



# Programozható elektronikák alkalmazásai

**Készítette**

Bagoly Gábor

Programtervező informatikus

**Témavezető**

Dr. Geda Gábor

Egyetemi docens

EGER, 2023

# Tartalomjegyzék

<b>Bevezetés</b>	<b>4</b>
<b>1. Piacon lévő jelentősebb okos otthon rendszerek</b>	<b>5</b>
1.1. Nagyobb cégek által létrehozott ökoszisztémák . . . . .	5
1.1.1. Mik is a virtuális asszisztensek? . . . . .	6
1.1.2. Amazon rendszere - Alexa Smart Home . . . . .	6
1.1.3. Google rendszere - Google Home és Google Nest . . . . .	7
1.1.4. Xiaomi rendszere - Mi Home . . . . .	8
1.2. Nyílt forráskódú rendszerek . . . . .	9
1.2.1. Mit is az, hogy open source? . . . . .	9
1.2.2. Home Assistant . . . . .	9
1.2.3. OpenHAB . . . . .	10
<b>2. Alkalmazott eszközök</b>	<b>11</b>
2.1. Hardver . . . . .	11
2.1.1. Mi is az hogy IoT? . . . . .	11
2.1.2. Raspberry Pi 4B . . . . .	11
2.1.3. NodeMCU - ESP-WROOM-32 . . . . .	12
2.1.4. ESP32-CAM - Wi-Fi-s kamera modul . . . . .	13
2.1.5. DHT22 - hőmérséklet és páratartalom szenzor . . . . .	14
2.1.6. Mi is az RFID? . . . . .	14
2.1.7. RFID-RC522 - RFID olvasó . . . . .	15
2.1.8. Szilárdtest relé . . . . .	15
2.2. Szoftverek és Programozási nyelvek . . . . .	16
2.2.1. C++ és Arduino IDE . . . . .	16
2.2.2. HTML és CSS . . . . .	17
2.2.3. PHP . . . . .	17
2.2.4. JavaScript . . . . .	17
2.2.5. jQuery . . . . .	18
2.2.6. Chart.js . . . . .	18
2.2.7. Tailwind CSS . . . . .	18
2.2.8. dbDiagram.io . . . . .	19

2.2.9. Font Awesome . . . . .	19
2.2.10. MySQL . . . . .	20
2.2.11. PlantUML . . . . .	20
2.2.12. Fritzing . . . . .	20
2.3. Laravel . . . . .	21
<b>3. A web alkalmazás felépítése és működése</b>	<b>22</b>
3.1. Raspberry PI4B alkalmazása, mint Wi-Fi, és hub . . . . .	22
3.2. Adatbázis felépítése . . . . .	22
3.3. Kezelő felület bemutatása és működése . . . . .	22
3.3.1. Főoldal . . . . .	22
3.3.2. Beállítások . . . . .	22
3.3.3. RFID beállítások . . . . .	22
3.3.4. RFID használati táblázat . . . . .	22
3.3.5. Hőmérsékleti és páratartalom előzmények - ChartJS . . . . .	22
3.4. Eszközök kommunikációja a webszerverrel . . . . .	22
3.4.1. Hőmérséklet és Páratartalom szenzor . . . . .	22
3.4.2. Eszköz kapcsoló . . . . .	22
3.4.3. Kamera . . . . .	22
3.4.4. RFID kártya olvasó . . . . .	22
<b>4. Tesztelések</b>	<b>23</b>
4.1. Tesztelések módjai és fontossága . . . . .	23
4.1.1. Cypress automatizált tesztelések . . . . .	23
4.1.2. Manuális tesztelések . . . . .	23
<b>5. Rendszer telepítése</b>	<b>25</b>
<b>Összegzés</b>	<b>26</b>
<b>Irodalomjegyzék</b>	<b>27</b>

# Bevezetés

A mai világunkban az okos otthonok és az okos eszközök rendkívül népszerűek és elterjedtek lettek. Általában, hogy ha megkérdezzük valakit ezzel a témával kapcsolatban, akkor nagy valószínűséggel azt tudják elmondani, hogy rendelkeznek legalább egy okos otthonban alkalmazható eszközzel.

Az okos otthonok és az okos eszközök lehetővé teszik a tulajdonosaik számára, hogy akár távolról is vezéreljék és felügyeljék az otthonukat. Ezek az eszközök lehetővé tehetik a kényelmesebb és hatékonyabb életmód megvalósítását.

Manapság már ha valaki az interneten rákeres valamilyen otthoni eszközre, és eléje írja azt hogy „okos”, akkor bizonyára léteznek már erre megoldások. - például okos WC-k, okos kenyérpíritók, még akár okos szőnyegek is, és még sorolhatnám ezt a listát tovább.

De mi is tesz egy eszközt okossá? Feltételezhetjük azt, hogy ha valami internetre kapcsolódik, esetleg távolról beállíthatjuk, vagy automatizálhatjuk előre meghatározott dolgokra, akkor azt az eszközt „okosnak” tudjuk mondani.

Hogy ha valaki már rendelkezik több ilyen eszközzel, akkor bizonyára találkoztak már azzal a problémával, hogy az egy bizonyos ökoszisztémában használatos eszközök nem feltétlenül tudnak működni egy másikban. Minderre próbáltam egy olyan megoldást kitalálni, ami abból a szempontból közelíti meg, hogy még egy 'nem okos' eszközt (például egy izzót) integrálok bele úgy a rendszerbe, hogy ezt egyszerűen tudjunk kezelni bárhonnan az otthonunkból, ehhez társulva különböző programozható elektronikák. Mindehhez egy olyan webes alkalmazást hoztam létre, amely felelős a háttérben lévő folyamatok lebonyolítására, és az általunk ismert legtöbb eszközön alkalmazható, amin internetezni is tudunk: legyen az számítógép, Androidos, vagy iOS-es telefon, tablet, vagy akár okos óra is.

Azért került erre a témára a választásom, mivel számomra felettébb érdekes az, hogy egy ilyen okos otthonban az eszközök hogyan is kommunikálnak, és szeretnék ebbe egy belátást nyerni, hogy hogyan is épül össze mindez.

Céлом az lenne ezzel, hogy belelássak egy ilyen rendszer működésébe, különböző programozható eszközök alkalmazását jobban megismerjem, és, hogy egy olyan általános kezelőfelületet tudjak létrehozni, amit könnyen tud a felhasználó alkalmazni.

# 1. fejezet

## Piacon lévő jelentősebb okos otthon rendszerek

### 1.1. Nagyobb cégek által létrehozott ökoszisztémák

Az okos otthon rendszerek széles választéka áll rendelkezésre a piacon felhasználók számára, amelyek között egyszerű rendszerektől kezdve a biztonsági kamerák, ajtózárok és akár a háztartási készülékek is alkalmazhatóak. Amikor az okos otthon rendszer kiválasztására kerül sor, fontos figyelembe venni az eszközök kiválasztását is.

Van néhány olyan eszköz, - például okos izzók vagy okos dugaljak -, amelyek több okos otthon rendszerrel is szabadon használhatóak, így könnyen integrálhatóak a választott rendszerbe. Ezek az eszközök általában ipari szabványokat használnak, mint például a Wi-Fi, Bluetooth LE vagy a Zigbee<sup>1</sup>, ami lehetővé teszi számukra, hogy kommunikáljanak a különböző okos otthon rendszerekkel.

Azonban vannak olyan eszközök is, amelyek kifejezetten egyetlen rendszerrel működnek együtt. Ezek az eszközök saját protokollokat vagy szoftvereket használhatnak, amelyek nem kompatibilisek más rendszerekkel, ami korlátozhatja az okos otthon rendszerünk rugalmasságát. Vegyük példának azt, hogy ha olyan okos ajtózárát vásárlunk, amely csak egy meghatározott okos otthon rendszerrel működik együtt, akkor az eszközt később nem tudjuk használni egy másik rendszerrel. Vagy hogy ha veszünk egy adott típusú eszközt, akkor azt későbbiekben nem tudjuk kombinálni egy másik rendszer eszközeivel.

Összességében az okos otthon eszközök kiválasztásakor fontos figyelembe venni a kompatibilitást választott okos otthon rendszerünkkel. Az iparági szabványokat támogató eszközök általában rugalmasabbak és szélesebb körű rendszerekkel használhatóak, míg a saját protokollokat használó eszközök korlátozottabbak lehetnek a kompatibilitás terén.

