Pipeline Management Systems with MicroSCADA

System Description

ABB Process Industries
MicroSCADA System Description

Table of Contents

1  ABB MICROSCADA - SYSTEM DESCRIPTION 1

1.1  MicroSCADA overview 1
1.1.1  Application areas 1
1.1.2  System configuration 2

1.2  Software description 2
1.2.1  Software structure 2
1.2.2  Base system features 3
1.2.2.1  User interface 3
1.2.2.2  SCIL language 6
1.2.2.3  Application objects 8
1.2.2.4  Process database 9
1.2.2.5  Report database 13
1.2.3  Base tools 14
1.2.4  Application engineering 14

1.3  LIB 5XX application engineering 15
1.3.1  LIB 5XX structure 15
1.3.1.1  Concept 15
1.3.1.2  Application engineering 16
1.3.1.3  Standard functions 16
1.3.1.4  Picture function 17
1.3.2  LIB 500 Base components 17
1.3.2.1  Backbone functions 17
1.3.2.2  Alarm list function 18
1.3.2.3  Event list function 21
1.3.2.4  Blocking list function 22
1.3.2.5  Trend reports 24
1.3.2.6  Measurement report function 25
1.3.2.7  System self supervision function 27
1.3.3  LIB 540 Pipeline Process 27
1.3.3.1  Basic Standard Functions 32
1.3.3.2  Standard Valve Device 32
1.3.3.3  Standard Pump Device 33
1.3.3.4  Control Valve Device with Control Unit 34
1.3.3.5  Monitoring Functions 35
1.3.4  Flow Visualization 41
List of Figures

Figure 1 - System overview 2
Figure 2 - MicroSCADA software structure 3
Figure 3 - Examples of Visual SCIL objects 5
Figure 4 - An example of a MicroSCADA screen as seen from the application engineer’s point of view. The screen shows the application object navigator and the data object tool. 6
Figure 5 - Object with some static and dynamic attributes 9
Figure 6 - The functions of the process database 11
Figure 7 - Library package structure 15
Figure 8 - An illustration of the application engineering using the LIB 5XX standard functions 16
Figure 9 - Some backbone pictures 17
Figure 10 - Alarm list, template 1 19
Figure 11 - The acknowledgment dialog 20
Figure 12 - Event list, main view 21
Figure 13 - Blocking list example 23
Figure 14 - Graphical presentation of a trend 24
Figure 15 - Report example 26
Figure 16 - System self supervision 27
Figure 17 - A process picture in form of a single line diagram 28
Figure 18 - Single-line diagram with control dialogs 31
Figure 19 - Single-line diagram with blocking dialogs 32
Figure 20 - Station and Group Standard function 32
Figure 21 - The On off valve standard function 33
Figure 22 - The On/off valve standard function main dialog 33
Figure 23 - The pump standard function 33
Figure 24 - The pump standard function main dialog 34
Figure 25 - The On off valve standard function 34
Figure 26 - The On/off valve standard function main dialog 35
figure 27 - Three types of measurement symbols 35
figure 28 - The measurement dialog 37
figure 29 - A station with a small and a zoomed Fast Trend 38
figure 30 - Level Indicator as tank measurement 38
figure 31 - Different symbols for the alarm indicator standard function 39
figure 32 - Main control dialog for the Collective and Single Alarm indicator 40
figure 33 - The last alarm dialog for the Alarm indicator 41
figure 34 - Flow visualization in a pump station 42

List of Tables

Table 1 - Correspondence between process object types and station type specific data types for RTUs 12
Table 2 - Default colors and status texts in alarm types 19
Table 3 - Sub functions for measurement 36
1 ABB MicroSCADA - System Description

1.1 MicroSCADA overview

ABB has years of experience in supervisory control and automation systems. ABB is a leading supplier in this field all over the world.

MicroSCADA is a microcomputer-based, distributed and programmable supervisory control and data acquisition system (SCADA). MicroSCADA provides integration with pipeline information, pipeline control and substation control to the consumer.

MicroSCADA is an integrated part of the Panorama concept as well as of the S.P.I.D.E.R. and Pyramid product families.

1.1.1 Application areas

MicroSCADA is a system for local and remote control applications suitable for non-electrical as well as electrical processes.

The MicroSCADA software is used for a number of applications areas, among others:

- transport pipeline control
- gas distribution pipeline control
- district heating
- water purification and distribution
- waste water treatment
- power transmission and distribution

The MicroSCADA software in general and the LIB 5XX application software library are designed and developed for the above mentioned application systems.
1.1.2 System configuration

The system is composed of one or more base systems, workstations and communication units.

![System overview diagram]

Figure 1 - System overview

1.2 Software description

1.2.1 Software structure

The software in a MicroSCADA system can be divided into the following levels: operating system, platform software and application software. For a MicroSCADA base system, the levels can be more precisely described as operating system, MicroSCADA kernel and MicroSCADA applications. The operating system and the MicroSCADA kernel together form a platform for the applications.

The operating system in a MicroSCADA rev. 8.4.4 base system is Microsoft Windows 2000.

The MicroSCADA kernel is a platform software, which is identical in all installations within the same MicroSCADA revision. It does not contain customer specific or application area specific features. It offers comprehensive mission-oriented services to the applications.

