# Projekt-Dokumentation: Dateninfrastruktur von Moki-Pro

## Umsetzung der Infrastruktur

Dieses Dokument beschreibt die Umsetzung der Infrastruktur zur Datenerfassung. Dabei werden Daten von Sensoren an der Schleifmaschine (Ultrasonic) erfasst, in InfluxDB gespeichert und in Grafana visualisiert.

### Überblick über die Infrastruktur

Die Abbildung 1 zeigt den strukturierten Prozessfluss innerhalb der entwickelten Infrastruktur:

Ein Bild, das Text, Screenshot, Schrift, Aufdruck enthält.

Automatisch generierte Beschreibung

Abbildung 1 Prozessfluss

1. **Raspberry Pi**: Erfasst Daten von den angeschlossenen Sensoren.
2. **Python-Skript**: Bereitet die Sensordaten auf dem Raspberry Pi auf.
3. **MQTT-Client**: Überträgt die aufbereiteten Daten an den MQTT-Broker.
4. **MQTT-Broker**: Empfängt die Daten und leitet sie an Telegraf weiter.
5. **Telegraf**: Verarbeitet die empfangenen Daten und speichert sie in InfluxDB.
6. **InfluxDB**: Archiviert die Messwerte.
7. **Grafana**: Visualisiert die gespeicherten Daten in benutzerfreundlichen Dashboards.

### 1.2 Hardwarekomponenten

Die verwendete Hardware umfasst:

* **Raspberry Pi 4 Model B Rev 1.1**
* **(Farbsensor (TCS34725RGB)**: Erfasst den Status der Maschine über Farb-LEDs.)
* **Temperatursensor (DS1820)**: Misst die Temperatur innerhalb der Schleifmaschine.
* **(Temperatur- und Feuchtigkeitssensor (DHT22)**: Misst die Umgebungstemperatur und Luftfeuchtigkeit im Labor.)
* **(Temperatur- und Feuchtigkeitssensor (SHR31)**: Misst die Umgebungstemperatur und Luftfeuchtigkeit im Labor.)

Die Sensoren sind über ein Breadboard mit dem Raspberry Pi verbunden.

### Sensoren-Schaltung

Die Sensoren werden einzeln über ein Breadboard mit dem Raspberry Pi verbunden. Die Schaltpläne für die Sensoren, in den folgenden Abbildungen zusehen, und nähere Details wie die Stromversorgung können auf den jeweiligen Shop-Seiten der Sensoren eingesehen werden (siehe Anhang).

Ein Bild, das Text, Screenshot, Schaltung, Design enthält.

Automatisch generierte Beschreibung

Abbildung 2 Schaltung DS18B20

Ein Bild, das Schaltung, Text, Elektronik, Elektronisches Bauteil enthält.

Automatisch generierte Beschreibung

Abbildung 3 Schaltung SHT31

Abbildung 4 zeigt die Verdrahtung der Sensoren mit dem Breadboard und dem Raspberry Pi. Hierbei repräsentieren die roten Kabel die Stromversorgung, die schwarzen Kabel den Ground und die gelben sowie blauen Kabel dienen der Datenübertragung. Der DHT-Sensor und der Farbsensor, werden nicht mehr verwendet. Stattdessen wird der SHT31-Sensor verwendet dabei ist die Verkabelung identisch zu dem Farbsensor.

