ProviewR
OPEN SOURCE PROCESS CONTROL

Home automation with Zigbee

Getting Started Guide

Beta version

2025-02-05

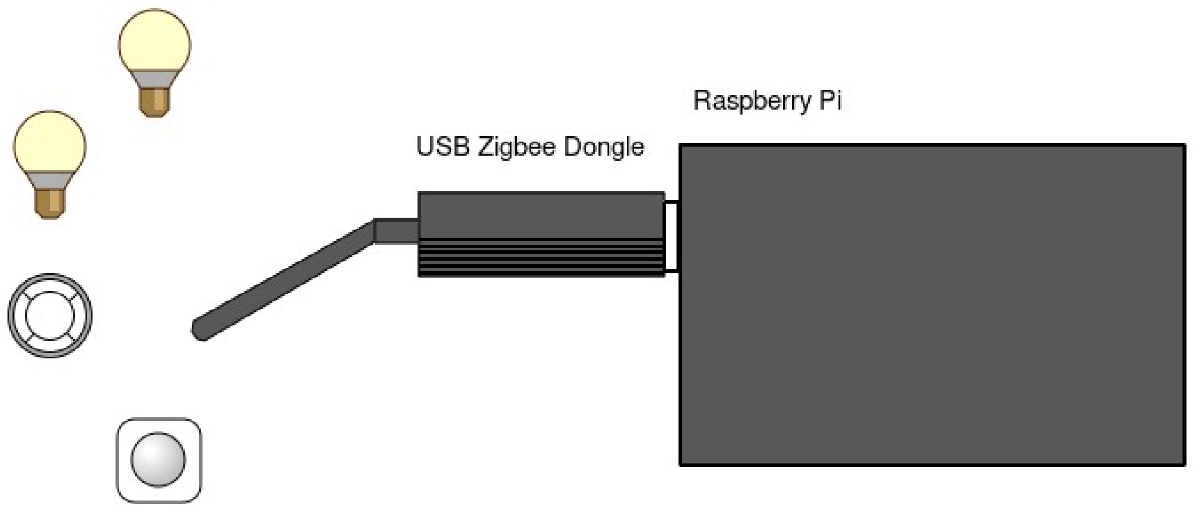
Table of Contents

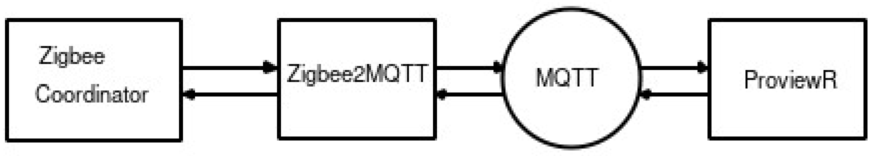
About this guide

This guide will take you through the steps of configuring and running a small ProviewR project with zigbee home automation devices. It’s an extension of to the Getting started guide and sections described in the Getting started guide is not repeated here. The guide does not intend to be comprehensive. For detailed documentation, please consult the Designer's Guide or the Ge (graphical editor) Manual. These documents are available at the ProviewR [http://www.proview.se] site.

Introduction

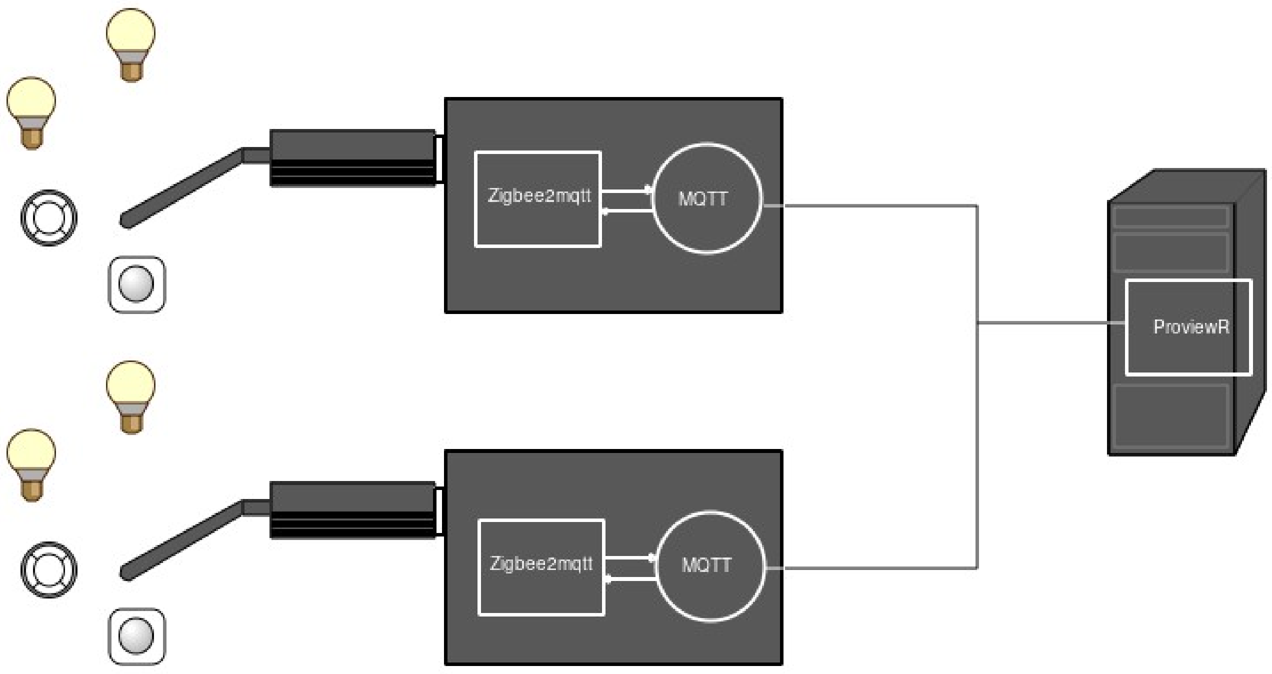
In this guide we will use four Zigbee devices from IKEA, a Zigbee USB dongle (Sonoff Zigbee 3.0 USB Dongle Plus), a Raspberry Pi, the software zigbee2mqtt and a MQTT server to create a minimal home automation system.





Zigbee2MQTT is an open source software that makes it possible to read and write data from an to the devices via a MQTT broker. Zigbee2MQTT supports over 3000 devices from 400 vendors. Before you order any devices make sure they are supported by Zigbee2MQTT.

As MQTT broker we will use Mosquitto. The broker is accessible over the network and ProviewR can handle multiple brokers which makes a variety of different configurations possible. In the figure below two Raspberry Pi are handled by ProviewR running in a PC. We will though create a minimal configuration and pack all into our Raspberry Pi.


Zigbee2mqtt
MQTT
Zigbee2mqtt
MQTT
ProviewR

Installation

Install the MQTT broker. In this guide the mosquitto server is used

```
sudo apt install -y mosquitto-server
```

It’s also convenient to install the mosquitto client for test and trouble shooting

```
sudo apt install -y mosquitto-clients
```

We will also need the mosquitto development archive to build our project in ProviewR

```
sudo apt install -y mosquitto-dev
```

I you want authentication to the mosquitto server, this is required for remote access, you create a password file on /etc/mosquitto and add this file in /etc/mosquitto/mosqitto.conf

```
sudo su
cd /etc/mosquitto
mosquitto_passwd -c mqtt_passwd.txt pwrp
Password:****
```

Edit mosqitto.conf and add the line

```
password_file /etc/mosquitto/mqtt_passwd.txt
```

Install zigbee2mqtt. Follow the instructions in https://www.zigbee2mqtt.io, Getting started / Installation / Linux. There are also instructions on how to download the Zigbee software to the USB dongle. For some dongles (as the Sonoff Zigbee 3.0 USB Dongle Plus the we are using) the software is preloaded.

Install ProviewR development package, pwr61 V6.1.4 or later. Download the package from sourceforge and install with dpkg.

```
sudo apt-get install -y libgtk-3-0 libasound2 libdb5.3 libdb5.3++ \
   libsqlite3-0 librsvg2-2 g++  xterm libmariadb3 librabbitmq4 \
   libusb-1.0-0 libhdf5-openmpi-103 librabbitmq4 libmosquitto1 \
   libgstreamer1.0-0 libgstreamer-plugins-base1.0-0 openjdk-17-jdk \
   xterm xfonts-100dpi sudo procps libpython3-dev python3


dpkg -i pwr61_6.1.4-i_arm64.deb
```

Devices

We will configure four devices

- An IKEA color bulb, LED2109G6, named ‘Bulb1’
- An IKEA bulb, LED1836G9, named ‘Bulb2’.
- An IKEA remote control, E1810, named ‘RemoteControl’.
- An IKEA motion sensor, E2134 (vallhorn), named ‘MotionSensor’.

