

OMRON

CP 系列功能块实践指南

Modbus RTU 主站

关于知识产权和商标

微软产品屏幕截图的翻印已获得微软公司许可。

Windows 是微软公司在美国和其他国家的注册商标。

本指南引用的公司名称和产品名称均是其各自持有者的商标或注册商标。

■ 简介

本指南介绍了功能块使用示例。

欧姆龙不保证功能块在实际程序和机器中能够始终正常工作。请获取使用设备的用户手册，并确保在尝试开始操作之前了解手册中的重要注意事项和提醒。

■ 面向读者

本指南主要供下列人员使用，这些人员必须具备电气系统相关知识(电气工程师或同等水平者)。

- 负责 FA 系统安装的人员
- 负载 FA 系统设计的人员
- 负责管理 FA 系统及设备的人员

■ 相关手册

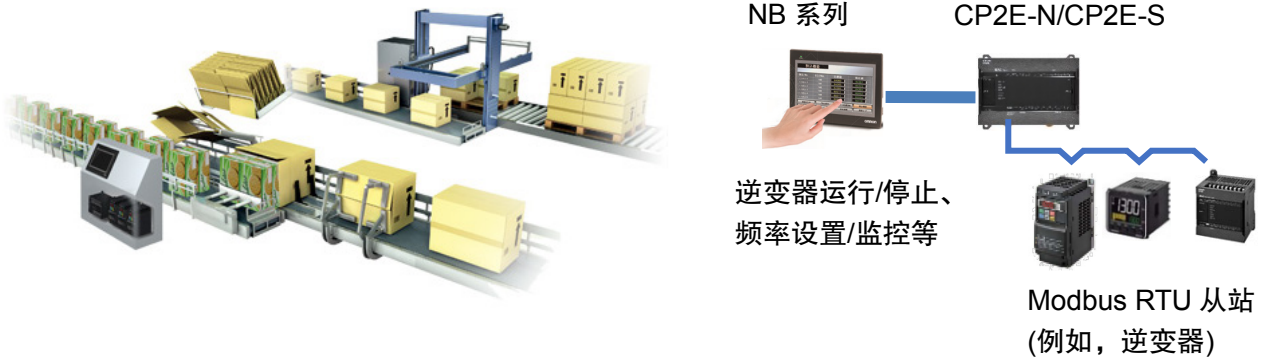
样本编号	型号	手册名称
W613 (CP2E)	CP2E-E□□D□-□ CP2E-S□□D□-□ CP2E-N□□D□-□	CP 系列 CP2E CPU 单元硬件操作手册
W614 (CP2E)	CP2E-E□□D□-□ CP2E-S□□D□-□ CP2E-N□□D□-□	CP 系列 CP2E CPU 单元软件操作手册
W483 (CP1E/CP2E)	CP1E-E□□D□-□ CP1E-N□□D□-□ CP2E-E□□D□-□ CP2E-S□□D□-□ CP2E-N□□D□-□	CP 系列 CP1E/CP2E CPU 单元指令参考手册
W446	CXONE-AL□□D-V4	CX-Programmer Ver.9.□操作手册

1. Modbus RTU 主站功能块

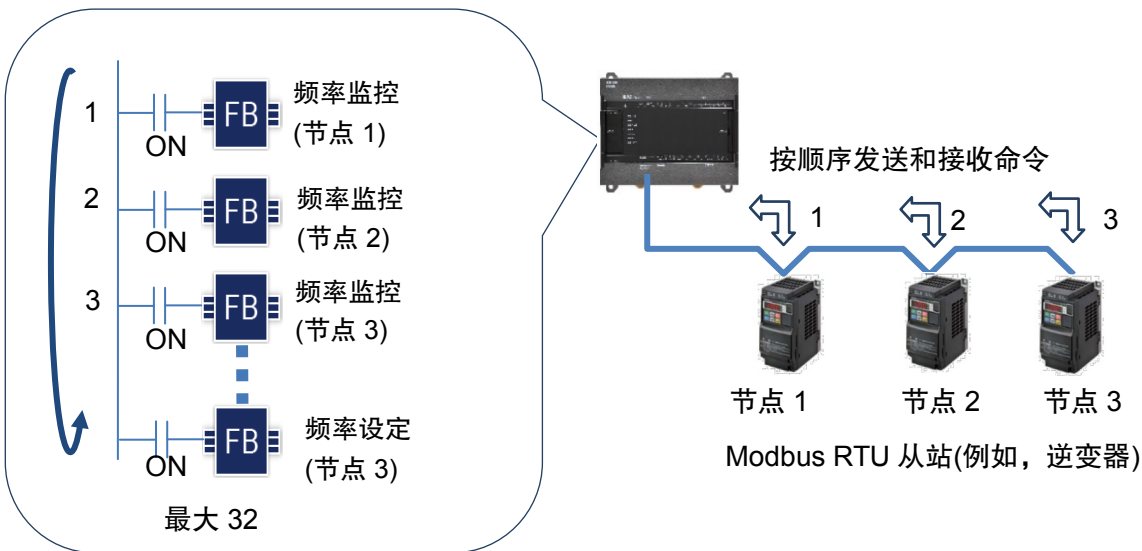
Modbus RTU 主站功能块使用串行端口发送 Modbus RTU 命令。

1.1 功能块概述

Modbus RTU 是广泛使用的串行通信协议，用于在 CP2E CPU 单元和设备之间进行数据交换。一个应用示例为包装机速度控制，在此应用中，RS-485 串行通信用于直接从 CP2E CPU 单元设置和监控逆变器。



启动标志在设定地址和数据后置 ON 时,此功能块通过端口自动且按顺序发送和接收多达 32 个 Modbus RTU 命令。



启动标志同时置 ON 时，按顺序发送和接收 Modbus RTU 命令。

<以往>

在 DM 分配区设置命令和数据，以使 CP2E CPU 单元的 Modbus-RTU 简易主站功能可通过串行端口发送和接收 Modbus RTU 命令。用户必须创建顺序控制程序以按顺序发送和接收多个命令，因为 CP2E CPU 单元的每个端口只能发送或接收一个命令。

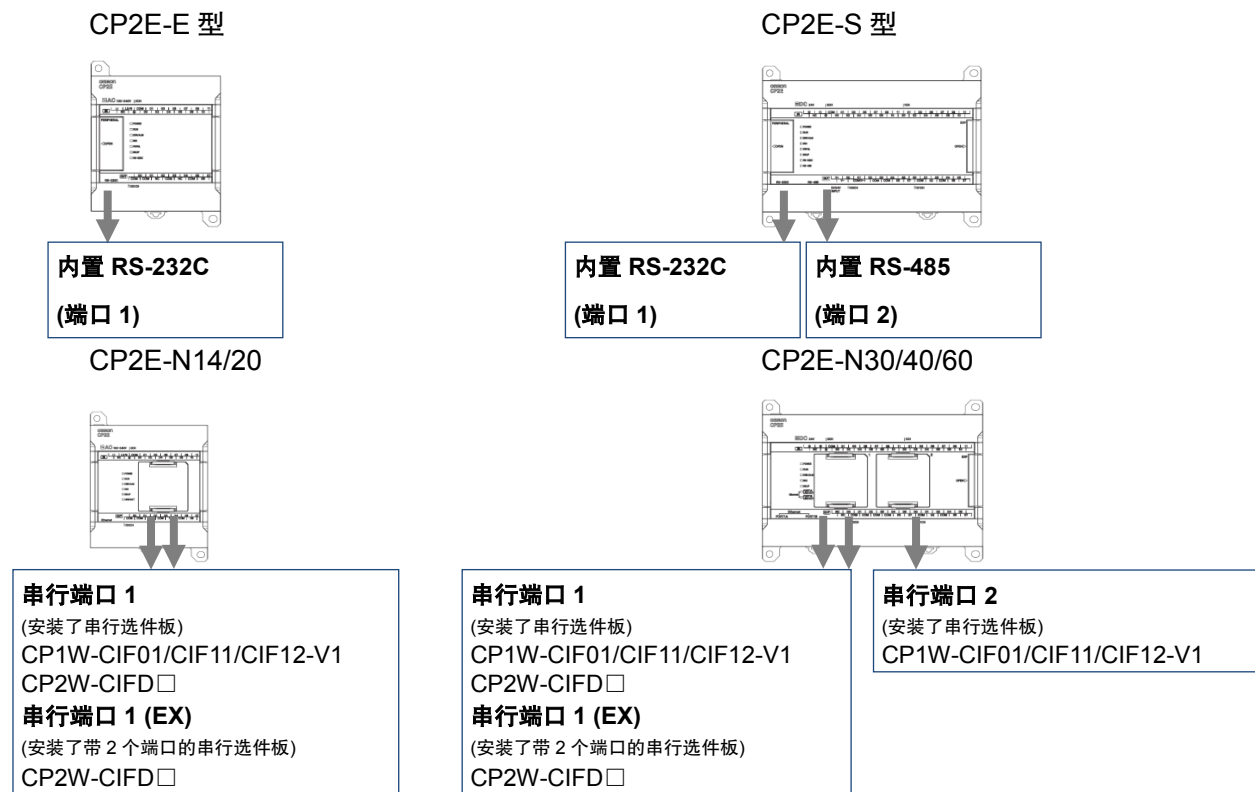
1.2 使用的功能块

Modbus RTU 主站功能块通过串行端口发送 Modbus RTU 命令。每个命令和每个串行端口提供一个功能块。有关功能块的详细信息，请参阅 *功能说明*。

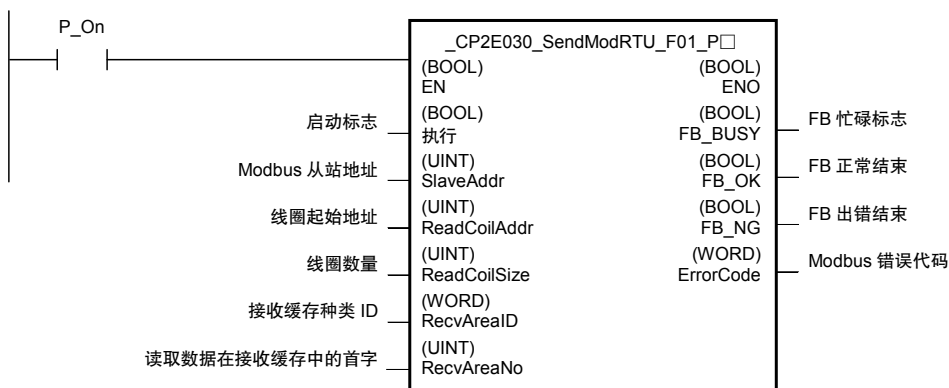
功能块

功能代码	命令名称	功能	串行端口	功能块名称
01 Hex	读取线圈	读取多个线圈	端口 1/内置 RS-232C	_CP2E030_SendModRTU_F01_P1
			端口 2/内置 RS-485	_CP2E030_SendModRTU_F01_P2
			端口 1 (EX)	_CP2E030_SendModRTU_F01_P1EX
03 Hex	读取保持寄存器	读取多个保持寄存器	端口 1/内置 RS-232C	_CP2E030_SendModRTU_F03_P1
			端口 2/内置 RS-485	_CP2E030_SendModRTU_F03_P2
			端口 1 (EX)	_CP2E030_SendModRTU_F03_P1EX
05 Hex	写入单线圈	写入单个线圈	端口 1/内置 RS-232C	_CP2E030_SendModRTU_F05_P1
			端口 2/内置 RS-485	_CP2E030_SendModRTU_F05_P2
			端口 1 (EX)	_CP2E030_SendModRTU_F05_P1EX
06 Hex	写入单寄存器	写入单个保持寄存器	端口 1/内置 RS-232C	_CP2E030_SendModRTU_F06_P1
			端口 2/内置 RS-485	_CP2E030_SendModRTU_F06_P2
			端口 1 (EX)	_CP2E030_SendModRTU_F06_P1EX
10 Hex	写入多寄存器	写入多个保持寄存器	端口 1/内置 RS-232C	_CP2E030_SendModRTU_F10_P1
			端口 2/内置 RS-485	_CP2E030_SendModRTU_F10_P2
			端口 1 (EX)	_CP2E030_SendModRTU_F10_P1EX

