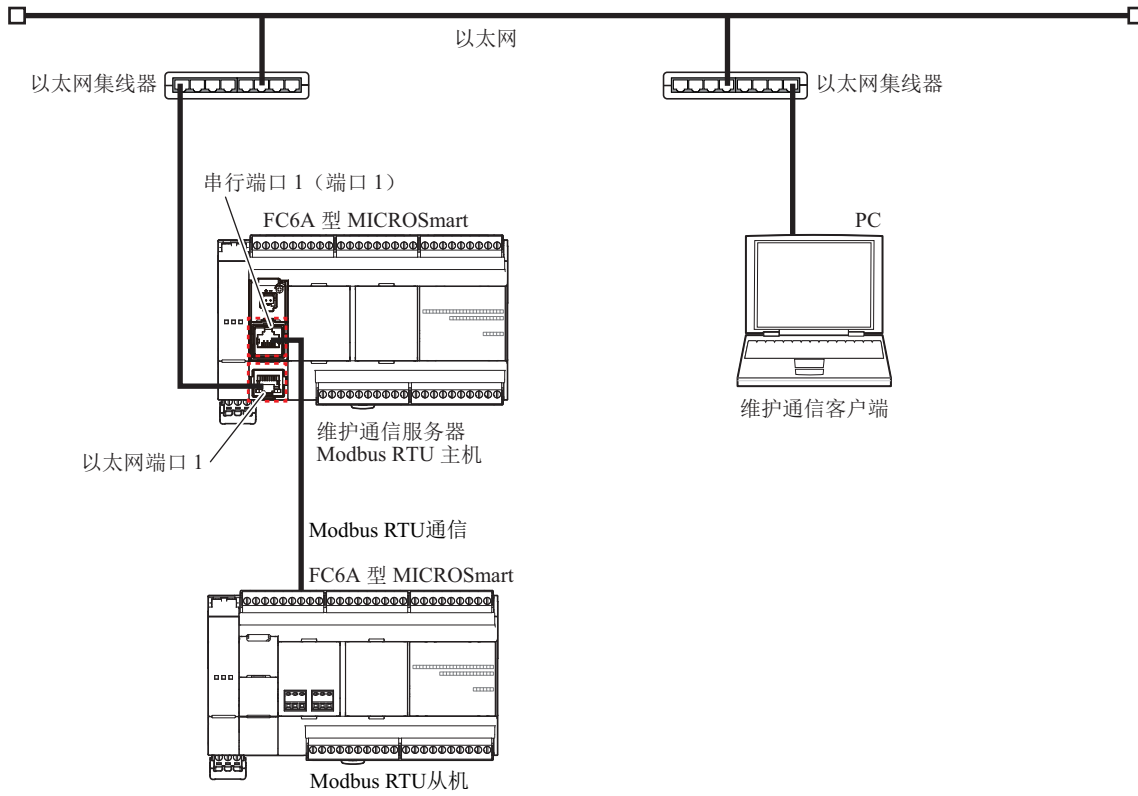
Modbus RTU Pass-Through 功能的规格

支持端口	以太网通信	All-in-One CPU 模块: 以太网端口 1 Plus CPU 模块: 以太网端口 1 及 2
	串行通信	All-in-One CPU 模块: 端口 1 Plus CPU 模块: 端口 1 ~ 33
通信模式	Modbus TCP 服务器 (从机) 和 Modbus RTU 主机	
启用 / 禁用设置	在“功能设置”中进行设置。 * 可启用的服务器连接数无限制。 * 可与经由维护通信服务器的 Pass-Through 并用。	
指定 Pass-Through 的位置	通过的 Modbus RTU 从机由 Modbus TCP 通信帧的单元 ID 进行指定。 <ul style="list-style-type: none"> 启用 Modbus RTU Pass-Through 功能时 单元 ID (0 ~ 254): 进行 Modbus RTU Pass-Through。 单元 ID (255): 由 Modbus TCP 服务器进行处理 禁用 Modbus RTU Pass-Through 功能时 单元 ID (0 ~ 255): 由 Modbus TCP 服务器进行处理。 Unit ID (0 to 255): Processed by the Modbus TCP server. 	
可通信的字节数	512 字节 * 超过 512 字节的数据 (帧) 不可进行 Pass-Through。接收 512 字节以上的数据时, 会丢弃包含之前的接收数据。	
Pass-Through 动作	单元 ID (从机站号) 并非 255 时, 在通信端口中发行 Pass-Through 请求, 等待来自 Modbus RTU 从机的响应。单元 ID 为 255 时, 会作为向本机通信进行分析, 对 Modbus TCP 客户端返回响应。 如果有来自 Modbus TCP 服务器的 Pass-Through 请求, 相较于 Modbus RTU 主机的请求处理请求, 应优先处理。	

6: MODBUS通信

使用维护通信服务器的 Modbus RTU Pass-Through

将 FC6A 型设为维护通信服务器，也可访问连接 Modbus RTU 网络的 Modbus RTU 从机。从可与安装有 WindLDR 或数据文件管理器的远程 PC 等的维护通信客户端，经由连接以太网的 FC6A 型（Modbus RTU 主机），可读取或写入设置在 Modbus RTU 从机中的 FC6A 型（Modbus RTU 从机）的设备及下载 / 上传项目数据。



设置 Modbus RTU Pass-Through 功能

设置 WindLDR

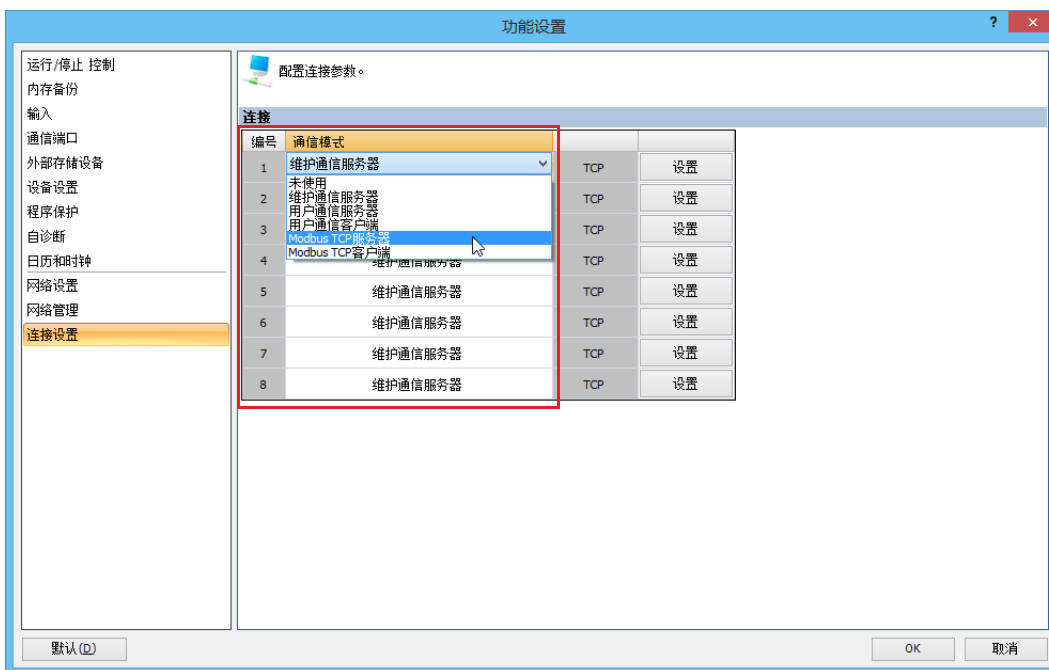
使用 Modbus RTU Pass-Through 功能时，将“功能设置”中设置的用户程序下载到 FC6A 型 All-in-One CPU 模块时

1. 从 WindLDR 菜单栏中，选择“设置”>“功能设置”>“连接设置”。

出现“功能设置”对话框。

2. 单击“端口 1”的“通信模式”，选择“Modbus TCP 服务器”。

将显示“Modbus TCP 服务器”对话框。



3. 选中“启用 Modbus RTU(端口 1)网关的 Modbus TCP”复选框，设置本地主机端口号和允许 IP 地址访问。请将使用 Pass-Through 功能端口的本地主机端口号指定为与其他服务器连接不同的编号。

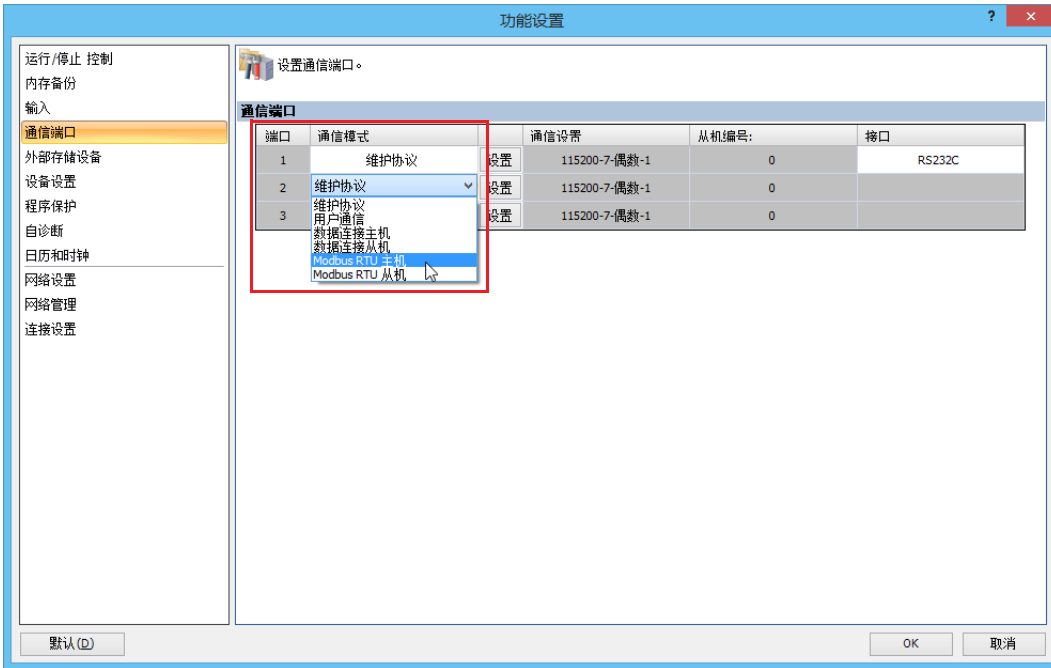


4. 单击“确定”。

6: MODBUS通信

- 单击“端口1”的“通信模式”，选择“Modbus RTU 主机”。

出现“Modbus RTU 主机请求列表”对话框。

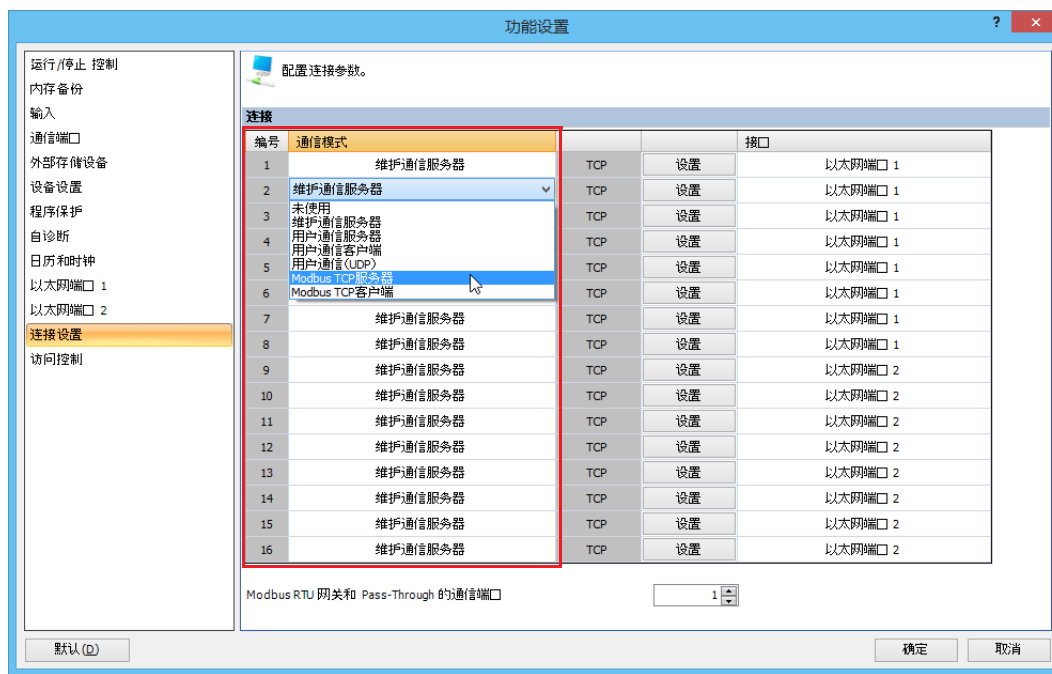


- 设置“Modbus RTU 主机请求列表”对话框的各项目。
有关详情，请参见第 6-3 页上的“Modbus RTU 主机通信”。
- 将用户程序下载到 FC6A 型。

至此，使用 Modbus RTU Pass-Through 功能的设置完成。

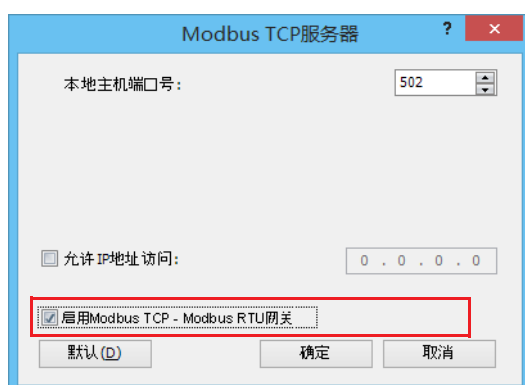
Plus CPU 模块时

1. 从 WindLDR 菜单栏中, 选择“设置”>“功能设置”>“连接设置”。
出现“功能设置”对话框。
2. 单击要使用连接编号的“通信模式”, 选择“Modbus TCP 服务器”。
将显示“Modbus TCP 服务器”对话框。



注释: 使用维护通信服务器的 Modbus RTU Pass-Through 时, 选择“维护通信服务器”。
将显示“维护通信服务器”对话框。

3. 选中“启用 Modbus TCP - Modbus RTU 网关”复选框, 设置本地主机端口号和允许 IP 地址访问。
请将使用 Pass-Through 功能端口的本地主机端口号设置为与其他服务器连接的端口号不同的编号。

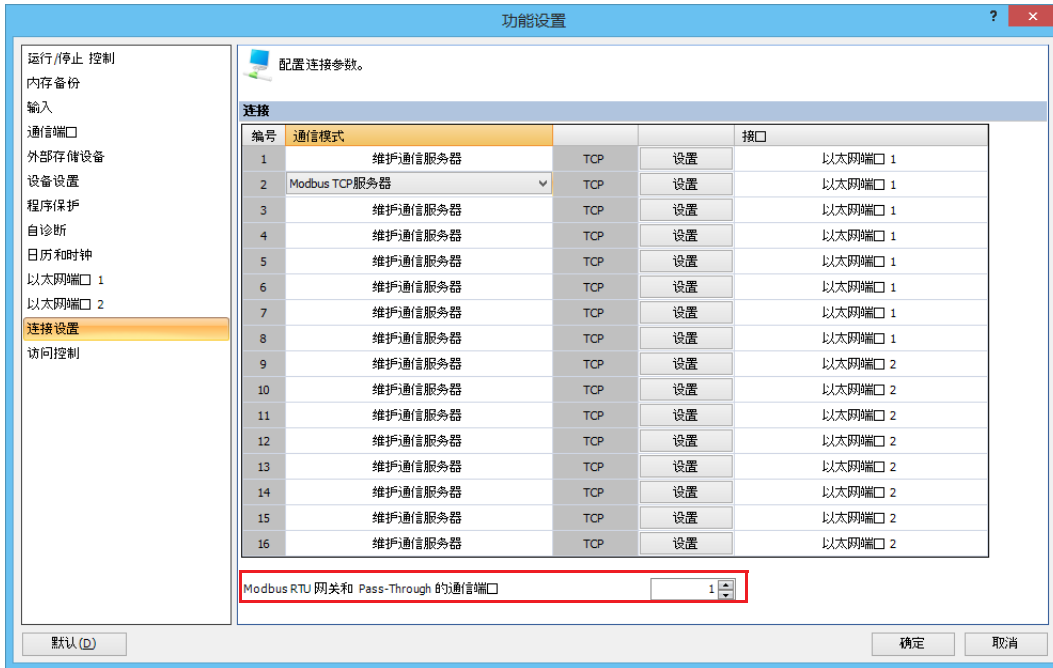


注释: 如果是“维护通信服务器”对话框, 应选中“通过 Modbus RTU 的直通功能”复选框, 设置本地主机端口号和允许 IP 地址访问。

4. 单击“确定”按钮。

6: MODBUS通信

5. 在“Modbus RTU 网关和 Pass-Through 的通信端口”中设置要使用的端口号（1～33）。



6. 将用户程序下载到 FC6A 型。

至此，使用 Modbus RTU Pass-Through 功能的设置完成。

7: 数据连接通信

简介

数据连接通信可在 1 台 FC6A 型（主站）上最多连接 31 台 FC6A 型（从站），并在主站与从站之间进行数据交换。主站拥有共计 13 个字的数据寄存器，包括每个从站中向从站发送的数据 6 个字、通过从站接收的数据 6 个字及通信状态 / 错误 1 个字。从站拥有共计 13 个字的数据寄存器，包括向主站发送的数据 6 个字、通过主站接收的数据 6 个字及通信状态 / 错误 1 个字。主站—从站间的通信处理不会与用户程序的处理同步进行，而是通过 END 处理进行数据寄存器内容的刷新（数据刷新）。

FC6A 型的数据连接功能与 FC5A 系列和 FC4A 系列的数据连接功能兼容。

一个 CPU 模块可以是一个主站也可是从站。数据连接主机和从机不能同时使用。

数据连接通信中使用的通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
端口 1	是	是	是	—	是 ^{*2}	是 ^{*2}
端口 2	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*1}	是 ^{*2}	是 ^{*2}
端口 3	—	—	是 ^{*1}	是 ^{*1}	是 ^{*3}	是 ^{*3}
端口 4 ~ 33	是（最多 3 个） ^{*4}			是（最多 15 个） ^{*5}		

*1 连接了 RS232C 通信盒或 RS485 通信盒时

*2 连接盒基本模块、连接了 RS232C 通信盒或 RS485 通信盒时

*3 连接 HMI 模块、连接了 RS232C 通信盒或 RS485 通信盒时

*4 All-in-One CPU 模块及 CAN J1939 All-in-One CPU 模块中最多可连接 3 台通信模块，以扩展端口 4 ~ 9。

*5 Plus CPU 模块中最多可连接 15 台通信模块，以扩展端口 4 ~ 33。

数据连接规格

模式	说明
电气规格	兼容 EIA RS485
通信速度	19,200、38,400、57,600bps
同步	开始—停止同步 开始位 :1 数据长度 :7 奇偶校验 :偶数 停止位 :1
通信电缆	屏蔽双绞电缆
最大电缆长度	总共 200m (656ft.)
最大从机站数量	31 个从机站
发送 / 接收数据	发送数据 : 最大 186 字，接收数据 : 最大 186 字 0 ~ 6 字，供从机站发送和接收

数据刷新

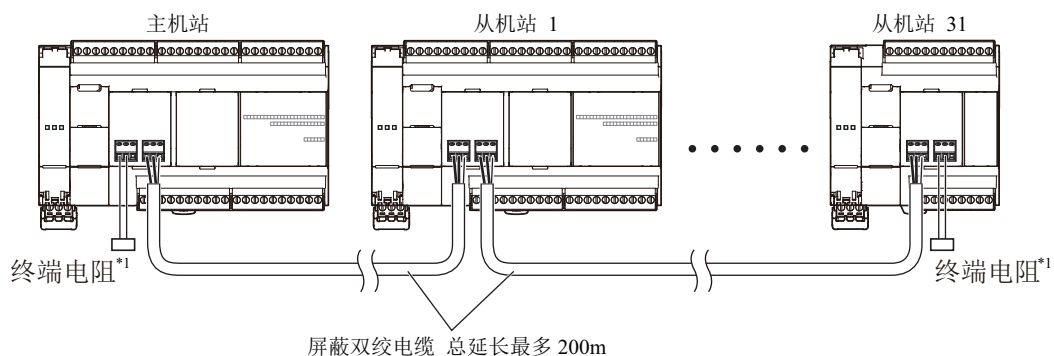
模式	说明
对扫描时间的影响	主站—从站间的通信处理不会与用户程序同步进行，因此不会影响扫描时间。
数据刷新时间	主站和从站的数据刷新均由 END 处理进行。可在通信完成标记中确认刷新时间。
主站支持类型	FC6A 型（FC4A 型 / FC5A 型）
从站支持类型	FC6A 型（FC4A 型 / FC5A 型）

主站会将发送字数的数据发送到从站，且接收该数据的从站会将接收字数的数据发送到主站，并进行数据交换。主站和从站的数据刷新均由 END 处理进行。主站可在 1 次通信处理中交换仅 1 个从站的数据。连接 31 台从站时，为了与所有从站交换数据，需要进行 31 次的通信处理。关于通信处理完成后的从站，主站会开启在通信完成后的仅 1 次扫描的通信完成继电器。

7: 数据连接通信

数据连接系统安装

将 FC6A 型的端口 1 设为 RS485 通信，或只要在端口 2 或端口 3 中使用 RS485 通信盒，即可使用数据连接功能。



*1 通信质量不稳定时，请将符合特性阻抗的终端电阻插入到两端。请使用额定 1/2W 以上的电阻。

指定发送 / 接收数据的数据寄存器

主机站有 12 个数据寄存器可用于与每个从机站进行数据通信。每个从机站有 12 个数据寄存器可用于与主机站进行数据通信。当数据存储在指定进行数据连接通信的主机站的数据寄存器中时，该数据将发送至从机站的相应数据寄存器中。当数据存储在指定进行数据连接通信的从机站数据寄存器中时，该数据将发送至主机站的相应数据寄存器中。

主机站

从机站编号	数据寄存器	发送 / 接收数据	从机站编号	数据寄存器	发送 / 接收数据
从机站 1	D900-D905	发送数据至从机站 1	从机站 17	D1092-D1097	发送数据至从机站 17
	D906-D911	自从机站 1 接收数据		D1098-D1103	自从机站 17 接收数据
从机站 2	D912-D917	发送数据至从机站 2	从机站 18	D1104-D1109	发送数据至从机站 18
	D918-D923	自从机站 2 接收数据		D1110-D1115	自从机站 18 接收数据
从机站 3	D924-D929	发送数据至从机站 3	从机站 19	D1116-D1121	发送数据至从机站 19
	D930-D935	自从机站 3 接收数据		D1122-D1127	自从机站 19 接收数据
从机站 4	D936-D941	发送数据至从机站 4	从机站 20	D1128-D1133	发送数据至从机站 20
	D942-D947	自从机站 4 接收数据		D1134-D1139	自从机站 20 接收数据
从机站 5	D948-D953	发送数据至从机站 5	从机站 21	D1140-D1145	发送数据至从机站 21
	D954-D959	自从机站 5 接收数据		D1146-D1151	自从机站 21 接收数据
从机站 6	D960-D965	发送数据至从机站 6	从机站 22	D1152-D1157	发送数据至从机站 22
	D966-D971	自从机站 6 接收数据		D1158-D1163	自从机站 22 接收数据
从机站 7	D972-D977	发送数据至从机站 7	从机站 23	D1164-D1169	发送数据至从机站 23
	D978-D983	自从机站 7 接收数据		D1170-D1175	自从机站 23 接收数据
从机站 8	D984-D989	发送数据至从机站 8	从机站 24	D1176-D1181	发送数据至从机站 24
	D990-D995	自从机站 8 接收数据		D1182-D1187	自从机站 24 接收数据
从机站 9	D996-D1001	发送数据至从机站 9	从机站 25	D1188-D1193	发送数据至从机站 25
	D1002-D1007	自从机站 9 接收数据		D1194-D1199	自从机站 25 接收数据
从机站 10	D1008-D1013	发送数据至从机站 10	从机站 26	D1200-D1205	发送数据至从机站 26
	D1014-D1019	自从机站 10 接收数据		D1206-D1211	自从机站 26 接收数据
从机站 11	D1020-D1025	发送数据至从机站 11	从机站 27	D1212-D1217	发送数据至从机站 27
	D1026-D1031	自从机站 11 接收数据		D1218-D1223	自从机站 27 接收数据
从机站 12	D1032-D1037	发送数据至从机站 12	从机站 28	D1224-D1229	发送数据至从机站 28
	D1038-D1043	自从机站 12 接收数据		D1230-D1235	自从机站 28 接收数据
从机站 13	D1044-D1049	发送数据至从机站 13	从机站 29	D1236-D1241	发送数据至从机站 29
	D1050-D1055	自从机站 13 接收数据		D1242-D1247	自从机站 29 接收数据
从机站 14	D1056-D1061	发送数据至从机站 14	从机站 30	D1248-D1253	发送数据至从机站 30
	D1062-D1067	自从机站 14 接收数据		D1254-D1259	自从机站 30 接收数据
从机站 15	D1068-D1073	发送数据至从机站 15	从机站 31	D1260-D1265	发送数据至从机站 31
	D1074-D1079	自从机站 15 接收数据		D1266-D1271	自从机站 31 接收数据
从机站 16	D1080-D1085	发送数据至从机站 16	—		
	D1086-D1091	自从机站 16 接收数据			

如果没有连接从机站，指定给空缺从机站的主机站数据寄存器可用作普通数据寄存器。

从机站

数据	数据寄存器	发送 / 接收数据
从机站数据	D900-D905	发送数据至主机站
	D906-D911	从主机站接收数据

从机站数据寄存器 D912 ~ D1271 可用作普通数据寄存器。

7: 数据连接通信

用于数据连接通信错误的特殊数据寄存器

除了指定进行数据通信的数据寄存器，主机站有 31 个特殊数据寄存器，而每个从机站有一个特殊数据寄存器，用于存储数据连接通信错误代码。如果数据连接系统中出现通信错误，通信错误代码将存储至主机站相应的数据寄存器中以示连接通信错误，以及从机站数据寄存器 D8069 中。有关连接通信错误代码的详细信息，请参见下文。如果数据连接通信系统中出现通信错误，数据将重新发送两次。如果三次重试后错误仍然存在，则错误代码存储至数据寄存器中以示数据连接通信错误。由于错误代码不在主机站和从机站之间通信，所以必须单独清除错误代码。

主机站

特殊数据寄存器	数据连接通信错误数据	特殊数据寄存器	数据连接通信错误数据
D8069	从机站 1 通信错误	D8085	从机站 17 通信错误
D8070	从机站 2 通信错误	D8086	从机站 18 通信错误
D8071	从机站 3 通信错误	D8087	从机站 19 通信错误
D8072	从机站 4 通信错误	D8088	从机站 20 通信错误
D8073	从机站 5 通信错误	D8089	从机站 21 通信错误
D8074	从机站 6 通信错误	D8090	从机站 22 通信错误
D8075	从机站 7 通信错误	D8091	从机站 23 通信错误
D8076	从机站 8 通信错误	D8092	从机站 24 通信错误
D8077	从机站 9 通信错误	D8093	从机站 25 通信错误
D8078	从机站 10 通信错误	D8094	从机站 26 通信错误
D8079	从机站 11 通信错误	D8095	从机站 27 通信错误
D8080	从机站 12 通信错误	D8096	从机站 28 通信错误
D8081	从机站 13 通信错误	D8097	从机站 29 通信错误
D8082	从机站 14 通信错误	D8098	从机站 30 通信错误
D8083	从机站 15 通信错误	D8099	从机站 31 通信错误
D8084	从机站 16 通信错误	—	—

如果没有连接从机站，指定给空缺从机站的主机站数据寄存器可用作普通数据寄存器。

从机站

特殊数据寄存器	数据连接通信错误数据
D8069	从机站通信错误

注释：从机站数据寄存器 D8070 ~ D8099 可用作普通数据寄存器。

数据连接通信错误代码

如果在数据连接时发生错误，则会重新发送（重试）最多 2 次的数据。

如果发送 3 次时仍发生错误，将在主站和相应从站的通信状态 / 错误用数据寄存器中调整错误编号。

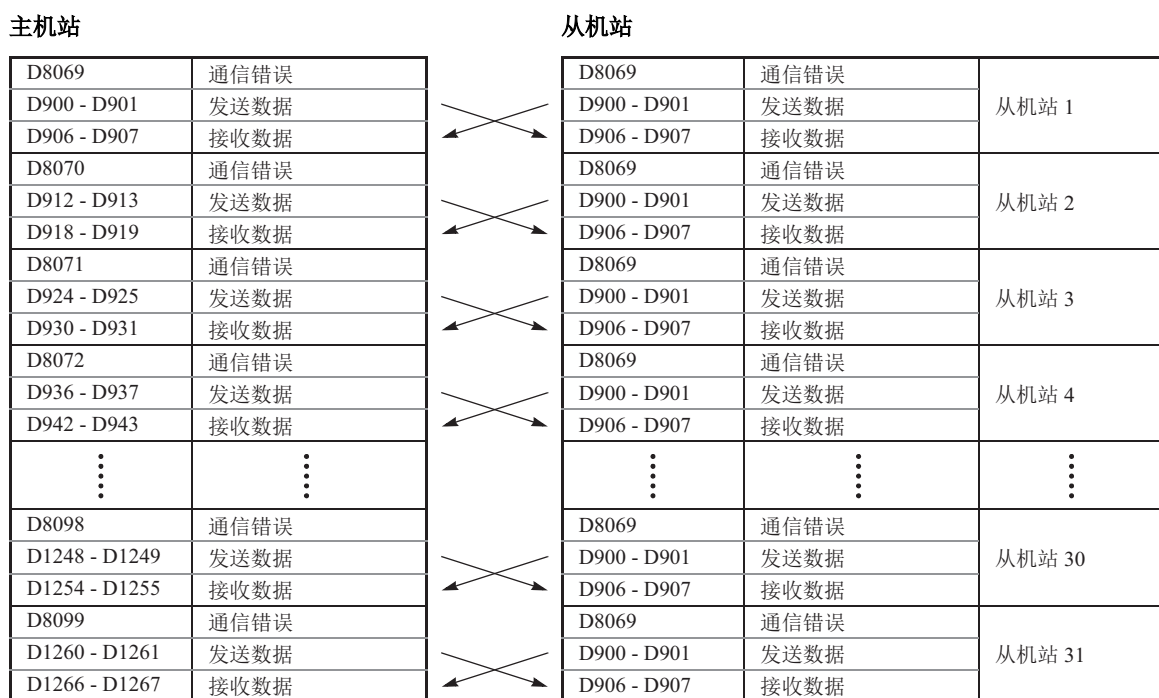
错误代码	错误详细信息
1h (1)	数据溢出错误（当接收数据寄存器已满时收到数据）
2h (2)	数据格式错误（无法检测开始或停止位）
4h (4)	奇偶校验错误（发现奇偶校验错误）
8h (8)	接收超时（连接断开）
10h (16)	BCC（块校验字符）错误（与 BCC 接收的数据不一致）
20h (32)	重试次数溢出（在 3 次通信重试中均出现的错误）
40h (64)	I/O 定义数量错误（发送 / 接收机站编号或数据量不一致）

当数据连接系统中检测到多个错误时，指示最大误差代码。例如，当出现数据格式错误（错误代码 2h）和 BCC 错误（错误代码 10h）时，将存储错误代码 12h（18）。

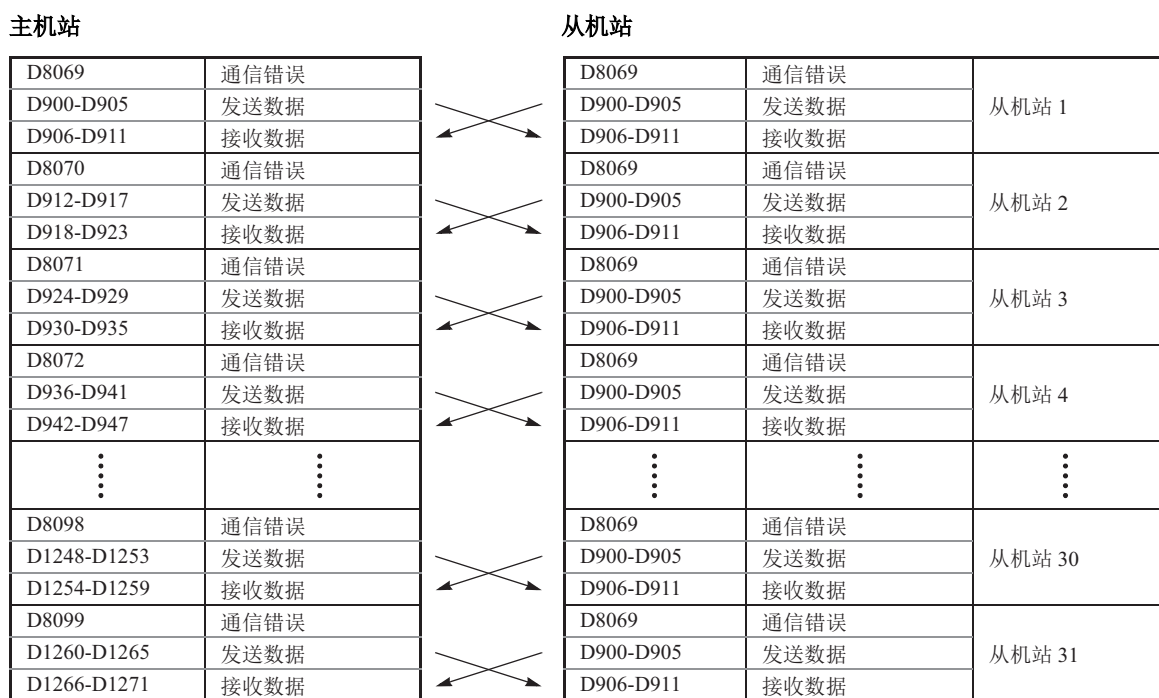
主机站和从机站之间的数据连接通信

主机站有 6 个数据寄存器用于发送数据至从机站，还有 6 个数据寄存器用于从从机站接收数据。可以使用 WinLDR 选择 0 ~ 6 个数据寄存器进行数据连接。以下示例介绍当使用 2 或 6 个数据寄存器与 31 个从机站进行数据连接通信时，主机站和从机站之间如何进行数据交换。

示例 1: 发送数据 2 字和接收数据 2 字



示例 2: 发送数据 6 字和接收数据 6 字



注释: 使用端口 2 或端口 3 时，通信状态 / 错误将变为“功能设置”中设置的寄存器。

7: 数据连接通信

用于数据连接通信的特殊内部继电器

可以指定特殊内部继电器 M8005 ~ M8007 和 M8080 ~ M8117 进行数据连接通信。

M8005 数据连接通信错误

当数据连接通信系统中出现错误时，M8005 打开。当错误被清除时，M8005 状态将保持不变，并保留到使用 WindLDR 将 M8005 复位时或 CPU 关闭时为止。通过使用“联机”>“监控”可以对通信错误的原因进行检查，然后选择“联机”>“PLC 状态”>“错误状态：详细信息”。请参见第 7-4 页。

通过端口 2 或端口 3 进行通信时，将禁用本功能。

M8006 数据连接通信禁止标记（主机站）

在数据连接系统中打开位于主机站的 M8006 时，将停止数据连接通信。当 M8006 关闭时，数据连接通信恢复。当 CPU 关闭时，将保持 M8006 状态直到使用 WindLDR 将 M8006 复位时为止。

当主机站的 M8006 打开时，数据连接系统中的从机站 M8007 打开。

M8007 数据连接通信初始化标记（主机站）

数据连接通信停止标记（从机站）

在数据连接通信系统的主机站或从机站上，M8007 具有不同的功能。

主机站：数据连接通信初始化标记

当运行中打开主机站的 M8007 时，将检查连接设置以初始化数据连接系统。当从机站在主机站之后打开电源时，请打开 M8007 初始化数据连接系统。在数据连接系统设置更改后，必须打开 M8007 以确保通信正确。

从机站：数据连接通信停止标记

在数据连接系统中，当从机站没有从主机站接收通信数据持续 10 秒钟以上时，M8007 将打开。若在初始化数据连接系统后的 10 秒钟内从机站没有接收数据，则从机站上的 M8007 也会打开。当从机站收到正确的通信数据时，M8007 将关闭。

M8080-M8116 从机站通信完成继电器（主机站）

特殊内部继电器 M8080 ~ M8116 用于指示数据刷新完成。当与从机站的数据连接通信完成时，指定给从机站的特殊内部继电器打开，持续时间为主机站一个扫描周期。

特殊内部继电器	从机站编号	特殊内部继电器	从机站编号	特殊内部继电器	从机站编号
M8080	从机站 1	M8092	从机站 11	M8104	从机站 21
M8081	从机站 2	M8093	从机站 12	M8105	从机站 22
M8082	从机站 3	M8094	从机站 13	M8106	从机站 23
M8083	从机站 4	M8095	从机站 14	M8107	从机站 24
M8084	从机站 5	M8096	从机站 15	M8110	从机站 25
M8085	从机站 6	M8097	从机站 16	M8111	从机站 26
M8086	从机站 7	M8100	从机站 17	M8112	从机站 27
M8087	从机站 8	M8101	从机站 18	M8113	从机站 28
M8090	从机站 9	M8102	从机站 19	M8114	从机站 29
M8091	从机站 10	M8103	从机站 20	M8115	从机站 30
—	—	—	—	M8116	从机站 31

M8080 通信完成继电器（从机站）

当与主机站的数据连接通信完成时，从机站特殊内部继电器 M8080 打开，并持续一个扫描周期。

M8117 所有从机站通信完成继电器

当与所有从机站的数据连接通信完成时，主机站的特殊内部继电器 M8117 打开，并持续一个扫描周期。从机站的 M8117 不打开。

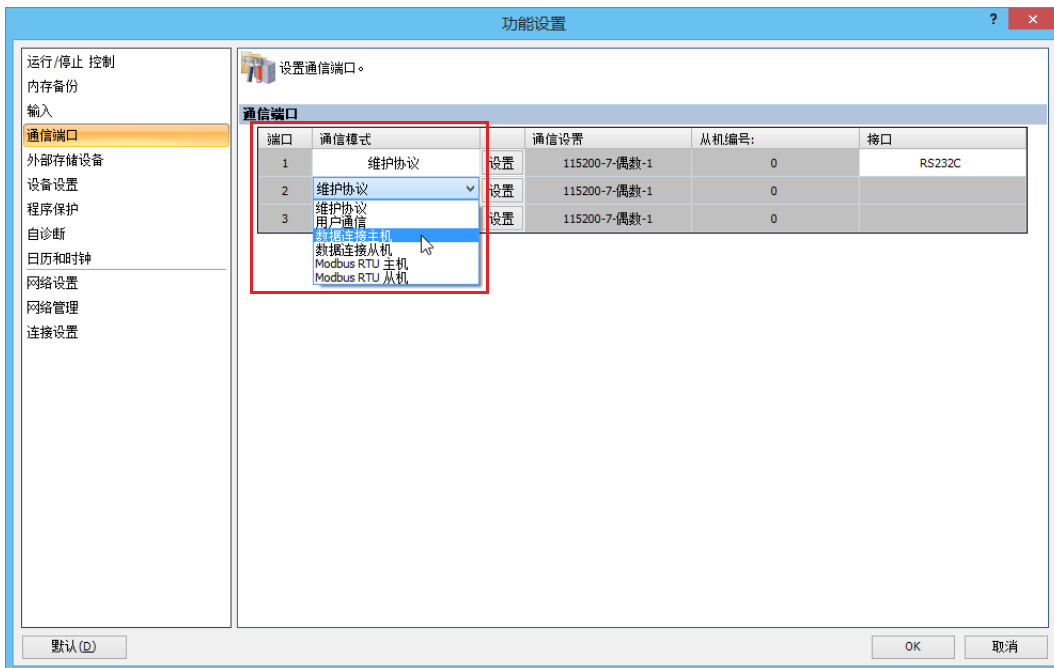
编程 WindLDR

“功能设置”中的通信选项用于编写数据连接主机站和从机站。

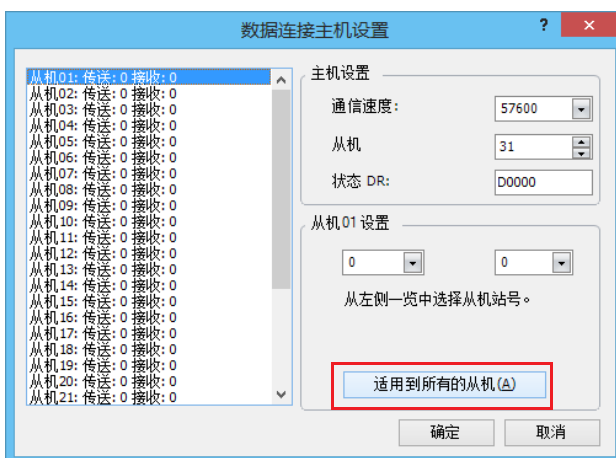
由于这些设置与用户程序相关，所以必须在更改设置后将用户程序下载到 CPU 模块。

数据连接主机站

1. 在 WindLDR 菜单栏中选择“设置”>“功能设置”>“通信端口”。此时出现通信端口的“功能设置”对话框。
2. 单击所使用端口编号的“通信模式”，选择“数据连接主机”。出现“数据连接主机设置”对话框。



3. 设置主设置的通信速度、从机及状态 DR。
4. 单击列表内的从机编号后进行选择，并设置发送字数和接收字数。要对列表的所有从站进行相同设置时，应单击“适用到所有的从机”按钮。



发送字数： 主站向从站发送的数据字数

接收字数： 主站通过从站发送的数据字数

7: 数据连接通信

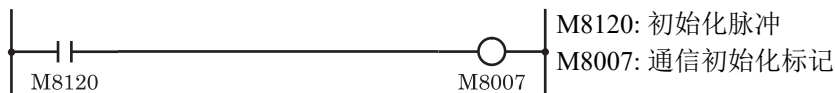
5. 单击“确定”按钮。
6. 创建用户程序。
7. 传送用户程序。

配置即完成。

注释：如果要使用数据连接通信，需要首先接通从站的电源再接通主站的电源。如果在接通主站电源后再接通从站电源，主站将无法识别从站。主站无法识别从站时，需要进行数据连接的初始化。进行数据连接的初始化时，在 WindLDR 中单击“初始化数据连接”按钮或在用户程序中开启特殊内部继电器 M8007。

1. 在“联机”选项卡的“监控”中单击“监控”，然后单击“启动监控器”。变为监控模式。
2. 在“联机”选项卡的“PLC”中单击“初始化”，然后单击“初始化数据连接”。主站可识别出从站。

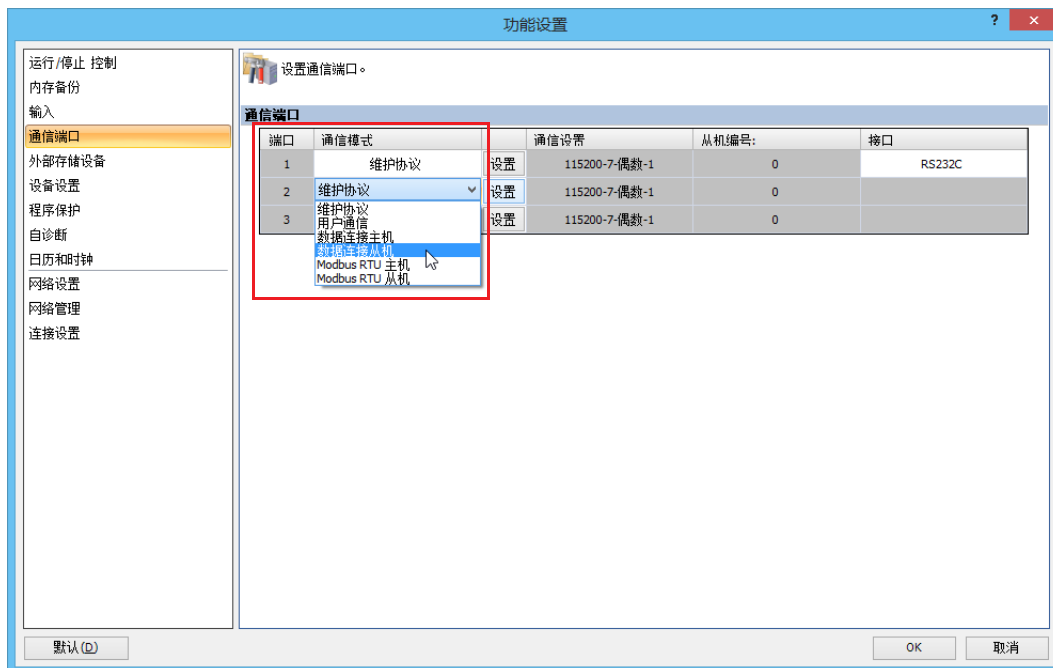
无法进行 WindLDR 的维护通信时，请在主站的用户程序中放入以下程序。
停止主站后，可通过运行来识别从站。



开始运行时的仅 1 次扫描 M8007 开启，进行数据连接的初始化。

数据连接从机站

1. 在 WindLDR 菜单栏中选择“设置”>“功能设置”>“通信端口”。此时出现通信端口的“功能设置”对话框。
2. 单击所使用端口编号的“通信模式”，选择“数据连接从机”。出现“数据连接从机设置”对话框。



3. 设置数据连接站号和通信速度。

在“数据连接站号”中选择“常量”或“数据寄存器”。选择“常量”时，即可设置数据连接站号。选择“数据寄存器”时，即可在特殊数据寄存器中写入数据连接站号。



4. 单击“确定”按钮。

5. 创建用户程序。

6. 传送用户程序。

配置即完成。

注释：在“数据连接从机设置”对话框的“数据连接编号”中选择“数据寄存器”时，可通过在以下特殊数据寄存器中写入数据连接站号，在未下载用户程序的情况下变更从站的数据连接站号。

通信端口	特殊数据寄存器
端口 1	D8100
端口 2	D8102
端口 3	D8103
端口 4 ~ 9 ^{*1}	D8040 ~ D8045
端口 10 ~ 33 ^{*2}	D8735 ~ D8758

*1 使用通信模块时

*2 仅 Plus CPU 模块

变更后，请正确进行主站的初始化及从站设置。

采用以下任意一种方法进行主站初始化。

(1) 切断主站电源后重新接通。

(2) 调整主站 M8007。

(3) 在“联机”选项卡的“PLC”中单击“初始化”，然后单击“初始化数据连接”。

另外，数据连接站号的值为 1 至 31 时有效。若为其他值，将启用“功能设置”中所设置的站号。

7: 数据连接通信

数据刷新

在数据连接通信中，在一个通信周期中主机站只与一个从机站通信。当从机站接收来自主机站的通信时，从机站返回用于数据连接通信的数据寄存器中存储的数据。从从机站收到数据后，主机站将数据存储在分配给每个从机站的数据寄存器中。将更新数据存入数据寄存器中称为刷新。当最多连接 31 个从机站时，主机站需要 31 个通信周期以与所有从机站进行通信。

模式	局部刷新模式
扫描时间	因为主机站和从机站之间的通信独立于用户程序扫描，所以不影响扫描时间。
数据刷新定时	在主机站和从机站，END 处理时都刷新接收的数据。可以使用通信完成特殊内部继电器 M8080 ~ M8117 确认刷新完成。
可用主机站	FC6A 型 (FC4A/FC5A MICROSmart)
可用从机站	FC6A 型 (FC4A/FC5A MICROSmart)

主机站和从机站都在 END 处理时刷新通信数据。当数据刷新完成后，主机站的通信完成特殊内部继电器 M8080 ~ M8116（从机站通信完成继电器）持续打开一个扫描周期。每个从机站上的特殊内部继电器 M8080（通信完成继电器）打开。

当主机站完成与所有从机站的通信后，主机站的特殊内部继电器 M8117（所有从机站通信完成继电器）持续打开一个扫描周期。

主机站与所有从机站通信的总刷新时间 (Trfn)

主机站需要下列时间刷新与所有从机站通信的发送和接收数据，这就是总刷新时间。

$$[\text{波特率 } 19,200\text{bps}]Trfn = \sum Trf = \sum \{4.2\text{ms} + 2.4\text{ms} \times (\text{发送字} + \text{接收字}) + 1 \text{ 次扫描时间} \}$$

$$[\text{波特率 } 38,400\text{bps}]Trfn = \sum Trf = \sum \{2.2\text{ms} + 1.3\text{ms} \times (\text{发送字} + \text{接收字}) + 1 \text{ 次扫描时间} \}$$

$$[\text{波特率 } 57,600\text{bps}]Trfn = \sum Trf = \sum \{1.6\text{ms} + 0.9\text{ms} \times (\text{发送字} + \text{接收字}) + 1 \text{ 次扫描时间} \}$$

刷新时间

当使用发送字 6、接收字 6、从机站 8 和平均扫描时间 20ms 等参数进行数据连接通信时，与 8 个从机站通信的总刷新时间 Trf8 将是：

$$[\text{波特率 } 19,200\text{bps}]Trf8 = \{4.2\text{ms} + 2.4\text{ms} \times (6 + 6) + 20\text{ms}\} \times 8 = 424.0\text{ms}$$

$$[\text{波特率 } 38,400\text{bps}]Trf8 = \{2.2\text{ms} + 1.3\text{ms} \times (6 + 6) + 20\text{ms}\} \times 8 = 302.4\text{ms}$$

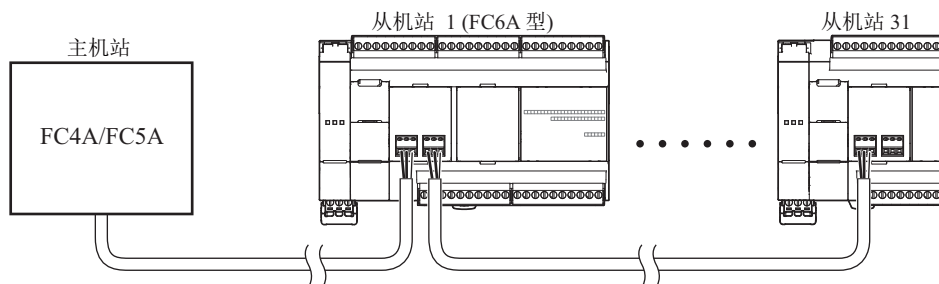
$$[\text{波特率 } 57,600\text{bps}]Trf8 = \{1.6\text{ms} + 0.9\text{ms} \times (6 + 6) + 20\text{ms}\} \times 8 = 259.2\text{ms}$$

使用其他 PLC 的数据连接

FC6A 型可通过与 FC4A 型、FC5A 型 MICROSmart 进行通信来实现数据的互换（数据连接）。

FC4A/FC5A 的数据连接

FC4A/FC5A 侧设置	设置
发送数据 6 个字 接收数据 6 个字 19,200bps/38,400bps	将数据连接从机设为 1 至 31



8: J1939 通信

本章将对 CAN J1939 All-in-One CPU 模块的 J1939 通信进行介绍。

为了简明扼要的进行介绍，正文中使用了以下缩略语。

正文中的缩略语	正式名称
CAN	Controller Area Network
ECU	Electronic Controller Unit
NIECU	Network Interconnection ECU
CA	Controller Application
PDU	Protocol Data Unit
PGN	Parameter Group Number
SA	Source Address
DA	Destination Address
SAE	Society of Automotive Engineers
SPN	Suspect Parameter Number

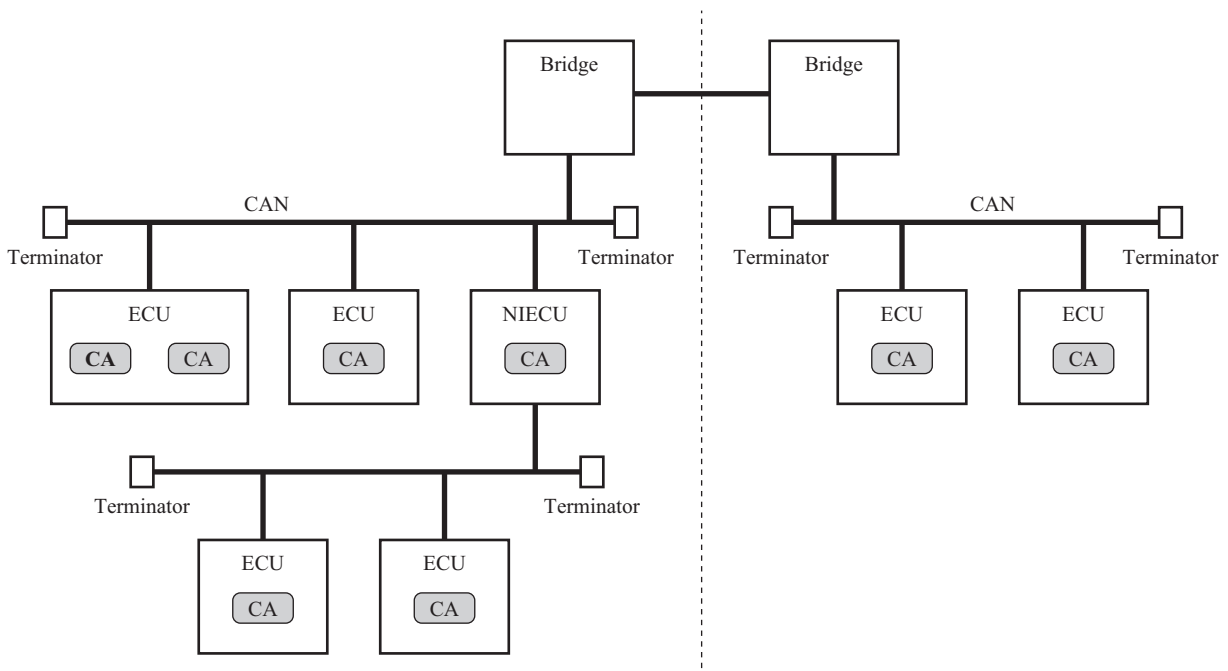
通过 CAN 进行 J1939 通信的概要

CAN J1939 All-in-One CPU 模块支持基于 SAE J1939 的 CAN 通信。使用 CAN 端口连接 J1939 通信网络，可与其他支持 J1939 通信的设备进行通信。有关 J1939 通信的详情，请参见 SAE J1939 的标准手册。

SAE J1939 的概要

J1939 通信网络

J1939 通信网络由 ECU（发动机、制动器等）和 NIECU（中继器、路由器等）构成，各 ECU 带有 1 个以上的 CA。CA 上分配了 0~253 个地址和 64 位的固定 ID（NAME），使用地址 CA 之间可收发信息。此外，在 CA 之间地址发生冲突时，将根据 NAME 所示优先度解决冲突。



8: J1939通信

设备地址

地址在 0~255 的范围内进行定义。

但是，254（空地址）仅在地址冲突后，无法获取地址时使用。此外，255（全局地址）不指定特定的 CA，在通过广播对所有 CA 发送信息时使用。

设备地址	说明	详情
0 ~ 253	CA 上分配的地址范围	根据连接设备的类型，确立部分地址的用途。有关详情，请参见 SAE J1939 的标准手册。
254	空地址	地址冲突后，在无法获取 0~253 范围的地址时进行设置。
255	全局地址	在通过广播发送信息时使用。

NAME（合计 64 位）

64 位的 NAME 分为多个域。部分域的内容已按照标准事先定义。NAME 需要在同一 J1939 通信网络中设置为不重复的值。各位定义的详情，请参见 SAE J1939 的标准手册。

域名	位长	已定义 SAE*1	项目
Arbitrary Address Capable	1（高位）	—	支持动态地址
Industry Group	3	是	产业集团
Vehicle System Instance	4	—	车辆系统实例
Vehicle System	7	是	车辆系统
Reserved	1	是	已预约（0 固定）
Function	8	是	功能
Function Instance	5	—	功能实例
ECU Instance	3	—	ECU 实例
Manufacturer Code	11	是	制造商代码
Identify Number	21（低位）	—	ID 编号

*1 “是”的项目已按标准定义。请根据您的网络的目的进行设置。

注释：使用的网络不明时，请将各域的最大值设为 NAME。

CAN J1939 All-in-One CPU 模块根据 Arbitrary Address Capable 位的值，解决与 CA 的地址冲突。

Arbitrary Address Capable 位为 0 时

CAN J1939 All-in-One CPU 模块作为分配固定地址的 CA 进行动作。CA 的地址与本机地址重复时，如果 CAN J1939 All-in-One CPU 模块优先级高，则可按照已设置的本地地址收发信息。

如果 CAN J1939 All-in-One CPU 模块的优先级低，则将本地地址设为 254（空地址）。

Arbitrary Address Capable 位为 1 时

CAN J1939 All-in-One CPU 模块作为分配动态地址的 CA 进行动作。CA 的地址与本机地址重复时，如果与相符的 CA 相比，CAN J1939 All-in-One CPU 模块的优先级低，则重复尝试获取其他地址。确定本机地址后，可收发信息。

最后未确定本机地址时，将本机地址设为 254（空地址）。

PGN

CA 接收和发送发动机旋转数和开关 ON/OFF 状态等的参数信息。各参数在事先定义，并分配固有的 ID 编号（SPN）。此外，根据内容汇总多个参数，每个参数上分配固有的 ID 编号（PGN）。

CA 按每个 PGN 收发信息。

例) 构成 PGN 1792 General Purpose Valve Pressure 的 SPN*1

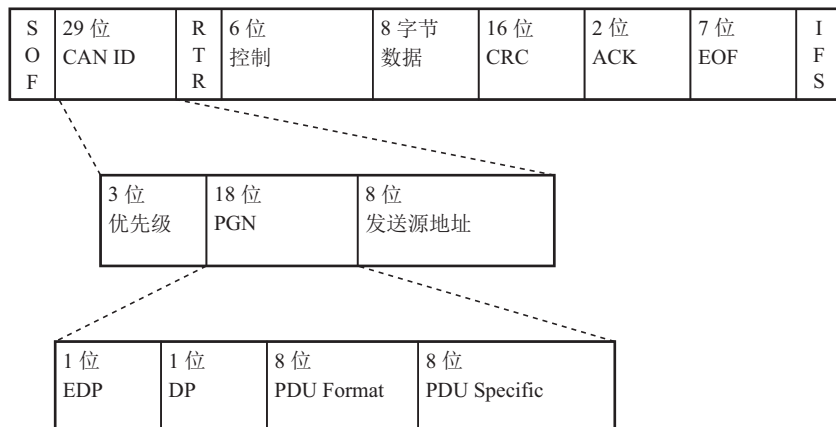
SPN	参数名称	数据大小
4086	Valve Load Sense Pressure	2 字节
4087	Valve Pilot Pressure	1 字节
4088	Valve Assembly Load sense Pressure	2 字节
4089	Valve Assembly Supply Pressure	2 字节

*1 根据 J1939-71

信息

CA 使用带有 29 位 CAN ID 的 CAN 的扩展格式的数据帧，收发信息。CAN ID 由优先级、PGN、发送源地址构成。另外，PGN 由 EDP、DP、PDU Format (PDUF)、PDU Specific (PDUS) 构成。

信息



PDU 1 格式

PDUF 为 00h ~ EFh 时，PGN 按照名为 PDU1 格式的形式进行定义。PGN 按照 EDP、DP、PDUF 进行定义，并作为用于 1 对 1 通信的 PGN 进行处理。此时，在 PDUS 上存储发送源地址。但是，在 PDUS 上指定 255（全局地址）时，不进行 1 对 1 通信，而是作为广播的 PGN 进行处理。

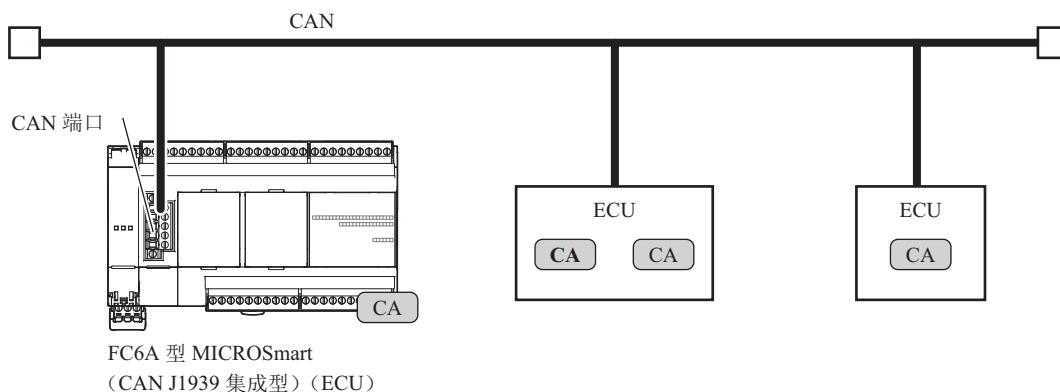
PDU 2 格式

PDUF 为 F0h ~ FFh 时，PGN 按照名为 PDU2 格式的形式进行定义。PGN 按照 EDP、DP、PDUF、PDUS 进行定义，并作为用于广播通信的 PGN 进行处理。

由于 CAN 的数据帧上最大可存储的数据为 8 字节，当 PGN 的数据为 9 字节以上时，将拆分到每个 8 字节的信息中并进行收发。

CAN J1939 All-in-One CPU 模块支持的 J1939 通信的概要

CAN J1939 All-in-One CPU 模块作为带有 1 个 CA 的 ECU，可与 J1939 通信网络进行连接。此外，由于支持 CA 间用于管理地址的网络管理功能，可解决地址冲突和动态地址分配。



8: J1939通信

规格

项目	说明	备注
支持的 SAE J1939	SAE J1939-11: Physical Layer, 250K bits/s, Twisted Shielded Pair SAE J1939-15: Reduced Physical Layer, 250K bits/s, Un-Shielded Twisted Pair SAE J1939-21: Data Link Layer SAE J1939-71: Vehicle Application Layer SAE J1939-73: Application Layer - Diagnostics SAE J1939-75: Application Layer - Generator Sets and Industrial SAE J1939-81: Network Management	
通信速度	250k [bps]	
发送方法	事件发送、周期发送	
接收方法	在梯形图的 END 进行接收和周期监视	
最大节点数	128 个节点	可管理地址的 CA 数
发送信息的最大数	100 个	
接收信息的最大数	200 个	
发送信息的最大长度	252 字节	
接收信息的最大长度	252 字节	
网络管理	启用	解决地址冲突、监视周边 CA 的地址信息等

通过 CAN J1939 All-in-One CPU 模块处理的特殊 PGN

部分 PGN 在 FC6A 型内部用于通信控制。这些 PGN 即使设置为接收信息也无法接收。

无法设置为接收信息的 PGN

PGN	参数名称	说明
59392 (E800h)	Acknowledgement	用于 CA 间的响应。
59904 (EA00h)	Request	用于请求发送 PGN。
60160 (EB00h)	Data Transfer Message	用于拆分 9 字节以上的信息并进行收发。
60416 (EC00h)	Connection Management Message	
60928 (EE00h)	Address Claim	用于网络内的地址管理。

通信控制

可使用特殊内部继电器和特殊数据寄存器，启用 / 禁用 J1939 通信和确认通信状态。

注释：读 / 写为读取 / 写入的缩写。读 / 写表示可以读取和写入。读表示只能读取。写表示只能写入。

特殊内部继电器分配

特殊内部继电器	说明	设置时间	读 / 写
M8300	J1939 通信许可标记		读 / 写
M8301	J1939 联机状态	每次扫描	读
M8302	J1939 本机地址确认状态	每次扫描	读
M8303	发生 J1939 通信错误输出	每次扫描	读
M8304	发生 J1939 通信总线关闭输出	每次扫描	读

J1939 通信许可标记 (M8300)

控制 J1939 通信的启用 / 禁用。打开 M8300，进行初始化处理后即开始 J1939 通信。

关闭：禁用通信（默认）

打开：启用通信

注释：任何 CA 均未对初始化处理所发送的信息进行响应时，将反复重试发送，并发生错误被动（错误代码 2002）。请在网络上的 CA 变为可收发状态后，打开 M8300。

J1939 联机状态 (M8301)

表示 J1939 通信的联机状态。脱机时，由于未连接到 J1939 通信网络，无法收发信息。

关闭：脱机

打开：联机

J1939 本机地址确认状态 (M8302)

显示 J1939 通信时的本机地址确认状态。本机地址在 0~253 时打开。联机时，在通信开始后或发生地址冲突后，本机地址变为 254（空地址）时关闭。M8302 关闭时，无法收发使用梯形图程序的信息。

关闭：未确定本机地址

打开：确定本机地址

发生 J1939 通信错误输出 (M8303)

表示 J1939 通信错误的发生状态。D8052（J1939 通信错误代码）上存储了“0”以外的值时，M8303 打开。

关闭：未发生通信错误

打开：正在发生通信错误

发生 J1939 通信总线关闭输出 (M8304)

表示 J1939 通信时的总线关闭状态。总线关闭为无法参与总线通信的状态。禁止收发全部动作。CAN J1939 All-in-One CPU 模块在内部带有发送错误计数器和接收错误计数器，该计数器达到一定数量时，进入总线关闭状态。

发生总线关闭时，CAN J1939 All-in-One CPU 模块停止通信并脱机。重新通信时，需要将 M8300 从关闭变为打开，并恢复到联机状态。M8300 从关闭变为打开时，M8304 关闭。

关闭：未发生总线关闭

打开：正在发生总线关闭

特殊数据寄存器分配

特殊数据寄存器	功能	设置时间	读 / 写
D8052	J1939 通信错误代码	发生错误时	读 / 写

J1939 通信错误代码 (D8052)

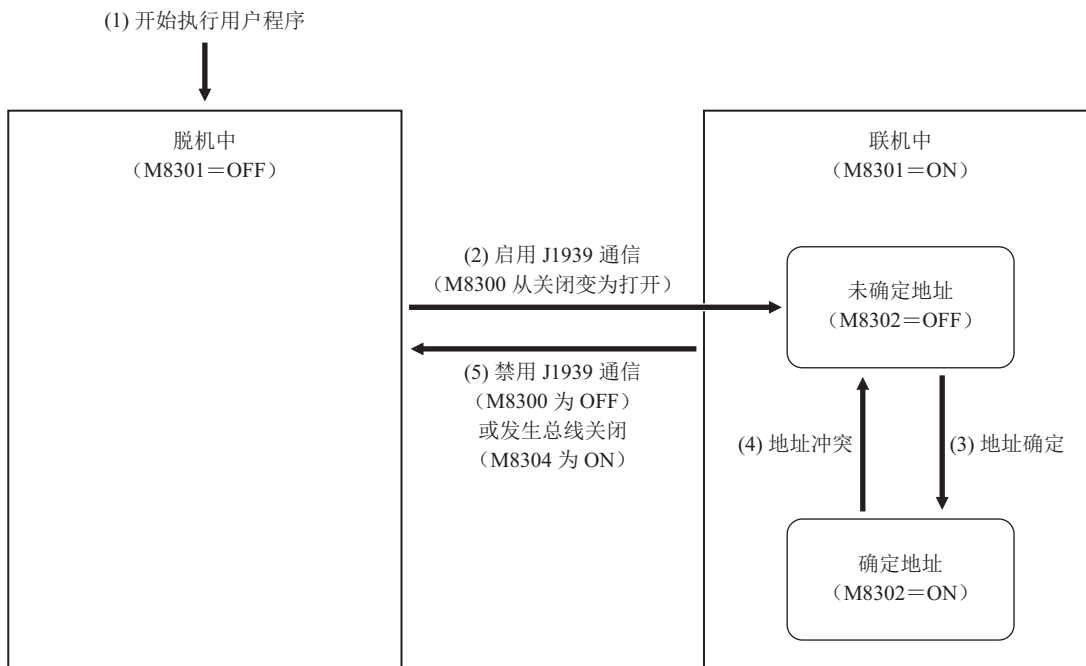
存储 J1939 通信错误代码。初始化时，请写入“0”。

J1939 通信错误代码的详情，如下所示。

状态代码	说明	详情
0	正常	
100	拆分收发错误	接收了非预期的 BAM 帧。
101	拆分收发错误	接收了非预期的 RTS 帧。
102	拆分收发错误	接收了非预期的 CTS 帧。
103	拆分收发错误	接收了非预期的 EOM 帧。
104	拆分收发错误	接收了非预期的 Connection Abort 帧。
105	拆分收发错误	接收了非预期的 DT 帧。
106	拆分收发错误	拆分接收信息的数据长度超出范围。
107	拆分收发错误	Connection Abort 信息发送失败。
110	NACK 发送失败	针对 PGN 的发送请求，无法发送 NACK。
200	内部发送队列上溢出	超出一次能够发送信息数的上限。
201	内部接收队列上溢出	超出一次能够接收信息数的上限。
1000	未确定本机地址	本机地址变成 254。
2000	CAN 通信初始化错误	无法初始化 CAN 控制器。
2001	CAN 通信初始化错误	无法复位 CAN 控制器。
2002	错误被动	发生错误 / 警告中断，CAN 控制器变成错误被动状态。
2003	错误主动	CAN 控制器变成错误主动状态。
2004	数据溢出错误	发生数据溢出中断。
3000	总线关闭错误	CAN 发生总线关闭。

通信控制的状态迁移

J1939 通信的通信状态和初始化处理迁移如下所示。



(1) 开始执行用户程序后，立即变成脱机状态。

(2) M8300 从关闭变为打开时，进入联机状态，执行初始化处理。

初始化处理会发送初始化用信息并确认地址是否重复，如果重复，则会解决冲突。

初始化过程中，在确认本机地址之前为地址未确认状态（M8302 = OFF）。

(3) 本机地址确定为 0 ~ 253 后，变成地址确认状态（M8302 = ON）。

可在 FC6A 型与 CAN 之间收发信息。本机地址为 254 时，一直保持地址未确认状态。

(4) 发生地址冲突，本机地址变成 254（空地址）时，变成地址未确认状态。

(5) M8300 变为关闭时，先停止通信处理，再变成脱机状态。

此外，通信过程中发生总线关闭（M8304 为 ON）时也会变成脱机状态。此时，M8300 保持 OFF 不变。

再次返回联机状态时，请先关闭 M8300，再重新打开。

设置 J1939 通信

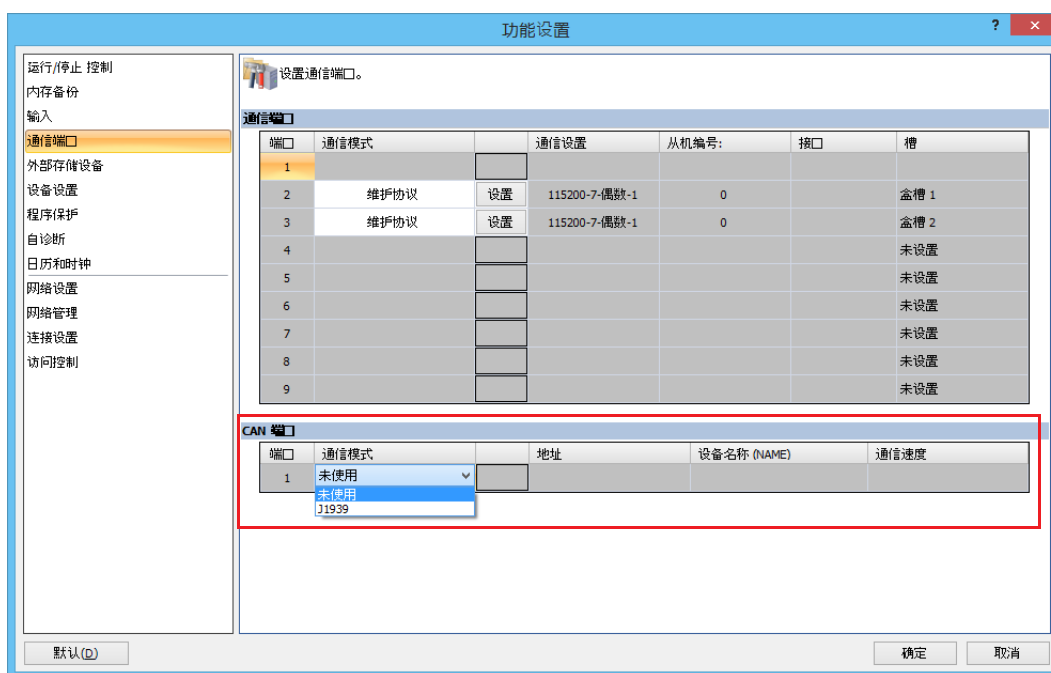
本节将对 J1939 通信相关的操作过程及各项目的详情进行介绍。

设置 WindLDR

按照标准操作条件设置 J1939 通信。

●操作过程

1. 从 WindLDR 菜单栏中, 选择“设置”>“功能设置”>“通信端口”。
出现“功能设置”对话框。
2. 从 CAN 端口组的通信模式中选择“J1939”。
将显示“J1939 通信设置”对话框。

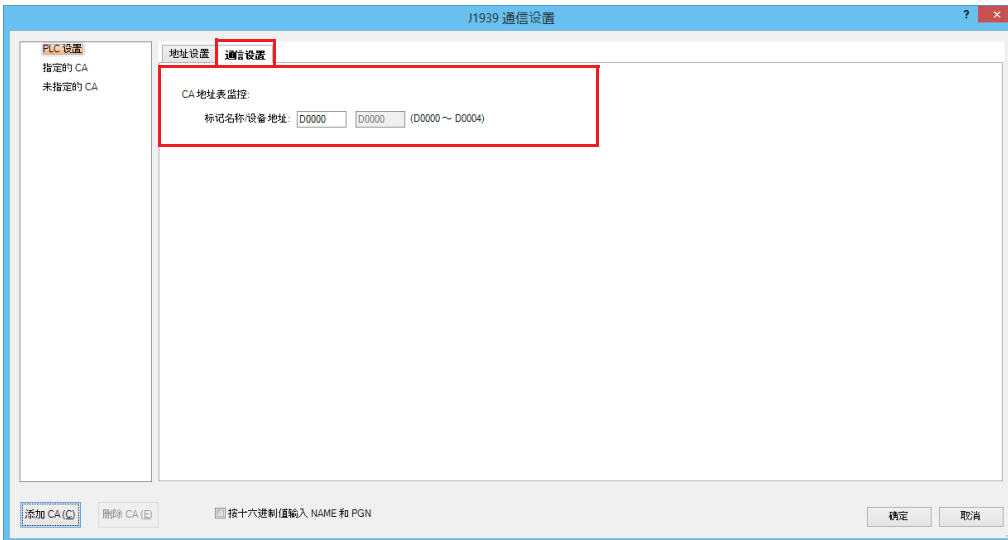


3. 在“PLC 设置”的“地址设置”选项卡中设置本机的地址及设备名称 (NAME)、CA Address Table。
设置本机的地址及设备名称 (NAME)。



8: J1939通信

4. 单击“通信设置”选项卡，设置存储 CA Address Table 的设备地址。



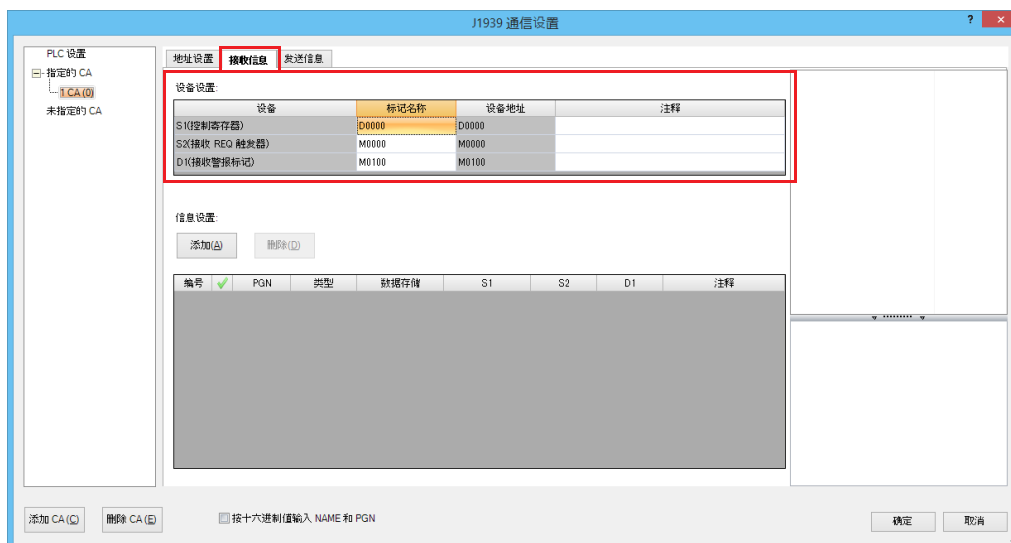
5. 单击“添加 CA”按钮，添加设置各 CA 的节点。



6. 单击要设置 CA 的对应节点，单击“地址设置”选项卡。设置各 CA 的地址、设备名称。



7. 单击“接收信息”选项卡，设置接收信息最先使用的设备。



8. 单击“添加”按钮，添加接收信息。



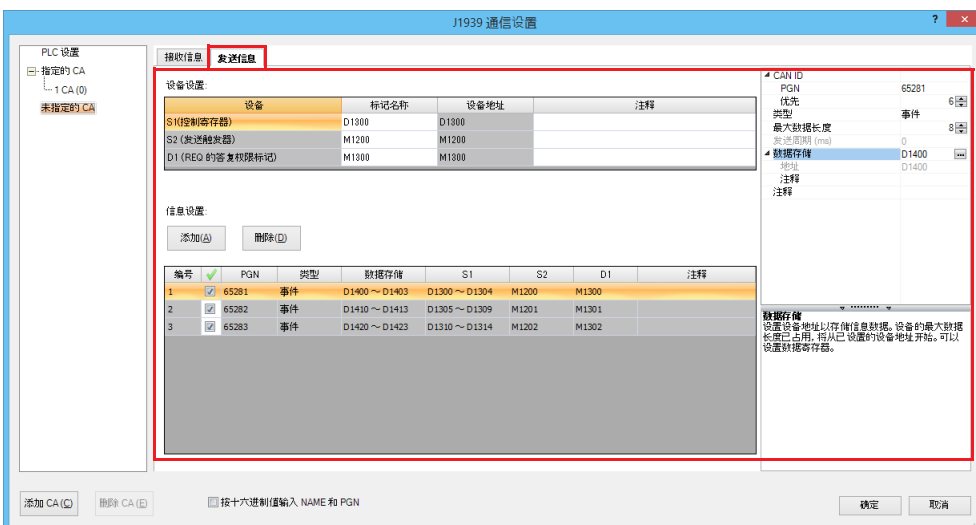
9. 选择信息设置中的任意单元格，会在右侧显示各信息的参数，因此可设置各项目。



10. 单击“发送信息”选项卡，进行与“接收信息”选项卡相同的设置。



11. 单击“未指定的 CA”。进行与“指定的 CA”的“接收信息”选项卡、“发送信息”选项卡相同的设置。



12. 单击“确定”。
 配置即完成。

设置 J1939 通信参数

本节将对使用 J1939 通信所需的“J1939 设置”对话框的各参数设置进行介绍。

PLC 设置

设置 CAN J1939 All-in-One CPU 模块的本机地址、本机 NAME 及 CA Address Table。

■ “地址设置”选项卡

设置 CAN J1939 All-in-One CPU 模块的本机地址和本机 NAME。



(1) 地址

在 0 ~ 253 的范围内设置 CAN J1939 All-in-One CPU 模块的本机地址初始值。

选中“按十六进制值输入 NAME 和 PGN”对话框时，以十六进制格式输入。

例) 十六进制时: 01、F0 等

与连接到同一网络的 CA 之间发生地址冲突时，根据本机 NAME 的 Arbitrary Address Capable 位重新获取本机地址。

(2) 设备名称 (NAME)

以 64 位设置 CAN J1939 All-in-One CPU 模块的本机 NAME。

选中“按十六进制值输入 NAME 和 PGN”对话框时，以十六进制格式输入。

例) 十六进制时: 123456789ABCDEF1、0000FFFFFFFFFOA 等

请设置与同一网络中其它 CA 不重复的值。有关详情，请参见第 8-2 页上的“NAME (合计 64 位)”。

设备名称 (NAME) 由设备名称域一览 (3) 的设置值构成。更改设备名称 (NAME) 后会反映到设备名称域一览的“设置值”中。

(3) 设备名称域一览

设备名称 (NAME) (2) 分域显示。更改“设置值”后会反映到设备名称 (NAME) 中。

■ “通信设置”选项卡

设置 CA Address Table 监控。CA Address Table 监控是对通过 CAN J1939 All-in-One CPU 模块（本机）及“指定的 CA”注册的 CA 的地址进行管理的平台。



(1) CA 地址表监控

设置存储 CA Address Table 的数据寄存器的起始地址。

通过 WindLDR 设置的值以初始值的形式存储到 CA Address Table，CAN J1939 All-in-One CPU 模块的地址因地址冲突而变化时，CA Address Table 会相应进行更改。此外，从 CA 接收到地址信息时，与 NAME 一致的 CA 的 Address Table 会更新。CAN J1939 All-in-One CPU 模块基于 CA Address Table 的地址执行信息的收发工作。

CA Address Table 监控的定义

CA	存储目的地	功能	设置内容
本机	起始编号 +0	本机 NAME	存储 CAN J1939 All-in-One CPU 模块的 NAME。
	起始编号 +1		
	起始编号 +2		
	起始编号 +3		
本机地址*1	起始编号 +4	本机地址*1	存储 CAN J1939 All-in-One CPU 模块的地址。联机状态下，地址发生变化时，后面的信息收发会基于新的地址进行处理。
	起始编号 +5	CA#1 NAME*2	存储 CA#1 的 NAME。
起始编号 +6			
起始编号 +7			
起始编号 +8			
CA#1 (5 个字)	起始编号 +9	CA#1 地址*2*3	存储 CA#1 的地址。在已设置 NAME 的情况下，联机状态下中途地址可能会发生变化。中途地址发生变化时，后面的信息收发会基于新的地址进行处理。
⋮	⋮	⋮	
CA#N*3 (5 个字)	起始编号 +5×N	CA#N NAME*2	存储 CA#N 的 NAME。
	起始编号 5×N+1		
	起始编号 +5×N+2		
	起始编号 +5×N+3		
	起始编号 +5×N+4	CA#N 地址*2*3	存储 CA#N 的地址。在已设置 NAME 的情况下，联机状态下中途地址可能会发生变化。中途地址发生变化时，后面的信息收发会基于新的地址进行处理。

*1 因与网络上的 CA 之间地址冲突而未分配地址时，变成 254（空地址）。此时，无法收发信息。

*2 地址不在 0 ~ 253 范围内时，无法与相应 CA 之间收发信息。

*3 N 为 1 ~ 253。

例 1) 本机地址因冲突而更新时

CA Address Table			
本机 NAME	+0 个字	9234...5678h	*Arbitrary Address Capable = 1 →更新成 128
本机 地址	+4 个字	30	
CA#1 NAME	+5 个字	1345...6789h	
CA#1 地址	+9 个字	150	
		⋮	
		⋮	
		⋮	

例 2) 接收 CA 地址信息后更新时

CA Address Table			CA 地址信息	
本机 NAME	+0 个字	2233...5678h	A844...1111h	NAME
本机 地址	+4 个字	128	130	设备地址
CA#1 NAME	+5 个字	2444...6789h		
CA#1 地址	+9 个字	150		
		⋮		
		⋮		
		⋮		
CA#4 NAME	+20 个字	A844...1111h		
CA#4 地址	+24 个字	50		
		⋮		
		⋮		

更新 NAME 一致的 CA 的地址
→更新成 130

例 3) 例 2 中 CA#4 的 NAME 设置了 0 时

CA Address Table			CA 地址信息	
本机 NAME	+0 个字	2233...5678h	A844...1111h	NAME
本机 地址	+4 个字	128	130	设备地址
CA#1 NAME	+5 个字	2444...6789h		
CA#1 地址	+9 个字	150		
		⋮		
		⋮		
		⋮		
CA#4 NAME	+20 个字	0h		
CA#4 地址	+24 个字	50		
		⋮		
		⋮		

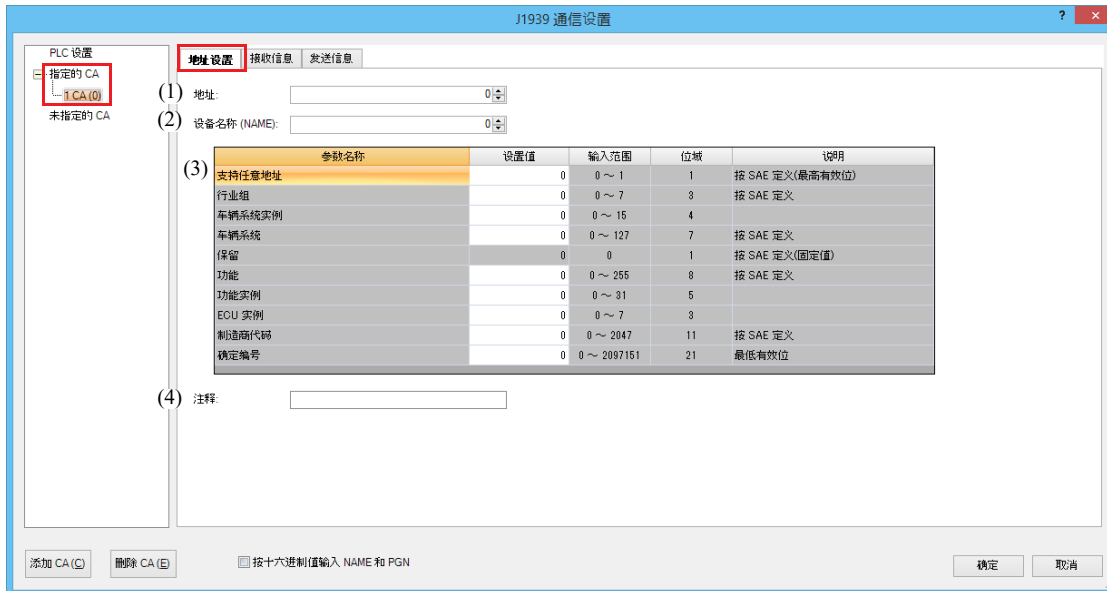
CA Address Table 中没有 NAME, 因此,
不更新 CA Address Table
不更新

指定的 CA

设置各个已添加 CA 的收发信息的 CA 地址、NAME 以及收发信息。

■ “地址设置”选项卡

设置通过 CA 一览选中的 CA 的地址和 NAME。



(1) 地址

在 0 ~ 254 的范围内设置收发信息的 CA 的地址。

选中“按十六进制值输入 NAME 和 PGN”对话框时，以十六进制格式输入。

例) 十六进制时: 01、F0 等

(2) 设备名称 (NAME)

以 64 位设置 CA 的 NAME。CA 的 NAME 未定义时，请设置 0。

选中“按十六进制值输入 NAME 和 PGN”对话框时，以十六进制格式输入。

例) 十六进制时: 123456789ABCDEF1、0000FFFFFFFFF0A 等

设备名称 (NAME) 由设备名称域一览 (3) 的设置值构成。更改设备名称 (NAME) 后会反映到设备名称域一览的“设置值”中。

(3) 设备名称域一览

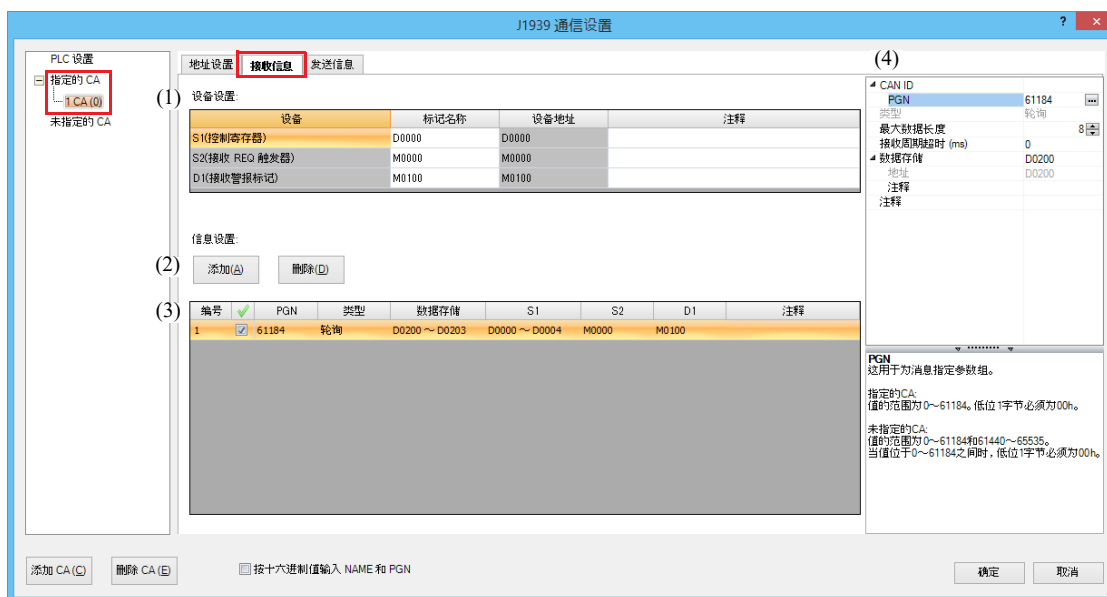
设备名称 (NAME) (2) 分域显示。更改“设置值”后会反映到设备名称 (NAME) 中。

(4) 注释

CA 注释最多可设置 127 个字符。反映到“指定的 CA”的 CA 一览。

■ “接收信息”选项卡

CAN J1939 All-in-One CPU 模块可设置从通过 CA 一览选中的 CA 接收的信息。



(1) 设备设置

设置接收信息所使用的数据寄存器。

设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

S1（控制寄存器）

设置存储接收结果的数据寄存器的起始地址。

存储目的地	项目	说明	
起始编号 +0	接收数据长度	存储已接收信息的数据长度。 接收大于 8 字节的信息时，将接收按 8 字节拆分后的信息。接收完成后，将拆分后的数据进行合并，存储到数据存储目的地。 接收到的信息的数据长度大于最大数据长度时，存储最大数据长度，并在接收结果状态中存储 100。	
起始编号 +1	保留		
起始编号 +2	接收结果状态	存储接收结果。执行初始化时，请写入 0。 有关接收结果状态的详情，请参见第 8-16 页上的“接收结果状态”。	
起始编号 +3	接收动作标记		
	位 0	接收缓存写入标记	将内部接收缓存中存储的信息的数据写入数据存储目的地之前，因以下信息而改写接收缓存时，只启动 1 次扫描。
	位 1 ~ 15	保留	
起始编号 +4	保留		

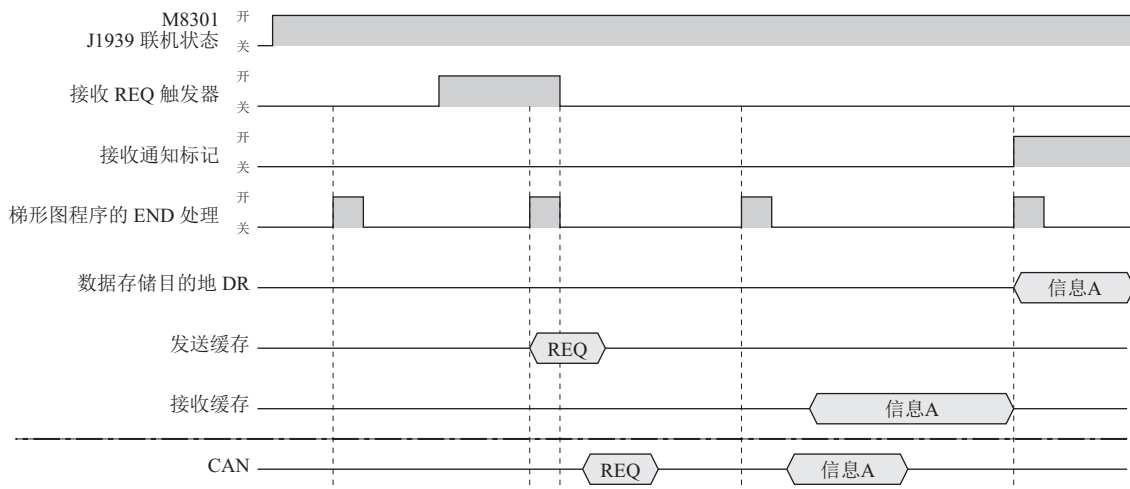
接收结果状态

状态代码	说明	详情
0	正常	
10	接收周期超时	距离上次接收的时间已超过接收周期超时的设置时间。
100	接收数据长度错误	接收到的信息的数据长度也大于最大数据长度。
101	接收数据长度错误	接收到的信息大于 252 字节。
102	拆分接收错误	无法接收数据长度大于 9 字节的信息。

S2（接收 REQ 触发器）

该功能可向对应 CA 请求发送信息。开启接收 REQ 触发器后，会通过梯形图程序的 END 处理向 CA 发送 PGN 59904（Request）。

接收 REQ 触发器的使用示例



D1（接收警报标记）

设置通知信息接收的标记。接收信息时，只开启 1 次扫描。

(2) “添加”按钮 / “删除”按钮

单击“添加”按钮，向信息一览 (3) 添加接收信息。

删除时，在信息一览中选中要删除的信息，单击“删除”按钮。

(3) 信息一览

显示信息参数设置 (4) 中的设置内容。此外，设置接收信息的启用 / 禁用。选中复选框时为启用，取消时为禁用。

注释：

单击复选框列的标题，会执行以下动作。

- 选中各接收信息的全部复选框，则全部变成禁用。
- 取消其中任意一个接收信息的复选框，则全部变成启用。

(4) 信息参数设置

设置通过信息一览选中的接收信息的详情。

CAN ID

PGN: 设置要接收信息的 PGN。可设置的 PGN 的范围为 0 ~ 61,184 (0000h ~ EF00h) (PDU1 格式)。以 PDU1 格式表示发送地址的低位 1 字节 (PDU Specific)，请设置为 00h。

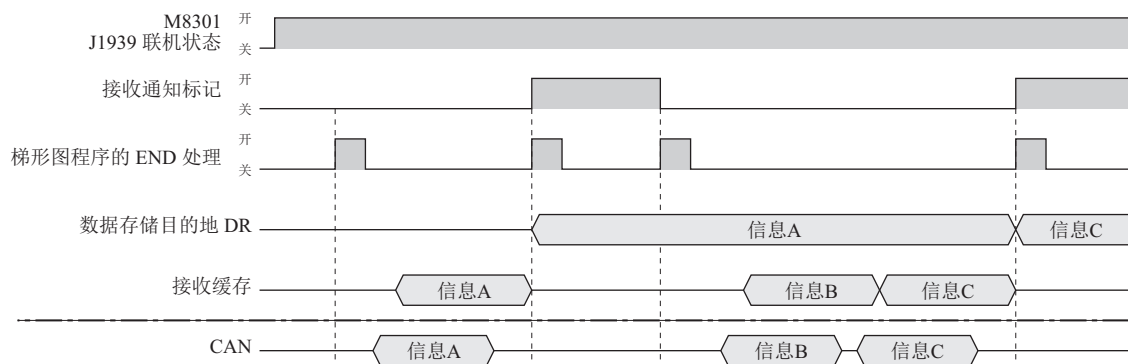
也可以通过单击“...”，弹出 PGN 管理器，再选择 PGN。有关详情，请参见第 8-29 页上的“PGN 管理器”。

类型

信息的接收方法固定为“轮询”。

已接收的信息会暂时保存在内部的接收缓存中，通过梯形图程序的 END 处理将接收缓存中的内容写入设置在数据存储目的地的数据寄存器中。此时，仅 1 次扫描接收通知标记时开启。

在梯形图程序的 END 处理到下一个 END 处理之间，从同一 CA 多次接收同一 PGN 的信息时，将最后一次接收到的信息的内容写入数据存储目的地。

轮询接收示例**最大数据长度**

设置要接收 PGN 的最大数据长度。可设置的最大值为 252 字节。

接收周期超时 (ms)

以 10ms 为单位，设置对呈周期性接收的信息的接收周期进行监控的时间。可设置范围为 0 ~ 655,350。设置为 0 时，将不进行接收周期监视。

数据存储

设置存储接收信息的数据寄存器。以设置的设备为起始，占用最大数据长度。

地址：显示设备的标记名称或设备地址对应的设备地址。

注释：显示设备的标记名称或设备地址对应的注释。可编辑。

注释

设置接收信息的注释。可设置最大 127 字节的注释。

8: J1939通信

■ “发送信息”选项卡

设置 CAN J1939 All-in-One CPU 模块向通过 CA 一览选中的 CA 发送的信息。



(1) 设备设置

设置发送信息所使用的数据寄存器。

设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

S1（控制寄存器）

设置存储发送设置及接收结果的数据寄存器的起始地址。

存储目的地	项目	说明	
起始编号 +0	发送数据长度	存储发送信息的数据长度。 以数据存储目的地为起始，向 CA 发送与发送数据长度对应的数据。发送数据长度存储为 0 时，向 CA 发送最大数据长度的数据。 无法设置大于最大数据长度的数据。 发送大于 8 字节的信息时，将发送按 8 字节拆分后的信息。	
起始编号 +1	保留		
起始编号 +2	发送结果状态	存储发送结果。执行初始化时，请写入 0。 有关发送结果状态的详情，请参见第 8-19 页上的“发送结果状态”。	
起始编号 +3	发送动作标记	存储与发送相关的动作标记等。	
	位 0	有 REQ 接收的标记	从 CA 接收到请求 PGN 时开启。发送对应的 PGN 后关闭。
起始编号 +4	位 1 ~ 15	保留	
	保留		

发送结果状态

状态代码	说明	详情
0	正常	
1	拆分发送完成	数据长度大于 9 字节的信息发送完成。
10	发送数据长度错误	发送信息的数据长度也大于最大数据长度。
20	内部发送队列上溢出	超出一次能够预约发送信息数的上限。
100	拆分发送错误	无法发送数据长度大于 9 字节的信息。
200	未确定本机地址	本机地址未确定，因此无法发送。
201	未确定发送目的地地址	发送目的地 CA 的地址未确定，因此无法发送。

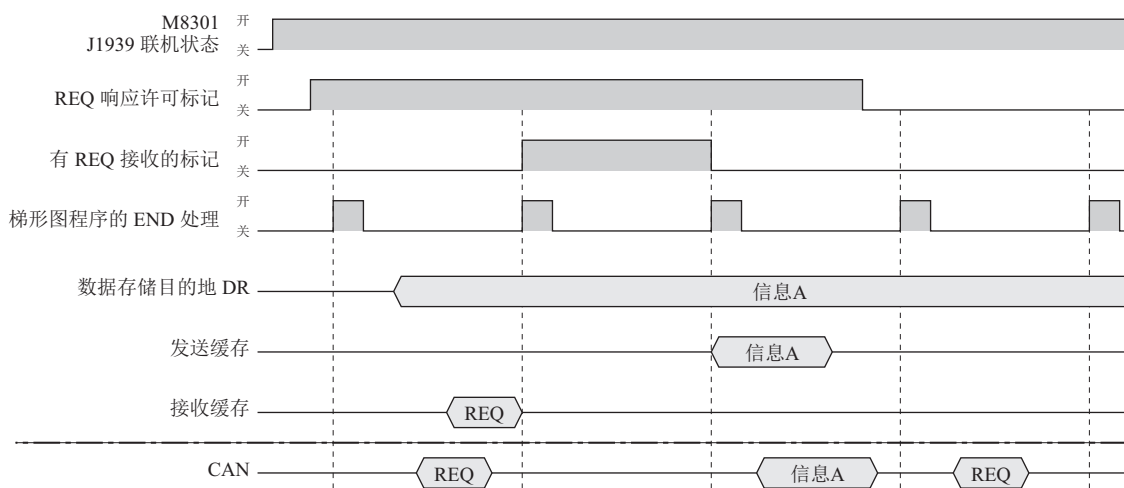
S2（发送触发器）

在“类型”中选择了“事件发送”时，设置发送触发器。开启发送触发器后，会通过梯形图程序的 END 处理发送信息。发送后，发送触发器关闭。

D1（REQ 的答复权限标记）

该功能可在 REQ 响应许可标记为 ON 时，根据从对应 CA 接收到的 PGN 的发送请求发送 PGN。从 CA 接收 PGN59904（Request）时，发送动作标记的有 REQ 接收的标记会打开，通过下一个 END 处理发送相对应的 PGN 信息。

Request 响应示例

**(2) “添加”按钮 / “删除”按钮**

单击“添加”按钮，向信息一览 (3) 添加发送信息的行。

删除时，在信息一览中选中要删除的信息，单击“删除”按钮。

(3) 信息一览

显示信息参数设置 (4) 中的设置内容。此外，设置发送信息的启用 / 禁用。选中复选框时为启用，取消时为禁用。

注释：

单击复选框列的标题，会执行以下动作。

- 选中各发送信息的全部复选框，则全部变成禁用。
- 取消其中任意一个发送信息的复选框，则全部变成启用。

(4) 信息参数设置

设置通过信息一览选中的发送信息的详情。

CAN ID

PGN: 设置要发送信息的 PGN。可设置的 PGN 的范围为 0 ~ 61,184 (0000h ~ EF00h) (PDU1 格式)。以 PDU1 格式表示发送地址的低位 1 字节 (PDU Specific)，请设置为 00h。

也可以通过单击“...”，弹出 PGN 管理器，再选择 PGN。有关详情，请参见第 8-29 页上的“PGN 管理器”。

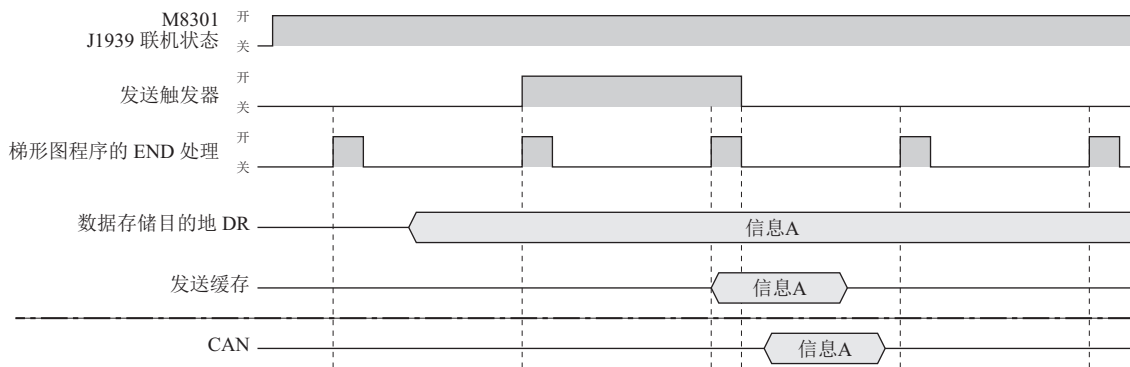
优先: 在 0 ~ 7 的范围内设置发送信息的优先级。

类型

设置信息的发送方法。可选择“事件”或“周期”。

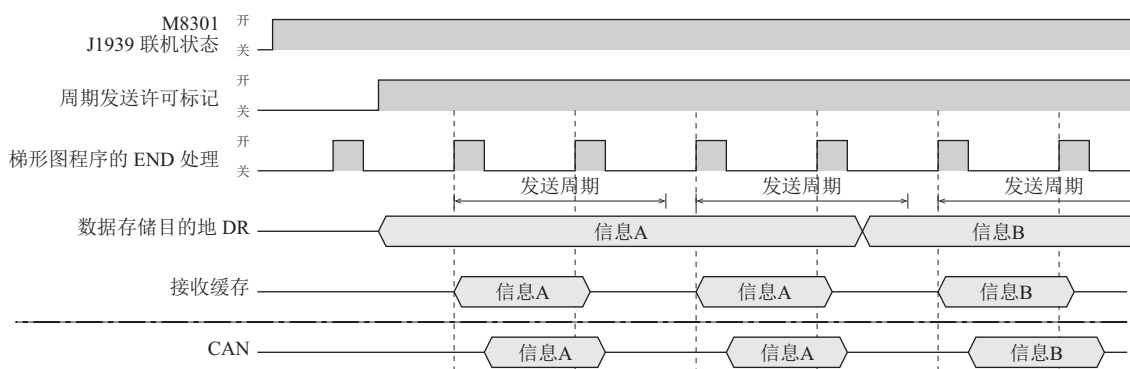
事件: 开启发送触发器时，数据存储目的地中存储的发送数据将通过梯形图程序的 END 处理进行发送。

指定的 CA 时的事件发送示例



周期: 开启周期发送许可标记的情况下，达到发送周期时，数据存储目的地中存储的发送数据将通过梯形图程序的 END 处理进行发送。实际发送的周期，会受到梯形图程序扫描时间的影响。

指定的 CA 时的周期发送示例



最大数据长度

设置要发送 PGN 的最大数据长度。可设置的最大值为 252 字节。

发送周期 (ms)

在“类型”中选择了“周期发送”时，以 10ms 秒为单位设置发送周期。可设置的数值范围为 10 ~ 655,350。

数据存储

设置存储发送信息的数据寄存器。以设置的设备为起始，占用最大数据长度。发送信息时，将数据存储目的地中的数据复制到内部发送缓存后再进行发送。

地址： 显示设备的标记名称或设备地址对应的设备地址。

注释： 显示设备的标记名称或设备地址对应的注释。可编辑。

注释

设置发送信息的注释。可设置最大 127 字节的注释。

8: J1939通信

未指定的 CA

设置不指定特定的 CA，使用广播进行收发信息。

■ “接收信息”选项卡

设置从其他 CA 发送到 J1939 通信网络的以广播形式发送的信息。



(1) 设备设置

设置接收信息所使用的数据寄存器。

设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

S1（控制寄存器）

设置存储接收结果的数据寄存器的起始地址。

存储目的地	项目	说明	
起始编号 +0	接收数据长度	存储已接收信息的数据长度。 接收大于 8 字节的信息时，将接收按 8 字节拆分后的信息。接收完成后，将拆分后的数据进行合并，存储到数据存储目的地。 接收到的信息的数据长度大于最大数据长度时，存储最大数据长度，并在接收结果状态中存储 100。	
起始编号 +1	发送源地址	存储已接收信息的发送源地址。	
起始编号 +2	接收结果状态	存储接收结果。执行初始化时，请写入 0。 有关接收结果状态的详情，请参见第 8-23 页上的“接收结果状态”。	
起始编号 +3	接收动作标记	存储与接收相关的动作标记等。	
	位 0	接收缓存写入标记	将内部接收缓存中存储的信息的数据写入数据存储目的地之前，因以下信息而改写接收缓存时，只启动 1 次扫描。
	位 1 ~ 15	保留	
起始编号 +4	保留		

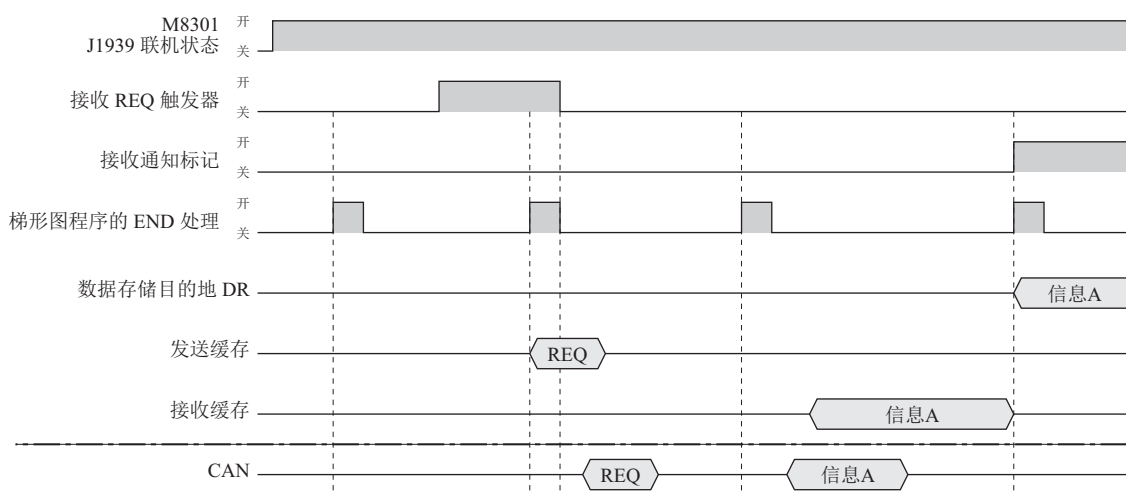
接收结果状态

状态代码	说明	详情
0	正常	
10	接收周期超时	距离上次接收的时间已超过接收周期超时的设置时间。
100	接收数据长度错误	接收到的信息的数据长度也大于最大数据长度。
101	接收数据长度错误	接收到的信息大于 252 字节。
102	拆分接收错误	无法接收数据长度大于 9 字节的信息。

S2 (接收 REQ 触发器)

该功能可在 J1939 通信网络中，以广播的形式进行 PGN 发送请求。开启接收 REQ 触发器后，会以广播的形式，通过梯形图程序的 END 处理发送 PGN 599034 (Request)。

接收 REQ 触发器的使用示例

**D1 (接收警报标记)**

设置通知信息接收的标记。接收信息时，只开启 1 次扫描。

(2) “添加”按钮 / “删除”按钮

单击“添加”按钮，向信息一览 (3) 添加接收信息的行。

删除时，在信息一览中选中要删除的信息，单击“删除”按钮。

(3) 信息一览

显示信息参数设置 (4) 中的设置内容。此外，设置接收信息的启用 / 禁用。选中复选框时为启用，取消时为禁用。

注释：

单击复选框列的标题，会执行以下动作。

- 选中各接收信息的全部复选框，则全部变成禁用。
- 取消其中任意一个接收信息的复选框，则全部变成启用。

(4) 信息参数设置

设置通过信息一览选中的接收信息的详情。

CAN ID

PGN: 以 2 字节设置要接收信息的 PGN。可设置的 PGN 范围为 0 ~ 61,184 (0000h ~ EF00h) (PDU1 格式)、61,440 ~ 65,535 (F000h ~ FFFFh) (PDU2 格式)。以 PDU1 格式表示发送地址的低位 1 字节 (PDU Specific)，请设置为 00h。

在 PDU1 格式范围内设置时，发送地址仅接收 255 (全局地址) 的信息。

也可以通过单击“...”，弹出 PGN 管理器，再选择 PGN。有关详情，请参见第 8-29 页上的“PGN 管理器”。

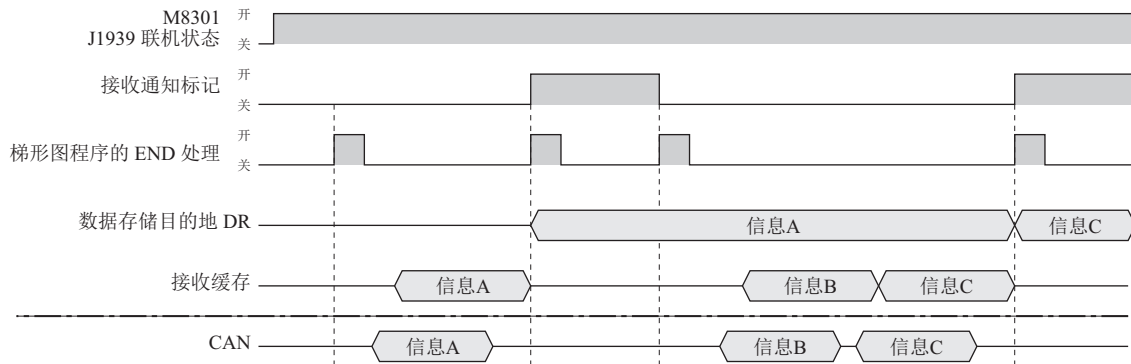
类型

信息的接收方法固定为“轮询”。

已接收的信息会暂时保存在内部的接收缓存中，通过梯形图程序的 END 处理将接收缓存中的内容写入设置在数据存储目的地的数据寄存器中。此时，仅 1 次扫描接收通知标记时开启。

在梯形图程序的 END 处理到下一个 END 处理之间，从同一 CA 多次接收同一 PGN 的信息时，将最后一次接收到的信息的内容写入数据存储目的地。

轮询接收示例



最大数据长度

设置要接收 PGN 的最大数据长度。可设置的最大值为 252 字节。

接收周期超时 (ms)

以 10ms 为单位，设置对呈周期性接收的信息的接收周期进行监控的时间。可设置范围为 0 ~ 655,350。设置为 0 时，将不进行接收周期监视。

数据存储

设置存储接收信息的数据寄存器。以设置的设备为起始，占用最大数据长度。

地址：显示设备的标记名称或设备地址对应的设备地址。

注释：显示设备的标记名称或设备地址对应的注释。可编辑。

注释

设置接收信息的注释。可设置最大 127 字节的注释。

■ “发送信息”选项卡

CAN J1939 All-in-One CPU 模块可设置 J1939 通信网络中以广播形式发送的信息。



(1) 设备设置

设置发送信息所使用的数据寄存器。

设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

S1（控制寄存器）

设置存储发送设置及接收结果的数据寄存器的起始地址。

存储目的地	项目	说明	
起始编号 +0	发送数据长度	存储发送信息的数据长度。 以数据存储目的地为起始，向 CA 发送与发送数据长度对应的数据。发送数据长度存储为 0 时，向 CA 发送最大数据长度的数据。 无法设置大于最大数据长度的数据。 发送大于 8 字节的信息时，将发送按 8 字节拆分后的信息。	
起始编号 +1	Request 响应地址	将所接收的请求 PGN 的发送源地址将作为 PGN 的响应地址进行存储。	
起始编号 +2	发送结果状态	存储发送结果。执行初始化时，请写入 0。 有关发送结果状态的详情，请参见第 8-26 页上的“发送结果状态”。	
起始编号 +3	发送动作标记	存储与发送相关的动作标记等。	
	位 0	有 REQ 接收的标记	从 CA 接收到请求 PGN 时开启。发送对应的 PGN 后关闭。
起始编号 +4	位 1 ~ 15	保留	
	保留		

发送结果状态

状态代码	说明	详情
0	正常	
1	拆分发送完成	数据长度大于 9 字节的信息发送完成。
10	发送数据长度错误	发送信息的数据长度也大于最大数据长度。
20	内部发送队列上溢出	超出一次能够预约发送信息数的上限。
100	拆分发送错误	无法发送数据长度大于 9 字节的信息。
200	未确定本机地址	本机地址未确定，因此无法发送。
201	未确定发送目的地地址	发送目的地 CA 的地址未确定，因此无法发送。

