Technology
Arts Sciences
TH Köln

Faculty of
Information, Media and
Electrical Engineering

Institute of
Communication Systems

Software

RhizoTech project documentation

Written by: Daniel Lohmann and Kai Lübeck

In cooperation with:

Universität zu Köln

CEPLAS
Cluster of Excellence on Plant Sciences

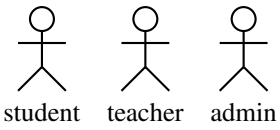
Sponsored by:

RheinEnergie Stiftung Jugend | Beruf Wissenschaft

Contents

1 Requirements

1.1 Users



1.2 Use Cases

2 General overview

A general system overview can be seen in figure **??**.

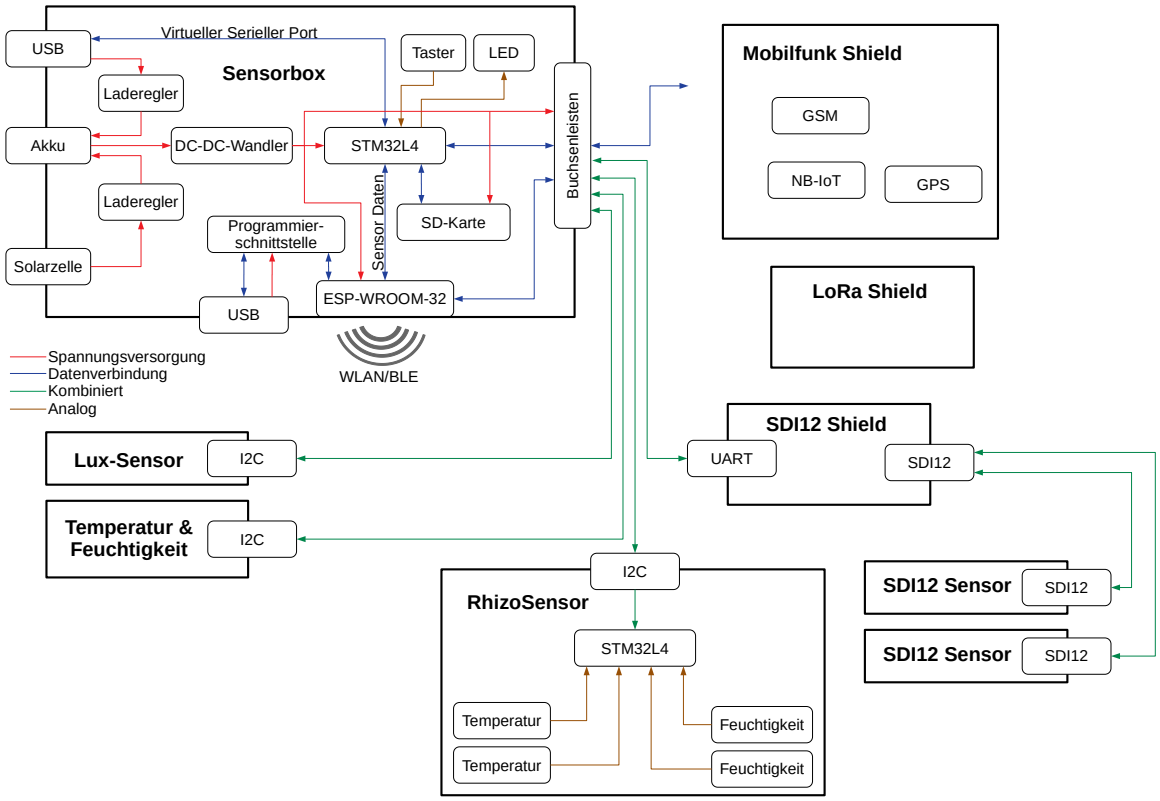


Figure 2.1: System structure

An overview of the communication channels can be found in figure **??**.

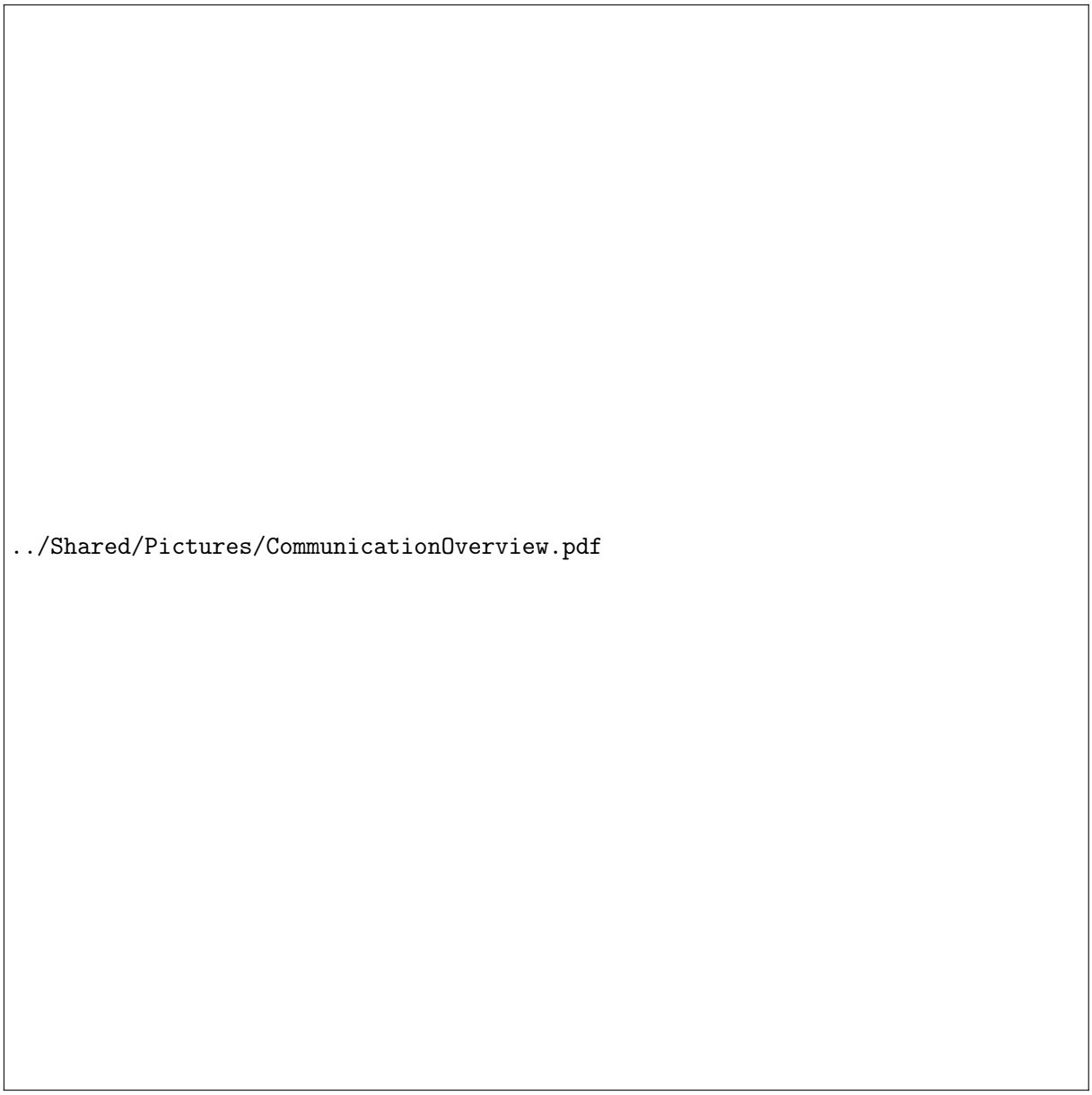


Figure 2.2: Communication overview

3 Development environment

3.1 Software Tools

The development was done under the newest versions of Ubuntu during the project. The latest used version is: 20.04. The following development software is needed for the development of this project:

- VSCode (Visual Studio Code) - Manual Install
- System Workbench for STM32 - Manual Install
- STM32CubeMX - Manual Install
- STM32CubeProgrammer - Manual Install
- ESP-IDF - Manual Install
- Inkscape
- Dia
- TexLive
- Valgrind
- Python3
- KiCad
- LibreOffice
- gcc
- make
- Doxygen
- Graphviz
- LibreCAD
- Git
- Mosquitto
- SQLiteBrowser
- MQTTfx - Manual Install
- Powershell Core
- OpenSSL
- Go

The most tools can be installed with the following commands

```
sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
sudo apt-get update
sudo apt install inkscape dia texlive-full latexmk valgrind
    ↪ python3 python3-pip build-essential doxygen graphviz gimp
    ↪ librecad git mosquitto sqlitebrowser tar zip unzip inotify-
    ↪ tools openssl fswebcam libssl-dev libwebsockets8
    ↪ libwebsockets-dev libc-ares2 libc-ares-dev uuid uuid-dev
    ↪ golang
sudo snap install powershell --classic
```

The required python packages can be install with, while being in the *Webservice/install* directory:

```
pip3 install -r requirements.txt
pip3 install pytest
pip3 install pylint
```

For Visual Studio Code exists a Workspace definition *RhizoTech.code-workspace* which can be opened via *File->Open Workspace*. After opening the workspace you will be asked to about workspace recommendations. All the recommended extensions should be installed.

If the packets should be installed with an activated conda enviroment (conda activate <EnvName>), the required python packages can be install with the following commands, while being in the *Webservice/install* directory:

```
conda install --file requirements_conda.txt
```

3.1.1 Install Software tools on MacOS systems

To install the software tools on a MacOS system the open-source software package management system homebrew have to be installed on the system. To install homebrew execute the following command in the terminal.

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/
    ↪ Homebrew/install/master/install.sh)"
```

With the following brew commands the necessary software tools would be installed.

```
brew install --cask visual-studio-code
brew install --cask powershell
brew install --cask inkscape
brew install --cask libreoffice
brew install --cask db-browser-for-sqlite
brew install --cask mactex-no-gui
brew install gcc
brew install --cask doxygen
brew install --cask miniconda
brew install graphviz
brew install --cask librecad
brew install mosquitto
brew install --cask mqttfx
```

3.2 Flashing of MCUs

3.2.1 STM32L4 micro controller

The first step is to Flash the Motherboard STM32L4 micro controller. On the first flashing of the micro controller you need to execute some additional steps.

1. Open Visual Studio Code Workspace
2. Press *Crtl+P*
3. Type *task Erase Flash*
4. Press *Enter*
5. Wait for completion of the command which should be shown at the bottom in the terminal. This command will erase the hole micro controllers flash memory.
6. Press *Ctrl+P*
7. Type *task Disable IWDG during standby*
8. Press *Enter*

9. Wait for completion of the command which should be shown at the bottom in the terminal. This command will set the option byte, which disables the **IWDG!** (**IWDG!**) in the standby mode.

After these steps use *System Workbench for STM32* for the final flashing of the micro controller and the development of the STM32L4.

3.2.2 ESP32 micro controller

The ESP32 micro controller is flash via the UART to USB adapter on the board. To flash the ESP micro controller the simples way is to configured all intervals to the same value and enable **WLAN!** (**WLAN!**) sending. The a measurement can be started and then the command the following command can be executed:

```
make flash
```

Which will then wait until the ESP is started for sending data via **WLAN!** and then the Flash process will start. The command needs to executed in the *Motherboard/Motherboard_ESP* directory. Maybe prior to the above command the device name needs to be set. This can be done via the command:

```
make menuconfig
```

After executing this command a graphical user interface will be shown in the command line which can be navigated with the keyboard. In the menu *Serial flasher config* is the setting *Default serial port*, which needs to set according to your specific value.

3.3 Simulated sensor data transmission

The C/C++ project `Motherboard_Testing` allows to construct messages which can be transmitted to the MQTT-Broker. The script `Motherboard/Motherboard_Testing/SendScript.py` executes the compiled version of the `Motherboard_Testing` software. The script `SendScript.py` can be configured to send messages to a local broker without Username, Password and TLS encryption or it can send messages which will be stored in the Webservice. The C/C++ project `Motherboard_Testing` can be compiled with *System Workbench for STM32*.

Commandline arguments of Motherboard_Testing:

```
Motherboard_Testing [--message
       [--serialnumber <high> <mid> <low>]
       [--airtemp <value>]
       [--airhumidity <value>]
       [--soiltemp <value>]
       [--soilmoisture <value>]
       [--illuminance <value>]
       [--battery <value>]
       [--pH <value>]
       [--dateNow]
       [--test]
```

- `-message`
  Generate a message as output and do not execute tests.
- `-serialnumber <high> <mid> <low>`
  Serial number split into three 32-bit numbers which should be the source of the message.

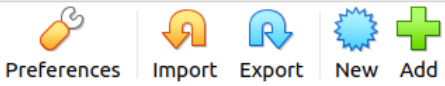
- `-airtemp <value>`
  Add an air temperature value to the message. This parameter can be specified multiple times.
- `-airhumidity <value>`
  Add an air humidity value to the message. This parameter can be specified multiple times.
- `-soiltemp <value>`
  Add a soil temperature value to the message. This parameter can be specified multiple times.
- `-soilmoisture <value>`
  Add a soil moisture value to the message. This parameter can be specified multiple times.
- `-illuminance <value>`
  Add an illuminance value to the message. This parameter can be specified multiple times.
- `-battery <value>`
  Add a battery voltage to the message. This parameter can be specified multiple times.
- `-ph <value>`
  Add a pH value voltage to the message. This parameter can be specified multiple times.
- `-datenow`
  The timestamp in the message should be the time of execution.
- `-test`
  Use a randomly generated message.

The order of the parameters which can be specified multiple times, is the order in which they will occur in the message. This is than also the oder of the sensors in the webservice.

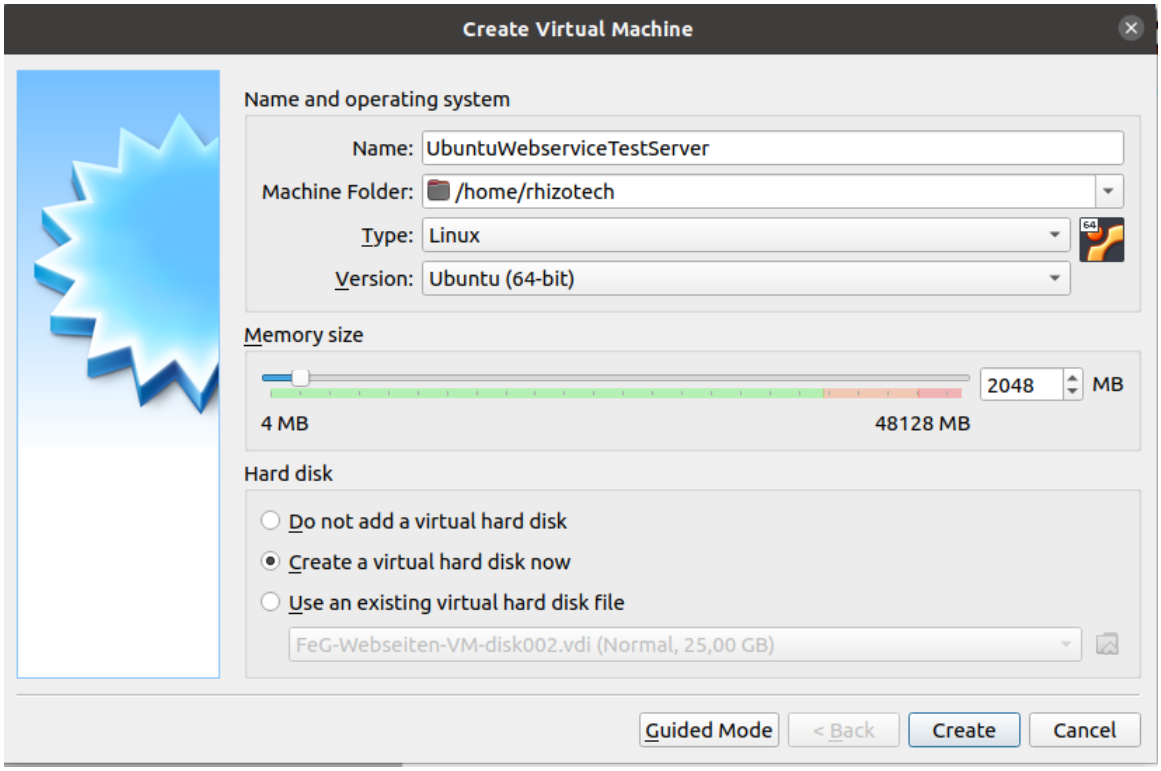
3.4 Webservice Real-Deployment Testing

The webservice in a real deployment is different from a local deployment. Normally locally only the Python web application is executed and on the real server also the mqtt broker is instantiated. How a virtual machine can be setup for testing the installation scripts of the webservice is outline below. The installation of the webservice is covered in a separate document about the webserver. In this document all necessary steps for an initial deployment of the webservice are outlined.

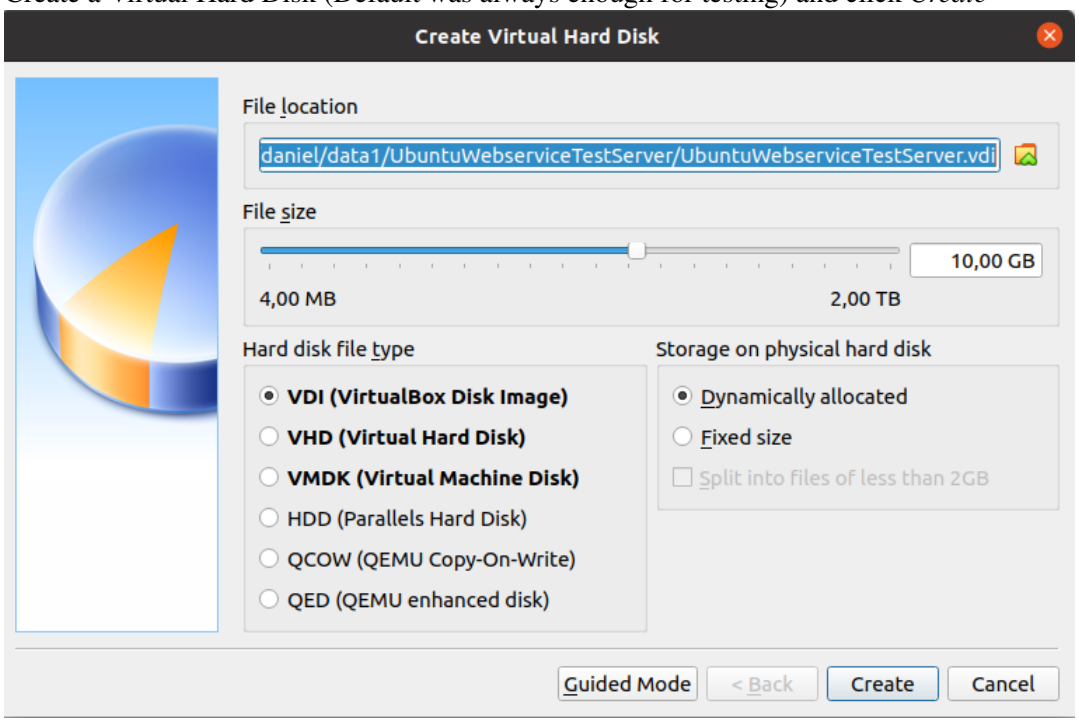
1. Install Virtual Box
2. Download new Ubuntu Server image
3. Create new Virtual Box Virtual Machine with *New* button



4. Set a *Name*, *Type* and *Version* of operating system. If possible give 2 GB. Then click *Create*

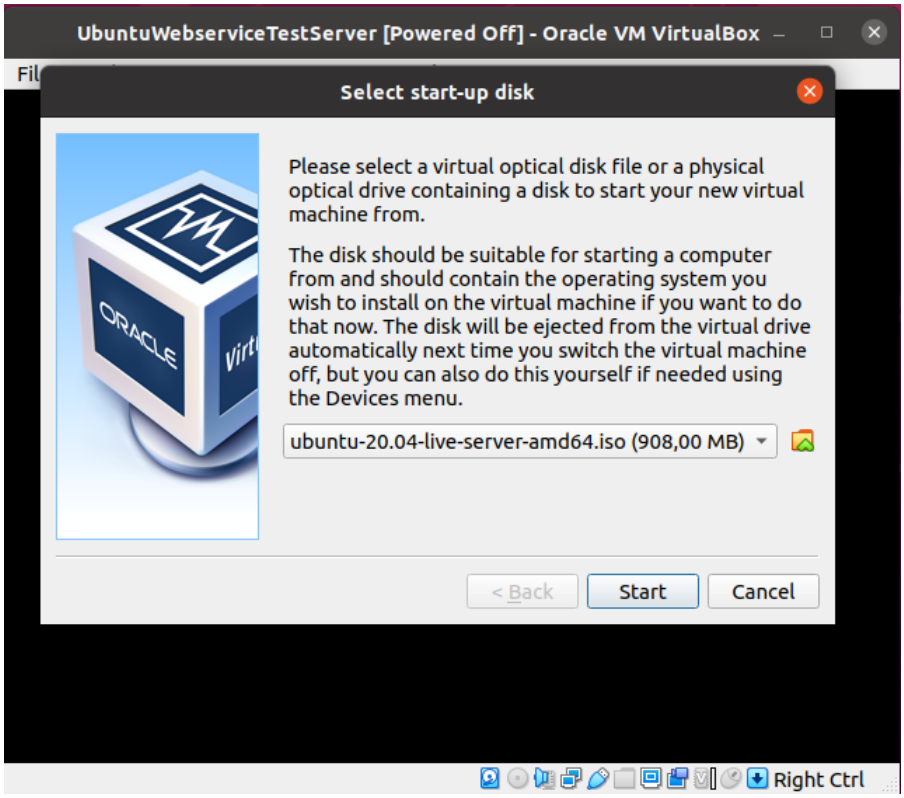


5. Create a Virtual Hard Disk (Default was always enough for testing) and click *Create*

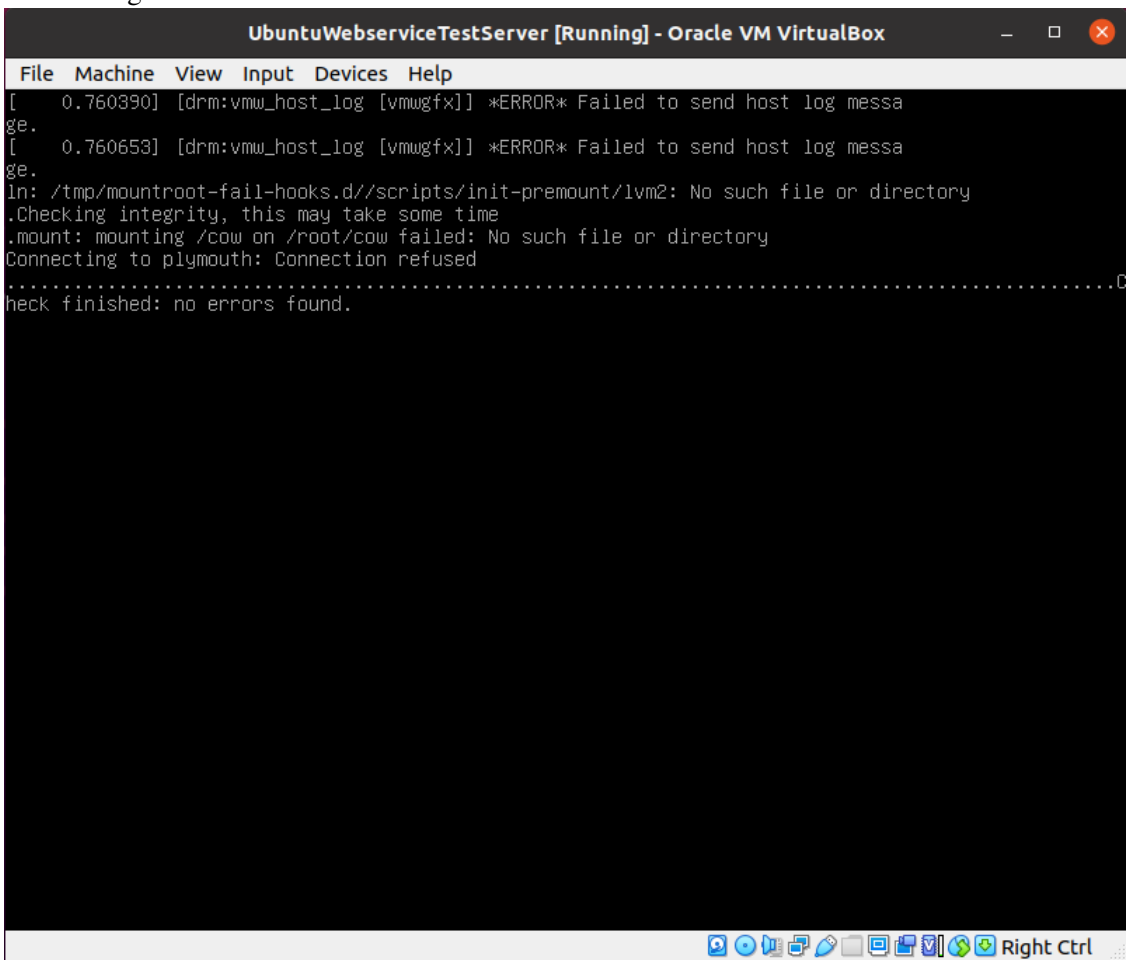


6. Double Click the created Virtual Machine

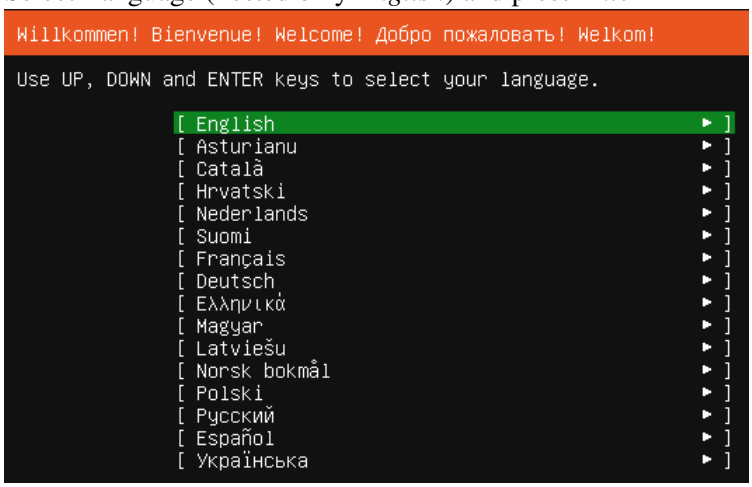
7. Select the Virtual Disk Image of Ubuntu which was previously download. Then click *Start*



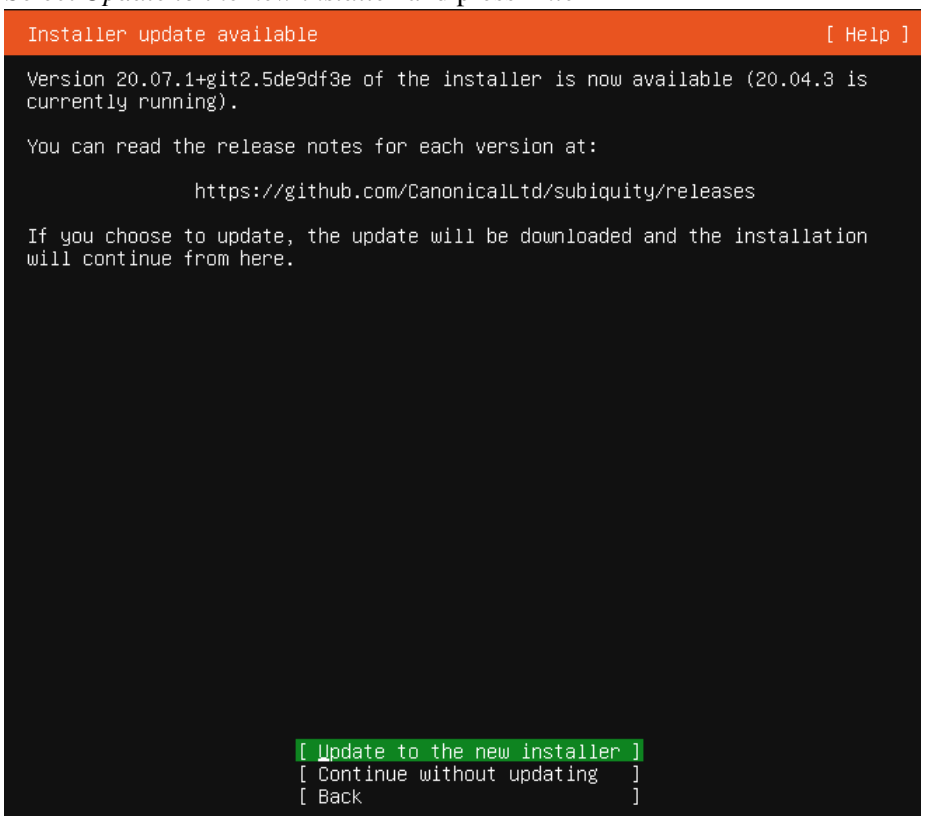
8. Wait during boot...



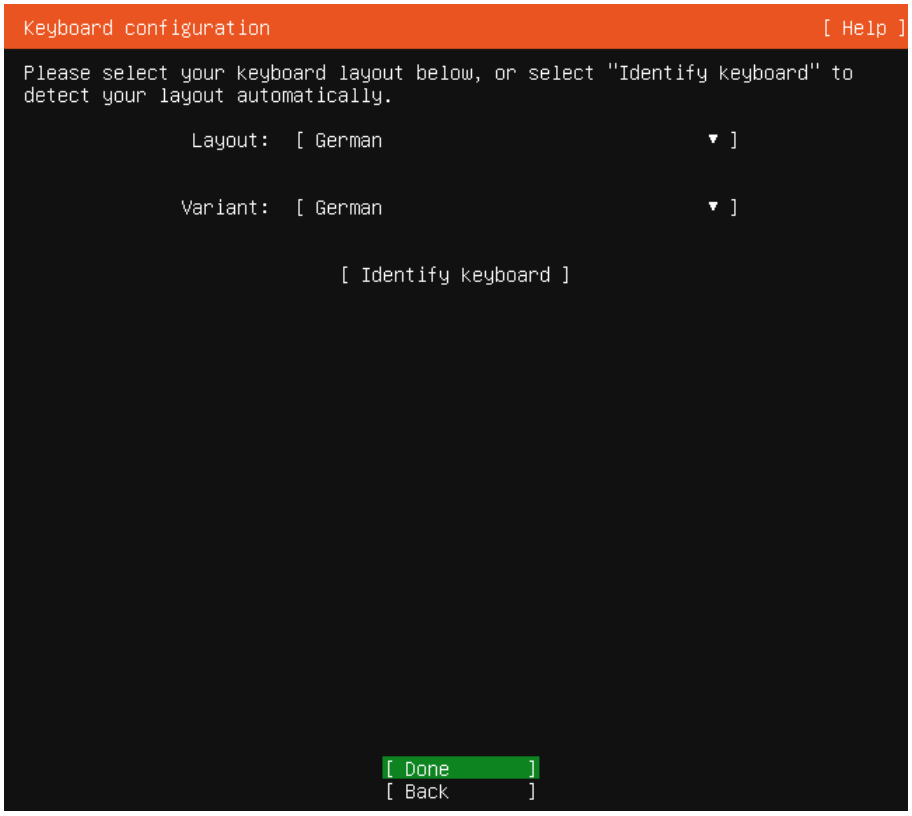
9. Select Language (Tested only *English*) and press *Enter*



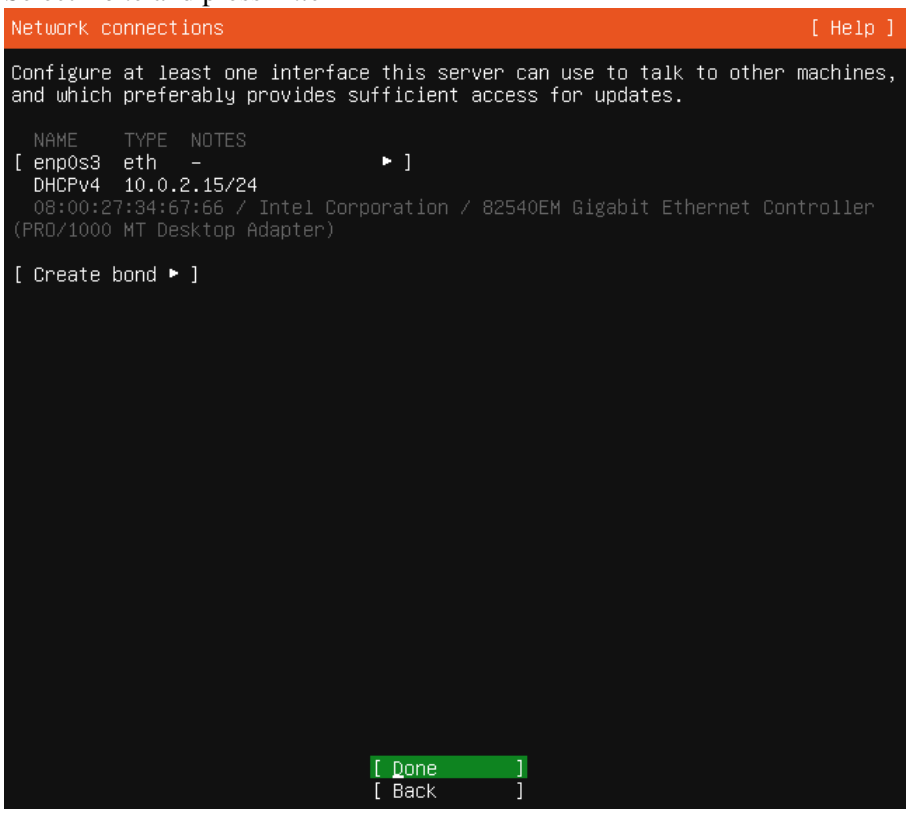
10. Select *Update to the new installer* and press *Enter*



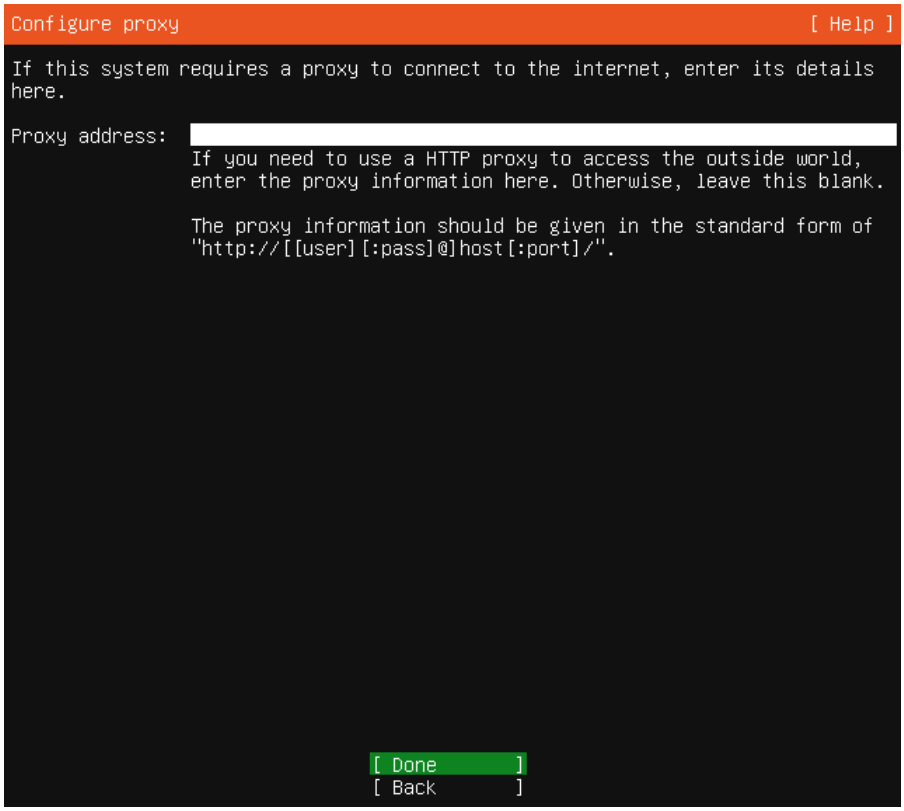
11. Select *Keyboard Layout* and *Keyboard Variant*. Then select *Done* and press *Enter*



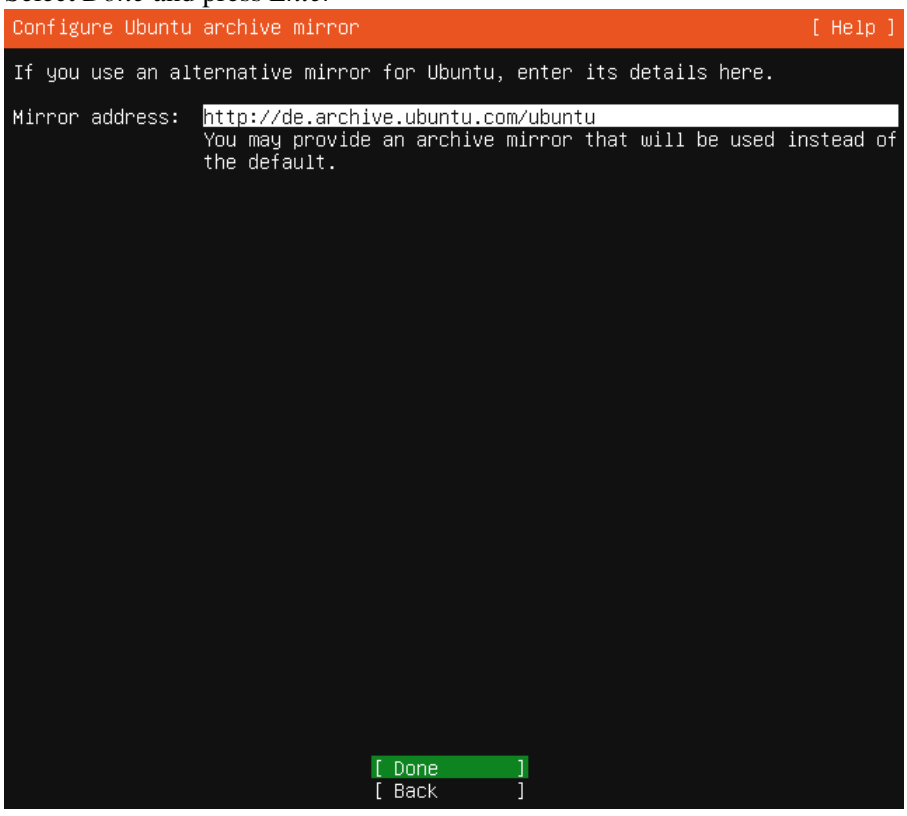
12. Select *Done* and press *Enter*



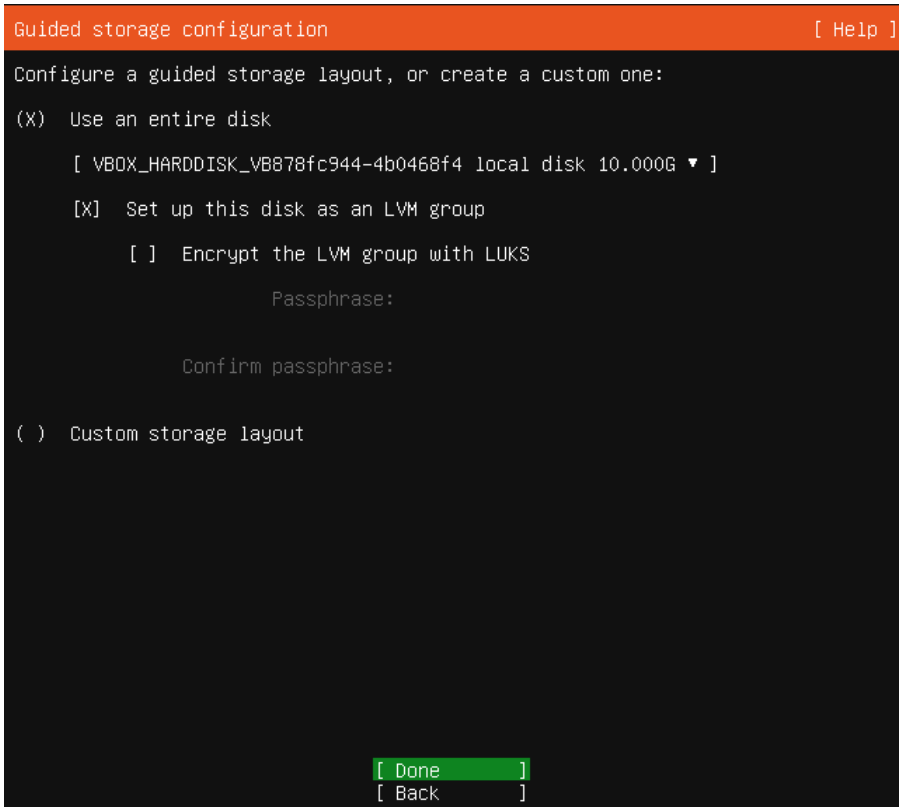
13. Select *Done* and press *Enter*



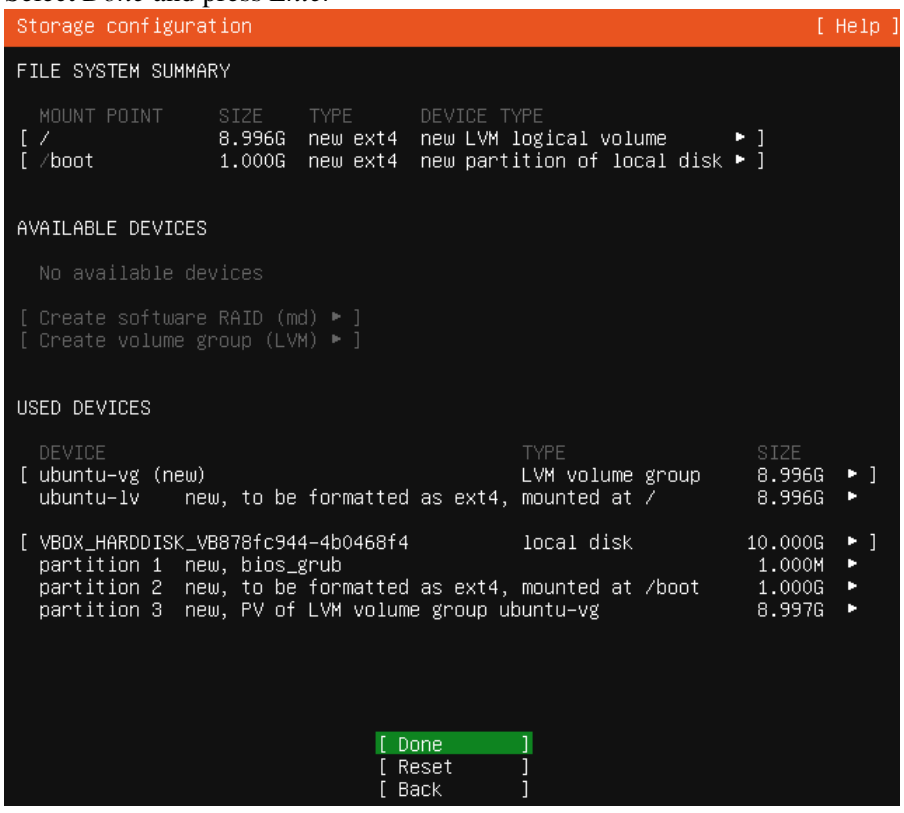
14. Select *Done* and press *Enter*



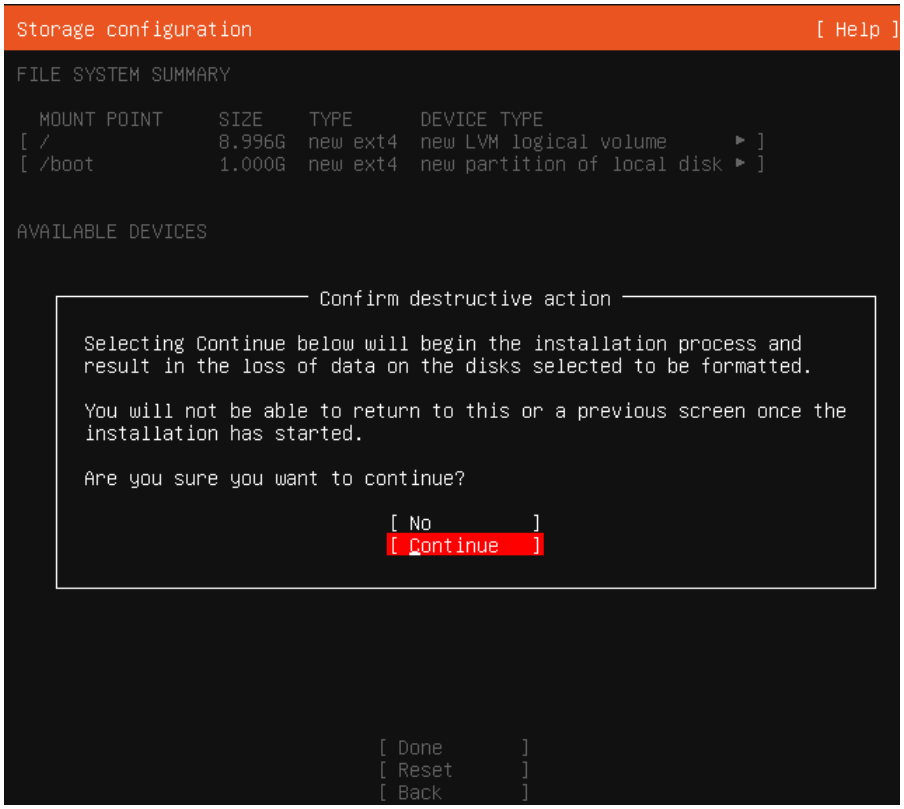
15. Select *Done* and press *Enter*



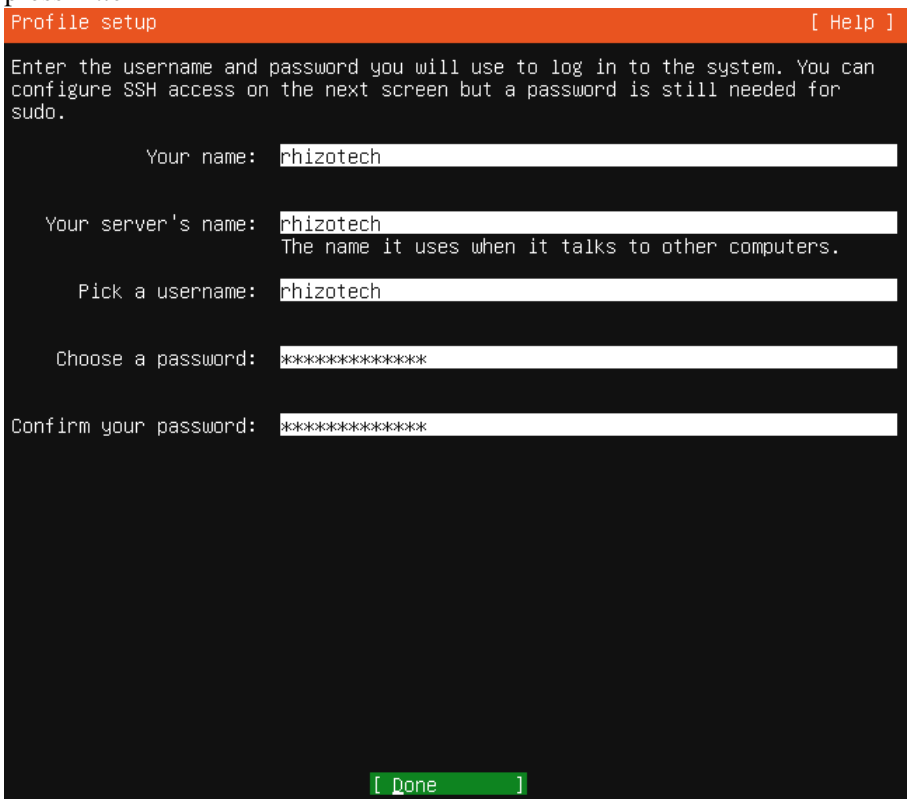
16. Select *Done* and press *Enter*



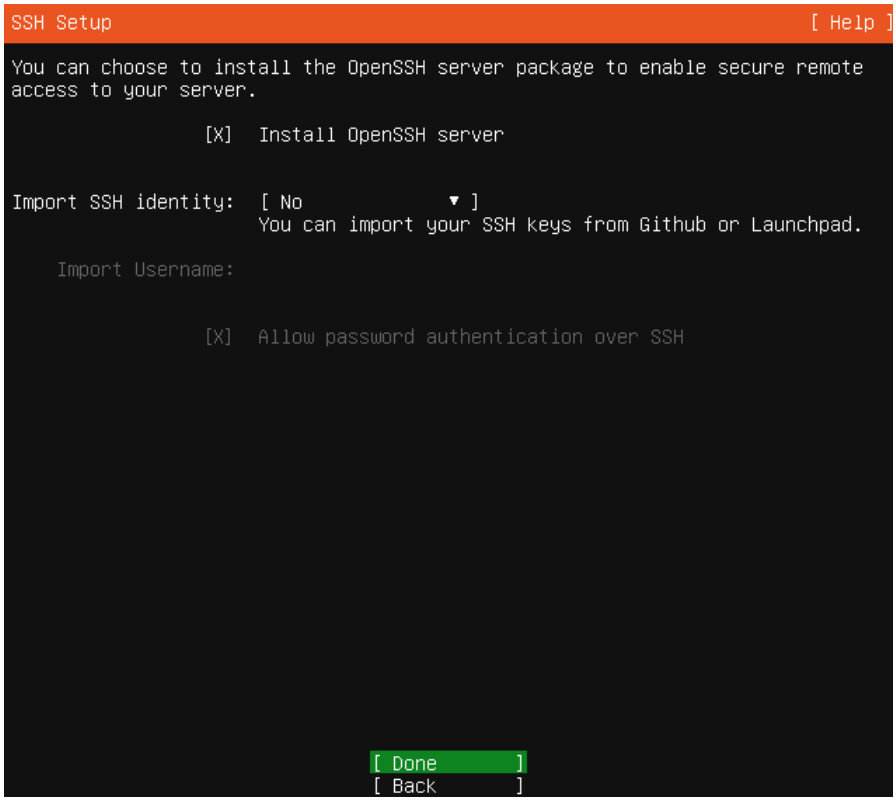
17. Select *Continue* and press *Enter*



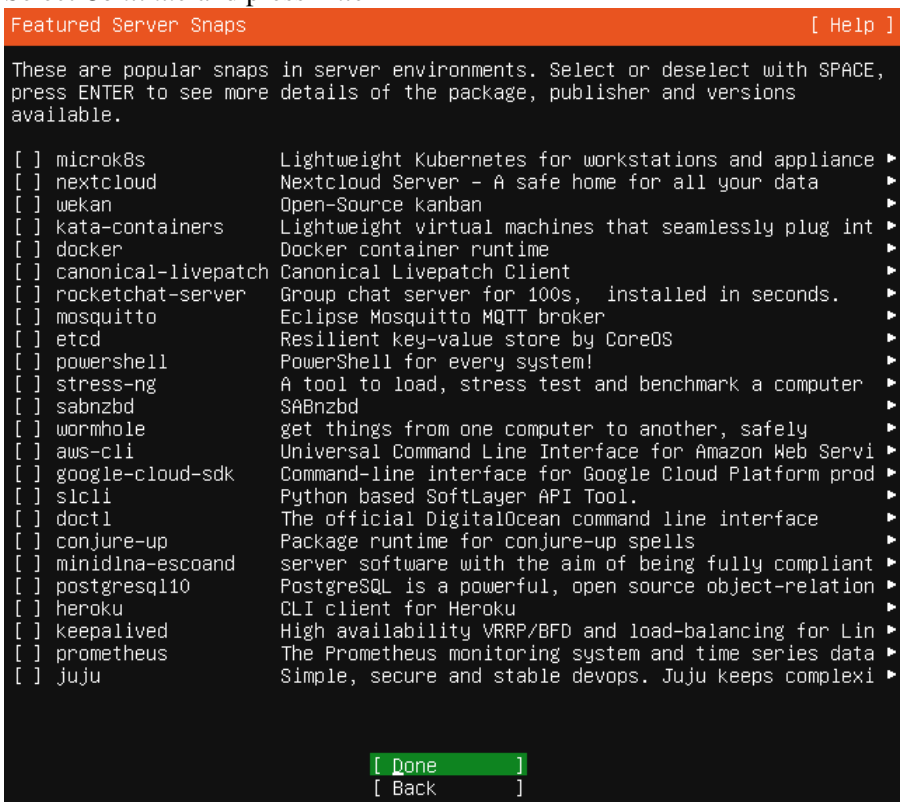
18. Set a *Name*, the *Hostname* of the Server, the *Username* and the *Password*. Then select *Done* and press *Enter*



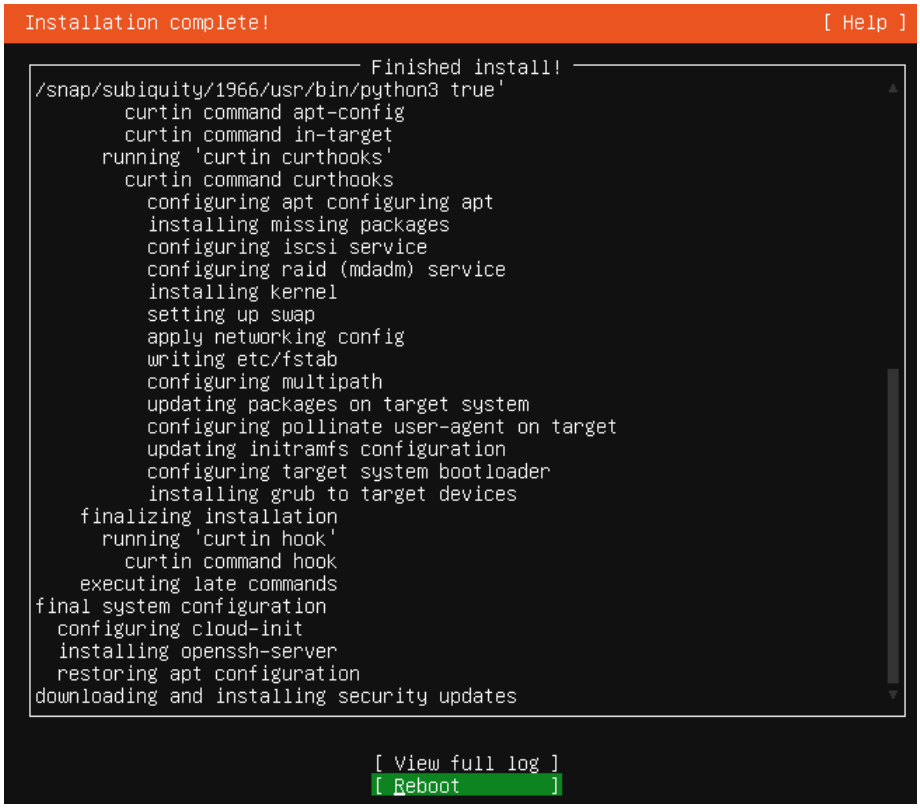
19. Select to install *OpenSSH server* and then select *Done* and press *Enter*



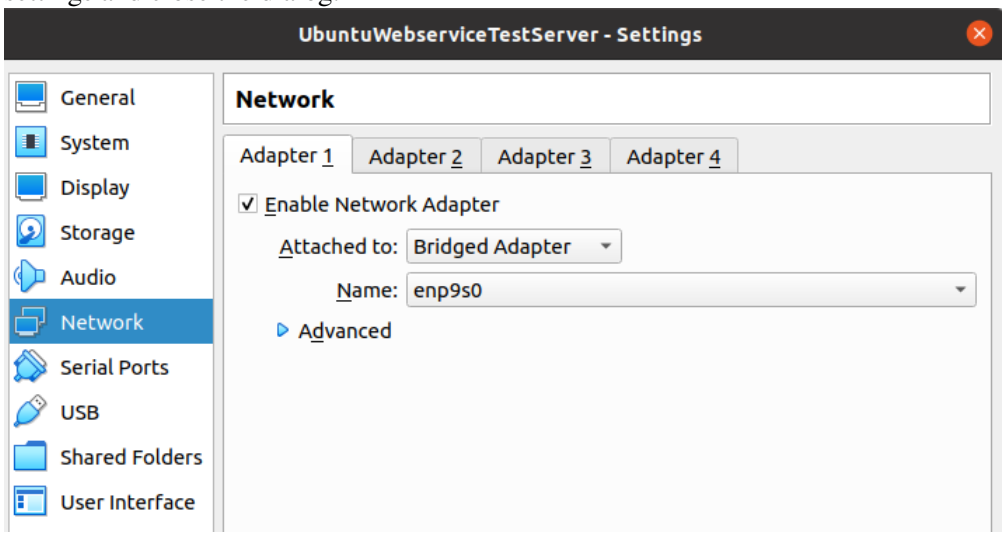
20. Select *Continue* and press *Enter*



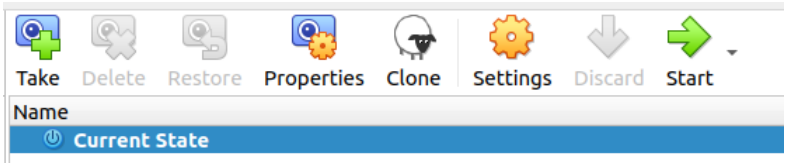
21. Wait for the installation to finished. Then select *Reboot* and press *Enter*. During reboot you might need to press enter if you thing it is stuck.



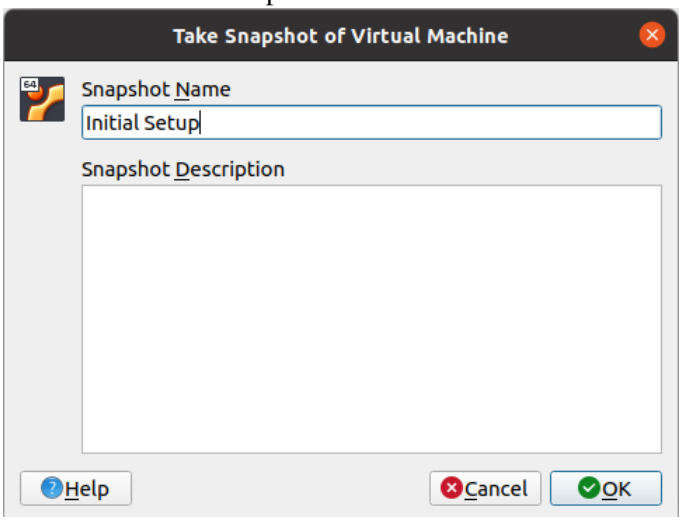
22. After reboot you can login with your *Username* and *Passwort*. At this stage you might see message printed on the screen and the login mask is not shown (only text based). You can press *Enter* to show the username input again, or you can type your *username* and press *Enter*. Then you type your *password* and press *Enter* again. Now you are logged in. With `ip a` you can show the ip-address of the virtual machine, which is not from your local network. To be able to access the virtual machine via ssh you need to configure the network settings.
23. Shut of the Virtual Machine after reboot and go into settings. There you need to set the network adapter to *Bridged Adapter* to access the SSH server and server in the Virtual Machine. Save the settings and close the dialog.



24. Then take a Snapshot of the Virtual Machine with the button *Take*.

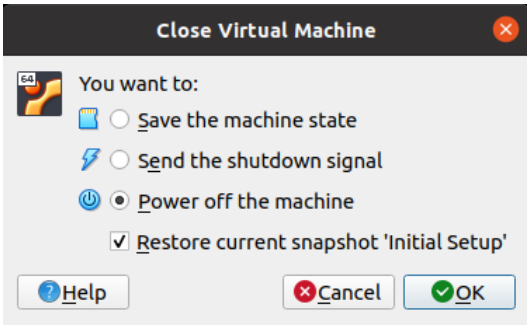


25. Set a name for the Snapshot and click OK.



26. Then you can boot the VM again an change settings. If you want to go back you can close the window and the following window appears. The option *Restore current snapshot 'Initial Setup'* resets the Virtual Machine after clicking in *OK*. This allows to test with a clean system the installation of the Software and Configuration of the Webservice.

27. At this stage you should be able to use ssh to log into the create virtual machine. Use `ssh <username>@<hostname>` to log into the virtual machine via ssh.



4 Program flow Motherboard STM32

4.1 Wakeup sources

The STM micro controller has multiple event source from which the controller could wakeup. The available sources are:

- Button 1 (WUF3)
- Button 2 (WUF2)
- Button 3 (WUF1)
- USB - Power connection (WUF5)
- Timer/RTC event (given period of time has elapsed) (WUTF)
- Reset button

4.2 Main

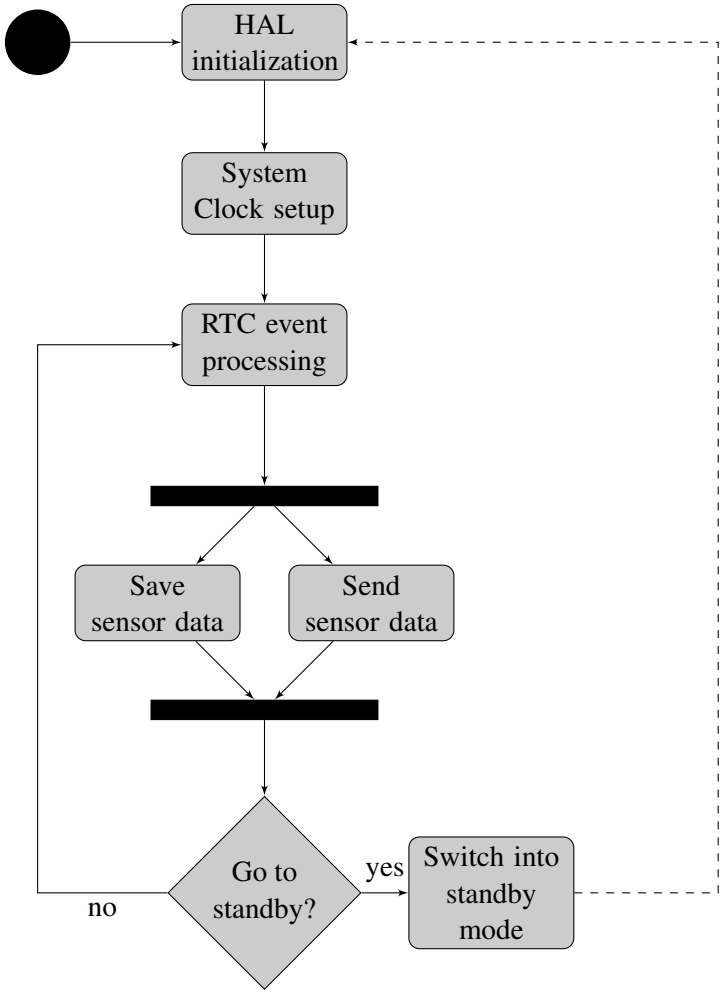


Figure 4.1: Main program flow

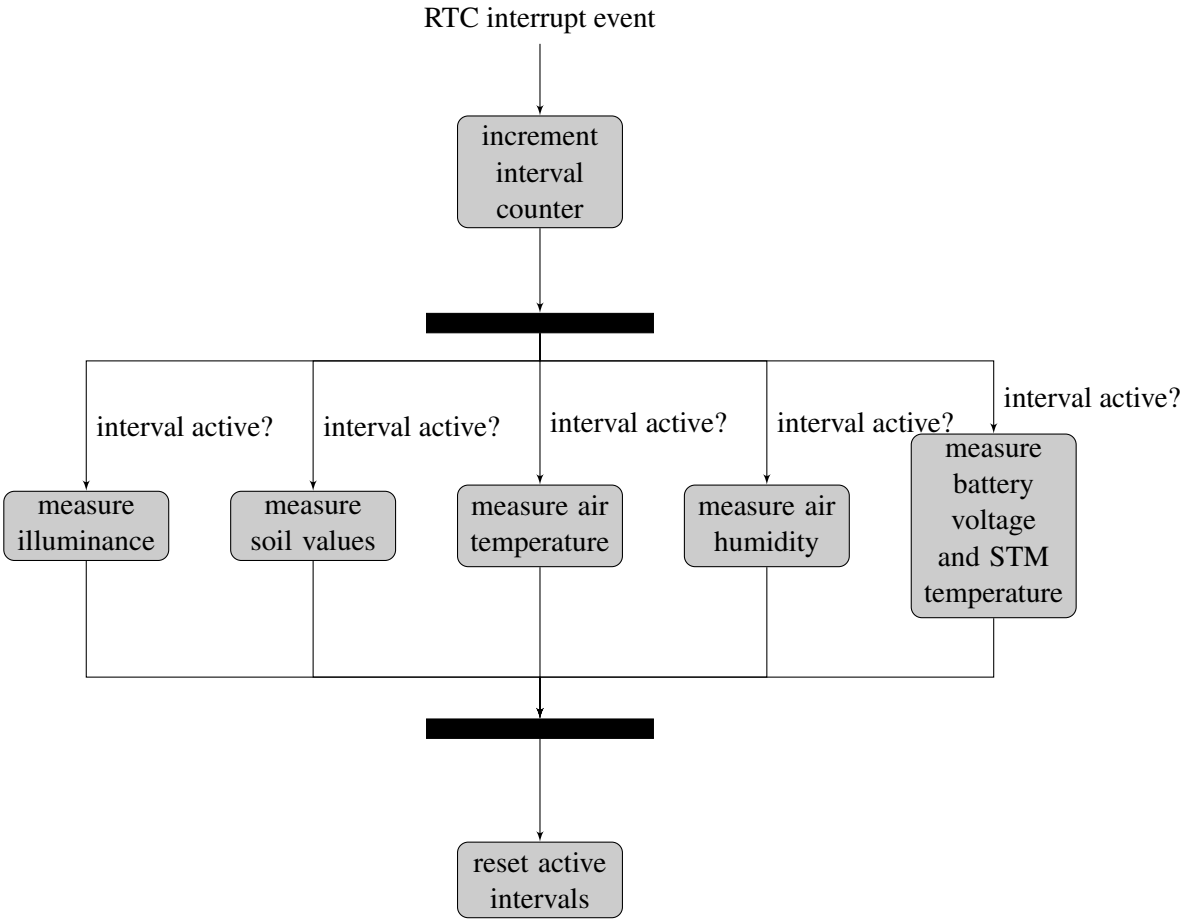


Figure 4.2: Measurement program flow

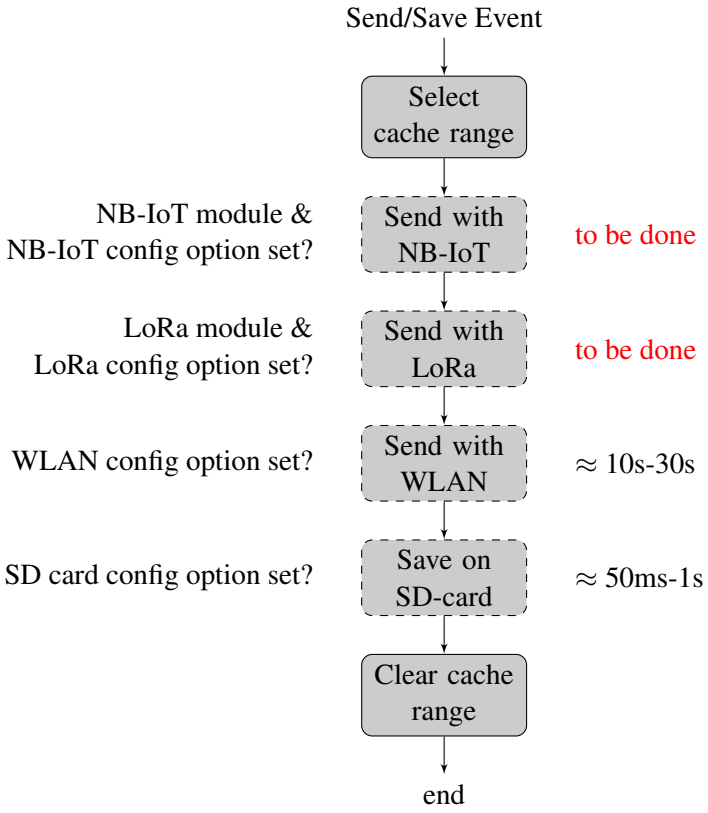


Figure 4.3: Sending/Saving program flow

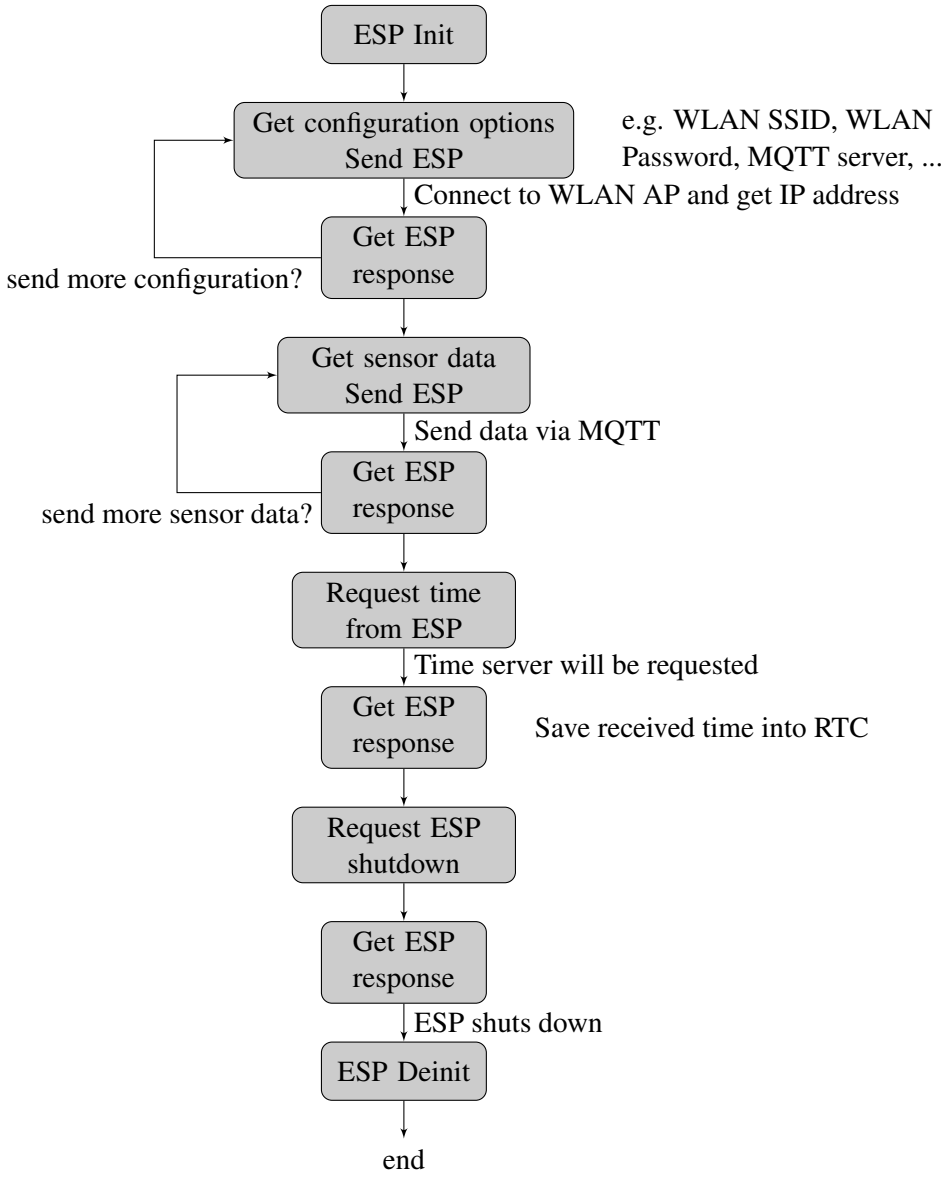


Figure 4.4: ESP communication flow, view from STM

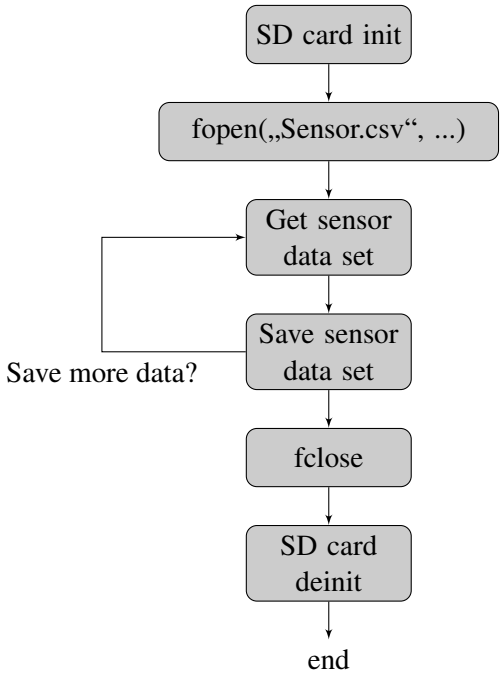


Figure 4.5: Saving onto SD card program flow

4.3 STM-ESP Communication

4.3.1 Packages

A package is a type identifier with data. The identifier is a 16bit number. Through the identifier the type/content of the package is recognized by the receiver. The following types are defined:

Number	Type	Datatype	Description
1	MQTT Server URL	string	URL or IP of the server where the MQTT-Broker is running.
2	MQTT Topic	string	Name of the topic on the MQTT server. The typical topic name is: `rhizotech/<username>/values`. Therefore this option is not mandatory!
3	MQTT Username	string	Username for authentication at the MQTT server.
4	MQTT Password	string	Password for authentication at the MQTT server.
5	MQTT Server Certificate		Certificate for checking the validity of the certificate of the MQTT server. The required certificate is the root-certificate of the trust chain.
6	NTP Server URL	string	If not provided NTP Server `ntp.pool.org` will be used. If an empty string was received no NTP request will be done.
7	WLAN SSID	string	**SSID!** (**SSID!**) (Name) of the WLAN, the connection will be established to.
8	WLAN Password	string	Password of the WLAN.
9	Sensor data	bytes	Sensor data message content.
10	MQTT ClientId	string	ClientID which will be used to connect to the MQTT-Broker this is a combination of a string e.g. with version info and the serial number.

100	WLAN/DHCP Ready	empty	Signals that a connection to the WLAN Access Point has been established and an IP address was received via **DHCP!** (**DHCP!**).
101	Turn off	empty	
102	Timestamp	timestamp	**UTC!** (**UTC!**)-Timestamp of the requested time, with maybe additional time which was counted by the ESP.
103	Configuration	string	Configuration information for the sensor box. (see **?? ??**)
104	Sensor data acknowledgement	array of bytes	Acknowledgements of the sensor data messages whether they have been sent or not

Table 4.1: Packages between STM and ESP

4.3.1.1 Strings

Strings can be transmitted without string end character, because the length of the data is transmitted and a string end character is added to all data.

4.3.1.2 Timestamp

