Anybus®
BY HMS NETWORKS

Anybus Edge Gateway

Reference Manual

HMS Connecting Devices™

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Industrial Networks of any inaccuracies or omissions found in this document. HMS Industrial Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Industrial Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Industrial Networks and is subject to change without notice. HMS Industrial Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

1   Preface

1.1   About This Document

This manual describes the installation and configuration of Anybus Edge gateway.

For additional documentation and software downloads, FAQs, troubleshooting guides and technical support, please visit www.anybus.com/support.

1.2   Document Conventions

The following conventions are used to indicate safety information and other important content in this document:

⚠	**WARNING** Instruction that must be followed to avoid a risk of death or serious injury.

⚠	**Caution** Instruction that must be followed to avoid a risk of personal injury.

❗	Instruction that must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.
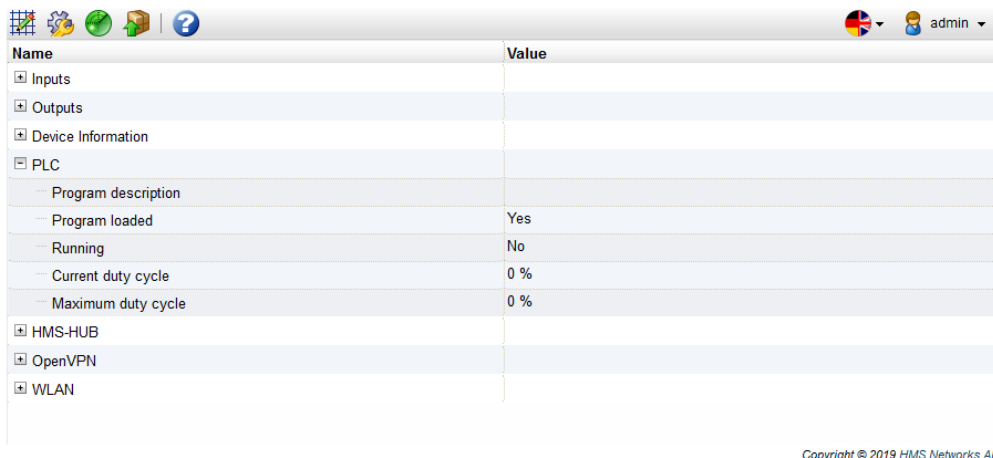
ⓘ   *Additional information which may facilitate installation and/or operation.*

1.3   Trademarks

Anybus® is a registered trademark of HMS Industrial Networks. All other trademarks mentioned in this document are the property of their respective holders.

2   Home Page

On the home page you can see information on the device, its I/Os and several other components.

2.1   Login Button

The login button is used to log in or out and to change your password. If you are not asked to log in when loading the website, you can use the login button to do so. Press the login button and chose *Log in* from the appearing menu. A dialog shows up through which you must provide your user name and password. When you are logged in, the menu of the login button allows you to change your password or to log out. When you choose *Change password* a dialog with the following input fields shows up:

Old password	Enter your previous password here.
New password	Enter your new password here. The requirements for the new password are shown below the input field.
Repeat password	Repeat your new password here to detect typing errors.

3   HMS Sequence Editor

The HMS Sequence firmware provides an editor through which you can edit the program of the PLC. The PLC program consists of two components. The diagram allows programming similar to the well-known CFCs (Continuous Function Charts). Here you can connect the device's inputs and outputs as well as variables through function blocks. The I/O mappings allow you to quickly map inputs to outputs directly. It is also possible to use variables as inputs or outputs.
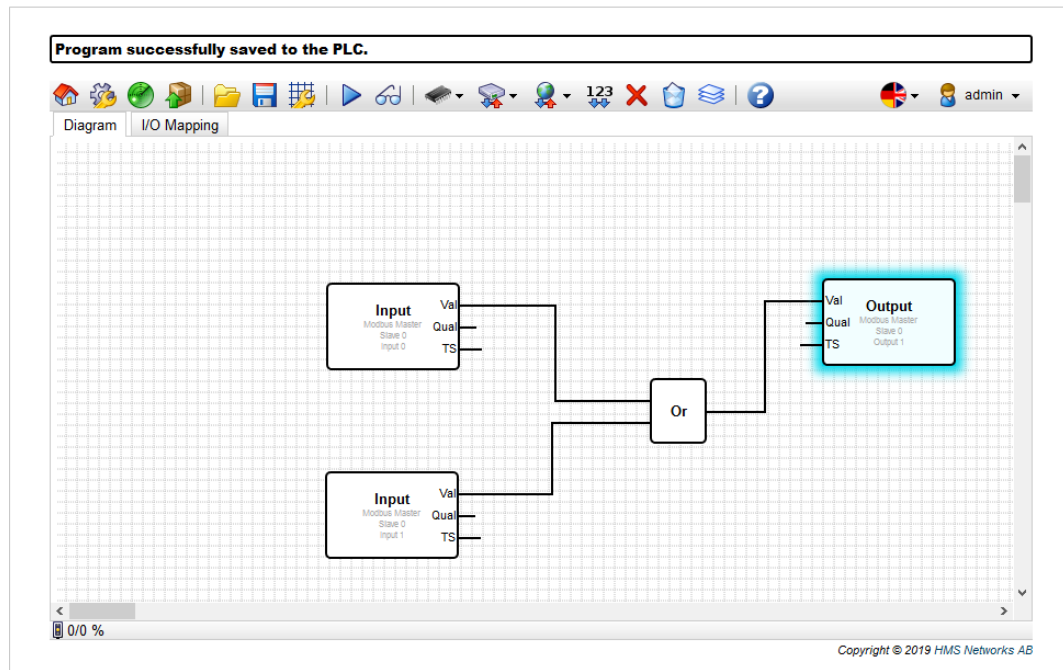
Fig. 1   Screenshot of the HMS Sequence editor

The image shows a screenshot of the HMS Sequence editor. It basically consists of three parts.

- At the top you will find a *Toolbar*.

- Below the toolbar there are the *Diagram* and the *I/O Mapping*. You can switch between those two through the tabs at the top.

- At the bottom there is a *Status Bar* that shows information on the state of the PLC.

3.1   Toolbar

The toolbar of the HMS Sequence editor contains buttons that can be used to navigate to other sites, to edit, load and store the diagram and the I/O mappings and to control the PLC.

In the following the buttons will be described in detail.

	Closes the editor and opens the home page. See *Home Page, p. 4*.
	Closes the editor and opens the settings page. See *Settings, p. 26*.
	Closes the editor and opens the diagnostics page. See *Diagnostics, p. 124*.
	Closes the editor and opens the update page. See *Update, p. 127*.
	Loads the program (diagram and I/O mappings) from the PLC. Local changes will be overwritten.
	Saves the current local program (diagram and I/O mappings) to the PLC. If the PLC is currently running, it will be stopped.

	Opens a dialog where you can edit the program settings. See *Program Settings, p. 22*.
	Starts/stops the PLC. The PLC can only be started if a non-empty program (diagram and/or I/O mappings) has been saved.

The following buttons are only available when the diagram tab is selected:

	Starts/stops *Live View, p. 24*.
	Shows/hides Processing Order. See *Diagram, p. 7*.
	Opens a menu where you can select a unit type to be added to the diagram view. See *Diagram, p. 7*.
	Opens a menu where you can select inputs or outputs to be added to the diagram view. The menu consists of several sub-menus for inputs/outputs, interfaces, devices and arrays. Note that for arrays you can either insert an I/O array element by selecting the topmost entry from the respective sub-menu or a plain I/O element representing just one value of the array by selecting the corresponding index from the respective sub-menu. See *Diagram, p. 7*.
	Opens a menu where you can select a portal variable to be added to the diagram view. See *Diagram, p. 7*.
	Adds a constant to the diagram view. See *Diagram, p. 7*.

The following buttons are only available when the I/O mapping tab is selected:

	Opens a dialog through which you can add a new I/O mapping. See *I/O Mapping, p. 21*.
	Opens a dialog through which you can edit a selected I/O mapping. See *I/O Mapping, p. 21*.
	Deletes the selected diagram element or I/O mapping respectively.
	Discards the whole diagram or all I/O mappings respectively.
	Opens the help system in a new window.
	Login button

3.2   Diagram

The function diagram is composed of up to 2048 units, local I/Os, variables and constants.

Elements are added to the diagram through the *Toolbar*. To select an element simply click on it. The selection will be indicated through a blue shadow drawn around the element. By clicking the Delete button in the toolbar or pressing the *Delete* key on your keyboard the selected element can be deleted. To move an element on the diagram view drag it like shown in the image below.
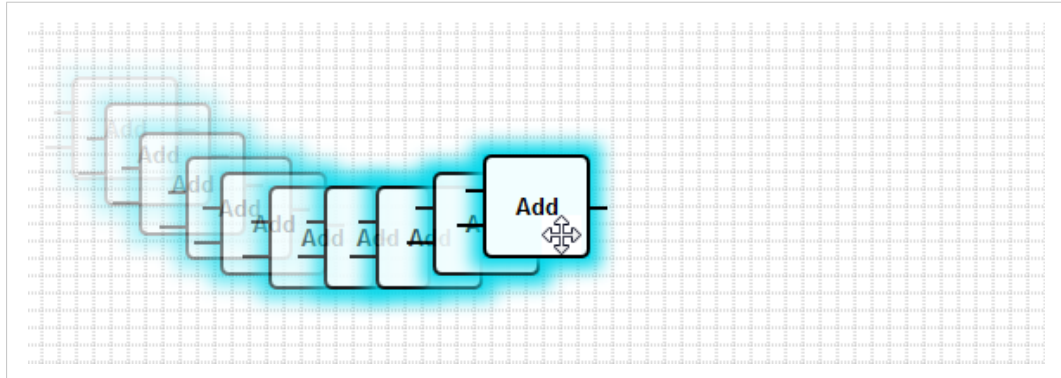
Fig. 2   Dragging an element on the diagram view

Each element has one or more ports through which it can be connected to other elements. Input ports are displayed on the left side of the element, output ports on the right. Each port has one of the following data types:

•   Boolean [0..1]

•   8-bit unsigned integer [0..255]

•   8-bit signed integer [-128..127]

•   16-bit unsigned integer [0..65,535]

•   16-bit signed integer [-32,768..32,767

•   32-bit unsigned integer [0..4,294,967,295]

•   32-bit signed integer [-2,147,483,648..2,147,483,647]

•   32-bit floating point number

•   64-bit floating point number

•   String (up to 4096 bytes)

Note that when data types are compared to each other, this will be done according to this list. *Boolean* is considered the smallest data type, *String* is considered the greatest data type. If ports with different data types are connected to each other, the value will be converted from the output port's data type to the input port's data type implicitly. Note that this may lead to loss of data. Open input ports are defined to have the value *0* (Boolean data type). To negate a port, double-click on it. The negation will be indicated through a bubble as can be seen in the image below. The negation is performed logically. A value of *0* will be negated to *1*. A value unequal to *0* will be negated to *0*. Negating a string will result in an empty string.
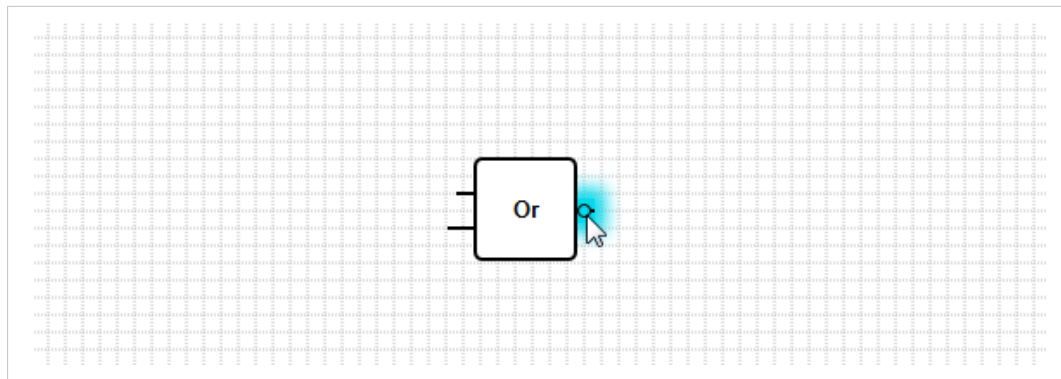
Fig. 3   An element with a negated output port

To connect an input port with an output port click on one of the ports, hold the mouse button pressed, drag the appearing blue connector over the other port (See below) and then release the mouse button. Note that one output port can be connected to several input ports. But it is not possible to connect two input ports or two output ports with each other. Connectors can be selected just like any other element. When selected they can be deleted by clicking the Delete button in the toolbar or by pressing the *Delete* key on the keyboard.
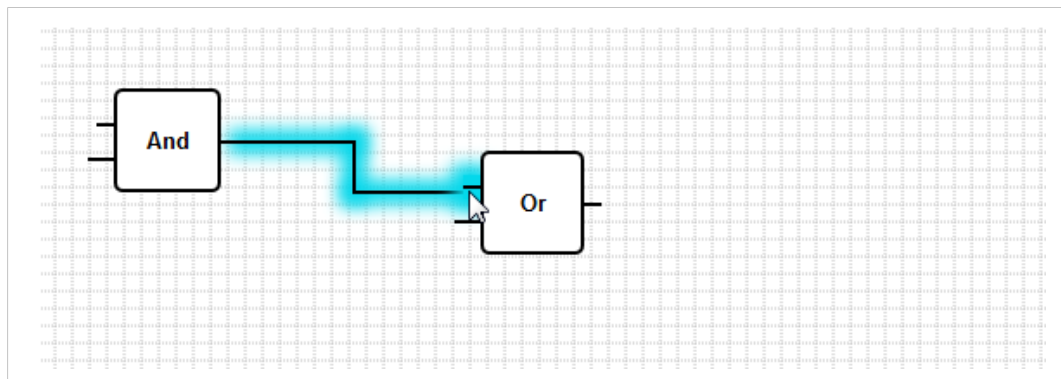
Fig. 4   Connecting an input port with an output port

Some elements allow to dynamically add or delete ports. To do so, right-click on the element or double-tap the element if touch functionality is supported. A pop-up menu will appear (See below). Here you can add and delete ports. Note that there is a minimum and maximum number of ports for each element. You can also jump to the help topic for the specific element through this pop-up menu.
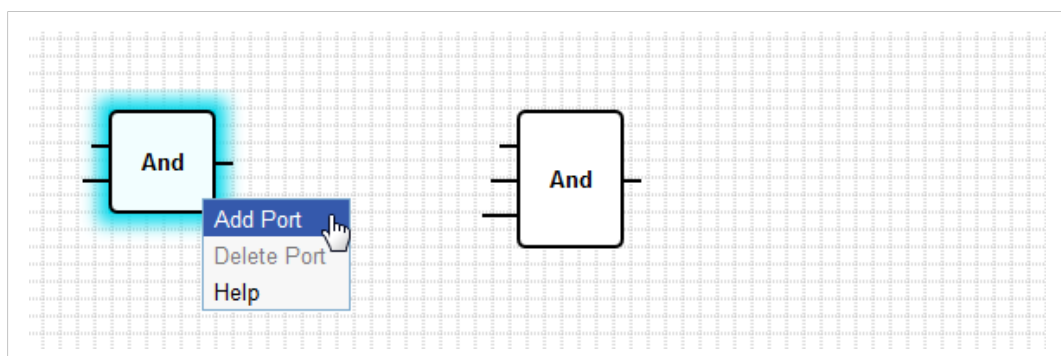
Fig. 5   Adding an input port to an Add Unit

The elements on the diagram are processed one after the other. To see the processing order, click on the Processing order button on the toolbar. The HMS Hub editor will show the processing order of each element in a small box in its upper right corner. The element with the processing order number 0 is processed first. Click the same button again to hide the processing order.
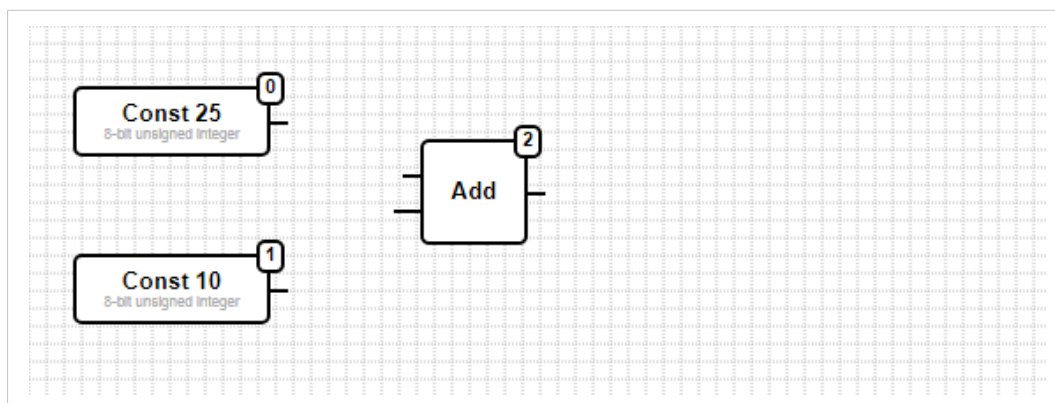
Fig. 6   **Processing order of elements**

The processing order can be changed. To do so, right-click on the element, or double-tap the element if touch functionality is supported. A pop-up menu will appear (See below). Select the entry *Change Processing Order*.
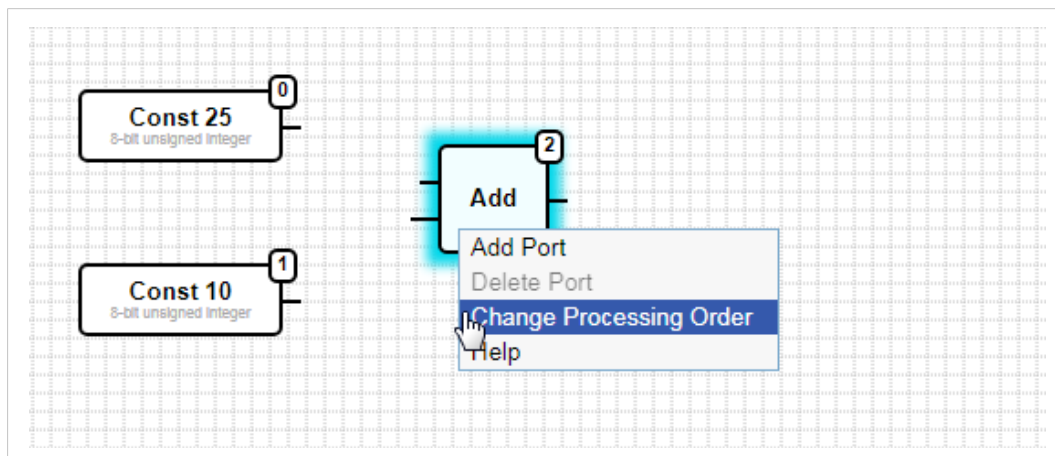
Fig. 7   **Selecting the element to change the processing order**
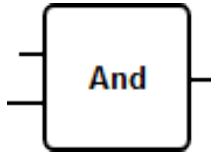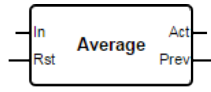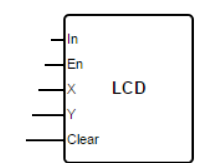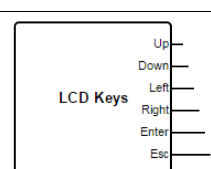
In the appearing dialog, the processing order value can be changed, ranging from 0 (processed first) to the number of elements minus one (processed last).
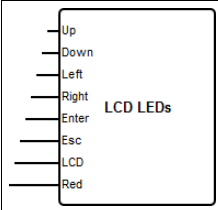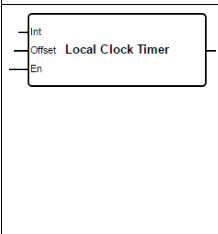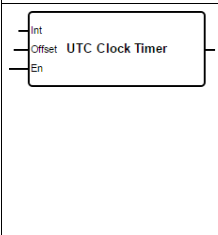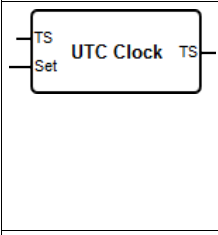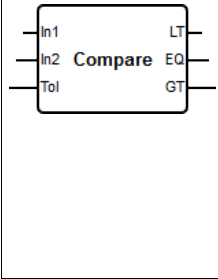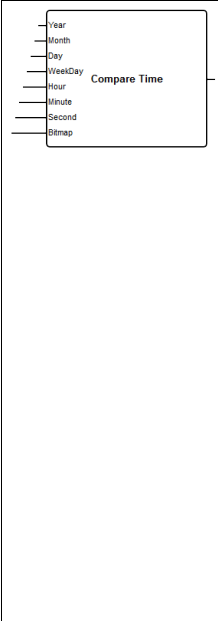
Fig. 8   **Changing the processing order of an element**

3.2.1   Units

In the following, the behavior of the available units will be described. If not stated otherwise, units are processed during the process stage of the *IPO Cycle*.
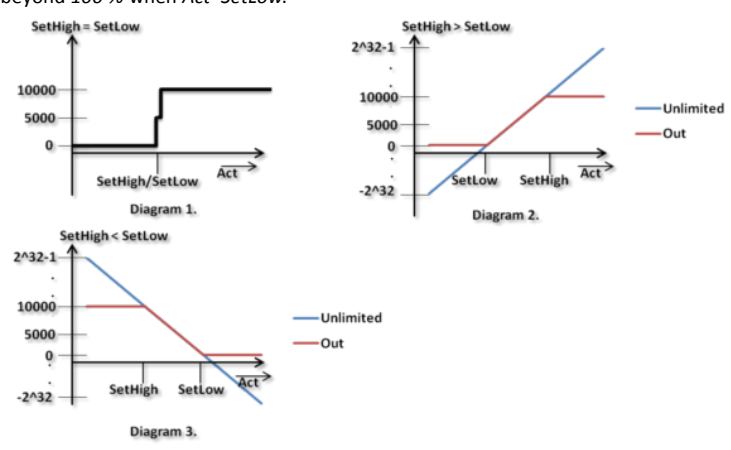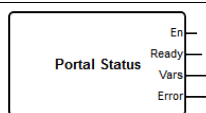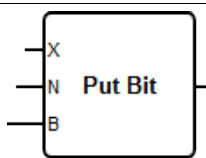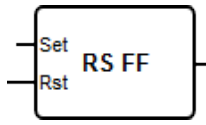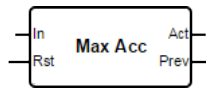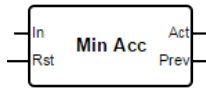
Add	This unit performs an addition. It adds the values of its input ports and writes the sum to its output port. The addition is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type. Input ports connected to string output ports will be treated as if they were connected to a 64-bit floating point output port. If. for example. one input port is connected to an 8-bit <u>unsigned</u> integer output port and another one to a 8-bit <u>signed</u> integer output port, the addition will be performed in 8-bit <u>signed</u> integer and this will also be the data type of all ports.
In Store Threshold Auto **Analog Trigger** Active Diff Ref	This unit compares the value of its *In* input port value with the stored reference value, which is put out on the *Ref* output port. The absolute difference of these two values is put out on the *Diff* output port. If this difference reaches or exceeds the value of the *Threshold* input port, the unit triggers. Then the *Active* output is set to *1* for one cycle. With a rising edge on the *Store* input port the current value of the *In* input port will be stored as the new reference value. If the value of the *Auto* input port is not *0*, the value will be automatically stored when the unit triggers.
And	This unit performs a binary AND. The AND is performed in the greatest <u>integer</u> data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type. Input ports connected to floating point or string output ports will be treated as if they were connected to a 32-bit signed integer output port. Examples: • 0 and $0 \Rightarrow 0$ • 0 and $1 \Rightarrow 0$ • 1 and $1 \Rightarrow 1$ • 11 (1011_2) and 14 (1110_2) $\Rightarrow 10$ (1010_2) • 5 (00000005_{16}) and -15 ($FFFFFFF1_{16}$) $\Rightarrow 1$ (00000001_{16})
In Rst **Average** Act Prev	This unit calculates an average value. It samples its *In* input port each IPO cycle and adds its current value to the average. The current average value is put out on the *Act* output port. A rising edge on the *Rst* input port resets the average. The previous average value is put out on the *Prev* output port until the next reset occurs. The reset is performed before the current value of the *In* input port is evaluated.
TS Set **Local Clock** TS Err	This unit can be used to get the current local time from the system clock or to set the system clock with local time. To use local time a timezone must be configured in the System Time settings. If no time zone is configured, this unit will behave similar to the *UTC Clock unit*. The value of the *TS* output port (64-bit unsigned integer) represents the number of microseconds that have elapsed since 00:00:00 January 1 1970. (The timestamp format can be changed via the Program settings.) When a rising edge is detected on the boolean *Set* input port, the system clock will be set to the time represented by the timestamp on the *TS* input port (64-bit unsigned integer). When the system clock is set, all elements will continue using the old system time for the rest of the current IPO cycle. The new system time will be used from the next IPO cycle. When the local clock is being switched forward at the beginning of the daylight saving time period, e.g. from *02:00* to *03:00*, it is possible to set illegal times. These times will be converted to times after the switch, e.g. setting *02:01* will result in *03:01*. When the local clock is being switched backward at the end of the daylight saving time period, e.g. from *03:00* to *02:00*, it is possible to set uncertain times, i.e. times that exist as well before and after the switch. These times will be interpreted as times after the switch. The Anybus Edge gateway's firmware includes daylight saving time rules for about 20 years starting from its release date. If you try to set a local time that lies before the release date or more than 20 years past the release date, setting the clock may fail. In this case the boolean *Err* output port will be set to *1* for the duration of one IPO.
In En X Y Clear **LCD**	This unit is deprecated and should be replaced by the corresponding I/O elements. Click here for more info: *LCD, p. 50*.
LCD Keys Up Down Left Right Enter Esc	This unit is deprecated and should be replaced by the corresponding I/O elements. Click here for more info: *LCD, p. 50*.

LCD LEDs Up Down Left Right Enter Esc LCD Red	This unit is deprecated and should be replaced by the corresponding I/O elements. Click here for more info: LCD, p. 50.
Local Clock Timer Int Offset En	This unit represents a clock-synchronous timer. It will periodically put out a rising edge on its boolean output port. The interval in seconds is determined through the value of the *Int* input port (32-bit unsigned integer). The signal is shifted relative to 00:00:00 (local time) by a time in seconds that is specified through the *Offset* input port (32-bit unsigned integer). The timer must be enabled through the *En* input port. For example: if the *Int* input port has a value of 21600 (6 hours) and the *Offset* input port has a value of 7200 (2 hours), the unit will trigger at 2 am, 8 am, 2 pm and 8 pm. The offset should be smaller than the interval. If 24 hours cannot be divided by the interval, the last period will be shorter.
UTC Clock Timer Int Offset En	This unit represents a clock-synchronous timer. It will periodically put out a rising edge on its boolean output port. The interval in seconds is determined through the value of the *Int* input port (32-bit unsigned integer). The signal is shifted relative to 00:00:00 (UTC time) by a time in seconds that is specified through the *Offset* input port (32-bit unsigned integer). The timer must be enabled through the *En* input port. For example: if the *Int* input port has a value of 21600 (6 hours) and the *Offset* input port has a value of 7200 (2 hours), the unit will trigger at 2 am, 8 am, 2 pm and 8 pm. The offset should be smaller than the interval. If 24 hours cannot be divided by the interval, the last period will be shorter.
UTC Clock TS Set TS	This unit can be used to get the current UTC time from the system clock or to set the system clock with UTC time. The value of the *TS* output port (64-bit unsigned integer) represents the number of microseconds that have elapsed since 00:00:00 January 1 1970. (The timestamp format can be changed via the Program settings.) When a rising edge is detected on the boolean *Set* input port, the system clock will be set to the time represented by the timestamp on the *TS* input port (64-bit unsigned integer). When the system clock is set, all elements will continue using the old system time for the rest of the current IPO cycle. The new system time will be used from the next IPO cycle.
Compare In1 In2 Tol LT EQ GT	This unit represents a comparator. It compares the values of the two input ports *In1* and *In2* and assigns values to the three boolean outputs. The output port *LT* will be set to *1* if *In1* is less than *In2*. The output port *EQ* will be set to *1* if *In1* is equal to *In2*. A tolerance value for the equation can be given through the *Tol* input port. This is especially useful for floating point values, where direct comparison may not work due to rounding errors. With a tolerance value of *0.5* the values *1.0* and *1.5* are considered equal. For string comparison the *Tol* input port is ignored and should be left open. The output port *GT* will be set to *1* if *In1* is greater than *In2*. The unit operates in the greatest data type of all output ports connected to the unit's input ports. All input ports of the unit will adopt this data type.
Compare Time Year Month Day WeekDay Hour Minute Second Bitmap	This unit performs a time comparison. It compares the date and time specified through its input ports with the current <u>local</u> time. If both times are equal, the boolean output port will be set to *1*. The *Bitmap* input port (8-bit unsigned integer) can be used to specify which components of the date and time should be compared, if it is set to *0* then all inputs will be ingnored and the output is set to *1*. There are input ports for the following date/time components: • *Year*: Compared if bit *0* of the *Bitmap* input port is set (16-bit unsigned integer) • *Month*: Compared if bit *1* of the *Bitmap* input port is set (8-bit unsigned integer, range [1..12]) • *Day*: Day of the month – compared if bit *2* of the *Bitmap* input port is set (8-bit unsigned integer, range [1..31]) • *WeekDay*: Days since Sunday – compared if bit *3* of the *Bitmap* input port is set (8-bit unsigned integer, range [0..6]) • *Hour*: Compared if bit *4* of the *Bitmap* input port is set (8-bit unsigned integer, range [0..23]) • *Minute*: Compared if bit *5* of the *Bitmap* input port is set (8-bit unsigned integer, range [0..59]) • *Second*: Compared if bit *6* of the *Bitmap* input port is set (8-bit unsigned integer, range [0..59])
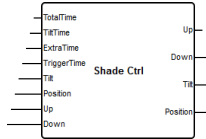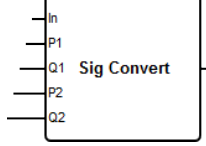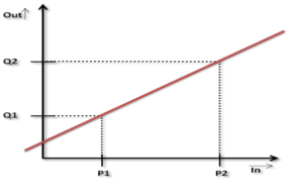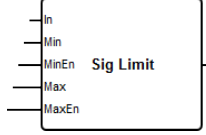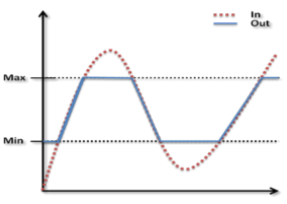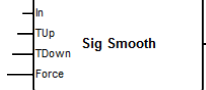
Clk Down Rst Counter RVal IVal Val Over	This unit represents a counter. Every rising edge (transition from *0* to *1*) on the boolean *Clk* input port will increase the numeric value on the *Val* output port (32-bit unsigned integer) by one. If the value on the boolean *Down* input port is *1*, the counter value will be decremented instead. The maximum value that can be put out on the *Val* port is *4,294,967,295* (232-1). The minimum value is *0*. An overflow (from 4,294,967,295 to 0 or vice versa) will be indicated through the boolean *Over* output port. Its value will be *1* until the next clock pulse increments or decrements the counter value again or the counter is reset. If the value on the boolean *Rst* input port is *1*, the counter value will be reset to the current value of the *RVal* input port (32-bit unsigned integer). The *IVal* input port can be used to assign an initial value to the counter.
K Out0 Store Demux In Out1	This unit represents a demultiplexer. It copies the value of the *In* input port to the *Out<K>* output port. The respective output port is selected through the *K* input port (8-bit unsigned integer). *Out0* is selected when *K* is 0. If the value of *K* is too low, *Out0* is selected. If the value of *K* is too high, the output port with the highest index is selected. The value of the boolean *Store* input port determines whether a previously selected output port will keep its value (Store *1*) or will be reset to *0* (Store *0*). The *In* input port and all the output ports will adopt the data type of the output port connected to the *In* input port.
Div	This unit performs a division. It devides the first input port through all subsequent input ports and writes the quotient to its output port. If one of the input ports is *0*, the output port is set to *0*. The division is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.
X N **Extract**	This unit extracts a single bit from the value of its *X* input port (32-bit unsigned integer). The respective bit is selected through the *N* input port (8-bit unsigned integer). The bit is written to the boolean output port. Bit numbering starts at 0.
First Cycle	This unit has a single boolean output port which is set to *1* only during the first cycle after the PLC has been started. This unit is processed during the input stage of the IPO cycle.
In Diff Trigger **Load Profile** Prev	With this unit a load profile can be captured. At a rising edge on the *Trigger* input port the current value of the *Counter* input port as well as the absolute difference between the values of the *Counter* input port and the *Prev* output port will be stored. The difference will be put out on the *Diff* output port. The stored counter value will be put out on the *Prev* output port from the following cycle.
Max	This unit determines the maximum value of all of its input ports and writes this value to its output port. The function is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.
Min	This unit determines the minimum value of all of its input ports and writes this value to its output port. The function is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.
Mul	This unit performs a multiplication. It multiplies all input ports and writes the product to its output port. The multiplication is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.
K In0 **Mux** In1	This unit represents a multiplexer. It copies the value of the *In<K>* input port to the output port. The respective input port is selected through the *K* input port (8-bit unsigned integer). *In0* is selected when *K* is 0. If the value of *K* is too low, *In0* is selected. If the value of *K* is too high, the input port with the highest index is selected. The *In<K>* input ports and the output port will adopt the greatest data type of all output ports connected to one of the *In<K>* input ports.
SetOn SetOff **On/Off Ctrl** Act En	This unit represents an on/off controller. It switches its boolean output port on or off when the value of the *Act* input port crosses the values of the *SetOn* or *SetOff* input ports respectively. The boolean *En* input port can be used to enable the unit. If it is *0*, the output port will always be *0*, too. When *SetOn* = *SetOff*, the unit's behaviour is represented by diagram 1. The output port will be switched on, when *Act SetOn*. The output port will be swiched off, when *Act <= SetOff*.
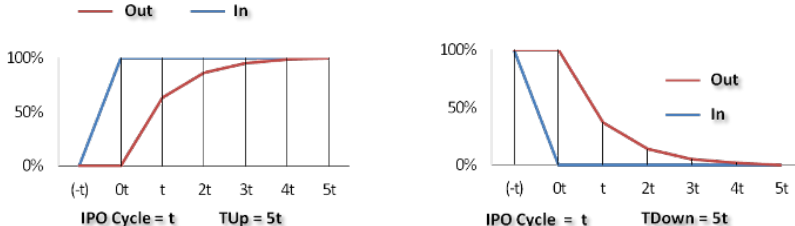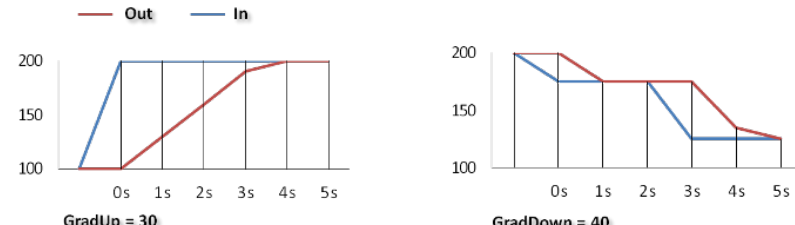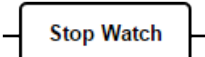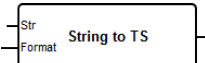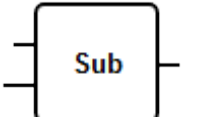
When *SetOn > SetOff*, the unit's behaviour is represented by diagram 2. The output port will be switched on, when *Act = SetOn* (green line). The output port will be switched off, when *Act <= SetOff* (red line).

When *SetOn < SetOff*, the unit's behaviour is represented by diagram 3. The output port will be switched on, when *Act <= SetOn* (red line). The output port will be switched off, when *Act = SetOff* (green line).

Diagram 1.

Diagram 2.

Diagram 3.

The *SetOn*, *SetOff* and *Act* input ports will adopt the greatest numeric data type of all output ports connected to one of them.

This unit can be used to control the OpenVPN client.

The client is started when the boolean *Start* input port changes from *0* to *1*. The client is stopped when the boolean *Stop* input port changes from *0* to *1*.

The boolean *Running* output port indicates whether the client is running. The boolean *Connected* output port indicates whether the client is connected to a server.

