Általános információk

A diplomaterv szerkezete:

1. Diplomaterv feladatkiírás
2. Címoldal
3. Tartalomjegyzék
4. A diplomatervező nyilatkozata az önálló munkáról és az elektronikus adatok kezeléséről
5. Tartalmi összefoglaló magyarul és angolul
6. Bevezetés: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása
7. A feladatkiírás pontosítása és részletes elemzése
8. Előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések
9. A tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása
10. A megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek
11. Esetleges köszönetnyilvánítások
12. Részletesés pontos irodalomjegyzék
13. Függelék(ek)

Felhasználható a következő oldaltól kezdődő Diplomaterv sablon dokumentum tartalma. Ügyeljen a konzulens nevét és a beadás évét jelölő szövegdobozokra, mert azokra külön ki kell adni a frissítést. A mezők tartalma a sablonban a dokumentum adatlapja alapján automatikusan kerül kitöltésre.

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az oldalak tükörmargóval készüljenek (mindenhol 2,5 cm, baloldalon 1 cm-es kötéssel). Az alapértelmezett betűkészlet a 12 pontos Times New Roman, másfeles sorközzel.

Minden oldalon – az első négy szerkezeti elem kivételével – szerepelnie kell az oldalszámnak.

A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetenként decimális számmal és kifejező címmel kell ellátni. A fejezeteket decimális aláosztással számozzuk, maximálisan 3 aláosztás mélységben (pl. 2.3.4.1.). Az ábrákat, táblázatokat és képleteket célszerű fejezetenként külön számozni (pl. 2.4. ábra, 4.2 táblázat vagy képletnél (3.2)). A fejezetcímeket igazítsuk balra, a normál szövegnél viszont használjunk sorkiegyenlítést. Az ábrákat, táblázatokat és a hozzájuk tartozó címet igazítsuk középre. A cím a jelölt rész alatt helyezkedjen el.

A képeket lehetőleg rajzoló programmal készítsék el, az egyenleteket egyenlet-szerkesztő segítségével írják le.

Az irodalomjegyzék szövegközi hivatkozása történhet a Harvard-rendszerben (a szerző és az évszám megadásával) vagy sorszámozva. A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címeit azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben. Minden publikáció a szerzők után a következő adatok szerepelnek: folyóirat cikkeknél a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóirat címeket csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internet hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal tulajdonosát és tartalmát (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Fontos:

* a szakdolgozat készítő/diplomatervező nyilatkozata (a jelen sablonban szereplő szövegtartalommal) kötelező előírás Karunkon, ennek hiányában a szakdolgozat/diplomaterv nem bírálható és nem védhető!
* mind a dolgozat, mind a melléklet maximálisan 15 MB méretű lehet!

Jó munkát, sikeres szakdolgozat készítést, ill. diplomatervezést kívánunk!

FeladatkiÍrás

TODO: pontosan mi idén a policy? Elvileg a tanszékvezető utólag aláírja.

A feladatkiírást a tanszék saját előírása szerint vagy a tanszéki adminisztrációban lehet átvenni, és a tanszéki pecséttel ellátott, a tanszékvezető által aláírt lapot kell belefűzni a leadott munkába, vagy a tanszékvezető által elektronikusan jóváhagyott feladatkiírást kell a Diplomaterv Portálról letölteni és a leadott munkába belefűzni (ezen oldal HELYETT, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell megismételni a feladatkiírást.

C:\Users\szarnyasg\Downloads\bme_logo_nagy.eps

**Budapesti Műszaki és Gazdaságtudományi Egyetem**

Villamosmérnöki és Informatikai Kar

Készítette

Papp Dominik Edvárd Konzulens

Attila

2023

Tartalomjegyzék

[Összefoglaló 7](#_Toc149816916)

[Abstract 8](#_Toc149816917)

[1. Bevezetés 9](#_Toc149816918)

[1.1. Motivációm 9](#_Toc149816919)

[1.2. A feladat rövid értelmezése 9](#_Toc149816920)

[1.3. Felhasználhatóság 10](#_Toc149816921)

[1.4. A szakdolgozat felépítésének rövid összefoglalása 10](#_Toc149816922)

[2. Rendszer architektúra 11](#_Toc149816923)

[2.1. Adatgyűjtő 11](#_Toc149816924)

[2.2. Szerver 11](#_Toc149816925)

[2.3. Adatbázis 11](#_Toc149816926)

[2.4. Web/Browser 12](#_Toc149816927)

[3. Alap alkotóelemek 13](#_Toc149816928)

[3.1. Programozási nyelvek 13](#_Toc149816929)

[3.1.1. C nyelv 13](#_Toc149816930)

[3.1.2. C++ nyelv 13](#_Toc149816931)

[3.1.3. Python nyelv 13](#_Toc149816932)

[3.1.4. Webfejlesztési nyelvek 13](#_Toc149816933)

[3.1.5. Webfejlesztési keretrendszerek 14](#_Toc149816934)

[3.1.6. FreeRTOS 15](#_Toc149816935)

[3.1.7. MySQL 16](#_Toc149816936)

[3.2. Verziókezelés 16](#_Toc149816937)

[3.3. Fejlesztőkörnyezetek, fejlesztőeszközök 16](#_Toc149816938)

[3.3.1. Visual Studio Code (VS Code) 16](#_Toc149816939)

[3.3.2. Node.js 17](#_Toc149816940)

[3.3.3. STM32CubeIDE 17](#_Toc149816941)

[3.3.4. TouchGFX 17](#_Toc149816942)

[3.3.5. MySQL Workbench 17](#_Toc149816943)

[3.3.6. CoolTerm 18](#_Toc149816944)

[3.3.7. Esptool.py 18](#_Toc149816945)

[3.4. Hardver 18](#_Toc149816946)

[3.4.1. STM32H750B-DK 18](#_Toc149816947)

[3.4.2. ESP8266-01 19](#_Toc149816948)

[3.4.3. SHT31 SEN0385 Szenzor 20](#_Toc149816949)

[3.4.4. Számítógépek 21](#_Toc149816950)

[3.5. Szoftver (szoftveres alkotóelemek) 21](#_Toc149816951)

[3.5.1. Mikrokontroller szoftvere 22](#_Toc149816952)