The kernel is a multi-process real-time software. Several MicroSCADA applications can run simultaneously on top of one kernel. The kernel provides each application with database structures, database handling mechanisms and file handling functionality. The picture and dialog system with associated tools are the base for the user interface of the application. The SCIL language is the key service provided by the kernel. SCIL programs can be run in pictures, dialogs and command procedures, and can be activated e.g. by operator actions, process events or cyclic time signals. The kernel offers an application programming interface for attaching functions as separate programs.
All application functions of an operating MicroSCADA system reside on an application software layer which can be changed and extended according to customer needs without affecting the main program. The application software comprises pictures, dialogs, programs, databases and other files.

The MicroSCADA application is built with engineering tools utilizing services provided by the platform. The application building involves installing standard functions from LIB 500, writing customer-specific SCIL programs, etc.

### Figure 2 - MicroSCADA software structure

<table>
<thead>
<tr>
<th>Application</th>
<th>MicroSCADA Kernel</th>
<th>Operating System</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Application functions from LIB500</td>
<td>• SCIL program execution</td>
<td>• Windows 2000</td>
<td>• Standard PC hardware</td>
</tr>
<tr>
<td>• Process pictures</td>
<td>• Open Interfaces</td>
<td></td>
<td>• Communication board</td>
</tr>
<tr>
<td>• Process and report objects</td>
<td>• Application object types (classes)</td>
<td></td>
<td>• Alarm unit etc.</td>
</tr>
<tr>
<td>• SCIL programs</td>
<td>• Database management for application objects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Event list, alarm lists, reports etc.</td>
<td>• User interface services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>etc.</td>
<td>• Communication services</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Configuration management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.2.2 Base system features

#### 1.2.2.1 User interface

A MicroSCADA user interface, as the operator sees it, is composed of one or more windows, each of which may provide overviews of the controlled process, presentations of the alarm and event state, history of the process, graphical images, operator dialogs, etc.

Seen from the application engineer’s point of view, the user interface is composed of pictures and dialogs. Pictures and dialogs can be combined: on the one hand pictures can be included in dialogs, and on the other hand dialogs can be opened from pictures.

### Pictures

The picture concept in MicroSCADA rev. 8.4 is compatible with earlier revisions. A picture comprises

- a static picture background
- function keys (buttons) with associated SCIL programs
• windows
   A window in a picture can be used to present e.g. dynamic values (from databases, from process, etc.) in numeric or graphical presentation, a complete picture, figures etc. The window definition is recursive. Window representations can be stored in libraries.
• picture functions, which are installed using LIB 500 standard functions
• SCIL programs for cyclic and event-based execution

A picture may be, for instance, a dynamic presentation of a distribution substation, a report, a list or a schematic illustration.

LIB 500 is mainly based on pictures.

The pictures are designed and programmed in the Picture Editor. The SCIL language provides a set of commands for picture handling.

Visual SCIL and dialogs

Visual SCIL is a new user interface concept in rev. 8.4. The key elements of a Visual SCIL user interface are dialogs and Visual SCIL objects.

The dialogs are windows which provide dialog functions, selection boxes, lists, browser functions, images, tools, etc. They are independent entities which can be placed and moved independently of other pictures and dialogs on the screen.

Dialogs are generally designed and programmed with the Dialog Editor, utilizing the extensive set of included Visual SCIL objects, e.g., texts, buttons, lists, numeric spinners, combo boxes, check boxes, menus bars and notebooks. The Dialog Editor includes geometry management facilities and supports language-dependent texts.

As a complement to the designing of dialogs with the Dialog Editor, Visual SCIL objects can also be loaded from files or directly created by SCIL statements in pictures and dialogs.
Figure 3 - Examples of Visual SCIL objects
When designing user interface elements, dialogs are in general suitable for menus, item selection boxes, operator dialogs, imported images, lists and textual reports. Pictures are more convenient for presentation of the process. MicroSCADA 8.4 includes a number of tools, which have been designed as dialogs with Visual SCIL objects.

### 1.2.2.2 SCIL language

SCIL is a powerful high-level application programming language. SCIL has been especially designed for building application functions in MicroSCADA systems. Like most other languages, SCIL includes features such as variable assignments, arithmetics, conditions, block structure, case statements, loops etc. Additionally SCIL provides means for

- process supervision and control
- picture handling
- database management
• system configuration and management

Generally most actions of a MicroSCADA application involve running of SCIL programs. SCIL programs are executed in pictures, dialogs and command procedures, and can be activated by operator actions, process events or cyclic time signals etc.

SCIL strongly supports the object concept of MicroSCADA by a common object syntax for different types of objects. SCIL statements can directly access both local and remote objects.

A SCIL program can contain up to 10000 statements. Statements may contain

• variable definitions, e.g. value assignments
• control commands
  Control commands execute operations concerning objects, system components, printouts, program executions, etc.
• picture commands
  Picture commands give the system orders concerning picture handling, e.g. producing pictures on the screen, showing and erasing windows, etc. Picture commands can be used in pictures only.
• graphics commands
  − Primitive graphics commands, which handle full graphical elements. These commands can be used in pictures and Visual SCIL objects.
  − Visual SCIL object management commands, which can be used in pictures and Visual SCIL objects provided that the application window has been defined as a VS type monitor.
  − Motif widget handling commands, which can be used in MicroSCADA pictures and Motif widgets. They can be used in pictures on X-monitors.