A timestamp package contains a byte which indicates whether the time synchronization was OK or not. 0x3B is for OK and 0x85 is for not OK. Then a unix timestamp with 64bit will be transmitted in network byte order. After this a CRC code will be used. This is the same code as in the MQTT messages!

4.3.1.3 Sensor data acknowledgement

The sensor data acknowledgement contains for every packet in the last set of packets a byte, in the received order, which indicates the success or failure of sending the sensor data message via MQTT.

Number	Description
1	Success
5	General failure

Table 4.2: Numbers in sensor data acknowledgements

4.3.2 Communication Flow

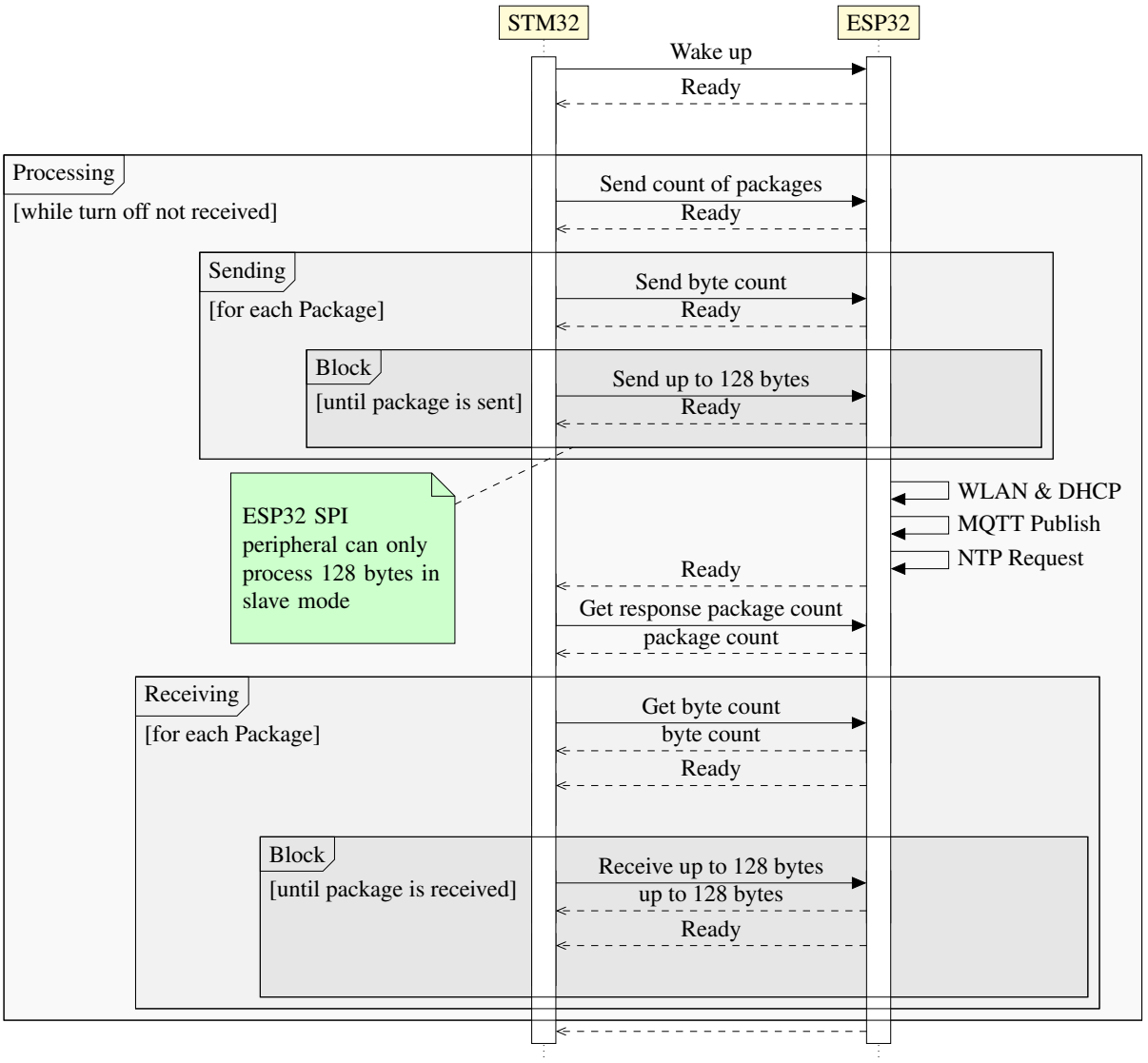


Figure 4.6: Message Sequence Chart STM-ESP communication

4.4 STM finite state machine

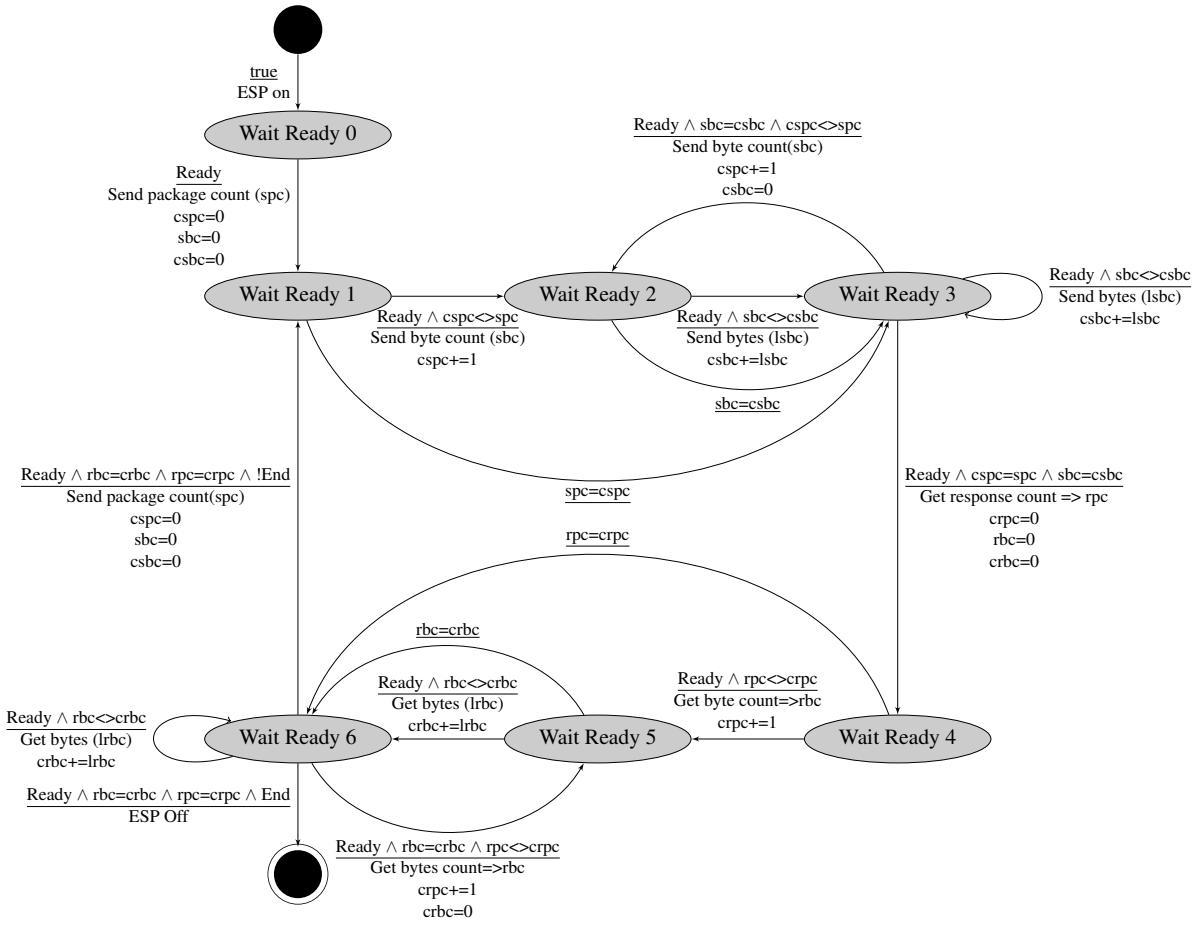


Figure 4.7: STM-ESP communication - STM finite state machine.

Variable	Long name	Description
spc	send packet count	Count of packets which will be send in the current processing round-
cspc	current send packet count	Count of sent and sending packets
sbc	send byte count	Count of bytes for the current packet
csbc	current send byte count	Send and sending count of bytes for the current packet
lsbc	local send byte count	Count of bytes for current transaction, of a packet. lsbc = sbc - csbc, limited to 128
rpc	receive packet count	Count of packets to receive.
crpc	current receive packet count	Count of received and receiving packets.
rbc	receive byte count	Count of bytes to receive for the current packet.
crbc	current receive byte count	Count of bytes received and receiving for the current packet.
lrbc	local receive byte count	Count of bytes of current transaction, of the current packet. lrbc = rbc - crbc, limited to 128
Ready	ESP ready signal	
End	ESP shutdown	The shutdown command was send to the ESP

Table 4.3: STM-ESP communication - STM finite state machine - variables

4.5 ESP finite state machine

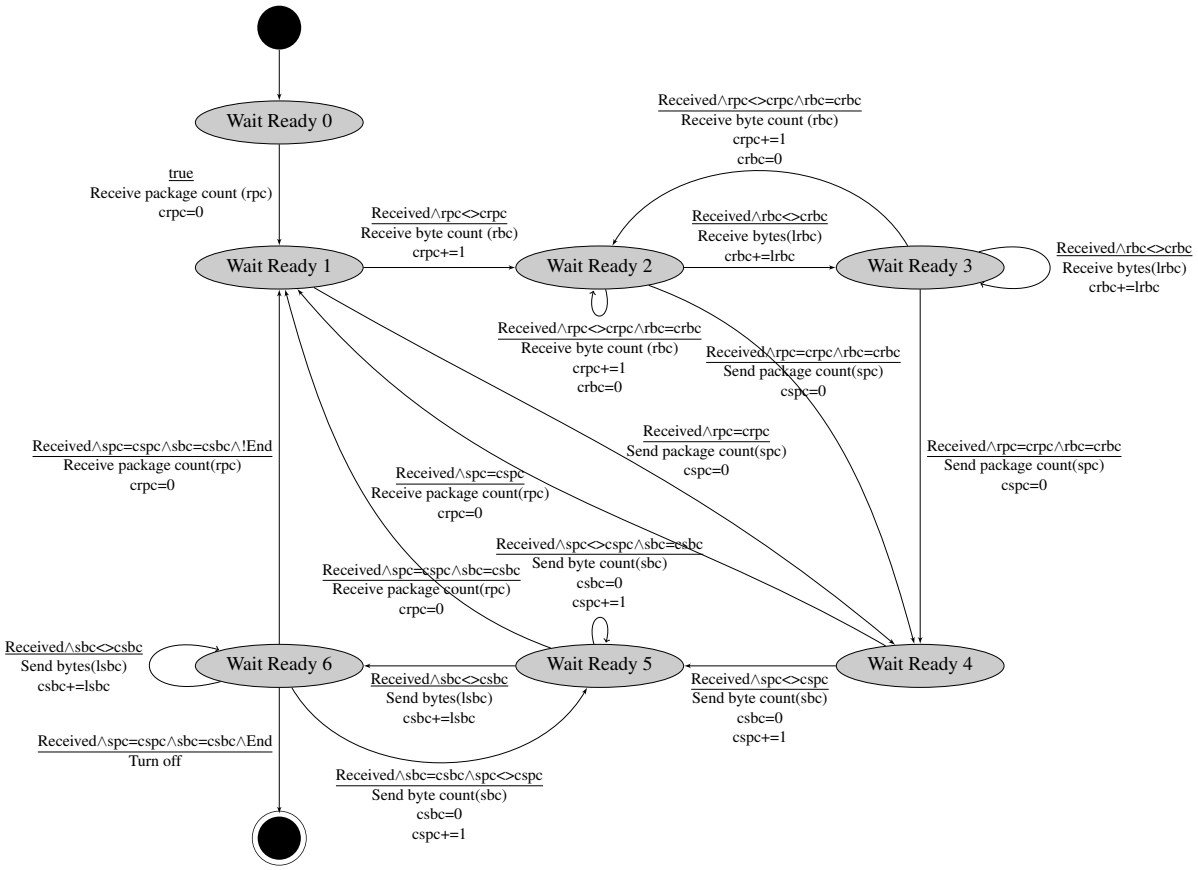


Figure 4.8: STM-ESP communication - ESP finite state machine.

Variable	Long name	Description
spc	send packet count	Count of packets which will be send in the current processing round.
cspc	current send packet count	Count of sent and sending packets.
sbc	send byte count	Count of bytes for the current packet.
csbc	current send byte count	Send and sending count of bytes for the current packet.
lsbc	local send byte count	Count of bytes for current transaction, of a packet. lsbc = sbc - csbc, limited to 128.
rpc	receive packet count	Count of packets to receive.
crpc	current receive packet count	Count of received and receiving packets.
rbc	receive byte count	Count of bytes to receive for the current packet.
crbc	current receive byte count	Count of bytes received and receiving for the current packet.
lrbc	local receive byte count	Count of bytes of current transaction, of the current packet. lrbc = rbc - crbc, limited to 128.

Received	received signal	
End	ESP shutdown	The shutdown command was send to the ESP

Table 4.4: STM-ESP communication - ESP finite state machine - variables

4.6 Sensor data storage

If the recording of sensor data values is enabled a file will be created on the SD-Card with the name `Sensor.csv` which is a **CSV!** (**CSV!**)-file which contains the recorded values. The file contains columns for all values which can be recorded with the current configuration. When the configured amount of sensors changes a new line which contains the header for the columns will be added to the csv file. If the file `Sensor.csv` doesn't exists it will be created and the column header will be added.

4.7 Saving of configurations/settings

4.7.1 Technical solution

Because the used micro controller doesn't have an **EEPROM!** (**EEPROM!**) the flash memory is used as storage. Therefore a space starting at `BASEADDRESS` is used as configuration storage.

Configuration options with a boolean, integer or floating point value are simple to save, because they have a defined length. When saving strings this gets more complicated, because the length is unknown. Sometimes a maximum length is known like in the case of the *WLAN-SSID*, which can only have up to 32 octets (bytes).

All options in memory will have a fixed memory location which is defined in the Software. For configurations with a dynamic data size on the „well "known" location a pointer is located. This pointer points to a memory location which holds the size of the following data block.

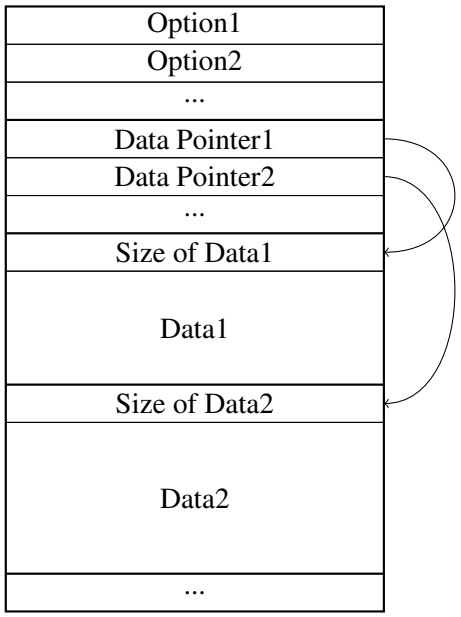


Figure 4.9: Memory mapping for configurations

When saving a block of data bytes they must be combined to blocks(lines) with 64 bits. An example is shown in the following table **??**. In the case that a line couldn't be filled the empty space can contain

anything.

i = 7	i = 6	i = 5	i = 4	i = 3	i = 2	i = 1	i = 0
i = 15	i = 14	i = 13	i = 12	i = 11	i = 10	i = 9	i = 8
...							
						i = 41	i = 40

Table 4.5: Memory mapping of block of bytes

How is the initial start handled?

The initial start is handled by a defined configuration option which should a known value, which is not correct at the initial startup. With this mechanism the initial start could be detected. For development this is also good, because when changing the amount of configuration options only the value in this configuration option needs to be changed.

All addresses in the memory are virtual addresses, which are starting at the `BASEADDRESS`.

4.7.2 Configuration options

4.7.2.1 General options

- **WLAN!**-Connection
  - SSID
  - Password
  - Certificates?
    When connections to WLAN networks other then WPA2-Personal/Preshared Key should be established maybe additional parameters are needed, like certificates in WPA2-Enterprise.
- **MQTT!** (**MQTT!**)-Endpoint
  - Server
  - Topic (should be preconfigured)
  - Username
  - Password
  - ...
- Time reference
  - NTP-Server name/ip address
  - Do time requests to NTP[1]-Server
  - Use NB-IoT???
- Go into standby
- Save data to SD-card
  - Do not save
  - Save directly
  - Cache a short time (Energy saving saving)
- Communication over:
  - None
  - WLAN (**WLAN!**)

Which possibilit should be available

[1] **NTP!** - **NTP!**

- LoRa (**LoRa!**)
- NB-IoT (**NB-IoT!**)

4.7.2.2 Interval (measurement, sending)

- Wake up (greatest common divisor of other options)
- Wireless transmitting
- Soil moisture & temperature
- Air temperature & humidity
- Illuminance
- Battery voltage
- pH value

4.7.2.3 Calibration

- Soil moisture
- pH
- Soil (temperature)

The soil sensors may should have some predefined calibrations for different types of soil.

4.7.3 Configuration Fileformat

4.7.3.1 Run requirement file format

`basicsetup.conf.txt`
Communication: (SD | WLAN | LoRa | NB-IoT | GSM)
The value *SD* is used in the case that the sensor data should be saved onto the SD-Card. The option *WLAN* is used in the case the sensor data should be transmitted via WLAN. If the sensor data should be transmitted via LoRa then option *LoRa* should be used. The option *NB-IoT* will use the standard NB-IoT for data tranmission and the option *GSM* will use the GSM standard for data transmission. It is allowed to combine multiple options. The combination is done with a space between the options.
WLAN-SSID: <SSID>
SSID (Name) of the Wifi the sensor box should be connected to
WLAN-Password: <Password>
Password of the Wifi the sensor box should be connected to
MQTT-Broker: <URL>
URL! (**URL!**) of the MQTT Broker (e.g. `iot.nt.th-koeln.de`)
MQTT-Username: <Username>
Username for the MQTT Broker
MQTT-Password: <Password>
Password for the MQTT Broker
NB-IoT-APN: <APN-URL>
Name of the APN e.g. `iot.1nce.net`. This option is only needed if transmission via *NB-IoT* is enabled.
NB-IoT-NetworkLocationAreaIdentification: <NetworkLocationAreaID>
The network location area identification is only needed if transmission via *NB-IoT* is enabled. An example for this value is `26201` for the 1nce network.
NB-IoT-DestIP: <DestIP>
Destination IP for the sensor data. This is the IP address of the server side **VPN!** (**VPN!**) tunnel interface to the APN. This option is only needed if transmission via *NB-IoT* is enabled.
NB-IoT-DestPort: <DestPort>

Destination Port for the sensor data. This is the Port number of the server side **VPN!** tunnel interface to the APN. This option is only needed if transmission via *NB-IoT* is enabled.
Server-CA-Certificate:
---BEGIN CERTIFICATE---

---END CERTIFICATE--

List of public server certificates. The certificates must be in PEM[2] format. A possibility to retrieve this certificate/certificates is to use openssl. In a linux environment the following is possible:

```
openssl s_client -showcerts -connect <BrokerURL>:8883 </dev/null
```

Listing 4.1: basicsetup.conf.txt example

```
Communication: SD WLAN LoRa NB-IoT
WLAN-SSID: ThisIsA_WLAN
WLAN-Password: ThisIsASavePassword
MQTT-Broker: iot.nt.th-koeln.de
MQTT-Username: ThisIsaMQTTBrokerUsername
MQTT-Password: ThisIsaMQTTBrokerPassword
NB-IoT-APN: iot.1nce.net
NB-IoT-Network: 26201
NB-IoT-DestIP: 10.64.72.81
NB-IoT-DestPort: 10000
Server-CA-Certificate:
-----BEGIN CERTIFICATE-----
MIIC8DCCAlmgAwIBAgIJAOD63PlXjJi8MA0GCSqGSIb3DQEBBQUAMIGQMQswCQYD
VQQGEwJHQjEXMBUGA1UECAwOVW5pdGVkIEtpbmdkb20xDjAMBgNVBAcMBURlcmJ5
MRIwEAYDVQQKDAlNb3NxdWl0dG8xCzAJBgNVBAsMAkNBMRYwFAYDVQQDDA1tb3Nx
dWl0dG8ub3JnMR8wHQYJKoZIhvcNAQkBFhByb2dlckBhdGNob28ub3JnMB4XDTEy
MDYyOTIyMTE1OVoXDTIyMDYyNzIyMTE1OVowgZAxCzAJBgNVBAYTAkdCMRcwFQYD
VQQIDA5VbmlOZWQgS2luZ2RvbTEOMAwGA1UEBwwFRGVyYnkxEjAQBgNVBAoMCU1v
c3F1aXR0bzELMAkGA1UECwwCQ0ExFjAUBgNVBAMMDW1vc3F1aXR0by5vcmcxHzAd
BgkqhkiG9w0BCQEWEHJvZ2VyQGF0Y2hvby5vcmcwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBAMYkLmX7SqOT/jJCZoQ1NWdCrr/pq47m3xxyXcI+FLEmwbE3R9vM
rE6sRbP2S89pfrCt7iuITXPKycpUcIU0mtcT1OqxGBV2lb6RaOT2gC5pxyGaFJ+h
A+GIbdYKO3JprPxSBoRponZJvDGEZuM3N7p3S/lRoi7G5wG5mvUmaE5RAgMBAAGj
UDBOMB0GA1UdDgQWBBTad2QneVztIPQzRRGj6ZHKqJTv5jAfBgNVHSMEGDAWgBTa
d2QneVztIPQzRRGj6ZHKqJTv5jAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUA
A4GBAAqw1rK4NlRUCUBLhEFUQasjP7xfFqlVbE2cRy0Rs4o3KS0JwzQVBwG85xge
REyPOFdGdhBY2P1FNRy0MDr6xr+D2ZOwxs63dG1nnAnWZg7qwoLgpZ4fESPD3PkA
1ZgKJc2zbSQ9fCPxt2W3mdVav66c6fsb7els2W2Iz7gERJSX
-----END CERTIFICATE-----
```

4.7.3.2 Operational settings

settings.conf.txt
[Intervals]
Starts a section in which only intervals will be configured. The order of options in the section has no

[2] **PEM! - PEM!**

relevance. If an option is used more than once the last parsed value will be used. An Interval is given in seconds. The minimal value for all intervals is one second. Depending on the type of interval the minimum might be higher. The maximum value for an interval is 65535 seconds, which is 18 hours, 12 minutes and 15 seconds.
AirTemperature: <Interval>
AirHumidity: <Interval>
SoilTemperature: <Interval>
SoilMoisture: <Interval>
Illuminance: <Interval>
BatteryVoltage: <Interval>
pHValue: <Interval>
Transmisson: (Direct | <Interval>)

[Time]
TimeSource: (None | NTP | GPS)
Source for time synchronisation. The value can be *None* for no time synchronisation. *NTP* for time synchronisation via Wifi. In this case the option `TimeURL` must be given. The options *GPS* specifies the time synchronisation via **GPS!** (**GPS!**).
TimeURL: <URL>
Optional URL of NTP time server.
The typical time server is: *pool.ntp.org*.
Inside a company/campus network a dedicated time server might be needed. Inside the TH-Köln this is *time.th-koeln.de*
TimeSynchronisationInterval: <Interval>
Interval in which a time synchronisation will be executed. This value should indicate a time synchronisation a few times a day. More often is not required and slower may increase the time difference between real and local time of the sensor box.

Listing 4.2: settings.conf.txt example

