

ETDK-dolgozat

Kovács Elek-Ákos

Szász Levente

XXVII. reál- és humántudományi Erdélyi Tudományos Diákköri Konferencia (ETDK)

Informatika II.: innovatív számítástechnikai termékek, alkalmazások szekció

Kolozsvár, 2024. május 15–18.

Sixth Sense Sprinkler

Standardizált okos üvegház rendszer



Szerzők:

Kovács Elek-Ákos

Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar, Informatika szak, III. év

Szász Levente

Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar, Informatika szak, III. év

Témavezetők:

dr. Sulyok Csaba, egyetemi adjunktus

Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar

Gálfi Csaba, szoftverfejlesztő,

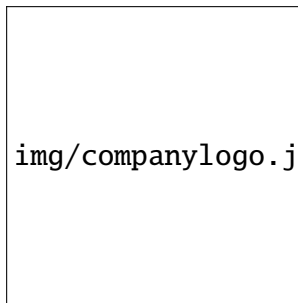
Codespring

Szabó Zsolt, szoftverfejlesztő,

Codespring

Sipos Áron, projektmenedzser,

Codespring



Sixth Sense Sprinkler: Standardizált okos üvegház rendszer

Kovács Elek-Ákos, Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar,
Informatika szak, III. év

Szász Levente, Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar, Informatika
szak, III. év

dr. Sulyok Csaba, egyetemi adjunktus

Babeş–Bolyai Tudományegyetem, Matematika és Informatika Kar

Gálfi Csaba, szoftverfejlesztő,

Codespring

Szabó Zsolt, szoftverfejlesztő,

Codespring

Sipos Áron, projektmenedzser,

Codespring

Az átlagos növénytermesztők számára az üvegházak működtetése rendkívül körülményes feladat. A folyamatos manuális karbantartási munkálatok jelentős időt és energiát vesznek igénybe, továbbá számos probléma merülhet fel egy rosszul beállított öntöző és szellőztető rendszer esetén, amely veszélyeztetheti a teljes termést.

A Sixth Sense Sprinkler projekt célja egy egyszerű és okos üvegház rendszer megalkotása, mely képes különböző érzékelők segítségével a fóliaház állapotáról valós idejű adatokat közvetíteni. Emellett egy mobilalkalmazáson keresztül lehetőséget nyújt öntöző és szellőztető rendszerek távoli irányítására. Minden komponens a Matter szabványra épül, amely a rendszer standard jellegét adja. Ez azt jelenti, hogy ha a felhasználó egy harmadik fél által gyártott hőmérséklet érzékelőt szeretne használni, a rendszer gond nélkül képes azt befogadni, azzal az egy feltétellel, hogy az is a Matter standardot követi.

A hálózat magját a Sprinkler Hubnak nevezett szerver képezi, amelyen keresztül a fóliaházban elhelyezett mikrokontrollereket lehet vezérelni. A mikrokontrollerek különböző szenzorokkal vannak felszerelve, mint például hőmérséklet, talajnedvesség, levegő páratartalom érzékelők, melyek a környezet állapotát figyelik. A Sprinkler Hubhoz való hozzáférést egy Android mobilalkalmazás valósítja meg, amely lehetővé teszi a felhasználók számára az eszközök üzembe helyezését, távoli vezérlését és a múltbeli mérések megtekintését grafikonokon keresztül.

A dolgozat bemutatja a rendszer működését, betekintést nyújt a használt technológiákba és útmutatást ad a funkcionalitások használatához.

Tartalomjegyzék

Bevezető	1
1. Architektúra	3
1.1. Az üvegház összeszerelése	3
1.1.1. Elektronikai komponensek	4
1.1.2. Mikrokontrollerek	6
1.2. Thread protokoll	8
1.2.1. Tulajdonságok	8
1.2.2. Eszközök típusai és szerepei	9
1.2.3. Hálózat	10
1.3. Matter protokoll	11
1.3.1. Hálózat	12
1.3.2. Felépítése	12
1.3.3. Biztonság és adatvédelem	14
1.4. Sprinkler mobilalkalmazás	14
1.4.1. Kommunikáció a Sprinker Hubbal	15
1.5. Sprinkler Hub	16
1.5.1. Felépítése	16
1.5.2. Matter hálózat vezérlése	17
1.5.3. Adattárolás	18
1.5.4. API	19
1.5.5. Kérések validálása	20
2. A mobilalkalmazás felhasználása	21
2.1. Eszközök előkészítése	21
2.1.1. Thread Border Router előkészítése	21
2.1.2. Mikrokontrollerek és érzékelők előkészítése	21
2.2. Bejelentkezés	22
2.3. Főoldal	23
2.4. Eszközök üzembe helyezése	24
2.5. Eszközök kezelése	24
2.6. Grafikonok lekérése	25

3. Technológiák	27
3.1. Git és Gitlab	27
3.2. nRF Connect	27
3.3. Zap Tool	28
3.4. Kotlin, Ktor és Jetpack Compose	28
3.5. Figma	29
Következtetések és továbbfejlesztési lehetőségek	30

Bevezető

Az üvegházakat, illetve üvegház rendszereket üzemeltetők számos kihívással néznek szembe a terményük megóvása érdekében. A legtöbb gond előidézője a nem megfelelően kialakított környezet. Egy kritikus pontja a növénytermesztésnek a kellő hőmérséklet megfigyelése és fenntartása, de ide tartozik a precíz öntözésnek és szellőztetésnek a hiánya is, amely olyan negatív következményeket vonhat maga után, mint a kiszáradás, a gombásodás veszélye, vagy a kártékony rovarok bevonása az üvegházba. Egy olyan rendszer megalkotása, amely minden lényeges aspektusát figyelembe veszi a növénytermesztés ezen szektorának tehát nem egyszerű feladat.

Az Internet of Things (IoT) [19] technológiáinak jelentős része már integrálódott a mezőgazdaságba [20], és számos olyan megoldás áll rendelkezésre, amelyek megkönnyítik az üvegházi műveleteket a digitalizáció eszközeivel, mint például a *Semios* [27] rovarirtó rendszerei és a *Sensaphone Sentinel* [28] különböző érzékelő- és megfigyelő berendezései. Azonban ezek megfelelő kiválasztása további kihívásokat jelenthet a kezdő és a tapasztalt gazdálkodók számára egyaránt. Gondoljunk csak arra a sokféle gyártóra és termékre, melyek közül ki kell választanunk azokat, amelyek a jelenlegi növényállományunk igényeit leginkább kielégítik, nem beszélve a kompatibilitási problémákról [2], amelyek akkor merülnek fel, ha több különböző rendszer egységesítésére kerül sor. Ezek a rendszerek a munkafolyamatok bonyolultságát is növelhetik.

A Sixth Sense Sprinkler projekt célja pontosan egy olyan okos rendszer megalkotása, amely az üvegházak menedzselését teszi hatékonyabbá és kényelmesebbé. Ez a rendszer egy új, az okoseszközökre optimalizált hálózati protokoll [34] segítségével kíván változásokat hozni a mezőgazdaság digitalizációjában. A rendszer lehetővé teszi a felhasználók számára, hogy egy mobilalkalmazáson keresztül nyomon követhessék üvegházaik aktuális állapotát. Ennek érdekében talajnedvesség érzékelőket, valamint egy összetett eszközt alkalmaz, amely a levegő páratartalmát és a belső hőmérsékletet méri. Az adatok gyűjtése egy lokális hálózaton keresztül történik, de a Sprinkler Hub bekapcsolásával ezek az információk távolról is elérhetővé válnak, egy múltbeli adatokat bemutató grafikon segítségével.

A globális piacon már léteznek hasonló megoldások, főként mobilalkalmazások formájában, mint például a *Google Home* [7] vagy az *Apple Home* [9], amelyek az Android és iOS környezetekben az alapértelmezett okosotthon-alkalmazások. Ezeknek a rendszereknek a fő problémái közé tartozik a párosítási folyamat bonyolultsága, mint például a harmadik féltől

származó applikációk vagy fiókok használatának szükségessége és a Google Home-mal való összekapcsolás. Ezenkívül problémát jelent a múltbeli mérések hiánya, hiszen sem az Apple Home, sem pedig a Google Home nem rendelkezik ilyen funkcionalitással.

A *Home Assistant* [10], mint szoftverrendszer, szintén népszerű választás lehet az okosotthon-automatizálásra, beleértve az üvegházakat is, főleg a rendszer konfigurálhatósága és rugalmassága miatt. Azonban vannak bizonyos korlátai és nehézségei, amelyek miatt nem mindenki választaná ezt az opciót. A Home Assistant bár előnyös a testreszabhatóság szempontjából, jelentős komplexitást is hordoz, ami kihívást jelenthet azok számára, akik nem rendelkeznek elegendő technikai ismerettel. Mivel ez egy helyben futtatandó operációs rendszer vagy platform, szükséges a felhasználó részéről bizonyos technikai háttérű ismeretek birtoklása.

A dolgozat négy fejezetből áll. Az 1. fejezet az üvegházrendszer egységeit részletezi. A 2. fejezet bemutatja ennek a funkcióit és alkalmazási lehetőségeit. A 3. fejezet a felhasznált eszközökkel és technológiákkal foglalkozik. Az utolsó, összegző részben kerülnek megemlítésre a fontosabb továbbfejlesztési lehetőségek.

A projekt alapötlete 2023 nyarán került megfogalmazásra. Az első fejlesztési fázisok a Codespring Mentorprogram ¹ részeként valósultak meg, majd az azt követő hat hónapban, a csoportos projekt tantárgy keretein belül további új csapattagokkal kiegészülve zajlott a projekt fejlesztése.

Külön köszönetet szeretnénk mondani Dr. Sulyok Csabának a projekt irányításáért, valamint Sipos Áronnak, Gálfi Csabának és Szabó Zsoltnak a mentorálásért és a szakmai segítségnyújtásért. Hálánkat fejezzük ki diáktársainknak, Veres Búlsúnak és Pápai Csaba-Zsoltnak is a csoportos projekt által nyújtott segítségükért. Végül, de nem utolsósorban, mély köszönet illeti a Codespringet a rendkívüli technikai támogatásért.

¹Codespring Mentorprogram főoldala: <https://edu.codespring.ro>

1. Architektúra

A projekt architektúrája (ld. 1. ábra) különböző alkotóelemekből áll, beleértve az elektronikai komponenseket, mikrokontrollereket, a Sprinkler Hub nevű szervert, a Sixth Sense Sprinkler Android alkalmazást és a Google Nest Hub 2nd Gen-t [21]. Ebben a fejezetben részletesen bemutatjuk ezeknek a komponenseknek a szerepét és azok közötti kommunikációt.