---

<sup>1</sup> Zigbee-t használ például az Amazon Echo, az IKEA okos otthon termékei, és a Philips Hue

### 1.1.1. Mik is a virtuális asszisztensek?

Virtuális asszisztensek olyan szoftver programok, amik különböző feladatokat tudnak elvégezni számunkra, amit például egy elő segéd is képes lenne. Annyi különbséggel, hogy ezek az asszisztensek online, vagy az eszközeinken léteznek.

Úgy lettek megalkotva, hogy a felhasználó hang vagy chat alapon tudjanak szimpla kérdésekkel kommunikálni és utasításokat adni. A virtuális asszisztensek olyan feladatokban tudnak segíteni, mint emlékeztetők létrehozása, üzenetek küldése, hívások indítása, internetes keresés, időjárás előrejelzések felolvasása, okos otthoni eszközök irányítása, és egyéb más dolgok.

Számos cég hozott már létre magának virtuális asszisztenst, amik közül a legismertebbek lehetnek a Google által létrehozott Google Asszisztens, Apple-nek Siri, Amazon-nak Alexa, Samsung Bixby-je, és Microsoft Cortana-ja.

### 1.1.2. Amazon rendszere - Alexa Smart Home

2014-ben lépett be a piacra az Amazon - *az akkor még újdonságnak számító* - okos hangszórójukkal, az Amazon Echo-val. Ekkor még leginkább csak annyira volt képes, hogy a felhasználó zenét tudja irányítani hang utasításokkal. Ez az eszköz úgy működik, hogy ebbe bele van integrálva az Amazon sajátos virtuális asszisztense, amit Alexának hívnak. Ugye mint kezdetleges szoftver, neki sem voltak a képességei túl szerteágazóak. Leginkább arra tudta az ember használni, hogy egyszerű kérdéseket tegyen fel az ember, és zenét tudjon elindítani, leállítani átugrani.

Nem is annyira később, amikor elkezdett egyre jobban fejlődni Alexa, úgy egyre több mindenre kezdhette el használni az ember: termosztátok beállítása, izzók ki- és bekapcsolására is már lehetett használni. Miután az Amazon egyre többet fektetett bele a rendszerükbe felettébb szerteágazó lett annak a használata az otthonokban. Még arra is volt lehetőség már ekkor, hogy akár bevásárló listát készítsen, és azokat meg is tudja rendelni a felhasználó az Amazonról, mindezt Alexa használatával. A cég 2017 május 23-án jelentette be, hogy a *Smart Home Skill API*-jukba<sup>2</sup> innentől kezdve megadható, hogy milyen eszközöket is csatlakoztatunk a rendszerükbe, ami kitárta a lehetőségeket az otthoni okos készülékek automatizációjára. Mindez vezetett az okos otthon piac és a virtuális asszisztensek szerepének növekedéséhez.

Eszközök leginkább hangvezérléssel irányíthatóak, miután követtük az általuk biztosított használati útmutatót. Néhány esetben az Amazon Alexa alkalmazás elegendő lehet, de ha olyan eszközt szeretnénk használni, amelyhez saját alkalmazás tartozik, akkor azt is le kell töltenünk.

Az Amazon által szolgáltatott okos otthon rendszer 2023-ra olyan népszerűségi szintre jutott, hogy az Amerikai Egyesült Államokban az ilyen hang vezérelt hang-

---

<sup>2</sup> A blog poszt amiben bejelentették a Smart Home Skill API bővítését[1]

szórók 68.2%-a az Amazon Echo.<sup>3</sup> Mindezzel napjainkban több ezer eszköz használható már ezen a rendszeren belül: legyen az biztonsági rendszer, háztartási gépek, vagy esetleg szórakoztató rendszerek. Érthető is, hogy sokan miért is szeretik ezt a rendszert használni.

Azonban ennek a rendszernek is megvannak azok a hátulütői, hogy nem minden eszközt tudjunk beintegrálni az adott okos otthon rendszerünkbe, amik nem támogatottak az Amazon által. Érdeemes arra odafigyelni, hogy amikor egy okos eszközt veszünk, hogy arra fel-e van tüntetve, hogy kompatibilis az Amazon rendszerével. Másik ilyen negatív tényező lehet számunkra az is, hogy az Amazon Alexa nem használható magyar nyelvvél.

### 1.1.3. Google rendszere - Google Home és Google Nest

Az Amazon Echo sikere után a Google is részesülni akart az okos otthon piacának sikereivel. Az akkor még Nest Labs által készült termékek olyanok voltak mint az alkalmazkodó termosztát - amit 2011-ben hoztak létre „Nest Learning Thermostat”<sup>4</sup> néven, ami Wi-Fi-re kapcsolható volt, és szenzorok segítségével alkalmazkodhatott a beltéri hőmérsékleti körülményekre. Ezt követte a következő termékük, a füst és szén-monoxid érzékelő, aminek a neve „Nest Protect” volt. 2014-ben felvásárolták a Dropcam nevű céget ami biztonsági kamerákat készítettek, amik után a Nest Labs ezt követő terméke a Nest Cam volt 2015-ben. A Nest Labs egyre jobban látszódó sikerének köszönhetően a Google felvásárolta 2014 januárjában. 2018-ig még önállóan működött a Nest, ami után beolvasztották a Google otthoni termék családba, ezzel létre hozva a Google Nest termékcsaládot, ami számos termékkel rendelkezik mostanára: termosztát, ajtócsengő, ajtózár, biztonsági kamera, virtuális asszisztenssel integrált érintőképernyős központi egység.<sup>5</sup>

Az Amazon Echo-hoz hasonló első terméke a Google-nek a Google Home volt, amit 2016 októberében jelentettek be. Ez a cég sajátos virtuális asszisztensével a Google Assistant-tel volt felszerelve, és ugyan úgy lehetett neki utasításokat adni, és kérdéseket feltenni. Azóta már több otthonon belül alkalmazható eszköz elérhető egyenesen a Google Store-ból.<sup>5</sup> Vagy egyéb támogatott harmadik felek által készített ilyen eszközökkel.

Az egyik legnagyobb előnye a Google Home rendszernek az integrációja más Google szolgáltatásokkal, mint például a Google Térkép, a Google Naptár és a Google Fotók. Ez azt jelenti, hogy a felhasználók kéz nélkül is hozzáférhetnek személyes információikhoz és ütemtervükhöz, egyszerűen csak a Google Asszisztensnek kérdést feltéve.

---

<sup>3</sup> Statisztikai adatokat az Earthweb: „How Many People Use Alexa in 2023? (U.S. Amazon Statistics)” cikkjéből [2]

<sup>4</sup> Magyarul: Nest Tanuló Termosztát

<sup>5</sup> További jelenleg elérhető Google Nest családba tartozó termékek: [https://store.google.com/gb/category/connected\\_home](https://store.google.com/gb/category/connected_home) oldalon megtalálható

A Google Home rendszer másik erőssége az, hogy képes felismerni a különböző hangokat, amely lehetővé teszi a személyre szabott válaszokat és információkat minden háztartási tag számára. Ez különösen hasznos lehet több személyes háztartásokban, ahol több ember is használja a rendszert.

Összességében a Google Home rendszer egy erős és rugalmas okos otthoni megoldás, amely számos funkciót és lehetőséget kínál. Az integrációja más Google szolgáltatásokkal és a széles körű, kompatibilis eszközökkel rendelkezése miatt a Google Home rendszer erős versenytársa a okos otthoni piacnak.

Viszont a Google Home-nak is meg vannak azok a hátrányai mint az Amazon rendszerének. Sajnos inkább csak akkor tudjuk kihasználni ennek a rendszernek az előnyét, ha angolul vagy más támogatott nyelven beszélünk vele, amibe nem tartozik bele a magyar.

