# 1．S7-1200 与 RF200 的产品介绍

## 1．1 S7-1200 的产品介绍

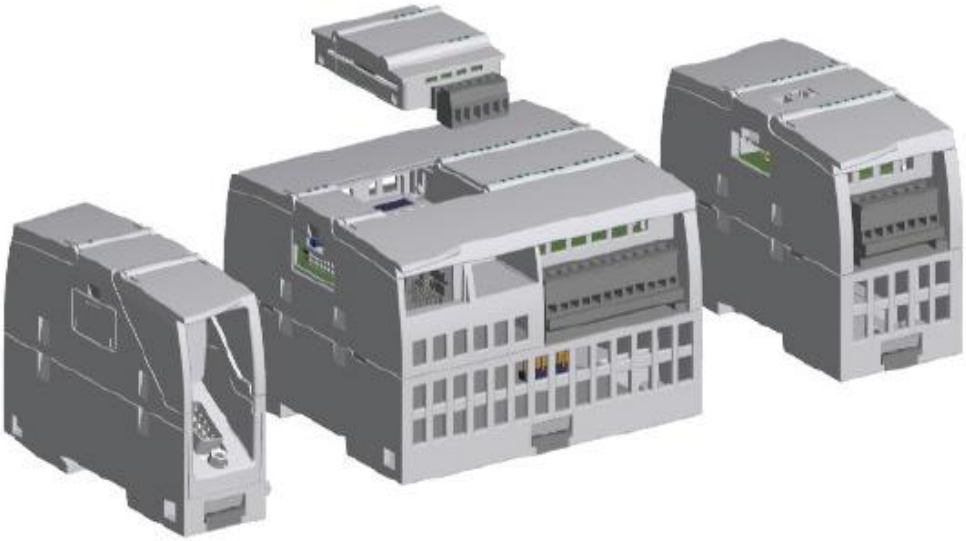


图 1-1：S7-1200 PLC

**S7-1200 概览**

- SIMATIC S7 系列的新型模块化微型控制器包括：
- 控制器配备集成的 PROFINET IO 控制器接口，用于 SIMATIC 控制器、人机界面、编程设备或其他自动化组件之间的通讯
- 带 PROFIBUS DP 主站接口的通讯模板
- 通讯模块 PROFIBUS DP 从站接口
- 用于连接到 GSM/G 移动电话网络的 GPRS 模块
- 集成的 web 服务器，带有标准和用户特定的网页
- 数据记录功能，用于在运行时从用户程序归档数据
- 强大的集成技术功能，如计数、测量、闭环控制和运动控制
- 集成数字量和模拟量 I/O
- 直接在控制器中使用的信号板
- 用于通过输入/输出通道扩展控制器的信号模块
- 用于通过附加通讯接口扩展控制器的通讯模块
- 附件，例如，电源、开关模块或 SIMATIC 存储卡

- 以最低的成本提供最大自动化性能的微型控制器。
- 安装、编程和操作都非常简单。
- 大规模集成，节省空间，功能强大。
- 适用于小型到中型的自动化工程应用。
- 既可用于简单的控制任务，也可以用于复杂的自动化任务。
- 所有 CPU 都可在独立模式、网络中和分布式结构内使用。
- 适用于可编程控制器过去在经济上不可行的应用。
- 具有出色的实时性能和功能强大的通讯选件

**S7-1200 应用范围**

SIMATIC S7-1200 是一款适用于机械设备制造以及工厂构建中开环和闭环控制任务的控制器。它可实现最大的自动化性能以及最为经济的成本。

由于同时采用了高性能的紧凑型模块化设计，SIMATIC S7-1200 适用于各种自动化应用。其使用范围从更换继电器和接触器一直到网络和分布式结构中复杂的自动化任务。

S7-1200 还不断开辟了以前由于经济原因而开发的特殊电子元件领域。

应用示例包括，例如：

- 布局系统
- 输送系统
- 电梯和自动扶梯
- 材料运输设备
- 金属加工机械
- 包装机械
- 印刷机器
- 纺织机械
- 混合系统
- 淡水处理厂
- 污水处理厂
- 外部显示屏
- 配电站
- 室温控制
- 加热/冷却系统控制

- 能量管理
- 消防系统
- 空调系统
- 照明控制
- 泵的控制
- 安防/门禁系统

**S7-1200 的设计**

SIMATIC S7-1200 系列由以下模块组成：

3 个控制器，具有不同类型的分级性能，可作为宽范围的交流或直流控制器

2 个信号板（模拟和数字），用于直接在 CPU 上进行的低成本模块化控制器扩展，安装空间保持不变

13 个不同的数字和模拟信号模块

2 个通讯模块 (RS232/RS485)，用于通过点对点连接进行的通讯

带有 4 个端口的以太网开关，用于执行许多不同的网络拓扑

PS 1207 稳定电源装置，115/230 V 交流线路电压，24 V 直流额定电压

**机械特点**

坚固耐用的紧凑型塑料外壳

易于接触的连接和控制元件，用前挡板保护

可拆卸连接端子，也适用于模拟或数字扩展模板

设备功能

符合国际标准：

SIMATIC S7-1200 符合 VDE、UL,、CSA 和 FM 等标准（Class I、Cat 2；危险区组别 A、B、C 和 D，T4A）。生产中采用的质量管理体系已通过 ISO 9001 认证

通讯

SIMATIC S7-1200 配备不同的通讯机制：

集成的 PROFINET 接口

通过通讯模板实现点对点连接

PROFINET 接口

集成的 PROFINET 接口允许进行下列通讯：

编程器

人机接口（HMI）设备

其他 SIMATIC 控制器

支持以下协议：

TCP/IP

ISO-on-TCP

S7 通讯

可以连接以下设备：

通过标准 CAT5 电缆在现场 PG 编程设备和 PC 之间通讯。

PG 和 SIMATIC S7-1200 的 CPU 之间的连接

SIMATIC HMI Basic 面板

基本面板和 SIMATIC S7-1200 的 CPU 之间的连接

更多的 SIMATIC S7-1200 控制器

通过 CSM 1277 以太网交换机连接多台设备

点到点接口，可自由编程接口模式

通讯模块允许通过点对点连接进行的通讯。使用 RS232 和 RS485 物理传输介质。在 CPU 的“自由端口 (Freeport)”模式中执行数据传输。使用用户专用、面向比特的通讯协议（例如，ASCII 协议、USS 或 MODBUS）。

可以连接带有串行接口的任何终端设备，例如，驱动器、打印机、条形码阅读器、调制解调器等。

在可编程的接口模式中，通过 CM 1241 实现的点对点连接

**功能：**

S7-1200 具有以下特点：

极为简单的入门解决方案：

专用的入门软件包和操作说明方便熟悉操作。

操作简单：

功能强大的标准命令，使用简单，结合易于使用的编程软件，可以将编程费用最小化。

出色的实时特性：

特殊的中断功能、快速计数器和脉冲输出适用于时间紧急的过程。

SIMATIC S7-1200 满足国内和国际标准：

UL 508

CSA C22.2 No. 142

FM Class I, div. 2, group A, B, C, D; T4A Class I, Zone 2, IIC, T4

VDE 0160

EN 61131-2

EMC 指令的要求符合 EN 50081-1、50081-2 和 50082-2

**技术特点**

下表是 S7-1200 的通用技术特点:

| 通用技术规范 | |
|---|---|
| 防护等级 | IP20，符合 IEC 529 |
| 环境温度 | |
| 操作<br>（95% 的湿度） | |
| 水平安装 | 0 ... 55 °C |
| 垂直安装 | 0 ... 45 °C |
| 运输/储存 | -40 ... +70 °C |
| 95% 的湿度 | 25 ... 55 °C |
| 绝缘 | |
| 5/24 V DC 电路 | 500 V 交流测试电压 |
| 115/230 V AC 电路接地 | 1500 V 交流测试电压 |
| 115/230 V AC 电路到 115/230 V AC 电路 | 1500 V 交流测试电压 |
| 230 V AC 电路到 5/24 V DC 电路 | 1500 V 交流测试电压 |
| 115 V AC 电路到 5/24 V DC 电路 | 1500 V 交流测试电压 |
| 电磁兼容性 | EMC 指示要求 |
| 抗干扰度符合<br>EN 50082-2 | 测试符合<br>IEC 801-2、IEC 801-3、IEC 801-4、 |

| | |
|---|---|
| | EN 50141、EN 50204、IEC 801-5 和 VDE 0160 进行测试 |
| 辐射干扰符合 EN 50081-1 和 EN 50081-2 | 测试符合<br>EN 55011 类 A，<br>第 1 组 |
| 机械强度 | |
| 振动，测试标准符合/测试方法 | IEC 68, Part 2-6:<br>10 ... 57 Hz;<br>平稳状态下<br>0.3 mm;<br>58 ... 150 Hz;<br>恒定加速度 1 g（安装在 DIN 导轨上）或<br>2 g（安装在配电盘中）；<br>振动模式：<br>振动频率为 1 频程/分钟；<br>振动持续时间：<br>在三个互相垂直轴的每个方向上，每根轴为 10 个频率扫描。 |
| 冲击，测试标准符合/测试方法 | IEC 68, Part 2-27/半正弦：<br>冲击强度 15 g（峰值），持续时间 11 ms，每三个互相垂直轴上有 6 个冲击。 |

表 1-1：S7-1200 通用技术特点

## 1．2 RF200 识别系统的介绍

RF200 识别系统的读写器及数据载体如下图所示：

图 1-2：RF200 识别系统的读写器及数据载体如图

RFID 系统 SIMATIC RF200 是一种紧凑型低成本读写装置，尤其适合在工业生产中的小型装配线和内部物流系统中使用。

通过 RF200，可极为经济有效地实现 HF 范围（13.56 MHz，ISO 15693）内的中等性能识别任务。RF200 读写装置可与 MOBY D 产品系列 (MDS Dxxx) 中的所有 ISO 数据载体一起使用。

可以使用可用于所有 MOBY 和 SIMATIC RF 系统（ASM 456、ASM 475、SIMATIC RF1xxC）的通讯模块来连接到 SIMATIC S7-300、PROFIBUS、PROFINET 和 TCP/IP (XML)。

**SIMATIC RF200 识别系统具有以下功能：**

- 13.56 MHz 工作频率（工作原理符合 ISO 15693）。
- 无源（不带电池）、免维护的数据载体 (MDS Dxxx)，存储容量高达 2000 字节 FRAM。
- 坚固耐用的紧凑部件，防护等级为 IP67。
- 借助于经过反复验证的函数块（FC44、FC45、FB45），可简便集成到 SIMATIC、PROFIBUS、PROFINET 和 TCP/IP 中。

**RF200 识别系统的优点：**

- 价格合理、节省空间的紧凑部件。
- 可与 MOBY D 系列中价格具有吸引力且不带电池的 ISO 15693 数据载体一起使用，投资和运行成本都较低。
- 拥有工业识别领域内完整和可扩展的产品系列，可实现灵活和经济的解决方案。
- 通过无缝集成到全集成自动化系统中而简化组态、调试、诊断和维护：
- 通过 PROFIBUS 和 PROFINET 通讯模块与自动化系统（如 SIMATIC、SIMOTION 或 SINUMERIK）进行集成总线连接
- 通过随时可用的函数块进行简便的 S7 软件集成

**因以下方面而具有较高投资安全性：**

- 开放式 ISO 15693 标准
- 西门子 RFID 系统之间具有软件兼容性
- 采用标准化的通讯接口。
- 通过各种通讯模块，可连接到不同厂商的不同总线系统以及不同 PC 环境，因而具有开放性。
- 全球服务与支持

**RF200 识别系统的应用**

RFID 系统 SIMATIC RF200 主要用于对容器、托盘和工件托架等进行非接触式识别，其性能（数据传输速率、存储器容量）符合 ISO 15693 标准。

SIMATIC RF200 的主要应用领域：

- 组装和搬运系统、装配线（识别工件载体），尤其是小型线。
- 生产物流（物料流控制、容器识别），内部物流
- 零件识别（例如，将数据载体安装到产品或托盘上）。
- 输送系统（例如，悬挂式单轨输送系统）。

**集成应用**

丰富的通讯模块、函数块和功能强大的驱动程序和功能库，使得在应用程序中的集成快捷、简便。

而且最棒的是：SIMATIC RF200 是全集成自动化的一部分，可非常容易地经济地融入 SIMATIC 环境中。

**技术特性：**

下表是 RF200 的通用技术特点:

| RFID 系统 | SIMATIC RF200 |
|---|---|
| 传输频率 | 13.56 MHz |
| 范围 | 最大 130 mm |
| 协议（空中接口） | ISO 15693<br>ISO 18000-3 |
| 存储器容量 | 最大 992 字节 (EEPROM) / 2000 字节 (FRAM) |
| 阅读器与数据载体之间的数据传输速率 | |

| RFID 系统 | SIMATIC RF200 |
| --- | --- |
| 读 | 最大 1.5 KB/s |
| 写 | 最大 0.5 KB/s |
| 多标签/批量处理能力 | X |
| 特点 | 设计极为紧凑<br>用于成本极低的 RFID 解决方案 |

表 1-2：RF200 通用技术特点

# 2 S7-1200 与 RF260R 通信的基本原理

## 2.1 S7-1200 与 RF260R 设备的基本连接

下图是 S7-1200 与 RF260R 设备的基本连接

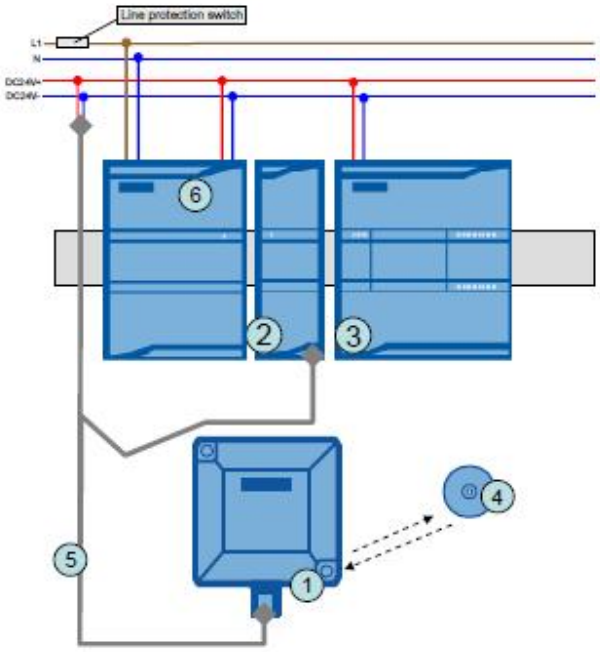



图 2-1： S7-1200 与 RF260R 设备的基本连接

1. RF260R 读写器
2. RS232 模块
3. S7-1200 PLC
4. MDSDXX 数据载体
5. 连接电源和通信电缆
6. 电源模块

S7-1200 PLC 通过 RS232 模块与 RF260R 读写器连接起来，使用的连接电缆是 RS232 电缆，如果是其他型号的 RF200 或 RF300R，可以使用 RS232 转 RS422 转换器来进行连接。RF260R 读写器通过 S7-1200 的专用软件库和 MDSDXX 数据载体进行读写通信。

## 2.2 S7-1200 与 RF260R 通信的原理

S7-1200 PLC 使用专用的软件库通过 RS232 模块和 RF200 或 RF300R 进行通信，其中使用的协议是 3964R 协议。

### 2.2.1 读取数据

RF260R 读取数据不是直接写到 S7-1200 控制器中，而是这些数据存储在 RS232 模块的内部缓冲区中。首先 S7-1200 被通知需要接收数据，然后 S7-1200 把需要的数据从 RS232 模块的缓冲区中取走。如下图所示：

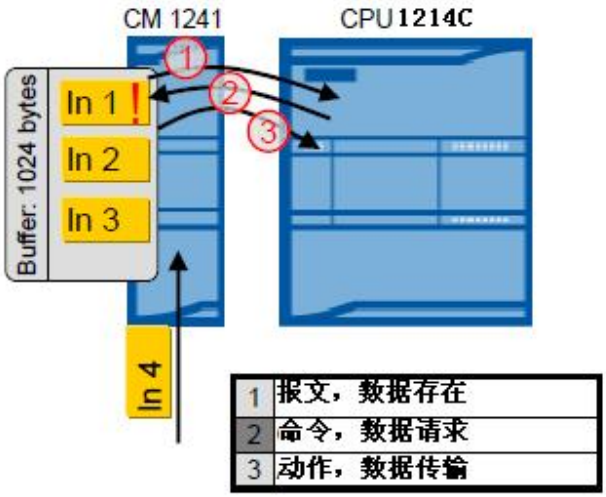



图 2-2：S7-1200 从 RF200 读取数据

### 2.2.2 写入数据

写入数据的机制，是写入数据尽可能快速地独立的发送，这样如果在 CPU 的一个扫描周期中，RS232 模块的缓冲区中有多个数据时，那么这些数据将同时被发送。如下图所示：

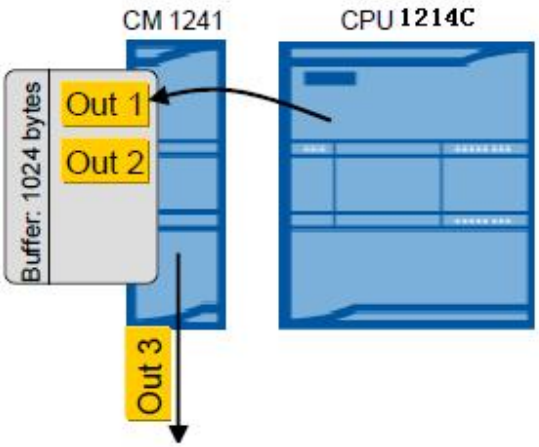



图 2-3：S7-1200 写入数据到数据载体

### 2.2.3 3964R 协议

**3964R 协议在 OSI 第一层上的执行：**

3964R 协议传输数据是全双工模式，因此在双方向上独立传输。

S7-1200 串口通信模块支持全双工模式，发送和接收数据独立进行。

对于 CPU 扫描来说，发送和接收机制是异步和独立传输的。

**3964R 协议在 OSI 第二层上的执行：**

3964R 协议是在两个主站（主主协议）之间进行通信的点到点传输协议，这两个节点都可以进行主动数据发送。

基于 S7-1200 串口通信模块的全双工通信模式，与 RF260R 的通信方式可能与 S7-200 的通信方式是相反的，包括：

发送字符或数据到 RF260R。

检测或接收 RF260R 的字符或数据。

**3964R 协议传输**

该协议定义了发送和接收数据过程的控制字符和以及 RFID 报文，3964R 协议对于 RF260R 到 S7-1200 控制器的响应报文使用奇校验方式。如下图所示：

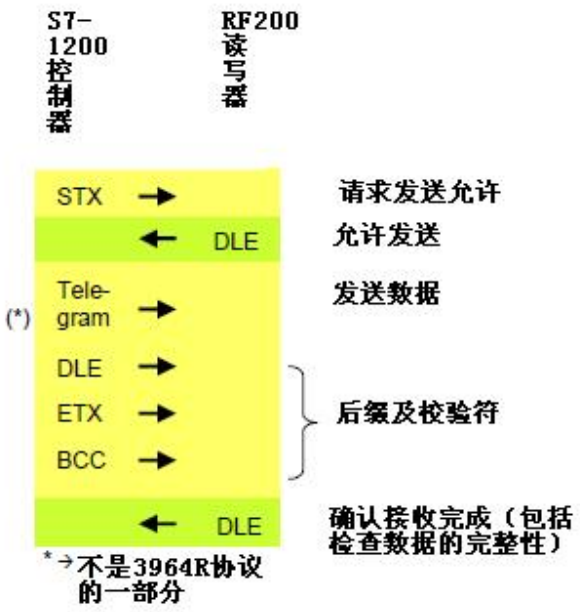



图 2-4：3964R 协议报文传输

3964R 协议在 OSI 第三层或第四层上的执行（RFID 报文），如下表所示：

| 序号 | 命令 | 描述 |
|---|---|---|
| 1 | 复位 | 此命令复位 RF200 读写器，删除所有的没有执行的命令，同时执行参数的传输，响应报文包含一个启动报文。 |
| 2 | 读 | 该命令请求从当前区域存在的 RFID 数据载体中读取定义的数据，响应报文包含请求的数据。 |
| 3 | 写 | 该命令写数据到当前区域存在的 RFID 数据载体中，响应报文包括一个确认报文。 |

表 2-1：3964R 协议报文传输

# 3 S7-1200 与 RF260R 通信的软件库

S7-1200 使用专用的软件库通过 RS232 通信模块和 RF200 进行数据的读取和写入, 读取数据的命令以及写入数据的命令在软件库中都已经做了定义，使用命令时需要对应写入的数据区域和读取的数据区域。

## 3.1 S7-1200 软件库

在 S7-1200 和 RFID 读写器之间的通信使用的专用软件库是执行 3964R 协议的基础，RF260R 或 RF380R 能够直接连接是因为有 RS232 接口，其他的 RF200，RF300R，RF600R 也能够使用这个软件库来进行读写的操作，但是必须有一个 RS422 到 RS232 的转换器。

一个 S7-1200 PLC 最多可以连接 3 个 RFID 读写器，因为一个 S7-1200PLC 最多可以连接 3 个 RS232 通信模块。

### 3.1.1 软件库的程序块

软件库的结构及描述如下表所示：

| 软件库 | 类型 | 文件夹 | 程序块 | 块号码 |
|---|---|---|---|---|
| rfid_serial | Types | | | |
| | Master copies | rfid_serial_com | Function block: com_serial | [FB 164] |
| | | | Instance Data block: own name | [DB own number] |
| | | | Global Data block: rfid_serial_read_A | [DB 165] |
| | | | Global Data block: rfid_serial_read_B | [DB 166] |
| | | | Global Data block: rfid_serial_read_C | [DB 167] |
| | | | Global Data block: rfid_serial_write_A | [DB 168] |
| | | | Global Data block: rfid_serial_write_B | [DB 169] |
| | | | Global Data block rfid_serial_write_C | [DB 170] |
| | | rfid_serial_chart | chart_cmd-return_A | - |
| | | | chart_cmd-return_B | - |
| | | | chart_cmd-return_C | - |
| | | | chart_rs232blocks_A | - |
| | | | chart_rs232blocks_B | - |
| | | | chart_rs232blocks_C | - |
| | | | chart_data_read_A | - |
| | | | chart_data_read_B | - |
| | | | chart_data_read_C | - |

表 3-1：软件库的结构及描述

为了和每一个 RFID 读写器进行数据交换，功能块“com_serial”必须被循环调用，因此对于每一次调用使用的背景数据块建议使用名字如“com_serial_DB_X”,这样可以方便的监控每一个读写器进行数据通信时的状态。