This unit performs a binary OR. The OR is performed in the greatest integer data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type. Input ports connected to floating point or string output ports will be treated as if they were connected to a 32-bit signed integer output port.

Examples:

- *0* and *0* ⇒ *0*

- *0* and *1* ⇒ *1*

- *1* and *1* ⇒ *1*

- *3* (0011_2) and *8* (1000_2) ⇒ *11* (1011_2)

This unit packs the values of its input ports into an 8, 16 or 32 bit unsigned integer depending on the number of input ports.

This unit represents an proportional controller. It puts out a percentage value on its *Out* output port (16-bit unsigned integer, range [*0..10,000*]) based on the relation of the value of its *Act* input port to the range of the values of its *SetLow* and *SetHigh* input ports. If the value of the *Act* port is outside of this range, the value of the *Out* port is limited to the range [*0..10,000*] (*0 %* to *100.00 %*), while the value the *Unlimited* output port (32-bit signed integer) can become greater than *10,000* or smaller than *0*. The boolean *En* input port can be used to enable the unit. If it is *0*, the output port will always be *0*, too.

When *SetOn = SetOff*, the unit's behavior is represented by diagram 1. The output port will be set to *0 %* (*0*), when *Act < SetLow*. It will be set to *50 %* (*5,000*), when *Act = SetLow*. It will be set to *100 %* (*10,000*), when *Act SetHigh*. *Unlimited* will have the same value as *Out*.

When *SetHigh SetLow*, the unit's behaviour is represented by diagram 2. *Out* will be *0 %* (*0*) when *Act <= SetLow*. It will rise from *0 %* to *100 %* when the value of *Act* rises between *SetLow* and *SetHigh*. And it will stay at *100 %* (*10,000*) when *Act* becomes >= *SetHigh*. *Unlimited* will go beneath *0 %* when *Act < SetLow* and beyond *100 %* when *Act SetHigh*.

When *SetHigh < SetLow*, the unit's behaviour is represented by diagram 3. *Out* will be *100 %* (*10,000*) when *Act <= SetHigh*. It will fall from *100 %* to *0 %* when the value of *Act* rises between *SetHigh* and *SetLow*. And it will stay at *100 %* (*10,000*) when *Act* becomes >= *SetLow*. *Unlimited* will go beneath *0 %* when *Act < SetHigh* and
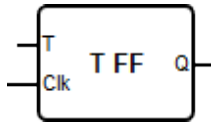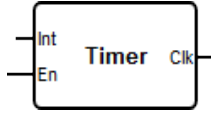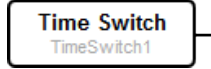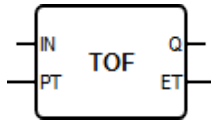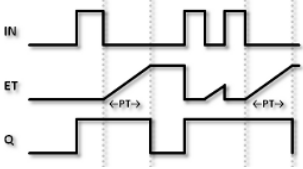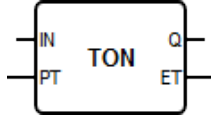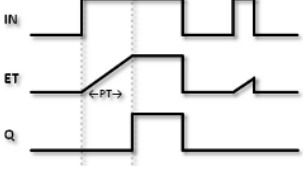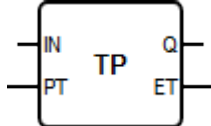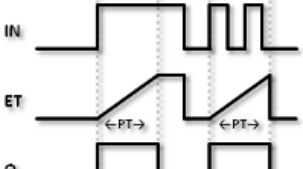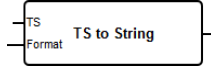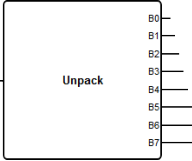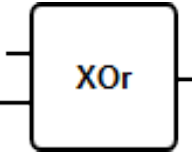
beyond *100 %* when *Act SetLow*.

The *SetLow, SetHigh* and *Act* input ports will adopt the greatest numeric data type of all output ports connected to one of them.

En Ready **Portal Status** Vars Error	This unit provides the following information on the portal communication: The boolean *En* output port will be set to *1* if portal communication is generally enabled. The boolean *Ready* output port will be set to *1* if the device has successfully logged in to the portal and is ready to exchange data. The boolean *Vars* output port will be set to *1* once all portal variables have been initialised. This is especially interesting for portal variables that are stored on the portal retentively. The boolean *Error* output port will be set to *1* if there currently is a communication error. Note that the *Ready* and *Error* output ports will not be updated with every IPO cycle. This unit is processed during the input stage of the IPO cycle.
X N **Put Bit** B	This unit takes the binary value of its *X* input port (32-bit unsigned integer), sets/resets the bit selected through the *N* input port (8-bit unsigned integer) to the state of the boolean *B* input port and writes the resulting value to its output port (32-bit unsigned integer). Bit numbering starts at 0.
Recv Msg 0	This unit can be used to detect when a message has been received. 25 different receive messages can be configured through the messages form including the expected sender and message text. When the respective message has been received the boolean output port of the unit will be set to *1* for the duration of one cycle. Received messages may contain values for variables. These values will be applied when the respective unit is processed, which will happen in the input stage of the IPO cycle.
Set **RS FF** Rst	This unit represents an RS Flipflop. Initially the boolean output port has a value of *0*. When the boolean *Set* input port's value becomes *1*, the output port's value changes to *1*. This value will be kept even if the *Set* input port's value changes back to *0*. When the boolean *Rst* input port's value becomes *1*, the output port's value changes to *0*. This value will be kept even if the *Rst* input port's value changes back to *0*. If both *Set* and *Rst* are activated at the same time, *Rst* has priority.
In **Max Acc** Act Rst Prev	This unit compares the current values of its *In* input port and its *Act* output port and puts out the greater value on the *Act* output port. On a rising edge on the *Rst* input port the current maximum value will be put out on the *Prev* output port and the *Act* output port is reset to the minimum value of the *In* input port's datatype. The reset is performed before the current value of the *In* input port is evaluated.
In **Min Acc** Act Rst Prev	This unit compares the current values of its *In* input port and its *Act* output port and puts out the smaller value on the *Act* output port. On a rising edge on the *Rst* input port the current minimum value will be put out on the *Prev* output port and the *Act* output port is reset to the maximum value of the *In* input port's datatype. The reset is performed before the current value of the *In* input port is evaluated.
OK Snd **Send Msg 0** Err	This unit can be used to send messages. There are 25 different message sending units available. The recipients and texts of the messages that are to be sent can be configured through the messages form. The respective message will be sent when there is a rising edge on the boolean *Snd* port and there is no previous send request pending. The boolean *OK* port will be set to *1* for one cycle when the message has been successfully sent. If sending the message failed, the boolean *Err* port will be set to *1* for the duration of one cycle.

This unit can be used to control window shades or shutters. Its four first input ports are used to define several time constants. The *TotalTime* input port (32-bit unsigned integer) defines the time (ms) needed by the shade to move from its lowest to its highest position or vice versa.

The *TiltTime* input port (32-bit unsigned integer) defines the time (ms) needed by the shade to tilt from the point it starts ascending to the point it starts descending. This time can be set to zero for simple shutters.

The value of the *ExtraTime* input port (32-bit unsigned integer, ms) is used when the shade should move to its lowest or highest position as a safeguard to ensure the shade closes or opens fully.

When the *Up* or *Down* input port is active for the amount of time defined by the value of the *TriggerTime* input port (32-bit unsigned integer, ms), the shade will be moved to its highest or lowest position respectively. Note that the shade controller cannot run the motor of the window shade for periods shorter than the cycle time. So all times should optimally be multiples of the cycle time.

The *Tilt* input port (8-bit signed integer, ms) is used to command the Shade Controller to tilt the shade to a certain position. Values from *-100* too *100* are accepted. *-100* represents the extreme tilt position where the fins are lower on the inside. *0* represents the horizontally aligned tilt position. And *100* represents the extreme tilt position where the fins are lower on the outside. The *Position* input port (8-bit unsigned integer, ms) is used to command the Shade Controller to move the shade to a certain vertical position. Values from *0* to *100* are accepted. *0* represents the shade's highest position. *100* represents the shade's lowest position. The unit will first try to achieve the target position and then the target tilt position.

The boolean *Up* and *Down* input ports can be used to manually move the shade. The manual input ports have priority over the automatic ones. So if either the *Up* or *Down* input port is activated, automatic operation will be paused until either the *Tilt* or *Position* input port change their value.

The boolean *Up* and *Down* output ports are used to control the shade's motor.

The *Tilt* and *Position* output ports (8-bit signed number) show the current position and tilt of the shade. This unit has an initialisation sequence which will move the shade to its lowest position when the diagram is started in order to reach a well-known position.

The *Down* output port will be set for (*TotalTime* + *TiltTime* + *ExtraTime*) ms. All input is ignored during this time.

This unit represents a 32-bit shift register.

With every rising edge on the boolean *Clk* input port the unit will shift the contents of its register left or right according to the value of the boolean *Right* input port (*1* ⇒ shift right). The value of the boolean *In* input port will be shifted into the register. The resulting register content will be written to the output port (32-bit unsigned integer).

This unit transforms the value of the *In* input port according to a linear function and writes the result to the output port. The linear function is defined through the two points *(P1, Q1)* and *(P2, Q2).*

The transformation is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.

This unit can be used to limit the value of the *In* input port. If *In* is smaller than *Min* and the minimum limit is activated through the boolean *MinEn* input port, the output is set to *Min*. If *In* is greater than *Max* and the maximum limit is activated through the boolean *MaxEn* input port, the output is set to *Max*.

The limitation is performed in the greatest numeric data type of all output ports connected to the unit's *In*, *Min* and *Max* input ports. These input ports will - as well as the output port - adopt this data type.

This unit is used to smooth a signal. When the value of the *In* input port changes, the value of the output port will be approximated to that new value using an exponential function.

The value of the *TUp* input port (32-bit unsigned integer) describes the time (ms) needed to reach the new value, if the old value is less than the new one.

The value of the *TDown* input port (32-bit unsigned integer) describes the time (ms) needed to reach the new value, if the old value is greater than the new one.

If the boolean *Force* input port is set, the value of the *In* input port will be immediately copied to the output port. The smoothing is performed in the numeric data type of the output port connected to the *In* input port. The *In* input port and the output port will adopt this data type.

This unit is used to limit the gradient of a signal. When the value of the *In* input port changes, the value of the output port will be tracked to that new value using a linear function.

The value of the *GradUp* input port defines the maximum gradient per second of this linear function, if the new value is greater than the old one. The value of the *GradDown* input port defines the maximum gradient per second of this linear function, if the new value is less than the old one. Both gradients must be positive numbers.

If the boolean *Force* input port is set, the value of the *In* input port will be immediately copied to the output port. The boolean *En* input port can be used to enable the unit. If it is *0*, the output port will always be *0*, too. The tracking is performed in the greatest numeric data type of all output ports connected to either the *In*, *GradUp* or *GradDown* input port. These input ports and the output port will adopt this data type.

This unit represents a State Machine's state. There are eight different State Machine units (*State Machine 0* to *State Machine 7*) available allowing you to implement eight independent state machines.

A state machine is implemented by placing various State Machine units with the same State Machine index (e.g. State Machine 0). The value of the *Sel* input port (32-bit signed integer) of each unit determines which state is handled by it. When a State Machine unit becomes active (State machine's state = *Sel*), it checks its boolean *Cond<X>* input ports. If one of them is *1*, the state machine changes to the state determined by the corresponding *Next<X>* input port (32-bit signed integer) after the current IPO cycle. The *State* output port (32-bit signed integer) shows the current state of the State Machine. The boolean *Active* output port indicates that the corresponding unit is currently active (State machine's state = *Sel*). The initial state is *0*.

The stop watch unit measures the time in milliseconds between two rising edges on is input port and puts out this time on its output port.

This unit converts a string that contains a timestamp into a numerical timestamp (number of microseconds since 00:00:00 January 1 1970; the timestamp format can be changed via the Program settings.). The string timestamp must be given at the *Str* input port. The numerical timestamp will be put out at the output port (64-Bit unsigned integer). At the *Format* input port another string is expected that defines the format of the string timestamp. Note that leading zeroes will be ignored and that spaces, tabulators and line breaks will be treated the same way. White space characters also may appear any number of times. If the *Format* input port is open, the default format string "%Y-%m-%dT%H:%M:%SZ" (according to ISO 8601) is used.
If the conversion fails, the unit puts out *0*.

This unit performs an subtraction. It subtracts the values of its second and following input ports from its first input port and writes the difference to its output port.
The subtraction is performed in the greatest numeric data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type.

T FF — T, Clk, Q	This unit represents a T Flipflop. Initially the boolean output port has a value of *0*. If the boolean *T* input port is *1*, the boolean output port will change its state (toggle) whenever there is a rising edge on the boolean *Clk* input port. If the *T* input port is *0*, the output port holds its previous value
Timer — Int, En, Clk	This unit represents a timer in milliseconds. It will output a periodic clock signal on the boolean *Clk* output port. The period of the signal is determined through the value of the *Int* (interval) input port (32-bit unsigned integer, ms) and must be a multiple of (at least twice) the PLC's cycle time. If the value of the *Int* input port is smaller than the cycle time or not divisible by the cycle time, the unit will internally round it up to the next multiple of the cycle time. The clock signal is only generated while the boolean *En* input port is *1*.
Time Switch TimeSwitch1	This unit represents a Time Switch. It switches its boolean output port based on a plan that can be configured through the Time Switches form. There are eight time switch units, which can be configured independently. This unit is processed during the input stage of the IPO cycle.
TOF — IN, PT, Q, ET	This unit represents a timer off-delay. When the boolean *IN* input port switches to *1*, the boolean *Q* output port will be set to *1*. When the *IN* input port switches to *0*, the timer will be started. The *ET* (elapsed time) output port (32-bit unsigned integer) will indicate the number of milliseconds elapsed since the timer has been started. When the value of the *ET* output port reaches the value of the *PT* (preset time) input port (32-bit unsigned integer), the timer will be stopped and the *Q* output will be reset. Note that if the value of the *PT* input port is smaller than or not divisible by the PLC's cycle time, it will be internally rounded up to the next multiple of the cycle time.
TON — IN, PT, Q, ET	This unit represents a timer on-delay. When the boolean *IN* input port switches to *1*, the timer will be started. The *ET* (elapsed time) output port (32-bit unsigned integer) will indicate the number of milliseconds elapsed since the timer has been started. When the value of the *ET* output port reaches the value of the *PT* (preset time) input port (32-bit unsigned integer), the timer will be stopped and the boolean *Q* output port will be set. The *Q* output port will be reset when the *IN* input port switches to *0*. Note that if the value of the *PT* input port is smaller than or not divisible by the PLC's cycle time, it will be internally rounded up to the next multiple of the cycle time.
TP — IN, PT, Q, ET	This unit represents a timer pulse. When the boolean *IN* input port switches to *1*, the boolean *Q* output port will be set to *1* and the timer will be started. The *ET* (elapsed time) output port (32-bit unsigned integer) will indicate the number of milliseconds elapsed since the timer has been started. When the value of the *ET* output port reaches the value of the *PT* (preset time) input port (32-bit unsigned integer), the timer will be stopped and the *Q* output port will be set to *0*. Note that if the value of the *PT* input port is smaller than or not divisible by the PLC's cycle time, it will be internally rounded up to the next multiple of the cycle time.
Trigger	This unit represents a trigger. It will set its boolean output port to *1* for one IPO cycle when it detects a rising edge on one of its boolean input ports. To detect a falling edge, simply negate the input port. To detect a rising or falling edge, connect the same input signal to two of the trigger's input ports and negate one of them.
TS to String — TS, Format	This unit converts a numerical timestamp (number of microseconds since 00:00:00 January 1 1970; the timestamp format can be changed via the Program settings.) to a string representation. The numerical timestamp is expected at the *TS* input port (64-bit unsigned

integer). The string timestamp is put out at the output port. At the *Format* input port another string is expected that defines the format of the string timestamp. If the *Format* input port is open, the default format string "%Y-%m-%dT%H:%M:%SZ" (according to ISO 8601) is used. If the conversion fails, the unit puts out *"Error"*.

Format string

The format string consists of normal characters and special format specifiers, which represent date and time components. They start with a percentage sign. The following format specifiers are available:

- %C : Century with leading zero (00-99)

- %d : Day of month with leading zero (01-31)

- %D : Milliseconds with leading zero (000-999)

- %e : Day of month with leading space (1-31)

- %F : Microseconds with leading zero (000-999)

- %H : Hour in 24h format with leading zero (00-23)

- %I : Hour in 12h format with leading zero (01-12)

- %j : Day of year with leading zeroes (001-366)

- %m : Month with leading zero (01-12)

- %M : Minute with leading zero (00-59)

- %n : Line break

- %p : "AM" or "PM"

- %S : Seconds with leading zero (00-61)

- %u : ISO 8601 week day (1-7), Monday as 1

- %t : Tabulator

- %U : Week number (00-53), the first Sunday is the first day of week 1

- %V : ISO 8601 week number (01-53)

- %w : Week day (0-6), Sunday as 0

- %W : Week number (00-53), the first Monday is the first day of week 1

- %y : The last two digits of the year with leading zero (00-99)

- %Y : Year

- %% : Percentage sign

Unpack (B0–B7)	This unit unpacks the bits of the value of its input port into single bits and writes them to its boolean output ports. Input values with a 8, 16 or 32 bit length are supported.
XOr	This unit performs a binary XOR. The XOR is performed in the greatest <u>integer</u> data type of all output ports connected to the unit's input ports. All ports of the unit will adopt this data type. Input ports connected to floating point or string output ports will be treated as if they were connected to a 32-bit signed integer output port. Examples: • *0* and *0* $\Rightarrow$ *0* • *0* and *1* $\Rightarrow$ *1* • *1* and *1* $\Rightarrow$ *0* • *15* (1111_2) and *8* (1000_2) $\Rightarrow$ *7* (0111_2)

3.2.2   I/Os

There is one element for each input and output. The number and structure of the I/Os depend on the particular Anybus Edge device and its configuration.
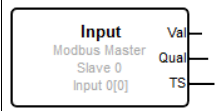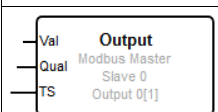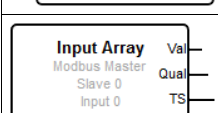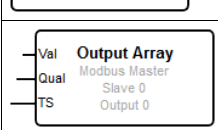
Each I/O element has three ports that can be connected to other elements as described above.

The *Val* port represents the value of the I/O and has the data type of the I/O.

The *Qual* port (8-bit unsigned integer) represents the quality. A value of 0 indicates good quality. Any other value indicates bad quality. Possible values depend on the particular interface. Please note that not all I/O interfaces support the transmission of qualities. In this case, for inputs, the quality will be set to a value unequal to 0, if the input could not be read for some reason; for outputs the quality is meaningless.

The *TS* port (64-bit unsigned integer) represents the timestamp (number of microseconds elapsed since 00:00:00 January 1 1970 UTC; the timestamp format can be changed via the Program settings) of when the value was last updated. Please note that not all I/O interfaces support the transmission of timestamps. In this case, for inputs, the timestamp will be set to the time of the last update inside the driver; for outputs the timestamp is meaningless.

The interface, device and value names will be shown in grey below the type of the element.
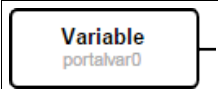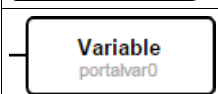
Input Val Modbus Master Qual Slave 0 Input 0[0] TS	This element represents an input. The value of the input is read from the physical inputs during the input stage of the IPO cycle.
Val **Output** Qual Modbus Master Slave 0 TS Output 0[1]	This element represents an output. The value of the output is written to the physical outputs during the output stage of the IPO cycle.
Input Array Val Modbus Master Qual Slave 0 Input 0 TS	This element represents an array of inputs. In contrast to the plain *Input* element it cannot be connected to any other elements except for an *Output Array* element. The I/O arrays can not be connected to each other, if they do not have the same length. The values of the inputs are read from the physical inputs during the input stage of the IPO cycle.
Val **Output Array** Qual Modbus Master Slave 0 TS Output 0	This element represents an array of outputs. In contrast to a plain *Output* element it cannot be connected to any other elements except for an *Input Array* element.The I/O arrays cannot be connected to each other if they do not have the same length. The values of the outputs are written from the physical outputs during the output stage of the IPO cycle.

Internally the I/Os are addressed via an interface, a device, a value and optionally an array index. For example, if the Anybus Edge features some digital I/Os and a Modbus Master interface, the interface index will select one of these two interfaces. The device index might for example select the Modbus slave and finally the value index would select the respective Modbus register. The array index will be used additionally, if the corresponding value addresses an element of an array.

When you hover over an I/O element, a tooltip will show the indexes, the data type and array size.

3.2.3   Variables

The *Variable* element represents variables that are reflected on the portal, but are not linked to any physical I/O. A *Variable* element corresponds to a variable with the same name and can be used either as an input or an output. Variables can be configured through the Variables form.

Variable portalvar0	This element represents the value of the variable. It is processed during the process stage of the IPO cycle
Variable portalvar0	This element represents a variable as an output. It will write its value to the variable. It is processed during the process stage of the IPO cycle

3.2.4   Constants

When you choose to add a constant via the toolbar, a dialog will show up where you can enter a fixed value. This value can be a decimal integer, floating point or string value. Integer values are

limited to the range [-2^{31}, 2^{32}-1]. Floating point values are limited to a precision of 15 digits. Strings are limited to 4096 bytes. The data type of a constant is the smallest data type that can represent the constant's value. Here are some examples:

- 0 ⇒ *Boolean (false)*

- 1 ⇒ *Boolean (true)*

- 120 ⇒ *8-bit unsigned integer*

- -5 ⇒ *8-bit signed integer*

- 500 ⇒ *16-bit unsigned integer*

- -200 ⇒ *16-bit signed integer*

- 70000 ⇒ *32-bit unsigned integer*

- -35000 ⇒ *32-bit signed integer*

- 1.5 ⇒ *32-bit floating point number*

- 3.14159265359 ⇒ *64-bit floating point number*

- 5m ⇒ *String*

Constants are processed in the input stage of the *IPO Cycle*.

The constant's value can be changed. To do so, right-click on the element, or double-tap the element if touch functionality is supported. A pop-up menu will appear (See image below). Select the entry *Change Value*. Alternatively you can double-click the element.
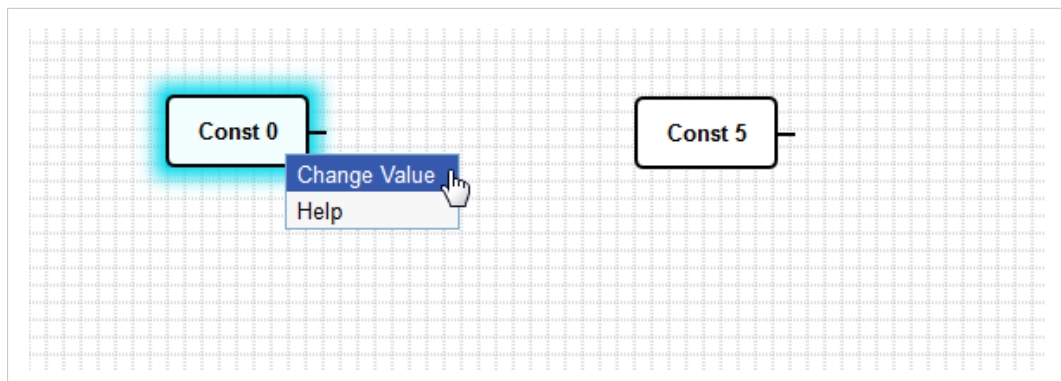
Fig. 9   Changing a constant's value

3.2.5   Data Type Conversion Units

Data type conversion units can be used to convert any value to a specific data type. Supported data types:

- Boolean

- 8-bit unsigned integer

- 8-bit signed integer

- 16-bit unsigned integer

- 16-bit signed integer

- 32-bit unsigned integer

- 32-bit signed integer

- 32-bit floating point number

- 64-bit floating point number

- String

3.3 I/O Mapping

Besides the diagram the HMS Sequence editor allows to directly map inputs to outputs using the I/O mappings. This can be done in a quick and easy manner using a mapping table. The table show inputs in the left column and outputs in the right column. Variables can also be used as inputs ot outputs respectively.

I/Os with different data types can be mapped to each other. The type conversion will be performed automatically. It is also possible to map arrays with different lengths. The PLC will copy as many elements as the shorter of the arrays has.

> **!** If you create an I/O mapping for an output (or a variable) and also write the output (or the variable) in the diagram, the value from the diagram will prevail.

3.3.1 Adding a Mapping

To add new mappings use the Add button from the toolbar. A dialog will be opened through which you can select the input and output you want to map. The drop-down lists contain all the inputs and outputs. The names of the I/Os are composed of the interface name (if any), the device name (if any) and the value name, separated by periods. Arrays have square brackets appended to their name. Array elements have their index in square brackets appended. Variables are prefixed with the text *Variables*.

To select the desired input or output as fast as possible, type the first few letters of the name. The list will immediately be filtered according to the entered text. Additionally the input text will be auto-completed as far as the names of all matching I/Os are equal. The completed part of the text is selected. You can remove the selection to accept the completed text and enter another part of the name at the end to further confine the selection. The fastest way to do so is by using the right arrow key of your keyboard.

Example

Assume there are several Modbus inputs available (as well client/master as server/slave). When you input the text "Mod", the list will only show Modbus values. Furthermore the text will be auto-completed to "Modbus ".
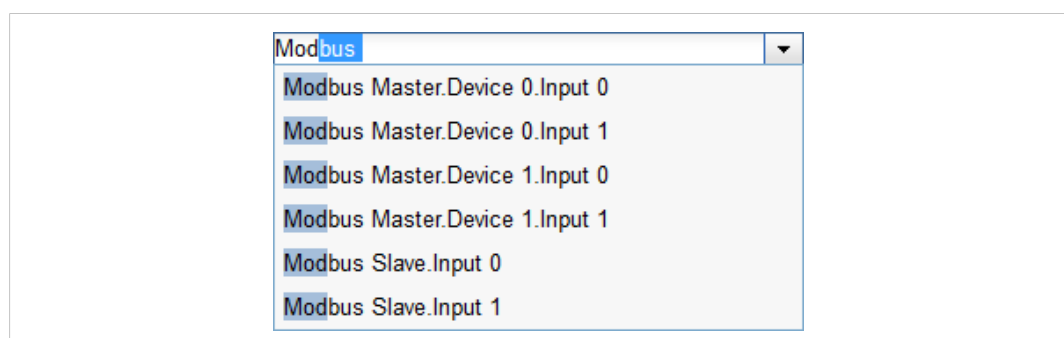
Fig. 10   Selection of an input – step 1

When you accept the completed text and supplement it to "Modbus M", the list will only show Modbus client/master values and the text will be auto-completed to "Modbus Master.Device ".
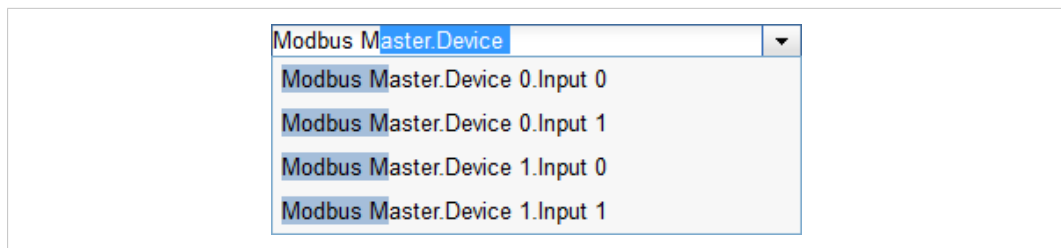
Fig. 11 Selection of an input – step 2

When you accept the completed text and supplement it to "Modbus Master.Device 1", the list will only show values of the second device and the text will be auto-completed to "Modbus Master.Device 1.Input ".
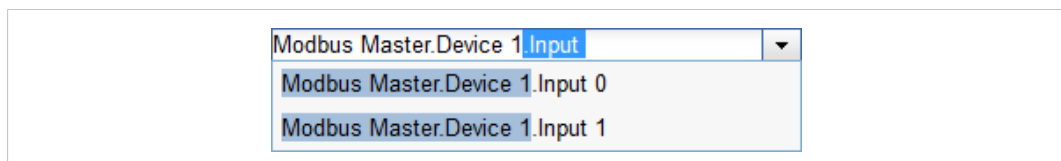
Fig. 12 Selection of an input – step 3

Now you can select one of the input values.

3.3.2 Edit, Delete Discard Mappings

Use the corresponding buttons in the toolbar to edit or delete a mapping or to discard the whole mapping table.

3.3.3 Limits

A maximum of 9216 I/O mappings can be added. That is enough to map all outputs and all variables.

3.4 Program Settings

By clicking the Program settings button in the toolbar, a dialog can be opened through which you can change the following program settings:

Diagram description	Here you can enter a description for the diagram. The description can consist of up to 1024 printable ASCII characters and line breaks.
Cycle time	Here you can configure the cycle time with which the PLC will run while executing the program. The cycle time can be configured between 10 and 1000 ms. Greater values allow for more complex programs, i.e. more diagram elements and I/O mappings. If the cycle time can not be maintained, a message will be emitted to the *Event Log*. In this case you may either increase the cycle time or reduce the complexity of the program.
Treat all numerical I/O values on the diagram as 32-bit signed integers	Before version 8.2.0 all I/O values were treated as 32-bit signed integer numbers by the diagram. For backward compatibility this checkbox can be used to mimic this behavior for all numerical data types. For programs created with older versions this checkbox is checked automatically, but it can be unchecked.
Use timestamps with seconds resolution instead of microseconds resolution (32-bit instead of 64-bit)	Before version 19.1.0 timestamps in UNIX timestamp format (32-bit, seconds since 1970-01-01 00:00:00) were used. For backward compatibility this checkbox can be used to mimic this behavior. This effects the Local Clock, UTC Clock, String to TS and TS to String unit as well as the I/O elements. For programs created with older versions this checkbox is checked automatically, but it can be unchecked.

3.5   Running the Program

Press the start button on the toolbar to start execution of the diagram that has been saved to the PLC. While the PLC is running, the start button will change to a stop button via which you can halt the execution.

MER HÄR!

3.5.1   IPO Cycle

The Anybus Edge run-time system executes an IPO cycle (Input-Process-Output) in each cycle. This cycle includes the following steps:

1. **Input** – During this step the following inputs will be read:
    –   Local inputs
    –   Portal variables

2. **Process** – During this step the PLC program is being processed. It consists of:
    a.   I/O mappings
    b.   Diagram – The diagram executes its own IPO cycle:
        i.   Input
            –   Input elements
            –   Variable input elements
            –   Constants
            –   Special units (see Diagram)
        ii.   Process
            –   Other units (see Diagram)
        iii. Output
            –   Output elements
            –   Variable output elements

3. **Output** – During this step the following outputs will be written:
    –   Local outputs
    –   Portal variables

The cycle time can be configured through the Program settings dialog. Note that events which persist for a shorter period of time might not be recognized.

3.5.2   Running State at Start-Up

The PLC will store its current running state in non-volatile memory. So if the PLC is running and you turn the power off, it will start running after next start-up.

3.5.3   Event Log Messages

The following section describes the messages that the PLC may log to the *Event Log*. All Messages are prefixed with the text *PLC Driver:*.

The PLC has been started.

The PLC has been successfully started.

The PLC has been stopped.

The PLC has been successfully stopped.

A new I/O mapping list has been saved.

A new I/O mapping list has been successfully saved to the PLC.

A new diagram has been saved.

A new diagram has been successfully saved to the PLC.

Task cycle is being maintained again.

This message appears after this message, when the task cycle is being maintained again.

Portal communication error *<x>*

A communication error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

If you see an error code that is not on the above list, please contact the customer support.

Task cycle has been violated.

The given task cycle of the PLC has been violated at least once. This is probably caused by a cycle time that is too short in relation to the complexity of the program. Please check the duty cycle of the PLC in the status bar of the editor. When the task cycle can be maintained again, this message will appear.

Stopping PLC due to missing licence.

Since the device does not have a valid HMS Hub licence, the PLC runs in demo mode and is now being stopped after a runtime of four hours.

Exception in task: *<message>*

A fatal exception has occurred inside the PLC task. A detailed description is provided. The PLC is stopped and the device will be restarted by the watchdog after a short time. This should not happen. Please contact the customer support.

3.6   Live View

Via the Live View function it is possible to see the current state of the PLC right in the diagram view. After clicking on the Live View button on the toolbar the HMS Sequence editor will start showing the actual value of each output port in a small blue box next to the port.
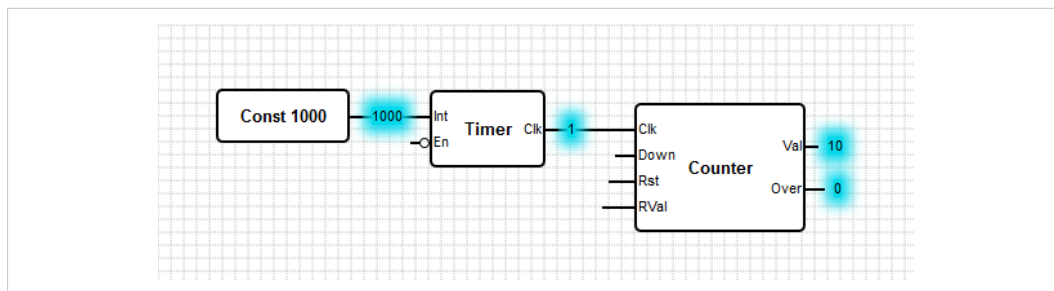
Fig. 13   Live view

The values will be updated as fast as possible. Note that changing the diagram while Live View is enabled will lead to errors, because the values from the PLC won't match the diagram anymore. In this case, the Live View will be automatically stopped.

To stop Live View manually, just click the same toolbar button.

3.7   Status Bar

Fig. 14   The status bar of the HMS Sequence editor

The image shows the status bar of the HMS Sequence editor. It shows information on the current state of the PLC.

In the following the information that may be shown will be described in detail.

	This icon will be displayed if the connection to the device is lost.
	The stoplight displays the current status of the PLC: • Red — No program loaded, PLC not running • Amber — Program loaded, PLC not running • Green — Program loaded, PLC running
Duty cycle	Behind the stoplight the duty cycle of the PLC is displayed – as well the current as the maximum duty cycle since the start of the PLC are shown. The duty cycle is indicated in percent. A duty cycle of 50 % means that the PLC uses 50 % of the available CPU processing time. **!** You must leave spare CPU processing time for other processes in the system. The duty cycle should be kept below 75 %. When your diagram becomes more complex you may increase the PLC's cycle time through the Program Settings. When the duty cycle reaches (nearly) 100 % you will not be able to control the Anybus Edge device.

4 Settings

On the settings page you can configure your Anybus Edge device.
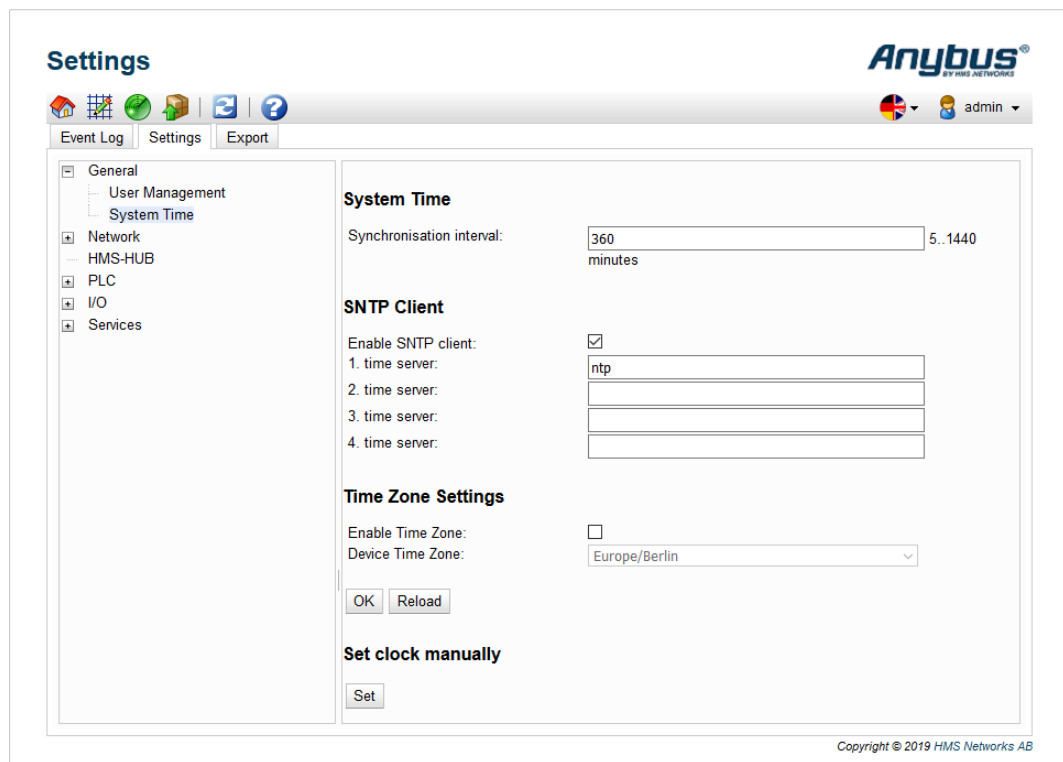
Fig. 15   Screenshot of the settings page

At the top of the page there is a toolbar. In the following, the toolbar's buttons will be described in detail.