```
[Intervals]
AirTemperature: 5
AirHumidity: 5
SoilTemperature: 5
SoilMoisture: 5
Illuminance: 5
BatteryVoltage: 100
pHValue: 30

[Time]
TimeSource: NTP
TimeURL:pool.ntp.org
TimeSynchronisationInterval: 21600
```

4.7.3.3 Calibration file format

`calibration.txt`

Listing 4.3: calibration.txt example

```
[SoilTemperature]
```

```
All:Linear, 23.23 25.0, 22.64 24.42, 46.34 48.83

[SoilMoisture]
1:Linear: 23.23 24.0, 22.64 24.42
2:Linear: 23.23 24.0, 22.64 24.42, 46.34 48.83

[pHValue]
1:Linear: 1.1242 4.00, 1.5354 10.00
2,3,4,12:Linear: 1.1242 4.00, 1.5354 10.00
```

The calibration file contains sections for every sensor type. *SoilTemperature*, *SoilMoisture*, *pHValue*, *AirTemperature*, *AirHumidity*, *Illuminance* and *BatteryVoltage*. In each of these sections either the calibration for all sensors or specific sensor could be given. Each line in a section contains the calibration values for a set of sensors. The set of sensors is specified by either the key word *All* which means that this calibration should be applied to all sensors in this list or by specifying the sensor number to which the calibration should be assigned. After the sensor selection a double point comes and the specification of the interpolation is given currently only *Linear* is possible. After a second double point a comma separated list of tupels is given. Each tupel consists of two values the first is the input value and the second is the output value.

4.7.3.4 Information file format

`information.txt`
Serialnumber: <Serialnumber>
Serial number of the sensor box which identifies the sensor box uniquely.
NB-IoT-Src <Src-IP>:<Src-Port>
Source IP Address and source port of the messages sent via NB-IoT.

Listing 4.4: information.txt example

