SIEMENS

SIMATIC

S7-300/S7-400 Software redundancy for SIMATIC S7

Function Manual

04/2010
A5E02171565-02

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E02171565-02
℗ 04/2010

Table of contents

Figures

Contents

1

How should I use this description? - A tip for readers 2

Introduction

The following sections describe how to use the "Software redundancy " package to increase the availability of SIMATIC S7 automation systems.

This description concerns the "Functional software redundancy" product with the following order nos.:

- Single-user license: 6ES7862-0AC01-0YA0, version 1.2 SP3
- Copy license: 6ES7862-0AC01-0YA1, version 1.2 SP3

The description of the product is also available as Online Help. You benefit from this feature as it lets you look up context-sensitive information while programming and configuring a project in STEP 7 on your programming device/PC. It also saves you from having to refer to printed media.

However, for those customers who still prefer to read the printed page, we summarized all the Help topics in a single document that you can view and print out using the Adobe Acrobat Reader. The corresponding ‘SWR_English.PDF’ document is available on the CD.

You need Acrobat Reader to open the document. This software is a license-free product of Adobe. You can install a current version of Acrobat Reader the "S7 Manual" subdirectory of the STEP 7 directory, if it was not already installed along with STEP 7.

Differences between versions 1.2 and 1.2 SP3

The differences between versions 1.2 and 1.2 SP3 are as follows:

- The response of the software redundancy following the return of power has been improved in version 1.2 SP3.
- The examples for WinCC were adapted to WinCC V6.2 and V7.0.
- Software redundancy is released for WinAC RTX 2008.

Target group

This description is aimed at readers who are already familiar with the S7-300 or S7-400 automation systems and the ET 200M distributed I/O device. It is also assumed that readers have a basic knowledge of working with the STEP 7 programming software.

Recommended procedure

This description covers several self-contained topics. We recommend that you start by reading the "Introduction" and “How software redundancy works” sections. Those sections outline the basic principles of using software redundancy.

If you have already acquired a certain expertise in working with STEP 7, take a look at our projects with examples for S7-300 and S7-400. The simplified application provided clearly demonstrates all the steps required.

However, if you would first like to get familiar with the blocks and parameters required, then please read the section "Blocks for software redundancy". All the essential information about the blocks is available at a glance in that section. It also contains two examples for S7-300 and S7-400 and corresponding projects with basic configurations. The projects are available after installation in the STEP 7 project folder. You can expand those projects to suit your specific requirements.

The "References and supplementary information" section deals with a number of separate topics that are intended to provide more in-depth information and answers to specific questions. That section provides a description of the function principle and of the components you require to configure a system with software redundancy.

Introduction 3

3.1 Why use a system with software redundancy?

Production down times cost time and money

The rising level of automation of industrial plants with the focus set on enhancing productivity and quality also increases dependence on the availability of automation systems. The failure of an automation system, e.g. due to CPU failure, and resultant loss of production and down times can incur high costs.

In many applications, the demands placed on the quality of redundancy, or the size of plant areas that require redundant automation systems, are not such that integration of a special fault-tolerant system is absolutely necessary.

In many scenarios, it is sufficient to provide straightforward software mechanisms that enable continuation of failed control tasks on a standby system.

Such requirements are fully met by means of software redundancy.

Increase of availability through software redundancy

Software redundancy runs on standard S7-300 and S7-400 automation systems.

Availability can be increased for single-channel distributed I/O that is installed in an ET 200M with redundant IM 153-2 DP slave interface. The DP slave interface modules feature two DP interfaces, one of which is connected to the DP master system on station A, while the other is connected to the DP master system on station B.

The software redundancy implemented on both automation systems allows for the continuation of fault-tolerant control tasks.

The term "fault-tolerant control task" denotes that component of the user program which must be continued by the standby station after failure of the master station. This can encompass either the entire user program, or only a certain component thereof.

Software redundancy enables you to manage the following types of failure:

- Failure of CPU components (power supply, backplane bus, DP master)
- CPU failure due to hardware faults or software errors
- Interruption of the bus cable for the redundant connection or for the redundant DP slave interface module
- Defective PROFIBUS module in the redundant DP slave interface module IM 153-2

3.2 What hardware is required?

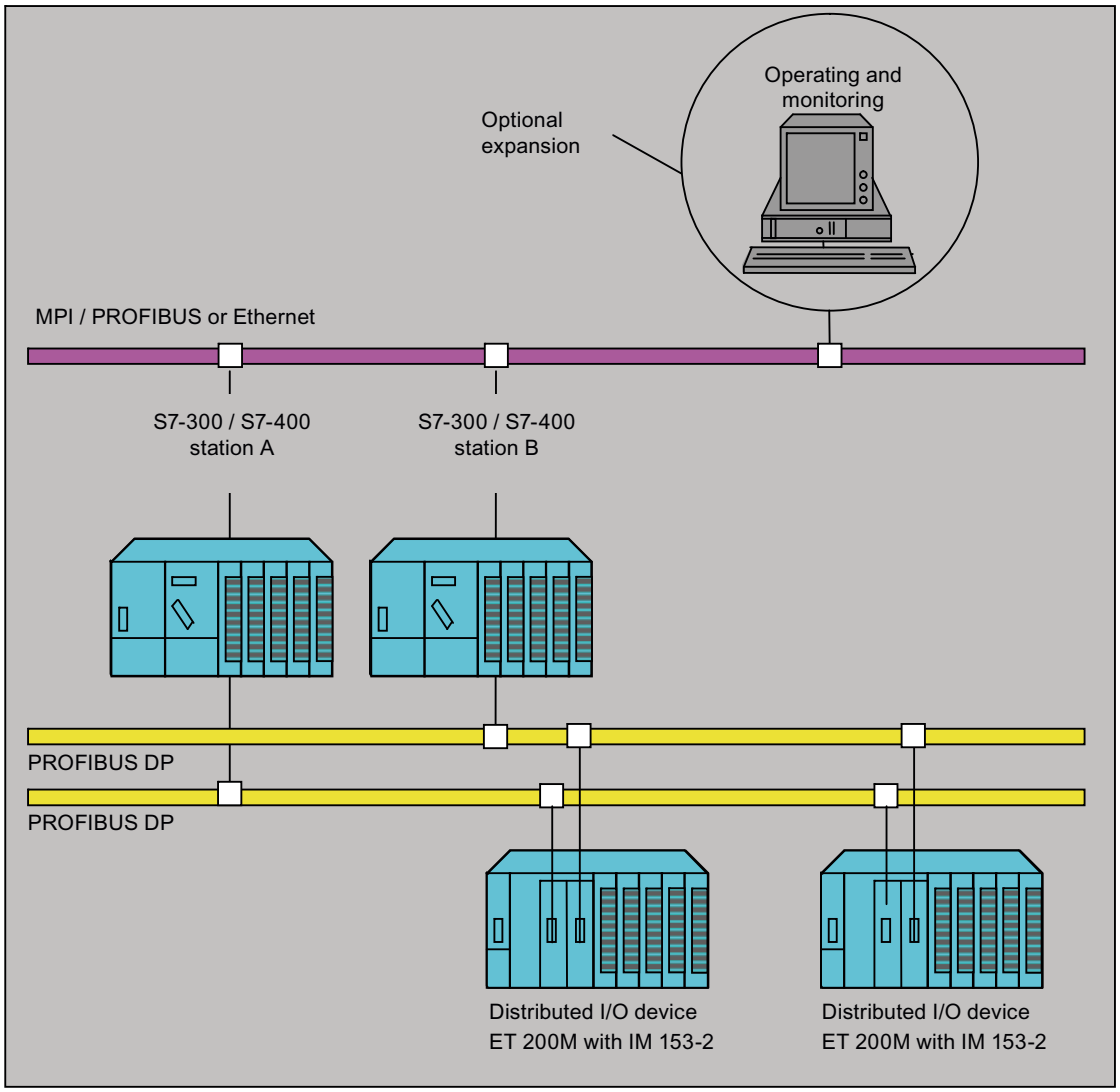
Two S7-300 and/or S7-400 stations form the core of hardware requirements and are each equipped with a CPU and a connection for a DP master system.

The two stations are linked by means of a bus system that you can use to exchange data.

The I/O devices are interconnected by way of two DP master systems: One DP master system on station A and on one station B.

ET 200M distributed I/O devices with redundant DP slave interface module IM 153-2 are connected to DP master systems. The DP slave interface module allows the failover from the first to the second interface in the event of a fault in order to forward process status data from the second DP master to the I/O.

Overview of the hardware configuration



3.3 What software is required?

STEP 7 programming software

You need the STEP 7 Basic Package V5.2 or higher to assign parameters the blocks for software redundancy.

Optional standard tools for SIMATIC NET and SIMATIC HMI

It goes without saying that you can use all of the optional engineering and configuring tools on systems with software redundancy.

The table below details the standard tools also used to set up the projects of our example applications.

Designation	Purpose of the tool
ProTool V3.01 or higher	Configuring SIMATIC HMI operator panels
WinCC V6.0 or higher	Graphic-based tool for configuring WinCC operator stations of the SIMATIC HMI product line

3.4 Where can I use software redundancy?

Software redundancy can be used in any scenario where central and particularly important plant components require greater levels of availability and where temporary failures such as the loss of a few processing cycles while switching from one station to another (master to reserve changeover) can be tolerated by the process. The following are examples of such plant components:

- Process control systems for cooling water circuits
- Process control systems for water conditioning plants
- Traffic supervision and control systems
- Systems for the monitoring and controlling fill levels
- Systems for monitoring and controlling the temperature in cold stores
- Systems for monitoring and controlling the temperature of furnaces

See also

Features and properties of software redundancy (Page 49)
Master to standby changeover (Page 50)

How software redundancy works

4

4.1 How does a system with software redundancy work?

Definition

A system with software redundancy is characterized by:

- Two S7-300 and/or S7-400 stations that are linked via bus system.
- A redundant user program that is loaded on both stations.
- Two DP master systems to which ET 200M distributed I/O devices with a redundant DP slave interface module such as the IM 153-2 are connected.
- Integration of the blocks provided in the "Software redundancy" package

Principle of software redundancy

The flow chart below shows the function principle of software redundancy under the aspect of the master and standby CPUs.

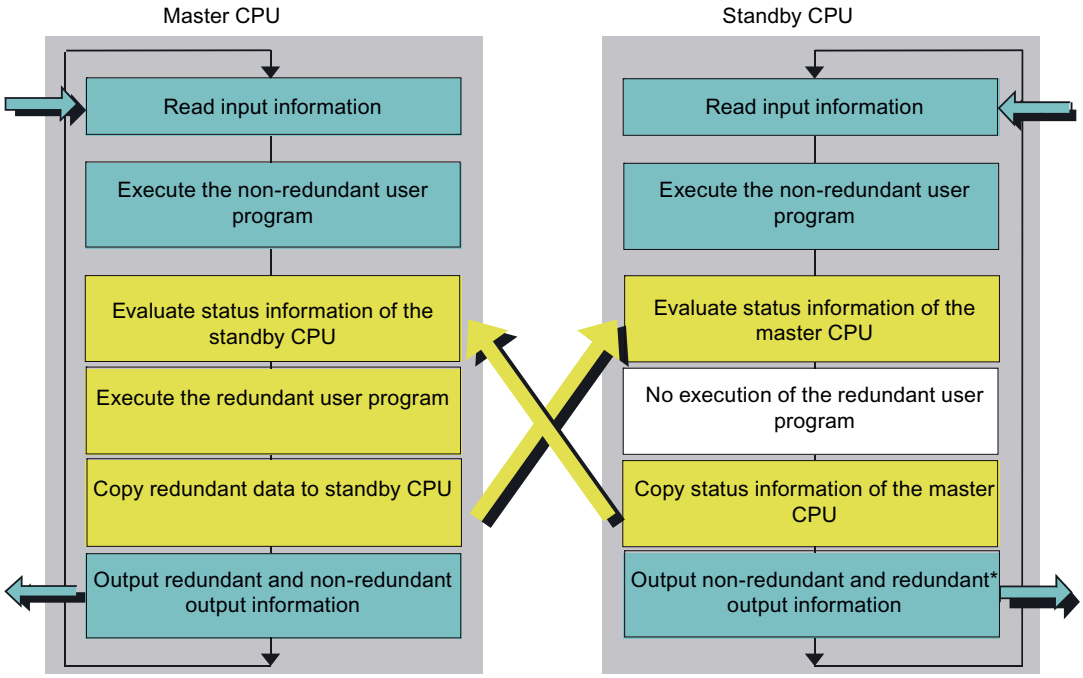


Figure 4-1 Principle of software redundancy

The fault-tolerant component of the software is loaded on both the master and the standby stations. While the master CPU is processing that component of the program, it is skipped in the standby CPU. Having the standby CPU skip that component of the program prevents inconsistency of the two program components, e.g. as a result of alarms, different cycle times etc. This means that the program on the standby station is always ready to take over processing.

General information

This type of standby mode, by the way, is referred to as warm standby, as opposed to hot standby used on the H systems, for example the S7-400H. In the latter, the processing on both CPUs is closely coordinated.

Master station continuously transfers updated data to standby station

To avoid having to start the fault-tolerant user program "from scratch" after failure of the master station, the master station continuously transfers its actual process data to the standby station.

However, the transfer of such data can take several cycles, depending on the method of communication chosen, or on the volume of data, i.e. processing at the standby station is always delayed for a certain number of cycles compared to the master, depending on communication performance and the data volume. A master to standby station changeover is initiated immediately after a fault has been detected at a CPU, at a DP master, or at a DP slave within the master station. With this type of changeover, the standby station takes over the process and assumes the master function.

Areas of the redundant software component

The redundant software component contains a process image, an IEC timer area, an IEC counter area, a bit memory address area, and a data block area. Only the redundant software is allowed write access to those areas.

Within the configuration phase you always have to bear in mind that contiguousness is absolutely essential at all the areas referred to above.

Those contiguous areas are scanned during the assignment of parameters for the "SWR_START" startup block.

Processing unilateral I/O devices

In addition to the redundant software component, it is, of course, also possible to load a program which controls the unilateral I/O devices of the relevant CPU. Software redundancy does not have any influence on that component of the program.

The term unilateral I/O devices denotes I/O modules that are not addressed in the redundant component of the user program, i.e. they are only assigned to one CPU. Physically, such modules can be connected as central or distributed devices to their own DP master system, or can be connected as distributed devices to one of the two DP master systems that contain the redundant DP slave interface modules.

Data exchange between the two stations

The non-redundant component of the program can exchange its data with the redundant software by means of suitable data blocks. Those data blocks are exchanged by means of the software redundancy system and are therefore made available to the partner station.

The inputs are written to the process image of inputs (PII) at the start of OB 1. The redundant software is processed before any data of the redundant software component (PIO, bit memories, DBs, timers/counters and instance DBs) is transferred to the standby system. The standby station must receive the data from the active station after having completed its startup, or after redundancy has been recovered in that software component.

At the end of OB 1, the data of the redundant PIO is written to the process image of outputs at the master and standby stations and is transferred from there to the I/O devices at the end of the OB cycle.

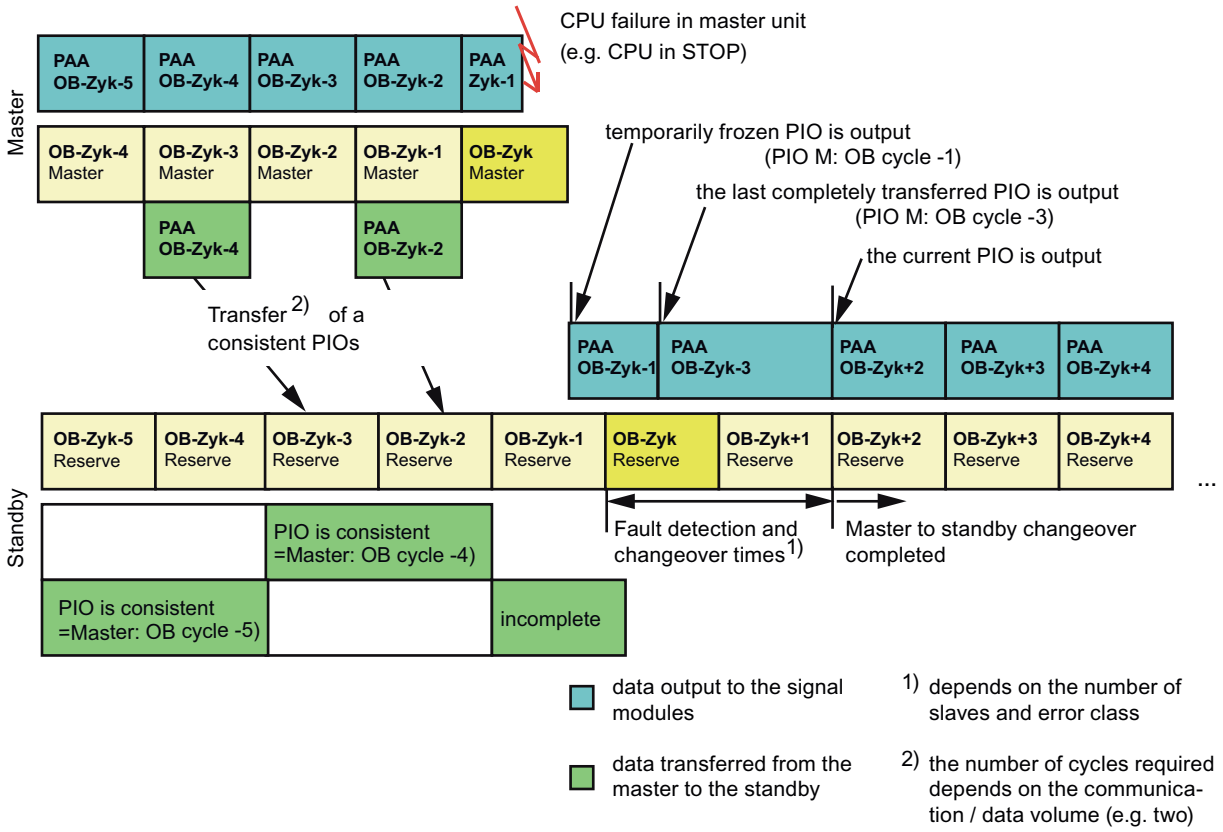
Interrupts can be received by the active station at any time and are processed immediately.

Interrupts can be lost if a changeover takes place at that moment or shortly afterwards.

Master to standby changeover in detail

In order to avoid having to start the standby station "from scratch" after master failure, a complete and consistent PIO of the fault-tolerant program component is transferred to the standby station to cover emergency / changeover situations.

The diagram below illustrates the transfer of relevant processing data to the fault-tolerant program that is ready to take over on the standby device.



The time required for the transfer may take longer than one cycle, depending on the communication mode and the volume of data to be transferred. In the example, it is assumed that it takes two cycles to transfer the entire process image (see the diagram).

That is, in our example, every second PIO is transferred from the master to the standby.

During normal operation, all redundant DP slave interface modules are assigned to the master station and output the data transferred by the DP master of the master station.

The standby station - or more precisely the DP master of the standby station - generally outputs the last PIO that was completely transferred to the standby station to the signal modules. However, that data is ignored by the DP slave interface modules because all slaves are assigned to the DP master of the master CPU.

In the course of an explicit changeover initiated by command, or of a fault-related implicit master-standby failover, the slave stations are changed over as well or the DP slave interface modules failover automatically.

The DP slave stations fail over automatically, for example after detection of a problem on the DP master or on the DP bus of the DP master station.

During this DP slave changeover, the most recently output PIO values are frozen on the DP slaves (see the diagram above).

If the DP slave stations have automatically failed over to the DP master of the former standby station, and if this station has not yet completed its failover from master to standby, the last PIO completely transferred to the standby station is output to the signal modules. A station-specific master to standby changeover can take several cycles, depending on the nature of the fault.

On completion of the master to standby changeover, the PIO determined by the new master is output (see the diagram above).

The changeover can be completed within a single cycle, given optimum communication conditions, small volumes of data and faults such as "CPU in STOP" (on an S7-400).

In the example, we consciously chose to illustrate a changeover involving a fail rate of 5 cycles.

Any manually initiated changeover will be optimized. That is, for example, it is not initiated until immediately after the PIO transfer has been completed.

Recovering software redundancy after repairs

To recover software redundancy, for example after failure of a CPU, all configuration data and the entire program must be loaded from the programming device or memory card to the replacement CPU. That CPU is then started.

General information

Once line voltage has been restored with a CPU in STOP operating mode, the second CPU will work in solo mode (Master). The Profibus of the CPU that is in STOP is active and no outputs will be enabled.

