

SIEMENS

PLC— PLC 之间的MPI 通信-----全局数据包通信方式
communication between PLCs through MPI network -----GD mode

Getting started

Edition 2007-03

<https://support.industry.siemens.com/cs/cn/zh/view/109481281>

摘要 本文简单介绍 S7-300/400 之间通过 MPI 网络使用全局数据包通信的设置

关键词 PLC、全局数据包、MPI

Key Words PLC、GD、MPI

目录

1 全局数据包通信简介	4
2 全局数据包的配置	4
3 多个 CPU 通信	6
4 通信的诊断	7
5 事件触发的数据传送	9

1 全局数据包通信简介

对于 PLC 之间的数据交换，只关心数据的发送区和接收区，全局数据包的通信方式是在配 PLC 硬件的过程中，组态所要通信 PLC 站之间的发送区和接收区，不需要任何程序处理，这种通信方式只适合 S7-300/300/400 PLC 之间相互通信。

下面将以举例的方式介绍全局数据包通信，使用的软硬件如下：

所需硬件：CPU315-2DP，CPU416-2DP

所需软件：STEP7 V5.2 SP1

2 全局数据包的配置

1) 首先打开编程软件 STEP7，建立一个新项目如 MPI_GD，在此项目下插入两个站 STATION1/CPU416-2DP，STATION2/CPU315-2DP，分别组态硬件，插入 CPU，配置 MPI 的站号和通信速率，在本例中 MPI 的站号分别设置为 2 号站和 4 号站，通信速率为 187.5Kbit/S。

这些工作完成以后，可以组态数据的发送区和接收区。点击项目名 MPI_GD 后出现 STATION1，STATION2 和 MPI 网，点击 MPI，再点击菜单“Options” ->

“Define Global Date” 进入组态画面如图 1、2:

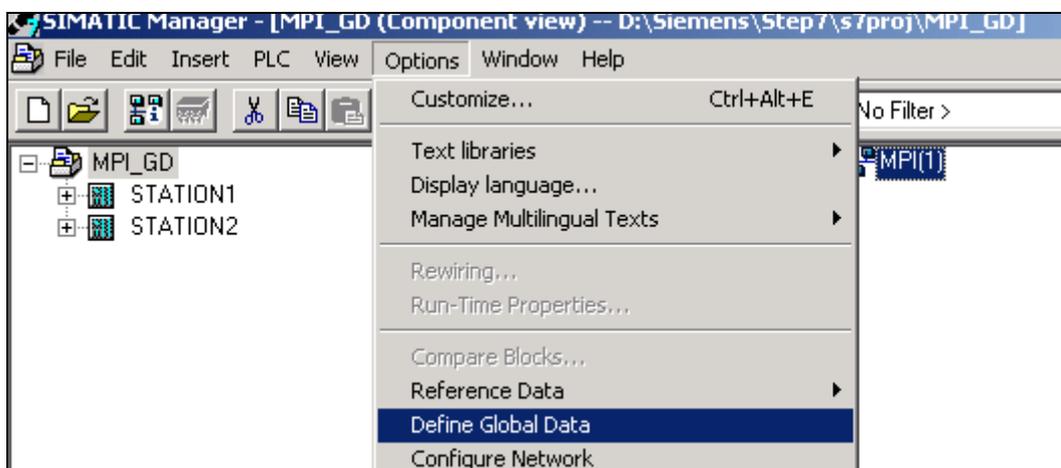


图 1 进入全局数据组态界面

	GD ID						
1	GD						
2	GD						
3	GD						
4	GD						
5	GD						
6	GD						
7	GD						
8	GD						
9	GD						
10	GD						
11	GD						
12	GD						

图 2 全局数据组态界面

2) 插入所有需要通信的 PLC 站 CPU

双击 GD ID 右边的 CPU 栏选择需要通信 PLC 站的 CPU。CPU 栏总共有 15 列，这就意味着最多有 15 个 CPU 能够参与通信。在每个 CPU 栏底下填上数据的发送区和接收区，例如：CPU416-2DP 的发送区为 DB1.DBB0~DB1.DBB21，可以填写为 DB1.DBB0: 22，然后在菜单“edit”选择“Sender”作为发送区。

开始地址 长度

CPU315-2DP 的接收区为 DB1.DBB0~21，可以填写为 DB1.DBB0: 22。编译存盘后，把组态数据分别下载到 CPU 中，这样数据就可以相互交换了。发送接收数据区的设置参考图 3:

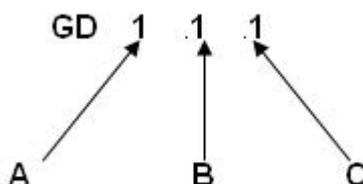
	GD ID	STATION2\ CPU 315-2 DP	STATION1\ CPU 416-2 DP
1	GD 1.1.1	DB1.DBB0:22	>DB1.DBB0:22
2	GD		
3	GD		
4	GD		
5	GD		

图 3 设置通信区

地址区可以为 DB, M, I, Q, 区, 长度 S7-300 最大为 22 个字节, S7-400 最大为 54 个字节。发送区与接收区应一致, 所以在上例中通信区最大为 22 个字节。

3 多个 CPU 通信

了解多个 CPU 通信首先要了解 GD ID, 编译以后, 每行通信区都会有 GD ID 号, 可以参考图 3 为



A: 全局数据包的循环数, 每一循环数表示和一个 CPU 通信, 例如两个 S7-300CPU 通信, 发送与接收是一个循环, S7-400 中三个 CPU 之间的发送与接收是一个循环, 循环数与 CPU 有关, S7-300CPU 最多为 4 个, 所以最多和 4 个 CPU 通信。S7-400CPU414-2DP 最多为 8 个, S7-400CPU416-2DP 最多为 16 个。

B: 全局数据包的个数。表示一个循环有几个全局数据包, 例如两个 S7 站相互通信一个循环有两个数据包。如图 4 所示:

	GD ID	STATION2\ CPU 315-2 DP	STATION1\ CPU 416-2 DP
1	GD 1.1.1	DB1.DBB0:22	>DB1.DBB0:22
2	GD 1.2.1	>MB20:20	MB20:20
3	GD		
4	GD		

图 4 一个循环两个数据包

C: 一个数据包里的数据区数

可以参考图 5: CPU315-2DP 发送 4 组数据到 CPU416-2DP, 4 个数据区是一个数据包。从上面可以知道一个数据包最大为 22 个字节, 在这种情况下每个额外的数据区占用两个字节, 所以数据量最大为 16 个字节。

	GD ID	STATION2\ CPU 315-2 DP	STATION1\ CPU 416-2 DP
1	GD 1.1.1	>DB1.DBB20	DB1.DBB20
2	GD 1.1.2	>MB20	MB20
3	GD 1.1.3	>QB0:10	QB0:10
4	GD 1.1.4	>DB1.DBW40	DB1.DBW40
5	GD		
6	GD		

图 5 一个数据包里的数据区数

对于 A,B,C 的介绍只是用于优化数据的接收区和发送区，减少 CPU 的通信负载，简单应用可以不用考虑这些，GD ID 编译后自动生成。

4 通信的诊断

在多个 CPU 通信时，有时通信会中断，是什么原因造成通信中断？我们编译完成后，在菜单“View”中点击“Scan Rates”和“GD Status”可以扫描系数和状态字。如图 6 所示：

	GD ID	STATION2\ CPU 315-2 DP	STATION1\ CPU 416-2 DP
1	GST	MD40	MD40
2	GDS 1.1	MD44	MD44
3	SR 1.1	225	2
4	GD 1.1.1	>DB1.DBB20	DB1.DBB20
5	GD 1.1.2	>MB20	MB20
6	GD 1.1.3	>QB0:10	QB0:10
7	GD 1.1.4	>DB1.DBB40	DB1.DBB40
8	GD 1.1.5	>DB1.DBB41	DB1.DBB41
9	GDS 2.1	MD48	MD48
10	SR 2.1	8	44
11	GD 2.1.1	>DB1.DBB42	DB1.DBB42
12	GD		

图 6 通信诊断

SR: 扫描频率系数。如图 6 SR1.1 为 225，表示发送更新时间为 $225 \times \text{CPU 循环时间}$ 。范围为 1~255。通信中断的问题往往设置扫描时间过快。可改大一些。