#### **1.1.4. Xiaomi rendszere - Mi Home**

A Xiaomi 2015 júniusában dobta piacra első okos otthon termékcsoportját, a „Smart Home Kit”-et, amely mozgásérzékelőt, lámpát és kapacitív kapcsolót tartalmazott. Ezeket az eszközöket egy alkalmazás segítségével lehetett vezérelni, ami lehetővé tette a felhasználók számára, hogy testre szabott utasításokat állítsanak be, például hogy a mozgásérzékelő észlelésekor a lámpa felkapcsoljon, vagy értesítést küldjön a felhasználónak, illetve a csomag támogatta a Xiaomi biztonsági kameráját is. [3]

Ma a Xiaomi okos otthon termékpalletája rendkívül sokrétű, a forrólevegős sütőtől, a robotporszívókon, és a szobamérlegeken át a hőmérséklet- és páratartalom-szenzorokig. Az összes termékük integrálható a Mi Home alkalmazásba, és a felhasználók harmadik féltől származó eszközöket is csatlakoztathatnak hozzájuk. A Xiaomi termékeivel akár komplett otthoni biztonsági rendszert is létre lehet hozni.

Nagy előnye a Xiaomi okos otthon termékeknek az, hogy viszonylag olcsóbb a konkurenciától, és fel is használható például a Google Home, és az Amazon Alexa Smart Home rendszeren belül is, és használata felettébb felhasználó barát.

Kiseb hátránya lehet neki az, hogy ezeket az eszközöket nem lehet asztali alkalmazáson keresztül irányítani, csak is az Androidos és iOS-es alkalmazáson keresztül, vagy esetleg a Google Home vagy Alexa Smart Home-on belül. Amazon Alexával a párosítás mostanában sajnos nehezebb, mivel valamilyen oknál fogva nagy százalékban nem tud csatlakozni valami hibánál fogva. Még olyan is megesik, hogy nem minden eszköz érhető el például a Google Home felületen, ilyen lehet példának a biztonsági kamerájuk, ami csak a saját alkalmazásukon érhető el. És mint az előző kettő rendszernél, ez sem használható magyar nyelven.



## 1.2. Nyílt forráskódú rendszerek

### 1.2.1. Mit is az, hogy open source?

Open source, azaz nyílt forráskódú szoftver az olyan, aminek a forráskódját szabadon lehet vizsgálni, módosítani, és akár ki is egészíteni. A kód az a része egy szoftvernek amit a legtöbb felhasználó bizonyára soha nem fog látni. Ez az a kód, amit a programozók változtathatnak hogy megváltozzon a program vagy alkalmazás működése. Azok a programozók, akik hozzáférhetnek egy szoftver forráskódjához, azokat feljeszthetik, és javíthatják azzal hogy például új funkciót adnak hozzá, vagy kijavítanak egy olyan részt, ami nem minden esetben működik helyesen.<sup>6</sup>

Azonban azt, hogy egy szoftver nyílt forráskódú, nem feltétlenül jelenti azt, hogy az adott szoftver ingyenes használatban áll. Nyílt forráskódot fejlesztő programozók kérhetnek pénzt azért a nyílt forrású szoftverért, hogy ha ők alkották meg, vagy hozzájárulásukkal készült el.

Bizonyos esetekben azonban, mivel a nyílt forráskódú licenc megkövetelheti a program forráskódjának kiadását amikor szoftvert adnak el másoknak, egyes programozók úgy találják, hogy jövedelmezőbb pénzt kérni a felhasználóktól a szoftverszolgáltatásokért és - támogatásért (nem feltétlenül magáért a szoftverért). Így a szoftvereik ingyenesek maradnak, és pénzt keresnek azzal, hogy másoknak segítenek telepíteni, használni és hibaelhárítást végezni.<sup>6</sup>

### 1.2.2. Home Assistant

A Home Assistant 2013 novemberére került publikálásra *GitHub.com*-on PAULUS SCHOUTSEN által. Ez ekkor még egy Python programozási nyelven megírt alkalmazás volt. Ekkor még ez egy egyszerű program volt, ami napnyugtakor felkapcsolta a lámpát. Azóta a szoftver elég érett lett, és körülbelül 20 aktív hozzájáruló dolgozik a projekten, ennek köszönhetően kéthetente érkezik frissítés a rendszerre.<sup>7</sup>

Ez a szoftver bármely olyan rendszeren működik, ami a Python 3 programozási nyelvet tudja futtatni. Biztosít mobil és számítógép alkalmazást is, mellyel több ezer támogatott eszközt tudunk irányítani. Annyi a különbség ez a rendszer és a nagyobb cégek által biztosított között, hogy ez teljesen lokálisan fut, és hogy ha esetleg internet kimaradás lenne, még akkor is tudjuk ugyan úgy irányítani eszközeinket. A rendszer által számos ismert okos eszköz támogatott, mint például a Philips Hue, IKEA TRÅDFRI, és akár az Amazon Echo, Google Home, és a Xiaomi Mi Home által támogatott eszközök, és még a virtuális asszisztensük is.<sup>8</sup> Mindez annak köszönhető, mert

---

<sup>6</sup> A nyílt forráskódú szoftverről az irodalomjegyzék [4]. részén található

<sup>7</sup> A történet bővebben az irodalomjegyzék [5]. részén megtalálható.

<sup>8</sup> A támogatott rendszerek a weboldalukon elérhető, ami az irodalomjegyzék [6]. részén megtalálható

Z-Wave és a Zigbee protokoll is támogatott a rendszer által, és mivel nyílt forráskódú a szoftver, ezért bármikor lehet az hogy valamelyik felhasználó vagy esetleg fejlesztő hozzáadja bizonyos termékeket kiegészítésként.

A rendszerrel háztartásunk energiafelhasználását is nyilván lehet tartani. Szabadon testre szabható a felület kinézete a felhasználók ízlése szerint. Szintén pozitívum lehet, hogy az szoftver támogatja a magyar nyelvet is.

Ámbár ez a rendszer azoknak ajánlott jobban, akik informatika terén ismertek.

### 1.2.3. OpenHAB

OpenHAB, azaz Open Home Automation Bus<sup>9</sup> fejlesztése 2010-ben kezdődött, és Java programozási nyelvben lett megírva. 2013-ban került elérhetővé az első stabil verziója a szoftvernek.

Az OpenHAB legelső körben azon a személyek számára ajánlott, akik jól ismertek informatika, és robotika terén. Ez azért szükséges, mert nekünk kell összeállítanunk, hogy mit szeretnénk elérni az okos otthon rendszerünkben. Az alkalmazás rendkívül rugalmas és testre szabható, és nagy a támogatottsága. Ez az egyik legelterjedtebb nyílt forráskódú okos otthon rendszer, és folyamatos fejlesztés alatt áll egy nonprofit szervezet által, aminek a fejlesztésébe bárki beállhat.

Ez a rendszer képes integrálni a piac különböző eszközeit, mint ahogyan azt a Home Assistant rendszernél is említettem. Szintén egy fontos tényezője, hogy lokálisan működik, ezért egy internet kimaradásnál az okos otthonunk ugyan úgy működni fog.

A szoftvernek hivatalos oldalon megtalálható a teljes dokumentációja, hogy mit hogyan kell összeállítani és használni, ezért lehet kedvelt olyan személyek számára, akik szeretik a dolgokat maguknak összeállítani. A rendszer nagy támogatást és közösséget élvez, így ha felmerülne bármilyen kérdés, biztosan választ kapunk rá.

Az ezzel létrehozott otthonunkat a legtöbb felületen el tudjuk érni, legyen az MacOS, Windows, Linux, Android vagy iOS.<sup>10</sup>

---

<sup>9</sup> Magyarul: Nyílt Otthon Automatizációs Busz

<sup>10</sup> Az OpenHAB hivatalos oldala megtalálható a az irodalomjegyzék [7]. részén, ahol olvasható a dokumentációjuk, a céljaik, és egyéb blogjaik.

## 2. fejezet

# Alkalmazott eszközök

Ebben a fejezetben azt fogom taglalni, hogy milyen eszközöket is használtam az okos otthon rendszerem létrehozása során. Legyenek ezek hardveres vagy szoftveres komponensek.

Mindezek közben fogok csatolni kötési rajzokat, amik arra tesznek betekintést, hogy hogyan is kell összekötni ezeket az eszközöket, hogy a rendszerben működjenek.

Szoftveres komponensek közé tartoznak azok a szoftverek, amik a projekt létrehozása közben fel voltak használva: legyen az keretrendszer, programozási nyelv, stílus megírásra használt külső komponens.