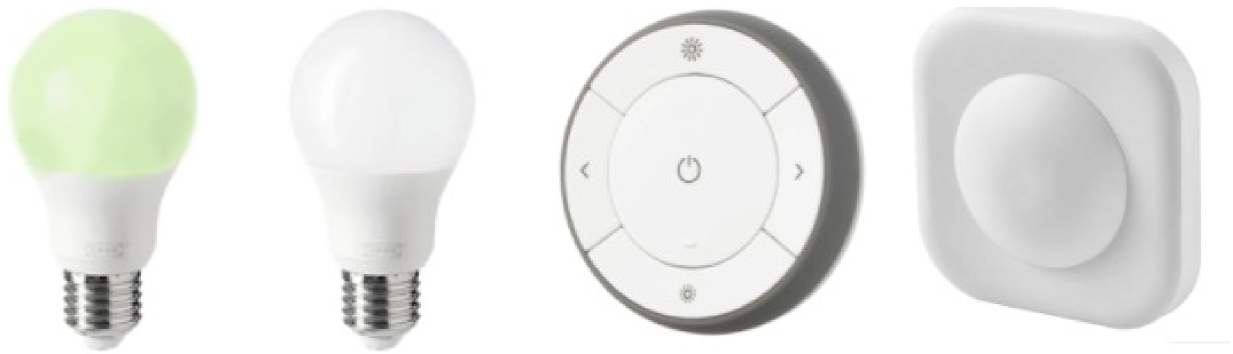


Fig Devices

Pairing

The devices should be paired with the Zigbee coordinator. Make sure that zigbee2mqtt is set in paring mode before you start the pairing.

The pairing mechanism differs between devices. The bulb for example should be turn on and off a number of times, while the remote control and the motion sensor has a pair button that should be pressed four times. See the documentation of each device for more info.

Check in the zigbee2mqtt log that the pairing is successful and note the device identity. Below is the log for the color bulb that has the identity '0x8c65a3fffe992a36'. We will need this later when we configure the device in ProviewR.

```
[2024-11-01 10:48:29] info: zh:controller: Successfully interviewed
'0x8c65a3fffe992a36'
[2024-11-01 10:48:29] info: z2m: Successfully interviewed
'0x8c65a3fffe992a36', device has successfully been paired
[2024-11-01 10:48:29] info: z2m: Device '0x8c65a3fffe992a36' is supported,
identified as: IKEA TRADFRI bulb E26/E27, color/white spectrum, globe, opal,
806 lm (LED2109G6)
```

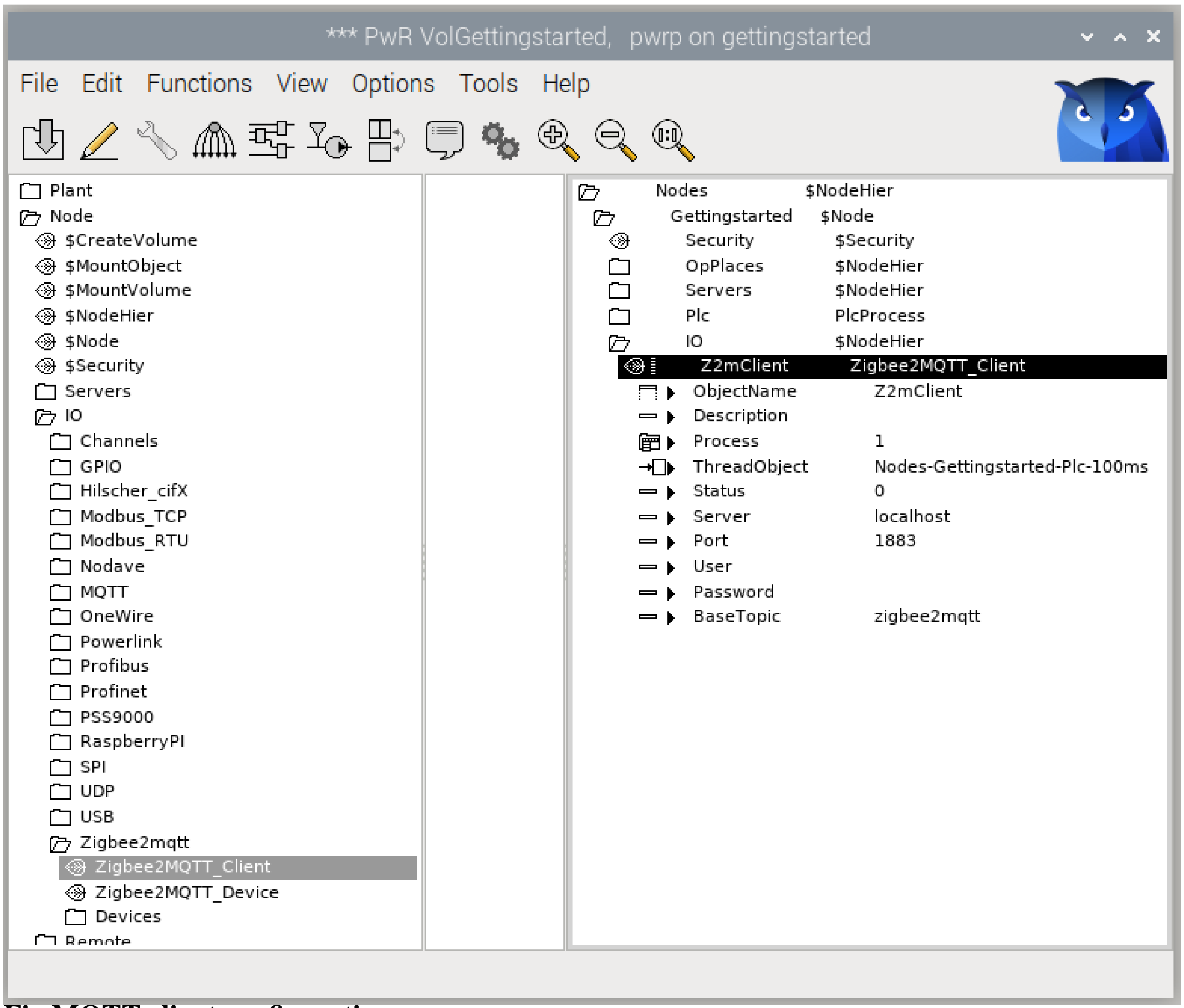
ProveiwR configuration

Create a project

Follow the instruction in Getting started guide to create a project.

Configure the MQTT client

The connection to the MQTT server is done with a Zigbee2MQTT_Client object. The object is found under Node/IO/Zigbee2mqtt in the palette. Positioned the object under the IO object in the Node hierarchy. If the server is running in a remote node, the host name or IP address should be inserted in the Server attribute. We will use a local server and use the default value 'localhost'. If the server requires user name and password this should also be inserted.



Nodes	$NodeHier
Gettingstarted	$Node
Security	$Security
OpPlaces	$NodeHier
Servers	$NodeHier
Plc	PlcProcess
IO	$NodeHier
Z2mClient	Zigbee2MQTT_Client
ObjectName	Z2mClient
Description	
Process	1
ThreadObject	Nodes-Gettingstarted-Plc-100ms
Status	0
Server	localhost
Port	1883
User	
Password	
BaseTopic	zigbee2mqtt

Fig MQTT client configuration

Device configuration

Devices are separated in two objects, a main object that is positioned in the Plant hierarchy, and an

IO object that is positioned in the Node hierarchy under the Zigbee2MQTT_Client object. For most devices there is also a function object that is used in the plc code.

We open the configurator for the root volume VolGettingStarted and create some hierarchies in the Plant hierarchy to describe our home installation. We also create a PlcPgm object that will be used later for the plc programming.

The main object for our color bulb, IKEA_LED2109G6, is found under Plant/Components/Zigbee2mqtt/IKEA in the palette. We create the object in the 'Livingroom' map and name it 'Bulb1'.