```
Serialnumber: 498758934275092375902375098 7
Git Hash: 5b642071c363e68a9bfe62dffd69ff261d6b57d8
Git Branch: master
Compile date: Fr 11. Okt 10:58:11 CEST 2019
Build configuration: DEBUG
NB-IoT-Src: XXX.XXX.XXX.XXX:XXXXX
```

Serialnumber: <Serialnumber>
Serial number of the Sensorbox which is needed in the Webservice to distinguish the sensorbox from each other. **Git Hash:** <Hash>
Git hash of the latest commit on the branch on which this build was generated with. This doesn't mean that there are any additional files changed, which may not have been commited! **Git Branch:** <Branch>
Git branch on which this build was created. This is only for a faster checking what feature set may be included during the development time. **Compile date:** <Time and Date>
Time and date on which this build was generated. (This is another time than the time of the commit on which the current build was generated.) **Build configuration:** <Configuration>
This is the build configuration which was used during the build. In the *RELEASE* build e.g. the logging is not available, but in a *DEBUG* build this is available, therefore this information is available.

4.8 Standby Cache

4.9 Standby Registers

4.10 NB-IoT

Currently from our provider for NB-IoT (1nce) there is no support for chaning the timer values T3412 and T3324. Therefore we can not use the **PSM!** (**PSM!**) and/or **eDRX!** (**eDRX!**) mode of NB-IoT. Therefore an implementation is taken which turns the NB-IoT module only for a short time on, where then the data is transmitted. The data will typically be cached so that a great amount of power can be saved.

4.10.1 Program flow

The NB-IoT sending begins after a measurement, there is then a check whether it is necessary to send data via NB-IoT, which will happen, when NB-IoT is configured and the sending interval has elapsed or the cache is full. The program flow of NB-IoT is depicted in figure **?? ??**

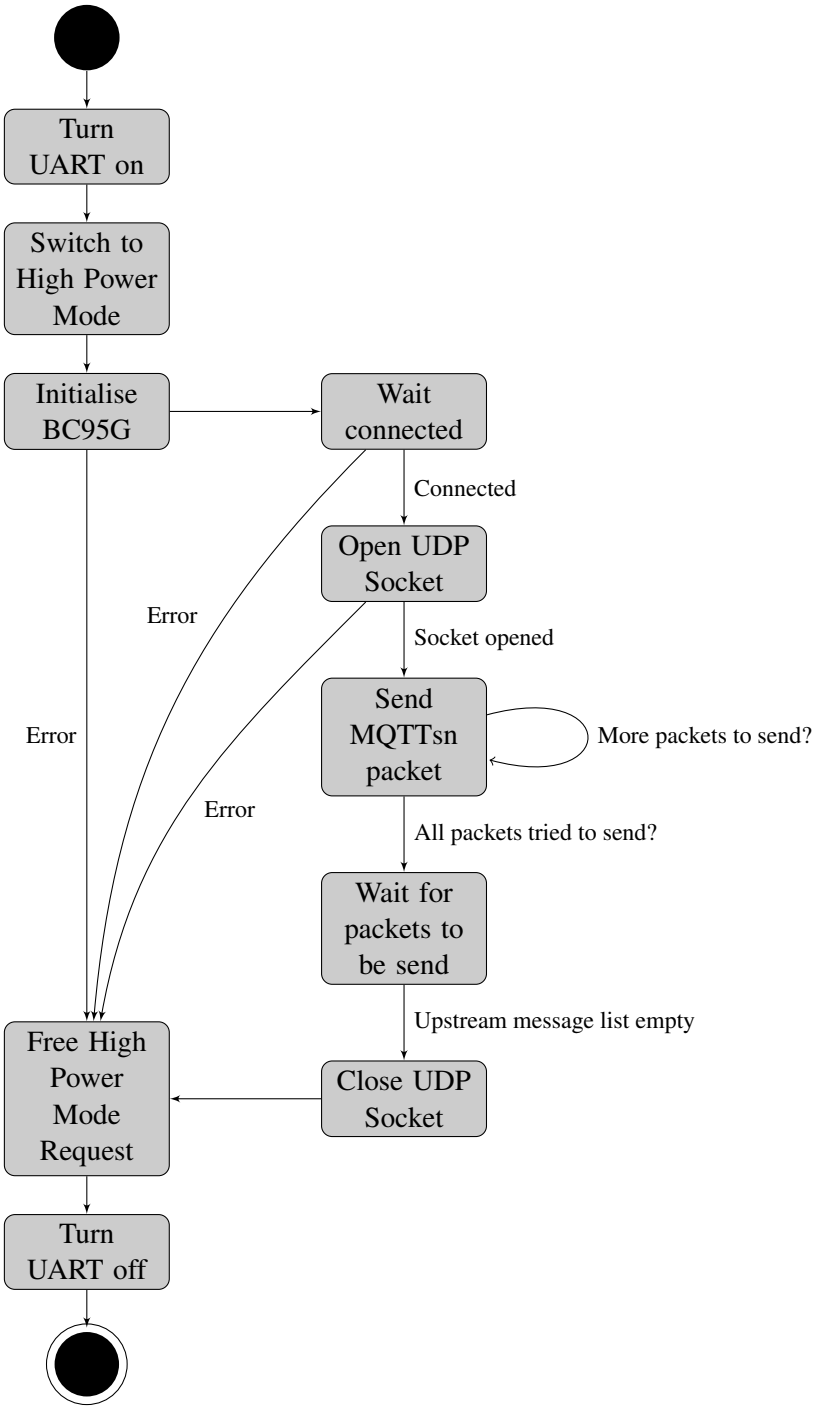


Figure 4.10: NB-IoT sequence

4.10.2 Implementation

The implementation of NB-IoT is based on the UART interface of the BC95G. The module is controlled via the so called AT commands. The AT commands have a defined time in which the module needs to answer the command. If this does not happen a command has timed out, which is interpreted as a failure and in the configuration phase this leads to the termination of the process. An AT command is answered with data and a success information `OK` or `ERROR`. A timeout and an error is interpreted the same. Beside the normal request-response model also non-synchronous messages can be received which are called

URC! (**URC!**). The messages can be send at any time from the module, to signal for example that the registration state of the module has changed. The messages are received via DMA interrupts from the UART peripheral.
The decoding of the messages is most of the time not interpreted after the function `AT_Command` but rather in the message receiving where the type of message is detected and directly decoded. The decoded information is then saved into the member of the class `BC95G` to be later accessed. An example for this is the command `+CEREG` which signals the registration status of the module. The registration status is saved into the member `registered`. The data is decoded for a request via the AT command `AT+CEREG?` and also for the **URC!** of the `CEREG` command.
The BC95G can store up to 255 queued upstream messages, which are identified by a sequence number. Normally the module acknowledges a send **UDP!** (**UDP!**) message, but this doesn't mean it has been really sent, which can be check by requesting the queue of upstream messages. Therefore at the end, before closing a socket the list of queued upstream message is checked. If a message has been acknowledged it is removed from the message cache.

4.11 MQTT-SN

MQTT-SN! (**MQTT-SN!**)

Transpar
Gateway

Messagel

> Note that because MQTT-SN does not support message fragmentation and reassembly, the maximum message length that could be used in a network is governed by the maximum packet size that is supported by that network, and not by the maximum length that could be encoded by MQTT-SN.

4.12 GPS

The **GPS!** (**GPS!**) is used as time source for time synchronization. In case of communication via NB-IoT it is not possible to have a reliable time synchronization via this link, therefore **GPS!** is used, which provides a time stamp, because the **GPS!** is based on a precise time.
To achive the time synchronization the A2235H module is used to received the GPS signal. The GPS module signals via **UART!** (**UART!**) a **NMEA!** (**NMEA!**)-**RMC!** (**RMC!**) which can be decoded to receive a up to date **UTC!** time. This is then set in the micro controllers **RTC!** (**RTC!**) as current time.

4.13 Logging and Debugging

For the development process there are multiple ways to get information about the system. In the following the debugging and logging tools will be explained which can be used in the development phase.

4.13.1 Debugging STM32

The STM32 can be debugged through the **SWD!** (**SWD!**) interface. As development tool *System Workbench for STM32* is used together with an *ST-Link V2*, which uses internally a *gdb* to debug the micro controller.
For the normal operation of the Sensorbox the debugger can not be used, because the debugger increases the standby current dramatically. Without debugger around 25 µA should be drained from the battery. With the debugger the current is at least 800µA. Because special instructions are used in the Debugging builds to improve the debugging experience there exists also a Release build which doesn't execute these instructions, allowing to reach the lowest current consumption. This current consumption can only be reached after the debugger has been detached and the power had been turned off for a few seconds!

4.13.2 Logging STM32

In the Debug build of the STM Motherboard software it is possible to get a textual logging output via a UART. To read the output a 3.3 V USB to UART/TTL/RS232 converter is needed. Only the RX (Receive) channel and ground needs to be connected.
Under linux system the command screen can be used to view the log output. The following can be used to install the tool under debian based systems:

```
sudo apt install screen

/bin/cat <<EOM >~/.screenrc2
# Enable mouse scrolling and scroll bar history scrolling
termcapinfo xterm* ti@:te@
EOM
```

The logging can be started with the command:

```
screen /dev/ttyUSB0
```

Depending on the connection order of other serial devices it might be possible that the device `ttyUSB0` has another name. After a restart this name can also change.
To temporarily stop the screen command press `Crtl+A` and then `D`. To restart then list the open connection and reopen the connection.

```
screen -ls
screen -r <id>
```

A final closing of a connection is achieved via pressing `Ctrl+A` and then type `:quit`
The log output is not only shown via the UART interface. If an SD card is inserted into the Sensorbox the log is written to the SD card into a log file.
The logging is only enabled in the Debug mode, because it uses additional energy and needs a lot more time for some commands to finish.
In the program a log entry can be created with the commands:

- `LOGV` - Verbose
- `LOGI` - Informational
- `LOGIS` - Informational Success
- `LOGIW` - Informational Warning
- `LOGIE` - Informational Error
- `LOGW` - Warning
- `LOGE` - Error

The syntax for the commands is nearly the same as of `printf`, with the change that every log-message will have a tag, which is the first parameter to the log-function.

```
LOGI("Tag", "Message %d", 2389);
```

4.13.3 Debug/Logging ESP32

The ESP can be debugged with the command `make monitor` from the *ESP-IDF*, or the log can also be shown with the command:

```
screen /dev/ttyUSB2 115200
```

The screen command can be used like described above. The additional parameters gives the baudrate, which is the default configured value in the `sdkconfig` which can be changed with the command `make menuconfig`. Note: The devices name depends on the amount and type of USB devices you have connected to your machine.

4.14 Time synchronization/Time source

The sensor box needs to have a correct time to save and send sensor values. The sensor box internal clock, the **RTC!** (**RTC!**), should be synchronized with the **UTC!** (**UTC!**). The following time sources can be used:

4.14.1 WLAN - NTP

In the case of **WLAN!** access a **NTP!** (**NTP!**) server can be used to synchronize the time. Therefore a time server needs to be configured. The **NTP!** protocol runs over the **UDP!** protocol therefore access trough the network with UDP packets on port 123 must be possible to use NTP time synchronization in the case of **WLAN!** as communication interface.

4.14.2 NB-IoT/WLAN - GPS

In the case of any transmission system it is possible to use **GPS!** (**GPS!**) as time source, because **GPS!** does provide a precise time reference. Therefore for all transmission systems it is possible to use **GPS!**. **GPS!** provides a **UTC!** timestamp. GPS is typically difficult to use in indoor applications because the signal strength of the GPS signal is not sufficient to receive the signals indoor.

4.14.3 Synchronization moments

In case of GPS the time is synchronized when the synchronization interval has elapsed or the time hasn't be synchronized successfully beforehand. In the case of WLAN it is a bit more difficult. There the time synchronization happens, when the time has not been synchronized before, then without sensor data a connection to the wireless network is established and the time is retrieved. Otherwise the time will be synchronized only after the time synchronization interval has elapsed and a data transmission via WLAN has been done. This happens because most of the energy can be saved, while the wireless module is already connected and turned on. In normal operation, no connection to time synchronization is established.

5 Webservice

5.1 General description

The webservices allows users to the access the sensor data of the SensorBox. The Webservice will also be capable of managing the SensorBox. Like providing examples for the basic configuration and guidance on how to set the configuration values.
A user who is registered on the webservice can view the data of a SensorBox. A user can be assigned to be a management user of a SensorBox. The management user of a SensorBox can change the Intervals/Configuration of the SensorBox. Also the security keys can be regenerated. The relationships a users can have with a SensorBoxes are depicted in the following picture.

Figure 5.1: Webservice user relation ship to SensorBox

5.2 Data structure

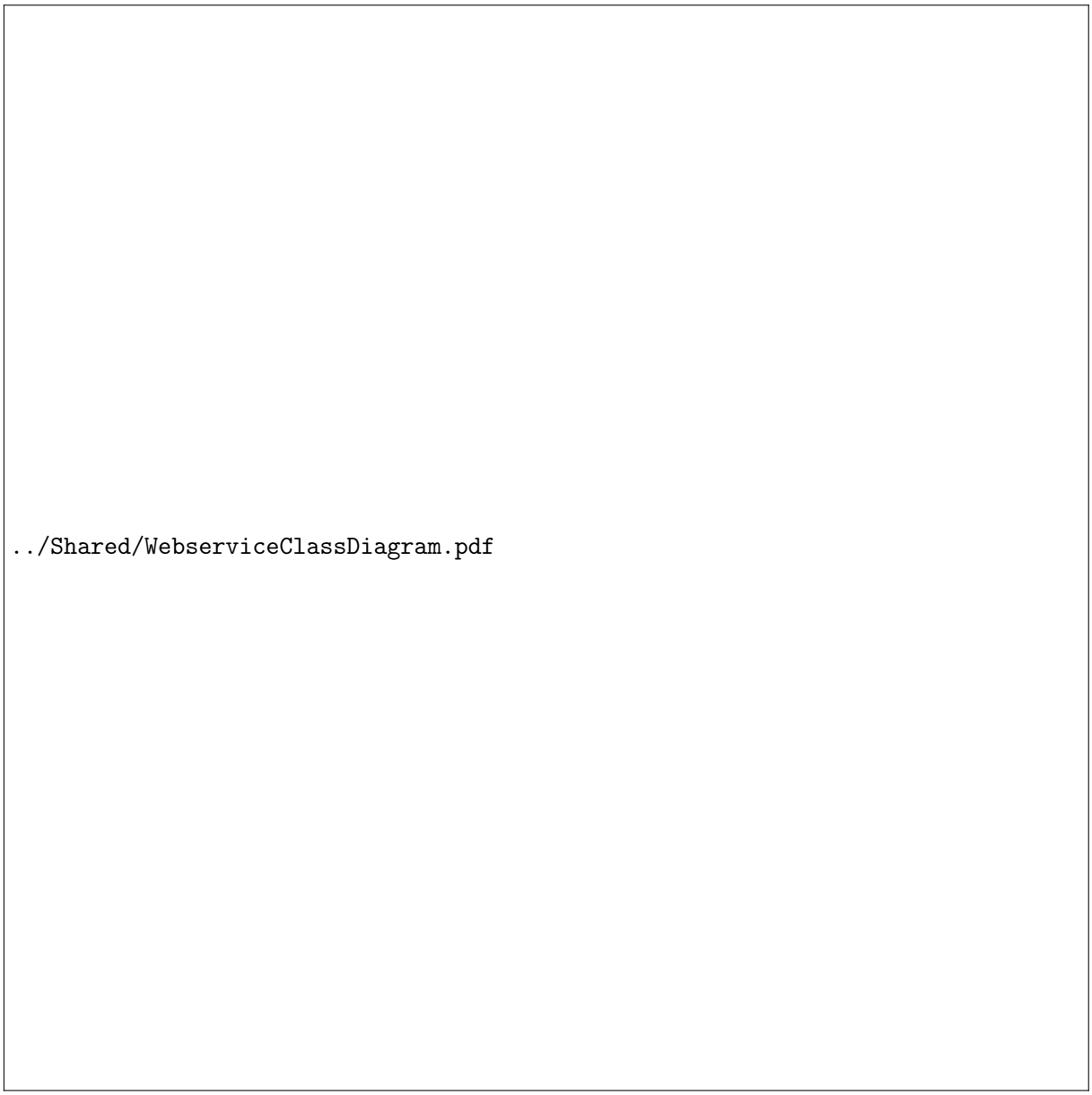


Figure 5.2: Webservice class digram

- User
  This data entity represents a user which can login into the system.
- Application
  This data entity represents a connection between a user and a sensor box. It allows the user to select a custom name for a sensor box.
- Widget
  This data entity represents a graph on the dashboard. It sets the properties for the graph.
- SensorBox
  This data entity represents a sensor box and its settings.
- DataPoint
  This data entity represents a data value from the sensor box. A data value is only one single value

from one sensor.
- Sensor
  This data entity represents a sensor e.g. a soil temperature sensor
- SensorType
  This data entity represents a type of sensor e.g. the soil temperature

5.3 Battery estimation

The Sensorbox are powered by a Lithium-polymer-akkumulator, to have a better overview about the remaining battery runtime, we implement a battery estimation algorithm. The algorithm used Mathwork Matlab's curve fitting application to calculate a mathematically function for the battery discharging curves. Based on this functions we can calculate the remaining battery runtime. The figure below shows the calculated discharging curve.

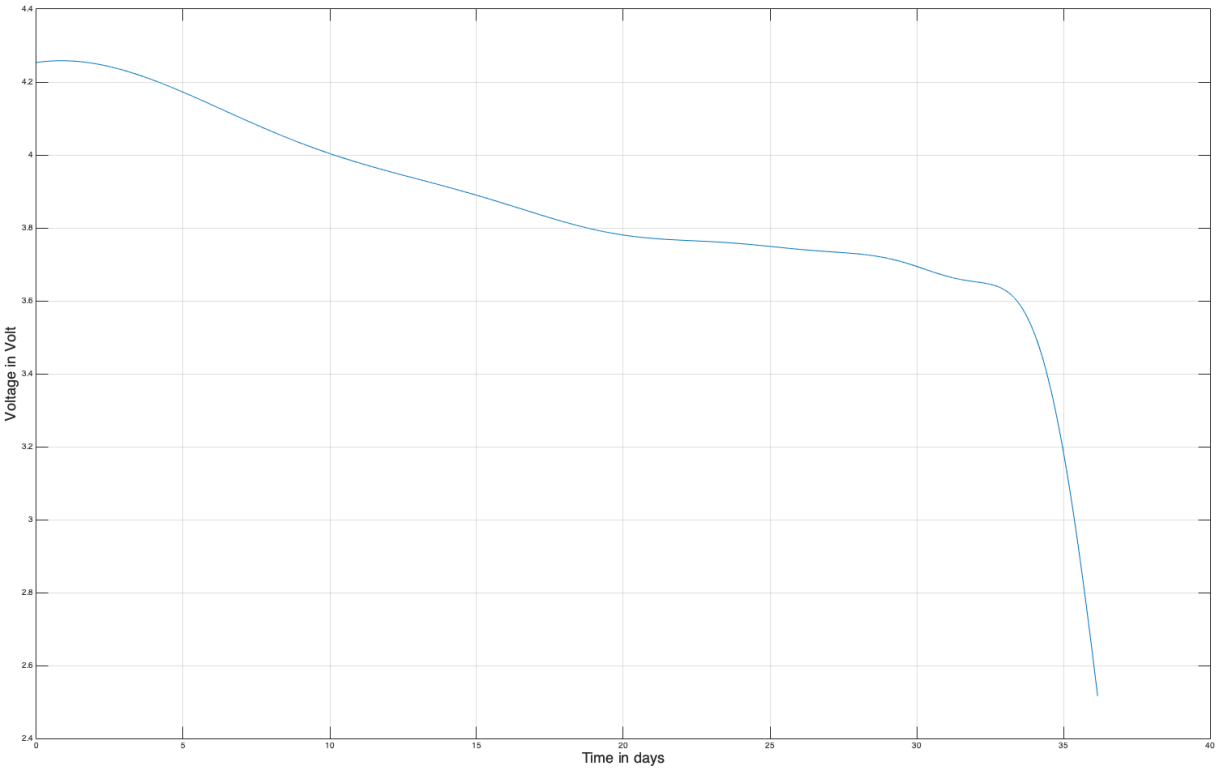


Figure 5.3: Battery discharging curve

$$y = a[0] \cdot e^{-\frac{x-b[0]}{c[0]}^2} + a[1] \cdot e^{-\frac{x-b[1]}{c[1]}^2} + a[2] \cdot e^{-\frac{x-b[2]}{c[2]}^2} + a[3] \cdot e^{-\frac{x-b[3]}{c[3]}^2} + a[4] \cdot e^{-\frac{x-b[4]}{c[4]}^2}$$

$$a = [4.162, 2.222, 2.092, 0.855, 0.634]$$

$$b = [-0.097, 2.576, 3.970, 4.307, 3.120]$$

$$c = [2.388, 1.459, 0.711, 0.340, 0.653]$$

The testbench approved that the accuracy was higher when the battery voltage fall. We have this relationship because the derivation gets higher, this leads to a bigger difference between two measurements and to the higher accuracy of the estimation. The figure below shows the root-mean-squared deviation between the calculated battery runtime and the reference battery runtime.

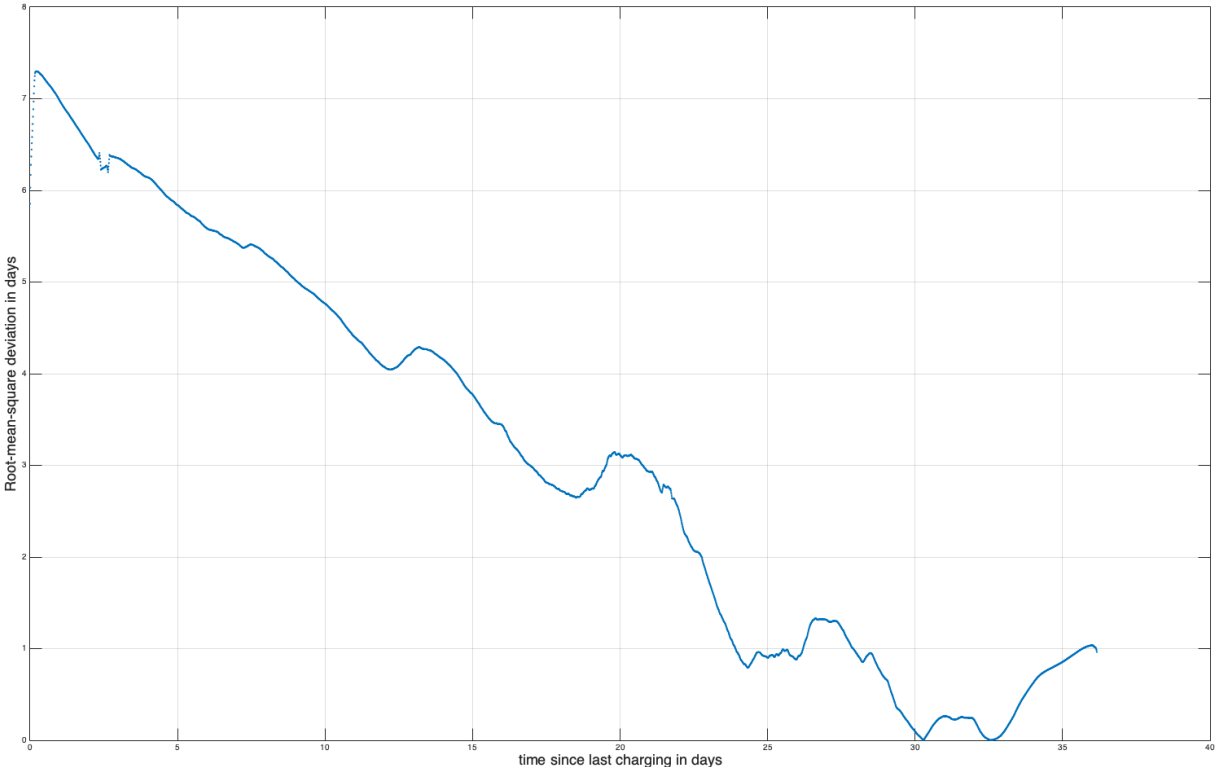


Figure 5.4: Root-mean-squared deviation

5.4 Webcam

To enable a webcam in the webservice an admin needs to add a webcam over the webcam management page, which can be accessed via the account page of the administrator user. After adding a webcam you get a primary and a secondary key for the webcam. You can use either of these as AuthenticationKey. This allows for a rolling key exchange.
To configure a webcam on a Debian/Ubuntu/Raspbian/Linux Mint you need to execute the following steps:

```
sudo apt install curl fswebcam
cat > capturePicture <<END \
fswebcam -r 1280x720 --jpeg 85 -D 1 -S 10 image.jpeg
curl -F "file=@image.jpeg" https://iot.nt.th-koeln.de/webcam/upload/<WebcamId> -H "
    ↪ Authorization: <AuthenticationKey>"
END
chmod +x capturePicture.sh
```

Maybe you need to modify the parameters for fswebcam, depending on your webcam. The pictures size, the delay until the picture recording should start and how many frames the webcam should run before a picture is taken. This is important because some webcams need time before their image is set correctly (exposure is wrong).

The script can then be executed with `./capturePicture.sh` to capture a frame of the webcam and send it to the webservice, where it can then be viewed later.
To execute the script periodically use crontab:

```
crontab -e
```

The add a new line to send a picture every hour:

```
0 * * * * <absolute path to the file capturePicture.sh>
```

5.5 Certificate

To guarantee a secure connection to the webservice this server uses SSL certificates. The process of generating and ordering a SSL certificate comprises multiple step where admin rights are needed.

5.5.1 Install openssl and generate a key pair

- install OpenSSL if necessary

  ```
  sudo apt-get install openssl
  ```

- generate a key pair with

  ```
  openssl genrsa -aes -out key.pem 2048
  ```

  where 2048 is the key length. The file `key.pem` contains the private and public key. A key file without password protection can be generated by leaving out the option `aes256`. However, the unprotected key file must be protected by other means, e.g. access restrictions.

5.5.2 Generation of an PKCS 10 Zertifikatrequests(CSR - Certificate Signing Request)

Now generate the PKCS 10 Zertifikatrequests (CSR - Certificate Signing Request) as .pem file by

```
openssl req -batch -sha256 -new -key key.pem -out request.pem -subj '/C=DE/ST=
    ↪ Nordrhein-Westfalen/L=Koeln/O=Technische Hochschule Koeln/OU=Fak7/CN=iot.
    ↪ nt.th-koeln.de/emailAddress=martin.seckler@th-koeln.de'
openssl req -new -key idp.example.org.key.pem -out idp example.org.csr.pem}
```

or

```
openssl req -new -key idp.example.org.key.pem -out idp example.org.csr.pem}
```

Then the parameters are prompted. Parameters are `O=Technische Hochschule Koeln,L=Koeln,ST=Nordrhein-Westfalen,C=DE iot.nt.th-koeln.de`
Organizational Unit can be left empty. `FQDN` is the full server name as `iot.nt.th-koeln.de` ,see `https://doku.tid.dfn.de/de:certificates` Key is the private key of the server, path may be necessary.