All these types of statements may contain expressions, which in their turn may contain e.g. function calls, object references and variable references. SCIL programs can also call other SCIL programs.
The SCIL language includes a large number of powerful predefined functions for various types of data processing. For instance the following important categories of functions are available in SCIL:

- arithmetical functions
- time functions
- string functions
- bit functions
- vector handling functions
- database functions
- file handling functions
- communication functions
- DDE Client functions
- SQL/ODBC functions
- printout functions
- data type handling functions

1.2.2.3 Application objects

The application objects are programmable units which perform various tasks, such as real time process supervision, control procedures, data registration and storage, calculations, automatic time and event activations, etc. Application objects are created, accessed, modified and deleted by SCIL.

Information associated with the objects - their values, functions, properties and activities - are described by attributes. An object normally has many different attributes. Each attribute can be accessed with SCIL. Different object types have different sets of attributes.

The attributes can be divided into static and dynamic attributes.

- Static attributes define the identification and properties of the object. Examples of static attributes of an object are the object name, possibly an object address, activation criteria, connections to other objects, alarm handling specifications, possibly a program or expression.
- Dynamic attributes are the object value and additional information related to the value, such as status code and time tag.
The application objects are classified into the following nine types:

- **Process objects.** The process objects are images of connected process units. The objects store and supervise the real time state of the process.
- **Scales.** These are algorithms for the scaling of the data transferred from the process units to the real values of the measured entity.
- **Data objects.** The data objects register and store sampled or calculated data.
- **Command procedures.** These are SCIL programs, which can be executed automatically or manually.
- **Time channels.** These are objects for automatic time-controlled start-up of data registration and program execution.
- **Event channels.** These are objects for automatic event-controlled start-up of data registration and program execution.
- **Free Type Objects.** The free type objects define user-defined process object types.
- **Event objects.** These objects activate automatic event-controlled updating in pictures. Event objects can be activated automatically or manually, but have no attributes.
- **Variable objects.** The variable objects are temporary lists of attributes and attribute values, used in SCIL programs.

Each MicroSCADA application contains a process database and a report database. The process database contains the process object, scale and optionally free type object application objects types. The reporting database consists of data objects, command procedures, time channels and event channels.

### 1.2.2.4 Process database

**Overview**

The process database is a fast real-time database containing process objects, scales and free type objects. The process communication of the system passes via the process objects. The behavior of each object can be customized.

The process database definitions are stored on disk, but all necessary data is brought into RAM for optimum run-time performance. Dynamic attribute values are normally stored in RAM only, but can be defined to be copied to the disk for selected objects.
Most of the process database functionality is based on the process objects. The process objects are data images of physical process devices, such as valves, pumps, pig-detectors, measurement-devices, metering-stations, etc. The devices are connected to MicroSCADA through remote terminal units (RTUs) and programmable logic controllers, all of which are here called process units with a common name.

The scale objects contain algorithms for the scaling of process object values containing analog data. The scaling algorithm may be one-to-one, linear or stepwise linear.

Free type objects are used to specify user-defined process object types.

The process objects supervise the signals registered in the process units and control the signals sent from the process units to the primary equipment. Generally, each input and output connection in the process units is represented by a process object in the MicroSCADA process database. In addition, other data stored in the process units can be represented by process objects (e.g. the event recording objects in RTUs).

There are also process objects which have no physical correspondence, nor any data correspondence in the process units. These ‘fictitious process objects are used for process simulations, for manually updated values, system message handling, etc.

The process objects in a process database are organized as groups. A group consists of a number of indexed process objects with the same logical name, typically related to the same physical process device.

**Function**

A process object contains the process data (object value), various quality information related to the data as well as alarm state information, and also definitions for scaling, automatic activations, etc.. Both the dynamic data that reflect the real-time state of the process and the definition features are attributes.

A process device is controlled by setting the object value of the corresponding output object with the #SET command.

Each updating of a process object value is stored together with quality and time stamp information. Updating, whether it comes from the process or from SCIL, may activate some of the following actions, ‘post-processing’, (depending on the process object definition and the value of the updating):

- alarm activation (input objects) including alarm signals, alarm list entry and alarm printout
- automatic printout
- event-based updating in picture through event objects
- activation of an event channel.
  Automatic execution of command procedures, data objects and time channels can be connected to a process value change via the event channel. The execution of command procedures via event channels provides the event-based actions with all the power of the SCIL language, enabling event-driven process calculation, control, forwarding of data to other systems, etc.
- registration in the history buffer for event list presentation
- history logging on disk
Figure 6 - The functions of the process database
Process object types

The type of a certain process object depends on the type of the corresponding input/output connection in the process unit. This connection corresponds to the object value (OV) attribute of the process object. MicroSCADA supports nine predefined process object types:

- analog input and output
- binary input and output
- digital input and output
- double indication (input)
- pulse counter (input)
- bit stream (output and input).

Different process unit types (station types) have different ways of naming data types. The following table shows the correspondence between process object types and process unit type specific data types for RTUs and SPACOM devices.

<table>
<thead>
<tr>
<th>Process object types</th>
<th>RTU data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Input, BI</td>
<td>Indication single</td>
</tr>
<tr>
<td></td>
<td>Indication single event recording</td>
</tr>
<tr>
<td>Binary Output, BO</td>
<td>Object command (one- or two-step)</td>
</tr>
<tr>
<td></td>
<td>Regulation command</td>
</tr>
<tr>
<td>Digital Input, DI</td>
<td>Digital value</td>
</tr>
<tr>
<td>Digital Output, DO</td>
<td>Digital setpoint</td>
</tr>
<tr>
<td>Analog Input, AI</td>
<td>Analog value</td>
</tr>
<tr>
<td></td>
<td>Analog event recording</td>
</tr>
<tr>
<td>Analog Output, AO</td>
<td>Analog setpoint</td>
</tr>
<tr>
<td></td>
<td>General persistent output</td>
</tr>
<tr>
<td>Pulse Counter, PC</td>
<td>Pulse counter</td>
</tr>
<tr>
<td>Double Binary Indication, DB</td>
<td>Indication double</td>
</tr>
<tr>
<td></td>
<td>Indication double event recording</td>
</tr>
</tbody>
</table>

Table 1 - Correspondence between process object types and station type specific data types for RTUs
User-defined types

In addition to the predefined process object types listed above, the MicroSCADA application engineer can define his own user-defined types. These process object types are defined by the free type objects.

