

PROKNX

by BleuComm'Azur



**KNX interface software
for BC4 Project Editor v1.4
(Bidirectional)**

Version 1.1

Bleu Comm' Azur EURL
180, imp. du Baou
F-06560 Valbonne
www.proknx.com

2011-09-22

CONTENT

- GENERAL INFORMATION 3**
- VERSION CONTROL 3
- INTRODUCTION 3
- SYSTEM REQUIREMENTS..... 3

- CONFIGURATION OF THE PROKNX IP GATEWAY..... 4**

- ETS PROGRAMMING WITH THE PROKNX GATEWAY 4
- CONFIGURATION OF THE PROKNX GATEWAY PARAMETER..... 4
- HOW TO CONFIGURE PROKNX OBJECTS IN ETS SOFTWARE 6

- USING BITWISE CONTROLS PROJECT EDITOR..... 7**

- STEP1: IMPORT OF THE SCRIPT DEVICE “PROKNX” 7
- STEP2: SELECT COMMUNICATION AND TWO WAY PARAMETER..... 7
- STEP3: DEFINE THE LOOPING MACRO 9
- STEP 4: IMPORT THE GUI GROUP..... 10
- STEP 5: CONFIGURING BUTTONS 10
- ScriptDevice Functions 11
- GUI Script Functions 12
- STEP 6: CONFIGURING FEEDBACK..... 13
- Types of feedback 13

- EXAMPLES..... 14**

- ABOUT..... 25**



General Information

Version Control

This document explains the configuration of the bidirectional software PROKNX – for bitwise controls v1.1

Introduction

The software PROKNX- for bitwise controls comes along with:

- ProKNX 2way project (ProKNX_2way_project_v1.1.bwe)
- GUI group with templates for different buttons and GUI Script (ProKNX_2way_GUI_GROUP_v1.1.grl)
- Script device to import (ProKNX_2way_scriptDevice_v1.1.bscp)
- KNX Database for the gateway (ProKNX_ip_baos_770.vd2)
- this documentation (PROKNX-for-bitwise_v1.1.pdf)

It allows controlling lighting, shutter, scene and temperature control functions like:

- On / Off switching
- Dimming control
- Scene switching
- Scene programming
- Room Temperature Controller Mode selection
- Temperature and other 2Byte set point adjustments
- Shutter, sunblind and curtain control

The feedback information may be used to display as:

- State (0 – 1)
- 1 Byte value as integer (0 – 255)
- 1 Byte value as interger (0% - 100%)
- Graphical Shutter position
- 2 Bytes values (float) like temperature, luminosity, wind speed

Note: the BC4 unit only allows the feedback to the first connected client. All further clients may control (command) but will not show the correct feedback!

System requirements

You will need the following equipment to connect a bitwise client to the KNX bus:

- ProKNX-IPbwc gateway
- bitwise controls BC4 controller

You will need at least the following software release:

- Bitwise Controls Project Editor Version 1.4.46.0 or later
- BC4 Device Firmware Version 1.093

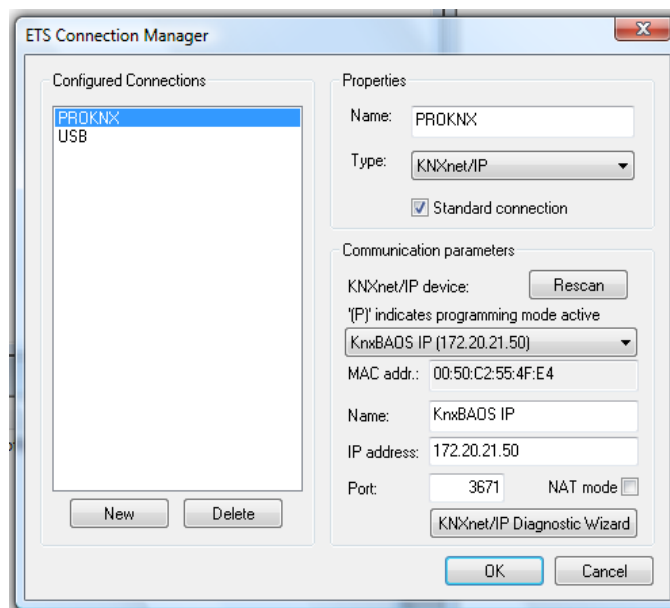
In case of questions how to get the newest firmware, programming software or PROKNX gateway, please see our website www.proknx.com or the website of <http://bitwisecontrols.com/supportforum/index.php> and view the download section.

Configuration of the PROKNX IP Gateway

ETS programming with the ProKNX gateway

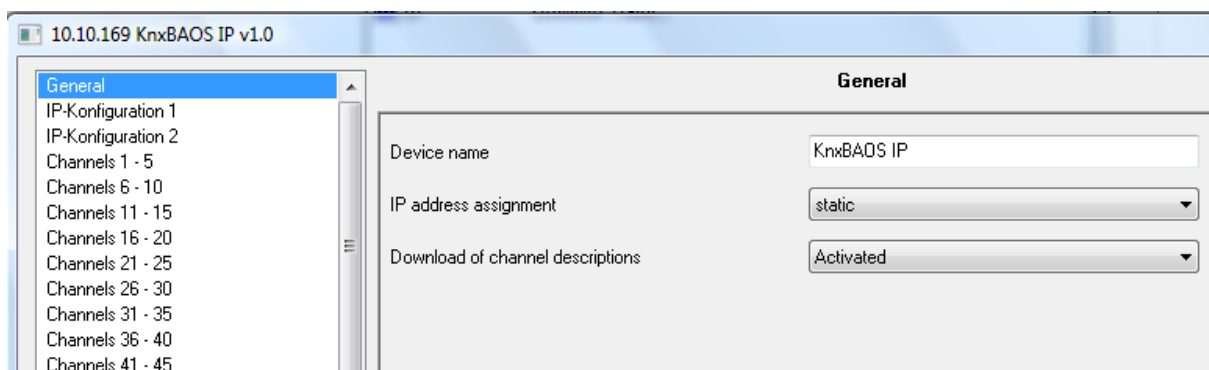
Beside the communication of the BC4 with the KNX-Bus, the gateway allows also the programming of the bus by means of your ETS software (tunneling protocol).

To define the communication of your ETS software with the bus through the gate, call extras->options --- thumbnail communication; Click on “configure interfaces...” and adjust the dialog as follows after clicking on “rescan”



Configuration of the ProKNX gateway Parameter

You'll find the ETS application for the gateway in the download section of our website. After importing the application you may configure the variables of the gateway in the “Edit parameters...” section of the context menu.

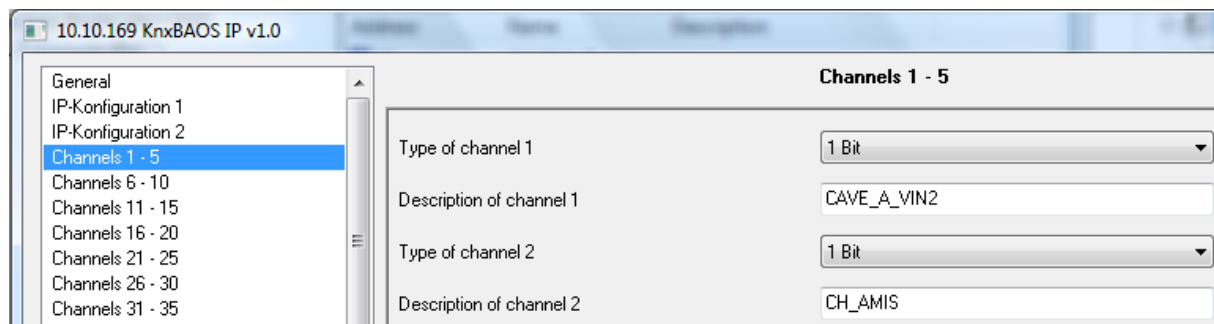


- Please use a static IP address to allow the BC4 controller to find the gate in the network
- Use IP-Konfiguration 1 and 2 to define the IP address
- Once you have configured the IP address, think that you have to rescan the gate as described above, if you are programming the bus with this gate.

The Gateway allows communicating through objects (channels) with the KNX bus. Each object data type has to be defined in the sub menus.

