**Installation der notwendigen Anwendungen und Bibliotheken**

Zunächst werde alle Kommandos aufgelistet, die im Raspberry Pi ausgeführt werden sollen, um die Anwendung dieses Projekts verwenden zu können. Bevor man im Raspberry Pi mit der Installation aller notwendigen Packages für dieses Projekt anfängt. Sollte man ein allgemeines Update durchführen.

**Allgemeinen Update**

***pi@raspberrypi:~$ sudo apt-get update && sudo apt-get upgrade && sudo apt-get clean***

Dies sollte auch die letzte Version von Python im Raspberry Pi zur Verfügung stellen.

**Installation von MQTT**

Für die Installation vom Mosquitto Broker gilt:

***pi@raspberrypi:~$ sudo apt install -y mosquitto mosquitto-clients***

Um beim Booten vom Raspberry Pi die MQTT Anwendung automatisch zu starten, gibt man den folgenden Befehl ein:

***pi@raspberrypi:~$ sudo systemctl enable mosquitto.service***

Um die Installation zu prüfen:

***pi@raspberrypi:~$ mosquitto -v***

Um den MQTT Service mit Python nutzen zu können, haben wir die Paho Bibliothek verwendet. Dafür benutzen wir den python-installer pip Version 3:

***pi@raspberrypi:~$ pip3 install paho-mqtt***

Somit hat man den MQTT Service installiert und jetzt kann man in Python die Paho-Library benutzen, um Nachrichten per MQTT an allen Clients zu schicken und von Clients Nachrichten zu empfangen.

**Installation von Node-Red**

Node-Red ist zu dem Raspberry Pi mittels MQTT abonniert und empfängt seine Nachrichten für die jeweiligen Topics. In unserem Fall kam der Raspberry Pi bereits mit der neuesten Version von Node-Red installiert. Wenn das aber nicht der Fall ist, muss man die nächsten Befehlen im Terminal ausführen, um Node-Red zu installieren:

***pi@raspberrypi:~$***

***$bash < (curl -sL https://raw.githubusercontent.com/nodered/linuxinstallers/master/deb/update-nodejs-and-nodered)***

Es ist auch möglich, das nächste Befehl durchzuführen:

***pi@raspberrypi:~$ sudo apt-get install nodered***

Für dieses Projekt ist es ebenfalls notwendig, Node-Red als automatische Service beim Booten zu deklarieren.

***pi@raspberrypi:~$ sudo systemctl enable nodered.service***

Um den Node-Red Editor sehen zu können, muss man im Internet-Browser die IP-Adresse des Raspberry Pi’s eingeben und die Portnummer 1880.

***http://<hostname>:1880***

**Installation von InfluxDB**

Die gleichen Voraussetzungen sind für die Installation von InfluxDB gewünscht. Man muss das Service installieren und beim Booten vom Raspberry Pi soll es automatisch starten.

***pi@raspberrypi:~$ sudo apt-get install influxdb***

***pi@raspberrypi:~$ sudo systemctl unmask influxdb***

***pi@raspberrypi:~$ sudo systemctl enable influxdb***

***pi@raspberrypi:~$ sudo systemctl start influxdb***

Um die Anwendung im Terminal zu starten und die InfluxDB-Datenbank zu verwalten, muss man folgender Befehl eingeben:

***pi@raspberrypi:~$ influx***

Für dieses Projekt haben wir die Datenbank „RaspberryPi“ kreiert. Sie erhält die Tabellen „Temperatur“ und „Schalter“. Diese wurden mittels Node-Red hergestellt. Die Erstellung der Datenbank selbst erfolgt durch den Terminal mit dem folgenden Befehl:

***pi@raspberrypi:~$ CREATE DATABASE RaspberryPi***

Weitere Befehle werden bei der Beschreibung der Funktionsweise unserer Anwendung besprochen.

**Installation von Grafana**

Für die Installation von Grafana muss man zuerst die Links zu den nötigen Packages in der Package-Liste vom Raspberry Pi eintragen und authentifizieren.

***pi@raspberrypi:~$ wget -q -O – https://packages.grafana.com/gpg.key | sudo apt-key add –***

***pi@raspberrypi:~$ echo “deb https://packages.grafana.com/oss/deb stable main” | sudo tee -a /etc/apt/sources.list.d/grafana.list***

***pi@raspberrypi:~$ sudo apt-get install -y grafana***

Wir wollen ebenfalls, dass der Grafana Server beim Starten vom Raspberry Pi zur Verfügung steht.

***pi@raspberrypi:~$ sudo systemctl enable grafana-server***

***pi@raspberrypi:~$ sudo systemctl start grafana-server***