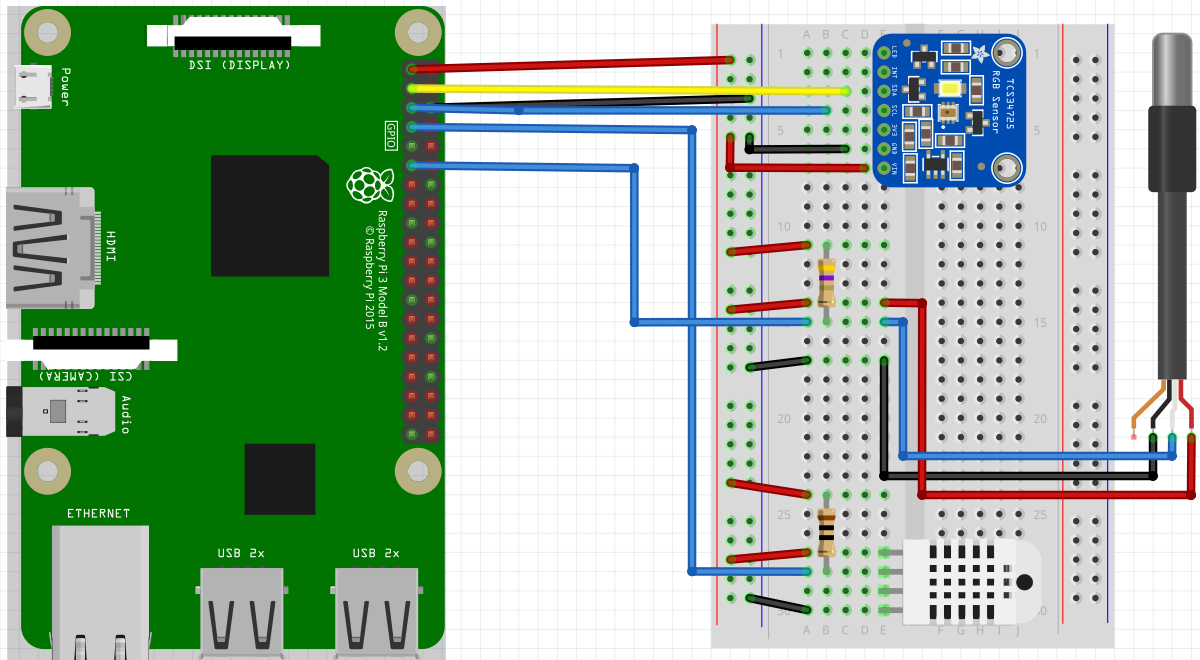


Abbildung 4 Schaltung der Sensoren

### Gehäuse und Kabelmanagement

Um die Hardwarekomponenten geordnet und geschützt aufzubauen, wurde ein speziell für den Raspberry Pi konzipiertes Gehäuse verwendet. Nach der Bohrung von Löchern konnten die Anschlüsse installiert werden. Die Kabel wurden an die Sensoren gelötet und im Gehäuse befestigt. Abbildung 6 zeigt den Versuchsaufbau. Die einzelnen Komponenten sind in den Quellen zu finden.

Ein Bild, das Elektronik, Elektrisches Bauelement, Elektronisches Bauteil, Elektrische Leitungen enthält.

Automatisch generierte Beschreibung

Abbildung 6 Versuchsaufbau

### Softwarekomponenten

Die Softwarekomponenten umfassen:

* **Betriebssystem**: Raspbian GNU/Linux 11 (bullseye)
* **Python-Skript**: Erfasst und überträgt Sensordaten an den MQTT-Broker
* **Docker-compose.yml**: dient zur Definition und Verwaltung von Multi-Container-Docker-Anwendungen
* **MQTT Mosquitto**: Als Broker zur Datenübertragung
* **Telegraf**: Als Datenkollektor und -verarbeiter
* **InfluxDB**: Zur Datenspeicherung
* **Grafana**: Zur Datenvisualisierung

### Softwarekonfiguration

#### DS1820 Sensor Konfiguration

Der DS1820-Sensor wurde erfolgreich an den Raspberry Pi angeschlossen und konfiguriert. Ein Python-Skript wurde verwendet, um die Datenerfassung zu ermöglichen. Für detaillierte Anweisungen wurde die externe Dokumentation von Random Nerd Tutorials herangezogen (<https://randomnerdtutorials.com/raspberry-pi-ds18b20-python/>).

#### NTP-Server Einstellungen

Zur Synchronisation der Zeit wurde der Raspberry Pi so konfiguriert, dass bevorzugt ein interner NTP-Server genutzt wird und bei dessen Ausfall automatisch auf externe NTP-Server umgeschaltet wird. Hierzu wurde **chrony** eingesetzt.

1. **Installation von chrony**  
   Chrony wurde installiert und aktualisiert:

sudo apt update

sudo apt install chrony

1. **Konfiguration anpassen**  
   Die Konfigurationsdatei wurde angepasst, um interne und externe NTP-Server einzutragen:

# Interner NTP-Server

server 192.168.1.100 iburst prefer

# Externe NTP-Server

server 0.de.pool.ntp.org iburst

server 1.de.pool.ntp.org iburst

server 2.de.pool.ntp.org iburst

server 3.de.pool.ntp.org iburst

Die Parameter **iburst** und **prefer** wurden verwendet, um die Synchronisation zu beschleunigen und den internen Server zu priorisieren.

1. **Konfiguration testen und anwenden**  
   Nach der Anpassung wurde chrony neu gestartet, und die Konfiguration wurde überprüft:

sudo systemctl restart chrony

chronyc sources -v

Der bevorzugte Server wurde mit einem ^\*-Symbol gekennzeichnet.

