WEINZIERL

KNX Stack for Linux

kTux

Manual

WEINZIERL ENGINEERING GmbH
Achatz 3-4
84508 Burgkirchen an der Alz
GERMANY

Tel.: +49 8677 / 916 36 – 0
E-Mail: info@weinzierl.de
Web: www.weinzierl.de

History / Change log

Description	Date	Author
Creation	2024-02-12	Gi

Content

1 Introduction 5

2 Medium TP 6
2.1 Requirements 6
2.2 Hardware Setup 7
2.3 Commissioning KNX network 8
2.4 Run the software 8
2.4.1 Prepare the Raspberry Pi 8
2.4.2 Test communication with Net'n Node 9
2.4.3 Commissioning KNX Stack with ETS 9
2.4.4 Test with demo application 9
2.5 Build the User Application with BAOS SDK 9
2.5.1 Update the System and install needed tools 10
2.5.2 Download and build the User Application and BAOS SDK 10
2.5.3 Start the User Application 10
2.5.4 Test, if User Application is working 10
2.6 Stop the KNX BAOS ObjectServer 10

3 Medium RF 11
3.1 Requirements 11
3.2 Hardware Setup 12
3.3 Commissioning KNX network 13
3.4 Run the software 13
3.4.1 Prepare the Raspberry Pi 13
3.4.2 Test communication with Net'n Node 13
3.4.3 Commissioning KNX Stack with ETS 14
3.4.4 Test with demo application 14
3.5 Build the User Application with BAOS SDK 14
3.5.1 Update the System and install needed tools 14
3.5.2 Download and build the User Application and BAOS SDK 15
3.5.3 Start the User Application 15
3.5.4 Test, if User Application is working 15
3.6 Stop the KNX BAOS ObjectServer 15

4 Medium IP 16
4.1 Requirements 16
4.2 Hardware Setup 17
4.3 Run the software 18
4.3.1 Prepare the Raspberry Pi 18
4.3.2 Test communication with Net'n Node 18
4.3.3 Commissioning KNX Stack with ETS 18
4.3.4 Test with demo application 19
4.4 Build the User Application with BAOS SDK 19
4.4.1 Update the System and install needed tools 19
4.4.2 Download and build the User Application and BAOS SDK 19

4.4.3 Start the User Application....19
4.4.4 Test, if User Application is working....19
4.5 Stop the KNX BAOS ObjectServer....20
5 Conclusions....21

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

1 Introduction

The purpose of this product is to connect a Linux device to the KNX network. The result is a full value KNX device which can be managed by ETS® commissioning software and can even be certified according the KNX Standard.

A binary executable file handles the management and communication of the KNX system completely by itself. An application can access the API of the KNX Stack via a TCP/IP socket. It uses the BAOS protocol, which is well established for many years.

For more informations about BAOS, visit the Weinzierl web page.

This product comes for three KNX media:

- TP (twisted pair)
- RF (radio frequency)
- IP (internet protocol)

This manual describes how to get this product running. The free version is provided “as is” for the RaspberryPi. If you need additional features, enhancements of the software or other hardware support, contact Weinzierl Engineering GmbH for a comercial solution.