全局数据块“rfid_serial_Read_X”和“rfid_serial_write_X”是 S7-1200 程序和 RFID 读写器的数据载体之间读取和写入数据的的用户接口。

“chart_cmd_return_X ”监控表允许对 “com_serial” 功能块直接访问输入和输出参数。

使用“chart_rs232blocks_X”监控表可以对点到点通信模块 RS232 模块进行监控。

对 RFID 标签直接读取数据或者写入数据，在监控表“chart_data_read_X”和“chart_data_write_X”中可以进行。

注意：X 表示通道 A，B，C，一个 RS232 模块是一个通道，每个通道连接一个 RFID 读写器，一个 S7-1200 PLC 最多可以连接 3 个 RS232 模块。

3.1.2　软件库的资源

RFID 软件程序块在存储器中的大小，一个通道占用大约 9.86K 字节，两个通道占用大约 11.385K 字节，三个通道占用大约 12.829K 字节。

可以使用的 PLC 的剩余的存储量的大小依赖于所使用的 S7-1200 PLC 的类型，如下图所示：

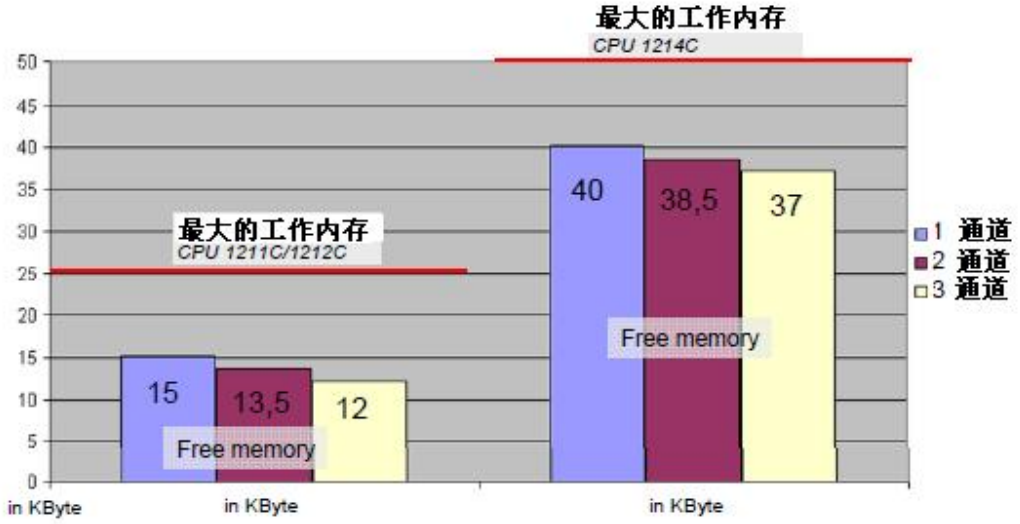



图 3-1：软件库所占用的 PLC 的内存

3.1.3　软件库的功能块“com_serial”FB164

软件库的功能块“com_serial　”FB164 如下图所示：

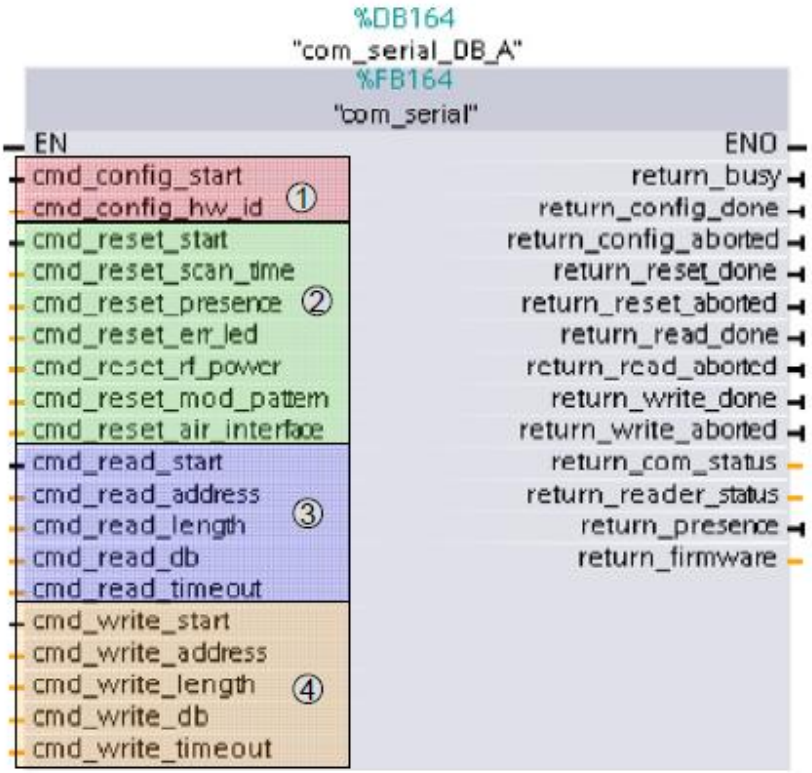



图 3-2：软件库的功能块 FB164

每个参数组的描述如下表所示：

| 参数组 | 名称 | 描述 |
|---|---|---|
| ① | Config | S7-1200 和 RFID 读写器进行通信的参数配置 |
| ② | Reset | 对 RFID 读写器进行复位 |
| ③ | Read | 从数据载体读取数据 |
| ④ | Write | 写数据到数据载体上 |

表 3-2：软件库的功能块 FB164 每个参数组的描述

①.S7-1200 和 RFID 读写器进行通信的参数配置

启动过程如下图所示：

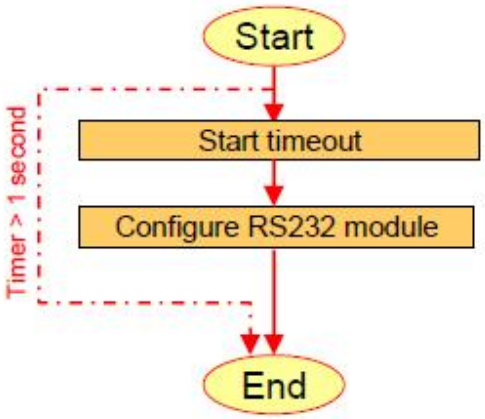



图 3-3：S7-1200 和 RFID 读写器进行通信的启动过程

设置 RS232 模块的参数如下所示：

1.传输速率　115.2k bit/s

2.数据位　　8 位

3.奇偶校验　奇校验

4.停止位　　1 位

②. 对 RFID 读写器进行复位

对 RFID 读写器的复位过程如下：

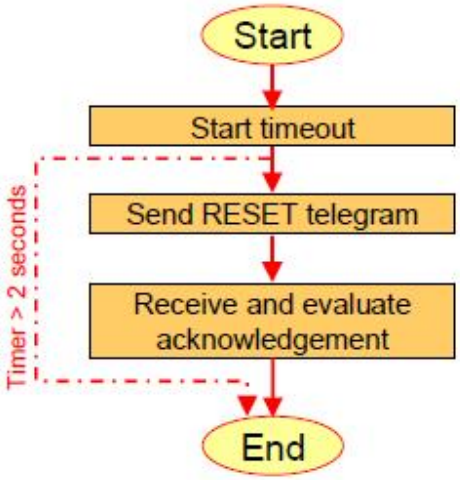



图 3-4：S7-1200 和 RFID 读写器进行通信的复位过程

在所有别的报文之前，必须要发送复位报文。如有任何错误，可以在任何时候发送复位报文。发送复位报文：

复位命令配置 RFID 读写器，传输的参数决定了设备的系统模式，下述的参数可以改变，如下表所示：

| 序号 | 参数 | 描述 |
|---|---|---|
| 1 | Cmd_reset_scam_time | 只对 RF600 读写器，使用该参数设置无线电的模式 |
| 2 | Cmd_reset_presence | 对 RFID 所有读写器,检测 RFID 数据载体是否离开 RFID 读写器时，发送到场报文 |
| 3 | Cmd_reset_err_led | 只对 RF300 读写器,RF 读写器发生错误时，LED 指示灯指示错误的类型，LED 灯可以通过复位报文来复位。 |
| 4 | Cmd_reset_rf_power | 只对 RF380 读写器,RF380 的发送功率能够被改变。 |
| 5 | Cmd_reset_mod_pattern | 只对 RF600 读写器,选择 RF600 调制器的模式。 |
| 6 | Cmd_reset_air_interface | 对 RF300 读写器,选择数据载体的类型；对 RF600 读写器,选择 ETSI 格式的通道。 |

表 3-3：软件库中不同的复位命令

**接收和评估确认信息**

在 RFID 收到复位报文后，立即对 RS232 模块发送一个确认信息。

在确认时传输的固件版本，由用户程序决定，在确认有效后，程序结束。

**③．从数据载体读取数据**

使用读报文，数据从数据载体中读取回来。如下图所示：

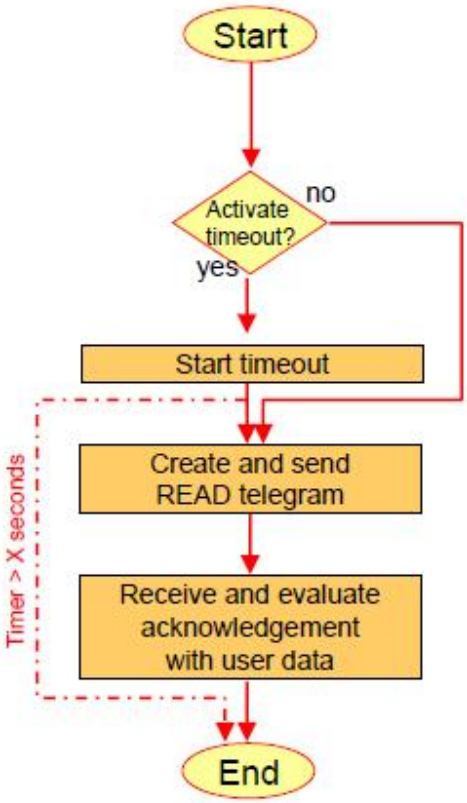



图 3-5：S7-1200 从 RFID 读写器读取数据

读命令存储在 RFID 读写器中，除非一个数据载体进入到读写器区域，主要原因是 RFID 的确认信息能够立刻或者在一定的时间内到达 S7-1200 控制器中，延时时间可以独立设定或者被禁止。

**创建发送报文**

在报文中传输的参数“_address”和“_length”决定了从数据载体中读取的数据存储区域。参数的值由所使用的数据载体决定。

例如：RF340 数据载体有一个 8K 字节的 FRAM 和一个 20 字节的 EEPROM 存储容量。如下图所示：

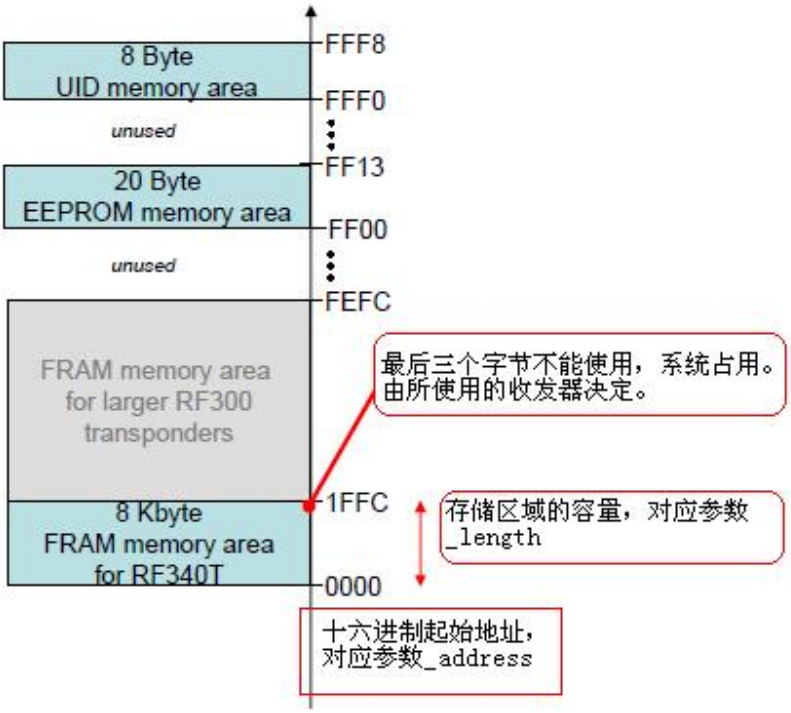



图 3-6：RF300 数据载体的存储区域

一次任务最多可以读 120 个字节，这是由 RFID 软件库所决定的。

关于数据载体更详细的信息参考 RFID 系统手册：

http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=21738946&caller=view

**接收和评估用户数据以及确认**

如果读报文的确认被接受，所需要的用户数据就会解包存储在全局数据块“rfid_serial_read_X”，用户程序可以使用这些数据。如下图所示：

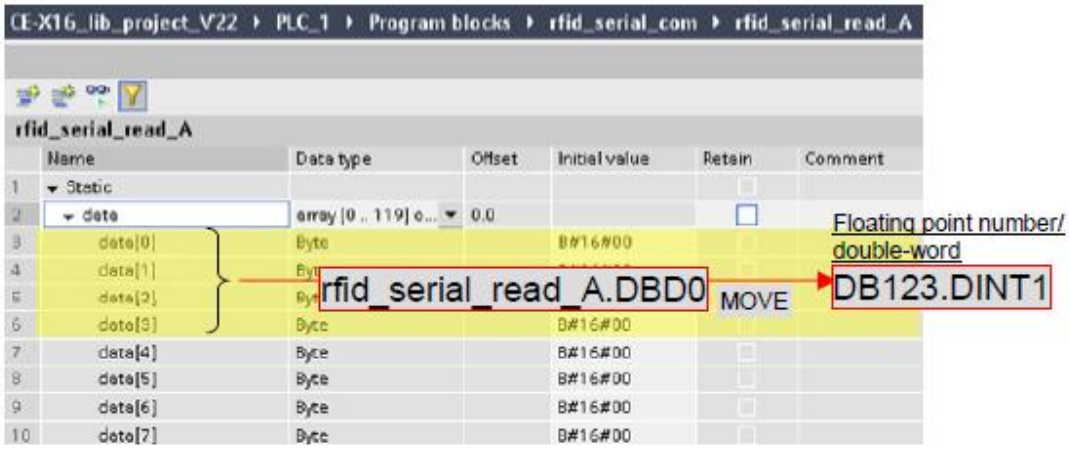



图 3-7：读取数据存储的数据块

全局数据块不是数据载体的存储表，读取的数据载体的数据总是存储在以数组.data(0)开始的数据中。

**④. 写数据到数据载体上**

写报文从 S7-1200PLC 写数据到数据载体的存储区中。写数据过程如下图所示：

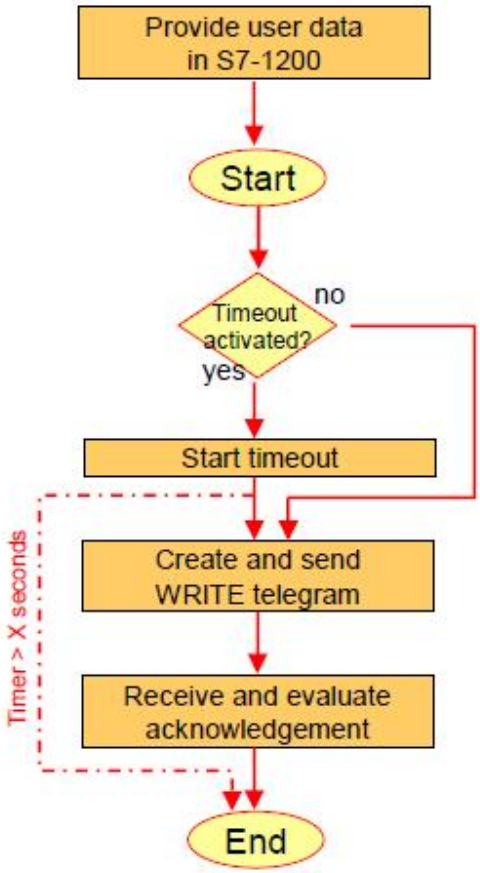



图 3-8：S7-1200 写数据到 RFID 数据载体

写命令存储在 RFID 读写器中，除非一个数据载体进入到读写器区域，主要原因是用户数据的写命令能够立刻或者在一定的时间内到达数据载体中，延时时间可以独立设定或者被禁止。

**在 S7-1200 中准备和发送用户数据**

在写命令被发送之前，用户数据首先要发送到全局数据块中使用的通道“rfid_serial_write_X”中。

如果发送一个双整数和一个字，那么六个字节的数据 data(0)...data(5)需要在全局数据块“rfid_serial_write_X ”设定。

如下图所示：

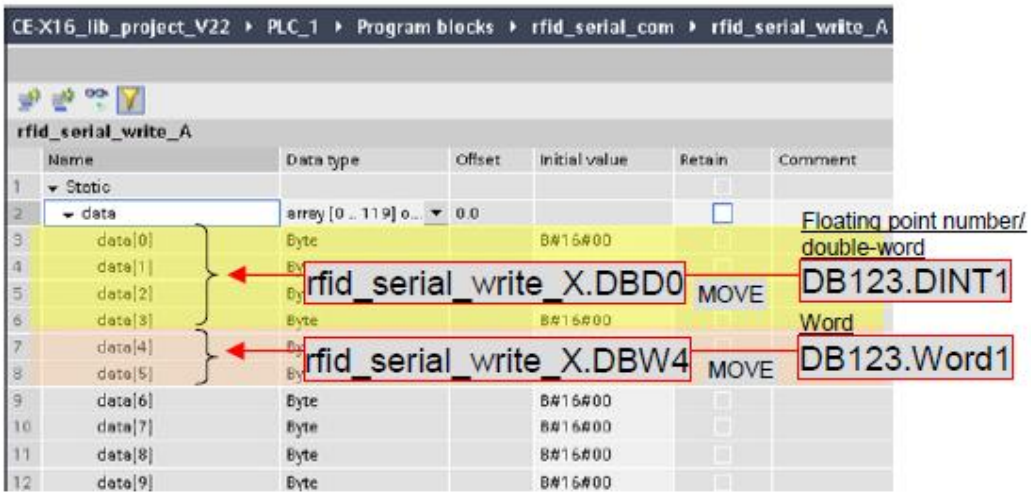



图 3-9：S7-1200 写数据到通道 A 的全局数据块中

全局数据块不是数据载体的存储表，用户数据总是存储在以 data(0)开始的数据中，与参数的地址“_address ” 16#0000 不同。

**创建和发送写报文**

参数“_address ”和“_length ”决定了数据载体需要被写入的存储区域，参数的值由所使用的数据载体决定。

例如：RF340 数据载体有一个 8K 字节的 FRAM 和一个 20 字节的 EEPROM 存储容量。如下图所示：

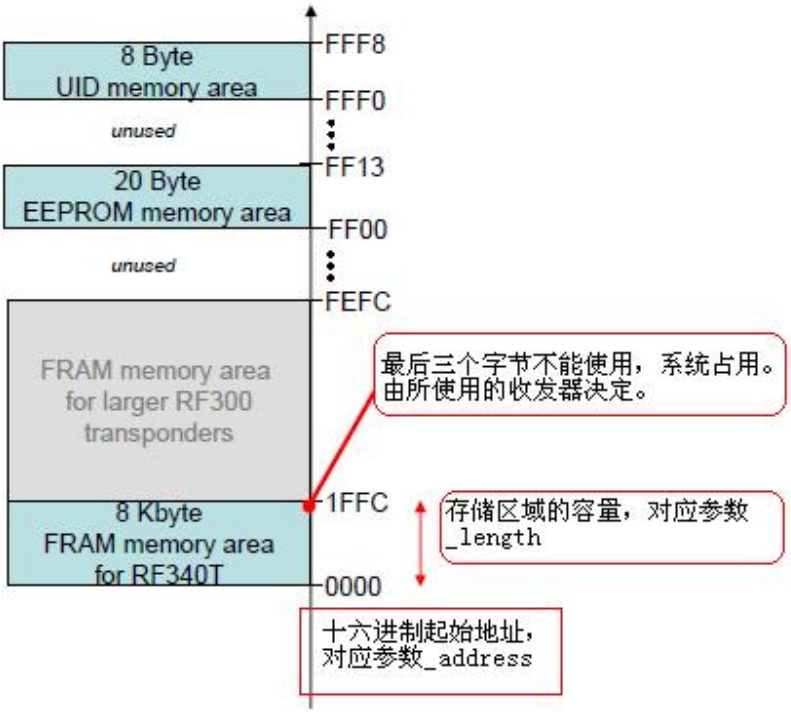



图 3-10：RF340 数据载体的存储区域分配

一次任务最多可以写 120 个字节，这是由 RFID 软件库所决定的。

**接收和评估用户数据以及确认信息**

在收到写报文后，RFID 读写器对 RS232 通信模块返回一个确认信息。

**⑤. 数据的执行**

在读写器和数据载体之间以及读写器处理过程的时间 ，还有完整的写时间，读取时间，以及 SIMATIC S7-1200 的传输时间，依据不同的环境，不同的负载，不同的变量，这些时间都有可能不同。最好确定读写的时间，需要测试至少 50 次。

下表是 S7-1200 CPU1214C 通过 RS232 模块和 RF260R 的读写时间。

| 类型 | 字节数量、 | 测量时间 |
|---|---|---|
| 读取 | 1 | 平均 89ms,最大 110ms |
| | 120 | 平均 198ms,最大 261ms |
| 写入 | 1 | 平均 93ms,最大 109ms |
| | 120 | 平均 151ms,最大 569ms |

表 3-4：S7-1200 CPU1214C 通过 RS232 模块和 RF260R 的平均读写时间

## 3.2 S7-1200 软件库的使用

### 3.2.1 集成软件库到 STEP7 V11 SP2 中

为了使用 RFID 软件库，需要首先把软件库集成到 STEP7 V11 SP2 中。具体集成的步骤见下面描述。

1. 从网上下载 CE-X16， S7-1200 RFID serial library.zip,然后解压缩后如下图所示：（软件库文件见附件)

