Soft & Control Technology s.r.o.

K2config

User manual

Document version 1.0
Revision 1.9
Dated 28th of April 2010

Content

1 K2config

1.1 Description

Application **K2config** is primary intended for direct configuration of **BaWiT** telemetry systems.

1.2 System Requirements

- **Supported operating systems:** Windows Server 2003; Windows Server 2008; Windows 7; Windows Vista; Windows XP Service Pack 2
- **Processor:** 400 MHz Pentium processor or equivalent (minimum); 1GHz Pentium processor or equivalent (recommended)
- **RAM:** 96 MB (minimum); 256 MB (recommended)
- **Hard Disk:** Up to 500 MB for Microsoft .NET Framework; up to 50 MB for K2config
- **Display:** 800 x 600, 256 colors (minimum); 1024 x 768 high color, 32-bit (recommended)
- **Installed frameworks:** Microsoft .NET Framework 3.5 SP1, Microsoft Chart Controls For .NET Framework 3.5 SP1

1.3 Installation

It is required to install *Microsoft .NET Framework 3.5 SP1* and *Microsoft Chart Controls For .NET Framework 3.5 SP1* before first application start. Installers for required *Microsoft .NET Framework libraries* are located on **K2config** installation CD (in the directory **dependencies**) or can be downloaded from Microsoft pages. It is advised to update your operating system after Framework install. K2config installer offers the installation of required *Microsoft .NET Framework*.
K2config installer is located on the installation CD in the root directory and its file name is **K2configSetup.exe**.
Application is looking for device configuration structure file at every start. If the file is not found, user is prompted to download required files from the Internet (see 5.1).
Application stores all user settings in directory **My Documents\SCT\K2config** (exact name depends on operating system language version).

2 Basic Operations

2.1 User Interface

Every information or available tool is displayed in application windows. Every application window can be moved or docked arbitrarily in the main application window by mouse. K2config saves settings of actual windows position and dock at the application end.
User can use workspaces to quickly switch between different window layouts. Workspace functions are located in the *Window* menu.

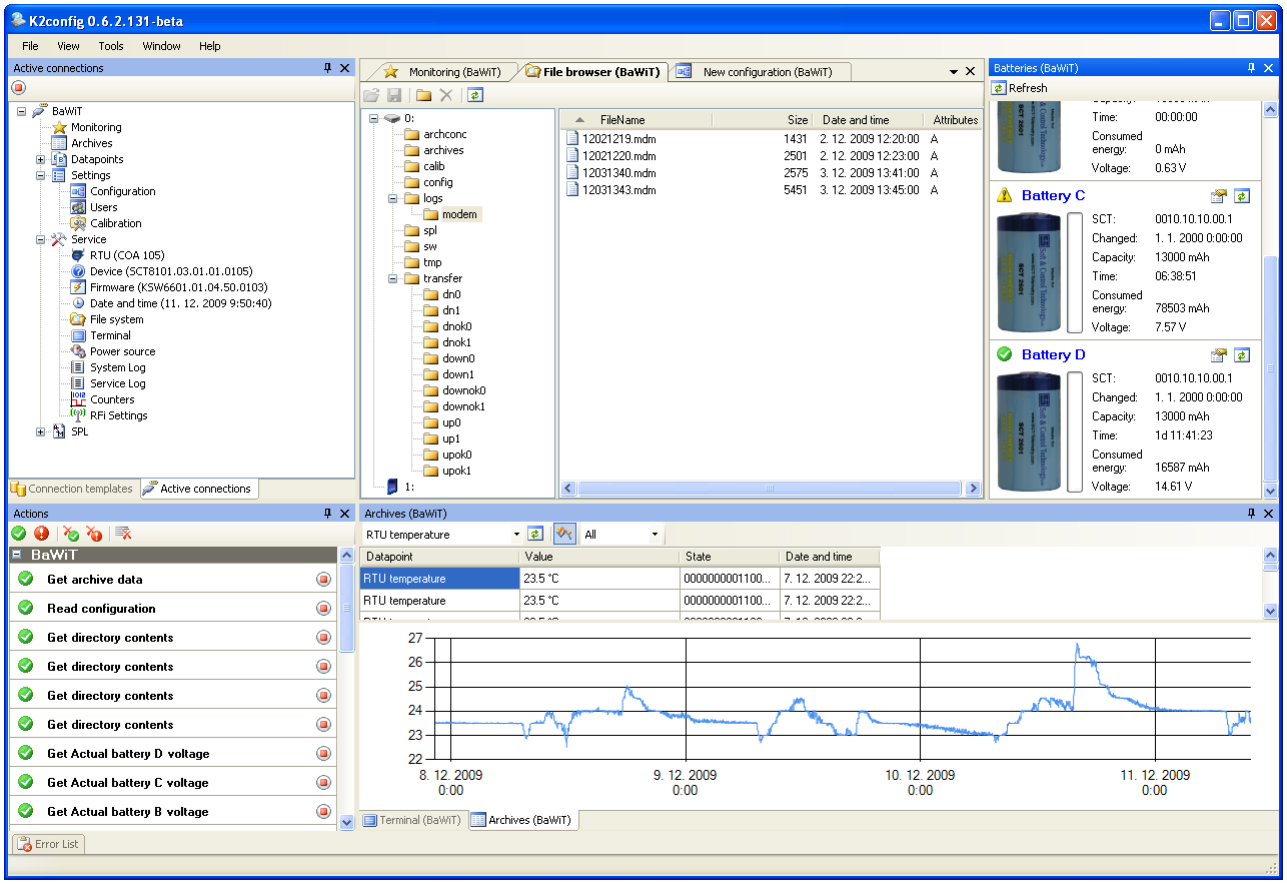

Figure 1 Sample window layout

2.2 Application Main Menu

Main menu of application is divided into 5 menus – *File*, *View*, *Tools*, *Window* and *Help*.
Menu *File*:

- **New** – used to create new empty configuration or SPL ASM source file
- **Open** – reads specified file from disk
- **Close** – closes active document
- **Close all** – closes all opened documents
- **Save** – saves actual document contents to disk using its actual filename
- **Save as** – saves actual document contents to disk prompting for a new filename
- **Save all** – saves all opened and changed documents to disk
- **Recent files** – history of recently opened files. Click to read file from disk.
- **Exit** – quit application

Menu *View* contains these functions:

- **Active connections** – opens window, that contains user opened active connections (see 2.3)
- **Actions** – opens window, that is used for displaying progress and history of executed and actually executing actions on all opened connections
- **Connection templates** – opens window used for connection templates management (see 2.3)
- **Error list** – opens window that displays information, warnings and errors from SPL ASM compiler

Menu *Tools*:

- **Configuration Templates Manager** – used for administration of configuration and monitoring templates
- **Options** – user and system settings

Menu *Window* contains workspace management commands:

- **Open workspace** – read workspace from file
- **Save workspace** – saves workspace to file using its current name
- **Save workspace as** – save workspace to file prompting user for a new name

Menu *Help*:

- **K2config User's Manual** – opens K2config user's manual using associated application for opening PDF files
- **BaWiT User's Manual** – opens BaWiT user's manual using associated application for opening PDF files
- **SPL ASM Programmer's Manual** – opens SPL ASM programmer's manual using associated application for opening PDF files
- **Check for updates** – check for new K2config updates
- **About K2config** – show application version information

2.3 Creating and Opening Connection

For successful communication with connected device one must configure connection parameters first. This functions are available in the *Connection templates* window (see Figure 2), that can be opened from *View* menu. To create new connection one can use function *Connect*. Function opens new window where user must enter serial connection parameters. These parameters will not be stored and before reopening the same connection, all parameters must be entered again.
To create new template one can use function *Add*. By using this command user can create many different connection templates that can be later quickly used by double clicking on the template caption or by selecting the template and pressing button *Connect*.

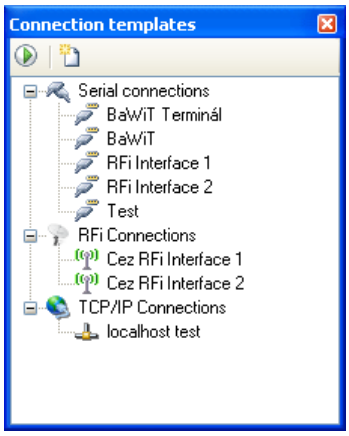

Figure 2 Connection templates management window

After the successful connection opening, application will display *Active connections* window (see Figure 3), which contains tree with all opened connections and under every connection node are available functions for connection.

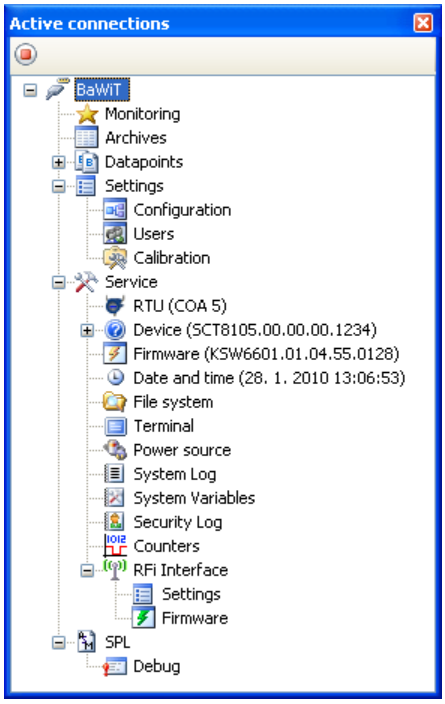

Figure 3 Active(opened) connections window

Immediately after the connection opening, SCT number (device hardware unique identifier), firmware version (KSW), and device actual date and time are read.

2.3.1 Connection settings

This chapter contains recommended default communication parameter settings. All settings are also part of the application and can be easily applied from connection template properties window (see Figure 4).

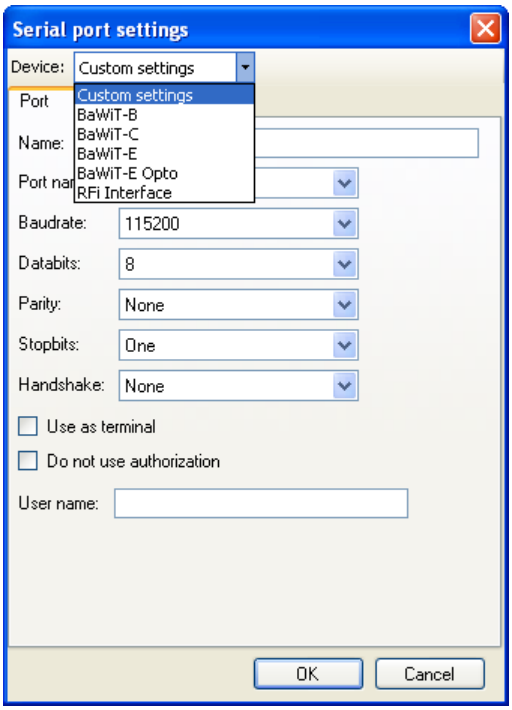

Figure 4 Connection template properties

Common serial port settings for all connection types

Port:
Databits: 8
Parity: None
Stopbits: One
Handshake: None
Layers:
Maximum LEN size: 1
Max sent: 1
Max received: 1
Max errors: 5
Timeout: 500
Retry timeout: 5000
INIT received packets: 1
INIT Timeout: 1000
IOA Size: 2

Serial connection to BaWiT-B/E devices

Port:
Baudrate: recommended max. 115200 Bd
Layers:
Maximum data segment length: recommended max. 200

Optical serial connection to BaWiT-E

Port:
Baudrate: 9600 Bd
Layers:
Maximum data segment length: recommended max. 200

Serial connection to RFi Interface

Port:

Baudrate: recommended max. 115200 Bd
Layers:
Maximum data segment length: 40

RFi connection to BaWiT-B/D/E devices

Port:
Baudrate: recommended max. 115200 Bd
Layers:
Maximum data segment length: 40
RFi:
Baudrate: recommended 4800 Bd
Channel and final power output settings are dependent on local country norms.

TCP/IP connection

Layers:
Check Use preamble
Maximum data segment length: recommended max. 200

2.3.2 Authorization

Some devices require user authorization before they can be configured. User login can be entered as a one parameter in connection template properties window. User will be prompted for password (see Figure 5) immediately after the connection opening.

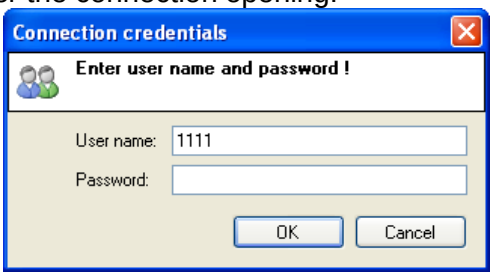

Figure 5 Login window

2.4 Device Current Date and Time

Functions for date and time operations are located in popup menu for *Date and time* node in *Active connections* window. In popup menu are functions for reading actual device date and time, setting device date and time according to computer date/time or any user entered date and time.

2.5 Firmware

Firmware node's popup menu contains functions for getting firmware version information, flashing new firmware, restarting the device and changing the work mode (for description of individual work modes see technical specification).
Node *Firmware* is a drag & drop node. So it is possible to drag specific file (e.g. from Windows Explorer window) and start firmware flashing.
One must take care, because K2config has no resources to check whether selected firmware is suitable for the device.

2.6 Editing Device Configuration

2.6.1 Creating New Configuration

New device configuration can be created by using the *New* command in the *File* menu. In opened window user must choose specific device type for which he intends to edit configuration.

2.6.2 Reading Configuration from the Device

To read or open the device configuration for editing user should use functions from *Configuration* node's popup menu in the *Active connections* window. In this menu are functions for reading and

opening the configuration for editing, only for reading the configuration (useful for displaying data points caption in the monitoring and archives windows) and for writing any configuration from file.

2.6.3 Writing Configuration into Device

To write opened configuration into device click on button *Write*. This button is available only if the configuration was read from device (not from file in computer).
To write any configuration file into device one can use respective function located in popup menu of *Configuration* node in *Active connections* window.
Configuration node is also **drag & drop** node, so it is possible to drag specific file (e.g. from Windows Explorer window) and start writing configuration into device.
K2config will check if selected file is valid configuration file.

