

# SBIridium 1.2

User Manual

# Table of contents

|   |    |
|---|----|
| How to work in SBlridium .....                      | 3  |
| Compatibility with iRidium Studio .....             | 3  |
| Supported Smart-Bus G4 devices.....                 | 3  |
| Licensing.....                                      | 4  |
| Replacement of license keys .....                   | 4  |
| Working in SBlridium in a demo-mode .....           | 4  |
| Installation and starting of SBlridium .....        | 5  |
| Installation of the database .....                  | 5  |
| Update driver database.....                         | 6  |
| Adding Smart-Bus G4 devices to the project .....    | 7  |
| Starting drivers .....                              | 9  |
| Linking graphic elements to the driver 4LED .....   | 11 |
| Linking graphic elements to the driver DMX 48.....  | 12 |
| Linking graphic elements to the driver Z-Audio..... | 13 |

# How to work in SBIdrium

## Compatibility with iRidium Studio

You can work with the SBIdrium database in the editor [iRidium Studio](#) version from 1.1.4.

## Supported Smart-Bus G4 devices

The database SBIdrium 1.2 allows you to work with the following Smart-Bus G4 devices:

|                            |   |
|----------------------------|---|
| Relay                      | <a href="#">SB-2R-UN</a><br><a href="#">SB-3R-UN</a><br><a href="#">SB-RLY4c20A-DN</a><br><a href="#">SB-RLY6c16A-DN</a><br><a href="#">SB-RLY8c16A-DN</a><br><a href="#">SB-RLY12c10A-DN</a> |
| Dimmers                    | <a href="#">SB-DIM2c6A-DN</a><br><a href="#">SB-DIM4c3A-DN</a><br><a href="#">SB-DIM6c2A-DN</a><br><a href="#">SB-6B0-10V-DN</a><br><a href="#">SB-DIM8C1A-DN</a>                             |
| Mixed modules              | <a href="#">SB-MIX24-DN</a><br><a href="#">SB-ZMIX23-DN</a>   |
| Discrete input and sensors | <a href="#">SB-4Z-UN</a><br><a href="#">SB-9in1T-CL</a><br><a href="#">SB-6in1T-CL</a><br><a href="#">SB-5in1-CL</a>  |
| Climate                    | <a href="#">SB-4T-UN</a><br><a href="#">SB-HVAC2-DN</a>   |
| Media                      | <a href="#">SB-IRM-UN</a><br><a href="#">SB-ZAudio2-DN</a>  |
| DMX and 4LED               | <a href="#">SB-DMX48-DN</a><br><a href="#">SB-4LED-DCV</a>  |
| Control panel              | <a href="#">SB-DDP-EU</a><br>SB-NDP   |

## Licensing

Licensing of library in the project is made in an account of Iridiummobile.net in three steps:

1. You pay the license and you receive a license key;
2. You enter this key in your account of Iridiummobile.net in the section "Pro" → "My licenses";
3. You link the obtained license to your object.

Get more detailed information about work with objects and licenses on [Iridiummobile website](#).

## Change control panels

The license doesn't become attached to specific certain panels. Install i3Pro app on the new control panel and start your project as you did it before.

You can operate the list of the activated panels in an account of Iridiummobile.net in the section "Users and Panels" of your object.

## Working in SBIdridium in a demo-mode

If you have the status "Silver" on the Iridiummobile.net, then you are given an opportunity to use full functionality of SBIdridium without licensing in a demo mode:

- 20 minutes of unlimited operation of the control panel. After this i3Pro it is required to restart;
- the Developer's panel mode for 2 of your control panels. This mode lifts a limit on time of continuous work;
- the 60-day trial period of work for testing of your projects on the customer's devices.