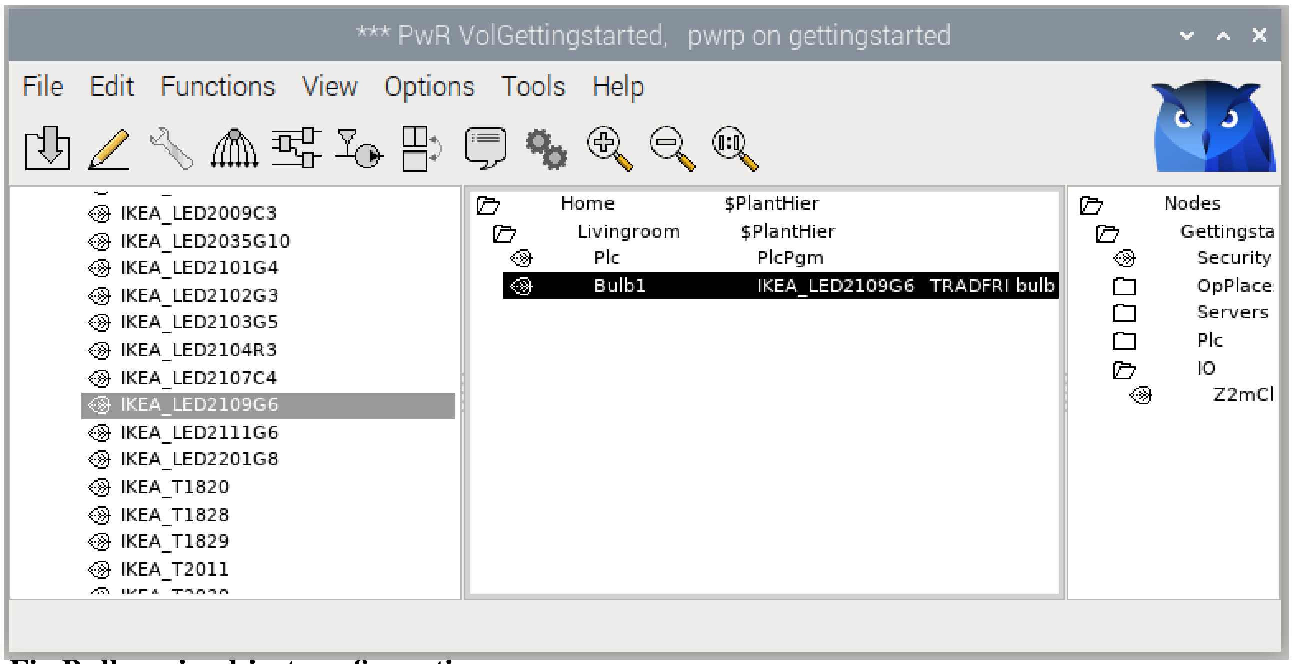
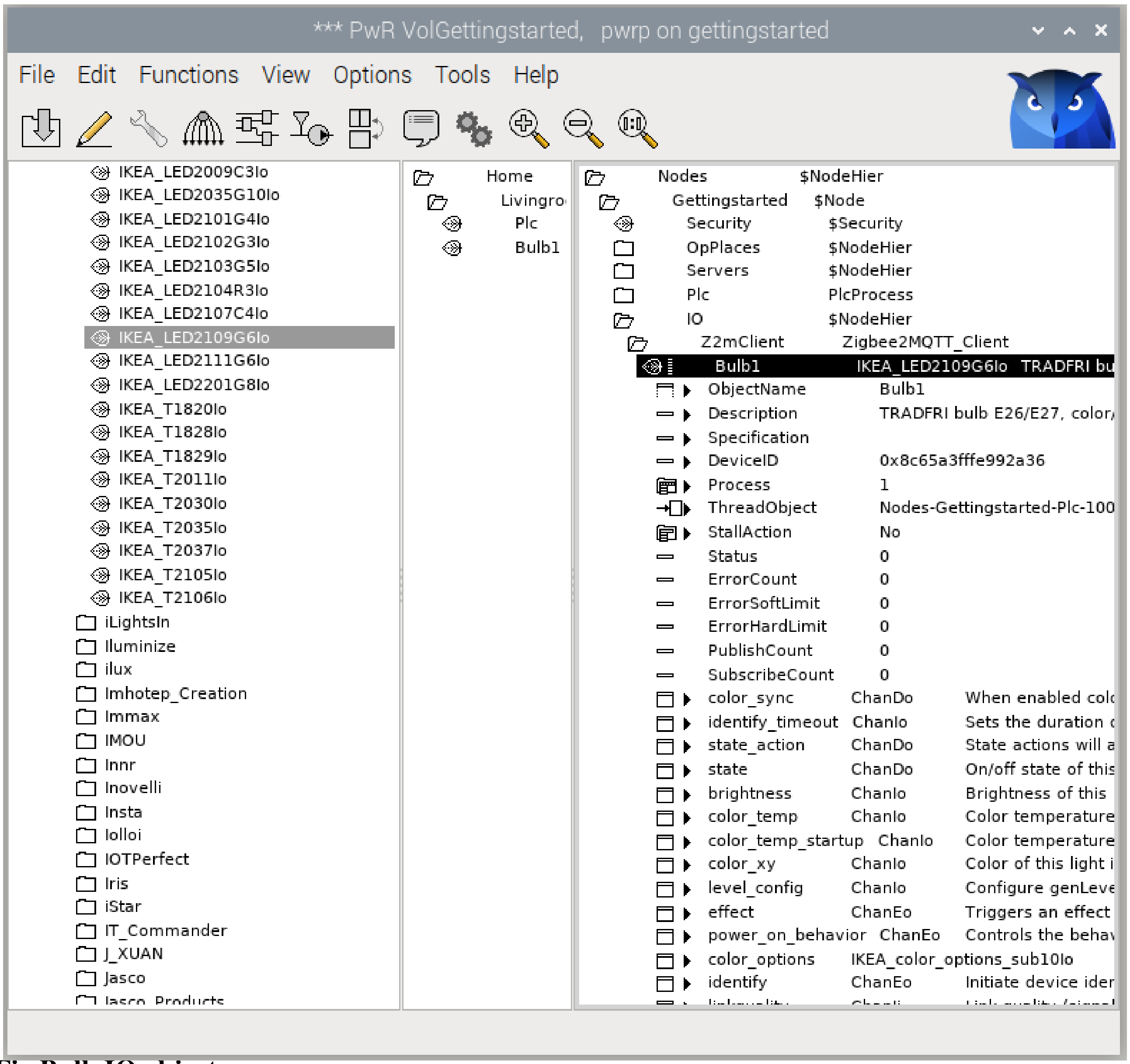


Fig Bulb main object configuration

The IO object is found under Node/IO/Zigbee2mqtt/Devices/IKEA in the palette, and is created below the Zigbee2MQTT_Client object. The device identity, that we fetched from the zigbee2mqtt log, is inserted into the DeviceID attribute.



