



ZÁRÓDOLGOZAT

OKOS OTTHON VEZÉRLŐ RENDSZER

Tanuló

DÖME BALÁZS MÁRK

Mentor

SZABÓ BEATRIX

Mórahalom, 2024



Tartalomjegyzék

Bevezetés	3
Projekt céljának ismertetése.....	4
Fejlesztői dokumentáció.....	5
Rendszerterv.....	5
Az alkalmazás ismertetése	6
Felhasználói dokumentáció	9
<i>Letöltés, telepítés és indítás</i>	<i>9</i>
<i>Program használata.....</i>	<i>9</i>
2.3.1. Főmenü	9
2.3.2. Statisztikák nyomon követése	9
Összefoglalás.....	10



Bevezetés

Táv vezérelt rendszer egy átfogó megoldást kínál a felhasználók számára, lehetővé téve számukra, hogy bármikor és bárhol egyszerűen és kényelmesen irányítsák különböző eszközeiket. Az innovatív platformunk révén nemcsak a lámpákat vagy a 3D nyomtatókat tudják távolról kezelni, hanem számos más eszközt is, amelyek mindennapi életünk részét képezik. A webes felület rendkívül felhasználóbarát, lehetővé téve az egyszerű navigációt és a funkciók könnyű elérését mind asztali számítógépről, mind mobilról. A rendszer telepítése és konfigurálása rendkívül egyszerű és gyors, és biztosítja a hosszú távú, megbízható működést, minimalizálva a felhasználókra nehezedő terheket.

A jövőbeli fejlesztéseink között szerepel egy statisztikai modul bevezetése, amely nemcsak segíti nyomon követni az eszközök használatát, hanem értékes adatokat is biztosít a felhasználásról és a fogyasztásról. Ezáltal lehetővé válik az okos döntések meghozatala és az erőforrások hatékonyabb felhasználása. Emellett terveink között szerepel az automatizált kapcsolási lehetőségek továbbfejlesztése az adatok alapján, amelyek segítségével a rendszer még inkább testreszabhatóvá és intelligensebbé válik. Ez a fejlesztés tovább növeli a rendszer funkcionalitását és a felhasználók kényelmét, hozzájárulva a mindennapi életük könnyedségéhez és hatékonyságához.



Projekt céljának ismertetése

A projekt célja egy olyan szoftver létrehozása volt, amely megkönnyíti a lámpák és munkagépek mindennapi ki- és bekapcsolását. A projektekben hasonló alkalmazások hiányoznak, ezért a projekt egyedi megoldást kínál erre a problémára. Az alkalmazás több programozási nyelven készült, hogy a legjobb megoldást nyújtsa a feladatra. A felhasznált nyelvek közé tartoznak a Golang, JavaScript, C/C++, és az adatbázis motor MySQL. Az alkalmazás egy virtualizált környezetben fut, ami dockerizációs technológiát alkalmazva könnyen indítható bármely számítógépen. Bár a fejlesztés főként GNU/Linux alapú rendszeren történt, a Windows és MacOS felhasználók is tudják használni.

A program számos funkciót tartalmaz:

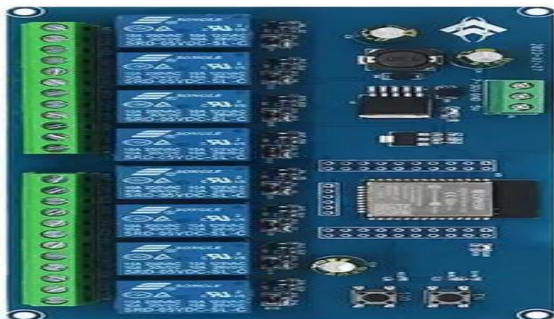
- Távoli vezérlés lámpák és munkagépek számára.
- Adatok tárolása adatbázisban, lehetőséget adva a statisztikák és elemzések kezelésére.
- Biztonságos adattovábbítás WIFI-n keresztül, XOR kulcs használatával a vezérlő és a szerver között.
- Skálázhatóság, maximum 255 eszköz vezérlésére tervezve, de nagyobb igény esetén könnyen bővíthető nagyobb vezérlővel.