2.6.4 Configuration Editor

Window for configuration editing is divided into two parts – on the left part is located tree that is used for displaying all configuration items grouped into logical sections (e.g. devices, transfers, data points etc.). Right part of window displays parameters and their values according to selected item in the tree.
To add new item one can use function *Add*. In the opened window user can select desired item type that will be added to configuration. Function is located in the upper panel and also in the tree popup menu.
Other available functions:

- **Write to**. This function writes configuration to other active connected device. After clicking on this, the list of active connections is displayed. These connections can be used to write edited configuration to selected device. Selected connection used for reading the configuration has bold format.
  Button **Write** always writes configuration to device, the configuration was read from and does not matter where the configuration was written for the last time.
- **Template selection**. Switches templates. Detail description of operation with templates is in chapter 3.
- **Show tree**. Used to turn on/off tree in the left part of window.
- **Previous/next item**. Buttons are visible only when the tree is turned off. Buttons are used for movement between configuration items.
- **Append file**. Used to append another configuration file into opened configuration.
- **Clone**. Creates copy of the selected item.
- **Export**. Used to export whole configuration into XLS file (Microsoft Excel). For successful export user must have Microsoft Excel installed.
- **Delete**. Deletes selected items or whole section from configuration.
- **Validate**. Checks entire configuration and in case of error (invalid values, duplicate keys, values out of range etc) selects problem parameter and displays window with the error description.
  Read device. Sends command to read selected device to BaWiT.
  Activate transfer. Sends command to start selected transfer to BaWiT.
- **Edit transfers**. Opens window where user can quickly and simply assign data points to desired transfers.
- **Convert To**. Function can be used to change version of different configuration item types. It is useful when there is change in system configuration structure. In normal conditions it is not necessary to use it.

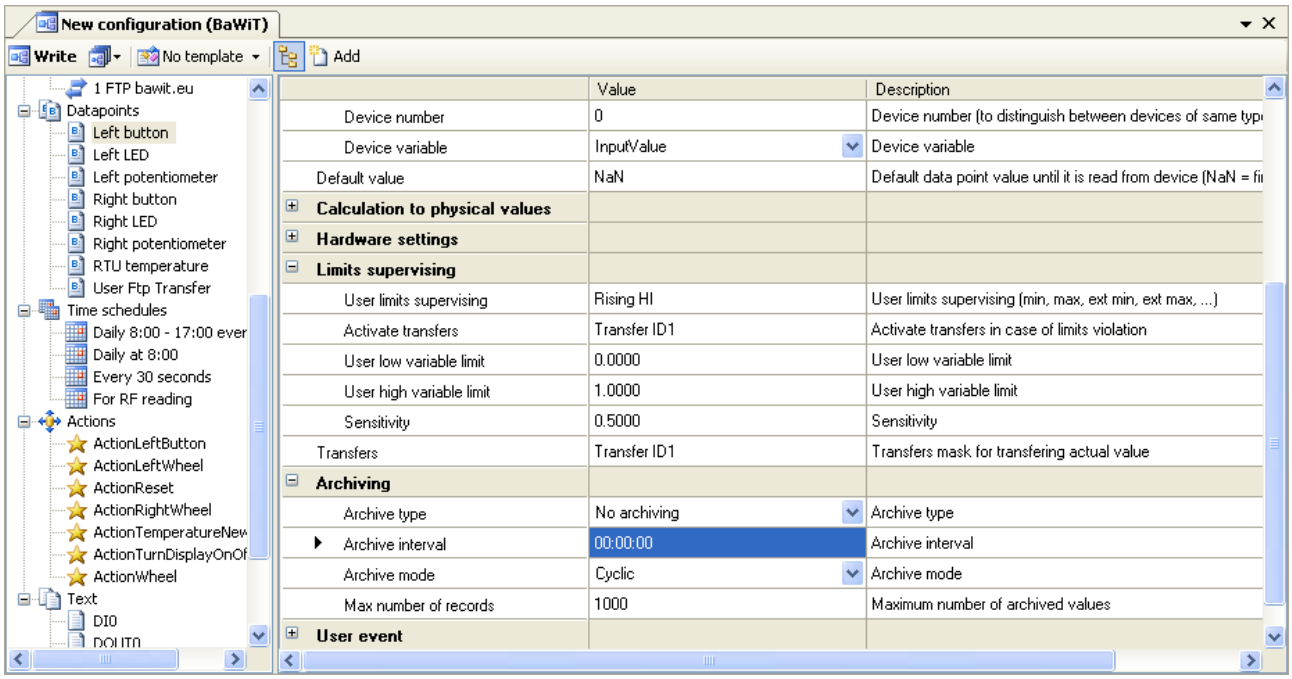

Figure 6 Sample of configuration editing

2.7 Monitoring

Monitoring window (see Figure 7) displays all actual values of measured data points. It is opened by double clicking on *Monitoring* node in *Active connections* window. It contains button for reading actual data and buttons used to turn on or off spontaneous mode. Further it contains function to change values of data points. It is possible to perform change only for one data point. At last the window contains function for switching between monitoring template and table presentation of values. This function is not available if the device does not have monitoring template or it is not located on computer's harddrive.
Device in spontaneous mode sends all changed data point values. Newly received values are highlighted for one second.

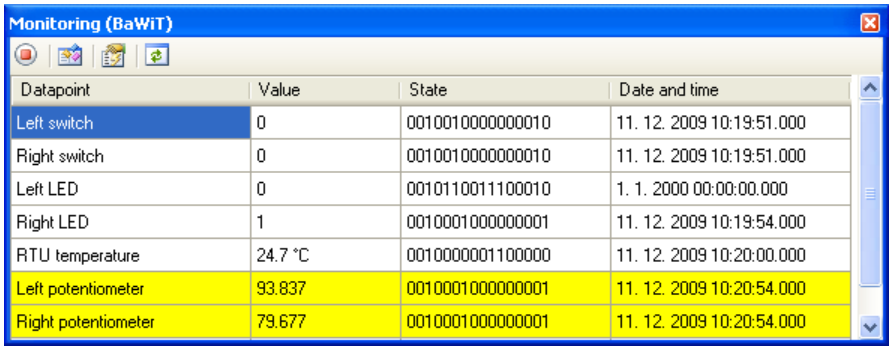

Figure 7 Monitoring window

By default there are only object and data point addresses displayed. After reading configuration from device, all addresses will be automatically translated to their respective textual representation.

2.8 Archives

Command is used for displaying archive values from one user specified data point. Window (see Figure 8) is opened by double clicking *Archives* node in *Active connections* window. For proper functioning it is required that device configuration is read (it does not matter if it is before or after window opening).
Upper part of window contains list of data objects and list of data points. Only data points that configured for archiving are displayed. Further there is panel for time period selection. Application supports these period types:

- **Date from** – time period can be set from specified date to specified date. Quick change of period is not possible.
- **Day** – period will be set to one day. Also start hour of day can be set.
- **Week from** – displayed period will be set to one week from specified date. User can also set start hour of day.
- **Month** – displayed period will be one month. User can set start hour of day.
- **Year** – displayed period will be one year. User can set start hour of day.
- **All** – All archive values will be displayed.

To quickly change displayed period one can use ◂ ▸ buttons.
By using button it is possible to turn on/off chart of archived values.

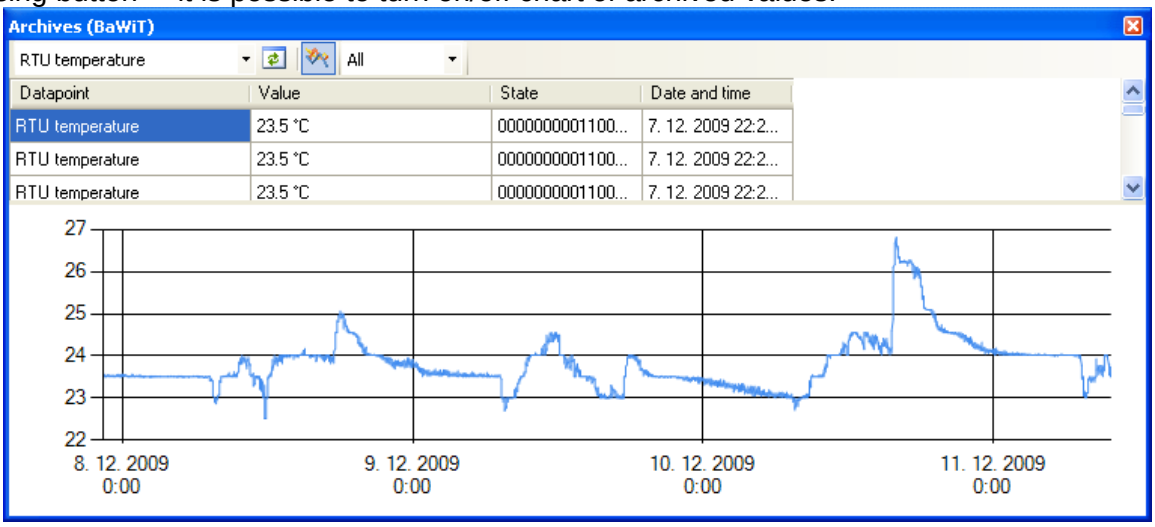

Figure 8 Archives with displayed chart

2.9 Data points

Node *Datapoints* in *Active connection* window presents configured device data points and further operation with them. It is necessary to read out the configuration from the device for correct operation.

2.9.1 Change of data points values

This function changes actual value of particular configured data point. To open data point value change window double click on requested data point, or this function is also available from context menu on data point. Opened window contains only one field, where an user can enter the requested value.

2.9.2 Deleting of data point archives

This function deletes archive content of particular data point. This function is available form context menu on data point (only in case, that given data point has configured archiving of values). Before the deleting of archives the user is asked to confirm this operation.

2.9.3 Deleting of all archives

This function deletes all archives of all configured data points. It is available from context menu of *Datapoints* node.

2.10 Power Sources

Command opens window which contains information about power sources connected to device. Window (see Figure 9) is opened by double clicking *Power source* node. For every battery are displayed information about SUID, battery change date, capacity, total connection time and total

consumed energy. Some of this information can be changed (useful when changing batteries) by clicking on battery label.

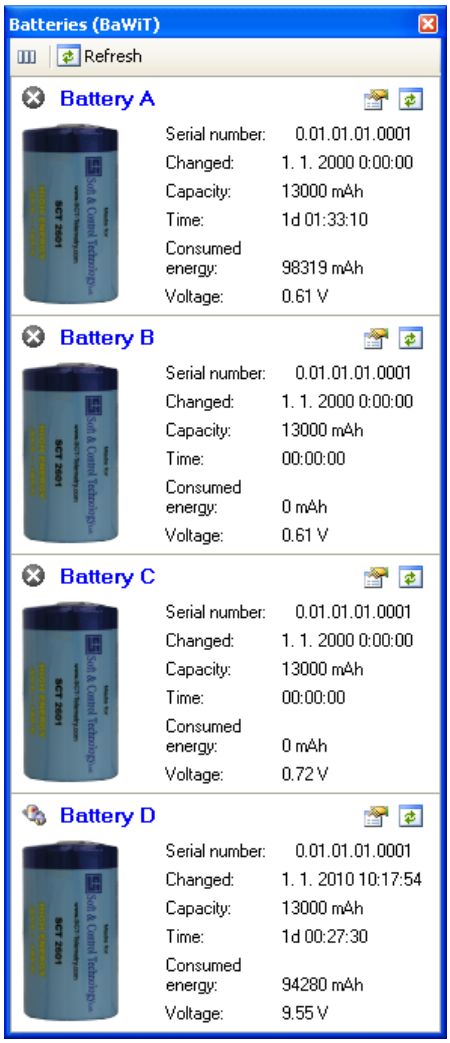

Figure 9 Information about connected power sources

2.11 Counters

This function presents information about counters in **BaWiT** device. To open the window, double click on *Counters* node. Functions for information update of all counters or particular counter and function to change settings of particular counter are available in this window.

3 Configuration and monitoring templates

Application **K2config** has option to define templates. These templates are used for designing of graphical presentation of device configuration and monitored data points of the device.
Currently three template types are supported:

- **Template for configuration structure** – These templates are defined above concrete configuration structures of various devices and are used to edit device configuration. It is possible to use these templates for multiple various configurations of devices, which have the same structure.
- **Template for device configuration** – These templates are defined above concrete device configurations and are used to edit device configuration. It is possible to use these templates only for that particular device configuration, for which they were created.
- **Monitoring template** – defined above concrete device configurations and they define method for presentation of configured data points in monitoring window. It is possible to use these templates only for that particular device configuration, for which they were created.

Files with templates are stored on computer hard drive in two directories. User templates are located in directory, where application stores all user settings (see chapter 1.3) in subdirectory **Templates\ConfigurationTemplates**. System templates are located in the same directory, where the application itself is located in the subdirectory **Templates\ConfigurationTemplates**.
Application searches for templates only in these two directories, the system template directory is being searched as first. All files with templates have extension *.templx*.

3.1 Templates for configuration structure

These templates are defined above concrete configuration structures of various devices and are used to edit device configuration. It is possible to use these templates for multiple various configurations of devices (they have the same configuration structure).

3.1.1 Editing the templates

To create new template of this type use menu **File** and command **New**. Select node *Configuration Templates* in the following opened window and double click on icon *Empty file* to create empty template. It is also possible to create template of this type trough template manager (see chapter 3.4). To edit existing template user has to use menu **File** and command **Open** or use the template manager (see chapter 3.4).
Edit window (see Figure 10) is divided into three parts. Left upper part contains tree with structure of all possible device configurations. Configurations are organized based on type (for example BaWiT Configuration). Each type is divided to areas (for example *RTU* or *Transfers*) and each area is organized to items (for example *RTU v0* or *RTU*). Template consist of several workspaces, while each workspace is bound to one configuration item (one item can have assigned only one workspace).
Right part contains actual workspace, where user designs the template view for particular configuration item. Left bottom part contains list of used control elements for currently presented workspace.

