**Ideenskizze für das IoT-Projekt**

**1. Allgemeine Informationen**

* **Ersteller:** *Jonah Eberlei*
* **Projektname:** *Kontrollpanel*
* **Datum:** *11.03.25 – 06.05.25*

**2. Projektbeschreibung**

**Kurzbeschreibung:**  
*(Beschreibt in 3–5 Sätzen die Kernidee des Projekts. Welche Problemstellung wird gelöst? Welche Funktionalitäten sind geplant?)*

Über Taster und eine Weboberfläche werden Steckdosen geschaltet. Mit zwei Sensoren werden die Raum- und Leistungsdaten erfasst und später auf der Weboberfläche visualisiert. Zusätzlich zur Werboberfläche gibt es ein kleines Display, welches die Daten ebenfalls anzeigt und eine Datenbank, in der die Daten abgelegt werden.

**3. Anforderungen und Funktionalitäten**

✅ **Sensorik**

* Luftfeuchtigkeit
* Temperatur
* Energiebezug Steckdosen, aktuell und alltime

✅ **Aktoren-Steuerung**

* Steckdosen
* ESP Hardreset

✅ **Webinterface & Benutzerinteraktion**

* Visualisierung Energiebezug
* Steckdosen einzeln oder als Gruppe schalten
* Button für ESP Hardreset

**4. Benötigte Komponenten**

**4.1 Hardware**

| **Komponente** | **Modell/Typ** | **Funktion** |
| --- | --- | --- |
| **Mikrocontroller** | ESP32-S3-DevKit-C |  |
| **Sensor 1** | AHT20 | Temperatur und Luftfeuchtigkeit |
| **Sensor 2** | Shelly plus 1PM | Energieüberwachung und 1 Schaltaktor |
| **Aktor 1** | SRD-03VDC SL C | 10A Relais |
| **Aktor 2** | 1,69“ IPS Display | Visualisierung |
| **Stromversorgung** | Mean Well HDR-60-5 | 230V –> 5V Netzteil |
|  | Step-Down Modul 3V3 | Spannungsversorgung Relais |
|  | Siemens 5SL61166 | B16 LS |
|  | Siemens 5SU13566KK10 | B10, 0.03A RCBO |
| **Weitere Bauteile** | Taster24V DC, NO | Einschalten des Displays und der Steckdosen |
|  | Hutschiene | Montage Netzteil und RCBO |
|  | Steckdosen Gira System/Standart 55 |  |
|  | Verschraubung M16 | Zugentlastung |
|  | Schuko Stecker | Verbindung Stromnetz |
|  | Unterverteilung |  |
|  | Verbindungs und Kleinmaterial |  |

**4.2 Software & Datenbank**

| **Komponente** | **Technologie** | **Funktion** |
| --- | --- | --- |
| **Microcontroller-Code** | MicroPython |  |
| **Webinterface** | Node-Red | Visualisierung und Usereingaben |
| **Datenbank** | MariaDB |  |

**6. Zeitplanung (Meilensteine)**

| **Datum** | **Aufgabe** |
| --- | --- |
| KW 1 | Detailplanung, Material beschaffen |
| KW 2 | Hardware für Tests zusammenbauen, Ansteuerung Relais |
| KW 3 | Mit MQTT-Broker und Node-Red verbinden, Shelly einbinden |
| KW 4 | Taster, AHT und Display einbinden |
| KW 5 | Node-Red Visualisierung und Buttons einbinden |
| KW 6 | Logs implementieren, Datenbank verbinden |
| KW 7 | Bug Fix und Optimierung |
| KW 8 | Dokumentation |

**7. Offene Fragen & Herausforderungen**

* - Der Shelly Schaltaktor arbeitet nicht sauber als NC Kontakt
* - Die Relais benötigen zu viel Energie vom ESP, wenn alle eingeschaltet sind
* - Die Relais reagieren nicht sicher auf Zustandsänderungen

**8. Zielsetzung**

* Das Panel soll mir nach Abschluss der Projektarbeit eine detaillierte Auskunft über den Energieverbrauch der angeschlossenen Geräte und die Umgebungsbedingungen im Raum geben.

**9. Fazit**

Der Frühe Umbau, von der Hutschiene in die Unterverteilung hat alle sehr übersichtlich gestaltet und alle weiteren arbeiten vereinfacht. Durch die Vorbereitung im Unterricht, war die Programmierung der MQTT-Schnittstelle und Node-Red schnell grundlegend bearbeitet. Durch die Unterstützung der KI-Modelle waren auch die fehlenden Java-Script Kenntnisse, die ebenfalls für den Shelly benötigt werden, kein großes Hindernis und die Fehler konnten schnell beseitigt werden.

Was ich stark unterschätzt habe, war die Kostenentwicklung und der Einfluss von EMV auf unsere Microcontroller. Die Taster Signale (Interrupts) leiden aufgrund der geringen Spannung sehr unter Einstreuungen, die u.a. beim Schalten des internen Shelly Relais entstehen. Für einen praktischen Anwendungsfall würde ich meine Taster nicht erneut über einen ESP die Relais ansteuern lassen oder die Steuerspannung erhöhen. Der Effekt der Einkopplungen wird durch die parallel liegenden Leitungen und die allgemeine Länge der Leitung verstärkt. 24V wäre beispielsweise deutlich weniger störanfällig als die 3,3V des ESP. Direkt alles nur über Shelly steuern zu lassen oder als Controller eine LOGO/SPS zu nehmen, würde die Fehleranfälligkeit und die Kostenentwicklung deutlich senken.

Um das ganze mit dem ESP umzusetzen war viel Kleinmaterial und teilweise auch ausprobieren nötig, bis ich das richtige Material gefunden habe. Das hat die Kosten nach oben getrieben, da es bei Leitungen zum Teil. mindestlängen gibt, die gekauft werden müssen.

Insgesamt konnte ich bei dem Projekt einige Dinge lernen, die ich in Zukunft anders machen würde. Ich hatte viel Spaß beim Zusammenbau der Unterverteilung. Diese habe ich während meiner Leerlaufzeit bei einer Ferienveranstaltung durchgeführt. Ich konnte mehreren interessierten Teilis dadurch das Thema anschaulich erklären und ihnen das Feld der Elektrotechnik etwas näherbringen.

Erweiterungen:

* Leuchttaster, die den Schaltzustand angeben
* Remote-Red Anbindung