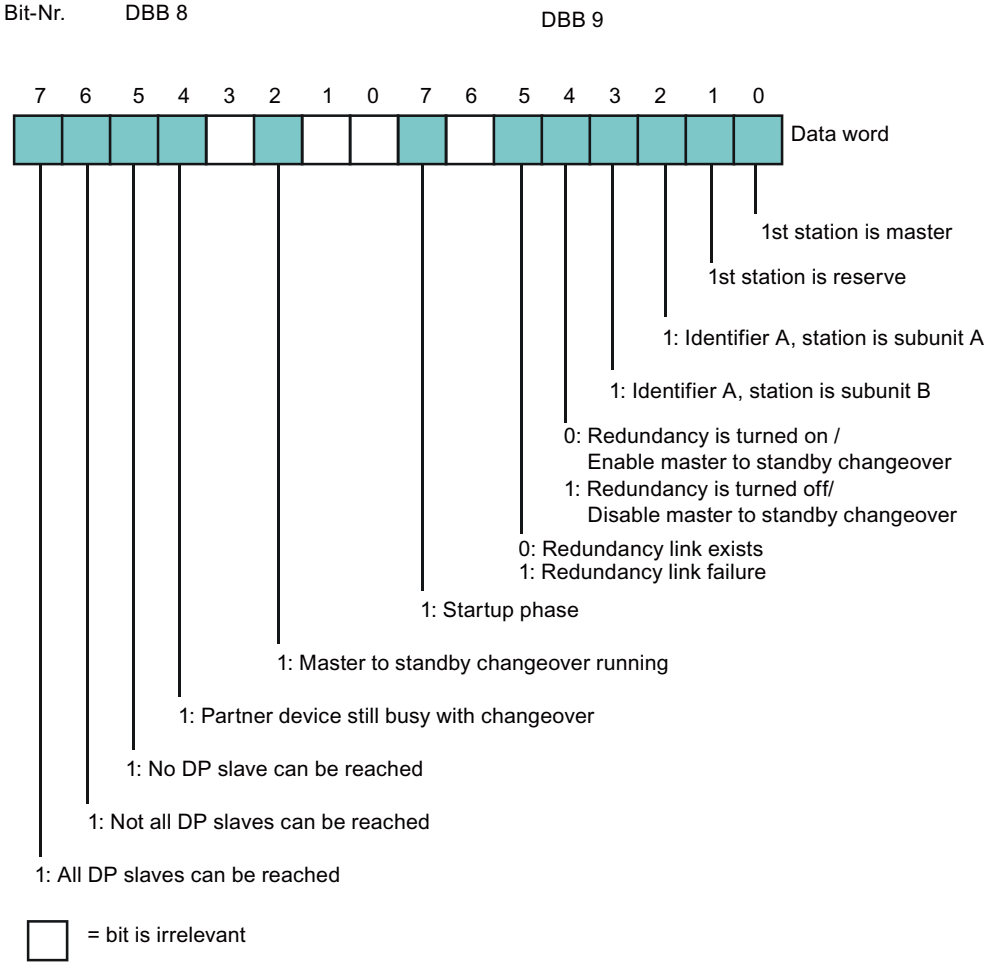
When the CPU changes from STOP operating mode to RUN operating mode, Profibus will switch to the second chain and the output will be enabled.

This behavior will only take place in case line voltage has been restored with a CPU in STOP operating mode.

4.2 Structure of the status word for software redundancy

The overview below shows the bit assignment of the status word. The status word is located in DBW 8 of the instance DB for FB 101 'SWR_ZYK'.

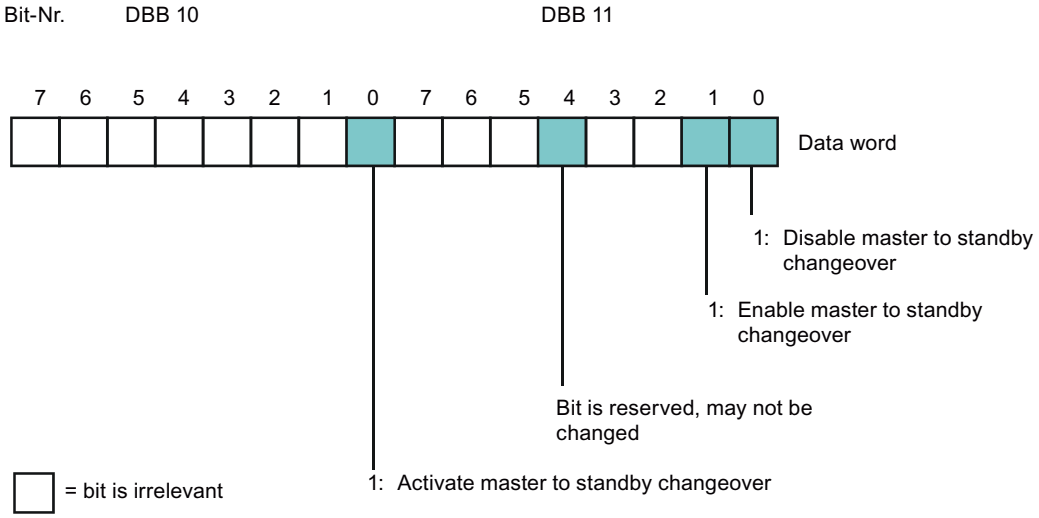
Status word for software redundancy



4.3 Structure of the control word for software redundancy

The overview below shows the bit assignment of the control word. The control word is located in DBW 10 of the instance DB for FB 101 'SWR_ZYK'.

Control word for software redundancy



For locking the master reserve change-over on the user level you will have to set bit 11.0 in the control word. The spare device writes zeroes to the PIO of the redundant DP slave interface module IM 153-2.

That status is retained until you re-enable redundancy (by setting bit 11.1 in the control word). If "Enable Master reserve change-over" was set, the control bits in the control word will be automatically reset to "0" after the setting. Changes will be visible in the status word.

Note

For locking the master reserve change-over on the user level you will have to set bit 11.0 in the control word. The spare device writes zeroes to the PIO of the redundant DP slave interface module IM 153-2.

That status is retained until you re-enable redundancy (by setting bit 11.1 in the control word). If "Enable Master reserve change-over" was set, the control bits in the control word will be automatically reset to "0" after the setting. Changes will be visible in the status word.

4.4 Rules for using software redundancy

The following sections provide a summary of all the rules to be followed when configuring and programming a system with functional software redundancy.

Hardware configuration rules

- The configuration of the ET 200M distributed I/O device with a redundant DP slave interface module, for example an IM 153-2, must be identical on both stations. To prevent any loss of consistency, always copy the entire DP master system from the first station to the DP master of the second station even after having made only minor changes. Copy the data by selecting **Edit > Insert Redundant Copy.**

  Execute the **Edit > Insert Redundant Copy** menu command to ensure consistency of the I/O addresses of the DP slaves at both stations.

  Moreover, if you want to use ET 200 distributed I/O devices such as ET 200B only on one station, configure those devices after having copied the DP master system (see also the description in the section How does a system with software redundancy work? (Page 17)).
- When configuring the hardware, remember that you can only use contiguous areas for software redundancy, for example outputs 0 to 20, bit memory address areas from 50 to 100, DP slave stations from 1 to 6, etc.
- Software redundancy supports standalone PROFIBUS DP master systems. If several DP master systems are required, you must implement several instances of software redundancy, that is, you need to several redundant subroutines.
- Valid baud rates for PROFIBUS DP

  Software redundancy supports only baud rates from 187.5 Kbaud to 12 Mbaud for the redundant DP slave interface module.

User program rules

- User program structuring

  If your user program in the two stations is only partially redundant, create a structure that separates the program component of the redundant device from that of the non-redundant device.

  Recommendation: Call the programs for the redundant and non-redundant part of the plant in different organization blocks:

  – Call the non-redundant part of the plant in OB1
  – Call the redundant part of the plant in OB35
- Redundant user program

  The redundant user program is included in two block calls of FB 101 ‘SWR_ZYK’. The first call of FB 101 ‘SWR_ZYK’ returns parameter CALL_POSITION=TRUE, while the second returns parameter CALL_POSITION=FALSE.

- Communication

  If using an S7 connection both for the redundancy link and also for other communication tasks, the job number R_ID must be greater than 2. The job numbers R_ID= 1 and R_ID=2 are reserved for software redundancy.

  If you are using 'FB 103 'SWR_SFCCOM' for communication, the communication blocks SFC 65 ‘X_SEND’ and SFC 66 ‘X_RCV’ with job numbers R_ID > 8000 0000_H are used for software redundancy.

  If you are using FB104 'SWR_AG_COM' for communication, software redundancy uses the communication blocks FC5 ‘AG_SEND’ and FC6 ‘AG_RCV’ with the job numbers R_ID > 8000 0000_H.

  If using FB 105 'SWR_SFBCOM' (BSEND, BRCV) for communication, you should always set "Send operating status messages ‘Yes’ " in your connection configuration so that any communication failure can be detected at the earliest time possible.

- Using timers and counters

  As a general rule, S7 timers and S7 counters cannot be used in the redundant software component as they cannot be updated. Use the IEC timers and IEC counters instead.

  It makes no sense to update a timer with a time setting less than the timer OB cycle, or less than the transfer time from the master to the standby station. You can also use S7 timers in such scenarios.

  If longer times are required, or if counters are used, you must make sure that the input signal edge for starting the timer/counter is detected reliably in changeover situations. You can achieve this by setting 1 to 0 or 0 to 1 pulses that are longer than the changeover time. Signal edge evaluation always has to be triggered both on the master and on the standby station if this condition is not met. The corresponding IEC timers/counters must not be updated. However, you can use S7 timers and S7 counters to cover this scenario.

Handling the blocks for software redundancy

- It is a prerequisite for proper generation of the multi-instance DB for software redundancy that all SFC and SFB system functions used for software redundancy have been installed in the S7 project.
- After having made changes to the configuration at startup block ‘SWR_START’, you must delete the following blocks to enable activation of new parameters and to prevent malfunctions:

DB_WORK_NO	Working DB for software redundancy
DB_SEND_NO	Send DB for software redundancy
DB_RCV_NO	Receive DB for software redundancy
DB_A_B_NO	DB for data exchange between the redundant software and the non-redundant component of the software of station A
DB_B_A_NO	DB for data exchange between the redundant software and the non-redundant component of the software of station B

OB 86 (rack failure)

You must not insert any tags in the first 20 bytes of the local tag of OB 86, as they are used and modified by software redundancy.

PIO in the software redundancy

Any parameter assignments of outputs in FC 100 ‘SWR_START’ that are not included in the PIO will lead to an I/O access error.

Master to standby changeover

Within the master to standby changeover phase the system temporarily operates with two masters or two standby stations.

Master to standby changeover by means of control bit

The status at the master and standby stations can be incorrect after a master to standby changeover was triggered by means of control bit. This scenario will develop if a slave fails during the changeover you initiated. To rectify this situation, repeat the master to standby changeover by means of control bit.

If only one CPU is in RUN (stand-alone mode)

It can happen that a CPU in STOP is assigned the active interface of the re-integrated redundant DP slave. Before you re-integrate a redundant DP slave, you must verify that one of the CPUs is switched off (POWER OFF).

Switching off a DP slave

If no other measure is taken, a master to standby failover is triggered after you switched off a DP slave. The measure to be taken to prevent the changeover is described in the example program below. Assumption: I 1.0 is the switch used to prevent the failover. This can also take the form of operator input or the like.

Example of OB 86 for switching off slaves without triggering a changeover:

```
 L #OB86_EV_CLASS
 L B#16#39
 ==I                                          //incoming event
 SPBN M001
 U I 1.0                                      //special input (in switched on
 SPBN M001                                    //Slave==1)-->no failover)
 AUF DB 3                                     //DB3 is the receiving DB (DB_EMPF)
 L DBW 4                                      //existing partner slave
 DEC 1                                        //reduce as initial measure
 T DBW 4                                      //to prevent the changeover
M001: NOP 0
 CALL "SWR_DIAG"                              //Call of FC 102 'SWR_DIAG'
 DB_WORK :=1                                  //Work DB for SWR
 OB86_EV_CLASS :=#OB86_EV_CLASS
 OB86_FLT_ID :=#OB86_FLT_ID
 RETURN_VAL :=MW14                            //block return value
```

Blocks for software redundancy

5

5.1 The library of blocks for software redundancy

The SWR_LIB library is available in STEP 7 after you installed the optional software package. You can access that library in SIMATIC Manager using menu command**File > Open > Libraries**

The SWR_LIB library contains five block packages. Of those packages, two are for use with S7-300 and three for use with S7-400. Always implement one of those packages to suit the connection type and network you are using to interconnect the two stations.

Block packages for S7-300

Select the package...	for this network...	and for this connection type...	Remark
XSEND_300	MPI	Non-configured connection	Network connection to the MPI interface of the CPU
AG_SEND_300	PROFIBUS	FDL connection	Network connection via CP 342-5
	Industrial Ethernet	ISO connection	Network connection via CP 345-1

Block packages for S7-400

Select the package...	for this network...	and for this connection type...	Remark
XSEND_400	MPI	Non-configured connection	Network connection to the MPI interface of the CPU
AG_SEND_400	PROFIBUS	FDL connection	Network connection via CP 443-5
	Industrial Ethernet	ISO connection	Network connection via CP 443-1
BSEND_400	MPI	S7 connection	Network connection via MPI interface of the CPU
	PROFIBUS		Network connection via CP 443-5
	Industrial Ethernet		Network connection via CP 443-1

See also

Content of the block packages (Page 30)

5.2 Content of the block packages

Each block package contains four blocks which are tuned for interactive operation. Never combine blocks from different block packages, as this could lead to malfunction at the stations.

Compatibility between V1.2 and V1.2 SP3

- You can replace the blocks of the predecessor version with blocks of Software Redundancy V1.2 SP3 without having to recompile the user program.
- Alongside with an update of the blocks in library partition SWR_AGSEND_300 or SWR_AGSEND_400 to V1.2 SP3 in the user program, the AG-Send (FC 5) and AG-Receive (FC 6) blocks in the "SIMATIC_NET_CP" library included with STEP 7 must also be updated.

Contents of block packages XSEND_300 and XSEND_400

Block	Comment
FC 100 'SWR_START'	The block must be called in the startup program OB 100.
FB 101 'SWR_ZYK'	The block must be called in the cyclic or time-controlled program. The block always has to be called before and after execution of the redundant user program.
FC 102 'SWR_DIAG'	The block must be called in diagnostics OB 86.
FB 103 'SWR_SFCCOM'	The block supports data transfers and is called in the background in FB 101 ‘SWR_ZYK’. You only have to load the block in both CPUs.

Contents of the block packages AGSEND_300 and AGSEND_400

Block	Comment
FC 100 'SWR_START'	The block must be called in the startup program OB 100.
FB 101 'SWR_ZYK'	The block must be called in the cyclic or time-controlled program. The block always has to be called before and after execution of the redundant user program.
FC 102 'SWR_DIAG'	The block must be called in diagnostics OB 86.
FB 104 'SWR_AG_COM'	The block supports data transfers and is called in the background in FB 101 ‘SWR_ZYK’. You only have to load the block in both CPUs.

Note

FB 104 'SWR_AG_COM' calls the FC 5 'AG_SEND' and FC 6 'AG_RCV' blocks in the background. Those blocks are components of NCM S7 and must be loaded in both CPUs.

Content of block package BSEND_400

Block	Comment
FC 100 'SWR_START'	The block must be called in the startup program OB 100.
FB 101 'SWR_ZYK'	The block must be called in the cyclic or time-controlled program. The block always has to be called before and after execution of the redundant user program.
FC 102 'SWR_DIAG'	The block must be called in diagnostics OB 86.
FB 105 'SWR_SFBCOM'	The block supports data transfers and is called in the background in FB 101 ‘SWR_ZYK’. You only have to load the block in both CPUs.

5.3 Overview of the blocks for software redundancy

The overview below lists all the blocks used for software redundancy:

Block	Block function
FC 100 'SWR_START'	The startup block provides the parameters and prepares them for further processing.
FB 101 'SWR_ZYK'	The cyclic block transfers data areas from the master to the standby station and coordinates communication and the changeover.
FC 102 'SWR_DIAG'	The diagnostics block executes the changeover and manages the diagnostics data of the slaves and prepares those data for FB 101 'SWR_ZYK'.
FB 103 'SWR_SFCCOM'	CPU communication by means of SFC 65 'X_SEND' and SFC 66 'X_RCV' is only relevant to MPI connections.
FB 104 'SWR_AG_COM'	CPU communication by means of FC 5 'AG_SEND' and FC 6 'AG_RCV' is relevant to PROFIBUS and Industrial Ethernet connections.
FB 105 'SWR_SFBCOM'	CPU communication by means of SFB 12 'BSEND' and SFB 13 'BRCV' is relevant to to MPI, PROFIBUS, Industrial Ethernet and point-to-point connections; those blocks cannot be used in S7-300.
DB_WORK_NO	Working DB for software redundancy
DB_SEND_NO	Data memory for redundant software: Send DB contains DBs, MBs, PIOs, DIs
DB_RCV_NO	Receive DB for redundant software components
DB_A_B_NO	Send-Receive DB for non-redundant data transferred from station A to station B
DB_B_A_NO	Send-Receive DB for non-redundant data transferred from station B to station A
DB_COM_NO	Instance DB for the communication blocks
FC 5 'AG_SEND'	This block is required if FDL connections are used for the redundant link.
FC 6 'AG_RCV'	This block is required if FDL connections are used for the redundant link.

NOTICE

The data blocks detailed above are generated with the required length once only at startup by FC 100 'SWR_START' (exception: DB_COM_NO). After changes were made to the parameters of FC 100 'SWR_START', it is usually required to edit the data blocks as well.

After changing the parameter settings for FC100 ‘SWR_START’, the CPU must always be restarted because if area lengths are changed, the send and receive DBs will have a new length and must be regenerated.

5.4 FC 100 'SWR_START'

Function

FC 100 'SWR_START' is used to initialize the two stations. Essentially, this block specifies the following:

- The I/O area of outputs, the bit memory address area, the data block area, the data blocks and the area for the instance DB for the IEC counters/timers that are used in the redundant user program. Each area must be assigned a contiguous range.
- Information pertaining to communication and to distributed I/Os.
- Three data blocks that the blocks for software redundancy require for storing internal data.

FC 100 'SWR_START' must called in startup OB 100.

Note on the parameterization of unused areas:

Enter the value 0 at the parameters for areas that are not used.

Example:

- If not using any IEC timers/counters, set the parameters IEC_NO = 0 and IEC_LEN = 0.
- If you have no outputs in the PIO area, assign parameter PIO_FIRST a value greater than PIO_LAST.

If not using the DB_A_B_NO and/or DB_B_A_NO data blocks, parameterize a user-specific DB number and a length with 0 value.

Example:

If not using DB_A_B_NO, set the parameters DB_A_B_NO = DB 255 and DB_A_B_NO_LEN = W#16#0. As the DB_A_B_NO and DB_B_A_NO data blocks are assigned data type Block-DB, you must set parameter values greater than DB 0 at those blocks, e.g. DB 255.

The data blocks DB_SEND_NO, DB_RCV_NO and DB_A_B_NO and DB_B_A_NO must be assigned consistent DB numbers on both stations.

Interruptibility

FC 100 'SWR_START' is interruptible.

Description of parameters

Parameters	Decl.	Data type	Description	Example
AS_ID	IN	CHAR	Station ID 'A' for station A. 'B' for station B.	'A'
DB_WORK_NO	IN	Block DB	Working DB for SW redundancy The DB only contains internal data.	DB1

Parameters	Decl.	Data type	Description	Example
DB_SEND_NO	IN	Block DB	DB in which data to be sent to the communication peer is collected. The DB only contains internal data.	DB2
DB_RCV_NO	IN	Block DB	DB in which the CPU collects the data received from the communication peer. The DB only contains internal data.	DB3
MPI_ADR	IN	INT	MPI address of the communication peer	4
LADDR	IN	INT	Logical base address of the communication processor (specified in the hardware configuration).	260
VERB_ID	IN	INT	Connection ID Number of the connection for the redundant link (specified in the hardware configuration).	1
DP_MASTER_SYS_ID	IN	INT	DP master system ID ID of DP master system to which the ET 200M slaves are connected (specified in the hardware configuration).	1
DB_COM_NO	IN	Block DB	Instance DB of FB 101 'SWR_ZYK'	DB5
DP-COMMUN	IN	INT	ID number for the DP master: 1, if the DP master is a CPU with integrated DP interface 2, if the DP master is a CP.	1
ADR_MODE	IN	INT	Increment size for the matrix in which the CPU allocates the I/O addresses (address matrix is CPU-dependent). 1, for base addresses 0, 1, 2, 3 ... 4, for base addresses 0, 4, 8, 12 ...	1
PIO_FIRST	IN	INT	Number of the first output byte used by an ET 200M with redundant IM 153.	0
PIO_LAST	IN	INT	Number of last output byte used by an ET 200M with redundant IM 153. Output bytes in the range PIO_FIRST to PIO_LAST must form a contiguous range and may only be used by the ET 200Ms with redundant IM 153. A maximum of 32 bytes of outputs can be configured for each redundant DP slave used.	4
MB_NO	IN	INT	Number of the first bit memory byte in the redundant user program	20
MB_LEN	IN	INT	Total number of bit memory bytes in the redundant user program. Bit memory bytes must be allocated contiguously.	30
IEC_NO	IN	INT	Number of the first instance DB for IEC counters/timers in the redundant user program.	111
IEC_LEN	IN	INT	Total number of instance DBs for IEC timers/counters in the redundant user program. Instance DBs must be allocated contiguously.	7
DB_NO	IN	INT	Number of the first data block in the redundant user program.	8

Parameters	Decl.	Data type	Description	Example
DB_NO_LEN	IN	INT	Total number of data blocks in the redundant user program. Data blocks must be allocated contiguously.	2
SLAVE_NO	IN	INT	Lowest PROFIBUS address for an ET 200M DP slave with redundant IM 153-2.	3
SLAVE_LEN	IN	INT	Total number of ET 200M DP slaves used. PROFIBUS addresses must be allocated contiguously.	1
SLAVE_DISTANCE	IN	INT	Identifier for the PROFIBUS address setting for IM 153-2: 1, if both interfaces have the same PROFIBUS address. 2, if the interfaces have PROFIBUS addresses n and n+1	1
DB_A_B_NO	IN	Block DB	Send DB for non-redundant data transferred from station A to station B.	DB11
DB_A_B_NO_LEN	IN	WORD	Number of data bytes used in DB_A_B_NO.	W#16#64
DB_B_A_NO	IN	Block DB	Send DB for non-redundant data transferred from station B to station A.	DB12
DB_B_A_NO_LEN	IN	WORD	Number of data bytes used in DB_B_A_NO.	W#16#64
RETURN_VAL	OUT	WORD	Block return value (for explanation see below)	MW2
EXT_INFO	OUT	WORD	Return value of a sublevel block (for explanation see below)	MW4

Block-specific values for RETURN_VAL and EXT_INFO

Error code	Explanation
W#16#0	No error
W#16#8001	Invalid value for parameter Teil-AG-Kennung.
W#16#8002	DB_WORK_NO could not be generated. Cause analysis by means of return value from SFC 22. Return value stored in EXT_INFO.
W#16#8003	DB_SEND_NO could not be generated. Cause analysis by means of return value from SFC 22. Return value stored in EXT_INFO.
W#16#8004	DB_RCV_NO could not be generated. Cause analysis by means of return value from SFC 22. Return value stored in EXT_INFO.
W#16#8005	DB_A_B_NO could not be generated. Cause analysis by means of return value from SFC 22. Return value stored in EXT_INFO.
W#16#8006	DB_B_A_NO could not be generated. Cause analysis by means of return value from SFC 22. Return value stored in EXT_INFO.
W#16#8007	Invalid value for parameter DP_MASTER_SYS_ID, or SLAVE_NO or SLAVE_LEN, or SLAVE_DISTANCE. Specified value does not match hardware configuration.
W#16#8008	Invalid value for parameter DP-KOMMUN if EXT_INFO=W#16#8888, or diagnostics could not be performed. Cause analysis by means of return value from SFC 51. Return value stored in EXT_INFO.
W#16#8009	Slave changeover lock could not be canceled. Cause analysis by means of return value from SFC 58. Return value stored in EXT_INFO.

