PROCESS AUTOMATION

Freelance 2019

Engineering Manual
Process Stations

PROCESS AUTOMATION

Freelance 2019

Engineering Manual
Process Stations

Document Number: 2PAA114393-111
Revision: A
Release: March 2019

Notice

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

Products described or referenced in this document are designed to be connected, and to communicate information and data via a secure network. It is the sole responsibility of the system/product owner to provide and continuously ensure a secure connection between the product and the system network and/or any other networks that may be connected.

The system/product owners must establish and maintain appropriate measures, including, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, and so on, to protect the system, its products and networks, against security breaches, unauthorized access, interference, intrusion, leakage, and/or theft of data or information.

ABB verifies the function of released products and updates. However system/product owners are ultimately responsible to ensure that any system update (including but not limited to code changes, configuration file changes, third-party software updates or patches, hardware change out, and so on) is compatible with the security measures implemented. The system/product owners must verify that the system and associated products function as expected in the environment they are deployed.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2014/30/EU and in Low Voltage Directive 2014/35/EU.

Trademarks

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Table of Contents

4 Commissioning

5 Controller redundancy

6 Processing and error handling

Appendix A Hardware object data

Index

About this book

Use of warning, caution, information, and tip icons

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

Electrical warning icon indicates the presence of a hazard which could result in *electrical shock.*

Warning icon indicates the presence of a hazard which could result in *personal injury.*

Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard which could result in *corruption of software or damage to equipment/property.*

Information icon alerts the reader to pertinent facts and conditions.

Tip icon indicates advice on, for example, how to design your project or how to use a certain function

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

Terminology

The Glossary contains terms and abbreviations that are unique to ABB or have a usage or definition that is different from standard industry usage. Please make yourself familiar to that.

You will find the glossary at the end of the ***Engineering Manual System Configuration.***

Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
- Press CTRL+C indicates that you must hold down the CTRL key while pressing the C key (to copy a selected object in this case).
- Press **ESC, E, C** indicates that you press and release each key in sequence (to copy a selected object in this case).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
  - The following convention is used for menu operations: MenuName > MenuItem > CascadedMenuItem. For example: select **File** > **New** > **Type**.
  - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.

- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

  ```
  Entered value is not valid. The value must be 0 to 30.
  ```

  You may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

  TIC132

  Variables are shown using lowercase letters.

  sequence name

1 Process stations

1.1 Overview

A scalable Freelance system essentially consists of the process and operator stations. You can perform the necessary configuration and commissioning tasks from your PC using the Freelance Engineering software. Diagnoses are performed in graphic interfaces.
Among the functions supported by Freelance Engineering are the following:

- Read and enhance configuration data from new field devices.
- Configure devices using the parameter setting dialogs and pre-defined I/O structures.
- Display bus cycle times and diagnostic data of devices.
- Search the buses for nodes with unknown device addresses.
- Exclude individual devices from cyclic communication.
- Integrate new devices without the need to re-initialize the bus line.

To configure your automation task, first create the required resources and programs in the project tree using editors conforming to IEC 61131-3. In the next step, these configured applications are allocated to the actually required hardware in the hardware editor and then loaded online into the process stations.

In order to facilitate the visualization of your process, powerful software packages such as Freelance Operations are available for operation.

A Freelance system supports different types of process stations (controllers) which can be combined as required. Currently, the controller types AC 900F, AC 800F and AC 700F are available.

The **Ethernet** system bus links the individual stations with each other. This bus transmits the data between the process stations, the operator stations and the engineering station. Communication within the process station, i.e. data

transmission between the CPU module and the connected modules, is realized via the internal station bus.

Communication between the AC 900F process stations and operator/gateway stations as well as lateral communication between the AC 900F process stations can optionally be performed via a redundant Control Net.

Control Net redundancy is supported by the following Freelance process stations:

- AC 900F
- AC 900F redundant
- AC 900F Lite
- AC 900F Lite redundant
- AC 900F Plus
- AC 900F Plus redundant

The following sections provide an overview of the currently available process stations. See the corresponding ***Mounting and Installation Instructions*** for more detailed information. Older hardware and process stations of previous system releases are also supported, but are not described in this manual. For information in this matter, see the manuals of the corresponding system releases.

1.1.1 AC 900F process station

The AC 900F process station is available in two variants, namely **Standard, Lite** and **Plus** version.
The AC 900F Lite process station is designed for smaller lines:

- there are only three instead of four Ethernet interfaces available,
- the available memory capacity and the controller performance are lower than with the Standard version.
- it is recommended not to configure more than 400 I / O signals

The AC 900F Plus process station is designed for larger lines:

- four slots for fieldbus communication interfaces,
- the available memory capacity and the controller performance are higher than with the Standard version.

Details and technical data of the variants are included in the ***Mounting and Installation Instructions, AC 900F Controller***.

The descriptions in the present document generally apply to all variants; differences are explicitly specified, where applicable.

The AC 900F process station in the **Standard version** exhibits the following basic features:

- PM 902F CPU module with
  - four Ethernet interfaces,
  - one diagnostic interface,
  - two serial interfaces,
  - one display unit (optional),
  - max. two fieldbus communication interfaces (PROFIBUS and/or CAN for Rack-I/O),
  - up to ten directly connected I/O modules with I/O terminal units.

The AC 900F process station in the **Plus version** exhibits the following basic features:

- PM 904F CPU module with
  - four Ethernet interfaces,
  - one diagnostic interface,
  - two serial interfaces,
  - one display unit (optional),
  - max. four fieldbus communication interfaces (PROFIBUS and/or CAN for Rack-I/O),
  - up to ten directly connected I/O modules with I/O terminal units.

The AC 900F process station in the **Lite version** exhibits the following basic features:

- PM 901F CPU module with
  - three Ethernet interfaces,
  - one diagnostic interface,
  - two serial interfaces,
  - one display unit (optional),
  - max. two fieldbus communication interfaces (PROFIBUS and/or CAN for Rack-I/O),
  - up to ten directly connected I/O modules with I/O terminal units.

Depending on the type and quantity of process signals, up to ten plug-in type input/output modules can be directly connected. Additionally, two (PM 901F / PM 902F) or four (PM 904F) extensive PROFIBUS lines can be connected via two/four PROFIBUS modules. The CAN module (max. one per controller) allows connection of a maximum of 5 rack I/O units and thus the connection of 45 I/O modules as used in the rack-based Freelance process stations.

The following directly connected I/O modules are available for an AC 900F (non-redundant controller only):

AI 723F	Analog input module: 16 AI/DI
AI 731F	Analog input module: 8 AI/DI
AO 723F	Analog output module 16 AO
AX 721F	Analog input/output module: 4 AI/DI, 4 AO
AX 722F	Analog input/output module: 8 AI/DI, 8 AO
DA 701F	Digital/analog input/output module: 16 DI, 8 DI/DO, 4 AI/DI, 2 AO
DC 732F	Digital input/output module: 16 DI, 16 DI/DO
DI 724F	Digital input module: 32 DI
DX 722F	Digital input/output module: 8 DI, 8 DO (relay contacts)
DX 731F	Digital input/output module: 8 DI, 4DO (relay contacts)

The following fieldbus communication interfaces are available:

CI 910F	CAN module for Rack-I/O
CI 930F	PROFIBUS module, suitable for redundant configurations
CM 772F	PROFIBUS module (only usable with PM 902F)
CI 773F	PROFIBUS module

The AC 900F internal interfaces support additional serial Ethernet fieldbus protocols such as Modbus, Send and Receive Function Blocks and Telecontrol Link-Up. Fieldbus-compatible components such as remote I/O, field devices and network components can be connected to the AC 900F process station and configured using Freelance Engineering. ABB offers equipment for applications covering standard and hazardous areas.

An AC 900F controller without direct I/O modules can be run in redundant configuration. For this purpose, two process stations of identical design are connected by means of a redundancy link. The ETH2 Ethernet interface must be

used for the redundancy link; see the ***Mounting and Installation Instructions, AC 900F Controller, Commissioning*** for details.

In a redundant AC 900F controller, the CI 930F and CI 773F PROFIBUS modules respectively the CI 910F CAN module can be used. The CM 772F PROFIBUS module does not support the redundancy mode.

The CI 930F PROFIBUS module allows for both PROFIBUS master redundancy and line redundancy. The CI 910F CAN module allows CAN master redundancy only.

See Interface modules for AC 900F and AC 700F on page 65 for PROFIBUS module details and ***Engineering Manual I/O Modules for AC 700F / AC 900F*** for details on directly connected I/O modules.

Control Net redundancy

A redundant Control Net can be connected using both ETH1 and ETH3 Ethernet interfaces. For this purpose, the interfaces must be configured through the display unit (TD 951F) or diagnostic interface of the controller. See the ***Mounting and Installation Instructions, AC 900F Controller, Commissioning*** for more detailed information.

Control Net redundancy with redundant AC 900F controller configuration

Failure of Control Net line A of the primary controller will cause a redundancy toggle, provided the secondary controller can communicate via at least one of the two Control Net lines. Failure of Control Net line B of the primary controller will not result in a controller redundancy toggle.

Prim. Controller		Sec. Controller		Redundancy-toggle
Line A	Line B	Line A	Line B	
ok	ok	ok / Failure	ok / Failure	-
Failure	ok	ok	ok / Failure	Yes
Failure	ok	Failure	ok	-
ok	Failure	ok / Failure	ok / Failure	-
Failure	Failure	ok	ok	Yes
Failure	Failure	Failure	ok	Yes
Failure	Failure	ok	Failure	Yes
Failure	Failure	Failure	Failure	-

CNetRed_RedCtrl_us.png

1.1.2 AC 800F process station

The basic features of an AC 800F controller are:

- a basic unit with CPU board and module slots,
- the power supply module,
- at least one Ethernet module,
- a maximum of four fieldbus modules.

There are different variants of the fieldbus modules for:

- PROFIBUS DP
- Serial (e.g. Modbus)
- CAN
- Ethernet (e.g. FIELDBUS Foundation HSE)

The AC 800F controller supports in particular the connection of different fieldbusses. In its basic configuration it consists of a housing and a main board. The two components together form a unit that can be equipped with various modules. One power supply module and one Ethernet module for the connection of the system bus are mandatory. The controller can be equipped with a maximum of 4 fieldbus modules selected from CAN, PROFIBUS, Ethernet and serial modules.

The CAN module allows connection of a maximum of 5 rack I/O units and thus the connection of 45 I/O modules as used in the rack-based Freelance process stations.

Each PROFIBUS module allows the connection of one PROFIBUS line, i.e. the connection of a maximum of 125 slaves. Each of these slaves can be of modular design, i.e. contain a maximum of 64 modules.

The Ethernet fieldbus module allows connecting FOUNDATION Fieldbus HSE or TCP/IP transmit and receive.

The serial module has two interfaces which can be occupied as required either with the Modbus master interface protocol, the Modbus slave interface protocol or the telecontrol interface protocol.

A process station in redundant configuration consists of two identical controllers. The controller acting as the Primary executes the application programs and synchronizes data with the Secondary. The Secondary takes over and continues processing if any error occurs in the Primary controller.

The following modules are available for the AC 800F controller:

Basic units:

PM 802F	Basic unit with 4 MByte flash EPROM and 4 MByte RAM
PM 803F	Basic unit with 8 MByte flash EPROM and 16 MByte RAM

Power supply modules:

SA 801F	Input voltages from 115 to 230 V AC for PM 802F
SA 811F	Input voltages from 115 to 230 V AC for PM 803F
SD 802F	Redundant DC voltage 24 V DC for PM 803F
SD 812F	Redundant DC voltage 24 V DC for PM 803F

Ethernet modules:

EI 801F	10Base2 module for PM 802F
EI 802F	AUI module for PM 802F
EI 803F	10BaseT module for PM 802F
EI 811F	10Base2 module for PM 803F
EI 812F	AUI module for PM 803F
EI 813F	10BaseT module for PM 803F

Fieldbus modules:

FI 810F	CAN module
FI 820F	Serial module
FI 830 F	PROFIBUS DP module
FI 840 F	100BaseT fieldbus Ethernet module

Battery modules:

AM 801F	Battery module for redundant battery back-up for PM 802F
AM 811F	Battery module for redundant battery back-up for PM 803F

1.1.3 AC 700F process station

The AC 700F controller is a low-cost solution for small applications. The basic features of an AC 700F controller are:

- CPU module PM 783F with
- up to eight directly connected I/O modules with I/O terminal units,
- one fieldbus communication interface.

Depending on the type and quantity of process signals, up to eight plug-in type input/output modules can be directly connected. Additionally, an extensive PROFIBUS line can be connected via a PROFIBUS module.

This controller type cannot be used in a redundant configuration.

The following directly connected I/O modules are available for the AC 700F controller:

AI 723F	Analog input module: 16 AI/DI
AI 731F	Analog input module: 8 AI/DI
AO 723F	Analog output module: 16 AO
AX 721F	Analog input/output module: 4 AI/DI, 4 AO
AX 722F	Analog input/output module: 8 AI/DI, 8 AO
DA 701F	Digital/analog input/output module: 16 DI, 8 DI/DO, 4 AI/DI, 2 AO
DC 732F	Digital input/output module: 16 DI, 16 DI/DO
DI 724F	Digital input module: 32 DI
DX 722F	Digital input/output module: 8 DI, 8 DO (relay contacts)
DX 731F	Digital input/output module: 8 DI, 4DO (relay contacts)

The following fieldbus communication interfaces are available:

CM 772F	PROFIBUS module
CI 773F	PROFIBUS module

See Interface modules for AC 900F and AC 700F on page 65 for PROFIBUS module details and ***Engineering Manual I/O Modules for AC 700F / AC 900F*** for details on directly connected I/O modules.

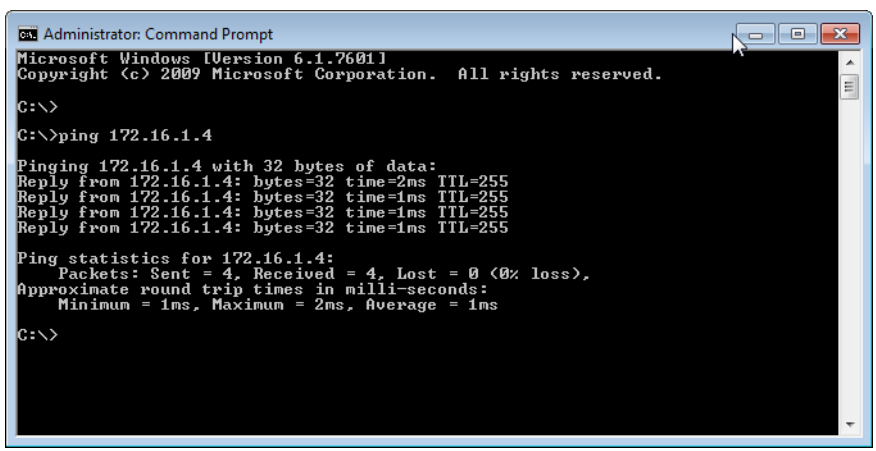
2 Firmware and operating system

2.1 Firmware update

The process station firmware consists of the boot software and the operating system. The firmware version of the process station must correspond to the Engineering software version. The firmware can be updated either during installation of the Freelance Software or at a later time via the Settings tool.

For this purpose, an Ethernet connection to the process station or – in a redundant configuration – to both process stations is required.

Prior to downloading a new firmware version to a CPU you should check that the network connection between the engineering PC and the process station works properly. For this purpose, a “ping” is entered in the command prompt window under Windows and then transmitted to the process station. A positive response indicates that there is a network connection to the process station.



ra002_us.png

Windows 7:
> Start > Programs > **Accessories** > **Command prompt**

Windows 10:
> Start menu > Windows System > Command Prompt

c:\> ping <IP address of the process station>

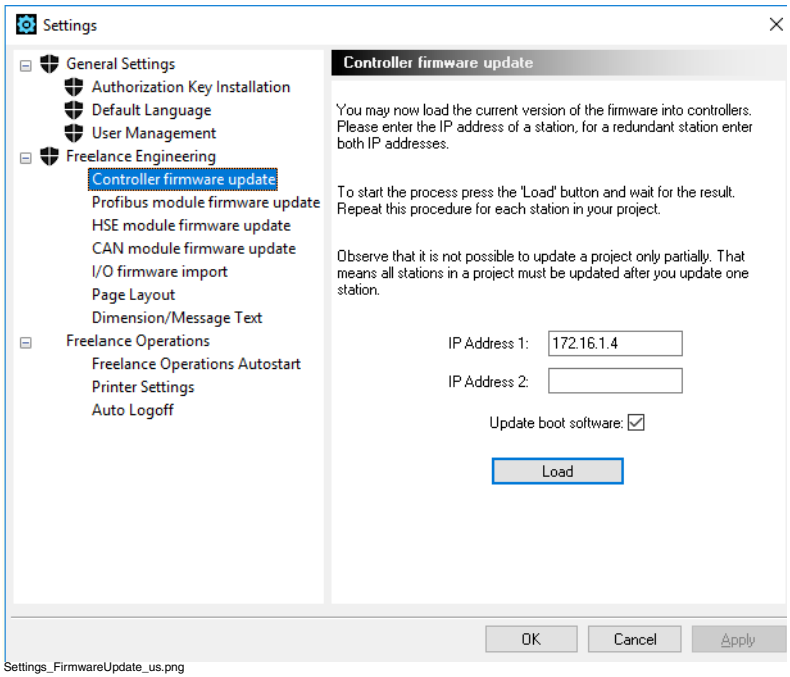
See the ***Mounting and Installation Manual*** of the corresponding process station for more information on setting and reading the IP address.

Start the **Settings** tool.

Windows 7:
> Start > Programs > ABB > Freelance <Version> > Settings

Windows 10:
> Start menu > ABB > Settings

In the tree structure, select *Controller firmware update*.



Settings_FirmwareUpdate_us.png

IP address Enter the process station IP address under *IP address 1*. Enter both addresses (*IP address 1* and *IP address 2*) for a redundant station.

Update boot software

The "Update Boot Software" option is only relevant for the AC 800F Controller.

☑ Automatically loads the boot loader along with the controller firmware (default).

☐ Only the firmware is loaded.
Select this option for loading the software to an AC 800F controller without updating it to a newer version.

Download The current version of the firmware is downloaded to the process station with the given IP address(es).

If you have more than one process station in your project, repeat the above-described procedure for each of them.

The firmware for the display is automatically loaded along with the AC 900F controller.

2.2 Operating system download

Usually, the operating system that has been downloaded together with the boot software already resides in the process station. An **Initialize all** command from Freelance Engineering deactivates the operating system. You can then start a **Boostrap** in the Engineering tool to download for the first time or update the operating system via the system bus.

The operating system bootstrap is normally executed automatically when the project is loaded for the first time by using the **Load whole station** option. However, it can also be performed upon a separate initialization action, independently of loading the user program.
When using two process stations in a redundant configuration, a **Bootstrap** will download the operating system to the Primary only. In this case, you have to use the **Settings** tool to download a new operating system version to the redundant controller as well.

2.3 Downloading the PROFIBUS module firmware

You can use the **Settings** tool to download the firmware to the PROFIBUS modules **CI 930F** (in an AC 900F controller), **FI 830F** (in an AC 800F controller) and **CI 773F** (in an AC 700F or AC 900F controller). In this case, the whole process station is initialized and the outputs of the connected I/O modules and field devices are de-energized.

The download of a new PROFIBUS firmware is always executed for all PROFIBUS modules of the addressed process station. The whole controller is initialized.

Using the **Settings** tool:

Select **PROFIBUS module firmware update**
> Enter process station IP address (enter both IP addresses for a process station in a redundant configuration)
> Download

Repeat the download operation for each controller.

2.4 Downloading the HSE module firmware

The **Settings** tool can be used to reconfigure the firmware of the **FI 840F** HSE module in an AC 800F controller. In this case, the whole process station is initialized and the outputs of the connected I/O modules and field devices are de-energized.

The download of a new HSE firmware is always executed for all FI 840F modules of the addressed process station. The whole controller is initialized.

Using the **Settings** tool:

Select **HSE module firmware update**
> Enter process station IP address (enter both IP addresses for a process station in a redundant configuration)
> Download

Repeat the download operation for each controller.

2.5 Downloading the CAN module firmware

The **Settings** tool can be used to reconfigure the firmware of the **CI 910F** CAN module in an AC 900F controller. In this case, the whole process station is initialized and the outputs of the connected I/O modules and field devices are de-energized.

The download of a new CAN firmware is always executed for all CAN modules of the addressed process station. The whole controller is initialized.

Using the **Settings** tool:

Select **CAN module firmware update**
> Enter process station IP address (enter both IP addresses for a process station in a redundant configuration)
> Download

Repeat the download operation for each controller.

3 Process station configuration

3.1 Configuration in the project tree

In the project tree, the various software components of a project are shown in a clear overview. The individual elements or objects, usually referred to as the project objects, are implemented according to the IEC 61131-3 standard. While the **Process station D-PS** and **Redundant process station D-PSR/RED** resources are used for the actual process data editing, the **Operator station D-OS** or **Gateway station D-GS** resources serve for process operation and monitoring and the **OPC Server OPC-S** resource for integrating data from other systems.

The **DP-S** or **D-PS/RED** resource in the project tree stands for a Freelance process station that is used to process the tasks and programs assigned to this resource. When inserting a resource, it is thus already defined whether the lower-level tasks and programs are processed in redundant or non-redundant manner. The assignment to the physical stations is made in the hardware manager. This assignment defines which available process, operator or gateway stations are used to process the individual programs.

The short name **D-PS** or **D-PS/RED** in the project tree indicates that no assignment to a physical station has been made yet in the hardware manager. Once the assignment has been made, the assigned station type is displayed. A redundant station is also represented by one resource in the project tree.

3.1.1 Inserting a resource in the project tree

All resources are inserted below the **Software (SW)** structural element.

Select the target position in the project tree.

> Insert > Above
a new process station is added above the selected object.

> Insert > Below
a new process station is added below the selected object.

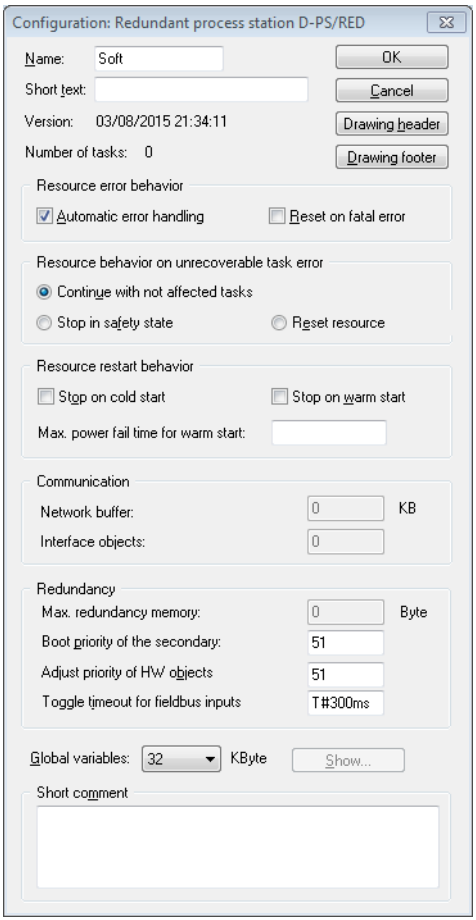
> Insert > Next level
a new process station is added to the next lower level.

The target position can be on the next lower level below the Software node, or on the Software node itself with Insert > Next level only.

> In the **Object Selection** dialog select **Process station D-PS** or **Redundant process station D-PS/RED** > **OK**

3.1.2 Configuring a process station

Double-click on the resource in the project tree to open the process station dialog. For a detailed description of all entries, see the ***Engineering Manual, System Configuration, Project tree***. This manual includes all parameters, the configuration of which has a direct impact on the behavior in the event of errors in particular. For further details, see Resource behavior in the event of an unrecoverable task error on page 124.



ProjectTree_ConfPS_us.png

Number of tasks

Indication of the number of tasks configured for this resource. With the **User application tasks** boot parameter, the memory required for this purpose will be reserved in the controller. See General boot parameters on page 39.

Resource error behavior

Automatic error handling

Errors in user programs (e.g. division by zero) are automatically corrected. See Automatic error handling at task level on page 121.

Reset on fatal error

A fatal error causes a stop of the CPU. When this item is selected, a cold start is performed after ten seconds and the application is continued.