A rendszer két hálózatot, úgynevezett Matter fabricet (bővebben az 1.3.1. fejezetben) biztosít a mikrokontrollerek számára. A felhasználók a mobil alkalmazás segítségével helyezhetik üzembe az érzékelőkkel felszerelt mikrokontrollereket. Ezeket a kéréseket a Google Nest Hub fogadja Wi-Fi-n keresztül, így létrejön a lokális Android fabric. A felhasználó megoszthatja eszközeit a szerverrel, csatlakoztatva őket a Sprinkler Hub fabrichez, ha szeretné, hogy az eszközei visszamenőlegesen elmentsék méréseiket, vagy részt vegyenek az automatizálásban.² A mobil alkalmazás és a szerver egy REST API-n [33] keresztül kommunikálnak.

A mikrokontrollerek a Nest Hubbal együtt Thread [12] hálózatot alakítanak ki. A kialakított hálózat internet hozzáférés nélkül is képes az érzékelők méréseit továbbítani a Nest Hub felé. A Nest Hub Wi-Fi-n keresztül továbbítja a kapott adatokat a mobil alkalmazásnak valamint a Sprinkler Hubnak.

1.1. Az üvegház összeszerelése

A projekt keretében kifejlesztett informatikai rendszer valóság-hű környezetbe való integrálása érdekében már a tervezési szakaszban felmerült egy miniaturizált üvegház-modell koncepciója. Az alapot egy SENAPSKÅL³ elnevezésű, kis üvegházra emlékeztető dekoráció adja, amely árának, szállíthatóságának és méretének köszönhetően kiváló választásnak bizonyult.

A keretben központi helyet kap egy műanyagból készített cserép, amelyben nem csak a növény, hanem két talajnedvesség érzékelő is helyet kap. A keret aljában található, 0.5 liter űrtartalmú víztartály az öntözőrendszert látja el vízzel. A víztartály alján elhelyezett DC 5 voltos motorral működtetett szivattyú a víztartályból a növények felé szállítja a vizet szilikon csöveken keresztül. Az öntözési funkciók mellett kiemelt figyelmet kap a szellőztető berendezés

²Ez a funkcionalitás jelenleg még nem elérhető.

³Megvásárolható dekoráció. Forrás: <https://www.ikea.com/us/en/p/senapskal-decoration-greenhouse-indoor-outdoor-white-20487774/>

img/architektura.png

1. ábra. A projekt architektúra diagramja.

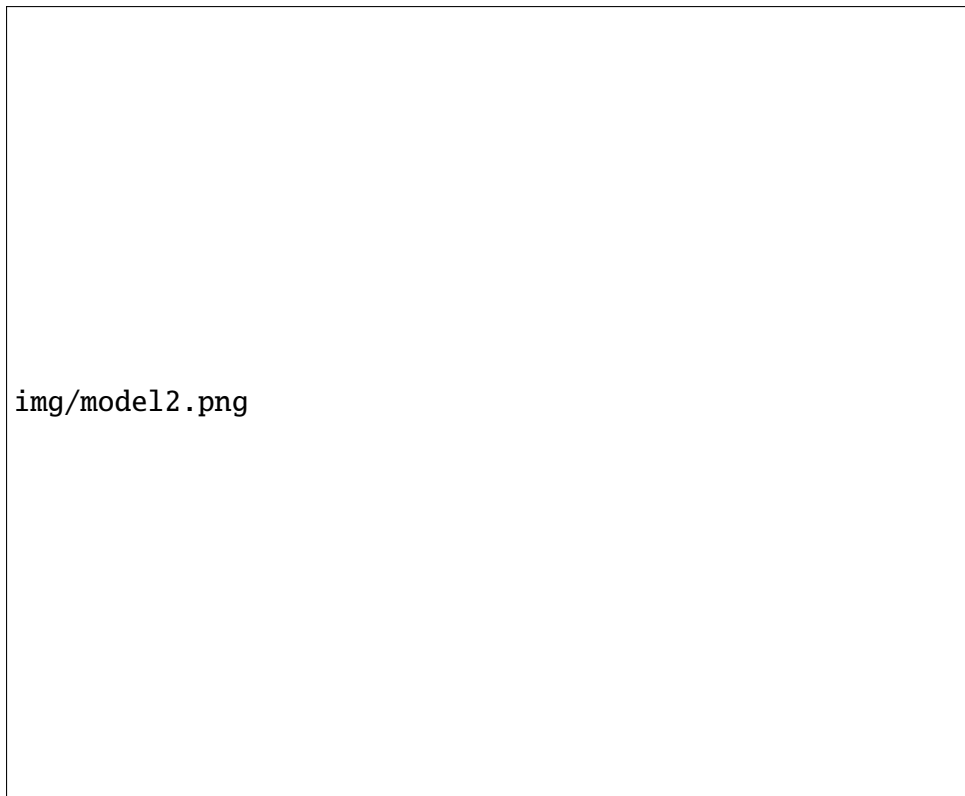
szimulálása is. A hőmérséklet és a levegő páratartalmának monitorizálására DHT22 típusú érzékelőt alkalmaznak, míg a szellőztetést egy az érzékelő közelében elhelyezett, szintén DC 5 voltos ventilátor imitálja, amelynek gyors légáramlata hozzájárul a hőmérséklet és páratartalom csökkentéséhez.

A ventilátor és a szivattyú 5 voltos egyenáramú betáplálást igényelnek, így nem csatlakoztathatók közvetlenül a hálózati feszültségre. Ennek a problémának a kiküszöbölésére AC-DC step down hálózati adaptereket használunk, hogy a 230 voltos váltóáramú feszültséget 5 voltos egyenáramú feszültségre alakítsuk át. Ezek az adapterek egyenként egy okos hálózati csatlakozókba kapcsolódnak, így a levegőztető és szivattyú vezérlése, könnyedén megvalósítható azok ki/be kapcsolásával.

1.1.1. Elektronikai komponensek

A felhasznált elektronikai komponensek két csoportba sorolhatóak:

- *Érzékelők* melyek a környezeti adatok, mint például a hőmérséklet mérésére használhatóak. A begyűjtött adatot digitális vagy analóg jel formájában továbbítják a mikrokontrollernek.



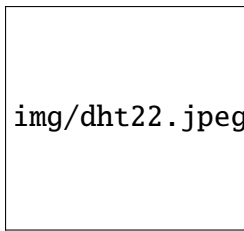
2. ábra. Az üvegház modellje. A komponensek összekötése és elhelyezése.

- *Aktuátorok* vagyis olyan elektromos eszközök, melyek valamilyen műveletet hajtanak végre a bemeneti jel alapján.

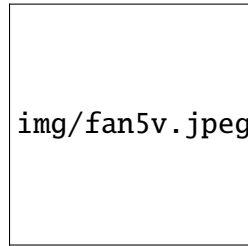
Talajnedvesség mérésére *Capacitive Soil Moisture Sensor v2.0* [24] típusú eszköz van felhasználva. A talaj elektrolit tartalmának változásai, befolyásolják annak kapacitását. A feszültség kimenet 1,2V-tól 3V-ig terjed, és az analóg tűskén (pinen) keresztül kapható meg. Továbbá a korrózióálló anyagokból készült konstrukció lehetővé teszi a hosszú távú és megbízható használatot.

A *DHT22* [1] érzékelő (ld. 3. ábra), amely *AM2302* néven is ismert, egyidejűleg méri a hőmérsékletet -40 és $+80$ °C között, valamint a levegő páratartalmát 0 és 100 százalék közötti tartományban. Ez az érzékelő az előzőtől eltérően nem analóg, hanem digitális jelet használ a kommunikációra, ami a használatát jelentősen egyszerűsíti. Képes 2 másodpercenként egy mérési adat olvasására, ami 0,5 Hz-es frekvenciának felel meg. Az átvitt adatcsomag összetétele öt bájtot foglal magában: 8 bit relatív páratartalom egész érték, 8 bit relatív páratartalom tizedes rész, 8 bit hőmérséklet egész rész, 8 bit hőmérséklet tizedes rész, valamint egy 8 bit checksum.

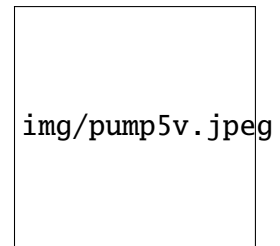
Fontos kihangsúlyozni, hogy az üvegház modellhez (ld. 2. ábra) igazodva csupán kis teljesítményű aktuátorok kerülnek alkalmazásra, amelyek inkább egy környezet működését



3. ábra. DHT22 érzékelő



4. ábra. DC 5 voltos ventilátor



5. ábra. DC 5 voltos elektromos szivattyú

	EFR32xG24 Explorer Kit	ESP32-C6-DevKitM-1	nRF52840 Dongle
Órajelfrekvencia	78 MHz	160 MHz	64 MHz
Flash memória	1.5 MB	4 MB	1 MB
Wifi	Támogatott	Támogatott	Támogatott
Bluetooth	Támogatott	Támogatott	Támogatott
Thread	Támogatott	Támogatott	Támogatott
Energiafogyasztás	Alacsony	Közepes	Nagyon alacsony
Közösségi támogatás	Jó	Jó	Kiváló
Ár	36\$	8\$	14\$

1. táblázat. Mikrokontrollerek összehasonlítása

szimulálják, mintsem egy teljes körű okos üvegház funkcióit biztosítanák.

A szellőztetés szimulációjához egy 5 voltos ventilátor (ld. 4. ábra [32]) kerül felhasználásra, amely közvetlenül a korábban említett DHT22 érzékelő mellé kerül. Ennek köszönhetően a ventilátor aktiválása esetén a mért hő megközelítőleg 2-3 fokkal csökken.

Az üvegházak elengedhetetlen részét képezi a csepegtetőrendszer. Jelen esetben a csepegtetés egy 5 milliméter átmérőjű szilikon csövön keresztül valósul meg, amely a vízforrástól a növényig szállítja a vizet. A rendszert egy elektromos 5 voltos merülő szivattyú (ld. 5. ábra [25]) hajtja meg.

Ezt a két aktuátort Meross Smart Plug Mini [18] okos konnektorok látják el energiával, ami kiemelten fontos, mivel ezek az eszközök már támogatják a WiFi-n keresztüli Matter protokollt. Az okos hálózati csatlakozó használata ráadásul költséghatékonyabb megoldást jelent, mint az okos szivattyúk és ventilátorok beszerzése.

1.1.2. Mikrokontrollerek

Mikrokontrollerek kiválasztása

A piacon számos mikrokontroller elérhető, azonban olyanra van szükségünk, amelynek SoC-ja (System-on-chip) támogatja a Thread és a Matter protokollokat. A legnépszerűbb

választások közé tartoznak a Silicon Labs EFR32⁴ sorozata, az Espressif ESP32-C3⁵ sorozata és a Nordic Semiconductor nRF52⁶ sorozata. Ezek a márkák részei a Connectivity Standard Alliance-nek [5], amely Matter eszközöket épít és támogat.

A mikrokontrollerek kiválasztásakor számos szempontot figyelembe vettünk, köztük a felszereltséget, árat és méretet (ld. 1. táblázat). Három említésre méltó mikrokontroller az EFR32xG24 Explorer Kit⁷, az ESP32-C6-DevKitM-1⁸ és az nRF52840 Dongle⁹.