### 2.1. Hardver

#### 2.1.1. Mi is az hogy IoT?

IoT<sup>1</sup>-k azok olyan eszközök, melyek Wi-Fi hálózaton keresztül párosíthatóak, és irányíthatóak. Ilyenek lehetnek a háztartásunkban levő olyan eszközök, melyeket távolról is tudunk irányítani interneten keresztül. Legyen ez például a Xiaomi forrólevegős sütője, vagy egy okos izzó.

Ebbe a kategóriába tartoznak a mikrokontrollerek is, amik mini programozható elektronikák, az alkalmazott Raspberry Pi 4B is.

#### 2.1.2. Raspberry Pi 4B

A projektnek szíve-lelke egy bankkártya méretű mini számítógép, ami szolgáltatja a vezetékek nélküli internetet az otthoni eszközök számára, ezzel megvalósítva a később említett programozható elektronikáknak is a kommunikációt.

A Raspberry-n dolgozódnak fel az adatok, és ez által szolgáltatott a webalkalmazás is. Felettébb sokoldalú a használata, és előszeretettel van használva okos otthon pro-

---

<sup>1</sup> (Internet of Things) - magyarul: Internet dolgai

jektek megvalósításában is, például hogy ha Home Assistant-el, vagy OpenHAB-bal szeretnénk azt megvalósítani.

Eben Upton 2012-ben hozta létre az első ilyen kis számítógépet, a Raspberry Pi 1 Model B-t. Raspberry magyarul azt jelenti, hogy málna mivel, ebben az időben többen is hoztak létre elektronikai termékeket gyümölcs nevekkal felruházva: legyen az Apple (alma), Acorn (makk), Apricot (sárgabarack).

Elsődleges célja az volt Eben Upton-nak hogy olcsó és könnyen elérhető számítógépeket juttassanak el fiatalok számára.

A Raspberry számítógépeknek elsődlegesen két fajtája volt: az A Model, ami olcsóbb, és a B Model, ami gyorsabb volt.<sup>2</sup>

Rendkívül sokoldalú ez a kis számítógép, és ezért is ennyire előszeretett a használata a programozók körében.

**Az általam használt Raspberry Pi 4B specifikációi:**

- **RAM:** 2GB
- **Tárhely:** 16GB-os microSD memória kártya
- **Operációs rendszer:** Raspbian, ami egy Debian alapú Linux operációs rendszer

### 2.1.3. NodeMCU - ESP-WROOM-32

A NodeMCU az Espressif Systems ESP8266-12E Wi-Fi System-On-Chip<sup>3</sup> rendszeren alapul, amely egy nyílt forráskódú, Lua-alapú firmware-rel<sup>4</sup> van felszerelve. tökéletes az IoT-alkalmazásokhoz és más olyan helyzetekhez, ahol vezeték nélküli kapcsolatra van szükség. Ez a chip nagyon sok hasonlóságot mutat az Arduino-val – mindkettő mikrokontrollerrel felszerelt prototípus lap, amely az Arduino IDE segítségével programozható.[12]

Az ESP-WROOM-32 mikrovezérlő egy erős és sokoldalú eszköz, amely remek választás lehet egy okos otthon projekt számára. Alacsony energiaigényű processzora és integrált Wi-Fi és Bluetooth képességei alkalmasak szenzorok és eszközök vezérlésére és kommunikációjára.

Az ESP-WROOM-32S mikrovezérlő adatlapja részletes technikai információkat tartalmaz a funkcióiról és képességeiről.[10]

Összességében az ESP-WROOM-32S mikrovezérlő egy erős és rugalmas eszköz az okos otthoni rendszerek építéséhez, és az adatlap széleskörű technikai információkat nyújt a fejlesztők számára, akik ezzel az eszközzel szeretnék tervezni és programozni.

---

<sup>2</sup> Az irodalomjegyzék [8]. részén megtalálható a bővebb története a Raspberry Pi-nak

<sup>3</sup> Rendszer a chipen

<sup>4</sup> Magyarul: Fix tárban tárolt információ, rendszer

Éppen ezért is lehet elsődleges választás robotika oktatás közben egy NodeMCU használata, én is például az egyetemi tanulmányaim során a Robotikai alapjai nevű tárgyon találkoztam ezzel az eszközzel először.

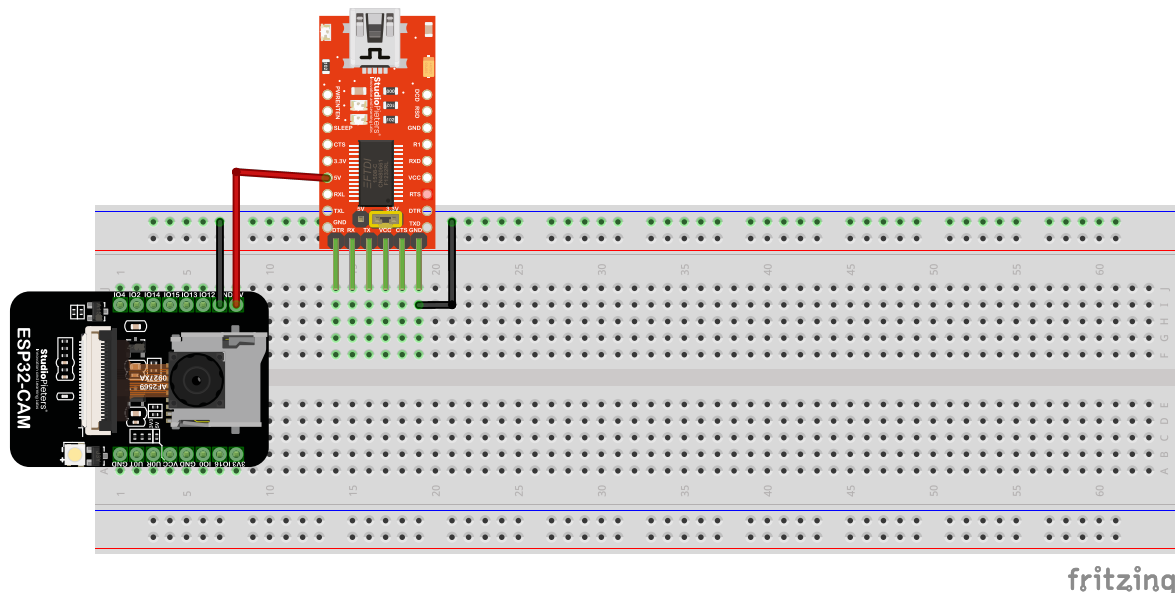
Rendszeremben ezek az eszközök úgy vannak használva, hogy – kihasználva a Wi-Fi-s felszereltségüket – rákapcsolódnak a Raspberry Pi-ra, és ezen keresztül küldenek olyan információkat, mint például az adott szobában lévő hőmérséklet és páratartalom szenzor által szolgáltatott információk, és a szobákban lévő eszközök ki-és bekapcsolása.

#### 2.1.4. ESP32-CAM - Wi-Fi-s kamera modul

Mint ahogyan a 2.1.3. szakaszban taglalt NodeMCU, az ESP32-CAM is az IoT és ESP32-es eszközök családjába tartozik, ami egy olyan programozható elektronika ami Wi-Fi-re képes csatlakoztatni.

Ebben a rendszerben olyan módon van alkalmazva, mint egy „biztonsági kamera”, és a főoldalon szolgálat élő közvetítést. Szintén oly módon csatlakozik rá a Raspberry Pi-ra, mint a többi mikrokontroller ebben a rendszerben.<sup>5</sup>

Lehetőség adott ezzel az eszközzel az is, hogy akár memóriakártyára mentünk képeket, amit a kamera felvesz, és van egy saját beépített LED lámpája is. Azonban ezeket a funkciókat a rendszer aktuális verziójában nem alkalmazom, csak és kizárólag élő közvetítésre.