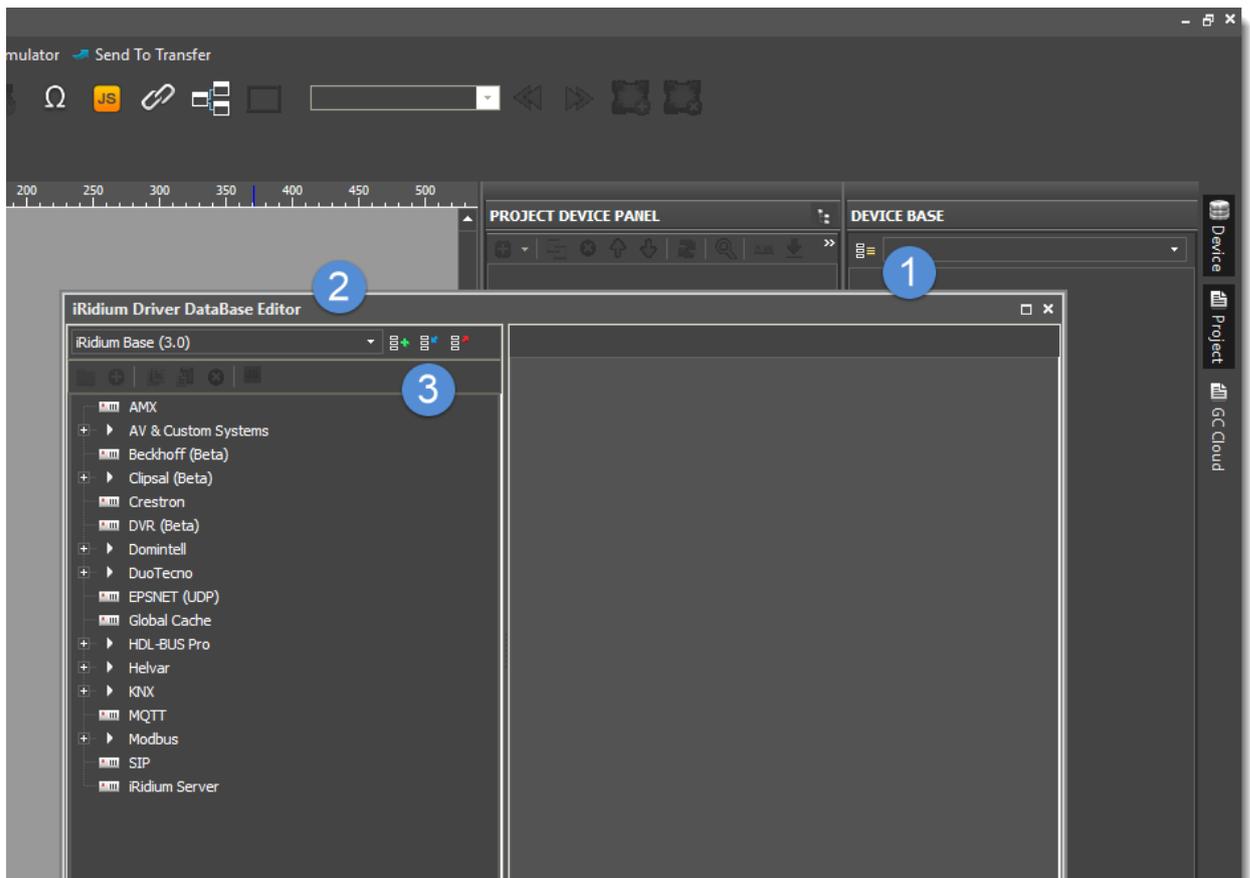
To get Silver status on Iridiummobile.net write to us to [Support@SBIdridium.com](mailto:Support@SBIdridium.com).

