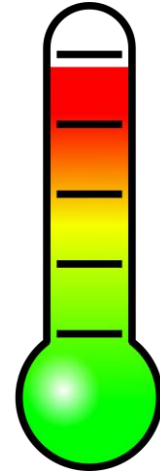


Room Quality Station

HTW Berlin

Project Computer Systems Engineering

WiSe 2022/23 – Team D



htw

Hochschule für Technik
und Wirtschaft Berlin

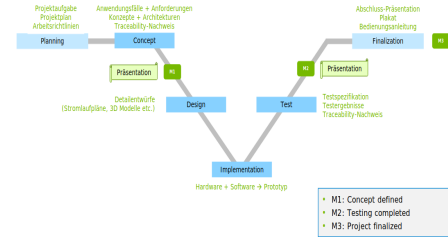
University of Applied Sciences

Max Lausch, Hai Nam La, Andreas Welter / 25.01.2023

Room Quality Station

Gliederung

- 1) Rückblick auf M1
- 2) Arbeitsstand
- 3) Hardware
- 4) Software
- 5) Test Cases
- 6) Vorführung
- 7) Aussicht auf M3

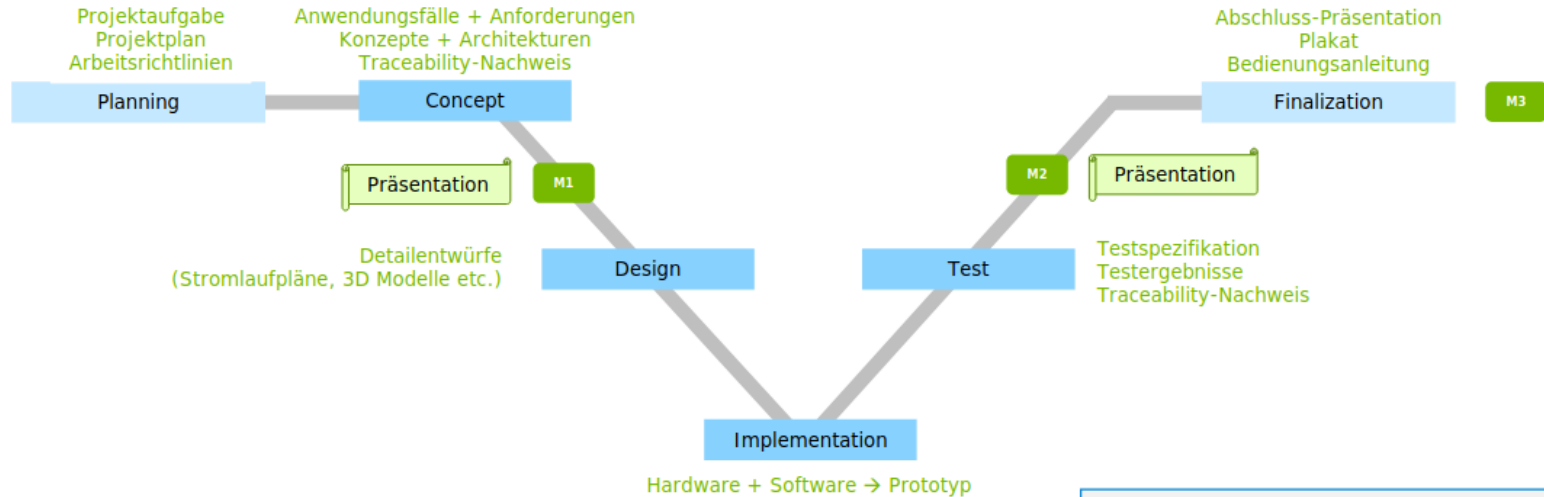


C



1) Rückblick auf M1

Roadmap



- M1: Concept defined
- M2: Testing completed
- M3: Project finalized

1) Rückblick auf M1

Rückblick



1) Rückblick auf M1

Veränderungen nach M1

- "Weather station" → "Room Quality Station"
- + Luftfeuchte, Luftqualität
- - CO2-level
- Schnittstelle von Pi zum Display: I²C → SPI
- Buzzer zum Projekt hinzugefügt