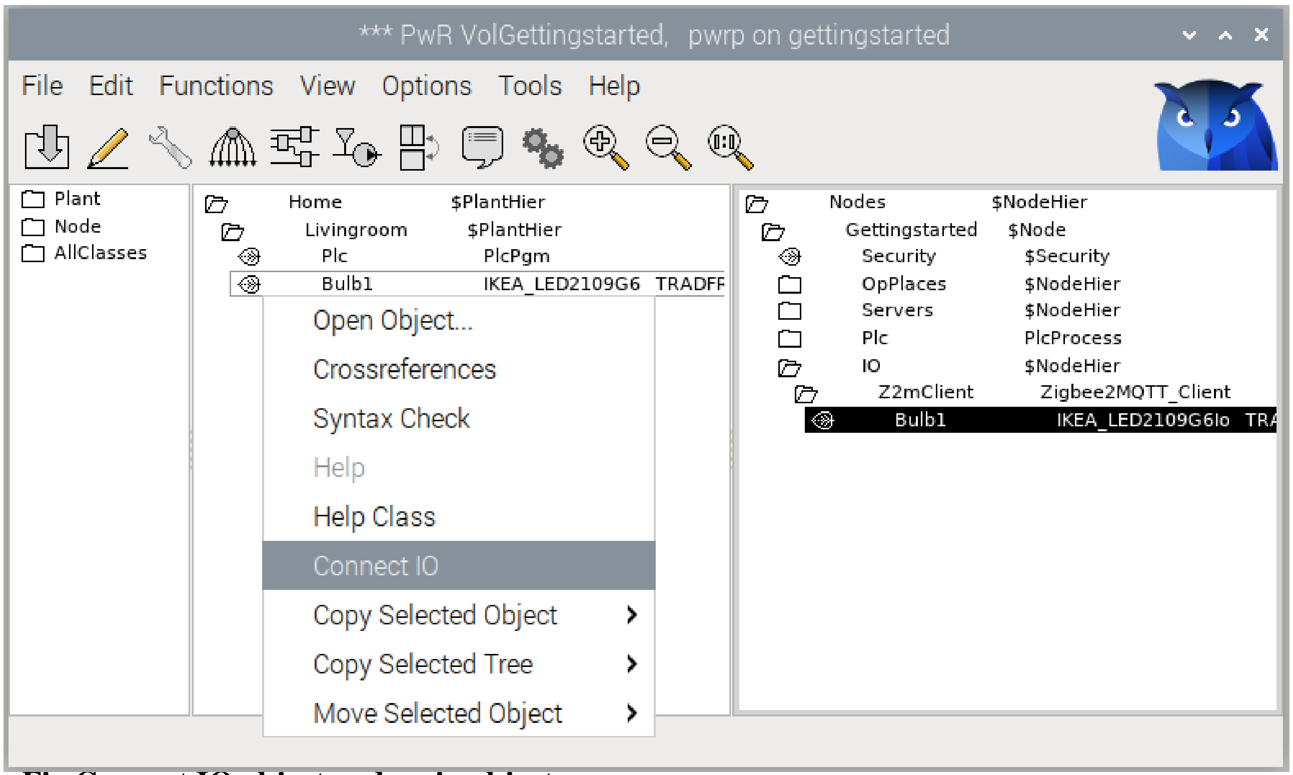
```
IKEA_LED2009C3Io
IKEA_LED2035G10Io
IKEA_LED2101G4Io
IKEA_LED2102G3Io
IKEA_LED2103G5Io
IKEA_LED2104R3Io
IKEA_LED2107C4Io
IKEA_LED2109G6Io
IKEA_LED2111G6Io
IKEA_LED2201G8Io
IKEA_T1820Io
IKEA_T1828Io
IKEA_T1829Io
IKEA_T2011Io
IKEA_T2030Io
IKEA_T2035Io
IKEA_T2037Io
IKEA_T2105Io
IKEA_T2106Io
iLightsIn
Iluminize
ilux
Imhotep_Creation
Immax
IMOU
Innr
Inovelli
Insta
Iolloi
IOTPerfect
Iris
iStar
IT_Commander
J_XUAN
Jasco
Jasco_Products
```

```
Home
  Livingro
    Plc
    Bulb1
```

```
Nodes                  $NodeHier
  Gettingstarted       $Node
    Security             $Security
    OpPlaces             $NodeHier
    Servers              $NodeHier
    Plc                  PlcProcess
    IO                   $NodeHier
      Z2mClient            Zigbee2MQTT_Client
        Bulb1                IKEA_LED2109G6Io  TRADFRI bu
          ObjectName           Bulb1
          Description          TRADFRI bulb E26/E27, color/
          Specification
          DeviceID             0x8c65a3fffe992a36
          Process              1
          ThreadObject         Nodes-Gettingstarted-Plc-100
          StallAction          No
          Status               0
          ErrorCount           0
          ErrorSoftLimit       0
          ErrorHardLimit       0
          PublishCount         0
          SubscribeCount       0
          color_sync         ChanDo       When enabled colo
          identify_timeout   ChanIo       Sets the duration o
          state_action       ChanDo       State actions will a
          state              ChanDo       On/off state of this
          brightness         ChanIo       Brightness of this
          color_temp         ChanIo       Color temperature
          color_temp_startup ChanIo       Color temperature
          color_xy           ChanIo       Color of this light i
          level_config       ChanIo       Configure genLeve
          effect             ChanEo       Triggers an effect
          power_on_behavior  ChanEo       Controls the behav
          color_options      IKEA_color_options_sub10Io
          identify           ChanEo       Initiate device ider
```

Fig Bulb IO object

The next step is to connect the IO object with the main object. Select the IO object, right click on the main object, and activate ‘Connect IO’ in the popup menu.

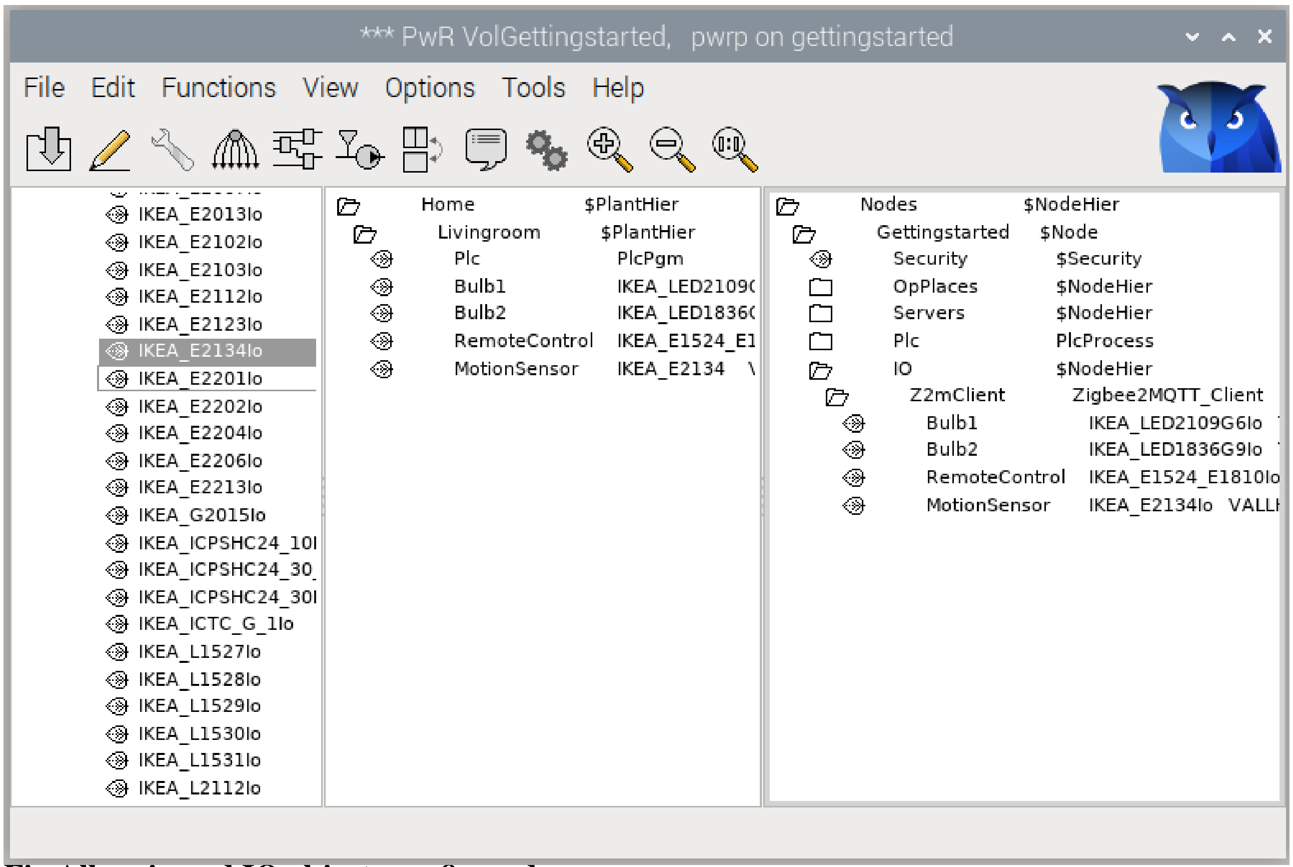


```
Home          $PlantHier
  Livingroom    $PlantHier
    Plc           PlcPgm
    Bulb1         IKEA_LED2109G6  TRADFF
```

```
Nodes              $NodeHier
  Gettingstarted     $Node
    Security           $Security
    OpPlaces           $NodeHier
    Servers            $NodeHier
    Plc                PlcProcess
    IO                 $NodeHier
      Z2mClient          Zigbee2MQTT_Client
        Bulb1              IKEA_LED2109G6Io  TRA
```