Resource behavior on unrecoverable task error

The radio buttons are used to define the behavior of the process station in the event of an unrecoverable error during runtime (e.g. division by zero without enabled error correction).

Continue with not affected tasks

Only the task containing the erroneous user program is set to the “not executable“ state. All other tasks that are not affected continue the program processing. (Default setting)

Stop in safety state

When an unrecoverable task error occurs, the station is set to safety state to avoid the output of inconsistent data to the process by not affected tasks. The CPU module stops the processing and the outputs of the I/O modules assume their safety values.
With redundant process station configurations, a redundancy toggle is caused. The system must be reset manually in order to restart a stopped station.

Reset resource

When an unrecoverable task error occurs, the station is set to safety state. The station is reset and automatically restarted after 10 seconds. Depending on the type of error, the station is rebooted by a cold start or initialized.

For detailed information on error behavior, see Resource behavior in the event of an unrecoverable task error on page 124.

3.1.3 Inserting a task or a redundant task

In the process station, the actual programs run in tasks. The programs are processed either with program lists or with sequential function charts.

All tasks of a process station are included in the USRTask (user tasks) and SYSTask (system tasks) lists.

The following **system tasks** are created for each resource:

- **ColdSt** - the cold start task is performed once upon cold start; transition from **cold start** or **cold start stopped** to **r**unning state.
- **WarmSt** - The warm start task is performed once upon warm start; transition from **warm start** or **warm start stopped** to **r**unning state.
- **Run** - The run task is performed once during transition from **s**topped to **r**unning state. After completion of the run task, the user tasks are started.
- **Stop** - The stop task is performed once; transition from **running** to **stopped** state.
- **Error** - The error task is performed once upon detection of an error in a user program.
- **LatCSnd** - Lateral communication send task; performs cyclic lateral communication required for sending.
- **LatCRcv** - Lateral communication receive task; performs cyclic lateral communication required for receiving.
- **RedSt** - Redundancy start task; for redundant resources only; is performed directly once upon redundancy toggle prior to the first user task computing; this allows user programs triggered by a redundancy toggle to be executed.

The following **user tasks** can be created in a resource:

- Task
- Redundancy task
- Default task

Task

A task forms the frame for processing user programs. The programs in the user tasks are processed on a cyclic basis. The minimum cycle time is 5 ms. The processing trigger can be set to *Equidistant* or *Load optimal*. With the *Load optimal* task setting, the next task start time is calculated in each task cycle on the basis of the current time and the task cycle. During normal operation, there is no difference between the tasks. Only if a short load peak in the system prevents tasks computing and leads to delayed task processing, this difference has an impact. While

Equidistant tasks adhere to the fixed time pattern, *Load optimal* tasks slightly extend the cycles in the event of overload to relieve the system.

Redundancy task

A redundant task features redundancy data, i.e. these are data of the process image and function blocks for which a data adjustment will be performed after each task cycle. This ensures that a redundancy toggle can be performed at any time.

All variables of a redundant task must be written via the process image to render all data of a redundant task capable of redundancy.

Redundant tasks (TASK/RED) and non-redundant tasks (TASK) can be configured under the redundant resource. Consequently, only those functions of the user program that actually need to be redundant must be configured redundantly within a redundant resource.

Block import of sub-projects enables a resource to be automatically made redundant via the menu item **Edit > Import block as redundant**. This means that the project elements are created and all variables are written in process image mode.

With the export of individual process stations, the module equipment and the I/O channel assignment - which would otherwise be lost - are taken over on re-import.

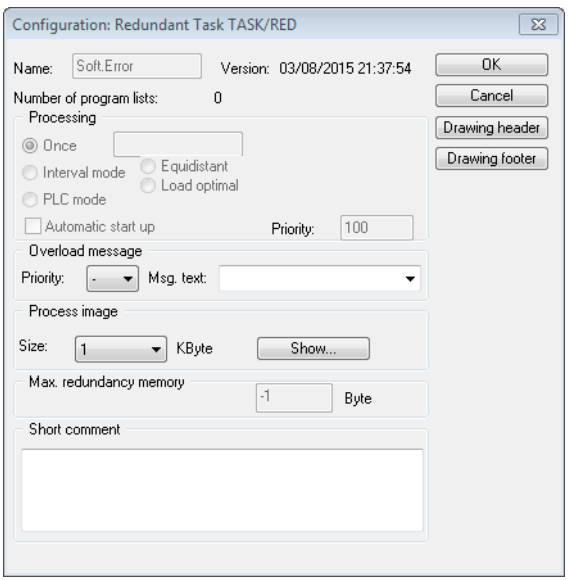
For more detailed information, see Changing a non-redundant process station into a redundant one on page 118.

Default task

One user task per resource can be generated as default task that computes in PLC mode. The default task has the lowest priority of all cyclic tasks and restarts automatically upon completion of each cycle. This results in a fastest possible cyclic processing of lower-level programs. Due to the low priority (50), the default task can be interrupted by all other tasks. For this reason, the default task is only executed when no other task is ready for computing.

3.1.4 Task configuration

Double-click on the project tree to open the task dialog. For a detailed description of all entries, see the ***Engineering Manual, System Configuration, Project tree***. This section describes all parameters that have a direct impact on the task behavior. For more detailed information, see Chapter 6, Processing and error handling.



ConfigreTask:_us.png

Processing Indication of the task mode *Once*, *Interval mode* or *PLC mode*. The task mode is defined by the type of task and cannot be set in the dialog.

Once Defined for system tasks only

Interval mode Standard mode for user tasks; with this setting, the task cycle time must be additionally configured.

PLC mode For default tasks only

Priority The task priority controls the processing of all tasks in “running” state. Multitasking of several tasks with identical cycle time is controlled by the priority.
User tasks: 51 (default value) to maximum 99 priorities can be assigned. All system tasks are configured with the priority 99; except for the error task = priority 100.

Lateral tasks: priorities of 51 to 99 can be assigned, as for user tasks.

Use the default task priorities, if possible. Configure redundant user tasks with task priorities below 95.

User tasks with excessively high priority, in particular when combined with fast task cycle times, may block the system communication and prevent data transmission to the operator and gateway stations, for instance.

Controller overload scenarios may be caused especially by redundant user tasks with high priority and fast task cycle times. Avoid using the redundant default task and possibly select task cycle times higher than 50 ms in order to minimize the overload risk of a redundant process station.

For more information about optimizing and analyzing CPU load, see Task Schedule Information - Controller Webpage on page 106.

3.2 Configuration in the hardware structure

Within the hardware structure, the resources defined in the project tree are assigned to the hardware actually required. Each D-PS resource in the project tree represents a process station in the hardware structure.

3.2.1 Inserting a process station

Process stations can be inserted into the system either in the tree view or in the system view. The selection of the station type defines whether the process station is operated in redundant or non-redundant mode.

> Select a system object (HWSYS) in the tree view
> **Edit** > **Insert**
> Select station type > Select a position for inserting > **OK**

or

> Double-click the free position in the system view
> Select station type > **OK**

Upon **Insert**, the object will appear in both the tree view and the system view. The object is shown in graphical form in the respective station position. Right-click the gray field in the header of the inserted process station in order to assign the resource.

3.2.2 Resource allocation

Right-click the gray field in the header of the inserted process station in order to assign the resource. When a station is assigned, it is automatically activated or a search operation is carried out in Freelance Engineering “Commissioning” mode to find the station on the system bus.

> Click the gray text field of the station
> **Edit** > **Resource allocation**
> Select the corresponding resource from the list

or

> Double-click the gray text field of the station
> Select the corresponding resource from the list

A redundant station must be assigned in the hardware structure for each redundant process station in the project tree.

3.2.3 Inserting modules in a process station

Modules can be inserted in the process station either in the tree view or in the station view. Select the appropriate slot and the desired type from the list to insert the module.

Upon **Insert**, the object appears in the tree view and the station view as well as in the system view.

For a redundant AC 900FR or AC 800FR process station, two identical stations are configured. In Freelance Engineering, insertion and parameterization of modules

must be performed only once for the upper IP1 controller (IP2 cannot be selected). All changes are automatically applied to the lower controller marked with IP2.

> Select process station in the tree view > **Edit** > **Insert**
> Select module type > Select a free slot

or

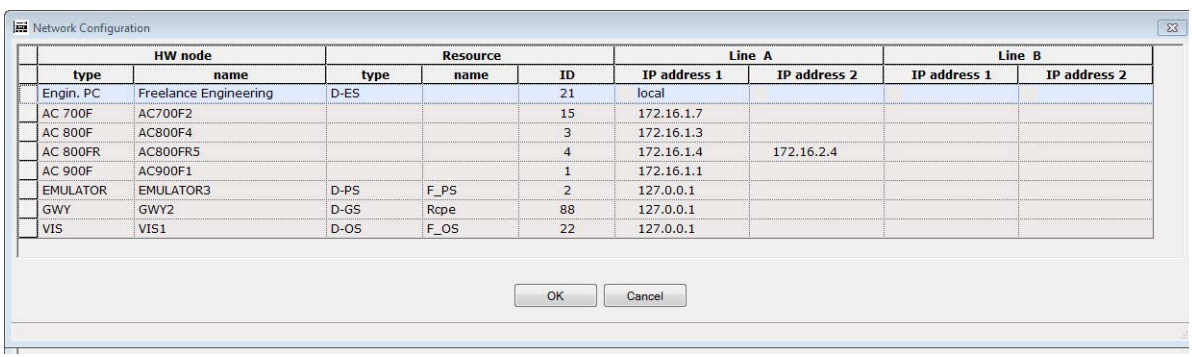
> Select a free slot in the station view > Double-click
> Select module type

When an object is inserted, a default name is assigned to it which represents the mounting position. When the object is copied, the object name is adapted to the new position. When the object is moved, the name remains unchanged.

3.2.4 Network configuration in the hardware structure

Each controller is provided with a resource ID and an IP address. In case of redundant process stations, the IP address is specified for each of the two controllers.
For redundant networks, the IP addresses for both networks - line A and line B - must be specified.

Configuration > System > Hardware structure > Network...



NetworkConfiguration_us.png

The assignment of the IP address 1 or 2 does **not** define whether the redundancy partner acts as Primary or Secondary.

3.3 General process station parameters

You can access the process station parameters by selecting the object in the tree view or graphics view.

Select the process station in the tree view > **Edit** > **Parameters...**

Each object in the hardware structure features the general data such as **Name**, **Short text** and **Long text**.

Name:	Name of the process station or the module. The name is taken from the tree structure and can be modified in the parameter definition dialog.
Short text:	Free text of up to 12 characters.
Long text:	Free text of up to 30 characters.

3.4 General boot parameters

In addition to the specific parameters for the different station types, the boot parameters for each process station must be configured.

Under normal circumstances, the boot parameters should not be modified. Each modification of a resource boot parameter will cause a project version error. The resource must be initialized to ensure that the modified boot parameter is effective.

Basic information that applies to all process stations is described below; the specific dialogs are explained in the following sections:
Parameters of the AC 900F controllers on page 43,
Parameters of the AC 800F and AC 800FR controllers on page 50 and
Parameters of the AC 700F controllers on page 59.

3.4.1 Process station memory

Although the actually available memory area depends on the type of process station used, the basic allocation is identical for all types. While the operating system is stored in the lower section, the configuration with the object directory and data as

well as the redundancy data memory for redundant stations are arranged above. The remaining memory area is available as RAM for current data.

The memory of a process station is composed of the following sections:

Memory for the operating system
+ Memory for the object directory (number of objects * 16 bytes)
+ Memory for the configuration data (PRAM)
+ Redundancy memory (for redundant stations only)
+ Memory for working data (RAM)

Using the Boot Parameters
- **Max. objects**
- **Configuration data (PRAM)** in kByte
- **Redundancy memory** (for redundant stations only)

the memory allocation within the process station can be modified.

The maximum number of objects is defined by the software architecture. The required minimum value depends on the system configuration. For this reason, it is recommended to start with the default number and to adjust the value accordingly. Bear in mind, that approximately 16 kByte are required for 1000 object entries and that the loaded objects need memory capacity as well.

There is no absolute limit in terms of adjustable storage areas. When a limit is reached, this limit can be modified. Such a change will reduce the remaining RAM.

If errors occur during loading into the process station, it can be tried to make the project loadable by changing the boot parameters **Max. objects** and **Configuration data (PRAM)**.

The current memory allocation of a process station can be checked with these **System variables** in commissioning mode:

<ps>.PRAM_SIZE	Size of the PRAM area set in the boot parameters of the process station
<ps>.PRAM_FREE	Proportion of free PRAM
<ps>.RAM_SIZE	Amount of RAM in use
<ps>.RAM_FREE	Proportion of free RAM

Attention should also be paid to the system variables PRAM_Free and RAM_Free after loading. If the PRAM_Free variable has a value of 0, the PRAM_SIZE value must be increased; If the RAM_Free variable has a value of 0, both the PRAM_ SIZE and the maximum number of objects must be reduced.

Max. objects

Maximum number of objects that can be configured on a resource. 1000 objects correspond to 16 kB.

Configuration data (PRAM)

Memory area in kByte that is reserved for configuration data. This memory area is cold-start resistant.

Redundancy memory

Memory area in kByte that is reserved for the transfer of redundancy data.

Currently required

Memory size in byte required by the project for the transfer of redundancy data as calculated by the system.

The required redundancy memory is displayed for redundant and plausible process stations only.

3.4.2 Communication management

In addition to the configured application programs, background processes for communication purposes are running in the process station. The memory for the operating system is subdivided into a section of fixed size and a variable section, the size of which depends on the number of communication links. Each communication link occupies internal system resources like additional memory capacity and system objects. For the AC 800F and the rack-based process stations, the **Network buffer** and **Interface objects** boot parameters can additionally be set.

Network buffer

Memory area in kByte reserved for communication links; the value determined by the system is displayed in the *Required* field, the *Reserve*d value is loaded into the process station.

Interface objects

Number of objects reserved for interfaces; the value determined by the system is displayed in the *R*equired field, the *Reserve*d value is loaded in to the process station.

Communication link	Required number of interface objects
Telecontrol - symmetric transmission FWK_DEV	6
Telecontrol - asymmetric transmission master FWK_DEV_M	6
Telecontrol - asymmetric transmission slave FWK_DEV_M	5
Telecontrol - data transmission FWK_DEV_TCP	6
PROFIBUS master PMDEV	2
PROFIBUS master PMEC1	3
PROFIBUS master PM930	4
PROFIBUS slave	0
Modbus master	3
Modbus slave	3
Modbus TCP master	3
Modbus TCP slave	1
Send	1
Receive	1
TCP Send and Receive SR_SRTCP(1)	2

(1) not supported with AC 900F and AC 700F

3.4.3 User application tasks

To ensure bumpless loading of configuration extensions into the process station, memory capacity is reserved during initialization for each configurable user application task. If this flexibility is not required, the reserved memory area can be reduced by the Number of user tasks boot parameter. By reducing the total number of configurable user tasks, additional memory capacity (approx. 12.5 kByte per task) is made available.

3.4.4 Communication links

A process station can manage up to 10 communication links to operator and gateway stations. The boot parameters **Operator stations** and **Gateway stations** define how many communication links to operator and gateway stations are provided for this process station. If the sum of both parameters is below the maximum admissible total number, additional memory capacity for the application is gained. See the ***Engineering Manual, System Configuration, Project tree.***

To calculate the actually required memory capacity of a process station, all configured boot parameters must be taken into consideration. If more than one parameter is set to a value close to the maximum value, the algorithm used for the plausibility check will not be completed.

Although no error is reported by the check routine, the loading process cannot be performed properly. The controller may signal a fatal error; in this case the user must remove the battery to perform a cold start.

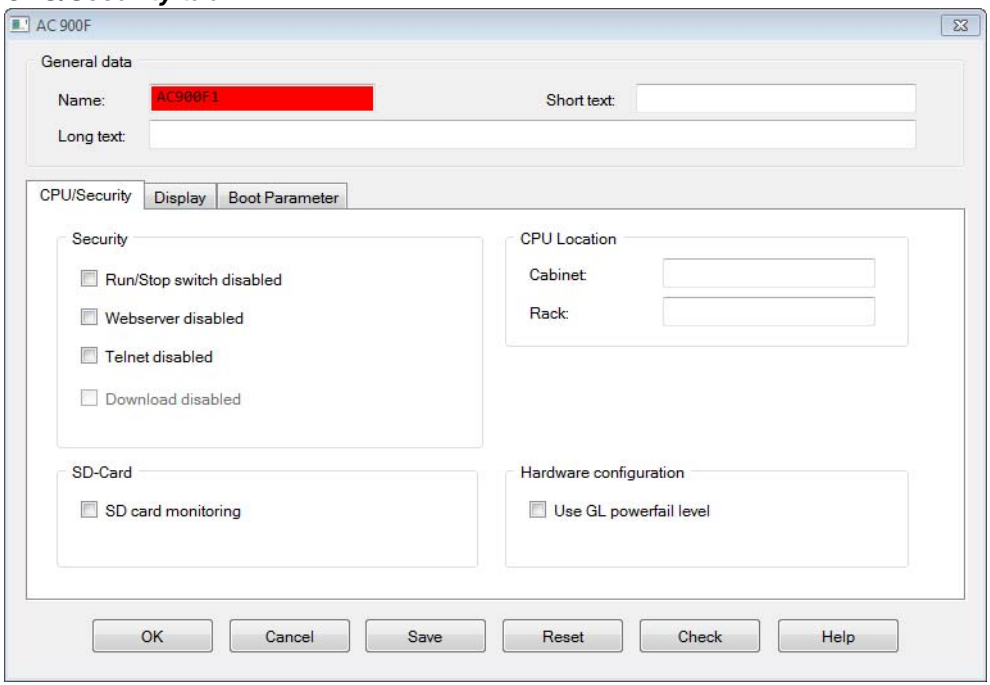
3.5 Parameters of the AC 900F controllers

The AC 900F process station can be used in six variants:

- AC 900F - Standard version
- AC 900FR - redundant configuration of the Standard version
- AC 900FL - Lite version with lower capacity
- AC 900FRL - redundant configuration of the Lite version
- AC 900FP - Plus version with higher capacity
- AC 900FRP - redundant configuration of the Plus version

As the parameter definition dialogs of these variants are almost identical, the following description applies to all of them.

***CPU/Security* tab**



AC 900F_1_us.png

This dialog is available for all AC 900F controller variants. For redundant controllers, the *Cabinet* and *Rack* fields are only provided once for each IP address.

Security

Run/Stop switch disabled

By ticking this checkbox, the Run/Stop switch at the module is disabled; the controller can thus not be switched on or off accidentally.

Webserver disabled

By ticking this checkbox, the webserver in the module is disabled; the controller website can thus not be accessed via the IP address.

Telnet disabled

By ticking this checkbox, the debug interface in the module is disabled; the controller can thus not be operated through a telnet program.

Download disabled

Display only. Blocking of downloads is done via the online functions of the Controller Display Unit (Main Menu / Controller / Security / F2).
If loading is disabled, then no configuration (application) can be loaded into the controller. The same applies to firmware updates. Interventions from Freelance Operations are not disabled. Furthermore, no online values can be displayed in the commissioning mode of the engineering, unless the variables have already been defined in the Value or Trend window before locking the controller. These previously defined variables can also be written (write value).

SD card

SD card monitoring

With activated card monitoring, a system message is released in Freelance Operations when the SD card is missing.

It is strongly recommended to activate the monitoring function if the “Auto Restore” service setting was selected for the controller.
Upon power failure, the controller checks the memory for the project configuration. If the memory is empty, the project configuration is automatically loaded from the SD card.
For detailed information, see the ***Mounting and Installation Manuals for the AC 900F and AC 700F Controller***s.

CPU location

Cabinet / Rack (IP1/IP2)

Free texts for the description of the CPU location can be used for the project documentation.
With redundant controllers, entries for IP1 and IP2 can be made.

Hardware configuration

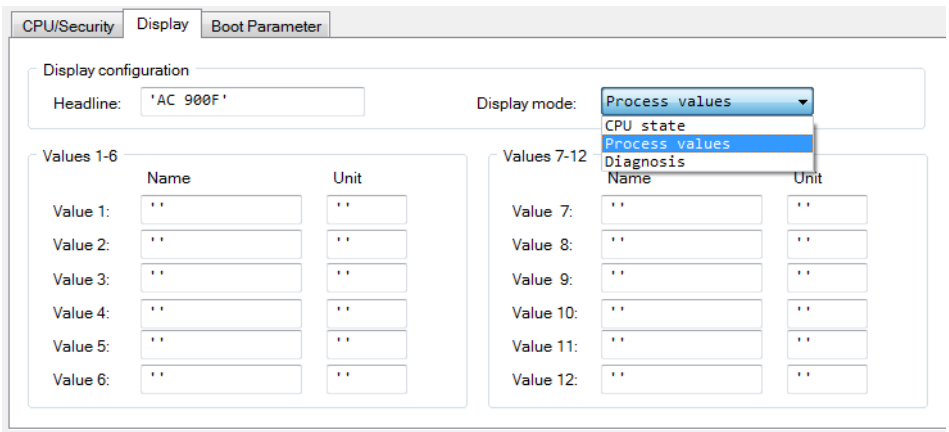
The power failure detection (powerfail) can be changed from the

default setting to the more exacting standards of Germanischer Lloyd (GL) according to the NAMUR guidelines.

Use Powerfail GL-Level

By ticking this checkbox, the higher requirements as per GL specification are activated for power failure detection.

Display tab



AC 900F_Display_us.png

This dialog is available for all AC 900F controller variants.

Display configuration

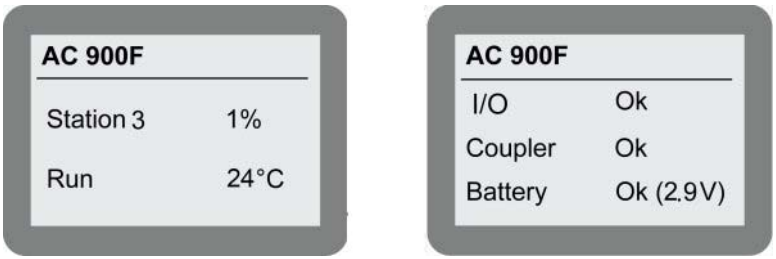
Configuration of the display on the front panel of the controller.

Headline Free text displayed in the header of the display. Up to 15 characters can be shown in the header of the display; the limiting ‘ characters are not displayed.

Display mode This setting is used to select the display page shown upon loading of the process station.

When to change the display mode from Process values to CPU state or Diagnosis, the change does not take effect automatically after loading the process station. A short change to the main menu and back (-> ESC -> OK) or a restart of the controller updates the display.

The **CPU state** and **Diagnosis** display pages are fixed:



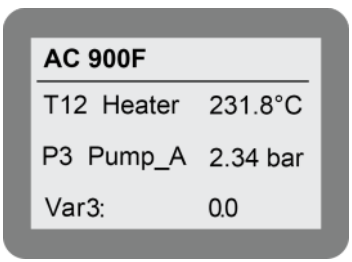
CPUstate_Diagnosis.png

For the process values page, 12 entries can be configured for the display of variables.

The I/O editor is used to configure the variables, the values of which are to be displayed.

Value 1-12, Unit:

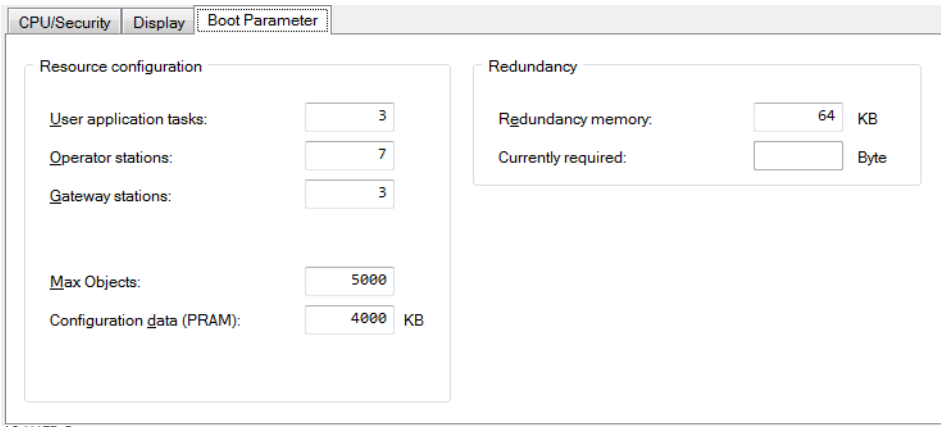
A name (max. 10 characters) and a unit (max. 4 characters) can be entered for each variable. The name and the unit are indicated before or after the variable value on the controller display. Without this configuration, the values are displayed as **Value 1** to **Value 12** without the unit being indicated.