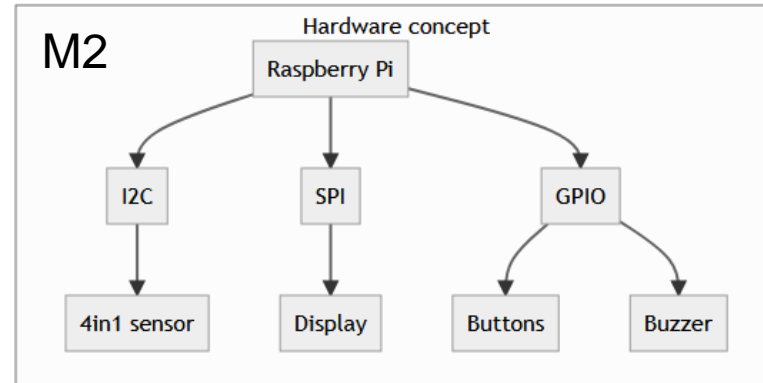
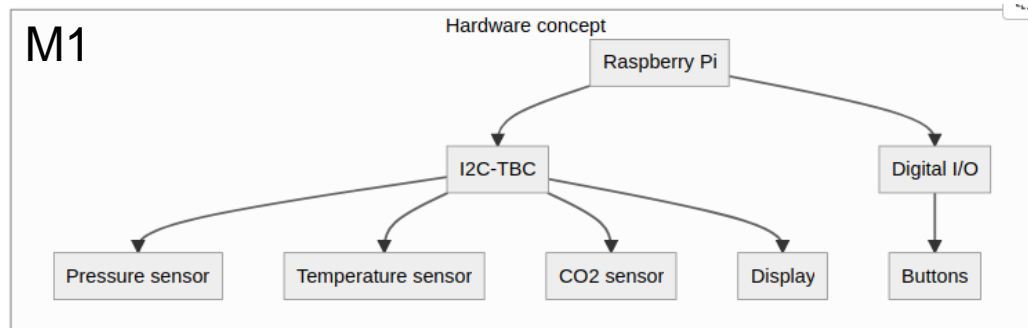


%



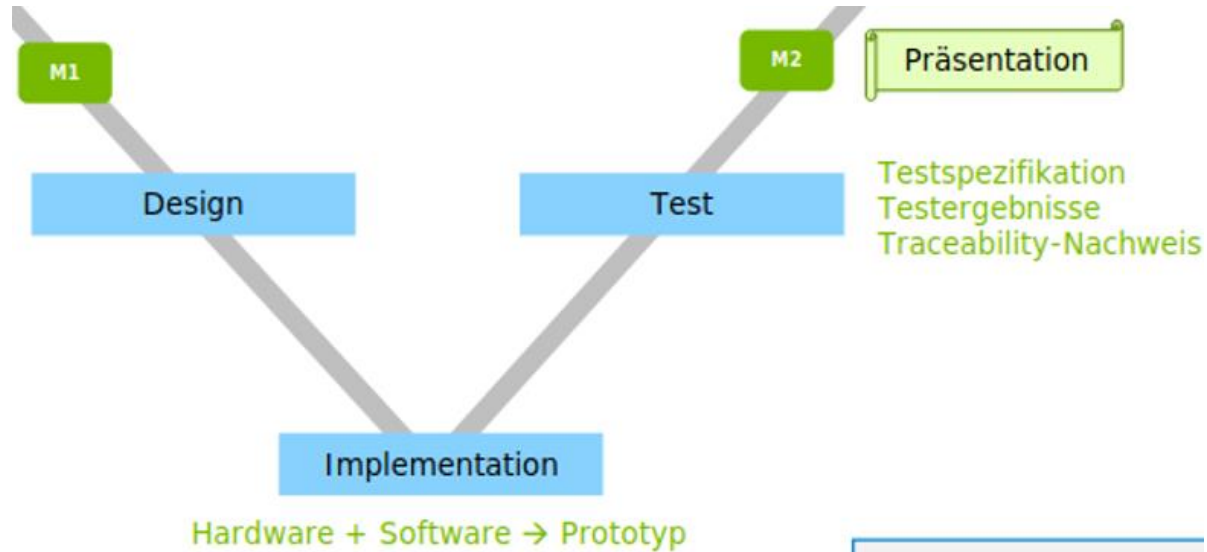
1) Rückblick auf M1

Veränderungen nach M1 am Hardwarekonzept



2) Arbeitsstand

Fortschritt



2) Arbeitsstand

Probleme/Risiken und Lösungsansätze

Probleme/Risiken	Lösungsansätze
Ausfall einer der Teammitglieder, Gefährdung des Meilensteinziels	<ul style="list-style-type: none">• Zeitlich realistisches Projekt• Gute Teamarbeit• Gutes Zeitmanagement
Display benötigt umständliche Hardwareanpassung für I ² C	<ul style="list-style-type: none">• Wechsel zu Schnittstelle SPI
Softwareumsetzung ist zeitaufwendig	<ul style="list-style-type: none">• Zeitlich realistisches Projekt