Error code	Explanation
W#16#800A	Status of DP slave interface could not be determined. Cause analysis by means of return value from SFC 59. Return value stored in EXT_INFO.
W#16#800B	Error determining the PIO area used. Cause analysis by means of return value from SFC 50. Return value stored in EXT_INFO.
W#16#800C	Invalid value for parameter ADR_MODUS.
W#16#800D	Invalid value for parameter SLAVE_DISTANCE.
W#16#800E	DB_WORK_NO cannot be read. Reload blocks.
W#16#800F	Invalid value for parameter DP_COMMUN (no interfaces specified).
W#16#80F1	Error determining the addresses of the PAA. Cause analysis using the return value of SFC 50. Return value stored in EXT_INFO. Details specified for PIO_FIRST and PIO_LAST do not match hardware configuration.
W#16#8027	Internal error.

See also

Data type CHAR (Page 116)
Data type INT (Page 113)
Data type WORD (Page 113)

5.5 FB 101 'SWR_ZYK'

Function

FB 101 'SWR_ZYK' must be called **before and after** execution of the redundant user program. Use FB 101 'SWR_ZYK' to initiate the transfer of data from the master to the standby device.

After it was called, FB 101 automatically processes the data transfer from the master to the standby device. FB 101 calls the functions/function blocks required for data transfer in the background.

Interruptibility

FB 101 'SWR_ZYK' is interruptible.

Instance DB

An instance DB must be specified at the call of FB 101 'SWR_ZYK'. The block number of the instance DB must have been be set at parameter DB_COM_NO during parameterization of FC 100 'SWR-START'.

Description of parameters

Parameters	Decl.	Data type	Description	Example
DB_WORK_NO	IN	Block DB	Working DB. The specification must be identical with that for parameter DB_WORK_NO of FC 100 'SWR_START'.	DB1
CALL_POSITION		BOOL	This parameter defines the position for the call of FB 101 'SWR_ZYK' in the user program. TRUE ,if called before having been called in the redundant user program FALSE, if called after having been called in the redundant user program	TRUE
RETURN_VAL	OUT	WORD	Block return value (for explanation see below)	MW6
EXT_INFO	OUT	WORD	Return value of a sublevel block (for explanation see below)	MW8

Block-specific values for RETURN_VAL and EXT_INFO

Error code	Explanation
W#16#0	No error
W#16#8008	Invalid value for parameter DP-KOMMUN if EXT_INFO=W#16#8888, or diagnostics could not be performed. Cause analysis by means of return value from SFC 51.
W#16#800A	Status of the DP slave interface could not be determined. Cause analysis by means of return value from SFC 59. Return value stored in EXT_INFO.
W#16#800F	Invalid value for parameter DP_COMMUN (no interfaces specified).
W#16#8010	Changeover of DP slaves failed. Cause analysis by means of return value of SFC 58. Return value stored in EXT_INFO.
W#16#8011	Connection setup failed. Teil-AG-Kennung invalid.
W#16#8012	No job present in communication FB (FB 103 'SWR_SFBCOM'), (faulty instance DB, or internal error).
W#16#8013	Send error (FB 103 'SWR_SFBCOM', FB 104 'SWR_AG_COM', FB 105 'SWR_SFCCOM'). Cause analysis by means of return value from SFC 65 'X_SEND', FC 5 'AG_SEND', SFB 12 'BSEND'. Return value stored in EXT_INFO.
W#16#8014	Receive error (FB 103 'SWR_SFCCOM', FB 104 'SWR_AG_COM', FB 105 'SWR_SFBCOM'). Cause analysis by means of return value from SFC 66 'X_RCV', FC 5 'AG_RCV', SFB 13 'BRCV'. Return value stored in EXT_INFO.
W#16#8015	Redundant link failure. Check the hardware.
W#16#8016	Communication peer status cannot be read (FB 103 'SWR_SFCCOM'). Cause analysis by means of return value from SFB 23 'USTATUS'. Return value stored in EXT_INFO.
W#16#8017	All DP slaves have failed.
W#16#8018	Cannot write to Send DB (FB 104 'SWR_AG_COM', FB 105 SWR_SFBCOM'). Cause analysis by means of return value from SFC 20. Return value stored in EXT_INFO.
W#16#8019	Cannot read Receive DB (FB 104 'SWR_AG_COM', FB 105 SWR_SFBCOM').
W#16#8020	Internal error

See also

Data type BOOL (Page 114)
Data type WORD (Page 113)

5.6 FC 102 'SWR_DIAG'

Function

FC 102 must be called in diagnostics OB 86. Do not edit the block number.

FC 102 'SWR_DIAG' triggers execution of an automatic master to standby changeover after failure of a DP slave.

Interruptibility

FC 102 'SWR_DIAG' is interruptible.

Description of parameters

Parameters	Decl.	Data type	Description	Example
DB_WORK	IN	INT	Number of the working DB for software redundancy. This number must be consistent with the number specified at parameter DB_WORK_NO of FC 100 'SWR_START'. The DB only contains internal data.	1
OB 86_EV_CLASS	IN	INT	Start info from diagnostics OB 86. Copy the tag from the declaration table of OB 86.	#OB86_EV_CLASS
OB 86_FLT_ID	IN	INT	Start info from diagnostics OB 86. Copy the tag from the declaration table of OB 86.	#OB86_FLT_ID
RETURN_VAL	OUT	WORD	Block return value (for explanation see below)	MW14

Block-specific values for RETURN_VAL and EXT_INFO

Error code	Explanation
W#16#0	No error
W#16#80F2	Invalid value at a parameter of FC 102 'SWR_DIAG'.
W#16#80F3	More DP slaves present than specified in FC 100 'SWR_START'. Check parameter SLAVE_NO or SLAVE_LEN.

See also

Data type INT (Page 113)

Data type WORD (Page 113)

5.7 FB 103 'SWR_SFCCOM', FB 104 'SW_AG_COM' and FB 105 'SWR_SFBCOM'

Each one of the block packages in the SWR-LIB library contains one of the three function blocks specified above. The block numbers of FB 103, FB 104 or FB 105 may not be changed.

The function blocks are called in the background by 'FB 101 'SWR_ZYK' and organize the data transfer from the master to the standby device.

Make sure that the necessary block is loaded on both CPUs of the redundant system.

Note

If using FB 104 'SWR_AG_COM', the FC 5 'AG_SEND' and FC 6 'AG_RCV' blocks must also be available in your project. The block numbers for C 5 'AG_SEND' and FC 6 'AG_RCV' must not be changed.

5.8 Data blocks DB_WORK_NO, DB_SEND_NO and DB_RCV_NO

The DB_WORK_NO, DB_SEND_NO and DB_RCV_NO data blocks are defined when you parameterize FC 100 'SWR_START'.

Function

The data blocks are used exclusively for storing internal data.

Important note!

The data blocks detailed above are generated once only by FC 100 'SWR_START' at startup and with the required length. After changes were made to the parameters of FC 100 'SWR_START', it is usually required to edit the data blocks as well. For that reason, delete all the old data blocks so that new data blocks of the required length can be generated at startup.

Malfunctions can occur if you make changes to the parameterization of FC 100 'SWR_START' and do not delete the data blocks.

5.9 Data blocks DB_A_B and DB_B_A for the exchange of non-redundant data

You define the DB_A_B_NO and DB_B_A_NO data blocks when parameterizing FC 101 'SWR_START'. The DB length must be set at parameters DB_A_B_NO_LEN and DB_B_A_NO_LEN. Enter the length value "0" for DBs that you do not use.

Function

Data blocks DB_A_B_NO and DB_B_A_NO are provided to enable the exchange of non-redundant data between both stations. Non-redundant data, for example, can reflect the status of an input module that is only installed in the central rack of station A (unilateral I/O).

Those two data blocks can be used to exchange any information between station A and station B. This usually involves non-redundant data that is evaluated only at one station and transferred to the second station.

This data exchange ensures data consistency at both stations. The redundant component of the user program can therefore exchange data with the non-redundant (standard) program.

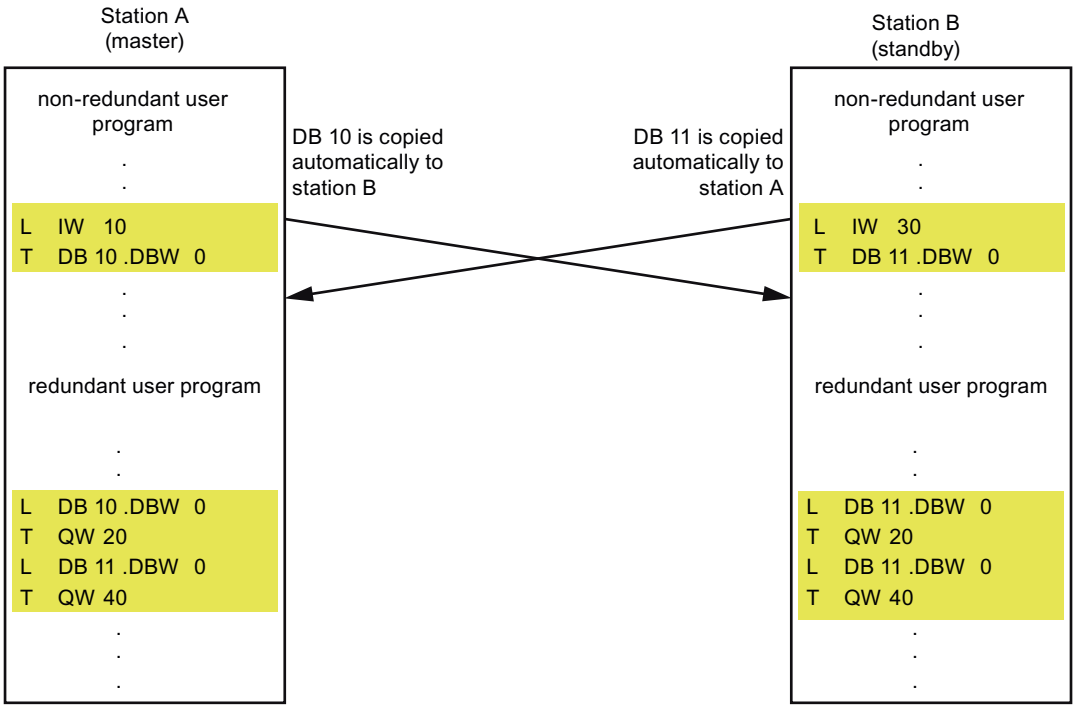
Example:

The CPU at station A contains unilateral I/O with input word IW 10, while the CPU of station B contains unilateral I/O with input word IW 30. The status of those input words is to be exchanged between the stations for display in the redundant program at the output words OW 20 and OW 40.

Procedure:

1. Specify the data blocks when parameterizing FC 100 'SWR_START', e.g. DB_A_B = DB 10 and DB_B_A = DB 11.
2. Program the necessary sequences in the user program of station A and station B.

Checking functionality



5.10 Data block DB_COM_NO

Data block DB_COM_NO is the instance DB of FB 101 'SWR_ZYK' and is defined when you parameterize FC 100 ’SWR_START’.

Function

In addition to the internal data for communication, DB_COM_NO also contains the status word and control word. DB_COM_NO is the instance DB of FB 101 'SWR_ZYK'.

Important note!

DB_COM_NO is the instance DB of FB 101 'SWR_ZYK' and is generated by STEP 7.

To enable generation of the block, all of the FB and SFC system functions used for software redundancy (SFB, SFC) must be available in your project. For a list of system functions used, refer to the chapter Technical data of the blocks (Page 47).

Structure of the data block

DBW	Meaning	Contents
0...6	Internal data	Input and output parameters of FB 101 'SWR_ZYK'
8	Status word	Status word of software redundancy Structure of the status word of software redundancy
10	Control word	Control word of software redundancy Structure of the control word of software redundancy
12 upwards	Internal data	Irrelevant

5.11 Example for getting started with a basic configuration

In order to get you started, we have compiled two example programs on the CD that are copied to the STEP 7 project directory by the installation program.

These are two executable examples: One example for S7-300 and another for S7-400. We selected CPU 315-2DP for the S7-300 example and CPU 414-2DP for the S7-400 example. The MPI interfaces of the CPUs were used for redundant coupling in both examples.

Of course, you can modify the examples to suit your requirements and, for example, use other CPUs. For such scenarios you must modify the hardware configuration accordingly.

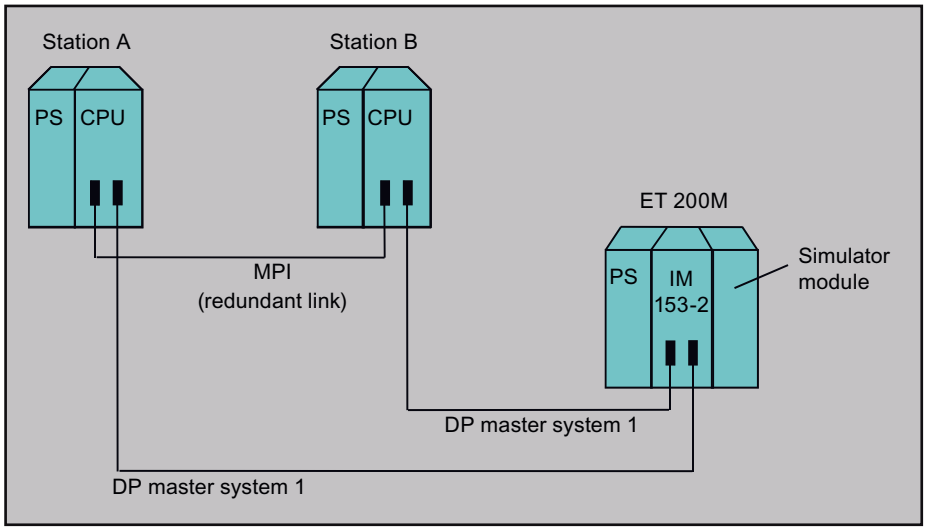
Hardware components for the S7-300 example

A basic configuration was selected for the S7-300 example. Each one of the two stations consist of a mounting rail, a power supply module and a CPU 315-2DP. The ET 200M distributed I/O device consists of a power supply module, a DP slave interface module IM 153-2 and a simulator module (1 byte for inputs and 1 byte for outputs, address 0).

Hardware components for the S7-400 example

A basic configuration was selected for the S7-400 example. Each one of the two stations consist of a rack, a power supply module and a CPU 414-2DP. The ET 200M distributed I/O device consists of a power supply module, a DP slave interface module IM 153-2 and a simulator module (1 byte for inputs and 1 byte for outputs, address 0).

Overview: Hardware configuration for the S7-300 or S7-400 example



Proceed as follows:

1. Open the example project.
2. Transfer the "hardware configuration" to station A and station B.
3. Transfer all blocks from both block containers to the relevant station.
4. Only for S7-400: Transfer the connection configuration to both stations.

Checking functionality

Set both stations to RUN and verify their proper functioning by checking the following states at each program using the tag table VAT1:

1. Read the status word from sation A (DB5.DBW8).

   The value 1000 0000 0000 0101 should be displayed. Meaning: The station is subunit A and master; all DP slaves can be addressed.

2. Read the status word from station B (DB5.DBW8):

   The value 1000 0000 0000 0101 should be displayed. Meaning: The station is subunit B and standby; all DP slaves can be addressed.

3. Set the bit in the control word for master to standby changeover (DB5.DBX10.0) and once again check the status.

   The status word bits DBX 9.0 and DBX 9.1 should change their status at both stations. The active interface of the IM 153-2 should also change.

5.12 Technical data of the blocks

Technical data of the blocks

Block	Memory requirements	System functions used
FC 100 'SWR_START'	2.6 Kbytes	SFC 22 'CREATE_DB', SFC 5 'GADR_LGC', SFC 50 'RD_LGADR', SFC 46 'STP', SFC 47 'WAIT'
FB 101 'SWR_ZYK'	3.7 Kbytes	SFC 64 'TIME_TCK', SFB 3 'TP'
FC 102 'SWR_DIAG'	2 Kbytes	SFC 51 'RDSSYST', SFC 58 'WR_REC', SFC 59 'RD_REC'
FB 103 'SWR_SFCCOM'	1.5 Kbytes	SFC 20 'BLKMOV', SFC 65 'X_SEND', SFC 66 'X_RCV'
FB 104 'SWR_AG_COM'	1.5 Kbytes	SFC 20 'BLKMOV', FC 5 'AG_SEND', FC 6 'AG_RCV'
FB 105 'SWR_SFBCOM'	1.5 Kbytes	SFB 12 'BSEND', SFB 13 'BRCV', SFB 23 'USTATUS'

References and supplementary information 6

6.1 Features and properties of software redundancy

The overview below summarizes the essential features of software redundancy:

Feature	Description/explanation
System availability	The system consists of two CPUs. One CPU, namely the master CPU (master station), executes the user program and also transfers the necessary information to the second CPU, namely the standby CPU (standby station) that requires this information to continue execution of the redundant (component of the) user program in the event of a fault. Instead of executing the available redundant user program, the standby station only executes its local, non-redundant user program. The second CPU continues execution of the user program after failure of the first CPU (master-standby principle).
Time for transferring updates from the master to the standby device	The update time depends on the CPU, the network used, or the communications protocol used, and on the size of the user program. See also: Duration of the data transfer from master to standby (Page 52)
Time required for the master to standby changeover	The time it takes to complete this depends on the reason for failover, the duration of the data transfer, and on the number of DP slaves connected. See also: Duration of the master to standby changeover (Page 51)
User program	Complete or only partially identical user programs are possible on both CPUs.
Programming languages	LAD, FBD, STL and SCL Software redundancy cannot be used with CFC.
Using standard function blocks	All function blocks can be used. Exception: Blocks that use S7 timers and/or S7 counters; only IEC timers or IEC counters are allowed.
Use of standard software controllers	No restrictions compared to standard SIMATIC S7. Exception: Blocks that use S7 timers and/or S7 counters
Interrupt processing in the user program	No restrictions compared to standard SIMATIC S7 However, interrupts can be lost during a master to standby changeover (interrupt processing could be discarded).
Supported number of ET 200M DP slaves	Depends on the CPU used (up to 64 ET 200M DP slaves are supported on CPU 414-2DP)
Digital/analog I/Os	All digital and analog modules that can be operated on the ET 200M distributed I/O device.
Function modules	ET 200M supports the FM 350 counter module.
Valid maximum volume of redundant data to be transferred	8 Kbytes for S7-300 64 Kbytes for S7-400
Second/ third and n fault	Only initial faults can be managed. That is, if a successive fault occurs while a fault is being processed, it could happen that execution of the redundant program is canceled, for example.

6.2 Master to standby changeover

Definition:

The term master-standby failover describes a scenario where the CPUs change their master/standby status and the DP slave interfaces change their active side.