Fig Connect IO object and main object

Now we create main object, IO object, insert device identity and activate Conntect IO for the other three devices.



```
IKEA_E2013Io
IKEA_E2102Io
IKEA_E2103Io
IKEA_E2112Io
IKEA_E2123Io
IKEA_E2134Io
IKEA_E2201Io
IKEA_E2202Io
IKEA_E2204Io
IKEA_E2206Io
IKEA_E2213Io
IKEA_G2015Io
IKEA_ICPSHC24_10I
IKEA_ICPSHC24_30_
IKEA_ICPSHC24_30I
IKEA_ICTC_G_1Io
IKEA_L1527Io
IKEA_L1528Io
IKEA_L1529Io
IKEA_L1530Io
IKEA_L1531Io
IKEA_L2112Io
```

```
Home          $PlantHier
  Livingroom    $PlantHier
    Plc             PlcPgm
    Bulb1           IKEA_LED2109(
    Bulb2           IKEA_LED1836(
    RemoteControl   IKEA_E1524_E1
    MotionSensor    IKEA_E2134   \
```

```
Nodes              $NodeHier
  Gettingstarted     $Node
    Security           $Security
    OpPlaces           $NodeHier
    Servers            $NodeHier
    Plc                PlcProcess
    IO                 $NodeHier
      Z2mClient          Zigbee2MQTT_Client
        Bulb1              IKEA_LED2109G6Io
        Bulb2              IKEA_LED1836G9Io
        RemoteControl      IKEA_E1524_E1810Io
        MotionSensor       IKEA_E2134Io   VALLH
```

Fig All main and IO objects configured

Plc programming

In the plc we will program the interaction between the devices. First we will connect the remote control with the color bulb, and make it possible to switch the bulb on and off from the remote, and also to change the brightness of the bulb.

Leave edit mode in the configurator and open the PlcPgm. The function object for Bulb1, IKEA_LED2109G6Fo, is found under Components/Zigbee2mqtt/IKEA in the palette.

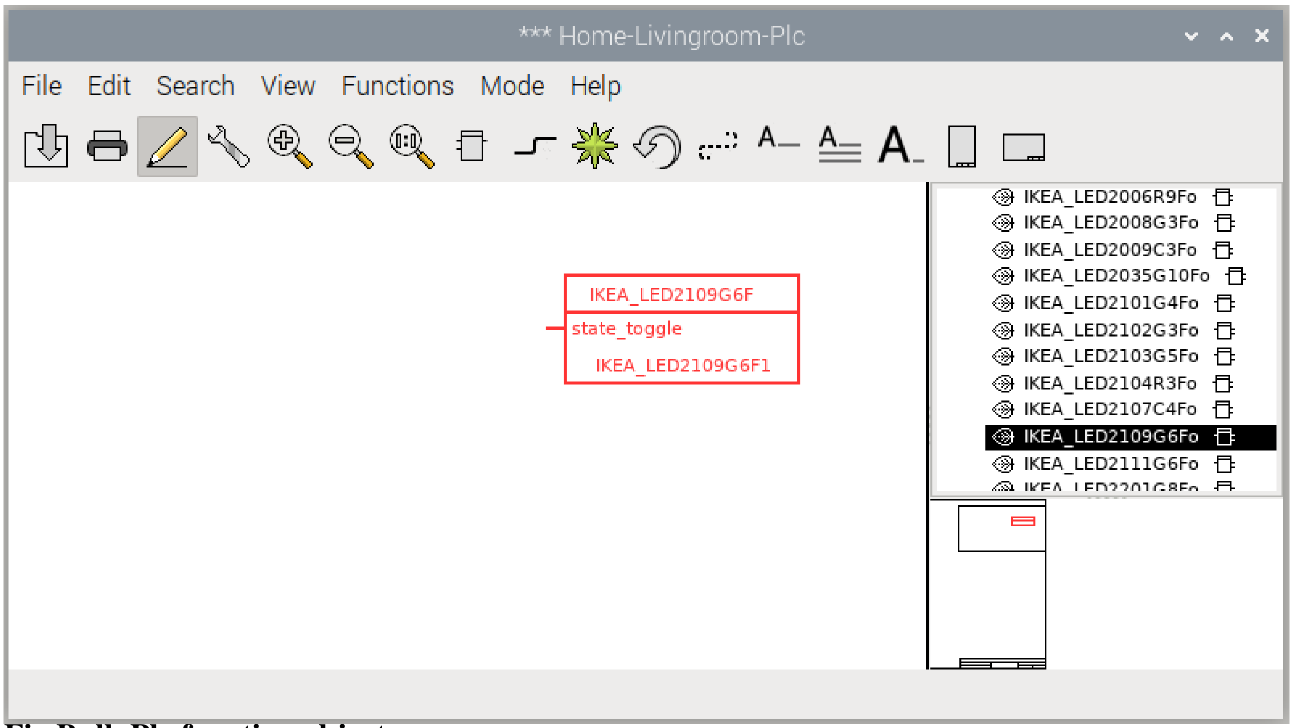
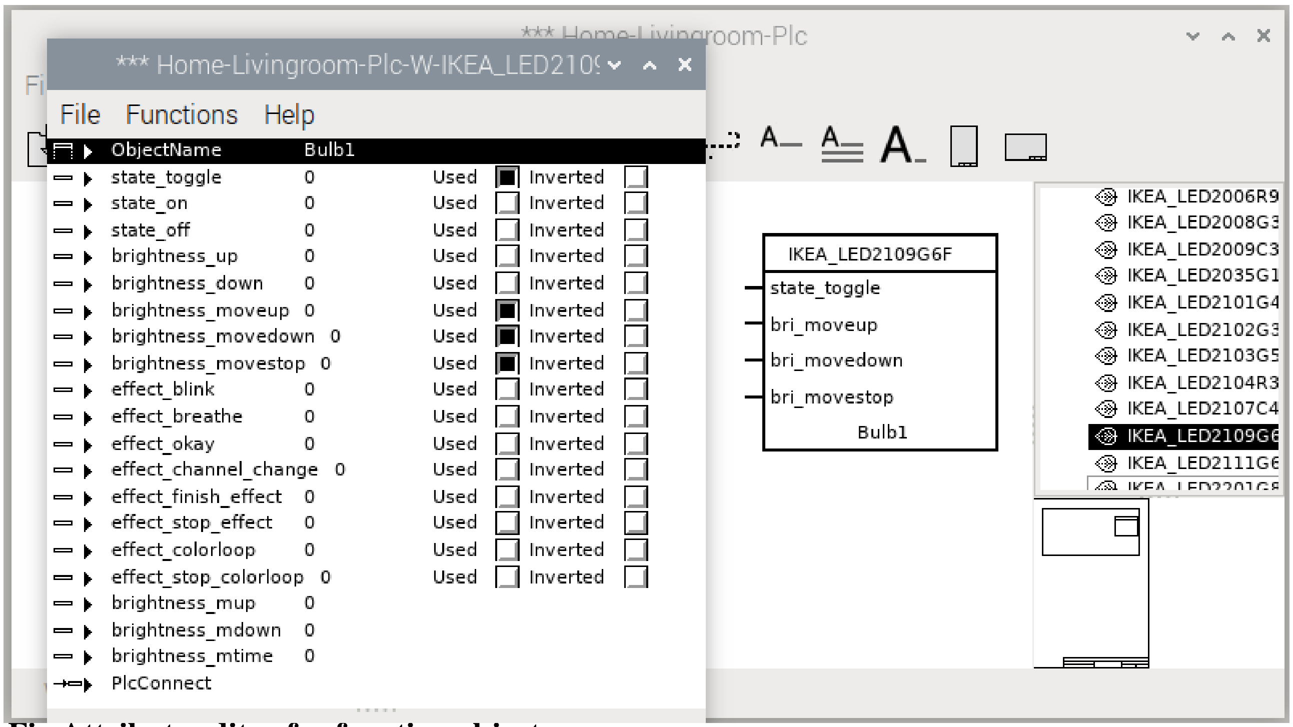


Fig Bulb Plc function object

Create the function object and open the Object Editor. Turning the bulb on or off is controlled by the State attribute, and by setting this to true or false, the bulb is turned on or off. By default the state_toggle input is visible, but we also want to control the brightness from the remote control. We check the Used boxes for brightness_moveup, brightness_movedown and brightness_movestop.



