

# Hardwareprojekt: Raspberry Pi

## Ziel des Hardwareprojekts

Sie sind in Gruppen aufgeteilt und die Verwaltung des Codes findet über GitHub statt.

Ziel ist es Temperaturmessdaten von einem Raspberry Pi über das MQTT-Netzwerkprotokoll bereitzustellen und auf anderen Systemen anzuzeigen. Auf dem Raspberry Pi wird ein angeschlossener Sensor mit einem Python-Script angesteuert und sendet die Messdaten alle drei Sekunden. Auf der Empfängerseite wird sowohl eine Python-GUI als auch eine Website als MQTT-Clients erstellt.

Auf beiden MQTT-Clients wird ein Plot dargestellt.

Die Art der Visualisierung wird selbst gewählt. Denkbar ist die Anzeige aller Werte auf Knopfdruck oder eine einem Oszilloskop vergleichbare Darstellung. Das GUI soll per Push-Button beendet werden, optional kann man die empfangenen Daten speichern.

## Hilfestellung

Es wird der Raspberry Pi 3B mit installiertem Raspbian bereitgestellt, dieser ist für das HAW-Netz freigeschaltet. Zusätzlich befindet sich auf diesem ein Datenserver. Zusätzlich erhalten Sie Beispielprogramme für die MQTT-Kommunikation und den Start eines Python-Webservers.

## Raspbian OS

Raspbian OS ist ein an den Raspberry Pi angepasstes Debian, eine verbreitete Linux-Distributionen. Viele Server und auch Embedded-Geräte nutzen Linux als Betriebssystem.

Der Großteil der Software, die Sie benötigen, ist bereits vorinstalliert. Nutzernamen und Passwort für den Raspberry lauten wie folgt: Name: pi, Passwort: raspberrz

Die Steuerung findet über die Konsole (BASH) statt. Wichtige Befehle hierfür sind:

Befehl	Bedeutung
ls <Pfad>	Listet die Dateien und Ordner im aktuellen Verzeichnis auf.
cd <Pfad>	Ändert das Verzeichnis in dem man sich befindet.
mkdir <Pfad>/Ordnername	Legt einen neuen Ordner an.
sudo <Befehl>	Den Befehl mit Administrator-Rechten ausführen.
ssh <User>@<Computer>	Stellt eine SSH-Verbindung her
screen -S <Jobname>	Startet eine Konsole im Hintergrund. Verlassen mit: STRG+A+D
screen -r <Jobname>	Kehrt zu der Konsole zurück.
STRG+C	Beendet den aktuell laufenden Befehl
date --set 'JJJJ-MM-DD hh:mm:ss'	Stellt die Uhr ein, nötig, da es keine CMOS-Uhr gibt

## MQTT-Server

Das MQTT-Protokoll ist gut für die Übertragung von Sensordaten geeignet. Es setzt das Publish-Subscriber-Model um. Der **Publisher** sendet die Daten an ein benanntes **Topic** auf den Server. Von dort aus wird dieses an alle **Subscriber** weitergeleitet, die das **Topic** abonniert haben. Verwenden Sie bitte den Gruppennamen/Gruppennummer als Namen des **Topics**.

Installation unter Linux: `apt install mosquitto-clients`

Unter Windows gibt es den [MQTT-Explorer](#)

Für Python gibt es z.B. das Modul [MQTT-Client](#)

Auf den Raspberry Pis läuft jeweils ein eigener MQTT-Server und ist unter dessen IP erreichbar.

Die Raspberrys haben einen Aufkleber mit der IP, die IPs beginnen mit 141.22.36.\*.

Testen lässt sich dieser indem man den MQTT-Client startet und auf alle Topics lauscht:

```
mosquitto_sub -h <IP_ADRESSE_RASPI> -t \#
```

Und auf das „Topic“ "`<Gruppenname>/Test`" sendet man mit einem anderen PC:

```
mosquitto_pub -h <IP_ADRESSE_RASPI> -t <Gruppenname>/Test -m "hello world"
```

Um den MQTT-Server lokal auf dem Windows-Rechner laufen zu lassen, gibt es auch für diesen den Mosquitto-Servers: <https://mosquitto.org/download/>

Diesen erreicht man über die IP des Rechners oder die Loopback Adresse (127.0.0.1).

## Wiederkehrende Abfragen

Sensorik sollte nicht ohne Pause in einer while-true-Schleife abgefragt und versendet werden, die nur durch die CPU-Geschwindigkeit begrenzt wird. Es ist sinnvoll zumindest abzuwarten, damit es nicht sehr viele Daten zu zufälligen Zeitpunkten gibt. Dafür kann man die Funktion `time.sleep(<SEKUNDEN>)` aus dem Modul `time` nutzen. Mit Zahlen im Bereich zwischen 0 und 1 aufgerufen, wartet diese im ms Bereich. In der fortgeschrittenen Programmierung wird mit Timern gearbeitet.

## SSH-Verbindung

Um an dem Raspberry Pi nicht mit Tastatur und Maus arbeiten zu müssen, kann man sich über das Netzwerk per SSH mit diesem verbinden. Hierüber können alle vorgestellten Linux-Befehle und die GIT-Befehle ausgeführt werden. Unter Linux ist es über X11-Forwarding möglich, dass Programme mit GUI ausgeführt werden. Unter Windows kann man für SSH die Powershell oder [Putty](#) nehmen.