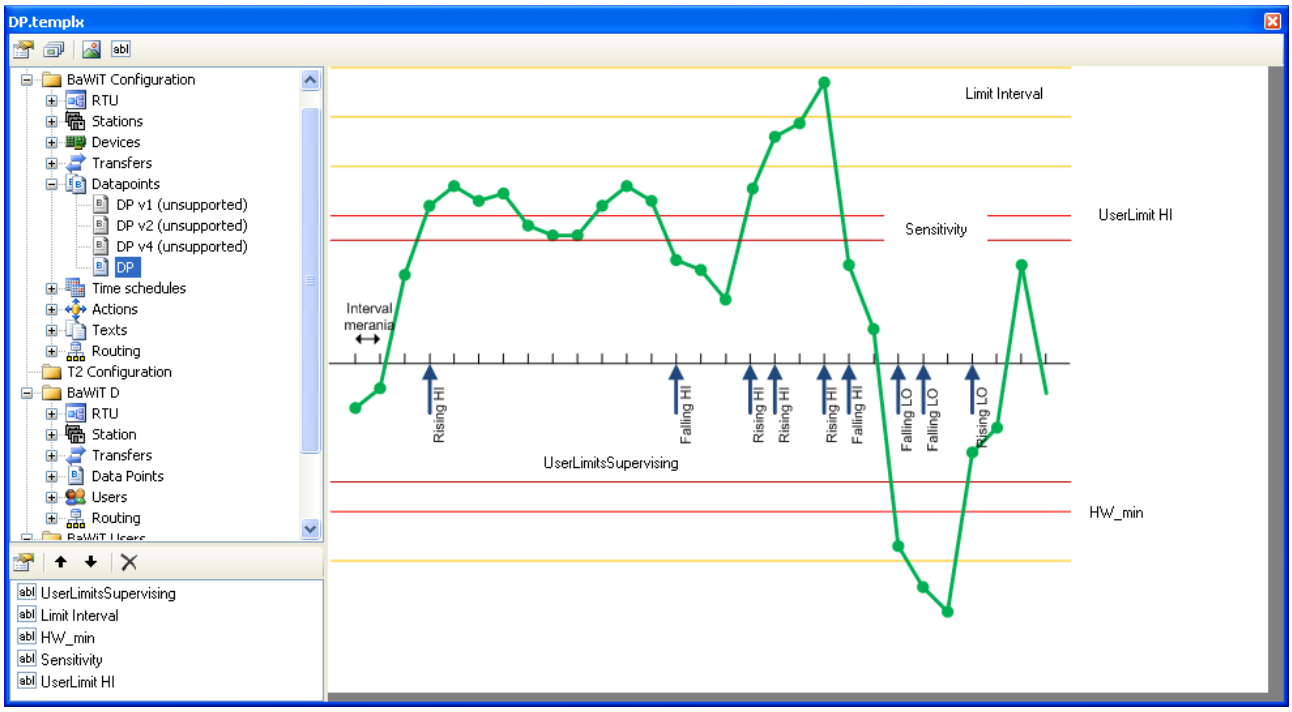

Figure 10 Edit of template for configuration structure

Upper control panel contains button **Template properties** to set basic properties of template (see Figure 11) – Caption, Name and Description.

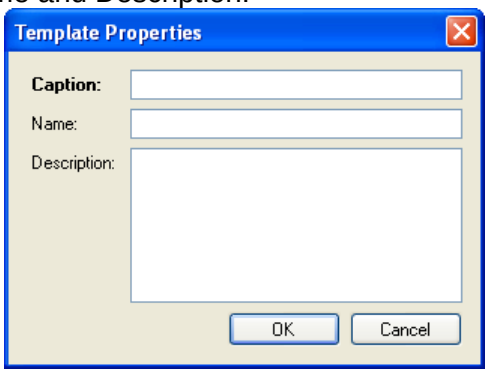

Figure 11 Window with basic template settings

Next button Template images is used to organizing the pictures available in the template. This function opens window (see Figure 12) to add new or remove unused pictures (by context menu above Template images). All pictures are part of file with template. The upper panel further contains control elements for using within template. To add new control element to template click on button of required control element (the button stays green) and click on required position of workspace, where the control element should be placed. It is possible to move or change size of elements on workspace.

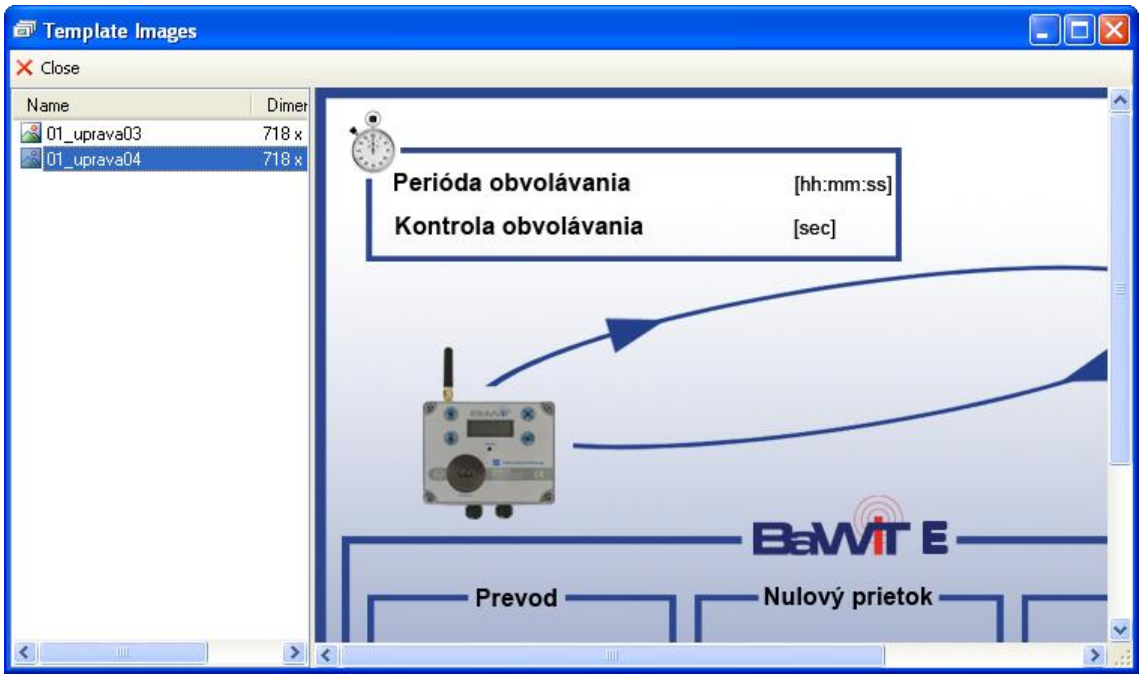

Figure 12 Window for image list management

In case that desktop for selected configuration item was not created, the right part of editing window contains message. Click on message text will open window (see Figure 13) where it is required to enter properties of workspace.

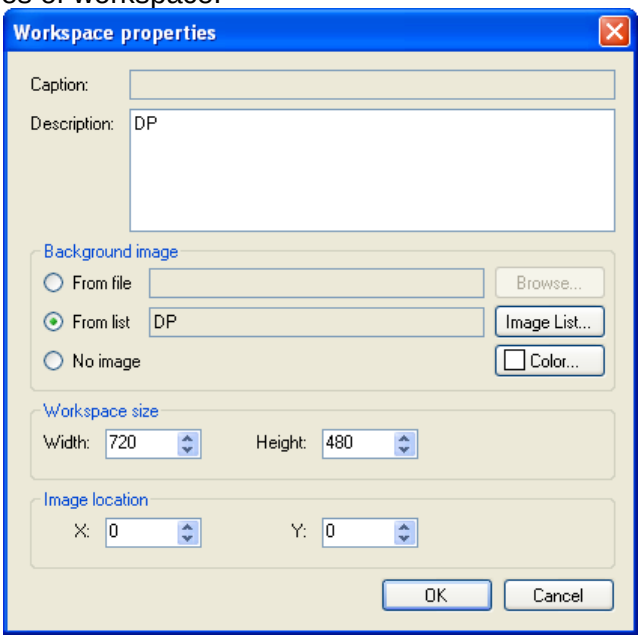

Figure 13 Workspace properties

It is possible to enter description of workspace, set the background color, background picture and picture position and dimensions of workspace. When selecting new background picture, this will automatically change workspace dimensions based on picture dimensions. In case that the picture is read from a file, it is automatically classified in list of template pictures. Function to open properties window of existing workspace is available by context menu of the workspace. This menu also contains function to remove workspace from template.
List of used control elements by template (left bottom part of editing window) contains its own control panel. In this panel are functions for presentation of properties for selected control element, moving the control elements (setting the order, which element is more to the top, etc.) and deleting of selected element.

3.1.2 Control elements

It is possible to use following control elements for configuration structure.

Static graphic element

The element presents static picture and text. This element in its final presentation does not affect editing in any way and is mainly used to present any additional information.

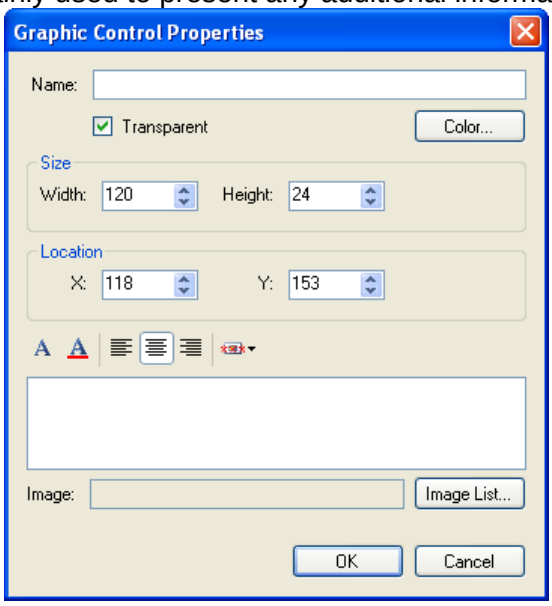

Figure 14 Static graphic control properties

Graphic Control Properties window (see Figure 14) offers setting dimensions and location of element on workspace (it is possible to this also by mouse in template editing window), presented text and settings of the text (font, color, alignment). It is also possible to select picture from picture list.

Parameter of configuration item

The element is used to make relation of template with particular parameter of configuration item. Main task of the element is to provide editor for selected parameter during configuration. This parameter is defined in parameter control properties window (see Figure 15). It is possible to select any parameter from actual selected configuration item the workspace is being created for. Button will lock final parameter editor and disable the changes for the user.

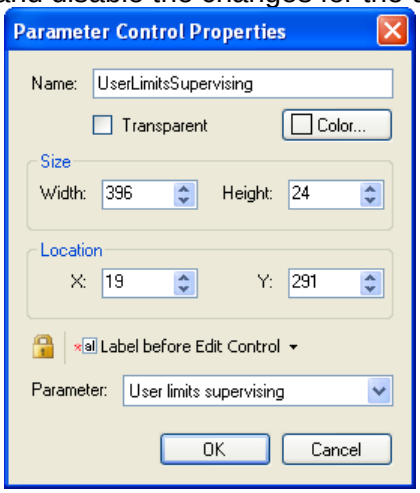

Figure 15 Configuration item parameter control properties

3.1.3 Using the templates

It is possible to use templates for configuration structure in window for editing the configuration (see Figure 6). This window contains button to change actual used template. Clicking on this button will present menu with list of available templates for configuration structure.

3.2 Templates for device configuration

These templates are defined above concrete device configurations and are used to edit device configuration. It is possible to use these templates only for that particular device configuration, for which they were created.

3.2.1 Editing the templates

Creating of new template is possible only during active connection when the user has to select command **Edit Template** by context menu of node *Settings* in *Active connections* window. If the configuration has not been read from device yet, it will be read automatically. Then the window will be presented (see Figure 16) to select which configuration types will be used for creating the template. If the template for this configuration existed already od the hard drive, required types will be checked in the window and it will be disabled for user to cancel their selection.

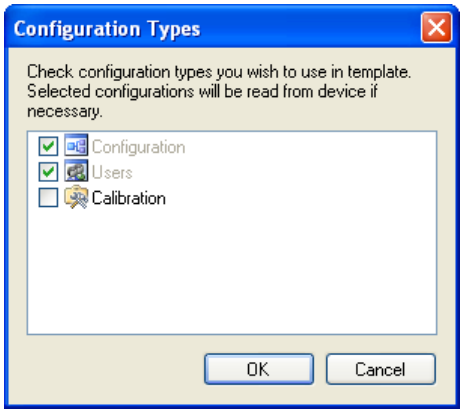

Figure 16 Window for selection of configuration types for template editing

After confirmation by button **OK** the readout of further required configurations (if not done yet) will be performed. Then the window for template editing (see Figure 17) will be presented.
Editing window is divided into three parts. Left upper part contains template workspace list. This template type can consist from multiple workspaces and it is possible to create references between them. One workspace can be signed as default, it means that during configuration editing by template, this workspace will be presented as first. This setting will be performed by appropriate command in context menu.
Right part presents actual workspace, where the user can design template appearance on selected workspace. Left bottom part contains list of used control elements on actual presented workspace.

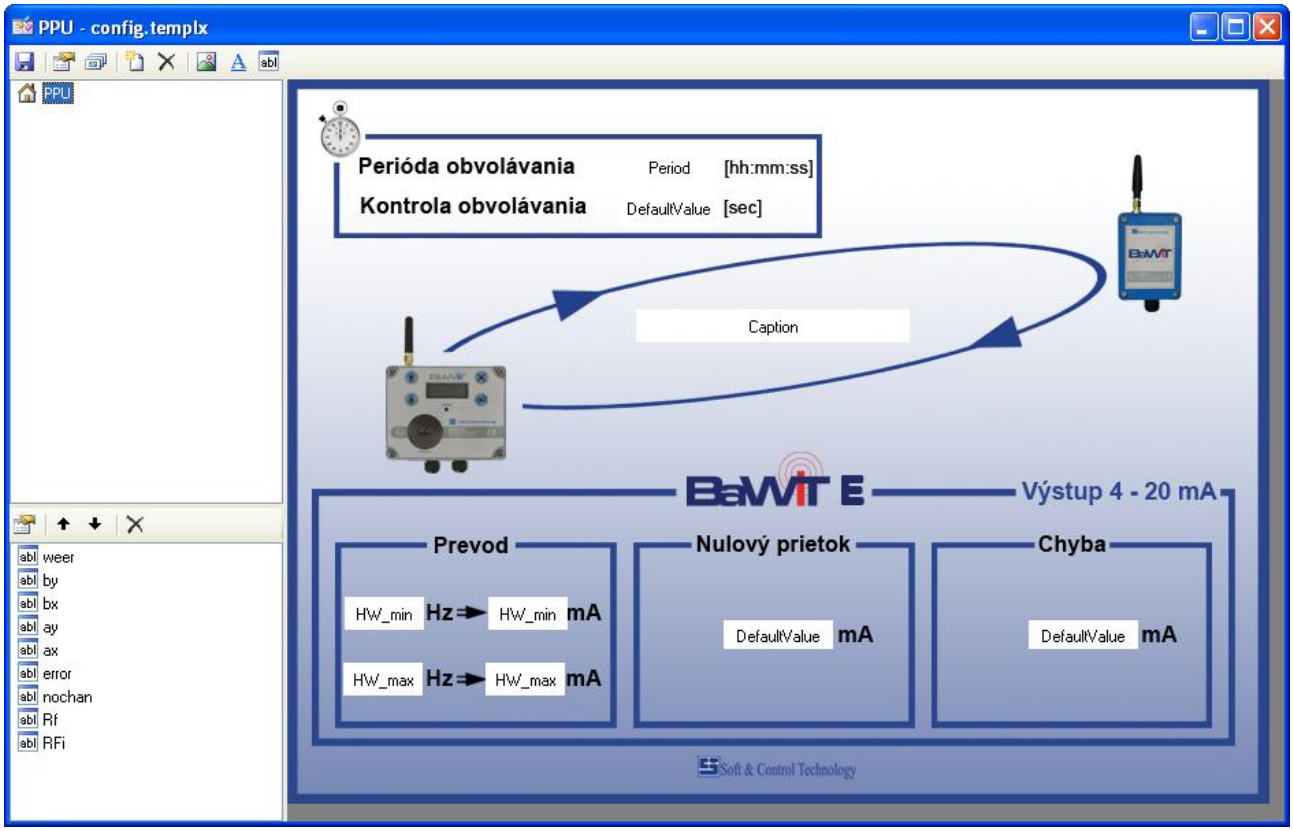

Figure 17 Window for configuration template editing

Comparing to window for configuration structure (chapter 3.1.1) this window upper control panel contains in addition function for creating new workspace and for deleting currently selected workspace upper control. There is also available function to save the template to disk. All template properties must be filled to successful saving to disk.
Closing window will cause automatic writing (if configuration didn't contain identification value about template) of required configurations (only those, to whom some control elements are related to) to the device. This is because there is also information as part of this configuration, which template on the disk has to be used for editing of given configuration.

