Control^{IT}

AC 800F and Control Builder F

New Features in Version 6.2

Notice

Information provided in this manual is subject to change without prior notice and represents no obligation on the part of ABB Automation Products.

The industrial standards and regulations (e.g. DIN, VDE, VDI, etc.) applicable in the Federal Republic of Germany are used. Outside the Federal Republic of Germany, the relevant national specifications, standards and regulations must be observed.

ABB Automation Products reserves all rights, especially those arising out of BGB, UWG, UrhG as well as out of industrial property rights (patents, utility models, trademarks, service trademarks and flavor samples).

The designations used and the products shown/mentioned in this manual have not been specifically marked regarding existing industrial property rights.

No part of this manual may be reproduced without prior written permission from ABB Automation Products.

Should you find any mistakes in this manual, please make a copy of the appropriate page(s) and send it/them to us with your comments. Any suggestions which may help to improve comprehension or clarity will also be gratefully accepted.

Please send your suggestions to:

Product Management Dept., DEAPR/LMS-Hannover, Fax: +49 (0)511 6782 701

AC 800F and Control Builder F
New Features in Version 6.2

Table of Contents

1 Industrial^{IT} for AC 800F and the effects on the Freelance 2000 Controllers

AC 800F is a part of the **Industrial**^{IT} Controller family. From this result some changes regarding the Freelance 2000 System licensing and naming conventions. The licensing models for the ABB Industrial^{IT} Controller series AC 800M, AC 800C and AC 800F have been standardized.

The options of all **Freelance 2000** controllers have been grouped in kits that are scalable through the number of I/Os. As a result, the licenses of each controller can be adapted to the customer's needs. These changes also have an effect on the licenses for the rack-based Freelance 2000 controllers.

DigiTool is replaced with **Control Builder F**, a programming tool from the **Engineer**^{IT} suite. Control Builder F can be used for configuring both the AC 800F and the Freelance 2000 System. The AC 800F hardware and software belong to the **Control**^{IT} suite.

The changes stated above will be effective with Version 6.2 and higher. Freelance 2000 projects created with Version 6.1 and lower can be upgraded to Version 6.2 using the project import function, as usual.

1.1 Engineer^{IT} – Control Builder F

Control Builder F is the programming tool for AC 800F and the known Freelance 2000 controllers like DCP02, DCP10, DFC01 (Freelance Select) and DFC02 (FieldController).

The following Control Builder F versions are available:

- Control Builder F Select
- Control Builder F Standard
- Control Builder F Professional

All Control Builder F versions support the following functions:

- Configuration and commissioning of user programs
- Tree view of the programs for convenient program overview and selection
- IEC 61131-3 programming (FBD, IL, LD, SFC)
- Graphical hardware configuration
- Integral fieldbus configuration for Profibus and Foundation Fieldbus
- IEC 61131-3 data types and user-defined (structured) data types
- Project-wide variables and function block lists
- Online cross-reference
- Efficient plausibility check
- Extensive online help

Additionally, online test functions (debugging functions) and a consistent graphical documentation of the entire user program are supported.

Besides the programs, all hardware components are displayed in graphical form and are submitted to a plausibility check together with the allocated software.

☞ Contrary to DigiTool, Control Builder F cannot be used for creating programs, since function block licenses are not included in its scope of delivery. Special controller software licenses have to be ordered for this purpose.

1.1.1 Control Builder F Select – the starter kit

Control Builder F Select can only be ordered together with a Freelance Select kit. With Version 6.2 and higher, the Freelance Select kit includes a control software Basic license for 500 I/Os. Compared to the Control Builder F Standard, the Control Builder F Select kit has the following limitations:

- A maximum of 1 link with a local DigiVis Select operator station
- A maximum of 1 Controller Select
- A maximum of 1 gateway
- A maximum of 2 user tasks
- A maximum of 8 control loops
- A maximum of 20 trend data recorders
- Tag names of max. 12 characters

See also Freelance Select Short Instructions.

1.1.2 Control Builder F Standard

Control Builder F Standard supports:

- Any number of programs
- A maximum of 100 clients like operator stations or gateways (DDE, OPC)
- A maximum of 10 links between a controller and clients like local operator stations and/or gateways.
- A maximum of 100 controllers
- A maximum of 9 user tasks per controller
- Tag names of max. 16 characters (so far special DigiTool option)

1.1.3 Control Builder F Professional

Control Builder F Professional supports all functions of Control Builder F Standard and additionally provides the following features:

- Access protection (lock function)
- User-defined function blocks (engineering license)
- DTM/FDT support

1.2 Control^{IT} for AC 800F and Freelance 2000

With V6.2 and higher a control software license is needed for every process station. Control IT offers two license types. They are different in terms of their functionality and of the number of supported I/Os. The license types are:

- Basic control software license
- Advanced software license

The license functionality can be adapted to the customer's needs for every controller, and the number of I/Os can be extended in steps of 100.

☞ Note that all controllers/process stations within a system must use the same control software license. Mixing basic and advanced control software licenses within one system is not permissible.

A Version 6.2 control software license allows you to:

- enable specific function block libraries for a controller
- enable in Control Builder F a specific number of I/O signals (I/O components) which may be used in the controller/process station configuration.

1.2.1 Basic control software license

The Basic control software license includes the following function block libraries:

- IEC 61131-3, digital and analog value processing
- controllers (without self-tuning)
- integration of fieldbuses (Profibus, Foundation Fieldbus)
- DigiVis support

1.2.2 Options for the Basic control software license

The standard Basic control software license can be extended with the following options:

- Serial Package (Modbus Master, Modbus Slave, Sartorius balance interface, Protronic coupling)
- Tune (PID self-tuning)
- Sequence of Events (with Rack I/O)
- Open Communication Package (DDE, OPC, trend server)
- Batch Function Block Package (PLI, FPX)
- Interbus
- Telecontrol Library (IEC 60870-5)
- DMS-API
- Maestro-UX Coupling

1.2.3 Advanced control software license

The Advanced control software license provides Operate IT B support (Operate IT A Support if available). It includes the function block libraries of the Basic license and additionally contains the following function block libraries and functionalities:

- Serial Package (Modbus Master, Modbus Slave, Sartorius balance interface, Protronic coupling)
- Open Communication Package (DDE, OPC, trend server)
- Tune (PID self-tuning)
- Sequence of Events
- Maestro-UX Coupling

☞ When you are extending your Basic license to an Advanced one at a later time, you can continue to use the existing options for the Basic license. Additionally, all options of the Advanced license are available.

1.2.4 Options for the Advanced control software license

The Advanced license already includes the majority of the Basic license options. Additionally, the following options are provided:

- Batch Function Block Package (PLI, FPX)
- Interbus
- Telecontrol Library (IEC 60870-5)
- DMS-API

1.3 Changes to the license model

Besides the Control Builder F engineering program both the used controllers/process stations with the respective number of I/Os and the options must be licensed. The number of licensed controllers/process stations and the number of I/Os and options licensed for a controller/process station are counted.

1.3.1 Counting the I/Os

The number of I/Os relevant for licensing is determined by counting only those I/Os that are in use, i. e.:

- the input / output variables used in a program that is running on a controller, or
- the input / output variables provided via a gateway (OPC, trend server, RTDS...)

The input/output variable definition depends on the respective bus/fieldbus:

- Rack I/O (CAN bus): Every channel used is considered as an I/O
- Profibus: All input or output components agreed and used on the controller are considered as I/Os.
- Foundation Fieldbus: Every variable transferred between the linking device and the controller/process station is considered as an I/O.
- Modbus: All Boolean data (coils) or registers used by the transmit and receive function blocks are considered as an I/O.
- Transmit/Receive function blocks (Ethernet): Not considered as I/O.

The following data are not considered as I/Os:

- diagnostic data (DP, PA)
- alarms (FF)
- parameters (PA, FF)
- global variables

☞ Only those controllers/process stations provided with an Advanced control software license can communicate I/Os to Operate IT.

1.4 Upgrading previous software versions

From Version 6.2 on there is no longer a difference between updating and upgrading the software. It is then possible to change from Version 5.1 to 6.2 or from Version 6.2 to 6.3 or 7.1.

- The change-over from Versions V3.x, V4.x and V5.x to Version 6.2 is handled as an upgrade. Additionally, the 1-year subscription offer from the software management program is included in this upgrade. The price of this offer is 30% of the list price.
- The change from Version 6.1 to Version 6.2 is handled as a subscription. The price of this subscription is 15% of the list price.

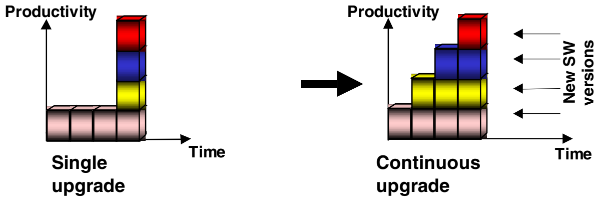
☞ When upgrading to Version 6.2, the customer decides which Control Builder F license, which control software licenses and how many I/Os he needs.
An upgrade requires a control software license for every existing process station license.
If DigitTool was licensed for five process stations, one Control Builder F license and five control software licenses have to be ordered for upgrading the system. If, for example, DigiTool was used with the user-defined function blocks, a Control Builder F Professional license is required (see **Engineer IT – Control Builder** F, page **9**).
The appropriate control software license is needed for all controllers/process stations (see **Control IT for AC 800F and Freelance 2000**, page **11**). If Operate IT is to be used on the operator level, then the Advanced control software is required for the controllers/process stations.!

☞ When upgrading your system to Version 6.2, always indicate the number of the existing hard key when ordering the software upgrade.

1.4.1 Software management program

The software management program allows you to subscribe in order to keep your installation always up to date and benefit from new features and improvement directly and without any delay. You will automatically receive the latest software versions. No need to send an additional order.

Maximum productivity through continuous software upgrades:



With every new order or upgrade the customer is entitled to receive free upgrades for one year. After one year the software management program can be renewed for one year for 15% of the list price.

2 DigiVis

2.1 Operating the logs

Log operation by keyboard entries is much easier now. When the keyboard focus is in the main display, you can directly jump from the file view to the file list by actuating ESC .

2.2 Criteria windows

The criteria windows of an SFC display can be resized as required.

2.3 Date display

The date is displayed in the DigiVis status bar in the format specified in your Windows system setup.

3 Control Builder F

3.1 Hardware structure

When you are configuring the hardware structure, the system automatically checks the computer for sufficient memory. If not enough memory is available, an alarm is generated, and you are prompted to save your project data.

Like in the tree view, the nodes of the hardware tree are marked through colored rectangles to indicate their plausibility state.

3.2 Tag list

The filter functions of the tag list have been enhanced. The function 'Only tags with faceplate' has been added.

3.3 Plausibility check

The dialog window for displaying the messages of the plausibility check can be resized as required

3.4 Configuring the messages

3.4.1 WAV files

When you are installing Control Builder F, the directory **freelance/wave** is automatically created for your WAV files. A reference pointing to this directory is entered in your Registry file under the name HKEY_LOCAL_MACHINE /DigiTool/ SETUP/ WAVFileDir. When you are configuring message items, a selection list with all WAV files available in the said directory is displayed.

3.5 Graphic editor

3.5.1 Graphic macros

All graphic macros of a project are displayed and can be edited in the project tree under a special node called **MakroPool P-MA.** New macros can be created under this node, or can be loaded into the project using the library function. The structuring element **STRUKT** is available for grouping the macros.

Like in the previous software versions, all macro functions can be called directly from the graphic display.

3.5.2 Fill levels

Besides the polygons available so far, ellipses, rectangles and circle segments can now also be used for creating fill levels.

3.5.3 Scalable text

The texts in the graphic display can now be resized using the cursor. The text displayed in the resized field is then automatically scaled to fit in.

4 Operate^{IT} B Integration

4.1 Loading data into the ConfigServer

Data configured in the Freelance system can be loaded directly from the tree view into the configuration database of an Operate IT B system. The functions **Load whole station, Load changed objects** and **Load selected objects** already known from loading data from the process, gateway or operator stations are now also available for this purpose.
The data is loaded asynchronously into the configuration server of the Operate IT B system without requiring the Control Builder F program. The progress of the loading procedure on the configuration server is recorded in a log file and can be checked at any time.

4.2 Gateway redundancy

Two OPC gateway stations can be configured in the Freelance project for realizing a redundant RTDS (RealTimeDataServer) from Operate IT B. The configurations of the two gateways are then matched with each other in the dialog of the RTDS node. The redundant RTDS is notified of both OPC servers and then activates / deactivates the appropriate one.

5 Foundation Fieldbus

5.1 Overview of Foundation Fieldbus objects

FF_FIO100 FF FIO-100 object:
Gateway for interfacing the Foundation Fieldbus with the AC 800F. The FIO-100 provides up to four Foundation Fieldbus H1 links.

FF_FIO_LINK FF FIO-100 H1 link object:
Fieldbus link for connecting FF devices using the H1 communication profile. The FIO-100 gateway provides up to four H1 links.

FF_SCHEDULE FF schedule object:
Editor for displaying the automatically-generated, time-based processing sequence of FF function blocks that have been configured in an FF link, which also provides the option of post-processing.

FF_FBAPP_D FF function block application diagram object:
Application that has been modelled using FF function blocks, and whose cycle time can be specified by the operator.

FF_FB_LIB FF block library object:
Library for accommodating the device classes for function blocks, resource blocks and transducer blocks that have been imported via FF device import. The block types contained in the block library are available for use in the project.

FF_DEV_LIB FF device library object:
Library for accommodating the device classes imported via FF device import. The device types contained in the device library are available for use in the project.

FF_FB_CLASS FF function block class object:
The function block class represents the type of a function block (analog input, PID controller etc.). The FF function block class object contains a set of initial parameters. When an instance of the function block is added to the application, it is assigned these initial parameters by default.

FF_DEV_CLASS FF device class object:
The device class represents the type of a device. The FF device class object contains the device-type-specific data from the device description (DD) and the capabilities files.

FF_FB FF function block instance object:
A specific function block with its own individual parameter data that is used in an application.

FF_DEVICE FF device instance object:
A specific field device with its own individual parameter data that is used in the project.

5.2 Technical overview

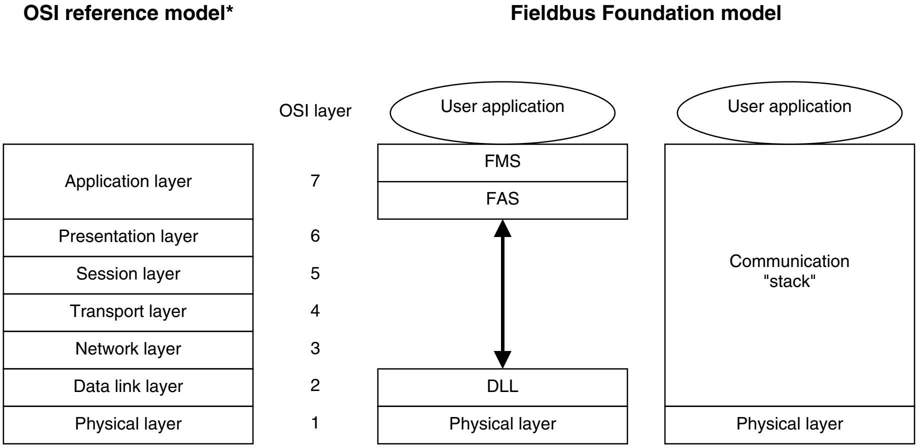
Foundation Fieldbus (FF) is a fieldbus protocol based on international standards and designed for applications in the manufacturing industry, process automation and buildings automation. The guidelines for this fieldbus standard are published by the Fieldbus FOUNDATION.

FF defines two communication profiles, H1 and HSE. The H1 profile allows transmission rates from 31.25 kbit/s to 2.5 Mbit/s. It is preferably used for direct communication between field devices in one link (H1 link). The HSE profile, with a transmission rate of 100 Mbit/s, is used primarily as a high-performance backbone for the link between H1 links. In future there will also be field devices that support the HSE profile.

The Foundation Fieldbus is interfaced with the AC 800F via the FIO-100, which provides four H1 links for this purpose. The overview below therefore refers to the H1 communication profile.

5.2.1 ISO/OSI reference model

The FF specification for the H1 communication profile is based on the ISO/OSI reference model ISO 7498 and consists of the following three functional components: the physical layer, the communication "stack" and the user application (user layer). Each layer contributes a proportion of the messages sent on the fieldbus.



*The "User application" is not defined in the OSI reference model.

ff001us.wp

Since a fieldbus system is actually a LAN which has to exchange data at high speed, only the OSI layers 1, 2 and 7 are implemented. The physical layer corresponds to OSI layer 1, which receives coded messages from the overlaid layers and converts them to physical signals on the transmission medium and back again. The communication stack corresponds to layers 2 and 7 of the OSI reference model. The OSI layer 7 (application layer) codes and decodes the commands from the user application. The OSI layer 2 (data link layer) controls via layer 1 the sending of messages over the fieldbus. This layer also controls the access rights to the fieldbus through a deterministic central bus scheduler, the link active scheduler (LAS).

The comprehensive specification of a standard user application based on function blocks and device descriptions (DD) ensures the compatibility between FF devices from different manufacturers.

5.2.2 Physical layer

FF H1 uses bus physics complying with IEC 61158-2 and ISA S50.02 Part 2.

The physical layer defines two different versions of power supply for a twisted pair cable. In the one version the field devices obtain their power via the fieldbus; this version supports low-power devices for use in intrinsically safe areas. The second version provides field devices with an external power supply.

Three transmission rates are available, ranging from 31.25 kbit/s to 2.5 Mbit/s. The transmission rates 31.25 kbit/s and 1 Mbit/s can be used in areas requiring intrinsic safety.

The table below lists the most important versions that are based on a wire conductor as a transmission medium.

Transmission rate	**31.25 kbit/s**	**1 Mbit/s**	**1 Mbit/s**	**2.5 Mbit/s**
Mode	Voltage	Voltage	Current	Voltage
Topology	Bus/tree	Bus	Bus	Bus
Power supply via bus	optional (DC)	no	optional (AC)	no
Intrinsic safety	optional	no	optional	no
Number of devices	2 – 32[1]	2 – 32	2 – 32[2]	2 - 32
Cable type	twisted pair cable	shielded twisted pair cable	shielded twisted pair cable	shielded twisted pair cable
Max. cable length	1900 m	750 m	750 m	500 m

[1] Intrinsic safety and/or power supply via the bus reduces this number.

[2] Power supply via the bus reduces this number.

5.2.3 Communication stack

5.2.3.1 Data link layer

Three types of devices are defined on the level of the data link layer (DLL):

- Basic device
- Link Master
- Bridge

Link masters are capable of taking over the role of the **Link Active Scheduler** (LAS); basic devices, on the other hand, are not capable of this. Bridges are used for connecting individual fieldbuses with larger networks.

The **Link Active Scheduler** (LAS) controls data traffic on the Foundation Fieldbus. For this purpose the LAS has a list of the transmission times of all data for cyclic transmission in all devices. Each time data transmission becomes necessary, the LAS first sends a message to the device. The device distributes the data using a broadcast on the bus (publisher). All other devices on the bus can receive this data (subscribers). In addition to this cyclical data traffic, acyclical data traffic is also possible for reading and writing parameters.

The LAS maintains a list of all devices present on the bus, the live list. New devices can be added to the bus at any time, and existing devices removed; the live list is updated automatically

The LAS is also responsible for the task of time synchronisation on the fieldbus. For this purpose it periodically broadcasts time distribution messages. It is important that the time is set exactly the same in all devices, because both the cyclical data traffic and the cyclical processing of function blocks within the user application are based on a common time.

5.2.3.2 Application layer

The **FAS** (Fieldbus Access Sublayer) as a component of the application layer provides various connection types.

- The connection type **Publisher/Subscriber** is used for a buffered 1-to-m communication. Buffered means that only the most recent data is available on the network; new data overwrites the old data. This connection type is used by the field devices for the cyclical time-controlled publishing of function block inputs and outputs. The control loop signals exchanged between the field devices, for example, use this connection type.

- The connection type **Client/Server** is used for a user-initiated, non-cyclical 1-to-1 communication between devices. Typical examples of this connection type are communication setpoint changes, user acknowledgements of messages and the downloading or reading back of device configuration data.

- The connection type **Report Distribution** is used for a user application-initiated, non-cyclical 1-to-n communication. For example, following a request by the LAS a device with a configured trend report will send the trend report to the group address defined for the purpose. Another typical use of this connection type is the sending of process alarms to the operator consoles.

FMS (Fieldbus Message Specification), as the highest-level component in the application layer, represents the interface to the user application. The FMS services allow the user application to communicate over the fieldbus using a standard set of message formats. The data items transferred over the fieldbus are described by object descriptions, which are compiled in an object dictionary (OD). The virtual field device (VFD) makes the local field device data that is described in the object dictionary available for communication via the fieldbus. The VFD is thus the external view of the local data in a field device.

5.2.4 User application

Above the OSI layer 7 (application layer) FF specifies a standard user application based on the concept of function blocks. This allows an automation task to be distributed to several FF devices. For this purpose, the standard user application defines the function block application process (FBAP). Using resource blocks (RB), function blocks (FB) and transducer blocks (TB) the automation task to be carried out is thus described. In addition, the user application also defines network management and system management.

☞ To configure the function block application process, Control Builder F uses the **function block application diagram (FBAD)**, which is produced with the help of the **FBAD editor**. See **Page 81**, **FBAD editor**.

Resource blocks describe general features of the device (device name, manufacturer ID, device type, serial number etc.). Every device has exactly one resource block.

Function blocks provide processing functions (analog input, PID controller, analog output etc.) within the field devices. Inputs and outputs of function blocks can be linked via the fieldbus. The time at which each function block is to be processed is specified exactly.

The Fieldbus FOUNDATION defines the following function blocks, known as the standard function blocks:

AAL	Analog Alarm
AI	Analog Input
AO	Analog Output
AR	Arithmetic
BG	Bias/Gain
CS	Control Selector
DC	Device Control
DI	Discrete Input
DO	Discrete Output
DT	Deadtime
IS	Input Selector
IT	Integrator (Totalizer)
LL	Lead Lag
ML	Manual Loader
OS	Output Splitter
PD	PD Control

PID	PID Control
RA	Ratio
SC	Signal Characterizer
SPG	Setpoint Ramp Generator
TMR	Timer

Transducer blocks decouple the function blocks from local input/output functions of the sensor and actuator hardware. They contain, for example, information on calibration and sensor type. There is normally one transducer block for each input or output function block.

In addition, the following objects are defined in the standard user application:

- **Link objects** define the links between inputs and outputs of function blocks both within a field device and via the fieldbus.
- **Trend objects** allow trend values for function block parameters to be logged locally; other field devices or host applications can access them.
- **Alarm objects** allow alarms and events to be reported.
- **Display objects** are pre-defined groupings of function block parameters for use in a Human System Interface (HSI).

Control Builder F allows all the objects listed above to be configured for use within the FF user application. Of these, however, the interface to the AC 800F currently only supports the link objects.

The standard user application defines the **virtual field device (VFD**) as a model used for describing the data and response of an automation device from the point of view of a partner in communication. A VFD contains all the objects and object descriptions of a user application that can be accessed by other bus subscribers.

A real FF device contains at least two virtual field devices:

- The **function block VFD** includes the objects and object descriptions in the user application that are visible over the bus network. These are essentially the function blocks and their parameters.
- The **management VFD** includes the corresponding network and system management objects. The network and system management functions are described below.

5.2.5 Network management

FF network management can be subdivided into three areas as follows:

- In the context of **configuration management**, communication parameters are initialized and modified.
- **Performance management** captures data for describing the performance of device communication.
- **Fault management** supports fault detection, fault location and fault correction.

The following specific tasks are carried out by network management:

- Loading the list of communication links;
- Configuring the communication stack;
- Loading the LAS (Link Active Scheduler);
- Monitoring the performance of device communication;
- Detecting communication errors.

An FF network has at least one network manager, who co-ordinates the network management tasks of the whole system. Each filed device also has a network management agent who manages the communication stack within the device. A network management agent carries out commands issued by the network manager, and can report events and status changes to the network manager. The network management objects and object descriptions that are required for data exchange are contained in the management VFD.

5.2.6 System management

System management co-ordinates the operation of the various field devices in a distributed fieldbus system.

The following specific tasks are carried out by system management:

- Assignment of the device address;
- Address resolution for user application tags;
- Synchronisation of the application time; the application time is the basis for time-stamping the process values on the device level;
- Time control of a distributed user application across a link;
- Commissioning of a new device on the bus;
- Monitoring of the operation of the entire system.

The system management objects and object descriptions that are required for data exchange are contained in the management VFD.

5.2.7 Object dictionary

In order to achieve a standard definition and usage of all the objects of a field device that are visible through the fieldbus network, descriptions of these objects are compiled in the **object dictionary** of the VFD. Each VFD contains exactly one object dictionary. The object dictionary of the function block VFD includes **object descriptions** of all a device's function block, resource block and transducer block parameters. An object description of this type contains, for example, the specification of a parameter's data type and length.

An object dictionary can be read on-line from the field device. The manufacturer of the device also supplies the object dictionaries in a form known as **capabilities files** on a data medium (diskette, CD-ROM). Control Builder F is based on these capabilities files; the reading of the object dictionaries from the field device is not supported at present.

There are also the files known as **value files**. A value file contains such information about the configuration of an individual field device as can change following delivery of the device. Unlike a capabilities file, a value file can be written to as well as read. It can be generated by reading the configuration data from the field device. Value files are not currently supported by Control Builder F.

5.2.8 Device descriptions

Along with the object descriptions, the **device descriptions** also form an important supplementary tool for ensuring the interoperability of FF devices. The device description contains an **extended description** for each object in the VFD which goes beyond the content of the object description. The device description includes information required by an automation system or a host in order to understand the meaning of the data in the VFD. This information includes, for example, the names of function block parameters which a host can utilize for the display, or help texts for the various objects of the device that are visible over the bus network.

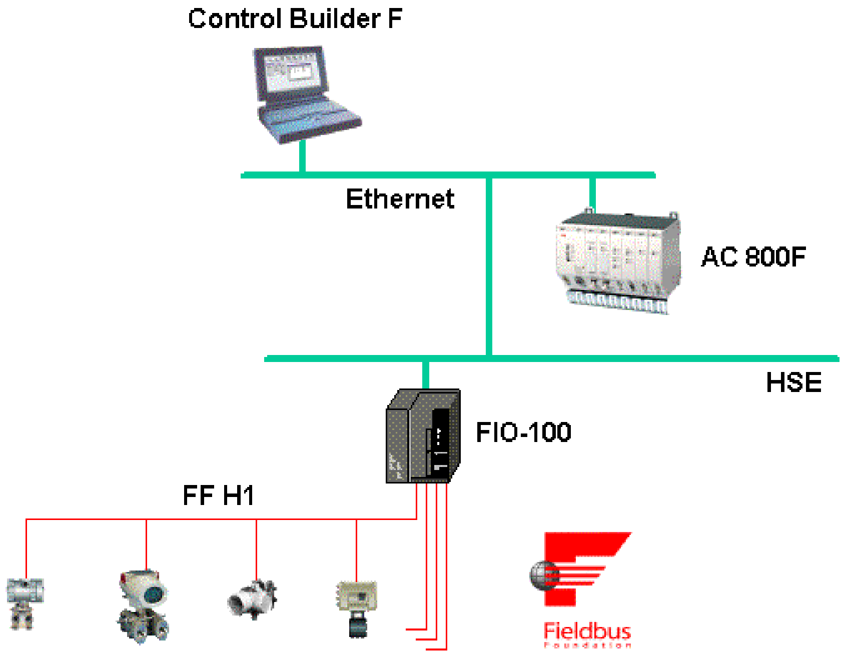
The device description is written in a standardized programming language, the **Device Description Language (DDL)** and subsequently converted to a machine-readable form by a program, the "Tokenizer". This form of the device description is referred to below as **DD (Device Description)**.

The Fieldbus FOUNDATION provides DDs for the standard function blocks and for transducer blocks. The device manufacturers supply DDs with their field devices; these refer to the standard DDs of the Fieldbus FOUNDATION and, where appropriate, complement them with manufacturer-specific features. The DDs from the device manufacturers are supplied on data carriers (diskette, CD-ROM), or may optionally be read directly from the field device.

Control Builder F offers the facility to evaluate DD files from the device manufacturers. The reading of a DD from the field device is not supported at present.