Az EFR32xG24 Explorer Kit ígéretesnek tűnik, de költséges, főleg mivel több mikrokontrollerre van szükségünk. Az ESP32-C6-DevKitM-1 jó választás lenne, de az nRF52840 dokumentációja sokkal részletesebbnek és kidolgozottabbnak bizonyult. Ráadásul a Nordic Semiconductor [22] sok hasznos webinárt és felhasználóbarát eszközt biztosít, amelyek segítenek nekünk saját projektünk fejlesztésében. Emellett a Dongle kicsi és a lapkán található USB csatlakozó miatt nem igényel micro-USB kábelt a PC-hez való csatlakozáshoz. Összefoglalva, úgy döntöttünk, hogy a projektünkhöz az nRF52840 Dongle-okat használjuk, mert ezek felelnek meg a legjobban az igényeinknek.

Mikrokontrollerek működése

A mikrokontrollerek feladata a rájuk kötött talajnedvesség, hőmérséklet, és levegő-páratartalom érzékelők méréseinek leolvasása, és továbbítása Thread protokollon keresztül. A mikrokontrollerek fel vannak szerelve egy factory reset gombbal, egy elem tartóval és Matter QR kóddal amely az üzembe helyezéskor szükséges. Mivel egyes mikrokontrollereken talajnedvesség érzékelők vannak, amelyek analóg jelet küldenek, ezért környezettől függően más-más értékeket kaphatunk. Annak érdekében, hogy a mérések minél pontosabbak legyenek, ezek a mikrokontrollerek el vannak látva egy piros gombbal is, amely segít felhasználóinak bekalibrálni az érzékelőt, azzal, hogy lekéri a levegőben, valamint a tiszta vízben mért analóg jelet. Egy ilyen eszköz látható a 6. ábrán.

A mikrokontrollerek firmware-je C++ nyelven íródott, a Nordic Semiconductor által biztosított nRF Connect SDK segítségével. Erről részletesebb leírás a 3.2 fejezetben található.

⁴Forrás: <https://www.silabs.com/wireless/technology>

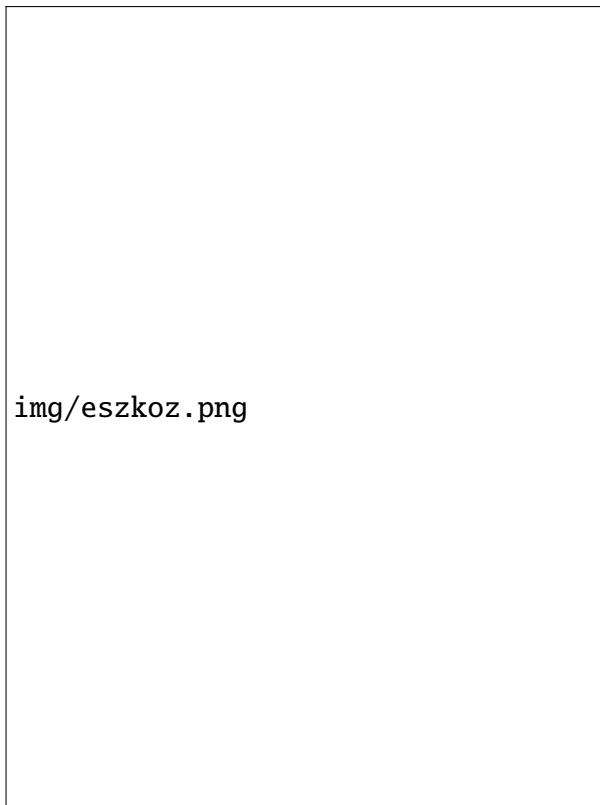
⁵Forrás: <https://www.espressif.com/en/products/socs/esp32-c3>

⁶Forrás: <https://docs.nordicsemi.com/category/nrf-52-series>

⁷Forrás: <https://www.silabs.com/development-tools/wireless/efr32xg24-explorer-kit?tab=overview>

⁸Forrás: https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitm-1/user_guide.html

⁹Forrás: <https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle>



6. ábra. Nordic nRF52840 Dongle mikrokontroller talajnedvesség érzékelővel.

1.2. Thread protokoll

A Thread egy vezeték nélküli hálózati protokoll, melyet a Thread Group szervezet 2015 nyarán tette elérhetővé. Ez a protokoll kifejezetten az IoT eszközök közötti kommunikációra lett tervezve, elsősorban az okos otthonok és az ipari automatizálás területén való alkalmazásra. A protokollt folyamatosan fejlesztik és bővítik, hogy megfeleljen az iparág növekvő igényeinek és kihívásainak.

Az OpenThread [13] egy Google által fejlesztett nyílt forráskódú implementációja a protokollnak, így fejlesztők széles körben használhatják, és hozzáadhatják saját projekteikhez vagy termékeikhez. A Sixth Sense Sprinkler eszközei is az OpenThread implementációt használják a kommunikáció kialakítására.

1.2.1. Tulajdonságok

A Thread protokoll energiatakarékos, ugyanis az IEEE 802.15.4¹⁰ szabványra épül, amely alacsony energiaigényű kommunikációt tesz lehetővé az eszközök között. Ez magába foglalja

¹⁰Szabvány az alacsony sebességű, vezeték nélküli hálózatokhoz. Forrás: <https://standards.ieee.org/ieee/802.15.4/7029/>

az alacsony adatsebességet és az adatsomagok hatékony kezelését, ami minimalizálja az energiaveszteséget a hálózatban. A maximális adatátviteli sebesség 250 kbps. A szabvány alapján a protokoll 2.4 GHz-es frekvenciasávon működik, amely széles körben használt vezeték nélküli kommunikációban.

A protokoll IPv6 alapú, ami számos előnnyel jár. Nagyobb címterületet biztosít, lehetővé téve minden eszköz egyedi azonosítójának megadását a hálózatban belül. Rugalmasabb hálózat- és eszközkezelést tesz lehetővé, valamint nagyobb skálázhatóságot biztosít a jövőbeni igényekhez igazítva. Emellett kompatibilis az IPv4-el, így lehetővé teszi a fokozatos átmenetet az IPv4-ről az IPv6-ra, és lehetőséget biztosít a már meglévő hálózati infrastruktúrák integrálására.

A protokoll számos beépített biztonsági funkcióval rendelkezik, amelyek lehetővé teszik az adatok védelmét és a hálózat biztonságát az IoT környezetekben. Az AES [8] titkosítás alkalmazása révén az adatokat szilárdan védelmezi a hálózaton való átvitel során, megakadályozva az illetéktelen hozzáférést és lehallgatást. Az autentikációs mechanizmusok segítenek megbizonyosodni arról, hogy csak az engedélyezett eszközöknek van hozzáférése a hálózathoz, és megakadályozzák a hamisított eszközök becsatlakozását.

1.2.2. Eszközök típusai és szerepei

A Thread specifikációban¹¹ két típusú eszköz létezik: a Teljes Thread Eszköz (FTD) és a Minimális Thread Eszköz (MTD). Az MTD a legkisebb követelményekkel rendelkezik az eszköz hardverét (pl. memória méretét) és az energiafogyasztást illetően, míg az FTD a legváltozatosabb a szerepek tekintetében, amelyeket egy Thread Hálózaton betölthet. [30]

A specifikációban számos szerepet definiáltak, a továbbiakban csak azokat említjük meg, melyeket a projekt eszközei betöltenek:

Router és Leader

A Thread Routerek FTD-k, amelyek routing szolgáltatásokat nyújtanak a hálózat eszközei számára. Emellett csatlakozási, és biztonsági szolgáltatásokat is biztosítanak azoknak az eszközöknek, amelyek csatlakozni próbálnak a hálózathoz. A csatlakozni kívánt eszköz jelerősség alapján kiválasztja melyik routerrel szeretne kommunikálni. Ha a kiválasztott router elfogadja a csatlakozási kérelmet ő lesz az eszköz szülő routere.

¹¹Thread specifikáció. Forrás: <https://www.threadgroup.org/ThreadSpec>

A Leader egy Router amelynek egyéb felelőssége is van. Döntéseket hozhat a hálózat állapotáról, például, hogy az eszközök között ki válhat Routerré. Leaderből csak egy lehet egy Thread hálózatban, ha meghibásodik egy másik Router veszi át a szerepét.

Minimal- és Sleepy End Device

A Minimal End Device (MED) csak a szülő routerén keresztül tud kommunikálni a hálózattal, és nem tud más eszközök számára üzeneteket továbbítani. Rádiója mindig be van kapcsolva, akkor is, amikor tétlen.

Ehhez képest a Sleepy End Device (SED) időnként elalszik, majd felkel, hogy beszélgesse a szülő routerével, ezzel energiát spórolva meg.

Border Router

A Border Router szerepe, hogy kapcsolatot teremtsen más IP alapú, akár Wi-Fi, vagy Ethernet hálózatokkal. Ez lehetővé teszi az adatok átvitelét a helyi Thread hálózatról más hálózatokba és fordítva. Egy Thread hálózat több Border Routert is támogathat.

A projektben a Google Nest Hub 2nd Gen egyszerre Leader és Border Router szerepet is betölt. Egy gyors setup után létrehozza Thread hálózatát és kinevezi magát Leaderré. Látható, hogy erre a termékre azért volt szükségünk, hogy könnyedén létrejöhessen a Thread hálózatunk, amely kommunikálhat a külvilággal, így ezeket a funkciókat mi nem kellett implementálnunk. A mikrokontrollerjeink Sleepy End Device szerepet töltenek be, melyeknek szülő routerjük a Nest Hub.

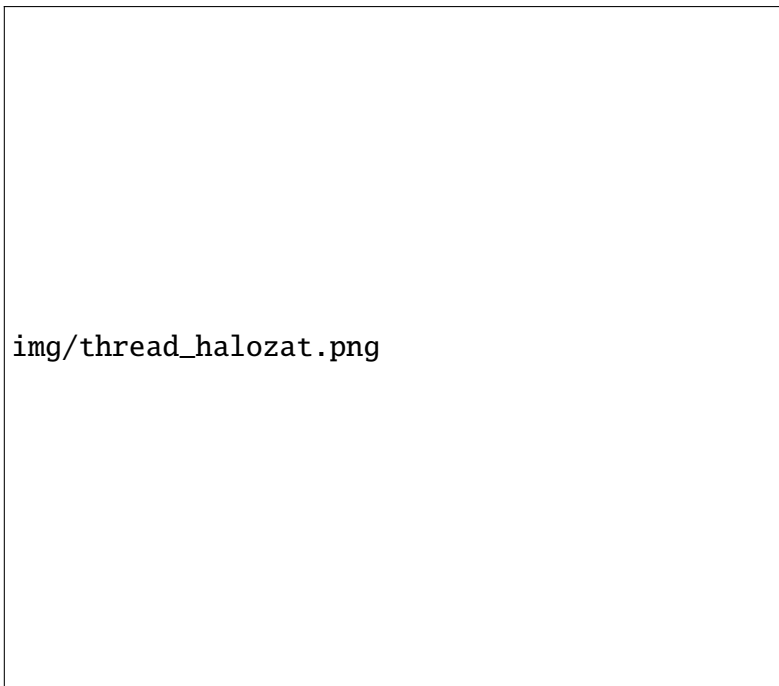
1.2.3. Hálózat

Egy Thread hálózatnak 3 különböző azonosítója van. Egy két bájtos PAN¹² ID, egy nyolc bájtos XPAN ID, és egy ember által olvasható név. Ezek segítenek a hálózatok keresésében és azonosításában.