3.2.2 Control elements

It is possible to use static graphical element (see chapter 3.1.2) and following control elements in templates for configurations:

Configuration parameter

The element is used for relating the template with particular parameter of concrete configuration item. Main task of the element is to provide particular editor for selected parameter during editing of configuration. This parameter is defined in parameter control properties window (see Figure 18). It is possible to select any parameter from configurations (see Figure 19), which user defined before starting of template creation.

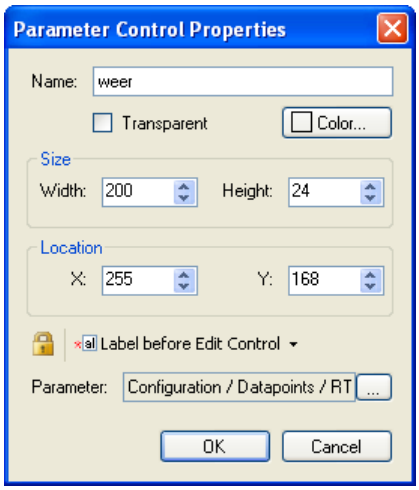

Figure 18 Configuration parameter control settings

. Button will lock final parameter editor and disable the changes for the user.

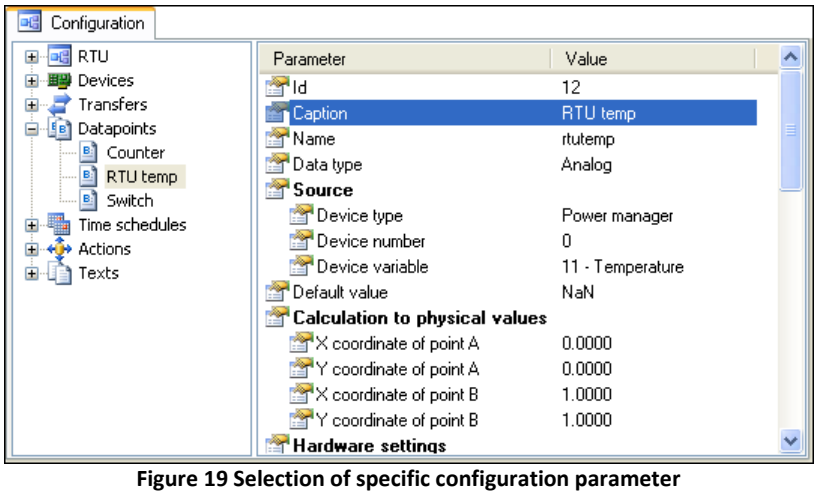

Figure 19 Selection of specific configuration parameter

Link to desktop

This element is used to creating active connection to other template workspace. Besides standard settings (dimensions and position, picture etc.) it is possible to define final appearance of connection to the workspace in this link control properties window (see Figure 20). Following options are available:

- **Link** – The link is presented as text clickable by user.
- **Button** – The link is presented as standard button of operation system.
- **Flat button** – The link is presented as button without visible edges and will be bring to forward when user will hover the mouse above.

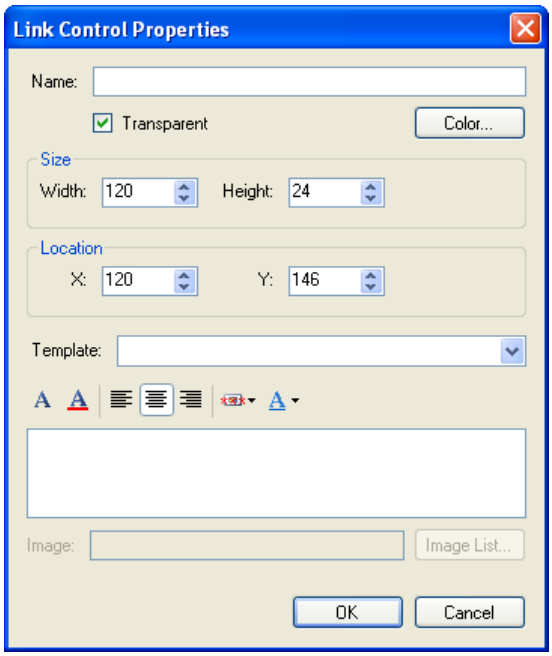

Figure 20 Link control properties window

3.2.3 Using the templates

Templates for configuration are used automatically (if they using was not globally disabled in application settings – see chapter 8) always at request for editing of device configuration. Editing of configuration is performed in window (see Figure 21) where only those parameters are available, for which control elements were defined.

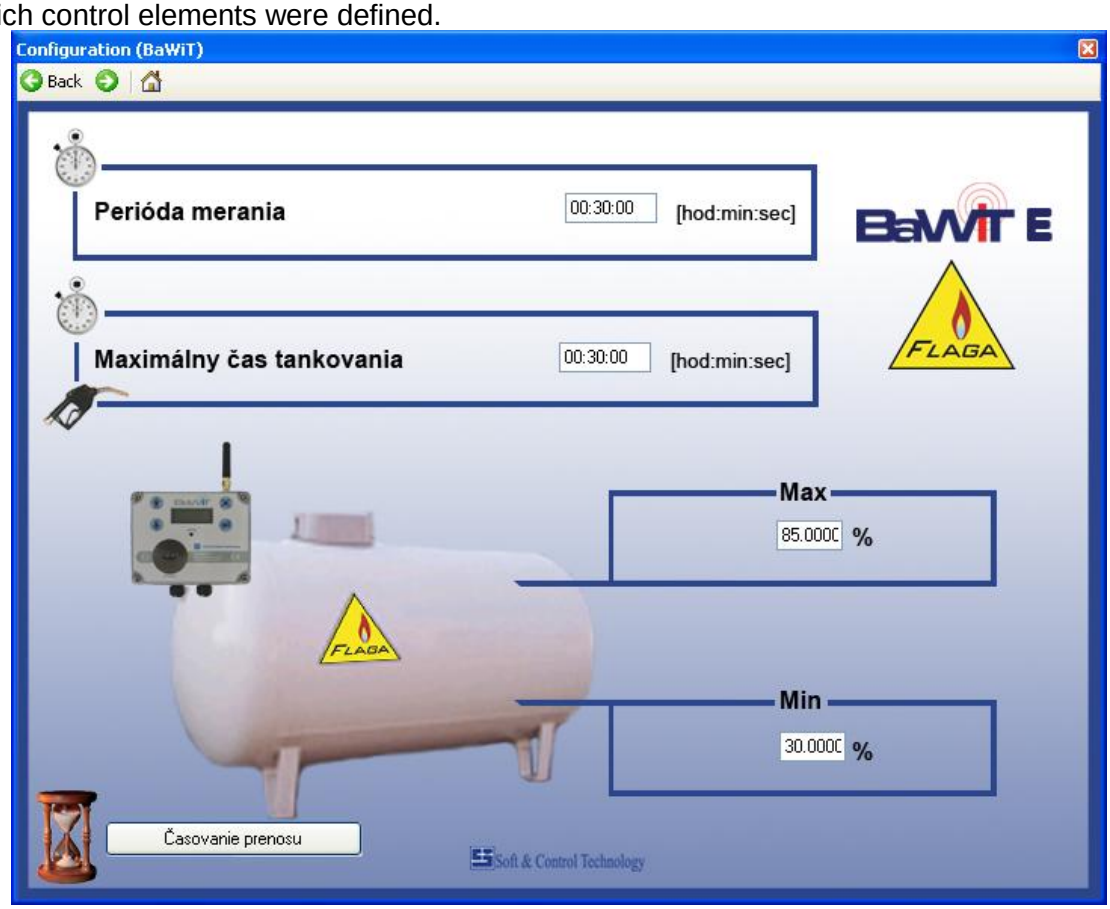

Figure 21 Configuration editing by using the template

Upper control panel contains buttons for switching visited workspaces (similar to internet browser).

3.3 Monitoring templates

These templates are defined above particular configurations of device and define presentations method of configured data points in monitoring window. This template type is available only for that device configuration, for which it was created.

3.3.1 Editing the templates

Creating of new template is possible only during active connection, when the user must select command **Edit Template** trough node context menu *Monitoring* in *Active connections* window. If the configuration has not been read from device yet, it will be read automatically. Operation in editing window is identical to operation with template for device configuration (see chapter 3.2.1).

3.3.2 Controlling elements

It is possible to is e static graphic element (see chapter 3.1.2) in monitoring templates and following control elements:

Element for value presentation

This element is used for numerical presentation of value of selected data point. List of available data points is acquired from configuration. Control element and data point are connected trough value of data point parameter *Id* in configuration, this means that after value change of this parameter the element will not present correct value.

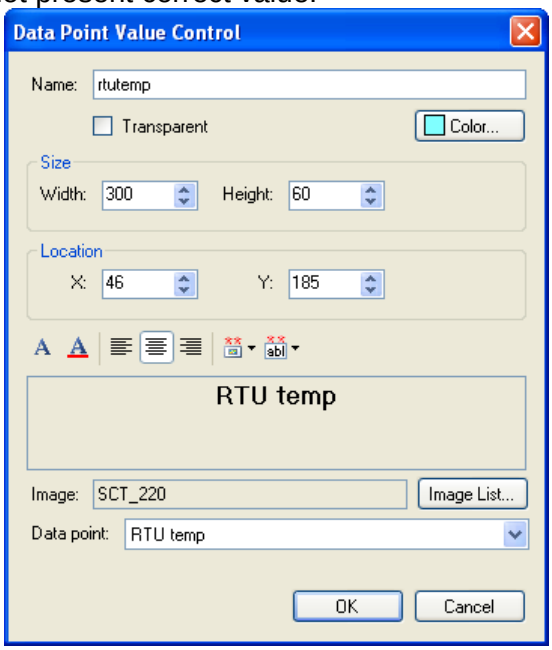

Figure 22 Data point value control properties window

Element for presentation of running

The element presents value of data point in running element. To correct value presentation it is required to define minimal and maximal presentable value.
It is possible to present the element horizontally and vertically.

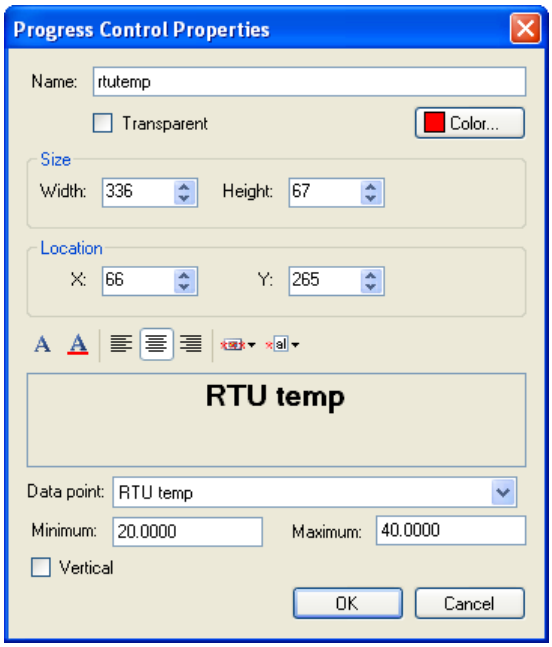

Figure 23 Progress control properties window

Graph

The element is used to present values of data points in graph form. It is possible to set maximal number of values to be presented.

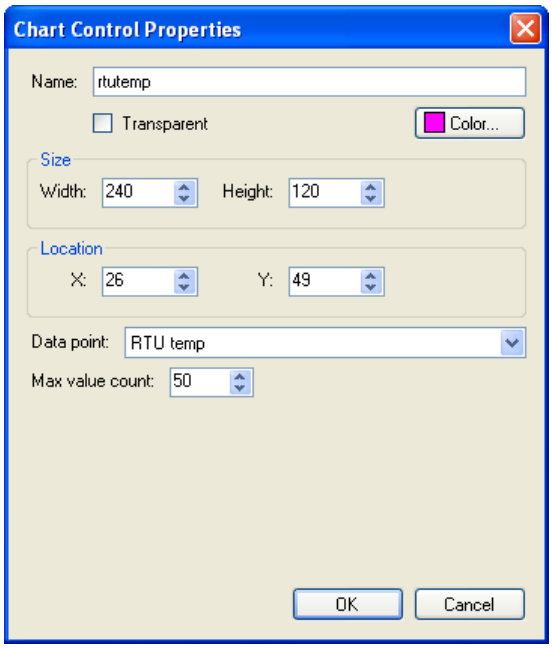

Figure 24 Chart control properties window

3.3.3 Using the templates

Monitoring templates are presented automatically (if they presentation was not globally disabled in application settings – see chapter 8) always at request for displaying the device monitoring window if in the moment of opening the window the device configuration was read (it is not read automatically).