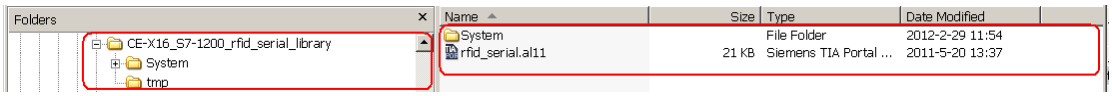



图 3-11：S7-1200 RFID serial library.zip,解压缩后的文件

2. 打开 STEP7 V11 SP2。
3. 点击全局库按钮，打开全局库。如下图所示：

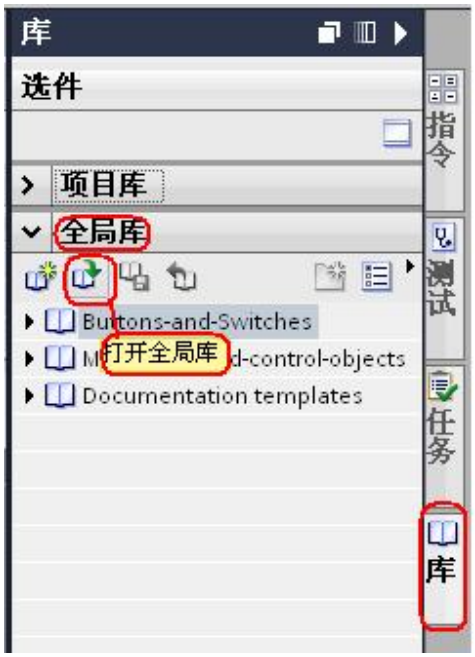



图 3-12：打开全局库按钮

4. 然后选择解压缩后的库文件,打开库文件中的执行文件。如下图所示：

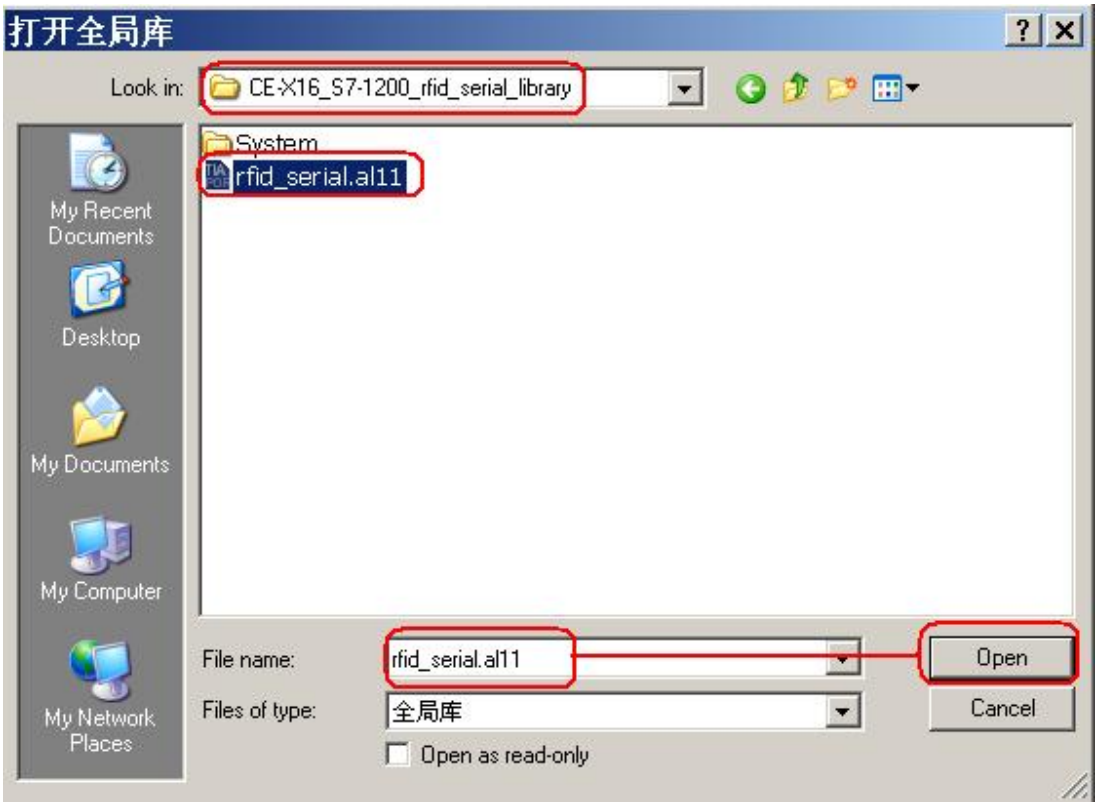



图 3-13：打开 RFID 软件库文件

5. RFID 软件库装载到 STEP7 V11 的全局库中。如下图所示：

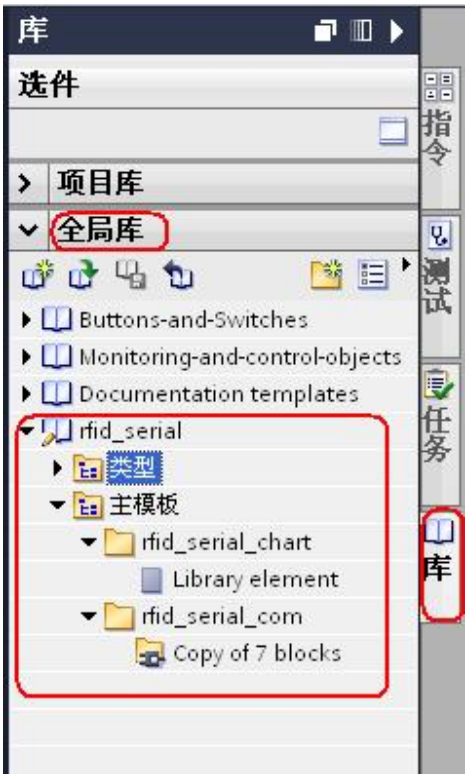



图 3-14： RFID 软件库文件集成后的显示

### 3.2.2 软件库的使用

1.打开 STEP7 V11SP2，新建一个 S7-1200 项目，添加硬件设备 CPU1214C。如下图所示：

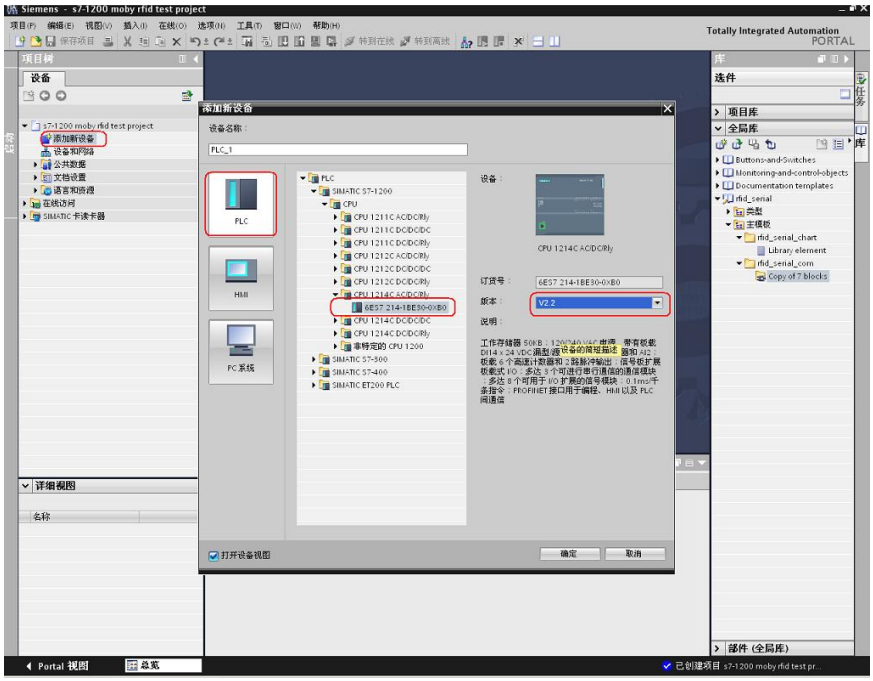


图 3-15： 添加 CPU 1214C AC/DC/RLY

2．点击程序块，打开 OB1 组织快。如下图所示：

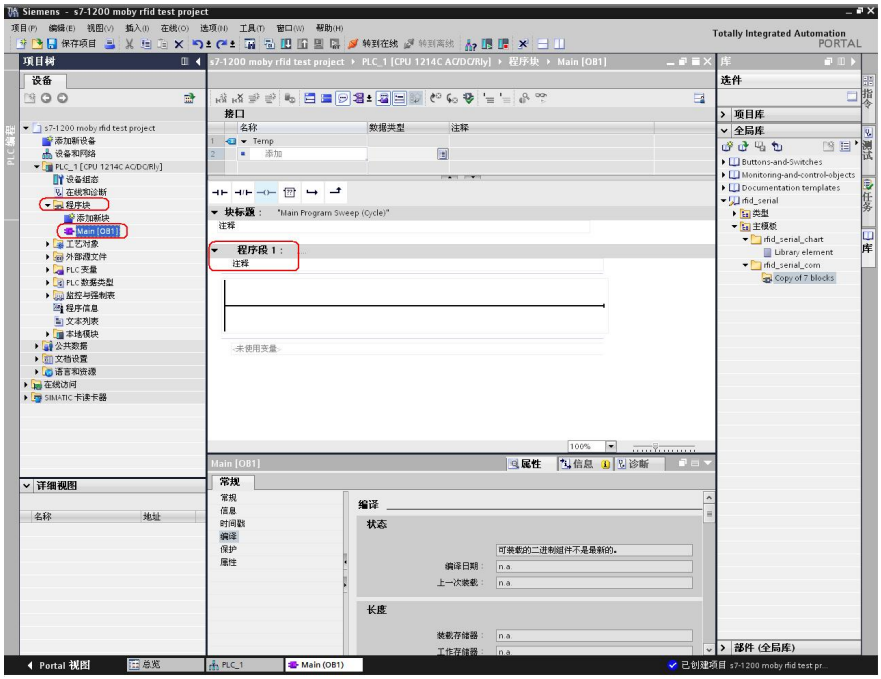



图 3-16：打开 OB1 组织快

3.点击库文件“rfid_serial”中的库文件夹“rfid_serial_com”,拷贝这些文件到程序块文件夹下。如下图所示：

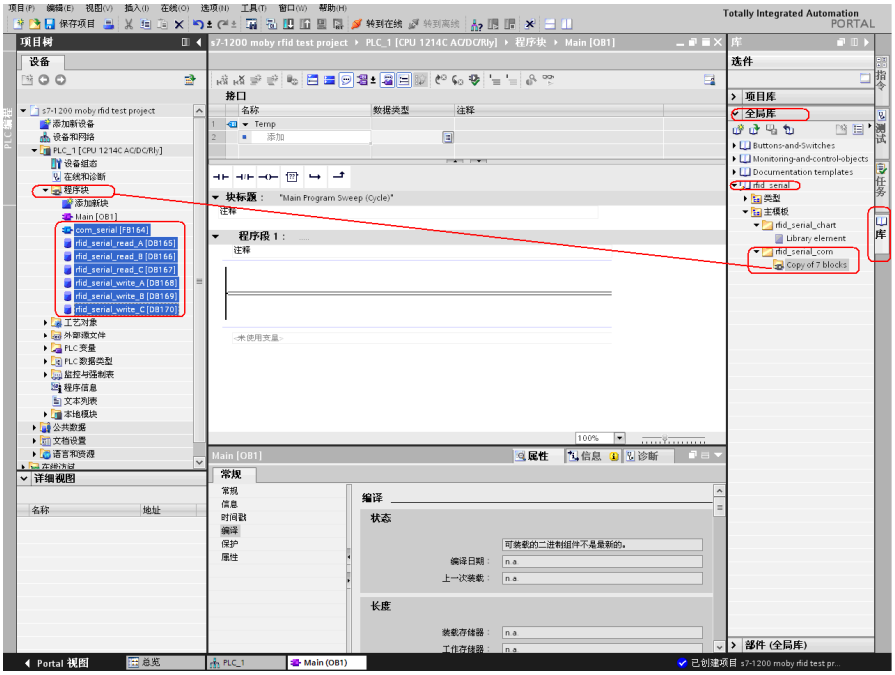


图 3-17：拷贝 RFID 软件库文件“rfid_serial_com”到程序块文件夹下

4. 点击库文件“rfid_serial”中的库文件夹“rfid_serial_chart”,拷贝这些文件到监控表中。如下图所示：

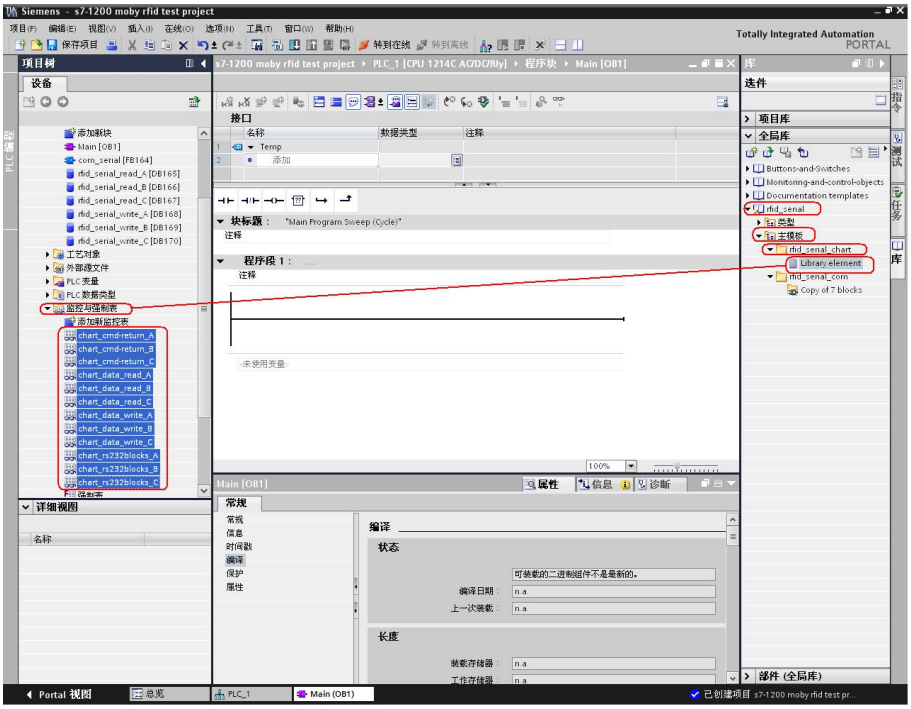


图 3-18：拷贝 RFID 软件库文件“rfid_serial_chart” 到监控表中

5.拖拽“com_serial”功能块到OB1中。

6.选择背景数据块的名字和数据块号，确认。如下图所示：

图3-19：选择“com_serial”功能块的背景数据块的名字和数据块号

7.软件库中的功能块的符号名以及编号能够自己设定，但是需要注意监控表中相对应的符号名。如下图所示：

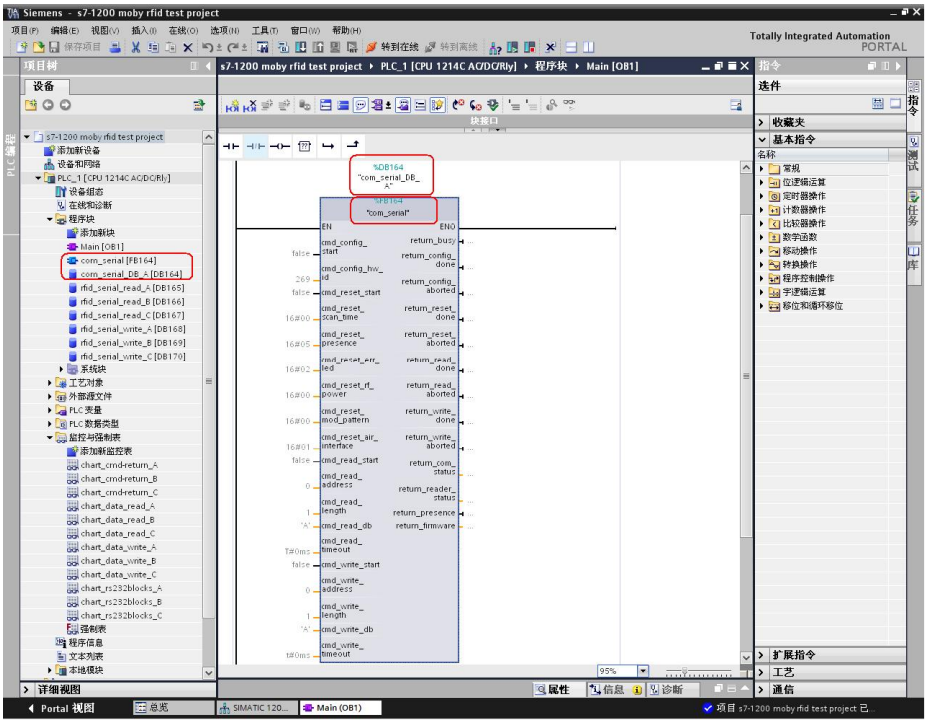



图 3-20：选择“com_serial”功能块的符号名以及编号

### 3.2.3 检查和更新软件库的版本

如果在已有的项目中更改软件库的版本，需要按照下述步骤来执行：

1.首先检查本项目库文件的版本号,如下图所示：

图 3-21：库文件的版本号

2.打开功能块或者数据块的属性窗口，检查当前的版本号。

3.如果需要更新软件库的版本号，需要添加新的版本的库文件。

4.删除本项目中程序块下所有的相关的库文件，不要删除 OB1。如下图所示：

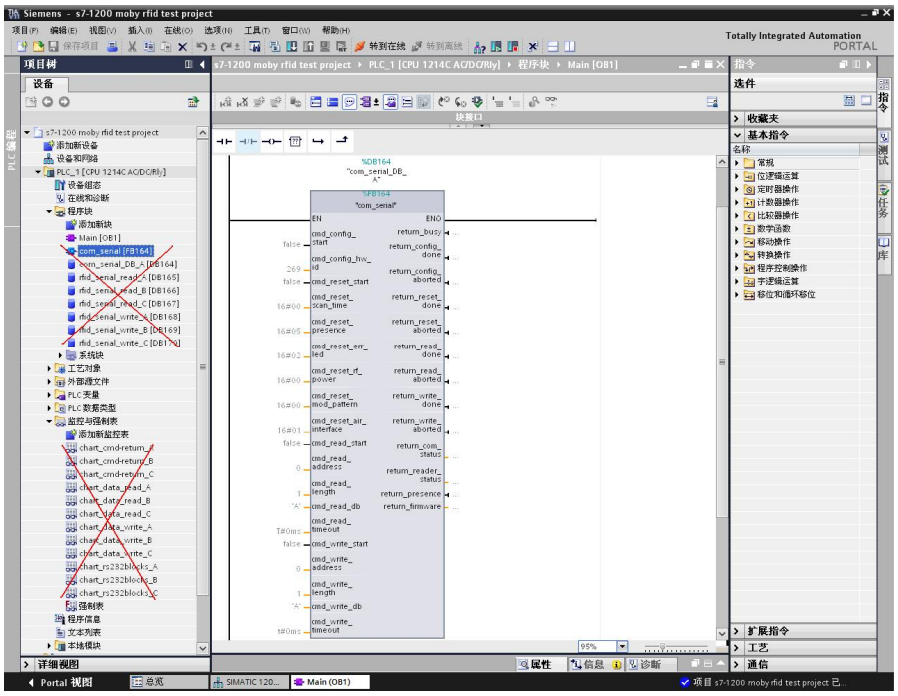



图 3-22：删除程序块下所有的相关的库文件

5.添加“rfid_serial_com”库文件到程序块下。

6.在 OB1 中调用的功能块“com_serial”显示丢失背景数据块。如下图所示：

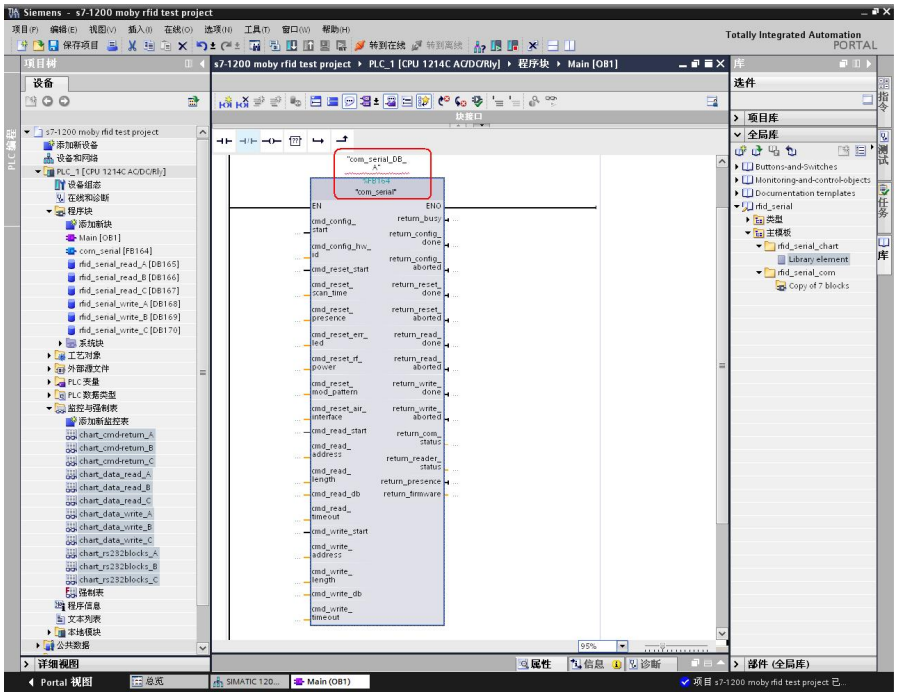



图 3-23：OB1 中调用的功能块“com_serial”显示丢失背景数据块

7.手动添加背景数据块，选择数据块类型为“com_serial”FB164 的背景数据块，定义数据块号。如下图所示：

图 3-24：添加 OB1 中调用的功能块“com_serial”的丢失背景数据块

8.更新完成，所有的库文件都已经更新完成。

3.3 S7-1200 软件库的详细描述

软件库的接口功能块，如下图所示：

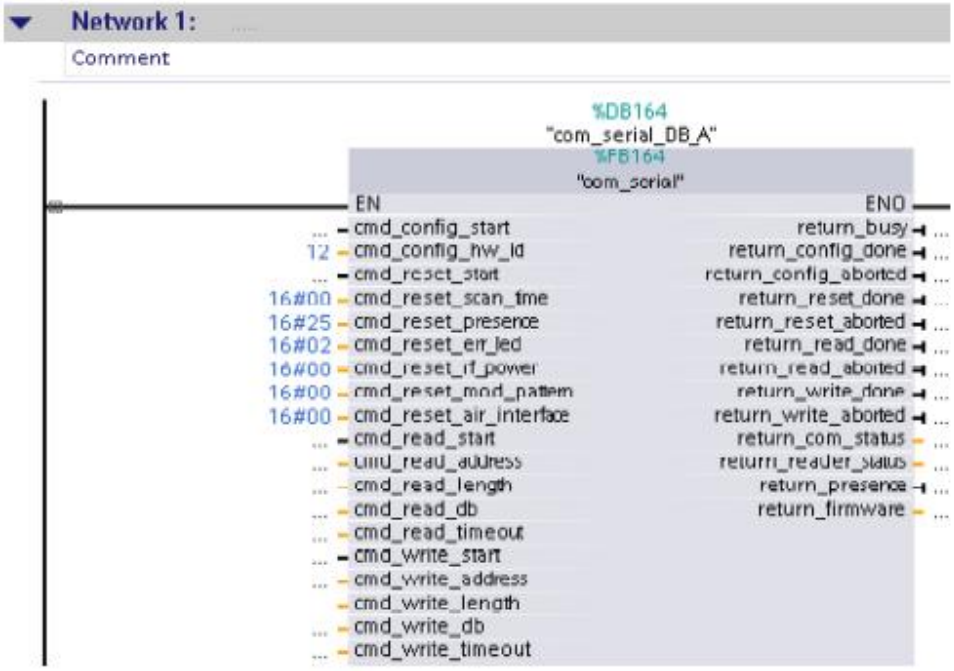



图 3-25：软件库的接口功能块“com_serial”