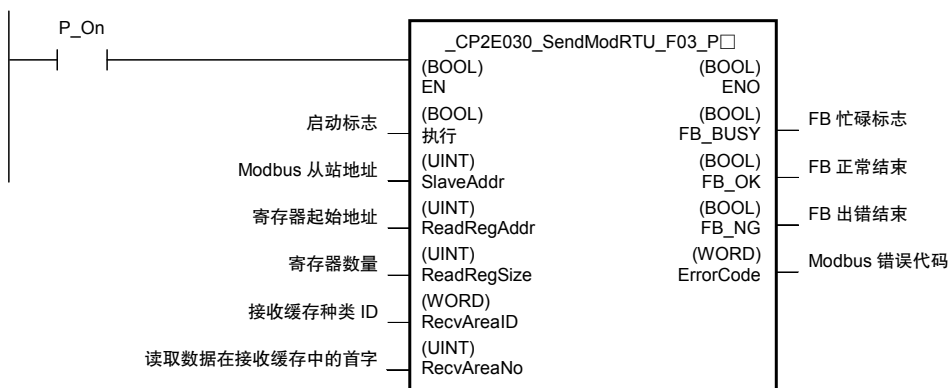
串行端口名称



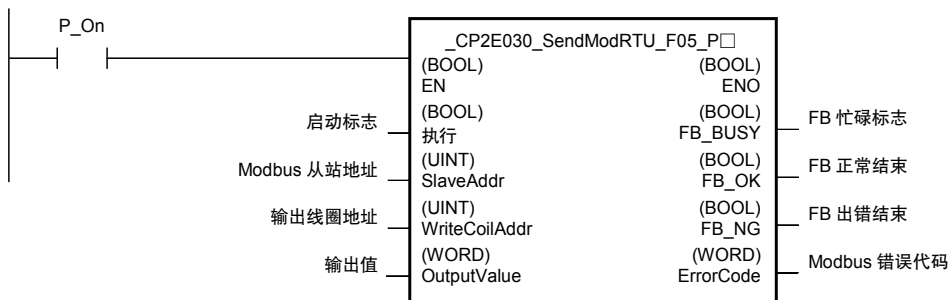
■ Function01: 读取线圈(_CP2E030_SendModRTU_F01_P□)



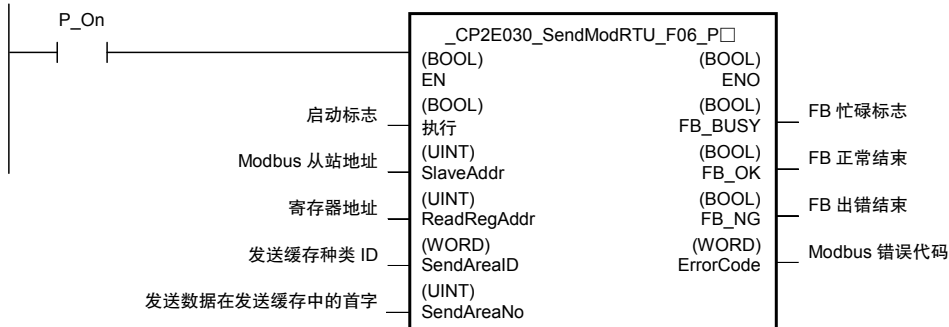
■ Function03: 读取保持寄存器(_CP2E030_SendModRTU_F03_P□)



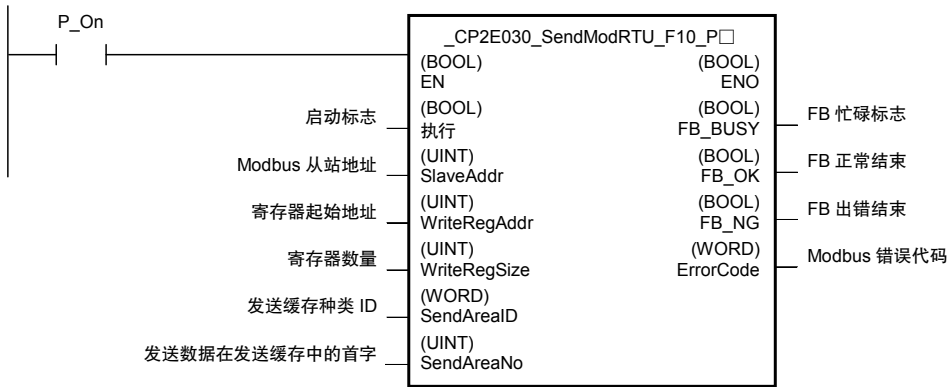
■ Function05: 写入单线圈(_CP2E030_SendModRTU_F05_P□)



■ Function06: 写入单寄存器(_CP2E030_SendModRTU_F06_P□)



■ Function10: 写入多寄存器(_CP2E030_SendModRTU_F10_P□)



功能块正确使用注意事项

- 这些功能块使用 Modbus RTU 简易主站功能。请勿对使用这些功能块的通信端口使用 Modbus-RTU 简易主站功能。
- 请勿将工作区用于程序。

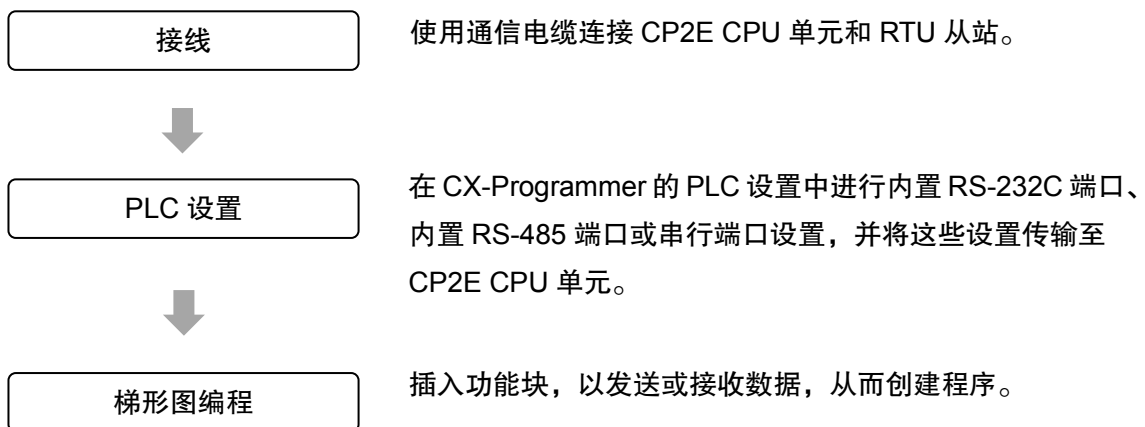
功能块工作区(该区域用于 Modbus RTU 主站功能块)

套接字号	DM 区	AR 区	工作区
内置 RS-232C 串行端口 1	D1200 ~ D1299	A640.00 ~ 02	W127, D3850 ~ D3885
内置 RS-485 串行端口 2	D1300 ~ D1399	A641.00 ~ 02	W127, D3900 ~ D3935
串行端口 1 (EX)	D1400 ~ D1499	A638.00 ~ 02	W127, D3950 ~ D3985

- 请勿同时将 EN 和启动标志(执行)置 ON。如果 EN 和 执行 同时置 ON，将不会发送命令。

2. 操作步骤

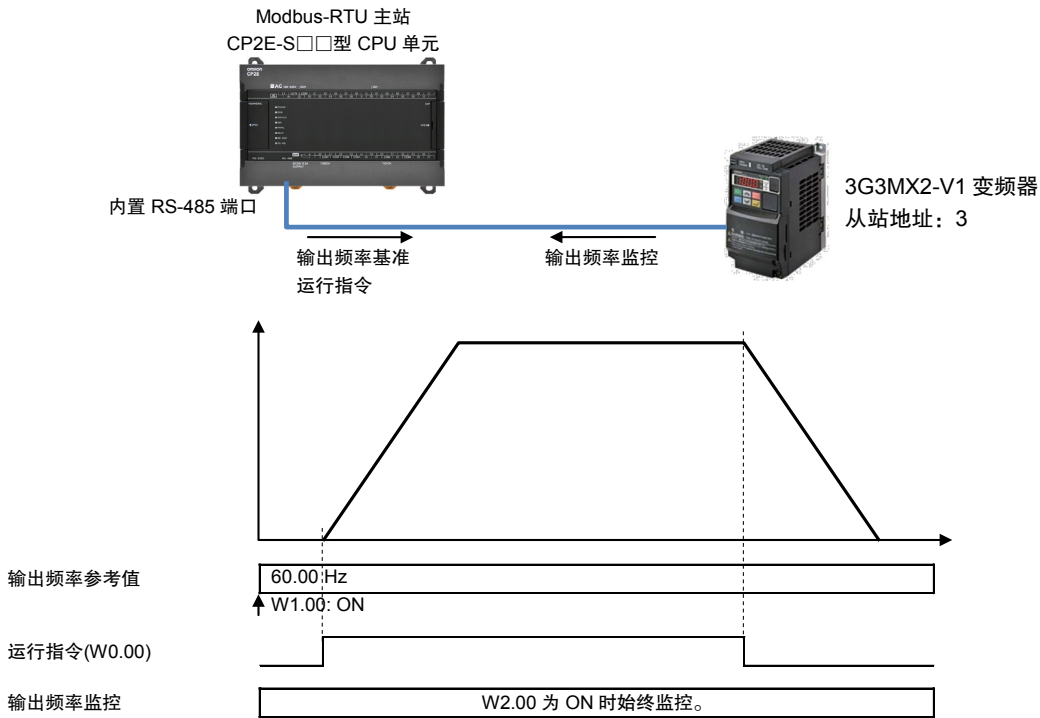
使用 Modbus RTU 通过 CP2E CPU 单元的串行端口与逆变器交换数据。



