

iSMA-B-AAC20

User Manual

Platform AAC20

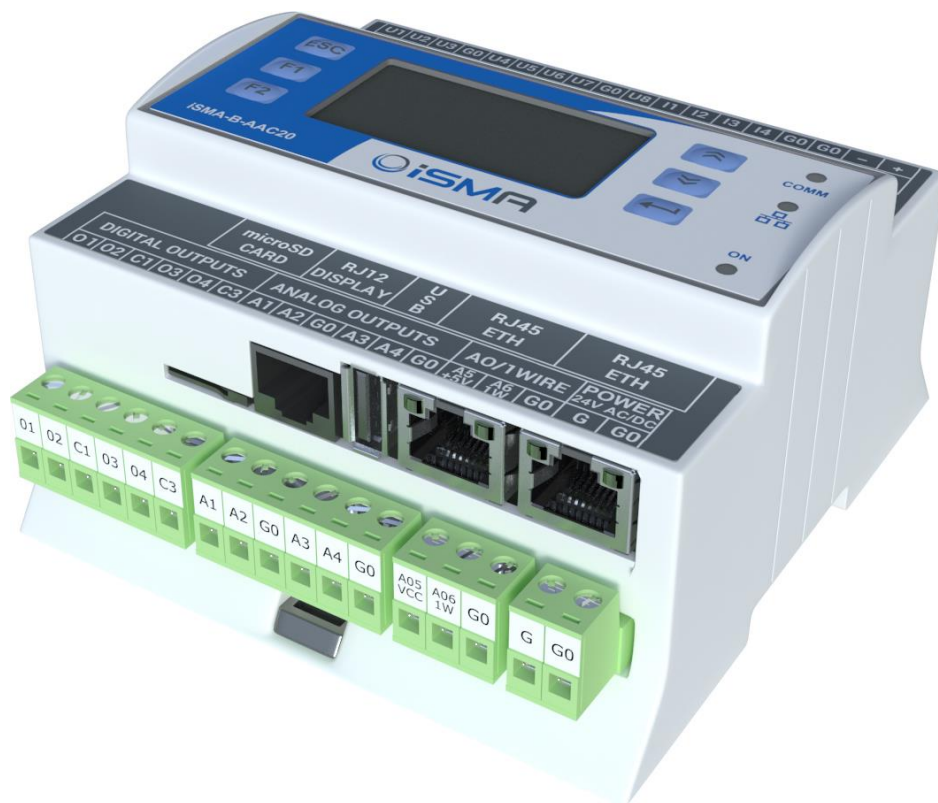


Table of contents

1	Revision history.....	4
2	Sedona Updater	4
2.1	IP connection	5
2.2	RS485 connection.....	5
2.3	Sending file to the device.....	6
2.4	Restoring the device's default settings.....	6
2.5	System Console.....	6
2.6	Firmware upgrade	8
2.7	Checking firmware version.....	9
3	Sedona in iSMA Tool	11
3.1	iSMA Tool Installation	11
3.2	Importing kits to iSMA Tool.....	13
3.3	Installing ismaUI module	Błąd! Nie zdefiniowano zakładki.
3.4	Connecting to Sedona device	14
4	Sedona Kits.....	16
4.1	Kit Manager	16
4.2	Backup/Restore Tool.....	Błąd! Nie zdefiniowano zakładki.
4.3	Application Manager.....	17
5	Licence.....	20
5.1	Licence Manager	20
5.2	Sedona licence – Offline mode.....	21
6	Sedona App component.....	22
6.1	Plat service	23
6.2	Change IP address	24
6.3	Users service.....	25
6.4	Date service	27
7	NV components	28
7.1	NVBooleanWritable.....	28
7.2	NVIntegerWritable	30
7.3	NVNumericWritable	31
7.4	NVMultiStateWritable.....	33
8	Historical	35
8.1	History service	35
8.2	History extensions – Interval.....	36
8.3	History extensions – COV.....	37
8.4	History extensions – COV_Interval	38
9	Alarms.....	39
9.1	Alarm service.....	39
9.2	Alarm Boolean points	40
9.3	Alarm Numeric points.....	41
10	Scheduler.....	42

11 SOX protocol..... 43

12 Local IO 44

 12.1 Adding Local IO components..... 45

 12.2 Local IO Config..... 46

13 One Wire 48

 13.1 Adding OneWire network 48

 13.2 Discovering OneWire sensors 49

 13.3 Adding sensors manually 49

Introduction

This manual contains information about commissioning iSMA-B-AAC20 controller

1 Revision history

Rev	Date	Description
1.1	28.08.2015	<ul style="list-style-type: none"> • First edition
1.2	20.04.2017	<ul style="list-style-type: none"> • Added DHCP option
1.3	05.02.2018	<ul style="list-style-type: none"> • Added DNS option
1.4	27.01.2020	<ul style="list-style-type: none"> • Added chapter about NVMMultiStateWritable • Changed description of Licence Manager • Replaced environment of programming from Workplace to iSMA Tool • Added description of Autologin slot in AAC20 Platform Service

2 Sedona Updater

Sedona Updater is a piece of software designed to administer an operating system of iSMA-B-AAC20 controller. The software is free and can be obtained from your distribution partner or download from GC5 support website: www.support.gc5.pl. The software allows the following actions (using the buttons in the lower part of the window):

- **Send File** – Sends a file into the device, displays the transfer progress bar at the bottom of the window, correct transfer is confirmed with a message "Transfer complete",
- **Reload Firmware** – Uploads a firmware file. This operation must be preceded by a transfer of an adequate firmware files to the device. Upon update, the device shall be rebooted,
- **Device Reboot** - Reboots the device,
- **Default Settings** - Sets default value of IP address, RS485 parameters, platform password,
- **Console** - Displays the operating system console,

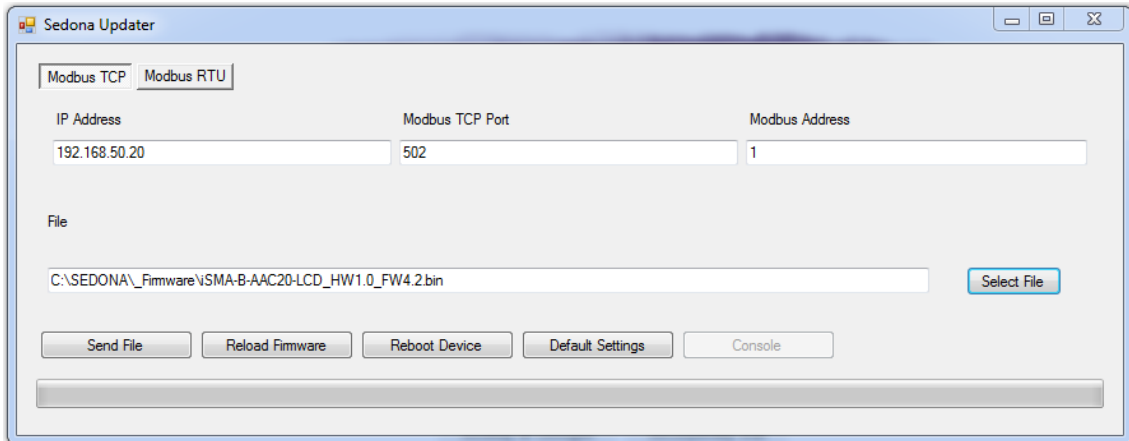
Sedona Updater communicates with the device in two ways:

- **Modbus TCP** - using the Ethernet port,

- **Modbus RTU** - using the RS485 port. In this type of communication, the device must be in the Bootloader mode.

The software allows to connect using IP connection, RS485 port or USB A-to-A cable connection (USB only for power supply or/and diagnostic to read console).

2.1 IP connection



To connect to the device using IP connection, open the SedonaUpdater.exe tool and click “Modbus TCP” button. Activity under the IP connection does not require the device to be in the “Bootloader” mode – it is enough to input: the device’s IP address (default 192.168.1.123), Modbus TCP/IP port number (default 502), and the device Modbus Address (default 1).

Figure 1 Sedona Updater

2.2 RS485 connection

To connect to the device using the RS485 connection, run SedonaUpdater.exe and click the “Modbus RTU” button. In the COM Port text field, enter the RS485 port number of your PC, set the Baud Rate (the default is 115200), and the device Modbus Address (default 1).

In order to operate using the RS485 port connection, you must run “Bootloader” mode on the device.

Bootloader in version with built-in display (iSMA-B-AAC20-LCD):

Step 1 - Disconnect the power and wait for the device to fully shut down,

Step 2 - Hold the Esc button down, and turn the power on again. The screen should now display “Bootloader”.

Step 3 - Restarting exits the Bootloader mode and returns to normal operation.

Bootloader in version without a display (iSMA-B-AAC20):

Step 1 - Disconnect the power and wait for the device to shut down fully,

Step 2 - With the device off, take the cover off and move the DIP Switch no 8 to ON position, turn the power on again. Activation of the Bootloader mode is confirmed by alternating power and communication LEDs.

Step 3 - To exit the Bootloader mode, switch 8 to OFF and restart the device.

2.3 Sending file to the device

The files can be sent using Modbus TCP protocol over an IP connection. This is the recommended method, as it does not require switching the device into the "Bootloader" mode, and provides the fastest way to transfer data.

If you send files using the RS485 port, the device must first be switched into the "Bootloader" mode, as described above. Then, using the "Select File" button, select a file to upload. The upload is done through selecting the right protocol (Modbus TCP/Modbus RTU) and approving the process by pressing the "Send File" button. Progress of the transfer is indicated by a progress bar at the bottom of the screen. After a successful upload, a message pops up saying "Transfer Complete".

2.4 Restoring the device's default settings

Pressing the "Default Setting", restore the factory settings, such as:

- IP address 192.168.1.123,
- Mask 255.255.255.0,
- Default Gateway 192.168.1.1,
- Modbus TCP/IP protocol port 502,
- The password for "platform" user to default 1000.

2.5 System Console

You can view system logs of the controller operating system. There are two options to view system console:

- By web page,
- By USB connection.

Web page system console

To view system console by web page enter controller IP address in web browser and login to system (default user: **platform**, default password **1000**). After login go to “Device Management” section and press “Refresh Log” button. All system logs will be displayed on the window bellow.

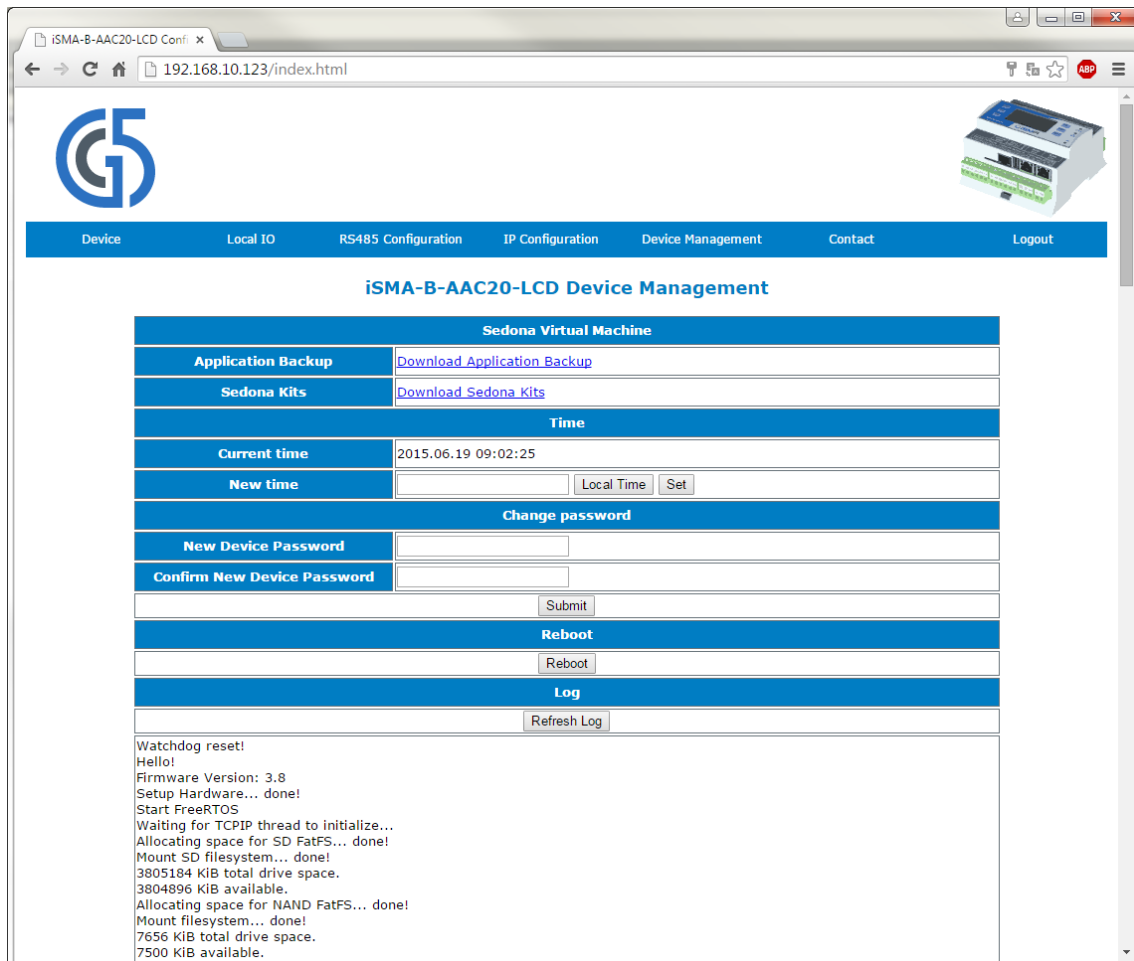


Figure 2 AAC20 Web page Device Manager

USB connection system console

To view the logs, use SedonaUpdater.exe software and connect device using the A-A USB cable. After connection the system automatically recognize controller without installing additional driver (HMI driver is a default one) and Console button will be active. Open System Console window by pressing Console button. There are available below display options:

- Log – print only last written data,
- Full log – print all log memory,
- Date – print controller current date,
- Stats – list of all running task and amount of processing time that has been used by each task,

- CPU usage – current CPU usage.

```

Hello!
Firmware Version: 3.8
Setup Hardware... done!
Start FreeRTOS
Waiting for TCPIP thread to initialize...
Allocating space for SD FatFS... done!
Mount SD filesystem... done!
 3805184 KiB total drive space.
 3804896 KiB available.
Allocating space for NAND FatFS... done!
Mount filesystem... done!
 7656 KiB total drive space.
 7372 KiB available.
RTC Time: 2015.06.19 09:22:13
-- MESSAGE [platform::Platform] Running SVM in Platform Mode.
fopen('app.sab', 'r') OK, emmo=0
file app.sab pointer 280a4910
-- MESSAGE [sys::App] Application starting.
  ** FS serving file: SD:alam.db
f_size(SD:alam.db) 200081
  ** FS serving file: SD:alam.db
fopen('SD:alam.db', 'm') OK, emmo=0
file SD:alam.db pointer 28105558
  ** FS serving file: SD:alam.db
f_size(SD:alam.db) 0
  ** FS serving file: SD:alam.db
f_size(SD:alam.db) 200081
fopen('app.sab', 'w') OK, emmo=0
file app.sab pointer 2814fc50
-- MESSAGE [sys::App] App saved : f_size(app.sab) 29195
29195 bytes)
fopen('db/history/his1.db', 'm') failed, emmo=5
-- MESSAGE [platform::Platform] Registered Version.
-- MESSAGE [sox::SoxService] Started port=1876

```

log full log date stats cpu usage

Figure 3 Sedona updater console view

This connection method is recommended when user cannot connect using IP or RS485 ports or don't know connection parameters for example IP address.

