# How do you connect a panel to a SIMATIC H station?

**Panels**

**FAQ · March 2010**

# Service & Support

Answers for industry.

**SIEMENS**

This entry originates from the Service&Support Portal of Siemens AG, Sector Industry, Industry Automation and Drive Technologies. The conditions of use specified there apply (www.siemens.com/nutzungsbedingungen).

Go to the following link to download this document.

http://support.automation.siemens.com/WW/view/de/23842653

# Question

How do you connect a panel to a SIMATIC H station?

# Answer

Follow the instructions and notes listed in this document for a detailed answer to the above question.

# Contents

# 1        Introduction

You cannot connect a panel directly to a SIMATIC H station with the same MPI/PROFIBUS addresses and IP addresses.

We describe the various options below in the following chapters:

- Unilateral panel connection to an H CPU
- Panel connection to H CPUs via two repeaters
- Redundant panel connection
- Connection of a scriptable panel with "software solution" to an H CPU
- Connection of a non-scriptable panel with "software solution" to an H CPU

**Communication options**

You have the following communication options for connecting a panel to the H CPU:

- Communication via MPI
  Panel and H-CPUs are connected to the MPI/DP interface via MPI.

- Communication via PROFIBUS
  The PROFIBUS interface of the H CPUs cannot be used for connecting to the panel, because you cannot set different PROFIBUS addresses. The connection to PROFIBUS is made via CPs, e.g. CP443-5.

- Communication via Industrial Ethernet
  The connection to Industrial Ethernet is made via CPs, e.g. CP443-1.

**Note on using direct keys**

The WinCC flexible function "Change Connection" does not change the connection of direct keys.

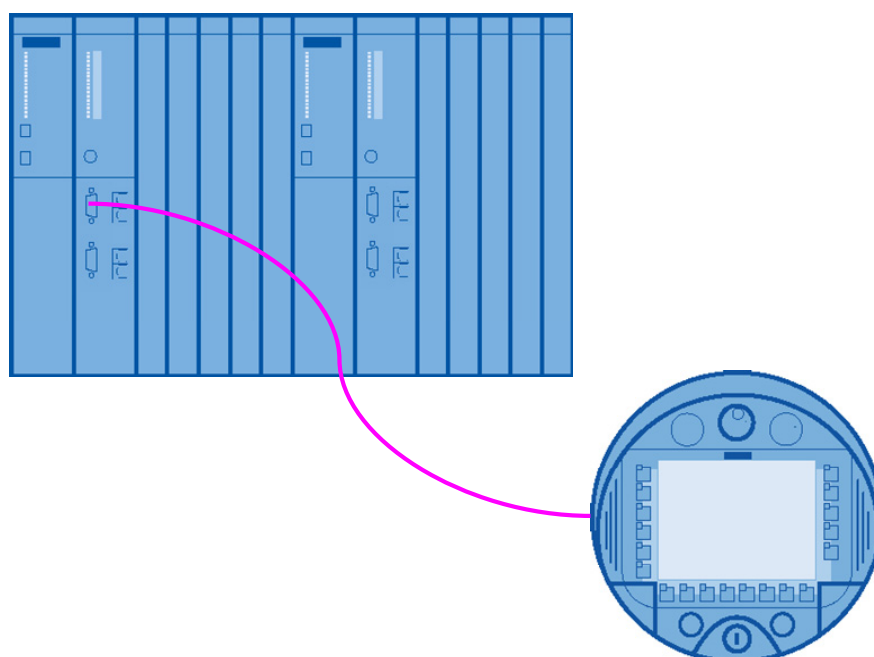# 2 Unilateral panel connection to an H CPU

You connect the panel to **one** H CPU.

You must set the **same** address on both H CPUs.

There must be an interface free for the panel on both H CPUs.

If the H CPU to which you have connected the panel fails, you must plug into the second H CPU.

Figure 2-1



| Note | In this version we recommend using a Mobile Panel. |
|---|---|

# 3 Redundant panel connection

You connect a panel to each CPU.

Figure 3-1



SIMATIC H Station
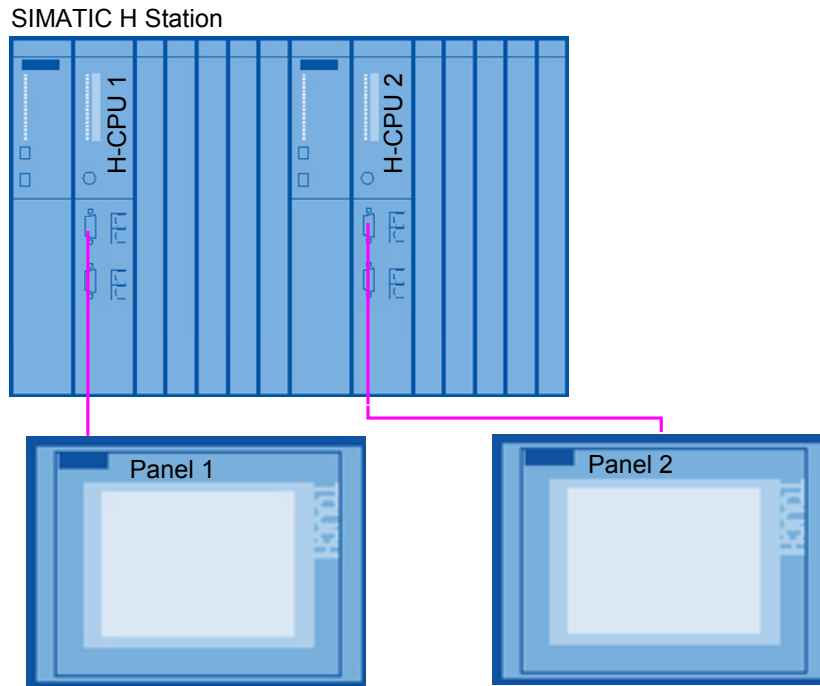
| Note | The "Alarm_S messages" and "CPU messages" can be imported only by H CPU 1. |
|------|---------------------------------------------------------------------------|

Proceed as follows to have all the messages **also** displayed on panel 2:

1. You must configure the connection from Panel 2 to **H CPU 1** in WinCC flexible. Figure 3-2
2. After the panel has started up, use the "**Change Connection**" function to connect Panel 2 to H CPU 2.

