**TFS CANSAT vizualizáció**

A képen szöveg, képernyőkép, térkép, Multimédiás szoftver látható

Automatikusan generált leírás

A vizualizációt vezérlő HTML kód egy interaktív weboldalt valósít meg, amely egy térképet, egy grafikonablakot, valamint az utolsó 10 GPS, P, T, RN, adatot tartalmazó listát jelenít meg. A megoldás dinamikusan frissíti az adatokat egy LORA rádióvevő által generált forrásból (adatok.txt), és az új adatokat a következőképpen dolgozza fel és jeleníti meg:

**Főbb Funkciók**

1. **Térkép (OpenStreetMap):**
   * **A kód a Leaflet.js könyvtárat használja a térkép megjelenítéséhez.**
   * **A térképen kék és piros pontok jelennek meg:**
     + **Kék pontok: az aktuális GPS koordináták pozíciói.**
     + **Piros pontok: a relatív magasságértékek vizualizálására használt pontok.**
   * **A kék és piros pontok között vonalak kötik össze az eredeti pozíciót és a relatív magasságot jelző pontot.**
2. **Grafikon (Chart.js):**
   * **A grafikon az utolsó 10 GPS adat magasságértékét jeleníti meg.**
   * **Az x tengelyen az egyes adatsorok egyszerű sorszámozása található (pl. Adatsor 1, Adatsor 2).**
   * **Az y tengely a magasságértékeket ábrázolja.**
   * **A jelmagyarázat és tengelycímek testreszabhatók vagy eltávolíthatók.**
3. **Adatsorok listája:**
   * **Az utolsó 10 GPS adat nyers szöveges formában jelenik meg egy külön ablakban.**
   * **Ez az adatblokk segít a grafikon és a térképi megjelenítések gyors ellenőrzésében.**
4. **Automatikus frissítés:**
   * **A kód 0,5 másodpercenként (setInterval) frissíti az adatokat az adatok.txt fájlból.**
   * **Az új adatok alapján frissül a térkép, a grafikon és az adatsorok listája is.**

**Felhasznált Könyvtárak**

1. **Leaflet.js: a térképkezeléshez.**
2. **Chart.js: a grafikon megjelenítéséhez.**
3. **jQuery: az adatbetöltéshez (fetch alternatívaként).**

**Főbb Komponensek**

1. **HTML Elemei:**
   * **Térkép: Az id="map" elem tartalmazza a térképet.**
   * **Grafikon: Az id="chart-container" tartalmazza a grafikon ablakát és egy <canvas> elemet a Chart.js számára.**
   * **Adatok listája: Az id="data-container" elem jeleníti meg a szöveges adatokat.**
2. **CSS Stílusok:**
   * **A komponensek egymástól elkülönülten, jól strukturáltan jelennek meg.**
   * **A grafikon és az adatsorok listája a jobb oldalon helyezkednek el, egymás alatt.**
3. **JavaScript Működése:**
   * **initializeMap: A térkép inicializálása.**
   * **initializeChart: A Chart.js diagram inicializálása.**
   * **updateData: Az adatok lekérése az adatok.txt fájlból, majd ezek továbbítása a megfelelő frissítési funkcióknak:**
     + **updateRoute: A térkép kék és piros pontjainak frissítése.**
     + **updateChart: A grafikon adatainak frissítése az utolsó 10 magasságértékkel.**
     + **updateLast10Lines: Az utolsó 10 adatsor szöveges megjelenítése.**
   * **Az adatfrissítési logika hibakezelést is tartalmaz, amely tájékoztat a betöltési problémákról.**

**HTML Kód Összefoglalása és Működése**

Ez a HTML kód a Leaflet.js használatával egy valós idejű térképet hoz létre, amely a **LORA vevő által gyűjtött GPS-koordinátákat** jeleníti meg, és frissíti a térképen az útvonalat.

**Fő Részei**

**1. Térkép Inicializálása**

* **initializeMap() függvény:**
  + Létrehozza a térképet a #map div elemen belül.
  + Beállítja az alapértelmezett középpontot:
    - **[47.500585, 19.258504]**: Ez Budapest (Petri u. 30 környéke).
  + OpenStreetMap csempéket használ, amelyeket a Leaflet.js-en keresztül tölt be.
  + Inicializál egy Polyline objektumot az útvonal megjelenítéséhez (kék vonal).

**3. Automatikus Frissítés**

* A térkép 0,5 másodpercenként (setInterval(updateData, 500)) automatikusan lekéri a legfrissebb adatokat, és frissíti a térképet.

**Vizualizált Működés**

1. **Útvonal Kirajzolása:**
   * Az összes GPS-koordinátából egy pontokat vonalat rajzol a térképre, amely az adó mozgásának útvonalát mutatja.
2. **Marker Frissítése:**
   * Az aktuális pozícióhoz tartozó marker frissül az utolsó koordináta alapján.

**Tesztelés**

1. **Alapértelmezett Pozíció:**
   * A térkép betöltéskor a Petri utca környékére mutat (Budapest, 47.500585, 19.258504).
2. **Útvonal Frissítése:**
   * Mozgasd az adót, és ellenőrizd, hogy az új koordináták megjelennek-e az útvonalon.
3. **Konzol Hibák Ellenőrzése:**
   * Nyisd meg a böngésző fejlesztői eszközeit (F12 > Console), hogy lásd, ha bármilyen hiba történik a JSON fájl elérésekor.