	Closes the settings page and opens the *Home Page*.
	Closes the settings page and opens the *HMS Sequence Editor*.
	Closes the settings page and opens the *Diagnostics* page.
	Closes the home page and opens the *Update* page.
	Reboots the device to apply changes.
	Opens the help system in a new window.
	Login Button

Below the toolbar you'll find a tab control with three tabs.

Event Log	This tab shows the *Event Log*.
Settings	This tab provides settings for the various functions of your Anybus Edge device.
Export	Through this tab you can *Export* the configuration of your Anybus Edge device.

Functions

Due to the many features and communication protocols available on the device, significant response time latencies (especially when communicating through the cellular modem) and CPU overload might be experienced during extensive usage.

4.1 Event Log

The event log is the central place where all components of the Anybus Edge firmware store their event messages. Each event message consists of the message text, a timestamp and an event type.

Click on a message's text in order to open the help system and show more detailed information on the respective message if available.

Timestamps are stored in the device in UTC. When displayed they are converted to the time zone of the system running the web browser. Note that the system clock of the Anybus Edge device may not be synchronised. In this case the timestamps will not display correctly.

The event types are represented by small icons:

	Information: Provides information about normal operation
	Warning: Informs the user about unexpected but unproblematic events.
	Error: Informs the user about a failure in one of the firmware's components.
	Fatal error: Typically prevents device operation and should not happen.
	Debug message: Is used during firmware development and should not appear in production firmware.

Events are stored in a circular buffer in RAM. That means that the event log will be emptied when the device is restarted and that new events may overwrite older events once the circular buffer is completely filled.

4.1.1 Export

Through the corresponding link you can download the events stored in the event log as a CSV file.

4.1.2 Messages Index

The following is an index of all messages that any of the PLC's components may log.

- Running The Program
    - The PLC has been started.
    - The PLC has been stopped.
    - A new I/O mapping list has been saved.
    - A new diagram has been saved.
    - Task cycle is being maintained again.
    - Portal communication error *<x>*
    - Task cycle has been violated.
    - Stopping PLC due to missing licence
    - Exception in task: *<message>*
- System Time
    - Running
    - System clock updated via SNTP
    - System clock updated from RTC
    - Could not get exclusive access to clock(s).
    - Daylight saving time rules will be used up within the next two years.
    - SNTP client error: *<message>*
    - RTC not found
    - Could not write to RTC: *<message>*
    - Could not read from RTC: *<message>*
    - Could not read daylight saving time file.
    - No daylight saving time rule found for the current time.
    - DNS error (*<error code>*) for server *<server name/IP>*
    - Could not open socket for server *<server name/IP>*
    - Could not send to server *<server name/IP>*
    - Failed to receive reply from server *<server name/IP>*
    - Unexpected packet format from server *<server name/IP>*
    - Timestamps not plausible from server *<server name/IP>*

- Cellular Modem

    – Stopped

    – Started

    – Running

    – Deactivated

    – Activated

    – Powered modem

    – Modem initialised (RSSI: *<x>* dBm)

    – SIM card ID: *<x>*

    – Registered to network

    – Roaming

    – Data connection established

    – Opening fallback connection

    – Closing fallback connection

    – GPS module initialised

    – SIM card not present

    – Invalid PIN - [not] retrying

    – SIM card locked, PUK required - [not] retrying

    – SIM card failure

    – Modem initialisation failed. (code *<x>*)

    – Network registration timed out ((not) searching)

    – Roaming disallowed

    – Network registration failed (code *<x>*, network status *<y>*)

    – Data connection failed (code *<x>*)

    – Network lost (code *<x>*, network status *<y>*)

    – Data connection broken ([DCD pin: *<x>*,]PPP status: *<y>*, Data connection: *<z>*)

    – Connection supervision failed

    – Modem reset

    – Hanging! (state: *<x>*)

    – Could not initialise GPS module

- WLAN
  - Started
  - Activated
  - Running
  - IP address obtained successfully through DHCP server
  - Stopping the driver
  - Stopped
  - <status>
  - Hardware initialization error *<error code>*
- Bluetooth
  - Started
  - The driver is running
  - Pairing button pressed, device pairable
  - Pairing button pressed, device unpairable
  - Pairing timeout, device unpairable
  - Pairing button long press, all stored remote devices removed
  - Stopping the driver
  - Stopped
- Network Address Translation
  - Running
  - *<x>* static mapping(s) read from configuration
  - No free mapping entry available for incoming connection from internal interface
  - No free firewall rule entry available for outgoing connection

- OpenVPN
    - Driver has been started.
    - Driver has been stopped.
    - Driver stopped.
    - Connection established.
    - Connection closed.
    - Authority's certificate uploaded.
    - Certificate uploaded.
    - Private key uploaded.
    - User and password file uploaded.
    - Diffie-Hellman file uploaded.
    - Could not start driver.
    - Could not stop driver.
    - Driver stopped unexpectedly.
    - Upload of authority's certificate failed.
    - Upload of certificate failed.
    - Upload of private key failed.
    - Upload of user and password file failed.
    - Upload of Diffie-Hellman file failed.

- DHCP Server
    - Running
    - Added static lease IP=*<x>*, MAC=*<y>*
    - Discover message received, CI=*<x>*, MAC=*<y>*
    - Request message received, CI=*<x>*, MAC=*<y>*
    - Leased IP=*<x>*, Leasing Time=*<y>*, Index=*<z>*
    - Release message received, CI=*<x>*, MAC=*<y>*
    - Release IP=*<x>*, Index=*<y>*
    - Network interface not configured
    - Failed to send response message, error *<x>*
    - No more client addresses available
    - Receive Error *<x>*
    - Received malformed message
    - No interface found for given IP address
    - Open socket failed
    - Bind socket failed
    - Only *<x>* clients possible due to netmask setting
    - Failed to allocate client data structure
    - Adding static lease IP=*<x>* failed, IP address is in use
    - Adding static lease IP=*<x>* failed, no free slot
    - Adding static lease IP=*<x>* failed, wrong IP address

- Portal

  - Opening connection to <*host*

  - Connection established

  - Connection closed

  - Clock synchronised

  - Logging in

  - Synchronising clock

  - Initiating close sequence

  - Connection closed by Broker

  - Connection closed by Broker - Broker is going down

  - Connection closed by Broker - Protocol error

  - Connection closed by Broker - Keep-alive timeout

  - Connection closed by Broker - Another client logged in with same user name or to same scope

  - Connection closed by Broker - User was still logged in - Retry

  - Logged in

  - Need to synchronise clock, but user has not set a clock update call-back function.

  - Close sequence timed out

  - Ack with sequence number <*sequence number*could not be matched

  - Nak with sequence number <*sequence number*could not be matched

  - Point <*path*not assigned to dataset: Point is inactive

  - Point <*path*not assigned to dataset: Data type does not match

  - Commit command could not be matched

  - Transaction ID <*ID*from cancel command could not be matched

  - RPC response does not match a request

  - RPC method <*method*not found

  - Could not enqueue value for point <*path*: Send queue is full

  - HTTP upgrade error (<*error code*)

  - Invalid character in challenge (<*hexadecimal character code*)

  - Empty challenge

  - Mandatory header field missing

  - Could not connect: Socket error (<*error code*)

  - Could not connect: Connect error (<*error code*)

  - Could not connect: TLS error (<*error code*)

  - Could not connect: Could not resolve host name.

  - Could not connect: Could not send HTTP request.

  - Could not connect: Could not receive HTTP response.

  - Could not connect: Protocol error

- – Could not connect: Could not log in to proxy server.

- – Emergency command timout

- – Could not send emergency command: Send error

- – Keep alive timeout

- – Invalid sequence number <*sequence number*

- – Invalid opcode <*hexadecimal opcode*

- – Invalid WebSocket opcode <*hexadecimal opcode*

- – Closing connection due to communication error (<*error code*)

- – Could not log in: Could not send Login command

- – Could not log in: Timeout

- – Could not synchronise clock: Send error

- – Could not synchronise clock: Timeout

- – Could not get publish hash: Send error

- – Could not get publish hash: Timeout

- – Could not enable publishing of all points: Send error

- – Could not enable publishing of all points: Timeout

- – Could not send write command

- – Write/commit command timeout for point <*path*

- – Could not add value for point <*path*to write command (<*error code*)

- – Could not send commit command for write command with sequence number <*sequence number*(<*error code*)

- – Could not send create node command

- – Could not send modify node command

- – Could not send delete node command

- – Could not send get node properties command

- – Could not send enable publish command

- – Waiting for publish to complete timed out

- – Node manipulation command timed out

- – Could not send keep-alive pong (<*error code*)

- – Mandatory RPC field missing

- – RPC server not initialised

- – Could not parse RPC: Out of memory

- – Could not parse RPC: <*error message*

- – Could not process RPC: Queue is full

- – Could not process RPC: Receive error

- – Could not process RPC: Unexpected error <*error code*

- – Ack with sequence number <*sequence number*contained additional payload data

- – Could not log in: Access denied

- – Could not log in: Unexpected error (<*error code*)

- – Could not get publish hash: Unexpected error (<*error code*)

- – Could not enable publishing of all active points: Unexpected error (<*error code*)

- – Node manipulation command failed (<*error code*)

- – Write command with sequence number <*sequence number*failed (<*error code,* <*point index*)

- – Write command with sequence number <*sequence number*failed (<*error code*)

- – Nak with sequence number <*sequence number*contained invalid payload data (<*error code*)

- – Could not synchronise clock: Receive error (<*error code*)

- – Could not synchronise clock: Get time response contained invalid payload data

- – Could not synchronise clock: Unexpected error (<*error code*)

- – Could not get publish hash: Receive error

- – Could not get publish hash: Out of memory

- – Could not get publish hash: Unexpected error (<*error code*)

- – Could not publish group: Invalid index <*group index*

- – Could not publish group: Re-publish <*known path*with path <*new path*

- – Could not publish group: Initial publish without path

- – Could not publish point: Invalid index <*point index*

- – Could not publish point: Invalid trigger mode <*trigger mode*

- – Could not publish point: Invalid trigger mode <*trigger mode*for data type <*data type*

- – Could not publish point: Invalid QoS %u

- – Could not publish point: Re-publish <*known path*with path <*new path*

- – Could not publish point: Initial publish for <*path*(uncertain) without path

- – Could not publish point: Initial publish without path

- – Could not process publish command: Invalid library state

- – Publishing of groups not enabled

- – Could not publish point: Invalid data type <*data type*

- – Could not process publish command: Receive error

- – Could not process publish command: Send error

- – Could not process publish command: Unexpected error (<*error code*)

- – Could not send Nak for publish command

- – Could not process un-publish command: Invalid library state

- – Could not un-publish <*path*: Invalid index

- – Could not process un-publish command: Receive error

- – Could not process un-publish command: Send error

- – Could not process un-publish command: Unexpected error (<*error code*)

- – Could not send Nak for un-publish command

- – Could not process (committed) write command: Library not ready

- – Could not process (committed) write command: Receive error

- – Could not process (committed) write command: Invalid node index <*point index*

- – Could not process (committed) write command: Invalid data type <*data type*

- – Could not process (committed) write command: Receive queue is full

- – Could not process (committed) write command: Protocol error

- – Could not process (committed) write command: Send error

- – Could not send Nak for write command

- – Could not process commit command: Library not ready

- – Could not process commit command: Receive error

- – Could not process commit command: Protocol error

- – Could not process commit command: Send error

- – Could not process cancel command: Library not ready

- – Could not process cancel command: Receive error

- – Could not process cancel command: Protocol error

- – Could not process cancel command: Send error

- – Could not process get node properties response: Invalid node type <*node type*

- – Could not process get node properties response: Invalid index <*index*

- – Could not process get node properties response: Invalid path <*path*

- – Could not process get node properties response: Invalid flags <*hexadecimal flags*

- – Could not process get node properties response: Invalid data type <*data type*

- – Could not process get node properties response: Invalid trigger mode <*trigger mode*

- – Could not process get node properties response: Invalid QoS <*quality of service*

- – Could not process get node properties response: Invalid frame length

- – Could not process get node properties response: Invalid library state

- – Could not process get node properties response: Receive error

- – Could not process get node properties response: Unexpected error (<*error code*)

- – Could not send RPC request: Send error

- – Could not send RPC request: Unexpected error (<*error code*)

- – RPC request for method <*method*timed out

- – RPC response contains neither a result nor an error object

- – Mandatory RPC request field missing

- – Could not send RPC response: Send error

- – Exception in RPC method <*error message*

- – Could not execute kolibri.reboot() RPC: Invalid parameter or reboot not allowed

- – Could not execute kolibri.getStats() RPC: Error getting statistics (<*error code*)

- – Could not enqueue value for point <*path*: Out of memory

- – Could not enqueue value for point <*path*: Unexpected error (<*error code*)

- – Could not connect: Out of memory

- – Could not connect: Unknown error (<*error code*)

- – Could not send emergency command: Unexpected error (<*error code*)

- – Could not log in: Unexpected error (<*error code*))

- – Received Ack for node manipulation command in wrong state

- – Received Nak for node manipulation command in wrong state

- – Could not process publish command: Out of memory

- – Could not process un-publish command: Out of memory

- – Could not process (committed) write command: Out of memory

- – Could not process (committed) write command: Unexpected error (<*error code*)

- – Could not process commit command: Unexpected error (<*error code*)

- – Could not process cancel command: Unexpected error (<*error code*)

- – Could not process get node properties response: Out of memory

- – Could not send RPC request: Out of memory

- – Could not build RPC error object for unknown method <*method*: Out of memory

- – Could not send RPC response: Out of memory

- – Could not send RPC response: Unexpected error (<*error code*)

- – Could not execute kolibri.getRpcInfo() RPC: Out of memory

- – Could not execute kolibri.reboot() RPC: Out of memory

- – Could not execute kolibri.getStats() RPC: Out of memory

- • Portal

  - – Starting I/O synchronisation with portal.

  - – I/Os have been synchronised with portal.

  - – Could not get list of portal nodes. (Error code *error code*)

  - – Could not get node properties. (Node path *node path*, error code *error code*)

  - – Could not delete node. (Node path *node path*, error code *error code*)

  - – Could not create node. (Node path *node path*, error code *error code*)

  - – Could not modify node properties. (Node path *node path*, error code *error code*)

  - – Could not generate node path. (position *position*, error code *error code*)

  - – Could not synchronise I/Os with portal.

- • Variables

  - – Starting Variables synchronisation with portal.

  - – Variables have been synchronised with portal.

  - – Could not get list of portal nodes. (Error code *error code*)

  - – Could not get node properties. (Node path *node path*, error code *error code*)

  - – Could not delete node. (Node path *node path*, error code *error code*)

  - – Could not create node. (Node path *node path*, error code *error code*)

  - – Could not modify node properties. (Node path *node path*, error code *error code*)

  - – Could not synchronise Variables with portal.

- Messages

  - Sent message *<x>*

  - Message *<x>* was received.

  - Opening connection to MQTT broker

  - Connection to MQTT broker established

  - Sending message *<x>* failed.

  - A message received through the *<channel>* channel from *<sender>* was discarded, because it did not match any template. Message text: "*<text>*"

  - Out of memory

  - Could not connect to MQTT broker (*<error code>*)

  - Could not subscribe to MQTT topic "*<topic>*" (*<error code>*)

  - Closing connection to MQTT broker due to communication error (*<error code>*)

- I/Os

  - Could not find I/O module

  - Could not Initialise I/O module: Could not set port configuration

  - Could not Initialise I/O module: Could not disable master intensity

  - Could not Initialise I/O module: Could not set configuration

  - Could not Initialise I/O module: Could not adjust PWM direction

  - Initialising I/O hardware failed

  - Output power lost

  - Output power supplied

  - Output *<x>* overloaded

  - Output *<x>* no longer overloaded

  - PWM was set to illegal mode = *<mode>*.

  - Could not read Firmware version or it is not supported.

  - Could not open a communication channel with the module.

  - Unable to open Sensor configuration file. The file is either damaged or missing.

  - Could not initialise RTD module.

  - Portal communication error *<error code>*

- Modbus Client/Master

    - Connected to TCP device at *<IP address>*:*<port>*.

    - Read/wrote from/to TCP device at *<IP address>*:*<port>* (unit: *<unit identifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*)

    - Read/wrote from/to RTU device *<serial port>*:*<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*)

    - The interval is being maintained again.

    - The interval has been violated.

    - Could not connect to TCP device at *<IP address>*:*<port>*. (Error code *<code>*).

    - Could not read/write from/to TCP device at *<IP address>*:*<port>* (unit: *<unit idenfifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*, result code: *<result code>*, error code: *<error code>*, exception code: *<exception code>*)

    - Read illegal floating point value from TCP device at *<IP address>*:*<port>* (unit: *<unit idenfifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*)

    - Could not read/write from/to RTU device *<serial port>*:*<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*, result code: *<result code>*, exception code: *<exception code>*)

    - Read illegal floating point value from RTU device *<serial port>*:*<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*)

    - Portal communication error *<error code>*

- Modbus Server/Slave

    - Portal communication error *<error code>*

- M-Bus

    - Read values from device "*<manufacturer>* - *<medium>* - *<serial number>*".

    - Read value "*<description>*" from device "*<manufacturer>* - *<medium>* - *<serial number>*".

    - The readout cycle is being maintained again.

    - Could not re-initialise M-Bus library.

    - Scanning for slaves failed

    - Determining slaves' configuration failed

    - The readout cycle has been violated.

    - Could not read values from device "*<manufacturer>* - *<medium>* - *<serial number>*".

    - Could not read value "*<description>*" from device "*<manufacturer>* - *<medium>* - *<serial number>*".

    - Read illegal floating point value "*<description>*" from device "*<manufacturer>* - *<medium>* - *<serial number>*".

    - Portal communication error *<error code>*

- EtherNet/IP Adapter
    - EtherNet/IP module detected
    - Resetting Anybus CompactCom module
    - IP configured
    - DHCP client was (activated or deactivated)
    - Received illegal floating point value
    - Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)
    - Connection to Anybus CompactCom module lost
    - Connection to Anybus CompactCom module reestablished
    - Class 1 connection with scanner closed
    - Class 1 connection idle
    - Class 1 connection active
    - Class 1 connection error or duplicate IP address
    - Unexpected state: *<state>*
    - Could not (set or get) DHCP
    - Could not (set or get) IP address
    - Could not (set or get) subnet mask
    - Could not (set or get) gateway IP address
- PROFIBUS DP Slave
    - PROFIBUS DP module detected
    - Resetting Anybus CompactCom module
    - Received illegal floating point value
    - Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)
    - Connection to Anybus CompactCom module lost
    - Connection to Anybus CompactCom module reestablished
    - Deactivating module due to empty process data configuration
    - Left state Data Exchange
    - Entered state Data Exchange, Master in Clear mode
    - Entered state Data Exchange, Master in Operate mode
    - Could not set slave address
    - Unexpected state: *<state>*

- PROFINET IO Device

  - PROFINET IO module detected

  - Advanced settings activated

  - Resetting Anybus CompactCom module

  - IP configured

  - DHCP client was (activated or deactivated)

  - Received illegal floating point value

  - Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)

  - Connection to Anybus CompactCom module lost

  - Connection to Anybus CompactCom module reestablished

  - IO connection with IO controller closed

  - IO controller in STOP mode

  - IO connection with IO controller established

  - Configuration data mismatch

  - Unexpected state: *<state>*

  - Could not set DHCP

  - Could not set IP address

  - Could not set subnet mask

  - Could not set gateway IP address

- IEC 60870-5-104 Server

  - Updated output at address *<information object address>*.

  - The interval is being maintained again.

  - Timestamps with 16 or 24 bit are not yet supported. Falling back to system time..

  - A cyclic transmission time of types 1, 3, 5, 7, 30, 31, 33, 34 and 36 is no longer supported and has been converted to 0.

  - Values of types 15 and 37 are lacking a cyclic transmission time.

  - A cyclic transmission time in milliseconds is no longer supported and has been converted to seconds. Please check your configuration at type 'Measured value, normalized value (9)'.

  - Could not send activation termination of pulse command at address *<information object address>*.

  - Could not get timestamp at end of pulse command at address *<information object address>*.

  - The interval has been violated.

  - Could not update output at address *<information object address>*. Error code *<code>*, error message *<message>*

  - Portal communication error *<error code>*

- IEC 60870-5-104 Client

    - Updated output at address *<information object address>*.

    - The interval is being maintained again.

    - The interval has been violated.

    - Could not update output at address *<information object address>*. Error code *<code>*, error message *<message>*

    - Portal communication error *<error code>*

- IEC 61850

    - Server IP: *<IP address>*, IED: *<device name>*, connection status: connected.

    - (Wrote or Read) value *<value name>* of device *<device name>* on *<interface name>* interface.

    - The interval is being maintained again.

    - Server IP: *<IP address>*, IED: *<device name>*, connection status: not connected.

    - Could not (Wrote or Read) value *<value name>* of device *<device name>* on *<interface name>* interface. Error code *<x>*

    - The interval has been violated.

    - Portal communication error *<error code>*

- DNP3 Outstation

    - Updated output with type ID *type ID* and index *index*

    - Could not update output with type ID *type ID* and index *index*. Error code *error code*

- OPC UA Server

    - The interval is being maintained again.

    - The interval has been violated.

    - Insecure security policy enabled

    - No security policy enabled

    - Missing or invalid application instance certificate for SecurityPolicy - Basic128Rsa15 (code *<code>*)

    - Missing or invalid application instance certificate for SecurityPolicy - Basic256Sha256 (code *<code>*)

    - Unhandled error (code *<code>*)

    - Out of memory

- SGIM

    - The interval has been violated.

    - CAN bus initialisation error: *<Error Number>*.

    - Limit event for *value name*, channel *channel number*

    - An averaging counter has overflown. Average values are no longer reliable.

    - A CAN-controller error has occurred.

    - The CAN-controller is in the Busoff state.

    - A CAN-controller overrun error has occured.

    - The CAN receive buffer has overflowed.

    - The CAN transmit buffer has overflowed.

    - The CAN controller is in the init/busoff state.

    - The CAN controller is in the error passive state.

- CODESYS Network Variables

    - The interval is being maintained again.

    - The interval has been violated.

    - The interval has been violated.

    - Error receiving CODESYS Network Variable pack with COB-ID *<COB-ID>*, sub-index *<Sub-Index>*. The received packet contains an invalid float value.

    - Error receiving CODESYS Network Variable pack with COB-ID *<COB-ID>*, sub-index *<Sub-Index>*. The received checksum is invalid.

    - Error in receiving transmission acknowledgement packet for CODESYS Network Variable packet with COB-ID *<COB-ID>*.

    - Error sending variable in CODESYS Network Variable list with COB-ID *<COB-ID>*, variable index *<Sub-Index>*. Error code: *<Error-Code>*.

    - Variable with name: *<Variable Name>*, in list with COB-ID *<COB-ID>* (sub-index *<Sub-Index>*) has an invalid or unsupported data type.

    - Received CODESYS Network Variable acknowledgement packet with COB-ID *<COB-ID>*, packet index *<Packet-Index>* with code Error code: *<Error-Code>*.

    - Portal communication error *<error code>*

- SNMP Manager

    - (Wrote or Read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*).

    - Could not (write or read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*). (error status *<x>*, error index *<x>*)

    - Could not (write or read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*). (*<exception>*)

    - Could not process value *<value name>* (OID *<value oid>*) in trap/inform request from agent *<agent name>* (IP *<agent address>*). (*<msg>*)

    - Could not process message from agent *<agent name>* (IP *<agent address>*). (*<exception>*)

    - Could not send response to INFORM request from agent *<agent name>* (IP *<agent address>*). (Error code *<x>*)

- COM Server
    - Started.
    - Stopped.
    - Incoming connection from *<x>*.
    - Requested unsupported parity, set to no parity.
    - Requested unsupported 1.5 bits stop size, set to 1 bit stop size.
    - Inbound flow control ignored.
    - Requested unsupported flow control.
    - Client doesn't support Telnet COM Port Protocol Option (RFC 2217), trying to serve anyway.
    - Unable to open serial port *<x>*. Exiting.
    - Socket open failed. Exiting.
    - Socket bind failed. Exiting.
    - Socket listen failed.
    - Socket accept failed.
    - Error writing to network. Close socket.
    - Error reading from network. Close socket.
- Data Logging
    - Online.
    - Offline.
    - Restoring connection.
    - Waiting for database connection.
    - Device configuration has changed.
    - Value(s) could not be found.
    - Could not create database.
    - Data was lost.
    - Could parse configuration file: *<error>*.
    - Erasing old entries.
    - Connection lost.
    - Online.
    - Offline.
    - Could not open.
    - Could not close.

- Export

    - Preparing export...

    - Packing update package...

    - Update package has been successfully packed

    - Preparation failed. Message: *<message>*

    - Packing failed. Message: *<message>*

- Update

    - Running

    - Package successfully loaded via local web server

    - Loading package from remote web server...

    - Package successfully loaded from remote web server

    - Checking package...

    - About to Reboot...

    - Loading package via local web server timed out

    - Target directory for remote update package does not exist

    - Target file for remote update package is not accessible

    - Loading package from remote web server failed (code *<x>*)

    - The type of the package's signature does not match the expected type of signature. (*<signature type>*)

    - Invalid package flag(s) (*<flags>*)

    - Firmware name does not match. This package is for "*<firmware name>*" firmware.

    - This package cannot be applied to the current firmware version.

    - Target name does not match. This package is for "*<target name>*" target.

    - Variant name does not match. This package is for "*<variant name>*" variant.

    - This package is limited to the device with the MAC address *<MAC address>*.

    - Invalid package signature

    - Could not open package. Message: *<message>*

    - Update disallowed by application (*<code>*)

    - Could not launch update.

    - Portal communication error *<x>*

- System Start-up
    - Started
    - Portal communication module initialised
    - I/O hardware manager initialised
    - NAT/firewall service initialised
    - Ethernet (1) driver started.
    - Cellular modem driver initialised
    - Remote Update deactivated
    - Offline Update deactivated
    - Update module initialised
    - PLC driver initialised
    - Message manager initialised
    - DHCP server initialised
    - Database Manager initialised
    - OpenVPN client initialised
    - Status web service initialised
    - System time manager initialised
    - COM Server initialised
    - Program loaded
    - WLAN driver initialised
    - Web configuration modules initialised
    - Update exporter module initialised
    - Update web service initialised
    - Running
    - Starting update from external medium.
    - Link detected at Ethernet (*<interface index>*), Port (*<port index>*).
    - Starting DHCP configuration at Ethernet (*<interface index>*), Port (*<port index>*).
    - DHCP configuration renewed at Ethernet (*<interface index>*).
    - DHCP configuration completed at Ethernet (*<interface index>*).
    - Could not initialise power fail handling.
    - Power fail condition present at start-up
    - Retentive data could not be loaded.
    - Not enough power fail capacity
    - Power fail handling not supported
    - The internal flash drive seems to be weak.
    - Flash write error. The internal flash drive is probably defect.
    - Could not create directory structure on external medium
    - One or more certificates could not be loaded.

- – Could not initialise NAT/firewall service. (*<error code>*)

- – Could not start Ethernet (1) driver.

- – Unsupported Ethernet (1) interface type. (*<interface type>*)

- – Could not initialise NAT/firewall service. (*<error message>*)

- – Could not initialise cellular modem driver: *<error message>*

- – Could not initialise WLAN driver: *<error message>*

- – Could not initialise LCD: *<error message>*

- – Could not initialise update module: *<error message>*

- – Could not initialise DHCP server.

- – Could not initialise Database Manager.

- – Cannot monitor external medium insertion

- – Could not initialise OpenVPN client: *<error message>*

- – Could not initialise system time manager: *<error message>*

- – This device is not licenced for HMS Hub.

- – Could not initialise COM server: *<error message>*

- – Could not initialise portal Event Log service: *<error message>*

- – Could not load and start program.

- – Could not load and start program: *<error message>*

- – Not starting program due to problems with retentive data

- – Could not initialise web configuration modules: *<error message>*

- – Could not initialise update exporter module: *<error message>*

- – Could not initialise update web service: *<error message>*

- – Could not initialise diagnostics console: *<error message>*

- – Could not start update from external medium.

- – External medium update file could not be found.

- – Link lost at Ethernet (*<interface index>*), Port (*<port index>*).

- – This RTOS version is not supported. Version (*<version number>*) is required.

- – This bootloader version is not supported. Version (*<version number>*) is required.

- – More than one clock synchronisation mechanism is active.

- – Device is secured with the default password, please change it.

- – Fatal error: *<error message>*

4.2   Settings

4.2.1   General

User Management

With this form users can be managed. The system allows up to 8 users. The users are displayed in a table that shows the user name and a description. With the buttons below the table users can be added, edited, deleted or cloned. When adding or editing a user a dialog shows up that allows to edit the following parameters:

User name	User name
Description	This field allows to describe the user. The description is shown only within the user management.
Password	The password that the user uses to authenticate. When editing a user the password fields can be left empty in order to keep the old password.
Password must be changed at next login	If this check box is set, the user will be forced to change his/her password on next login.
Access rights	Here access rights for various device functions can be granted. Often the are two check boxes, one for the right to show data and one for the right to modify data. The access rights are grouped. At the top of each group there are check boxes that can be used to grant or revoke all access rights in the group.

Further on in the user management form the password policy can be configured. It determines the minimum length of passwords and the types of characters that passwords must contain. Additionally a blacklist can be defined. Passwords on the blacklist must not be used. The list contains one password per line.

System Time

Through this site you can adjust several settings concerning the system time.

In the first section you can define the interval (in minutes) in which the system clock will be synchronised through SNTP or with the real-time clock.

SNTP Client

In this section you can enable and configure the SNTP client. If you enable it, the given time servers will be contacted to determine the current time and date. Then the system clock will be updated accordingly. You can specify up to four time servers, which will be queried one after the other until one of them answers. The server names can be given either as host names (e.g. ntp. company.org) or as IP addresses (e.g. 192.168.0.23). Note that contacting the time servers produces data traffic.

Time Zone Settings

In this section you can enable and configure the Anybus Edge device time zone settings. If you enable it, the Anybus Edge device will update its local time according to the time zone and daylight saving time rules of its configured location.

Set Clock Manually

In this section you can set the system clock manually. When you click on the *Set* button, the system clock will be set to the current time of your computer.

RTC

If your Anybus Edge features a real-time clock (RTC), it will be automatically used without any configuration.

Error Log Messages

The following section describes the messages that the system time manager and the SNTP client may log to the Event Log. All Messages are prefixed with the text *System time manager:* or *SNTP client:*.

- **Running**

  The system time manager is now running.

- **System clock updated via SNTP**

  The system clock (and if available the real-time clock) has/have just been updated with the time gotten via SNTP.

- **System clock updated from RTC**

  The system clock has just been updated with the time from the real-time clock.

- **Could not get exclusive access to clock(s)**

  The system time manager could not get exclusive access to the system clock and (if available) the real-time clock. Thus the system clock could not be set or the real-time clock could not be read.

- **Daylight saving time rules will be used up within the next two years**

  The firmware of your Anybus Edge device includes daylight saving time rules for a limited time period. New rules will be added with new firmware versions. This message indicates that the rules will be used up within the next two years. You should apply a firmware update to get new rules.

- **SNTP client error: *<message>***

  The SNTP client has returned the shown error message.

- **RTC not found**

  The device should feature a real-time clock, but it cannot be accessed.

- **Could not write to RTC: *<message>***

  The real-time clock could not be written. The message shows an error message returned by the real-time clock driver.

- **Could not read from RTC: *<message>***

  The real-time clock could not be read. The message shows an error message returned by the real-time clock driver.

- **Could not read daylight saving time file**

  The daylight saving time file for this time zone is missing or corrupted.

- **No daylight saving time rule found for the current time**

  No daylight saving time rule could be found for the current time. That means either that the daylight saving time rules are outdated or that the Anybus Edge's time is set wrongly.

- **DNS error (<error code>) for server <server name/IP>**

  The SNTP client could not determine the IP address of the server with the given name. The message also shows the error code returned by the TCP/IP stack.

- **Could not open socket for server *<server name/IP>***

  The SNTP client could not open a UDP socket in order to contact the time server with the given name or IP address.

- **Could not send to server *<server name/IP>***

    The SNTP client could not send a request to the time server with the given name or IP address.

- **Failed to receive reply from server *<server name/IP>***

    The SNTP client could not receive a reply from the server with the given name or IP address.

- **Unexpected packet format from server *<server name/IP>***

    The SNTP client received an invalid reply from the server with the given name or IP address.

- **Timestamps not plausible from server *<server name/IP>***

    The SNTP client received a reply from the server with the given name or IP address where the timestamps are not plausible.

LCD

Through this form you can configure general settings concerning the LCD of the device.

Info	The LCD can draw 16 x 2 dot matrix characters.
Programmable	If this option is set, the LCD will become freely programmable stopping its default functions. The Interface LCD will appear in the device I/O tree. If the old Diagram elements are still used they will reserve the LCD access.
Backlight timeout	Timeout in seconds after which the LCD backlight will be deactivated.
Refresh rate	Refresh rate in seconds after which the LCD will be redrawn.

LCD Interface Help.

Info	The LCD has the following dimensions 480 x 272 px.
LCD Ctrl output group	The LCD Ctrl writes a value to the LCD. At a rising edge on the *En* output the current value of the *In* output will be printed to the LCD. The X (column) and Y (row) coordinates can be given through the *X* and *Y* outputs. If the *Clear* output's value is nonzero, the display will be cleared before printing.
LCD Keys output group	The LCD Keys provides the state of the keys next to the LCD. If a key is pressed, the corresponding output has the value *1* else *0*.
LCD Keys input group	Through the LCD LEDs various LEDs of the LCD can be controlled. If an input has a nonzero value, the corresponding LED will be on: Up Backlight of the *Up* key Down Backlight of the *Down* key Left Backlight of the *Left* key Right Backlight of the *Right* key Enter Backlight of the *Enter* key Esc Backlight of the *Esc* key LCD Backlight of the LCDs Red Red status LED

Website	The website shown on the Display.
Orientation	The Display orientation.

4.2.2　Network

Via this form you can configure general network settings.

In the first section you can configure the device name and enable IP forwarding. The device name will be registered as a NetBIOS name. So inside your local network you should be able to use the name instead of the IP address to access the Anybus Edge device. In addition to the configurable name another name built from the prefix *MAC* and the Anybus Edge device's MAC ID (e.g. *MAC003056A1DB30*) will be registered. The default gateway may be overwritten by means of dynamic IP configuration, e.g. via DHCP or when a cellular modem connection is established. If you enable IP forwarding, the Anybus Edge device will forward IP traffic from one network interface to another. This is for example needed for NAT.

Ethernet (0)

In this section you can configure the IP address settings of your Anybus Edge device's (first) Ethernet interface. Note that after changing these settings, the Anybus Edge device may not be reachable under the same address any more.

Ethernet (1)

If your Anybus Edge device features a second Ethernet interface this section will show up. Here you can configure the IP address settings for the second Ethernet interface.

DNS

In this section you can explicitly configure the IP address of a DNS server. If your Anybus Edge device is configured to get its IP configuration via DHCP or if you use a cellular modem connection, you probably will not need this setting. The DNS server will be configured automatically in this case. If you still enter a DNS server manually, it will always be queried before the automatically configured DNS servers.

HTTP Proxy

In this section you can configure a proxy server that should be used for HTTP connections.

Please contact your local network administrator if you're unsure about which values to enter here.