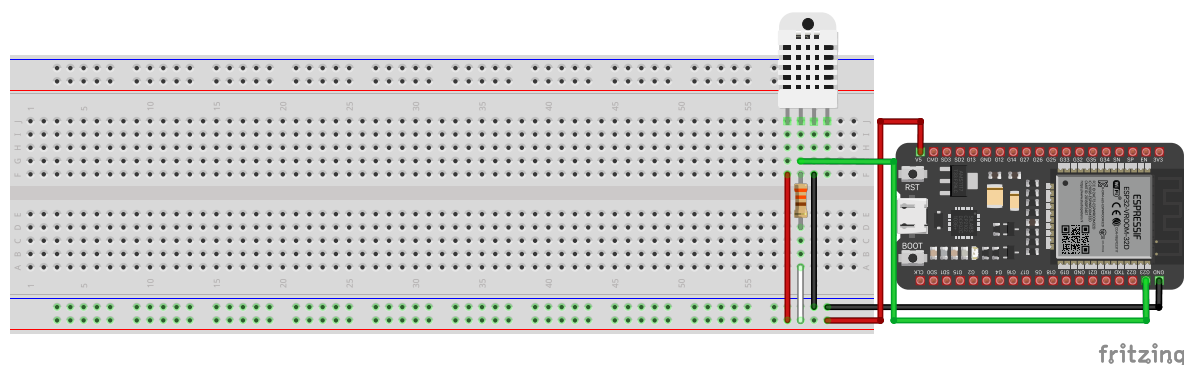
2.1. ábra. Az ESP32-CAM bekötése

<sup>5</sup> A 3.4 részben ennek a működése részletezve lesz.

### 2.1.5. DHT22 - hőmérséklet és páratartalom szenzor

Az DHT22 érzékelő egy olyan eszköz, amely méri a levegő hőmérsékletét és a relatív páratartalmát. Az összegyűjtött adatokat fejlett digitális technológiával dolgozza fel és továbbítja, ami megbízható és stabil működést biztosít. Minden érzékelőt gondosan kalibrálnak egy speciális kamrában annak érdekében, hogy pontos eredményeket kapjunk, és a kalibrációs információk a készülék memóriájában tárolódnak. Az DHT22 kis mérete, alacsony energiafogyasztása és hosszú adatátviteli távolsága miatt széles körben használható különböző alkalmazásokban.[13]

Ebben a rendszerben úgy van alkalmazva, hogy egy ESP-WROOM-32-höz van kapcsolva, és ez 10 másodpercenként elküldi az adatokat a webalkalmazásnak, mely eltárolja az adott pillanatban mért adatokat az alkalmazott szobában, és jeleníti meg a főoldalon, mely alapján meg lehet tekinteni az elmúlt 24 órai adatokat egy grafikonon is.<sup>5</sup>



2.2. ábra. Az ESP-WROOM-32 bekötése DHT22 hőmérséklet és páratartalom szenzorral

### 2.1.6. Mi is az RFID?

A Radio Frequency Identification<sup>6</sup> (RFID) olyan vezeték nélküli rendszer, amely két összetevőből áll: tag-ekből<sup>7</sup> és olvasókból. Az olvasó olyan eszköz, amely egy vagy több antennával rendelkezik, amelyek rádióhullámokat bocsátanak ki, és jeleket fogadnak vissza az RFID-tag-ről. A tag-ek, amelyek rádióhullámokat használnak személyazonosságuk és egyéb információk közlésére a közeli olvasókkal, lehetnek passzívak vagy aktívak. A passzív RFID-címkéket az olvasó táplálja, és nincs akkumulátoruk. Az aktív RFID címkék akkumulátorral, vagy hálózatról működnek.

Az RFID-címkék számos információt tárolhatnak egy sorozatszámtól több oldalnyi adatig. Az olvasók lehetnek mozgathatóak, így kézben is hordhatók, vagy oszlopra vagy

<sup>6</sup> Magyarul: rádiófrekvenciás azonosítás

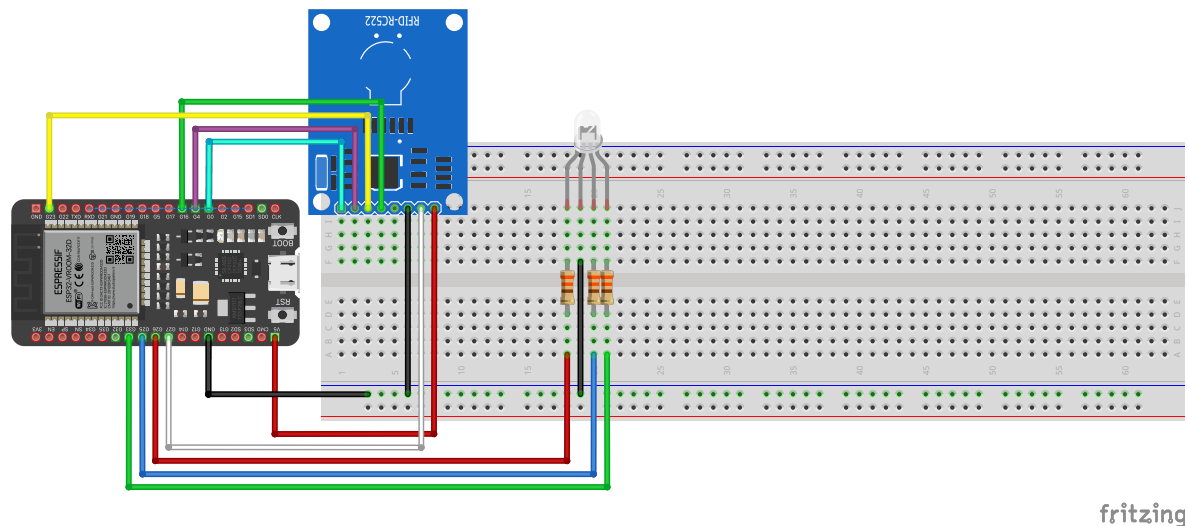
<sup>7</sup> Magyarul: címkékből

fej fölött rögzíthetők. Az olvasórendszerek egy szekrény, szoba vagy épület architektúrájába is beépíthetők.[14]

### 2.1.7. RFID-RC522 - RFID olvasó

Mint ahogyan a 2.1.6 . szakaszban említve lett, ez egy RFID olvasó, amely képes RFID tag-eknek az azonosítójukat, és egyéb adatait beolvasni rádióhullámok alkalmazásával.

A rendszerben úgy van jelen, mint egy kezdetleges „biztonsági rendszer”, amit például ajtóknak, vagy szekrényeknek zárására lehetne használni. Jelen esetben úgy van alkalmazva, hogy az RFID olvasót működtető ESP-WROOM-32 elküldi a Raspberry Pi számára az adott tag adatait: hogy, ha ez megegyezik az adatbázisban jelen levő azonosító egyikével, akkor – jelenlegi rendszerben – az RGB LED<sup>8</sup> zölden kezd villogni, ellenkező esetben pedig pirosan fog villogni.<sup>5</sup>