5.3 Configuration

5.3.1 Structure of FF objects



ff054us.bmp

The Foundation Fieldbus gateway FIO-100 is used for interfacing the Foundation Fieldbus with the AC 800F; this gateway is linked with the AC 800F via Ethernet. The FIO-100 provides up to four Foundation Fieldbus H1 links

The FIO-100 operates as a gateway between the AC 800F and the field devices in the H1 links, both for the field device configuration data and for the process data that is exchanged cyclically between AC 800F and the field devices.

The FIO-100 object and the lower-level FF objects listed below must be configured in the hardware structure. See **Engineering Manual**, **System Configuration**, **Hardware Structure**.

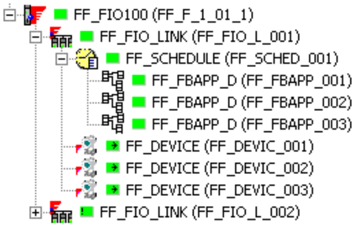
The FIO-100 object (FF_FIO100) is added to the hardware structure below an Ethernet module.

Up to four FIO-100 H1 link objects (FF_FIO_LINK) can be added below the FIO-100 object. An AC 800F supports a maximum of 32 FIO-100 H1 link objects. The bus parameters for an H1 link are set and the Foundation Fieldbus variables for this H1 link are defined in an FIO-100 H1 link object.

A schedule object (FF_SCHEDULE) is added automatically below an FIO-100 H1 link object. The graphic view of the schedule object allows the automatically-generated, time-based processing sequence (schedule) of FF function blocks that have been configured in this H1 link to be displayed and optionally post-processed.

On the same level, under an FIO-100 H1 link object, up to 32 device instance objects (FF_DEVICE) can be added. Each of these objects represents an actual field device that is present in the H1 link.

Below a schedule object, function block application diagram objects (FF_FBAPP_D) can be added. These objects represent the user applications that have been modelled using function blocks.



```
FF_FIO100 (FF_F_1_01_1)
  FF_FIO_LINK (FF_FIO_L_001)
    FF_SCHEDULE (FF_SCHED_001)
      FF_FBAPP_D (FF_FBAPP_001)
      FF_FBAPP_D (FF_FBAPP_002)
      FF_FBAPP_D (FF_FBAPP_003)
    FF_DEVICE (FF_DEVIC_001)
    FF_DEVICE (FF_DEVIC_002)
    FF_DEVICE (FF_DEVIC_003)
  FF_FIO_LINK (FF_FIO_L_002)
```

ff002us.bmp

For each FF object in the **hardware structure** the following states are displayed.

The **display state** is described by the **form** of the object node:

+ The branch is collapsed, there are further branches.

- The branch is expanded.

. There is no further branching.

The **check state** is described by the **color** of the object node:

(green) Check completed with no errors.

(pink) No check has yet been carried out following a configuration change, or an error has been detected during checking.

In the event of an error-free check, the **necessity to load** is described by the following **supplementary symbol** for the object node:

→ The object must be loaded.

! At least one object in the collapsed branch must be loaded.

!→ The object plus at least one object in the collapsed branch must be loaded.

If no supplementary symbol appears on an object, then neither that object nor any other object in the branch which may be collapsed must necessarily be loaded.

5.3.2 FF libraries

When working with FF objects a distinction is made between **classes** and **instances**.

The class determines the functionality and visual appearance of an object. For example, FF block types and FF device types are represented as classes. The classes are stored together in libraries.

To use a class, instances of the class are formed. For example, a user application is configured in the FBAD editor using FF function block instances. FF device instances, on the other hand, are used for the configuration of an H1 link in the **hardware structure**.

Both the class and the instance have a parameter input dialog. Changes carried out in the parameter input dialog for the class affect all instances for that class that are already in use as well as all instances of the class that are created later. Changes in the parameter input dialog for an instance, on the other hand, affect only that instance.

In the **hardware structure**, underneath the tree view, a window pane showing libraries can be overlaid. This window pane is shown/hidden thus:

🖱 → *Edit* → *Templates* → Choose *Libraries* tab

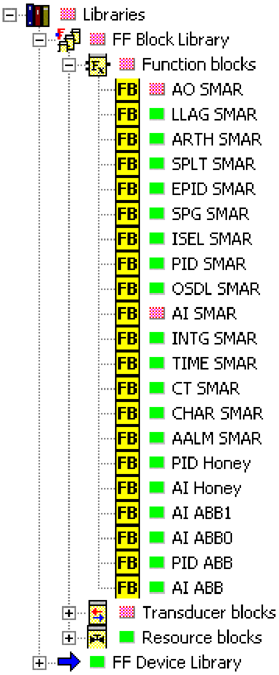
The FF block library and the FF device library are displayed, The contents of these libraries are extended by importing FF devices. See **Page 33**, **Importing FF devices**.

5.3.2.1 FF block library

The FF block library contains the block types that can be used in the project in the form of FF block classes. This library contains the following branches:

- **Function blocks** provide processing functions in the field devices.
- **Resource blocks** describe general features of the device. Every device has exactly one resource block class.
- **Transducer blocks** decouple the function blocks from local input/output functions of the sensor and actuator hardware. There is normally one transducer block class for each input or output function block class.

The FF block library, which is empty when the system is delivered, is filled on a project-specific basis by importing FF devices. See **Page 33**, **Importing FF devices**.



- ▨ Libraries
  - ▨ FF Block Library
    - ▨ Function blocks
      - FB ▨ AO SMAR
      - FB 🟩 LLAG SMAR
      - FB 🟩 ARTH SMAR
      - FB 🟩 SPLT SMAR
      - FB 🟩 EPID SMAR
      - FB 🟩 SPG SMAR
      - FB 🟩 ISEL SMAR
      - FB 🟩 PID SMAR
      - FB 🟩 OSDL SMAR
      - FB ▨ AI SMAR
      - FB 🟩 INTG SMAR
      - FB 🟩 TIME SMAR
      - FB 🟩 CT SMAR
      - FB 🟩 CHAR SMAR
      - FB 🟩 AALM SMAR
      - FB 🟩 PID Honey
      - FB 🟩 AI Honey
      - FB 🟩 AI ABB1
      - FB 🟩 AI ABB0
      - FB 🟩 PID ABB
      - FB 🟩 AI ABB
    - ▨ Transducer blocks
    - 🟩 Resource blocks
  - 🟩 FF Device Library

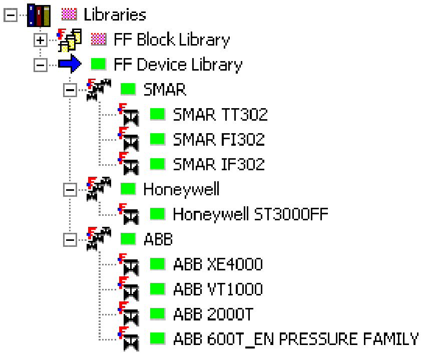
ff003us.bmp

☞ In the FF block library it is not possible to add library elements explicitly. This is performed implicitly by importing FF devices.

☞ In the FF block library it is not possible to delete library elements explicitly. This is performed implicitly by deleting FF devices in the FF device library.

5.3.2.2 FF device library

The FF device library contains the device types that can be used in the project in the form of FF device classes. By importing FF devices, the FF device library - which was empty when the system was delivered - is filled according to the specific project. See **Page 33**, **Importing FF devices**.



```
Libraries
  FF Block Library
  FF Device Library
    SMAR
      SMAR TT302
      SMAR FI302
      SMAR IF302
    Honeywell
      Honeywell ST3000FF
    ABB
      ABB XE4000
      ABB VT1000
      ABB 2000T
      ABB 600T_EN PRESSURE FAMILY
```

ff005us.bmp

When a **device class is deleted**, the elements associated with this device class in the FF block library are deleted automatically. A device class can therefore only be deleted if the following conditions are satisfied:

- the project does not contain any device instances for this device class;
- there are no block instances in the project which refer to the block classes imported with this device.

☞ In the FF device library it is not possible to add library elements explicitly. This is performed implicitly by importing FF devices.

☞ When an element in the FF device library is deleted, the elements associated with this device class in the FF block library are deleted automatically.

5.3.3 Adding FF objects

5.3.3.1 FF FIO-100 object

An FF FIO-100 object can be added within the AC 800F below the Ethernet module.

→ Select Ethernet module → *Edit* → *Insert*
→ Select FF FIO-100 object (FF_FIO100)

5.1.1.2 FF FIO-100 H1 link object

Up to four FF FIO-100 H1 link objects can be added below the FF FIO-100 (FF_FIO100).

🖱 → Select FF FIO-100 (FF_FIO100) → *Edit* → *Insert*
→ Select FF FIO-100 H1 link object (FF_FIO_LINK)
→ Select number of link (1 .. 4)

☞ When the FIO-100 H1 link object is added, an FF schedule object is added automatically to the **hardware structure** below the newly-added object.

5.3.3.3 FF schedule object

It is not possible to add an FF schedule object to the **hardware structure** explicitly since this occurs automatically when the higher-level FIO-100 H1 link object is added.

5.3.3.4 FF function block application diagram object

Below a schedule object, FF function block application diagram objects (FF_FBAPP_D) can be added.

🖱 → Select FF schedule object (FF_SCHEDULE) → *Edit* → *Insert*
→ Select FF function block application diagram object (FF_FBAPP_D)

5.3.3.5 FF device instance object:

Up to 32 FF device instance objects can be added below the FF FIO-100 H1 link object (FF_FIO_LINK).

🖱 → Select FF FIO-100 H1 link object (FF_FIO_LINK) → *Edit* → *Insert*
→ Select FF device instance object (FF_DEVICE)
→ Select device type → Press OK Button

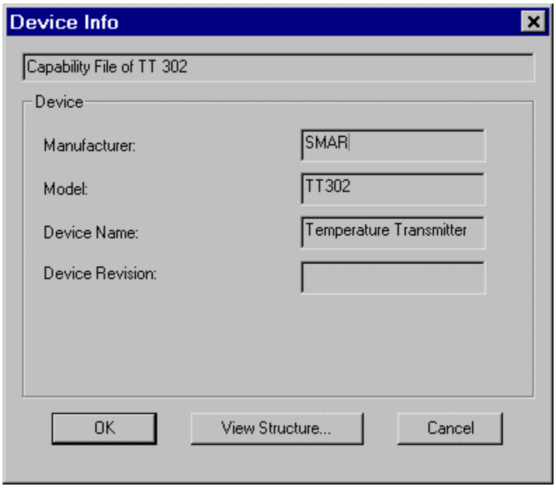
5.3.4 Importing FF devices

When an FF device is imported, this provides Control Builder F with the information needed for configuring and setting parameter values for the device and all the function blocks it supports. When a device is imported, not only a new **device type** is added to the device library; **resource and transducer blocks** as well as **function block types** that are supported by the field device are also added to the block library.

The FF device files, i.e. capabilities files (see **Page 27**, **Technical overview**, **Object dictionary)** and device description files (see **Page 27**, **Technical overview**, **Device descriptions**) provided by the device manufacturers, form the basis for the import.

Within the device library, FF devices are imported via the menu item *Insert*. With *Insert* the device selection dialog is displayed.

→ Choose device library
→ *Edit* → *Insert*
→ Select FF device class object (FF_DEV_CLASS) → Press OK button
→ Select device file with the filename extension "cff"
→ Press the OK button in the device info window


Device Info

Capability File of TT 302

Device

Manufacturer: SMAR

Model: TT302

Device Name: Temperature Transmitter

Device Revision:

ff006us.bmp

After the device has been imported, all the information needed to configure and set parameter values for the imported device type is available.

It is not possible to update a device type that already exists in the device library. An updated device description is always imported as a **new device**.

The parameters MANUFACTURER, (Manufacturer ID) DEVICE_TYPE, DEVICE_REVISION and DD_REVISION (Revision of the device description) are used for **identification of a device description**. If two device descriptions show no differences in any of these identification parameters, then they are considered to be identical; each device can be imported only once. If a device description has at least one identification parameter that differs from an existing device description, when the second device description is imported a new device is created in the device library.

In accordance with the FF specification, the path to the device files contains the **manufacturer ID** and the **device type**. The filename of a capabilities file is composed of three pairs of characters; these pairs identify the revision of the **device**, the **device description** and the **capabilities file**. The filename extension is "cff". When the device is imported, files with extensions "sym" and "ffo" which are stored in the same directory are also copied in.

Following this, the path of a capabilities file for the pressure transmitter 600T_EN from ABB Instruments is shown:

000320/0002/010302.cff

It is made up as follows:

000320	Manufacturer ID: ABB Instruments
0002	Device type: 600T_EN
01	Revision of the device
03	Revision of the device description
02	Revision of the capabilities file

The following table of manufacturer IDs does not claim to be comprehensive.

Manufacturer ID	Manufacturer
0x00012C	KROHNE Messtechnik
0x000302	SMAR International Co.
0x000309	Fuji Electric Co., Ltd.
0x000310	Micro Motion, Inc.
0x000320	ABB Instruments
0x000457	Ohmart/Vega
0x001151	Rosemount, Inc.
0x004649	Fieldbus, Inc.
0x0021C3	Bürkert Werke GmbH & Co.
0x005100	Fisher Controls International, Inc.
0x00E099	Samson AG
0x00FF01	Rotork Controls Limited
0x0B87A3	El-O-Matic
0x0DFC96	Yamatake Corporation
0x385884	Foxboro (and Foxboro-Eckardt)
0x445644	Dresser Valve Division
0x452B48	Endress + Hauser
0x464C4F	Flowserve
0x48574C	Honeywell IAC
0x4E4943	National Instruments
0x502B46	Pepperl + Fuchs
0x524149	Rosemount Analytical Inc.
0x545758	TopWorx
0x594543	Yokogawa Electric Corporation

5.3.5 Displaying the FF objects

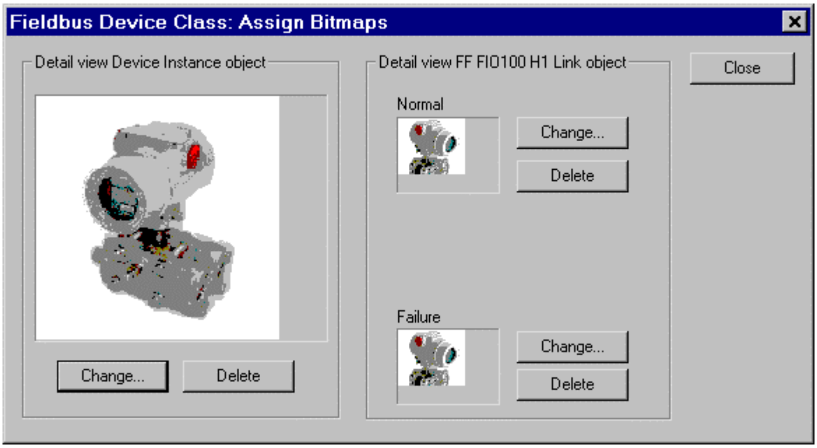
5.3.5.1 Bitmaps for field devices

A field device can be displayed using field-device-type-specific bitmaps in the detail view of the FIO-100 H1 link object and of the device instance object.

The graphical representation of a device in these views is configured in the parameter input dialog for the device class (see **Page 62**, **Parameter , FF device class object, FF_DEV_CLASS**). The following parameters can be set:

- Representation of the field device in the **detail view for the FIO-100 H1 link object**
  - with operating status "Normal" and
  - operating status "Failure"

  In each case, as 70 x 40 pixel (width x height) bitmap can be entered with up to 256 colors. If no bitmap is specified, a standard display is presented.
- Representation of the field device in the **detail view for the device instance object**.
  The recommended size for the bitmap to be entered here is 400 x 300 pixels (width x height), with up to 256 colors. The maximum size of 800 x 600 pixels (256 colors) should not be exceeded. If no bitmap is specified, then no graphic will be displayed in the detail view for the device instance.


Fieldbus Device Class: Assign Bitmaps
Detail view Device Instance object
Change...
Delete
Detail view FF FIO100 H1 Link object
Normal
Change...
Delete
Failure
Change...
Delete
Close

ff008us.bmp

CHANGE	A bitmap is assigned or an existing assignment is changed.
DELETE	An existing bitmap allocation is deleted.

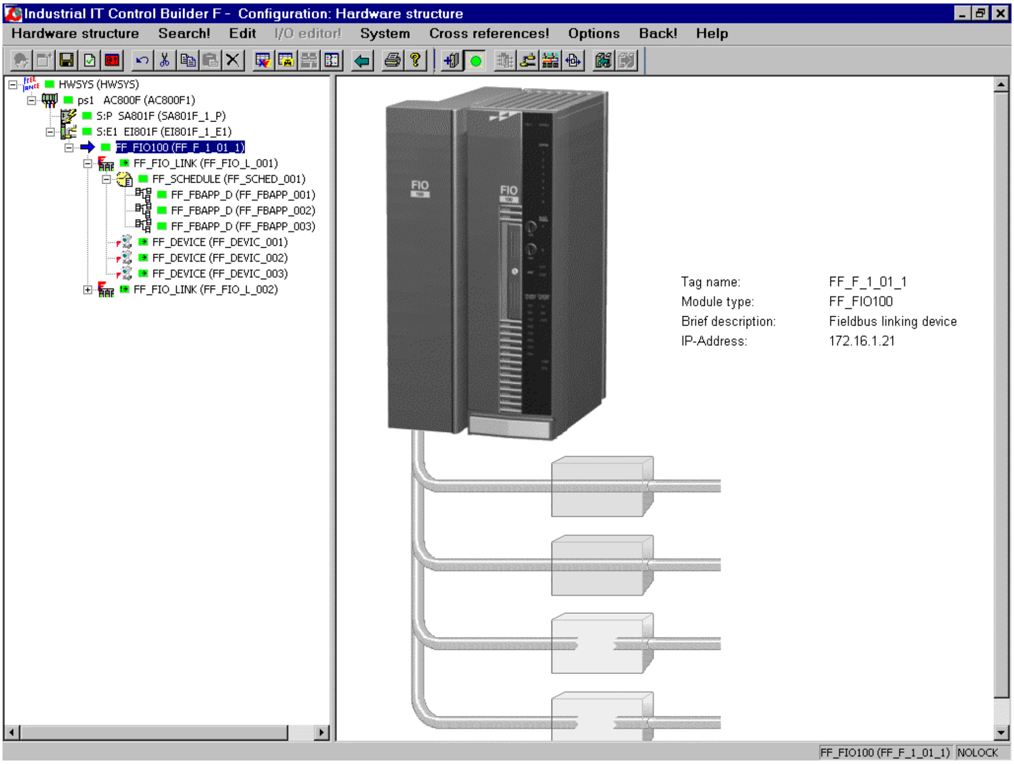
☞ While assigning a bitmap the bitmap file is stored in directory <cbf_install_dir>\bitmaps.

Bitmap files are not stored and distributed in the project. I.e. they are also not loaded along with the project onto an Operate^{IT} PC.

5.3.5.2 Detail view of the FF FIO-100 object

The detail view of the FIO-100 shows the FIO-100 and the added H1 links. For the FIO-100, the most important data such as tag name, short description, module type and configured IP address are shown.

→ Select an FF FIO-100 object (FF_FIO100) from the tree view.



```
⊟ HWSYS (HWSYS)
  ⊟ ps1   AC800F (AC800F1)
      S:P  SA801F (SA801F_1_P)
    ⊟ S:E1  EI801F (EI801F_1_E1)
      ⊟ FF_FIO100 (FF_F_1_01_1)
        ⊟ FF_FIO_LINK (FF_FIO_L_001)
          ⊟ FF_SCHEDULE (FF_SCHED_001)
              FF_FBAPP_D (FF_FBAPP_001)
              FF_FBAPP_D (FF_FBAPP_002)
              FF_FBAPP_D (FF_FBAPP_003)
            FF_DEVICE (FF_DEVIC_001)
            FF_DEVICE (FF_DEVIC_002)
            FF_DEVICE (FF_DEVIC_003)
        ⊞ FF_FIO_LINK (FF_FIO_L_002)
```

Tag name:	FF_F_1_01_1
Module type:	FF_FIO100
Brief description:	Fieldbus linking device
IP-Address:	172.16.1.21

FF_FIO100 (FF_F_1_01_1) NOLOCK

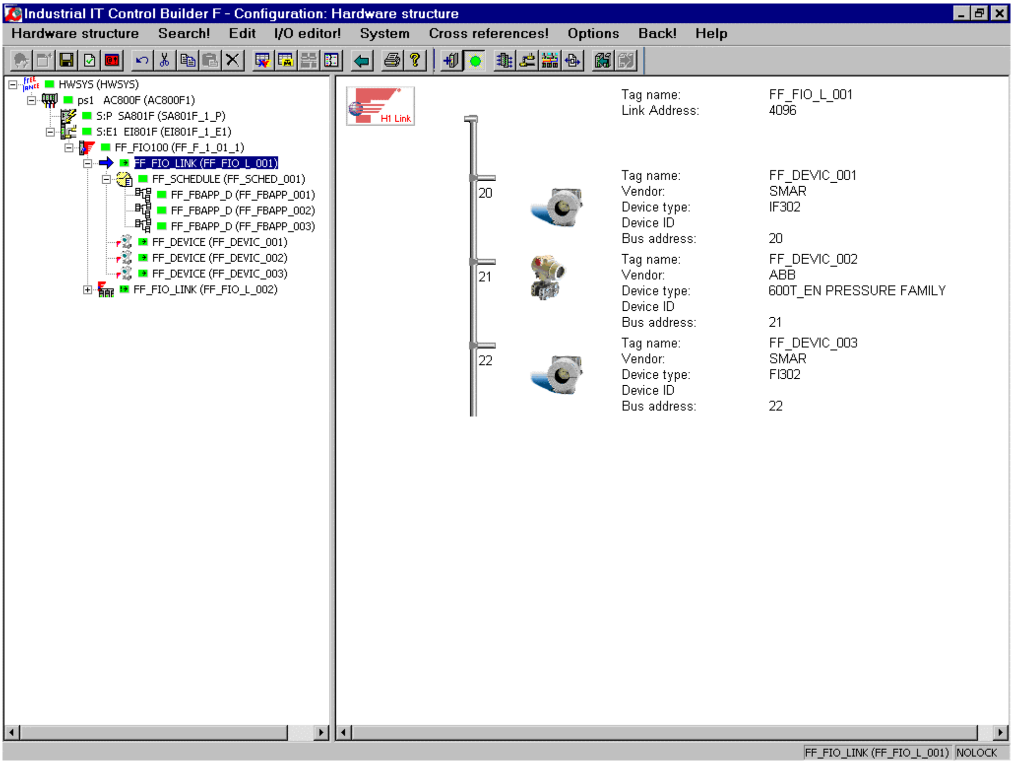
ff009us.bmp

A double-click on an FIO-100 H1 link branches to the H1 link.

5.3.5.3 Detail view of the FF FIO-100 H1 link object

The detail view of the FF FIO-100 H1 link shows the devices in the H1 link in the form of 70 x 40-pixel-sized bitmaps. If no bitmap has been assigned to the device class, then the standard display for an FF device is displayed. For the H1 link, the tag name and the H1 link address are shown. For each device the most important data is shown, such as tag name, manufacturer, device type, device ID and bus address. Other details can be found in the parameter input dialog for the object.

→ Select an FF FIO-100 H1 link object (FF_FIO_LINK) from the tree view or detail view of the FIO-100.



HWSYS (HWSYS)
ps1 AC800F (AC800F1)
S:P SA801F (SA801F_1_P)
S:E1 EI801F (EI801F_1_E1)
FF_FIO100 (FF_F_1_01_1)
FF_FIO_LINK (FF_FIO_L_001)
FF_SCHEDULE (FF_SCHED_001)
FF_FBAPP_D (FF_FBAPP_001)
FF_FBAPP_D (FF_FBAPP_002)
FF_FBAPP_D (FF_FBAPP_003)
FF_DEVICE (FF_DEVIC_001)
FF_DEVICE (FF_DEVIC_002)
FF_DEVICE (FF_DEVIC_003)
FF_FIO_LINK (FF_FIO_L_002)

Tag name:	FF_FIO_L_001
Link Address:	4096
Tag name:	FF_DEVIC_001
Vendor:	SMAR
Device type:	IF302
Device ID	
Bus address:	20
Tag name:	FF_DEVIC_002
Vendor:	ABB
Device type:	600T_EN PRESSURE FAMILY
Device ID	
Bus address:	21
Tag name:	FF_DEVIC_003
Vendor:	SMAR
Device type:	FI302
Device ID	
Bus address:	22

FF_FIO_LINK (FF_FIO_L_001) NOLOCK

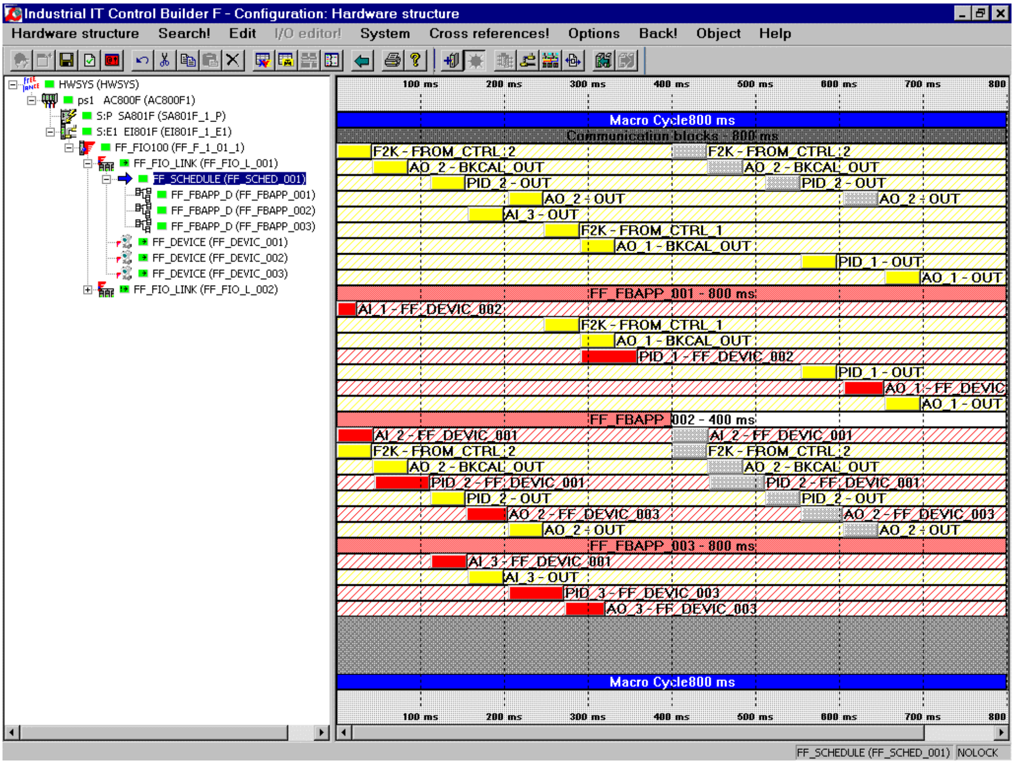
ff010us.bmp

A double-click on a device branches to the device.

5.3.5.4 Detail view of the FF schedule object

The detail view of the FF schedule object shows the schedule editor (see **Page 105**, **Schedule editor**).

→ Select an FF schedule object (FF_SCHEDULE) from the tree view.

