

Stredná priemyselná škola - Ipari Szakközépiskola

Petőfiho 2, Komárno

KOMPLEX SZAKDOLGOZAT

Tanulmányi szak: elektrotechnika 2675 M 00

Weboldal/Program: Okos otthon vezérlés

Kidolgozta: Keszeg Balint

Évfolyam: 4.G

Iskolai év: 2019/2023

Konzultáns: Ing. Štefan Bucz, PhD.

Názov školy: Stredná priemyselná škola strojnícka a elektrotechnická -
Gépipari és Elektrotechnikai Szakközépiskola, Petőfiho 2, Komárno
Školský rok: 2022/2023

Praktická časť odbornej zložky maturitnej skúšky

Forma: Obhajoba vlastného projektu

Študijný odbor:	2675 M elektrotechnika
Dĺžka trvania praktickej úlohy:	20 minút
Miesto konania úlohy:	odborné elektrotechnické laboratórium
Meno maturanta:	Bálint Keszeg, 4.G

Názov témy: Praktická aplikácia softwaru

Zadanie:

1. Navrhnete mikropočítačom podporované ovládanie inteligentnej domácnosti "Smart Home".
2. Vytvorte počítačový 3-D model inteligentnej domácnosti.
3. Navrhnete riadiaci systém inteligentnej domácnosti pomocou webovej stránky.
4. Realizujte nasledovné funkcionality webovej stránky: ovládanie vybraných zariadení, ich stav a diagnostiku inteligentnej domácnosti.
5. Realizujte vhodné softwarové riešenie.
6. Otestujte navrhnuté zariadenie a software.
7. Vytvorte používateľskú dokumentáciu k malému vozidlu.

Pokyny:

1. V dokumentácii uveďte:
 - a. Teoretický rozbor riešenia.
 - b. Vývojový diagram riadenia inteligentnej domácnosti.
 - c. Schému zapojenia celkového zariadenia.
 - d. Počítačový 3-D model a fotografie zariadenia.
 - e. Používateľskú príručku k ovládaniu domácnosti cez webovú stránku.
 - f. Výpis softwaru.
 - g. Zoznam použitej literatúry.
2. Dodržiavajte zásady bezpečnosti a ochrany zdravia pri práci.


Konzultant:


Maturant:


Ing. Ján Vetter
riaditeľ školy

V Komárne dňa 30. 9. 2022

Nyilatkozat!

Én, Keszeg Balint, a Stredná priemyselná škola strojnícka a elektrotechnická-
Gépipari és Elektrotechnikai Szakközépiskola Petőfiho 2, Komarno IV.G osztályos
tanulója kijelentem, hogy a dokumentációban tárgyalt munkát önállóan valósítottam meg
az elejétől a végéig, segítségemre volt a konzultásom Ing.Stefan Butcz PhD. ill. a használt
szakirodalom.

2023.1.30

.....Keszeg

Köszönet!

Ezzel a néhány sorral szeretném kifejezni hálámat a segítségért és a tanácsokért a munkámhoz Ing. Štefan Bucz, a segítsége nélkül sokkal nehezebb dolgom lett volna. Köszönöm szépen!

Bevezető	6
2 Mi is az a “Smart Home” rendszer?	6
2.1 Raspberry	6
2.2 Node-Red	7
2.3 A programhoz felhasznált nyelvek:	7
2.3.1 Java	7
2.3.2 SVG	8
2.3.3 YAML.....	8
2.3.4 JavaScript:.....	9
3 Felhasznált programok:	9
3.1 Node-Red:	9
3.2 Notepad++:.....	9
3.3 Home Assistant Core:	10
3.4 No-IP:	10
4 A munka kivitelezése:	10
4.1 Alap core program létrehozása:	10
4.2 Node-red programozása:	11
4.3 Tokenes bejelentkeztető rendszer:	11
4.4 Könnyen kezelhető UI felület:	11
4.5 Smart Home szerver elindítása:	12
5 Problémák és megoldásuk:	13
6 Oldaltérkép:	13
7 Kiértékelés:.....	13
8 Felhasznált irodalom:	13
9 Mellékletek:	14

¹ Bevezető

Azért kezdtem bele ebbe a munkába mert ihletett kaptam egy Erasmus+ által megrendezett 2 hetes prax alatt tanultak alapján működő programról. Ott bemuták hogy ez a program(Node-Red)mikép vezérel a marógépeket,és ekkor jött számomra az ötlet hogy ezt a technikát és elvet átvihetném a egyszerű éltbe mint egy okos otthon vezérlésében. Plusz a közel áll a téma az érdeklődési körömhöz is ami csak még inkább motivált az elkészítésében.