2.6 Firmware upgrade

Firmware can be updated by uploading the device with an appropriate firmware file, and reloaded with new firmware. The following files are necessary to upgrade the firmware (all files are available on GC5 support website: www.support.gc5.pl):

- **SedonaUpdater.exe** – Software can send files to controller and do system actions,
- **Firmware file** – Binary file with firmware code,
- **The latest kits** – All kits must be updated to the current version, they can be downloaded and imported as one bundle zip file,
- **Default app.sab file** – (optional) Default Sedona application file, adapted to the current firmware version,
- **Default kits.scodes** – (optional) Default Sedona kits image, adapted to the current firmware version.

Step 1 – Download files from GC5 support website: www.support.gc5.pl to local PC drive,

Step 2 – Upgrade kits files in iSMA Tool. (see Importing kits to iSMA Tool chapter),

Step 3a - Send files by IP connection – Using IP connection controller will automatically run Bootloader mode.

Open SedonaUpdater.exe and set up communication parameters :

- Tab Modbus TCP,
- IP Address – Controller IP address (default for new devices 192.168.1.123),
- Modbus TCP/IP port – Controller Modbus TCP/IP port (default 502),
- Modbus Address – Controller Modbus address (default 1).

Step 3b - Send files by RS485 connection.

WARNING! Using RS485 port controller must be manually switched to Bootloader mode (see RS485 connection chapter).

Open SedonaUpdater.exe and set up communication parameters :

- Tab Modbus RTU,
- COM port – PC RS485 COM port number,
- Baud Rate – Controller RS485 port baud rate (default 115200),
- Modbus Address – Controller Modbus address (default 1).

Step 4 - Select the firmware file (.bin file) using the "Select File" button and send to the controller using the "Send File" button, after success transfer you will get message "Transfer OK".

Step 5 – Send default app.sab and kits.scode files (optional).

WARNING! This operation will erase controller application, not recommended if you want to keep your application or backup application on your PC drive.

Select app.sab file and send to the controller after "Transfer OK" message select second file kits.scode and send to the controller.

Step 6 – Reboot controller using Reload Firmware button. The controller will load new firmware and restart.

Step 7 – Check controller firmware version (see Checking firmware version) if it does not change repeat steps of this chapter.

2.7 Checking firmware version

There are three options to check controller current firmware version.

By iSMA Tool

To verify the current firmware version, log in to the device and go to the Plat component. The Plat component is located in the Service folder, directly below the App component (Device -> App -> Service -> plat). Plat contains the "Firmware Version" slot identifying the working firmware version.

By Web page

To verify the current firmware version, open browser, enter controller IP address and enter user platform and 4 digits password (default 1000, it can be changed in Sedona -> App -> Service -> plat component in slot "Device Password"). Firmware version is displayed on Device tab in Basic Information.

By LCD system menu

To verify the current firmware version, login to system menu by holding F1 button, enter LCD platform user 4 digits password (default 1000 or from plat component), and open Info page.

3 Sedona in iSMA Tool

As a significant part of end-to-end iSMA solution, iSMA Tool gives customer a convenient way to create and manage custom applications for Sedona-based iSMA controller.

iSMA Tool now covers all needs one could have to create a perfect application: wire sheet for convenient visual programming, Property sheets for details, kit management, real time monitoring of system states and slot values, logs and historical data, deployment and backup.

3.1 iSMA Tool Installation

iSMA Tool is a software created for modern Microsoft Windows system, such as Windows 10. The oldest supported version of the operating system is Windows 7. iSMA Tool is delivered as a compressed folder, which needs to be extracted in a chosen location and hard drive, unless the access to the extracted folder is restricted by the system (e.g. "Program Files" is not a recommended location).

The extraction provides the folders and additional files described below . One of the files is "iSMATool.exe", used to run iSMA Tool.

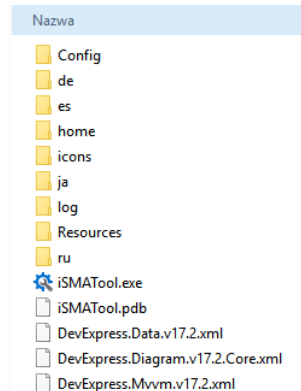


Figure 4 iSMATool.exe file

The extracted folders serve the following functions:

- Config - a folder containing a record of user's individual settings regarding window location and other iSMA Tool work settings, such as the language chosen for iSMA Tool interface.
- home - a folder, where all the data created by user are saved, i.e. device backups, applications, etc. It's also a folder, where Kits library, available in iSMA Tool, is located.
- icons - a folder with graphical files such as iSMA Tool interface icons.

- Localization - a folder with text files responsible for iSMA Tool's multilingualism.

log - a folder, where logs of iSMA Tool, which also appeared in Console window, are saved. When contacting GC5 technical support, it is advised to copy the last file with logs from that folder

- de, es, ja, Resources, ru – are folders with system libraries.

To properly install and work with iSMA Tool the User's computer must meet the following minimal requirements:

- Processor (CPU): Intel Core i3-3xxx or equivalent;
- Memory: 4GB RAM;
- Storage: 50 GB internal hard driver;
- Ethernet 100 Mbit or 1Gbit NIC;
- MS Windows 7 (recommended MS Windows 10);
- .NET Framework 4.6.2 or higher

WARNING! When iSMA Tool is run for the first time, you will be asked to accept EULA licence. Only accepting the licence will allow you to run the program. Lack of acceptance closes iSMA Tool.

Note: For iSMA Tool to work properly it needs to be run periodically, at least once a month, on a computer connected to Internet for about an hour, depending on data transfer rate. This will enable iSMA Tool to automatically download the latest data, such as Kits and updates.

iSMA Tool is a portable software. It is "transferable" and can be installed on a portable data storage device such as USB memory stick. This allows iSMA Tool to be run directly from portable data storage device on any PC, including offline ones.

3.2 Importing kits to iSMA Tool

To import files to iSMA Tool it is recommended to use “Sedona” tab from Tools menu and choose “Import Sedona files”. The latest kits and kits bundle can be downloaded from support.gc5.pl web site. There are two options to import kits to Workplace:

- Singly – by choosing single kit file with .kit extension

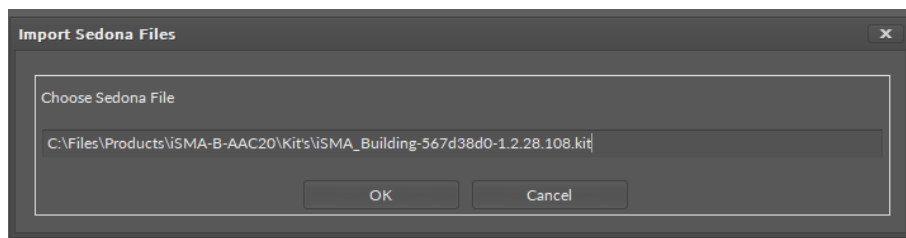


Figure 5 Sedona environment files installer single file

Open Sedona Tab from Tools menu in iSMA Tool, mark “Install Sedona files” option and select kit file. Press next button to finish importing process. This process copies this file to Sedona kits folder.

- Groups – by choosing prepared bundle file with zip extension

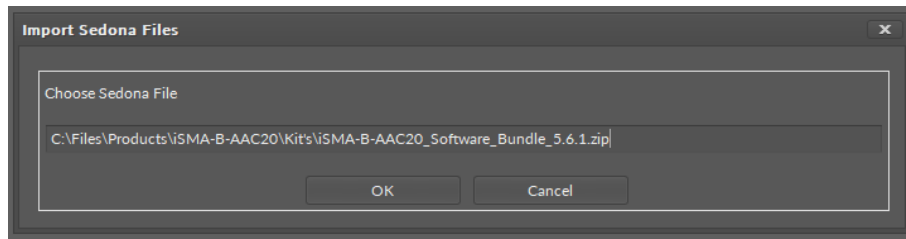
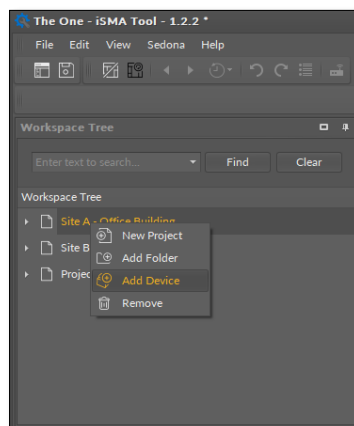


Figure 6 Sedona environment files installer software bundle

Open Sedona Tab from Tools menu in iSMA Tool, mark “Install Sedona files” option and select zip file. Press next button to finish importing process. This process copies all kit files from zip archives to Sedona kits folder.

3.3 Connecting to Sedona device

In order to connect to the device, select the following in the menu: Right click on the project folder and select “Add Device”.



After selecting the above option, a device login window pops up.

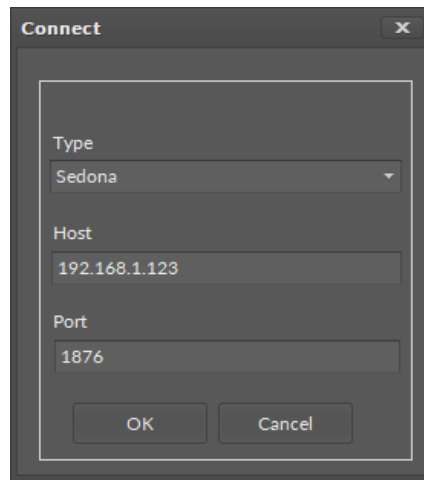


Figure 7 Sedona Device connection window

By default, a new device should have the following values:

- IP address – 192.168.1.123,
- Port – 1876,
- User – admin,
- Password – empty box (no characters).

The connection parameters can be changed after connect to controller:

- IP address – in plat component (see Change IP Address chapter),
- Sox Port - in Sox service (see Sox chapter),
- Admin password/new user – in user service (see user chapter).

4 Sedona Kits

Sedona environment kits are the sets of components for programming applications. The device is provided with a set of basic kits and in order to develop applications, the set should be uploaded to your device. To add the new or update the kit, it must first be imported into iSMA Tool (see chapter Importing kits to iSMA Tool). In device, the set of kits is stored in file kits.scode. To manage controller kits and compile kits.scode the iSMA Tool use tool Kit Manager (see Kit Manager chapter).

4.1 Kit Manager

The Kit Manager Tool is used to manage kits on Sedona devices. The kits.scode file is a file which contains a set of kits in a Sedona Framework code image file with an .scode extension. The scode image is executed by the SVM (Sedona Virtual Machine) in the device. Using the Kit Manager, you can add and remove kits, as well as change versions of kits deployed in the scode running on the device. Also, you have the option to force the kits.scode to be rebuilt. The kits.scode must be rebuilt whenever a change affects the schema. This ensures such changes are included in the recompiled binary .scode image. When you make any kit selection changes in the Kit Manager, it automatically selects and check this option. The only time you would select this option is when you want to force the scode to be rebuilt (even if no change was made to kit selections). This option is most useful in a development environment where the actual code in a kit might change while the kit version and checksum do not. In this case, no change is required in the Kit Manager but the scode must be regenerated in order to deploy the new code on a device.

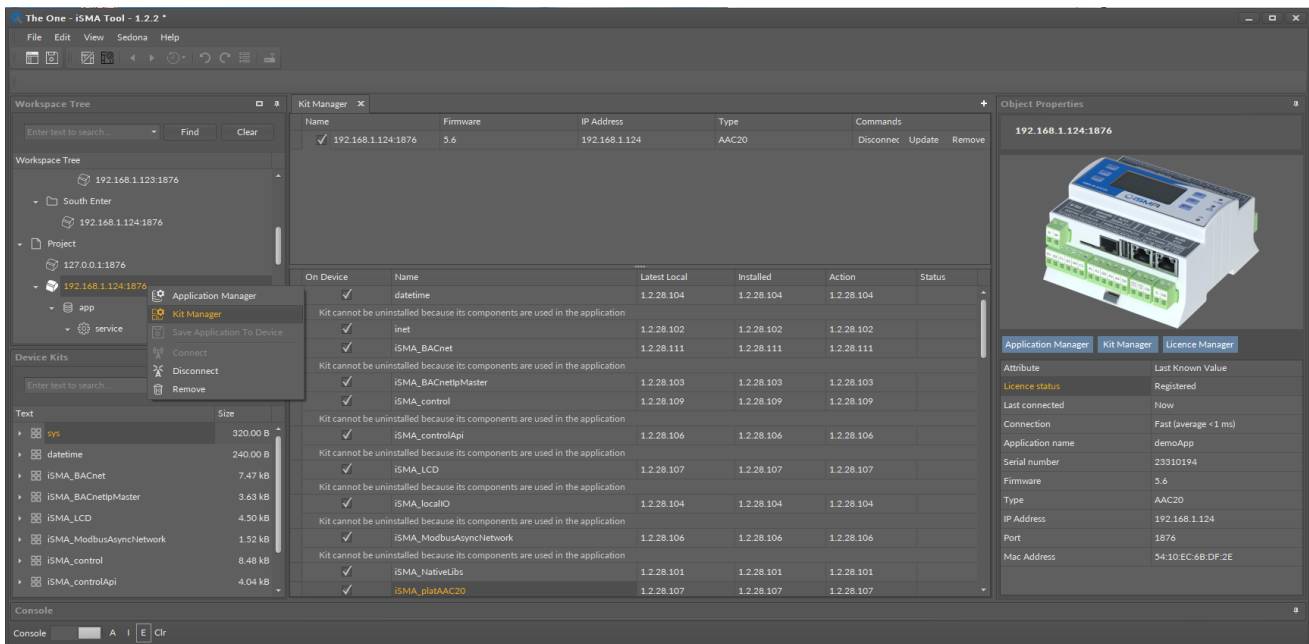


Figure 8 Kit Manager

The “Action” drop-down list options can be one or more of the following:

- Keep at [the current version],
- Upgrade to [a newer version],
- Downgrade to [an older version].

The newer/older kit version levels shown in the Action drop-down menu are available in iSMA Tool. They are stored in the iSMA_Tool\home\kits directory.

In order to remove a kit, remove all its components from the application first. If the application does not contain any kit components, its icon in Kit Manager shall be replaced with a check box. To delete it, you must deselect it in the “name” column (the “action” column will now show “remove”). The following steps are the same as in the case of installation.