2 Medium TP

2.1 Requirements

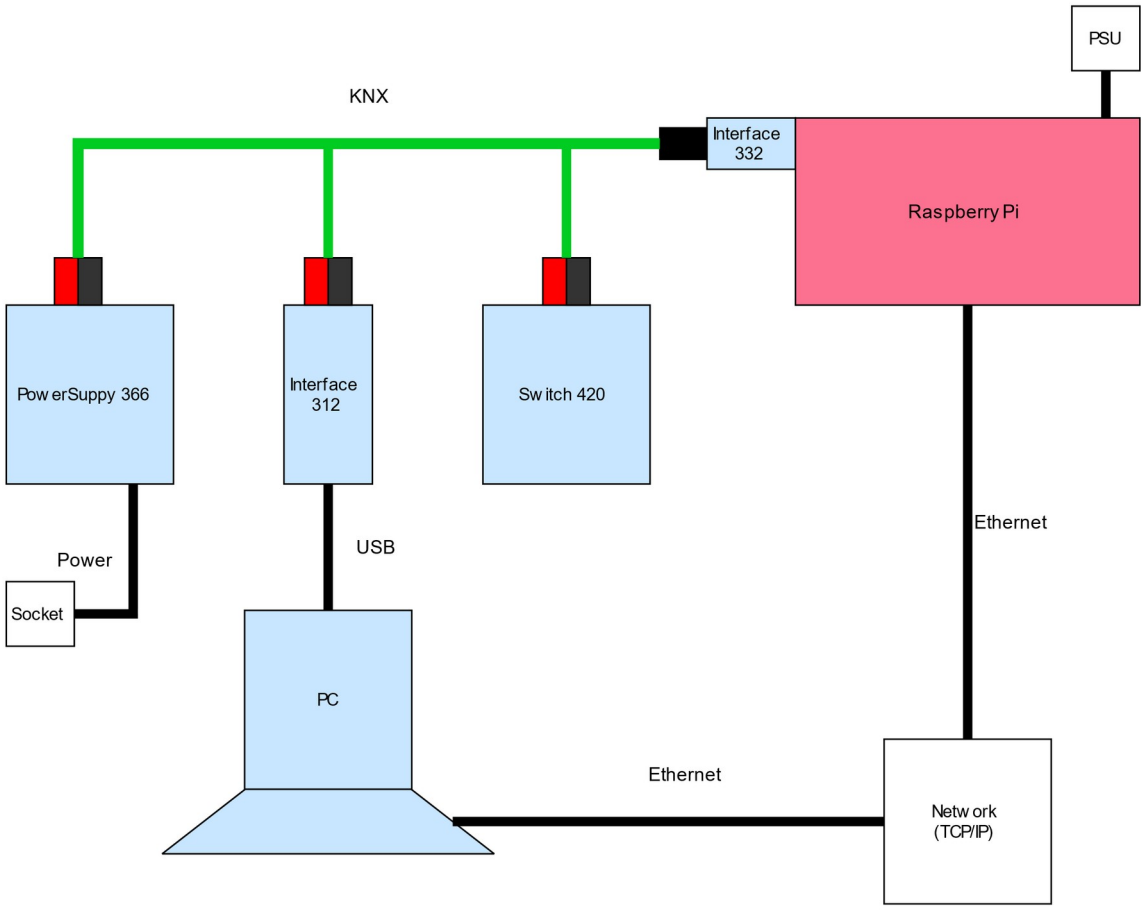
The following hardware components are required/recommended:

- Raspberry Pi with SD-Card (min. 2 GB) for Raspberry Pi OS
- Power supply for Raspberry Pi (PSU)
- KNX Power supply, e. g. KNX Power supply 366 from Weinzierl
- KNX TP USB Interface 332 from Weinzierl (no other brand)
- KNX TP USB Interface, e. g. KNX TP Interface 312 from Weinzierl
- KNX TP Push Button, e. g. KNX TP Push Button 420 secure from Weinzierl.
  As alternative Net'n Node can be used to send telegrams for testing.
- USB A to USB B cable for connecting PC to Interface 312

The following software is required:

- ETS software min. ETS5 for configuring the KNX devices
- ETS product entries (databases) for all used KNX devices
- Operating system Raspberry Pi OS (Lite version is sufficient)
- kTux TP (kTux-tp) with demo application (05_BaosIndications)
  Download BAOS-SDK-v2 at Weinzierl web page. The file is in "bin/samples".

2.2 Hardware Setup



- Connect all KNX TP devices to KNX bus.
- Connect the PC to the KNX Interface 312 (USB).
- Make sure the PC can access the Raspberry Pi via network.
- Plug in the KNX Interface 332 to a USB port at the Raspberry Pi.
- Power up the PowerSupply 366 and Raspberry Pi (PSU).

2.3 Commissioning KNX network

Download and start ETS5. A free demo version is available from knx.org.

Import the Project “kTux-tp.knxproj”.

In the Topology, we find the KNX devices

- KNX Power supply 366
- KNX USB Interface 312
- KNX TP Push Button Insert 420 secure
- kTux-tp for the kTux KNX Stack running on Raspberry Pi

Configure the **KNX Power supply** and the **KNX Push Button** with ETS. Activate the programming mode on one device after another and download the device configuration (physical address & application) with ETS.

2.4 Run the software

2.4.1 Prepare the Raspberry Pi

Download Raspberry Pi OS (Lite is sufficient) and store it on SD-Card and boot. For more informations about that, see https://www.raspberrypi.org/software/operating-systems.