2 Mi is az a “Smart Home” rendszer?

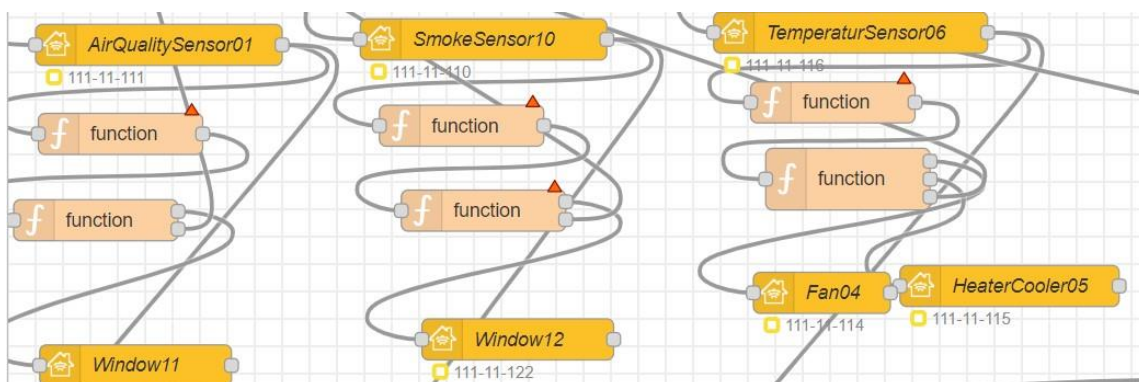
A smart home ill. Okos otthon az egy komplexebb rendszer amit általában egy kisebb mikrovezérlővel lehet irányítani. Többnyire szenzorokból és okos eszközökből áll amelyet a mikrovezérlő egy belső program szerint irányít és elemez. Ilyen folyamatok például a lámpák státuszának érzékelése és kezelése, belső körök érzékelés(elektromos fogyasztás,viz fogasztás),multimédiás eszközök és kamerák vezérlése. Manapság az egyik legfontosabb paraméterei egy ilyen rendszernek az hogy bárholon elérhető legyen és biztonságos legyen egyaránt.

2.1 Raspberry

“A **Raspberry Pi** egy bankkártya méretű, egyetlen áramköri lapra/kártyára integrált BCM2835 alapú egykártyás____számítógép, amelyet az Egyesült Királyságban fejlesztettek oktatási célokra. A gép különböző Linux-disztribúciókkal működtethető, illetve elérhető egy RISC OS verzió is. Az eredeti két *4 Model B*, amely 2019. június 24-én jelent meg. Ez 1,5 GHz-es, 64 bites, négymagos ARM A72 processzorral, integrált WiFi-vel és Bluetooth 5-tel, USB 3 csatlakozóval és 4K-s dupla HDMI monitor csatlakozóval rendelkezik. A Pi 4 az első modell, amely különböző nagyságú RAM-mal szerelve, háromféle változatban (1, 2 és 4 GB) volt elérhető a kiadása idején. Az 1 GB-os változatot később kivették a piacról. 2020. május 28-án megjelent a 8 GB-os verzió”

2.2 Node-Red

“A Node-RED egy vizuális programozásra szolgáló, áramlásalapú fejlesztőeszköz, amelyet eredetileg az IBM fejlesztett ki hardvereszközök, API-k és online szolgáltatások összekapcsolására a dolgok internetének részeként. A Node-RED egy webböngésző-alapú flow-szerkesztőt biztosít, amely JavaScript-funkciók létrehozására használható. Az alkalmazások elemei elmenthetők vagy megoszthatók újrafelhasználás céljából. A futásidő a Node.js-re épül. A Node-RED-ben létrehozott áramlásokat JSON segítségével tárolják.”