ObjectName	Bulb1				
state_toggle	0	Used	■	Inverted	☐
state_on	0	Used	☐	Inverted	☐
state_off	0	Used	☐	Inverted	☐
brightness_up	0	Used	☐	Inverted	☐
brightness_down	0	Used	☐	Inverted	☐
brightness_moveup	0	Used	■	Inverted	☐
brightness_movedown	0	Used	■	Inverted	☐
brightness_movestop	0	Used	■	Inverted	☐
effect_blink	0	Used	☐	Inverted	☐
effect_breathe	0	Used	☐	Inverted	☐
effect_okay	0	Used	☐	Inverted	☐
effect_channel_change	0	Used	☐	Inverted	☐
effect_finish_effect	0	Used	☐	Inverted	☐
effect_stop_effect	0	Used	☐	Inverted	☐
effect_colorloop	0	Used	☐	Inverted	☐
effect_stop_colorloop	0	Used	☐	Inverted	☐
brightness_mup	0				
brightness_mdown	0				
brightness_mtime	0				
PlcConnect					

Fig Attribute editor for function object

We also create the function object for the remote control, IKEA_E1524_E1819Fo, and enable the output pins action_brightness_up_hold, action_brightness_up_release, action_brightness_down_hold and and action_brightness_down_release. The figure below shows how to connect the function objects.

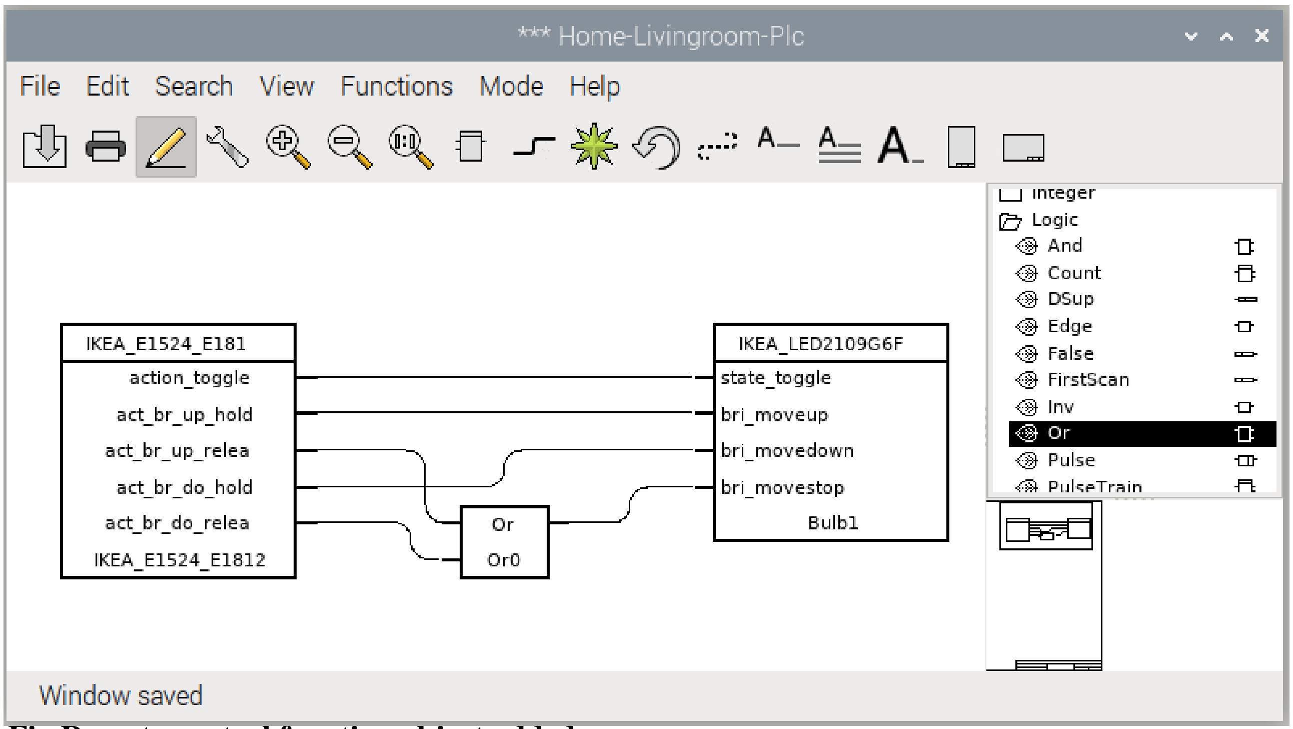
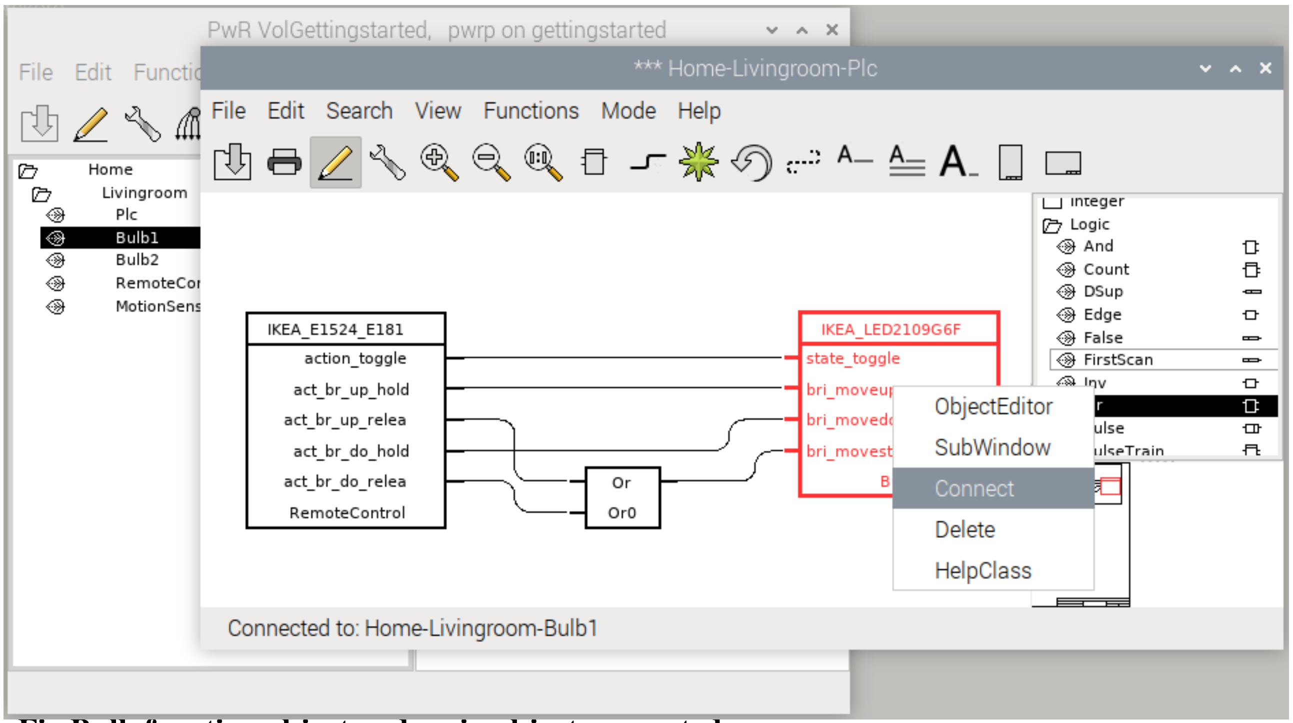


Fig Remote control function object added

The function object for a device has to be connected to the main object, and this is done by selecting the main object in the configurator, right clicking on the function object and activating Connect in the popup menu. This is done for both the bulb and the remote control.



Connected to: Home-Livingroom-Bulb1

Fig Bulb function object and main object connected

We also write the code for the second bulb that will be controlled by the motion sensor. The function object for the bulb, IKEA_LED1836G9Fo, is created. It’s more convenient to use the inputs state_on / state_off instead of state_toggle, thus state_on and state_off is checked in the Object editor, and the check for state_toggle is removed.

The motion sensor, IKEA_E2134, doesn’t have a function object, and we have to fetch the value of the occupancy Di signal with a GetDi instead. Create a GetDi from Signals/Digital/GetDi in the palette. Connect the occupancy Di by opening the main object for the motion sensor in the configurator and selecting the occupancy Di. Then right click on the GetDi and activate Connect in the popup menu. The occupancy is connected to the bulb as in the figure below, where the bulb is turned on at positive edge of occupancy, while the turn off is delayed 30 seconds after negative edge of occupancy.