Copy the files “kTux-tp” and “05_BaosIndications” to the home directory of your Raspberry Pi. E.g. via scp command available in Linux as well as in Windows 10:

```
scp kTux-tp 05_BaosIndications pi@raspberrypi:.
```

Enable access for the pi user to the KNX USB Interface 332:

```
sudo chmod 0666 /dev/usb/hiddev0
```

This command works if the KNX USB Interface is the first hid device connected to the Raspberry Pi. The command has to be executed each time you restart the Raspberry Pi.

Start the application in the background (use: &). Optionally, use the interface name for BAOS connection (defaults to the one connected to a gateway).

```
./kTux-tp interface eth0 &
```

Now the KNX BAOS ObjectServer is running. It is listening to the socket INET:12004 for BAOS communication.

Do not remove the KNX USB Interface while kTux is running. If so, kTux must be stopped and started again.

2.4.2 Test communication with Net'n Node

As a first step we try to connect to the KNX BAOS server running on the Raspberry Pi via IP. We use the BAOS protocol via IP while the KNX bus is connected via USB interface.

Start Net'n Node and scan KNX interfaces. You will find a "BAOS IP" with the IP Address of the Raspberry Pi. Connect to this interface and activate the BAOS View (Menu: BAOS/BAOS View). Push "Read" button and a list of server items will be shown.

Change to tab "Datapoints" and see that the list is empty because the device is not configured yet.

2.4.3 Commissioning KNX Stack with ETS

Use the BAOS View in Net'n Node to activate the programming mode by pressing sever item #15 "On". Note that now the KNX Stack running on the Raspberry Pi is in programming mode while the external USB stick just works as bus interface. The programming mode on the USB stick is not activated and both LEDs should be green.

Download the device configuration (physical address & application) with ETS. During the download both LEDs on the USB stick will show activity.

In Net'n Node, push the "Read" button in the BAOS View again and switch to the "Datapoints" tab. There is now one data point with its initial value 0.

Push the Switch (A0) of the KNX TP Push Button and the data point id 1 changes from 0 to 1, which is visible in Net'n Node's BAOS View/Datapoints. The button A1 switches back to 0.

Now your KNX Stack on Raspberry Pi is running and configured and is prepared to be used by an application.

2.4.4 Test with demo application

Start the demo application at the Raspberry Pi:

```
./05_BaosIndications <ip of the raspberry pi>
```

This application is a simple indications listener and shows changes of data points like this:

```
Received datapoint value indication for id 1: 01
```

Push the Switch (A0) and the data point id 1 changes from 0 to 1. This is also visible in Net'n Node's BAOS View/Datapoints.

End the demo application by pressing Enter.

Optionally disconnect Net'n Node from the BAOS IP interface.

2.5 Build the User Application with BAOS SDK

To build your own application, download the BAOS SDK and build it. This can be done at the Raspberry Pi or, if a cross development system is installed, on the PC. In this manual, we build on a Raspberry Pi.

2.5.1 Update the System and install needed tools

To use the BAOS SDK, we need CMake. Install it if necessary on the Raspberry Pi:

```
sudo apt update
sudo apt upgrade
sudo apt install cmake
```

2.5.2 Download and build the User Application and BAOS SDK

Download from Weinzierl web page the BAOS SDK, if not already done. Unpack it and compile the samples according to the README.md.

2.5.3 Start the User Application

After built, start the user application

```
cd build
./05_BaosIndications <ip of the raspberry pi>
```

2.5.4 Test, if User Application is working

Every time, we push the Switch (A0/A1), the user application gets an event:

```
Received datapoint value indication for id 1: 01
Received datapoint value indication for id 1: 00
```

To end the user application, press Enter.

2.6 Stop the KNX BAOS ObjectServer

To stop the KNX BAOS ObjectServer, send a CTRL-C or SIGHUP:

```
kill -HUP %1
```

Note: %1 addresses the first background process started within the current shell.