Figure 3-2
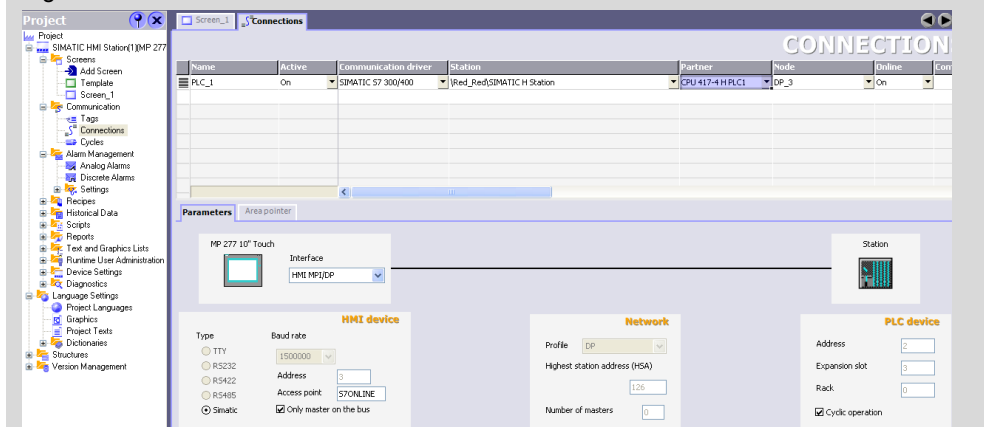
# 4 Panel connection to H CPUs via two repeaters

You connect the panel to the H CPUs via two repeaters.

You must set the **same** address on both H CPUs.

If one repeater is switched off, the panel has a connection to one H CPU.

Figure 4-1
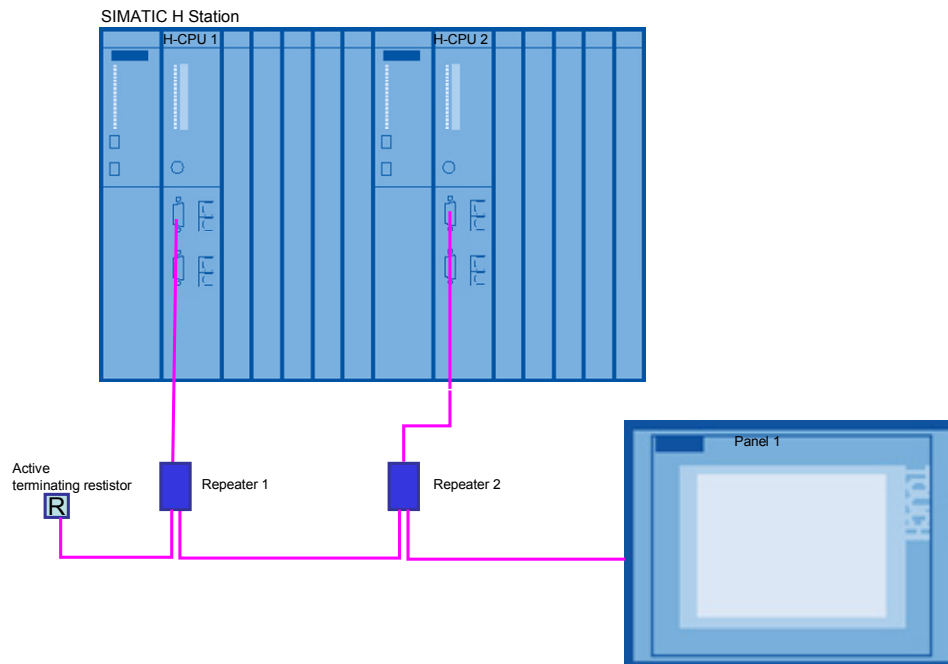


The operating system of the SIMATIC H station does a data comparison.

If you enter data on one panel, it is available on both H CPUs.

Supply power to just one repeater.

Power on/off options:

- Manually via a switch
- Automatically via the digital outputs

| Note | • If you switch off Repeater 1, the internal terminator is not active. So you need an active terminator. |
| --- | --- |

Table 4-1

| Components | MLFB |
| --- | --- |
| Repeater | 6ES7 972-0AA01-0XA0 |
| Active terminator | 6ES7 972-0DA00-0AA0 |

- The distributed I/O must be connected in a separate network.