PROKNX – IPbwc is using only the following data types:

- 1 Bit channels (e.g. for switching circuits on/off)
- 1 Byte channels (e.g. for setting or reading dimming values, scenes, positions...)
- 2 Bytes channels (e.g. for temperatures)



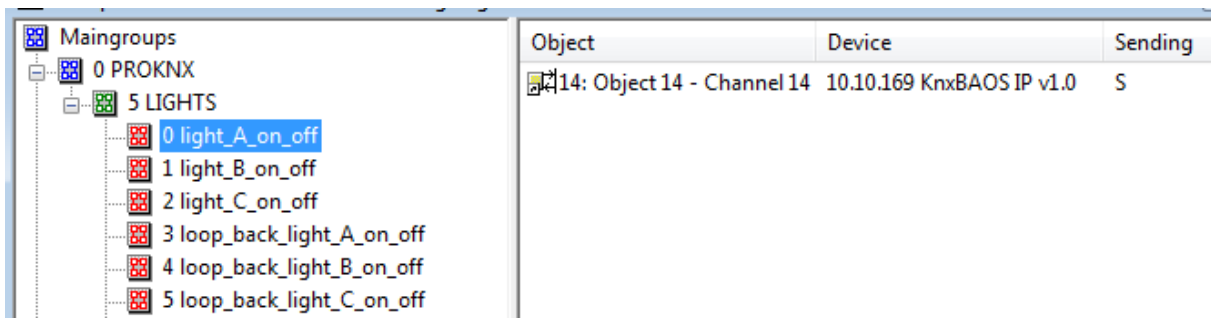
The channel description is not necessary for the functionality and may be used for your documentation.

How to configure PROKNX objects in ETS Software

To get the correct state of the KNX actor it is necessary to define „loop back“ group addresses. Here an example of group addresses in your ETS software for dimming actors.

Address	Name	Description
0	light_A_on_off	
1	light_B_on_off	
2	light_C_on_off	
3	loop_back_light_A_on_off	
4	loop_back_light_B_on_off	
5	loop_back_light_C_on_off	
6	dimm_val_light_A	
7	dimm_val_light_B	
8	dimm_val_light_C	
9	loop_back_dimm_val_light_A	
10	loop_back_dimm_val_light_A	
11	loop_back_dimm_val_light_A	

Please assign the same PROKNX IP object (channel) first to the “command” group address and then to the “loop back” group address. The object in the “command” group address has to have the Sending attribute!



The screenshot shows the ETS software interface. On the left, a tree view under 'Maingroups' shows a group '0 PROKNX' containing a sub-group '5 LIGHTS' with objects 0 through 5. Object 0 is highlighted. On the right, a table shows the configuration for object 14.

Object	Device	Sending
14: Object 14 - Channel 14	10.10.169 KnxBaos IP v1.0	S

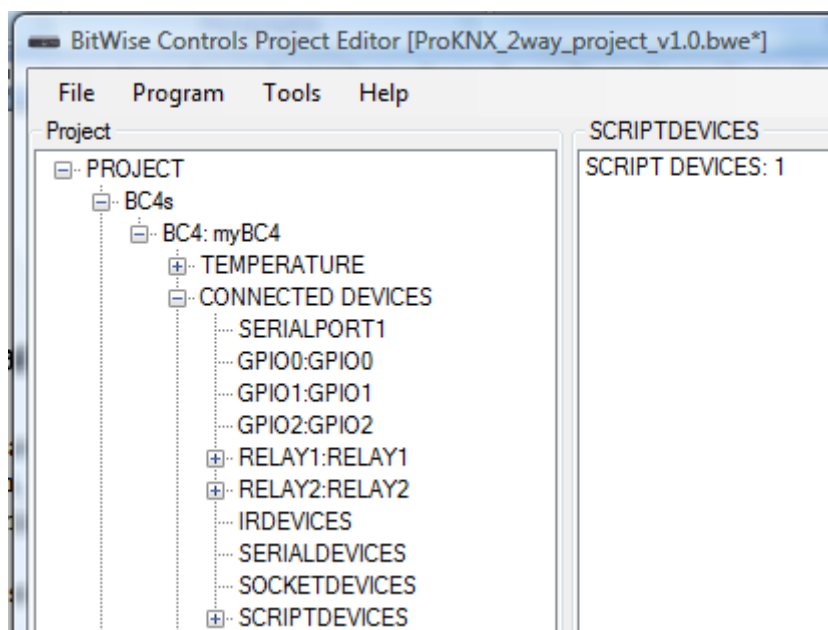
Using Bitwise Controls Project Editor

If you do not have an existing configuration, which should be completed with the KNX communication, please just open the ProKNX_2way_project_v1.0.bwe file with the Project Editor. After this you may only see step 2, step 5 and step 6 of this documentation.

The following complete procedure allows adding the ProKNX communication to an existing project:

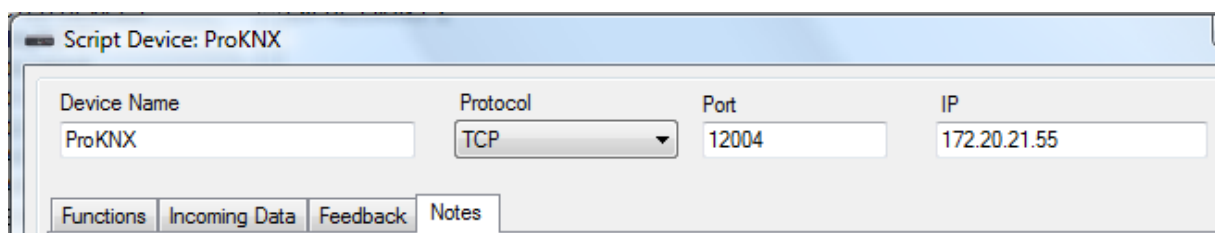
Step1: Import of the script device “ProKNX”

Open the Project Editor and open your project. Expand the “BC4s” line and highlight “SCRIPTDEVICES”. Then click right and import the “ProKNX_2way_scriptDevice_v1.0.bscp” file.

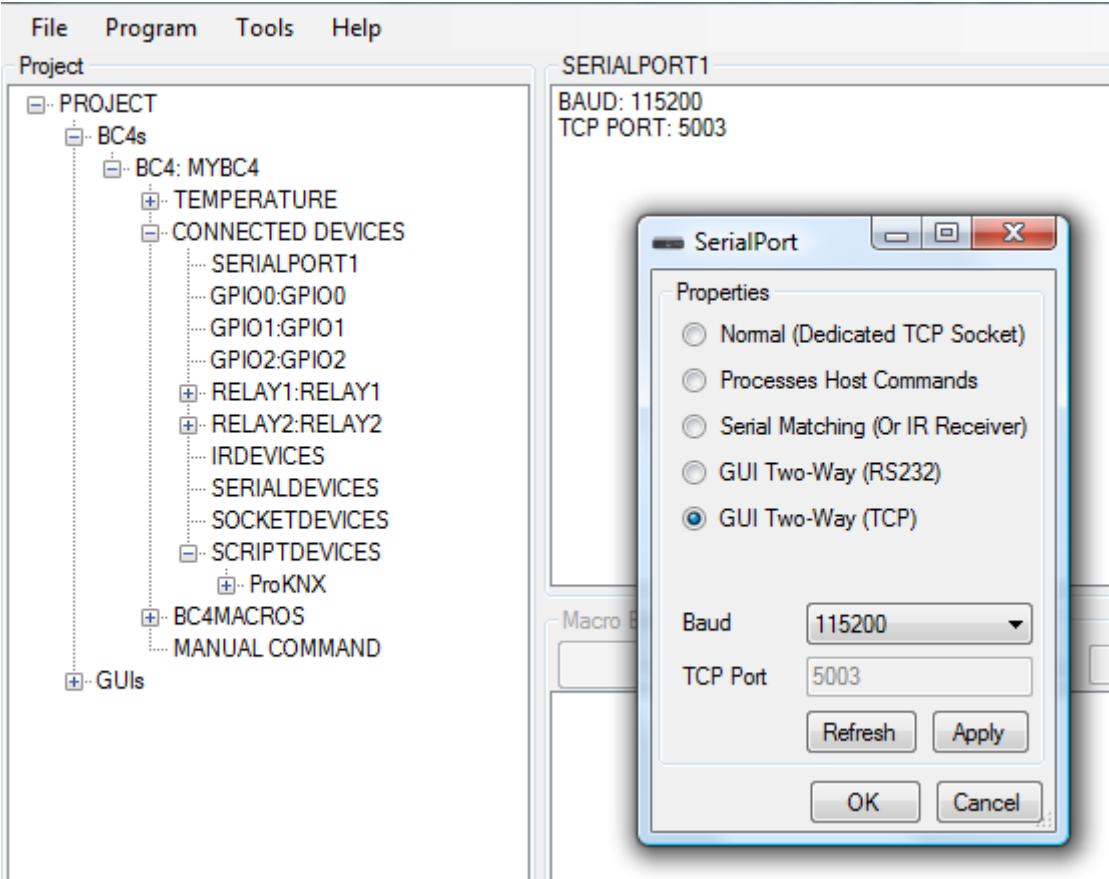


Step2: Select communication and two way parameter

Highlight the “ProKNX” Scriptdevice, click right and select “Properties”. Insert in the dialog box Protocol “TCP”, Port “12004” and insert as IP address the address you’ve selected for the gateway in ETS ([see here](#)).