2.3. ábra. Az ESP-WROOM-32 bekötése, RFID-RC522 olvasóval és egy RGB LED-del

### 2.1.8. Szilárdtest relé

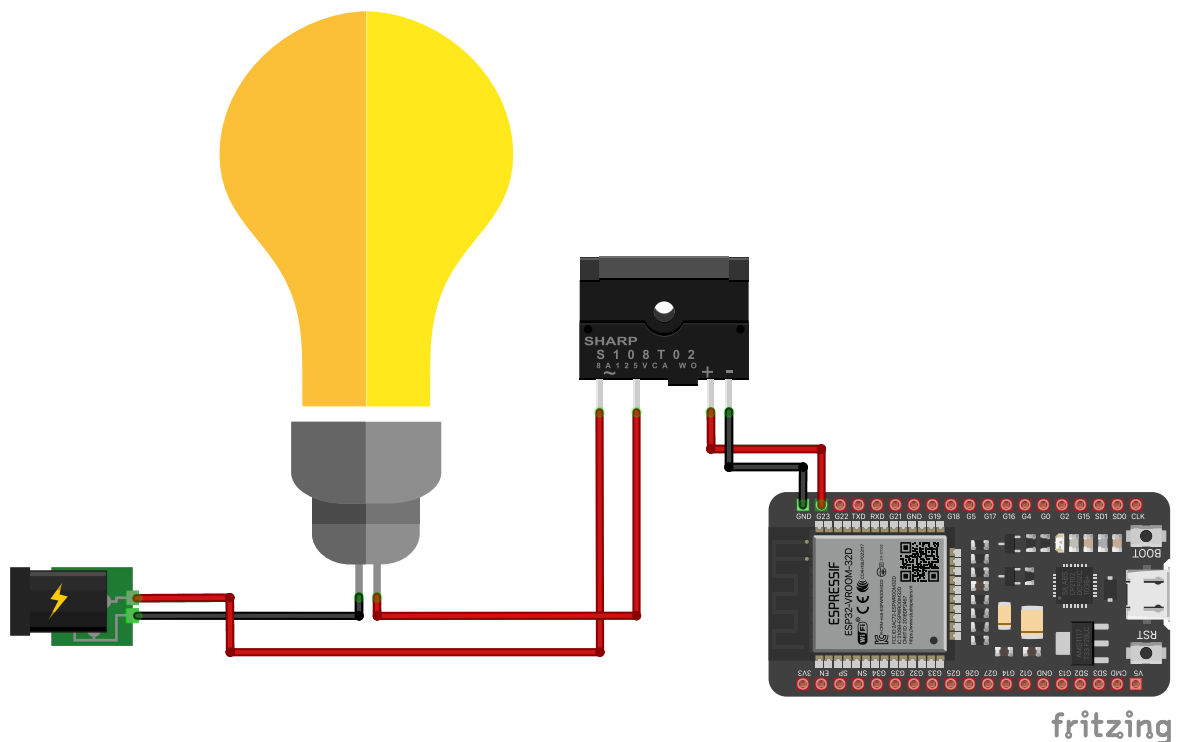
A szilárdtest relé az egy elektronikus kapcsolóeszköz. Az elektromechanikus reléhez hasonlóan képes be- vagy kikapcsolni a terhelést, ha külső vezérlőjelet adnak át a vezérlőkapcsolón. A szilárdtest relék azonban nem tartalmaznak mozgó alkatrészeket, ezért nem hangosak és hosszabb életűek az elektromechanikus reléhez képest.

A félvezetők elektromos és optikai tulajdonságait használják fel a kapcsolás végrehajtására, és teljes leválasztást biztosítanak a vezérlőáramkör és a terhelési áramkör között.[16]

<sup>8</sup> Red Green Blue Light Emitting Diode, magyarul Piros Zöld Kék Fényt Kibocsátó Dióda

Ebben a rendszerben úgy lett alkalmazva ez, hogy egy ESP-WROOM-32 és egy izzó van csatlakoztatva egy szilárdtest reléhez. Az ESP-WROOM-32 rákapcsolódva a Raspberry Pi-ra képes azt a felhasználó a felületen keresztül ki- vagy bekapcsolni.<sup>5</sup>

De akár számos más olyan eszközt lehet ezzel alkalmazni, amit ki- vagy be lehetne kapcsolni: mint például egy ventilátor.



2.4. ábra. Az ESP-WROOM-32 bekötése szolidtest-relével és egy izzóval

## 2.2. Szoftverek és Programozási nyelvek

### 2.2.1. C++ és Arduino IDE

A Bjarne Stroustrup és a Bell Laboratories csapata által 1979-ben kifejlesztett C++ egy általános célú programozási nyelv, és ami lényegében a C programozási nyelv továbbfejlesztett változata. A főbb kiegészítések közé tartozott az objektumorientált programozás, és a hiba- és végrehajtás kezelés.[17]

A mikrokontrollerek legtöbbjét C++-al szokták programozni, amihez az Arduino IDE az egyik legnépszerűbb választás, ami számos kiegészítővel, és mikrokontrollerek programozásához használatos funkciókkal rendelkezik, legyen ez a kód feltöltése, vagy a soros monitorra kiírt információk kiírása. Külső könyvtárak is hozzáadhatóak a különböző eszközök támogatása érdekében.



### 2.2.2. HTML és CSS

Hyper Text Markup Language,<sup>9</sup> azaz HTML egy leírónyelv, amely szövegek leírására szolgál. Számos funkciója van ennek, mint a bekezdések, táblázatok, vagy képek beszúrása. Ezt legtöbbször weboldalak elkészítésénél használják. Mezőket „<>”-en belül határozzuk meg és lezárni ehhez hasonlóan „</>”-el kell. Mindezt annak megfelelően kell párosítani, hogy mit is írunk: legyen ez például egy bekezdés, ami „<p><sup>10</sup> </p>” aminek a két tag közé kell beírni a szöveget.

Mindez kevés arra, hogy a szöveg kinézetét változtassuk, ezért is társítják melléje a CSS-t, azaz Cascading Style Sheet-et<sup>11</sup>, ami HTML vagy XML nyelvek által meghatározott megjelenés formázására szolgál. Ezeket a stílus leírásokat „<style></style>” tag-ek közé szoktuk leírni. Megvan ennek is a sajátos meghatározási módja, hogy mit és hogyan alkalmazunk.

A webalkalmazásnál máshogyan állítom be a kinézetet, amit a 2.2.7 részben említék, hogy hogyan is működik a Tailwind.

### 2.2.3. PHP

A PHP, azaz PHP: Hypertext Preprocessor<sup>12</sup> egy általános felhasználású szkriptnyelv, és fordító, amely szabadon elérhető és széles körben használt webfejlesztésnél. A nyelvet elsősorban szerveroldali szkriptek készítésére használják, bár használható parancssori szkriptekre is, és korlátozott mértékben alkalmazásokra.[18]

C programozási nyelv jellegű a szintaxisa, és rendkívül engedékeny programozási nyelv. Amikor PHP-ban akarunk valamit írni, akkor azt .php-re végződő fájlban írjuk, aminek a belső tartalma „<?php”-vel kezdődik, és „?>”-vel végződik. Ezek között írható le a kódunk. Legtöbb esetben HTML-el van társítva.

### 2.2.4. JavaScript

JavaScript egy objektum orientált programozási nyelv, elsősorban interaktív elemek létrehozására egy weboldalon és alkalmazásokon. Kezdetekben ez csak böngészőkön és weboldalakon volt használva, azonban ez áttért sok más környezetre és használati területre.

A JavaScript előnyeikhez tartoznak, hogy reszponzív és interaktív elemeket tudunk létrehozni, a megtanulása és futtatása felettébb egyszerű, és rendkívül híres és gyakran használt programozási nyelv.[19]

---

<sup>9</sup> Magyarul: Hiper Szöveg Leíró Nyelv

<sup>10</sup> azért „p” a bekezdés, mert angolul ez a szó úgy van, hogy paragraph

<sup>11</sup> Magyarul: Kaszkádolt Stílus Lap

<sup>12</sup> Magyarul: PHP: hiperszöveg előfeldolgozó

A webalkalmazásomban is a felületen sok helyen JavaScript-et, azon belül is a jQuery-t használom funkciók kezelésére, legyen az például egy lámpa felkapcsolásának kezdeményezése.<sup>5</sup>

### 2.2.5. jQuery

A jQuery egy gyors, kisméretű és funkciókban gazdag JavaScript könyvtár. Sokkal egyszerűbbé teszi az olyan dolgokat, mint a HTML-dokumentumok bejárása és kezelése, az eseménykezelés, és az animáció egy könnyen használható API-val, amely számos böngészőben működik. A sokoldalúság és a bővíthetőség kombinációjával a jQuery emberek millióinak JavaScript-írási módját változtatta meg. [20]

### 2.2.6. Chart.js

Chart.js egy népszerű, nyílt forráskódú JavaScript könyvtár, amely lehetővé teszi a fejlesztők számára, hogy könnyen létrehozzanak reszponzív és testre szabható diagramokat vagy grafikonokat a weben. Többféle diagram típust támogat.

A könyvtár a HTML5-ös vászonra épül, amely lehetővé teszi a gördülékeny animációkat és az interaktivitást. A Chart.js egyszerű és intuitív API-t kínál, amely lehetővé teszi a fejlesztők számára a diagramok egyszerű beállítását és stílusának testreszabását. Több beépített testreszabási lehetőséget is biztosít, mint például a diagramszínek, címkék, jelmagyarázatok, és eszköztipp.[21]

Chart.js nagy előnye, hogy a hivatalos oldalukon részletesen le van dokumentálva, hogy hogyan kell telepíteni, és mit hogyan is lehet használni, példákon keresztül.

A Chart.js a egy adott szobában lévő hőmérséklet és páratartalom szenzor adatok grafikonon való megjelenítésére volt használva a weboldalon.