Cellular Modem

Via this form you can enable and configure the cellular modem. In the first section you can generally enable the modem. Please enter the SIM PIN if your SIM requires one. You can allow or disallow the modem to log into a network other than your home network (aka roaming). Note that roaming connections will normally produce higher costs. To establish a data connection via GPRS or UMTS, check the respective box.

There is also an option to use the cellular modem connection as a fallback connection. In this case no data connection will be established through the cellular modem as long as any other network interface provides internet connectivity. Only when all other network interfaces lose their internet connectivity, the cellular modem data connection will be established. The connection supervision (see below) will be used to supervise as well the cellular modem connection as the other connections. When the cellular modem connection is established, it will be used to check if one of the other interfaces provides internet connectivity again. Only after the connection supervision has succeeded on another network interface three times in a row, the cellular modem data connection will be closed.

In the second section you must configure the access data, if you want to establish a data connection. The most important setting here is the APN (access point name). If authentication is mandatory, you can configure the authentication type as well as user name and password here. You should have received the data to put into these fields from your mobile phone provider.

Connection Supervision

The connection supervision will try to connect to the first URL whenever there has been no payload traffic for the given period of time. The period can be specified between 60 and 60000 seconds. If connecting to the first URL fails, the second URL will be tried. If the second URL fails, too, the cellular modem connection will be re-established. Use this function if you fear there might be connection problems not recognised by the cellular modem driver. But note that this function may produce additional network traffic. Note that the connection supervision may not be disabled, if the cellular modem connection is used as a fallback connection (see above).

> **!** For applications that require continuous data collection (for example the PORTAL or a telecontrol protocol) the connection supervision has to be enabled. This function supervises the cellular modem connection and resets the modem when the connection is broken. Also note that the predefined supervision URLs are only reachable with SIM cards that allow access to the internet.

> **!** When the cellular modem connection is used as a fallback connection, the connection supervision URLs must resolve to IP addresses that are not used by any other component of the device, because these addresses may not be reachable while the connection supervision is checking disconnected network interfaces.

Portal

The cellular modem driver will transmit the RSSI, the data service type and the SIM ID to the portal. The data service type is transmitted numerically. The values have the following meaning:

0	No data service
1	GPRS
2	EDGE
3	UMTS
4	UMTS with HSDPA
5	UMTS with HSUPA
6	UMTS with HSDPA and HSUPA

Also the GPS coordinates latitude, longitude and altitude will be transmitted. If the GPS module cannot provide a horizontal fix, the quality for latitude and longitude will be set to *bad*. The same applies to the altitude's quality, if the GPS module cannot provide a vertical fix.

GPS

If the cellular modem built into your Anybus Edge device features a GPS module, you can enable the GPS module. Once enabled, the GPS module will provide the coordinates latitude, longitude and altitude. Please note that getting the coordinates may take several minutes as long as the module does not have any knowledge of its position (cold start). This does not affect any data connection, but neither the RSSI can be determined nor short messages can be sent or received while the driver queries the GPS module.

Base Station Assisted Location

If the "Base station assisted location" option is enabled, the devices' location will be determined by triangulating the base stations with which the device is connected.

Using GPS and base station assisted location simultaneously is not possible.

The device handles base station assisted location requests automatically. It will send up to three requests at system start until the highest location certainty has been archived. After system start it will send a request each hour. This service can only be used for up to 1000 requests per calendar month. If the maximum number of requests to this service is exceeded acquiring the devices' position will be no longer possible.

Base station assisted location only supports two dimensional location of the device (no altitude).

Signal Strength

The signal quality has decisive influence on the speed of the data transmission. Low signal strength may decrease the transmission speed significantly or even completely drop the connection and may result in a loss of function of the device. A general signal strength to connection quality correlation is given by the following table:

Under -101 dBm	Bad
-101 dBm to -90 dBm	Weak
-89 dBm to -80 dBm	Medium
-79 dBm to -65 dBm	Good
Over -65 dBm	Excellent

To ensure a stable connection and an adequate performance of all the communication functions of the device, please make sure that the signal level is at least -79 dBm (Good).

Critical Applications

Cellular telecommunication devices operate using radio signals and therefore are not guaranteed to have a reliable connection in all conditions. You should not rely solely on a wireless device for reliable continuous data transmission.

Antenna

Keeping your body close to a cellular antenna might negatively affect your health. Usage of antennas with higher a gain of more than 2.3 dBi might require further approval.

Event Log Messages

The following section describes the messages that the cellular modem driver may log to the *Event Log*. All messages are prefixed with the text *Cellular Modem Driver:*.

- **Stopped**

  The cellular modem driver has been stopped.

- **Started**

  The cellular modem driver has been started.

- **Running**

  The cellular modem driver is now running.

- **Deactivated**

  The cellular modem driver has been temporarily deactivated.

- **Activated**

  The cellular modem driver has been re-activated after temporary deactivation.

- **Powered modem**

  The supply voltage for the cellular modem hardware has been switched on.

- **Modem initialised (RSSI: *<x>* dBm)**

  The cellular modem has been successfully initialised. The RSSI value ranges from <= *-113* to = *-51* dBm.

- **SIM card ID: *<x>***

  This message informs about the SIM card's ID.

- **Registered to network**

  The cellular modem has successfully registered to a cellular network.

- **Roaming**

  The cellular modem has registered to a network other than the home network. Note that roaming connections will normally produce higher costs.

- **Data connection established**

  A data connection has been successfully established.

- **Opening fallback connection**

  The cellular modem is being used as a fallback interface. The connection is being established, because no other network interfaces provides internet connectivity.

- **Closing fallback connection**

  The cellular modem is being used as a fallback interface. The connection is being closed, because any other network interface provides internet connectivity.

- **GPS module initialised**

  The cellular modem's GPS module has been initialised.

- **SIM card not present**

  No SIM card is present in the SIM card slot.

- **Invalid PIN - [not] retrying**

  The given PIN is not correct for the inserted SIM card. The message also indicates whether the driver will retry to unlock the SIM card with the given PIN. This will be the case, if the SIM card could be unlocked at least once before. Note that if the SIM card is exchanged at run-time, the wrong PIN may be retried repeatedly and the SIM card may become locked.

- **SIM card locked, PUK required - [not] retrying**

  The SIM card is locked. The PUK is required. The message also indicates whether the driver will retry to unlock the SIM card with the given PIN. This will be the case, if the SIM card could be unlocked at least once before.

- **SIM card failure**

  The SIM card has been permanently locked. Please contact your mobile network provider.

- **Modem initialisation failed. (code *<x>*)**

  Initialisation of the celullar modem hardware failed. Possible codes:

14	The SIM card is busy.
-102	The cellular modem hardware returned an error.
-103	The cellular modem hardware did not answer.
-105	A reply from the cellular modem hardware was too long.
-134	The cellular modem hardware is invalid.

  If you see an error code that is not on the above list, please contact the customer support.

- **Network registration timed out ((not) searching)**

  No cellular network could be found within a reasonable time. The message also indicates whether the cellular modem hardware was still searching for networks.

- **Roaming disallowed**

  The cellular modem did not connect to any network, because the home network was not available and roaming has been disallowed via the configuration.

- **Network registration failed (code *<x>*, network status *<y>*)**

  The cellular modem could not register to any network. Possible codes:

0	See network status.
14	The SIM card is busy.
-102	The cellular modem hardware returned an error.
-103	The cellular modem hardware did not answer.
-105	A reply from the cellular modem hardware was too long.
-134	The cellular modem hardware is invalid.

  Possible network statuses:

0	Not registered to any network
1	Registered to home network
2	Not registered to any network; searching for a network to register to.
3	Registration has been denied by the designated network.
4	Unknown network status
5	Registered to a network other than the home network (roaming)

  If you see an error code or network status that is not on the above lists, please contact the customer support.

- **Data connection failed (code *<x>*)**

  No data connection could be established. Possible codes:

-3	Invalid user name or password
-4	The network interface could not be opened. This is a quite general error. It could for example mean that logging in to your mobile provider's network failed.
-6	An invalid IP configuration was received from the provider.
-7	No PPP connection could be established.

  If you see an error code that is not on the above list, please contact the customer support.

- **Network lost (code *<x>*, network status *<y>*)**

  The cellular modem lost the connection to the network. Possible codes:

0	See network status.
14	The SIM card is busy.
-102	The cellular modem hardware returned an error.
-103	The cellular modem hardware did not answer.
-105	A reply from the cellular modem hardware was too long.
-134	The cellular modem hardware is invalid.

  Possible network statuses:

0	Not registered to any network
1	Registered to home network
2	Not registered to any network; searching for a network to register to.
3	Registration has been denied by the designated network.
4	Unknown network status
5	Registered to a network other than the home network (roaming)

  If you see an error code or network status that is not on the above lists, please contact the customer support.

- **Data connection broken ([DCD pin: *<x>*,]PPP status: *<y>*, Data connection: *<z>*)**

  The data connection has been unexpectedly closed. The message also shows the statuses of the cellular modem hardware's DCD pin (if available, 1: connected, 0: not connected), the data connection as indicated by the cellular modem (1: connected, 0: not connected) and the status of the PPP client network interface:

-1	The PPP client is not running.
0	Link is down
1	Link opening in progress
2	Link is established
3	The PPP client has been temporarily paused.

  If you see an error code that is not on the above list, please contact the customer support.

- **Connection supervision failed**

  The connection supervision has failed to connect to the configured URLs. The connection will be closed.

- **Modem reset**

  The cellular modem hardware has been reset.

- **Hanging! (state: *<x>*)**

  The cellular modem driver is hanging in an invalid state. This should not happen.

- **Could not initialise GPS module**

  The cellular modem's GPS module could not be initialised.

WLAN

Via this form you can activate and configure the WLAN interface of your Anybus Edge device, in case it provides one.

Settings

The following table describes the available settings for the WLAN module.

Enable WLAN	Enables the module
SSID	Network name/identifier (0-32 printable characters). If no SSID is specified, the driver stops its association attempts.
Security	Select a security method: Open, or WPA(2)-PSK (recommended).
WPA Passphrase	The passphrase for WPA(2)-PSK security must consist of 8-63 characters.
Use DHCP	If this option is set, the module will get its IP configuration through DHCP.
IP address	IP address for the WLAN interface.
IP netmask	Netmask for the WLAN interface.

> ! It is strongly recommended to modify the WLAN settings only via the Ethernet (hard-wired) connection. Changing the configuration via the WLAN connection itself may result in a loss of the currently established connection, after a device reboot.

Wireless Scanning

Press the *Scan* button to search for available wireless networks.

The scan takes about 20 seconds. Afterwards the found wireless networks will be shown in a dialog.

Select the desired wireless network using the *Select* button or be double-clicking the corresponding entry in the list. The selected wireless network's settings will be applied to the configuration pase. If necessary the WPA passphrase must be provided respectively.

WPS Push-button Method

Press the *WPS* button to initiate a WPS connection. The wireless module attempts to connect to an access point that has also had its WPS Push-button method activated.

The WPS Push-button method must be activated on both devices (Anybus Edge and access point) within 2 minutes.

After a successful connection establishment, the settings for the connected wireless network are automatically saved on the device.

Event Log Messages

The following section describes the messages that the WLAN driver may log to the *Event Log*. All messages are prefixed with the text *WLAN Driver:*.

- **Started**

  The WLAN module driver has been started.

- **Activated**

  The WLAN module driver has been activated.

- **Running**

  The WLAN module driver is now running.

- **IP address obtained successfully through DHCP server**

  The WLAN module driver has configured its IP address through the DHCP protocol.

- **Stopping the driver**

  The WLAN module driver is being stopped.

- **Stopped**

  The WLAN module driver has been stopped.

- ***<status>***

  This message of the WLAN driver indicates a change of the module state. Possible states are the following:

Connected	Success, module is connected
Disconnected	Module has been disconnected from the access point
Access point not found	Selected access point was not found
Authentication failed	Authentication with the access point failed
hanging	The module is not accessible
restarting	The module restarts

- **Hardware initialization error *<error code>***

  A general error has occurred during the hardware initialization. Possible error codes:

-1	Waiting for module response timed out
-2	Communication with the module could not be established

  If you see an error code that is not on the above list, please contact the customer support.

Bluetooth

Via this form you can activate and configure the Bluetooth interface of your Anybus Edge device, in case it provides one.

Settings

The following table describes the available settings for the Bluetooth module.

Enable Bluetooth	Enables the module.
Device Name	Local device name shown on remote devices during discovery (0-248 printable characters).
PIN	PIN code used for older (legacy) pairing method (1-16 printable characters).
IP address	IP address for the Bluetooth interface.
IP netmask	Netmask for the Bluetooth interface.
Enable DHCP Server	Enables the DHCP server for the Bluetooth network interface.
Lease time	Duration of an assigned IP address, before renewal.

Pairing Button

The Anybus Edge Bluetooth module is equipped with a physical pairing button. The user may use this button as follows:

With a short press of the button, the Anybus Edge Bluetooth module becomes pairable for 2 minutes. If the user presses the button again before the 2 minute timeout has passed, the Anybus Edge becomes unpairable again.

If the user presses the button for longer than 2 seconds, then all stored Bluetooth devices are deleted, and they must attempt to pair again with the Anybus Edge.

Event Log Messages

The following section describes the messages that the Bluetooth driver may log to the *Event Log*. All messages are prefixed with the text *Bluetooth Driver:*.

- **Started**

    The Bluetooth module driver has been started.

- **The driver is running**

    The Bluetooth module driver is now running.

- **Pairing button pressed, device pairable**

    After the user has pressed the pairing button once, the Bluetooth device is pairable for 2 minutes.

- **Pairing button pressed, device unpairable**

    After the user has pressed the pairing button once again, the Bluetooth device is unpairable again.

- **Pairing timeout, device unpairable**

    2 minutes have passed since the user pressed the pairing button, the Bluetooth device is unpairable again.

- **Pairing button long press, all stored remote devices removed**

    After the user has pressed the pairing button for more than 2 seconds, all stored devices are deleted.

- **Stopping the driver**

    The Bluetooth module driver is being stopped.

- **Stopped**

  The Bluetooth module driver has been stopped.

Firewall

Your Anybus Edge device features a Firewall which improves the system's security by analysing network traffic and blocking unallowed traffic.

The Anybus Edge device Firewall has a blocking policy. This means that while outgoing connections are generally allowed, incoming connections will be generally blocked except if there is a rule allowing the specific connection.

At the top of the settings form you can enable the Firewall and configure the maximum number of rules. This determines the number of connections (incoming and outgoing) that can be handled simultaneously.

> **!** When enabling the Firewall, always make sure that you defined some static rules that allow you to access the web server etc. in order to be able to disable the Firewall if needed. If you enable the Firewall without any appropriate rules, you will be unable to access your device in any way. You will need to reset the device to factory settings in this case.

Dynamic Rules

Dynamic rules are created by the Firewall for outgoing connections. You have to configure how many minutes dynamic rules will be valid (time to live).

Static Rules

You can add up to 50 static Firewall rules. Static rules allow access to a specific service on your Anybus Edge device from the network. Each rule consists of the following parameters:

Network interface	Determines the network interface from which you want to accept connections. You may e. g. want to allow access to the device's web server only from the local area network. So you would choose the Ethernet interface here. If you do not care about the network interface, select *Any*.
Service	This selection provides a set of predefined services which you may want to allow to be connected to. If the desired service is not on the list, choose *Other* in order to manually specify the service's protocol and port(s) (see below).
Protocol	This parameter will show up only if you choose *Other* for the *Service* field. It determines which base protocol (TCP or UDP) the service you want to allow uses.
Minimum/maximum port	These parameters will show up only if you choose *Other* for the *Service* field. They determine the port(s) the service you want to allow runs on. You can specify a single port (set minimum and maximum to the same value) or a port range (e.g. 10000 to 10005). Or you can set both parameters to 0, which means that you want to allow connections on any port (wildcard).
Minimum/maximum source IP address	Determines the IP addresses you want to allow to connect to the specified service. You can specify a single IP address (set minimum and maximum to the same value) or an IP address range (e.g. 192.168.0.1 to 192.168.0.10). Or you can set both parameters to 0.0.0.0, which means that you want to allow connections from any IP address (wildcard).

Event Log Messages

The Firewall is implemented inside the NAT service module. Event Log messages are documented in the *Network Address Translation* section.

Network Address Translation

The HMS Hub firmware is able to perform network and port address translation (aka NAT/PAT) in order to route traffic between a local (internal) and a global (external) network. This feature can for example be used to allow another device connected to the Anybus Edge device's

Ethernet interface (local network) to use the Anybus Edge device's cellular modem internet connection (global network).

At the top of the settings form you can enable the NAT module and configure the internal and external network interface as well as the maximum number of mapping entries. This is the number of connections (incoming and outgoing) that can be handled simultaneously.

> ⚠ Note that IP forwarding will be implicitly enabled, because it is needed in order to use NAT. When you later disable NAT, IP forwarding will not be automatically disabled. This can be done via *Network*.

Dynamic Mappings

Dynamic mapping is used to enable devices from the local network to access the global network. Requests from the local network will be given the Anybus Edge device's global IP address and the replies will be routed to the respective local device. You have to configure how many minutes dynamic mappings will be valid (time to live).

Static Mappings

Static mappings are used to make a local device's service accessible from the global network under the Anybus Edge device's global IP address. Up to 50 static mappings can be configured. For each you have to configure these options:

Protocol	TCP or UDP
External port	The Anybus Edge device's port at which the service should be reachable from the global network.
Internal address	The IP address of the device in the local network whose service is to be published.
Internal port	The port number at which the service is available on the device in the local network.

One-to-one Mappings

One-to-one mappings are used to statically map two networks to each other one-to-one. The first host address in one network will be mapped to the first host address in the other network etc. If you want to alter the destination of incoming connections from 1.2.3.0/24 (external) to 5.6.7.0/24 (internal) for example, 1.2.3.0 with subnet mask 255.255.255.0 would be used as external address and 5.6.7.0 with subnet mask 255.255.255.0 as internal address. Up to 50 static mappings can be configured. For each you have to configure these options:

External address:	IP address of the external network
External subnet mask:	The subnet mask used for the external addresses. Ideally the size of both networks should be the same so that all host addresses can be mapped.
Internal address:	IP address of the internal network
Internal subnet mask:	The subnet mask used for the internal addresses. Ideally the size of both networks should be the same so that all host addresses can be mapped.

Event Log Messages

The following section describes the messages that the NAT service module may log to the *Event Log*. Messages are prefixed with the text *NAT service:*. Note that the NAT service module also implements the Firewall service. So there are also some Firewall-related messages.

- **Running**

  The NAT service module has been successfully started.

- **<x> static mapping(s) read from configuration**

  The static mapping entries have been successfully read from the configuration. The message also shows the number of static mappings read.

- **No free mapping entry available for incoming connection from internal interface**

  A client from the internal network tried to open a connection through the NAT service module. But there was no free entry in the mapping table. So the connection could not be established. Try to increase the maximum number of mappings to solve the problem.

- **No free firewall rule entry available for outgoing connection**

  A new dynamic rule has to be added to the Firewall to allow establishment of an outgoing connection. But there was no free entry in the internal rules table. So the connection could not be established. Try to increase the maximum number of rules to solve the problem.

OpenVPN

Via this form OpenVPN can be configured. You can use it to integrate your device into a virtual private network.

Connect at start-up:	If this option is checked OpenVPN will be started at device start-up.
Allow connection to be controlled via HMS Hub:	If this option is checked OpenVPN can be started and stopped though the portal. HMS Hub devices have a portal node at address */device-info/openvpn/running*. Setting this node's value to 1 will start OpenVPN. Of course you can also stop OpenVPN by resetting this node to 0.

OpenVPN configuration file:	Here the OpenVPN configuration file can be edited. There are some restrictions concerning the available options:
	
	• For the *dev* option only the values *tun* and *tap* (without Suffix) are allowed.
	
	• Options that refer to uploaded files (see below) must have a fixed value. This value is automatically set when storing the configuration. However these options will not be added automatically. This applies to the following options:
	
	   – ca ca.crt
	
	   – cert cert.crt
	
	   – key cert.key
	
	   – auth-user-pass userpass.txt
	
	   – dh dh.pem

- The following options are not allowed:

  – ipchange

  – iproute

  – route-up

  – route-pre-down

  – up

  – down

  – script-security

  – user

  – group

  – cd

  – chroot

  – setcon

  – daemon

  – syslog

  – errors-to-stderr

  – inetd

  – log

  – log-append

  – suppress-timestamps

  – writepid

  – nice

  – echo

  – remap-usr1

  – status

  – management

  – management-client-user

  – management-client-group

  – plugin

  – ifconfig-pool-persist

  – client-connect

  – client-disconnect

  – client-config-dir

  – tmp-dir

  – learn-address

  – auth-user-pass-verify

  – secret

  – replay-persist

  – capath

  – extra-certs

  – pkcs12

  – askpass

  – tls-verify

  – tls-export-cert

	– genkey
	– tls-crypt
	– keying-material-exporter
Authority's certificate file (ca.crt)	Here the authority' certificate file can be selected. It can be referenced in the configuration file as *ca.crt*.
Certificate file (cert.crt)	Here the certificate file can be selected. It can be referenced in the configuration file as *cert. crt*.
Private key file (cert. key)	Here the private key file can be selected. It can be referenced in the configuration file as *cert. key*.
User and password file (userpass.txt)	Here the user and password file can be selected. It can be referenced in the configuration file as *userpass.txt*.
Diffie-Hellman file (dh. pem)	Here the Diffie-Hellman file can be selected. It can be referenced in the configuration file as *dh.pem*.
TLS pre-shared key file (tlsauth.key)	Here the TLS pre-shared key file can be selected. It can be referenced in the configuration file as *tlsauth.key*.

> **!** Changes to the OpenVPN configuration may take effect before restarting the device when the OpenVPN connection is (re-)established. It is recommended to reboot the device shortly after changing the configuration to preserve a consistent state.

Event Log Messages

The following section describes the messages that OpenVPN may log to the Event Log. All Messages are prefixed with the text *OpenVPN:*. Note that the following list in not complete. You will also see messages originating directly from OpenVPN itself. Please refer to OpenVPN's documentation for further information.

• **Driver has been started**

   The OpenVPN driver has been successfully started.

• **Driver has been stopped**

   The OpenVPN driver has been successfully stopped.

• **Driver stopped**

   The OpenVPN driver has exited.

• **Connection established**

   An OpenVPN connection has been established.

• **Connection closed**

   An OpenVPN connection has been closed.

• **Authority's certificate uploaded**

   The certification authority's certificate file (*ca.crt*) has been successfully uploaded.

• **Certificate uploaded**

   The certificate file (*cert.crt*) has been successfully uploaded.

• **Private key uploaded**

   The private key file (*cert.key*) has been successfully uploaded.

• **User and password file uploaded**

   The user and password file (*userpass.txt*) has been successfully uploaded.

• **Diffie-Hellman file uploaded**

   The Diffie-Hellman file (*dh.pem*) has been successfully uploaded.

- **Could not start driver**

  The OpenVPN driver could not be started.

- **Could not stop driver**

  The OpenVPN driver could not be stopped.

- **Driver stopped unexpectedly**

  The OpenVPN driver has stopped unexpectedly. This could for example happen, if the configuration is invalid. Please look out for error messages from the OpenVPN driver.

- **Upload of authority's certificate failed**

  The certification authority's certificate file (*ca.crt*) could not be uploaded.

- **Upload of certificate failed**

  The certificate file (*cert.crt*) could not be uploaded.

- **Upload of private key failed**

  The private key file (*cert.key*) could not be uploaded.

- **Upload of user and password file failed**

  The user and password file (*userpass.txt*) could not be uploaded.

- **Upload of Diffie-Hellman file failed**

  The Diffie-Hellman file (*dh.pem*) could not be uploaded.

DHCP Server

The HMS Hub firmware features a DHCP server (Dynamic Host Configuration Protocol) which you can enable and configure via this form.

> **!** The DHCP server loses all information on allocated IP addresses on reboot. So after the Edge gateway device has restarted, all DHCP clients must be restarted, too.

In the first section you have to configure the maximum number of clients that will be serviced (max. 100, including static mappings), the leasing time, the server's own IP address and the netmask. The server will lease IP addresses from the particular network except for his own IP address. Additionally you can specify a gateway and name server IP address that will be distributed as well.

Static Leases

In this section you can configure up to 20 static leases. A static lease will assure that a client with a specific MAC address will be given a predefined IP address and that this IP address is not leased to another client. The MAC address of the client has to be noted without any delimiters (e. g. 003056A1DB30).

Event Log Messages

The following section describes the messages that the DHCP server may log to the Event Log. Messages are prefixed with the text *DHCP server:*.

- **Running**

  The DHCP server has been successfully started.

- **Added static lease IP=*<x>*, MAC=*<y>***

  A static lease has been added. The message also shows the respective IP and MAC address.

- **Discover message received, CI=<*x*>, MAC=<*y*>**

  A DHCP discover message was received from a client. The message also shows the client identifier (as a hex string) and the client's MAC address.

- **Request message received, CI=<*x*>, MAC=<*y*>**

  A DHCP request message was received from a client. The message also shows the client identifier (as a hex string) and the client's MAC address.

- **Leased IP=<*x*>, Leasing Time=<*y*>, Index=<*z*>**

  The given IP address has been leased to the client. The message also shows the leasing time in seconds and the internal leasing table index.

- **Release message received, CI=<*x*>, MAC=<*y*>**

  A DHCP release message was received from a client. The message also shows the client identifier (as a hex string) and the client's MAC address.

- **Release IP=<*x*>, Index=<*y*>**

  The given IP address has been released. The message also shows the internal leasing table index.

- **Network interface not configured**

  The network interface the DHCP server is to serve on does not have a valid IP configuration.

- **Failed to send response message, error <*x*>**

  A DHCP response message could not be sent. The message also shows the error code returned by the TCP/IP stack.

- **No more client addresses available**

  A client's request could be served, because all available addresses are already leased to or reserved for other clients.

- **Receive Error <*x*>**

  No DHCP message could be received. The message also shows the error code returned by the TCP/IP stack.

- **Received malformed message**

  A malformed DHCP message was received from a client.

- **No interface found for given IP address**

  The DHCP server could not find a network interface for the configured IP address at start-up.

- **Open socket failed**

  The DHCP server could not open a UDP socket at start-up.

- **Bind socket failed**

  The DHCP server could not bind its UDP socket to the designated port at start-up.

- **Only <*x*> clients possible due to netmask setting**

  Warning: The configured netmask allows less IP addresses than configured.

- **Failed to allocate client data structure**

  The DHCP client failed to allocate memory for its internal leasing table at start-up.

- **Adding static lease IP=<*x*> failed, IP address is in use**

  A static lease could not be added, because the respective IP address is already in use. The message also shows the IP address of the static lease that should be added.

- **Adding static lease IP=<*x*> failed, no free slot**

  A static lease could not be added, because there was no free space inside the internal leasing table. The message also shows the IP address of the static lease that should be added.

- **Adding static lease IP=<*x*> failed, wrong IP address**

  A static lease could not be added, because the respective IP address did not match the network address. The message also shows the IP address of the static lease that should be added.

4.2.3 Portal

Through this form you can configure various settings concerning the communication with the portal.

Enable	If this checkbox is checked, the Edge gateway device will attempt to communicate with the portal.
Host	Domain name of the portal host the Edge gateway device should connect to (*kolibri://project.**domain.tld**/ scope/group/point*).
Project	Name of the portal project to log in to (*kolibri://**project**.domain.tld/scope/group/point*). The project name must start and end with a letter or digit ([a-zA-Z0-9]) and it may contain up to 32 characters from the character class [a-zA-Z0-9-]. It is case-insensitive.
Scope	Scope the Edge gateway device logs in to on the project (*kolibri://project.domain.tld/**scope**/group/point*). The scope must start and end with a letter or digit ([a-zA-Z0-9]) and it may contain up to 32 letters, digits, underscores, periods or hyphens ([a-zA-Z0-9_.-]). The scope is case-sensitive.
User name	User name used to log in to the portal. The user name must start and end with a letter or digit ([a-zA-Z0-9]) and it may contain up to 32 letters, digits, underscores, periods or hyphens ([a-zA-Z0-9_.-]). The user name is case-insensitive.
Password	Password used to log in to the portal.
Timeout	General TCP communication timeout. This timeout determines how long the portal communication module will wait for replies from the portal before a request is considered failed and will possibly be repeated. The worse your network connectivity is, the longer should be the timeout. For a wired connection typically 10 to 20 seconds are a good choice. For a good cellular connection 60 seconds should be enough. For a bad cellular connection you may need to configure a timeout of several minutes. Also note that the higher the TCP timeout is, the longer it will take to notice that the connection is broken.
Keep-alive interval	When there is no payload data to be transmitted, a keep-alive handshake will be initiated in order to make sure that all infrastructure components between the Edge gateway device and the portal keep their information on the connection. The keep-alive interval determines the maximum number of seconds with no payload data traffic before the keep-alive handshake will be carried out. A smaller interval produces more data traffic, but a too great interval may result in connection loss. The required interval depends on your network infrastructure.
Keep-alive timeout	Number of seconds the peers should wait for each other's keep-alive messages before the connection is considered broken. The keep-alive timeout behaves similar to the general TCP timeout.

Watchdog

The portal watchdog can be used in use cases where the connection to the portal is the main feature of your application. If there is no connection to the portal for the given number of minutes, the whole device will be rebooted making sure that all hardware and software components are re-started.

Event Log Messages

The following section describes the messages that the portal communication module may log to the Event Log. Messages are prefixed with the text *Portal module:* .

- **Opening connection to <*host*>**

  The device is connecting to the portal at the indicated host address.

- **Connection established**

  The device has established a connection to the portal.

- **Connection closed**

  The connection to the portal has been closed.

- **Clock synchronised**

  The device has synchronised its clock with the portal.

- **Logging in**

  The device is logging in to the portal.

- **Synchronising clock**

  The device is synchronising its clock with the portal.

- **Initiating close sequence**

  The device closes the portal connection.

- **Connection closed by Broker**

  The portal has closed the connection.

- **Connection closed by Broker - Broker is going down**

  The portal has closed the connection, because the Broker is going down (e.g. for maintenance).

- **Connection closed by Broker - Protocol error**

  The portal has closed the connection, because it encountered a protocol error.

- **Connection closed by Broker - Keep-alive timeout**

  The portal has closed the connection, because the keep-alive mechanism timed out.

- **Connection closed by Broker - Another client logged in with same user name or to same scope**

  The portal has closed the connection, because another client logged in with the same user name or to the same scope. It is not allowed to log in with the same user name from several clients. And only one producer is allowed to log in to the same scope.

- **Connection closed by Broker - User was still logged in - Retry**

  The portal has closed the connection, because the same user was still logged in. Both the existing and the new session have been closed. The device will retry to connect to the portal.

- **Logged in**

  The device has logged in successfully to the portal.

- **Need to synchronise clock, but user has not set a clock update call-back function.**

  The device needs to synchronise its clock in order to communicate via the Kolibri protocol. This warning message means that the device is unable to synchronise the clock with the portal, because of an internal error. Please contact the manufacturer if you encounter this error message.

- **Close sequence timed out**

  A communication timeout occurred while closing the portal connection.

- **Ack with sequence number *<sequence number>* could not be matched**

  An Ack reply was received from the portal that could not be matched to any request.

- **Nak with sequence number *<sequence number>* could not be matched**

  An Nak reply was received from the portal that could not be matched to any request.

- **Point *<path>* not assigned to dataset: Point is inactive**

  The portal published the point with the indicated path. But the point was not assigned to a dataset on the device, because the point is inactive.

- **Point *<path>* not assigned to dataset: Data type does not match**

  The portal published the point with the indicated path. But the point was not assigned to a dataset on the device, because the data type does not match.

- **Commit command could not be matched**

  A commit command received from the portal could not be matched to a committed write command.

- **Transaction ID *<ID>* from cancel command could not be matched**

  A cancel command received from the portal could not be matched to a committed write command.

- **RPC response does not match a request**

  An RPC response received from the portal could not be matched to an RPC request.

- **RPC method *<method>* not found**

  An RPC method with the indicated name called by the portal was not found.

- **Could not enqueue value for point *<path>*: Send queue is full**

  A value for the point with the indicated path could not be added to the send queue, because the queue is full. This message will only be emitted for the first point in a set of points for that a value could not be enqueued.

- **HTTP upgrade error (*<error code>*)**

  The HTTP connection to the portal could not be upgraded to the Kolibri protocol. The indicated error code is an HTTP client error code.

- **Invalid character in challenge (*<hexadecimal character code>*)**

  The password challenge received from the portal contained the indicated invalid character.

- **Empty challenge**

  The password challenge received from the portal was empty.

- **Mandatory header field missing**

  A mandatory header field was missing in the HTTP response received from the portal at connection establishment.

- **Could not connect: Socket error (*<error code>*)**

  A socket error occurred while connecting to the portal. The indicated error code is an HTTP client error code.

- **Could not connect: Connect error (*<error code>*)**

  A connect error occurred while connecting to the portal. The indicated error code is an HTTP client error code.

- **Could not connect: TLS error (*<error code>*)**

  A TLS error occurred while connecting to the portal. The indicated error code is an HTTP client error code.

- **Could not connect: Could not resolve host name**

  The host name of the portal could not be resolved to an IP address.

- **Could not connect: Could not send HTTP request**

  An HTTP request could not be sent while trying to connect to the portal.

- **Could not connect: Could not receive HTTP response**

  An HTTP response could not be received while trying to connect to the portal.

- **Could not connect: Protocol error**

  An HTTP protocol error occurred while trying to connect to the portal.

- **Could not connect: Could not log in to proxy server.**

  The device failed to log in to the proxy server while trying to connect to the portal.

- **Emergency command timeout**

  A communication timeout occurred while sending an emergency command to the portal.

- **Could not send emergency command: Send error**

  An error occurred while sending an emergency command to the portal.

- **Keep alive timeout**

  The keep-alive handshake carried out with the portal timed out.

- **Invalid sequence number *<sequence number>***

  A packet with the indicated invalid sequence number was received from the portal.

- **Invalid opcode *<hexadecimal opcode>***

  A packet with the indicated invalid opcode was received from the portal.

- **Invalid WebSocket opcode *<hexadecimal opcode>***

  A packet with the indicated invalid WebSocket opcode was received from the portal.

- **Closing connection due to communication error (*<error code>*)**

  The connection to the portal is being closed, because the indicated communication error occurred.

- **Could not log in: Could not send Login command**

  The login command could not be sent while logging in to the portal.

- **Could not log in: Timeout**

  A communication timeout occurred while logging in to the portal.

- **Could not synchronise clock: Send error**

  The get time command could not be sent while synchronising the device's clock with the portal.

- **Could not synchronise clock: Timeout**

  A communication timeout occurred while synchronising the device's clock with the portal.

- **Could not get publish hash: Send error**

  The get hash command could not be sent while synchronising published nodes with the portal.

- **Could not get publish hash: Timeout**

  A communication timeout occurred while synchronising published nodes with the portal.

- **Could not enable publishing of all points: Send error**

  The enable publish command could not be sent while synchronising published nodes with the portal.

- **Could not enable publishing of all points: Timeout**

  A communication timeout occurred while synchronising published nodes with the portal.

- **Could not send write command**

  The write command could not be sent to the portal.

- **Write/commit command timeout for point *<path>***

  A communication timeout occurred while transmitting a value for the indicated point to the portal.

- **Could not add value for point *<path>* to write command (*<error code>*)**

  A value for the indicated point could not be added to a write command. The indicated error code is a Kolibri library error code.

- **Could not send commit command for write command with sequence number *<sequence number>* (*<error code>*)**

  The commit command for the committed write command with the indicated sequence number could not be sent to the portal. The indicated error code is a Kolibri library error code.

- **Could not send create node command**

  The create node command could not be sent to the portal in order to create a new node.

- **Could not send modify node command**

  The modify node command could not be sent to the portal in order to modify a node's properties.

- **Could not send delete node command**

  The delete node command could not be sent to the portal in order to delete a node.

- **Could not send get node properties command**

  The get node properties command could not be sent to the portal in order get a node's properties.

- **Could not send enable publish command**

  The enable publish command could not be sent to the portal in order to get a list of all nodes.

- **Waiting for publish to complete timed out**

  A communication timeout occurred while getting a list of all nodes from the portal.

- **Node manipulation command timed out**

  A communication error occurred while creating, modifying, deleting a node, getting a node's properties or getting a list of all nodes.

- **Could not send keep-alive pong (*<error code>*)**

  The pong command could not be sent while carrying out the keep-alive handshake with the portal.