3. 编程示例

CP2E-S CPU 单元控制连接至 RS-485 端口的 3G3MX2-V1 逆变器。

CP2E-S CPU 单元始终监控逆变器的频率，同时更改频率并操作和停止逆变器。



执行以下操作：

1. 当 W1.00 置 ON 时，频率设置为 60.00 Hz。
2. 当 W0.00 置 ON 时，开始操作。当 W0.00 置 OFF 时，停止操作。
3. 当 W2.00 置 ON 时，始终监控频率。

* 逆变器根据其上的加速/减速和方向设置进行操作

使用以下功能块：

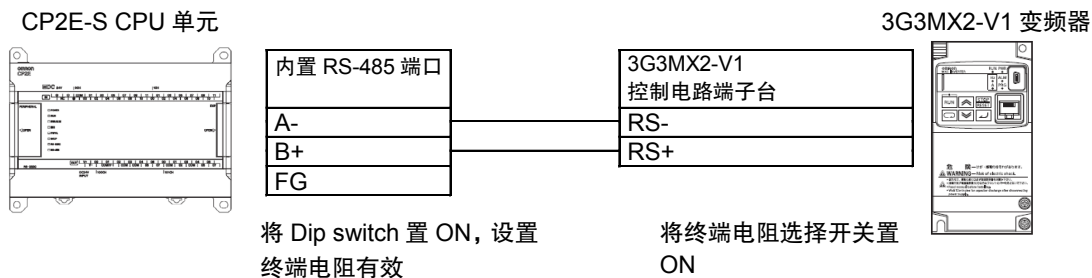
设定	CP2E 功能块		3G3MX2-V1 参数	
输出频率设置	_CP2E030_Send ModRTU_F06_P2	写入单个保持寄存器	输出频率设置参数 编号 F001(低)	Modbus 寄存器规格编号： 0001 Hex (寄存器编号：0002Hex-1)
运行指令	_CP2E030_Send ModRTU_F05_P2	写入单个线圈	运行指令 0：停止，1：运行	Modbus 线圈规格编号 0000 Hex (线圈号 0001Hex-1)
输出频率监控	_CP2E030_Send ModRTU_F03_P2	读取多个保持寄存器	输出频率监控参数 编号 d001(低)	Modbus 寄存器规格编号： 1001 Hex (寄存器编号：1002Hex-1)

使用以下 I/O 存储器：

I/O 存储器	描述	设定值	备注
D100	输出频率参考设置值	&6000	
D110	输出频率监控值	-	

3.1 接线示例

将 3G3MX2-V1 逆变器连接至 CP2E-S CPU 单元。



3.2 设定

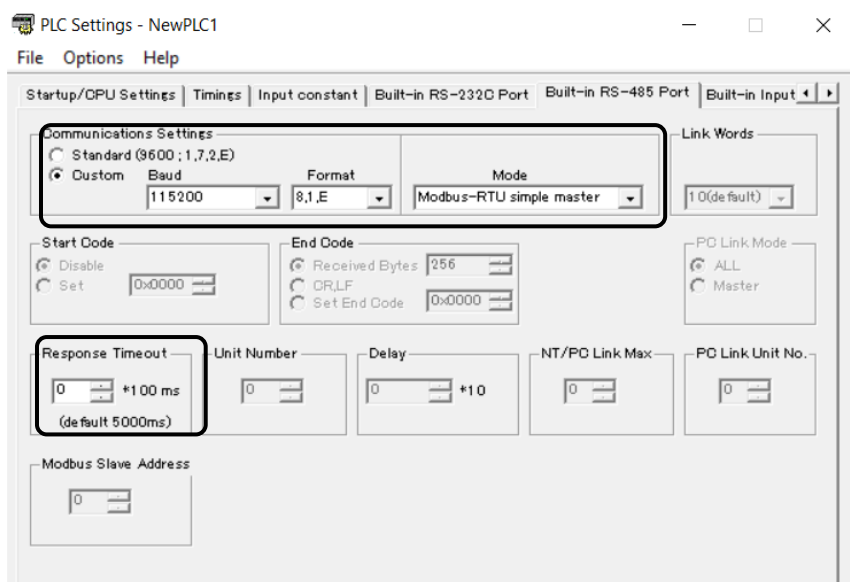
■ CP2E-S CPU 单元的 PLC 设置

(1) 串行端口设定

打开 CX-Programmer。

在“PLC Settings”(PLC 设置)中选择“Built-in RS-485 Port”(内置 RS-485 端口)标签。

设定 *Communications Settings*(通信设置)字段和 *Response Timeout*(响应超时)字段。



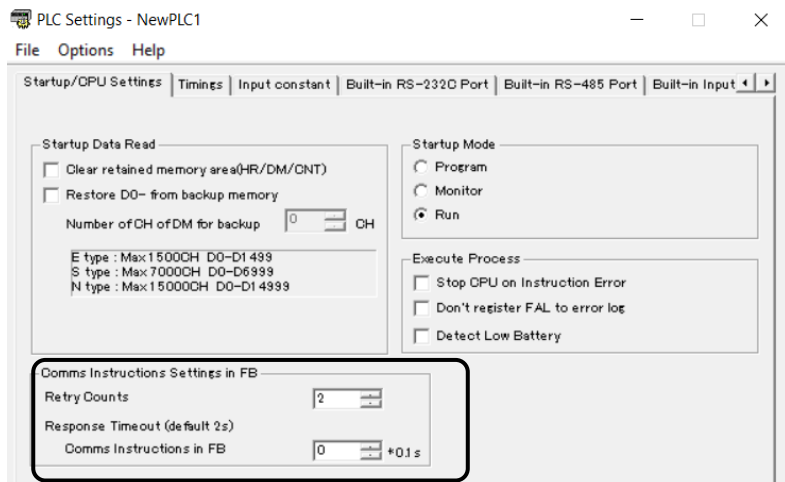
设定示例

项目	描述
通信设定	选择 <i>Custom</i> (自定义)选项，然后将波特率设定为“115200”，将格式设定为“8、1、E”。 * 使用与连接的逆变器相同的设置。
模式	选择 <i>Modbus-RTU simple master</i> (Modbus-RTU 简易主站)。
响应超时	设定值为 0(此时使用默认值：5000ms)

(2) 功能块通信设定

在“PLC Settings”(PLC 设置)中选择“Startup/CPU Settings”(启动/CPU 设置)标签。

在 *Comms Instructions Settings in FB*(FB 中的通信指令设定)字段中将重试计数设定为 2。



■ 设定 3G3MX2-V1

根据 CP2E CPU 单元的设置设定 Modbus 通信参数。

参数编号	功能名称	数据	默认值	设定值
A001/A201	第一/第二频率指令选择	03: Modbus 通信	02	03
A002/A202	第一/第二运行指令选择	03: Modbus 通信	02	03
C071	通信速度选择	03: 2400bps 04: 4800bps 05: 9600bps 06: 19.3 kbps 07: 38.4kbps 08: 57.6kbps 09: 76.8kbps 10: 115.2kbps	05	10 115.2kbps
C072	通信站号选择 (Modbus 从站地址)	1 ~ 247	1	3
C074	通信校验选择	00: 无校验 01: 偶校验 02: 奇校验	00	01 偶校验
C075	通信停止位选择	1: 1 位 2: 2 个位	1	1 1 位
C076	通信出错时的动作选择	00: 跳闸 01: 减速停止后跳闸 02: 无视 03: 自由运转停止 04: 减速停止	02	02 无视

参数编号	功能名称	数据	默认值	设定值
C077	通信错误超时	0.00: 禁止超时 0.01 ~ 99.99	0.00	0.00 禁用
C078	通信等待时间	0 ~ 1000	0 ms	0 ms

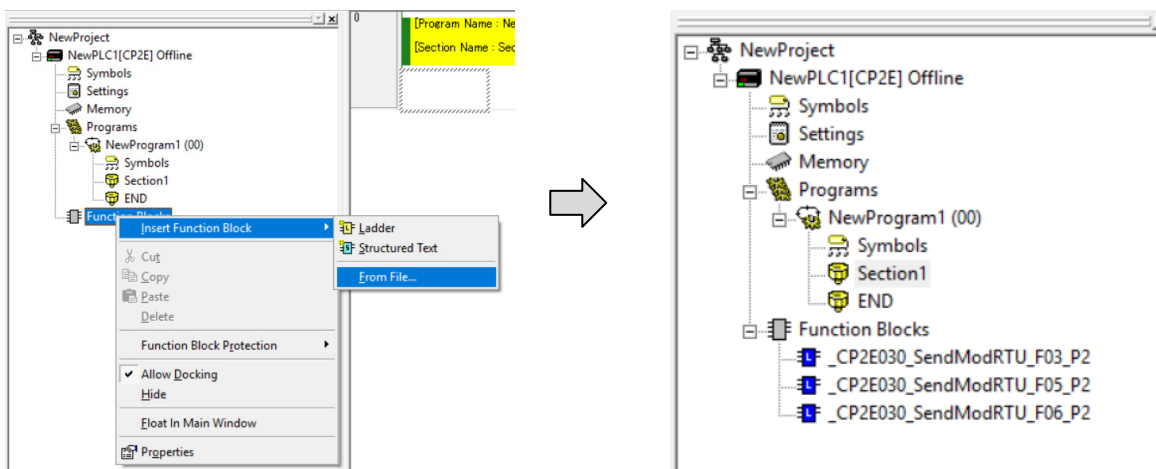
* 请参阅《MX2 系列 V1 型用户手册》(样本编号 I585), 以了解设定。