### 2.2.7. Tailwind CSS

A Tailwind CSS egy népszerű CSS keretrendszer, amely előre definiált osztályokat biztosít a webalkalmazások stílusának egyszerűsítéséhez és optimalizálásához. A keretrendszer funkcionális megközelítést alkalmaz a CSS stílusok alkalmazásában, amelyek osztályok formájában vannak meghatározva, és leírják a kívánt vizuális hatást, ahelyett, hogy egyedi CSS stílusokat írnának az egyes elemekre.

Például egy bekezdéshez így tudjuk azt megadni, hogy a szövege jobbra zárt és félkövér legyen: `<p class="text-right font-bold"13> ez egy bekezdés </p>`

A Tailwind stílus alkalmazása lehetővé teszi a fejlesztők számára a gyors, összehangolt és egységes megjelenések létrehozását, miközben javítja a kód alakíthatóságát és skálázhatóságát, azonban ezzel rontva a HTML fájl átláthatóságát. Széles választékban

---

<sup>13</sup> text-right, azaz szöveg-jobb, font-bold, azaz szöveg-félkövér

kínál előre elkészített osztályokat a közös UI<sup>14</sup> komponensekhez, mint például gombok, űrlapok, grid rendszerek és tipográfiák, valamint a bonyolultabb elrendezésekhez és pozicionálási feladatokhoz is. Fejlesztést pedig hivatalos dokumentációval segítik, mivel mindenre van nekik példa és leírás.[22]

Az egyik fő előnye a Tailwindnek a rugalmasság és testreszabhatóság. Konfigurációs fájl biztosít a fejlesztőknek, amely lehetővé teszi a beépített osztályok testreszabását és saját egyedi osztályok létrehozását az alkalmazás egészében való használatra.

### 2.2.8. dbDiagram.io

A dbDiagram.io szoftverkomponens a projekt adatbázis séma tervezéséhez és megjelenítéséhez volt használva. A dbdiagram.io egy webes eszköz, amely lehetővé teszi a felhasználók számára az adatbázis séma létrehozását egy intuitív vizuális felületen keresztül.

Az eszköz támogatja különböző adatbázis rendszereket, mint például a MySQL-t, a PostgreSQL-t és a MongoDB-t. Szintén hasznos funkciókat kínál, mint például automatikus SQL kódgenerálás, verziókezelés és a diagramok importálása és exportálása különböző formátumokban.[23]



Szakdolgozatom során ez volt használva arra, hogy elkészítsem az adatbázis tervét, és majd a 3.2 szakaszon lesz róla egy mellékelt kép is, hogy hogyan is lett mindez kialakítva.

### 2.2.9. Font Awesome

A Font Awesome egy elég népszerű nyílt forráskódú ikon könyvtár, amely skálázható vektor ikonokat kínál<sup>15</sup>, amelyek testreszabhatóak és stílusosak CSS segítségével. A könyvtár több ezer ikont tartalmaz, amelyek széles körű kategóriákat fednek le, például webalkalmazásokat, márká- és közösségi média ikonokat. Lehetőség van fizetős és ingyenes ikonok használatára is.

A Font Awesome egy rugalmas és könnyen használható megoldást kínál az ikonok hozzáadásához weboldalakhoz vagy alkalmazásokhoz, lehetővé téve a fejlesztőknek, hogy javítsák projektjeik kinézetét. A könyvtár folyamatosan frissül új ikonokkal és funkciókkal.[24]









**Az általam használt ikonok, a rendszerben:**

-  „<i class=“fa-solid fa-arrow-left”></i>”
-  „<i class=“fa-solid fa-clock-rotate-left”></i>”

---

<sup>14</sup>User Interface: azaz Felhasználói Felület

<sup>15</sup>Ez azt jelenti, hogy egy algoritmus segítségével mindig ugyan olyan minőségű lesz a kép, bármekkora is nagyítjuk.

-  „*<i class="fa-solid fa-plus"></i>*”
-  „*<i class="fa-regular fa-pen-to-square"></i>*”
-  „*<i class="fa-solid fa-trash-can"></i>*”
-  „*<i class="fa-solid fa-rotate-right"></i>*”
-  „*<i class="fa-solid fa-ban"></i>*”
-  „*<i class="fa-solid fa-gear"></i>*”
-  „*<i class="fa-solid fa-key"></i>*”
-  „*<i class="fa-solid fa-id-card-clip"></i>*”

### 2.2.10. MySQL

A MySQL egy nyílt forráskódú adatbázis-kezelő rendszer, amelyet széles körben használnak webalkalmazásokhoz. Lehetővé teszi a felhasználók számára, hogy relációs adatbázisokat hozzanak létre, kezeljenek és karbantartsanak az SQL használatával.

A MySQL megbízhatóságáról, és hatékonyságáról ismert, és sok nagy webhely és alkalmazás használja. Több platformmal és programozási nyelvvel is kompatibilis, ami rugalmas választást kínál az adatkezeléshez.

A XAMPP egy webszerver szoftver, amely tartalmazza a MySQL adatbázis-kezelő rendszert, és a PHP-t is, és telepíthető a helyi számítógépre webfejlesztéshez és teszteléshez, mint ahogyan én is alkalmaztam lokális fejlesztés során.

### 2.2.11. PlantUML

A PlantUML egy ingyenes és nyílt forráskódú eszköz, amely egyszerű szövegszintaxist használ a különböző formátumokban létrehozott UML<sup>16</sup> diagramok készítéséhez. Többféle diagramtípus támogatása mellett használható szoftverfejlesztéshez és rendszertervezéshez.

Ennek segítségével rajzoltam le azt, hogy NodeMCU és a Raspberry Pi hogyan is kommunikál a rendszeren belül.<sup>5</sup>

### 2.2.12. Fritzing

A Fritzing az egy nyílt forráskódú szoftver, aminek a segítségével elektronikai eszközök kötési és sematikus rajzait, de akár szimulációt is lehet benne készíteni.

---

<sup>16</sup> Unified Modeling Language, magyarul Egységesített Modellező Nyelv

Ez egy olyan program, aminek a letöltéséért cserébe fizetni kell, ezzel is támogatva a program fejlesztését és karbantartását. Letölthetőek hozzá külső könyvtárak, ezzel is növelve a lehetőségeket ilyen rajzok létrehozásában.

Ennek használatával hoztam létre a 2.1 .szakaszban a kötési rajzokat.

### **Külső könyvtárak, amik voltak használva a rajzokban:**

- A 2.1. ábrán az ESP-WROOM-32, az ESP32-CAM, és az FTDI Adapter[25]
- A 2.2. ábrán az ESP-WROOM-32 és a DHT22 szenzor [25]
- A 2.3. ábrán az ESP-WROOM-32[25] és az RFID-RC522[28]
- A 2.4. ábrán az ESP-WROOM-32[25] az izzó[26] és a szilárdtest relé[27]

## **2.3. Laravel**

A Laravel egy ingyenes és nyílt forrású PHP webalkalmazás-keretrendszer, amelyet skálázható és nagy méretű webalkalmazások építésére használnak. Egy elegáns szintaxist, erős eszközöket és moduláris csomagolási rendszert biztosít, hogy a fejlesztők tiszta és karbantartható kódot írassanak. A Laravel követi az MVC<sup>17</sup> architektúrális mintát, és beépített funkciókkal rendelkezik, mint például az azonosítás, a route-olás és gyorsítótár.[29] A fejlesztők dolgát azzal könnyítik meg leginkább, hogy hivatalos dokumentációja van, ahol minden kis részletre adnak példát és leírást.[30]

### **MVC Keretrendszer:**

Az MVC az egy szoftvertervezési minta, amelyet a felhasználói felületek fejlesztéséhez használnak.

### **Az alkalmazást három összekapcsolt komponensre bontja:**

A **modell**, amely az adatokat és a logikát képviseli, a **nézet**, amely megjeleníti az adatokat a felhasználónak, és a **vezérlő**, amely kezeli a felhasználói bemenetet és kapcsolatban áll a modellel és a nézettel. Az MVC keretrendszerek, mint például a Laravel, strukturált megközelítést biztosítanak a webalkalmazások fejlesztéséhez, és segítik a fejlesztőket a moduláris és karbantartható kód írásában.

Éppen ezért is esett a választásom a projekt tervezési fázisában a Laravel keretrendszerre. Ez a webalkalmazás még Laravel 9-nél indult el.