**NetPro settings in STEP 7**

1. You set the same MPI/PROFIBUS address on both H CPUs.

2. You set the panel to "Networked: No"

Figure 4-2
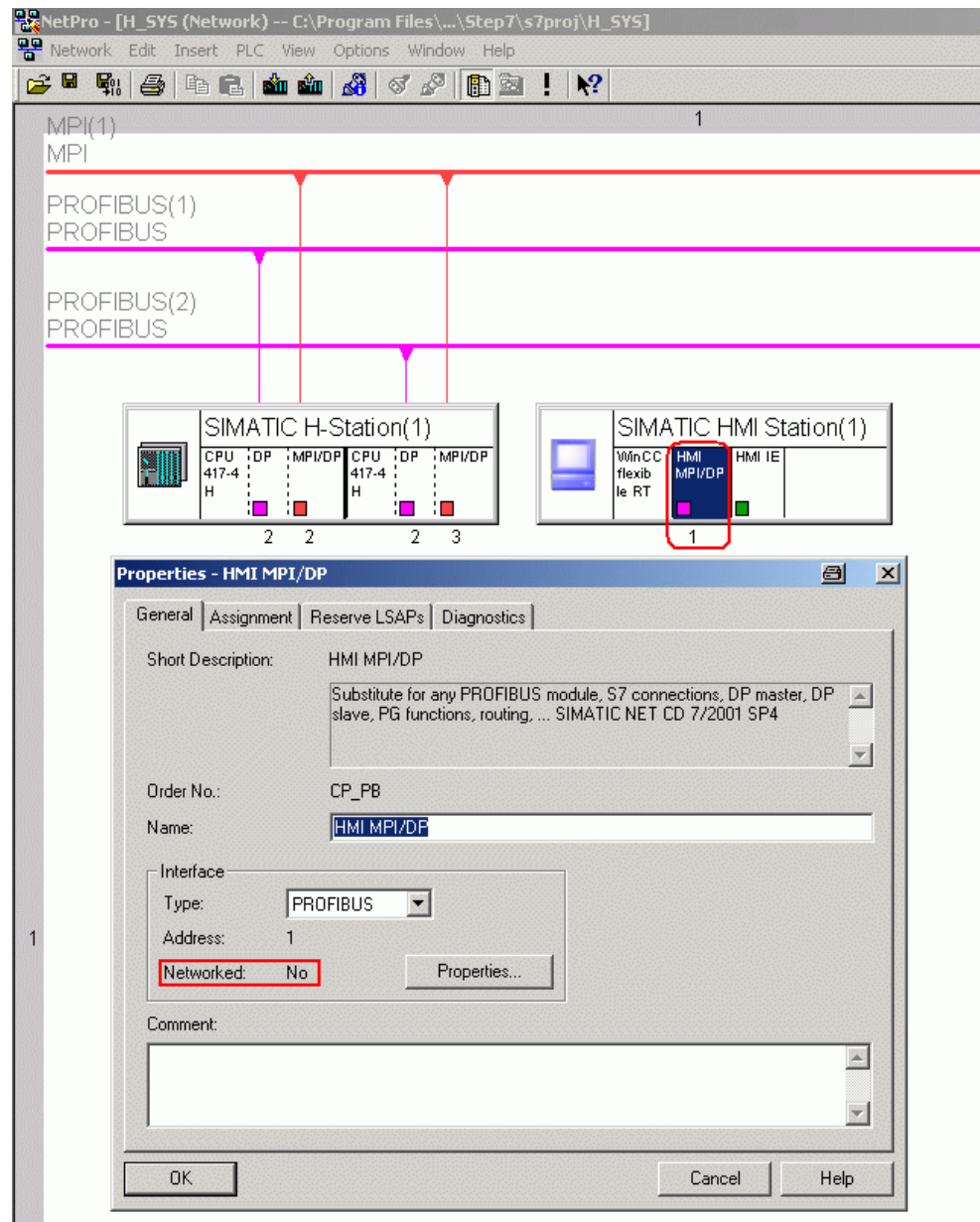


The following is checked when a connection is set up between the panel and an H CPU:

- Number of the rack

- Expansion slot

- Node address

Both H CPUs have **different** rack numbers.

**Communication setting in WinCC flexible**

You must make the following settings on the controller for a connection setup between panel and SIMATIC H station:

- Expansion slot 0
- Rack 0
- Unique MPI/PROFIBUS address

Then, regardless of the rack and slot, a connection is set up to the H CPU.

The panel finds the H CPU through the node address.

Figure 4-3

# 5 Connection of a scriptable panel with "software solution" to an H CPU
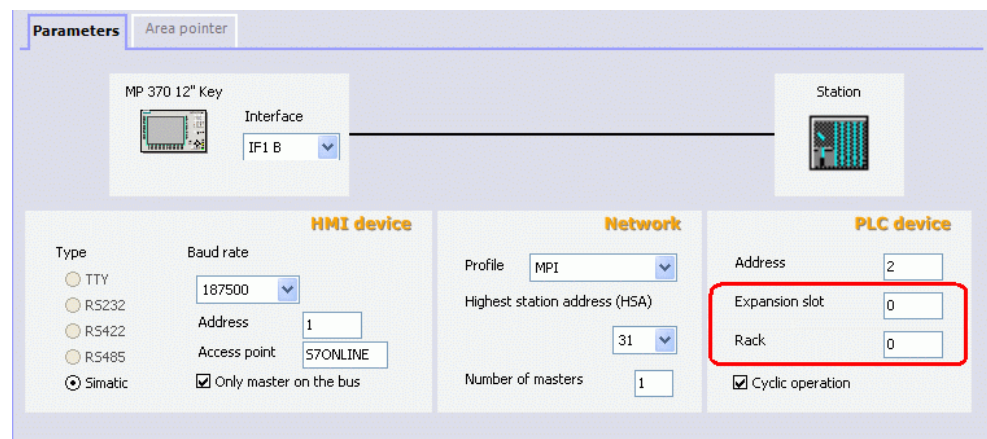
## 5.1 Basic functions

With this software solution, the connection is switched over automatically in the following error cases:

- STOP of CPU addressed
- CP failure
- Cable fault
- EMC

The following script templates are available as txt files for this solution.