- **Mandatory RPC field missing**

  A mandatory field was missing in an RPC request or response received from the portal.

- **RPC server not initialised**

  An RPC request was received from the portal, but the device is not an RPC server.

- **Could not parse RPC: Out of memory**

  An out-of-memory condition occurred while parsing an RPC request or response received from the portal.

- **Could not parse RPC: *<error message>***

  The indicated error occurred while parsing an RPC request or response received from the portal.

- **Could not process RPC: Queue is full**

  An RPC request or response received from the portal could not be processed, because the RPC queue was full.

- **Could not process RPC: Receive error**

  A receive error occurred while receiving an RPC request or response from the portal.

- **Could not process RPC: Unexpected error *<error code>***

  An unexpected error occurred while processing an RPC request or response from the portal. The indicated error code is a Kolibri library error code.

- **Ack with sequence number *<sequence number>* contained additional payload data**

  An Ack reply with the indicated sequence number received from the portal contained more payload data than expected.

- **Could not log in: Access denied**

  Access was refused while logging in to the portal. You should check user name and password.

- **Could not log in: Unexpected error (*<error code>*)**

  An unexpected error occurred while logging in to the portal. The indicated error code is a Kolibri library error code.

- **Could not get publish hash: Unexpected error (*<error code>*)**

  An unexpected error occurred while synchronising published nodes with the portal.

- **Could not enable publishing of all active points: Unexpected error (*<error code>*)**

  An unexpected error occurred while synchronising published nodes with the portal.

- **Node manipulation command failed (*<error code>*)**

  An error occurred while creating, modifying, deleting a node, getting a node's properties or getting a list of all nodes. The indicated error code is a Kolibri library error code.

- **Write command with sequence number *<sequence number>* failed (*<error code>*, *<point index>*)**

  The write command with the indicated sequence number was not accepted by the portal. The indicated error code is a Kolibri library error code. The indicated point index announces the point that caused the problem.

- **Write command with sequence number *<sequence number>* failed (*<error code>*)**

  The write command with the indicated sequence number was not accepted by the portal. The indicated error code is a Kolibri library error code.

- **Nak with sequence number *<sequence number>* contained invalid payload data (*<error code>*)**

  A Nak reply with the indicated sequence number received from the portal contained invalid payload data. The indicated error code is a Kolibri library error code.

- **Could not synchronise clock: Receive error (*<error code>*)**

  A receive error occurred while synchronising the device's clock with the portal. The indicated error code is a Kolibri library error code.

- **Could not synchronise clock: Get time response contained invalid payload data**

  A get time response command received from the portal contained invalid payload data.

- **Could not synchronise clock: Unexpected error (*<error code>*)**

  An unexpected error occurred while synchronising the device's clock with the portal. The indicated error code is a Kolibri library error code.

- **Could not get publish hash: Receive error**

  A receive error occurred while synchronising published nodes with the portal.

- **Could not get publish hash: Out of memory**

  An out-of-memory condition occurred while synchronising published nodes with the portal.

- **Could not get publish hash: Unexpected error (*<error code>*)**

  An unexpected error occurred while synchronising published nodes with the portal. The indicated error code is a Kolibri library error code.

- **Could not publish group: Invalid index *<group index>***

  The portal published a group with the indicated invalid index.

- **Could not publish group: Re-publish *<known path>* with path *<new path>***

  The portal published a group with the indicated new path that had already been published before with the indicated known path.

- **Could not publish group: Initial publish without path**

  The portal published a group, but did not include the group's path.

- **Could not publish point: Invalid index *<point index>***

  The portal published a point with the indicated invalid index.

- **Could not publish point: Invalid trigger mode *<trigger mode>***

  The portal published a point with the indicated invalid trigger mode.

- **Could not publish point: Invalid trigger mode *<trigger mode>* for data type *<data type>***

  The portal published a point with the indicated trigger mode which is invalid for the point's indicated data type.

- **Could not publish point: Invalid QoS %u**

  The portal published a point with the indicated invalid quality of service level.

- **Could not publish point: Re-publish *<known path>* with path *<new path>***

  The portal published a point with the indicated new path that had already been published before with the indicated known path.

- **Could not publish point: Initial publish for *<path>* (uncertain) without path**

  The portal published a point without a path.

- **Could not publish point: Initial publish without path**

  The portal published a point without a path.

- **Could not process publish command: Invalid library state**

  The portal published points or groups, but the Kolibri library was not ready to process the publish command.

- **Publishing of groups not enabled**

  The portal published groups although publishing of groups was disabled.

- **Could not publish point: Invalid data type <*data type*>**

  The portal published a point with the indicated invalid data type.

- **Could not process publish command: Receive error**

  An error occurred while receiving a publish command from the portal.

- **Could not process publish command: Send error**

  A send error occurred while processing a publish command from the portal.

- **Could not process publish command: Unexpected error (<*error code*>)**

  An unexpected error occurred while processing a publish command from the portal. The indicated error code is a Kolibri library error code.

- **Could not send Nak for publish command**

  An error occurred while sending a Nak reply for a publish command received by the portal.

- **Could not process un-publish command: Invalid library state**

  The portal un-published points or groups, but the Kolibri library was not ready to process the un-publish command.

- **Could not un-publish <*path*>: Invalid index**

  The portal un-published a point or group with the indicated invalid index.

- **Could not process un-publish command: Receive error**

  An error occurred while receiving an un-publish command from the portal.

- **Could not process un-publish command: Send error**

  A send error occurred while processing an un-publish command from the portal.

- **Could not process un-publish command: Unexpected error (<*error code*>)**

  An unexpected error occurred while processing an un-publish command from the portal.

- **Could not send Nak for un-publish command**

  An error occurred while sending a Nak reply for an un-publish command received by the portal.

- **Could not process (committed) write command: Library not ready**

  The portal wrote a points value, but the Kolibri library was not ready to process the (committed) write command.

- **Could not process (committed) write command: Receive error**

  An error occurred while receiving a (committed) write command from the portal.

- **Could not process (committed) write command: Invalid node index <*point index*>**

  The portal wrote a point's value with the indicated invalid point index.

- **Could not process (committed) write command: Invalid data type *<data type>***

    The portal wrote a point's value with the indicated invalid data type.

- **Could not process (committed) write command: Receive queue is full**

    A (committed) write command received from the portal could not be processed, because the receive queue is full.

- **Could not process (committed) write command: Protocol error**

    A protocol error occurred while processing a (committed) write command received from the portal.

- **Could not process (committed) write command: Send error**

    An error occurred while processing a (committed) write command received from the portal.

- **Could not send Nak for write command**

    An error occurred while sending a Nak reply for a write command received by the portal.

- **Could not process commit command: Library not ready**

    The portal wrote a point's value, but the Kolibri library was not ready to process the commit command.

- **Could not process commit command: Receive error**

    An error occurred while receiving a commit command from the portal.

- **Could not process commit command: Protocol error**

    A protocol error occurred while processing a commit command received from the portal.

- **Could not process commit command: Send error**

    A send error occurred while processing a commit command received from the portal.

- **Could not process cancel command: Library not ready**

    The portal cancelled writing a point's value, but the Kolibri library was not ready to process the cancel command.

- **Could not process cancel command: Receive error**

    An error occurred while receiving a cancel command from the portal.

- **Could not process cancel command: Protocol error**

    A protocol error occurred while processing a cancel command received from the portal.

- **Could not process cancel command: Send error**

    A send error occurred while processing a cancel command received from the portal.

- **Could not process get node properties response: Invalid node type *<node type>***

    The portal returned the indicated invalid node type with a get node properties response.

- **Could not process get node properties response: Invalid index *<index>***

    The portal returned the indicated invalid index with a get node properties response.

- **Could not process get node properties response: Invalid path *<path>***

    The portal returned the indicated invalid path with a get node properties response.

- **Could not process get node properties response: Invalid flags *<hexadecimal flags>***

    The portal returned the indicated invalid flags with a get node properties response.

- **Could not process get node properties response: Invalid data type *<data type>***

  The portal returned the indicated invalid data type with a get node properties response.

- **Could not process get node properties response: Invalid trigger mode *<trigger mode>***

  The portal returned the indicated invalid trigger mode with a get node properties response.

- **Could not process get node properties response: Invalid QoS *<quality of service>***

  The portal returned the indicated invalid quality of service level with a get node properties response.

- **Could not process get node properties response: Invalid frame length**

  A protocol error occurred while processing a get node properties response received from the portal.

- **Could not process get node properties response: Invalid library state**

  The Kolibri library was not ready to process the get node properties response received from the portal.

- **Could not process get node properties response: Receive error**

  An error occurred while receiving a get node properties response from the portal.

- **Could not process get node properties response: Unexpected error (*<error code>*)**

  An unexpected error occurred while processing a get node properties response received from the portal.

- **Could not send RPC request: Send error**

  An error occurred while sending an RPC request to the portal.

- **Could not send RPC request: Unexpected error (*<error code>*)**

  An unexpected error occurred while sending an RPC request to the portal. The indicated error code is a Kolibri library error code.

- **RPC request for method *<method>* timed out**

  A communication timeout occurred while calling the indicated RPC method.

- **RPC response contains neither a result nor an error object**

  An RPC response received from the portal contained neither a result nor an error object.

- **Mandatory RPC request field missing**

  A mandatory field was missing in an RPC request received from the portal.

- **Could not send RPC response: Send error**

  A send error occurred while sending an RPC response to the portal.

- **Exception in RPC method *<error message>***

  An exception with the indicated error message occurred while executing and RPC method.

- **Could not execute kolibri.reboot() RPC: Invalid parameter or reboot not allowed**

  The kolibri.reboot() RPC could not be executed. Either a parameter was invalid or the reboot is not allowed.

- **Could not execute kolibri.getStats() RPC: Error getting statistics (*<error code>*)**

  Statistics could not be gotten while executing the kolibri.getStats() RPC. The indicated error code is a Kolibri library error code.

- **Could not enqueue value for point <*path*>: Out of memory**

  An out-of-memory condition occurred while putting a value for the point with the indicated path to the send queue. This message will only be emitted for the first point in a set of points for that a value could not be enqueued.

- **Could not enqueue value for point <*path*>: Unexpected error (<*error code*>)**

  An unexpected error occurred while putting a value for the point with the indicated path to the send queue. The indicated error code is a Kolibri library error code. This message will only be emitted for the first point in a set of points for that a value could not be enqueued.

- **Could not connect: Out of memory**

  An out-of-memory condition occurred while connecting to the portal.

- **Could not connect: Unknown error (<*error code*>)**

  An unexpected error occurred while connecting to the portal. The indicated error code is a Kolibri library error code.

- **Could not send emergency command: Unexpected error (<*error code*>)**

  An unexpected error occurred while sending an emergency command to the portal. The indicated error code is a Kolibri library error code.

- **Could not log in: Unexpected error (<*error code*>)**

  An unexpected error occurred while logging in to the portal. The indicated error code is a Kolibri library error code.

- **Received Ack for node manipulation command in wrong state**

  The portal sent an Ack reply for creating, modifying, deleting a node, getting a node's properties or getting a list of all nodes. But none of these commands was pending.

- **Received Nak for node manipulation command in wrong state**

  The portal sent an Nak reply for creating, modifying, deleting a node, getting a node's properties or getting a list of all nodes. But none of these commands was pending.

- **Could not process publish command: Out of memory**

  An out-of-memory condition occurred while processing a publish command received from the portal.

- **Could not process un-publish command: Out of memory**

  An out-of-memory condition occurred while processing an un-publish command received from the portal.

- **Could not process (committed) write command: Out of memory**

  An out-of-memory condition occurred while processing a (committed) write command received from the portal.

- **Could not process (committed) write command: Unexpected error (<*error code*>)**

  An unexpected error occurred while processing a (committed) write command received from the portal.

- **Could not process commit command: Unexpected error (<*error code*>)**

  An unexpected error occurred while processing a commit command received from the portal.

- **Could not process cancel command: Unexpected error (<*error code*>)**

  An unexpected error occurred while processing a cancel command received from the portal.

- **Could not process get node properties response: Out of memory**

  An out-of-memory condition occurred while processing a get node properties response received from the portal.

- **Could not send RPC request: Out of memory**

  An out-of-memory condition occurred while sending an RPC request to the portal.

- **Could not build RPC error object for unknown method <*method*>: Out of memory**

  An out-of-memory condition occurred while sending an error response for the indicated unknown RPC method to the portal.

- **Could not send RPC response: Out of memory**

  An out-of-memory condition occurred while sending a RPC response to the portal.

- **Could not send RPC response: Unexpected error (<*error code*>)**

  An unexpected error occurred while sending a RPC response to the portal.

- **Could not execute kolibri.getRpcInfo() RPC: Out of memory**

  An out-of-memory condition occurred while executing the kolibri.getRpcInfo() RPC method.

- **Could not execute kolibri.reboot() RPC: Out of memory**

  An out-of-memory condition occurred while executing the kolibri.reboot() RPC method.

- **Could not execute kolibri.getStats() RPC: Out of memory**

  An out-of-memory condition occurred while executing the kolibri.getStats() RPC method.

Portal

The HMS Hub firmware publishes the following information on the portal:

- Inputs and outputs (see *inputs* and *outputs* groups)

- Variables (see *variables* group)

- General device information (e.g. MAC address, see *device-info/system* group)

- General firmware information (e.g. firmware version, see *device-info/system* group)

I/Os are read-only from the portal's point of view. This is because inputs are read-only in general and outputs are permanently written by the PLC. If you would write to an output from the portal, the value would be overwritten by the PLC immediately. Variables can also be written from the portal. Make sure that each variable is only written either from the PLC or from the portal. It is possible to write an output via a variable from the portal. The variable must be written from the portal. The PLC must read the variable and write its value to the output.

I/O Portal Synchronization

At start-up the structure of the Edge gateway devices I/Os will be synchronised with the portal. For devices with a bus system the I/O structure may change - for example if another I/O device is attached to the bus. The HMS Hub firmware will create portal nodes for new I/Os and delete portal nodes for I/Os that are no longer present. For existing nodes it will also overwrite the description as well as the properties for scaling and formatting provided the corresponding I/O driver provides this information. The overwriting of node properties can be deactivated through the I/O settings. See for more information.

Round-trip Time

Data round-trip time is indicative of the quality of the communication link between the device and the Hub.

Event Log Messages

The following section describes the messages that the portal communication module may log to the Event Log during I/O Portal.

- **Starting I/O synchronisation with portal**

  The synchronisation of the device's I/O structure with the portal is now being started.

- **I/Os have been synchronised with portal**

  The device's I/O structure has been successfully synchronised with the portal.

- **Could not get list of portal nodes. (Error code *error code*)**

  The I/O portal synchronisation module failed to get the list of all portal nodes. The indicated error code is a Kolibri library error code.

- **Could not get node properties. (Node path *node path*, error code *error code*)**

  The I/O portal synchronisation module failed to get the properties of the portal node with the indicated path. The indicated error code is a Kolibri library error code.

- **Could not delete node. (Node path *node path*, error code *error code*)**

  The I/O portal synchronisation module failed to delete the portal node with the indicated path. The indicated error code is a Kolibri library error code.

- **Could not create node. (Node path *node path*, error code *error code*)**

  The I/O portal synchronisation module failed to create the portal node with the indicated path. The indicated error code is a Kolibri library error code.

- **Could not modify node properties. (Node path *node path*, error code *error code*)**

  The I/O portal synchronisation module failed to modify the properties of the portal node with the indicated path. The indicated error code is a Kolibri library error code.

- **Could not generate node path. (position *position*, error code *error code*)**

  The I/O portal synchronisation module failed to generate the path of a portal node. This should not happen.

- **Could not synchronise I/Os with portal**

  The device's I/O structure could not be synchronised with the portal. Further information about the reason of the problem is not available.

4.2.4   PLC

Variables

Through this form you can configure the Hub Variables.

At the top there is a list with the defined variables. It shows for every variable the name, the data type, the portal path and whether it is retentive. You can add, edit, delete or clone variables through the buttons below the list.

If the Edge gateway device is connected to the portal, the variables will be synchronised with the nodes underneath the *variables* group on the portal at startup.

When adding or editing a variable, a dialog will show up. Here the following parameters can be configured:

Name	The variable name is shown in the Diagram Editor and used as description of the corresponding portal node.
Data type	Variable's data type

Portal path	Path to the portal node which represents the variable. Only the last part of the path is defined here. All variable nodes are created underneath the *Variables* group
Retentive	If checked, the variable will retain its value on power fail. Don't confuse this with the property of the same name that can be set for portal nodes. This one uses the hardware's power fail feature to retain the data.
	! Note that not all Edge gateway devices are equipped with a power fail circuit. If your device is not equipped with a power fail circuit, a warning message will be shown in the event log.

Limits

A maximum of 1024 variables can be configured. All retentive variables together may use up to 4096 bytes and all non-retentive variables together up to 1048576 bytes.

Event Log Messages

The following section describes the messages that the Variables web service may log to the Event Log. All Messages are prefixed with the text *Variables Web Service:*.

- **Starting Variables synchronization with portal**

  The synchronization of the variables with the portal nodes representing them has been started.

- **Variables have been synchronized with portal**

  The variables have been synchronized with the portal nodes representing them.

- **Could not get list of portal nodes. (Error code *error code*)**

  The Variable portal synchronization module failed to get the list of all portal nodes. The indicated error code is a Kolibri library error code.

- **Could not get node properties. (Node path *node path*, error code *error code*)**

  The Variable portal synchronization module failed to get the properties of the portal node with the indicated portal path. The indicated error code is a Kolibri library error code.

- **Could not delete node. (Node path *node path*, error code *error code*)**

  The Variable portal synchronization module failed to delete the portal node with the indicated portal path. The indicated error code is a Kolibri library error code.

- **Could not create node. (Node path *node path*, error code *error code*)**

  The Variable portal synchronization module failed to create the portal node with the indicated portal path. The indicated error code is a Kolibri library error code.

- **Could not modify node properties. (Node path *node path*, error code *error code*)**

  The Variable portal synchronization module failed to modify the properties of the portal node with the indicated portal path. The indicated error code is a Kolibri library error code.

- **Could not synchronize Variables with portal**

  The device's Variable structure could not be synchronized with the portal. Further information about the reason of the problem is not available.

Time Switches

Through this form you can configure the HMS Hub *Time Switches*. The Time Switches functionality is based on the Unix Cron utility. Notable differences are the two additional non-standard fields *Years* and *Seconds* and that the Trigger Type *List* is restricted to five values.

At the top there is a list of time switches. The list shows a name and the trigger conditions for each time switch. Entries can be edited or reset through the buttons below the list.

When editing a value, a dialog will show up. Here the following parameters can be configured:

Time Switch Name	Name of the time switch unit
Year(s)	The year(s) at which the time switch will trigger can be set as a single value, wildcard, list or range.
Month(s)	The month(s) at which the time switch will trigger can be set as a single value (1–12), wildcard, list or range.
Day(s) of the Week	The day(s) of the week at which the time switch will trigger can be set as pain value (0–6, 0 for Sunday), wildcard, list or range.
Day(s) of the Month	The day(s) of the month at which the time switch will trigger can be set as a single value (1–31), wildcard, list or range.
Hour(s)	The hour(s) at which the time switch will trigger can be set as a single value (0–23), wildcard, list or range.
Minute(s)	The minute(s) at which the time switch will trigger can be set as a single value (0–59), wildcard, list or range.
Second(s)	The second(s) at which the time switch will trigger can be set as a single value (0–59), wildcard, list or range.

Trigger Types

Value	Use this type to trigger the time switch each time the corresponding time component is equal to the configured value. For example set all other fields except the *Second(s)* field to trigger type *Wildcard*, and set the *Second(s)* field to trigger type *Value* and value *5*. If the corresponding unit is used in the program it will set its output port to *1* every minute in second *5*.
Wildcard	Use this trigger type when the time switch should ignore the corresponding time component. If for example a time switch should trigger once every hour, you must set all fields except for *Minute(s)* to *Wildcard*.
List	for this trigger type up to five comma-separated values (see *Value*) can be given.
Range	Use this type to trigger the time switch each time the corresponding time component lies within the configured range. If for example the range *2009-2011* is configured for the *Year(s)* field, the time switch can trigger in the years 2009, 2010 and 2011 (depending on what is configured for the other fields).

Examples

Expected Schedule	Cron expression	TimeSwiches equivalent
Run once a year the first of January at 00:00:00.	0 0 0 1 1 *	Will fire every time sec = 0, min = 0, hours = 0, month = 1, year = *, month Day = 1, week Day = *.
Run once the first of each month at 07:30:00.	30 7 1 * *	Will fire every time sec = 0, min = 30, hours = 7, month = *, year = *, month Day = 1, week Day = *.
Run each Sunday at 15:15:00.	15 15 * * 0	Will fire every time sec = 0, min = 15, hours = 15, month = *, year = *, month Day = *, week Day = 0.
Run each day at 23:00:00.	23 * * * *	Will fire every time sec = 0, min = 0, hours = 23, month = *, year = *, month Day = *, week Day = *.

Schedules

Since the time switches trigger only once, two of them are needed in order to switch something on and off based on a schedule. One of them implements the *on function* and the other one the *off function*. The output ports of the time switches are connected to one input port of an *RS Flipflop* respectively as shown in image 1. In the example something should be switched on every Sunday, Tuesday, Thursday and Saturday from 7 am to 1 pm.

ON_Switch configuration	Will fire every time sec = 0, min = 0, hours = 7, month = *, year = *, month Day = *, week Day = 0,2,4,6.
OFF_Switch configuration	Will fire every time sec = 0, min = 0, hours = 13, month = *, year = *, month Day = *, week Day = 0,2,4,6.
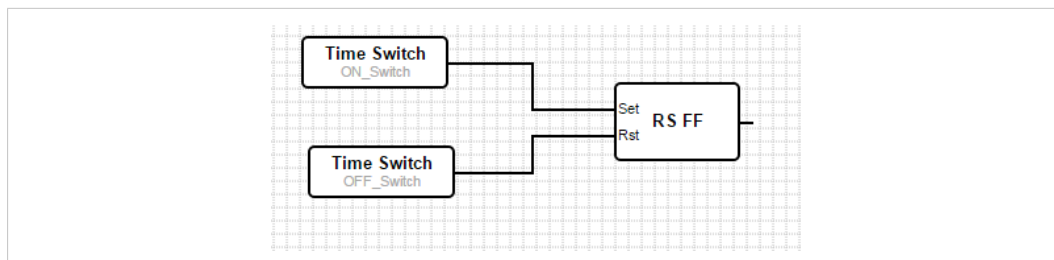
Fig. 16   Scheduling example

Limits

You can define up to eight time switches. Time Switches with the same name are not allowed.

> ❗ Note that the time switches use the local system time.

Messages

Your Edge gateway is capable of sending and receiving messages. Messages can be transmitted via various channels. The available channels depend on the concrete configuration of your device.

Send Messages

In this section you can configure up to 25 messages that can be sent. For each message you can configure the following parameters:

Active	Through this check box the message can be activated or deactivated respectively.
Channel	Channel through which the message should be sent
Recipient/Topic	For short messages (SMS): Recipient's phone number composed of an optional leading + and up to 30 numerical digits (e.g. +491701234567) For MQTT: Topic to which the message should be sent
Text	Message text For short messages (SMS): Only up to 160 ASCII characters (Latin characters, numbers, punctuation characters, no umlauts etc.) are allowed. If, after replacing all references, the message text is longer, it will be cut. A counter that is shown below the message text input field will show you the number of plain characters and references in your message text. References You can add references to variables, inputs or outputs to the text. Before a message is sent, the references will be replaced by the actual values. The references start and end with a percentage sign. Between the percentage signs there must be either the name of a variable or the word *input* or *output* followed by the respective interface, device, value and optionally array index, separated by commas. For example *%portalvar3%* references the variable *portalvar3*. *%input0,1,2%* references the input at interface 0, device 1, value 2. Invalid references will not be replaced. You can also - much more conveniently - insert references via two menu buttons that will show up in the message editing dialog. To insert a plain percentage sign into a message, type two consecutive percentage signs.

Receive Messages

In this section you can configure up to 25 messages that can be received. If a message with the defined text is received from the respective Sender/Topic, the corresponding receive message unit will be triggered in your diagram. Note that receive messages will not be checked if the corresponding receive message unit is not used inside your diagram. For each message you can configure the following parameters:

Active	Through this check box the message can be activated or deactivated respectively.
Channel	Channel through which the message should be received

Sender/Topic	For short messages (SMS): Sender's phone number composed of an optional leading + and up to 30 numerical digits (e.g. +491701234567); note that the phone number must match exactly. Typically phone numbers are transmitted in international format (e.g. +491701234567). So you should enter the phone number in this format. For MQTT: Topic from which the message should be received; wildcards (+ and #) can be used.
Text	Message text For short messages (SMS): Only up to 160 ASCII characters (Latin characters, numbers, punctuation characters, no umlauts etc.) are allowed. If, after replacing all references, the message text is longer, it will be cut. A counter that is shown below the message text input field will show you the number of plain characters and references in your message text. References Again you can add references to the expected text. For receive messages only variable references are allowed. If your predefined text contains a variable reference and the actual received text contains a corresponding value at the same place, the variable will be set to this value. For example if you define the text "Set %portalvar1%!" and receive a message with the text "Set 5!", the variable *portalvar1* will be set to the value 5. Received messages must exactly match the predefined text. Even the character cases must match.

> **!** Note that references may become invalid or the I/O or variable particularly referenced may change, if you change the configuration of your Edge gateway device. The device will not notice this. So if you change the configuration, re-check the references.

MQTT

If you want to send and receive messages via MQTT you can configure the connection to the MQTT broker here.

Enable	Through this check box the MQTT client is activated.
MQTT version	Here you can chose the protocol version that should be used for the connection.
Use TLS	Through this check box you can activate TLS. > **!** If you don't activate TLS, all your data including user name and password will be transmitted as clear text.
Host	Host name of the MQTT broker
Port	Port number (by default 1883 without TLS and 8883 with TLS)
Client ID	The client ID identifies the client on the broker.
Client certificate chain file	Here the client certificate chain file in PEM format can be uploaded. It should contain the client certificate and the certificate of the certification authority (CA). Optionally it can also contain the private key of the client certificate.
Client certificate's private key file	Here the client certificate\'s private key file in PEM format can be uploaded. Optionally the private key can also be contained in the client certificate chain file.
User name	User name for the authentication
Password	Password for the authentication
QOS	The QOS (Quality of Service) determines how messages are transmitted: 0 — Every message is transmitted at most once. If it does not reach the communication peer for any reason, it is <u>not</u> repeated. 1 — Every message is transmitted at least once. If it does not reach the communication peer for any reason, it is repeated. It is possible that a message arrives twice. 2 — Every message is transmitted exactly once. If it does not reach the communication peer for any reason, it is repeated. A two-phase commit protocol assures that no message arrives twice.
Connect timeout	Timeout for connection establishment in seconds
Keep-alive interval:	To detect broken connections so-called keep-alive packets will be transmitted on a regular basis - as long as there is no payload data. Here the interval in seconds can be configured.
Last will topic	Topic to which the last will message will be sent
Last will message	Here the so-called last will message can be defined. The Broker will send this message in the name of the client, if the connection breaks unexpectedly.

Connect to a Microsoft Azure IoT Hub via MQTT

Via the MQTT protocol you can also connect your Edge gateway device to a Microsoft Azure IoT Hub and send device-to-cloud messages as well as receive cloud-to-device messages. An appropriate description on how to use the Azure IoT Hub with MQTT can be found here.

Device twin properties and direct methods are not supported.

Connect to Amazon AWS IoT via MQTT

Via the MQTT protocol you can also connect your Edge gateway device to Amazon AWS IoT and send and receive messages. An appropriate description on how to use AWS IoT with MQTT can be found here.

Event Log Messages

The following section describes the messages that the message manager may log to the Event Log. All messages are prefixed with the text *Message manager:*.

- **Sent message *<x>***

  Message number *x* has been successfully sent.

- **Message *<x>* was received**

  Message number *x* has been successfully received and decoded (i.e. values of referenced variables were read).

- **Opening connection to MQTT broker**

  The MQTT client is about to open a connection to the MQTT broker.

- **Connection to MQTT broker established**

  The MQTT client has successfully established a connection to the MQTT broker.

- **Sending message *<x>* failed**

  Sending message number *x* failed.

- **A message received through the *<channel>* channel from *<sender>* was discarded, because it did not match any template. Message text: "*<text>*"**

  A message has been received that did not match any of the configured receive messages. Thus the message has been discarded. Channel, Sender/Topic and message text are provided.

- **Out of memory**

  Memory allocation for a received message failed.

- **Could not connect to MQTT broker (*<error code>*)**

  The MQTT client could not connect to the MQTT broker. An error code from the MQTT Library or from the following list provides more information:

1	The MQTT broker does not support the MQTT protocol version requested by the MQTT client.
2	The MQTT client ID is invalid.
3	The network connection has established, but the MQTT service is unavailable.
4	The data in the user name or password is malformed.
5	The user is not authorised to connect to the MQTT broker.

- **Could not subscribe to MQTT topic "*<topic>*" (*<error code>*)**

  The MQTT client could not subscribe the given topic. An error code from the MQTT Library or from the following list provides more information:

128	The MQTT broker refused the subscription.

- **Closing connection to MQTT broker due to communication error (*<error code>*)**

  The connection to the MQTT broker is being closed due to a communication error. An error code from the [MQTT Library](#) provides more information.

Error codes of the MQTT library

-4	The maximum number of messages, that are on their way between MQTT client and broker, has been reached.
-3	There is currently no connection to the MQTT broker.
-1	Socket error. The server does not answer.

4.2.5   I/Os

Through this form you can configure general settings concerning the I/Os of the device.

Portal

Don't overwrite Format and Scaling properties	Some I/O drivers provide format and scaling properties that will be written to the corresponding portal nodes at startup during the Portal synchronization process. If this option is set, these properties will not be written any more. This allows custom format and scaling properties to be set at the portal.
Don't overwrite Description property	The I/O drivers provide a default description property that will be written to the corresponding portal nodes at startup during the Portal synchronization process. If this option is set, the description property will not be written any more. This allows a custom description property to be set at the portal.
Don't overwrite Trigger properties	The SGIM drivers provide trigger properties that will be written to the corresponding portal nodes at startup during the Portal synchronization process. If this option is set, the trigger property will not be written any more. This allows a custom trigger property to be set at the portal.
Don't overwrite QoS properties	The SGIM drivers provide QoS properties that will be written to the corresponding portal nodes at startup during the Portal synchronization process. If this option is set, the QoS property will not be written any more. This allows a custom QoS property to be set at the portal.

Local I/O

Your Edge gateway device features the following digital and/or analog inputs and outputs that are part of the device (local).

- 4 digital inputs (named *Input 0 - Input 3*)

- Power supply status (named *Power Supply*)

- 4 digital outputs (named *Output 0 - Output 3*).

- 2 S0 inputs. For the S0 inputs you can read as well the current state (named *Input 0* and *Input 1*) as the number of pulses that have been counted on the input (named *Counter 0* and *Counter 1*).

- Power supply status (named *Power Supply*)

- 2 digital outputs (named *Output 0* and *Output 1*)

- 6 digital inputs (named *Input 0 - Input 5*)

- 1 analog input (named *Analog Input*) It delivers a value between 0 and 255, representing a voltage between GND and VCC.

- 8 digital outputs (named *Output 0 - Output 7*)

- 1 PWM (pulse width modulation) output The PWM output can be enabled via the output named *PWM On*. The PWM mode can be switched between 16-bit mode (value 0, frequency 122 Hz) and 8-bit mode (value 1, frequency 1 kHz) via the output named *PWM Mode*. The duty cycle can be controlled via the output named *PWM Duty Cycle*. In 16-bit mode valid values range from 0 (0 %) to 65535 (100 %). In 8-bit mode valid values range from 5 (5 %) to 95 (95 %).

The Device also features an M-Bus interface.

- 8 digital inputs (named *Input 0 - Input 7*)

- 8 analog inputs (named *Analog Input 0 - Analog Input 7*) Each analog input supports two modes that can be configured through the I/O configuration page:

  – The *voltage* mode which delivers a value between 0 and 10.100 mV (±100 mV), representing a voltage between GND and VCC.

  – The *current* mode which delivers a value between 0 and 25.000 µA (±100 µA), representing the current between GND and VCC.

- Power supply status (named *Power Supply*)

- 8 digital outputs (named *Output 0 - Output 7*)

- 4 digital inputs (named *Input 0 - Input 3*)

- 4 analog inputs (named *Analog Input 0 - Analog Input 3*) Each analog input supports two modes that can be configured through the I/O configuration page:

  – The *voltage* mode which delivers a value between 0 and 10.100 mV (±100 mV), representing a voltage between GND and VCC.

  – The *current* mode which delivers a value between 0 and 25.000 µA (±100 µA), representing the current between GND and VCC.

- 2 RTD sensor inputs (named *Temp Sensor 0 - Temp Sensor 1*) Each RTD (*resistance temperature detector*) sensor input supports three modes that can be configured through the I/O configuration page. This configuration modes are used to translate the RTD generated input value to Celsius based on the sensor type. Values are measured in m°C. The driver supports the following sensors:

  – *PT100*: +850 °C – -200 °C

  – *PT500*: +850 °C – -200 °C

  – *PT1000*: +500 °C – -80 °C

- Power supply status (named *Power Supply*)

- 4 digital outputs (named *Output 0 - Output 3*)

- 2 analog outputs (named *Analog Output 0 - Analog Output 1*) Each analog output can deliver a value between 0 and 10.560 mV (±100 mV) based on the input value.
  Using *1500* as input the output will generate an electric potential difference of 1.5 V or 1500 mV with the analog ground port.

- 4 digital inputs (named *Input 0 - Input 3*)

- 4 analog inputs (named *Analog Input 0 - Analog Input 3*) Each analog input supports two modes that can be configured through the I/O configuration page:

  - The *Temperature* mode, Resolution 0.01 °C, Accuracy 0.25%.

    - *PT100*: +850 °C – -200 °C

    - *PT500*: +850 °C – -200 °C

    - *PT1000*: +150 °C – -50 °C

  - The *Resistance* mode which delivers a value between 0 and 5000 ohm, Resolution 0.1 ohm, Accuracy 0.25%.

  Analog 3 and Analog 4 further support the two following modes:

  - The *voltage* mode which delivers a value between 0 and 10V, Resolution 0.001V, Accuracy 0.25%.

  - The *current* mode which delivers a value between 0 and 20µA, Resolution 0.001µA, Accuracy 1%.

  In order to use Analog 3 and Analog 4 the following jumpers must also be correctly set:

  - *Temperature* AI3: J1=OFF, J2=ON; AI4: J3=OFF, J4=ON;

  - *Resistance* AI3: J1=OFF, J2=ON; AI4: J3=OFF, J4=ON;

  - *voltage* AI3: J1=ON, J2=OFF; AI4: J3=ON, J4=OFF;

  - *current* AI3: J1=OFF, J2=OFF; AI4: J3=OFF, J4=OFF; and plug in an external resistor of 125 Ohm between terminals AIx and AIC.
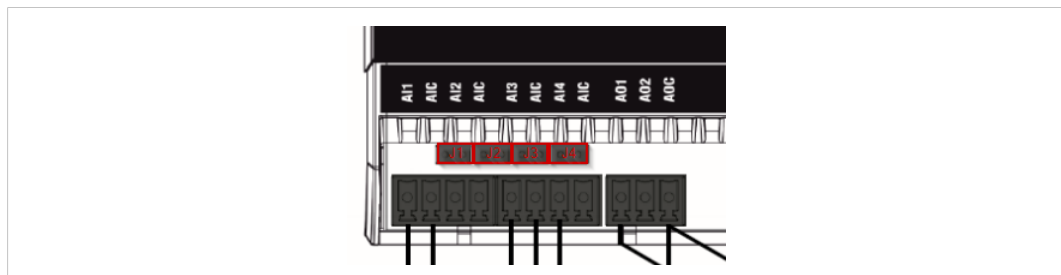
Fig. 17   CTI 117 Jumpers

- 4 Relay outputs (named *Relay 0 - Relay 3*)

- 2 Solid State Relay outputs (named *solidStateRelay 0 - solidStateRelay 1*)

- 2 analog outputs (named *Analog Output 0 - Analog Output 1*)

  Each analog output can deliver a value between 0 and 10V, max 10mA.

Each SGIM Digital I/O module supports:

- 8 digital inputs (named *Digital In 0 - Digital In 7*)

- 2 digital outputs (named *Digital Out 0 - Digital Out 1*)

Limits

For all I/O interfaces a maximum of 8192 I/O values can be defined. For this limit each array element is counted separately.