Causes of a master to standby changeover

A master to standby changeover can can be caused by different events:

- Request for master to standby changeover at user level
  (Bit set in the control word)
- Failure of the master station (POWER OFF or STOP)
- Fault in the DP master system of the master device
- Failure of a redundant DP slave interface module

See also

Duration of the master to standby changeover (Page 51)

How does a system with software redundancy work? (Page 17)

6.3 Duration of the master to standby changeover

The worst-case time it takes for a master to standby changeover includes:

- The fault detection time
- The data transfer time
- The time required for changeover of the DP slaves

Worst-case scenario:

Duration of the master to standby changeover =	The fault detection time + the data transfer time + the time required for changeover of the DP slaves

See also

Duration of the data transfer from the master to the standby device (Page 52)
Changeover times for DP slaves of ET200M (Page 54)
Fault detection times in the redundant system (Page 56)

6.4 Duration of the data transfer from the master to the standby device

The time it takes to transfer the data from the master to the standby device depends on several factors:

- Communication performance of the CPU used
- Network, connection type used and the transmission rate
- Volume of data to be transferred

As a rule, it is not possible for all data to be transferred from one station to the other within a single cycle. In order to prevent excess load on the program cycle as a result of the data transfer, the data is fragmented and transferred in small packets in several cyclic frames.

The volume of data transferred includes the PIO area, the bit memory address area, the data block area you specified in FC 100 'SWR_START' and other internal data.

Rule of thumb for estimating the volume of data transferred

The following rule of thumb has proved reliable for estimating the volume of data transferred:

- Data volume = 3 x the number of output bytes used

The tables below show the typical transfer times for CPU 315-2DP and CPU 414-2DP:

Transfer times at a redundant system with two 315-2DP CPUs

Data transfer with FB 104 'SWR_AG_COM' is segmented in blocks of 240 bytes, and with FB 103 'SWR_SFCCOM' in blocks of 76 bytes.That is, only one block can be transferred per call of software redundancy. The volume of data to be transferred therefore depends on the call interval for software redundancy.

Transmission rate on PROFIBUS (AG_SEND): 187.5 kbaud to 1.5 Mbaud	Transmission rate on Industrial Ethernet (AG_SEND): 10 MBaud	Transmission rate with MPI connection (XSEND): 187.5 kbaud
60 ms per block with 240 bytes	48 ms per block with 240 bytes	152 ms per block with 76 bytes

Note on the table for CPU 315-2DP:

The times specified apply to networks to which only the two stations of the redundant system are connected. The redundant user program is written in OB 1. OB 1 is has a maximum runtime of 10 ms.

The transfer can take longer if more than two nodes are connected to the network, depending on the selected baud rate. The transfer time is nearly constant at transmission rates of 1.5 MBaud and 10 MBaud.

Transfer times at a redundant system with two 414-2DP CPUs

Number of bytes to be transferred	Transfer rate on PROFIBUS/Industrial Ethernet from 187.5 kbaud to 12 Mbaud	Transfer time for the MPI connection at 187.5 kbaud
1 Kbytes	250 ms	340 ms
4 Kbytes	1 s	1.36 s
16 Kbytes	4 s	5.44 s
64 Kbytes	16 s	21.76 s

Note on the table for CPU 414-2DP:

The times quoted are valid for networks to which only the two stations of the redundant system are connected and if communication is handled by means of the BSEND/BRCV blocks.

The transfer can take longer if more than two nodes are connected to the network, depending on the selected baud rate.

The transfer time can be prolonged (CPU 412) or be reduced (CPU 416), depending on the communication performance (communication bus) of the CPU.

6.5 Changeover times for DP slaves of ET200M

During master to standby changeover, the ET 200M DP slaves are automatically changed over from the DP master system of the master to the DP master system of the standby device. An S7-300 supports the automatic changeover of up to four DP slaves per call interval, while the S7-400 supports up to eight DP slaves accordingly. More than four, or more than eight DP slaves are changed over in groups at several call intervals.

Requirements for the call interval of OB 1 or OB 35

The call interval between two OB 1, or between two timer OBs, must always be greater than the changeover time for four or eight DP slaves. The call interval may be only be shorter (see table for times) if you are using less than four or eight DP slaves.

CPU 315-2DP with integrated DP master

Number of DP slaves	12 MBaud	1.5 MBaud	500 kbaud	187.5 kbaud
1	6 ms	6 ms	7 ms	12 ms
2	12 ms	12 ms	14 ms	24 ms
4	25 ms	25 ms	30 ms	50 ms
8	2 x 25 ms	2 x 25 ms	2 x 30 ms	2 x 50 ms
16	4 x 25 ms	4 x 25 ms	4 x 30 ms	4 x 50 ms
32	8 x 25 ms	8 x 25 ms	8 x 30 ms	8 x 50 ms
64	16 x 25 ms	16 x 25 ms	16 x 30 ms	16 x 50 ms

CPUs of S7-400 stations with integrated DP Master

Number of DP slaves	12 MBaud	1.5 MBaud	500 kbaud	187.5 kbaud
1	5 ms	9 ms	13 ms	20 ms
2	10 ms	18 ms	26 ms	40 ms
4	20 ms	36 ms	39 ms	80 ms
8	40 ms	64 ms	78 ms	160 ms
16	2 x 40 ms	2 x 64 ms	2 x 78 ms	2 x 160 ms
32	4 x 40 ms	4 x 64 ms	4 x 78 ms	4 x 160 ms
64	8 x 40 ms	8 x 64 ms	8 x 78 ms	8 x 160 ms

CP (CP 443-5) as DP master for an S7-400 station

Number of DP slaves	187.5 kbaud to 12 Mbaud
1	55 ms
2	100 ms
4	200 ms
8	400 ms
16	2 x 400 ms
32	4 x 400 ms
64	8 x 400 ms

6.6 Fault detection times in the redundant system

The tables below show the maximum fault detection times of the system and the system response to various causes of faults.

Faults on the master device

Fault cause	Fault detection time	Response
CPU of the master station in STOP or NETWORK OFF at the master station	Approx. 1 s*	The DP interfaces are automatically switched over to new master. Automatic master to standby changeover The status word indicates a "Redundant link failure".
Failure of the DP master in the master station or Failure of the entire DP master system at the master station	A few ms	The DP interfaces are automatically switched over to new master. Automatic master to standby changeover The status word indicates "No DP slave present"
*The fault detection time on systems with S7-400 is reduced from 1 s to 100 ms if the block package BSEND is used and the operating status messages are transferred automatically (must be parameterized in the connection configuration).		

Faults on the standby device

Fault cause	Fault detection time	Response
CPU of the standby device in STOP or NETWORK OFF at the standby device	Approx. 1 s	The master station ignores this state and continues operation without any changes. The status word indicates a "Redundant link failure".
Failure of the DP master in the standby device or Failure of the entire DP master system at the standby device	A few ms	The master station ignores this state and continues operation without any changes. The status word in the standby device indicates "No DP slave present".

Faults at the redundant link

Fault cause	Fault detection time	Response
Redundant link failure	Approx. 1 s*	Both stations assume the master mode. DP slaves remain assigned to the previous master. Failure of the CPU connection is reported. The status word indicates a "Redundant link failure".
*At longer call intervals (> 1 s) of software redundancy, the time for the detection of redundancy failure takes at least three to four call intervals.		

Faults at distributed I/O devices

Fault cause	Fault detection time	Response
Failure of the ET 200M DP interface (IM 153-2) connected to the master station	A few ms	The DP interface in ET 200M is changed over to the standby device. All other DP slaves are changed over to the standby device. Automatic master to standby changeover.
Failure of the ET 200M DP interface (IM 153-2) connected to the standby device	A few ms	No response at the master station. The master continues operation as before. The status word of the standby device indicates "Not all (or no) DP slaves present".
Failure of the ET 200M (IM 153-2) power supply	A few ms	All addressable DP slaves are changed over. Automatic master to standby changeover.

6.7 Networks for linking the two stations

The two stations can always be linked via MPI, PROFIBUS, or Industrial Ethernet. However, due to its slow transmission speed the MPI link can only be used to transfer smaller data volumes (max. 1 KB).

You must copy the blocks for software redundancy from the specified library according to the logical connection configured.

Options of linking S7-300 stations

Stations are networked via…	Network connection via interface...	Transmission speed	Connection required	Blocks required in library...
MPI	CPU	187.5 kbaud	Non-configured connection	XSEND_300
PROFIBUS	CP 342-5	max. 1.5 MBaud	FDL connection	AS_SEND_300
Industrial Ethernet	CP 345-1	10 MBaud	ISO connection	AS_SEND_300

Options of linking S7-400 stations

Stations are networked via…	Network connection via interface...	Transmission speed	Connection required	Blocks required in library...
MPI	CPU	187.5 kbaud	Non-configured connection	XSEND_400
			S7 connection	BSEND_400
PROFIBUS	CP 443-5	max. 12 MBaud	FDL connection	AS_SEND_400
			S7 connection	BSEND_400
Industrial Ethernet	CP 443-1	10 MBaud	ISO connection	AS_SEND_400
			S7 connection	BSEND_400

6.8 Editing configuration data and the user program in RUN

It is usually necessary to disable redundancy before you make any changes in runtime (CiR). Set the ‘Disable redundancy’ bit accordingly in the control word at user level. After that bit has been set, the master station continues execution of the user program as before. In that situation, the master station has the same properties as any standard S7-300 or S7-400 device.

After having disabled redundancy, edit the user program on the 'standby device' and then on the 'master device'. Set the ‘Enable redundancy’ bit in the control word after having downloaded the modified user program to both CPUs. After that bit has been set, the redundant link is restored and the system operates with increased availability again.

It is not possible to change the range of redundant data areas. Data areas are also changed by a new FB call, as this involves creation of a new instance DB. The data content can be edited, of course, as long as the range of the data area remains the same. Any change to the length of a data block also affects the range of areas used for the redundant data.

Tip: Make sufficient allowances for the volume of your data if expansions can be expected at runtime.

The sections below provide a description of procedures for editing the program and the configuration of redundant software, as well as of integration mechanisms.

Editing the program at the redundant software component in RUN

Proceed as follows:

1. Disable redundancy (by setting bit 11.0 of the control word)
2. Edit and test the user program on the standby CPU
3. Re-enable redundancy (by setting bit 11.1 of the control word)
4. If necessary, execute a master to standby changeover

Result: The CPU executes the modified user program after the master to standby changeover was completed.

You can now edit the program in the second CPU in the same way.

It is not possible to change the range of redundant data areas.

Re-integrating a failed DP slave of ET 200M (IM 153-2) in the redundant component

You have two alternatives:

- Replacement of the faulty interface module
- Recovery of the power supply

Result: The software redundancy function automatically reconnects the DP slave with the interface module that is assigned to the master CPU.

Integrating a new DP slave ET 200M (IM 153-2) in the redundant component

Proceed as follows:

1. Disable redundancy (by setting bit 11.0 of the control word)
2. Set the standby CPU to STOP
3. Configure the new DP slave and transfer the hardware configuration.
4. Edit the relevant parameters in the call of FC 100 'SWR_START' (PAA_FIRST, PAA_LAST, SLAVE_NO, SLAVE_LEN).
5. Delete the DB_WORK_NO, DB_SEND, DB_RCV, DB_A_B_NO and DB_B_A_NO data blocks
6. Switch the CPU to RUN again (this CPU is operating with redundant data that has not been updated)
7. Set the other CPU to STOP (the CPU with the new configuration takes over control of the process)
8. Configure the new DP slave and transfer the hardware configuration
9. Edit the relevant parameters in the call of FC 100 'SWR_START' (PAA_FIRST, PAA_LAST, SLAVE_NO, SLAVE_LEN)
10. Delete the DB_WORK_NO, DB_SEND, DB_RCV, DB_A_B_NO and DB_B_A_NO data blocks
11. Switch the CPU to RUN again

Result: The new DP slave ET 200M is now integrated in the redundant software component.

Note: You can carry out an upgrade without having to reinstall the redundant area by using a second, self-contained redundancy program with inherent data area. This add-on redundancy program manages the additional, new data areas.

CPU replacement or firmware update

Proceed as follows:

1. Switch the CPU to be replaced to STOP
2. Replace the CPU and transfer the hardware configuration, the user program blocks and the connection configuration
3. Switch the CPU to RUN again

Result: The new CPU operates in standby mode.

Removing and inserting I/O modules

I/O modules can be removed and inserted in the same way as with standard S7. You must safely prevent any master to standby changeover while replacing a module, for example by disabling redundancy (inhibit master to standby changeover).

6.9 Modules that support software redundancy

The following modules currently support systems with software. The range of those modules is continuously expanded.

For the updated lists of modules for systems with software redundancy, refer to the SIMATIC FAQs at (http://support.automation.siemens.com)

Supported CPUs

Designation	Order number
S7-300	
CPU 313C-2DP	6ES7313-6CE00-0AB0
CPU 314	6ES7314-1AG13-0AB0
CPU 314C-2DP	6ES7314-6CF0x-0AB0 6ES7314-6CG0x-0AB0
CPU 315-2DP	6ES7315-2AFxx-0AB0 6ES7315-2AG10-0AB0 6ES7 315-2AH14-0AB0
CPU 315-2 PN/DP	6ES7 315-2EG1x-0AB0 6ES7 315-2EH1x-0AB0 6ES7 315-2FH1x-0AB0 6ES7 315-6Tx1x-0AB0
CPU 315F-2 PN/DP	6ES7 315-2FJ1x-0AB0
CPU 316-2DP	6ES7316-2AGxx-0AB0
CPU 317-2	6ES7 317-2AJ10-0AB0
CPU 317-2 PN/DP	6ES7 317-2Ex1x-0AB0 6ES7 317-6FF0x-0AB0 6ES7 317-6Tx1x-0AB0
CPU 317F-2 PN/DP	6ES7 317-2Fx1x-0AB0
CPU 318-2DP	6ES7318-2AJxx-0AB0
CPU 319-3 PN/DP	6ES7 318-3xL0x-0AB0
S7-400	
CPU 412-1	6ES7412-1XFxx-0AB0 6ES7412-1FK03-0AB0 6ES7 412-1XJ05AB0
CPU 412-2	6ES7412-2XGxx-0AB0 6ES7 412-2XJ05-0AB0
CPU 413-1	6ES7413-1XGxx-0AB0
CPU 413-2DP	6ES7413-2XGxx-0AB0
CPU 414-1	6ES7414-1XGxx-0AB0
CPU 414-2DP	6ES7414-2XGxx-0AB0 6ES7414-2XJxx-0AB0 6ES7 414-2XK05-0AB0
CPU 414-3DP	6ES7414-3XJxx-0AB0 6ES7 414-3XM05-0AB0 6ES7 414-3EM05-0AB0
CPU 416-1	6ES7416-1XJxx-0AB0

Designation	Order number
CPU 416-2DP	6ES7416-2XKxx-0AB0 6ES7416-2XLxx-0AB0 6ES7416-2XN05-0AB0
CPU 416F-2DP	6ES7 416-2FN05-0AB0
CPU 416-3DP	6ES7416-3XLxx-0AB0 6ES7 416-3XR05-0AB0 6ES7 416-3ER05-0AB0
CPU 416F-3DP	6ES7 416-3FR05-0AB0
CPU 417-4	6ES7417-4XLxx-0AB0 6ES7 417-4XT05-0AB0

Supported communications modules with DP master function

Designation	Order number
Communication module CP 443-5 Extended (for connection to PROFIBUS network)	6GK7443-5DXxx-0XE0
DP master interface module IM 467 or IM 467-FO (can only be used in version 1.1)	6ES74675GJxx-0AB0 6ES74675FJxx-0AB0

Supported communications modules for linking the stations

Designation	Order number
Communications module CP 342-5	6ES7342-5DA00-0XE0 6GK7342-5DA02-0XE0
Communications module CP 343-1	6GK7343-1BA00-0XE0 6GK7343-1EX11-0XE0
Communications module CP 343-1 Lean (for connection to Industrial Ethernet)	6GK7343-1CX10-0XE0
Communications module CP 443-5 Extended (for connection to PROFIBUS network)	6GK7443-5DXxx-0XE0
Communications module CP 443-1 ISO1 (for connection to Industrial Ethernet)	6GK7443-1BXxx-0XE0 6GK7443-1EXxx-0XE0 6GK7443-1GXxx-0XE0

Modules supported for operation on the ET 200M Distributed I/O Device

Designation	Order number
2 x DP slave interface module IM 153-2	6ES7153-2AA02-0XB0, product version 2 or higher (bus module 6ES7 7HD00-0XA0) 6ES7153-2AB0x-0XB0, product version 2 or higher (bus module 6ES7 7HD10-0XA0)
All digital and analog modules for ET 200M	See catalog ST70

Designation	Order number
Counter module FM 350	6ES7350-1AH0x-0AE0
CP 341 (20 mA TTY, RS232, RS422/485)	6ES7341-1xH01-0AE0
CP 341 (RS232) (point-to-point connection)	6ES7341-1AH02-0AE0

Note

ET 200M stations must always be set up with active bus modules (6ES7195-7HB00-0XA0 or 6ES7195-7HC00-0XA0), even when the function "removing and inserting I/O modules in Run mode of the CPU" is not possible due to the design principle of S7-300 CPUs.

NOTICE

If operating two DP segments in redundant mode, you must configure both segments for operation in DPV1 or S5-compatible mode.

6.10 Communication with other stations

A system with software redundancy is, of course, capable of communicating with other stations. The following sections demonstrate a number of possible solutions.

As it it not allowed to operate communication modules in the ET 200M distributed I/O device, communication must be handled using the CPs installed in the CPU.

In order to increase the availability of communication tasks, you must insert one CP in the CPU of station A, and a second in the CPU of station B.

See also

Communication with an S7-300/S7-400 station (Page 65)

Communication with a second system with software redundancy (Page 67)

6.11 Communication with an S7-300/S7-400 station

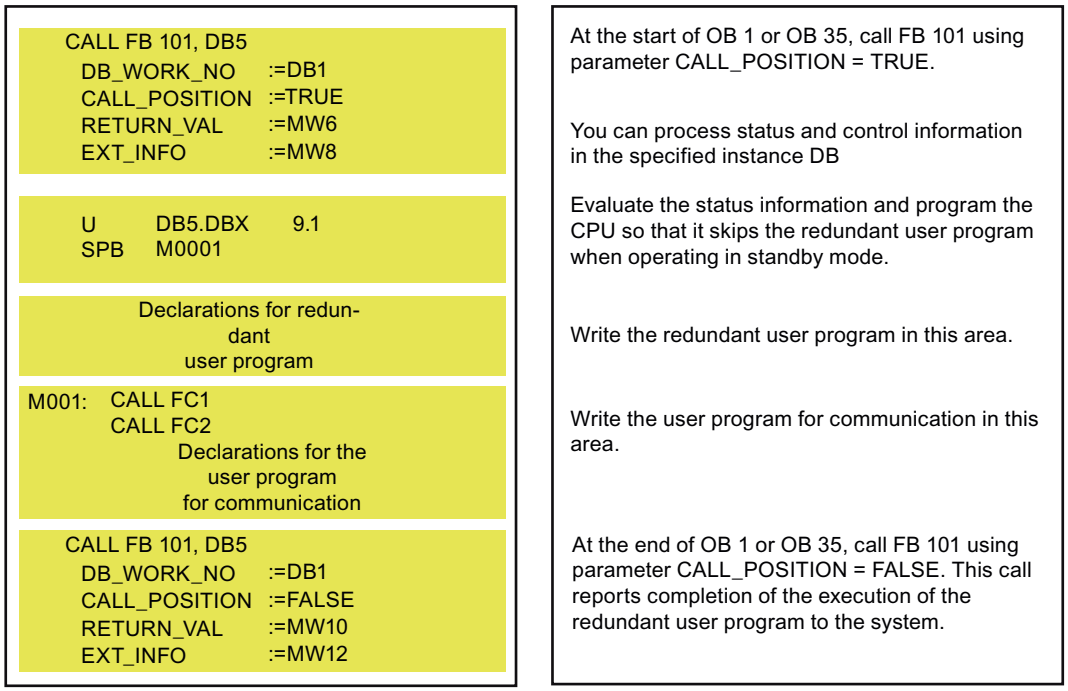
Configuring the connections to the standard system

1. Configure one connection from station A to the S7-300/400 target station
2. Configure a second connection from station B to the S7-300/400 target station

User program for station A and B

The standby device also has to process the communication modules in order to prevent communication failure. For that reason, we recommend the following structure for the redundant user program:

Cyclic program OB 1, or time-controlled program OB 35

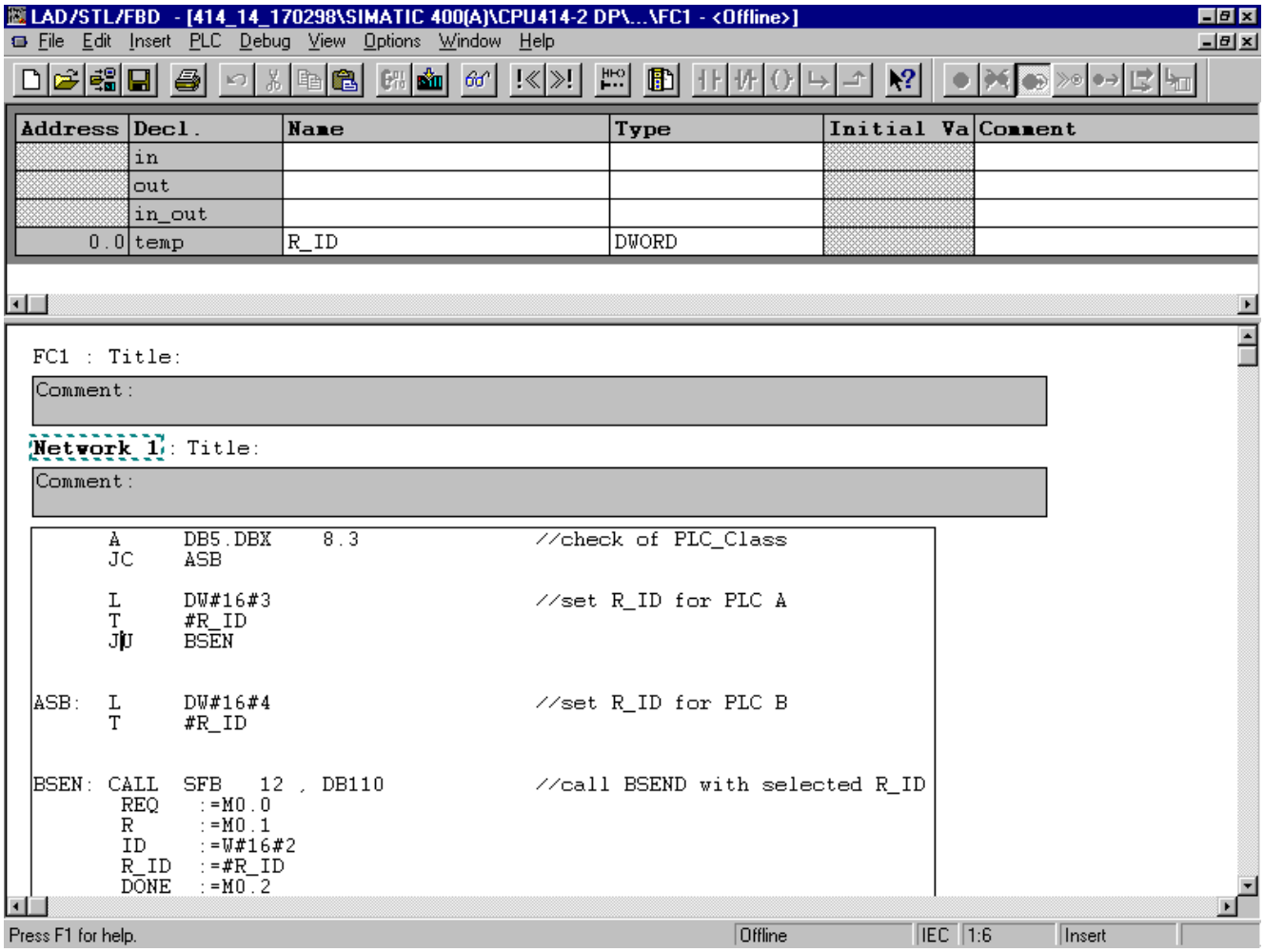


Program the calls for the communication blocks in FC 1. Note that that at least the job number R_ID must be different for station A and station B.