A Thread specifikáció megengedi a hálózat routerjeinek *full mesh*¹³ topológiáját. Természetesen, ha a hálózat csak egy routert tartalmaz, mint ahogy a projektünk esetében is, akkor csillag topológia alakul ki, ahol minden eszköz az egyetlen Routerrel kommunikál. A 7. ábrán megtekinthető egy lehetséges Thread hálózat Mesh topológiával.

¹²Private Area Network

¹³Minden eszköz közvetlenül kapcsolódik minden másik eszközhöz



7. ábra. Egy lehetséges mesh Thread hálózat.

Egy új eszköz csatlakozása során a Thread hálózathoz, először megkeresi a hálózatban lévő routereket. Az eszköz válaszul kap egy hálózati kulcsot, amely biztosítja az adatok titkosítását és az eszköz azonosságának megerősítését. Miután sikeresen megszerezte a hálózati kulcsot, az eszköz csatlakozik a hálózathoz. Ezt követően az eszköz megkapja a hálózat konfigurációját és beállításait, így lehetőséget kap az adatok küldésére és fogadására a hálózaton keresztül.

1.3. Matter protokoll

A Matter [3], eredetileg *Project CHIP* (Connected Home over IP) néven ismert, egy alkalmazási réteg protokoll, melynek célja az okos otthoni ökoszisztémák szabványosítása. A *Connectivity Standards Alliance* 2019 decemberében kezdte meg a fejlesztést, és az első verzió, a Matter 1.0 specifikációja 2022 októberében került nyilvánosságra. A Matter legfontosabb előnye, hogy lehetővé teszi a különböző gyártók által készített okos eszközök közötti kommunikációs akadályok megszüntetését. Ezáltal minden olyan eszköz, amely implementálja ezt a szabványt, képes kommunikálni más hasonló eszközökkel, függetlenül a gyártótól, a típustól vagy a használati környezettől. Ez nem csak a fejlesztők számára nyújt előnyöket, hanem a felhasználóknak is, mivel egyszerűsíti az eszközök integrálását és kezelését, mindezt akár egyetlen egységes felületen keresztül.

1.3.1. Hálózat

A Matter hálózat a Matter eszközök és az IPv6 hálózatok közötti kapcsolatot határozza meg, egy közös bizalmi teret létrehozva. A különféle IPv6 hálózatok központi csomópontok segítségével képesek kommunikálni más hálózatokkal, például a Thread hálózattal. E csomópontok, más néven hubok, Thread esetében Thread Border Routerként ismertek, amelyekről részletesen az 1.2.2 fejezet tárgyal.

A hálózat alapvető eleme a *Node (Csomópont)*, amely egy Matter eszköz egyedi, fizikai példányát jelenti. A projektben is megjelenik egy ilyen, például a talajnedvesség érzékelő formájában. A Node fő attribútumai közé tartozik a NodeID, ProductID és VendorID.

A *Controller (Vezérlő)* olyan eszköz, amely képes más eszközöket az IPv6 vagy BLE [4] segítségével a rendszerbe integrálni, továbbá rendelkezik azok irányítására való jogosultsággal. Például egy Matter okos lámpát egy ilyen Controller segítségével kapcsolhatunk be vagy ki. A projekt keretében két ilyen eszköz kerül felhasználásra: a mobil alkalmazás, amely rendelkezik a BLE-n keresztüli beiktatási képességgel, valamint a Sprinkler Hub, amely Matter eszközöket figyel statisztikák létrehozásának érdekében.

Egy Controller egyszerre csak egy hálózatot, csoportot hozhat létre, amelyet *fabric*nek neveznek. Azonban ez nem akadályozza meg a Node-okat abban, hogy egyszerre több fabric részesei legyenek. Azt a jelenséget, amikor egy Matter eszköz, egyszerre több fabric része *multi-admin*nek hívják.

A Node-ok kontextusában további szerepkörök is léteznek, amelyek jelenleg nincsenek a projekt által felhasználva, de fontos továbbfejlesztési lehetőségeket kínálnak. Ilyenek például az *OTA¹⁴ Provider* és *OTA Requestor* Node-ok, amelyek képesek kérni és fogadni, illetve szolgáltatni a firmware-t frissítő csomagokat.

1.3.2. Felépítése

A protokoll standard jellege elsősorban az eszközök adatmodelljének felépítésében nyilvánul meg. Az adatmodell hierarchikus szerkezetű (ld. 8. ábra), ahol a legfelsőbb szinten maga a kész eszköz helyezkedik el. Egy eszközt egyetlen Node határoz meg, azonban összetett eszközök esetében több Node használata is megengedett, viszont minden Node saját egyedi címmel rendelkezik egy adott fabricen belül.

Egy Node több úgynevezett *Endpoint*ből (Végpont) áll, amelyek különféle

¹⁴Over-the-air. Firmware verzió frissítése vezeték nélkül



img/weatherstationdm.png

8. ábra. Érzékelővel felszerelt eszköz adatmodellje.

funkcionalitásokat izolálnak el egymástól. Például az első végpont felelhet a hőmérsékletmérésért, míg a második végpont a levegő páratartalmának méréséért.

Az endpointokon belül az eszközöknek rendelkezniük kell *Clusterekkel* (Klaszter), amelyek a konkrét funkcionalitást határozzák meg. Ilyen például a hőmérséklet mérés klaszter, amely a hőmérsékletmérési műveleteket foglalhatja magában. Egy Cluster *attribútumokat* tartalmaz, mint például a *MeasuredValue* vagy a *MinMeasuredValue*, amelyek leírják az eszköz által mért jelenlegi értékeket és az érzékelő által mért minimális értéket. Emellett egy Cluster tartalmazhat *Commandokat*, azaz parancsokat, amelyek képesek olvasni és módosítani bizonyos attribútumokat. Az adatmodell utolsó eleme az *Event*, vagyis Események, amelyek tranzakciókként értelmezhetőek, és az állapotváltozásokat követik nyomon.

A Matter specifikáció megköveteli bizonyos Clusterek használatát egy adott eszköz típus (*Matter Device Type*) megvalósításakor.

A Clusterek nem csupán funkcionalitásuk alapján oszthatók. Léteznek *kliens* és *szerver* típusú Clusterek is. A szerver rendelkezik állapottal és tárolja az attribútumokat, eseményeket, valamint a parancsokat, míg a kliens állapot nélküli és az interakciók kezdeményezéséért felelős. Ez magában foglalja a szerver parancsainak meghívását, valamint általános írási és olvasási műveleteket.

1.3.3. Biztonság és adatvédelem

A Matter protokoll jelentős erőfeszítéseket tesz a biztonság terén, védelmet nyújtva számos kiberfenyegetéssel szemben. Ezáltal a szabvány egy olyan biztonsági réteget alakított ki az okos eszközök számára, amelyben az adatok integritása nem függ más protokollok, például a WiFi vagy a Thread biztonsági intézkedéseitől. Ez a funkció a Matter protokoll alapvető részét képezi, amelyet minden Matter eszköznek meg kell valósítania.

A Matter protokoll a legmodernebb kriptográfiai technológiákat alkalmazza, amelyek a múltban már bizonyították hatékonyságukat a védelemben. Már a gyártási folyamat során az eszközök úgynevezett "*device certificate*"-eket, azaz gyártási tanúsítványokat kapnak, amelyeket szigorú tesztek előznek meg. Ezeket a tanúsítványokat a Connectivity Standards Alliance (CSA) állítja ki és a vásárolt eszközök hitelességét igazolják.

Amikor egy felhasználó megvásárol egy Matter kompatibilis eszközt, az egy egyedi azonosítóval rendelkezik, amely gyakran QR kód vagy számjegyekből álló kód formájában jelenik meg. Ez az azonosító lehetővé teszi a felhasználó számára, hogy igazolja az eszköz birtoklását, ami a "*proof-of-possession*" elvén alapul. Ezenkívül ezt a kódot felhasználják a közös kriptográfiai kulcsok generálásához is, amelyek kulcsfontosságúak a biztonságos kommunikáció szempontjából.

Miután az eszközök csatlakoznak a Matter hálózathoz, vagyis a Matter fabrichez, egy második, a hálózat commissioner-e által kiadott, *operational certificate*-et kapnak. Ez a tanúsítvány igazolja az eszközök hálózati jogosultságait, lehetővé téve számukra, hogy bizonyítottan részt vegyenek a Matter kommunikációs rendszerében.

A Matter protokoll biztonságos kommunikációt valósít meg több kriptográfiai algoritmus használatával, beleértve a SHA-256 [26], HMAC-SHA-256, ECC Curve NIST P-256 [15] és az AES 128-bites változatát. A hivatalos Matter specifikáció részletesen bemutatja ezeket a felsorolt biztonsági lépéseket.

1.4. Sprinkler mobilalkalmazás

A Sprinkler mobil alkalmazás egy Kotlin [14] nyelvben megírt, Android platformra tervezett Matter Controller. Segítségével könnyedén kommunikálhatunk eszközeinkkel egy felhasználóbarát felületen keresztül. Az alkalmazás jelenleg csak angol nyelven érhető el, használatát a 2. fejezet tárgyalja.

Az alkalmazás egy lokális Fabricet hoz létre, amelyre a felhasználók csatlakoztathatják

eszközöket, és irányíthatják azokat. Erre két környezetet is biztosít, név szerint *Greenhouse* és *Garden*. Ezek persze csak szimbolikus környezetek, hogy a felhasználó könnyebben megszerezhesse különböző eszközeit. Az alkalmazás emellett lehetőséget biztosít az eszközök be- és kikapcsolására, megosztására hasonló szolgáltatásokkal és statisztikák lekérésére a Sprinkler Hubon keresztül.

Az alkalmazás használata előtt a felhasználók át kell lépjenek egy *Auth0* autentikáción. Ehhez be kell, hogy jelentkezzenek Google felhasználójukkal, vagy egy, a rendszerre regisztrált email cím, jelszó párossal.

