

常问问题 • 11 月/2014 年

S7-1500 串口模板自由口通讯编程注意事项

S7-1500 串口通讯模板，自由口通讯

目录

1	功能块调用注意事项	3
1.1	程序块的触发 (EN 和 REQ)	3
1.2	端口地址管脚参数设置 (PORT)	3
1.3	发送缓冲区参数设置 (BUFFER 和 LENGTH)	4
1.4	发送不同区域或不连续区域的数据.....	5
1.5	发送不同区域或不连续区域的数据.....	5

1 功能块调用注意事项

S7-1500 通过串口通讯模板 CM_P2P 可以实现自由口通讯。除了在硬件上会由于不同的硬件接口类型（232 和 422/485 等接口）而导致接线上有区别外，用户都需要调用相应的功能块来实现数据的发送和接收，这里介绍的是编程时需要注意的事项，否则可能会导致无法正常通讯。

1.1 程序块的触发（EN 和 REQ）

S7-1500 实现自由口通讯时，需要用户在程序中调用发送和接收功能块才能实现数据的收发，这两个功能块分别是“ Send_P2P”和“ Receive_P2P”。在调用这两个功能块时，需要注意，其使能端 EN 为“ 1”时，发送、接收块才可以被执行。而发送功能块还有一个 REQ 端，必须为上升沿触发，每次触发将发送一次，因此可以考虑在编程时将 REQ 的触发位设置为“系统时钟存贮位（例如:M0.0~M0.7）”（图 1-1），这样，将实现自动定时触发发送功能（图 1-2）。

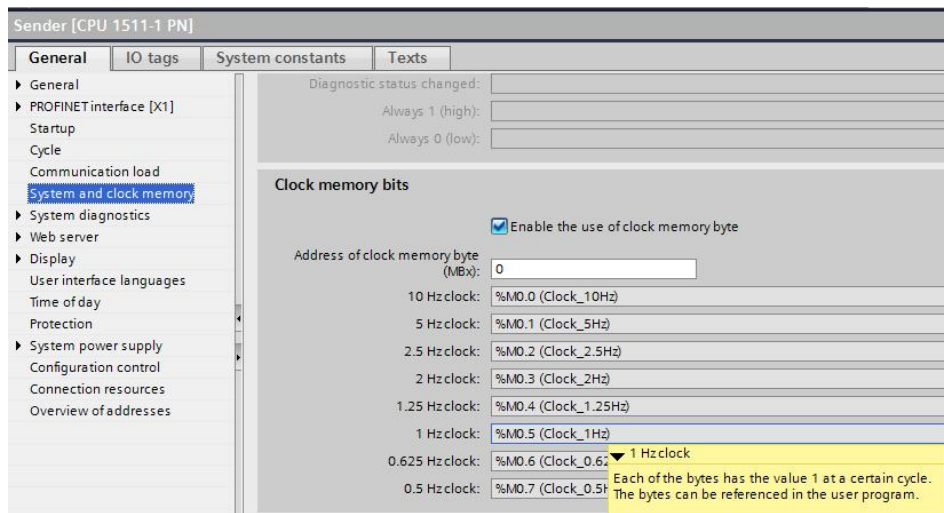


图 1-1 设置 CPU 的系统时钟存贮位

1.2 端口地址管脚参数设置（PORT）

发送、接收功能块需定义相应的串口地址，该地址是模板在组态时分配的物理地址（例如地址为：258，可在硬件组态中找到），另外在编程过程中，也可以通过 PLC Tag 表中的“系统 Tag”查找，然后直接用鼠标选中，拖拽到相应的管脚（图 1-2）。注意，这里的 Tag 是不能拷贝的，只能拖拽。