4.2 Application Manager

The App Manager tool is used to manage app.sax file. The app.sax is Sedona Application XML file which is a simple XML representation of a Sedona Framework app that is easily generated and consumed by Sedona Framework software tools.

Application Manager allows to:

- manage locally saved applications in iSMA Tool
- load applications to a specific device

- download applications from a specific device

WARNING! Loading an application to a device will overwrite the application already present in the device.

WARNING! Each loading of application to a device will require restarting the device.

There are a few ways to initiate Application Manager, by:

- initiating Application Manager from the context menu
- initiating Application Manager from Object Properties window

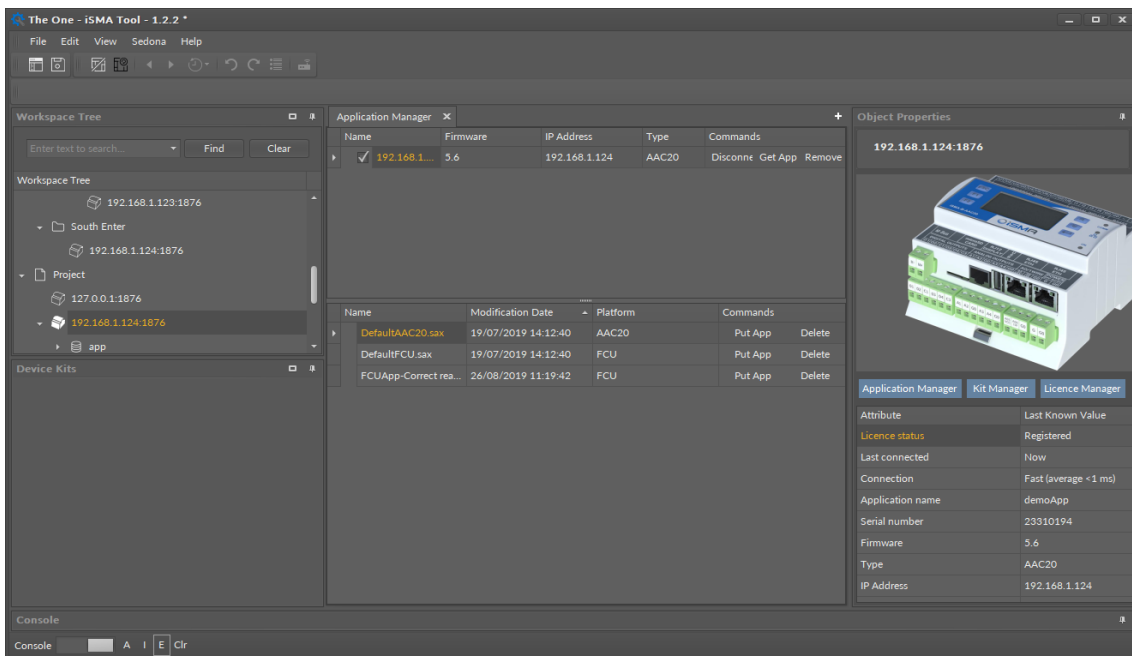


Figure 9 Application Manager

After connecting with a chosen device we can download its application by pressing the “Get App” button, see picture below.

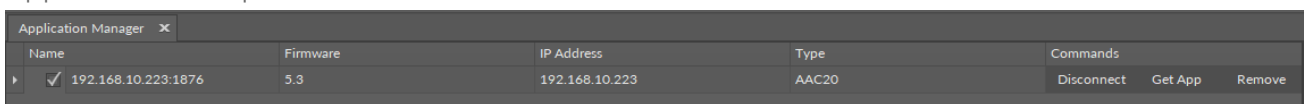


Figure 11 “Get App”

Note: Downloading an application from a device does not stop or restart the device. Downloading does not disrupt the device's work.

Note: Application downloading process is monitored in detail in Console window, which helps analysing the situation, where downloading the application couldn't be completed.

Application uploading procedure needs to begin with defining the device to work with Application Manager by selecting a checkbox for a particular device in the upper part of manager view.

Next one of the available applications in the lower part of the manager needs to be defined by pressing "Put App" button in "Commands" column.

WARNING! Pressing "Put App" button will result in overwriting of the application already installed in device and resetting the device in the final phase of the process.

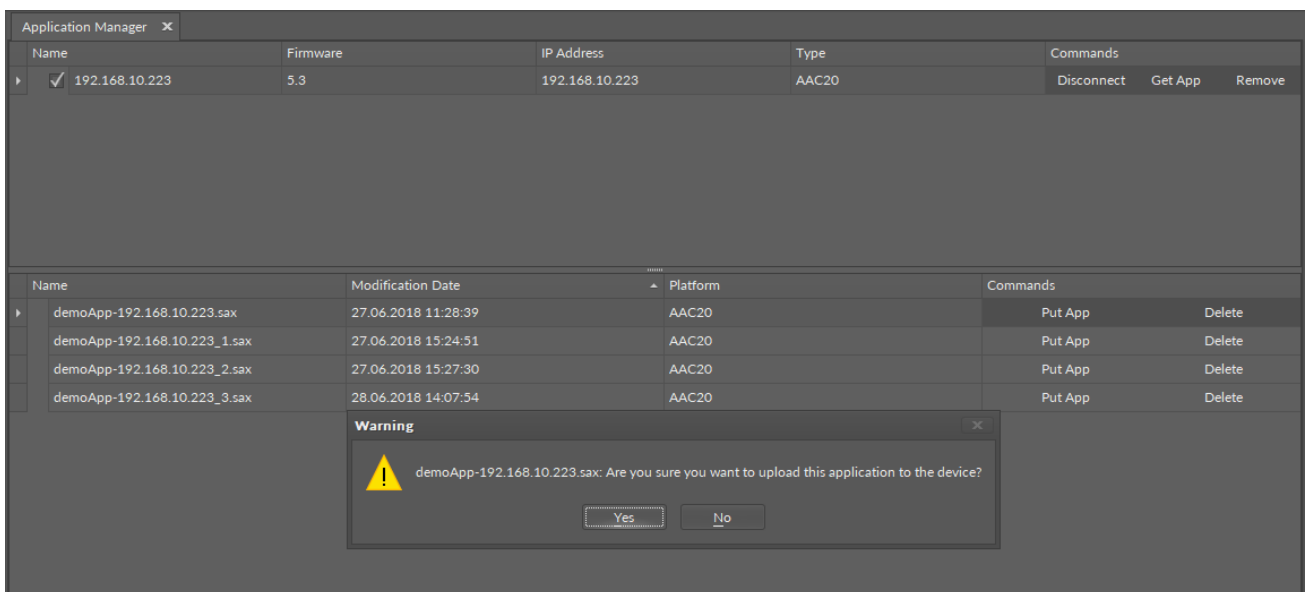


Figure 12 Put App

5 Licence

Sedona controllers have built-in licence mechanism. All out of box controllers have installed standard licence version and it is ready to work. But in case of licence changing or after memory clear the licence must be re-uploaded. There are two modes of doing it::

- Online mode,
- Offline mode.

The licences is generated by serial number or MAC address. The proper file name format is "MAC address without dots".dat.

5.1 Licence Manager

Using Sedona Online mode iSMA Tool can connect to licence server, download and save it on the controller. To open the licence manager double click on the controller and in the Object Properties window choose Licence Manager. In the pop-up window it is possible to check device licence status and invoke one of two actions:

- Upload licence from remote database
- Upload licence from file

To download licence from server please use "Get Licence from Remote Database". During download process will pop-up windows with status progress. After completed download operation use "Get Licence from Local Database" to send licence file to the controller. After device reboot Sedona will boot with new licence.

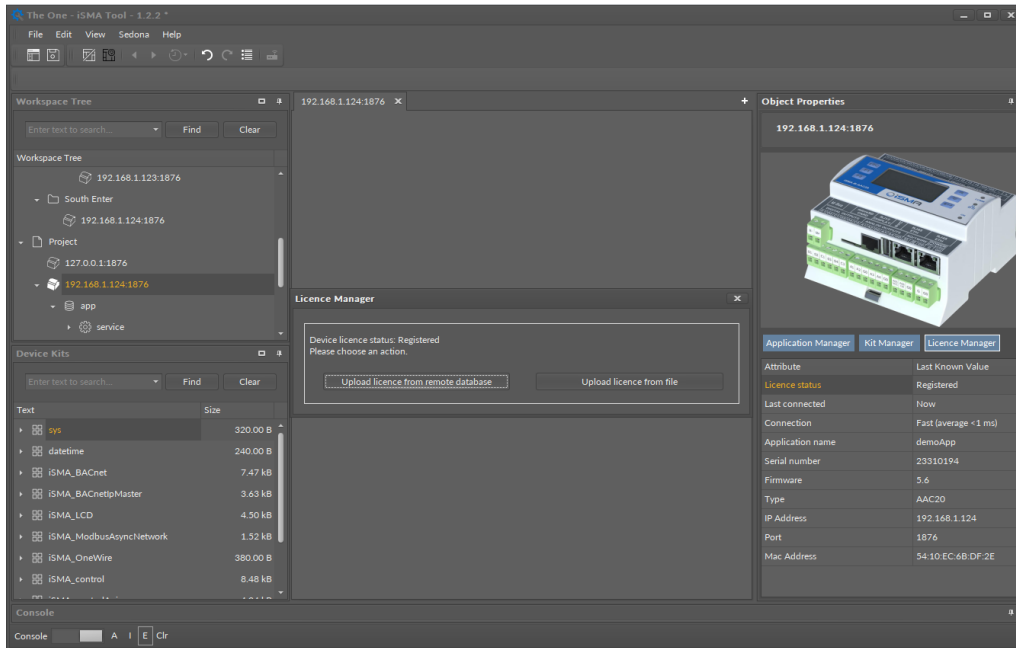


Figure 10 License Manager view

5.2 Sedona licence – Offline mode

Licence in offline mode can be sent to controller as a standard file using “Upload licence from file” in iSMA Tool or using SedonaUpdater software. Licence file can be download using ismaUI to local database, copy from other PC or sent by email from local iSMA distributor. Open SedonaUpdater.exe software and choose licence file (proper file format is “MAC address (without dots)”.dat). After success upload reboot controller using “Reboot Device” button. (For more information about sending files see “Sending file to device” chapter). After device reboot Sedona will boot with new licence.

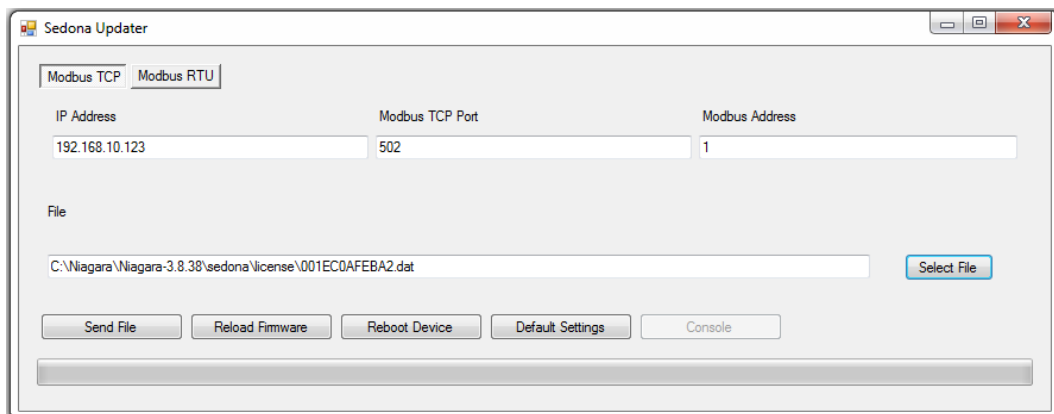


Figure 11 Sedona Updater license file sending

6 Sedona App component

Application consists of services and components available from the Palettes. Components are processed in every working cycle of the device. Services cover certain components enabling system functions, such as user management. All items should be placed under the App main component. When the application is modified, the App icon is displayed with reminder that application should be saved. There is a possibility to turn on the autosave of the application.

The component has the following actions available under the right mouse button click:

- Save – saves Sedona application in the device's flash memory,
- Restart – restarts application (Sedona Virtual Machine),
- Reboot – reboots device.
- Quit – Closes sox connection
- Hibernate – Not active

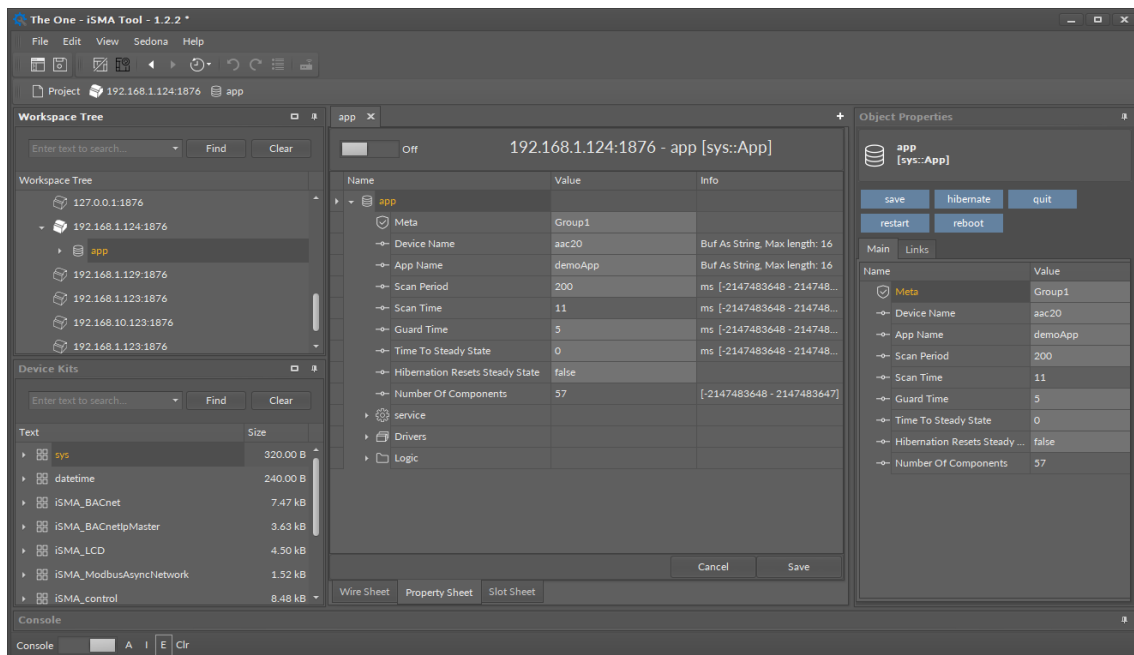


Figure 12 Sedona application structure view

- Device Name – Device Name text,
- App Name – Application Name text,
- Scan Period – One cycle execution time,
- Scan Time – Real time of one cycle execution,
- Guard Time – Reserve time to finish system tasks,
- Time To Steady State - Time from app start to steady state,
- Hibernation Resets Steady State – not active in AAC20.
- Number Of Components – Number of components used in application.

6.1 Plat service

The Plat service is a component which shows device's main parameters. This component is placed under the Service folder and is associated with device hardware. The component has to be placed under Service component.