[3.5.2. Adatbázis 22](#_Toc149816953)

[3.5.3. Server 22](#_Toc149816954)

[3.5.4. Weboldal 22](#_Toc149816955)

[4. Komponensek 23](#_Toc149816956)

[4.1. Adatgyűjtő 23](#_Toc149816957)

[4.1.1. ESP 23](#_Toc149816958)

[4.1.2. Sensor 23](#_Toc149816959)

[4.1.3. Szoftver 23](#_Toc149816960)

[4.2. Server 23](#_Toc149816961)

[4.3. Adatbázis 23](#_Toc149816962)

[4.4. Weblap 23](#_Toc149816963)

[5. Komponensek közvetlen kapcsolatai 24](#_Toc149816964)

[5.1. Adatgyűjtő és szerver 24](#_Toc149816965)

[5.2. Szerver és weblap 24](#_Toc149816966)

[6. Komponensek közvetett kapcsolatai 25](#_Toc149816967)

[6.1. Adatgyűjtő és adatbázis 25](#_Toc149816968)

[6.2. Adatgyűjtő és weblap 25](#_Toc149816969)

[6.3. Adatbázis és weblab 25](#_Toc149816970)

[7. Értékelés 26](#_Toc149816971)

[7.1. Elemzés 26](#_Toc149816972)

[7.2. Továbbfejleszthetőség 26](#_Toc149816973)

[Köszönetnyilvánítás 27](#_Toc149816974)

[Irodalomjegyzék 28](#_Toc149816975)

[Függelék 30](#_Toc149816976)

Hallgatói nyilatkozat

Alulírott Papp Dominik Edvárd, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2023. 11. 03.

Papp Dominik Edvárd

# Összefoglaló

A szakdolgozat, vagy diplomaterv elkészítése minden egyetemi hallgató életében egy fontos mérföldkő. Lehetőséget ad arra, hogy az egyetemi évei során megtanultakat kamatoztassa és eredményeit szélesebb közönség előtt bemutassa, s mérnöki rátermettségét bizonyítsa. Fontos azonban, hogy a dolgozat elkészítésének folyamata számos csapdát is rejt magában. Rossz időgazdálkodás, hiányos szövegszerkesztési ismeretek, illetve a dolgozat készítéséhez nélkülözhetetlen „műfaji” szabályok ismeretének hiánya könnyen oda vezethetnek, hogy egy egyébként jelentős időbefektetéssel készült kiemelkedő szoftver is csak gyengébb minősítést kapjon a gyenge minőségű dolgozat miatt.

E dokumentum – amellett, hogy egy általános szerkesztési keretet ad a dolgozatodnak – összefoglalja a szakdolgozat/diplomaterv írás írott és íratlan szabályait. Összeszedjük a Word kezelésének legfontosabb részeit (címsorok, ábrák, irodalomjegyzék stb.), a dolgozat felépítésének általános tartalmi és szerkezeti irányelveit. Bár mindenkire igazítható sablon természetesen nem létezik, megadjuk azokat az általános arányokat, oldalszámokat, amelyek betartásával jó eséllyel készíthetsz egy színvonalas dolgozatot. A részletes és pontokba szedett elvárás-lista nem csupán a dolgozat írásakor, de akár más dolgozatok értékelésekor is kiváló támpontként szolgálhat.

Az itt átadott ismeretek és szemléletmód nem csupán az aktuális feladatod leküzdésében segíthet, de hosszútávon is számos praktikus fogással bővítheti a szövegszerkesztési és dokumentumkészítési eszköztáradat.

# Abstract

English translation of the abstract of the thesis work. This summarises the content of the thesis in 0.5–1 pages and is uploaded to the Thesis Work Portal as well.

# Bevezetés

## Motivációm

A motivációm a feladatra többrétű. A specializációm a beágyazott rendszerek világába vezetett be. Szerettem volna a webfejlesztés és az internet világával megismerkedni mindamellett, hogy a megszerzett beágyazott rendszerekhez kapcsolódó tudásomat felhasználom, fejlesztem és kamatoztatom. Szerettem volna továbbá számomra új, korszerű programnyelvekkel és technológiákkal megismerkedni. Célom volt az is, hogy ergonomikus felhasználói élményt tudjak biztosítani, kreativitásomat fel tudjam használni.

A fejemben ezeket a feltételeket összegezvén és az engem érdeklő ipari területekkel összevetve, az olvasó által továbbiakban olvasott project ötlete fogant meg bennem.

## A feladat rövid értelmezése

A szakdolgozat célja egy olyan mérnöki alkotás elkészítése, amely képes egy távoli, a külvilággal fizikai összeköttetésekkel nem rendelkező szenzor állomás (adatgyűjtő) adatait helyben (weboldalon) megjeleníteni.

Ehhez négy egység szükséges:

1. Egy adatgyűjtő állomás, amely képes nem fizikai kapcsolattal adatokat gyűjteni majd azokat továbbítani, és egy kijelző segítségével felhasználói utasításokat végrehajtani.
2. Egy szerver, amely képes az adatgyűjtő állomás adatait kezelni, valamint képes a weboldal felé adatokat, illetve webfileokat továbbítani.
3. Egy weboldal, amely segítségével megtekinthetjük az adatokat.
4. Egy adatbázis, amely képes az adatgyűjtő adatait és más segítő célú adatokat tárolni

Ennek a négy egységnek a logikai kapcsolatait az 1-es ábra szemlélteti.



ábra

## Felhasználhatóság

A szakdolgozatom munkáját egy alapváznak tekintem, amelyet számos területen fel lehet használni kisebb-nagyobb továbbfejlesztésekkel. A munka összköltségétől és az adott környezetben való gazdasági megfizethetőségétől most eltekintek, pusztán az ideális felhasználhatósága a szempont.

Felhasználható:

1. Okosotthonokban helyiségek szobahőmérsékletének, illetve egyéb tulajdonságainak a mérése és továbbítása a központi számítógép felé. Ez előnyös lehet olyan esetekben, ahol fizikai UDP kapcsolat létrehozása már túl költséges opció volna.
2. Kültéri mérőállomásokon gyűjtött adatok továbbítására távoli központokba.
3. Folyamatok elemzéséhez szükséges adatok összegyűjtésére és továbbítására.

