Procedure for Configuring GD Communications

General:

GD Communications permit

- **Cyclic data exchange between CPUs via the MPI interface (or via the backplane bus of the S7-400)**. Cyclic data exchange works with the normal process image.
- **Event-controlled data transfer S7-400**. The time of the data exchange is determined by the call of system functions (SFC) in the user program. The SFC 60 "GD_SND" (global data send) is available for sending global data and the SFC 61 "GD_RCV" (global data receive) for receiving it. If there is to be just one event-controlled data transfer, then you must enter the reducing factor "0" in the GD table. If there is a value greater than "0" entered, the global data is transferred both cyclically and event-controlled.

The exchange of global data in the **cyclic mode** is as follows:

- The Send CPU sends the global data at the end of a cycle.
- The Receive CPU reads the data at the beginning of a cycle.

➢ ***Note:***
In the following is a summary of how to configure cyclic GD communications with STEP 7. For the most part it contains excerpts from the STEP 7 online help. There you will also find detailed descriptions of the individual configuring steps for GD Communications.

Requirements for Configuring GD Communications:

You have configured an MPI subnetwork with all the necessary stations.

Observe the following rules for assigning MPI addresses MPI subnetwork:

- Assign the MPI addresses in ascending order.
- CPUs are supplied with the default address 2. Since you may only use this address once, you must change the default address in all the other CPUs.
- Reserve the MPI address 0 for a PG.
- You can link together up to 126 (addressable) users in an MPI subnetwork; up to 8 users with a transfer rate of 19.2 kbit/s.
- All the MPI addresses of an MPI subnetwork must be different.

Only when all the modules of a subnetwork have different addresses and your actual setup matches that of the network configuration, should you load the settings via the network.

- **Special feature: MPI addresses of FMs and CPs (S7-300)**

When planning the MPI addresses for several CPUs, you must provide "MPI address gaps" for FMs and CPs with their own MPI addresses in order to avoid double assignment of addresses. Your MPI address is determined automatically by the CPU and assigned as follows:

- First CP / first FM after the CPU: MPI address of the CPU + 1
- Second CP / second FM after the CPU: MPI address of the CPU + 2

Newer CPUs of the S7-300 (see manual or product information) permit the free MPI address assignment for such CPs and FMs (can be set via the "General" tab of the module).

Other rules for setting up a network are to be found in the manuals for setting up a SIMATIC 300 or SIMATIC 400.

Step 1: Determining the Communications Capacity of the GD Resources

When starting to configure a GD communication you have to check out a number of things with regard to the communications capacity of your CPUs. You can determine the GD communications capacity of an S7 CPU from the following technical data ("**GD Resources**"):

- Number of GD circles (to which the CPU can belong)
- Max. amount of net data per GD package
- Max. number of Receive GD packages per GD circle
- Length of the consistent data per package

The above-mentioned technical data also provides information indirectly on how much data can be exchanged cyclically by the CPUs that are linked via an MPI subnetwork or via backplane bus of the S7-400. The illustrations in the following sections show clearly how the send data is "tied together" into GD packages and how many GD circles are "consumed" for this.

1.1 Determining the number of GD packages required

A GD package is a message that is sent "in one go" from exactly one CPU to one or more other CPUs.

A GD package contains at most the following amount of net data (see also the technical data of the CPUs):

- Max. 22 bytes with S7-300
- Max. 54 bytes with S7-400

Rules

- If you want to send not only from one operand area, then from the maximum amount of net data you must subtract two bytes per supplementary operand area used. ***For this, read through the examples in the STEP 7 Online Help.***
- One bit operand (e.g. M 4.1) "consumes" one byte of net data in the GD package.

1.2 Defining the number of GD circles required

- All CPUs that are involved in the exchange of a **common** data package as Sender or Recipient "consume" one GD circle.
- If more data than "fits" into one GD package is to be sent and received, then an additional GD circle is "consumed".
- An additional GD circle is also "consumed" if the Send and Receive CPUs **are not the same** (then a new GD package will have to be "tied up").

Examples for determining the number of GD packages and GD circles are given in your STEP 7 Online Help.

Step 2: Starting Global Data Configuring in NetPro

There are two ways of opening a GD table:

- Open the complete GD table of a subnetwork
- Open the GD table of a CPU

In general, global data configuring is started by opening the complete GD table of a subnetwork. This is why only this case is dealt with here.

2.1 Opening the GD table

You have opened **NetPro**.

1. In Network View you select an MPI subnetwork for which you want to configure global data communications.
2. Select the menu command **Tools > Define Global Data**.