1.2.2.5 Report database

The report database offers functionality for storing the history of raw and calculated data values as well as for time and event-based execution of SCIL programs, etc. The object types in the report database are data objects, command procedures, time channels and event channels. The object definitions are stored on disk. Dynamic attributes such as object value and registration time can be defined to be stored either in RAM only or also on disk. Due to performance reasons, the MicroSCADA kernel maintains a run-time copy of the most frequently accessed information (both dynamic and static attributes) in RAM.

Data objects sample, calculate, register and store data. A data object can contain one or more registered values. Each registered value has a time stamp and a status code describing the origin and quality of the value. Data objects are used for storing trends, value history, data for system configuration, optimization, calculation, estimations of future values, order-booking etc. Each data registration is done according to a SCIL expression and a logging function. Examples of selectable logging functions are sum, integral, pulse difference, maximum and minimum. The execution of a data registration can be started manually or automatically.

The command procedures contain SCIL programs, which can be started automatically (via time channel, event channel or by other command procedures) or manually. They can be used for all kinds of automatic operations, e.g., calculations, control operations, report printouts, automatic system and communication configuration. Command procedures are, e.g., used for the execution of automatic operations at system start-up. When command procedures are executed by event channels or time channels, they can not handle user interface related operations.

The time channels provide schedules for automatic time-activated start-up of operations in the report database: the registration of data objects and the execution of command procedures. Time channels are used for cyclic program execution or data registration, time dependent reports, trends, regular checks, time control, etc. One time channel can start one or more objects. If a time channel starts several objects, they are started in priority order. A time channel is generally activated at predefined times, either at an absolute point of time or cyclically with programmable intervals. Discontinuous time activation is handled by means of conditions in the form of SCIL expressions.

The event channels are facilities for automatic event-activated start-up of operations in the report database. An event channel can start the registration of data objects, the execution of command procedures and the activation of time channels. Event channels are typically activated by means of process events (changes in the process object values). In other words, the event channels transmit the process events from the process database to the report database where they activate consequential operations. Each event channel activation is queued with a snapshot of the most important attribute values of the activating process object. The event channels can also be activated by SCIL statements.

Up to 15 parallel queues can be defined for concurrent execution of reporting database objects. By default all the queues have the same execution priority, the priority of each parallel queue can optionally be set via a base system object attribute.
1.2.3  **Base tools**

The base tools comprise a selection of tools for building, modifying and viewing system and application components. Tools are included e.g. for the following tasks:

- picture and dialog editing
- SCIL program editing
- application object management: viewing, modification, creation of
  - process objects
  - scales
  - event channels
  - data objects
  - command procedures
  - time channels
- testing of application functionality
- system object creation, modification and viewing
- searching strings from SCIL programs in the application

Part of the tools in rev. 8.4 are completely new, while others have been extended or copied from rev. 8.2.

1.2.4  **Application engineering**

MicroSCADA application engineering is a procedure for building an application consisting of pictures, dialogs, SCIL programs and necessary object definitions, database objects as well as system objects.

The MicroSCADA kernel offers flexible support for the application engineering. The most efficient application engineering method is the library method, which is based on the application software library LIB 5XX (described in the section LIB 5XX Application Engineering). The idea behind the library method is to put together customer applications by installing pre-made standard functions. The standard functions cover picture components as well as required object definitions.

The library method is recommended to be applied for all application areas, which it supports. Even if an application area is not supported by LIB 5XX, usually at least the LIB 5XX base (including e.g. alarm and event lists) can be utilized as such. Additionally, if the same kind of functions are to be used many times in one or several customer applications, the most efficient approach is generally to start with building appropriate standard functions, which are then installed (reused) where required.

The SCIL language offers almost unlimited possibilities to extend the application functionality. Dialog building with Visual SCIL, calculations, file handling, conversions, external communication via different protocols, communication to Windows applications via DDE and ODBC are just a few examples. However, when utilizing the possibilities of SCIL, it should always be kept in mind that the extent and the performance requirements of the programmed application functionality should match the performance provided by the base system computer.
1.3 LIB 5XX application engineering

1.3.1 LIB 5XX structure

1.3.1.1 Concept

The LIB 5XX Application Software Library consists of product packages for building the applications. It includes pictures, dialogs and required application objects for local and remote monitoring and control of Oil and Gas Pipeline Networks but also power transmission, distribution substations etc. LIB 5XX is aimed for local engineering centers to make it easier, faster and safer to engineer systems for their local customers.

The LIB 500 Base package includes the common application platform functionality and is always required when the Application Software Library is used.

The LIB 5XX Application Software Library consists of different product packages (LIB 510, LIB 540 etc.) for building the applications. It includes pictures, dialogs and required application objects for local and remote monitoring and control of Oil and Gas Pipeline Networks but also power transmission, distribution substations etc. LIB 5XX is aimed for local engineering centers to make it easier, faster and safer to engineer systems for their local customers.