The component has the following actions:

- Restart – restarts the application (Sedona Virtual Machine),
- Reboot – reboots the device,
- Copy From Nv To Default – copies values from Out to Default slot in all NV components (see NV component chapter),
- Copy From Nv To User – copies values from Out to User slot in all NV components (see NV component chapter),
- Copy From Default To NV – copies values from Default to Out slot in all NV components (see NV component chapter),
- Copy From User To NV – copies values from User to Out slot in all NV components (see NV component chapter),
- Set All Nv In Auto – sets all NV components in auto mode.

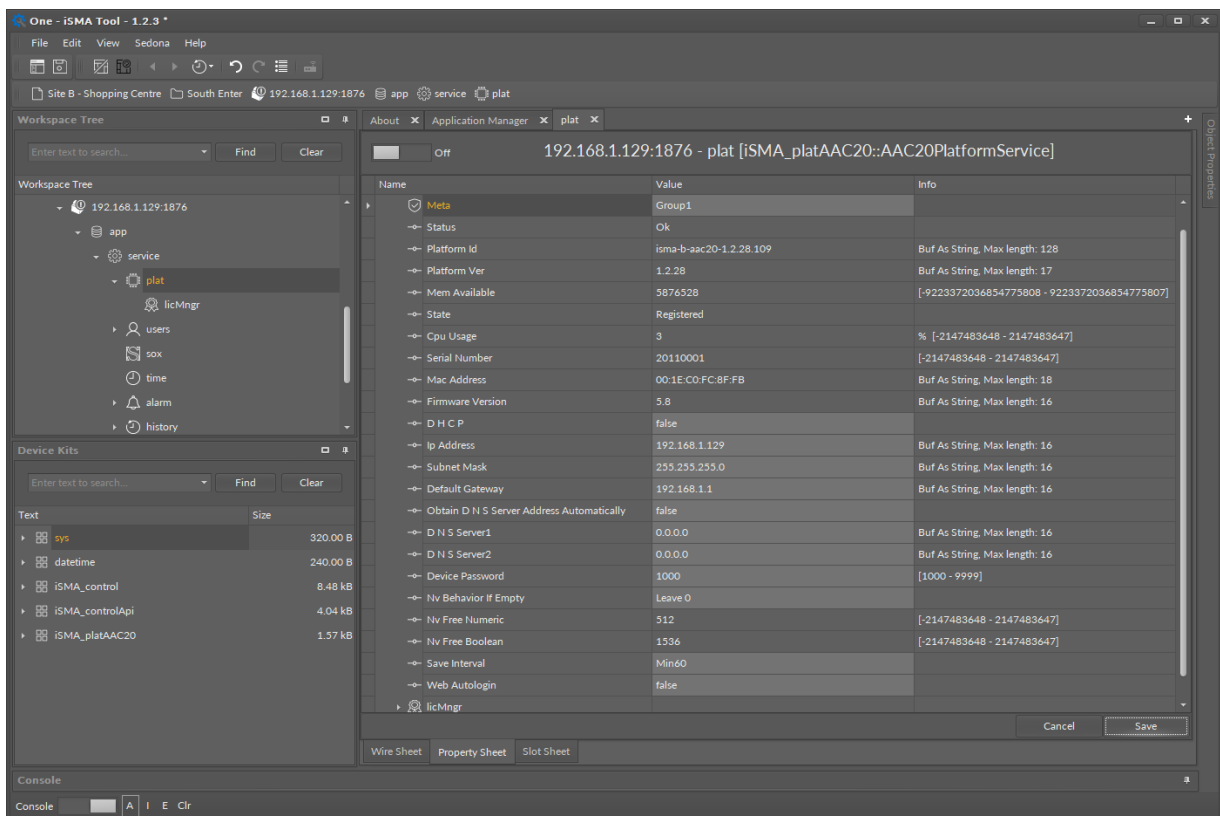


Figure 13 Plat component view

The component has following slots:

- Status – Platform status,
- Platform ID – Platform ID,
- Platform Ver – Platform Version,
- Mem Available – Controller RAM memory available,
- State – Licence status,
- Cpu Usage – CPU usage from last 5 seconds,
- Serial Number – Device unique serial number,
- Mac Address – Ethernet mac address,
- Firmware Version – Controller firmware version,
- DHCP – Address setting by DHCP server
- IP Address – Device IP address (when DHCP is enable this slot is showing current IP address),
- Subnet Mask – Device subnet mask, (when DHCP is enable this slot is showing current IP mask),
- Default Gateway – Device default gateway Ip address, (when DHCP is enable this slot is showing current gateway IP address),
- Obtain DNS Server Address Automatically – when DHCP is true, this option allow to get DNS IP address automatically form DHPC or manually,
- DNS Server 1 – First DNS server IP address,
- DNS Server 2 – Second DNS server IP address,
- Device Password – password for “platform” user,
- Nv Behavior If Empty – Non-Volatile components after copy output value (Options Leave 0, copy value from Default slot),
- Nv Free Numeric – number of available Numeric Non-Volatile components,
- NV Free Boolean – number of available Boolean Non-Volatile components,
- Save Interval – application auto save interval
- Web Autologin - slot for enabling/disabling password protection to access the Web Page

6.2 Change IP address

The device has the built-in 2 Ethernet ports working in switch mode. By default, the new device's address is set to 192.168.1.123, subnet mask to 255.255.255.0 and default gateway to 192.168.1.1. The device offers the option to change the IP address at three levels: the iSMA Tool application level, the website, and the device display (only devices with built-in display). All those levels allow to set static IP address or enable option to addressing by DHCP server.

Changing the IP address in the application

To change the IP address in the application, log into the device (admin or user with authorisation to change the address in the plat component). Then go to "plat" (Device -> App -

> Service -> plat) and make changes to the slots: IP Address, Subnet Mask, Default Gateway, and save the application (App Component -> Save action). To make changes, restart the device using the Reboot action in the App component. The device shall restart with the new IP address.

Changing the IP address on the website

To change the IP address of the device, go to website by entering the current IP address in the URL field of any browser. Log in, using platform username and a numerical password from the "Device password" slot in the plat component (default - 1000). Then, go to IP Configuration tab and make appropriate changes in the following slots: IP Address, Subnet Mask, Gateway (Slot Mac Address is read-only), click Submit to save the application and restart the device using the Reboot button. The device will start up with the new IP address. Attention! After IP controller reboot, change URL address in browser.

Changing the IP address using the display (only devices with built-in display)

To change the IP address from the display, hold the F1 function key down until the screen shows "Enter password". Then input a numeric password (slot "Device password", plat component, the default is 1000). In the system menu, use arrows on the right side to go to the Network Config line and press "Enter". Then select the appropriate line and press "Enter" again. The changes make the arrow on the right side (+ / -) move to the next position by pressing the "Enter" key. To validate the change, hold "Enter" down until transferred to the system menu and then select "Reboot". The device shall start up with the new IP address.

6.3 Users service

Users Service is the service responsible for supporting users. Here is where users can be added and removed as well as their access rights to individual components can be defined. Each Sedona application component has a Meta slot, used to assign it to one or more groups. Sedona has 4 predefined groups.

Warning! In User service only Sedona users can be modified. User platform is the control operating system and can be modified (password change) only from plat component, built in display or from controller web page.

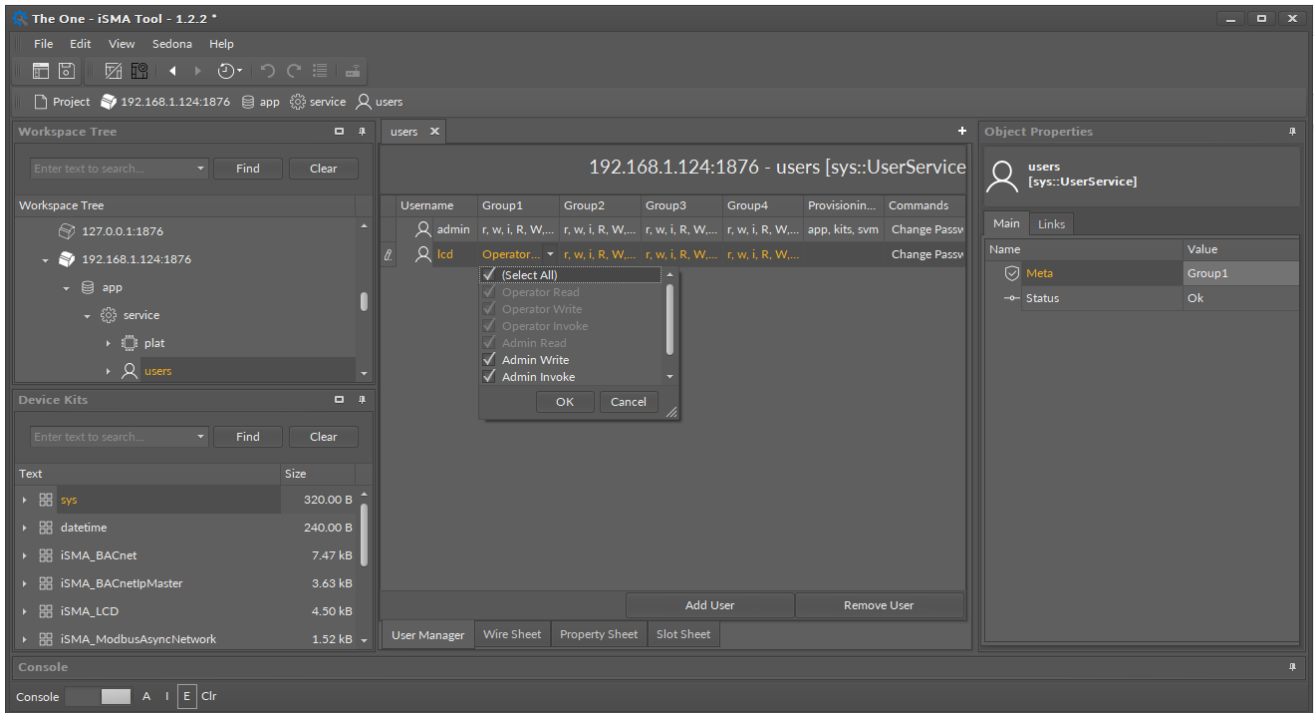


Figure 14 User Manager view

Users can have the following types of rights:

- Operator Read – Read components, read values of operator properties,
- Operator Write – Change values of operator properties,
- Operator Invoke – Invoke operator actions,
- Admin Read – Read values of properties, read links, generate components links,
- Admin Write – Change values properties, add components, sort dub components, rename components, generate links to the components, delete links to the components,
- Admin Invoke – Invoke admin actions of the components,
- Admin User – User management (read, write, edit, delete).

Provisioning Permissions:

- Can provision app – can read/write app.sab file,
- Can provision app – can read/write kits.scode file,
- Can provision app – can read/write SVM files.

For devices with built-in LCD, displaying components such as action editing/invoking is defined by assigning the given component to a group and defining access rights for user from this group.

6.4 Date service

Date service is the component, which handles the built-in real time clock (RTC). The clock can be synchronized with a local PC's clock, or by using iSMAsox driver, with Niagara device.

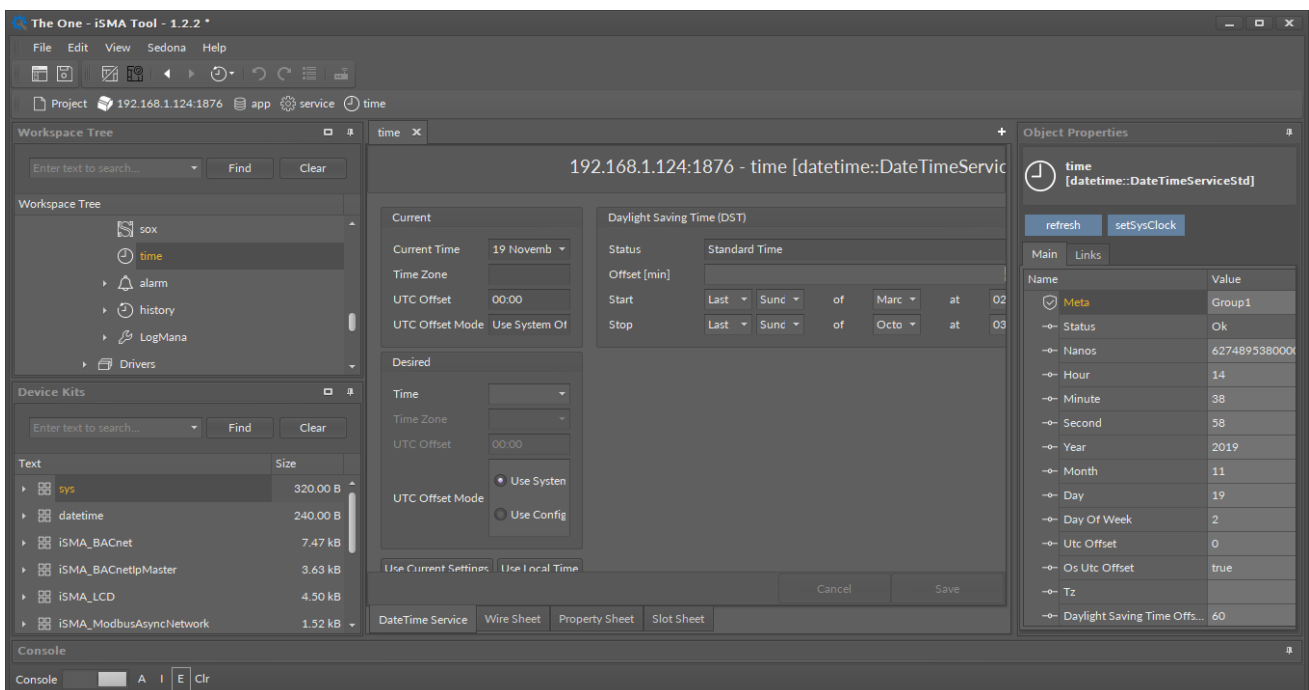


Figure 15 Time Service view

7 NV components

NV Components (Non-Volatile) are those components whose value can be recorded in an EEPROM device's non-volatile memory. Whenever the device is restarted or the power is down, the values of NV components remain saved. The device has three types of NV components, broken down by the type of variables they support. They include:

- Boolean variables - component NVBooleanWritable
- Integer variables - component NVIntegerWritable
- Numeric (float) variables - component NVNumericWritable

NV components can operate in the Auto mode (the "In" slot values are transferred to the "Out" slot) or in the Hand mode (the "Out" value is entered manually by the user and can't be changed by the application).

There is available 512 memory cells for numeric (float/numeric and integer) values and 1536 memory cells for boolean values in iSMA-B-AAC20. NVNumeric and NVInteger components use one numeric cell memory for Out value and one boolean memory cell for Auto / Hand switch mode. NVBoolean components use two boolean memory cells one for Out value and one for Auto / Hand switch mode.