The status word should be included in the data area transferred, so that the target device can detect the active connection. Any further analysis of the data received takes place only on the master device.

When writing the user program in CFC, start by programming FC 1 in LAD, FBD or STL. The block may not contain any process tags or message numbers.

Example of a program sequence in FC 1



6.12 Communication with a second system with software redundancy

Configuring the connections

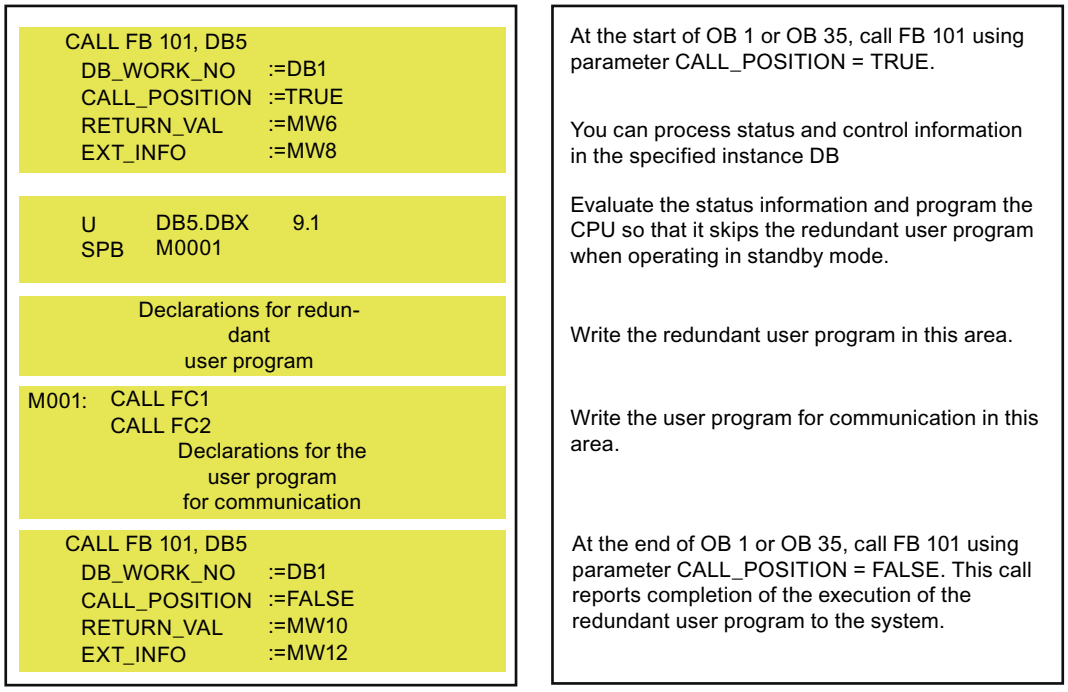
You must configure altogether four connections so that the two systems can independently initiate a changeover.

1. Configure two connections from station A to the redundant system
2. Configure two connections from station B to the redundant system

User program for station A and B

The standby device also has to process the communication modules in order to prevent communication failure. For that reason, we recommend the following structure for the redundant user program:

Cyclic program OB 1, or time-controlled program OB 35

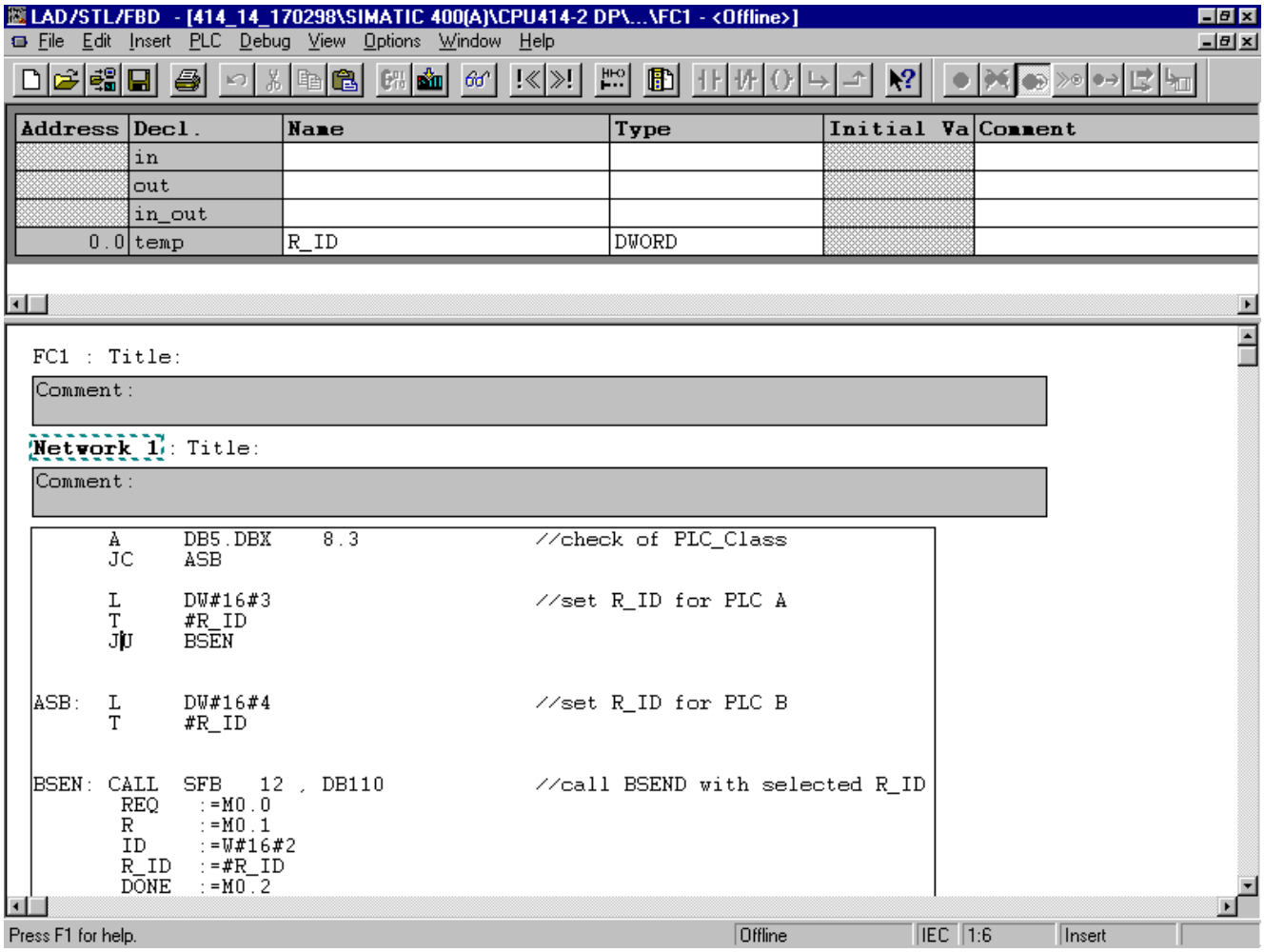


Program the communication block calls in FC 1 for station A of the communication peer. Program the communication block calls in FC 2 for station B of the communication peer. Note that that at least the job number R_ID must be different for station A and station B.

The status word should be included in the data area transferred, so that the target device can detect the active connection. Any further analysis of the data received takes place only on the master device.

When writing the user program in CFC, start by programming FC 1 in LAD, FBD or STL. The block may not contain any process tags or message numbers.

Example of a program sequence in FC 1 or FC 2



6.13 Standby concept for software redundancy

In addition to the standard scenario in which two stations form a master/standby configuration, there is also another variant referred to as the standby concept.

The term standby concept may be new to you but you are bound to be familiar with the principle of the standby concept. You are surely familiar with the organizational concept in automotive industry where one person always acts as standby for any absent employees, namely the standby concept we are referring to.

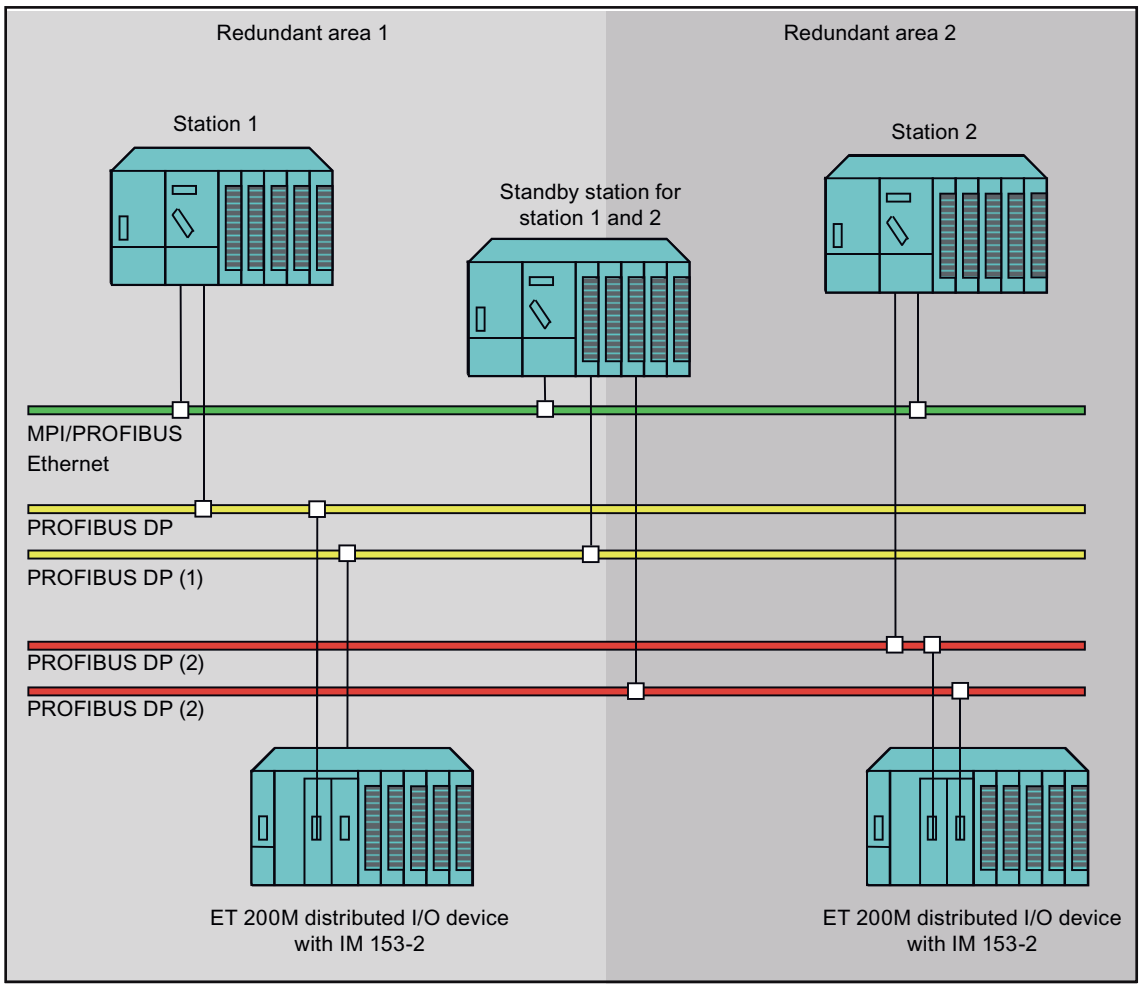
The standby concept for software redundancy works in just the same way. If one or several stations fail (station 1 or station 2 in the diagram), a standby device (station R in the diagram) takes over the task of the failed device.

What do I have to observe in terms of the standby concept for software redundancy?

There are basically three requirements for the standby concept:

- There must be one redundant link (connection) between station 1 and station R and a second between station 2 and station R
- The user programs d station 1 and station 2 must have been loaded in the standby device (station R)
- The standby device (station R) must be able to access the ET 200M distributed I/O devices of station 1 and station 2 (there are two DP masters on station R)

Standby redundancy



6.14 Using error OBs

To prevent the system from going into STOP as a response to errors/events, you should make use of the option of assigning responses to priority classes (organization blocks).

To prevent the system from going into STOP as a response to the failure of a DP slave, the following error OBs should also be available on the CPU in addition to OB 86 (with FC 102 'SWR_DIAG'):

- OB 80: A watchdog interrupt can be generated during a master to standby failover.
- OB 82: Diagnostics interrupt from a module in the redundant DP slave interface module (e.g. IM 153-2)
- OB 83: Removal/insertion interrupt from modules in the DP slave interface module
- OB 85: Program execution error; occurs after failure of a DP slave interface module.
- OB 87: Communication error
- OB 122: I/O access error (failure of IM 153-2, or of a module in the station).

Those OBs enable the response to corresponding faults in the user program. Software redundancy neither evaluates those OBs, nor initiates any further response.

You can enhance availability by loading additional interrupt OBs.

Example: Software redundancy with S7-300 7

7.1 Example: Software redundancy with S7-300

We have created a project template to get you started with minimum effort. The project template can be executed and be modified to suit your requirements.

Using the simplified model of a road tunnel supervision system, we show you how simple it is to create the necessary configuration and programs. The example is based on two stations with a 315-2DP CPU.

Details on particular tasks and settings specific to software redundancy are provided on the following pages. General information that you already know from configuring and programming an S7-300 or S7-400, such as creating a project or configuring the CPU, is included only where it is necessary for comprehension of the examples.

7.2 Definition of the task and the technology scheme

Description of the task

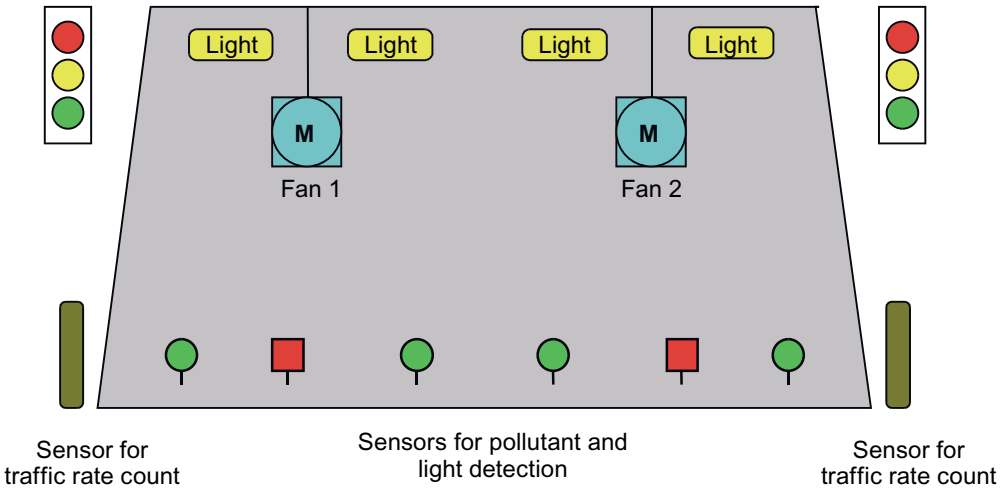
Two fans are used to ventilate a tunnel. Each fan has two speeds (stages) which are activated depending on the measured pollutant concentration in the air. The pollutant concentration is measured by means of two analog sensors.

The fans are central components of the plant with higher demands on availability. The user program for controlling the fans is therefore loaded on both stations.

The daily traffic rate in the tunnel is recorded for statistical purposes. Vehicles entering and leaving the tunnel are detected by road sensors at each end of the tunnel. This element requires only the availability level provided by standard S7 and is therefore loaded only on station A.

The lighting is monitored by means of four binary sensors. A lighting failure in any one of the four sections is indicated by means of corresponding binary output signal. This element requires only the availability level provided by standard S7 and is therefore loaded only on station B.

Technology scheme "Tunnel supervision"

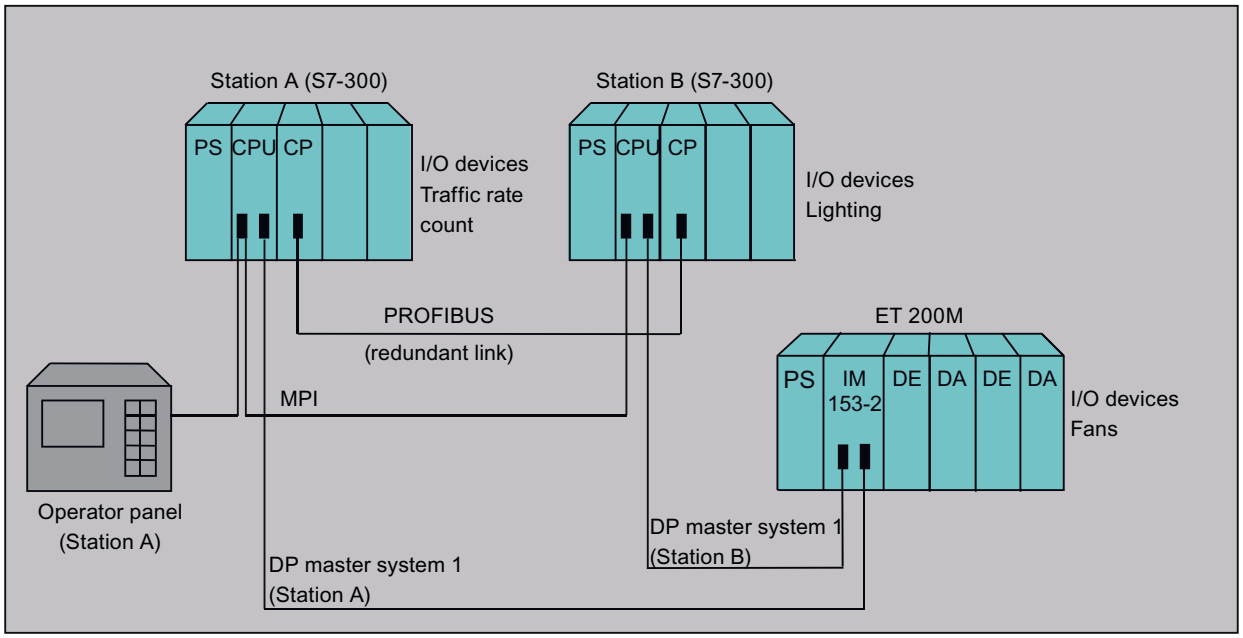


7.3 Hardware configuration for the example with S7-300

The diagram below shows the hardware configuration required. It consists of two S7-300 stations, each one with a CPU 315-2DP and an ET 200M DP slave. The IM 153-2 DP interface of ET 200M has two connections, i.e. one to the CPU in station A, and one to the CPU in station B.