### 3.3.1 参数的配置 _cmd_config_

图 3-26：RFID 通信配置参数

具体描述如下表所示：

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| 1. | cmd_config_start | IN | Bool | • Enables the configuration routine<br>• Reacts to a positive edge<br>• The start command is stored as long as the "com" function block is already processed. The block saves only one start command at a time, provided it cannot be processed instantly. |
| 2. | cmd_config_hw_id | IN | Port | • Hardware ID of the RS232 communication module<br>• Default value: 269<br>• Check hardware ID in device information/RS232_1-Properties/RS232 interface/I/O addresses hardware ID |

表 3-5：RFID 通信配置参数

### 3.3.2 复位命令参数 _cmd_reset_

图 3-27：RFID 复位参数

具体描述如下表所示：

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| 1. | cmd_reset_start | IN | Port | • Enables the routine for sending a RESET telegram<br>• Interrupts the routine READ or WRITE when it is already running (If the READ or WRITE routine is not yet active it will be executed normaly after this RESET routine)<br>• Reacts to a positive edge<br>• The start command is stored as long as the "com_serial" function block is already processed. The block saves <u>only one</u> start command at a time, provided it cannot be processed instantly. |
| 2. | cmd_reset_scan_time | IN | Byte (Hex) | • Only for RF600-Reader<br>• Configures the RESET telegram for the reader with regards to the radio profile<br>• Permissible values see /5/, chapter 5.2.2, parameter "scaning_time"<br>• Default value: 16#00 |
| 3. | cmd_reset_presence | IN | Byte (Hex) | • For all RFID-Reader<br>• Configures the RESET telegram for the reader with regards to the presence message<br>• Permitted values<br>- 16#25: operation with presence message<br>- 16#05: operation without presence message<br>• More information see /4/, chapter 3.2.2.1, parameter "param"<br>• Default value: 16#05 |
| 4. | cmd_reset_err_led | IN | Byte (Hex) | • Only for RF200/300-Reader:<br>• Configures the RESET telegram for the reader with regards to the ERROR-Led<br>• Permitted values<br>- 16#02: reset ERROR-Led<br>- 16#00: do not reset ERROR-Led<br>• More information see /4/, chapter 3.2.2.1, parameter "opton1" |

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| | | | | • Default value: 16#02 |
| 5. | cmd_reset_rf_power | IN | Byte (Hex) | • Only for RF380-Reader:<br>• Configures the RESET telegram with regards to the send power of the RFID-Reader<br>• Permitted values<br>- 16#02: 0.5 W<br>- 16#03: 0.75 W<br>- 16#04: 1.0 W<br>- 16#00 or 16#05: 1.25 W<br>- 16#06: 1.5 W<br>- 16#07: 1.75 W<br>- 16#08: 2.0 W<br>• More information see /4/, chapter 3.2.2.1, parameter "dili"<br>• Default value: 16#00 |
| 6. | cmd_reset_mod_pattern | IN | Byte (Hex) | • Only for RF600-Readers<br>• Configures the RESET telegram with regards to the used modulation pattern of the reader<br>• Permissible values see /5/, chapter 5.2.2, parameter "fcon"<br>• Default value: 16#00 |
| 7. | cmd_reset_air_interface | IN | Byte (Hex) | • Different meaning for RF200/300-Reader and RF600-Reader<br>• Following meaning for RF200/300-Reader<br>• Configures the RESET telegram with regards to the used transponder type<br>• Permitted values<br>- 0: RF300<br>- 1: ISO (general)<br>- 3: ISO (Infineon-Chip)<br>- 4: ISO (Fujitsu-Chip)<br>- 5: ISO (NXP-Chip)<br>- 6: ISO (TI-Chip)<br>- 7: ISO (ST-Chip)<br>• More information see /4/, chapter 3.2.2.1, parameter "ftim"<br>• Default value: 1 |

表 3-6：RFID 复位参数

### 3.3.3 读命令参数 _cmd_read_

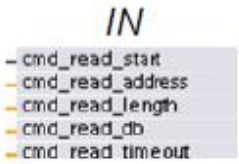



图 3-28：RFID 读取参数

具体描述如下表所示：

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| 1. | cmd_read_start | IN | Bool | • Enables the routine for sending a READ telegram<br>• Reacts to a positive edge<br>• The start command is stored as long as the "com" function block is already processed. The block saves only one start command at a time, provided it cannot be processed instantly. |
| 2. | cmd_read_address | IN | Word (Hex) | • Specifies from which address onward the user data shall be fetched from the transponder<br>• Permissible value: depending on the transponder type |
| 3. | cmd_read_length | IN | Int | • Specifies how much data (in bytes) shall be fetched from the transponder<br>• The entry of this parameter also determines how many user data are stored in the global data block "rfid_serial_read_X" (from array element .data[0])<br>• Permitted values: 1 to 120 |
| 4. | cmd_read_DB | IN | Char | • Choose the global data block where you want to store you read Transponder-data<br>• • Permitted values:<br>- A: chooses the global data block „rfid_serial_read_A"<br>- B: chooses the global data block „rfid_serial_read_B"<br>- C: chooses the global data block „rfid_serial_read_C" |
| 5. | cmd_read_timeout | IN | Time | • Defines how long to wait for the acknowledgement telegram<br>• Input in milliseconds<br>• Permitted values:<br>- 0: deactivates the timer (waits indefinitely)<br>- >0: activates the timer and specifies the wait time. No less than 500ms should be entered.<br>• Default value: 0ms |

表 3-7：RFID 读取参数

### 3.3.4 写命令参数 _cmd_write_

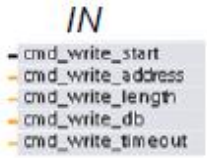



图 3-29：RFID 写入参数

具体描述如下表所示：

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| 1. | cmd_write_start | IN | Bool | • Enables the routine for sending a WRITE telegram<br>• Reacts to a positive edge<br>• The start command is stored as long as the "com" function block is already processed. The block saves only one start command at a time, provided it cannot be processed instantly. |
| 2. | cmd_write_address | IN | Word (Hex) | • Specifies from which address onward the user data shall be written to the transponder<br>• Permissible value: depending on the transponder type |
| 3. | cmd_write_length | IN | Int | • Specifies how much data (in bytes) shall be written to the transponder<br>• The entry of this parameter also determines how many user data are stored in global data block "rfid_serial_write_X" (from array element .data[0])<br>• Permitted values: 1 to 120 |
| 4. | cmd_write_DB | IN | Char | • Choose the global data block where the data is stored you want to send/write<br>• • Permitted values:<br>- A: chooses the global data block „rfid_serial_write_A"<br>- B: chooses the global data block „rfid_serial_write_B"<br>- C: chooses the global data block „rfid_serial_write_C" |
| 5. | cmd_write_timeout | IN | Time | • Defines how long to wait for the acknowledgement telegram<br>• Input in milliseconds<br>• Permitted values:<br>- 0: deactivates the timer (waits indefinitely)<br>- >0: activates the timer and specifies the wait time. No less than 500ms should be entered.<br>• Default value: 0 ms |

表 3-8：写入参数

### 3.3.5 返回命令参数 return_

图 3-30：RFID 状态参数

具体描述如下表所示：

| No. | Name | Transfer | Data type | Description/note |
|---|---|---|---|---|
| 1. | return_busy | OUT | Bool | • Signals when the "com_serial" block is busy with processing a routine<br>• Takes on the TRUE state once a "cmd_" command has been triggered.<br>• Takes on the FALSE state as soon the routine terminates |
| 2. | return_config_done | OUT | Bool | • Gives feedback if the associated routine was terminated without error ("done") or was terminated with error ("aborted")<br>• If an error occurred during the processing ("aborted) of a routine the parameter No. 10 of this table is relevant<br>• Default value ("done"): FALSE<br>• Default value ("aborted"): TRUE |
| 3. | return_config_aborted | OUT | Bool | |
| 4. | return_reset_done | OUT | Bool | |
| 5. | return_reset_aborted | OUT | Bool | |
| 6. | return_read_done | OUT | Bool | |
| 7. | return_read_aborted | OUT | Bool | |
| 8. | return_write_done | OUT | Bool | |
| 9. | return_write_aborted | OUT | Bool | |
| 10. | return_com_status | OUT | Int | • In the case of an error, returns the status to be able to localize the cause of the error<br>• Always to be considered in combination with the status list in chapter 4<br>• Error number is generated in the program |
| 11. | return_reader_status | OUT | Word (Hex) | • In the case of an error, it returns the status to be able to localize the cause of the error<br>• Always to be considered in combination with the document in /4/, Appendix A.2<br>• Error number is generated in the RFID-Reader |
| 12. | return_presence | OUT | Bool | • Signals when a transponder has entered or left the field of the RFID-Reader<br>• Only relevant in combination with activated presence message (see Table 3-2 no. 2) |
| 13. | return_firmware | OUT | Word (Hex) | • Specifies the firmware version of the used RFID-Reader<br>• Only output after successful processing of the "_reset_" routine |

表 3-9：RFID 状态参数

## 3.4 RFID 软件库的报警信息和故障原因

功能块“com_serial”数据块指示了出现故障的原因，符号表地址是“return_com_status”。状态字的输出是一个十进制的形式，下表中可以查到它的各种状态。

另外，RFID 读写器的错误状态也有输出，输出的状态字是十六进制形式，符号表的变 量地址是“return_reader_status”。

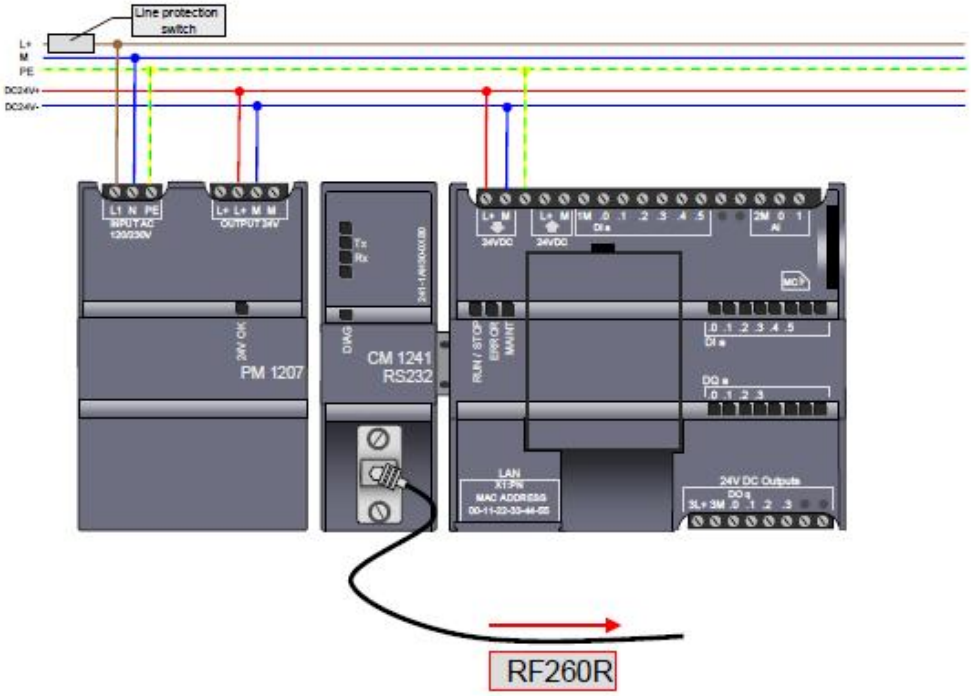
| State | Description |
|---|---|
| 1000 | The port configuration was finished incorrectly: PORT_CFG |
| 1010 | The configuration for receiving the characters was finished incorrectly: RCV_CFG |
| 1020 | The configuration for sending the characters was finished incorrectly: SEND_CFG |
| 1999 | A time-out occurred during the configuration of the RS232 module. |
| 2051 | The send permission was not given by the RFID-Reader. The Reader receipted with NAK. |
| 2052 | The send permission was not give by the RFID-Reader. The Reader sends no character to the S7-1200 PLC. |
| 2081 | Our telegram was accepted by the RFID-Reader, but the acknowledgement telegram from the reader contains a faulty entry |
| 2082 | Our telegram was refused by the RFID-Reader. |
| 2999 | A time-out occurred during the RESET routine. |
| 3000 | Faulty input for parameter "_length" at the block input for the READ routine |

| State | Description |
|---|---|
| 3051 | The send permission was not given by the RFID-Reader. The Reader receipted with NAK. |
| 3052 | The send permission was not give by the RFID-Reader. The Reader sends no character to the S7-1200 PLC. |
| 3081 | Our telegram was accepted by the RFID-Reader, but the acknowledgement telegram from the reader contains a faulty entry |
| 3082 | Our telegram was refused by the RFID-Reader. |
| 3999 | A time-out occurred during the READ routine. |
| 4000 | Faulty input for parameter "_length" at the block input for the WRITE routine |
| 4061 | The send permission was not given by the RFID-Reader. The Reader receipted with NAK. |
| 4062 | The send permission was not give by the RFID-Reader. The Reader sends no character to the S7-1200 PLC. |
| 4081 | Our telegram was accepted by the RFID-Reader, but the acknowledgement telegram from the reader contains a faulty entry |
| 4082 | Our telegram was refused by the RFID-Reader. |
| 4999 | A time-out occurred during the WRITE routine The acknowledgement telegram from RFID-Reader did not arrive |

表 3-10：RFID 读写器以及通信状态信息代码

## 4. S7-1200 与 RF260R 的配置

### 4.1 S7-1200 与 RF260R 的接线图



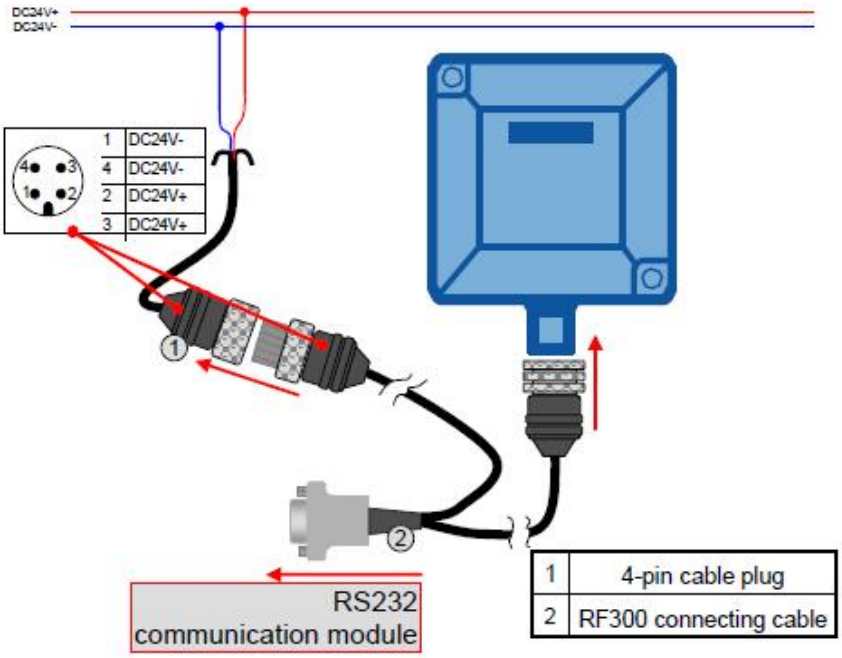




图 4-1：S7-1200 RS232 模块与 RF260R 的接线图

## 4．2 S7-1200 与 RF260R 连接的硬件和软件

### 4.2.1 硬件部分

S7-1200 与 RF260R 连接的硬件列表如下：

| 硬件 | 数量 | 订货号 | 注释 |
|---|---|---|---|
| SIMATIC PM 1207 | 1 | 6ES7241-1AH30-0XB0 | 2.5 A |
| SIMATIC CPU 1211C | 1 | 6ES7211-1AD30-0XB0 | |
| SIMATIC CM 1241 | 1 | 6ES7241-1AH30-0XB0 | RS232 |
| SIMATIC RF260R | 1 | 6GT2821-6AE00 | 读写器 |
| SIMATIC MDSD126 | 3 | 6GT2600-0AC00 | 收发器 |
| SIMATIC RF300 connecting cable | 1 | 6GT2891-0KH50 | |
| 保护开关 | 1 | 5SX2116-6 | 1 pol. B, 16A |
| 4-pin cable plug | 1 | | |
| 以太网电缆 | 1 | | |

表 4-1：S7-1200 与 RF260R 连接的硬件列表

注意：

这种解决方案对于别的 RFID 读写器 RF200，RF300 或者 RF600（甚至 1 个系统的不同模块）都是可行的，如果是 RS422 连接形式则需要一个 RS232/RS422 转换器。

### 4.2.2 软件部分

| 软件部分 | 数量 | 订货号 |
|---|---|---|
| SIMATIC STEP7 Basic V11 | 1 | 6ES7822-0AA01-0YA0 |

表 4-2：S7-1200 与 RF260R 连接的软件列表

## 5. S7-1200 与 RF260R 通信的编程

### 5. 1 S7-1200 STEP7 V11 建立项目以及参数设置

1. 进行硬件配置，添加 CPU 和 RS232 模块。如下图所示：

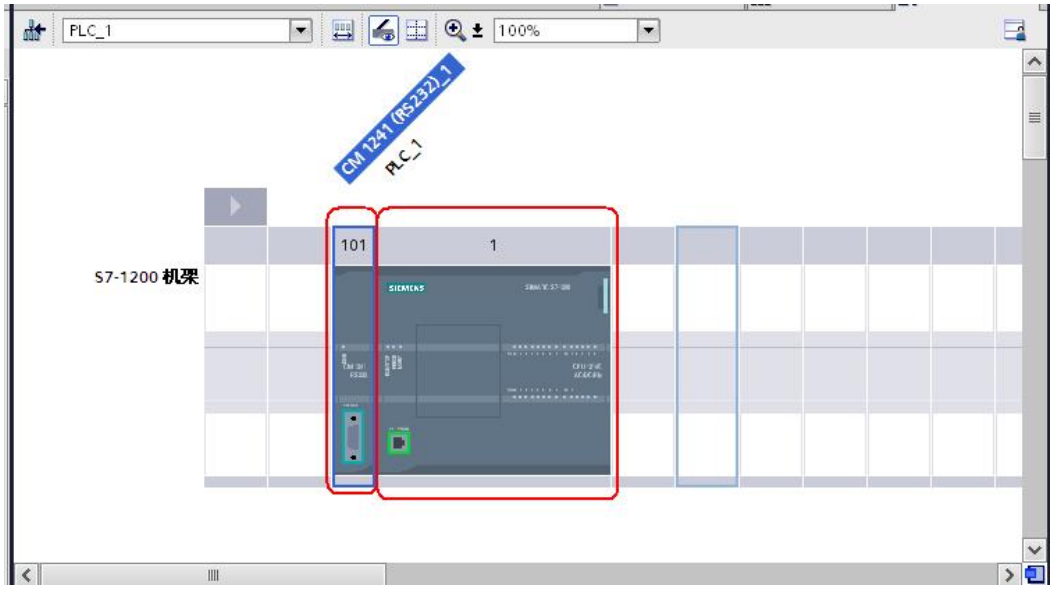



图 5-1：硬件配置

2. RS232 模块的参数设置，如下图所示：

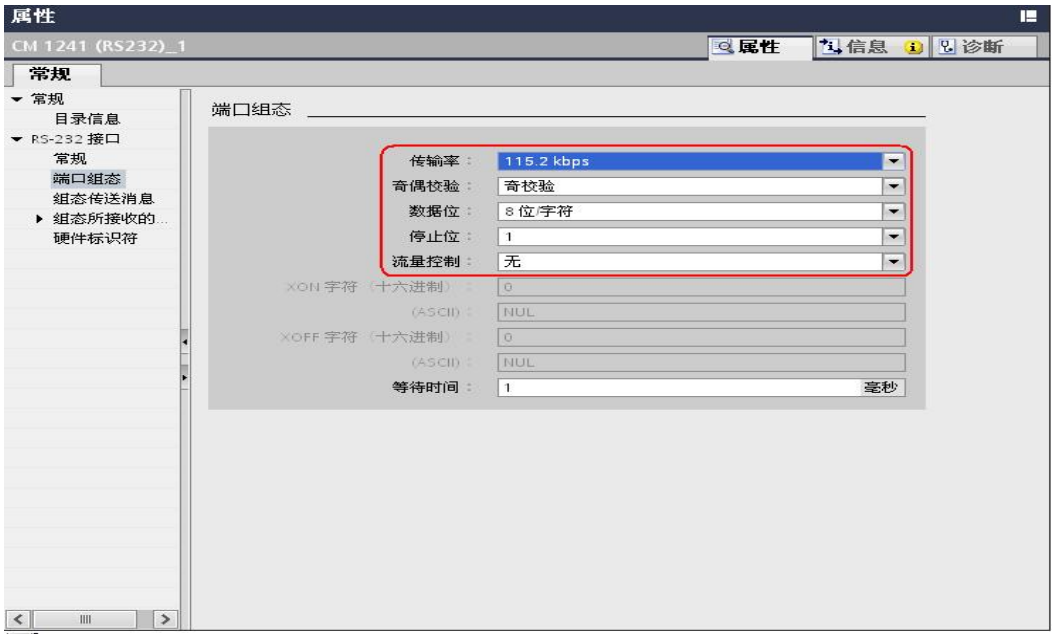



图 5-2：RS232 模块的参数设置

由于 RF260R 的传输速率只有 19200，57600，115200 baud,因此 RS232 模块的传输速率也只能在这三种速率之间选择，设置其它的速率都会导致不能通信。