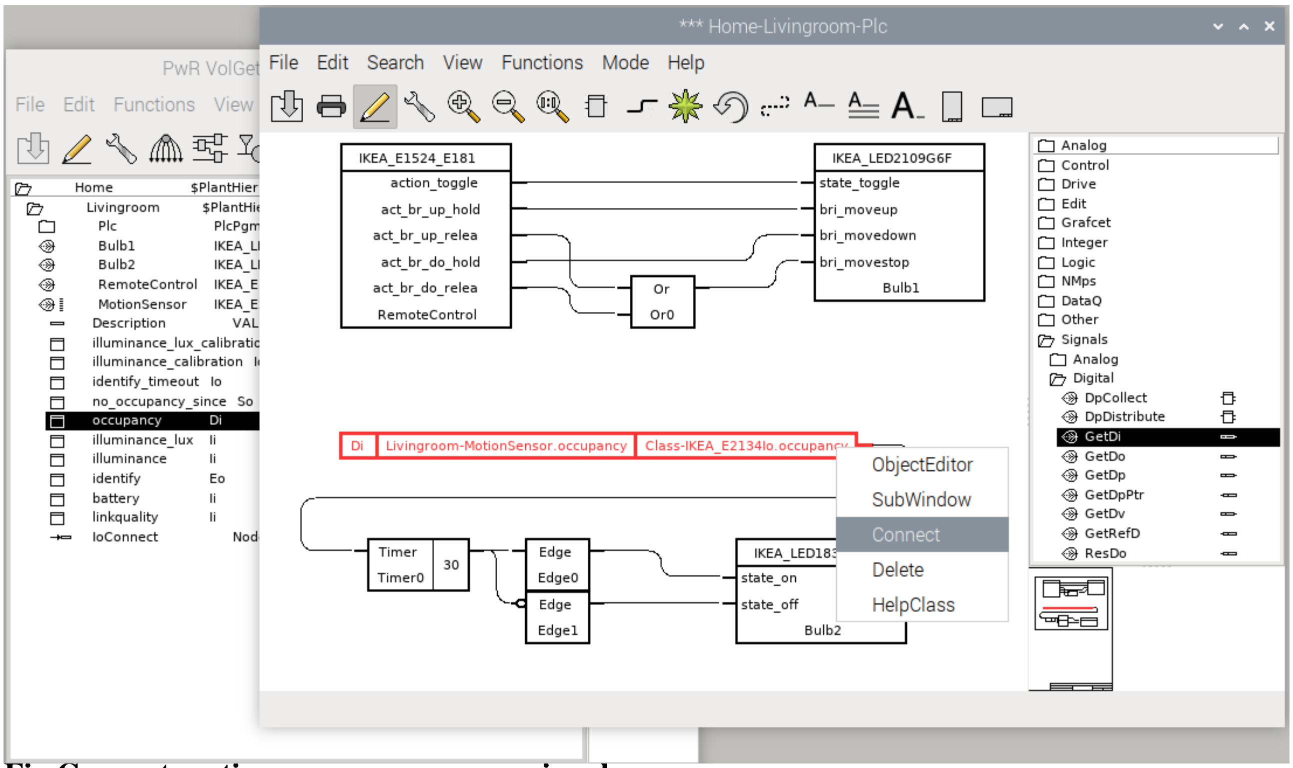


Fig Connect motion sensor occupancy signal

Enable Zigbee2mqtt

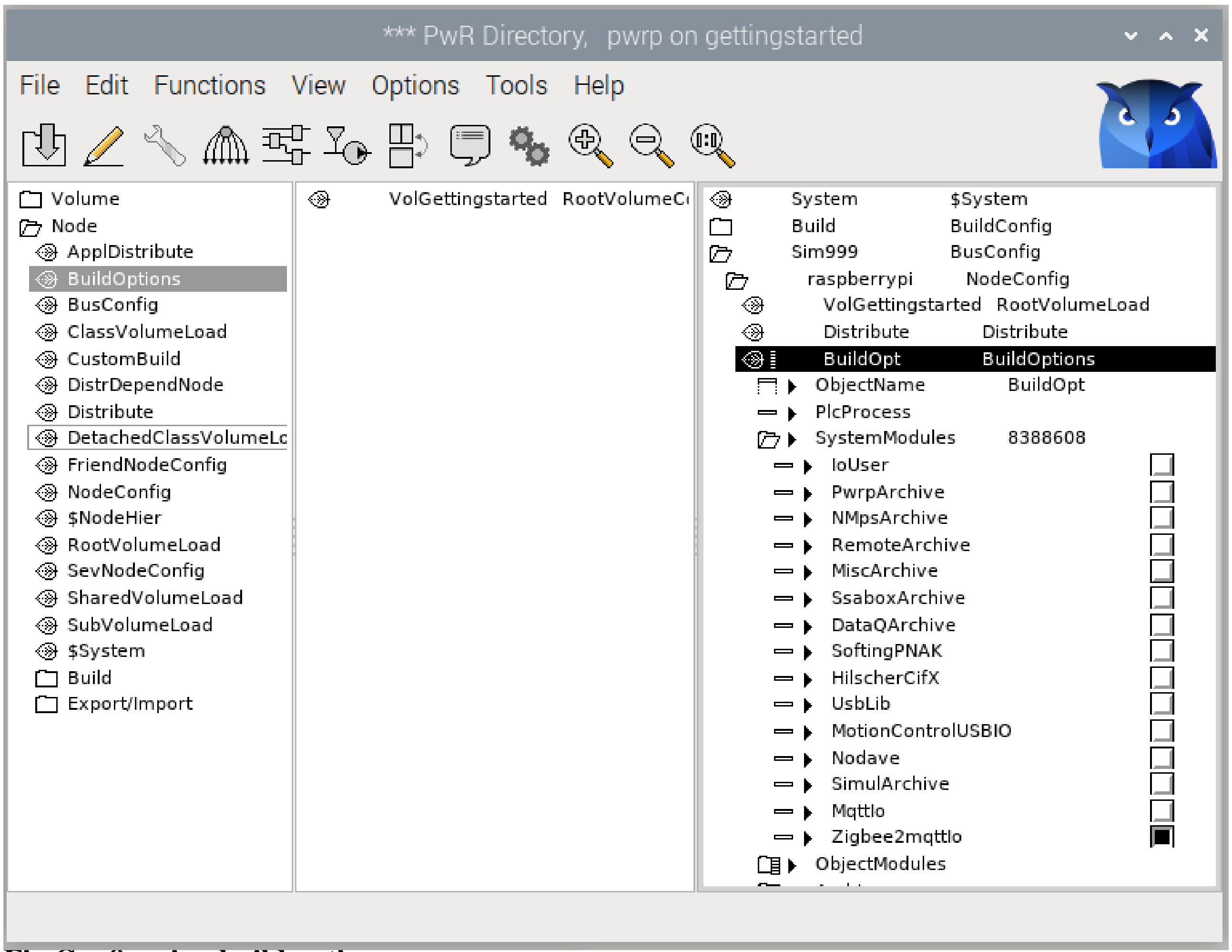
I you haven’t built the GettingStarted project before you have to build the node (see Getting started guide) with lots of error messages before you can enable the zigbee2mqtt module in the project. After the build, edit $pwrp_inc/ra_plc_user.h and insert the lines

```
#include "pwr_zigbee2mqttclasses.h"
#include "z2m_plc_zigbee2mqtt.h"
```

Note! In V6.1.4-1 instead insert

```
#include "pwr_zigbee2mqtt1classes.h"
#include "pwr_zigbee2mqtt2classes.h"
#include "pwr_zigbee2mqtt3classes.h"
#include "pwr_zigbee2mqtt4classes.h"
#include "pwr_zigbee2mqtt5classes.h"
#include "pwr_zigbee2mqtt6classes.h"
#include "pwr_zigbee2mqtt7classes.h"
#include "pwr_zigbee2mqtt8classes.h"
#include "pwr_zigbee2mqtt9classes.h"
#include "z2m_plc_zigbee2mqtt.h"
```

Open the directory volume and create a BuildOptions object under the NodeConfig object. Set PlcProcess to ‘plc’ and set Zigbee2mqtt in SystemModules.