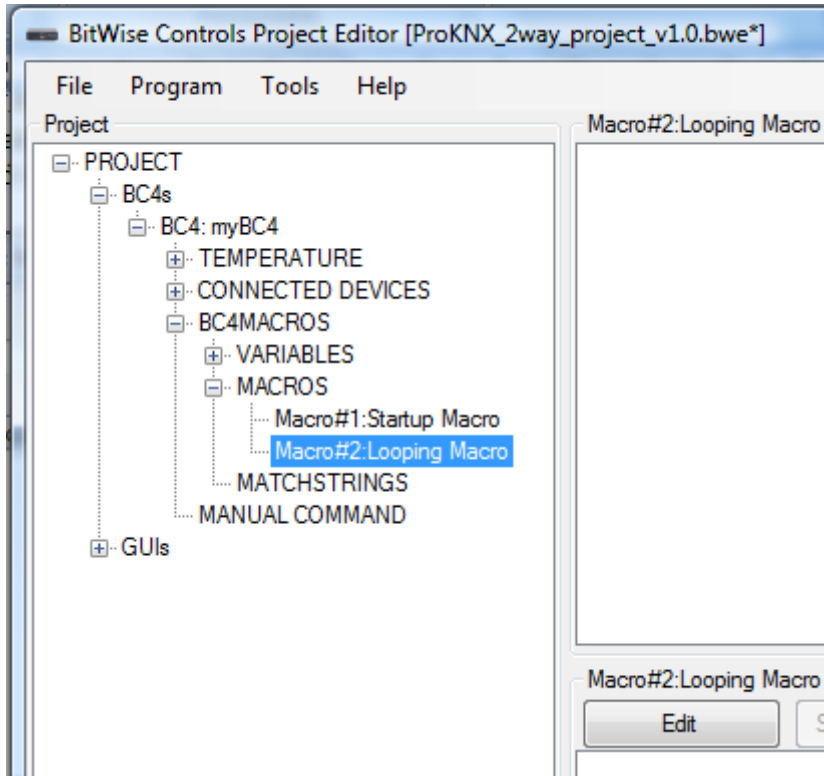


Highlight the “SERIALPORT1” line in “CONNECTED DEVICES”, click right and select “Properties”. In the dialog box choose “GUI Two-Way (TCP)” and apply settings.

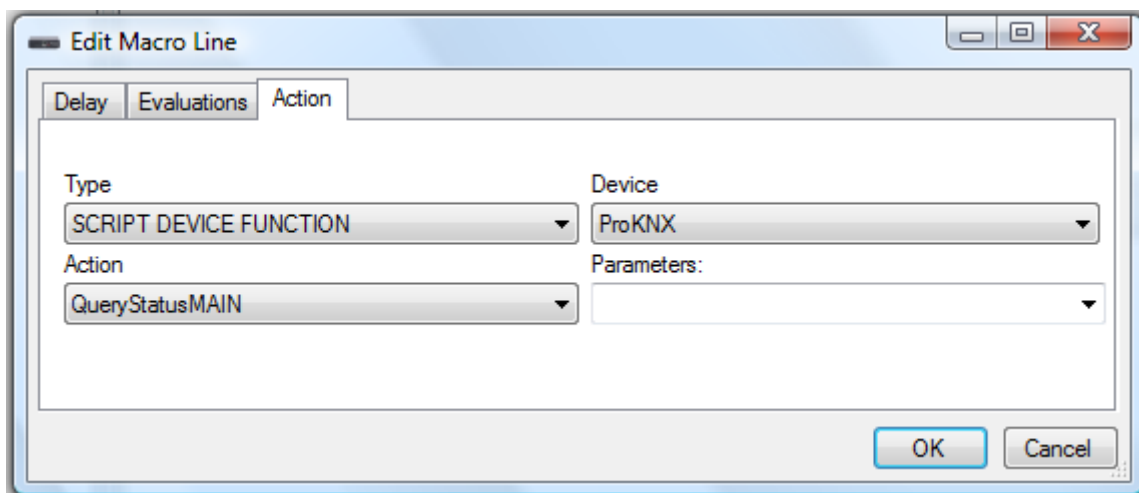


Step3: Define the looping macro

The looping macro makes sure, that the communication port stays open. Highlight “Looping Macro” as shown and push the “Edit” button.



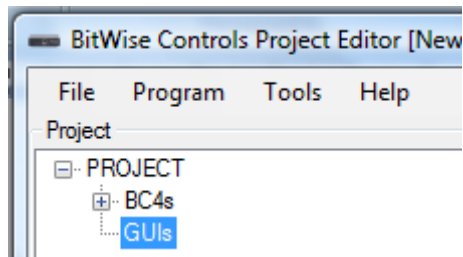
Now click right in the lower / right window and select “Insert Command”. Edit the following parameter:



Now choose the “Delay” Tab and insert 25 sec. Close the dialog with OK and don’t forget to push the “Save” button.

Step 4: Import the GUI Group

Highlight “GUIs”, click right and select “Import GUI Group”.



Browse for the “ProKNX_2way_GUI_GROUP_v1.0.grl” file of the software package and load it. The images and the script will be loaded in the default directories of the Project Editor!

IMPORTANT: If you want to use an existent GUI GROUP, you have to create in this existent group a script, using the “Add Script” command. Then open the “SCRIPT ProKNX_Query” which is located in the imported “ProKNX” GUI group, and copy the content to the new script that you have created in your existent GUI group.

Step 5: Configuring Buttons

Of course you may use any button design together with this ProKNX software for bitwise controls.

However you can use also some prepared buttons that you may easily copy and paste from the “Templates” page. You’ll find:

- Switching Buttons
- Dimming Buttons / Sliders
- Shutter Buttons
- Scene Buttons

Copy the button with the corresponding function from the template page to your GUI page.

Buttons may be composed out of two, three or more single elements. Try to highlight all elements and copy and paste them together!

Button actions

Button actions may be done with “scriptdevice” functions as well as with “GUI script” functions. Scriptdevice functions may be chosen in a drop down menu – all necessary parameters are shown. GUI script functions have to be called in a correct syntax.

Please note, some Control Functions need to be defined as action “On Press” and “On Release”!

A function that is used as On Press action has the prefix “oP_”, a function for On Release action has the prefix “oR_”.

ScriptDevice Functions

To select a script device function, use the following parameter for “Type” and “Device”:

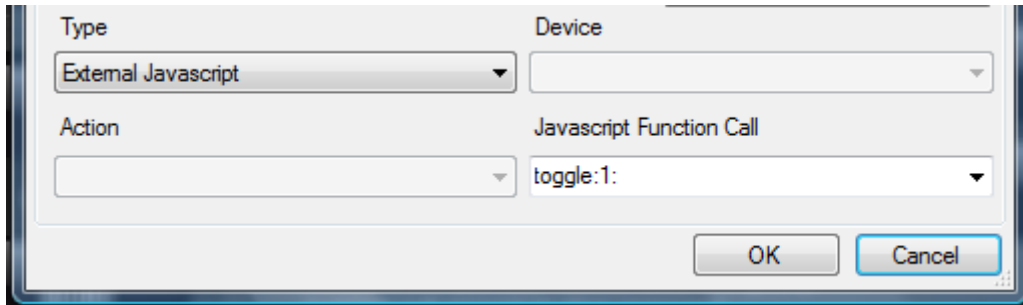
Following actions are possible:

action name	Parameter	Description
sendVal	obj, val	sends a 1bit to 1 byte Value to the specified object in the gateway
shutterUp	startObj	Starts a movement upwards of a shutter, “startObj” has to be assigned to the Start command group address.
shutterDown	startObj	Starts a movement downwards of a shutter, “startObj” has to be assigned to the Start command group address.
shutterStop	stopObj	Starts a movement of a shutter, “stopObj” has to be assigned to the Stop command group address.
ON	obj	sends an ON to the specified object in the gateway
OFF	obj	sends an OFF to the specified object in the gateway
callScene	obj, sceneNumber	calls the scene “sceneNumber”. Assign “obj” to the scene groupe address.
saveScene	obj, sceneNumber	saves the scene “sceneNumber”. Assign “obj” to the scene groupe address. Remember that scene saving has to be allowed in the parameters of the actuator.
oP_dimmUp	obj4bit	Use this action with the “On Press” selection. It will start increasing a dimmer. Remember: the used object “obj4bit” has a 4 Bit format.
oP_dimmDown	obj4bit	Use this action with the “On Press” selection. It will start decreasing a dimmer. Remember: the used object “obj4bit” has a 4 Bit format.
oR_dimmStop	obj4bit	Use this action with the “On Release” selection. It will stop dimming. Remember: the used object “obj4bit” has a 4 Bit format.