1.4.1. Kommunikáció a Sprinkler Hubbal

Az alkalmazás egy API-on keresztül kommunikál a Sprinkler Hubbal, amely az 1.5.4. fejezetben van bemutatva. A HTTP kérések küldésére az alkalmazás Ktor klienszt működtet. Ez segít a kérések egyszerű megfogalmazásában. Hibák esetén értesíti a felhasználót. Az 1. kódrészlet a Ktor kliens létrehozását, míg a 2. kódrészlet a statisztikák lekérését szemlélteti.

```
1 companion object {
2     private val httpClient = HttpClient(Android) {
3         install(ContentNegotiation) {
4             json(Json {
5                 prettyPrint = true
6                 isLenient = true
7                 ignoreUnknownKeys = true
8             })
9         }
10    }
11
12    fun create(): HttpService {
13        return HttpServiceImpl(client = httpClient)
14    }
15 }
```

1. kódrészlet. Ktor kliens létrehozása.

```
1 val response = client.get(HttpRoutes.fetchSensorDataURL(nodeId, endpointId,
2 ↪ clusterId)) {
3     url {
4         parameters.append("from", startUtcTimestamp)
5         parameters.append("to", endUtcTimestamp)
6         parameters.append("resolution", resolutionParameter)
7     }
8     header(HttpHeaders.Authorization, "Bearer $authToken")
9 }
```

2. kódrészlet. Statisztikák lekérése a Sprinkler Hubtól Ktor klienszt használva.

1.5. Sprinkler Hub

A projekt fejlesztése során számos probléma merült fel, melyeket sem az 1.3. fejezetben tárgyalt Matter protokoll, sem pedig a saját fejlesztésű Android alkalmazás nem tud megoldani. Ezek a problémák a következők :

- Az érzékelők által végzett mérések csupán pillanatnyi értékek, melyek nem kerülnek tárolásra. Ez azt eredményezi, hogy az alkalmazás nem képes statisztikai adatokat szolgáltatni az üvegház állapotáról a felhasználó számára.
- A Matter szabvány lokális hálózatot hoz létre, ami egy központi egység, a Matter Controller jelenlétét igényli, amely általában az üvegházban és annak Matter eszközei közelében van. Amennyiben ezt a szerepet a felhasználó mobiltelefonja tölti be, például a Sixth Sense Sprinkler alkalmazáson keresztül, akkor a készülékek kapcsolata megszűnik, ha a telefon távozik az üvegház területéről, létrehozva egy potenciális *single point of failure*-t
- A fejlesztés során felmerült az automatizálási rutinok megvalósításának a lehetősége, ami a nap 24 órájában működne, függetlenül a mobilalkalmazástól vagy a felhasználó fizikai jelenlététől.

Az összegzés alapján egy olyan eszköz létrehozása vált szükségessé, amely képes csatlakozni a Matter hálózathoz, figyelemmel kísérni az eszközök állapotváltozásait, és az információkat egy adatbázisban tárolni. Ezáltal a felhasználók hozzáférhetnek ezekhez az adatokhoz mobilkészülékükön keresztül.

1.5.1. Felépítése

A Sprinkler Hub egy Node.js [11] környezetben, Typescript [6] nyelvben megírt program amely egy Raspberry Pi 4-es [16] hardverre van kitelepítve, és az üvegház közelébe kell kihelyezni.

A hub Matter technológiát használ a matter.js¹⁵ könyvtáron keresztül, szerepet vállalva a Matter hálózatban mint Commissioner és Controller, így menedzselve az eszközöket. Adatok tárolására SQLite [23] adatbázist használ, melyhez a TypeORM [31] keretrendszer biztosít kapcsolatot. Az adatokat az Express.js [17] alapú webszerver teszi elérhetővé a mobilalkalmazás számára. A technológia verem megtekinthető a 9. ábrán.

¹⁵Javascript csomag, a Matter megvalósítása Typescriptben



9. ábra. Sprinkler Hub technológia verem

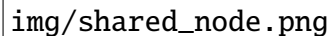
1.5.2. Matter hálózat vezérlése

A protokoll megvalósításához a matter.js könyvtár került felhasználásra, amely tartalmazza a főbb szerepkörökhöz szükséges forráskódot.

Ahogy a Matter hálózatot leíró fejezetben is szerepel (ld. 1.3.1) egy Fabricet csak egy Matter Controller képes létrehozni, valamint ahhoz, hogy további készülékekre lehessen csatlakozni Commissioner szerepkörrel kell kiegészíteni. Ezt a folyamatot a 3. kódszrészlet mutatja be. Újraindításakor a hub, az előzőleg párosított eszközökhöz próbál csatlakozni, ha a kapcsolat kialakítása nem sikeres (például, mert az eszköz már nem része a hálózatnak), a hub abbahagyja a próbálkozást és a sorban következő eszközzel próbálkozik.

Mivel a Raspberry Pi nem rendelkezik beépített kamerával, az eszközök párosításakor nem lehet QR kódot használni, helyette a párosító kódot alkalmazzák. Ez lehetővé teszi az eszközök azonosítását a hálózaton egy *discriminator* (megkülönböztető kód) segítségével. Ez a folyamat nem igényel újabb Bluetooth Low Energy (BLE) kapcsolatot a hub és az eszköz között, mivel a mobilalkalmazás korábban már csatlakoztatta az eszközt a lokális hálózathoz, így a multicast DNS protokollon keresztül érhető el.

Miután egy eszköz sikeresen csatlakozik, a hub feliratkozik az eszköz minden Clusterében található attribútumra, és figyelemmel kíséri ezek változásait.



10. ábra. Sprinkler Hub. Példa fabriccek közötti NodeId megfeleltetésre

Minden Matter eszköz egyedi NodeID-val rendelkezik, mind az Android fabricben, mind a hub fabricben. Mivel a mobilalkalmazás nem ismeri a hub által kiosztott NodeID-kat, ezeket az azonosítókat kommunikálni kell a hub felé a kérések során. A Sprinkler Hub egy adatbázisban nyilvántartja az eszközök hálózati azonosítóinak megfeleltetéseit (ld. 10. ábra), így biztosítva az eszközök azonosítását és kezelését.

1.5.3. Adattárolás

A Sprinkler Hub az alap Matter adatmodellen túl további adatokat is tárol az adatbázisában.

- A *shared_node* tábla az eszközök NodeID megfeleltetéseit tárolja a különböző fabriccek között, azaz a mobilalkalmazástól és a hubtól kapott egyedi azonosítókat.
- A *device_attribute* tábla az eszközök állapotváltozásai során mért adatokat rögzíti, beleértve a NodeID-t, az EndpointId-t, a clusterId-t (amelyek az adott attribútumot azonosítják), az attribútum nevét, a mért értéket, és a mérés időpontját.

Az SQLite adatbázis és a program közötti kapcsolatot a TypeORM biztosítja amely egy TypeScript alapú Object Relational Mapping (ORM) megoldás. Az ORM lehetővé teszi osztályok adatbázis táblákba való konvertálását és fordítva. Interfészeket nyújt a CRUD (create, read, update, delete) műveletek elvégzésére, ami leegyszerűsíti az adatokon történő alapl műveletek végrehajtását. Ezenkívül a TypeORM *QueryBuilder* funkciója elegáns lekérdezések készítését teszi lehetővé TypeScript szintaxis használatával, amelyek SQL parancsokká alakulnak át.

```

1 export class SprinklerController {
2     this.matterController = new MatterServer(this.storageManager, { mdnsInterface:
      ↪ this.netInterface });
3
4     this.commissioningController = new CommissioningController({
5         autoConnect: false,
6     });
7
8     await
      ↪ this.matterController.addCommissioningController(this.commissioningController);
9
10    this.matterController.start();
11 }

```

3. kódrészlet. Matter hálózat elindítása

1.5.4. API

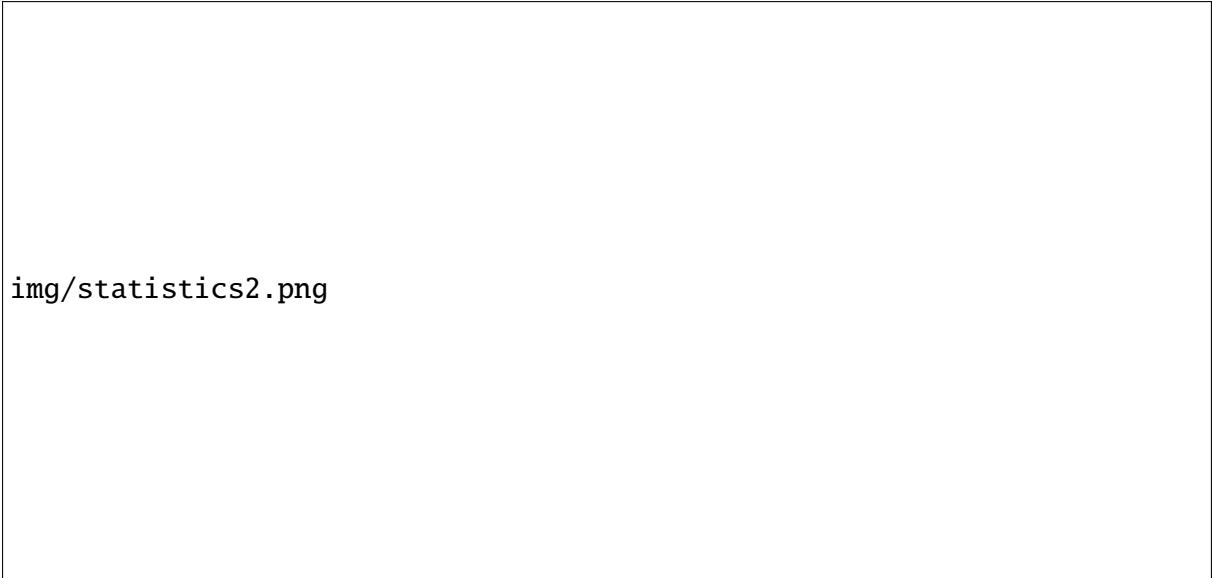
Az Application Programming Interface (API) egy szabályrendszer, amely lehetővé teszi a szoftverkomponensek közötti kommunikációt és adatcserét. A hubon futó webszerver egy Representational State Transfer (REST) alapú API-t használ, amely az erőforrásokhoz HTTP kérések segítségével fér hozzá.

A Sprinkler Hub képes HTTP GET és POST metódusú kéréseket fogadni. Ezek a kérések lehetővé teszik új eszközök hálózatba való integrálását, valamint lehetőséget biztosítanak a múltbeli adatok kliensek számára történő kiszolgáltatására. Az említett erőforrásokat a megfelelő útvonal megadásával lehet elérni, illetve lehetőség van query paraméterek megadására, az erőforrások szűréséhez vagy csoportosításához.

A főbb útvonalak közé tartozik a *POST /commission*, ami ismeretlen Matter eszközt vezet be a rendszerbe. A kérésnél kötelező megadni az Android fabricból származó NodeId-t, és a párosító kódot *JSON* objektum formájában. A *GET /values* útvonalról a rendszerben már jelen levő eszközök által mért adatokat lehet lekérni, melyeket opcionálisan szűrhetők a *from* és a *to* query paraméterekkel. Ha viszont egy készülék konkrét attribútumára szeretnénk rékérdezni, akkor a Matter specifikációban szereplő Endpoint és Cluster id-val kell az útvonalat kiegészíteni: */values/{nodeId}/endpoint/{endpointId}/cluster/{clusterId}*.

