**Ideenskizze für das IoT-Projekt**

1. **Allgemeine Informationen**

* **Projektname:** Zutrittskontrolle über einen Fingerabdrucks Sensor
* **Datum:** 11.03.2025 – 06.05.2025
* **Ersteller:** Marek Kötter ETS24

**2. Projektbeschreibung**

**Kurzbeschreibung:**

In diesem Projekt wird ein ESP32 als zentrales Steuerungselement verwendet, um eine sichere und effiziente Türöffnung zu realisieren. Ein Fingerabdrucksensor dient dabei als primäres Eingabegerät. Sobald der Benutzer seinen Finger auf den Sensor legt, wird der Fingerabdruck gescannt und mit den gespeicherten Vorlagen verglichen. Wird der Fingerabdruck erfolgreich erkannt, entriegelt sich die Tür automatisch.

Ein Display wird zur Anzeige des Benutzers verwendet. Bei Inaktivität des Systems zeigt das Display stattdessen Umweltdaten wie Temperatur, Luftfeuchtigkeit und Helligkeit an. Diese Daten werden in Echtzeit vom ESP32 erfasst und auf dem Display angezeigt.

Zusätzlich wird der Zutritt des Benutzers mit einem Zeitstempel (Timestamp) in einer HeidiSQL-Datenbank gespeichert. Dies ermöglicht eine detaillierte Protokollierung aller Zugriffsereignisse. Die gespeicherten Daten werden mithilfe von Node-RED visualisiert, sodass eine benutzerfreundliche und übersichtliche Darstellung der Zutrittsprotokolle in Echtzeit verfügbar ist.

**3. Anforderungen und Funktionalitäten**

* Die Umweltsensoren werden benötigt, um die Daten zu erfassen, die später auf dem Display angezeigt werden. Der AHT10-Sensor misst die Temperatur und Luftfeuchtigkeit, während der BH1750 die Lichtstärke der Umgebung ermittelt.  
  Die auf beiden Seiten der Tür installierten Fingerabdrucksensoren dienen der Zutrittskontrolle. Der Benutzer kann die Tür durch das Auflegen seines Fingers öffnen.
* Das Türschloss dient als zentrales Öffnungselement der Tür und kann von beiden Seiten entsperrt werden. Das Display dient der Visualisierung für den Benutzer. Es zeigt an, ob seine Benutzerrechte ausreichen, indem eine positive oder negative Nachricht ausgegeben wird.
* Die Sensordaten werden in einer HeidiSQL-Datenbank gespeichert und anschließend über Node-Red ansprechend visualisiert.

**4. Benötigte Komponenten**

**4.1 Hardware**

| **Komponente** | **Modell/Typ** | **Funktion** |
| --- | --- | --- |
| **Mikrocontroller** | ESP32-S3-WROOM-1 | Kommunikation |
| **Helligkeitssensor** | BH1750 | Infoanzeige auf Display |
| **Temperatur,**  **Feuchtigkeits-**  **snesor** | AHT10 | Infoanzeige auf Display |
| **Elektromagnetisches**  **Türschloss** | LD-0620 | Türentriegelung |
| **Display** | TFT-Display 1,69 SPI | Anzeige von Umweltdaten und Benutzer |
| **Netzteil, Akku** | Powerbank Anker | Spannungsversorgung |
| **Fingerabdrucks Sensor** | R307 Sensor | Finger einlesen/scannen |
| **1 Kanal Relaimodul** | SRD-05VDC-SL-C | Ansteuerung Schloss |

**4.2 Software & Datenbank**

| **Komponente** | **Technologie** | **Funktion** |
| --- | --- | --- |
| **Microcontroller-Code** | Python | Programmierung |
| **Webinterface** | Node-Red / Ui | Visualisierung |
| **Datenbank** | HeidiSQL | Datenspeicherung |

**5. Systemarchitektur**

**5.1 Elektrische Verbindung der Komponenten**

|  |  |  |
| --- | --- | --- |
| **Komponente** | **Verbindung zum ESP32** | **Protokoll/Signal** |
| Fingerabdrucksensor (R307) | Über serielle Schnittstelle verbunden | UART (Serielle Kommunikation) |
| Helligkeitssensor (BH1750) | Über I²C-Bus verbunden | I²C |
| Temperatur- & Feuchtigkeitssensor (AHT10) | Über I²C-Bus verbunden | I²C |
| Elektromagnetisches Türschloss (LD-0620) | Über Relaismodul angesteuert | Digital (High/Low) |
| 1-Kanal Relaismodul (SRD-05VDC-SL-C) | Vom ESP32 angesteuert, schaltet Türschloss | Digital (High/Low) |
| TFT-Display (1,69" SPI oder I²C) | Kommuniziert mit ESP32 über I²C oder SPI | I²C oder SPI |

**5.2 Logische Abläufe & Kommunikation**

**1) Fingerabdruck-Authentifizierung**

1. Der Benutzer legt seinen Finger auf den R307-Sensor.
2. Der ESP32 kommuniziert über UART mit dem Fingerabdrucksensor und fragt den Fingerabdruck ab.
3. Der ESP32 vergleicht den Fingerabdruck mit gespeicherten Mustern.
4. Wenn erfolgreich:
   * Der ESP32 sendet ein High-Signal an das Relaismodul.
   * Das Relais schaltet das elektromagnetische Schloss frei.
   * Das Display zeigt eine positive Meldung an.
5. Wenn nicht erfolgreich:
   * Das Display zeigt eine Fehlermeldung.

**2) Umweltdaten erfassen & anzeigen**

1. Der BH1750 (Lichtsensor) und AHT10 (Temp./Luftfeuchte) sind über I²C mit dem ESP32 verbunden.
2. Der ESP32 liest die Sensordaten regelmäßig aus.
3. Bei Inaktivität zeigt das TFT-Display die Umweltdaten an.

**3) Datenbank-Speicherung & Visualisierung**

1. Jeder Zutrittsversuch (erfolgreich oder nicht) wird mit Zeitstempel erfasst.
2. Der ESP32 sendet die Daten über WLAN an einen Server mit HeidiSQL.
3. Node-RED ruft die Daten aus der Datenbank ab und visualisiert sie.

**4) Gesamtübersicht – Wie alles zusammenarbeitet**

1. Benutzer legt Finger auf den R307-Sensor → ESP32 verarbeitet ihn über UART.
2. Erfolgreiche Identifikation → ESP32 gibt Signal ans Relaismodul → Türschloss öffnet sich.
3. Zutrittsversuch wird mit Zeitstempel in die HeidiSQL-Datenbank geschrieben.
4. Node-RED visualisiert die Zutrittsprotokolle.
5. AHT10 & BH1750 liefern Sensordaten → ESP32 zeigt sie auf dem TFT-Display an.

**6. Zeitplanung (Meilensteine)**

| **Datum** | **Aufgabe** |
| --- | --- |
| KW 1 | Projektstrukturierung |
| KW 2 | Code schreiben |
| KW 3-4 | Code schreiben |
| KW 5 | Komponenten mit dem Code verknüpfen |
| KW 6 | Code schreiben (Fehler Behebung) |
| KW 7 | Fertigstellung |

**7. Offene Fragen & Herausforderungen**

* Ansteuerung des Türschlosses
* Code Funktionalität
* Einlesen der Fingerabdrücke
* Eventuell muss für das Türschloss ein Optokoppler eingebracht werden

**8. Fazit & Zielsetzung**