*** PwR Directory,   pwrp on gettingstarted
File Edit Functions View Options Tools Help

```
Volume
Node
  ApplDistribute
  BuildOptions
  BusConfig
  ClassVolumeLoad
  CustomBuild
  DistrDependNode
  Distribute
  DetachedClassVolumeLc
  FriendNodeConfig
  NodeConfig
  $NodeHier
  RootVolumeLoad
  SevNodeConfig
  SharedVolumeLoad
  SubVolumeLoad
  $System
  Build
  Export/Import
```

```
VolGettingstarted  RootVolumeC
```

```
System                 $System
Build                  BuildConfig
Sim999                 BusConfig
  raspberrypi            NodeConfig
    VolGettingstarted    RootVolumeLoad
    Distribute           Distribute
    BuildOpt             BuildOptions
      ObjectName           BuildOpt
      PlcProcess
      SystemModules        8388608
        IoUser                                  ☐
        PwrpArchive                             ☐
        NMpsArchive                             ☐
        RemoteArchive                           ☐
        MiscArchive                             ☐
        SsaboxArchive                           ☐
        DataQArchive                            ☐
        SoftingPNAK                             ☐
        HilscherCifX                            ☐
        UsbLib                                  ☐
        MotionControlUSBIO                      ☐
        Nodave                                  ☐
        SimulArchive                            ☐
        MqttIo                                  ☐
        Zigbee2mqttIo                           ■
      ObjectModules
```

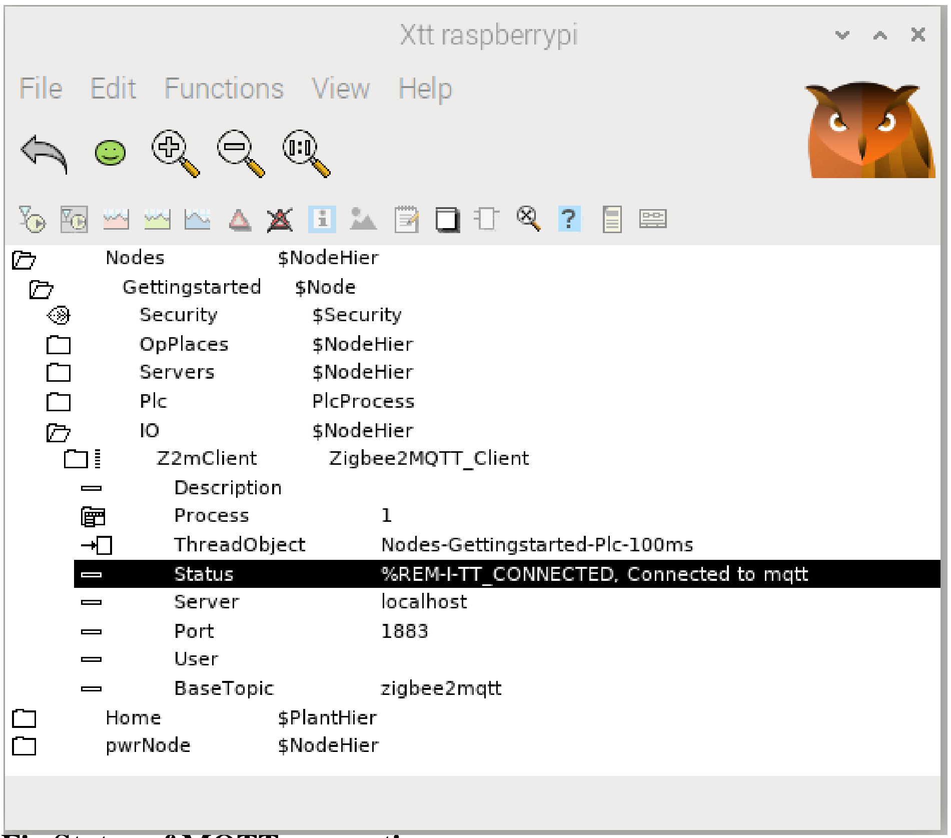
Fig Configuring build options

Now you should be able to build the node without error messages.

Runtime

Follow the Getting started guide to start ProviewR runtime and xtt. Make sure zigbee2mqtt is started also.

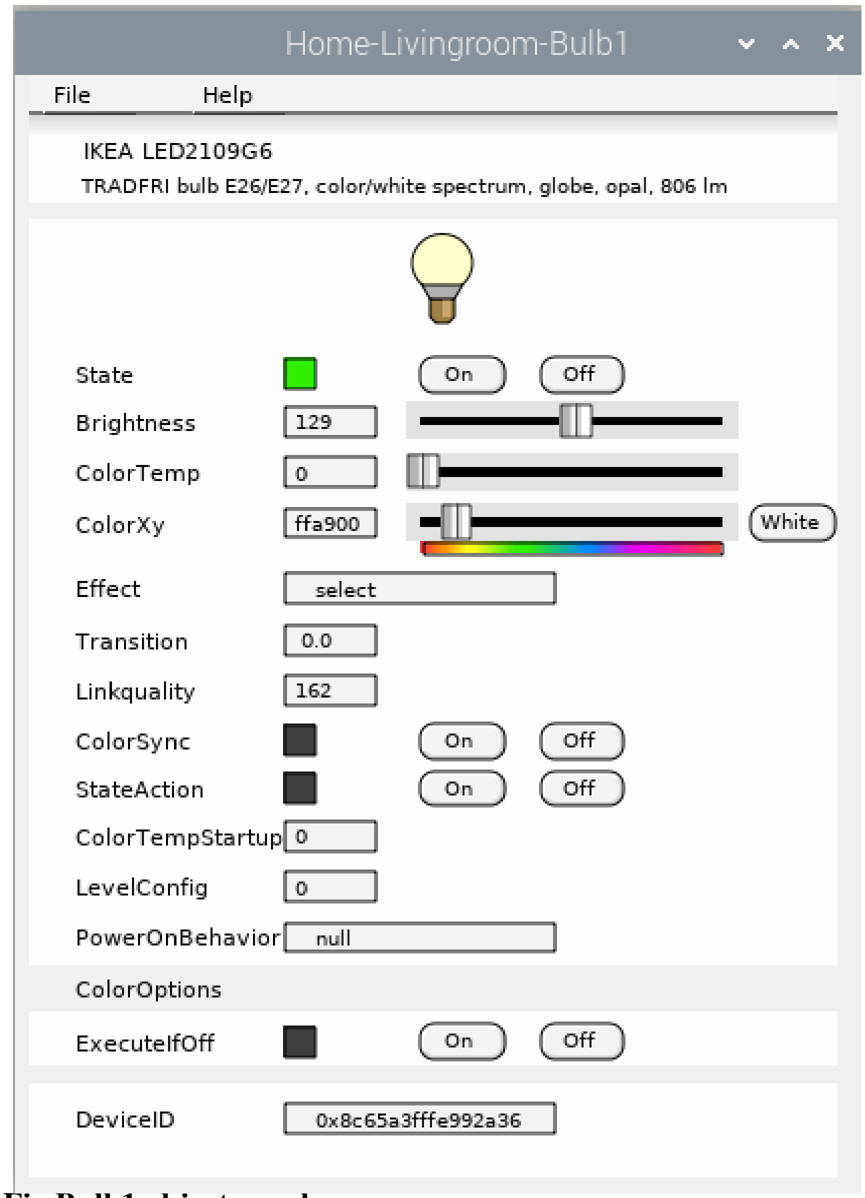
Open Nodes/Gettingstarted/IO/Z2mClient and check that the connection to the MQTT server is ok. It should be *%REM-I-TT_CONNECTED, Connected to mqtt.*



```
Nodes                $NodeHier
  Gettingstarted       $Node
    Security             $Security
    OpPlaces             $NodeHier
    Servers              $NodeHier
    Plc                  PlcProcess
    IO                   $NodeHier
      Z2mClient            Zigbee2MQTT_Client
        Description
        Process                  1
        ThreadObject             Nodes-Gettingstarted-Plc-100ms
        Status                   %REM-I-TT_CONNECTED, Connected to mqtt
        Server                   localhost
        Port                     1883
        User
        BaseTopic                zigbee2mqtt
Home                 $PlantHier
pwrNode              $NodeHier
```

Fig Status of MQTT connection

Now we can open the object graph for the color bulb, by selecting the main object and activate Object graph in the popup menu. The bulb can be switched on and of the the On/Off buttons, the color can be changed with the color slider, and the brightness with the brightness slider. You can also apply different effect from the Effect menu.



Home-Livingroom-Bulb1

File Help

IKEA LED2109G6
TRADFRI bulb E26/E27, color/white spectrum, globe, opal, 806 lm

State	🟩	On	Off
Brightness	129		
ColorTemp	0		
ColorXy	ffa900		White
Effect	select		
Transition	0.0		
Linkquality	162		
ColorSync	■	On	Off
StateAction	■	On	Off
ColorTempStartup	0		
LevelConfig	0		
PowerOnBehavior	null		

ColorOptions

ExecuteIfOff	■	On	Off

DeviceID 0x8c65a3fffe992a36

Fig Bulb1 object graph