3 Medium RF

3.1 Requirements

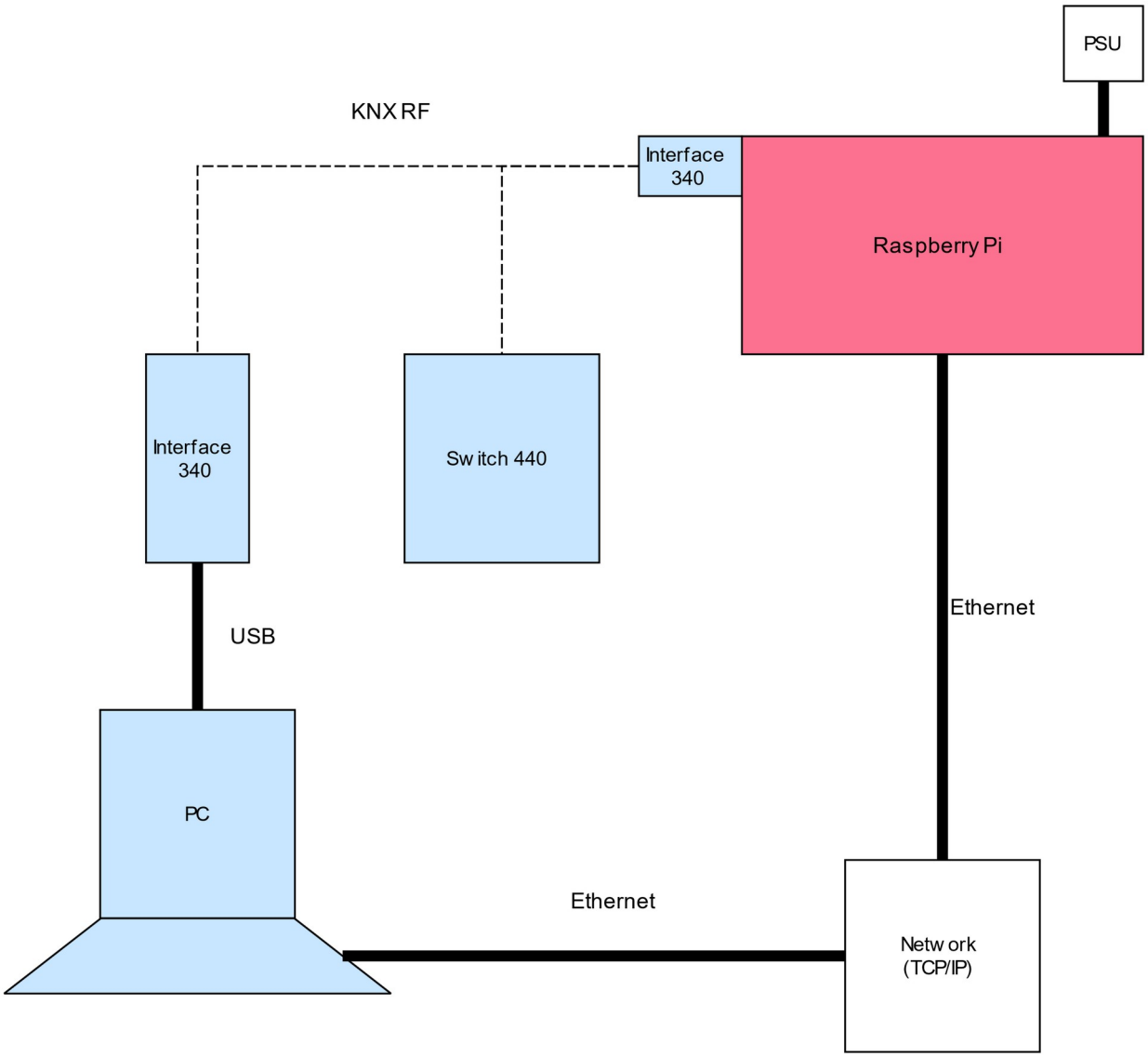
The following hardware components are required/recommended:

- Raspberry Pi with SD-Card (min. 2 GB) for Raspberry Pi OS
- Power supply for Raspberry Pi (PSU)
- KNX RF USB Interface 340 from Weinzierl (no other brand)
- KNX RF USB Interface, e. g. KNX RF USB Interface 340 from Weinzierl
- KNX RF Push Button, e. g. KNX RF Push Button 440 secure from Weinzierl.
  As alternative Net'n Node can be used to send telegrams for testing.
- USB A to USB B cable for connecting PC to Interface 340

The following software is required:

- ETS software min. ETS5 for configuring the KNX devices
- ETS product entries (databases) for all used KNX devices
- Operating system Raspberry Pi OS (Lite version is sufficient)
- kTux RF (kTux-rf) with demo application (05_BaosIndications)
  Download BAOS-SDK-v2 at Weinzierl web page. The file is in "bin/samples".