Process_Values.png

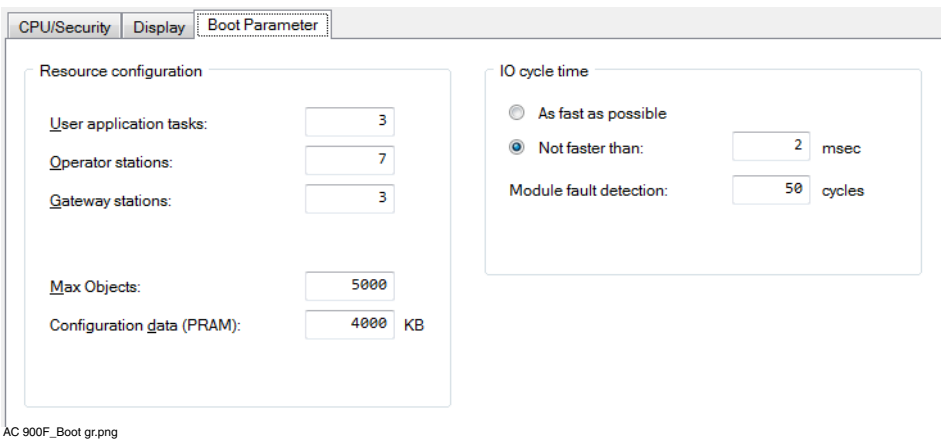
Boot Parameter **tab**

Boot parameters for redundant stations



AC 900FR_Boot gr.png

Boot parameters for non-redundant stations



AC 900F_Boot gr.png

Each modification of a resource boot parameter will cause a project version error. The resource must be initialized to ensure that the modified boot parameter is effective.

For detailed information, see General boot parameters on page 39.

Resource configuration

User application tasks

Maximum number of user tasks in this resource.

Operator stations

Number of permissible communication links to the operator stations. See ***Engineering Manual, System Configuration, Project tree.***

Gateway stations

Number of permissible communication links to the gateways (e.g. Trend, UNI). See ***Engineering Manual, System Configuration, Project tree.***

Max. objects

Maximum number of objects that can be configured on the resource.

Configuration data (PRAM)

Memory size in kByte that is reserved for the configuration data. A battery-buffered memory of up to 8 MB is available for the PRAM and RAM areas.

Redundancy - for redundant stations only

Redundancy memory

Memory area in kByte that is reserved for the transfer of redundancy data.

Currently required:

Shows the maximum size of memory in byte required by the project for the transfer of redundancy data.

I/O cycle time - for non-redundant stations only

Information on the cycle time of the I/O modules.

As fast as possible

The I/O communication determines the fastest possible cycle for I/O communication according to the configured modules. Depending on the configuration, the fastest possible cycle is between 0.8 ms and up to 2 ms.

Not faster than:

The cycle time can be limited to the value specified here in order to reduce the CPU load through I/O communication. The I/O communication will then calculate a cycle time that does not fall

below the specified minimum value. The possible value range is between 1 and 20 milliseconds.

Module fault detection

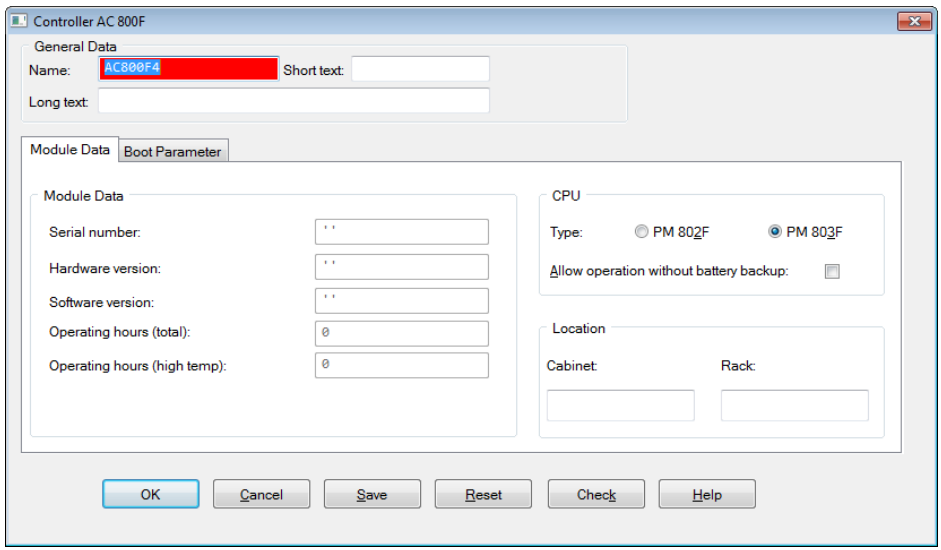
Number of I/O cycles for the detection of faults on an I/O module. Valid entries are 10 to 50 cycles.
The monitoring time results from the current I/O cycle time multiplied by the number of detection cycles.

When the monitoring time has elapsed, the input values immediately assume their configured safety value (hold last value or configured substitute value). The I/O modules recognize the interruption of communication when the monitoring time has elapsed and then assume their safety values.

See Interface modules for AC 900F and AC 700F on page 65 for fieldbus module details and ***Engineering Manual I/O Modules for AC 700F / AC 900F*** for details on directly connected I/O modules.

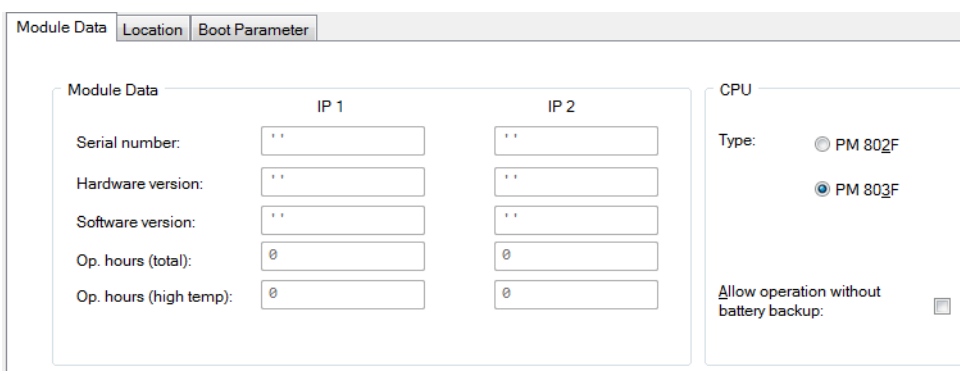
3.6 Parameters of the AC 800F and AC 800FR controllers

AC 800F *Module Data* tab



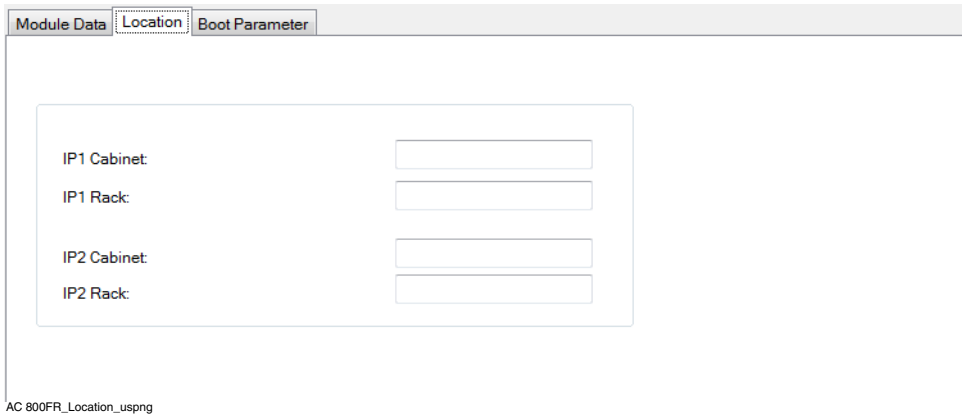
AC 800F_Module_data_uspng

AC 800FR *Module Data* tab



AC 800FR_Module_data_uspng

AC 800FR *Location* tab



Serial number

Shows the serial manufacturing number.

Hardware version

Indicates the hardware version of the AC 800F controller.

Software version

Indicates the firmware version.

Operating hours (total)

Shows the total number of operating hours.

Operating hours (high temp)

Shows the number of operating hours at excess temperature (excess temperature in the AC 800F controller refers to temperatures of above 70 °C).

These values are only available in commissioning mode.

For information on up-dating the operating system, see Operating system download on page 26.

CPU

PM 802F

type of the basic unit used

◉ PM 802F basic unit with 4 MByte flash EPROM and 4 MByte RAM.

PM 803F ◉ PM 803F basic unit with 8 MByte flash EPROM and 16 MByte RAM.

Allow operation without battery back-up

This parameter is only evaluated by the PM 803F basic unit. If no module with increased battery capacity is installed in slot E1 or E2, this will be reported during the plausibility check.

☑ A plausibility warning is released when the battery back-up is missing.

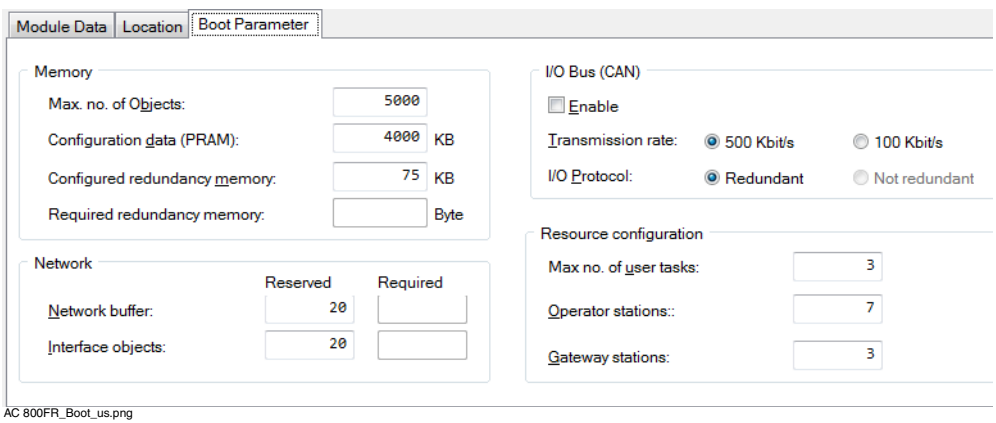
☐ A plausibility error is reported when the battery back-up is missing.

Location

Cabinet / Rack (IP1/IP2)

Free texts for the description of the controller location can be used for the project documentation.

Boot parameters - AC 800F and AC 800FR



AC 800FR_Boot_us.png

Each modification of a resource boot parameter will cause a project version error. The resource must be initialized to ensure that the modified boot parameter is effective.

For detailed information, see General boot parameters on page 39.

Memory Memory allocation within the resource. with an extremely high number of objects in the process station, errors may occur on loading the objects. By changing this setting it may be possible to load the project.

Maximum no. of objects
Maximum number of objects that can be configured on the resource.

Configuration data (PRAM)
Memory size in kByte that is reserved for the configuration data. This memory area is cold-start resistant.
4 or 16 MB of battery-buffered memory is available in total.

Configured redundancy memory (for redundant controllers only)
Memory size in kByte that is reserved for the transmission of redundancy data.

Required redundancy memory (for redundant controllers only)
Maximum memory size in byte required by the project for the transmission of redundancy data.

Network Information on processes that run in the background. Every communication link occupies internal system resources like additional memory capacity or system objects.

Network buffer
Memory size in kByte reserved for communication links.

Interface objects
Number of objects reserved for communication interfaces.
See Communication management on page 41.

I/O bus (CAN)

Enable Used to configure the F1 slot of the AC 800F controller. If a CAN module is connected to the F1 slot of the AC 800F controller, the CAN bus must be enabled here. If another module is used on the F1 slot, the CAN bus must be disabled.

Transmission rate
Selection of the transmission rate of 100 or 500 kbit/s on the station bus (default 500 kbit/s). The maximum distance of the I/O modules of a process station is linked to the transmission rate.

Transmission rate	CAN bus length
500 kbit/s	80m
100 kbit/s	400m

I/O protocol Selection of the I/O protocol

Redundant Default setting. All I/O modules connected by CAN modules must be equipped with EPROMs that are capable of redundancy.

Not redundant

This I/O protocol does not support redundancy. Provided for the mixed operation of I/O modules that are capable of redundancy or not capable of redundancy. This mode can only be selected for non-redundant resources.

Resource configuration

Max. no. of user tasks

Maximum number of user tasks in this resource.

Operator stations

Number of permissible communication links to the operator stations. See the ***Engineering Manual, System Configuration, Project tree***.

Gateway stations

Number of permissible communication links to the gateways (e.g. OPC, Trend, UNI). See the ***Engineering Manual, System Configuration, Project tree***.

3.6.1 AC 800F modules

Power supply module SA 801F/SD 802F/SA 811F/SD 812F

Power supply module for the AC 800F controller. This module must be installed in each basic unit in the P slot (first slot on the left side of the basic unit).

There are different variants:

- The SA 801F power supply module for 115/230 V AC and the SD 802F power supply module for 24 V DC for the PM 802F basic unit.
- The SA 811F power supply module for 115/230 V AC and the SD 812F power supply module for 24 V DC for the PM 803F basic unit.

Ethernet modules EI 801F/EI 802F/EI 803F/EI 811F/EI 812F/EI 813F

Ethernet connection module of the AC 800F controller. This module must be installed in each basic unit in the E1 or E2 slot (second and third slots on the left side of the basic unit).

There are three different variants:

- EI 801F / EI 811F Ethernet module with 10Base2 (Cheapernet or BNC),
- EI 802F / EI 812F Ethernet module with AUI connector and
- EI 803F / EI 813F Ethernet module with twisted pair connection.

The Ethernet modules EI 811F, EI 812F and EI 813F feature higher battery capacity for the supply of the PM 803F basic unit.

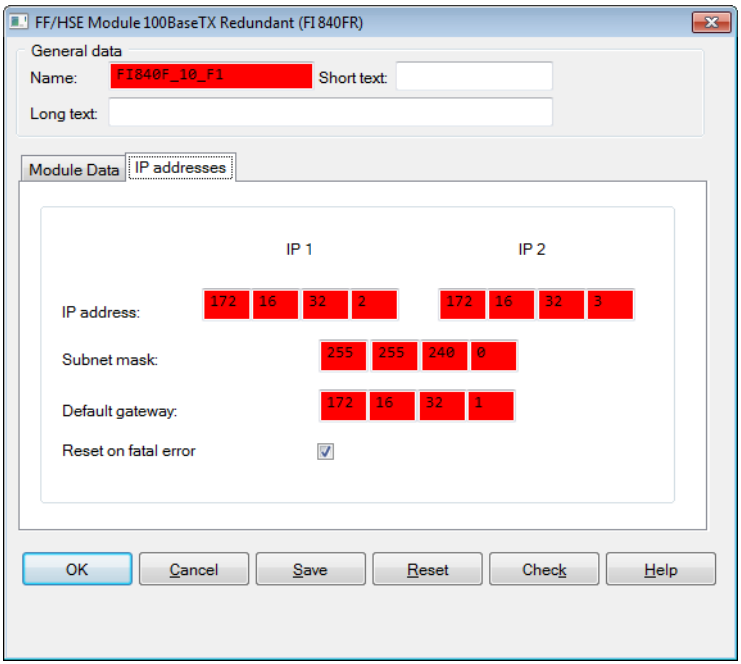
Battery modules AM 801F/AM 811F

Battery module of the AC 800F controller. The module can be additionally installed in each basic unit and serves for redundant battery buffering of the internal memory (RAM). The module must be installed in the E1 or E2 slot (second and third slots on the left side of the basic unit).

The AM 811F battery module features a higher battery capacity than the PM 803F basic unit.

Fieldbus Ethernet module FI 840F

Fieldbus Ethernet module of the AC 800F controller. The module features a 100BaseT Ethernet interface that can be used for the FOUNDATION fieldbus HSE protocol and for TCP/IP send and data receiving communication (UDP sub-protocol). The module can be installed on the F1 to F4 slots.



FFHSE_IP_addr_us.PNG

IP address

IP 1/2 IP address of the FI 840F module in the AC 800F controller at position IP1 or IP2.

For a non-redundant AC 800F, only one IP address must be entered.

Subnet mask Subnet mask for the FI 840F modules in both AC 800F controllers.

Default gateway

IP address of the default gateway.

Reset on fatal error

☑ In the event of a fatal error, the CPU of the FI 840F module is re-started.

☐ In the event of a fatal error, the CPU of the FI 840F module is stopped. The FI 840F module must be switched off and on again to be re-started.

During normal operation, the *Reset on fatal error* parameter should be activated.

Using the **Settings** tool, the firmware on the HSE module can be updated. See the ***Manual Introduction – Getting Started, Installation***.

PROFIBUS module FI 830F

PROFIBUS module of the AC 800F controller. Each PROFIBUS module allows the connection of one PROFIBUS line, i.e. the connection of a maximum of 125 slaves. Each of these slaves can also be modular, i.e. contain a maximum of 64 modules. The module can be installed on the F1 to F4 slots.

Using the **Settings** tool, the firmware on the PROFIBUS module can be updated. See Downloading the PROFIBUS module firmware on page 26.

FI 820F serial communication interface

Serial communication interface of the AC 800F controller. The serial module has two interfaces that can be occupied either by the Modbus master interface, the Modbus slave interface or the telecontrol interface protocol. The interface objects can be configured below the serial module. The module can be installed on the F1 to F4 slots.

FI 810F CAN module

CAN module of the AC 800F controller. The CAN module allows connection of a maximum of 5 rack I/O units and thus the connection of 45 I/O modules as used in the rack-based Freelance process stations. One CAN master must be configured below the CAN module.

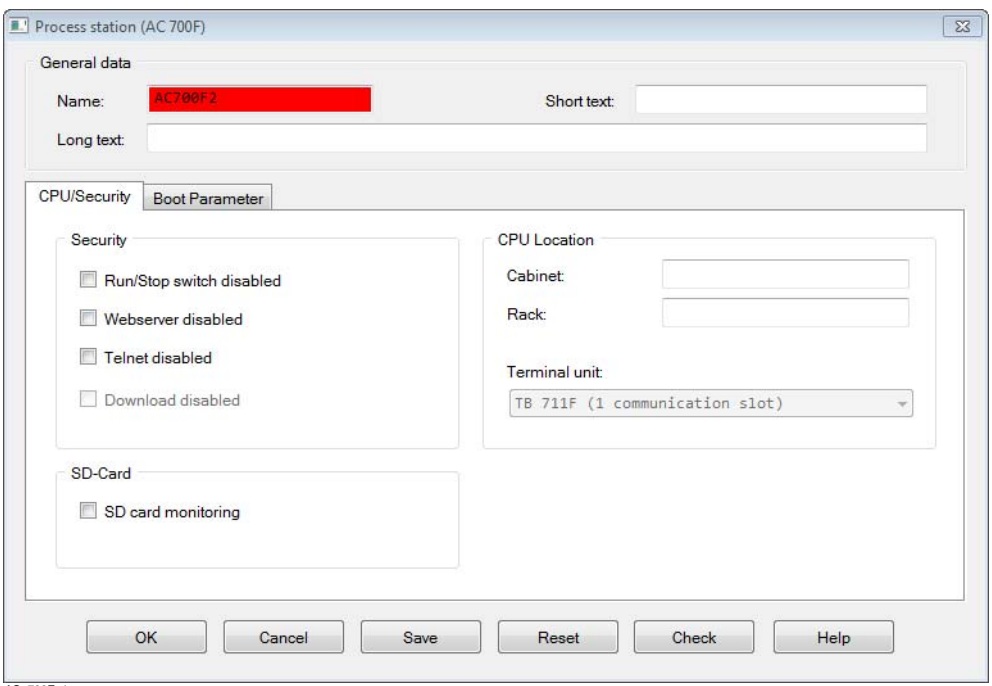
Only one station bus line with a maximum of 5 sub-racks can be connected per AC 800F controller.

The F1 slot of the FI 810 module is specified for this purpose. If a CAN module is used on the F1 slot of the AC 800F controller, the CAN bus (stations bus) must be enable in the boot parameters of the resource. If another module is used on the F1 slot, the CAN bus must be disabled.

For detailed information on parameterization and object data, see Parameters of the AC 800F and AC 800FR controllers on page 50 and STA, error code encoding - AC 900F and AC 900FR on page 141.

3.7 Parameters of the AC 700F controllers

***CPU/Security* tab**



AC_700F_1_us.png

Security

Run/Stop switch disabled

By ticking this checkbox, the Run/Stop switch at the module is disabled; the controller can thus not be switched on or off accidentally.

Webserver disabled

By ticking this checkbox, the webserver in the module is disabled; the controller website can thus not be accessed via the IP address.

Telnet disabled

By ticking this checkbox, the debug interface in the module is disabled; the controller can thus not be operated through a telnet program.

Download disabled

Only available during commissioning.
After having loaded the controller, this parameter prevents program changes from being downloaded in the process station.

SD card

SD card monitoring

With activated card monitoring, a system message is released in Freelance Operations when the SD card is missing.

It is strongly recommended to activate the monitoring function if the “Auto Restore” service setting was selected for the controller.
Upon power failure, the controller checks the memory for the project configuration. If the memory is empty, the project configuration is automatically loaded from the SD card.
For detailed information, see the ***Mounting and Installation Manuals for the AC 900F and AC 700F Controller***s.

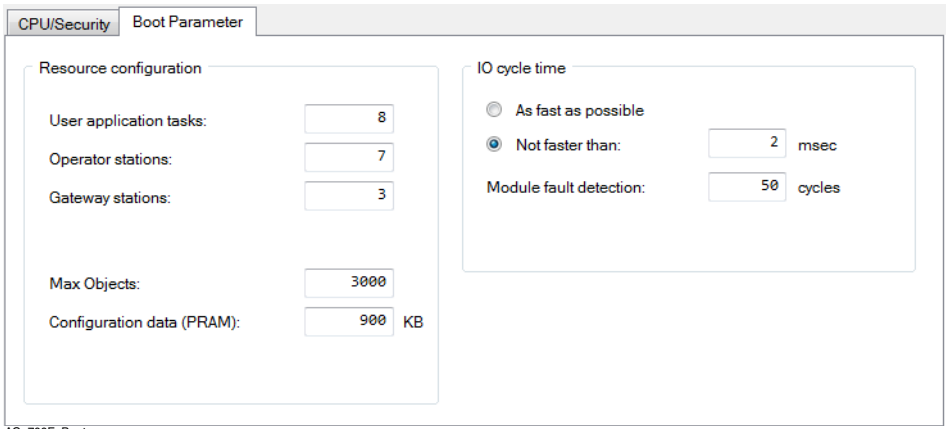
CPU Location

Cabinet / Rack

Free texts for the description of the CPU location can be used for the project documentation.

Terminal unit Type of CPU terminal unit.

Boot Parameter tab



AC_700F_Boot_us.png

Each modification of a resource boot parameter will cause a project version error. The resource must be initialized to ensure that the modified boot parameter is effective.

For detailed information, see General boot parameters on page 39.

Resource configuration

User application tasks

Maximum number of user tasks in this resource.

Operator stations

Number of permissible communication links to the operator stations. For a detailed description see ***Engineering Manual System Configuration, Project tree***.

Gateway stations

Number of permissible communication links to the gateways (e.g. OPC, Trend). For a detailed description see ***Engineering Manual System Configuration, Project tree***.

Max. objects

Maximum number of objects that can be configured on the resource.

Configuration data (PRAM)

Memory size in kByte that is reserved for the configuration data.

This memory area is cold-start resistant.
A battery-buffered memory of up to 2 MB is available for the PRAM and RAM areas.

I/O cycle time Information on the cycle time of the I/O modules.

As fast as possible:

The I/O communication determines the fastest possible cycle for I/O communication according to the configured modules. Depending on the configuration, the fastest possible cycle is between 0.8 ms and up to 2 ms.