Tulajdonképpen minden olyan helyzetben felhasználható, ahol az adatokra nem (csak) helyben van szükség, hanem egy távoli ponton (is).

## A szakdolgozat felépítésének rövid összefoglalása

A szakdolgozat olvasása során, a továbbiakkal fogunk megismerkedni, sorban:

1. A Rendszer átfogó architektúrájával. Itt bemutatásra kerül az alkotás működése átfogólag, a legabsztraktabb szinten.
2. Az alkotáshoz felhasznált diszkrét alap alkotóelemekkel, beleértve a fejlesztőkörnyezeteket és segédszoftvereket.
3. Az alap alkotóelemekből alkotott komponensekkel, az 1-es ábrán bemutatott négy egységgel.
4. Az előbb említett komponensek közötti közvetlen kapcsolatokkal. Pl.: hogyan kommunikál az adatgyűjtő állomás a szerverrel.
5. A közvetlen kapcsolattal nem rendelkező komponensek közötti közvetett, logikai kapcsolatokkal
6. A szakdolgozat lezárásával, bemutatásával, értékelésével.

# Rendszer architektúra

A rendszer architektúrája első ránézésre bonyolultnak tűnhet, azonban kellően logikusan tagolt és több, könnyen kezelhető részegységre bontható.

Ebben a fejezetben be szeretném mutatni a rendszert a lehető legnagyobb absztrakcióval élve, ugyanis részletes tárgyalására számos későbbi fejezetben sor kerül.



ábra

## Adatgyűjtő

Az adatgyűjtő egy szenzor-mikrokontroller-ESP hármasból áll. Feladata az adat gyűjtése és továbbítása egy TCP szerver felé. Ez a kommunikáció egyirányú.

## Szerver

A szerver egy TCP szerverből és egy Web Backendből áll. A kettő között kapcsolat nincs.

A TCP szerver adatokat fogad és elküldi egy adatbázis felé mentési kéréssel.

A Backend szolgál webserverként, azaz ki tud szolgálni egy weboldalt a megfelelő fileokkal és szolgál API endpointként is. A API endpoint képes adatokat menteni és lekérdezni egy adatbázistól.

## Adatbázis

Az adatbázis képes több kapcsolaton keresztül kéréseket kiszolgálni.

## Web/Browser

A webes felület képes jelezni szükségét adatok kérése és mentése iránt, valamint azokat megjeleníteni formázottan a felhasználó képernyőjén.

# Alap alkotóelemek

Ahogy az előző fejezet 2-es ábráján láthattuk, a project meglehetősen komplex. Emiatt sok és változatos technológiai eszközöket igényel a megvalósítása. A felhasznált technológiákat ebben a fejezetben szeretném bemutatni az olvasónak.

## Programozási nyelvek

A projectben több programozási nyelv is felhasználásra került. Ebben az alfejezetben szeretném összegyűjteni ezeknek az összességét és röviden bemutatni azokat.

### C nyelv

A [C programnyelv](https://en.wikipedia.org/wiki/C_(programming_language))[1] (TODO: ha végleges mindenhova [] rakni) minden elektronikus eszköz programozásának egyik alappillére. Jellegzetessége, hogy rendkívül hardverközeli programozást tesz lehetővé és mégis könnyebben kezelhető, mint az Assembly programozás. A C nyelvet sok területen használják, például operációs rendszerek fejlesztésében, beágyazott rendszerek programozásában és számos más alkalmazásfejlesztési területen.

### C++ nyelv

A [C++](https://en.wikipedia.org/wiki/C%2B%2B) egy kiterjesztése és fejlesztése a [C programozási nyelvnek](#_C_nyelv). Ebbe a nyelvbe az objektumorientált programozás elvei is beépültek. A C++ lehetővé teszi a programozók számára, hogy adatokat és funkciókat csoportosítsanak osztályokba, ami nagyobb szervezettséget és újrafelhasználhatóságot eredményez. Az objektum orientáltság egy kritikus összetevője a jól skálázható, komplex szoftvereknek.

A C++-nak sok alkalmazási területe van, például szoftvertervezés, játékfejlesztés, valós idejű rendszerek tervezése és egyéb alkalmazások fejlesztése.

### Python nyelv

A [Python](https://www.python.org/) egy magas szintű, általános célú programozási nyelv, melyet Guido van Rossum kezdett el fejleszteni az 1980-as években és azóta rendkívül népszerűvé vált a programozói közösségben. A Python két fő vonzereje az olvashatóság és a könnyen tanulhatóság.

A Python széles körű alkalmazhatósággal rendelkezik és sokféle területen használják, beleértve a webfejlesztést (pl. Django keretrendszerrel), az adatelemzést (pl. NumPy és Pandas könyvtárakkal), a gépi tanulást (pl. TensorFlow és Scikit-learn), az automatizálást és még sok más területet. A nyelvnek erős és aktív fejlesztői közössége van, ami folyamatosan bővíti a Python ökoszisztémáját, és számos hasznos könyvtárat és eszközt kínál a programozók számára.

A Python a folyamatos fejlesztések miatt kényes a verzió típusra. Kényszerű megemlítenem, hogy a projectben mindenhol a Python 3.11.3-as verziója van használva!

### Webfejlesztési nyelvek

#### HTML

A [HTML](https://en.wikipedia.org/wiki/HTML) (HyperText Markup Language) egy szöveg alapú jelölőnyelv a weboldalak készítéséhez. Az alapötlete az, hogy az oldal tartalmát, struktúráját és formázását címkékkel határozzuk meg. Ezek a címkék olyan elemeket reprezentálnak, mint fejlécek, bekezdések, képek, linkek és egyebek. Ezek a címkékhez, elemekhez attribútumokat is hozzárendelhetünk, amelyek további információkat szolgáltatnak az elemhez.

A nyelv kódja böngészők által értelmezett és megjelenített, így a felhasználók láthatják a formázott weboldalt. A HTML az alapja a weboldalaknak, és más technológiákkal, például [CSS](#_CSS) (Cascading Style Sheets) és [JavaScript](#_JavaScript) segítségével személyre szabhatjuk az oldal stílusát és funkcionalitását. Ez a nyelv az első lépés a webfejlesztésben, és a tartalom valamint a struktúra meghatározására szolgál a weben.