LIB 5XX consists of following packages:

- The LIB 500 Base package includes the common application platform functionality and is always required when the Application Software Library is used.
- The LIB 540 pipeline process package is an engineering tool for building SCADA applications for Oil and Gas Pipelines and other processes.
- The LIB 510 medium-voltage process package for medium-voltage substation and distribution automation. This package is also suitable for managing power supply facilities of Pipeline plants.
- Some other packages for High voltage station automation.

<table>
<thead>
<tr>
<th>Pipeline Process</th>
<th>MV Process</th>
<th>Other Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIB 540</td>
<td>LIB 510</td>
<td>LIB ...</td>
</tr>
<tr>
<td>• Pipeline process</td>
<td>• MV process</td>
<td></td>
</tr>
</tbody>
</table>

LIB 500 Base

- Back Bone functions
- Application setting
- Help Dialog
- User Management
- Installation tool

- Event & Alarm Handling
- System Self Supervision
- Language converter

Figure 7. Library package structure
1.3.1.2 Application engineering

Today there are two complementary application engineering methods to be used when creating MicroSCADA applications. One method is to build the application from the beginning using the SCIL language and the other one is to use the LIB 5XX Application Software Library. The SCIL method gives total flexibility but suffers from lacking standard solutions. The training and know-how level of the engineers ought to be high and the testing of the application is extensive. On the other hand, when using the LIB 5XX method the application engineering is performed by composing LIB 5XX standard functions to pictures. The application objects connected to various process functions are automatically created together with the graphical presentations and dialogs when the standard functions are installed in the application pictures. All the functions are type-tested and the complexity hidden in standard functions allows a lower entry level for the engineering. In many cases, when specific customizations are needed, a combination of the LIB 500 and the SCIL methods is used.

1.3.1.3 Standard functions

By using standard functions, several types of process pictures may be created in an application. A standard function may contain all available picture elements, such as background, start, update and exit programs and function keys. The only restriction, in technical respect, is that it may not contain another standard function. The visual background may, in principle, be of any size. Standard functions are created with the picture editor.

The standard functions are grouped and organized in a menu structure in accordance with their purpose. The LIB 500 Base package is used as a base for the application design. The process pictures and the data objects are generated by installing standard function building blocks into the base picture containing the header and the base functionality.

Figure 8 - An illustration of the application engineering using the LIB 5XX standard functions
1.3.1.4 Picture function

The application pictures are composed of picture function elements (also other picture elements may be included). When a standard function is installed into an application picture, it becomes a picture function.

A picture function is a part of the application picture though it has its own picture elements such as picture background, start, update and exit programs, function keys and windows. Each picture function is equipped with a name that must be unique within the picture. A picture function may be repositioned. All elements of a picture function can be viewed.

1.3.2 LIB 500 Base components

1.3.2.1 Backbone functions

An important backbone function is the so called base standard picture function witch is used in all LIB 5XX applications regarding menu configuration and default links from the default menu items.

The base picture which acts as a background for the application pictures contains a header with time and date presentation, pull-down menus and some short-cut pushbuttons. The pull-down menus may be edited.

In addition, the back bone package includes a start-up picture, accessory pictures, the picture for password handling, automatic printout picture, basket picture and note book.

![Figure 9 - Some backbone pictures](image-url)

Features

- title bar
- date and time presentation
- toolbar short-cuts to event and alarm lists
- alarm indication
- alarm row drop-down for alarm acknowledgment
• local/remote indication
• system debugging tool
• application setting tool
  – language settings
  – time-out for log-ins
  – process control settings
    – close/keep dialog after control execution
• print/do not print user name on control execution
• note marker
• help in all dialogs
• User management tool
  – The application framework contains a user authorization mechanism with four user levels:
    – VIEW LEVEL. The operator is allowed to view the application, but is not allowed to
      make control operations, nor to use the programming and system configuration tools.
      Default setting for new users is the view level.
    – CONTROL LEVEL. The operator is allowed to make control operations, but has no
      access to the engineering and system configuration tools.
    – ENGINEERING LEVEL. All rights are granted, excluding user management.
    – SYSTEM MANAGEMENT LEVEL. This is the system manager level. All rights are
      granted, including the rights to add and remove users. Each application has one and only
      one system manager.
  – Each application object can be grouped to programmable authorization groups. For each user it
    is possible to give a certain user level per authorization group. If an object is not defined to a
    specified authorization group, it belongs to the general group.
  – In addition, a workstation with a unique IP address can be programmed to VIEW mode only. If
    this function is not programmed, the normal authorization mechanism will be used for the
    workstation.

1.3.2.2 Alarm list function

The alarm list displays the present alarm situation of the supervised process. Each alarm is presented as an
alarm predefined text line which describes the alarm in the process. The alarm text line consists of a time
stamp, object identification, a signal text and a text indicating alarm status. The alarms are shown in
chronological order. The newest alarm can be presented on the top or at the end of the page.