2.3 A programhoz felhasznált nyelvek:

2.3.1 Java

A **Java** általános célú, objektumorientált programozási nyelv, amelyet a Sun Microsystems fejlesztett. A Java alkalmazásokat jellemzően *bájtkód* formátumra alakítják, de közvetlenül natív (gépi) kód is készíthető Java forráskódból. A bájtkód futtatása a Java virtuális géppel történik, ami vagy interpretálja a bájtkódot, vagy natív gépi kódot készít belőle, és azt futtatja az adott operációs rendszeren. Létezik közvetlenül Java *bájtkódot* futtató hardver is, az úgynevezett Java processzor.

A Java nyelv a szintaxisát főleg a C és a C++ nyelvektől örökölte, viszont sokkal egyszerűbb objektummodellel rendelkezik, mint a C++. A JavaScript szintaxisa és neve hasonló ugyan a Java-hoz de nem ugyan az a kettő.

2.3.2 SVG

A skálázható vektorgrafika (SVG) egy XML-alapú vektoros képformátum kétdimenziós grafikák definiálására, amely támogatja az interaktivitást és az animációt. Az SVG specifikáció a World Wide Web Consortium által 1999 óta fejlesztett nyílt szabvány.

Az SVG-képeket vektorgrafikus formátumban definiálják és XML-szövegfájlokban tárolják. Az SVG-képek így minőségromlás nélkül méretezhetők, az SVG-fájlok pedig kereshetők, indexelhetők, szkriptelhetők és tömöríthetők. Az XML szövegfájlok szövegszerkesztőkkel vagy vektorgrafikus szerkesztőkkel hozhatók létre és szerkeszthetők, és a leggyakrabban használt webböngészők renderelik őket.

2.3.3 YAML

A YAML egy ember által olvasható adat-sorozataalkotási nyelv. Általában konfigurációs fájlokban és olyan alkalmazásokban használják, ahol adatokat tárolnak vagy továbbítanak. A YAML számos olyan kommunikációs alkalmazást céloz meg, mint az Extensible Markup Language (XML), de minimális szintaxisa szándékosan különbözik a Standard Generalized Markup Language (SGML) nyelvtől. Egyrészt Python-stílusú behúzást használ az egymásba ágyazottság jelzésére, másrészt egy kompaktabb formátumot, amely [...] a listákhoz és { ... } a térképekhez használja, így a JSON fájlok érvényes YAML 1.2..

Az egyéni adattípusok megengedettek, de a YAML natívan skalárokat (például karakterláncokat, egész számokat és lebegőszámokat), listákat és asszociatív tömböket (más néven térképeket, szótárakat vagy hash-eket) kódol. Ezek az adattípusok a Perl programozási nyelven alapulnak. A YAML-t folyamban kell olvasni és írni. A YAML olvasásának és írásának támogatása számos programozási nyelvben elérhető. Néhány forráskód-szerkesztő, például a Vim, az Emacs, és a különböző integrált fejlesztőkörnyezetek rendelkeznek olyan funkciókkal, amelyek megkönnyítik a YAML szerkesztését, például az egymásba ágyazott struktúrák felhajtásával vagy a szintaxis hibák automatikus kiemelésével.

2.3.4 JavaScript:

”A Java napjaink talán legígéretesebb programozási nyelve, melyet a Sun Microsystems és a Netscape Communications Corporation együtt fejlesztettek ki. A Java nyelv nem más, mint egy hatalmas, tisztán objektum-orientált programozási nyelv, melynek nagy ereje, hogy segítségével platformfüggetlen alkalmazások készíthetők. A Java nyelven megírt programokat egy ún. köztes kódra (byte kód) kell lefordítani.

A JavaScript a Java nyelv "kis testvérkéje"- mondják sokan. A JavaScript, mint neve is mutatja, egy script nyelv. A JavaScript-eket HTML lapokba lehet beágyazni, a lappal együtt töltődnek le, majd a böngésző értelmezi és futtatja azokat. Hasonlít a Java-ra, de ugyanakkor sokkal kisebb, egyszerűbb, korlátozottabb működésű is annál. A JavaScript utasításoknak a SCRIPT elemben kell lenniük az alábbi módon: <SCRIPT>

JavaScript-utasítások

</SCRIPT>”^[6]

3 Felhasznált programok:

Munkámban a következő programokkal dolgoztam.