Fejlesztői dokumentáció

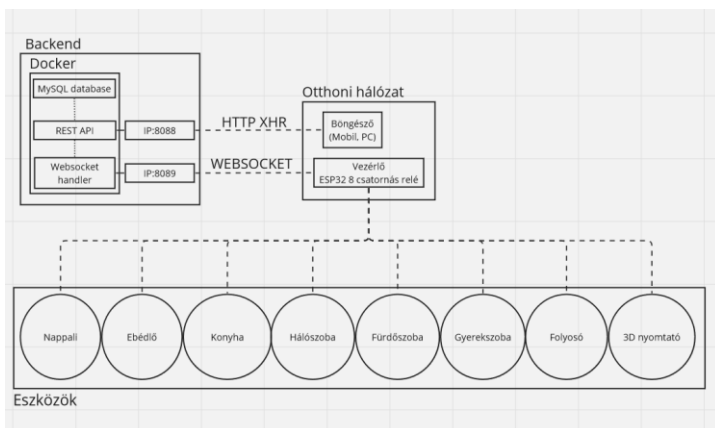
A szerverhez kapcsolódó kliens egy ESP32 mikrokontrollerrel felszerelt relé vezérlő. Ez az eszköz alkalmas a háztartási áramköri vezérlésre, mivel a relék 250V feszültséget bírnak el. Az ESP32 beépített Wifi és Bluetooth modullal rendelkezik, ami könnyűvé teszi a mikrokontroller programozását és integrálását különböző projektekbe. Az ESP32 és az ESP266 vezérlők népszerűek az alacsony áruk, széles támogatottságuk és a fejlesztőközösség által készített könyvtárak miatt.

A fejlesztés során egy 8 vagy akár 32 relével felszerelt vezérlőt használtunk, ami megfelel egy átlagos háztartás vagy kisebb műhely igényeinek. A vezérlő programozása Arduino keretrendszerrel történt, ami C/C++ alapokon nyugszik. Az Arduino keretrendszerben a `setup()` és `loop()`



függvények használata kötelező. A PlatformIO nevű szoftvercsomagot alkalmaztuk a fejlesztéshez, amely nyílt forráskódú és lehetővé teszi a beágyazott rendszerek fejlesztését és kezelését. PlatformIO támogatja többek között az Arduino, ESP32, STM32 és más mikrovezérlő platformokat, így egyszerűsíti a fejlesztést és integrációt.

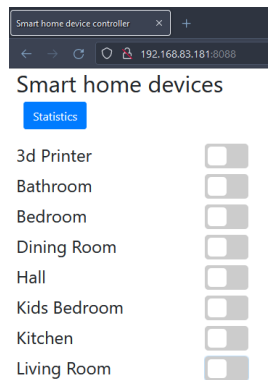
Rendszerterv





Az alkalmazás ismertetése

Az alkalmazás böngészőn keresztüli betöltése lehetővé teszi, hogy bármilyen eszközről, ami rendelkezik böngészővel - például PC, laptop vagy mobil eszközök -, hozzáférjen a szolgáltatáshoz a wifi hálózaton keresztül. A szolgáltatás futtatható Raspberry Pi-on vagy más számítógépen is.



[SB1] megjegyzést írt: Te beszélj a mobil apáról! Tedd be a kódját, magyarázd el!

[DBM2R1] megjegyzést írt: Nincs mobil app responsive webapp van

Összességében a következőkben bemutatott kód részlet leírja a betöltés működését.

```
function init() {
  getDevices()
    .then(data => {
      if (data !== null) {
        const rooms = JSON.parse(data);
        for (let i = 0; i < rooms.length; i++) {
          addDevice(i, rooms[i].device);
          if (rooms[i].state === true) {
            setSliderState(makeStringFancy(rooms[i].device), rooms[i].state)
          }
        }
      }
    });
}

function getDevices() {
  return fetch('/api/getDevices')
    .then(response => {
      if (!response.ok) {
        throw new Error('HTTP error! State: ${response.state}');
      }
      return response.text();
    })
    .then(data => {
      return data;
    })
    .catch(error => {
      console.error('Error making the GET request: ${error.message}');
      return null;
    });
}
```



Az oldal betöltésekor meghívjuk az `initMain()` függvényt. Az `initMain()` függvény meghívja a `getDevices()` nevű függvényt ami lekérdezi az eszközök neveit, illetve állásukat. Állásuk lehet bekapcsolt(true) és kikapcsolt(false). Egy ilyen lekérdezésre egy JSON tömböt ad vissza a backend amely JSON objektumokat tartalmaz a következő módon.

```
JSON
▶ 0: Object { device: "3d_printer", state: false }
▶ 1: Object { device: "bathroom", state: false }
▶ 2: Object { device: "bedroom", state: false }
▶ 3: Object { device: "dining_room", state: false }
▶ 4: Object { device: "hall", state: false }
▶ 5: Object { device: "kids_bedroom", state: false }
▶ 6: Object { device: "kitchen", state: false }
▶ 7: Object { device: "living_room", state: false }
```

