

Felhasználói útmutató

Az alábbi felhasználói útmutató segítségével könnyen működésre bírhatja a projektet.

Előkövetelmények

Telegram

A kommunikációhoz szükségünk van egy Telegram bot felhasználó és csoport létrehozására.

Bot felhasználó létrehozása

1. Kezdjünk egy új beszélgetést a **@BotFather** nevű felhasználóval!
2. Adjuk ki a **/newbot** parancsot, majd kövessük a lépéseket!
3. Ezután kapunk egy tokenet (pl. **4839574812:AAFD39kkdpWt3ywyRZergyOLMaJhac60qc**), ami lehetővé teszi, hogy az API-n keresztül használjuk az új botunkat.
4. Őrizzük meg a tokenet, a környezeti változók beállításakor szükségünk lesz rá!

Csoport létrehozása

1. Az alkalmazásban hozzunk létre egy új csoportot, tetszőleges névvel!
2. Adjuk hozzá az előző pontban létrehozott botunkat!
3. Kezdjünk egy új beszélgetést **@RawDataBot**-tal, majd nyomjunk a **Start** gombra!
4. A JSON válaszból keressük meg az **id** kulcsot a megfelelő objektumban (ahol a **name** megegyezik a csoportnévvel)!
5. Őrizzük meg az **id** kulcs értékét, a környezeti változók beállításakor szükségünk lesz rá!

Cloudflare

Megjegyzés: Szükségünk lesz egy általunk birtokolt domainre!

1. Regisztráljunk egy Cloudflare fiókot [ezen](#) a linken!
2. A **Websites** fül alatt nyomjunk rá az **Add site** gombra!
3. Adjuk meg a domainünket (pl. [kiskutya.hu](#)), majd nyomjunk rá az **Add site** gombra!
4. A domainregiszterünkénél állítsuk be a Cloudflare DNS szervereit, majd várjuk meg, míg befutnak a változások!

Üzembe helyezés

Boot SD kártya készítése

1. Töltsd le és telepítsd a Raspberry Pi Imager legújabb verzióját!
2. Csatlakoztass egy SD-kártyaolvasót úgy, hogy az SD-kártya benne legyen!
3. Nyisd meg a Raspberry Pi Imagert, és válaszd ki a kívánt operációs rendszert a megjelenő listából!
4. Válaszd ki azt az SD-kártyát, amelyre a képfájlt szeretnéd írni!
5. Állítsa be előre a következőket (**Ctrl+Shift+X**)!
 - a. Wi-Fi
 - b. SSH
 - c. Felhasználónév/jelszó
 - d. Időzóna
 - e. Billentyűzetkiosztás
6. Nézd át a beállításokat, és kattints az **Írás** gombra!

Első boot után

A naprakész operációs rendszer feltétele, hogy a csomagok is frissek legyenek rajta.

Debian-alapú rendszereken ez az `apt` csomagkezelő segítségével történik.

```
sudo apt update -y && sudo apt upgrade -y  
sudo reboot now
```

A cache ürítése és a nem használt package-ek törlése elengedhetetlen, ha szűkös a tárhelye az eszközünknek.

```
sudo apt autoclean -y  
  
sudo apt autoremove -y
```

Cloudflare Tunnel kliens telepítése és konfigurálása

A Telegram által küldött adatok fogadásához szükségünk van a `cloudflared` kliens telepítésére.

Először telepítenünk kell az előkövetelményeket:

- `curl` a CLI webes lekérdezésekhez
- `lsb-release` a megfelelő architektúra automatikus hozzáfűzéséhez

```
sudo apt install curl lsb-release
```

Aptitude repository hozzáadása

```
curl -L https://pkg.cloudflare.com/cloudflare-main.gpg | sudo tee  
/usr/share/keyrings/cloudflare-archive-keyring.gpg >/dev/null  
  
echo "deb [signed-by=/usr/share/keyrings/cloudflare-archive-  
keyring.gpg] https://pkg.cloudflare.com/cloudflared $(lsb_release -  
cs) main" | sudo tee /etc/apt/sources.list.d/cloudflared.list
```

Csomaglisták frissítése

```
sudo apt update -y
```

cloudflared csomag telepítése

```
sudo apt install cloudflared
```

Ezután be kell lépünk a Cloudflare fiókunkkal, amivel felvettük a domainünket.

```
cloudflared tunnel login
```

A terminálablakban megjelenő linken való belépés után válasszuk ki a domaineink között azt, amihez társítani akarjuk majd a tunneleinket.

Ezután le is generálódik a `/home/<USER>/.cloudflared` könyvtárban a `cert.pem` nevű tanúsítvány, amellyel azonosítani fog minket a program a továbbiakban.