Since the values of the components are not stored in Sedona application but in the non-volatile memory of the device, when an application is copied between two devices, output values are not saved and will assume the values stored in the local EEPROM memory. To copy NV components to another device along with their values (e.g. setpoint), use global actions of the plat component:

Step 1 - Use global action "Copy From NV The Default / Copy From NV To User"

Step 2 - Save the application and copy it to another device

Step 3 - Use global action on the target device "Copy From Default To NV / Copy From User To NV"

7.1 NVBooleanWritable

NVBooleanWritable is the component that stores the output value in non-volatile memory of EEPROM device. After rebooting the device or the power failure, the component value is restored from this particular memory. iSMA-B-AAC20 has 1536 Boolean memory cells. The occupied space meter for EEPROM is embedded in the plat component. NVBooleanWritable

component occupies two Boolean memory cells (value of the component, and position of the Auto / Hand switch).

The NVBooleanWritable has the following actions:

- Set - this option sets User slot and In slot if there is no link on In slot
- Set In Hand - this option sets the value on Out slot and blocks changing from any other slots,
- Set In Auto - this option switches off the Hand mode and sets Out slot according to the In slot's value.

The NVBooleanWritable component allows adding extensions, such as:

Alarm Change Of State - to generate alarms when the component is in a particular state,

History - Record the value at a point in time, over a defined period of time, a change by a certain value, or in both of these cases,

Totalize - counting the time when the value of a point in a particular state.

The NVBooleanWritable component is also used to integrate Boolean variables from various sources. This is done using the "reverse following the link" function. The Out slot is connected to the In slots of various protocols, for example LCD or Modbus variable. When changing a value in one of the components, the device will perform the Set action on the NV component to synchronize the values in all the connected components.

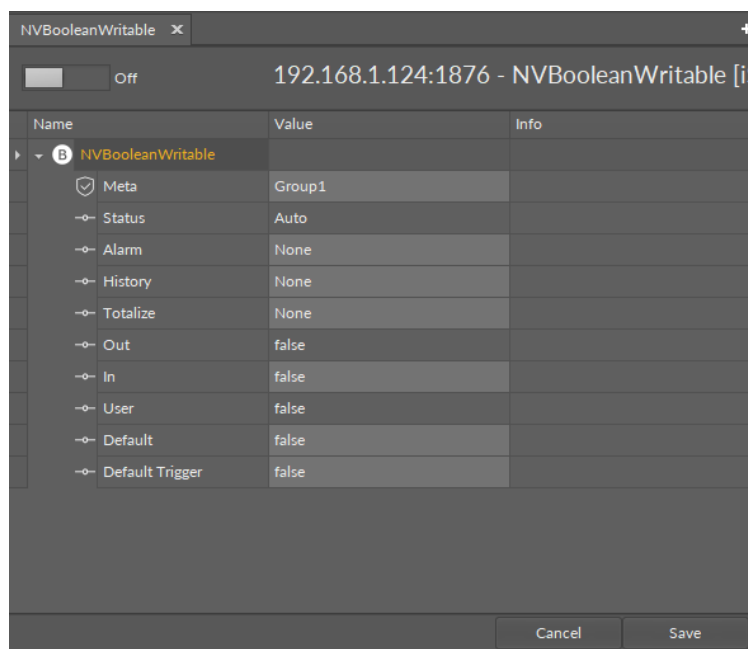


Figure 16 NV Boolean Writable view

The component has the following slots:

- Status – Point actual status Auto / Hand,
- Alarm – Alarm extension,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

7.2 NVIntegerWritable

NVIntegerWritable is the component that stores the output value in non-volatile memory of EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. iSMA-B-AAC20 has 1536 Boolean memory cells and 512 numeric cells. Space meter of the occupied EEPROM's memory is located in the plat component. NVIntegerWritable component occupies a single memory cell of the Numeric type (component value) and one memory cell of the Boolean type (switch position Auto/Hand).

NVIntegerWritable Component allows to add extensions such as:

History - Record the point in time in a defined period of time, to change a certain value, or in both of these cases,

Totalize - counting the time when the point value is not zero.

The NVIntegerWritable has the following actions:

- Set – this option sets User slot and In slot if there is no link on In slot ,
- Set In Hand – this option sets value on Out slot and block changing from any other slots,
- Set In Auto – this option switches off Hand mode and sets Out slot according to In slot value.

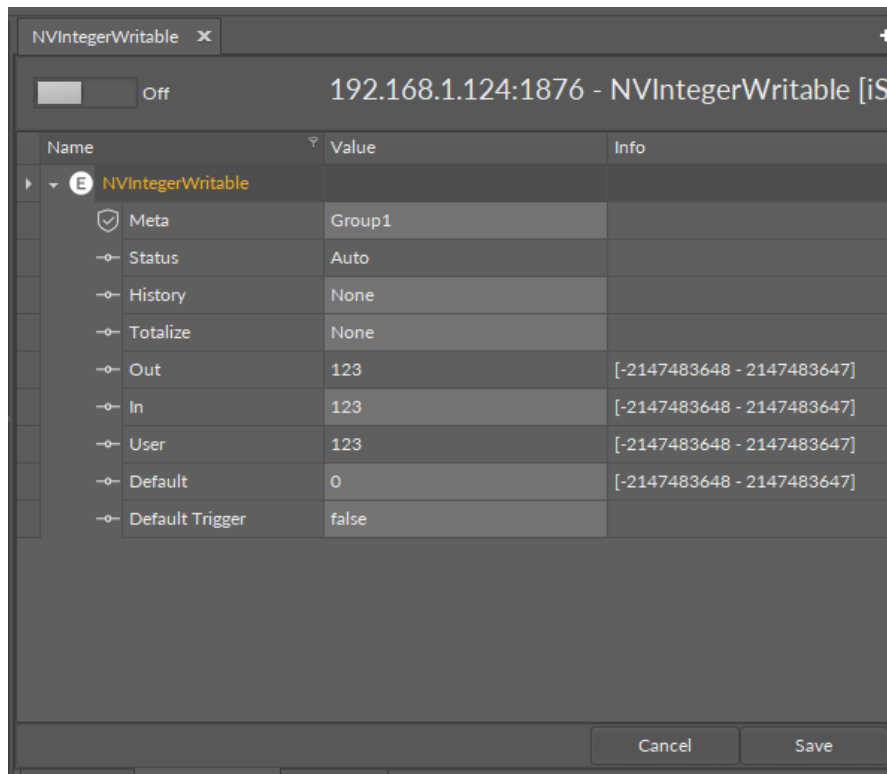


Figure 17 NV Integer Writable view

The component has the following slots:

- Status – Point actual status Auto / Hand,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

7.3 NVNumericWritable

NVNumericWritable is the component that stores the output value in non-volatile memory of EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. iSMA-B-AAC20 has 1536 Boolean memory cells and 512 numeric cells. Space meter of the occupied EEPROM's memory is located in the plat component. NVNumericWritable component occupies a single memory cell of the Numeric type (component value) and one memory cell of the Boolean type (switch position Auto/Hand).

NVNumericWritable has the following actions:

- Set – this option sets User slot and In slot if there is no link on In slot ,
- Set In Hand – this option sets value on Out slot and block changing from any other slots,
- Set In Auto – this option switches off Hand mode and sets Out slot according to In slot value.

NVNumericWritable Component allows you to add extensions such as:

History - Record the point in time in a defined period of time, to change a certain value, or in both of these cases,

Totalize - counting the time when the point value is not zero.

NVNumericWritable component is also used to integrate Numeric (float) variables from various sources. This is done using the "reverse following the link" function. The Out slot is connected to the In slots of various protocols, for example LCD or Modbus variable. When changing the value in one of the components, the device will perform the Set action on the NV component to synchronize the values in all the connected components.

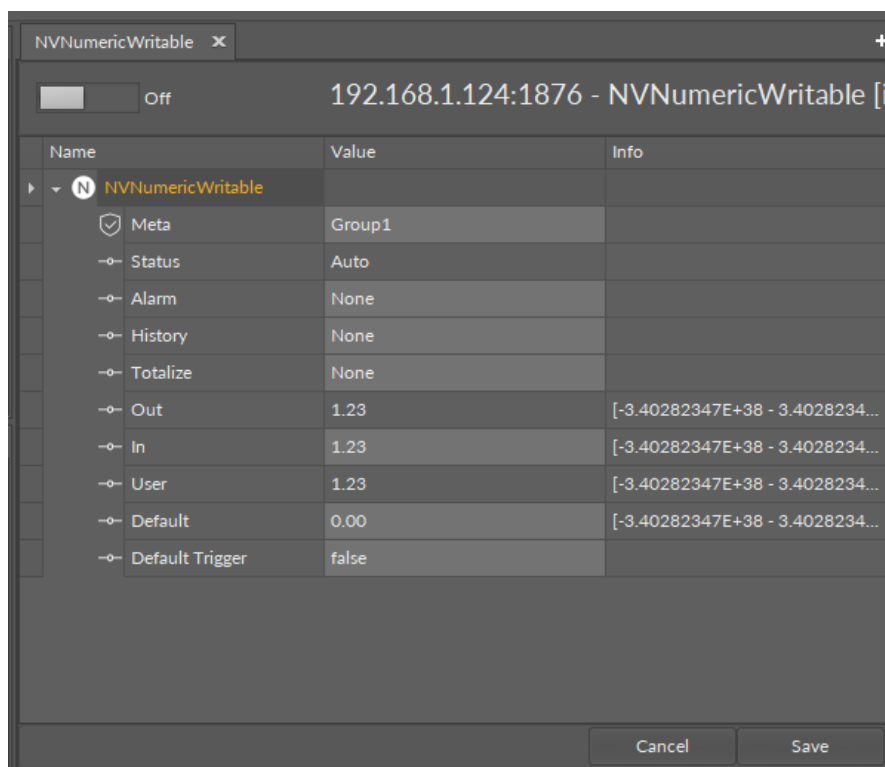


Figure 18 NV Numeric Writable view

The component has the following slots:

- Status – Point actual status Auto / Hand,

- Alarm – Alarm extension,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

7.4 NVMultiStateWritable

NVMultiStateWritable is the component that stores the output value in non-volatile memory of EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. iSMA-B-AAC20 has 1536 Boolean memory cells and 512 numeric cells. Space meter of the occupied EEPROM's memory is located in the plat component. NVMultiStateWritable component occupies a single memory cell of the Numeric type (component value) and one memory cell of the Boolean type (switch position Auto/Hand).

NVMultiStateWritable has the following actions:

- Set – this option sets User slot and In slot if there is no link on In slot ,
- Set In Hand – this option sets value on Out slot and block changing from any other slots,
- Set In Auto – this option switches off Hand mode and sets Out slot according to In slot value.

NVMultiStateWritable Component allows you to add extensions such as:

Alarm Out Of Range – generating alarms whenever the value of the component exceeds a given value,

History - Record the point in time in a defined period of time, to change a certain value, or in both of these cases,

Totalize - counting the time when the point value is not zero.

In NVMultiStateWritable component it is possible to define sixteen string outputs depending on the Integer value on the Input of the component.

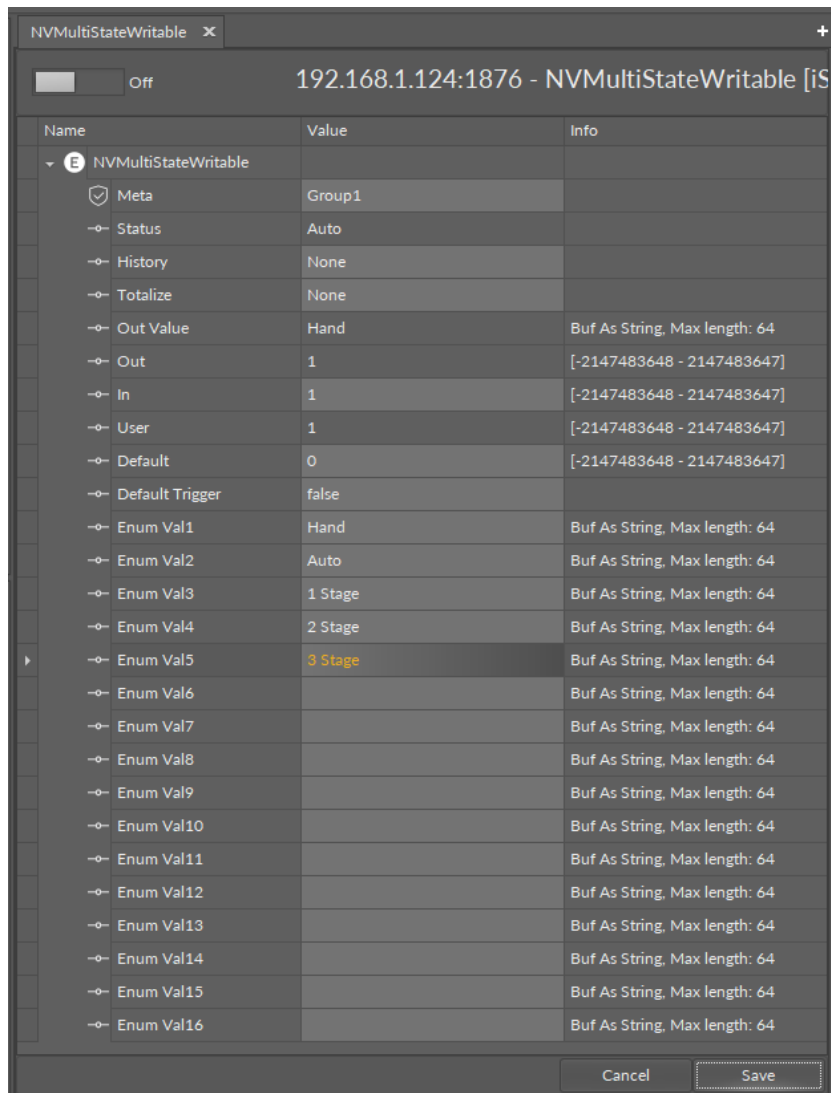


Figure 20 NV Multi State Writable view

The component has the following slots:

- Status – Point actual status Auto / Hand,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.
- Enum Val1-Val16 – String values which will be shown on the output depending on the Input value.

8 Historical

iSMA-B-AAC20 device has an embedded historical extension for writing trends in microSD card memory. By default, the device is not equipped with an SD card, Historical Service SD card should be installed when necessary. The historical extension can be added only in NVBooleanWritable, NVIntegerWritable and NVNumericWritable components. To add the extension you should choose one of the three available types in NV History slot components. Every historical has a predefined 2500 recordable samples. After reaching the limit, the oldest recordings are rewritten. While creating a historical, the device opens a database file on the SD card. Any such file can serve up to 100 extensions. Once the limit is reached, the new database file is opened.

8.1 History service

History service is the component, which is responsible for writing historical values in microSD memory. History Service component is available from iSMA_controlApi kit palette and the component must be placed under the Service folder.

After you add any historical extension on NV component, a historical database will be created under the history service component in order to store NV points of historical components.