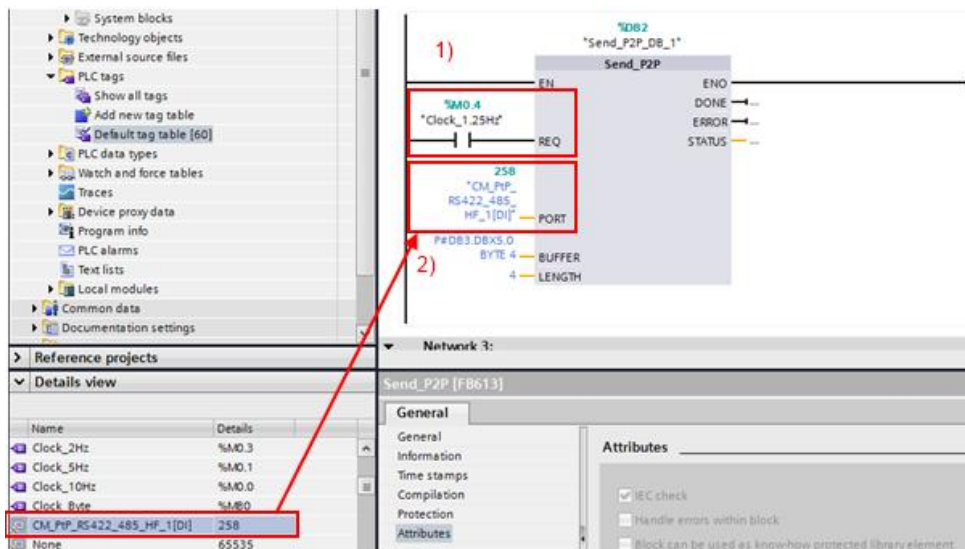


图 1-2 周期性触发及端口地址参数分配

1.3 发送缓冲区参数设置（BUFFER 和 LENGTH）

BUFFER 管脚定义的是自由口通讯所需要发送的数据区，该数据区在定义时，需注意：该地址区域的数据类型为 Variant（S7-1500）。

对于 S7-1500 来讲，默认情况下，DB 块属性都是“优化的块访问”，因此此时地址应写成该 DB 的符号名。例如，该 DB 的符号名定义为“SenderDB”，起始地址名为“Send_Data”，则该地址应写成：“SenderDB.Send_Data”（图 1-3）。当然，还是建议通过拖拽的方式，这样不容易出错。

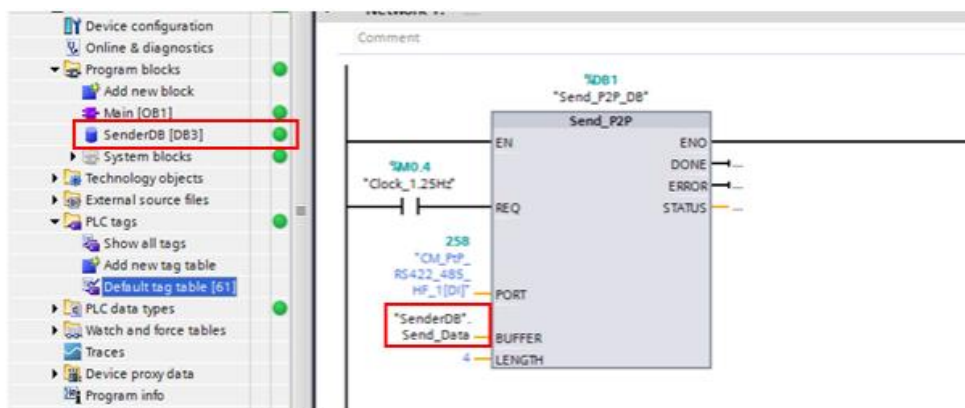


图 1-3 发送数据区的地址定义

但如果此时也想写成指针数据类型的方式（例如：P#DB3.DBX0.0 byte 20），则需要将 DB 块属性中的“优化块访问”的方式“去激活”（默认是激活的）（图 1-4）。

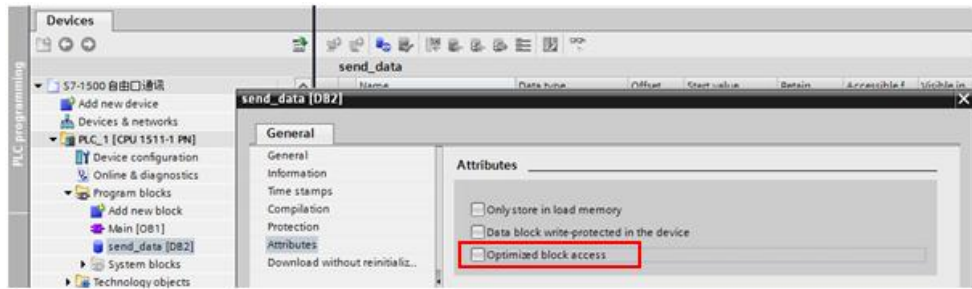


图 1-4 DB 块的“非优化访问”设置

而 LENGTH 为准备发送的数据长度，该长度应在 BUFFER 定义的数据区的长度范围内，比如，BUFFER 区定义了 20 个字节的长度范围，则 LENGTH 的取值范围为 0~20。如果定义为 0，则表示将整个数据区都发送出去。