Not faster than:

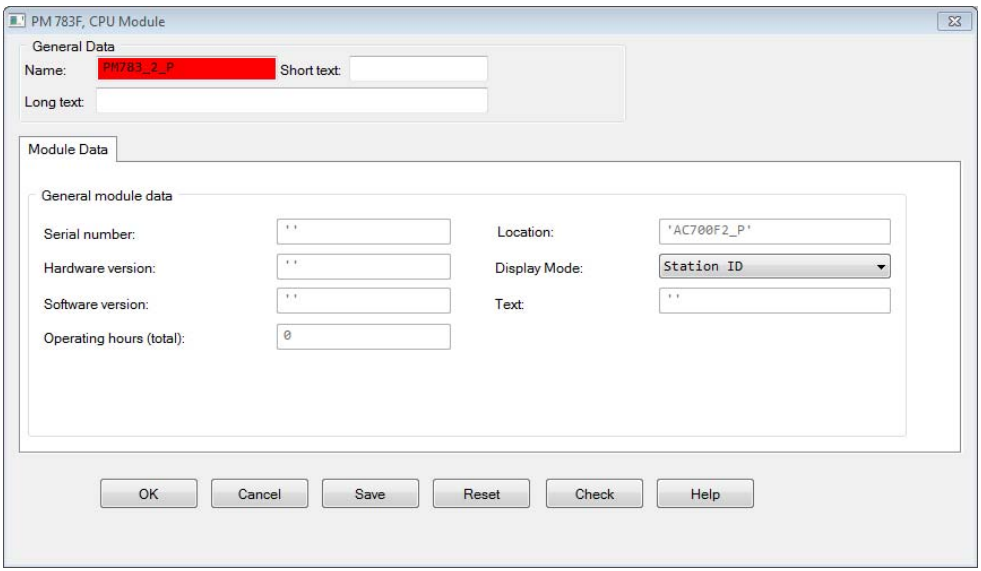
The cycle time can be limited to the value specified here in order to reduce the CPU load through I/O communication. The I/O communication will then calculate a cycle time that does not fall below the specified minimum value. The possible value range is between 1 and 20 milliseconds.

Module fault detection

Indicates the monitoring cycles of the I/O modules. 10 up to max. 50 cycles are possible. The monitoring time results from the current I/O cycle time multiplied by the number of monitoring cycles.

When the monitoring time has elapsed, the input values immediately assume their configured safety value (hold last value or configured substitute value). The I/O modules recognize the interruption of communication when the monitoring time has elapsed and then assume their safety values.

Parameters of the PM 783F CPU module



PM_783F_CPU_us.png

General Data:

Serial number:

Shows the serial manufacturing number.

Hardware version:

Shows the hardware version of the module.

Software version:

Shows the version of the software.

Operating hours (total):

Shows the total number of operating hours.

These values are only available in commissioning mode.

Location: Indicates the module location in the hardware structure. The location is composed of the resource name and the slot number. The location is provided by the system and cannot be edited here.

Display mode:

Defines the values indicated on the CPU module display. **Free text**, the **Station ID**, **Run/Stop state**, the **CPU load** or **Process value**s can be displayed.

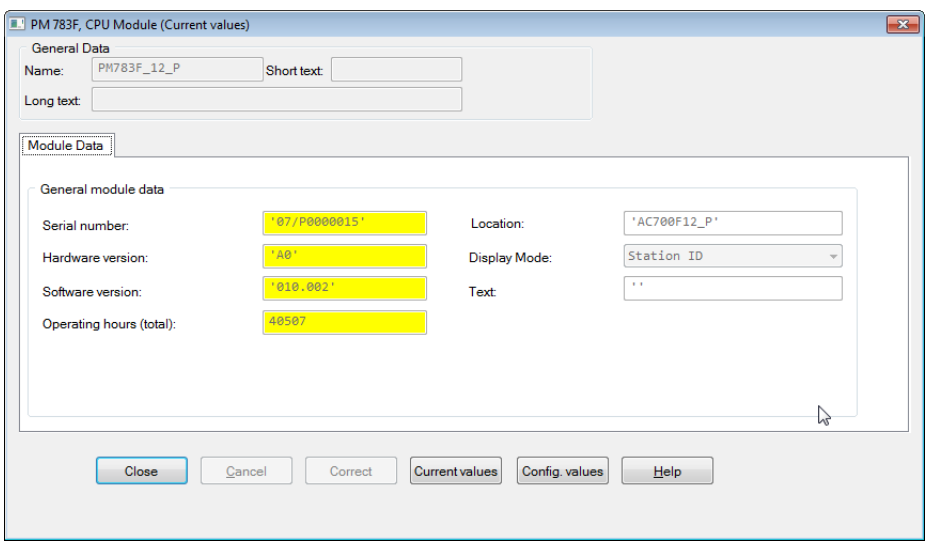
Text: If the “Free text” option is selected, free text (max. 6 characters) can be entered here and displayed on the LCD (6-digit, 7-segment display) of the CPU module.

See Interface modules for AC 900F and AC 700F on page 65 for fieldbus module details and ***Engineering Manual I/O Modules for AC 700F / AC 900F*** for details on directly connected I/O modules.

3.8 Module data/general module data

The values of module data or general module data are only available in commissioning mode. Depending on the selected process station type, these data are directly displayed in the dialogs of the process station or in the dialogs of the CPU module.

The dialog entries are described below using the PM 783F CPU module as an example:



PM_783F_Commissioning_us.png

With redundant process stations, two entries are displayed, i.e. for the modules with the IP addresses 1 and 2.

Serial number:
Shows the serial manufacturing number.

Hardware version:
Shows the hardware version of the module.

Software version:
Shows the version of the software.

Operating hours (total):
Shows the total number of operating hours.

3.9 Interface modules for AC 900F and AC 700F

3.9.1 PROFIBUS module CI 930F

The CI 930F interface module is the PROFIBUS DP master for an AC 900F process station. The CI 930F can be used in both a redundant and in a non-redundant AC 900F controller configuration.

It can be plugged either into the C1 or the C2 slot on the left side of the CPU module of the process station.

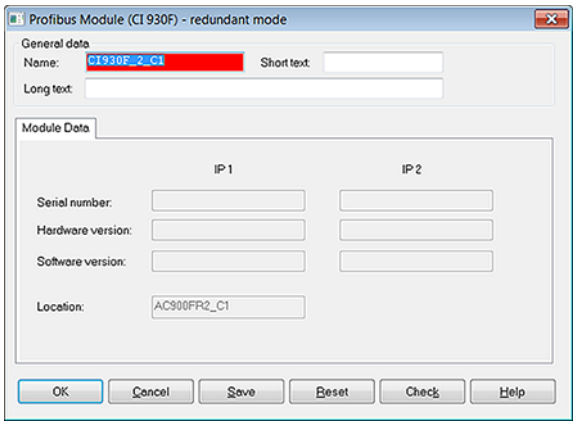
3.9.2 CI 930F parameters

The PROFIBUS module parameters can be called up for display via the parameter setting masks. The current values can also be read in commissioning mode.

Select the module in the tree view > Right-click > Select **Parameters**...

or

> Double-click the module in the graphic structure



CI930F_parameters_us.png

General data

Name Name of the PROFIBUS module. The name is taken over from the tree structure and can be edited.

Short text Free text of up to 12 characters.

Long text Free text of up to 30 characters.

Module data

Serial number Displays the serial manufacturing number.

Hardware version
Indicates the hardware version of the module.

Software version
Indicates the version of the software.

These values are only available in commissioning mode.

Location Indicates the module location in the hardware structure. The location name is composed of the resource name (here: *AC900FR2*) and the slot name (*C1*). The location is provided by the system and cannot be edited.

3.9.3 PROFIBUS module CM 772F / CI 773F

The CM 772F or CI 773F communication interface is the PROFIBUS DP master of an AC 900F and AC 700F process station. It can be plugged either into the C1 or the C2 slot on the left side of the CPU module of the process station.

The module type CM 772F is supported by PM 902F, but not by PM 901F and PM 904F.

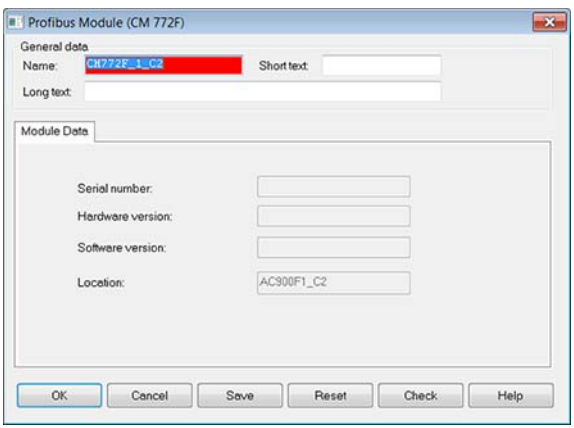
3.9.4 CM 772F / CI 773F Parameters

The PROFIBUS module parameters can be called up for display via the parameter setting masks. The current values can also be read in commissioning mode.

Select the module in the tree view > Right-click > Select **Parameters**...

or

> Double-click the module in the graphic structure



CM772F_AC900F_us.png

General data

Name	Full name of the PROFIBUS module. The name is taken over from the tree structure and can be edited.
Short text	Free text of up to 12 characters.
Long text	Free text of up to 30 characters.

Module data

Serial number Displays the serial manufacturing number.

Hardware version
Indicates the hardware version of the module.

Software version
Indicates the version of the software.

These values are only available in commissioning mode.

Location Indicates the module location in the hardware structure. The location name is composed of the resource name (here: *AC900F*1) and the slot name (*C*2). The location is provided by the system and cannot be edited.

3.9.5 CAN module CI 910F

The CI 910F interface module is the CAN master for an AC 900F process station. The CI 910F can be used in both a redundant and in a non-redundant AC 900F controller configuration.

It can be plugged either into the C1 or the C2 slot on the left side of the CPU module of the process station.

The CI 910F CAN modules can be used for Freelance Rack I/O only with firmware versions > 4.0 which supports redundancy.

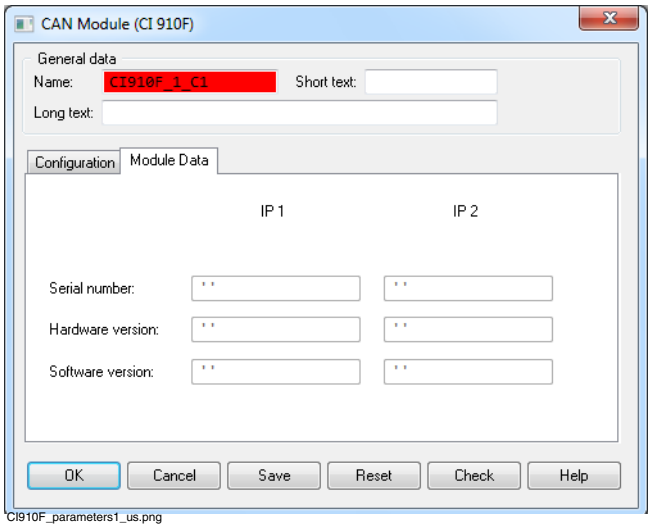
3.9.6 CI 910F parameters

The module parameters can be called up for display via the parameter setting masks. The current values can also be read in commissioning mode.

Select the module in the tree view > Right-click > Select **Parameters**...

or

> Double-click the module in the graphic structure



CI910F_parameters1_us.png

General data

Name Name of the CAN module. The name is taken over from the tree structure and can be edited.

Short text Free text of up to 12 characters.

Long text Free text of up to 30 characters.

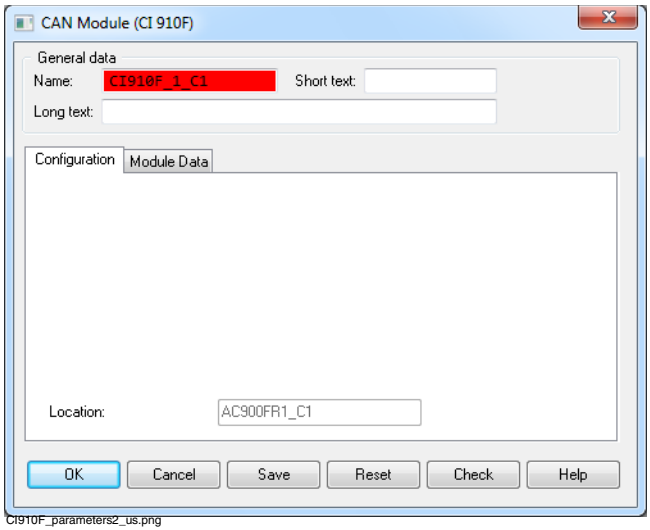
Module data

Serial number Displays the serial manufacturing number.

Hardware version
Indicates the hardware version of the module.

Software version
Indicates the version of the software.

These values are only available in commissioning mode.



CI910F_parameters2_us.png

Configuration

Location Indicates the module location in the hardware structure. The location name is composed of the resource name (here: *AC900FR1*) and the slot name (*C1*). The location is provided by the system and cannot be edited.

Configuration: CAN master object DNETP

CAN master objects can be inserted below the CAN module CI 910F. Following the **INSERT** command, the object appears in both tree and module views. A CAN master object (DNETP) can be configured for each CAN module. The bus parameter for the CAN bus will be defined here. For more information on configuration of a CAN Module in the hardware structure, refer to ***Engineering Manual System Configuration, Hardware Structure***.

Parameterizing the CAN master object

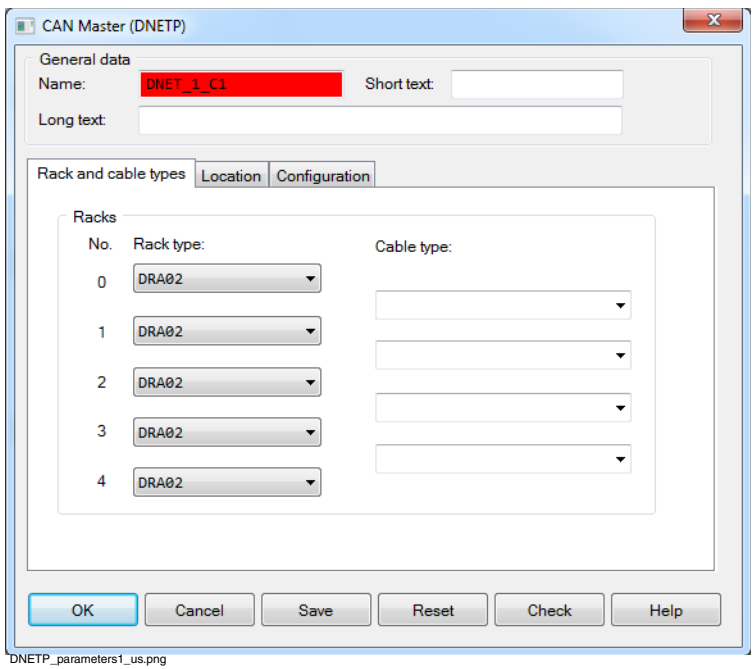
The CAN master object (DNETP) represents the CAN master in Freelance. The CAN master object is used to configure interface and bus parameters for CAN bus.

The I/O data will not be available. The I/O data is assigned through the connected slaves (rack station).

Select the CAN master object in the tree view > Right-click > Select **Parameters**...

or

> Double-click the CAN master object in the graphic structure



DNETP_parameters1_us.png

General data

Name Name of the CAN master object. The name is taken over from the tree structure and can be edited.

Short text Free text of up to 12 characters.

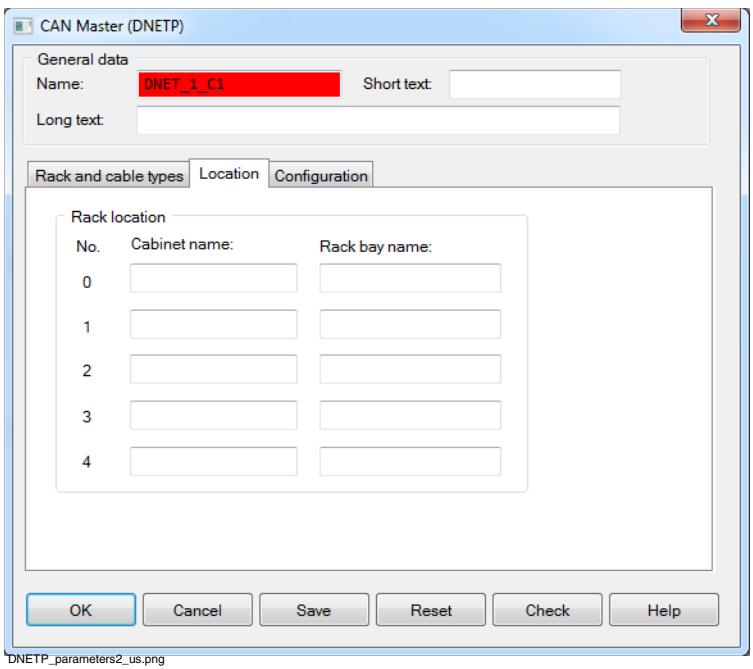
Long text Free text of up to 30 characters.

Rack and cable types

Rack type

The connected rack type can be selected for project documentation:
DRA 01 (delivery until end of 1997)
DRA 02 with 10 module slots
DRA 03 with 3 module slots
DRA 04 with 5 module slots

Cable type The used CAN cable tpye for station bus can be selected for project documentation.

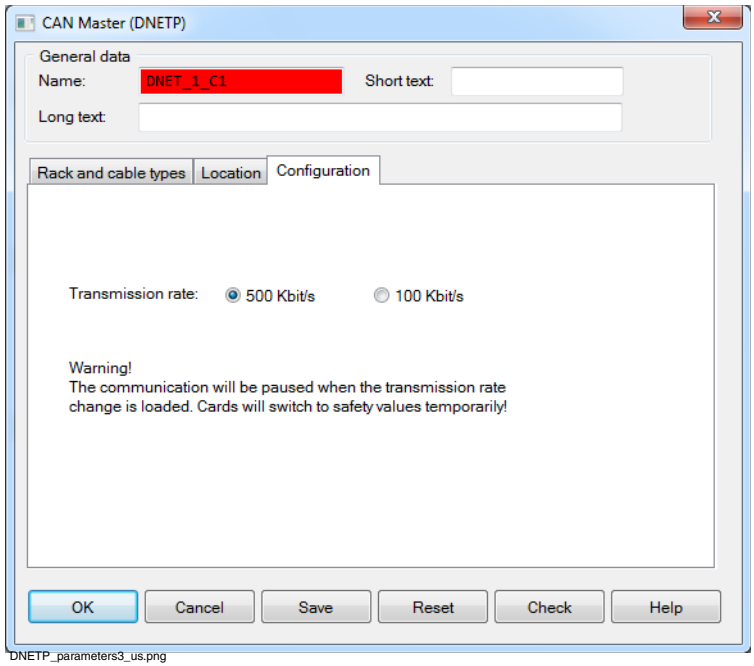


DNETP_parameters2_us.png

Location

Cabinet name / Rack bay name

Free texts for the description of the location of connected rack station can be used for the project documentation.



DNETP_parameters3_us.png

Configuration

Transmission rate

Selection of the transmission rate of 100 or 500 kbit/s on the station bus (default 500 kbit/s). The maximum distance of the I/O modules of a rack station is linked to the transmission rate.

Transmission rate	CAN bus length
500 kBit/s	80m
100 kBit/s	400m

3.10 Permanent storage of user variables on AC 900F

In a Freelance controller all user data (e.g. IEC1131 global variables) is stored together with the configuration in battery buffered RAM. This memory gets initialized on configuration replacement (“Load whole station”), at cold start, firmware update or on power down with empty or removed battery.

For certain applications it is desirable that values can be stored permanently. With the function “Non-volatile Data Storage” 24 REAL variables, 24 DINT variables and 6 DWORD variables can be stored permanently.

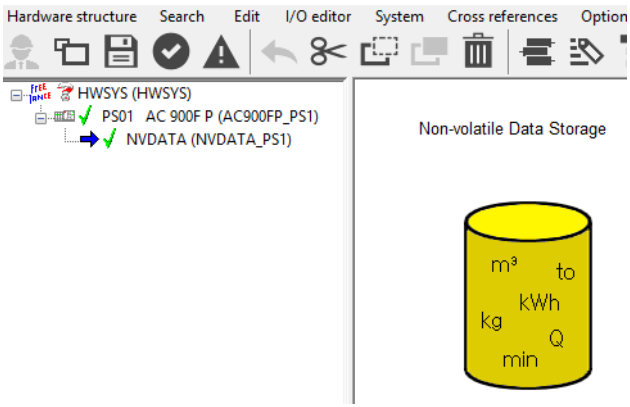
The non-volatile data storage (NVDATA) feature is only available for AC 900F controllers.

The current values of the variables configured for permanent storage are written cyclically into the permanent memory. The cycle time is configurable in the range of 3min to 20min. In addition, these variables are also written to permanent memory during a scheduled cold start of the controller. For example, when loading (whole station), cold start via the display menu or during cold start by means of a function block (FCS). Updating the permanent memory is not possible during hardware-controlled restarts (e.g. pressing reset button >4s). However, the last actual values are still available when the controller performs a warm start.

3.10.1 Inserting non-volatile storage function block NVDATA

> Select AC 900F controller object in the tree view (Hardware structure)
> **Edit** > **Insert** > select NVDTA > **OK**
or
> Select AC 900F controller object in the tree view (Hardware structure)
> **right mouse click** > **Insert** > select NVDATA > **OK**

After inserting, the object appears in the tree view.

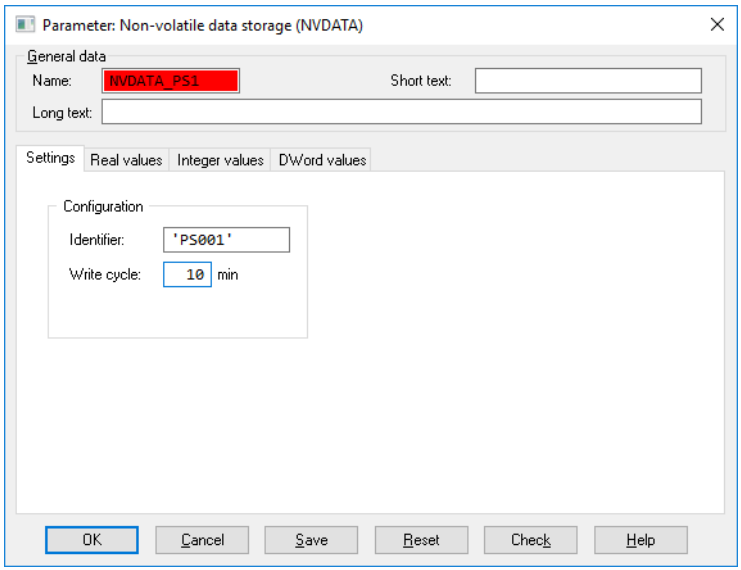


DNVDATA1_en.png

3.10.2 Configuration of Non-Volatile Data Storage (NVDATA)

The NVDATA function parameters can be called up for display via the parameter setting masks. The current values can also be read in commissioning mode.

Select the NVDATA object in the tree view > Right-click > Select **Parameters**...
or
> Double-click the NVDATA object in the hardware structure



DNVDATA_Conf11_en.png

General data

Name — Name of the NVDATA object. The name is taken over from the tree structure and can be edited.

Short text — Free text of up to 12 characters

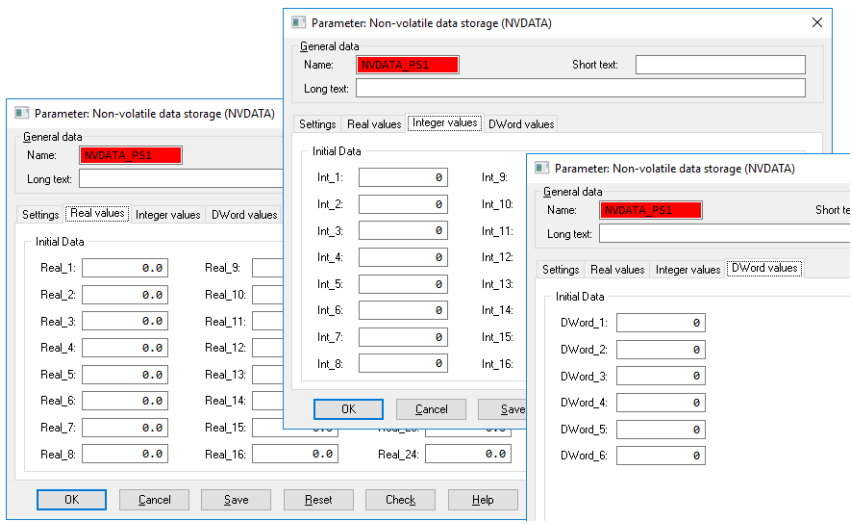
Long text — Free text of up to 30 characters

Settings

Identifier — To avoid using data from another application (for example, in the case of a controller replacement), a user-configurable name is provided as an identifier. If the configured name does not match the identifier present in the controller at startup, the memory contents are reset to the configured initial values. The minimum length of the name is 5 characters.