Alarms are presented in two separate lists: the upper one contains active acknowledged and unacknowledged
alarms (persisting alarms), while the lower one contains inactive unacknowledged alarms (fleeting alarms). Both lists can be scrolled.
Figure 10 - Alarm list, template 1

Each alarm is presented as a single alarm text line. This text line consists of a time stamp (date and time), object identification, signal text and status text. The length and position of each text column can be defined within certain limits. The color of each alarm type can be selected from a predefined set of colors. The default colors and related status texts of the presented alarm types are as follows:

<table>
<thead>
<tr>
<th>Alarm type</th>
<th>Default color</th>
<th>Status text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active unacknowledged*</td>
<td>Red</td>
<td>Alarm</td>
</tr>
<tr>
<td>Active acknowledged*</td>
<td>White</td>
<td>Ack.</td>
</tr>
<tr>
<td>Inactive unacknowledged**</td>
<td>Green</td>
<td>Normal</td>
</tr>
</tbody>
</table>

* = included in persisting alarms
** = included in fleeting alarms

Table 2 - Default colors and status texts in alarm types
Acknowledgment of a single alarm is done by clicking the line of the desired alarm on the list. If the selected alarm is unacknowledged, the line is shown highlighted and the acknowledgment dialog is opened. At the same time, the alarm list is set to frozen mode to prevent unwanted scrolling.

*Figure 11 - The acknowledgment dialog*

Conditions are used when the user likes to concentrate on specific information contained by the alarm buffer. This is done by introducing a criteria to filter out unwanted information. A single criteria or multiple criteria can be used. These criteria can be freely combined to create grid-like filters. Station, bay and device names can be saved into an application-specific text file.

**Features**

- two types of alarm list templates
- time-based and object-based conditions
  - time (lower and upper time limit, one second accuracy)
  - five-level object identification structure
  - function type
  - alarm class
- persisting and fleeting alarms separated
- alarm list setting tool for colors and text layout
- updating/frozen presentation modes
- alarm acknowledgment
  - single object
  - all objects
- alarm reset function
- authorization support
- help function available in all dialogs
1.3.2.3 Event list function

The event list presents the process events from the monitored process in time order. Each event is normally presented by displaying a predefined event text line which describes the event in the process. Event text lines consist of a time stamp, object identification, a signal text and a text indicating status. The events are presented in chronological order so that the latest event appears on the bottom line of the first page. The event list contains keys for browsing the list forward and backward.

Events are normally stored in the history buffer in the computer’s RAM memory, and they are also stored on the computer hard disk.

Figure 12 - Event list, main view

The default event text that describes the event consists of:

- status sign
- time stamp
- object identification
- signal name
- status text
In some applications, there is a need to change or to extend the text. It is then possible to configure the length of the columns, the order of the columns and the column labels.

Conditions are used when the user likes to concentrate on specific information in the event list. This is done by introducing criteria to filter out unwanted information. These criteria can freely be combined to create grid-like filters. Station and device names can be saved into an application specific text file.

**Features**

- The event list layout is configurable.
- Printouts are configurable.
- The event log file is configurable to day file, week file, month file, or year file.
- Supervision of available hard disk space. The default limit value is 50 MB free space.
- Scroll functions such as go to first page, scroll one line, scroll one page, scroll one interval or standard scrollbar.
- Time-based and object-based conditions:
  - time (lower and upper time limit, one second accuracy)
  - five-level object identification structure
  - function type
  - alarm class
- updating/frozen presentation modes
- authorization support
- help function available in all dialogs

**1.3.2.4 Blocking list function**

The blocking list is a summary display of the present blocking situation in the process which is supervised. Each blocked signal is presented as a predefined signal text line. In addition to the signal text, a check box indicates the blocking status.

An object-blocking procedure is done by choosing it from the application picture or by request from the process database and consequently activating the chosen blocking condition. Automatic blocking procedures are possible as well. For instance, when the station is taken into local position, all objects connected to the station can automatically be alarm-blocked.
Figure 13 - Blocking list example

The MicroSCADA version 8.4 provides a wide range of blocking attributes, which all are included in the blocking list as well. To provide a blocking handling mechanism in a more clear and rational way, the following blocking types are provided by the LIB 500 blocking list:

- alarm blocking: alarms are not raised, regardless of object state
- update blocking: indications are not updated from the process
- control blocking: operation commands are not sent to the process
- event blocking: event registrations are not made, events are not shown in the event list.
- action blocking: event channel activation is blocked

Features

- easy signal selection
- selection of signals for blocking/deblocking
- blocking list setting tool
1.3.2.5 Trend reports

The trend reports can be presented in graphical mode as full-graphic curves or in tabular mode. These two modes share the same process data, but otherwise these modes can be used independently.

The graphical form of the LIB 510 trend picture contains up to ten curves from the process data log. They are presented as full-graphic curves on a two-dimensional coordinate system that consists of horizontal time (X) axis and vertical value (Y) axis. The curves can be scrolled in both directions, X and Y, and the parameters of both axis can be changed, as well as the line-scaling parameters of the trend curves. All curves can temporarily be erased from the screen.

Figure 14 - Graphical presentation of a trend
ABB Process Industries

Features

- graphical trend presentation of up to 10 curves
- tabular trend presentation of up to 10 columns
- hairline function
- color configuration
- curve object identification
- line styles
- scaleable axis
- scrolling in X and Y direction
- on/off switching of each curve
- process data logging activation from station picture
- calculation formulae; direct, mean, sum and difference
- trend data saving to file (possible to import to e.g. Excel)
- zoom function
- save/open preconfigurations
- possibility to enter values manually to a trend
- printout option
- authorization support
- help function available in all dialogs

1.3.2.6 Measurement report function

This standard function is used when building measurement reports such as pressure, flow, density etc. reports within LIB 500 applications.

All data for the reports are calculated and stored in real time. Report data are collected and calculated either at certain points of time or after certain events. The most common method is to fetch raw data from the process, thereafter refine it and store it in the report database.