3.3 梯形图编程举例

(1) 插入功能块

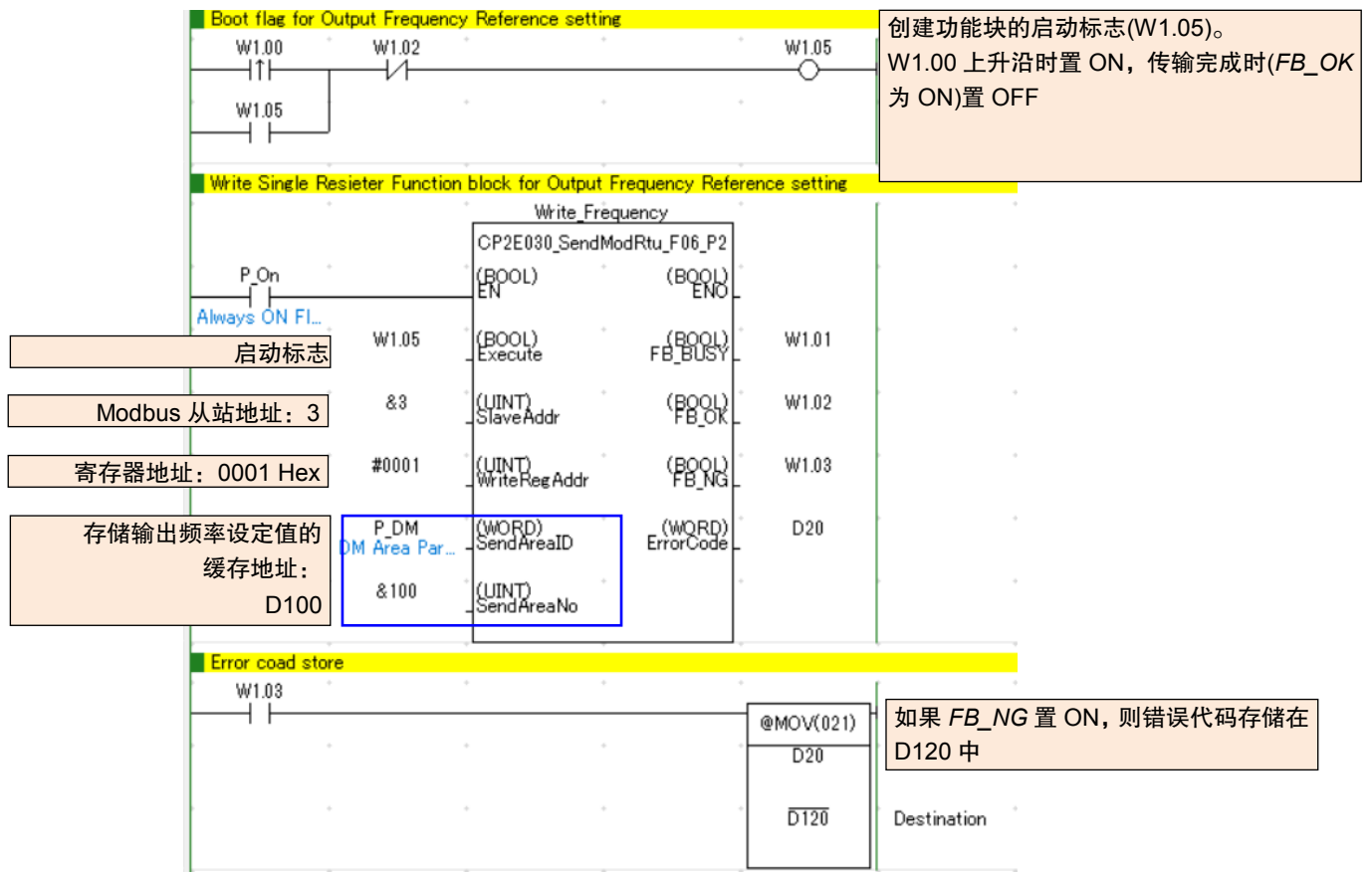
预先保存功能块文件, 以供 PC 使用。

右键单击 CX-Programmer 中的项目工作区的 **Function Blocks**(功能块), 然后从弹出菜单选择 **Insert Function Blocks - From File**(从文件插入功能块), 以载入 CP2E030_SendModRtu_F03_P2.cfx 文件、CP2E030_SendModRtu_F05_P2.cfx 文件和 _CP2E030_SendModRtu_F06_P2.cfx 文件。



(2) 频率参考设定的梯形图编程

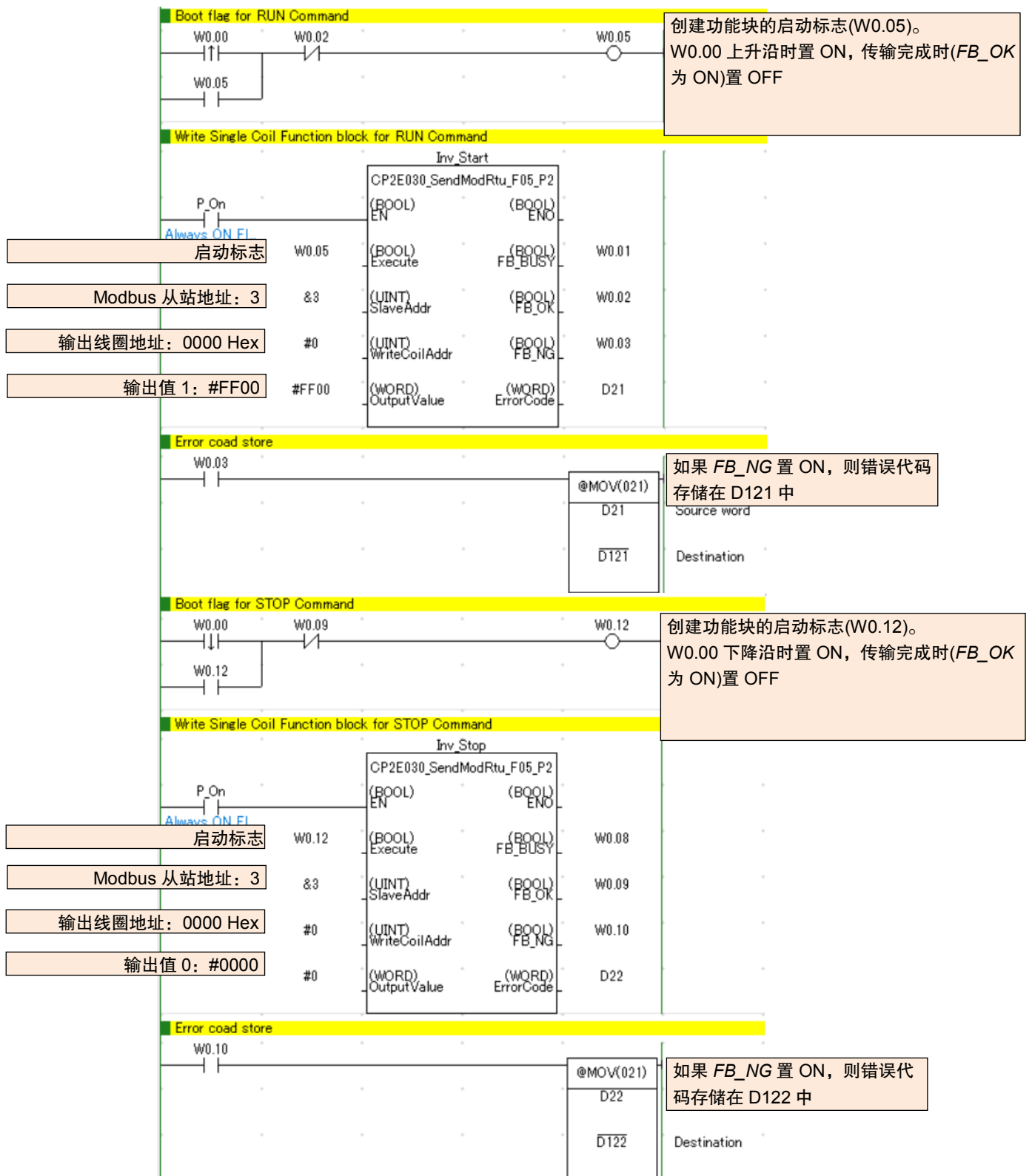
创建一个程序，以在 W1.00 的上升沿时将输出频率命令发送至 3G3MX2-V1 逆变器。



- 在频率命令 (W1.00) 的上升沿, 启动标志 (W1.05) 置 ON 时, 功能块发送命令, 将输出频率 (D100) 值写入 Modbus 从站地址为 3 的逆变器的寄存器地址为 0001Hex 的寄存器中。
(例如, 如果要设置输出频率为 60.00 HZ 时, 请将 D100 设定为&6000。)
- FB_BUSY (W1.01)在等待发送或发送期间置 ON。
- 发送完成后, FB_OK (W1.02)置 ON, 启动标志(W1.05)置 OFF。

(3) 启动和停止操作的梯形图编程

创建一个程序，以在 W0.00 的上升沿时启动操作，且在 W0.00 的下降沿时停止操作。

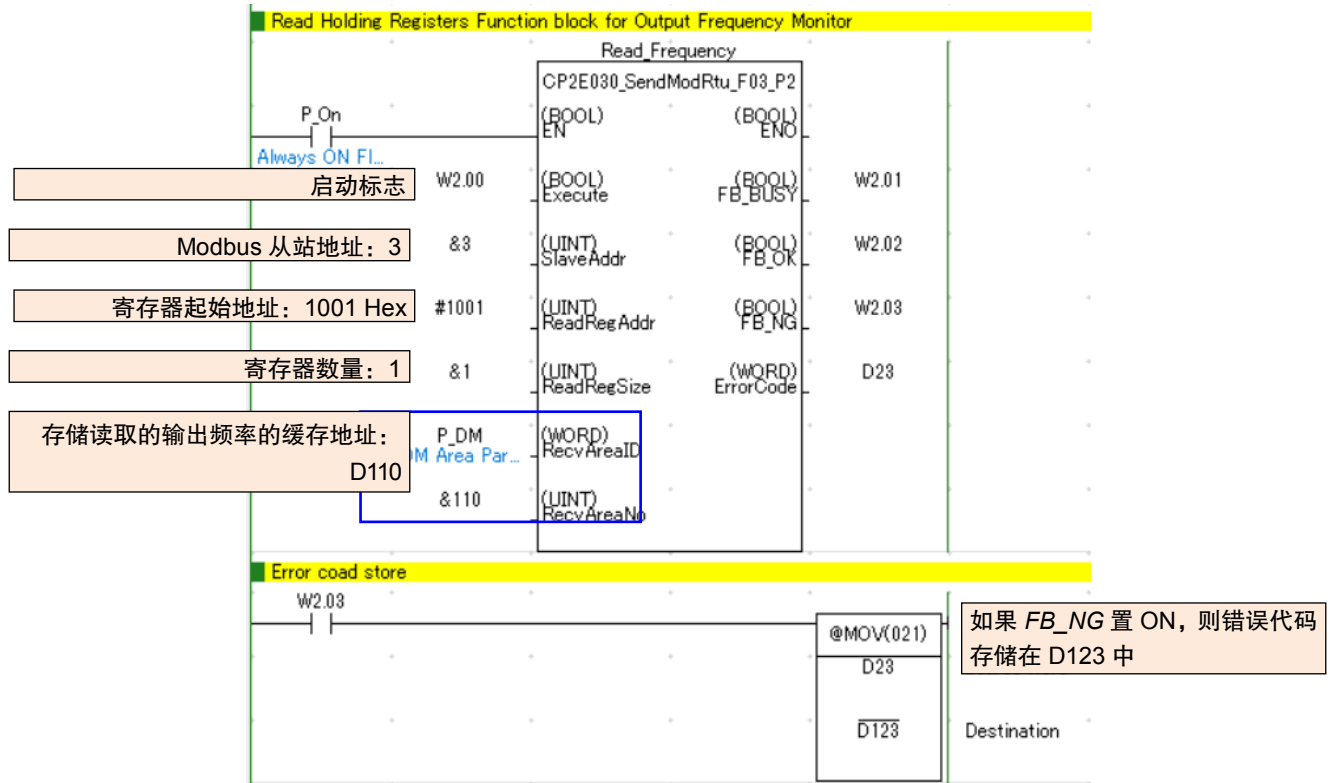


- 当运行起始位(W0.00)的上升沿，启动标志(W0.05)置 ON，功能块发送命令，将运行命令(1: #FF00)写入 Modbus 从站地址为 3 的逆变器的线圈地址为 0000Hex 的线圈中。