GDS: 每包数据的状态字（双字）。可根据状态字编写相应的错误处理程序，结构如下：

第一位：发送区域长度错误。

第二位：发送区数据块不存在。

第四位：全局数据包丢失。

第五位：全局数据包语法错误。

第六位：全局数据包数据对象丢失。

第七位：发送区与接收区数据对象长度不一致。

第八位：接收区长度错误。

第九位：接收区数据块不存在。

第十二位：发送方从新启动。

第三十二位：接收区接收到新数据。

GST: 所有 GDS 相“OR”的结果

如果编程者使用 CP5511/5611 编程卡可以首先诊断一下连线是否可靠，如上例中 S7-300 MPI 地址是 2，S7-400MPI 地址是 4，用 CP 卡连接到 MPI 网上（PROFIBUS 接头必须有编程口）可以直接读出 2,4 号站，在“控制面板”->“PG/PC interface”->“Diagnostics”->“read”读出所以网上站号，如图 7 所示：

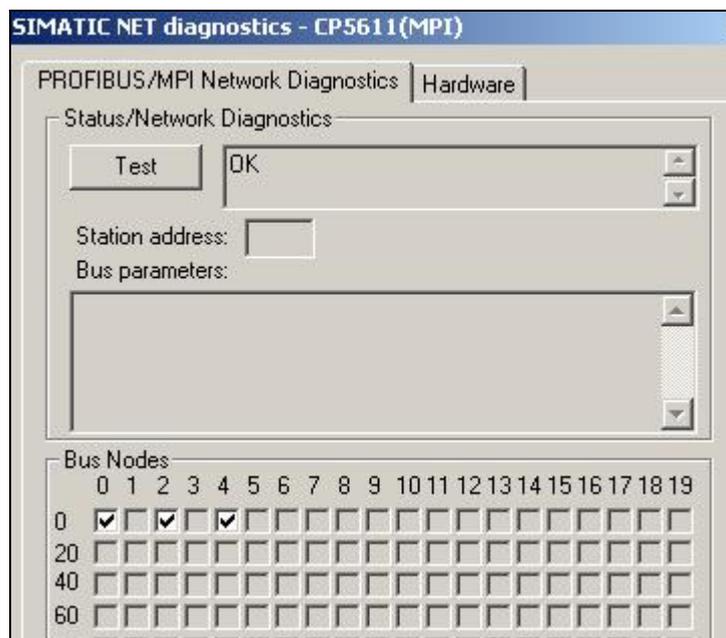


图 7 使用网卡诊断站号

0号站位 CP5611 站号，如果没有读出 2，4 号站，说明连线有问题或 MPI 网传输速率不一致，可以把问题具体化。

5 事件触发的数据传送

如果需要控制数据的发送与接收，在某一事件，某一时刻，接收和发送所需要的数据，这时将用到事件触发的数据传送。这种通信方式通过调用 CPU 的系统功能 SFC60 (GD_SND)，SFC61(GD_RCV)来完成的，而且只有 S7-400CPU 支持，并且必须设置 CPU 的 SR（扫描频率）为 0，可参考图 8 全局数据的组态画面：

	GD ID	STATION2\ CPU 315-2 DP	STATION1\ CPU 416-2 DP
1	SR 1.1	8	0
2	GD 1.1.1	>MB20:10	MB40:10
3	SR 1.2	8	0
4	GD 1.2.1	MB60:10	>MB60:10
5	GD		

图 8 在 S7-400CPU 侧的 SR 为 0

与上面作法相同编译存盘后下载到相应的 CPU 中，然后在 S7-400 侧中调用 SFC60/61 控制接收与发送。

所需硬件：CPU315-2DP，CPU416-2DP。

所需软件：STEP7 V5.2 SP1

```

A      M      1.1
FP     M      1.2
=      M      1.3
A      M      1.3

JCN    M1

CALL  "GD_RCV"
      CIRCLE_ID:=B#16#1           //循环数
      BLOCK_ID :=B#16#1           //数据包数
      RET_VAL  :=MW4

M1:   A      M      1.0
      FP     M      1.4
      =      M      1.5
      A      M      1.5
      JCN    M2

CALL  "GD_SND"
      CIRCLE_ID:=B#16#1
      BLOCK_ID :=B#16#2
      RET_VAL  :=MW2

M2:   SET
      R      M      1.1
      R      M      1.0

```

CIRCLE_ID, BLOCK_ID 参考 GD A。B。C。

例子中 M1.1 为 1 时 CPU416 接收 CPU315 的数据 MB20~MB29 放到 MB40~MB49 中，
M1.0 为 1 时 CPU416 发送数据 MB60~MB69 到 CPU315MB60~MB69 中