#### CSS

A [CSS](https://en.wikipedia.org/wiki/CSS) (Cascading Style Sheets) egy fontos eszköz a webfejlesztésben, amely segít a weboldalak kinézetének meghatározásában. Az alapötlete az, hogy különválasszuk a tartalmat ([HTML](#_HTML)) és a stílust (CSS). CSS-szabályokban meghatározhatjuk a színeket, betűtípusokat, az elrendezést, és más stilisztikai tulajdonságokat. Ezek a szabályokat elhelyezhetjük külön fájlokban vagy közvetlenül a HTML-dokumentumban is.

A CSS alkalmazása nemcsak esztétikus megjelenést eredményez, hanem lehetővé teszi a responszív tervezést is. Ez azt jelenti, hogy az oldalak alkalmazkodnak különböző kijelzőméretekhez és eszközökhöz, például asztali számítógépekhez, táblagépekhez és mobiltelefonokhoz. A CSS segítségével egységes megjelenést érhetünk el a weboldalakon, valamint könnyen karbantartható és skálázható stílusokat hozhatunk létre. Ezek által a CSS elengedhetetlen eszköz a modern webfejlesztésben, amely segít a felhasználók számára vonzó és jól működő weboldalak készítésében.

#### JavaScript

A [JavaScript](https://en.wikipedia.org/wiki/JavaScript) egy weboldalak interaktivitásának és dinamizmusának létrehozására használt programnyelv. A JavaScript a böngészők által támogatott, tehát a felhasználók eszközein fut anélkül, hogy külön szoftvert kellene telepíteni. Ez lehetővé teszi a weboldalak dinamikus, lokális frissítését és az interaktivitás bevezetését. A nyelvnek számos beépített funkciója van, amelyek lehetővé teszik a [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) (Document Object Model) manipulálását, eseménykezelést, AJAX hívásokat és még sok minden mást.

A JavaScript lehetővé teszi webalkalmazások fejlesztését, beleértve a játékokat, űrlapkezelést, adatkezelést, animációkat és sok más funkciót. Gyakran használják más technológiákkal, például [HTML](#_HTML) és [CSS](#_CSS), hogy teljes körű webalkalmazásokat hozzanak létre. A nyelv ökoszisztémája gazdag és számos keretrendszer, könyvtár és eszköz elérhető a hatékonyabb fejlesztés érdekében. Az egyszerűsége és sokoldalúsága miatt a JavaScript ma az egyik legfontosabb nyelv a webfejlesztés területén.

### Webfejlesztési keretrendszerek

A webfejlesztési keretrendszerek jelentős szerepet játszanak a modern webalkalmazások fejlesztésében. Ezek az eszközök elősegítik a hatékony és gyors fejlesztést, lehetővé teszik a könnyű újrafelhasználást és javítják a projekt skálázhatóságát. Nélkülük egy mai korszerű weblap belátható időn belüli elkészítése szinte elképzelhetetlen. A keretrendszerek szabványosított és bevált megoldásokat kínálnak a gyakori webfejlesztési kihívásokra, mint például az adatkezelés, a felhasználói felület tervezése, a biztonság és a teljesítmény optimalizálása.

#### React és JSX

A [React](https://react.dev/) egy népszerű [JavaScript](#_JavaScript) könyvtár a felhasználói felületek ([UI](https://en.wikipedia.org/wiki/User_interface), User Interface) építésére. A JSX egy kiegészítő szintaxis, amely lehetővé teszi a React komponensek deklaratív leírását. A React és a JSX segítségével webalkalmazásokat készíthetünk, amelyek dinamikusan reagálnak a felhasználói interakciókra. A React komponensek egységességet és újrafelhasználhatóságot biztosítanak, valamint könnyen integrálhatók más projektekbe. A React az egyik legnépszerűbb front-end keretrendszer, amelyet sok nagy vállalat és fejlesztő használ a modern webalkalmazások fejlesztéséhez.

#### Bootstrap

[Bootstrap](https://getbootstrap.com/) egy nyílt forráskódú [CSS](#_CSS) keretrendszer, amely segít az egységes és reszponzív webdizájn kialakításában. Bootstrap tartalmaz előre elkészített stílusokat, komponenseket és [JavaScript](#_JavaScript) funkciókat, amelyek felgyorsítják a weboldalak tervezését és fejlesztését. A Bootstrap alapvetően lehetővé teszi a fejlesztők számára, hogy gyorsan és egyszerűen hozzanak létre modern, mobilbarát weboldalakat anélkül, hogy mélyebb CSS vagy JavaScript tudással rendelkeznének. Bootstrap-et gyakran használják front-end fejlesztés során és egyesítik a React vagy más keretrendszerekkel, hogy még hatékonyabbá tegyék a webalkalmazásaik fejlesztését

#### FontAwesome

A [FontAwesome](https://fontawesome.com/) egy népszerű ikonkészlet és webes ikonfont, amelyet a webfejlesztés során gyakran használnak ikonok beillesztéséhez weboldalakba és webalkalmazásokba. Az ikonok különböző kategóriákban érhetők el, és segítségükkel könnyen hozzáadhatók olyan vizuális elemek, amelyek javítják a weboldalak és alkalmazások felhasználói felületét.

### FreeRTOS

A [FreeRTOS](https://www.freertos.org/) egy nyílt forráskódú valós idejű operációs rendszer (RTOS, Real Time Operating System), amelyet beágyazott rendszerekben használnak. Az RTOS olyan operációs rendszer, amely biztosítja a programok számára a valós idejű ütemezést, ami azt jelenti, hogy a rendszer képes szigorú időkorlátok mellett végrehajtani a feladatokat és az eseményeket.

A FreeRTOS hatékony és kis memóriaigényű, így ideális beágyazott rendszerekhez, ahol a rendelkezésre álló memória és erőforrások korlátozottak. Az RTOS célja az alkalmazások strukturált szervezése és a prioritások kezelése, amelyek révén biztosítja, hogy a legfontosabb feladatok elsőbbséget élveznek. Emellett a FreeRTOS-t számos processzorra és mikrovezérlőre lehet „telepíteni”.

A FreeRTOS-hoz gazdag fejlesztői támogatás és dokumentáció is tartozik, amelyek segítik a hatékony alkalmazások fejlesztését. Az egyszerűsége és a valós idejű ütemezés képességei miatt a FreeRTOS a beágyazott rendszerek tervezésében és fejlesztésében népszerű megoldás.