Note: Script device functions may be easily used also in BC4 macros. This allows combining KNX commands with media system commands (e.g. IR or serial commands).

GUI Script Functions

Some more performing functions have to be called as an external Javascript. To select an external Javascript function, use the following parameter for “Type”, and fill in the complete function as shown in the following table:



Function	Parameter	Description
toggle	:<obj>:	Toggles the binary value in the specified object in the gateway
oP_handleScene	:<obj>:<sceneNumber>:	One button scene function: Pushing button shortly will call the scene, pushing it long will save the scene
oR_handleScene	:<obj>:<sceneNumber>:	
oP_moveUp	:<startObj>:<stopObj>:	Two button shutter and venetian blind function: Similar to wall buttons, short push will stop movement and change angle of the slats of a blind. Long push will start the movement.
oR_moveUp	:<startObj>:<stopObj>:	
oP_moveDown	:<startObj>:<stopObj>:	
oR_moveDown	:<startObj>:<stopObj>:	
oP_inc2Byte	:<obj>:<delta>:[<enableObj>:]	Function allows to increment (or decrement, if <delta> is negative) a 2 Byte value that is specified by <obj>. Note that <delta> may be a float value; however use the point as decimal point (not a comma). The < enableObj > parameter (1Bit or 1Byte) is optional and enables this function if the value of the specified object = 1. This is quite useful as you can directly specify the object of the operation mode of a room controller: Set point adjusting will be only possible in comfort mode.
oR_inc2Byte	:<obj>:<delta>:[<enableObj>:]	

Step 6: Configuring Feedback

You may use button states, labels and sliders to display feedback information. Buttons will show released state, when feedback is zero and pressed state in all other cases.

Types of feedback

8 Types of feedback are available and may be used for “Feedback text ID” and/or “Feedback state ID”:

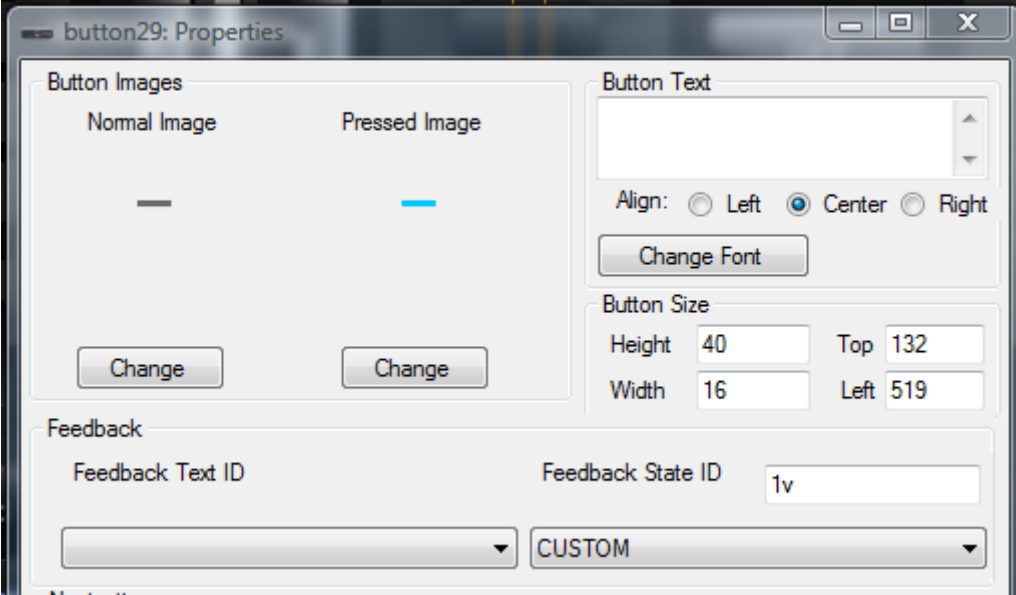
Feedback ID	Parameter	May be used as Text ID / State ID	Description
ProKNX connected		State ID	Shows when gateway is connected
CUSTOM	<objNumber>v	State ID + Text ID	Returns the value of the gateway object “objNumber” as an integer from 0 to 255. The gateway object may have the size of 1 bit to 1 byte.
CUSTOM	<objNumber>p	Text ID	Returns the value of the gateway object “objNumber” as a string from 0% to 100%. The gateway object may have the size of 1 byte.
CUSTOM	<objNumber>s	State ID	Returns the value 0, 1, 2, 3 or 4. The gateway object “objNumber” may have the size of 1 byte. This is only used to show the position of shutters.
CUSTOM	<objNumber>:<sceneNumber>	State ID	Returns 0 or 1 according to the selected scene “sceneNumber” in object “objNumber”. This is only used for the button feedback of the single scene button control.
CUSTOM	<objNumber>T	Text ID	Returns the value of the gateway object “objNumber” as a temperature string (e.g. 21.2 °C). The gateway object must have the size of 2 bytes.
CUSTOM	<objNumber>L	Text ID	Returns the value of the gateway object “objNumber” as a luminosity string (e.g. 51280 lux). The gateway object must have the size of 2 bytes.
CUSTOM	<objNumber>S	Text ID	Returns the value of the gateway object “objNumber” as a speed string (e.g. 20.5 km/h). The gateway object must have the size of 2 bytes.

Examples

Example1:

State image

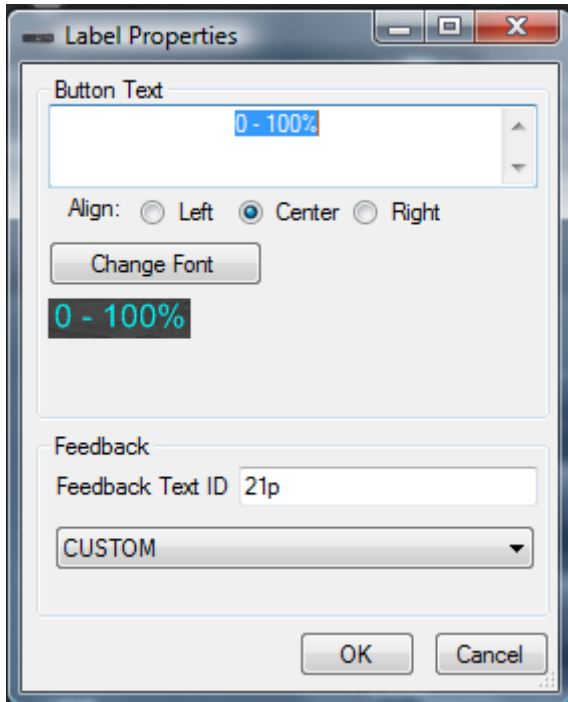
The following button configuration will change the image of the button according to the state of object 1. When the value of object 1 is zero, “Normal Image” will be shown, in all other cases the “Pressed Image” will be shown.



Example2:

Simple text display

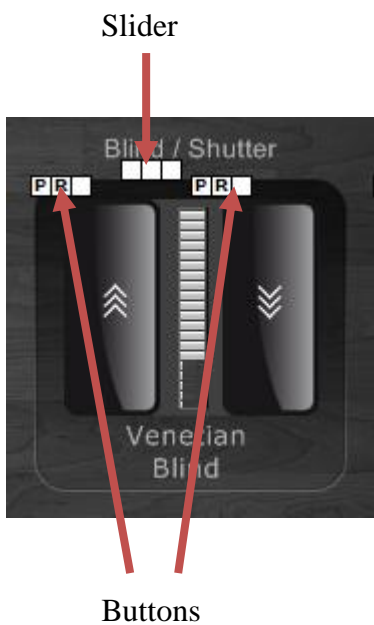
The following label configuration will show as content of the label the value of object 21 as per cent value.