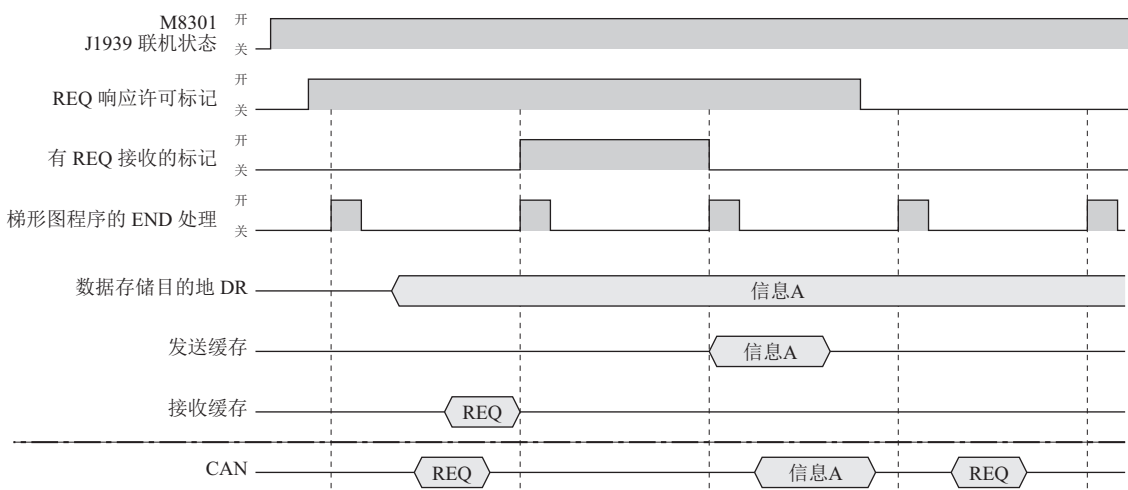
S2（发送触发器）

在“类型”中选择了“事件发送”时，设置发送触发器。开启发送触发器后，会通过梯形图程序的 END 处理发送信息。发送后，发送触发器关闭。

D1（REQ 的答复权限标记）

该功能可响应广播所发送信息的发送请求。在接收广播发送的 PGN 59904（Request）时，若对应 PGN 的 REQ 响应许可标记为开启，则 PGN 59904（Request）的发送源地址将被存储到 Request 响应地址，发送动作标记的有 REQ 接收的标记变为开启。在下次 END 处理后，将对 Request 响应地址发送所请求的 PGN 信息。

Request 响应示例



(2) “添加”按钮 / “删除”按钮

单击“添加”按钮，向信息一览 (3) 添加发送信息的行。
删除时，在信息一览中选中要删除的信息，单击“删除”按钮。

(3) 信息一览

显示信息参数设置 (4) 中的设置内容。此外，设置发送信息的启用 / 禁用。选中复选框时为启用，取消时为禁用。

注释:

单击复选框列的标题，会执行以下动作。

- 选中各发送信息的全部复选框，则全部变成禁用。
- 取消其中任意一个发送信息的复选框，则全部变成启用。

(4) 信息参数设置

设置通过信息一览选中的发送信息的详情。

CAN ID

PGN: 设置要发送信息的 PGN。可设置的 PGN 范围为 0 ~ 61,184 (0000h ~ EF00h) (PDU1 格式)、61,440 ~ 65,535 (F000h ~ FFFFh) (PDU2 格式)。以 PDU1 格式表示发送地址的低位 1 字节 (PDU Specific)，请设置为 00h。在 PDU1 格式范围内设置时，作为发送地址为 255 (全局地址) 的信息进行发送。

也可以通过单击“...”，弹出 PGN 管理器，再选择 PGN。有关详情，请参见第 8-29 页上的“PGN 管理器”。

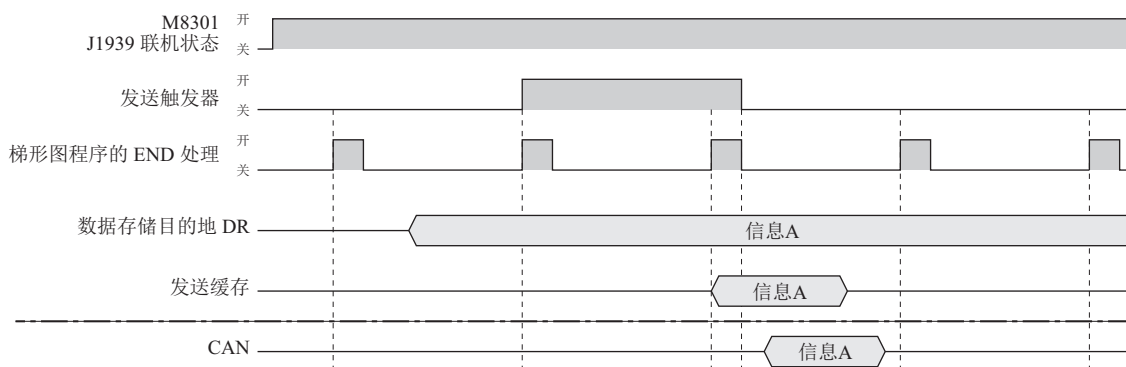
优先: 在 0 ~ 7 的范围内设置发送信息的优先级。

类型

设置信息的发送方法。可选择“事件”或“周期”。

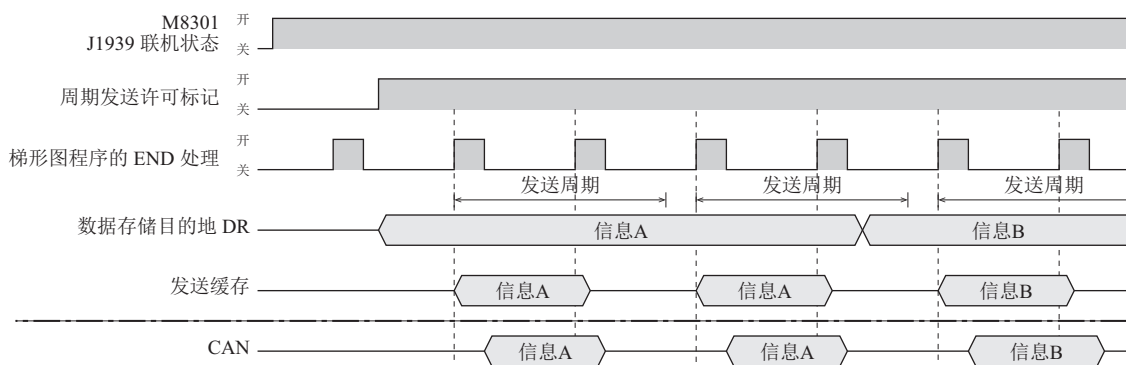
事件: 开启发送触发器时，数据存储目的地中存储的发送数据将通过梯形图程序的 END 处理进行发送。

未指定的 CA 时的事件发送示例



周期: 开启周期发送许可标记的情况下，达到发送周期时，数据存储目的地中存储的发送数据将通过梯形图程序的 END 处理进行发送。实际发送的周期，会受到梯形图程序扫描时间的影响。

未指定的 CA 时的周期发送示例



最大数据长度

设置要发送 PGN 的最大数据长度。可设置的最大值为 252 字节。

发送周期 (ms)

在“类型”中选择了“周期发送”时，以 10ms 秒为单位设置发送周期。可设置的数值范围为 10 ~ 65,350。

8: J1939通信

数据存储

设置存储发送信息的数据寄存器。以设置的设备为起始，占用最大数据长度。发送信息时，将数据存储目的地中的数据复制到内部发送缓存后再进行发送。

地址： 显示设备的标记名称或设备地址对应的设备地址。

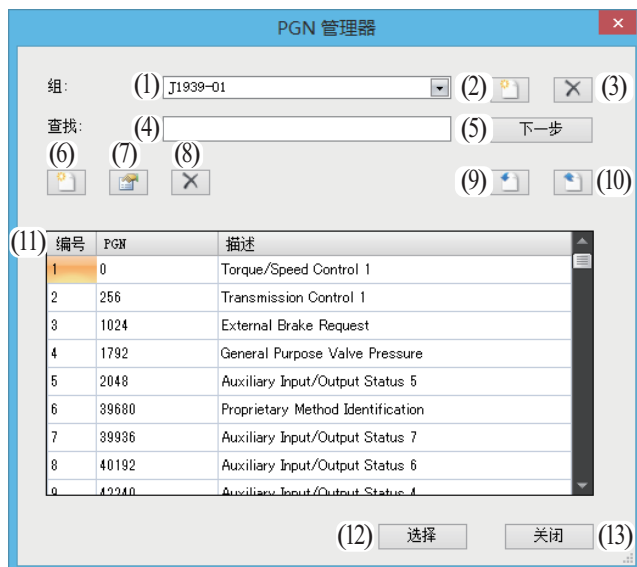
注释： 显示设备的标记名称或设备地址对应的注释。可编辑。

注释

设置发送信息的注释。可设置最大 127 字节的注释。

PGN 管理器

收发信息的 PGN，可通过 PGN 管理器进行统一管理。



(1) 组列表

显示已注册的 PGN 组。

更改组时，注册在所选组中的 PGN 将显示在 PGN 一览 (11) 中。

已事先注册“J1939”。包括“J1939”在内，最多可注册 20 个组。

(2) 组添加按钮

添加组。

单击此按钮，将显示 PGN 组编辑器。使用 PGN 组编辑器添加新组。有关详情，请参见第 8-31 页上的“PGN 组编辑器”。

新增的组，将被添加到组列表中。

(3) 组删除按钮

删除组。

从组列表中选择组，并单击此按钮，即可删除组。

无法删除默认的组“J1939”。

(4) 查找

在 PGN 一览中输入要查找的字符串，最多可输入 128 字节。

(5) “下一步”按钮

在 PGN 一览中，对“查找”中所输入的字符串进行查找。在“查找”中输入字符串，并单击此按钮。

(6) PGN 添加按钮

在组中添加 PGN。

在组列表中选择要添加 PGN 的组，并单击此按钮，将显示 PGN 编辑器。

使用 PGN 编辑器添加新的 PGN。有关详情，请参见第 8-31 页上的“PGN 编辑器”。

新增的 PGN，将被添加到 PGN 一览中。

(7) PGN 编辑按钮

编辑 PGN。

在组列表中选择要编辑的 PGN 所注册的组，并在 PGN 一览中选择要编辑的 PGN。单击此按钮，将显示 PGN 编辑器。

使用 PGN 编辑器编辑 PGN。有关详情，请参见第 8-31 页上的“PGN 编辑器”。

更新 PGN 一览中所选的 PGN 内容。

(8) PGN 删除按钮

删除 PGN。

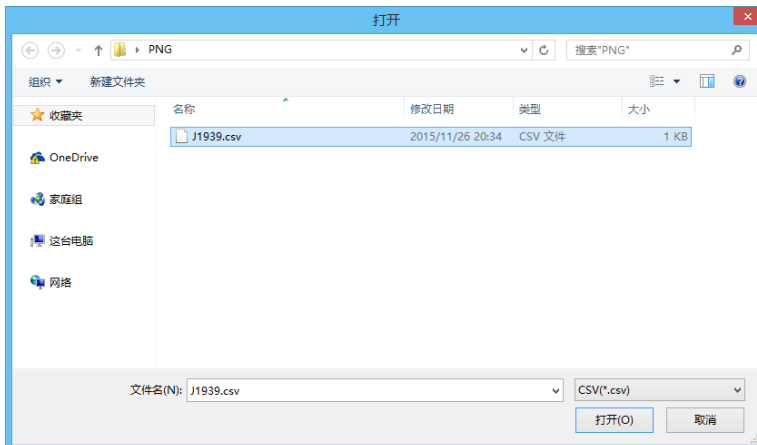
在组列表中选择要删除的 PGN 所注册的组，并在 PGN 一览中选择要删除的 PGN。单击此按钮，PGN 将被删除。

(9) PGN 导入按钮

导入以 CSV 文件格式保存的 PGN。

在组列表中选择要导入 PGN 的组。已被注册在所选组中的 PGN，将在执行导入前被全部删除。

单击此按钮，将显示“打开”对话框。

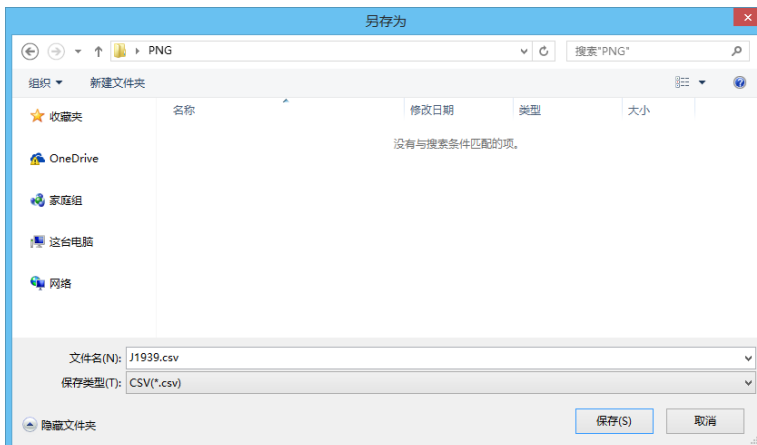


选择保存 PGN 的 CSV 格式文件，单击“打开”按钮，即可将文件内的 PGN 导入到组列表中所选组的 PGN 一览中。

(10) PGN 导出按钮

将注册在组列表中所选组中的 PGN，作为 CSV 格式的文件进行保存。

单击此按钮，将显示“另存为”对话框。



选择要保存的位置，输入文件名并单击“保存”按钮，即可以 CSV 格式保存注册在组中的 PGN。

(11) PGN 一览

显示注册在组列表中所选组中的 PGN。

编号：显示被注册的 PGN 管理编号（1 ~ 65535）。

PGN：显示 PGN。

描述：显示 PGN 的名称。

(12) “选择”按钮

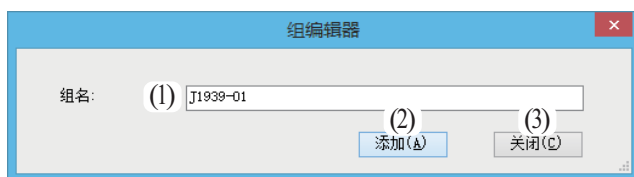
关闭 PGN 管理器，在调用源的信息参数设置“CAN ID”中设置 PGN 一览中所选的 PGN。

(13) “关闭”按钮

关闭 PGN 管理器。

■ PGN 组编辑器

添加新组。



(1) 组名

注册 PGN 的组名称最多可输入 64 字节。

(2) “添加”按钮

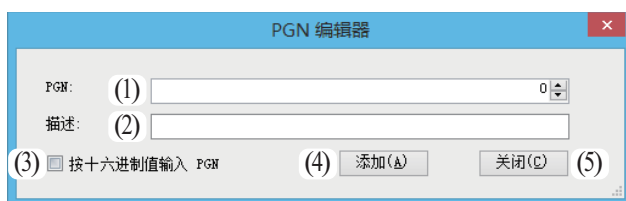
关闭 PGN 组编辑器，并添加到 PGN 管理器的组列表中。

(3) “关闭”按钮

关闭 PGN 组编辑器。

■ PGN 编辑器

添加新的 PGN 或编辑现有的 PGN。



(1) PGN

注册的 PGN 最多可输入 5 个字符。

请在取消“按十六进制值输入 PGN”复选框时以十进制进行输入，选中时以十六进制进行输入。

例) 十六进制时：0001、FF00 等

(2) 描述

注册的 PGN 名称最多可输入 128 字节。

(3) “按十六进制值输入 PGN”复选框

设置 PGN 的输入形式。

以十进制输入时设为关闭，以十六进制输入时设为开启。

(4) “添加”按钮

关闭 PGN 编辑器，并添加到 PGN 管理器的组列表中。

(5) “关闭”按钮

关闭 PGN 编辑器。

9: Bluetooth 通信

本章将对 FC6A 型的 Bluetooth 通信功能进行介绍。

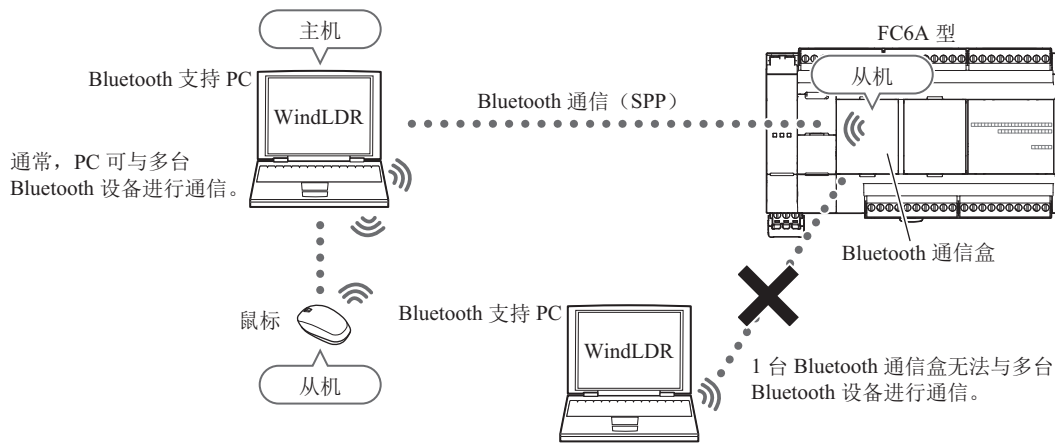
概要

通过将 Bluetooth 通信盒连接到 FC6A 型，即可使用 Bluetooth 通信。可在盒插槽 1、2、及仅 Plus CPU 模块时加上盒插槽 3 的共 3 处连接 Bluetooth 通信盒。Bluetooth 通信盒可与支持 SPP 轮廓的 Bluetooth 设备进行串行通信。

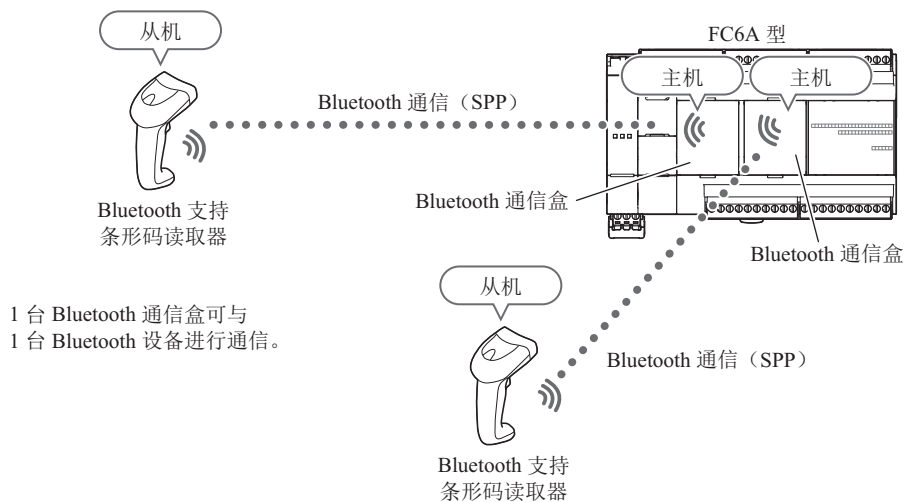
连接 Bluetooth 通信盒的 FC6A 型可作为主机或从机连接到 Bluetooth 网络。即使 FC6A 型为主机或从机的任意一个，对于 1 台 Bluetooth 通信盒，仅可连接及通信 1 台 Bluetooth 设备。

在 Bluetooth 网络中，将连接源称为主机，将连接目标称为从机。

连接 PC 和 FC6A 型的示例



连接 FC6A 型和 Bluetooth 设备的示例



9: BLUETOOTH通信

无线通信规格

型号	FC6A-PC4
Bluetooth 规格	Bluetooth ver.2.1 + EDR
文件夹	SPP (Serial Port Profile) iAP (iPod Accessory Protocol)
频率范围	2,402MHz ~ 2,480MHz
传输距离 *1	10m (Class 2)
多配对台数	8 台
通信协议	维护通信、用户通信
取得 Bluetooth 无线认可国家以及地区 *2	日本、中国、美国、加拿大、澳大利亚、新西兰、欧洲

*1 Bluetooth 通信的连接的有效范围，因障碍物（人体、金属、墙壁等）或电波状态而异。

*2 根据各国家或地区的标准规范，可能需要对装置实施评估。

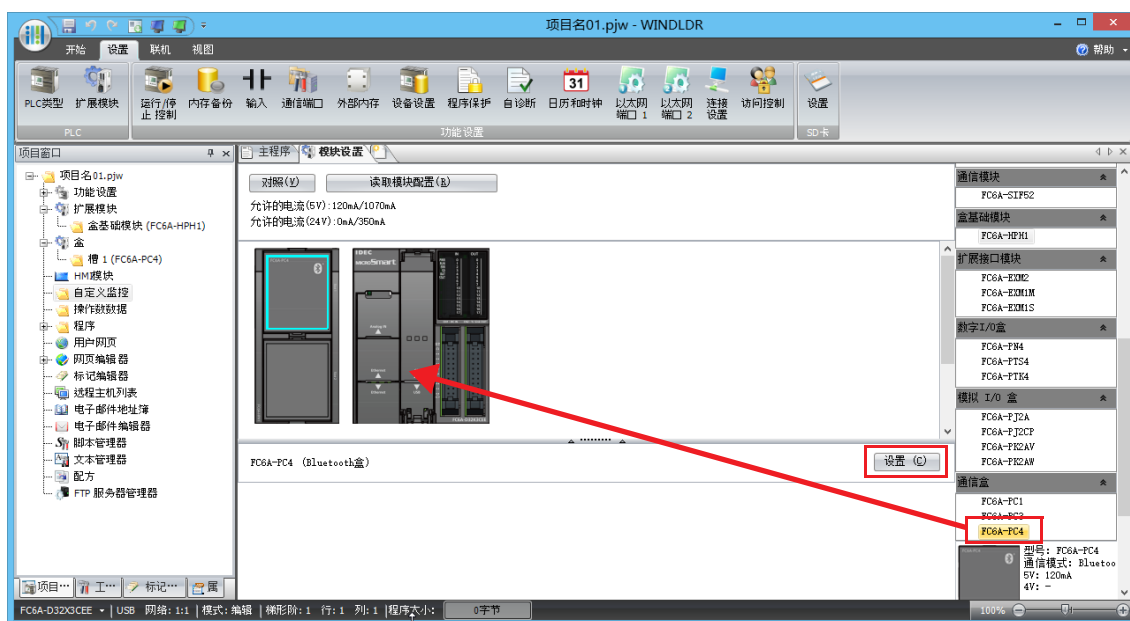
注释：维护通信时的通信性能（所要时间）如下。

- 上转相当于 10,000 步的客户程序：约 2 分 40 秒
- 下载相当于 10,000 步的客户程序：约 2 分 30 秒
- 上转相当于 20,000 步的客户程序：约 5 分 20 秒
- 下载相当于 20,000 步的客户程序：约 5 分
- 读出 100KB 的 CSV 文档：约 1 分 30 秒
- 读出 200KB 的 CSV 文档：约 3 分

Bluetooth 通信盒的设置

设置 WindLDR

1. 设置 Bluetooth 通信盒时，通过模块构成编辑器进行。
从 WindLDR 菜单栏中，选择“设置”>“PLC”>“扩展模块”。
2. 单击模块构成区域中所插入的 Bluetooth 通信盒，并单击“设置”按钮。
将显示“Bluetooth 通信盒设置”对话框。



3. 设置 Bluetooth 通信盒的各设置项目。

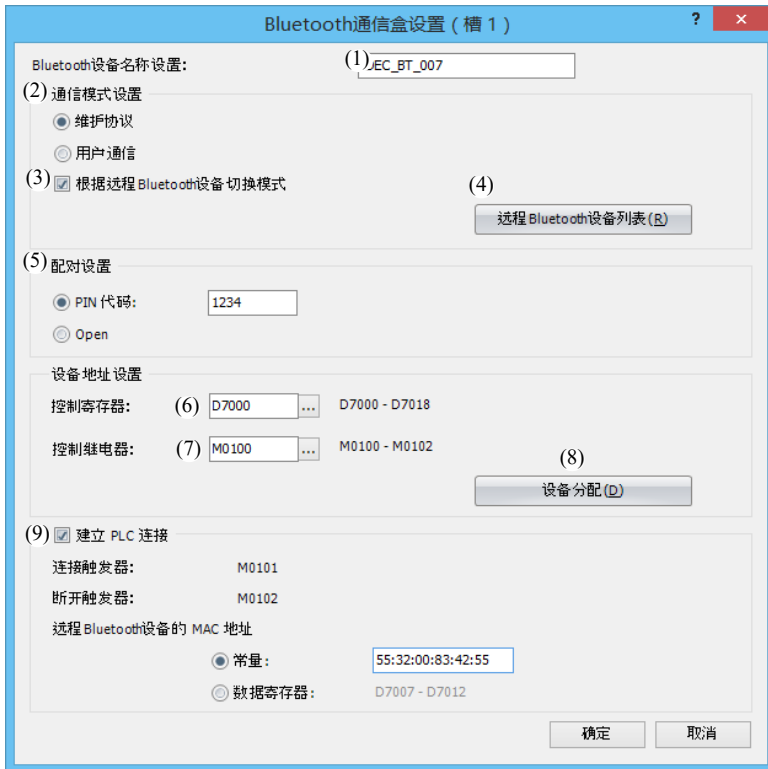


9: BLUETOOTH通信

4. 将用户程序下载到 FC6A 型。

至此，Bluetooth 通信盒的设置完成。

Bluetooth 通信盒设置



(1) Bluetooth 设备名称设置

设置通过 PC 或智能手机搜索周围的 Bluetooth 设备时显示于 PC 或智能手机中的名称。

(2) 通信模式设置

将 Bluetooth 通信时的 FC6A 型的通信模式选择为“维护通信”或“用户通信”。

设置根据远程 Bluetooth 设备切换模式 (3) 时，优先选择根据远程 Bluetooth 设备切换模式 (3) 的设置。

(3) 根据远程 Bluetooth 设备切换模式

根据连接的 Bluetooth 设备切换 FC6A 型的通信模式时，应选中该复选框。

例如，连接 FC6A 型和智能手机时想要设为维护通信，而切断智能手机后，在连接 FC6A 型和条形码读取器时想要设为用户通信时使用。

取消该复选框、或即使选中仍在“远程 Bluetooth 设备列表”对话框中未登录连接目标的 Bluetooth 设备时，FC6A 型会在通信模式设置 (2) 中所设置的通信模式下运行。

设置根据远程 Bluetooth 设备切换模式 (3) 时，FC6A 型会在开始连接 Bluetooth 设备时判断通信模式。因此，FC6A 型在开始连接 Bluetooth 后数秒内无法接收 SPP 数据。已切换通信模式时，如果更改控制寄存器 (6) 的 Bluetooth 通信盒（盒插槽）的状态，即可进行 SPP 通信。

(4) “远程 Bluetooth 设备列表”按钮

在根据远程 Bluetooth 设备切换模式 (3) 中使用的连接目标 Bluetooth 设备的 MAC 地址和切换的通信模式。

单击此按钮，将显示“远程 Bluetooth 设备列表”对话框。

对话框中将显示各 Bluetooth 设备的 MAC 地址、通信模式、注释。有关详情，请参见第 9-6 页上的“远程 Bluetooth 设备列表”。

(5) 配对设置

从以下选项中选择连接 Bluetooth 时的安全措施。

PIN 代码 : 与 Bluetooth 设备连接时, 将要求输入已设置的 PIN 码。加密通信内容。
一般为最高 Bluetooth v2.0 版本的设备所采用的配对方式。

Open : 不加密通信内容。
最高 Bluetooth v2.0 版本的设备在不具备安全措施时所使用的设置。

(6) 控制寄存器

设置存储连接 FC6A 型的 Bluetooth 通信盒或连接目标的 Bluetooth 设备 MAC 地址的数据寄存器起始地址。
无法设定特殊数据寄存器。

设备地址	项目	说明	读 / 写
起始编号 +0	Bluetooth 通信盒 (盒插槽) 的状态	存储 Bluetooth 通信的状态和通信模式。 0: 空操作 255: 处理中 (等待连接 / 切断 Bluetooth 等) 1: 维护通信中 2: 用户通信中 默认值为 0 (空操作)。	读
起始编号 +1	Bluetooth 通信盒 (盒插槽) 的 MAC 地址	存储连接盒插槽的 Bluetooth 通信盒的 MAC 地址。 例) MAC 地址: AA-BB-CC-DD-EE-FF 起始编号 + 1 = AA、起始编号 + 2 = BB、起始编号 + 3 = CC、 起始编号 + 4 = DD、起始编号 + 5 = EE、起始编号 + 6 = FF	读
起始编号 +2			
起始编号 +3			
起始编号 +4			
起始编号 +5			
起始编号 +6	Bluetooth 通信盒 (盒插槽) 的连接目标的 Bluetooth 设备 MAC 地址	存储连接中的 Bluetooth 设备 MAC 地址。 例) MAC 地址: AA-BB-CC-DD-EE-FF 起始编号 + 1 = AA、起始编号 + 2 = BB、起始编号 + 3 = CC、 起始编号 + 4 = DD、起始编号 + 5 = EE、起始编号 + 6 = FF	读 / 写
起始编号 +7			
起始编号 +8			
起始编号 +9			
起始编号 +10			
起始编号 +11	— 保留 —		—
起始编号 +12			
起始编号 +13			
起始编号 +14			
起始编号 +15			
起始编号 +16			
起始编号 +17			
起始编号 +18			

(7) 控制继电器

设置 Bluetooth 通信盒 (盒插槽) 的通信状态、存储建立 PLC 连接 (9) 的连接触发器和切开触发器的内部继电器起始地址。无法设定特殊内部继电器。

设备地址	项目	说明	读 / 写
起始编号 +0	Bluetooth 通信盒 (盒插槽) 的状态	0: 无法与连接目标 Bluetooth 设备进行通信的状态 1: 可与连接目标 Bluetooth 设备进行通信的状态	读
起始编号 +1	Bluetooth 通信盒 (盒插槽) 的连接触发器	0: 不处理 1: 连接连接对象 Bluetooth 设备	读 / 写
起始编号 +2	Bluetooth 通信盒 (盒插槽) 的切开触发器	0: 不处理 1: 切断连接对象 Bluetooth 设备	读 / 写

(8) “设备分配”按钮

单击此按钮，将显示“设备分配”对话框。

对话框中将显示 Bluetooth 通信盒的各设置内容和数据寄存器、内部继电器的对应表。有关详情，请参见第 9-7 页上的“设备分配”。

(9) 建立 PLC 连接

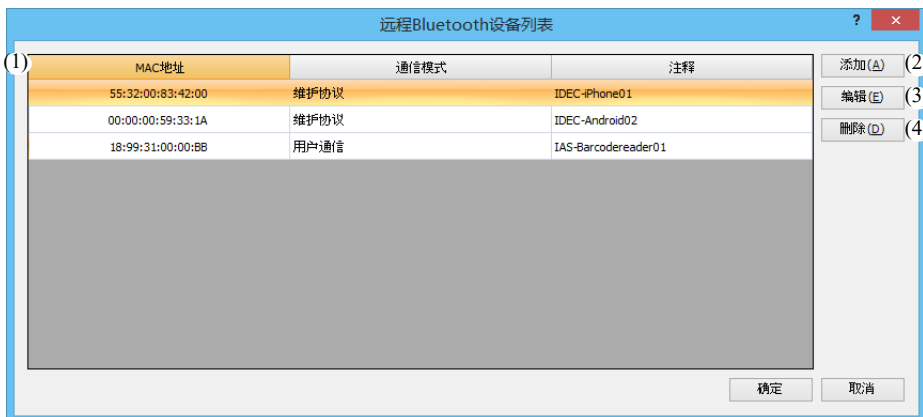
FC6A 型为主机时，如果连接及切断连接对象 Bluetooth 设备，应选中该复选框。选中复选框时，无法连接 iOS 设备。

连接触发器：将显示控制继电器 (7) 中设置的设备地址 + 1 的连接触发器。从关闭变为打开时，连接“远程 Bluetooth 设备的 MAC 地址”中设置的连接对象 Bluetooth 设备。

切断触发器：将显示控制继电器 (7) 中设置的设备地址 + 2 的切断触发器。从关闭变为打开时，切断“远程 Bluetooth 设备的 MAC 地址”中设置的连接对象 Bluetooth 设备。

远程 Bluetooth 设备的 MAC 地址：将连接对象选择为“常量”或“数据寄存器”。
 选择了“常量”时，“Bluetooth 通信盒设置”对话框中设置的 MAC 地址变为连接对象。
 选择了“数据寄存器”时，控制设备 (6) 的起始编号 + 7 ~ + 12 中设置的 Bluetooth 设备的 MAC 地址变为连接对象。

远程 Bluetooth 设备列表



(1) Bluetooth 设备一览

将显示各 Bluetooth 设备的 MAC 地址、通信模式、注释。

(2) 添加

单击此按钮，将显示“远程 Bluetooth 设备设置”对话框。输入连接目标 Bluetooth 设备的 MAC 地址、开始连接时切换的通信模式、注释。

最多可注册 20 台。有关详情，请参见第 9-7 页上的“远程 Bluetooth 设备设置”。

(3) 编辑

在 Bluetooth 设备一览 (1) 中选择 Bluetooth 设备，单击此按钮，将显示“远程 Bluetooth 设备设置”对话框。编辑连接目标 Bluetooth 设备的 MAC 地址、开始连接时切换的通信模式、注释。有关详情，请参见第 9-7 页上的“远程 Bluetooth 设备设置”。

(4) 删除

在 Bluetooth 设备一览 (1) 中选择 Bluetooth 设备，单击此按钮，即可删除 Bluetooth 设备。

远程 Bluetooth 设备设置



(1) MAC 地址

设置连接目标 Bluetooth 设备的 MAC 地址。

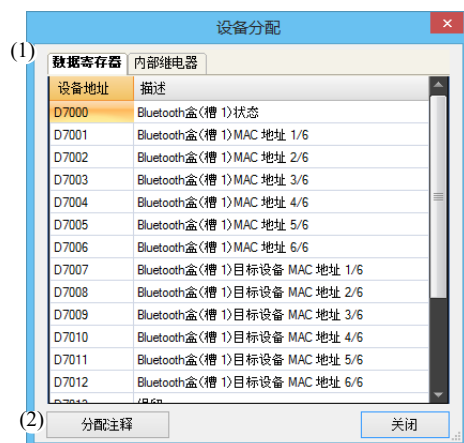
(2) 通信模式

将开始连接时切换的通信模式选择为“维护通信”或“用户通信”。

(3) 注释

输入连接目标 Bluetooth 设备的 MAC 地址。

设备分配



(1) 设备分配一览

将显示 Bluetooth 通信盒的各设置内容和数据寄存器、内部继电器的对应表。

(2) “分配注释”按钮

单击此按钮后，对应表的内容可作为各设备的注释进行设置。

与 Bluetooth 通信的设置和通信 与 PC 进行维护通信

本节将对使用 Bluetooth 通信功能，与 PC 进行维护通信的方法进行介绍。
连接已安装 WindLDR 的 PC，即可监控或更改设备值、下载或上传用户程序。
在 Bluetooth 网络上的动作中，FC6A 型会作为从机等待连接，且将 PC 作为主机连接。



FC6A 型的设置

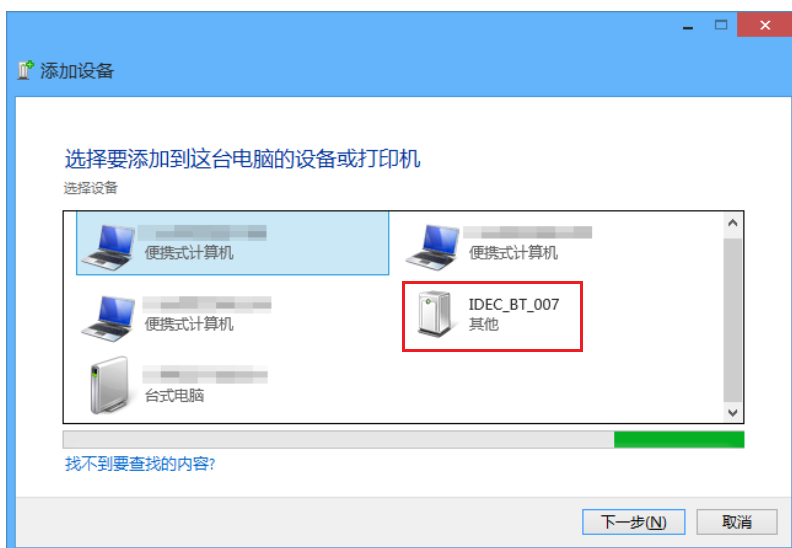
1. 将 Bluetooth 通信盒连接到 FC6A 型。
2. 通过 WindLDR 设置 Bluetooth 通信盒。
有关详情，请参见第 9-3 页上的“设置 WindLDR”。
例) “Bluetooth 通信盒设置”对话框的设置内容

设置项目	说明	
Bluetooth 设备名称设置	IDEC_BT_007	
通信模式	维护通信	
配对设置	PIN 码“1234”	
设备设置	控制寄存器	D7000
	控制继电器	M0100

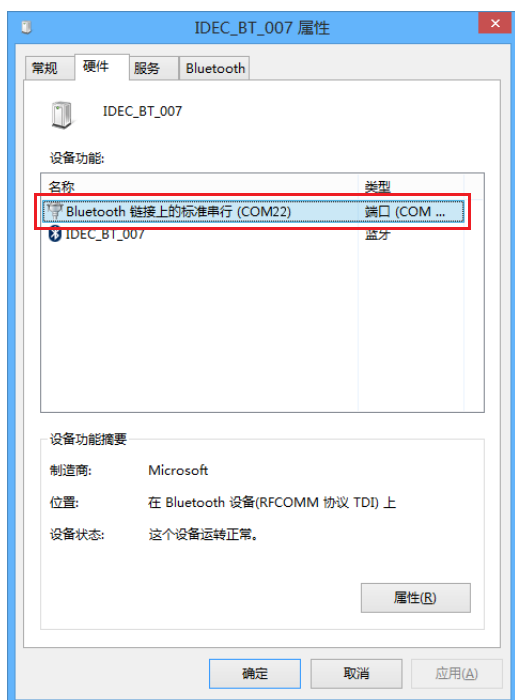
3. 将用户程序下载到 FC6A 型。

设置 PC

1. 在 PC 中搜索 Bluetooth 通信盒的名称。
名称为在 WindLDR 的“Bluetooth 通信盒设置”对话框的“Bluetooth 设备名称设置”中设置的名称。
(IDEC_BT_007)

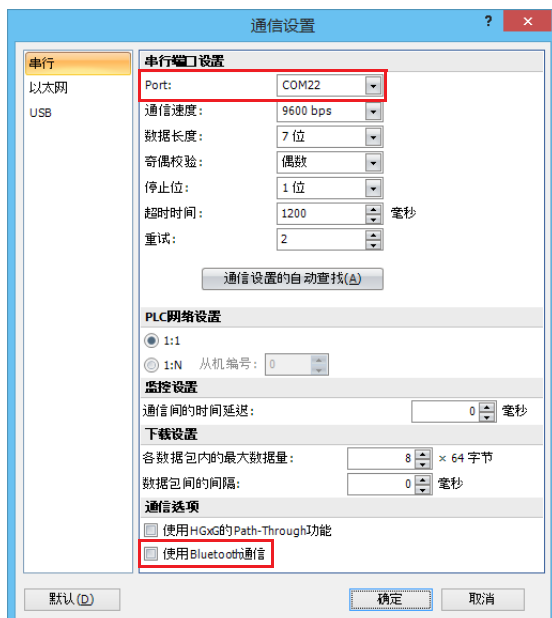


2. 连接到 Bluetooth 通信盒，生成虚拟 COM 端口。初次连接时需要配对，会要求输入 PIN 码，因此输入“1234”。



确认 Bluetooth 通信

1. 使用虚拟 COM 端口，进行串行通信。
选择“使用 Bluetooth 通信”。



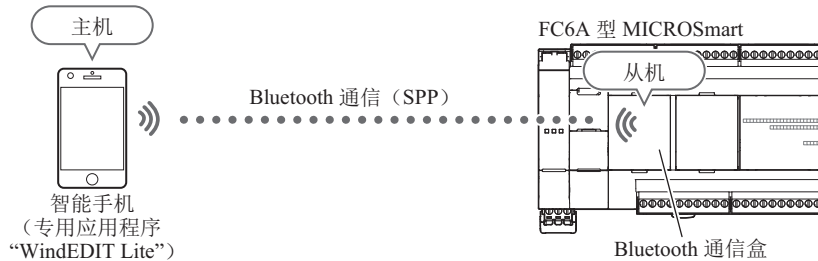
2. 通过 WindLDR 执行监控。
3. 确认表示 Bluetooth 通信状态的设备值。
确认“Bluetooth 通信盒设置”对话框的“设备设置”中设置的控制继电器起始编号+0: Bluetooth 通信盒（盒插槽）的状态（M0100）为 1。

与智能手机进行维护通信

本节将对使用 Bluetooth 通信，与智能手机进行维护通信的方法进行介绍。

连接已安装专用应用程序的智能手机，即可监控或更改设备值、下载或上传用户程序。

在 Bluetooth 网络上的动作中，FC6A 型会作为从机等待连接，且将智能手机作为主机连接。



而且，还可使用密码认证限制访问，能够阻截大量匿名访问。

FC6A 型的设置

1. 将 Bluetooth 通信盒连接到 FC6A 型。
2. 通过 WindLDR 设置 Bluetooth 通信盒。
有关详情，请参见第 9-3 页上的“设置 WindLDR”。

例) “Bluetooth 通信盒设置”对话框的设置内容

设置项目	说明	
Bluetooth 名称设置	IDEC_BT_007	
通信模式	维护通信	
配对设置	PIN 码“1234”	
设备设置	控制寄存器	D7000
	控制继电器	M0100

3. 将用户程序下载到 FC6A 型。

设置智能手机

1. 在智能手机中搜索 Bluetooth 通信盒的名称。
变为在 WindLDR 的“Bluetooth 通信盒设置”对话框的“Bluetooth 名称设置”中设置的名称。
(IDEC_BT_007)



注释: Bluetooth 通信设备的搜索画面因您所使用的智能手机的机型而异。

2. 通过专用应用程序连接 Bluetooth 通信盒。



确认 Bluetooth 通信

1. 通过专用应用程序“WindEDIT Lite”确认表示 Bluetooth 通信状态的设备值。
确认“Bluetooth 通信盒设置”对话框的“设备设置”中设置的控制继电器起始编号+0: Bluetooth 通信盒（盒插槽）的状态（M0100）为1。

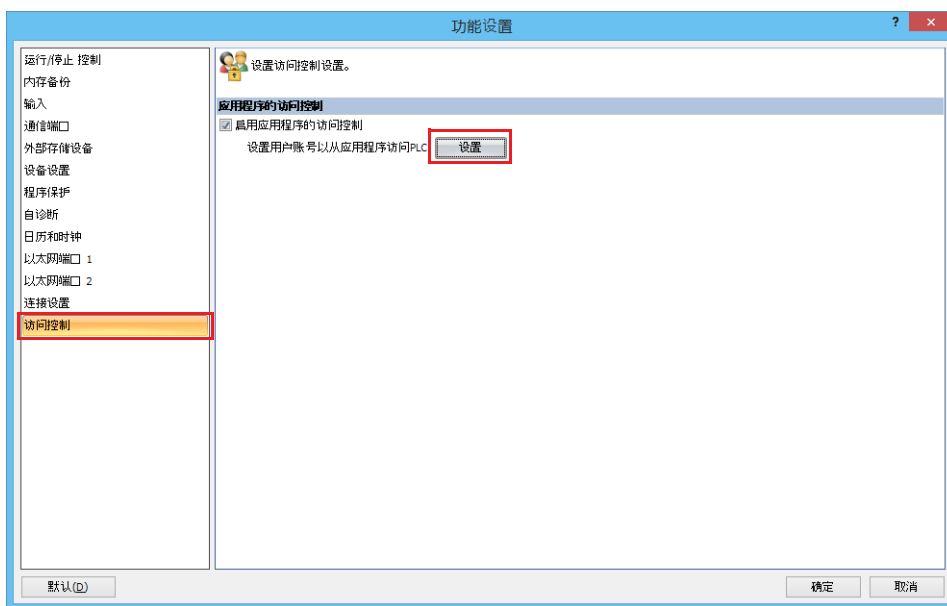
注释：有关专用应用程序“WindEDIT Lite”的操作方法，请参见《WindEDIT Lite 用户手册》。

用户帐户设置

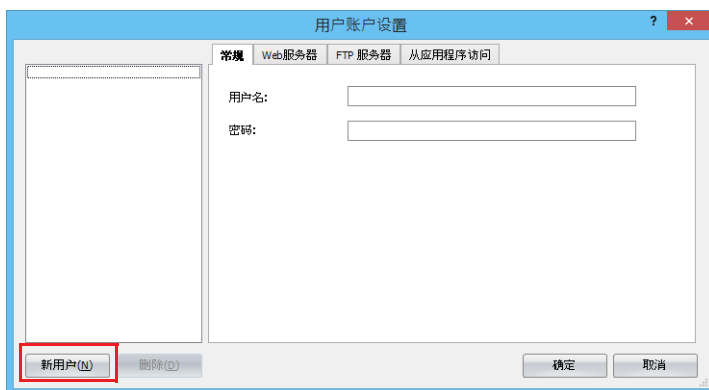
创建用户帐户

● 操作过程

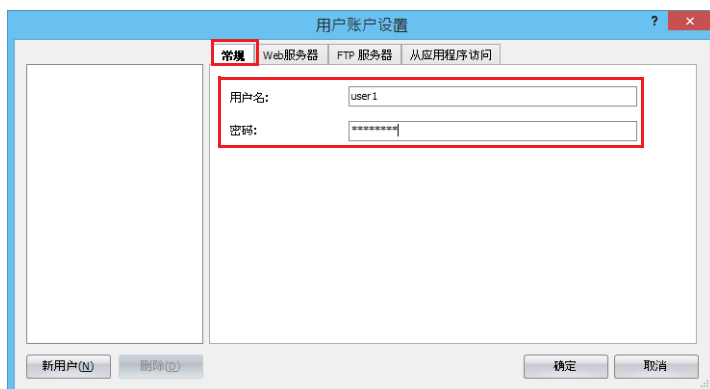
1. 在“功能设置”对话框的“访问控制”中选中“启用应用程序的访问控制”复选框。
2. 单击“设置”按钮。
将显示“用户帐户设置”对话框。



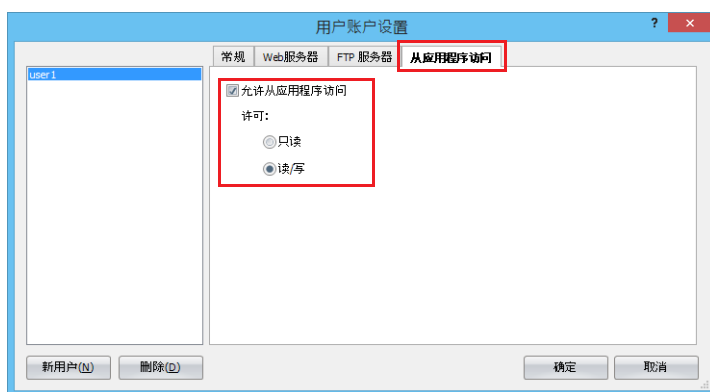
3. 单击“新用户”按钮。
将在列表中创建用户帐户。



4. 在“常规”选项卡中分别设置“用户名”和“密码”。



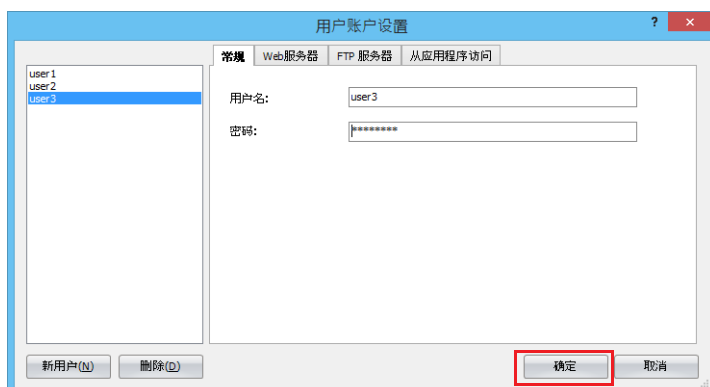
5. 单击“从应用程序访问”选项卡，然后选中“允许从应用程序访问”复选框。
6. 在“许可”中，将帐户的权限选择为“只读”或“读/写”。



7. 重复步骤 2 ~ 6 的操作，创建所需数量的用户帐户。

注释：在列表中选择用户帐户后，可编辑所选用户帐户的“常规”选项卡、“从应用程序访问”选项卡的内容。

8. 单击“确定”按钮。

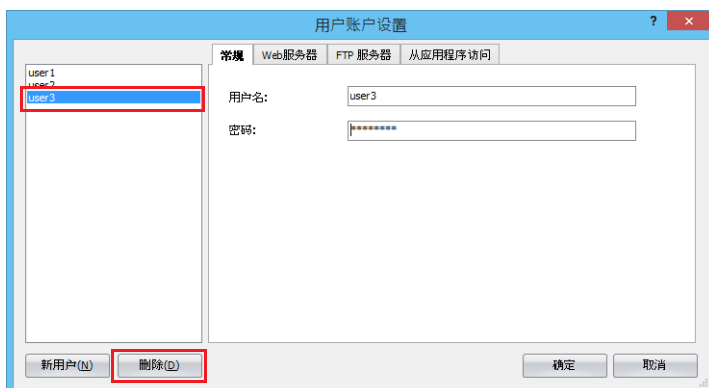


至此，用户帐户的创建完成。

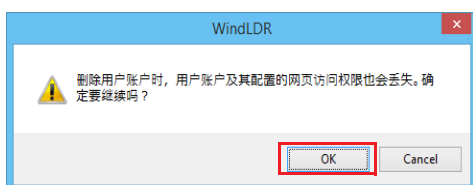
删除已创建的用户帐户

●操作过程

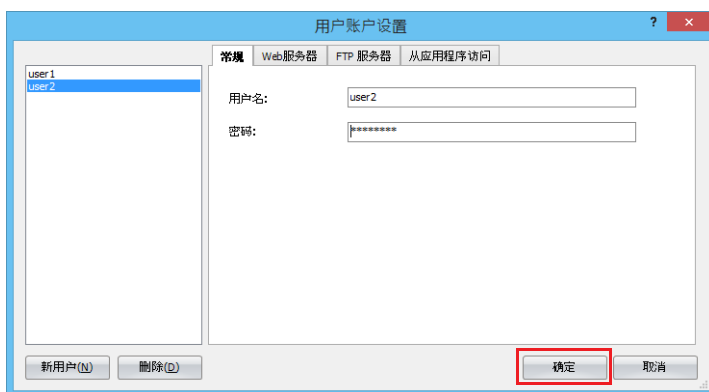
1. 在列表中选择用户帐户，然后单击“删除”按钮。
将显示确认消息。



2. 单击“OK”按钮。



3. 单击“确定”按钮。



至此，完成用户帐户的删除。

与条形码读取器进行用户通信

本节将对使用 Bluetooth 通信功能，与条形码读取器进行用户通信的方法进行介绍。

可与 FC6A 型支持的条形码读取器连接，并进行用户通信。

设置方法因使用的条形码读取器的规格而异。以下说明中，在 Bluetooth 网络上的动作中，条形码读取器会作为从机等待连接，且将 FC6A 型作为主机连接。



注释：推荐条形码读取器为 GBT4400（IDEC 制）。

FC6A 型的设置

1. 将 Bluetooth 通信盒连接到 FC6A 型。
2. 通过 WindLDR 设置 Bluetooth 通信盒。
有关详情，请参见第 9-3 页上的“设置 WindLDR”。

例）“Bluetooth 通信盒设置”对话框的设置内容

设置项目		说明
Bluetooth 设备名称设置		IDEC_BT_007
通信模式		用户通信
配对设置		PIN 码“1234”
设备设置	控制寄存器	D7000
	控制继电器	M0100
建立 PLC 连接		开启 ^{*1}

*1 需要有条形码读取器的 MAC 地址方可进行连接。

3. 将用户程序下载到 FC6A 型。

设置条形码读取器

1. 设置条形码读取器的 PIN 码。
通过条形码读取器读取 PIN 码设置用条形码以设置。因条形码读取器的规格而异。
在 WindLDR 的“Bluetooth 通信盒设置”对话框的“配对设置”中设为 PIN 码。
(1234)
2. 将条形码读取器设为等待连接状态。
通过条形码读取器读取等待连接设置用的条形码以设置。因条形码读取器的规格而异。

确认 Bluetooth 通信

1. 通过 FC6A 型打开连接触发器。
将“Bluetooth 通信盒设置”对话框的“设备设置”中设置的控制继电器起始编号 +1: Bluetooth 通信盒（盒插槽）的连接触发器（M0101）设为 1。控制寄存器的起始编号 +0: Bluetooth 通信盒（盒插槽）的状态（D7000）为 255。
2. 通过 WindLDR 执行监控。
3. 确认表示 Bluetooth 通信状态的设备值。
确认“Bluetooth 通信盒设置”对话框的“设备设置”中设置的控制继电器起始编号 + 0: Bluetooth 通信盒（盒插槽）的状态（M0100）为 1。

10: FTP 服务器 / 客户端功能

本章将对 Plus CPU 模块的 FTP 服务器功能及 FTP 客户端功能进行介绍。

FTP 服务器功能

FTP 服务器功能仅可使用 Plus CPU 模块。

FTP 服务器功能的概要

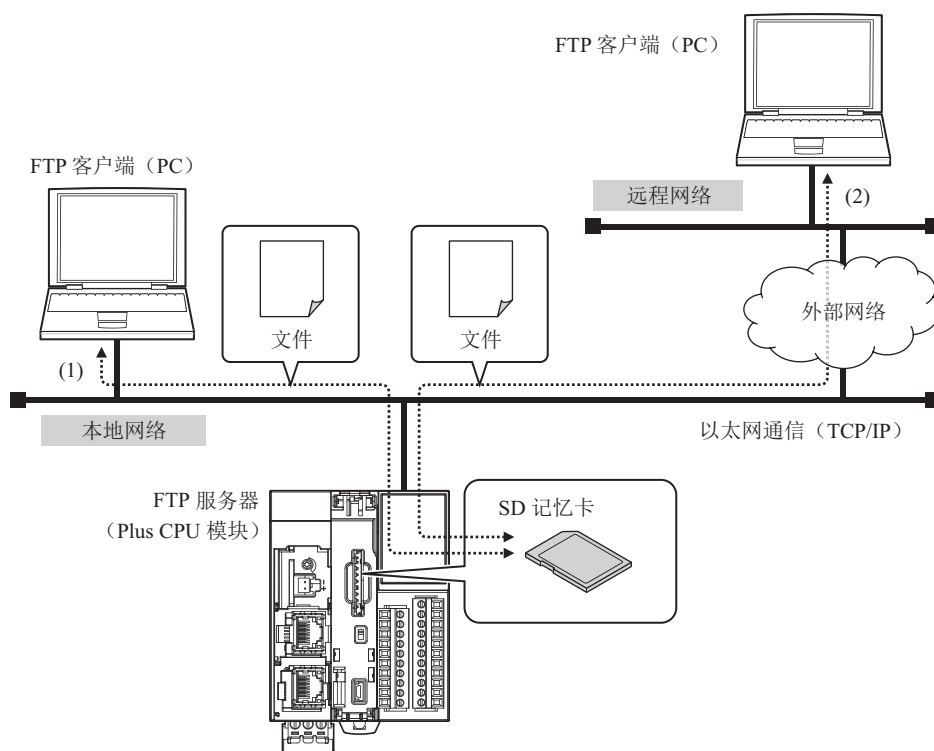
FTP 服务器功能可实现的操作

FTP 服务器功能是将 Plus CPU 模块作为 FTP 服务器，与 PC 等的 FTP 客户端进行通信的功能。可通过 FTP 客户端实现以下操作。

- 从插入到 Plus CPU 模块中的 SD 记忆卡读取文件
- 将文件写入至插入到 Plus CPU 模块中的 SD 记忆卡

系统构成示例

使用 FTP 服务器功能的系统构成示例如下所示。

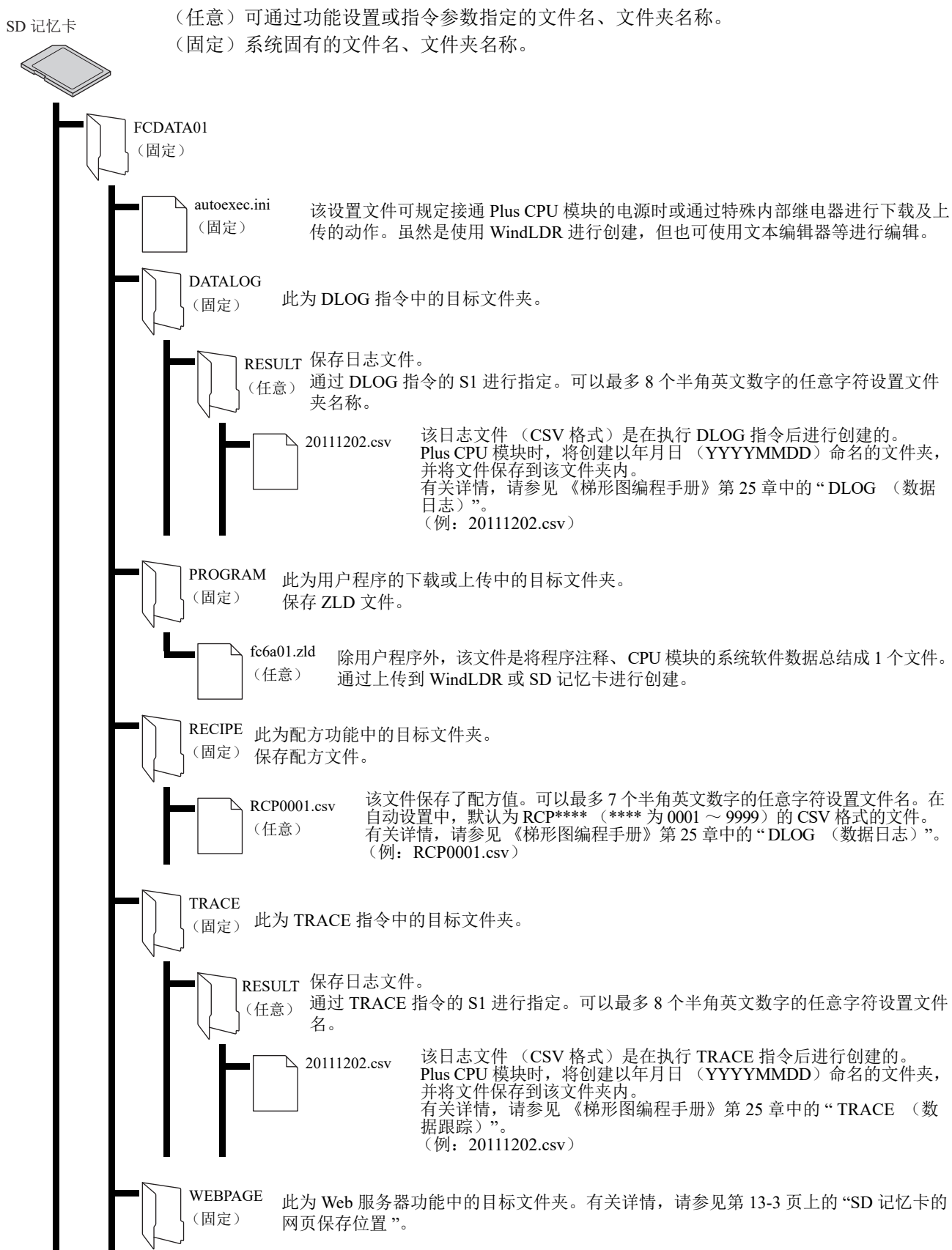


将 Plus CPU 模块的以太网端口 1 连接到网络。

- (1) 通过连接网络的 FTP 客户端访问 Plus CPU 模块，以读取或写入插入到 Plus CPU 模块中的 SD 记忆卡的文件。
- (2) 如果将网络连接到外部网络，可在远程网络所连接的 FTP 客户端中设置本地网络的网关、路由器等。通过远程 FTP 客户端访问 Plus CPU 模块，以读取或写入插入到 Plus CPU 模块中的 SD 记忆卡的文件。关于网关、路由器等的设置，请咨询连接 Plus CPU 模块的网络管理员。

可以 FTP 服务器处理的文件

可通过 FTP 服务器访问插入到 Plus CPU 模块中的 SD 记忆卡内的所有文件。代表了一个 SD 记忆卡的文件夹和文件结构的例子。



支持指令及连接方式

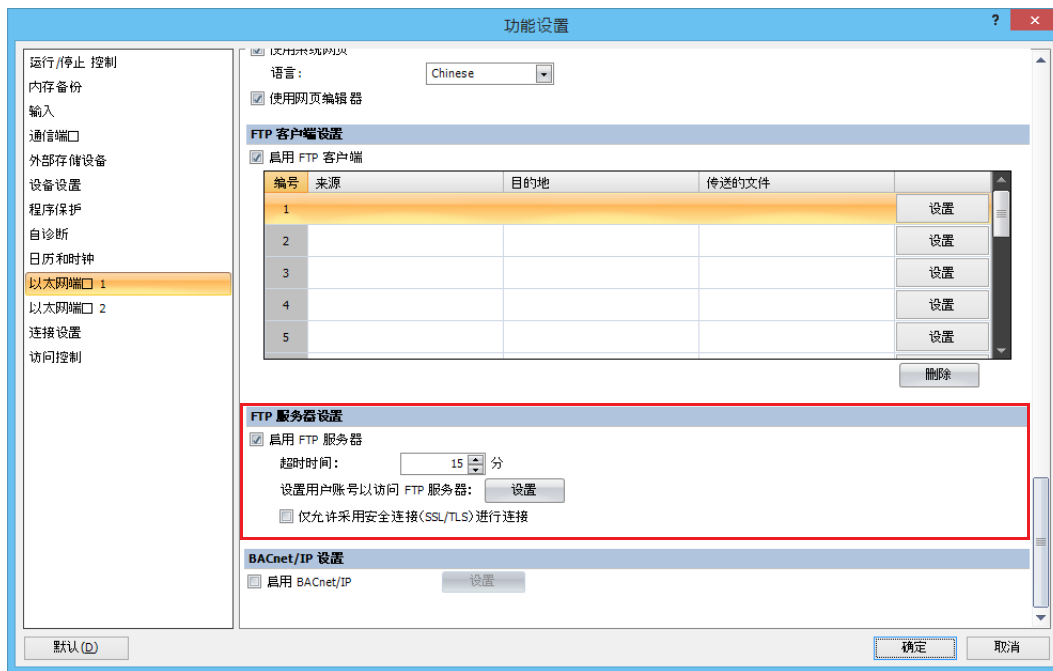
- 支持按照 RFC959 进行定义的指令。
- 支持主动模式及被动模式。
- 支持 FTPS (FTP over SSL/TLS)。支持 Explicit 模式。
- 可同时连接的 FTP 客户端为 1 台。

推荐的 FTP 客户端软件如下所示。

- FFFTP
- FileZilla
- Core FTP Lite

设置 WindLDR

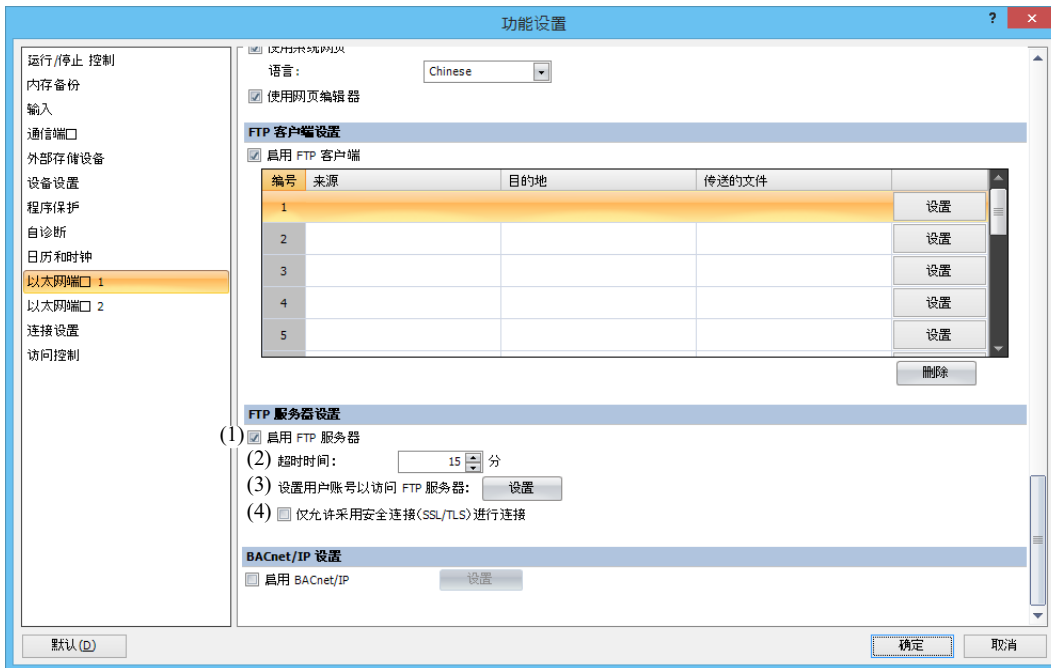
1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。
2. 在“FTP 服务器设置”中选中“启用 FTP 服务器”复选框。
3. 设置 FTP 服务器设置的各设置项目。



4. 将用户程序下载到 Plus CPU 模块。

至此，FTP 服务器的设置完成。

FTP 服务器设置



(1) 启用 FTP 服务器

使用 FTP 服务器功能时，应选中该复选框，设置用户账户。可通过 FTP 客户端读取或写入插入到 Plus CPU 模块中的 SD 记忆卡的文件。FTP 服务器功能所使用的端口编号为 21。

(2) 超时时间

指定登录到 Plus CPU 模块后，如果 Plus CPU 模块与 FTP 客户端之间没有任何通信，到自动切断连接为止的时间。（默认：15 分）

(3) 设置用户账号以访问 FTP 服务器

请单击“设置”按钮，在所显示的“用户帐户设置”对话框中创建用户帐户。有关详情，请参见第 10-5 页上的“用户帐户设置”。

(4) 仅允许采用安全连接（SSL/TLS）进行连接

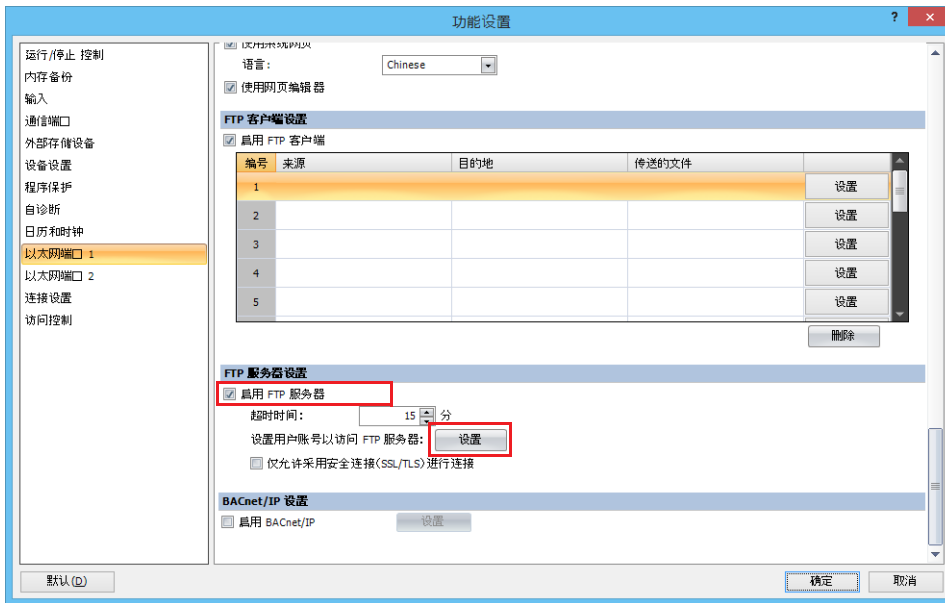
仅 SSL 通信或 TLS 通信中允许登录 FTP 服务器时，应选中该复选框。

用户帐户设置

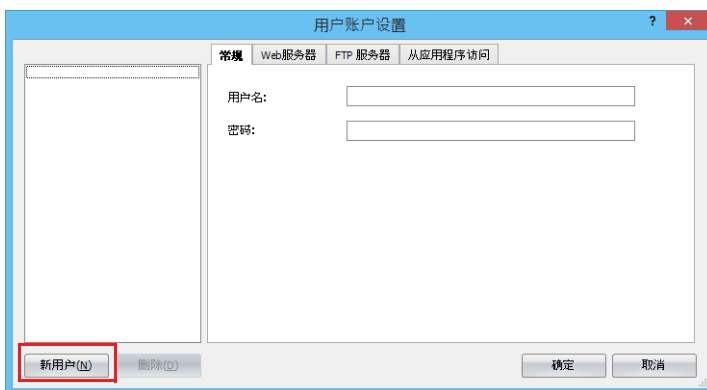
创建用户帐户

●操作过程

1. 在“功能设置”对话框的“FTP 服务器设置”中选中“启用 FTP 服务器”复选框。
2. 单击“设置”按钮。
将显示“用户帐户设置”对话框。

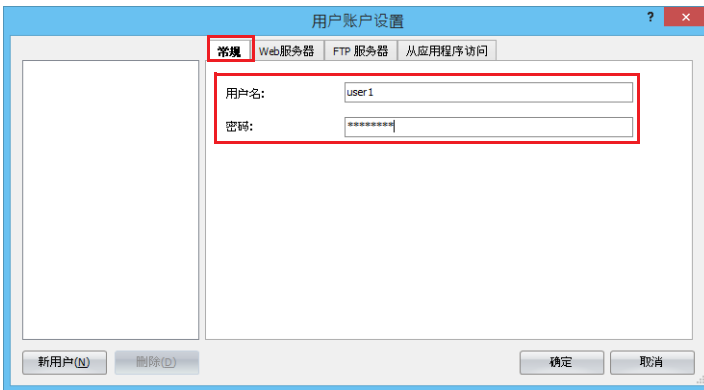


3. 单击“新用户”按钮。
将在列表中创建用户帐户。

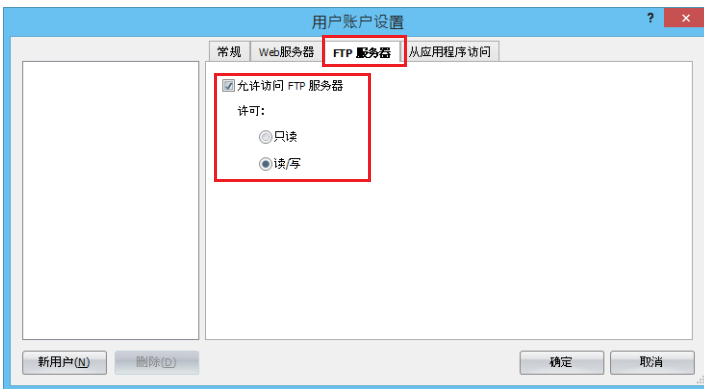


10: FTP服务器/客户端功能

- 在“常规”选项卡中分别设置“用户名”和“密码”。



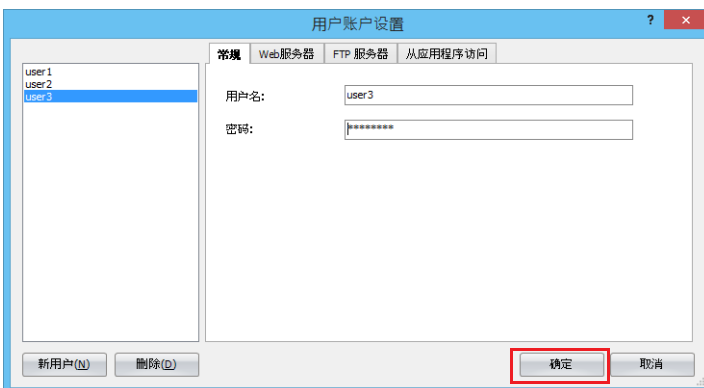
- 单击“FTP 服务器”选项卡，然后选中“允许访问 FTP 服务器”复选框。
- 在“许可”中，将帐户的权限选择为“只读”或“读/写”。



- 重复步骤 2 ~ 6 的操作，创建所需数量的用户帐户。

注释：在列表中选择用户帐户后，可编辑所选用户帐户的“常规”选项卡、“FTP 服务器”选项卡的内容。

- 单击“确定”按钮。

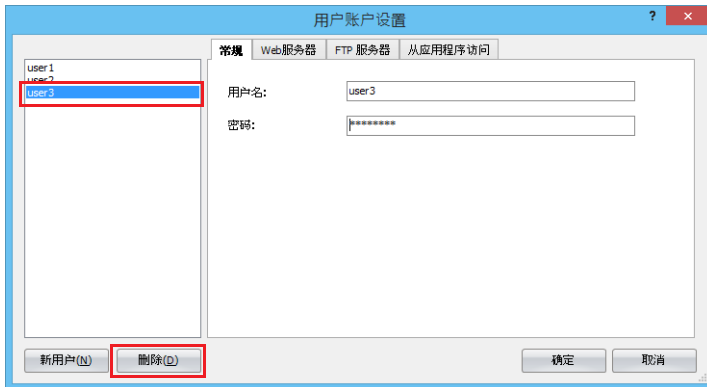


至此，用户帐户的创建完成。

删除已创建的用户帐户

●操作过程

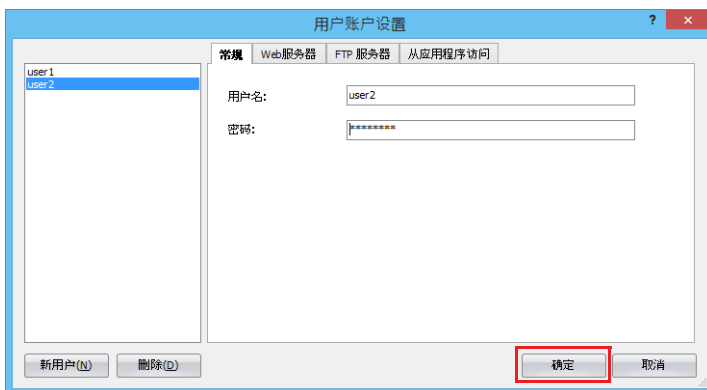
1. 在列表中选择用户帐户，然后单击“删除”按钮。
将显示确认消息。



2. 单击“OK”按钮。



3. 单击“确定”按钮。



至此，完成用户帐户的删除。

FTP 客户端功能

FTP 客户端功能仅可使用 Plus CPU 模块。

FTP 客户端功能的概要

FTP 客户端功能可实现的操作

FTP 客户端功能是将 Plus CPU 模块作为 FTP 客户端，在插入到 Plus CPU 模块中的 SD 记忆卡和 FTP 服务器之间复制或移动文件的功能。

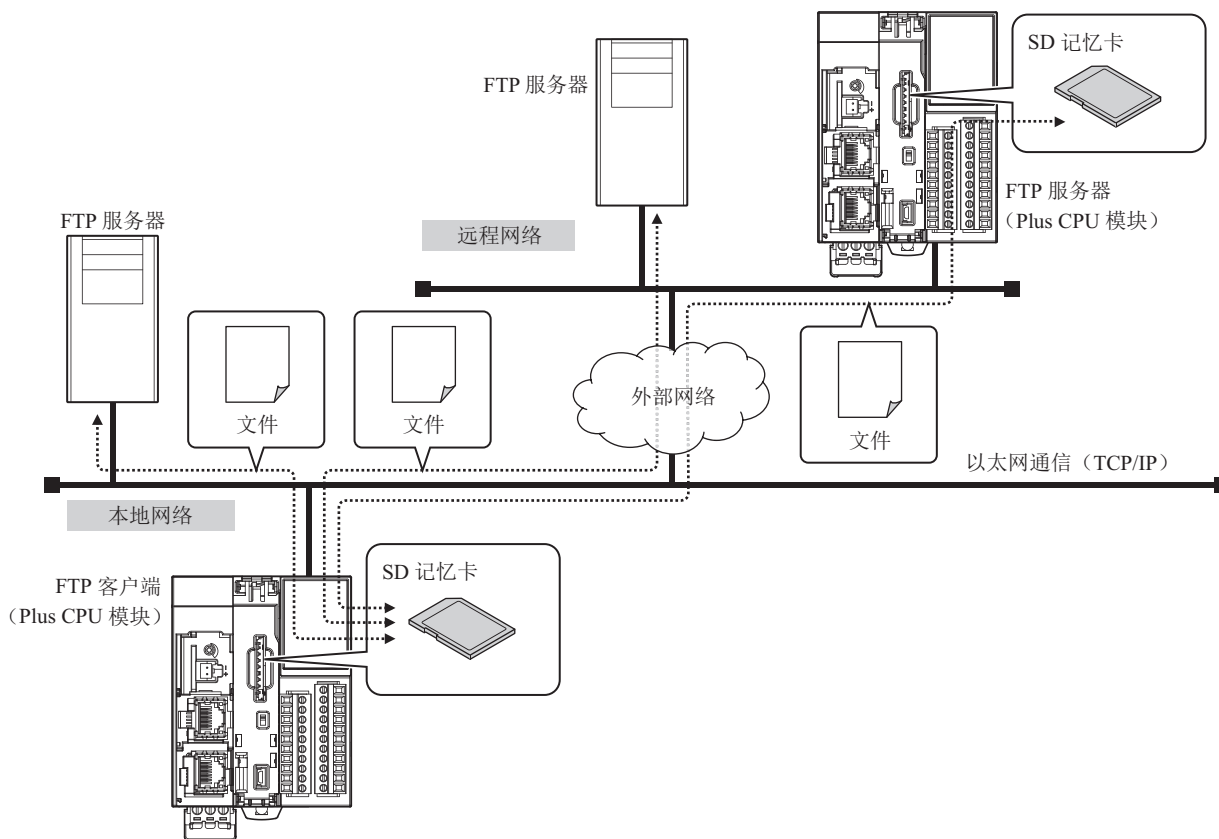
作为 FTP 客户端运行，可实现以下操作。

- 从插入到 Plus CPU 模块中的 SD 记忆卡，将文件复制或移动到 FTP 服务器
- 从 FTP 服务器，将文件复制到插入到 Plus CPU 模块中的 SD 记忆卡

系统构成示例

使用 FTP 客户端功能的系统构成示例如下所示。

将 Plus CPU 模块的以太网端口 1 连接到网络。



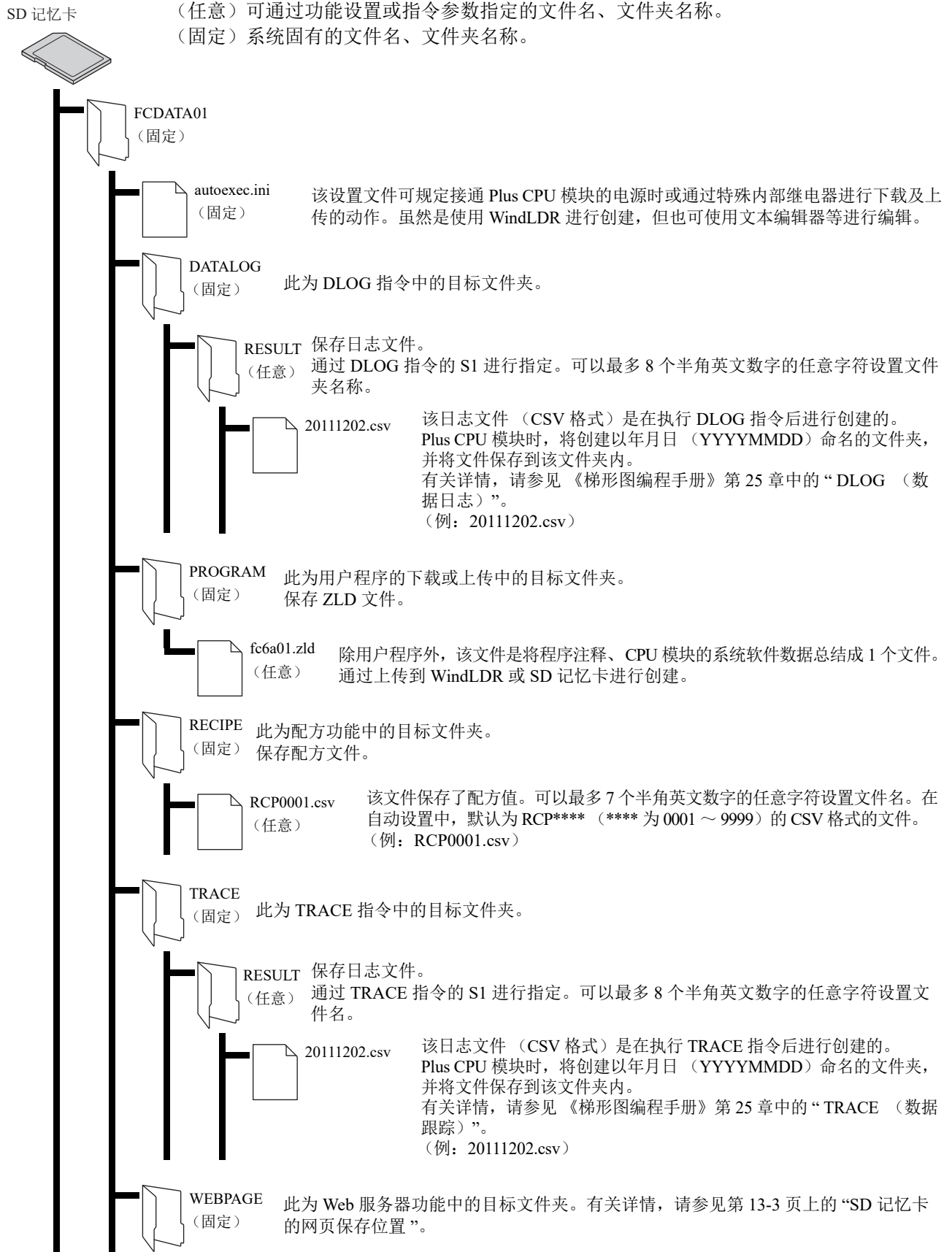
进行 Plus CPU 模块的以太网设置（IP 地址、子网掩码、默认网关），然后连接到本地网络。
在 Plus CPU 模块中登录连接目标的 FTP 服务器。

FTP 客户端功能（文件传送设置）的执行条件成立时执行以下内容。

- 从插入到 Plus CPU 模块中的 SD 记忆卡，将文件复制或移动到 FTP 服务器
- 从 FTP 服务器，将文件复制到插入到 Plus CPU 模块中的 SD 记忆卡

可以 FTP 客户端处理的文件

可通过 FTP 客户端访问插入到 Plus CPU 模块中的 SD 记忆卡内的所有文件。代表了一个 SD 记忆卡的文件夹和文件结构的例子。



10: FTP服务器/客户端功能

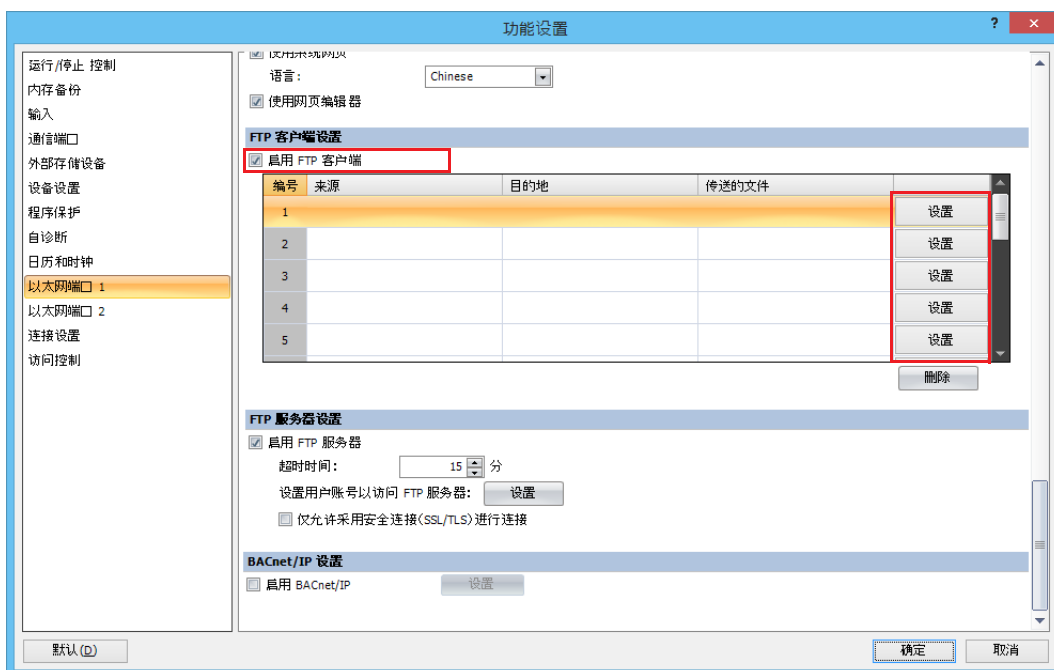
支持指令及连接方式

- 支持按照 RFC959 进行定义的指令。
- 支持主动模式及被动模式。
- 支持 FTPS （FTP over SSL/TLS）。

注释：推荐的 FTP 服务器软件为 Microsoft Internet Information Services。

设置 WindLDR

1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。
2. 选中“启用 FTP 客户端”复选框。
3. 单击 FTP 客户端登录一览中设置的 FTP 客户端的“设置”按钮。
将显示“文件传送设置”对话框。



4. 设置“常规”选项卡、“触发条件”选项卡的各设置项目。



5. 单击“确定”按钮。
6. 重复步骤 3 ~ 5 的操作，创建所需数量的 FTP 客户端设置。
7. 将用户程序下载到 Plus CPU 模块。

至此，FTP 客户端的设置完成。

文件传送设置

“常规”选项卡



(1) 来源

选择保存复制或移动文件的“SD 记忆卡”或“（FTP 服务器）”。

（FTP 服务器）中将显示 FTP 服务器管理器中设置的 FTP 服务器的“编号”+“.”（句号）+“服务器名称”。

例）“编号”为 1、“服务器名称”为 TestServer1 时
“1.TestServer1”

单击“FTP 服务器管理器”按钮，将显示 FTP 服务器管理器。可添加或更改 FTP 服务器。有关详情，请参见第 10-17 页上的“FTP 服务器管理器”。

(2) 位置

选择复制或移动文件的保存位置的指定方法。

文件夹路径 / 文件路径 : 以字符串指定复制或移动文件的保存位置的文件夹路径或文件路径。最大字符数为 247 个半角英文数字。

例）复制或移动 SD 记忆卡的“FCDATA01”文件夹的“DATALOG”文件夹的文件时

FCDATA01\DATALOG

数据寄存器 : 作为复制或移动文件的保存位置的文件夹路径，指定所使用数据读取源的字设备。以指定的数据寄存器为起始依序读取数值，将终端字符 NULL（0x00）之前的对象作为字符数据进行处理，然后作为文件夹路径或文件路径。

10: FTP服务器/客户端功能

(3) 传送的文件

选择复制或移动文件的指定方法。

指定条件 : 以条件指定复制或移动的文件。从以下选项中选择条件。
“所有文件”、“仅最新文件”、“在 24 小时内刷新的文件”、
“在 7 天内刷新的文件”

指定文件名 : 以字符串指定复制或移动的文件。最大字符数为 247 个半角英文数字。

(4) 目的地

选择保存复制或移动文件的“SD 记忆卡”或“(FTP 服务器)”。

(FTP 服务器) 中将显示 FTP 服务器管理器中设置的 FTP 服务器的“编号”+“.(句号)”+“服务器名称”。

例) “编号”为 1、“服务器名称”为 TestServer1 时
“1.TestServer1”

单击“FTP 服务器管理器”按钮，将显示 FTP 服务器管理器。可添加或更改 FTP 服务器。有关详情，请参见第 10-17 页上的“FTP 服务器管理器”。

(5) 位置

选择复制或移动文件的保存位置的指定方法。

文件夹路径 : 以字符串指定复制或移动文件的保存位置的文件夹路径。最大字符数为 247 个半角英文数字。

例) 将复制或移动文件保存到 FTP 服务器的“20170423”文件夹的“DATALOG”文件夹时
20170423\DATALOG

数据寄存器 : 作为复制或移动文件的保存位置的文件夹路径，指定所使用数据读取源的字设备。以指定的数据寄存器为起始依序读取数值，将终端字符 NULL (0x00) 之前的对象作为字符数据进行处理，然后作为文件夹路径。

(6) 操作

将文件的传送方法选择为“复制”或“传送”。在“来源”中选择了“(FTP 服务器)”时，仅可设置“复制”。

(7) 如果目的地文件存在

将传送目标中存在同名文件时的处理方法选择为“跳跃”或“覆盖”。

注释: 要传输的文件必须标有扩展名。如果缺少扩展名，文件可能不会被传输。

(8) 传送失败时存储错误信息

如果在复制或移动文件过程中发生错误时将错误信息存储到设备地址中，应选中该复选框，并指定写入目标的设备地址。开始复制或移动文件时在设备地址中写入 0，如果发生错误，则在相应的位中写入 1。

指定的设备地址的各位中设有以下错误信息。

位位置	错误信息	内容	对策
0	SD 记忆卡访问错误	<ul style="list-style-type: none">未插入作为传送源或传送目标设置的 SD 记忆卡。无法访问作为传送源或传送目标设置的 SD 记忆卡。	<ul style="list-style-type: none">插入可访问的 SD 记忆卡。安装外部内存 SD 记忆卡。
1	SD 记忆卡读写错误	<ul style="list-style-type: none">无法从作为传送源或传送目标设置的 SD 记忆卡中读取文件夹或文件的内容。无法对作为传送源或传送目标设置的 SD 记忆卡进行文件夹创建、文件写入及文件清空等操作。	<ul style="list-style-type: none">插入可读写的 SD 记忆卡。插入可用空间充足的 SD 记忆卡。
2	FTP 服务器连接错误	即使经过连接超时时间指定的时间，仍无法访问作为传送源或传送目标设置的 FTP 服务器。	<ul style="list-style-type: none">确认 LAN 电缆是否正确连接。确认 Plus CPU 模块的网络设置是否正确。确认 FTP 服务器的 IP 地址及端口编号是否正确。

位位置	错误信息	内容	对策
3	FTP 服务器验证错误	用户名或密码错误，无法访问 FTP 服务器。	确认用户名及密码是否正确。
4	FTP 服务器指令错误	对发送到 FTP 服务器的指令，传回了错误。	咨询 FTP 服务器的管理员。
5	FTP 服务器传送错误	即使经过传送超时时间设置的时间，仍没有来自 FTP 服务器的响应。	咨询 FTP 服务器的管理员。
6	— 保留 —	—	—
7	其他错误	其他异常	咨询 FTP 服务器的管理员。
8 ~ 15	— 保留 —	—	—

(9) 传送结束时报告

如果在复制或移动文件结束时通知，应选中该复选框，并指定写入目标的设备地址。

注释：

- 无法从 FTP 服务器复制或移动文件到 FTP 服务器。
- 无法在插入到 Plus CPU 模块中的 SD 记忆卡之间复制或移动文件。
- 无法复制或移动子文件夹。
- 无法在文件夹路径或文件路径中使用以下半角字符。
; : * ? " < > |
- 在“数据寄存器”中设置超过限制或无法使用的字符时的文件夹路径如下所示。
 - 如果文件夹路径或文件路径的字符串超过最大设备地址点数（无 NULL），将为可存储到从起始到最大设备地址点数的设备地址中的字符。
 - 设置无法使用的字符时，将为之前的字符。

“触发条件”选项卡

文件传送设置

常规 触发条件

(1) 设备:

触发类型: 上升沿

设备地址: M0300

(2) 周期:

星期:

星期日 星期一 星期二 星期三 星期四 星期五 星期六

(3) 触发时间(分:时):

09 : 00

17 : 00

确定 取消

(1) 设备

以设备地址指定复制或移动文件的条件时，应选中该复选框。
从“上升沿”或“下降沿”中选择，并指定条件的设备。仅可使用内部继电器。

(2) 周期

以星期及时间指定复制或移动文件的条件时，应选中该复选框，然后选中执行星期的复选框。

(3) 触发时间

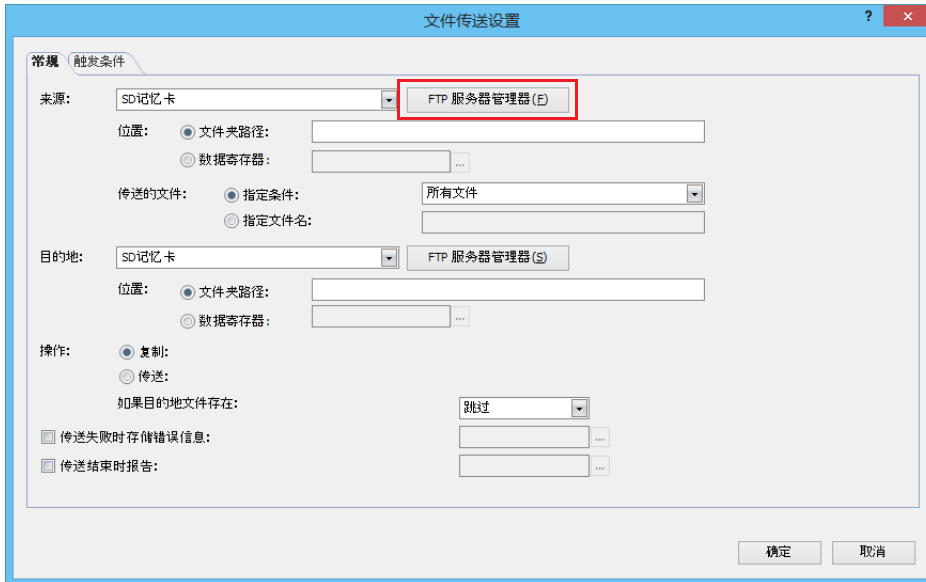
以时间指定复制或移动文件的条件时，应选中该复选框，然后指定时间。最多可设置2个从0时00分到24时00分的范围。

FTP 服务器设置

设置连接目标的 FTP 服务器

●操作过程

1. 在“文件传送设置”对话框的“常规”选项卡中单击“FTP 服务器管理器”按钮。将显示 FTP 服务器管理器。



2. 在 FTP 服务器一览中选择登录 FTP 服务器的编号，然后单击“编辑”按钮。将显示所选编号的“FTP 服务器 n 设置”对话框（n: FTP 服务器管理器编号）。



注释: 删除登录的 FTP 服务器时，在 FTP 服务器一览中选择 FTP 服务器，然后单击“删除”按钮。

10: FTP服务器/客户端功能

3. 设置 FTP 服务器的各项目，然后单击“确定”按钮。

FTP 服务器 1 设置

FTP 服务器设置:

FTP 服务器名称: FTPServer1

IP 地址: IP 地址: 192.168.0.44 (D0000 - D0020) 主机名称:

用户名: test_user001 (D0021 - D0041)

密码: ***** (D0042 - D0062)

端口号: 21 (D0063)

传输模式: 被动 主动 (D0064)

连接超时时间: 60 秒 (D0065)

传输超时时间: 60 秒 (D0066)

启用安全连接 (SSL/TLS) (D0067)

高级设置:

通过设备地址指定 FTP 服务器设置

顶端设备地址: D0000 ...

将设置作为默认值复制到设备地址 M0000 ...

确定 取消

4. 重复步骤 2 和 3 的操作，登录所需的 FTP 服务器。

5. 单击“确定”按钮。

FTP 服务器管理器

编号	服务器名称	IP 地址	用户名	密码	端口编号	传输模式	连接超时时间	传输超时时间	编辑 (E)	删除 (D)
1	FTPServer1	192.168.0.44	test_user001	****	21	被动	60秒	60秒		
2		0.0.0.0			21	被动	60秒	60秒		
3		0.0.0.0			21	被动	60秒	60秒		
4		0.0.0.0			21	被动	60秒	60秒		
5		0.0.0.0			21	被动	60秒	60秒		
6		0.0.0.0			21	被动	60秒	60秒		
7		0.0.0.0			21	被动	60秒	60秒		
8		0.0.0.0			21	被动	60秒	60秒		
9		0.0.0.0			21	被动	60秒	60秒		
10		0.0.0.0			21	被动	60秒	60秒		
11		0.0.0.0			21	被动	60秒	60秒		

确定 取消

至此，FTP 服务器的设置完成。

FTP 服务器管理器



(1) FTP 服务器一览

一览显示各编号的 FTP 服务器的设置。双击单元，将显示“FTP 服务器 n 设置”对话框（n：FTP 服务器管理器编号），即可进行编辑。有关详情，请参见第 10-18 页上的“FTP 服务器的个别设置”。

- 编号：将显示管理 FTP 服务器设置的编号（1 ~ 16）。
- 服务器名称：将显示 FTP 服务器的名称。
- IP 地址：将显示 FTP 服务器的 IP 地址。
- 用户名：将显示 FTP 服务器的用户名。
- 密码：FTP 服务器的密码会以 *（星号）显示。
- 端口编号：将显示 FTP 服务器的端口编号。
- 传送模式：将显示 FTP 服务器的传送模式。
- 连接超时时间：将显示“文件传送设置”中设置的“触发条件”成立后，如果没有来自尝试连接的 FTP 服务器的响应，到自动取消连接处理为止的时间。
- 传送超时时间：将显示“文件传送设置”中设置的“触发条件”成立后，如果在确立数据连接后没有来自 FTP 服务器的响应，到自动取消文件传送处理为止的时间。
- 顶端设备地址：将显示以设备地址的值指定 FTP 服务器的设置时的顶端设备地址。
以设置内容作为初始值复制到设备地址中：
将显示以 FTP 服务器设置的设置内容作为初始值复制条件的位设备。

(2) “编辑”按钮

登录或更改所选编号的设置。

从 FTP 服务器一览 (1) 中选择编号，单击此按钮，将显示“FTP 服务器 n 设置”对话框。将“FTP 服务器 n 设置”对话框的设置反映到所选编号中。

有关详情，请参见第 10-18 页上的“FTP 服务器的个别设置”。

(3) “删除”按钮

删除在 FTP 服务器一览 (1) 中所选编号的设置。

FTP 服务器的个别设置

FTP 服务器 1 设置

FTP 服务器设置:

(1) FTP 服务器名称: FTPServe1 (D0000 - D0020)

(2) IP 地址: IP 地址: 192.168.0.44 (D0000 - D0020)
 主机名称: (D0021 - D0041)

(3) 用户名: test_user001 (D0021 - D0041)

(4) 密码: ***** (D0042 - D0062)

(5) 端口号: 21 (D0063)

(6) 传送模式: 被动 主动 (D0064)

(7) 连接超时时间: 60 秒 (D0065)

(8) 传送超时时间: 60 秒 (D0066)

(9) 启用安全连接 (SSL/TLS) (D0067)

高级设置:

(10) 通过设备地址值指定 FTP 服务器设置
顶端设备地址: D0000

(11) 将设置作为默认值复制到设备地址
M0000

确定 取消

(1) FTP 服务器名称

输入 FTP 服务器的名称。最大字符数为 40 个字符。
默认为“FTPServn”。(n: FTP 服务器管理器的编号)

(2) IP 地址

选择 FTP 服务器的识别方法。

IP 地址 : 输入 FTP 服务器的 IP 地址。

输入形式为“xxx.xxx.xxx.xxx”。“xxx”中输入 0 至 255 的值。

主机名称: 输入 FTP 服务器的主机名称。

主机名称的最大字符数为 40 个半角字符。仅可使用英文数字及符号。

(3) 用户名

输入 FTP 服务器的用户账户名称。

用户名的最大字符数为 40 个半角字符。仅可使用英文数字及符号。

(4) 密码

输入 FTP 服务器的密码。

密码的最大字符数为 40 个半角字符。仅可使用英文数字及符号。

(5) 端口号

指定 FTP 服务器的端口编号 (0 ~ 65535)。

(6) 传送模式

选择 FTP 服务器的传送模式。

被动 : 使用被动模式, 确立数据连接。

主动 : 使用主动模式, 确立数据连接。

(7) 连接超时时间

指定“文件传送设置”中设置的“触发条件”成立后, 如果没有来自尝试连接的 FTP 服务器的响应, 到自动取消连接处理为止的时间 (10 ~ 300 秒)。

(8) 传送超时时间

指定“文件传送设置”中设置的“触发条件”成立后, 如果在确立数据连接后没有来自 FTP 服务器的响应, 到自动取消文件传送处理为止的时间 (10 ~ 300 秒)。

(9) 启用安全连接 (SSL/TLS)

如果所使用的 FTP 服务器中需要 SSL 通信或 TLS 通信，应选中该复选框。

(10) 通过设备地址值指定 FTP 服务器设置

如果以设备地址指定“FTP 服务器设置”，应选中该复选框。

指定“顶端设备地址”中使用的设备地址。以指定的设备地址的地址编号为起始，分配“FTP 服务器设置”的设置项目。

(11) 将设置作为默认值复制到设备地址

如果以“FTP 服务器设置”的设置内容作为初始值复制到设备地址中，应选中该复选框。

指定作为复制设置内容条件的设备地址。

在该设备地址中写入 1 后，以“顶端设备地址”中设置的设备地址的地址编号为起始，在分配的设备地址中写入“FTP 服务器设置”中设置的值。

以设备地址的值指定 FTP 服务器时的地址分配

选中“以设备地址的值指定 FTP 服务器的设置”复选框时，以“顶端设备地址”中设置的设备地址的地址编号为起始，分配“设置 FTP 服务器”的设置项目。分配的详情如下所示。

设置项目	地址编号	字数	数据类型
IP 地址或主机名称	+0 ~ +20	21 ^{*1*2}	数值
用户名	+21 ~ +41	21 ^{*2*3}	字符串
密码	+42 ~ +62	21 ^{*2*3}	字符串
端口编号	+63	1	数值
传送模式 (0: 被动、1: 主动)	+64	1	数值
连接超时时间 (以秒为单位)	+65	1	数值
传送超时时间 (以秒为单位)	+66	1	数值
启用安全连接 (SSL/TLS) (0: 使用 SSL/TLS、1: 不使用 SSL/TLS)	+67	1	数值

*1 IP 地址时，从起始仅使用 4 个字，剩余 17 个字为预约区域。

*2 第 21 个字与数据寄存器的值无关，会作为终端字符 NULL (0x00) 进行处理。

*3 设置的字符串小于 20 个字时，请作为字符串的末尾添加终端字符 NULL (0x00)。

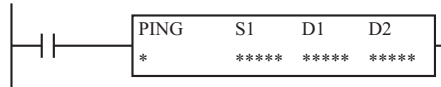
11: PING 发送功能

简介

本章介绍向指定的远程主机发送 PING 数据包及确认是否可在网际协议（IP）层上进行通信的 PING 指令。执行 PING 发送功能时，应使用 PING 指令。

PING (Ping)

PING 指令用于向指定的远程主机发送 ping 数据包。在 Plus CPU 模块中，可选择以太网端口 1 或 2 使用。



操作详情

当 PING 指令的输入端开启后，PING 指令向 S1 指定的远程主机发送 ping 数据包。FC6A 型一旦接收到远程主机的答复，D1 指定的完成输出位即会开启，运行状态（运行转换状态和错误代码）将储存在 D2 指定的设备中。D2+1 是为系统预留的。

在输入开启时执行 PING 指令，并发送 PING 要求。如果其输入继续保持开启状态，则在发送 PING 后将执行相同的 PING 指令。为避免持续执行 PING 指令，PING 指令的输入变为仅 1 次扫描开启，请将 SOTU（上升沿微分）指令和 SOTD（下降沿微分）指令组合使用。有关 SOTU 指令和 SOTD 指令的详情，请参见《梯形图编程手册》第 4 章中的“SOTU 和 SOTD（上升沿微分和下降沿微分）”。

若要使用 PING 指令，需要事先创建 FC6A 型的远程主机列表。有关远程主机列表的详情，请参见第 3-21 页上的“远程主机列表”。

有效设备

	设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1	源 1	远程主机号	-	-	-	-	-	-	X	-	1 - 255	-
D1	目标 1	完成输出位	-	X	X*1	-	-	-	-	-	-	-
D2	目标 2	运行状态	-	-	-	-	-	-	X	-	-	-

*1 不能使用特殊内部继电器。

PING 指令中的设备

S1（源 1）

指定远程主机号。可以指定常量或数据寄存器。

D1（目标 1）

指定内部继电器或输出点作为完成输出位。当 FC6A 型收到远程主机的答复时，完成输出位开启。如果未在“功能设置”中设置的指定的超时时间内收到远程主机的答复，完成输出位也会开启。有关超时时间设置的详情，请参见第 3-13 页上的“PING 设置”。

D2（目标 2）

指定数据寄存器作为运行状态。目标 2 占用以 D2 指定的数据寄存器开始的两个连续的数据寄存器。运行状态存储在 D2 中。D2+1 是为系统预留的。请指定设备范围内的起始编号。

11: PING发送功能

注释:

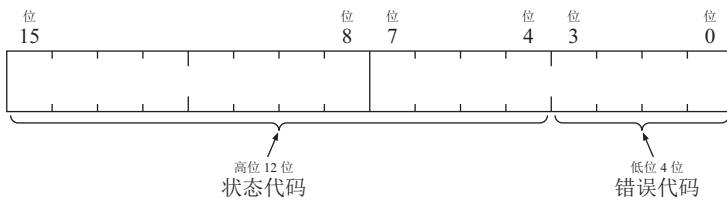
- 当数据寄存器指定为 S1 时，不要在执行 PING 指令的同时更改数据寄存器值。
- FC6A 型不会响应自行发送的 ping 数据包。

运行状态包括运行转换状态（状态代码）和错误详情（错误代码）。掩盖运作状态（最低 4 位）可以获得状态代码。错误代码存储在运行状态（最低 4 位）中。

运行状态的数据寄存器的值除以 16，其余数即为错误代码。

[运行状态的数据寄存器的值为“66”时]

$66 \div 16 = 4$ 余 2，
则错误代码为“2”。



状态代码	运行转换状态	说明
16 (000000010000)	准备转换	从 PING 指令输入开启之时起，直到 PING 数据包发送到指定的远程主机为止
32 (000000100000)	等待响应	从向指定的远程主机发送 ping 数据包之时起，直到收到响应信号为止
64 (000001000000)	PING 指令完成	完成 PING 指令执行后，允许处理下一传送操作

注释: 如果状态代码不在以上所列代码之中，则可能发生 PING 指令错误。

错误代码	错误详情	完成输出
0 (0000)	正常	ON (状态代码为 64)
1 (0001)	-	-
2 (0010)	超时错误	ON
3 (0011)	无法使用 DNS 解析目标远程主机的主机名	ON
4 (0100)	指定的远程主机不在远程主机列表中	ON
5 (0101)	同时执行多个 PING 指令	ON
6 (0110)	参数错误	ON

12: EMAIL 发送功能

简介

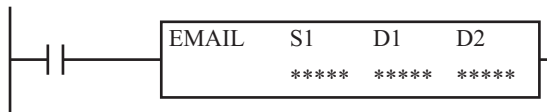
本章介绍事先注册的发送电子邮件的 EMAIL 指令。

概述

Plus CPU 模块的以太网端口 1 及 HMI 模块的 HMI- 以太网端口支持 EMAIL 发送功能。通过执行 EMAIL 指令，可发送最多 255 种电子邮件。

EMAIL 指令（发送电子邮件）

EMAIL 指令用于发送电子邮件。



操作详情

输入开启后，可通过 Plus CPU 模块的以太网端口 1 及 HMI 模块 HMI- 以太网端口发送以 S1 指定的电子邮件编号的电子邮件。FC6A 型 MICROSmart 一旦接收到电子邮件发送服务器的响应信号，D1 指定的完成输出口即会开启，运行状态（运行转换状态和错误代码）将储存在 D2 指定的设备中。D2+1 是为系统预留的。

当指令输入端口开启后，执行 EMAIL 指令。如果其输入继续保持开启状态，则在发送电子邮件后将执行相同的 EMAIL 指令。为避免持续执行 EMAIL 指令，EMAIL 指令的启动输入变为仅 1 次扫描开启，请将 SOTU（上升沿微分）指令和 SOTD（下降沿微分）指令组合使用。有关 SOTU 指令和 SOTD 指令的详情，请参见《梯形图编程手册》第 4 章中的“SOTU 和 SOTD（上升沿微分和下降沿微分）”。

使用 EMAIL 指令时，如果要使用 HMI 模块的 HMI- 以太网端口，单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮，在所显示的“HMI 模块设置”对话框中设置电子邮件服务器等，如果要使用 Plus CPU 模块的以太网端口 1，则在“功能设置”的“以太网端口 1”中设置电子邮件服务器等，并需要事先创建 FC6A 型 MICROSmart 的电子邮件。

有效设备

	设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1	源 1	电子邮件编号	-	-	-	-	-	-	X ^{*1}	-	1 - 255	-
D1	目标 1	完成输出	-	X	X ^{*2}	-	-	-	-	-	-	-
D2	目标 2	运行状态	-	-	-	-	-	-	X ^{*1}	-	-	-

*1 无法使用特殊数据寄存器。

*2 不能使用特殊内部继电器。

12: EMAIL 发送功能

EMAIL 指令中的设备

设置以太网端口号

设置发送电子邮件的以太网端口。设置 Plus CPU 模块的以太网端口 1 或 HMI 模块的 HMI- 以太网端口。

S1 (源 1)

指定电子邮件编号。可以指定常量或数据寄存器。

D1 (目标 1)

指定内部继电器或输出点作为完成输出位。当 FC6A 型 MICROSmart 收到电子邮件发送服务器的答复时，完成输出位开启。如果在超时时间内未收到电子邮件发送服务器的答复，完成输出位也会开启。

D2 (目标 2)

指定数据寄存器作为运行状态。目标 2 占用以 D2 指定的数据寄存器开始的两个连续的数据寄存器。运行状态存储在 D2 中。D2+1 是为系统预留的。可以指定数据寄存器 D0000 至 D7998 以及 D10000 至 D55998。请指定设备范围内的起始编号。

注释：当数据寄存器指定为 S1 时，不要在执行 EMAIL 指令的同时更改数据寄存器值。

运行状态包括运行转换状态（状态代码）和错误详情（错误代码）。掩盖运作状态（最低 4 位）可以获得状态代码。错误代码存储在运行状态（最低 4 位）中。

运行状态的数据寄存器的值除以 16，其余数即为错误代码。

[运行状态的数据寄存器的值为“66”时]

$66 \div 16 = 4$ 余 2，
则错误代码为“2”。

状态代码	运行转换状态	说明
16 (000000010000)	准备转换	从开启 EMAIL 指令启动输入端口，直至将电子邮件发送到电子邮件发送服务器。
32 (000000100000)	等待响应	从电子邮件发送到电子邮件发送服务器，直至接收到响应为止。
64 (000001000000)	EMAIL 指令完成	完成 EMAIL 指令执行后，允许处理下一传送操作。

如果状态代码不在以上所列代码之中，则可能发生 EMAIL 指令错误。

错误代码	错误详情	完成输出
0 (0000)	正常	ON (状态代码为 64)
1 (0001)	HMI 模块连接错误 • 未连接 HMI 模块 • HMI 模块未正常动作	ON
2 (0010)	超时错误 • 端口号可能有误。	ON
3 (0011)	• 无法使用 DNS 解析电子邮件发送服务器的主机名。 • 找不到电子邮件发送服务器。 • 电子邮件发送服务器不支持验证功能 • 以太网电缆可能断开或损坏。FC6A 型 MICROSmart 可能没有正确连接到网络。	ON
4 (0100)	未设置指定的电子邮件。	ON
5 (0101)	同时执行多个 EMAIL 指令。	ON
6 (0110)	参数错误 • 启用“使用数据寄存器配置电子邮件设置”时，数据寄存器中未存储设置	ON
7 (0111)	其他错误 *1	ON
8 (1000)	数据寄存器值转换失败 *2	ON
9 (1001)	添加的日志文件的 1 文件大小超过以 DLOG/TRACE 指令设置的上限大小 *3	ON
10 (1010)	不存在指定添加的日志文件 *3	ON
11 (1011)	添加的日志文件数超过 7 *4	ON

*1 错误代码 7 的详情存储在特殊数据寄存器 D8457 (HMI- 以太网端口) 或 D8759 (以太网端口 1) 中。

*2 当数据寄存器值转换失败时，---- 代替数据寄存器值被插入到电子邮件的正文中。

- *3 不添加错误代码 9 或 10 的相应日志文件，可发送无附件的电子邮件。
有关日志数据文件大小的上限设置，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“功能和设置”。
- *4 可发送添加最多 7 个文件的电子邮件。

可通过电子邮件（SMTP）发送服务器的中继将电子邮件发送至收件人。在 EMAIL 指令执行之后，即使存储在 D2 中的值为 64，但如果满足下列条件之一，电子邮件的收件人也可能无法接收电子邮件。此外，无法判断 FC6A 型 MICROSmart 是否能进行接收。

- 中继电子邮件的 SMTP 服务器没有正常工作。
- 电子邮件收件人通过指定电子邮件地址或电子邮件域名过滤接收的电子邮件。

特殊数据寄存器

确认 EMAIL 指令的错误详情

当存储在 EMAIL 指令 D2 中的错误代码为 7 时，可以使用特殊数据寄存器确认电子邮件发送服务器的响应。特殊数据寄存器的分配如下所示。

发送电子邮件的以太网端口	可存储电子邮件的详细错误代码的特殊数据寄存器
HMI 模块的 HMI- 以太网端口	D8457
Plus CPU 模块的以太网端口 1	D8759