- connection_PLC1.txt
- connection_PLC2.txt
- connection_lost.txt

The script templates are in the zip file named 23842653_WinCC_flexible_redundant_communication.zip on the Internet page from which you downloaded this document.
http://support.automation.siemens.com/WW/view/de/23842653

There is one *trigger* per connection. Each of these *triggers* triggers a script (connection_PLC1 and connection_PLC2).

The scripts have identical functions and differ only in their connection parameters.

The following functions are executed in the scripts:

1. Initialization:
   The connection that reaches the *trigger* writes the connection name to the tag for the *connection memory of the data link*.

2. Reset:
   The *connection status tag* is set to 1 to receive the "OK" status.

3. Fault detection:
   The *connection status tag* of the other connection is increased.
   This procedure detects a fault of the other connection after a number of cycles and checks whether the data link needs to be switched over.

4. Switchover:
   If the other connection cannot reset the *connection status tag*, it reaches a limit value. Then a check is made as to whether the data link is on this failed connection.
   If it is, the connection is switched over.

5. Reentry:
   This section deals with the return after a complete disconnection.
   The connection that first becomes active again switches the data link to itself.

Complete disconnection:

If there is a complete disconnection, a rapid trigger is no longer available to execute scripts.
The WinCC flexible scheduler with a cycle of one minute is used to detect and

display a total failure.

The script used in the WinCC flexible scheduler (connection_lost) includes the following functions:

1. Fault marking:
   The *connection status tag* is set to 5 for both connections. If both connections are active, the minute-by-minute action has no effect. 5 corresponds to the connection status "OK".

2. Deactivating:
   If the *connection status tags* of both connections are on 5 or higher, they are set to 100.
   100 corresponds to the connection status "deactivated".
   "connection_lost" is written to the *connection memory of the data link*.

Table 2 Status of the connection status tags

| Values | Status |
|--------|--------|
| 0 | Initialization, waiting for first trigger |
| 1 to 5 | OK |
| 6 to 10 | Faulty |
| 11 to 20 | Failure |
| 100 | Deactivated |

## 5.2 Sample configuration for Industrial Ethernet with integrated WinCC flexible project

The example shows a configuration with "Change Connection" with the following hardware and software configuration:

**Hardware used**

- Multi Panel MP370 10" Touch
- 2 x CPU 417-4H
- For communication via Industrial Ethernet: 2 x CP443-1

**Software used**

- WinCC flexible 2008 SP1
- STEP 7 V5.4 SP5

### 5.2.1 STEP 7 configuration

**Network configuration**

Select configured Industrial Ethernet connection and open the object with a right click. NetPro opens.

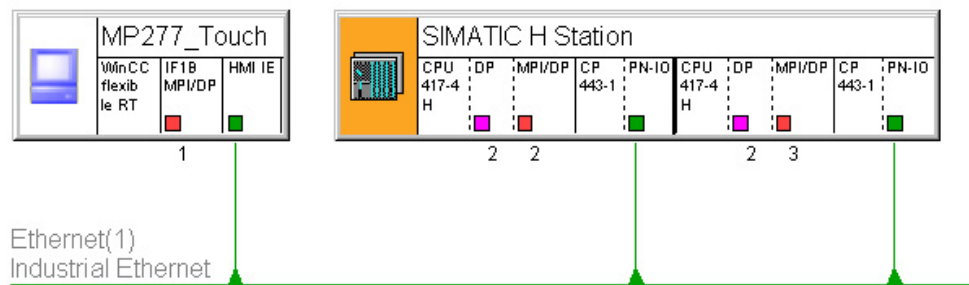- Connect both H CPUs to Industrial Ethernet via CP 443-1.

- Connect the MP 277 to the same Industrial Ethernet.
- You must create the following two "S7 connections":
  - S7 connection_1 → CPU417-4 H to "WinCC flexible RT"
  - S7 connection_2 → CPU417-4 H(1) to "WinCC flexible RT"

**IP addresses in example**

- CPU_1:     192.168.0.130
- CPU_2:     192.168.0.131
- MP:        192.168.0.3

Figure 5-1



**Hardware configuration, activation of a clock marker byte**

- Open the Properties of the H CPU
- "Cycle/Clock marker" tab     Subitem "Clock marker"
- Activate clock marker and enter the required marker byte. Marker byte 0 is entered in the example.

### 5.2.2      Connection parameterization in WinCC flexible

Three connections have been created.

- PLC_1
- PLC_2
- PLC_Changer_12

**Industrial Ethernet**

"PLC_1" is connected with the first H CPU via Industrial Ethernet.

"PLC_2" is connected with the second H CPU via Industrial Ethernet.

You must connect "PLC_Changer_12" "manually" to an S7-300/400 via Industrial Ethernet.

The IP address of the third connection "PLC_Changer_12" should correspond to the IP address of one of the first two connections "PLC_1" or "PLC_2".

Make sure that you enter the extension slot and rack number correctly.