3.1 Node-Red:

A Node-Red egy Java parancsokon alapuló program melyt grafikusán lehet megírni blokk formájában. Itt lehetett megvalósítani magát a vezérlést és a csatlakozást a szenzorok között majd a adatok továbbítását az UI felületre.

3.2 Notepad++:

A Notepad++ egy ún. felturbózott jegyzettömb, amiben több funkciót találunk, és sokkal hasznosabb HTML forráskódok írásához. Ennek oka, hogy a program külön színnel jelöli a HTML parancsokat így átláthatóbb, valamint a Notepad++ egy beírt betű után azonnal lehetőségeket kínál fel, hogy milyen parancsot írhatunk be.

3.3 Home Assistant Core:

A "HAC" maga a motorja az UI nek. Ebben a programban lett megvalósítva a emberfelé kivetített felület. Itt magát megírt YAML fájlokat alakítsa át weblap kinézetre és biztosítja a kezelőfelület működését mint admin, mint felhasználó szinten is.

3.4 No-IP:

A No-IP dinamikus DNS-szolgáltatásokat nyújt. Az alapszolgáltatás ingyenesen használható, amennyiben a felhasználó időről időre frissíti a hozzáférését, és így aktívan tartja azt. A DDNS frissítése vásárlásokból történik. A dinamikus IP-címek gyakoriak a lakossági kábel- vagy DSL-szélessávú számlákon, és a DDNS tipikus felhasználói az ilyen típusú internetkapcsolatok felhasználói lennének. A szolgáltatás lehetővé teszi a felhasználók számára, hogy legfeljebb három hostnevet hozzanak létre egy No-IP tartományon. Windows, OS X és Linux operációs rendszerekhez szoftveres kliensek állnak rendelkezésre, amelyek lehetővé teszik, hogy a DDNS kapcsolódjon ezekhez az operációs rendszerekhez. Az ilyen DDNS-konfigurációkban azonban gyakrabban használnak routereket.

4 A munka kivitelezése:

Az első feladatom a felhasznált programok telepítése és áttanulmányozása volt.

4.1 Alap core program létrehozása:

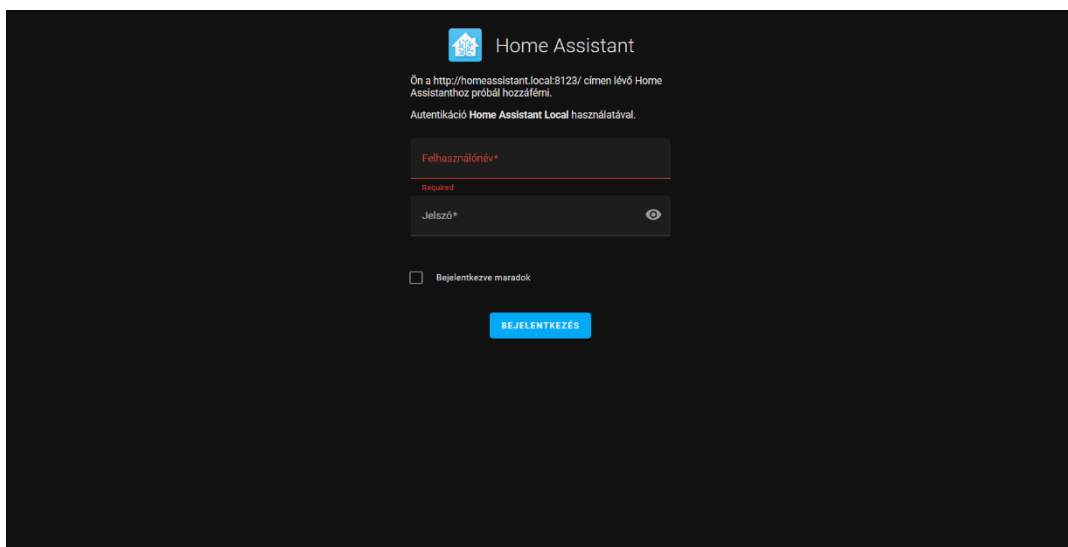
Először is készítettem egy bootolható USB meghajtót melyen maga a Node-Red és Home Assistant Core is futt. Konfiguráltam magát a Raspberryt a hálózatra és megadtam magát a admin bejelentkezési adatokat is.