5．2 调用 RFID 库文件到主程序中

根据最新的库文件版本，调用新的库文件到相应的 CPU 硬件版本下，如下图所示：

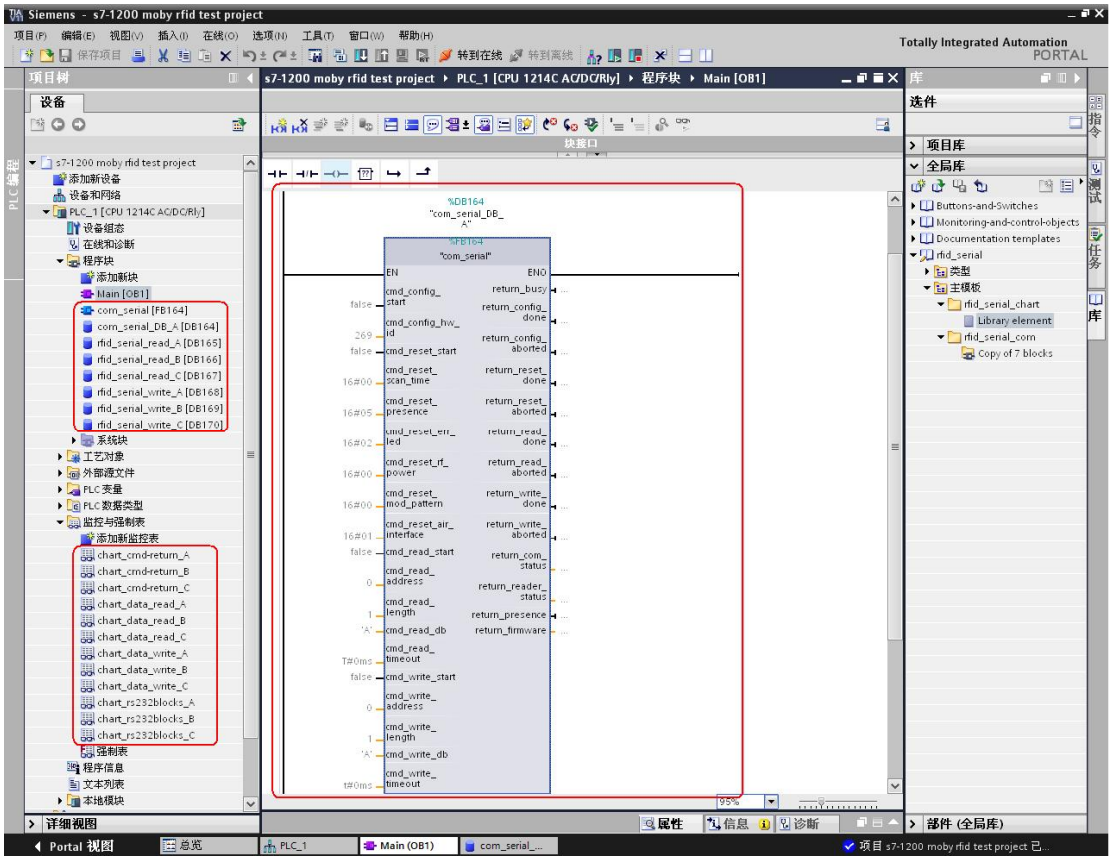



图 5-3：调用 RFID 库文件到主程序中

5．3 在 S7-1200 中编写 RFID 调用程序

在 S7-1200 中，对 RFID 软件库进行编程，编写相应的配置，复位，读取，写入程序，以及相应的状态监控。如下图所示：

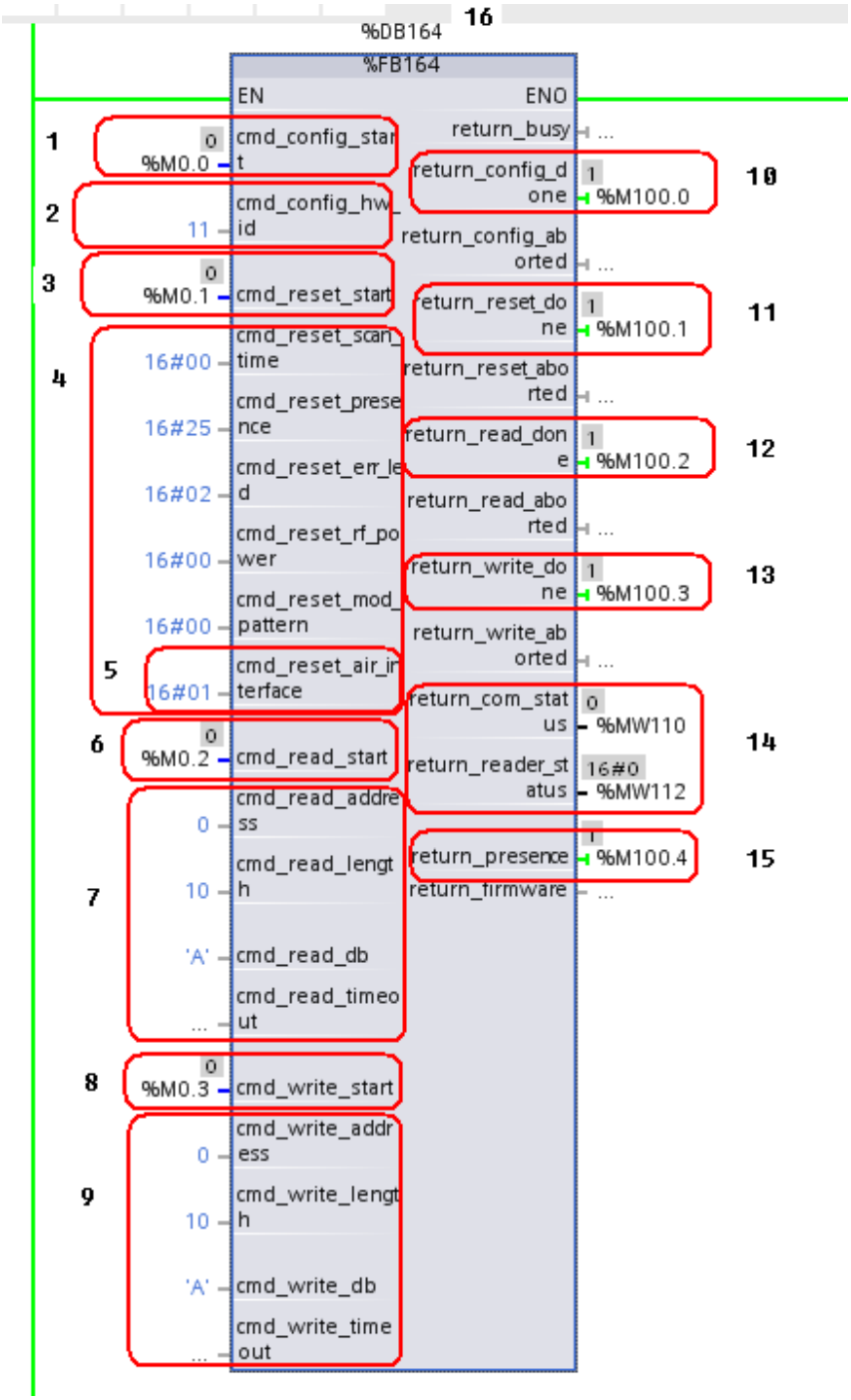



图 5-4：对 RFID 库文件进行编程

各参数的具体含义及设置如下：

1. Cmd_config_start　　布尔变量，上升沿有效。
2. cmd_config_hw_id　　RS232 模块的硬件编号，在此选择 RS232 模块即可，最多可添加 3 个 RS232 模块。
3. cmd_reset_start　布尔变量，上升沿有效。
4. cmd_reset_scan_time, cmd_reset_presence, cmd_reset_err_led, cmd_reset_rf_power, cmd_reset_mod_pattern, cmd_reset_air_interface,复位接口参数，对于不同的 RFID 读写器有不同的参数相对应。
5. 对于 RF260R,

cmd_reset_presence　　设置为 16#25　表示对于数据载体是否存在状态进行显示，

cmd_reset_err_led　　设置为 16#02　表示对读写器上的错误指示灯进行复位，

cmd_reset_air_interface　设置为 16#01　表示对于数据载体的类型设置位 ISO 标签类型。

6. cmd_read_start　布尔变量，上升沿有效。
7. cmd_read_address: 设置为 0，表示从数据载体上的 0 地址开始读。
   cmd_read_length　读取的字节长度。
   cmd_read_db　　读取的数据放置的数据块。
   cmd_read_time　　等待确认报文的时间。
8. cmd_write_start　布尔变量，上升沿有效。
9. cmd_read_address: 设置为 0，表示从数据载体上的 0 地址开始读。
   cmd_write_length　写入的字节长度。
   cmd_write_db　　写入的数据放置的数据块。
   cmd_write_time　　等待确认报文的时间。
10. return_config_done :设置完成的标示，读写器第一次上电后，必须进行初始化设置，设置是否完成，依此位的标示来表示，只有此变量为 1，才能进行下一步操作。
11. return_reset_done:　复位完成的标示，设置完成后才能进行复位操作。
12. return_read_done: 读取完成的标示，复位完成后才能进行读取操作。
13. return_write_done:写入完成的标示，复位完成后才能进行写入操作。
14. return_com_status,return_reader_status: 程序错误和读写器的错误状态。
15. return_presence: RFID 数据载体是否到达读写器的区域。
16. DB164:　FB164 功能块的背景数据块。

5．4 编程对于 RFID 读写器的初始化

1．进行 config 操作，return_config_done 完成标示位必须为 1，才表示配置完成，否则必须修改配置参数，直到配置完成。如下图所示：

没有进行初始化之前的状态是：RF260R 读写器的状态 LED 指示灯每 0.5 秒闪烁一次。如下图所示：

图 5-5：进行配置和初始化之前的状态

**进行 config 操作，如下图所示：**

图 5-6：Config 操作后的状态显示

进行 config 操作后，config done 完成标示位为 1，此时 RF260R 的指示灯还没有变化，还在每 0.5 秒闪烁一次，表示初始化还没有完成。

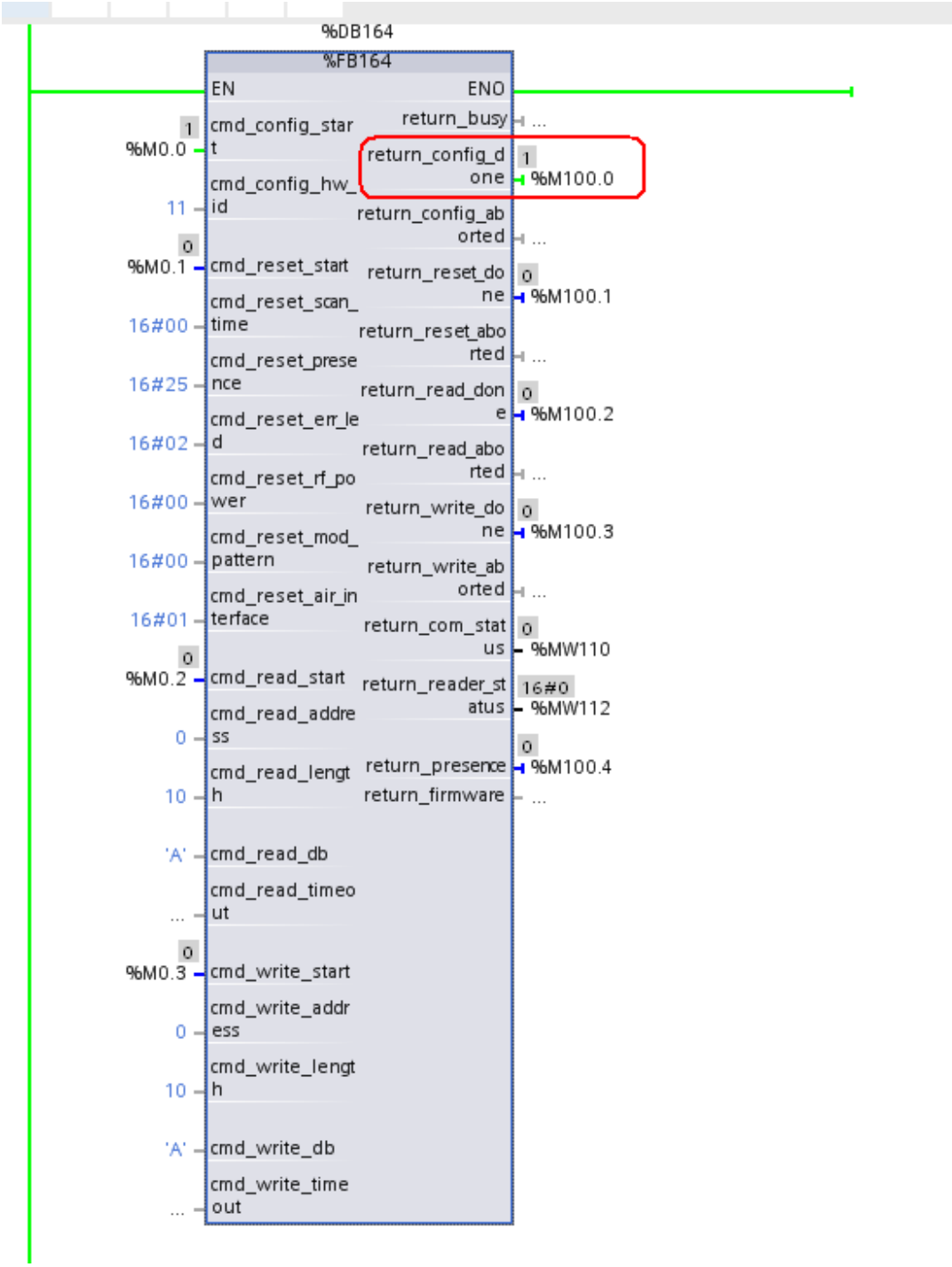



图 5-7：Config 操作后的状态显示

进行 **reset** 操作，如下图所示：

mobyv105test ▸ PLC_1 ▸ 监视表格 ▸ 监视表格_1

| | 名称 | 地址 | 显示格式 | 监视值 | 修改值 | 注释 |
|---|---|---|---|---|---|---|
| 1 | "Tag_1" | %M0.0 | 布尔型 | TRUE | TRUE | config start |
| 2 | "Tag_6" | %M0.1 | 布尔型 | TRUE | TRUE | reset start |
| 3 | "Tag_3" | %M0.2 | 布尔型 | FALSE | FALSE | read start |
| 4 | "Tag_4" | %M0.3 | 布尔型 | FALSE | FALSE | write start |
| 5 | | %M0.4 | 布尔型 | FALSE | | |
| 6 | "Tag_5" | %M0.5 | 布尔型 | FALSE | | |
| 7 | | %M0.6 | 布尔型 | FALSE | | |
| 8 | "Tag_7" | %M100.0 | 布尔型 | TRUE | | config done |
| 9 | "Tag_8" | %M100.1 | 布尔型 | TRUE | | reset done |
| 10 | "Tag_9" | %M100.2 | 布尔型 | FALSE | | read done |
| 11 | "Tag_10" | %M100.3 | 布尔型 | FALSE | | write done |
| 12 | "Tag_13" | %M100.4 | 布尔型 | FALSE | | presence |
| 13 | | %M100.5 | 布尔型 | FALSE | | |
| 14 | | %M100.6 | 布尔型 | FALSE | | |
| 15 | | %M100.7 | 布尔型 | FALSE | | |
| 16 | | %M101.0 | 布尔型 | FALSE | | |
| 17 | "Tag_11" | %MW110 | 带符号十进制 | 0 | | |
| 18 | "Tag_12" | %MW112 | Hex | 0000 | | |
| 19 | | | | | | |

图 5-8：reset 操作后的状态显示

Reset 操作后，reset done 完成标示位为 1，此时 RF260R 的指示灯有变化了，每 0.5 秒闪烁一次的指示灯变成常绿的了，表示初始化成功完成。如下图所示：

图 5-9： RF260R 在复位参数成功后的 LED 状态指示

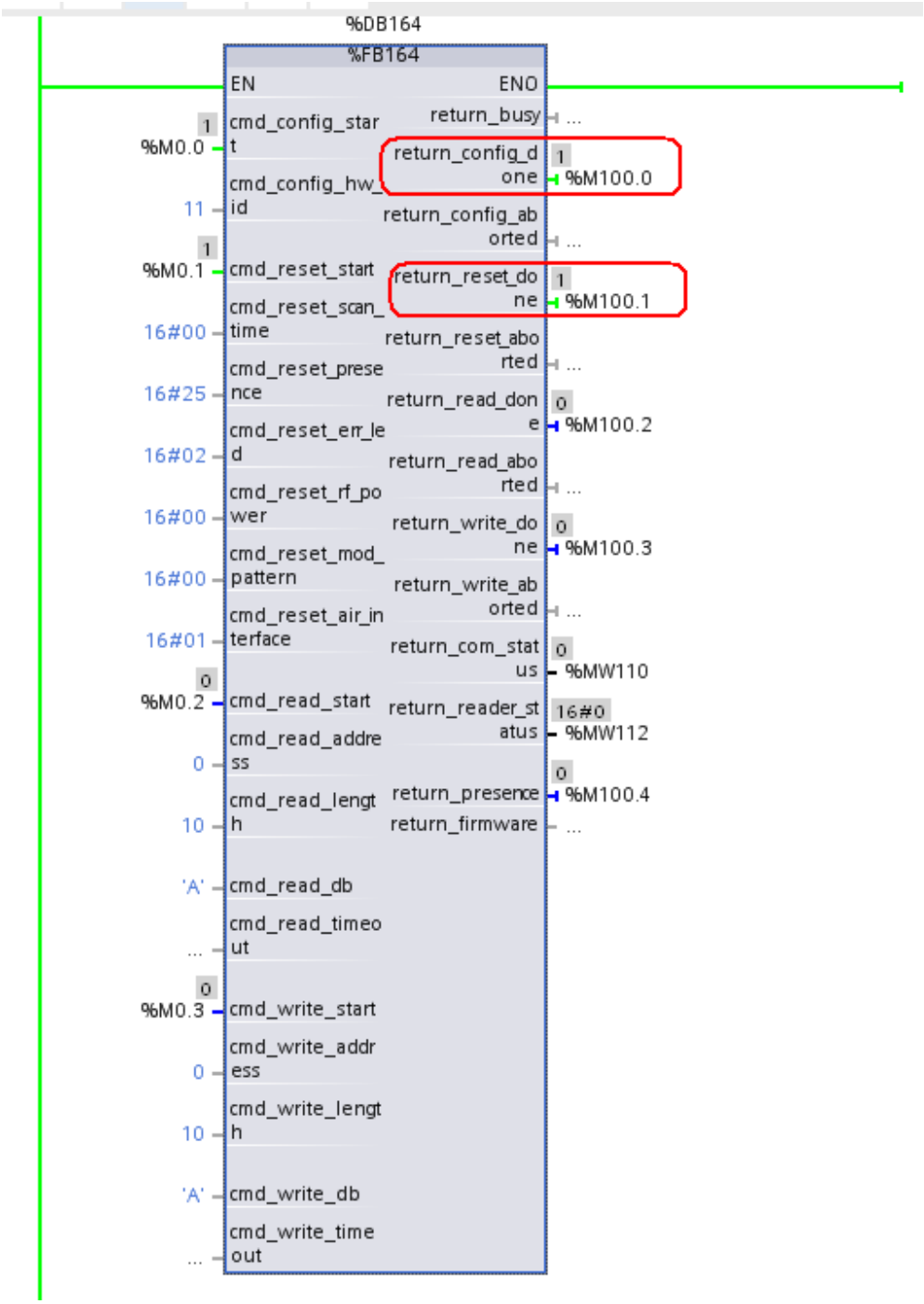



图 5-10：reset 操作后的状态显示

5．4 编程从 S7-1200 写数据到 RF260R 的数据载体

初始化成功后，可以进行从 S7-1200 到 MOBY 数据载体之间的读写操作了，此时把 MOBY D MDS D126 放置到 RF260R 的读写区域内，可以看到数据载体的是否存在的状态了。如下图所示：

mobyv105test ▸ PLC_1 ▸ 监视表格 ▸ 监视表格_1

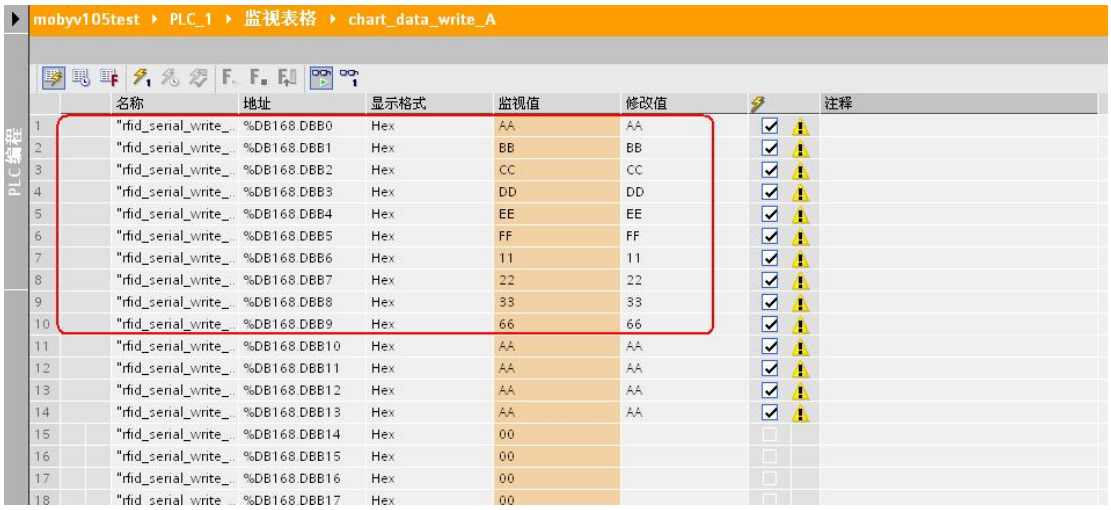
| | 名称 | 地址 | 显示格式 | 监视值 | 修改值 | | 注释 |
|---|---|---|---|---|---|---|---|
| 1 | "Tag_1" | %M0.0 | 布尔型 | TRUE | TRUE | ☑ | config start |
| 2 | "Tag_6" | %M0.1 | 布尔型 | TRUE | TRUE | ☑ | reset start |
| 3 | "Tag_3" | %M0.2 | 布尔型 | FALSE | FALSE | ☑ | read start |
| 4 | "Tag_4" | %M0.3 | 布尔型 | FALSE | FALSE | ☑ | write start |
| 5 | | %M0.4 | 布尔型 | FALSE | | | |
| 6 | "Tag_5" | %M0.5 | 布尔型 | FALSE | | | |
| 7 | | %M0.6 | 布尔型 | FALSE | | | |
| 8 | "Tag_7" | %M100.0 | 布尔型 | TRUE | | | config done |
| 9 | "Tag_8" | %M100.1 | 布尔型 | TRUE | | | reset done |
| 10 | "Tag_9" | %M100.2 | 布尔型 | FALSE | | | read done |
| 11 | "Tag_10" | %M100.3 | 布尔型 | FALSE | | | write done |
| 12 | "Tag_13" | %M100.4 | 布尔型 | TRUE | | | presence |
| 13 | | %M100.5 | 布尔型 | FALSE | | | |
| 14 | | %M100.6 | 布尔型 | FALSE | | | |
| 15 | | %M100.7 | 布尔型 | FALSE | | | |
| 16 | | %M101.0 | 布尔型 | FALSE | | | |
| 17 | "Tag_11" | %MW110 | 带符号十进制 | 0 | | | |
| 18 | "Tag_12" | %MW112 | Hex | 0000 | | | |
| 19 | | | | | | | |