3) Hardware

Komponenten



Raspberry Pi 3B



OLED-15W-C
Display
und Buttons



BME680
4in1 Sensor



KY-012
Buzzer

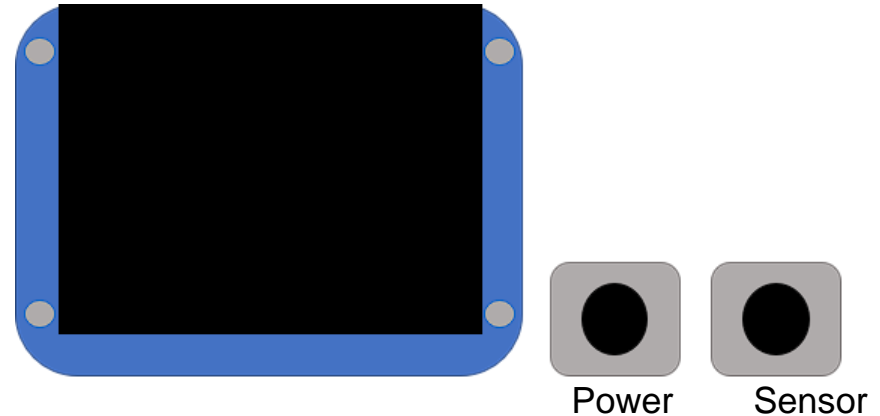
3) Hardware

Bedienungskonzept

Zustand des Systems: Aus

Interaktion:

Anforderung: System ist aus



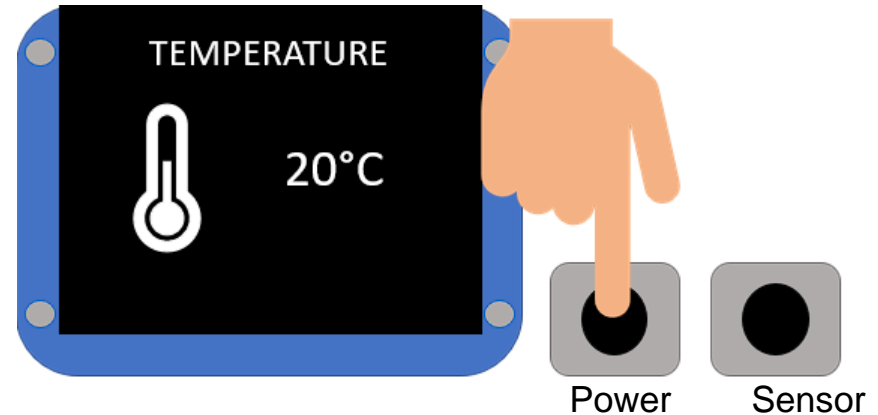
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Powerbutton wird gedrückt

Anforderung: Endnachricht



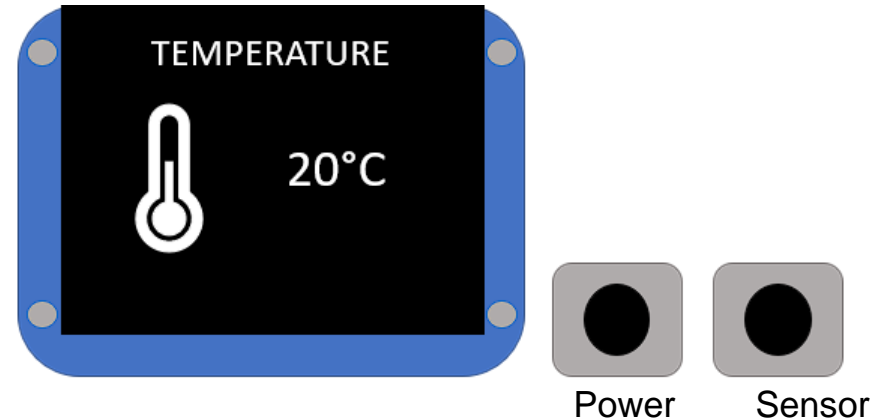
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



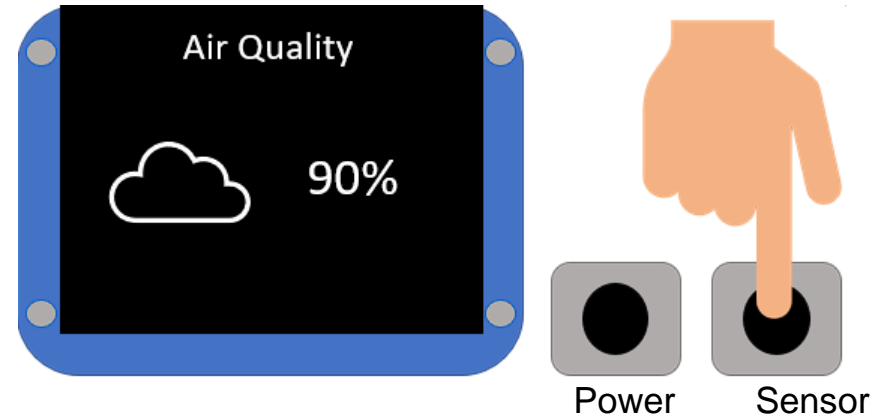
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