4.2 Node-red programozása:

Alap programok feltelepítése után nekilátam a vezérlő program megírásának. Külsőleg megírtam a JSON fájlokat majd be importáltam a Node-ba. Kisebb nagyobb módosításokat is végeztem az alap programba hogy optimizáltabb és megbízhatóbb legyen maga majd a kész termék. Mivel az alap program csak a előre telepített modulokat használta csak ezért több kiegészítő modult is csatoltam a fájlok mellé azzal az indokkal hogy a vezérlés több mindenre is képes legyen.

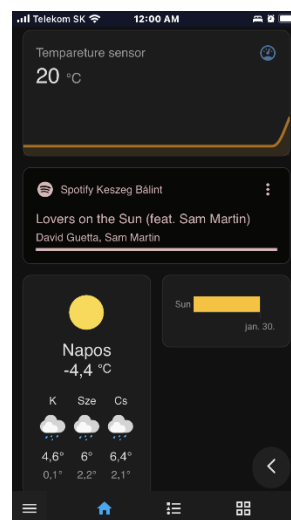
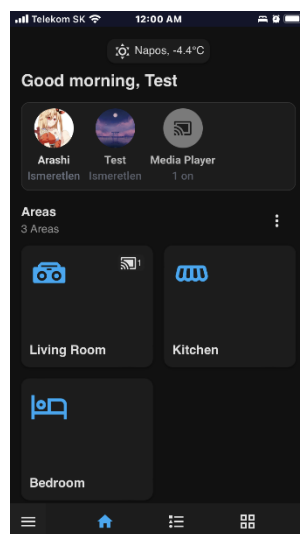
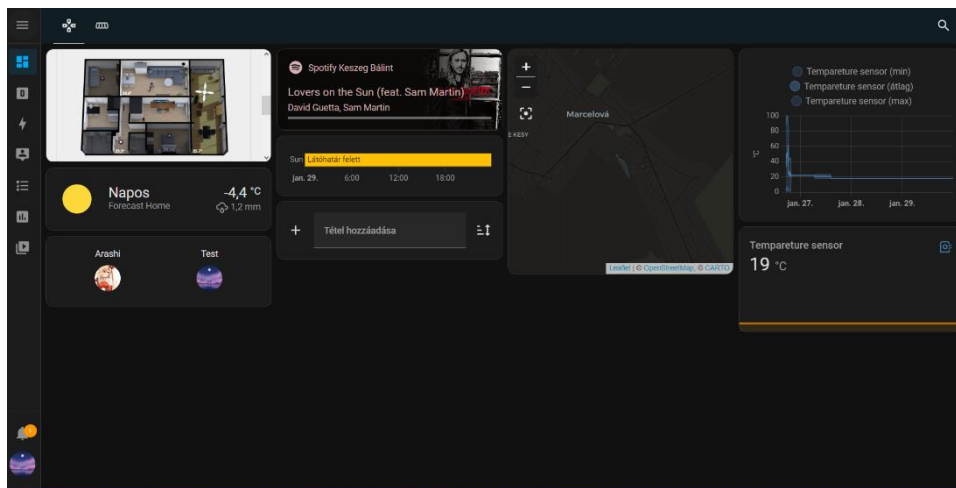
4.3 Tokenes bejelentkeztető rendszer:

Mivel maga a kezelőfelület bárki álltal irányítható volt és biztonság kicsi volt így arra az ötletre jutottam, hogy egy bejelentkeztető rendszert kell létrehoznom. Mivel nem rendelkeztem Email-serverrel ezért egy úgynevezett tokkenes rendszerrel álltam elő. Ez annyit jelent hogy kizárólag a admin tud felhasználó nevek és jelszók mellé egy teljesen véletlen legenerált tokkent létrehozni ami szerint a program beazonosítja a személyt és az szerint alapítsa meg a rangját és az hogy mire jogosult a UI-n.



4.4 Könnyen kezelhető UI felület:

Egy könnyen kezelhető UI hozztam létre mely a laikus felhasználóknak is könnyű kezelést nyújt a programba. Ezt mind a YAML és JavaScript segítségével hozztam létre. Maga a felület metro szerű csempés kinézetel bir. Több csempe rendelkezik felugró ablakokkal is. CSS is került a felületre ami grafikonok és egyéb effektek megjelenítését szolgálja a JavaScriptel együtt. Meglett az is oldva hogy a admin és a egyszerű felhasználó felületei ne legyen egyben és ne férjen a felhasználó a konfigurációs programokhoz.