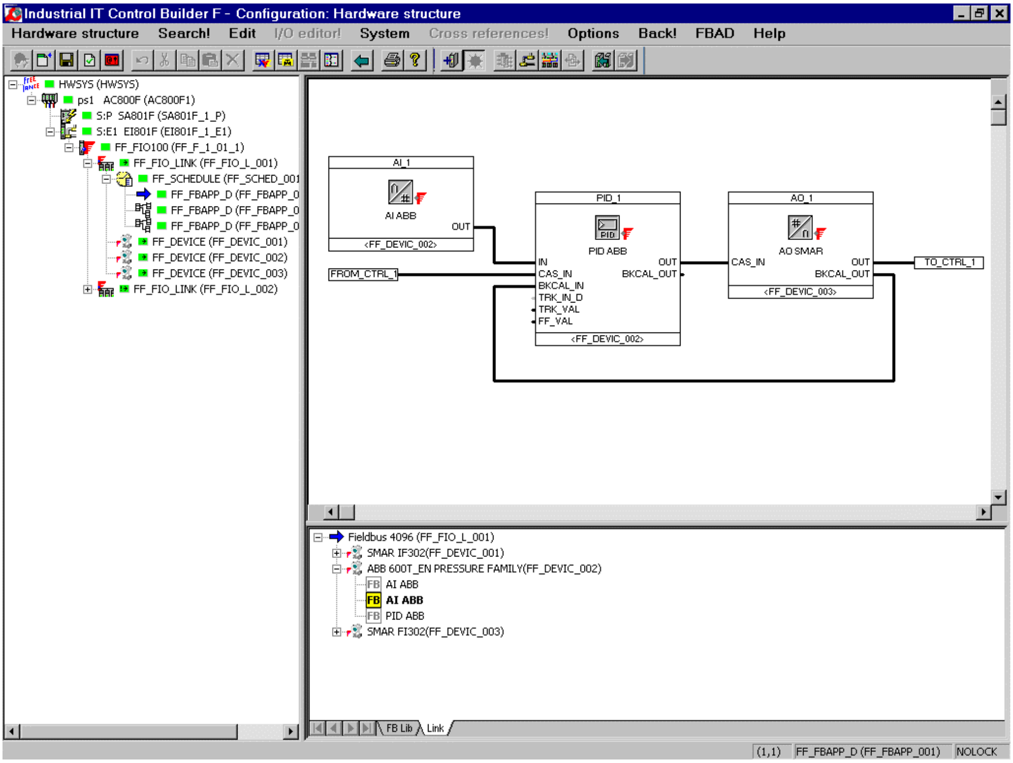


ff011us.bmp

5.3.5.5 Detail view of the FF function block application diagram object

The detail view of the FF function block application diagram object shows the FBAD editor (see **Page 81**, **FBAD editor**).

→ Select an FF function block application diagram object (FF_FBAPP_D) from the tree view.

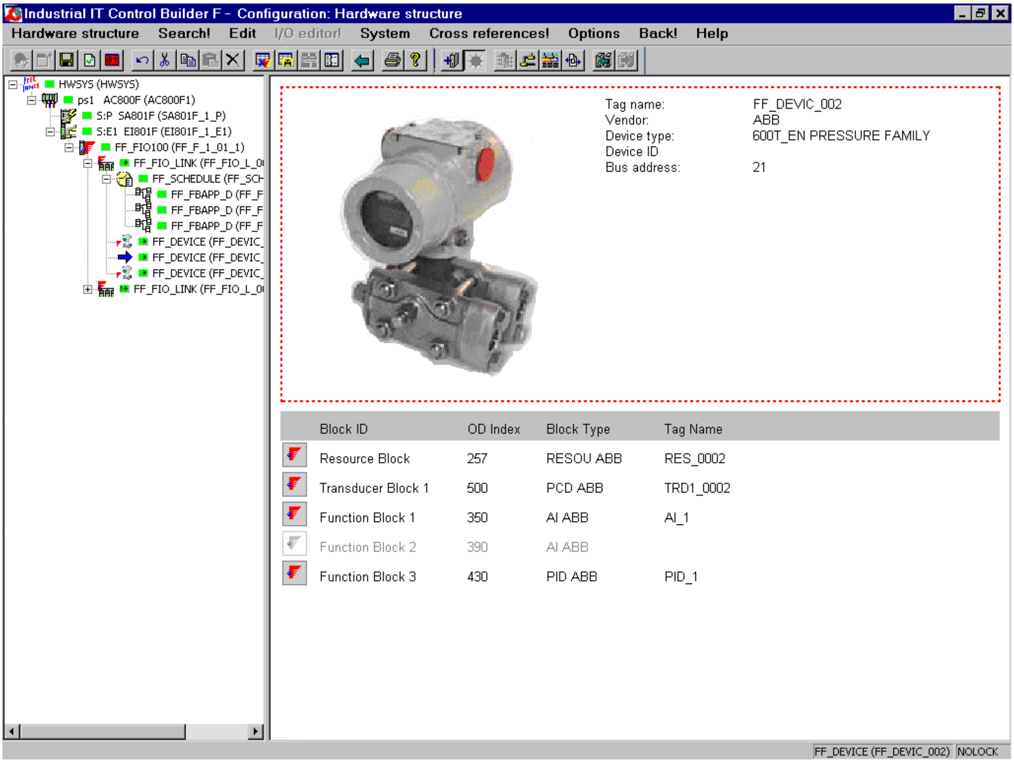


ff012us.bmp

5.3.5.6 Detail view of the FF device instance object

The detail view of the device instance shows detailed information about the field device. These include tag name, manufacturer, device type, device ID and bus address. The instances of resource blocks, transducer blocks and function blocks provided by this device instance are shown along with this information. For each instance of a block, block ID, OD index, block type and block name (tag name) are shown. Function block instances which are not yet being used in an FBAD are displayed in gray.

In order to achieve a realistic representation of the device, a bitmap can be assigned to the device class. If no bitmap has been assigned, then no graphic will be displayed in the detail view.



```
HWSYS (HWSYS)
  ps1   AC800F (AC800F1)
    S:P  SA801F (SA801F_1_P)
    S:E1  EI801F (EI801F_1_E1)
      FF_FIO100 (FF_F_1_01_1)
        FF_FIO_LINK (FF_FIO_L_0
          FF_SCHEDULE (FF_SCH
            FF_FBAPP_D (FF_F
            FF_FBAPP_D (FF_F
            FF_FBAPP_D (FF_F
          FF_DEVICE (FF_DEVIC_
          FF_DEVICE (FF_DEVIC_
          FF_DEVICE (FF_DEVIC_
        FF_FIO_LINK (FF_FIO_L_0
```

Tag name: FF_DEVIC_002
Vendor: ABB
Device type: 600T_EN PRESSURE FAMILY
Device ID
Bus address: 21

Block ID	OD Index	Block Type	Tag Name
Resource Block	257	RESOU ABB	RES_0002
Transducer Block 1	500	PCD ABB	TRD1_0002
Function Block 1	350	AI ABB	AI_1
Function Block 2	390	AI ABB	
Function Block 3	430	PID ABB	PID_1

FF_DEVICE (FF_DEVIC_002) NOLOCK

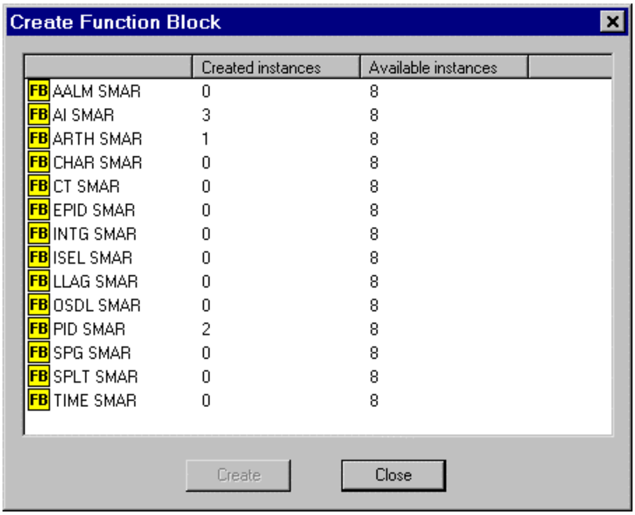
ff013us.bmp

→ Select a device instance object (FF_DEVICE) from the tree view or detail view of the FF FIO-100 H1 link.

Create FF function block instance

There are FF devices for which the FF function block instances must first be created. This is performed in the detail view for the device instance object.

→ Shortcut menu in the detail view for the device instance object
→ *Create function block*
→ Select function block type from the list → *Create*
→ Create further function blocks if required
→ *Close*



	Created instances	Available instances
FB AALM SMAR	0	8
FB AI SMAR	3	8
FB ARTH SMAR	1	8
FB CHAR SMAR	0	8
FB CT SMAR	0	8
FB EPID SMAR	0	8
FB INTG SMAR	0	8
FB ISEL SMAR	0	8
FB LLAG SMAR	0	8
FB OSDL SMAR	0	8
FB PID SMAR	2	8
FB SPG SMAR	0	8
FB SPLT SMAR	0	8
FB TIME SMAR	0	8

ff027us.bmp

Meaning of the columns

Block type The type of FF function block

Created Instances Number of instances of the function block type concerned that have already been created

Available instances Maximum number of instances of the function block type concerned that can still be created. This parameter is based on the **total** resources available in the field device for creating function blocks. When a function block is created, these resources are reduced; as a rule, therefore, the maximum number of additional function blocks of other types that can be created is also reduced.

Delete FF function block instance

This action is only available for FF devices for which the FF function block instances must be created in the manner described above. In the device instance object detail view for such devices an unused function block instance (displayed in grey) can be deleted.

🖱 → Shortcut menu of an unused function block instance
→ *Delete*

The function block instance in the field device is deleted; the resources are released.

Set parameter values for FF function block instance

Function block instances that are already in use (displayed in black) can have parameter values set via the shortcut menu.

🖱 → Shortcut menu of a used function block instance
→ *Parameters*

For details of the parameter setting process see **Page 66**, **FF function block instance object, FF_FB**.

5.4 Parameter settings

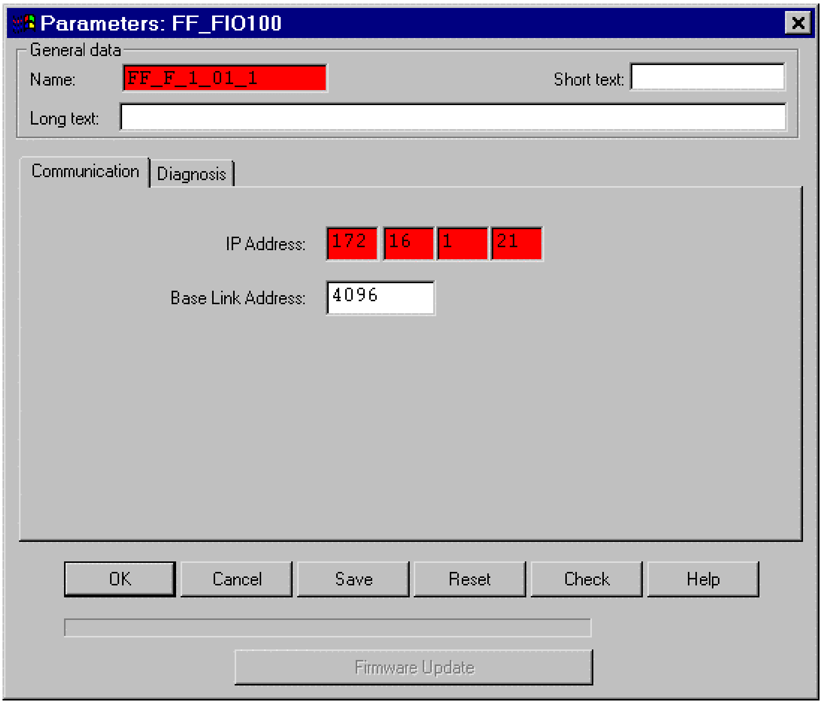
5.4.1 FF FIO-100 object, FF_FIO100

In the parameter dialog of the FIO-100 the FIRMWARE UPDATE button is available in commissioning mode. This is used for loading the current version of the firmware onto the FIO-100.

⚠ In the course of a firmware update the FIO-100 loses its configuration data. This also applies to the local configuration data of the FIO-100 H1 links.

5.4.1.1 Communication tab

On this tab parameters are input for the addresses required for communication between the AC 800F and the FIO-100.



Parameters: FF_FIO100

Name: FF_F_1_01_1

Short text:

Long text:

IP Address: 172 16 1 21

Base Link Address: 4096

ff036us.bmp

IP address IP address set on the FIO-100. It is used for communication between the AC 800F and the FIO-100.

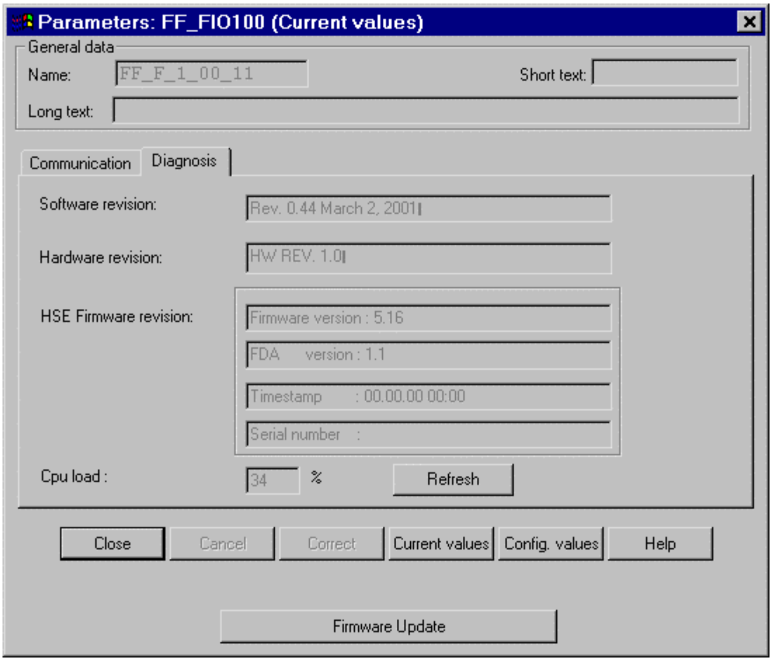
Base link address

Link address of the first H1 link of the FIO-100. The link addresses of the other H1 links of the FIO-100 are produced by incrementing this base link address.
The value range is 4096...65276.

☞ The base link address should be chosen such that the link addresses of the H1 links of the FIO-100 are unique within the HSE subnet.

5.4.1.2 Diagnosis tab

On this tab, in commissioning mode, the software revision, the hardware revision and the HSE firmware revision of the FIO-100 is shown.



Parameters: FF_FIO100 (Current values)

General data

Name: FF_F_1_00_11

Short text:

Long text:

Communication | Diagnosis

Software revision: Rev. 0.44 March 2, 2001

Hardware revision: HW REV. 1.0

HSE Firmware revision:
Firmware version : 5.16
FDA version : 1.1
Timestamp : 00.00.00 00:00
Serial number :

Cpu load : 34 %

ff063us.bmp

Software revision Software revision of the FIO-100.

Hardware revision Hardware revision of the FIO-100.

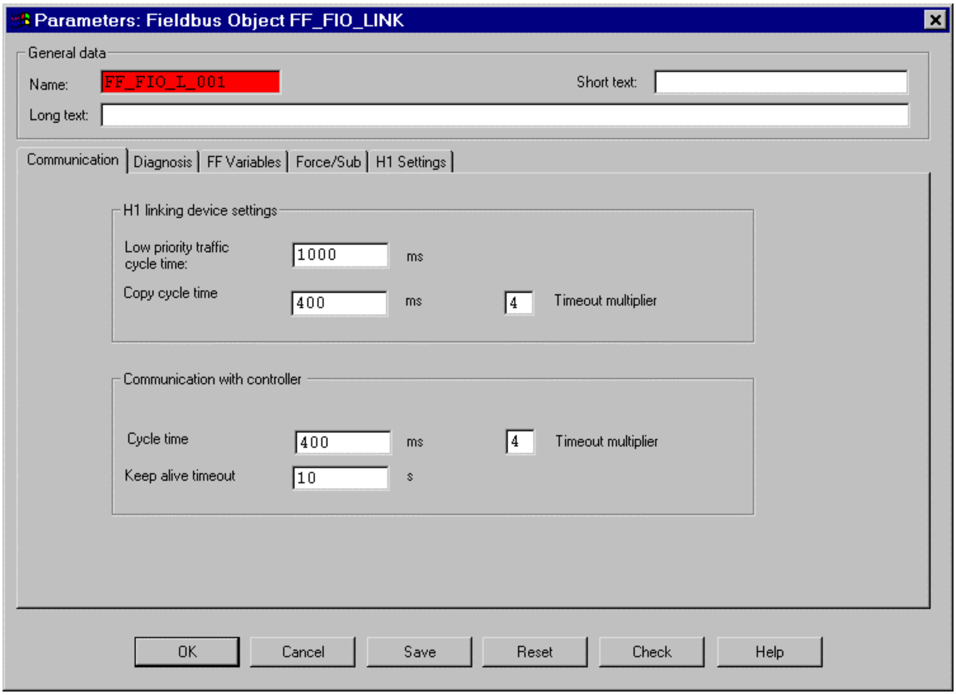
HSE firmware revision

HSE firmware revision of the FIO-100. The communication between the FIO-100 and Control Builder F is based on the HSE communication profile.

5.4.2 FF FIO-100 H1 link object, FF_FIO_LINK

5.4.2.1 Communication tab

On this tab, parameter values are set for the Ethernet communication between the AC 800F and an H1 link of the FIO-100. The **cycle time** for transferring data packets is set for both transmission directions, and the maximum number of data packets that are allowed to be lost before the **connection** is deemed to have been **terminated** is specified.



General data
Name: FF_FIO_L_001
Short text:
Long text:

H1 linking device settings
Low priority traffic cycle time: 1000 ms
Copy cycle time 400 ms
4 Timeout multiplier

Communication with controller
Cycle time 400 ms
4 Timeout multiplier
Keep alive timeout 10 s

ff037us.bmp

H1 linking device settings

Low priority traffic cycle time

Cycle time with which I/O FF variables of low priority (LP) are communicated between the FIO-100 and the field devices in the H1 link. The macro cycle for the H1 link is produced from this cycle time and the cycle times of the FBADs.

Copy cycle time

Cycle time with which I/O FF variables are copied from the Ethernet interface to the H1 link of the FIO-100 and vice versa.

Timeout multiplier Maximum number of copy cycles for which the same (non-updated) value may be transferred before a disconnection is detected for the source in the H1 link.

Ethernet communication with controller

Cycle time

Cycle time with which data packets are transferred from the FIO-100 to the controller and vice versa.

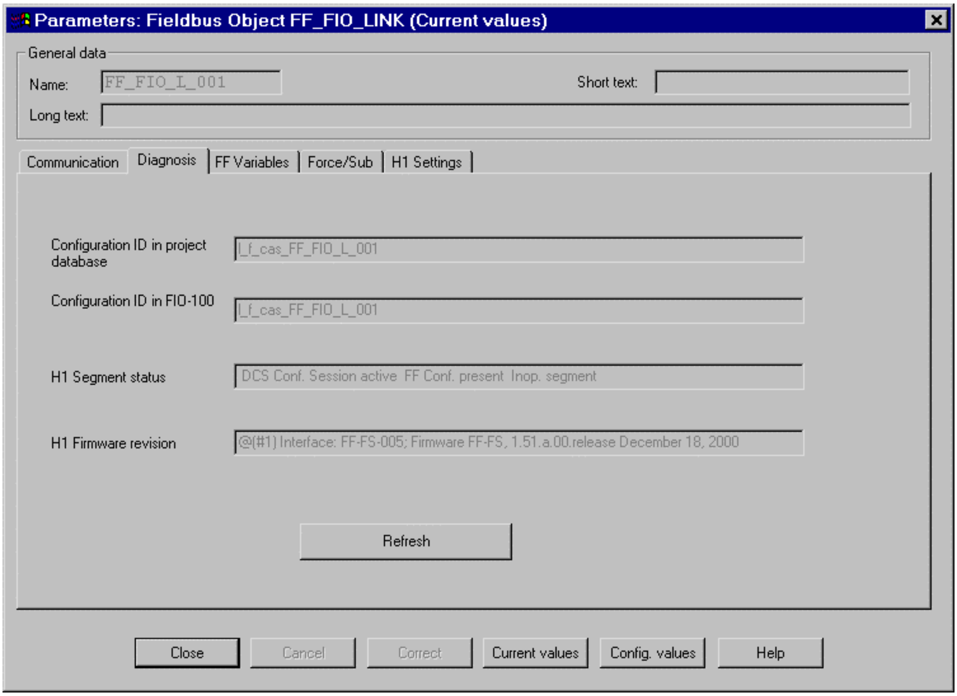
Timeout multiplier

Maximum number of data packets that can be omitted at the controller or the FIO-100 before a disconnection is detected.

Keep-alive timeout Maximum time allowed between two live messages. When the timeout set on the FIO-100 or the controller is exceeded, a disconnection is detected.

5.4.2.2 Diagnosis tab

This tab contains diagnostic information for the FIO-100 H1 link.



Parameters: Fieldbus Object FF_FIO_LINK (Current values)

General data

Name: FF_FIO_L_001

Short text:

Long text:

Configuration ID in project database: l_f_cas_FF_FIO_L_001

Configuration ID in FIO-100: l_f_cas_FF_FIO_L_001

H1 Segment status: DCS Conf. Session active FF Conf. present Inop. segment

H1 Firmware revision: @(#1) Interface: FF-FS-005; Firmware FF-FS, 1.51.a.00.release December 18, 2000

ff068us.bmp

Configuration ID in project database

Configuration ID of the FIO-100 H1 link in the project database.

Configuration ID in the FIO-100

Configuration ID of the FIO-100 H1 link in the FIO-100 (only available in commissioning mode).

H1 Link status Status of the FIO-100 H1 link (only available in commissioning mode).

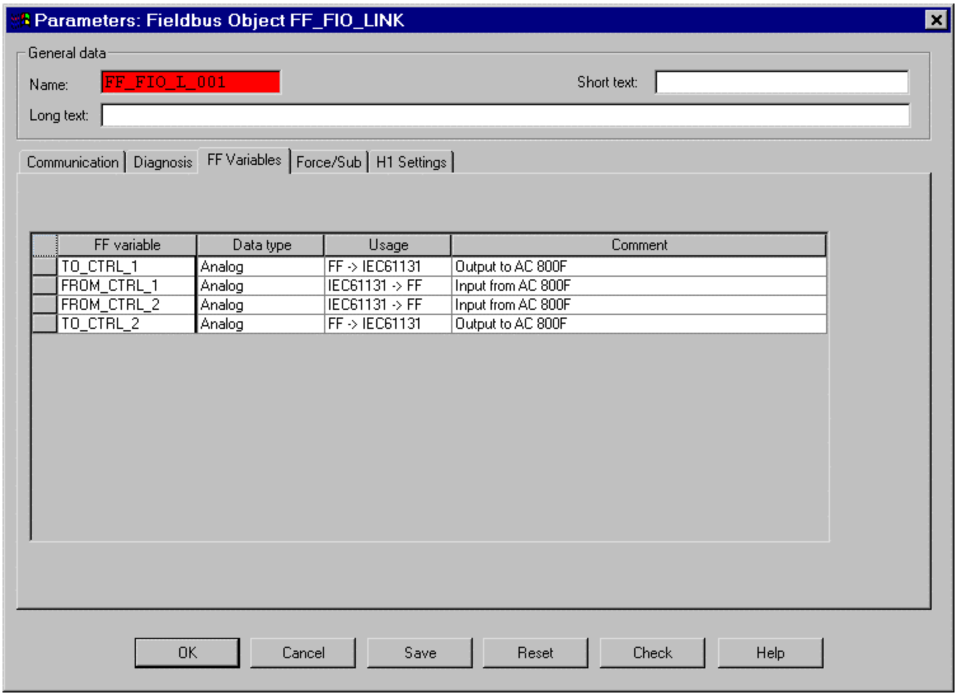
H1 Firmware revision

Firmware revision of the FIO-100 H1 link (only available in commissioning mode).

5.4.2.3 FF variables tab

This tab shows the FF variables list. It contains all the FF variables of an H1 link. For each FF variable the **variable name**, the **data type** (analog, discrete or bitstring), the **usage** (IEC 61131->FF, FF->IEC 61131 or FF local) and an optional **comment** are displayed.

Details of the FF variables list and the parameters that can be set for it, see **Page 73**, **FF variables list.**



Parameters: Fieldbus Object FF_FIO_LINK

General data

Name: FF_FIO_L_001

Short text:

Long text:

FF variable	Data type	Usage	Comment
TO_CTRL_1	Analog	FF -> IEC61131	Output to AC 800F
FROM_CTRL_1	Analog	IEC61131 -> FF	Input from AC 800F
FROM_CTRL_2	Analog	IEC61131 -> FF	Input from AC 800F
TO_CTRL_2	Analog	FF -> IEC61131	Output to AC 800F

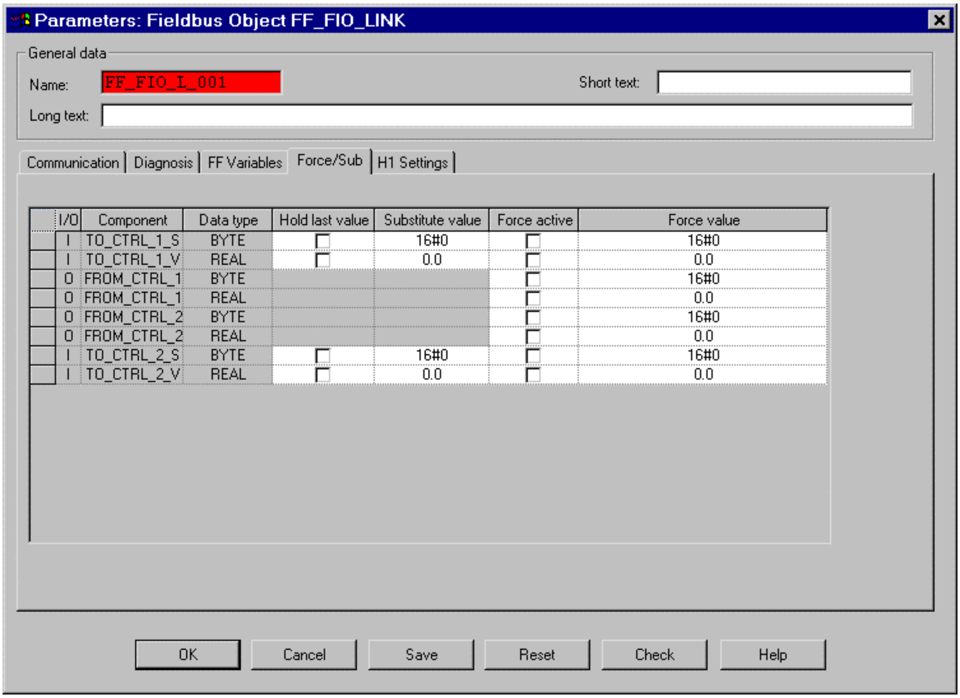
ff030us.bmp

5.4.2.4 Force values / substitute values tab

On this tab, **force values** can be configured for all components for inputs and outputs that have been generated automatically through the creation of I/O FF variables. **Substitute values** can also be assigned for input values (_V). Force values and substitute values can be written and corrected in commissioning mode of the Control Builder F

Substitute values for output values (_V), output statuses (_S) and input statuses (_S) cannot be configured. These components are identified accordingly.

The substitute values are used if the cyclical data exchange between the H1 link of the FIO-100 and the controller is interrupted.



Parameters: Fieldbus Object FF_FIO_LINK

General data

Name: FF_FIO_L_001

Short text:

Long text:

I/O	Component	Data type	Hold last value	Substitute value	Force active	Force value
I	TO_CTRL_1_S	BYTE	□	16#0	□	16#0
I	TO_CTRL_1_V	REAL	□	0.0	□	0.0
O	FROM_CTRL_1	BYTE			□	16#0
O	FROM_CTRL_1	REAL			□	0.0
O	FROM_CTRL_2	BYTE			□	16#0
O	FROM_CTRL_2	REAL			□	0.0
I	TO_CTRL_2_S	BYTE	□	16#0	□	16#0
I	TO_CTRL_2_V	REAL	□	0.0	□	0.0

ff039us.bmp

I/O
Indicates whether the component is at the input or output.
I - input / O - output

Component
Name of the component for which force values and substitute values are being specified.

Data type
Data type of the component for which force values and substitute values are being specified.

Hold last value Defines whether in the event of a fault the last value is held or the substitute value used.
□ The substitute value is used.
☑ The last value is held.

Substitute value The value used as a substitute.