Automatisches Starten des Sensorerfassungsskripts

Das Sensorerfassungsskript wurde so eingerichtet, dass es automatisch beim Systemstart ausgeführt wird. Hierfür wurde **systemd** genutzt.

1. **Service-Datei erstellen**  
   Eine Datei mit dem Namen /etc/systemd/system/sensor\_script.service wurde erstellt und folgender Inhalt hinzugefügt:

[Unit]

Description=Sensor Data Script

After=network.target

[Service]

ExecStart=/usr/bin/python3 /path/to/your/script.py

Restart=always

User=pi

Group=pi

[Install]

WantedBy=multi-user.target

1. **Service aktivieren und starten**  
   Der Service wurde aktiviert und gestartet:

sudo systemctl enable sensor\_script.service

sudo systemctl start sensor\_script.service

sudo systemctl status sensor\_script.service

Dadurch wurde sichergestellt, dass das Skript automatisch neu startet, falls es abstürzt.

#### Quellen:

|  |  |
| --- | --- |
|  | Informationen zu den Sensoren |
| Datenblatt DHT22: | <https://cdn.shopify.com/s/files/1/1509/1638/files/DHT_22_-_AM2302_Temperatur_und_Luftfeuchtigkeitssensor_Datenblatt.pdf?11983326290748777409> |
| Schaltungsplan  DHT22: | <https://cdn.shopify.com/s/files/1/1509/1638/files/DHT_22_-_AM2302_Temperatur-_und_Luftfeuchtigkeitssensor_atmega_Schematics_1.pdf?v=1677138007> |
| Datenblatt  DS18B20: | <https://cdn.shopify.com/s/files/1/1509/1638/files/DS18B20_TO-92_datasheet_0ec6827c-2bee-4eca-8a6d-5d68ae26bd8f.pdf?v=1656592898> |
| Schaltungsplan  DS18B20: | <https://cdn.shopify.com/s/files/1/1509/1638/files/DS18B20_Temperatursensors_Connection_Diagrams.pdf?v=1615282043> |
| Datenblatt  TCS3472: | <https://cdn.shopify.com/s/files/1/1509/1638/files/TCS3472_datasheet.pdf?v=1675927629> |
| Schaltungsplan  TCS3472: | <http://www.getmicros.net/raspberry-pi-and-tcs34725-color-sensor-java-example.php> |
| Schaltungsplan  SHT31 | <https://learn.adafruit.com/adafruit-sht31-d-temperature-and-humidity-sensor-breakout/python-circuitpython> |
| Einrichtung für den SHT31 | <https://www.reddit.com/r/raspberry_pi/comments/x24aem/raspberry_pi_and_reading_sht3x_sensors/> |
|  | Komponenten |
| Raspberry Pi 4 Model B Rev 1.1 | <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> |
| DHT22-Sensor | <https://www.reichelt.de/entwicklerboards-temperatur-feuchtigkeitssensor-dht22-debo-dht-22-brd-p266104.html?PROVID=2788&gclid=CjwKCAjwgsqoBhBNEiwAwe5w07Tew4_fYczufxC7GeSqlt5688w8__-LK-xz1gMHgw_wZBId205oRRoCEHkQAvD_BwE> |
| DS18B20-Sensor | <https://www.reichelt.de/shelly-temperatur-sensor-ds18b20-shelly-ds18b20-p287127.html?&trstct=pos_0&nbc=1> |
| TCS3472-Sensor | <https://www.az-delivery.de/products/tcs34725-rgb-color-sensor?_pos=1&_sid=addfee278&_ss=r> |
| Gehäuse Cube | <https://www.reichelt.de/gehaeuse-fuer-raspberry-pi-4-cube-dunkelgrau-rpi-case4-cb-dg-p279773.html> |
| Breadboard | <https://www.reichelt.de/experimentier-slide-steckboard-300-100-kontakte-steckboard-s4-p177331.html?&trstct=pos_6&nbc=1> |
| M 8, 4-pol, Buchse | <https://www.reichelt.de/sal-m-8-4-pol-buchse-gerade-conec-42-01245-p249162.html?&trstct=vrt_pdn&nbc=1> |
| Sensorleitung, M8, 4 pol | <https://www.reichelt.de/sensorleitung-m8-4-pol-st-offenes-ende-2-m-lut-0810-0-2-p223081.html?&trstct=vrt_pdn&nbc=1> |
|  |  |
|  |  |
| Betriebssystem  Raspberry: | <https://www.raspberrypi.com/software/operating-systems/> |
|  |  |