# Installation and starting of SBIRidium

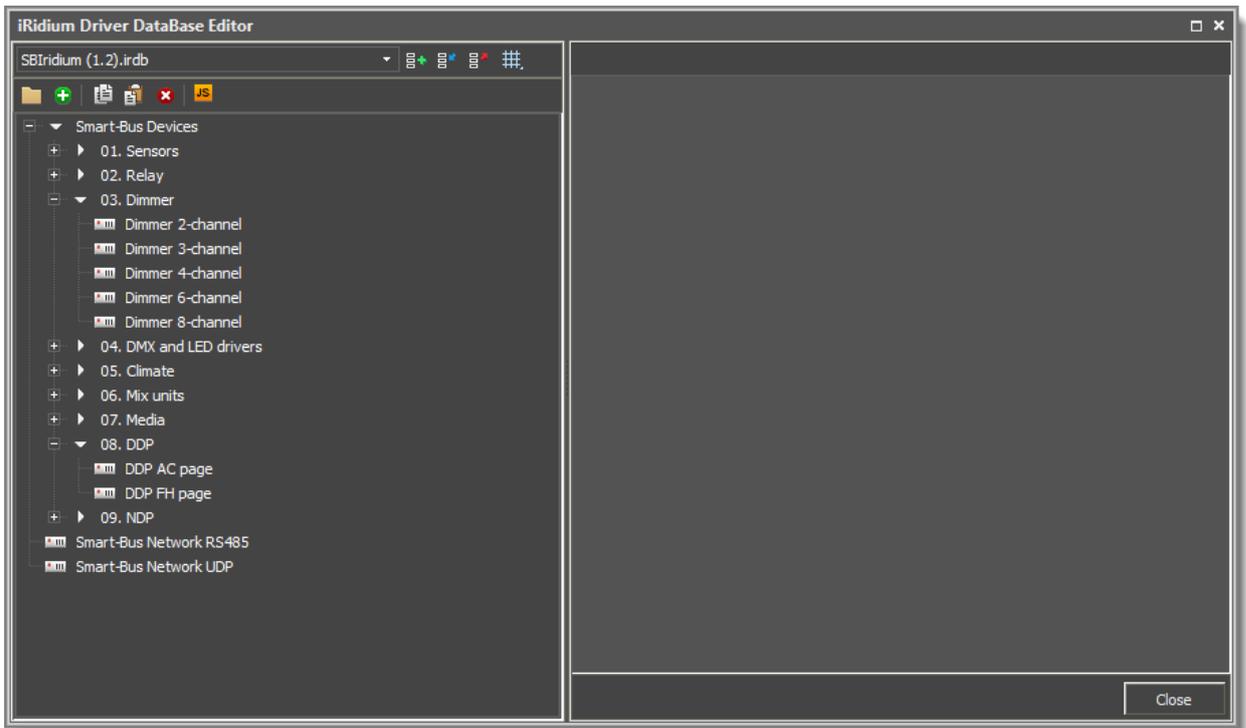
## Installation of the database

[Download](#) the latest version of the database SBIRidium.irdb.

Start iRidium Studio, turn on the panel Device Base, (1) open the window iRidium Driver DataBase Editor (2) and press Attach (3) to attach SBIRidium to your studio.



After you attach the database, the new line SBIRidium (1.2) will appear in the dropdown list.



The driver database has been installed.