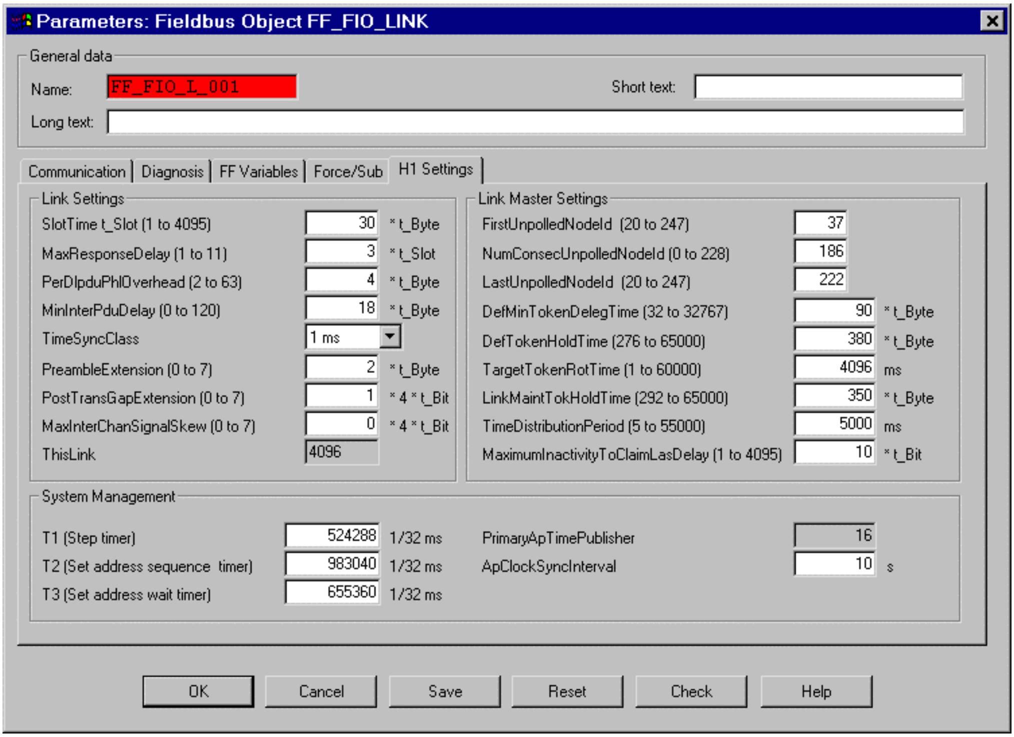
Force active Defines whether the configured force value is used.
□ The actual process value is used.
☑ The configured force value is used.

Force value The value used as a force value.

5.4.2.5 Tab H1 settings

Communication parameters and system management parameters for the H1 link are set on this tab. It is not necessary as a rule to change the default values. However, it may be expedient to change individual values in order to optimize communication.

A knowledge of the basics of FF network management and FF system management is a prerequisite for making a sensible change to the default values. You are referred for this purpose to the FF specifications *Network Management (FF-801)* and *System Management (FF-880)*. The telegram structure is described in the IEC specification *Digital data communications for measurement and control - Field bus for use in industrial control systems (IEC 61158-1..5)*.



Parameters: Fieldbus Object FF_FIO_LINK

General data

Name: FF_FIO_L_001

Short text:

Long text:

Link Settings

SlotTime t_Slot (1 to 4095)	30	* t_Byte
MaxResponseDelay (1 to 11)	3	* t_Slot
PerDlpduPhlOverhead (2 to 63)	4	* t_Byte
MinInterPduDelay (0 to 120)	18	* t_Byte
TimeSyncClass	1 ms	
PreambleExtension (0 to 7)	2	* t_Byte
PostTransGapExtension (0 to 7)	1	* 4 * t_Bit
MaxInterChanSignalSkew (0 to 7)	0	* 4 * t_Bit
ThisLink	4096	

Link Master Settings

FirstUnpolledNodeId (20 to 247)	37	
NumConsecUnpolledNodeId (0 to 228)	186	
LastUnpolledNodeId (20 to 247)	222	
DefMinTokenDelegTime (32 to 32767)	90	* t_Byte
DefTokenHoldTime (276 to 65000)	380	* t_Byte
TargetTokenRotTime (1 to 60000)	4096	ms
LinkMaintTokHoldTime (292 to 65000)	350	* t_Byte
TimeDistributionPeriod (5 to 55000)	5000	ms
MaximumInactivityToClaimLasDelay (1 to 4095)	10	* t_Bit

System Management

T1 (Step timer)	524288	1/32 ms
T2 (Set address sequence timer)	983040	1/32 ms
T3 (Set address wait timer)	655360	1/32 ms
PrimaryApTimePublisher	16	
ApClockSyncInterval	10	s

ff055us.bmp

Some of the values shown on the tab are multiples of the *bit transmission time* t_{bit}, the *byte transmission time* t_{byte} or the *slot time* t_{slot}.

t_{Bit} = 32µs — The bit transmission time is the time taken to transfer one bit. When a transmission rate of 31.25 kbit/s is used in the H1 link, this time is 32µs.

t_{Byte} = 256µs The byte transmission time is the time taken to transfer one byte. The value is eight times the bit transmission time.

t_{Slot} The slot time is a configuration parameter of the link (see below).

Link settings

The communication parameters assembled in this group are known to every bus subscriber on the H1 link.

SlotTime This parameter of the H1 link must be greater than the greatest minimum required slot time for any of the devices being operated in the H1 link. The value to be set depends on the reaction time of the devices being used.
The value of this parameter is also used by every device in the link as the basis for calculating monitoring times. These are specified as a multiple of the slot time. Link master devices use the slot time to determine the length of time they monitor the link for inactivity before trying to take over the LAS role. In the case of link master devices with consecutive addresses, the inactivity monitoring times differ by exactly one slot time.
The value range is (1..4095) * t_{Byte}.

MaxResponseDelay This parameter for the H1 link must be greater than the greatest response delay time on any of the devices being operated in the H1 link with acyclical communication. The maximum response delay time of a device is the time between the end of receiving a request message and the beginning of sending the immediately following response message.
The value range is (1..11) * t_{Slot}.

PerDlpduPhlOverhead
This parameter specifies the time interval between two messages caused by the physical layer. It takes into account the transfer times for preamble, start flag, end flag, *PreambleExtension* and *PostTransGapExtension*.
The value range is (2..63) * t_{Byte}.

☞ This value cannot be edited; if the values of *PreambleExtension* or *PostTransGapExtension* are altered, it is adjusted automatically.

MinInterPduDelay This parameter specifies the minimum interval between two messages in the H1 link. It must be at least as great as the greatest value for any of the devices being operated in the H1 link.
The value range is (0..120) * t_{Byte}.

☞ If the value chosen for this parameter is greater than the product of *MaxResponseDelay* and *SlotTime*, then no communication is possible in the H1 link concerned.

TimeSyncClass

This parameter determines the quality of time synchronisation in the H1 link. The time synchronisation class must be selected such that all devices in the H1 link support at least that time synchronisation class if not one with a higher resolution.

☞ This parameter must be set to a suitable value according to the parameter *TimeDistributionPeriod*.

PreambleExtension

Each transmission contains a preamble sequence 8 bits in length. This parameter allows the preamble to be increased by a further 0..7 8-bit sequences.
The value range is (0..7) * t_{Byte}.

☞ Increasing this value leads to an increase in *PerDlpduPhlOverhead*.

PostTransGapExtension

Following every transmission, a gap of 4 bit transmission times is observed. This parameter allows the gap to be increased by a further 0..7 4-bit transmission times.
The value range is (0..7) * 4* t_{Bit}.

☞ Increasing this value leads to an increase in *PerDlpduPhlOverhead*.

MaxInterChanSignalSkew

Devices that receive data via several ports from the same H1 link must take into account a maximum time shift (signal skew) of 5 bit transmission times. This parameter allows the maximum permissible time shift for the H1 link to be increased by 0..7 4-bit transmission times.
The value range is (0..7) * 4* t_{Bit}.

ThisLink

This parameter contains the link address of the H1 link.

☞ The H1 link address must be unique within the HSE subnet.

☞ The H1 link address is set using the base link address of the FIO-100 in the parameter dialog for the FIO-100 (see **Page 43, FF FIO-100 object, FF_FIO100**).

Link Master settings

The Link Master, which handles the role of the LAS, uses the communication parameters compiled in this group for controlling the communication in the H1 link .

FirstUnpolledNodeId

This parameter shows the **first** bus address from a sequence of consecutive bus addresses, which is excluded from the regular polling for detecting newly joined bus subscribers.
In order to detect newly joined bus subscribers automatically, all bus addresses that are not contained in the live list of the LAS are addressed at regular intervals (polling). With this and one of the following two parameters, a cohesive range of addresses can be defined which is not subject to this polling.
The value range is 20...247.

☞ See also *NumConsecUnpolledNodeId*, *LastUnpolledNodeId*.

NumConsecUnpolledNodeId

This parameter specifies the **number** of a range of consecutive bus addresses which is excluded from the regular polling for detecting newly-added bus subscribers.
The value range is 0..228.

☞ If this parameter is changed, the value of parameter *LastUnpolledNodeId* is automatically adjusted accordingly. See also *FirstUnpolledNodeId*, *LastUnpolledNodeId*.

LastUnpolledNodeId

This parameter specifies the **last** bus address from a number of consecutive bus addresses which are excluded from the regular polling for detecting newly-added bus subscribers.
The value range is 0..247.

☞ If this parameter is changed, the value of parameter *NumConsecUnpolledNodeId* is automatically adjusted accordingly. See also *FirstUnpolledNodeId*, *NumConsecUnpolledNodeId*.

DefMinTokenDelegTime

In order to perform a token transfer for handling non-cyclical data traffic, at least the length of time specified in this parameter must be available. If less time than that is available before the beginning of the next cyclic data transmission, then the LAS will not transfer the token before. This parameter should be set such that short messages can be sent within the time specified.
The value range is (32..32767) * t_{Byte}.

DefTokenHoldTime

This parameter defines the length of time that the LAS initially allows to each device for non-cyclical data traffic within a token rotation. This parameter differs from the parameter *DefMinTokenDelegTime* in that the time can be distributed over several token transfers.
The value range is (276..65000) * t_{Byte}.

TargetTokenRotTime

This parameter specifies the target upper time limit for a complete token rotation.
The value range is 1..60,000 ms.

☞ When defining this value, the maximum number of devices being operated in the H1 link and the values of parameters *DefTokenHoldTime* and *LinkMaintTokHoldTime* should be taken into account.

LinkMaintTokHoldTime

This parameter is used to specify the time that the LAS uses within a token rotation for maintenance of the H1 link. This includes the detection of new bus subscribers and the broadcasting of LAS status and of the time.
The value range is (292..65000) * t_{Byte}.

TimeDistributionPeriod

This parameter is used to specify the maximum time interval between the cyclical time broadcasting messages (scheduling time) in the H1 link. The value set must be appropriate for the time synchronisation class selected (see *TimeSyncClass* parameter). The following table shows the relationships.
The value range is 5..55,000 ms.

TimeSyncClass	**Maximum value for *TimeDistributionPeriod***
1 µs	5 ms
10 µs	50 ms
100 µs	500 ms
1 ms	5 s
10 ms	10 s
100 ms	25 s
1 s	55 s

☞ To avoid increasing the load on the bus unnecessarily, the maximum time interval for the selected time synchronisation class, which is shown in the right-hand column in the table, should be set.

MaximumInactivityToClaimLasDelay

The value of this parameter must be greater than the greatest internal delay time of any of the link master devices being operated in the H1 link.
This delay time for a link master device is the time interval between the expiry of the inactivity monitoring time and the sending of the necessary message for taking over the LAS role.
The value range is (1..4095) * t_{Bit}.

System management

This group contains the system management parameters.

T1

This parameter is used to define the maximum acknowledgement delay for the SETADDRESS service.
The value range is (0..2^{32}-1) * (1/32) ms.

T2

The allocation of a new address, which starts with the receipt of a SETADDRESS message, must be completed within the time that has been set using this parameter.
The value range is (0..2^{32}-1) * (1/32) ms.

T3

When a new address is assigned to a device, it must report back under the new address on the bus at least before the expiry of the time set here.
The value range is (0..2^{32}-1) * (1/32) ms.

PrimaryApTimePublisher

This is the address of the device which manages and broadcasts the application time.

ApClockSyncInterval

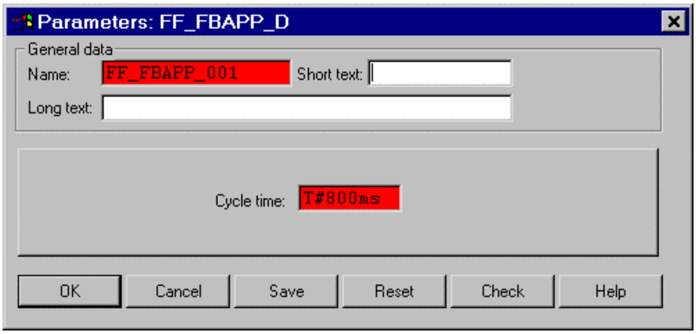
This parameter is used to define the maximum time interval between the cyclical distribution telegrams for the application time in the H1 link.
The value range is 1..255 s.

5.4.3 FF schedule object, FF_SCHEDULE

The parameter dialog for the FF schedule object contains no parameter data other than the general data. Parameters can be defined in the detail view of the FF schedule object, the schedule editor. See **Page 105**, **Schedule editor.**

5.4.4 FF Function block application diagram object, FF_FBAPP_D

All the FF function blocks used in a function block application diagram (FBAD) are processed cyclically with the same **cycle time**. This time, which is also known as FBAD cycle time, is specified on the tab. This parameter is preset to a default value.



Parameters: FF_FBAPP_D

General data

Name: FF_FBAPP_001

Short text:

Long text:

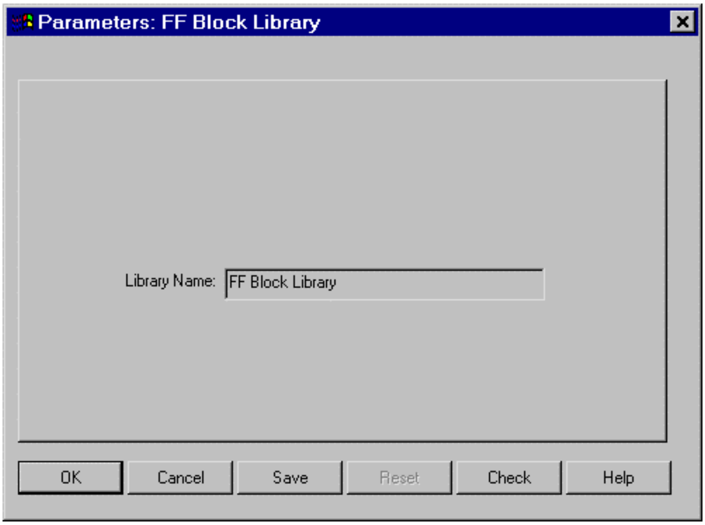
Cycle time: T#800ms

ff040us.bmp

A detailed description of the FBAD can be found elsewhere in the documentation. See **Page 81, FBAD editor**.

5.4.5 FF Block library object, FF_FB_LIB

The preset **name** of the FF block library is displayed on this tab. This parameter cannot be changed.



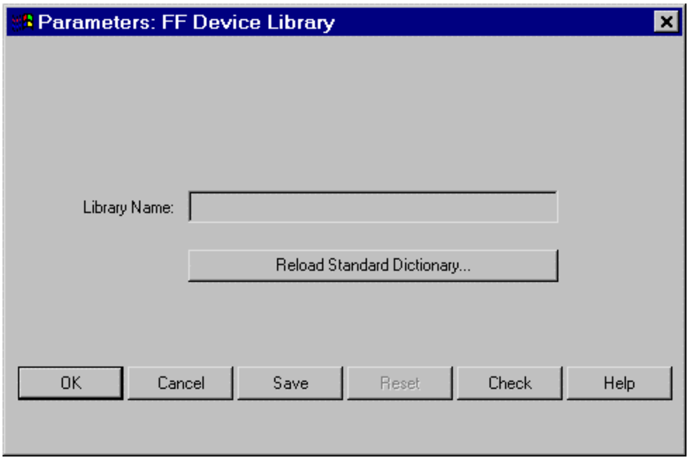
Parameters: FF Block Library

Library Name: FF Block Library

ff041us.bmp

5.4.6 FF Device library object, FF_DEV_LIB

In this parameter dialog the **name** of the FF device library is displayed. This parameter cannot be changed.



ff042us.bmp

RELOAD STANDARD DICTIONARY

This button can be used to reload the **standard dictionary** provided by the Fieldbus FOUNDATION. The standard dictionary contains text in several languages for objects that are defined in the FF specification. The standard dictionary contains, for example, the names for inputs and outputs of the FF standard function blocks and the names and help texts for block parameters.

☞ Since the standard dictionary is loaded automatically when a project is newly created, reloading is only necessary if an updated version is to be used.

5.4.7 FF Function block class object, FF_FB_CLASS

The **header area** of the parameter dialog contains the following data.

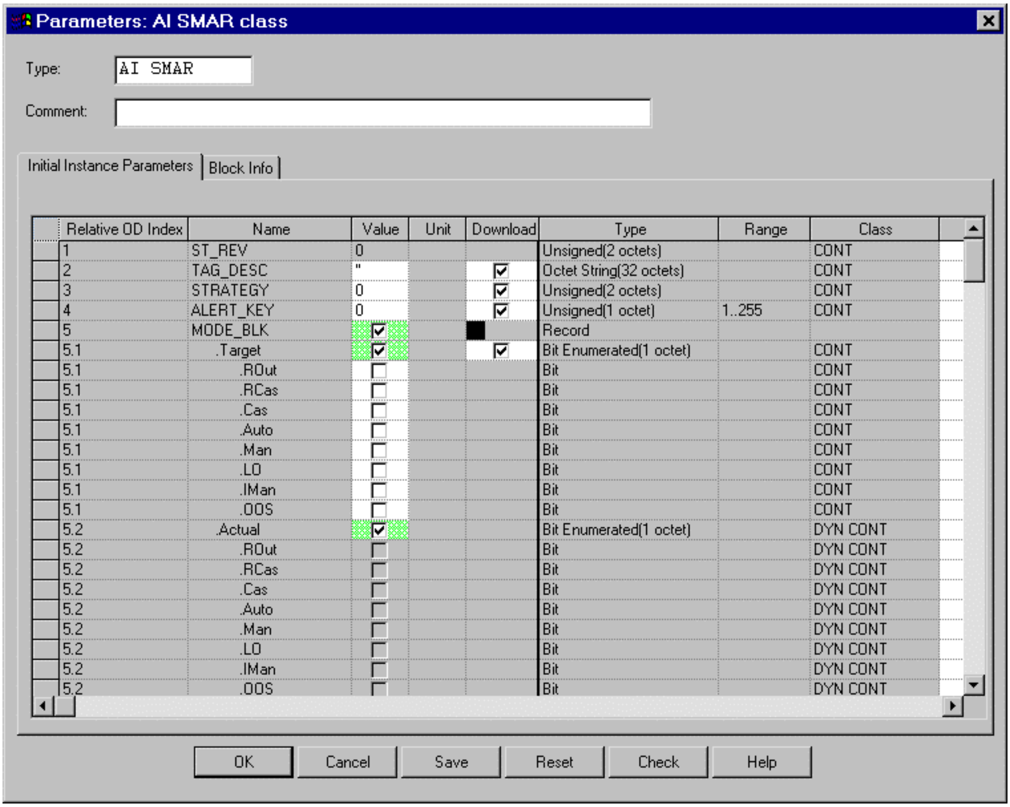
Type Name of the **FF function block class** (max. 12 characters). This name is defined during the FF device import, and can be changed later. It must be unique within the project. This name is displayed in the FF block library and as a type name for a function block instance in the FBAD amongst other places.

Comment Free comment text (max. 42 characters).

5.4.7.1 Initial instance parameters tab

A complete set of initial instance parameters is stored on the FF function block class object. A newly created function block instance is assigned these parameter values by default.

On this tab, the initial instance parameter values for the FF function blocks are set. The layout and contents of the tab correspond to those of the FF function block instance. See **Page 68**, **FF function block instance object, FF_FB, Parameters tab**.



Parameters: AI SMAR class

Type: AI SMAR

Comment:

Initial Instance Parameters | Block Info

Relative OD Index	Name	Value	Unit	Download	Type	Range	Class
1	ST_REV	0			Unsigned(2 octets)		CONT
2	TAG_DESC	"		☑	Octet String(32 octets)		CONT
3	STRATEGY	0		☑	Unsigned(2 octets)		CONT
4	ALERT_KEY	0		☑	Unsigned(1 octet)	1..255	CONT
5	MODE_BLK	☑		■	Record		
5.1	.Target	☑		☑	Bit Enumerated(1 octet)		CONT
5.1	.ROut	☐			Bit		CONT
5.1	.RCas	☐			Bit		CONT
5.1	.Cas	☐			Bit		CONT
5.1	.Auto	☐			Bit		CONT
5.1	.Man	☐			Bit		CONT
5.1	.LO	☐			Bit		CONT
5.1	.IMan	☐			Bit		CONT
5.1	.OOS	☐			Bit		CONT
5.2	.Actual	☑			Bit Enumerated(1 octet)		DYN CONT
5.2	.ROut	☐			Bit		DYN CONT
5.2	.RCas	☐			Bit		DYN CONT
5.2	.Cas	☐			Bit		DYN CONT
5.2	.Auto	☐			Bit		DYN CONT
5.2	.Man	☐			Bit		DYN CONT
5.2	.LO	☐			Bit		DYN CONT
5.2	.IMan	☐			Bit		DYN CONT
5.2	.OOS	☐			Bit		DYN CONT

ff044us.bmp

The column *download* is mentioned here because the values can only be edited at the function block class; it is not possible to edit them at the block instance. For each parameter it is determmined whether or not it shall be downloaded with the device configuration data.

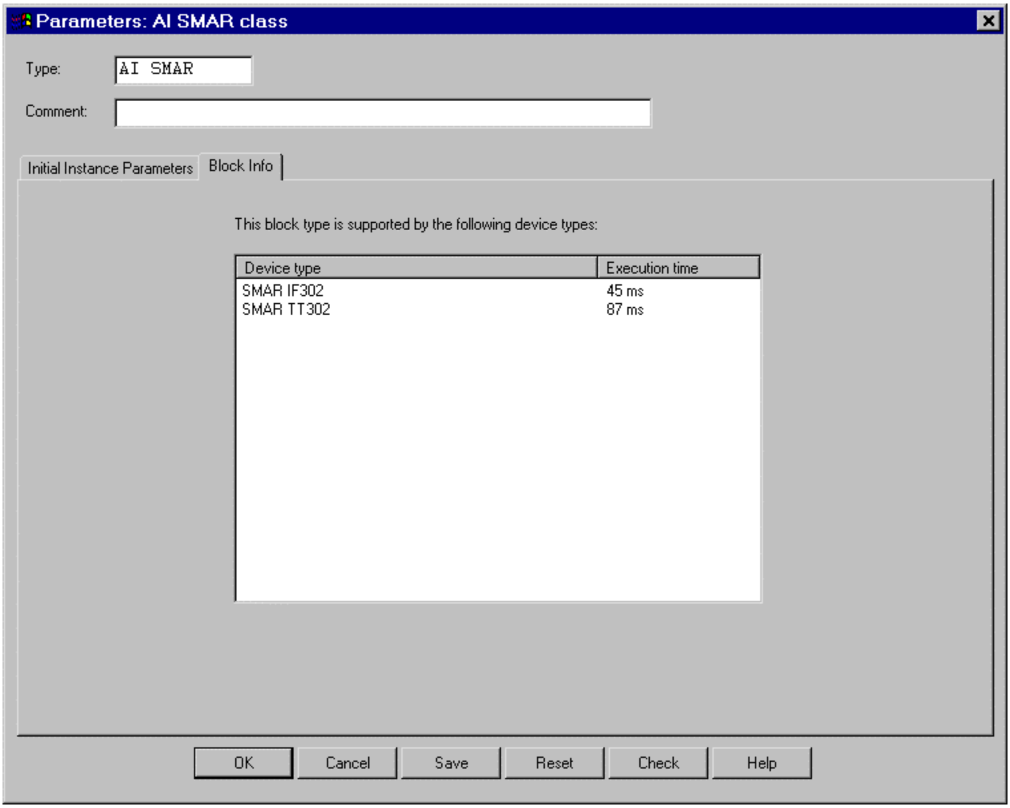
☐ The parameter is not downloaded to the FF device.

☑ A parameter value that is present at the function block instance is written to the FF device, provided that it is a writable parameter. If no parameter value is present in the value field at the function block instance, this parameter is not downloaded with the device configuration data.

In the case of a structural element the selection state of the data structure is displayed.

■ All elements of the data structure are selected for downloading.
◪ Some elements of the data structure are selected for downloading.
□ No element of the data structure is selected for downloading.

5.4.7.2 Block Info tab



Type: AI SMAR

Comment:

This block type is supported by the following device types:

Device type	Execution time
SMAR IF302	45 ms
SMAR TT302	87 ms

ff043us.bmp

The table lists the device types that support this block type.

Device type Name of the **FF device class** that supports this FF block class. This means that an instance of this block may be assigned to a device of the given type.

Execution time Time taken for execution of the function block in a device of the given type.

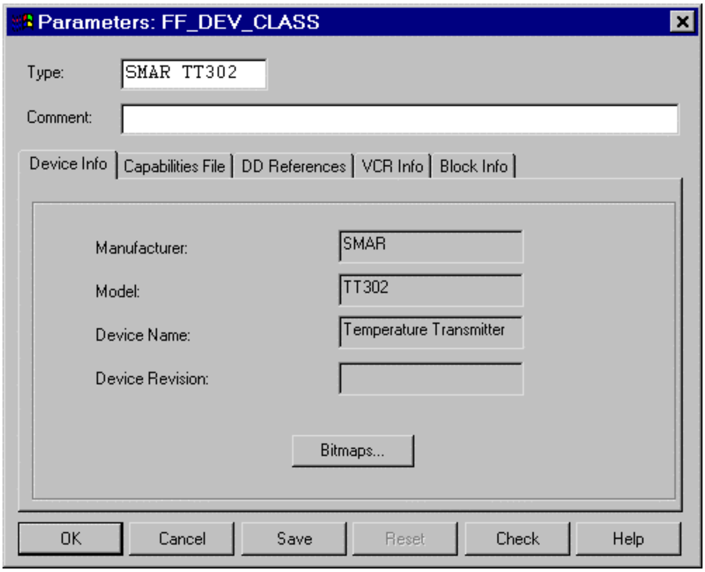
5.4.8 FF device class object, FF_DEV_CLASS

The **header area** of the parameter dialog contains the following data.

Type Name of the **FF device class** (max. 12 characters). This name is defined during the FF device import, and can be changed later. It must be unique within the project. This name is displayed in the FF device library and other places.

Comment Free comment text (max. 42 characters).

5.4.8.1 Device info tab



Parameters: FF_DEV_CLASS

Type: SMAR TT302

Comment:

Manufacturer: SMAR

Model: TT302

Device Name: Temperature Transmitter

Device Revision:

ff045us.bmp

Manufacturer Name of manufacturer from the device description.

Model Name of device model from the device description.

Device name Name of device from the device description.

Device revision Revision of device from the device description.

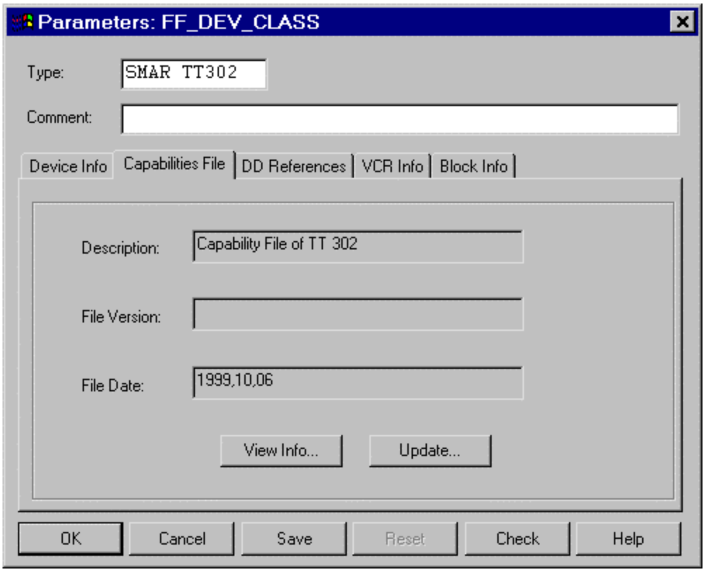
BITMAPS
This button is used to configure the graphical representation of device instances. The following parameters can be set:

- Representation of the field device in the detail view for the FIO-100 H1 link object.
- Representation of the field device in the detail view for the device instance object.

☞ For details of the allocation of bitmaps see **Page 36**, **Bitmaps for field devices**.

5.4.8.2 Capabilities file tab

This tab contains data for identifying the capabilities file.



Parameters: FF_DEV_CLASS

Type: SMAR TT302

Comment:

Description: Capability File of TT 302

File Version:

File Date: 1999,10,06

ff046us.bmp

Description Descriptive text for the capabilities file.

File version Version of the capabilities file.

File date Creation date of the capabilities file.