- 当运行起始位(W0.00)的下降沿、启动标志(W0.12)置 ON 时，功能块发送命令，将停止命令(0: #0000)写入 Modbus 从站地址为 3 的逆变器的线圈地址为 0000Hex 的线圈中。
- 为每个 FB 实例提供一个不同的名称。
- 如果功能块出错结束，则错误代码存储在 D121、D122 中。

(4) 输出频率监控的梯形图编程

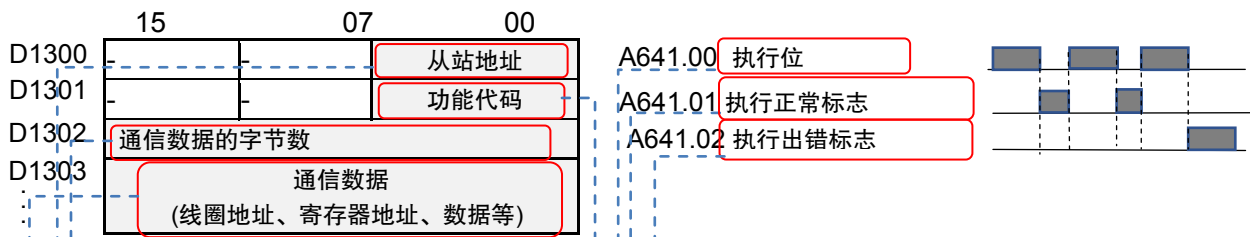
创建一个程序，以在 W2.00 置 ON 时监控输出频率。



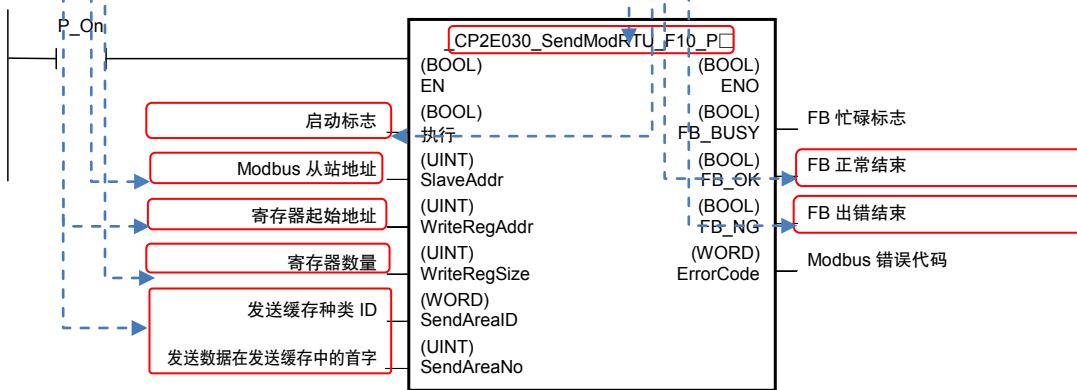
- 当启动标志(W2.00)置 ON 时，功能块从 Modbus 从站地址为 3 的逆变器的寄存器地址为 1001Hex 的寄存器中读取数据，然后将数据写入 D110。(例如，如果要设置输出频率为 60.00 Hz 时，写入&6000。)
- 当启动标志(W2.00)置 ON 时，停止读取。
- 如果功能块出错结束，则错误代码存储在 D123 中。

参考：不同分配 Modbus-RTU 简易主站功能和功能块

■ Modbus-RTU 简易主站功能的存储器分配



■ Modbus RTU 主站功能块示例



- 根据功能代码选择功能块。
- Modbus RTU简易主站功能通信数据数量的单位是字节。在功能块中，它是寄存器的数量(以字为单位)。
- 简易Modbus RTU主站的通信数据中描述的线圈地址/寄存器地址在功能块中用线圈/寄存器(起始)地址设定，数据用写入/读出数据在存储缓存中的首字设定。
- 功能块的执行正常结束标志、执行错误结束标志置 ON 仅持续 1 个周期。
- 功能块启动标志需要保持 ON，直至 FB 正常结束。

■ 实践指南 - 修订记录

修订代码	日期	修订内容
A	2019 年 11 月	初版

功能说明

基本功能	从串行通信端口发送 Modbus RTU 命令。												
符号	<div style="margin-bottom: 20px;"> <p>■ Function01: 读取线圈</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"></td> <td style="width: 40%; border: 1px solid black; padding: 5px;"> _CP2E030_SendModRTU_F01_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 线圈起始地址 ReadCoilAddr (BOOL) FB_OK FB 出错结束 线圈数量 ReadCoilSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo </td> <td style="width: 30%;"></td> </tr> </table> </div> <div style="margin-bottom: 20px;"> <p>■ Function03: 读取保持寄存器</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"></td> <td style="width: 40%; border: 1px solid black; padding: 5px;"> _CP2E030_SendModRTU_F03_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器起始地址 ReadRegAddr (BOOL) FB_OK FB 出错结束 寄存器数量 ReadRegSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo </td> <td style="width: 30%;"></td> </tr> </table> </div> <div style="margin-bottom: 20px;"> <p>■ Function05: 写入单线圈</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"></td> <td style="width: 40%; border: 1px solid black; padding: 5px;"> _CP2E030_SendModRTU_F05_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 输出线圈地址 WriteCoilAddr (BOOL) FB_OK FB 出错结束 输出值 (WORD) (WORD) Modbus 错误代码 OutputValue ErrorCode </td> <td style="width: 30%;"></td> </tr> </table> </div> <div style="margin-bottom: 20px;"> <p>■ Function06: 写入单寄存器</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"></td> <td style="width: 40%; border: 1px solid black; padding: 5px;"> _CP2E030_SendModRTU_F06_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器地址 WriteRegAddr (BOOL) FB_OK FB 出错结束 发送缓存种类 ID SendAreaID (WORD) (WORD) Modbus 错误代码 发送数据在发送缓存中的首字 SendAreaNo (UINT) SendAreaNo </td> <td style="width: 30%;"></td> </tr> </table> </div>		_CP2E030_SendModRTU_F01_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 线圈起始地址 ReadCoilAddr (BOOL) FB_OK FB 出错结束 线圈数量 ReadCoilSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo			_CP2E030_SendModRTU_F03_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器起始地址 ReadRegAddr (BOOL) FB_OK FB 出错结束 寄存器数量 ReadRegSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo			_CP2E030_SendModRTU_F05_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 输出线圈地址 WriteCoilAddr (BOOL) FB_OK FB 出错结束 输出值 (WORD) (WORD) Modbus 错误代码 OutputValue ErrorCode			_CP2E030_SendModRTU_F06_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器地址 WriteRegAddr (BOOL) FB_OK FB 出错结束 发送缓存种类 ID SendAreaID (WORD) (WORD) Modbus 错误代码 发送数据在发送缓存中的首字 SendAreaNo (UINT) SendAreaNo	
	_CP2E030_SendModRTU_F01_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 线圈起始地址 ReadCoilAddr (BOOL) FB_OK FB 出错结束 线圈数量 ReadCoilSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo												
	_CP2E030_SendModRTU_F03_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器起始地址 ReadRegAddr (BOOL) FB_OK FB 出错结束 寄存器数量 ReadRegSize (WORD) Modbus 错误代码 接收缓存种类 ID RecvAreaID 读取数据在接收缓存中的首字 RecvAreaNo												
	_CP2E030_SendModRTU_F05_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 输出线圈地址 WriteCoilAddr (BOOL) FB_OK FB 出错结束 输出值 (WORD) (WORD) Modbus 错误代码 OutputValue ErrorCode												
	_CP2E030_SendModRTU_F06_P□ (BOOL) (BOOL) EN ENO (BOOL) (BOOL) 启动标志 执行 FB_BUSY FB 忙碌标志 Modbus 从站地址 SlaveAddr (BOOL) FB_OK FB 正常结束 寄存器地址 WriteRegAddr (BOOL) FB_OK FB 出错结束 发送缓存种类 ID SendAreaID (WORD) (WORD) Modbus 错误代码 发送数据在发送缓存中的首字 SendAreaNo (UINT) SendAreaNo												