电子邮件发送服务器返回的错误代码可能因各电子邮件发送服务器而有所不同。有关错误代码的详情，请与电子邮件发送服务器的管理员联系。错误内容的示例如下所示。

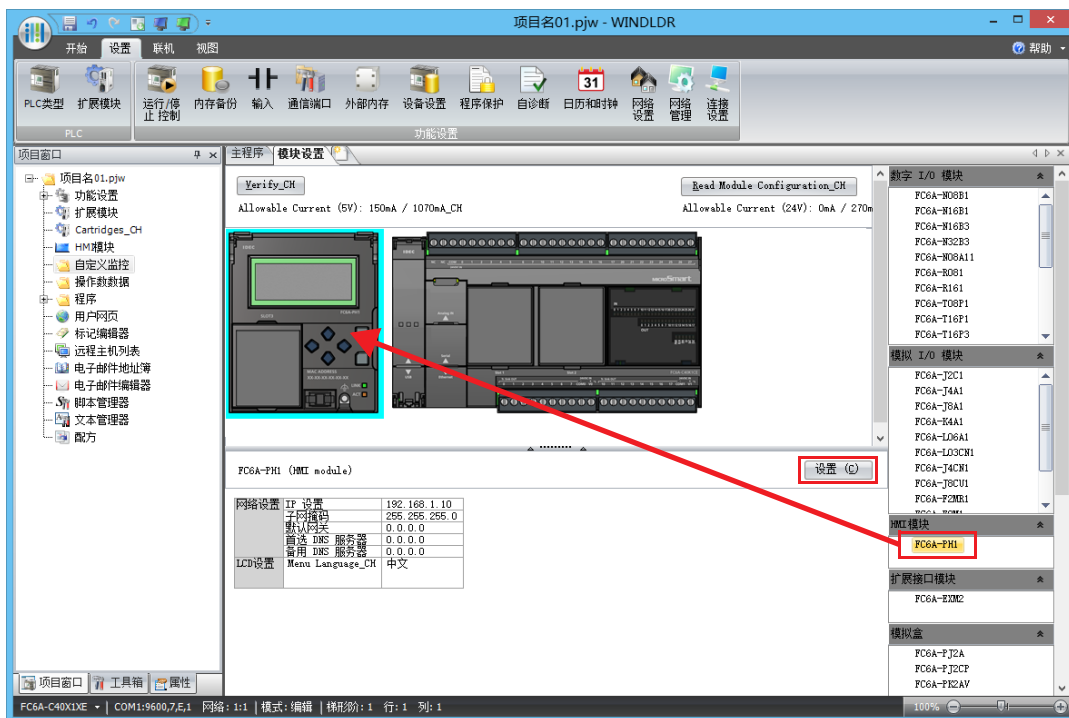
错误代码	说明	可能的原因
451	请求的行动已取消	电子邮件发送服务器未正常工作
452	系统空间不足	电子邮件服务器的空间不足
500	语法错误、无法识别的指令	发送了电子邮件服务器无法解释的指令
501	参数或引数的语法错误	指令的参数不正确
502	未安装指令	发送了电子邮件服务器未安装的指令
504	未安装指令的参数	发送了电子邮件服务器未安装的指令参数
521	不接收邮件	视为垃圾邮件
530	访问被拒绝	清除“发送电子邮件前需要身份验证”复选框
535	验证错误	验证的账户名称或密码不正确
550	不可使用邮箱	因策略上的理由拒绝了指令
552	超过分配容量	信息过长
554	交易失败（或开始连接时的响应时无 SMTP 服务）	原因为电子邮件服务器

12: EMAIL发送功能

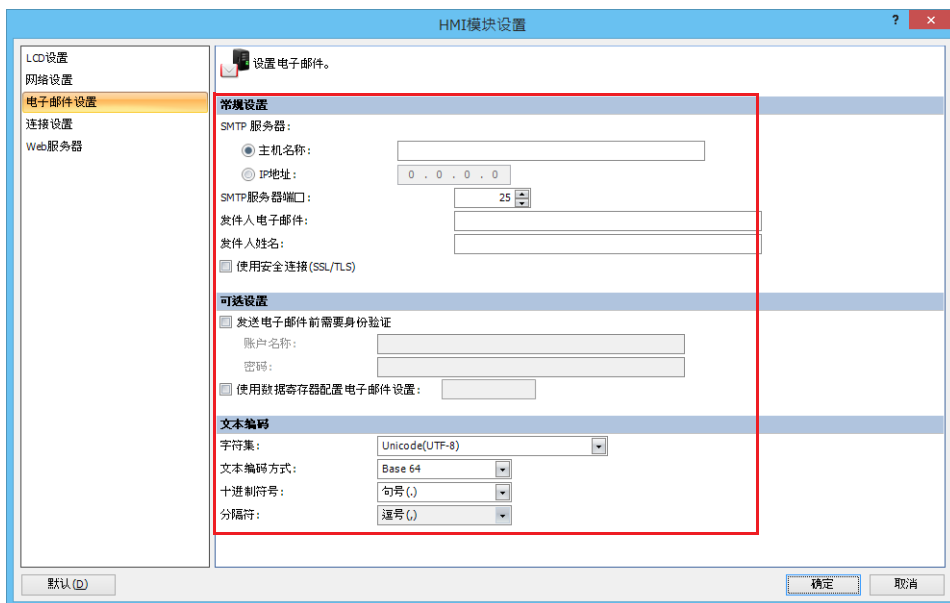
设置 WindLDR

1. 可通过模块构成编辑器进行 HMI 模块的 EMAIL 设置。
在“设置”选项卡的“PLC”中单击“扩展模块”。
2. 单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮。
将显示 HMI 模块的设置对话框。

注释：在项目窗口中双击“HMI 模块”，也可显示 HMI 模块的设置对话框。



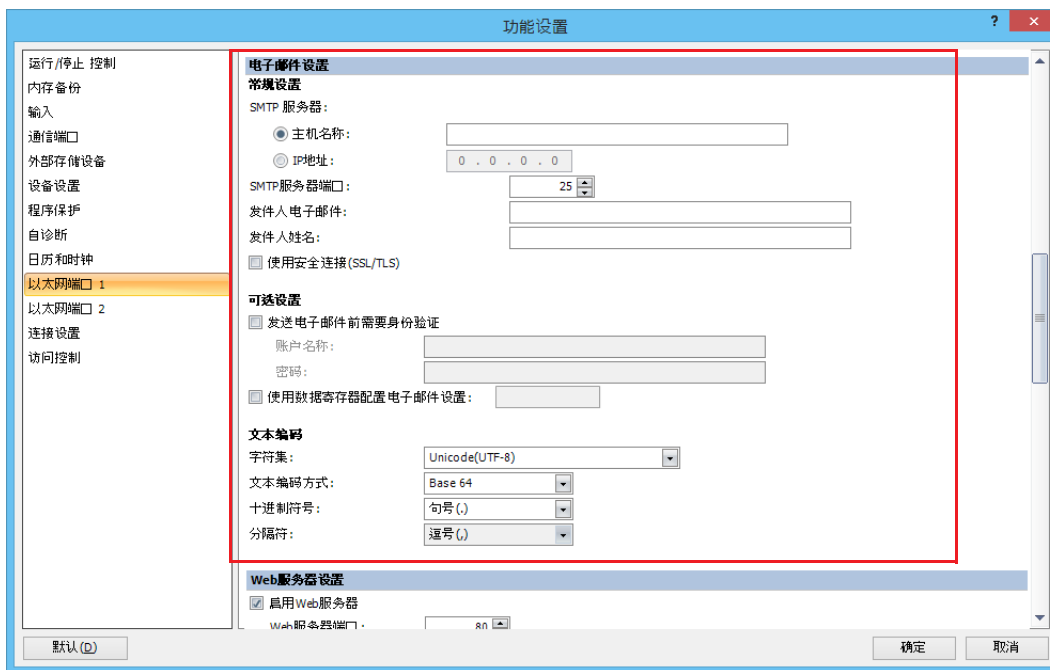
3. 单击“电子邮件设置”选项卡。
4. 设置常规设置、可选设置和文本编码的各设置项目。



5. 将用户程序下载到 FC6A 型 MICROSmart。
至此，完成电子邮件设置。

Plus CPU 模块的以太网端口 1

1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。
2. 在“电子邮件设置”中设置常规设置、可选设置和文本编码的各设置项目。



3. 将用户程序下载到 Plus CPU 模块。
至此，完成电子邮件设置。

■ 常规设置

HMI模块设置

LCD设置
网络设置
电子邮件设置
连接设置
Web服务器

设置电子邮件。

常规设置

(1) SMTP 服务器:

● 主机名称:

● IP地址:

(2) SMTP服务器端口:

(3) 发件人电子邮件:

(4) 发件人姓名:

(5) 使用安全连接(SSL/TLS)

可选设置

(6) 发送电子邮件前需要身份验证

(7) 账户名称:

(8) 密码:

(9) 验证方法:

(10) 使用数据寄存器配置电子邮件设置:

文本编码

(11) 字符集:

(12) 文本编码方式:

(13) 十进制符号:

(14) 分隔符:

默认(D) 确定 取消

(1) SMTP 服务器

指定电子邮件发送服务器（用于发送电子邮件）的 IP 地址或主机名。最多可输入 40 个 ASCII 字符。

(2) SMTP 服务器端口

指定电子邮件发送服务器的端口号。通常，SMTP 端口号使用 25；SMTP-AUTH 端口号使用 587；SMTPs 端口号使用 465。

(3) 发件人电子邮件

指定包含在电子邮件发件人字段中的电子邮件地址，邮件将从 FC6A 型 MICROSmart 发送。最多可输入 40 个 ASCII 字符。

(4) 发件人姓名

指定包含在电子邮件发件人字段中的名称，邮件将从 FC6A 型 MICROSmart 发送。最多可输入 40 个 ASCII 字符。

(5) 使用安全连接 (SSL/TLS)

如果所使用的电子邮件发送服务器中需要 SSL 通信或 TLS 通信，应选中该复选框。

注释:

- 在使用 IP 地址指定 SMTP 服务器时，值的高位数字中的 0 将被忽略。例如，IP 地址“192.168.1.234”和“192.168.001.234”视为相同的 IP 地址。
- 在使用 SMTP 服务器名称指定 SMTP 服务器时，必须使用 DNS 解析 SMTP 服务器主机名。有关 DNS 和 DNS 服务器设置的详情，请参见第 3-3 页上的“网络设置”。

设置示例：发件人的电子邮件地址和发件人名称将被分组并显示在电子邮件的发件人字段中。如果发件人的电子邮件地址为 test@sample.com，发件人的名称为 Test Mail，将在发件人字段中显示如下文本：

Test Mail <test@sample.com>

■ SMTP 验证

通过电子邮件发送服务器发送电子邮件时，有时需要进行 SMTP-AUTH 或 SMTPs 的验证。

FC6A 型 MICROSmart 支持 SMTP-AUTH 或 SMTPs 的验证，因此若是需要验证的电子邮件发送服务器，则可使用该功能发送电子邮件。

关于要使用的电子邮件发送服务器是否需要验证，请咨询相应的电子邮件发送服务器的管理员。

(6) 发送电子邮件前需要身份验证

检查电子邮件发送服务器是否需要 SMTP 验证登录方式。

(7) 账户名称

最多可输入 40 个 ASCII 字符。

(8) 密码

最多可输入 40 个 ASCII 字符。

(9) 验证方法

设置用于验证的验证方法。验证方法可以从以下 5 种类型中设定。

验证方法	说明
自动	使用 CRAM-MD5、DIGEST-MD、LOGIN 和 PLAIN 认证方法与外发邮件服务器进行通信，依次使用成功连接的认证方法。
CRAM-MD5	使用 CRAM-MD5 连接到外发送服务器。
DIGEST-MD5	使用 DIGEST-MD5 连接到外发送服务器。
LOGIN	使用 LOGIN 连接到外发送服务器。
PLAIN	使用 PLAIN 连接到外发送服务器。

注释：如果你把验证方法设置为“自动”，你可能无法连接到某些外发送服务器。在这种情况下，要设置在外发送服务器上使用的验证方法。

12: EMAIL 发送功能

(10) 使用数据寄存器配置电子邮件设置

选中该项目的复选框时，应将常规设置及验证的设置使用数据寄存器的字符串及数值进行操作。使用 107 字的数据寄存器（从指定数据寄存器开始）中存储的设置发送电子邮件。

对于每个设置项目，起始地址和结束地址以及设置值的相关详细信息如下所示。

设置项目	数据类型	所用字数	设置项目起始地址	设置项目结束地址	设置值	
SMTP 服务器	字符串 (40 个字符)	21* ¹	起始数据寄存器	起始数据寄存器 +20	以字符串形式，指定电子邮件发送服务器（用于发送电子邮件）的 IP 地址或主机名。 ^{*2}	
SMTP 服务器端口	十进制值	1	起始数据寄存器 +21		以十进制值形式，指定电子邮件发送服务器（用于发送电子邮件）的端口号。	
发件人电子邮件	字符串 (40 个字符)	21* ¹	起始数据寄存器 +22	起始数据寄存器 +42	以字符串形式，指定包含在电子邮件发件人字段中的电子邮件地址，邮件将从 FC6A 型 MICROSmart 发送。 ^{*2}	
发件人姓名	字符串 (40 个字符)	21* ¹	起始数据寄存器 +43	起始数据寄存器 +63	以字符串形式，指定包含在电子邮件发件人字段中的名称，邮件将从 FC6A 型 MICROSmart 发送。 ^{*2}	
发送电子邮件前需要身份验证 / 验证方法	十进制值	1	起始数据寄存器 +64	设置是否需要验证，以及所使用的外发送服务器上的验证方法。		
				值	是否需要验证	验证方法
				0	不需要	—
				1	需要	自动
				3		CRAM-MD5
				5		DIGEST-MD5
7	LOGIN					
9	PLAIN					
账户名称	字符串	21* ¹	起始数据寄存器 +65	起始数据寄存器 +85	以字符串的形式，指定用于验证的账户名。 ^{*2}	
密码	字符串	21* ¹	起始数据寄存器 +86	起始数据寄存器 +106	以字符串的形式，指定用于验证的密码。 ^{*2}	
使用安全连接 (SSL/TLS)	十进制值	1	起始数据寄存器 +107		如果所使用的电子邮件发送服务器中需要 SSL 通信或 TLS 通信，应设置 1。如果不需要 SSL 通信或 TLS 通信，应设置 0。	

*1 字符串设为“所用字数 - 1”字时为有效的设置值。为表示字符串结尾，将像在其中存储了 0000h 一样处理最后一字，而无论数据寄存器中存储的实际值为何。

*2 如果字符串短于“所用字数 - 1”字，则使用 00h 填充字符串后的数据。

设置示例： 下方所示的电子邮件设置是通过数据寄存器配置的。

SMTP 服务器 : smtp.example.com
 SMTP 服务器端口 : 587
 发件人电子邮件 : test@example.com
 发件人姓名 : Test
 发送电子邮件前需要身份验证 : 已确认
 账户名称 : test_account
 密码 : test_password
 验证方法 : LOGIN
 使用安全连接 (SSL/TLS) : 已确认

当指定 D100 作为起始数据寄存器时，按照如下所示设置数据寄存器 D100 到 D207 的值。

设置项目	设置项目 起始地址	设置项目 结束地址	设置值										
SMTP 服务器	D100	D120	数据寄存器	D100	D101	D102	D103	D104	D105	D106	D107	D108	D109 至 D120
			ASCII	's' 'm'	't' 'p'	'!' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'!' 'c'	'o' 'm'	'\0' '\0'	'\0' '\0'
			值 (十六进制)	736Dh	7470h	2E65h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h	0000h
SMTP 服务器端口	D121		数据寄存器	D121									
			值 (十进制)	587									
发件人电子 邮件	D122	D142	数据寄存器	D122	D123	D124	D125	D126	D127	D128	D129	D130	D131 至 D142
			ASCII	't' 'e'	's' 't'	'@' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'!' 'c'	'o' 'm'	'\0' '\0'	'\0' '\0'
			值 (十六进制)	7465h	7374h	4065h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h	0000h
发件人姓名	D143	D163	数据寄存器	D143	D144	D145	D146 至 D163						
			ASCII	'T' 'e'	's' 't'	'\0' '\0'	'\0' '\0'						
			值 (十六进制)	5465h	7374h	0000h	0000h						
发送电子 邮件前需要 身份验证 / 验证方法	D164		数据寄存器	D164									
			值 (十进制)	7									
账户名称	D165	D185	数据寄存器	D165	D166	D167	D168	D169	D170	D171	D172 至 D185		
			ASCII	't' 'e'	's' 't'	'_' 'a'	'c' 'c'	'o' 'u'	'n' 't'	'\0' '\0'	'\0' '\0'		
			值 (十六进制)	7465h	7374h	5F61h	6363h	6F75h	6E74h	0000h	0000h		
密码	D186	D206	数据寄存器	D186	D187	D188	D189	D190	D191	D192	D193 至 D206		
			ASCII	't' 'e'	's' 't'	'_' 'p'	'a' 's'	's' 'w'	'o' 'r'	'd' '\0'	'\0' '\0'		
			值 (十六进制)	7465h	7374h	5F70h	6173h	7377h	6F72h	6400h	0000h		
使用安全连接 (SSL/TLS)	D207		数据寄存器	D207									
			值 (十进制)	1									

注释： 数据寄存器定义的字符串由 1 个或多个连续的数据寄存器组成。1 字的数据当作 2 字节进行处理，并按照高位字节到低位字节的顺序使用，结尾字符为 00h。

12: EMAIL 发送功能

特殊数据寄存器 / 特殊内部继电器

初始化电子邮件发送服务器设置

“功能设置”中设定的值可以使用特殊内部继电器 M8211，作为初始值反映到对应的数据寄存器。

特殊内部继电器

设备地址	说明	运行
M8211	初始化 HMI 模块的电子邮件发送服务器设置	开启此继电器后，功能设置中设定的值可以作为初始值反映到对应的数据寄存器。
M8344	初始化以太网端口 1 的电子邮件发送服务器设置	开启此继电器后，功能设置中设定的值可以作为初始值反映到对应的数据寄存器。

注释：字符串可以存储在数据寄存器中。字符串从指定数据寄存器开始。每个字符串的结尾通过 00h 来指定，其存储在数据寄存器中值的高位字节或低位字节上。1 字的数据当作 2 字节进行处理，并按照高位字节到低位字节的顺序使用。

■ 文本编码

(11) 字符集

可以指定电子邮件主题、正文和附件所用的字符集。

ASCII：当电子邮件主题和正文仅由 ASCII 字符组成时指定。

日文 (ISO-2022-JP)：当电子邮件主题和正文由 ASCII 和日文字符组成时指定。附件所用的字符集为 Shift_JIS。

中文 (GB2312)：当电子邮件主题和正文由简体中文字符组成时指定。

西欧 (ISO-8859-1)：当电子邮件主题和正文由西欧字符组成时指定。

Unicode (UTF-8)：当使用 Unicode 字符时指定。

一般情况下，Unicode 可以使用任何字符。从 FC6A 型 MICROSmart 发送的电子邮件可能无法到达收件人或可能无法在邮件程序中正确查看，视中继电子邮件的电子邮件发送服务器或收件人使用的邮件程序而定。

当使用 ASCII 时，虽然仅可使用 ASCII 字符，但却可以相同的方式在任何邮件程序中查看电子邮件。

请与电子邮件发送服务器的管理员联系，以确认所支持的字符集。为收件人的邮件程序选择适当的字符集。

(12) 文本编码方式

可以指定电子邮件正文的编码格式。不能发送使用 8 位字符的电子邮件，视中继电子邮件的电子邮件发送服务器而定。在此情况下，可使用 Base64 编码，将 8 位字符转换成 7 位字符。

无：电子邮件正文未编码。

Base 64：使用 Base64 格式对电子邮件正文进行编码。

(13) 十进制符号

可以指定浮点值的十进制符号。当嵌入在电子邮件正文中的数据寄存器的数据类型为浮动状态时，数据寄存器值将被转换并在电子邮件正文中显示为浮点值。

句号 (.)：使用点号 “.”(2Eh)。

逗号 (,)：使用逗号 “,”(2Ch)。

(14) 分隔符

通过选定的“十进制符号”自动确定附件 CSV 文件的分隔符。

十进制符号	分隔符字符
点号 ! (2Eh)	逗号 ! (2Ch)
逗号 ! (2Ch)	分号 ! (3Eh)

注释：文本编码在 HMI-以太网端口与以太网端口 1 中为相同设置。

例如，在“HMI 模块设置”对话框的“电子邮件设置”中设置的文本编码内容，将自动反映到“功能设置”对话框的“以太网端口 1”文本编码中。

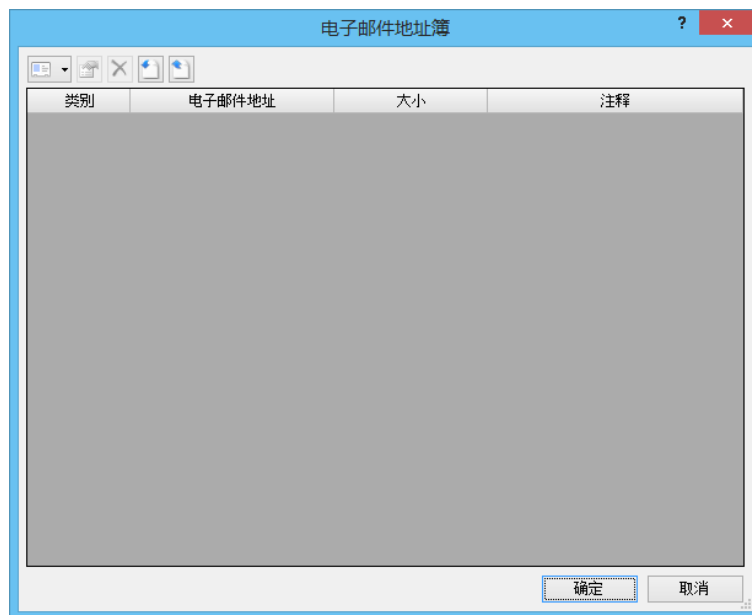
电子邮件地址簿

可以在电子邮件地址簿对话框中设置电子邮件地址和电子邮件地址组。可以将电子邮件地址归入电子邮件地址组。可在各电子邮件中使用电子邮件地址组指定一组电子邮件地址。

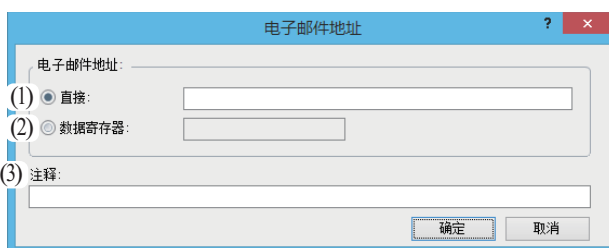
电子邮件地址和电子邮件组合计最多可注册 255 件。

设置 WindLDR

1. 在“项目窗口”中双击“电子邮件地址簿”。
出现“电子邮件地址簿”对话框。



2. 单击“新的电子邮件地址”的“▼”，然后单击“新的电子邮件地址”。或选择现有的电子邮件地址，单击“编辑”。
将显示“电子邮件地址”对话框。
3. 设置各设置项目。



(1) 直接

在电子邮件地址簿对话框中指定要注册的电子邮件地址。最多可输入 40 个 ASCII 字符。

(2) 数据寄存器

指定用于存储电子邮件地址的数据寄存器。发送电子邮件时，数据寄存器中存储的字符串用作电子邮件收件人（收件人和抄送）的电子邮件地址。这些寄存器从指定数据寄存器开始，最多 30 字。通过使用数据寄存器的字符串指定电子邮件地址时，电子邮件地址的长度被视为固定 60 字节。使用数据寄存器指定所有电子邮件地址时，最多可指定 8 个电子邮件收件人（收件人和抄送）的电子邮件地址。

(3) 注释

可以分配电子邮件地址的注释。注释的内容或长度对 CPU 模块的运行没有影响。

12: EMAIL 发送功能

设置示例：要通过指定数据寄存器 D100 发送电子邮件到 test@example.com，可将值存储在下列数据寄存器中。

数据寄存器	D100	D101	D102	D103	D104	D105	D106	D107	D108
ASCII	't' 'e'	's' 't'	'@' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'.' 'c'	'o' 'm'	'\0' '\0'
值（十六进制）	7465h	7374h	4065h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h

要通过指定数据寄存器 D123 发送电子邮件到 you@example.com，可将值存储在下列数据寄存器中。

数据寄存器	D123	D124	D125	D126	D127	D128	D129	D130
ASCII	y' 'o'	'u' '@'	'e' 'x'	'a' 'm'	'p' 'l'	'e' 'l'	'c' 'o'	'm' '\0'
值（十六进制）	796Fh	7540h	6578h	616Dh	706Ch	652Eh	636Fh	6D00h

注释：

- 数据寄存器定义的字符串由 1 个或多个连续的数据寄存器组成。1 字的数据当作 2 字节进行处理，并按照高位字节到低位字节的顺序使用，结尾字符为 00h。
 - 如果从指定为电子邮件地址的数据寄存器起的 30 字（60 字节）中不包含 00h，则全部 60 字节都将用作电子邮件地址，而且 FC6A 型 MICROSmart 将按照第 61 字节为 00h 运行。
4. 单击“确定”。
 5. 单击“新的电子邮件地址”的“▼”，然后单击“新的电子邮件组”。或选择现有的电子邮件组，单击“编辑”。
将显示“电子邮件组”对话框。
 6. 设置各设置项目。



(1) 组名

在电子邮件地址簿对话框中指定要注册的电子邮件地址组的名称。

(2) 注释

可以分配电子邮件地址组的注释。注释的内容或长度对 CPU 模块的运行没有影响。

未添加到电子邮件地址组的电子邮件地址显示在左侧的列表框中。电子邮件地址以其在电子邮件地址簿对话框中注册的顺序列出。若要将电子邮件地址添加到电子邮件地址组，请选择要添加的电子邮件地址，然后单击 >> 按钮。所选的电子邮件地址移至右侧的列表框并列于电子邮件的底部。

电子邮件地址组的电子邮件地址显示在右侧的列表框中。电子邮件地址以其添加到电子邮件地址组的顺序列出。若要将电子邮件地址从电子邮件地址组中移除，请选择要移除的电子邮件地址，然后单击 << 按钮。所选的电子邮件地址移至左侧的列表框并列于电子邮件的底部。

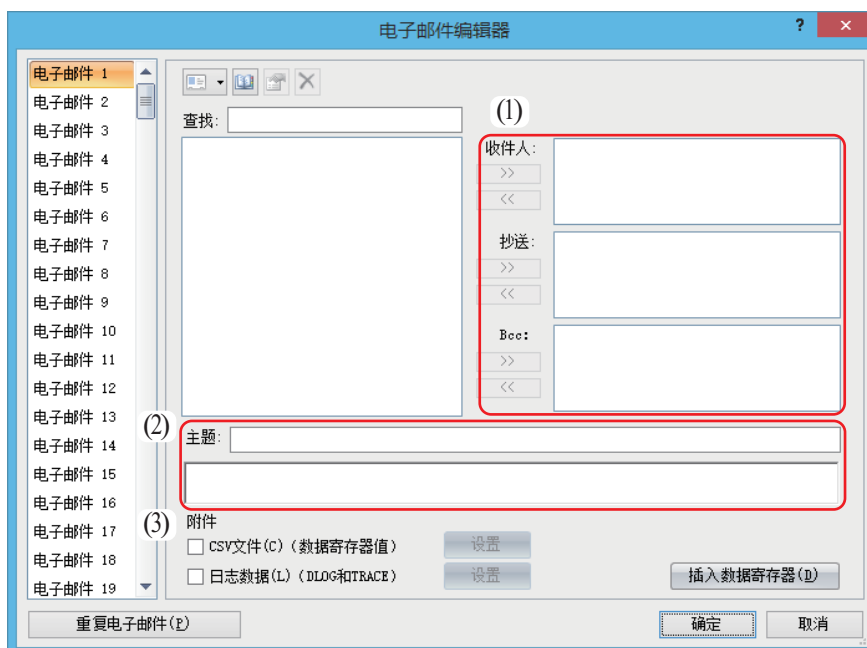
若要从电子邮件地址簿中删除不用的电子邮件地址或电子邮件地址组，请选择要删除的电子邮件地址或电子邮件地址组，然后在电子邮件地址簿对话框中单击删除按钮。如果在电子邮件地址组中使用已删除的电子邮件地址，则所删除的电子邮件地址将从所有的电子邮件地址组中移除。

7. 单击“确定”。

配置即完成。

电子邮件编辑器

可在电子邮件编辑器对话框中设置电子邮件。应就每份电子邮件设置以下五个参数。



电子邮件编辑器中的设置

(1) 电子邮件收件人“收件人”、“抄送”和“Bcc”

可以将电子邮件地址或电子邮件组指定为收件人“收件人”、“抄送”或“Bcc”文本的最大大小为 512 字节。电子邮件地址之间用逗号 (,) 字符隔开。例如，电子邮件组“ccc”包含两个电子邮件地址“aa@example.com”（15 字节，含逗号）和“bbb@example.com”（16 字节，含逗号），总大小为 31 字节。如果电子邮件地址的大小为 30 字节，则在“收件人”、“抄送”和“Bcc”中可分别指定 16 个电子邮件地址。一封电子邮件可同时发送到 32 个电子邮件地址。

(2) “主题”和“正文”

“电子邮件主题”的文本大小上限为 256 字节，“正文”上限为 4,096 字节。电子邮件主题和正文由下列元素组成。

- 使用在“功能设置”中“电子邮件设置”下面设定的“字符集”的字符串
- 通过插入数据寄存器嵌入到正文中的数据寄存器值的数值字符串
- 空格和换行符

“主题”和“正文”的大小是由上述元素组合而成的文本的大小。

根据“功能设置”中“电子邮件设置”下的“文本编码”进行文本编码的结果，作为电子邮件发送出去。

仅使用单字节 ASCII 字符时，电子邮件主题可输入约 200 个字符，正文可输入约 3,500 个字符。使用多字节字符时，电子邮件主题可输入约 100 个字符，正文可输入约 2,000 个字符。

(3) 附件

一封电子邮件可以附加一个 CSV 文件。附件可以包含文本和数据寄存器值。CSV 文件的文件名固定为“data.csv”。附件的大小上限为 4,096 字节。

附件由下列元素组成。

- 使用在“功能设置”中“电子邮件设置”下面设定的“字符集”的字符串
- 通过插入数据寄存器嵌入到正文中的数据寄存器值的数值字符串
- 空格、分隔符和换行符

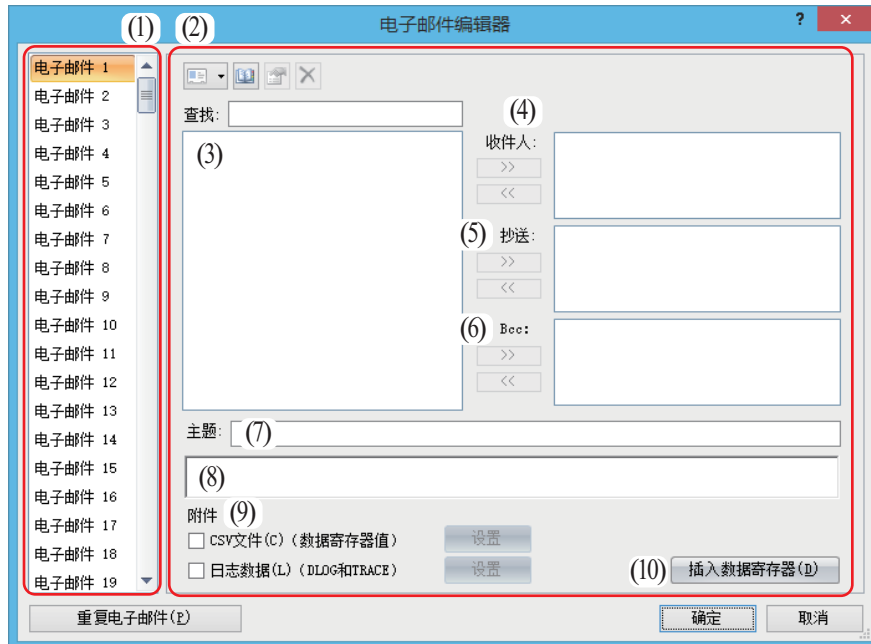
“附件”的大小是由上述元素组合而成的文本的大小。

根据“功能设置”中“电子邮件设置”下的“文本编码”进行文本编码的结果，作为附件添加到电子邮件中。

此外，通过 Plus CPU 模块的以太网端口 1 发送电子邮件时，可添加以 DLOG 指令或 TRACE 指令创建的日志文件。

设置 WindLDR

1. 在“项目窗口”中双击“电子邮件编辑器”。
出现“电子邮件编辑器”对话框。
2. 创建电子邮件。



(1) 电子邮件

将显示已注册的电子邮件。

(2) 电子邮件的内容

将显示所选电子邮件的内容。

由“收件人”“抄送”“Bcc”“主题”“正文”和“附件”这六大项目组成。

(3) 项目

将在电子邮件地址中显示已注册的项目。

在“收件人”、“抄送”或“Bcc”中选择要追加的项目，单击要追加侧的“>>”，可将相应的项目追加到“收件人”、“抄送”或“Bcc”中。

在“收件人”、“抄送”或“Bcc”列表框的项目中，选择要删除的项目，单击要删除侧的“<<”，可将相应项目从“收件人”、“抄送”或“Bcc”删除。

(4)“收件人”/ (5)“抄送”/ (6)“Bcc”

指定发送电子邮件的收件人。

(7)“主题”

指定发送电子邮件的主题。

(8)“正文”

指定电子邮件正文。

以多行形式写入电子邮件正文。

(9) 附件

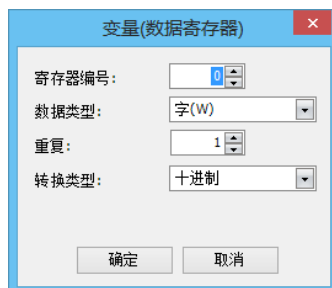
在电子邮件中添加文件时，应选中复选框。

CSV 文件 (C) (数据寄存器值) : 包含文本或数据寄存器值的附件 (CSV 文件) 通过附件编辑器设置。单击“编辑”按钮，将显示附件编辑器。项目及其布局可以在文件大小上限的范围中自由配置。有关详情，请参见第 12-16 页上的“附件编辑器”。

日志数据 (L) (DLOG 和 TRACE) : Plus CPU 模块时，可添加以 DLOG 指令或 TRACE 指令创建的日志文件。在“附加日志数据”对话框中设置。单击“设置”按钮，将显示“附加日志数据”对话框。有关详情，请参见第 12-19 页上的“附加日志数据设置”。

(10) 插入数据寄存器

出现“变量(数据寄存器)”对话框。



当 Web 服务器发送电子邮件时，可将数据寄存器值嵌入到电子邮件正文中。

项目	说明	
寄存器编号	指定数据寄存器编号。	
数据类型	字 (W)	存储在指定数据寄存器中 1 字值被转换为无符号的 16 位值。
	整数 (I)	存储在指定数据寄存器中 1 字值被转换为有符号的 15 位值。
	双字 (D)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为无符号的 32 位值。
	长整 (L)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为有符号的 31 位值。
	浮点 (F)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为 IEEE754 格式的浮点值。可在 1 到 7 之间指定小数位数。
	字符串	存储在指定数据寄存器中的值被转换为以文本编码的字符集设置的字符串。以字为单位，最多可设置 1 ~ 512 个字的字符数（包含终端 NULL）。

当选择字 (W) 和双字 (D) 时，可以指定转换类型。例如，当数据寄存器值为 4660 (1234h) 时，数据寄存器值将被转换并嵌入在电子邮件正文中，如下所示：

十进制：4660

十六进制：1234

当设置重复次数时，可将连续数据寄存器中的值嵌入在电子邮件正文中。在数据寄存器值之间插入空格。例如，当数据寄存器值为 (D100) = 1234h、(D101) = 5678h 以及 (D102) = ABCDh，数据寄存器编号为 100，数据类型为字，重复次数为 3，转换类型为十六进制时，以下文本将嵌入在电子邮件正文中：

1234 5678 ABCD

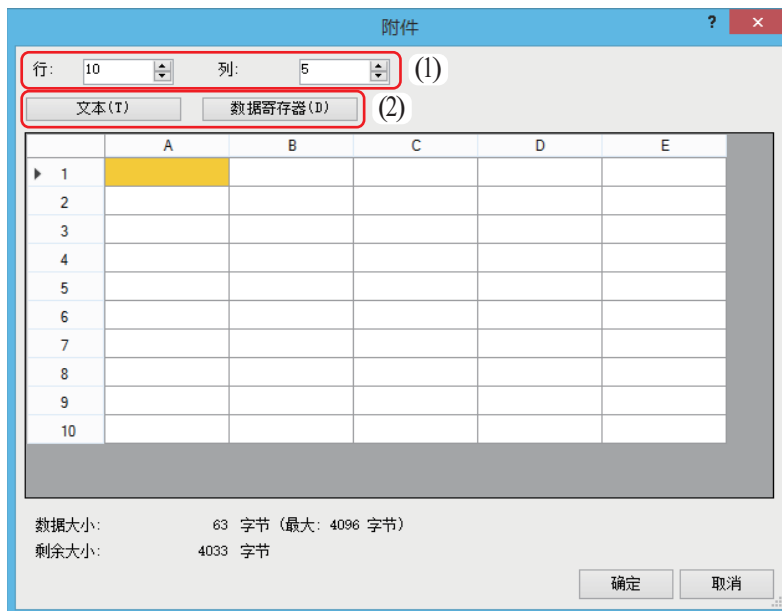
该电子邮件编辑器所创建的正文将根据“功能设置”的“电子邮件设置”所指定的“编辑器方式”的设置进行编码和发送。

3. 单击“确定”。

附件编辑器

编辑添加在电子邮件中的 CSV 文件内容。

CSV 文件中可以包含所需的文本和数据寄存器值。CSV 文件的文件名固定为“data.csv”。



附件编辑器中的设置

(1) “行”、“列”

可以更改 CSV 文件中的行数和列数。行数和列数可以在 1 到 64 范围内指定。如果缩小编辑范围，则编辑范围之外的设置值将被清除。

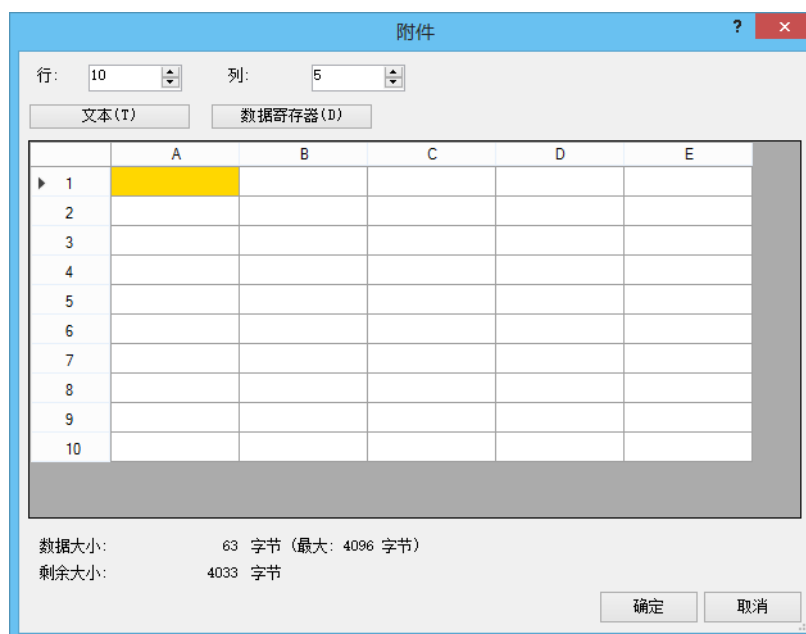
(2) “文本”、“数据寄存器”

设置表格中显示的文本和数据寄存器，以确定附件的内容。CSV 文件的大小上限为 4,096 字节。

注释：功能设置中配置的字符集、分隔符和十进制符号将应用到所有附件。有关这些设置的详情，请参见第 12-4 页上的“设置 WindLDR”。

设置 WindLDR

1. 选择“电子邮件编辑器”中的“附件”复选框，然后单击“编辑”。
出现“附件”对话框。
2. 指定“行”和“列”。
设置 CSV 文件中的行数和列数，以确定编辑范围。



可以在对话框的上半部分修改 CSV 文件的内容。对话框的底部显示当前文件大小，以及可用于编辑内容的剩余文件大小。您可以通过更改对话框大小来扩展网格区域。文件大小包括 CSV 文件中包含的分隔符和换行符。

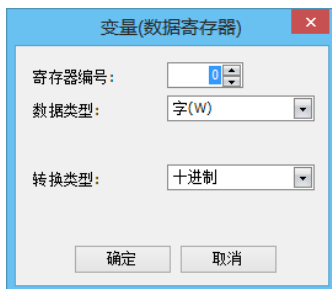
3. 选择任意单元格，然后单击“文本”。
出现“文本”对话框。
4. 在所选单元格中输入任意文本。
可以指定的文本长度上限为 63 字节。



5. 单击“确定”。
返回到“附件”对话框。
6. 选择任意单元格，然后单击“数据寄存器”。
出现“变量（数据寄存器）”对话框。

12: EMAIL 发送功能

7. 设置各设置项目。



发送电子邮件时的数据寄存器值可以包含在选定单元格中。

项目	说明	
寄存器编号	指定数据寄存器编号。	
数据类型	字 (W)	存储在指定数据寄存器中 1 字值被转换为无符号的 16 位值。
	整数 (I)	存储在指定数据寄存器中 1 字值被转换为无符号的 16 位值。
	双字 (D)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为无符号的 32 位值。
	长整 (L)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为有符号的 31 位值。
	浮点 (F)	存储在两个连续的数据寄存器（以指定的数据寄存器开始）中 2 字值被转换为 IEEE754 格式的浮点值。可在 1 到 7 之间指定小数位数。

当选择字 (W) 和双字 (D) 时，可以指定转换类型。例如，当数据寄存器值为 4660 (1234h) 时，数据寄存器值将被转换并嵌入在电子邮件正文中，如下所示：

十进制：4660

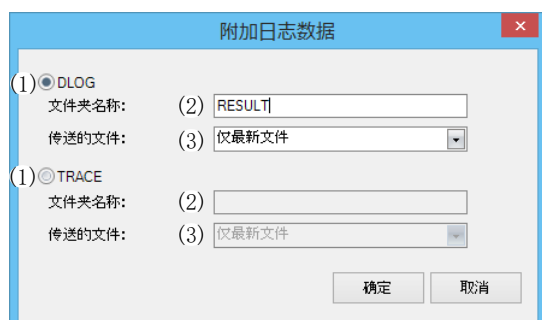
十六进制：1234

8. 单击“确定”。
返回到“附件”对话框。

9. 单击“确定”。

附加日志数据设置

设置添加在电子邮件中的日志文件。



设置项目

(1) 日志文件类型

选中要添加的日志文件类型的复选框。

“DLOG”、“TRACE”

(2) 文件夹名称

指定使用 DLOG 指令或 TRACE 指令指定的日志文件的保存位置文件夹名称。最多可输入 8 个字符。

(3) 传送的文件

从以下选项中选择日志文件的指定方法。

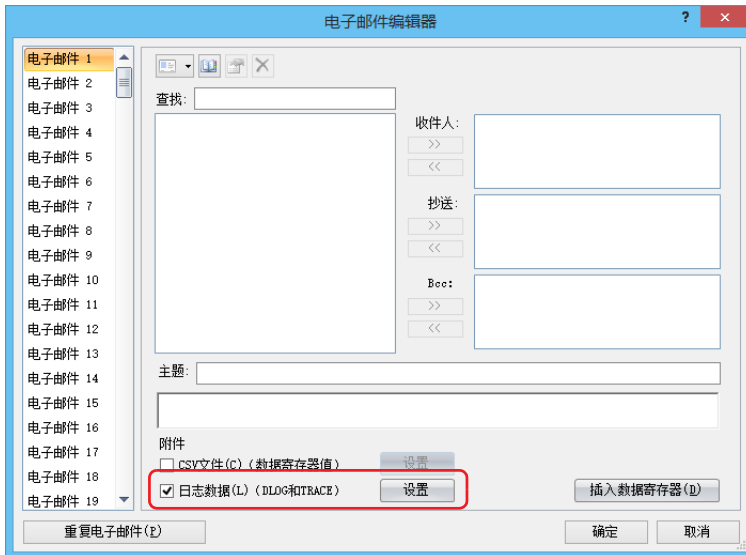
- 在 24 小时内刷新的文件
- 在 7 天内刷新的文件
- 仅最新文件

注释：可添加的日志文件最大数为 7 个文件。第 8 个文件以后不添加日志文件，错误代码 11 将存储到 EMAIL 指令的目标 2 中。

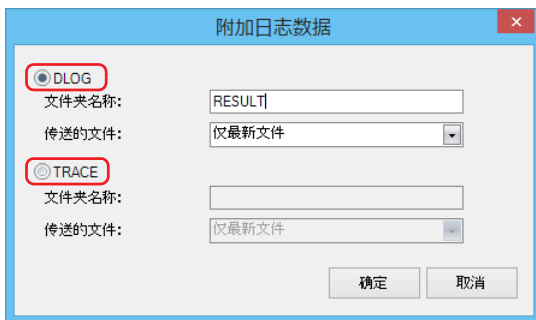
12: EMAIL发送功能

设置 WindLDR

1. 选中“电子邮件编辑器”的“日志数据（DLOG 和 TRACE）”复选框，然后单击“设置”按钮。将显示“附加日志数据”对话框。



2. 选中添加在电子邮件中的日志文件类型的复选框。



3. 在“文件夹名称”中指定日志文件的保存位置文件夹名称。最多可输入 8 个字符。
4. 在“传送的文件”处从以下选项中选择日志文件的指定方法。
 - 在 24 小时内刷新的文件
 - 在 7 天内刷新的文件
 - 仅最新文件
5. 单击“确定”按钮。

配置即完成。

13: Web 服务器

简介

本章介绍 FC6A 型中的 Web 服务器功能。

概述

Plus CPU 模块和 HMI 模块支持网络服务器功能。

Web 服务器功能可显示三种类型的网页。

网页	说明
系统网页	可监控 FC6A 型的运行状态并在网络浏览器上更改设备值。
用户网页	用户可使用 HTML 或 JavaScript 自由创建。
自定义网页	可使用 WindLDR 的网页编辑器创建。

可使用这些网页为每个项目创建一个简易网站。

网页可以存储在 Plus CPU 模块的内存、SD 记忆卡或 HMI 模块的内存中。有关详情，请参见第 13-3 页上的“网页保存位置”。

Plus CPU 模块的 Web 服务器功能支持 SSL/TLS，可防止数据被第三方窃听、篡改和假冒。

支持机型及通信端口

支持 Web 服务器功能的机型和通信端口如下所示。

是：支持 Web 服务器功能

否：不支持 Web 服务器

—：CPU 模块未配备

CPU 模块	通信端口		
	以太网端口 1	以太网端口 2	HMI- 以太网端口
All-in-One CPU 模块	否	—	是 *2*3
CAN J1939 All-in-One CPU 模块			
Plus CPU 模块	是 *1	否	

*1 支持 SSL/TLS 通信。

*2 当 HMI 模块连接到 CPU 模块，并添加 HMI-Ethernet 端口。

*3 不支持 SSL/TLS 通信。

操作环境

推荐用于浏览下载到 FC6A 型的网页的操作系统和网络浏览器如下。

启用 SSL/TLS 通信时

OS	支持浏览器
Windows 10	Google Chrome 90 或更高版本 Mozilla Firefox 88 或更高版本
Android	Google Chrome 90 或更高版本
iOS	Safari 14 或更高版本

禁用 SSL/TLS 通信时

OS	支持浏览器
Windows 10/8/7	Google Chrome 47 或更高版本 Mozilla Firefox 42 或更高版本 Microsoft Internet Explorer 11
Android	Google Chrome 47 或更高版本
iOS	Safari 8 或更高版本

注释：除使用推荐操作系统和推荐网络浏览器以外，仍可其他使用网络服务器功能，但在网页显示和操作上可能会遇到问题。

基本规格

网页保存位置

Plus CPU 模块可以设置网页保存的目的地。 All-in-One CPU 模块只能在 HMI 模块的内存中保存网页。

是：可保存

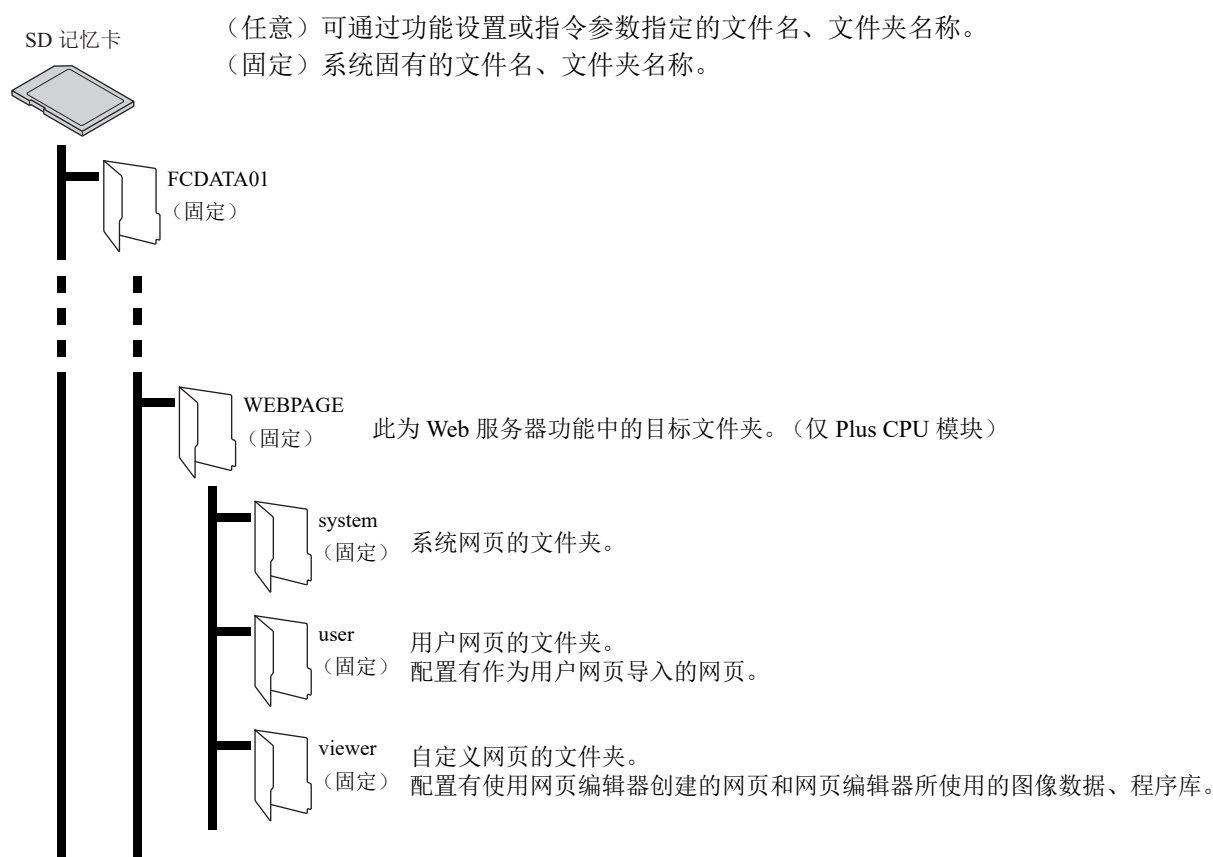
否：不可保存

CPU 模块	CPU 模块的内存	CPU 模块上插入的 SD 记忆卡	HMI 模块的内存
All-in-One CPU 模块	否	否	是 (最大 5 MB)
CAN J1939 All-in-One CPU 模块			
Plus CPU 模块	是 (最大 5 MB)	是 (最大 32GB*1)	

*1 一个文件的最大为 4 GB。

SD 记忆卡的网页保存位置

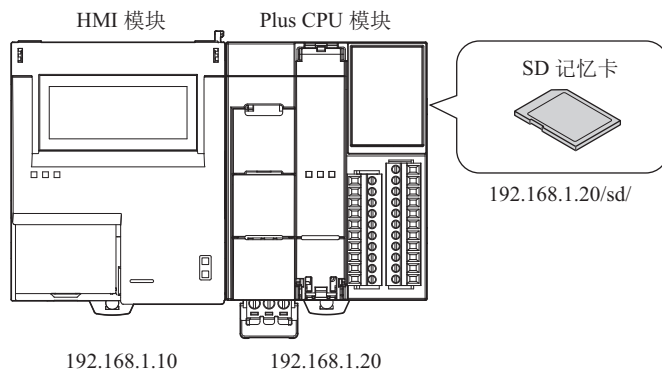
如果使用 Plus CPU 模块的网络服务器功能，可将网页保存到 SD 记忆卡中。将网页保存到 SD 记忆卡时，保存位置如下。



访问目的地和显示网页

访问下载的网页时，URL 因保存位置而异。

例如，Plus CPU 模块的 IP 地址为 192.168.1.20、Plus CPU 模块上所连接 HMI 模块的 IP 地址为 192.168.1.10 时，如下设置 URL。



网页保存位置	访问网页时的 URL
Plus CPU 模块的内存	http://192.168.1.20/
Plus CPU 模块上插入的 SD 记忆卡	http://192.168.1.20/sd/
HMI 模块的内存	http://192.168.1.10/

此外，将网页存储在 Plus CPU 模块的内存中并使用网络浏览器访问网络服务器时，将显示以下网页。

访问目的地	显示的网页
http://192.168.1.20/	显示在 Web 服务器设置的“重定向目标”中设置的网页。如果选中“禁用重定向”，则会发生错误。
http://192.168.1.20/system/	显示系统网页的 PLC 状态显示页面 ^{*1} 。
http://192.168.1.20/user/	如果用户文件夹中有名为 index.html 的文件，则会显示该 index.html。

*1 有关 PLC 状态显示页面的详细信息，请参见第 13-26 页上的“PLC 状态”。

注释：

- 访问 http://192.168.1.20/viewer/ 不会重定向到自定义网页。如果要显示自定义网页，请在 Web 服务器设置的“重定向目标”中设置自定义网页，然后访问 http://192.168.1.20/。
- 系统将提示您输入用户名和密码以访问该网页。有关详情，请参见第 13-9 页上的“(10) 用户设置”。

连接数

如果在 Plus CPU 模块的 Web 服务器上启用 SSL/TLS 通信，则最多可以建立 8 个连接。

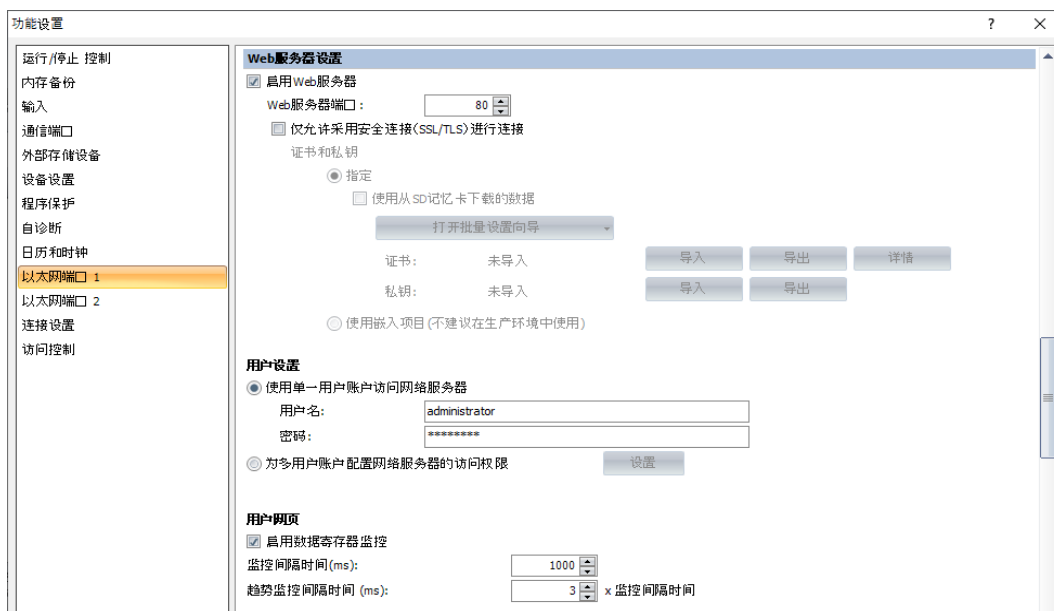
注释：根据连接到 Plus CPU 模块的 Web 服务器的客户端的设置，客户端在连接和浏览网页时使用的连接数根据设置而不同。可使用 D8822 检查当前连接数（Web 服务器当前连接数（Plus CPU 模块））。有关详情，请参见第 2-19 页上的“特殊数据寄存器”

Plus CPU 模块 Web 服务器

本节介绍在 Plus CPU 模块中使用 Web 服务器功能时的 Web 服务器设置。

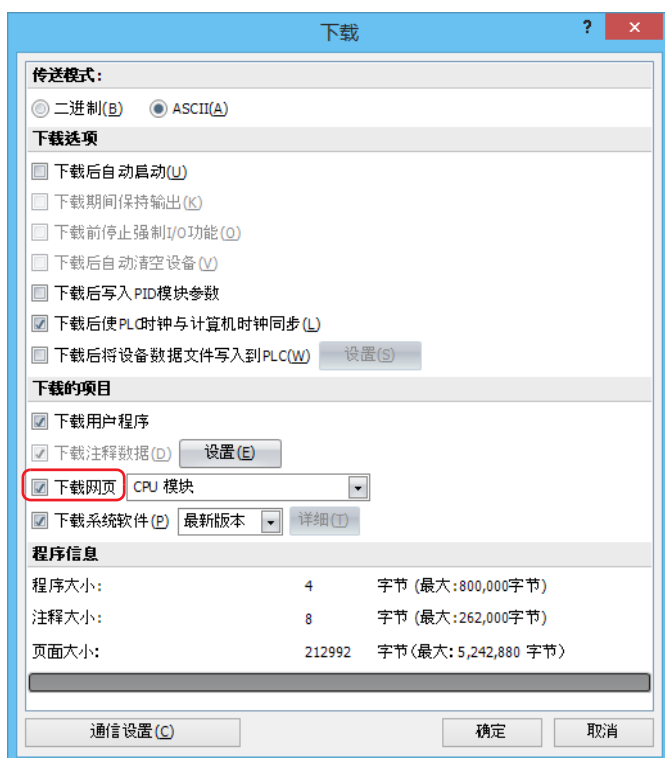
设置步骤

1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。
2. 在“Web 服务器设置”中设置 Web 服务器设置、网页、选项的各设置项目。



有关各设置项目详情，请参见第 13-7 页上的“Web 服务器设置”。

3. 将用户程序下载到 Plus CPU 模块。
从 WindLDR 菜单栏选择“联机”>“传送”>“下载”。
出现“下载”对话框。
4. 选中“下载网页”复选框。



13: WEB服务器

5. 从以下选项中选择网页的保存位置。

项目	说明
无下载	不下载网页。
CPU 模块	将下载的网页保存至 Plus CPU 模块的内存。
CPU 模块（高速和非保持型）	将下载的网页保存至 Plus CPU 模块的内存。但在 Plus CPU 模块的电源关闭时下载的网页将被删除。
CPU 模块（SD 记忆卡）	将下载的网页保存至 Plus CPU 模块上插入的 SD 记忆卡。

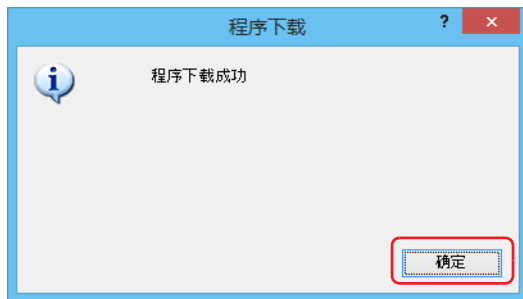
注释：“CPU 模块（高速和非保持）”可比选中“CPU 模块”时更快地下载网页，但是当 Plus CPU 模块的电源关闭时，下载的网页将被删除。若想重复修改和下载网页时选择此项，例如检查创建的网页的操作。

注释：选中“CPU 模块 (SD 记忆卡)”时，请确认已将未锁的 SD 记忆卡插入 Plus CPU 模块后进行下载。

6. 单击“确定”按钮。

网页将下载至已选保存位置。

显示以下信息后表示下载成功。单击“确定”按钮。



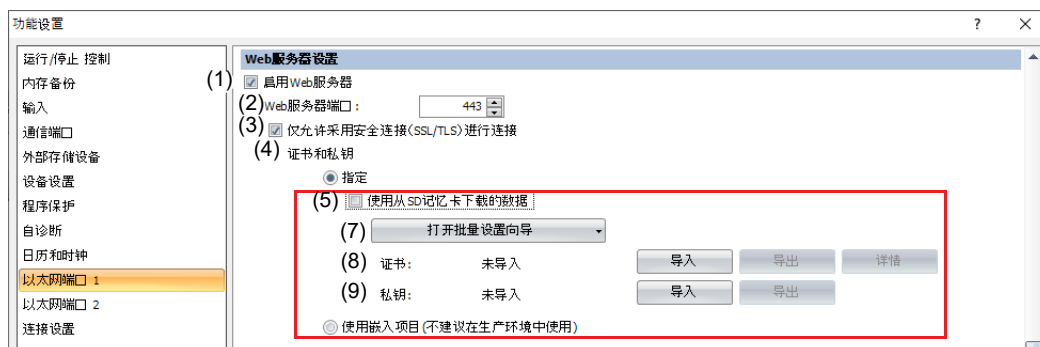
至此，完成 Web 服务器的设置。

Web 服务器设置

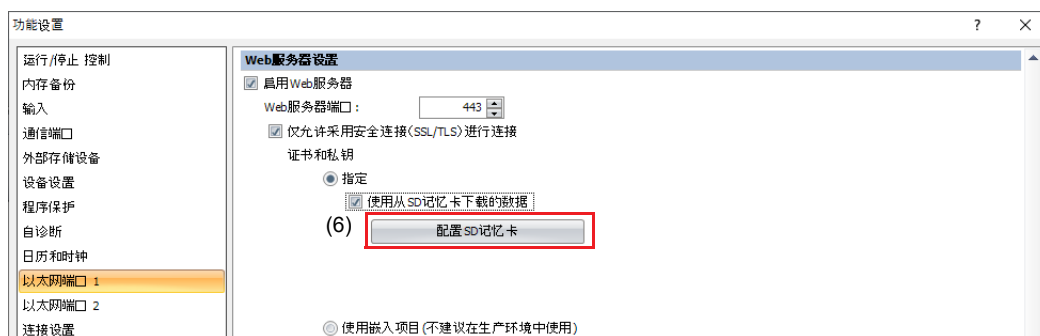
在“Web 服务器设置”中，设置使用的 Web 服务器功能。

在“设置”选项卡的“功能设置”中，单击“以太网端口 1”。

未选中“使用从 SD 记忆卡下载的数据”复选框时



选中“使用从 SD 记忆卡下载的数据”复选框时



(1) 启用 Web 服务器

如果要使用 Web 服务器功能，请选中此复选框。默认值为开启。

(2) Web 服务器端口

打开 Plus CPU 模块 Web 服务器并设置端口号。通常 HTTP 为 80，HTTP over TLS 为 443。端口号设置在 0 到 65535 的范围内。默认值为 80。

(3) 仅允许采用安全连接（SSL/TLS）进行连接

如果要启用 SSL/TLS 通信，请选中此复选框。默认值为关闭。有关启用 SSL/TLS 通信时的限制，请参见第 13-58 页上的“限制事项”。

(4) 证书和私钥

在 SSL/TLS 通信中所使用的证书和私钥可从以下中选择。默认值为“指定”。

设置	内容
指定	可自由指定用于 SSL/TLS 通信的证书和私钥。
使用嵌入项目	使用 Plus CPU 模块内的标准自签名证书和私钥。

注释：如果选择“使用嵌入项目”，在浏览器地址栏中输入“https://...”访问 Plus CPU 模块的 Web 服务器时，浏览器会显示警告。忽略此警告，则可以使用 SSL/TLS 通信访问 Plus CPU 模块的 Web 服务器。

(5) 使用从 SD 记忆卡下载的数据

指定是否使用 SD 记忆卡将证书和私钥写入 Plus CPU 模块。

选中此复选框将显示 (6) 并隐藏 (7) 到 (9)。在这种情况下，在“SD 记忆卡”对话框中设置 (7) 到 (9)。有关“SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(6) 配置 SD 记忆卡

打开 SD 记忆卡对话框。尽在选中“使用从 SD 记忆卡下载的数据”复选框时显示。有关“SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(7) 打开批量设置向导

打开“证书和私钥的批量设置”向导，以向导形式设置证书和私钥。请按照显示的步骤进行设置。

注释：在此向导中，设置证书颁发机构的根证书并将其导入 Windows 证书存储。此外，设置 Plus CPU 模块的 Web 服务器功能使用的证书和私钥，并将它们导入到功能区域设置中。有关详情，请参见「第 13 章 “证书和私钥的批量设置”向导」（13-11 页）。

(8) 证书

单击“导入”按钮以导入用于 SSL/TLS 通信的证书。可导入 pem 或 der 文件。WindLDR 将 pem 文件转换为 der 文件。无法导入大于 2 KB 的 der 文件。只能在选中“仅允许采用安全连接（SSL/TLS）进行连接”复选框，且未选中“使用从 SD 记忆卡下载的数据”复选框时设置。

单击“详情”按钮，显示“证书信息”对话框，其中显示导入的证书信息。

单击“导出”按钮，导出导入的证书（der 文件）。

(9) 私钥

单击“导入”按钮，导入用于 SSL/TLS 通信的私钥（加密方式 RSA，私钥长度可达 2048bit）。WindLDR 支持以下格式的文件：

- PKCS#1 格式 pem 或 der 文件
- PKCS#8 格式（未加密） pem 或 der 文件

只能在选中“仅允许采用安全连接（SSL/TLS）进行连接”复选框，且未选中“使用从 SD 记忆卡下载的数据”复选框时设置。

单击“导出”按钮，导出导入的私钥（der 文件）。



(10) 用户设置

从以下选择要创建的帐户数（1 个或多个）。默认值为“使用单一用户帐户访问网络服务器”。

用户设置	内容
使用单一用户帐户访问网络服务器	选择使用单一用户帐户访问网络服务器时。
为多用户配置网络服务器上的访问权限	选择为多用户配置网络服务器上的访问权限时。

注释：在“用户帐户设置”对话框中设置多个用户帐户。有关详情，请参见第 13-23 页上的“用户帐户设置”。

(11) 用户名

(12) 密码

设置用户名和密码。用户名的最大字符数为 40，密码的最大字符数为 16。只能使用英文字母、数字和符号。仅当“使用单一用户帐户访问网络服务器”单选按钮为开启时，才能设置此项。

项目	出厂设置
用户名	administrator
密码	password

(13) “设置”按钮

打开“用户帐户设置”对话框。在“用户帐户设置”对话框中创建用户帐户。有关详情，请参见第 13-23 页上的“用户帐户设置”。

(14) 启用数据寄存器监控

在以下情况下选中此复选框：

- 在 HTML 页面中使用带有嵌入元字符的数据寄存器监视器时
- 使用图形库时

只有带有 html 和 htm 扩展名的文件才有效。

(15) 监控间隔时间 (ms)

进行监控时，通过 Web 浏览器以 ms 为单位，将发送请求的周期设置到 FC6A 型。

(16) 趋势监控间隔时间 (ms)

使用趋势图进行监控时，应以监控周期的整数倍设置从 Web 浏览器向 FC6A 型发送请求的周期 (ms)。

(17) 运行 CGI 程序前请求用户名和密码

如果在运行 CGI 程序之前需要用户名和密码，请选中此复选框。默认值为开启。有关 CGI 程序的详情，请参见第 13-56 页上的“CGI”。

(18) 重定向目标

可以重定向到“重定向目标”中设置的网页。

设置	说明	重定向目标设置
禁用重定向	不进行重定向。	不需要
系统网页	重定向到系统网页的首页。	
用户网页和自定义网页 ^{*1}	重定向到设置的用户网页或自定义网页。	需要

*1 选择导入或保存的文件夹名称和文件名。

重定向目标	说明
CPU 模块	重定向到保存在 CPU 模块内存中的重定向目标。
SD 记忆卡	重定向到保存在 SD 记忆卡上的重定向目标。

例如，Plus CPU 模块的以太网口 1 的 IP 地址为 192.168.1.20，且将重定向目标设置为 SD 记忆卡并将网页下载到 SD 记忆卡，可以在以下网址显示重定向目标。

- <http://192.168.1.20/sd/>

注释： 请将网页的重定向目的地和下载目的地设置为同一位置。

(19) 使用系统网页

使用系统网页时，应选中该复选框。

(20) 语言

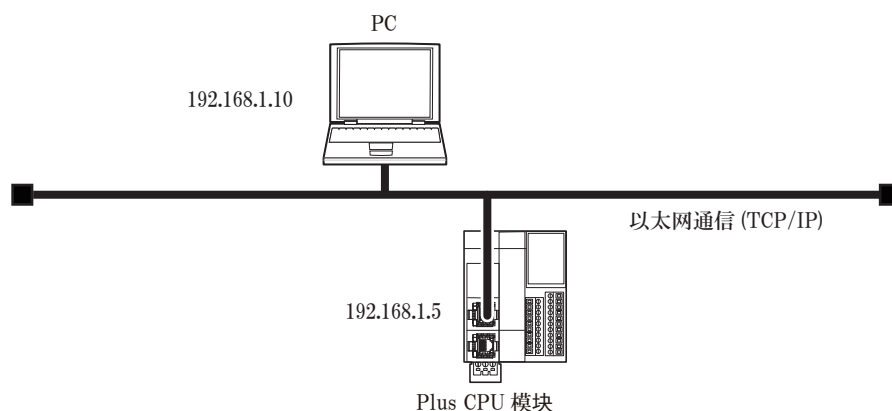
系统网页的语言可从日语、英语、中文、德语或西班牙语中选择。

(21) 使用网页编辑器

使用网页编辑器时，应选中该复选框。有关详情，请参见第 13-44 页上的“网页编辑器”。

“证书和私钥的批量设置”向导

在 Plus CPU 模块上，可以使用“证书和私钥批量设置”向导来设置用于 SSL/TLS 通信的证书和私钥。本节介绍在下图所示的系统设置中使用 Web 服务器功能时，设置证书和私钥的步骤。



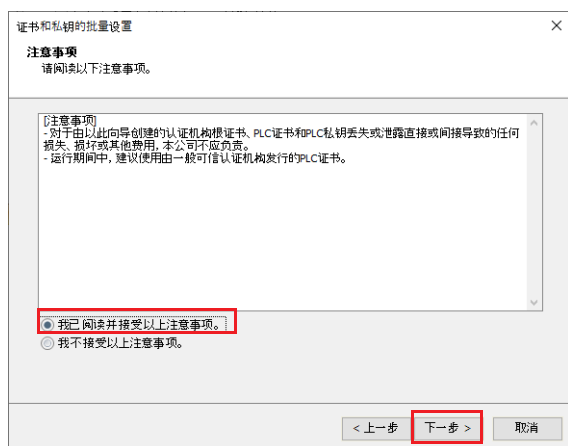
●操作过程

1. 单击“打开批量设置向导”。
将显示“证书和私钥的批量设置”向导。
2. 单击“下一步”。



注释：单击“证书存储区（逻辑系统存储区：“可信根认证机构”）”链接将打开当前登录到 Windows 的用户的证书存储区（certmgr.msc）。

3. 选中“我已阅读并接受以上注意事项。”并单击“下一步”。



4. 设置所需项目，并单击“下一步”。如果没有要设置的项目，请单击“下一步”，不必设置每个项目。

各设置项目的说明如下所示。

(1) 使用以下设置创建认证机构的根证书

创建新的证书颁发机构根证书时打开。默认值为打开。

(2) 通用名称

设置证书颁发机构根证书的通用名称。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

仅在“使用以下设置创建证书颁发机构的根证书”单选按钮打开时可以设置。默认值为 MyRootCA。

(3) 国家 / 地区

设置证书颁发机构的所在地（国家 / 地区）。最大字符数为 2。只能使用半角字母字符（大写字母）。在“使用以下设置创建证书颁发机构的根证书”单选按钮与“国家 / 地区”复选框都为打开时可以设置。

(4) 省份 / 城市

设置证书颁发机构的所在地（省份 / 城市）。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“使用以下设置创建证书颁发机构的根证书”单选按钮与“省份 / 城市”复选框都为打开时可以设置。

(5) 区县

设置证书颁发机构的所在地（区县）。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“使用以下设置创建证书颁发机构的根证书”单选按钮与“区县”复选框都为打开时可以设置。

(6) 组织名称

设置证书颁发机构的组织名称。最大字符数为 64。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“使用以下设置创建证书颁发机构的根证书”单选按钮与“组织名称”复选框都为打开时可以设置。

(7) 组织单位

设置证书颁发机构的组织单位。最大字符数为 64。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“使用以下设置创建证书颁发机构的根证书”单选按钮与“组织单位”复选框都为打开时可以设置。

(8) 生效日期

设置证书颁发机构根证书的有效时间（开始生效日）。默认值为 PC 日期。仅在“使用以下设置创建证书颁发机构的根证书”单选按钮打开时可以设置。

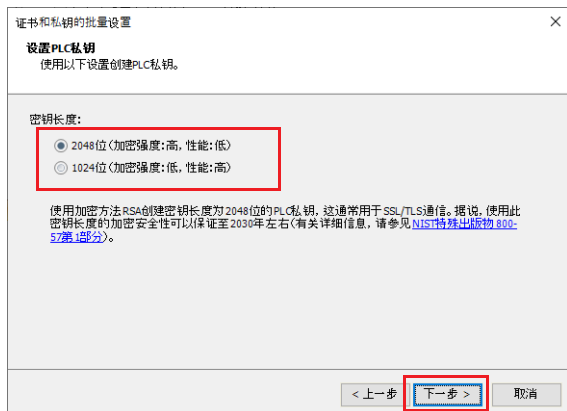
(9) 终止日期

设置证书颁发机构根证书的有效时间（生效截止日）。默认值为 PC 日期的 365 天后。仅在“使用以下设置创建证书颁发机构的根证书”单选按钮打开时可以设置。

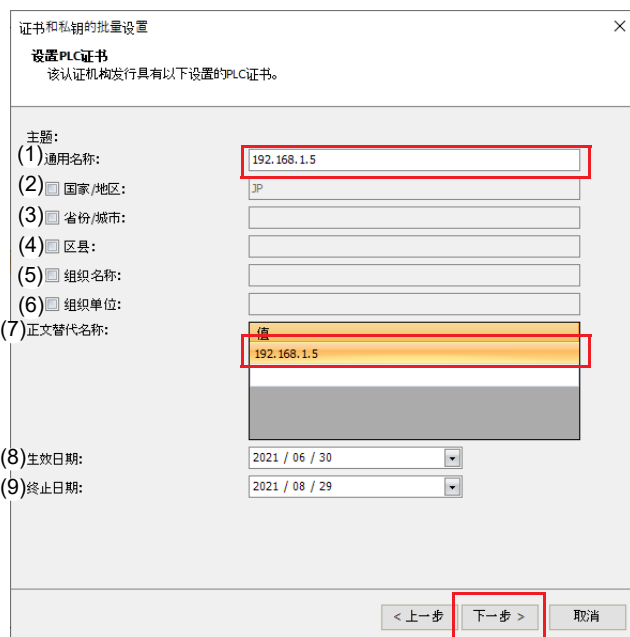
(10) 重复使用之前创建的认证机构的根证书

重复使用之前创建的认证机构的根证书时打开。仅在存在之前创建的认证机构的根证书时可设置。单击“详情”按钮，显示“证书信息”对话框，并且显示之前创建的认证机构的根证书信息。

5. 选中“2048位”或“1024位”并单击“下一步”。



6. 在“通用名称”和“正文替代名称”中设置 Plus CPU 模块的以太网端口 1 的 IP 地址，并单击“下一步”。如果需要设置其他项目，请设置需要项目，并单击“下一步”。



各设置项目的说明如下所示。

(1) 通用名称

设置 PLC 的域名或 IP 地址。域名的最大字符数为 128。只能使用半角字母、数字、连字符和句号。IP 地址的输入格式为“xxx.xxx.xxx.xxx”。“xxx”可输入 0 到 255 之间的值。

注释：可以通过指定“*.”为开头的域名来创建通配符证书。

(2) 国家 / 地区

设置证书颁发机构的所在地（国家 / 地区）。最大字符数为 2。只能使用半角字母字符（大写）。在“国家 / 地区”复选框为打开时可以设置。

(3) 省份 / 城市

设置证书颁发机构认证目标的所在地（省份 / 城市）。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“省份 / 城市”复选框为打开时可以设置。

(4) 区县

设置证书颁发机构认证目标的所在地（区县）。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“区县”复选框为开启时可以设置。

(5) 组织名称

设置证书颁发机构认证目标的组织名称。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“组织名称”复选框为开启时可以设置。

(6) 组织单位

设置证书颁发机构认证目标的组织单位。最大字符数为 128。只能使用半角字母、数字以及以下特殊字符和空格。

'()-. / : ?

在“组织单位”复选框为开启时可以设置。

(7) 正文替代名称

设置 PLC 的域名或 IP 地址。域名的最大字符数为 128。只能使用半角字母、数字、连字符和句号。IP 地址的输入格式为“xxx.xxx.xxx.xxx”。“xxx”可输入 0 到 255 之间的值。可设置的域名或 IP 地址的最大个数为 10 个。

注释：可以通过指定“*”为开头的域名来创建通配符证书。

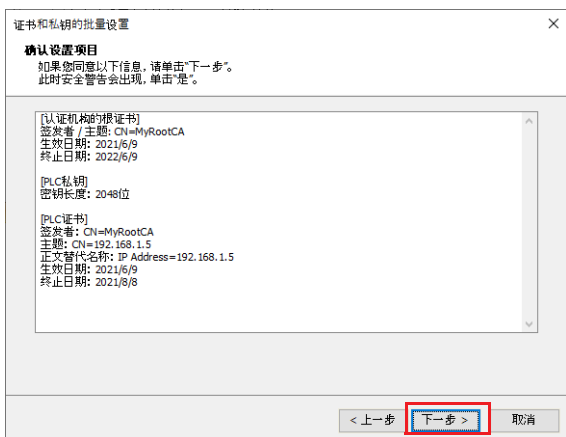
(8) 生效日期

设置 PLC 证书的有效时间（开始生效日）。默认值为 PC 日期。不可设置为认证机构的根证书的有效时间（开始生效日）之前的日期。

(9) 终止日期

设置 PLC 证书的有效时间（生效截止日）。默认值为 PC 日期的 60 天后。不可设置为认证机构的根证书的有效时间（生效截止日）之后的日期。

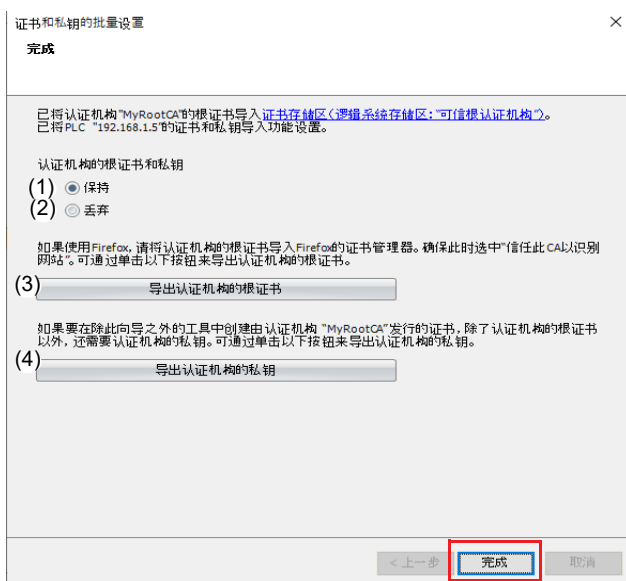
7. 确认设置内容，并单击“下一步”。



8. 单击“确定”。



9. 单击“确认”按钮。



各设置项目及按钮的说明如下所示。

(1) 保持

如果将认证机构的根证书和私钥保存在项目中则打开。默认值为打开。

(2) 丢弃

如果要认证机构的根证书和私钥从项目中删除则打开。

(3) “导出认证机构的根证书”按钮

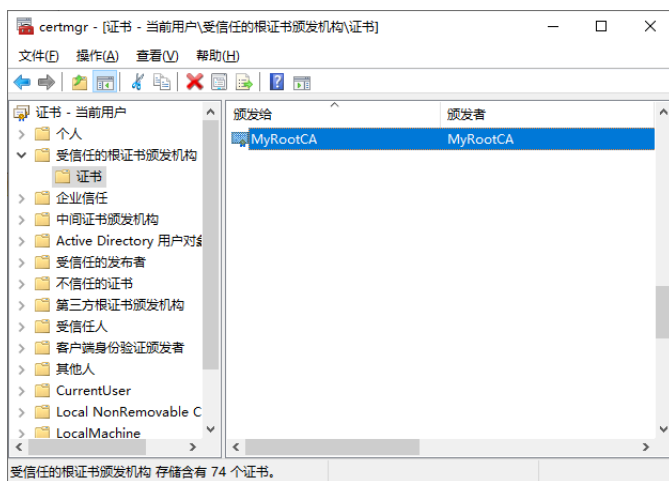
导出认证机构的根证书 (der 文件)。

(4) “导出认证机构的私钥”按钮

导出认证机构的私钥 (der 文件)。

注释:

- 单击“证书存储区 (逻辑系统存储区: “可信根认证机构)”链接, 则会打开当前 Windows 所登录的用户的证书存储区 (certmgr.msc)。如果在左侧窗口中展开“可信根认证机构”并单击“证书”, 可以看到已添加 MyRootCA (认证机构的根证书)。

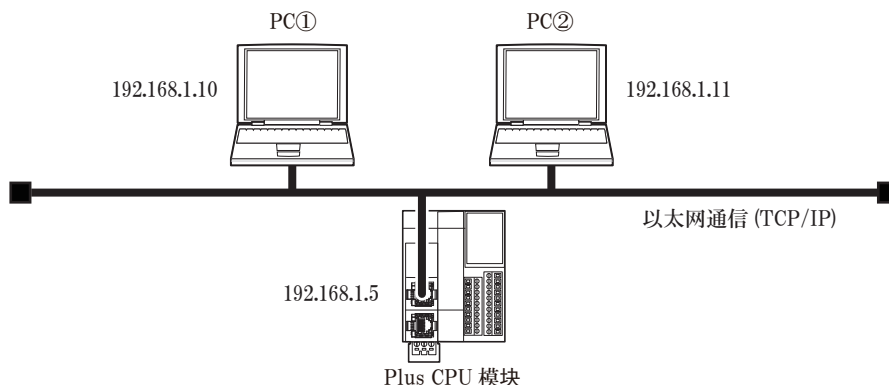


- 未将认证机构的根证书导入证书存储区的 PC 通过网络浏览器访问 Web 服务器时, 网络浏览器上会显示警告。单击“导出认证机构的根证书”按钮, 导出认证机构的根证书, 并导入 PC 中, 可消除此警告。有关详情, 请参见第 13-16 页上的“导入认证机构的根证书”。

至此, 完成证书和私钥的设置。

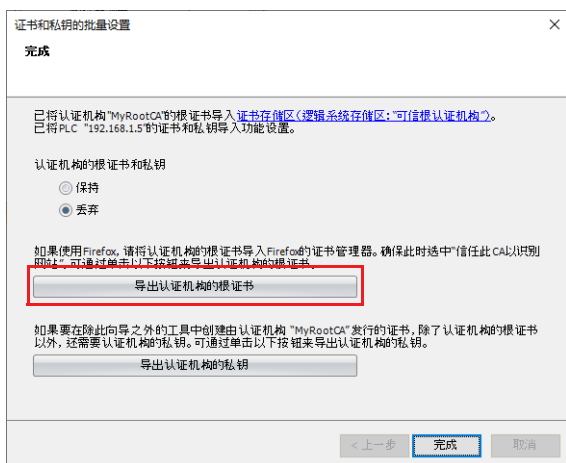
导入认证机构的根证书

未将认证机构的根证书导入证书存储区的 PC 通过网络浏览器访问 Web 服务器时，网络浏览器上会显示警告。将认证机构的根证书导入 PC 中，可消除此警告。设置下图中的 PC ①上认证机构的根证书，并导入 PC ②的证书存储区，过程如下。

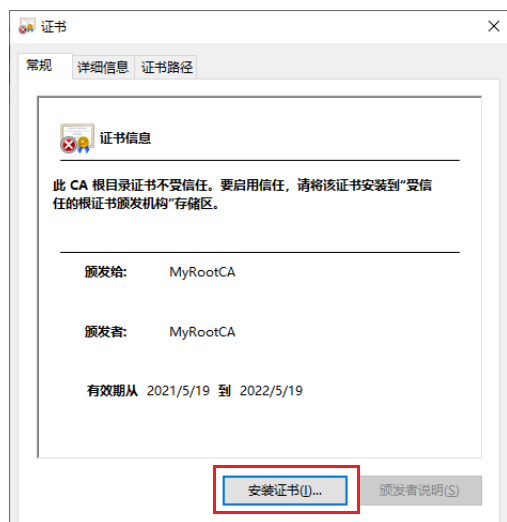


●操作过程

1. 对 PC ①进行「第 13 章 “证书和私钥的批量设置” 向导」(13-11 页)的步骤 1 到 8。
2. 单击“导出认证机构的根证书”按钮，将认证机构的根证书保存在 PC ①中。



3. 步骤 1 所保存的认证机构的根证书拷贝到 PC ②中。
4. 在 PC ②的文件资源管理器上双击认证机构的根证书并打开属性，单击“安装证书”。



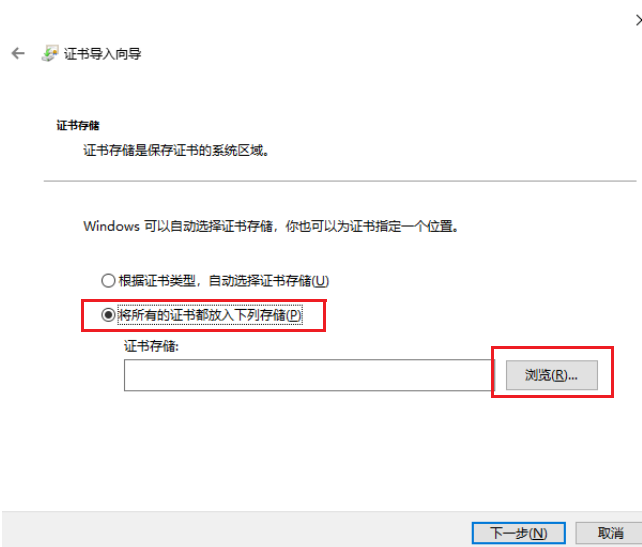
打开证书的导入向导。

5. 单击“下一步”。

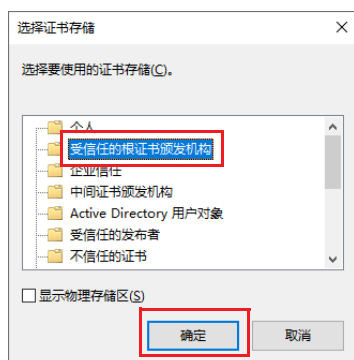


注释：如果需要认证机构的根证书对 PC ② 的所有用户可见，请打开本地计算机。将认证机构的根证书导入本地计算机的证书存储区 (certlm.msc)。（需要管理员权限）

6. 选中“在以下所有的存储区中配置证书”，并单击“浏览”。

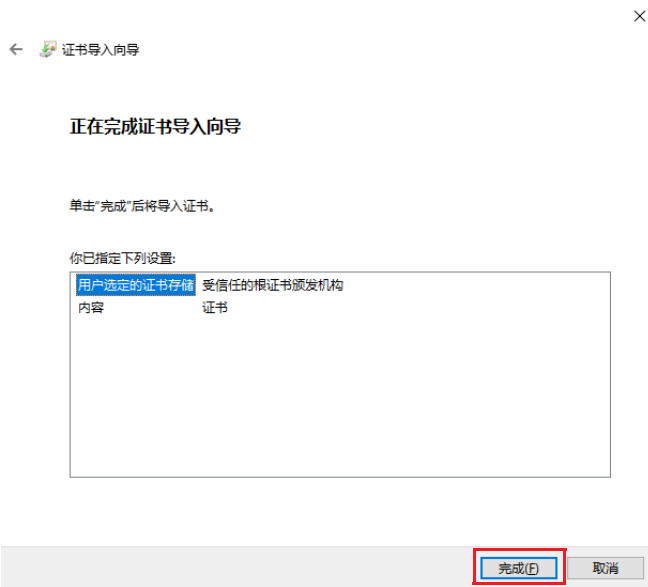


7. 选中“可信根认证机构”，并单击“确定”。

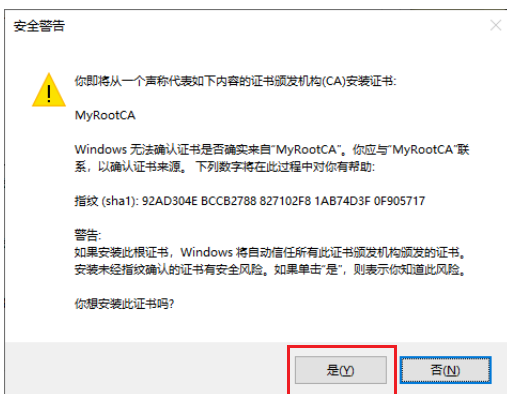


13: WEB服务器

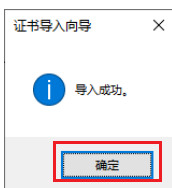
8. 并单击“完成”。



9. 单击“是”。



10. 单击“确定”。



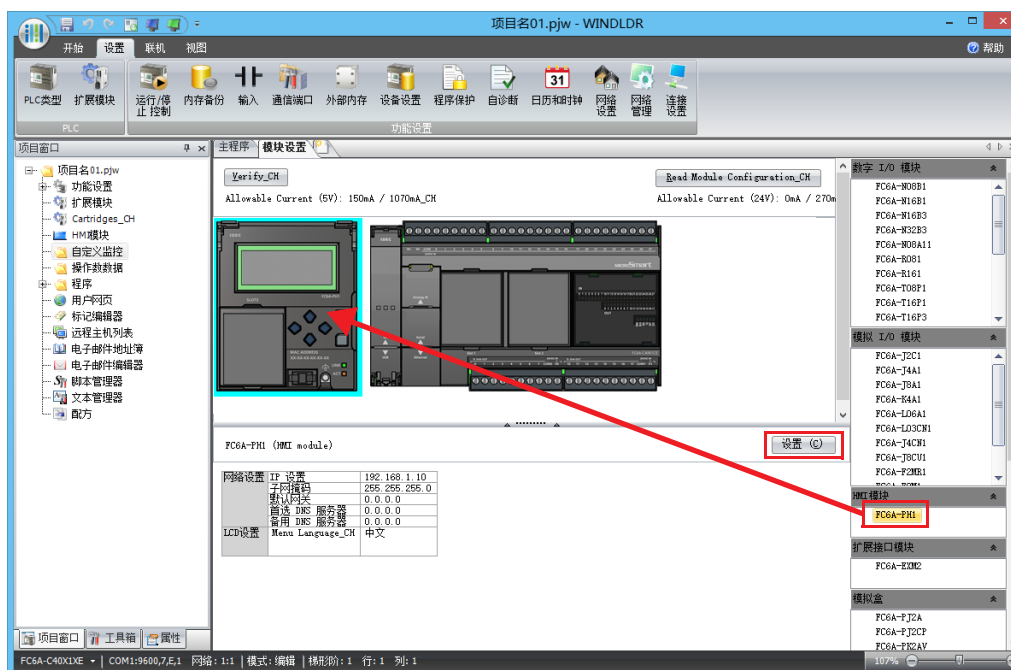
至此，完成将认证机构的根证书导入 PC ②。

HMI 模块的 Web 服务器设置

本节介绍在 HMI 模块的 HMI-Ethernet 端口上使用 Web 服务器功能时的 Web 服务器设置。

●操作过程

1. 可通过模块构成编辑器进行 HMI 模块的 Web 服务器设置。
在“设置”选项卡的“PLC”中单击“扩展模块”。
2. 单击模块构成区域中插入的 HMI 模块，然后单击“设置”按钮。
将显示“HMI 模块的设置”对话框。

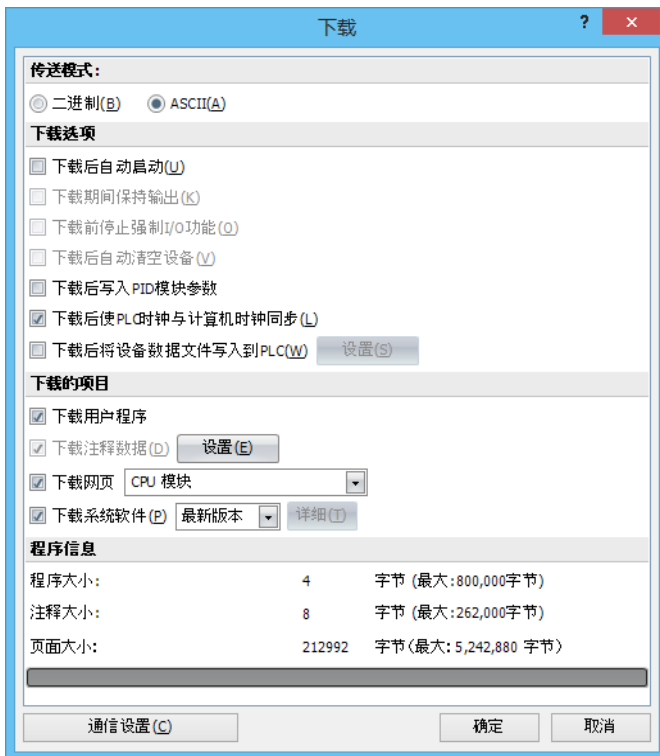


注释：还可以通过以下方式显示 HMI 模块设置对话框：

- 在项目窗口中双击 HMI 模块
 - 双击模块设置区域中的 HMI 模块
3. 单击“Web 服务器”选项卡，设置各项目。有关各设置项目的详情，请参见 Web 服务器设置 13-7 页。
 4. 将用户程序下载到 CPU 模块。
在“联机”选项卡上的“传送”单击“下载”按钮。
显示下载对话框。

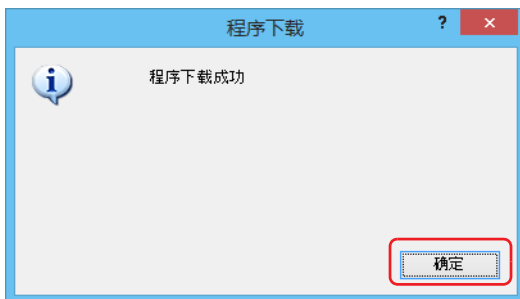
13: WEB服务器

- 选中“下载网页”复选框。



注释：网页保存在 HMI 模块的内存中。

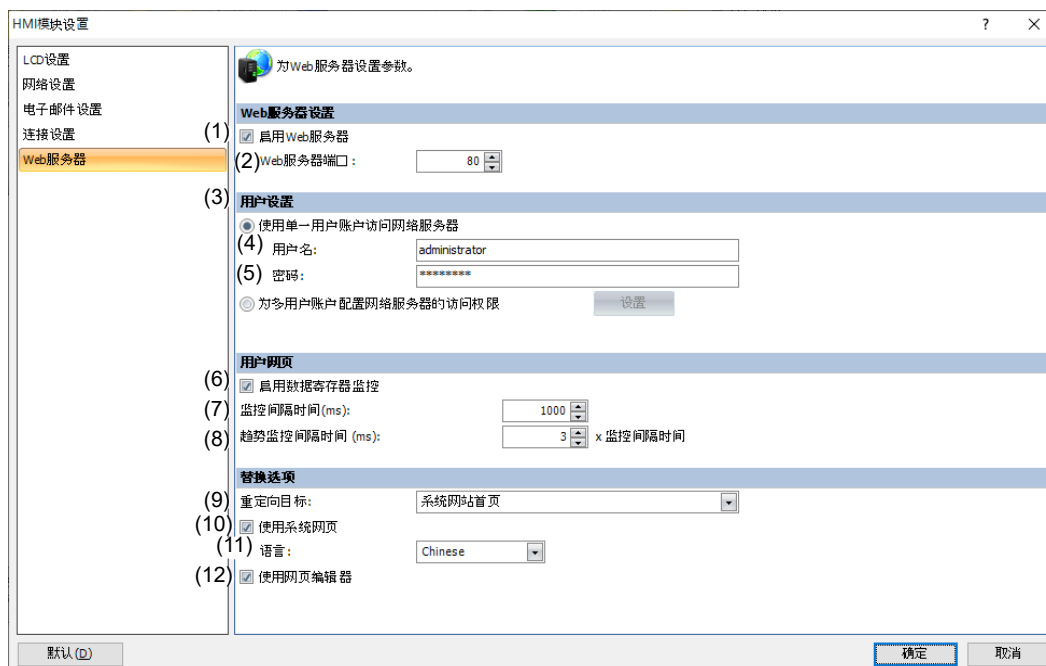
- 单击“确定”按钮。
创建的网页将下载至 HMI 模块的内存中。
显示以下信息后表示下载成功。单击“确定”按钮。



至此，完成 Web 服务器的设置。

Web 服务器设置

在“HMI 模块设置”对话框的“Web 服务器”选项卡中，设置所使用的 Web 服务器功能。



(1) 启用 Web 服务器

使用 Web 服务器功能时，应选中该复选框。默认值为打开。

(2) Web 服务器端口

打开 HMI 模块的 Web 服务器后设置端口号。端口号可在 0 到 65535 的范围内设置。默认值为 80。

(3) 用户设置

设置创建的账户数（1 个或多个）。可在以下中设置。默认值为“使用单一用户账户访问网络服务器”。

用户设置	内容
使用单一用户账户访问网络服务器	使用单一用户账户访问网络服务器时选择。
为多用户账户配置网络服务器的访问权限	使用多用户账户配置网络服务器的访问权限时选择。

注释：多用户账户在“用户账户设置”对话框中设置。有关详情，请参见“用户账户”。

(4) 用户名

(5) 密码

设置用户名和密码。用户名的最大字符数为 40，密码的最大字符数为 16。只能使用字母、数字和符号。仅在“使用单一用户账户访问网络服务器”单选按钮打开时可设置。

项目	出厂设置
用户名	administrator
密码	password

(6) 启用数据寄存器监控

在以下情况下应选中该复选框：

- 在 HTML 页面中使用带有嵌入元字符的数据寄存器监视器时
- 使用图形库时

只有带有 html 和 htm 扩展名的文件才有效。

(7) 监控间隔时间 (ms)

进行监控时，通过 Web 浏览器以 ms 为单位，将发送请求的周期设置到 FC6A 型。

(8) 趋势监控间隔时间 (ms)

使用趋势图进行监控时，应以监控周期的整数倍设置从 Web 浏览器向 FC6A 型发送请求的周期 (ms)。

13: WEB服务器

(9) 重定向目标

可以重定向到“重定向目标”中设置的网页。

设置	说明
禁用重定向	不进行重定向。
系统网页	重定向到系统网站首页。
用户网页和自定义网页 ^{*1}	重定向到设置的用户网页或自定义网页。

*1 选择导入的或保存到文件夹名以及文件名。

例如，如果 HMI 模块的 HMI-Ethernet 端口的 IP 地址为 192.168.1.10，则可以在以下网址查看重定向目标：

- <http://192.168.1.10/>

(10) 使用系统网页

使用系统网页时，应选中该复选框。

(11) 语言

系统网页多语言可从日语、英语、中文、德语或西班牙语中设置。

(12) 使用网页编辑器

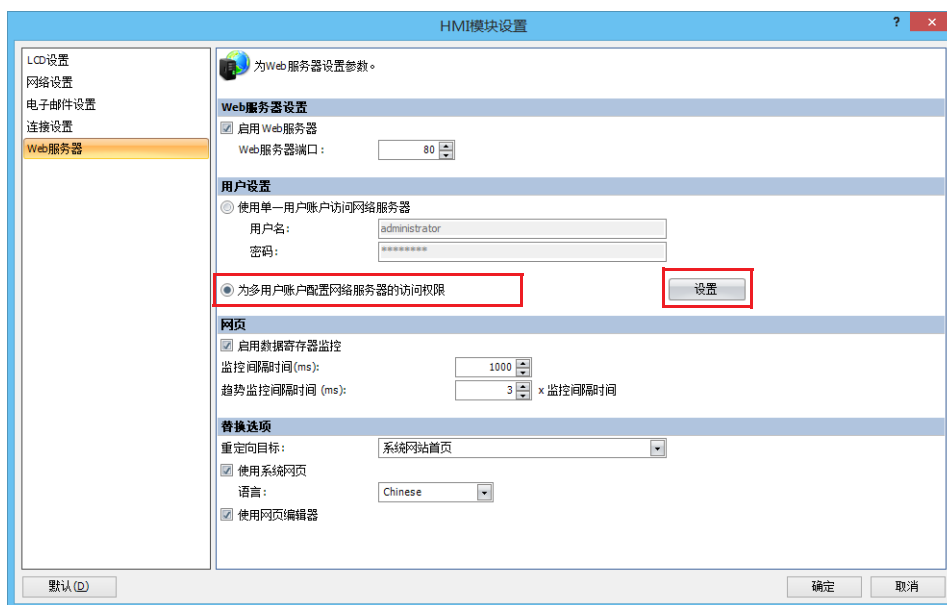
用网页编辑器时，应选中该复选框。有关详情，请参见第 13-44 页上的“网页编辑器”。

用户帐户设置

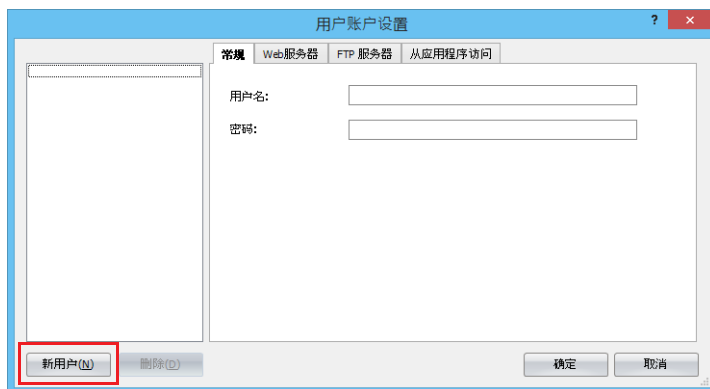
■ 为多用户账户配置 Web 服务器的访问权限

● 操作过程

1. 在“功能设置”对话框“以太网端口 1”的“Web 服务器设置”，或“HMI 模块设置”对话框的“Web 服务器”选项卡中，选择“将多用户的访问限制设置为 Web 服务器”，然后单击“设置”按钮。将显示“用户帐户设置”对话框。

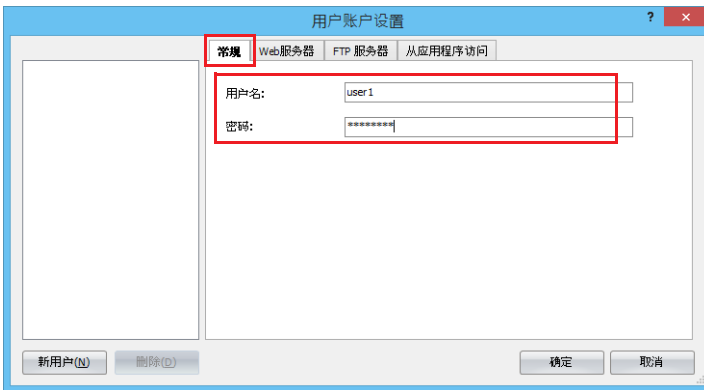


2. 单击“新用户”按钮。
将在列表中创建用户帐户。

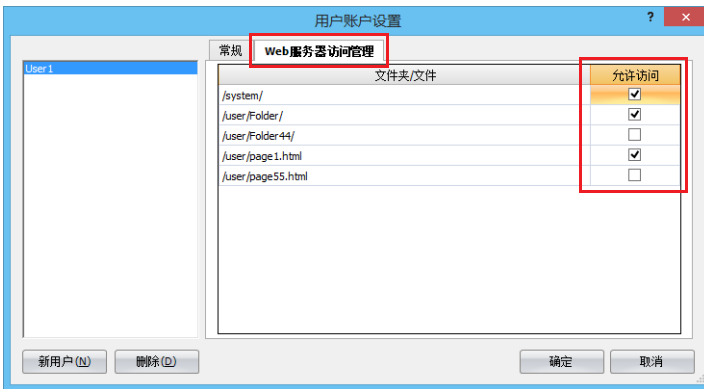


13: WEB服务器

- 在“常规”选项卡中分别设置“用户名”和“密码”。



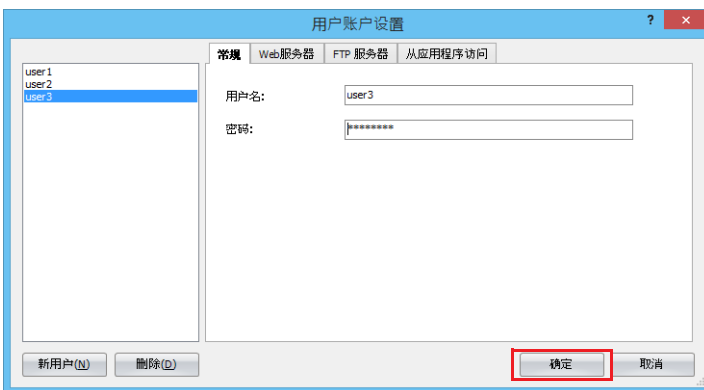
- 单击“Web 服务器访问管理”选项卡，然后选中允许访问的文件夹及服务器的复选框。



- 重复步骤 2 ~ 4 的操作，创建所需数量的用户帐户。

注释：在列表中选择用户帐户后，可编辑所选择用户帐户的“常规”选项卡、“Web 服务器访问管理”选项卡的内容。

- 单击“确定”按钮。

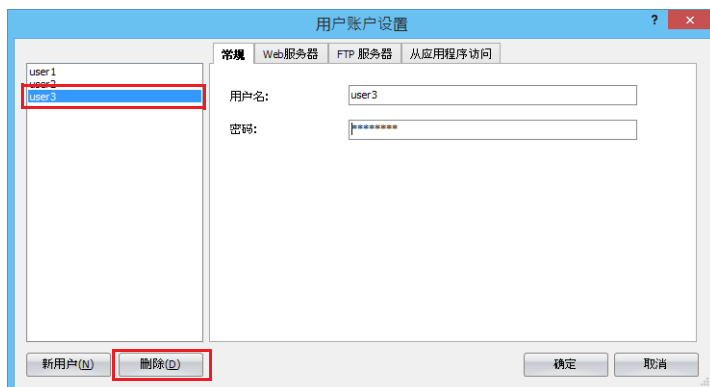


至此，完成多个用户帐户的创建。

■ 删除已创建的用户帐户

● 操作过程

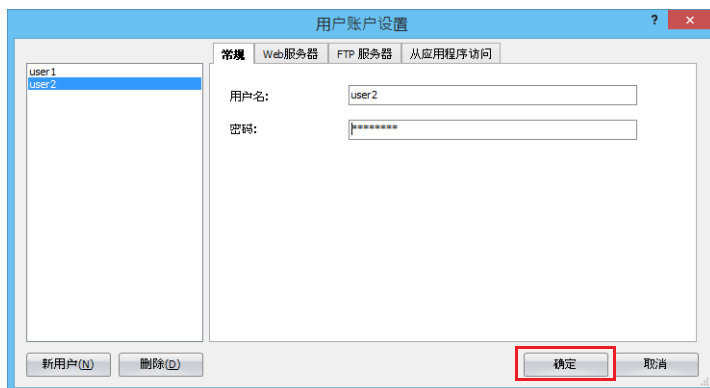
1. 在列表中选择用户帐户，然后单击“删除”按钮。
将显示确认消息。



2. 单击“OK”按钮。



3. 单击“确定”按钮。



至此，完成用户帐户的删除。

系统网页

系统网页由四个网页组成：PLC 状态页面、多点监控页面、注册监控页面和 LCD 监控页面。

■ PLC 状态

在 PLC 状态页面，可以确认 FC6A 型的状态，如系统程序版本或扫描时间等。您还可以使用运行 / 停止按钮运行或停止 FC6A 型。

MicroSmart系统状态

(1) 系统信息

项目	值
PLC类型	FC6A-C40
系统程序版本	1.00

(2) 运行状态

项目	值
运行/停止状态	<input type="button" value="停止"/>
扫描时间(当前值)	2ms
扫描时间(最大值)	11ms
电池电压	3182mV
通用错误代码	0000

(3) 日期/时钟

项目	值
时钟(年/月/日/星期/时/分/秒)	2015/12/02(周三) 16:40:36

(4) 网络设置

项目	值
MAC地址	00-03-7b-f0-16-5f
IP地址	192.168.1.5
子网掩码	255.255.255.0
默认网关	0.0.0.0
首选DNS服务器	0.0.0.0
备用DNS服务器	0.0.0.0

(5) 连接设置

项目	值
连接1 - 维护通信服务器	0.0.0.0
连接2 - 维护通信服务器	0.0.0.0
连接3 - 维护通信服务器	0.0.0.0
连接4 - 维护通信服务器	0.0.0.0
连接5 - 维护通信服务器	0.0.0.0
连接6 - 维护通信服务器	0.0.0.0
连接7 - 维护通信服务器	0.0.0.0
连接8 - 维护通信服务器	0.0.0.0

For details about web server, see FC6A User's Manual - Communication Volume_CH

可在 PLC 状态页确认以下数据：

(1) 系统信息

可以确认 FC6A 型的型号和系统程序版本。

(2) 运行状态

可以确认运行 / 停止、扫描时间及错误代码。点击运行 / 停止状态中的按钮，可以启动或停止 FC6A 型。

(3) 日期 / 时钟

可确认 Sntp 所获得的日期和时钟数据。

(4) 网络设置

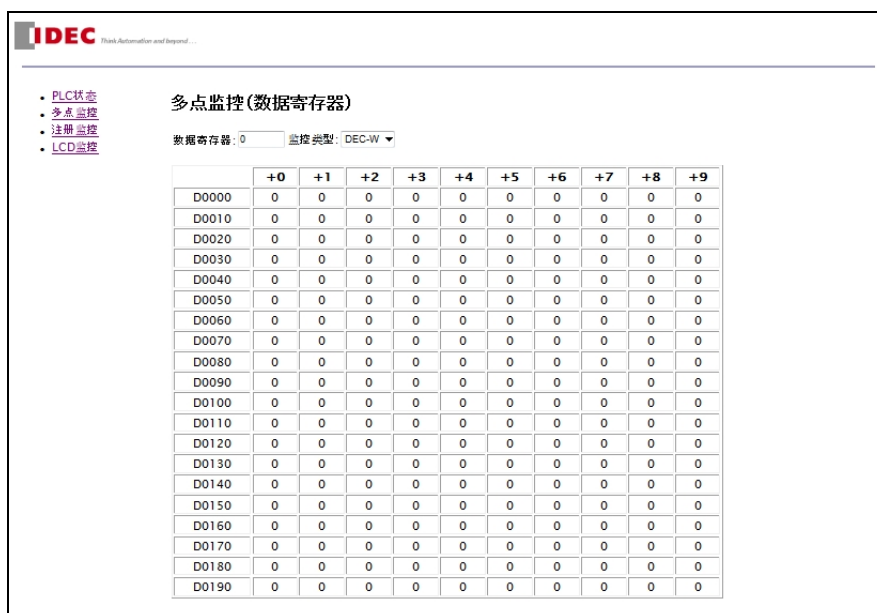
可以确认 FC6A 型的网络设置。

(5) 连接设置

FC6A 型与之通信的设备的 IP 地址。显示访问服务器端口的客户端设备的 IP 地址，用于“维护通信服务器”和“服务器连接”。显示 FC6A 型与之通信的远程主机的 IP 地址，用于“客户端连接”。Plus CPU 模块时，将显示连接 1 ~ 16 的通信目标的 IP 地址。

■ 多点监控

在多点监控中，可以使用指定的数据类型监视和控制 200 个连续的数据寄存器。有关数据类型详情，请参见第 13-29 页上的“数据类型”。



• 操作过程

1. 在“设备编号”中指定要监控的设备编号。
监控以指定的数据寄存器开始的 200 个连续的数据寄存器。可设置的设备编号如下表所示。设置超出范围的值时，或连续 200 字的最后数据超过范围时，可自动转换为范围内的值进行输入。

CPU 模块类型	设备编号的范围
All-in-One CPU 模块	0000 ~ 7800
	8000 ~ 8300
	10000 ~ 55800
Plus CPU 模块	0000 ~ 7800
	8000 ~ 8700
	10000 ~ 61800
	70000 ~ 269800

2. 选择“监控类型”，以所需的格式显示数据寄存器。
以监控类型指定的格式显示数据寄存器值。有关可用的数据类型，请参见 Web 数据类型。
3. 单击数据寄存器值所在的单元格。
显示“写入数据”对话框。输入值被写入到 FC6A 型中。

■ 注册监控

在注册监控中，最多可指定 30 个设备和相应的数据类型进行监视和控制。有关数据类型详情，请参见第 13-29 页上的“数据类型”。

注册监控

输入数据寄存器/内部继电器，选择监控类型进行监控。

编号	设备地址	类型	值
1	D0000	DEC-W	324
2	D0034	DEC-I	67
3		DEC-W	
4		DEC-W	
5		DEC-W	
6		DEC-W	
7		DEC-W	
8		DEC-W	
9		DEC-W	
10		DEC-W	
11		DEC-W	
12		DEC-W	
13		DEC-W	
14		DEC-W	
15		DEC-W	
16		DEC-W	
17		DEC-W	
18		DEC-W	
19		DEC-W	
20		DEC-W	
21		DEC-W	
22		DEC-W	
23		DEC-W	
24		DEC-W	
25		DEC-W	
26		DEC-W	
27		DEC-W	
28		DEC-W	
29		DEC-W	
30		DEC-W	

• 操作过程

1. 在“设备地址”中输入设备（类型和编号）。
可以指定 "D"（数据寄存器）或 "m"（内部继电器）。以十进制指定数据设备编号，如 D2058 或 m0112。
2. 选择“类型”，以所需的格式显示相应的设备。
以监控类型指定的格式显示数据寄存器值或内部继电器值。有关可用的数据类型，请参见 Web 数据类型。
3. 单击“值”中所显示的监控中的数据。
显示“写入数据”对话框。输入值被写入到 FC6A 型中。

■ LCD 监控页面

在 LCD 监控页面中，可监控 HMI 模块的画面。此外，还可通过单击所显示的键或快捷键来操作 HMI 模块。仅在使用 HMI 模块的 HMI- 以太网端口时可显示。