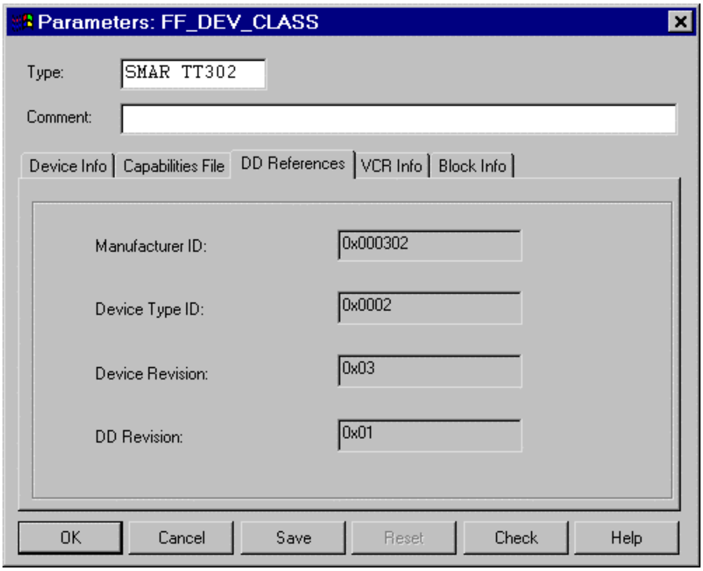
VIEW INFO The content of the capabilities file is displayed in a tree structure.

UPDATE An up to date revision of the capabilities file is imported.

☞ Updating the capabilities file will result in inplausible device instances and associated block instances in the project.

5.4.8.3 Device description tab

This tab contains identification data from the device description.



Parameters: FF_DEV_CLASS

Type: SMAR TT302

Comment:

Manufacturer ID: 0x000302

Device Type ID: 0x0002

Device Revision: 0x03

DD Revision: 0x01

ff047us.bmp

Manufacturer ID Internationally unique identification number for the device manufacturer. The manufacturer IDs are administered by the Fieldbus FOUNDATION.

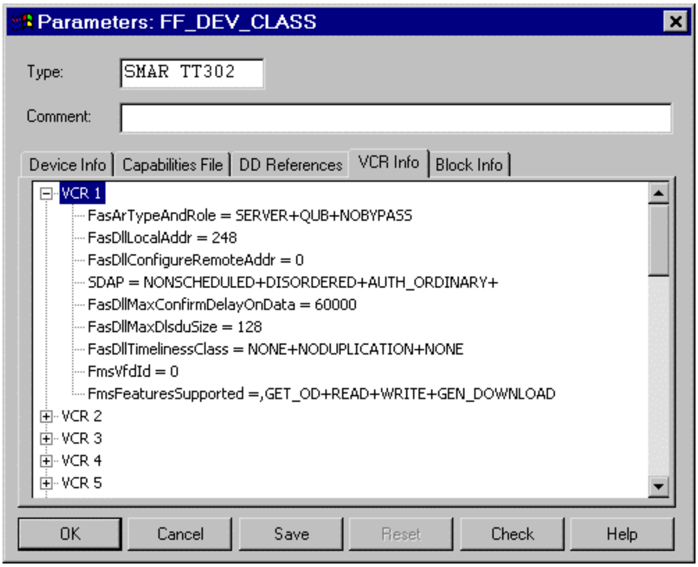
Device type ID Identification number for the device type. The device type ID is assigned by the device manufacturer.

Device revision Revision of the device.

DD revision Revision of the device description.

5.4.8.4 VCR info tab

This tab contains the pre-defined communication relationships (VCR – Virtual Communication Relationship) in the device.



Parameters: FF_DEV_CLASS

Type: SMAR TT302

Comment:

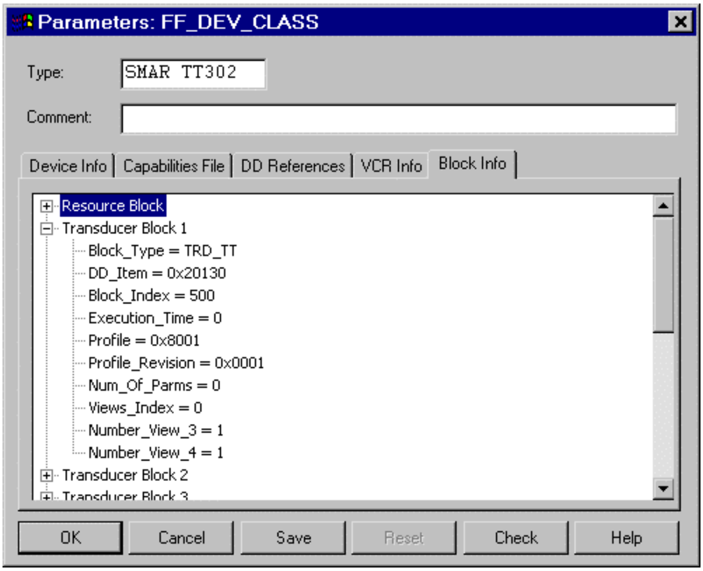
Device Info | Capabilities File | DD References | VCR Info | Block Info

```
⊟ VCR 1
    FasArTypeAndRole = SERVER+QUB+NOBYPASS
    FasDllLocalAddr = 248
    FasDllConfigureRemoteAddr = 0
    SDAP = NONSCHEDULED+DISORDERED+AUTH_ORDINARY+
    FasDllMaxConfirmDelayOnData = 60000
    FasDllMaxDlsduSize = 128
    FasDllTimelinessClass = NONE+NODUPLICATION+NONE
    FmsVfdId = 0
    FmsFeaturesSupported =,GET_OD+READ+WRITE+GEN_DOWNLOAD
⊞ VCR 2
⊞ VCR 3
⊞ VCR 4
⊞ VCR 5
```

ff067us.bmp

5.4.8.5 Block info tab

This tab contains a list of the blocks provided by the device with their most important identification data.



Type: SMAR TT302

Comment:

- Resource Block
- Transducer Block 1
  - Block_Type = TRD_TT
  - DD_Item = 0x20130
  - Block_Index = 500
  - Execution_Time = 0
  - Profile = 0x8001
  - Profile_Revision = 0x0001
  - Num_Of_Parms = 0
  - Views_Index = 0
  - Number_View_3 = 1
  - Number_View_4 = 1
- Transducer Block 2
- Transducer Block 3

ff049us.bmp

5.4.9 FF function block instance object, FF_FB

Handling the parameter dialogs

Using the standard function block "Analog Input" as an example, the basic layout of the parameter dialogs and the elements used in them is explained below:

General data Name, short text and long text for the FF function block.

Input field background color

Red: Mandatory parameter
Blue: Marked for overwriting

Text field E.g. for entering the block name and long text.

☞ The block name can also be selected using function key F2 from the tag list.

The optional parameters Short Text and Long Text cannot be entered before a block name has been assigned.

Table

Used for displaying and, if appropriate, inputting parameters such as lower and upper range limits. The facility for entering input and output parameter values is provided irrespective of whether or not a signal flow line is connected to the assigned function block pin.

Table background color

Grey: Display only; i.e. value **cannot be edited**.
White: Input; i.e. value **can be edited**.
Green: **Structural element**. By clicking on the check box the data structure can be expanded or collapsed.
□ The data structure is collapsed.
☑ The data structure is expanded.
Yellow (only in commissioning mode): The current value read from the device **is different from** the configured value.
Red (only in commissioning mode): **Communication is interrupted**; an error code is displayed.

List

There are lists in which only the default list entry is visible. The invisible part of the list can be expanded. The desired list entry is accepted by clicking in the input field.

☞ Virtual keys and buttons see **Chapter General Hints.**

☞ An FF function block is assigned by default the initial instance parameters that were present at the time of instancing on the FF function block class.

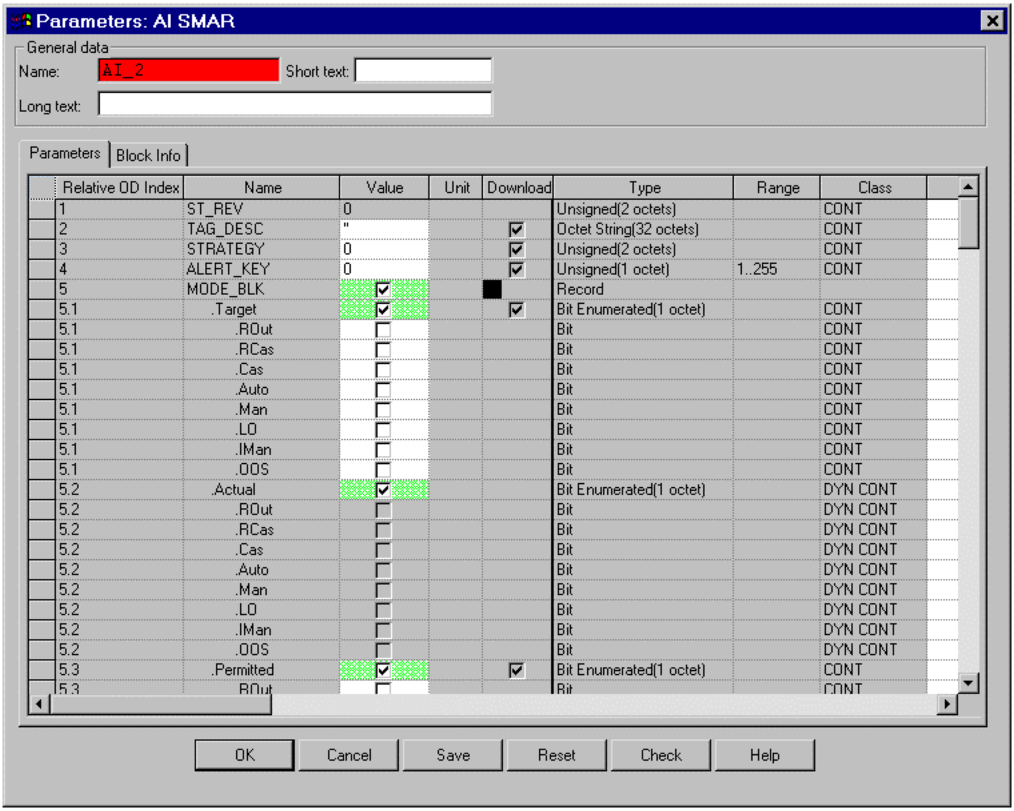
☞ To increase transparency, it is recommended that you enter short and long texts for the FF function blocks.

☞ Short and long text can only be entered after the function block name has been assigned.

5.4.9.1 Parameters tab

This tab displays the block parameters in tabular form. These include inputs, outputs and internal parameters.

With regard to the names and the meaning of the block parameters you should refer to the FF specifications *FF-890*, *FF-891* and *FF-892* and to the documentation provided by the device manufacturers.



General data

Name: AI_2

Short text:

Long text:

Relative OD Index	Name	Value	Unit	Download	Type	Range	Class
1	ST_REV	0			Unsigned(2 octets)		CONT
2	TAG_DESC	"		☑	Octet String(32 octets)		CONT
3	STRATEGY	0		☑	Unsigned(2 octets)		CONT
4	ALERT_KEY	0		☑	Unsigned(1 octet)	1..255	CONT
5	MODE_BLK	☑		■	Record		
5.1	.Target	☑		☑	Bit Enumerated(1 octet)		CONT
5.1	.ROut	☐			Bit		CONT
5.1	.RCas	☐			Bit		CONT
5.1	.Cas	☐			Bit		CONT
5.1	.Auto	☐			Bit		CONT
5.1	.Man	☐			Bit		CONT
5.1	.LO	☐			Bit		CONT
5.1	.IMan	☐			Bit		CONT
5.1	.OOS	☐			Bit		CONT
5.2	.Actual	☑			Bit Enumerated(1 octet)		DYN CONT
5.2	.ROut	☐			Bit		DYN CONT
5.2	.RCas	☐			Bit		DYN CONT
5.2	.Cas	☐			Bit		DYN CONT
5.2	.Auto	☐			Bit		DYN CONT
5.2	.Man	☐			Bit		DYN CONT
5.2	.LO	☐			Bit		DYN CONT
5.2	.IMan	☐			Bit		DYN CONT
5.2	.OOS	☐			Bit		DYN CONT
5.3	.Permitted	☑		☑	Bit Enumerated(1 octet)		CONT
5.3	.ROut	☐			Bit		CONT

ff050us.bmp

The table contains the following columns:

Relative OD index The relative OD index specifies the relative position of the parameter in the object dictionary. It is required, for example, to localize errors when an H1 link is loaded.

Name Name of the parameter.

Value Value of the parameter. In commissioning mode a choice can be made here to display either the configured value from the database or the current value read from the device.

In the case of data type *Bit Enumerated* , the values of the individual bits are displayed by means of the state of a check box:
□ Bit is reset; value = 0.
☑ Bit is set; value = 1.

Unit Physical unit of the parameter.

Download This entry displays whether or not the parameter is downloaded with the device configuration data.

□ The parameter is not downloaded to the FF device.
☑ A present parameter value is written to the FF device, provided that it is a writable parameter. If no parameter value is present in the value field, this parameter is not downloaded with the device configuration data.

In the case of a structural element the selection state of the data structure is displayed.

■ All elements of the data structure are selected for downloading.
◩ Some elements of the data structure are selected for downloading.
□ No element of the data structure is selected for downloading.

Type The data type describes the format of the value.

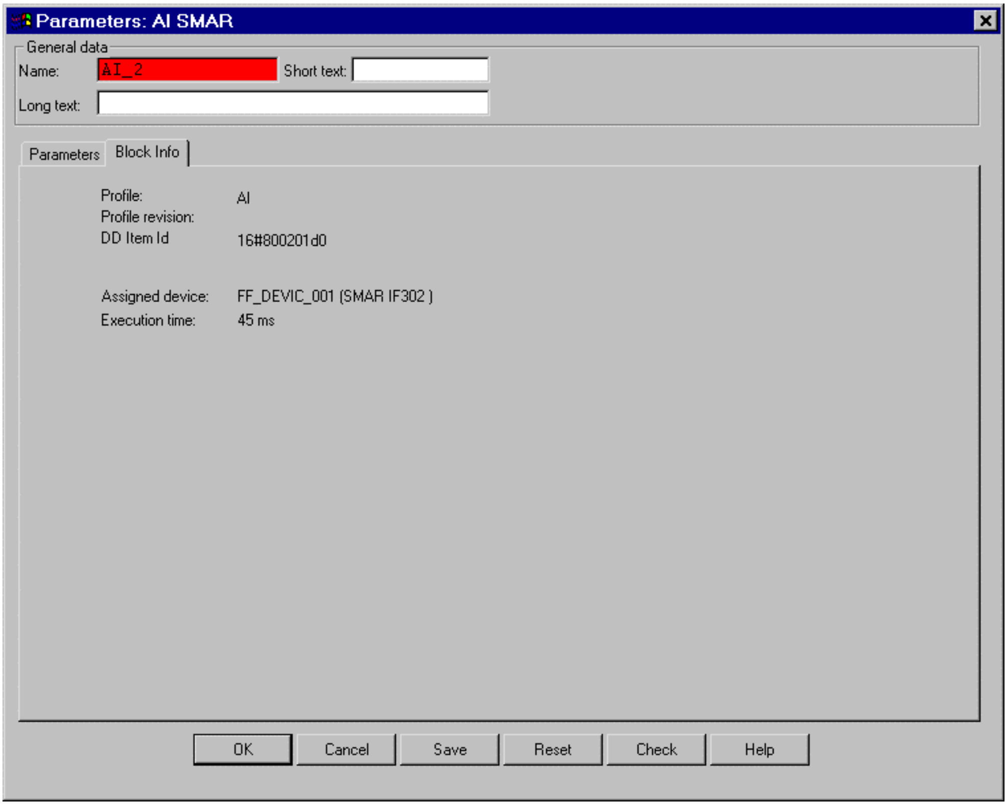
Range Permissible value range.

Class The class determines how the parameter is used in the field device. Each parameter belongs to one and only one of the classes INPUT, OUTPUT or CONTAINED. In addition there are the optional classes DYNAMIC, DIAGNOSTIC, SERVICE, OPERATE, ALARM, TUNE and LOCAL.

Comment Optional entry of free comment text.

5.4.9.2 Block Info tab

This tab contains identification data and characteristics for the block.



Parameters: AI SMAR

General data

Name: AI_2

Short text:

Long text:

Profile: AI

Profile revision:

DD Item Id 16#800201d0

Assigned device: FF_DEVIC_001 (SMAR IF302)

Execution time: 45 ms

ff051us.bmp

Profile Profile on which the block definition is based. Amongst other things the profile determines whether a function block is standard, enhanced or manufacturer-specific. The profile number for standard function blocks is assigned by the Fieldbus FOUNDATION .

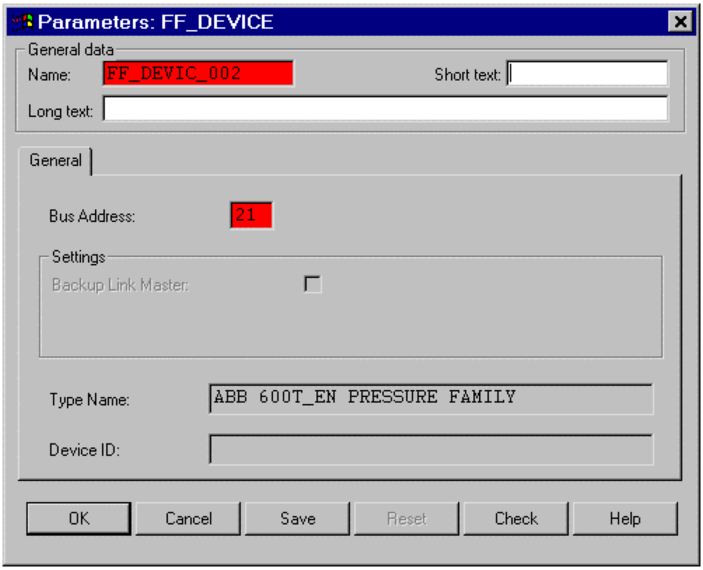
Profile revision Revision of the block profile. Amongst other things this determines whether a function block is standard or enhanced.

DD item ID Identification number of the block within the device description.

Assigned device Device in which the block is executed.

Execution time Time taken for execution of the block.

5.4.10 FF device instance object, FF_DEVICE


Parameters: FF_DEVICE

General data

Name: FF_DEVIC_002

Short text:

Long text:

General

Bus Address: 21

Settings

Backup Link Master: □

Type Name: ABB 600T_EN PRESSURE FAMILY

Device ID:

ff053us.bmp

Bus address Address of the device in the H1 link.
The value range is 20..247.

Backup Link Master Determines whether or not the device attempts to take over the role of LAS (Link Active Scheduler) if necessary.

□ The device does not attempt to take over the LAS role.

☑ The device attempts to take over the LAS role.

This check box is enabled only for devices that are capable of functioning as a Link Master; otherwise it is displayed in grey.

Operational Powerup

System management parameter which determines the state which the device takes on when it is switched back on again following a failure of the operating power.

□ After the operating power is switched back on, the device takes on the state *INITIALIZED* . In this state, the device has the default bus address. It does not participate in the normal bus communication. Not until a bus address is assigned is the state *SM_OPERATIONAL* achieved, which enables the device to participate in normal bus communication and to start the user application.

☑ After the operating power is switched back on, the device takes on the state *SM_OPERATIONAL*. It thus participates in the normal bus communication, and the user application starts in the manner configured.

Type name Device type.

Device ID Unique identification number for this device. The device manufacturer assigns a unique device ID to each individual device, and this ID never changes.

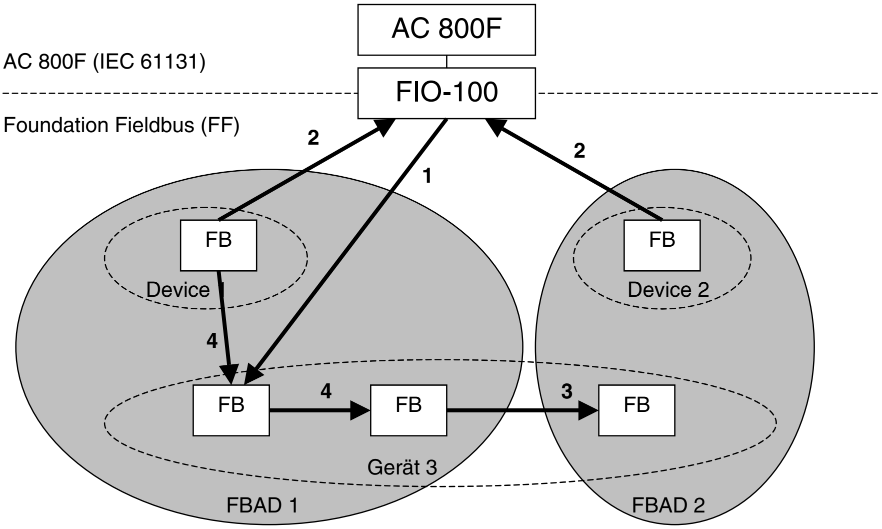
5.5 FF variables

There are two types of FF variables:

- **I/O FF variables** are used primarily for configuring the exchange of data between the FF subsystem and the AC 800F. An I/O FF variable whose source lies in the AC 800F is an **input for the FF subsystem** and an output for the AC 800F. An I/O FF variable whose source lies in the FF subsystem is an **output for the FF subsystem** and an input for the AC 800F.
- **Local FF variables** are used exclusively for configuring the **communication within the FF subsystem**. They are required for exchanging data between different FBADs. As an option, they can be used in place of a signal line in the FBAD.

☞ I/O FF variables with the supplement "(LP)" (low priority) are communicated with a transfer cycle that has been configured for the H1 link (see **Page 46**, **Communication tab**). They are a means for reducing the communication load on the H1 link and the system load in the AC 800F. Use of this variable type is recommended for variables that are used for data exchange with Operate^{IT} and for which a longer transmission cycle time is sufficient.

☞ An I/O FF variable whose source lies in the AC 800F is read-only in the FF subsystem; but can be read more than once. An I/O FF variable whose source lies in the FF subsystem must be written once in the FF subsystem and can be read more than once here.



1 I/O FF variable (input for FF subsystem)
2 I/O FF variable (output for FF subsystem)
3 Local FF variable (data exchange between FBADs)
4 Signal line in FBAD; optional: local FF variable

ff014us.wp

All the FF variables in an H1 link are entered by the system into the **FF variables list** for that link.

When an I/O FF variable is created in the FF variables list, two **I/O components** are generated automatically for the AC 800F. One of these components represents the **value** of the FF variable, and the other component represents the **status**. The component name is produced from the name of the FF variable by appending the ending „_V“ for the value or „_S“ for the status.

5.5.1 FF variables list

It contains all the FF variables of an H1 link. For each FF variable the **variable name**, the **data type** (analog, discrete or bitstring), the **usage** (IEC 61131->FF, FF->IEC 61131 or FF local) and an optional **comment** are displayed.

5.5.1.1 Call FF variables list

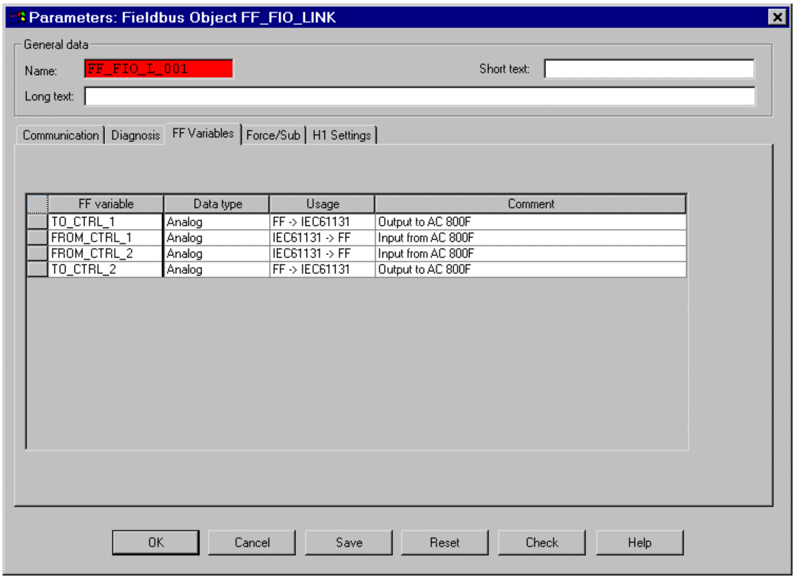
The FF variables list can be edited in the parameter dialog of the FIO-100 H1 link object on the *FF variables* tab.

→ Select FF FIO-100 H1 link object (FF_FIO_LINK) in the **hardware structure**
→ *Edit*
→ *Parameters*
→ Select FF variables tab

New FF variables can be created here, or existing ones modified or deleted.

5.5.1.2 Structure of the FF variables list

The structure of the FF variables list is as follows:



Parameters: Fieldbus Object FF_FIO_LINK

General data

Name: FF_FIO_L_001

Short text:

Long text:

FF variable	Data type	Usage	Comment
TO_CTRL_1	Analog	FF -> IEC61131	Output to AC 800F
FROM_CTRL_1	Analog	IEC61131 -> FF	Input from AC 800F
FROM_CTRL_2	Analog	IEC61131 -> FF	Input from AC 800F
TO_CTRL_2	Analog	FF -> IEC61131	Output to AC 800F

ff030us.bmp

FF variable Name of FF variable (max. 14 characters). This name is displayed in the FF variable symbol in the FBAD. For I/O FF variables, the names of the automatically-generated components for inputs and outputs of the AC 800F are derived automatically from this name.

Data type Data type of the FF variable, such as analog, discrete or bitstring.

Usage This determines whether the FF variable is used for data exchange with the AC 800F (I/O FF variable) or whether it is used exclusively for communication within the FF subsystem (local FF variable). The following values can be entered:

FF local local FF variable.

FF -> IEC61131 I/O FF variable whose source lies in the FF subsystem.

FF -> IEC61131 (LP) I/O FF variable whose source lies in the FF subsystem; uses the transfer cycle for low-priority I/O FF variables which is configured for the H1 link.

IEC61131 -> FF I/O FF variable whose source lies in the AC 800F.

IEC61131 -> FF (LP) I/O FF variable whose source lies in the AC 800F; uses the transfer cycle for low-priority I/O FF variables which is configured for the H1 link.

Comment Comment on the FF variable (max. 40 characters).

5.5.1.3 Edit list entries

A number of functions are provided for editing individual list entries. These can be used to add new entries, modify and delete existing entries.

Add new FF variable to list

🖱 → Shortcut menu → *Create component* or INS key

A new line is added to the FF variables list. The following values should be entered here:

FF variable Enter the variable name (max. 14 characters).

Data type Select the data type from the data types list.

Usage Select usage from the selection list.

Comment Optional entry of free comment text (max. 40 characters).

Change FF variable entry

🖱 → Select the required field with a left mouse click

Depending on which field is selected, the new value can be either entered directly or chosen from a selection list.

☞ If an FF variable is renamed, then the places in an FBAD at which that FF variable is used are represented with the status *Error present* (red).

Delete FF variable

🖱 → Select the desired line in the list by clicking the left mouse button on the line select field.
→ Shortcut menu → *Delete component* or DEL key

☞ The places in an FBAD at which the deleted variable is used are represented with the status *Error present* (red).

5.5.1.4 Block editing

The editing steps described below are carried out with a block. A block is a number of complete lines in a list that are selected. All the lines in a block do not necessarily have to be adjacent to one another. In this case a block consists of several partial blocks.

Select block

🖱 → Left click on the desired start of block
→ Keeping the left mouse button pressed down, drag the mouse across the block area up to the end of the first partial block
→ Release left mouse button
→ If necessary, hold down the INS key and select additional partial blocks.

Cut

🖱 → Select block → Shortcut menu: *Cut*

A defined block is removed from the FF variables list and stored on the clipboard. Using the command *Paste*, this block can be inserted in an FF variables list.

Copy

🖱 → Select block → Shortcut menu: *Copy*

A defined block is copied and stored on the clipboard. Using the command *Paste*, this block can be inserted in an FF variables list.

Paste

🖱 → Select block → Shortcut menu: *Paste*

A block in the clipboard that has been copied or cut is inserted in a FF variables list at the point indicated by the cursor.

☞ If the names of the FF variables being inserted already exist in the FF variables list, when they come to be saved a prompt appears requesting that you should change the duplicated variable names.

Delete

→ Select block → Shortcut menu: *Delete component* or DEL key

A defined block is deleted from the FF variables list.