```
user@host:~\$ openssl req -new -key idp.example.org.key.pem -out idp.example.org.csr.pem
# Sie werden Folgendes abgefragt, evtl. befüllt mit Vorgaben aus Ihrer /etc/ssl/openssl.cnf
Country Name (2 letter code) [DE]:
State or Province Name (full name) []:            <== Bundesland
```

```
Locality Name (eg, city) []:                    <== Stadt
Organization Name (eg, company) []:             <== Einrichtung
Organizational Unit Name (eg, section) []:      <== ggf. Abteilung o.ä.
Common Name (eg, YOUR name) []:SERVERNAME       <== HIER DEN FQDN EINSETZEN!
A challenge password []:
An optional company name []:
```

Check the generated request file by:

```
\texttt{openssl req -in idp.example.org.pem -text -noout}
```

The result should look like:

```
Certificate Request:
Data:
Version: 0 (0x0)
Subject: C=DE, ST=Nordrhein-Westfalen, L=Koeln, O=Technische Hochschule Koeln, CN=iot.nt.th
```

5.5.3 Uploading the request.pem at the CampusIT

Load up the `request.pem` file at `https://pki.pca.dfn.de/dfn-ca-global-g2/cgi-bin/pub/pki?cmd=pkcs10_req;id=1;menu_item=2;XSEC=4f1fd046ff159defb4966e0d495959da555adc929503f44a813c1151c47940c1&RA_ID=1520`, fill in the form and print out the generated .pdf file. Then sign this application and load it up at the Campus IT address: `https://selfservice.th-koeln.de/tas/public/ssp/content/serviceflow?unid=9cac894a866544dfbcb0a858ad4a8a7f&from=8c3e5b66-1a76-4165-ade5-584fc7ddc1bc&openedFromService=true`

5.5.4 Receiving the server certificate and further steps

After processing the request an email is sent by the CAmpus IT containing the server certificate as a `.pem` file.
According to the manual of DFN to generate a PKCS#12 Datei with the private and public key, the corresponding server certificate and the CA Chain the following fiels must be present:

1. the RSA key strored in key.pem
2. the server certificate received from DFN (e.g. certificate.pem)
3. The CA-key chain from https://pki.pca.dfn.de/<Name Ihrer CA>/pub/cacert/chain.txt (file name: ca-chain.txt)

Use the following openssl command to generate a PKCS#12 file named pkcs12-file.p12

```
openssl pkcs12 -export -inkey key.pem -in certificate.pem certfile ca_chain.txt
    ↪ out pkcs12-file.p12
```

In the following the alternative way is described that we used to generate the PKCS#12 file that is used on our iot.nt.th-koeln.de web server.

5.5.5 Edit certificates

At this point we have the chain of certificates and our servercertificate. The next step is to edit the certificates in case of put the servercertificate at the top of the chain of certificates. To edit the certificates they can be opened with any texteditor and put together. The result is one file which contains four different certificates. This output file has to be saved as `servercert.pem`.

5.5.6 Push certificate to Webserver

The last step is to push the certificate to the webserver. To push the servercertificate to the webserver we use the scp command. So you have to open an terminal window and navigate to the direction of the 'servercert.pem' file and entering the following command.

```
scp servercert.pem username@iot.nt.th-koeln.de:/home/username
```

After typing in your password the file would be transferd to the given direction.
Now we have to move the 'servercert.pem' file to the correct direction on the webserver. To connect to the webserver the following ssh command can be used.

```
ssh username@iot.nt.th-koeln.de
```

After entering this command the server asks for a password. When you connected to the webserver successfully you can navigate to your home directory with the following command.

```
cd /home/username
```

Moving the 'servercert.pem' file to the correct direction can be done with the following command.

```
mv servercert.pem /home/dlohmann/RhizoTech/Cert
```

Just in case that the 'servercert.pem' file already exists in the target direction the file can be renamed with the following command after navigated to the target direction.

```
mv servercert.pem servercert_old.pem
```

When all this changes are done successfully the server services apache2 and mosquitto have to be restarted. Because the services can only be restarted with root permissions we have to type in the following command and confirm with your password.

```
sudo -i
```

Now you can restart the services with the following commands.

```
sudo service apache2 restart
sudo service mosquitto restart
```

6 Lizenzen

Folgende Softwarebibliotheken werden in der Software genutzt:

6.1 ST-Mircoelectronics - HAL (Hardware Abstraction Layer)

Die Lizenz für die STM-HAL Biliothek ist wie folgt beschrieben. In jeder Codedatei in der code von STM steht muss diese Lizenz wiederholt werden:

```
Copyright (c) 2019 STMicroelectronics International N.V.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted, provided that the following conditions are met:

1. Redistribution of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of other
   contributors to this software may be used to endorse or promote products
   derived from this software without specific written permission.
4. This software, including modifications and/or derivative works of this
   software, must execute solely and exclusively on microcontroller or
   microprocessor devices manufactured by or for STMicroelectronics.
5. Redistribution and use of this software other than as permitted under
   this license is void and will automatically terminate your rights under
   this license.

THIS SOFTWARE IS PROVIDED BY STMICROELECTRONICS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS, IMPLIED OR STATUTORY WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY
RIGHTS ARE DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. IN NO EVENT
SHALL STMICROELECTRONICS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.2 ESP-IDF

Die ESP-IDF ist unter die Apache 2.0 Lizenz gestellt. Weitere Firmware Komponenten stehen unter anderen Lizenzen: `https://docs.espressif.com/projects/esp-idf/en/latest/COPYRIGHT`.

`html`

6.3 SystemWorkbench for STM32

6.4 GoogleTest

Die Lizenz für GoogleTest ist wie folgt:

```
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
    * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.5 Python

Die Python Lizenz kann unter folgendem Link eingesehen werden: `https://docs.python.org/3/license.html#psf-license-agreement-for-python-release`

6.6 Mosquitto

Mosquitto ist lizenziert unter der EPL (Eclipse Public License 1.0) und der EDL (Eclipse Distribution License 1.0)

```
Eclipse Public License - v 1.0
```

```
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
    ↪ LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE
    ↪ PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
    ↪ distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
    ↪ distributed by that particular Contributor. A Contribution 'originates'
    ↪ from a Contributor if it was added to the Program by such Contributor
    ↪ itself or anyone acting on such Contributor's behalf. Contributions do
    ↪ not include additions to the Program which: (i) are separate modules of
    ↪ software distributed in conjunction with the Program under their own
    ↪ license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are
    ↪ necessarily infringed by the use or sale of its Contribution alone or
    ↪ when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
    ↪ including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free copyright license to
    ↪ reproduce, prepare derivative works of, publicly display, publicly
    ↪ perform, distribute and sublicense the Contribution of such Contributor,
    ↪ if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free patent license under
    ↪ Licensed Patents to make, use, sell, offer to sell, import and otherwise
    ↪ transfer the Contribution of such Contributor, if any, in source code and
    ↪  object code form. This patent license shall apply to the combination of
    ↪ the Contribution and the Program if, at the time the Contribution is
    ↪ added by the Contributor, such addition of the Contribution causes such
    ↪ combination to be covered by the Licensed Patents. The patent license
    ↪ shall not apply to any other combinations which include the Contribution.
    ↪  No hardware per se is licensed hereunder.
```

```
c) Recipient understands that although each Contributor grants the licenses to
    ↪ its Contributions set forth herein, no assurances are provided by any
    ↪ Contributor that the Program does not infringe the patent or other
    ↪ intellectual property rights of any other entity. Each Contributor
    ↪ disclaims any liability to Recipient for claims brought by any other
    ↪ entity based on infringement of intellectual property rights or otherwise
    ↪ . As a condition to exercising the rights and licenses granted hereunder,
    ↪  each Recipient hereby assumes sole responsibility to secure any other
    ↪ intellectual property rights needed, if any. For example, if a third
    ↪ party patent license is required to allow Recipient to distribute the
    ↪ Program, it is Recipient's responsibility to acquire that license before
    ↪ distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright
    ↪  rights in its Contribution, if any, to grant the copyright license set
    ↪ forth in this Agreement.
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its
    ↪  own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and
    ↪ conditions, express and implied, including warranties or conditions of
    ↪ title and non-infringement, and implied warranties or conditions of
    ↪ merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages
    ↪ , including direct, indirect, special, incidental and consequential
    ↪ damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by
    ↪ that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor,
    ↪ and informs licensees how to obtain it in a reasonable manner on or
    ↪ through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the
    ↪ Program.

Each Contributor must identify itself as the originator of its Contribution, if
    ↪ any, in a manner that reasonably allows subsequent Recipients to identify
    ↪  the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
    ↪ respect to end users, business partners and the like. While this license
```

```
    ↪ is intended to facilitate the commercial use of the Program, the
    ↪ Contributor who includes the Program in a commercial product offering
    ↪ should do so in a manner which does not create potential liability for
    ↪ other Contributors. Therefore, if a Contributor includes the Program in a
    ↪  commercial product offering, such Contributor ("Commercial Contributor")
    ↪  hereby agrees to defend and indemnify every other Contributor ("
    ↪ Indemnified Contributor") against any losses, damages and costs (
    ↪ collectively "Losses") arising from claims, lawsuits and other legal
    ↪ actions brought by a third party against the Indemnified Contributor to
    ↪ the extent caused by the acts or omissions of such Commercial Contributor
    ↪  in connection with its distribution of the Program in a commercial
    ↪ product offering. The obligations in this section do not apply to any
    ↪ claims or Losses relating to any actual or alleged intellectual property
    ↪ infringement. In order to qualify, an Indemnified Contributor must: a)
    ↪ promptly notify the Commercial Contributor in writing of such claim, and
    ↪ b) allow the Commercial Contributor to control, and cooperate with the
    ↪ Commercial Contributor in, the defense and any related settlement
    ↪ negotiations. The Indemnified Contributor may participate in any such
    ↪ claim at its own expense.

For example, a Contributor might include the Program in a commercial product
    ↪ offering, Product X. That Contributor is then a Commercial Contributor.
    ↪ If that Commercial Contributor then makes performance claims, or offers
    ↪ warranties related to Product X, those performance claims and warranties
    ↪ are such Commercial Contributor's responsibility alone. Under this
    ↪ section, the Commercial Contributor would have to defend claims against
    ↪ the other Contributors related to those performance claims and warranties
    ↪ , and if a court requires any other Contributor to pay any damages as a
    ↪ result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "
    ↪ AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER
    ↪ EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR
    ↪ CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A
    ↪ PARTICULAR PURPOSE. Each Recipient is solely responsible for determining
    ↪ the appropriateness of using and distributing the Program and assumes all
    ↪  risks associated with its exercise of rights under this Agreement ,
    ↪ including but not limited to the risks and costs of program errors,
    ↪ compliance with applicable laws, damage to or loss of data, programs or
    ↪ equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
    ↪ CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT,
    ↪ INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING
    ↪ WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF
```

```
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
    ↪ DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
    ↪ HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable
    ↪ law, it shall not affect the validity or enforceability of the remainder
    ↪ of the terms of this Agreement, and without further action by the parties
    ↪  hereto, such provision shall be reformed to the minimum extent necessary
    ↪  to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-
    ↪ claim or counterclaim in a lawsuit) alleging that the Program itself (
    ↪ excluding combinations of the Program with other software or hardware)
    ↪ infringes such Recipient's patent(s), then such Recipient's rights
    ↪ granted under Section 2(b) shall terminate as of the date such litigation
    ↪  is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
    ↪ comply with any of the material terms or conditions of this Agreement and
    ↪  does not cure such failure in a reasonable period of time after becoming
    ↪  aware of such noncompliance. If all Recipient's rights under this
    ↪ Agreement terminate, Recipient agrees to cease use and distribution of
    ↪ the Program as soon as reasonably practicable. However, Recipient's
    ↪ obligations under this Agreement and any licenses granted by Recipient
    ↪ relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
    ↪ order to avoid inconsistency the Agreement is copyrighted and may only be
    ↪  modified in the following manner. The Agreement Steward reserves the
    ↪ right to publish new versions (including revisions) of this Agreement
    ↪ from time to time. No one other than the Agreement Steward has the right
    ↪ to modify this Agreement. The Eclipse Foundation is the initial Agreement
    ↪  Steward. The Eclipse Foundation may assign the responsibility to serve
    ↪ as the Agreement Steward to a suitable separate entity. Each new version
    ↪ of the Agreement will be given a distinguishing version number. The
    ↪ Program (including Contributions) may always be distributed subject to
    ↪ the version of the Agreement under which it was received. In addition,
    ↪ after a new version of the Agreement is published, Contributor may elect
    ↪ to distribute the Program (including its Contributions) under the new
    ↪ version. Except as expressly stated in Sections 2(a) and 2(b) above,
    ↪ Recipient receives no rights or licenses to the intellectual property of
    ↪ any Contributor under this Agreement, whether expressly, by implication,
    ↪ estoppel or otherwise. All rights in the Program not expressly granted
    ↪ under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
```

```
    ↪ intellectual property laws of the United States of America. No party to
    ↪ this Agreement will bring a legal action under this Agreement more than
    ↪ one year after the cause of action arose. Each party waives its rights to
    ↪  a jury trial in any resulting litigation.
```

```
Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
    ↪  are permitted provided that the following conditions are met:

    Redistributions of source code must retain the above copyright notice, this
        ↪ list of conditions and the following disclaimer.
    Redistributions in binary form must reproduce the above copyright notice,
        ↪ this list of conditions and the following disclaimer in the
        ↪ documentation and/or other materials provided with the distribution.
    Neither the name of the Eclipse Foundation, Inc. nor the names of its
        ↪ contributors may be used to endorse or promote products derived from
        ↪ this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
    ↪ ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
    ↪ IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
    ↪ PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
    ↪ CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
    ↪ EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
    ↪ PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
    ↪ PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
    ↪ SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.7 Mosquitto-Go-Auth

```
MIT License

Copyright (c) 2018 Ignacio Gomez

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

6.8 OpenVPN

OpenVPN steht unter der GNU GENERAL PUBLIC LICENSE.

6.9 MQTT-sn Gateway

Das MQTT-sn Gateway ist auf GitHub zu finden `https://github.com/eclipse/paho.mqtt-sn.embedded-c`. Das Paho Project Lizenziert seine Software unter der EPL (Eclipse Public License - v 1.0) und der EDL (Eclipse Distribution License - v 1.0).

```
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
    ↪ LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE
    ↪ PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
    ↪ distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
    ↪ distributed by that particular Contributor. A Contribution 'originates'
    ↪ from a Contributor if it was added to the Program by such Contributor
    ↪ itself or anyone acting on such Contributor's behalf. Contributions do
    ↪ not include additions to the Program which: (i) are separate modules of
    ↪ software distributed in conjunction with the Program under their own
    ↪ license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are
    ↪ necessarily infringed by the use or sale of its Contribution alone or
    ↪ when combined with the Program.
```

```
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
    ↪ including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free copyright license to
    ↪ reproduce, prepare derivative works of, publicly display, publicly
    ↪ perform, distribute and sublicense the Contribution of such Contributor,
    ↪ if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free patent license under
    ↪ Licensed Patents to make, use, sell, offer to sell, import and otherwise
    ↪ transfer the Contribution of such Contributor, if any, in source code and
    ↪  object code form. This patent license shall apply to the combination of
    ↪ the Contribution and the Program if, at the time the Contribution is
    ↪ added by the Contributor, such addition of the Contribution causes such
    ↪ combination to be covered by the Licensed Patents. The patent license
    ↪ shall not apply to any other combinations which include the Contribution.
    ↪  No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to
    ↪ its Contributions set forth herein, no assurances are provided by any
    ↪ Contributor that the Program does not infringe the patent or other
    ↪ intellectual property rights of any other entity. Each Contributor
    ↪ disclaims any liability to Recipient for claims brought by any other
    ↪ entity based on infringement of intellectual property rights or otherwise
    ↪ . As a condition to exercising the rights and licenses granted hereunder,
    ↪  each Recipient hereby assumes sole responsibility to secure any other
    ↪ intellectual property rights needed, if any. For example, if a third
    ↪ party patent license is required to allow Recipient to distribute the
    ↪ Program, it is Recipient's responsibility to acquire that license before
    ↪ distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright
    ↪  rights in its Contribution, if any, to grant the copyright license set
    ↪ forth in this Agreement.
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its
    ↪  own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and
    ↪ conditions, express and implied, including warranties or conditions of
    ↪ title and non-infringement, and implied warranties or conditions of
    ↪ merchantability and fitness for a particular purpose;
```

```
ii) effectively excludes on behalf of all Contributors all liability for damages
    ↪ , including direct, indirect, special, incidental and consequential
    ↪ damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by
    ↪ that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor,
    ↪ and informs licensees how to obtain it in a reasonable manner on or
    ↪ through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the
    ↪ Program.

Each Contributor must identify itself as the originator of its Contribution, if
    ↪ any, in a manner that reasonably allows subsequent Recipients to identify
    ↪  the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
    ↪ respect to end users, business partners and the like. While this license
    ↪ is intended to facilitate the commercial use of the Program, the
    ↪ Contributor who includes the Program in a commercial product offering
    ↪ should do so in a manner which does not create potential liability for
    ↪ other Contributors. Therefore, if a Contributor includes the Program in a
    ↪  commercial product offering, such Contributor ("Commercial Contributor")
    ↪  hereby agrees to defend and indemnify every other Contributor ("
    ↪ Indemnified Contributor") against any losses, damages and costs (
    ↪ collectively "Losses") arising from claims, lawsuits and other legal
    ↪ actions brought by a third party against the Indemnified Contributor to
    ↪ the extent caused by the acts or omissions of such Commercial Contributor
    ↪  in connection with its distribution of the Program in a commercial
    ↪ product offering. The obligations in this section do not apply to any
    ↪ claims or Losses relating to any actual or alleged intellectual property
    ↪ infringement. In order to qualify, an Indemnified Contributor must: a)
    ↪ promptly notify the Commercial Contributor in writing of such claim, and
    ↪ b) allow the Commercial Contributor to control, and cooperate with the
    ↪ Commercial Contributor in, the defense and any related settlement
    ↪ negotiations. The Indemnified Contributor may participate in any such
    ↪ claim at its own expense.

