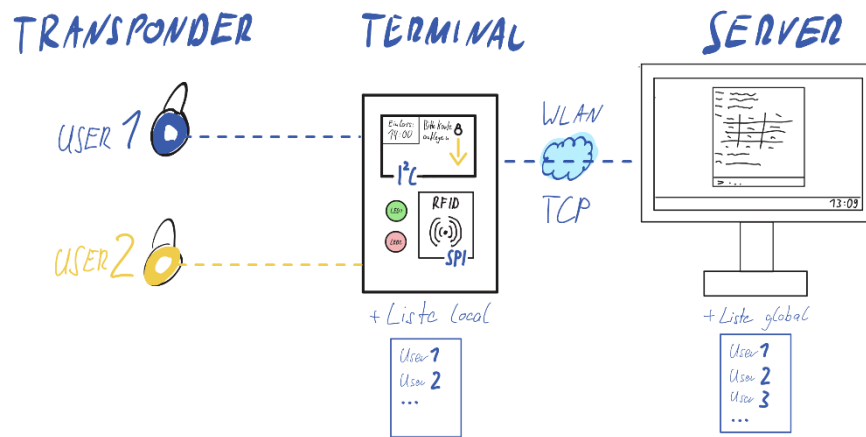


Projektarbeit Datenkommunikation Welzel/Ettl

- RFID-Zutrittskontrolle mittels ESP32 & RC522



1. Ziel der Aufgabe

Entwicklung eines elektronischen Zutrittskontrollsystems auf Basis eines **ESP32**.

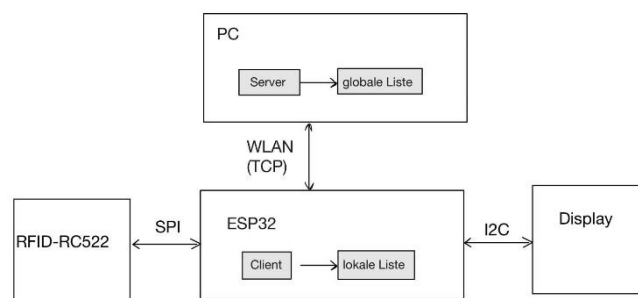
Der ESP32 liest über einen **RFID-Leser (RC522)** eine **Personalnummer** ein, sendet diese über **WLAN (TCP)** an einen Server und erhält eine Rückmeldung, ob der Zugang erlaubt oder verweigert wurde. Der ESP32 zeigt das Ergebnis auf einem **I2C-Display** an.

Zusätzlich kann der Server ein **Sperrzeit-Fenster** setzen, in dem für alle Personen der Zutritt blockiert wird. Dieser Status wird ebenfalls vom ESP32 empfangen und angezeigt.

Wenn keine Verbindung mit dem Server besteht, kann der ESP32 auf eine lokale Liste zurückgreifen und damit die Zugriffsanfragen bearbeiten.

2. Hardware-Komponenten

- **ESP32** Entwicklungsboard
- **RFID-Reader RC522**
- **Display** OLED Display
- **LEDs** für Statusanzeige (rot/grün)



3. Funktionsablauf

3.1 RFID-Initialisierung (separater Programmaufruf)

1. Eine RFID-Karte oder -Transponder wird aufgelegt.
2. Daten werden auf Karte oder Transponder gespeichert.

3.2 RFID-Scan

1. Eine RFID-Karte oder -Transponder wird aufgelegt.
2. Der ESP32 liest die Personalnummer aus.
3. Bei keiner Verbindung zum Server wird die Personalnummer gegen die lokale Liste auf dem ESP32 geprüft

3.2 Kommunikation mit dem Server

1. Der ESP32 verbindet sich mit einem WLAN-Netzwerk.
2. Aufbau einer **TCP-Verbindung** zum Server.
3. Die gelesene Personalnummer wird an den Server gesendet.

4. Server-Funktionen

Der Server:

- prüft die gesendete Personalnummer gegen eine **Zutrittsliste**
- sendet an den ESP32 eines der Ergebnisse zurück:
 - "ALLOW" – Zugang erlaubt
 - "DENY" – Zugang verweigert
- Ausgabe aller Zutrittsversuche im Terminal

Zusätzlich kann der Server:

- ein **Sperr-Zeitfenster** setzen, in dem grundsätzlich niemand Zutritt erhält
- diesen Status jederzeit an den ESP32 übermitteln
- wird über Befehlseingabe im Terminal des Servers realisiert

Befehl	Ausgabe
update_local_list	Synchronisiert die lokale Liste auf dem ESP32 EEPROM mit der globalen des Servers
set_lock_start(x, y)	Setzt den Beginn des Sperr-Zeitfensters auf x:y Uhr (z.B. set_lock_start(8, 45) -> 8:45 Uhr)
set_lock_end(x, y)	Setzt das Ende des Sperr-Zeitfensters auf x:y Uhr (z.B. set_lock_end(17, 15) -> 17:15 Uhr)

5. Anzeige auf dem Display

Der ESP32 stellt die Informationen wie folgt dar:

Zustand	Anzeige auf Display
Zugang erlaubt	„ACCESS GRANTED“
Zugang verweigert	„ACCESS DENIED“
Sperrzeit aktiv	„ACCES BLOCKED“
Warten / Initialisierung	"Place your card on the reader"

Zusätzlich informiert der ESP32 den User, wenn keine Serverkommunikation stattfindet, also die lokale Liste verwendet wird:

Zustand	Anzeige auf Display
Mit Server verbunden	-
Keine Serververbindung möglich	„NO CONNECTION TO SERVER“

6. Software-Komponenten (MicroPython)

Auf ESP32:

- WLAN-Verbindung
- TCP-Client
- MFRC522-Treiber (MicroPython-Bibliothek)
- I2C-Display-Bibliothek
- Hauptprogramm für Ablaufsteuerung:
 - RFID lesen
 - Serververbindung prüfen
 - Daten versenden oder mit lokaler Liste abgleichen
 - Serverantwort auswerten
 - Display aktualisieren

Auf dem Server:

- TCP-Server
- Verwaltung der Zulassungsliste
- Prüflogik für Personalnummern
- Logik für Zeitfenster-Sperre
- Rücksenden des Status an den ESP32