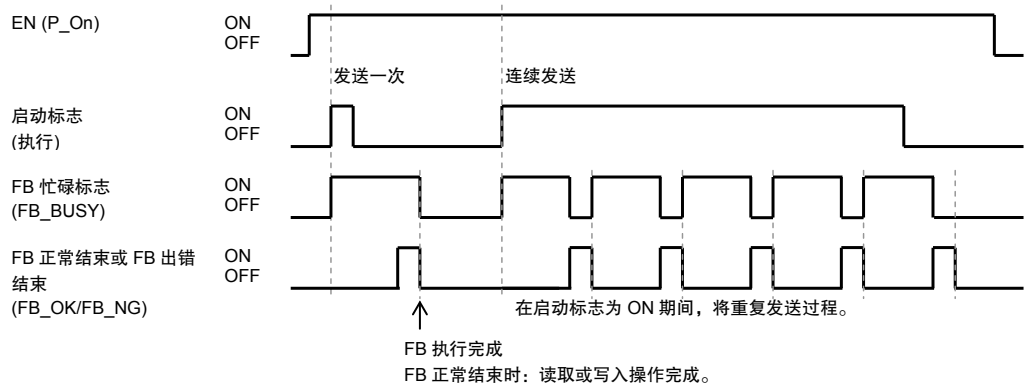
符号	<p>■ Function10: 写入多寄存器</p>																				
文件名称	<p>■ CP2E-S/E 的内置 RS-232C 端口/CP2E-N 的串行端口 1</p> <p>_CP2E030_SendModRtu_F01_P1.cxf: Function 01(读取线圈)</p> <p>_CP2E030_SendModRtu_F03_P1.cxf: Function 03(读取保持寄存器)</p> <p>_CP2E030_SendModRtu_F05_P1.cxf: Function 05(写入单线圈)</p> <p>_CP2E030_SendModRtu_F06_P1.cxf: Function 06(写入单寄存器)</p> <p>_CP2E030_SendModRtu_F10_P1.cxf: Function 10(写入多寄存器)</p> <p>■ CP2E-S 的内置 RS-485/CP2E-N 的串行端口 2</p> <p>_CP2E030_SendModRtu_F01_P2.cxf: Function 01(读取线圈)</p> <p>_CP2E030_SendModRtu_F03_P2.cxf: Function 03(读取保持寄存器)</p> <p>_CP2E030_SendModRtu_F05_P2.cxf: Function 05(写入单线圈)</p> <p>_CP2E030_SendModRtu_F06_P2.cxf: Function 06(写入单寄存器)</p> <p>_CP2E030_SendModRtu_F10_P2.cxf: Function 10(写入多寄存器)</p> <p>■ CP2E-N 串行端口 1(EX)</p> <p>_CP2E030_SendModRtu_F01_P1EX.cxf: Function 01(读取线圈)</p> <p>_CP2E030_SendModRtu_F03_P1EX.cxf: Function 03(读取保持寄存器)</p> <p>_CP2E030_SendModRtu_F05_P1EX.cxf: Function 05(写入单线圈)</p> <p>_CP2E030_SendModRtu_F06_P1EX.cxf: Function 06(写入单寄存器)</p> <p>_CP2E030_SendModRtu_F10_P1EX.cxf: Function 10(写入多寄存器)</p>																				
适用型号	CPU 单元	CP2E																			
	CX-Programmer	9.72 版或更高版本																			
使用条件	<p>■ 设定</p> <ul style="list-style-type: none"> PLC 设置: FB 中的通信指令设定 重试计数(默认值: 0) <p>■ 共享资源</p> <ul style="list-style-type: none"> Modbus RTU 简易主站的 DM 固定区和辅助区 Modbus RTU FB 的工作区 <table border="1" data-bbox="387 1648 1367 1928"> <thead> <tr> <th rowspan="2">套接字号</th> <th colspan="2">Modbus RTU 简易主站</th> <th rowspan="2">工作区</th> </tr> <tr> <th>DM 区</th> <th>AR 区</th> </tr> </thead> <tbody> <tr> <td>内置 RS232C 串行端口 1</td> <td>D1200 ~ D1299</td> <td>A640.00 ~ A640.02</td> <td>W127 D3850 ~ D3885</td> </tr> <tr> <td>内置 RS-485 串行端口 2</td> <td>D1300 ~ D1399</td> <td>A641.00 ~ A641.02</td> <td>W127 D3900 ~ D3935</td> </tr> <tr> <td>串行端口 1 (EX)</td> <td>D1400 ~ D1499</td> <td>A638.00 ~ A638.02</td> <td>W127 D3950 ~ D3985</td> </tr> </tbody> </table>			套接字号	Modbus RTU 简易主站		工作区	DM 区	AR 区	内置 RS232C 串行端口 1	D1200 ~ D1299	A640.00 ~ A640.02	W127 D3850 ~ D3885	内置 RS-485 串行端口 2	D1300 ~ D1399	A641.00 ~ A641.02	W127 D3900 ~ D3935	串行端口 1 (EX)	D1400 ~ D1499	A638.00 ~ A638.02	W127 D3950 ~ D3985
套接字号	Modbus RTU 简易主站		工作区																		
	DM 区	AR 区																			
内置 RS232C 串行端口 1	D1200 ~ D1299	A640.00 ~ A640.02	W127 D3850 ~ D3885																		
内置 RS-485 串行端口 2	D1300 ~ D1399	A641.00 ~ A641.02	W127 D3900 ~ D3935																		
串行端口 1 (EX)	D1400 ~ D1499	A638.00 ~ A638.02	W127 D3950 ~ D3985																		

功能描述

■ 功能块操作

- 当启动标志(执行)上升沿时, 将 Modbus RTU 命令发送至由 Modbus 从站地址(SlaveAddr)指定的 Modbus RTU 从站。
 - 在启动标志(执行)为 ON 期间, 将重复发送过程。
 - 在启动标志(执行)为 OFF 期间, 将停止发送过程。
 - 发送读取命令时, 从 Modbus RTU 从站读取线圈/寄存器起始地址(ReadCoilAddr/ReadRegAddr)和线圈/寄存器数量(ReadCoilSize/ReadRegSize)指定的数据。
 - 发送写入命令时, 将由线圈/寄存器起始地址(WriteCoilAddr/WriteRegAddr)和线圈/寄存器数量(WriteRegSize)指定的数据发送至 Modbus RTU 从站。
 - 使用写入/读取数据区 ID(Send/RecvAreaID)和写入/读取数据的首字(SendAreaNo/RecvAreaNo)指定发送和接收数据字名称。
- 例如, 对于 D1000, 区类型设定为 P_DM, 且首字地址设定为 &1000。
- 如果发送过程正常完成, 则 ON 设定为 FB 正常结束(FB_OK)且"#0000"保存在 Modbus 错误代码(ErrorCode)中。
 - 如果发送过程出错, 则 ON 设定为 FB 出错结束(FB_NG)且 Modbus 错误代码将输出至 Modbus 错误代码(ErrorCode)。

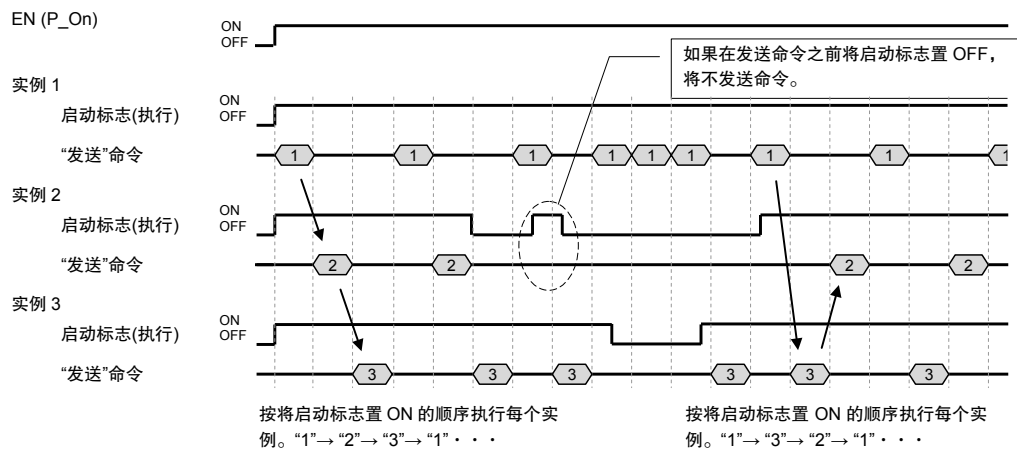
时序图



■ 实例操作

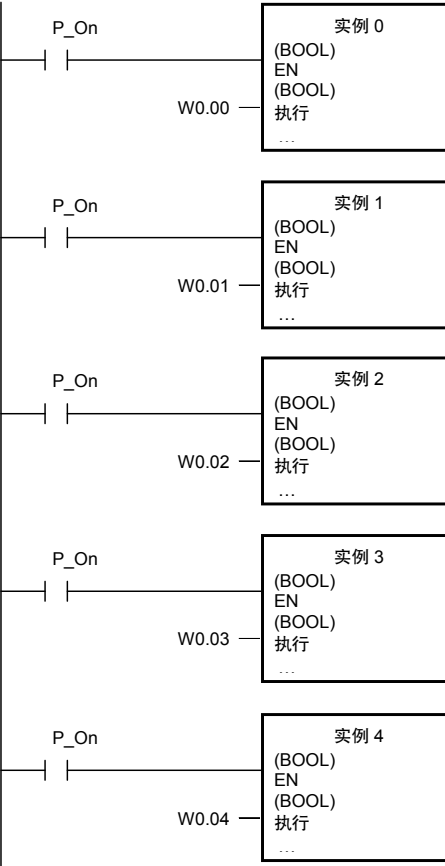
- 可执行串行端口以仅发送一个命令。按将启动标志(执行)置 ON 的顺序执行每个实例。
 - 不执行启动标志(执行)置 OFF 的实例。
- 如果启动标志在发送 Modbus RTU 命令之前置 OFF, 则不发送 Modbus RTU 命令。

时序图: 三个实例(示例)



支持的命令

功能代码	命令名称	功能
01 Hex	读取线圈	读取多个线圈
03 Hex	读取保持寄存器	读取多个保持寄存器
05 Hex	写入单线圈	写入单个线圈
06 Hex	写入单寄存器	写入单个保持寄存器
10 Hex	写入多寄存器	写入多个保持寄存器