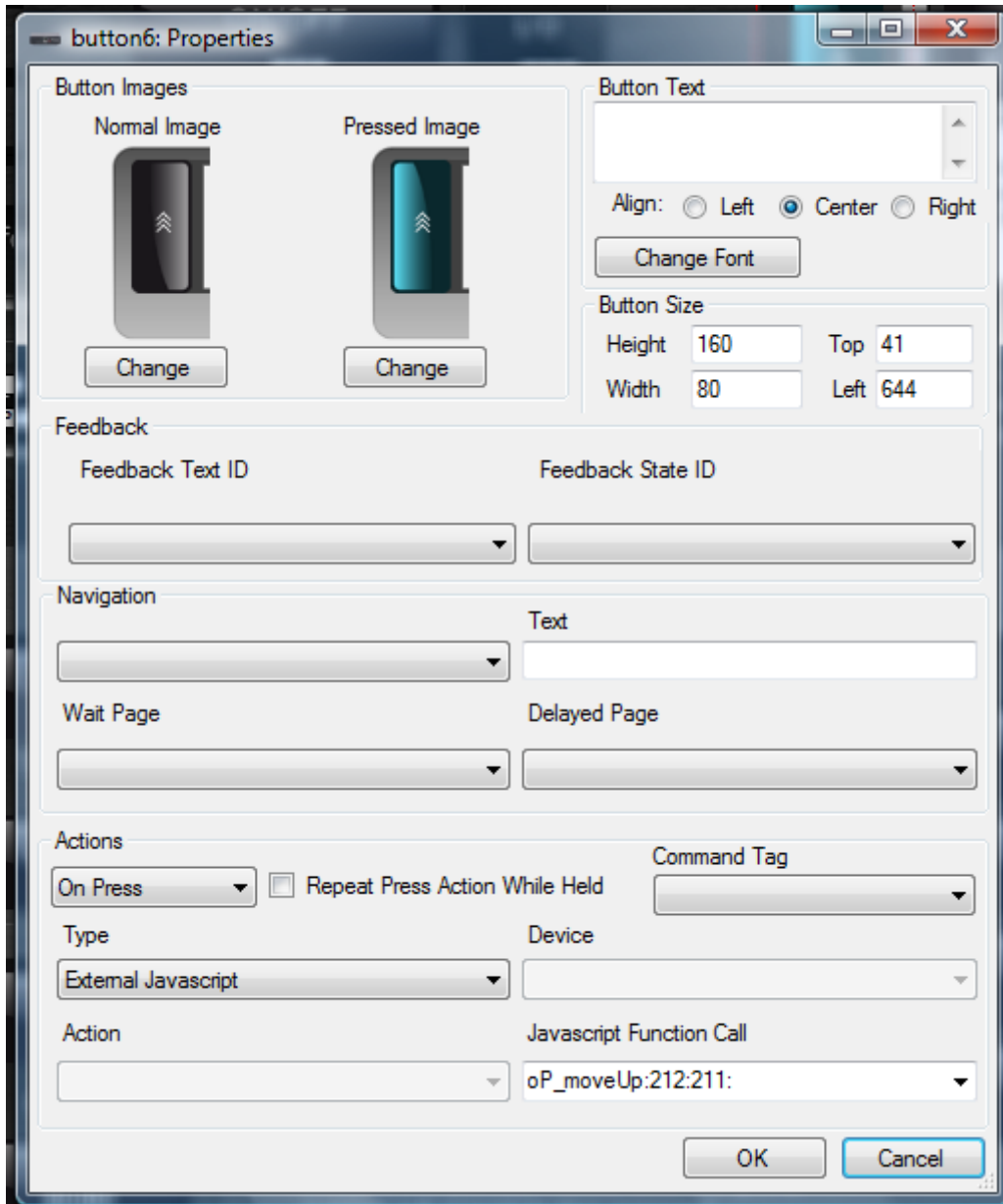


Example3:

Shutter Control

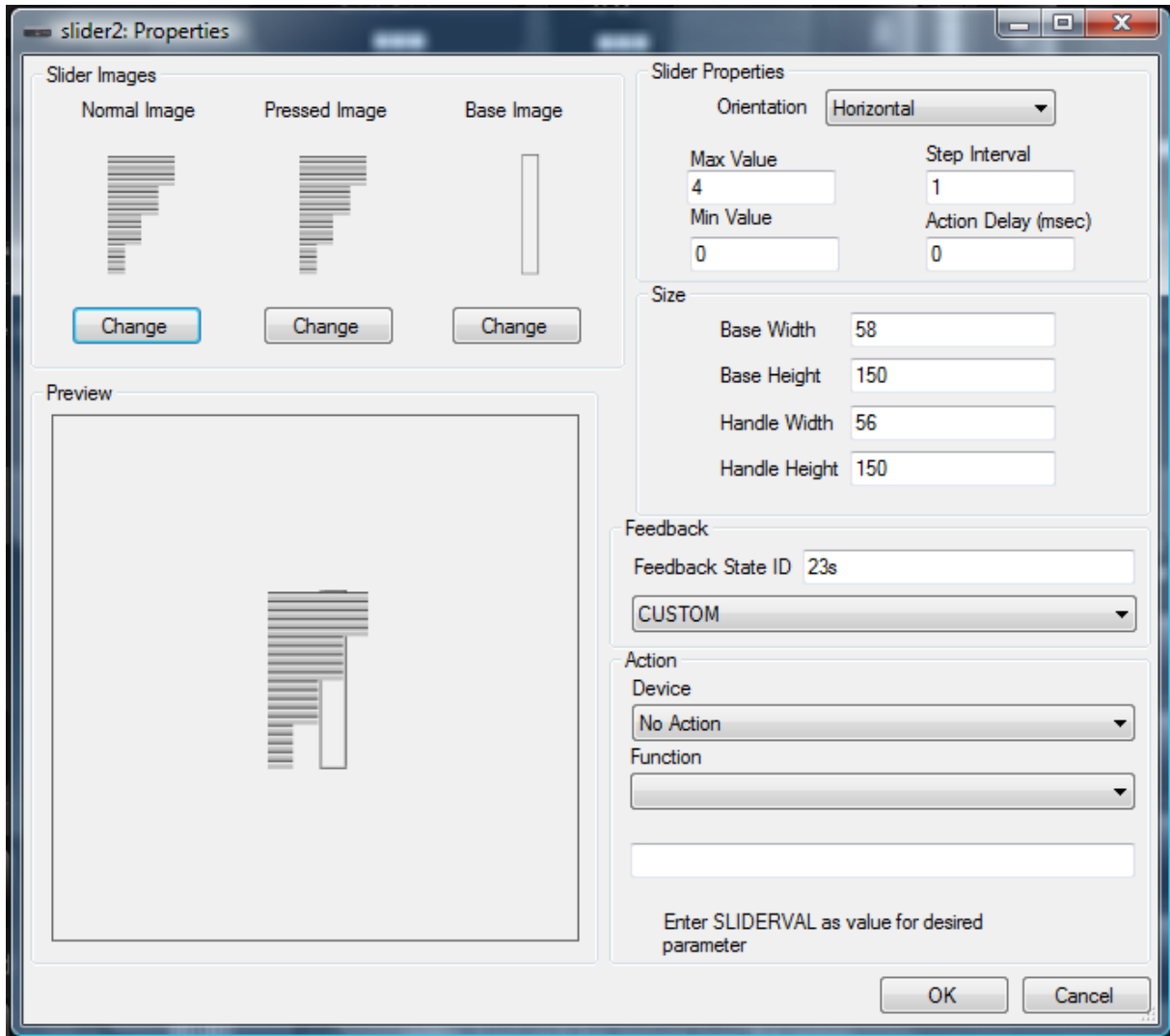
The following configuration is for shutter control and permits to show the shutter position. Two buttons are positioned above a slider, which will be moved in five different positions.