Figure 5-2
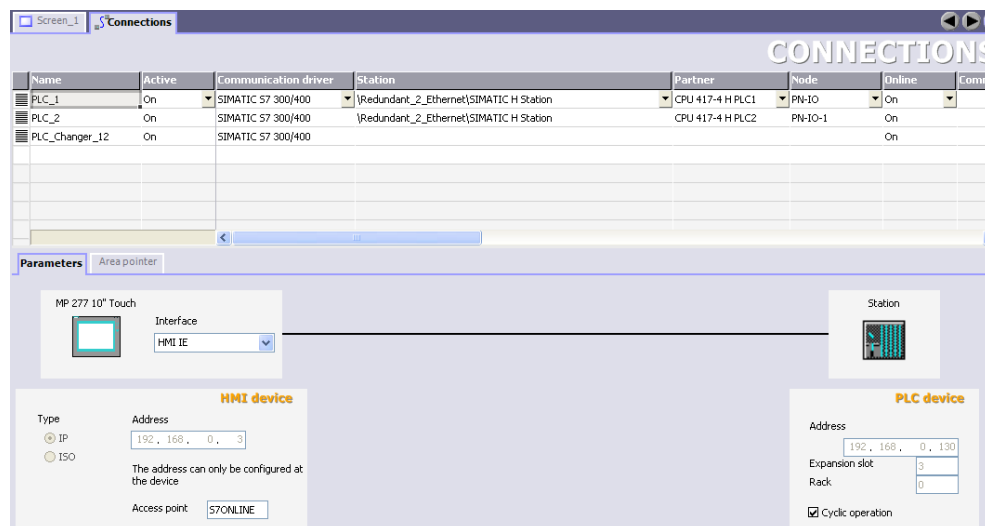


### 5.2.3 Tag parameterization in WinCC flexible

You need the following tags for fault detection and automatic switchover in case of fault. They are defined in the project tree under "Communication" → "Tags".

Table 3

| Variable name | Connection | Data type/ address | Acquisition | Acquisition cycle | Use |
|---|---|---|---|---|---|
| trigger_PLC1 | PLC_1 | Bool/M0.4 | Cyclic continuous | 500ms | Trigger for the VB scripts |
| trigger_PLC2 | PLC_2 | Bool/M0.4 | Cyclic continuous | 500ms | Trigger for the VB scripts |
| con_state_PLC1 | Internal tag | Integer | Cyclic when used | 1s | Connection status tags |
| con_state_PLC2 | Internal tag | Integer | Cyclic when used | 1s | Connection status tags |
| connected_to | Internal tag | String | Cyclic when used | 1s | Connection memory of the data link |
| Clock memory | PLC_Changer_12 | Byte/MB0 | Cyclic continuous | 100ms | Clock memory |

### 5.2.4 Script creation and implementation in WinCC flexible

Insert 3 scripts in the project tree under "Scripts" → "Add script". A code is stored in a txt file for each script (see 5.1); you can use the contents.

VB scripts in WinCC flexible need a cyclic trigger for cyclic calculations.

For this, you must assign the "connection_PLC1" and "connection_PLC2" scripts to the trigger tag.

- Communication > Tags "trigger_PLC1" Properties> Events >Change in Value> Select script "connection_PLC1".

- Communication > Tags "trigger_PLC2" Properties > Events > Change in Value> Select script "connection_PLC2".

You must store the "connection_lost" script for recognizing a complete disconnection in the scheduler.

- Device Settings > Scheduler > Add task with the event "1 Minute" and the function "connection_lost".

| Note | The panel and the SIMATIC H station must communication via the "PLC_Changer_12" connection. The "PLC_1" and "PLC_2" connections are for implementing the switchover function in case of error; therefore you should not load these connections with other communication. **The IP addresses for the connection change must be adjusted in the scripts "connection_PLC1" and "connection_PLC2" at the function "ChangeConnection".** |
|------|---|

### 5.2.5    Sample scenario

**Initialization**

- The "PLC_1" and "PLC_2" connections are both active.

- According to the predefined setting, the "PLC_Changer_12" connection is switched over to "PLC_1" ("connected_to" = "PLC_1").

**Fault detection and reset**

The "connection_CPU1" and "connection_CPU2" scripts are triggered by reading of the *triggers* ("trigger_PLC1" and "trigger_PLC2").

- Each time it is executed, "connection_PLC1" increments by 1 the *connection status tag* of the connection to "PLC_2" and sets its own *connection status tag* to 1.
  - con_state_PLC2  = con_state_PLC2 + 1
  - con_state_PLC1  = 1

- Each time it is executed, "connection_PLC2" increments by 1 the *connection status tag* of the connection to "PLC_1" and sets its own *connection status tag* to 1.
  - con_state_PLC1  = con_state_PLC1 + 1
  - con_state_PLC2  = 1

**Scenario 1**

The "PLC_1" connection fails due to a fault.

- The *trigger* can no longer be read via this connection.

- The "connection_PLC1" script is not executed.

- The "connection_PLC2" script continues to be executed cyclically.

The *connection status tag* of the connection "PLC_1" ("con_state_PLC1") is no longer reset and therefore the *connection status tag* reaches the limit value 11 (see Table 2 Status of the connection status tags).

The "connection_PLC2" script checks the *connection memory* "connected_to". Since this memory is located on "PLC_1", the "Change Connection" switches over the data link to the parameters of the "PLC_2" connection. The *connection memory* is set to "PLC_2".

- connected_to = PLC_2

**Scenario 2**

The "PLC_2" connection also fails due to a fault so that the "connection_PLC2" script can no longer be executed.

The scheduler executes the "connection_lost" script every minute.

Both *connection status tags* are set to 5 in the first cycle.

- con_state_PLC1 = 5
- con_state_PLC2 = 5

Both *connection status tags* are set to 100 in the second cycle. The *connection memory* is set to "connection_lost".

- con_state_PLC1 = 100
- con_state_PLC2 = 100
- connected_to = connection_lost

→ There is a Complete Disconnection.