Station A and station B are interconnected by means of a CP 342-5 via PROFIBUS network.

Overview: Hardware configuration for the example with S7-300



Hardware used

For details about the modules used in the example, refer to the hardware configuration of the project template.

7.4 Configuring the hardware

If you want to reproduce or modify the hardware configuration of the project template, proceed as follows:

1. Create a project with two stations, e.g. station A and station B, and then open station A.
2. Select a rack from the hardware catalog.
3. Open the rack for station A and insert the power supply module, the CPU 315-2DP and the necessary central I/Os.
4. Open the second station and repeat steps 2 and 3.
5. Drag-and-drop the IM 153-2 to the DP master system ("railway track").
6. Insert the I/O devices of ET 200M.
7. Repeat steps 5 and 6 if you want to connect several ET 200M DP slaves to the DP master system.
8. Copy the entire DP segment to the second DP master system.

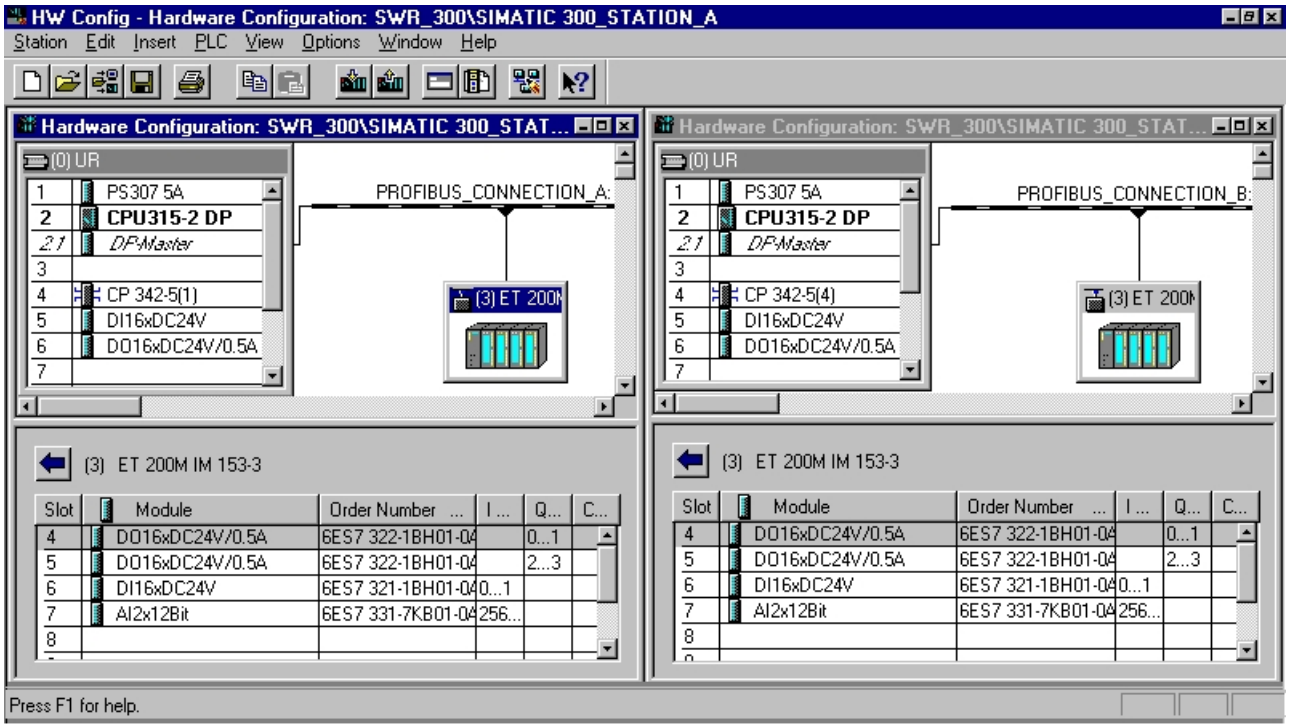
Hardware configuration rules

The configuration of distributed I/O devices must be consistent on both stations. To prevent any loss of consistency, always copy all slaves of the entire DP master system from the first station to the DP master of the second station even after making only minor changes. Copy the data by selecting **Edit > Insert Redundant Copy.**

Execute the **Edit > Insert Redundant Copy** menu command to ensure consistency of the I/O addresses of the DP slaves at both stations.

Example of the hardware configuration at station A and station B

The figure below shows an example of a consistent hardware configuration in both DP master systems.



7.5 Configuring the networks

If you want to reproduce or modify the network configuration of the project template, you should observe the following instructions.

What networks are there in a system with software redundancy?

On systems with software redundancy, a distinction is made between:

- The network that interconnects the two stations, also referred to as network for redundant link. Data is exchanged between the two stations via this network.
- The PROFIBUS DP networks to which the DP master systems and the ET 200M distributed I/O devices are connected. The stations use those networks to process the distributed I/O devices.

Network for data exchange between the two stations

The data can be transferred between the master and the standby device via MPI, PROFIBUS, or Industrial Ethernet.

In our example, the data is exchanged on the PROFIBUS network using communications modules.

1. Create a PROFIBUS network.
2. Network the CP of station A on PROFIBUS and select a node address, e.g. PROFIBUS address 3.
3. Network the CP of station B on PROFIBUS and select a node address, e.g. PROFIBUS address 4.

PROFIBUS DP networks for distributed I/O devices

ET 200M distributed I/O devices feature two DP interfaces, of which one is connected to the DP master system of station A and the other to the DP master system of station B.

Proceed as follows:

1. Create two PROFIBUS DP networks (for both DP master systems).
2. Select the DP connection of the CPU on station A and connect it to the first PROFIBUS DP network.
3. Select the DP connection of the CPU on station B and connect it to the second PROFIBUS DP network.
4. Select IM 153-2 from the hardware catalog. The IM 153-2 is available at PROFIBUS DP, folder ET 200M.

7.6 Configuring the connections

If you want to reproduce or modify the configuration of the connections in the project template, you should observe the following instructions.

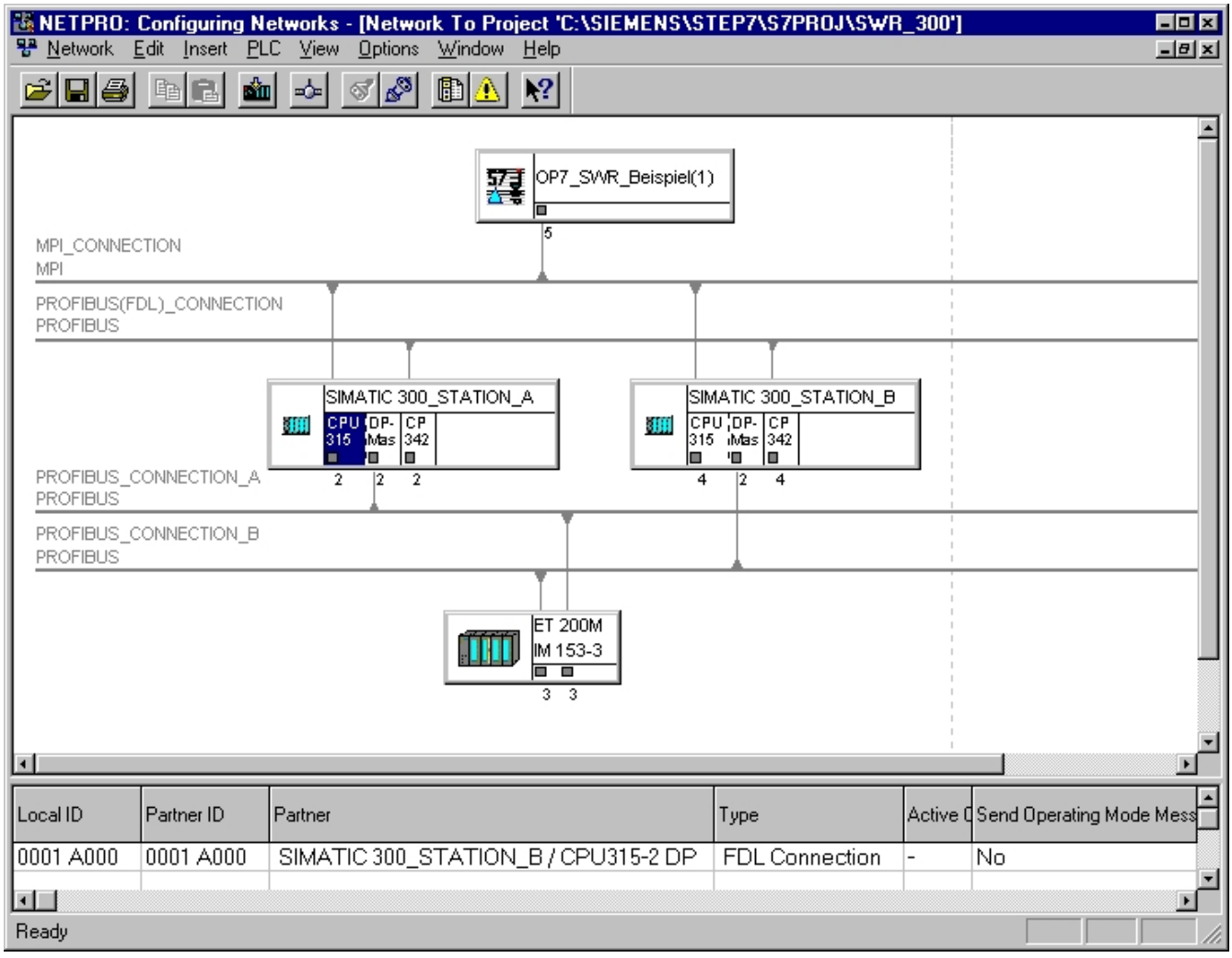
In the project template, a PROFIBUS network with FDL connections was selected for the data exchange between the two stations.

Create the required logical connection as follows:

1. Change from SIMATIC Manager to the network view
2. Select **View > DP Slaves** so that the DP slaves are also displayed in the network view
3. Double-click in the the connections table in the network view

   Result: A dialog box for defining the connection opens.
4. Select the two stations and specify an FDL connection

Network view with DP slaves and connections table



7.7 Creating the user program

If you want to reproduce or modify the user program for the project template, you should follow the instructions below.

The user program for the sample project for S7-300 consists of the following:

- A redundant program that is consistent on both stations and executed in the time-controlled program OB 35
- A non-redundant standard user programs that is different on each of the two stations and executed with the cyclic program of OB 1.

Structure of the user program

The following overview shows the points at which you have to call the blocks of software redundancy.

Startup program OB 100

```
CALL FC 100
   AS_ID          :='A'
   DB_WORK_NO     :=DB1
   DB_SEND_NO     :=DB2
   DB_RCV_NO      :=DB3
   MPT_ADR        :=4
          etc.
```

Call the FC 100 function in the startup OB. In FC 100, you inform the system of the addresses used for communication and of the data areas to be exchanged / updated between both station.

Data areas are: Process image of inputs, bit memory areas, data blocks and instance DBs for IEC timers / IEC counters.

Time-controlled program OB 35

```
CALL FB 101, DB5
   DB_WORK_NO     :=DB1
   CALL_POSITION  :=TRUE
   RETURN_VAL     :=MW6
   EXT_INFO       :=MW8

U     DB5.DBX     9.1
SPB   M001

   Declaration for redundant user program

(program component is available in station A and
                station B)

M0001: CALL FB 101, DB5
   DB_WORK_NO     :=DB1
   CALL_POSITION  :=FALSE
   RETURN_VAL     :=MW10
   EXT_INFO       :=MW12
```

At the start of OB 35, call FB 101 using parameter CALL_POSITION = TRUE. You can process status and control information in the specified instance DB

Evaluate the status information and program the CPU so that it skips the redundant user program when operating in standby mode.

Program your redundant user program in this area.

At the end of OB 35, call FB 101 using parameter CALL_POSITION = FALSE. This call reports completion of the execution of the redundant user program to the system.

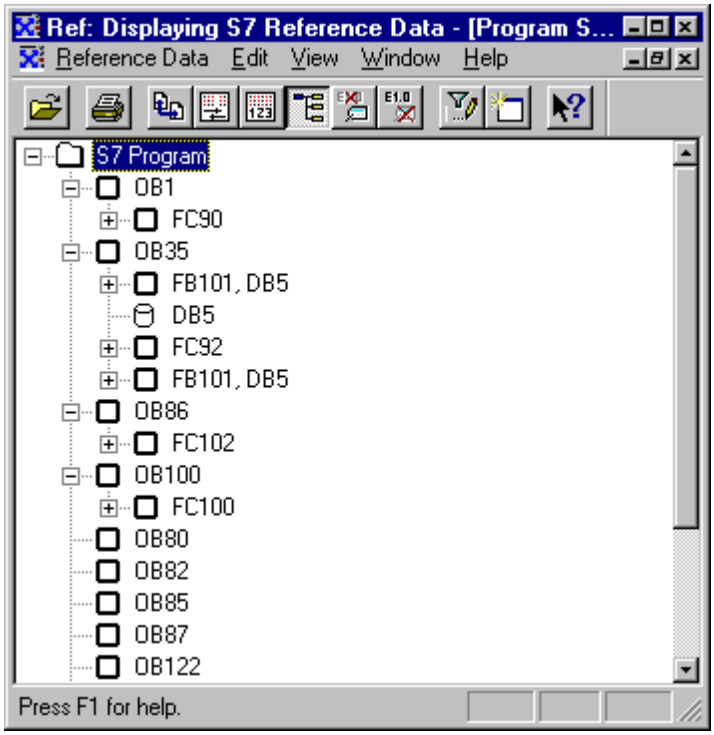
Diagnostics program OB 86

```
CALL FB 101, DB5
   DB_WORK        :=1
   OB86_EV_CLASS  :=OB86_EV_CLASS
   OB86_FLT_ID    :=OB86_FLT_ID
   RETURN_VAL     :=MW14
          :
          :
```

At the start of OB 86, call FC 102 with the corresponding start information. This call is necessary so that the system can automatically respond to the failure of a DP slave (automatic master to standby changeover).

Block structure

The figure below shows you the structure of the user program for the example with S7-300 and the nesting depth of the blocks.



User program rules

The user program organization should allow for strict separation of the redundant and non-redundant program components.

You may only use IEC counters and IEC timers in the redundant program component. It is not allowed to use S7 counters and/or S7 timers, as those addresses cannot be exchanged between the two stations.

See also

FC 100 'SWR_START' (Page 33)

FB 101 'SWR_ZYK' (Page 37)

FC 102 'SWR_DIAG' (Page 39)

7.8 Connecting HMI devices

SIMATIC S7 provides a new generation of operator panels that are particularly easy to use for visualization of process values and alarms.

The OP 7 and OP 17 operator panels are particularly suited for operation in redundant systems. Both operator panels support manual changeover between several stations at the touch of a button. This functionality lets you change from station A and station B and vice versa at any time for operating and monitoring the process.

An OP 7 operator panel is selected in the sample project for the S7-300 example. In the project template, the display of the status and control words and a number of alarms texts with reference to the user program are already configured for OP 7. You can edit the alarm texts to suit your requirements. You need the ProTool software to configure alarm texts.

See also:

Description of the OP 7 and OP 17 operator panels and of the ProTool engineering tool

Example: Software redundancy with S7-400 8

8.1 Example: Software redundancy with S7-400

We have created a project template to get you started with minimum effort. The project template can be executed and be modified to suit your requirements.

Using the simplified model of a road tunnel supervision system, we show you how simple it is to create the necessary configuration and programs. The example is based on two stations with a 414-2DP CPU.

Details on particular tasks and settings specific to software redundancy are provided on the following pages. General information that you already know from configuring and programming an S7-300 or S7-400, such as creating a project or configuring the CPU, is included only where it is necessary for comprehension of the examples.

8.2 Definition of the task and the technology scheme

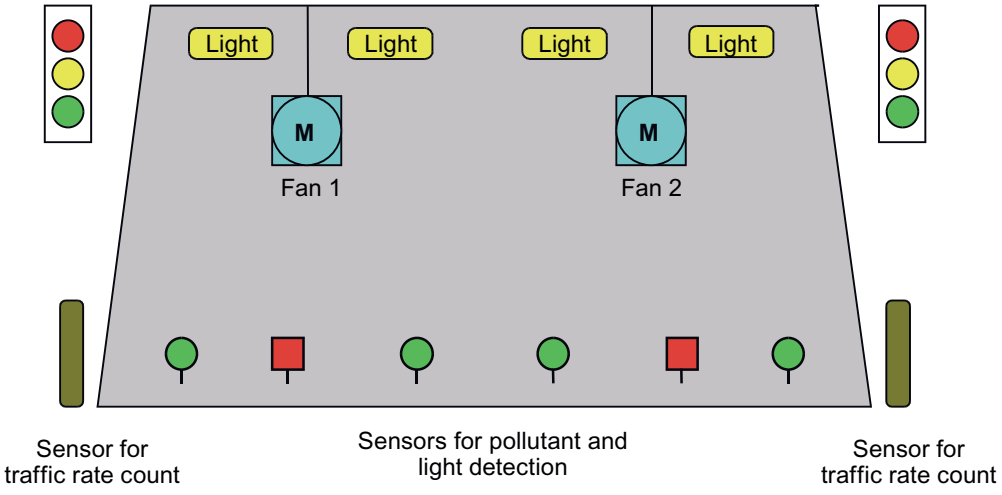
Description of the task

Two fans are used to ventilate a tunnel. Each fan has two speeds (stages) which are activated depending on the measured pollutant concentration in the air. The pollutant concentration is measured by means of two analog sensors. The fans are central components of the plant with higher demands on availability. The user program for controlling the fans is therefore loaded on both stations.

The tunnel is to be closed if the maximum permissible pollutant concentration prevails for more than two minutes. The tunnel entrance is controlled by a set of two traffic lights. For safety reasons, this element also requires a higher level of availability.

Details on particular tasks and settings specific to software redundancy are provided on the following pages. General information that you already know from configuring and programming an S7-300 or S7-400, such as creating a project or configuring the CPU, is included only where it is necessary for comprehension of the examples.

Technology scheme "Tunnel supervision"

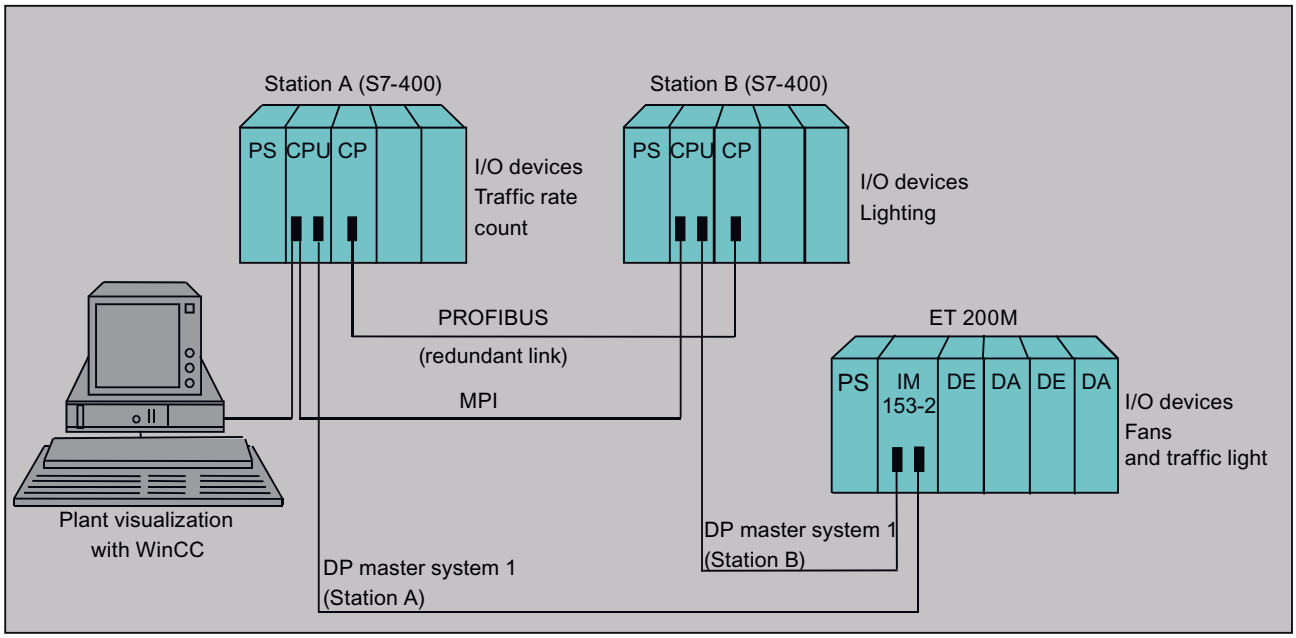


8.3 Hardware configuration for the example with S7-400

The diagram below shows the hardware configuration required. It consists of two S7-400 stations, each one with a CPU 414-2DP and an ET 200M DP slave. The IM 153-2 DP interface of ET 200M has two connections, i.e. one to the CPU in station A, and one to the CPU in station B.

Station A and station B are interconnected by means of a CP 443-5 via PROFIBUS network.

Overview: Hardware configuration for the example with S7-400



System visualization in WinCC

An operator station is used in the project template for operation, monitoring and visualization of the plant.

A faceplate was created and configured for comfortable operation and monitoring of the plant. The project template contains a corresponding configuration.