3.4 Template management

Function **Configuration Templates Manager** is designed for management of available templates. This function is accessible in main menu **Settings**, opens window with presented user and system templates and contains following functions:

- **Install**. Installs templates not located in defined directories. This functions in its nature just copies selected files by user to user templates.
- **New template**. Creates new template for configuration structure.
- **Move**. Moves selected user templates to system templates and vice versa.
- **Delete**. Deletes selected templates from hard drive. Files are moved to recycle bin.
- **Refresh**. Reads again list of user and system templates. It is required to use it in case when new user or system template was added, and it was not created by one of these listed functions.

In addition, the context menu for configuration structure contains function **Edit** to open window for editing selected template.

4 Advanced Operations

4.1 Device File System

Some **BaWiT** telemetric systems have internal file system, which is used for storing system files, logs, archives etc. File browser is primary intended for service, but it is useful for retrieving logs from modem transfers or SPL debug logs.
All file browser functions are available through popup menu. Contents of this menu are displayed according to current context – device drive (0:, 1:), directory or file. There are standard functions as create directory, delete directory/file, get free space or internal file copy available. Any file can be downloaded to user's computer or copy files from user's computer to connected device. There is also drive format command. To transfer files from computer to device one can use **drag & drop** operations.

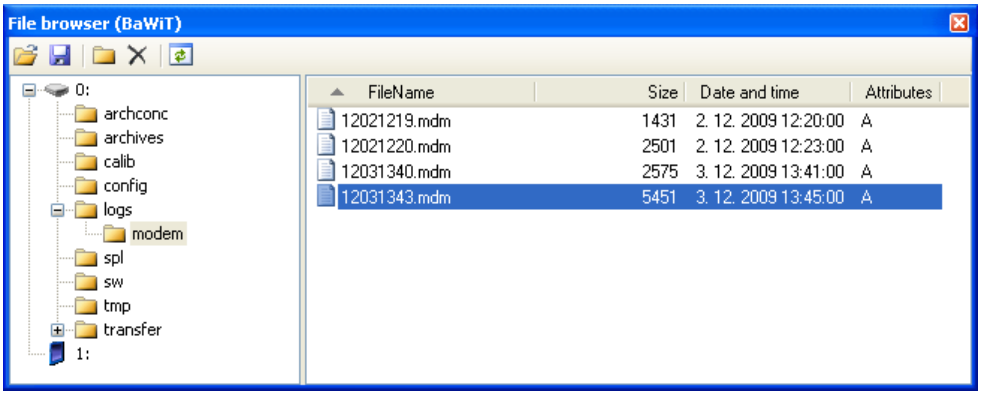

Figure 25 Device file system browser

4.2 Terminal

Terminal window (see Figure 26) is available on double click on *Terminal* node in *Active connections* window. Its main function is to display communication information and **BaWiT** debug messages.

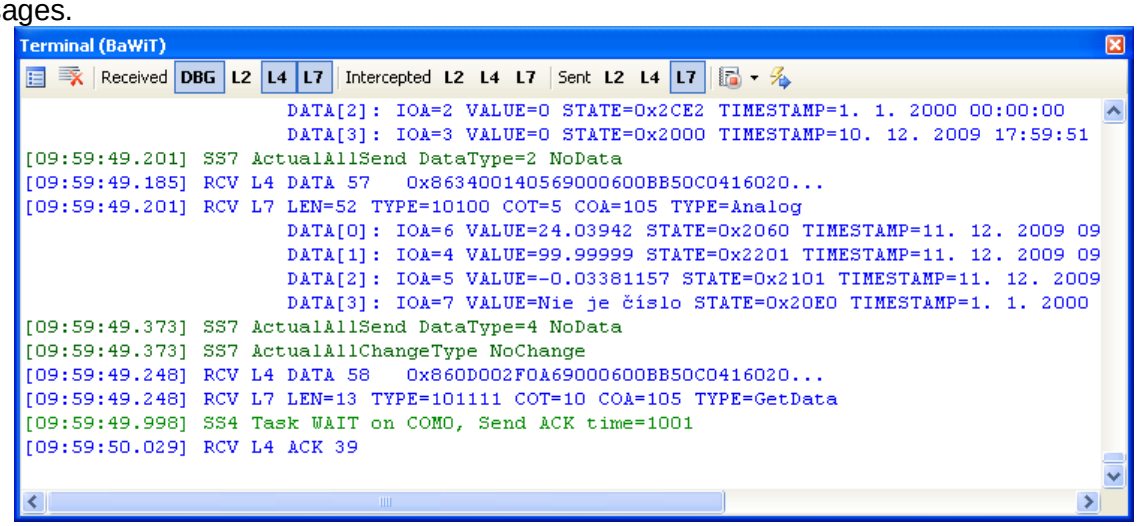

Figure 26 Terminal

Upper panel contains buttons for turning macro panel on/off, clearing terminal and button for writing all terminal messages into file. File log is located in the subdirectory **Logs** of application settings directory (see 1.3).
Buttons *L2*, *L4*, *L7* can be used to turn on displaying of sent/received communication packets between **BaWiT** and **K2config** application. In case of RFi connection, these buttons are also available for presentation of intercepted packets (though connected RFi interface must be

configured to retain all RF packets and the user must have sufficient license rights at **K2config** application level).

In case, that BaWiT is in loader mode, user can use function *Send File via XModem* to write firmware into device. Function opens window (Figure 27) where user should choose the firmware by pressing the button *Browse*. Thereafter user can choose communication protocol (by default XModem-CRC) and press the button *Send* to start firmware transfer.

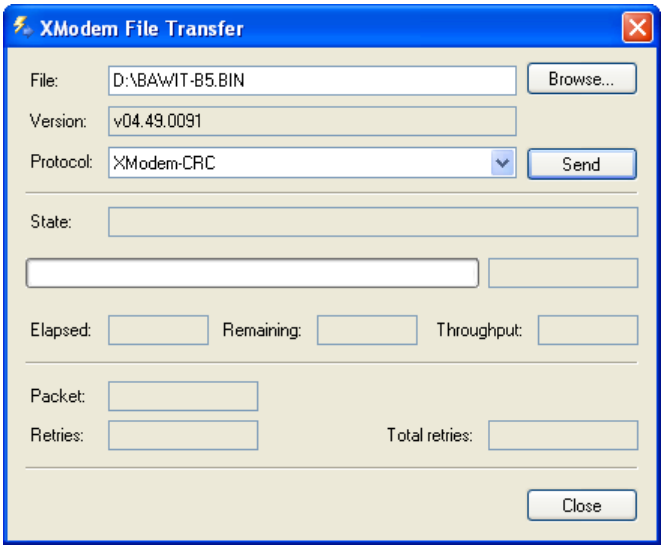

Figure 27 XModem window

4.3 System log

The *System log* window (see Figure 28) is available by double click on the *System log* node in the *Active connections* window. It presents system events logged by **BaWiT**. It is possible to presents these values only for particular time period or sort them arbitrary.

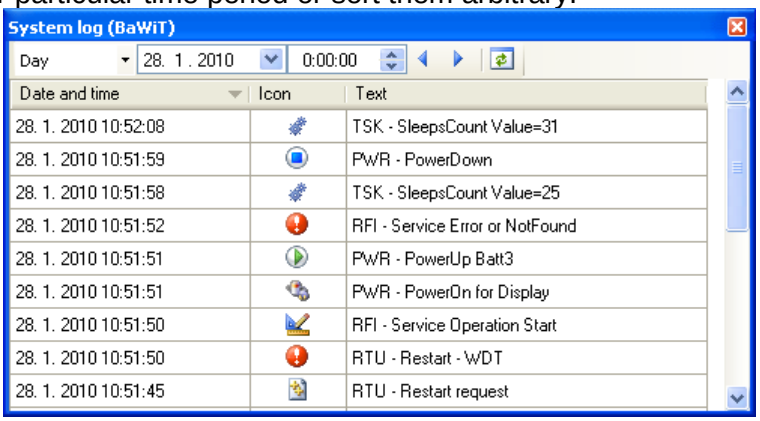

Figure 28 System log window

4.4 System Variables

System variables window (see Figure 29) is available by double click on the *System variables* node in the *Active connections* window. It presents various system information such as modem power supplying voltage, SIM card serial number etc.

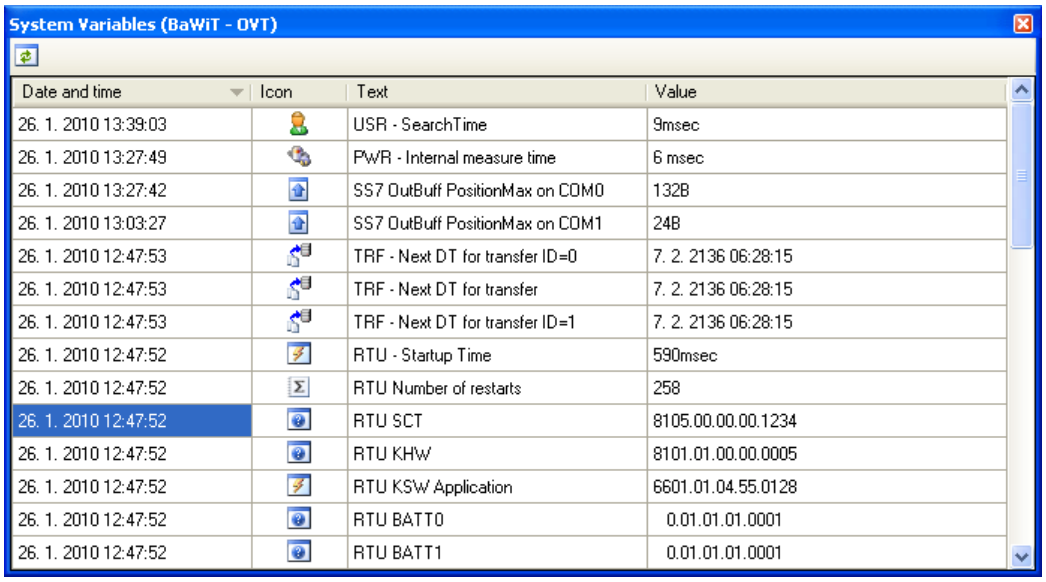

Figure 29 Service log window

4.5 Security log

Window Security log (see Figure 30) is available by double clicking the *Security log* node in *Active connections* window. It is used to display user activities in device.

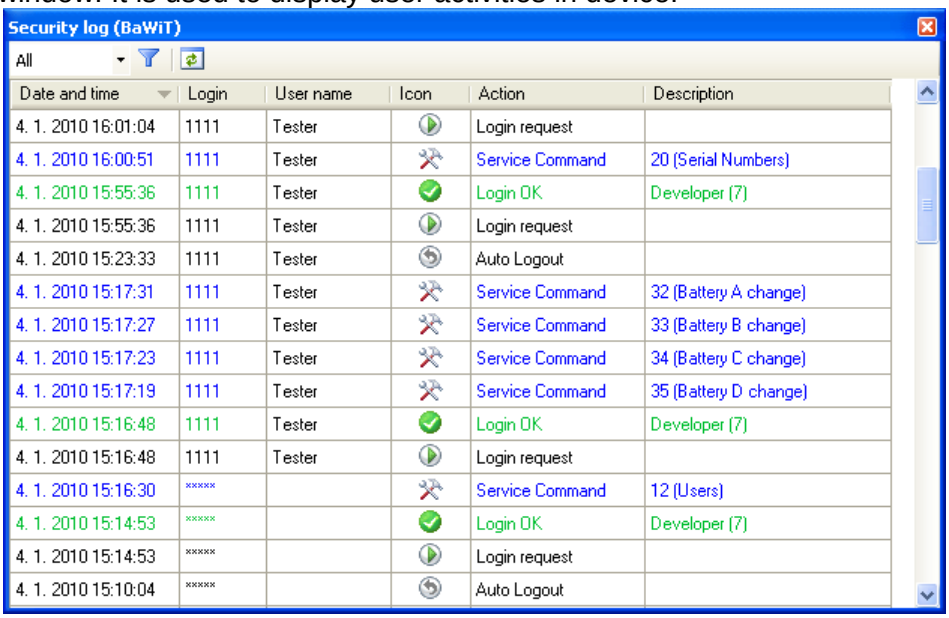

Figure 30 Security log window

Records in table can be filtered according to date and time. To apply filter settings one need to use button . By clicking this button no new security log records are read from device. To refresh the security log records one need to use button .

4.6 Calibration

This function is used to edit calibration settings of configured devices of **BaWiT** device.

4.6.1 Creating of new configuration file

New device calibration file is created by command *New* in menu *File*. It is necessary to select specific device type in opened window, for which the user wants to edit calibration.

4.6.2 Reading out of calibration file from the device

To read or open device calibration file for editing, the user should use the popup *Calibration* node menu in the *Active connections* window. This menu contains functions to readout and opens the calibration in window for editing, only for readout and writing any calibration file from file.

4.6.3 Writing calibration file into device

To write edited calibration into device click on button *Write*. This button is available only if the edited configuration was read from device (not from a file on hard drive).
Function to write any calibration file to device is available from the *Calibration* node popup menu in the *Active connections* menu.
Calibration node is also **drag & drop** node, so it is possible to drag specific file (for example from Windows Explorer window) and execute exchange of calibration file.
Before writing the selected file the **K2config** application checks if selected file has valid content for device calibration.

4.6.4 Calibration editor

Window for calibration editing is divided into two parts (see Figure 31) – on the left part is located tree that is used for displaying all calibration items. Right part of window displays parameters and their values according to selected item in the tree.
To add new item one can use function *Add*. In the opened window user can select desired item type that will be added to configuration. Function is located in the upper panel and also in the tree popup menu.

