Google Home Integráció

Norbert

May 13, 2020

1 Bevezetés

A következő projektbe létrehozunk egy kommunikációs kapcsolatot a PLC CP2000-es készülék és a Google Home / Amazon Alexa között amivel egy egyszerű lámpát fogunk kezelni. A könnyebb átláthatóság érdekében a riportot, 5 részre osztom:

- Mosaic
- Node-Red telepítés és konfiguráció
- Android beállítás
- Különböző szolgáltatások összehasonlítása
- További fejlesztések

A CP2000-es készülékbe beépített PLCComS szerver segítségével könnyedén összehangolhatjuk kedvenc otthoni asszisztensünket a Google Home-t vagy Alexa-t. Ennek köszönhetően akár hangvezérlessel is kapcsolhatjuk a lámpákat, konnektorokat es egyéb alkalmas eszközeinket amit a CP2000-es készülékünk vezérel. A kommunikáció a következőképpen zajlik:

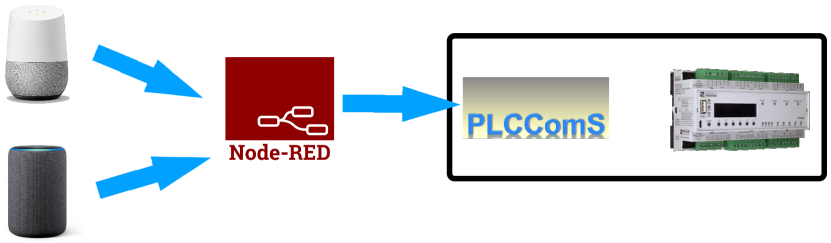


Figure 1: Home asszisztens kommunikációs folyamat

A következő részbe átnézzük a beállításokat és létrehozzuk a fent megemlített tervet.

2 Mosaic

A Mosaic alkalmazásban, hozzunk létre egy új project-et és a beépített könyvtárak segítségével rakjunk össze egy lámpa kapcsolót.

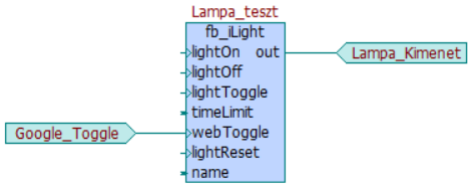


Figure 2: Blokk

Itt a Google_ Toggle lesz a bemeneti változónk amit vagy egy webes felületen vagy az Asszisztensünk fog majd kezelni. A kimeneti változó, "Lampa_ Kimenet" egyértelműen a lámpára utal aminek értéke 0 (Ki) vagy 1 (Be) lehet. Amikor létrehozzuk a bemeneti változót (Google_ Toggle), fontos hogy jelöljük meg publikusra.
Bármilyen változót amit szeretnénk kívülről látni, publikusnak kell lennie. Összefoglalva a Mosaic-os blokkunknak így kell kinézni:

```
FUNCTION_BLOCK Simple_Lamp
  VAR_INPUT
  Google_Toggle {PUBLIC} : BOOL R_EDGE;
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  Lampa_teszt : fb_iLight;
  Lampa_Kimenet  : BOOL;
  END_VAR
  VAR_TEMP
  END_VAR
```

Listing 1: Lampa function blokk

Miután megcsináltuk a "Function Blokkot", behúzzuk a Fő Programba majd az F9 lenyomásával, összerakjuk. A kapcsolónk megvan, most már több időt fordíthatunk a project beállítására.
A project manager-be beállítjuk hogy a publikus értékeket, a mi helyzetünkbe Google_ Toogle exportaljuk a PLC-re. Ez azért szükséges mert a PLCComS szerver beolvassa ezeket a publikus értékeket és így tudjuk majd kezelni őket.

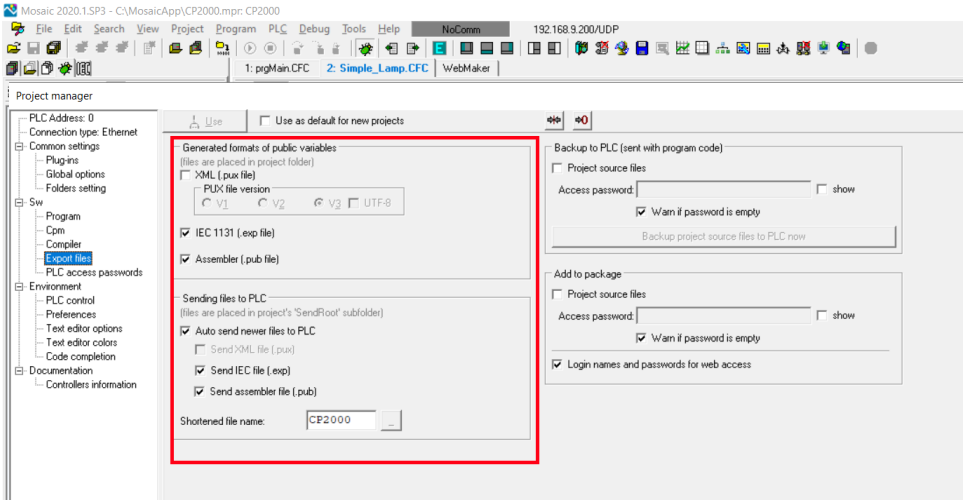


Figure 3: Project Manager konfiguracio

Project manager → SW → Export files és itt bejelöljük a beállításokat ahogy az ábrán latjuk. A "Shortened file name"-hez beírjuk a project nevet és megjegyezzük.

Érdekességképpen .pub file a következően néz ki, amibe látjuk a publikusra jelölt változót:

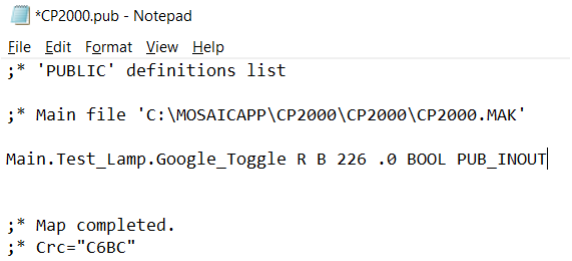


Figure 4: CP2000 pub fajl

Elmentjük a beállításokat, majd nyomjunk egy F9-et és töltsük fel a programot a PLC-re.
Következő teendőnk, a PLCComS konfigurálásra.
Nyissuk meg a kedvenc böngészőnket és lépjünk be a PLC kezelőfelületére amit a PLC_ IP:8443 alatt találnunk. Pl.: 192.168.9.200:8443. A belepési adatok a PLC hátoldalán találhatóak. Alaphelyzetbe a belépési adatok:
Username: Teco
Password: 1234
(Érdemes ezeket a PLC-n megváltoztatni a nagyon biztonság érdekében).
Miután beléptünk a kezelőfelületre, lépjünk a baloldalon található PLCComS menüpontra majd "Servers":

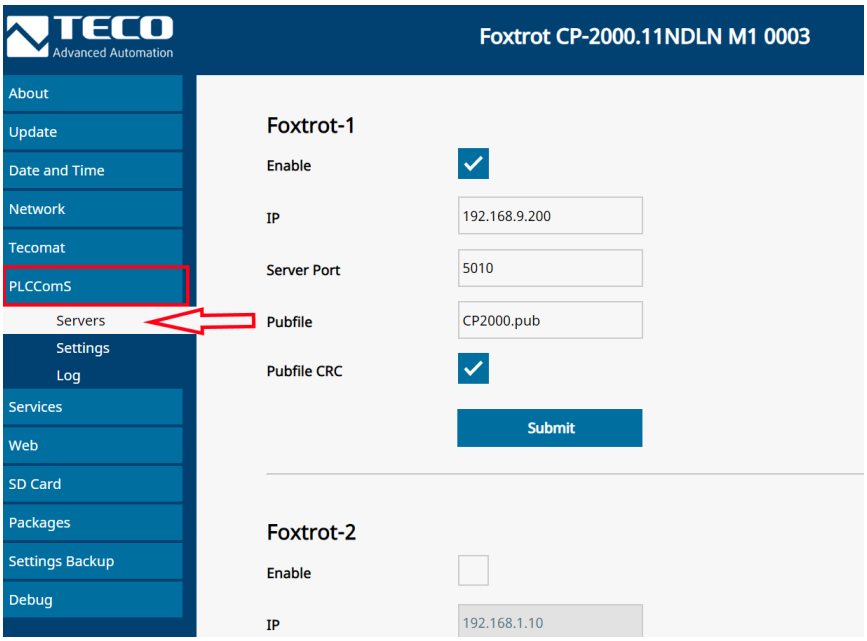


Figure 5: Tecomat kezelőfelület

Itt engedélyezzük a Foxtrot-1 -et és írjuk be a következő beállításokat:
IP: a PLC IP címe
Server Port: 5010
Pubfile: Az előzőleg exportál-t file neve (CP2000.pub)
Public CRC: Pipa
Miután beállítottuk a paramétereket, kattintsunk a “Submit” gombra.
Ezzel sikeresen elindítottunk egy PLCComS szerver-t a publikus értékünkkel.

2.1 Putty ellenorzes

Ha szeretnénk letesztelni hogy a szerverünk valóban üzemel, Putty segítségével ellenőrizhetjük. Putty megnyitás után, beírjuk a következő paramétereket amivel elérjük a PLCComS szerverünket:
IP address: PLC IP cime
Port: 5010
Connection type: Telnet

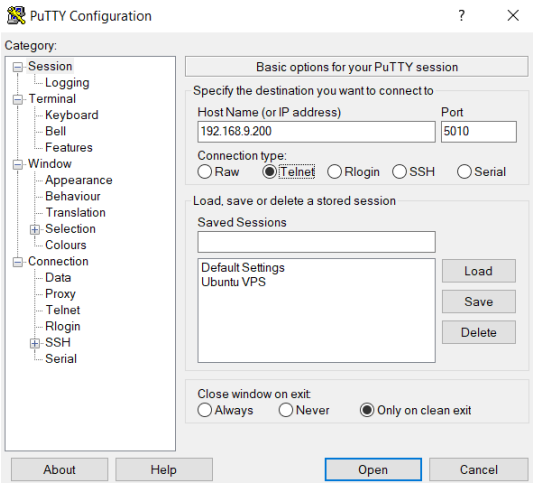


Figure 6: Putty kommunikacio inditasa

Próbaképpen írjuk be a "GETINFO:" parancsot ami a következő választ adja:

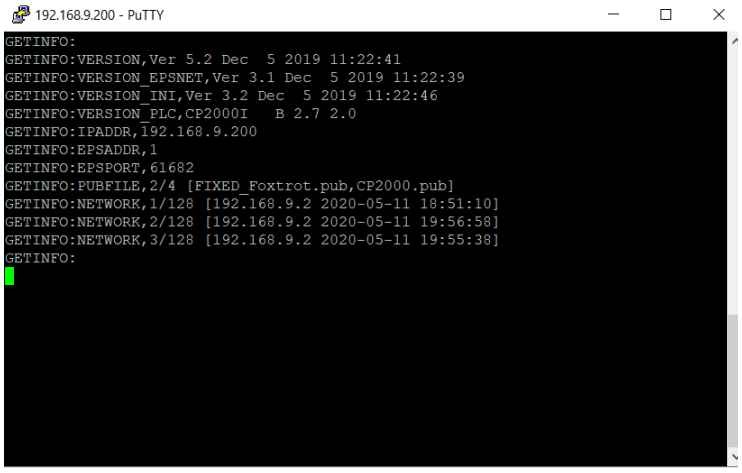


Figure 7: PLCComS teszt parancs

Itt látjuk hogy a "GETINFO:PUBFILE" válasz alatt ott van az előzőleg exportált és megnevezett CP2000.pub fájl.

3 Node-Red

Ebben a részben, feltelepítjük és bekonfiguráljuk azt a szervert ami összeköti a Google Home/Alexa-t a PLC készülékkel. Egy alacsony szintű kódolás ami eseményvezérelt alkalmazásokhoz alkalmas, Node-Red. Egy kis háttér a Node-Red-rol:

A Node-RED egy úgynevezett folyamatáram-alapú programozó eszköz, amelyet eredetileg az IBM Emerging Technology Services csapata fejlesztett ki, és most a JS Alapítvány része.

Az 1970-es években J. Paul Morrison által feltalált folyamatáram-programozás amiben "csomópontokkal" elemezhetjük a programok viselkedését. Minden csomópontnak jól meghatározott feladata van; kapnak bizonyos adatokat, csinál valamit ezekkel az adatokkal, majd továbbítják őket.

A modell nagyon jól alkalmazkodik a vizuális megjelenítéshez, és szélesebb felhasználói kör számára is elérhető. Ha valaki több lépésre bont egy komplex problémát, akkor megnézheti a folyamat áramlását, és könnyebben értelmezheti azt a folyamatot; anélkül, hogy egy egész sor kódot kellene értelmezni egyben.

Kezdésképpen, töltsük le és telepítsük a Nodejs-t az alábbi linkről: `https://nodejs.org/en/` (12.16.3 LTS). Miután feltelepült, nyissuk meg a parancssort adminisztrátorként, majd írjuk be a következőt ami feltelepíti a szükséges bináris komponenseket:

npm install –global –production windows-build-tools

Miután befejeződött a telepítés, nincs más dolgunk mint elindítani a node-red-et. A parancssor-ba simán írjuk be hogy "node-red" majd az enter billentyű lenyomásával indítsuk el:

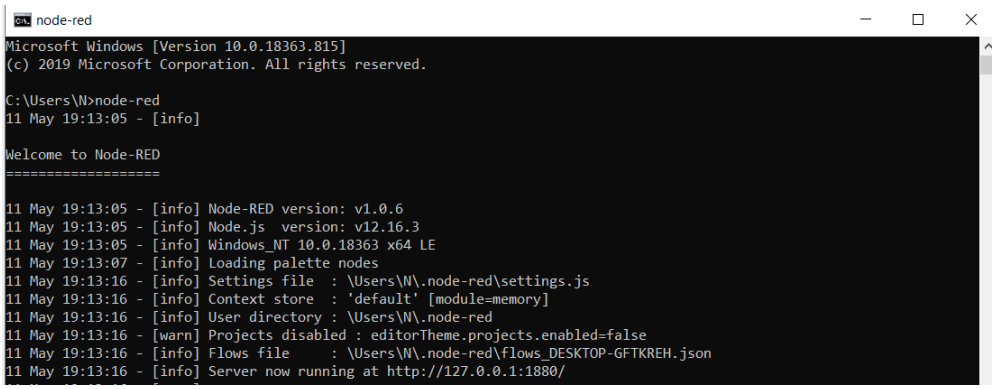


Figure 8: Node-Red telepites is inditasa

Az utolsó sorban látjuk hogy a Node-RED szerver sikeresen elindult és a következő helyi címen fut: 127.0.0.1:1880. Nyissuk meg a böngészőbe a fent említett címet és a következő weboldalra érkezünk:

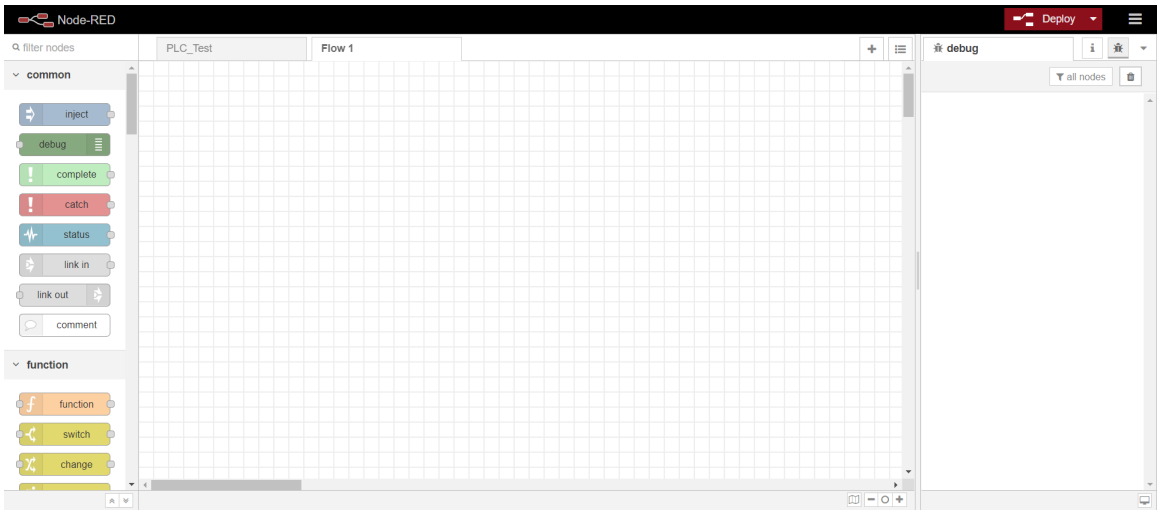


Figure 9: Node-Red programozási felület

Ahhoz hogy a PLC-t tudjuk kezelni, szükség lesz néhány csomópont (Node) telepítésére.
Jobb oldalon, a "Deploy" mellett levő 3 vízszintes csíkra kattintva, nyissuk meg a "Manage palette" menüpontot:

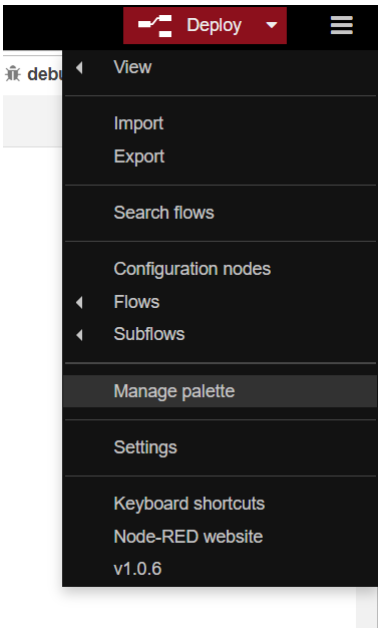


Figure 10: Manage Palette menüpont

Az "Install" menüpont alatt keressünk rá a "foxtrot" csomagra ahogy a képen látható, majd kattintsunk a szöveg doboz jobb oldalán található "Install" gombra.

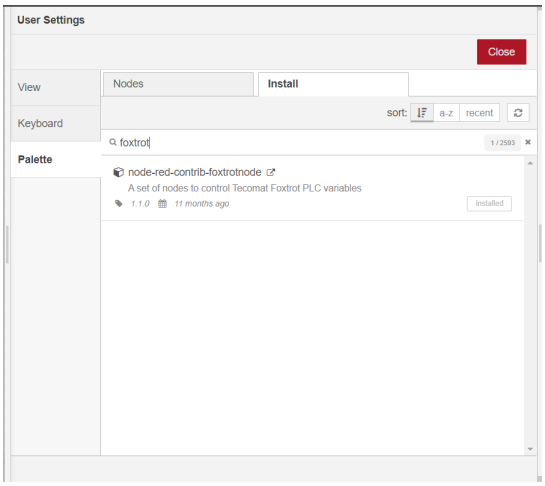


Figure 11: Foxtrot Node telepítés

Ismételjük meg a fenti műveletet csak most keressünk a “node-red-contrib-alexa-smart-home” csomagra:

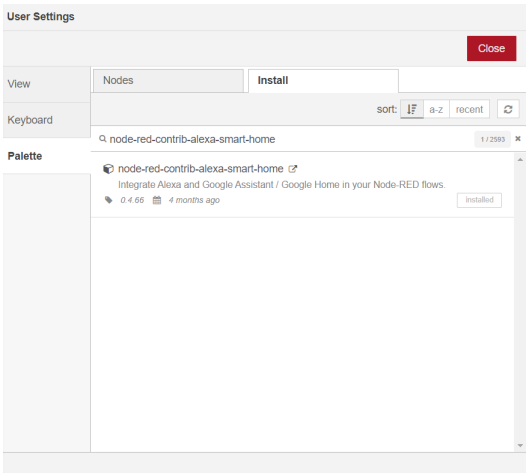
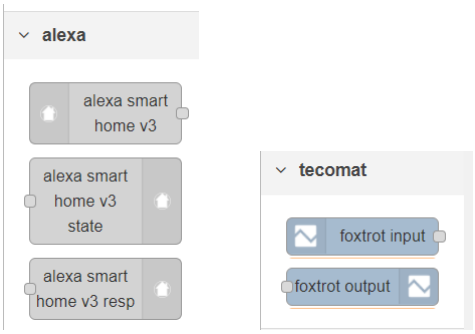


Figure 12: Alexa Node telepítés

Miután sikeresen feltelepítettük a csomagokat, a programozói felületen, baloldalt megjelennek az új ”csomópontok”:



(a) Alexa (b) Tecomat

Figure 13: Frissen telepitett csomópontok

Fogjuk meg a “Foxtrot output” csomópontot és húzzuk a képernyő közepére, majd kattintsunk rá kétszer:

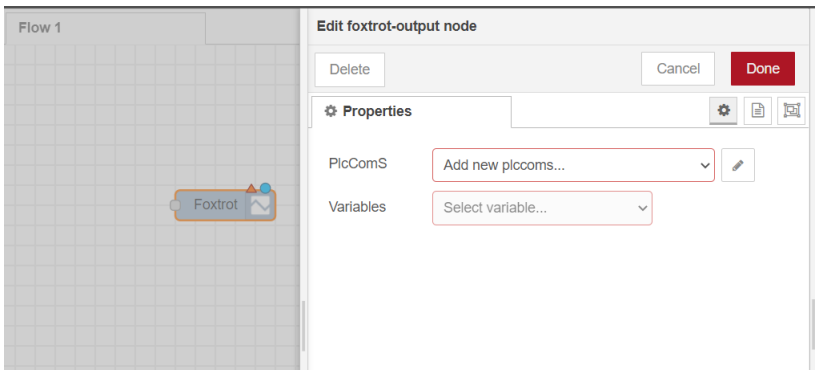


Figure 14: Foxtrot Node telepites

Miután megnyitottuk, kattintsunk a fenti ábrán, pirosan bejelölt gombra, majd adjuk meg a következő adatokat:

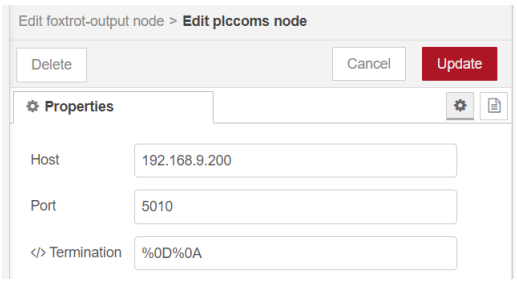


Figure 15: Foxtrot Csomópont konfigurálás

Host: PLC IP Cime
Port: 5010
Termination: %0D%0A$ Majd kattintsunk az “Update”-re. Ha jól állítottuk be

az értékeket és a PLCComS szerverünk rendesen üzemel akkor az előbbiekben “publikusan” megjelölt értéknek, meg kell jelennie a ”Variable” fül alatt:

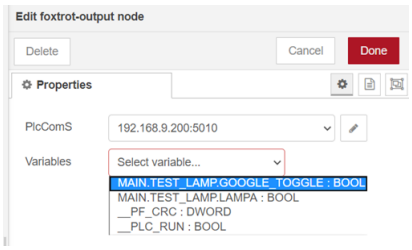


Figure 16: Foxtrot Node telepites

Válasszuk ki a “Google_ Toggle”-t, majd “Done”.
Ezeket a konfigurációkat csak egyszer kell beállítani, többi lámpánál csak az új értéket kell kiválasztani.

A következő lépéshez, harmadik fél szolgáltatást fogunk alkalmazni. Több féle szolgáltatás létezik ami alkalmas összekötni a PLC-t a home asszisztenssel és ennek bővebb összehasonlítását a project végén találunk.

3.1 RED.CB fiók

Hozzunk létre egy fiókot a következő weboldalon:
`https://red.cb-net.co.uk/new-user`
Majd a létrehozás után, jelentkezzünk be:
`https://red.cb-net.co.uk/devices`

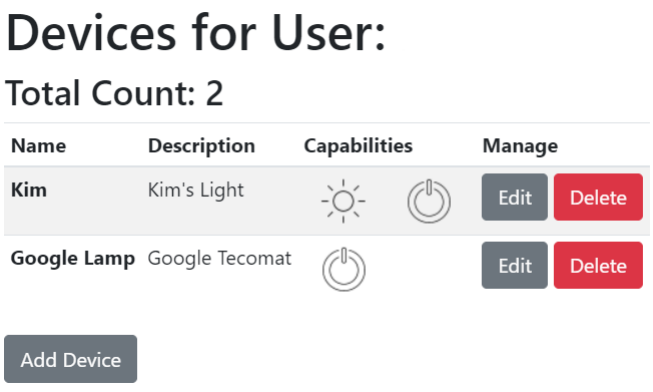


Figure 17: Keszulek hozzaadasa

Bejelentkezés után adjunk hozzá egy készüléket az “Add Device” gombbal. A projekthez használt egyszerű lámpánkat a következőképpen konfiguráljuk és elmentjük:

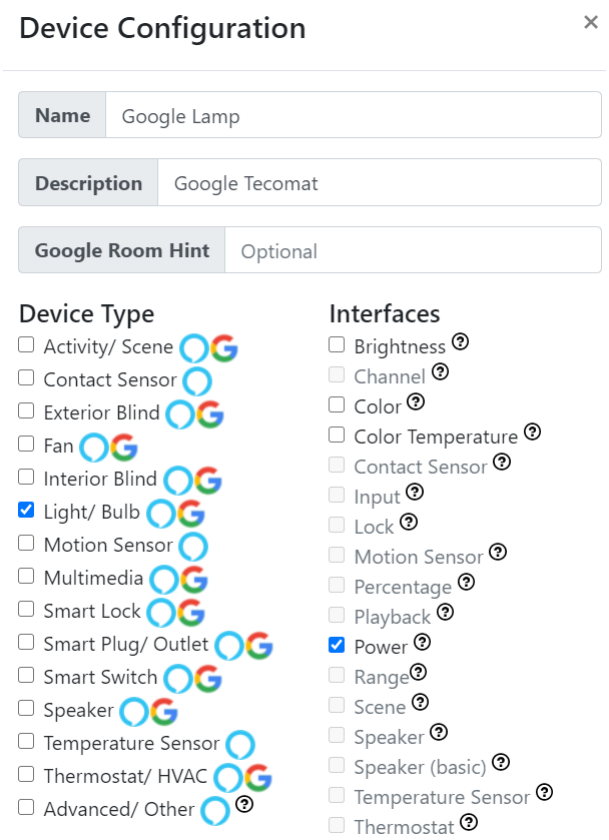


Figure 18: Keszulek hozzaadasa

3.2 Node-Red folytatas

Miután hozzá adtuk a lámpát, lépjünk vissza a Node-Red programozási felületre. Most fogjuk meg az “Alexa Smart Home v3” csomópontot és húzzuk az előzőleg bejegyzett és konfigurált csomópont baloldalára és kattintsunk rá kétszer majd hasonlóképpen ahogy az előbb csináltuk, konfiguráljuk be ahol a Username és Password mezőbe írjuk az előbbi weboldalon regisztrált adatokat.
Sikeres konfiguráció esetén a ”Device” fül alatt megtaláljuk a ”Google Lamp”-et.

Jelenleg a felületnek, így kellene kinéznie:

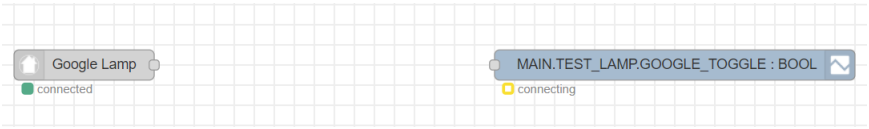


Figure 19: Konfigurált csomópontok

Ezzel meg nem vagyunk kész ugyanis a Google Lamp ”payload” kimeneti üzenete nem egyezik meg a jobb oldalon található bemeneti típussal. Ugye a

Mosaic-ban a "Google_ Toggle" az egy Boolean értek ami lehet Igaz ("1") vagy Hamis ("0"). A "Google Lamp" kimeneti érteke pedig egy String; "ON" vagy "OFF". A Tecomat csomópont String-et tud fogadni viszont a Mosaic-on belül a mi helyzetünkbe egy Boolean.
Az "ON" "OFF"-ot egy olyan értekre kell átírni amit a Mosaic elfogad azaz "TRUE"/"FALSE" vagy "1"/"0".
Hogy áthidaljuk ezt a problémát, egy "Change" csomópontot alkalmazunk amit a "Function" csoport alatt találunk. A "Change" csomópontot középre illesztjük majd kétszeres kattintással, konfiguráljuk:

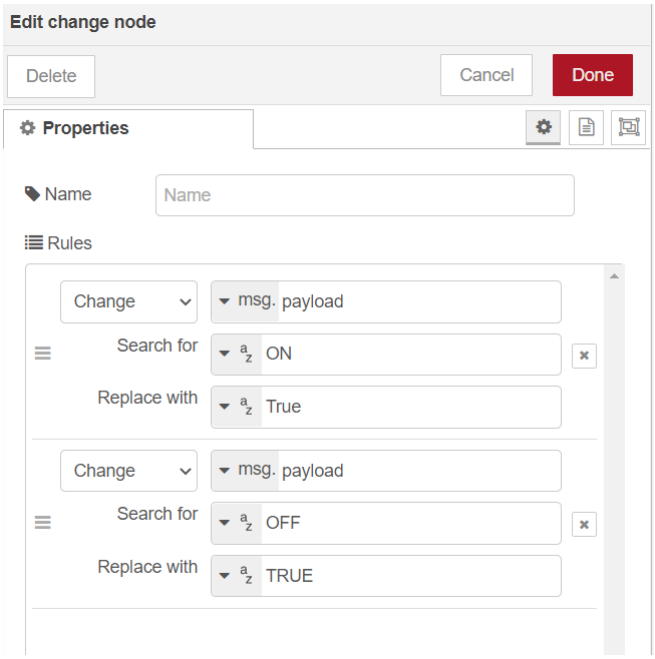
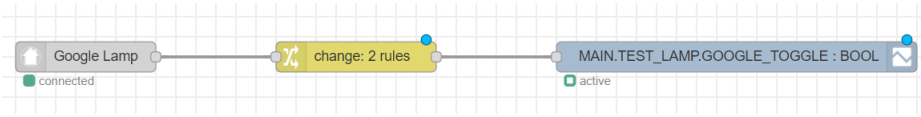


Figure 20: Keszulek hozzaadasa

Itt kicseréljük a String payload-ot a Boolean megfelelőjére: True/False. Miután ezzel megvagyunk, összekötjük a 3 csomópontot és a jobb felső sarokban "Deploy"-al elindítjuk.

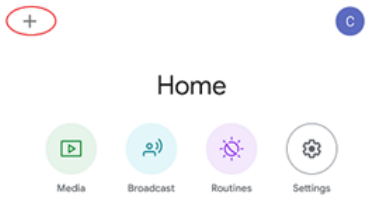


Ha jól konfiguráltuk be a csomópontokat akkor latjuk hogy az "active" vagy a "connected" megjelölést a csomópontok bal also sarkaban. Ezzel sikerült összekötni az Assitant/Home szolgáltatást a PLC-vel, pontosabban a Google Lamp-el.

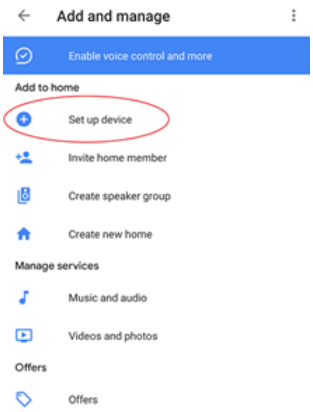
4 Android

Utolsó teendőnk, összepárosítani a Home Assistant a Node-Red szolgáltatással, amit ebben a részben megteszünk.

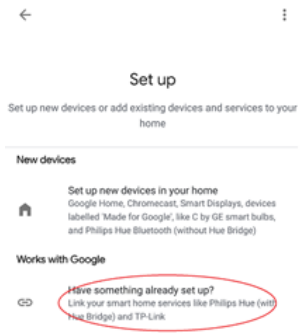
Android-os készüléken nyissuk meg a "Google Home" a alkalmazást. Bal felső sarokba lépjünk a "+" gombra:



Válasszuk ki a "Setup Device"



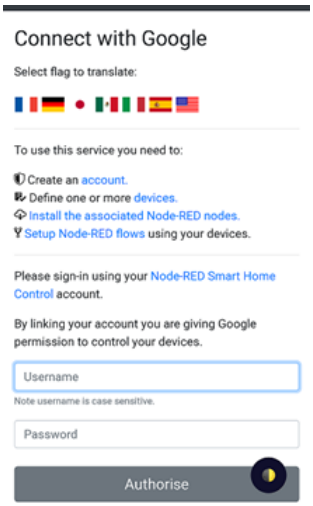
A "Works with Google" alatt lépjünk a "Have something already setup" menüpontra



A keresőbe írjuk be:

Node-RED Smart Home Control

Jelentkezzünk be a felhasználó nevünkkel és jelszóval amit a "Node-Red" részben készítettünk.



Miután beléptünk, megjelenik a "Google Lamp" a felületen:

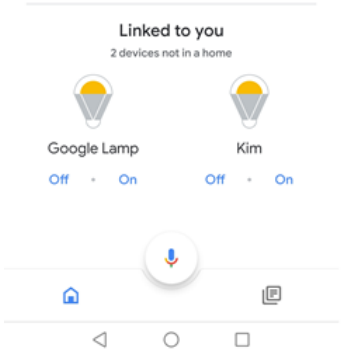


Figure 21: Keszulek hozzaadasa

Tesztelésképpen próbáljuk ki a "Hey Google, turn on the Google Lamp" parancsot és ezzel a lámpa felkapcsol.

5 További szolgáltatások

Ebben a részben átnézzük a projekthez kiválasztott harmadik fél szolgáltatást (Node-RED Smart Home Control) előnyeit és hátrányait majd a további választási lehetőségeket térünk.

5.1 Node-RED Smart Home Control

A projekthez ezt a szolgáltatást használtuk aminek telepítése egyszerűen zajlott. Legjobban, a Google Home és Alexa támogatás különbözteti meg a többi szolgáltatástól. Amikor hozzá adtuk a webes felületen a Google Lamp-et és kiválasztottuk a Típus-t "Light/Bulb", megfigyelhettük a Google es az Alexa logo-t. Ez azt mutatja hogy melyik asszisztenst támogatja a készülék.

Elonyok	Hatranyok
• Konnyu telepites • Google Home es Alexa tamogatas • Egyszeru kezeles • Ingyenes	• Keszulek hozzaadas csak a weboldalon keresztul lehetseges • Harmadik fel szolgaltatas

Table 1: Node-RED Smart Home Control elonyok es hatranyok

5.2 NORA

A NOde-Red home Automation (NORA) egy hasonló szolgáltatás amit a project folyamán használtunk annyi különbséggel hogy a weboldal helyett kapunk egy token azonosítót amit a Node-Red-be bekonfigurálva tudunk használni.
Ennek köszönhetően a Node-Red-en tudjuk hozza adni a lámpákat, konnektorokat stb...

Csomag telepítéshez a Node-Reden:
node-red-contrib-nora

Elonyok	Hatranyok
• Könnyű telepítés • Csak a Node-Red felületen adjuk hozza a készülékeket • Egyszerű kezelés • Ingyenes	• Alap készülékek csak. Konnektor, Lampa stb... • Harmadik fel szolgáltatás

Table 2: NORA előnyök és hátrányok

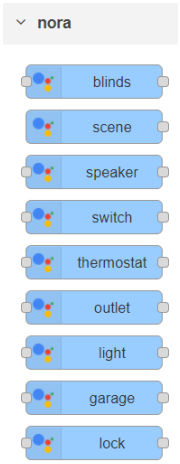


Figure 22: NORA csomópontok

5.3 Google Home Actions v2

Ha valaki a komplikált dolgokat szereti, akkor a "Google Home Actions V2" használata ajánlatos ami "egyszerűen" elküldi az összes vonatkozó szándékot a Google-fiókhoz. Ennek köszönhetően nincs szükség harmadik fél szolgáltatásra viszont cserébe egy elég nehéz csomóponttal van dolgunk. Telepítés nagyon egyszerű viszont a konfiguráláshoz, hitelesített NodeRed szerver szükséges és egy kis idő amire kitaláljuk hogy melyik a helyes "action" az adott feladathoz.

Elonyok	Hatranyok
• Komplikalt hang utasitasokat kothetunk a Tecomat rendszerhez • Ingyenes	• SSL szerver szukseges • Komplikalt Node • Google Home Actions telepites szukseges

Table 3: Google Home Actions V2 előnyök és hátrányok

5.4 SmartHome

Ha valaki úgy gondolja hogy túl sok időt töltött a Google Home Actions v2 konfigurálásával és nem jutott 1-rol a 2-re akkor érdemes megpróbálni a SmartHome Csomópontot. Hasonlóan működik mint a Google Home Actions v2, SSL szerver és Goolge Home Actions szukseges, viszont egyszerűbb leírást találunk hozza és könnyebben tudjuk alkalmazni.

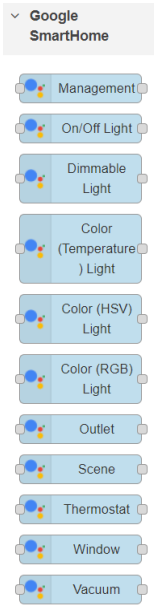


Figure 23: Google V2

Előnyök	Hátrányok
• Egyszerűbb csomópontok • Komplikalt hang utasításokat köthetünk a Tecomat rendszerhez • Ingyenes	• SSL szerver szükséges • Google Home Actions telepítés szükséges

Table 4: Google Home Actions V2 elonyok es hatranyok

6 Tovabbi fejlesztesek

A project-ben egy egyszerű lámpát kezeltünk amit fel vagy lekapcsoltuk hang utasítással Google Home / Assistant-on keresztül.
Tegyük fel, ha a Tecomat rendszerhez egy Dim-es lámpánk van aminek fényereje 0%-100% között van akkor azt a következő paranccsal tudnánk kezelni "Hey Google, change Google Lamp to 65%" vagy "Hey Google, increase/decrease/dim Google Lamp".
A Node-Red felületen amikor adunk egy hang utasítást a Google Home/Alexa felé, látjuk hogy milyen üzenetekre bontja a csomópont. pl.: a "Hey Google, change Google Lamp to 65%" a msg.payload eredménye
msg.payload.brightness : number 65
msg.payload.on : boolean on

Mivel a Tecomat csomópont nem képes 2 üzenetet egyszerre fogadni ezért fontos hogy szétválasszuk az eredeti üzenetet, két különböző üzenetre:

1. Státusz = on
2. Fényerő = 65

A Mosaic programba létrehozunk egy bemeneti változót ami a "Fényerő" lesz és publikusra tesszük hogy a Node-Red-be lássuk a változót. Ezzel már a lámpánk fényerejét is tudjuk kezelni.
Hasonlóképpen tudjuk kezelni a színes lámpákat, konnektorokat és egyéb hasznos, támogatott eszközöket amiket a PLC2000-es tud kezelni.