Hardware used

For details about the modules used in the example, refer to the hardware configuration of the project template.

8.4 Configuring the hardware

If you want to reproduce or modify the hardware configuration of the project template, proceed as follows:

1. Create a project with two stations, e.g. station A and station B, and then open station A
2. Select a rack from the hardware catalog
3. Open the rack for station A and insert the power supply module, the CPU 414-2DP and the necessary central I/Os
4. Open the second station and repeat steps 2 and 3
5. Drag-and-drop the IM 153-2 to the DP master system ("railway track")
6. Insert the I/O devices of ET 200M
7. Repeat steps 5 and 6 if you want to connect several ET 200M DP slaves to the DP master system
8. Copy the entire DP master system to the DP master of the second station

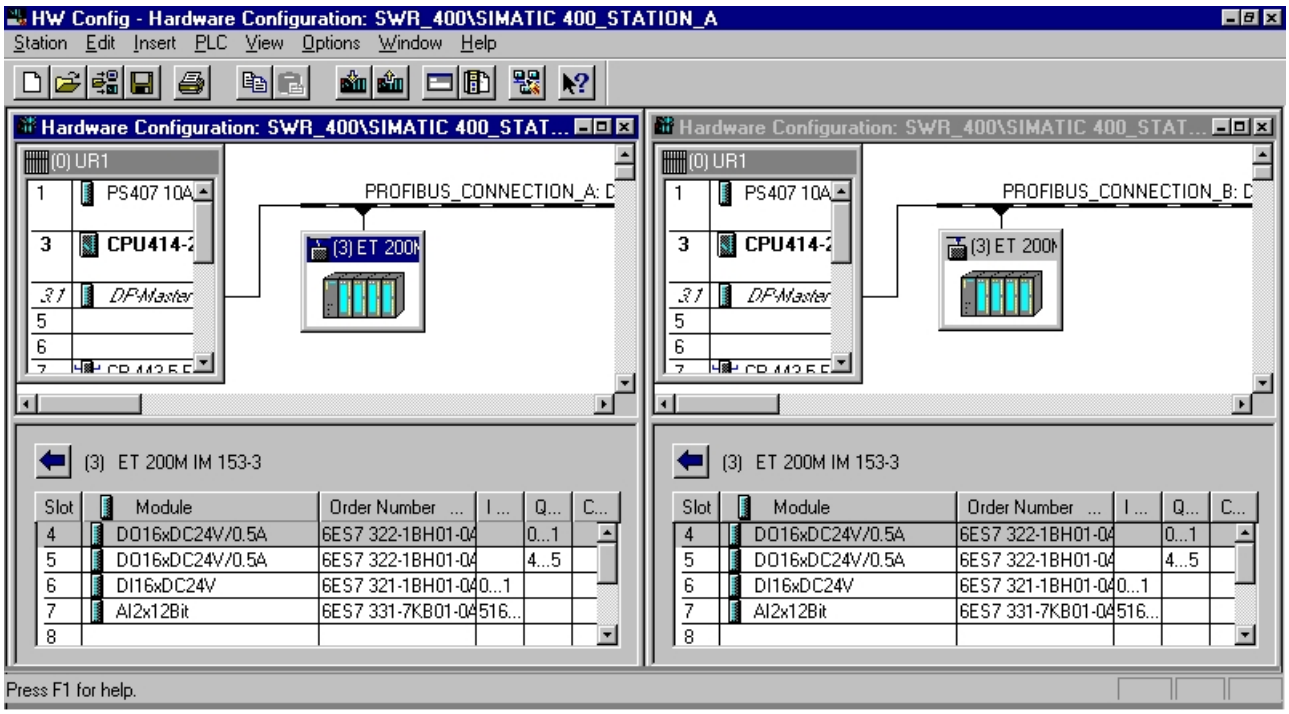
Hardware configuration rules

The configuration of distributed I/O devices must be consistent on both stations. To prevent any loss of consistency, always copy all slaves of the entire DP master system from the first station to the DP master of the second station even after making only minor changes. Copy the data by selecting **Edit > Insert Redundant Copy.**

Execute the **Edit > Insert Redundant Copy** menu command to ensure consistency of the I/O addresses of the DP slaves at both stations.

Example of the hardware configuration at station A and station B

The figure below shows an example of the consistent hardware configuration in both DP master systems.



8.5 Configuring the networks

If you want to reproduce or modify the network configuration of the project template, you should observe the following instructions.

What networks are there in a system with software redundancy?

On systems with software redundancy, a distinction is made between the following networks:

- The network that interconnects the two stations, also referred to as network for redundant link. Data is exchanged between the two stations via this network.
- The PROFIBUS DP networks to which the DP master systems and the ET 200M distributed I/O devices are connected. The stations use those networks to process the distributed I/O devices.

Network for data exchange between the two stations

The data can be transferred between the master and the standby device via MPI, PROFIBUS, or Industrial Ethernet.

In our example, the data is exchanged via PROFIBUS network using communications modules.

1. Create a PROFIBUS network.
2. Network the CP of station A on PROFIBUS and select a node address, e.g. PROFIBUS address 3.
3. Network the CP of station B on PROFIBUS and select a node address, e.g. PROFIBUS address 4.

PROFIBUS DP networks for distributed I/O devices

ET 200M distributed I/O devices feature two DP interfaces, of which one is connected to the DP master system of station A and the other to the DP master system of station B.

Proceed as follows:

1. Create two PROFIBUS DP networks (for both DP master systems).
2. Select the DP connection of the CPU on station A and connect it to the first PROFIBUS DP network.
3. Select the DP connection of the CPU on station B and connect it to the second PROFIBUS network.
4. Select IM 153-2 from the hardware catalog. The IM 153-2 is available at PROFIBUS DP, folder ET 200M.

8.6 Configuring the connections

If you want to reproduce the configuration of the connections provided in the project template, or create a user-specific configuration for the connections, you should observe the following instructions.

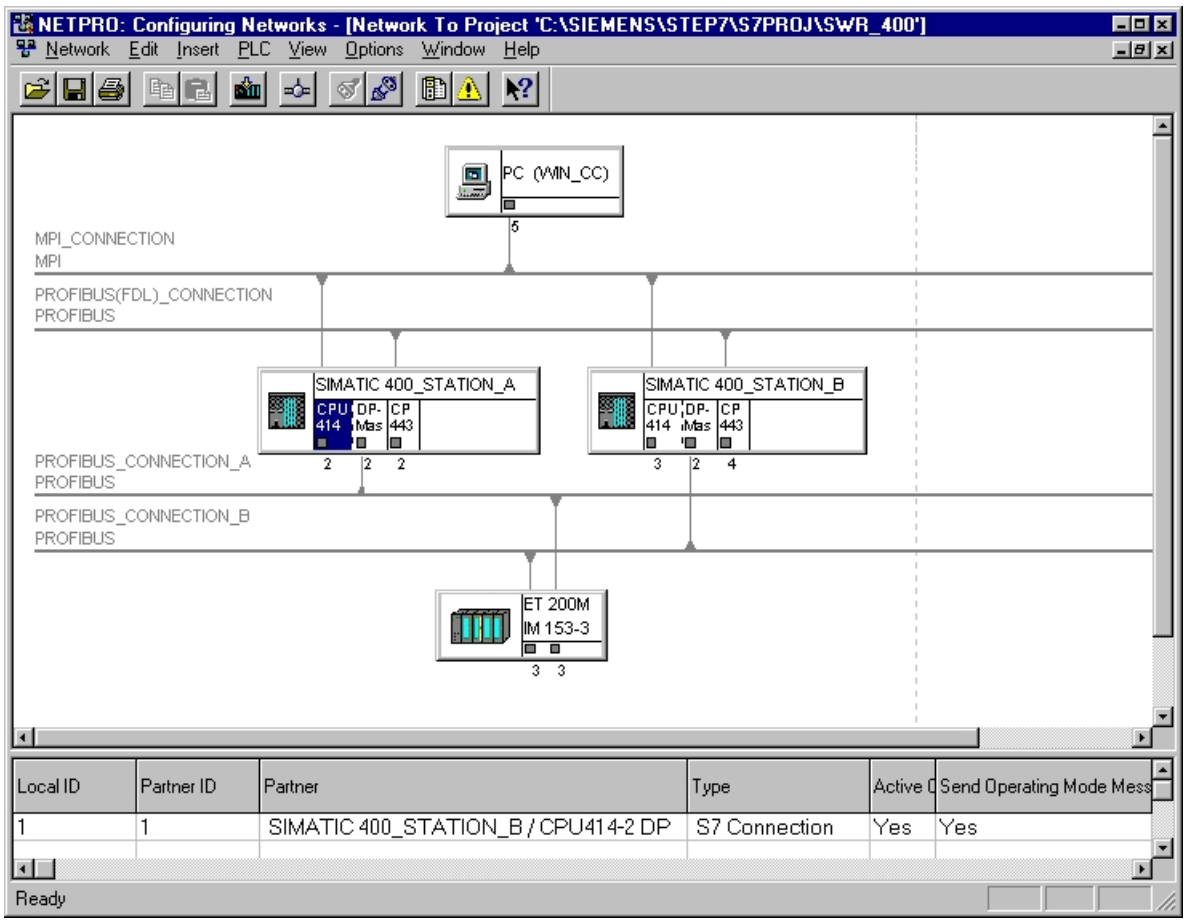
In the project template, a PROFIBUS network with FDL connections was selected for the data exchange between the two stations.

Create the required logical connection as follows:

1. Change from SIMATIC Manager to the network view
2. Select the **View > DP Slaves** menu command so that the DP slaves are also displayed in the network view
3. Double-click in the the connections table in the network view

   Result: A dialog box for defining the connection opens.
4. Select the two stations and specify an FDL connection

Network view with DP slaves and connections table



8.7 Creating the user program

If you want to reproduce or modify the user program for the project template, you should follow the instructions below.

The user program of the project template contains a fully redundant configuration. It is consistent at both stations and is executed within the cyclic program of OB 1.

Structure of the user program

The following overview shows the points at which you have to call the blocks of software redundancy.

OB 100 Startup Program

```
CALL FC 100
 AG_KENNUNG     :='A'
 DB_WORK_NO     :=DB1
 DB_SEND_NO     :=DB2
 DB_RCV_NO      :=DB3
 MPI_ADR        :=4
        etc.
```

The startup OB should invoke FC 100.
FC 100 should then inform the system which addresses are to be used for communication and which data areas are to be exchanged/updated between the two stations. Data areas are the process image of the inputs, bit memory address areas, data blocks and the instance data blocks for IEC timers/counters.

OB 1 Cyclic Program

```
CALL FB 101, DB5
 DB_WORK_NO     :=DB1
 CALL_POSITION  :=TRUE
 RETURN_VAL     :=MW115
 EXT_INFO       :=MW117

UN    DB5.DBX     8.1
        SPB   M001

Instructions for redundant-backup
      application program
(program section exists on station A
        and station B)

M001: CALL FB 101, DB5
 DB_WORK_NO     :=DB1
 CALL_POSITION  :=FALSE
 RETURN_VAL     :=MW119
 EXT_INFO       :=MW121
```

FC 101 should be invoked at the beginning of OB 1 citing the parameter CALL-POSITION = TRUE.
The specified instance DB can be used to process status and control information.

Analyse the status information and program the CPU to skip the redundant-backup application program when it is acting as the reserve unit.

This is where you insert your redundant-backup application program.

FC 101 should be invoked at the end of OB 1 citing the parameter CALL_POSITION = FALSE.
In this way the system is informed that processing of the redundant-backup application program has been completed.

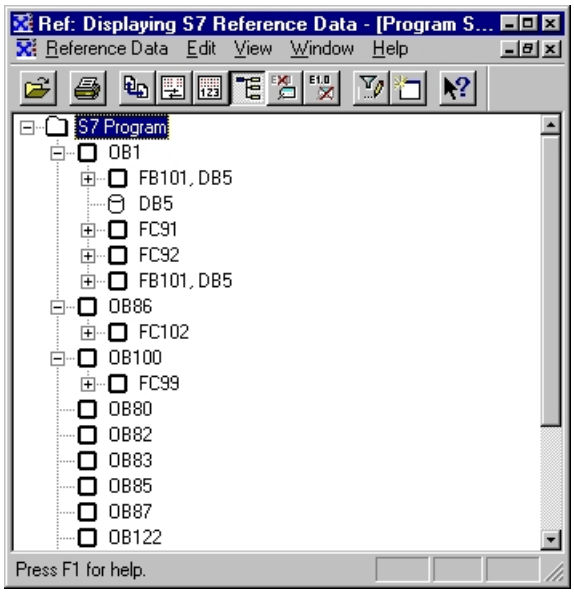
OB 86 Diagnostic Program

```
CALL FC 102
 DB_WORK        :=W#16#1
 OB86_EV_CLASS  :=#OB86_EV_CLASS
 OB86_FLT_ID    :=#OB86_FLT_ID
 RETURN_VAL     :=MW130
        .
        .
        .
```

OB 86 should invoke FC 102 citing the relevant startup information.
This function call is required in order that the system can respond automatically to the failure of a DP slave (automatic changeover from master to reserve).

Block structure

The figure below shows you the structure of the user program for the example with S7-400 and the nesting depth of the blocks.



User program rules

You may only use IEC counters and IEC timers in the redundant program component. It is not allowed to use S7 counters and/or S7 timers, as those addresses cannot be exchanged between the two stations.

See also

FC 100 'SWR_START' (Page 33)

FC 102 'SWR_DIAG' (Page 39)

FB 101 'SWR_ZYK' (Page 37)

8.8 Connecting HMI devices

Description

SIMATIC S7 offers a new generation of operator panels which are particularly easy to use for visualization of process values and alarms.

An operator station is used for plant visualization in the project template with S7-400. A faceplate was created and configured for comfortable operation and monitoring of the plant.

The faceplate facilitates execution of the following functions on an operator station (OS):

- Initiation of a master to standby changeover
- Disabling or enabling redundancy between the master and standby device, including the display of the redundancy state.
- Display of the status of the CPU connection (redundant link)
- Display of the standby status of the DP slaves

See also

Faceplate for operating and monitoring tasks (Page 97)

Software redundancy and operator stations with WinCC

9

9.1 Faceplate for operating and monitoring tasks

Description

The software package contains a preconfigured faceplate to facilitate your operating and monitoring tasks. The SETUP program of software redundancy automatically installs this faceplate, provided WinCC is installed on your system.

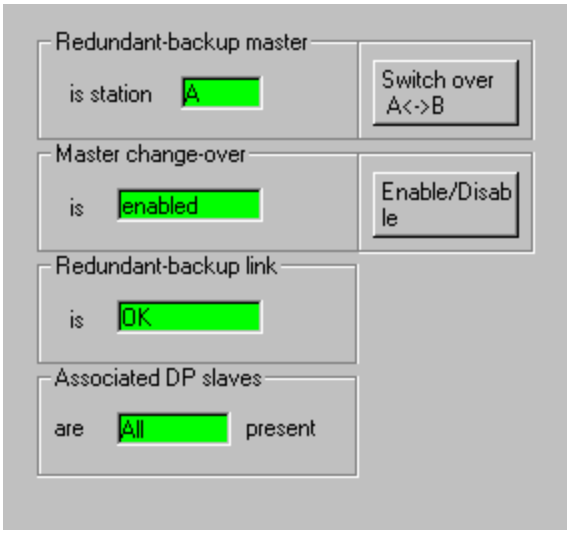
The sections below show you how to configure the faceplate in WinCC. In addition to this configuration, you also have to set up a redundant link on your operator station so that the update of the faceplate is also retained if the master station fails, or if a master to standby changeover is made. How to set up the link and what special considerations need to be observed is summarized in a separate description. The description is available in the ‘SWR_WinCC_English.doc’ or ‘SWR_WinCC_English.pdf’ files on your CD.

Purpose of the faceplate

The faceplate facilitates execution of the following functions on an operator station (OS):

- Initiation of a master to standby changeover
- Disabling redundancy between the master and standby device (inhibit master to standby changeover) or enabling redundancy (enable master to standby changeover)
- Display of the status of the CPU connection (redundant link)
- Display of the standby status of the DP slaves

View of the faceplate



See also

Configuring the faceplate in WINCC (Page 99)

9.2 Configuring the faceplate in WINCC

You install the faceplate in a screen using WinCC. Configure the faceplate by means of corresponding properties dialogs.

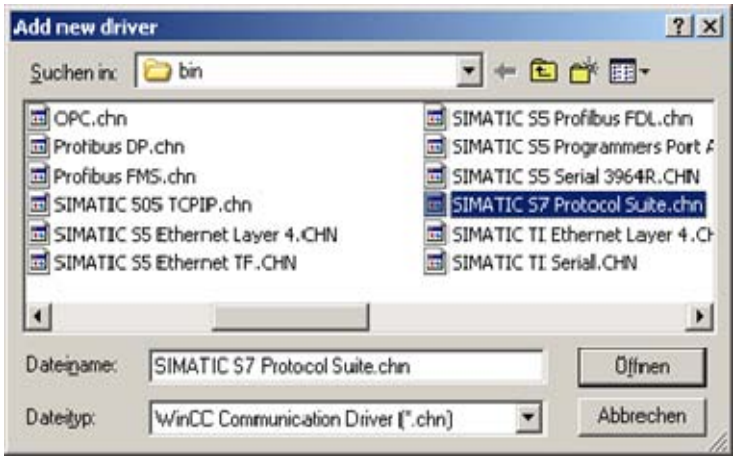
Configuration procedures:

1. Configure the connection for WinCC (Page 100)
2. Defining the faceplate tags (Page 102)
3. Insert the faceplate in a screen (Page 104)
4. Interconnect the display fields with the tags (dynamizing the screen) (Page 107)

9.3 Configuring the connection for WinCC

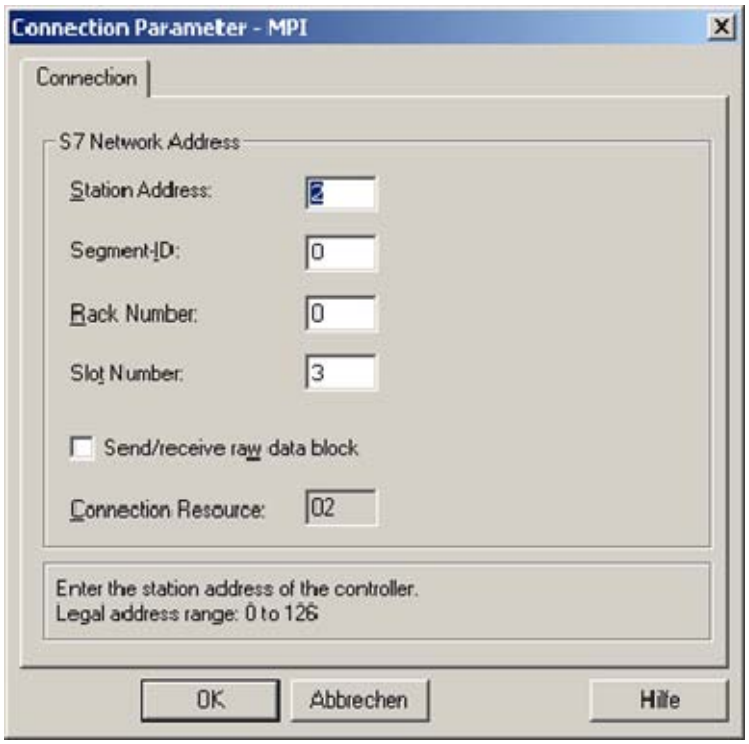
You must configure a connection to the redundant system so that you can interconnect your WinCC station with the automation system. This only requires one connection from the operator station to station A, because the connection to station B is set up by means of the WinCC changeover function.

1. Adding new drivers: Open the "Tag Management" directory and right-click on "Add New Driver". Select the driver in the "C:\Programs\SIEMENS\WINCC\bin" directory



2. Open the "SIMATIC S7 PROTOCOL SUITE" directory in the Control Center. The directory is located in the "Tag Management" container.
3. Select the folder in which you want to create the connection, e.g. MPI.
4. Right-click in the folder and insert a new connection.
5. Select the inserted connection and assign it a name, for example 'SW_Redundancy'.

6. Right-click and select "Properties" from the shortcut menu.
7. Enter the node address of the station for which the connection is to be created (recommendation: Enter the node address of station A).



9.4 Defining the faceplate tags

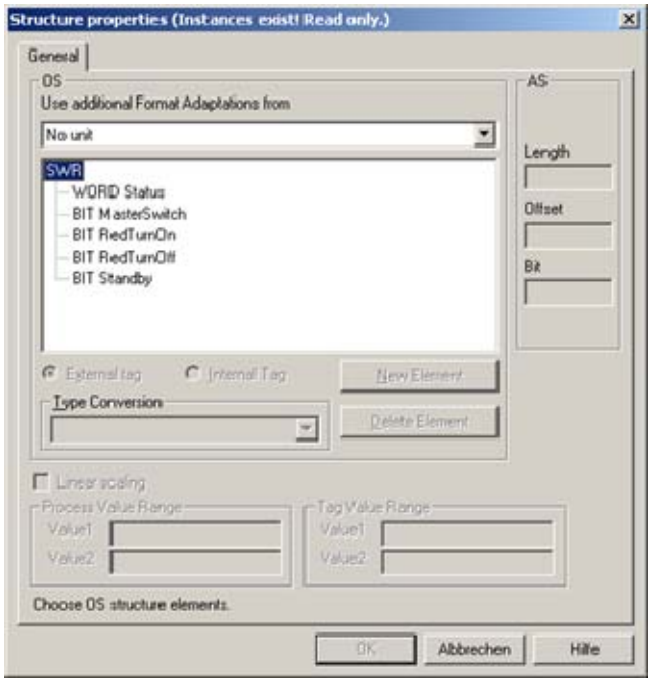
It is advisable to define the faceplate tags after having created a connection between the operator station and a station.