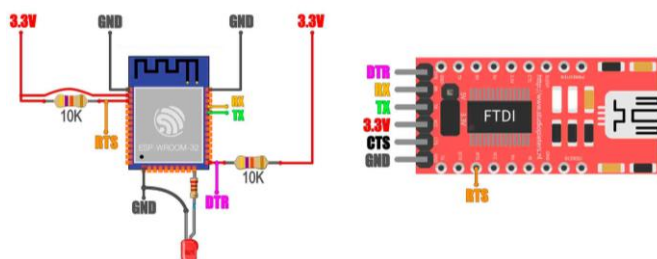
A lekérdezést követően abban az esetben ha a lekérdezés státusz kódja 200 tehát sikeres visszaadja az `initMain()` függvénynek a tömböt amin az `initMain()` függvény végig iterál és az `addDevice()` függvény segítségével a kapott értékek alapján létrehoz DOM objektumokat amik mindegyike tartalmazza az eszköz nevét és egy csúszka gombot ami segítségével az eszköz áramellátása változtatható.

Minden gombhoz tartozik egy `EventListener`, ami lehetővé teszi, hogy csak a gomb megnyomásával tudjunk változtatást létrehozni, nincs szükség küldés gombra.

Programozás előtt fel kell kábelezni a vezérlőt, mivel nincs beépített programozó rajta költségcsökkentés miatt. Szükség van egy FTDI vezérlőre, amit rákötünk az ESP32 lábaira, azon keresztül lehet már lefordított bináris fájlokat feltölteni rá.

A bekötése következő ábra szerint kell történnjen.

- GND földelés, közös mindenhol
- 3.3V, ezt szokás VCC-nek is jelölni bár azt inkább 5v-on
- RTS reset teljes nevén, arra szolgál hogy az FTDI programozó boot módba tudja rakni az ESP32-ot, mivel csak akkor lehet flasht írni
- TX transmit teljes nevén, UART protocolon ezen keresztül továbbít adatot, fogadó oldalon RX-be kell kötni
- RX receive teljes neve, UART protocolon ezen keresztül fogad adatot, adó oldalon TX-be kell kötni
- DTR használata opcionális(olyan logikát valósít meg ami az FTDI modulban már bele van építve)
- CTS-t nem kell használni, mivel UART-ot használunk, nem a komplexebb UART CTS-t





Felhasználói dokumentáció

Az alkalmazás célja az otthonok és kisebb műhelyek mindennapi működésének könnyebbé tétele. Egyszeri beüzemelés után a szolgáltatás könnyen használható. A szerver Golang nyelven, míg a vezérlő C/C++ nyelven lett fejlesztve, hogy a hardverközeli megvalósítást hatékonyan kezelje. Adatbázisként MySQL-t választottunk, de igény szerint könnyen lecserélhető.

Rendszer követelmény:

A szerver futtatásához a következő hardver követelményeknek kell megfelelnie:

- Operációs rendszer: Windows 7 vagy újabb / GNU/Linux 5.0
- Processzor: 500 MHz-es vagy gyorsabb processzor
- Memória: Legalább 500 MB RAM
- Tárhely: Legalább 2 GB tárhely

A vezérlő célhardverre lett tervezve, így annak a követelményei nem változnak.

Letöltés, telepítés és indítás

Szakember segítségével egyszeri felszerelés és konfiguráció után rögtön igénybe vehető a szolgáltatás.

Program használata

Az Okos Otthon Vezérlő Rendszer használata az alábbiak szerint történik:

2.3.1. Főmenü

Az alkalmazás használatához meg kell nyitni egy böngészőben a szerveren futó weboldalt. Ez belső hálózaton elérhető a szerver IP címét illetve a portot beírva a böngészőbe.

2.3.2. Statisztikák nyomon követése

A "Statistics" menüpontban követheti nyomon a statisztikákat. Ezek tartalmazzák az elmúlt 24 órában az eszközök összegzett futási idejüket és oszlopdiagram segítségével vizualizálja.

Szegedi SZC
Tóth János Mórahalmi Szakképző Iskola és Szilágyi Mihály Kollégium



Összefoglalás

A fejlesztői dokumentáció bemutatja a szoftver funkcióit, a program és a hardver működését. A program használatát ismertettem, minden lépést és funkciót áttekintettem. A leegyszerűsített felület nem hozzáértő felhasználók számára is egyértelművé teszi a szoftvercsomag használatát. A projekt további életében lesznek kisebb fejlesztések és javítások. A rendszer használatba lesz helyezve a projekt leadása után, lévén egy valós probléma megoldása.