Result:
The GD table for the MPI subnetwork is opened.

2.2 Filling in the GD table

Requirement

The view for the reducing factor and GD status is deactivated in the GD table.

Entering CPUs in the table header

1. In the GD table click a column in the table header. This marks the selected column.
2. Select the menu command **Edit > CPU**. The "Open" dialog box opens. You can also open this dialog box by double-clicking the column head.
3. Select your current project and with a double-click open the station in which the desired CPU is located.
4. Select the CPU and acknowledge your choice by clicking the "OK" button.

Result:
The name of the selected CPU is displayed in the table header.

Entering data in GD lines

Requirement: you have entered the CPU concerned in the table header (see above).

1. Position the insert marker in a table field and enter the required operand. You can only enter absolute operands (e.g. EW0); symbolic entries are not possible.
   Tip: connected operands of the same data type only need one entry in the GD table. In this case you enter a colon after the operand and then the repeat factor. The repeat factor defines the size of the data area.

   Example: EW4: 3 means 3 words as of EW4.
2. Define the Send and Receive CPU; for this, position the insert marker in a table field and execute the menu commands **Edit > Sender, Edit > Receiver**.
3. Press the F2 key to switch from Overwrite mode to Insert mode.
4. Edit the table as usual. Here you can also use the menu commands **Edit > Cut**, **Edit > Copy** or **Edit > Paste**.
5. Complete your entries with RETURN.

2.3 Saving and initial compiling of the GD table

Saving

With Save you store in a source file the data that you have entered in your GD table.

- Select the menu command **GD table > Save.**

Or:

1. Select the menu command **GD table > Save as...**
2. Navigate to the project in which you want to save the GD table.
3. Acknowledge with "OK".

Compiling

The data you have entered in the GD table must also be compiled in language that the CPUs understand. From the clear GD table therefore you get the system data that can be processed by the CPUs. When compiling, for each CPU column exactly that system data is generated that is necessary for the communication of the CPUs concerned. This is why there is a separate GD configuration for each CPU.

- Click the corresponding symbol in the toolbar or select the menu command **GD table > Compile**. The GD table is now compiled according to **Phase 1**.

  Result:
  STEP 7 checks

  - The validity of the CPUs specified in headers of the CPU columns.
  - The syntax of the operands that you have entered in the table fields.
  - The size of the data areas for Sender and Recipient (data areas for Sender and Recipient must be the same size).
  - That the global data of a line are exchanged either only via K bus (backplane bus) or only via MPI subnetwork. Mixed mode is not possible.

After the first successful compilation the GD table is in Phase 1. In Phase 1 you can now edit status lines and reducing factors in the GD table.

Step 3: Setting Send and Receive Conditions

Introduction

The exchange of global data proceeds as follows:

- The Send CPU sends the global data at the end of a cycle.
- The Receive CPU reads the data at the beginning of a cycle.

Using the reducing factor you can define after how many cycles sending or receiving data is to take place.

Special case:
Reducing factor "0" means that the GD package is transferred event-controlled, i.e. not cyclically (only possible with S7-400 with SFC 60/SFC 61).

Reducing factor on Sender side

The following requirements should be met to keep the communications load of the CPU low:

- S7-300 CPUs: reducing factor * cycle time >= 60 ms
- S7-400 CPUs: reducing factor * cycle time >= 10 ms

Reducing factor on Recipient side

In order to prevent the loss of GD packages, GD packages must be received more often than sent. To ensure this you have the following rule:

- Reducing factor (Recipient) * cycle time (Recipient) < Reducing factor (Sender) * cycle time (Sender).

Reaction time

You can make an approximate calculation of the reaction time for two stations that exchange GD packages via an MPI subnetwork.

Requirement for calculating the reaction time

- Transfer rate 187.5 kbit/s
- No other communication via MPI (e.g. through connected PGs/OPs)

Calculation:

Reaction time = reducing factor (Sender) * cycle time (Sender) + reducing factor (Recipient) * cycle time (Recipient) + Number (of MPI users) * 10ms

- **Note**:
  For higher transfer rates the factor is "* 10ms" less; however, it does not drop linearly with increasing transfer rate.

3.1 Entering reducing factors

Procedure

1. Compile the GD table if this is not yet in Phase 1 (this can be seen from the entry in the status line at the bottom edge of the screen).
2. If there aren't any reducing factors displayed yet in the GD table, select the menu command **View > Reducing factors**.
3. Enter the desired reducing factors. You can only enter data in the columns in which the assigned GD package has entries.

   Note:

If you display the status lines and/or the reducing factor lines, you can only edit those lines and no other lines.

4. Recompile the GD table (Phase 2).

3.2 Entering status lines

Introduction

For each GD package you can define a status double-word per CPU "affected". Status double-words are identified in the table by "GDS". If you assign the status double-word (GDS) to a CPU operand of the same format, you can evaluate the status in the user program or in the status line (GDS). The STEP 7 online Help informs you in detail of the structure of a status word.

General status

STEP 7 gives a general status for all the GD packages (GST).

The general status, likewise a double-word with the same structure as the status double-word (GDS), comes from the OR link of all status double-words.

Procedure

1. Compile the GD table, if this is not yet in Phase 1 (this can be seen from the entry in the status line at the bottom edge of the screen).
2. If there aren't any GD status lines displayed yet in the GD table, select the menu command **View > GD Status**.
3. Enter the desired status double-words. You can only enter data in the columns in which the assigned GD package has entries. When entering the operands take the syntax of STEP 7 programming languages as a guideline.
   Warning: if you display the status lines and/or the reducing factor lines, you can only edit those lines and no other lines.
4. Recompile the GD table (**Phase 2**).

After editing the status lines and reducing factor lines, recompile the GD table in order to enter the new data into the system data.

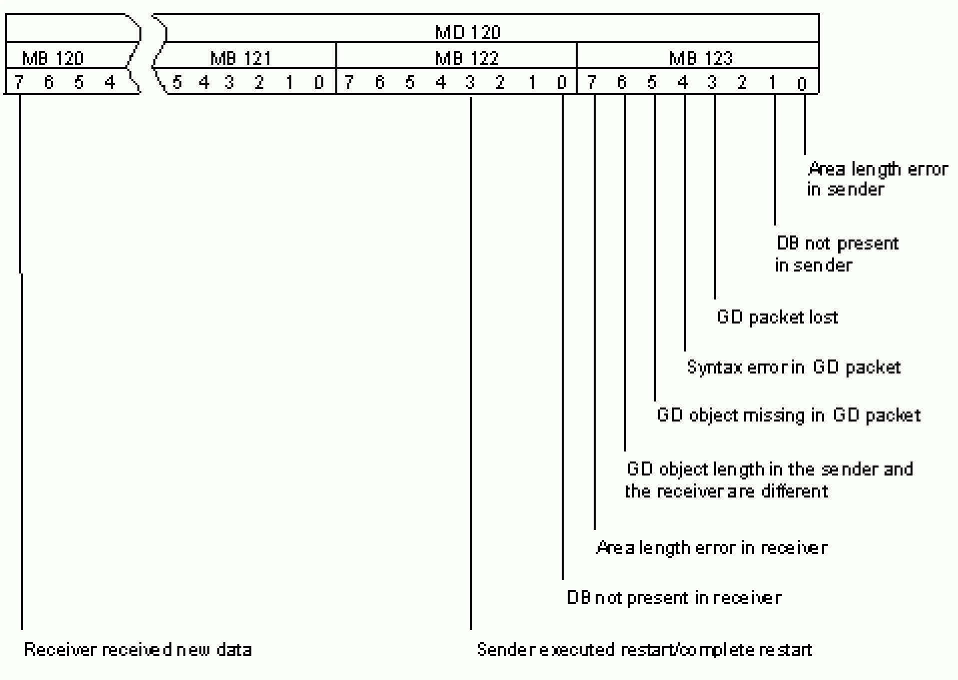
- **Note**:
  The system data created in Phase 1 is sufficient for executable GD communications. It can be loaded from the PG data management into the CPUs. Phase 2 is only necessary if you want to change default values for the reducing factor or make entries in the status lines.

3.3 Structure of the Status Double Word

The following figure shows the structure of the status double word and an explanation of the bits set.

A bit remains set until it is reset by the user program or via a programming device operation.

Any bits not listed are reserved and have no significance at present. The global data status occupies a double word; to make it clearer, MD120 has been used in the figure.


MD 120
MB 120
MB 121
MB 122
MB 123
7 6 5 4
5 4 3 2 1 0
7 6 5 4 3 2 1 0
7 6 5 4 3 2 1 0
Area length error in sender
DB not present in sender
GD packet lost
Syntax error in GD packet
GD object missing in GD packet
GD object length in the sender and the receiver are different
Area length error in receiver
DB not present in receiver
Receiver received new data
Sender executed restart/complete restart

Step 4: Loading the Global Data Configuration

Upon compilation the data of the GD table is converted into system data. If no errors are displayed after compiling, you can transfer the system data into the CPUs:

- Select the menu command **Target System > Load**.