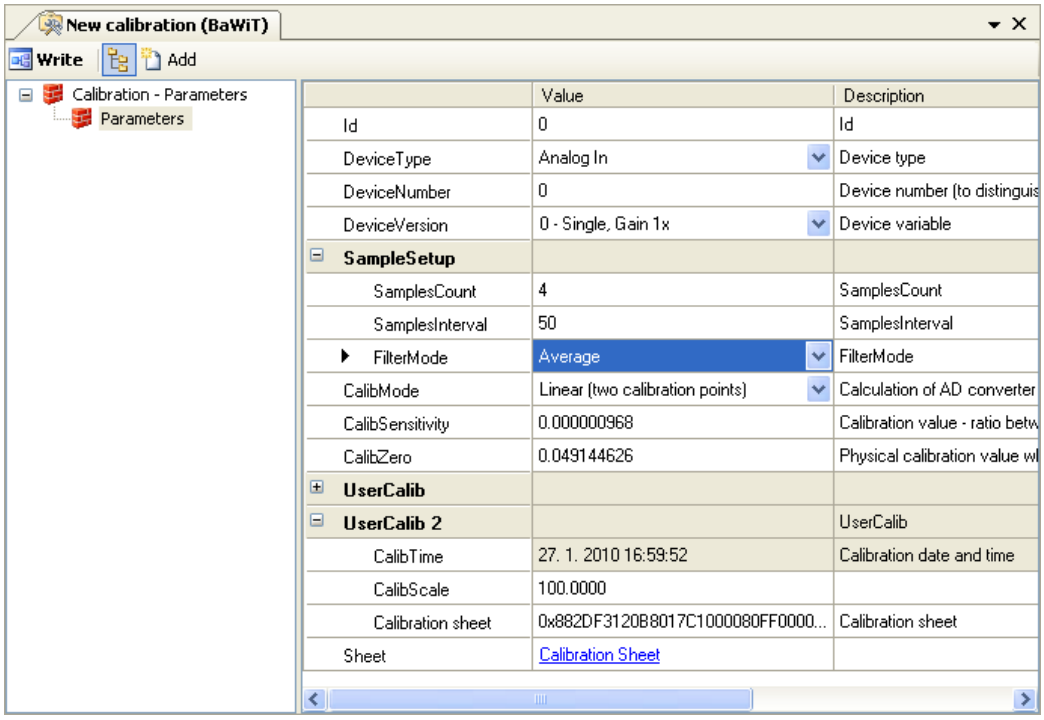

Figure 31 Calibration editor

Editing of calibration list values is performed by clicking on link with title *Calibration list*. This will open new window (see Figure 32) for editing individual values of calibration list. New calibration record is added by clicking on button . Pressing the will read out actual value of A/D converter from device. Then user have to fill value in column *Voltage* (the only column where values can be changed). If button is pressed during value reading from A/D converter the actual voltage value from connected Agilent device will be read out.

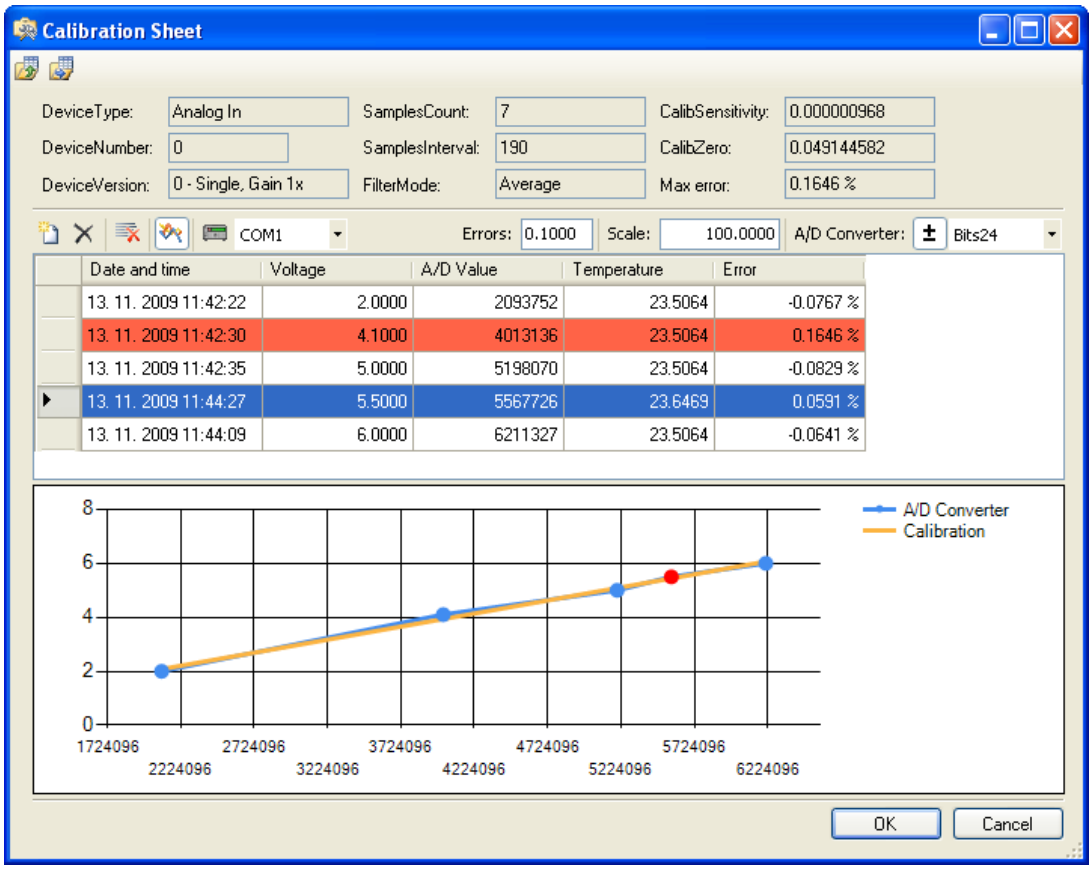

Figure 32 Calibration sheet editor

4.7 RFi Interface

4.7.1 Settings

This function is used for changing the parameters of internal RFi interface of connected **BaWiT** device. To execute the function double click on *RFi settings* node in the *Active connections* window. The function will read out actual device settings and then will present window (see Figure 13) to change these parameters. This window will be presented also in the case, if the function fails to read out the settings from connected device.

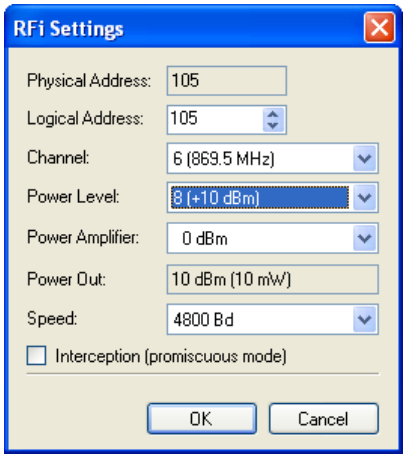

Figure 33 BaWiT's Internal RFi Interface settings

4.7.2 Firmware

Function is used to change firmware of device's internal RFi Interface. Function is similar to device firmware change described in chapter 2.5.

4.8 SPL settings

Settings of SPL language interpreter are available by child nodes of *SPL* node in the *Active connection* window.

4.8.1 Settings debugging

This function sets method of creating the SPL interpreter debug listings. The function will read out actual device settings and then will present window (see Figure 14) to change these parameters. This window will be presented also in the case, if the function fails to read out the settings from connected device.

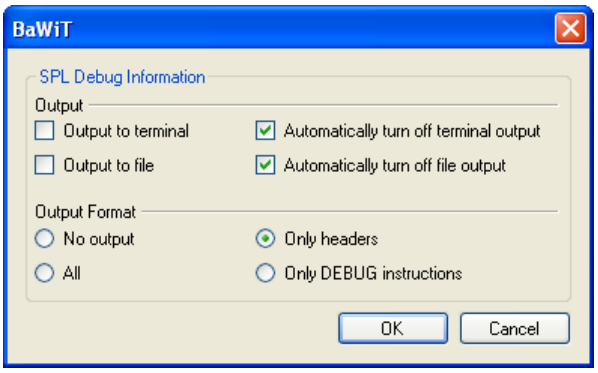

Figure 34 SPL interpreter debug settings

Meaning of individual settings of *Output*:

- **Output to terminal** – debug outputs are transferred to terminal
- **Output to file** – debug output are written to particular debug files
- **Automatically turn off terminal output** – turns off the output to terminal automatically after execution of first debug output to terminal
- **Automatically turn off file output** – turns off the output to file automatically after execution of first debug output to file

Meaning of individual settings of *Output Format*:

- **No output** – SPL interpreter does not generate any debug information
- **Only headers** – SPL interpreter will generate only basic debug information for actually running actions
- **Only DEBUG instructions** – SPL interpreter will generate debug information for all instructions with DEBUG bit set
- **All** – SPL interpreter will generate full debug information

4.8.2 SPL interpreter watch

Function is used for watching actual state of SPL interpreter. Watch window (see Figure 48) contains information about event count in event queue, information about actually running or last executed action and actual register values. For further description of debugging see chapter 7.3.

4.9 System users

In every BaWiT-B/C/D/E device can be configured two system users, who are not part of any standard user's configuration. One system user has access right *manufacturer* and the second user has access right *service user*. These users can be changed by using the appropriate functions in popup menu of *Users* node in *Active connections* window. The function will first read out the settings for required user and then will present window (see Figure 35) where the credentials can be changed.

Figure 35 Changing the service user login and password

4.10 Serial numbers

4.10.1 Read the serial numbers list

Serial numbers list can be read by using the function *Get Serial Numbers List* from *Device* node's popup menu in *Active connections* window. After successful read out the *Device* node can be expanded and it will contain all read serial numbers (see Figure 36).

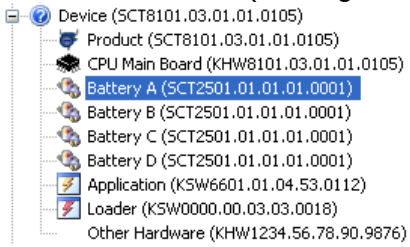

Figure 36 Serial numbers list

4.10.2 Editing the serial numbers list

The serial numbers list contains records that can be edited by the user with sufficient access rights. The function will open the window (see Figure 37) where user can set the type of serial number and serial number itself. To delete the record user should choose the *Deleted* type.

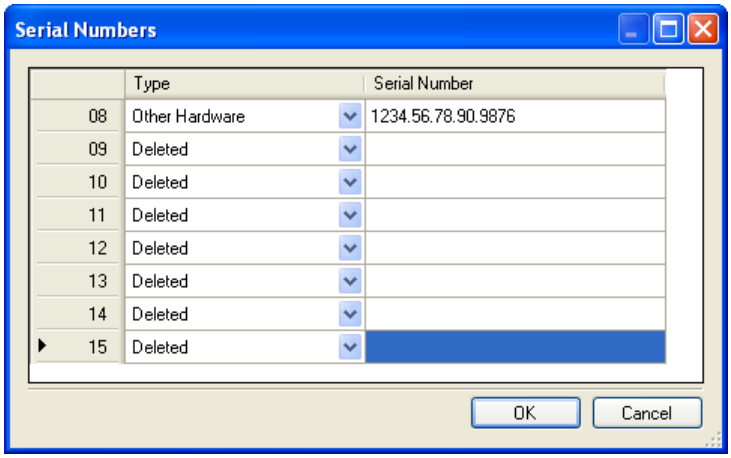

Figure 37 Serial numbers list edit window

4.11 Dataflash information

Function is used to display information about internal dataflash memory. To execute the function double click on *Dataflash* node in *Active Connections* window. The newly opened window (see Figure 38) will contain unique information from dataflash manufacturer, unique information from device manufacturer and list of most used dataflash pages. Every page contains number of write operations.

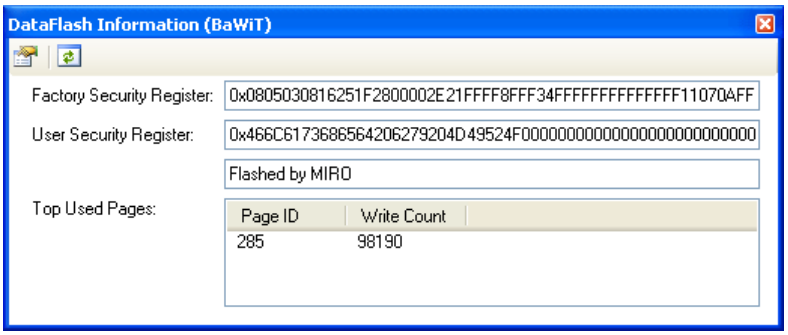

Figure 38 DataFlash information window

4.12 Searching for RFi interfaces

This function searches available RFi interfaces. User must have RFi Interface device connected to the computer and at least one template of RFi type connection to be able to use this function. The function is executed form context menu in window *Connection Templates* on node corresponding to RFi connection. Function opens window (see Figure 39) with automatic start of searching the available RFi interfaces.

This window is divided into two parts. Left part contains information about found RFi interfaces. Right part contains a circle graph presenting approximate distance (stated as arithmetic average of *RSSI r* and *RSS I* signal power) of individual RFi interfaces from user RFi Interface. Graph by no means represents direction of positioned devices.

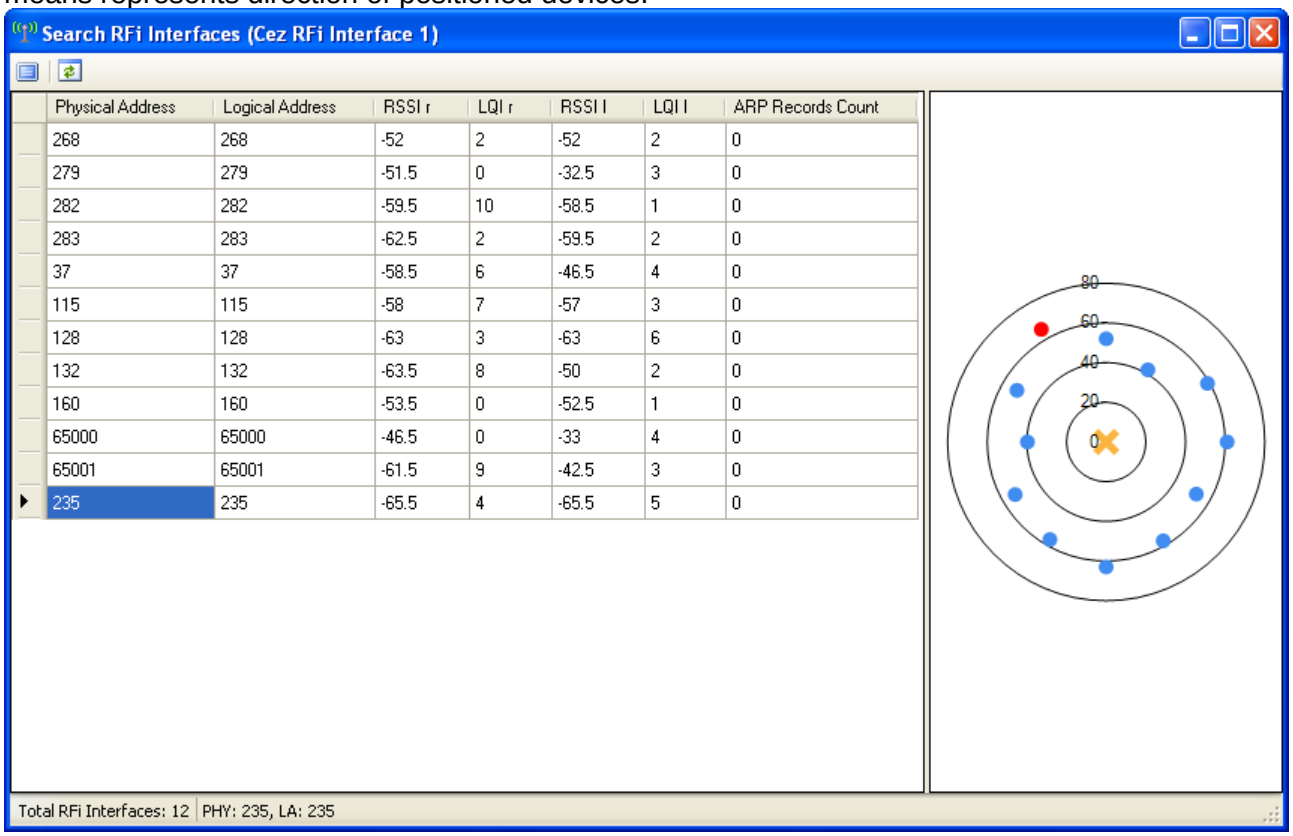

Figure 39 Window for searching for the RFi interfaces

Upper control panel contains function to open terminal window (see chapter 4.2) where it is possible to view various listings. The function will start searching again.