3.2 Hardware Setup



- Connect the PC to the KNX Interface 340 (USB).
- Make sure the PC can access the Raspberry Pi via network.
- Plug in the KNX Interface 340 to a USB port at the Raspberry Pi.
- Make sure the KNX Switch 440 is powered (battery).
- Power up the Raspberry Pi (PSU).

3.3 Commissioning KNX network

Download and start ETS6. A free demo version is available from knx.org.

Import the Project “kTux-rf.knxproj”.

In the Topology, we find the KNX devices

- KNX USB Interface 340
- KNX TP Push Button Insert 440 secure
- kTux-rf for the kTux KNX Stack running on Raspberry Pi

Configure the KNX Power supply and the KNX Push Button with ETS. Activate the programming mode on one device after another and download the device configuration (physical address & application) with ETS.

3.4 Run the software

3.4.1 Prepare the Raspberry Pi

Download Raspberry Pi OS (Lite is sufficient) and store it on SD-Card and boot. For more informations about that, see https://www.raspberrypi.org/software/operating-systems.

Copy the files “kTux-rf” and “05_BaosIndications” to the home directory of your Raspberry Pi. E.g. via scp command available in Linux as well as in Windows 10:

```
scp kTux-rf 05_BaosIndications pi@raspberrypi:.
```

Enable access for the pi user to the KNX USB Interface 332:

```
sudo chmod 0666 /dev/usb/hiddev0
```

This command works if the KNX USB Interface is the first hid device connected to the Raspberry Pi. The command has to be executed each time you restart the Raspberry Pi.

Start the application in the background (use: &). Optionally, use the interface name for BAOS connection (defaults to the one connected to a gateway)

```
./kTux-rf interface eth0 &
```

Now the KNX BAOS ObjectServer is running. It is listening to the socket INET:12004 for BAOS communication.

Do not remove the KNX USB Interface while kTux is running. If so, kTux must be stopped and started again.

3.4.2 Test communication with Net’n Node

As a first step we try to connect to the KNX BAOS server running on the Raspberry Pi via IP. We use the BAOS protocol via IP while the KNX bus is connected via USB interface.

Start Net'n Node and scan KNX interfaces. You will find a "BAOS IP" with the IP Address of the Raspberry Pi. Connect to this interface and activate the BAOS View (Menu: BAOS/BAOS View). Push "Read" button and a list of server items will be shown.

Change to tab "Datapoints" and see that the list is empty because the device is not configured yet.

3.4.3 Commissioning KNX Stack with ETS

Use the BAOS View in Net'n Node to activate the programming mode by pressing sever item #15 "On". Note that now the KNX Stack running on the Raspberry Pi is in programming mode while the external USB stick just works as bus interface. The programming mode on the USB stick is not activated and both LEDs should be green.

Download the device configuration (physical address & application) with ETS. During the download both LEDs on the USB stick will show activity.

In Net'n Node, push the "Read" button in the BAOS View again and switch to the "Datapoints" tab. There is now one data point with its initial value 0.

Push the Switch (A0) of the KNX TP Push Button and the data point id 1 changes from 0 to 1, which is visible in Net'n Node's BAOS View/Datapoints. The button A1 switches back to 0.

Now your KNX Stack on Raspberry Pi is running and configured and is prepared to be used by an application.

3.4.4 Test with demo application

Start the demo application at the Raspberry Pi:

```
./05_BaosIndications <ip of the raspberry pi>
```

This application is a simple event listener and shows changes of data points like this:

```
Received datapoint value indication for id 1: 01
```

Push the Switch (A0) and the data point id 1 changes from 0 to 1. This is also visible in Net'n Node's BAOS View/Datapoints.

End the demo application by pressing Enter.

Optionally disconnect Net'n Node from the BAOS IP interface.

3.5 Build the User Application with BAOS SDK

To build your own application, download the BAOS SDK and build it. This can be done at the Raspberry Pi or, if a cross development system is installed, on the PC. In this manual, we build on a Raspberry Pi.

3.5.1 Update the System and install needed tools

To use the BAOS SDK, we need CMake. Install it if necessary on the Raspberry Pi:

```
sudo apt update
sudo apt upgrade
```

```
sudo apt install cmake
```

3.5.2 Download and build the User Application and BAOS SDK

Download from Weinzierl web page the BAOS SDK, if not already done. Unpack it and compile the samples according to the README.md.