A FreeRTOS-nak több verziója is elérhető. A szakdolgozatomban a [CMSIS-RTOS2](https://arm-software.github.io/CMSIS_5/latest/RTOS2/html/index.html) verzióját használtam.

### MySQL

A [MySQL](https://www.mysql.com/) egy népszerű nyílt forráskódú relációs adatbázis-kezelő rendszer és programozási nyelv, amelyet széles körben használnak az alkalmazások adatbázis-kezelésére. A MySQL teljes körű adatbázis-kezelési megoldás, amely támogatja a relációs adatbázisok tervezését és az SQL (Structured Query Language) használatát. Lehetővé teszi a felhasználók, jogosultságok és szerepkörök definiálását a biztonságos adathozzáféréshez és az adatintegritás megőrzéséhez.

A MySQL kiváló teljesítményt és skálázhatóságot nyújt, valamint támogatja az indexeket, a tranzakciókat és a replikációt. Összességében a MySQL a világ egyik legelterjedtebb adatbázis-kezelő rendszere, és számos alkalmazásban használják adatainak hatékony kezelésére és tárolására.

## Verziókezelés

A [Git](https://git-scm.com/) egy terminálból vezérelhető verziókezelő rendszer, amelyet fejlesztők és csapatok használnak szoftverfejlesztési projektek során az együttműködésre és a kódváltoztatások nyomon követésére egyaránt. A Git rendkívül hatékony és alkalmas a kisebb-nagyobb projektekhez egyaránt. A Git-nek elérthető webes felülete is, a [GitHub](https://github.com/).

A fájlok változtatásai csak akkor kerülnek mentésre és verzióba (etalon kódbázisba), amikor a fejlesztő explicit módon megadja. Ez lehetővé teszi a precíz és tiszta verziókezelést, valamint a könnyű visszatérést korábbi verziókhoz.

A szakdolgozatom számos szoftver fejlesztéséből állt, így elengedhetetlen eszköz volt a Git. Mivel több számítógépet is használtam, így a számítógépek között is könnyen tudtam hordozni a kódjaimat.

## Fejlesztőkörnyezetek, fejlesztőeszközök

A project több, különböző programnyelv használatát tette szükségessé, amelyekkel való fejlesztéshez más-más fejlesztőkörnyezetek alkalmatosak. Bizonyos folyamatok megkönnyítésére, feladatok ellátására, illetve programnyelvek felhasználásához fejlesztőeszközökre van szükség. Ezeket a fejlesztőkörnyezeteket és fejlesztőeszközöket gyűjtöttem össze és írtam róluk röviden ebben az alfejezetben.

### Visual Studio Code (VS Code)

A [Visual Studio Code](https://code.visualstudio.com/) egy Microsoft által fejlesztett ingyenesen elérhető, nyílt forráskódú fejlesztőkörnyezet. A VS Code könnyen testre szabható, bővíthető, és számos kiegészítővel rendelkezik, amelyek segítik a kódszerkesztést és a projektfejlesztést. Ez egy népszerű választás a web- és szoftverfejlesztéshez, és számos programozási nyelvet és keretrendszert támogat.

### Node.js

A [Node.js](https://nodejs.org/en) egy szerveroldali [JavaScript](#_JavaScript) környezet, amely lehetővé teszi a JavaScript programok futtatását a szervereken és a háttérben. A Node.js a Chrome V8 JavaScript motoron alapszik (Chrome böngésző JavaScript engine-je), és lehetővé teszi a JavaScript kód végrehajtását szerveroldalon is, nem csak a böngészőben. Ez egy nyílt forráskódú platform, amely széles körben használt különböző alkalmazások és webes szolgáltatások fejlesztésében.

(TODO átrakni webkomponenshez, vagy vegyem ki?)

Node.js segítségével lehet [React](#_React_és_JSX) projectet indítani, amelyhez szervert is indít a localhost-on a könnyebb fejlesztés érdekében. Továbbá lehet vele [Bootstrap](#_Bootstrap) és [FontAwesome](#_FontAwesome) könyvtárakat hozzáadni a React projecthez.

Példakódként, a terminálból a következő képpen lehet kezelni:

npm i -g create-react-app // React telepítő telepítése

npm create-react-app app // app nevű React project létrehozása

cd app // app nevű projectkönyvtárba való belépés

npm start // React applikáció elindítása (webservert is indít)

npm i bootstrap // Bootstrap installálása a projectbe

npm i fontawesome // Fontawesome installálása projectbe

### STM32CubeIDE

Az [STM32CubeIDE](https://www.st.com/en/development-tools/stm32cubeide.html) egy integrált fejlesztőkörnyezet, amelyet a [STMicroelectronics](https://www.st.com/) fejlesztett ki az STM32 mikrovezérlőkön alapuló alkalmazások fejlesztéséhez. A szoftver tartalmaz eszközöket a kódírás, a debuggolás és az eszközkezelés támogatására. Az STM32CubeIDE használatával a fejlesztők egyszerűen hozhatnak létre és tesztelhetnek beágyazott rendszereket az STM32 platformon. Támogatja a C, C++ kód fejlesztését, valamint elérhető a FreeRTOS telepíthető szoftverkomponensként az STM32H750B-DK fejlesztőkártyára.

### TouchGFX

A [TouchGFX](https://www.st.com/en/development-tools/touchgfxdesigner.html) egy STMicroelectronics által fejlesztett nyílt forráskódú, GUI (Graphical User Interface) keretrendszer olyan alkalmazásokhoz, amelyek érintőképernyővel működnek. A TouchGFX segítségével könnyen tervezhetők és fejleszthetők az alkalmazások grafikus felületei. A keretrendszer támogatja a modern, vonzó és reszponzív felhasználói élmény kialakítását különböző beágyazott eszközökön.

A szoftver teljes funkcionalitása a dolgozat írásakor még csak Windows operációs rendszeren volt elérhető, macOS X és Linux nem volt támogatott.