## Update driver database

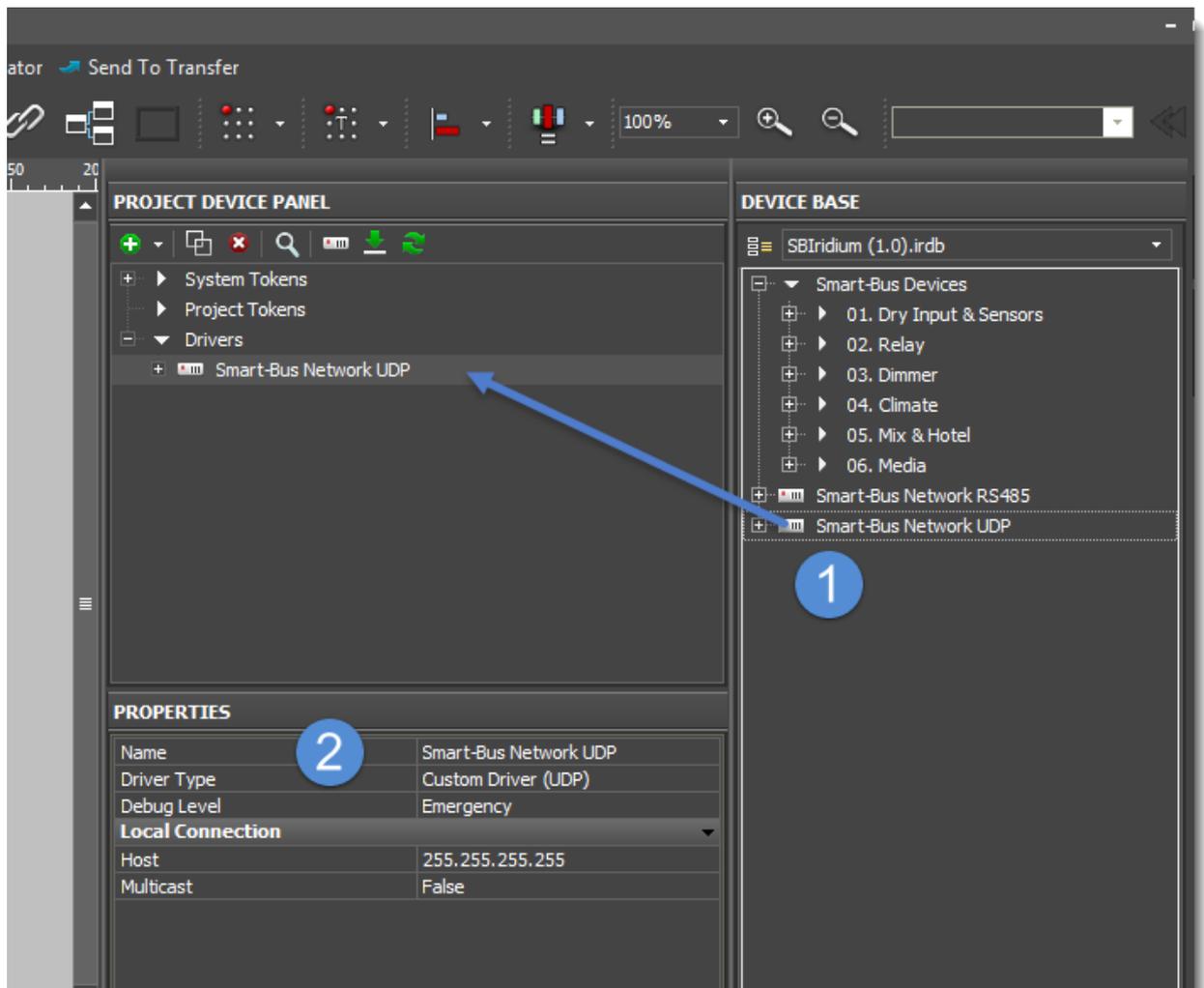
If you have already installed SBIridium 1.1, detach it in the window «iRidium Driver Database Editor» and install SBIridium 1.2.

## Adding Smart-Bus G4 devices to the project

To work with Smart-Bus G4 devices, add drivers from the database SBIRidium to your project iRidium.