---

<sup>17</sup> Model-View-Controller: magyarul Modell-Nézet-Vezérlő

## 3. fejezet

# A web alkalmazás felépítése és működése

3.1. Raspberry PI4B alkalmazása, mint Wi-Fi, és hub

3.2. Adatbázis felépítése

3.3. Kezelő felület bemutatása és működése

3.3.1. Főoldal

3.3.2. Beállítások

3.3.3. RFID beállítások

3.3.4. RFID használati táblázat

3.3.5. Hőmérsékleti és páratartalom előzmények - ChartJS

3.4. Eszközök kommunikációja a webszerverrel

3.4.1. Hőmérséklet és Páratartalom szenzor

3.4.2. Eszköz kapcsoló

3.4.3. Kamera

3.4.4. RFID kártya olvasó

## 4. fejezet

# Tesztelések

### 4.1. Tesztelések módjai és fontossága

#### 4.1.1. Cypress automatizált tesztelések

#### 4.1.2. Manuális tesztelések

<b>Teszt leírása</b>	<b>Elvárt eredmények</b>	<b>Tapasztalatok</b>
Access Control Test: RFID tags assigned to users	The system should recognize the RFID tags assigned to each user and grant or deny access to specific devices or areas of the home accordingly	The test was successful, with the system accurately recognizing and responding to each user's assigned RFID tag
Device Control Test: Using RFID tags to control devices	The system should allow users to control smart home devices (such as lights or locks) using RFID tags, without requiring additional input	The test was partially successful, with the system accurately recognizing the RFID tags but experiencing some delays in device response times
Durability Test: RFID tags in high traffic areas	The RFID tags should remain functional and readable even in high traffic areas, with no degradation in performance	The test was successful, with the RFID tags remaining fully functional and readable even with heavy usage
Security Test: Unauthorized access prevention	The system should be designed to prevent unauthorized access by detecting and alerting the user to the presence of unrecognized RFID tags	The test was successful, with the system detecting and preventing access by an unrecognized RFID tag, and sending an alert to the user's mobile device

4.1. táblázat. Manuális tesztek a webalkalmazásra



## 5. fejezet

# Rendszer telepítése

# Összegzés

# Irodalomjegyzék

- [1] JEFF BLANKENBURG: *Define Your Appliance Category for a Better Customer Experience*

URL: <https://developer.amazon.com/blogs/alexa/post/a89f7243-08a0-4c73-8fc5-eb604a93f437/define-your-appliance-category-for-a-better-customer-experience>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [2] JASON WISE: *How Many People Use Alexa in 2023? (U.S. Amazon Statistics)*

URL: <https://earthweb.com/alexa-users/>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [3] HANDWIKI: *Engineering:Xiaomi Smart Home* URL: [https://handwiki.org/wiki/Engineering:Xiaomi\\_Smart\\_Home](https://handwiki.org/wiki/Engineering:Xiaomi_Smart_Home)

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [4] WHAT IS OPEN SOURCE

URL: <https://opensource.com/resources/what-open-source>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [5] ERIC BROWN: *Home Assistant: The Python Approach to Home Automation*

URL: <https://www.linux.com/topic/embedded-iot/home-assistant-python-approach-home-automation/>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [6] HOME ASSISTANT - INTEGRATIONS

URL: <https://www.home-assistant.io/integrations/>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [7] OPENHAB HIVATALOS OLDALA

URL: <https://www.openhab.org/>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [8] THE EPIC STORY OF THE RASPBERRY PI

URL: <https://raspberrypi.com/raspberry-pi-history/>

LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [9] OFFICAL RASPBERRY PI 4B DOKUMENTATION  
URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-4>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [10] ESPRESSIF SYSTEMS: *ESP32-WROOM-32 Datasheet*  
URL: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [11] OFFICAL ESP32 DEVICES DOCUMENTATION  
URL: <http://esp32.net/#Info>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [12] ANAT ZAIT: *NODEMCU - A PERFECT BOARD FOR IOT*  
URL: <https://www.circuito.io/blog/nodemcu-esp8266/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [13] AOSONG ELECTRONICS CO.,LTD: *Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302)*  
URL: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [14] U.S. FOOD AND DRUG ADMINISTRATION: *Radio Frequency Identification (RFID)*  
URL: <https://www.fda.gov/radiation-emitting-products/electromagnetic-compatibility-emc/radio-frequency-identification-rfid>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [15] HANDSON TECHNOLOGY: *RC522 RFID Development Kit*  
URL: <http://www.handsontec.com/dataspecs/RC522.pdf>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [16] JAYESH UPADHYAY: *How do Solid State Relays work?*  
URL: <https://www.circuitbread.com/ee-faq/how-do-solid-state-relays-work>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [17] ROMAIN JUILLET: *Essentials of Programming in C++*  
URL: <https://www.bocasay.com/essentials-programming-c/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [18] ROBERT SHELDON: *DEFINITION: PHP (Hypertext Preprocessor)*  
URL: <https://www.techtarget.com/whatis/definition/PHP-Hypertext-Preprocessor>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [19] SUSAN SIMKINS: *Quick Start to JavaScript: Volume 1*  
URL: <https://www.pluralsight.com/courses/quick-start-javascript-1-1870>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [20] JQUERY  
URL: <https://jquery.com/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [21] CHART.JS HIVATALOS DOKUMENTÁCIÓ OLDALA  
URL: <https://www.chartjs.org/docs/latest/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [22] TAILWIND CSS HIVATALOS DOKUMENTÁCIÓ OLDALA  
URL: <https://tailwindcss.com/docs>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [23] DBDIAGRAM.IO OLDALA  
URL: <https://dbdiagram.io/home>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [24] FONTAWESOME - FREE ICONS  
URL: <https://fontawesome.com/search?o=r&m=free>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [25] ACHIM PIETERS *Fritzing – New Parts*  
URL: <https://www.studiopieters.nl/fritzing/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [26] ALFRED DAGENAIS *Fritzing Components - Light bulb*  
URL: <https://github.com/alfreddagenais/fritzing-components/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [27] FRITZING FORUM *SSR-40va solid state relay*  
URL: <https://forum.fritzing.org/t/ssr-40va-solid-state-relay/16832>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

- [28] AMONTANES *RFID-RC522*  
URL: <https://fritzing.org/projects/mfrc522>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [29] LARAVEL - OVERVIEW  
URL: [https://www.tutorialspoint.com/laravel/laravel\\_overview.htm](https://www.tutorialspoint.com/laravel/laravel_overview.htm)  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [30] OFFICAL LARAVEL DOCUMENTATION  
URL: <https://laravel.com/docs/10.x>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [31] CONNECT RFID TO PHP & MYSQL DATABASE WITH NODEMCU ESP8266  
URL: <https://iotprojectsideas.com/connect-rfid-to-php-mysql-database-with-node>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [32] ESP8266 NODEMCU HTTP GET AND HTTP POST WITH ARDUINO IDE  
(JSON, URL ENCODED, TEXT)  
URL: [https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/  
#http-get-1](https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/#http-get-1)  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [33] ARDUINO TO LARAVEL COMMUNICATION  
URL: <https://www.instructables.com/Arduino-to-Laravel-Communication/>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [34] RASPBERRY CONFIGURATION FOR WI-FI  
URL: [https://www.raspberrypi.com/documentation//computers/  
configuration.html#setting-up-a-routed-wireless-access-point](https://www.raspberrypi.com/documentation//computers/configuration.html#setting-up-a-routed-wireless-access-point)  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.
- [35] OFFICAL CYPRESS DOCUMENTATION  
URL: <https://docs.cypress.io/guides/overview/why-cypress>  
LINK UTOLJÁRA ELLENŐRIZVE: 2023.03.31.

# Nyilatkozat

Alulírott *Bagoly Gábor*, büntetőjogi felelősségem tudatában kijelentem, hogy az általam benyújtott, *Programozható elektronikák alkalmazásai* című szakdolgozat önálló szellemi termékem. Amennyiben mások munkáját felhasználtam, azokra megfelelően hivatkozom, beleértve a nyomtatott és az internetes forrásokat is.

Aláírással igazolom, hogy az elektronikusan feltöltött és a papíralapú szakdolgozatom formai és tartalmi szempontból mindenben megegyezik.

Eger, 2023. április 2.

aláírás