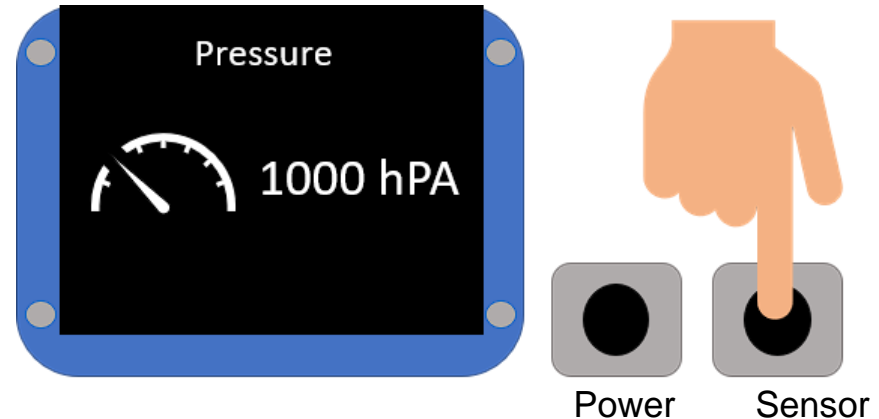
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



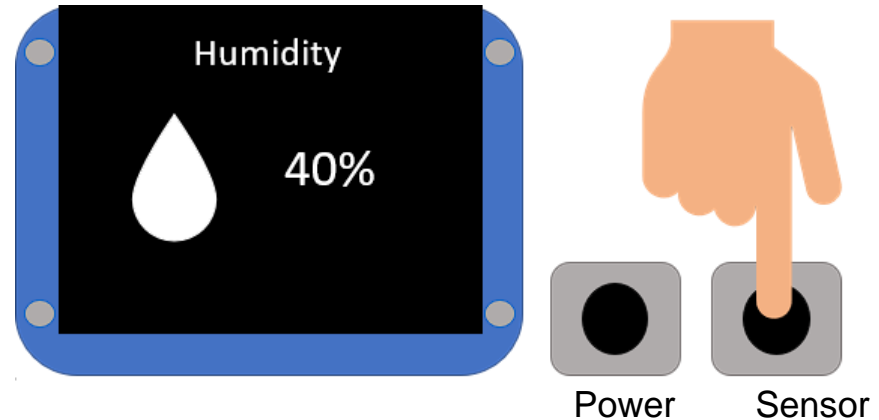
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



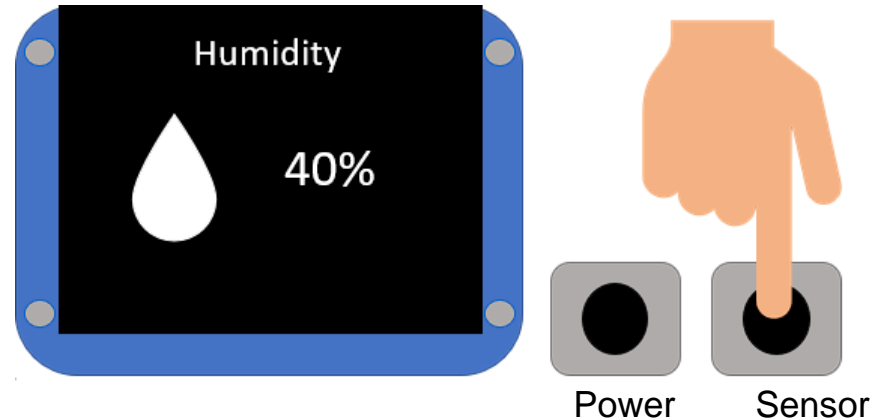
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



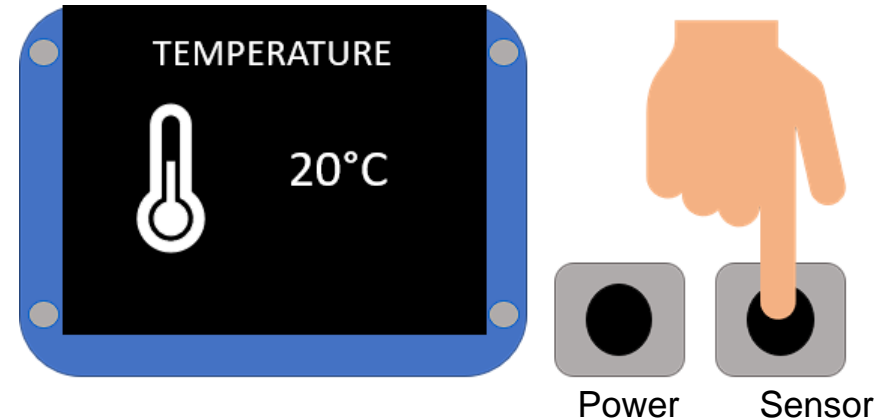
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion: Sensorbutton wird gedrückt