Table and graph provide context menu to the user. This menu contains functions for connection to selected RFi interface and function for acquiring further properties (name, serial number and software version number).

4.13 Testing of RFi interface availability

Function tests availability of single particular RFi interface (ping). User must have RFi Interface device connected to the computer and at least one template of RFi type connection to be able to use this function. The function is executed form context menu in window *Connection Templates* on node corresponding to RFi connection.
Function opens window (see Figure 40) divided into two parts. Upper part contains information about sent and received ping packets. Bottom part contains graph presenting line of *RSSI r* and *RSSI r* signal power.

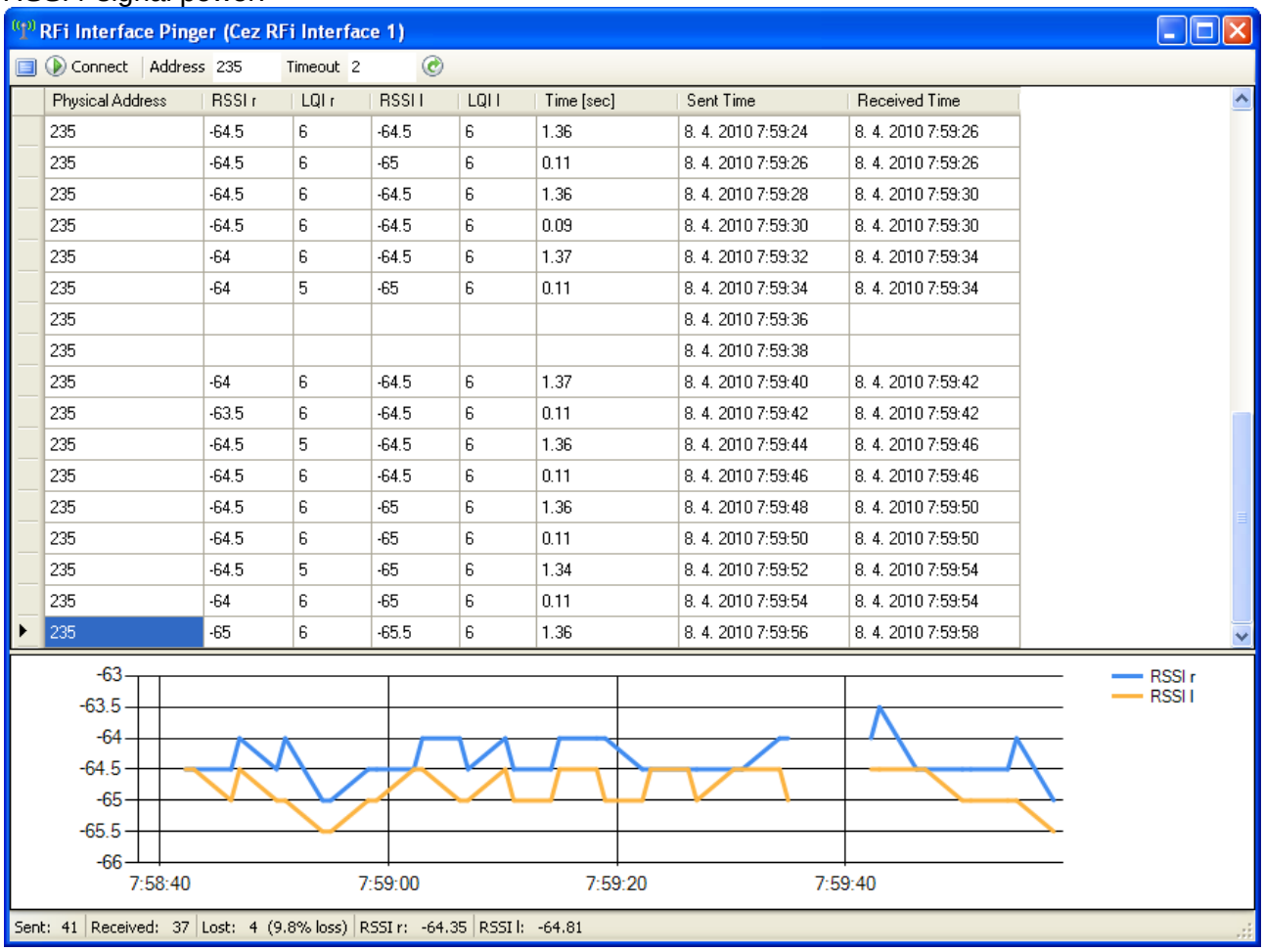

Figure 40 RFi Interface ping window

Upper control panel contains function to open terminal window (see chapter 4.2) where it is possible to view various listings. Next function connects to RFi interface being tested. This panel further contains two fields, where it is required to enter physical address of RFi interface being tested and time limit in seconds. This time limit defines interval when a single ping packet is sent to RFi interface and also defines time to receive answer from RFi interface being tested.

5 Updates

5.1 Application Updates

This function is intended for updating **K2config** application files. For successful update process, it is required that logged operating system user has write right for directory where **K2config** application files are located.
Function is available through menu *Help* / *Check for updates*. Automatically after opening the window (see Figure 41) will start updates check. Current update progress is displayed in the window. In case of new updates, they will be automatically downloaded and after user confirmation will be installed.

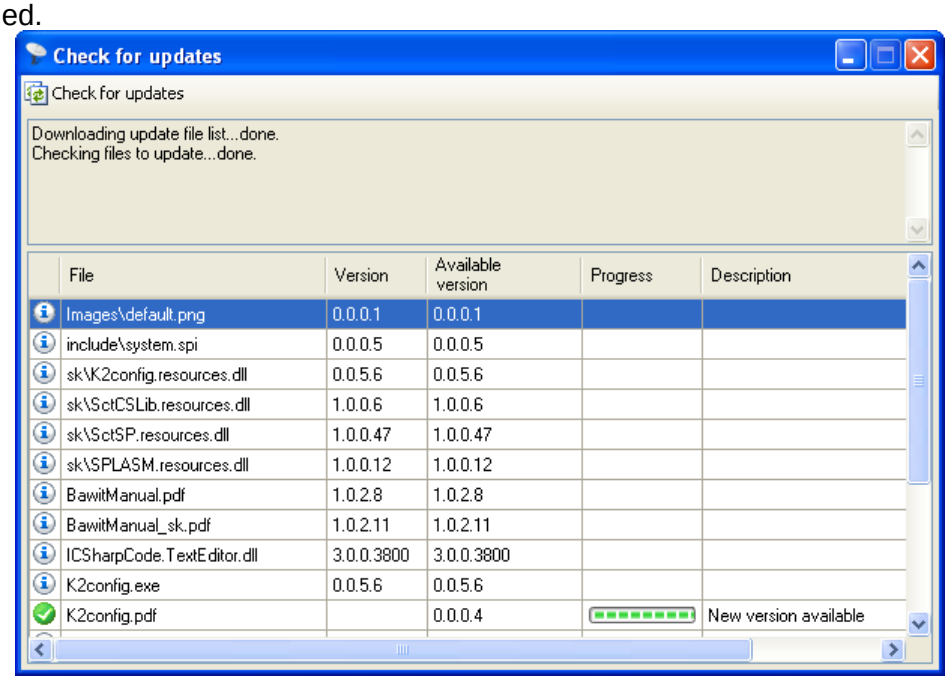

Figure 41 Checking and downloading new application updates

Application files are updated through internet from configured URL. This address can be changed in application settings (*Tools* menu, command *Options*). There is also option to turn on automatic checking for updates. When automatic checking is turned on, application will notify (see Figure 42) the user that there are new updates available. This notification is clickable and opens window for updates download.

Figure 42 New updates notification

6 Licenses

Application **K2config** uses to limit accessibility of some of its functionalities licensing system. This means that only user with valid license can perform particular operations.

6.1 Information about license

Information about actual license is available in main menu **Help** under command **License**. Function opens window (Figure 43) with information about actual license.

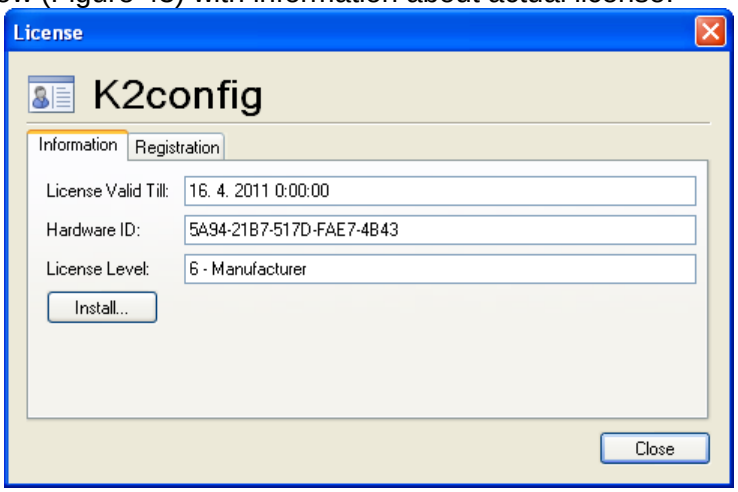

Figure 43 License information

In case that license is not available or is not valid, the following window will be displayed (Figure 44) for the user.

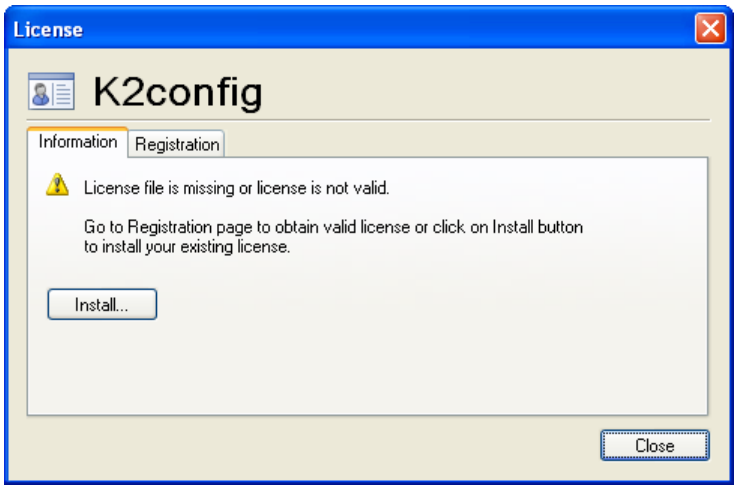

Figure 44 Information when license is not valid

In both cases the window contain button **Install** to copy license file from computer hard drive to system directory of the application. After successful copy of the file the program will notify the user to restart the application.

6.2 Acquiring the license

User can anytime send a request for valid license and tab *Registration* (Figure 45) in license window is used for this.

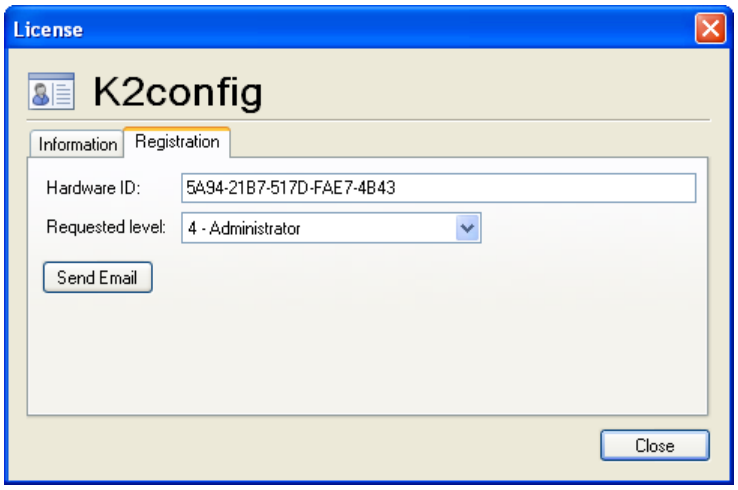

Figure 45 Acquiring new license

User must define in this tab what license level he requires and then click on the button **Send Email**. Application will open window in system of default email client and fill up some information. User must enter his full name, company where he works, telephone contact and the reason why he requests for assigning required license level.

6.3 License levels

The application recognizes 7 license levels. Listed levels are ordered from the lowest level to the highest together with functions available for particular level. It is a rule, that higher level automatically makes available all functions from lower levels.

.

- **Guest**
  - Monitoring
  - Archives viewer
  - Configuration editing
  - Users editing
  - Calibration editing
  - Terminal
  - Power sources
  - System log
  - System variables
  - Counters
- **User**
  - Spontaneous mode in monitoring
  - Filesystem browser
  - Read directory or file contents
  - Dataflash
- **Power User**
  - Change data point value
  - Write configuration into the device
  - Change work mode
  - Change date and time
  - Change SPL debug settings
  - SPL watch
  - SPL source files editing
- **Administrator**
  - Delete data point archive
  - Delete all archives
  - Write users into the device
  - Write calibration into the device

  - Flash firmware
  - File system format
  - Create new file or directory
  - Delete file or directory
  - Rename file or directory
  - Display intercepted packets
  - Change battery information
  - Security log
  - Change counters
  - Change RFi settings
- **Service man**
  - Set service man user
  - Change serial numbers list
- **Manufacturer**
  - Set manufacturer user
  - Change COA
  - Change product serial number
  - Set user dataflash registry
- **Developer**
  - Read data using different modes
  - Read/write processor EEPROM
  - Configuration structures editing
  - Enums editing
  - Memory structures editing
  - Icon editing
  - Localization editing
  - Structures publishing
  - Save password in connection templates
  - MD5 tool

7 SPL ASM Language

K2config environment contains internal compiler for the SPL language (SCT Programming Language). This language enables user to extend device functions which cannot be achieved by device configuration. Language source files are edited directly in the application. For this purpose, the application contains its own editor with syntax highlighting.