• 操作过程

监控步骤如下所示。

1. 将显示 HMI 模块的 LCD 中所显示的菜单或消息。
2. 单击键，在 HMI 模块中进行与按下 HMI 模块的键时相同的动作。

■ 数据类型

可以在系统网页的多点监控页面或注册监控页面设置显示值的类型。以及输入数据寄存器的值和内部继电器的状态。

可以设置的数据类型如下。

数据类型	说明
DEC-W	字（十进制）
DEC-I	整数（十进制）
DEC-D	双字（十进制）
DEC-L	长整（十进制）
DEC-F	浮点（十进制）
HEX-W	字（十六进制）
HEX-D	双字（十六进制）
BIN-B	二进制

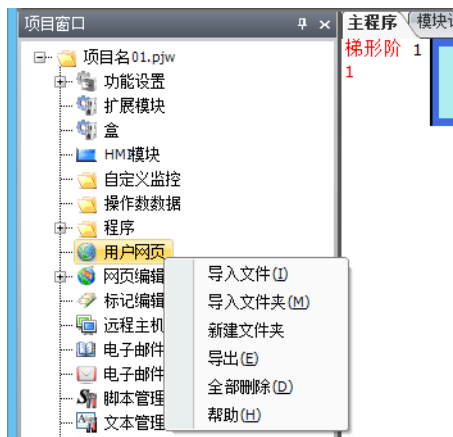
有关字、整数、双字、长整以及浮点的详情，请参见《梯形图编程手册》第3章中的“数据类型”。

用户网页

通过导入以 HTML 和 JavaScript 创建的网页，可以创建自由度更高的用户网页。

用户网页树状图的操作

可针对项目窗口中的“用户网页”项目及作为“网页”注册的文件或文件夹，进行以下操作，可构建用户网页树状图。



网页树状图的创建

在项目窗口中创建网页树状图。

以下步骤将以创建下述网页树状图结构的方法为例进行介绍。

网页树状图结构

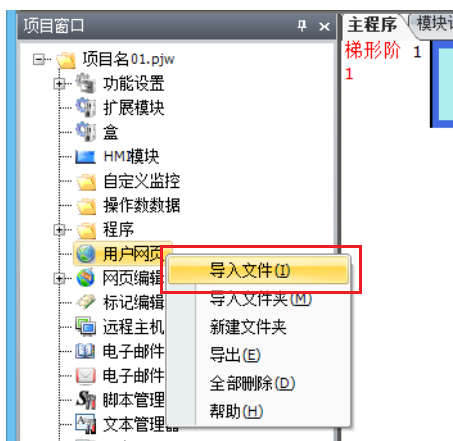
page1.html

Folder/page2.html

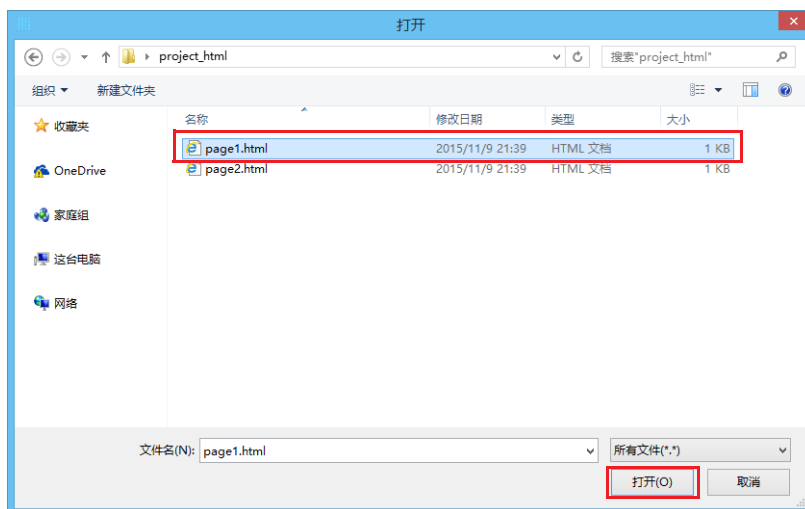
事先创建 FC6A 型上所显示的 page1.html 及 page2.html。

●操作过程

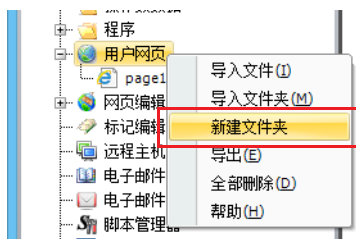
1. 右击“用户网页”，然后单击“导入文件”。
出现“打开”对话框。



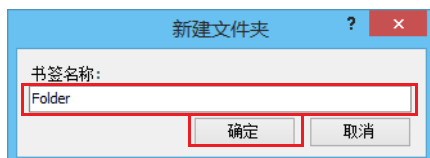
- 选择“page1.html”，然后单击“打开”按钮。



- 右击“用户网页”，然后单击“新建文件夹”。
将显示“新建文件夹”对话框。

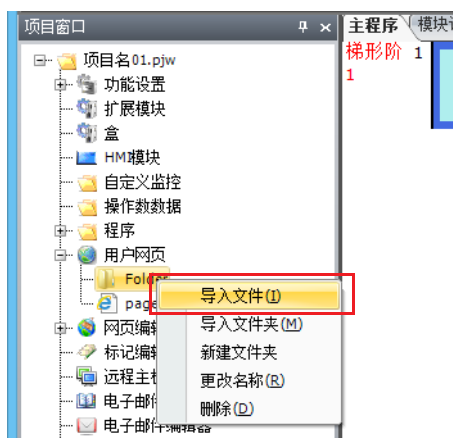


- 输入“Folder”作为新建文件夹名称，然后单击“确定”按钮。
可在“用户网页”中创建“Folder”文件夹。

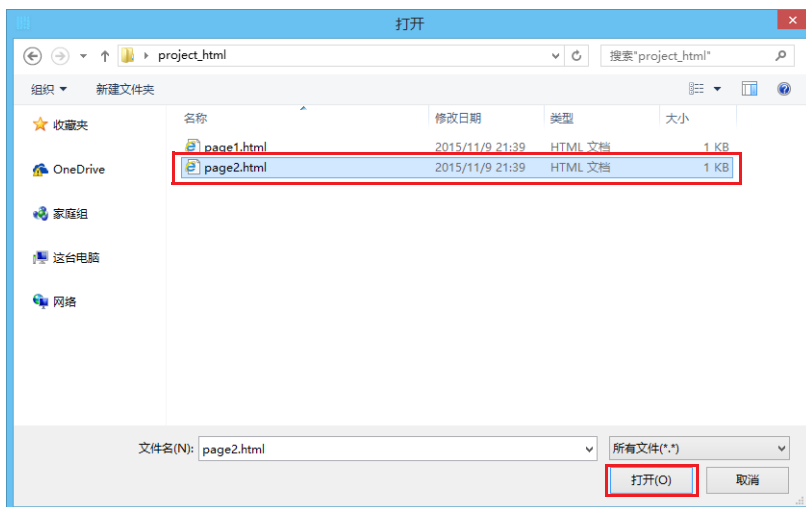


如果输入了错误的文件夹名称，应右击相应文件夹，然后单击“重命名文件夹”，可更改文件夹名称。

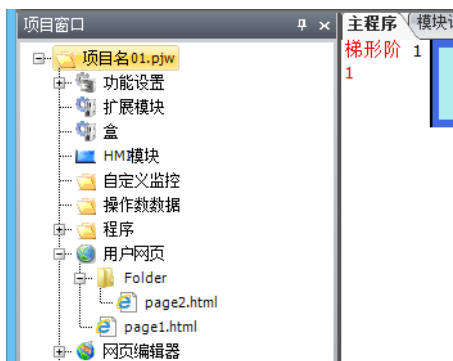
- 右击步骤 4 中所创建的“Folder”文件夹，然后单击“导入文件”。
出现“打开”对话框。



6. 选择“page2.html”，然后单击“打开”按钮。



项目窗口中将显示如下内容。



网页树状图的导入

如果存在事先考虑到树状图结构的 HTML 文件树状图，则可整体导入该树状图，并添加到用户网页中。右击项目窗口中的“用户网页”或“用户网页”以下的文件夹，然后单击“导入文件夹”，将显示对话框。选择存在导入网页树状图的文件夹后，所设置文件夹以下的所有文件都将被导入。

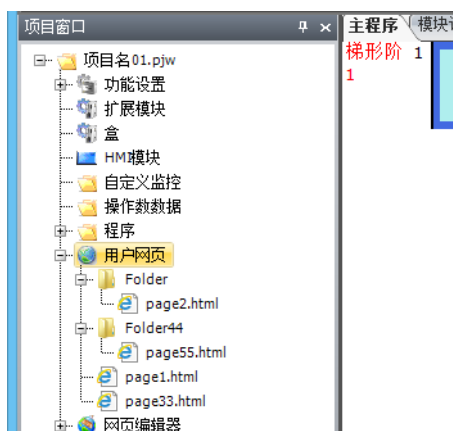
网页树状图的导出

通过导出用户网页数据，可重新构建网页树状图和变更文件等。右击项目窗口中的“用户网页”，然后单击“导出文件夹”，将显示对话框。设置导出目标文件夹后，用户网页的数据将被导出。

用户网页的删除

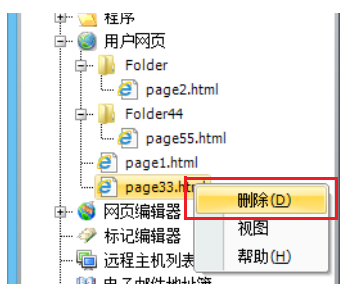
在项目窗口中删除网页树状图。

以下步骤将以从网页树状图中删除 page33.html 和 Folder44 的方法为例进行介绍。

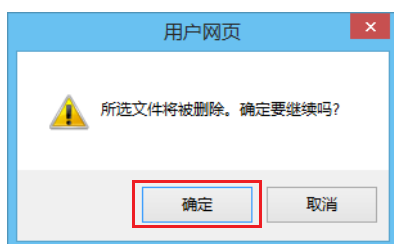


●操作过程

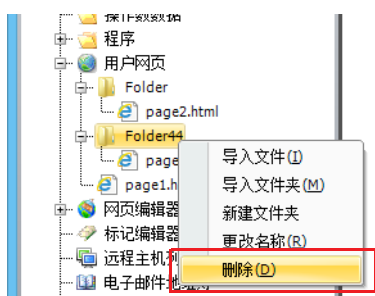
1. 右击“page33.html”，然后单击“删除”按钮。
将显示删除的确认消息。



2. 单击“确定”。
将删除“page33.html”。

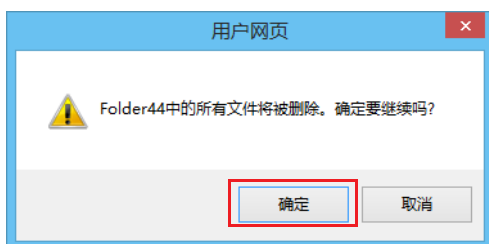


3. 右击“Folder44”文件夹，然后单击“删除”。
将显示删除的确认消息。



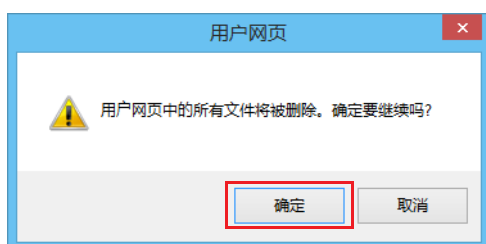
13: WEB服务器

- 单击“确定”按钮。
“Folder44”文件夹以下的所有文件及文件夹将被删除。



注释:

- 如果想要删除所有的用户网页，右击项目窗口中的“用户网页”，然后单击“全部删除”。由于将显示删除的确认消息，因此单击“确定”按钮，可删除已注册的所有文件及文件夹。



- 作为用户网页导入的 htm/html 文件，编码请使用 UTF8。此外，导入到 WindLDR 中的文件，请另行在 PC 上进行备份。Web 浏览器的动作有可能会因 Web 浏览器的种类或版本而发生变化。请使用实际访问 Web 服务器的设备的 Web 浏览器，充分进行动作验证。

监控功能

如果用户网页包含特定格式的字符串（元字符），则将该字符串替换为数据寄存器中的值或内部继电器的状态并显示。使用该功能时，请在以下对话框中选中“启用数据寄存器的监控”复选框。

- “功能设置”对话框的“以太网端口1”的“Web服务器设置”
- “HMI模块设置”对话框的“Web服务器”选项卡

注释：使用 WindLDR 转换程序时，在导入的 htm/html 文件的 head 元素内添加以下脚本元素。

```
<script type="text/javascript" src="/system/lib/jquery.js"></script>
<script type="text/javascript" src="/system/lib/idec.js"></script>
<script type="text/javascript" src="/system/lib/idec_glib.js"></script>
<script type="text/javascript" src="/system/lib/idec_tag_start.js"></script>
```

元字符的格式（字设备）

■ 数据寄存器的值

{{ 读写类型, 设备, 设备编号, 驱动器编号, 网络编号, 数据类型, 最大大小 }}

参数	说明
读写类型	可设置 R 或 W。设置为 R 时仅进行读取，W 时仅进行写入。
设备	设置希望监控的设备。固定为 D（数据寄存器）。
设备编号	以十进制设置希望监控的设备的编号。
驱动器编号	驱动器编号。设置为 0。
网络编号	设置网络编号。
数据类型	设置数据类型。*1
最大大小	如果数据类型为数值型时无需设置。 如果数据类型是字符串型，则以字节为单位设置读取或写入的最大大小。 最小值为 1，最大值为 63。

*1 从以下中设置。

种类	数据类型	说明
数值型	DEC-W	字（十进制）
	DEC-I	整数（十进制）
	DEC-D	双字（十进制）
	DEC-L	长整（十进制）
	DEC-F	浮点（十进制）
	HEX-W	字（十六进制）
	HEX-D	双字（十六进制）
字符串型	STR_ASCII	ASCII 字符串
	STR_HOSTNAME	ASCII 字符串
	STR_EMAILADR	ASCII 字符串
	STR_PASSWORD	ASCII 字符串

有关字、整数、双字、长整以及浮点的详情，请参见《梯形图编程手册》第 3 章中的“数据类型”。

注释：

- 如果在“读写型”中设置了 W（写入），单击网络浏览器上的值会显示写入对话框，并可以将值写入设备。
- 如果与 Plus CPU 模块或 HMI 模块无法正常通信，或者尝试获取不存在的设备编号的数据，则元字符将替换为 -（连字符）。

当“数据类型”中设置了字符串类型的数据类型时，读取和写入以字为单位，按照高位字节和低位字节的顺序进行。从设置设备的高位字节读取到终止符 NULL (0x00) 或“最大大小”中指定大小的数据。写入时，从设置设备的高字节开始写入“最大大小”÷2 个字（四舍五入为整数）的数据。在输入的字符串结束后，填充终止符 NULL (0x00)。

注释：

- 如果在“读写类型”中设置了 W（写）并且在“数据类型”中设置了 STR_HOSTNAME，则仅当在写对话框中输入的字符串是正确的主机格式时才执行写操作，并在写入过程开始时检查格式。
- 如果在“读写类型”中设置了 W（写入）并且在“数据类型”中设置了 STR_EMAILADR，则仅当在写入对话框中输入的字符串格式正确时才执行写入作为电子邮件地址，并在写入过程开始时检查格式。
- 当“数据类型”中设置 STR_PASSWORD 时，数据寄存器的值以隐藏字符显示。隐藏字符中使用的字符因网络浏览器而异。

■ 内部继电器的状态

{{ 读写类型, 设备, 设备编号, 驱动器编号, 网络编号, 数据类型, OFF 数据, ON 数据 }}

参数	说明
读写类型	可设置 R 或 W。设置为 R 时仅进行读取，W 时仅进行写入。
设备	设置希望监控的设备。固定为 m（内部继电器）。
设备编号	以十进制设置希望监控的设备的编号。
驱动器编号	驱动器编号。固定为 0。
网络编号	设置网络编号。固定为 0。
数据类型	设置读取或写入的数据类型。*1
OFF 数据	如果数据类型为 BIT_FORM 时，可设置已设置的内部继电器为关闭时显示的字符串。 当数据类型为 BIT 时，设置已设置的内部继电器为关闭时显示的图像。 图像由文件名和该用户网页所在文件夹的相对路径设置。
ON 数据	如果数据类型为 BIT_FORM 时，设置内部继电器为打开时显示的字符串。 如果数据类型为 BIT，设置内部继电器为打开时显示的图像。 图像由文件名和该用户网页所在文件夹的相对路径设置。

*1 从以下中设置。

种类	说明
BIT_FORM	以字符串形式显示位设备的状态。
BIT	以图像形式显示位设备的状态。

注释:

- 如果在“读写类型”中设置了 W（写）并且在“数据类型”中设置了 BIT_FORM，则显示表单按钮。内部继电器的 ON/OFF 状态显示在按钮上。可以通过单击按钮切换 ON 和 OFF。
- 如果在“读写类型”中设置了 W（写）并且在“数据类型”中设置了 BIT，则可以通过单击图像切换 ON 和 OFF。
- 如果与 Plus CPU 模块或 HMI 模块无法正常通信，或者尝试获取不存在的设备编号的数据，则元字符将替换为 -（连字符）

例

记述例	内容
{{R,m,123,0,0,BIT_FORM,BITOFF,BITON}}	M0123 为 OFF 时显示“BITOFF”。 M0123 为 ON 时显示“BITON”。
{{R,m,123,0,0,BIT,img/BIT_IMG_OFF.jpg,img/BIT_IMG_ON.jpg}}	当 M0123 为 OFF 时，显示 img 文件夹中的 BIT_IMG_OFF.jpg。 当 M0123 为 ON 时，显示 img 文件夹中的 BIT_IMG_ON.jpg。

注释: 将用户网页保存到 SD 记忆卡时，即使在“关闭数据”和“打开数据”中设置了 WEBPAGE 文件夹更上的文件夹，也会被视作设置了 WEBPAGE 文件夹。

■ 使用元字符的注意事项

在 {{ 和 }} 之间写了额外的空格字符或换行符也无任何影响。

条形图（纵）

启用监控功能时，将通过 HTML 文件格式中的 div 标签和在该 div 标签内记述参数，自动转换为 JavaScript 的请求，并以条形图（纵）形式显示通过 CGI 所获取的数据寄存器的值。

为了描绘条形图（纵），需要对 div 标签设置表示各图表固有 ID 的属性 id 字符串，然后设置“vbar”作为表示数据种类的属性 data-graph，并记述 div 标签内的参数。

可设置的参数如下所示。如果参数维持默认值即可，则无需进行设置。

device	: 设备。请设置为“D”（数据寄存器）。
address	: 设备编号。请以数值进行设置。
driver	: 驱动器编号。请设置为 0。
net_no	: 网络编号。请设置为 0。
format	: Web 数据类型的字符串。
width	: div 框的宽度（像素数）。默认值为 300。
height	: div 框的高度（像素数）。默认值为 300。
line_col	: div 框的线颜色。默认为“#000000”。
barvgutter	: 相对于图表的垂直方向距 div 框的间隔（像素数）。默认值为 30。
barwidth	: 图表的宽度（像素数）。默认值为 20。
gutter	: 显示多个图表时，各图表间的间隔（像素数）。默认值为 20。
type	: 图表顶端的形状。可从“square”（方形）、“round”（圆形）、“sharp”（锐头）、“soft”（钝头）中选择。默认为“square”。
scalewidth	: 刻度的宽度（像素数）。默认值为 5。
labelvgutter	: 垂直方向距显示标签图表的间隔（像素数）。默认值为 20。
labelhgutter	: 水平方向距显示标签图表的间隔（像素数）。默认值为 30。
bars[]	: 进行各图表设置的数组的参数。各数组的参数在 {} 内描述，数组与数组以“,”进行区隔。

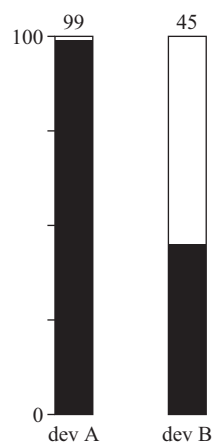
可设置为 bars[] 要素的参数如下所示。如果参数维持默认值即可，则无需进行设置。

min_val:	: 图表的最小值。必要参数。
max_val	: 图表的最大值。必要参数。
label	: 标签名称。默认为组合设备与设备编号的字符串。
back_col	: 图表的背景颜色。默认为“#FFFFFF”。
front_col	: 图表的前景颜色。默认为各要素颜色各异。
scale_on	: 有无对图表的刻度。默认值为 false（无刻度）。
scale_lbl_on	: 有无图表刻度的标签。默认值为 false（无标签）。

如果进行以下记述，则会显示以 D2040 和 D2042 的双字（十进制）数据为对象的 2 个条形图（纵）。

```
<div id="div11" data-graph="vbar">
```

```
device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",  
width:300, height:300, line_col:"#000000",  
barvgutter:30, barwidth:20,  
gutter:20, type:"square", scalewidth:5,  
bars:[  
  {  
    min_val:0, max_val:100, label:"dev A",  
    back_col:"#FFFFFF", front_col:"#0000FF",  
    scale_on:true, scale_lbl_on:true  
  },  
  {  
    min_val:0, max_val:100, label:"dev B",  
    back_col:"#FFFFFF", front_col:"#FF0000",  
    scale_on:false, scale_lbl_on:false  
  }  
]  
</div>
```



条形图（横）

启用监控功能时，将通过 HTML 文件格式中的 div 标签和在该 div 标签内记述参数，自动转换为 JavaScript 的请求，并以条形图（横）形式显示通过 CGI 所获取的数据寄存器的值。

为了描绘条形图（横），需要对 div 标签设置表示各图表固有 ID 的属性 id 字符串，然后设置“hbar”作为表示数据种类的属性 data-graph，并记述 div 标签内的参数。

可设置的参数如下所示。如果参数维持默认值即可，则无需进行设置。

device	: 设备。请设置为“D”（数据寄存器）。
address	: 设备编号。请以数值进行设置。
driver	: 驱动器编号。请设置为 0。
net_no	: 网络编号。请设置为 0。
format	: Web 数据类型的字符串。
width	: div 框的宽度（像素数）。默认值为 300。
height	: div 框的高度（像素数）。默认值为 300。
line_col	: div 框的线颜色。默认为“#000000”。
barhgutter	: 相对于图表的水平方向距 div 框的间隔（像素数）。默认值为 30。
barwidth	: 图表的宽度（像素数）。默认值为 20。
gutter	: 显示多个图表时，各图表间的间隔（像素数）。默认值为 20。
type	: 图表顶端的形状。可从“square”（方形）、“round”（圆形）、“sharp”（锐头）、“soft”（钝头）中选择。默认为“square”。
scaleshift	: 刻度的宽度（像素数）。默认值为 5。
labelvgutter	: 垂直方向距显示标签图表的间隔（像素数）。默认值为 20。
labelhgutter	: 水平方向距显示标签图表的间隔（像素数）。默认值为 30。
bars[]	: 进行各图表设置的数组的参数。各数组的参数在 {} 内描述，数组与数组以“;”进行区隔。

可设置为 bars[] 要素的参数如下所示。如果参数维持默认值即可，则无需进行设置。

min_val	: 图表的最小值。必要参数。
max_val	: 图表的最大值。必要参数。
label	: 标签名称。默认为组合设备与设备编号的字符串。
back_col	: 图表的背景颜色。默认为“#FFFFFF”。
front_col	: 图表的前景颜色。默认为各要素颜色各异。
scale_on	: 有无对图表的刻度。默认值为 false（无刻度）。
scale_lbl_on	: 有无图表刻度的标签。默认值为 false（无标签）。

如果进行以下记述，则会显示以 D2040 和 D2042 的双字（十进制）数据为对象的 2 个条形图（横）。

```
<div id="div21" data-graph="hbar">  
  device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",  
  width:300, height:300, line_col:"#000000",  
  barhgutter:30, barwidth:20,  
  gutter:20, type:"square", scalewidth:5,  
  bars:[  
    {  
      min_val:0, max_val:100, label:"dev A",  
      back_col:"#FFFFFF", front_col:"#0000FF",  
      scale_on:true, scale_lbl_on:true  
    },  
    {  
      min_val:0, max_val:100, label:"dev B",  
      back_col:"#FFFFFF", front_col:"#FF0000",  
      scale_on:false, scale_lbl_on:false  
    }  
  ]  
</div>
```



趋势图

启用监控功能时，将通过 HTML 文件格式中的 div 标签和在该 div 标签内记述参数，自动转换为 JavaScript 的请求，并以趋势图形式显示通过 CGI 所获取的数据寄存器的值。

为了描绘趋势图，需要对 div 标签设置表示各图表固有 ID 的属性 id 字符串，然后设置“trend”作为表示数据种类的属性 data-graph，并记述 div 标签内的参数。

可设置的参数如下所示。如果参数维持默认值即可，则无需进行设置。

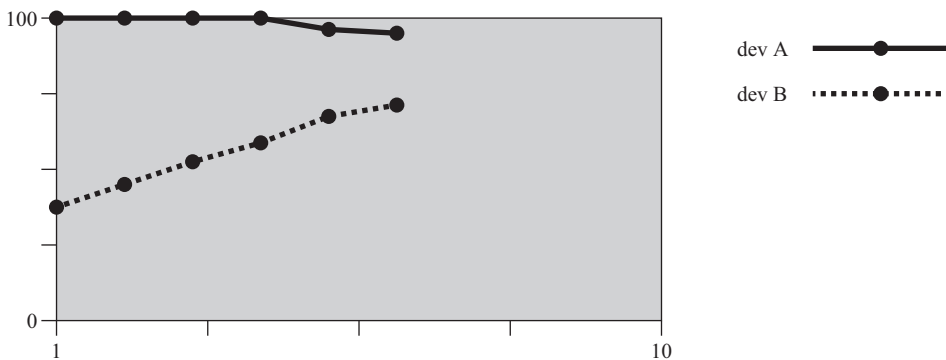
device	: 设备。请设置为“D”（数据寄存器）。
address	: 设备编号。请以数值进行设置。
driver	: 驱动器编号。请设置为 0。
net_no	: 网络编号。请设置为 0。
format	: Web 数据类型的字符串。
width	: div 框的宽度（像素数）。默认值为 300。
height	: div 框的高度（像素数）。默认值为 300。
line_col	: div 框的线颜色。默认为“#000000”。
min_val	: 图表的最小值。必要参数。
max_val	: 图表的最大值。必要参数。
plot_num	: 设置制图数据的数量。必要参数。
scale_col	: 刻度的颜色。默认为“#000000”。
x_val_col	: X 轴的标签颜色。默认为“#000000”。
y_val_col	: Y 轴的标签颜色。默认为“#000000”。
g_x	: 图表框水平方向的描绘开始位置（像素数）。 默认值为 50。
g_y	: 图表框垂直方向的描绘开始位置（像素数）。默认值为 30。
g_width	: 图表框的水平宽度（像素数）。 默认值为 200。
g_height	: 图表框的垂直宽度（像素数）。 默认值为 100。
g_line_col	: 图表框的颜色。默认为与 line_col 相同的值。
g_line_width	: 图表框的粗度（像素数）。 默认值为 1。
g_back_col	: 图表框的背景。默认为“#C0C0C0”。
mode	: 数据超过制图区域时，设置要省略的数据。可从“one”（仅省略 1 个）、“half”（省略全体的一半）、“all”（全部省略）中选择。默认为“all”。
legend_gutter	: 从图表框到凡例的宽度（像素数）。 默认值为 30。
legend_margin_x	: 凡例内水平方向的余白（像素数）。 默认值为 10。
legend_margin_y	: 凡例内垂直方向的余白（像素数）。 默认值为 10。
legend_line_width	: 凡例内线的长度（像素数）。 默认值为 20。
legend_line_gutter	: 凡例内显示多条线时垂直方向的宽度（像素数）。 默认值为 20。
legend_line_col	: 示例内的线的颜色。默认为与 line_col 相同的值。
legend_width	: 示例内水平方向的宽度（像素数）。 默认值为 100。
line_width	: 图表上线粗度的初始值（像素数）。 默认值为 1。
marker_on	: 图表上是否有标志的初始值。默认值为 false（无标志）。
marker_width	: 图表上标志大小的初始值（像素数）。 默认值为 3。
lines[]	: 进行各线设置的数组的参数。各数组的参数在 {} 内描述，数组与数组以“,”进行区隔。

可设置为 lines[] 要素的参数如下所示。如果参数维持默认值即可，则无需进行设置。

label	: 标签名称。默认为组合设备类型与设备编号的字符串。
front_col	: 各线可设置的线的前景颜色。 默认为各要素颜色各异。
marker_col	: 线标志的颜色。默认为与 front_col 相同的值。
line_width	: 各线可设置的线的粗细（像素数）。 优先度高于 line_width 的初始值。
marker_width	: 各线可设置的标志的大小（像素数）。 优先度高于 marker_width 的初始值。

如果进行以下记述，则会显示以 D2040 和 D2042 的双字（十进制）数据为对象的 10 点制图数、带标志的趋势图。

```
<div id="div31" data-graph="trend">
device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",
width:400, height:300, line_col:"#000000",
min_val:0, max_val:100, plot_num:10,
line_width:3, marker_on:true, marker_width:5,
lines:[
{
label:"dev A",
front_col:"#0000FF"
},
{
label:"dev B",
front_col:"#FF0000"
}
]
</div>
```



注释:

绘图相关注意事项

- 记述图表的 <div> ~ </div> 之间的字符串将作为 JavaScript 程序的参数传输。因此，虽然可对任意位置插入空白或改行，但无法插入注释，记述时敬请注意。
- 不设置为参数的数值请以 “” 圈起。
- 所有参数都请以 “,” 进行区隔。此外，最终参数之后请勿输入 “,”。如果参数不符合格式，将无法描绘为图表。
- 根据 Web 浏览器的种类不同，显示方法可能也有所不同。
- 根据更新频率和数据制图数等不同，Web 浏览器的显示更新有可能会变慢。

JavaScript 函数

启用监控功能时，可通过 JavaScript 的函数进行设备数据的读写。

由于可使用通过 CGI 获取的原始数据，因此与使用元数据时相比，可进行更加复杂的处理。有关 CGI 的接口，请参见附录。

■ 设备数据读取函数

`Idec.device_read (device, address, length, driver, net_no)`

- `device` : 以字符串设置读取对象设备。
当前仅可使用 D（数据寄存器）。
- `address` : 设置希望读取的设备编号。
- `length` : 设置希望以 `address` 为起始读取的数据的大小。
可以十进制在 1 ~ 64 范围内进行设置。
- `driver` : 驱动器编号。请设置为 0。
- `net_no` : 网络编号。请设置为 0。

返回值为十六进制数值的字符串、“XXXX”形式。`length` 为 2 以上时，将在数据与数据之间插入“_”。服务器无响应时或参数不正确时，返回值为“”。

注释： D2058 为 49910（C2F6h）、D2059 为 59768（E978h）时，如下所示。

```
var raw_data = idec.device_read ("D", 2058, 2, 0, 0);
```

正常结束时的 `raw_data` 为“C2F6_E978”。

■ 设备数据写入函数

`idec.device_write (device, address, length, driver, net_no, data)`

- `device` : 以字符串设置写入对象设备。当前仅可使用 D（数据寄存器）。
- `address` : 设置希望写入的设备编号。
- `length` : 设置希望以 `address` 为起始写入的数据的大小。可以十进制在 1 ~ 64 范围内进行设置。
- `driver` : 驱动器编号。请设置为 0。
- `net_no` : 网络编号。请设置为 0。
- `data` : 以十六进制数值的字符串设置希望写入的数据。`length` 为 2 以上时，将在数据与数据之间插入“_”。

返回值在正常结束时为 `true`，除此之外则为 `false`。

注释： D2058 中写入 49910（C2F6h）、D2059 中写入 59768（E978h）时，如下所示。

```
var status_write = idec.device_write ("D", 2058, 2, 0, 0, "C2F6_E978");
```

正常结束时的 `status_write` 为 `true`。

自定义网页

可以使用 WindLDR 提供的网页编辑器，根据目的和用途轻松地创建网页（自定义网页）。有关创建自定义网页的详情，请参见第 13-44 页上的“网页编辑器”。

网页编辑器

使用网页编辑器可创建 FC6A 型中下载自定义网页。

FC6A 型中下载的自定义网页可通过 PC 等终端以 Web 浏览器进行访问，且可显示 FC6A 型的各设备的值所对应的自定义网页。

要使用网页编辑器时，需要单击模块构成区域中所插入的 HMI 模块，然后在单击“设置”按钮所显示的“HMI 模块设置”对话框中选中“使用网页编辑器”复选框。

通过联动所创建的自定义网页与 WindLDR 的模拟功能，可确认 FC6A 型各设备的值所对应的自定义网页的动作，而非将自定义网页下载到 FC6A 型中。

自定义网页的创建

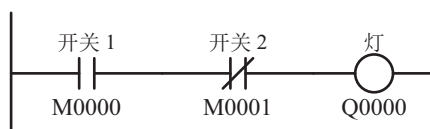
本节将对使用网页编辑器创建自定义网页的操作过程进行介绍。

创建输入 M0000 为 ON、输入 M0001 为 OFF 时输出 Q1 开启的网页。

网页中配置的显示部件



使用 WindLDR 创建的程序



注释:

有关设备的详情，请参见第 2-1 页上的“设备地址”。

有关指令的详情，请参见《梯形图编程手册》。

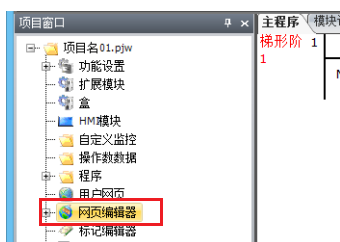
●操作过程

1. 创建输入 M0000 为 ON、输入 M0001 为 OFF 时输出 Q1 开启的程序。

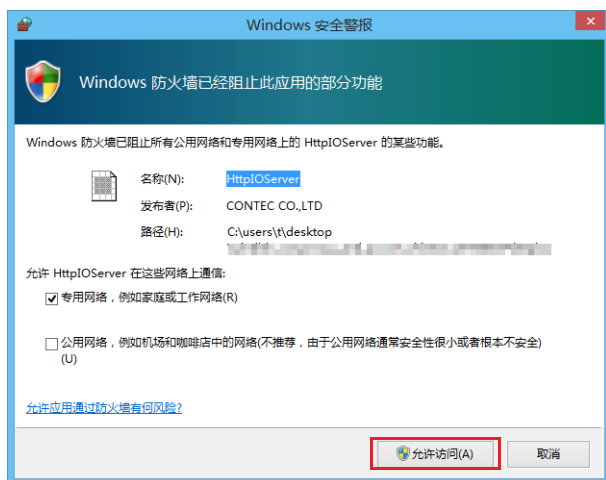
有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 4 章中的“创建一个梯形图程序”。



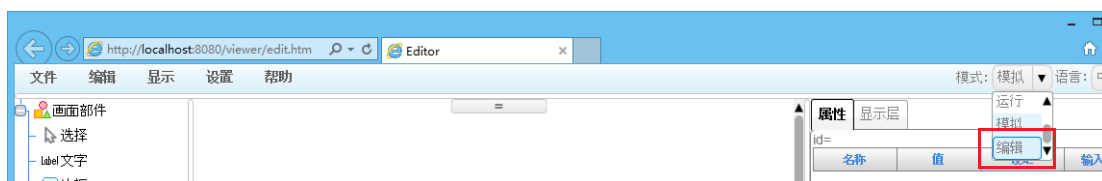
2. 双击项目窗口中的“网页编辑器”。
- 网页编辑器启动。



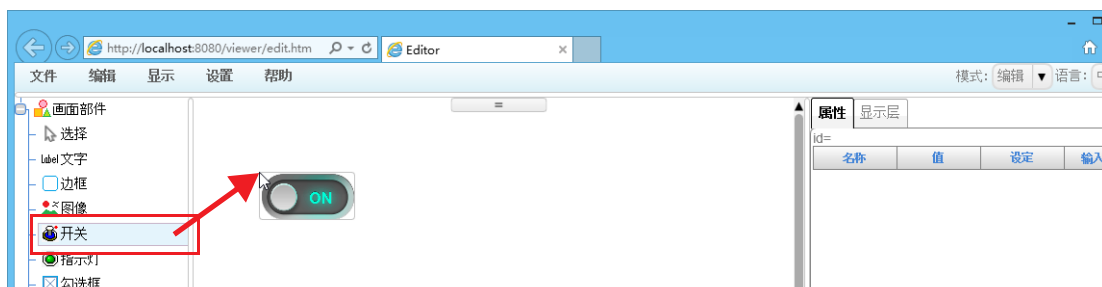
注释：显示“Windows 安全警报”对话框时，单击“允许访问”按钮。




3. 在网页编辑器的“模式”中单击“编辑”。

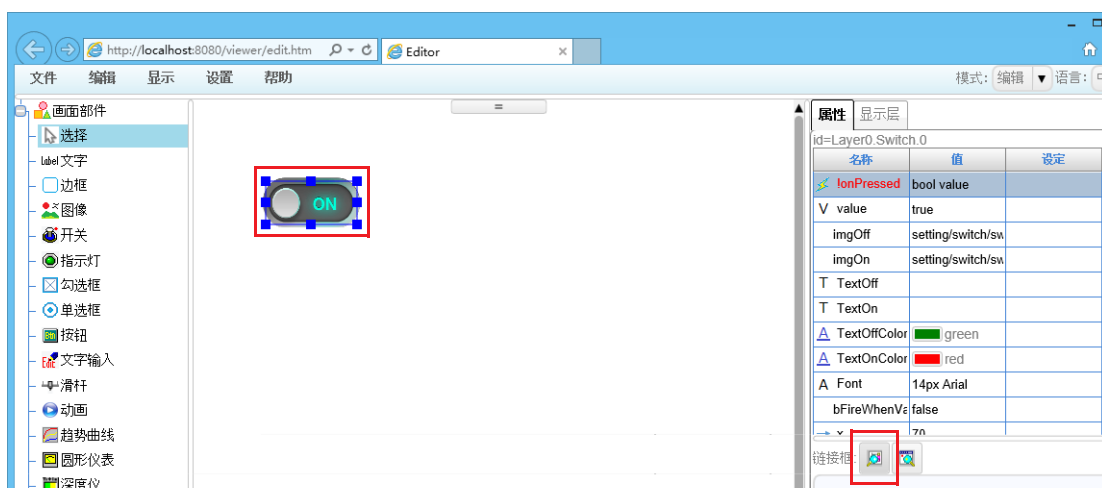


4. 配置为开关 1，设置 A 常开（M000）。
从“画面部件”列表中选择“开关”，拖放到制图区域中。
开关将以规定大小进行配置。



注释：如果以任意大小配置开关，从“画面部件”列表中单击“开关”，然后在制图区域中拖动十字光标。

5. 单击所配置的开关，然后再单击“属性”选项卡“链接框”中的  按钮。
将显示“设备树”对话框。



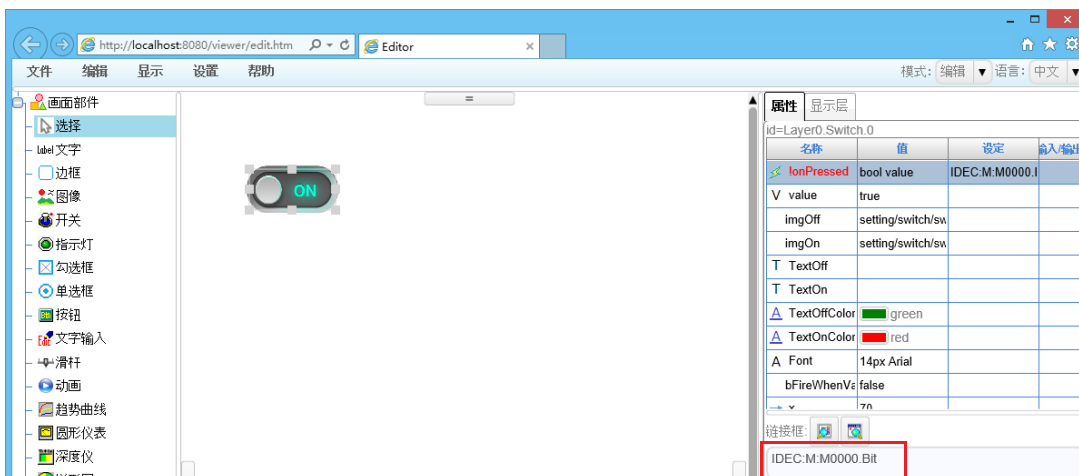
6. 单击开关 1 中设置的设备地址，然后单击“确认”按钮。



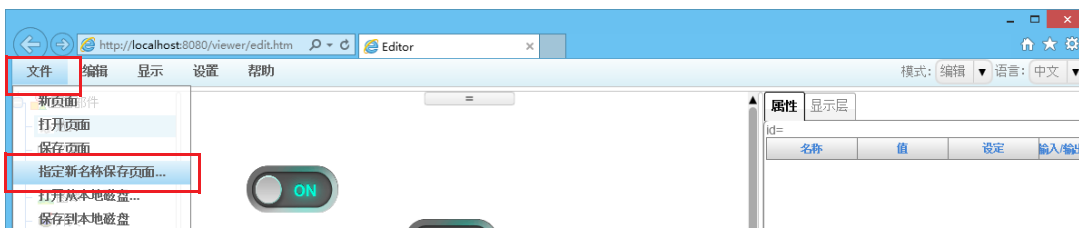
注释:

- “设备树”对话框中将显示正在 WindLDR 中使用的设备。
- 如果没有希望设置到设备树中的设备地址，请单击“放弃”按钮，在“属性”选项卡“链接框”中的文本框中直接输入设备地址。

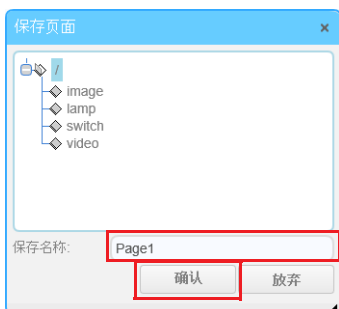
至此，开关 1 已设置为 A 常开（M0000）。



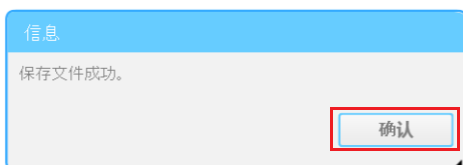
7. 重复步骤 4 ~ 6 的操作，设置开关 2、灯。
8. 在菜单栏的“文件”菜单中单击“指定新名称保存页面”。将显示“保存页面”对话框。



9. 在“保存名称”中输入文件名，并指定保存位置文件夹后单击“确认”按钮。将显示确认消息。



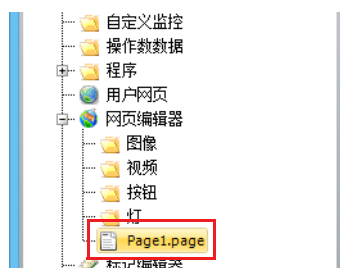
10. 单击“确认”按钮。



至此，完成自定义网页的创建。

注释：

- 有关网页编辑器的详情，请在菜单栏的“帮助”菜单中单击“帮助”，参见所显示网页编辑器的帮助内容。
- 所创建的自定义网页将被创建为项目窗口的网页编辑器。此外，自定义网页中所导入的图像文件、视频文件、开关部件的图像文件、灯部件的图像文件将一览显示在各自的文件夹内。
双击已创建的自定义网页，网页编辑器启动，并打开已创建的自定义网页。可编辑已创建的自定义网页。



自定义网页的动作确认

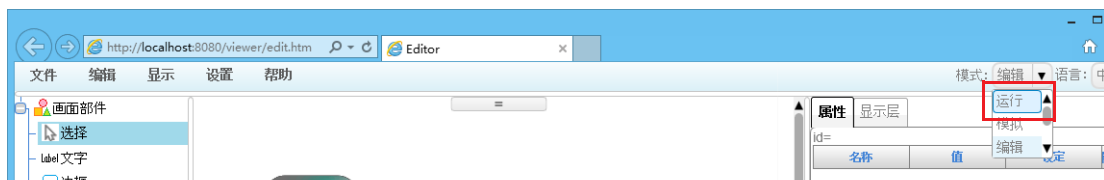
本节将对使用网页编辑器执行所创建自定义网页的动作，以及确认动作的操作过程进行介绍。

●操作过程

1. 在 WindLDR 的“联机”选项卡中单击“模拟”。

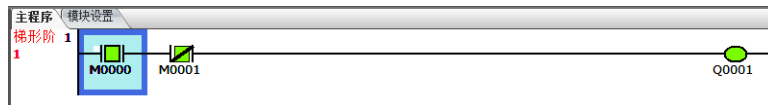
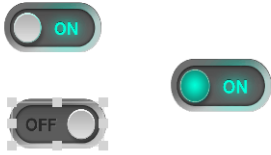


2. 在网页编辑器的“模式”中单击“运行”。

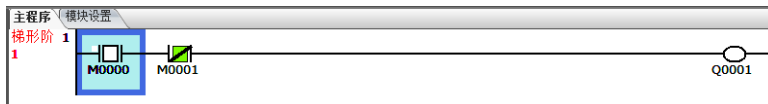
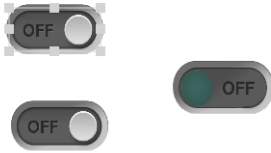


3. 使开关 1 和开关 2 开启 / 关闭，确认以下动作。

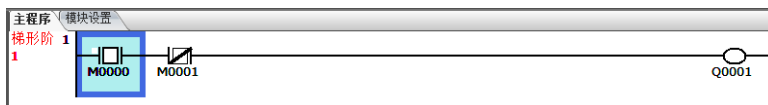
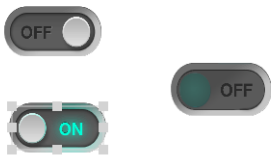
- 开关 1 为 ON 且开关 2 为 OFF 时，灯应开启



- 开关 1 为 OFF 时，灯应关闭



- 开关 2 为 ON 时，灯应关闭



注释：WindLDR 的程序也将根据网页编辑器的状态进行开启 / 关闭。

至此，完成动作确认。

保存所创建的自定义网页后，如果通过 WindLDR 将用户程序下载到 FC6A 型中，则可通过 Web 浏览器观看自定义网页。

有关设置方法的详情，请参见第 13-19 页上的“HMI 模块的 Web 服务器设置”。

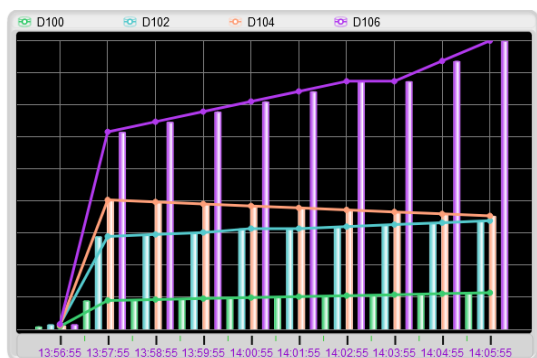
自定义网页的日志数据联动

使用网页编辑器创建的自定义网页可与日志数据联动。可联动的日志数据如下所示。

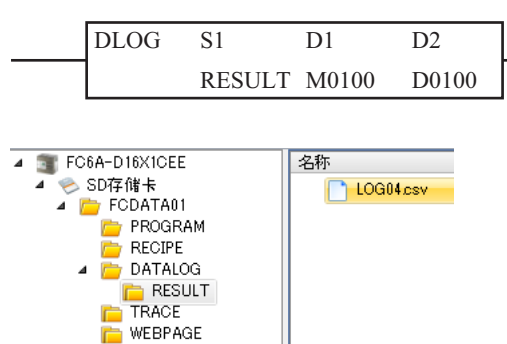
- 以 DLOG（数据日志）指令保存在 SD 记忆卡中的日志数据

本节将对通过使用网页编辑器创建的自定义网页，参照以 DLOG（数据日志）指令保存在 SD 记忆卡中的日志数据的操作过程进行介绍。

网页中配置的部件



使用 WindLDR 创建的程序



注释:

- 有关 DLOG（数据日志）指令的详情，请参见《梯形图编程手册》第 25 章中的“DLOG（数据日志）”。
- 自定义网页的日志数据联动仅在 Plus CPU 模块的内存以及 Plus CPU 模块上插入的 SD 记忆卡中使用。日志数据无法与保存到 HMI 模块内存的自定义网页联动。
- 在外部存储器的设置中，将日志数据 CSV 文件格式的分隔字符设置为分号（;）时，日志数据无法与自定义网页联动。有关外部存储器设置的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“功能和设置”。
- 作为自定义网页的日志数据联动部件，网页编辑器中具备生产状态图部件、表部件、列表部件、下拉列表部件，使用这些部件也可设置日志数据联动。



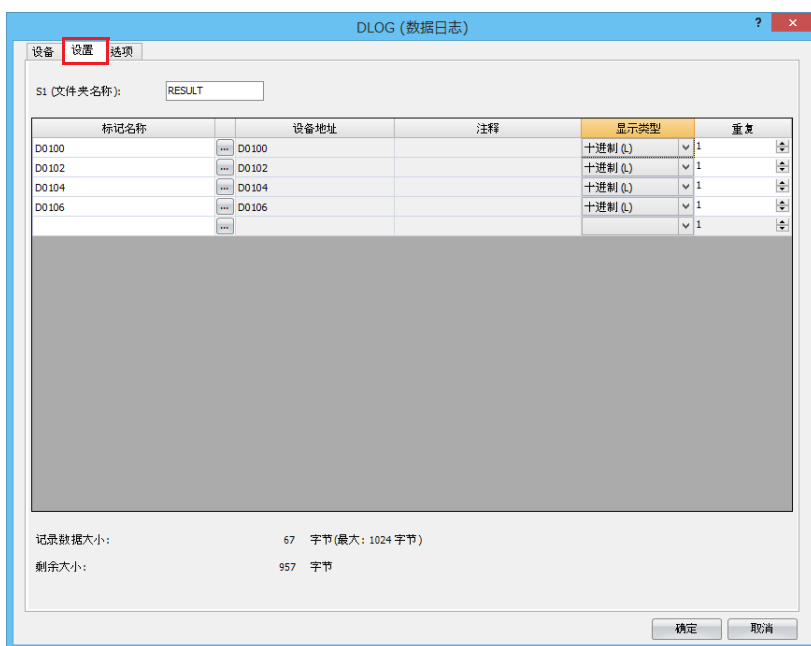
- 使用 WindLDR 的模拟功能时，日志数据无法与自定义网页联动。

●操作过程

1. 插入 DLOG 指令。



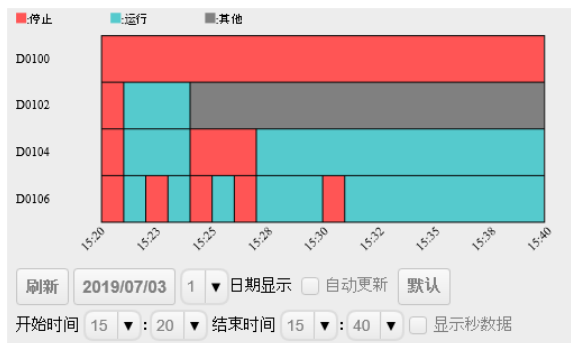
2. 在“设置”选项卡中设置输出到 CSV 文件的数据设备地址。
已设置的设备地址数值，将显示在自定义网页日志数据联动的部件中。



注释: 在自定义网页日志数据联动部件中使用生产状态图部件时，请通过 DLOG 指令编写梯形图程序，以确保输出到 CSV 文件的数据数值与生产状态图的输入值一致。

在本示例中，将输出到 CSV 文件的数据数值作为 0: 停止、1: 运行的生产状态，输出到生产状态图部件。

生产状态图



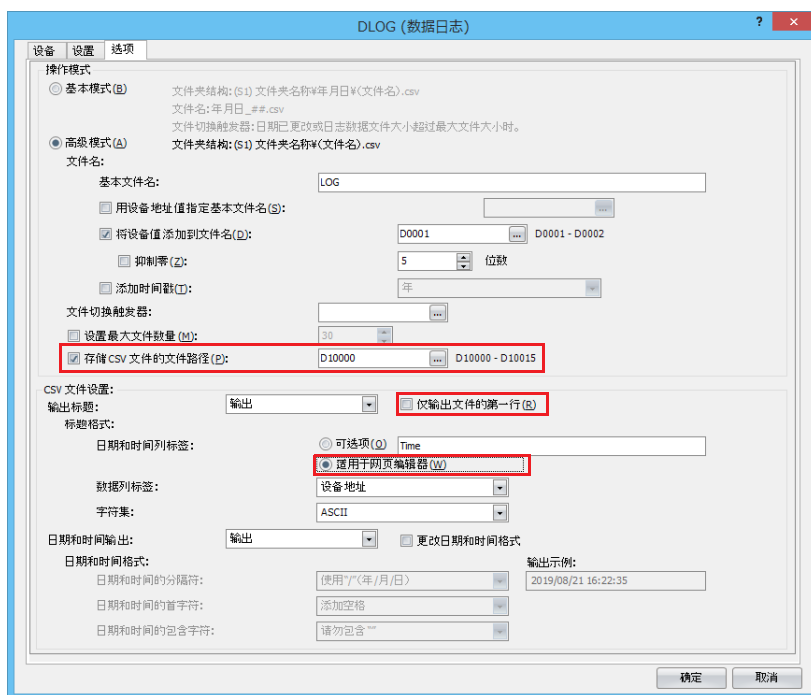
The '选项' (Options) dialog box for the production status chart. It has tabs for '示例' (Example), '一般' (General), and '标题' (Title). The '示例' tab is active, showing a table for mapping input values to output colors:

项目	输入值(≤)	输出颜色
停止	0	红色 #FF5555
运行	1	青色 #55CADC
其他		灰色 #808080

CSV 文件

#_date_time_	D0100	D0102	D0104	D0106
2019/7/3 15:18:01	0	0	0	0
2019/7/3 15:19:01	0	0	0	1
2019/7/3 15:20:01	0	0	0	0
2019/7/3 15:21:01	0	1	1	1
2019/7/3 15:22:01	0	1	1	0
2019/7/3 15:23:01	0	1	1	1
2019/7/3 15:24:01	0	2	0	0
2019/7/3 15:25:01	0	3	0	1
2019/7/3 15:26:01	0	4	0	0
2019/7/3 15:27:01	0	5	1	1
2019/7/3 15:30:15	0	6	1	0
2019/7/3 15:31:15	0	7	1	1

3. 在“输出标题”中选中“仅输出文件的第一行”复选框。
 在“标题格式”的“日期和时间列标签”中选择“适用于网页编辑器”。
 选中“存储 CSV 文件的文件路径”复选框，指定执行存储的起始数据寄存器。



“存储 csv 文件的文件路径”：已选中时，仅需在趋势棒状图部件的“链接框”中设置设备地址，即可进行图表显示。

未选中时，如果未直接在趋势棒状图部件的属性中指定 CSV 文件的文件路径，则不会进行图表显示。

“仅输出文件的第一行”：选中时，如果将 FC6A 型的状态由停止变为运行，将作为标题内容反映到趋势棒状图部件的 X 轴上。

未选中时，即使将 FC6A 型的状态由停止变为运行，也不会趋势棒状图部件中进行图表显示。

“标题格式”

“日期和时间列标签”：选择了“适用于网页编辑器”时，将作为标题内容反映到趋势棒状图部件的 X 轴上。选择了“可选项”时，不会在趋势棒状图部件中进行图表显示。

“输出标题”：设置为输出时，将作为标题内容反映到趋势棒状图部件的 X 轴上。设置为不输出时，不会在趋势棒状图部件中进行图表显示。

关于上述以外的“选项”设置，无需进行趋势棒状图部件的设置。有关“选项”设置的详情，请参见《梯形图编程手册》第 25 章中的“DLOG（数据日志）”。

13: WEB服务器

注释：如需在网页编辑器中联动日志数据并显示 ASCII 字符以外的字符串，请在“字符集”中选择“带 BOM 的 Unicode (UTF-8)”或“无 BOM 的 Unicode (UTF-8)”之一。

“带 BOM 的 Unicode (UTF-8)”时，BOM 值 (0xEF 0xBB 0xBF) 会输出为开始 3 字节。

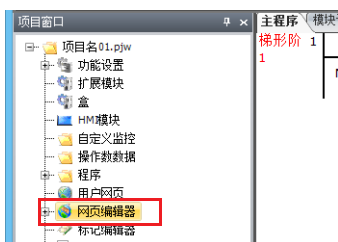
用特定应用程序打开“无 BOM 的 Unicode (UTF-8)”时，无法以“UTF-8”打开，可能会出现乱码。

使用表部件、列表部件、下拉列表部件时，“选项”选项卡的“常规设置”中设置的内容因各部件中可使用的属性而异。

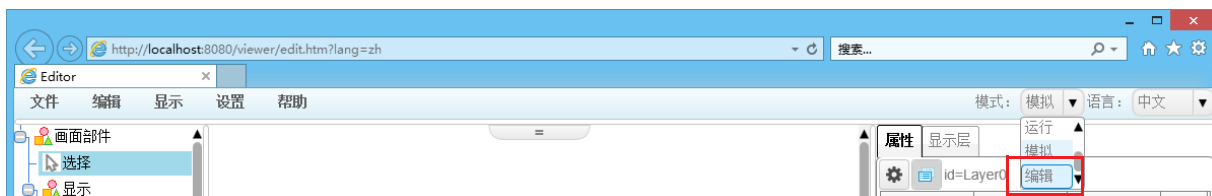
有关各部件中可使用的属性，请在网页编辑器菜单栏的“帮助”菜单中单击“帮助”，参见所显示网页编辑器的帮助内容。

4. 双击项目窗口中的“网页编辑器”。

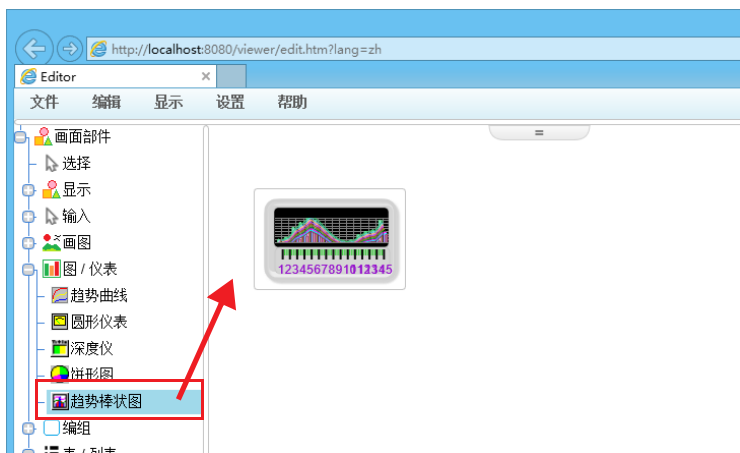
网页编辑器启动。



5. 在网页编辑器的“模式”中单击“编辑”。




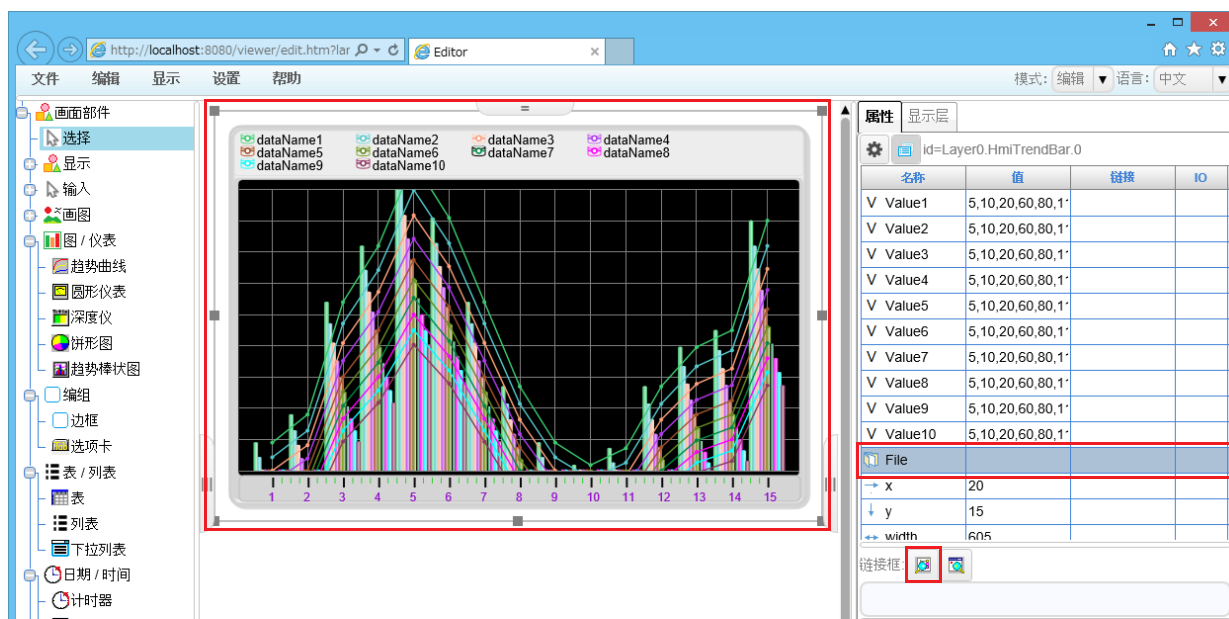
6. 配置趋势棒状图，设置数据寄存器（D0100）。
从“画面部件”列表中选择“趋势棒状图”，拖放到制图区域中。
趋势棒状图将以规定大小进行配置。



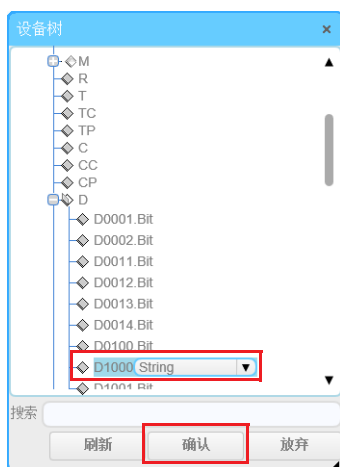
注释：

- 如需以任意大小配置趋势棒状图，从“画面部件”列表中单击“趋势棒状图”，然后在制图区域中拖动十字光标。
- 如需使用表部件、列表部件、下拉列表部件，请选择“表”、“列表”、“下拉列表”，拖放到制图区域中。有关表部件、列表部件、下拉列表部件的详情，请在菜单栏的“帮助”菜单中单击“帮助”，参见所显示网页编辑器的帮助内容。

7. 单击所配置的趋势棒状图，再单击“属性”选项卡的“File”，然后单击“链接框”的  按钮。将显示“设备树”对话框。



8. 单击趋势棒状图中设置的设备地址，然后单击“确认”按钮。



注释:

- “设备树”对话框中将显示正在 WindLDR 中使用的设备。
- 如果没有希望设置到设备树中的设备地址，请单击“放弃”按钮，在“属性”选项卡“链接框”中的变量链接输入区域中直接输入设备地址。
- 将文件路径直接指定为 File 属性的值时，将在文件夹名称的前面附加“<D>”。
文件路径的格式：
<D>\ 文件夹名称 \CSV 文件名
此外，使用 DLOG 指令时，更新过程中的文件的文件路径将存储于“存储 CSV 文件的文件路径”所指定的数据寄存器中。

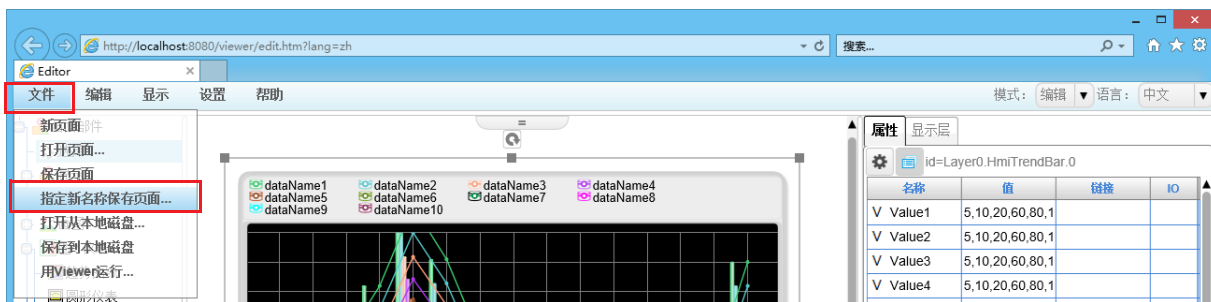
9. 双击所配置的趋势棒状图。
将显示趋势棒状图的“选项”对话框。

10. 在“X-轴”选项卡的“类型”中选择“CSV文件”。



注释：“选项”对话框中包括“一般”选项卡、“数据项”选项卡、“主面板”选项卡以及“示例图”选项卡，可设置部件内的颜色、范围、数据项目、单位、工具小部件、凡例等。

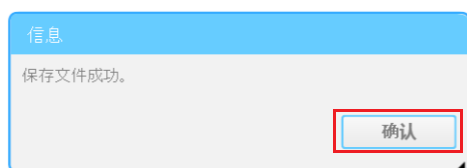
11. 在菜单栏的“文件”菜单中单击“指定新名称保存页面”。
将显示“保存页面”对话框。



12. 在“保存名称”中输入文件名，并指定保存位置文件夹后单击“确认”按钮。
将显示确认消息。

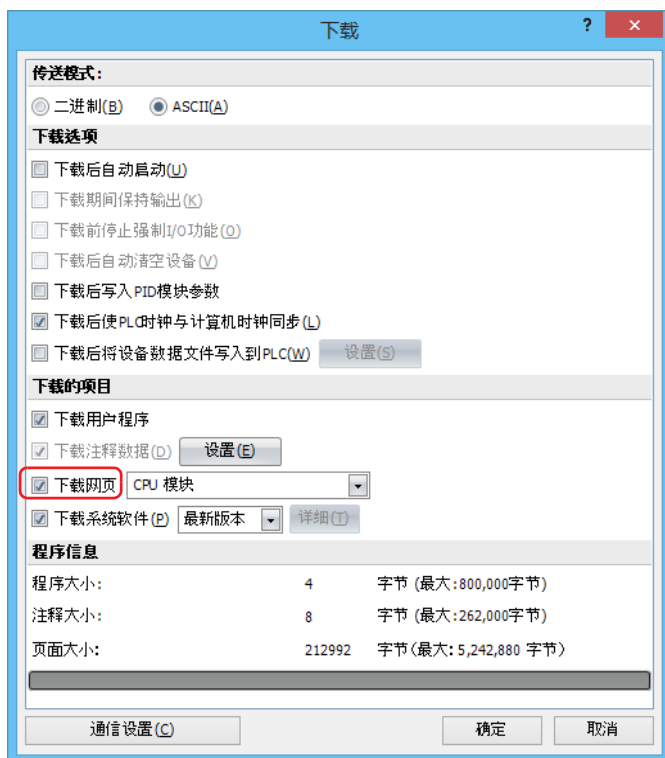


13. 单击“确认”按钮。



14. 将用户程序下载到 Plus CPU 模块。
在“联机”选项卡的“传送”中单击“下载”按钮。
将显示“下载”对话框。

15. 选中“下载网页”复选框。



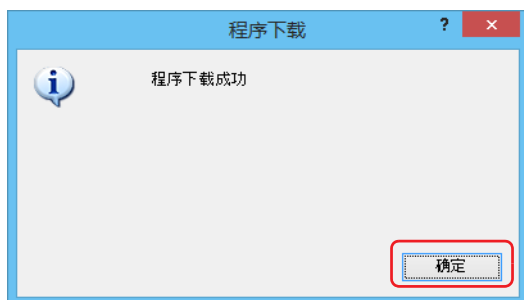
16. 网页保存位置选择为“CPU 模块”或“CPU 模块（SD 记忆卡）”。

注释: 在 Plus CPU 模块上未插入 SD 记忆卡的状态下选择“CPU 模块（SD 记忆卡）”时，不会下载网页。

17. 单击“确定”按钮。

网页将下载至已选保存位置。

显示以下信息后表示下载成功。单击“确定”按钮。



18. 运行 DLOG（数据日志）指令时，CSV 文件将保存到 SD 记忆卡中。可通过自定义网页以图表显示的形式查阅保存的 CSV 文件。

注释: 如果 CSV 文件的文件大小较大，可能经过较长时间才会进行网页的图表显示。

请根据通信回线的状况，调整 CSV 文件的文件大小上限。

有关日志数据文件大小的上限设置，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“功能和设置”。

CGI

Plus CPU 模块和 HMI 模块支持设备数据的读取与写入的 CGI 程序。本节介绍 CGI 格式。

请求格式（读取设备数据）

可使用 GET 方法用以下格式从网络服务器 CPU 模块读取设备数据。

```
/system/device_read.cgi?device=DEV&address=XXXX&length=XXXX&driver=XX&net_no=XX
```

CGI 参数如下：

/system/device_read.cgi	: CGI 文件名
device	: 用字符指定目标设备的设备类型。支持“D”（数据寄存器）。
address	: 以十六进制指定目标设备的设备编号。
length	: 以十六进制指定要读取的数据寄存器的数目。有效范围为 1 至 64（0001h 至 0040h）。
driver	: 预留给系统。指定“00”。
net_no	: 预留给系统。指定“00”。

例如：从 D2058 读取两个连续的数据寄存器时，CGI 的格式如下：

```
/system/device_read.cgi?device=D&address=080A&length=0002&driver=00&net_no=00
```

答复格式（读取设备数据）

网络服务器 CPU 模块在收到上述请求后，返回以下答复。

```
Content-type: text/javascript+json; charset=utf-8
{
  "STATUS": "0",
  "DEVICE": "D",
  "ADDRESS": "080A",
  "LENGTH": "0002",
  "DRIVER": "00",
  "NET_NO": "00",
  "DATA": "C2F6_E978"
}
```

STATUS: 0 表示正常答复。任何其他值表示错误。

DEVICE、ADDRESS、LENGTH、DRIVER 和 NET_NO: 存储请求中的参数。

DATA: 存储指定设备的值。当长度大于或等于 2 时，在设备值之间插入 "_"。

请求格式（写入设备数据）

可使用 POST 方法用以下格式向网络服务器 CPU 模块写入设备数据。

```
/system/device_write.cgi?device=DEV&address=XXXX&length=XXXX&driver=XX&net_no=XX&data=XXXX
```

CGI 参数如下：

/system/device_write.cgi	: CGI 文件名
device	: 用字符指定目标设备的设备类型。支持 "D"（数据寄存器）。
address	: 以十六进制指定目标设备的设备编号。
length	: 以十六进制指定要写入的数据寄存器的数目。有效范围为 1 至 64（0001h 至 0040h）。
driver	: 预留给系统。指定 "00"。
net_no	: 预留给系统。指定 "00"。
data	: 以十六进制指定写入到目标数据寄存器中的值。

当长度大于或等于 2 时，在写入数据寄存器的值之间插入 "_"。

例如：向 D2058 写入 49910 (C2F6h) 以及向 D2059 写入 59768 (E978h) 时，CGI 的格式如下：

```
/system/device_write.cgi?device=D&address=080A&length=0002&driver=00&net_no=00&data=C2F6_E978
```

答复格式（写入设备数据）

网络服务器 CPU 模块在收到上述请求后，返回以下答复。

```
Content-type: text/javascript+json; charset=utf-8
{
  "STATUS": "0",
  "DEVICE": "D",
  "ADDRESS": "080A",
  "LENGTH": "0002",
  "DRIVER": "00",
  "NET_NO": "00",
}
```

STATUS: 0 表示正常答复。任何其他值表示错误。

DEVICE、ADDRESS、LENGTH、DRIVER 和 NET_NO: 存储请求中的参数。

SD 记忆卡读取（文件下载）

通过 Plus CPU 模块使用 Web 服务器功能时，可以将保存在 SD 记忆卡中的 WEBPAGE 文件夹或 DATALOG 文件夹中的文件下载到计算机中。

下载保存在 SD 记忆卡上的 WEBPAGE 文件夹中的文件时，请访问 SD 记忆卡上的网页的 URL（参见“显示为访问目的地的网页”（第 13-4 页）），指定 SD 记忆卡中 WEBPAGE 的文件夹路径。

下载保存在 SD 记忆卡的 DATALOG 文件夹中的文件时，在访问内存的网页的 URL 中输入 /viewer/（参见第 13-4 页上的“访问目的地和显示网页”）。添加 <D>/ 指定 SD 记忆卡中 DATALOG 的文件夹路径。

例

`http://192.168.1.5/sd/viewer/img/picture.png`

SD 记忆卡中的 \FCDATA01\WEBPAGE\viewer\img\picture.png 文件被下载到 PC 中。

`http://192.168.1.5/viewer/%3CD%3E/RESULT/LOG04.csv`

SD 记忆卡中的 \FCDATA01\DATALOG\RESULT\LOG04.csv 文件被下载到 PC 中。

注释：当文件下载中时，网页的更新会暂停几秒钟。

限制事项

启用 SSL/TLS 通信时，有以下限制。

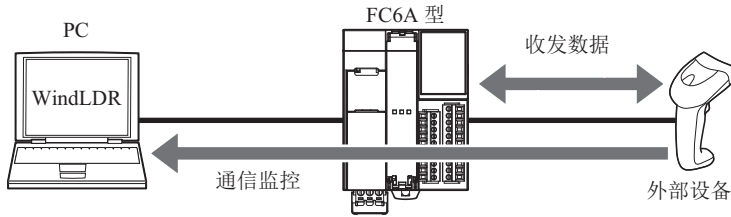
- 网页完全显示需要花费少许时间。
- 如果在访问网页，由其他页浏览器访问同一网页的话，网页的更新会暂停几秒钟。
- 如果在低速或代理的网络环境中访问网络服务器，可能会超过网络服务器上最大同时连接数，而使网页无法正常显示。在自定义网页，如拆分网页请减少放置在一个网页上的部件种类。

14: 通信监控

本章将对通信监控功能进行介绍。

概要

通信监控是 FC6A 型将与外部设备收发的通信数据在 WindLDR 的“通信监控”对话框中进行监控的功能。可监控的通信协议为用户通信及 Modbus 通信。通信监控仅可使用 Plus CPU 模块。

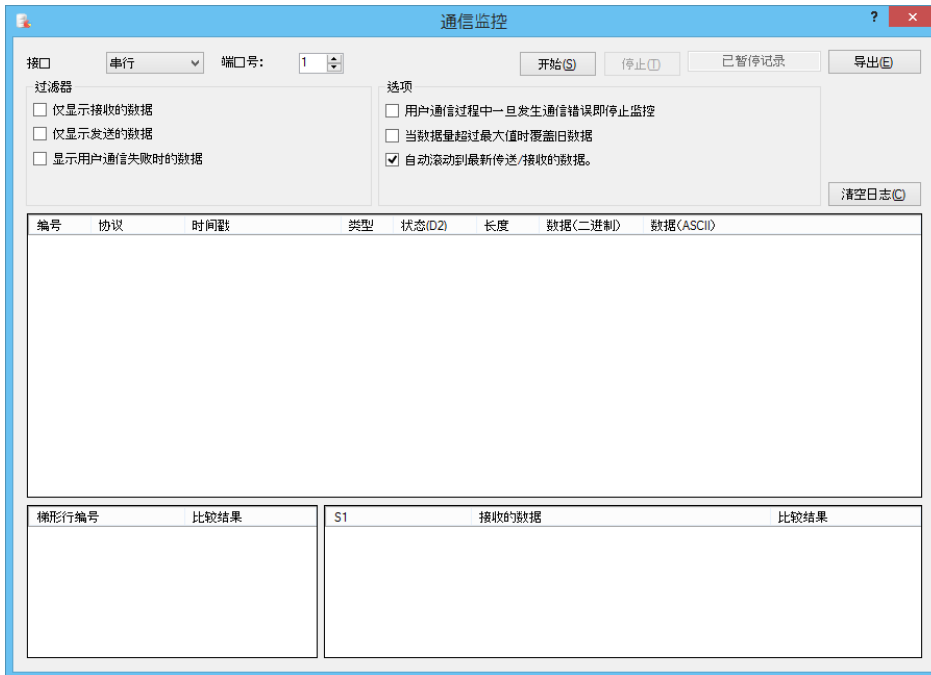


通信监控的开始和停止

本节将对启动通信监控、开始及停止与外部设备监控的方法进行介绍。

●操作过程

1. 在“联机”选项卡的“监控”中单击“通信监控”。



监控的外部设备由通信接口、端口编号及连接编号指定。

监控的通信数据为从单击“开始”按钮时由 FC6A 型暂存在内部内存中的数据。通过单击“停止”按钮或关闭“通信监控”对话框，会停止保存到内部内存。因此，无法找到通信监控未监控期间的收发数据。

可监控的数据最多为 65,535 件。一旦到达最大数据件数，就会自动停止监控。即使到达最大数据件数仍想要继续监控时，请选中“当数据量超过最大值时覆盖旧数据”复选框。从旧数据开始依序清空并继续监控。

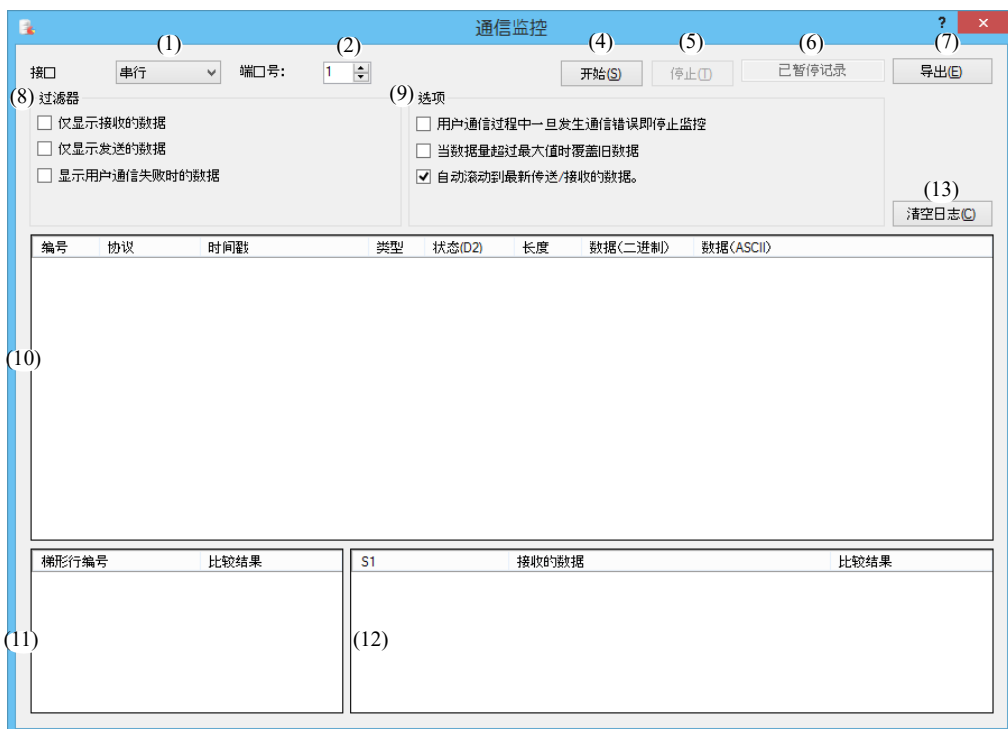
WindLDR 上的通信监控更新收发数据的显示间隔大于与监控的外部设备之间的通信间隔时，FC6A 型暂存数据的内存空间可能会不足。此时，将停止保存到内存。

在全部显示内存中暂存的所有收发数据后，停止保存到内存后的通信监控显示将切换为监控停止状态。

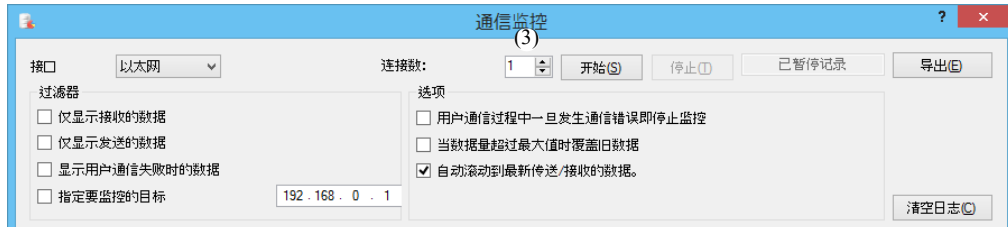
14: 通信监控

通信监控

串行通信的监控



以太网通信的监控



(1) 接口

将监控的通信接口选择为“串行”或“以太网”。

(2) 端口号

设置监控的通信接口的端口。仅接口(1)为“串行”时可进行设置。

端口：1～33

(3) 连接数

设置监控的连接编号。

仅接口(1)为“以太网”时可进行设置。

连接：1～16

(4) “开始”按钮

单击此按钮，开始通信监控。WindLDR会以固定周期通过PLC获取通信数据。

(5) “停止”按钮

单击此按钮，停止通信监控。

停止中，将停止累积PLC的通信数据。

(6) 状态显示

将显示通信监控的监控状态。

(7) “导出”按钮

可导出收发数据显示 (10) 中显示的通信数据。

单击此按钮，在“另存为”对话框中输入文件名，然后单击“保存”按钮。

(8) 过滤器

筛选收发数据显示 (10) 中显示的项目。选中显示内容的复选框。

可设置多选的过滤器。

选中“指定要监控的目标”复选框时，输入 IP 地址。

(9) 选项

在下次处理中使监控运行时，应选中复选框。

用户通信过程中一旦发生通信错误即停止监控：

在发生错误时停止监控。

当数据量超过最大值时覆盖旧数据：

如果通信数据到达最大数据件数（65,535 件）时不停止监控，就会从旧数据开始依序清空并继续监控。

自动滚动到最新传送 / 接收的数据。：

添加收发数据后，自动显示最新的收发数据。

(10) 收发数据显示

将显示 FC6A 型和外部设备的通信数据。显示内容如下所示。

- 协议
- 时间戳（Plus CPU 模块的系统版本为 1.01 以上）
- 收发类型
- 发送 / 接收对象的 IP 地址（仅以太网时）
- 状态
- 数据长度
- 数据（二进制及 ASCII）

注释：以监控开始时刻的 FC6A 型的日历 / 时钟数据为基准，并加上监控开始后对收发数据附加的时间差，显示为时间戳。其目的在于显示通信数据的通信间隔标准，可能与 PC 的当前时间不一致，或因调整夏时制等导致日期时间出现偏差。此外，时间戳显示类型根据 PC 的日历 / 时钟显示类型而定。

(11) 选择用户通信指令

将显示收发数据显示 (10) 中所选发送及接收数据的用户通信一览。协议为 Modbus 时无法显示。

梯形行编号：将显示 WindLDR 上的用户通信指令（RXD/ERXD/TXD/ETXD）的行编号。

相应的用户通信指令会根据通过 PLC 获取的 D2 设备地址，搜索 WindLDR 上的指令而决定。

比较结果：将显示各用户通信接收指令的接收结果。（发送时无法显示。）

(12) 核对结果的详情

将显示选择用户通信指令 (11) 中所选的用户通信指令的格式（S1）与通信数据的核对结果详情。因此，可确认用户通信接收指令的格式与实际接收数据的差异。但协议为 Modbus 时无法显示。

S1：将显示选择用户通信指令 (11) 中所选的用户通信接收指令的格式。

- 常量（字符）
- 常量（十六进制）
- 数据寄存器
- BCC
- 跳过（仅接收数据）

接收的数据 / 发送的数据：将显示通信数据的内容。

比较结果：将显示比较结果。

一致：OK

不一致：NG

注释：对于以 WindLDR 上的用户通信指令指定的格式，可核对 FC6A 型接收的数据，并关联与格式的核对结果进行显示。因此，需要将 WindLDR 上的用户程序和 FC6A 型内的用户程序保持一致。如果用户程序不一致，在显示“通信监控”对话框时，将显示无法显示正确分析结果的警告，并仅显示收发数据一览显示 (10)。

14: 通信监控

- (13) “清空日志”按钮
清空显示的通信数据。

15: BACnet/IP

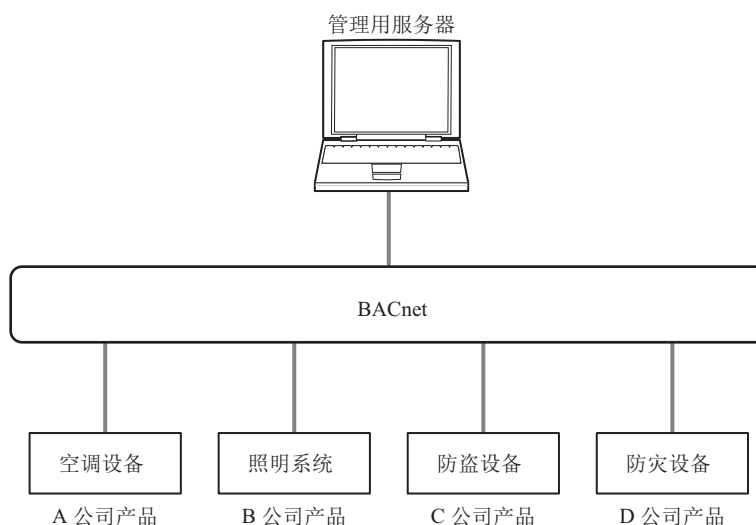
本章将对 BACnet 通信功能进行介绍。

概要

Plus CPU 模块支持使用 Internet Protocol (IP) 的 BACnet 通信 (BACnet/IP)。

BACnet 是“Building Automation and Control Networking Protocol”的缩写，属于楼宇网络用通信规格，是一种对楼宇设备中不同厂商所构建的系统进行互连的标准化开放式协议。

以往空调、照明、防盗、防灾系统等均采用楼宇设备、系统厂商独创的方式进行连接，如今通过使用 BACnet 协议，可以通用方式进行连接、监视等。



BACnet/IP 中使用的通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
以太网端口 1	—	—	—	—	是	是
以太网端口 2	—	—	—	—	—	—
HMI- 以太网端口	—	—	—	—	—	—

端口 1 ~ 33 不能使用 BACnet/IP。

BACnet 规格

型号	FC6A-D16R1CEE、 FC6A-D16R4CEE FC6A-D16K1CEE、 FC6A-D16K4CEE FC6A-D16P1CEE、 FC6A-D16P4CEE	FC6A-D32K3CEE、 FC6A-D32K4CEE FC6A-D32P3CEE、 FC6A-D32P4CEE
通信端口	以太网端口 1	
基准规格	ANSI/ASHRAE135-2012	
基本规格	协议	BACnet/IP
	轮廓	B-ASC
	对象类型	Device Object、 Analog Input Object、 Analog Output Object、 Analog Value Object、 Binary Input Object、 Binary Output Object、 Binary Value Object
	对象数	最多 256 个 *1
	BIBBs	DS-RP-B、 DS-WP-B、 DS-RPM-B、 DS-WPM-B、 DS-COV-B、 DS-COVU-B、 DM-DDB-B、 DM-DOB-B、 DM-DCC-B
	BBMD	None-BBMD Device
	Virtual Device	No
	Foreign Device	Yes
Subscribed COV 功能	可受理请求数目	最多 256 项请求
Unsubscribed COV 功能	发送单位	每个对象
	发送周期	1 ~ 65,535 [s]*2
外部设备功能	注册方法	基于注册触发设备的实时注册
	Lifetime	0 ~ 65,535 [s]
设备联动功能	<ul style="list-style-type: none"> • 属性、设备间的同步 *3 • Present_Value 的数据型转换 *4 • Present_Value 的系数转换 *4 	

*1 不包含 Device Object。

*2 批量设置所有对象的发送周期。

*3 进行内部内存中生成的对象属性与指定设备的同步。

*4 支持的对象为 Analog Input Object、 Analog Output Object、 Analog Value Object。

BACnet

基准规格

Plus CPU 模块的 BACnet/IP 适用于下列 BACnet 规格。

- ANSI/ASHRAE135-2012

轮廓

Plus CPU 模块的 BACnet/IP 支持下列轮廓。

- B-ASC

对象

由支持 BACnet/IP 的设备处理的输入输出值等信息，将被以对象为单位进行管理。根据对象的内容，对象会被分为数个种类，被称为不同的对象类型。Plus CPU 模块的 BACnet/IP 所支持的对象类型如下所示。

对象类型			规格	
名称	简称	标识符	ANSI/ASHRAE135-2012	
基本输入输出	Analog Input Object Type	AI	0	是
	Analog Output Object Type	AO	1	是
	Analog Value Object Type	AV	2	是
	Binary Input Object Type	BI	3	是
	Binary Output Object Type	BO	4	是
	Binary Value Object Type	BV	5	是
BACnet 设备的特性	Device Object Type	DV	8	是

Plus CPU 模块可根据不同的对象类型，分别设置为任意个数的对象，并从同一 BACnet/IP 网络的 BACnet 设备中读写相应信息。

有关各对象的详情，请参见第 15-33 页上的“对象”。

属性

属性，指的是各对象的详细信息及属性。对象所具备的部分属性，可以被分配到数据寄存器等 Plus CPU 模块设备中，并从梯形图程序进行读取、写入。

有关各对象类型的属性一览，请参见第 15-33 页上的“对象”。

服务

服务，就是用于进行 BACnet 设备间信息交换的接口。根据类型可分为发行服务的客户端，以及执行服务的服务器。

Plus CPU 模块支持的服务如下所示。

服务	发行 ^{*1}	执行 ^{*2}
ReadProperty	—	是
ReadPropertyMultiple	—	是
WriteProperty	—	是
WritePropertyMultiple	—	是
SubscribeCOV	—	是
ConfirmedCOVNotification	是	—
UnconfirmedCOVNotification	是	—
Who-Is	—	是
I-Am	是	—
Who-Has	—	是
I-Have	是	—
DeviceCommunicationControl	—	是

*1 Plus CPU 模块向其他 BACnet 设备发行服务。

*2 Plus CPU 模块会执行其他 BACnet 设备发行的服务。

注释：有关服务的详情，请参见规格书 ANSI/ASHRAE 135-2012（ISSN 1041-2336），或一般社团法人 电气设备学会发行的书籍《BACnet 楼宇自动化用数据通信协议》。

BIBB

BACnet 互操作基本块（BIBB），是将各项实际功能，转化为由多项服务构成的组块。BIBB 可分为使用功能的客户端，以及提供功能的服务器。客户端及服务器的 BIBB 末尾，分别附带有“-A”和“-B”。BACnet 设备利用 BIBB，定义自身支持的功能。

Plus CPU 模块的 BACnet/IP 支持下列 BIBB。

BIBB 分类	BIBB		服务
Data Sharing	DS-RP-B	Data Sharing Read Property B	ReadProperty
	DS-WP-B	Data Sharing Write Property B	ReadPropertyMultiple
	DS-RPM-B	Data Sharing Read Property Multiple B	WriteProperty
	DS-WPM-B	Data Sharing Write Property Multiple B	WritePropertyMultiple
	DS-COV-B	Data Sharing COV B	SubscribeCOV ConfirmedCOVNotification UnconfirmedCOVNotification
	DS-COVU-B	Data Sharing COV Unsubscribed B	UnconfirmedCOVNotification
Device & Network Management	DM-DDB-B	Device Management Dynamic Device Binding B (Who-Is, I-Am)	Who-Is I-Am
	DM-DOB-B	Device Management Dynamic Object Binding B (Who-Has, I-Have)	Who-Has I-Have
	DM-DCC-B	Device Management Device Communication Control B	DeviceCommunicationControl

注释：有关 BIBB 的详情，请参见书籍《BACnet 楼宇自动化用数据通信协议》。

功能

Plus CPU 模块作为 BACnet 设备单机，可提供下列功能。

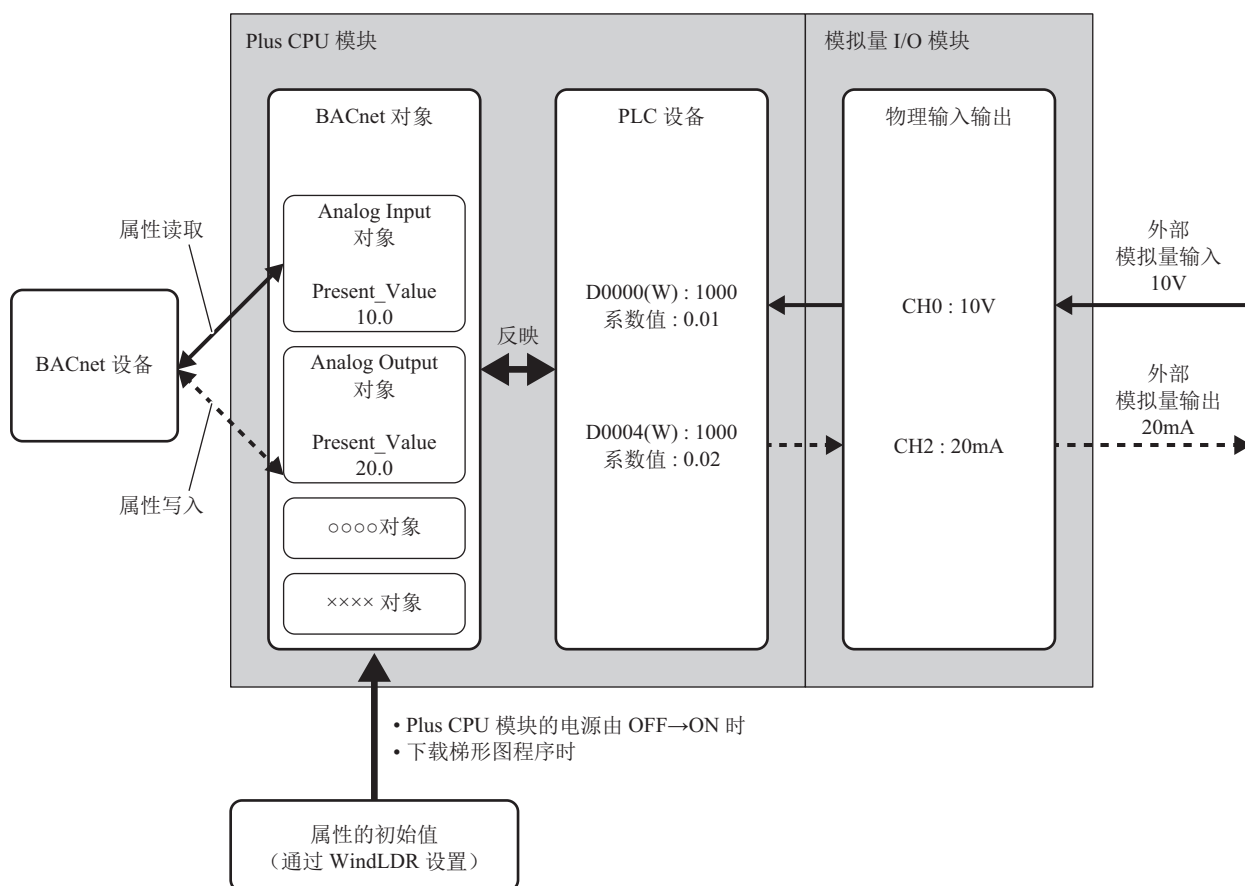
- 对象与设备的联动功能
- 属性的读取功能
- 属性的写入功能
- Subscribed COV 功能
- Unsubscribed COV 功能
- 外部设备功能

对象与设备的联动功能

Plus CPU 模块内部生成对象的部分属性，可以被分配到数据寄存器等设备中，并从梯形图程序进行读取、写入。

例如，通过将 Plus CPU 模块的模拟量输入值，存储到作为 Analog Input 对象 Present_Value 分配的数据寄存器（D0000）中，可以使 BACnet 设备读取相应的模拟量输入值。此外，通过将存储有 Plus CPU 模块模拟量输出值的数据寄存器（D0004），作为 Analog Output 对象的 Present_Value 进行分配，可以从 BACnet 设备更改相应的模拟量输出值。

对象的种类及个数可设置为任意值。通过 WindLDR 设置属性的初始值。



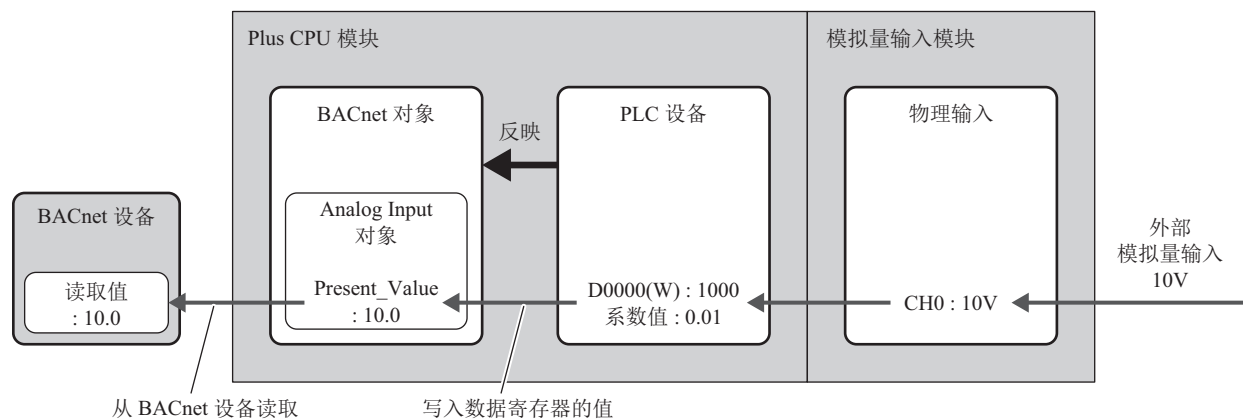
注释:

- 相互反映属性与设备的处理，与梯形图程序的执行周期无关。梯形图程序的执行过程中，被分配为对象的设备将执行读取及写入，请正确创建梯形图程序，确保设备能够正常执行参照及刷新。
 - 设备以比反映到对象属性中的周期还短的间隔变化时，该变化可能无法反映到属性中。要将该变化反映到属性中时，请将设备保持 1 秒钟。
- 例如，分配到 Binary input 对象的 Present_Value 中的 M0000 仅打开 10ms 时，可能无法反映到属性中。要将该变化反映到属性中时，请将 M0000 打开 1 秒钟。

属性的读取功能

属性的读取功能，指的是当 BACnet 设备发出属性读取请求时，由 Plus CPU 模块反馈属性值的功能。通过使用 DS-RP-B（Data Sharing Read Property B）、DS-RPM-B（Data Sharing Read Property Multiple B），可实现本功能。

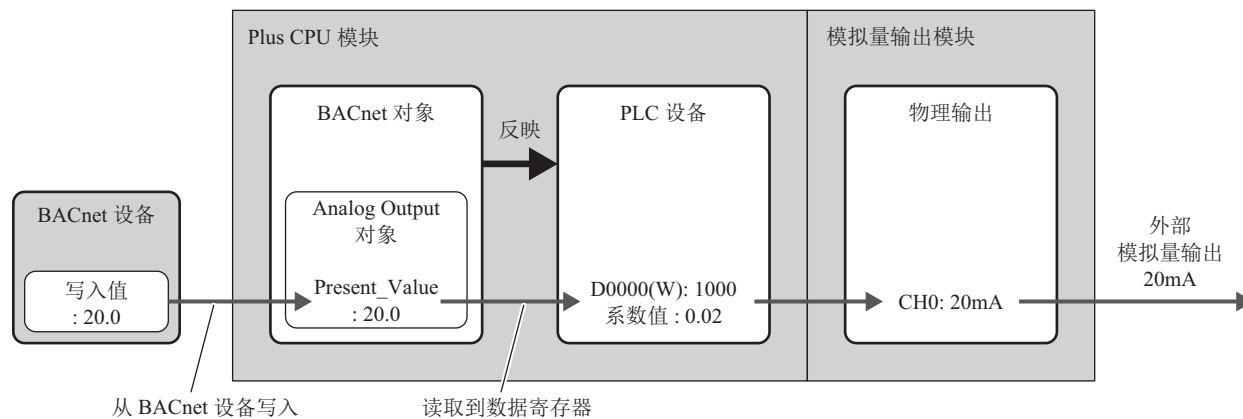
下图所示的是 BACnet 设备读取 Analog Input 对象所对应的 Plus CPU 模块模拟量输入值的过程。读取 Plus CPU 模块模拟量输入值的示意图。



属性的写入功能

属性的写入功能，指的是当 BACnet 设备发出属性写入请求时，由 Plus CPU 模块将值写入对象属性的功能。通过使用 DS-WP-B（Data Shring Write Property B）、DS-WPM-B（Data Sharing Write Property Multiple B），可实现本功能。

下图所示的是 BACnet 设备更改 Analog Output 对象所对应的 Plus CPU 模块模拟量输出值的过程。



Subscribed COV (COV) 功能

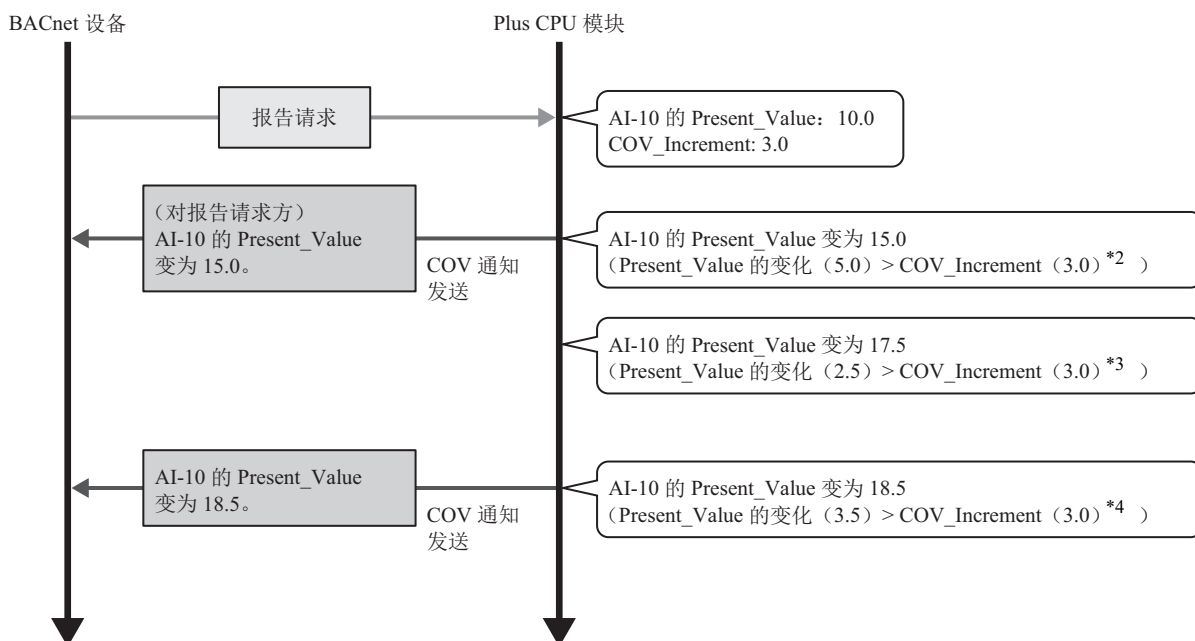
COV (Change Of Value) 功能, 指的是由 Plus CPU 模块监视从 BACnet 设备接收到报告请求的对象, 并在 Present_Value 或 Status_Flags 值发生变化时, 将变化通知给 BACnet 设备的功能。

COV 功能支持的对象属性及变化通知时间如下所示。

对象	属性	变化通知时间
Analog Input Analog Output Analog Value	Present_Value Status_Flags	在下列 (1) 或 (2) 时, 通知变化。 (1) Present_Value 变化为超出 COV_Increment 设置值的值时 (以上一次发送 COV 通知的时间为起点) *1 (2) Status_Flags 之一的位发生变化时
Binary Input Binary Output Binary Value	Present_Value Status_Flags	在下列 (1) 或 (2) 时, 通知变化。 (1) Present_Value 发生变化时 (2) Status_Flags 之一的位发生变化时

*1 下图为 AI-10 (Analog Input 对象, 实例编号 10) 在下表状态下接收到 BACnet 设备发出的报告请求时的动作示意图。

AI-10 的属性	值
Present_Value	10.0
COV_Increment	3.0



*2 该变化相当于通知变化的时间 (1)。

Present_Value 的变化 = 当前值 - 接收到报告请求时的值 = $15.0 - 10.0 = 5.0$
由于 $COV_Increment = 3.0$, 满足 Present_Value 的变化 $\geq COV_Increment$ 。

*3 Present_Value 的变化 = 当前值 - 上次发送 COV 通知时的值 = $17.5 - 15.0 = 2.5$

Present_Value 的变化 $< COV_Increment$, 不满足通知变化的时间 (1) 的条件。
因此, 不发送 COV 通知。

*4 该变化相当于通知变化的时间 (1)。

Present_Value 的变化 = 当前值 - 上次发送 COV 通知时的值 = $18.5 - 15.0 = 3.5$
由于 $COV_Increment = 3.0$, 满足 Present_Value 的变化 $\geq COV_Increment$ 。

注释:

- 若值的变化速度快于对象与设备的同步周期, 可能出现无法通知的情况。
- 发行 BACnet 服务的发送队列为 30 个。同时存在超出发送队列上限的发送请求时, 可能无法发送。

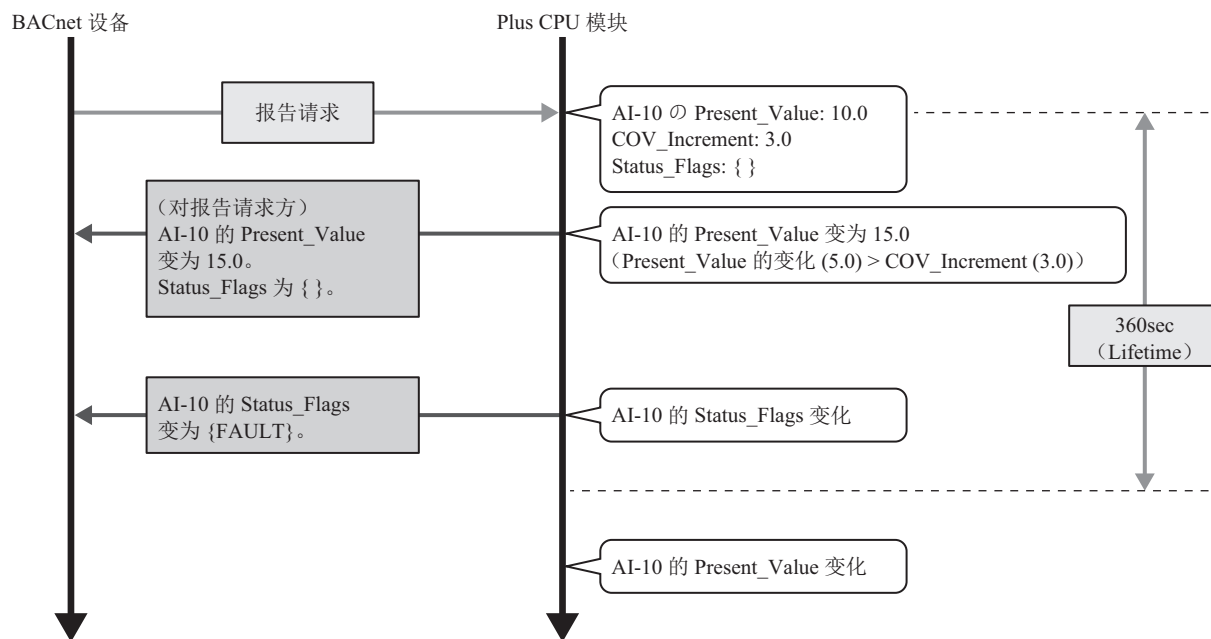
Plus CPU 模块在接收到 BACnet 设备发送的报告请求（Subscribe COV 服务）后，将根据报告请求中包含的参数，通过向报告请求方的 BACnet 设备发送 Confirmed COV Notification 服务 / Unconfirmed COV Notification 服务，来实现 COV 功能。

报告请求中包含的主要参数如下所示。

主要参数	说明
Monitored Object Identifier	启用 COV 功能的对象种类及 ID。
Issue Confirmed Notifications	选择是 / 否确认由 Plus CPU 模块向 BACnet 设备发送的信息。 • 确认（ConfirmedCOVNotification） • 不确认（UnconfirmedCOVNotification）
Lifetime	启用 COV 功能的时间（1sec 单位）。 设定为 0 或省略时，将始终启用 COV 功能。

接收到 BACnet 设备发送的下列报告请求时，Plus CPU 模块将做出如下动作。

报告请求的主要参数	说明
Monitored Object Identifier	Analog Input 对象（实例编号：10）
Issue Confirmed Notifications	确认（ConfirmedCOVNotification）
Lifetime	360sec



注释：

- 最多可注册 256 个 COV。
- 有关报告请求的各参数的详情，请参见书籍《BACnet 楼宇自动化用数据通信协议》。

Unsubscribed COV (COVU) 功能

COVU (Change Of Value Unsubscribed) 功能指的是, 当特定对象的 Present_Value 或 Status_Flags 发生变化时, Plus CPU 模块主动向连接在同一网络上的所有 BACnet 设备发出通知的功能。

COVU 功能支持的对象属性及变化通知时间如下所示。

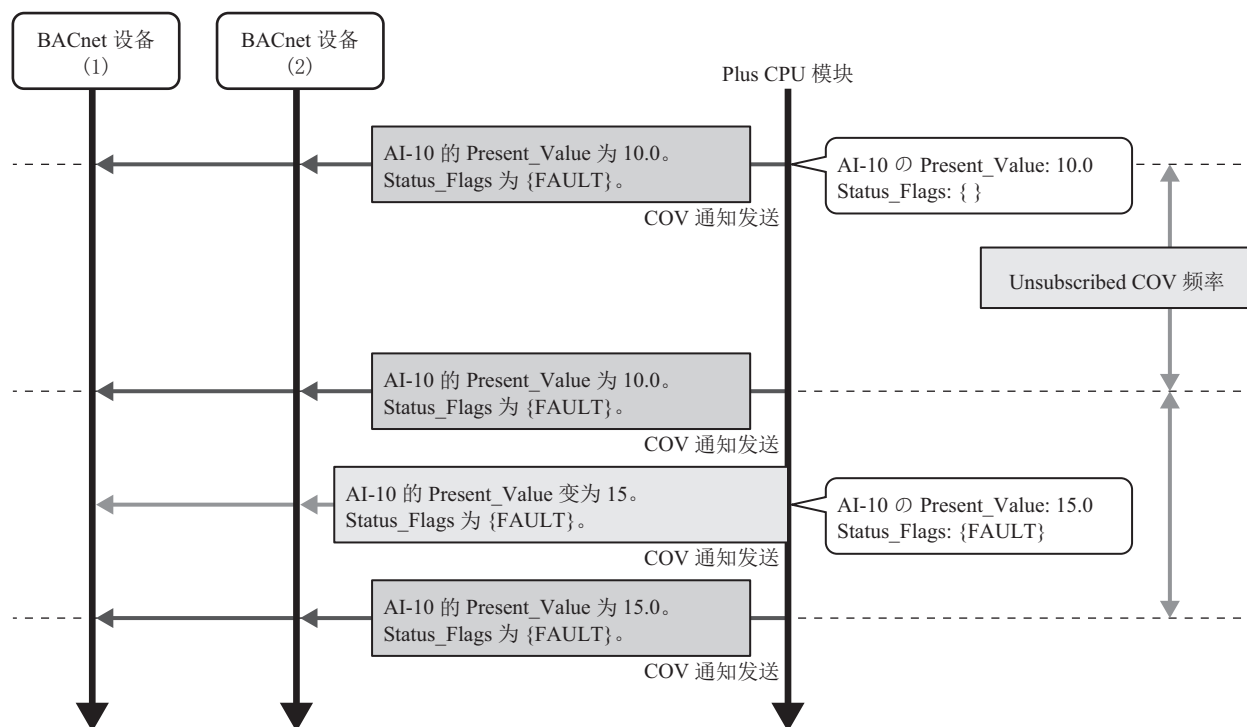
对象	属性	变化通知时间
Analog Input Analog Output Analog Value	Present_Value Status_Flags	以 Unsubscribed COV 频率 *1 为间隔, 通知状态。 或在下列 (1) 或 (2) 时, 通知变化。 (1) Present_Value 变化为超出 COV_Increment 设置值的值时 (以上一次发送 COV 通知的时间为起点) (2) Status_Flags 之一的位发生变化时
Binary Input Binary Output Binary Value	Present_Value Status_Flags	在下列 (1) 或 (2) 时, 通知变化。 (1) Present_Value 发生变化时 (2) Status_Flags 之一的位发生变化时

*1 请参见第 15-14 页上的“基本设置”。

Plus CPU 模块通过向 BACnet 设备发送 Unconfirmed COV Notification 服务的形式, 实现 COVU 功能。

例如, 当 AI-10 (Analog Input 对象、实例编号 10) 在下表状态下开始 BACnet/IP 时, 将按照预设的周期 (Unsubscribed COV 频率), 通知属性状态。

AI-10 的属性	值
Present_Value	10.0
COV_Increment	3.0
Status_Flags	{ }

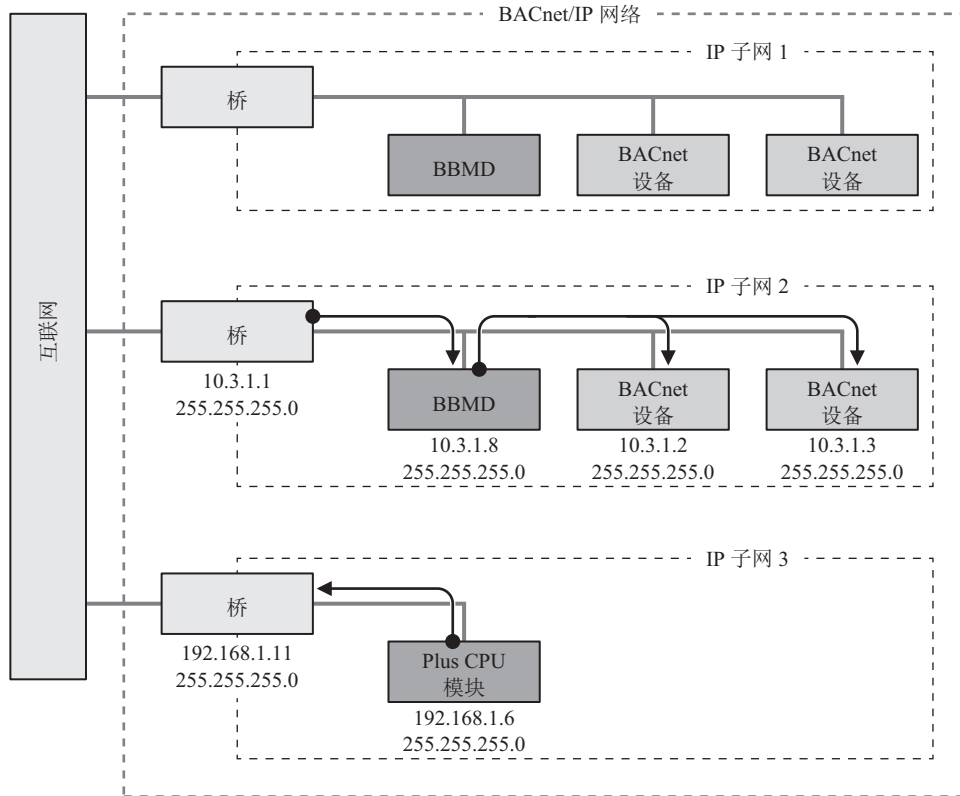


注释:

- 使用 COVU 功能时, 无论属性值的变化情况如何, 都可定期向 BACnet 设备发送属性值。
- COVU 功能可分对象设置。
- 可设置 1 个统一的 Unsubscribed COV 频率。不能分对象设置不同的周期。
- 将 Unsubscribed COV 频率设置为 0 时, 将不会定期通知。仅在对象属性发生变化时通知。
- 通过设备更改 Unsubscribed COV 频率后, 将在下一次发行后生效。
- 若值的变化速度快于对象与设备的同步周期, 可能出现无法通知的情况。
- 发行 BACnet 服务的发送队列为 30 个。同时存在超出发送队列上限的发送请求时, 可能无法发送。

外部设备功能

由多个 IP 子网构成 BACnet/IP 网络时，各 IP 子网将设置 1 台 BBMD（BACnet Broadcast Management Device）。BBMD 是用于将 BACnet 设备的广播通信发送到不同 IP 子网的装置。BACnet 设备经由 BBMD，与不同 IP 子网中的 BACnet 设备进行广播通信。外部设备功能指的是，即使 Plus CPU 模块的 IP 子网中无 BBMD，也能与不同 IP 子网的 BACnet 设备进行广播通信的功能。通过将 Plus CPU 模块作为外部设备，注册到指定的 BBMD，能够与 BACnet/IP 网络内的 BACnet 设备进行广播通信。



BACnet/IP 的动作

BACnet/IP 使用的特殊设备

BACnet/IP 使用的特殊数据寄存器及特殊内部继电器如下所示。

特殊数据寄存器	说明	
D8782	BACnet 运行状态	存储 BACnet 通信的运行状态。 0: 停止中 1: 运行准备中 2: 运行中 3: 错误停止中
D8783	BACnet 错误信息	存储 BACnet 通信中发生的错误信息。 存储最近发生的错误信息。 0: 正常 1: 设备 ID 不正确 2: IP 地址不正确 3: BBMD 的 IP 地址不正确 4: BBMD 注册失败

特殊内部继电器	说明	
M8450	BACnet 通信许可	启用 / 禁用 BACnet 通信。 OFF: 禁用 BACnet 通信 ON: 启用 BACnet 通信

注释: 若需在 BACnet 运行状态 (D8782) 为错误停止中的状态下重新开始 BACnet 通信, 请暂时将 M8450 切换到 OFF, 等待 BACnet 运行状态 (D8782) 变为停止中后, 再将 M8450 切换到 ON。

RUN 及 STOP 中的动作

BACnet/IP 的启用 / 禁用与 RUN/STOP 的状态无关, 将依据 BACnet 通信许可 (M8450) 进行切换。

分配设备的属性, 将参照相应设备执行动作。

启用 BACnet/IP 时, 若 STOP 中也有属性或与属性相关联的设备值被更改, 将按照更改后的值执行动作。

RUN/STOP 状态	BACnet/IP 启用 / 禁用	分配设备的属性及相应设备的值
STOP	启用	联动
	禁用	不联动

注释: STOP 中的输出与 BACnet/IP 的启用 / 禁用无关, 一律根据 M8025 (停止时维持输出) 的状态执行输出。有关 M8025 (停止时维持输出) 的详情, 请参见第 2-12 页上的“M8025: 停止时维持输出”。

STOP → RUN 时的动作

分配设备的属性，将参照相应设备执行动作。

若 Plus CPU 模块的电源 OFF/ON 及 STOP/RUN 切换、梯形图程序设备值的更改等中，有与属性相关联的设备值被更改，将按照更改后的值执行动作。因此，可能会出现超出预期的动作，敬请注意。

例如，在将 M0000 分配到 Out_Of_Service 时，根据设备的内存备份设置，将出现如下所示的 Out_Of_Service 值差异。

(1) STOP → RUN 将内部继电器“全部清除”时

RUN/STOP 状态	BACnet/IP 启用 / 禁用	Out_Of_Service	M000
STOP	启用	TRUE	ON
↓			
RUN	启用	FALSE	OFF

由于 RUN 开始时，内部继电器的状态被清除，将发生 Out_Of_Service= FALSE。

(2) STOP → RUN 将内部继电器“全部保持”时

RUN/STOP 状态	BACnet/IP 启用 / 禁用	Out_Of_Service	M000
STOP	启用	TRUE	ON
↓			
RUN	启用	TRUE	ON

由于内部继电器的状态被保持，保持 Out_Of_Service= TRUE 不变。

注释:

- 请进行设置，确保分配到属性的设备由 STOP → RUN，并保持相应状态。
- 内部继电器及数据寄存器的内存备份默认设置如下所示。

设备	默认设置
内部继电器	全部清除
数据寄存器	全部保持

有关设备内存备份的设置，请参见 FC6A 型 MICROSmart 用户手册“第 5 章 功能和设置”。

BACnet/IP 的设置

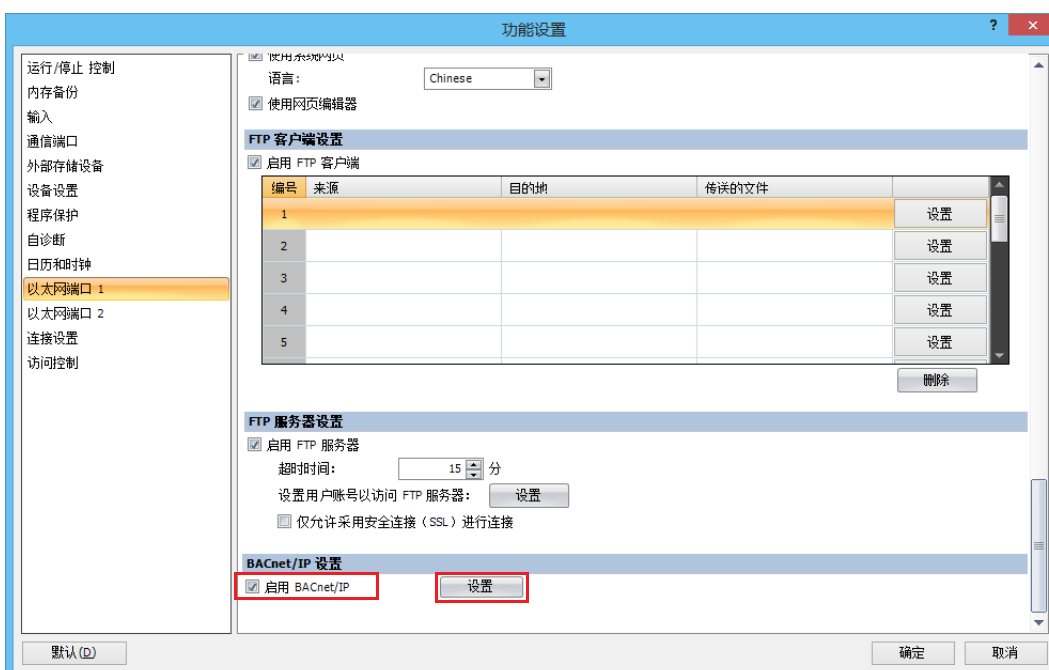
本节将对使用 BACnet/IP 通信的操作过程及各项目的详情进行介绍。

●操作过程

1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。
出现“功能设置”对话框。



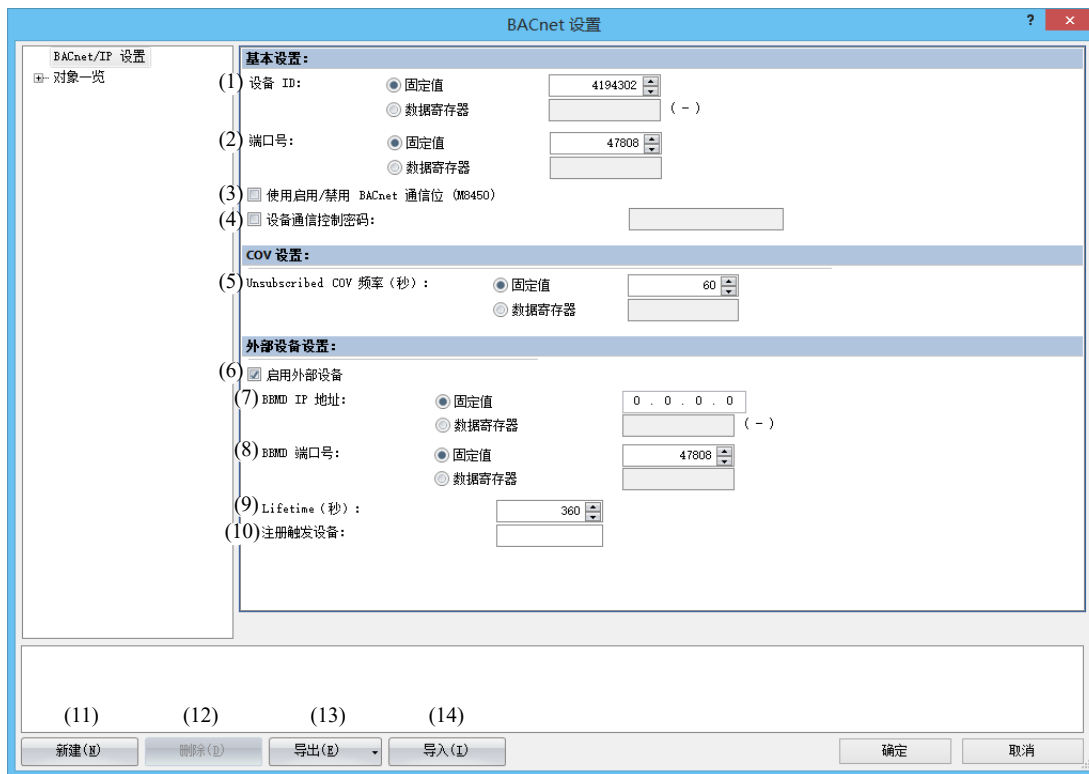
2. 在 BACnet/IP 设置中选中“启用 BACnet/IP”复选框，然后单击“设置”按钮。
将显示“BACnet 设置”对话框。



基本设置

本节将对执行 BACnet/IP 的基本设置进行介绍。

下图是在“BACnet 设置”对话框中选择“BACnet/IP 设置”时的示例。



基本设置

(1) 设备 ID

在 BACnet/IP 网络中，为了识别各 BACnet 设备而分配给设备的 ID。
Plus CPU 模块的设备 ID 可通过如下 2 种方式进行设置。

设置方法	说明
固定值	将设备 ID 设置为固定值。 在 1 ~ 4194302 的范围内设置固定值。默认值为 4194302。
数据寄存器	用数据寄存器设置设备 ID。 从指定的数据寄存器开始，使用连续的 2 字数据寄存器。请指定设备范围内的第一数据寄存器。 在 1 ~ 4194302 的范围内设置设备 ID。

(2) 端口号

设置执行 BACnet/IP 的端口编号。
端口编号可通过如下 2 种方式进行设置。

设置方法	说明
固定值	将端口编号设置为固定值。 在 0 ~ 65535 的范围内设置固定值。默认值为 47808 (BAC0h)。
数据寄存器	通过数据寄存器设置端口编号。 在 0 ~ 65535 的范围内设置端口编号。

(3) 使用启用 / 禁用 BACnet 通信位 (M8450)

运行 (RUN) 开始时, 在第 1 次扫描的 END 处理中, 设置是否将 BACnet 通信许可 (M8450) 切换到 ON。联机编辑中, 在完成用户程序下载后最初的扫描 END 处理中, BACnet 通信许可 (M8450) 变为 ON。

各状态如下表所示。

M8450 的状态	RUN/STOP 状态	BACnet/IP	Presnet_Value*1 与设备的值	Status_Flags*2 的 Overridden 标记
ON	STOP	启用	联动	TRUE
	RUN			FALSE
OFF	STOP	禁用	不联动	—
	RUN			—

*1 有关 Present_Value, 请参见第 15-33 页上的“对象”、第 15-41 页上的“Present_Value”。

*2 有关 Status_Flags, 请参见第 15-33 页上的“对象”、第 15-54 页上的“Status_Flags”。

注释:

BACnet 通信许可 (M8450) 在下列情况下变为 OFF。

- 无论是在 STOP 中还是 RUN 中, 均在下载用户程序之后
- 关闭 Plus CPU 模块的电源时

(4) 设备通信控制密码

设置从 BACnet 设备接收设备通信控制服务时执行请求的密码。最多可设置 20 个半角英文数字。Plus CPU 模块在接收到设备通信控制服务后, 在指定时间内不会执行服务的发行及响应。

COV 设置**(5) Unsubscribed COV 频率 (秒)**

以 1 sec 为单位, 设置 COVU 功能中定期通知属性值的周期。

可设置范围为 0 ~ 65535。设置为 0 时, COVU 功能停止, 仅会在对象属性发生变化时通知。

设置方法	说明
固定值	将周期设置为固定值。默认值为 60sec。
数据寄存器	用数据寄存器设置周期。 RUN 中可更改周期。

注释:

- 可在 Plus CPU 模块中设置 1 个统一的 Unsubscribed COV 频率, 但不能分对象设置不同频率。
- 可为不同的对象分别选择是否使用 COVU 功能。还能指定数据寄存器, 通过更改相应的值, 切换 COVU 功能的启用 / 禁用。
- 对 Unsubscribed COV 频率设置数据寄存器, 通过更改数据寄存器的值来更改 Unsubscribed COV 频率时, 更改后的 Unsubscribed COV 频率将在下次发行后生效。

外部设备设置

(6) 启用外部设备

设置是否使用外部设备功能。

(7) BBMD IP 地址

(8) BBMD 端口号

在 Plus CPU 模块的 IP 子网无 BBMD 的状态下连接到 BACnet/IP 网络时，设置其他 IP 子网中 BBMD IP 地址及端口编号。Plus CPU 模块会被注册到将自身作为外部设备设置的 BBMD 中。

BBMD IP 地址可通过如下 2 种方式进行设置。

设置方法	说明
固定值	将 BBMD IP 地址设置为固定值。
数据寄存器	通过数据寄存器设置 BBMD IP 地址。 从指定的数据寄存器开始，使用连续的 4 字数据寄存器。请指定设备范围内的第一数据寄存器。 例如，将 BBMD IP 地址设置为 192.168.2.5 时，将进行如下写入。 起始数据寄存器 +0 = 192 起始数据寄存器 +1 = 168 起始数据寄存器 +2 = 2 起始数据寄存器 +3 = 5

BBMD 端口编号可通过如下 2 种方式进行设置。

设置方法	说明
固定值	将 BBMD 端口编号设置为固定值。 在 0 ~ 65535 的范围内设置固定值。默认值为 47808 (BAC0h)。
数据寄存器	通过数据寄存器设置 BBMD 端口编号。 在 0 ~ 65535 的范围内设置 BBMD 端口编号。

(9) Lifetime (秒)

以 1sec 为单位，设置将 Plus CPU 模块作为外部设备注册到 BBMD 的时间。

注册后超过 (Lifetime + 30) sec 时，已注册的内容将被从 BBMD 中删除。

可设置范围为 0 ~ 65535。默认值为 360sec。

设置方法	说明
固定值	将 Lifetime 设置为固定值。默认值为 360sec。

(10) 注册触发设备

用于将 Plus CPU 模块作为外部设备，注册到上述 (7) 中设置的 BBMD 的设备。将注册触发设备由 OFF 切换到 ON 时，将把 Plus CPU 模块注册到 BBMD 中。需要连续注册到 BBMD 时，请在经过 (Lifetime + 30) sec 之前，利用注册触发设备再次进行注册。

对注册触发设备设置内部继电器。

对象的添加与删除

(11)“新建”按钮

添加新的对象。

单击新建按钮，将显示“新建对象”对话框。设置对象类型与实例编号。对于实例编号，请在 0 ~ 4194302 的范围内，设置不与同一对象类型的其他对象重复的编号。



(12)“删除”按钮

删除在对象一览中选择的节点以下的对象。

(13)“导出”按钮

以 CSV 格式，导出下述对象的设置内容。

Analog Input Object、Analog Output Object、Analog Value Object、Binary Input Object、Binary Output Object、Binary Value Object

(14)“导入”按钮

导入已导出的 CSV 文件，自动生成对象。

CSV 文件的格式不正确，及对象达到最大数时，将无法执行导入。

新建对象的添加

本节将对添加新对象的步骤进行介绍。

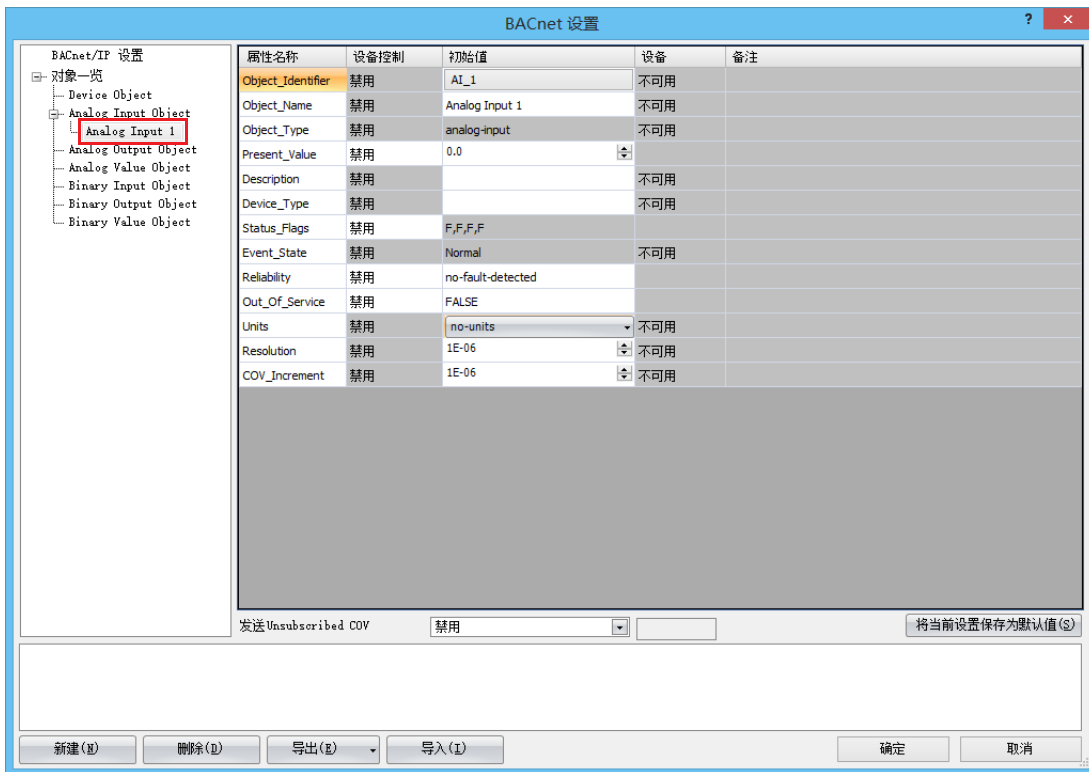
1. 单击“新建”按钮。
将显示“新建对象”对话框。



2. 选择需要注册的对象的对象类型，分配实例编号。
单击“确定”按钮。



- 单击对象一览中已注册的对象 ID。
将显示属性。



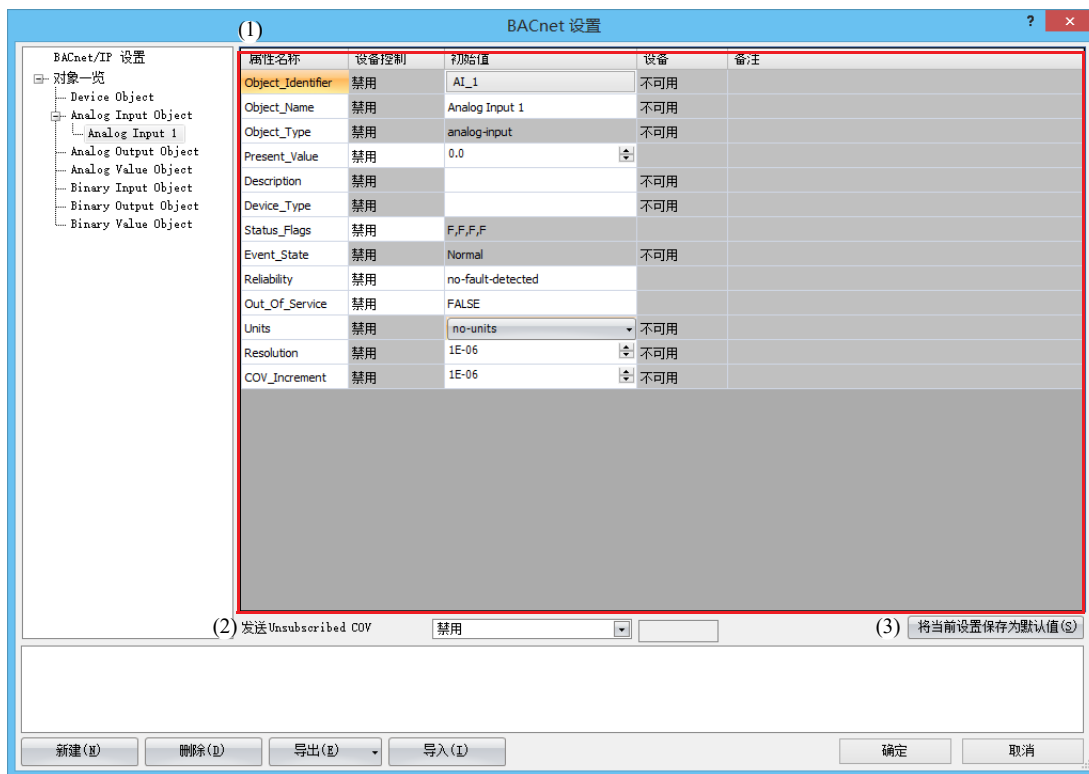
- 设置各属性，然后单击“确定”按钮。
有关详情，请参见第 15-20 页上的“对象设置”。

注释：最多可注册 256 个对象。

对象设置

本节将对对象属性的设置进行介绍。

下图是在“BACnet 设置”对话框中选择已注册对象 ID 时的示例。



(1) 属性设置

显示在对象一览中选择的对象 ID 的属性。
部分属性可编辑。

(2) 发送 Unsubscribed COV

设置发送 / 不发送 Unsubscribed COV。
可通过如下 3 种方式进行设置。

- 启用
- 禁用
- 按设备控制

设置	说明
启用	BACnet 通信许可 (M8450) ON 时, COVU 功能始终启用, 发送 Unconfirmed COV Notification 服务。
禁用	COVU 功能始终禁用, 不发送 Unconfirmed COV Notification 服务。
按设备控制	可以设置内部继电器, 通过切换 ON/OFF, 设置启用 / 禁用。 无法设置特殊内部继电器。

(3) “将当前设置保存为默认值”按钮

将各属性的设置值保存为对象的默认值。

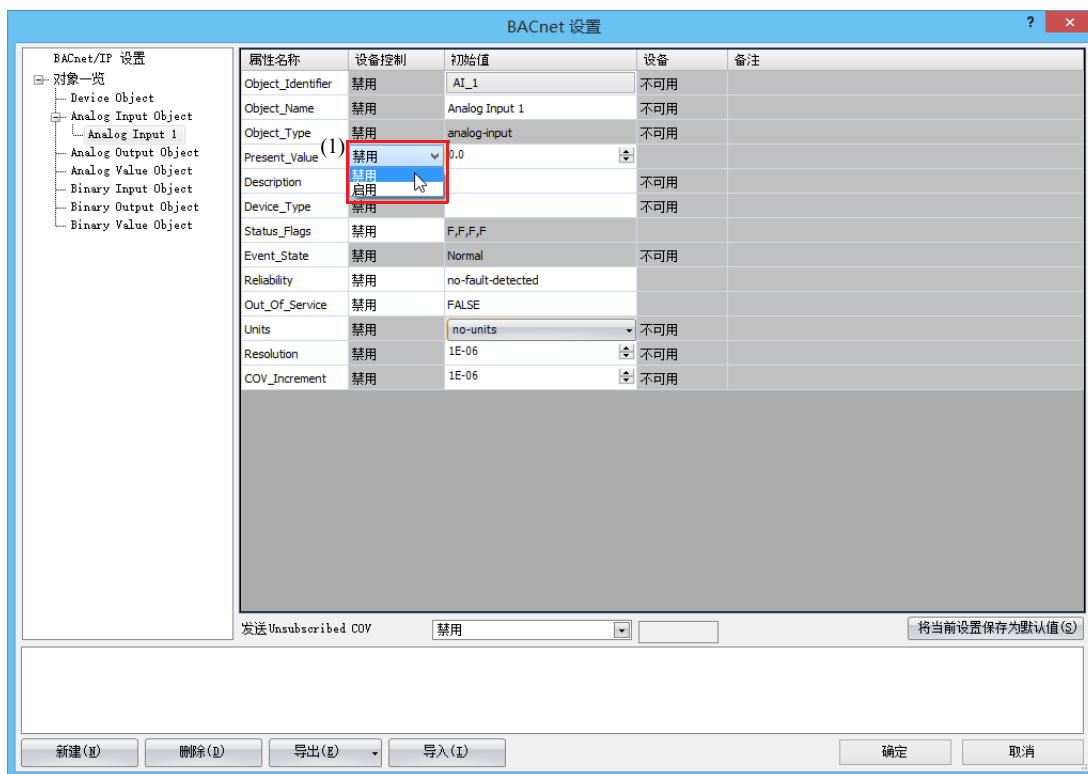
Present_Value 设置

本节将对 Present_Value 的设置方法进行介绍。

Analog Input 对象

将 Analog Input 对象的 Present_Value 设置为固定值或设备。

下图是在对象一览中选择 Analog Input 对象 ID 时的示例。



(1) 设备控制

选择是将 Present_Value 设置为固定值，还是分配数据寄存器。

设备控制	说明
禁用	将 Present_Value 设置为固定值
启用	将数据寄存器分配到 Present_Value

■ 将 Present_Value 设置为固定值时

将 Present_Value 的设备控制 ((1)) 设置为“禁用”，在初始值中输入固定值。



■ 将设备分配到 Present_Value 中时

将 Present_Value 的设备控制 ((1)) 设置为“启用”，在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置存储 Present_Value 的设备。
能够分配到 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

根据“转换类型” ((2)), 从指定的数据寄存器开始, 使用连续的 1 字或 2 字数据寄存器。请指定设备范围内的第一数据寄存器。

Present_Value 设备	存储目的地	
	字 整数	双字 长整 浮点 (无转换)
用于写入的 Present_Value	起始编号 +0	起始编号 +0、起始编号 +1

(2) 转换类型

设置分配 Present_Value (浮点型) 的设备的数据类型。有关详情, 请参见第 15-45 页上的“Analog Input 对象”的 Present_Value。

(3) 系数

数据寄存器中存储的值乘上系数值后, 即为 Present_Value。
Present_Value = 数据寄存器中存储的值 × 系数值

Present_Value 的计算如下表所示。

Plus CPU 模块			系数值	BACnet 设置设备
数据寄存器	数据类型	值		Present_Value
D0000	字	1000	0.01	10.0
D0000、D0001	浮点	2.5	0.5	1.25

注释:

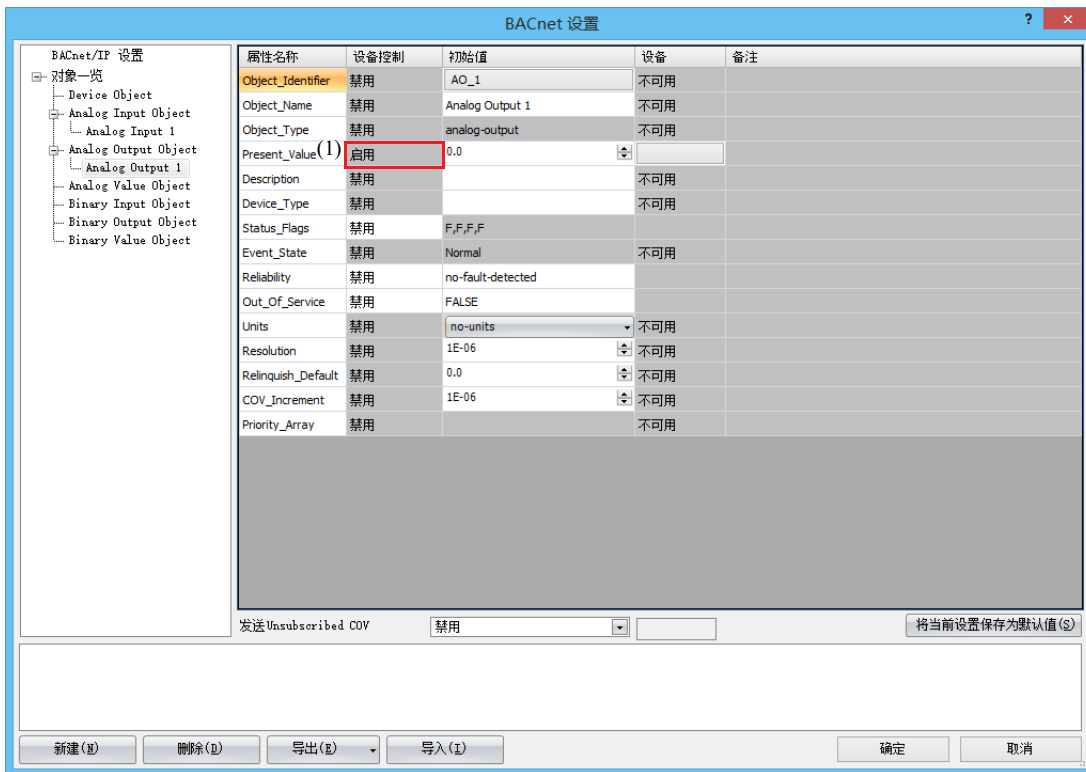
乘上系数值的计算使用浮点型数值。转换顺序如下所示。

- (1) 将数据寄存器的值转换为浮点型。
- (2) 将 (1) 的转换结果乘上系数值。

Analog Output 对象

将设备设置到 Analog Output 对象的 Present_Value。

下图是在对象一览中选择 Analog Output 对象 ID 时的示例。



(1) 设备控制

Present_Value 的设备控制固定为“启用”。

注释： 不能将 Analog Output 对象的 Present_Value 设置为固定值。

■ 将设备分配到 Present_Value 中时

在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置存储 Present_Value 的设备。
能够分配到 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

根据“转换类型” (2)，从指定的数据寄存器开始，使用连续的 1 字或 2 字数据寄存器。请指定设备范围内的第一数据寄存器。

Present_Value 设备	存储目的地	
	字 整数	双字 长整 浮点 (无转换)
用于读取的 Present_Value	起始编号 +0	起始编号 +0、起始编号 +1

(2) 转换类型

设置分配 Present_Value (浮点型) 的设备的数据类型。有关详情，请参见第 15-46 页上的“Analog Output 对象”的 Present_Value。

(3) 系数

Present Value 乘上 1/ 系数值后的值，将被存储到数据寄存器中。
数据寄存器的值 = Present_Value × (1/ 系数值)

例如，设置如下系数值及数据类型时，Present_Value 以如下所示的方式被存储到数据寄存器中。

BACnet 设备 Present_Value	系数值	Plus CPU 模块		
		数据寄存器	数据类型	值
10.0	0.01	D0000	字	1000
1.25	0.5	D0000、D0001	浮点	2.5

注释:

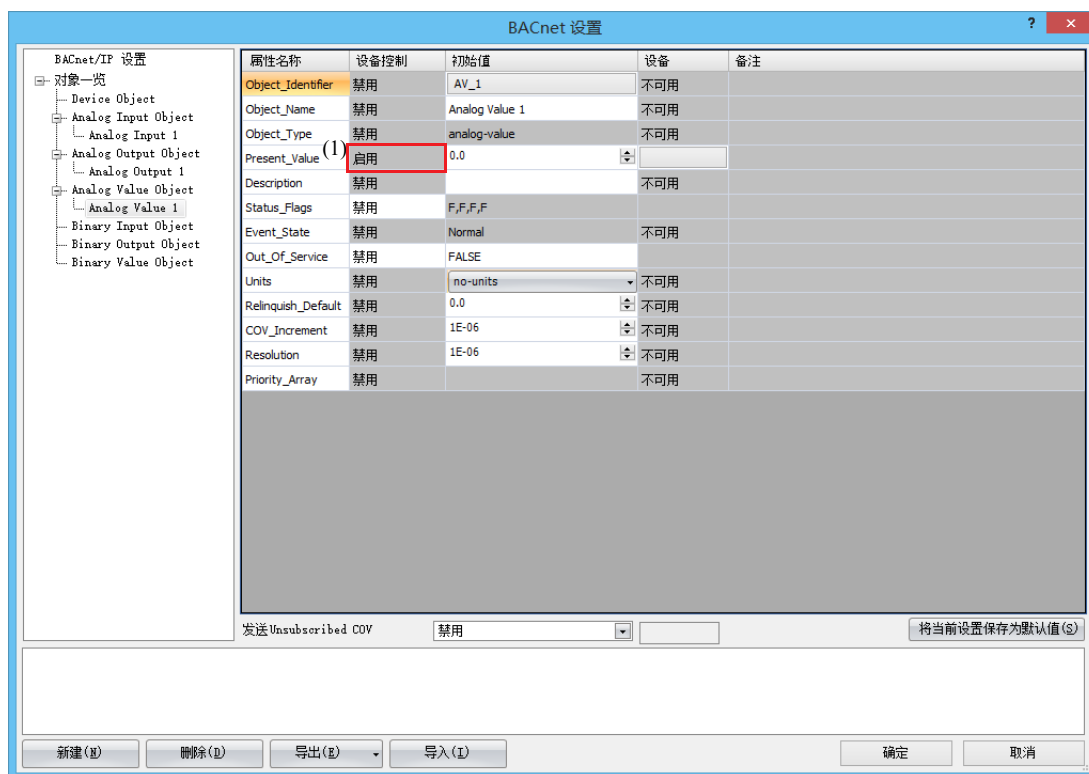
乘上系数值的计算使用浮点型数值。转换顺序如下所示。

- (1) 将 Present_Value 乘上 (1/ 系数值)。
- (2) 对 (1) 的结果 (浮点型) 执行数据类型转换。

Analog Value 对象

将设备设置到 Analog Value 对象的 Present_Value。

下图是在对象一览中选择 Analog Value 对象 ID 时的示例。



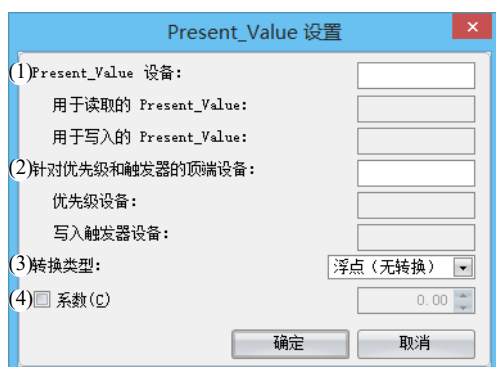
(1) 设备控制

Present_Value 的设备控制固定为“启用”。

注释： 不能将 Analog Value 对象的 Present_Value 设置为固定值。

■ 将设备分配到 Present_Value 中时

在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置 Present_Value 的读取用设备及 Present_Value 的写入用设备。
能够分配到 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

根据已设置的设备及“转换类型”(3)，自动分配用于读取的 Present_Value 和用于写入的 Present_Value 的设备。从指定的数据寄存器开始，使用连续的 2 字或 4 字数据寄存器。请指定设备范围内的第一数据寄存器。

Present_Value 设备	存储目的地	
	字 整数	双字 长整 浮点（无转换）
用于读取的 Present_Value	起始编号 +0	起始编号 +0、起始编号 +1
用于写入的 Present_Value	起始编号 +1	起始编号 +2、起始编号 +3

(2) 针对优先级和触发器的顶端设备

用于将设备的值写入 Present_Value 时。有关详情，请参见第 15-47 页上的“Analog Value 对象”的 Present_Value。

设置数据寄存器后，将自动分配“优先级设备”及“写入触发器设备”。从指定的数据寄存器开始，使用连续的 2 字数据寄存器。请指定设备范围内的第一数据寄存器。

存储目的地	说明	
起始编号 +0	优先级设备	
	位 15	NULL 写入标记*1 OFF: 写入用于写入的 Present_Value 设备的值 ON: 写入 NULL
	位 14 ~ 5	禁用
	位 4 ~ 0	优先*2
起始编号 +1	写入触发器设备	

*1 在 NULL 写入标记 ON 的情况下，将写入触发器设备由 OFF 切换到 ON，NULL 将被写入代表优先级的目录编号 Priority_Array 中。

*2 优先级请指定 1 ~ 16。如果指定了超过范围的优先级，即使操作写入触发器设备仍无法写入。

(3) 转换类型

设置分配 Present_Value（浮点型）的设备的数据类型。有关详情，请参见第 15-47 页上的“Analog Value 对象”的 Present_Value。

(4) 系数

Present_Value 乘上 1/ 系数值后的值，将被作为用于读取的 Present_Value，存储到分配的数据寄存器中。
用于读取的 Present_Value = Present_Value × (1/ 系数值)

此外，写入触发器设备由 OFF 切换到 ON 时，作为用于写入的 Present_Value 分配的数据寄存器值乘上系数值后的值，将被设置为 Present_Value。

Present_Value = 用于写入的 Present_Value × 系数值

Present_Value	系数值	转换类型	作为用于读取 / 写入的 Present_Value 分配的数据寄存器	
			地址	值
10.0	0.01	字	D0000	1000
1.25	0.5	浮点	D0002、D0003	2.5

乘上系数值的计算使用浮点型数值。转换顺序如下所示。

数据寄存器 → Present_Value

- (1) 将数据寄存器的值转换为浮点型。
- (2) 将 (1) 的转换结果乘上系数值。

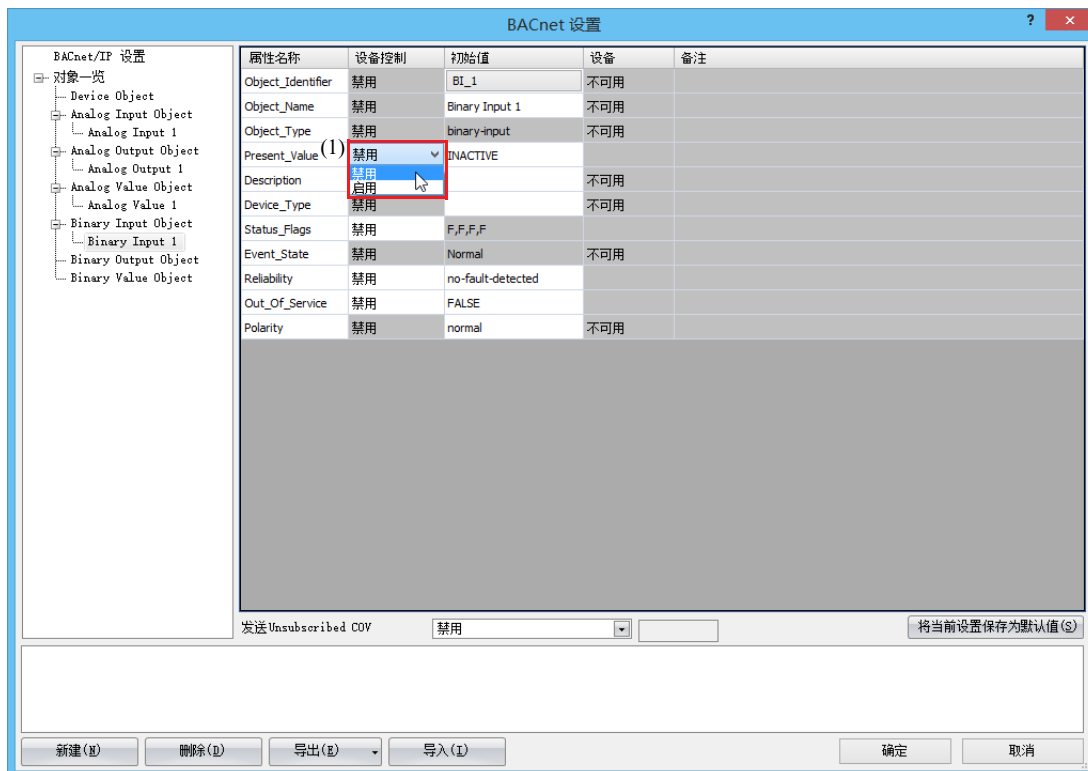
Present_Value → 数据寄存器

- (1) 将 Present_Value 乘上 (1/ 系数值)。
- (2) 对 (1) 的结果 (浮点型) 执行数据类型转换。

Binary Input 对象

将 Binary Input 对象的 Present_Value 设置为固定值或设备。

下图是在对象一览中选择 Binary Input 对象 ID 时的示例。



(1) 设备控制

选择是将 Present_Value 设置为固定值，还是分配数据寄存器。

设备控制	说明
禁用	将 Present_Value 设置为固定值
启用	将数据寄存器分配到 Present_Value

■ 将 Present_Value 设置为固定值时

将 Present_Value 的设备控制 ((1)) 设置为“禁用”，在初始值中设置固定值。



■ 将设备分配到 Present_Value 中时

将 Present_Value 的设备控制 (1) 设置为“启用”，在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置存储 Present_Value 的设备。

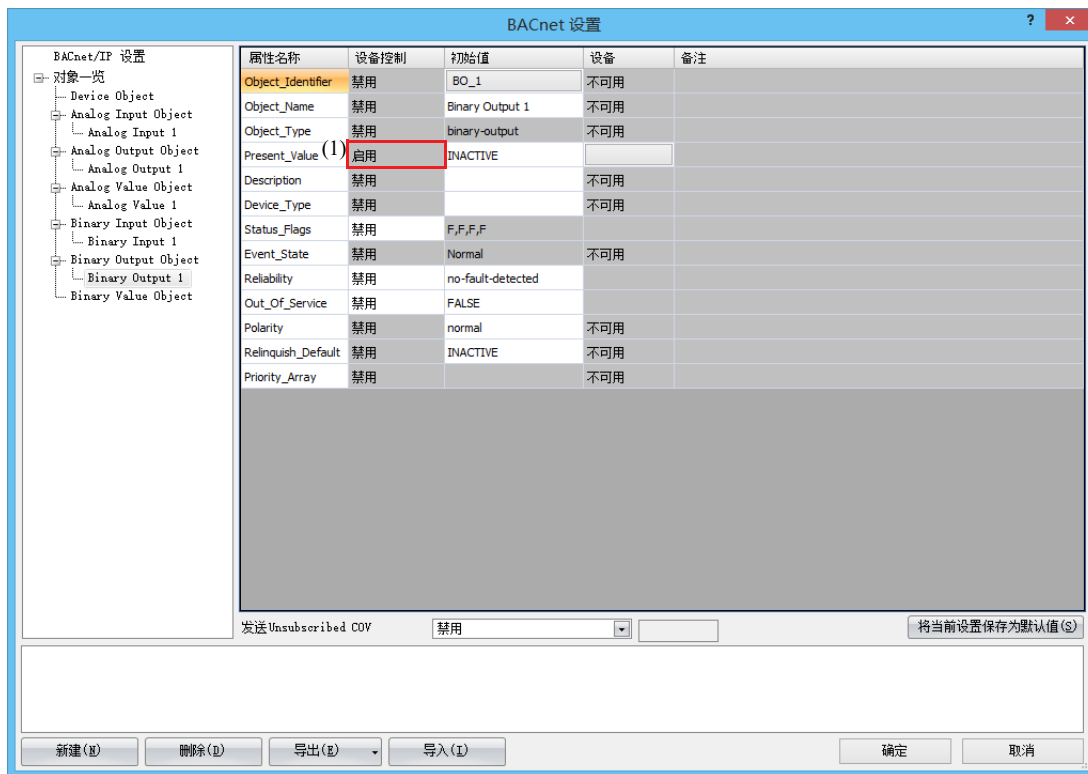
属性	I	Q	M	R	T	C	D	常量
Present_Value	是	—	是*1	—	—	—	—	—

*1无法使用特殊内部继电器。

Binary Output 对象

将设备设置到 Binary Value 对象的 Present_Value。

下图是在对象一览中选择 Binary Output 对象 ID 时的示例。



(1) 设备控制

Present_Value 的设备控制固定为“启用”。

注释： 不能将 Binary Output 对象的 Present_Value 设置为固定值。

■ 将设备分配到 Present_Value 中时

在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置存储 Present_Value 的设备。

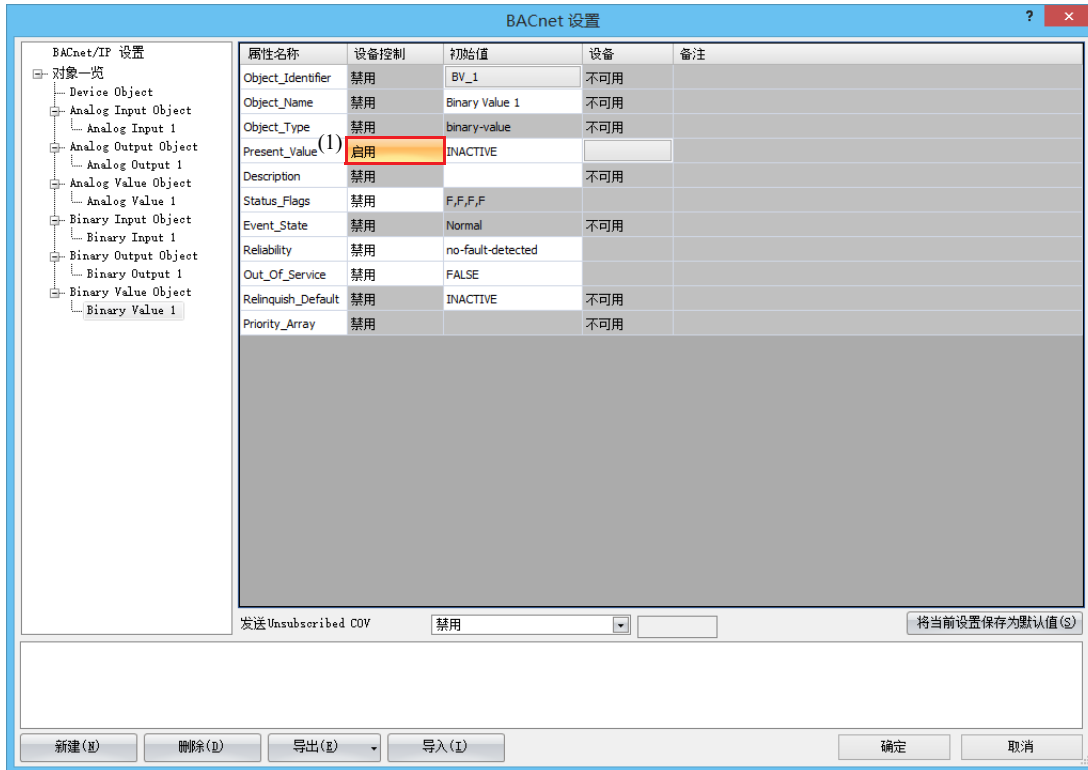
属性	I	Q	M	R	T	C	D	常量
Present_Value	—	是	是*1	—	—	—	—	—

*1无法使用特殊内部继电器。

Binary Value 对象

将设备设置到 Binary Value 对象的 Present_Value。

下图是在对象一览中选择 Binary Value 对象 ID 时的示例。



(1) 设备控制

Present_Value 的设备控制固定为“启用”。

注释: 不能将 Binary Value 对象的 Present_Value 设置为固定值。

■ 将设备分配到 Present_Value 中时

在“Present_Value 设置”对话框中设置各参数。



(1) Present_Value 设备

设置 Present_Value 的读取用设备及 Present_Value 的写入用设备。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	是*1	—	—	—	—	—

*1无法使用特殊内部继电器。

根据已设置的设备，自动分配用于读取的 Present_Value 和用于写入的 Present_Value 的设备。从指定的数据寄存器开始，使用连续的 2 位位设备。请指定设备范围内的首位设备。

Present_Value 设备	存储目的地
用于读取的 Present_Value	起始编号 +0
用于写入的 Present_Value	起始编号 +1

(2) 针对优先级和触发器的顶端设备

用于将设备的值写入 Present_Value 时。有关详情，请参见第 15-52 页上的“Binary Value 对象”的 Present_Value。

设置数据寄存器后，将自动分配“优先级设备”及“写入触发器设备”。从指定的数据寄存器开始，使用连续的 2 字数据寄存器。请指定设备范围内的第一数据寄存器。

存储目的地	说明	
起始编号 +0	优先级设备	
	位 15	NULL 写入标记*1 OFF: 写入用于写入的 Present_Value 设备的值 ON: 写入 NULL
	位 14 ~ 5	禁用
	位 4 ~ 0	优先*2
起始编号 +1	写入触发器设备	

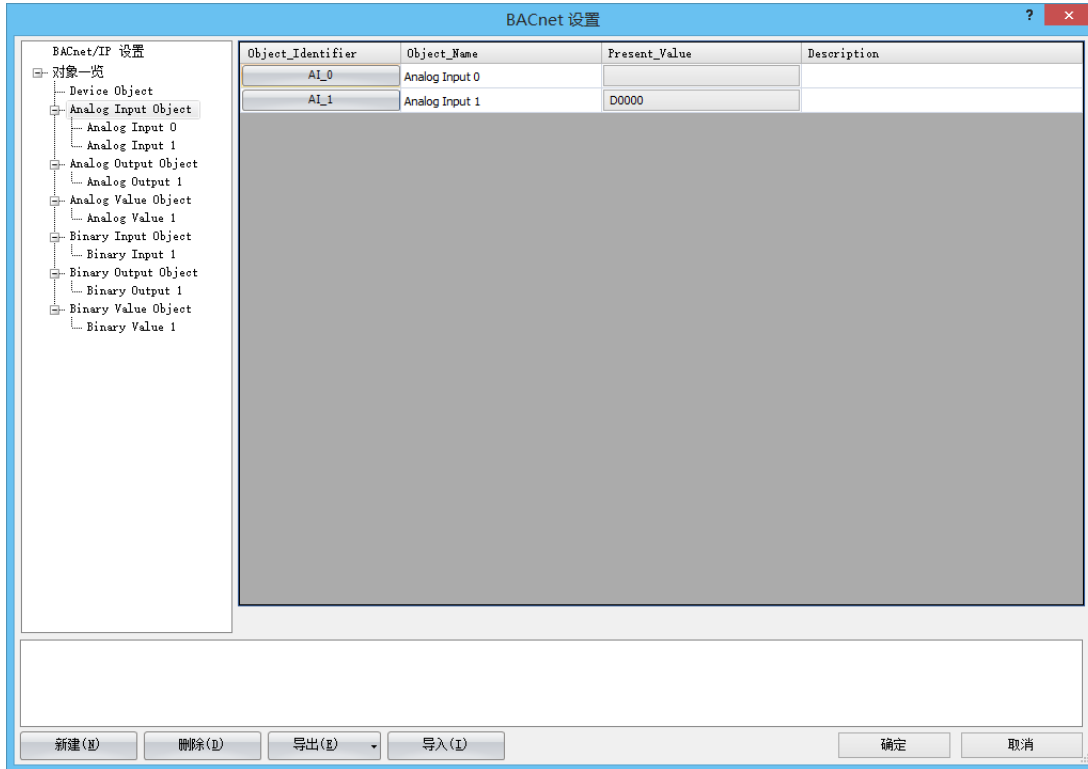
*1 在 NULL 写入标记 ON 的情况下，将写入触发器设备由 OFF 切换到 ON，NULL 将被写入代表优先级的目录编号 Priority_Array 中。

*2 优先级请指定 1 ~ 16。如果指定了超过范围的优先级，即使操作写入触发器设备仍无法写入。

对象一览

选择各对象的节点后，将显示已注册的对象一览。

例如，选择 Analog Input 时，将显示 Analog Input 的对象一览，选择对象一览节点后，将显示所有对象一览。此外，在显示一览时也可以更改各属性。



对象

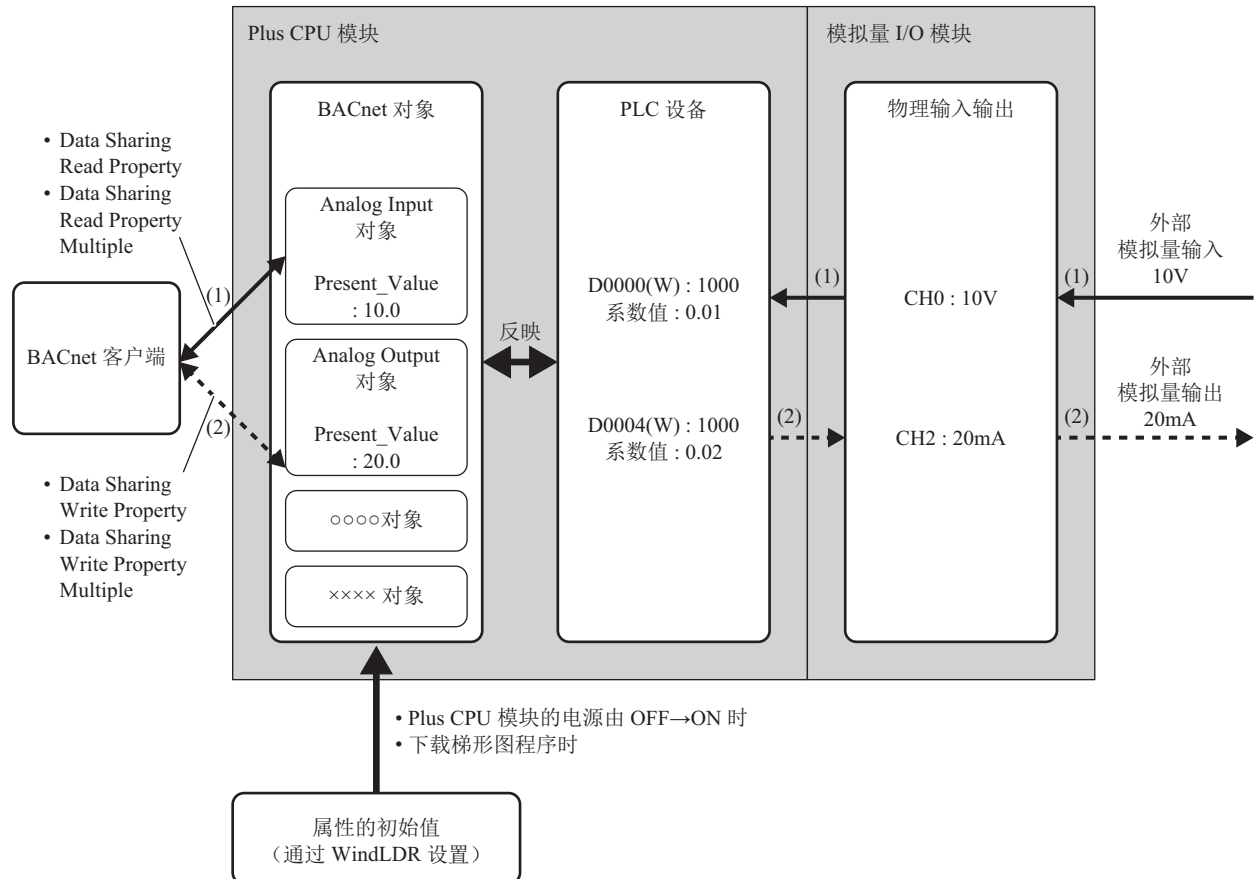
Plus CPU 模块会将 WindLDR 注册的对象，保存在内部内存中。对象所具备的部分属性，可以被分配到数据寄存器中，并从梯形图程序进行读取、写入。BACnet 设备可使用服务，读取、写入 Plus CPU 模块对象的各属性。Plus CPU 模块对象的属性及分配到属性的设备值实时同步。

有关使用 WindLDR 的对象注册方法，请参见第 15-18 页上的“新建对象的添加”。此外，最多可注册 256 个对象。

详情请见下列 2 种示例。

(1) BACnet 设备读取 Analog Input 对象的模拟量输入值

(2) BACnet 设备写入 Analog Output 对象的模拟量输出值



有关对象所具备各属性的详情，请参见书籍《BACnet 楼宇自动化用数据通信协议》。

Analog Input 对象

管理浮点型数值的对象。Plus CPU 模块可以向 BACnet 设备公开任意数值。在处理导入到模拟量输入模块中的模拟量值、测得的室内温度等数据时使用。

属性一览

显示 Analog Input 对象支持的属性。

- (1): 从 BACnet 设备读取 / 写入
- (2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读：仅可读取、写：仅可写入、读 / 写：可读取及写入、-：不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串*1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	实数	读	写	参见第 15-41 页上的“Present_Value”
Description	字符串*1	读	-	通过 WindLDR 设置固定值。
Device_Type	字符串*1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 的值是否可信。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Units	BACnetEngineeringUnits	读 / 写	-	通过 WindLDR 设置初始值。
Resolution	实数	读	-	通过 WindLDR 设置固定值。
COV_Increment	实数	读 / 写	-	参见第 15-54 页上的“COV_Increment”
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8)，最大为 64 个字节。

Analog Output 对象

管理浮点型数值的对象。Plus CPU 模块可从 BACnet 设备接收任意数值。在从 BACnet 设备接收模拟量输出模块输出的模拟量值、作为动作参数的设置温度等时使用。

属性一览

显示 Analog Output 对象支持的属性。

(1): 从 BACnet 设备读取 / 写入

(2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读: 仅可读取、写: 仅可写入、读 / 写: 可读取及写入、-: 不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	实数	读	读	参见第 15-41 页上的“Present_Value”
Description	字符串 *1	读	-	通过 WindLDR 设置固定值。
Device_Type	字符串 *1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 的值是否可信。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Units	BACnetEngineeringUnits	读 / 写	-	通过 WindLDR 设置初始值。
Resolution	实数	读	-	通过 WindLDR 设置固定值。
Priority_Array	BACnetPriority_Array	读 / 写	-	参见第 15-54 页上的“Priority_Array”
Relinquish_Default	实数	读 / 写	-	参见第 15-54 页上的“Relinquish_Default”
COV_Increment	实数	读 / 写	-	参见第 15-54 页上的“COV_Increment”
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8)，最大为 64 个字节。

Analog Value 对象

用法与 Analog Input 对象或 Analog Output 对象相同。

属性一览

显示 Analog Value 对象支持的属性。

- (1): 从 BACnet 设备读取 / 写入
- (2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读: 仅可读取、写: 仅可写入、读 / 写: 可读取及写入、-: 不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	实数	读	读	参见第 15-41 页上的“Present_Value”
Description	字符串 *1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 是否为可信值。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Units	BACnetEngineeringUnits	读 / 写	-	通过 WindLDR 设置初始值。
Priority_Array	BACnetPriority_Array	读 / 写	读 / 写	参见第 15-54 页上的“Priority_Array”
Relinquish_Default	实数	读 / 写	-	参见第 15-54 页上的“Relinquish_Default”
COV_Increment	实数	读 / 写	-	参见第 15-54 页上的“COV_Increment”
Resolution	实数	读	-	通过 WindLDR 设置固定值。
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8), 最大为 64 个字节。

Binary Input 对象

管理二进制值（ON/OFF）的对象。在 Plus CPU 模块向 BACnet 设备公开二进制值时使用。

属性一览

显示 Binary Input 对象支持的属性。

(1): 从 BACnet 设备读取 / 写入

(2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读: 仅可读取、写: 仅可写入、读 / 写: 可读取及写入、-: 不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	BACnetBinaryPV	读	写	参见第 15-41 页上的“Present_Value”
Description	字符串 *1	读	-	通过 WindLDR 设置固定值。
Device_Type	字符串 *1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 的值是否可信。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Polarity	BACnetPolarity	读 / 写	-	参见第 15-54 页上的“Polarity”
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8)，最大为 64 个字节。

Binary Output 对象

管理二进制值（ON/OFF）的对象。在 Plus CPU 模块从 BACnet 设备接收二进制值时使用。

属性一览

显示 Binary Output 对象支持的属性。

- (1): 从 BACnet 设备读取 / 写入
 (2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读：仅可读取、写：仅可写入、读 / 写：可读取及写入、-：不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串*1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	BACnetBinaryPV	读	读	参见第 15-41 页上的“Present_Value”
Description	字符串*1	读	-	通过 WindLDR 设置固定值。
Device_Type	字符串*1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 的值是否可信。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Polarity	BACnetPolarity	读 / 写	-	参见第 15-54 页上的“Polarity”
Priority_Array	BACnetPriority_Array	读 / 写	-	参见第 15-54 页上的“Priority_Array”
Relinquish_Default	BACnetBinaryPV	读 / 写	-	参见第 15-54 页上的“Relinquish_Default”
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8)，最大为 64 个字节。

Binary Value 对象

用法与 Binary Input 对象或 Binary Output 对象相同。

属性一览

显示 Binary Value 对象支持的属性。

(1): 从 BACnet 设备读取 / 写入

(2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读: 仅可读取、写: 仅可写入、读 / 写: 可读取及写入、-: 不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	通过 WindLDR 设置固定值。
Object_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	通过 WindLDR 设置固定值。
Present_Value	BACnetBinaryPV	读	读	参见第 15-41 页上的“Present_Value”
Description	字符串 *1	读	-	通过 WindLDR 设置固定值。
Status_Flags	BACnetStatus_Flags	读	读	参见第 15-54 页上的“Status_Flags”
Event_State	BACnetEventState	读	-	Normal 固定。
Reliability	BACnetReliability	读	读 / 写	显示 Present_Value 的值是否可信。
Out_Of_Service	逻辑值	读 / 写	读 / 写	参见第 15-55 页上的“Out_Of_Service”
Priority_Array	BACnetPriority_Array	读 / 写	读 / 写	参见第 15-54 页上的“Priority_Array”
Relinquish_Default	BACnetBinaryPV	读 / 写	-	参见第 15-54 页上的“Relinquish_Default”
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。

*1 编码为 ISO 10646 (UTF-8), 最大为 64 个字节。

Device 对象

在向 BACnet 设备公开 Plus CPU 模块的基本信息时使用。

属性一览

显示 Device 对象支持的属性。

(1): 从 BACnet 设备读取 / 写入

(2): 从将属性分配到设备时的梯形图程序执行读取 / 写入

读: 仅可读取、写: 仅可写入、读 / 写: 可读取及写入、-: 不可读取且不可写入

属性的标识符	属性的数据类型	(1)	(2)	备注
Object_Identifier	BACnetObjectIdentifier	读	-	WindLDR 无法编辑。
Object_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Object_Type	BACnetObjectType	读	-	WindLDR 无法编辑。
System_Status	BACnetDeviceStatus	读	-	WindLDR 无法编辑。
Vendor_Name	字符串 *1	读	-	WindLDR 无法编辑。
Vendor_Identifier	16 位无符号整数	读	-	WindLDR 无法编辑。
Model_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。
Firmware_Revision	字符串 *1	读	-	WindLDR 无法编辑。
Application_Software_Version	字符串 *1	读	-	通过 WindLDR 设置固定值。
Location	字符串 *1	读 / 写	-	通过 WindLDR 设置初始值。
Description	字符串 *1	读 / 写	-	通过 WindLDR 设置初始值。
Protocol_Version	无符号整数	读	-	WindLDR 无法编辑。
Protocol_Revision	无符号整数	读	-	WindLDR 无法编辑。
Protocol_Services_Supported	BACnetServicesSupported	读	-	WindLDR 无法编辑。
Protocol_Object_Types_Supported	BACnetObjectType Supported	读	-	WindLDR 无法编辑。
Object_List	BACnetObjectIdentifier 型 BACnetARRAY[N]	读	-	WindLDR 无法编辑。
Max_APDU_Length_Accepted	无符号整数	读	-	WindLDR 无法编辑。
Segmentation_Supported	BACnetSegmentation	读	-	WindLDR 无法编辑。
Local_Time	时间	读	-	WindLDR 无法编辑。
Local_Date	日期	读	-	WindLDR 无法编辑。
APDU_Timeout	无符号整数	读	-	WindLDR 无法编辑。
Number_of_APDU_Retries	无符号整数	读	-	WindLDR 无法编辑。
Device_Address_Binding	BACnetAddressBiding 型 BACnetLIST	读	-	WindLDR 无法编辑。
Database_Revision	无符号整数	读	-	WindLDR 无法编辑。
Property_List	BACnetPropertyIdentifier 型 BACnetARRAY[N]	Hidden	-	WindLDR 中不显示。
Profile_Name	字符串 *1	读	-	通过 WindLDR 设置固定值。

*1 编码为 ISO 10646 (UTF-8)，最大为 64 个字节。

主要属性

Present_Value

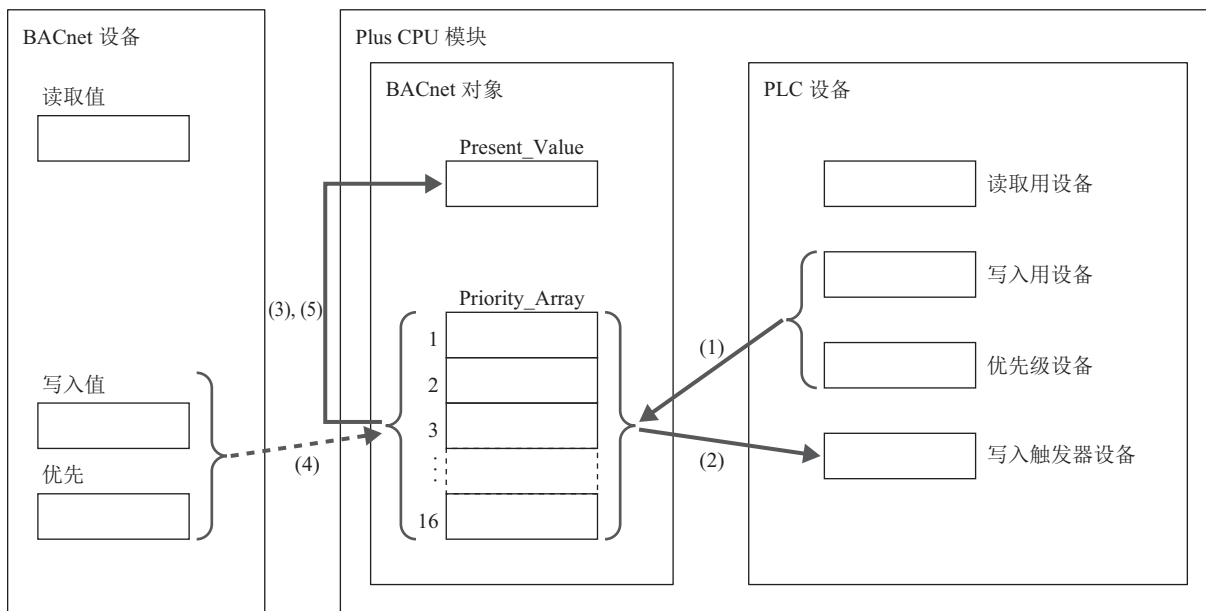
代表当前值的属性。处理与对象相关联的输入输出值等的属性。

优先顺序结构

具备可从 BACnet 设备写入的 Present_Value 的对象^{*1}，通过利用 Priority_Array 的优先顺序结构，为写入指令进行排序，决定 Present_Value 的值。此时，无法将值直接写入 Present_Value。需要将值写入 Present_Value 时，暂时将值存储到“优先”（(4)第 15-21 页上的“Present_Value 设置”）所示目录编号的 Priority_Array（第 15-54 页上的“Priority_Array”）中。随后，在已存储的非 NULL 值中，被存储到目录编号最小的 Priority_Array 中的值，将被作为 Present_Value 的值。

*1 Analog Output/ Analog Value/ Binary Output/ Binary Value 对象

■ 从梯形图程序将数据寄存器的值作为 Present_Value 写入，或从 BACnet 设备写入 Present_Value 时



有关上图中涉及的各项目的说明，请参见下表。

项目	说明
读取用设备	存储从 BACnet 设备读取的 Present_Value 的设备。
写入用设备	存储作为 Present_Value，写入到 BACnet 设备中的值的设备。
优先级设备	存储保存有优先顺序值排列（Priority_Array）目录编号的设备。
写入触发器设备	1 时将存储在写入用设备中的值存储到优先级设备中存储的目录编号 Priority_Array 中。
Present_Value	由 Plus CPU 模块的 BACnet 对象保持的 Present_Value。
Priority_Array	参见第 15-54 页上的“Priority_Array”
读取值	当前的 Present_Value。
写入值	从 BACnet 设备写入的 Present_Value。

从梯形图程序写入

- (1) 当“优先级设备”的位 15 为 0，“写入触发器设备”为 1 时，“优先级设备”中存储的目录编号 Priority_Array 中，将写入“写入用设备”的值。
- (2) “写入触发器设备”自动恢复为 0。
- (3) 在已存储的非 NULL 值中，被存储到目录编号最小的 Priority_Array 中的值，将被作为 Present_Value 的值。（在作为 Present_Value 使用的目录编号 Priority_Array 值恢复为 NULL 之前，相应的目录编号 Priority_Array 值将始终被作为 Present_Value 的值。）

注释：当“优先级设备”的位 15 为 1，“写入触发器设备”为 1 时，“优先级设备”中存储的目录编号 Priority_Array 中将写入 NULL。

有关优先级设备位分配的详情，请参见第 15-25 页上的“Analog Value 对象”的第 15-26 页上的“(2) 针对优先级和触发器的顶端设备”。

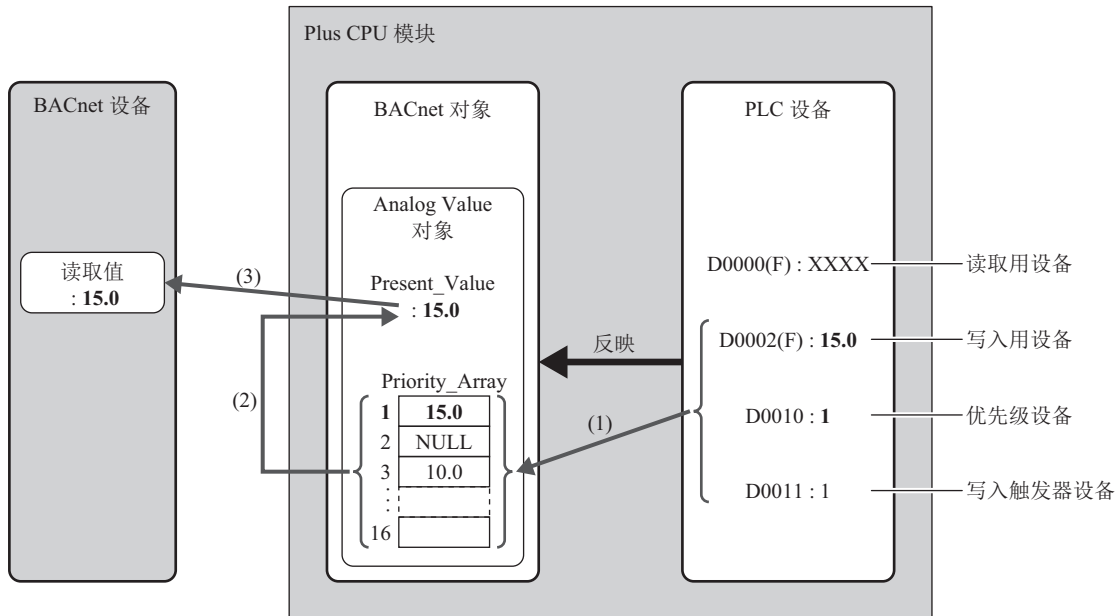
从 BACnet 设备写入

- (4) 将 Present_Value 的值，写入 BACnet 设备指定的优先级所示的目录编号的 Priority_Array 中。
- (5) 在已存储的非 NULL 值中，被存储到目录编号最小的 Priority_Array 属性中的值，将被作为 Present_Value 的值。（在作为 Present_Value 使用的目录编号 Priority_Array 值恢复为 NULL 之前，相应的目录编号 Priority_Array 值将始终被作为 Present_Value 的值。）

注释：Priority_Array 中存储的值全部为 NULL 时，第 15-54 页上的“Relinquish_Default”将被作为 Present_Value。

从梯形图程序将数据寄存器的值写入 Present_Value 时

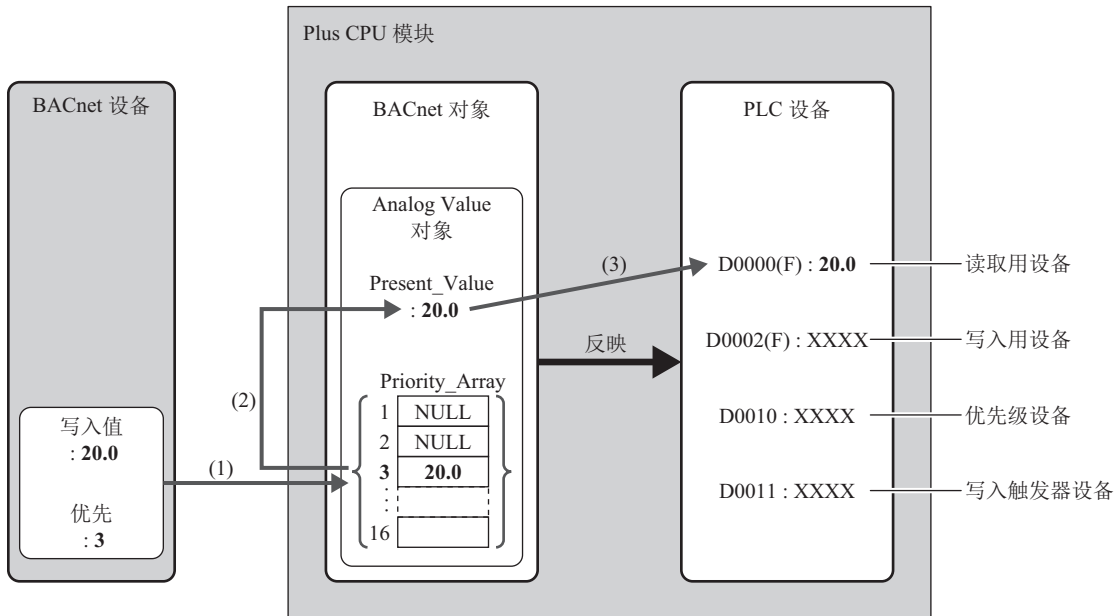
例如，将 D0000 分配到 Present_Value 设备，将 D0010 分配到针对优先级和触发器的顶端设备，将浮点（无转换）分配到转换类型时，将进行读取用设备：D0000、写入用设备：D0002、优先级设备：D0010、写入触发器设备：D0011 的分配。



- (1) 写入触发器设备（D0011）为 1 时，优先级设备（D0010）为 1，因此将写入用设备（D0002）的值（15.0）写入到 Priority_Array 的第 1 要素。写入完成后，将写入触发器设备（D0011）恢复为 0。
- (2) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 1 要素（15.0），15.0 将被作为 Present_Value 的值。
- (3) 将从 BACnet 设备读取 Present_Value（15.0）。

从 BACnet 设备将值写入 Present_Value 时

例如，将 D0000 分配到 Present_Value 设备，将 D0010 分配到针对优先级和触发器的顶端设备，将浮点（无转换）分配到转换类型时，将进行读取用设备：D0000、写入用设备：D0002、优先级设备：D0010、写入触发器设备：D0011 的分配。

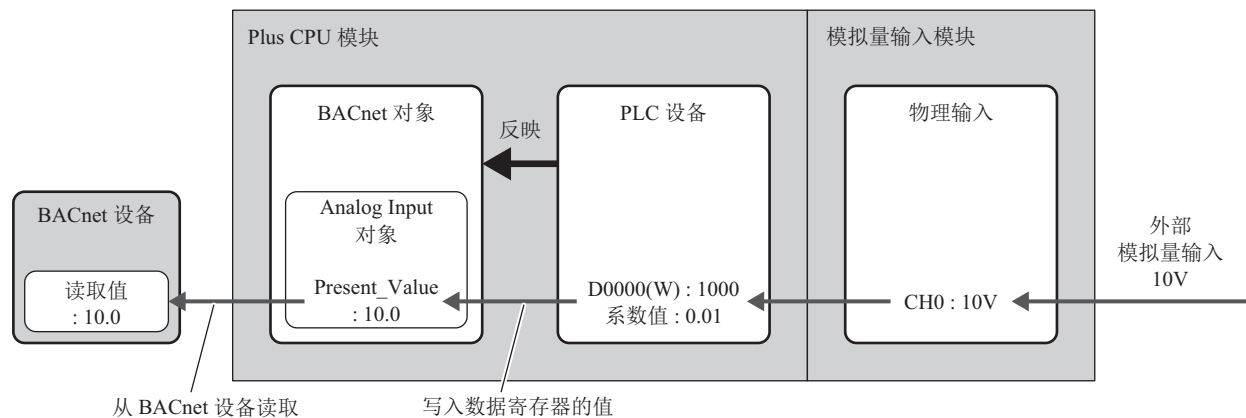


- (1) 将写入值 (20.0)，写入到 BACnet 设备指定的优先级 (3) 所示的目录编号 Priority_Array 中。
- (2) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 3 要素 (20.0)，20.0 将被作为 Present_Value 的值。
- (3) 将 Present_Value (20.0) 写入读取用设备 (D0000)。

Analog Input 对象

对于 Analog Input 对象的 Present Value，可以设置固定值，也可以分配数据寄存器，设置相应的数据寄存器值。Present_Value 为浮点型数值。将数据寄存器分配到 Present_Value 时，相应数据寄存器的值乘上系数值后，即为 Present_Value 的设置值。

下图所示的是，将数据寄存器分配到 Analog Input 对象的 Present_Value 时，BACnet 设备读取模拟量输入值的情形。



能够分配到 Analog Input 对象 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

将数据寄存器的值作为 Present_Value 写入时，Present_Value 将发生如下变化。

数据类型	数据寄存器的值	BACnet 设备的 Present_Value
字	各数据类型范围内的值	数据寄存器的值
整数		
双字		
长整		
浮点（无转换）	±0	±0.0
	非规格化数	数据寄存器的值
	规格化数	
	±∞（±无限大） 非数	Present_Value 不变

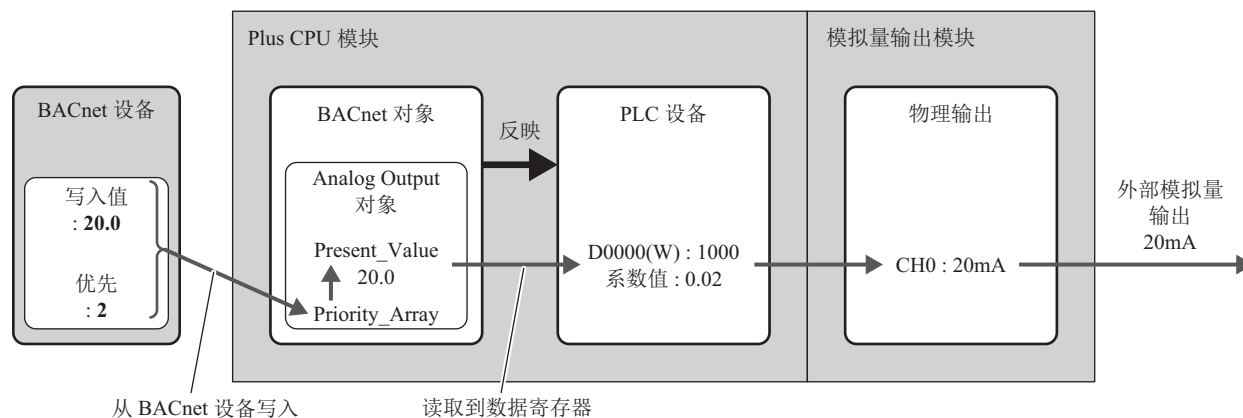
Plus CPU 模块处理的 Analog Input 对象 Present_Value，会根据相应的数据类型，出现写入所必需的数据寄存器数量不同的情况。

Plus CPU 模块处理的 Present_Value 数据类型	数据寄存器的数量
字	1
整数	
双字	2
长整	
浮点（无转换）	

Analog Output 对象

可以将数据寄存器分配到 Analog Output 对象的 Present_Value 中，并设置相应的数据寄存器值。Present_Value 为浮点型数值。将数据寄存器分配到 Present_Value 时，Present_Value 乘上 1/ 系数值后所得的值，会被存储到数据寄存器中。

下图所示的是，将数据寄存器分配到 Analog Output 对象的 Present_Value 时，BACnet 设备写入模拟量输出值的情形。



注释： 不能从梯形图程序更改 Analog Output 对象的 Present_Value。

能够分配到 Analog Output 对象 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

将 Present_Value 读取到数据寄存器时，必须注意数据类型。根据数据寄存器的数据类型，以如下所示的方式存储到数据寄存器中。请根据 Present_Value 的值，设置相应的数据类型。Present_Value 的数据类型，在“Present_Value 设置”对话框（第 15-24 页）的第 15-24 页上的“(2) 转换类型”中设置。

数据类型	BACnet 设备的 Present_Value	数据寄存器的值
字	0 ~ 65535 范围内的值	Present_Value
	0 ~ 65535 范围外的值	0
整数	-32768 ~ 32767 范围内的值	Present_Value
	-32768 ~ 32767 范围外的值	0
双字	0 ~ 4294967295 范围内的值	Present_Value
	0 ~ 4294967295 范围外的值	0
长整	-2147483648 ~ 2147483647 范围内的值	Present_Value
	-2147483648 ~ 2147483647 范围外的值	0
浮点（无转换）	—	Present_Value

Plus CPU 模块处理的 Analog Output 对象 Present_Value，会根据相应的数据类型，出现读取所必需的数据寄存器数量不同的情况。

Plus CPU 模块处理的 Present_Value 数据类型	数据寄存器的数量
字	1
整数	
双字	2
长整	
浮点（无转换）	

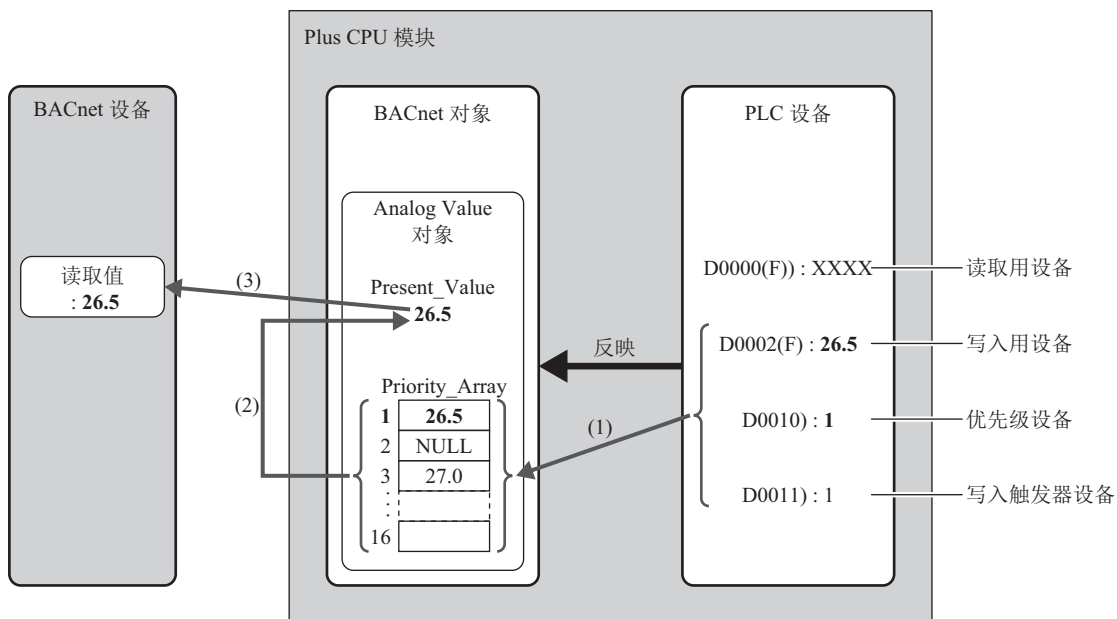
Analog Value 对象

Analog Value 对象的 Present_Value，支持 Analog Input 对象和 Analog Output 对象的 Present_Value 两种用法。Present_Value 为浮点型数值。

下图所示的是，将数据寄存器分配到 Analog Value 对象的 Present_Value 时，办公室空调温度由基准温度（27.0°C）暂时降低到 26.5°C 的情形。

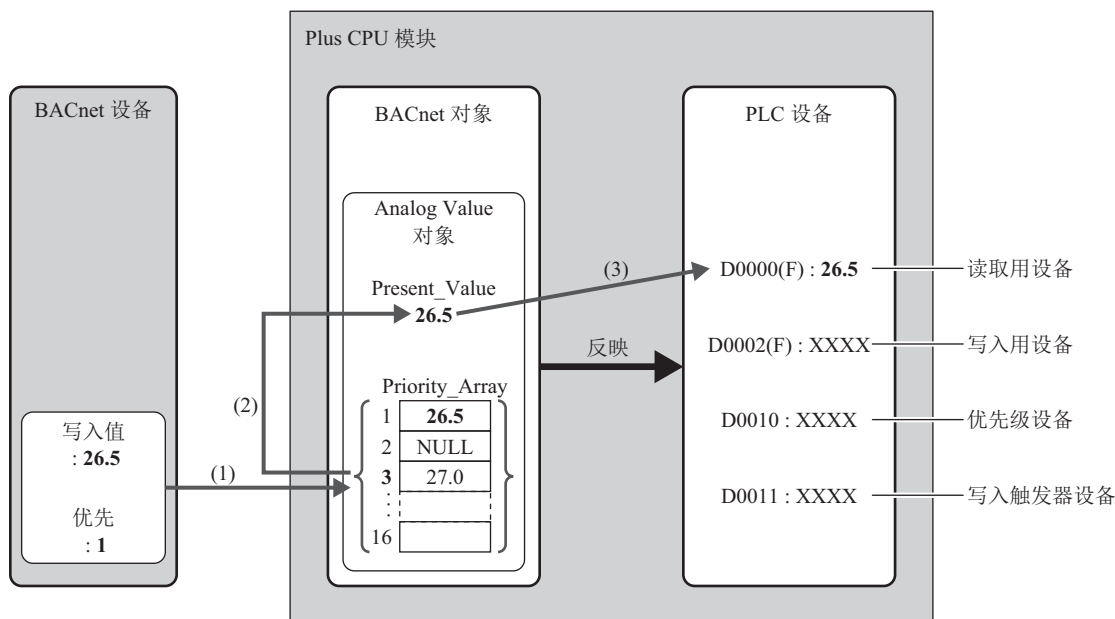
例如，将 D0000 分配到 Present_Value 设备，将 D0010 分配到针对优先级和触发器的顶端设备，将浮点（无转换）分配到转换类型时，将进行读取用设备：D0000、写入用设备：D0002、优先级设备：D0010、写入触发器设备：D0011 的分配。

从梯形图程序将数据寄存器的值写入 Present_Value 时



- (1) 写入触发器设备 (D0011) 为 1 时，将写入用设备 (D0002) 的值写入到优先级设备 (D0010) 中存储的目录编号 Priority_Array 中。
- (2) 写入触发器设备 (D0011) 自动恢复为 0。
- (3) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 1 要素 (26.5)，26.5 将被作为 Present_Value 的值。

从 BACnet 设备将值写入 Present_Value 时



- (1) 将写入值 (26.5)，写入到 BACnet 设备指定的优先级 (1) 所示的目录编号 Priority_Array 中。
- (2) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 1 要素 (26.5)，26.5 将被作为 Present_Value 的值。
- (3) 将 Present_Value (26.5) 写入读取用设备 (D0000)。

注释: 在作为 Present_Value 使用的 Priority_Array 第 1 要素 (26.5) 恢复为 NULL 之前，第 1 要素将始终被作为 Present_Value 的值。第 1 要素恢复为 NULL 后，第 3 要素 (27.0) 将被作为 Present_Value 的值。有关 NULL 写入方法的详情，请参见第 15-25 页上的“Analog Value 对象”的第 15-26 页上的“(2) 针对优先级和触发器的顶端设备”。

能够分配到 Analog Value 对象 Present_Value 的设备仅限数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

根据 Plus CPU 模块处理的 Analog Value 对象 Present_Value 的数据类型，会出现读取 / 写入所必需的数据寄存器数量不同的情况。以指定的数据寄存器为起始，占用 2 个或 4 个数据寄存器。

Plus CPU 模块处理的 Present_Value 数据类型	数据寄存器的数量		
	读取	写入	合计
字	1	1	2
整数			
双字	2	2	4
长整			
浮点 (无转换)			

根据分配数据寄存器的数据类型，Present_Value 以如下所示的方式被存储到数据寄存器中。请根据 Present_Value 的值，设置相应的数据类型。Present_Value 的数据类型，在“Present_Value 设置”对话框（第 15-25 页）的第 15-26 页上的“(3) 转换类型”中设置。

数据类型	BACnet 设备的 Present_Value × (1/ 系数值)	数据寄存器的值
字	0 ~ 65535 范围内的值	Present_Value
	0 ~ 65535 范围外的值	0
整数	-32768 ~ 32767 范围内的值	Present_Value
	-32768 ~ 32767 范围外的值	0
双字	0 ~ 4294967295 范围内的值	Present_Value
	0 ~ 4294967295 范围外的值	0
长整	-2147483648 ~ 2147483647 范围内的值	Present_Value
	-2147483648 ~ 2147483647 范围外的值	0
浮点（无转换）	—	Present_Value

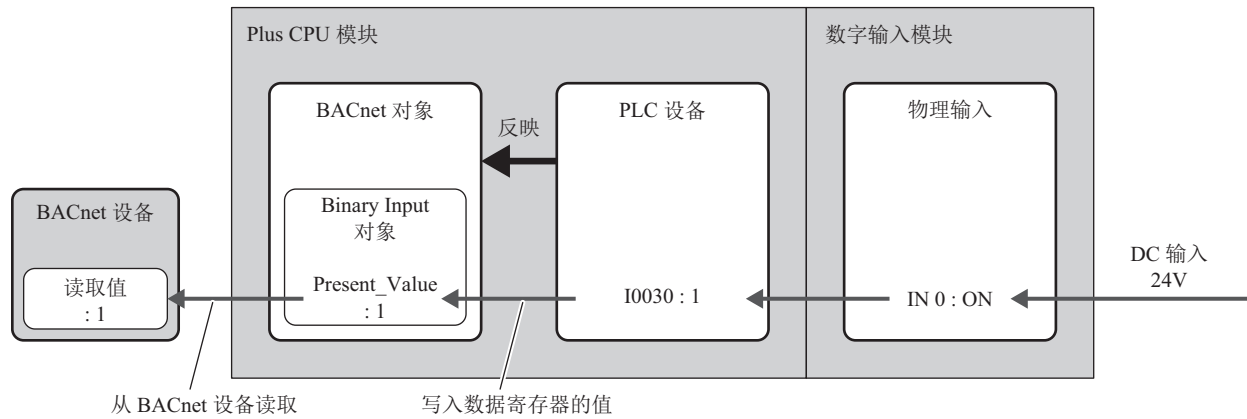
此外，将数据寄存器的值写入 Present_Value 时，Present_Value 将发生如下变化。

数据类型	数据寄存器的值	BACnet 设备的 Present_Value
字	各数据类型范围内的值	数据寄存器的值 × 系数值
整数		
双字		
长整		
浮点（无转换）	±0	±0.0
	非规格化数	数据寄存器的值 × 系数值
	规格化数	
	±∞（± 无限大）	Present_Value 不变
非数		

Binary Input 对象

对于 Binary Input 对象的 Present_Value，可以设置固定值，也可以分配位设备，设置相应的位设备值。Present_Value 为二进制值。

下图所示的是，将外部输入分配到 Binary Input 对象的 Present_Value，BACnet 设备读取外部输入状态的情形。



能够分配到 Binary Input 对象 Present_Value 的设备如下所示。

属性	I	Q	M	R	T	C	D	常量
Present_Value	是	—	是*1	—	—	—	—	—

*1 无法使用特殊内部继电器。

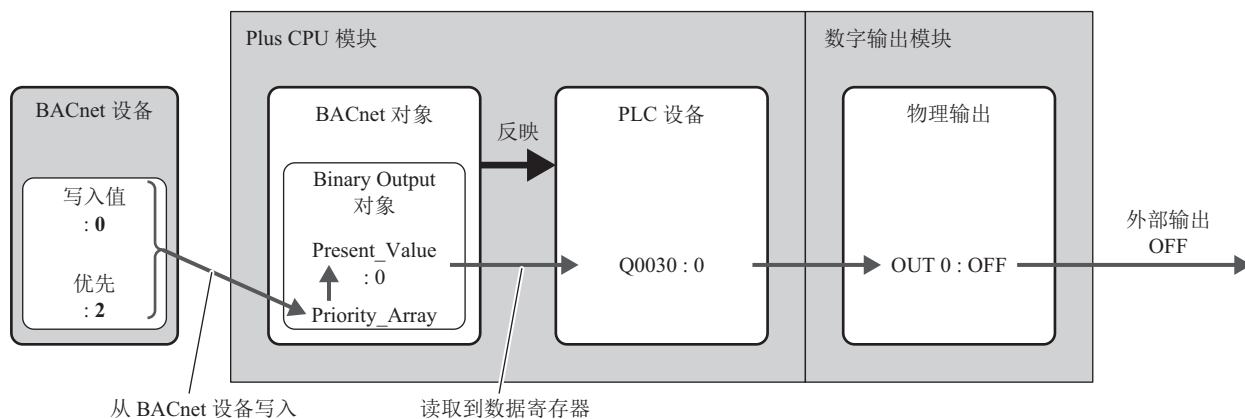
有关 Present_Value、Polarity 及输入的物理状态，如下表所示。

Present_Value	Polarity	输入的物理状态
INACTIVE	NORMAL	OFF 或 INACTIVE
ACTIVE	NORMAL	ON 或 ACTIVE
INACTIVE	REVERSE	ON 或 ACTIVE
ACTIVE	REVERSE	OFF 或 INACTIVE

Binary Output 对象

对于 Binary Output 对象的 Present_Value，可分配位设备，设置相应的位设备值。Present_Value 为二进制值。

下图所示的是，将数据寄存器分配到 Binary Output 对象的 Present_Value 时，BACnet 设备写入外部输出状态的情形。



能够分配到 Binary Output 对象的 Present_Value 的设备如下所示。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	是	是*1	—	—	—	—	—

*1 无法使用特殊内部继电器。

有关 Present_Value、Polarity 及输出的物理状态，如下表所示。

Present_Value	Polarity	输出的物理状态
INACTIVE	NORMAL	OFF 或 INACTIVE
ACTIVE	NORMAL	ON 或 ACTIVE
INACTIVE	REVERSE	ON 或 ACTIVE
ACTIVE	REVERSE	OFF 或 INACTIVE

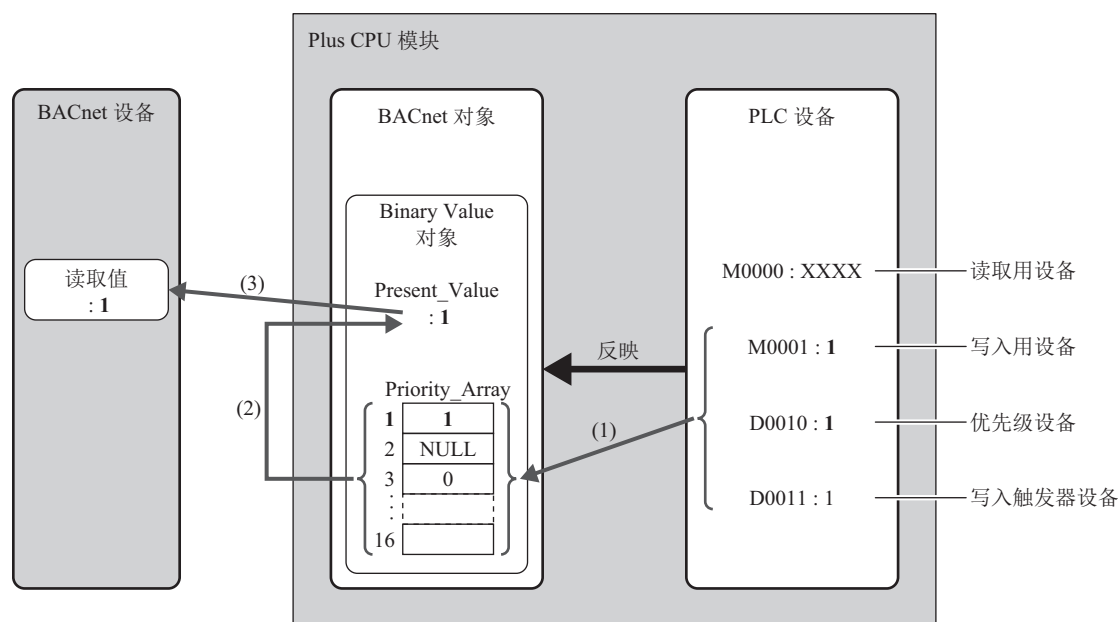
Binary Value 对象

Binary Value 对象的 Present_Value，支持 Binary Input 对象和 Binary Output 对象的 Present_Value 两种用法。Present_Value 为二进制值。

下图所示的是，将内部继电器分配到 Binary Value 对象的 Present_Value 时，办公室照明由 OFF 暂时切换到 ON 的情形。

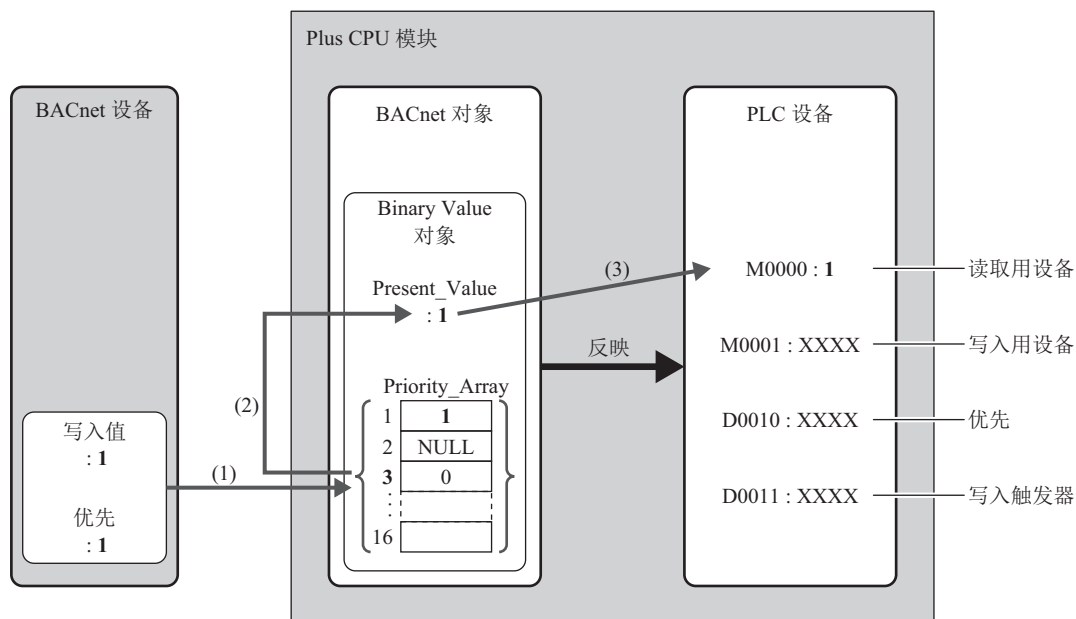
例如，将 M0000 分配到 Present_Value 设备，将 D0010 分配到针对优先级和触发器的顶端设备时，将进行读取用设备：M0000、写入用设备：M0001、优先级设备：D0010、写入触发器设备：D0011 的分配。

从梯形图程序将内部继电器的值写入 Present_Value 时



- (1) 写入触发器设备（D0011）为 1 时，将写入用设备（M0001）的值写入到优先级设备（D0010）中存储的目录编号 Priority_Array 中。
- (2) 写入触发器设备（D0011）自动恢复为 0。
- (3) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 1 要素（1），1 将被作为 Present_Value 的值。

从 BACnet 设备将值写入 Present_Value 时



- (1) 将写入值 (1)，写入到 BACnet 设备指定的优先级 (1) 所示的目录编号 Priority_Array 中。
- (2) 在已存储的非 NULL 值中，由于目录编号最小的 Priority_Array 为第 1 要素 (1)，1 将被作为 Present_Value 的值。
- (3) 将 Present_Value (1) 写入读取用设备 (M0000)。

注释：在作为 Present_Value 使用的 Priority_Array 第 1 要素 (1) 恢复为 NULL 之前，第 1 要素将始终被作为 Present_Value 的值。第 1 要素恢复为 NULL 后，第 3 要素 (0) 将被作为 Present_Value 的值。有关 NULL 写入方法的详情，请参见第 15-30 页上的“Binary Value 对象”的第 15-31 页上的“(2) 针对优先级和触发器的顶端设备”。

能够分配到 Binary Value 对象的 Present_Value 的设备如下所示。

属性	I	Q	M	R	T	C	D	常量
Present_Value	—	—	是*1	—	—	—	—	—

*1 无法使用特殊内部继电器。

Plus CPU 模块处理的 Binary Value 对象 Present_Value 中，必须分别配备读取用及写入用内部继电器。以指定的内部继电器为起始，占用 2 个内部继电器。

设备	读取	写入	合计
内部继电器	1	1	2

Status_Flags

表示对象当前状态（报警中 / 故障中 / 维护中等）的属性。

Status_Flags	值	逻辑值	条件
IN_ALARM*1	0	FALSE	Event State 取值 Normal 时
	1	TRUE	上述以外
FAULT	0	FALSE	下述以外
	1	TRUE	Reliability 存在，不取值 no-fault-detected 时
OVERRIDDEN*1	0	FALSE	下述以外
	1	TRUE	Present_Value 及 Reliability 不跟随物理输入的变化
OUT_OF_SERVICE	0	FALSE	Out_Of_Service 为 FALSE 时
	1	TRUE	Out_Of_Service 为 TRUE 时

*1 Plus CPU 模块始终 FALSE。

Status_Flags 可读取到数据寄存器。

属性	I	Q	M	R	T	C	D	常量
Status_Flags	—	—	—	—	—	—	是*1	—

*1 无法使用特殊数据寄存器。

数据寄存器中各标记的分配（位分配）如下所示。

位	标记	值
位 15 ~ 8	保留	不确定
位 7	IN_ALARM	0 固定
位 6	FAULT	0 / 1
位 5	OVERRIDDEN	0 固定
位 4	OUT_OF_SERVICE	0 / 1
位 3 ~ 0	保留	不确定

COV_Increment

代表 Present_Value 中最小变化量的属性。

如果最后发送的 COV 通知的 Present_Value 值变化为超出 COV_Increment 设置值的值，将发送 COV 通知。COV_Increment 不能读取到设备。通过 WindLDR 设置初始值。

Priority_Array

Priority_Array 属于只读属性，代表存储有优先顺序值的排列。

在 16 个要素（第 1 要素 ~ 第 16 要素）中，存储有非 NULL 值的目录编号最小 Priority_Array 中存储的值将被作为 Present_Value 的值。Priority_Array 中存储的值全部为 NULL 时，Relinquish_Default 将被作为 Present_Value。

Relinquish_Default

Priority_Array 中存储的值全部为 NULL 时，Relinquish_Default 将被作为 Present_Value 的默认值。

Polarity

对于 Binary Input 及 Binary Output 对象，代表输入输出物理状态与 Present_Value 所示逻辑状态间关系的属性。

Polarity	输入输出的物理状态	Present_Value	设备的物理状态
NORMAL	OFF/ INACTIVE	INACTIVE	非运行中
NORMAL	ON/ ACTIVE	ACTIVE	运行中
REVERSE	ON/ ACTIVE	INACTIVE	非运行中
REVERSE	OFF/ INACTIVE	ACTIVE	运行中

Out_Of_Service

Out_Of_Service 是代表 Present_Value 与物理输入输出是否被断开的属性。

Out_Of_Service 为 TRUE 时，代表维护中。

属性	值	逻辑值	条件
Out_Of_Service	0	FALSE	服务中 (Present_Value 与物理输入输出联动)
	1	TRUE	维护中 (Present_Value 与物理输入输出已断开)

可以将 Out_Of_Service 读取到内部继电器，或将内部继电器的状态作为 Out_Of_Service 写入。

属性	I	Q	M	R	T	C	D	常量
Out_Of_Service	—	—	是 *1	—	—	—	—	—

*1 无法使用特殊内部继电器。

注释： Out_Of_Service = TRUE 用于模拟。

Reliability

代表对象属性可靠性的属性。

各对象类型 Reliability 的定义如下表所示。

是：启用、否：禁用

定义	值	Analog Input	Analog Output	Analog Value	Binary Input	Binary Output	Binary Value
no-fault-detected	0	是	是	是	是	是	是
no-sensor	1	是	否	否	是	否	否
over-range	2	是	否	是	否	否	否
under-range	3	是	否	是	否	否	否
open-loop	4	是	是	否	是	是	否
shorted-loop	5	是	是	否	是	是	否
no-output	6	否	是	否	否	是	否
unreliable-other	7	是	是	是	是	是	是
process-error	8	否	否	否	否	否	否
multi-state-fault	9	否	否	否	否	否	否
configuration-error	10	否	否	否	否	否	否
-- enumeration value 11 is reserved for a future addendum	11	是	是	否	是	是	是
communication-failure	12	是	是	是	是	是	是
member-fault	13	否	否	否	否	否	否
monitored-object-fault	14	否	否	否	否	否	否
tripped	15	否	否	否	否	否	否

可以将 Reliability 读取到数据寄存器，或将数据寄存器的值作为 Reliability 写入。

属性	I	Q	M	R	T	C	D	常量
Reliability	—	—	—	—	—	—	是 *1	—

*1 无法使用特殊数据寄存器。

System_Status

显示 Plus CPU 模块的物理状态及逻辑状态。

参数	值
OPERATIONAL	0
OPERATIONAL_READ_ONLY	1
DOWNLOAD_REQUIRED	2
DOWNLOAD_IN_PROGRESS	3
NON_OPERATIONAL	4
BACKUP_IN_PROGRESS	5

Plus CPU 模块的 System_Status 固定为 OPERATIONAL。

Firmware_Revision

Plus CPU 模块的系统固件版本将被设置。

Application_Software_Version

已创建梯形图程序的修改日期等，与应用程序相对应的信息，将由 WindLDR 设置为固定的字符串。

Protocol_Services_Supported

显示 Plus CPU 模块支持的服务种类。

Protocol_Object_Types_Supported

显示 Plus CPU 模块支持的对象种类。

Object_List

显示已创建对象的一览。

16: EtherNet/IP 通信

本章将对 EtherNet/IP 通信功能进行介绍。

概要

Plus CPU 模块支持 EtherNet/IP (扫描仪*1) 通信。

EtherNet/IP 是“Ethernet Industrial Protocol”的缩略，是一种使用以太网的生产用多厂商网络。

Plus CPU 模块为 EtherNet/IP 通信的扫描仪，可与支持 EtherNet/IP 通信的设备进行通信。

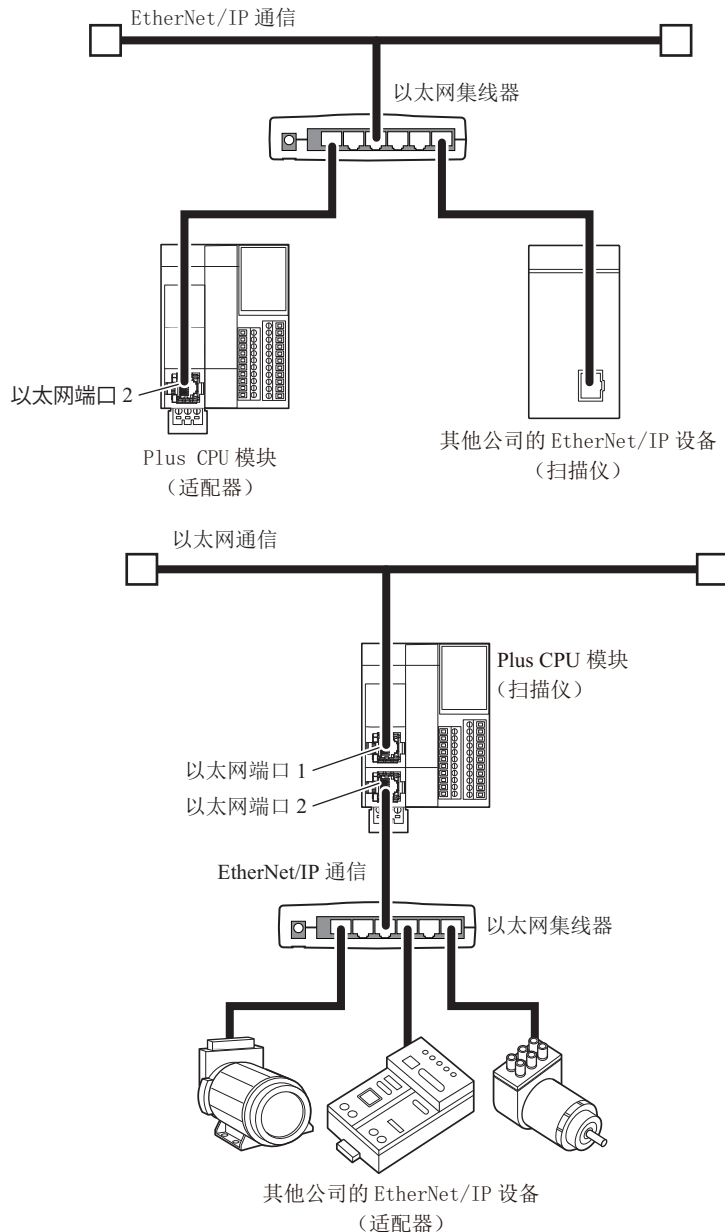
由于 EtherNet/IP 通信采用了标准的以太网技术，可以混用各类支持以太网的设备来构建网络。Plus CPU 模块不仅具备扫描仪功能，还具备了适配器*1 功能，可以作为 EtherNet/IP 通信的适配器。

*1 EtherNet/IP 通信将主机称为“扫描仪”，将从机称为“适配器”。通常，将具备接收设立 CIP 连接功能请求的 EtherNet/IP 设备称为适配器，将同时也具备请求功能的 EtherNet/IP 设备称为扫描仪。

*2 接下来，本手册将把支持 EtherNet/IP 通信的设备称为 EtherNet/IP 设备。

EtherNet/IP 设备之间的多厂商网络

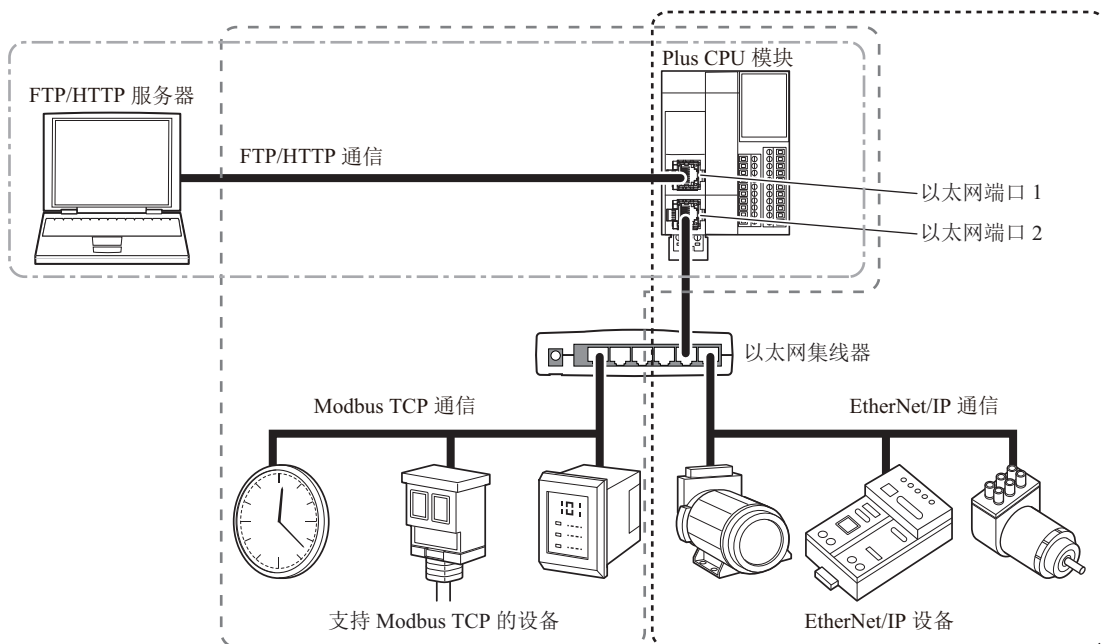
可与 EtherNet/IP 设备之间构建多厂商网络。



16: ETHERNET/IP通信

与通用以太网设备混合存在的网络

可构建 EtherNet/IP 设备与通用以太网设备混合存在的网络。



注释: 在同一网络内同时进行 EtherNet/IP 通信和其他以太网通信时，根据负载情况可能无法按照预期时间进行收发数据。此时，请使用支持 QoS 功能的以太网交换机构建网络，调整网络内流通数据的优先级。

EtherNet/IP 中使用的通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
以太网端口 1	—	—	—	—	—	—
以太网端口 2	—	—	—	—	是	是
HMI- 以太网端口	—	—	—	—	—	—

EtherNet/IP 规格

模块		Plus CPU 模块	
支持端口		以太网端口 2	
I/O 信息通信功能	CIP 连接数		32 个*1
	CIP 连接点	设置数	8 个
		已预定义	实例 ID 198 (Input Only 输出用) 实例 ID 199 (Listen Only 输出用)
	RPI (通信周期)		10 ~ 10000ms (以 1ms 为单位)
	CIP 连接类型		Input Only/Exclusive Owner/Listen Only
	发送触发		循环 /COS (Change Of State) *2
	每个 CIP 连接的最大数据大小		504 字节或 1444 字节*3,*4
	I/O 信息 通信允许通信范围	504 字节 / 数据包	600pps
		1444 字节 / 数据包	200pps
多点传输过滤功能*5		支持 (IGMP 客户端功能)	
Explicit 信息通信功能	Class 3 (连接类型)	服务器	CIP 连接数: 32 个*6
		客户端	不支持
	UCMM	服务器	同时执行数: 32
		客户端	不支持