Write — Cycle time in which the data is written (3 - 20 min)

Due to the technically limited lifetime of the permanent memory, the cycle time should be set as high as possible. With a write cycle of 10 min (default value), a life time of min. 35 years can be expected.



NVDATA_Initialvalues_en.png

Real Values, Integer Values, DWord Values

Initial data Real_1 - Real_24

Initial data Int_1 - Int_24

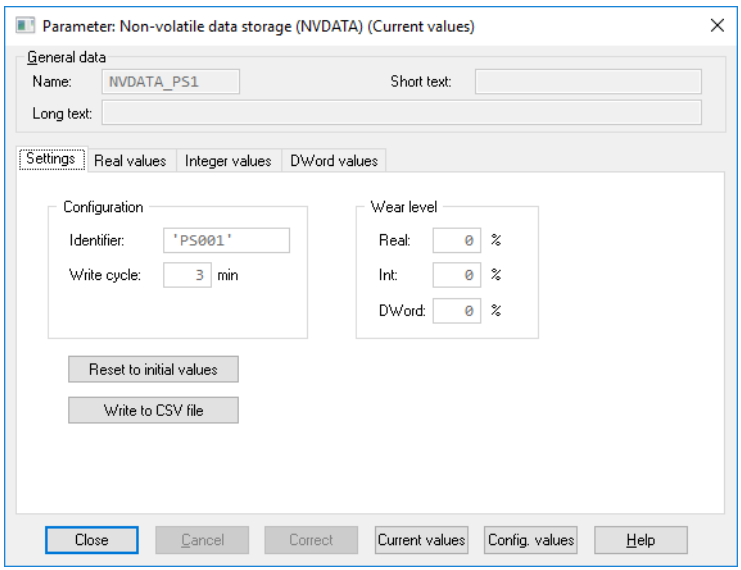
Initial data DWord_1 - DWord_6

The user can configure an initial value for each of these variables. These initial values are written to permanent memory when valid data is not available in the controller or in case of a manual reset.

The I/O Editor can be used to assign variable names to the NVDATA variables of the project (variable list). This makes it possible to write and read the values via variable names in the same way as with standard I/O.

Commissioning

Settings



NVDATA_Conf1_IBN_en.png

Reset to initial values

By clicking the button **Write**, all Real, Integer and DWord variables are reset to the configured initial values. (The button **Correct** has no function here)

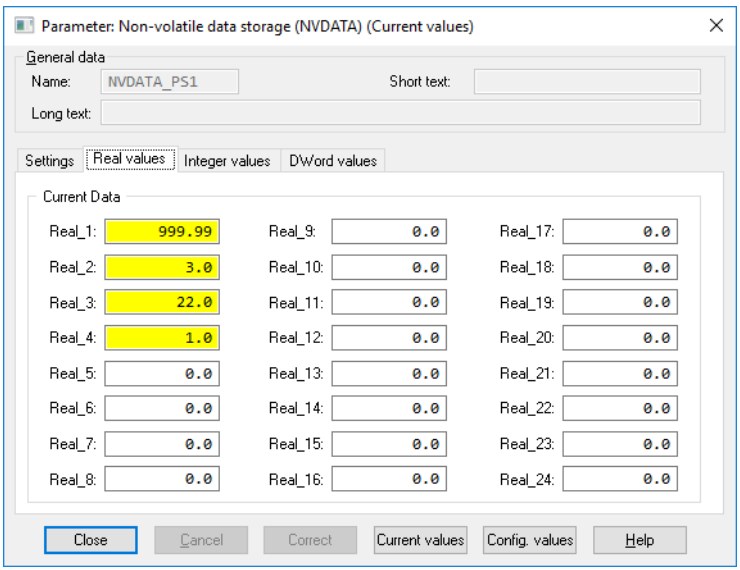
Write to CSV file

Writes the current variables to a CSV file.

The wear level of the permanent memory is displayed in the parameter screen. If one of the values reaches 100%, a replacement of the controller should be considered.

On redundant systems, the wear level is displayed for the primary controller only.

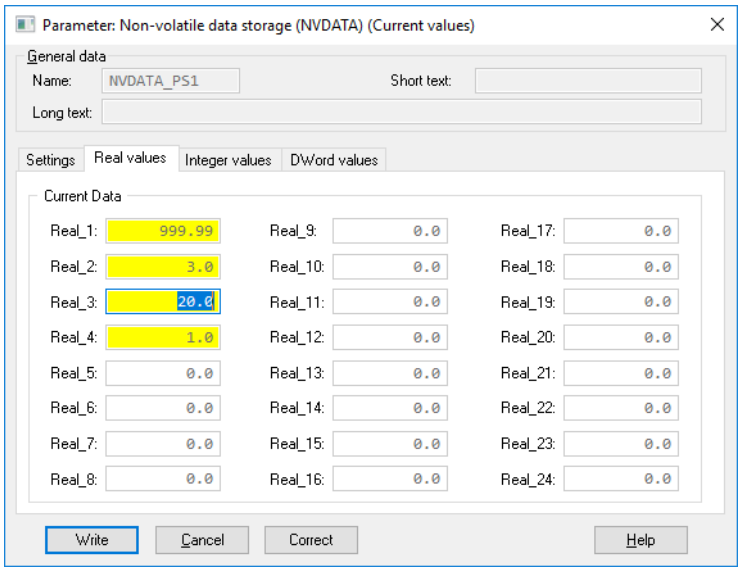
Real values, Integer values, DWord values *- current values*



NVDATA_Conf2_IBN_en.png

Current values When the parameter setting mask is called in commissioning mode, the current values are displayed. If these differ from the initial values, these are highlighted in yellow.

Config. values The initial values are displayed.



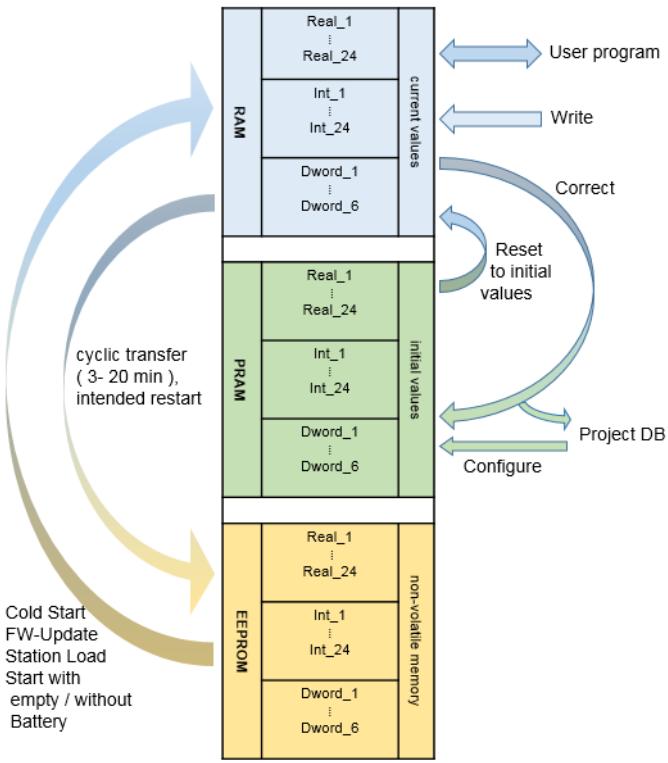
NVDATA_Conf2a_IBN_en.png

Write Changes the currently edited variables, but not the initial values.

Correct Changes the currently edited variables and the initial values in the controller and in the project database.

The current values can be copied to the initial values using the **load parameter** function and transferred to the project database. See also ***Engineering Manual System Configuration, Uploading the Current Block Parameters.***

3.10.3 Functional overview



Functional Overview_en.png

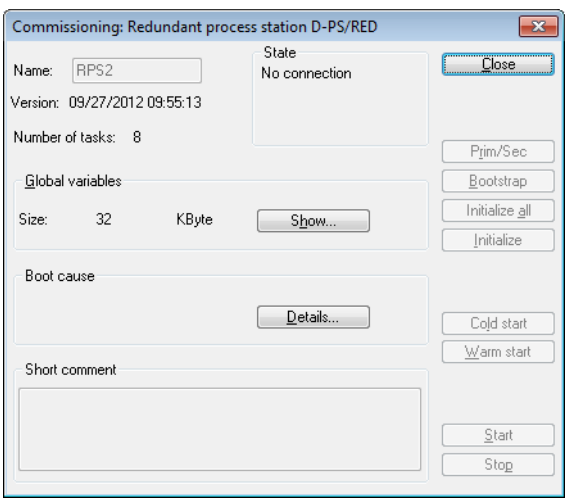
4 Commissioning

During commissioning, the configured programs and applications are loaded into the stations in order to be executed. Upon selection of a resource in the project tree, the process station can be loaded.

With a redundant station, one of both controllers automatically becomes the Primary and the other one the Secondary. It cannot be controlled which of both controllers is used as Primary.

A redundancy change-over can also be released by the “Force redundancy toggle” (PRIM_SEC) system function.

4.1 Resource in the project tree



Comm_ResPS_us.png

4.1.1 Resource status

A resource can assume various statuses that are displayed in the project tree after the objects and in the dialog mask of the resource. The display of resource status is divided into three parts:

- Status: *running, stopped, error*
- Sub-status, such as *warm start stopped* or *running partially*
- Status of the redundant resource, e.g. *no Secondary*

`No connection`

The system bus connection has been interrupted, **initialize all** is currently being executed or no connection is possible, e.g. due to an invalid network address or a wrong station number.

An interruption of the system bus is displayed after approx. 45 seconds.

`No operating system`

Displayed when the system is switched on or after **initialize all** of resource.
No operating system means that the operating system is loaded but not yet executed. The operating system is downloaded during the firmware bootstrap.

The AC 900F, AC 800F and AC 700F controllers file the operating system in the flash memory, it cannot be deleted but only be overwritten by another version. For redundant operation, however, both redundancy partners must feature the same operating system to ensure that the **Sync** status is provided.

`Stopping` Resource in transition from **Running** to **stopped.**

This status is only displayed if task processing takes a relatively long period of time.

`Stopped [HW]`

Stop of the resource using the RUN/STOP switch on the controller.

`Stopped [SW]`

Resource **Stop** was selected in commissioning mode.

If the resource is **stopped** while a task is at a **breakpoint**, then this task switches to **not executable** status. The error task is **not** initiated in this case.

`Cold start stopped`

The command to cold start the resource was initiated by the person responsible for commissioning or by the "Force cold start" system function. Indicates that the CPU module self-test was OK, but the RUN/STOP switch was set to STOP or the resource was stopped by the person responsible for commissioning before the running status was achieved.

The cold start task, < Resource name>**. ColdS**t is executed only when the resource is started (STOP to RUN). During cold start, all resource data are initialized, the operating system is not affected and the output module channels assume their safety values.

`Warm start stopped`

The command to warm start the resource was initiated by the person responsible for commissioning or a power failure (power-fail signal from the power supply unit). The RUN/STOP switch on the controller was set to STOP or the resource was stopped by the person responsible for commissioning before the running status was achieved.

The warm start task, < Resource name>. **WarmSt** is executed only when the resource is started (STOP to RUN). During warm start, all data and variable values of the resource remain unaffected and the output module channels assume their safety values.

`Starting` Resource in transition from **stopped** to **running**

This status is only displayed if task processing takes a relatively long period of time.

`Running` The resource is started and the Run/Stop switch is set to RUN.

`Running partially`
Not all objects of the resource are working e.g. task is in Stop status or program list is OFF.

`Version error, running`
There are user program version disparities between the active station and the assigned resource. Such disparities are displayed independently of the run status of the resource and are therefore also displayed when stopped.

`[del2]` Resource objects have been deleted, (2 objects in this case).

`Wrong station number`
The station number in the process station does not correspond to the station number in the network configuration.

`Incompatible DMS version`
The firmware or operating system version of the controller does not correspond to the current Freelance software.

Firmware and operating system must be updated, see Firmware update on page 23.

4.1.2 Additional statuses of a redundant resource

`non-redundant`
The operating system is loaded but the user program is not yet loaded.

`no secondary` Operating system and user programs are completely loaded. The Secondary has not yet logged on at the Primary.

`not sync` The Secondary controller is present, the operating system and the user programs have been loaded but the processes have not yet been updated and the synchronization is therefore not yet concluded.

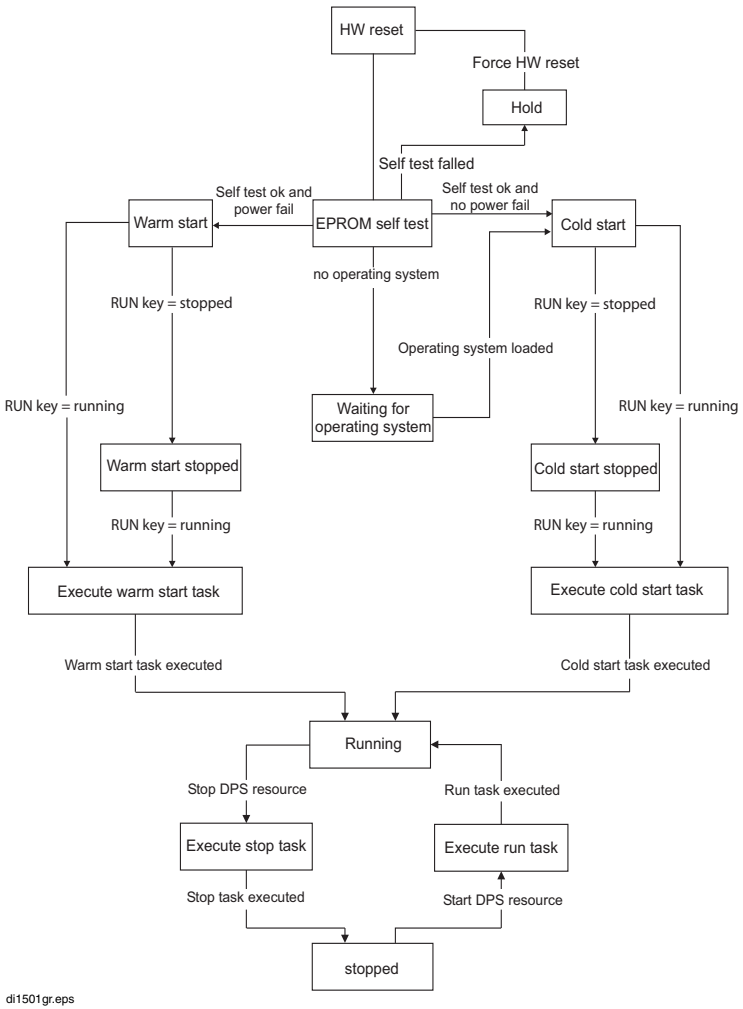
`sync` Normal status of a redundant process station. The Secondary is synchronized and ready for a redundancy toggle. The cyclic data adjustment between the two controllers runs during the program processing.

`redundancy error`
: Redundancy not possible.

The “sync” status cannot be achieved. By stopping and then restarting the resource, it was attempted several times to reach the “sync” status. This may be attributable to the following causes:
* the loading of the user program could not be successfully completed;
* the system utilization is too high, the redundancy link cannot be maintained;
* the configured task cycle times are too short.

4.1.3 Resource status diagram

For information on the transition statuses of redundant controllers, see Chapter 5, Controller redundancy.



4.1.4 Resource processing

Prim/Sec For redundant resources only.
A redundancy toggle is initiated (change-over between Primary and Secondary). A change-over is possible in SYNC status only.

Bootstrap The controller is loaded with the operating system and is initialized. With redundant configurations, only the Primary is loaded with the operating system.

Operation of a redundant process station is only possible on condition that both controllers are provided with the same operating system version.

Using the **Settings** tool for loading the operating system ensures that both controllers contain the same software.

Initialize all The controller is deleted. Initialize all means that the user program is deleted; the operating system is not executed. In case of redundant process stations, both controllers are reset. The Primary becomes active first and can be bootstrapped again. After **Initialize all**, the redundant mode is no longer active. The operating system in the Secondary is restarted after the user program has been loaded.

Initialize During initialization, the controller performs a cold start. Any user program already present is also deleted. The redundancy is now no longer **active**. The Secondary changes to “No OS” (no operating system is executed). The cold start task is performed once. The task is performed during transition from *cold start* or *cold start stopped* to *running* status. During a cold start, all the variables and working data of the function blocks are initialized. The initial values of the variables can be preset individually for each variable in the Variables List. After the cold start task has been completed, the outputs are rewritten in accordance with the programs.

Cold start The controller performs a cold start; with redundant stations, both controllers perform a cold start. The previously active controller becomes again the Primary. The cold start task is performed once. It is computed before the cyclic user tasks are started.

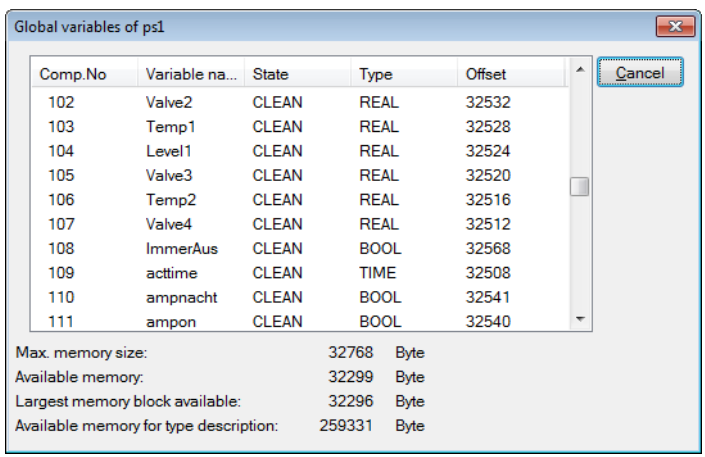
Warm start The controller performs a warm start; with redundant stations, both controllers perform a warm start. The previously active controller becomes again the Primary. The warm start task is performed once. It is computed before the cyclic user tasks are started.

The systemvariable "xxxPowerOffTim" contains the duration of last power failure which led to a warm start. It is counted from the time the power failure occurred to the restarting of the operating system.
For AC 800F and DCP the number of warm starts is counted in a system variable (xxxx.NoPowerFail) that is reset to zero in the event of a cold start.

Start/Stop The controller performs the **Start** or **Stop** command; with redundant stations, both controllers perform the command. The LEDs or the display on the Primary indicate the current status. Upon redundancy toggle, the last status is retained, irrespective of the RUN/STOP switch position.

Display variables area

The size of the variables area is indicated in kByte. Details are called up by pressing the **Display...** button.



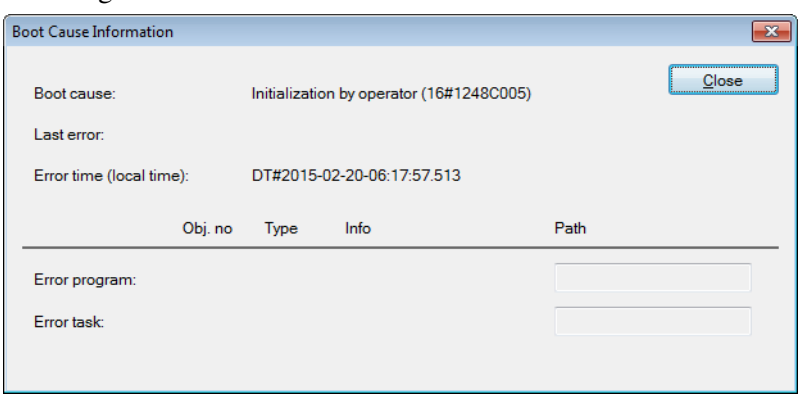
ShowVars_us.png

The memory allocation and the loading status of the global resource variables are indicated. Global variables are user-defined variables and system variables.

Comp. no.	Component number of the variable
Variable name	Name of the global variable
State	
CLEAN	Object is plausible and loaded.
DIRTY	Object version in the engineering station does not correspond to the object version in the process station.
CREATE	Object not yet loaded into the process station.
DELETE	Object deleted in project database, but still present in the process station.
TYPE	Data type, e.g. REAL, BOOL, UINT etc.
Offset	Offset of the memory address.

Boot cause

The cause of rebooting is displayed in short format. Rebooting may be caused by an operator intervention like “Boodstrap by the operator” or “Warm start by the reset button” or rebooting can be performed from the safety state (“Cold start from safety state”) or released by a “Software error” (fatal error). Detailed information on rebooting can be accessed via **Details…**:



RestartCause_us.png

Boot cause

Cause of the last restart of the resource. The code in hexadecimal format is indicated in brackets.

Last error In the event of a restart after an error has occurred, the error cause is displayed (e.g. INT/DINT overflow with arithmetic error and disabled automatic error handling).

Error time The time of a restart released by an operator intervention.
If the resource is restarted from safety state, the time when the safety state is actually reached is displayed.
If the resource is restarted after an error, the time when the error occurred is displayed.

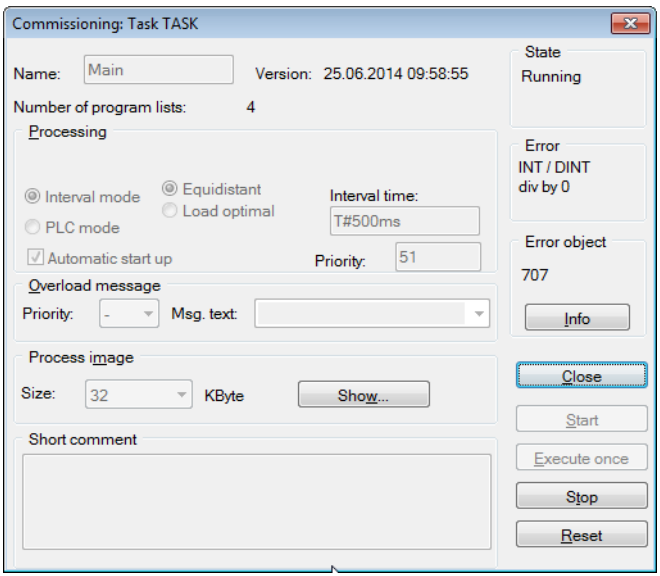
Error program

If the resource is restarted after an error, the error-producing program or function block is indicated. In this case, the object number, type, information and the path (position in the project tree) are displayed. The default setting in the path field is the end node (program). Scroll down or use the tooltip to view the complete path name.

Error task If the resource is restarted after an error, the task containing the error-producing program or function block is indicated. In this case, the object number, type, information and the path (position in the project tree) are displayed. The default setting in the path field is the end node (task). Scroll down or use the tooltip to view the complete path name.

4.1.5 Task

Select **task** in the project tree > **Project** > **Header**



Comm_Task_us.png

For a description of the configured data, see ***Engineering Manual, System configuration, Project tree, Task TASK*** and ***Redundant Task TASK/RED***.

If configured for autostart, the task starts automatically when the resource is started. When the program list parameter is set to ON, all lower-level program lists and programs are started together with the task.

Task status display

`Ready` The task has been loaded and can be started manually.

If no **autostart** has been configured for a task, it returns to the **ready** status after each cold start and has to be restarted.

`Non-existent`
Task is loaded but cannot be started. A software version error exists.

`Not loaded`

A change to a task or a program within the task, which produces a side effect on the task, has not yet been loaded.

`Not executable`

During execution of the task either an unrecoverable error occurred or a recoverable error was detected while automatic error correction was switched off.

Automatic error correction for a task is switched on in the default setting. If automatic error correction is switched off, the task changes to the **not executable** status on any error, even if the error is recoverable. The error appears in text form in the task header, and the object number of the faulty project object is also displayed.

An exception is constituted by tasks which, although loadable, have no lower-level user program. Such a task is unable to initiate execution of a program. If such a task is started despite this fact, it changes to the not executable status.

`Running` After manual **start** of the task or when the task has been configured with **Autostart**.