Collection and calculation of report data as well as printout of reports can be initiated in the following ways:

- at predefined time intervals
- when a predefined event occurs
- as a result of a calculation
- based on a condition
- on the operator’s request
Figure 15 - Report example

- Following standard functions are included:
  - time columns
  - measured columns
  - measured columns, double tariff
  - total column
  - total column, double tariff
  - basic column
  - mid column
  - peak column
  - excess column
  - 1h max. column
  - 3h max. column
  - matrix column
  - matrix column, double tariff
  - day sum column
  - night sum column
  - max. day column
  - max. night column
1.3.2.7 System self supervision function

The system self supervision function provides standard functions for the supervision of system objects. The system overview displays the status information of the hardware, system components, communication equipment and process control and automation units.

The status information of hardware and software can be displayed by both color information or numerical information. The status information can also be stored in the history buffer and be set to give alarm at critical values or limits.

Figure 16 - System self supervision

1.3.3 LIB 540 Pipeline Process

LIB 540 contains the standard software package for engineering SCADA systems for Oil and Gas pipeline industry. The package can be also used for other processes like automation of chemical plants and water distribution etc.
Purpose

The package consists of a library of MicroSCADA standard functions for visualization and control of field instruments and devices in the pipeline environment like valves, pumps and measurements. It supports the protocols RTU 2xx RP570, SPACOM and IEC 870-5-101. MODBUS and PROFIBUS are also available by gateways.

The task of the Pipeline process components is to monitor the processes and control the primary equipment. Most of the Pipeline process standard functions are shown in Figure 17. For example station, group, On/Off valve, Pump, Control valve, Control unit, level indicators, measurement functions and Alarm Indicator are represented.

Figure 17 - A process picture in form of a single line diagram
Pipeline Process Standard Functions

Pipeline Process contains following standard functions:

- Station
- Updating of data from the process to the station
- Station blocking/deblocking
- Local/remote handling
- Updating of data from the process
- blocking/deblocking of devices
- Group interlocking
- On Off valve device (“Standard Valve”)
- Control valve
- Pump
- Control Unit
- Measurements
- Up to four measurements per standard function
- Minimum and maximum value presentation
- Bar/unit presentation
- Zero dead band settings
- Fast trend curve presentation
- Level indicator
- State Indicator
- Operation switch

Common Functionality

Common Functionality included in most standard functions:

- Versatile configuration options
- Support of IEC 870-5-101 protocol
- Alarm state
- Blockings
- State settings
- Forced operation
- Operation countings values presented
System Description

MicroSCADA PMS

- Predefined object messages
- Operation simulation function
- Authorization handling
- Help function available in all dialogs

Pipeline Process Engineering

The Pipeline process library includes standard functions needed to build the single-line process applications. The engineering is divided into two subtasks:

- Single-line editing (static picture background editing and standard function installation)
- Single-line configuration (standard function configuration and creation of application objects)

The engineering of the substation single-line diagram is done either by creating a new application or by rebuilding an existing single-line process application.

After installation of the base picture, the static background of the single-line process picture is drawn with the picture editor. After that the Pipeline process standard functions are installed in the picture one by one.

Most of the Pipeline process representation pictures of the standard functions are available in three sizes. In the representation editor various different symbols and their properties like line thickness, color and orientation can be chosen.

The structure of the Pipeline process standard function is hierarchical, normally based on three levels: Station, Group and Device. Device comprises all the other Pipeline Process standard functions, except from Station and Group. One or more devices belong to a group (use the same group name). Local / remote handling is done by the group standard function. This means that all devices of a group are set to local resp. remote as a whole. All devices belonging to the same station (e.g. pump station) are using the same station name.

Controlling Objects

Control of the devices (valve, pump etc.) from the operator workplace is made via control dialogs. This control pictures are automatically displayed when the device to be controlled is selected. To prevent unauthorized control, authority handling control protects window call up. The control window is also coupled to the control hierarchy so that a control command to the same object cannot be output from two different workplaces at the same time. An example of a single line diagram and control dialogs opened from it is shown in Figure 18.
The standard functions for the pumps and valves are used for controlling and monitoring the devices. Their main functions are position indication, operation, and status information. Position indication is shown with different symbols while the status is normally shown with different colors. In addition to the color coding message dialogs inform about abnormal status.

The control of the device is done in a control dialog, which is opened when the standard function symbol is clicked. Viewing possible options and opening or closing the object is done by clicking the corresponding buttons. Before a selected operation is carried out, the user is asked to verify it.

### Blocking Functions

Blocking conditions may be selected on station level, group level and on device level. Furthermore, there are dialogs for control, blocking, monitoring and setting functions for all standard functions used as building blocks for the single-line process picture. See Figure 19.
1.3.3.1 Basic Standard Functions

Station and Group standard functions are used for building the hierarchical structure of the Pipeline processes (see Figure 20). See also the section Pipeline Process Engineering.

![Station and Group Standard function](image)

**Figure 20 - Station and Group Standard function**

1.3.3.2 Standard Valve Device

The Standard Valve Device contains visualization and control functions for controlling of different kinds of On/ off valves with versatile configuration possibilities. Valves with or without Stop functionality can be operated. The state of not motorized valves is presented. The valve position and if available valve running indications are shown as different symbol. The valve can be configured with or without Automatic/ manual and Local/ remote functionality. Time supervision of operations can be activated etc.