### Smart-Bus Network UDP, Smart-Bus Network RS485

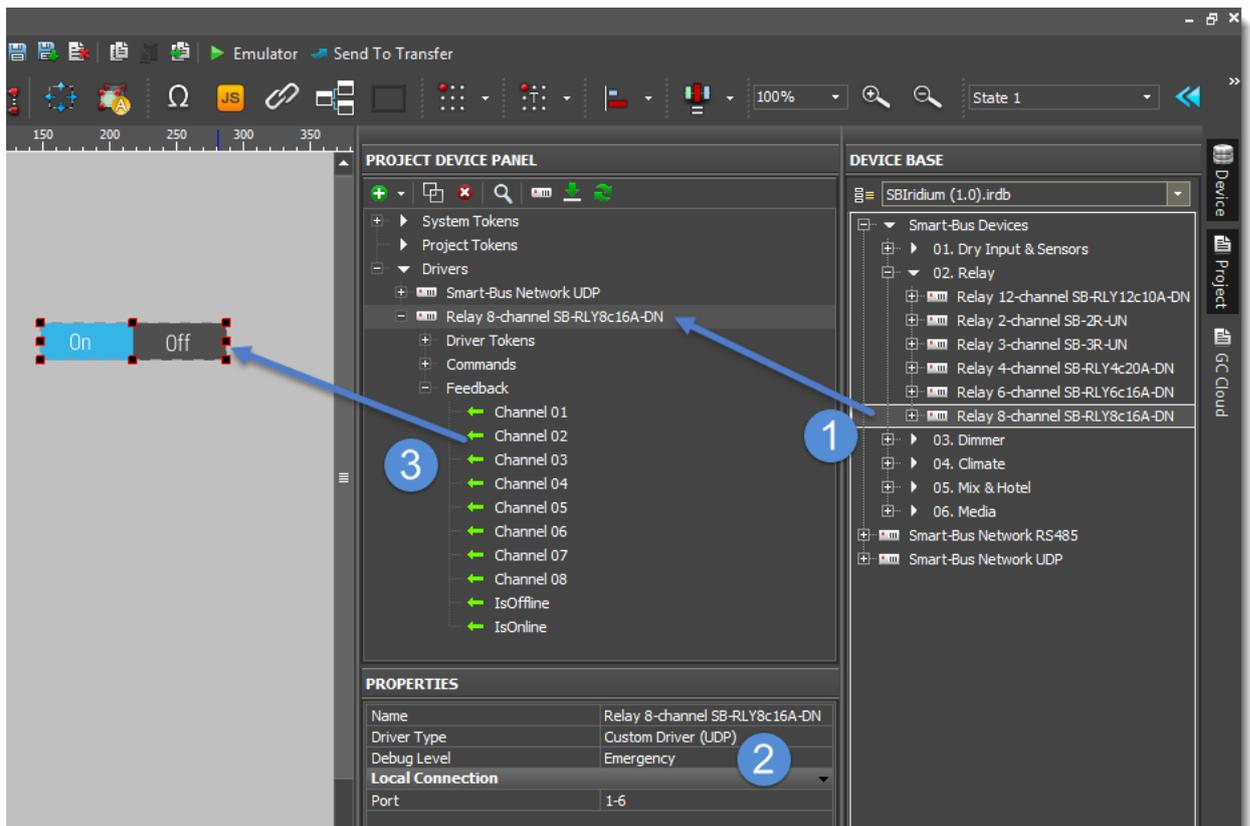
Depending on the type of the network on the object, Ethernet or RS485, add the respective driver to the project (1) and indicate required network settings (2).