FB 定义类型	连接“常 ON”类型 将 EN 输入连接至常 ON 标志(P_On) <ul style="list-style-type: none"> 不能在两个或更多位置使用同一实例。 																																																																								
FB 注意事项	<ul style="list-style-type: none"> 跨多个循环处理 FB。FB 忙碌标志(FB_BUSY)输出变量可用于检查是否正在处理 FB。 每个串行端口上可设定最大 32 个实例。 此 FB 使用 Modbus RTU 简易主站功能。使用此 FB 时，请勿将 Modbus RTU 简易主站功能用于此 FB 使用的串行端口。 使用此 FB 时，请勿使用共享资源的工作区。 																																																																								
EN 输入条件	<ul style="list-style-type: none"> 将 EN 输入连接至常 ON 标志(P_On)。 在 EN 上连接其他(非 P_On)触点，触点 OFF 时，本 FB 输出状态保持不变。 																																																																								
限制输入变量	<ul style="list-style-type: none"> 将常 ON 标志(P_On)用于 EN。 如果启动标志(执行)在发送 Modbus RTU 命令之前置 OFF，则不发送 Modbus RTU 命令。FB 正常结束(FB_OK)置 ON 后，将启动标志(执行)置 OFF。 请勿同时将 EN 和启动标志(执行)置 ON。如果将 EN 和执行同时置 ON，将不执行线性插补。 如果输入变量超出范围，ENO 将置 OFF 且将不处理 FB。 																																																																								
应用示例	<p>将 Modbus RTU 命令发送至 Modbus RTU 从站。 根据 W0.00 ~ W0.04 的 ON/OFF 状态发送。</p>  <table border="1" data-bbox="861 840 1380 1019"> <caption>启动标志</caption> <thead> <tr> <th></th> <th>W0.00</th> <th>W0.01</th> <th>W0.02</th> <th>W0.03</th> <th>W0.04</th> </tr> </thead> <tbody> <tr> <td>实例 0</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>ON</td> </tr> <tr> <td>实例 1</td> <td>ON</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>实例 2</td> <td>ON</td> <td>ON</td> <td>ON</td> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>实例 3</td> <td>ON</td> <td>ON</td> <td>OFF</td> <td>ON</td> <td>OFF</td> </tr> <tr> <td>实例 4</td> <td>ON</td> <td>ON</td> <td>OFF</td> <td>ON</td> <td>OFF</td> </tr> </tbody> </table> <table border="1" data-bbox="861 1064 1380 1254"> <caption>发送命令状态</caption> <thead> <tr> <th>实例</th> <th>实例 0</th> <th>实例 1</th> <th>实例 2</th> <th>实例 3</th> <th>实例 4</th> </tr> </thead> <tbody> <tr> <td>实例 0</td> <td>发送</td> <td>发送</td> <td>发送</td> <td>发送</td> <td>发送</td> </tr> <tr> <td>实例 1</td> <td>发送</td> <td>挂起</td> <td>挂起</td> <td>挂起</td> <td>挂起</td> </tr> <tr> <td>实例 2</td> <td>发送</td> <td>发送</td> <td>发送</td> <td>挂起</td> <td>挂起</td> </tr> <tr> <td>实例 3</td> <td>发送</td> <td>发送</td> <td>挂起</td> <td>发送</td> <td>挂起</td> </tr> <tr> <td>实例 4</td> <td>发送</td> <td>发送</td> <td>挂起</td> <td>发送</td> <td>挂起</td> </tr> </tbody> </table>		W0.00	W0.01	W0.02	W0.03	W0.04	实例 0	ON	ON	ON	ON	ON	实例 1	ON	OFF	OFF	OFF	OFF	实例 2	ON	ON	ON	OFF	OFF	实例 3	ON	ON	OFF	ON	OFF	实例 4	ON	ON	OFF	ON	OFF	实例	实例 0	实例 1	实例 2	实例 3	实例 4	实例 0	发送	发送	发送	发送	发送	实例 1	发送	挂起	挂起	挂起	挂起	实例 2	发送	发送	发送	挂起	挂起	实例 3	发送	发送	挂起	发送	挂起	实例 4	发送	发送	挂起	发送	挂起
	W0.00	W0.01	W0.02	W0.03	W0.04																																																																				
实例 0	ON	ON	ON	ON	ON																																																																				
实例 1	ON	OFF	OFF	OFF	OFF																																																																				
实例 2	ON	ON	ON	OFF	OFF																																																																				
实例 3	ON	ON	OFF	ON	OFF																																																																				
实例 4	ON	ON	OFF	ON	OFF																																																																				
实例	实例 0	实例 1	实例 2	实例 3	实例 4																																																																				
实例 0	发送	发送	发送	发送	发送																																																																				
实例 1	发送	挂起	挂起	挂起	挂起																																																																				
实例 2	发送	发送	发送	挂起	挂起																																																																				
实例 3	发送	发送	挂起	发送	挂起																																																																				
实例 4	发送	发送	挂起	发送	挂起																																																																				
相关手册	CP2E CPU 单元软件用户手册(W614) 14-4 Modbus RTU 简易主站功能																																																																								

■变量表

输入变量

Function01: 读取线圈

名称	变量名称	数据类型	默认值	范围	描述
EN	EN	BOOL			1 (ON): FB 已启动。 0 (OFF): FB 未启动。
启动标志	执行	BOOL			1 (ON): 启动以发送 Modbus 命令 0 (OFF): 停止以发送 Modbus 命令
Modbus 从站地址	SlaveAddr	UINT	&1	&1 ~ &31	
线圈起始地址	ReadCoilAddr	UINT	&0	&0 ~ &65535	
线圈数量	ReadCoilSize	UINT	&1	&1 ~ &16	读取线圈: 最大 16 位
接收缓存种类 ID	RecvAreaID	WORD	#00B1	-	P_WR (#00B1): 工作区
读取数据在接收缓存中的首字	RecvAreaNo	UINT	&0	&0 ~ &126	W0 ~ W126

Function03: 读取保持寄存器

名称	变量名称	数据类型	默认值	范围	描述
EN	EN	BOOL			1 (ON): FB 已启动。 0 (OFF): FB 未启动。
启动标志	执行	BOOL			1 (ON): 启动以发送 Modbus 命令 0 (OFF): 停止以发送 Modbus 命令
Modbus 从站地址	SlaveAddr	UINT	&1	&1 ~ &31	
寄存器起始地址	ReadRegAddr	UINT	&0	&0 ~ &65535	
寄存器数量	ReadRegSize	UINT	&1	&1 ~ &20	读取寄存器: 最大 20 个字
接收缓存种类 ID	RecvAreaID	WORD	#0082	-	P_DM (#0082): 数据存储区
读取数据在接收缓存中的首字	RecvAreaNo	UINT	&0	&0 ~ &3849	D0 ~ D3849 将[读取数据的首字 + 寄存器数量]设定低于范围。 D0- D3849

Function05: 写入单线圈

名称	变量名称	数据类型	默认值	范围	描述
EN	EN	BOOL			1 (ON): FB 已启动。 0 (OFF): FB 未启动。
启动标志	执行	BOOL			1 (ON): 启动以发送 Modbus 命令 0 (OFF): 停止以发送 Modbus 命令
Modbus 从站地址	SlaveAddr	UINT	&1	&1 ~ &31	
输出线圈地址	WriteCoilAddr	UINT	&0	&0 ~ &65535	
输出值	输出值	WORD	#0000		ON: #FF00 OFF: #0000

Function06: 写入单个寄存器

名称	变量名称	数据类型	默认值	范围	描述
EN	EN	BOOL			1 (ON): FB 已启动。 0 (OFF): FB 未启动。
启动标志	执行	BOOL			1 (ON): 启动以发送 Modbus 命令 0 (OFF): 停止以发送 Modbus 命令
Modbus 从站地址	SlaveAddr	UINT	&1	&1 ~ &31	
寄存器地址	WriteRegAddr	UINT	&0	&0 ~ &65535	
发送缓存种类 ID	SendAreaID	WORD	#0082	-	P_DM (#0082): 数据存储区
发送数据在发送缓存中的首字	SendAreaNo	UINT	&0	&0 ~ &3849	D0 ~ D3849

Function10: 写入多寄存器

名称	变量名称	数据类型	默认值	范围	描述
EN	EN	BOOL			1 (ON): FB 已启动。 0 (OFF): FB 未启动。
启动标志	执行	BOOL			1 (ON): 启动以发送 Modbus 命令 0 (OFF): 停止以发送 Modbus 命令
Modbus 从站地址	SlaveAddr	UINT	&1	&1 ~ &31	
寄存器起始地址	WriteCoilAddr	UINT	&0	&0 ~ &65535	
寄存器数量	WriteRegSize	UINT	&1	&1 ~ &20	写入寄存器: 最大 20 个字
发送缓存种类 ID	SendAreaID	WORD	#0082	-	P_DM (#0082): 数据存储区
发送数据在发送缓存中的首字	SendAreaNo	UINT	&0	&0 ~ &3849	D0 ~ D3849 [写入数据的首字 + 寄存器数量]将设定为低于范围。 D0- D3849

输出变量

名称	变量名称	数据类型	范围	描述
ENO	ENO	BOOL		1 (ON): FB 已正常处理。 0 (OFF): FB 未处理或出错结束。
FB 忙碌标志	FB_BUSY	BOOL		处理完成后自动置 OFF。
FB 正常结束	FB_OK	BOOL		处理正常结束时在一个循环内置 ON。
出错结束	FB_NG	BOOL		处理出错结束时在一个循环内置 ON。
错误代码	ErrorCode	DWORD		执行出错结束时, 输出 Modbus 错误代码。 ErrorCode 为 0088 Hex 时, 异常代码将被设为 ErrorCode+1。

■Modbus RTU 错误代码

错误代码	目录	进程
0000 Hex	正常结束	-
0004 Hex	串行通信模式错误	当串行通信模式不是 Modbus-RTU 简易主站时, 执行 Modbus-RTU 简易主站功能模式
0084 Hex	CRC 错误	出现 CRC 错误
0085 Hex	响应地址不正确	响应从站地址与发送命令不匹配
0086 Hex	响应功能代码不正确	响应功能代码与发送命令不匹配
0087 Hex	响应大小超限	响应数据大小较大
0088 Hex	例外响应	来自 Modbus 从站的例外响应 异常代码将被设为 ErrorCode+1。
008F Hex	其他错误	出现其他错误。
FFFD Hex	实例超限	使用的实例超过 32
FFFF Hex	超出变量输入范围	超出变量输入范围

Modbus 命令详细信息

■Function 01 读取线圈：从多个线圈读取到工作区(W)。

请求

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	01 Hex
线圈起始地址	2 个字节	ReadCoilAddr 0 ~ FFFF Hex (&0 ~ &65535)
线圈数量	2 个字节	ReadCoilSize 0 ~ 10 Hex (&0 ~ &16)

响应

字段名称	数据长度	数据或变量输入名称
从站地址	1 字节	与 SlaveAddr 相同
功能代码	1 字节	01 Hex
字节计数	1 字节	1 或 2 Hex
线圈状态	1 或 2 字节	通过读取数据区(RecvAreaID)、读取数据字(RecvAreaNo)指定的工作区(W)。 示例：读取数据区：P_WR 读取数据字：&10 读取到 W10CH

*：未在 FB 上确认线圈状态除外的响应数据。

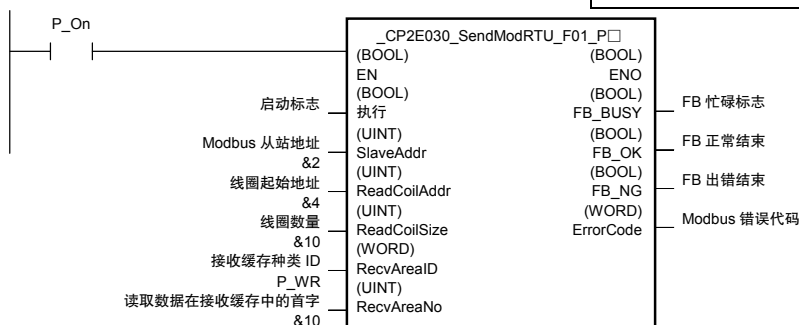
示例：从从站地址 2 上的线圈起始地址 4(0004 Hex)将 10 位(10 10001010 =28A Hex)读取到 W10。

请求(CP2E: Modbus 主站)

字段名称	数据
从站地址	02 Hex
功能代码	01 Hex
线圈起始地址(高)	0004 Hex
线圈起始地址(低)	(自&4)
线圈数量(高)	000A Hex
线圈数量(低)	(10 位)

响应(Modbus 从站)

字段名称	数据
从站地址	02 Hex
功能代码	01 Hex
字节计数	02 Hex
线圈状态 4 ~ 11	8A Hex W10.08 ~ W10.15
线圈状态 12 ~ 13	02 Hex W10.00 ~ W10.01



Modbus 从站 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		(1)	(0)	(1)	(0)	(0)	(0)	(1)	(0)	(1)	(0)				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

*：阴影框中的下标编号表示线圈地址和 ON/OFF (1/0)状态。



CP2E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W9CH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
W10CH	(1)	(0)	(0)	(0)	(1)	(0)	(1)	(0)	-	-	-	-	-	-	(1)	(0)
W11CH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*：阴影框中的下标编号表示 ON/OFF (1/0)状态。