图 5-11：MOBY 数据载体的存在状态

实际的 MOBY 数据载体放置到 RF260R 的检测范围之内后，RF260R 的状态指示灯有常绿变成橙色，如下图所示：

图 5-12：MOBY 数据载体的存在状态

在数据块“rfid_serial_write_A”写入十个字节的数据，如下图所示：




| | 名称 | 地址 | 显示格式 | 监视值 | 修改值 | 注释 |
|---|---|---|---|---|---|---|
| 1 | "rfid_serial_write_... | %DB168.DBB0 | Hex | AA | AA | |
| 2 | "rfid_serial_write_... | %DB168.DBB1 | Hex | BB | BB | |
| 3 | "rfid_serial_write_... | %DB168.DBB2 | Hex | CC | CC | |
| 4 | "rfid_serial_write_... | %DB168.DBB3 | Hex | DD | DD | |
| 5 | "rfid_serial_write_... | %DB168.DBB4 | Hex | EE | EE | |
| 6 | "rfid_serial_write_... | %DB168.DBB5 | Hex | FF | FF | |
| 7 | "rfid_serial_write_... | %DB168.DBB6 | Hex | 11 | 11 | |
| 8 | "rfid_serial_write_... | %DB168.DBB7 | Hex | 22 | 22 | |
| 9 | "rfid_serial_write_... | %DB168.DBB8 | Hex | 33 | 33 | |
| 10 | "rfid_serial_write_... | %DB168.DBB9 | Hex | 66 | 66 | |
| 11 | "rfid_serial_write_... | %DB168.DBB10 | Hex | AA | AA | |
| 12 | "rfid_serial_write_... | %DB168.DBB11 | Hex | AA | AA | |
| 13 | "rfid_serial_write_... | %DB168.DBB12 | Hex | AA | AA | |
| 14 | "rfid_serial_write_... | %DB168.DBB13 | Hex | AA | AA | |
| 15 | "rfid_serial_write_... | %DB168.DBB14 | Hex | 00 | | |
| 16 | "rfid_serial_write_... | %DB168.DBB15 | Hex | 00 | | |
| 17 | "rfid_serial_write_... | %DB168.DBB16 | Hex | 00 | | |
| 18 | "rfid_serial_write_... | %DB168.DBB17 | Hex | 00 | | |

图 5-13：在数据块“rfid_serial_write_A”写入十个字节的数据

编程把数据块“rfid_serial_write_A”的十个字节的数据写入到数据载体从地址 0 开始的十个字节中。

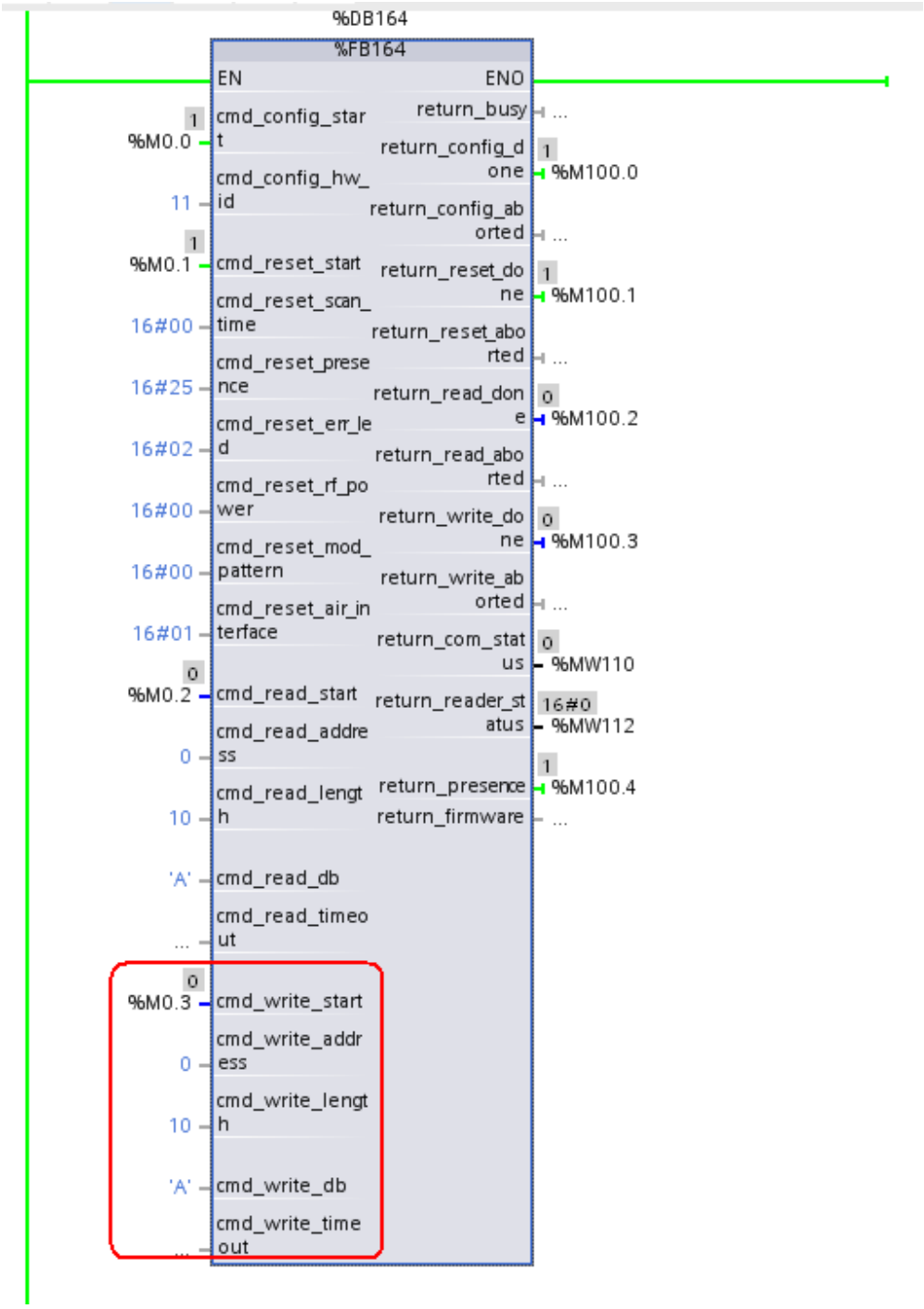



图 5-14：编程对数据载体写入十个字节的数据

进行写入操作，状态标示位 return_write_done 为 1，表示写入完成。

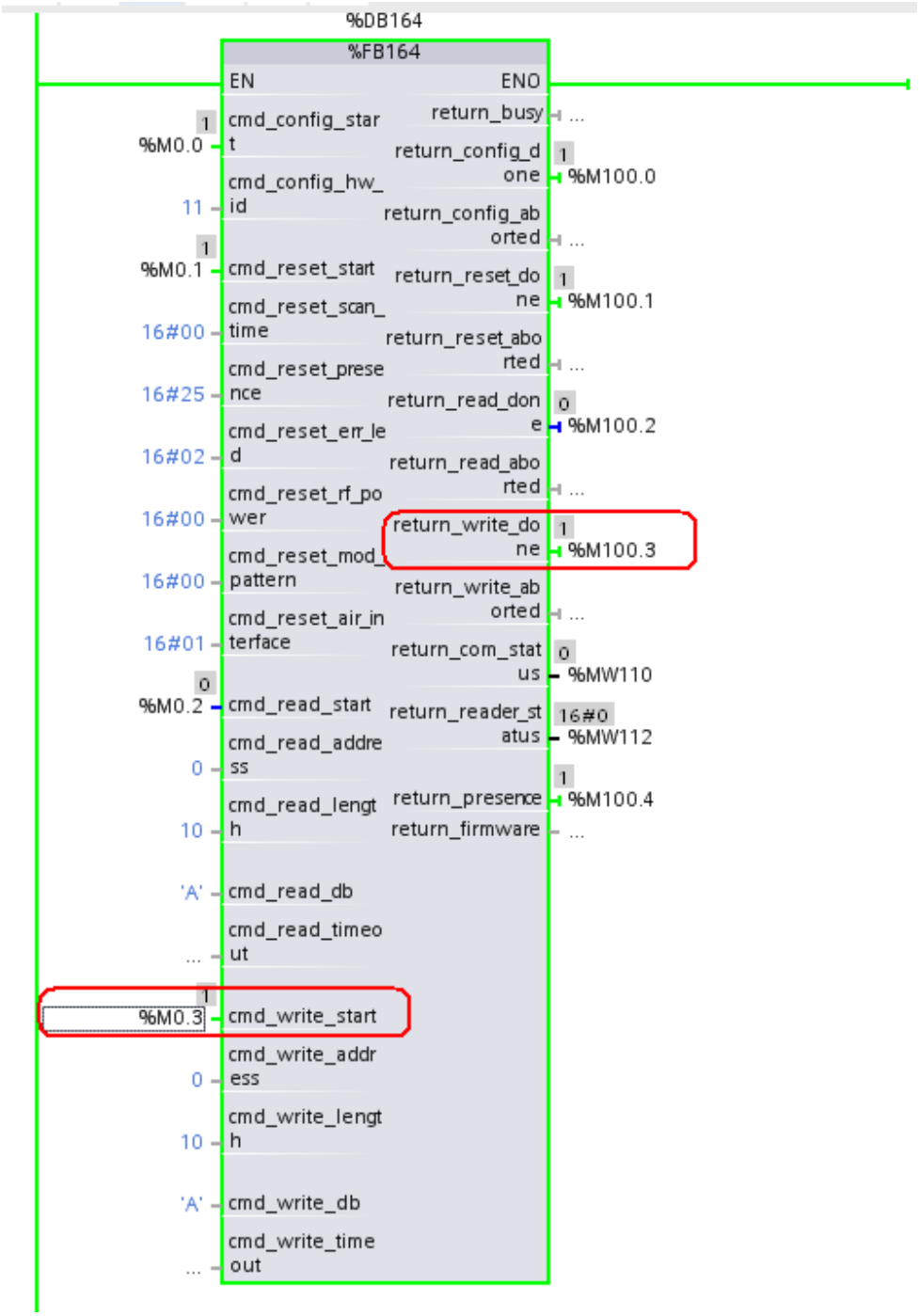



图 5-15：写入数据完成状态显示

5．5 编程从 RF260R 的数据载体读数据到 S7-1200 中

编程从 MOBY 数据载体的从地址 0 开始的十个字节读取到数据块“ rfid_serial_read_A ”中。

图 5-16：编程读取数据载体的数据

进行读取操作，读取操作完成后，return_read_done 完成标示位为 1，表示读取操作完成，如下图所示：

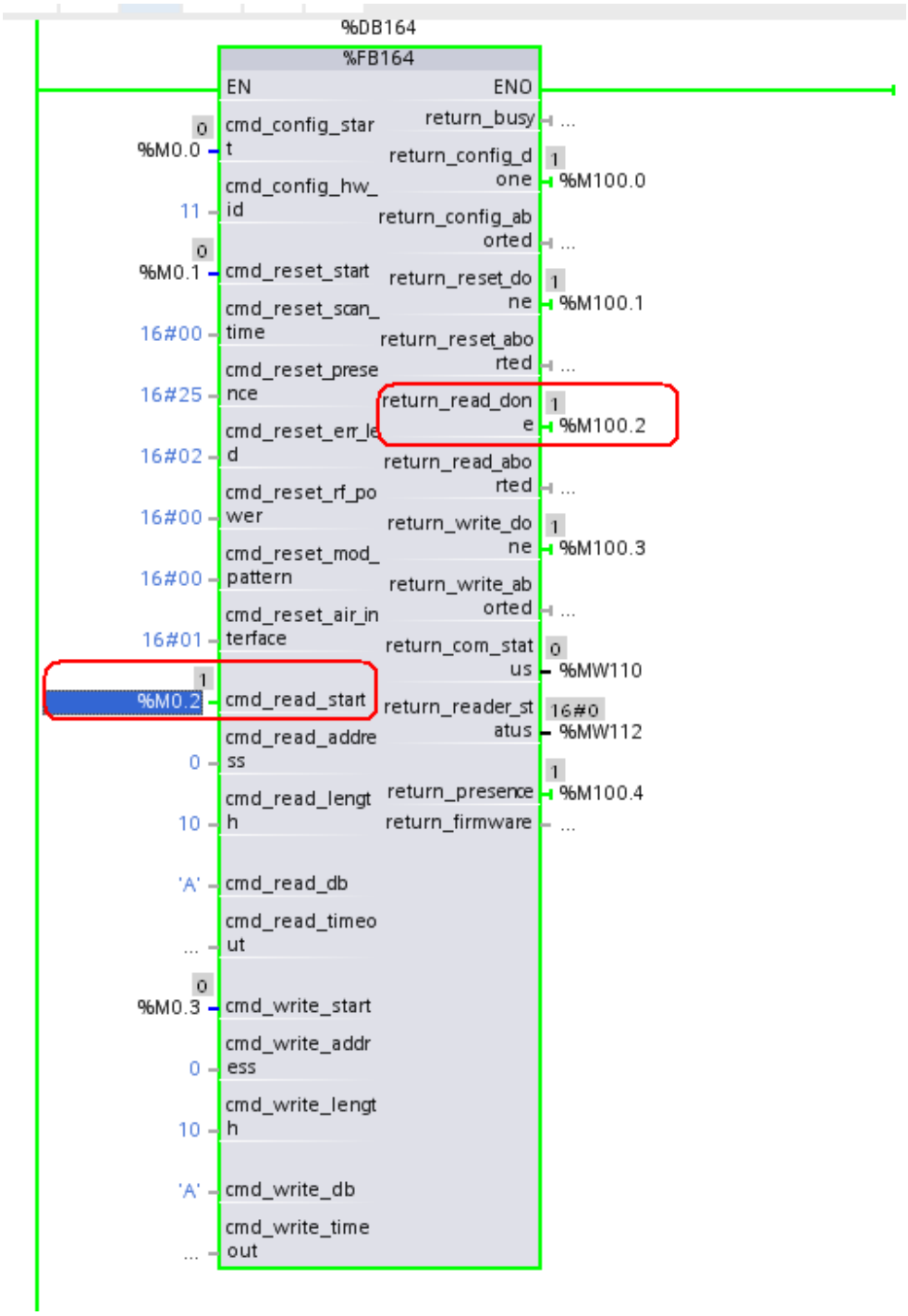
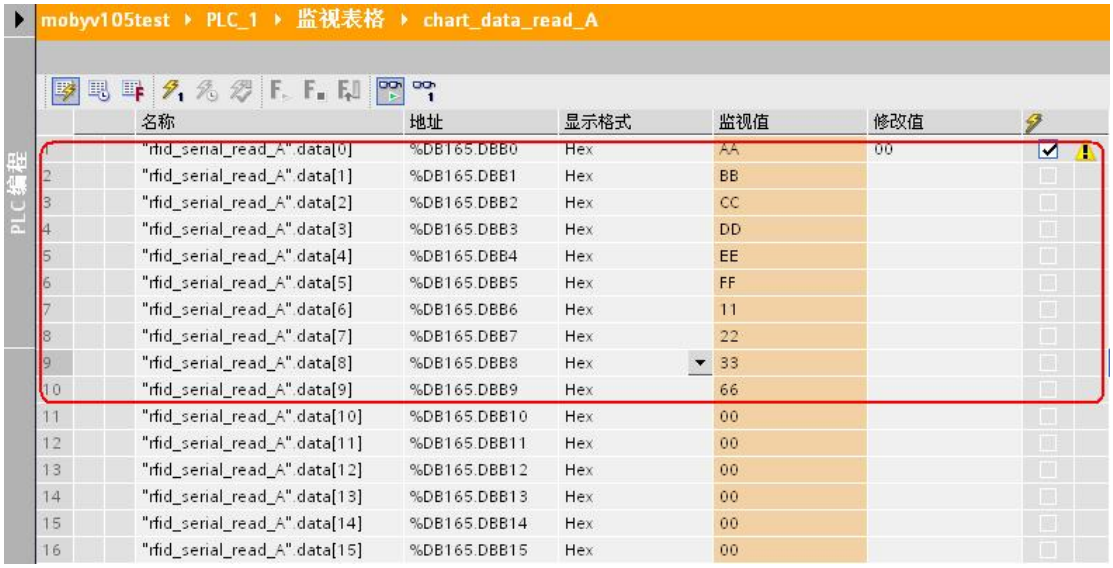



图 5-17：读取数据完成状态显示

在监控表中读取到的“rfid_serial_read_A”中的前十个数据即是 MOBY D 数据载体中的数据，可以看到和我们写入的数据是一致的，表示读取成功。如下图所示：




| | 名称 | 地址 | 显示格式 | 监视值 | 修改值 |
|---|---|---|---|---|---|
| 1 | "rfid_serial_read_A".data[0] | %DB165.DBB0 | Hex | AA | 00 |
| 2 | "rfid_serial_read_A".data[1] | %DB165.DBB1 | Hex | BB | |
| 3 | "rfid_serial_read_A".data[2] | %DB165.DBB2 | Hex | CC | |
| 4 | "rfid_serial_read_A".data[3] | %DB165.DBB3 | Hex | DD | |
| 5 | "rfid_serial_read_A".data[4] | %DB165.DBB4 | Hex | EE | |
| 6 | "rfid_serial_read_A".data[5] | %DB165.DBB5 | Hex | FF | |
| 7 | "rfid_serial_read_A".data[6] | %DB165.DBB6 | Hex | 11 | |
| 8 | "rfid_serial_read_A".data[7] | %DB165.DBB7 | Hex | 22 | |
| 9 | "rfid_serial_read_A".data[8] | %DB165.DBB8 | Hex | 33 | |
| 10 | "rfid_serial_read_A".data[9] | %DB165.DBB9 | Hex | 66 | |
| 11 | "rfid_serial_read_A".data[10] | %DB165.DBB10 | Hex | 00 | |
| 12 | "rfid_serial_read_A".data[11] | %DB165.DBB11 | Hex | 00 | |
| 13 | "rfid_serial_read_A".data[12] | %DB165.DBB12 | Hex | 00 | |
| 14 | "rfid_serial_read_A".data[13] | %DB165.DBB13 | Hex | 00 | |
| 15 | "rfid_serial_read_A".data[14] | %DB165.DBB14 | Hex | 00 | |
| 16 | "rfid_serial_read_A".data[15] | %DB165.DBB15 | Hex | 00 | |

图 5-18：监控表中读取到 MOBY D 数据载体中的数据

## 5．5 编程 RF260R 的数据载体和 S7-1200 的读写数据注意事项

1． 注意 S7-1200 的工作存储器的容量。
2． 注意读写的时间问题。
3． 注意读写的最大字节，每次进行读写最多 120 个字节。
4． 注意不同的数据载体的读写地址的区别。
5． 注意不同的读写器对应的数据载体的区别。
6． 注意自动与手动读写的区别。

## 6．S7-1200 与 RF260R 的通信的示例应用

S7-1200 与 RF260R 的通信应用，西门子提供了一个示例程序，在网上下载此压缩程序 CE-X16_S7-1200_RFID_serial_startup ，然后解压后即得使用的程序。