Figure 21 shows different representations of the valve.
1.3.3 Standard Pump Device

The Standard Pump picture function contains visualization and control functions for controlling of different kinds of On/ off pumps (main pumps as well as booster pumps). The pump can be Start/ Stop controlled. Automatic/ manual and Local/ remote functionality is available. Time supervision of operations can be activated.

Figure 23 shows different representations of the valve.

In Figure 24 the main dialog is presented coming up if the symbol button is pressed.
Also status information from the local control unit during a sequence startup of a pump are shown on the station picture.

Additional status signals like torque, motor high temperature and vibration alarms etc. are displayed separately with the alarm indicator standard function described below.

### 1.3.3.4 Control Valve Device with Control Unit

The Control Valve Device with control unit standard function operates e.g. Pressure Control Valves with local regulator (PCV). Switching between automatic and manual mode is possible. In automatic mode the set point (rated value) for the local regulator can be given. The controlled feedback value (e.g. Pressure measurement) is displayed on the dialog. In manual mode the valve position can be controlled by setpoint between 0 and 100% or via step open and close pulse. If available the analog valve position can also be presented.

The Control valve standard function can be configured with limit switches and/or analog position information.

Figure 25 shows a typical PCV installation with control unit, control valve and feedback measurement.

Figure 25 - The On off valve standard function

Figure 26 the control unit main dialog for operating a PCV is presented.
1.3.3.5 Monitoring Functions

- Measurements
- Level indicator
- Alarm Indicator

Measurements

Figure 26 - The On/off valve standard function main dialog

Figure 27 - Three types of measurement symbols

Measurement function provides an easy and fast way to monitor the processes on the picture. The measurements are presented as numerical values, as bars or just as a measurement symbol depending on the configuration. The advantage by presenting the measurements as bars is that the alarm and warning limits are shown together with the measurement, hence the level of the measurement compared to
the limits is very obvious. Different colors show the states normal, warning or alarm of the measurement.

By clicking the measurement symbol the measurement dialog is opened. Different types of measurement symbols are shown in figure 27. From this dialog the sub functions can be opened or settings can be changed. Each measurement function consists of 1 to 4 measurement values of the analog input type or pulse counters for energy measurements. The measurement also contains sub functions listed in Table 3. The measurement dialog is shown in figure 28.

<table>
<thead>
<tr>
<th>Subfunction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm state</td>
<td>Shows persisting or fleeting alarms for the measurements and the unacknowledged alarms can be acknowledged.</td>
</tr>
<tr>
<td>Blockings</td>
<td>Makes blockings for the database in MicroSCADA</td>
</tr>
<tr>
<td>Edit Limits</td>
<td>Sets the alarm and warning levels for the measurements in the MicroSCADA database and sends in some cases the limits down to used device.</td>
</tr>
<tr>
<td>Fast Trending</td>
<td>Enables fast trending of the measurement and can be used as small or zoomed Fast Trend. Settings can be made from a separate Settings dialog in the zoomed FT.</td>
</tr>
<tr>
<td>Dead band</td>
<td>Setting of the zero dead band for the measurement</td>
</tr>
<tr>
<td>Object messages</td>
<td>Presents the object state.</td>
</tr>
</tbody>
</table>

### Features

- Installation to process picture
- Up to four measurements per standard function
- Bar/Unit presentation
- Measurement Dialog
- Minimum and maximum values presented
- Alarm state
- Blockings
- Edit limits
- Fast trending
- Dead band settings
- Object messages
Fast Trending

Pipeline process function Measurements contains a sub-function called Fast Trending, shown in figure 29. It is used to show value of the selected data on screen.
figure 29 - A station with a small and a zoomed Fast Trend

Level indicator

Level indicator function has the same functionality as measurement. Additionally the tank level is shown as a vertical bar including limits. Width and height of the bar can be verified. Different background pictures (e.g. globular or cylindrical tank pictures) can be selected (figure 30).

figure 30 - Level Indicator as tank measurement
Alarm Indicator

Alarm indicator is aimed for monitoring the alarming state of input signals. The status of each signal can be seen in the representations shown on the single line diagram (figure 31) or right-hand side of the main dialog (figure 32).

**figure 31 - Different symbols for the alarm indicator standard function**

Two basic types are available: The single alarm indicator for presentation and handling of one alarm and the collective alarm indicator handling up to 8 alarms. The main functionality is as listed below:

- Status information
- The Alarm indicator has the following sub-functionality:
  - Time stamp for last alarm
  - Alarm state/acknowledgement
  - Blocking/deblocking of update, alarm, event, printout and reprocessing within the database.
  - Simulation

**Functionality**

figure 32 presents the main control dialog for the Collective Alarm indicator which is opened by clicking the picture function. It is aimed for monitoring the alarming state of input signals. The status and alarm class of each signal can be seen in the representations shown on the right-hand side of the main dialog.
The object identification of each signal connected to alarm indicator is shown beside the status symbol res. In the upper part of the window.

For the single alarm indicator the time stamp of the last alarm state change is shown. For the collective alarm indicator this information is presented when clicking the button beside the signal text (see figure 33).
figure 33 - The last alarm dialog for the Alarm indicator

The object identification of the signal connected to alarm indicator is shown on top of the dialog. The field "Last Alarm" shows date and time of the last alarm (resolution milliseconds).

1.3.4 Flow Visualization

The pipelines e.g. of a tank farm or pump station often form a more or less complex network with many possibilities to lead the product. The flow visualization will color passing trough pipeline segments so it is easy for the operator to see if e.g. the pipeline to a tank is free or still blocked by valves (figure 34).
figure 34 - Flow visualization in a pump station