Elérkeztünk a lényegi részre, létre is hozhatjuk a tunnelünket!

```
cloudflared tunnel create <TUNNEL_NAME>
```

Git repo klónozása

git kliens telepítése, ha még nem lenne fent.

```
sudo apt install git
```

Klónozzuk le a repókat, jelen esetben az `/opt/buliszerviz` könyvtárba.

```
git clone https://github.com/szkly/sze-mikroelektro-buliszerviz.git
```

```
sudo mv sze-mikroelektro-buliszerviz /opt/buliszerviz
```

Telepítsük fel a **pip**-et, a Python csomagkezelőjét a **venv** nevű Python csomaggal együtt.

```
sudo python3 -m ensurepip --upgrade
```

```
sudo pip install venv
```

Ezután feltelepíthetjük a webalkalmazás Python függőségeit, melyek verzióval ellátva a **/opt/buliszerviz/webapp/requirements.txt** fájlban találhatóak.

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
sudo pip install -r requirements.txt
```

A frontend függőségek kezeléséhez szükségünk lesz a Node.js-re és az **npm** csomagkezelőre.

```
sudo apt install nodejs npm -y
```

Függőségek telepítése a **package.json** leíró alapján:

```
cd /opt/buliszerviz
```

```
npm i --prefix webapp/src
```

Adafruit PiOLED beállítás és tesztelés

Csomagok telepítése és az I2C bus beállítása

```
sudo apt install -y i2c-tools libgpiod-dev
```

```
ls /dev/i2c* /dev/spi*
```

```
sudo i2cdetect -y 1
```

Tesztprogram futtatása

```
cd /opt/buliszerviz
```

```
python3 tests/oled.py
```

TowerPro SG92R szervomotor teszt

Tesztprogram futtatása

```
cd /opt/buliszerviz
```

```
python3 tests/servo.py
```

LED teszt

Tesztprogram futtatása

```
cd /opt/buliszerviz
```

```
python3 tests/led.py
```

Bluetooth hangszóró beállítása

Bluetooth konfigurációs fájl módosítása, hogy automatikusan elinduljon a BT kontroller

```
vim /etc/bluetooth/main.conf
```

```
Enable=Source,Sink,Media,Socket
```

```
[Policy]
```

```
AutoEnable=true #Auto enable BT controller on boot
```

Hangszóró párosítása

```
bluetoothctl
```

```
power on
```

```
discoverable on
```

```
scan on
```

```
pair 70:99:1C:2E:6B:6C
```

```
connect 70:99:1C:2E:6B:6C
```

```
trust 70:99:1C:2E:6B:6C
```

```
power off
```

Audioprotokoll beállítása

```
pactl set-card-profile bluez_card.70_99_1C_2E_6B_6C a2dp_sink
```

Végül újra bekapcsoljuk a kontrollert

```
bluetoothctl
```

```
power on
```

Környezeti változók beállítása

Másoljuk le a mintafájlt

```
cd /opt/buliszerviz/webapp/src
```

```
cp .env.example .env
```

Szerkesszük meg a kedvenc szerkesztőnkkel a `.env` fájlt

```
FLASK_DB_FILE_PATH="db/party_event.db"
```

```
FLASK_TELEGRAM_BOT_TOKEN="XXXXXXXXXX:AAAABBBBCCCCZZZZDDDDDEEEEEEGGGGGG"
```

```
FLASK_TELEGRAM_WEBHOOK_URL="https://<TUNNEL_NAME>.<DOMAIN>"
```

```
FLASK_TELEGRAM_CHAT_ID="<TELEGRAM_GROUP_ID>"
```

```
FLASK_SONG_FILE_PATH="static/party_pi.mp3"
```

Alkalmazás indítása

A `package.json`-ban definiált `prod` script indítása az `npm` segítségével

```
cd /opt/buliszerviz
```

```
npm run prod --prefix webapp/src
```

Telegram webhook beállítása

```
curl https://<TUNNEL_NAME>.<DOMAIN>/set-webhook
```

Használat

Felhasználó

1. Nyisd meg [ezt](#) a weboldalt!
2. Írd be a neved a megfelelő mezőbe!
3. Ha még nem próbálkoztál, akkor várd meg, míg az adminok döntenek!
4. Kérelmed elfogadása esetén 5 másodperc áll a rendelkezésedre, hogy belépj, ezek után az ajtó automatikusan záródik.

Új admin

1. Töltsd le a [Telegram](#) alkalmazást!
2. Az admin csoportba való meghívás után fogadd el a meghívót!
3. Belépési kérelemről szóló értesítés esetén dönts a felugró gombok segítségével!
4. A rendszer ezután értesíti a felhasználót a döntésről.