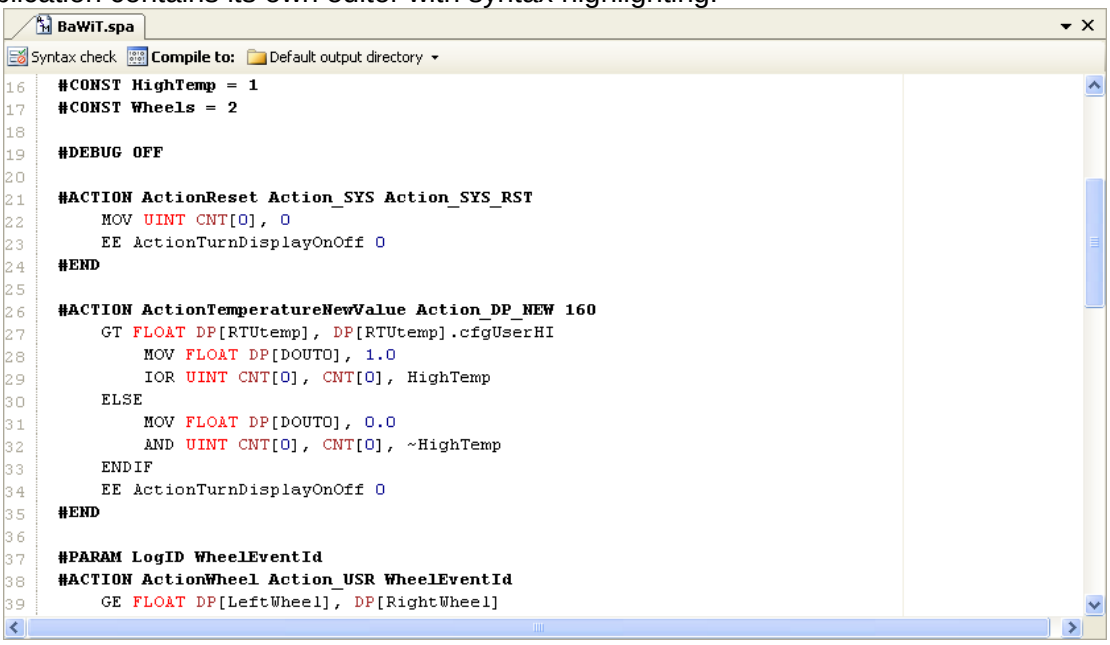

Figure 46 SPL ASM Editor

New SPL language file is created through the function *New* in the *File* menu. In the opened window user should choose file type SPL ASM.

7.1 Compilation

Window for editing of the source language files contains upper panel, which contains following functions:

Syntax Check — Command provides syntax checking of the opened source file; it does not generate compiled code.

Compile To: — Button starts the compilation of the source file. Compilation destination can be set by menu that is located right from that button.
In menu, there are following items:
Default output directory – the directory can be set in compiler settings (see 7.2).
Source file directory – resulting code is saved into the directory that contains the source file.
The menu also contains the list of all open configurations. When the code is saved into the open configuration, all actions and texts in the configuration are deleted first.

All outputs of the compiler are shown in the *Error list* window that can be opened using the command *Error list* in *View* menu.

7.2 Compiler Settings

Compiler settings are located in the window *Options*. It can be opened via function *Options* in the menu *Tools*. Left panel contains tree node with the name Compiler. Settings of the compiler are accessible through its child nodes.
Page *Options*:

- **DEBUG ON** – determines default state of the preprocessor command `#DEBUG` at the compilation start. Checked box is equivalent to the command `#DEBUG ON`.
- **Big Endian** – generates byte code by using Big Endian mode.
- **Default result type** – determines default result type of the specific instructions. It is equal to preprocessor command `#RESULT`.
- **Parameters** – equals to preprocessor command `#PARAM`.
  - **LogDestination** – determines the destination, where the log (file system, serial line, etc.) will be generated. BaWiT devices support only file system at the moment.
  - **LogNumber** – Log identification. BaWiT will use this number to generate the file that contains debug information about action execution.
  - **LogSize** – sets maximum size of the debug log generated by the device.
- **String type** – determines string type, that will be used for all text defined by command `#STRING`.
  - **Short string** – string type of short string (max. 8 characters).
  - **Long string** – string type of long string (max. 64 characters).
  - **Automatically** – string type is determined automatically according to the defined text length.
- **Generate explicit END instruction** – when checked, instruction END (unconditional action termination) is automatically added to the end of every compiled language action.
- **Generate separated files** – when checked, and if text identifiers of the data points are used in source file, two files are generated during compilation with output saved to the disk (not to the device configuration). The actions file will have the file extension *.spx*; text file will have the file extension *.spr*.
- **Output directory** – used to set the default output compilation directory.

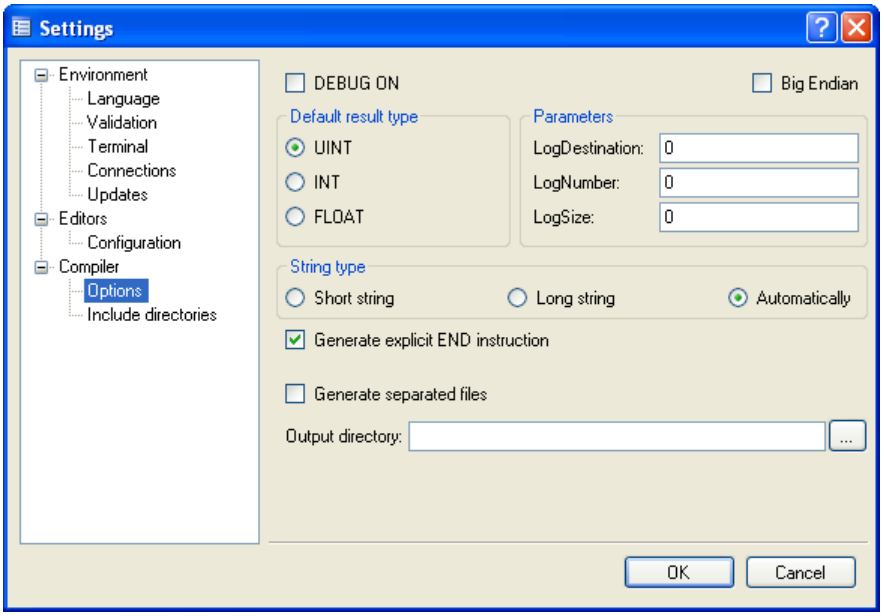

Figure 47 SPL ASM Compiler settings

Page *Include directories* determines directories, that are sequentially searched by compiler when using preprocessor command `#INCLUDE`. In directory name you can use variable `$(SPLAC)`, which represents directory where is SPL ASM compiler located (eg. `$(SPLAC)\include`).
For detailed description of compiler settings see SPL ASM programmer's manual.

7.3 Debugging of SPL applications

Function **SPL Watch** is intended for debugging SPL application and monitoring the state of SPL interpreter (performed action, registers and etc.). This function is available via node *Watch* in *Active Connections* window. Function opens window (se Figure 48) splint into two parts.

Left window part contains information about actual state of SPL interpreter. There are located information about number of events waiting in queue for processing, actually executed (or last executed) action, duration time (tick) of action in milliseconds, ordinal number (ON) of SPL instruction being executed and internal state conditional register T. Further, there is table presenting actual state of SPL registers W, CNT, CTC and ALR.
Immediate refresh of information in left part is possible to perform by clicking on button or turn on automatic refresh by button. Detail description of automatic refresh is listed below.
Right window part contains main source SPL file and is intended to setting interrupt points (breakpoints) and presentation of actually executed instruction.

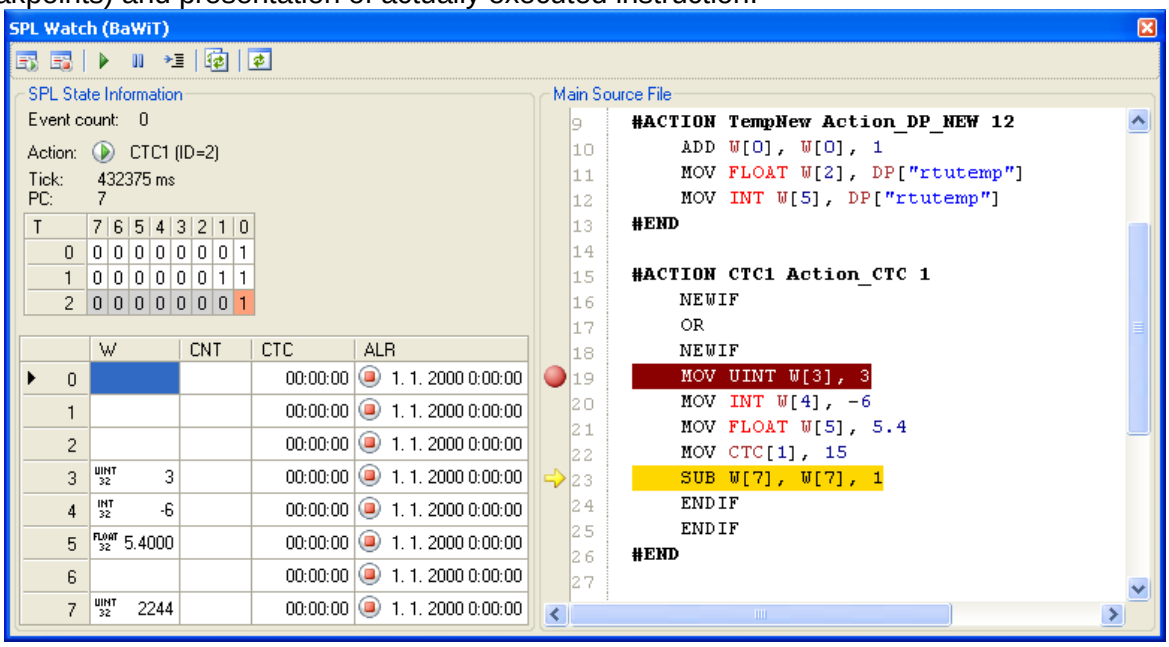

Figure 48 Window for debugging SPL applications

Window for debugging SPL applications contains following functions:

Turn breakpoints on. Turns on global sensitivity of SPL interpreter to interrupt points that means SPL interpreter will stop at each instruction, which have set point for interrupt.

Notification: This setting will be forgotten also after restart and can cause unwanted stopping at instructions, which have set BREAK bit at compilation by SPL directive `#DEBUG BREAK`. Interrupt points set by user while debugging will not be forgotten after restart.

Turn all breakpoints off. Will turn off global sensitivity of SPL interpreter to interrupt points that means SPL interpreter will not stop at any instruction, which has set interrupt point.

Continue. Switches SPL interpreter into state, in which all following instructions are executed until it will not get instruction with set interrupt point (and the global sensitivity is turned on). Function turns on automatic refresh in the same time.

Pause. Stops processing of SPL actions at „the closest“ instruction. Stopping need not be immediate if the queue of waiting instructions for processing is empty. This functions also turns on automatic refresh.

Step. Executes actual instruction and stops execution of the next instructions. This function refreshes all state information of SPL interpreter.

Refresh automatically. Turns on/off automatic data refresh. When the refresh is turned on, the data are automatically refreshed by actually processed/being processed SPL action. In case that some SPL action is currently in state *being processed*, the automatic refresh is turned off and state of registers W, CNT, CTC and ALR is read.

Refresh. Refreshes state information of SPL interpreter (executed action, conditional register T and registers W, CNT, CTC and ALR).

It is possible to set/cancel the interrupt point on any instruction by clicking mouse in right window part (containing source SPL file) left from presented row number. Interrupt point is signed by red color on given row and by the icon  . Actual instruction to be executed is marked with yellow color on particular row and by icon  .
Setting the interrupt points and presentation of actual instruction is supported only in case, that user has read configuration and main SPL source file from device and these are identical.

8 Application Settings

K2config application settings are accessible through function *Options* in the menu *Tools*.
The page *Environment* / *Language* allows user to set the language version of the environment. At the moment, Slovak language and English language are available. It is necessary to restart the application when the language is set.
The *Environment* / *Validation* allows user to set different automatic check ups. User can choose from:

- **Check for device date/time difference greater than X minutes** – Anytime the date and time are read from device, application compares it with the current date and time of local PC and warns user if the difference is greater than specified number of minutes.
- **Ask before writing software to device** – Whenever new firmware has to be written into device, user is prompted for confirmation.
- **Ask before writing configuration to device** – Whenever new configuration has to be written into device, user is prompted for confirmation.
- **Automatically validate configuration before save** – Before writing configuration into device, the configuration is automatically validated. The write operation is canceled if the configuration is not valid (see chapter 2.6.4 Configuration Editor function *Validate*).

The page *Environment* / *Terminal* allows user to set colored highlight of **BaWiT**'s debug information displayed in *Terminal* window.
Tab *Environment* / *Connections* contains settings for establishing connection to BaWiT device. Following settings are currently available:

- **Use FTDI D2XX library** – Defines using the library from company FTDI (http://www.ftdichip.com/Drivers/D2XX.htm) for communication with devices. If the setting is not checked, the standard communication by serial port will be used. This communication is part of installed .NET Framework 3.5 SP1 but it behaves defective under particular circumstances (for example disconnection of cable during communication).

Tab *Editor* / *Configuration* contains the following settings:

- **Edit configuration by using the template** – Defines if at request for configuration editing the editing trough set template automatically shows (only if device has set configuration template and this is available on computer hard drive).
- **Automatically switch to monitoring template** – Defines if at request for monitoring presentation the presentation by template will be automatically switched (only if device has set monitoring template and this is available on computer hard drive).

The page *Environment* / *Updates* allows user to enter the Internet address in order to download new updates of the **K2config** application and system configuration files. The address is set to http://bawit.eu/updates/K2config by default, and it is not necessary to change it under normal conditions. At the page, there is also a setting for automatic updates checking and its frequency. Also, when necessary, the user can set proxy server parameters for Internet connection here.