When a higher-level resource is stopped and re-started, the lower-level task returns to its old status. A task with **Autostar**t, changes automatically to **running**. The status of the task may also be changed when the resource is stopped. The changed status will be reassumed when the resource is restarted.

`Stopped` After execution of **Stop** task, **Stop** station (HW) or **Stop** higher-level resource (SW).

If a stopped task is reset to the ready status (not visible) with **Reset**, no change to its **stopped** status is displayed to the user. When Stop is changed to Start, the RUN task is executed.

A task is started automatically when the switch on the process station is set to **RUN** and the previous status of the task was **running**.

`Halted` The task is at a break point while the debugger is running. This status can be quit using **Single step** or **Go** in the program with the breakpoint.

If the resource is **stopped** while a task is at a **breakpoint**, then this task switches to **non-executable** status. The error task is **not** initiated in this case.

Error message of the tasks

ErrorNo	Error text	**Description**
1	INT/DINT overflow	Signed integer or time operation caused overflow.
2	INT/DINT underflow	Signed integer or time operation caused underflow.
3	INT/DINT div by 0	Signed integer division by zero.
4	UINT/DINT div by 0	Unsigned integer division by zero
5	INT/DINT St overflow	Signed 16-bit-integer operation caused overflow on storage.
6	REAL overflow	Real operation caused overflow.
7	REAL underflow	Real operation caused underflow.
8	REAL div by 0	Real division by zero.
9	REAL undef. value	Real operation with undefined value
10	DT overflow	Operation with DT and TIME caused overflow.
11	DT underflow	Operation with DT and TIME or with DT and DT caused underflow
12	UINT/UDINT overflow	Unsigned integer operation caused overflow.
13	UINT/UDINT underflow	Unsigned integer operation caused underflow.

ErrorNo	Error text	Description
128	Program exec. error	Unloaded object (e.g. program, FB, etc.) to be executed.
129	FB error integer	A function block from the function block library has triggered an integer overflow or underflow, the error was only detected during further processing of the block. For this reason, no recovery is possible.
130	FB error real	A function block from the function block library has triggered a REAL overflow or underflow, the error was only detected during further processing of the block. For this reason, no recovery is possible.
131	Process image read error	Unrecoverable error occurred on reading process image.
132	Process image write error	Unrecoverable error occurred on writing process image
133	Execution abort	If the resource is stopped in a program in which an endless loop is running, the task concerned switches to not executable status. Execution of the task is terminated in the endless loop and the program is not computed to the end.
134	Invalid Cmd in Break	If the resource is stopped while a task is at a breakpoint, then this task switches to not executable status. The error task is not initiated in this case.
135	Illegal array index	The computed index for an array access is outside the defined array boundaries.

Task error object

If an error occurs in processing the task, the number of the object that caused the error is displayed. The path to the object giving rise to the error can be displayed via **Info**.

For details on the task behavior in the event of an error, see Chapter 6, Processing and error handling.

Task processing

Start The selected task is started. It is processed either in cycle mode or as soon as possible (PLC mode), depending on the configuration.

Execute once The selected task is computed precisely once. The task then changes to the stopped status.

Stop The selected task is stopped.

A stop can also be released using the RUN/STOP switch on the process station.

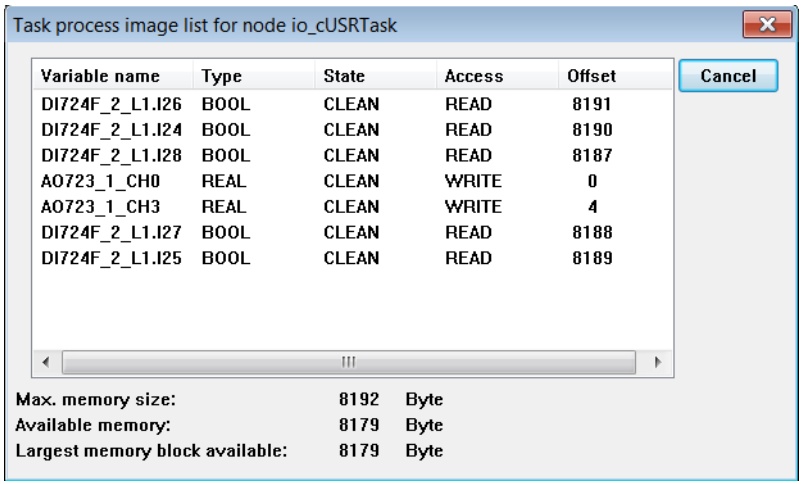
Reset The current task returns to the configured status.

Close OK exits the dialog box. Returns to the project tree.

Show... The process image of the selected task is displayed. The process image size can be configured. The process image is displayed automatically when the operating system is loaded.

Process image display

Select **task** > **Project** > **Header** > **Show**...



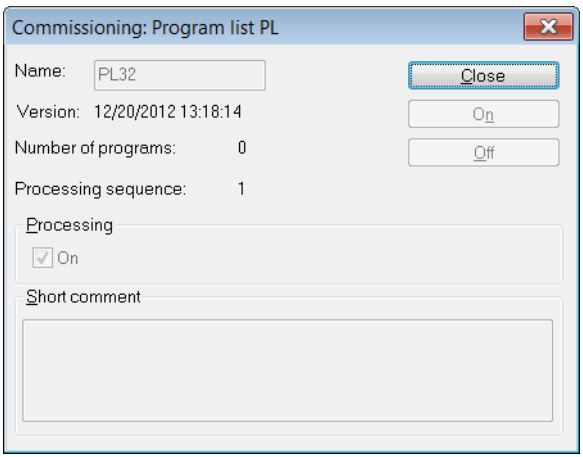
Process_image_us.png

Variable name		Variable name, max. 16 characters
Type		Data type, e.g. REAL, BOOL, UINT etc.
State	`CLEAN`	Object is correct and loaded.
	`DIRTY`	Object version in engineering station does not correspond to station version.
	`CREATE`	Object not yet loaded into station.
	`DELETE`	Object deleted in project database, but still present in station.
Access	`READ`	Read-only
	`WRITE`	Write-only
Offset		Memory address offset

See also ***Engineering Manual, System Configuration, Project tree.***

4.1.6 Program list (PL)

Select **program list > Project > Header**



Comm_Prog_list_us.png

Program list status display

`versions error`
: Program list was loaded but the versions do not match.

`not loaded`
: Program list is not yet loaded.

`on`
: Program list is switched on.

`off`
: Program list is switched off.

Processing of the program list

Close
: The dialog is terminated; back to the project tree.

On
: Processing of program list is switched on.

Off
: Processing of program list is switched off.

4.1.7 Status displays in the project tree

In contrast to configuration, the objects are shown with status information during commissioning. This status information appears after each project object in brackets (like the object types) and is updated as it changes. The status information reflects the status of this object in the process station. If arrows appear before the nodes, these project objects must first be loaded or reloaded into the station because of a configuration change. An exclamation point indicates that other objects at levels below the one so marked need to be updated because of changes in their configuration.

The color of the node in the display provides information about the effects of its configuration change on other objects. Higher-level information may also need to be updated in the station.

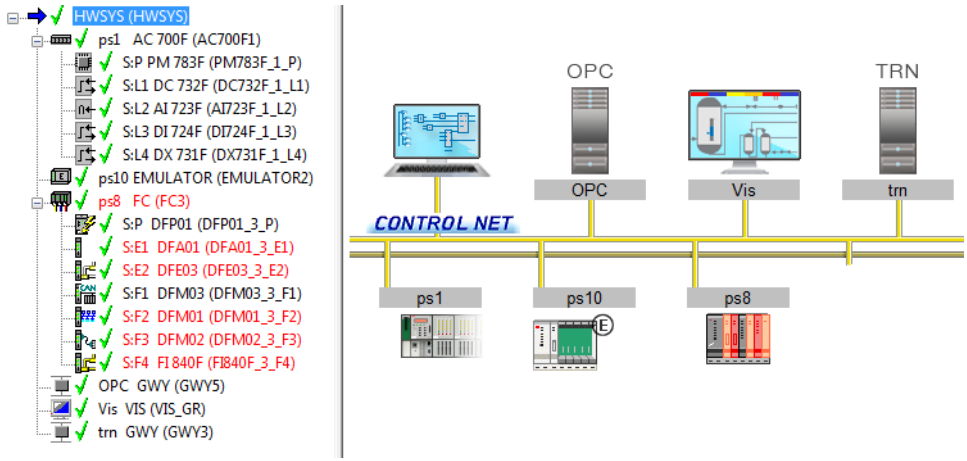
During commissioning, **nodes** are displayed in the same formats as in configuration. See ***Engineering Manual, System Configuration, Project tree, Display of the project element status.***

All statuses except for **Running** are displayed in red.

4.2 Status displays in the hardware structure

After loading the resource in the station, the system or station knows which modules or field devices are configured in the station. As the components (modules, devices) may not have been installed yet or the position of the installed component may not correspond to the slot configured for this purpose, these errors are indicated by colored status information.

System > Hardware structure



HW_Error_Indication.pngg

Status in the tree view

In the tree view, status information is shown as follows: active (black), faulty (red) and inactive (gray). The color coding is applied to the object information (text) rather than the icon.

Module status in the system and station views

Depending on the current status, the modules of the sub-rack are displayed in different colors.

Color and status

transparent (rear view)
: There is no module configured in this slot, neither is any module installed in the station.

gray
: The module is configured and the correct module type installed.

red
: A module is configured in this slot, but not present in the station.

yellow
: An installed module was identified on this slot, but there is no configuration available.

orange
: The type configured in the hardware structure does not correspond to the installed module type.

Force status

green	A green rectangle is displayed at the top of the module (only for I/O modules). At least one channel on the module has been forced.

Status of the basic unit

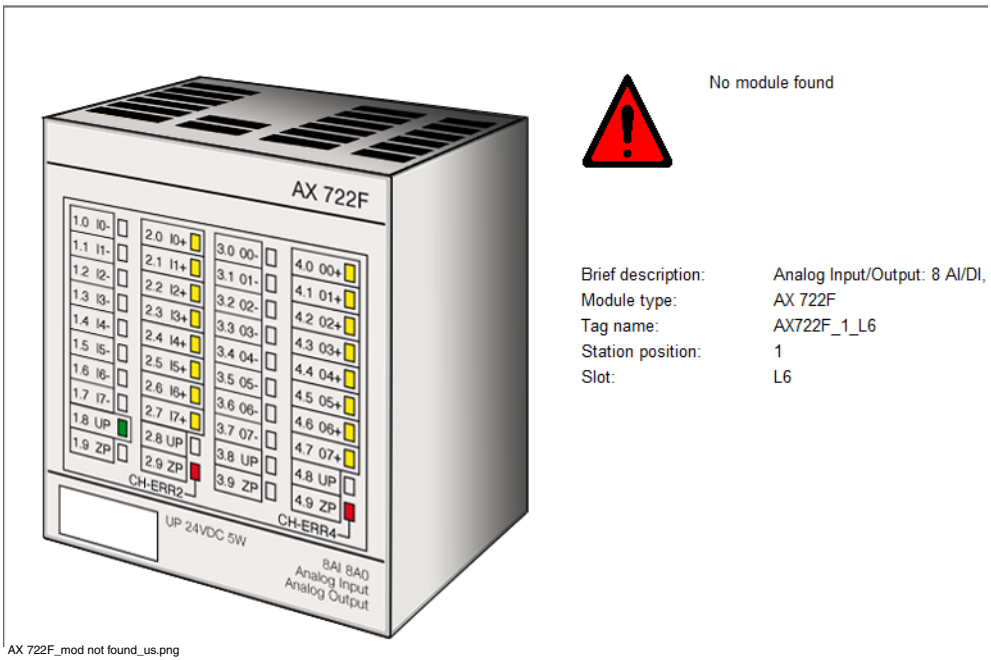
Upon insertion into the hardware structure, the principle process station structure is displayed. The color shows whether the unit is active, inactive or faulty. When a fault is indicated, this is generally due to the unit not being detected at all, i.e. not being installed or being impossible to address via the system bus.

dark gray	The station is activated
light gray	The station is not activated
red	The station cannot be addressed

Status of the modules

The **detail view** provides detailed information on the I/O modules of the process station to ensure easy identification.

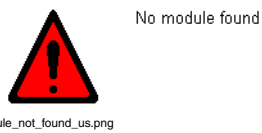
The status of modules is indicated by a colored triangle.



AX 722F_mod not found_us.png

No module found (red warning sign)

A module is configured in this slot, but not present in the station.



Module_not_found_us.png

Wrong module type (orange warning sign)

The type configured in the hardware structure does not correspond to the installed module type.

Wrong_module_us.png

Forced channels (green warning sign)

At least one channel of the module has been forced. See the ***Engineering Manual, Communications and Fieldbusses, Forcing and Substituting Values.***

Config fault (red exclamation mark)

The PROFIBUS-Master has detected a configuration fault. See ***Engineering Manual, Communications and Fieldbusses, PROFIBUS***.

Parameter fault (red exclamation mark)

The PROFIBUS-Master has a detected a parameter fault. See ***Engineering Manual, Communications and Fieldbusses, PROFIBUS.***

Diagnostic value detected (blue exclamation mark)

A diagnostic value has been recorded for the slave or for a module of the slave. The diagnostic value is shown in the list in the slave detail view, below the graphical representation. See ***Engineering Manual, Communications and Fieldbusses, PROFIBUS.***

4.3 Controller Webpage

The controller web page provides many information about the status of the process station, hardware and software configuration, diagnostics, memory statistics, task load, event and error messages. These information are primarily for service and diagnostic purposes, but also offers the user a variety of information about the state of the process station. Furthermore, the web display of the function blocks Boolean switch (DSPSW), Binary values (DSPBIN), Integer values (DSPINT) and Real values (DSPRE) can be called up via the controller web page (Application Displays).

The controller web page is opened by entering the controller IP address in a web browser.

To open the controller web pages the execution of Java-script must be (temporary) allowed.

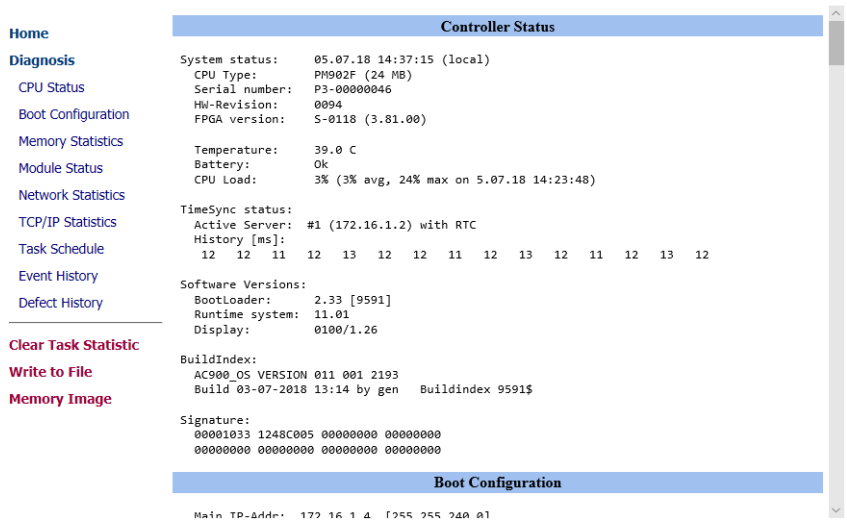
If display problems occur in Internet Explorer the option "Compatibility Mode" should be disabled in IE.

Startpage (Home):

CtrlWeb_Start.png

Via the menu on the left side the different information can be displayed.

Diagnosis:



CtrlWeb_Diag.png

The individual information can be displayed via vertical scrolling or via the menu on the left side.

Task Schedule Information

The Task Schedule Information section contains information that is useful for optimizing and analyzing CPU load.

Task Schedule Information

Task Id	Obj	Conf Time	Cycle Time avg	Cycle Time max	Cycle Time ovr	Execution Time min	Execution Time avg	Execution Time max	Redundancy Time min	Redundancy Time avg	Redundancy Time max
AT09	3458	500	540	554	0	30	33	35	29	32	34
AT10	3460	200	240	325	73	1	17	40	1	6	32
AT11	3462	180	180	194	0	125	128	144	30	32	38
AT12	3464	500	540	550	0	44	51	71	1	1	23

CtrlWeb_Task.png

Task ID, Obj	task and object ID
Conf Time	configured cycle time of the task
Cycle Time	real task cycle time

Execution Time
task runtime

Redundancy Time
duration of the redundancy adjustment

Display the task object ID:
> select task in tree view > **System** > **Show selected objects**

Hints of CPU load optimization

- The real cycle time of the task (Cycle Time) should not show major deviations from the configured cycle time (Conf Time). If the configured cycle time can not be reached frequently, the reason for this is probably a high CPU load or an unfavorable ratio between cycle time and runtime.
- The runtime of a task should be much lower in relation to the cycle time.
- The duration of the redundancy time should be much lower in relation to the cycle time.

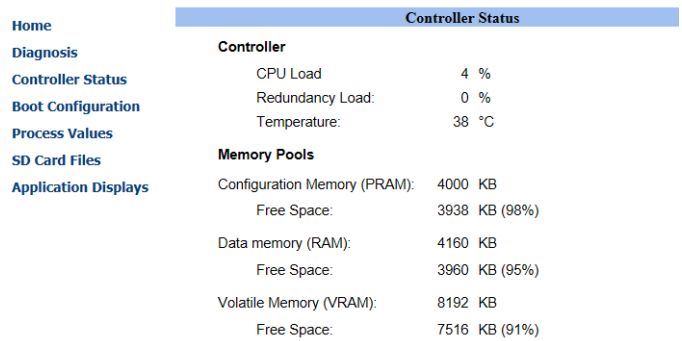
The displayed values depend on the configured application and can fluctuate significantly, in particular for non-cyclic functions (e.g. SFC).

The values stored under Task Schedule Information can be reset via the menu item **Clear Task Statistic**.

The menu item **Write to File** opens a new browser window in which the diagnostic functions are displayed as ASCII text.

The menu item **Memory Image** saves a memory image of the controller into a file (for service purposes only).

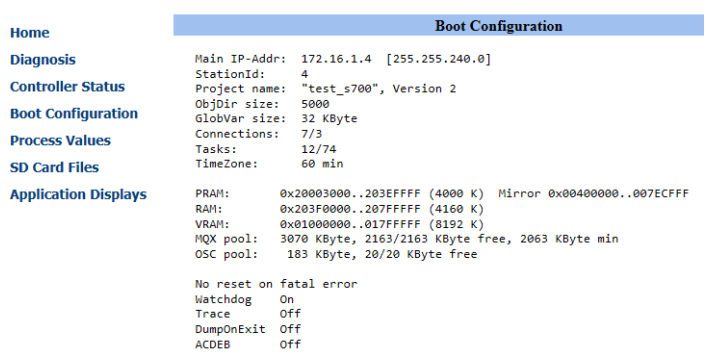
Controller Status:



CtrlWeb_CtrlState.png

The Controller Status page shows the current CPU load, the CPU temperature and the memory utilization.

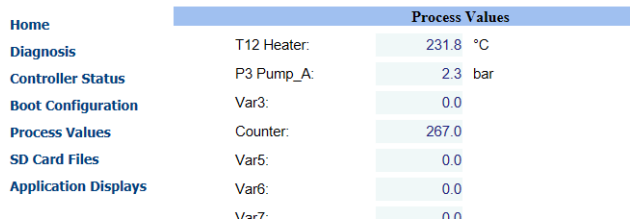
Boot Configuration



CtrlWeb_Diag.png

Shows the current boot configuration parameter.

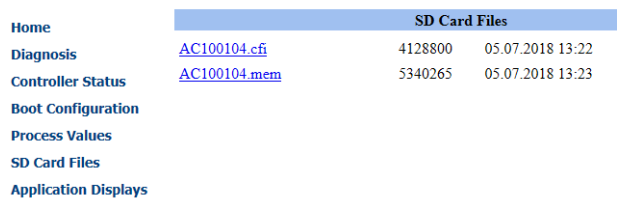
Process Value:



CtrlWeb_ProcVal.png

Displays the values to be configured via the Configure Display tab.

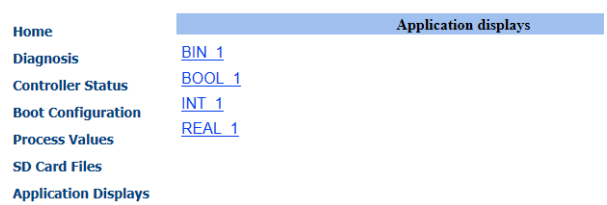
SD Card Files:



CtrlWeb_SDcard.png

Lists the SD card files.

Application displays:



CtrlWeb_AppDisp.png

The Application Display website can be used to call up the display modules configured for this controller.

5 Controller redundancy

5.1 Overview

The availability of the AC 900F and AC 800F process stations can be increased by operation in redundant mode, i.e. the entire process station is configured twice.

In case of redundancy, a bumpless toggle is achieved between the active controller (Primary) and the passive controller (Secondary), so that:

- in case of failure of the CPU module, PROFIBUS or CAN communication interface or 24 VDC power supply, the outputs of I/O components at connected fieldbusses are maintained,
- the internal statuses (component data, variable values) are maintained,
- configuration and working data are automatically adjusted between the active and passive controller,

The list box for a redundant process station features only the modules that can be used in redundant configuration. These modules are marked with an R for the AC 800F, e.g. FI 830FR. The parameter definition dialogs of redundant modules slightly differ from those for modules that cannot be used in redundant configuration. As the hardware of modules that are capable of redundancy is identical with the hardware of modules that are not capable of redundancy, these modules are interchangeable.

The redundancy link between both controllers that is used for redundancy adjustment is automatically generated and need not be configured. For detailed information on the IP address setting for the exchange of redundancy data, see the corresponding ***Mounting and Installation Instructions***.

To distinguish between both controllers, they are marked with IP1 and IP2. A redundant process station is always configured with two IP addresses, with the IP address 1 being used to address the upper IP1 controller and the IP address 2 for the lower IP2 controller.

In a redundant network, the IP addresses 1 and 2 must be configured for the network lines A and B.

The assignment of the IP address 1 or 2 does not define which controller acts as Primary or Secondary. With online connection, either the Primary or the Secondary may be in the upper position. See Network configuration in the hardware structure on page 38.

5.1.1 Synchronization

The basic behavior of all redundant process stations is identical. The terms Primary and Secondary in the following text refer to the active and/or passive controller.

The active Primary automatically ensures that the passive Secondary can take over in the event of redundancy. The adjustment of user programs and redundancy data required in the event of serious malfunctions is performed via a redundancy connection, the redundancy link.

For detailed information on installation, see the corresponding ***Mounting and Installation Instructions***.

When the process station is bootstrapped, one of the two controllers automatically becomes the Primary. The second controller recognizes a Primary on the station bus and reports to the Primary as Secondary.

A newly plugged-in controller recognizes automatically that another controller is already active on the station bus and adopts automatically the passive status.

A connection based on the redundancy link is created during the adjustment between active and passive controller. The user program is transferred from the Primary to the Secondary.

The Secondary performs a cold start with these data and then logs on at the Primary. The current process data are then transmitted to the Secondary. After successful transmission, both controllers change to “sync” status. From this moment on, the

Secondary is ready for a bumpless redundancy toggle. The relevant process data are adjusted cyclically.

The synchronization is performed simultaneously with the processing of the user programs in the controller.

Depending on the boot priority of the Secondary, side-effects can be avoided when synchronizing tasks. The higher the boot priority of the Secondary as compared to the processing priority of the task (51 – 99), the faster the controller reaches the **Sync** status. For detailed information, see the ***Engineering Manual, System Configuration.***

If a different module equipment has been detected in both AC 900F controllers, the controller cannot reach the **sync** status.

5.1.2 Redundancy toggle

If any of the following conditions occur at the Primary, a redundancy toggle is initiated **deliberately** between Primary and Secondary:

- the Prim/Sec **TOGGLE** switch of the controller is actuated,
- the Prim/Sec button of the resource is actuated in commissioning mode,
- the **RESET** switch is actuated > 5 s, resulting in a cold start of the Primary,
- the system function “Force redundancy toggle” (PRIM_SEC) is executed.

In typical applications, the Secondary will continue the application processing after 350 to 400 ms when a redundancy toggle is initiated deliberately. In the event of an error, the actual toggle time will generally be longer, as in addition to the time for change-over required by the controller, the error detection and the change of the connected field busses will take time.