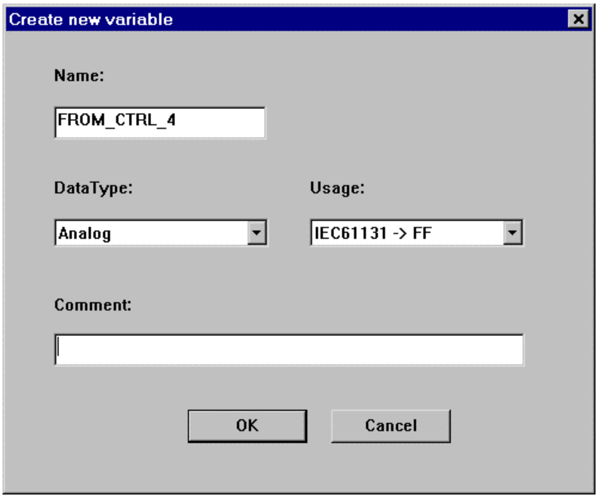
☞ The places in an FBAD at which the deleted variable is used are represented with the status *Error present* (red).

5.5.1.5 Create new FF variable in the FBAD

An alternative method of adding new variables to the FF variables list is offered by the FBAD editor. Access to the FBAD editor is from the parameter dialog that can be called up for an FF variable symbol.

→ Double-click on an FF variable symbol in the FBAD editor
→ Enter the name of a new FF variable in the *Variable* field

If the name of an FF variable that is not contained in the FF variables list is entered in the parameter dialog, the following dialog for creating a new FF variable will appear.



Create new variable

Name: FROM_CTRL_4

DataType: Analog

Usage: IEC61131 -> FF

Comment:

ff033us.bmp

The following values should be entered here:

Name	Enter the variable name (max. 14 characters).
Data type	Select the data type from the data types list.
Usage	Select usage from the selection list.
Comment	Optional entry of free comment text (max. 40 characters).

The FF variable that is defined in this manner is included in the FF variables list for the H1 link.

5.5.2 I/O editor for FF objects

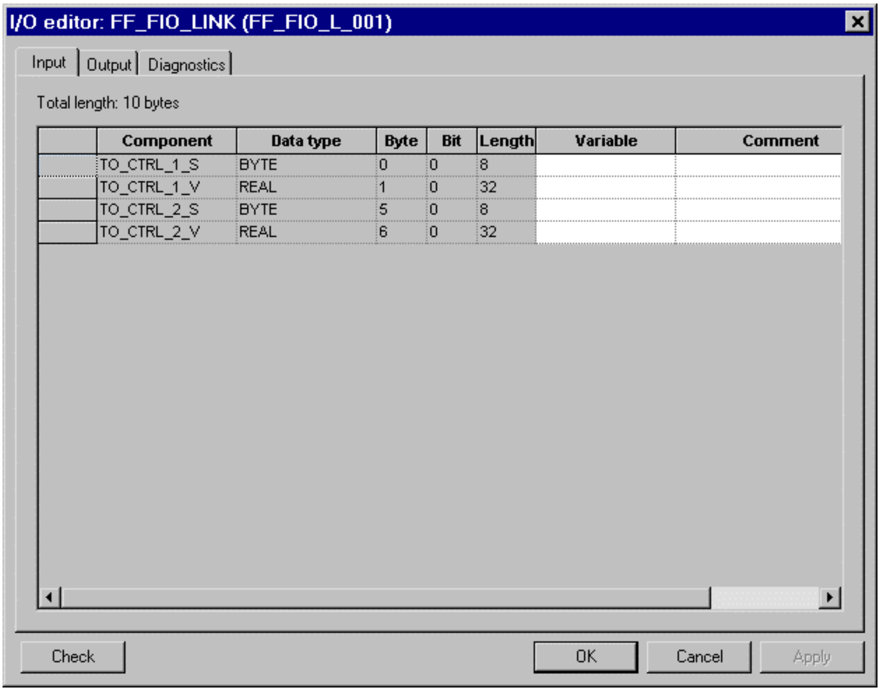
The I/O editor can be used to edit the components for inputs and outputs that were created automatically by creating I/O FF variables, and the components for diagnostic data created by the system. The editing options here are limited to optionally assigning a **variable** (in the IEC 61131 system) and entering an optional **comment** for each component.

The defined input, output and diagnostic data can be accessed by two different ways:

- All components can be accessed directly for reading or writing with <H1 link tag name>.<component name>.
- If a variable has also been defined for a component, then the input, output or diagnostic data can also be read or written via the variable.

→ Select FF FIO-100 H1 link → *I/O editor*

The I/O editor contains separate tabs for inputs, outputs and diagnostic data.



I/O editor: FF_FIO_LINK (FF_FIO_L_001)

Total length: 10 bytes

	Component	Data type	Byte	Bit	Length	Variable	Comment
	TO_CTRL_1_S	BYTE	0	0	8		
	TO_CTRL_1_V	REAL	1	0	32		
	TO_CTRL_2_S	BYTE	5	0	8		
	TO_CTRL_2_V	REAL	6	0	32		

ff016us.bmp

Each tab shows the total data length that is transferred in the data block between the AC 800F and the FIO-100.

Component Name of component. For inputs and outputs this name is generated automatically from the name of the underlying FF variable. The component representing the **value** of the FF variable obtains its name by appending the ending "_V“ to the FF variable name, while the component representing the **status** obtains its name by appending "_S“. The component name cannot be changed. For diagnostic data, the component name is fixed in advance by the system.

Data type AC 800F data type of the component - BYTE, REAL, WORD etc.. It is derived from the data type of the underlying FF variable, and cannot be changed. The AC 800F data type defines the length of the component. The complete length of the AC 800F data type is represented in the data block transferred between AC 800F and FIO-100.

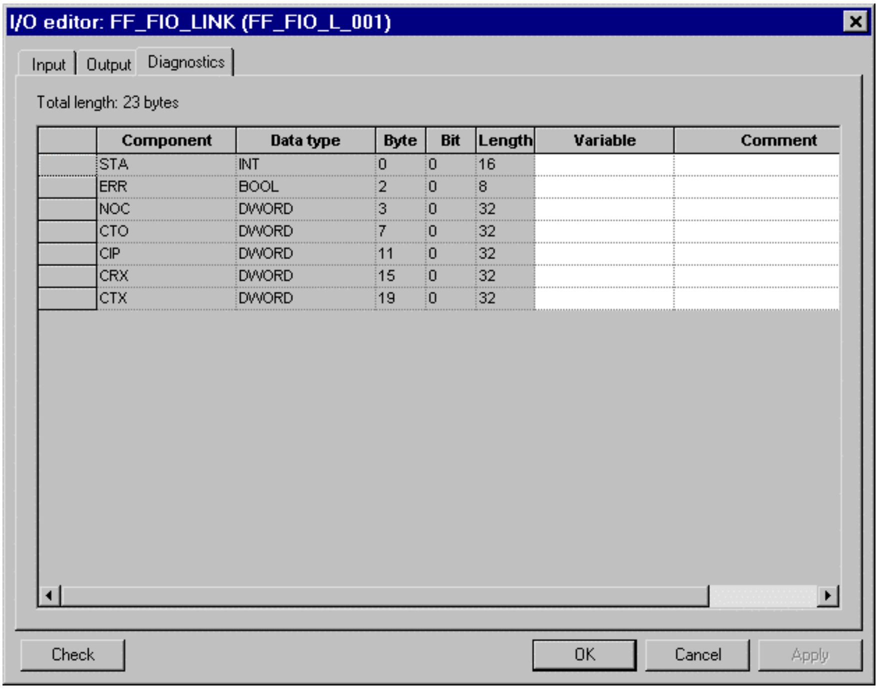
Byte, Bit, Length These values describe the position of the AC 800F data type in the data block that is transferred between AC 800F and FIO-100. This data cannot be changed.

Variable Each component may have a variable assigned to it. The variable can be defined in the I/O editor or selected from the list of already defined variables. The data type of the variable must match with the data type of the component. The assignment of a variable is optional.

Comment Comment to describe the component. Assignment of a comment is optional.

☞ The I/O editor shows the variables exchanged between the AC 800F and the FF subsystem from the viewpoint of the AC 800F. A variable whose source lies in the AC 800F is an output for the AC 800F. A variable whose source lies in the FF subsystem is an input for the AC 800F.

5.5.2.1 Diagnosis



Total length: 23 bytes

	Component	Data type	Byte	Bit	Length	Variable	Comment
	STA	INT	0	0	16		
	ERR	BOOL	2	0	8		
	NOC	DWORD	3	0	32		
	CTO	DWORD	7	0	32		
	CIP	DWORD	11	0	32		
	CRX	DWORD	15	0	32		
	CTX	DWORD	19	0	32		

ff017us.bmp

This tab contains diagnostic data relating to the communication between the controller and the FIO-100.

Component	Explanation
STA	Error code for the latest error to occur. If ERR has the value TRUE, this is the error code for the error that is currently present. 1 Internal error 2 FIO-100 has no configuration data 3 The current configuration of the FIO-100 does not correspond to the data in the database 4 FIO-100 declines to establish a connection with the controller 5 Internal error 6 The current configuration of the FIO-100 does not correspond to the data in the database
ERR	TRUE, if an error is present
NOC	Number of failed connection attempts
CTO	Number of timeouts
CIP	Number of datagrams received (UDP)
CRX	Number of data packets received
CTX	Number of data packets sent

5.6 FBAD editor

5.6.1 Function block application diagram

To configure the function block application process described in the FF specification, Control Builder F uses the **function block application diagram (FBAD)**. This is generated using the FBAD editor. An FBAD comprises the function block application process of an FF H1 link, or part of it.

In accordance with the FF specification, the function block application process is described with the aid of function blocks. Function blocks create for themselves repetitive time-critical functions as paramaterisable algorithms which convert input parameters into output parameters taking into account the specified operating mode and other internal parameters. For this purpose, Control Builder F uses **FF function blocks**, which are placed in an FBAD. All the FF function blocks from an FBAD are processed using the **same cycle time**. This cycle time is assigned to the FBAD. Parameters for the block algorithm are set using the parameter dialog for the FF function block; this dialog can be called from the FBAD editor.

Function blocks are linked with one another by linking the output of one function block with the input of another. The linked function blocks may be located either on the same device or different devices. In the FBAD the connections are represented by signal flow lines. Signal connections between different FBADs are modelled using **local FF variables**. **I/O FF variables** are used for signal exchange with the AC 800F.

In addition to the function blocks and link objects, the FF specification also provides for the use of **resource and transducer blocks** for modelling a function block application process. These device-related blocks do not appear in the FBAD editor. The parameter dialogs for these blocks can be called up using the corresponding device instance objects in the hardware structure.

This version of Control Builder F does not support the **trend, alarm and display objects** provided in the FF specification for modelling a function block application process.

The FBAD thus comprises the following **graphical elements**:

- FF function blocks,
- Signal flow lines and
- FF variables.

The work area is laid out over 10 x 10 pages. The separate pages can be reached scrolling horizontally and vertically. The entire work area is provided with a grid. The page breaks are indicated by a dashed line. When the program documentation is printed out page-by-page, each page exactly reflects a page in the work area.

The signal flow in an FBAD passes from a reading FF variable or from an FF function block output to a writing FF variable or a function block input. The signal flow lines are edited either by holding the CTRL key down or by activating "line drawing“ mode.

The processing sequence of the FF function blocks is determined automatically according to the signal flow, and can subsequently be changed using the schedule editor. See **Page 105**, **Schedule editor**.

FF variables can be declared in the FBAD editor and then added to the FBAD. FF variables that have already been declared can be selected from the FF variables list for the FIO-100 H1 link and added to the FBAD. See also **Page 72**, **FF variables**.

5.6.1.1 Create FBAD

A new FBAD is created in the **hardware structure**. FF function block application diagram objects (FF_FBAPP_D) can be added there underneath a schedule object.

→ Select FF schedule object (FF_SCHEDULE → *Edit* → *Insert*
→ Select FF function block application diagram object (FF_FBAPP_D)

Each new FBAD has an empty graphics area and a default cycle time of 500 ms. Name, short text and long text contain null strings. The editing status is **incorrect.**

5.6.1.2 Edit FBAD

The FBAD editor is the detail view of each FF function block application diagram object (FF_FBAPP_D). An FBAD is therefore called up for editing by simply selecting the corresponding object in the **hardware structure**. When an FBAD is called up, it is displayed with its content (function blocks, signal flow lines etc.) and can be modified.

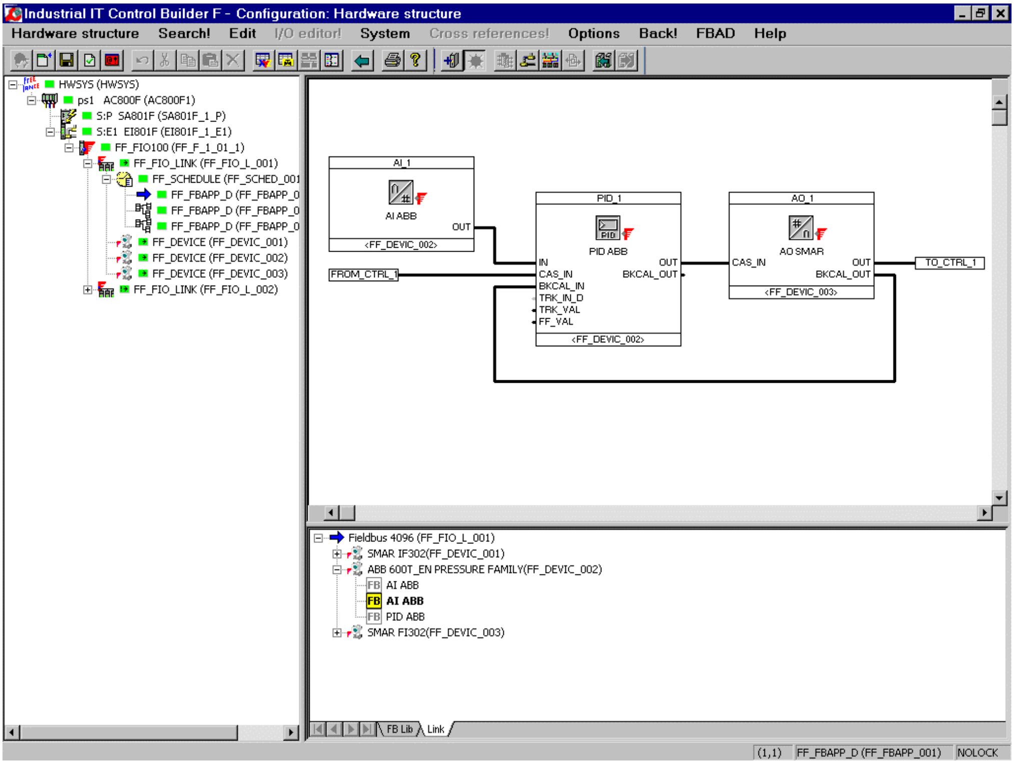
→ Select FF function block application diagram object (FF_FBAPP_D) in the **hardware structure**

5.6.2 FBAD editor interface

5.6.2.1 Structure of the configuration interface

The configuration interface of the FBAD editor comprises two windows: the **graphics area** and the **function block selection area.**

The **status line** at the bottom of the screen displays the current FBAD status.



ff012us.bmp

FF function blocks and FF variables are placed in the **graphics area** and linked by means of signal flow lines. The graphics area is provided with a grid to enable the elements to be positioned easily and for minimum clearance between elements to be maintained. Graphic elements can only be positioned on this grid in the graphics area. The grid display can be switched on or off.

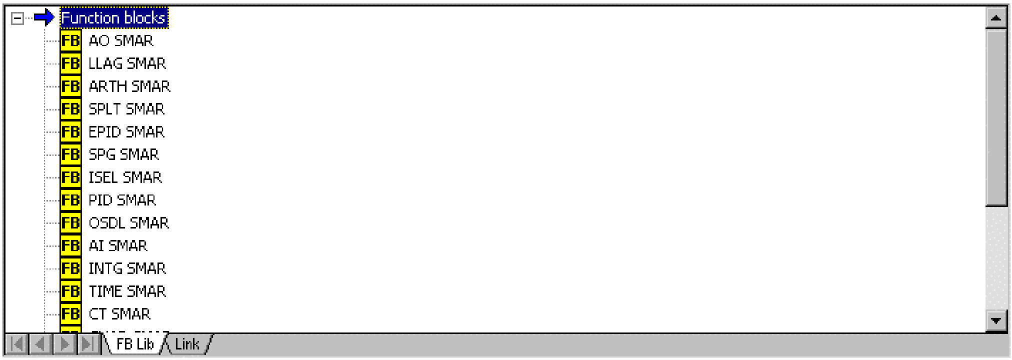
An FBAD can be up to 10 x 10 pages in size. The separate pages are divided by dashed lines. Care should be taken to avoid positioning any objects on the dashed lines as they would then be split up over two pages when the documentation is printed.

The **function block selection area** is used for selecting the FF function blocks that are to be placed in the FBAD. It comprises two tabs which enable FF function blocks to be selected from **library view** or **link view**.

FB Lib tab

If an FF function block is to be added to an FBAD, the required **FF function block type** can be chosen from the library view. The library view contains all the FF function block types that can be used in the project, and is structured in the same way as the FF block library.

If the selection is made from the library view, then the FF function block that is placed in the FBAD is not yet assigned to a device.



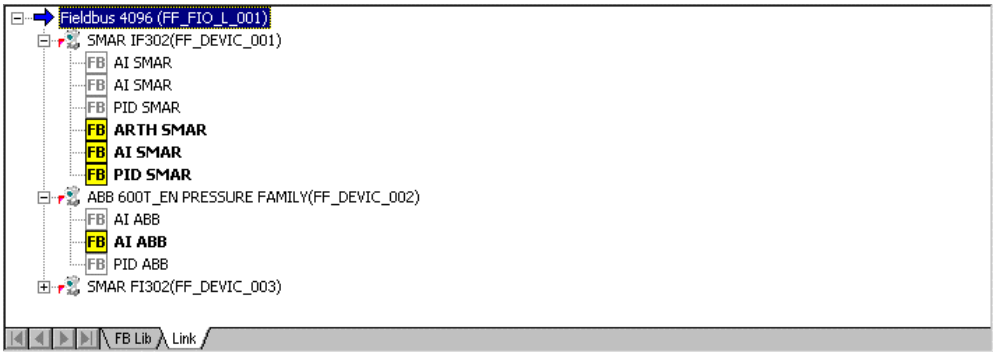
Function blocks
- FB AO SMAR
- FB LLAG SMAR
- FB ARTH SMAR
- FB SPLT SMAR
- FB EPID SMAR
- FB SPG SMAR
- FB ISEL SMAR
- FB PID SMAR
- FB OSDL SMAR
- FB AI SMAR
- FB INTG SMAR
- FB TIME SMAR
- FB CT SMAR

ff034us.bmp

Link tab

The link view offers an alternative means of selection when adding an FF function block to an FBAD. The required **FF function block instance** from a device configured in the H1 link can be selected here. For this purpose, the link view shows the devices configured in an H1 link in a tree structure along with their function blocks. The graphical representation of the function blocks differentiates between those that have been assigned and those that have not been assigned.

When selected from the link view, the FF function block that is placed in the FBAD is automatically assigned to a device.



Fieldbus 4096 (FF_FIO_L_001)
- SMAR IF302(FF_DEVIC_001)
  - FB AI SMAR
  - FB AI SMAR
  - FB PID SMAR
  - FB **ARTH SMAR**
  - FB **AI SMAR**
  - FB **PID SMAR**
- ABB 600T_EN PRESSURE FAMILY(FF_DEVIC_002)
  - FB AI ABB
  - FB **AI ABB**
  - FB PID ABB
- SMAR FI302(FF_DEVIC_003)

ff035us.bmp

5.6.2.2 Changing the defaults

Showing / hiding the grid

🖱 → *FBAD* → *Raster*

The positioning grid in the graphics area is shown if it was previously hidden, and vice versa. When this setting is changed, the changed setting remains in force until a different window is opened.

Change colors of FBAD

🖱 → *FBAD* → *Color*
→ Select object whose color is to be changed
(for example the color for the function block border)
→ Select desired color

5.6.3 Description of the FBAD elements

5.6.3.1 Signal flow lines

Horizontal and vertical connection lines can be attached to FF variables and FF function block pins. These **signal flow lines** represent the signal flow in an FF function block application process. Signal flow lines always join a **signal source** (e.g. reading FF variable or FF function block output) with **one or more signal sinks** (e.g. writing FF variable or FF function block input).

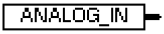
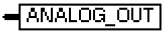
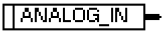
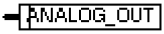
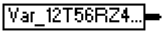
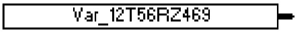
5.6.3.2 FF variables

FF variables can be placed anywhere in the FBAD, and are displayed and/or edited in a rectangle.

There are different graphical representations for **I/O FF variables** and **local FF variables**. The symbol for a local FF variable is distinguished by an additional vertical line on the left-hand edge of the symbol.

The graphical representation also enables an FF variable that is to be **read** to be distinguished from an FF variable that is to be **written**. The symbol for an FF variable to be read has an output pin on the right-hand side of the symbol. The symbol for an FF variable to be written has an input pin on the left-hand side of the symbol.

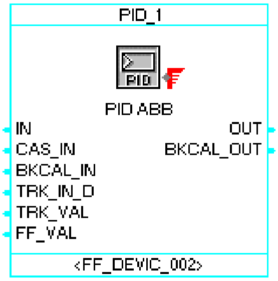
Furthermore, there are both **short** and **long** versions of the graphical representation of an FF variable. The short version can display 10 characters. If the label is more than 10 characters in length, the overflow is represented by '...'. The long version can accommodate the greatest label length that is possible.

Symbol	Description/function:
	**I/O FF variable for reading**
	**I/O FF variable for writing**
	**Local FF variable for reading**
	**Local FF variable for writing**
	**Short version** Max. 10 characters can be displayed, overflow indicated by '...'
	**Long version** Max. possible label length

A red FF variable symbol indicates that the FF variable assigned to this incidence point does not exist, e.g. because it has been renamed or deleted in the FF variables list.

5.6.3.3 FF function blocks

The following illustration provides an example of the graphical representation of an FF function block.



ff025us.bmp

Block border — The block border delineates the selection area of the FF function block. Its **color** indicates whether or not the function block is selected. The colors used for this purpose can be changed. See **Page 85, Changing the defaults**.

Block name — The block name is the **tag name** of the FF function block. It is displayed in the header area of the graphical representation of every FF function block. All FF function block names are found again in the system-wide **tag list**.

Icon — An icon symbolizes the type of the FF function block. An FF standard function block and all the enhanced function blocks based on it are represented by the same icon. Custom function blocks are represented by a standard FF icon.

Type name — The type name shown below the icon **uniquely** identifies the **type of the FF function block** within a project. The type name is the name of the related FF function block class. It is defined during the FF device import and appears in the FF block library amongst other places.

Input/output pins — Input/output pins represent the input and output parameters of an FF function block. In accordance with the signal flow, the inputs are always on the left and the outputs always on the right. The **color and line width** of the input/output pins – as with the signal flow lines – provide information about the data type that is required or has been set.

Pin designation A code adjacent to each input/output pin in an FF function block **identifies** the pin within the block and reflects its function. Thus, for example, **IN** designates the primary input value for a function block. The pin designations of the input and output parameters of FF function blocks can be found in the FF specification and in the manufacturer's documentation.

Assigned device In the footer area of the graphical representation of an FF function block appears the **tag name of the device**, in which the function block is processed. Function blocks that have not yet been assigned to a device display the string **<?>**.

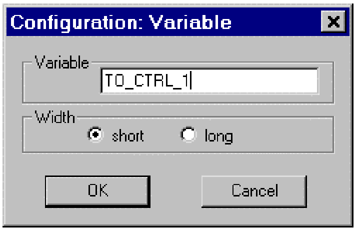
5.6.4 Set parameters for the FBAD elements

In order to set parameters for an FBAD element, the following steps must be taken.

→ Select FBAD element for which parameters are to be set
→ *Edit* → *Parameters*.

→ Double-click on the FBAD element

5.6.4.1 FF variables



Variable: TO_CTRL_1

Width: ● short ○ long

ff026us.bmp

Variable Name of FF variable.
An FF variable can be selected from the FF variables list by pressing F2. If, on the other hand, a variable name is entered which is not yet included in the FF variables list, then a dialog box is displayed for creating a new FF variable (see **Page 77**, **Create new FF variable in the FBAD**).

Width

Short The short version of FF variable is chosen, in which only 10 characters can be displayed. If the variable name is more than 10 characters in length, this is indicated by '...'.

Long The long version of FF variable is chosen, in which variable names of maximum length can be displayed.

5.6.4.2 FF function blocks

Parameter types

In this respect, parameters are defined as data that is necessary for a block to be processed and displayed in the system. Distinctions are made between the following types of parameter:

Mandatory Essential data such as the block name (tag name).

Optional Not necessarily essential data such as short text and long text.

External External parameters are passed to a function block by connecting a signal flow line and vice versa.

Internal Internal parameters are entered within a parameter dialog. These include details such as block name and limit values.

Calling up the parameter dialogs

→ Select the FF function block for which parameters are to be entered
→ *Edit* → *Parameters*

→ Double-click on the FF function block for which parameters are to be entered

Control passes to the first parameter dialog for an FF function block. Any other selected elements are now automatically deselected. After exiting from the parameter dialog, the FF function block whose parameters have been changed is re-displayed accordingly.

Entering mandatory parameters

To enable an FBAD to be signed off as correct, the **mandatory parameters** of the individual FF function blocks in the FBAD must be specified. All the mandatory parameters are displayed with a **red** background in the parameter dialogs. As a rule these are only the **block name** (max. 16 characters) of an FF function block.

All block names that are entered for FF function blocks are assembled system-wide in the tag list. See **Engineering Manual, IEC 61131-3 Programming Tags**.

Alternative facility for entering the block name

→ Select text field *Name:* → F2
→ Choose block name from tag list

Contents and interface of the parameter dialogs

See **Page 66**, **FF function block instance object, FF_FB**.

5.6.5 Editing an FBAD

5.6.5.1 Drawing signal flow lines

The FBAD editor has a special draw mode in which horizontal and vertical signal flow lines can be drawn.

Activating drawing mode:

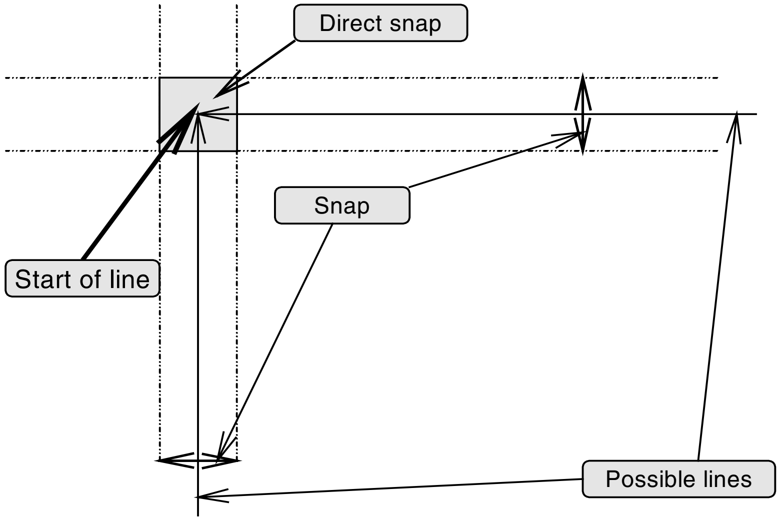
→ *FBAD* → *Draw lines*

→ Right mouse button (Shortcut menu) → *Draw lines*
(The mouse pointer changes to a cross)

A single click of the left mouse button marks the start of the line. When the mouse is moved, a horizontal or vertical line is drawn as long as the cursor is within the snap, and as long as the line does not cut into any function block or FF variable.

Another click of the left mouse button marks the end of the line and the start of a new line. A left click in the direct snap of the start of a line or outside the snap has the effect of ending a line.

The illustration below clarifies the drawing mode for drawing a line. The snap is exactly two grid units in width.