3.5.3 Start the User Application

After built, start the user application

```
cd build
./05_BaosIndications <ip of the raspberry pi>
```

3.5.4 Test, if User Application is working

Every time, we push the Switch (A0/A1), the user application gets an event:

```
Received datapoint value indication for id 1: 01
Received datapoint value indication for id 1: 00
```

To end the user application, press Enter.

3.6 Stop the KNX BAOS ObjectServer

To stop the KNX BAOS ObjectServer, send a CTRL-C or SIGHUP:

```
kill -HUP %1
```

Note: %1 addresses the first background process started within the current shell.

4 Medium IP

4.1 Requirements

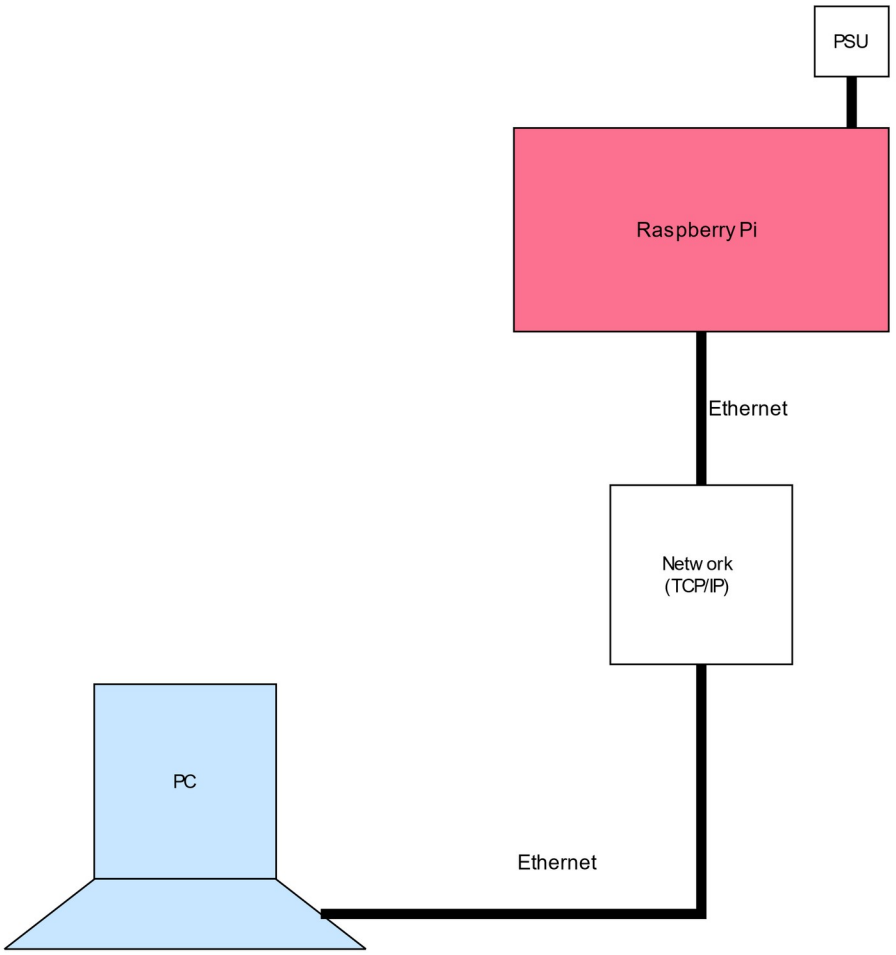
The following hardware components are required/recommended:

- Raspberry Pi with SD-Card (min. 2 GB) for Raspberry Pi OS
- Power supply for Raspberry Pi (PSU)
- An IP networking infrastructure (e.g. Router)
- Net’n Node to send telegrams for testing
- IP network cables for Raspberry Pi

The following software is required:

- ETS software min. ETS5 for configuring the KNX devices
- ETS product entries (databases) for all used KNX devices
- Operating system Raspberry Pi OS (Lite version is sufficient)
- kTux IP (kTux-ip) with demo application (05_BaosIndications)
  Download BAOS-SDK-v2 at Weinzierl web page. The file is in “bin/samples”.

4.2 Hardware Setup



- Make sure the PC can access the Raspberry Pi via network.
- Power up Raspberry Pi.