## Starten eines Python-Scriptes über die Konsole

Sie erhalten drei Scripte mit Beispiel-Code: `readMQTT.py`, `sendMQTTShort.py` und `webserver.py`

Diese und andere, z.B. eigene Scripte, starten Sie wie folgt:

```
python <Dateiname.py>
```

Wenn Sie sich nicht in dem selben Ordner befinden, wie die Python-Datei, dann müssen Sie den Pfad angeben. Bei dem Pfad handelt es sich um den Ordner, in dem die Datei liegt, welche Sie ausführen möchten.

## Sensorik

Für den Anschluss ist es wichtig, dass Sie sich herausuchen, mit wie viel Volt der Sensor betrieben wird. Der Raspberry stellt 3,3V und 5V bereit.

Auch müssen Sie sich entscheiden, an welchen der GPIO-Pins das Datensignal des Sensors angeschlossen wird. Die verfügbaren Pins sind hier ganz gut zu erkennen:

<https://www.elektronik-kompodium.de/sites/raspberry-pi/1907101.htm>

Die Pins der Sensoren werden dann mit Verbindungskabeln mit den Pins der Raspberry-Steckerleiste verbunden. Am besten zuerst die Masse (GND), dann das Signal und am Ende die

Versorgungsspannung anschließen. Aktueller Code zur Ansteuerung der Sensorik findet sich auf der Herstellerseite: <https://sensorkit.joy-it.net/de/>

## Projekt-Daten mit dem Raspberry synchronisieren

Man kann das Textverarbeitungs-Programm "nano" verwenden und damit die Python-Scripte direkt auf dem Raspberry ändern. Es bietet sich an, ein GIT-Repository für die Synchronisation des Codes zu nutzen. Dadurch kann man auf einem Desktop-Rechner eine Entwicklungs-Umgebung nutzen und von dort aus die Änderungen in das GIT hochladen. Dann kann man sich per SSH mit dem Raspberry verbinden, mit "git clone" einmal das Projekt anlegen und dann die Änderungen jeweils mit "git pull" herunterladen. Bei GitHub kann hierfür ein Access-Token erzeugt werden.

## Arbeiten ohne Zugriff auf den Raspberry

Einen Großteil der Zeit können Sie ohne Raspberry arbeiten, da Sie hauptsächlich die Kommunikation per MQTT verwenden. Dazu gibt es zwei Möglichkeiten:

- 1.) Sie senden per MQTT-Explorer an das Topic Ihrer Gruppe händisch Daten, die dann von dem Desktop-Programm oder Webserver-Programm empfangen und angezeigt werden.
- 2.) Sie senden mit einem Python-Programm, ähnlich wie dem "sendMQTTShort.py"-Beispiel, simulierte Sensor-Daten an das Topic Ihrer Gruppe.

## GUI-Programmierung

Es empfiehlt sich eines der möglichen Frameworks auszusuchen und dann gemäß der beigefügten Links daran zu arbeiten, um z.B. Matplot in einem Fenster darzustellen. Hilfreiche Links:

<https://datatofish.com/matplotlib-charts-tkinter-gui/>

<https://build-system.fman.io/pyqt5-tutorial>

<https://www.pythonguis.com/tutorials/plotting-matplotlib/>

[https://matplotlib.org/stable/gallery/user\\_interfaces/embedding\\_in\\_tk\\_sgskip.html](https://matplotlib.org/stable/gallery/user_interfaces/embedding_in_tk_sgskip.html)

[https://matplotlib.org/stable/api/as\\_gen/matplotlib.animation.FuncAnimation.html#matplotlib.animation.FuncAnimation](https://matplotlib.org/stable/api/as_gen/matplotlib.animation.FuncAnimation.html#matplotlib.animation.FuncAnimation)

<https://www.pythontutorial.net/tkinter/tkinter-grid/>

## Darstellung per Web-Oberfläche

Für die Darstellung per Web-Oberfläche wird das Framework [Bottle](#) genutzt. Der Aufbau ist wie folgt: Auf dem Raspberry Pi läuft das Programm mit dem die Temperaturen ausgelesen und per MQTT gesendet werden. Zusätzlich wird auf dem Raspberry Pi ein Webserver gestartet, welcher eine Website erzeugt, welche die Live-Temperatur im Browser anzeigt und zusätzlich die gesammelten Temperatur-Daten zum Download als csv-Datei anbietet, um diese für Machine-Learning nutzen zu können.

Auf der Projektseite werden die nötigen Informationen bereitgestellt. Es empfiehlt sich das "Hello-World-Programm" zu erzeugen: <https://bottlepy.org/docs/dev/tutorial.html#quickstart-hello-world>

Um die Website zu besuchen, wird im Browser lokal die angegebene URL verwendet, soll die auf dem Raspberry Pi gestartete Website besucht werden, muss statt „localhost“ die IP-Adresse verwendet werden: [http://<IP\\_ADRESSE\\_PI>:8080/hello](http://<IP_ADRESSE_PI>:8080/hello)

Das Framework muss wie jedes Python Modul installiert werden: ***pip install bottle***

Man kann einen Webserver im Hintergrund starten, indem man die Threading Bibliothek nutzt.

## Fehlervermeidung

Raspberrys mit „poweroff“ herunterfahren, sonst kann es zu Problemen mit dem Dateisystem kommen. Es muss eine Verbindung per VPN-Client oder ein RZBT-Rechner-Pool genutzt werden, um auf den MQTT-Server zugreifen zu können.

Über die „cmd.exe“ kann man mit dem Befehl „ping <IP>“ die Netzwerkerreichbarkeit prüfen.