*There should be only one network driver in the project.*

### Smart-Bus Devices

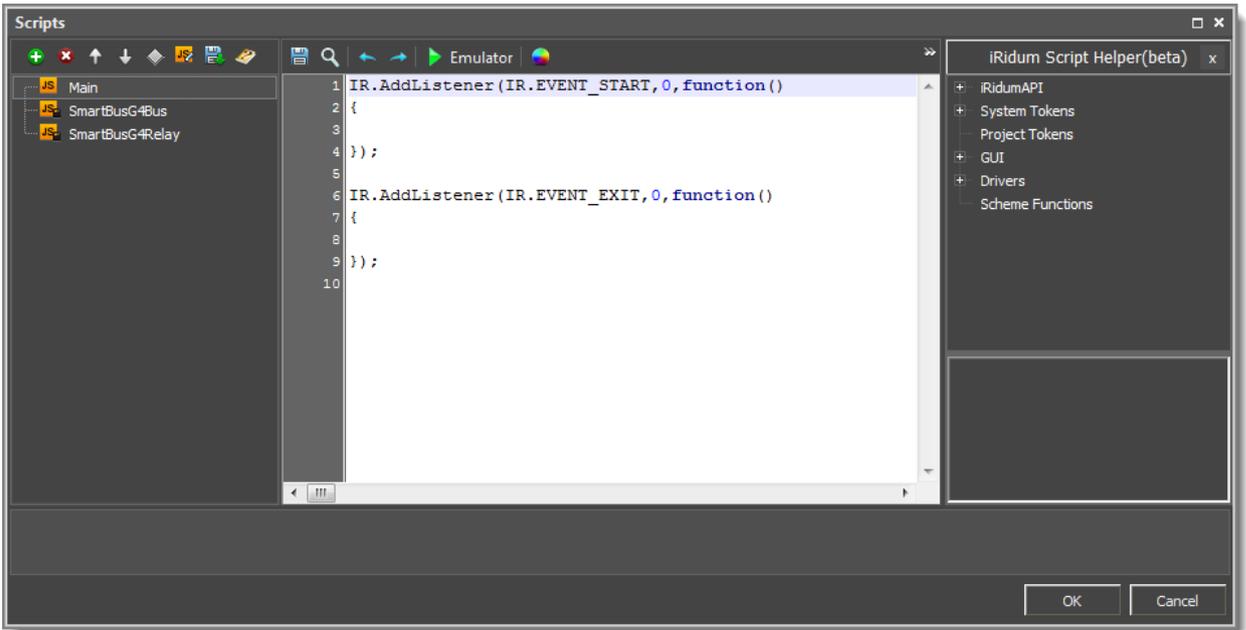
Drag (1) the required device, e.g. the 8-channel relay module, to your project, indicate the module address in the Smart-Bus network in the field Port (2) and do required feedback and commands binding to graphic elements of your project.



*It is important that there are no devices with identical Smart-Bus addresses in the project; the field Port shall be different for each device. Otherwise, the studio will ignore these devices when starting the project.*

### What happens when a device is added to the project and a message about duplication

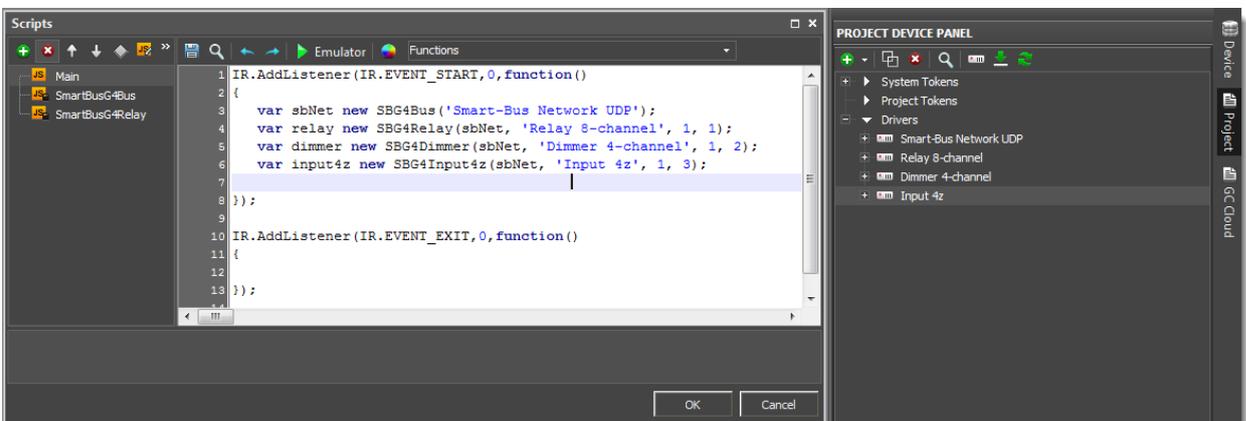
When you drag the device from the SBIridium database the js-file with description of the class of this device is copied to your project. For example, when you add the Smart-Bus Network, the js-file SmartBusG4Bus.js with class description SBG4Bus is added to the project; for a relay it will be SmartBusG4Relay.js with the class SBG4Relay.



## Starting drivers

To start the driver, it is necessary to create instances of all devices added to the project in the initializing script of the project and start the project.