Pulling a line

→ With drawing mode activated: define start and end of line with a left mouse-click

→ With drawing mode not activated: define start and end of line with a left mouse-click while simultaneously holding down the CTRL key

Deactivating drawing mode:

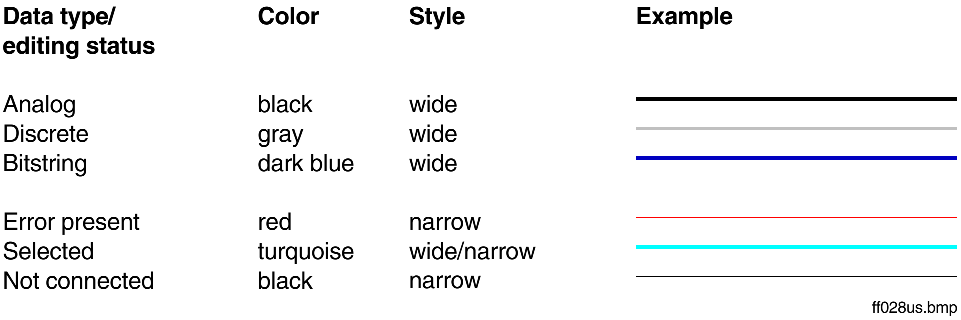
🖱 → Right click

⌨ → Esc key

Displaying the signal flow lines

A signal flow line indicates the transported data type. The editing status of the signal flow line, such as **error present**, **selected** or **not connected**, is also displayed.

The transported data type and the editing status of the signal flow line can be seen from the width and color of the line, and the colors can be set freely according to the user's preference (see **Page 85, Changing the defaults**). The relationship between data type, editing status, line width and the default color is shown by the following illustration:



Data type/ editing status	Color	Style	Example
Analog	black	wide	━━━━━
Discrete	gray	wide	━━━━━
Bitstring	dark blue	wide	━━━━━
Error present	red	narrow	─────
Selected	turquoise	wide/narrow	━━━━━
Not connected	black	narrow	─────

ff028us.bmp

5.6.5.2 Adding an FF variable to an FBAD

When an FF variable is added to an FBAD, first only an incidence point of the FF variable is added in the FBAD. Only if the FF variable to be used is not yet included in the FF variables list for the H1 link is the new FF variable automatically added to the FF variables list. See **Page 73**, **FF variables list**.

🖱 → *FBAD* → *Variable (for reading)* or *Variable (for writing)*

After the variable symbol that is to be added for a read or write FF variable has been chosen, the cursor is displayed as an outline of the FF variable. The variable can be placed freely (with a left mouse-click). If it is not possible to place the variable, it continues to be displayed with an outline and a warning tone is emitted.

If the placing was successful, the outline cursor continues to be displayed and further FF variables of the type just selected can be added. The insert operation is terminated by clicking the right mouse button.

Next, the parameter input dialog for this incidence point of the FF variable is called up with a double-click of the left mouse button on the variable symbol. See **Page 88**, **FF variables**.

FF variables that have already been defined can be selected directly from the FF variables list.

🖱 → F2 → Select an FF variable that already exists in the project from the list

If an FF variable is being used for the first time in the project, it is automatically included in the FF variables list for the FIO-100 H1 link.

☞ Multiple-write usage of the same FF variable is not permitted, and will result in an error.

5.6.5.3 Modifying an FF variable in an FBAD

Strictly speaking, only an **incidence point of an FF variable** can be modified in the FBAD. When so doing, if the name of an FF variable is added which is not yet included in the FF variables list for the FIO-100 H1 link, then this new FF variable is automatically added to the FF variables list. See **Page 73**, **FF variables list**.

🖱 → Double-click on the FF variable to be modified
→ Change FF variable name → Enter
or
→ F2 → select one of the existing FF variables in the H1 link in the "Select variable" window

☞ The old FF variable is retained in the FF variables list.

☞ If the modified FF variable is used in more than one place, the other incidence points remain unaffected by the change.

5.6.5.4 Adding an FF function block

To add an FF function block to the FBAD, the required function block is selected from the **function block selection area** and dragged to the graphics area of the FBAD (drag and drop). There are two ways of achieving this, provided by the **Library view** and **Link view** tabs respectively; these two methods differ both in procedure and result, and they are described below. In the one case, the function block that is added is not yet assigned to a device, while in the other case the device assignment already exists.

Selection from library view

→ Select the desired FF function block type from the library view
→ Move the selected element to the desired position in the graphics area by holding the left mouse button down and dragging it
→ Insert the FF function block by releasing the left mouse button
→ Either position the next function block of the same type by clicking the left mouse button or
→ Terminate the operation with a right mouse button
→ End positioning: ESC or right mouse button at any time

After a function block type has been selected, a function block instance is positioned in the graphics area. During this operation the function block is displayed in diagrammatic form. After the function block has been inserted a border is displayed again to indicate that another function block of the same type can now be inserted.

The function block added to the FBAD in this way is **not yet assigned to a device**. Assignment to a device must be performed subsequently.

Each FF function block has a parameter dialog which contains as defaults the values that were entered for the function block class. There are no defaults for the mandatory parameters **function block name** and **assigned device**.

⚠ The representation of the block must not overlap with other graphic elements in the FBAD. The minimum clearance of three grid points from input/output pins and two grid points vertically from other function blocks must be maintained.

Selection from the link view

→ Select the desired FF function block from the link view
→ Move the selected element to the desired position in the graphics area by holding the left mouse button down and dragging it
→ Insert the FF function block by releasing the left mouse button.

The function block added to the FBAD in this way is already **assigned to a device**

Each FF function block has a parameter dialog which contains as defaults the values that were entered for the function block class. Only the mandatory parameter **function block name** has no default.

⚠ The representation of the block must not overlap with other graphic elements in the FBAD. The minimum clearance of three grid points from input/output pins and two grid points vertically from other function blocks must be maintained.

5.6.5.5 Assigning an FF function block to a device

An FF function block that has been added to the FBAD from the library view of function block selection is not initially assigned to a device. This assignment must be performed subsequently by means of a drag and drop operation, as described below.

→ Select an unassigned FF function block from the link view
→ Move the selected element to an unassigned FF function block of the same type in the graphics area by holding the left mouse button down and dragging it
→ Perform the assignment by releasing the left mouse button

The FF function block in the FBAD is assigned to the corresponding function block in the device.

If the FF function block from the library view is dropped on an **already-assigned** FF function block in the graphics area, the assignment is **changed** following confirmation from the user.

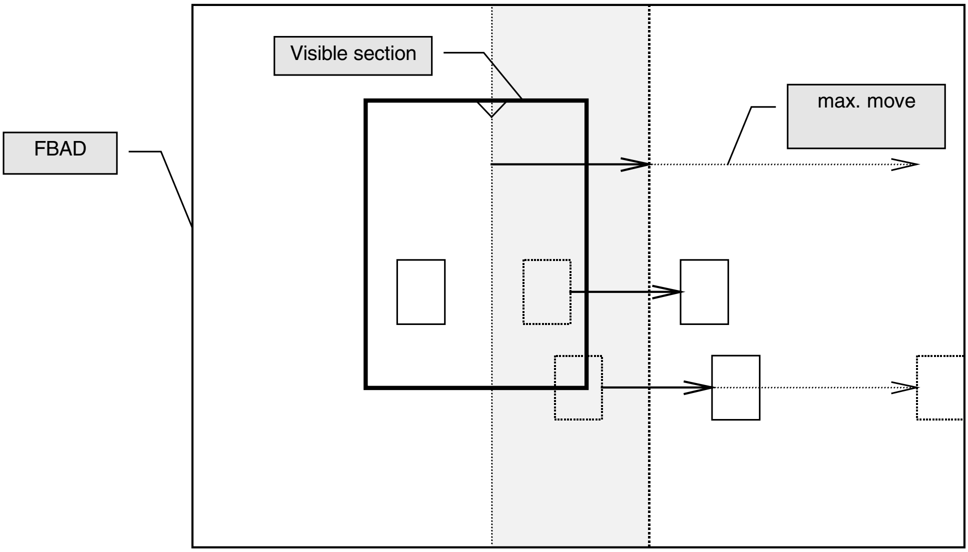
An existing device assignment can be **cancelled** via *Cancel assignment* on the shortcut menu of the FF function block.

5.6.5.6 Inserting columns and rows

Insert columns

The insertion of columns affects the current FBAD. In the area of the upper and lower border of the FBAD graphics area the form of the cursor changes to a black or red horizontal double arrow. In the case of the black double arrow, when the left mouse button is pressed a triangle with a vertical line appears, which can be moved to the right in steps the size of a grid unit. This has the effect of inserting a corresponding number of columns, with the part of the network to the right of the vertical line being moved by the same number of grid units to the right.

When the mouse is moved, if the right-hand visible edge is reached the view scrolls. The partial network can be moved only if it is not in contact with the right edge of the FBAD and if the vertical line only cuts through horizontal connections. The illustration below clarifies the procedure for inserting columns.



☞ A red double arrow at the edge of the graphics area indicates that it is not possible to insert columns in this position, e.g. because FF function blocks or FF variables are cut.

☞ Horizontal signal flow lines are elongated accordingly when columns are inserted.

Inserting rows

The insertion of rows is similar to the insertion of columns. The double arrow and triangle is positioned on the left or right-hand edge of the graphics area. The movement is marked with a horizontal line. Vertical signal flow lines are elongated accordingly when rows are inserted.

5.6.5.7 Block operations

Selecting individual FBAD elements

→ Select by left-clicking on the desired FBAD element

The whole area of the FBAD element forms the select field. The FBAD element is selected for further editing and is displayed accordingly.

☞ The default state is unselected
Connection points of signal flow lines are never displayed as selected

Selecting several FBAD elements at the same time

→ Holding the left mouse button down, stretch the frame around the elements to be selected.

All elements that are completely enclosed within the frame are selected, and they are displayed accordingly. In the case of signal flow lines, this applies to all sections that lie entirely within the frame. Following selection, the desired operation can now be performed as for individual elements, e.g. *Edit* → *Cut*.

Selecting additional FBAD elements

→ Press and hold SHIFT key → Select additional element

An element is selected in addition to the existing selection and is displayed accordingly.

☞ It is also possible to select several elements by holding down SHIFT and stretching the border.

Unselecting all selected FBAD elements

→ Left-click on an unoccupied position in the graphics area or select an unselected element

→ Using the ARROW keys move the cursor to an unoccupied position in the editor
→ Press SPACE

The FBAD elements are unselected and displayed accordingly.

If a different window is opened, any existing selection is automatically cancelled.

Removing individual FBAD elements from a selection

→ Press and hold SHIFT key and click on the element to be unselected

→ Using the ARROW keys move the cursor to the element to be unselected
→ SHIFT + SPACE

An element is removed from the existing selection and displayed accordingly.

Copy

🖱 → *Edit* → *Copy*

🖱 → Shortcut menu → *Copy*

⌨ → CTRL + C

During copying, the selected elements are transferred into an internal buffer. Any elements that have been transferred to the internal buffer through an earlier copying operation are overwritten. It is possible to see whether or not there are any elements present in the internal buffer from the menu item *Paste* on the Edit menu or in the shortcut menu. If this menu item is disabled, this means that the internal clipboard is empty.

☞ When function blocks are copied, their parameters are retained. However, the tag name for the copy is deleted as it must be unique.

Cutting and deleting

🖱 → *Edit* → *Cut* or *Delete*

🖱 → Shortcut menu → *Cut* or *Delete*

⌨ → CTRL + X to cut or DEL to delete

If the selected elements have been cut, then they can be placed back in the same FBAD by means of *Paste*. Cut has the effect of overwriting any elements that are already present in the internal buffer.

☞ If the elements are deleted, then they can no longer be pasted afterwards. Deleted elements can only be restored by exiting the FBAD without saving.

When function blocks are cut, their parameter data and the tag name are transferred along with them to the internal buffer, so that all the data is available again the next time they are pasted.

Paste

The following options are available for pasting previously copied or cut elements:

🖱 → *Edit* → *Paste*

🖱 → Shortcut menu → *Paste*

⌨ → CTRL + V

Following the paste, a surrounding rectangle with a dashed border appears at the position at which the block was previously cut or copied.

Pasted function blocks are assigned the status **incorrect**. Their parameter data is pasted with them. The tag name is only available if the function block was previously cut.

Move block

The following options are available for moving a block:

🖱 A selected block can be clicked on and the left mouse button held down. A rectangle is then displayed surrounding the selected block. The block can now be moved by moving the mouse. At the target position the left mouse button is released again. If the block cannot be inserted at the target position, a warning tone is emitted to alert the user and the surrounding rectangle remains active.

If the cursor is moved into the rectangle that is displayed after a block is pasted, it changes into a cross with an arrow for each horizontal and vertical direction of movement. The block can then be moved by holding down the left mouse button, moving the mouse, then releasing the mouse button at the target position. If the block cannot be inserted at the target position, a warning tone is emitted to alert the user and the surrounding rectangle remains active.

⌨ The cursor can be moved over a selected block or into the rectangle that appears after a block is pasted, and the space bar pressed. This causes the cursor to change into a cross with an arrow for each horizontal and vertical direction of movement. The block can then be moved using the cursor keys or dragged using the mouse. At the target position the block is inserted by pressing the space bar. If the block cannot be inserted at the target position, a warning tone is emitted to alert the user and the surrounding rectangle remains active.

While the selected elements are being moved to a new position their outlines remain visible. The move process has the effect of disconnecting any signal flow lines that are involved. These must be corrected later. Function blocks' parameter data and tag names are retained.

Undo action

🖱 → *Edit* → *Undo*

🖱 → Shortcut menu → *Undo*

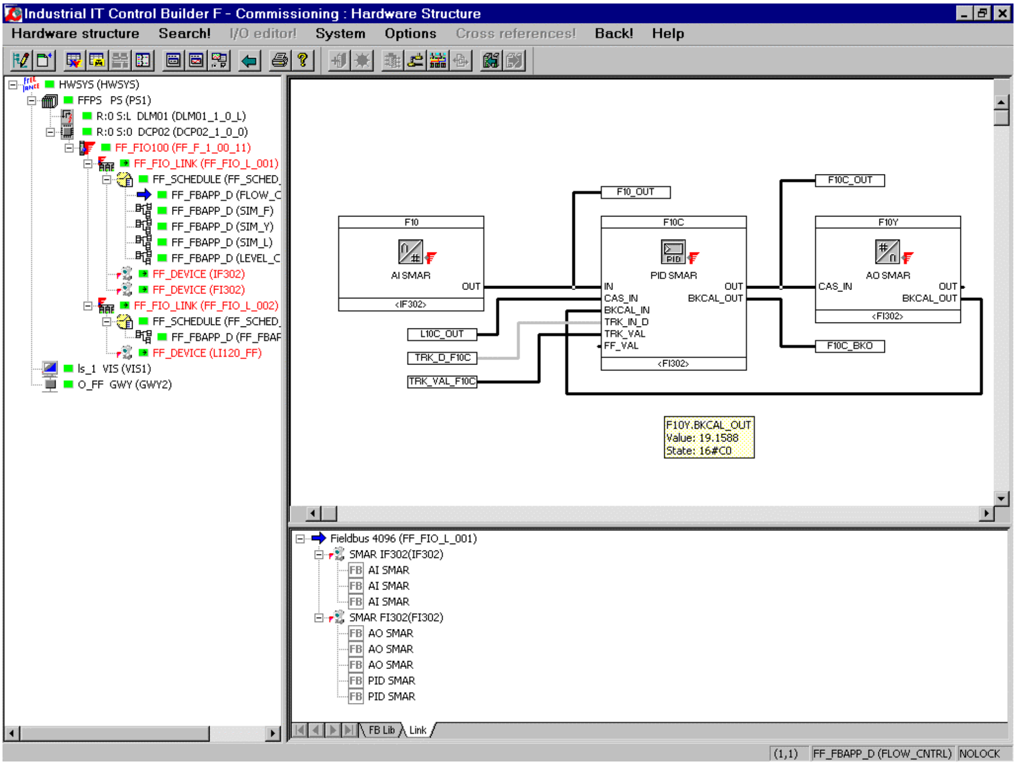
⌨ → CTRL + Z

This function enables the last action performed to be undone. The status of the FBAD remains incorrect following the undo until the next check is performed.

5.6.6 Commissioning an FBAD

The FBAD is displayed in the same way in commissioning mode as in configuration mode. However, in commissioning mode the structure of the FBAD cannot be altered.

The individual FF function blocks can be selected, and parameter values entered for them. The commissioner also has certain functions at his/her disposal for testing the FBAD. Thus, for example, if the FF variables or the pins of FF function blocks are overrun, the current values are displayed.



F10Y.BKCAL_OUT
Value: 19.1588
State: 16#C0

ff029us.bmp

5.6.7 FF variables list and tag list

5.6.7.1 FF variables list

→ Select FF variable in the FBAD editor → F2

Control branches to the FF variables list for the FIO-100 H1 link. In this list, out of all the FF variables of the H1 link one FF variable can be selected and transferred to the FBAD. For further details see **Page 73**, **FF variables list.**

5.6.7.2 Tag list

→ *System* → *Tag list*

The tag list is called up. It contains a list of all the tag names that have been assigned in the system. For details see **Engineering Manual, IEC 61131-3 Programming, Tags**.

⚠ If tag names that have been assigned to FF function blocks in the FBAD have been deleted in the tag list, then after returning from the tag list the entries in the corresponding parameter dialogs will be empty and must be entered again.

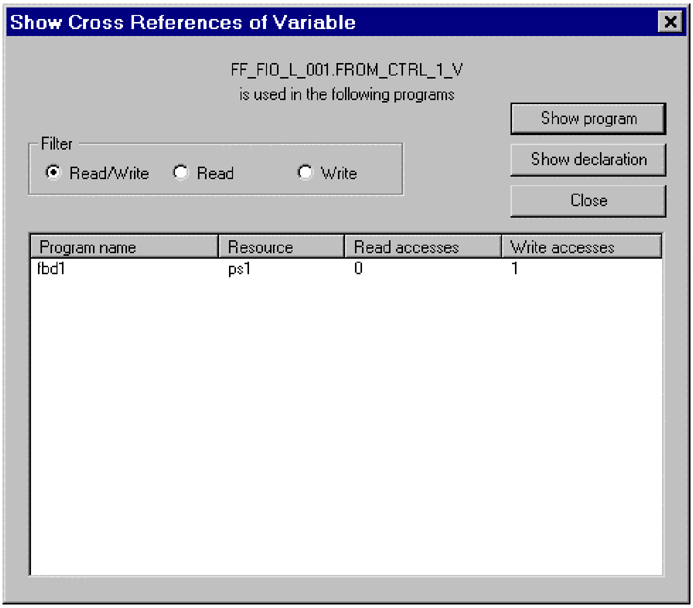
5.6.8 Cross references

The cross references can be selected directly from the FBAD.

→ Select FF variable or tag in the FBAD editor
→ *Cross references* or F5
→ If necessary select IO component (value, status or tag) to branch to and press the Ok button

The window offers various sorting and filtering options. The settings that are entered here are stored.

5.6.8.1 FF variables



Show Cross References of Variable

FF_FIO_L_001.FROM_CTRL_1_V
is used in the following programs

Filter
● Read/Write ○ Read ○ Write

Program name	Resource	Read accesses	Write accesses
fbd1	ps1	0	1

ff062us.bmp

SHOW PROGRAM Call up an FBAD with the FF variable preselected.

SHOW DECLARATION Branch to the declaration of the FF variable in the FF variables list.

It is possible to branch to the FBADs listed as cross references. If the menu item *Back!* is activated in the FBAD that control has branched to, then control jumps back to the original FBAD.

5.6.8.2 Tags

For tags, unlike FF variables, no read/write access rights are defined.

Any of the following actions can be performed:

SHOW PROGRAM Call the incidence point for the tag.

SHOW DECLARATION Branch to the declaration of the tag in the tag list.

5.6.9 General processing functions

5.6.9.1 Save FBAD

→ *Hardware structure* → *Save*

The FBAD is saved without exiting from it. Even incorrect FBADs can be saved and completed at a later time.

If the FBAD currently being edited - i.e. the detail view of the FF function block application diagram object - is exited, the user is prompted to choose whether or not the changes made since the last save are to be saved.

⚠ If changes are not saved on closing or previously in the project tree, then changes in the FBAD do not take effect.

5.6.9.2 Document FBAD

→ *Hardware structure* → *Document*

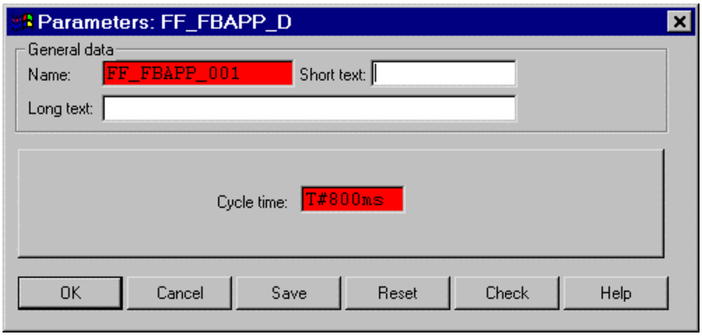
Control is switched from the FBAD to documentation management. Here, the project documentation is defined and printed on a user-specific basis. **For details see Engineering Manual, System Configuration, Documentation.**

5.6.9.3 Setting FBAD parameters

→ Select FBAD object (FF_FBAPP_D) in the hardware structure
→ *Edit* → *Parameters*

→ Double-click on the FBAD object in the hardware structure

The parameter dialog for the FBAD is opened.



Parameters: FF_FBAPP_D

General data

Name: FF_FBAPP_001

Short text:

Long text:

Cycle time: T#800ms

ff040us.bmp

Cycle time All the FF function blocks in an FBAD are processed cyclically using the FBAD cycle time.

5.6.9.4 End FBAD editing

The FBAD editor is the detail view of each FF function block diagram object (FF_FBAPP_D). The editing of an FBAD is therefore ended by selecting another object in the hardware structure. The user is asked to choose whether or not to save changes made since the last save.

If the project is not saved when it is closed and has not been saved earlier in the project tree, then the changes will not take effect in the FBAD.

5.6.9.5 Producing hard copy

→ *Options* → *Hardcopy*
→ Confirm the printer settings in the Windows printing dialog

The content of the screen is output to the printer.

5.6.9.6 Check FBAD elements

🖱 → Select FBAD element
→ *Hardware structure* → *Check*

All inputs relevant to the function are checked for syntactical and contextual correctness. Any errors and warnings that are discovered are displayed in the form of an error list. If the check reveals errors, the editing status of the FBAD element is set to **incorrect.**

☞ Newly added, copied or moved FBAD elements have the editing status **incorrect**.

5.6.9.7 Delete FBAD

🖱 → Hardware structure → Select FBAD that is to be deleted → *Edit* → *Delete*

The FBAD is deleted. The FF variables and tag names in other programs, in the FF variables list and the tag list are not affected, and can be assigned again.

5.6.9.8 FBAD copy and paste

🖱 → Hardware structure → Select the FBAD that is to be copied → *Edit* → *Copy* or CTRL + C
→ Select FF schedule object to which the FBAD is to be copied
→ *Edit* → *Paste* or CTRL + V
→ Assign FBAD name

The FBAD is copied and assigned under a **new, unique name** to an FIO-100 H1 link. The FBAD is copied in its current configuration, and including parameters. The names of the FF variables and FF function blocks are **not copied.** The copying of an FBAD does not affect the declaration of FF variables or tags. The copied FBAD is labeled as **incorrect.**

5.6.9.9 Data exchange

Data exchange between FBAD and IEC 61131 program

FBADs are linked with IEC 61131 programs through **I/O FF variables**. See **Page 72**, **FF variables**.

Data exchange between FBADs in an FIO-100 H1 link

The FBADs in an FIO-100 H1 link are linked with one another through **local FF variables**.

An **I/O FF variable** that has been set up for data exchange with IEC 61131 programs can also be used for data exchange with other FBADs in the same FIO-100 H1 link. See **Page 72**, **FF** variables.

5.7 Schedule editor

5.7.1 FF schedule

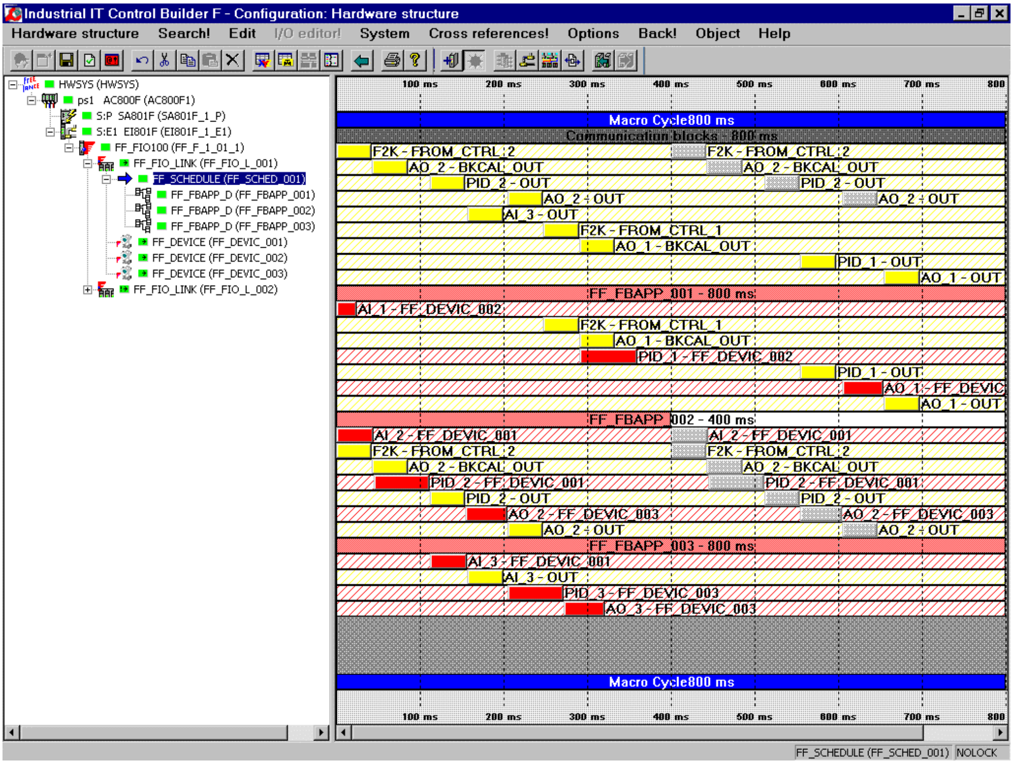
The FF scheduler produces the **chronological sequence of processing (FF schedule)** for an H1 link. Based on information from the FBADs and the devices of the link, the start times for the function blocks and the communication times within a macro cycle are determined. The FF schedule forms the basis for the configuration data that is to be loaded onto the field devices.

With the help of the schedule editor, the automatically generated processing sequence can be shown and modified in a graphical form.

5.7.2 Schedule editor interface

In the schedule editor the chronological processing sequence for an H1 link within a macro cycle is represented in a graphical form.

5.7.2.1 Structure of the graphical display



ff011us.bmp

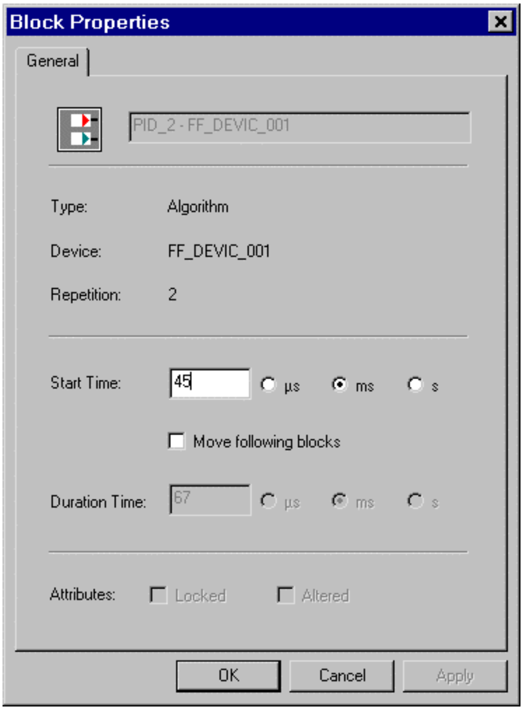
The times for **function block processing** and **bus communication** are represented by blocks of corresponding length arranged on the time axis. The display interval is a macro cycle.

All the communication blocks are grouped together in the upper section. These represent the whole **cyclical communication** taking place in the H1 link.

Additional sections containing both algorithm blocks and communication blocks are grouped below. Each of these sections shows the chronological processing sequence of an FBAD, also known as a **loop**.