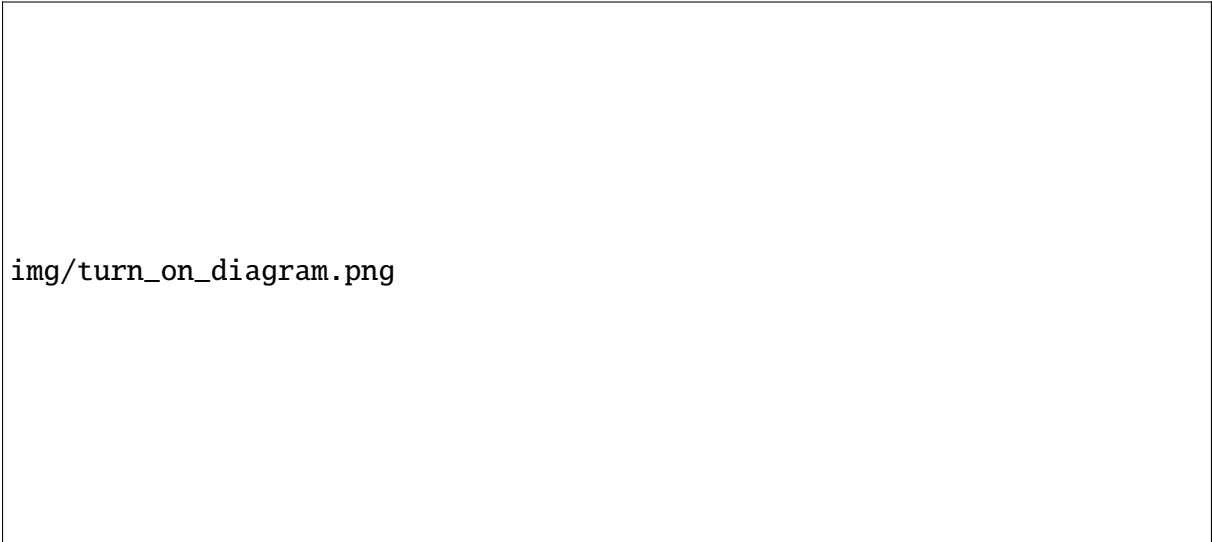
Mivel a mérések gyakorisága sok adatátvitellel jár, így az értékek havi, napi és óránkénti átlagainak lekérésére is lehetőség van. A mérések lekérési folyamata látható a 11. ábrán.

Az eszközök távoli irányításához a */toggle*, */turnOn*, vagy a */turnOff* útvonalra kell HTTP kérést küldeni. Egy eszköz bekapcsolásának folyamatát a 12. ábra mutatja be.



img/statistics2.png

11. ábra. A mérések lekérési folyamata a Sprinkler Hubtól



img/turn_on_diagram.png

12. ábra. Egy eszköz bekapcsolásának folyamata

1.5.5. Kérések validálása

A mobilalkalmazástól érkező kérések ellenőrzéséért a *middleware* csomagban található *authMiddleware* felelős. Minden HTTP kérés Authorization fejlécében egy JWT [29] alapú *ID Token* kell szerepeljen, melyet a felhasználó a mobilalkalmazásba történő belépés után szerez.

Amennyiben a kérésekből a fejléc vagy token hiányzik, vagy a token érvénytelen, a kérésre a szerver egy *HTTP 401 Unauthorized* hibakóddal válaszol.

2. A mobilalkalmazás felhasználása

Ebben a fejezetben bemutatjuk, hogyan alakíthatják ki a felhasználók saját okos üvegház rendszerüket a Sixth Sense Sprinkler eszközeinek segítségével. Ismertetjük a mikrokontrollerek gombjainak használatát, és külön hangsúlyt helyezünk az Android alkalmazás kezelésére, az elérhető funkciók gyakorlati alkalmazásának bemutatására.

2.1. Eszközök előkészítése

2.1.1. Thread Border Router előkészítése

Ahhoz, hogy használni tudjuk az érzékelőket, szükségünk van egy Thread Border Routerre. Ahogy már azt többször is említettük, mi a Google által gyártott Nest Hub 2nd Gen-t használjuk, melyet használat előtt szükséges előkészíteni. Ehhez le kell töltenünk a Google Home alkalmazást a telefonunkra, majd az Add -> Google Nest or partner device gomb segítségével bescannelni a Nest Hubon található QR kódot, majd követni az alkalmazásban megjelenő utasításokat. Pár percen belül a Nest Hub elindul, létre hozva a Thread hálózatot. Alternatíván a felhasználó más Border Routert is használhat.

2.1.2. Mikrokontrollerek és érzékelők előkészítése

Ami után az elemet először helyesen behelyeztük a tárolójába, a mikrokontrolleren lévő LED elkezd sárgán villogni. Ez azt jelenti, hogy BLE protokollon keresztül hirdeti magát. Ilyenkor üzembe helyezhetjük a Sprinkler alkalmazással a 2.4. alfejezet szerint. Ha netán mégsem sikerül az üzembe helyezés, akkor a mikrokontrolleren található fehér gomb 6 másodpercig tartó lenyomásával visszaállíthatjuk a gyári beállításokat (factory reset). Ekkor már villogni fog a sárga LED, jelezve, hogy készen áll az üzembe helyezésre.

Használat előtt érdemes a talajnedvesség érzékelőt kalibrálni. Nyomjuk meg a mikrokontrolleren található nagy piros gombot, amely hatására egy LED elkezd pirosan villogni. Ilyenkor arra vár, hogy az érzékelőt tartsuk a levegőbe. Ügyeljünk arra, hogy az érzékelőt az aljától fogva tartsuk, ne érzékelje a kezünket, csak a levegőt. Ha ezt megtettük, a megerősítéshez nyomjuk meg még egyszer a gombot. A LED pirosan fog világítani 3 másodpercig, jelezve, hogy méri a levegőben való nedvességet, amely egyértelműen alacsony kell, hogy legyen, ez az érték felel majd meg a 0%-nak. Ezután a LED elkezd kéken villogni. Ekkor tegyük az érzékelőt egy pohár vízbe, figyelve arra, hogy a fehér vonalon túl már ne



13. ábra. A bejelentkezés lépései.

érje víz, mert kárt tehet benne. Megerősítésért nyomjuk meg utoljára a gombot. A LED három másodpercig kéken fog világítani, jelezve, hogy a vízben méri a nedvességet, amely a 100%-os páratartalomnak fog megfelelni. Végül a LED egy másodpercre átvált zöldbe, jelezve, hogy sikerült a kalibráció, és az érzékelő készen áll a használatra.

2.2. Bejelentkezés

Az alkalmazás használatához először be kell jelentkezünk. A nyitó képernyőn egy “*Sign in with Google*” feliratú gomb fogad minket, melynek megnyomásával megjelenik egy weboldal, ahol bejelentkezhünk Google felhasználónkkal, vagy e-mail cím és jelszó párossal regisztrálhatunk. Sikeres bejelentkezés után az alkalmazás átirányít a főoldalra. A 13. ábrán megtekinthető a bejelentkezés lépései.

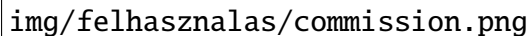


14. ábra. A főoldal, abban az esetben, ha nincs üzembe helyezett eszközünk, és abban az esetben, ha van.

2.3. Főoldal

A főoldal tetején, jobboldalon egy szöveg fogad minket, amely a jelenlegi környezet neve. Közvetlen mellette található egy ikon, melynek megnyomásával megjelenik egy legördülő menü. Segítségével környezetet válthatunk. Baloldalon a bejelentkezett fiókunk profilképe jelenik meg. Az oldal alján a navigációs bár látható, három opcióval: *Devices*, amely a főoldalra vezet, *History*, amely a grafikonokat generáló oldalra irányít, és a *Settings* amely a beállításokhoz navigál. Ez a navigációs bár az összes többi oldalon is megtalálható. A navigációs bár felett van egy *Add Device* gomb, amely segítségével eszközöket helyezhetünk üzembe.

Az oldal közepén a már üzemben lévő eszközeinket láthatjuk kártyák formájában. Egy kártyán megjelenik az eszköz neve, egy ikon, amely az eszköz típusát jelzi (talajnedvesség, vagy hőmérséklet és páratartalom érzékelő), valamint az ON vagy OFF felirat, a szerint, hogy

A screenshot of a mobile application interface, likely for a smart home system. The image shows a step in the process of adding a device, with the text 'img/felhasznalas/commission.png' overlaid on the screen.

15. ábra. Egy eszköz üzembe helyezése a mobil alkalmazás segítségével.

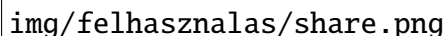
az eszköz be van kapcsolva, vagy sem. Ha még nincsenek üzembe helyezett eszközeink, akkor az oldal közepe üres. A 14. ábrán a főoldal látható üzembe helyezett eszközök nélkül és üzembe helyezett eszközökkel is.

2.4. Eszközök üzembe helyezése

Az eszközök üzembe helyezéséhez az *Add device* gombot kell megnyomni. Ilyenkor az alkalmazás az aktívan kiválasztott környezethez próbálja hozzáadni az eszközt. Miután megnyomtuk a gombot, és engedélyeztük az alkalmazásnak a wifi, bluetooth, valamint a kamera használatát, be kell szkenneljük az eszközön található Matter QR kódot. Ha túl sötét van, akkor megnyomhatjuk a képernyőn található zseblámpa ikont, bekapcsolva ezzel a telefon vakuját. Alternatíván a *Set up without QR code* gombra kattintva az eszközök párosító kódját is használhatjuk. A következő lépésben már csak várnunk kell 1-2 percet, amíg az eszköz bekerül a Matter Fabricbe. Ha ez megtörténik, akkor egy előugró ablak jelenik meg, ahol az eszköznek egy tetszőleges nevet adhatunk. A 15. ábrán az üzembe helyezés lépései láthatóak. Figyeljünk arra, hogy az eszköz legyen minél közelebb a Nest Nubhoz (lehetőleg közvetlen mellette) az üzembe helyezés során, és legyen megfelelően előkészítve.

2.5. Eszközök kezelése

Az üzembe helyezett eszközökkel a következő műveleteket végezhetjük: Az eszközöket ki/be kapcsolhatjuk, megoszthatjuk őket egyéb szolgáltatásokkal, mint például a Google Home, vagy a Sprinkler Hub, valamint törölhetjük őket.

A screenshot of a mobile application interface showing the sharing process for a Sprinkler Hub. The image is currently missing, but the text indicates it shows the steps for sharing the device.

