

IoT Backend Széf – Komplettn Dokumentáció

Áttekintés

Ez a projekt egy **IoT alapú széf** megvalósítását célozza, amely három fő komponensből áll:

1. **Frontend:** A széf állapotának megjelenítésére és kódbevitelére szolgáló vizuális felület.
2. **Backend:** Az API végpontok kezelése, az adatok feldolgozása és a széf logikájának működtetése.
3. **ESP32 Interfész:** Fizikai gombokkal történő vezérlés, amely a backend szerverhez csatlakozik Wi-Fi kapcsolaton keresztül.

A rendszer célja, hogy a felhasználók egyszerűen adhassanak meg egy négyjegyű kódot, amely alapján a széf nyitása vagy zárása történik. A rendszer támogatja a reset funkciót, illetve az ESP32-ről vagy a frontendről érkező kódbevitel ellenőrzését.

1. Frontend

Főbb Funkciók

1. **Kódmezők és Állapot Vizualizációja:**
 - Négy input mező (**első**, **masodik**, **harmadik**, **negyedik**) jeleníti meg az aktuális kódot.
 - Egy kép (**kep**) mutatja a széf állapotát:
 - **Zárva:** **close.jpg**.
 - **Nyitva:** **open.jpg**.
 2. **Automatikus Szinkronizáció:**
 - Egy JavaScript alapú időzítő (**setInterval**) 1 másodpercenként szinkronizálja az adatokat a backenddel.
 - A **/numbers**, **/is_checked**, és **/numbers/check** végpontokat használja.
 3. **Felhasználói Élménynövelés:**
 - Automatikus állapotfrissítés.
 - Hibaüzenet (**alert**), ha a megadott kód hibás.
-

Backend

Főbb Funkciók

1. **API Végpontok:**

- A felhasználói kód (**userNumbers**) kezelésére és a biztonságos kód (**secureNumbers**) ellenőrzésére szolgáló HTTP végpontok.
 - Reset és állapotlekérdezés funkciók.
2. **Végpontok Leírása:**
- **GET /numbers**: Az aktuális felhasználói kód lekérdezése.
 - **GET /numbers/first/:value, second, third, fourth**: Egy adott számjegy módosítása.
 - **GET /numbers/reset**: A felhasználói kód alaphelyzetbe állítása.
 - **GET /numbers/check**: A bevitt kód ellenőrzése.
 - **GET /is_checked**: Ellenőrzési állapot lekérdezése.
3. **Express.js és CORS:**
- Az **Express.js** keretrendszer és a **CORS** biztosítja az API működését és az ESP32, valamint a frontend csatlakozását.
-

3. ESP32 Interfész

Főbb Funkciók

1. **Wi-Fi Kapcsolat:**
 - Az ESP32 csatlakozik a Wi-Fi hálózathoz, és HTTPS-en keresztül kommunikál a backenddel.
 2. **Fizikai Gombok:**
 - Hat gombot kezel:
 - Négy számjegy bevitelére.
 - Reset (**/numbers/reset**).
 - Ellenőrzés (**/numbers/check**).
 3. **HTTP Kérések:**
 - Az ESP32 GET kéréseket küld a backend végpontokra a felhasználói interakciók alapján.
-

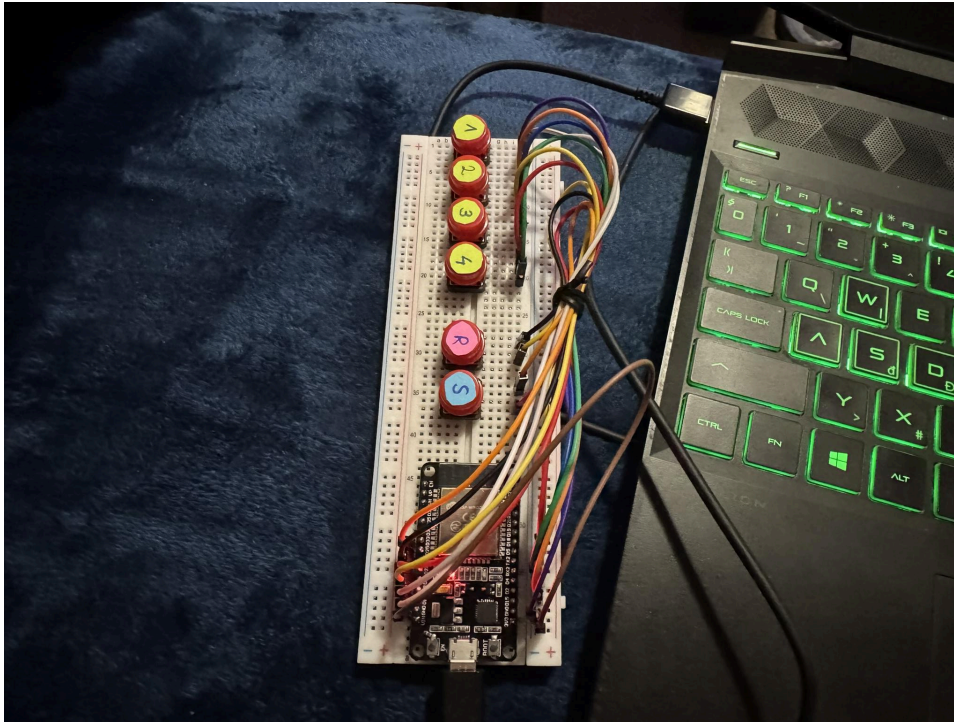
ESP32 Kód:

A kódot az Arduino IDE-ben kell feltölteni az ESP32-re. Tartalmazza a Wi-Fi konfigurációt, a gombnyomások feldolgozását és a backendhez való kapcsolódást.

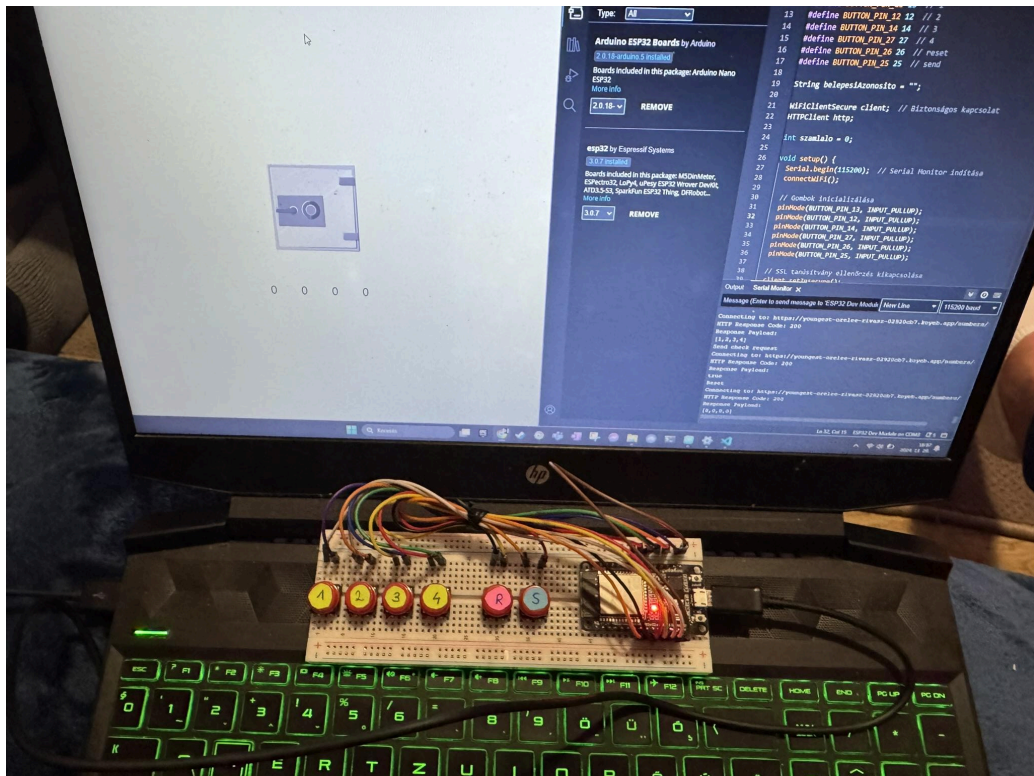
Integrációs Adatfolyam

1. **ESP32 vagy Frontend Interakció:**
 - Kérések érkeznek a backend API-hoz.
2. **Backend Feldolgozás:**

- Az aktuális állapot frissítése és válasz küldése.
3. **Frontend vagy ESP32 Visszajelzés:**
- A széf állapotának vizuális megjelenítése vagy soros monitorra történő kiírása.



4.



5.