For example, you have added to the project Smart-Bus Network UDP, a module relay, a dimmer and the module 4z. To start all these drivers, you need to add the definition of the instance for each of the drivers when starting the application.



The list of constructors for SBirdium devices is given below:

| Device              | Constructor  |
|---------------------|--|
| Smart-Bus Network x | SBG4Bus(deviceName)                                |
| Input 4z            | SBG4Input4z(sbNet, deviceName, subnetID, deviceID) |

| Device              | Constuctor  |
|---------------------|---|
| Sensor x in 1       | SBG4Sensor(sbNet, deviceName, subnetID, deviceID, [units])                          |
| Relay x-channel     | SBG4Relay(sbNet, deviceName, subnetID, deviceID)                                    |
| Dimmer x-channel    | SBG4Dimmer(sbNet, deviceName, subnetID, deviceID, [rememberLastControl])            |
| HVAC2               | SBG4Hvac2(sbNet, deviceName, subnetID, deviceID, [units])                           |
| Input 4t            | SBG4Input4t(sbNet, deviceName, subnetID, deviceID, [units])                         |
| Mix24               | SBG4Mix24(sbNet, deviceName, subnetID, deviceID, [rememberLastControl])             |
| Zone-Beast 23       | SBG4ZBeast23(sbNet, deviceName, subnetID, deviceID, [units], [rememberLastControl]) |
| IR Emitter          | SBG4IREmitter(sbNet, deviceName, subnetID, deviceID)                                |
| Z-audio             | SBG4ZAudio(sbNet, deviceName, subnetID, deviceID)                                   |
| 4LED                | SBG4Led(sbNet, deviceName, subnetID, deviceID)                                      |
| DMX 48              | SBG4DMX(sbNet, deviceName, subnetID, deviceID, [rememberLastControl])               |
| DDP AC page         | SBG4DdpAC(sbNet, deviceName, subnetID, deviceID, [units])                           |
| DDP heat floor page | SBG4DdpFH(sbNet, deviceName, subnetID, deviceID, [units])                           |
| NDP                 | SBG4ndpAC(sbNet, deviceName, subnetID, deviceID, [units], [pageNo])                 |

where:

deviceName - the name of the driver in the project;

key1..key10 - license keys;

sbNet - the instance of the network "Smart-Bus Network x";

subnetID - address of the subnetwork of the device;

deviceID - address of the device;

units - code of temperature units: 0- Celsius, 1- Fahrenheit;

rememberLastControl - for dimmers to remember the last settings: 1=yes; 0=no.

## Linking graphic elements to the driver 4LED

To work correctly 4LED driver, you need attach it to the graphic controls. For this driver provides methods:

| Method                               | Description   |
|--------------------------------------|---|
| <code>attachColorPicker(item)</code> | Binding to color picker   |
| <code>attachColorPreset(item)</code> | Binding to an element with the preset colors.<br>When you click on a preset, driver will give out on the RGB-tape color of this preset.<br>For one driver can be linked as much as the number of presets. |
| <code>attachColorBox(item)</code>    | Binding to an element that must be filled in the current color RGB-tapes  |

Linking graphic elements should be made in the project script once after declaring of the driver.



```
1 IR.AddListener(IR.EVENT_START,0,function()
2 {
3   var sbNet = new SBG4Bus('Smart-Bus');
4   var led = new SBG4Led(sbNet, 'LED Driver', 1, 14);
5
6   led.attachColorPicker(IR.GetPage("pageMain").GetItem("colorPicker"));
7   led.attachColorBox(IR.GetPage("pageMain").GetItem("colorCurrent"));
8   led.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset1"));
9   led.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset2"));
10  led.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset3"));
11  led.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset4"));
12  led.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset5"));
13
14
15 });
```

To get a link to a graphic element, use system method iRidium Studio [GetItem\(\)](#).

Use assistant iRidium Script Helper in the right pane, the script editor to quickly get a link to the item.