Anforderung: Anzeige des nächsten Sensors



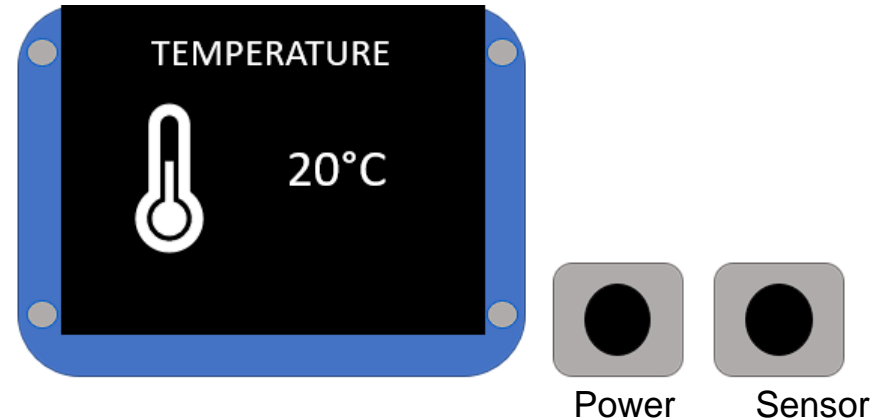
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion:

Anforderung: Temperaturanzeige



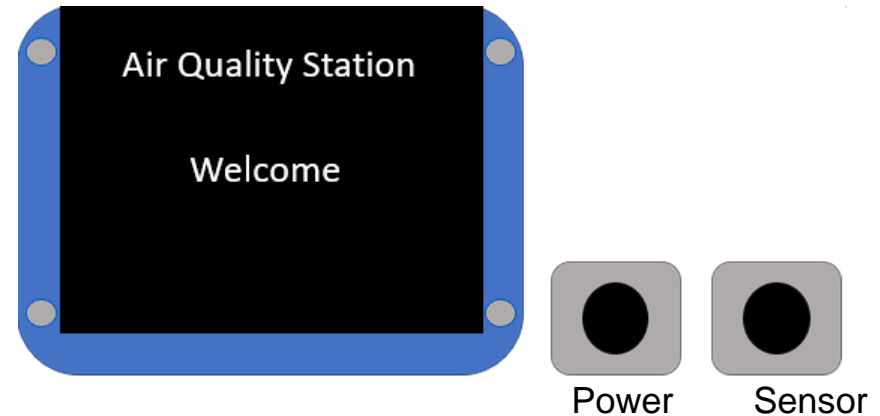
3) Hardware

Bedienungskonzept

Zustand des Systems: An

Interaktion:

Anforderung: Startnachricht



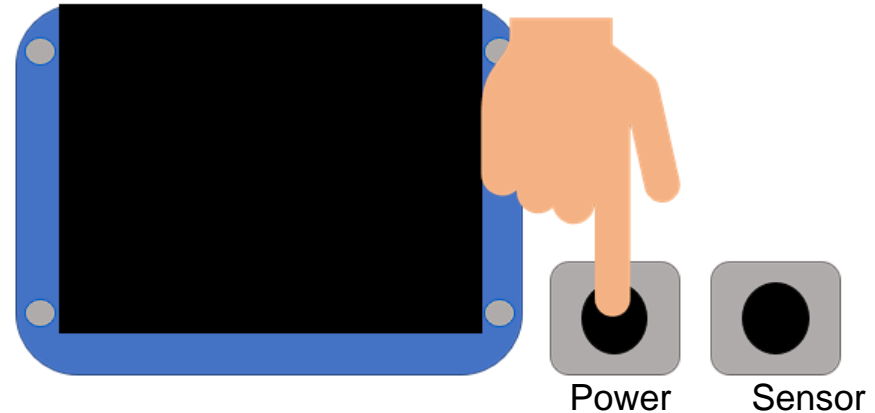
3) Hardware

Bedienungskonzept

Zustand des Systems: Aus

Interaktion: Powerbutton wird gedrückt

Anforderung: System startet



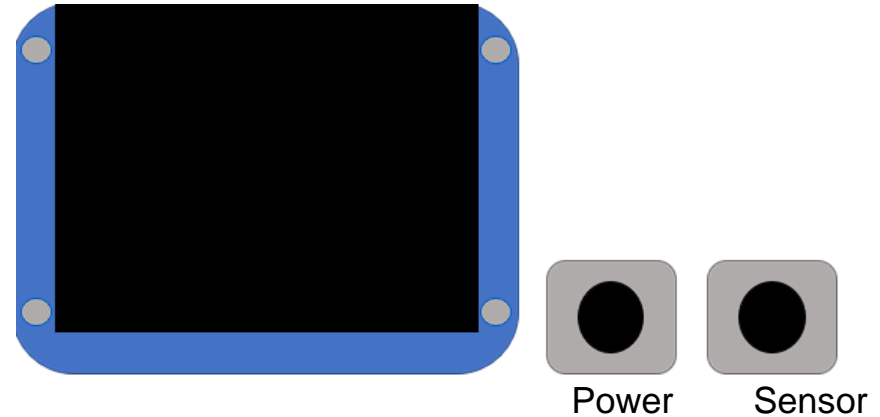
3) Hardware

Bedienungskonzept

Zustand des Systems: Aus

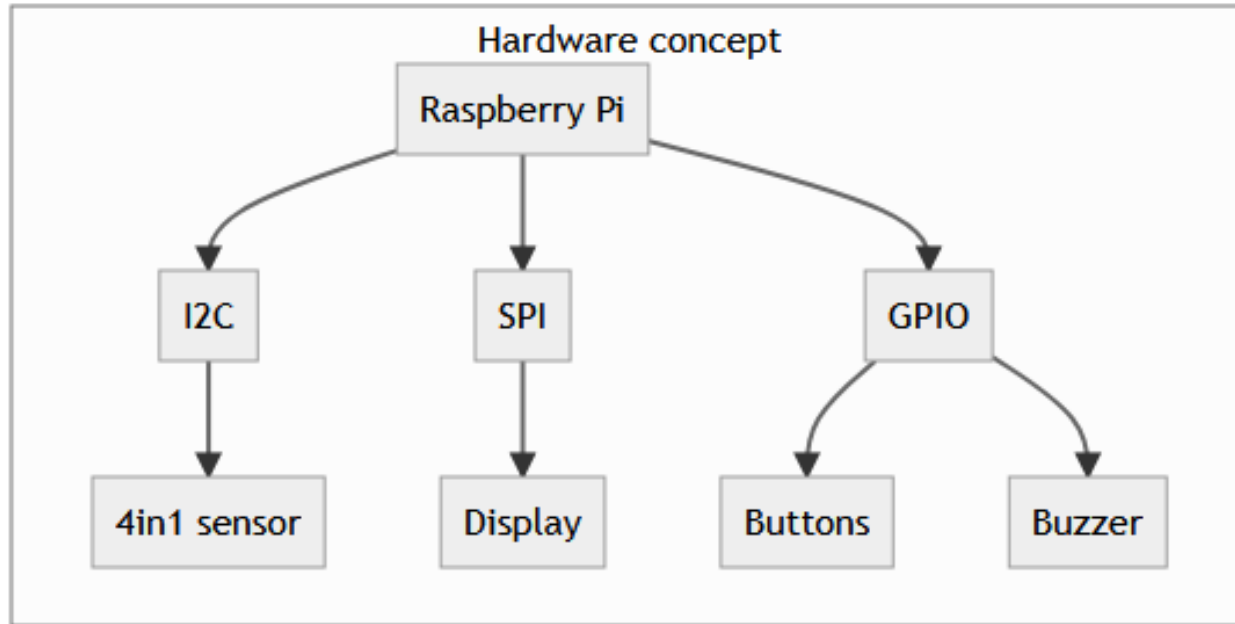
Interaktion:

Anforderung:



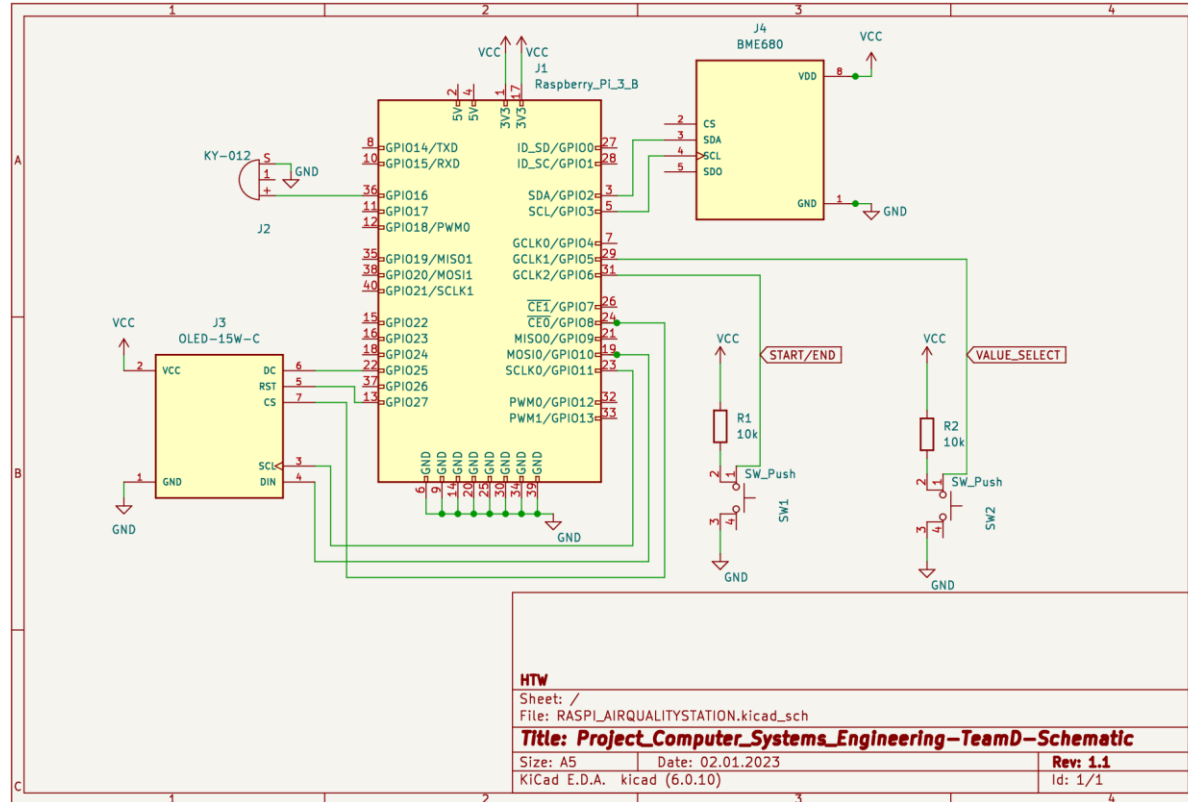
3) Hardware

Hardwarekonzept



3) Hardware

Schaltung



4) Software

Display – Waveshare Treiber Software

- **Waveshare-Driver API:**
 - Aktivierung zu nutzender GPIOs
 - Vereinfachtes lesen/setzen von Einstellungen.
 - Lesen/Schreiben bestimmter Register.
 - Definierung von Display-Ausgabe Funktionen
- **Todo als Benutzer:**
 - I2C oder SPI Kommunikation aktivieren.
 - Displayausgabe implementieren

4) Software

Display – Custom display API

Datenstruktur:

- `Ubyte BlackImage;`

Display initialisieren:

- `void OLED_1in5_test(void);`

Ausgabe zum Display

- `void OLED_while(bmedata variable);`

Freigeben:

- `void OLED_1in5_Clear()`

- `void DEV_ModuleExit()`

4) Software

BME680 Sensor – Bosch Treiber Software

Bosch-Driver API:

- Konvertierung von Messwerten.
- Vereinfachtes lesen/setzen von Einstellungen.
- Schlichtes auslesen der Messdaten.

Todo als Benutzer:

- I2C oder SPI Kommunikation implementieren.
- Sensoreinstellungen konfigurieren.
- Überprüfen der Messwerte.

4) Software

BME680 Sensor – Custom Sensor API

Datenstrukturen:

- **struct** bme680_dev* sensor;
- **struct** bme680_field_data* data;

Sensor initialisieren:

- **sensor_init(sensor);**

Abfragen der Messwerte:

- **sensor_get_all_data(sensor, data);**

Freigeben:

- **sensor_disable(sensor, data);**

4) Software

Buzzer

Datenstrukturen:

```
bmedata;
```

Buzzer initialisieren:

```
void buzzer_initialize();
```

Buzzer an- und ausschalten :

```
void buzzer_on();
```

```
void buzzer_off();
```

```
void buzzer_toggle();
```

Freigeben:

```
void buzzer_end();
```

Steuerfunktion:

```
void set_buzzer_alarm(bmedata data);
```

4) Software

Buzzer

Steuerfunktion:

```
void set_buzzer_alarm(bmedata data){
```

```
-> if      ( Grenzwertüberschreitung ) :      buzzer_on();
```



```
-> else if ( Grenzwertunterschreitung ):      buzzer_on();
```



```
-> else:      buzzer_off();  
}
```



4) Software

Polling interrupt – Warum und wie?

- **Warum:**

- Taster
- Um die die Hauptfunktion des Systems zu ändern ohne es abubrechen

- **Wie:**

- Parallele Schleife zur Überprüfung des Tastendrucks
- Zuweisung eines Threads
- Änderung einer Variable, die in der Hauptfunktion im Polling abgefragt wird

4) Software

GPIO Interrupt Kernelmodul – Warum und wie?

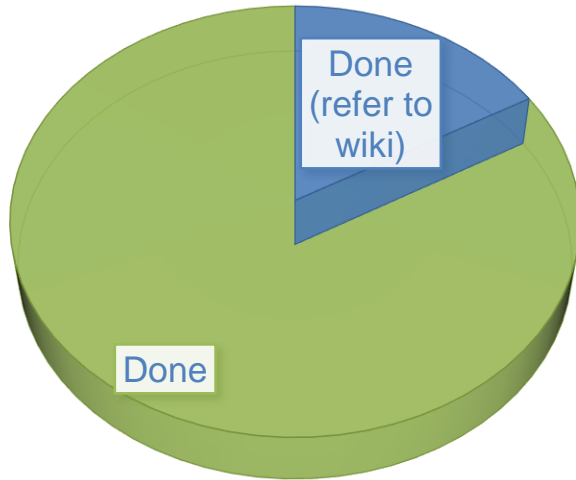
Warum:

- Um Leistung zu sparen.
- Reagieren auf Tastendruck zuverlässiger machen.
- Kernel-Modul für schnellere Hardware-Zugriffe.