### 6．1 下载到 S7-1200 CPU 中

解压后使用的程序软件，如下图所示：

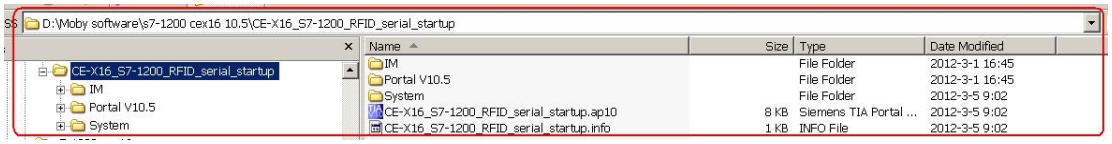



图 6-1：解压后使用的 RFID 示例应用程序

下载时注意 S7-1200 CPU 的类型和版本号，以及示例程序的版本号。

### 6．2 设置 PC/PG 接口连接

在控制面板中设置 PC/PG 的接口，打开控制面板中的 SET PG/PC interface

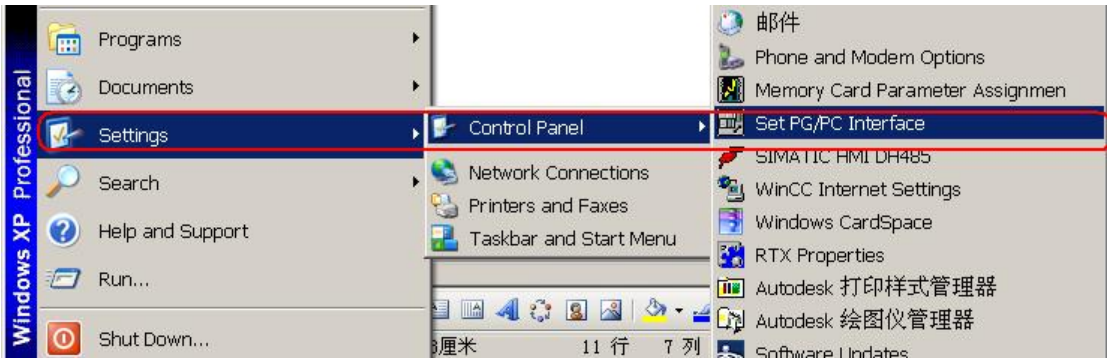



图 6-2：打开 PC/PG 的接口界面

在 PC/PG 的接口窗口中设置为 S7ONLINE (STEP7) ---→ TCP/IP，如下图所示：

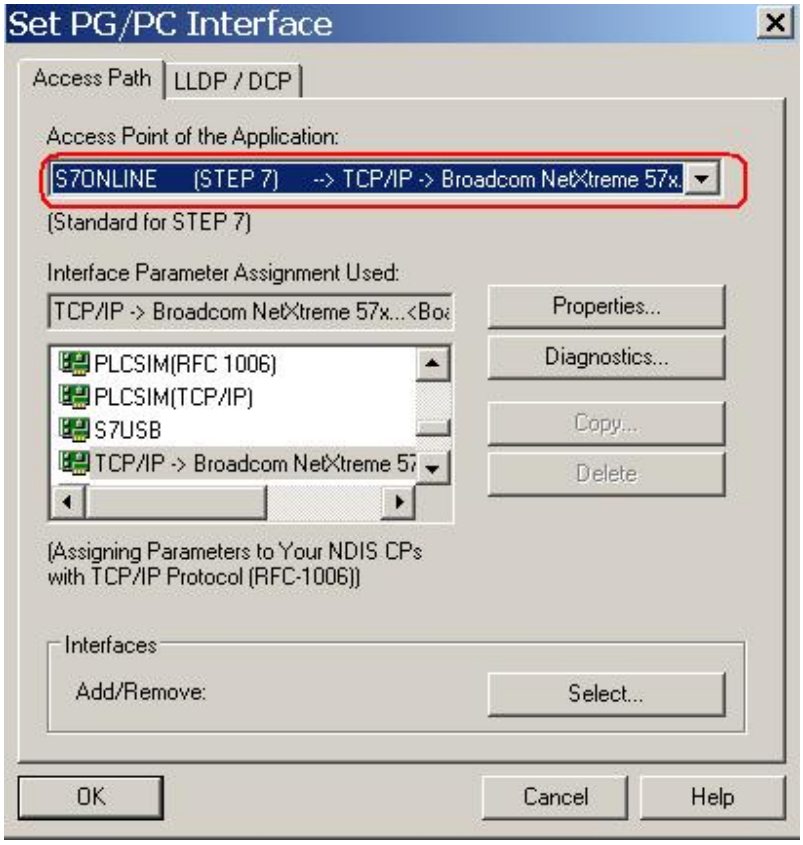



图 6-3：设置 PC/PG 的接口

## 6．3 运行 HMI runtime 程序

下载示例程序到 S7-1200 CPU 中，运行 S7-1200 CPU,此时点击 HMI 程序，运行工具栏上的 RT (RUNTIME) 按钮，如下图所示：

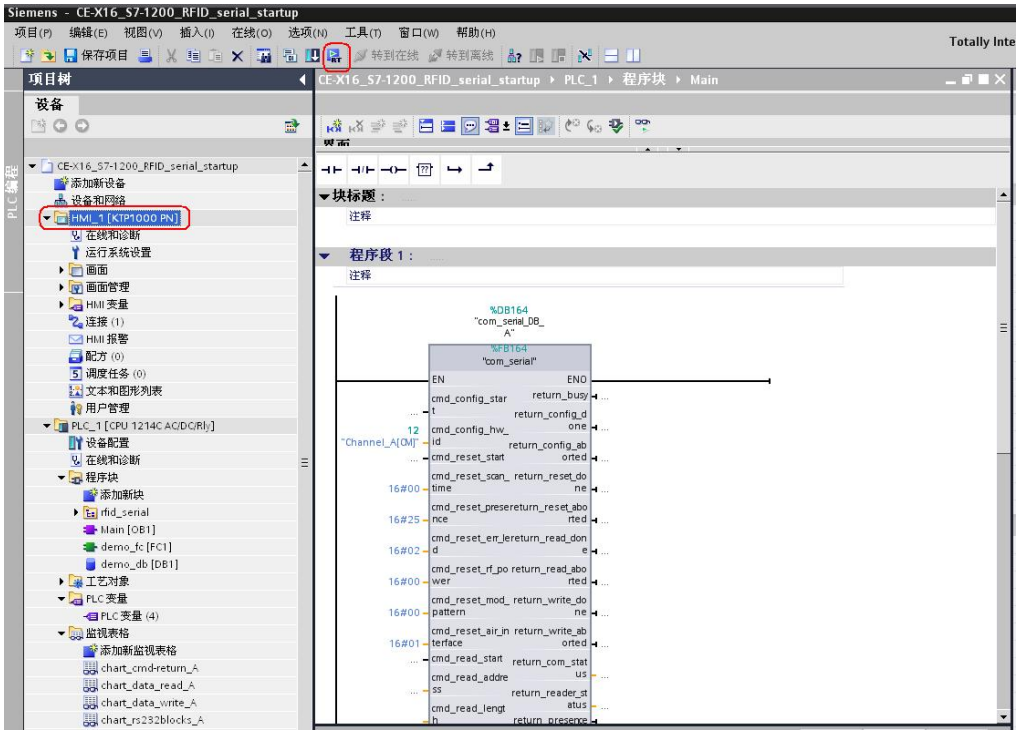



图 6-4：运行 HMI RUNTIME 程序

## 6．4 示例程序的运行界面

在运行了 RUNTIME 程序后，在窗口显示示例程序运行界面，如下图所示：

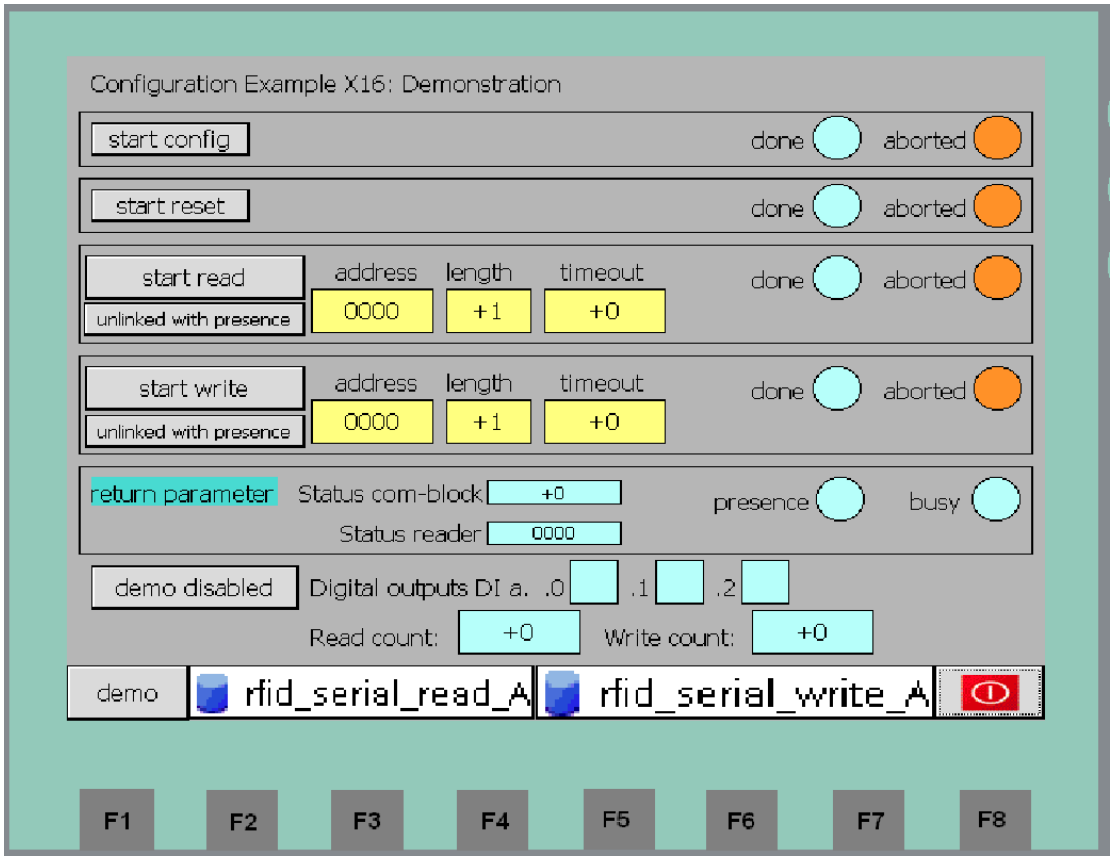



图 6-5： HMI RUNTIME 程序初始界面

6．5 在示例程序的运行界面进行读写 MOBY RF260R 读写器的读写操作

### 6.5.1 配置参数

在进行读写器的操作之前，必须配置参数，只有配置参数完成，才能进行下述的操作。配置完成后，出现配置完成标示，如下图所示：

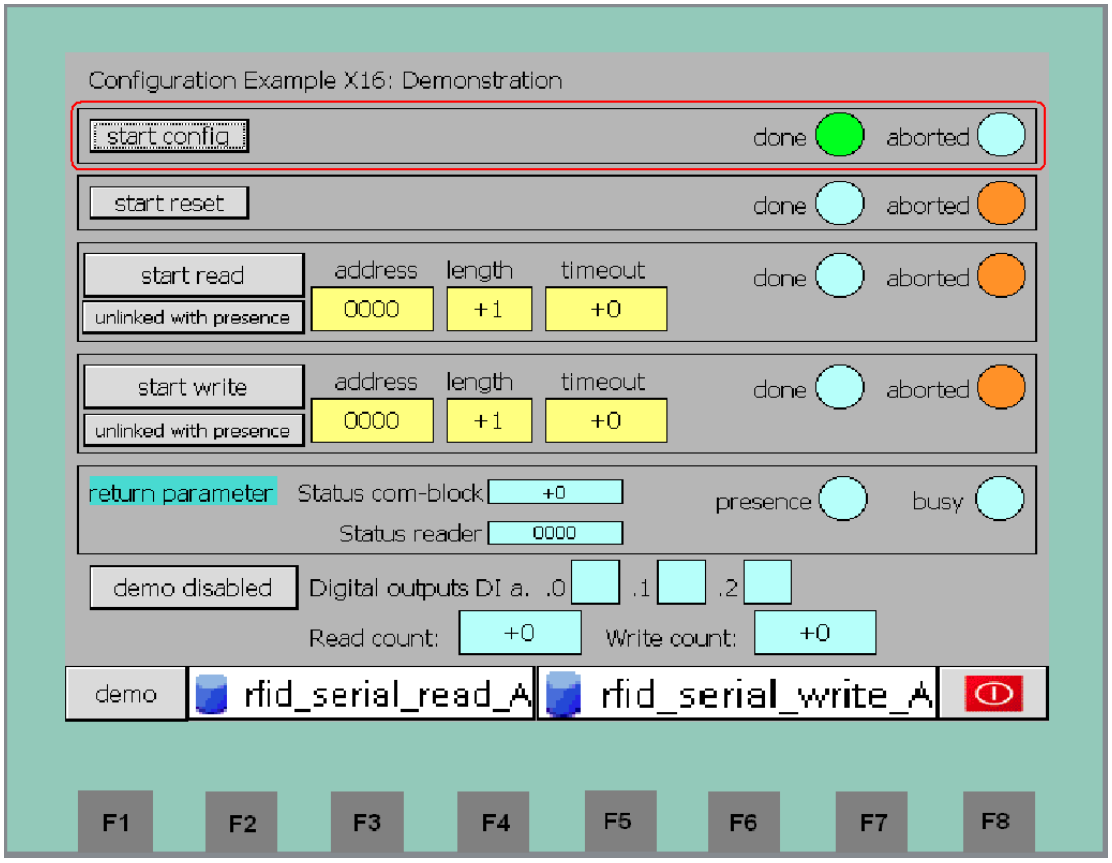



图 6-6： HMI RUNTIME 程序配置通信参数后界面

### 6.5.2 复位操作

配置参数完成后，进行 MOBY 读写器的初始化操作，初始化操作完成后，出现复位完成标示后，可以进行 MOBY 读写器的读写操作了，如下图所示：

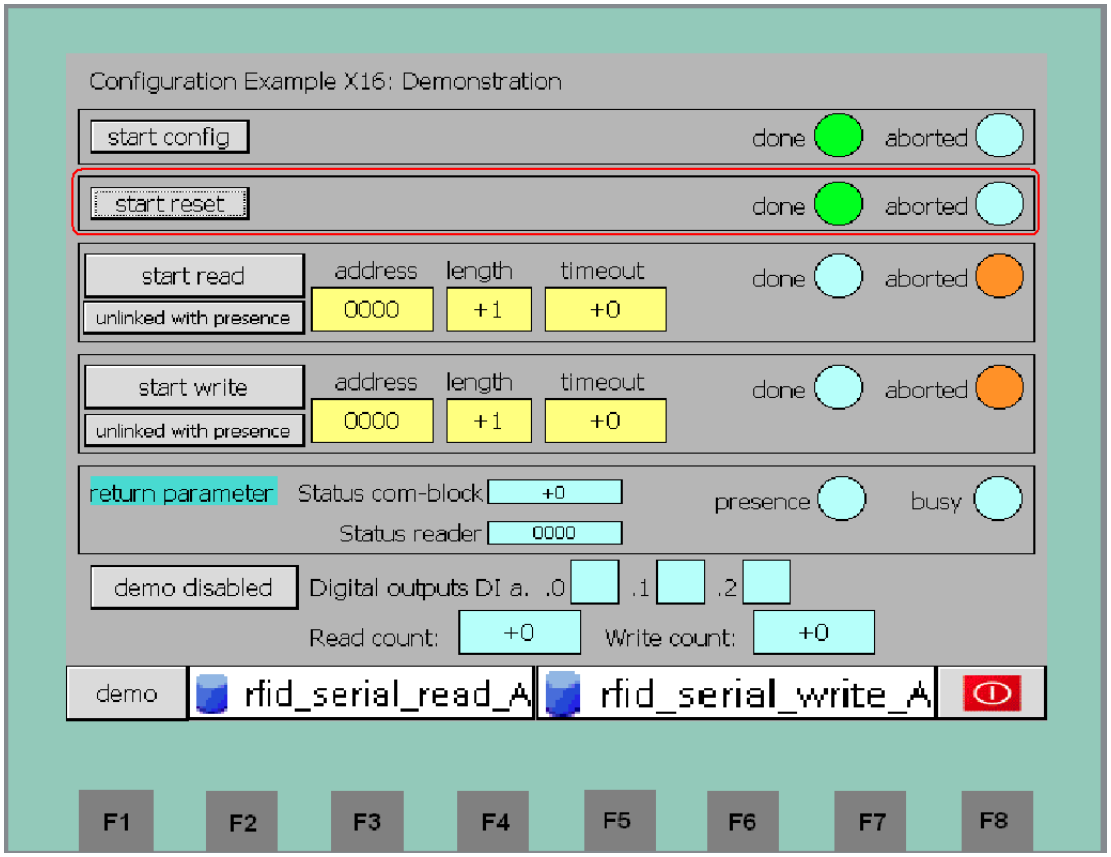



图 6-7： HMI RUNTIME 程序复位参数后界面

进行完复位操作后，MOBY 读写器 RF260R 上的指示灯，由每 0.5 秒闪烁一次变成常绿显示了。如下图所示：

图 6-8： RF260R 在复位参数后的 LED 状态指示

### 6.5.3 读取操作

从 MOBY 的数据载体 MDS D126 上读取数据到 S7-1200 CPU 中。

首先把 MOBY 的数据载体 MDS D126 放到 MOBY RF260R 的读取范围之内，此时我们在运行界面上可以看到 MOBY 的数据载体 MDS D126 的存在状态，如下图所示：

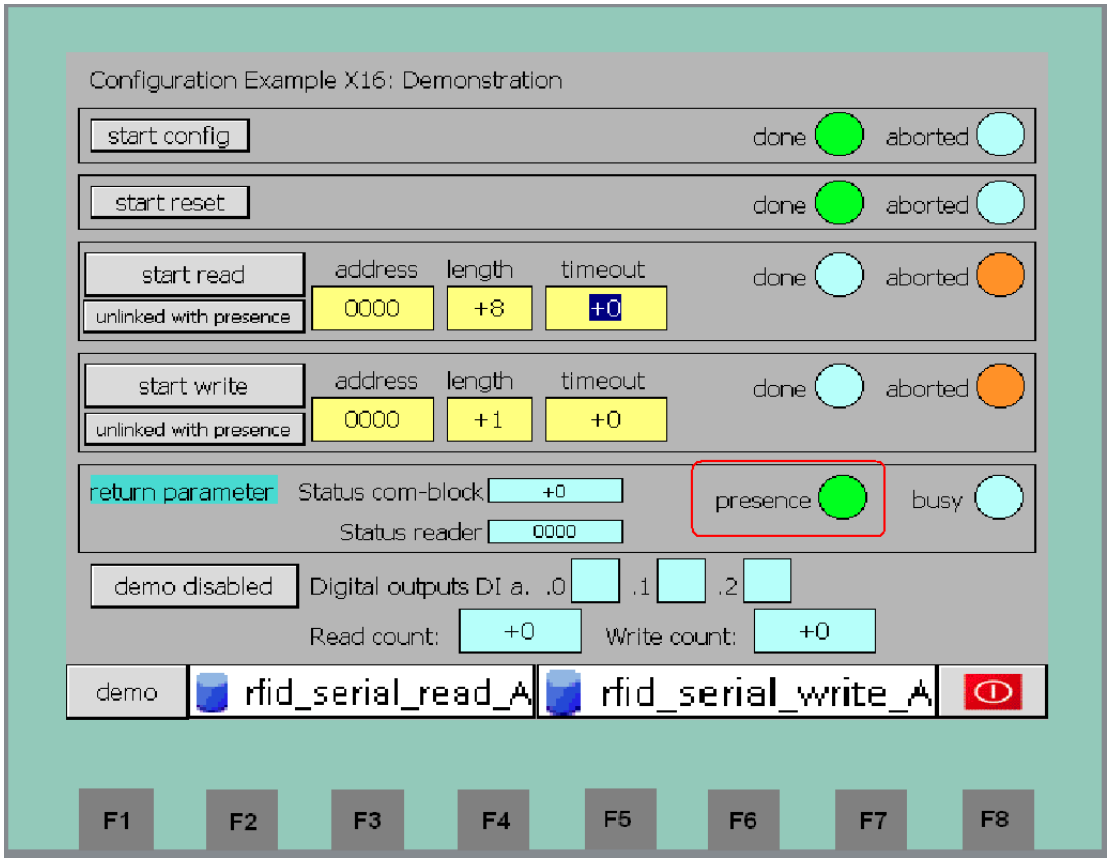



图 6-9： MOBY 的数据载体 MDS D126 的状态

点击 start read 进行读取操作，address 表示 MOBY 的数据载体 MDS D126 的起始地址，不同的数据载体，它的地址可能是不同的。读取完成后，有读取完成标示以及读取完成次数，如下图所示：