16. ábra. A Sprinkler Hubbal való megosztás lépései a mobil alkalmazásban.

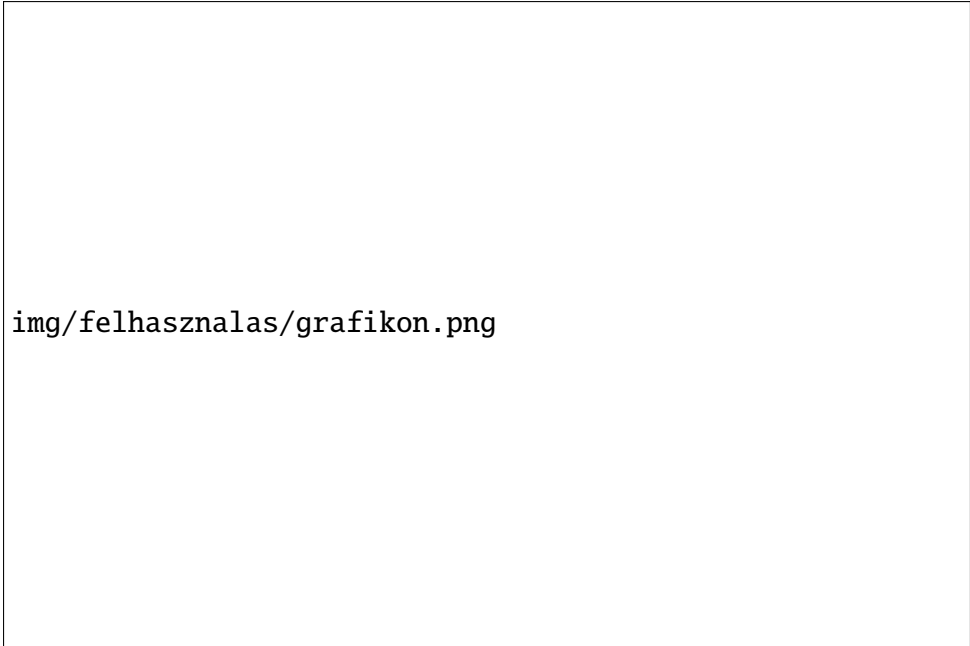
Az eszközök kapcsolgatásához bökjünk rá a kívánt eszköz kártyájára a főoldalon. Ha a kártyára hosszan nyomunk, akkor megjelenik az eszköz saját oldala, ahol szintén kapcsolgathatjuk. Ezen az oldalon végezhető el az eszközhöz tartozó többi művelet is.

Az eszköz megosztásához nyomjunk rá a *Share Device* gombra. A megjelenő ablakban válasszuk ki a szolgáltatást, ahova meg szeretnénk osztani. Itt a Sprinkler Hub nem fog megjeleni, ezért ha azt kívántjuk, akkor a *Use pairing code* opciót kell kiválasztanunk. Másoljuk ki a megjelent párosítási kódot, majd navigáljuk vissza az eszköz oldalához. Itt nyomjuk meg az *Add to Sprinkler Hub* gombot. A megjelent szövegdobozba illesszük be a kódot, és nyomjuk meg a *Confirm* gombot. Egy kis idő elteltével a művelet befejeződik. A 16. ábrán megtekinthető a Sprinkler Hubbal való megosztás lépései.

Az eszköz törléséhez nyomjuk be a *Remove Device* gombot. Az alkalmazás kérni fogja, hogy erősítsük meg a törlési szándékunkat. Miután a *Confirm* gombra rányomtunk, az eszköz törlődik a rendszerből.

2.6. Grafikonok lekérése

Ahhoz, hogy a múltbéli mérésekről adatokat kérjünk le és grafikonokat készítsünk belőle, navigáljunk a *History* menüpontra. Itt három legördülő menüben állítsuk be, hogy melyik érzékelő, melyik mérés érdekel minket, és válasszuk ki, hogy milyen felbontásban (percenként, óránként, napi vagy havi) szeretnénk megtekinteni az adatokat. Majd az *Open Date Picker* gombbal egy naptár jeleníthető meg. Itt adjunk meg egy kezdő és végső dátumot. Naptár helyett a ceruza ikonra nyomva manuálisan beírhatjuk a két dátumot. Miután kész vagyunk, nyomjunk



img/felhasznalas/grafikon.png

17. ábra. Grafikon lekérésének lépései a mobil alkalmazásban.

rá a pipára, és megjelenik a kívánt grafikon. Természetesen a kiválasztott eszköz megosztva kell, hogy legyen a Sprinkler Hubbal, hiszen onnan érkezik majd az adat. Ellenkező esetben nem fogunk grafikont látni. A 17. ábrán láthatóak a grafikon lekérésének a lépései.

3. Technológiák

A dolgozat ezen fejezetében bemutatásra kerülnek azok a felhasznált technológiák és eszközök, amelyek a Sixth Sense Sprinkler projekt fejlesztése során kerülnek használatra.

3.1. Git és Gitlab

A Git egy ingyenes és nyílt forráskódú verziókövető rendszer, amely lehetővé teszi a kódok különböző változatainak nyomon követését, felgyorsítva a fejlesztők közötti kollaborációt.

A Gitlab, ahogy a neve is jelzi, egy Gitre épülő weboldal, amely számos hasznos funkcióval ellátja a felhasználót: lehetőséget nyújt a kódbázis tárolására és megosztására. Védelmet biztosít privát repositorykon keresztül. A fejlesztők különböző metodológiákat követve, felvezethetik a felmerülő problémákat és feladatokat. Egy fejlesztő által elkészített munkát ellenőrizni lehet, mielőtt azt felvezetnék a kódbázisba. Ugyanakkor pipeline-okat használva lehetőség van automatizációk elvégzésére, mint például tesztek futtatására.

3.2. nRF Connect

Az NRF Connect SDK¹⁶ egy fejlesztői csomag a Nordic Semiconductors mikrokontrollereihez, amely speciális könyvtárakat tartalmaz a Matter over Thread eszközök fejlesztésének felgyorsítására. Ilyenek például az OpenThread, valamint a Thread hálózati topológiát monitorizáló nRF Sniffer.

Az nRF Connect for Desktop egy asztali alkalmazás, amit a mikrokontrollerek fejlesztésére és tesztelésére használnak, támogatva számos protokollt, mint például a BLE. A projekt során különösen hasznosnak bizonyult a Programmer eszköz, amely lehetővé teszi firmware egyszerű feltöltését Matter eszközökre, valamint a Serial Terminal, ami a mikrokontroller és a számítógép közötti soros kommunikációt valósítja meg.

Megemlíthető még a Toolchain Manager, amellyel az SDK verziókat lehet követni és telepíteni, lehetőség nyílik ezen keresztül egy VSCode kiegészítő telepítésére is.

¹⁶Forrás: <https://www.nordicsemi.com/Products/Development-software/nRF-Connect-SDK>

3.3. Zap Tool

A zap-tool¹⁷, vagyis a ZCL Advanced Platform egy konfigurációs eszköz, amelyet kifejezetten Matter Cluster sablonok készítésére fejlesztettek. Eredetileg csak a Zigbee Cluster Library-t tartalmazta, majd a Matter protokoll megjelenésével (A Matter Cluster Library alapja a ZCL), Matter eszközökre is kibővítették.

Segítségével könnyedén létrehozhatunk egy Matter eszköz-sablont, melyből a valós firmware boilerplate részét képes kigenerálni C és C++ kódok formájában. A programon belül létrehozhatunk Endpointkat, ezekhez hozzárendelhetjük a Matter specifikációban is megjelenő Clustereket, valamint attribútumokat és parancsokat adhatunk ezekhez, vagy távolíthatunk el.

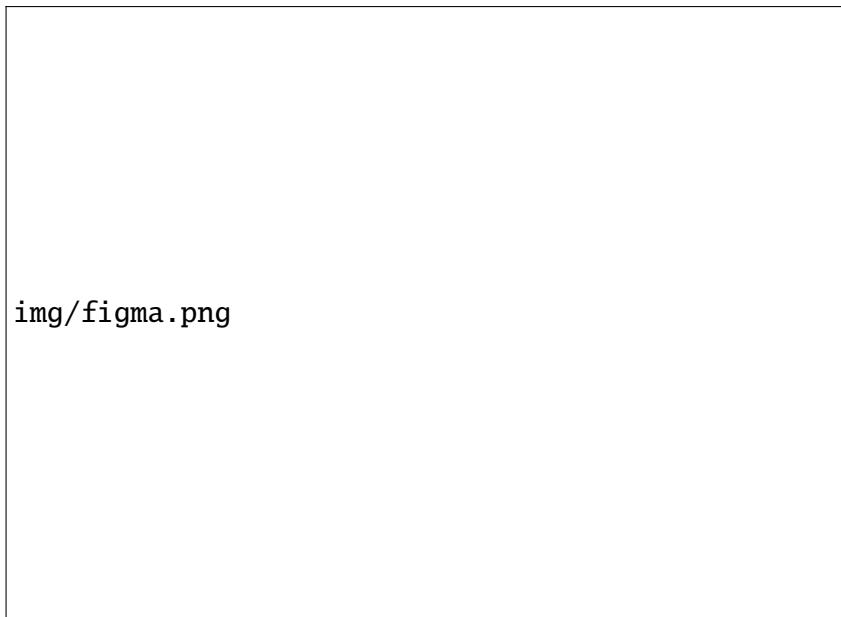
3.4. Kotlin, Ktor és Jetpack Compose

A Kotlin egy modern, nyílt forráskódú programozási nyelv, amelynek elsődleges célja a Java helyettesítése a JVM (Java Virtual Machine) platformon. A Kotlin támogatja a funkcionális és objektumorientált programozási paradigma összekapcsolását, amely lehetővé teszi a hatékony és kényelmes kódolást. Az Android alkalmazásfejlesztésben széles körben alkalmazzák a Kotlin nyelvet a Java helyett, mivel modernebb és kevésbé boilerplate kódot hoz létre.

A Ktor egy könnyűsúlyú, modern webes keretrendszer, amelyet a Kotlin nyelvhez terveztek. A Ktor segítségével könnyedén fejleszthetünk aszinkron, nagyteljesítményű webalkalmazásokat vagy mikroszolgáltatásokat. A keretrendszer lehetővé teszi a gyors és egyszerű beállítást, valamint a kiterjeszhető architektúrákat, amelyekkel könnyedén kezelhetjük a HTTP kéréseket és válaszokat. A Ktor nagyrészt szerveroldali alkalmazások fejlesztésében terjedt el, de megfelelő kliensek programozására is, a projektben erre használtuk.

A Jetpack Compose egy modern UI keretrendszer Kotlin alkalmazásokhoz, amely fokozatosan felváltja a hagyományos XML alapú felhasználói felületeket. A Jetpack Compose segítségével deklaratív módon készíthetünk felhasználói felületeket, ami könnyű és intuitív módon teszi lehetővé a UI elemek kialakítását és működését. Az egyszerűség és a rugalmasság révén a Jetpack Compose lehetővé teszi a gyorsabb és hatékonyabb Android alkalmazásfejlesztést.