For example, a Contributor might include the Program in a commercial product
    ↪ offering, Product X. That Contributor is then a Commercial Contributor.
    ↪ If that Commercial Contributor then makes performance claims, or offers
    ↪ warranties related to Product X, those performance claims and warranties
    ↪ are such Commercial Contributor's responsibility alone. Under this
    ↪ section, the Commercial Contributor would have to defend claims against
```

```
    ↪ the other Contributors related to those performance claims and warranties
    ↪ , and if a court requires any other Contributor to pay any damages as a
    ↪ result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "
    ↪ AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER
    ↪ EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR
    ↪ CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A
    ↪ PARTICULAR PURPOSE. Each Recipient is solely responsible for determining
    ↪ the appropriateness of using and distributing the Program and assumes all
    ↪  risks associated with its exercise of rights under this Agreement ,
    ↪ including but not limited to the risks and costs of program errors,
    ↪ compliance with applicable laws, damage to or loss of data, programs or
    ↪ equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
    ↪ CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT,
    ↪ INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING
    ↪ WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
    ↪ DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
    ↪ HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable
    ↪ law, it shall not affect the validity or enforceability of the remainder
    ↪ of the terms of this Agreement, and without further action by the parties
    ↪  hereto, such provision shall be reformed to the minimum extent necessary
    ↪  to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-
    ↪ claim or counterclaim in a lawsuit) alleging that the Program itself (
    ↪ excluding combinations of the Program with other software or hardware)
    ↪ infringes such Recipient's patent(s), then such Recipient's rights
    ↪ granted under Section 2(b) shall terminate as of the date such litigation
    ↪  is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
    ↪ comply with any of the material terms or conditions of this Agreement and
    ↪  does not cure such failure in a reasonable period of time after becoming
    ↪  aware of such noncompliance. If all Recipient's rights under this
    ↪ Agreement terminate, Recipient agrees to cease use and distribution of
    ↪ the Program as soon as reasonably practicable. However, Recipient's
```

```
    ↪ obligations under this Agreement and any licenses granted by Recipient
    ↪ relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
    ↪ order to avoid inconsistency the Agreement is copyrighted and may only be
    ↪  modified in the following manner. The Agreement Steward reserves the
    ↪ right to publish new versions (including revisions) of this Agreement
    ↪ from time to time. No one other than the Agreement Steward has the right
    ↪ to modify this Agreement. The Eclipse Foundation is the initial Agreement
    ↪  Steward. The Eclipse Foundation may assign the responsibility to serve
    ↪ as the Agreement Steward to a suitable separate entity. Each new version
    ↪ of the Agreement will be given a distinguishing version number. The
    ↪ Program (including Contributions) may always be distributed subject to
    ↪ the version of the Agreement under which it was received. In addition,
    ↪ after a new version of the Agreement is published, Contributor may elect
    ↪ to distribute the Program (including its Contributions) under the new
    ↪ version. Except as expressly stated in Sections 2(a) and 2(b) above,
    ↪ Recipient receives no rights or licenses to the intellectual property of
    ↪ any Contributor under this Agreement, whether expressly, by implication,
    ↪ estoppel or otherwise. All rights in the Program not expressly granted
    ↪ under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
    ↪ intellectual property laws of the United States of America. No party to
    ↪ this Agreement will bring a legal action under this Agreement more than
    ↪ one year after the cause of action arose. Each party waives its rights to
    ↪  a jury trial in any resulting litigation.
```

```
Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
    ↪  are permitted provided that the following conditions are met:

    Redistributions of source code must retain the above copyright notice, this
        ↪ list of conditions and the following disclaimer.
    Redistributions in binary form must reproduce the above copyright notice,
        ↪ this list of conditions and the following disclaimer in the
        ↪ documentation and/or other materials provided with the distribution.
    Neither the name of the Eclipse Foundation, Inc. nor the names of its
        ↪ contributors may be used to endorse or promote products derived from
        ↪ this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
    ↪ ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

```
↪ IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
↪ PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
↪ CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
↪ EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
↪ PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
↪ PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
↪ SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.10 Webservice

Folgende Softwarebibliotheken werden im Webservice genutzt:

- Python
- Flask
  - flask
  - flask-login
  - flask-sqlalchemy
  - flask-bcrypt
  - flask-wtf
- paho-mqtt
- pandas
- numpy
- SQLite
- Bootstrap
- chart.js
- popper.js
- moment.js
- jquery.js
- tail.DateTime
- EventSource Polyfill
- Vis
- ffmpeg
- x264
- opencv-python

6.10.1 Python

Die Python Lizenz kann unter folgendem Link eingesehen werden: `https://docs.python.org/3/license.html#psf-license-agreement-for-python-release`

6.10.2 Flask

6.10.2.1 Flask

Die Lizenz ist unter folgendem Link zu finden: `http://flask.pocoo.org/docs/1.0/license/#flask-license`

6.10.2.2 Flask-Login

Die Lizenz ist unter folgendem Link zu finden: https://github.com/maxcountryman/flask-login/blob/master/LICENSE

6.10.2.3 Flask-SQLAlchemy

Nutzt die selbe Lizenz wir Flask: **?? ??**

6.10.2.4 Flask-Bcrypt

Die Lizenz ist unter folgendem Link zu finden: https://github.com/maxcountryman/flask-bcrypt/blob/master/LICENSE

6.10.2.5 Flask-Wtf

Flask-Wtf benutzt die BSD-Lizenz: https://flask-wtf.readthedocs.io/en/stable/license.html

6.10.3 paho-mqtt

Eclipse Public License v1.0 - https://www.eclipse.org/legal/epl-v10.html
Eclipse Distribution License v1.0 - https://www.eclipse.org/org/documents/edl-v10.html

```
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
    ↪ LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE
    ↪ PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
    ↪ distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
    ↪ distributed by that particular Contributor. A Contribution 'originates'
    ↪ from a Contributor if it was added to the Program by such Contributor
    ↪ itself or anyone acting on such Contributor's behalf. Contributions do
    ↪ not include additions to the Program which: (i) are separate modules of
    ↪ software distributed in conjunction with the Program under their own
    ↪ license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are
    ↪ necessarily infringed by the use or sale of its Contribution alone or
    ↪ when combined with the Program.
```

```
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
    ↪ including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free copyright license to
    ↪ reproduce, prepare derivative works of, publicly display, publicly
    ↪ perform, distribute and sublicense the Contribution of such Contributor,
    ↪ if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
    ↪ Recipient a non-exclusive, worldwide, royalty-free patent license under
    ↪ Licensed Patents to make, use, sell, offer to sell, import and otherwise
    ↪ transfer the Contribution of such Contributor, if any, in source code and
    ↪  object code form. This patent license shall apply to the combination of
    ↪ the Contribution and the Program if, at the time the Contribution is
    ↪ added by the Contributor, such addition of the Contribution causes such
    ↪ combination to be covered by the Licensed Patents. The patent license
    ↪ shall not apply to any other combinations which include the Contribution.
    ↪  No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to
    ↪ its Contributions set forth herein, no assurances are provided by any
    ↪ Contributor that the Program does not infringe the patent or other
    ↪ intellectual property rights of any other entity. Each Contributor
    ↪ disclaims any liability to Recipient for claims brought by any other
    ↪ entity based on infringement of intellectual property rights or otherwise
    ↪ . As a condition to exercising the rights and licenses granted hereunder,
    ↪  each Recipient hereby assumes sole responsibility to secure any other
    ↪ intellectual property rights needed, if any. For example, if a third
    ↪ party patent license is required to allow Recipient to distribute the
    ↪ Program, it is Recipient's responsibility to acquire that license before
    ↪ distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright
    ↪  rights in its Contribution, if any, to grant the copyright license set
    ↪ forth in this Agreement.
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its
    ↪  own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and
    ↪ conditions, express and implied, including warranties or conditions of
    ↪ title and non-infringement, and implied warranties or conditions of
    ↪ merchantability and fitness for a particular purpose;
```

```
ii) effectively excludes on behalf of all Contributors all liability for damages
    ↪ , including direct, indirect, special, incidental and consequential
    ↪ damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by
    ↪ that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor,
    ↪ and informs licensees how to obtain it in a reasonable manner on or
    ↪ through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the
    ↪ Program.

Each Contributor must identify itself as the originator of its Contribution, if
    ↪ any, in a manner that reasonably allows subsequent Recipients to identify
    ↪  the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
    ↪ respect to end users, business partners and the like. While this license
    ↪ is intended to facilitate the commercial use of the Program, the
    ↪ Contributor who includes the Program in a commercial product offering
    ↪ should do so in a manner which does not create potential liability for
    ↪ other Contributors. Therefore, if a Contributor includes the Program in a
    ↪  commercial product offering, such Contributor ("Commercial Contributor")
    ↪  hereby agrees to defend and indemnify every other Contributor ("
    ↪ Indemnified Contributor") against any losses, damages and costs (
    ↪ collectively "Losses") arising from claims, lawsuits and other legal
    ↪ actions brought by a third party against the Indemnified Contributor to
    ↪ the extent caused by the acts or omissions of such Commercial Contributor
    ↪  in connection with its distribution of the Program in a commercial
    ↪ product offering. The obligations in this section do not apply to any
    ↪ claims or Losses relating to any actual or alleged intellectual property
    ↪ infringement. In order to qualify, an Indemnified Contributor must: a)
    ↪ promptly notify the Commercial Contributor in writing of such claim, and
    ↪ b) allow the Commercial Contributor to control, and cooperate with the
    ↪ Commercial Contributor in, the defense and any related settlement
    ↪ negotiations. The Indemnified Contributor may participate in any such
    ↪ claim at its own expense.

For example, a Contributor might include the Program in a commercial product
    ↪ offering, Product X. That Contributor is then a Commercial Contributor.
    ↪ If that Commercial Contributor then makes performance claims, or offers
    ↪ warranties related to Product X, those performance claims and warranties
    ↪ are such Commercial Contributor's responsibility alone. Under this
    ↪ section, the Commercial Contributor would have to defend claims against
```

```
    ↪ the other Contributors related to those performance claims and warranties
    ↪ , and if a court requires any other Contributor to pay any damages as a
    ↪ result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "
    ↪ AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER
    ↪ EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR
    ↪ CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A
    ↪ PARTICULAR PURPOSE. Each Recipient is solely responsible for determining
    ↪ the appropriateness of using and distributing the Program and assumes all
    ↪  risks associated with its exercise of rights under this Agreement ,
    ↪ including but not limited to the risks and costs of program errors,
    ↪ compliance with applicable laws, damage to or loss of data, programs or
    ↪ equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
    ↪ CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT,
    ↪ INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING
    ↪ WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
    ↪ DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
    ↪ HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable
    ↪ law, it shall not affect the validity or enforceability of the remainder
    ↪ of the terms of this Agreement, and without further action by the parties
    ↪  hereto, such provision shall be reformed to the minimum extent necessary
    ↪  to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-
    ↪ claim or counterclaim in a lawsuit) alleging that the Program itself (
    ↪ excluding combinations of the Program with other software or hardware)
    ↪ infringes such Recipient's patent(s), then such Recipient's rights
    ↪ granted under Section 2(b) shall terminate as of the date such litigation
    ↪  is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
    ↪ comply with any of the material terms or conditions of this Agreement and
    ↪  does not cure such failure in a reasonable period of time after becoming
    ↪  aware of such noncompliance. If all Recipient's rights under this
    ↪ Agreement terminate, Recipient agrees to cease use and distribution of
    ↪ the Program as soon as reasonably practicable. However, Recipient's
```

```
    ↪ obligations under this Agreement and any licenses granted by Recipient
    ↪ relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
    ↪ order to avoid inconsistency the Agreement is copyrighted and may only be
    ↪  modified in the following manner. The Agreement Steward reserves the
    ↪ right to publish new versions (including revisions) of this Agreement
    ↪ from time to time. No one other than the Agreement Steward has the right
    ↪ to modify this Agreement. The Eclipse Foundation is the initial Agreement
    ↪  Steward. The Eclipse Foundation may assign the responsibility to serve
    ↪ as the Agreement Steward to a suitable separate entity. Each new version
    ↪ of the Agreement will be given a distinguishing version number. The
    ↪ Program (including Contributions) may always be distributed subject to
    ↪ the version of the Agreement under which it was received. In addition,
    ↪ after a new version of the Agreement is published, Contributor may elect
    ↪ to distribute the Program (including its Contributions) under the new
    ↪ version. Except as expressly stated in Sections 2(a) and 2(b) above,
    ↪ Recipient receives no rights or licenses to the intellectual property of
    ↪ any Contributor under this Agreement, whether expressly, by implication,
    ↪ estoppel or otherwise. All rights in the Program not expressly granted
    ↪ under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
    ↪ intellectual property laws of the United States of America. No party to
    ↪ this Agreement will bring a legal action under this Agreement more than
    ↪ one year after the cause of action arose. Each party waives its rights to
    ↪  a jury trial in any resulting litigation.
```

```
Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
    ↪  are permitted provided that the following conditions are met:

    Redistributions of source code must retain the above copyright notice, this
        ↪ list of conditions and the following disclaimer.
    Redistributions in binary form must reproduce the above copyright notice,
        ↪ this list of conditions and the following disclaimer in the
        ↪ documentation and/or other materials provided with the distribution.
    Neither the name of the Eclipse Foundation, Inc. nor the names of its
        ↪ contributors may be used to endorse or promote products derived from
        ↪ this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
    ↪ ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

```
    ↪ IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
    ↪ PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
    ↪ CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
    ↪ EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
    ↪ PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
    ↪ PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
    ↪ SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.10.4 pandas

Pandas steht unter folgender Lizenz:

```
BSD 3-Clause License

Copyright (c) 2008-2012, AQR Capital Management, LLC, Lambda Foundry, Inc. and
    ↪ PyData Development Team
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.10.5 NumPy

NumPy steht unter folgender Lizenz:

```
Copyright (c) 2005-2019, NumPy Developers.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
    ↪  are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice, this
    ↪ list of conditions and the following disclaimer.

 * Redistributions in binary form must reproduce the above copyright notice,
    ↪ this list of conditions and the following disclaimer in the
    ↪ documentation and/or other materials provided with the distribution.

 * Neither the name of the NumPy Developers nor the names of any contributors
    ↪ may be used to endorse or promote products derived from this software
    ↪ without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
    ↪ ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
    ↪ IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
    ↪ PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
    ↪ CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
    ↪ EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
    ↪ PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
    ↪ PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
    ↪ LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
    ↪ NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
    ↪ SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

6.10.6 SQLite

SQLite steht unter folgender Lizenz:

```
# SQLite Is Public Domain

All of the code and documentation in SQLite has been dedicated to the public
    ↪ domain by the authors. All code authors, and representatives of the
    ↪ companies they work for, have signed affidavits dedicating their
    ↪ contributions to the public domain and originals of those signed
    ↪ affidavits are stored in a firesafe at the main offices of Hwaci. Anyone
    ↪ is free to copy, modify, publish, use, compile, sell, or distribute the
    ↪ original SQLite code, either in source code form or as a compiled binary,
    ↪  for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code and documentation in
    ↪ SQLite - those parts of the SQLite library that you actually bundle and
    ↪ ship with a larger application. Some scripts used as part of the build
    ↪ process (for example the "configure" scripts generated by autoconf) might
    ↪  fall under other open-source licenses. Nothing from these build scripts
    ↪ ever reaches the final deliverable SQLite library, however, and so the
```

```
    ↪ licenses associated with those scripts should not be a factor in
    ↪ assessing your rights to copy and use the SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has
    ↪  been taken from other projects or from the open internet. Every line of
    ↪ code can be traced back to its original author, and all of those authors
    ↪ have public domain dedications on file. So the SQLite code base is clean
    ↪ and is uncontaminated with licensed code from other projects.

# Open-Source, not Open-Contribution

SQLite is open-source, meaning that you can make as many copies of it as you
    ↪ want and do whatever you want with those copies, without limitation. But
    ↪ SQLite is not open-contribution. In order to keep SQLite in the public
    ↪ domain and ensure that the code does not become contaminated with
    ↪ proprietary or licensed content, the project does not accept patches from
    ↪  unknown persons.

All of the code in SQLite is original, having been written specifically for use
    ↪ by SQLite. No code has been copied from unknown sources on the internet.

# Warranty of Title

SQLite is in the public domain and does not require a license. Even so, some
    ↪ organizations want legal proof of their right to use SQLite.
    ↪ Circumstances where this occurs include the following:

 * Your company desires indemnity against claims of copyright infringement.
 * You are using SQLite in a jurisdiction that does not recognize the public
     ↪ domain.
 * You are using SQLite in a jurisdiction that does not recognize the right of
     ↪ an author to dedicate their work to the public domain.
 * You want to hold a tangible legal document as evidence that you have the
     ↪ legal right to use and distribute SQLite.
 * Your legal department tells you that you have to purchase a license.

If any of the above circumstances apply to you, Hwaci, the company that employs
    ↪ all the developers of SQLite, will sell you a Warranty of Title for
    ↪ SQLite. A Warranty of Title is a legal document that asserts that the
    ↪ claimed authors of SQLite are the true authors, and that the authors have
    ↪  the legal right to dedicate the SQLite to the public domain, and that
    ↪ Hwaci will vigorously defend against challenges to those claims. All
    ↪ proceeds from the sale of SQLite Warranties of Title are used to fund
    ↪ continuing improvement and support of SQLite.

# Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, the
    ↪ project does not accept patches. If you would like to make a suggested
```

```
    ↪ change, and include a patch as a proof-of-concept, that would be great.
    ↪ However please do not be offended if we rewrite your patch from scratch.
```

6.10.7 Bootstrap 4

Bootstrap steht unter der MIT Lizenz:

```
The MIT License (MIT)

Copyright (c) 2011-2018 Twitter, Inc.
Copyright (c) 2011-2018 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

```
Bootstrap is released under the MIT license and is copyright 2018 Twitter.
    ↪ Boiled down to smaller chunks, it can be described with the following
    ↪ conditions.

# It requires you to:

 * Keep the license and copyright notice included in Bootstrap's CSS and
     ↪ JavaScript files when you use them in your works

# It permits you to:

 * Freely download and use Bootstrap, in whole or in part, for personal, private
     ↪ , company internal, or commercial purposes
 * Use Bootstrap in packages or distributions that you create
 * Modify the source code
 * Grant a sublicense to modify and distribute Bootstrap to third parties not
     ↪ included in the license

 # It forbids you to:
```

```
 * Hold the authors and license owners liable for damages as Bootstrap is
    ↪ provided without warranty
 * Hold the creators or copyright holders of Bootstrap liable
 * Redistribute any piece of Bootstrap without proper attribution
 * Use any marks owned by Twitter in any way that might state or imply that
    ↪ Twitter endorses your distribution
 * Use any marks owned by Twitter in any way that might state or imply that you
    ↪ created the Twitter software in question

# It does not require you to:

    Include the source of Bootstrap itself, or of any modifications you may have
       ↪  made to it, in any redistribution you may assemble that includes it
    Submit changes that you make to Bootstrap back to the Bootstrap project (
       ↪ though such feedback is encouraged)

The full Bootstrap license is located in the project [repository](https://github
    ↪ .com/twbs/bootstrap/blob/v4.0.0/LICENSE) for more information.
```

6.10.8 chart.js

ChartJs wird unter der MIT-Lizenz zur verfügung gestellt: `https://www.chartjs.org/docs/latest/notes/license.html`

6.10.9 popper.js

Popper.js wird unter der MIT Lizenz verteilt.

```
The MIT License (MIT)
=====================

Copyright 2016 Federico Zivolo and contributors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
```

```
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

6.10.10 moment.js

Moment.js wird unter der MIT Lizenz verteilt.

```
Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

6.10.11 jquery.js

Die jquery Lizenz ist unter folgendem link einzusehen: `https://jquery.org/license/`

6.10.12 tail.DateTime

Die javascript bibliothek tail.DateTime ist unter der MIT lizenz veröffentlicht.

```
MIT License

Copyright (c) 2018 SamBrishes (pytesNET) AND Josef Broz (PureJSCalendar)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

6.10.13 EventSource Polyfill

Quelle: `https://github.com/amvtek/EventSource`
Das Polyfill wurde unter der MIT Lizenz veröffentlicht:

```
The MIT License (MIT)

Copyright (c) 2014 AmvTek

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

6.10.14 Vis

TODO

6.10.15 ffmpeg

FFmpeg is licensed under the GNU Lesser General Public License (LGPL) version 2.1 or later. However, FFmpeg incorporates several optional parts and optimizations that are covered by the GNU General Public License (GPL) version 2 or later. If those parts get used the GPL applies to all of FFmpeg.

Read the license texts to learn how this affects programs built on top of FFmpeg or reusing FFmpeg. You may also wish to have a look at the GPL FAQ.
Note that FFmpeg is not available under any other licensing terms, especially not proprietary/commercial ones, not even in exchange for payment.

6.10.16 x264

`https://www.videolan.org/developers/x264.html`
In addition to being free to use under the GNU GPL, x264 is also available under a commercial license. Contact x264licensing@videolan.org for more details.

6.10.17 opencv-python

License: MIT License (MIT)

7 Acronyms