### MySQL Workbench

A [MySQL Workbench](https://www.mysql.com/products/workbench/) egy népszerű [MySQL](#_MySQL) adatbáziskezelő kliensalkalmazás, amely lehetővé teszi az adatbázisokhoz való kapcsolódást, a lekérdezéseket és az adatmanipulációt. A MySQL Workbench segítségével a felhasználók könnyen létrehozhatnak, módosíthatnak és törölhetnek adatbázisokat, valamint futtathatnak lekérdezéseket a táblák és rekordok manipulálására. A MySQL Workbench beépített funkciókat kínál a tranzakciók kezeléséhez, az adatbázisok tervezéséhez és az adatmodell létrehozásához is.

### CoolTerm

A [CoolTerm](http://freeware.the-meiers.org/) egy egyszerű soros kommunikációs szoftver, egy [PuTTY](https://www.putty.org/) alternatíva. A CoolTerm lehetővé teszi a soros porton keresztüli adatkapcsolatot, adatátvitelt és kommunikációt különböző eszközökkel, például mikrovezérlőkkel vagy egyéb hardverrel. A PuTTY nem elérhető macOS rendszereken és a CoolTerm egy kompetens alternatívának bizonyult.

### Esptool.py

Az [esptool.py](https://docs.espressif.com/projects/esptool/en/latest/esp32/) egy parancssori eszköz az Espressif [ESP8266](#_ESP8266-01) és ESP32 WiFi mikrovezérlők programozásához és firmware frissítéséhez. A fejlesztők az esptool.py segítségével tudnak kommunikálni az ESP-eszközökkel, és feltölteni vagy lecserélni a firmware-t. A szoftver gyakran használják a beágyazott IoT (Internet of Things) eszközök fejlesztése során.

## Hardver

A project megvalósításához fizikai eszközökre is szükség van. Ebben az alfejezetben a felhasznált fizikai elemeket mutatom be.

### STM32H750B-DK

Az STMicroelectronics által gyártott [STM32H750B-DK](https://www.st.com/en/evaluation-tools/stm32h750b-dk.html#overview) (Discovery Kit) egy fejlesztői kártya, amelyet a STM32 mikrovezérlőkkel történő alkalmazásfejlesztéshez terveztek. A kártya számos funkciót, hardvert, perifériát, szoftvert és beépített debuggert kínál mind a komplex, mind a változatos igényű projectek fejlesztéséhez.

A close up of a circuit board

Description automatically generated

ábra

#### Mikrovezérlő

A kártya középpontjában az STM32H750 mikrovezérlő található, amely egy erős és sokoldalú ARM Cortex-M7 32 bites processzormaggal rendelkezik. Az órajel sebessége a szoftveres beállítástól függ, de akár 480 MHz-ig is terjedhet. Ez a mikrovezérlő lehetővé teszi a magas teljesítmény igényű alkalmazások futtatását.

#### Kijelző

Az STM32H750B-DK egy színes, 480 x 272 pixeles felbontású LCD érintőképernyőt is tartalmaz, amely lehetővé teszi a grafikus felhasználói felületek kialakítását és tesztelését. Ennek a fejlesztéséhez az STMicroelectronics biztosít egy [TouchGFX](#_TouchGFX) nevű szoftvert.

#### Csatlakozások

A kártya számos beépített csatlakozóval rendelkezik, ideértve az USB, Ethernet és Audio interfészeket. Ez megkönnyíti a külső eszközökkel való kommunikációt.

#### Szenzorok

Az STM32H750B-DK szenzorokat is tartalmaz, például gyorsulásmérőt és giroszkópot, amelyek segítségével érzékelhetjük a kártya mozgását.

#### Bővítési lehetőségek

A kártya számos bővítőhelyet kínál, például Arduino-illesztőknek megfelelő csatlakozókat, illetve egy úgynevezett „[Fanout board](https://www.st.com/resource/en/user_manual/um2695-stmod-fanout-expansion-board-for-stm32-discovery-kits-and-evaluation-boards-stmicroelectronics.pdf)” is tartozéka. A Fanout Boardon többek között található: UART, SCL, I2C interface és ESP-01 foglalat. A Board fejlesztői kártyához való csatlakoztatása nem úgy történik ahogy elsőre gondolná az ember, ugyanis nem csak az intuícióval megy szembe a módja, de a gyártó által biztosított képeken is rosszul van feltüntetve. Az [1-es ábrán](#ábra1) látható csatlakozási mód nem alkalmas. Vagy a kártyát, vagy a Fanout Boardot meg kell fordítani.

### ESP8266-01

#### Ismertető

Az [ESP8266-01](https://nurdspace.nl/ESP8266) egy kis méretű, olcsó, de erős Wi-Fi modul, amelyet a Espressif Systems fejlesztett ki. Mivel képes kezelni a TCP/IP protokollt és apró méretű, nagyon népszerűvé vált az elektronikai és az IoT közösségben. Az ESP8266-01 számos alkalmazást támogat, például a szenzorok vezérlését, adatok továbbítását vagy akár távoli vezérlést Wi-Fi hálózatokon keresztül. A modul 8 kivezetéssel rendelkezik és UART protokollt használ kommunikációra 115200-as baudratettel. A memória mérete változó. Az általam felhasznált eszközé 1MB.

#### Bekötése

A bekötése a következő:

PIN1: Föld, PIN2: UART\_Rx (ESP ezen a lábon küld), PIN3: 3V3, PIN4: 3V3, PIN5: 3V3, PIN6: 3V3, PIN7: UART\_Tx (ESP ezen a lábon fogad), PIN8: 3V3

A diagram of a circuit board

Description automatically generated

ábra

Ez a bekötés használati üzemmódba helyezi a modult. Programozási üzemmódhoz más bekötés szükséges.

#### Programozás

A modul programozása eltérő huzalozást igényel. Programozható üzemmódba bootoláshoz a PIN5 lábat földre kell kötni mielőtt tápot kapna és csak utána biztosítani a feszültséget. Ebben az üzemmódban [esptool.py](#_Esptool.py) illetve Arduino IDE segítségével programozható. (TODO: áthelyezni a továbbiakat a komponens leíráshoz, mivel itt csak az alapfunkciók jerülnek bemutatásra, vagy vegyem ki?) A szakdolgozatom során az esptool.py-t használtam és [AT-Commandokat](https://docs.espressif.com/projects/esp-at/en/release-v2.2.0.0_esp8266/index.html) támogató firmwaret töltöttem rá.