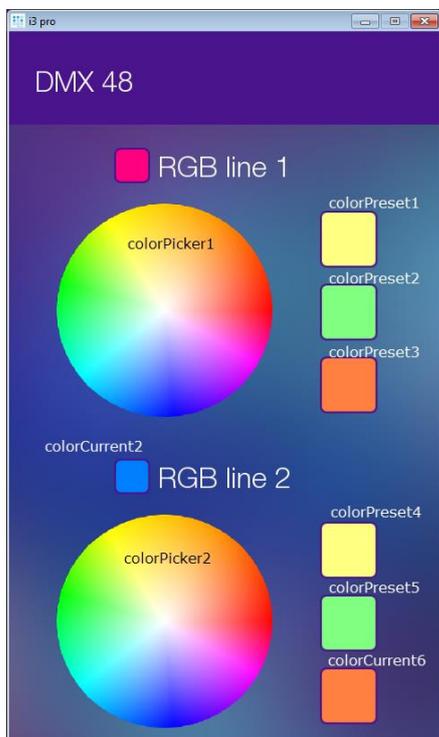
## Linking graphic elements to the driver DMX 48

To work correctly DMX48 driver, you need attach it to the graphic controls and specify which channels are controlled RGB-tape. For this driver provides methods:

| Метод   | Описание  |
|---|---|
| <code>attachColorPicker(item, channelRed, channelGreen, channelBlue)</code> | Binding to color picker   |
| <code>attachColorPreset(item, channelRed, channelGreen, channelBlue)</code> | Binding to an element with the preset colors. When you click on a preset, driver will give out on the RGB-tape color of this preset. For one driver can be linked as much as the number of presets. |
| <code>attachColorBox(item, channelRed, channelGreen, channelBlue)</code>    | Binding to an element that must be filled in the current color RGB-tapes  |

For each method of binding must be specified 48 DMX numbers channels: channelRed, channelGreen, channelBlue, which will be controlled by this graphic element.

Linking graphic elements should be made in the project script once after declaring of the driver.



```

1 IR.AddListener(IR.EVENT_START,0,function()
2 {
3   var sbNet = new SBG4Bus('Smart-Bus');
4   var dmx = new SBG4DMX(sbNet, 'DMX48', 1, 48, 1);
5
6   // Attach channels 1,2,3
7   dmx.attachColorPicker(IR.GetPage("pageMain").GetItem("colorPicker1"), 1,2,3);
8   dmx.attachColorBox(IR.GetPage("pageMain").GetItem("colorCurrent1"), 1,2,3);
9   dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset1"), 1,2,3);
10  dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset2"), 1,2,3);
11  dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset3"), 1,2,3);
12
13  // Attach channels 4,5,6
14  dmx.attachColorPicker(IR.GetPage("pageMain").GetItem("colorPicker2"), 4,5,6);
15  dmx.attachColorBox(IR.GetPage("pageMain").GetItem("colorCurrent2"), 4,5,6);
16  dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset4"), 4,5,6);
17  dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset5"), 4,5,6);
18  dmx.attachColorPreset(IR.GetPage("pageMain").GetItem("colorPreset6"), 4,5,6);
19
20
21 });

```

To get a link to a graphic element, use system method iRidium Studio [GetItem\(\)](#).

Use assistant iRidium Script Helper in the right pane, the script editor to quickly get a link to the item.

## Linking graphic elements to the driver Z-Audio

To display the list of albums and songs you need to attach to the driver graphic elements of type List, placed on your project visualization page. For this driver provides methods:

| <b>Метод</b>                                     | <b>Описание</b>  |
|--|--|
| <code>attachSDLlists(albumList, songList)</code> | Binding to elements of type List to display the list of albums and songs from the SD-card    |
| <code>attachFtpLists(albumList, songList)</code> | Binding to elements of type List to display the list of albums and songs from the FTP-server |

Linking graphic elements should be made in the project script once after declaring of the driver.

To get a link to a graphic element, use system method iRidium Studio [GetItem\(\)](#).

Use assistant iRidium Script Helper in the right pane, the script editor to quickly get a link to the item.