*1 与 Explicit 信息通信功能的 Class3 (连接类型) 使用的 CIP 连接数的合计最大为 32 个。

*2 可与使用 Change Of State (状态变化时发送数据) 方式输出数据的 EtherNet/IP 设备进行通信。

*3 1444 字节时, EtherNet/IP 设备需要支持 Large Forward Open (CIP 可选规格)。

*4 请将每个 I/O 信息通信功能使用的 CIP 连接的数据大小合计控制在 10240 字节内。

*5 由于具有 IGMP 客户端功能 (IGMPv2), 因此可以使用支持 IGMP Snooping 的以太网交换机来过滤不必要的多播数据包。

*6 I/O 信息通信功能使用的 CIP 连接数的合计最大为 32 个。

关于 EtherNet/IP

支持规格

Plus CPU 模块的 EtherNet/IP 通信支持下列规格。

- EtherNet/IP

供应商 ID

供应商 ID 为 159。

设备轮廓

Plus CPU 模块的 EtherNet/IP 通信支持扫描仪。

对象

Plus CPU 模块的 EtherNet/IP 通信支持下列对象。

对象名称	分类代码
Identity	01H
Message Router	02H
Assembly	04H
Connection Manage	06H
TCP/IP Interface	F5H
Ethernet Link	F6H

有关各对象的详情，请参见第 16-41 页上的“对象”。

功能

Plus CPU 模块作为 EtherNet/IP 设备的扫描仪，可提供下列功能。

- I/O 信息通信（发起者 / 目标）功能（第 16-5 页）
- Explicit 信息通信（服务器）功能（第 16-10 页）
- 数据与设备的联动功能（第 16-11 页）

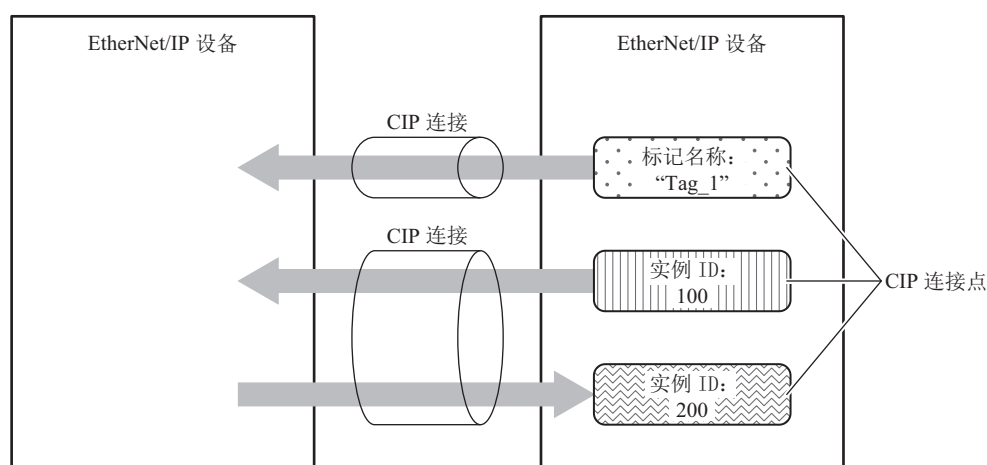
I/O 信息通信功能

本节将对 I/O 信息通信功能和使用方法进行介绍。

■ I/O 信息通信

EtherNet/IP 设备间周期性读写数据的功能。将读写数据的对象的单位称为 CIP 连接点。CIP 连接点上分配 CIP 标记名称或实例 ID。I/O 信息通信中，指定 CIP 连接点的 CIP 标记名称或实例 ID 来读写数据。

I/O 信息通信的概念图



■ CIP 连接

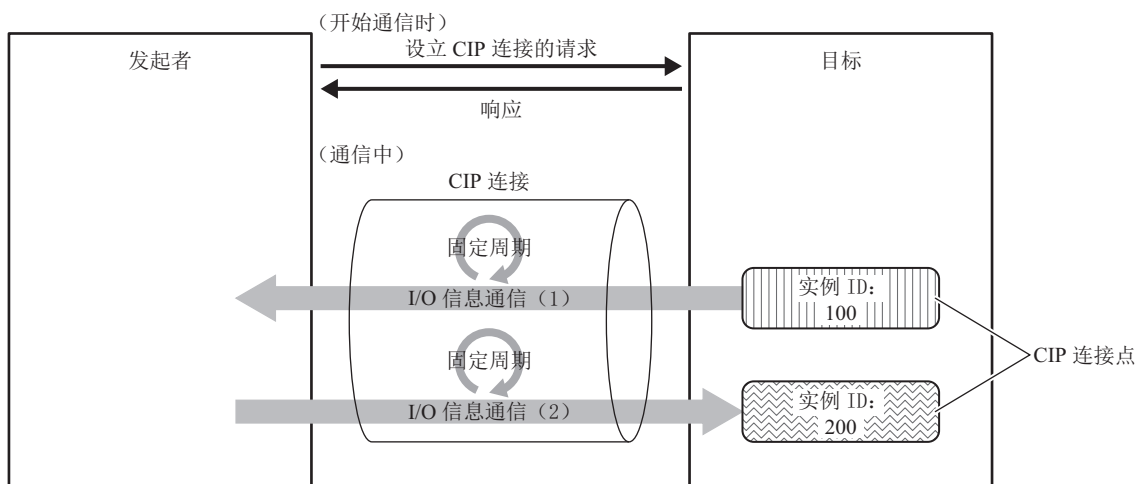
CIP 连接是 EtherNet/IP 设备间的虚拟通信线路。I/O 信息通信中，一方设备向目标设备的 CIP 连接点请求设立 CIP 连接，成功后读写数据。请求设立 CIP 连接的一方称为发起者，接到请求的一方称为目标。Plus CPU 模块既可成为发起者也可成为目标。Plus CPU 模块可同时设立 32 个 CIP 连接。

■ CIP 连接的种类

CIP 连接有以下 3 类。可收发（Exclusive Owner）与仅可接收（Input Only、Listen Only）。

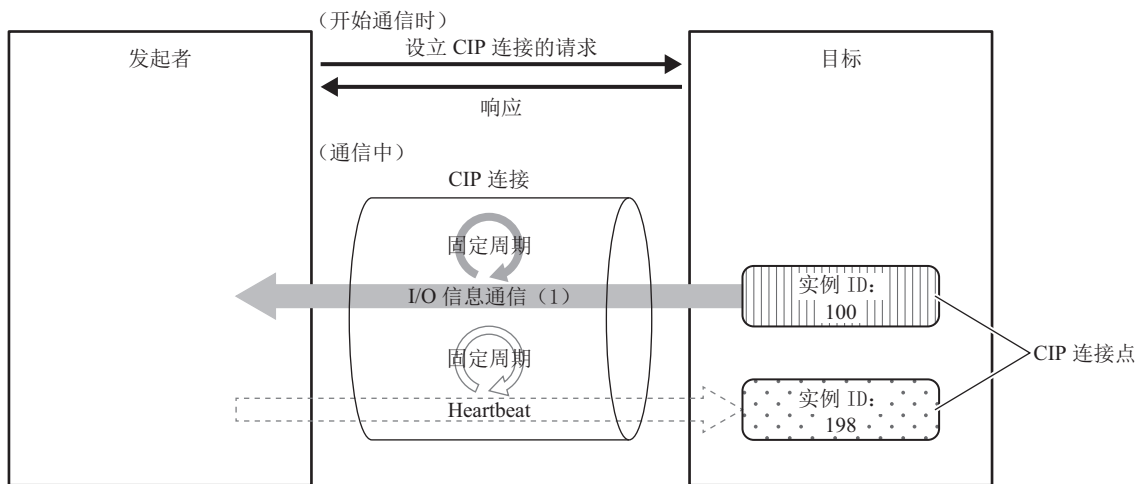
(1) Exclusive Owner

周期性将数据从目标发送到发起者 (1)，以及将数据从发起者发送到目标 (2) 的 CIP 连接。根据实例 ID 或 CIP 标记名称指定 CIP 连接点。设立 CIP 连接时按照指定周期（RPI）进行通信。发起者设立连接后可收发数据。



(2) Input Only

周期性将数据从目标发送到发起者 (1) 的 CIP 连接。根据实例 ID 或 CIP 标记名称指定 CIP 连接点。设立 CIP 连接时按照指定周期（RPI）进行通信。目标为了监控发起者的存在状态，周期性从发起者向目标发送不包含数据的 Heartbeat。发起者作为 Heartbeat 的发送目的地，指定目标拥有的 Input Only 用的实例 ID（实例 ID：198）。按照指定周期（RPI）执行 Heartbeat。

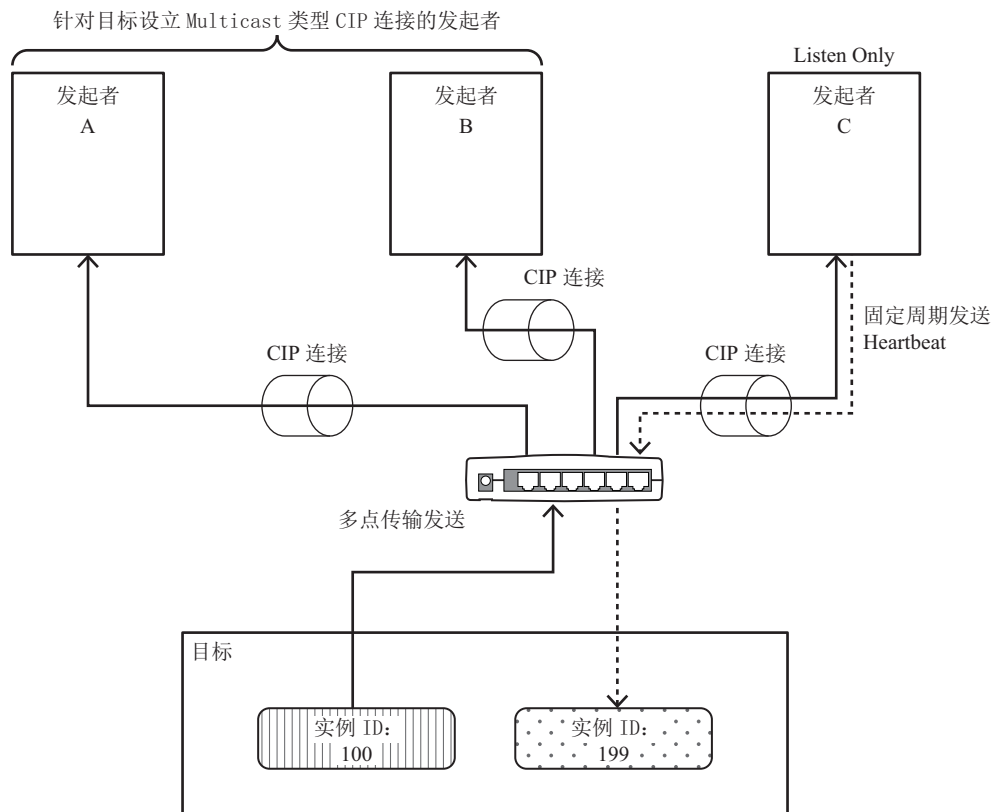


注释:

- 运用 Input Only，根据实例 ID 设置从目标接收输入数据的 CIP 连接点时，也请根据实例 ID 设置向目标发送 Heartbeat 的 CIP 连接点。
- 运用 Input Only，根据 CIP 标记名称设置从目标接收输入数据的 CIP 连接点时，由于不存在向目标发送输出数据的 CIP 连接点的 CIP 标记名称，不进行设置。
- 运用 Input Only 时必定执行 Heartbeat。

(3) Listen Only

从已设立了使用 Exclusive Owner 和 Input Only 的 Multicast 类型 CIP 连接的目标，向发起者周期性执行数据接收的 CIP 连接。目标向其他发起者多点传输发送数据时，在同时接收该数据的情况下进行设置。根据实例 ID 指定 CIP 连接点。设立 CIP 连接时按照指定周期（RPI）进行通信。目标为了监控发起者的存在状态，周期性从发起者向目标发送不包含数据的 Heartbeat。发起者作为 Heartbeat 的发送地址，指定目标拥有的 Listen Only 用的实例 ID（实例 ID：199）。按照指定周期（RPI）执行 Heartbeat。

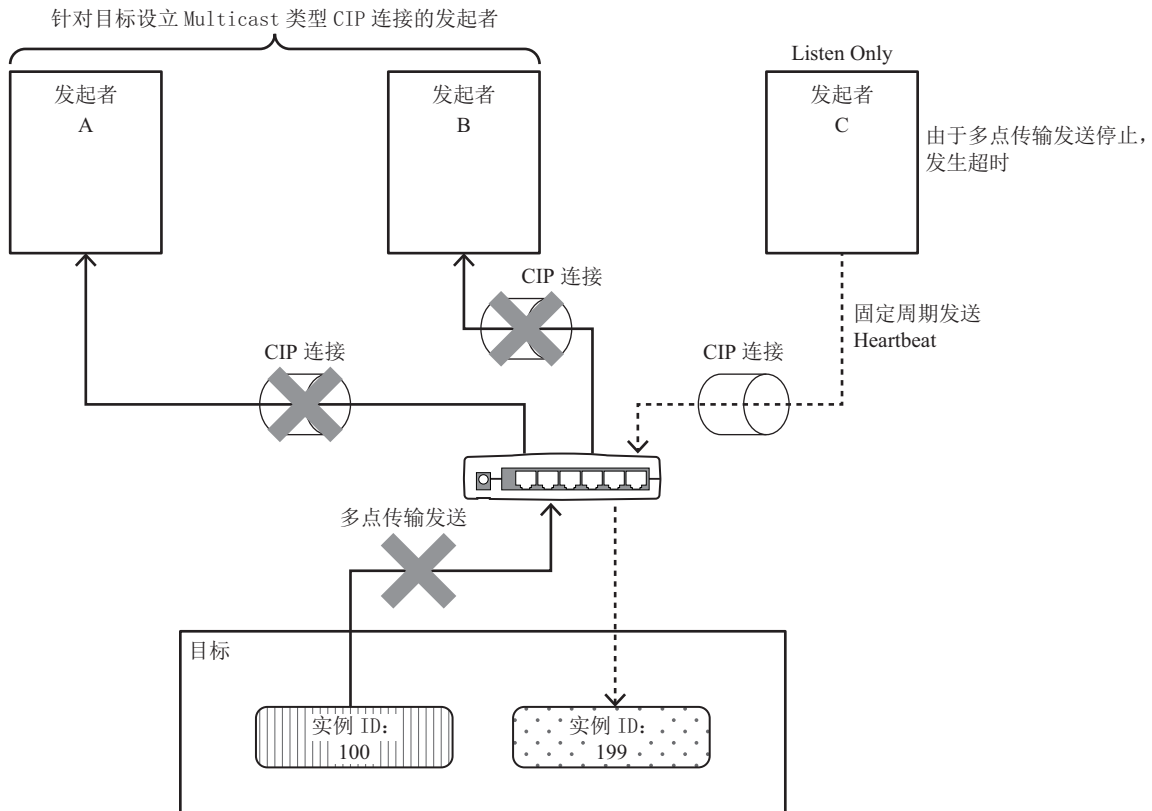


注释:

- 运用 Listen Only，根据实例 ID 设置从目标接收输入数据的 CIP 连接点时，也请根据实例 ID 设置向目标发送输出数据的 CIP 连接点。
- 运用 Listen Only 时必定执行 Heartbeat。
- 可以使用支持 IGMP Snooping 的以太网交换机来过滤不必要的多播数据包。在这种情况下，请安装支持 IGMP 客户端功能的发起方和定期发送 IGMP 查询的 IGMP 查询器（路由器等）。如果没有 IGMP 查询器，则发起者将无法在一段时间后接收多播数据包。

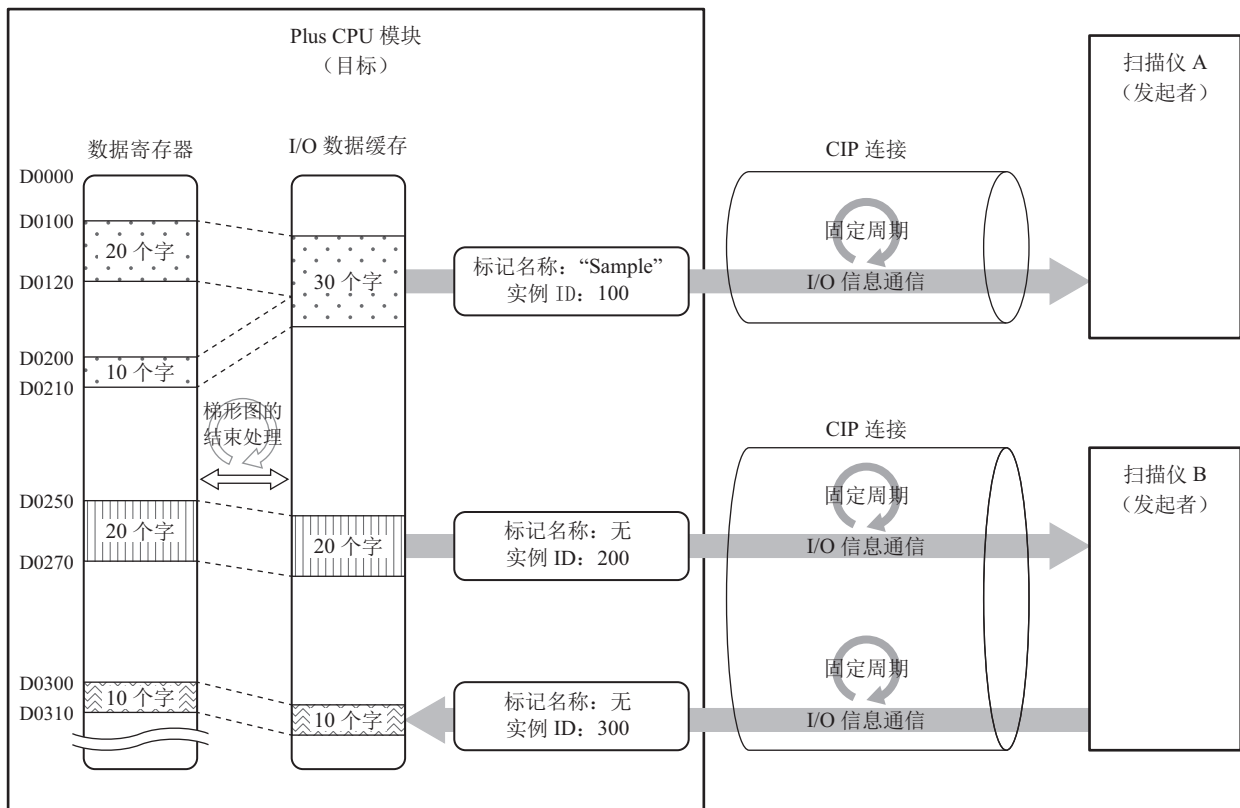
16: ETHERNET/IP通信

在未设立 Multicast 类型 CIP 连接的状态下，无法设立 Listen Only 的 CIP 连接。此外，已设立了使用 Exclusive Owner 和 Input Only 的 Multicast 类型 CIP 连接的发起者和目标的 CIP 连接切断后，停止从目标向已设立 Listen Only 的 CIP 连接的发起者进行多点传输发送。



I/O 信息通信（目标）功能

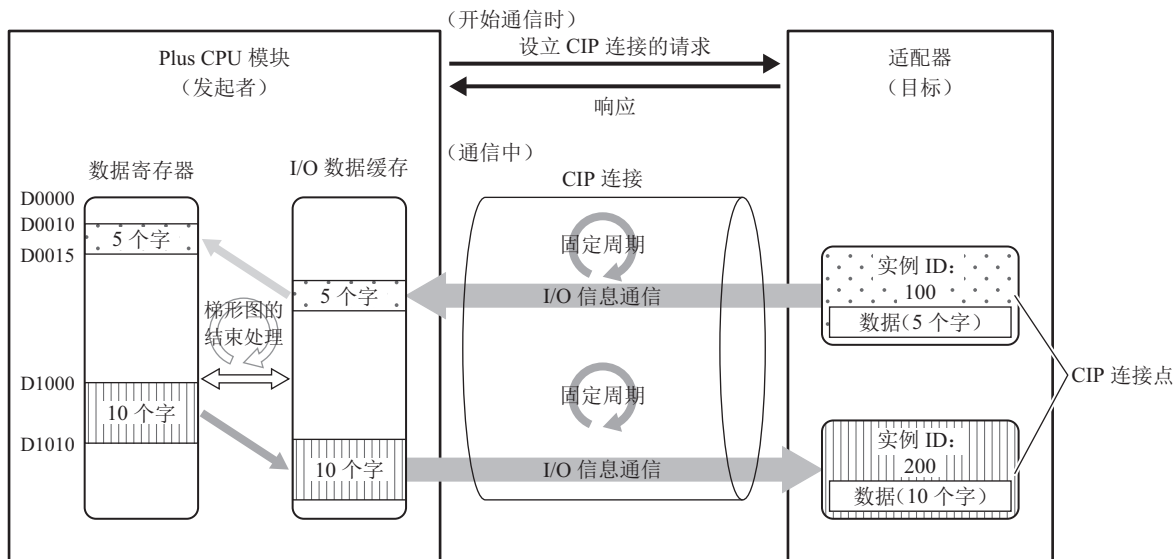
Plus CPU 模块作为目标拥有 CIP 连接点，发起者周期性读写数据的功能。可定义的 CIP 连接点最大为 8 个。发起者向 Plus CPU 模块的 CIP 连接点请求设立 CIP 连接，I/O 信息通信中，成功后读写 CIP 连接点的数据。Plus CPU 模块将数据寄存器的值写入 I/O 数据缓存*¹后发送至发起者。此外，将接收自发起者的数据写入 I/O 数据缓存后存储在数据寄存器中。



*1 I/O 数据缓存是 Plus CPU 模块内部拥有的按照 I/O 信息通信存储收发数据的缓存。有关 I/O 数据缓存和数据寄存器联动的详情，请参见第 16-11 页上的“数据与设备的联动功能”。

I/O 信息通信（发起者）功能

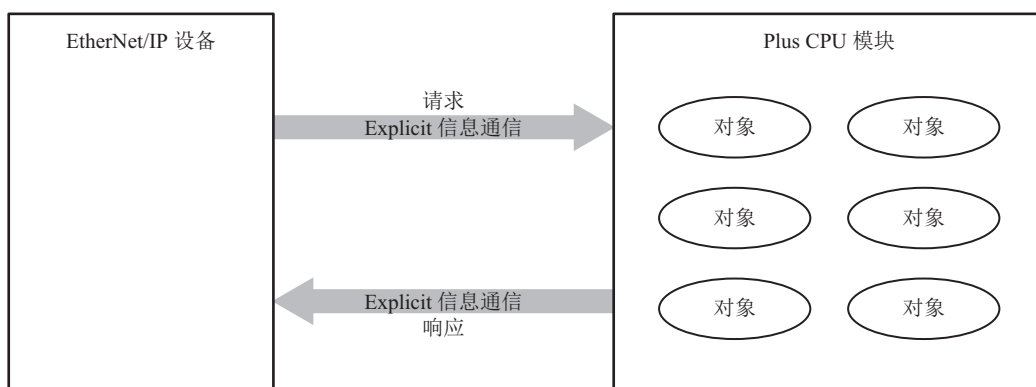
Plus CPU 模块对目标的 CIP 连接点的数据进行周期性读写的功能。打开 EtherNet/IP 通信许可（M8460）后，Plus CPU 模块向已在扫描列表*1 中注册的目标的 CIP 连接点请求设立 CIP 连接，成功后*2 读写 CIP 连接点的数据。Plus CPU 模块将接收自目标的数据写入 Plus CPU 模块内部 I/O 数据缓存*3 后存储在数据寄存器中。此外，将数据寄存器的值写入 I/O 数据缓存后向目标发送。



- *1 扫描列表是发起者 Plus CPU 模块设立的 CIP 连接的一览。在“EtherNet/IP 设置”对话框的 EtherNet/IP 树形区域，显示已设置的目标或 CIP 连接的一览。有关详情，请参见第 16-20 页上的“EtherNet/IP 树形区域”。
- *2 设立 CIP 连接失败时，连续重试直到成功。在 I/O 信息通信中，经过指定超时时间仍无法从目标接收 I/O 信息时，自动重新设立新的 CIP 连接。设立新的 CIP 连接失败时，连续重试直到成功。
- *3 I/O 数据缓存是 Plus CPU 模块内部拥有的按照 I/O 信息通信存储收发数据的缓存。有关 I/O 数据缓存和数据寄存器联动，请参见第 16-11 页上的“数据与设备的联动功能”。

Explicit 信息通信（服务器）功能

该功能用来从其他 EtherNet/IP 设备执行 Plus CPU 模块的对象*1 支持的服务。

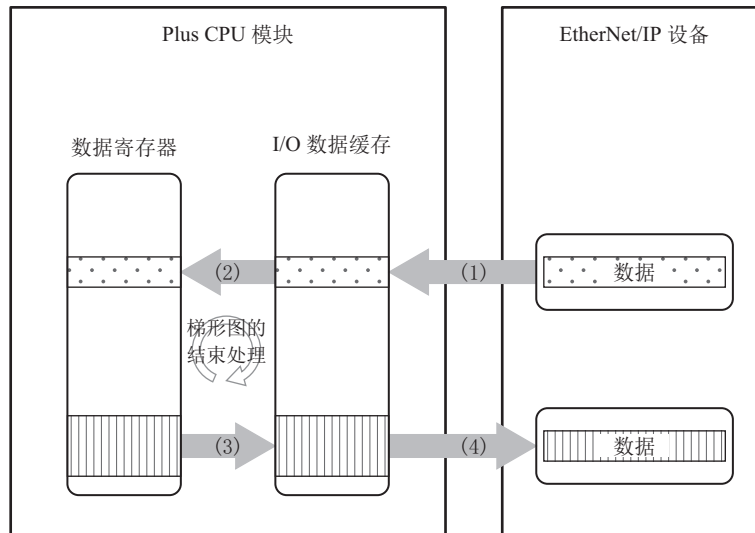


- *1 对象抽象表现 EtherNet/IP 设备的功能和数据。

数据与设备的联动功能

Plus CPU 模块将扫描列表和 CIP 连接点设置的 IN 数据*1 和 OUT 数据*2 分配到 Plus CPU 模块内部的 I/O 数据缓存中。

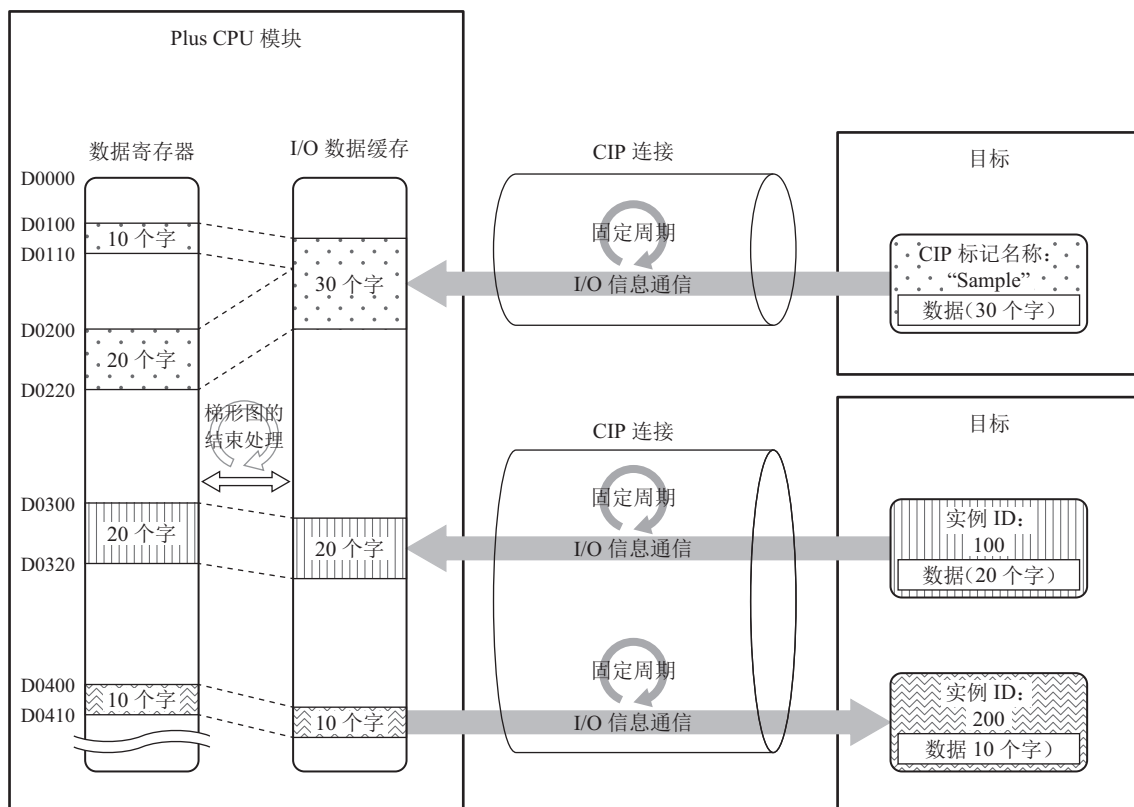
Plus CPU 模块运用 I/O 信息通信将接收自目标设备的数据，暂时写入 I/O 数据缓存 (1)，然后存储到数据寄存器中 (2)。此外，将存储在数据寄存器的值暂时写入 I/O 数据缓存 (3)，然后向目标设备发送 (4)。相互反映 I/O 数据缓存与数据寄存器的处理，通过梯形图程序的结束处理来执行。



*1 IN 数据是发起者使用 I/O 信息通信从目标接收的数据。

*2 OUT 数据是发起者使用 I/O 信息通信向目标发送的数据。

例如，Plus CPU 模块（发起者）周期性读写目标 CIP 连接点的数据时，接收自目标的数据暂时写入 I/O 数据缓存，然后存储到数据寄存器。此外，数据寄存器的值写入 I/O 数据缓存后向目标发送。



■ 刷新上限数

1次结束处理可以相互反映 I/O 数据缓存与数据寄存器的大小最大为 400 个字。可在 1 ~ 400 个字范围内将该大小设置为刷新上限数。例如，将刷新上限数设置为 400 个字时，1次结束处理最大可从 I/O 数据缓存向数据寄存器反映 256 个字，最大可从数据寄存器向 I/O 数据缓存反映 256 个字。设置超出刷新上限数的 IN 数据和 OUT 数据时，拆分为多个结束处理执行。增加刷新上限数后，由于结束处理反映的数据量增多，梯形图程序的扫描时间延长。请根据目标的系统进行调整。

注释：高频次执行 I/O 信息通信。接收自 EtherNet/IP 设备的值，以比扫描时间还短的间隔变化时，该变化可能无法反映到数据寄存器中。

EtherNet/IP 通信的动作

EtherNet/IP 通信使用的特殊设备

EtherNet/IP 通信使用的特殊数据寄存器及特殊内部继电器如下所示。

特殊数据寄存器	说明		读 / 写
D8790	EtherNet/IP 运行状态	存储 EtherNet/IP 通信的动作状态。 0000h: 停止中 0100h: 联机准备中 0200h: 联机中 0300h: 重新启动中 0F00h: 错误停止中	读
D8791	EtherNet/IP 错误信息	存储 EtherNet/IP 通信中发生的错误。 0000h: 正常 0800h: 等待连接超时 8000h: 等待通信开始超时 8100h: 等待通信停止超时 EtherNet/IP 通信在联机准备中 ((D8790)=0100h) 时复位为 0。 EtherNet/IP 通信中发生多个错误时, 存储最后发生的错误信息。	读
特殊内部继电器	说明		读 / 写
M8460	EtherNet/IP 通信许可	启用 / 禁用 EtherNet/IP 通信。 OFF: 禁用 EtherNet/IP 通信 ON: 启用 EtherNet/IP 通信 RUN 状态下, 将 M8460 从 OFF 设为 ON 后, Plus CPU 模块会确立与目标设备的 CIP 连接, 开始 EtherNet/IP 通信。将 M8460 设为 OFF 后, 会切断 CIP 连接, 停止 EtherNet/IP 通信。RUN → STOP 时, M8460 自动变为 OFF。	读 / 写

注释:

- 若需在 EtherNet/IP 运行状态 (D8790) 为错误停止中的状态下重新开始 EtherNet/IP 通信, 请暂时将 M8460 切换到 OFF, 等待 EtherNet/IP 运行状态 (D8790) 变为停止中后, 再将 M8460 切换到 ON。
- Plus CPU 模块的电源 ON 时以及 RUN 开始时, EtherNet/IP 通信许可 (M8460) OFF。若要开始 EtherNet/IP 通信, 请使用梯形图程序将 M8460 设为 ON, 或者将已选中“EtherNet/IP 设置”对话框中“自动打开 EtherNet/IP 通信位”复选框的用户程序下载后运行。

基本动作

本节将对 EtherNet/IP 通信的基本动作进行介绍。

■ EtherNet/IP 通信的开始和停止

RUN 状态下，将 EtherNet/IP 通信许可（M8460）从 OFF 设为 ON 后，Plus CPU 模块与已注册到扫描列表的目标或 CIP 连接点确立 CIP 连接，开始 EtherNet/IP 通信（开始通过 I/O 信息通信进行周期性数据交换）。

将 EtherNet/IP 通信许可（M8460）设为 OFF 后，会停止 EtherNet/IP 通信（停止 I/O 信息通信，切断所有 CIP 连接）。RUN → STOP 后，EtherNet/IP 通信许可（M8460）自动变为 OFF。

■ EtherNet/IP 通信中的动作

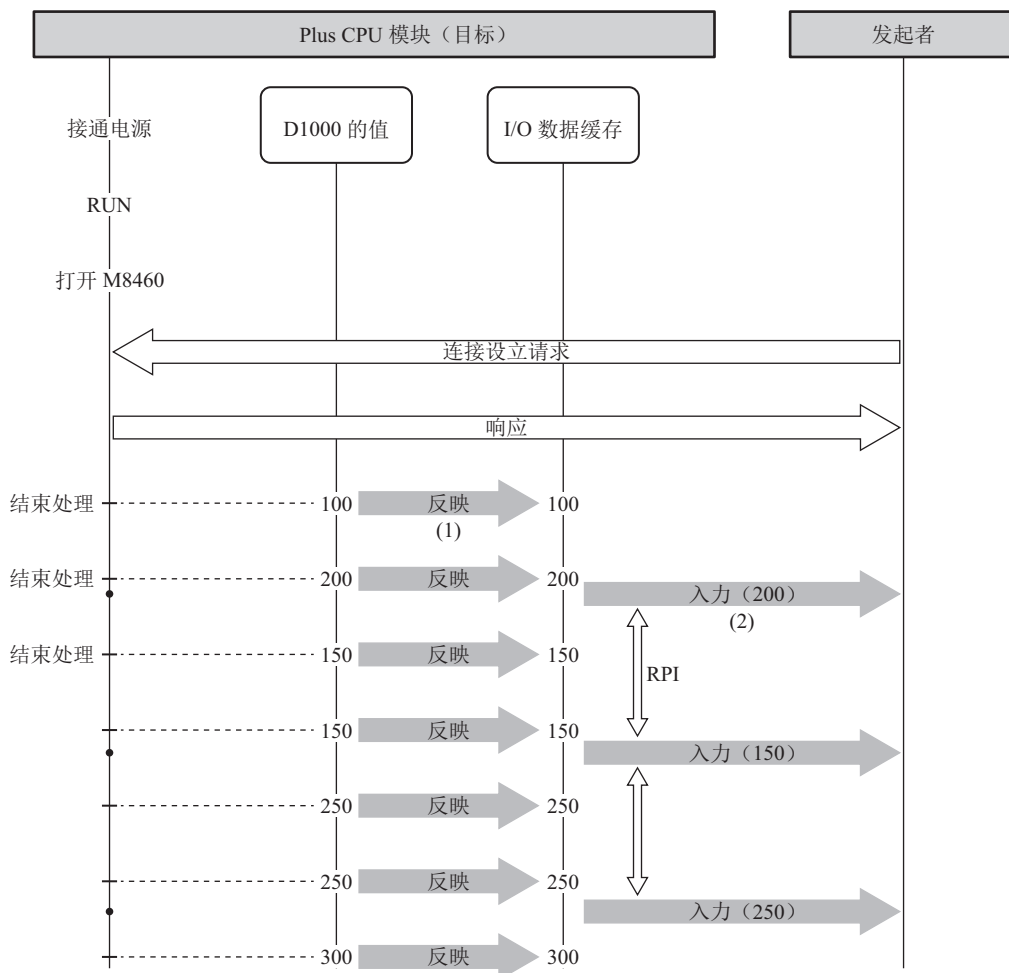
Plus CPU 模块拥有的 I/O 数据缓存对应数据寄存器，通过梯形图程序的结束处理相互反映数据。

相互反映 I/O 数据缓存与数据寄存器的处理，不与 Plus CPU 模块和 EtherNet/IP 设备间的 I/O 信息通信同步进行。

注释：由于高频次执行 I/O 信息通信，接收自 EtherNet/IP 设备的值，以比扫描时间还短的间隔变化时，该变化可能无法反映到数据寄存器中。

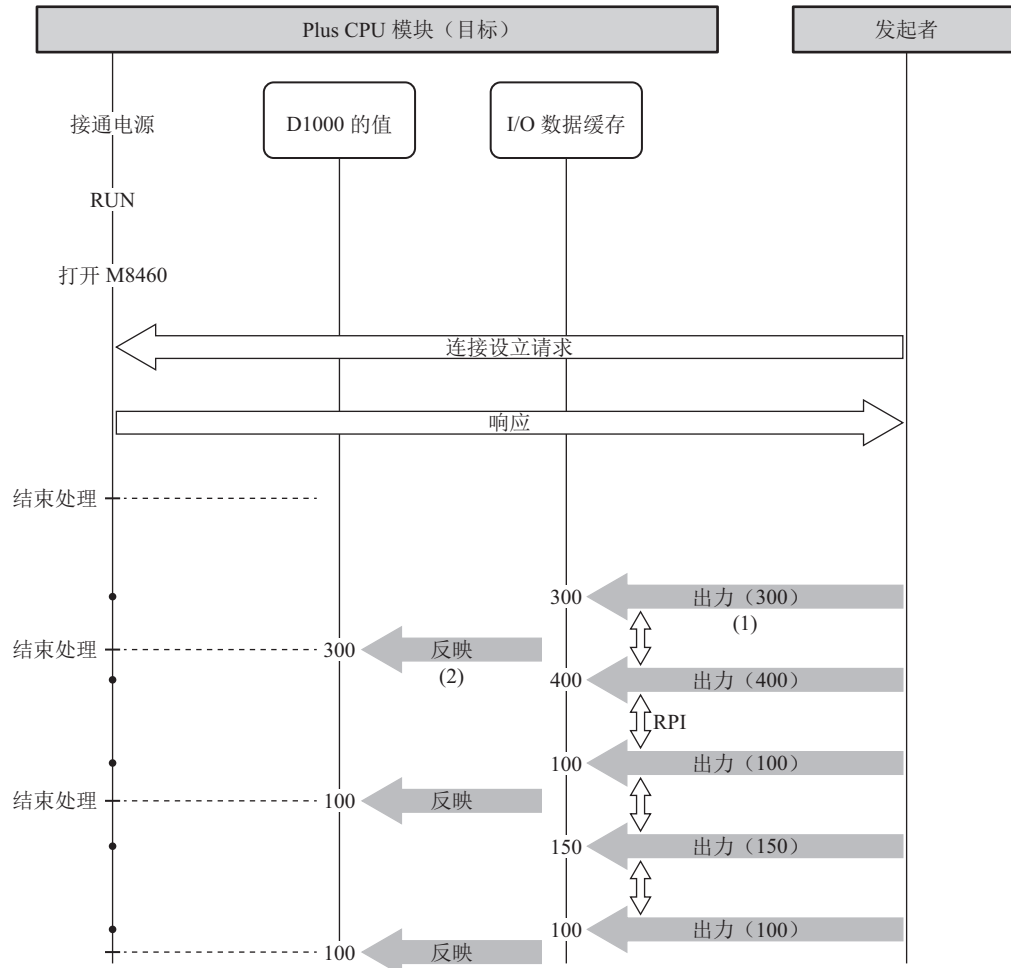
Plus CPU 模块（目标）向发起者发送（输入）数据时

例如，向发起者发送存储在 D1000 中的值时，通过数据发送前的梯形图程序的结束处理，将 D1000 中存储的值反映到 I/O 数据缓存中 (1)。然后，通过之后的 I/O 信息通信向目标发送 I/O 数据缓存的数据 (2)。



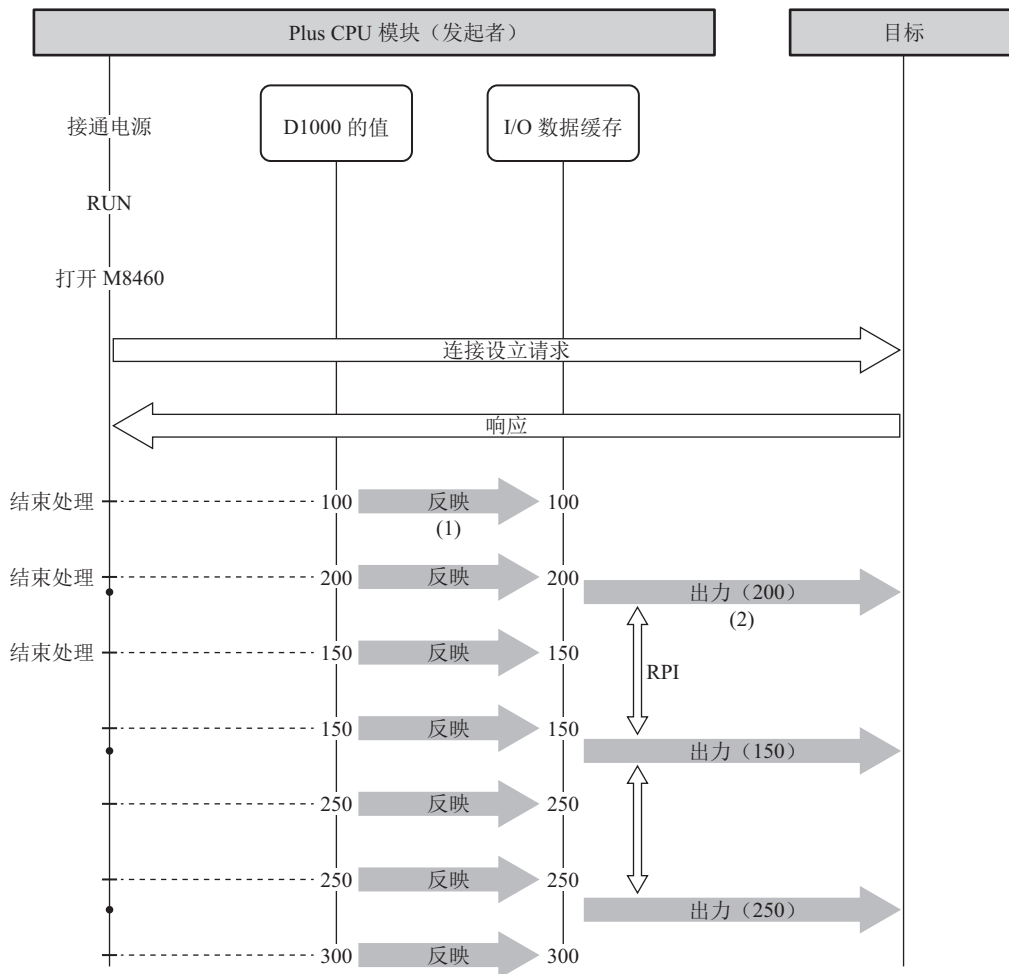
Plus CPU 模块（目标）接收（输出）来自发起者的数据时

例如，将接收自发起者的数据存储到 D1000 时，Plus CPU 模块运用 I/O 信息通信将接收自发起者的数据，暂时写入 I/O 数据缓存中 (1)。然后，通过接收数据后的梯形图程序的结束处理，将 I/O 数据缓存的数据反映到数据寄存器中 (2)。



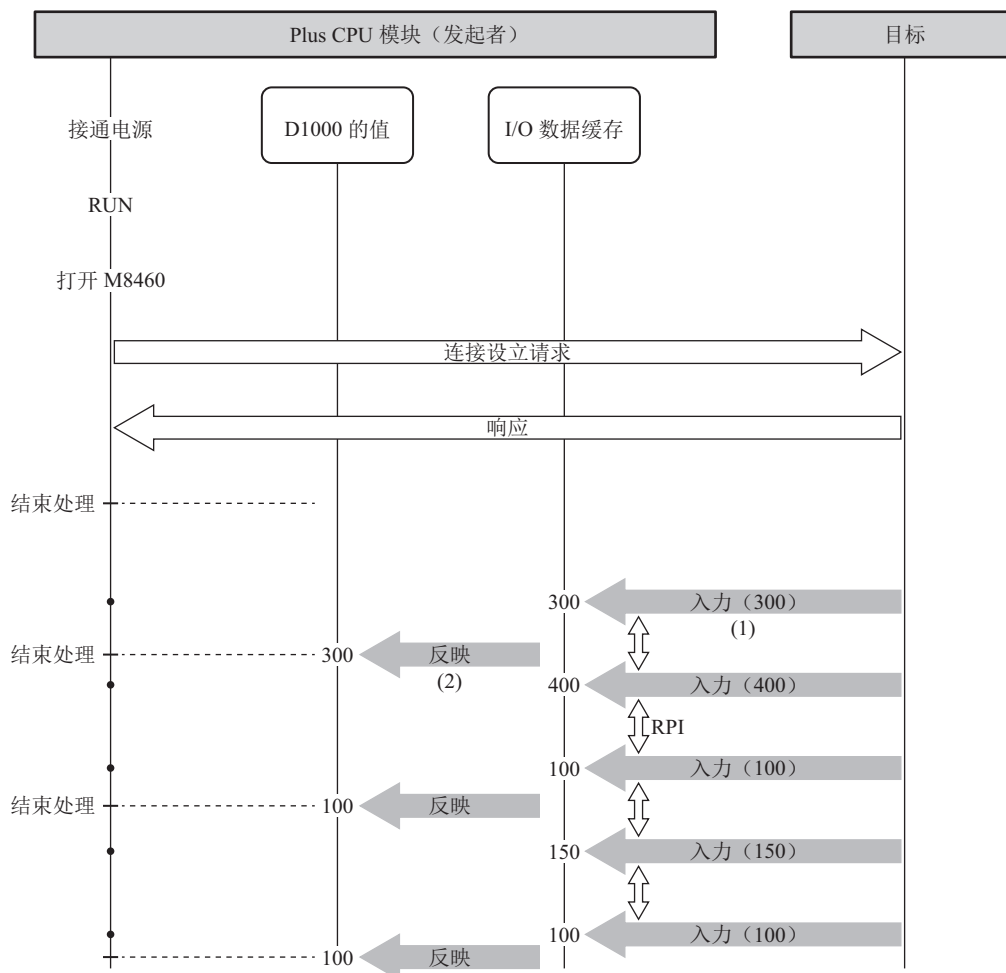
Plus CPU 模块（发起者）向目标发送（输出）数据时

例如，向目标发送存储在 D1000 中的值时，通过数据发送前的梯形图程序的结束处理，将 D1000 中存储的值反映到 I/O 数据缓存中 (1)。然后，通过之后的 I/O 信息通信向目标发送 I/O 数据缓存的数据 (2)。



Plus CPU 模块（发起者）从目标接收（输入）数据时

例如，将接收自目标的数据存储到 D1000 时，Plus CPU 模块运用 I/O 信息通信将接收自目标的数据，暂时写入 I/O 数据缓存中 (1)。然后，通过接收数据后的梯形图程序的结束处理，将 I/O 数据缓存的数据反映到数据寄存器中 (2)。



RUN 及 STOP 中的动作

显示 RUN 和 STOP 中的 EtherNet/IP 通信的状态。

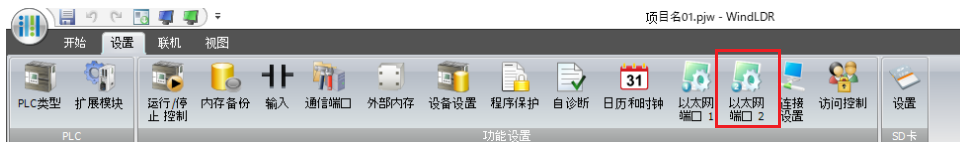
RUN/STOP 状态	M8460 的状态	EtherNet/IP 通信
STOP	OFF	停止
	ON	
RUN	OFF	开始
	打开时间	通信中
	ON	通信中
	关闭时间	结束

EtherNet/IP 通信的设置

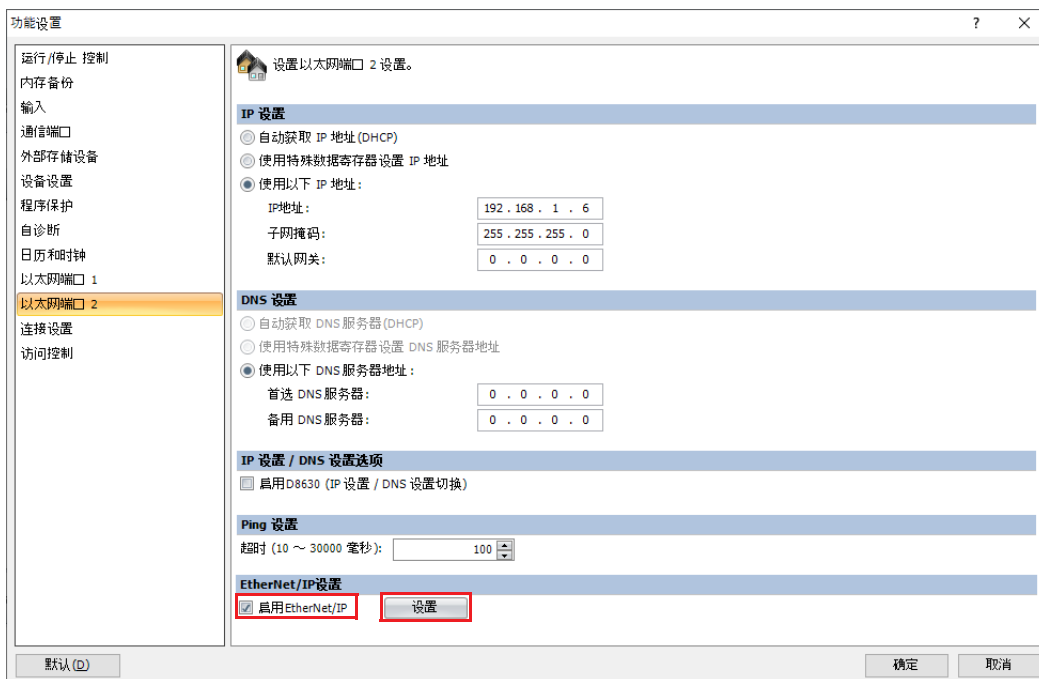
在“EtherNet/IP 设置”对话框中执行 EtherNet/IP 通信的设置。本节将对“EtherNet/IP 设置”选项卡和对应使用方法的设置进行介绍。

● “EtherNet/IP 设置”对话框的显示步骤

1. 在“设置”选项卡的“功能设置”中单击“以太网端口 2”。
出现“功能设置”对话框。



2. 在 EtherNet/IP 设置中选中“启用 EtherNet/IP”复选框，然后单击“设置”按钮。
将显示“EtherNet/IP 设置”对话框。



EtherNet/IP 设置的概要

与 EtherNet/IP 通信所需设置相关的 WindLDR 的设置对话框如下所示。

说明	所需设置	WindLDR 的设置对话框	可用功能
执行 EtherNet/IP 通信	EtherNet/IP 通信的基本设置	EtherNet/IP 设置	EtherNet/IP 通信通用
将 Plus CPU 模块 用作目标	CIP 连接点的设置 • CIP 标记名称或实例 ID 的设置 • 设备的分配	CIP 连接点设置	I/O 信息通信（目标）功能
将 Plus CPU 模块 用作发起者	扫描列表的创建 • 目标的设置 • CIP 连接的设置	目标设置 CIP 连接设置	I/O 信息通信（发起者）功能

“EtherNet/IP 设置”对话框

“EtherNet/IP 设置”对话框由以下 4 个区域构成。

(1) EtherNet/IP 树形区域 (第 16-20 页)

显示已设置的 CIP 连接点、目标及 CIP 连接的一览。

(2) 参数设置区域 (第 16-22 页)

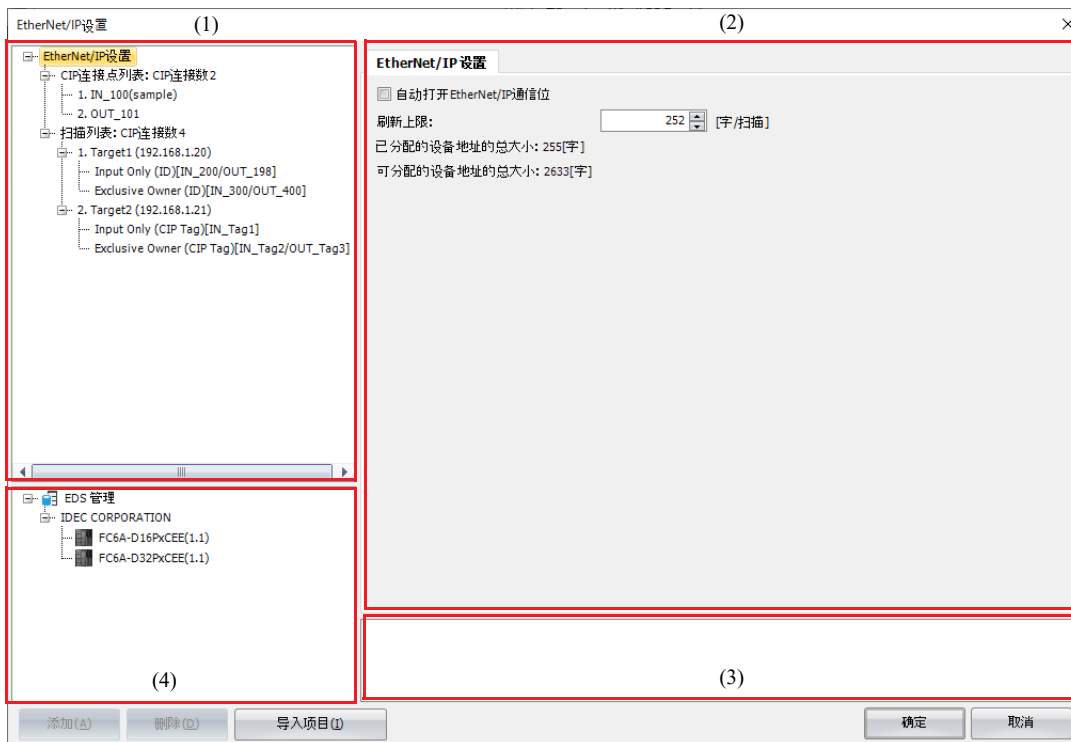
显示 EtherNet/IP 树形区域中已选节点的详情。

(3) 信息显示区域 (第 16-30 页)

“EtherNet/IP 设置”对话框中设置的内容存在错误时，显示错误内容。

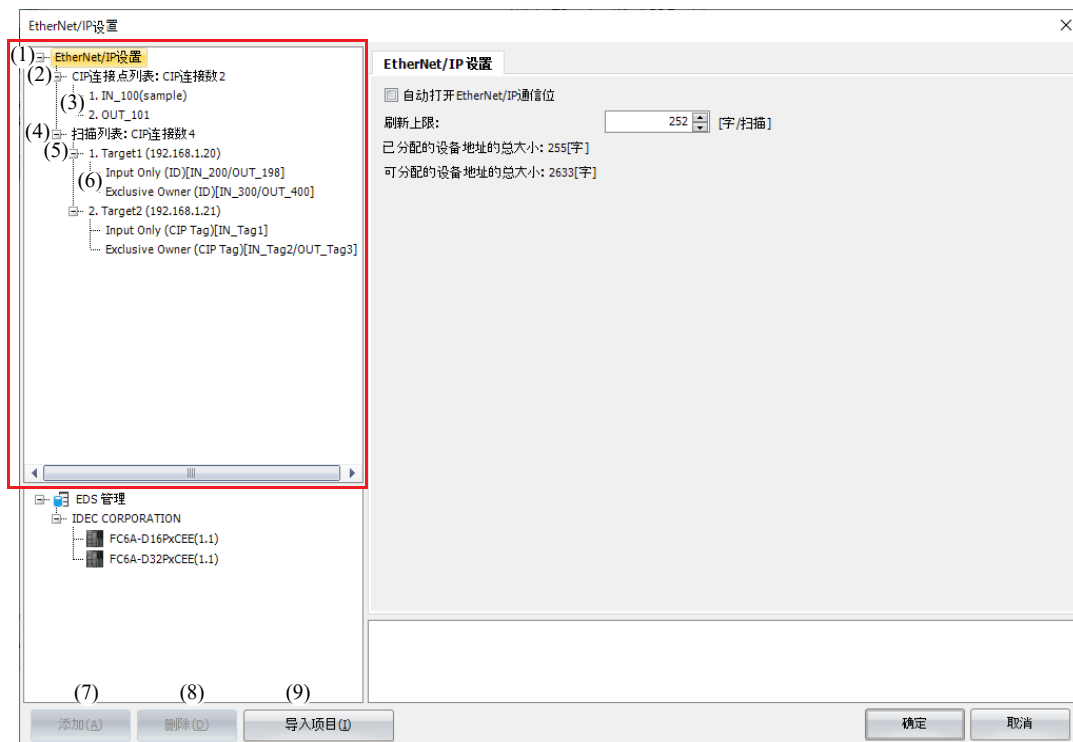
(4) EDS 管理区域 (第 16-31 页)

显示被导入 PC 中的 EDS 文件一览。



EtherNet/IP 树形区域

显示已设置的 CIP 连接点、目标及 CIP 连接的一览。



(1) “EtherNet/IP 设置” 节点

选择并展开“EtherNet/IP 设置”节点，以查看已配置的 CIP 连接点和扫描列表的一览。

(2) “CIP 连接点列表” 节点

显示 CIP 连接点数。展开“CIP 连接点列表”节点后，显示已设置的 CIP 连接点的一览。

(3) “CIP 连接点” 节点

显示已设置的 CIP 连接点的 CIP 标记名称以及实例 ID。

(4) “扫描列表” 节点

显示 CIP 连接数。展开“扫描列表”节点后，显示已设置的目标的一览。

(5) “目标” 节点

显示已设置的目标的节点名称和 IP 地址。选择并展开各“目标”节点后，显示已设置的 CIP 连接的一览。

(6) “CIP 连接” 节点

显示已设置的 CIP 连接的 CIP 连接名称以及 CIP 连接点。

(7) “添加” 按钮

对应 EtherNet/IP 树形区域中已选节点，新增节点。也可从右击节点时显示的菜单中添加。

(8) “删除” 按钮

删除 EtherNet/IP 树形区域中已选节点。也可从右击节点时显示的菜单中删除。

(9) “导入项目” 按钮

从 WindLDR 的项目文件中仅导入 EtherNet/IP 设置。

本节介绍通过右键单击 EtherNet/IP 树形区域中的每个节点显示的上下文菜单的操作。

节点	上下文菜单	操作
EtherNet/IP 设置	—	—
CIP 连接点列表	添加 CIP 连接点设置	添加 1 个 CIP 连接点。
	删除所有 CIP 连接点设置	删除全部 CIP 连接点。
CIP 连接点	添加 CIP 连接点设置	添加 1 个 CIP 连接点。
	删除所选 CIP 连接点设置	删除已选的 CIP 连接点。
扫描列表	添加目标	添加一个目标
	删除所有目标	删除全部目标。
目标	添加 CIP 连接	添加 1 个 CIP 连接。
	删除所选目标	删除已选目标。
	将 EDS 文件导入“EDS 管理”	将已选目标的 EDS 文件导入 EDS 管理区域。如导入 EDS 文件，即在其他项目中也可使用。
CIP 连接	添加 CIP 连接	添加 1 个 CIP 连接。
	删除所选 CIP 连接	删除已选的 CIP 连接。

参数设置区域

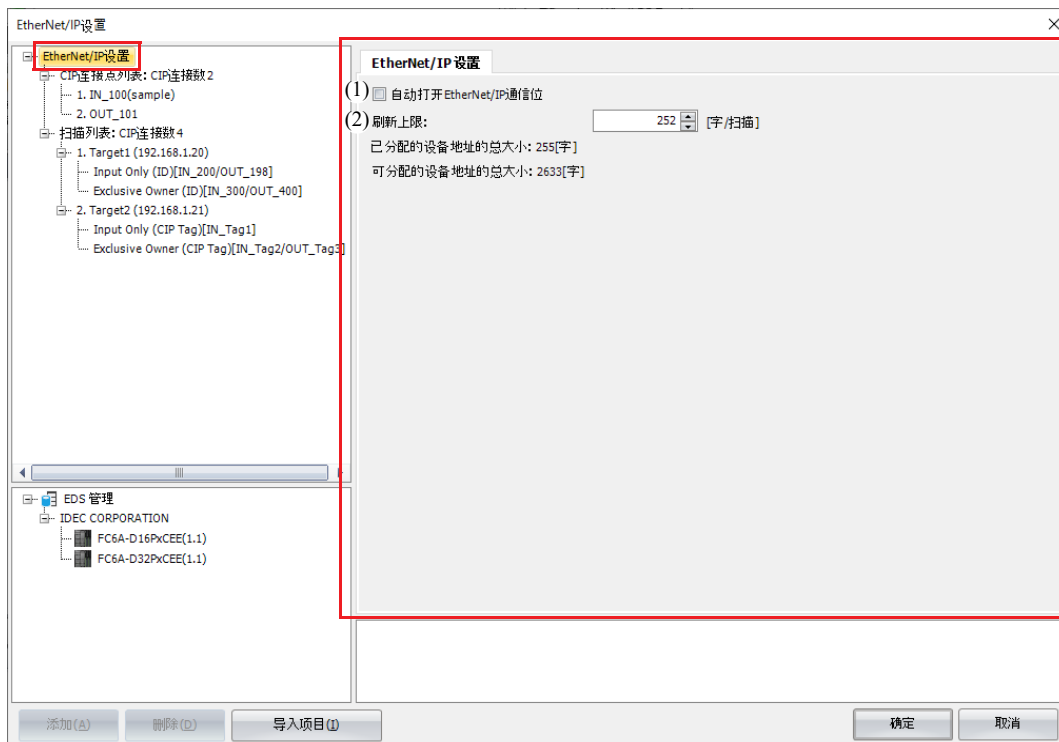
显示 EtherNet/IP 树形区域中已选节点的详情。

已选择“Ethernet Net/IP 设置”节点时

选择 EtherNet/IP 树形区域的“Ethernet Net/IP 设置”节点后，在参数设置区域上显示“Ethernet/IP 设置”选项卡。

■ “Ethernet/IP 设置”选项卡

执行用于 EtherNet/IP 通信的基本设置。



(1) 自动打开 EtherNet/IP 通信位

设置 STOP → RUN 时是否自动将 EtherNet/IP 通信许可（M8460）切换到 ON。默认值为关闭。关闭时，不会自动将 EtherNet/IP 通信许可切换到 ON。

设置方法	说明
开启	STOP → RUN 时自动将 EtherNet/IP 通信许可（M8460）切换到 ON。 打开 EtherNet/IP 通信许可（M8460）后，Plus CPU 模块根据扫描列表中注册的内容与目标设备进行 EtherNet/IP 通信。
关闭	不会自动将 EtherNet/IP 通信许可（M8460）切换到 ON。

(2) 刷新上限

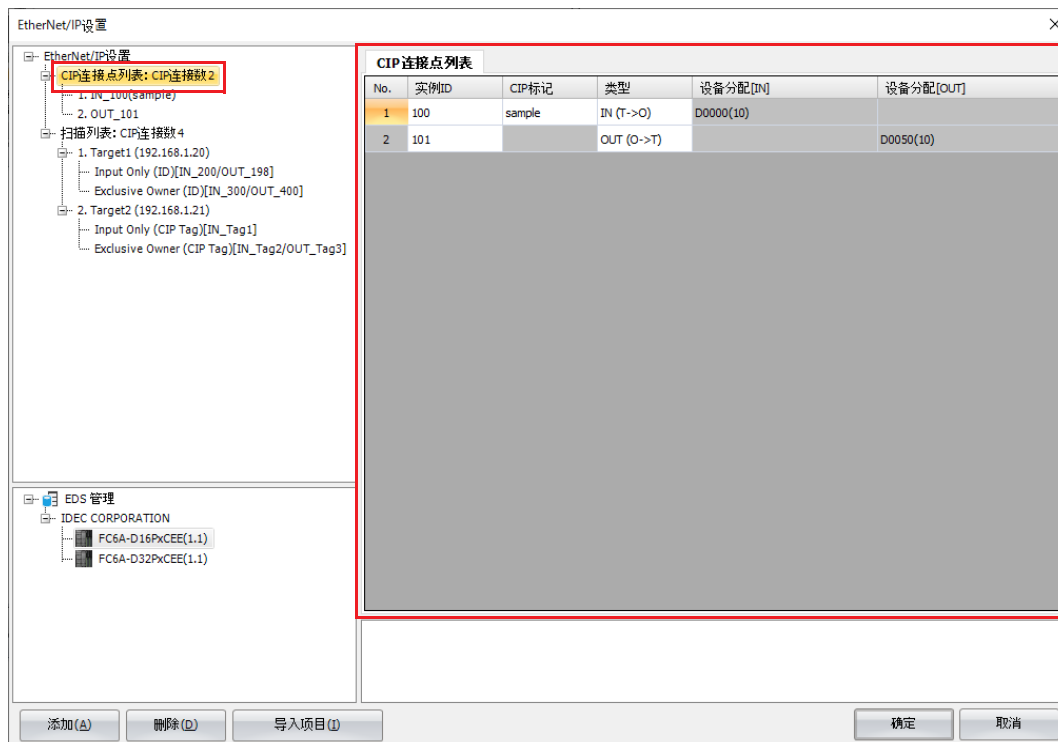
设置 1 次结束处理可以相互反映 I/O 数据缓存与数据寄存器的最大数据大小。在 1 ~ 400 个字 / 扫描的范围内设置刷新上限。默认值为 252 个字 / 扫描。1 次结束处理从 I/O 数据缓存写入数据寄存器的最大数据大小，以及从数据寄存器写入 I/O 数据缓存的最大数据大小，均为刷新上限的一半。

已选择“CIP 连接点列表”节点时

选择 EtherNet/IP 树形区域的“CIP 连接点列表”节点后，在参数设置区域上显示“CIP 连接点列表”选项卡。

■ “CIP 连接点列表”选项卡

显示已设置的 CIP 连接点的一览。可以变更 CIP 标记、实例 ID 以及类型。双击设备分配的单元后，显示“CIP 连接点列表”选项卡，可移动至设备分配的相应单元。

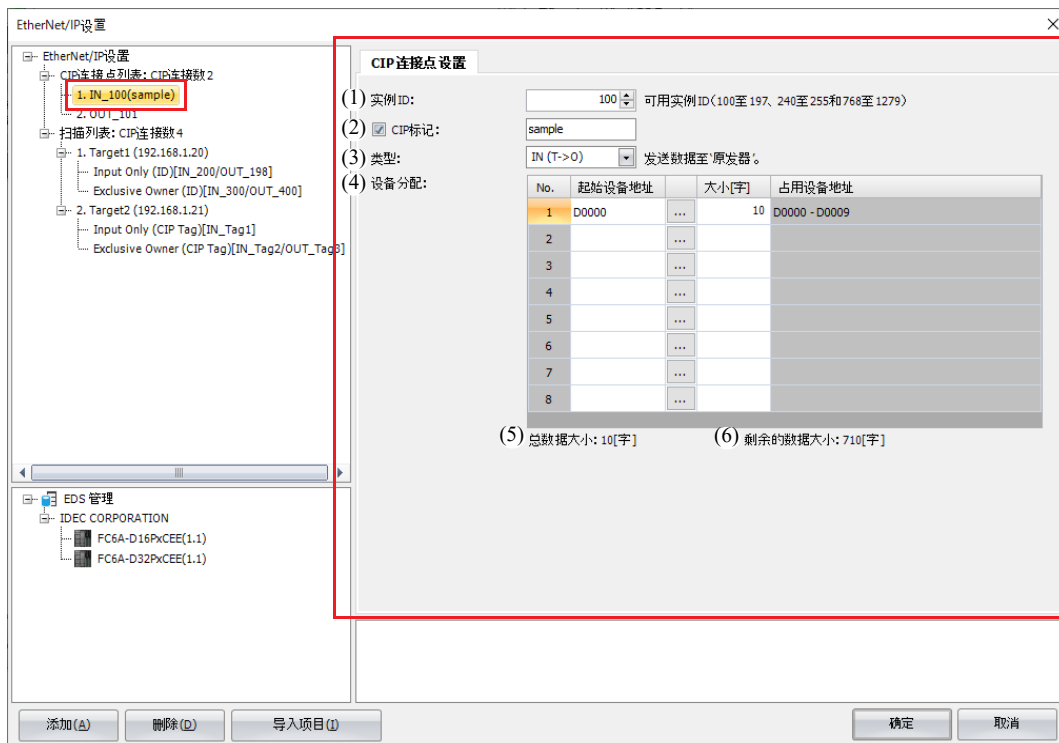


已选择“CIP 连接点”节点时

选择 EtherNet/IP 树形区域的“CIP 连接点”节点后，在参数设置区域上显示“CIP 连接点设置”选项卡。

■ “CIP 连接点设置”选项卡

设置 CIP 连接点。



(1) 实例 ID

设置实例 ID。Plus CPU 模块（目标）与无法处理 CIP 标记名称的其他扫描仪设备（发起者）进行 I/O 信息通信时，发起者对实例 ID 设置 CIP 连接。在 100 ~ 1279 的范围内设置实例 ID。默认值为 100。请设置与其他 CIP 连接不重复的值。

注释： 100 ~ 1279 范围内存在无法使用的实例 ID。请设置在 WindLDR 中显示的可用实例 ID 内的值。

(2) CIP 标记

设置 CIP 标记名称。最多可设置 64 字节。其他扫描仪设备（发起者）对该标记名称设置 CIP 连接，与 Plus CPU 模块（目标）进行 I/O 信息通信。选中复选框后，可设置 CIP 标记。

(3) 类型

设置 I/O 信息通信处理的数据的方向。可通过如下 2 种方式进行设置。默认值为“IN(T → O)”。

设置	说明
IN(T → O)	目标（Plus CPU 模块）向发起者发送数据。
OUT(O → T)	目标（Plus CPU 模块）从发起者接收数据。

(4) 设备分配

设置数据寄存器用于存储 I/O 信息通信中接收或发送的数据。从设置的数据寄存器开始，按照设置的数据大小使用对应的数据寄存器。在 1 ~ 720 的范围内设置数据大小。默认值为 1。请设置第一数据寄存器，不要超出设备的范围。

(5) 总数据大小

显示 I/O 信息通信中使用的数据寄存器的合计值。

(6) 剩余的数据大小

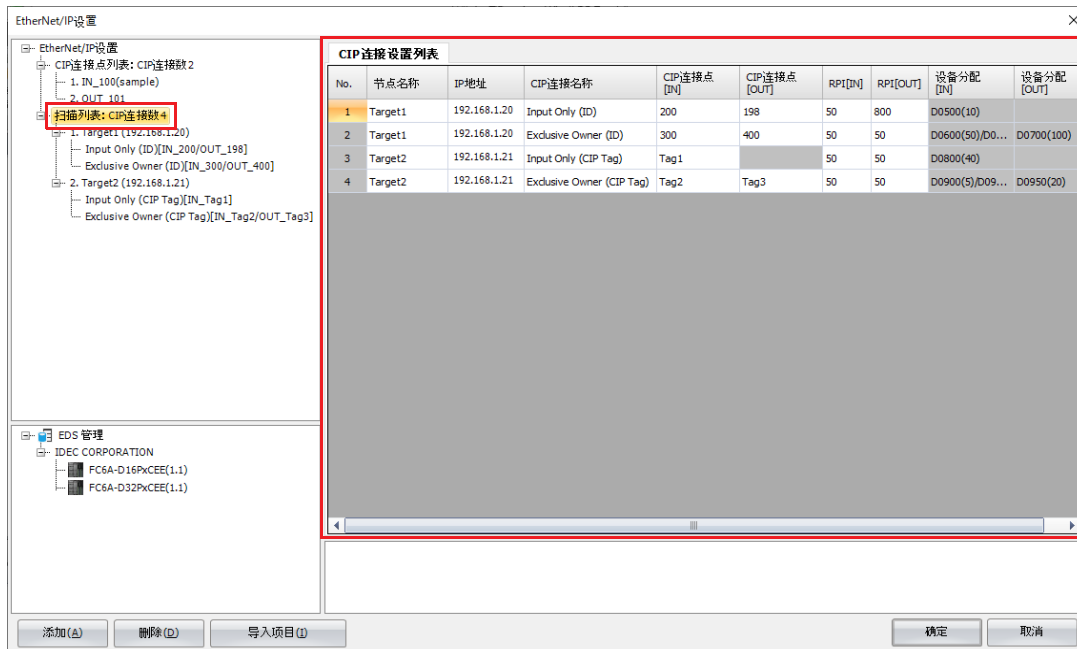
显示可用剩余数据大小（从 720 字节中减去使用的数据寄存器的合计值）。

已选择“扫描列表”节点时

选择 EtherNet/IP 树形区域的“扫描列表”节点后，在参数设置区域上显示“CIP 连接设置列表”选项卡。

■ “CIP 连接设置列表”选项卡

显示已设置的 CIP 连接的一览。可变更节点名称、IP 地址、CIP 连接名称、CIP 连接点以及 RPI。双击设备分配的单元后，显示“CIP 连接设置”选项卡，可移动至设备分配的相应单元。

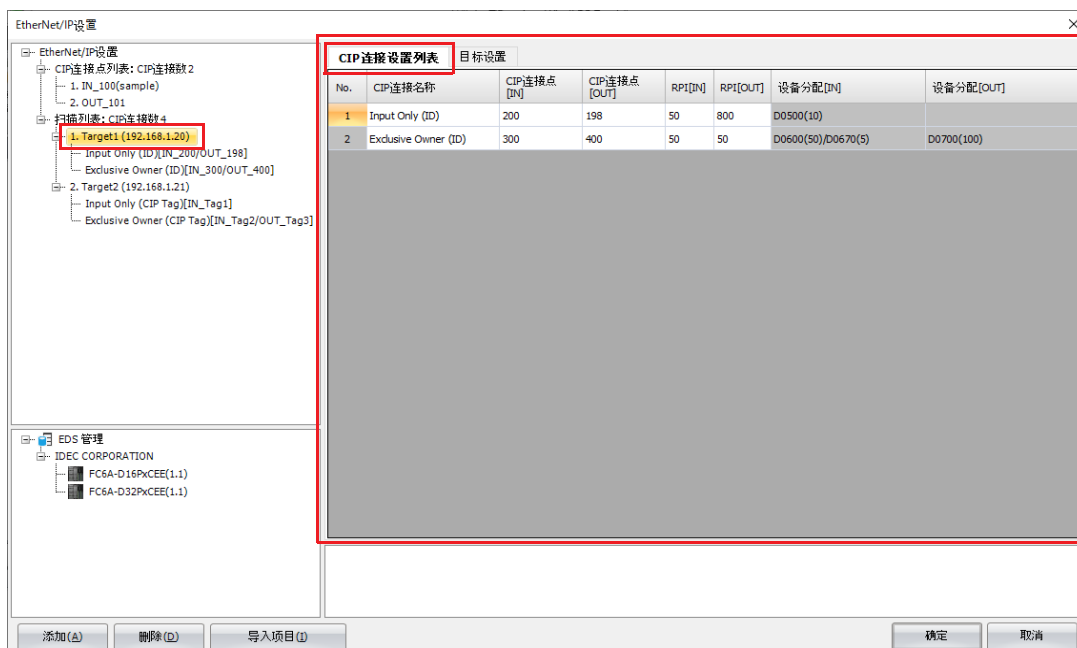


已选择“目标”节点时

选择 EtherNet/IP 树形区域的“目标”节点后，在参数设置区域上显示“CIP 连接设置列表”选项卡和“目标设置”选项卡。

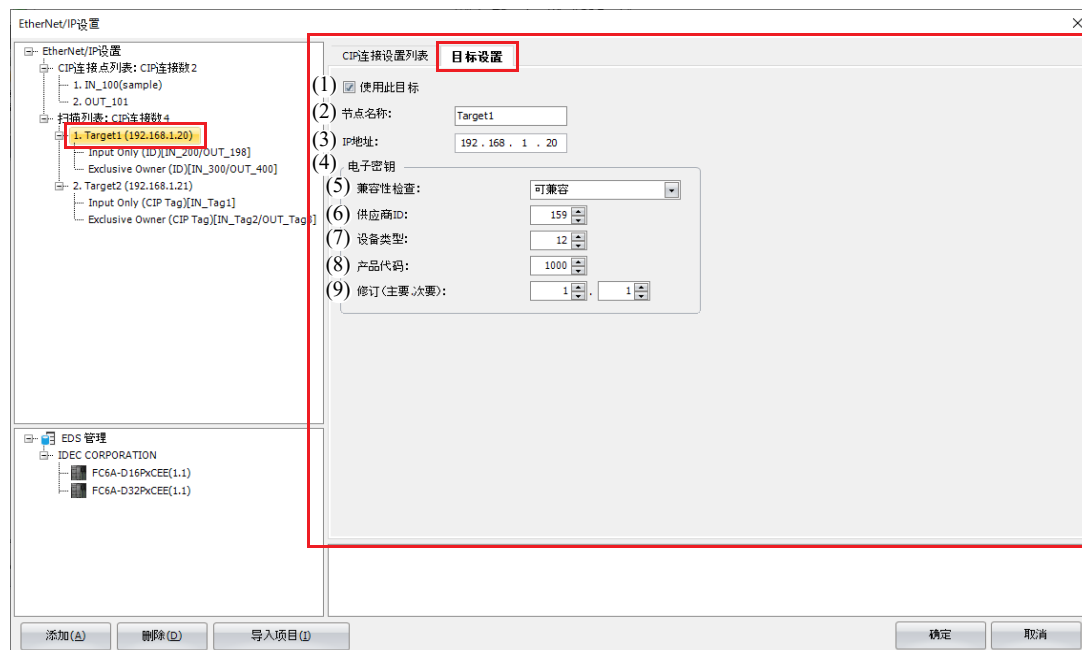
■ “CIP 连接设置列表”选项卡

显示已设置的目标的 CIP 连接一览。可在该选项卡中变更 CIP 连接名称、CIP 连接点以及 RPI（通信周期）。



■ “目标设置”选项卡

设置目标设备。



(1) 使用此目标

将扫描列表的 CIP 连接以目标单位设置为启用 / 禁用。启用 CIP 连接后，EtherNet/IP 通信许可（M8460）打开时，Plus CPU 模块与目标进行 I/O 信息通信。禁用 CIP 连接后，Plus CPU 模块不执行 I/O 信息通信。启用 CIP 连接时，选中复选框。

(2) 节点名称

设置目标的节点名称。在扫描列表中显示设置的节点名称。最多可设置 30 字节。

(3) IP 地址

设置目标的 IP 地址。

(4) 电子密钥

设立 CIP 连接时，根据兼容性检查的设置，检查扫描列表上注册的目标与进行实际连接的 EtherNet/IP 设备是否一致。仅许可与核对后一致的 EtherNet/IP 设备进行 I/O 信息通信。不一致时，Plus CPU 模块不会与 EtherNet/IP 设备进行 I/O 信息通信。

(5) 兼容性检查

可通过如下 3 种方式设置兼容性检查的方法。

兼容性检查	说明
禁用	不核对电子密钥。
兼容	仅许可与满足以下所有条件的目标进行 I/O 信息通信。 <ul style="list-style-type: none"> 设置与实际设备的供应商 ID、设备类型以及产品代码一致 实际设备的主要修订和次要修订大于设置
完全一致	仅与设置的电子密钥完全一致的目标进行 I/O 信息通信。

(6) 供应商 ID

设置 EtherNet/IP 设备的供应商 ID。在 0 ~ 65535 范围内进行设置。

(7) 设备类型

设置 EtherNet/IP 设备的设备类型。在 0 ~ 65535 范围内进行设置。

(8) 产品代码

设置 EtherNet/IP 设备的产品代码。在 0 ~ 65535 范围内进行设置。

(9) 修订（主要、次要）

设置 EtherNet/IP 设备的主要修订和次要修订。

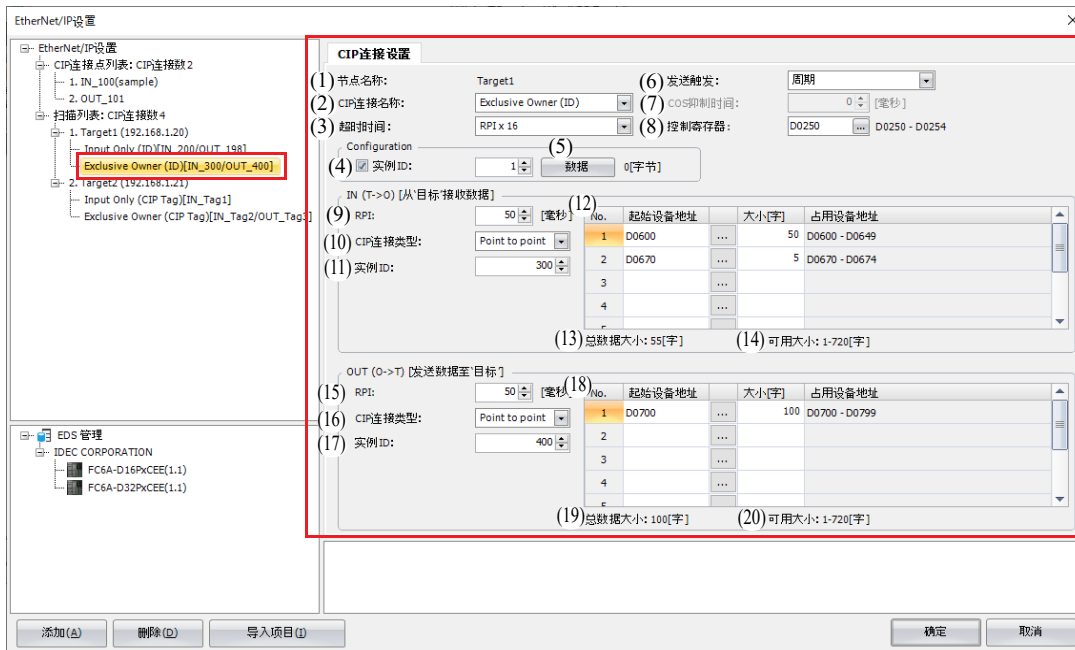
修订	设置范围
主要修订	0 ~ 127
次要修订	0 ~ 255

已选择“CIP 连接”节点时

选择 EtherNet/IP 树形区域的“CIP 连接”节点后，在参数设置区域上显示“CIP 连接设置”选项卡。

■“CIP 连接设置”选项卡

设置目标的 CIP 连接。



(1) 节点名称

显示“目标设置”选项卡中设置的节点名称。设置拥有该节点名称的目标的 CIP 连接。

(2) CIP 连接名称

设置拥有该节点名称 (1) 的目标的 CIP 连接名称。可通过如下 5 种方式进行设置。

设置	说明
Exclusive Owner (ID)	Plus CPU 模块与目标进行数据收发时进行设置。 Exclusive Owner (ID) 根据实例 ID 设置目标的 CIP 连接点。
Exclusive Owner (CIP Tag)	Exclusive Owner (CIP Tag) 根据 CIP 标记名称设置目标的 CIP 连接点。 Plus CPU 模块根据 CIP 标记名称或实例 ID 指定目标的 CIP 连接点，与目标进行 I/O 信息通信。
Input Only (ID)	仅在 Plus CPU 模块从目标接收数据时进行设置。 Input Only (ID) 根据实例 ID 设置目标的 CIP 连接点。
Input Only (CIP Tag)	Input Only (CIP Tag) 根据 CIP 标记名称设置目标的 CIP 连接点。 Plus CPU 模块根据 CIP 标记名称或实例 ID 指定目标的 CIP 连接点，与目标进行 I/O 信息通信。
Listen Only (ID)	其他发起者和目标使用 Exclusive Owner 和 Input Only 已设立 CIP 连接的情况下，该目标多点传输发送数据时，Plus CPU 模块在接收该数据时进行设置。 Listen Only (ID) 根据实例 ID 指定目标的 CIP 连接点。

注释： Plus CPU 模块无法根据 CIP 标记名称指定 Listen Only 的 CIP 连接。

(3) 超时时间

设置 I/O 信息通信的超时时间。Plus CPU 模块和目标互相监控接收数据的时间间隔。可设置范围为 RPI(9)、(15) 的整数倍 (4、8、16、32、64、128、256、512 倍)。默认值为 RPI 的 16 倍。

(4) 实例 ID

设置目标的 Configuration 的实例 ID。选中复选框后，可设置实例 ID。在 1 ~ 65535 的范围内设置实例 ID。默认值为 1。

16: ETHERNET/IP通信

(5) 数据

设置目标的 Configuration 用数据。Plus CPU 模块设立 CIP 连接时，与 Configuration 的实例 ID 一起发送到目标。根据目标的规格，可按照十六进制设置 0 ~ 400 字节的参数。

注释：在添加到扫描列表中的 Ethernet/IP 设备的 EDS 文件中定义了与 CIP 连接开设相关的参数信息时，单击 [数据] 按钮会显示该参数的列表。且这个参数可以更改。

(6) 发送触发

设置 Plus CPU 模块向目标发送数据的方法。可通过如下 2 种方式进行设置。默认值为“循环”。

设置	说明
循环	Plus CPU 模块与目标按照设置的 RPI（通信周期）(9) 发送数据。
Change Of State (COS)	Plus CPU 模块与目标在设置的 RPI（通信周期）(9) 或值发生变化时发送数据。

(7) COS 抑制时间

使用发送触发 (6) 选择 Change Of State (COS) 时，设置从值发生变化后到发送之间的时间。Plus CPU 模块在检测到值变化后到经过设置时间这段时间内不会发送数据。

注释：值的变化表示 Plus CPU 模块内部拥有的 I/O 数据缓存的数据的变化。通过结束处理从数据寄存器向 I/O 数据缓存反映。

(8) 控制寄存器

设置 CIP 连接使用的数据寄存器。

设备地址	项目	说明
起始编号 +0	位 0	连接状态 ON: 连接并可通信的状态 OFF: 未连接状态（正在重试等）
	位 1 ~ 15	保留
起始编号 +1	保留	
起始编号 +2		
起始编号 +3		
起始编号 +4		

在 (9) ~ (14) 中设置与 I/O 信息通信的输入方向（目标→Plus CPU 模块（发起者））相关的参数。

(9) RPI

设置 Plus CPU 模块（发起者）和目标进行 I/O 信息通信时的输入方向（目标→发起者）的通信周期。可设置范围为 10 ~ 10000ms。默认值为 50ms。

(10) CIP 连接类型

设置 Plus CPU 模块（发起者）接收目标发出的数据的方法。可通过如下 2 种方式进行设置。默认值为“Point to point”。

设置	说明
Point to point	在 Plus CPU 模块 1 对 1 接收来自目标的数据时进行设置。
Multicast	含有 Plus CPU 模块的多台发起者从 1 台目标接收数据时进行设置。在这种情况下，请设置超时 (3) 和 RPI(9)，以使 IN (T->O) 超时时间为 250ms 或更长。

(11) 实例 ID/CIP 标记

设置目标的 CIP 连接点。根据 CIP 连接名称，设置目标的实例 ID 或 CIP 标记名称。可通过如下 5 种方式进行设置。

CIP 连接名称	CIP 连接点	设置范围
Exclusive Owner (ID)	实例 ID	1 ~ 65535
Exclusive Owner (CIP Tag)	CIP 标记名称	UTF-8、最大 64 字节（含字符串终端）
Input Only (ID)	实例 ID	1 ~ 65535
Input Only (CIP Tag)	CIP 标记名称	UTF-8、最大 64 字节（含字符串终端）
Listen Only (ID)	实例 ID	1 ~ 65535

(12)设备分配 (IN (T->O))

设置 Plus CPU 模块（发起者）用于存储接收自目标的数据的数据寄存器。可以字为单位，将数据分配到多个数据寄存器中。最多可设置 8 个区域。

分配区域的总数据大小为最大 720 个字。从设置的数据寄存器开始，按照设置的数据大小使用对应的数据寄存器。在 1 ~ 720 的范围内设置数据大小。默认值为 1。请设置第一数据寄存器，不要超出设备的范围。

(13)总数据大小 (IN (T->O))

显示设备分配 (IN (T->O)) (12) 中使用的数据寄存器的合计值。

(14)剩余的数据大小 (IN (T->O))

显示可用剩余数据大小（从 720 字节中减去使用的数据寄存器的合计值）。

在 (15) ~ (20) 中设置与 I/O 信息通信的输出方向（Plus CPU 模块（发起者）→目标）相关的参数。

(15)RPI

设置 Plus CPU 模块（发起者）和目标进行 I/O 信息通信时的输出方向（发起者→目标）的通信周期。可设置范围为 10 ~ 10000ms。默认值为 50ms。

CIP 连接名称上已设置 Input Only 和 Listen Only 时，该 RPI 变为 Heartbeat 的周期。自动设置 16 倍的 IN (T->O) 数据的 RPI 的值，可任意变更。

(16)CIP 连接类型

设置 Plus CPU 模块（发起者）向目标发送数据的方法。仅可设置“Point to point”。

设置	说明
Point to point	在 Plus CPU 模块 1 对 1 发送来自目标的数据时设置。

(17)实例 ID/CIP 标记

设置目标的 CIP 连接点。根据 CIP 连接名称，设置目标的实例 ID 或 CIP 标记名称。可通过如下 5 种方式进行设置。

CIP 连接名称	CIP 连接点	设置范围
Exclusive Owner (ID)	实例 ID	1 ~ 65535
Exclusive Owner (CIP Tag)	CIP 标记名称	UTF-8、最大 64 字节（含字符串终端）
Input Only (ID)	实例 ID	1 ~ 65535*1
Input Only (CIP Tag)	CIP 标记名称	无需设置
Listen Only (ID)	实例 ID	1 ~ 65535*2

*1 请设置目标的 Input Only 用的实例 ID。

*2 请设置目标的 Listen Only 用的实例 ID。

(18)设备分配 (OUT (O->T))

设置 Plus CPU 模块（发起者）存储向目标发送的数据的数据寄存器。可以字为单位，将数据分配到多个数据寄存器中。最多可设置 8 个区域。

分配区域的总数据大小为最大 720 个字。从设置的数据寄存器开始，按照设置的数据大小使用对应的数据寄存器。在 1 ~ 720 的范围内设置数据大小。默认值为 1。请设置第一数据寄存器，不要超出设备的范围。

(19)总数据大小 (OUT (O->T))

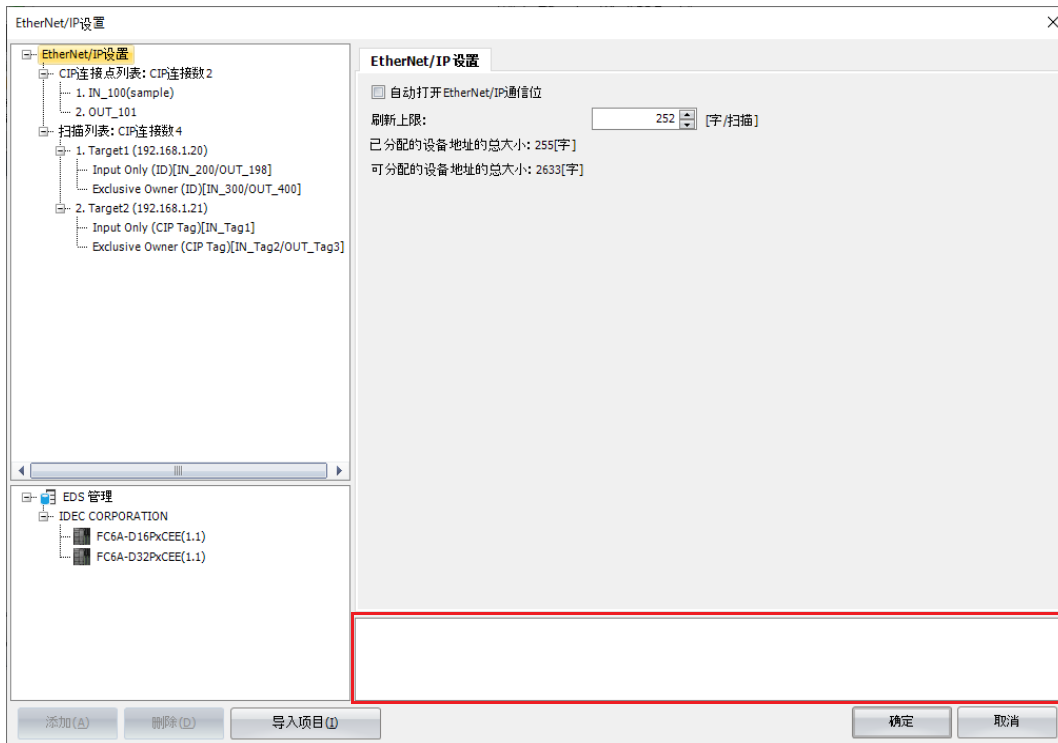
显示设备分配 (OUT (O->T)) (18) 中使用的数据寄存器的合计值。

(20)剩余的数据大小 (OUT (O->T))

显示可用剩余数据大小（从 720 字节中减去使用的数据寄存器的合计值）。

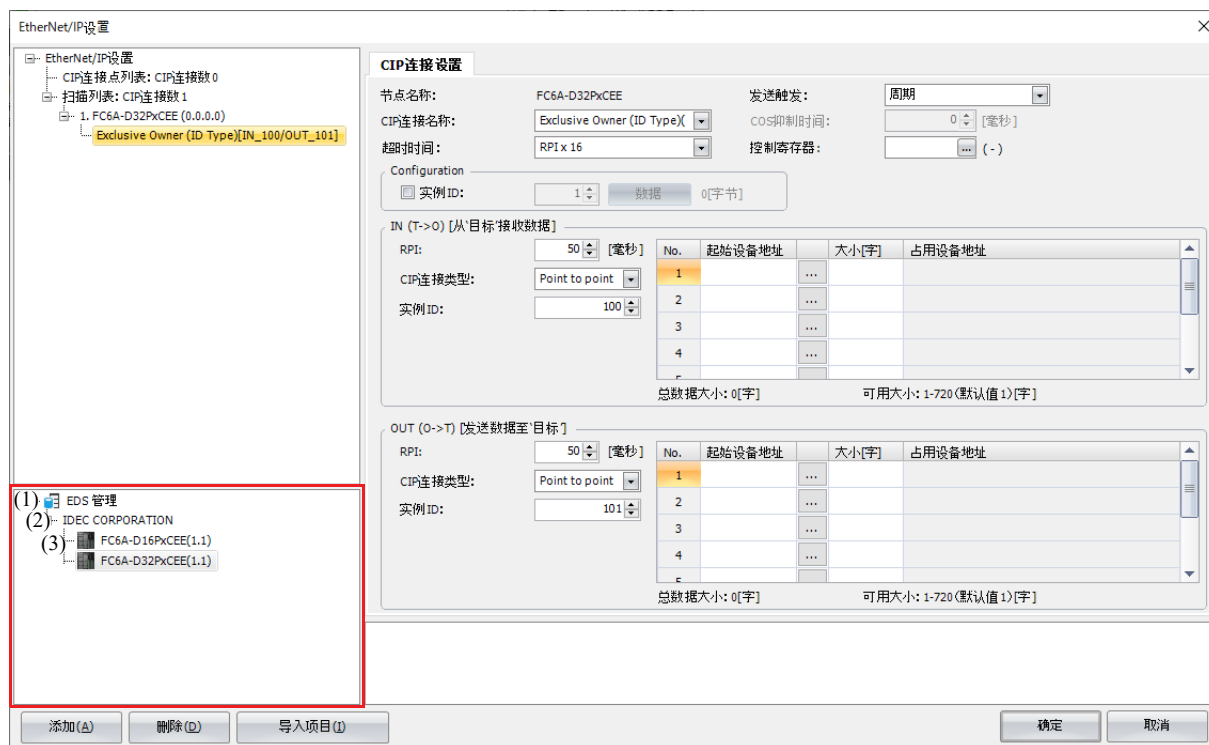
信息显示区域

“EtherNet/IP 设置”对话框中设置的内容存在错误时，显示错误内容。
单击错误内容后，显示错误的位置。



EDS 管理区域

显示导入的 EDS 文件中设置的 EtherNet/IP 设备的列表。



(1) “EDS 管理”节点

选择“EDS 管理”节点并将其展开以显示在导入的 EDS 文件中设置的 EtherNet/IP 设备信息的列表。

(2) “供应商”节点

显示在导入的 EDS 文件中设置的 EtherNet/IP 设备的供应商名称。

(3) “设备”节点

显示在导入的 EDS 文件中设置的 EtherNet/IP 设备的名称及其修订版。

右键单击“EDS 管理”区域中的每个节点介绍显示的上下文菜单的操作。

节点	上下文菜单	操作
EDS 管理	导入	导入 EDS 文件。
供应商	删除所选供应商的 EDS 文件	删除所选供应商的 EDS 文件。
设备	导出	已选 EtherNet/IP 设备的 EDS 文件以文本形式输出。
	打开	打开已选 EtherNet/IP 设备的 EDS 文件。
	删除	删除已选 EtherNet/IP 设备的 EDS 文件。
	添加至扫描列表	将已选的 EtherNet/IP 设备添加至扫描列表。

注释:

- 在选择了“设备”节点的状态下，如果拖放到 Ethernet Net/IP 树形区域的扫描列表中，则可以将选择的 Ethernet/IP 设备添加到扫描列表中。
- 由于导入的 EDS 文件会保存在 PC 内部，所以下次启动 WindLDR 时，在“EDS 管理”区域也会一览显示设置在该 EDS 文件中的 Ethernet/IP 设备的信息。
- 本公司产品的 Ethernet/IP 机器的 EDS 文件是 WindLDR8.18.0 以后的版本最开始导入的。

EtherNet/IP 通信的设置流程

EtherNet/IP 通信的基本设置

选择 EtherNet/IP 树形区域的“Ethernet Net/IP 设置”节点*1 后，在参数设置区域上显示“EtherNet/IP 设置”选项卡。在该选项卡中执行用于 EtherNet/IP 通信的基本设置。有关详情，请参见第 16-22 页上的“已选择“Ethernet Net/IP 设置”节点时”。

*1 有关“Ethernet Net/IP 设置”节点的详情，请参见第 16-20 页上的“EtherNet/IP 树形区域”。

将 Plus CPU 模块用作目标

介绍将 Plus CPU 模块用作目标时所需的设置。

Plus CPU 模块（目标）与其他扫描仪设备（发起者）进行数据收发时，需要设置 CIP 连接点。

CIP 连接点的设置

Plus CPU 模块与发起者进行数据收发的设置。设置存储通信数据的数据寄存器、数据大小、数据输入 / 输出方向等。

选择 EtherNet/IP 树形区域的“CIP 连接点”节点*1 后，在参数设置区域上显示“CIP 连接点设置”选项卡。在该选项卡中设置 CIP 连接点。有关详情，请参见第 16-24 页上的“已选择“CIP 连接点”节点时”。

*1 有关“CIP 连接点”节点的详情，请参见第 16-20 页上的“EtherNet/IP 树形区域”。

将 Plus CPU 模块用作发起者

介绍将 Plus CPU 模块用作发起者时所需的设置。

需要设置与 Plus CPU 模块进行 I/O 信息通信的目标设备，并设置 CIP 连接。这些设置通常由各个公司提供的 EDS 文件来简单地设定。有关详情，请参见第 16-32 页上的“使用 EDS 文件创建扫描列表”。

目标的设置

设置与 Plus CPU 模块进行 I/O 信息通信的目标的 IP 地址等，对目标设备进行设置。

选择 EtherNet/IP 树形区域的“目标”节点*1 后，在参数设置区域上显示“目标设置”选项卡。在该选项卡中设置目标设备。有关详情，请参见第 16-25 页上的“已选择“目标”节点时”。

*1 有关“目标”节点的详情，请参见第 16-20 页上的“EtherNet/IP 树形区域”。

CIP 连接的设置

Plus CPU 模块与目标进行数据收发的设置。在每个目标设置 CIP 连接名称、CIP 连接类型、存储通信数据的数据寄存器、数据大小、I/O 信息通信的 RPI（通信周期）等。

选择 EtherNet/IP 树形区域的“CIP 连接”节点*1 后，在参数设置区域上显示“CIP 连接设置”选项卡。在该选项卡中设置 CIP 连接。有关详情，请参见第 16-27 页上的“已选择“CIP 连接”节点时”。

*1 有关“CIP 连接”节点的详情，请参见第 16-20 页上的“EtherNet/IP 树形区域”。

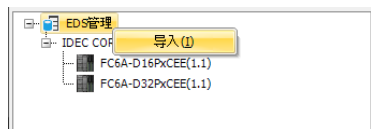
使用 EDS 文件创建扫描列表

Plus CPU 模块为了与目标进行数据的接收和发送所必需的目标设备设置以及 CIP 连接设置，可通过 EDS（Electric Data Sheets）文件简单设置。EDS 文件是一种用于定义 EtherNet/IP 设备固有信息的文件，例如供应商名称，数据发送 / 接收设置和参数规格。EDS 文件由各个 EtherNet/IP 设备的供应商创建并提供。请从供应商的官网等获取 EDS 文件。

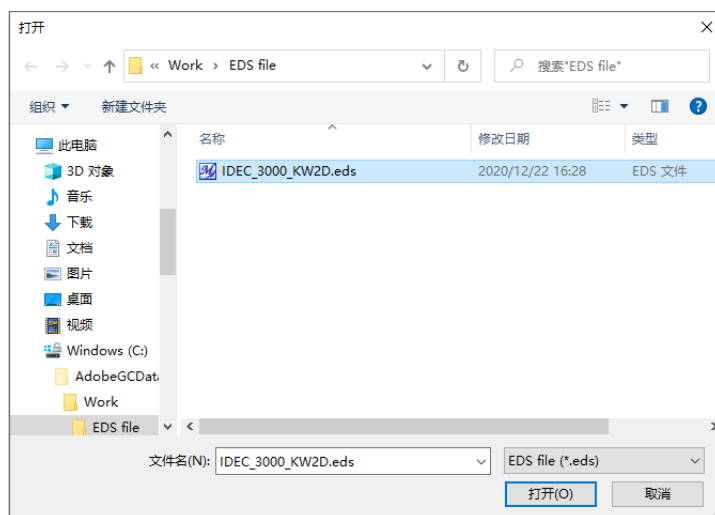
本节对导入 EDS 文件并创建扫描列表的步骤进行介绍。

●操作过程

1. 在“EDS 管理”区域中，右键单击“EDS 管理”节点，然后单击“导入”。

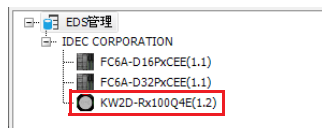


显示“打开”对话框。

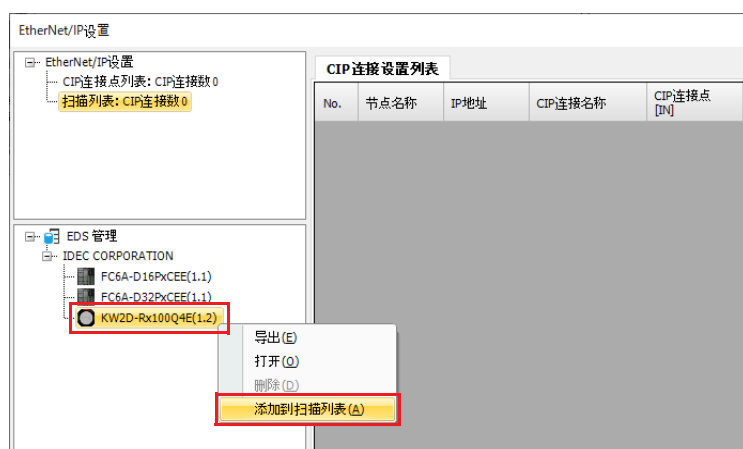


2. 选择 EDS 文件，然后单击“打开”按钮。

所选的 EDS 文件将导入到“EDS 管理”区域。



3. 在“EDS 管理”区域中，右键单击“EDS 管理”节点，然后单击“添加到扫描列表”。



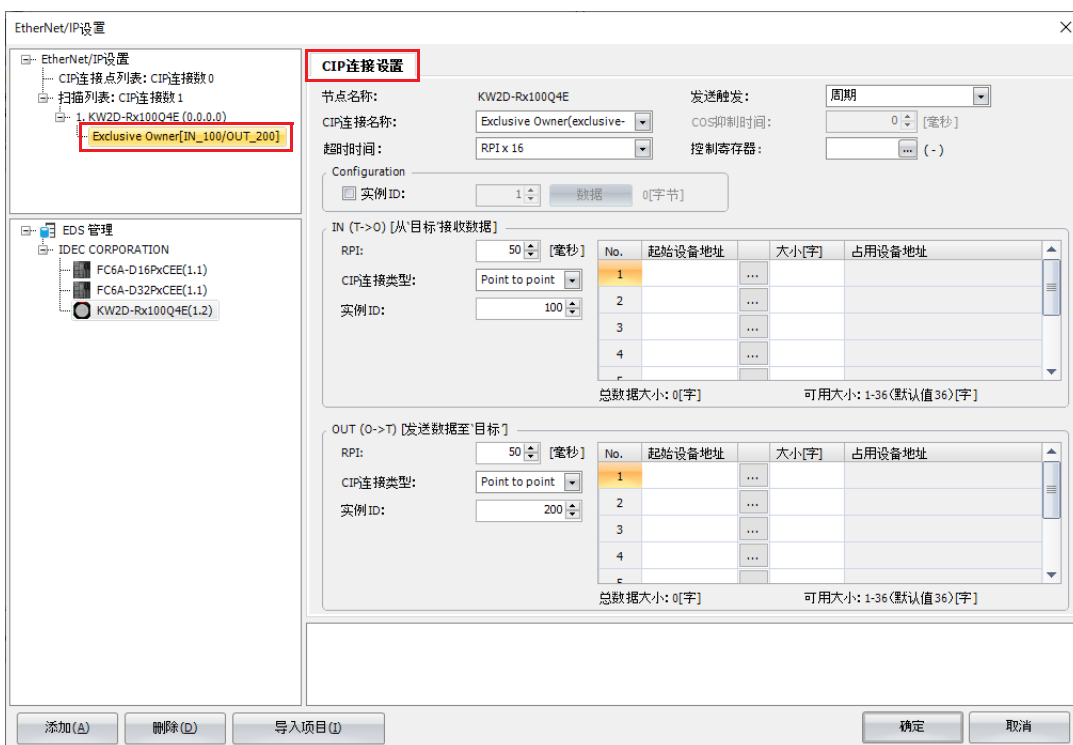
所选 EDS 文件的设置将添加到扫描列表中。

16: ETHERNET/IP通信

- 在 EtherNet/IP 树形区域中选择“目标”节点，然后在“目标设置”选项卡上设置 IP 地址。另外，确认“目标设置”选项卡上的设置，如有必要，更改设置。



- 在 EtherNet/IP 树形区域中选择“CIP 连接”节点，然后在“CIP 连接设置”选项卡上设置超时、控制寄存器以及设备分配。另外，确认 [CIP 连接设置] 选项卡上的设置，如有必要，更改设置。



- 单击“确定”按钮。

至此，完成扫描列表的创建。

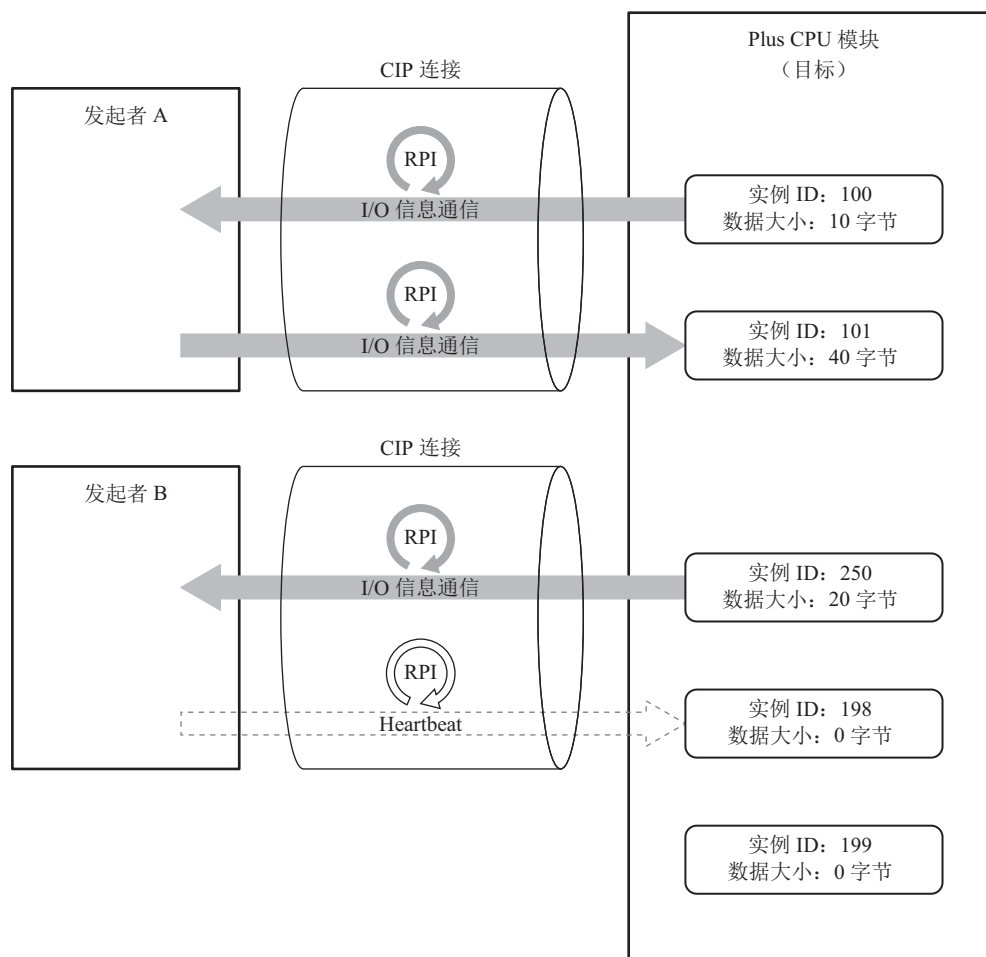
注释： EDS 文件由各 EtherNet/IP 设备制造商准备。请与各 EtherNet/IP 设备制造商联系，以获取 EDS 文件。关于 EDS 文件的获取方式，请咨询各 EtherNet/IP 设备的供应商。

EtherNet/IP 的设置示例

将 Plus CPU 模块用作目标

介绍将 Plus CPU 模块用作目标，与发起者连接时的设置示例。

系统构成图



向 Plus CPU 模块设置的规格

将 Plus CPU 模块作为目标，设置从发起者 A 和 B 读写的 CIP 连接点。

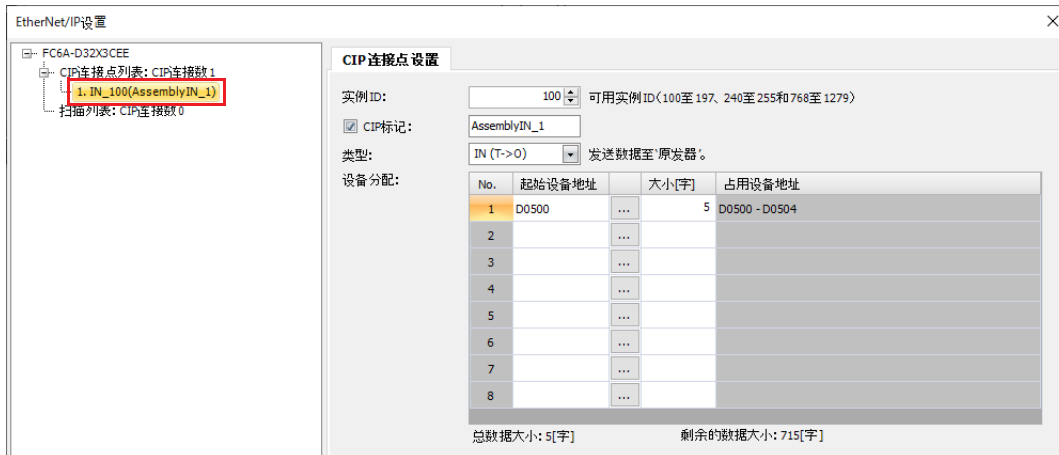
输入 / 输出类型	CIP 连接点		数据存储用的数据寄存器
	实例 ID	数据大小	
输入	100	10 字节 (5 个字)	D0500
输出	101	40 字节 (20 个字)	D0600
输入	250	20 字节 (10 个字)	D0700
输出 (Input Only 用)	198 (系统固定)	0 字节	—
输出 (Listen Only 用)	199 (系统固定)	0 字节	—

设置

设置 Plus CPU 模块的 CIP 连接点。

■ Plus CPU 模块的 CIP 连接点列表设置

Plus CPU 模块（目标）将存储在数据寄存器中的 CIP 连接点（实例 ID：100）的数据发送到发起者。



设置项目	说明
1. CIP 标记	“AssemblyIN_1”
实例 ID	100
类型	IN (T->O)
设备分配	D0500 ~ D0504（5 个字）

Plus CPU 模块（目标）从发起者接收 CIP 连接点（实例 ID：101）的数据后存储到数据寄存器中。



设置项目	说明
2. CIP 标记	“AssemblyOUT_1”
实例 ID	101
类型	OUT (O->T)
设备分配	D0600 ~ D0619（20 个字）

Plus CPU 模块（目标）将存储在数据寄存器中的 CIP 连接点（实例 ID：200）的数据发送到发起者。



设置项目	说明
3. CIP 标记	“AssemblyIN_2”
实例 ID	250
类型	IN (T->O)
设备分配	D0700 ~ D0709 (10 个字)