A feltöltendő firmware [innen](https://github.com/CytronTechnologies/esp-at-binaries/blob/main/Cytron_ESP-01S_AT_Firmware_V2.2.0.bin) ingyenesen letölthető. Ugyan elérhetőek több parancsot támogató firmwarek, azok nem megfelelőek. Az általam felhasznált ESP memória mérete 1MB, azok pedig nagyobb helyigényűek. Meglepően nagy kihívást jelentett egy alkalmas méretű firmware megtalálása.

Ha a modul csatlakoztatva van a számítógéphez programozói üzemmódban és elérhető az esptool.py, akkor a következő paranccsal tölthető fel a firmware:

esptool.py --chip auto --port /dev/tty.usbserialname --baud 115200 --before default\_reset --after hard\_reset write\_flash -z --flash\_mode dio --flash\_freq 40m --flash\_size 1MB 0x0 #PathToBinFile/Cytron\_ESP-01S\_AT\_Firmware\_V2.2.0.bin

,ahol „/dev/tty.usbserialname” a csatlakoztatott programozó neve (cmd paranccsal való meghatározás Mac-en: ls /dev/tty.\*) és „#PathToBinFile” a letöltött .bin file abszolút elérési útvonala. Ezek után nem programozói üzemmódban (PIN5: 3V3) való rebootolás után AT Commandok segítségével vezérelhető is a modul.

### SHT31 SEN0385 Szenzor

A szenzor hőmérséklet és relatív páratartalom mérésére egyaránt alkalmas. Kellően pontos és a mérési tartományai is tágak. Pontos adatok az [adatlapjában](https://dfimg.dfrobot.com/nobody/wiki/88b31350da4f54d00989c74c6fa392f7.pdf), valamint emészthetőbb bemutató leírás a [Dfrobot WIKI](https://wiki.dfrobot.com/SHT31_Temperature_Humidity_Sensor_Weatherproof_SKU_SEN0385) oldalán található meg.

A szenzorral I2C interfacen lehet kommunikálni legfeljebb 1MHz SCK órajel sebességgel. 3,3V – 5V feszültséggel optimálisan működtethető. C nyelvű drivert konzulensem, Scherer Balázs biztosított hozzá.

A close-up of a cable

Description automatically generated

ábra

### Számítógépek

A szoftverek fejlesztéséhez két számítógépet használtam. A fejlesztés döntő része macOS Ventura és macOS Sonoma operációs rendszereken történt, ezeken azonban a [TouchGFX](#_TouchGFX) nem volt támogatva. A TouchGFX-szel történő fejlesztés Windows 10-en, egy külön számítógépen esett meg.

## Szoftver (szoftveres alkotóelemek)

Az előző [Hardver](#_Hardver) fejezetben bemutatott elemek kezeléséhez, illetve a project egyéb komponenseinek a megvalósításához különböző szoftverek megírására volt szükség. Ebben a fejezetben az általam írt szoftverek vannak összegyűjtve és átfogó funkcióik, felhasznált technológiáik által röviden bemutatva.

Csak a szoftverek célja, milyen problémát milyen szoftver old meg milyen technológiákkal. Részletes bemutatás később.

### Adatbázis

Az adatbázis egy [MySQL](#_MySQL) adatbázis és a [MySQL Workbench](#_MySQL_Workbench) segítségével könnyen manipulálhatóak az adattáblák. Kapcsolat is létesíthető az adatbázissal, ha az adott programnyelven megvan a megfelelő könyvtár hozzá.

Az adatbázis parancsokkal is vezérelhető, úgynevezett [*query*](https://www.techtarget.com/searchdatamanagement/definition/query#:~:text=A%20database%20query%20is%20either,other%20forms%20of%20data%20manipulation.)-kkal. Ezek, és a megfelelő könyvtár segítségével távolról is módosítható az adatbázis.

### Mikrokontroller szoftvere

A mikrokontrollert vezérlő szoftver két fő egységre osztható. Egyrészről kezelnie kell az érintőkijelzőt, másrészről adatokat kell beolvasnia és továbbítania.

#### Főprogram

A főprogram [FreeRTOS](#_FreeRTOS), taszkok segítségével látja el feladatait. A kódja [STM32CubeIDE](#_STM32CubeIDE) fejlesztőkörnyezetben íródott [C nyelven](#_C_nyelv).

Feladatai között megemlítjük:

1. Perifériák inicializálása és kezelése
2. Adatok beolvasása mérőeszközről, majd azok megformázása
3. Adatok továbbítása

#### Touchgfx

Ez az egység ugyan a főprogram egyik eleme, mégis külön egységként kezelendő. Ennek oka, hogy nem csak egy külön nyelven, külön fejlesztőkörnyezet segítségével is íródott, hanem egy nagyon konkrét feladatot is lát el: az érintőkijelzőt hivatott teljes mértékben kezelni.

[C++ nyelven](#_C++_nyelv), a [TouchGFX](#_TouchGFX) és [STM32CubeIDE](#_STM32CubeIDE) fejlesztőkörnyezetek segítségével íródott. A kijelzőn megjelenítendő elemek felfoghatók objektumokként, így szinte evidens a C++ használata. A TouchGFX szoftver segítségével viszonylag egyszerűen lehet vizuális elemeket létrehozni. Ezeket az elemeket a szoftver automatikusan C++ kóddá alakítja. Bizonyoson funkciók megvalósításához szükség volt a generált kód kiegészítéséhez, amit STM32CubeIDE segítségével lehetett eszközölni.

A Touchgfx szoftveregység feladata a kijelzőn kapott érintések értelmezése, és a kijelzőn ezeknek megfelelő változások kirajzolása. Ezek mellett valamilyen módon a C nyelvű főprogram felé továbbítja a felhasználó által bevitt és felhasználandó adatokat.

### Szerver

A szervernek két nagyon fontos feladata van: TCP üzeneteket tudjon fogadni és HTTP kéréseket tudjon kiszolgálni.

#### TCP server

A TCP szerver [Python](#_Python_nyelv) nyelven íródott a [VS Code](#_Visual_Studio_Code) fejlesztőkörnyezetben.