# 6 Connection of a non-scriptable panel with "software solution" to an H CPU

## 6.1 Basic functions

By evaluating the status of the H CPU (Master / Reserve) and through the internal system function "Change Connection" it is possible to connect a panel directly to an H CPU.

For this, you must set different addresses for the H CPUs.

If the SIMATIC H station detects the failure of an H CPU (e.g. H CPU_1), then the "Change Connection" function sets up a connection to the other H CPU.

The "Change Connection" function disconnects the connection to the controller currently being used and sets up a new connection to the controller specified.

There are two ways of using the "Change Connection" function:

- Manual switchover with a function key: configure the system function "Change Connection" on the "Press" event.

- Automatic switchover: automatic call of "Change Connection" on the "High limit exceeded" event, for example.

**Restrictions**

The panels below do not support this procedure:

- OP 73
- TP 170A
- Micro Panels

## 6.2 Sample configuration for MPI, PROFIBUS and Industrial Ethernet

The example shows a configuration with "Change Connection" with the following hardware and software configuration:

**Hardware used**

- Multi Panel MP277 10" Touch
- Operator Panel OP 77B (not Industrial Ethernet)
- 2 x CPU 417-4H
- For communication via PROFIBUS: 2 x CP443-5
- For communication via Industrial Ethernet: 2 x CP443-1

**Software used**

- WinCC flexible 2008 SP1
- STEP 7 V5.4 SP5

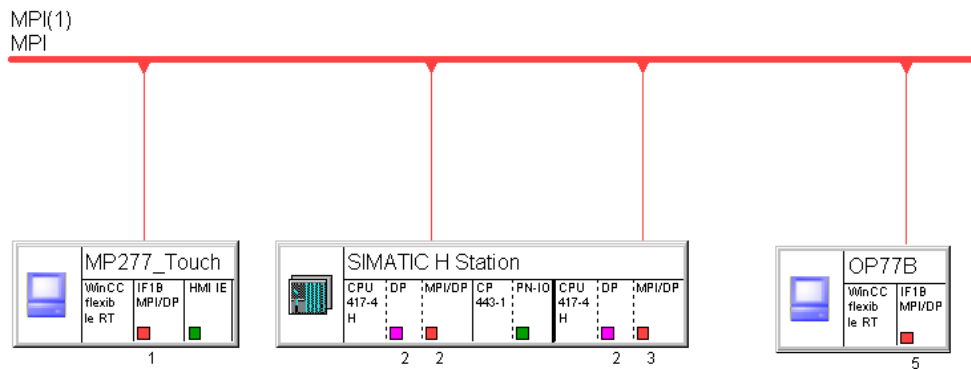### 6.2.1 STEP 7 configuration

**Network configuration**

Select configured MPI connection and open the object with a right click. NetPro opens.

- Connect both H CPUs to MPI via the MPI/DP interface.
- Connect the MP 277 and the OP 77B to the same MPI bus.

**MPI addresses in example**

- CPU_1: **2**
- CPU_2: **3**
- MP: **1**
- OP: **5**

Figure 6-1
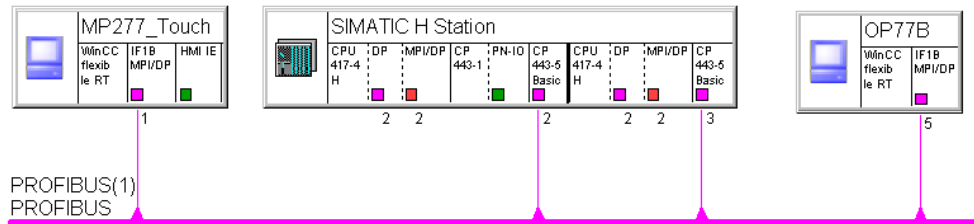


**Network configuration**

Select configured PROFIBUS connection and open the object with a right click. NetPro opens.

- Connect both H CPUs to PROFIBUS via CP 443-5.
- Connect the MP 277 and the OP 77B to PROFIBUS.

**PROFIBUS addresses in example**

- CPU_1: **2**
- CPU_2: **3**
- MP: **1**
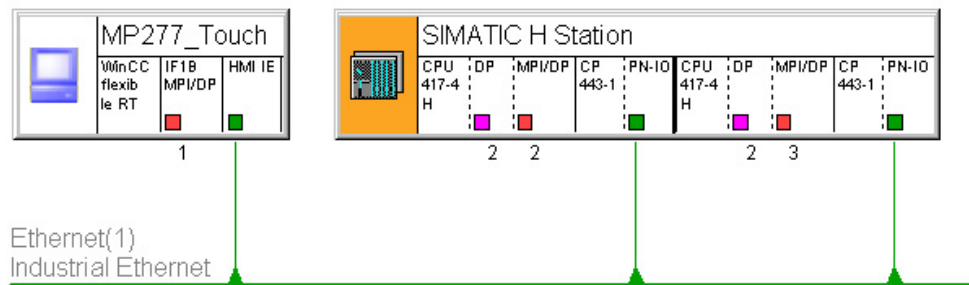- OP: **5**

Figure 6-2



**Network configuration**

Select configured Industrial Ethernet connection and open the object with a right click. NetPro opens.

- Connect both H CPUs to Industrial Ethernet via CP 443-1.
- Connect the MP 277 to the same Industrial Ethernet.

**IP addresses in example**

- CPU_1: 192.168.0.130
- CPU_2: 192.168.0.131
- MP: 192.168.0.3

Figure 6-3



**Program block**

The FB523 is called in the FB10 used. Function block "FB523" permits output of the operating modes "RUN/STOP" and the "Master/Reserve" status of an H system.