Event Log Messages

The following section describes the messages that the local I/O drivers may log to the Event Log. Depending on the specific driver, messages may be prefixed with one of the following texts:

- Local I/O driver (EXT01):

- Local I/O driver (EXT03):

- Local I/O driver (EXT05):

- Local I/O driver (EXT07):

- Local I/O driver (EXT08):

- Local I/O driver (EXT10):

- Local I/O driver (EXT12):

- SGIM I/O Driver: :

- **Could not find I/O module**

    A digital I/O module could not be discovered.

- **Could not Initialise I/O module: Could not set port configuration**

    The port configuration of a digital I/O module could not be set.

- **Could not Initialise I/O module: Could not disable master intensity**

    The output power of a digital I/O module could not be set.

- **Could not Initialise I/O module: Could not set configuration**

    The configuration of a digital I/O module could not be set.

- **Could not Initialise I/O module: Could not adjust PWM direction**

    The PWM direction of a digital I/O module could not be set.

- **Initialising I/O hardware failed**

    A general error has occurred during the hardware initialisation.

- **Output power lost**

    The power supply voltage for the local outputs has been removed.

- **Output power supplied**

    The power supply voltage for the local outputs has been supplied.

- **Output *<x>* overloaded**

    Digital output number *x* has been overloaded.

- **Output *<x>* no longer overloaded**

    Digital output number *x* is no longer overloaded.

- **PWM was set to illegal mode = *<mode>*.**

    The PWM output was set to an illegal mode with the given mode number. The setting was ignored.

- **Could not read Firmware version or it is not supported**

    The Device could either not establish a connection with the module, or the module firmware version is outdated.

- **Could not open a communication channel with the module**

    The necessary resources to open a communication channel with the module could not be reserved. Communication with the module could not be established.

- **Unable to open Sensor configuration file. The file is either damaged or missing**

    The configuration file for this sensor is either missing or could not be accessed.

- **Could not initialise RTD module**

  The resistance temperature detector could not be initilaized. Values that are associated this module are probably wrong.

- **Portal communication error *<error code>***

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

If you see an error code that is not on the above list, please contact customer support.

Modbus Client/Master

The Modbus client/master can be configured through this form.

Devices and values

At the top there is a list of Modbus devices and their values. You can add, edit, delete or clone devices and values (input or output) via the buttons below the list. For each device or value the list shows a name, the communication parameters and an index. The name can be freely defined and will e.g. be shown on the portal. The indexes are used internally to address the devices and their values.

When adding or editing a device, a dialog will show up where you can configure the following parameters:

Name	The device's name
Type	Here you have to select whether the device will be accessed via Modbus TCP or RTU. The Modbus RTU option will only be available after you have generally enabled Modbus RTU (see below).
Interface	Only present for Modbus RTU devices Serial interface the device is connected to
Address	Only present for Modbus RTU devices Device's address
IP Address	Only present for Modbus TCP devices Device's IP address (e.g. 192.168.0.3)
Port	Only present for Modbus TCP devices Device's TCP port number - typically 502

When adding or editing a value, another dialog will show up. Here you can configure the following parameters:

Name	The value's name
Unit identifier	Only present for Modbus TCP servers For Modbus TCP/RTU gateways this is the address of the RTU device to read/write from/to. For non-gateway devices this value should typically be 255.
Function	Modbus function code used to read/write the value The function code also defines whether the value is interpreted as an input or output.
Start address	Register/coil address to start reading/writing from/to Note that addresses start at 0.
Quantity	Number of registers/coils to read/write If more registers/coils are read/written than fit into the selected data type (see below), the value will become an array. For example 2 registers would fit into one 32-bit unsigned integer value. The value would be scalar (no array). 4 registers would make up an array with 2 32-bit unsigned integer elements.
Data type	Data type of the value
Swap words	Modbus transmits the two bytes of one register in network byte order (big endian). Thus for multi-register values, the registers are expected to be in the same order. However, some devices store multi-register values in the opposite order. This option can be used to correct the word order.

Limits

You can define up to 128 devices and 4096 values. However, since values can become arrays when reading/writing several registers/coils, there is another limit: The number of all scalar (non-array) values plus the size of all arrays must not exceed 8192.

Settings

Below the device list there are several fields through which some general settings can be configured:

Interval	Defines the interval in milliseconds in which all Modbus values will be read/written; Note that if the interval is too short to read/write all values, a warning message will be shown on the event log. In this case you should increase the interval. In the worst case reading/writing a value may take twice the time specified for the timeout (see below), i.e. when the driver has to connect to a Modbus TCP device (first timeout) before reading/writing the value (second timeout).
Timeout	Defines the maximum time in milliseconds to wait for the reply of a device when reading/writing one of its values or when connecting to a Modbus TCP device.

Additionally the communication parameters for Modbus RTU can be set. If your Edge gateway device features several serial interfaces those can be configured separately.

Enable RTU	Generally enables Modbus RTU.
Mode	Defines the mode for Modbus RTU. There are three possible options: • RS232 with RTS/CTS flow control • RS232 without flow control • RS485
Baud rate	Defines the baud rate for Modbus RTU.
Parity	Defines the parity (and stop bits) for Modbus RTU. There are four possible options: • None (2 stop bits) • Odd • Even • None (1 stop bit) - This option is illegal according to the Modbus specification, but still some slaves use it.
Minimum silent time between frames	Defines the minimum time for which there has to be silence on the bus between two frames. The Modbus RTU driver calculates the silence time depending on the baud rate. Normally this should be sufficient. However you can use this setting to increase the silent time. If the time you configure here is less than the calculated time, your setting will be ignored.

Event Log Messages

The following section describes the messages that the Modbus driver may log to the Event Log. All Messages are prefixed with the text *Modbus driver:*.

• **Connected to TCP device at *<IP address>:<port>*.**

The driver successfully connected to the Modbus TCP server with the given IP address and port. This message will only be shown if the driver failed to connect to the same server before.

• **Read/wrote from/to TCP device at *<IP address>:<port>* (unit: *<unit identifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*)**

The driver successfully read/wrote a value from/to the Modbus TCP server with the given IP address and port. The corresponding value is identified by the Modbus function code used to read/write it, its address (starting from 0) and the quantity of registers/coils. This message will only be shown if the driver failed to read/write the same value before.

• **Read/wrote from/to RTU device *<serial port>:<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*)**

The driver successfully read/wrote a value from/to the Modbus RTU slave with the given address. The corresponding value is identified by the Modbus function code used to read/write it, its address (starting from 0) and the quantity of registers/coils. This message will only be shown if the driver failed to read/write the same value before.

- **The interval is being maintained again**

  This message appears after this message, when the interval is being maintained again.

- **The interval has been violated**

  The driver was still busy reading/writing slave/client values when a new cycle should be started. You should probably increase the interval time.

- **Could not connect to TCP device at *<IP address>:<port>*. (Error code *<code>*).**

  The driver could not connect to the Modbus TCP server with the given IP address and port. The message also shows an error code:

-2	Could not open TCP socket
-3	Could not switch TCP socket to blocking mode
-4	TCP connect failed
-5	Setting options on TCP socket failed

  If you see an error code that is not on the above list, please contact the customer support.

- **Could not read/write from/to TCP device at *<IP address>:<port>* (unit: *<unit idenfifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*, result code: *<result code>*, error code: *<error code>*, exception code: *<exception code>*)**

  The driver failed to read/write a value from/to the Modbus TCP server with the given IP address and port. The corresponding value is identified by the Modbus function code used to read/write it, its address (starting from 0) and the quantity of registers/coils. The message also shows a result code, an error code from the TCP/IP stack and a Modbus exception code. For documentation on Modbus exception codes please refer to the Modbus application protocol specification. The following table shows the possible result codes:

-1	Waiting for server response timed out. Either the server is not available or you should increase the timeout.
-2	Received invalid packet (protocol error)
-5	TCP/IP error

  If you see an error code that is not on the above list, please contact the customer support.

- **Read illegal floating point value from TCP device at *<IP address>:<port>* (unit: *<unit idenfifier>*, function code: *<function code>*, address: *<address>*, quantity: *<quantity>*)**

  The driver read an illegal floating point value from the Modbus TCP server with the given IP address and port. The corresponding value is identified by the Modbus function code used to read it, its address (starting from 0) and the quantity of registers/coils.

- **Could not read/write from/to RTU device *<serial port>:<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*, result code: *<result code>*, exception code: *<exception code>*)**

  The driver failed to read/write a value from/to the Modbus RTU slave with the given address. The corresponding value is identified by the Modbus function code used to read/write it, its address (starting from 0) and the quantity of registers/coils. The message also shows a result code and a Modbus exception code. For documentation on Modbus exception codes please refer to the Modbus application protocol specification. The following table shows the possible result codes:

-1	Waiting for slave response timed out. Either the slave is not available or you should increase the timeout.
-2	Received invalid packet (Checksum error)
-3	Sending request failed

  If you see a result code that is not on the above list, please contact the customer support.

- **Read illegal floating point value from RTU device *<serial port>*:*<device address>* (function code: *<function code>*, address: *<value address>*, quantity: *<quantity>*)**

  The driver read an illegal floating point value from the Modbus RTU slave with the given address. The corresponding value is identified by the Modbus function code used to read it, its address (starting from 0) and the quantity of registers/coils.

  Portal communication error *<error code>*

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

Modbus Server/Slave

The Modbus server/slave can be configured through this form.

Values

At the top there is a list of Modbus values. You can add, edit, delete or clone values (input or output) via the buttons below the list. For each value the list shows a name, the communication parameters and an index. The name can be freely defined and will e.g. be shown on the portal. The indexes are used internally to address the values.

When adding or editing a value, a dialog will show up. Here you can configure the following parameters:

Name	The value's name
Function	Modbus function code used to read/write the value The function code also defines whether the value is interpreted as an input or output.
Start address	Register/coil address to start reading/writing from/to Note that addresses start at 0.
Quantity	Number of registers/coils to read/write If more registers/coils are read/written than fit into the selected data type (see below), the value will become an array. For example 2 registers would fit into one 32-bit unsigned integer value. The value would be scalar (no array). 4 registers would make up an array with 2 32-bit unsigned integer elements.
Data type	Data type of the value

Limits

You can define up to 4096 values. However, since values can become arrays when reading/writing several registers/coils, there is another limit: The number of all scalar (non-array) values plus the size of all arrays must not exceed 8192. The Modbus TCP Server/Slave can serve up to 32 concurrent connections.

Settings

Below the value list there are several fields through which some general settings can be configured:

Mode	Defines whether to operate as a Modbus RTU Slave or a Modbus TCP Server. According to this particular setting, two different groups of additional settings are available.

TCP settings

Port	TCP port number the Modbus TCP server will be listening on - typically 502. Note that the unit ID is not evaluated in TCP server mode.
Keep-alive enable	Enables the TCP keep-alive mechanism for connections from clients. The keep-alive mechanism is used to detect dead connections. When a dead connection is detected, resources consumed by this connection can be released.
Keep-alive timeout	Idle time in seconds after which the first keep-alive probe will be sent
Keep-alive interval	Interval in seconds in which keep-alive probes will be sent
Keep-alive count	Number of keep-alive probes that will be sent before closing the connection

RTU settings

Interface	Serial interface the Modbus RTU Slave should run on. This selection is only available if your Edge gateway device features more than one serial interface.
Address	RTU slave's address
RTU mode	Defines the mode for Modbus RTU. There are three possible options: • RS232 with RTS/CTS flow control • RS232 without flow control • RS485
Baud rate	Defines the baud rate for Modbus RTU
Parity	Defines the parity (and stop bits) for Modbus RTU. There are three possible options: • None (2 stop bits) • Odd • Even

Event Log Messages

The following section describes the messages that the Modbus Slave driver may log to the Event Log. All Messages are prefixed with the text *Modbus Slv driver:*.

• **Portal communication error *<error code>***

A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

If you see an error code that is not on the above list, please contact the customer support.

M-Bus

If your Edge gateway device features an M-Bus interface, you can configure it via this form.

At the top of the page you can configure several settings.

Baud rate	Baud rate at which the M-Bus will operate.
Addressing mode	Mode used to address the M-Bus slave devices (Primary, secondary or reverse secondary).
Append new devices to the list	When this option is set, the scan results are appended to the saved List.
Maximum number of pages per device	Via this option you can limit the number of pages to be requested from each device. Typically the first page contains the most important values.
Maximum number of	Via this option you can limit the number of values to be read from each device. Note that the total number of values read from all devices is limited to 4096 and that at least one value will be read from any device. The number of devices to be read is limited to 64.

values per device	
Readout interval	Interval in seconds in which the M-Bus devices will be read.
Request timeout	Timeout until first data are received in milliseconds
Idle timeout	Timeout used in detecting end of communication occurrences in milliseconds
Full timeout	Timeout of the whole request for receiving a data packet in milliseconds.
Request Mode	Defines the request mode for the M-Bus interface; there are three possible modes: • Standard - *REQ_UD2* request for Class 2 Data • Extended 1 - *REQ_UD2* request for Class 2 Data with *Get-All-Data(DIF 7E)* • Extended 2 - *REQ_UD2* request for Class 2 Data with *Get-All-Data(DIF/VIF 7F 7E)*
Reset Mode	Defines the reset mode for the M-Bus interface, There are five possible modes: • Before - *SND_NKE* request before reading the meter to its primary address, or broadcast if secondary addressing is used • After - *SND_NKE* request after reading the meter to its primary address, or broadcast if secondary addressing is used • None • Extended 1 - *SND_NKE* request to the 0xFD primary address and the meter primary address, or broadcast if secondary addressing is used • Extended 2 - *SND_NKE* and a Application reset request to both the 0xFD address and the meter primary address, or broadcast if secondary addressing is used

Below there is a list of meter devices and their values. The list shows a name, an index, the primary address (only for devices), a scale factor (values only) and a unit (values only). For devices the name is build from the manufacturer name, the measured medium, the serial number and (in braces) the version of the device. For values the name explains the meaning of the value. The indexes are used internally to address the devices and their values.

> **!** Meter devices typically do not deliver their values in the units they state. They also deliver a scale factor that has to be applied to reach the designated unit. The HMS Hub firmware generally works with unscaled values to avoid roundoff errors. Only when passing the meter values to the portal, the scale factor will be passed, too and the portal will scale the values appropriately.

After changing the physical bus configuration, you will typically scan the bus by clicking the scan button below the list. A dialog will appear. If you have chosen primary addressing, the dialog will let you set the address range to scan. In secondary and reverse secondary addressing mode the dialog will let you set a 32-bit mask (First 8 bits: device version, next 8 bits: medium type, last 16 bits: manufacturer ID) in order to query only certain devices. Use the mask FFFFFFFF to scan for any devices. After the scan has completed, the list will show the new configuration. Then you should save the new configuration to the device by clicking the *Save* button. Note that the HMS Hub firmware Run Time System will still be working with the old configuration. To apply the new configuration, you have to reboot the device.

Event Log Messages

The following section describes the messages that the M-Bus driver may log to the Event Log. All messages are prefixed with the text *M-Bus driver:*.

• **Read values from device "*<manufacturer>* - *<medium>* - *<serial number>*".**

The values of the given slave were successfully read. This message is shown only if reading the slave's values has failed before.

- **Read value "<description>" from device "<manufacturer> -<medium> -<serial number>"**

  The given value was successfully read from the given slave. This message is shown only if reading the value has failed before.

- **The readout cycle is being maintained again.**

  This message appears after this message, when the readout cycle is being maintained again.

- **Could not re-initialise M-Bus library**

  When scanning the internal M-Bus library has to be re-initialised in order to change settings like baud rate or addressing mode. If this message shows up, re-initialising the M-Bus library failed. You should restart the device.

- **Scanning for slaves failed**

  The driver could not scan for slaves.

- **Determining slaves' configuration failed**

  After the driver has scanned the bus for slaves, it could not determine the values provided by the slaves.

- **The readout cycle has been violated.**

  The driver was still busy reading the slaves' values when a new readout cycle should be started. You should probably increase the readout cycle.

- **Could not read values from device "*<manufacturer> - <medium> - <serial number>*".**

  The values of the given slave could not be read.

- **Could not read value "*<description>*" from device "*<manufacturer> - <medium> - <serial number>*".**

  The given value could not be read from the given slave.

- **Read illegal floating point value "*<description>*" from device "*<manufacturer> - <medium> - <serial number>*".**

  An illegal floating point value was read for the given value from the given slave.

- **Portal communication error *<error code>***

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

EtherNet/IP Adapter/Slave

If your device features an EtherNet/IP module, the EtherNet/IP adapter can be configured through this form. The EtherNet/IP module is based on the Anybus CompactCom module by HMS Industrial Networks.

Groups and values

At the top there is a list where you can define groups and the values that are available inside these groups. Every group has a name. Every value has a name, a data type and a length. Additionally the list shows an index for each group and value. The indexes are used internally to address the groups and their values. You can add, edit, delete or clone groups and values (input or output) via the buttons below the list.

When adding or editing a group, a dialog will show up where you can configure the following parameters:

Name	The group's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same group name cannot be used more than once.

When adding or editing a value, another dialog will show up. Here you can configure the following parameters:

Name	The value's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same value name cannot be used more than once inside the same group.
Data type	This is the data type of the value.
Length	If the length is greater than 1, an array will be defined. Arrays can have up to 255 elements.

Limits

You can define up to 128 groups and 2896 values. The number of all scalar (non-array) values plus the size of all arrays must not exceed 2896. The number of bytes transmitted in each direction is limited to 1448.

Settings

Below the group and value list general settings can be configured:

Use DHCP	If this option is set, the module will get its IP configuration through DHCP.
IP-Adresse	IP address for the EtherNet/IP network interface. This option is deactivated if DHCP is used.
IP Netmask	Netmask for the EtherNet/IP network interface. This option is deactivated if DHCP is used.
Gateway IP address	Gateway IP address for the EtherNet/IP network interface. This option is deactivated if DHCP is used.
Store changes from external tools	If this option is set, changes to the module's IP configuration (DHCP, Address, Netmask, gateway) through external tools will be stored. Otherwise these changes are lost on the next reboot.

Download EDS file

Through the Download button, you can download the EDS file. The EDS (*Electronic Data Sheet*) file can be imported into a EtherNet/IP Scanner Tool. It includes the values defined above.

Acyclic data access

The I/O values can be accessed acyclically through the CIP object 0xA2. Each instance of this object starting from instance 1 represents a group. Input groups appear before output groups. For each group a set of parameters can be accessed:

Attribute 1	Group name (SHORT_STRING, read only)
Attribute 2	Array that holds the data type of each value in the group (USINT, read only). For array values there is one entry per array element. The following data types may appear: 0 BOOL 1 SINT 2 INT 3 DINT 4 USINT 5 UINT 6 UDINT 7 REAL

Attribute 3	Number of elements (USINT, read only) counting all array elements separately.
Attribute 4	Array that holds an access descriptor for each element (USINT, read only). If bit 0 is set, the corresponding element is readable through EtherNet/IP. If bit 1 is set, it is writeable.
Attribute 5	Values. The size of this attribute depends on the data types (attribute 2) and the number of elements (attribute 3). Whether a value is writeable depends on the corresponding access descriptor (attribute 4).

Event Log Messages

The following section describes the messages that the EtherNet/IP Adapter driver may log to the Event Log. All messages are prefixed with the text *Anybus CompactCom driver (EXT09):*.

- **EtherNet/IP module detected**

    An Anybus CompactCom module of EtherNet/IP type has been detected.

- **Resetting Anybus CompactCom module**

    The Anybus CompactCom module will be reset - either due to an error or due to a request from the network.

- **IP configured**

    An IP address was assigned to the Anybus CompactCom module.

- **DHCP client was (activated or deactivated)**

    The Anybus CompactCom module's DHCP client was activated or deactivated from an external tool.

- **Received illegal floating point value**

    An illegal floating point value has been received from the network.

- **Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)**

    An error occurred in the base driver for the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Connection to Anybus CompactCom module lost**

    There was a communication error between the main processor and the Anybus CompactCom module.

- **Connection to Anybus CompactCom module reestablished**

    After a communication error between the main processor and the Anybus CompactCom module the communication was reestablished successfully.

- **Class 1 connection with scanner closed**

    The class 1 connection with the EtherNet/IP scanner has been closed.

- **Class 1 connection idle**

    The class 1 connection with the EtherNet/IP scanner is idle.

- **Class 1 connection active**

    The class 1 connection with the EtherNet/IP scanner is now active.

- **Class 1 connection error or duplicate IP address**

    There is either a class 1 connection error or a duplicate IP address has been detected.

- **Unexpected state: *<state>***

    The state machine of the base driver for the Anybus CompactCom module has reached the indicated unexpected state. This should not happen. Please contact the customer support.

- **Could not (set or get) DHCP**

  The DHCP option could not be written or read into the Anybus CompactCom module. This
  should not happen. Please contact the customer support.

- **Could not (set or get) IP address**

  The IP address could not be written or read into the Anybus CompactCom module. This
  should not happen. Please contact the customer support.

- **Could not (set or get) subnet mask**

  The IP netmask could not be written or read into the Anybus CompactCom module. This
  should not happen. Please contact the customer support.

- **Could not (set or get) gateway IP address**

  The gateway IP address could not be written or read into the Anybus CompactCom module.
  This should not happen. Please contact customer support.

PROFIBUS DP Slave

If your Edge gateway device features a PROFIBUS module, the PROFIBUS DP Slave can be
configured through this form. The PROFIBUS module is based on the Anybus CompactCom
module by HMS Industrial Networks.

Modules and values

At the top there is a list where you can define your modules and the values that are available
inside these modules. Every module has a name and a module ID. Every value has a name, a data
type and a length. Additionally the list shows an index for each module and value. The indexes
are used internally to address the modules and their values. You can add, edit, delete or clone
modules and values (input or output) via the buttons below the list.

When adding or editing a module, a dialog will show up where you can configure the following
parameters:

Name	The module's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same module name cannot be used more than once.
Module ID	The module ID is used to address the module on the PROFIBUS. It is a numerical value between 1 and 255. Note that the same module ID cannot be used more than once.

When adding or editing a value, another dialog will show up. Here you can configure the
following parameters:

Name	The value's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same value name cannot be used more than once inside the same module.
Data type	This is the data type of the value.
Length	If the length is greater than 1, an array will be defined. Arrays can have up to 255 elements.

Limits

You can define up to 48 modules and 488 values. The number of all scalar (non-array) values plus
the size of all arrays must not exceed 488. The number of bytes transmitted in each direction is
limited to 244.

The maximum number of modules is reduced the more data the modules contain. The data size
of a module is hereinafter calculated in bytes if the byte count is odd, else it is calculaed in words
(two bytes). The identifier of a module, which is e.g. also used in the GSD file, consists of 5 bytes
for up to 64 bytes/words of data. For each 64 bytes/words of data another 3 bytes are added to
the identifier. All module identifiers concatenated must not consume more than 244 bytes.

Only a limited number of data bytes per module can be transmitted consistently through PROFIBUS. If the number of data bytes of a module is even, up to 128 bytes can be transmitted consistently. If the number of data bytes is odd, up to 63 bytes can be transmitted consistently. Note that the data of a certain module cannot be transmitted consistently if you exceed these limits.

Settings

Below the module and value list general settings can be configured:

Slave Address	If you have several Edge gateway devices with different configurations in your PROFIBUS network, the PROFIBUS Master must be able to distinguish these devices. Thus they need to have different Slave Addresses. You can chose a Slave Address between 2 and 125.

Download GSD file

Through the Download button, you can download the GSD file including a corresponding graphic file. The GSD file can be imported into the PROFIBUS Master. It includes the modules and values defined above.

Acyclic data access

The defined modules and their values can also be accessed acyclicly. Modules can only be accessed in a whole. It is not possible to access individual values. Modules are not addressed through slot numbers. Instead requests must always go towards slot 1 and the index determines the module ID. So if you want to access module 6, send a request with slot number 1 and index 6.

Several Anybus Edge devices in the same network

If you want to use several Edge gateway devices in your PROFIBUS DP network and these don't have the same configuration, you should do the following:

• Don't use the same module ID on different Edge gateway devices.

• Export the GSD files of all Edge gateway devices.

• Merge the GSD files into one. The files will be the same except for the module entries. Copy all module entries into one GSD file.

• Use the merged GSD file for all devices, but only use the modules that have been configured on the corresponding Edge gateway device.

Event Log Messages

The following section describes the messages that the PROFIBUS DP Slave driver may log to the Event Log. All messages are prefixed with the text *Anybus CompactCom driver (EXT09):*.

• **PROFIBUS DP module detected**

  An Anybus CompactCom module of PROFIBUS DP type has been detected.

• **Resetting Anybus CompactCom module**

  The Anybus CompactCom module will be reset - either due to an error or due to a request from the network.

• **Received illegal floating point value**

  An illegal floating point value has been received from the network.

• **Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)**

  An error occurred in the base driver for the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Connection to Anybus CompactCom module lost**

  There was a communication error between the main processor and the Anybus CompactCom module.

- **Connection to Anybus CompactCom module reestablished**

  After a communication error between the main processor and the Anybus CompactCom module the communication was reestablished successfully.

- **Deactivating module due to empty process data configuration**

  Due to the nature of the PROFIBUS networking system, the PROFIBUS DP slave module cannot be activated, if no process data has been defined. This message informs about the fact that the module is being deactivated, because no process data has been defined.

- **Left state Data Exchange**

  The PROFIBUS DP slave module has left the Data Exchange state. No MS0 connection is open. The PROFIBUS DP state machine is in either the Power On/Reset, Parametrization (WAIT_PRM) or I/O Configuration (WAIT_CFG) state.

- **Entered state Data Exchange, Master in Clear mode**

  The PROFIBUS DP slave module has entered the Data Exchange state. The MS0 connection is established. The PROFIBUS DP state machine is in the Data Exchange (DATA_EXCH) state. The PROFIBUS DP master is in the Clear Mode.

- **Entered state Data Exchange, Master in Operate mode**

  The PROFIBUS DP slave module has entered the Data Exchange state. The MS0 connection is established. The PROFIBUS DP state machine is in the Data Exchange (DATA_EXCH) state. The PROFIBUS DP master is in the Operate Mode.

- **Could not set slave address**

  The slave address of the PROFIBUS DP module could not be set. This should not happen. Please contact the customer support.

- **Unexpected state: *<state>***

  The state machine of the base driver for the Anybus CompactCom module has reached the indicated unexpected state. This should not happen. Please contact the customer support.

PROFINET IO Device

If your Edge gateway device features a PROFINET module, the PROFINET IO Device can be configured through this form. The PROFINET module is based on the Anybus CompactCom module by HMS Industrial Networks.

Modules and values

At the top there is a list where you can define your modules and the values that are available inside these modules. Every module has a name and a module ID. Every value has a name, a data type and a length. Additionally the list shows an index for each module and value. The indexes are used internally to address the modules and their values. You can add, edit, delete or clone modules and values (input or output) via the buttons below the list.

When adding or editing a module, a dialog will show up where you can configure the following parameters:

Name	The module's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same module name cannot be used more than once.
Module ID	The module ID is used to address the module on the PROFINET. It is a numerical value between 1 and 65535. Note that the same module ID cannot be used more than once.

Advance settings	If this option is set, it allows the configuration of the IP settings and station name. Normally these settings are assigned via the PROFINET DCP protocol, but in some cases it might be useful to preconfigure a device. If a factory default reset command is received by the module, the IP settings and Station name will be reset to their default values.
Use DHCP	If this option is set, the module will get its IP configuration through DHCP. In the default PROFINET settings is DHCP deactivated.
IP address	IP address of the module; this option is deactivated if DHCP is used. In the default PROFINET settings the standard IP address 0.0.0.0 is used.
IP Netmask	Netmask of the module; this option is deactivated if DHCP is used.
Gateway IP address	Gateway IP address of the module; this option is deactivated if DHCP is used.
Station name	The station name identifies the module on the PROFINET network.
Store changes from external tools	If this option is set, changes to the module's IP configuration (DHCP, Address, Netmask, gateway) through external tools will be stored. Otherwise these changes are lost on the next reboot.

When adding or editing a value, another dialog will show up. Here you can configure the following parameters:

Name	The value's name can be freely defined and will for example be shown on the portal. It can consist of 1 to 64 printable ASCII characters. Note that the same value name cannot be used more than once inside the same module.
Data type	This is the data type of the value.
Length	If the length is greater than 1, an array will be defined. Arrays can have up to 255 elements.

Limits

You can define up to 128 modules and 2880 values. The number of all scalar (non-array) values plus the size of all arrays must not exceed 2880. The number of bytes transmitted in each direction is limited to 1440. Note that each module consumes one byte in each direction and that the DAP (Device Access Point) implicitly defines 4 modules.

Settings

Below the module and value list general settings can be configured:

Device ID	If you have several Edge gateway devices with different configurations in your PROFINET network, the IO Supervisor must be able to distinguish these devices. Thus they need to have different device IDs. You can chose a device ID between 1 and 100.
Device name suffix	Edge gateway devices with different configurations and device IDs should also have different names. Thus you can configure a suffix that will be appended to the device name. The suffix can consist of up to 32 alphanumeric characters or hyphens. It must not end with a hyphen.

Download GSD file

Through the Download button, you can download the GSD file including a corresponding graphic file. The GSD file can be imported into the IO Supervisor. It includes the modules and values defined above.

Acyclic data access

The defined modules and their values can also be accessed acyclicly. Modules can only be accessed in a whole. It is not possible to access individual values. Modules are not addressed through slot and subslot numbers. Instead requests must always go towards slot 1 and the index determines the module ID. So if you want to access module 6, send a request with slot number 1, subslot number 1 and index 6.

Event Log Messages

The following section describes the messages that the PROFINET IO device driver may log to the Event Log. All messages are prefixed with the text *Anybus CompactCom driver (EXT09):*.

- **PROFINET IO module detected**

    An Anybus CompactCom module of PROFINET IO type has been detected.

- **Advanced settings activated**

    The Advanced settings are active.

- **Resetting Anybus CompactCom module**

    The Anybus CompactCom module will be reset - either due to an error or due to a request from the network.

- **IP configured**

    An IP address was assigned to the Anybus CompactCom module.

- **DHCP client was (activated or deactivated)**

    The Anybus CompactCom module's DHCP client was activated or deactivated from an external tool.

- **Received illegal floating point value**

    An illegal floating point value has been received from the network.

- **Anybus CompactCom base driver error. (Error code *<error code>*, Info *<information code>*)**

    An error occurred in the base driver for the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Connection to Anybus CompactCom module lost**

    There was a communication error between the main processor and the Anybus CompactCom module.

- **Connection to Anybus CompactCom module reestablished**

    After a communication error between the main processor and the Anybus CompactCom module the communication was reestablished successfully.

- **IO connection with IO controller closed**

    The IO connection with the IO controller has been closed.

- **IO controller in STOP mode**

    The IO controller with which an IO connection is established is in STOP mode or the IO controller has not provided valid output at least once.

- **IO connection with IO controller established**

    The IO connection with the IO controller has been established and valid output data has been provided at least once.

- **Configuration data mismatch**

    There is a mismatch between the I/O data configuration expected by the IO controller and the actual I/O data configuration of the Edge gateway device.

- **Unexpected state: *<state>***

    The state machine of the base driver for the Anybus CompactCom module has reached the indicated unexpected state. This should not happen. Please contact the customer support.

- **Could not set DHCP**

    The DHCP option could not be written into the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Could not set IP address**

    The IP address could not be written into the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Could not set subnet mask**

    The IP netmask could not be written into the Anybus CompactCom module. This should not happen. Please contact the customer support.

- **Could not set gateway IP address**

    The gateway IP address could not be written into the Anybus CompactCom module. This should not happen. Please contact the customer support.

IEC 60870-5-104 Server

Via this form the IEC 60870-5-104 Server can be configured.

Values

At the top there is a list of values (Information objects). You can add, edit, delete or clone values (input or output) via the buttons below the list. The list shows for each value a name, the type ID, information object address, range (number of conscutive objects), interrogation group, cyclic transmission time and an index. The name can be freely defined and will e.g. be shown on the portal. The indexes are used internally to address the values.

When adding or editing a value, a dialog will show up. Here you can configure the following parameters:

Name	Value's name
Type ID	Value's 60870-5-104 type ID The type ID also defines whether the value is interpreted as an input or output. Note that for type IDs with time tag timestamps before 1980-01-01T00:00:00 are transmitted as 1980-01-01T00:00:00.
IOA (complete)	Value's 60870-5-104 information object address (bits 23..0)
IOA3 | IOA2 | IOA1	Value's 60870-5-104 information object address in single bytes: IOA3 - high byte (bits 23..16) IOA2 - medium byte (bits 15..8) IOA3 - low byte (bits 7..0)
Range	Number of consecutive objects to be read/written If e.g. the information object address is set to 1 and the range is set to 3, three objects with the addresses 1, 2 and 3 will be read/written. If the range is greater than 1, the value will become an array.
Interrogation group	Value's 60870-5-104 interrogation group
Cyclic transmission time	Value's cyclic transmission time in ms. Additionally the value will be transmitted spontaneously when it changed. If the cyclic transmission time will be set to 0 the value will be transmitted only spontaneously.

Limits

You can define up to 4096 values. However, since values can become arrays when reading/writing several objects, there is another limit: The total number of objects must not exceed 8192.

Settings

Below the list there are several fields via which some general settings can be configured:

Interval	Interval in which the IEC 60870-5-104 Server driver updates its values. The more values you define, the more time the driver will need to update them and the greater should the interval be. Note that if the interval is too short to update all values, a warning message will be shown on the event log. In this case you should increase the interval.
Common Address	Server's common address
K	Number of unacknowledged I format APDUs after which the connection will be closed
W	Number of received I format APDUs after which reception will be acknowledged at the latest

T1	Timeout of send or test APDUs in seconds
T2	Timeout after which an S frame will be sent in seconds
T3	Timeout after which a test frame will be sent in case of long idle state in seconds
TCP Port	TCP port number
Short Pulse Time	Short Pulse Time in milliseconds
Long Pulse Time	Long Pulse Time in milliseconds
Buffer Size	Number of events that are stored and will be sent to the client in case of a connection loss
Maximum number of connections	Maximum number of clients, that can connect to this server. More than 1 connection enables also redundant connections.
Generate activation termination	If true, an activation termination will be generated - e.g. when a pulse command terminates
Accept clock synchronisation command	If this option is set, the clock synchronisation command (C_CS, 103) will be accepted and the time from the client will be transferred to the system clock. **!** Please note that the clock synchronisation via IEC 60870-5-104 may conflict with other clock synchronisation mechanisms. It is recommended to activate only one clock synchronisation mechanism.

Please note that the IEC 60870-5-104 Server will only be available if your Edge gateway device has a valid IEC 60870-5-104 Server licence.

Event Log Messages

The following section describes the messages that the IEC 60870-5-104 Server driver may log to the Event Log. All messages are prefixed with the text *IEC 60870-5-104 server driver:*.

- **Updated output at address*<information object address>* .**

  The output with the given information object address has been successfully updated. This message will only be shown, if updating the same output has failed before.

- **The interval is being maintained again.**

  This message appears after this message, when the interval is being maintained again.

- **Timestamps with 16 or 24 bit are not yet supported. Falling back to system time..**

  Timestamps with size 16 bit or 24 bit are not supported in this version.

- **A cyclic transmission time of types 1, 3, 5, 7, 30, 31, 33, 34 and 36 is no longer supported and has been converted to 0.**

  A cyclic transmission time of the following types is no longer supported and has been converted to 0. To permanently remove this warning, you have to save the current IEC 60870-5-104 server configuration.

    - Single-point information (1)

    - Double-point information (3)

    - Step position information (5)

    - Bitstring of 32 bit (7)

    - Single-point information with time tag CP56Time2a (30)

    - Double-point information with time tag CP56Time2a (31)

    - Bitstring of 32 bit with time tag CP56Time2a (33)

    - Measured value, normalized value with time tag CP56Time2a (34)

    - Measured value, short floating point value with time tag CP56Time2a (36)

- **Values of types 15 and 37 are lacking a cyclic transmission time.**

  For the following types a cyclic transmission time must now be given. It is missing in your configuration. Please adapt your configuration accordingly. Caution: The spontaneous transmission is no longer supported for these types.