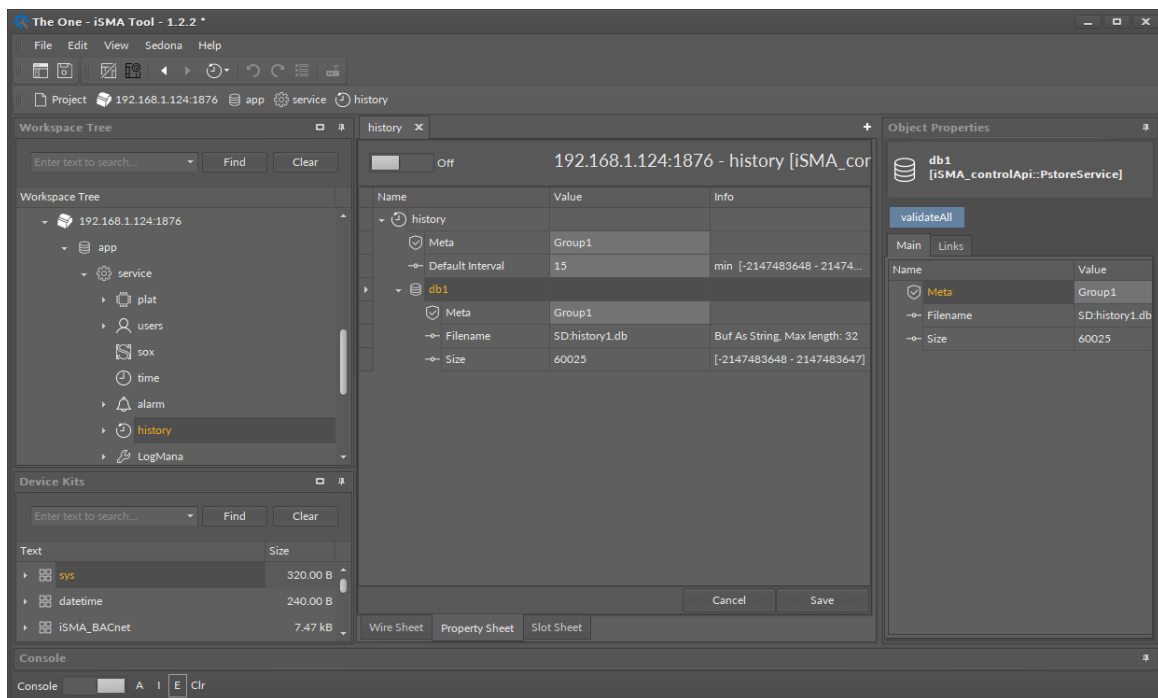


Figure 19 History Service property sheet view

8.2 History extensions – Interval

Interval history extension is used to collect data at specified time intervals. When you choose interval option in NV component, a history interval will be created under the history and link to NV point Out slot will be created. For configuration and settings, please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.

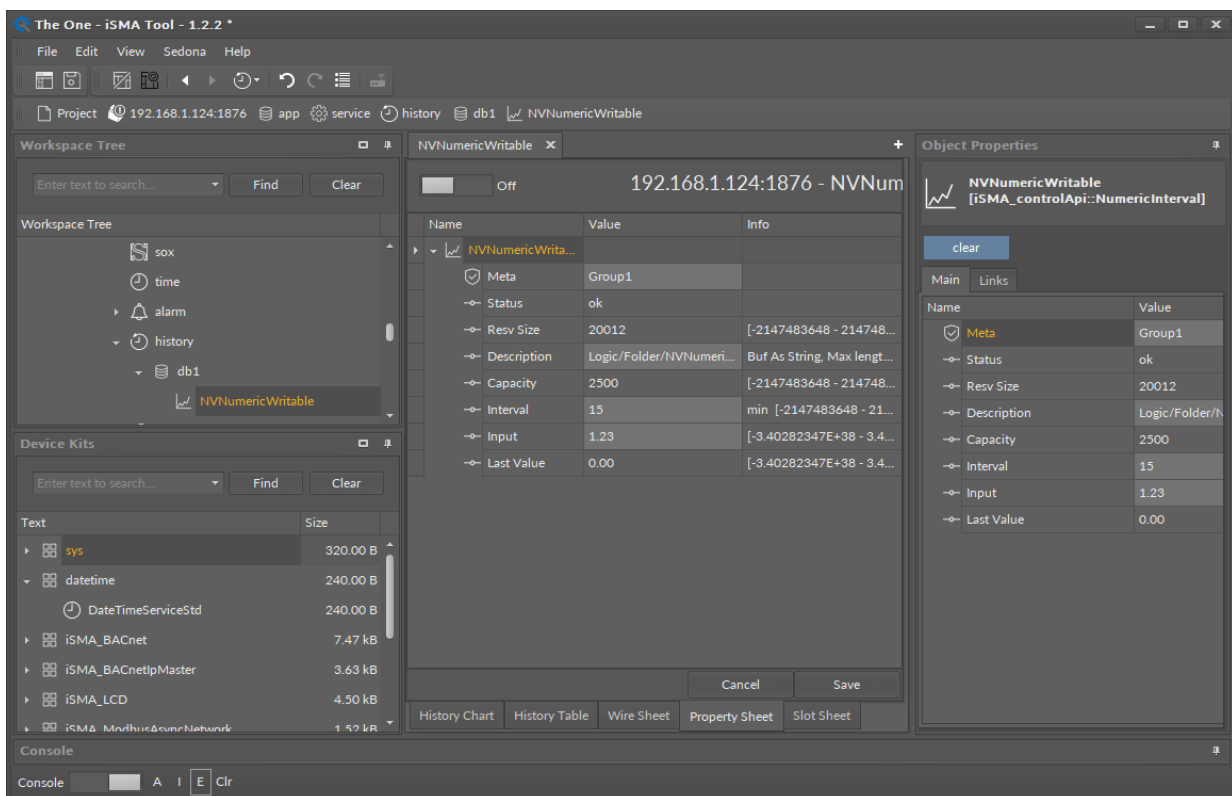


Figure 20 History Interval property sheet view

The component has the following slots:

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Interval – Saving time interval,
- Input – Current value,
- Last Value – Last saved value.

8.3 History extensions – COV

COV history extension is used to collect data whenever required changes has been made. When you chose COV option in NV component, a history COV component will be created under the history service and a link with NV point Out slot will be made. For configuration settings please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.

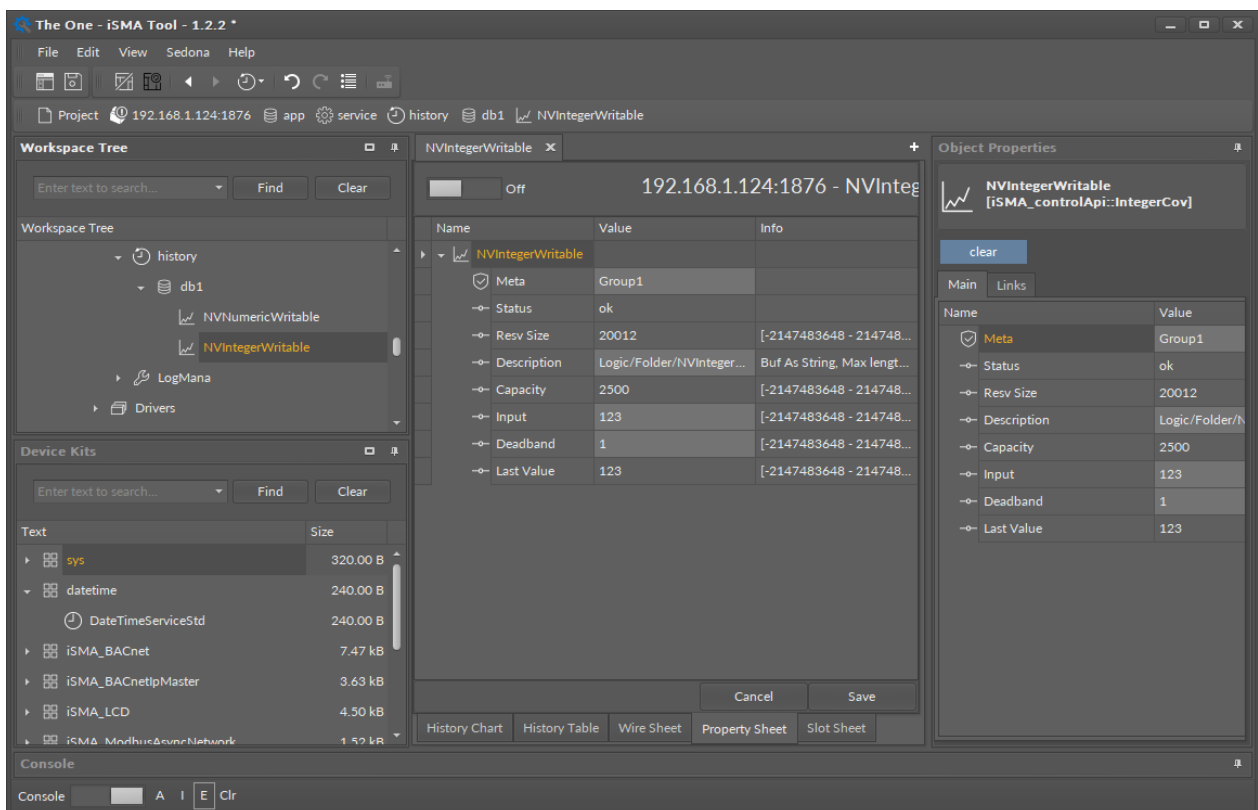


Figure 21 History COV property sheet view

The component has the following slots:

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Deadband – Saving hysteresis,
- Input – Current value,
- Last Value – Last saved value.

8.4 History extensions – COV_Interval

COV_Interval history extension is used to collect data whenever required changes has been made or at specified time intervals (whatever happens first). When you chose COV_Interval option in NV component, a history COVInterval component is created under the history service and a link with NV point Out slot will be made. For configuration settings please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.

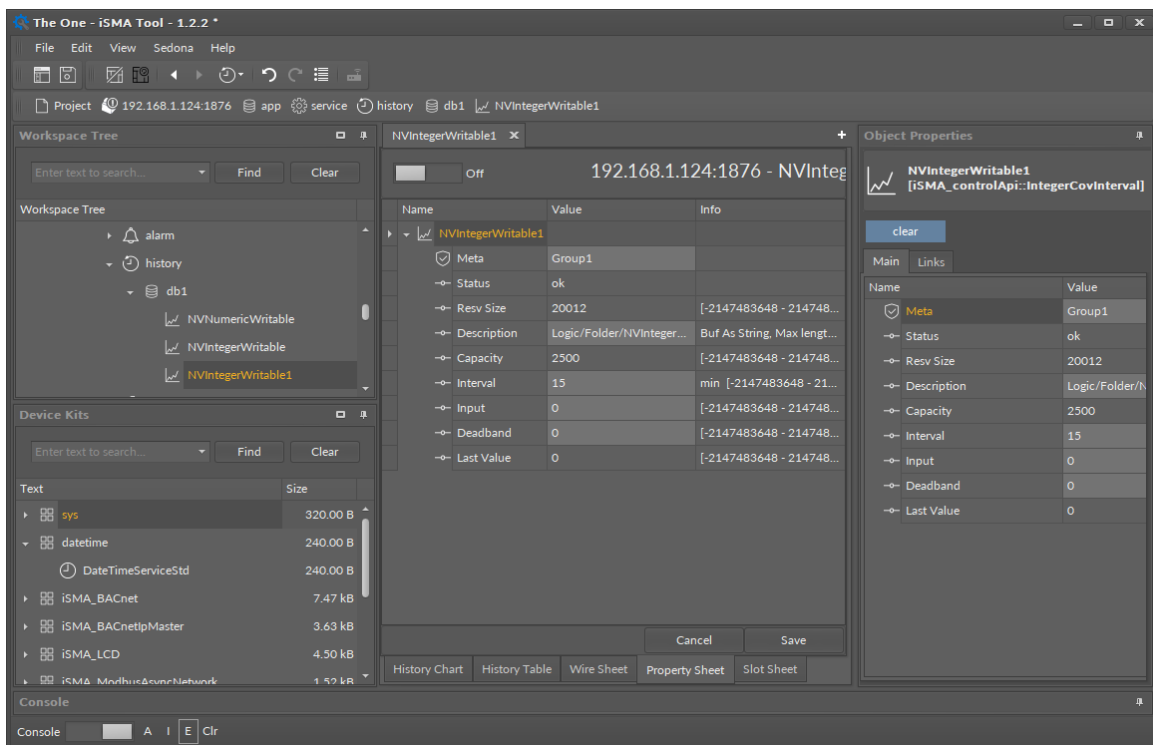


Figure 22 History Interval COV property sheet view

The component has the following slots:

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Interval – Saving time interval,
- Deadband – Saving hysteresis,
- Input – Current value,
- Last Value – Last saved value.

9 Alarms

The device has a service that allows recording events / alarms. Alarms can be generated off the NV-type components only. Alarms can be recorded only on the SD card. By default, the device is not equipped with an SD card, to use Alarm Service, SD card should be installed. When you create an extension of the alarm, the database file is created on the SD card. The device stores up to 2500 alarm events. If you exceed this number, the new event will overwrite the eldest records.

To add an extension to the component alarm you must select an appropriate option in the Alarm slot (Change Of State for Boolean components, and Out Of Range for Numeric components). After changing and saving the settings, an alarm extension is created under the component, where you are able to make all the settings associated with the process of generating an alarm event.

9.1 Alarm service

Alarm service is the component, which is responsible for servicing alarms events. The Alarm Service component is available from iSMA_controlApi kit palette and the component must be placed under Service folder.