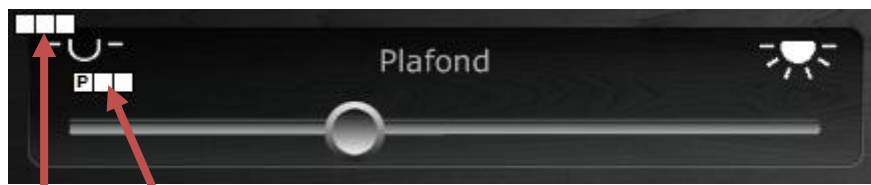
Please remember the configuration:

Function	Parameter	Description
oP_moveUp	:<startObj>:<stopObj>:	Two button shutter and venetian blind function: Similar to wall buttons, short push will stop movement and change angle of the slats of a blind. Long push will start the movement.
oR_moveUp	:<startObj>:<stopObj>:	
oP_moveDown	:<startObj>:<stopObj>:	
oR_moveDown	:<startObj>:<stopObj>:	

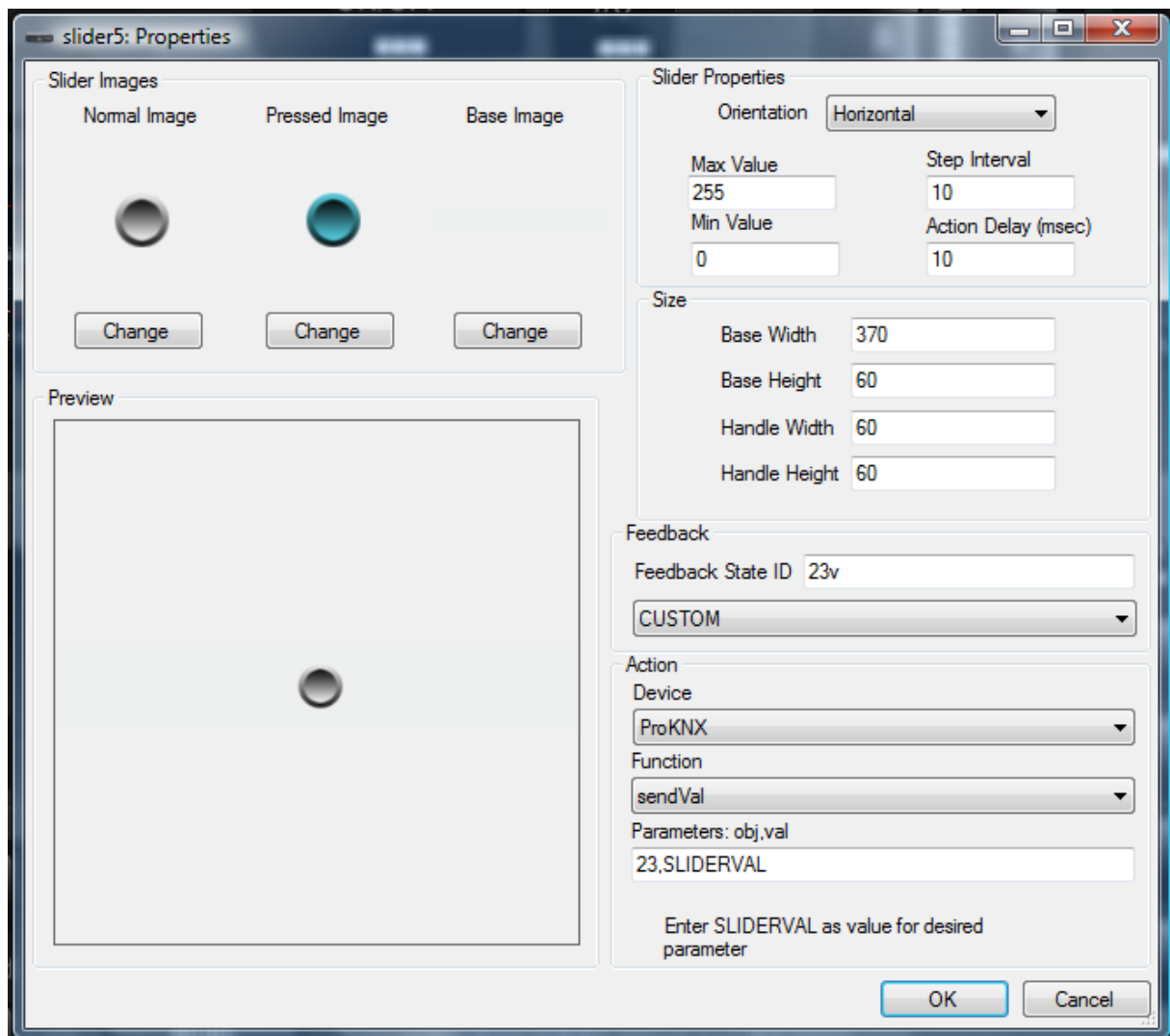


Example 4:

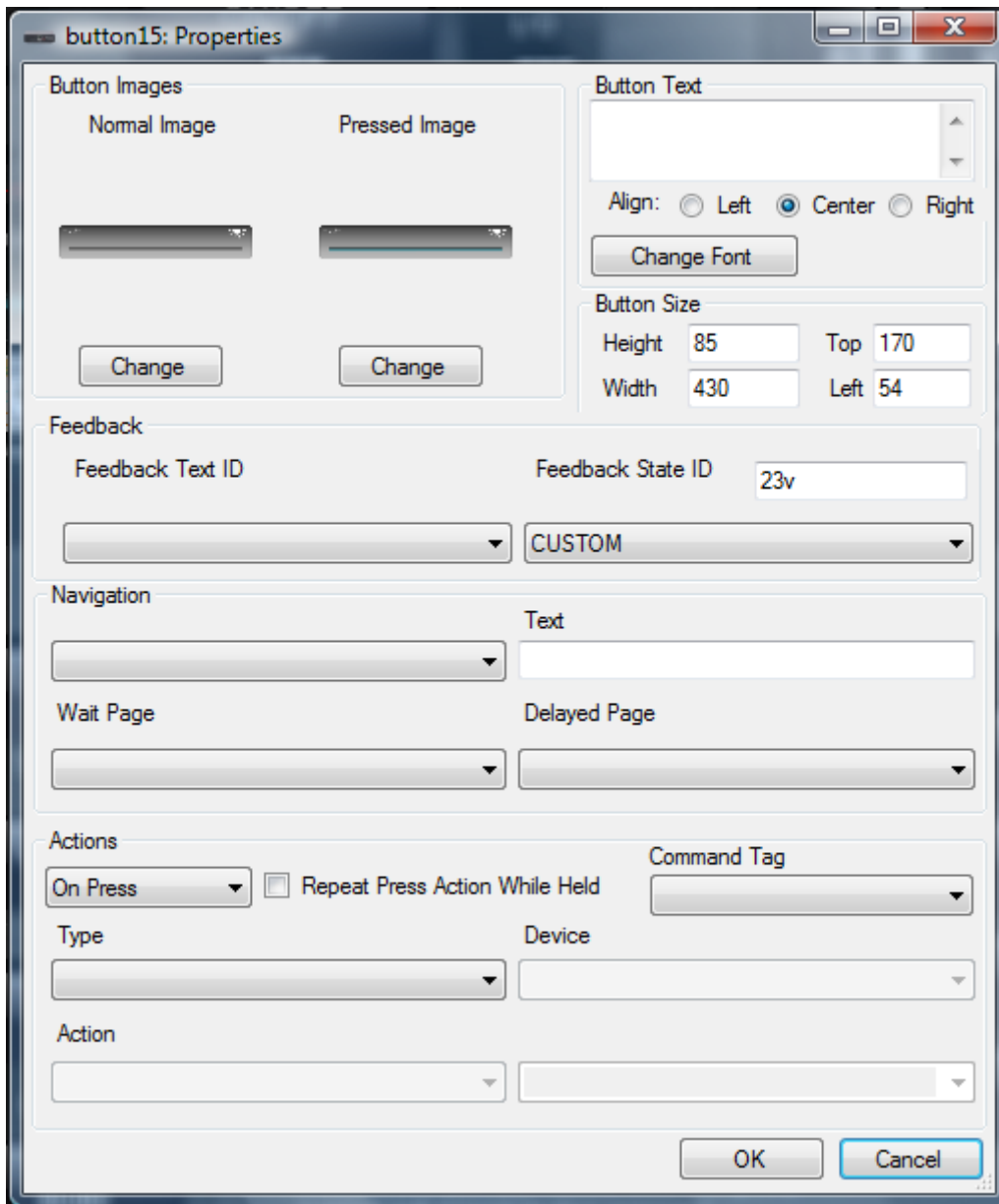
Slider for Dimming



Button
Slider



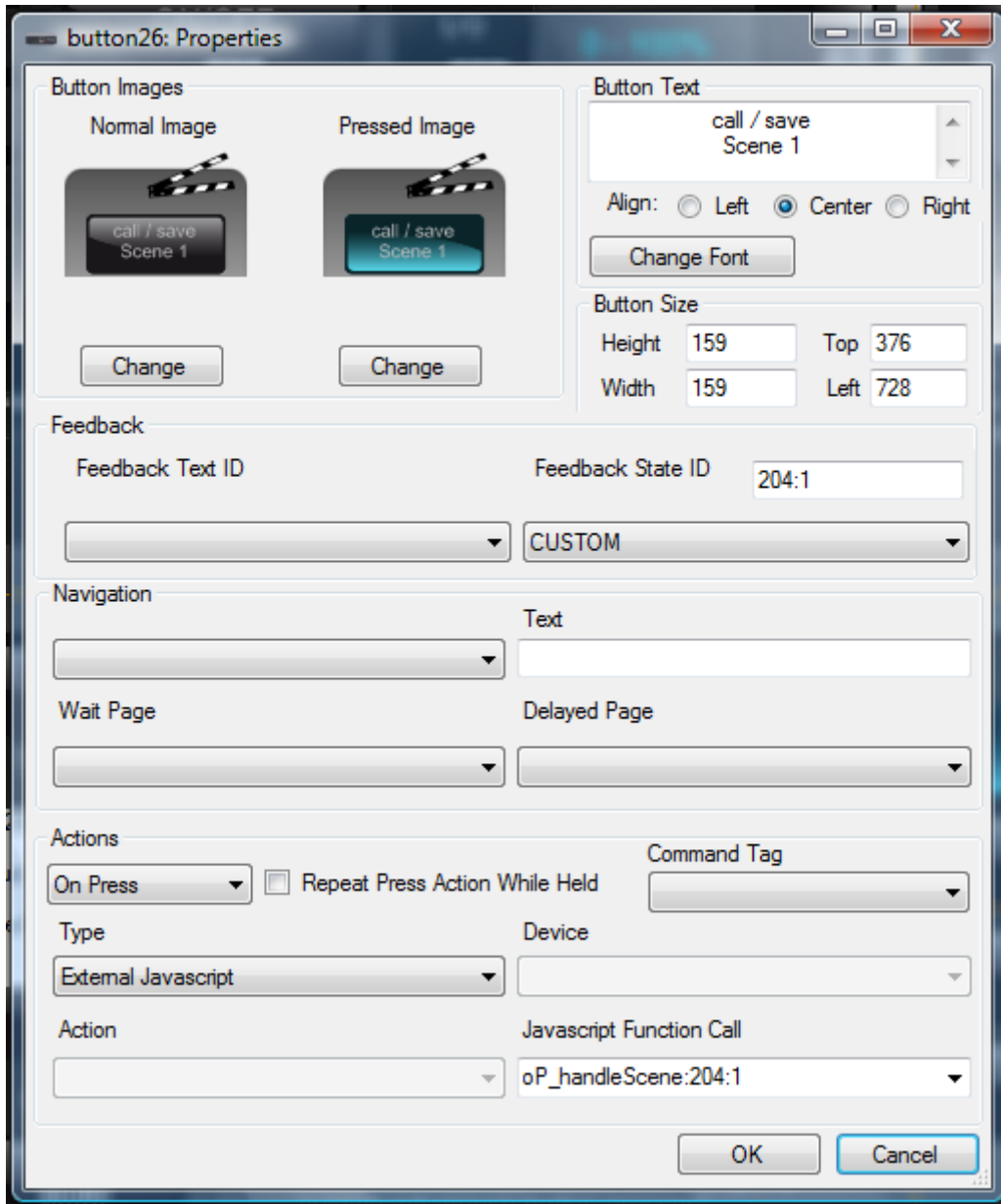
Note: There is an invisible Base Image



Example 5:

One button Scene control





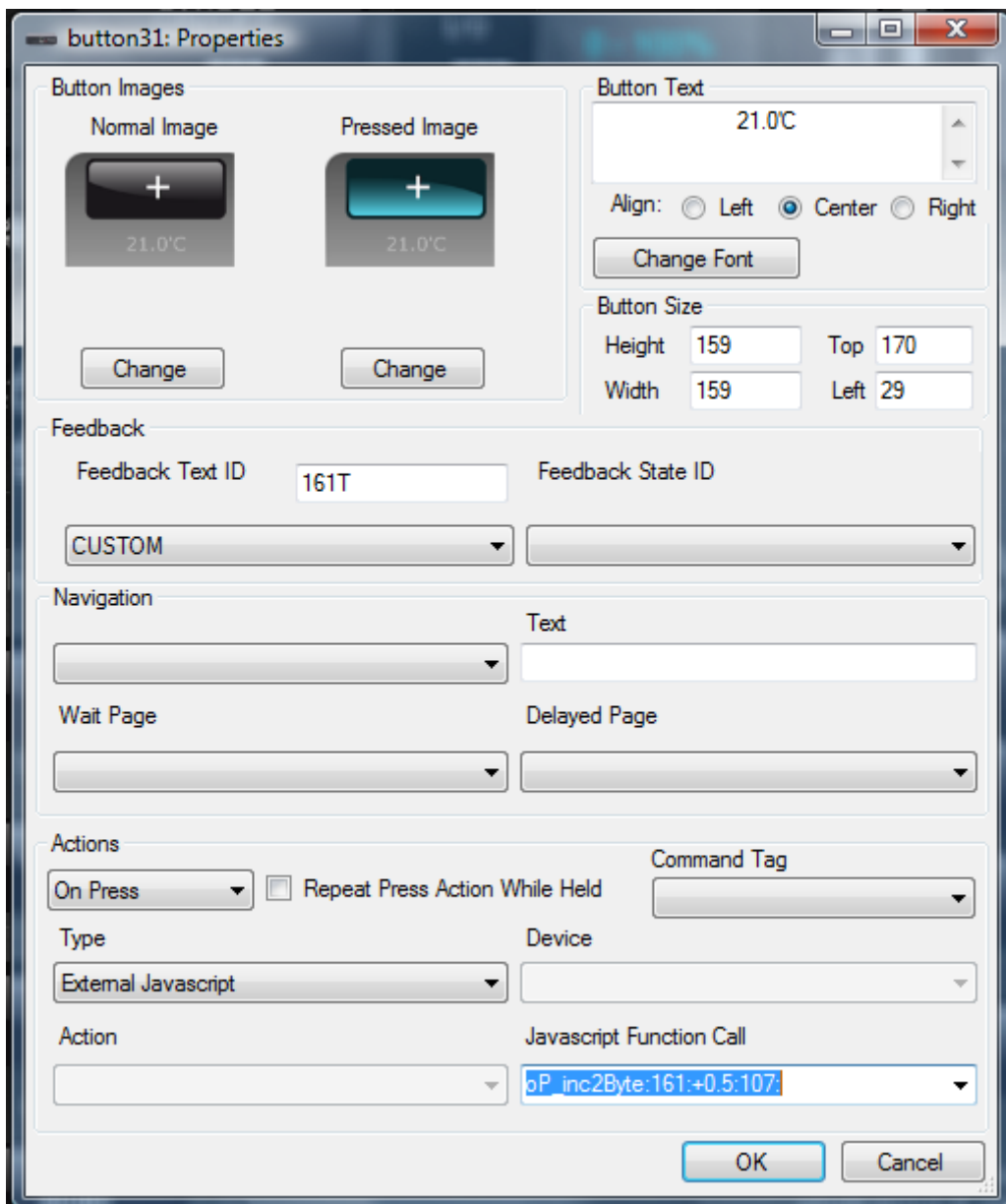
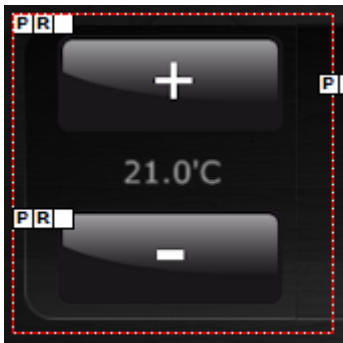
This smart control button will call a scene when it is pressed less than 2 seconds. If it is pressed longer than 2 seconds it will save the scene and will toggle it's colour, so that the user is informed about scene saving.

Please remember the configuration:

Function	Parameter	Description
oP_handleScene	:<obj>:<sceneNumber>:	One button scene function: Pushing button shortly will call the scene, pushing it long will save the scene
oR_handleScene	:<obj>:<sceneNumber>:	

Example 6:

Temperature Set Point Adjustment



These control buttons will change the set point of Room Temperature Controller. Pressing the button once will change the value (specified by the first parameter) with the specified delta (which is the second parameter). If you hold it down, it will continuously change the value until the button is released.

This function is enabled, if you do not use a third parameter, or if the here specified object has the value “1”.