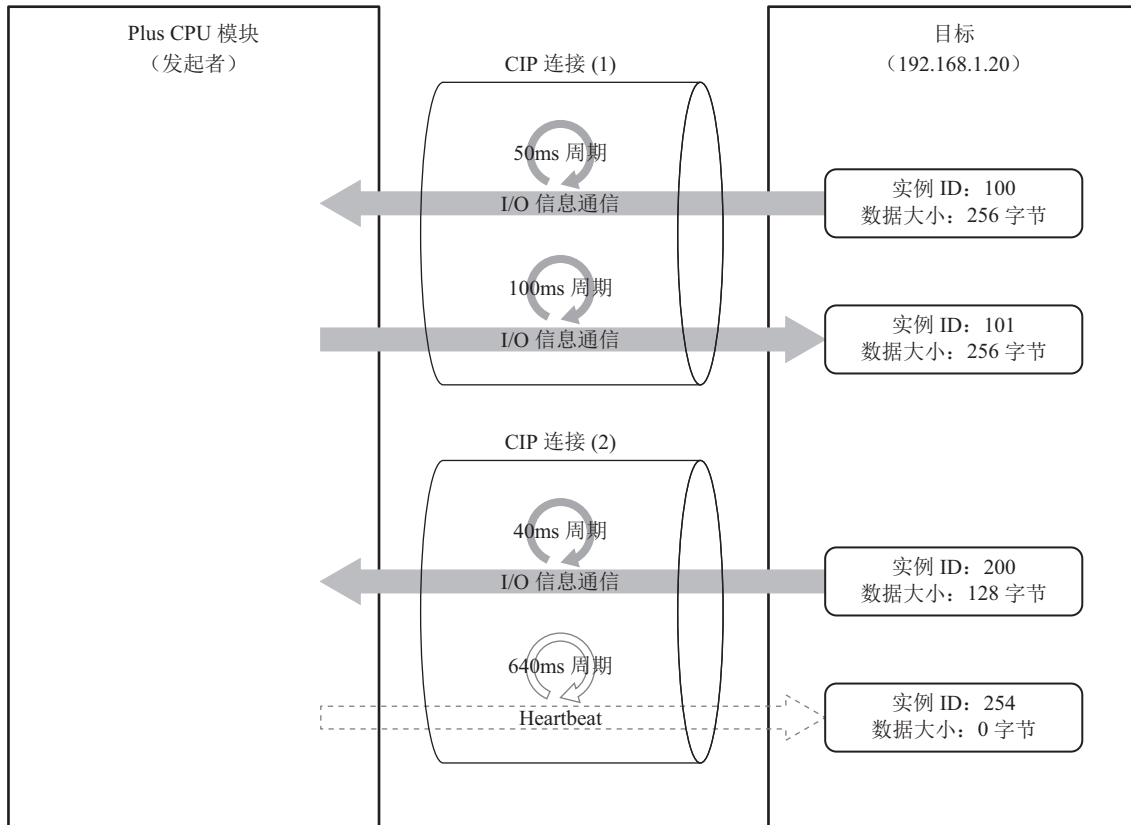
注释：在发起者设立 CIP 连接时，指定 RPI 和 Exclusive Owner 等的 CIP 连接的种类。

配置即完成。

将 Plus CPU 模块用作发起者

介绍将 Plus CPU 模块用作发起者与目标连接时的设置示例。

系统构成图



连接目标的规格

IP 地址	192.168.1.20			
CIP 连接点	下表显示 Plus CPU 模块作为发起者连接的目标的 CIP 连接点。			
	CIP 连接	类型	实例 ID	数据大小
	CIP 连接 (1)	Configuration	1	0 字节
		输入	100	256 字节 (128 个字)
		输出	101	256 字节 (128 个字)
	CIP 连接 (2)	Configuration	1	0 字节
输入		200	128 字节 (64 个字)	
输出 (Heartbeat)		254	0 字节	

设置

执行目标设备、CIP 连接的设置。

■ Plus CPU 模块的扫描列表设置（目标设备的设置）

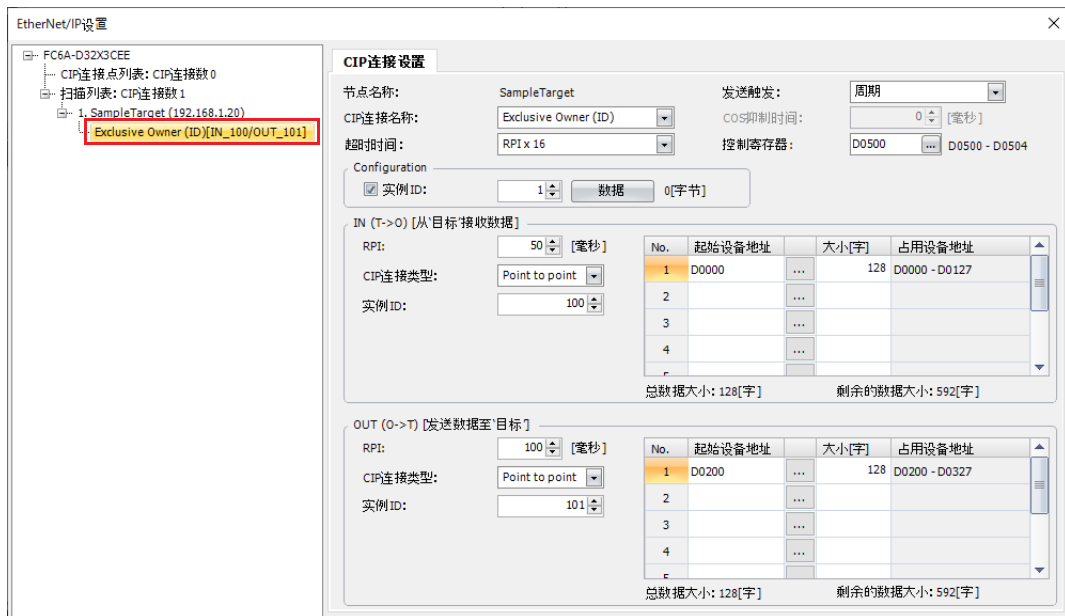


设置项目	说明
使用此目标	启用
节点名称	SampleTarget
IP 地址	192.168.1.20
电子密钥	兼容性检查
	禁用

■ Plus CPU 模块的扫描列表设置（CIP 连接的设置）

CIP 连接 (1)

Plus CPU 模块（发起者）向 CIP 连接点（实例 ID：100 和 101）请求设立 CIP 连接，成功后每 50ms 从目标接收数据，每 100ms 向目标发送数据。



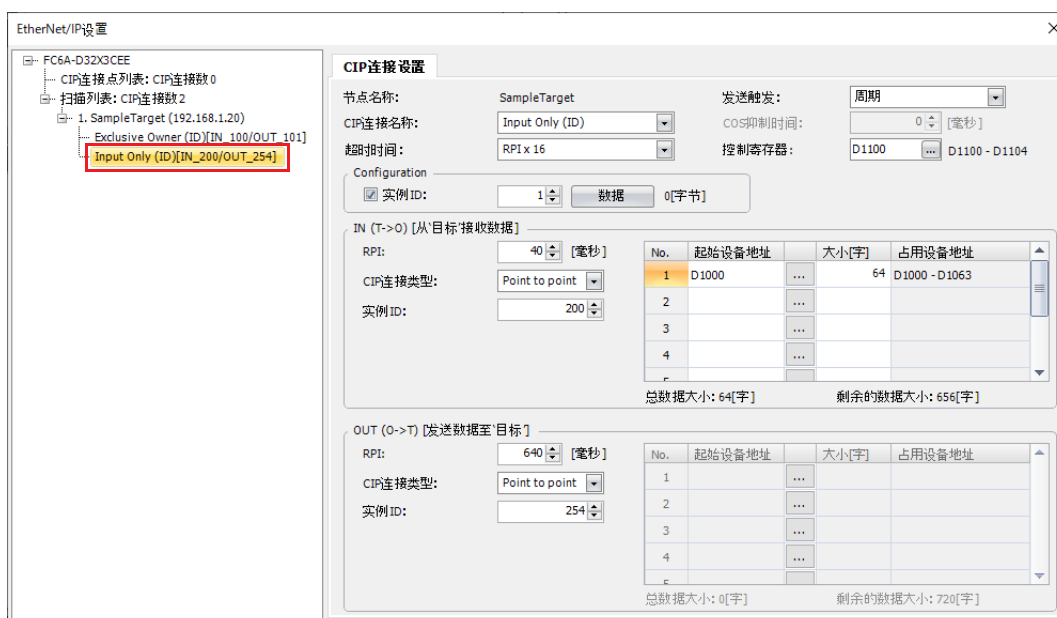
设置项目	说明
CIP 连接名称	Exclusive Owner (ID)
发送触发	循环
超时时间	RPI x 16
控制寄存器	D0500
Configuration	实例 ID
	数据
	1
	0 字节

16: ETHERNET/IP通信

设置项目		说明
IN (T->O)	RPI	50msec
	CIP 连接类型	Point to point
	实例 ID	100
	设备分配	D0000 ~ D0127 (128 个字)
OUT (O->T)	RPI	100msec
	CIP 连接类型	Point to point
	实例 ID	101
	设备分配	D0200 ~ D0327 (128 个字)

CIP 连接 (2)

不向目标发送数据时，请在 CIP 连接名称上设置 Input Only。Plus CPU 模块（发起者）向 CIP 连接点（实例 ID：200 和 254）请求设立 CIP 连接，成功后每 40ms 从目标接收数据。不会向目标发送数据，而是每 640ms 发送 Heartbeat。



设置项目		说明
CIP 连接名称		Input Only (ID)
发送触发		循环
超时时间		RPI x 16
控制寄存器		D1100
Configuration	实例 ID	1
	数据	0 字节
IN (T->O)	RPI	40msec
	CIP 连接类型	Point to point
	实例 ID	200
	设备分配	D1000 ~ D1063 (64 个字)
OUT (O->T)	RPI	640msec
	CIP 连接类型	Point to point
	实例 ID	254

配置即完成。

对象

Identity 对象

分类

■ 分类服务

支持 Get_Attribute_Single (0EH)、Get_Attributes_All (01H)。

ID	服务
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■ 类属性 (实例 ID: 0)

ID	访问	名称	数据类型
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Number of Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

实例

■ 实例服务

支持 Get_Attribute_Single (0EH)、Get_Attributes_All (01H)、Reset 服务 (05H)。

ID	服务
0EH	Get_Attribute_Single
01H	Get_Attributes_All
05H	Reset

■ 实例属性 (实例 ID: 1)

ID	访问	名称	数据类型	说明	属性值
1	Get	Vendor ID	UINT	供应商识别编号	159
2	Get	Device Type	UINT	常规设备类型	14 (Programmable Logic Controller)
3	Get	Product Code	UINT	产品的识别代码	*1
4	Get	Revision	STRUCT	Identity 对象的修订	系统软件版本
		Major Revision	USINT	主要修订	
		Minor Revision	USINT	次要修订	
5	Get	Status	WORD	设备的当前状态	
6	Get	Serial Number	UDINT	序列号	
7	Get	Product Name	SHORT-STRING	产品名称	*1

*1 每个产品的属性值不同。

	FC6A-D16R1CEE, FC6A-D16R4CEE	FC6A-D16P1CEE, FC6A-D16P4CEE	FC6A-D16K1CEE, FC6A-D16K4CEE	FC6A-D32P1CEE, FC6A-D32P4CEE	FC6A-D32K1CEE, FC6A-D32K4CEE
Product Code	2000	2010	2020	2030	2040
Product Name	"FC6A-D16RxCEE"	"FC6A-D16PxCEE"	"FC6A-D16KxCEE"	"FC6A-D32PxCEE"	"FC6A-D32KxCEE"

Message Router 对象

分类

■ 分类服务

ID	服务
0EH	Get_Attribute_Single

■ 类属性 (实例 ID: 0)

ID	访问	名称	数据类型
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Number of Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

实例

■ 实例服务

ID	服务
0EH	Get_Attribute_Single

■ 实例属性 (实例 ID: 1)

ID	访问	名称	数据类型
1	Get	Object_list	STRUCT
	Get	Number	UINT
	Get	Classes	UINT 的排列
2	Get	Number Available	UINT

Assembly 对象

分类

■ 分类服务

ID	服务
0EH	Get_Attribute_Single

■ 类属性 (实例 ID: 0)

ID	访问	名称	数据类型
1	Get	Revision	UINT

实例

■ 实例服务

ID	服务
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■ 实例属性

(1) 发起者→目标

ID	访问	名称	数据类型
3	Get/Set	Data	BYTE 的排列
4	Get	Size	UINT

(2) 目标→发起者

ID	访问	名称	数据类型
3	Get	Data	BYTE 的排列
4	Get	Size	UINT

Connection Manager 对象

分类

■ 分类服务

无分类服务。

■ 类属性

无类属性。

实例

■ 实例服务

ID	服务
54H	Forward_Open
5BH	Large_Forward_Open
4EH	Forward_Close

■ 实例属性

无实例属性。

TCP/IP Interface 对象

分类

■ 分类服务

ID	服务
0EH	Get_Attribute_Single

■ 类属性

ID	访问	名称	数据类型
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Num Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

实例

■ 实例服务

ID	服务
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■ 实例属性

ID	访问	名称	数据类型
1	Get	Status	DWORD
2	Get	Configuration Capability	DWORD
3	Get	Configuration Control	DWORD
4	Get	Physical Link Object	STRUCT
		Path size	UINT
		Path	EPATH
5	Get	Interface Configuration	STRUCT
		IP Address	UDINT
		Network Mask	UDINT
		Gateway Address	UDINT
		Name Server	UDINT
		Name Server 2	UDINT
6	Get	Domain Name	STRING
6	Get	Host Name	STRING
13	Get/Set	Encapsulation Inactivity Timeout	UINT

Ethernet Link 对象

设置和读取以太网通信。此外，读取以太网通信的状态。

分类**■ 分类服务**

ID	服务
0EH	Get_Attribute_Single

■ 类属性

ID	访问	名称	数据类型
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Num Instances	UINT

实例**■ 实例服务**

支持 Get_Attribute_Single (0EH)、Get_Attributes_All (01H)。

ID	服务
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■ 实例属性

ID	访问	名称	数据类型
1	Get	Interface Speed	UDINT
2	Get	Interface Flags	DWORD
3	Get	Physical Address	ARRAY of 6 USINTs
11	Get	Interface Capability	STRUCT
		Capability Bits	DWORD
		Speed/Duplex Options	STRUCT
		Speed/Duplex Array Count	USINT
		Speed/Duplex Array	ARRAY of STRUCT
		Interface Speed	UINT
		Interface Duplex Mode	USINT

17:MQTT 通信

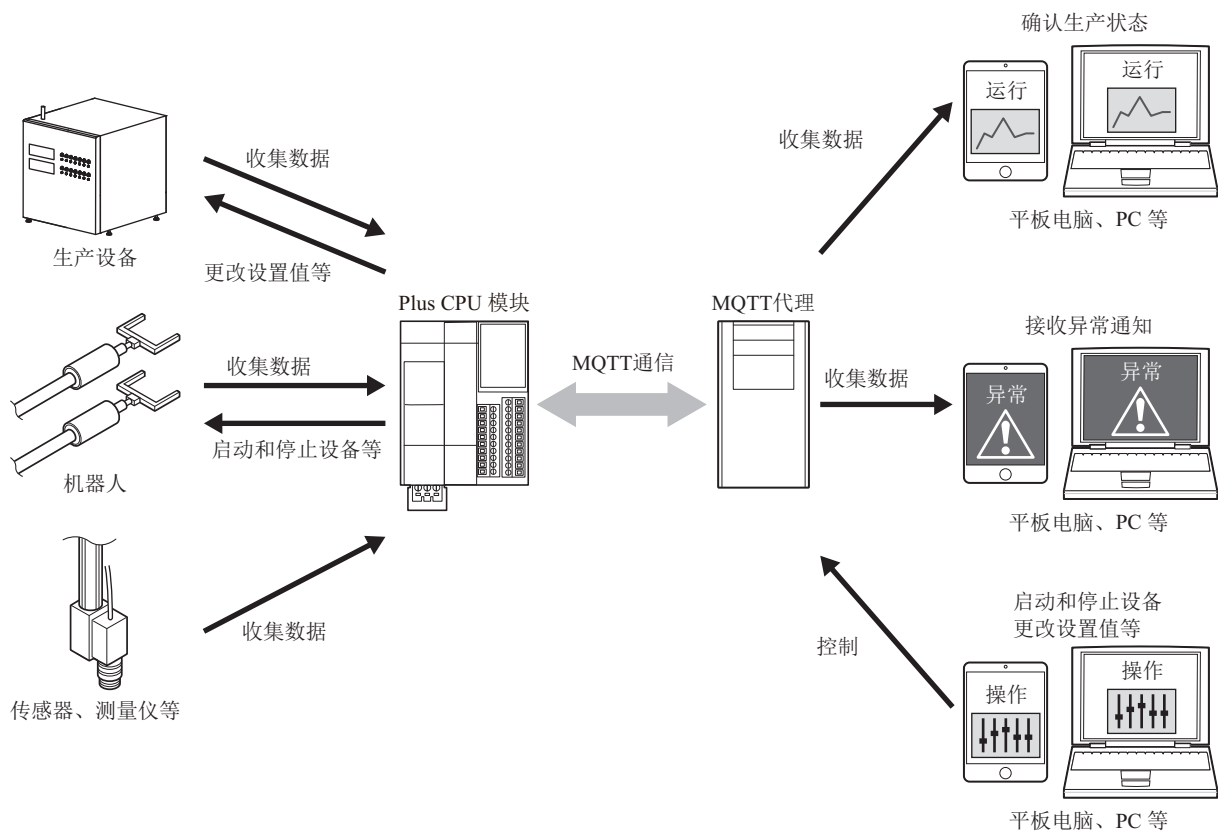
本章将对 Plus CPU 模块的 MQTT 通信进行介绍。

概要

Plus CPU 模块支持 MQTT 通信。

MQTT 是一种协议，它的特点是收发数据轻量化以及易于扩展。可通过使用 MQTT 通信实现以下操作。

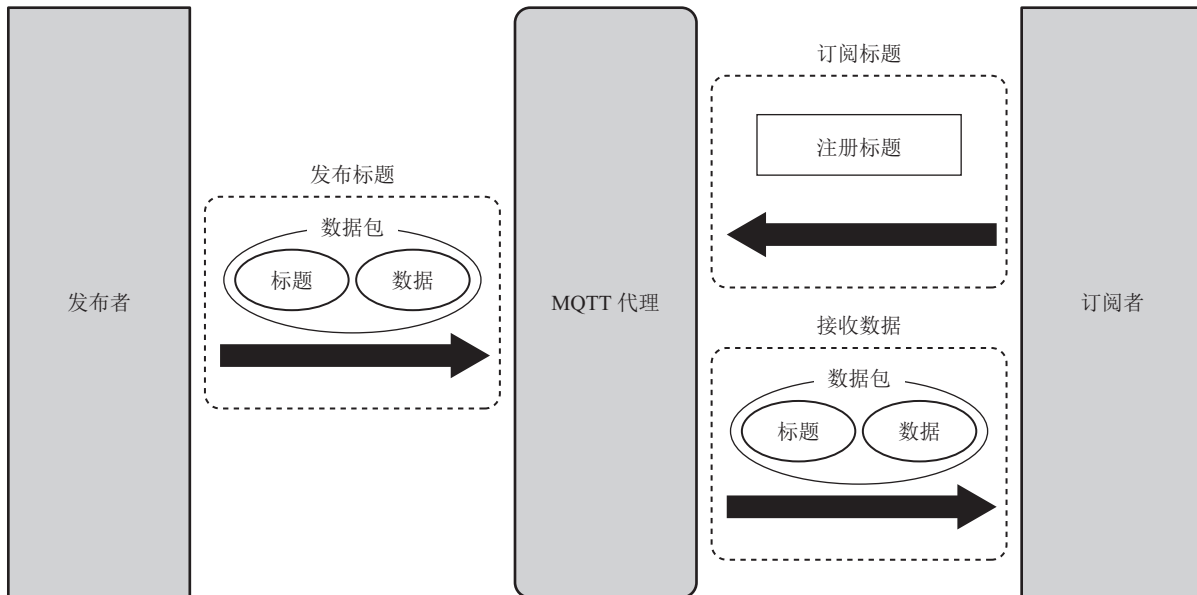
- 收集远程设备数据，确认生产状态
- 在远程设备发生异常时接收通知
- 启动和停止远程设备，或者更改设置值



17: MQTT通信

Plus CPU 模块可以作为 MQTT 通信中的客户端（发布者和订阅者）与代理通信。有关 MQTT 通信的详情以及 MQTT 通信中的常用术语说明，请参见 OASIS（Organization for the Advancement of Structured Information Standards）公开的 MQTT 规格书。

MQTT 的网络由代理、发布者和订阅者构成。订阅者连接到代理，注册要订阅的标题。发布者连接到代理，向代理发送包含标题和数据的数据包。代理参照数据包内的标题，将此数据包发送给订阅了该标题的订阅者，然后订阅者接收到此数据包内的数据。



注释：

- 订阅者在代理上注册要订阅的标题这一过程称为“订阅标题”。
- 发布者向代理发送包含标题和数据的数据包这一过程称为“发布标题”。
- 在 MQTT 通信中收发数据包被称为 MQTT 数据包。
- 标题是用于区分数据种类的信息。发布者根据要发送数据的种类附加标题，并发送数据。订阅者根据要接收数据的种类订阅标题，仅接收所需数据。

支持机型以及通信端口

支持的机型和通信端口如下所示。

通信端口	All-in-One CPU 模块			CAN J1939 All-in-One CPU 模块	Plus CPU 模块	
	16-I/O 型	24-I/O 型	40-I/O 型		Plus 16-I/O 型	Plus 32-I/O 型
以太网端口 1	—	—	—	—	是	是
以太网端口 2	—	—	—	—	—	—
HMI- 以太网端口	—	—	—	—	—	—

MQTT 通信无法通过用于串行通信的通信端口 1 ~ 33 进行使用。

基本规格

基本规格如下所示。

模块		Plus CPU 模块
通信端口		以太网端口 1
基准规格		MQTT Version 3.1.1
基本规格	主机名称	最多为 128 字节
	验证	支持
	账户名	最多为 128 字节
	密码	最多为 496 字节
	SSL/TLS	支持
	客户端 ID	最多为 128 字节
	保持活跃	5 ~ 65535 秒
	QoS	0、1、2
	标题	最多为 256 字节
	数据格式	JSON 格式，长度最大为 32KB（发布）和 8KB（订阅）
发布规格	Retain	支持
	Will	不支持
订阅规格	通配符（#、+）	支持
	Persistent Session	不支持

支持的 MQTT 代理和云服务

以下 MQTT 代理和云服务用于检查连接、发布和订阅的基本操作。（截至 2022 年 7 月）

- Mosquitto
- AWS IoT Core（Amazon.com）（端口编号 8883）*1
- Azure IoT Hub（azure.microsoft.com）（端口编号 8883）*1

你可以使用 Plus CPU 模块和上述 MQTT 代理或云服务所支持的服务。然而，由于云服务规范的变化，AWS IoT Core 和 Azure IoT Hub 可能无法连接。

Plus CPU 模块还支持以下与 MQTT 代理和云服务的连接方式。

■ 连接至一个一般用途的 MQTT 代理

Plus CPU 模块可以连接至一个一般用途的 MQTT 代理。

■ 连接至 AWS IoT Core

在 AWS IoT Core 事前注册的 Plus CPU 模块可以连接至 AWS IoT Core。

■ 使用 SAS 连接至 Azure IoT Hub

在 Azure IoT Hub 事前注册的 Plus CPU 模块可以使用 Shared Access Signature（SAS）连接至 Azure IoT Hub。SAS 形符用于将 Plus CPU 模块与 Azure IoT Hub 连接。

■ 使用 X.509 证书连接至 Azure IoT Hub

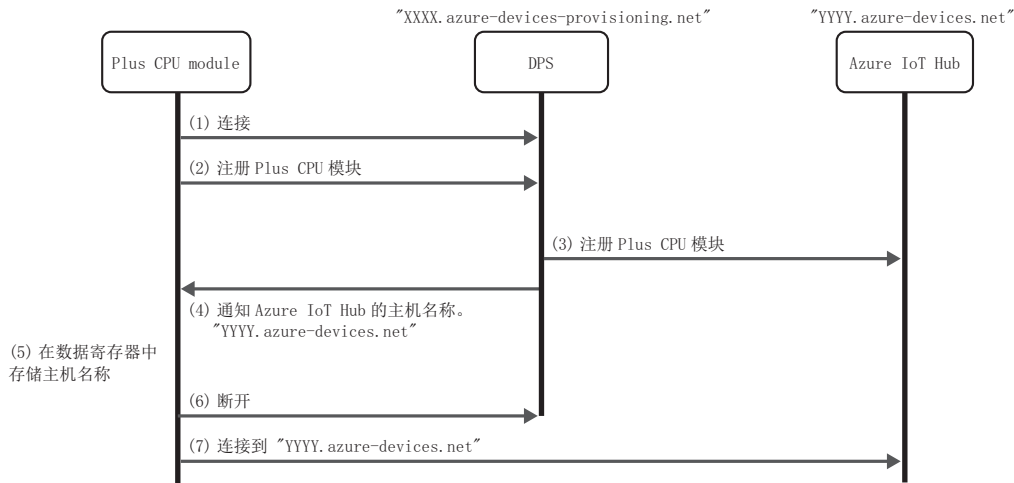
在 Azure IoT Hub 事前注册的 Plus CPU 模块可以使用 X.509 证书连接至 Azure IoT Hub。X.509 证书被指定为设备端证书（客户端证书）。

■ 通过 DPS 连接至 Azure IoT Hub

你可以通过 Device Provisioning Service（DPS）将 Plus CPU 模块注册至 Azure IoT Hub，并连接至 Azure IoT Hub。SAS 形符用于将 Plus CPU 模块连接到 DPS 和 Azure IoT Hub。

*1 无法连接到 AWS IoT Core / Azure IoT Hub 的 443 号端口。

从连接到 DPS 到连接到 Azure IoT Hub 的流程如下图所示。



- (1) Plus CPU 模块连接到 DPS。
- (2) 将 Plus CPU 模块注册到 DPS。
- (3) DPS 将 Plus CPU 模块注册到 Azure IoT Hub。
- (4) DPS 通知 Plus CPU 模块所连接的 Azure IoT Hub 的主机名称。
- (5) Plus CPU 模块在数据寄存器中存储主机名称。
- (6) Plus CPU 模块与 DPS 断开连接。
- (7) Plus CPU 模块通过存储在 (5) 中的数据寄存器中的主机名称连接到 Azure IoT Hub。

注释： 如果 Plus CPU 模块再次连接到已经注册的 DPS， Plus CPU 模块可能会被注册到另一个 Azure IoT Hub，这取决于注册组的重新配置。有关详细信息，请参见文件 Azure IoT Hub Device Provisioning Service。

MQTT 通信的设置

在“MQTT 设置”对话框中执行 MQTT 通信的设置。本节将对“MQTT 设置”对话框和相应使用方法的设置进行介绍。

● “MQTT 设置”对话框的显示步骤

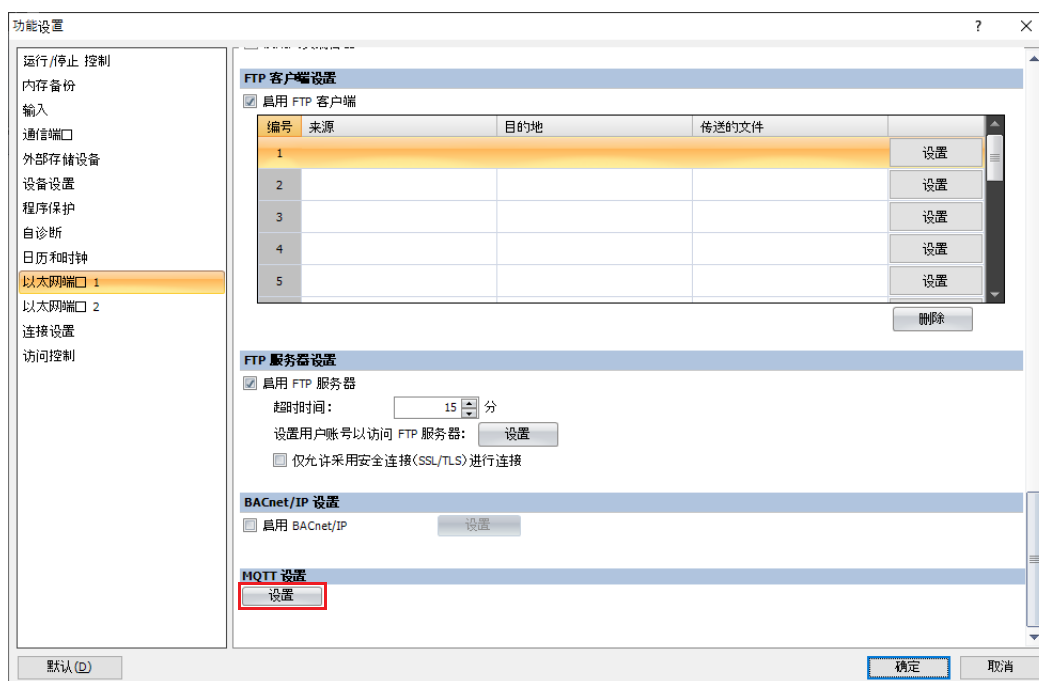
1. 在“设置”选项卡的“功能设置”中单击“以太网端口 1”。

出现“功能设置”对话框。



2. 在 MQTT 设置中单击“设置”按钮。

将显示“MQTT 设置”对话框。



“MQTT 设置”对话框

设置 Plus CPU 模块连接的 MQTT 代理或云服务，以及它的连接方式。

(1) 启用 MQTT

指定是否启用 MQTT。

(2) 云服务名称

设置要连接的云服务名称。

云服务名称	说明
一般用途	将 Plus CPU 模块连接至一个通用的 MQTT 代理。 有关详细信息，请参见第 17-7 页上的“连接至一个通用的 MQTT 代理”。
AWS IoT Core	将 Plus CPU 模块连接至 AWS IoT Core。 有关详细信息，请参见第 17-12 页上的“连接至 AWS IoT Core”。
Azure IoT Hub	将 Plus CPU 模块连接至 Azure IoT Hub。 有关详细信息，请参见第 17-6 页上的“MQTT 连接方法”。

MQTT 设置

在“MQTT 设置”对话框中选择“启用 MQTT”后，将显示“MQTT 设置”选项卡。

在该选项卡中执行用于 MQTT 通信的基本设置。

■ MQTT 连接方法

只有当“云服务名称”被设置为“Azure IoT Hub”时，才会显示“MQTT 连接方法”组，并设置 Plus CPU 模块和 Azure IoT Hub 之间的连接方法。连接方法，可从以下 3 种来设置。

MQTT 连接方法		说明
直接连接至 Azure IoT Hub	使用 Shared Access Signatures (SAS)	使用 SAS 连接至 Azure IoT Hub。 有关详细信息，请参见第 17-17 页上的“使用 SAS 连接至 Azure IoT Hub”。
	使用 X.509 证书	使用 X.509 证书连接至 Azure IoT Hub。 有关详细信息，请参见第 17-21 页上的“使用 X.509 证书连接至 Azure IoT Hub”。
通过 DPS 连接至 Azure IoT Hub		通过 DPS 连接至 Azure IoT Hub。 有关详细信息，请参见第 17-25 页上的“通过 DPS 连接至 Azure IoT Hub”。

连接至一个通用的 MQTT 代理

■ MQTT 基本设置

未选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT
云服务器名称: 一般用途

MQTT 基本设置

(1) 用 SD 记忆卡指定

代理

(3) 主机名称:
 IP 地址: 0 . 0 . 0 . 0

(4) 端口号: 8883

(5) 保持活跃: 60 秒

(6) 客户端 ID:

MAC地址:
 固定值: 4ae83dd4-e584-445b-bd63-0d4e71b84b4c (7) 产生随机 ID
 数据寄存器: (-)

(8) 必须进行身份验证才能连接到代理

(9) 用户名:
(10) 密码:

(11) 启用安全连接(SSL/TLS)

(12) 根证书: 未导入 导入 详情 删除
(13) 客户端证书: 未导入 导入 详情 删除
(14) 客户端私钥: 未导入 导入 删除

设备

(15) 连接控制: ...
(16) 连接状态: (-)

(17) 从项目导入(I) (18) 确定 (19) 取消

选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT

MQTT 基本设置

用 SD 记忆卡指定 (2) 配置 SD 记忆卡

设备

(1) 用 SD 记忆卡指定

指定是否使用 SD 记忆卡，向 Plus CPU 模块写入 MQTT 通信的基本设置。

选中“用 SD 记忆卡指定”复选框后显示 (2)，(3) ~ (14) 变为不可见。此时，可在“配置 SD 记忆卡”对话框中设置细节。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(2) 配置 SD 记忆卡

打开“配置 SD 记忆卡”对话框。仅当选“用 SD 记忆卡指定”复选框时显示。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(3) 主机名称 / IP 地址

设置代理的主机名称或 IP 地址。主机名称的最大字符数为 128 个字符。仅可使用英文数字及符号。

(4) 端口号

设置代理的端口号。一般来说，MQTT 使用 1883 端口，MQTT over TLS 使用 8883 端口。在 0 ~ 65535 的范围内设置端口号。默认值为 1883。如果端口号设置为 0，则取决于启用安全连接（SSL/TLS）（(12)），当未选中复选框时使用 1883 端口，否则使用 8883 端口。

(5) 保持活跃

设置在与代理连接期间，对代理执行连接确认的时间间隔。Plus CPU 模块不收发数据包的时间超过设置的时间间隔时，向代理进行连接确认。可设置范围为 5 ~ 65535 秒。默认值为 60 秒。

(6) 客户端 ID

设置客户端 ID。客户端 ID 可以通过以下 3 种方式来设置。

客户端 ID	说明
MAC 地址	设置 Plus CPU 模块上的以太网端口 1 的 MAC 地址 *1。
固定值	设置任意字符串。 最大字符数为 128 个字符。仅可使用英文数字及符号。
数据寄存器	设置用于存储客户 ID 的数据寄存器。 从指定的数据寄存器的高位字节中依次读出该值，作为字符数据处理，并作为客户端 ID 使用 *2。 从指定的数据寄存器开始，使用连续的 64 字数据寄存器 *3。 请指定设备范围内的起始数据寄存器。客户端 ID 只能使用英文数字及符号。

*1 如果 Plus CPU 模块上的以太网端口 1 的 MAC 地址是 12-34-56-78-9A-BC（D8324=0012h，D8325=0034h，D8326=0056h，D8327=0078h，D8328=009Ah，D8329=00BCh），客户端 ID 是 "123456789ABC"。

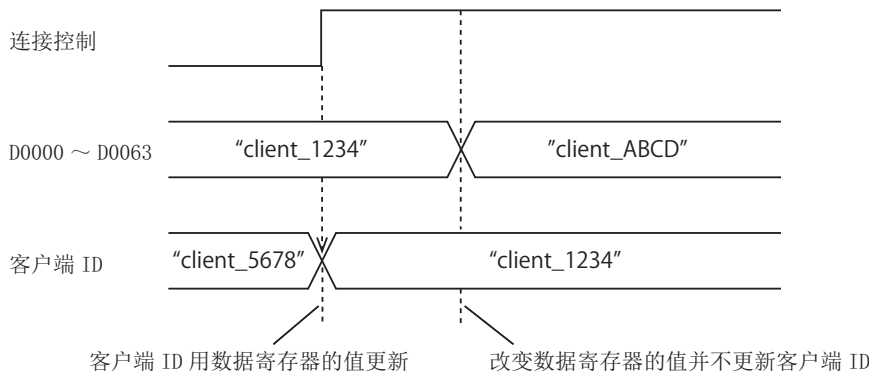
*2 例如：如果指定了 D0000，并且在每个数据寄存器中存储了以下值，那么客户端 ID 就是 "client_1234"。

数据寄存器	存储值	
	高位字节	低位字节
D0000	"c"=63h	"l"=6Ch
D0001	"i"=69h	"c"=65h
D0002	"n"=6Eh	"t"=74h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*3 如果要设置的字符串短于 128 字节（64 个字），请添加一个终止符 NULL（00h）作为字符串的结束。指定的数据寄存器的高位字节到结束字符 NULL（00h）前为止被视为字符数据并作为客户 ID 使用。

注释:

- 如果要用数据寄存器设置客户端 ID，可以用 MOVC（传送字符）指令在数据寄存器中存储字符串。有关 MOVC 指令的详细信息，请参见《梯形图编程手册》第 5 章中的“MOVC（传送字符）”
- 当你把连接控制（(15)）从关闭切换到开启时，存储在 Plus CPU 模块中的客户端 ID 将被更新。例如：如果使用数据寄存器（D0000）设置客户端 ID，客户端 ID 将在以下时间更新。



(7) 产生随机 ID

如果客户端 ID ((6)) 被设置为一个固定值，单击这个按钮将输出一个随机的客户端 ID。

(8) 必须进行身份验证才能连接到代理

设置当连接代理时是否进行账户名和密码验证。如果选中复选框，则在连接代理时进行账户名和密码验证。默认值为关闭。

(9) 账户名 / (10) 密码

设置当选中“必须进行身份验证才能连接到代理”复选框时所需的账户名和密码。账户名的最大字符数为 128 个字符，密码的最大字符数为 496 个字符。仅可使用英文数字及符号。

(11) 启用安全连接 (SSL/TLS)

如果连接的代理需要 SSL/TLS 通信，应选中该复选框。默认值为关闭。

(12) 根证书

如果选中“启用安全连接 (SSL/TLS)”复选框，可单击“导入”按钮，读取用于与代理进行 SSL/TLS 通信的服务器证书的根证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件不能被导入。如果 pem 文件包含一个以上的根证书，起始两个根证书将被转换为 der 文件。单击“详情”按钮，将出现“证书信息”对话框，显示读取的服务器证书的根证书信息。单击“删除”按钮，将删除导入的证书。

(13) 客户端证书

如果选中“启用安全连接 (SSL/TLS)”复选框，可单击“导入”按钮，读取用于与代理进行 SSL/TLS 通信的客户端证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件不能被导入。单击“详情”按钮，将出现“证书信息”对话框，显示读取的客户端证书信息。单击“删除”按钮，将删除导入的证书。

(14) 客户端私钥

如果选中“启用安全连接 (SSL/TLS)”复选框，可单击“导入”按钮，读取用于与代理进行 SSL/TLS 通信的客户端私钥（加密方式为 RSA，私钥长度最大为 3072 位）。WindLDR 支持以下格式的文件。

- PKCS#1 格式 pem 或 der 文件
- PKCS#8 格式（未加密） pem 或 der 文件

单击“删除”按钮，将删除导入的私钥。

■ 设备

(15) 连接控制

设置用于连接和断开与代理连接的内部继电器。打开所设置的内部继电器后，连接代理。关闭后断开与代理的连接。

注释： 如果要用数据寄存器设置客户端 ID ((6))，当你把连接控制从关闭切换到开启时，存储在 Plus CPU 模块中的客户端 ID 将被更新。

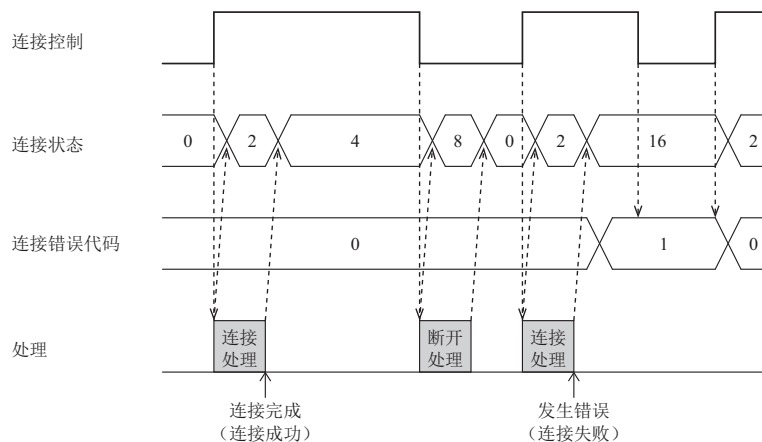
(16) 连接状态

设置存储连接代理时的状态和错误信息的数据寄存器。从指定的数据寄存器开始，使用 2 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

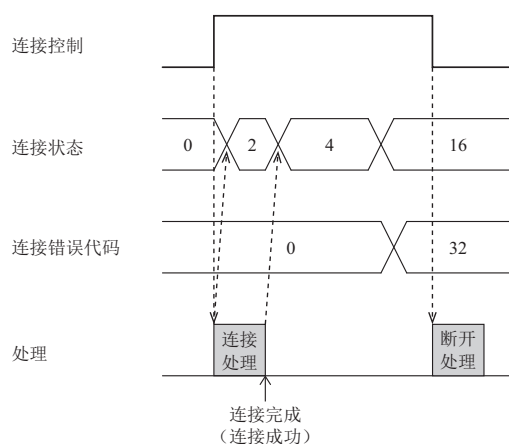
存储目的地	项目	说明																																
起始编号 +0	连接状态	存储连接代理时的状态。																																
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态 (断开状态)</td> </tr> <tr> <td>2 (0002h)</td> <td>连接处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>连接状态</td> </tr> <tr> <td>8 (0008h)</td> <td>断开处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>连接处理错误</td> </tr> <tr> <td>32 (0020h)</td> <td>断开处理错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态 (断开状态)	2 (0002h)	连接处理中	4 (0004h)	连接状态	8 (0008h)	断开处理中	16 (0010h)	连接处理错误	32 (0020h)	断开处理错误																		
		状态代码	状态																															
		0 (0000h)	初始状态 (断开状态)																															
		2 (0002h)	连接处理中																															
		4 (0004h)	连接状态																															
		8 (0008h)	断开处理中																															
		16 (0010h)	连接处理错误																															
32 (0020h)	断开处理错误																																	
起始编号 +1	连接错误代码	存储连接代理时发生的错误信息。																																
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败</td> </tr> <tr> <td>4 (0004h)</td> <td>客户端ID的格式不正确</td> </tr> <tr> <td>16 (0010h)</td> <td>接收到未知数据包</td> </tr> <tr> <td>32 (0020h)</td> <td>接收到不正确的 MQTT 数据包</td> </tr> <tr> <td>64 (0040h)</td> <td>保持活跃超时错误</td> </tr> <tr> <td>80 (0050h)</td> <td>未能到达目标主机</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>112 (0070h)</td> <td>TLS 错误</td> </tr> <tr> <td>256 (0100h)</td> <td>连接代理被拒绝 (无效的 MQTT 协议版本)</td> </tr> <tr> <td>512 (0200h)</td> <td>连接代理被拒绝 (客户端 ID 不正确)</td> </tr> <tr> <td>768 (0300h)</td> <td>连接代理被拒绝 (代理不可用)</td> </tr> <tr> <td>1024 (0400h)</td> <td>连接代理被拒绝 (账户名或密码不正确)</td> </tr> <tr> <td>1280 (0500h)</td> <td>连接代理被拒绝 (无权限)</td> </tr> <tr> <td>32768 (8000h)</td> <td>代理响应异常</td> </tr> </tbody> </table>	错误代码	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败	4 (0004h)	客户端ID的格式不正确	16 (0010h)	接收到未知数据包	32 (0020h)	接收到不正确的 MQTT 数据包	64 (0040h)	保持活跃超时错误	80 (0050h)	未能到达目标主机	96 (0060h)	MQTT 数据包接收超时错误	112 (0070h)	TLS 错误	256 (0100h)	连接代理被拒绝 (无效的 MQTT 协议版本)	512 (0200h)	连接代理被拒绝 (客户端 ID 不正确)	768 (0300h)	连接代理被拒绝 (代理不可用)	1024 (0400h)	连接代理被拒绝 (账户名或密码不正确)	1280 (0500h)	连接代理被拒绝 (无权限)	32768 (8000h)	代理响应异常
		错误代码	错误内容																															
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络																															
		2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败																															
		4 (0004h)	客户端ID的格式不正确																															
		16 (0010h)	接收到未知数据包																															
		32 (0020h)	接收到不正确的 MQTT 数据包																															
		64 (0040h)	保持活跃超时错误																															
		80 (0050h)	未能到达目标主机																															
		96 (0060h)	MQTT 数据包接收超时错误																															
		112 (0070h)	TLS 错误																															
		256 (0100h)	连接代理被拒绝 (无效的 MQTT 协议版本)																															
		512 (0200h)	连接代理被拒绝 (客户端 ID 不正确)																															
		768 (0300h)	连接代理被拒绝 (代理不可用)																															
1024 (0400h)	连接代理被拒绝 (账户名或密码不正确)																																	
1280 (0500h)	连接代理被拒绝 (无权限)																																	
32768 (8000h)	代理响应异常																																	

注释:

- 通过时序图显示 Plus CPU 模块与代理连接时的动作。



- 即使在已连接到代理的状态（连接状态 4（0004h））下，也可能会出现连接错误代码 16（0010h）~ 112（0070h）。



- STOP 时不进行 MQTT 通信。如果在与代理连接期间由 RUN 变为 STOP，则关闭连接控制，并在连接状态中存储断开状态 0（0000h）。

(17) “从项目导入”按钮

关闭“MQTT 设置”对话框，不保存设置。

(18) “确定”按钮

保存设置，关闭“MQTT 设置”对话框。

注释: 如果要用数据寄存器设置客户端 ID ((6))，按“确定”按钮将以客户端 ID 为 1 个字节来计算标题的大小。如果设置的标题超过 256 字节，将发生错误。

(19) “取消”按钮

关闭“MQTT 设置”对话框，不保存设置。

连接至 AWS IoT Core

■ MQTT 基本设置

未选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT
云服务名称: AWS IoT Core

MQTT 基本设置

(1) 用 SD 记忆卡指定

AWS IoT Core

(3) 端点:

(4) 端口号:

(5) 保持活跃: 秒

(6) 客户端 ID:

MAC 地址:

固定值: (7)

数据寄存器:

安全连接 (SSL/TLS)

(8) 根证书: 已导入

(9) 客户端证书: 未导入

(10) 客户端私钥: 未导入

设备

(11) 连接控制:

(12) 连接状态:

(13) (14) (15)

选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT

MQTT 基本设置

用 SD 记忆卡指定 (2)

设备

(1) 用 SD 记忆卡指定

指定是否使用 SD 记忆卡，向 Plus CPU 模块写入 MQTT 通信的基本设置。

选中“用 SD 记忆卡指定”复选框后显示 (2)，(3) ~ (10) 变为不可见。此时，可在“配置 SD 记忆卡”对话框中设置细节。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(2) 配置 SD 记忆卡

打开“配置 SD 记忆卡”对话框。仅当选中“用 SD 记忆卡指定”复选框时显示。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(3) 端点

设置 AWS IoT Core 端点。端点的最大字符数为 128 个字符。只能使用英文数字和符号 (-)。

(4) 端口号

设置 AWS IoT Core 的端口号。在 0 ~ 65535 的范围内设置端口号。默认值为 8883，由 AWS IoT Core 使用。如果端口号被设置为 0，则使用端口号 8883。

(5) 保持活跃

设置在与 AWS IoT Core 连接期间，为 AWS IoT Core 执行连接确认的时间间隔。Plus CPU 模块不收发数据包的时间超过设置的时间间隔时，向 AWS IoT Core 进行连接确认。可设置范围为 5 ~ 65535 秒。默认值为 60 秒。

(6) 客户端 ID

设置客户端 ID。客户端 ID 可以通过以下 3 种方式来设置。

客户端 ID	说明
MAC 地址	设置 Plus CPU 模块上的以太网端口 1 的 MAC 地址*1。
固定值	设置任意字符串。 最大字符数为 128 个字符。仅可使用英文数字及符号。
数据寄存器	设置用于存储客户 ID 的数据寄存器。 从指定的数据寄存器的高位字节中依次读取值，作为字符数据处理，并作为客户端 ID 使用*2。 将使用从指定的数据寄存器开始，连续的 64 字数据寄存器*3。 请指定设备范围内的起始数据寄存器。客户端 ID 只能使用英文数字及符号。

*1 如果 Plus CPU 模块上的以太网端口 1 的 MAC 地址是 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh)，客户端 ID 是 "123456789ABC"。

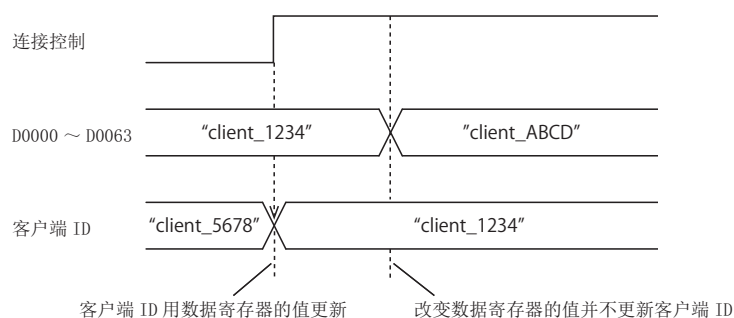
*2 例如：如果指定了 D0000，并且在每个数据寄存器中存储了以下值，那么客户端 ID 就是 "client_1234"。

数据寄存器	存储值	
	高位字节	低位字节
D0000	"c"=63h	"l"=6Ch
D0001	"i"=69h	"e"=65h
D0002	"n"=6Eh	"t"=74h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*3 如果要设置的字符串短于 128 字节（64 个字），请添加一个终止符 NULL（00h）作为字符串的结束。指定的数据寄存器的高位字节到结束字符 NULL（00h）前为止被视为字符数据并作为客户 ID 使用。

注释：

- 如果要用数据寄存器设置客户端 ID，可以用 MOVC（传送字符）指令在数据寄存器中存储字符串。有关 MOVC 指令的详细信息，请参见《梯形图编程手册》第 5 章中的“MOVC（传送字符）”
- 当把连接控制 ((11)) 从关闭切换到开启时，存储在 Plus CPU 模块中的客户端 ID 将被更新。例如：如果使用数据寄存器（D0000）设置客户端 ID，客户端 ID 将在以下时刻更新。

**(7) 产生随机 ID**

如果客户端 ID ((6)) 被设置为一个固定值，单击按钮将输出一个随机的客户端 ID。

(8) 根证书

单击“导入”按钮，导入用于与 AWS IoT Core 进行 SSL/TLS 通信的根证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。如果 pem 文件包含一个以上的根证书，起始两个根证书将被转换为 der 文件。单击“详情”按钮，将出现“证书信息”对话框，显示读取的服务器证书的根证书信息。单击“删除”按钮，将删除导入的证书。

(9) 客户端证书

单击“导入”按钮，导入用于与 AWS IoT Core 进行 SSL/TLS 通信的客户端证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。单击“详情”按钮，将出现“证书信息”对话框，显示读取的客户端证书信息。单击“删除”按钮，将删除导入的证书。

(10) 客户端私钥

单击“导入”按钮，导入用于与 AWS IoT Core 进行 SSL/TLS 通信的客户端私钥（加密方式为 RSA，私钥长度最大为 3072 位）。WindLDR 支持以下格式的文件。

- PKCS#1 格式 pem 或 der 文件
- PKCS#8 格式（未加密） pem 或 der 文件

单击“删除”按钮，将删除导入的私钥。

■ 设备

(11) 连接控制

设置用于连接和断开与 AWS IoT Core 连接的内部继电器。打开所设置的内部继电器后，连接 AWS IoT Core。关闭后断开与 AWS IoT Core 的连接。

注释：如果要用数据寄存器设置客户端 ID ((6))，当你把连接控制从关闭切换到开启时，存储在 Plus CPU 模块中的客户端 ID 将被更新。

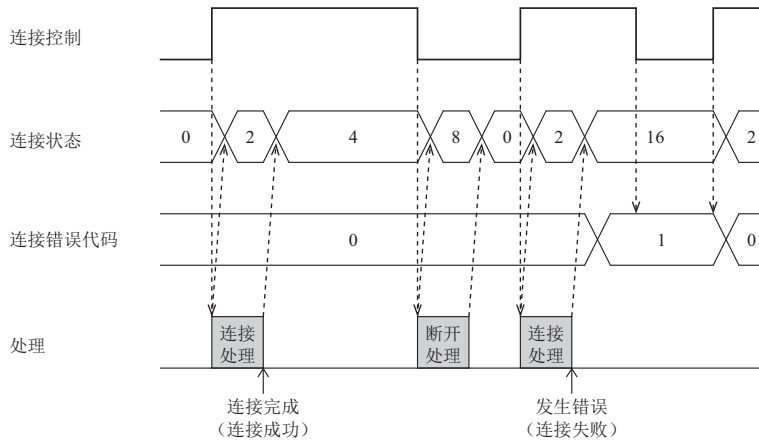
(12) 连接状态

设置存储连接 AWS IoT Core 时的状态和错误信息的数据寄存器。从指定的数据寄存器开始，使用 2 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

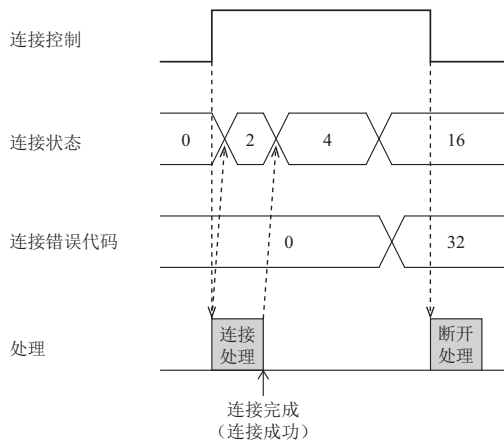
存储目的地	项目	说明																																
起始编号 +0	连接状态	存储连接 AWS IoT Core 时的状态。																																
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态 (断开状态)</td> </tr> <tr> <td>2 (0002h)</td> <td>连接处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>连接状态</td> </tr> <tr> <td>8 (0008h)</td> <td>断开处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>连接处理错误</td> </tr> <tr> <td>32 (0020h)</td> <td>断开处理错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态 (断开状态)	2 (0002h)	连接处理中	4 (0004h)	连接状态	8 (0008h)	断开处理中	16 (0010h)	连接处理错误	32 (0020h)	断开处理错误																		
		状态代码	状态																															
		0 (0000h)	初始状态 (断开状态)																															
		2 (0002h)	连接处理中																															
		4 (0004h)	连接状态																															
		8 (0008h)	断开处理中																															
		16 (0010h)	连接处理错误																															
32 (0020h)	断开处理错误																																	
起始编号 +1	连接错误代码	存储连接 AWS IoT Core 时发生的错误信息。																																
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>未从 SD 记忆卡下载验证信息，或者下载验证信息读取失败</td> </tr> <tr> <td>4 (0004h)</td> <td>客户端ID的格式不正确</td> </tr> <tr> <td>16 (0010h)</td> <td>接收到未知数据包</td> </tr> <tr> <td>32 (0020h)</td> <td>接收到不正确的 MQTT 数据包</td> </tr> <tr> <td>64 (0040h)</td> <td>保持活跃超时错误</td> </tr> <tr> <td>80 (0050h)</td> <td>未能到达目标主机</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>112 (0070h)</td> <td>TLS 错误</td> </tr> <tr> <td>256 (0100h)</td> <td>连接 AWS IoT Core 被拒绝 (无效的 MQTT 协议版本)</td> </tr> <tr> <td>512 (0200h)</td> <td>连接 AWS IoT Core 被拒绝 (客户端 ID 不正确)</td> </tr> <tr> <td>768 (0300h)</td> <td>连接 AWS IoT Core 被拒绝 (AWS IoT Core 不可用)</td> </tr> <tr> <td>1024 (0400h)</td> <td>连接 AWS IoT Core 被拒绝 (账户名或密码不正确)</td> </tr> <tr> <td>1280 (0500h)</td> <td>连接 AWS IoT Core 被拒绝 (无权限)</td> </tr> <tr> <td>32768 (8000h)</td> <td>AWS IoT Core 响应异常</td> </tr> </tbody> </table>	错误代码	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载验证信息读取失败	4 (0004h)	客户端ID的格式不正确	16 (0010h)	接收到未知数据包	32 (0020h)	接收到不正确的 MQTT 数据包	64 (0040h)	保持活跃超时错误	80 (0050h)	未能到达目标主机	96 (0060h)	MQTT 数据包接收超时错误	112 (0070h)	TLS 错误	256 (0100h)	连接 AWS IoT Core 被拒绝 (无效的 MQTT 协议版本)	512 (0200h)	连接 AWS IoT Core 被拒绝 (客户端 ID 不正确)	768 (0300h)	连接 AWS IoT Core 被拒绝 (AWS IoT Core 不可用)	1024 (0400h)	连接 AWS IoT Core 被拒绝 (账户名或密码不正确)	1280 (0500h)	连接 AWS IoT Core 被拒绝 (无权限)	32768 (8000h)	AWS IoT Core 响应异常
		错误代码	错误内容																															
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络																															
		2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载验证信息读取失败																															
		4 (0004h)	客户端ID的格式不正确																															
		16 (0010h)	接收到未知数据包																															
		32 (0020h)	接收到不正确的 MQTT 数据包																															
		64 (0040h)	保持活跃超时错误																															
		80 (0050h)	未能到达目标主机																															
		96 (0060h)	MQTT 数据包接收超时错误																															
		112 (0070h)	TLS 错误																															
		256 (0100h)	连接 AWS IoT Core 被拒绝 (无效的 MQTT 协议版本)																															
		512 (0200h)	连接 AWS IoT Core 被拒绝 (客户端 ID 不正确)																															
		768 (0300h)	连接 AWS IoT Core 被拒绝 (AWS IoT Core 不可用)																															
1024 (0400h)	连接 AWS IoT Core 被拒绝 (账户名或密码不正确)																																	
1280 (0500h)	连接 AWS IoT Core 被拒绝 (无权限)																																	
32768 (8000h)	AWS IoT Core 响应异常																																	

注释:

- 通过时序图显示 Plus CPU 模块与 AWS IoT Core 连接时的动作。



- 即使在已连接到 AWS IoT Core 的状态（连接状态 4（0004h））下，也可能会出现连接错误代码 16（0010h）~ 112（0070h）。



- STOP 时不进行 MQTT 通信。如果在与 AWS IoT Core 连接期间由 RUN 变为 STOP，则关闭连接控制，并在连接状态中存储断开状态 0（0000h）。

(13) “从项目导入”按钮

关闭“MQTT 设置”对话框，不保存设置。

(14) “确定”按钮

保存设置，关闭“MQTT 设置”对话框。

注释: 如果要用数据寄存器设置客户端 ID ((6))，按“确定”按钮将以客户端 ID 为 1 个字节来计算标题的大小。如果设置的标题超过 256 字节，将发生错误。

(15) “取消”按钮

关闭“MQTT 设置”对话框，不保存设置。

使用 SAS 连接至 Azure IoT Hub

■ MQTT 连接方法

在“MQTT 连接方法”组中，选择“直接连接至 Azure IoT Hub”并设置“使用 Shared Access Signature (SAS)”。

■ MQTT 基本设置

未选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT
云服务名称: Azure IoT Hub

MQTT 连接方法

直接连接至 Azure IoT Hub: 使用 Shared Access Signatures(SAS)

通过 DPS 连接至 Azure IoT Hub

MQTT 基本设置

(1) 用 SD 记忆卡指定

(3) 连接字符串:

Azure IoT Hub

(4) 主机名称: N/A

(5) 端口号: 8883

(6) 保持活跃: 60 秒

设备

(7) 设备 ID: N/A

安全连接(SSL/TLS)

(8) 根证书: 已导入

设备

(9) 连接控制: ...

(10) 连接状态: (-)

(11) (12) (13)

选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT

MQTT 基本设置

用 SD 记忆卡指定 (2)

设备

(1) 用 SD 记忆卡指定

指定是否使用 SD 记忆卡向 Plus CPU 模块写入 MQTT 通信的基本设置。

选中“用 SD 记忆卡指定”复选框后显示 (2)，(3) ~ (8) 变为不可见。此时，可在“配置 SD 记忆卡”对话框中设置细节。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(2) 配置 SD 记忆卡

打开“配置 SD 记忆卡”对话框。仅当选“用 SD 记忆卡指定”复选框时显示。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(3) 连接字符串

设置分配给 Azure IoT Hub 中 Plus CPU 模块的连接字符串（主要或次要连接字符串）。连接字符串包含 Azure IoT Hub 的主机名称，设备 ID 和 SharedAccessKey。连接字符串的最大字符数为 300 个字符。仅可使用英文数字及符号。

(4) 主机名称

显示连接字符串中 HostName 的值。

例如：HostName=abcd.azure-devices.net 将显示 abcd.azure-devices.net 作为主机名称。

(5) 端口号

设置 Azure IoT Hub 的端口号。在 0 ~ 65535 的范围内设置端口号。默认值为由 Azure IoT Hub 使用的 8883。如果端口号被设置为 0，则使用端口号 8883。

(6) 保持活跃

设置与 Azure IoT Hub 连接时，执行 Azure IoT Hub 连接确认的时间间隔。Plus CPU 模块不收发数据包的时间超过设置的时间间隔时，向 Azure IoT Hub 进行连接确认。可设置范围为 5 ~ 65535 秒。默认值为 60 秒。

(7) 设备 ID

显示连接字符串中 DeviceId 的值为设备 ID。

例如：DeviceId=1234 将显示 1234 作为设备 ID。设备 ID 被用作 MQTT 通信的客户端 ID。

(8) 根证书

单击“导入”按钮，导入用于与 Azure IoT Hub 进行 SSL/TLS 通信的根证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。如果 pem 文件包含一个以上的根证书，起始两个根证书将被转换为 der 文件。单击“详情”按钮，将出现“证书信息”对话框，显示读取的服务器证书的根证书信息。单击“删除”按钮，将删除导入的证书。

■ 设备

(9) 连接控制

设置用于连接和断开与 Azure IoT Hub 连接的内部继电器。从指定的内部继电器开始，使用 5 个位的内部继电器。请指定设备范围内的起始内部继电器。

存储目的地	项目	说明
起始编号 +0	连接控制 *1	执行与 Azure IoT Hub 连接及断开。打开该内部继电器后，连接 Azure IoT Hub。关闭该内部继电器后，断开与 Azure IoT Hub 的连接。
起始编号 +1	保留	—
起始编号 +2	保留	—
起始编号 +3	保留	—
起始编号 +4	保留	—

*1 在生成 SAS 形符以连接到 Azure IoT Hub 时，使用了从 Plus CPU 模块时钟计算出来的当前时间（UTC）。如果 Plus CPU 模块的时钟与当前实际时间相差甚远，Plus CPU 模块可能无法连接到 Azure IoT Hub。请设置 Plus CPU 模块的时钟和时钟相关功能。

当从 SNTP 服务器获取当前时间（UTC）时

参考以下设置项目，设置 Plus CPU 模块的时钟和与时钟有关的功能。

设置项目	说明	参考
SNTP 设置	从 SNTP 服务器获取当前时间（UTC）并调整 Plus CPU 模块上的时钟。	第 3-11 页上的“SNTP 设置”
时区	设置该地区不同时区相对于标准时间的差异。	《FC6A 型 MICROSmart 用户手册》第 5 章“时区”
夏时制	设置夏时制的时间段。	《FC6A 型 MICROSmart 用户手册》第 5 章“夏时制”

注释： Plus CPU 模块上的时钟会随着时间的推移而出现偏差，这取决于其精度。如果距离上次从 SNTP 服务器检索当前时间（UTC）已超过 6 个月，可能无法再连接到 Azure IoT Hub。以适当的频率从 SNTP 服务器获取当前时间（UTC）。

当直接配置 Plus CPU 模块的时钟时

请参考以下设置项目，设置 Plus CPU 模块的时钟和与时钟有关的功能。

设置项目	说明	参考
时钟功能	使用 WindLDR 或特殊数据寄存器和特殊内部继电器配置 Plus CPU 模块的时钟。	《FC6A 型 MICROSmart 用户手册》第 5 章“时钟功能”
时区	设置该地区不同时区相对于标准时间的差异。	《FC6A 型 MICROSmart 用户手册》第 5 章“时区”
夏时制	设置夏时制的时间段。	《FC6A 型 MICROSmart 用户手册》第 5 章“夏时制”

注释： Plus CPU 模块上的时钟会随着时间的推移而出现偏差，这取决于其精度。如果自上次设置时钟以来已经超过 6 个月，它可能不再连接到 Azure IoT Hub。请以适当的频率从 SNTP 服务器获取当前时间（UTC）。

(10) 连接状态

设置存储连接 Azure IoT Hub 时的状态和错误信息的数据寄存器。从指定的数据寄存器开始，使用 5 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

存储目的地	项目	说明																														
起始编号 +0	连接状态	存储连接 Azure IoT Hub 时的状态。																														
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态（断开状态）</td> </tr> <tr> <td>2 (0002h)</td> <td>连接处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>连接状态</td> </tr> <tr> <td>8 (0008h)</td> <td>断开处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>连接处理错误</td> </tr> <tr> <td>32 (0020h)</td> <td>断开处理错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态（断开状态）	2 (0002h)	连接处理中	4 (0004h)	连接状态	8 (0008h)	断开处理中	16 (0010h)	连接处理错误	32 (0020h)	断开处理错误																
		状态代码	状态																													
		0 (0000h)	初始状态（断开状态）																													
		2 (0002h)	连接处理中																													
		4 (0004h)	连接状态																													
		8 (0008h)	断开处理中																													
		16 (0010h)	连接处理错误																													
32 (0020h)	断开处理错误																															
起始编号 +1	连接错误代码	存储连接 Azure IoT Hub 时发生的错误信息。																														
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败</td> </tr> <tr> <td>16 (0010h)</td> <td>接收到未知数据包</td> </tr> <tr> <td>32 (0020h)</td> <td>接收到不正确的 MQTT 数据包</td> </tr> <tr> <td>64 (0040h)</td> <td>保持活跃超时错误</td> </tr> <tr> <td>80 (0050h)</td> <td>未能到达目标主机</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>112 (0070h)</td> <td>TLS 错误</td> </tr> <tr> <td>256 (0100h)</td> <td>连接 Azure IoT Hub 被拒绝（无效的 MQTT 协议版本）</td> </tr> <tr> <td>512 (0200h)</td> <td>连接 Azure IoT Hub 被拒绝（设备 ID 不正确）*1</td> </tr> <tr> <td>768 (0300h)</td> <td>连接 Azure IoT Hub 被拒绝（Azure IoT Hub 不可用）</td> </tr> <tr> <td>1024 (0400h)</td> <td>连接 Azure IoT Hub 被拒绝（账户名或密码不正确）*1*2</td> </tr> <tr> <td>1280 (0500h)</td> <td>连接 Azure IoT Hub 被拒绝（无权限）</td> </tr> <tr> <td>32768 (8000h)</td> <td>Azure IoT Hub 响应异常</td> </tr> </tbody> </table>	错误代码	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败	16 (0010h)	接收到未知数据包	32 (0020h)	接收到不正确的 MQTT 数据包	64 (0040h)	保持活跃超时错误	80 (0050h)	未能到达目标主机	96 (0060h)	MQTT 数据包接收超时错误	112 (0070h)	TLS 错误	256 (0100h)	连接 Azure IoT Hub 被拒绝（无效的 MQTT 协议版本）	512 (0200h)	连接 Azure IoT Hub 被拒绝（设备 ID 不正确）*1	768 (0300h)	连接 Azure IoT Hub 被拒绝（Azure IoT Hub 不可用）	1024 (0400h)	连接 Azure IoT Hub 被拒绝（账户名或密码不正确）*1*2	1280 (0500h)	连接 Azure IoT Hub 被拒绝（无权限）	32768 (8000h)	Azure IoT Hub 响应异常
		错误代码	错误内容																													
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络																													
		2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败																													
		16 (0010h)	接收到未知数据包																													
		32 (0020h)	接收到不正确的 MQTT 数据包																													
		64 (0040h)	保持活跃超时错误																													
		80 (0050h)	未能到达目标主机																													
		96 (0060h)	MQTT 数据包接收超时错误																													
		112 (0070h)	TLS 错误																													
		256 (0100h)	连接 Azure IoT Hub 被拒绝（无效的 MQTT 协议版本）																													
		512 (0200h)	连接 Azure IoT Hub 被拒绝（设备 ID 不正确）*1																													
		768 (0300h)	连接 Azure IoT Hub 被拒绝（Azure IoT Hub 不可用）																													
1024 (0400h)	连接 Azure IoT Hub 被拒绝（账户名或密码不正确）*1*2																															
1280 (0500h)	连接 Azure IoT Hub 被拒绝（无权限）																															
32768 (8000h)	Azure IoT Hub 响应异常																															
起始编号 +2	保留	—																														
起始编号 +3	保留	—																														
起始编号 +4	保留	—																														

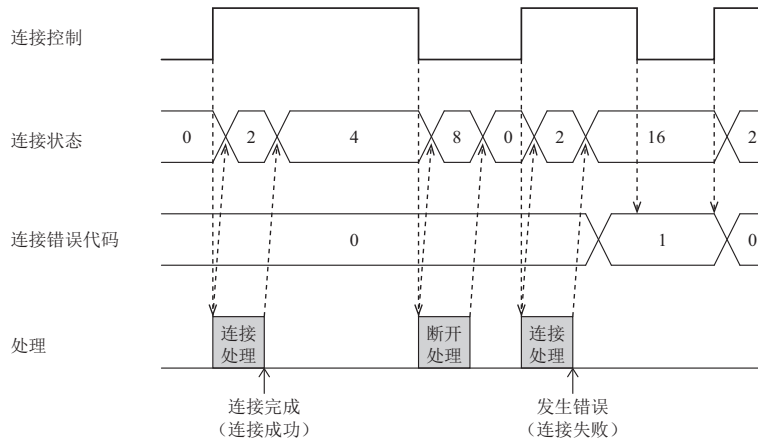
*1 请确认连接字符串中的 DeviceId 的值。

*2 设置 Plus CPU 模块的时钟和时钟相关功能。

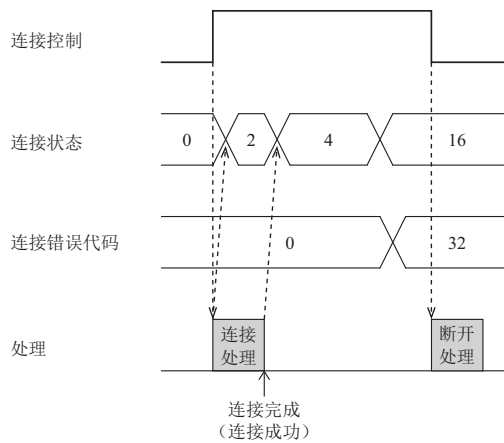
17: MQTT 通信

注释:

- 通过时序图显示 Plus CPU 模块与 Azure IoT Hub 连接时的动作。



- 即使在已连接到 Azure IoT Hub 的状态 (连接状态 4 (0004h)) 下, 也可能出现连接错误代码 16 (0010h) ~ 112 (0070h)。



- STOP 时不进行 MQTT 通信。如果在与 Azure IoT Hub 连接期间由 RUN 变为 STOP, 则关闭连接控制, 并在连接状态中存储断开状态 0 (0000h)。

(11) “从项目导入”按钮

关闭“MQTT 设置”对话框, 不保存设置。

(12) “确定”按钮

保存设置, 关闭“MQTT 设置”对话框。

(13) “取消”按钮

关闭“MQTT 设置”对话框, 不保存设置。

使用 X.509 证书连接至 Azure IoT Hub

■ MQTT 连接方法

在“MQTT 连接方法”组中，选择“直接连接至 Azure IoT Hub”并设置“使用 X.509 证书”。

■ MQTT 基本设置

未选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT
云服务名称: Azure IoT Hub

MQTT 连接方法

直接连接至 Azure IoT Hub: 使用 X.509 证书
 通过 DPS 连接至 Azure IoT Hub

MQTT 基本设置

(1) 用 SD 记忆卡指定

Azure IoT Hub

(3) 主机名称:

(4) 端口号: 8883

(5) 保持活跃: 60 秒

设备

(6) 设备 ID:

必须进行身份验证才能连接到代理

(7) 账户名: (8) 生成

安全连接 (SSL/TLS)

(9) 根证书:	已导入	导入	详情	删除
(10) 客户端证书:	未导入	导入	详情	删除
(11) 客户端私钥:	未导入	导入		删除

设备

(12) 连接控制: ...

(13) 连接状态: ... (-)

(14) 从项目导入 (I) (15) 确定 (16) 取消

选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT

MQTT 基本设置

用 SD 记忆卡指定 (2) 配置 SD 记忆卡

设备

(1) 用 SD 记忆卡指定

指定是否使用 SD 记忆卡，向 Plus CPU 模块写入 MQTT 通信的基本设置。

选中“用 SD 记忆卡指定”复选框后显示 (2)，(3) ~ (11) 变为不可见。此时，可在“配置 SD 记忆卡”对话框中设置细节。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(2) 配置 SD 记忆卡

打开“配置 SD 记忆卡”对话框。仅当选中“用 SD 记忆卡指定”复选框时显示。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(3) 主机名称

设置 Azure IoT Hub 的主机名称。主机名称的最大字符数为 128 个字符。只能使用英文数字及符号 (-)。

(4) 端口号

设置 Azure IoT Hub 的端口号。在 0 ~ 65535 的范围内设置端口号。默认值为 8883 由 Azure IoT Hub 使用。如果端口号被设置为 0，则使用端口号 8883。

(5) 保持活跃

设置与 Azure IoT Hub 连接时，执行 Azure IoT Hub 连接确认的时间间隔。Plus CPU 模块不收发数据包的时间超过设置的时间间隔时，向 Azure IoT Hub 进行连接确认。可设置范围为 5 ~ 65535 秒。默认值为 60 秒。

(6) 设备 ID

设置注册到 Azure IoT Hub 的 Plus CPU 模块的设备 ID。设备 ID 的最大字符数为 128 个字符。仅可使用英文数字及符号。设备 ID 被用作 MQTT 通信的客户端 ID。

(7) 账户名

设置用于连接至 Azure IoT Hub 的账户名。账户名的最大字符数为 128 个字符。仅可使用英文数字及符号。

(8) 生成

单击按钮，可以根据主机名称 ((3)) 和设备 ID ((6)) 生成用于连接 Azure IoT Hub 的账户名。

(9) 根证书

单击“导入”按钮，导入用于与 Azure IoT Hub 进行 SSL/TLS 通信的根证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。如果 pem 文件包含一个以上的根证书，起始两个根证书将被转换为 der 文件。单击“详情”按钮，将出现“证书信息”对话框，显示读取的服务器证书的根证书信息。单击“删除”按钮，将删除导入的证书。

(10) 客户端证书

单击“导入”按钮，导入用于与 Azure IoT Hub 进行 SSL/TLS 通信的客户端证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。单击“详情”按钮，将出现“证书信息”对话框，显示读取的客户端证书信息。单击“删除”按钮，将删除导入的证书。

(11) 客户端私钥

单击“导入”按钮，导入用于与 Azure IoT Hub 进行 SSL/TLS 通信的客户端私钥（加密方式为 RSA，私钥长度最大为 3072 位）。WindLDR 支持以下格式的文件。

- PKCS#1 格式 pem 或 der 文件
- PKCS#8 格式（未加密）pem 或 der 文件

单击“删除”按钮，将删除导入的私钥。

■ 设备

(12) 连接控制

设置用于连接和断开与 Azure IoT Hub 连接的内部继电器。从指定的内部继电器开始，使用 5 个位的内部继电器。请指定设备范围内的起始内部继电器。

存储目的地	项目	说明
起始编号 +0	连接控制	连接至 Azure IoT Hub 并断开其连接。 当打开时，连接至 Azure IoT Hub。将其关闭会断开与 Azure IoT Hub 的连接。
起始编号 +1	保留	—
起始编号 +2	保留	—
起始编号 +3	保留	—
起始编号 +4	保留	—

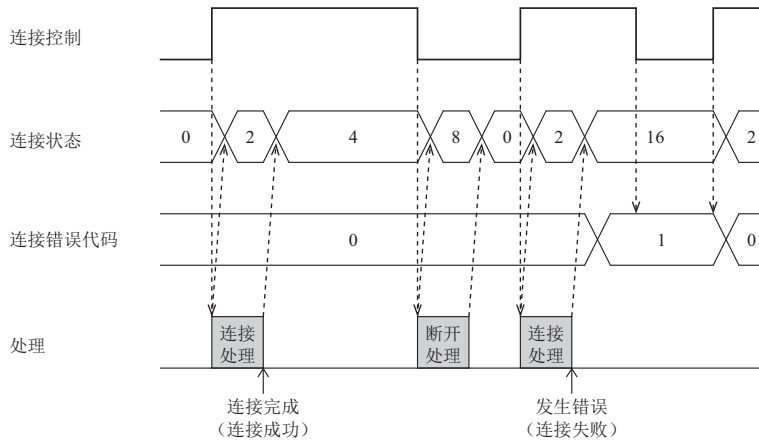
(13) 连接状态

设置存储连接 Azure IoT Hub 时的状态和错误信息的数据寄存器。从指定的数据寄存器开始，使用 5 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

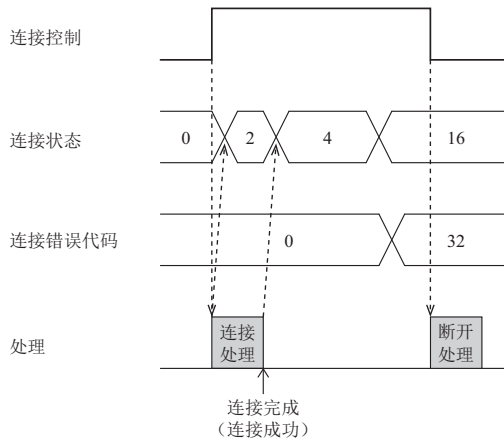
存储目的地	项目	说明																																
起始编号 +0	连接状态	存储连接 Azure IoT Hub 时的状态。																																
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态 (断开状态)</td> </tr> <tr> <td>2 (0002h)</td> <td>连接处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>连接状态</td> </tr> <tr> <td>8 (0008h)</td> <td>断开处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>连接处理错误</td> </tr> <tr> <td>32 (0020h)</td> <td>断开处理错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态 (断开状态)	2 (0002h)	连接处理中	4 (0004h)	连接状态	8 (0008h)	断开处理中	16 (0010h)	连接处理错误	32 (0020h)	断开处理错误																		
		状态代码	状态																															
		0 (0000h)	初始状态 (断开状态)																															
		2 (0002h)	连接处理中																															
		4 (0004h)	连接状态																															
		8 (0008h)	断开处理中																															
		16 (0010h)	连接处理错误																															
32 (0020h)	断开处理错误																																	
起始编号 +1	连接错误代码	存储连接 Azure IoT Hub 时发生的错误信息。																																
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败</td> </tr> <tr> <td>4 (0004h)</td> <td>客户端ID的格式不正确</td> </tr> <tr> <td>16 (0010h)</td> <td>接收到未知数据包</td> </tr> <tr> <td>32 (0020h)</td> <td>接收到不正确的 MQTT 数据包</td> </tr> <tr> <td>64 (0040h)</td> <td>保持活跃超时错误</td> </tr> <tr> <td>80 (0050h)</td> <td>未能到达目标主机</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>112 (0070h)</td> <td>TLS 错误</td> </tr> <tr> <td>256 (0100h)</td> <td>连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)</td> </tr> <tr> <td>512 (0200h)</td> <td>连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)</td> </tr> <tr> <td>768 (0300h)</td> <td>连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)</td> </tr> <tr> <td>1024 (0400h)</td> <td>连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)</td> </tr> <tr> <td>1280 (0500h)</td> <td>连接 Azure IoT Hub 被拒绝 (无权限)</td> </tr> <tr> <td>32768 (8000h)</td> <td>Azure IoT Hub 响应异常</td> </tr> </tbody> </table>	错误代码	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败	4 (0004h)	客户端ID的格式不正确	16 (0010h)	接收到未知数据包	32 (0020h)	接收到不正确的 MQTT 数据包	64 (0040h)	保持活跃超时错误	80 (0050h)	未能到达目标主机	96 (0060h)	MQTT 数据包接收超时错误	112 (0070h)	TLS 错误	256 (0100h)	连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)	512 (0200h)	连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)	768 (0300h)	连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)	1024 (0400h)	连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)	1280 (0500h)	连接 Azure IoT Hub 被拒绝 (无权限)	32768 (8000h)	Azure IoT Hub 响应异常
		错误代码	错误内容																															
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络																															
		2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败																															
		4 (0004h)	客户端ID的格式不正确																															
		16 (0010h)	接收到未知数据包																															
		32 (0020h)	接收到不正确的 MQTT 数据包																															
		64 (0040h)	保持活跃超时错误																															
		80 (0050h)	未能到达目标主机																															
		96 (0060h)	MQTT 数据包接收超时错误																															
		112 (0070h)	TLS 错误																															
		256 (0100h)	连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)																															
		512 (0200h)	连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)																															
		768 (0300h)	连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)																															
1024 (0400h)	连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)																																	
1280 (0500h)	连接 Azure IoT Hub 被拒绝 (无权限)																																	
32768 (8000h)	Azure IoT Hub 响应异常																																	
起始编号 +2	保留	—																																
起始编号 +3	保留	—																																
起始编号 +4	保留	—																																

注释:

- 通过时序图显示 Plus CPU 模块与 Azure IoT Hub 连接时的动作。



- 即使在已连接到 Azure IoT Hub 的状态（连接状态 4（0004h））下，也可能会出现连接错误代码 16（0010h）~ 112（0070h）。



- STOP 时不进行 MQTT 通信。如果在与 Azure IoT Hub 连接期间由 RUN 变为 STOP，则关闭连接控制，并在连接状态中存储断开状态 0（0000h）。

(14) “从项目导入”按钮

关闭“MQTT 设置”对话框，不保存设置。

(15) “确定”按钮

保存设置，关闭“MQTT 设置”对话框。

(16) “取消”按钮

关闭“MQTT 设置”对话框，不保存设置。

通过 DPS 连接至 Azure IoT Hub

■ MQTT 连接方法

在“MQTT 连接方法”组中，选择“通过 DPS 连接至 Azure IoT Hub”。

■ MQTT 基本设置

未选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT
云服务名称: Azure IoT Hub

MQTT 连接方法

直接连接至 Azure IoT Hub: 使用 Shared Access Signatures (SAS)

通过 DPS 连接至 Azure IoT Hub

MQTT 基本设置

(1) 用 SD 记忆卡指定

Azure IoT Hub

(3) 主机名称: [] (-)

(4) 端口号: 8883

(5) 保持活跃: 60 秒

设备

(6) 设备 ID: MAC 地址: 固定值: 2aa65e3c-4b07-4375-9191-b687be8f76df (7) 产生随机 ID 数据寄存器: [] (-)

Device Provisioning Service (DPS)

(8) 服务端点: []

(9) ID 范围: []

登记组:

(10) 对称密钥: []

(11) 端口号: 8883

安全连接 (SSL/TLS)

(12) 根证书: 已导入 [导入] [详情] [删除]

设备

(13) 连接控制: []

(14) 连接状态: [] (-)

(15) [从项目导入] (16) [确定] (17) [取消]

选中“用 SD 记忆卡指定”复选框时

MQTT 设置

启用 MQTT

MQTT 基本设置

用 SD 记忆卡指定 (2) [配置 SD 记忆卡]

设备

(1) 用 SD 记忆卡指定

指定是否使用 SD 记忆卡，向 Plus CPU 模块写入 MQTT 通信的基本设置。

选中“用 SD 记忆卡指定”复选框后显示 (2)，(3) ~ (12) 变为不可见。此时，可在“配置 SD 记忆卡”对话框中设置细节。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(2) 配置 SD 记忆卡

打开“配置 SD 记忆卡”对话框。仅当选中“用 SD 记忆卡指定”复选框时显示。有关“配置 SD 记忆卡”对话框的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

(3) 主机名称

设置数据寄存器以存储从 DPS 得到的 Azure IoT Hub 的主机名称（ASCII 码）。从指定的数据寄存器开始，使用 64 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

(4) 端口号

设置 Azure IoT Hub 的端口号。使用与 Device Provisioning Service (DPS) 端口号（11）相同的端口号。

(5) 保持活跃

设置与 Azure IoT Hub 连接时，执行 Azure IoT Hub 连接确认的时间间隔。Plus CPU 模块不收发数据包的时间超过设置的时间间隔时，向 Azure IoT Hub 进行连接确认。可设置范围为 5 ~ 65535 秒。默认值为 60 秒。

(6) 设备 ID

设置设备 ID，通过 DPS 注册到 Azure IoT Hub。设备 ID 可以从以下 3 种类型中设置，也可作为 MQTT 通信的客户端 ID。

设备 ID	说明
MAC 地址	设置 Plus CPU 模块上的以太网端口 1 的 MAC 地址*1。
固定值	设置任意字符串。 最大字符数为 128 个字符。仅可使用英文数字及符号。
数据寄存器	设置用于存储设备 ID 的数据寄存器。 从指定的数据寄存器的高位字节中依次读出该值，作为字符数据处理，并作为设备 ID 使用*2。 从指定的数据寄存器开始，使用连续的 64 字数据寄存器*3。 请指定设备范围内的起始数据寄存器。设备 ID 只能使用英文数字及符号。

*1 如果 Plus CPU 模块上的以太网端口 1 的 MAC 地址是 12-34-56-78-9A-BC（D8324=0012h，D8325=0034h，D8326=0056h，D8327=0078h，D8328=009Ah，D8329=00BCh），客户端 ID 是 "123456789ABC"。

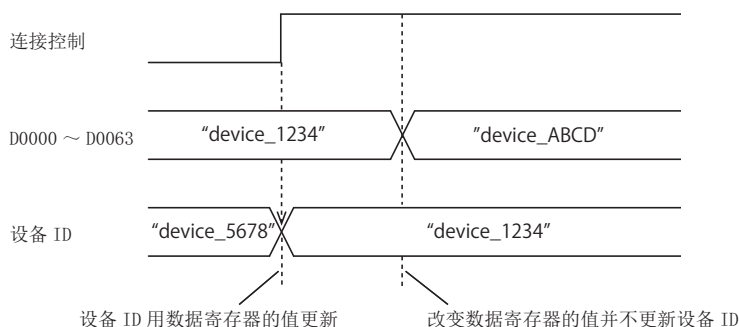
*2 例如：如果指定了 D0000，并且在每个数据寄存器中存储了以下值，那么客户端 ID 就是 "device_1234"。

数据寄存器	存储值	
	高位字节	低位字节
D0000	"d"=64h	"e"=65h
D0001	"v"=76h	"i"=69h
D0002	"c"=63h	"e"=65h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*3 如果要设置的字符串短于 128 字节（64 个字），请添加一个终止符 NULL（00h）作为字符串的结束。指定的数据寄存器的高位字节到结束字符 NULL（00h）前为止被视为字符数据并作为客户 ID 使用。

注释:

- 如果要用数据寄存器设置设备 ID，可以用 MOVC（传送字符）指令在数据寄存器中存储字符串。有关 MOVC 指令的详细信息，请参见《梯形图编程手册》第 5 章中的“MOVC（传送字符）”。
- 当你把连接控制 ((13)) 从关闭切换到开启时，存储在 Plus CPU 模块中的设备 ID 将被更新。例如：如果使用数据寄存器（D0000）设置设备 ID，设备 ID 将在以下时间更新。

**(7) 产生随机 ID**

如果设备 ID ((6)) 被设置为一个固定值，单击按钮将输出一个随机的设备 ID。

(8) 服务端点

设置 DPS 的服务端点。设备 ID 的最大字符数为 81 个字符。仅可使用英文数字及符号。

(9) ID 范围

设置 DPS 的 ID 范围。ID 范围的最大字符数为 11 个字符。仅可使用英文数字。

(10) 对称密钥

设置 Plus CPU 模块所注册的 DPS 的注册组的对称密钥（主密钥或次密钥）。对称密钥的最大字符数为 88 个字符。仅可使用英文数字及符号。

(11) 端口号

设置 Device Provisioning Service (DPS) 的端口号。在 0 ~ 65535 的范围内设置端口号。默认值为由 Azure IoT Hub 使用的 8883。如果端口号被设置为 0，则使用端口号 8883。

(12) 根证书

单击“导入”按钮，导入用于与 DPS 和 Azure IoT Hub 进行 SSL/TLS 通信的根证书。导入的文件可以是 pem 或 crt 格式文件。WindLDR 将 pem 和 crt 格式文件转换为 der 格式文件。大于 2K 字节的 der 格式文件无法被导入。如果 pem 文件包含一个以上的根证书，起始两个根证书将被转换为 der 文件。单击“详情”按钮，将出现“证书信息”对话框，显示读取的服务器证书的根证书信息。单击“删除”按钮，将删除导入的证书。

■ 设备

(13) 连接控制

设置用于连接和断开与 DPS 和 Azure IoT Hub 连接的内部继电器。从指定的内部继电器开始，使用 5 个位的内部继电器。请指定设备范围内的起始内部继电器。

存储目的地	项目	说明
起始编号 +0	连接控制 *1	连接至 DPS 和 Azure IoT Hub 并断开连接。 当连接控制开启时，Plus CPU 模块会根据 DPS 连接许可的状态（起始编号 +1）连接至 DPS 和 Azure IoT Hub。关闭连接控制会断开与 DPS 和 Azure IoT Hub 的连接。
起始编号 +1	DPS 连接许可	允许连接至 DPS。 当 DPS 连接许可打开时，打开连接控制（起始编号 +0），Plus CPU 模块将通过 DPS 连接至 Azure IoT Hub。*2 当 DPS 连接许可关闭时，打开连接控制（起始编号 +0），Plus CPU 模块将直接连接至 Azure IoT Hub，直接连接到存储在主机名称（(3)）中的 Azure IoT Hub，而不通过 DPS。*3
起始编号 +2	保留	—
起始编号 +3	保留	—
起始编号 +4	保留	—

*1 在生成 SAS 形符以连接至 DPS 和 Azure IoT Hub 时，使用了从 Plus CPU 模块时钟计算出来的当前时间（UTC）。如果 Plus CPU 模块的时钟与当前实际时间相差甚远，Plus CPU 模块可能无法连接至 DPS 和 Azure IoT Hub。请设置 Plus CPU 模块的时钟和时钟相关功能。

当从 SNTP 服务器获取当前时间（UTC）时

参考以下设置项目，设置 Plus CPU 模块的时钟和与时钟有关的功能。

设置项目	说明	参考
SNTP 设置	从 SNTP 服务器获取当前时间（UTC）并调整 Plus CPU 模块上的时钟。	第 3-11 页上的“SNTP 设置”
时区	设置该地区不同时区相对于标准时间的差异。	《FC6A 型 MICROSmart 用户手册》第 5 章“时区”
夏时制	设置夏时制的时间段。	《FC6A 型 MICROSmart 用户手册》第 5 章“夏时制”

注释： Plus CPU 模块上的时钟会随着时间的推移而出现偏差，这取决于其精度。如果距离上次从 SNTP 服务器检索当前时间（UTC）已超过 6 个月，可能无法再连接至 DPS 和 Azure IoT Hub。以适当的频率从 SNTP 服务器获取当前时间（UTC）。

当直接配置 Plus CPU 模块的时钟时

请参考以下设置项目，设置 Plus CPU 模块的时钟和与时钟有关的功能。

设置项目	说明	参考
时钟功能	使用 WindLDR 或特殊数据寄存器和特殊内部继电器配置 Plus CPU 模块的时钟。	《FC6A 型 MICROSmart 用户手册》第 5 章“时钟功能”
时区	设置该地区不同时区相对于标准时间的差异。	《FC6A 型 MICROSmart 用户手册》第 5 章“时区”
夏时制	设置夏时制的时间段。	《FC6A 型 MICROSmart 用户手册》第 5 章“夏时制”

注释： Plus CPU 模块上的时钟会随着时间的推移而出现偏差，这取决于其精度。如果自上次设置时钟以来已经超过 6 个月，它可能不再连接至 DPS 和 Azure IoT Hub。以适当的频率从 SNTP 服务器获取当前时间（UTC）。

*2 此时，从 DPS 获得的 Azure IoT Hub 的主机名称被存储在主机名称（(3)）中设置的数据寄存器。

*3 如果设备 ID 已经改变，请再次连接至 DPS 来注册设备。

(14) 连接状态

设置存储连接至 DPS 和 Azure IoT Hub 时的状态和错误信息的数据寄存器。从指定的数据寄存器开始，使用 5 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

存储目的地	项目	说明																																												
起始编号 +0	连接状态	存储连接至 DPS 和 Azure IoT Hub 时的状态。																																												
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态 (断开状态)</td> </tr> <tr> <td>2 (0002h)</td> <td>Azure IoT Hub 连接处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>Azure IoT Hub 连接状态</td> </tr> <tr> <td>8 (0008h)</td> <td>Azure IoT Hub 断开处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>Azure IoT Hub 连接处理错误</td> </tr> <tr> <td>32 (0020h)</td> <td>Azure IoT Hub 断开处理错误</td> </tr> <tr> <td>512 (0200h)</td> <td>DPS 连接处理中</td> </tr> <tr> <td>1024 (0400h)</td> <td>DPS 连接状态</td> </tr> <tr> <td>2048 (0800h)</td> <td>DPS 断开处理中</td> </tr> <tr> <td>4096 (1000h)</td> <td>DPS 连接处理错误</td> </tr> <tr> <td>8192 (2000h)</td> <td>DPS 断开处理错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态 (断开状态)	2 (0002h)	Azure IoT Hub 连接处理中	4 (0004h)	Azure IoT Hub 连接状态	8 (0008h)	Azure IoT Hub 断开处理中	16 (0010h)	Azure IoT Hub 连接处理错误	32 (0020h)	Azure IoT Hub 断开处理错误	512 (0200h)	DPS 连接处理中	1024 (0400h)	DPS 连接状态	2048 (0800h)	DPS 断开处理中	4096 (1000h)	DPS 连接处理错误	8192 (2000h)	DPS 断开处理错误																				
		状态代码	状态																																											
		0 (0000h)	初始状态 (断开状态)																																											
		2 (0002h)	Azure IoT Hub 连接处理中																																											
		4 (0004h)	Azure IoT Hub 连接状态																																											
		8 (0008h)	Azure IoT Hub 断开处理中																																											
		16 (0010h)	Azure IoT Hub 连接处理错误																																											
		32 (0020h)	Azure IoT Hub 断开处理错误																																											
		512 (0200h)	DPS 连接处理中																																											
		1024 (0400h)	DPS 连接状态																																											
		2048 (0800h)	DPS 断开处理中																																											
		4096 (1000h)	DPS 连接处理错误																																											
8192 (2000h)	DPS 断开处理错误																																													
起始编号 +1	连接错误代码	存储连接至 DPS 和 Azure IoT Hub 时发生的错误信息。																																												
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败</td> </tr> <tr> <td>4 (0004h)</td> <td>客户端ID的格式不正确</td> </tr> <tr> <td>8 (0008h)</td> <td>数据寄存器中存储的 Azure IoT Hub 的主机名称不正确。</td> </tr> <tr> <td>16 (0010h)</td> <td>接收到未知数据包</td> </tr> <tr> <td>32 (0020h)</td> <td>接收到不正确的 MQTT 数据包</td> </tr> <tr> <td>64 (0040h)</td> <td>保持活跃超时错误</td> </tr> <tr> <td>80 (0050h)</td> <td>未能到达目标主机 *1</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>112 (0070h)</td> <td>TLS 错误</td> </tr> <tr> <td>256 (0100h)</td> <td>连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)</td> </tr> <tr> <td>512 (0200h)</td> <td>连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)</td> </tr> <tr> <td>768 (0300h)</td> <td>连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)</td> </tr> <tr> <td>1024 (0400h)</td> <td>连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)</td> </tr> <tr> <td>1280 (0500h)</td> <td>连接 Azure IoT Hub 被拒绝 (无权限) *2*3*4</td> </tr> <tr> <td>4352 (1100h)</td> <td>连接 DPS 被拒绝 (无效的 MQTT 协议版本)</td> </tr> <tr> <td>4608 (1200h)</td> <td>连接 DPS 被拒绝 (设备 ID 不正确)</td> </tr> <tr> <td>4864 (1300h)</td> <td>连接 DPS 被拒绝 (Azure IoT Hub 不可用)</td> </tr> <tr> <td>5120 (1400h)</td> <td>连接 DPS 被拒绝 (账户名或密码不正确)</td> </tr> <tr> <td>5376 (1500h)</td> <td>连接 DPS 被拒绝 (无权限) *2*3*4</td> </tr> <tr> <td>32768 (8000h)</td> <td>Azure IoT Hub/ DPS 响应异常</td> </tr> </tbody> </table>	错误代码	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败	4 (0004h)	客户端ID的格式不正确	8 (0008h)	数据寄存器中存储的 Azure IoT Hub 的主机名称不正确。	16 (0010h)	接收到未知数据包	32 (0020h)	接收到不正确的 MQTT 数据包	64 (0040h)	保持活跃超时错误	80 (0050h)	未能到达目标主机 *1	96 (0060h)	MQTT 数据包接收超时错误	112 (0070h)	TLS 错误	256 (0100h)	连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)	512 (0200h)	连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)	768 (0300h)	连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)	1024 (0400h)	连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)	1280 (0500h)	连接 Azure IoT Hub 被拒绝 (无权限) *2*3*4	4352 (1100h)	连接 DPS 被拒绝 (无效的 MQTT 协议版本)	4608 (1200h)	连接 DPS 被拒绝 (设备 ID 不正确)	4864 (1300h)	连接 DPS 被拒绝 (Azure IoT Hub 不可用)	5120 (1400h)	连接 DPS 被拒绝 (账户名或密码不正确)	5376 (1500h)	连接 DPS 被拒绝 (无权限) *2*3*4	32768 (8000h)	Azure IoT Hub/ DPS 响应异常
		错误代码	错误内容																																											
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络																																											
		2 (0002h)	未从 SD 记忆卡下载验证信息，或者下载的验证信息读取失败																																											
		4 (0004h)	客户端ID的格式不正确																																											
		8 (0008h)	数据寄存器中存储的 Azure IoT Hub 的主机名称不正确。																																											
		16 (0010h)	接收到未知数据包																																											
		32 (0020h)	接收到不正确的 MQTT 数据包																																											
		64 (0040h)	保持活跃超时错误																																											
		80 (0050h)	未能到达目标主机 *1																																											
		96 (0060h)	MQTT 数据包接收超时错误																																											
		112 (0070h)	TLS 错误																																											
		256 (0100h)	连接 Azure IoT Hub 被拒绝 (无效的 MQTT 协议版本)																																											
		512 (0200h)	连接 Azure IoT Hub 被拒绝 (设备 ID 不正确)																																											
		768 (0300h)	连接 Azure IoT Hub 被拒绝 (Azure IoT Hub 不可用)																																											
		1024 (0400h)	连接 Azure IoT Hub 被拒绝 (账户名或密码不正确)																																											
		1280 (0500h)	连接 Azure IoT Hub 被拒绝 (无权限) *2*3*4																																											
		4352 (1100h)	连接 DPS 被拒绝 (无效的 MQTT 协议版本)																																											
		4608 (1200h)	连接 DPS 被拒绝 (设备 ID 不正确)																																											
4864 (1300h)	连接 DPS 被拒绝 (Azure IoT Hub 不可用)																																													
5120 (1400h)	连接 DPS 被拒绝 (账户名或密码不正确)																																													
5376 (1500h)	连接 DPS 被拒绝 (无权限) *2*3*4																																													
32768 (8000h)	Azure IoT Hub/ DPS 响应异常																																													
起始编号 +2	保留	—																																												

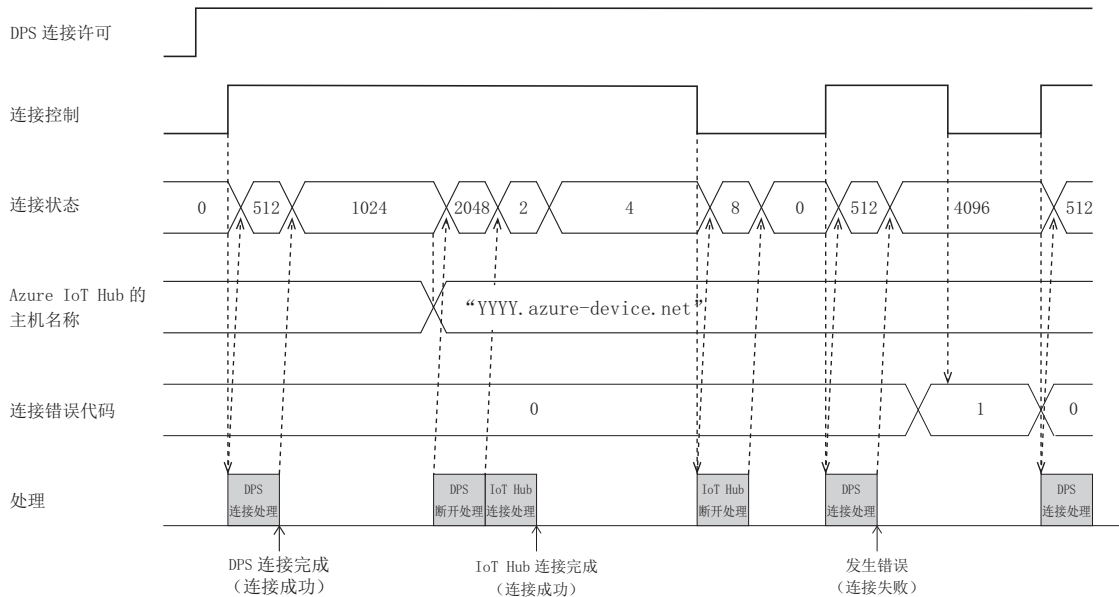
17: MQTT通信

存储目的地	项目	说明
起始编号 +3	保留	—
起始编号 +4	保留	—

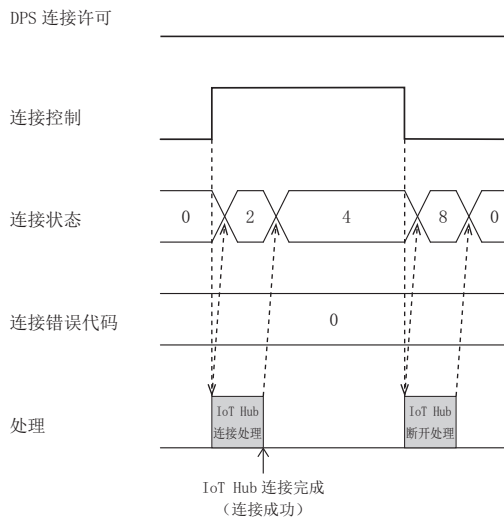
- *1 检查 DPS 服务端点或 Azure IoT Hub 的主机名称。
- *2 检查服务端点 ((8)), ID 范围 ((9)) 和对称密钥 ((10))。
- *3 设置 Plus CPU 模块的时钟和时钟相关功能。
- *4 检查 Plus CPU 模块是否在 DPS 和 Azure IoT Hub 中注册。如果有必要, 重新连接至 DPS。

注释:

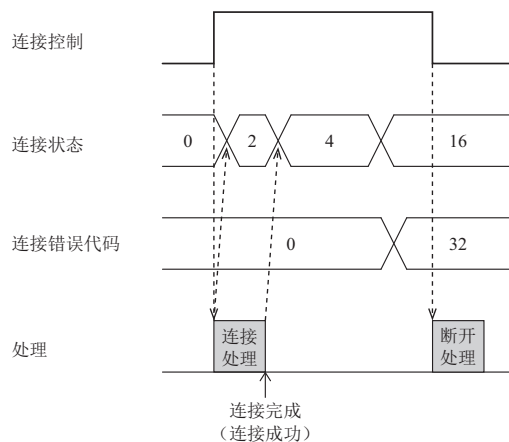
- 当 DPS 连接许可处于开启状态, 通过时序图显示 Plus CPU 模块与 DPS 和 Azure IoT Hub 连接时的动作。



- 时序图显示了 DPS 连接许可关闭时, Plus CPU 模块直接连接至 Azure IoT Hub 时的表现。



- 即使在已连接到 Azure IoT Hub 的状态（连接状态 4（0004h））下，也可能出现连接错误代码 16（0010h）～ 112（0070h）。



- STOP 时不进行 MQTT 通信。如果在与 Azure IoT Hub 连接期间由 RUN 变为 STOP，则关闭连接控制，并在连接状态中存储断开状态 0（0000h）。

(15) “从项目导入”按钮

关闭“MQTT 设置”对话框，不保存设置。

(16) “确定”按钮

保存设置，关闭“MQTT 设置”对话框。

注释：如果要用数据寄存器设置客户端 ID ((6))，按“确定”按钮将以客户端 ID 为 1 个字节来计算标题的大小。如果设置的标题超过 256 字节，将发生错误。

(17) “取消”按钮

关闭“MQTT 设置”对话框，不保存设置。

发布

在“MQTT 设置”对话框中选择“发布”后，将显示“发布”选项卡。在该选项卡中执行发布设置。



(1) 按钮

删除选定行的设置内容。

(2) 标题

设置标题。标题为 UTF-8 编码，最多可设置 256 字节。

单击 按钮，将显示“标题”对话框。有关详情，请参见第 17-39 页上的““标题”对话框”。

注释：

- 最多可设置 10 个标题。
- 当你把发布控制 ((5)) 从关闭切换到开启时，在“标题”对话框中设置的标题将被生成。如果生成的标题是无效的，你将无法向该标题发布。有关错误的详情，请参见发布状态 ((6))。
- 包含 +、# 或 / 的客户端 ID 不能用于标题。

(3) 有效载荷

设置有效载荷内的数据。单击“设置”按钮，将显示“有效载荷”对话框。有关详情，请参见第 17-41 页上的““有效载荷”对话框”。

(4) 操作模式

设置操作模式。单击操作模式，将显示“操作模式设置”对话框。有关详情，请参见第 17-36 页上的““操作模式设置”对话框”。

(5) 发布控制

设置用于发布标题的内部继电器。打开所设置的内部继电器后，根据操作模式 ((4)) 发布标题。

(6) 发布状态

设置存储发布时状态等的的数据寄存器。从指定的数据寄存器开始，使用 4 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

存储目的地	项目	说明																
起始编号 +0	发布状态	存储发布时的状态。																
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态</td> </tr> <tr> <td>2 (0002h)</td> <td>发布处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>发布完成状态</td> </tr> <tr> <td>16 (0010h)</td> <td>发布错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态	2 (0002h)	发布处理中	4 (0004h)	发布完成状态	16 (0010h)	发布错误						
		状态代码	状态															
		0 (0000h)	初始状态															
		2 (0002h)	发布处理中															
4 (0004h)	发布完成状态																	
16 (0010h)	发布错误																	
起始编号 +1	发布错误代码	存储发布时发生的错误信息。																
		<table border="1"> <thead> <tr> <th>错误代码 *1</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>在连接状态为 0 (0000h) (断开状态) 时发布</td> </tr> <tr> <td>4 (0004h)</td> <td>你试图发布到一个格式不正确的标题，或发布到一个大小超过 256 字节的标题</td> </tr> <tr> <td>5 (0005h)</td> <td>尝试发布大小大于 32768 字节的 JSON 格式数据</td> </tr> <tr> <td>7 (0007h)</td> <td>当数据类型为浮点时发布了 $\pm\infty$ (\pm 无限大) 或非数，或者当数据类型为字符串时发布了非 UTF-8 的字符码</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>32768 (8000h)</td> <td>代理响应异常</td> </tr> </tbody> </table>	错误代码 *1	错误内容	1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络	2 (0002h)	在连接状态为 0 (0000h) (断开状态) 时发布	4 (0004h)	你试图发布到一个格式不正确的标题，或发布到一个大小超过 256 字节的标题	5 (0005h)	尝试发布大小大于 32768 字节的 JSON 格式数据	7 (0007h)	当数据类型为浮点时发布了 $\pm\infty$ (\pm 无限大) 或非数，或者当数据类型为字符串时发布了非 UTF-8 的字符码	96 (0060h)	MQTT 数据包接收超时错误	32768 (8000h)	代理响应异常
		错误代码 *1	错误内容															
		1 (0001h)	以太网电缆脱落或断线， Plus CPU 模块未能正确连接到网络															
		2 (0002h)	在连接状态为 0 (0000h) (断开状态) 时发布															
		4 (0004h)	你试图发布到一个格式不正确的标题，或发布到一个大小超过 256 字节的标题															
		5 (0005h)	尝试发布大小大于 32768 字节的 JSON 格式数据															
		7 (0007h)	当数据类型为浮点时发布了 $\pm\infty$ (\pm 无限大) 或非数，或者当数据类型为字符串时发布了非 UTF-8 的字符码															
96 (0060h)	MQTT 数据包接收超时错误																	
32768 (8000h)	代理响应异常																	
起始编号 +2	发送数据字节数	发布状态变为 4 (0004h) 时，以字节为单位存储发送数据的大小。																
起始编号 +3	发布错误 ID	发布错误代码变为 7 (0007h) 时，存储发生错误的 ID*2。																

*1 当发布错误代码为 4 (0004h) 到 7 (0007h) 以外的代码时，连接控制也将关闭。当发布错误代码为 4 (0004h) 到 7 (0007h) 时，保持发布控制和连接控制的状态。

*2 ID 是在“有效载荷”对话框中用于识别各要素而分配的编号。有关详情，请参见第 17-41 页上的““有效载荷”对话框”。

(7) QoS

在 0 ~ 2 范围内设置 QoS。默认值为 0。

(8) Retain

设置 Retain 的启用 / 禁用。选中复选框时启用 Retain。默认值为关闭。

(9) 占用设备地址

显示因设置发布控制和发布状态而被占用的设备地址。

(10) 将发布失败的 MQTT 数据包保存在至 SD 记忆卡并重新发布

设置未能发布的 MQTT 数据包是否被存储在 SD 记忆卡上并再次发布。勾选这个复选框可以再次发布，这取决于重新发布控件的状态 ((11))。

注释：

- 发布失败的 MQTT 数据包将作为重新发布的数据存储在 SD 记忆卡上。发布重新发布数据被称为重新发布。
- 重新发布时，将 QoS ((7)) 设置为 1 或 2。如果 QoS ((7)) 被设置为 0，这个 MQTT 数据包就不会作为重新发布的数据存储在 SD 记忆卡上，因为在没有来自 MQTT 代理的响应时不会发生错误。
- 发布和订阅「第 17-37 页上的“订阅”」过程优先于重新发布过程。
- 如果客户端 ID 被改变，在旧客户端 ID 下保存的重新发布数据将被删除。

(11) 重新发布控制

设置一个内部继电器来重新发布。从指定的内部继电器开始，使用 5 个位的内部继电器。请指定设备范围内的起始内部继电器。

存储目的地	项目	说明
起始编号 +0	重新发布许可	允许重新发布。 当重新发布许可为打开时，如果重新发布的数据存储在 SD 记忆卡上，它将被重新发布。 ^{*1*2*3} 当重新发布许可是关闭时，即使重新发布的数据存储在 SD 记忆卡上，也不会重新发布。
起始编号 +1	重新发布数据存储许可	允许在 SD 记忆卡上存储重新发布的数据。 当重新发布数据存储许可为打开时，未能发布的 MQTT 数据包 ^{*4} 将作为重新发布的数据存储在 SD 记忆卡上。 ^{*5} 当重新发布数据存储许可关闭时，如果发布失败，MQTT 数据包不会存储在 SD 记忆卡上。
起始编号 +2	保留	—
起始编号 +3	保留	—
起始编号 +4	保留	—

*1 重新发布存储在 SD 记忆卡上的数据，不分先后。如有需要请，在有效载荷上设置一个时间戳。

*2 如果在重新发布成功到删除重新发布的数据中途，SD 记忆卡被移除或 Plus CPU 模块被关闭，重新发布的数据可能不会从 SD 记忆卡中删除。在这种情况下，当你下次连接到 MQTT 代理时，它将重新发布未被删除的重新发布数据。

*3 在执行完重新发布 200 毫秒以上后，会执行下一次重新发布。重新发布过程所需的时间，以及完成重新发布过程到下一次重新发布过程之间的时间，在以下情况下会增加。

使用其他功能（如配方功能，FTP 服务器 / 客户端功能）或指令（如 DLOG，TRACE 指令）来访问 SD 记忆卡。
执行发布或者订阅。

*4 如果发布错误代码（(6) 为 4（0004h）至 7（0007h），则无法生成 MQTT 数据包，因为主题或有效载荷无效。在这种情况下，重新发布的数据不会存储在 SD 记忆卡上。

*5 如果在发布失败和重新发布数据被保存期间，SD 记忆卡被移除或 Plus CPU 模块被关闭，重新发布的数据可能不会被保存到 SD 记忆卡。

注释： 在安装或移除 SD 记忆卡时，请关闭重新发布许可和重新发布数据存储许可。

(12) 重新发布状态

设置数据寄存器，用于存储重新发布的数据或重新发布时存储状态和其他信息。从指定的数据寄存器开始，使用 5 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