5.1.3 Redundancy criteria

A redundancy toggle between Primary and Secondary **is forced** when the self-test diagnosis reveals the following conditions:

- a module fault has occurred (a module fault is also detected, when a fieldbus module is switched off),
- an equipment fault has been detected on the fieldbus module,

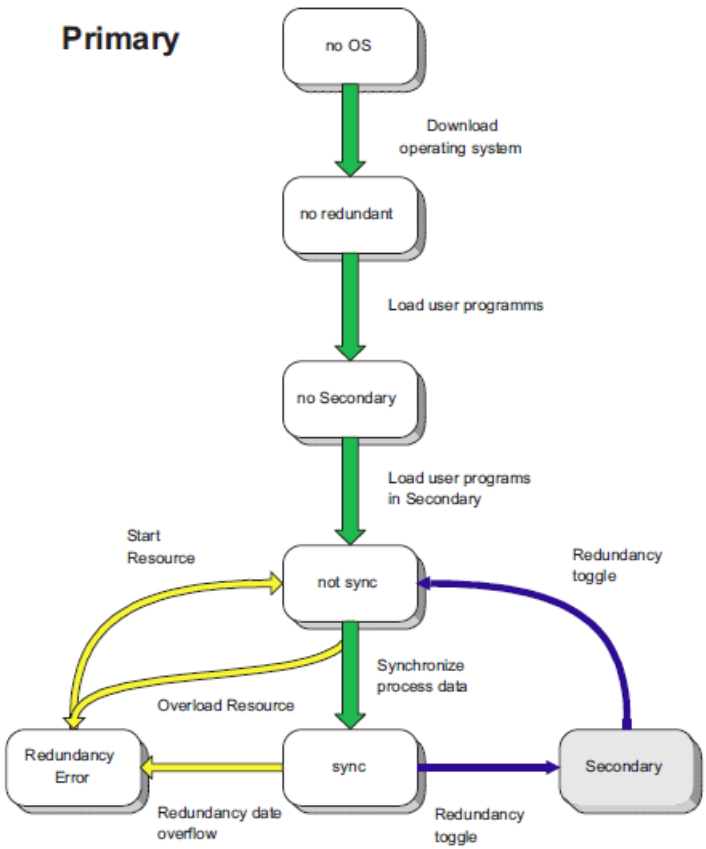
- the watchdog has responded due to a controller hardware error,
- a power fail of the power supply is reported,
- a network fault has occurred,
- a “fatal software error” has occurred.

If the redundant link is interrupted, a redundancy toggle is initiated. The previous Primary remains in the “no operating system“ status. When the redundancy link is re-established, a reset must be initiated in the Secondary.

If the Ethernet interface for the redundancy link on the Secondary fails, no redundancy toggle is performed. When the Ethernet interface for the redundancy link on the Secondary operates again, a reset must be initiated in the Secondary.

5.1.4 Redundancy statuses

Status diagram of the Primary



Redundancy states_us.png

no OS	No active operating system
no redundant	The operating system is loaded but the user program is not yet loaded.
no Secondary	Operating system and user programs are completely loaded. The Secondary has not yet logged on at the Primary.

not sync
: The Secondary controller is present, the operating system and the user program have been loaded, but the process statuses have not yet been adjusted and the synchronization is therefore not yet concluded.

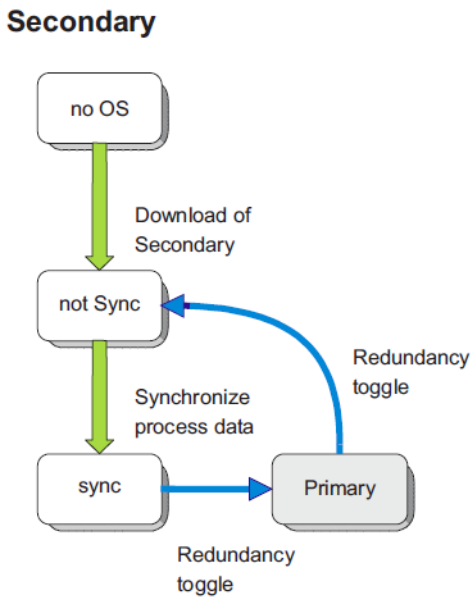
sync
: Normal status of a redundant process station. The Secondary is synchronized and ready for a redundancy toggle. The cyclic data adjustment between the two redundancy partners is performed during program processing.

Redundancy Error
: Redundancy not possible. The controller cannot achieve “sync” status. By stopping and then restarting the resource, it was attempted several times to reach the “sync” status.

Possible causes of a redundancy error:

- loading of the user program could not be successfully completed;
- the system utilization is too high, the redundancy link cannot be maintained;
- the configured task cycle times are too short;
- both controllers are equipped with different modules;
- a module reports a fault.

Status diagram of the Secondary



Redundancy toggle_us.png

no OS	Operating system not active. The Secondary controller is waiting for the transmission of the user program.
not Sync	The Secondary has logged in on the Primary, the user program has been loaded, but the process statuses have not yet been adjusted and the synchronization is not yet concluded.
sync	The Secondary is synchronized and ready for a redundancy toggle. The cyclic data adjustment between the two redundancy partners is performed during program processing.

The redundancy status of a resource can be evaluated using the system variable xxxx.RedState.

See Redundancy statuses on page 115 and ***Engineering Manual, IEC 61131-3 Programming, System variables.***

5.1.5 Technical redundancy data

Failure detection

Both redundancy partners (Primary and Secondary) monitor each other via the redundancy link. While the Secondary detects a failure of the Primary (redundancy toggle) within 40 ms, the Primary detects a failure of the Secondary within 500 ms.

Take-over

In a typical application, the Secondary takes over from the Primary within approximately 1 second. With complex applications, high task cycle times, a large quantity of remote I/O modules, etc. a higher take-over time must be expected, due to the working principle.

5.2 Changing a non-redundant process station into a redundant one

1. The following exports must be performed from each process station which is intended to operate in redundant mode
   a) the block of the resource process station in the project tree and
   b) the block of all hardware objects of the process station in the hardware structure.

Block import of sub-projects enables a resource to be automatically made redundant via the menu item **Edit > Import block as redundant**. This means that the project elements are filed and all variables are written in process image mode. With the export of individual process stations, the module equipment and the I/O channel assignment - which would otherwise be lost - are taken over on re-import.

Special advice for AC 900F controllers:

A redundant AC 900F controller can only be used in combination with CI 930F and CI 773F PROFIBUS modules respectively CI 910F CAN module; directly connected I/O modules and CM 772F PROFIBUS modules cannot be operated in redundant mode. This is why these modules cannot be imported into a redundant AC 900FR controller.

2. The variables of the task to be implemented redundantly must be written in process image mode, otherwise they will not be redundancy-capable. Please check your user programs for variables which are not written in the process image (column P in the variables list). If you consider that these variables must not be written in process image mode because the information within the task must be immediately available and not only be updated at the end of a task cycle, you should directly renew the resource allocation for the variables or relocate the program lists and sequential function chart individually (see also next item).
3. Delete the resources intended for redundant operation. Otherwise, the redundant resources are automatically renamed on import into the project pool, since the objects with the same name are still located in the project tree. Alternatively, you can create a new redundant resource, insert redundant tasks and relocate the program lists and sequence programs individually. Do not forget the contents of the system tasks. This method should be chosen if all the variables cannot be written in process image mode or if not all the tasks have to operate redundantly.
4. In the project tree and using **Edit > Import block as redundant**, insert the resources which were previously exported as redundant resources into the project pool.
5. Renew the resources assignment for the variables. This can be done manually or automatically.
6. The global system variables for the previously non-redundant resources, which have been used in programs or pictures, still have the previous resource allocation in the Variables List. The resource allocations must be changed to a new resource name, if any.

These now undefined Identifiers are reported as errors in the plausibility check.

7. In the hardware structure, the individually exported blocks of the process stations must now be imported into a redundant process station, so that all information on insertion and I/O components are available again.

When converting a rack-based process station into a redundant AC 900FR/AC 800FR controller, the I/O modules must be separately imported as a block below the CAN master.

8. In the hardware structure, the redundant project tree resource must then be assigned to the redundant process stations.
9. Plausibility check the project.

6 Processing and error handling

This section provides information on error handling at task level, on the resource behavior in the event of errors, restoring a connection and on the behavior of the modules at power failure.

6.1 Automatic error handling at task level

Error handling at task level is described in the flow chart, see Task behavior in the event of an error on page 123.

If an error is detected in a user program, the error task is executed once. The error task has the highest priority (priority 100) in a resource and is used for handling errors in user programs by user programs. If an unrecoverable error is detected, the error-producing task changes over to the not executable status. If the error is recoverable, the error-producing task can remain in running status, provided the automatic error correction has been enabled for the resource. Due to its high priority, it cannot be interrupted by other tasks. However, the execution of the error task can be deactivated.

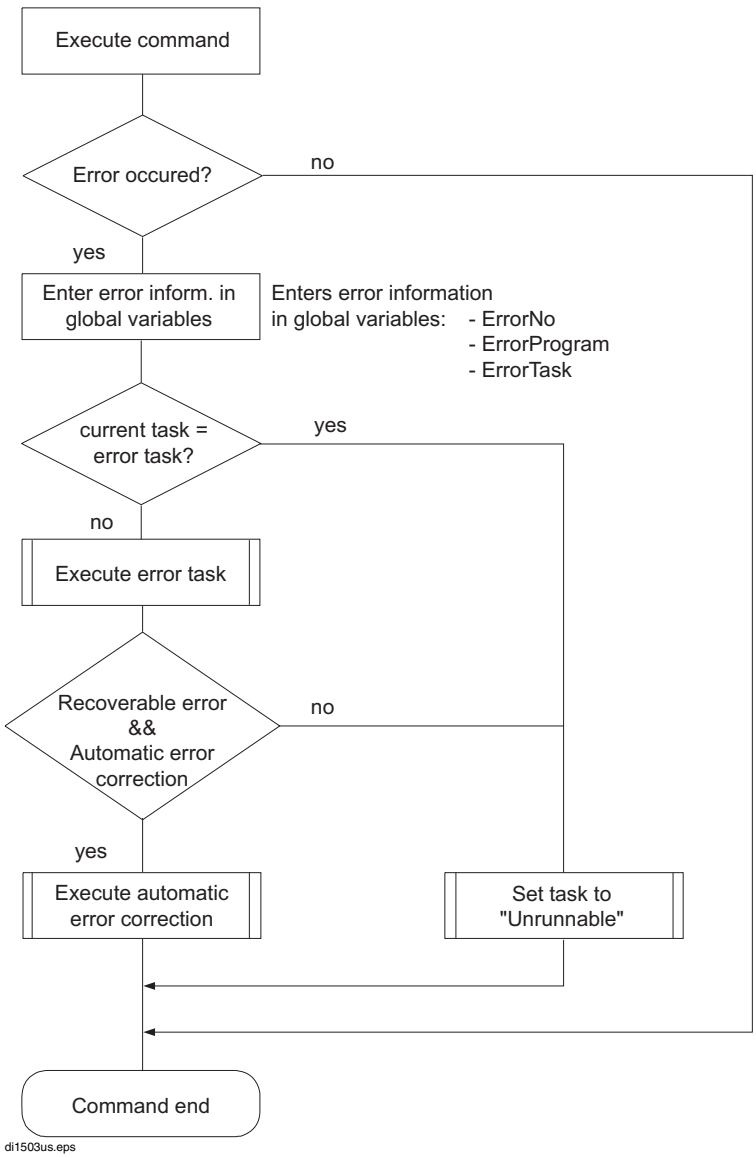
Error handling of the task is activated by default, but can be deactivated in the resource configuration (see ***Engineering Manual, System Configuration, Project tree)***. When error handling is disabled, an error causes the task to change to the not executable status, even if the error is recoverable. The error appears in text form in the task header, and the object number of the faulty project object is also displayed. If automatic error handling is active and a recoverable error is detected, the affected task remains in the running status. In this case, the cause of the error can only be identified by evaluation of the system variables.

Each runtime error detected during execution of a command results in an entry recording the cause and location of the error in system variables specially defined for error handling. The cause of the error, e.g. 4 for UNIT Div by 0, is saved in the

variable "ErrorNo". The variable "ErrorProgra" is written with the object number of the program or function block that caused the error. In addition, the object number of the affected task is saved in the "ErrorTask" variable, but only displayed in the task header if error handling is deactivated. The affected object can then be localized in the object list with its name, status, type and position in the project tree. If the task, that triggered the error, is not the error-handling task, the error handling task itself is now triggered. Error handling specific to the cause of the error and/or its location can be undertaken in this task, with the aid of the system variables. If an error occurs in the error-handling task itself, the error-handling task is marked as **not executable**.

See the ***Engineering Manual IEC 61131-3 Programming, Variables*** for more detailed information.

6.2 Task behavior in the event of an error



6.2.1 Resource behavior in the event of an unrecoverable task error

With the task behavior in the event of an error described above, priority is placed on station availability as only the error-producing task changes over to the not executable status in the case of unrecoverable errors (or with recoverable errors and deactivated error handling). The output values of the programs processed by this task maintain the last value. All other tasks that are not affected by the error continue program processing and update the outputs assigned to them.

If additional error handling is necessary to ensure proper functioning of the plant and the use of the error task will not be sufficient for this purpose, the behavior of the complete station can be configured for this type of error. See the ***Engineering Manual, System configuration, Project tree, Configuration of the project elements, Process station***.

With strong interdependencies between the programs of the individual tasks, for example, continuous processing of non-affected tasks could lead to the calculation of inconsistent values that are transmitted via the remaining station output channels to the process.

In these cases, the complete station can be set to “Safety state” instead of handling only the error-producing task as an isolated problem.

Stop of the station in safety state

In “Safety state”, the controller stops program processing and the outputs of the I/O modules assume the configured safety values (this applies to all I/O and fieldbus modules). A “Fatal Error on Task...” entry in the controller logfile indicates such an error.

Depending on the process requirements, the safety values can be configured to “Hold the last value” or to a particular value/status. The output channels of field devices feature manufacturer-specific safety values.

If the change to safety state is configured, it can be defined whether the station remains in this status or is automatically rebooted after having been reset. (During process station configuration, select *Stop in safety state* or *R*eset *resource* in the project tree.)

It may be necessary to maintain the safety state in order to perform additional measures aimed at ensuring the proper functioning of the plant or for error analysis. In this case, the station must be reset manually.

If rebooting is configured, the station starts after 10 seconds and performs a cold start or an initialization.

Booting in safety state

Depending on the type of error, the station is rebooted by a cold start or initialized. With a station stopped in safety state, the different rebooting processes are performed in both modes, i.e. automatic booting and manual station reset.

Error types resulting in cold start rebooting:

- Arithmetic error with automatic error handling being disabled
- Program execution error (the valid field index range of an ST program is violated)
- Operator intervention, such as
  - a task is at the debugger break point, a stop command is sent to the station and the warning dialog is omitted.
  - a tasks program requires an extremely long computing time (or is in an “endless loop“), a stop command is sent to the station and the warning dialog is omitted.

These error conditions will not occur after a cold start; this is why the station is started with a cold start.

Error types resulting in rebooting with initialization:

- Operator interventions that lead to an undefined loading sequence, e.g. isolated loading of individual object
- Program structure error

Errors that affect the structure of the user program are not eliminated by a cold start and will occur again: this is why the station and thus the user program are

initialized. The initialization involves complete canceling of the user program. In this case, the user must reload the configuration.

A station in safety state is no longer connected by the system bus and cannot be accessed in Freelance Engineering.

The reboot procedure (cold start or initialization) is displayed at the station by the Error and Run/Stop status LEDs.

If not automatic reboot from safety state is configured, the station can only be reset manually.

Manual process station rebooting from safety state

To start the controller from safety state, proceed as follows:

AC 800F

Stop the controller using the Run/Stop switch. Press the Reset button or briefly switch off the supply voltage. The controller will perform a warm start before it adopts the Stop status. Load the revised user task and start the controller using the Run/Stop switch.

AC 700F/AC 900F

Switch off the controller. Actuate the Run switch and simultaneously switch on the supply voltage. The controller will perform a warm start before it adopts the Stop status. Load the revised user task and start the controller using the Run switch.

Tasks without user program

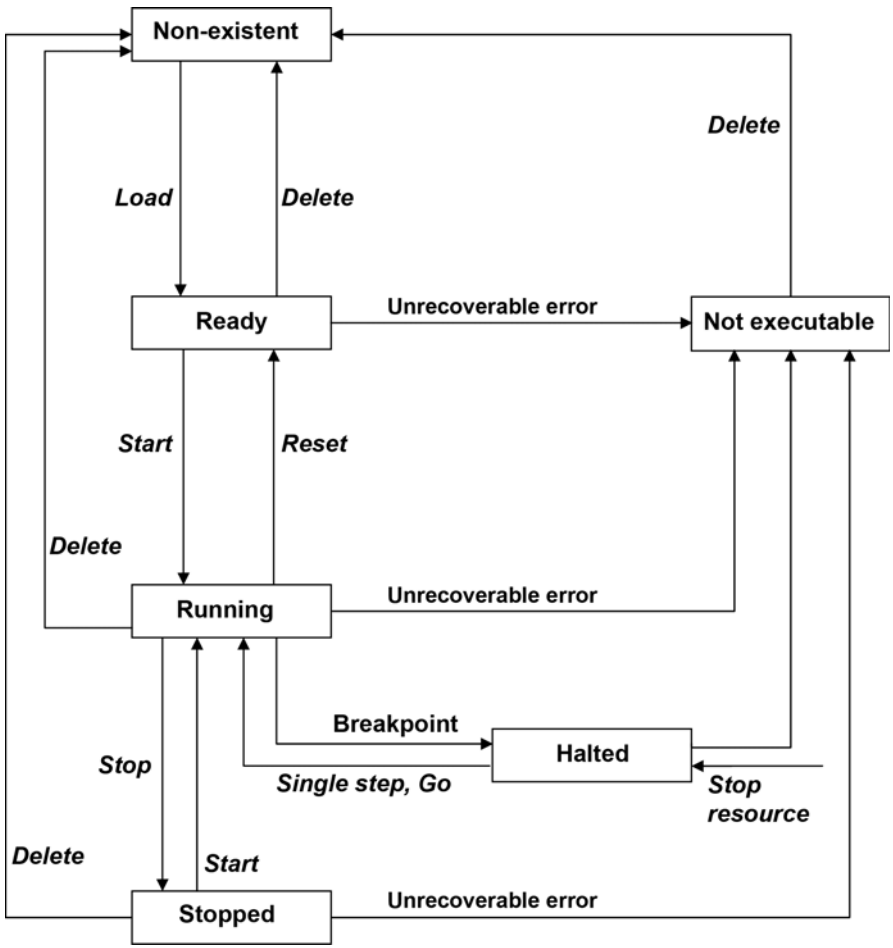
Tasks without lower-level user programs can be loaded. However, these tasks are unable to initiate execution of a program. If such a task is started despite this fact, it changes to the *running* status.

The task will not assume the *not executable* status, which implies that tasks without user program cannot cause the safety state.

6.2.2Status diagram of the task - safety state not configured

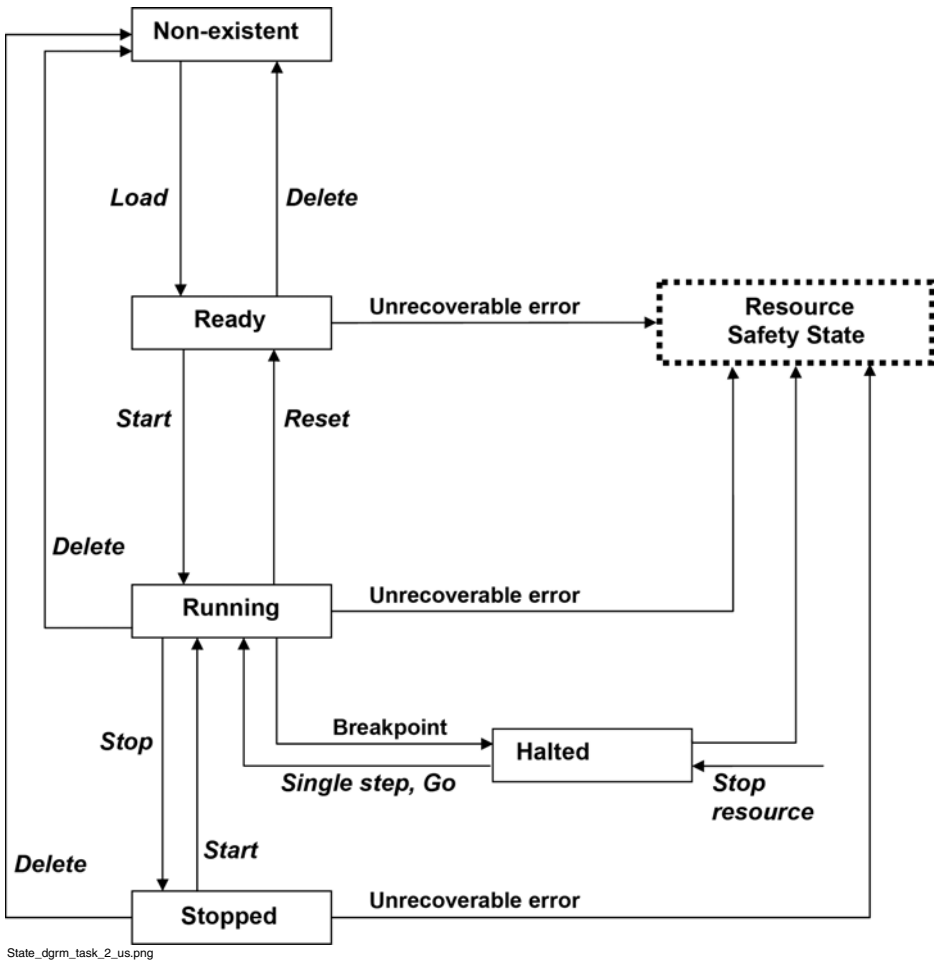
Status diagram of the task

Unrecoverable errors are errors in configuration, for example division by zero. In this case the task, and in particular the program, should be checked.

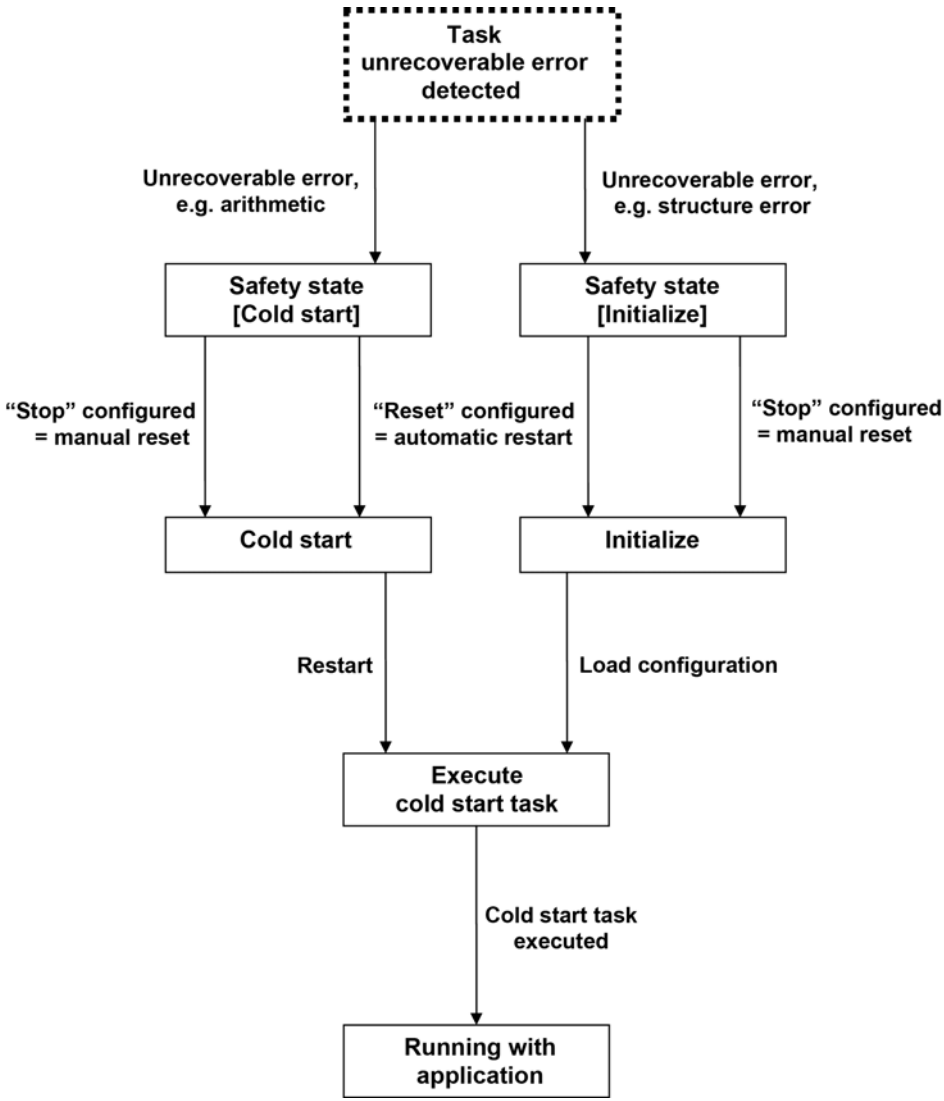


State_dgrm_task_1_us.png

6.2.3Status diagram of the task - safety state configured



6.2.4Resource status diagram - safety state configured



State_dgram_ressource_5_us.png

Redundancy behavior in the event of an unrecoverable task error

The redundancy mechanisms also adjust the task status changes on the Secondary. In the event of an unrecoverable task error (such as a non corrected arithmetic error or an array index error), the closely synchronized operation of the controllers will produce the same error on the Secondary controller. In the case of a structure error in a synchronized station (e.g. missing object due to undefined loading order), this error also affects the Secondary as the downloads are performed in synchronized mode.