4.5 Smart Home szerver elindítása:

A szervert egy úgy nevezett Dinamikus ip keresztül működtetek amit a No-IP alkalmazással hoztam létre amely a router keresztül történik meg. Kaptam egy külső Domain címet amellyel elehet a weboldat érni bármilyen külső hálózatról is biztonságosan mivel HTTPS protokollt használ.

5 Problémák és megoldásuk:

Ami nagyobb problémát okozott, az a interaktív térkép és a bárhonan való elérés volt. Elején sokk hibát vétetem mind a 2 terén de egy kisebb utánaolvasás után sikkeresen megoldotam a nehézségeket és még fejleszteni is tudtam a dolgokon.

6 Oldaltérkép:

- Kezdőlap
- Energia fogyasztás
- Multimédiás fül
- Felhasználó
- Egyébb beállítások

7 Kiértékelés:

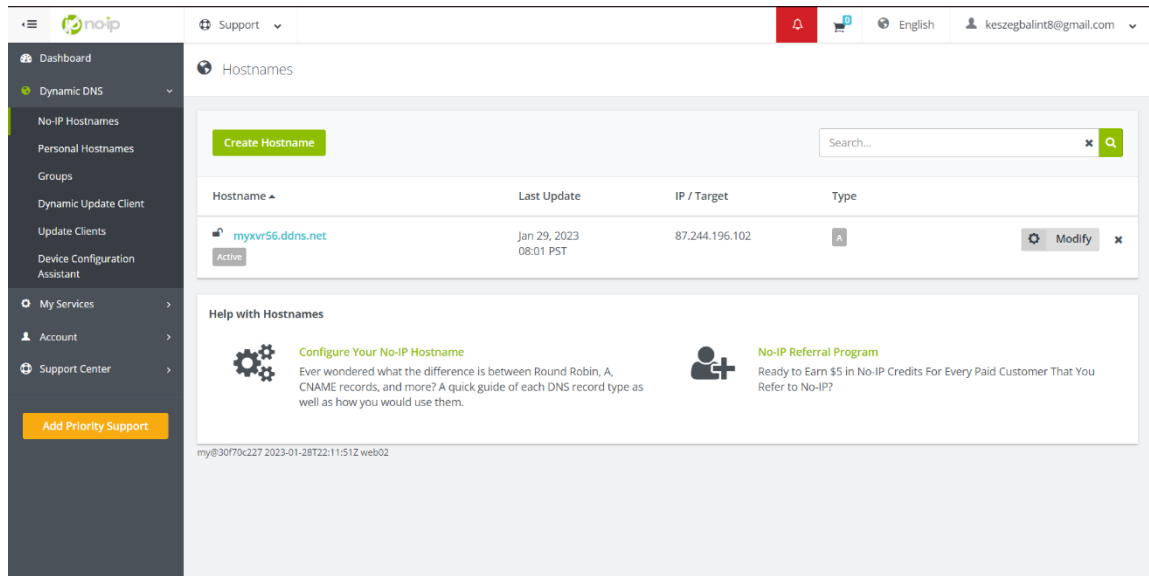
A feladatom sikerült elkészíteni olyan minőségben, amilyenben tőlem telt, a munka kezdete viszonylag egyszerű volt, mivel már találkoztam ez előtt a Node-Red programal, a nehézségek onnan kezdődtek, amikor optimalizálni és elérhetővé kellett tennem a oldalt. Több új nyelvet is sikerült megtanulnom és fejlődöt a megoldó képeségem. Sikerült új technikákat alkalmaznom. Sikeresen megoldotam a feladatban leirtakat és még plusz dolgokat is sikerült beépítenem.