Please remember the configuration:

Function	Parameter	Description
oP_inc2Byte	:<obj>:<delta>:[<enableObj>:]	Function allows to increment (or decrement, if <delta> is negative) a 2 Byte value that is specified by <obj>. Note that <delta> may be a float value; however use the point as decimal point (not a comma). The < enableObj > parameter (1Bit or 1Byte) is optional and enables this function if the value of the specified object = 1. This is quite useful as you can directly specify the object of the operation mode of a room controller: Set point adjusting will be only possible in comfort mode.
oR_inc2Byte	:<obj>:<delta>:[<enableObj>:]	

Good to know:

Most of the KNX room temperature controllers are providing one object to receive the set point temperature, and one object that will send the actual set point according to the operation mode. Please define two group addresses for these two communication objects. Then insert the 2Byte object of this function first in the group address to set the set point, and then a second time in the group address that will give the feedback of the actual set point.

We highly recommend changing the set point only when comfort mode is active. This may be done very easily; you just have to define as the third parameter the object that is assigned to the operation mode.

You may change the text that will be displayed in the GUI when this function is not enabled: Please open the GUI script “ProKNX_Query” and edit the marked line:

```

6 //*****
7 //***** you may adjust the next line *****
8 //***** that will define the message *****
9 //***** when 2Byte value changing is *****
10 //***** not possible: *****
11 //*****
12
13 var message = "select comfort!"; // only change text within the apostrophes|
14
15 //*****
16 //*****
17 //*****
18 //***** don't change anything behind this line *****
19 //*****

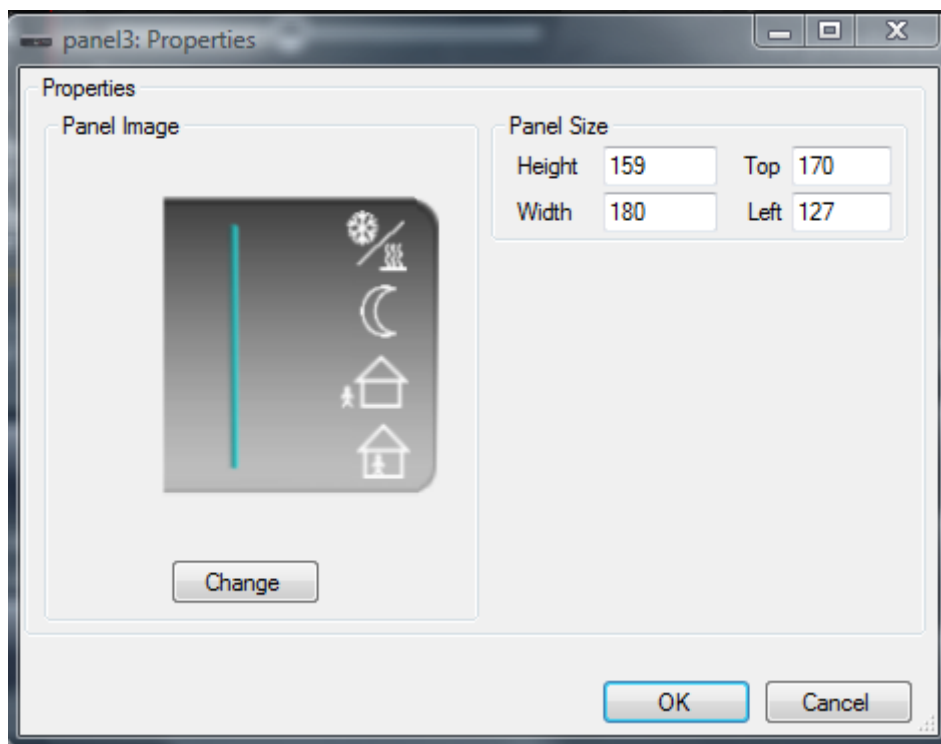
```

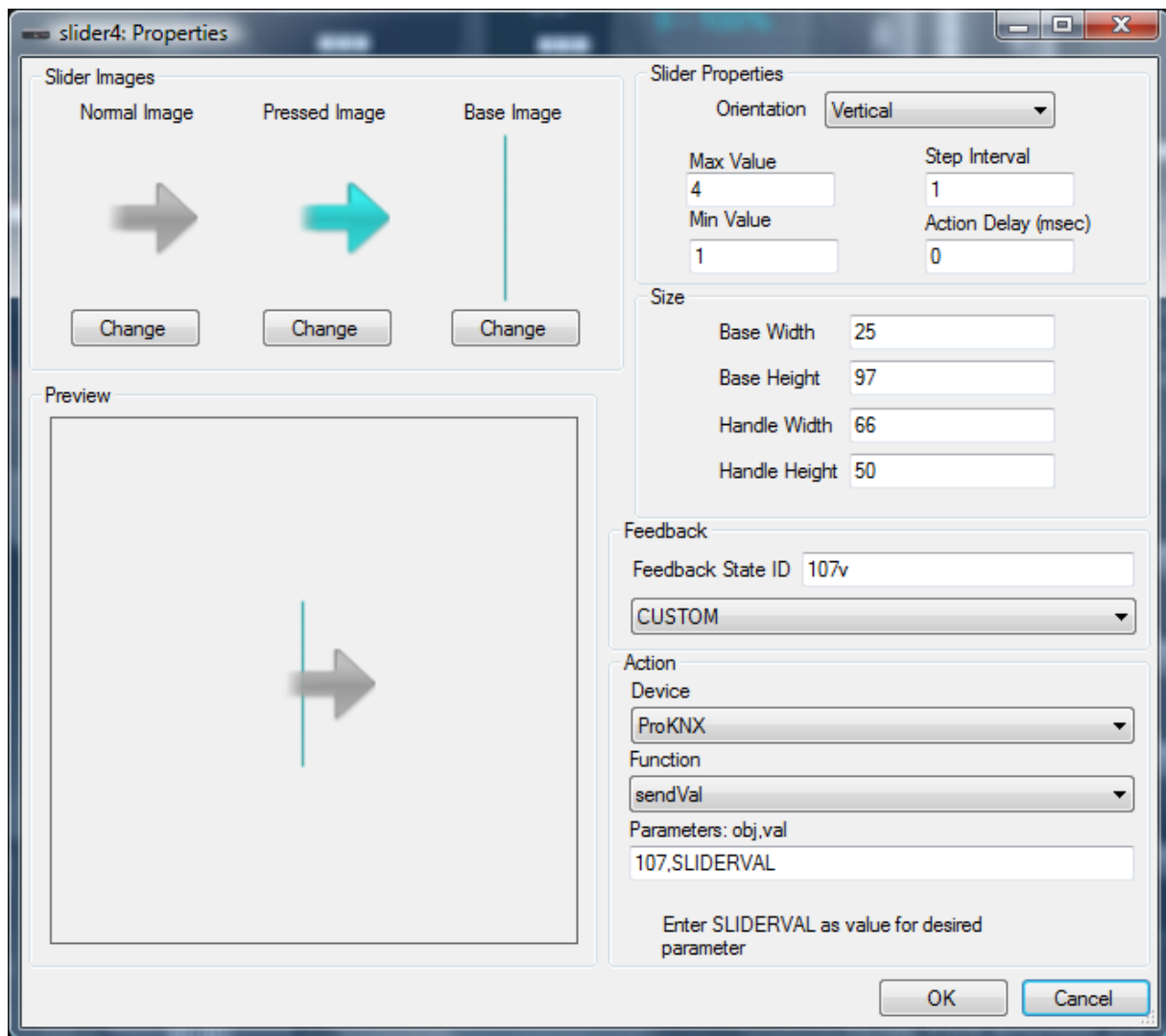
Example 7:

Adjusting and showing Controller mode



This GUI element consists out of one Panel and one Slider:





Object 107 is assigned to the controller mode group address (1Byte).

The slider has only 4 positions:

- 1: comfort mode
- 2: standby
- 3: night
- 4: freeze/heat protection

About

PROKNX for bitwise controls

Bidirectional KNX interface software
for BC4 Project Editor v1.4

Copyright 2011
by BleuCommAzur EURL
All rights reserved

If you have any suggestions to this software, please don't hesitate to contact us.

This work may be used in whole or part for commercial or non-commercial use if it is distributed with one of the "PROKNX" IP Gateways. The Gateway may be ordered directly through our website www.proknx.com. Redistribution of the software or the documentation without the Gateway is prohibited and requests for same should be sent to the author.

"bitwise controls" is a trademark of Bitwise Controls LLC, USA.