Via the evaluation of which of the two H CPUs is master, there is a change of connection in WinCC flexible.

| Note | Entry ID: 19537149 contains a detailed function description of FB523. |
|------|----------------------------------------------------------------------|

## 6.2.2 Connection parameterization in WinCC flexible

Three connections have been created.

- PLC_1
- PLC_2
- PLC_Changer_12

**MPI/PROFIBUS**

"PLC_1" is connected via MPI/PROFIBUS to the first H CPU.

"PLC_2" is connected via MPI/PROFIBUS to the second H CPU.

"PLC_Changer_12" has been connected "manually" via MPI/PROFIBUS to an S7-300/400.

The CPU address is arbitrary.

For all three connections, make sure that the "Only master on the bus" parameter is disabled in the "Panel" field and that you enter the extension slot and rack number correctly.

Figure 6-4
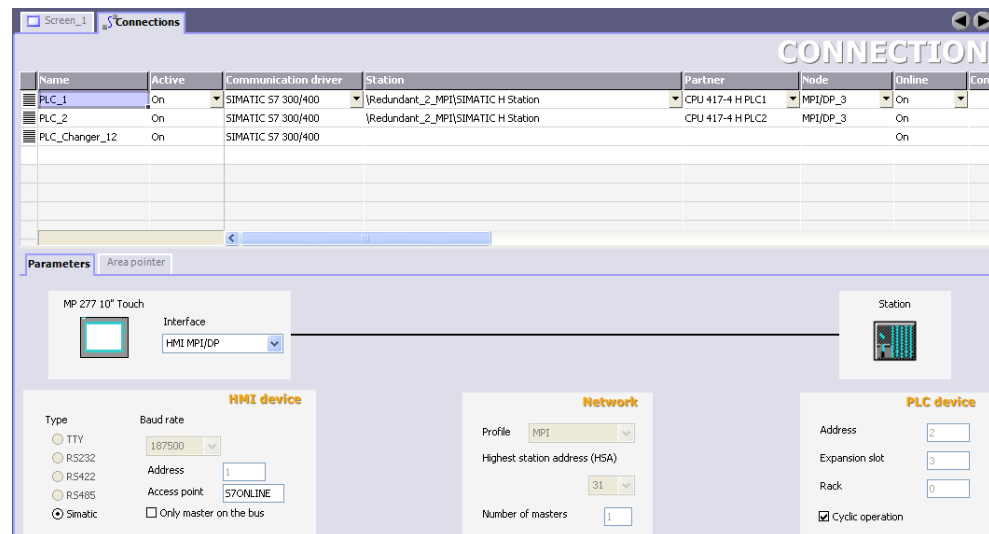


**Industrial Ethernet**

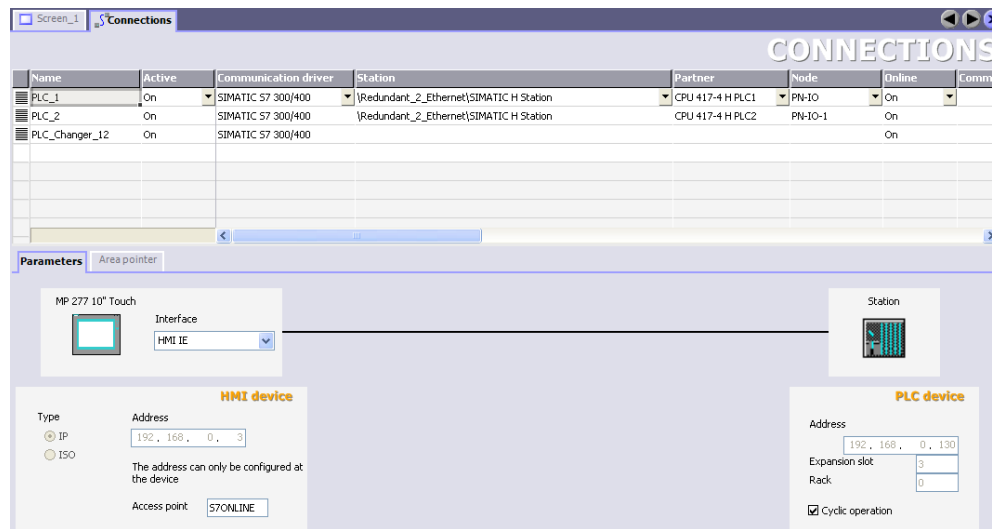"PLC_1" is connected with the first H CPU via Industrial Ethernet.

"PLC_2" is connected with the second H CPU via Industrial Ethernet.

"PLC_Changer_12" was connected "manually" to an S7-300/400 via Industrial Ethernet.

The controller address is arbitrary.

Make sure that you enter the extension slot and rack number correctly.

Figure 6-5



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Name** | **Active** | **Communication driver** | **Station** | | **Partner** | **Node** | **Online** | **Comm** |
| PLC_1 | On | SIMATIC S7 300/400 | \Redundant_2_Ethernet\SIMATIC H Station | | CPU 417-4 H PLC1 | PN-IO | On | |
| PLC_2 | On | SIMATIC S7 300/400 | \Redundant_2_Ethernet\SIMATIC H Station | | CPU 417-4 H PLC2 | PN-IO-1 | On | |
| PLC_Changer_12 | On | SIMATIC S7 300/400 | | | | | On | |

| Note | Communication between panel and H CPU is via the "PLC_Changer_12" connection created. |
|---|---|

### 6.2.3 Tag parameterization in WinCC flexible

1. You configure a tag for each of the "PLC_1" and "PLC_2" connections (DB1.DBB4 / DB1.DBB5). Acquisition "**Cyclic continuous**".