8 Felhasznált irodalom:

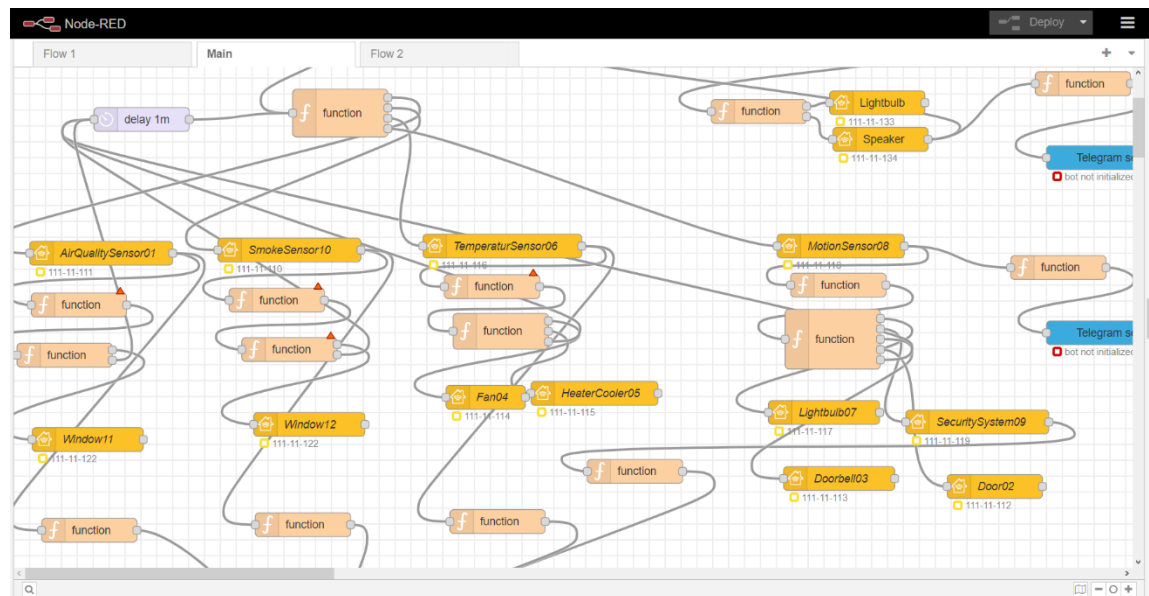
- [1] 2023.01.29 <https://en.wikipedia.org/wiki/Java>
- [2] 2023.01.29 <https://en.wikipedia.org/wiki/SVG>
- [3] 2023.01.29 <https://www.noip.com/support/>
- [4] 2023.01.29 <https://experiencelovelace.github.io/ha-floorplan/>

9 Mellékletek:

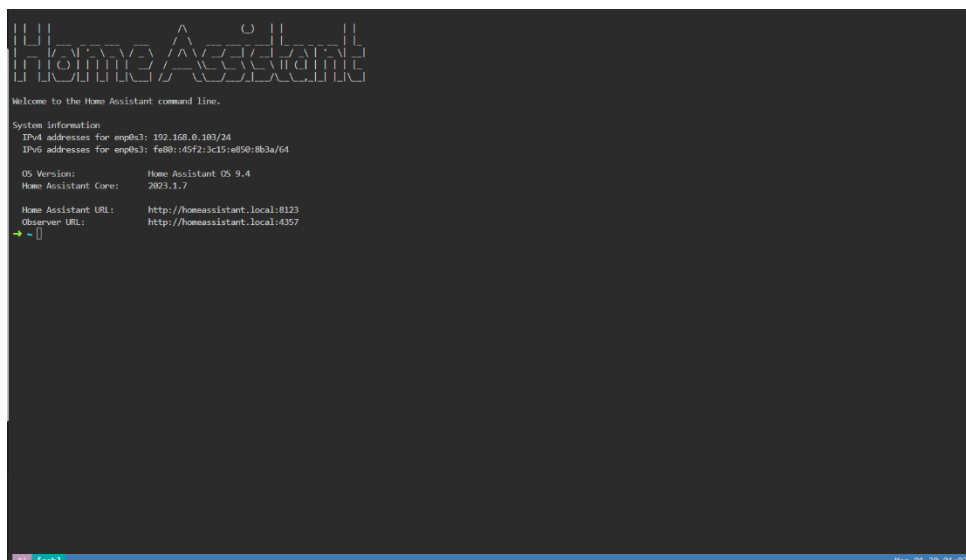
No-IP megvalósítás:



Node-Red kezelőfelület:



Home Assistant Core:



Elérhetősége a weboldalnak:

