



Okos otthon hub és irányítóközpont

Készítette

Lovász Ákos

Programtervező informatikus BSc

Témavezető

Dr. Tajti Tibor

Egyetemi adjunktus

EGER, 2021

Tartalomjegyzék

1. A rendszer alapjai	4
1.1. A kiszolgáló hardver	4
1.2. Az Android alkalmazás	5
1.3. Node-RED	5
1.4. MQTT	5
1.5. Okos eszközök	6
2. Hardver	7
2.1. Orange Pi Zero	7
2.2. Okos eszközök	7
2.3. Androidos telefon	7
3. Szoftver	8
3.1. Node-RED	8
3.2. MQTT	8
3.3. Android	8
3.4. Tesztelés	8
4. A rendszer működése	9
4.1. Első indításra felkészítés	9
4.2. Telefon csatlakoztatása kiszolgálóhoz	9
4.3. Okos eszközök csatlakoztatása kiszolgálóhoz	9
4.4. Okos eszközök kezelése az alkalmazásban	9
5. Továbbfejlesztési lehetőségek	10

Bevezetés

Tanulmányaim folyamán számos technológiával ismerkedtem meg, melyek mindegyike rengeteg lehetőséget tárt fel előttem, viszont a szakmai gyakorlatom során kiemelkedően megragadta a fantáziámat az Andoid fejlesztés és a hardverprogramozás összekapcsolása által kialakult rendszerek lehetősége.

Az Android alkalmazások fejlesztése iránt mindig is érdeklődtem, egy-egy kisebb alkalmazást gyakorlásként már készítettem ezt megelőzően, de komolyabban itt kezdtem vele foglalkozni, megismerkedni a vele járó sajátosságokkal.

Az ilyen jellegű eszközök kapcsolata és kommunikációja már korai gondolataimban is az okos otthonok felépítésére emlékeztetett, ezért is gondoltam megfelelő támaszává választásnak.

A döntést követő kutatás során szembetűnő hátránya volt az okos otthon rendszereknek, hogy a legtöbb „márkás” megoldás elsősorban drága és csak felületes hozzáférést tesznek lehetővé, melyet teljes mértékben a rendszer gyártója határoz meg.

Az alternatív, olcsóbb rendszerek bár nyíltabb hozzáállással próbálnak előnyt szerezni, viszont sokszor erősen a technikai oldalába mélyednek, így egy átlagos felhasználónak bonyolultnak, nehezen kezelhetőnek tűnhetnek. Ezen felül gyakran futhatunk olyan problémába, hogy az általunk választott rendszerben lévő hiányosságokat csak más gyártótól származó eszköz nyújthatna megoldást, viszont különböző gyártók eszközei nagyon ritkán kompatibilisek egymással.

Ezeket az észrevételeket figyelembe véve egyértelműnek tűnt, hogy van lehetőség egy olyan rendszer kivitelezésére, ami elsősorban olcsóbb, de ugyanakkor nem túlbonyolított, felhasználóbarát marad. Fontos a nyitottság, a bővíthetőség, és a széleskörű kompatibilitás lehetősége, hogy a felhasználó biztos lehessen abban, hogy a jövőben felmerülő hiányosságok egyszerűen pótolhatók.

1. fejezet

A rendszer alapjai

A rendszer két fő komponensből áll. A kiszolgáló, mely egy Orange PI Zero egykártyás számítógép, amin fut a Node-RED, egy olyan webes felületet biztosító szolgáltatás, mely grafikusan kezelhető komponensek összekapcsolásával teszi lehetővé a rendszer működését befolyásolni, és a Mosquitto MQTT bróker, ami lehetővé teszi a Node-RED[1] és az Androidos alkalmazás közötti kommunikációt.

1.1. A kiszolgáló hardver



(a) Orange Pi Zero[3]

1.1. ábra. A Node-RED-hez és az MQTT brókerhez használt eszköz

Az eszköz egy nyílt forráskódú egykártyás számítógép, amin Armbian[9] (ARM processzor architektúrára specializált Debian) operációs rendszer fut, de lehetőséget nyújt Ubuntu, vagy akár Android operációs rendszer telepítésére is. Ezen fut a Node-RED

felület és a Mosquitto MQTT bróker, melyeket a helyi hálózaton bármely eszköz el tud érni, csupán az eszköz IP címét kell ismernie.

1.2. Az Android alkalmazás

Az Androidos alkalmazás célja a felhasználónak hozzáférést nyújtani az összes elérhető eszközhöz, azok állapotát megjeleníteni és felületet biztosítani azok irányítására, állapotuk megváltoztatására.

A kiszolgálóval való kommunikációt az Eclipse nyílt forráskódú Paho[6] Androidos kliens oldali MQTT implementációját használatba véve valósítja meg az alkalmazás.

A kommunikáció két irányú, azaz nem csak az alkalmazás tudja az okos eszközöket irányítani, hanem fogad üzeneteket a kiszolgálótól, így tud naprakész információt prezentálni az eszközök állapotáról a felhasználó számára.

A felület rugalmasságából adódóan az alkalmazás nem kizárólag okos otthon kezelésére alkalmas, bármilyen MQTT protokoll alapú rendszeren való kommunikációra képes, viszont mivel a fejlesztés során az okos otthonok kezelése volt az elsődleges szempont, így erre a célra használva a leghatékonyabb a felhasználói élmény.

1.3. Node-RED

A Node-RED egy nyílt forráskódú, „flow” alapú programozási eszköz az IBM Emerging Technologies[12] által fejlesztve az OpenJS Foundation[13] részeként. Ez egy Node.js alapú fejlesztési eszköz, aminek a felületét egy böngészőn keresztül lehet elérni ahol „node”-okat elhelyezve a felületen egy funkcióhálózatot létrehozva lehet úgynevezett programozni. Ez a funkcióhálózat egy „Deploy” gomb hatására bekerül a futási környezetbe, így effektíve az eddigi viselkedést felülírva, változtatásainkat elmentve.

1.4. MQTT

Az MQTT (Message Queueing Telemetry Transport)[4] egy OASIS szabványú kommunikációs protokoll, melyet IoT (Internet of Things) eszközök kommunikációjához fejlesztettek ki. A protokoll alapja a „Publish/Subscribe” alapú kommunikáció, azaz egy eszköznek lehetősége van egy adott „topic”-ra üzenetet továbbítani, vagy feliratkozni, azaz az adott „topic”-on beérkező üzeneteket megkapni. Ezek az üzenetek egy brókeren keresztül érik el céljukat, mivel a bróker tárolja hogy mely eszköz milyen témára iratkozott fel, ez alapján tudja a megfelelő klienseknek továbbítani a megfelelő üzenetet. Mivel az MQTT IoT eszközök kommunikációjához készült, így fejlesztése alatt különös figyelmet fordítottak az erőforrások megspórolásához, ezért ez a protokoll nagyon kevés erőforrást vesz igénybe, szinte bármilyen eszköz használatba tudja venni.

1.5. Okos eszközök

kis eszköz amivel lehet okos eszközt szimulálni (vagy integrálásával akár készíteni persze)

2. fejezet

Hardver

orange-py, bármi android telefon, okos eszközök (esp32)

2.1. Orange Pi Zero

olyan mint a raspberry csak olcsóbb

2.2. Okos eszközök

esp32

2.3. Androidos telefon

min andr 6.0 kell

3. fejezet

Szoftver

nodered, mqtt, android

3.1. Node-RED

kifejtés az aktuális rendszerről

3.2. MQTT

kifejtés az aktuális rendszerről

3.3. Android

kifejtés az aktuális rendszerről

3.4. Tesztelés

4. fejezet

A rendszer működése

Itt írom le a kész rendszer működését,

4.1. Első indításra felkészítés

4.2. Telefon csatlakoztatása kiszolgálóhoz

4.3. Okos eszközök csatlakoztatása kiszolgálóhoz

4.4. Okos eszközök kezelése az alkalmazásban

5. fejezet

Továbbfejlesztési lehetőségek

user access? felület új eszköztípusok felvételéhez? cloud service for out of home control

Köszönetnyilvánítás

Köszönöm a vscode-nak hogy van,

Köszönöm magyarországnak hogy jobban teljesít

Köszönöm a covidnak, hogy átmentem nummatból

Irodalomjegyzék

- [1] Node Red forrás,
<https://nodered.org>
- [2] Node Red Dashboard forrás,
<https://flows.nodered.org/node/node-red-dashboard>
- [3] Orange Pi forrás,
<http://www.orangepi.org/>
- [4] MQTT protokoll forrás,
<https://mqtt.org>
- [5] Mosquitto bróker forrás,
<https://mosquitto.org>
- [6] Paho Android MQTT implementáció,
<https://www.eclipse.org/paho/index.php>
- [7] Material design forrás
<https://material.io/components/>
- [8] Android fejlesztői dokumentáció
<https://developer.android.com/guide>
- [9] Armbian operációs rendszer
<https://www.armbian.com/>
- [10] IoT based Smart Environment Using Node-Red and MQTT
Deepthi, B. & Kolluru, Venkata Ratnam & Varghese, George & Narne,
Rajendraparasad & Srimannarayana, Nerella. (2020). IoT based Smart
Environment Using Node-Red and MQTT. Journal of Advanced Research in
Dynamical and Control Systems. 12. 10.5373/JARDCS/V12I5/20201684.
https://www.researchgate.net/publication/342327250_IoT_based_Smart_Environment_Using_Node-Red_and_MQTT

- [11] AndroidTMNotes for Professionals book
<https://books.goalkicker.com/AndroidBook/>
- [12] IMB Emerging Technologies
<https://emerging-technology.co.uk/>
- [13] OpenJS Foundation
<https://openjsf.org/>