存储目的地	项目	说明																		
起始编号 +0	重新发布状态	重新发布 在保存或重新发布数据时存储数据的状态。																		
		<table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态</td> </tr> <tr> <td>1 (0001h)</td> <td>没有重新发布的数据</td> </tr> <tr> <td>2 (0002h)</td> <td>重新发布处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>重新发布完成状态</td> </tr> <tr> <td>16 (0010h)</td> <td>重新发布错误 *1</td> </tr> <tr> <td>512 (0200h)</td> <td>重新发布的数据被保存</td> </tr> <tr> <td>1024 (0400h)</td> <td>重新发布的数据保存完成</td> </tr> <tr> <td>2048 (0800h)</td> <td>重新发布的数据保存错误 *1</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态	1 (0001h)	没有重新发布的数据	2 (0002h)	重新发布处理中	4 (0004h)	重新发布完成状态	16 (0010h)	重新发布错误 *1	512 (0200h)	重新发布的数据被保存	1024 (0400h)	重新发布的数据保存完成	2048 (0800h)	重新发布的数据保存错误 *1
		状态代码	状态																	
		0 (0000h)	初始状态																	
		1 (0001h)	没有重新发布的数据																	
		2 (0002h)	重新发布处理中																	
		4 (0004h)	重新发布完成状态																	
		16 (0010h)	重新发布错误 *1																	
		512 (0200h)	重新发布的数据被保存																	
1024 (0400h)	重新发布的数据保存完成																			
2048 (0800h)	重新发布的数据保存错误 *1																			
起始编号 +1	重新发布错误代码	存储发布时发生的错误信息。 *2																		
		<table border="1"> <thead> <tr> <th>错误代码</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>3 (0003h)</td> <td>客户端 ID 不被记住，所以重新发布的数据不能被存储 *3*6</td> </tr> <tr> <td>8 (0008h)</td> <td>由于无效的主题或有效载荷，无法生成重新发布数据 *3</td> </tr> <tr> <td>16 (0010h)</td> <td>没有插入 SD 记忆卡 *3</td> </tr> <tr> <td>17 (0011h)</td> <td>SD 记忆卡上的可用空间小于如果可用空间大于指定的大小，它将被保存 ((13)) 中设定的大小 *3</td> </tr> <tr> <td>18 (0012h)</td> <td>由于 SD 记忆卡的读 / 写错误，无法保存重新发布的数据 *3</td> </tr> <tr> <td>19 (0013h)</td> <td>由于 SD 记忆卡的读 / 写错误，无法重新发布 *4</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误 *4*5</td> </tr> <tr> <td>32768 (8000h)</td> <td>代理响应异常 *4*5</td> </tr> </tbody> </table>	错误代码	错误内容	3 (0003h)	客户端 ID 不被记住，所以重新发布的数据不能被存储 *3*6	8 (0008h)	由于无效的主题或有效载荷，无法生成重新发布数据 *3	16 (0010h)	没有插入 SD 记忆卡 *3	17 (0011h)	SD 记忆卡上的可用空间小于如果可用空间大于指定的大小，它将被保存 ((13)) 中设定的大小 *3	18 (0012h)	由于 SD 记忆卡的读 / 写错误，无法保存重新发布的数据 *3	19 (0013h)	由于 SD 记忆卡的读 / 写错误，无法重新发布 *4	96 (0060h)	MQTT 数据包接收超时错误 *4*5	32768 (8000h)	代理响应异常 *4*5
		错误代码	错误内容																	
		3 (0003h)	客户端 ID 不被记住，所以重新发布的数据不能被存储 *3*6																	
		8 (0008h)	由于无效的主题或有效载荷，无法生成重新发布数据 *3																	
		16 (0010h)	没有插入 SD 记忆卡 *3																	
		17 (0011h)	SD 记忆卡上的可用空间小于如果可用空间大于指定的大小，它将被保存 ((13)) 中设定的大小 *3																	
		18 (0012h)	由于 SD 记忆卡的读 / 写错误，无法保存重新发布的数据 *3																	
		19 (0013h)	由于 SD 记忆卡的读 / 写错误，无法重新发布 *4																	
96 (0060h)	MQTT 数据包接收超时错误 *4*5																			
32768 (8000h)	代理响应异常 *4*5																			
起始编号 +2	保留	—																		
起始编号 +3	保留	—																		
起始编号 +4	保留	—																		

*1 下次重新发布或保存重新发布的数据时，重新发布状态会被更新。

*2 下一次重新发布失败或重新发布数据存储失败时，重新发布错误代码会被更新。

*3 当重新发布的数据被存储到 SD 记忆卡时发生的错误。在这种情况下，重新发布状态是 2048 (0800h)。

*4 执行重新发布时发生的错误。在这种情况下，重新发布状态是 16 (0010h)。

*5 如果重新发布错误代码是 96 (0060h) 到 32768 (8000h)，连接控制关闭。

*6 请将连接控制 (起始编号 +0) 从关闭转到开启，以存储客户端 ID。

注释：重新发布失败的数据不会从 SD 记忆卡中删除。

(13) 如果可用空间大于指定的大小，它将被保存

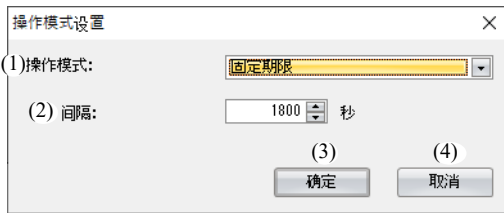
设置 SD 记忆卡上的可用空间 (64MB/ 128MB/ 256MB/ 512MB/ 1024MB)，以存储重新发布数据。默认值是 256MB。

当 SD 记忆卡上的可用空间大于指定大小时，发布失败的 MQTT 数据包将作为重新发布的数据进行记忆。

当 SD 记忆卡上的可用空间小于指定大小时，即使发布失败，MQTT 数据包也不会被保存为重新发布的数据。

“操作模式设置”对话框

在“操作模式设置”对话框中设置发布时的操作模式。



(1) 操作模式

设置发布时的操作模式。

操作模式	说明
上升沿	将发布控制从关闭设为打开时，发布指定的标题。发布完成后，发布控制自动关闭。
固定期限	当发布控制打开时，以在间隔（(2)）中设置的周期，发布指定的标题。关闭发布控制后，发布结束。

(2) 间隔

在 1 ~ 3600 秒的范围内设置从开始发布处理到开始下一次发布处理的时间间隔。

默认值为 60 秒。在开始下一次发布处理时，如果上一次的发布处理还未完成，下一次发布处理的开始会进行等待。

(3) “确定”按钮

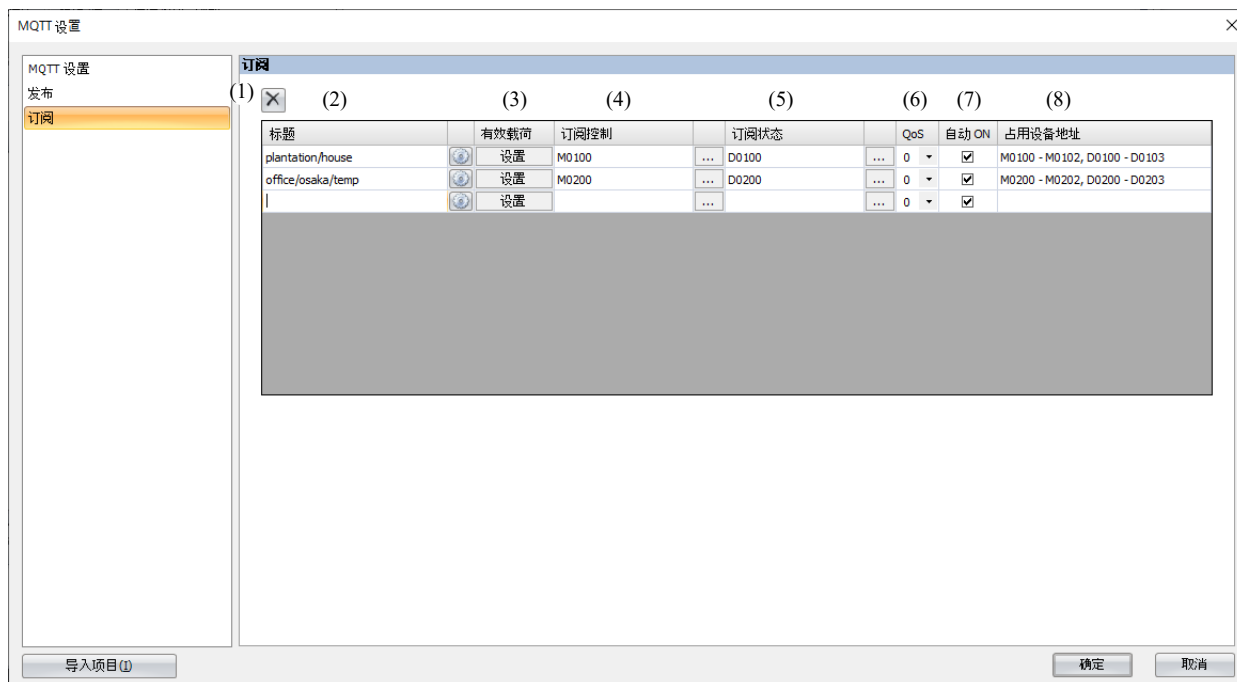
单击“确定”按钮后，设置操作模式和间隔。

(4) “取消”按钮

单击“取消”按钮后，不会保存操作模式和间隔。

订阅

在“MQTT 设置”对话框中选择“订阅”后，将显示“订阅”选项卡。在该选项卡中执行订阅设置。



(1) 按钮

删除选定行的设置内容。

(2) 标题

设置标题。标题为 UTF-8 编码，最多可设置 256 字节。此外还可以设置通配符（# 和 +）。

单击 按钮，将显示“标题”对话框。有关详情，请参见第 17-39 页上的““标题”对话框”。

注释：

- 最多可设置 10 个标题。
- 当你把订阅控制 ((4)) 从关闭切换到开启时，在“标题”对话框中设置的标题将被生成。如果生成的标题是无效的，你就不能订阅它。有关错误的详情，请参见订阅状态 ((5))。
- 包含 +、# 或 / 的客户端 ID 不能用于标题。

(3) 有效载荷

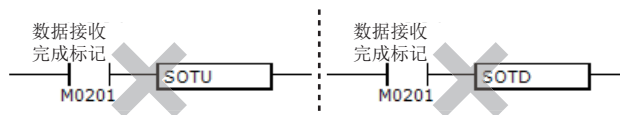
设置有效载荷内的数据。单击“设置”按钮，将显示“有效载荷”对话框。有关详情，请参见第 17-41 页上的““有效载荷”对话框”。

(4) 订阅控制

设置用于订阅标题的内部继电器。从指定的内部继电器开始，使用 3 个位的内部继电器。请指定设备范围内的起始内部继电器。

存储目的地	项目	说明
起始编号 +0	订阅控制	打开时订阅指定的标题。关闭时取消订阅。
起始编号 +1	数据接收完成标记	正常接收数据时（订阅状态保持为 4（0004h））打开。如果想要检测下一次数据接收，请通过梯形图程序关闭此项。
起始编号 +2	所有数据存储标记	正常接收数据时（订阅状态保持为 4（0004h）），以及在“有效载荷”对话框中设置的所有设备中存储值时打开。

注释：请勿将 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令与数据接收完成标记的输入结合使用。如果将这些指令与数据接收完成标记的输入结合使用，数据接收可能不会被识别。



(5) 订阅状态

设置存储订阅标题时状态等的的数据寄存器。从指定的数据寄存器开始，使用 4 个字的数据寄存器。请指定设备范围内的起始数据寄存器。

存储目的地	项目	说明																				
起始编号 +0	订阅状态	存储订阅状态。 <table border="1"> <thead> <tr> <th>状态代码</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0 (0000h)</td> <td>初始状态</td> </tr> <tr> <td>2 (0002h)</td> <td>订阅处理中</td> </tr> <tr> <td>4 (0004h)</td> <td>订阅中 (订阅错误代码为 5 (0005h) ~ 8 (0008h) 时, 状态为 20 (0014h))</td> </tr> <tr> <td>8 (0008h)</td> <td>订阅取消处理中</td> </tr> <tr> <td>16 (0010h)</td> <td>订阅错误</td> </tr> <tr> <td>32 (0020h)</td> <td>订阅取消错误</td> </tr> </tbody> </table>	状态代码	状态	0 (0000h)	初始状态	2 (0002h)	订阅处理中	4 (0004h)	订阅中 (订阅错误代码为 5 (0005h) ~ 8 (0008h) 时, 状态为 20 (0014h))	8 (0008h)	订阅取消处理中	16 (0010h)	订阅错误	32 (0020h)	订阅取消错误						
状态代码	状态																					
0 (0000h)	初始状态																					
2 (0002h)	订阅处理中																					
4 (0004h)	订阅中 (订阅错误代码为 5 (0005h) ~ 8 (0008h) 时, 状态为 20 (0014h))																					
8 (0008h)	订阅取消处理中																					
16 (0010h)	订阅错误																					
32 (0020h)	订阅取消错误																					
起始编号 +1	订阅错误代码	存储订阅时发生的错误信息。 <table border="1"> <thead> <tr> <th>错误代码 *1</th> <th>错误内容</th> </tr> </thead> <tbody> <tr> <td>1 (0001h)</td> <td>以太网电缆脱落或断线, Plus CPU 模块未能正确连接到网络</td> </tr> <tr> <td>2 (0002h)</td> <td>在连接状态为 0 (0000h) (断开状态) 时订阅或取消订阅</td> </tr> <tr> <td>4 (0004h)</td> <td>你试图订阅一个格式不正确的标题, 或试图订阅一个大小超过 256 字节的标题</td> </tr> <tr> <td>5 (0005h)</td> <td>接收数据的大小或要素数超出范围</td> </tr> <tr> <td>6 (0006h)</td> <td>接收的数据不是 JSON 格式</td> </tr> <tr> <td>7 (0007h)</td> <td>接收到数据类型范围以外的值</td> </tr> <tr> <td>8 (0008h)</td> <td>未在“有效载荷”对话框设置的所有设备中存储值</td> </tr> <tr> <td>96 (0060h)</td> <td>MQTT 数据包接收超时错误</td> </tr> <tr> <td>32768 (8000h)</td> <td>代理响应异常</td> </tr> </tbody> </table>	错误代码 *1	错误内容	1 (0001h)	以太网电缆脱落或断线, Plus CPU 模块未能正确连接到网络	2 (0002h)	在连接状态为 0 (0000h) (断开状态) 时订阅或取消订阅	4 (0004h)	你试图订阅一个格式不正确的标题, 或试图订阅一个大小超过 256 字节的标题	5 (0005h)	接收数据的大小或要素数超出范围	6 (0006h)	接收的数据不是 JSON 格式	7 (0007h)	接收到数据类型范围以外的值	8 (0008h)	未在“有效载荷”对话框设置的所有设备中存储值	96 (0060h)	MQTT 数据包接收超时错误	32768 (8000h)	代理响应异常
错误代码 *1	错误内容																					
1 (0001h)	以太网电缆脱落或断线, Plus CPU 模块未能正确连接到网络																					
2 (0002h)	在连接状态为 0 (0000h) (断开状态) 时订阅或取消订阅																					
4 (0004h)	你试图订阅一个格式不正确的标题, 或试图订阅一个大小超过 256 字节的标题																					
5 (0005h)	接收数据的大小或要素数超出范围																					
6 (0006h)	接收的数据不是 JSON 格式																					
7 (0007h)	接收到数据类型范围以外的值																					
8 (0008h)	未在“有效载荷”对话框设置的所有设备中存储值																					
96 (0060h)	MQTT 数据包接收超时错误																					
32768 (8000h)	代理响应异常																					
起始编号 +2	接收数据字节数	以字节为单位存储接收数据的大小。																				
起始编号 +3	订阅错误 ID	当发生以下任一错误时, 存储最初发生错误的 ID*2。 <ul style="list-style-type: none"> 在接收到的 JSON 格式数据中, 没有与“有效载荷”对话框中所设置 ID 的深度级数、名称以及格式一致的要素 接收到数据类型范围以外的值 																				

*1 当订阅错误代码为 4 (0004h) 到 8 (0008h) 以外的代码时, 连接控制也将关闭。当订阅错误代码为 4 (0004h) 到 8 (0008h) 时, 保持订阅控制和连接控制的状态。

*2 ID 是在“有效载荷”对话框中用于识别各要素而分配的编号。有关详情, 请参见第 17-41 页上的““有效载荷”对话框”。

(6) QoS

在 0 ~ 2 范围内设置 QoS。默认值为 0。

(7) 自动 ON

设置在将 Plus CPU 模块连接到代理后, 是否立即自动打开订阅控制。如果选中复选框, 在连接到代理后, 立即自动打开订阅控制。默认值为 ON。

(8) 占用设备地址

显示因设置订阅控制和订阅状态而被占用的设备地址。

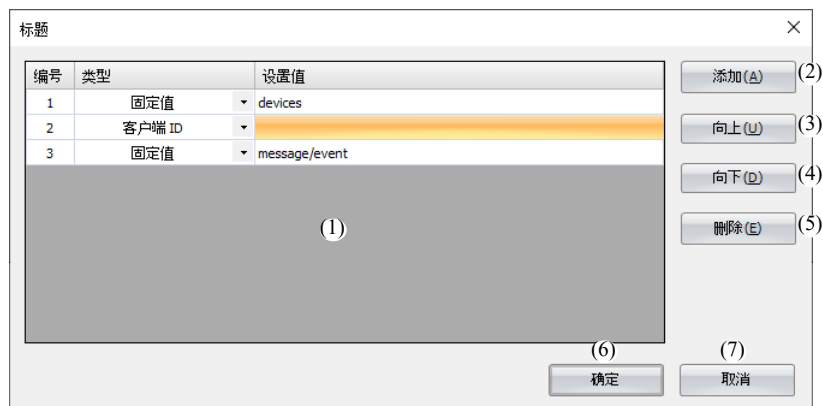
“标题”对话框

在“标题”对话框中设置标题。

如果你从“发布”选项卡打开“标题”对话框，设置要发布的标题。

如果你从“订阅”选项卡打开“标题”对话框，设置要订阅的标题。

将每一行设置为一个字符串，形成一个标题。如果你设置了多行，标题将是一个包含每行数值的字符串，按编号顺序，用“/”连接。



(1) (设置一览)

一览显示设置。

编号 : 显示识别各行的编号 (1 ~ 10)。

类型 : 显示构成标题的字符串的类型。

类型	内容
固定值	设置任意字符串。
客户端 ID/ 设备 ID	“MQTT 设置”选项卡中的“MQTT 基本设置”组中设置的客户端 ID 或设备 ID 将被设置。

设定值 : 显示与“类型”对应的设定值。

(2) “添加”按钮

在 (设置一览) 中选择的行下面增加一行。

(3) “向上”按钮

将 (设置一览) 中选择的行与它上面一行进行交换。

(4) “向下”按钮

将 (设置一览) 中选择的行与它下面一行进行交换。

(5) “删除”按钮

删除在 (设置一览) 中选择的行。

(6) “确定”按钮

保存设置，关闭“标题”对话框。

(7) “取消”按钮

关闭“标题”对话框，不保存设置。

17: MQTT通信

注释:

- 一个标题最多可以设置 10 个字符串（编号为 1 至 10）。
- 根据 Plus CPU 模块的系统软件版本，可以用不同的方式设置相应的标题。

系统软件	如何设置一个相应的主题
低于 2.10 版本	只用一个固定值来设置
版本 2.10 以及 2.10 以上	上述以外

设置一个只有一个固定值的标题的例子：

标题		
编号	类型	设置值
1	固定值	devices/message/event

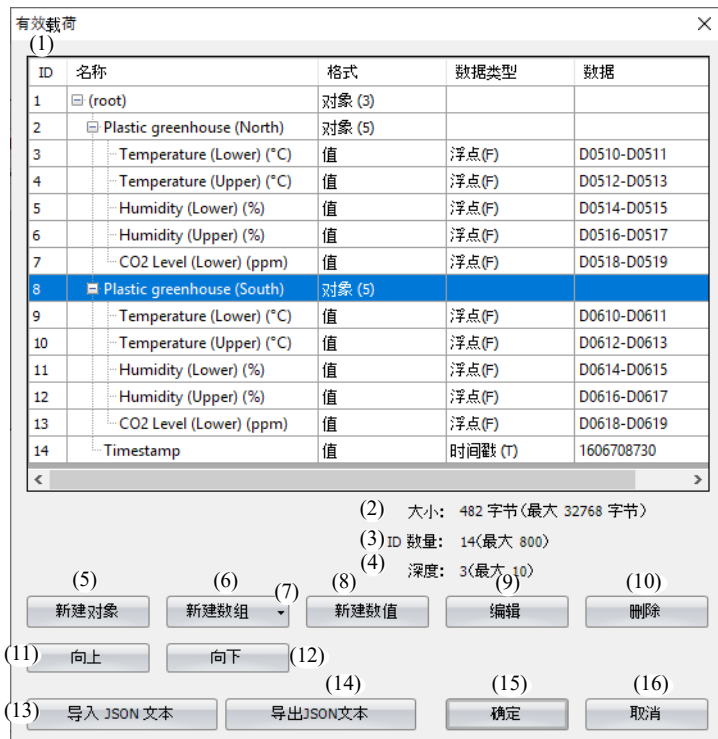
设置一个具有多个固定值的标题的例子：

标题		
编号	类型	设置值
1	固定值	devices
2	固定值	message/event

“有效载荷”对话框

设置 JavaScript Object Notation (JSON) 格式有效载荷内的数据。

当从“发布”选项卡打开“有效载荷”对话框时，设置要发送的 JSON 格式数据的内容。当从“订阅”选项卡打开“有效载荷”对话框时，设置将接收到的 JSON 格式数据的哪个值存储在哪个设备中。



(1) (设置一览)

一览显示设置。

- ID : 显示识别各行的编号。
- 名称 : 显示各 ID 的名称。
- 格式 : 显示各 ID 的格式 (对象、数组或值)。对象和数组的子要素数量也会显示。
- 数据类型 : 显示各值的数据类型。当数据类型为字符串 (S) 时, 也会显示其大小。
- 数据 : 显示固定值或设备地址。如果显示设备地址, 会显示被占用设备地址的范围。

注释: ID1 的“名称”是 (root)。数组子要素的“名称”为从 0 开始的连续编号。

(2) 大小

显示在 (设置一览) 中设置的 JSON 格式数据的最大大小。当从“发布”选项卡打开“有效载荷”对话框时, 最大大小是 32768 字节, 当从“订阅”选项卡打开“有效载荷”对话框时, 最大大小是 8192 字节。

(3) ID 数量

显示 ID 的数量。当从“发布”选项卡打开“有效载荷”对话框时, 最大值是 800, 当从“订阅”选项卡打开“有效载荷”对话框时, 最大值是 200。

(4) 深度

显示在 (设置一览) 中设置的 JSON 格式数据的当前嵌套深度。最大值为 10。

注释: 深度级数 1 称为根。

(5) “新建对象”按钮

在 (设置一览) 中选择的对象或数组的末尾处添加对象。如果 (设置一览) 为空, 则在根处添加对象。

(6) “新建数组”按钮

在 (设置一览) 中选择的对象或数组的末尾处添加数组。如果 (设置一览) 为空, 则在根处添加数组。

(7) “批处理”菜单

添加将连续设备作为子要素的数组。单击此按钮, 将显示“批处理”对话框。有关详情, 请参见第 17-45 页上的““批处理”对话框”。

(8) “新建数值”按钮

在（设置一览）中选择的对象或数组的末尾处添加值。如果（设置一览）为空，则在根处添加值。单击此按钮，将显示“新建数值”对话框。有关详情，请参见第 17-43 页上的““新建数值”或“编辑”对话框”。

(9) “编辑”按钮

单击此按钮，将显示“编辑”对话框。有关详情，请参见第 17-43 页上的““新建数值”或“编辑”对话框”。

(10) “删除”按钮

如下所示根据格式删除在（设置一览）中选择的 ID。

选定 ID 的格式	动作
对象	删除选定对象及其子要素。
数组	删除选定数组及其子要素。
值	删除选定值。

(11) “向上”按钮

交换在（设置一览）中选择的 ID 和位于相同深度的上一个 ID。

(12) “向下”按钮

交换在（设置一览）中选择的 ID 和位于相同深度的下一个 ID。

(13) “导入 JSON 文本”按钮

单击此按钮，将显示“导入 JSON 文本”对话框。有关详情，请参见第 17-46 页上的““导入 JSON 文本”对话框”。

(14) “导出 JSON 文本”按钮

单击此按钮，将显示“导出 JSON 文本”对话框。有关详情，请参见第 17-47 页上的““导出 JSON 文本”对话框”。

(15) “确定”按钮

保存设置，关闭“有效载荷”对话框。

(16) “取消”按钮

关闭“有效载荷”对话框，不保存设置。

“新建数值”或“编辑”对话框

编辑对象或数组。此外还可以添加或编辑值。

(1) 名称

显示或设置值的名称。名称为 UTF-8 编码，最多可设置 255 字节。

(2) 数据类型

设置值的数据类型。仅在添加或编辑值时显示。各数据类型可处理的数据范围如下所示。

数据类型	范围
字 (W)	0 ~ 65535
整数 (I)	-32768 ~ 32767
双字 (D)	0 ~ 4294967295
长整 (L)	-2147483648 ~ 2147483647
浮点 (F)	32 位单精度浮点数可处理的范围*1 但是不包括 $\pm \infty$ (\pm 无限大) 和非数。
布尔型 (B)	true (1) 或 false (0)
字符串 (S)	字符码: UTF-8 最大大小: 1023 字节
时间戳 (T)*2	时区: 使用日历和时钟中的设置*3, UTC+14 ~ UTC-12 格式*4: 当地时间, UTC, UNIX 时间

*1 请参见梯形图编程手册第 3 章“高级指令的数据类型”。

*2 仅当从“发布”选项卡打开“有效载荷”对话框时显示。

*3 使用“功能设置”对话框的“日历和时钟”标签中设置的时区。

*4 有关各格式的说明，请参见下表。

格式	说明
当地时间 (s)	ISO8601 基本格式 / 扩展格式的当地时间
YYYYMMDDThhmmss+/-hhmm YYYY-MM-DDThh:mm:ss+/-hh:mm	
当地时间 (ms)*5	ISO8601 基本格式 / 扩展格式的 UTC
YYYYMMDDThhmmssSSS+/-hhmm YYYY-MM-DDThh:mm:ss.SSS+/-hh:mm	
UTC (s)	ISO8601 基本格式 / 扩展格式的 UTC
YYYYMMDDThhmmssZ YYYY-MM-DDThh:mm:ssZ	
UTC (ms)*5	从 1970 年 1 月 1 日 (UTC+0) 开始、除去 闰秒后经过的时间 (单位: 秒)
YYYYMMDDThhmmssSSSZ YYYY-MM-DDThh:mm:ss.SSSZ	
UNIX 时间 (s)	从 1970 年 1 月 1 日 (UTC+0) 开始、除去 闰秒后经过的时间 (单位: 毫秒)
UNIX 时间 (ms)*5	

*5 如果指定单位为 ms，则 ms 数字总是 000。例如：如果选择了“当地时间 (ms)”中的“YYYY-MM-DDThh:mm:ss.SSS+/-h:mm”格式，那么 2022-05-11T08:30:05.000+09:00 被显示。

注释：时区可使用 D8413（时区偏移量）以 15 分钟为单位进行调整。有关详细信息，请参见第 3-12 页上的“时区的调整 (D8413)”。

(3) 数据

设置值的数据。仅在添加或编辑值时显示。

数据类型	设置方法	说明
字 (W) / 整数 (I) / 双字 (D) / 长整 (L) / 浮点 (F)	固定	设置数值。 可设置的数值范围因数据类型而不同。
	设备	设置数据寄存器。*1
布尔型 (B)	固定	设置 true 或 false。
	设备	设置内部继电器。
字符串 (S)	固定	设置字符串。
	设备	在 1 ~ 1023 的范围内设置数据寄存器及大小 (字节)。*2
时间戳 (T) *3	—	设置时区和格式。

*1 数据类型为 D (双字)、L (长整) 或 F (浮点) 时, 从指定的数据寄存器开始, 使用 2 个字的数据寄存器。

*2 将设置的数据寄存器作为起始地址, 使用设置大小的地址。

*3 仅当从“发布”选项卡打开“有效载荷”对话框时显示。

(4) “确定”按钮

单击“确定”按钮后, 在 (设置一览) 中添加值或保存编辑内容。

(5) “取消”按钮

单击“取消”按钮后, 不会添加值或者保存编辑内容。

“批处理”对话框

添加数组及该数组的子要素（值）。

(1) 名称

显示或设置要添加数组的名称。名称为 UTF-8 编码，最多可设置 255 字节。

(2) 数据类型

设置数组中各子要素（值）的数据类型。

数据类型	范围
字 (W)	0 ~ 65535
整数 (I)	-32768 ~ 32767
双字 (D)	0 ~ 4294967295
长整 (L)	-2147483648 ~ 2147483647
浮点 (F)	32 位单精度浮点数可处理的范围 *1 但是不包括 $\pm \infty$ (\pm 无限大) 和非数。
布尔型 (B)	true (1) 或 false (0)
字符串 (S)	字符码: UTF-8 最大大小: 1023 字节

*1 请参见梯形图编程手册第 3 章“高级指令的数据类型”。

(3) 数据

设置数组子要素（值）的数据。

数据类型	设置方法	说明
字 (W) / 整数 (I) / 双字 (D) / 长整 (L) / 浮点 (F)	设备	设置数据寄存器。 *1
布尔型 (B)		设置内部继电器。
字符串 (S)		在 1 ~ 1023 的范围内设置数据寄存器及大小（字节）。 *2

*1 数据类型为 D（双字）、L（长整）或 F（浮点）时，从指定的数据寄存器开始，使用 2 个字的数据寄存器。

*2 将设置的数据寄存器作为起始地址，使用设置大小的地址。

(4) 长度

设置数组子要素（值）的数量。长度最多可设置为可使用的剩余 ID 数量。默认值为 1。

(5) “确定”按钮

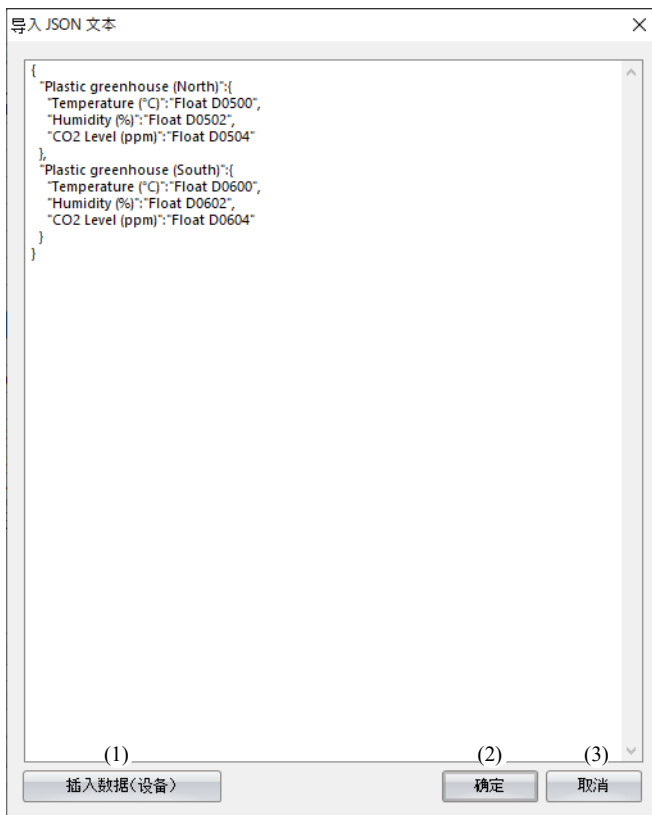
单击“确定”按钮后，添加数组。

(6) “取消”按钮

单击“取消”按钮后，不会添加数组。

“导入 JSON 文本”对话框

设置想要导入的 JSON 格式文本。



(1) 插入数据（设备）

打开“插入数据（设备）”对话框。

(2) “确定”按钮

文本内容会出现在“有效载荷”对话框的（设置一览）中。

注释：

如果在 JSON 文本内各键值对的值部分中指定了数据类型和设备，则转换为设备进行导入。

例) { "key": "Word D100" }

ID2 的数据类型为字（W），设备为 D0100。

{ "key": "String D200 10" }

ID2 的数据类型为字符串（S），设备为 D0200。大小为 10 字节。

{ "Timestamp": "Timestamp UTC+0 UnixTime" }

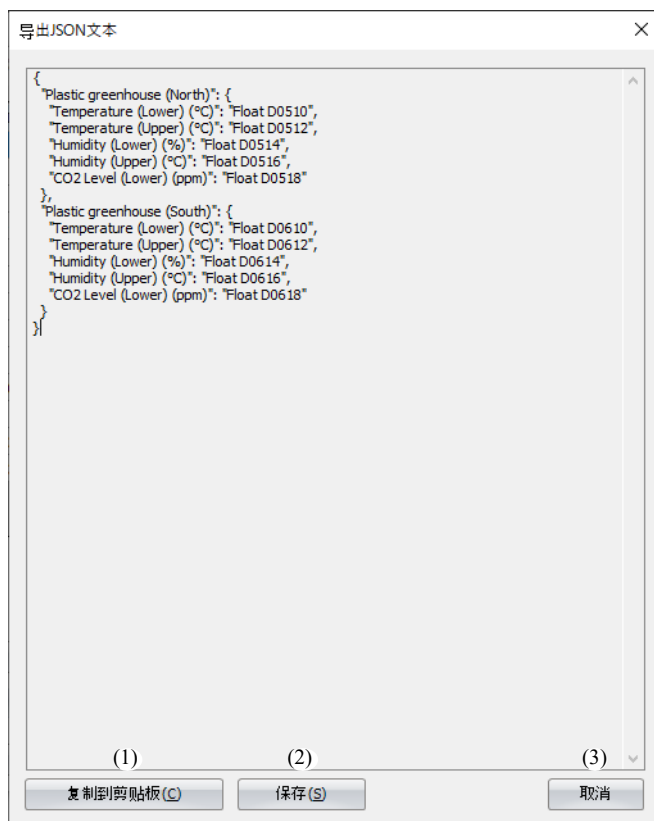
ID2 的数据类型为时间戳（T），时区为 UTC+0，格式为 UNIX 时间。

(3) “取消”按钮

不在“有效载荷”对话框的（设置一览）中显示文本内容，返回到“有效载荷”对话框。

“导出 JSON 文本”对话框

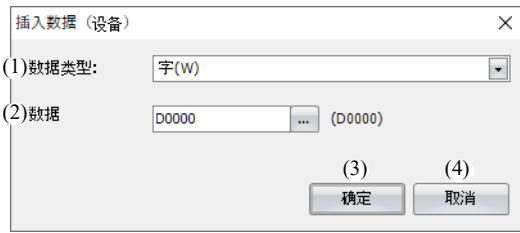
“有效载荷”对话框中设置的有效载荷以 JSON 格式文本输出。



- (1) “复制到剪贴板”按钮
将文本内容复制到剪贴板上。
- (2) “保存”按钮
将文本内容保存为文本格式（.txt）。
- (3) “取消”按钮
关闭“导出 JSON 文本”对话框。

“插入数据（设备）”对话框

创建指定设备和数据类型的字符串。



(1) 数据类型

设置要插入数据（设备）的数据类型。

数据类型	范围
字（W）	0 ~ 65535
整数（I）	-32768 ~ 32767
双字（D）	0 ~ 4294967295
长整（L）	-2147483648 ~ 2147483647
浮点（F）	32 位单精度浮点数可处理的范围*1 但是不包括 $\pm \infty$ （ \pm 无限大）和非数。
布尔型（B）	true（1）或 false（0）
字符串（S）	字符码：UTF-8 最大大小：1023 字节

*1 请参见梯形图编程手册第 3 章“高级指令的数据类型”。

(2) 数据

设置要插入数据（设备）的数据。

数据类型	设置方法	说明
字（W） / 整数（I） / 双字（D） / 长整（L） / 浮点（F）	设备	设置数据寄存器。*1
布尔型（B）		设置内部继电器。
字符串（S）		在 1 ~ 1023 的范围内设置数据寄存器及大小（字节）。*2

*1 数据类型为 D（双字）、L（长整）或 F（浮点）时，从指定的数据寄存器开始，使用 2 个字的数据寄存器。

*2 将设置的数据寄存器作为起始地址，使用设置大小的地址。

(3) “确定”按钮

根据设置的数据类型和数据创建字符串，并将其插入到“导入 JSON 文本”对话框的光标位置。

(4) “取消”按钮

停止插入数据（设备），返回“导入 JSON 文本”对话框。

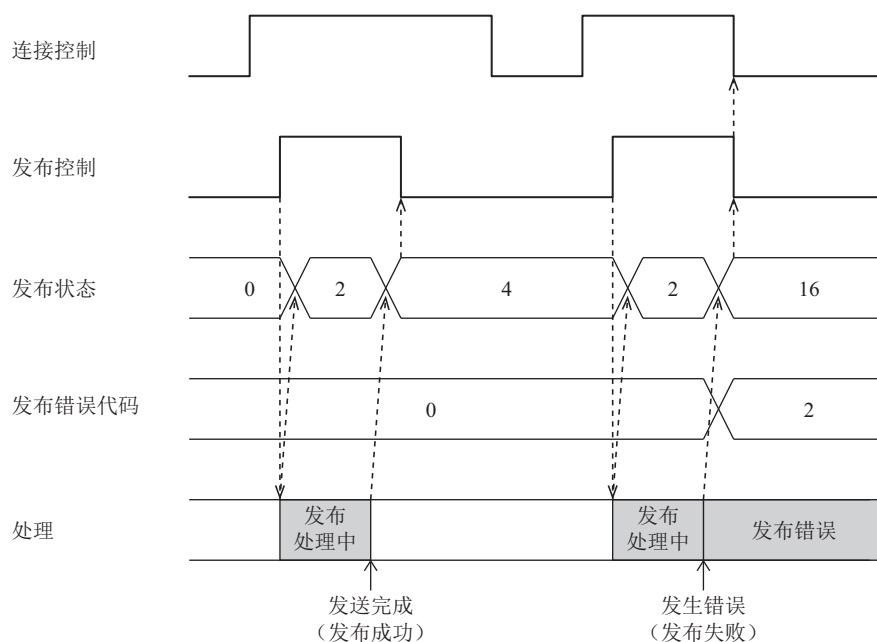
发布标题

Plus CPU 模块根据“操作模式设置”对话框中的“操作模式”设置发布。有关详情，请参见第 17-36 页上的““操作模式设置”对话框”。

当在操作模式中设置为上升沿时

将 Plus CPU 模块连接到代理后，如果将发布控制从关闭设为打开，则发布指定标题，并在发布状态中存储状态。

如果发布成功，存储状态 4 (0004h)，发布控制关闭。如果发布失败，存储状态 16 (0010h)，并在发布错误代码中存储错误代码，发布控制关闭。当错误代码为 7 (0007h) 以外的代码时，连接控制也关闭。当错误代码为 7 (0007h) 时，保持发布控制和连接控制的状态。



注释:

- 请勿在梯形图程序中使发布控制始终处于打开的状态。如果发布成功，存储状态 4 (0004h)，发布控制关闭。如果发布控制在梯形图程序中始终处于打开状态，会重复执行发布。这样可能会产生大量数据包通信费用，敬请注意。
- 在通过梯形图程序关闭发布控制时，如果在不是状态 2 (0002h) 时关闭发布控制，可能不会执行发布。在状态 2 (0002h) 时，请勿更改在“有效载荷”对话框中设置的设备值。请在不是状态 2 (0002h) 的时候进行更改。

17: MQTT 通信

操作示例

操作

- 在草莓栽培塑料大棚内 2 个位置（北侧、南侧）测量温度、湿度以及 CO₂ 浓度。
- 当测得的温度、湿度以及 CO₂ 浓度超出以下范围时，发布标题“plantation/house”。

项目	范围
温度	15.0 ~ 25.0 °C
湿度	60.0 ~ 80.0 %
CO ₂ 浓度	400.0 ppm ~

设置

在“MQTT 设置”对话框的“MQTT 设置”选项卡内进行基本设置后，在各选项卡内设置如下项目。

选项卡	项目	说明
“MQTT 设置”选项卡	连接控制	M0000
	连接状态	D0000
“发布”选项卡	标题	“plantation/house”
	“有效载荷”对话框的设置	参见下图
	操作模式	上升沿
	发布控制	M0100
	发布状态	D1000
	QoS	0
	Retain	禁用

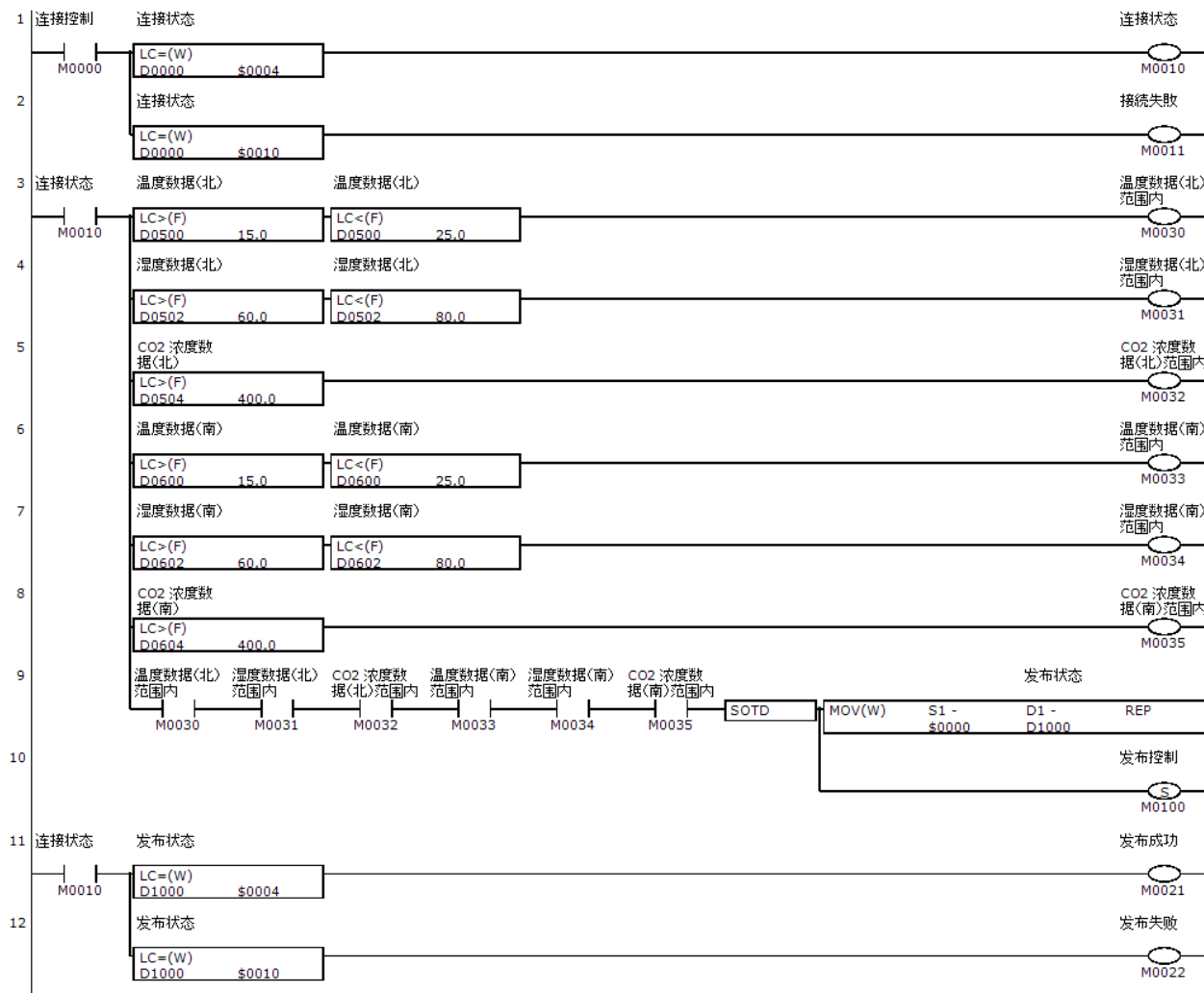
“有效载荷”对话框的设置

有效载荷

ID	名称	格式	数据类型	数据
1	(root)	对象 (3)		
2	Plastic greenhouse (North)	对象 (3)		
3	Temperature (°C)	值	浮点(F)	D0500-D0501
4	Humidity (%)	值	浮点(F)	D0502-D0503
5	CO2 Level (ppm)	值	浮点(F)	D0504-D0505
6	Plastic greenhouse (South)	对象 (3)		
7	Temperature (°C)	值	浮点(F)	D0600-D0601
8	Humidity (%)	值	浮点(F)	D0602-D0603
9	CO2 Level (ppm)	值	浮点(F)	D0604-D0605
10	Timestamp	值	时间戳(T)	1606742014

大小: 276 字节(最大 32768 字节)
ID 数量: 10(最大 800)
深度: 3(最大 10)

梯形图程序



设备地址	注释
M0000	连接控制
M0010	连接状态
M0011	连接失败
M0021	发布成功
M0022	发布失败
M0030	温度数据（北）范围内
M0031	湿度数据（北）范围内
M0032	CO2 浓度数据（北）范围内
M0033	温度数据（南）范围内
M0034	湿度数据（南）范围内
M0035	CO2 浓度数据（南）范围内
M0100	发布控制

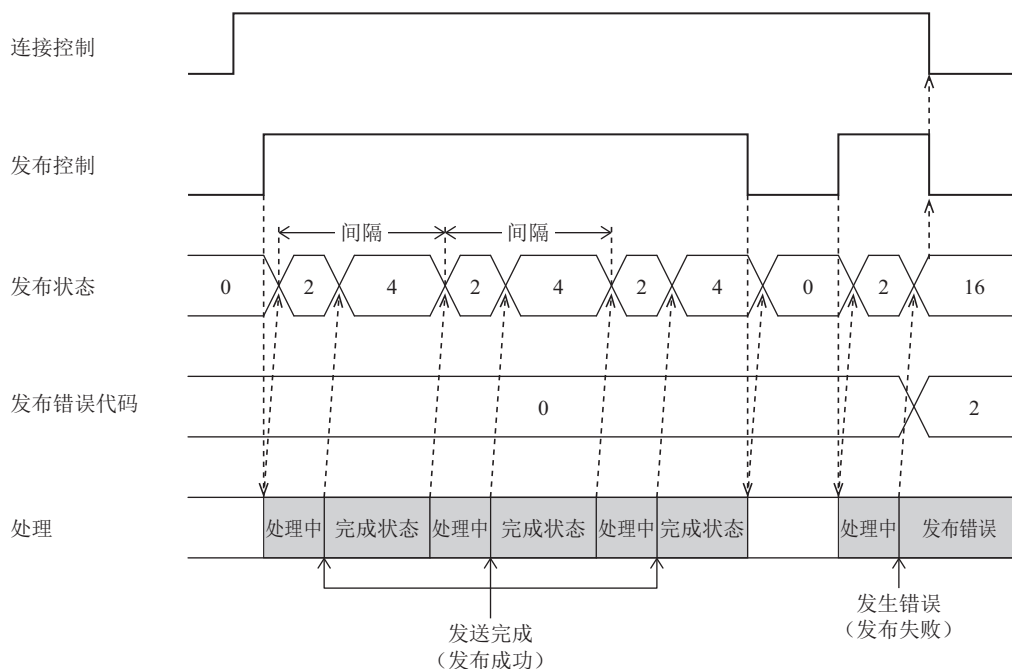
设备地址	注释
D0000	连接状态
D0500	温度数据（北）
D0502	湿度数据（北）
D0504	CO2 浓度数据（北）
D0600	温度数据（南）
D0602	湿度数据（南）
D0604	CO2 浓度数据（南）
D1000	发布状态

梯形图程序行	说明												
1	打开 M0000 后，开始连接到代理。M0010 打开时，处于连接到代理的状态。												
2	如果无法连接到代理，则 M0011 打开。												
3 ~ 8	当 M0010 打开时（处于连接到代理的状态时），如果在塑料大棚内的 2 个位置（北侧、南侧）测得的温度、湿度以及 CO ₂ 浓度在设置范围内，则 M0030 ~ M0035 打开。												
9、10	当 M0010 打开时（处于连接到代理的状态时），如果温度、湿度以及 CO ₂ 浓度中任意一个超出设置范围，则相应的内部继电器（M0030 ~ M0035）从打开变为关闭。此时，将在 D1000 中存储 0（0000h），清除发布状态。此外，M0100 打开，并发布标题。（发布完成后，M0100 会自动关闭。）												
11	<p>发布成功时，M0021 打开。</p> <p>如果是下表中的情况，则（D0500、D0501）=18.5、（D0502、D0503）=55.5、（D0504、D0505）=410.1、（D0600、D0601）=26.1、（D0602、D0603）=64.5、（D0604、D0605）=420.2。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>项目</th> <th>塑料大棚内（北侧）</th> <th>塑料大棚内（南侧）</th> </tr> </thead> <tbody> <tr> <td>温度</td> <td>18.5 °C</td> <td>26.1 °C</td> </tr> <tr> <td>湿度</td> <td>55.5 %</td> <td>64.5 %</td> </tr> <tr> <td>CO₂ 浓度</td> <td>410.1 ppm</td> <td>420.2 ppm</td> </tr> </tbody> </table> <p>此外，当内置时钟的当前时间数据（D8008 ~ D8014）为 2020 年 12 月 9 日 9 时 5 分 46 秒，而且 D8413（时区偏移量）的值为 0 时，发送的 JSON 格式数据如下所示。</p> <pre> { "Plastic greenhouse (North)": { "Temperature (°C)": 18.5, "Humidity (%)": 55.5, "CO2 Level (ppm)": 410.1 }, "Plastic greenhouse (South)": { "Temperature (°C)": 26.1, "Humidity (%)": 64.5, "CO2 Level (ppm)": 420.2 }, "Timestamp": 1607504746 } </pre> <p>此时在 D1002（发送数据字节数）中存储 224。</p>	项目	塑料大棚内（北侧）	塑料大棚内（南侧）	温度	18.5 °C	26.1 °C	湿度	55.5 %	64.5 %	CO ₂ 浓度	410.1 ppm	420.2 ppm
项目	塑料大棚内（北侧）	塑料大棚内（南侧）											
温度	18.5 °C	26.1 °C											
湿度	55.5 %	64.5 %											
CO ₂ 浓度	410.1 ppm	420.2 ppm											
12	发布失败时，M0022 打开。												

当在操作模式中设置为固定期限时

将 Plus CPU 模块连接到代理后，当发布控制打开时，以在间隔 (2) 中设置的周期发布指定标题，并在发布状态中存储状态。关闭发布控制后，发布结束。此时会存储状态 0 (0000h)。

如果发布失败，存储状态 16 (0010h)，并在发布错误代码中存储错误代码，发布控制关闭。当错误代码为 7 (0007h) 以外的代码时，连接控制也关闭。当错误代码为 7 (0007h) 时，保持发布控制和连接控制的状态。



注释: 在状态 2 (0002h) 时，请勿更改在“有效载荷”对话框中设置的设备值。请在不是状态 2 (0002h) 的时候进行更改。

操作示例

操作

- 在草莓栽培塑料大棚内 2 个位置，每隔 30 分钟测量一次温度、湿度以及 CO₂ 浓度，并发布标题“plantation/house”。
- 远程确认塑料大棚内的状态。

设置

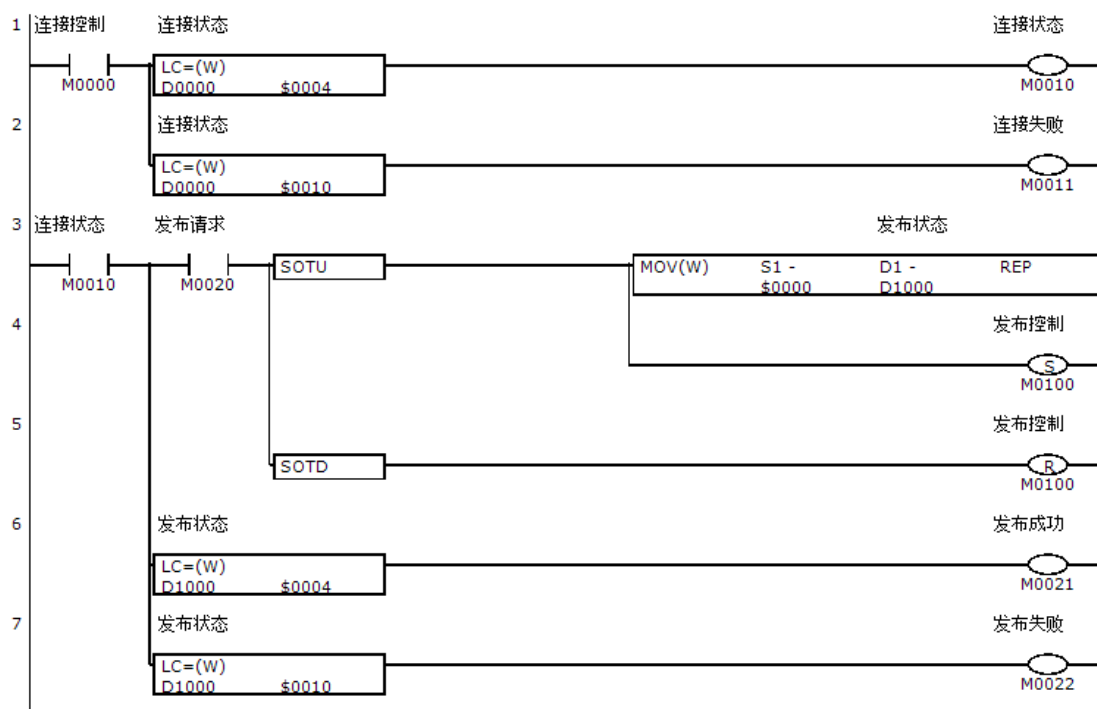
在“MQTT 设置”对话框的“MQTT 设置”选项卡内进行基本设置后，在各选项卡内设置如下项目。

选项卡	项目	说明	
“MQTT 设置”选项卡	连接控制	M0000	
	连接状态	D0000	
“发布”选项卡	标题	“plantation/house”	
	“有效载荷”对话框的设置	参见下图	
	操作模式	操作模式：固定期限	
		间隔：1800 秒	
	发布控制	M0100	
	发布状态	D1000	
	QoS	1	
Retain	禁用		

“有效载荷”对话框的设置



梯形图程序



设备地址	注释
M0000	连接控制
M0010	连接状态
M0011	连接失败
M0020	发布请求
M0021	发布成功
M0022	发布失败
M0100	发布控制

设备地址	注释
D0000	连接状态
D0500	温度数据 (北)
D0502	湿度数据 (北)
D0504	CO2 浓度数据 (北)
D0600	温度数据 (南)
D0602	湿度数据 (北)
D0604	CO2 浓度数据 (北)
D1000	发布状态

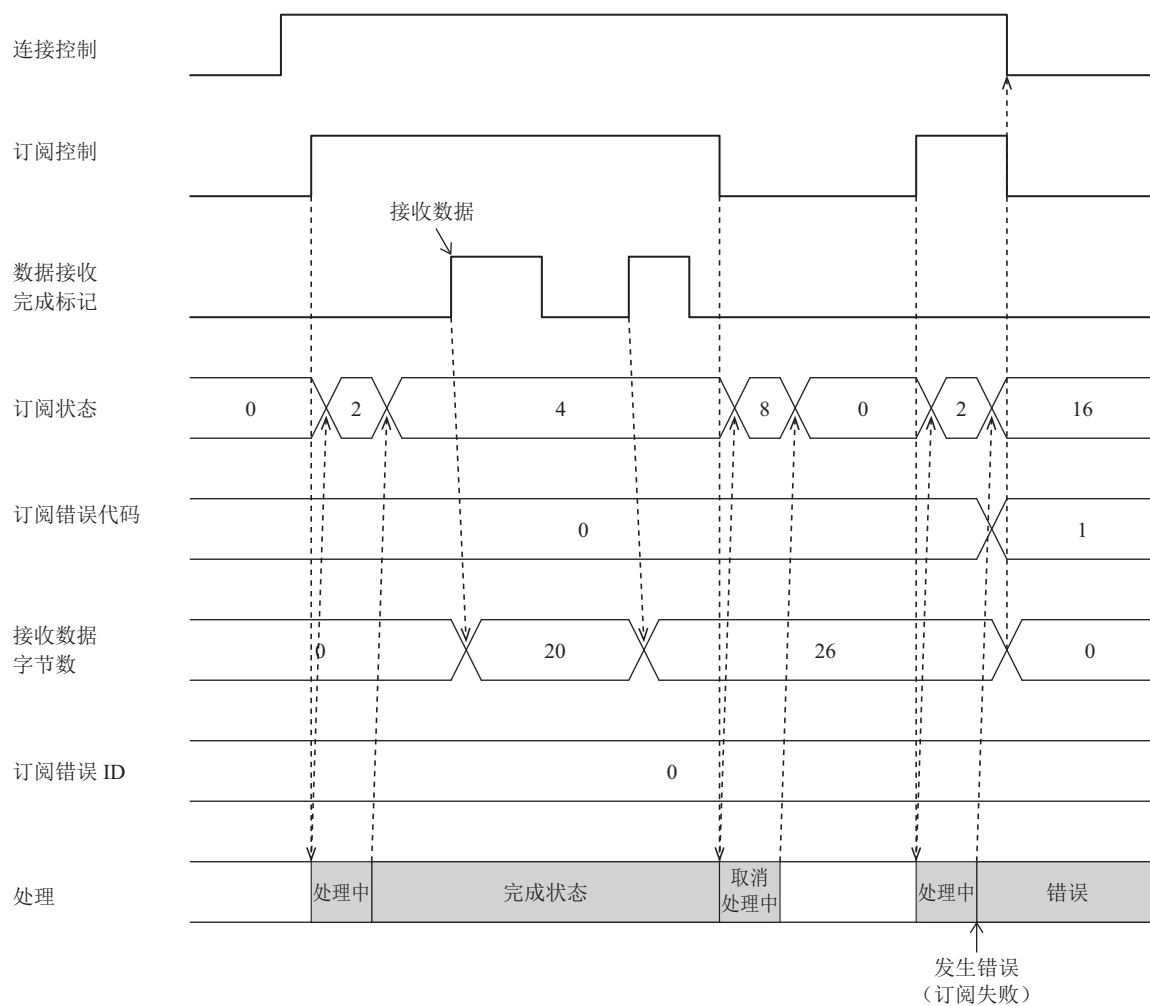
梯形图程序行	说明												
1	打开 M0000 后，开始连接到代理。M0010 打开时，处于连接到代理的状态。												
2	如果无法连接到代理，则 M0011 打开。												
3	当 M0010 打开时（处于连接到代理的状态时），如果将 M0020 从关闭设为打开，将在 D1000 中存储 0（0000h），并清除发布状态。												
4	M0100 打开，每隔 1800 秒（30 分钟）发布标题。												
5	关闭 M0020 后，发布结束。												
5	<p>发布成功时，M0021 打开。 如果是下表中的情况，则（D0500、D0501）=20.5、（D0502、D0503）=60.5、（D0504、D0505）=410.1、（D0600、D0601）=26.1、（D0602、D0603）=64.5、（D0604、D0605）=420.2。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>项目</th> <th>塑料大棚内（北侧）</th> <th>塑料大棚内（南侧）</th> </tr> </thead> <tbody> <tr> <td>温度</td> <td>20.5 °C</td> <td>26.1 °C</td> </tr> <tr> <td>湿度</td> <td>60.5 %</td> <td>64.5 %</td> </tr> <tr> <td>CO₂ 浓度</td> <td>410.1 ppm</td> <td>420.2 ppm</td> </tr> </tbody> </table> <p>此外，当内置时钟的当前时间数据（D8008 ~ D8014）为 2020 年 11 月 9 日 10 时 42 分 8 秒，而且 D8413（时区偏移量）的值为 0 时，发送的 JSON 格式数据如下所示。</p> <pre>{ "Plastic greenhouse (North)": { "Temperature (°C)": 18.5, "Humidity (%)": 55.5, "CO2 Level (ppm)": 410.1 }, "Plastic greenhouse (South)": { "Temperature (°C)": 26.1, "Humidity (%)": 64.5, "CO2 Level (ppm)": 420.2 }, "Timestamp": "2020-11-09T10:42:08Z" }</pre> <p>此时在 D1002（发送数据字节数）中存储 236。</p>	项目	塑料大棚内（北侧）	塑料大棚内（南侧）	温度	20.5 °C	26.1 °C	湿度	60.5 %	64.5 %	CO ₂ 浓度	410.1 ppm	420.2 ppm
项目	塑料大棚内（北侧）	塑料大棚内（南侧）											
温度	20.5 °C	26.1 °C											
湿度	60.5 %	64.5 %											
CO ₂ 浓度	410.1 ppm	420.2 ppm											
12	发布失败时，M0022 打开。												

订阅标题

订阅与取消订阅

将 Plus CPU 模块连接到代理后，如果将订阅控制从关闭设为打开，则订阅指定标题，并在订阅状态中存储状态。

订阅成功后，存储状态 4（0004h）。在此状态下，当代理发布指定标题时，数据接收完成标记变为 ON，接收到数据的长度将以字节为单位存储在接收数据字节数中。如需了解如何处理接收到的数据，请参见第 17-60 页上的“将接收到的 JSON 格式数据的内容存储到设备中”。



操作示例

操作

- 管理草莓栽培塑料大棚内的温度、湿度以及 CO₂ 浓度。
- 管理员将塑料大棚内温度、湿度的上下限值或 CO₂ 浓度的下限值远程发布标题 “plantation/house”。
- Plus CPU 模块订阅标题 “plantation/house”，并按照管理员发布的设置执行操作。

设置

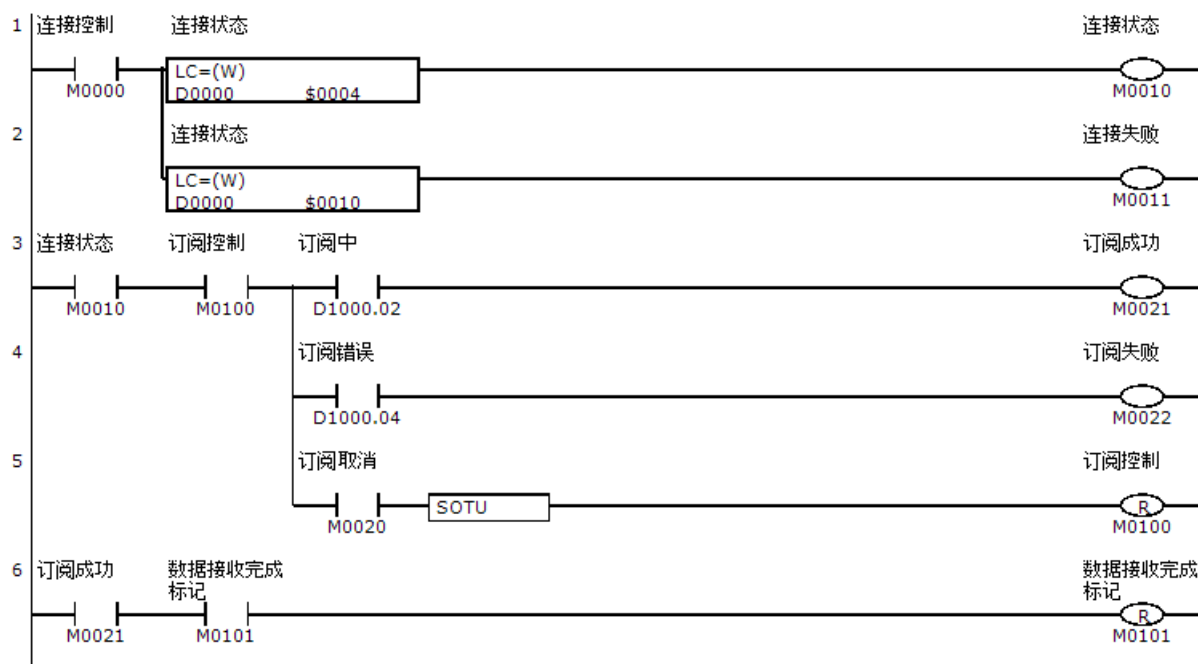
在 “MQTT 设置” 对话框的 “MQTT 设置” 选项卡内进行基本设置后，在各选项卡内设置如下项目。

选项卡	项目	说明
“MQTT 设置” 选项卡	连接控制	M0000
	连接状态	D0000
“订阅” 选项卡	标题	“plantation/house”
	“有效载荷” 对话框的设置	参见下图
	订阅控制	M0100
	订阅状态	D1000
	QoS	1
	自动 ON	启用

“有效载荷” 对话框的设置



梯形图程序



设备地址	注释
M0000	连接控制
M0010	连接状态
M0011	连接失败
M0020	订阅取消
M0021	订阅成功
M0022	订阅失败
M0100	订阅控制
M0101	数据接收完成标记

设备地址	注释
D0000	连接状态
D0510	温度数据（北）下限值
D0512	温度数据（北）上限值
D0514	湿度数据（北）下限值
D0516	湿度数据（北）上限值
D0518	CO2 浓度数据（北）下限值
D0610	温度数据（南）下限值
D0612	温度数据（南）上限值
D0614	湿度数据（南）下限值
D0616	湿度数据（南）上限值
D0618	CO2 浓度数据（南）下限值
D1000	订阅状态

梯形图程序行	说明
1	打开 M0000 后，开始连接到代理。M0010 打开时，处于连接到代理的状态。
2	如果无法连接到代理，则 M0011 打开。
3	M0010 打开后（连接到代理后），M0100 会自动打开，并订阅标题。订阅成功时，M0021 打开。
4	订阅失败时，M0022 打开。
5	打开 M0020 后，取消订阅。
6	接收数据后，关闭 M0101。

将接收到的 JSON 格式数据的内容存储到设备中

从头开始依次查找接收到的 JSON 格式数据和在“有效载荷”对话框中设置的 ID。

如果在接收到的 JSON 格式数据中，存在与“有效载荷”对话框中所设置 ID 的深度级数、名称以及格式一致的要素，而且该要素的值在其 ID 的数据类型范围内，则将该值存储到设备中。

注释:

- 如果在接收到的 JSON 格式数据中，没有与“有效载荷”对话框中所设置 ID 的深度级数、名称以及格式一致的要素，则订阅错误 ID 中将存储相应 ID。
- 如果在接收到的 JSON 格式数据中，存在与“有效载荷”对话框中所设置 ID 的深度级数、名称以及格式一致的要素，但是该要素的值超出其 ID 的数据类型范围，则在订阅错误代码中存储错误代码 7 (0007h)，并在订阅错误 ID 中存储发生错误的 ID。

例

- Plus CPU 模块控制着工厂的 2 条生产线。
- 接收到设置值后，即会根据该设置值进行处理。虽然有多个设置值，但是实际上只接收有更改的设置值。

下面将对在“有效载荷”对话框中执行下图所示设置的情况进行介绍。

“有效载荷”对话框

ID	名称	格式	数据类型	数据
(1) 1	(root)	对象 (2)		
(2) 2	production line1	对象 (1)		
3	settings	对象 (3)		
4	production count	值	字(W)	D0100
5	conveyor1 speed	值	字(W)	D0101
6	conveyor2 speed	值	字(W)	D0102
(3) 7	production line2	对象 (1)		
(4) 8	settings	对象 (3)		
(5) 9	production count	值	字(W)	D0200
(6) 10	conveyor1 speed	值	字(W)	D0201
(7) 11	conveyor2 speed	值	字(W)	D0202

接收到的 JSON 格式数据

梯形行编号	说明
1	{
2	"production line2" :{
3	"settings":{
4	"conveyor1 speed":50
5	}
6	}
7	}

处理流程

- (1) 接收到的 JSON 格式数据从第 1 行到第 7 行被大括号“{ }”所包围。它与“有效载荷”对话框的 ID1 设置一致，因此会继续查找 ID2。
- (2) 接收到的 JSON 格式数据中没有 ID2，因此跳过查找 ID2 及其子要素，并在订阅错误 ID 中存储 2 (0002h)。
- (3) 接收到的 JSON 格式数据从第 2 行到第 6 行被大括号“{ }”所包围，设置名称为“production line2”。它与 ID7 设置一致，因此会继续查找 ID8。
- (4) 接收到的 JSON 格式数据从第 3 行到第 5 行被大括号“{ }”所包围，设置名称为“settings”。它与 ID8 设置一致，因此会继续查找 ID9。

- (5) 接收到的 JSON 格式数据中没有 ID9，因此会跳过 ID9，继续查找 ID10。
- (6) 接收到的 JSON 格式数据第 4 行的名称“conveyor1 speed”与 ID10 的名称一致，而且该要素的值（50）在字（W）的范围内，因此在 D0201 中存储 50。
- (7) 接收到的 JSON 格式数据中没有 ID11，因此会跳过 ID11，在订阅错误代码中存储 0（0000h）。

18: MC 协议通信

简介

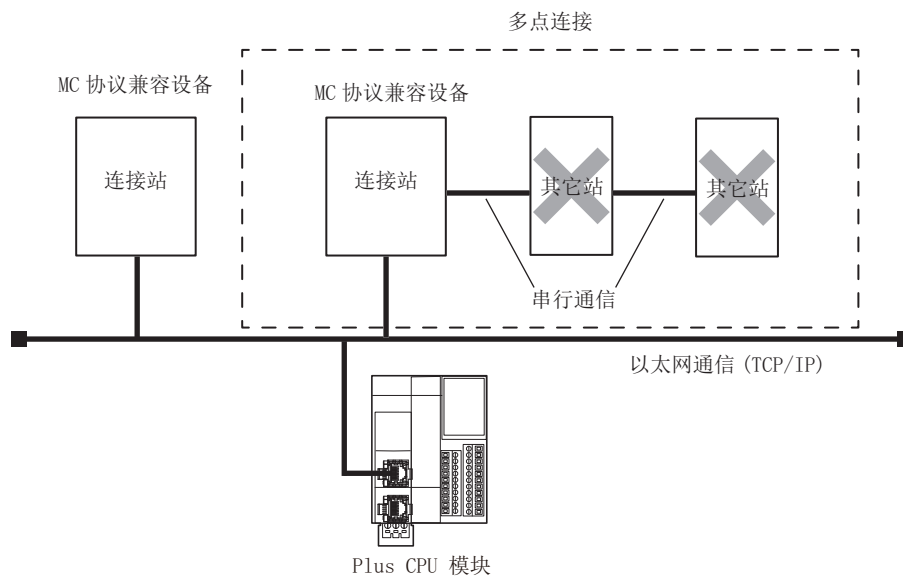
本章介绍 FC6A 型的 MC 协议通信功能。

概述

MC 协议是 MELSEC 通信协议的缩写，MELSEC 通信协议是用于与如三菱电机公司制造的 PLC 之类的 MC 协议兼容设备（以下称为 MC 协议兼容设备）进行通信的协议。作为 MC 协议通信的客户端，Plus CPU 模块可以读取和写入 MC 协议兼容设备的设备值。有关 MC 协议的详细信息，请参见《MELSEC 通信协议参考手册》。

访问范围

MC 协议通信的网络配置如下所示。Plus CPU 模块可以与支持 QnA 兼容 3E 帧的 MC 协议兼容设备（连接站）进行通信。Plus CPU 模块无法与连接到连接站（多点连接）的其他 MC 协议兼容设备（其它站）进行通信。



基本规格

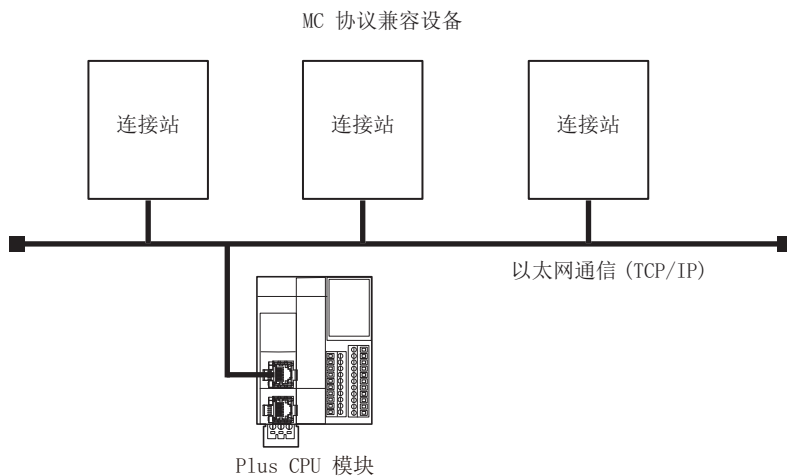
基本规格如下所示。

模块	Plus CPU 模块
通信端口	以太网端口 1 和 2
支持框架	QnA 兼容 3E 帧
支持的指令和副指令	字单位批量读取（指令：0401，副指令：0000） 位单位批量读取（指令：0401，副指令：0001） 字单位批量写入（指令：1401，副指令：0000） 位单位批量写入（指令：1401，副指令：0001）
支持的代码	二进制代码
链接数	最大 16 个*1

*1 1 个链接可以链接 255 个服务器

通过以太网通讯进行的 MC 协议通讯

Plus CPU 模块在以太网端口 1 和以太网端口 2 上支持 MC 协议通信客户端，并且最多可以将 16 个连接分配给 MC 协议通信客户端。每个连接都可以与多个（1 到 255 个）MC 协议设备通信。

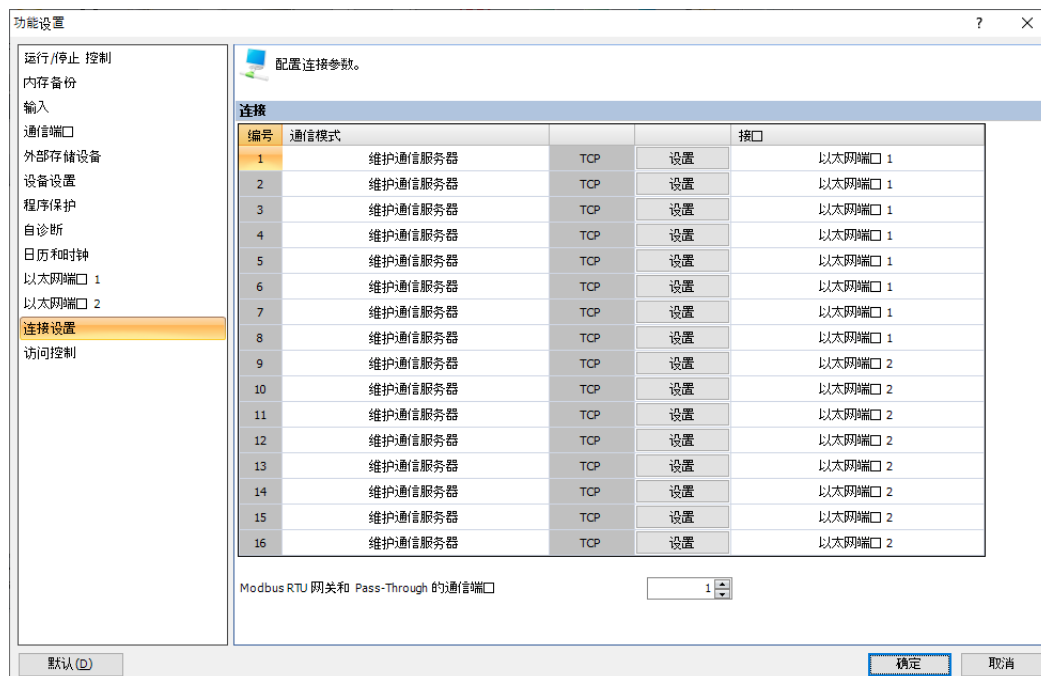


注释： Plus CPU 模块可以与支持 QnA 兼容 3E 帧（二进制代码）的 MC 协议兼容设备进行通信。

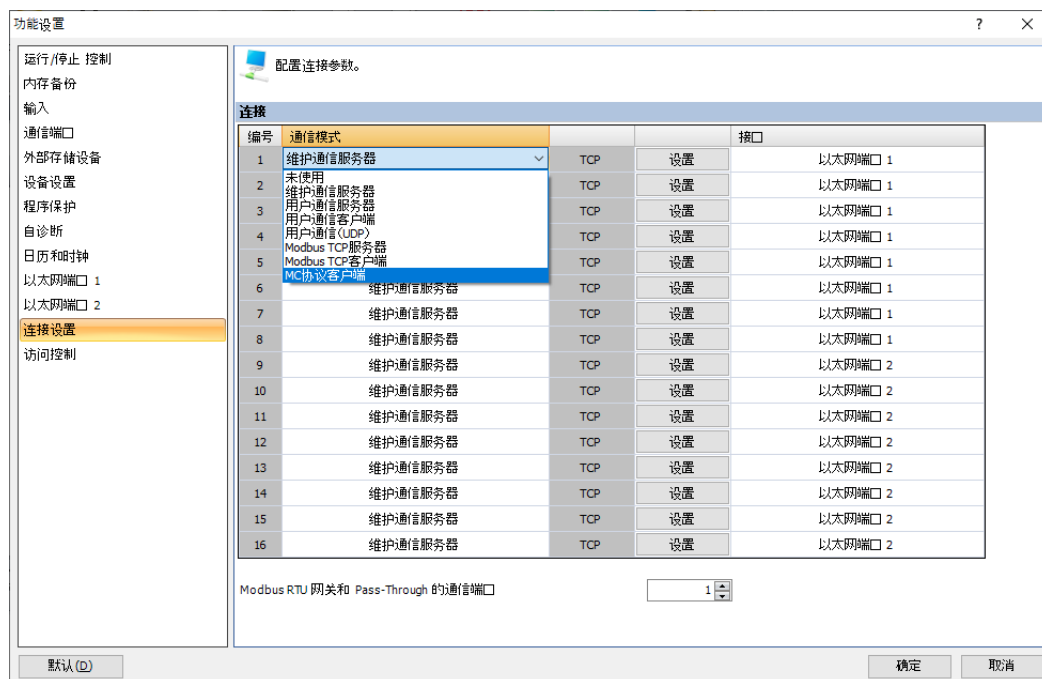
设置 MC 协议通信

使用 WindLDR 设置 MC 协议通信

- 在 WindLDR 菜单栏中选择“设置”>“功能设置”>“通信端口”。
此时出现通信端口的“功能设置”对话框。



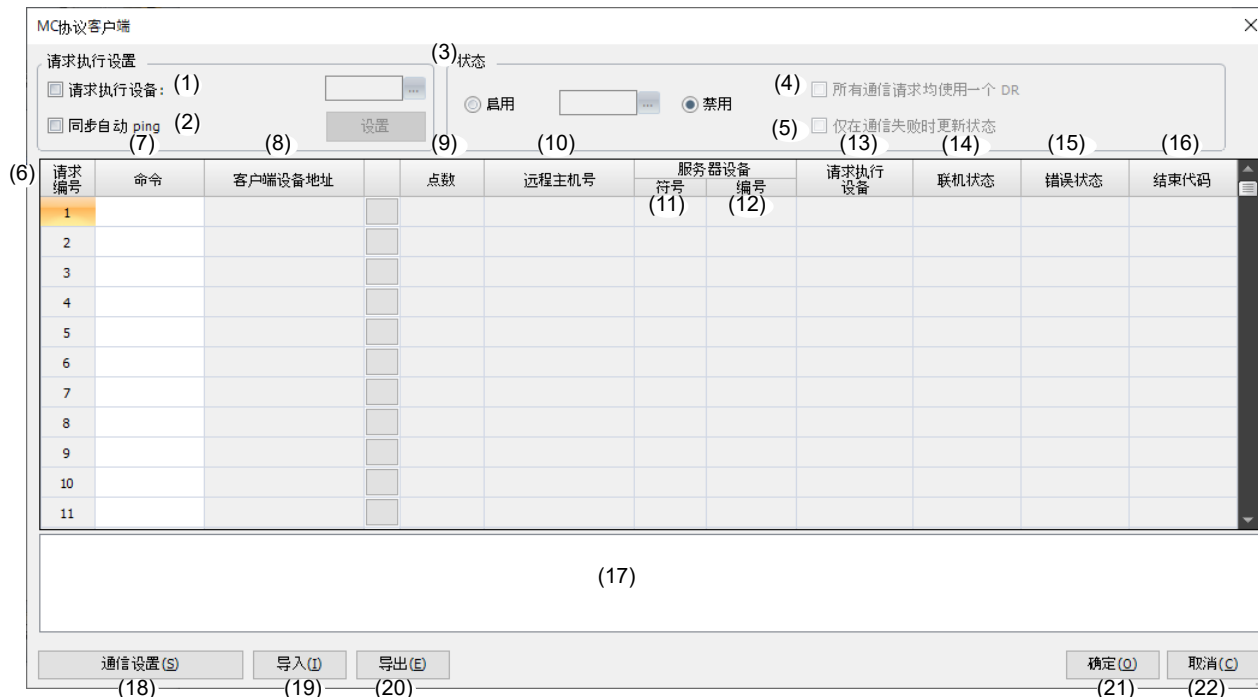
2. 在“通信模式”中单击“MC 协议客户端”。



将显示“MC 协议客户端”对话框。

“MC 协议客户端”对话框

在“MC 协议客户端”对话框中设置 MC 协议通信。



注释：“MC 协议客户端”对话框中央部分的表称为请求列表。请求列表的每一行被称为请求。

(1) 请求执行设备

Plus CPU 模块的请求的执行由通信执行设备的 ON/OFF 控制。

当通信执行设备从关闭变为打开时，Plus CPU 模块在第 18-10 页上的“(2) 发送等待时间”中设置的时间过后发送请求。发送请求并接收到响应后，相应的通信执行设备将自动关闭。

如果无法接收到响应，请在第 18-10 页上的“(1) 接收超时”中设置的时间过去后重试。重试将执行两次。之后，相应的通信执行设备将自动关闭。这一系列处理称为请求执行。另外，当接收到正常响应时，该请求被认为是成功执行。当收到异常响应或重试后未收到响应时，该请求被认为是执行失败。

当通讯执行设备关闭时，Plus CPU 模块不发送请求。

通信执行设备可以配置有内部继电器或数据寄存器。从设置的设备开始，按请求编号的顺序将设备分配给在请求列表中注册的请求。当在通信执行设备中设置了数据寄存器时，以请求编号的顺序分配通信执行设备，从数据寄存器的最低位开始。

请求执行设备	说明
不使用 (禁用状态)	登录在请求列表中的请求会按照请求编号的顺序重复执行。
使用 (启用状态)	对于在请求列表中注册的请求，以所设置的设备为开头，按照在请求列表中登记的请求号码顺序分配请求执行设备。将要执行的请求的请求执行设备打开时，开始执行请求。

注释：到请求执行设备自动关闭为止，请不要变更读取以及写入的设备值。可能会出现变更前的设备值无法读取、写入的情况。

同时将多个通信执行设备从关闭切换为打开时，将在之前执行的请求编号之后最近的请求编号开始对请求逐一处理。执行完成（成功或失败）之后处理下一个请求。

(2) 同步自动 Ping

是否执行请求根据自动 Ping 的结果控制。本功能会在远程主机由于故障等原因没有响应时，以最小的延迟与另一台远程主机执行 MC 协议通信。

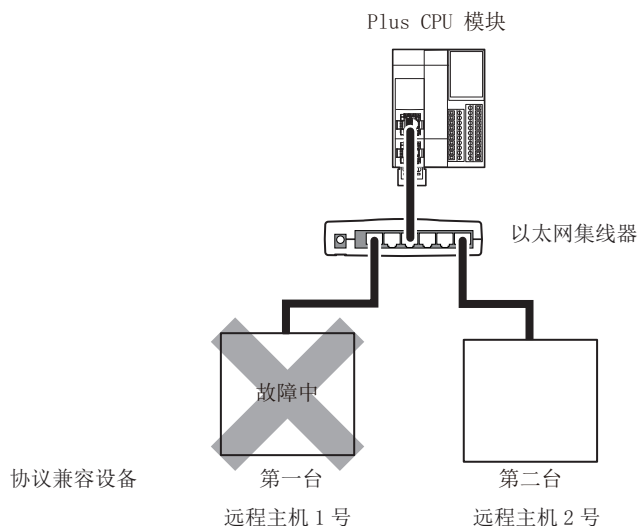
同步自动 Ping	说明
不同步自动 Ping (禁用状态)	登录在请求列表中的请求会按照请求编号的顺序重复执行。
同步自动 Ping (启用状态)	<ul style="list-style-type: none"> 作为自动 Ping 发送对象的远程主机 联机状态（储存已发送 Ping 的结果的内部继电器）为打开时，执行相应的请求。 联机状态为关闭时，不执行相应的请求。 <ul style="list-style-type: none"> 不是自动 Ping 发送对象的远程主机 无关于联机状态的内部继电器的状态，都执行相应请求。

在执行 MC 协议通信时，必须将通信对象的 MC 协议兼容设备注册为远程主机列表中的远程主机。如果选中“同步自动 Ping”的复选框，则将显示“设置”按钮。点击“设置”按钮，会显示“远程主机列表”对话框，请进行远程主机的注册。有关详细信息，请参见第 3-21 页上的“远程主机列表”。

本节介绍不同步自动 Ping 时和与同步自动 Ping 时的请求处理。

系统配置示例**设置内容**

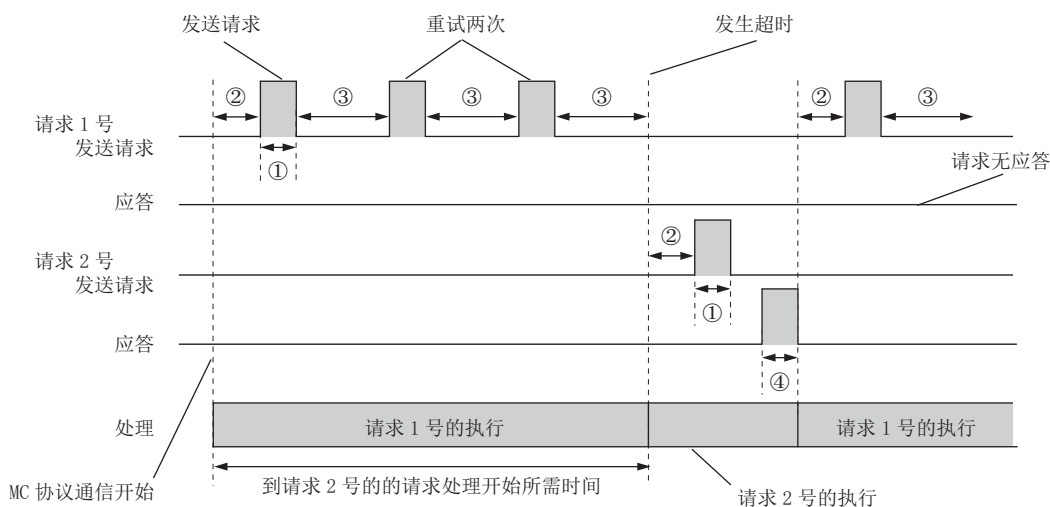
请求执行设备：未使用

**■不同步自动 Ping（未选中“同步自动 Ping”复选框时）**

按照请求编号的顺序执行请求。

请求编号	远程主机编号	请求的执行
1	1	执行
2	2	执行

请求处理时序图



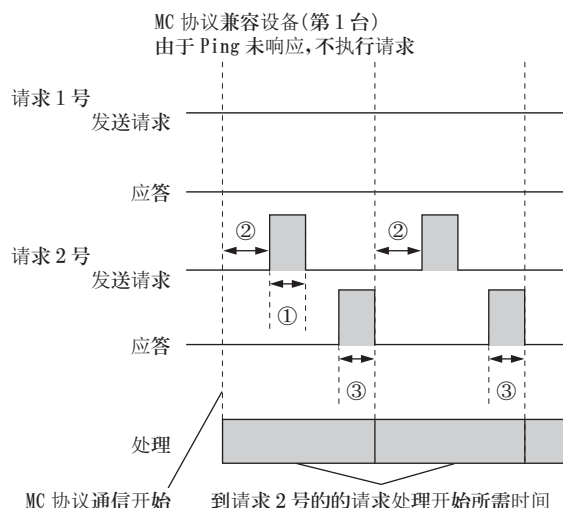
- ① 发送请求
- ② 从请求执行开始到发送为止的发送等待时间
- ③ 在“接收超时”中设置的时间过去后会重试。
- ④ 应答

■同步自动 Ping（选中“同步自动 Ping”复选框时）

仅向联机的远程主机发出请求。

请求编号	远程主机编号	联机状态*1	请求的执行
1	1	OFF	执行
2	2	ON	执行

*1 联机状态是一个用于存储发送的 Ping 结果的内部继电器。有关详情，请参见第 18-10 页上的“(14) 联机状态”
请求处理时序图



- ① 发送请求
- ② 从请求执行开始到发送为止的发送等待时间
- ③ 应答

注释：本功能是在某一远程主机由于故障等原因没有响应的状况下，要与其他远程主机进行最小延迟 MC 协议通信也依旧有效。无响应的远程主机完成配置的请求所花费的时间是在“通信设置”对话框的“接收超时”中设置的时间的三倍。

有关自动 Ping 的详细信息，请参见第 3-23 页上的“自动 Ping 功能”。

(3) 状态

执行请求时在设备上设置是否存储错误状态和结束代码。如果要在设备上存储错误状态和结束代码，请选择“使用”并设置数据寄存器。每个请求使用 2 个字节的数据寄存器。

注释：创建最大请求数（255 个）后，将如下表所示分配数据寄存器。

设备地址	内容		读 / 写
开头编号 + 0	请求编号的状态 1	错误状态	读
开头编号 + 1		结束代码	读
开头编号 + 2	请求编号的状态 2	错误状态	读
开头编号 + 3		结束代码	读
...
开头编号 + 508	请求编号的状态 255	错误状态	读
开头编号 + 509		结束代码	读

错误状态

远程主机号存储在高位字节中，错误代码存储在低位字节中。

错误状态 (十六进制)	
高位字节	低位字节
远程主机编号	错误代码

错误代码的详细信息如下

错误代码 (十六进制)	内容	详情
0x04	异常退出	从 MC 协议兼容设备接收数据的结束代码不是 0x0000 (正常关闭)
0x16	超时错误	MC 协议兼容设备无应答

结束代码

存储 MC 协议通信的结束代码。有关结束代码的详细信息，请参见通信对象设备的 MC 协议兼容设备的手册。

注释：如果在通信过程中发生错误，例如断开以太网连接或切断 MC 协议兼容设备的电源时，将发生超时错误，错误代码将为 0x16。此时，将保存在发生错误之前从 MC 协议兼容设备读取的数据在第 18-8 页上的“(8) 客户端设备”中设置的设备中。如果要清除此设备上的数据，请写入 0。

(4) 所有通信请求均使用一个 DR

当“状态”为“使用”时，设置如何分配用于存储错误状态和结束代码的数据寄存器。

详情如下所示。

所有通信请求均使用一个 DR	说明
不使用 (未选中复选框)	存储每个请求的错误状态和结束代码。 首先使用在“状态”中设置的数据寄存器，每个请求使用 2 个字节（最多 510 个字节）的数据寄存器。
使用 (选中复选框)	存储最后一个请求的错误状态和结束代码。 从“状态”中设置的数据寄存器开始，使用 2 字节数据寄存器。

(5) 仅在通信失败时更新状态

设置更新错误状态和结束代码的时机。

如果选中“仅在通信失败时更新错误状态”复选框，则仅在请求执行失败时才更新请求的状态（错误状态和结束代码）。如果未选中此复选框，它将更新所有请求的状态。

18: MC 协议通信

(6) 请求编号

每个请求的编号（1～255 个）。

注释：使用请求执行设备以及状态时，将已设置的设备为开头，按照请求编号的顺序自动分配设备。如果删除请求列表中间请求或更改请求的顺序，则会更新请求执行设备和错误状态分配，这可能会导致与用户程序中使用的设备不一致。如果更改请求列表，请仔细检查用户程序。

(7) 指令

Plus CPU 模块从以下内容中选择向 MC 协议兼容设备发送数据的指令。

指令	说明
04010000 字单位读取	将在“符号”以及“编号”所指定的设备中指定的“点数”的点数份的字设备值储存在“客户端设备”中“点数”所指定的点数份的字设备中。
04010001 位单位读取	将在“符号”以及“编号”所指定的设备中指定的“点数”的点数份的位设备值储存在“客户端设备”中“点数”所指定的点数份的位设备中。
14010000 字单位写入	将在“符号”以及“编号”所指定的设备中指定的“点数”的点数份的字设备值写入“客户端设备”中“点数”所指定的点数份的字设备中。
14010001 位单位写入	将在“符号”以及“编号”所指定的设备中指定的“点数”的点数份的位设备值写入“客户端设备”中“点数”所指定的点数份的位设备中。

(8) 客户端设备

设置 Plus CPU 模块的设备。根据“指令”可以设置的设备种类不同。详情如下所示。

指令	输入 (I)	输出 (Q)	内部继电器 (M)	特殊内部继电器 (M)	数据寄存器 (D)	特殊数据寄存器 (D)
04010000 字单位读取	—	—	—	—	是	—
04010001 位单位读取	—	—	是	—	—	—
14010000 字单位写入	是	是	是	—	是	—
14010001 位单位写入	是	是	是	—	—	—

注释：请在可设备范围内设置客户端设备。

(9) 点数

可将批读取或批写入的个数设置为 1 到 512。

(10) 远程主机编号

设置与 Plus CPU 模块通信的 MC 协议兼容设备的 IP 地址和端口编号。

在网络上注册和管理通信对象设备（远程主机）的列表称为远程主机列表。有关远程主机列表的详细信息，请参见第 3-21 页上的“远程主机列表”。

注释： 请求执行顺序和连接间断时间

断开和连接所需的时间可能会根据执行请求的顺序而变化。

执行请求时，它会连接到远程主机并发送请求。连接将一直保持到下一个请求发出为止。如果下一个要执行的请求的远程主机与连接中的远程主机相同，则在保持连接的同时发送请求。如果远程主机不同，请断开连接，连接到下一个要执行的请求的远程主机，然后发送该请求。通过在同一远程主机上顺序执行已配置的请求，可以节省断开连接和连接所需的时间。

请求编号	远程主机编号
1	1
2	1
3	2
4	2

↓

当仅执行请求编号 1 和 3 时，现有连接会断开连接，并且将建立新连接。

↓

可以节省断开连接和连接所需的时间。

请求编号	远程主机编号
1	1
2	2
3	1
4	2

↓

对每个请求编号进行请求时，它都会断开现有连接并建立新连接。

↓

断开连接和连接需要花费时间。

(11) 服务器设备名称**(12) 服务器设备编号**

设置 MC 协议兼容设备的设备名称和设备编号。

符号

可以设置的设备因“指令”设置而异。详情如下所示

指令	输入 (X)	输出 (Y)	内部继电器 (M)	特殊内部继电器 (SM)	数据寄存器 (D)	特殊数据寄存器 (SD)
04010000 字单位读取	是	是	是	是	是	是
04010001 位单位读取	是	是	是	是	—	—
14010000 字单位写入	—	—	—	—	是	—
14010001 位单位写入	是	是	是	—	—	—

编号

根据“符号”的不同服务器设备编号的可设置范围也不同。详情如下所示。

符号	编号
输入 (X)	0000 ~ FFFFFFF*1
输出 (Y)	0000 ~ FFFFFFF*1
内部继电器 (M)	0 ~ 16777215
特殊内部继电器 (SM)	0 ~ 16777215
数据寄存器 (D)	0 ~ 16777215
特殊数据寄存器 (SD)	0 ~ 16777215

*1 以十六进制设置。

注释： 可设置的设备编号根据于 MC 协议的兼容设备而不同。检查 MC 兼容协议设备的手册基础上，请设置不超出设备范围的“服务器设备编号”。

18: MC 协议通信

(13) 请求执行设备

显示分配给每个请求编号的请求执行设备。仅在选中“请求执行设备”的复选框时进行。

注释: 如果删除请求列表中的请求、更换请求的顺序或修改现有请求的请求编号, 则该请求的请求执行设备也将被修改。在这种情况下, 可能会与用户程序中使用的设备发生不匹配, 请仔细确认用户程序。

(14) 联机状态

选中“同步自动 Ping”的复选框时, 在“自动 Ping 设置”对话框中显示各远程主机的“联机状态”中设置的内部继电器。“自动 Ping 设置”对话框的详情请参见第 3-21 页上的“远程主机列表”。

(15) 错误状态

(16) 结束代码

显示分配给每个请求编号的通信执行设备。仅在选中“请求执行设备”的复选框时进行。

注释: 如果未选中“所有通信请求均使用一个 DR”复选框, 在删除请求列表中的请求, 或更换请求的顺序, 修改现有请求的请求编号, 则该请求的通信执行设备也将被修改。在这种情况下, 可能会与用户程序中使用的设备发生不匹配, 请仔细确认用户程序。

(17) 错误信息显示区域

如果在“MC 协议客户端”对话框中设置的内容中有错误, 则显示错误的内容。

(18) “通信设置”按钮

单击“通信设置”按钮打开“通信设置”对话框。有关“通信设置”对话框的详细信息, 请参见第 18-10 页上的“通信设置对话框”。

(19) “导入”按钮

导入请求列表的内容。

(20) “导出”按钮

导出请求列表的内容。

可以批量编辑请求列表的内容, 也可导入其他项目。

(21) “确定”按钮

保存设定并关闭“MC 协议客户端”对话框。

(22) “取消”按钮

不保存设定并关闭“MC 协议客户端”对话框。

通信设置对话框

使用此对话框可以进行 MC 协议通信的详细设置。



(1) 接收超时

设置从 Plus CPU 模块向 MC 协议兼容设备发送数据到从 MC 协议兼容设备返回响应之间的等待时间。设置范围为 300 毫秒至 25500 毫秒 (间隔为 100 毫秒)。

(2) 发送等待时间

设置两次执行请求之间的等待时间, 范围为 0ms 至 5000ms。

MC 协议客户端操作

请求处理和通信是独立的，并且请求列表中的每个请求都按照请求编号的顺序由 END 处理重复执行。通信与用户程序的执行是非同步的。是否执行请求取决于以下设置的组合。详情如下所示。

设置		同步自动 Ping		
		不使用	使用	
			自动 Ping 的发送对象	不是自动 Ping 的发送对象
请求执行设备	不使用	执行请求。	请求的“联机状态”在打开时执行请求；在关闭时不执行请求。	执行请求。
	使用	请求的“请求执行设备”在打开时执行请求；在关闭时不执行请求。	请求的“联机状态”以及“请求执行设备”都为打开时执行请求，否则不执行请求。	请求的“请求执行设备”在打开时执行请求；在关闭时不执行请求。

请求的执行完成后，将执行下一个编号的请求。以下情况，请求都将完成：

- 使用“通信执行设备”的情况下，为执行请求，通信执行设备时自动由打开变为关闭
- 使用“状态”的情况下，更新存储错误状态的数据寄存器的值时

如果请求成功执行（正常接收应答）

写入请求错误代码（0x00）和结束代码（0x0000），并且在“发送等待时间”中设置的时间过去后，将执行下一个请求。仅当在“状态”中选中“使用”并且清除“仅在通信失败时更新状态”复选框时，才写入错误代码和结束代码。

如果请求执行失败

0x0000 以外的值被写入请求错误代码和结束代码。仅当写入错误代码和结束代码并且在“状态”中选择“使用”时，才执行此操作。在以下情况下，在“发送等待时间”中设置的时间过去后，将执行下一个请求。

- 收到异常响应时
- 当无法接收到响应（正常响应或异常响应）并且即使重试两次后所有重试均失败。

注释：

- 使用请求执行设备时，执行请求后，对应的请求执行设备将自动关闭。需要可随时发送请求时，请始终使用用户程序的 OUT 指令将打开相应的请求执行设备。
- 通信执行设备发生错误时，在两次重试失败后，将自动关闭。
- 如果请求中设置的远程主机编号（以下称为主机①）与上一个请求中设置的远程主机编号（以下称为主机②）不同，则切断主机②号的远程主机连接，并执行下一个请求。如果请求中设置的主机①与上一个请求中设置的主机②号同，则保持连接并执行下一个请求。

• 使用的特殊设备

在 MC 协议通信中所使用的的特殊设备如下所示。

地址	功能	说明	读 / 写
D8278	通信模式信息（链接 1 ~ 4）	显示连接号 1 ~ 4 的通信模式。	读
D8279	通信模式信息（链接 5 ~ 8）	显示连接号 5 ~ 8 的通信模式。	读
D8760	通信模式信息（链接 9 ~ 12）	显示连接号 9 ~ 12 的通信模式。	读
D8761	通信模式信息（链接 13 ~ 16）	显示连接号 13 ~ 16 的通信模式。	读

设备内的分配，请参见第 2-19 页上的“特殊数据寄存器”。

索引

- #** 1:1 维护通信系统 1-9
- A**
- ADD-2comp 5-58
 - Analog Input 对象 15-34, 15-45
 - Present_Value 设置 15-21
 - Analog Output 对象 15-35, 15-46
 - Present_Value 设置 15-23
 - Analog Value 对象 15-36, 15-47
 - Present_Value 设置 15-25
 - Application_Software_Version 15-56
 - ASCII
 - 字符代码表 5-53
 - ASCII 字符代码表 5-53
 - Assembly 对象 16-43
- B**
- BACnet/IP 15-1
 - BACnet/IP 的动作 15-11
 - BACnet/IP 的设置 15-13
 - BIBB 15-4
 - Binary Input 对象 15-37, 15-50
 - Present_Value 设置 15-27
 - Binary Output 对象 15-38, 15-51
 - Present_Value 设置 15-29
 - Binary Value 对象 15-39, 15-52
 - Present_Value 设置 15-30
 - Bluetooth 的维护通信 4-6, 4-16
 - Bluetooth 设备设置 9-7
 - Bluetooth 通信盒设置 9-4
 - Bluetooth 设备列表 9-6
 - Bluetooth 通信盒 1-4
 - 本地主机端口号 4-9, 4-19, 6-30
 - 比较 BCC 代码 5-20
- C**
- 操作模式设置对话框 17-36
 - CGI
 - 读取设备数据 13-56
 - 写入设备数据 13-57
 - 插入数据（设备）对话框 17-48
 - 从机编号 4-15
 - Connection Manager 对象 16-43
 - COV_Increment 15-54
 - 常量 5-16
 - 串行端口 1 1-3
 - 串行接口
 - 模块 7-10
 - 串行通信模块 1-3
 - 从机编号 4-5
 - 从机站
 - 通信完成继电器
 - M8080-M8116 7-6
 - M8117 7-6
 - 错误
 - 代码
 - 数据连接通信 7-4
 - 用户通信 5-52
 - 错误代码
 - 电子邮件 12-2
- D**
- D1 5-8
 - D1 (目标 1) 5-21
 - D2 5-8
 - D2 (目标 2) 5-21
 - D2+0 (发送状态寄存器) 5-8
 - D2+0 (接收状态) 5-21
 - D2+1 (发送字节数) 5-9
 - D2+1 (接收字节数) 5-21
 - 单元 ID 6-32
 - 导入 JSON 文本对话框 17-46
 - Device 对象 15-40
 - DSR
 - 控制信号状态 5-34
 - DTR
 - 控制信号状态 5-34
 - 输出控制信号选项 5-37
 - 对象 15-33, 16-41
 - 对象设置 15-20
 - 对象一览 15-32
 - 对象与设备的联动功能 15-5
 - 电缆 5-54
 - 电子邮件
 - 常规设置 12-6
 - 错误代码 12-2
 - 十进制符号 12-10
 - 数据寄存器 12-15
 - 特殊数据寄存器 12-3
 - 文本编码 12-10
 - 验证（登录） 12-7
 - 字符集 12-10
 - 电子邮件编辑器 12-13
 - 订阅 17-37
 - 订阅标题 17-57
 - 端口
 - CAN 1-14
 - Modbus 通信 1-11
 - 数据连接通信 1-11
 - 用户通信 1-10
 - 端口 1 的维护通信 4-4
 - 端口号 3-21
 - 多点监控 13-27
- E**
- EDS 文件 16-32
 - ERXD 5-43
 - Ethernet Link 对象 16-45
 - EtherNet/IP 的设置示例 16-35
 - EtherNet/IP 通信的设置 16-18
 - EtherNet/IP 通信 1-13, 16-1
 - ETXD 5-43
 - Explicit 信息通信（服务器）功能 16-10
- F**
- 发布 17-32
 - 发布标题 17-49
 - 分配 ERXD（以太网用户通信接收）指令的接收取消标记 5-50
 - Firmware_Revision 15-56
 - 发送
 - 数据 5-4

- 位数 5-5
- 字节 5-5
- 分隔符 12-10
- 附件编辑器 12-16
- 服务 15-3
- 服务器
 - 用户通信 5-47
- G**
 - 功能代码 6-32
 - 高级指令
 - ERXD 5-23
 - ETXD 5-23
 - 功能代码 6-27
 - 规格
 - 数据连接 7-1
 - 用户通信模式 5-24
- J**
 - I/O 信息通信功能 16-5
 - J1939 通信 8-1
 - 寄存器
 - 用于发送 / 接收数据 7-3
 - Identity 对象 16-41
 - 交易 ID 6-32
 - 接收超时 4-15
 - 借位 2-11
 - 进位 2-11
 - 进位 / 借位 2-11
 - 奇偶校验 4-15
 - IP 地址 3-21, 4-10
 - 监控 / 更改
 - 设备值 4-1
 - 监控功能 13-35
 - 接收
 - 超时 5-25
 - 格式 5-10, 5-11
 - 完成输出 5-10
 - 位数 5-12
 - 指令取消标志 5-22
 - 状态 5-10
 - 接收超时 (毫秒) 4-5, 4-9, 4-19
 - 结束分隔符 5-16
- K**
 - 控制信号状态 2-35
 - 扩展通信端口的维护通信 4-13
 - 开始
 - 分隔符 5-13
 - 控制信号
 - 选项
 - DTR 5-37
 - 状态 5-32, 5-34
- L**
 - LRC 5-58
 - 连接状态 3-15
- M**
 - MC 协议客户端 对话框 18-4
 - MC 协议通信
 - 错误状态 18-7
 - 服务器设备编号 18-9
 - 服务器设备名称 18-9
 - 结束代码 18-7
 - 远程主机编号 18-9
 - 指令 18-8
 - Message Router 对象 16-42
 - Modbus 6-1
 - ASCII 5-58
 - 通信 1-11
 - 通过 RS232C/RS485 进行 6-2
 - 通过以太网通信进行 6-21
 - Modbus TCP
 - 服务器规格 6-27
 - modbus TCP
 - 错误状态 6-26
 - 服务器 6-27
 - 功能代码 6-25
 - 客户端 3-21, 6-22
 - 客户端规格 6-22
 - 请求执行设备 6-26
 - 通信错误 6-22
 - 通信格式 6-30
 - 远程主机号 6-26
 - Modbus 主机请求表 6-5
 - MQTT 设置 17-6
 - MQTT 设置对话框 17-6
 - MQTT 通信 17-1
 - MQTT 通信的设置 17-5
 - MQTT 通信 1-14
 - 忙
 - 控制 5-36
 - 信号 5-54
 - O**
 - Object_List 15-56
 - Out_Of_Service 15-55
 - P**
 - 批处理对话框 17-45
 - PING
 - 指令 11-1
 - ping
 - 错误代码 11-2
 - PLC 状态 13-26
 - Polarity 15-54
 - Present_Value 15-41
 - Present_Value 设置 15-21
 - Priority_Array 15-54
 - programming
 - Modbus master using WindLDR 18-2
 - Protocol_Object_Types_Supported 15-56
 - Protocol_Services_Supported 15-56
 - Q**
 - 奇偶校验 4-5
 - 趋势图 13-41
 - 确认输入数据 5-17
 - R**
 - Reliability 15-55
 - Relinquish_Default 15-54
 - RS232C
 - 线控信号 5-32
 - RS232C DSR 控制信号状态 2-37
 - RS232C DTR 输出控制信号选项 2-39
 - RS232C 通信盒 1-4
 - RS485
 - 通信适配器 7-2
 - RS485 通信盒 1-4
 - RUN 模式下的 DTR 控制信号状态 5-34
 - RUN 模式控制信号状态 5-34
 - S**
 - 设备地址 2-1
 - 数据长度 4-15
 - 数据连接

- 规格 8-1
 - 属性的读取功能 15-6
 - 属性的写入功能 15-6
 - SNTP 设置 3-11
 - Status_Flags 15-54
 - STOP 模式下的 DTR 控制信号状态 5-34
 - STOP 模式控制信号状态 5-34
 - Subscribed COV (COV) 功能 15-7
 - System_Status 15-56
 - 上传
 - 用户程序 4-1
 - 设备数据读取函数 13-43
 - 设备数据写入函数 13-43
 - 设置
 - 使用 WindLDR 的 Modbus 从机 6-13
 - 使用 WindLDR 的 RXD 指令 5-29
 - 使用 WindLDR 的 TXD 指令 5-26
 - 使用 WindLDR 用户通信 5-25
 - 数据连接使用 WindLDR 7-7
 - 特殊数据寄存器 5-54
 - 设置列表 3-1
 - 十进制符号
 - 电子邮件 12-10
 - 使用 BACnet 通信 1-12
 - 使用 J1939 通信 1-12
 - 使用 WindLDR 设置
 - 主机 6-5
 - 数据
 - 设置就绪 DSR 5-35
 - 刷新 7-10
 - 数据长度 4-5
 - 数据连接
 - 其他 PLC 7-10
 - 通信 7-1
 - 初始化标记 M8007 7-6
 - 错误 7-4
 - 错误代码 7-4
 - 错误 M8005 7-6
 - 禁止标记 M8006 7-6
 - 停止标记 M8007 7-6
- T**
- TCP/IP Interface 对象 16-44
 - 特殊内部继电器设备地址 2-4
 - 特殊数据寄存器一览 2-19
 - 停止位 4-15
 - 通过串行通信 (Bluetooth) 进行的用户通信 5-39
 - 通信速度 4-15
 - 特殊功能 3-1
 - 特殊内部继电器
 - 用于数据连接通信 7-6
 - 特殊数据寄存器的特殊数据寄存器 5-32
 - 电子邮件 12-3
 - 用于数据连接通信错误 7-4
 - 跳过 5-18
 - 条形图
 - 横 13-39
 - 纵 13-37
 - 停止位 4-5
 - 通信 6-1
 - 参数 6-6, 6-13
 - 设置 5-25, 5-55, 5-56
 - 完成继电器 M8080 7-6
 - 选项卡 5-25, 6-5, 6-13, 7-7, 7-8
 - 通信盒的维护通信 4-13
 - 通信盒信息 2-34
 - 通信接口 1-3
 - 通信速度 (bps) 4-5
- W**
- 外部设备功能 15-10
 - 网络设置 3-3
 - Web 服务器设置 10-4
 - WindLDR
 - programming
 - Modbus
 - master 18-2
 - 设置
 - Modbus 从机 6-13
 - Modbus 主机 6-5
 - RXD 指令 5-29
 - 数据连接 7-7
 - TXD 指令 5-26
 - 用户通信 5-25
 - Unsubscribed COV (COVU) 功能 15-9
 - USB 端口 1-3
 - 网络服务器 13-1
 - 多点监控 13-27
 - PLC 状态 13-26
 - 注册监控 13-28
 - 网页编辑器 13-44
 - 维护
 - 通信 1-9, 4-1
 - 通过 USB 端口 4-3
 - 通过以太网端口 4-7
 - 文本编码
 - 电子邮件 12-10
- X**
- 协议 ID 6-32
 - 新建数值或编辑对话框 17-43
 - 信息长度 6-32
 - 系统
 - 安装
 - 数据连接 7-2
 - 选项连接信息 2-34
 - 下载
 - 用户程序 4-1
 - 线路
 - 控信号 RS232C 5-32
- Y**
- 用户通信 (UDP) 5-50
 - 用户帐户设置 9-12, 10-5
 - 验证
 - 电子邮件 12-7
 - 以太网
 - 用户通信 3-21
 - 用户
 - 通信 1-10
 - 通过串行通信进行 5-24
 - 通过以太网通信进行 5-40
 - 用户通信
 - 常量 5-16
 - 错误 5-52
 - 代码 5-52
 - 服务器 5-47
 - 接收指令取消标志 5-22
 - 取消标记 5-44, 5-47

- 指令 5-1
- 用户网页的创建 13-44
- 用户网页的动作确认 13-47
- 用户网页树状的操作 13-30
- 有效载荷对话框 17-41
- 远程主机号
 - modbus TCP 6-26
- 元字符的格式
 - 字设备 13-35
- 允许 IP 地址访问 4-9, 4-19, 6-30
- Z**
- 支持的 Web 浏览器 13-56
- 指令
 - ERXD 5-43
 - ETXD 5-43
 - PING 11-1
- 用户通信 5-1
- 重复
 - 次数 5-6, 5-12
- 注册监控 13-28
- 主机
 - 站 7-7
- 主机名 3-21
- 转换
 - 类型 5-5, 5-12
- 字符集
 - 电子邮件 12-10
- 纵向冗余校验 5-58

产品保修说明

(1) 保修期限

客户所购产品的保修期限，自原始购买日，或产品送达指定地点之日起享有 3 年保修期。

* 超出保修范围的情形

电池已达到使用寿命，以及继电器开闭次数超过限定值（10 万次）

(2) 保修范围

在上述保修期限内发生非用户原因造成的故障，本公司将负责对相应的产品故障部分进行免费更换，或提供免费维修。

万一出现故障时，请您携带能够直接证明您购入本产品日期的材料，至销售门店或本公司申请保修服务。

* 因更换产品所产生的安装及工程费用，需由客户自行承担。

(3) 保修免责条款

以下情况所导致的故障不在保修范围内。

- 1) 在超出样本、性能规格书、使用说明书中记载的条件以及环境下使用产品的情况
- 2) 故障并非是由所购产品导致的情况
- 3) 产品经过非本公司人员改造或修理的情况
- 4) 将产品用于原设计用途以外用途的情况
- 5) 因火灾、地震、水灾、闪电及其他自然灾害，或异常电压（电压频率）等，非本公司原因导致的故障及损坏等
- 6) 购入后因移机、搬运、掉落等导致的故障及损坏
- 7) 因安装不当导致的故障及损坏
- 8) 因未遵照使用说明书中规定的维护与检修所导致的故障及损坏

* 客户有责任检查任何可编程产品的操作，风险自担。在任何情况下，本公司不对客户所设定的程序操作或因操作而造成的损失负责。

此外，本条款中涉及的保修，仅针对产品个体本身，对于因产品故障而间接产生的损失，请恕本公司概不负责。

(4) 有偿服务项目

产品价格中，不含技术人员外派等服务费用，下列费用需由客户自行承担。

- 1) 安装调试指导及试运转跟进服务（含应用程序编程、操作试验等）
- 2) 维护检修、调整及维修
- 3) 技术指导及技术教育
- 4) 应客户要求实施的产品试验及检查

IDEC 株式会社

日本大阪府大阪市淀川区西宫原 2-6-64



IDEC China Apps

爱德克电气贸易（上海）有限公司

北京分公司

广州分公司

香港和泉电气有限公司

 idecchina.cn

200070 上海市静安区共和路 209 号 企业中心第二座 8 楼
电话：021-6135-1515 传真：021-6135-6225/6226 E-mail: idec@cn.idec.com

100026 北京市朝阳区光华路甲 8 号 和乔大厦 B 座 310 室
电话：010-6581-6131 传真：010-6581-5119

510610 广州市天河区林和西路 157 号 保利中汇广场 A 栋 907 号
电话：020-8362-2394 传真：020-8362-2394

香港九龙观塘观塘道 370 号 创纪之城 3 期 16 楼 01 室
电话：852-2803-8989 传真：852-2565-0171/2561-8732 E-mail: info@hk.idec.com

- 本手册内所记载的公司名称以及商品名称，为各公司的注册商标。
- 本手册中的规格及其他说明若有改变，恕不另行通知。

B-1731 (17) 本资料记载内容为 2023 年 11 月的信息。

 IDEC