¹⁷Forrás: <https://github.com/project-chip/zap?tab=readme-ov-file>



18. ábra. Figmaban készült UI tervek

3.5. Figma

A Figma egy népszerű webalkalmazás, amelyet UI tervezésre fejlesztettek. Támogatja a valós idejű együttműködést, vektorgrafikát, ikonokat és alakzatokat, valamint eszközöket nyújt prototípusok és animációk készítésére. Projektünk során a mobilalkalmazás vizuális tervezésére használtuk (ld. 18. ábra).

Következtetések és továbbfejlesztési lehetőségek

A Sixth Sense Sprinkler projekt keretében olyan Matter-kompatibilis eszközöket fejlesztettünk ki, amelyek képesek érzékelők segítségével a talajnedvesség, hőmérséklet és levegő páratartalom mérésére és továbbítására egy Matter Hub felé. Az elkészített Android alkalmazásunk segítségével a felhasználók könnyedén kezelhetik eszközeiket, ezáltal hatékonyabbá téve az üvegházak karbantartását. A Sprinkler Hub szerverünkön keresztül a felhasználók számos statisztikát is megtekinthetnek eszközeik múltbeli méréseiről.

A jövőben törekszünk arra, hogy számos új funkcióval még jobbá és hasznosabbá tegyük a projektet. Az alkalmazás továbbfejlesztésének részeként tervezzük egy felhasználói felület kialakítását, amely lehetővé teszi a felhasználók számára az automatizációs szabályok testreszabását. Ez kritikus fontosságú, mivel minden növény más környezeti feltételeket igényel, így fontos, hogy a gazdák személyre szabottan beállíthassák a rendszert a saját növényeik igényeinek megfelelően.

Egy másik fontos fejlesztési cél, hogy az Android alkalmazásban látható legyen az érzékelők méréseinek valós idejű értékei is, ne csak a grafikonok megjelenítésével legyenek olvashatók. Emellett az alkalmazás értesítéseket küldhetne a rendszer állapotáról. Például, ha a magas hőmérséklet miatt beindul a szellőztető rendszer, vagy meghibásodik egy eszköz, és lecsatlakozik a rendszerről, érdemes ezen eseményekről értesíteni a felhasználót.

Jelenleg az eszközöket csak "Greenhouse" és "Garden" helyiségekbe lehet csoportosítani. A jövőben tervezzük, hogy a felhasználóknak lehetőségük legyen saját környezeteket létrehozni és elnevezni, hogy azok jobban tükrözzék saját üvegházait.

Nem utolsó sorban terveink között szerepel az eszközök Over The Air frissítési funkciójának bevezetése, hogy a jövőben megjelenő frissítéseket egységesen és távolról lehessen telepíteni és frissíteni. Ezáltal biztosítva a rendszer folyamatos fejleszthetőségét és javíthatóságát.

Hivatkozások

- [1] Yasser Asrul Ahmad és mások. „On the Evaluation of DHT22 Temperature Sensor for IoT Application”. *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*. 2021, 131–134. oldal. DOI: 10.1109/ICCCE50029.2021.9467147.
- [2] Sami S. Albouq és mások. „A Survey of Interoperability Challenges and Solutions for Dealing With Them in IoT Environment”. *IEEE Access* 10 (2022), 36416–36428. oldal. DOI: 10.1109/ACCESS.2022.3162219.
- [3] *Build With Matter*. URL: <https://csa-iot.org/all-solutions/matter/> (elérés dátuma: 2024. máj. 2.)
- [4] Keuchul Cho és mások. „Analysis of Latency Performance of Bluetooth Low Energy (BLE) Networks”. *Sensors* 15.1 (2015), 59–78. oldal. ISSN: 1424-8220. URL: <https://www.mdpi.com/1424-8220/15/1/59>.
- [5] *Connectivity Standard Alliance hivatalos oldala*. URL: <https://csa-iot.org> (elérés dátuma: 2024. ápr. 17.)
- [6] Fernando Doglio. „An Introduction to TypeScript”. *Introducing Deno: A First Look at the Newest JavaScript Runtime*. Berkeley, CA: Apress, 2020, 27–62. oldal. ISBN: 978-1-4842-6197-2. DOI: 10.1007/978-1-4842-6197-2_2. URL: https://doi.org/10.1007/978-1-4842-6197-2_2.
- [7] *Google Home*. URL: <https://home.google.com/get-app/> (elérés dátuma: 2024. ápr. 29.)
- [8] Abdullah Al Hasib és Abul Ahsan Md. Mahmudul Haque. „A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography”. *2008 Third International Conference on Convergence and Hybrid Information Technology*. 2. kötet. 2008, 505–510. oldal. DOI: 10.1109/ICCIT.2008.179.
- [9] *Home app*. URL: <https://www.apple.com/home-app/> (elérés dátuma: 2024. ápr. 29.)
- [10] *Home Assistant*. URL: <https://www.home-assistant.io/> (elérés dátuma: 2024. ápr. 29.)
- [11] Xiaoping Huang. „Research and Application of Node.js Core Technology”. *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*. 2020, 1–4. oldal. DOI: 10.1109/ICHCI51889.2020.00008.
- [12] Sohaib Bin Altaf Khattak és mások. „Performance Evaluation of an IEEE 802.15.4-Based Thread Network for Efficient Internet of Things Communications in Smart Cities”. *Applied Sciences* 13.13 (2023), 7745. oldal. DOI: 10.3390/app13137745. URL: <https://www.mdpi.com/2076-3417/13/13/7745>.

- [13] Hyung-Sin Kim, Sam Kumar és David E. Culler. „Thread/OpenThread: A Compromise in Low-Power Wireless Multihop Network Architecture for the Internet of Things”. *IEEE Communications Magazine* 57.7 (2019), 55–61. oldal. DOI: 10.1109/MCOM.2019.1800788.
- [14] *Kotlin Programming Language*. URL: <https://kotlinlang.org/> (elérés dátuma: 2024. máj. 2.)
- [15] Zhe Liu és mások. „Memory-Efficient Implementation of Elliptic Curve Cryptography for the Internet-of-Things”. *IEEE Transactions on Dependable and Secure Computing* 16.3 (2019), 521–529. oldal. DOI: 10.1109/TDSC.2018.2825449.
- [16] T Maragatham, P Balasubramanie és M Vivekanandhan. „IoT Based Home Automation System using Raspberry Pi 4”. *IOP Conference Series: Materials Science and Engineering* 1055.1 (2021. febr.), 12081. oldal. DOI: 10.1088/1757-899X/1055/1/012081. URL: <https://dx.doi.org/10.1088/1757-899X/1055/1/012081>.
- [17] Azat Mardan. „Using Express.js to Create Node.js Web Apps”. *Practical Node.js: Building Real-World Scalable Web Apps*. Berkeley, CA: Apress, 2018, 51–87. oldal. URL: https://doi.org/10.1007/978-1-4842-3039-8_2.
- [18] *Matter Smart Plug with Energy Monitor*. URL: <https://www.meross.com/en-gc/smart-plug/matter-smart-plug/159> (elérés dátuma: 2024. ápr. 30.)
- [19] Radouan Ait Mouha. „Internet of Things (IoT)”. *Journal of Data Analysis and Information Processing* 9.2 (2021), 77–101. oldal.
- [20] Emerson Navarro, Nuno Costa és António Pereira. „A Systematic Review of IoT Solutions for Smart Farming”. *Sensors* 20.15 (2020). DOI: 10.3390/s20154231. URL: <https://www.mdpi.com/1424-8220/20/15/4231>.
- [21] *Nest Hub (2nd Gen)*. URL: https://store.google.com/us/product/nest_hub_2nd_gen (elérés dátuma: 2024. ápr. 29.)
- [22] *Nordic Semiconductor*. URL: <https://www.nordicsemi.com/> (elérés dátuma: 2024. ápr. 30.)
- [23] Dirk Pawlaszczyk. „SQLite”. *Mobile Forensics – The File Format Handbook: Common File Formats and File Systems Used in Mobile Devices*. Szerkesztette Christian Hummert és Dirk Pawlaszczyk. Cham: Springer International Publishing, 2022, 129–155. oldal. URL: https://doi.org/10.1007/978-3-030-98467-0_5.
- [24] Pisana Placidi és mások. „Characterization of Low-Cost Capacitive Soil Moisture Sensors for IoT Networks”. *Sensors* 20.12 (2020), 3585. oldal. DOI: 10.3390/s20123585. URL: <https://www.mdpi.com/1424-8220/20/12/3585>.

- [25] *Pompa de apa 3-6V*. URL: https://ardushop.ro/ro/electronica/390-pompa-de-apa-3-6v.html?search_query=pompa&results=4 (elérés dátuma: 2024. máj. 5.)
- [26] D Rachmawati, J T Tarigan és A B C Ginting. „A comparative study of Message Digest 5(MD5) and SHA256 algorithm”. *Journal of Physics: Conference Series* 978.1 (2018. márc.), 12116. oldal. DOI: 10.1088/1742-6596/978/1/012116. URL: <https://dx.doi.org/10.1088/1742-6596/978/1/012116>.
- [27] Semios. *Insect Pest Management Solutions*. URL: <https://semios.com/solutions/insect-pest-management> (elérés dátuma: 2024. ápr. 29.)
- [28] Sensaphone. *Cloud-Based Remote Monitoring - Sentinels*. URL: <https://www.sensaphone.com/sentinels> (elérés dátuma: 2024. ápr. 29.)
- [29] Prabath Siriwardena. „JWT, JWS, and JWE”. *Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE*. Berkeley, CA: Apress, 2014, 201–220. oldal. URL: https://doi.org/10.1007/978-1-4302-6817-8_13.
- [30] *Thread Group specifikációja az eszközök típusairól és szerepeiről*. URL: https://www.threadgroup.org/Portals/0/documents/support/Thread%20Network%20Fundamentals_v3.pdf#page=8 (elérés dátuma: 2024. ápr. 20.)
- [31] *TypeORM - Amazing ORM for TypeScript and JavaScript*. URL: <https://typeorm.io/> (elérés dátuma: 2024. máj. 2.)
- [32] *Ventilator 5V 30mm Raspberry Pi*. URL: https://ardushop.ro/ro/home/800-ventilator-raspberry-pi-3.html?search_query=fan&results=16 (elérés dátuma: 2024. máj. 5.)
- [33] *What is REST?* URL: <https://restfulapi.net/> (elérés dátuma: 2024. ápr. 29.)
- [34] Wondimu Zegeye, Ahamed Jemal és Kevin Kornegay. „Connected Smart Home over Matter Protocol”. *2023 IEEE International Conference on Consumer Electronics (ICCE)*. 2023, 1–7. oldal. DOI: 10.1109/ICCE56470.2023.10043520.