2. For the tags, under "Properties > Limits" you define a value of "**0**" for the Upper limit. See Figure 6-6

3. Under "Events > High limit exceeded" you configure the "Change Connection" function.
   Use "PLC_Changer_12" as "Connection partner".
   The addresses, extension slot and rack number are parameterized according to the connection. See Figure 6-7 Sample MPI Address 2
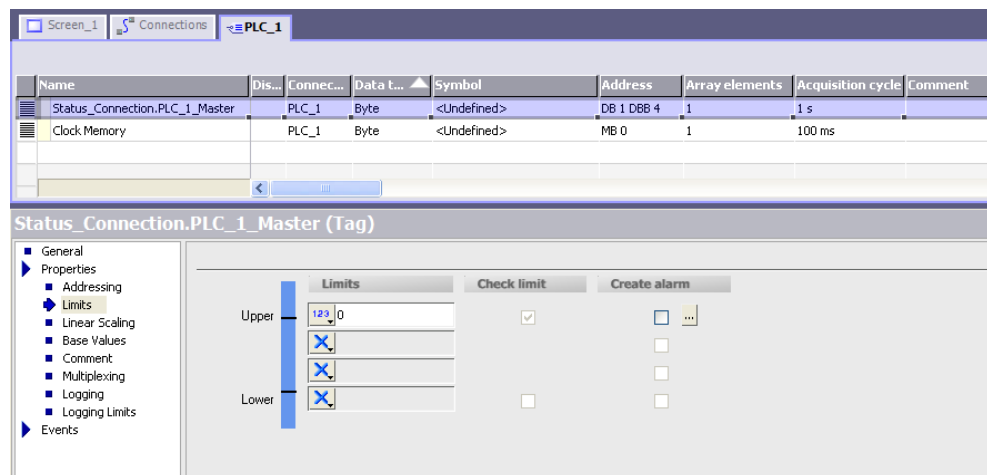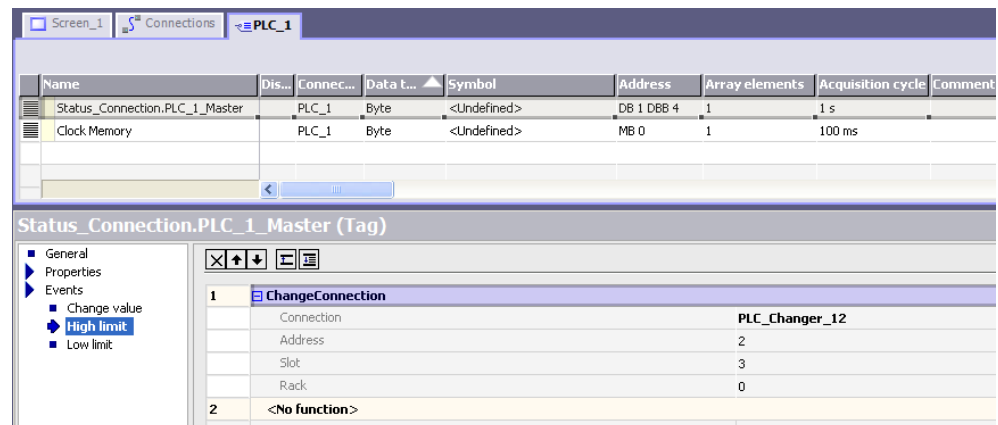
Figure 6-6

Figure 6-7



## 6.2.4 Project tags in WinCC flexible

All the project-relevant tags (except the tags described in 6.2.3) are assigned to the "PLC_Changer_12" connection.

It is not possible in this case to have a symbolic "address assignment".

**Tip:**

1. First insert all the tags via "Symbolic Address Assignment".
2. Under "Connection", change the name of the connection (PLC_1 → PLC_Changer_12).
3. Under "Symbol", you change the entry to "<Undefined>".
4. Edit multiple entries via "Edit mass data"; mark the first entry and drag it down with the mouse.

Figure 6-8



## 6.2.5 Function

**PLC_1 is master**

In WinCC flexible, the upper limit value of the "DB1.DBB4" tag is evaluated (limit value "0"). If the "DB1.DBX4.0" bit is set, the "Limit" value is exceeded and the "Change Connection" function is executed.

The connection changes from the defined connection of "PLC_Change_12" to the parameterized connection parameters of "PLC_1".

The connection to the PLC_1 is set up.

**PLC_2 is master**

In WinCC flexible, the upper limit value of the "DB1.DBB5" tag is evaluated (limit value "0"). If the "DB1.DBX5.0" bit is set, the "Limit" value is exceeded and the "Change Connection" function is executed.

The connection changes from the defined connection of "PLC_Change_12" to the parameterized connection parameters of "PLC_2".

The connection to the PLC_2 is set up.

Table 6-1

| H CPU | Address | Bit status | Result |
|-------|---------|------------|--------|
| PLC _1 | DB1.DBX4.0 | 1 | Master |
| PLC _2 | DB1.DBX5.0 | 0 | Reserve |
| PLC _1 | DB1.DBX4.0 | 0 | Reserve |
| PLC _2 | DB1.DBX5.0 | 1 | Master |

### 6.2.6    Connection testing

When the connection to the H CPU is established, the H CPU that is master is displayed.

Via a configured counter you can see that it is still possible to operate after a "connection change".