Additional information on each of the objects displayed can be shown using the *Properties* window. Some of the data displayed in the *Properties* window can be edited. The window is opened as follows:

→ Mark object → Shortcut menu: *Properties*



Block Properties

General

PID_2 - FF_DEVIC_001

Type: Algorithm

Device: FF_DEVIC_001

Repetition: 2

Start Time: 45 ○ µs ● ms ○ s

□ Move following blocks

Duration Time: 67 ○ µs ● ms ○ s

Attributes: □ Locked □ Altered

ff057us.bmp

The *Block properties* window for an algorithm block contains, for example, the start time of the associated function block with reference to the start of the macro cycle and its execution time.

5.7.2.2 Change defaults

Change colors

→ Shortcut menu for schedule → *Color*
→ Select object type whose color is to be changed
(e.g. color for a communication block)
→ Select desired color

Objects of the corresponding type are displayed in the selected color.

Change scaling of the time axis

🖱 → Shortcut menu for schedule → *Time scale*
→ Select desired scaling

The FF schedule is displayed with modified time axis scaling.

The scaling can be changed incrementally using the "+" or "-" keys on the numeric keypad.

⌨ → "+“ key or "-“ key on numeric keypad

5.7.3 Edit FF schedule

The schedule editor is the detail view of each FF schedule object (FF_SCHEDULE). An FF schedule (chronological processing sequence for an H1 link) is therefore called up for editing by simply selecting the corresponding object in the **hardware structure**. The FF schedule that was called is displayed and can be modified.

🖱 → Select FF schedule object (FF_SCHEDULE) in the **hardware structure**

When the FF schedule is opened for the first time, the automatically-generated *optimized* schedule for the H1 link is displayed.

5.7.3.1 Move block

A single block or several blocks of a loop can be moved on the time axis using the mouse or by changing the start time in the *Properties* window for the block.

Move a single block

🖱 → Hold left mouse button down and move a block

🖱 → Shortcut menu for block → *Properties*
→ Change *start time*

The selected block is moved on the time axis; it is assigned a different start time within the macro cycle.

Move several blocks

→ Holding the CTRL key and the left mouse button down, move a block

→ Shortcut menu for block → *Properties*
→ Select check box *Move following blocks*
→ Change *start time*

The selected block and all the dependent blocks of the loop in the signal flow (these are generally the blocks lying to the right of the chosen block) are moved on the time axis; they are assigned different start times within the macro cycle.

5.7.3.2 Lock and unlock loop

A loop can be protected against unintended modification by locking it. Locking is recommended in particular after manual changes have been carried out (blocks moved) in a loop.

In a locked loop and the associated FBAD, no **actions which change the signal flow** can be performed. Thus, for example, it is not possible to move blocks or change the cycle time in this loop. Likewise, the deletion or addition of function blocks and signal lines and the modification of device assignments in the associated FBAD are also not permitted. Setting values of parameters for function blocks and variables, on the other hand, is permitted.

Lock loop

→ Shortcut menu for the loop (select the header row of a loop section displaying the name and cycle time of the associated FBAD)
→ *Lock*

The loop and the associated FBAD are locked.

Unlock loop

→ Shortcut menu for the loop (select the header row of a loop section displaying the name and cycle time of the associated FBAD)
→ *Unlock*

The lock on the loop and the associated FBAD is cancelled.

5.7.3.3 Optimize FF schedule

An FF schedule is automatically optimized after changes have been made to an FBAD from the H1 link.

However, the optimization of the FF schedule can also be started manually.

🖱 → Shortcut menu for the FF schedule → *Optimize*

☞ During optimization of the FF schedule, locked loops are not changed.

5.7.3.4 Save FF schedule

🖱 → Shortcut menu for FF schedule → *Save*

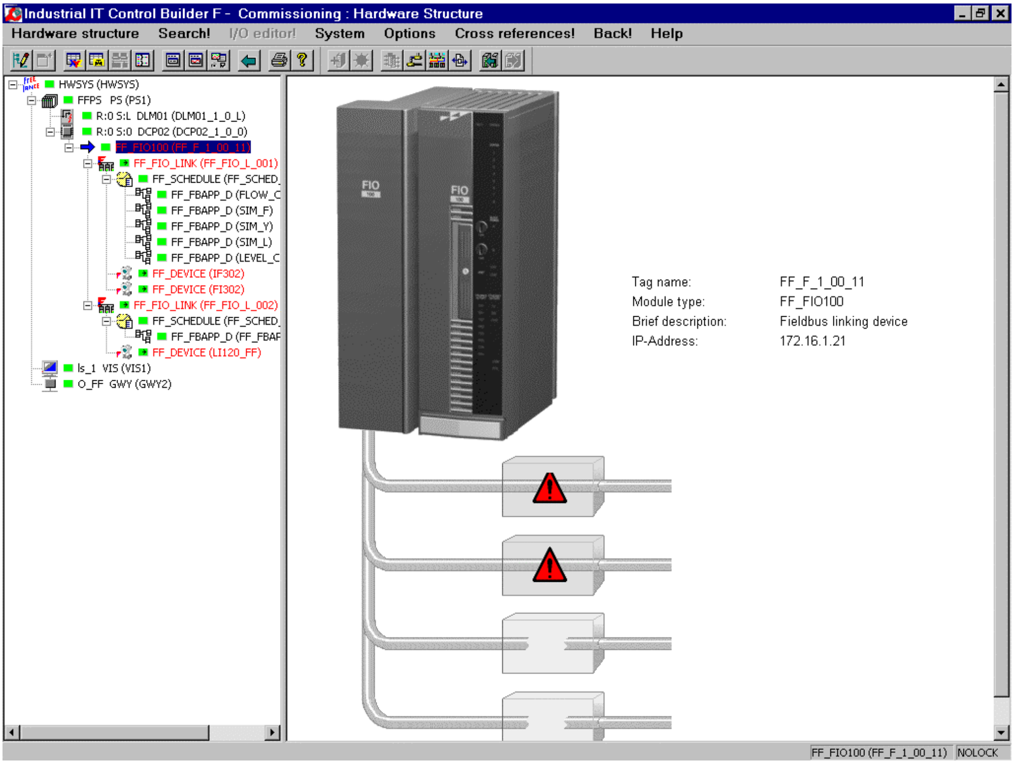
The FF schedule is saved.

5.8 Commissioning FF objects

5.8.1 Display during commissioning

5.8.1.1 FF FIO-100 object

In the detail view of the FF FIO-100 object, status information is displayed for both the FIO-100 and the FIO-100 H1 links.



Tag name: FF_F_1_00_11
Module type: FF_FIO100
Brief description: Fieldbus linking device
IP-Address: 172.16.1.21

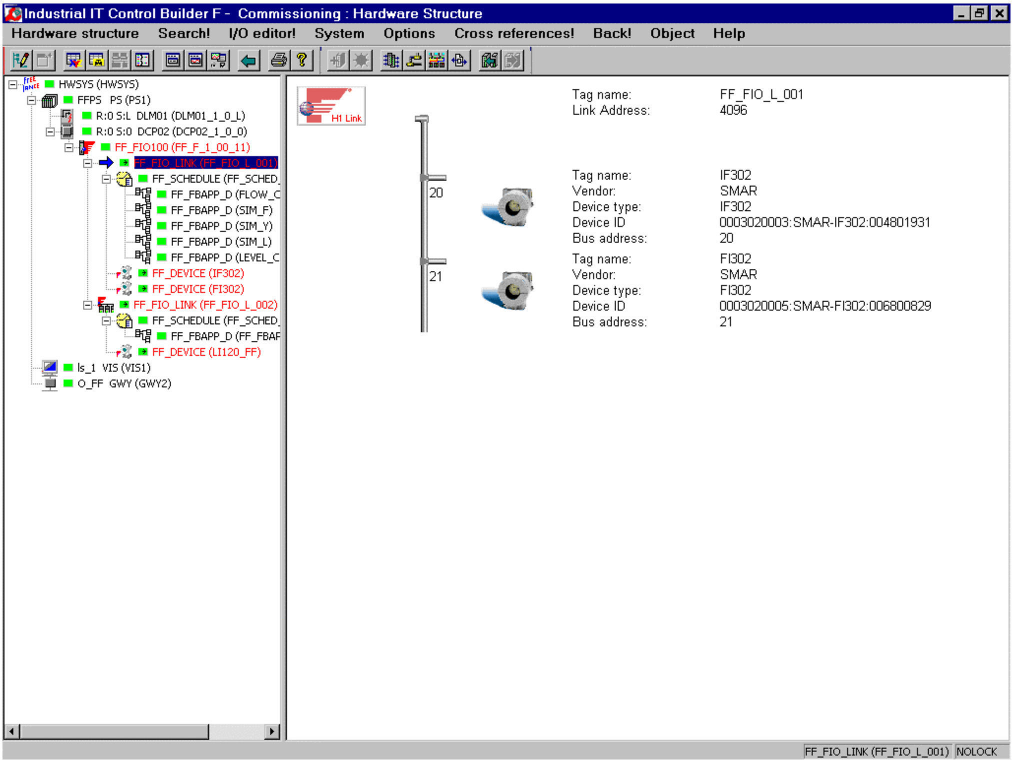
ff058us.bmp

The FIRMWARE UPDATE button is provided in the parameter dialog for the FIO-100. This is used to load the current version of the firmware onto the FIO-100.

⚠ During a firmware update the FIO-100 loses its configuration data. This also applies to the local configuration data for the FIO-100 H1 links.

5.8.1.2 FF FIO-100 H1 link object

The detail view of the FF FIO-100 H1 link object contains status information for the FIO-100 H1 link and for the field devices in this link.



```
HWSYS (HWSYS)
  FFPS   PS (PS1)
    R:0 S:L  DLM01 (DLM01_1_0_L)
    R:0 S:0  DCP02 (DCP02_1_0_0)
      FF_FIO100 (FF_F_1_00_11)
        FF_FIO_LINK (FF_FIO_L_001)
          FF_SCHEDULE (FF_SCHED_
            FF_FBAPP_D (FLOW_C
            FF_FBAPP_D (SIM_F)
            FF_FBAPP_D (SIM_Y)
            FF_FBAPP_D (SIM_L)
            FF_FBAPP_D (LEVEL_C
          FF_DEVICE (IF302)
          FF_DEVICE (FI302)
        FF_FIO_LINK (FF_FIO_L_002)
          FF_SCHEDULE (FF_SCHED_
            FF_FBAPP_D (FF_FBAP
          FF_DEVICE (LI120_FF)
  ls_1  VIS (VIS1)
  O_FF  GWY (GWY2)
```

Tag name: FF_FIO_L_001
Link Address: 4096

Tag name: IF302
Vendor: SMAR
Device type: IF302
Device ID 0003020003:SMAR-IF302:004801931
Bus address: 20

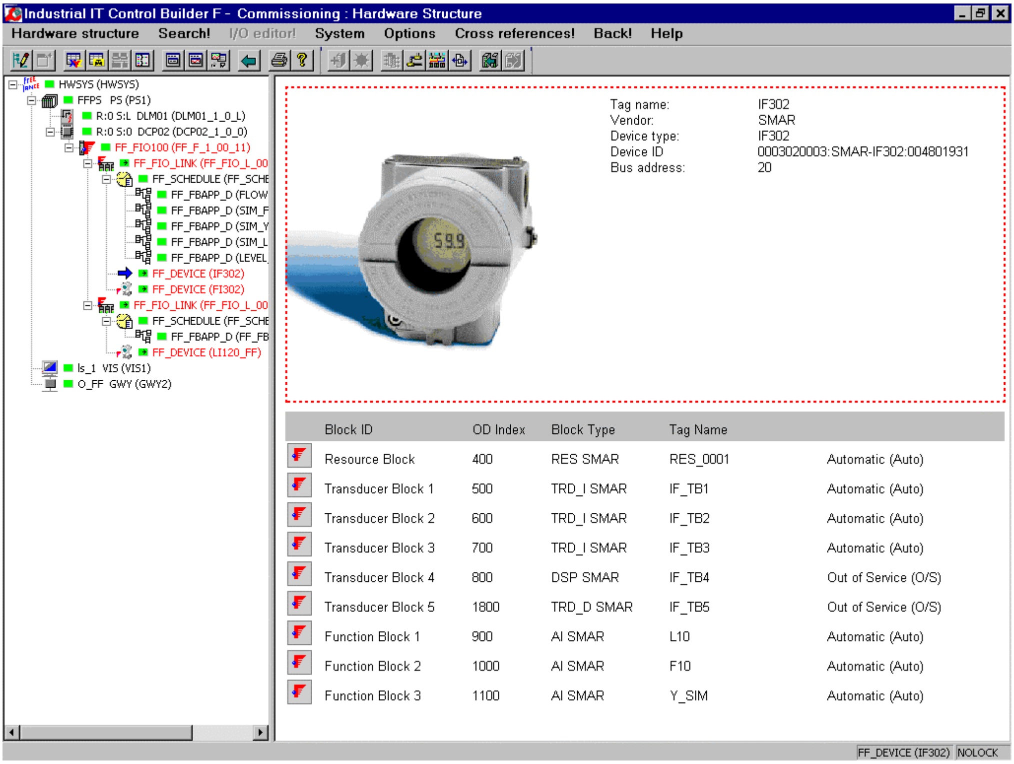
Tag name: FI302
Vendor: SMAR
Device type: FI302
Device ID 0003020005:SMAR-FI302:006800829
Bus address: 21

FF_FIO_LINK (FF_FIO_L_001) NOLOCK

ff059us.bmp

5.8.1.3 FF device instance object

The detail view of the FF device instance object contains status information for the field device. The current operating mode of the resource blocks, transducer blocks and function blocks is also displayed.



Tag name:	IF302
Vendor:	SMAR
Device type:	IF302
Device ID	0003020003:SMAR-IF302:004801931
Bus address:	20

	Block ID	OD Index	Block Type	Tag Name	
	Resource Block	400	RES SMAR	RES_0001	Automatic (Auto)
	Transducer Block 1	500	TRD_I SMAR	IF_TB1	Automatic (Auto)
	Transducer Block 2	600	TRD_I SMAR	IF_TB2	Automatic (Auto)
	Transducer Block 3	700	TRD_I SMAR	IF_TB3	Automatic (Auto)
	Transducer Block 4	800	DSP SMAR	IF_TB4	Out of Service (O/S)
	Transducer Block 5	1800	TRD_D SMAR	IF_TB5	Out of Service (O/S)
	Function Block 1	900	AI SMAR	L10	Automatic (Auto)
	Function Block 2	1000	AI SMAR	F10	Automatic (Auto)
	Function Block 3	1100	AI SMAR	Y_SIM	Automatic (Auto)

ff060us.bmp

5.8.1.4 Status displays for the objects

FF FIO-100 H1 object

No connection to the FIO-100 (red exclamation mark)
No communication connection exists between the Control Builder F and the FIO-100.

FF FIO-100 H1 link object

No connection to the H1 link (red exclamation mark)
No communication connection exists between the Control Builder F and the H1 link.

H1 link not loaded (red exclamation mark)
The configuration data for the H1 link have not been loaded onto the FIO-100.

☞ The configuration data for the H1 link must be loaded onto the FIO-100. See **Page 115**, **FIO-100 .**

FF device instance object

FF device does not exist (red exclamation mark)
No FF device exists at the specified address.

Wrong FF device (red exclamation mark)
The device ID of the current FF device does not correspond to the device ID in the database.

☞ A device allocation must be performed for this FF device. See **Page 116**, **Assign devices**.

Wrong version (red exclamation mark)
The loaded configuration data for the FF device does not correspond to the configuration data in the database.

☞ The configuration data for the H1 link must be loaded. See **Page 117**, **Load configuration of the H1 link**.

5.8.2 Force values and substitute values

Force values and substitute values are set in the FIO-100 H1 link object. Force values and substitute values can be written and corrected when Control Builder F is in commissioning mode. See **Page 50**, **Force values / substitute values tab.**

5.8.3 Commissioning an H1 link

The steps described below for commissioning an H1 link are carried out in commissioning mode in Control Builder F. They assume that the AC 800F has been loaded along with its configuration data. The commissioning of the H1 link comprises the following three steps:

- The FIO-100 is initialized and addresses are assigned. This step is required once per FIO-100. Whenever an H1 link is added during configuration this step must be carried out again during commissioning of the added H1 link.
- Each device that is currently present in the H1 link is assigned to a configured device.
- The FIO-100 H1 port and all devices in the H1 link are loaded along with their configuration data.

The list known as the HSE live list serves as a tool for commissioning; this is a hierarchical display of all the FF devices currently available on the HSE subnet.

5.8.3.1 FIO-100 initialization and address assignment

The first step for commissioning an H1 link is the initialization and address assignment of the FIO-100. During this step the FIO-100 is initialized and the tag name and the FIO-100 link addresses are loaded.

→ Shortcut menu for the FIO-100 (FF_FIO100)
→ *Precommissioning*
→ *FIO-100 initialization and address assignment*

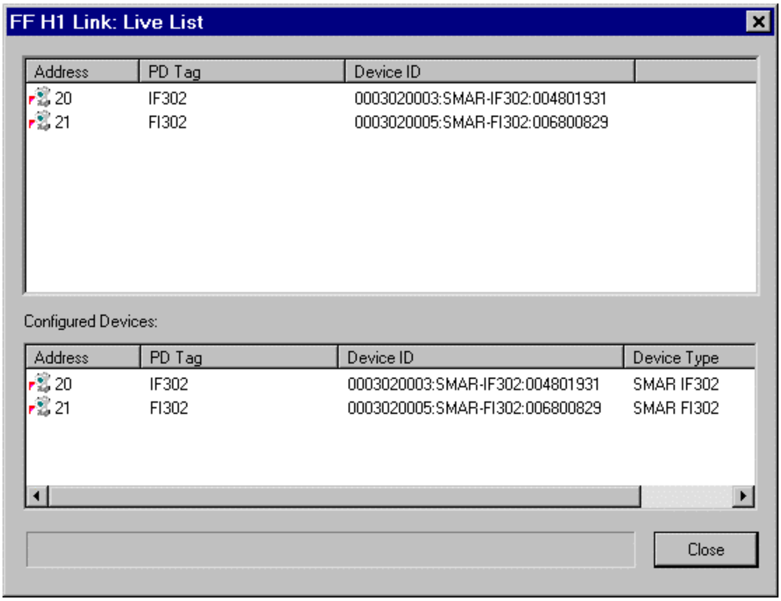
If another H1 link is added to an FIO-100 in configuration mode after at least one H1 link of this FIO-100 has been commissioned, then during commissioning of the added H1 link the initialization and address assignment has to be carried out again for the FIO-100. As this initializes the H1 ports of the FIO-100, the **communication in all H1 links of the FIO-100 is stopped**. To restart communication in the H1 links already commissioned, the H1 ports of the FIO-100 must be downloaded.

→ Shortcut menu for the FIO-100 (FF_FIO100)
→ *Download*
→ *Download FIO-100 with H1 ports only*

5.8.3.2 Assign devices

The second step in the commissioning process is the assignment of each device currently available in the H1 link to a configured device. Here, the configured tag name and the bus address are written to the device. The device ID is also read from the field device and added to the database.

→ Shortcut menu for the FIO-100 H1 link (FF_FIO_LINK)
→ *Live list*
→ Select a device that is currently available in the H1 link (top half of window)
→ Press and hold down the left mouse button and move it to the desired configured device (bottom half of window)
→ Repeat the last two steps until an actual device is assigned to each configured device



FF H1 Link: Live List

Address	PD Tag	Device ID
20	IF302	0003020003:SMAR-IF302:004801931
21	FI302	0003020005:SMAR-FI302:006800829

Configured Devices:

Address	PD Tag	Device ID	Device Type
20	IF302	0003020003:SMAR-IF302:004801931	SMAR IF302
21	FI302	0003020005:SMAR-FI302:006800829	SMAR FI302

ff064us.bmp

Address Bus address of the device. When assigning a device, the configured bus address is written to the actual device.

Tag name Tag name of the device. When assigning a device, the configured tag name is written to the actual device.

Device ID Unique identification number for the device. The device manufacturer assigns a unique device ID to each individual device, and this ID never changes. When a device is assigned, the device ID is read from the device and added to the database.

Device type Device type from the device description. The device type is shown only in the list of configured devices (bottom half of window). It serves as supplementary information for the user when assigning the device.

A configured device which has not yet had a bus subscriber assigned to it appears with no device ID in the list of configured devices (bottom half of window).

The shortcut menu for a device offers an alternative method of assigning the device.

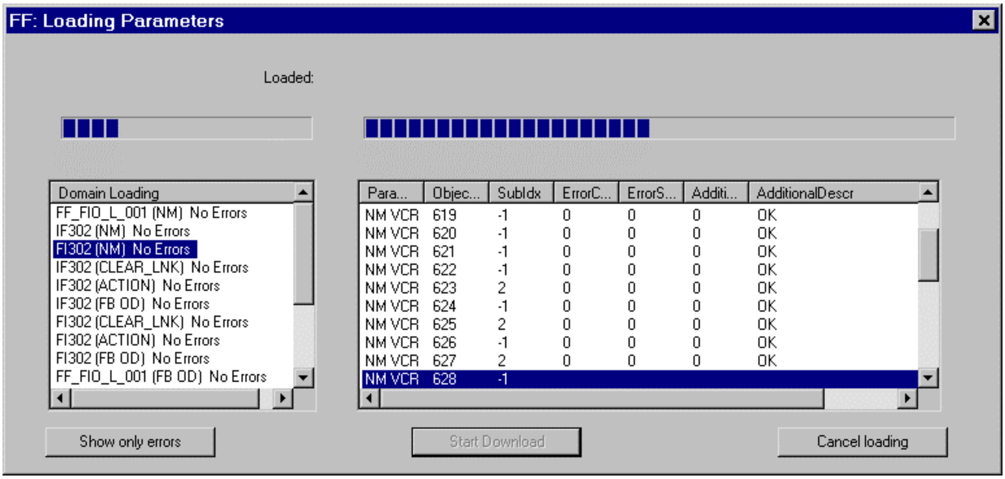
→ Shortcut menu for a device (FF_DEVICE) to which a bus subscriber is to be assigned
→ *Assign device*
→ Select an actual device that is available in the H1 link

5.8.3.3 Load configuration of the H1 link

In the third step of the commissioning process, the FIO-100 H1 port and all devices in the H1 link are loaded along with their configuration data.

→ Shortcut menu for the FIO-100 H1 link (FF_FIO_LINK)
→ *Download*
→ *Whole link*

The progress of the loading operation and any errors or warnings that may occur are displayed in the following window.



Domain Loading
FF_FIO_L_001 (NM) No Errors
IF302 (NM) No Errors
FI302 (NM) No Errors
IF302 (CLEAR_LNK) No Errors
IF302 (ACTION) No Errors
IF302 (FB OD) No Errors
FI302 (CLEAR_LNK) No Errors
FI302 (ACTION) No Errors
FI302 (FB OD) No Errors
FF_FIO_L_001 (FB OD) No Errors

Para...	Objec...	SubIdx	ErrorC...	ErrorS...	Additi...	AdditionalDescr
NM VCR	619	-1	0	0	0	OK
NM VCR	620	-1	0	0	0	OK
NM VCR	621	-1	0	0	0	OK
NM VCR	622	-1	0	0	0	OK
NM VCR	623	2	0	0	0	OK
NM VCR	624	-1	0	0	0	OK
NM VCR	625	2	0	0	0	OK
NM VCR	626	-1	0	0	0	OK
NM VCR	627	2	0	0	0	OK
NM VCR	628	-1				

ff065us.bmp

ONLY DISPLAY ERRORS Only errors are displayed; warnings are suppressed.

DISPLAY WARNINGS Errors and warnings are displayed.

CANCEL LOADNG The loading of configuration data is cancelled.

Each line in the right-hand half of the window that has not been completed with "OK" indicates an error. In order to determine the cause of the error, the error class and error subclass are shown (see table below).

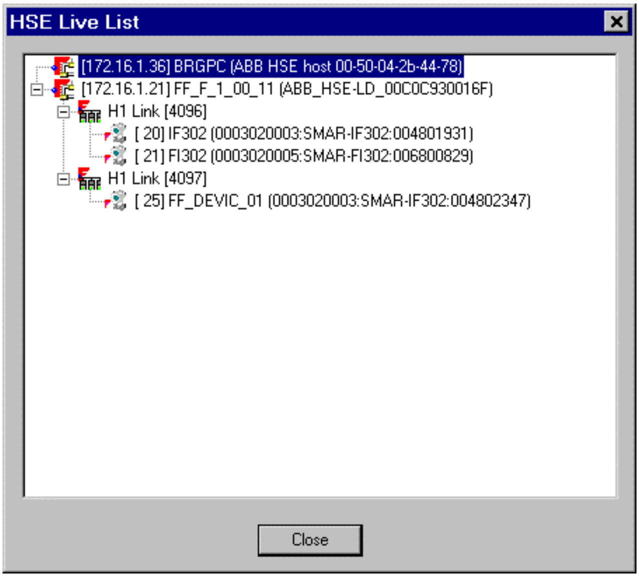
Error class	Error subclass	Explanation
0x01		VFD state
	0	Other
0x02		Application reference
	0	Other
	0x01	Application unreachable
0x03		Definition
	0	Other
	0x01	Object undefined
	0x02	Object attributes inconsistent
	0x03	Name already exists
0x04		Resource
	0	Other
	0x01	Memory unavailable
0x05		Service
	0	Other
	0x01	Object state conflict
	0x02	PDU size
	0x03	Object constraint conflict
	0x04	Parameter inconsistent
	0x05	Illegal parameter
0x06		Access
	0	Other
	0x01	Object invalidated
	0x02	Hardware fault
	0x03	Object access denied
	0x04	Invalid address
	0x05	Object attribute inconsistent
	0x06	Object access unsupported
	0x07	Object non-existent
	0x08	Type conflict
	0x09	Named access unsupported
	0x0A	Access to element unsupported

Error class	Error subclass	Explanation
0x07		OD (Object dictionary)
	0	Other
	0x01	Name length overflow
	0x02	OD overflow
	0x03	OD write-protected
	0x04	Extension length overflow
	0x05	OD description length overflow
	0x06	Operational problem
0x08		This error class contains manufacturer-specific error subclasses

5.8.3.4 Show subscribers on the HSE subnet

The HSE live list is a tool for commissioning. It shows in a hierarchical format all the FF devices currently available on the HSE subnet. FF devices from other projects that may be present on the subnet are also displayed.

→ Shortcut menu for the FIO-100 (FF_FIO100)
→ *HSE live list*



```
[172.16.1.36] BRGPC (ABB HSE host 00-50-04-2b-44-78)
[172.16.1.21] FF_F_1_00_11 (ABB_HSE-LD_00C0C930016F)
    H1 Link [4096]
        [ 20] IF302 (0003020003:SMAR-IF302:004801931)
        [ 21] FI302 (0003020005:SMAR-FI302:006800829)
    H1 Link [4097]
        [ 25] FF_DEVIC_01 (0003020003:SMAR-IF302:004802347)
```

ff066us.bmp

5.8.4 Replacing FF devices

5.8.4.1 Replacing an FIO-100

In the event of **failure** of an FIO-100, the **cyclical communication** in the associated H1 links is generally maintained as long as backup link masters have been configured.

In this event the **replacement** of the FIO-100 can also be carried out without interrupting the cyclical communication in the H1 links as long as the power supply to the field devices in the H1 links is ensured. After the FIO-100 has been replaced, the following steps must be carried out in order to restart.

→ Parameter dialog for the FIO-100 (FF_FIO100)
→ FIRMWARE UPDATE button
→ Wait for the FIO-100 to restart (view HSE Live List)
→ Shortcut menu for the FIO-100 (FF_FIO100)
→ *Download*
→ *Download FIO-100 with H1 ports only*

First, the current firmware is loaded onto the FIO-100. Next, the configuration data for the FIO-100 including the local configuration data for the H1 links is loaded. The FF devices in the H1 links are **not loaded** during this process.

5.8.4.2 Replacing an FF device

When an FF device in an H1 link **fails**, the device must be reassigned after it has been replaced. Next, the configuration of the H1 link must be loaded.

→ Shortcut menu for the FIO-100 H1 link (FF_FIO_LINK)
→ *Live list*
→ Select the replacement device (top half of window)
→ Press and hold down the left mouse button and move the device to the appropriate configured device (bottom half of window)
→ Shortcut menu for the FIO-100 H1 link (FF_FIO_LINK)
→ *Object*
→ *Download whole H1 link*

Index

Gross Automation, 1725 South Johnson Road, New Berlin, WI 53146, www.ssacsales.com, 800-349-5827