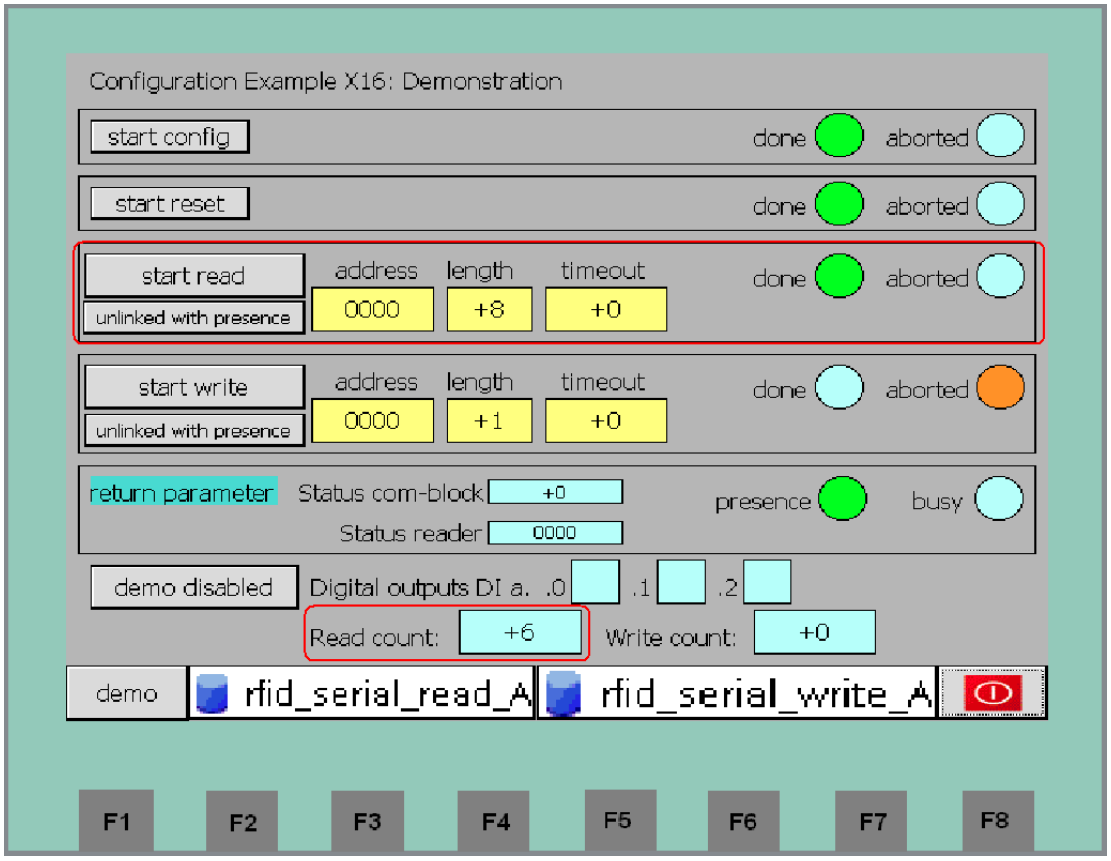



图 6-10： 从 S7-1200 读取完成 MOBY 的数据载体 MDS D126 的状态

点击 unlinked with presence,可以对 MOBY 的数据载体进行自动读取或手动读取的切换。

读取完成后，可以在 rfid_serial_read_A 中看到读取的 MOBY 数据载体上的数据，如下图所示：

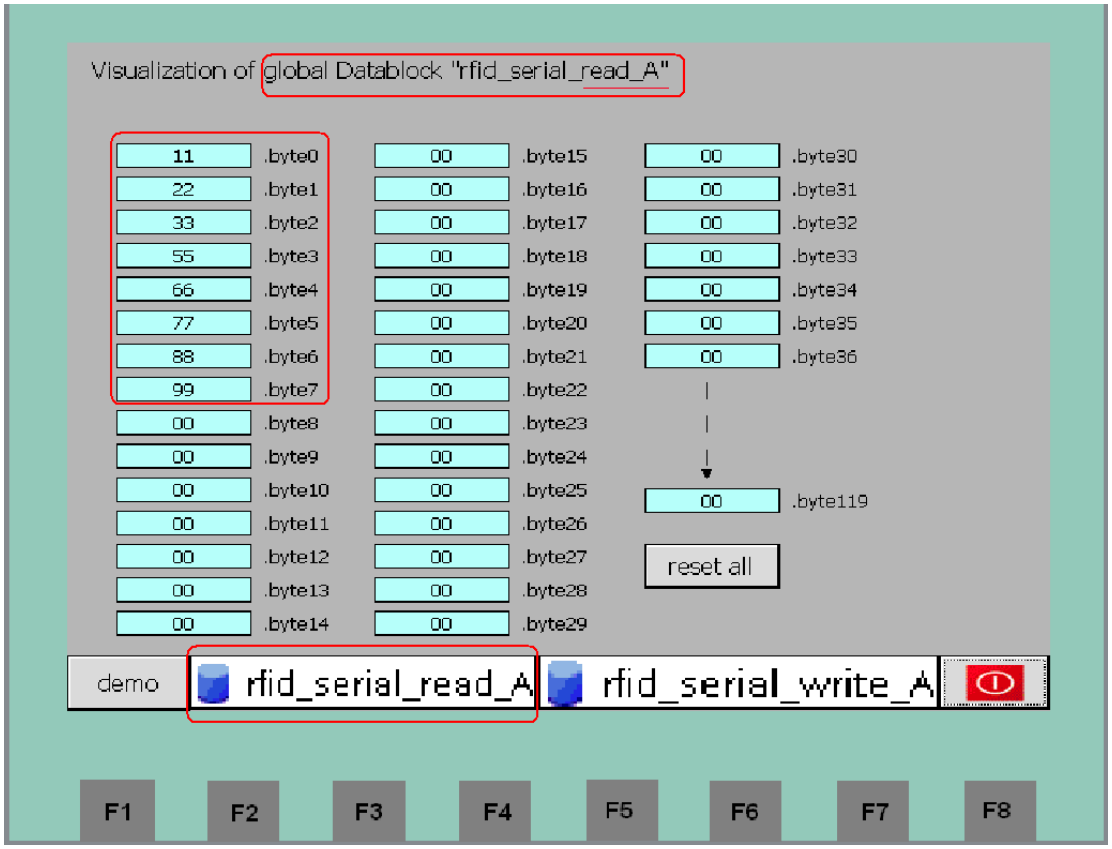



图 6-11：读取 MOBY 的数据载体 MDS D126 的数据

### 6.5.4 写入操作

从 S7-1200 CPU 中写数据到 MOBY 的数据载体中。

首先从 rfid_serial_write_A 中输入数据，这表示将要被写到 MOBY 数据载体的数据。如下图所示：

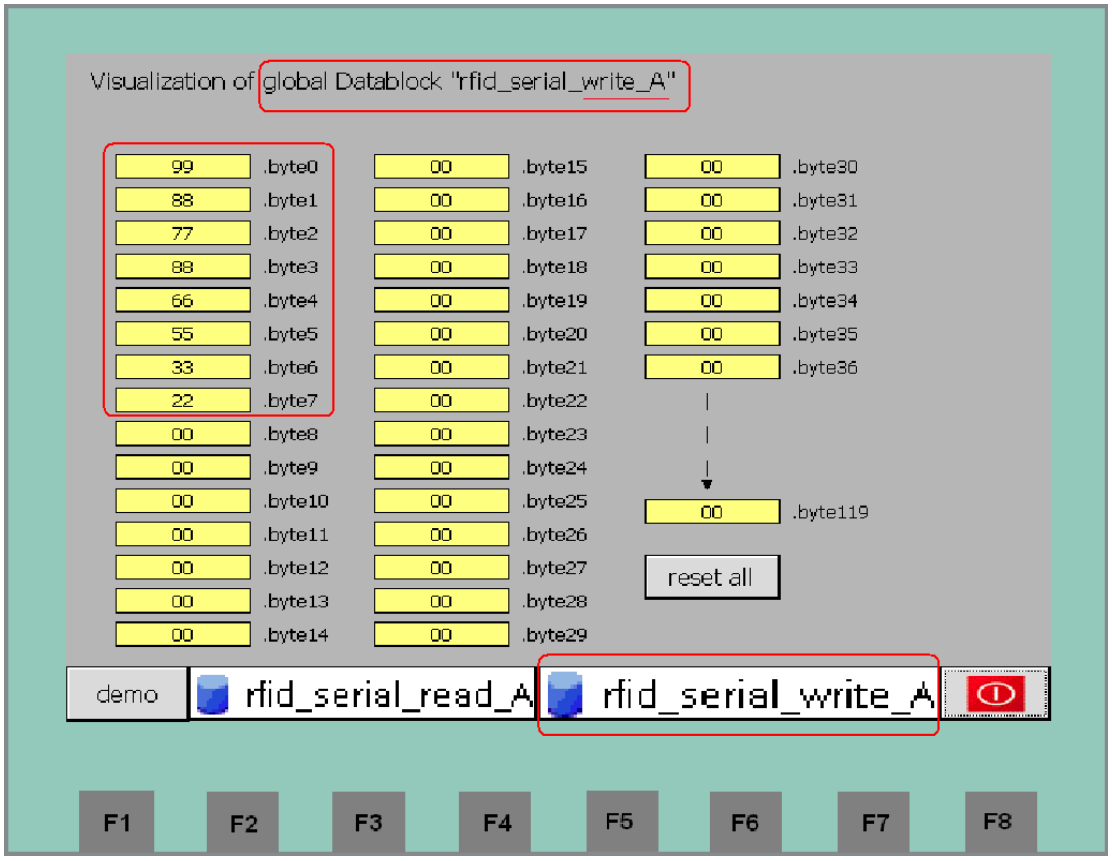



图 6-12：rfid_serial_write_A 中输入将要写入到数据载体的数据

点击 start write,写入数据到 MOBY 的数据载体中，写入完成后，有写入完成标示以及写入完成的次数，如下图所示：

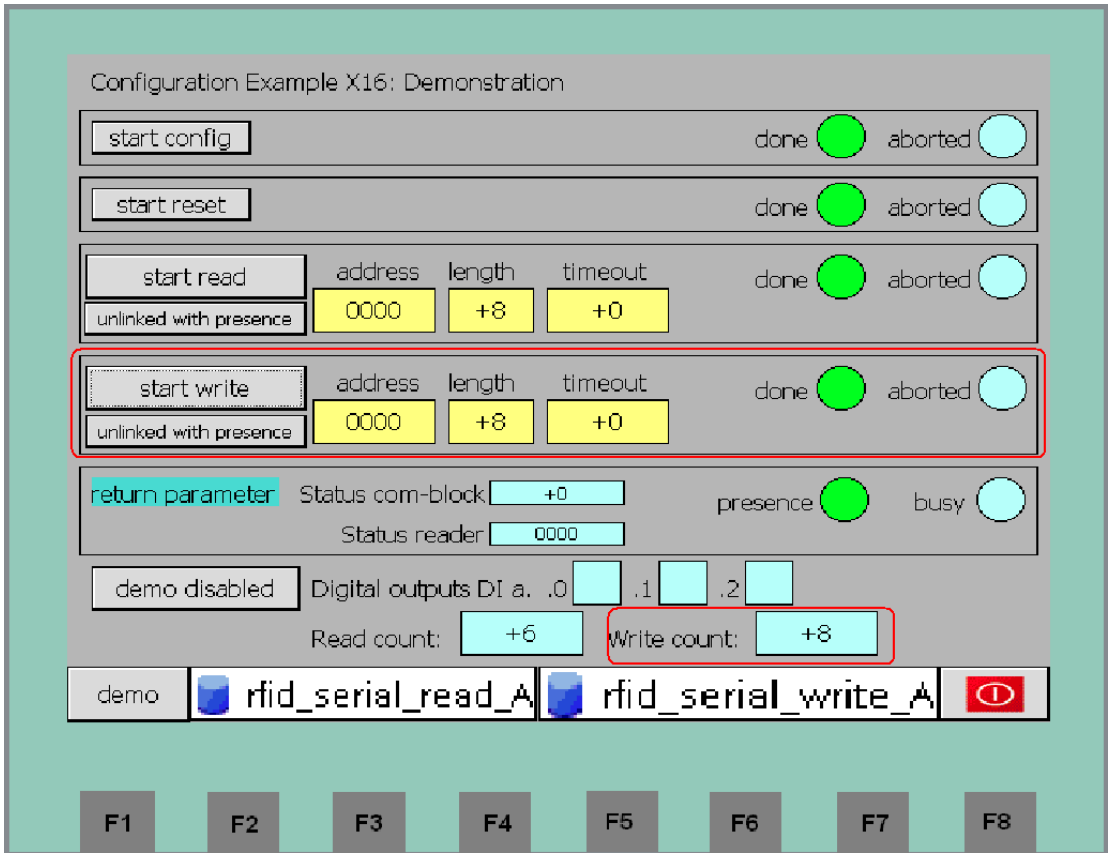



图 6-13：写入数据到数据载体完成状态

6.5.5 状态显示

1. Return parameter： 表示返回的状态
2. Com_block： 表示通信的状态
3. Reader： 表示读写器的状态
4. Demo disabled： 表示操作界面禁止
5. Digit outpts DI / DO ：表示 PLC 实际的输入/输出显示
6. Power button ： 表示关闭操作界面

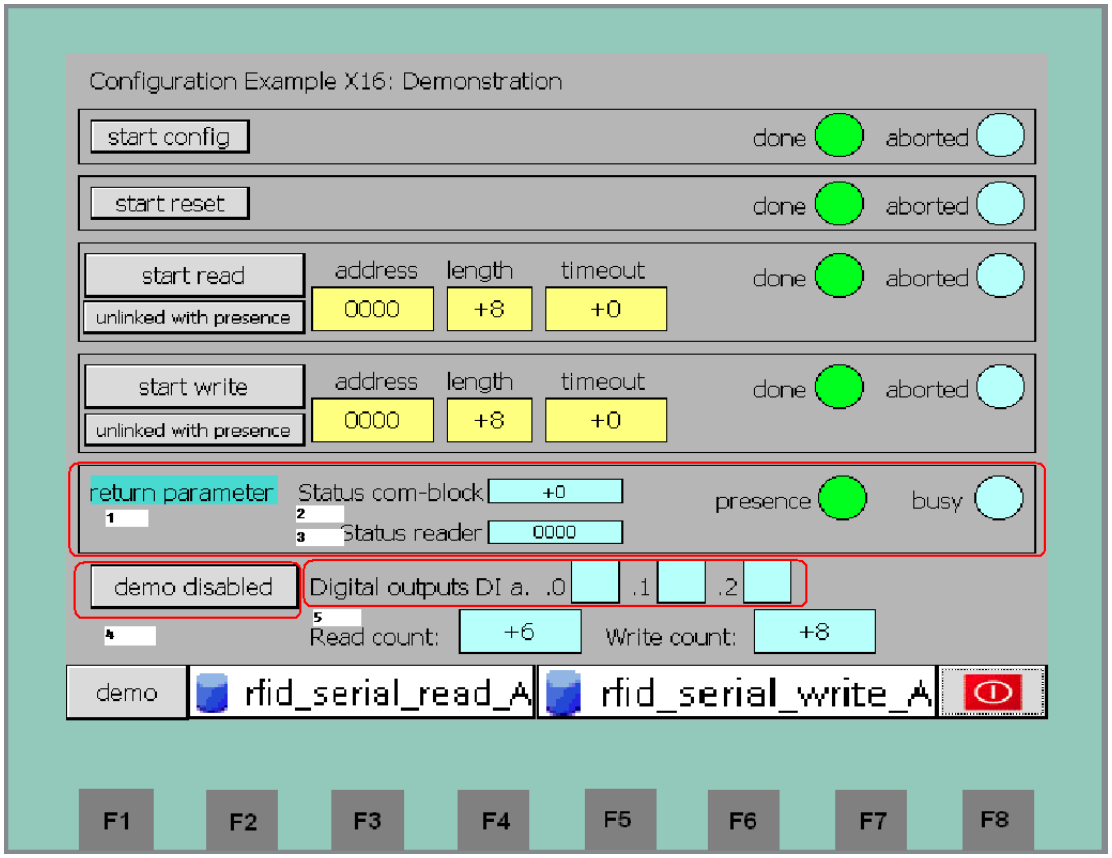



图 6-14：示例程序界面的显示状态

## 6．6　实际使用时的注意事项

在实际使用时，需要注意如下事项：

1．下载的 CPU 与该示例程序所使用的 CPU 是否相同类型。

2．下载的 CPU 与该示例程序所使用的 CPU 硬件版本是否相同。

3．下载的示例程序所使用的软件库版本与当前的 CPU 硬件版本是否相同。

4．所使用的 MOBY 读写器是否与示例程序中所使用的相同。

5．所使用的 MOBY 数据载体是否与示例程序中所使用的相同。

## 7. S7-1200 的硬件版本与 RFID 软件库之间的版本之间的对应关系

S7-1200 的硬件版本以及 STEP7 V11 与 RFID 软件库之间的版本关系如下表所示：

| 序号 | S7-1200 CPU 硬件版本 | STEP7 版本 | RFID 软件库版本 | 备注 |
|---|---|---|---|---|
| 1 | CPU 版本<br>V1.0.0 没有错误。 | STEP7 V10.5 | V2.3 | |
| 2 | CPU 版本<br>V1.0.0，没有错误。<br>V2.2.0，没有错误。 | STEP7 V11 SP2 | V2.4 | |
| 3 | V2.1.2 不能下载，<br>V2.0.2 不能下载，<br>V2.0.3 不能下载。 | STEP7 V11 SP2 | V2.4 | |

表 7-1：S7-1200 的硬件版本以及 STEP7 V11 与 RFID 软件库之间的版本关系

**注意：**

对于 S7-1200 CPU 下载 RFID 软件库时，会出现下述故障显示，中英文界面都是这种情况。如下图所示：

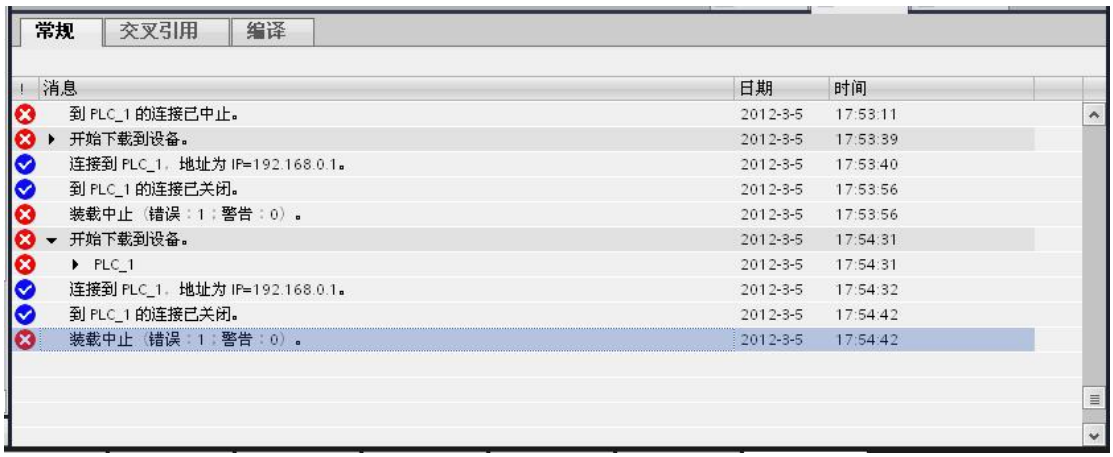



图 7-1：RFID 软件库下载时出现错误

在没有恢复工厂设置以前，会出现下述故障显示，如下图所示：

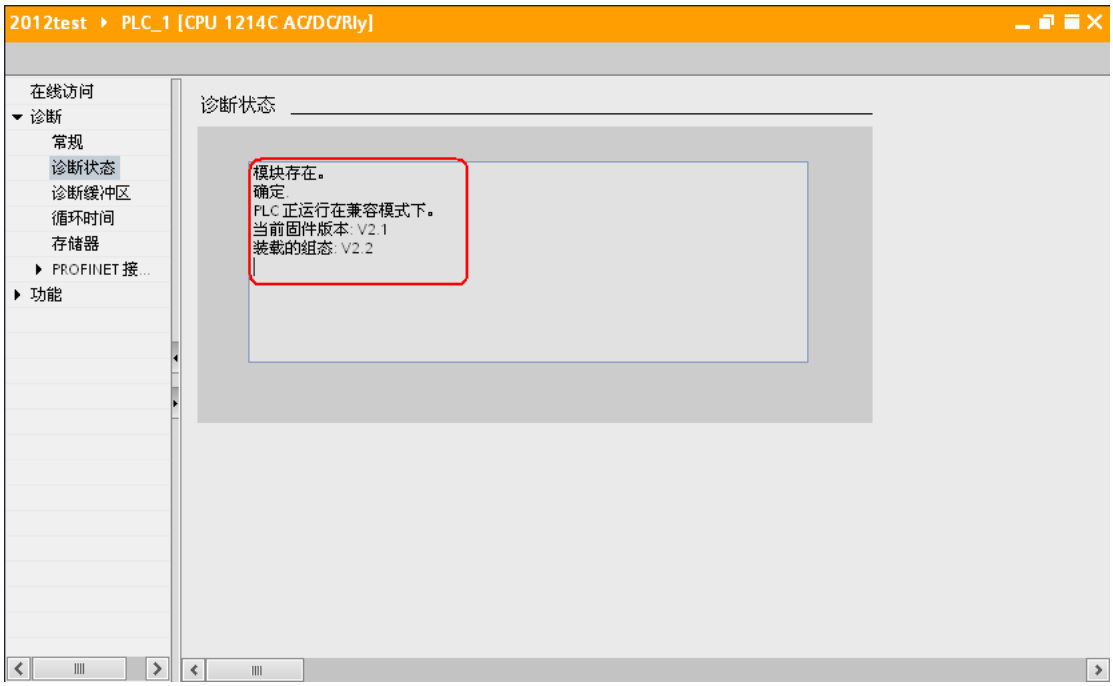



图 7- 2 ：恢复工厂设置后下载时在线诊断的状态

恢复工厂设置，重新下载，还是不能下载，诊断不出故障原因，

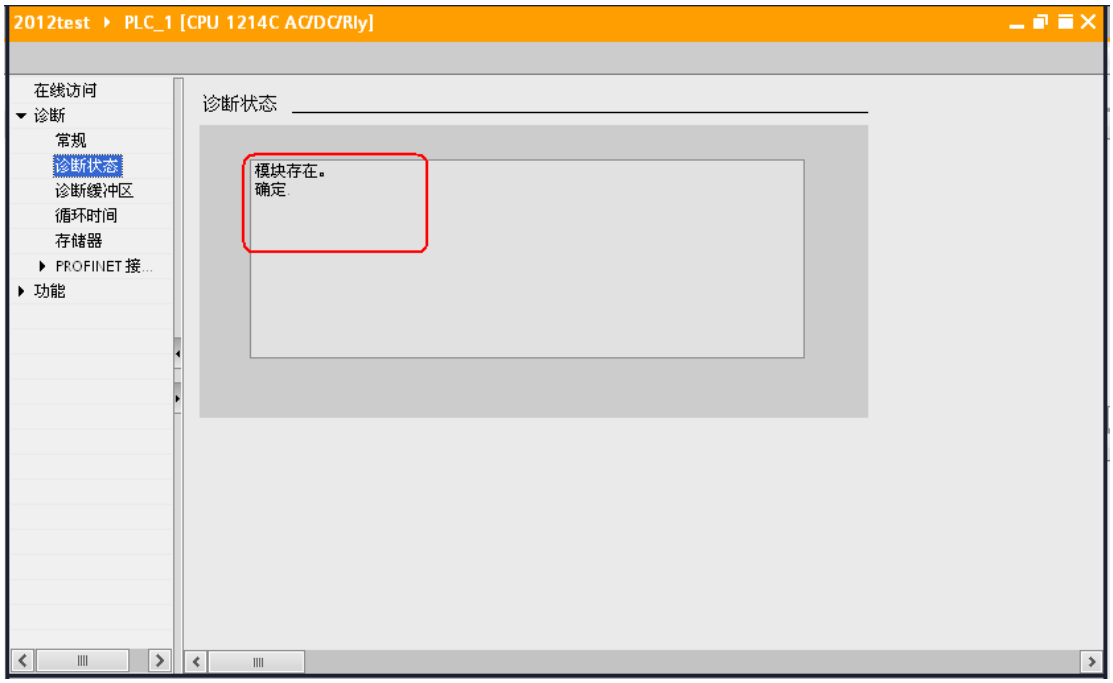



图 7- 3 ：恢复工厂设置后下载时在线诊断的状态

诊断时，可以看到ＣＰＵ的硬件版本与所配置的硬件版本一致，如下图所示：

图 7- 4 ：在线诊断的 CPU 的版本

对于 S7-1200 CPU V2.2.0 以前的版本，RFID 软件库有的可以下载，有的不可以下载，建议将 CPU 版本升级到 V2.2.0，V2.2.0 版本经过测试后，可以确认没有错误，可以编程，下载，调试。

在编程调试时，注意不同的 CPU,不同的操作软件，以及不同的软件库之间可能设置的参数也不尽相同，具体的设置要根据实际情况来做决定。

## 8. 参考手册

S7-1200 ,RF200,RF300,RFID 参考手册如下表所示：

| 序号 | 文档及链接 |
|---|---|
| /1/ | S7-1200 Manual<br>http://support.automation.siemens.com/WW/view/en/36932465 |
| /2/ | SIMATIC RF300 System Manual<br>http://support.automation.siemens.com/WW/view/en/21738946 |
| /3/ | 3964R Protocol<br>http://de.wikipedia.org/wiki/3964R |
| /4/ | RF300 command telegrams (attached on the same HTML page)<br>http://support.automation.siemens.com/WW/view/en/38471716 |
| /5/ | RFID Systems<br>RF600 command telegrams<br>3964R/LAUF (attached on the same HTML page)<br>http://support.automation.siemens.com/WW/view/en/38471716 |
| /6/ | SIMATIC RF200 System Manual<br>http://support.automation.siemens.com/WW/view/en/47189592 |