4.3 Run the software

4.3.1 Prepare the Raspberry Pi

Download Raspberry Pi OS (Lite is sufficient) and store it on SD-Card and boot. For more informations about that, see https://www.raspberrypi.org/software/operating-systems.

Copy the files “kTux-ip” and “05_BaosIndications” to the home directory of your Raspberry Pi. E.g. via scp command available in Linux as well as in Windows 10:

```
scp kTux-ip 05_BaosIndications pi@raspberrypi:.
```

Start the application in the background (use: &). Optionally, use the interface name for BAOS connection (defaults to the one connected to a gateway)

```
./kTux-ip interface eth0 &
```

Now the KNX BAOS ObjectServer is running. It is listening to the socket INET:12004 for BAOS communication.

4.3.2 Test communication with Net’n Node

As a first step we try to connect to the KNX BAOS server running on the Raspberry Pi via IP. We use the BAOS protocol via IP while the KNX bus is connected via USB interface.

Start Net’n Node and scan KNX interfaces. You will find a “BAOS IP” with the IP Address of the Raspberry Pi. Connect to this interface and activate the BAOS View (Menu: BAOS/BAOS View). Push “Read” button and a list of server items will be shown.

Change to tab “Datapoints” and see that the list is empty because the device is not configured yet.

4.3.3 Commissioning KNX Stack with ETS

Download and start ETS5. A free demo version is available from knx.org.

Import the Project “kTux-ip.knxproj”.

In the Topology, we find the KNX devices

- kTux-ip for the kTux for the KNX Stack running on Raspberry Pi

Use the BAOS View in Net’n Node to activate the programming mode by pressing sever item #15 “On”.

In Net’n Node, push the “Read” button in the BAOS View again and switch to the “Datapoints” tab. There is now one data point with its initial value 0.

Now your KNX Stack on Raspberry Pi is running and configured and is prepared to be used by an application.

4.3.4 Test with demo application

Start the demo application at the Raspberry Pi:

```
./05_BaosIndications <ip of the raspberry pi>
```

This application is a simple event listener and shows changes of data points like this:

```
Received datapoint value indication for id 1: 01
```

Use ETS to send a telegram: Diagnosis: Start Group Monitor. Enter Group Address 1/3/65. Change the value to “On” and push the “Write” button. The data point id 1 changes from 0 to 1. This is also visible in Net’n Node’s BAOS View/Datapoints

End the demo application by pressing Enter.

Optionally disconnect Net’n Node from the BAOS IP interface.

4.4 Build the User Application with BAOS SDK

To build your own application, download the BAOS SDK and build it. This can be done at the Raspberry Pi or, if a cross development system is installed, on the PC. In this manual, we build on a Raspberry Pi.

4.4.1 Update the System and install needed tools

To use the BAOS SDK, we need CMake. Install it if necessary on the Raspberry Pi:

```
sudo apt update
sudo apt upgrade
sudo apt install cmake
```

4.4.2 Download and build the User Application and BAOS SDK

Download from Weinzierl web page the BAOS SDK, if not already done. Unpack it and compile the samples according to the README.md.

4.4.3 Start the User Application

After built, start the user application

```
cd build
./05_BaosIndications <ip of the raspberry pi>
```

4.4.4 Test, if User Application is working

Every time, we push the Switch (A0/A1), the user application gets an event:

```
Received datapoint value indication for id 1: 01
Received datapoint value indication for id 1: 00
```

To end the user application, press Enter.

4.5 Stop the KNX BAOS ObjectServer

To stop the KNX BAOS ObjectServer, send a CTRL-C or SIGHUP:

```
kill -HUP %1
```

Note: %1 addresses the first background process started within the current shell.

5 Conclusions

This simple sample shows how basically the KNX BAOS works. The BAOS SDK can be used to develop more sophisticated applications. The base is the BAOS protocol. See https://www.weinzierl.de for more information.

The user application can also be developed on another host (PC). This application can connect the KNX BAOS ObjectServer remotely via the IP-Socket on port 12004. This port can server up to 4 connections.

We recommend your tool Net'n Node to monitor, manipulate the KNX Bus. This Tool can also connect to the KNX BAOS ObjectServer and is a great help to get used to the BAOS protocol.

WEINZIERL ENGINEERING GmbH
Achatz 3-4
84508 Burgkirchen an der Alz
GERMANY

Tel.: +49 8677 / 916 36 - 0
E-Mail: info@weinzierl.de
Web: www.weinzierl.de

2024-02-12