  – Integrated totals (15)

  – Integrated totals with time tag CP56Time2a (37)

- **A cyclic transmission time in milliseconds is no longer supported and has been converted to seconds. Please check your configuration at type 'Measured value, normalized value (9)'.**

  The unit of the cyclic transmission time has been changed from milliseconds to seconds and the configuration has been converted accordingly. To permanently remove this warning, you have to save the current IEC 60870-5-104 server configuration.

- **Could not send activation termination of pulse command at address *<information object address>*.**

  An activation termination at the end of a pulse command with the shown address could not be sent.

- **Could not get timestamp at end of pulse command at address *<information object address>*.**

  Could not get the timestamp for the activation termination at the end of a pulse command with the shown address.

- **The interval has been violated.**

  The driver was still busy updating the values when a new cycle should be started. You should probably increase the interval.

- **Could not update output at address *<information object address>*. Error code *<code>*, error message *<message>***

  Updating the output object with the given information object address failed. This should not happen. Please contact the customer support.

- **Portal communication error *<error code>***

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

Licensed Software

The IEC 60870-5-104 library by SystemCORP Embedded Technology Pty. Ltd. has been used to implement the IEC 60870-5-104 interface. Website: www.systemcorp.com.au

IEC 60870-5-104 Client

The IEC 60870-5-104 client/master/controlling station can be configured through this form.

Devices and Values

At the top there is a list of IEC 60870-5-104 devices/servers/slaves/ controlled stations and their values (information objects). You can add, edit, delete or clone devices and values (input or output) via the buttons below the list. The list shows for each device or value a name, the communication parameters and an index. The name can be freely defined and will e.g. be shown on the portal. The indexes are used internally to address the values.

When adding or editing a device, a dialog will show up where you can configure the following parameters:

Name	The device's/servers's name.
IP addresses and corresponding ports:	The IEC 61870-5-104 client supports redundant connections. Here up to four pairs of IP address and corresponding port can be configured. Usually only the first pair will be used. But if the connection is lost, the other pairs will be tried.
Common Address	Server's common address.
K	Number of unacknowledged I format ADPUs after which the connection will be closed.
W	Anzahl von empfangenen I-Format-APDU bevor der Empfang spätestens bestätigt wird.
T0	Timeout of connection establishment in seconds.
T1	Timeout of send or test APDUs in seconds.
T2	Timeout after which an S frame will be sent in seconds.
T3	Timeout after which a test frame will be sent in case of long idle state in seconds.
General interrogation interval	Cyclically send general interrogation command. Cycle time in seconds. Use zero to disable.
Interrogation group 1 to 16 interval	Cyclically send interrogation command for the specified group. Cycle time in seconds. Use zero to disable.
Counter interrogation interval	Cyclically send counter interrogation command. Cycle time in seconds. Use zero to disable.
Counter interrogation group 1 to 4 interval	Cyclically send counter interrogation command for the specified group. Cycle time in seconds. Use zero to disable.
Command timeout	Timeout for acknowledgement of commands in milliseconds.
Server uses ACTTERM	Check if the connected server notifies command execution termination with ACTTERM. Otherwise uncheck.

When adding or editing a value, a dialog will show up. Here you can configure the following parameters: Value's name, the type ID (it also defines whether the value is interpreted as an input or output) and the information object address, range. The range setting can be used to configure a number of consecutive values. If the information object address is set to 1 and the range is set to 3, three values with the addresses 1, 2 and 3 will be created.

Name	Value's name
Type ID	Value's 60870-5-104 type ID The type ID also defines whether the value is interpreted as an input or output. Note that for type IDs with time tag timestamps before 1980-01-01T00:00:00 are transmitted as 1980-01-01T00:00:00.
IOA (complete)	Value's 60870-5-104 information object address (bits 23..0)
IOA3 | IOA2 | IOA1	Value's 60870-5-104 information object address in single bytes: IOA3 - high byte (bits 23..16) IOA2 - medium byte (bits 15..8) IOA3 - low byte (bits 7..0)
Range	Number of consecutive objects to be read/written If e.g. the information object address is set to 1 and the range is set to 3, three objects with the addresses 1, 2 and 3 will be read/written. If the range is greater than 1, the value will become an array.

Limits

You can define up to 32 devices and 4096 values. However, since values can become arrays when reading/writing several objects, there is another limit: The total number of objects must not exceed 8192.

Settings

Below the list general settings can be configured:

Interval	Interval in which the IEC 60870-5-104 Client driver sends its requests. The more values you define, the more time the driver will need for the requests and the greater should the interval be. Note that if the interval is too short to update all values, a warning message will be shown on the event log. In this case you should increase the interval.

Please note that the IEC 60870-5-104 client will only be available if your Edge gateway device has a valid IEC 60870-5-104 client licence.

Event Log Messages

The following section describes the messages that the IEC 60870-5-104 Client driver may log to the Event Log. All messages are prefixed with the text *IEC 60870-5-104 client driver:*.

- **Updated output at address<information object address>.**

  The output with the given information object address has been successfully updated. This message will only be shown, if updating the same output has failed before.

- **The interval is being maintained again**.

  This message appears after this message, when the interval is being maintained again.

- **The interval has been violated.**

  The driver was still busy updating the values when a new cycle should be started. You should probably increase the interval.

- **Could not update output at address *<information object address>*. Error code *<code>*, error message *<message>***

  Updating the output object with the given information object address failed. This should not happen. Please contact the customer support.

- **Portal communication error *<error code>***

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

Licensed Software

The IEC 60870-5-104 library by SystemCORP Embedded Technology Pty. Ltd. has been used to implement the IEC 60870-5-104 interface. Website: www.systemcorp.com.au

IEC 61850

The IEC 61850 interface can be configured through this form.

The standard series IEC 61850 (Communication networks and systems for power utility automation) comprises mainly: Information models (logical nodes and DataObjects), configuration of a system, the information, the communication used, and the communication protocols. The standard series IEC 61400-25 (Communication networks for wind power systems) is an extension of IEC 61850. The HMS Hub firmware supports both standards with the MMS mapping.

The HMS Hub firmware implements IEC 61850 Client, Server, Publisher and Subscriber. The IEC 61850 inputs and outputs are automatically configured by an interpretation of the corresponding SCL files for the server and/or the client role.

Notes:

1. The application of the IEC 61850 requires a good understanding of the core standards of the IEC 61850 series.

2. In particular the understanding and use of the ICD and CID files for the client and server roles is a prerequisite of the integration of IEC 61850.

Logical devices and values

At the top there is a list of Logical devices and their values. You can add, or remove devices and values (input or output) by selecting them in the respective tree generated from parsing the ICD/CID file. For each device or value the list shows a name, a device type, a datatype and an index. The name is parsed from the ICD/CID file. The device type shows if the Logical device is a server or a client. The datatype field is only available for values. The indexes are used internally to address the devices and their values.

Settings

Below the Logical device list there are several fields through which some general settings can be configured:

Communication role	Defines whether to operate as a Server, Client or both
Access debug	For debugging purposes: Logs object accesses in event log
Read debug	For debugging purposes: Logs object read accesses in event log
Write debug	For debugging purposes: Logs object write accesses in event log
Update debug	For debugging purposes: Logs object update accesses in event log
Operate debug	For debugging purposes: Logs object command accesses in event log

Client Options

Connection supervision interval	Defines the interval in seconds in which it is checked if the connection to the server is still open. Note that if the interval is too short, a warning message will be shown on the event log. In this case you should increase the interval.
Polling interval	Defines the interval in seconds in which all values will be read from the server. Note that if the interval is too short to read all values, a warning message will be shown on the event log. In this case you should increase the interval.
Client SCL	Here an ICD/CID file can be uploaded or downloaded to or from the device. Once a file has been parsed, values can be selected or deselect from the generated tree.

Server Options

Enable GOOSE subscription	Enable GOOSE subscription; note that the provided ICD/CID file must also support this feature.
Server SCL	Here an ICD/CID file can be uploaded or downloaded to or from the device. Once a file has been parsed, values can be selected or deselect from the generated tree.

Event Log Messages

The following section describes the messages that the IEC 61850 driver may log to the Event Log. All messages are prefixed with the text *IEC 61850 driver:*.

- **Server IP:<IP address> , IED:<device name> , connection status: connected.**

  The driver successfully connected to the TCP server with the given IP address. This message will only be shown if the driver failed to connect to the same server before.

- **(Wrote or Read) value *<value name>* of device *<device name>* on *<interface name>* interface.**

  A value was succesfully written or read. This message will only be shown if the driver failed to read or write the same value before.

- **The interval is being maintained again.**

  This message appears after this message, when the interval is being maintained again.

- **Server IP:<IP address> , IED:<device name> , connection status: not connected.**

  The driver could not connect to the server with the given IP address.

- **Could not (Wrote or Read) value *<value name>* of device *<device name>* on *<interface name>* interface. Error code *<x>***

  A value could not be successfully read or written. Possible reasons (see *x*):

-11	There is no ready connection to given Data Attribute ID

- **The interval has been violated.**

  The driver was still busy reading client values when a new cycle should be started. You should probably increase the read or check interval time.

- **Portal communication error *<error code>***

  A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

Licensed Software

The PIS-10 library by SystemCORP Embedded Technology Pty. Ltd. has been used to implement the IEC 61850 interface.

Website: www.systemcorp.com.au

DNP3 Outstation

With this form the DNP3 Outstation can be configured.

Values

At the top is a list to define the values. For each value a name and some additional parameters are shown. The indexes (last column) are used internally to address the values. All other fields will be described below. Via the buttons below the list values can be added, edited, deleted and cloned.

When adding or editing a value, a dialog to configure the following parameters is opened:

Name	The value name is for example shown on the home page. It can consist of 1 to 64 printable ASCII characters. Note that the same value name must not be used more than once.
Portal Name	The portal name is used to build the path of the corresponding node on the portal. It can consist of 1 to 64 alphanumeric ASCII characters, underscores, dots or hyphens. The first character must be alphanumeric. Note that the same portal name must not be used more than once.
DNP3 data type	DNP3 data type of the value The data types are named from the outstation's point of view. Typically they are named from the master's point of view, "Input" and "output" must be exchanged then.
Length	If the length is greater than 1, an array is defined.
Static variation	The static variation determines in which form (data type, flags, timestamp) the value is transmitted to the master by default.
Event variation	The event variation determines in which form (data type, flags, timestamp) events concerning this value are transmitted to the master.
Class	Classes can be used to group values. The master can poll all events regarding values of a certain class together. In this way the bandwidth can be used optimally.

Deadband	With the deadband the transmission rate of frequently changing values can be reduced. A new value is only transmitted if the difference to the last transmitted value is equal or larger than the deadband.
Control Relay Output Block model	The Control Relay Output Block model determines how binary inputs (outputs from the master's point of view) are switched. Activation model In this model the binary input is activated or deactivated on command for a certain time. Complementary latch model In this model the binary input is activated or deactivated on command and remains in this state until the next command is received. Complementary two-output model In this model the binary input represents two binary inputs - *Close* and *Trip*. These are represented as a two bit value. Bit 0 (0b00000001) represents the *Close* input, bit 1 (0b00000010) the *Trip* input. At any time none or at most one of these two inputs can be active. So valid values are 0b00000000, 0b00000001 and 0b00000010. For reading back binary inputs DNP3 only provides one bit. Thus for the Complementary two-output model *Close* and *Trip* cannot be distinguished. Only the information whether any of the two inputs is active is transmitted.

Limits

You can define up to 4096 values. The maximum number of all scalar (non-array) values plus the size of all arrays is 8192.

Settings

Below the value list general settings can be configured:

Interval	Update interval of DNP3 Outstation driver interval in milliseconds.
Links	up to 4 links can be defined. A link is a connection between a Master and an Outstation. For each link the local link address and the Master's link address must be configured(valid values: 0-65519).
Port	TCP port number the DNP3 outstation is listening on
Event queue sizes	The event queue size determines how many events can be cached for the corresponding data type until the Master next polls for an event. The queue size must be between 1 and 1000.

Event Log Messages

The following section describes the messages that the DNP3 Outstation driver may log to the Event Log. All messages are prefixed with the text *DNP3 outstation driver:*.

- **Updated output with type ID *type ID* and index *index***

  The output value with the given data type ID and index was updated. This message is only shown if the corresponding output value could not be updated before.

- **Could not update output with type ID *type ID* and index *index*. Error code *error code***

  The output value with the given data type ID and index could not be updated. The corresponding DNP3 error code is shown.

- **DNP3 data type IDs**

  The data type IDs that are shown in the event log messages have the following meaning:

  0 — Binary output

  1 — Double binary output

  2 — Binary input

  3 — Analog output

  4 — Counter output

  5 — Frozen counter output

  6 — Analog input

Licensed Software

The PIS-23 library by SystemCORP Embedded Technology Pty. Ltd. has been used to implement the DNP3 outstation interface.

Website: www.systemcorp.com.au

OPC UA Server

Through this form the OPC UA Server can be configured.

Groups and values

At the top there is a list through which you can define groups and values belonging to these groups. For each group and each value a name and some additional parameters are shown. The indexes (last column) are used internally to address the groups and values. All other fields will be described below. Via the buttons below the list you can add, edit, delete or clone groups and values (input or output).

When adding or editing a group, a dialog will show up where you can configure the following parameters:

Name	The group name is e.g. shown on the home page. It is also used as *DisplayName* of the corresponding node in the OPC UA address space. It can consist of 1 to 64 printable ASCII characters. Note that the same group name must not be used more than once.
Identifier	The group identifier is used as *BrowseName* of the corresponding node in the OPC UA address space. It can consist of 1 to 32 alphanumeric ASCII characters, underscores dots or hyphens. The first character must be alphanumeric or an underscore. Note that the same group identifier must not be used more than once.
Portal Name	The portal name is used to build the path of the corresponding node on the portal. It can consist of 1 to 64 alphanumeric ASCII characters, underscores, dots or hyphens. The first character must be alphanumeric. Note that the same group portal name must not be used more than once.

When adding or editing a value, another dialog will show up. Here you can configure the following parameters:

Name	The value name is e.g. shown on the home page. It is also used as *DisplayName* of the corresponding node in the OPC UA address space. It can consist of 1 to 64 printable ASCII characters. Note that the same value name must not be used more than once inside the same group.
Identifier	The value identifier is used as *BrowseName* of the corresponding node in the OPC UA address space. It can consist of 1 to 32 alphanumeric ASCII characters, underscores dots or hyphens. The first character must be alphanumeric or an underscore. Note that the same value identifier must not be used more than once inside the same group.
Portal Name	The portal name is used to build the path of the corresponding node on the portal. It can consist of 1 to 64 alphanumeric ASCII characters, underscores, dots or hyphens. The first character must be alphanumeric. Note that the same portal name must not be used more than once inside the same group.
Data type	Data type of the value

Maximum string length	If the *String* data type is selected the maximum string length must be defined. Longer strings values will be cut.
Length	If the length is greater than 1, an array will be defined.

Limits

You can define up to 128 groups and 4096 values. The number of all scalar (non-array) values plus the size of all arrays must not exceed 8192. Up to 4 clients can be connected to the OPC UA server at the same time.

Settings

Below the group and value list general settings can be configured:

Port	TCP port the OPC UA server should listen on. Typically this is port 4840.
Interval	The OPC UA server updates its values in this interval in milliseconds.
User name	Part of the credentials a client may use to log in to the OPC UA server. It can consist of 4 to 16 printable non-whitespace ASCII characters.
Password	Part of the credentials a client may use to log in to the OPC UA server. The password requirements are shown below the input field. In delivery status the password is the same as the one you need to access the secured parts of the local website. If you don't want to change the password you can leave the two password fields empty. In this case the current password will be applied. Please note that you must initially change the password. If you leave the password fields empty when storing the configuration for the first time, you will get an error message stating that you have supplied a password that is not allowed.
Repeat password	To avoid faulty insertions the same password must be entered a second time.
Maximum number of monitored items	This value defines the maximum number of elements that can be monitored by clients.
Application URI prefix Application URI suffix	The Application URI identifies the UPC UA server instance globally uniquely. It is build from a fixed prefix and a configurable suffix. The suffic can consist of 1 to 128 characters. Allowed characters are A-Z, a-z 0-9 and !#$%'()*+,-./:;=?@_. The suffix should be build by concatenating the device's domain name or IP address, a colon and an application name, e.g. *192.168.0.1: myApp*.
Security policies	OPC UA supports different security policies. These essentially describe the strength of the encryption being used in the communication between server and client. Here you can enable several security policies. At least one is required. Security policies without encryption and those which use algorithms that are not considered secure anymore, should not be used in productive systems, because doing so poses a security risk.

Upload file(s)

Here you can upload various certificate files.

Authority's certificate file	This is the certificate of the certification authority (CA). It is only needed if the application instance certificates have been signed by a certification authority.
Application certificate file for SecurityPolicy - Basic128Rsa15	The application instance certificate is used to authenticate the server with the clients and to exchange the communication keys. This application instance certificate is used for the *SecurityPolicy - Basic128Rsa15*.
Application certificate's private key file for SecurityPolicy - Basic128Rsa15	This is the private key for the application instance certificate for the *SecurityPolicy - Basic128Rsa15*.
Application certificate file for SecurityPolicy - Basic256Sha256	This is the application instance certificate for the *SecurityPolicy - Basic256Sha256*.
Application certificate's private key file for SecurityPolicy - Basic256Sha256	This is the private key for the application instance certificate for the *SecurityPolicy - Basic256Sha256*.

Firstly choose the type of file. Secondly chose a file from the local file system and press the *Start upload* button.

Certificate generation

The above mentions application instance certificates can be generated with OpenSSL. The following description assumes you have basic knowledge of certificate generation. First add a new section to your OpenSSL configuration file (*openssl.cfg*). The name of this section can be chosen freely. Here we name this section *opcua_app_inst_ext*. It should have the following content:

```
[opcua_app_inst_ext]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyCer
extendedKeyUsage = serverAuth, clientAuth
subjectAltName = IP:<IP ADDRESS>, URI:<APPLICATION URI>
#subjectAltName = DNS:<DNS NAME>, URI:<APPLICATION URI>
subjectKeyIdentifier = hash
authorityKeyIdentifier = issuer
```

The *subjectAltName* entry must be adapted. In the example it exists twice. The second one is commented out. The first entry is needed if your device is only accessible through its IP address. Replace *<IP ADDRESS>* with the IP address and *<APPLICATION URI>* with the application URI. If your device is accessible through a DNS name, you should use the second entry. Comment the first entry out and the second entry in. Replace *<DNS NAME>* with the DNS name and *<APPLICATION URI>* with the application URI. Please refer to the OpenSSL documentation if your device is accessible through more than one IP address or DNS name.

The following command generates a self-signed certificate for the *SecurityPolicy - Basic256Sha256* with a validity of 365 days and an RSA key length of 4096 bits. The validity (-*days*) can be arbitrarily adapted. The key length (*rsa:4096*) can optionally be reduced to 2048 bits minimum.

```
openssl req -x509 -newkey rsa:4096 -sha256 -nodes -days 365 -outform DER \
            -out cert.der -keyout key.pem -extensions opcua_app_inst_ext
```

To generate a certificate signing request (CSR) instead of a sef-signed certificate, replace the -*x509* parameter with -*new*.

```
openssl req -new -newkey rsa:4096 -sha256 -nodes -days 365 -outform DER \
            -out cert.der -keyout key.pem -extensions opcua_app_inst_ext
```

For the *SecurityPolicy - Basic128Rsa15* the RSA key length is reduced to 1024 to 2048 bits and the hashing algorithm SHA-1 is used. Please note that this algorithm is not considered secure anymore.

```
openssl req -x509 -newkey rsa:2048 -sha1 -nodes -days 365 -outform DER \
            -out cert.der -keyout key.pem -extensions opcua_app_inst_ext
openssl req -new -newkey rsa:2048 -sha1 -nodes -days 365 -outform DER \
            -out cert.der -keyout key.pem -extensions opcua_app_inst_ext
```

Event Log Messages

The following section describes the messages that the OPC UA Server driver may log to the Event Log. All messages are prefixed with the text *OPC UA server driver:*.

• **The interval is being maintained again.**

   This message appears after this message, when the interval is being maintained again.

• **The interval has been violated.**

   The driver was still busy reading/writing values when a new cycle should be started. You should probably increase the interval time.

- **Insecure security policy enabled**

  At least one of the enabled security policies is considered insecure. Such security policies should not be used in a productive system.

- **No security policy enabled**

  None of the security policies has been enabled. To be able to communicate with the OPC UA server, at least one security policy must be enabled.

- **Missing or invalid application instance certificate for SecurityPolicy - Basic128Rsa15 (code *<code>*)**

  The application instance certificate for the *SecurityPolicy - Basic128Rsa15* is missing or invalid. The given code is a hexadecimal OPC UA error code. You have to generate a valid certificate including private key and upload it.

- **Missing or invalid application instance certificate for SecurityPolicy - Basic256Sha256 (code *<code>*)**

  The application instance certificate for the *SecurityPolicy - Basic256Sha256* is missing or invalid. The given code is a hexadecimal OPC UA error code. You have to generate a valid certificate including private key and upload it.

- **Unhandled error (code *<code>*)**

  The OPC UA server has noticed an error that it could not handle autonomously. The corresponding OPC UA error code is shown.

- **Out of memory**

  The OPC UA server doesn't have enough memory at its disposal.

Licensed Software

The Flex OPC UA SDK by Matrikon International, Inc. has been used to implement the OPC UA server interface.

Website: www.matrikonopc.com

SGIM

The SGIM (Smart Grid Interface Module) driver can be configured through this form.

Channel and Value Settings

Over this grid it is possible to configure settings for the channels and values.

Channels

There are three kind of channels:

- General: Contains values that are identical to all channels, like voltage

- CT (current transformer): Channel uses a current transformer

- RC (Rogowski coil): Channel uses a Rogowski coil

For CT channels the CT ratio can be configured. RC channels and the General channel can not be further configured.

Values

For each value a high and a low limit can be set. The limits are only functional if the Portal is enabled. When a limit is exceeded, values sampled 60 seconds before and 60 seconds after the limit event are communicated to the Portal with a special quality value of 100.
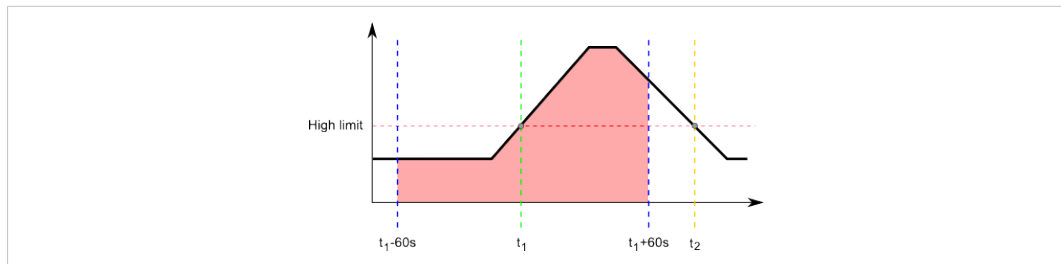
Fig. 18   **SGIM Limit example**

The above image is an example about how an SGIM limit functions. The user enables and sets the high limit for a value (red dashed line). Values are continuously saved in a ring buffer. The value reaches the limit at time t_1 and at time $t_1 + 60$ s all values marked in pink (i.e. from time $t_1 - 60$ s until time $t_1 + 60$ s, 120 seconds in total) are transmitted to the Portal with the quality "100". A new limit will be triggered only after the time-point t_2 i.e. after the limit has been reset by dipping below the high limit point.

Settings

If the "Trigger meta-values over HMS Hub element" option is enabled, SGIM meta-values (average, maximum and minimum) can only be triggered over the "Power values trigger" and "Non-power values trigger" elements for power values (Active Power, Reactive Power and Apparent Power) and non-power values (all remaining values) respectively.

Event Log Messages

The following section describes the messages that the SGIM driver may log to the Event Log. All messages are prefixed with the text *SGIM driver:*.

- **The interval has been violated.**

    The driver was still busy reading values when a new cycle should be started.

- **CAN bus initialisation error: *<Error Number>*.**

    CAn bus could not be initialised.

- **Limit event for *value name*, channel *channel number***

    Limit event occurred for value *<value name>* and channel *<channel number>*.

- **An averaging counter has overflown. Average values are no longer reliable.**

    An averaging counter has overflown. Average values are no longer reliable.

- **A CAN-controller error has occurred.**

    A CAN-controller error has occurred.

- **The CAN-controller is in the Busoff state.**

    The CAN-controller is in the Busoff state.

- **A CAN-controller overrun error has occurred.**

    A CAN-controller error has occurred.

- **The CAN receive buffer has overflowed.**

    The CAN receive buffer has overflowed

- **The CAN transmit buffer has overflowed.**

    The CAN transmit buffer has overflowed

- **The CAN controller is in the init/busoff state.**

  The CAN controller is in the init/busoff state.

- **The CAN controller is in the error passive state.**

  The CAN controller is in the error passive state.

CODESYS Network Variables

The CODESYS Network Variable functionality of the device can be configured through this form.

Variable Lists

The topmost table contains information about imported CODESYS Network Variable lists. Using the controls below you can edit or delete imported lists. For each list, in the "Name" column an entry with the name of the list or "Unnamed list" will be created, followed by the COB-ID (list index) of the imported list, and under "List Parameters" the parameters of the list will be shown. By expanding each list its containing variables will be displayed. The "Name" column contains the name, the "Type" column the type, and the "Index" column the Sub-index of each variable.

Please note that the driver only accepts valid CODESYS Network Variable lists. A valid list is a list that begins with the keyword "VAR_GLOBAL" and ends with the keyword "END_VAR". If the list is exported using the "export" function of CODESYS, additional metadata is contained in the list in the form of comments. The driver can parse this metadata and automatically complete the list settings from it. If you need to change some of those settings, please use the "Edit" button. If the data is not completed automatically you will be prompted to complete it manually. Also note that not all data types are supported by the driver. Supported data types are:

BOOL	Boolean type (one byte long)
USINT	Unsigned short integer type (one byte long)
SINT	Signed short integer type (one byte long)
UINT	Unsigned Integer type (two bytes long)
INT	Signed Integer type (two bytes long)
UDINT	Unsigned Double integer type (four bytes long)
DINT	Signed Double integer type (four bytes long)
LINT	Signed Long integer type (eight bytes long)
ULINT	Unsigned Long integer type (eight bytes long)
REAL	Float type (four bytes long)
LREAL	Double float type (eight bytes long)
BYTE	Byte type (one byte long)
WORD	Word type (two bytes long)
DWORD	Double word type (four bytes long)
LWORD	Long word typ (eight bytes long)
TIME	Unix time stamp (four bytes long)
STRING	String are supported up to a length of 255 characters.

When adding a list where no metadata could be found or editing one, a dialogue will show up where you can configure the following parameters:

Name	The name of the list
Portal Name	The name of the list in the portal
COB-ID	The list COB-ID. As per CODESYS documentation it must be a non-negative integer. Please note that each list must have a unique COB-ID across the network.
IP address	The IP address of the list. This field is only visible when the Write checkbox is selected.
Port	Defines the network port on which the variable packets will be broadcasted.

Network interface	Determines the network interface from which you want to send and receive packets.
Write	This checkbox determines if the list will be a read- or write-list. If the box is enabled, the list is a write-list, which means that the values contained in the list's variables will be broadcasted to the other devices of the network. If the list is a read-list it will poll the network for values that correspond to its variables.
Pack variables	This checkbox determines if the variables in the list will be transmitted in packed or unpacked mode. In unpacked mode, each variable is transmitted in a separate UDP, while in packed mode many variables are transmitted at the same time. Each packet can only contain up to 256 bytes of data, so each variable that exceeds this value will be broadcasted with additional packets.
Transmit checksum	Use a checksum to verify the validity of each list. Incoming packets are automatically determined if they need a checksum check or not, so this option is only used for write-lists. Please note that this checksum does not protect against data transfer errors but is only used to check the correct consistency of the list (that the correct data types are used for each variable).
Acknowledgement	Wait for an acknowledging packet from the receiver device to ensure that the transmission was successful. Please note that the acknowledgement must take place once during each cycle. Interval times that are too short might not allow the receiving device to answer in time.

Upload a CODESYS variable list

Using the button you can navigate to your exported CODESYS Network variable list. By pressing the "Upload List" button, the list will be uploaded and parsed.

Limits

You can define up to 128 lists and 4096 variables

Settings

There are several fields through which some general settings can be configured:

Interval	Defines the interval in milliseconds in which the CODESYS Network Variable driver task will be run and values for the variables will be read and written from and to the network respectively. Note that if the interval is too short to read/write all values, a warning message will be shown in the event log. In this case you should increase the interval. Please note that the timing of reading variables also depends on the rate with which the sending device is broadcasting packets.
Group variables by list	If this option is set, variables will be grouped by list. For other I/O interfaces this grouping is mandatory. For the CODESYS network variable interface is was introduced later and is thus optional to assure backward compatibility.

Event Log Messages

The following section describes the messages that the CODESYS Network Variables driver may log to the Event Log. Messages are prefixed with the text "CODESYS Network Variables driver:".

- **The interval is being maintained again.**

  This message appears after this message, when the interval is being maintained again.

- **The interval has been violated.**

  The driver was still busy reading/writing values when a new task cycle started. You probably should increase the interval time.

- **Error receiving CODESYS Network Variable pack with COB-ID *<COB-ID>*, sub-index *<Sub-Index>*. The received packet contains an invalid float value.**

  The driver has received a REAL (float) or LREAL (double) value that is invalid.

- **Error receiving CODESYS Network Variable pack with COB-ID *<COB-ID>*, sub-index *<Sub-Index>*. The received checksum is invalid.**

  The driver has received a variable packet that contains an erroneous crc32 check value.

- **Error in receiving transmission acknowledgement packet for CODESYS Network Variable packet with COB-ID *<COB-ID>*.**

    The driver requested an acknowledgement of its transmission from the receiving device but has not received one during this cycle for this packet.

- **Error sending variable in CODESYS Network Variable list with COB-ID *<COB-ID>*, variable index *<Sub-Index>*. Error code: *<Error-Code>*.**

    There was an error during packet transmission. Please contact support.

- **Variable with name: *<Variable Name>*, in list with COB-ID *<COB-ID>* (sub-index *<Sub-Index>*) has an invalid or unsupported data type.**

    The data type of the named variable is not supported by the driver. Please look at the list above for supported data types.

- **Received CODESYS Network Variable acknowledgement packet with COB-ID *<COB-ID>*, packet index *<Packet-Index>* with code Error code: *<Error-Code>*.**

    Received an acknowledgement that reports an error. To determine the error please refer to CODESYS documentation or contact support.

- **Portal communication error *<error code>***

    A general error has occurred while exchanging data with the portal. Possible error codes:

-2	Out of memory
-3	The internal data queue has overflown, because data was produced faster than it could be processed.

    If you see an error code that is not on the above list, please contact the customer support.

SNMP Manager

Your Edge gateway device features an SNMP Manager, which supports the protocol version SNMPv2c. It can be configured through this form.

Agents and Values

At the top there is a list of SNMP agents and their values. You can add, edit, delete or clone agents and values (input or output) via the buttons below the list. For each agent the list shows a name, an IP address, a portal name, a community string and an index. For each value the list shows a name, an OID, a portal name, a data type and an index. Most of these fields can be edited (see below). The indexes are used internally to address the agents and their values.

When adding or editing a agent, a dialog will show up where you can configure the following parameters:

Name	The agent's name.
Portal Name	A name used to build the path of the corresponding node on the portal.
IP Address	The agent's IP address (e.g. 192.168.0.3).
Community String	Community String used for authentication with the agent.

When adding or editing a value, another dialog will show up, the value will be added to the currently selected group (Inputs or Outputs). Here you can configure the following parameters:

Name	The value's name.
Portal Name	A name used to build the path of the corresponding node on the portal.
OID	Object ID used to address the value through the SNMP protocol.
SNMP Data Type	The SNMP protocol data type: Integer

	Primitive integer data type
	Unsigned32
	Primitive unsigned 32-bit data type
	OID
	Object ID represented as a string
	Octet String
	A sequence of octets; used to transmit strings of up to *Maximum string length* bytes
	IP Address
	An IPv4 address
	Counter32
	Primitive unsigned 32-bit data type
	Time Ticks
	Primitive unsigned 32-bit data type
Data Type	The actual data type the SNMP data type will be represented as. The options available here are determined by the selected SNMP data type.
Maximum string length	If the SNMP *Octet String* data type is selected the Maximum string length must be defined, strings greater than the Maximum length are shortened to fit.

Limits

You can define up to 128 agents and 4096 values.

Trap values

For each device there are two additional implicit input values that are not requested by the SNMP manager, but are expected to be sent from the agents with traps and inform requests. These values are:

- **sysUpTime** (OID 1.3.6.1.2.1.1.3.0): The time (in hundredths of a second) since the network management portion of the system was last re-initialized

- **snmpTrapOID** (OID 1.3.6.1.6.3.1.1.4.1.0): The authoritative identification of the notification currently being sent; the SNMP manager provides up to 256 bytes to store this OID. Longer OIDs cannot be processed.

Settings

Below the SNMP agents there are several fields through which some general settings can be configured:

Interval	Defines the interval in milliseconds in which all SNMP values will be read/written.
Timeout	Defines the maximum time in milliseconds to wait for the reply of a agent when reading/writing one of its values.
Trap Community String	Community String used by agents for authentication with the manager when sending traps or inform requests.

Event Log Messages

The following section describes the messages that the SNMP-Manager driver may log to the Event Log. All messages are prefixed with the text *SNMP manager driver:*.

- **(Wrote or Read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*).**

  A value was successfully written or read.

- **Could not (write or read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*). (error status *<x>*, error index *<x>*)**

  A value belonging to the specified agent could not be successfully read or written. Check error status for exact reason.

- **Could not (write or read) value *<value name>* (OID *<value oid>*) on agent *<agent name>* (IP *<agent address>*). (*<exception>*)**

  A value belonging to the specified agent could not be successfully read or written. Possible reasons (see *exception*):

No such object	The object with the given OID is not available on this agent.
No such instance	The instance with the given OID is not available on this agent.
End of MIB view	The end of the MIB tree has been reached. This cannot happen, because the SNMP manager does not iterate through the tree.
Timeout	Waiting for a reply from the SNMP agent has timed out.

  If you see an exception that is not on the above list, please contact the customer support.

- **Could not process value *<value name>* (OID *<value oid>*) in trap/inform request from agent *<agent name>* (IP *<agent address>*). (*<msg>*)**

  A trap/inform request delivered a value that could not be processed. The message also provides a detailed error description.

- **Could not process message from agent *<agent name>* (IP *<agent address>*). (*<exception>*)**

  A message from *agent name* could not be processed. The message also provides a detailed error description.

- **Could not send response to INFORM request from agent *<agent name>* (IP *<agent address>*). (Error code *<x>*)**

  Could not send response to INFORM request from agent. The message also shows an error code:

-2	Could not open TCP socket
-3	Could not switch TCP socket to blocking mode
-4	TCP connect failed
-5	Setting options on TCP socket failed

  If you see an error code that is not on the above list, please contact customer support.

4.2.6 Services

COM Server

The HMS Hub firmware features a COM Server according to RFC 2217 which lets you access your Edge gateway device's serial port via the network. The COM Server can be configured via this form. Besides the TCP port the server should listen on and possibly the serial interface you can set some compatibility options:

Compatibility with Cisco IOS	Respects a small bug in Cisco's IOS
Compatibility with HW Virtual Serial Port	Respects an incompatibility in the Virtual Serial Port software from HW group

Event Log Messages

The following section describes the messages that the COM server may log to the Event Log. Messages are prefixed with the text *COM Server:*.

- **Started.**

  The COM server has been started.

- **Stopped.**

  The COM server has been stopped.

- **Incoming connection from *<x>*.**

    The COM server accepted an incoming connection from the given IP address.

- **Requested unsupported parity, set to no parity.**

    The client has tried to configure a parity that is not supported. Parity will be configured to *none*.

- **Requested unsupported 1.5 bits stop size, set to 1 bit stop size.**

    The client has tried to configure a stop bit size of 1.5 bits, which is not supported. Stop bit size will be configured to 1 bit.

- **Inbound flow control ignored.**

    The client has tried to configure flow control only for inbound serial line data. This is not supported. Flow control will not be configured.

- **Requested unsupported flow control.**

    The client has tried to configure an unsupported type of flow control. Flow control will not be configured.

- **Client doesn't support Telnet COM Port Protocol Option (RFC 2217), trying to serve anyway.**

    The client indicates that it does not support the Telnet COM Port Protocol Option according to RFC 2217. The COM server will try to serve the client anyway.