1.4 发送不同区域或不连续区域的数据

特殊情况下，可能需要发送不同区域的数据，例如：同一个数据块的不同区域的数据。由于对于 S7-1500 系统的自由口通讯方式中，当多次调用发送块（Send_P2P）时，由于只有一个发送端口，因此当某个发送块执行时，其他块均停止发送，并且会导致接收方的数据被覆盖，因此并不推荐多次调用发送功能块来发送不同区域的数据。此时可以通过调用 MOVE 指令库中相应的 MOVE 指令将不同的数据区的数据先送到发送 BUFFER 区，再通过发送块发出去。

1.5 发送不同区域或不连续区域的数据

接收程序同发送程序类似，需要激活使能端“EN”、定义相应的端口，另外定义接收区也同发送端类似（图 1-5）。

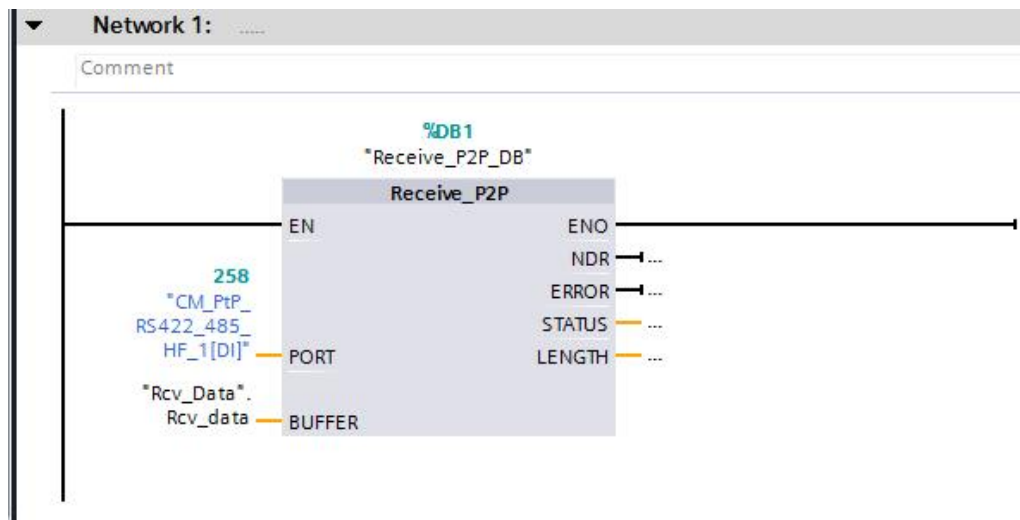


图 1-5 接收功能块的调用

但注意，如果发送端多次调用发送块（Send_P2P），且发送不同区域的数据，由于接收端无法改变接收端口，都是通过同一端口接收数据（例如：258），因此无论接

收端调用一次还是多次该接收功能块（Receive_P2P），所接收到的数据都将会被来自不同发送区域的数据所相互覆盖（图 1-6）。

Name	Data type	Start value	Monitor value
Static			
Rcv_data	Array[0..40] of Byte		
Rcv_data[0]	Byte	16#0	16#33
Rcv_data[1]	Byte	16#0	16#22
Rcv_data[2]	Byte	16#0	16#32
Rcv_data[3]	Byte	16#0	16#42

Name	Data type	Start value	Monitor value
Static			
Rcv_data	Array[0..40] of Byte		
Rcv_data[0]	Byte	16#0	16#E1
Rcv_data[1]	Byte	16#0	16#E2
Rcv_data[2]	Byte	16#0	16#E3
Rcv_data[3]	Byte	16#0	16#E4

图 1-6 接收到的数据相互覆盖

因此一般不推荐调用多次发送块，如果真的有这种应用，需用户自己编程将数据进行分拣。