■Function 03 读取保持寄存器：从多个保持寄存器读取到数据存储区(D)。

请求

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	03 Hex
寄存器起始地址	2 个字节	ReadRegAddr 0 ~ FFFF Hex (&0 ~ &65535)
寄存器数量	2 个字节	ReadRegSize 1 ~ 0014 Hex (&1 ~ &20)

响应

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	与 SlaveAddr 相同
功能代码	1 字节	01 Hex
字节计数	1 字节	2×N(N: 寄存器数量)
读取寄存器数据	2×N 字节	通过读取数据区(RecvAreaID)、读取数据字(RecvAreaNo)指定的数据存储区(D)。 示例：读取数据区：P_DM 读取数据字：&1000 从 D1000 读取

* : 未在 FB 上确认读取寄存器数据状态除外的响应数据。

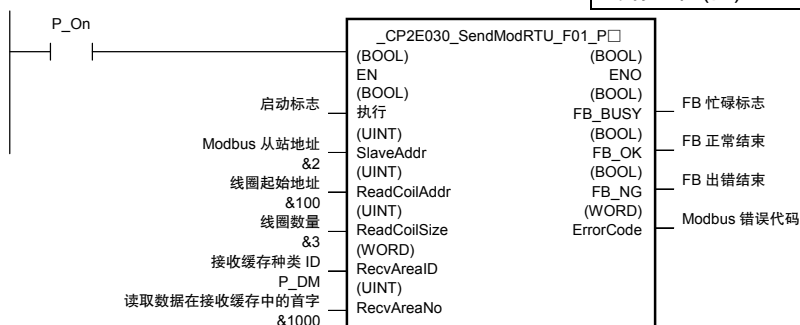
示例：从从站地址 2 上的寄存器起始地址 100 (0064 Hex)将 3 个字(AB12、5678、9713 Hex)读取到 D1000、D1001 和 D1002。

请求(CP2E: Modbus 主站)

字段名称	数据
从站地址	02 Hex
功能代码	03 Hex
寄存器起始地址(高)	0064 Hex
寄存器起始地址(低)	(自&100)
寄存器数量(高)	0003 Hex
寄存器数量(低)	(3 字)

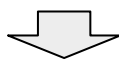
响应(Modbus 从站)

字段名称	数据
从站地址	02 Hex
功能代码	03 Hex
字节计数	06 Hex
寄存器值(高)	AB12 Hex
寄存器值(低)	(D1000)
寄存器值(高)	5678 Hex
寄存器值(低)	(D1001)
寄存器值(高)	9713 Hex
寄存器值(低)	(D1002)



Modbus 从站 2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100	A				B				1				2			
101	5				6				7				8			
102	9				7				1				3			



读取

CP2E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D1000	A				B				1				2			
D1001	5				6				7				8			
D1002	9				7				1				3			

■Function 05 写入单线圈：写入单线圈

请求

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	05 Hex
输出线圈地址	2 个字节	WriteCoilAddr 0 ~ FFFF Hex (0 ~ &65535)
输出值	2 个字节	ON(1): #FF00 OFF(0): #0000

响应

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	与 SlaveAddr 相同
功能代码	1 字节	05 Hex
输出线圈地址	2 个字节	WriteCoilAddr 0 ~ FFFF Hex (0 ~ &65535)
输出值	2 个字节	ON(1): FF00 Hex OFF(0): 0000 Hex

* : 未在 FB 上确认线圈状态除外的响应数据。

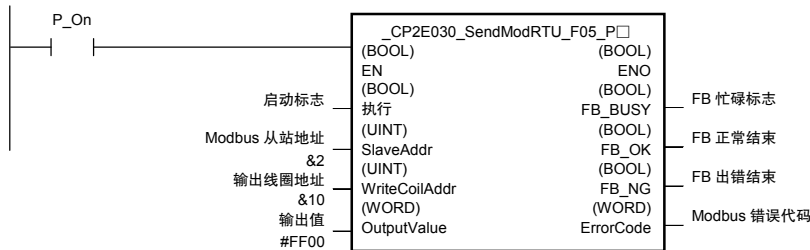
示例：在从站地址 2 上设定输出线圈地址 10 的线圈。

请求(CP2E: Modbus 主站)

字段名称	数据
从站地址	02 Hex
功能代码	05 Hex
输出线圈地址(高)	000A Hex
输出线圈地址(低)	(&10)
输出值	FF00 Hex (ON)

响应(Modbus 从站)

字段名称	数据
从站地址	02 Hex
功能代码	05 Hex
输出线圈地址(高)	000A Hex
输出线圈地址(低)	(&10)
输出值	FF00 Hex (ON)

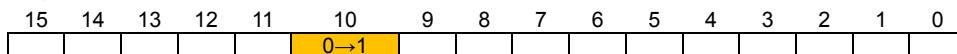


CP2E

输出线圈地址：&10，写入数据： FF00 Hex (ON)



Modbus 从站 2



■Function 06 写入单寄存器：从数据存储区(D)写入到单个保持寄存器

请求

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	06 Hex
寄存器地址	2 个字节	WriteRegAddr 0 ~ FFFF Hex (0 ~ &65535)
寄存器值	2 个字节	通过读取数据区(RecvAreaID)、读取数据字(RecvAreaNo)指定的数据存储区(D)。 示例：读取数据区：P_DM 读取数据字：&500 写入到 D500

响应

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	与 SlaveAddr 相同
功能代码	1 字节	06 Hex
寄存器地址	2 个字节	与 WriteRegAddr 相同 0 ~ FFFF Hex (0 ~ &65535)
寄存器值	2 个字节	0000 ~ FFFF Hex

*：未在 FB 上确认读取寄存器数据状态除外的响应数据。

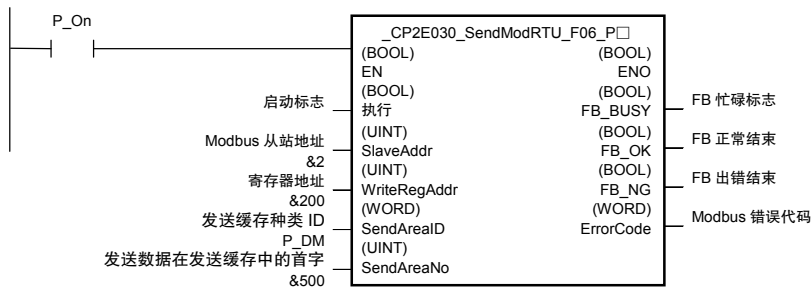
示例：从从站地址 2 上的寄存器地址 200 将 1 个字(3AC5 Hex)写入到 D500。

请求(CP2E: Modbus 主站)

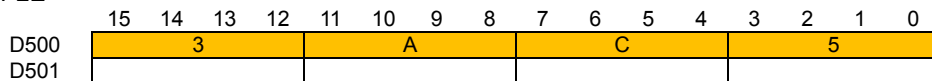
字段名称	数据
从站地址	02 Hex
功能代码	06 Hex
寄存器地址(高)	00C8 Hex
寄存器地址(低)	(200)
寄存器数量(高)	3AC5 Hex
寄存器数量(低)	(D500)

响应(Modbus 从站)

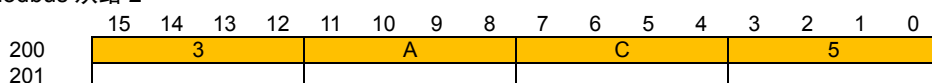
字段名称	数据
从站地址	02 Hex
功能代码	06 Hex
寄存器地址(高)	00C8 Hex
寄存器地址(低)	(200)
寄存器数量(高)	3AC5 Hex
寄存器数量(低)	



CP2E



Modbus 从站 2



■Function 10 写入多寄存器：从数据存储区(D)写入到多个保持寄存器

请求

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	10 Hex
寄存器起始地址	2 个字节	WriteRegAddr 0 ~ FFFF Hex (0 ~ &65535)
寄存器数量	2 个字节	WriteRegSize 0 ~ 0014 Hex (0 ~ &20)
字节计数	1 字节	2×N(N: 寄存器数量)
寄存器值	2×字节计数字节	通过读取数据区(RecvAreaID)、读取数据字(RecvAreaNo)指定的数据存储区(D)。 示例：读取数据区：P_DM 读取数据字：&2000 从 D1000 读取

响应

字段名称	数据长度	数据或输入变量名称
从站地址	1 字节	SlaveAddr
功能代码	1 字节	10 Hex
寄存器起始地址	2 个字节	与 WriteRegAddr 相同
寄存器数量	2 个字节	与 WriteRegSize 相同

* : 未在 FB 上确认读取寄存器数据状态除外的响应数据。

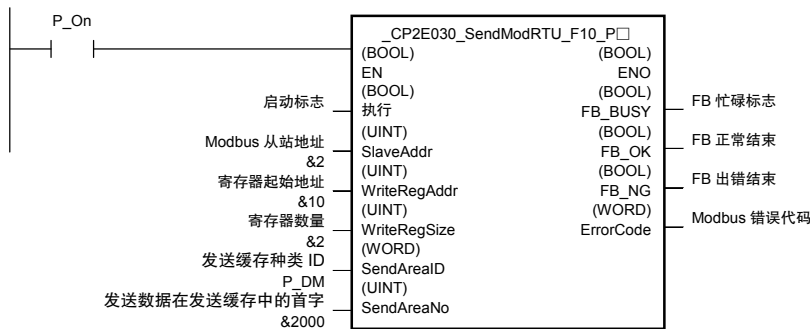
示例：从从站地址 2 上的 D2000,D2002 (0100, 3A98 Hex)写入寄存器起始地址 10 (000A Hex)的 2 个字。

请求(CP2E: Modbus 主站)

字段名称	数据
从站地址	02 Hex
功能代码	10 Hex
寄存器起始地址(高)	000A Hex
寄存器起始地址(低)	(自 10)
寄存器数量(高)	02 Hex
寄存器数量(低)	(2 字)
字节计数	04 Hex
寄存器值 1(高)	0100 Hex
寄存器值 1(低)	(D2000)
寄存器值 2(高)	3A98 Hex
寄存器值 2(低)	(D2001)

响应(Modbus 从站)

字段名称	数据
从站地址	02 Hex
功能代码	10 Hex
寄存器起始地址(高)	000A Hex
寄存器起始地址(低)	
寄存器数量(高)	02 Hex
寄存器数量(低)	(2 字)



CP2E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D2000	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
D2001	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	4



Modbus 从站 2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	4

■ 版本历史

版本	日期	目录
001	2019.11	新产品

注

本手册是一本介绍功能块功能的参考手册。

并未介绍单元、组件、或单元和组件组合的操作限制。在使用之前，请务必阅读并了解系统单元和其他组件的操作手册。