This is why the safety state is initiated on the Secondary as well when unrecoverable tasks errors occur in redundant stations.

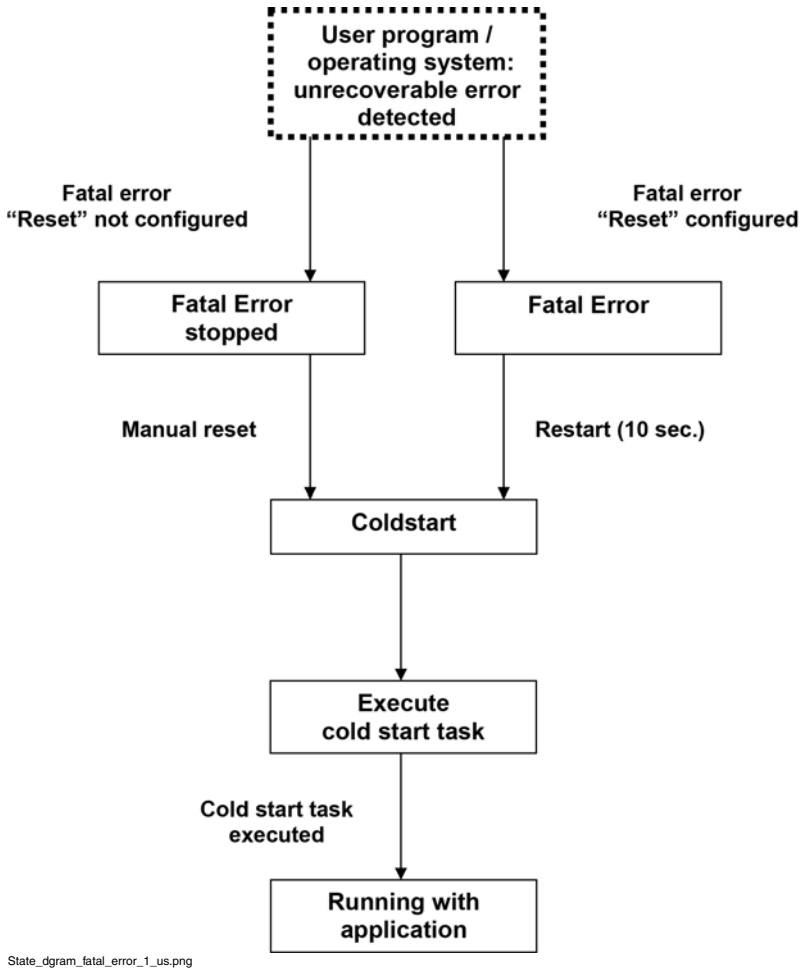
Resource behavior on fatal errors

If an internal error is detected in the sequence of the resource during user program processing (fatal error), the controller stops processing. With non-redundant systems, the outputs adopt the configured safety values and the error LED on the controller flashes.

Upon a fatal error, the station is no longer connected by the system bus and cannot be accessed in Freelance Engineering.

Using the *Reset on fatal error* entry in the resource configuration of the process station, it can be defined that the CPU quits such an error status after 10 seconds via a cold start.

Resource status diagram (fatal error)



State_dgram_fatal_error_1_us.png

The restart process from safety state (cold start or initialization) or a fatal error is indicated by the status display and/or the Error LED and the Run/Stop LED on the station.

Error information after restart

The error information shown in the header of the task and in the specific system variables defined for error handling upon an unrecoverable error is no longer displayed after the resource has been restarted.

This information is stored in the resource object of the station in a cold-start resistant manner. Upon a restart, the corresponding information is shown in the commissioning dialog.

6.3 Startup and shutdown performance of the modules

The input and output values are communicated on a cyclic basis between the I/O modules and CPU module of the process station. If no communication occurs for a configurable number of cycles, both sides register an interruption of the communication.

In this case, the process station transmits a system error message to the operator station. The output modules adopt the safety values. The input data for input modules are also set to the configured safety values on the controller. Depending on the process requirements, the safety values can be configured to "Hold the last value" or to a particular value/status.

A process station warm start or cold start always leads to output of the safety values.

For "Load whole station" and at resource Initialization/Overall reset, the output levels of the output modules go to zero current.

6.4 Power fail signal (PF) and power fail

I/O modules maintain the last values until they have recognized the connection interruption and then adopt safety values until the CPU module sends actual values after restart. In case of power failure of the I/O module, output levels adopt zero voltage and/or zero current.

After voltage returns, I/O modules only change their output level when the CPU module communicates actual values.

6.4.1 Power fail signal (PF) and no power fail

PF < 15 ms While the power fail signal is active, the CPU module or the AC 800F controller does not communicate current values; I/O modules independently maintain the last values.

PF > 15 ms I /O modules maintain the last values until they have recognized the connection interruption to the CPU module or to the AC 800F controller before they adopt the configured safety values The safety value are maintained until the AC 800F controller has performed a warm start and provides the I/O modules with new values.

6.4.2 Power fail signal (PF) and power fail

I/O modules maintain the last values until they have recognized the connection interruption and then adopt safety values until the CPU module or the AC 800F-Controller provides them with the current values upon restart. In case of power failure of the I/O module, output levels adopt zero voltage and/or zero current.

When power is supplied again, the I/O modules change their output level only when the CPU module or the AC 800F controller communicates current values. This does not apply to the analog output modules. As long as there is no failure in the external power supply, they keep the last values. Only when the power supply of the AC 800F controller is ensured again, the outputs are de-energized.

6.4.3 Safety values for field devices

The cyclic data exchange between the PROFIBUS master and the connected slaves is monitored. When the cyclic data exchange is interrupted, the safety values can be assumed. The output channels feature manufacturer-specific safety values. Depending on the process requirements, the safety values for input channels can be configured to "Hold the last value" or to a particular value/status.

A possible power failure can be evaluated by means of a system variable and used in the user program.

Appendix A Hardware object data

A.1 Calling up object data

When the **I/O editor** is selected in the main menu, the object data of the selected module can be viewed and configured. All inputs, output and diagnostic data are displayed with the name, data type and position in the parameter block (Byte = position of the first byte in the parameter block, Bit = position of the first bit in the start byte, Length = number of bits). A global variable can be assigned to each module component. For detailed information, see the ***Engineering Manual - System Configuration, Hardware Structure, I/O Editor.***

Key for the following tables:

Conf.	*y*es = value can be configured by the user *no* = value is determined by the system.
Acc.	possible access to the user, e.g. via an OPC connection *RO* = Read Only *RW* = Read Write

Diagnostic data can only be read and not be configured.

A.2 Object data: AC 900F and AC 900FR

There are no differences between the object data of the AC 900F controller in the Plus, Standard and Lite versions; i.e. the AC 900F, AC 900FP and AC 900FL data are identical. This also applies to the AC 900FR, AC 900 FR P and AC 900FR L data.

A.2.1 Parameters - AC 900F and AC 900FR

Name AC 900F	Name AC 900FR	Data type	Conf.	Acc.	Comment
SnrNo	SnrNoI, SnrNoII	STR16	no	RO	Serial manufacturing number
HrvNo	HrvNoI, HrvNoII	STR8	no	RO	Module hardware version
SrvNO	SrvNoI, SrvNoI	STR8	no	RO	Module software version
Dur	DurI, DurII	DINT	no	RO	Total number of operating hours
DurHot	DurHotI, DurHotII	DINT	no	RO	Total number of operating hours with high temperature
Temp	TempI, TempII	DINT	no	RO	CPU module temperature
ModName	ModNameI, ModNameII	STR16	no	RO	CPU module type name
PwrFailLev	PwrFailLev	BOOL	yes	RW	TRUE: Powerfail GL-Level must be used
IsRed	IsRed	WORD	no	RO	Internal parameter redundancy
-	RSStateI, RSStateII	INT	no	RO	Run/Stop status IP1 or IP2
-	LocRSStateI, LocRSStateII	INT	no	RO	Internal Run/Stop status IP1 or IP2 parameter
-	RedStateI, RedStateII	INT	no	RO	IP1 or IP2 redundancy status
-	SBootReaOldI, SBotReaOldII	WORD	no	RO	Boot cause of the Secondary memorized, no standard adjustment

Name AC 900F	Name AC 900FR	Data type	Conf.	Acc.	Comment
-	SSWErrorOldI, SSWErorOldII	WORD	no	RO	Software error of the Secondary memorized, no standard adjustment
-	SecBootReaI, SecBootReaII	WORD	no	RO	Active boot cause of the Secondary, no standard adjustment
-	SecSWErrorI, SecSWErrorII	WORD	no	RO	Active software error of the Secondary, no standard adjustment
DispText	DispText	STR16	yes	RO	Text for display header
VName1..12	VName1..12	STR16	yes	RO	Variable name for indication on the display
VType1..12	VType1..12	INT	no	RO	Data type of the variables on the display
VUnit1..12	VUnit1..12	STR8	yes	RO	Units for indication on the display
FastExe	FastExe	BOOL	yes	RO	TRUE: I/O cycle fastest possible
RSLock	RSLock	BOOL	yes	RW	TRUE: Run/Stop switch disabled
WebLock	WebLock	BOOL	yes	RW	TRUE: Webserver deactivated
TelnLock	TelnLock	BOOL	yes	RW	TRUE: Telnet access deactivated
DnldLock	DnldLock	BOOL	yes	RO	TRUE: Loading deactivated

Name AC 900F	Name AC 900FR	Data type	Conf.	Acc.	Comment
SDAvail	SDAvail, SDAvail2	BOOL	yes	RO	TRUE: SD card monitoring active
SafeCfg	SafeCfg	BOOL	no	RO	TRUE: Saving the configuration (for commissioning only)

A.2.2 Outputs - AC 900F and AC 900FR

Name	Data type	Conf.	Acc.	Comment
Display-Val1..12	REAL	yes	RO	Any process variable can be assigned to each output variable. The value of the variable is indicated on the controller display. Only values between 99999.9 and -9999.9 can be indicated on the display; values beyond this range are displayed as **High** or **Low**.

A.2.3 Diagnostic data - AC900F and AC900FR

Name AC 900F	Name AC 900FR	Data type	Comment
TempHi	TempHi_IP1, TempHi_IP2	BOOL	TRUE: Excessively high module temperature (controller IP1 or IP2)
BTL	BTL_IP1, BTL_IP2	BOOL	TRUE: Battery voltage too low (controller IP1 or IP2)
PNE_1..4	PNE_IP1_1..4, PNE_IP2_1..4	BOOL	TRUE: Physical network fault for Ethernet port 1..4 (for controller IP1 or IP2)
SDCard	SDCard_IP1 SDCard_IP2	BOOL	TRUE: SD card inserted
ERR	ERR	BOOL	TRUE: Error on module
STA	STA	UDINT	Bit-encoded error
CPUTemp	CPUTemp	DINT	CPU temperature (in case of redundant controller the temperature of the active controller)

STA, error code encoding - AC 900F and AC 900FR

Error code HEX	Error
0x0001	Battery voltage too low
0x0002	Module temperature to high
0x0004	No SD card inserted (SD card monitoring must be activated)
0x0008	Configuration error (CPU)
0x0010	Network fault ETH1

Error code HEX	**Error**
0x0020	Network fault ETH2
0x0040	Network fault ETH3
0x0080	Network fault ETH4

A.3 Object data: AC 800F and AC 800FR

A.3.1 Parameters - AC 800F and AC 800FR

Name AC 800F	Name AC 800FR	Data type	Conf.	Acc.	Comment
EAOffs	-	WORD	no	RO	Offset of the output data
EamIn	-	DWORD	no	RO	Internal address for input values
SnrNo	SnrNoI, SnrNoII	STR16	no	RO	Serial manufacturing number
HrvNo	HrvNoI, HrvNoII	STR8	no	RO	Module hardware version
SrvNO	SrvNoI, SrvNoI	STR8	no	RO	Module software version
Dur	DurI, DurII	DINT	no	RO	Total number of operating hours
DurHot	DurHotI, DurHotII	DINT	no	RO	Total number of operating hours with high temperature
RSState	RSStateI, RSStateII	INT	no	RO	Run/Stop status IP1 or IP2
-	LocRSStateI, LocRSStaeII	INT	no	RO	Internal Run/Stop status IP1 or IP2 parameter
-	RedStateI, RedStateII	INT	no	RO	IP1 or IP2 redundancy status
EIAct	-	BOOL	no	RO	TRUE: Ethernet 1 active
EIIAct	-	BOOL	no	RO	TRUE: Ethernet 2 active
IOEnable	IOEnable	BOOL	yes	RO	TRUE: CAN I/O bus activated

Name AC 800F	Name AC 800FR	Data type	Conf.	Acc.	Comment
NoBattChk	NoBattChk	BOOL	yes	RO	TRUE: no check of the battery status
CPUType	CPUType	INT	yes	RO	CPU type: 1= PM 802F; 2= PM 803F
CPUType-Save	CPUTypeSave	INT	no	RO	Internal parameter for configured CPU type: 1_ PM 802F; 2= PM 803F
ObjID	ObjID	WORD	no	RO	Internal object ID of the instance
Pval0	-	DWORD	no	RO	Last value for message point 0
Ast1..4	Ast1..11	BYTE	no	RO	Status of message points 1..4 or 1..11
MP1..4	MP1..11	BYTE	yes	RO	Message priority
Pas1..4	-	BOOL	no	RO	Last message state of message points 1..4
-	SBootReaOldI, SBotReaOldII	WORD	no	RO	Boot cause of the Secondary memorized, no standard adjustment
-	SSWErrorOldI, SSWErorOldII	WORD	no	RO	Software error of the Secondary memorized, no standard adjustment
-	SecBootReaI, SecBootReaII	WORD	no	RO	Active boot cause of the Secondary, no standard adjustment
-	SecSWErrorI, SecSWErrorII	WORD	no	RO	Active software error of the Secondary, no standard adjustment

A.3.2 Diagnostic data - AC 800F and AC 800FR

Name AC 900F	Name AC 900F	Data type	Comment
TMP	TMP_IP1, TMP_IP2	BOOL	TRUE: Excessively high module temperature (controller IP1 or IP2)
BTL	BTL_IP1, BTL_IP2	BOOL	TRUE: Battery voltage too low (controller IP1 or IP2)
-	PNE_IP1, PNE_IP2	BOOL	TRUE: Physical network error (on controller IP1 or IP2)
ERR	ERR	BOOL	TRUE: Error on module
STA	STA	UDINT	Bit-encoded error

STA, error code encoding - AC 800F and AC 800FR

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
0	1	0x0001	Self-test error	The module has detected a non-tolerable error in the cyclic self-test.
1	2	0x0002	Configuration does not match	The configured module type does not correspond to the connected module type.
2	4	0x0004	Boot test error	Boot test of the module was not successful.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
3	8	0x0008	Slot not used	The configuration was loaded, but the module was not connected.
4	16	0x0010	Communication error	The module has not correctly signaled within the specified time period (no keep alive signal).

A.3.3 Diagnostic data - AC 800F modules

Diagnostic data of an object can be evaluated by assigning variables to the diagnosis components and using them in programs.

Module	Description	Name	Data type
All	Excess temperature	TMP	BOOL
All	Defective module	ERR	BOOL
All	Error code	STA	UDINT

Module	Description	Name	Data type
EI 801F EI 802F EI 803F EI 811F EI 812F EI 813F AM 801F AM 811F	Battery low	BTL	BOOL
AM 801F AM 811F SA 801F SD 802F SA 811F SD 812F FI 810F FI 820F FI 830F FI 840F	A message for these modules of the AC 800F controller can only be sent on condition that the operating system is loaded.	PFL1, PFL2	BOOL

STA error codes - FI 830F (non-redundant)

The various errors are stored in bit-encoded form in the STA diagnostic data.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
0	1	0x0001	Wrong module	Wrong module type connected. The configured module type does not correspond to the connected module type.
1	2	0x0002	Empty slot	Empty slot (for example: FI 830F is configured, but no module is connected)

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
2	4	0x0004	Identification error	No identification of module possible.
3	8	0x0008	Self-test error	Module error (self-test error) issued (e.g. by PROFIBUS master). Error code is communicated with alarm message.
4	16	0x0010	Old self-test	Old hardware self-test errors

Example:

STA = 24 (decimal) => 0x0018 => old hardware self-test errors in module and self-test error in module.

STA error codes - FI 830F (redundant)

In the redundant mode, the statuses of both controllers are combined to STA diagnosis data:

- The status of controller IP1 is set to “Low nibble” of STA diagnostic data.
- The status of controller IP2 is set to “High nibble” of STA diagnostic data.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
0	1	0x0001	Wrong module in controller IP1	Wrong module type connected. The configured module type does not correspond to the connected module type in controller IP1.
1	2	0x0002	Empty slot in controller IP1	Empty slot (for example: FI 830F is configured, but no module is connected) in controller IP1.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
2	4	0x0004	Identification error in the module of controller IP1	No identification of module possible in controller IP1.
3	8	0x0008	Self-test error in the module of the IP1 controller	Module error (self-test error), issued (for example, by a PROFIBUS master) in controller IP1. Error code is communicated with alarm message.
4	16	0x0010	Old self-test error in the module of the IP1 controller	Old hardware self-test errors in controller IP1

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
8	256	0x0100	Wrong module in controller IP2	Wrong module type connected. The configured module type does not correspond to the connected module type in controller IP2.
9	512	0x0200	Empty slot in controller IP2	Empty slot (for example: FI 830F is configured, but no module is connected) in controller IP2.
10	1024	0x0400	Identification error in the module of controller IP2	No identification of module possible in controller IP2.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
11	2048	0x0800	Self-test error	Module error (self-test error), issued (for example, by a PROFIBUS master) in controller IP2. Error code is communicated with alarm message.
12	4096	0x1000	Old self-test in the module of controller IP2.	Old hardware self-test errors in controller IP2

Example:

STA = 4096 (decimal) => 0x1002 => old hardware errors in module IP2 and one empty slot in module IP1.

A.4 Object data: AC 700F

An AC 700F process station features no own object data; the information is provided by the PM 783F CPU module:

A.4.1 Parameters - PM 783F CPU module

Name	Data type	Conf.	Acc.	Comment
EASid	INT	no	RO	Module location
SnrNo	STR16	no	RO	Serial manufacturing number
HrvNo	STR8	no	RO	Module hardware version
SrvNO	STR8	no	RO	Module software version
Dur	DINT	no	RO	Total number of operating hours
DispMode	WORD	yes	RO	Configured display mode
DispText	STRING8	yes	RO	The configured display text, if “Free text” has been selected for the display mode
RSLock	BOOL	yes	RW	TRUE: Run/Stop switch disabled
WebLock	BOOL	yes	RW	TRUE: Webserver deactivated
TelnLock	BOOL	yes	RW	TRUE: Telnet access deactivated
DnldLock	BOOL	yes	RO	TRUE: Loading deactivated
SDAvail	BOOL	yes	RO	TRUE: SD card monitoring active

A.4.2 Diagnostic data - PM 783F CPU module

Name	Data type	Comment
STA	UDINT	Bit-encoded error
BTL	BOOL	TRUE: Low battery voltage
PNE	BOOL	TRUE: Physical network error for Ethernet interface
ERR	BOOL	TRUE: Error on module

STA, error code encoding - AC 700F

Error code HEX	Error
0x0001	Battery voltage too low
0x0010	Network fault ETH1

A.5 Object data: PROFIBUS modules

A.5.1 Parameters - CM 772F/CI 773F PROFIBUS module

Name	Data type	Conf.	Acc.	Comment
SlotID	INT16	yes	RO	Slot of module 0..3
BoardType	WORD	no	RO	Internal hardware designation

A.5.2 Parameters - CI 930F PROFIBUS module

Name	Data type	Conf.	Acc.	Comment
SlotID	INT16	yes	RO	Slot of module 0..3
BoardType	WORD	no	RO	Internal hardware designation
IsRed	BOOL	no	RO	TRUE: active redundancy mode

A.5.3 Diagnostic data - PROFIBUS modules

The PROFIBUS modules CM 772F/CI 773F and PM 930F (redundant and non-redundant) provide the same diagnostic data; only the STA bit encoding is different for redundant and non-redundant use.

Name	Data type	Comment
ERR	BOOL	TRUE: Error on module
STA	UDINT	Bit-encoded error

STA error codes - CM 772F//CI 773F and CI 930F (non-redundant)

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
0	1	0x0001	Wrong module	Wrong module type plugged (for example, CI 930F configured, but CM 772F plugged)
1	2	0x0002	Empty slot	Empty slot (for example, CI 930F configured, but no module plugged)
2	4	0x0004	Identification error	No identification of module possible
3	8	0x0008	Self-test error	Module error (self-test error) issued (for example, by the PROFIBUS master). Error code is communicated with alarm message.
4	16	0x0010	old self-test error	Old hardware self-test errors

Example:

STA = 24 (decimal) => 0x0018 => old hardware errors in module and self-test error in module.

STA error code - CI 930F (redundant)

In the redundant mode the states of both controllers are combined into STA diagnosis data:

- The status of controller IP1 is set to “Low nibble” of STA diagnosis data.
- The status of controller IP2 is set to “High nibble” of STA diagnosis data.

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
0	1	0x0001	Wrong module in controller IP1	Wrong module type plugged (for example, CI 930F configured, but CM 772F plugged) in controller IP1
1	2	0x0002	Empty slot in controller IP1	Empty slot (for example, CI 930F configured, but no module plugged) in controller IP1
2	4	0x0004	Identification error in controller IP1	No identification of module possible in controller IP1
3	8	0x0008	Self-test error in module of controller IP1	Self-test error, module error issued (for example, by the PROFIBUS master) in controller IP1. Error code is communicated with alarm message.
4	16	0x0010	Old self-test error in module of controller IP1	Old hardware self-test errors in controller IP1
8	256	0x0100	Wrong module in controller IP2	Wrong module type plugged (for example, CI 930F configured, but CM 772F plugged) in controller IP2
9	512	0x0200	Empty slot in controller IP2	Empty slot (for example, CI 930F configured, but no module plugged) in controller IP2
10	1024	0x0400	Identification error in controller IP2	No identification of module possible in controller IP2

Bit # in STA	Decimal error code	Hexadecimal error code	Error	Cause
11	2048	0x0800	Self-test error in module of controller IP2	Self-test error, module error issued (for example, by the PROFIBUS master) in controller IP2. Error code is communicated with alarm message.
12	4096	0x1000	Old self-test error in module of controller IP2	Old hardware self-test errors in controller IP2

Example:

STA = 4098 (decimal) => 0x1002 => old hardware errors in module IP2 and empty slot in module IP1.

Index

A

B

C

D

E

F

H

I

N

O

P

R

S

T

W

—

www.abb.com/freelance
www.abb.com/controlsystems

—

We reserve the right to make technical changes to the products or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not assume any responsibility for any errors or incomplete information in this document.

2PAA114393-111 A