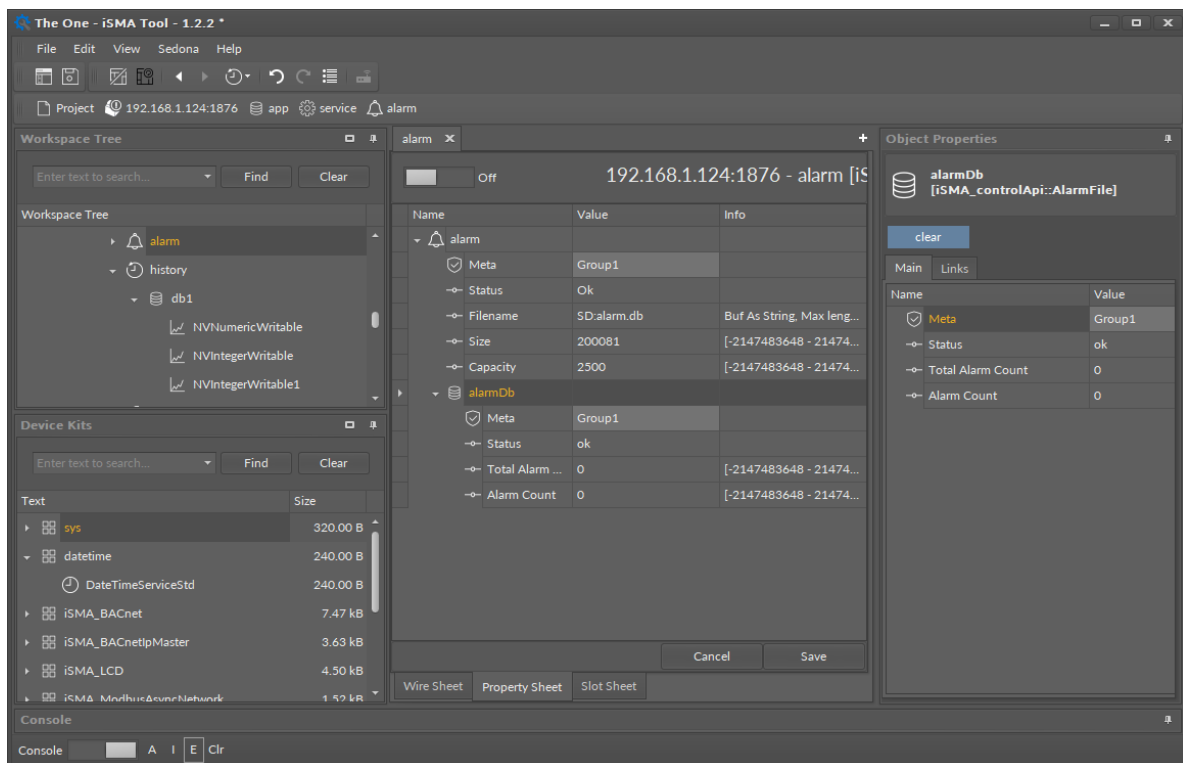


Figure 23 Alarm Service property sheet view

9.2 Alarm Boolean points

The Boolean alarm extension can be added only in NVBooleanWritable components. To add the extension, choose “Change Of State” in Alarm slot and an alarm extension will be created automatically under NV component.

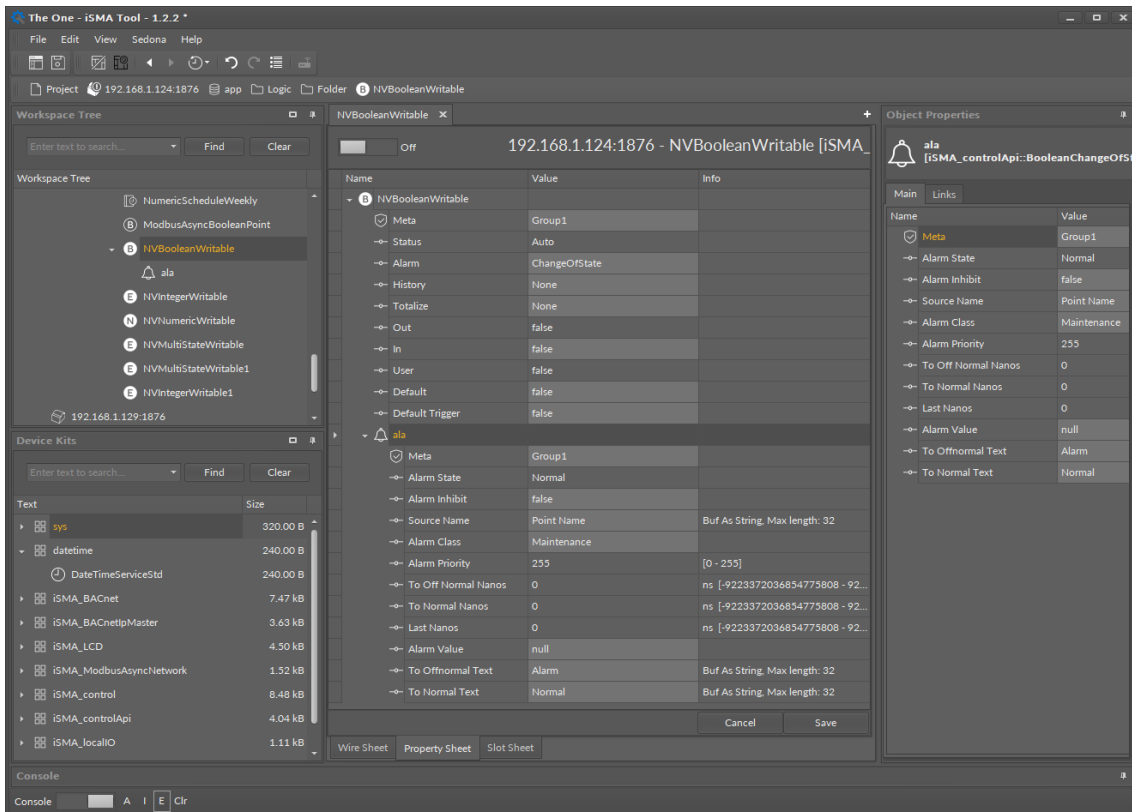


Figure 24 Alarm boolean extension property sheet view

Alarm extension slots:

- Alarm Inhibit – Enable / Disable alarm function,
- Source Name – Alarm name in alarms database,
- Alarm Class – Alarm class and priority definition, available options:
- Live Safety – priority 15,
- Critical – priority 75,
- Maintenance – 155,
- Alarm Priority – read only alarm priority number defined by alarm class slot (default value 255),
- To Off Normal Nanos – Time point of an alarm condition,
- To Normal Nanos - Time point of a normal condition,
- Alarm state – Actual alarm state,
- Alarm Value – Alarm generation value,
- To Offnormal Text – Description in alarm state,
- To Normal Text - Description in normal state.

9.3 Alarm Numeric points

The Numeric alarm extension can be added only in NVNumericWritable components. To add the extension, choose “Out Of Range” in Alarm slot, and the alarm extension will be created automatically.

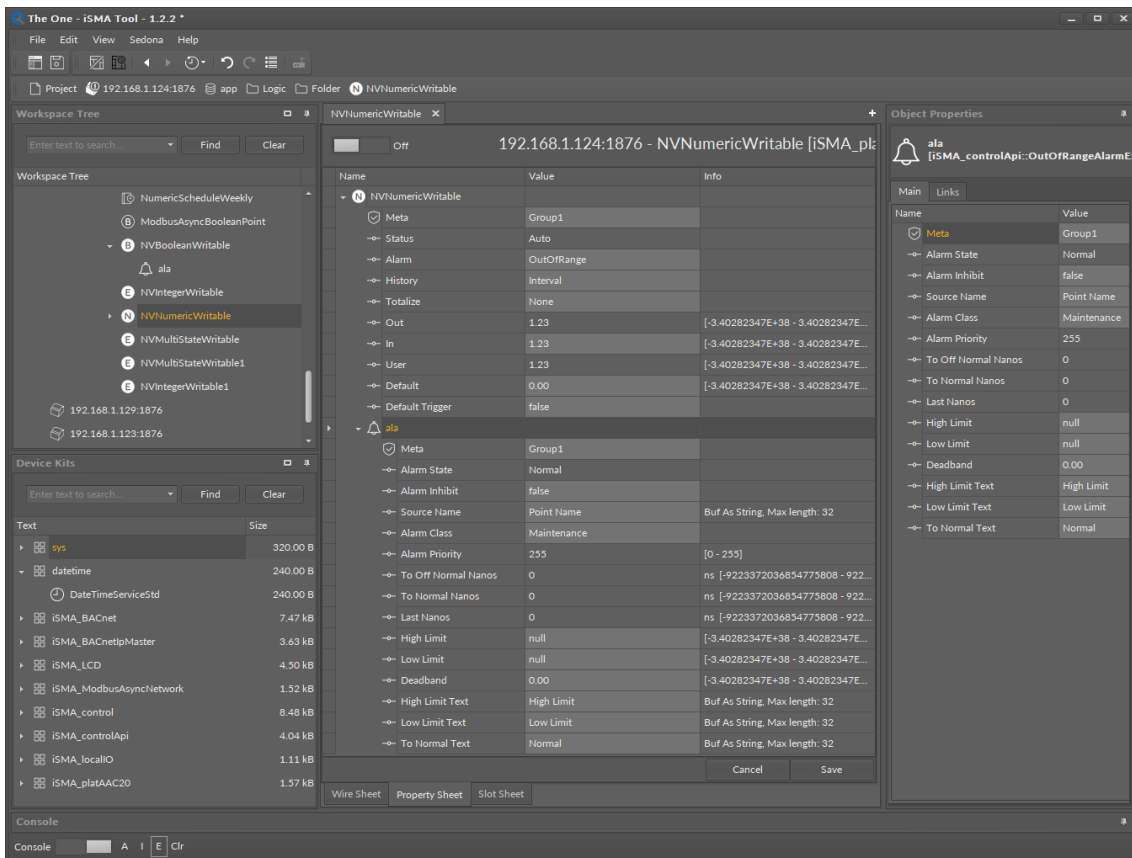


Figure 25 Alarm numeric extension property sheet view

The component has the following slots:

- Alarm Inhibit – Enable / Disable alarm function,
- Source Name – Alarm name in alarms database,
- Alarm Class – Alarm class and priority definition, available options:
- Live Safety – priority 15,
- Critical – priority 75,
- Maintenance – 155,
- Alarm Priority – read only alarm priority number defined by alarm class slot (default value 255),
- To Off Normal Nanos – Time point of an alarm condition,
- To Normal Nanos - Time point of a normal condition,
- Alarm state – Actual alarm state (Normal / Alarm),
- High Limit – High limit alarm generation value,

- Low Limit – low limit alarm generation value,
- Deadband – High and low limit dead band value
- High Limit Text – Description in alarm high state,
- Low Limit Text – Description in alarm low state,
- To Normal Text - Description in normal state.

10 Scheduler

Schedules in AAC20 controller are being created using schedule components, as found in iSMA_controlApi kit (kit is install in default application). There are available two types of Schedulers Boolean Weekly for Boolean values and Numeric Weekly for numeric values. In both types, there are available max 8 events for 7 week days. In component we define start time and value. If value is null (for Boolean) or nan (for Numeric) the event is discarding. For non-defined event scheduler will take value from Def Val slot.

The screenshot displays the 'NumericScheduleWeekly' property sheet in The One - iSMA Tool. The central table lists various properties and their values:

Name	Value	Info
Meta	Group1	
Out	null	[-3.40282347E+38 - 3.40282347E...
Next Event Value	null	[-3.40282347E+38 - 3.40282347E...
Next Event Time	628102680000000000	[-9223372036854775808 - 92233...
Time	627497940000000000	[-9223372036854775808 - 92233...
Sunday Schedule1	0	min [0 - 65535]
Sunday Val1	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule2	0	min [0 - 65535]
Sunday Val2	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule3	0	min [0 - 65535]
Sunday Val3	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule4	0	min [0 - 65535]
Sunday Val4	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule5	0	min [0 - 65535]
Sunday Val5	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule6	0	min [0 - 65535]
Sunday Val6	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule7	0	min [0 - 65535]
Sunday Val7	null	[-3.40282347E+38 - 3.40282347E...
Sunday Schedule8	0	min [0 - 65535]
Sunday Val8	null	[-3.40282347E+38 - 3.40282347E...
Monday Schedule1	0	min [0 - 65535]
Monday Val1	null	[-3.40282347E+38 - 3.40282347E...
Monday Schedule2	0	min [0 - 65535]
Monday Val2	null	[-3.40282347E+38 - 3.40282347E...
Monday Schedule3	0	min [0 - 65535]

The right-hand panel shows the 'Object Properties' for 'NumericScheduleWeekly' with the following values:

Name	Value
Current Time	19/11/2019 16:59:00
Current Output	null
Default Value	null
Next Event Value	null
Next Event Time	26/11/2019 16:58:00

Figure 26 Schedule property sheet view

11 SOX protocol

Sox is the standard protocol used to communicate with Sedona devices. To serve Sox protocol Sox is also used by iSMA Tool to program controller. Note that Sox is service type and it is execute after application components. So if there is a little deference between scan period and Scan Time (see App component) the services don't have time execute and the programming interface can slow down. Sox is designed to be run over UDP/IP protocol (default port 1876).

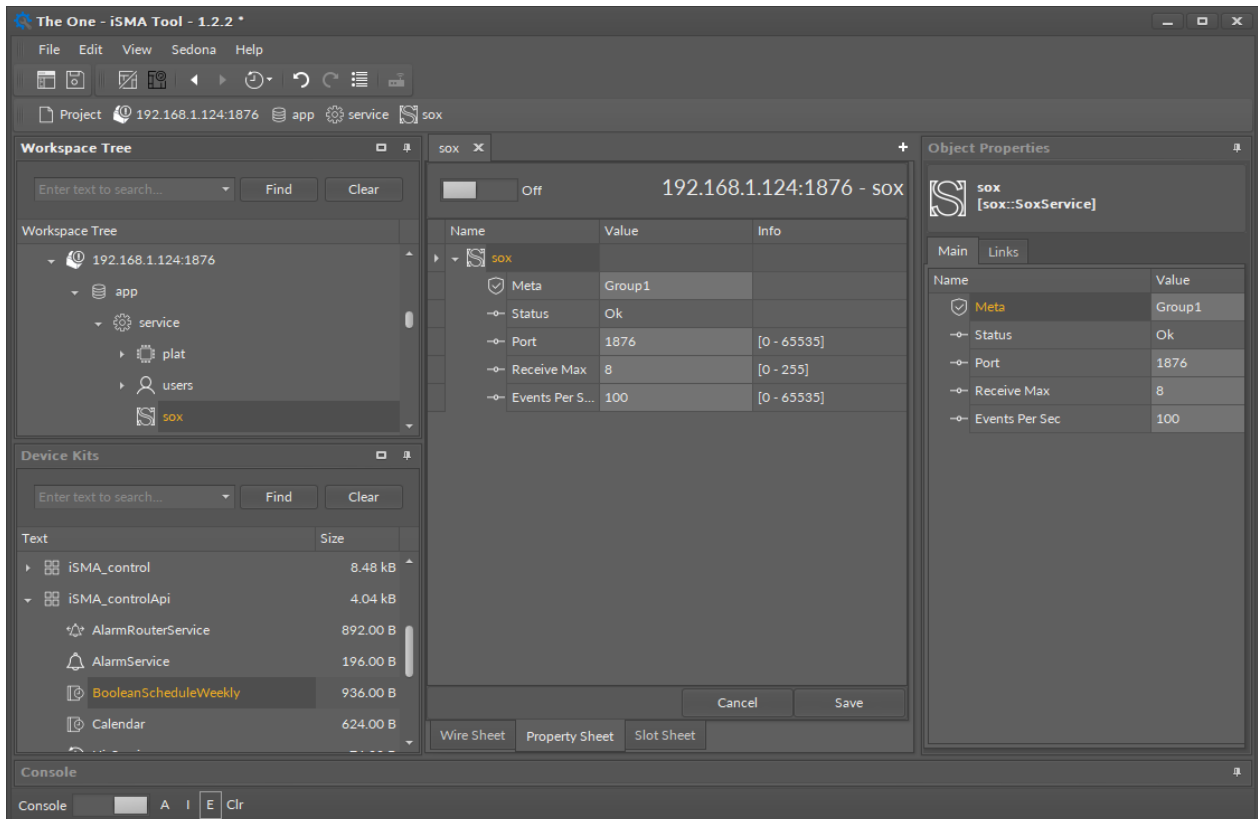


Figure 27 SOX property sheet view

The component has the following slots:

- Status – Sox driver actual status,
- Port – Sox UDP port (default value 1876),
- Receive Max – Max of number of messages in receiving window,
- Events Per Sec – Max of number of async events (telegrams) sent per second

12 Local IO

The iSMA-B-AAC20 device has 22 built-in physical inputs and outputs:

8x Universal Inputs

All Universal Inputs have 16-bit resolution which support the following types of inputs:

- Temperature Input support the following types of sensors: 10K3A1, 10K4A1, Carel 10K, 20K6A1, 2.2K3A1, 3K3A1, 30K6A1, SIE1, TAC1, SAT1, Pt1000, Ni1000, accuracy $\pm 0,1^{\circ}\text{C}$
- Voltage Input 0-10VDC o input resistance 100k Ω o accuracy $\pm 1\text{mV}$
- Resistive Input 0-1000k Ω • Dry contact input.