Proceed as follows:

1. Open the 'Structure types' folder in the Control Center.
2. Right-click and insert a new structure type.

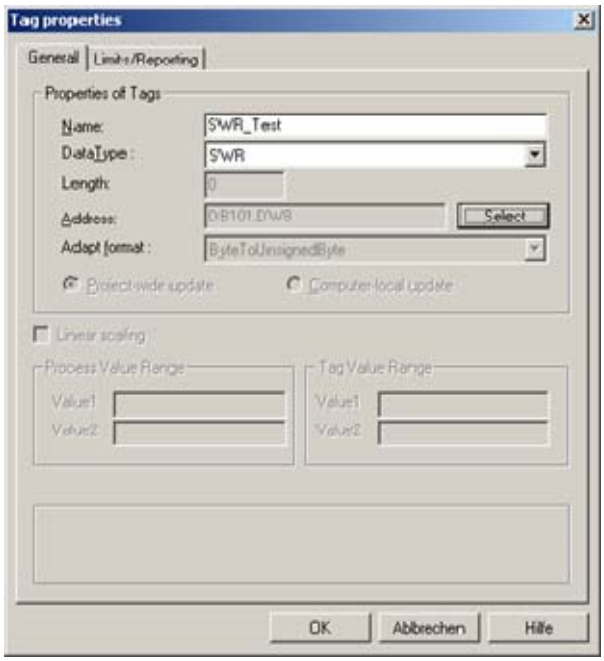
   Result: The "Structure properties" window opens.
3. Enter a name for the structure tag, for example "SWR".
4. Click the "New element" button and insert the faceplate tags (4 tags).
5. Assign each tag the corresponding name and data type.

Name	Data type	Offset	Bit
OFFICE WORD	OFFICE WORD	0	0
BIT MasterSwitch	BIT	2	0
BIT RedTurnOn	BIT	2	9
BIT RedTurnOff	BIT	2	8



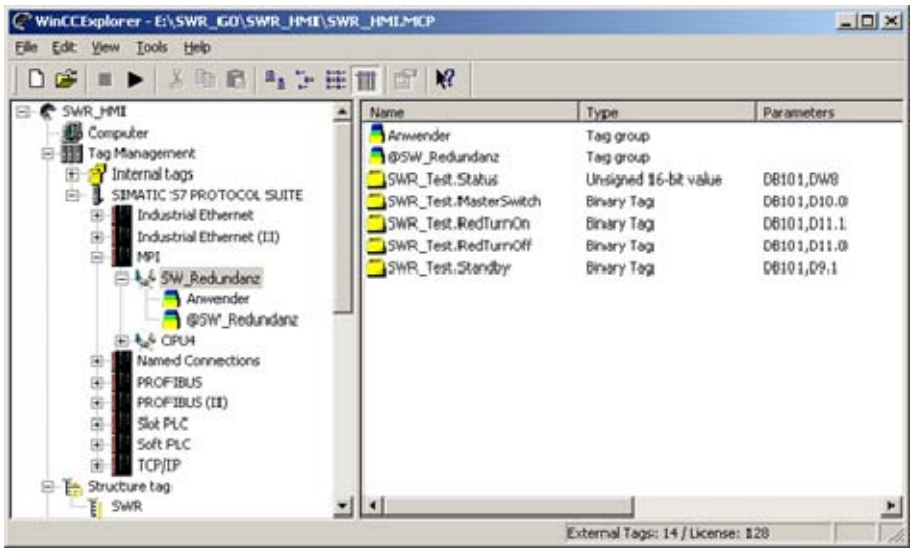
6. Select the previously inserted connection ("SW_Redundancy") from the "SIMATIC S7 PROTOCOL SUITE" folder.
7. Right-click in the field and insert a new tag.

8. Assign a tag name such as "SWR_Test" and then select the "SWR" data type.



9. Define the number of the instance DB and the offset for the structure tag in the ‘Addresses’ input box (offset is DW 8).

   Result: The faceplate now knows the status word and control bits it has to access.

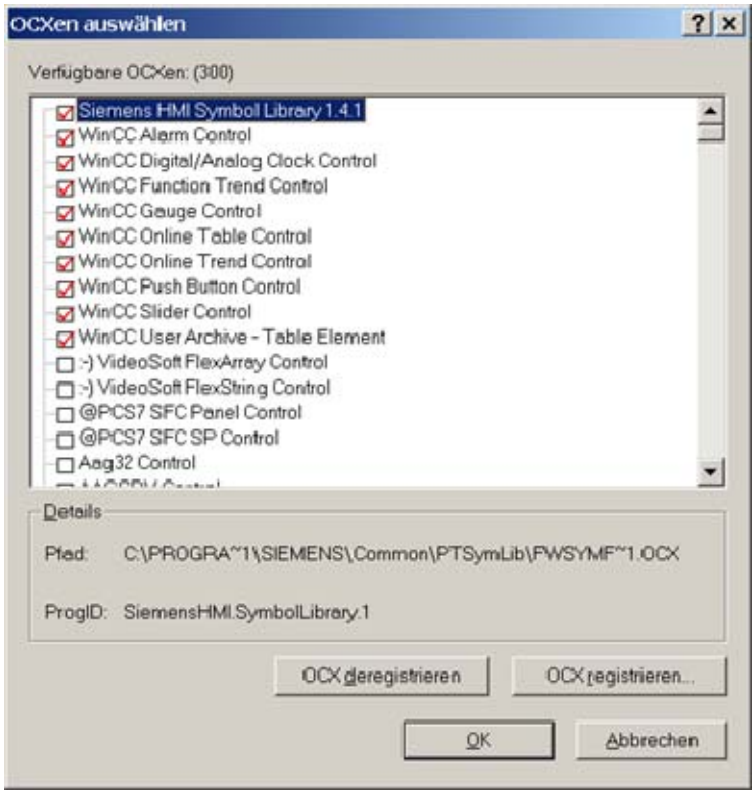


9.5 Inserting the faceplate in a screen

Technically, the faceplate is implemented as an Active X control. Insert the faceplate in a screen as follows:

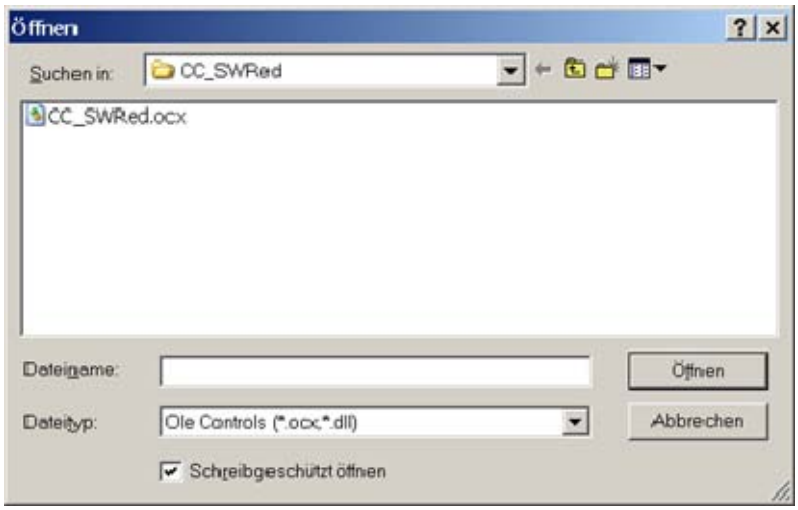
1. Open a picture in the Control Center using the Graphic Designer.
2. Select the control using the menu command **Object Palette > Controls**.
3. Right-click and select "Add/Remove".

   Result: A window appears for registering the faceplate after you released the mouse button.

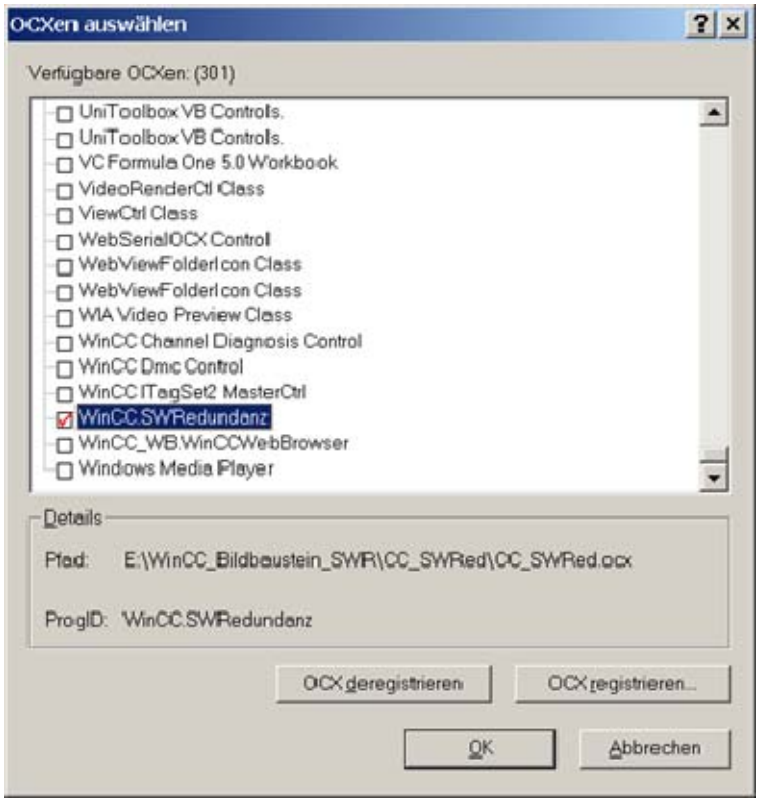


4. Select "Register OCX"

5. Open the "CC_SWRed.ocx" control.



6. Select the "WinCC Software Redundancy" check box in the "Select OCX Controls" window.



Result: The faceplate is visible in the screen and known in WinCC.


Objects
Object Palette
Selection
Siemens HMI Symbol Library 1.4.1
WinCC Alarm Control
WinCC Digital/Analog Clock Control
WinCC Function Trend Control
WinCC Gauge Control
WinCC Online Table Control
WinCC Online Trend Control
WinCC Push Button Control
WinCC Slider Control
WinCC User Archive - Table Element
WinCC.SWRedundanz
Standard
Controls

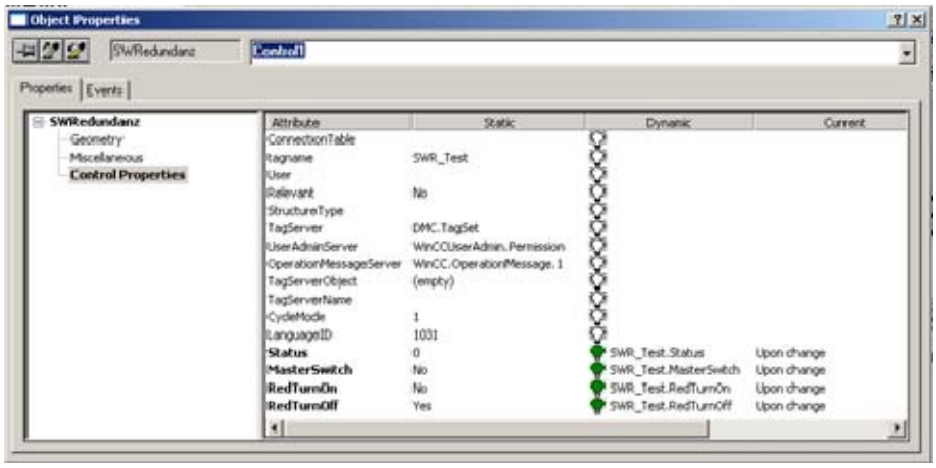
9.6 Interconnecting the display fields with the tags (dynamizing the screen)

Interconnect the display fields with the tags after having inserted the faceplate in the screen.

Proceed as follows (the tag names are examples):

1. Select the faceplate.
2. Right-click the faceplate and select **Properties** from the shortcut menu.

   Result: The "Object Properties" dialog box opens.



3. Select "Control Properties" in the left pane of the window.
4. Go to the right pane of the window and enter the name "SWR_Test" for the "tagname" attribute.
5. Click the display icon (light bulb) in the "Dynamics" row and select "SWR_Test.Status" from the displayed list box.
6. Click the display icon (light bulb) in the "MasterSwitch" row and select "SWR_Test.MasterSwitch" from the displayed list box.
7. Click the display icon (light bulb) in the "RedTurnOn" row and select "SWR_Test.RedTurnOn" from the list box.
8. Click the display icon (light bulb) in the "RedTurnOff" row and select "SWR_Test.RedTurnOff" from the list box.
9. Save the changes in the Graphic Designer.

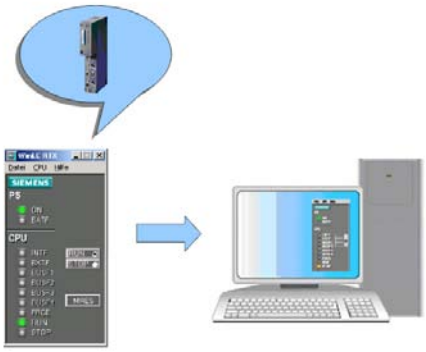
   Result: You can now use the faceplate and start it with "WinCC Runtime".

Software redundancy with WinAC RTX 10

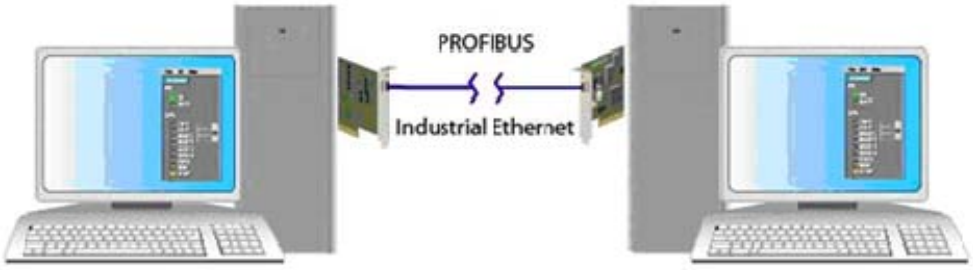
10.1 PC-Based Control

The WinAC (Windows Automation Center) PC-based controllers are executed on a standard PC and provide the same functionality as SIMATIC S7 CPUs (hardware controllers). WinAC RTX is a programmable software PLC - a software application that runs on a standard computer (PC).

You can now use software redundancy on WinAC RTX as of version 2008.



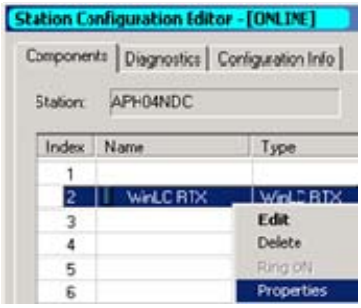
10.2 Advanced PC configuration for software redundancy



10.2.1 Setting up the PC station

Follow the steps below to set up the PC station:

1. Open the PC station folder in the project and double-click the symbol for the configuration to call up the STEP 7 HW Config.
2. Navigate to your specific controller under SIMATIC PC Station.
3. Drag the controller into the same index it occupies in the Station Configuration Editor on the target computer.

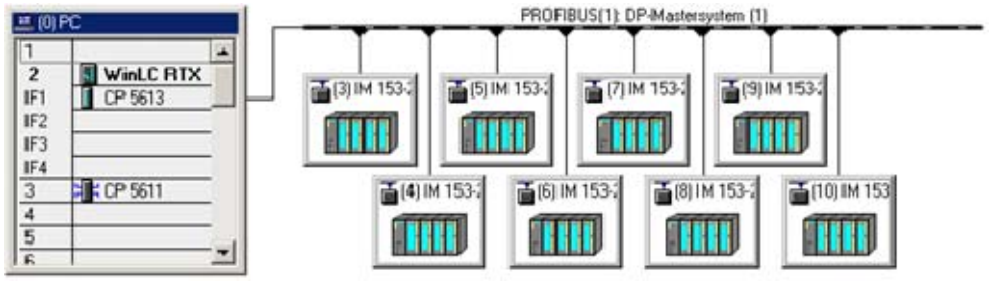


4. Verify that the name of the controller matches the name of the controller configured in the Station Configuration Editor.
5. Drag the CP defined as a submodule from the hardware catalog to the interface slots (IF slots) of the WinLC RTX controller.

10.2.2 Creating a configuration for a SIMATIC PC station in STEP 7

Follow the steps below to create a SIMATIC PC station in STEP 7:

1. Select the components from the hardware catalog.
2. Place the components as in the Station Configuration Editor.
3. Download the configuration to the controller.



Additional references

11

11.1 Data type INT

If the formal parameter of the SFCs/SFBs is the data type INT (16-bit integer) you can assign it the formal parameter the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
27	MW 100	#TYP_INT
-25	IW 0	
	QW 0	
	DBW 1	

11.2 Data type WORD

If the formal parameter of the SFCs/SFBs is of data type WORD you can assign it the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
W#16#1F12	MW 100	#TYP_WORD
2#0001111100010010	IW 0	
C#32	QW 0	
B#(5.25)	DBW 1	

11.3 Data type BYTE

If the formal parameter of the SFCs/SFBs is of data type BYTE you can assign it the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
B#16#1F	MB 100	#TYP_BYTE
	IB 0	
	QB 0	
	DBB 1	

11.4 Data type BOOL

If the formal parameter of the SFCs/SFBs is of data type is of data type you can assign it the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
TRUE	M 100.0	#OK_BIT_MEMORY
	I 0.0	
	Q 0.0	
	DBX 3.0	

11.5 Data type ANY

If the formal parameter of the SFCs/SFBs is of data type ANY you can assign it the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
P#M0.0 BYTE 20 (meaning: 20 bytes, starting at M 0.0) P#DB58.DBX16.0 BYTE 14 (meaning: 14 bytes in DB 58, starting at data bit 16.0)	I 0.0 MB 5 QW 2	#TYP_ANYTYP (arrays and structures can be transferred using parameters of data type ANY)

Note

Data type ANY allows input of any elementary data type entered as global parameter.

Observe the following syntax rules for direct input of values for data type ANY:

- The value always begins with the "P#" prefix, followed by the bit address of a STEP 7 operand (e.g. M0.0) and a length declaration (elementary data type, e.g. BOOL, BYTE, WORD or DWORD or composite data type, e.g. DATE_AND_TIME, or STRING)

The bit address of the STEP 7 operand must be 0 for all length declarations with the exception of BOOL.

11.6 Symbolic representation

The following requirements must be met to enable the use of symbols for actual parameters:

- For global data, the name (= symbol) must be defined in the global symbols table
- For local data, the name (= symbol) must be specified in the declaration table of the block. Symbols for local data must begin with "#" prefix.

11.7 Global data

Global data can be accessed from any code block (FC, FB, OB). That includes in particular bit memories (M), inputs (E), outputs (A), timers, counters and elements of data blocks (DBs). Global data can be accessed by means of absolute or symbolic addressing.

11.8 Memory areas

I = for the process image of inputs

Q = for the process image of outputs

M = for bit memories

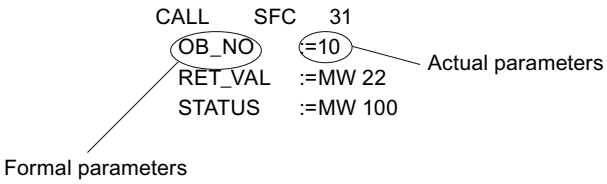
D = for data blocks

L = for local data

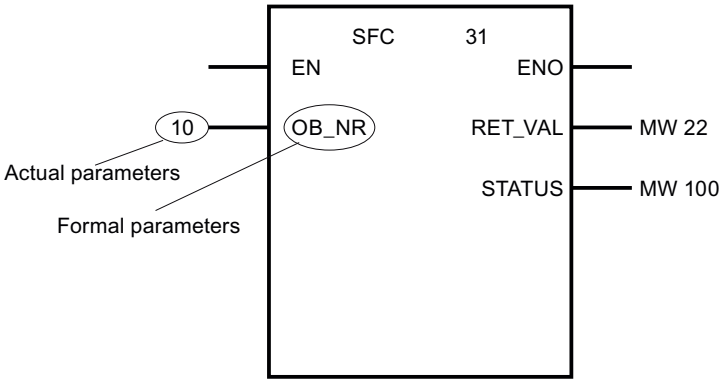
11.9 Formal parameters / actual parameters

A formal parameter is assigned a name and data type when the programmable block is created and is declared as INPUT or OUTPUT. The programming device automatically displays the list of formal parameters after the block was called (e.g. CALL SFC 31).

Example (STL)



Example (LAD)



11.10 Data type CHAR

If the formal parameter of the SFCs/SFBs is of data type CHAR (ASCII CHARacter set) you can assign it the following actual parameters:

Direct input (example)	Input of a global parameter	Symbolic input
"I"	MB 100	#TYP_CHAR
	IB 0	
	QB 0	
	DB 1	

Index