- **Unable to open serial port *<x>*. Exiting.**

    The COM server was unable to open the given serial port at start-up.

- **Socket open failed. Exiting.**

    The COM server could not open a TCP socket at start-up.

- **Socket bind failed. Exiting.**

    The COM server could not bind its TCP socket to the designated port at start-up.

- **Socket listen failed.**

    The COM server could not listen on its TCP socket.

- **Socket accept failed.**

    The COM server could not accept an incoming connection on its TCP socket.

- **Error writing to network. Close socket.**

    The COM server could not send data on its TCP socket.

- **Error reading from network. Close socket.**

    The COM server could not receive data on its TCP socket.

Data Logging

Data Logging can be used to store values in the HMS Hub database. Those values can then be exported in the "**.csv**" file format.

At the top there is a list with the defined elements. It shows for every element its name, its actual value and its trigger. You can add, edit, delete or clone elements through the buttons below the list.

When adding or editing a element, a dialog will show up. Here the following parameters can be configured:

Name	The element name is used as description of the corresponding database entry.
Element	The value which will be stored in the database when this element is triggered. Multiple elements for the same value are allowed.
TriggerMode	Set the trigger mode with which values are stored to the database. • cyclic • deadband
Cycle	Is used when the trigger mode is cyclic. After this many seconds the value will be stored in the database.
Deadband	Is used when the trigger mode is deadband. A new value is only stored if the difference to the last stored value is equal or larger than the deadband. If the deadband is again met before the value is stored a message will be shown in the eventlog.

Export collected values

By clicking **Start export** a dialog will appear. There all stored values can be selected to be exported from the database to a comma separated file "**.csv**".

> ! Not only the Elements of the current configuration are available to be exported, older configurations and their values are retained until the Erase all entries in next startup checkbox is set.

In the dialog the following parameters can be configured:

Line ending	This paramater sets the line ending characters of the file. • Windows - <CR><LF> • Linux - <LF>
Separator	Separator that can be used to distinguish the values of the file. If this option is left empty the semicolon will be used as a separator.
Separate file for each element	For each element a separate export file will be created.
Timestamp in seconds resolution	Timestamps are exported in econds resolution.
Select date range	When this option is set only values in the date range are exported.
Element list	A list with all the configured elements. Each element can be select or deselected via its checkbox.

Erase all entries in next startup

By selecting this option you can remove all entries from Data Logging in the next startup, this option is reset afterwards.

Event Log Messages

The following section describes the messages that the Data Logging webservice may log to the Event Log. All Messages are prefixed with the text *Data Logging:*.

• **Online.**

   Data Logging is now online.

• **Offline.**

   Data Logging is now offline.

• **Restoring connection.**

   Data Logging is restoring connection.

• **Waiting for database connection.**

   Data Logging is waiting for the database connection to be activated.

- **Device configuration has changed.**

  Data Logging configuration file was created with a different Device configuration. Please recheck the elements and store the new configuration.

- **Value(s) could not be found.**

  Not all Data Logging configured elements could not be found. Please recheck the Data Logging configuration.

- **Could not create database.**

  Data Logging failed to create the database. The database file is write protected or malformed.

- **Data was lost.**

  The configured elements produced data quicker as the Data Logging could store.

- **Could not parse configuration file: <error>.**

  Data Logging failed to parse the provided configuration file.

- **Erasing old entries.**

  Data Logging deleted all old entries.

- **Connection lost.**

  Data Logging lost the database connection.

Database

Through this form you can configure the HMS Hub Database.

> **!** The database file itself is not encrypted. This means that physical access to the file also allows access to the values stored in it. The database has no limit. It can grow until it has exhausted all system resources and this might have a negative impact on
> the update function of the Edge gateway

Event Log Messages

The following section describes the messages that the database may log to the Event Log. All Messages are prefixed with the text *Database manager:*.

- **Online.**

  The Database is now online and can server other interfaces.

- **Offline.**

  The Database is now offline.

- **Could not open.**

  The Database file could not be opened.

- **Could not close.**

  The Database file was not finalized correctly. There might be data loss.

MicroBrowser

Your Edge gateway device has a special so-called *CGI-BIN* interface for *iniNet Solutions' MicroBrowser*. The *MicroBrowser* can use this interface to read the values of the Edge gateway device's I/Os and to read and write variables.

This form can be used to enable this interface. You can also specify whether variables can be written through the interface. And you can download a variable list file that can be imported into *iniNet Solutions' SpiderControl SCADA Editor*.

4.3 Export

Via this form you can export the configuration of your Edge gateway device. It will be packed into an update package that can be applied to other Edge gateway devices. You can find information on how to use the extracted update package on the Update help site.

Select which settings you want to export by checking the corresponding options and then click the *OK* button. After the update package has been created, it will be provided for download.

4.3.1 Event Log Messages

The following section describes the messages that the update system may log to the Event Log. Messages are prefixed with the text *Update Exporter:*.

- **Preparing export...**

  The export of the Edge gateway device's configuration is now being prepared.

- **Packing update package...**

  The configuration is now being packed into an update package.

- **Update package has been successfully packed**

  The configuration has been successfully packed into an update package.

- **Preparation failed. Message: *<message>***

  The preparation of the configuration export failed. A detailed error description is appended to the message.

- **Packing failed. Message: *<message>***

  Packing the configuration into an update package failed. A detailed error description is appended to the message.

4.3.2 Open-source Software

The following open-source software has been used to implement the export functionality:

5   Diagnostics

On the diagnostics page you can check the correct function of the device or get additional information in case of a malfunction.
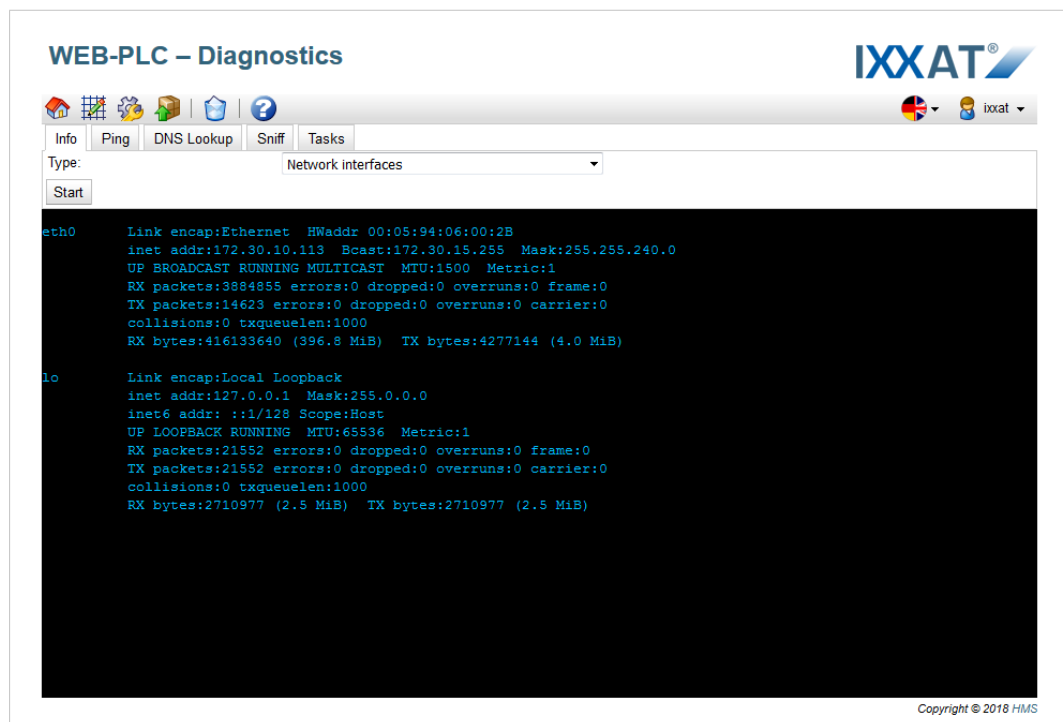

Fig. 19   Screenshot of the diagnostics page

At the top of the page there is a toolbar. In the following the toolbar's buttons will be described in detail.

🏠	Closes the diagnostics page and opens the home page.
▦	Closes the diagnostics page and opens the HMS Hub editor.
⚙	Closes the diagnostics page and opens the settings page.
📦	Closes the diagnostics page and opens the update page.
🗑	Clears the contents of the console.
❓	Opens the help system in a new window.
👤	Login button

Below the toolbar you'll find the diagnostics console.

5.1   Diagnostics Console

The Diagnostics console provides different commands that can be used to check the correct function of the device or to get additional information in case of a malfunction.

The commands can be selected and started via a tabbed control. Below there is the actual console where all commands emit their outputs.

Please note that only one command can be executed at the same time.

5.1.1 Info Command

The Info command can be executed inside the diagnostics console. It shows different information. Which information should be displayed is selected via a drop down field:

Network interfaces

Information on the available network interfaces is displayed. (Corresponds to the Linux command *ifconfig*)

Routing table

The routing table is displayed. (Corresponds to the Linux command *route -n*)

DNS configuration

The DNS configuration is displayed. (Corresponds to the content of the file */etc/resolv.conf* on Linux)

The command is started via the *Start* button.

5.1.2 Ping Command

The Ping command can be executed inside the diagnostics console. It sends ping packets to another host in the network and evaluates the answers. The following options can be configured:

Host	The name or the IP address of the host that should be pinged
Packet count	Defines how many packets will be sent. The value must be between 1 and 2,147,483,647. Please note that the command cannot be aborted.
Data bytes	Defines how many data bytes are sent along with every packet. The value must be between 1 and 65,507.

The command is started via the *Start* button.

5.1.3 DNS Lookup Command

The DNS lookup command can be executed inside the diagnostics console. It tries to resolve a host name into an IP address. The following options can be configured:

Host	Host name that should be resolved
Name server IP address	Here the IP address of a name server can be provided. If the field is left empty, the name servers stored inside the device will be used.

The command is started via the *Start* button.

5.1.4 Sniff Command

The Sniff command can be executed inside the diagnostics console. It captures network packets. When started it runs in the background. This means that you can start other commands in the meantime. The captured packets are temporarily stored on the SD card. After the capture has been stopped the created *.cap* files will be automatically downloaded. They can for example be opened with the program *Wireshark*. A separate file will be created for every network interface. Via a drop down field a certain network interface can be selected on which packets will be captured.

The command is started via the *Start* button. It is stopped via the *Stop* button.

5.1.5 Tasks Command

The tasks command can be executed inside the diagnostics console. It shows information on the tasks running inside the system. There are two different output types:

RTOS tasks

Shows the task list in RTOS format. This output type corresponds to the @CHIP-RTOS' tasks command. The list shows each task's ID, the CPU load, the priority, the stack size and usage, the task status bitmap and possibly the name of the parent task. The list is sorted by task priority. At the end of the list the total CPU usage is shown. This does not work before the second call.

Linux tasks

Shows the task list in Linux format. This output type corresponds to the Linux command *top -n1*.

The command is started via the *Start* button.

5.1.6 Database Command

The database command can be executed in the diagnostics console. It displays information from the database running in the system.

Database information: Returns the database info headers.

The command is started via the *Start* button.

6   Update

On this page you can apply update packages. Update packages can contain either a new firmware version or an exported configuration.
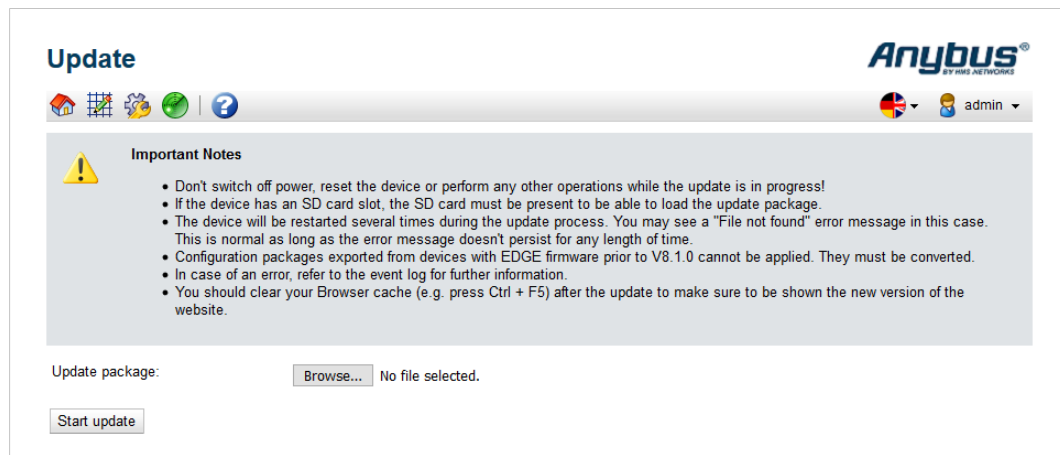

Fig. 20   Screenshot of the update page

At the top of the page there is a toolbar. In the following the toolbar's buttons will be described in detail.

	Closes the update page and opens the home page.
	Closes the update page and opens the HMS Hub editor.
	Closes the update page and opens the settings page.
	Closes the update page and opens the page.
	Opens the help system in a new window.
	Login button

Below the toolbar there is the *Update* form.

6.1   Update

Via this form you can apply update packages.

To start an update select the update package (*.cup) and click on *Start update*.

The update package will first be stored on the SD card that must be present in the slot of your Edge gateway. At start-up a directory named *Edge gateway* has been created on the card. This directory will be used to store the contents of the update package. If you have plugged in the SD card after start-up, the directory might not exist and the upload will fail. Please reboot the device with the SD card plugged in in this case.

> ! While accessing the SD card, the device may not be able to update the physical I/Os.

After the update package has been completely uploaded to the SD card it will be programmed. While the device is being programmed, the website may not be able to contact it. Thus communication errors might be reported. This is normal as long as the error message does not persist for any length of time. After the update has been programmed you will be forwarded to the index page.

> **!** Note that your device will not work anymore if you reset it while the update is being programmed.

6.1.1 Remote update

Software updates may also be distributed via the portal. The HMS Hub firmware creates three nodes on the portal via which you can trigger and monitor the remote update. The update package will be downloaded from any HTTP server.

/device-info/ update/url	Write the URL of the update package (e.g. http://www.company.com/download/firmware.cup) to this node before starting the remote update.

/device-info/ update/status	This node is used to trigger and monitor the update process. Write one of the trigger values (see below) to this node in order to start the update. The node will then show the current state of the update process:
	
	• 0 – No update has been started. This is also the state you will see, after the update has been applied and the device has restarted.
	
	• 1 – Must be written to trigger download and installation of a remote update.
	
	• 9 – Must be written to trigger download of an update package from a remote web server. After the update package has been downloaded, the state will change to 11. The installation of the update package must then be triggered by writing 10.
	
	• 10 – Must be written to trigger installation of an update package downloaded from a remote web server after the state has changed to 11.
	
	• 2 – The update package is being loaded via the local web server.
	
	• 3 – The update package is being loaded from a remote web server.
	
	• 11 – The update package has been successfully downloaded from a remote web server. Installation of the update package must now be started by writing 10.
	
	• 4 – The update package is being checked.
	
	• 5 – The update package is being unpacked.
	
	• 6 – The update package is being programmed.
	
	• 7 – The system is being cleaned up.
	
	• 8 – The update module is shutting down.
	
	• -1 – Error: Loading the update package failed.
	
	• -2 – Error: The update package did not pass the check. Reasons may be:
	
	  – Illegal update package
	
	  – The update package was intended for another firmware.
	
	  – The update package required another existing firmware version.
	
	• -3 – Error: Unpacking the update package failed.
	
	• -4 – Error: Programming the update package failed.
	
	• -5 – Error: Cleaning up failed.
	
	• -6 – Error: Loading the update package failed, because the target directory does not exist. Maybe an external medium is missing or write-protected.
	
	• -7 – Error: Loading the update package failed, because the target file already exists, but could not be overwritten. Maybe an external medium is write-protected.
	
	• -8 – Error: Loading the update package failed. Socket error.
	
	• -9 – Error: Loading the update package failed. Could not connect to server.
	
	• -10 – Error: Loading the update package failed. SSL/TLS error.
	
	• -11 – Error: Loading the update package failed. Could not send request.
	
	• -12 – Error: Loading the update package failed. Could not receive reply.
	
	• -13 – Error: Loading the update package failed. Internal buffer is too small.
	
	• -14 – Error: Loading the update package failed. Could not resolve host name.
	
	• -15 – Error: Loading the update package failed. File I/O error.
	
	• -16 – Error: Loading the update package failed. Protocol error.
	
	• -17 – Error: Loading the update package failed. Timeout.
	
	• -18 – Error: Loading the update package failed. Out of memory.

/device-info/ update/counter	This node holds a counter that simply will be incremented on each successful update. So if you apply an update package without any visible effect - such as a change in the firmware version - you can use this counter to check if the update package has been successfully applied.

6.1.2 Event Log Messages

The following section describes the messages that the update system may log to the Event Log. All Messages are prefixed with the text *Update Loader:* or *Update Portal:*.

- **Running**

  The update loader is now running.

- **Package successfully loaded via local web server**

  An update package has been successfully uploaded via the local web server.

- **Loading package from remote web server...**

  An update package is being downloaded from a remote web server.

- **Package successfully loaded from remote web server**

  An update package has been successfully downloaded from a remote web server.

- **Checking package...**

  An update package has been successfully loaded. The package is now being checked for correctness and compatibility.

- **About to Reboot...**

  The Edge gateway will now reboot in order to start the programming of the current update package.

- **Loading package via local web server timed out**

  A timeout occurred while an update package was uploaded via the local web server. The upload is being aborted.

- **Target directory for remote update package does not exist**

  Loading an update package from a remote web server failed, because the target directory does not exist. Maybe an external medium is missing or write-protected.

- **Target file for remote update package is not accessible**

  Loading an update package from a remote web server failed, because the target file already exists, but could not be overwritten. Maybe an external medium is write-protected.

- **Loading package from remote web server failed (code *<x>*)**

  Loading an update package from a remote web server failed. The message shows a code, which is either a positive HTTP response code or a negative error code from the HTTP client:

-1	An invalid URL was provided.
-2	Opening TCP socket failed
-3	Setting TCP socket options failed
-4	Connection could not be established
-5	Connection refused by remote web server
-6	SSL client could not be started
-7	SSL handshake failed
-8	Sending data to remote web server failed
-9 or -10	Receiving data from remote web server failed
-11	Internal buffer is too small
-12	IP address of remote web server could not be determined
-13	File input/output failed
-14	Invalid HTTP header format
-15	SSL session could not be created/closed
-18	Operation timed out
-19	Out of memory

  If you see an error code that is not on the above list, please contact the customer support.

- **The type of the package's signature does not match the expected type of signature. (*<signature type>*)**

  Update packages are signed. From time to time new types of signatures are introduced. If the signature type of the update package does not match the signature type expected by the firmware, the update package was probably generated for/by an older or newer firmware version. It may be possible to convert the update package to the expected format using a special tool.

- **Invalid package flag(s) (*<flags>*)**

  The current update package has invalid flag(s) set. The message shows the package flags bitmap as a hexadecimal number.

- **Firmware name does not match. This package is for "*<firmware name>*" firmware.**

  The current update package is not intended for the HMS Hub firmware. The firmware that the package is intended for is shown.

- **This package cannot be applied to the current firmware version.**

  The current update package cannot be applied to the current firmware version, because the package requires a minimum or maximum existing firmware version and the current firmware version does not match.

- **Target name does not match. This package is for "*<target name>*" target.**

  The current update package is intended for another target hardware. The target that the package is intended for is shown.

- **Variant name does not match. This package is for "*<variant name>*" variant.**

  The current update package is not intended for this variant of the HMS Hub firmware. The variant that the package is intended for is shown.

- **This package is limited to the device with the MAC address *<MAC address>*.**

  The current update package is intended for a certain device. The message indicates the MAC address of the device the package is intended for.

- **Invalid package signature**

  Update packages are signed. If an invalid package signature is indicated, there can be two reasons:

  - The update package has been corrupted or manipulated.

  - The update package uses a different signature type than expected by the existing firmware. Maybe the update package was intended for an older firmware version. Ask the issuer of the update package for an updated version.

- **Could not open package. Message: *<message>***

  The current update package could not be opened. The message includes a more detailed description of the problem.

- **Update disallowed by application (*<code>*)**

  The programming of the current update package could not be started, because the application does not allow it in its current state. The indicated code may provide further information about the reason why the update was disallowed.

- **Could not launch update.**

  The programming of the current update package could not be started. This is not a problem of the update package.

- **Portal communication error *<x>***

  A communication error has occurred while exchanging data with the portal. Possible error codes:

-10	Out of memory
-12	The internal data queue has overflown, because data was produced faster than it could be processed.

  If you see an error code that is not on the above list, please contact the customer support.

6.2   Offline Update

It is also possible to apply update or configuration packages without connection to the portal and without access to the local website of the Edge gateway device. The package must be located on the SD card in the *Edge gateway* directory and named *update.cup*. The update is started using the reset button. The position of the reset button is described in the start-up guide that is enclosed with your Edge gateway device. It can e.g. be pressed using a paper clip. Proceed as follows:

- Attach the Edge gateway device to the power supply and wait until it has started up.

- Press the reset button.

- Wait for the Edge gateway device to acknowledge the start of the update by flashing all LEDs twice.

- Wait until the update has been applied. Don't detach the Edge gateway device from the power supply before.

7   System Start-up

At start-up, the HMS Hub firmware will go through several steps in order to bring the system into operation.

- Check if there is an update package to be installed

- Check for factory reset request

- Load configuration data

- Initialise I/O hardware

- Initialise PLC and sidekick components (Cellular Modem driver, Firewall etc.)

- Load program

- Set up web services

- Start program

- Supervise components

7.1   Event Log Messages

The following section describes the messages that the firmware may log to the Event Log at start-up or possibly at a later time. All Messages are prefixed with the text *Run Time System:*.

- **Started**

  The firmware has just been started.

- **Portal communication module initialised**

  The portal communication module has been successfully initialised.

- **I/O hardware manager initialised**

  The I/O hardware manager that manages the physical I/O hardware has been successfully initialised.

- **NAT/firewall service initialised**

  The NAT and firewall service has been successfully initialised.

- **Ethernet (1) driver started.**

  The driver for the second Ethernet interface has been successfully started.

- **Cellular modem driver initialised**

  The cellular modem driver has been successfully initialised.

- **Remote Update deactivated**

  The remote update functionality is deactivated.

- **Offline Update deactivated**

  The offline update functionality is deactivated.

- **Update module initialised**

  The update module, which manages the loading and programming of update packages, has been successfully initialised.

- **PLC driver initialised**

  The PLC driver has been successfully initialised.

- **Message manager initialised**

  Der Nachrichten-Manager, der das Senden und Empfangen von Nachrichten handhabt, wurde erfolgreich initialisiert.

- **DHCP server initialised**

  The DHCP server has been successfully initialised.

- **Database Manager initialised**

  The Database Manager has been successfully initialised.

- **OpenVPN client initialised**

  The OpenVPN client has been successfully initialised.

- **Status web service initialised**

  The status web service, which provides the data to be displayed at the home page has been successfully initialised.

- **System time manager initialised**

  The system time manager has been successfully initialised.

- **COM Server initialised**

  The COM server has been successfully initialised.

- **Program loaded**

  A program has been successfully loaded.

- **WLAN driver initialised**

  The WLAN driver has been successfully initialised.

- **Web configuration modules initialised**

  The web configuration modules, that accept configuration data from the settings website, have been successfully initialised.

- **Update exporter module initialised**

  The update exporter module, which exports the configuration into an update package, has been successfully initialised.

- **Update web service initialised**

  The update web service, which manages the upload of update packages via the local web server, has been successfully initialised.

- **Running**

  The run-time system has finished initialisation.

- **Starting update from external medium.**

  The user has triggered an update from the external medium.

- **Link detected at Ethernet (*<interface index>*), Port (*<port index>*).**

  The Ethernet interface with the given index has been connected to the network.

- **Starting DHCP configuration at Ethernet (*<interface index>*), Port (*<port index>*).**

  The run time system is trying to get a new IP configuration for the Ethernet interface with the given index from a DHCP server.

- **DHCP configuration renewed at Ethernet (*<interface index>*).**

  The run time system has renewed its IP configuration for the Ethernet interface with the given index from the DHCP server.

- **DHCP configuration completed at Ethernet (*<interface index>*).**

  The run time system has gotten a new IP configuration for the Ethernet interface with the given index from a DHCP server.

- **Could not initialise power fail handling.**

  The power fail handling could not be initialised. It will not be possible to store data retentively.

- **Power fail condition present at start-up**

  A power fail condition was already present at start-up. Probably the device's supply voltage is insufficient.

- **Retentive data could not be loaded.**

  Retained data (e.g. portal variables) could not be loaded although the device is equipped with a power fail circuit. This message is normal at the very first start-up of the device. If it occurs later, there is probably a problem with the power fail circuit.

- **Not enough power fail capacity**

  The device is equipped with a power fail circuit, but the capacity supplied is not sufficient to store data (e.g. portal variables) retentively. This should not happen.

- **Power fail handling not supported**

  The device is not equipped with a power fail circuit. Data (e.g. portal variables) cannot be stored retentively.

- **The internal flash drive seems to be weak.**

  A write operation to the internal flash drive needed to be retried several times. This is an indication for the flash drive reaching the end of its life.

- **Flash write error. The internal flash drive is probably defect.**

  A write operation to the internal flash drive failed. The flash drive seems to have reached the end of its life. The device is no longer usable.

- **Could not create directory structure on external medium**

  The required directory structure could not be created on the external medium.

- **One or more certificates could not be loaded.**

  At least one of the SSL certificate files could not be loaded.

- **Could not initialise NAT/firewall service. (*<error code>*)**

  The NAT and firewall service could not be initialised. The message also shows an internal error code returned by the service's initialisation routine.

- **Could not start Ethernet (1) driver.**

  The driver for the second Ethernet interface could not be started.

- **Unsupported Ethernet (1) interface type. (*<interface type>*)**

  The HMS Hub firmware could not find a driver for the second Ethernet interface. The message also shows the numerical Ethernet interface type.

- **Could not initialise NAT/firewall service. (*<error message>*)**

  The NAT and firewall service could not be initailised. The message also shows an error message returned by the service's initialisation routine.

- **Could not initialise cellular modem driver: *<error message>***

  The cellular modem driver could not be initialised. The message also shows an error message returned by the driver's initialisation routine.

- **Could not initialise WLAN driver: *<error message>***

  The WLAN driver could not be initialised. The message also shows an error message returned by the driver's initialisation routine.

- **Could not initialise LCD: *<error message>***

  The LCD could not be initialised. The message also shows an error message returned by the LCD driver.

- **Could not initialise update module: *<error message>***

  The update module, which manages the loading and programming of update packages, could not be initialised. The message also shows an error message returned by the module's initialisation routine.

- **Could not initialise DHCP server.**

  The DHCP server could not be initialised.

- **Could not initialise Database Manager.**

  The Database Manager could not be initialised.

- **Cannot monitor external medium insertion**

  It is not possible to monitor insertion of external media. Normally the HMS Hub creates some directories on insertion of an external medium. This will not be possible.

- **Could not initialise OpenVPN client: *<error message>***

  The OpenVPN client could not be initialised. The message also shows an error message returned by the client's initialisation routine.

- **Could not initialise system time manager: *<error message>***

  The system time manager could not be initialised. The message also shows an error message returned by the manager's initialisation routine.

- **This device is not licenced for HMS Hub.**

  The device does not have a valid HMS Hub licence. The PLC runs in demo mode and will be stopped after a runtime of four hours.

- **Could not initialise COM server: *<error message>***

  The COM server could not be initialised. The message also shows an error message returned by the server's initialisation routine.

- **Could not initialise portal Event Log service: *<error message>***

  The portal Event Log service could not be initialised. The message also shows an error message returned by the service's initialisation routine.

- **Could not load and start program.**

  The program could not be loaded and thus the PLC could not be started. Probably no program (diagram and/or I/O mapping) has been saved.

- **Could not load and start program: *<error message>***

  The program could not be loaded and thus the PLC could not be started. An error occured while loading the program. An appropriate error message is included in the event message.

- **Not starting program due to problems with retentive data**

  The PLC is not being started automatically, because problems occurred while loading retentive variables or while initialising the power fail functionality. The program could be operating on invalid data.

- **Could not initialise web configuration modules: *<error message>***

  The web configuration modules, that accept configuration data from the settings website, could not be initialised. The message also shows an error message returned by the modules' initialisation routines.

- **Could not initialise update exporter module: *<error message>***

  The update exporter module, which exports the configuration into an update package, could not be initialised. The message also shows an error message returned by the module's initialisation routine.

- **Could not initialise update web service: *<error message>***

  The update web service, which manages the upload of update packages via the local web server, could not be initialised. The message also shows an error message returned by the service's initialisation routine.

- **Could not initialise diagnostics console: *<error message>***

  The diagnostics console could not be initialised. The message also shows an error message returned by the diagnostics consoles's initialisation routine.

- **Could not start update from external medium.**

  The user has triggered an update from the external medium, but the update could not be started.

- **External medium update file could not be found.**

  The user has triggered an update from the external medium, but the update package file could not be found.

- **Link lost at Ethernet (*<interface index>*), Port (*<port index>*).**

  The Ethernet interface with the given index has been disconnected from the network.

- **This RTOS version is not supported. Version (*<version number>*) is required.**

  The current RTOS version is not the version required, please install the version specified in the Event Log.

- **This bootloader version is not supported. Version (*<version number>*) is required.**

  The current bootloader version is not the version required, please install the version specified in the Event Log.

- **More than one clock synchronisation mechanism is active.**

  More than one clock synchronisation mechanism is active. It is recommended to activate only one clock synchronisation mechanism.

- **Device is secured with the default password, please change it.**

  This message is shown when the default password is still used. For security purposes, it is recommended to change the password. You can do so through the login settings.

- **Fatal error: <error message>**

  A fatal error prevented firmware start-up. The message also shows an error message describing the error.

8   Status LEDs

Your Edge gateway device has five status LEDs:

8.1   PLC

This LED indicates the status of the PLC:

Off:

No program loaded, PLC not running.

Slowly blinking (0.5 Hz):

Program loaded, PLC not running.

On:

Config Application loaded, Config Application running.

Fast blinking (4 Hz):

The task cycle of the PLC has been violated at least once. Further information can be found here.

8.2   Portal

This LED indicates the status of the portal connection:

Off:

The portal connection is not enabled.

Slowly blinking (0.5 Hz):

The connection is being established.

On:

The connection has been established.

8.3   GSM

This LED indicates the status of the cellular modem connection if a cellular modem is installed on your device :

Off:

The cellular modem connection is not enabled.

Slowly blinking (0.5 Hz):

The connection is being established.

On:

The connection has been established.

Fast blinking (4 Hz):

A fatal error has occurred. The cellular modem driver no longer tries to establish a connection.

8.4   WiFi

This LED indicates the status of the WLAN connection:

Off:

The WLAN connection is not enabled.

Slowly blinking (0.5 Hz):

The connection is being established.

On:

The connection has been established.

Fast blinking (4 Hz):

A fatal error has occurred. The WLAN driver no longer tries to establish a connection.

8.5   Bluetooth

This LED indicates the status of the Bluetooth connection:

Off:

There is no active connection and the module is unpairable.

Slowly blinking (0.5 Hz):

The module is pairable.

On:

There is one or several active connections.

Fast blinking (4 Hz):

A fatal error has occurred. The Bluetooth driver no longer tries to establish a connection.

8.6   Power

This LED indicates whether the Edge gateway device is supplied with power.

9 Factory Reset

Should you ever be unable to access your Edge gateway device (e.g. because your forgot the password), you can perform a factory reset through the reset button. The position of the reset button is described in the start-up guide that is enclosed with your Edge gateway device. Proceed as follows:

• Detach the Edge gateway device from the power supply.

• Hold down the reset button.

• Attach the Edge gateway device to the power supply.

• Wait for the Edge gateway device to acknowledge the reset by flashing all LEDs twice.

• Release the reset button.

• Wait until the Edge gateway device has started up. Don't detach it from the power supply before.

10   Recommended Browsers

We recommend that you always use the current version of browsers to make sure that the web-pages of the HMS Hub firmware are properly displayed. If you use older browsers, you might notice that some functions and features stop working. The following browsers are recommended:

- Mozilla Firefox

- Google Chrome

- Microsoft Edge

- iOS Safari

- Microsoft Internet Explorer (with reservations)

11 Open Source Software

11.1 Open Source Software

11.1.1 Used Components

The Edge gateway firmware uses the following open source components. The changed source codes of those components are available on written request to sales@beck-ipc.com. Sources of open-source components that are licenced under GPL or LGPL can be downloaded directly. A package with these sources can be found next to each software update in the download section of our website.

Das U-Boot

Project website: denx.de/wiki/U-Boot Licence: U-Boot licence

Linux Kernel

Project website: kernel.org Licence: Linux Kernel licence

Cryptodev-linux

Project website: cryptodev-linux.org Licence: Cryptodev-linux licence

BusyBox

Project website: busybox.net Licence: BusyBox licence

GNU C Library (glibc)

Project website: gnu.org/software/libc Licence: glibc licence

OpenSSL

Project website: openssl.org Licence: OpenSSL licence

lighttpd

Project website: lighttpd.net Licence: lighttpd licence

Pure-FTPd

Project website: www.pureftpd.org Licence: Pure-FTPd licence

Dropbear SSH

Project website: matt.ucc.asn.au/dropbear/dropbear.html Licence: Dropbear SSH licence

E2fsprogs: EXT2/3/4 File System Utilities

Project website: e2fsprogs.sourceforge.net Licence: EXT2 file system utilities licence

dosfstools

Project website: github.com/dosfstools/dosfstools Licence: dosfstools licence

libpcap

Project website: tcpdump.org Licence: libpcap licence

libsocketcan

Project website: pengutronix.de/cgit/tools/libsocketcan Licence: libsocketcan licence

libfuse

Project website: github.com/libfuse/libfuse Licence: libfuse licence

Netlink Protocol Library Suite (libnl)

Project website: www.infradead.org/~tgr/libnl Licence: LGPL

binutils Runtime Library

Project website: gnu.org/software/binutils Licence: binutils runtime library licence

GCC Runtime Library

Project website: gcc.gnu.org/onlinedocs/gccint/Libgcc.html Licence: GCC runtime librariy licence

GDB Server

Project website: gnu.org/software/gdb Licence: GDB server licence

AutoIP

Project website: - Licence: AutoIP licence

CIOPFS

Project website: brain-dump.org/projects/ciopfs Licence: CIOPFS licence

openresolv

Project website: roy.marples.name/projects/openresolv Licence: openresolv licence

wpa_supplicant & hostapd

Project website: w1.fi Licence: BSD licence

iptables

Project website: www.netfilter.org Licence: iptables licence

rng-tools

Project website: sourceforge.net/projects/gkernel Licence: GPL

pppd

Project website: ppp.samba.org Licence: pppd licence

BlueZ

Project website: www.bluez.org Licence: BlueZ licence

D-Bus

Project website: dbus.freedesktop.org Licence: D-Bus licence

Expat XML Parser

Project website: libexpat.github.io Licence: Expat licence

glib

Project website: developer.gnome.org/glib Licence: glib licence

libffi

Project website: sourceware.org/libffi Licence: libffi licence

zlib

Project website: zlib.net Licence: zlib licence

tslib

Project website: www.tslib.org Licence: LGPL

Opendnp3

Project website: www.automatak.com/opendnp3 Licence: Apache License

Asio

Project website: think-async.com Licence: Boost Software License

NetBIOS over TCP/IP name registration (NBTNS)

Project website: ubiqx.org/cifs Licence: LGPL

OpenVPN

Project website: openvpn.net Licence: OpenVPN licence

sredird

Project website: freecode.com/projects/sredird Licence: GPL

WLAN driver modules

Project website: www.redpinesignals.com Licence: Redpine licence

libarchive

Project website: libarchive.org Licence: libarchive licence

Eclipse Paho MQTT C client library

Project website: www.eclipse.org/paho/clients/c Licence: Eclipse Distribution Licence

SQLite

Project website: www.sqlite.org Licence: SQLite Licence

Dojo Toolkit

Project website: dojotoolkit.org Licence: Dojo Licence

FileSaver.js

Project website: github.com/eligrey/FileSaver.js Licence: FileSaver.js licence

clipboard.js

Project website: clipboardjs.com Licence: clipboard.js licence

LZ4

Project website: lz4.github.io/lz4/ Library Licence: LZ4 library licence Executable Licence: LZ4 executable licence