4x Digital Inputs

- Dry contact inputs
- High-speed pulse counter up to 100Hz

4/6x Analogue Outputs

All Analog Outputs are equipped with 12-bit ADC provides 10mV resolution and accuracy less than $\pm 0,5\%$. They support the following output types:

- Output 0-10V DC
- Digital Output with maximum load up to 20mA

4x Digital Outputs

- Relay Output max. 3A, 250V AC / 30 VDC

12.1 Adding Local IO components

Local IO's are supported by the components of the iSMA_LocalIO Kit. The components of this kit should be placed in a special folder localIO.

Note: It is recommended to pass the signal inputs and outputs through NV components.

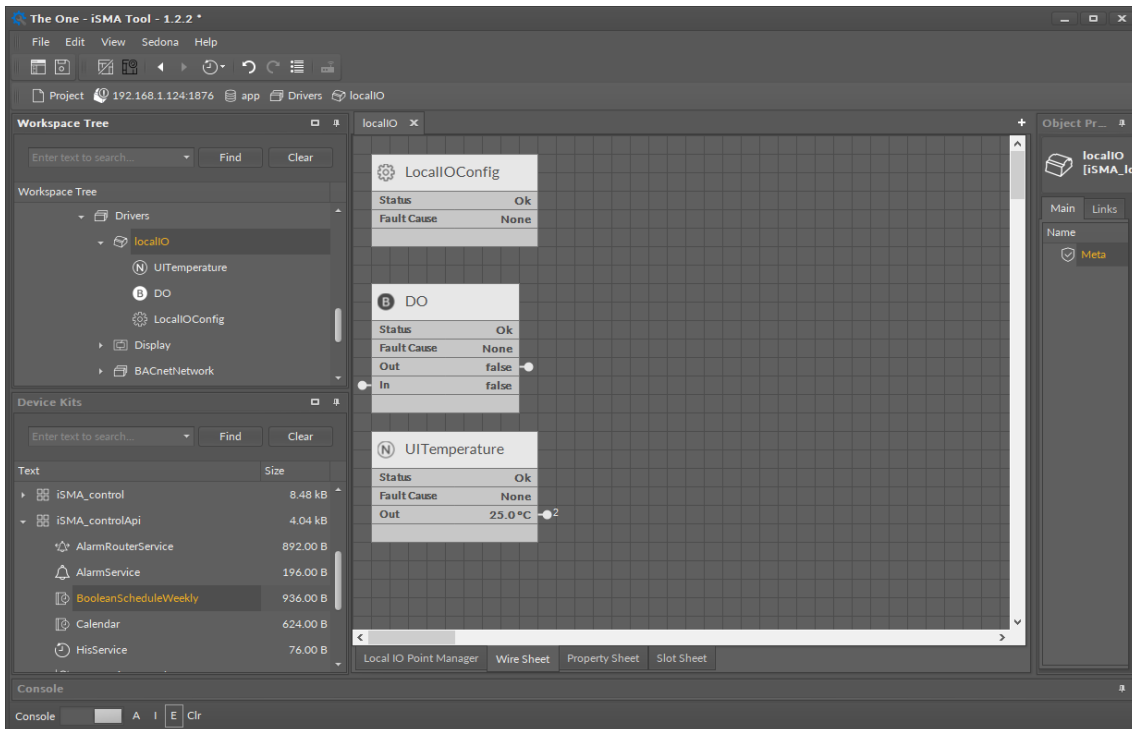


Figure 28 Local IO

LocalIO kit consists of the following components:

- **LocalIO** – Main component servicing the physical I/O
- **LocalIOFolder** – Folder for grouping IO components,
- **DI** - Digital Input handling component
- **UIResistance** - Component for servicing Universal Input in the resistance read mode
- **UITemperature** - Component for servicing Universal Input in the temperature read mode (in order to get reliable readings, you must select the appropriate sensor type in the component LocalIOConfig)
- **UIVoltage** - Component for servicing Universal Input in the voltage read mode (in order to improve the reading, you must activate the component LocalIOConfig for voltage reading)
- **UIDigital** - Component for servicing Universal Input in the Dry Contact read mode
- **DICounter** - Component reading high-speed counter of Digital Inputs.
- **DO** - Component handling Digital Output
- **AOVoltage** - Component for servicing Universal Output in the voltage read mode

- **AODigital** - Component for servicing Universal Output in the digital mode (false - 0V, true - 10V)
- **LocalIOConfig** - Component for configuration of physical input/output parameters of the device.

12.2 Local IO Config

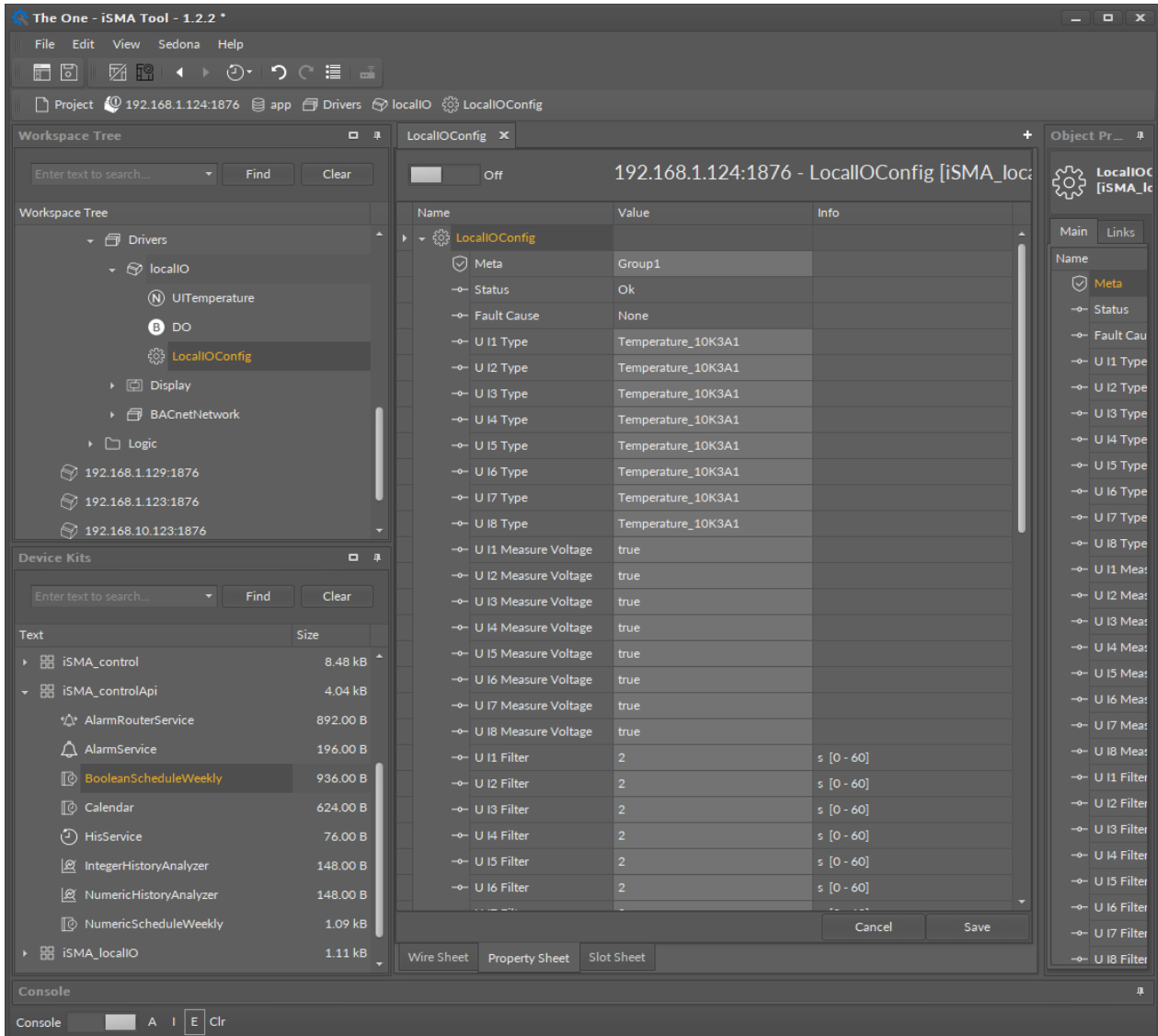


Figure 29 Local IO config property sheet view

LocalIOConfig is the component designed to configure the physical input/output of the device. It consists of the following slots:

- Ulx Type - The type of temperature sensor connected to Universal Input; table stored in the device allows you to convert the value of sensor resistance into temperature. Available options: Disable_Resistance, Temp_10K3A1, Temp_10K4A1, Temp_10K, Temp_20K6A1, Temp_2_2K3A1, Temp_3K3A1, Temp_30K6A1, Temp_SEI1, Temp_TAC1, Temp_SAT1, Temp_Pt1000, Temp_Ni1000.
- Ulx Measure Voltage - enables / disables the measurement of the voltage at the device's universal input,
- Ulx Filter - The time constant of the low pass filter (to eliminate signal noise)
- Ulx Resolution – Measurement resolution of a Universal Input. **WARNING!**16-bit value increases the time to read a single entry. It should be used only when you connect sensors PT1000 and NI1000.
- AOx Type - Analog Output Mode: Voltage 0-10V or PWM
- Default State Of Analog Output – The Analogue Output default value after your reboot the controller expressed in mV.
- Default Digital State Of Analog Output - The default value of the Analogue Output, working in digital mode, when you reboot the device.
- Default State Of Digital Output - Digital Output default when you reboot the device.
- **Note:** Adding the LocalIOConfig component to the application shall disable the possibility of I / O configuration from the Web Site and Modbus array.
- Any settings of the LocalIOConfig component are carefully stored in the component and can be transferred to other devices (quick setup of multiple devices).

13 One Wire

One Wire is the special protocol for device which is made by Dallas company. Temperature sensors use DS18D20 chip. OneWire bus use 3 wire cable (+5 VCC, Data, GND) max 100m long (it is recommended to use shorter distance) and up to 32 sensors. All iSMA-B-AAC20 devices are equipment with OneWire port which use two Analog Outputs AO5 and AO6.

WARNING! Before connecting OneWire device to controller please add OneWire network first. It will block the possibility of increasing voltage above 5 V and protect OneWire sensor against damage.

13.1 Adding OneWire network

To use OneWire protocol iSMA_OneWire kit needs be installed on the controller using Kit Manager (see Kit Manager chapter). Then OneWireNetwork component needs to be placed under Drivers folder.

Note: if LocalIO are being used in AO5 slots LocalIOConfig component, Type and AO6 Type value will change to OneWire and it can't be changed manually. If OneWire are being used, AO5 and AO6 Analog Outputs can't be utilized.

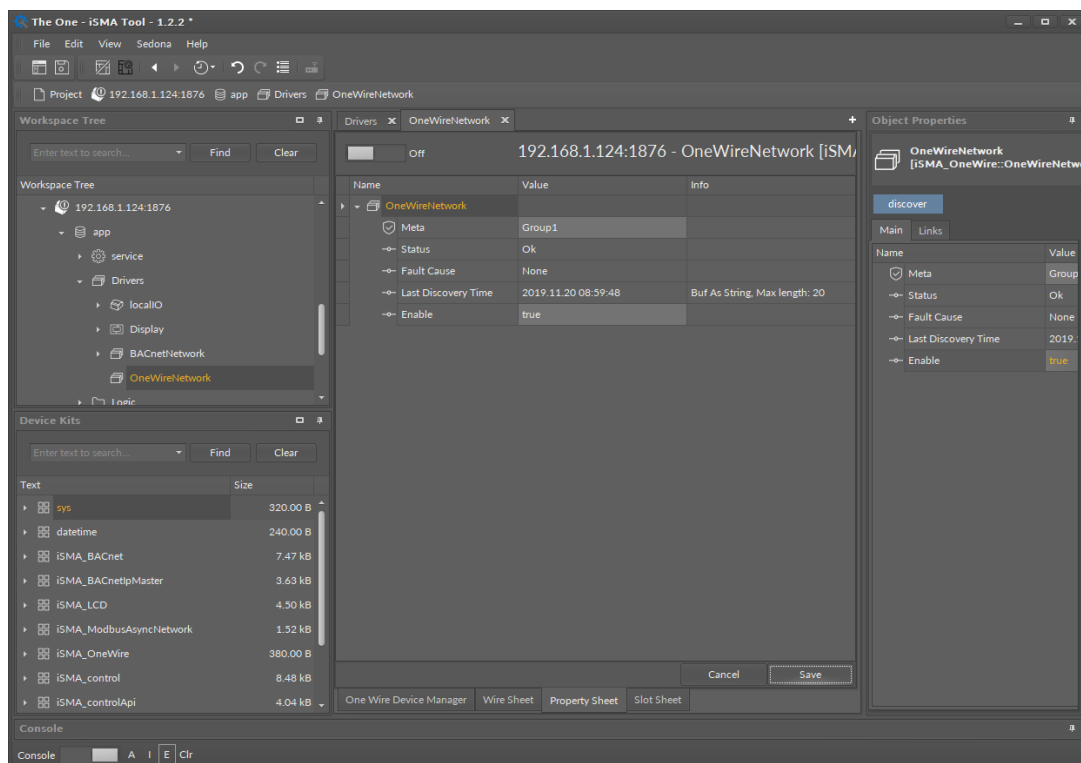


Figure 30 One Wire property sheet view

13.2 Discovering OneWire sensors

OneWire network has Discover action (under right mouse button). This action will automatically searched and detect all sensors connected to the bus. It will also create Discover folder and place all found sensors there. All sensors have unique address assigned by manufacturer (slot Address). There is also Description slot to describe sensor function / place using max 20 characters. All sensors components can be moved and grouped in OneWireFolders (folders must be also placed under OneWireNetwork component).

13.3 Adding sensors manually

To add sensor manually, OneWireThermometer component needs to be moved from iSMA_OneWire kit palette and be placed under OneWireNetwork. Then sensor address needs to be entered and application must be saved. If the connection and address are correct, status slot should have ok value and Out slot should have sensor temperature given in Celsius degrees.