Wie:

- GPIO-Konfiguration.
- Verbindung zwischen ISR und IRQ.
- /proc Dateisystem zur Kommunikation.

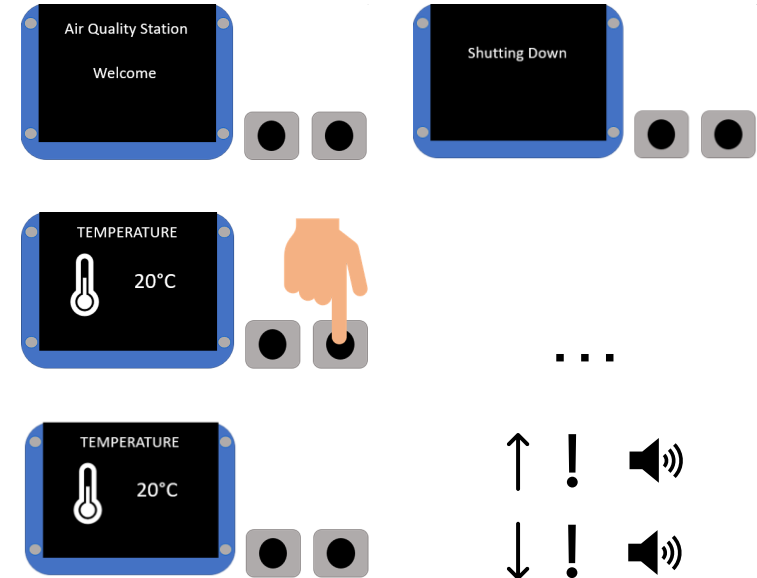
5) Test cases



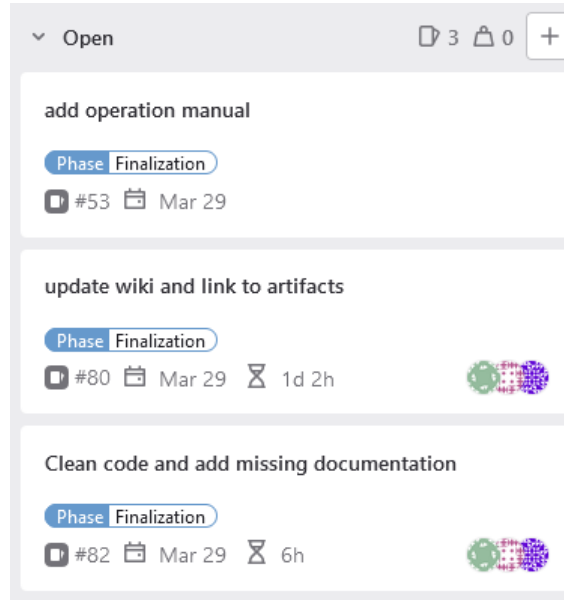
- T001: Boot-Up (Done)
- T006: Start and end screen (Done)
- T002: Mode-Button (Done)
- T005: Measurement tolerance (Done, ref)
- T003: Alarm test max threshold (Done)
- T004: Alarm test min threshold (Done)

6) Vorführung

- T001: Boot-Up (Done)
- T006: Start and end screen (Done)
- T002: Mode-button (Done)
- T005: Measurement tolerance (Done, ref)
- T003: Alarm test max threshold (Done)
- T004: Alarm test min threshold (Done)



6) Ausblick auf M3



An aerial photograph of an industrial city, likely Berlin, featuring a river on the left, several large industrial buildings with multiple stories and many windows, and several tall, thin smokestacks rising from the buildings. The image has a green tint.

Vielen Dank für Ihre Aufmerksamkeit.

www.htw-berlin.de



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de

Bildquellen

Folie / Bild	Quelle
1, 2 / Raspberry Pi Grafik	Internet-Link-Wikimedia
1 / Temperaturmessung Grafik	Lizensfrei auf Pixabay
2, 3, 4, 7 / Roadmap	Vorlesungsfolien Project Computer Systems Engineering
2, 10, 22 / Button- und Display Grafik	Siehe Gitlab: Link
5 / CO2 Grafik	Lizensfrei auf Pixabay (Bearbeitet)
5 / Wolke mit Tropfen	Powerpoint-Piktogramme
6, 11 / Hardwarekonzept Grafik	Siehe Gitlab: Link 5, 9
5, 9 / Buzzer	Internet-Link-Berrybase
9 / Button	Internet-Link-Reichelt
9 / Display	Internet-Link-Berrybase
9 / 4in1 Sensor	Internet-Link-Berrybase
12 / Schaltung	In KiCad 6.0 erstellt
18 / Piktogramme: Pfeile, Check, Lautsprecher	Powerpoint-Piktogramme
21 / Kreisdiagramm	Powerpoint-Vorlage
23 / Screenshot Open Issues	Siehe Gitlab: Link