Főbb feladatai:

1. TCP kapcsolatok létesítése több porton is, párhuzamosan
2. TCP üzenetekből adatok kinyerése
3. A kinyert adatok formázása és lementése adatbázisba

Az adatbázishoz való hozzáférés miatt feltétlenül szükség van a [mysql-connector](https://pypi.org/project/mysql-connector-python/) könyvtárra. A TCP kapcsolatok kezelése végett pedig a [socket](https://docs.python.org/3.11/library/socket.html) könyvtárra van szükség.

#### Web backend

A szerver szolgál webszerverként is és API endpointként is. Webserver, mert képes kiszolgálni egy weboldal forrás filejait, és API endpoint, mert képes HTTP protokoll segítségével adatokat szolgáltatni, jelen esetben adatbázis adatait.

Mindkét funkcionalitás megvalósításához a [flask](https://flask.palletsprojects.com/en/3.0.x/) könyvtárat használtam, amely rendelkezik beépített fejlesztői szerverrel. Production szervernek a [gunicorn](https://gunicorn.org/) HTTP szervert használtam.

Az adatbázishoz való hozzáféréshez itt is elengedhetetlen a mysql-connector használata.

### Weboldal

A weboldalnak egy fő feladata van, mégpedig, hogy adatokat tudjon megjeleníteni. Ennek eléréséhez a [React](#_React_és_JSX) könyvtárat használtam a számos alkönyvtárával együtt. API endpointokon képes adatokat kérni. A weboldal kiszolgálásáért a már tárgyalt [szerver](#_Szerver) felel. A weboldal esztétikájának megvalósításához a [Bootstrap](#_Bootstrap) és a [FontAwesome](#_FontAwesome) CSS keretrendszereket alkalmaztam.

# Komponensek

(Nagy)Komponensek közti kapcsolatokról semmi, csak izolált funkciók(ide le is írni mit lehet majd olvasni, felkészíteni az olvasót logikailag mi következik)

Nagy komponsensek közötti kapcsolatokról nem írok, de hogy az alkomponenseken(pl adatgyűjtőben az esp stb) belüli kapcsolatok mik, igen!

## Adatgyűjtő

Kb 2.1-es ábra bezoomolva, esp szerepe, sensor szerepe, stm32 a fő ”agy”, mi miben íródott

### ESP

### Sensor

### Szoftver

#### Touchgfx

#### Főprogram

## Server

Kb 2.1-es ábra bezoomolva, programnyelv, tcp, szálak, mi a logika, felszabadítás, használat

## Adatbázis

Tábla(reverse engineer)

## Weblap

Kb 2.1-es ábra bezoomolva, API, programnyelvek, react

# Komponensek közvetlen kapcsolatai

Komponensek közti követlen kapcsolatok, ki kivel kommunikál és hogyan (ide le is írni mit lehet majd olvasni, felkészíteni az olvasót logikailag mi következik)

## Adatgyűjtő és szerver

Tcp kapcsolat, „q” lezárás, adatstruktúra, gyakoriság, kapcsolódás stb

## Szerver és weblap

API, webserver?

# Komponensek közvetett kapcsolatai

Logikai kapcsolatok a komponensek között és nem ténylegesen meglévő kapcsolatok

## Adatgyűjtő és adatbázis

Id a kapcsolat

## Adatgyűjtő és weblap

Id a kapcsolat

## Adatbázis és weblab

Szerveren keresztül api-val

Nem redundáns ez a kapcsolat?

# Értékelés

## Elemzés

Mennyire jó megoldás, design, esztétika, cyber security

## Továbbfejleszthetőség

Touchgfx, weblap, tcp server security (kóddal való csatlakozás)

# Köszönetnyilvánítás

A köszönetnyilvánítás nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

# Irodalomjegyzék

1. C programozási nyelv angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/C_(programming_language)>
2. C++ programozási nyelv angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/C%2B%2B>
3. Python programozási nyelv hivatalos honlapja: [https://www.python.org](https://www.python.org/)
4. HTML programozási nyelv angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/HTML>
5. CSS programozási nyelv angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/CSS>
6. JavaScript programozási nyelv angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/JavaScript>
7. Document Object Model angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/Document_Object_Model>
8. React könyvtár hivatalos honlapja: <https://react.dev>
9. Felhasználói felületek (UI) angol nyelvű Wikipedia leírása: <https://en.wikipedia.org/wiki/User_interface>
10. Bootstrap hivatalos honlapja: <https://getbootstrap.com>
11. FontAwesome hivatalos honlapja: [https://fontawesome.com](https://fontawesome.com/)
12. FreeRTOS hivatalos honlapja: [https://www.freertos.org](https://www.freertos.org/)
13. CMSIS\_RTOS2 dokumentációja: <https://arm-software.github.io/CMSIS_5/latest/RTOS2/html/index.html>
14. MySQL hivatalos honlapja: [https://www.mysql.com](https://www.mysql.com/)
15. Git hivatalos honlapja: [https://git-scm.com](https://git-scm.com/)
16. GitHub hivatalos honlapja: [https://github.com](https://github.com/)
17. VS Code hivatalos honlapja: [https://code.visualstudio.com](https://code.visualstudio.com/)
18. Node.js hivatalos honlapja: <https://nodejs.org/en>
19. STM32CubeIde hivatalos weboldala: <https://www.st.com/en/development-tools/stm32cubeide.html>
20. STMicroelectronics hivatalos honlapja: <https://www.st.com>
21. TouchGFX hivatalos weboldala: <https://www.st.com/en/development-tools/touchgfxdesigner.html>
22. MySQL Workbench hivatalos weboldala: <https://www.mysql.com/products/workbench/>
23. Roger Meier’s Freeware, CoolTerm honlapja: <http://freeware.the-meiers.org>
24. PuTTY hivatalos honlapja: <https://www.putty.org>
25. Espressif esptool.py dokumentációja: <https://docs.espressif.com/projects/esptool/en/latest/esp32/>
26. STM32H750B-DK mikrokontroller hivatalos weboldala: <https://www.st.com/en/evaluation-tools/stm32h750b-dk.html#overview>
27. STM32H750B-DK fanout board ismertető letöltő linkje: <https://www.st.com/resource/en/user_manual/um2695-stmod-fanout-expansion-board-for-stm32-discovery-kits-and-evaluation-boards-stmicroelectronics.pdf>
28. ESP8266-01 WIKI jellegű ismertető: <https://nurdspace.nl/ESP8266#Uploading_code>
29. Espressif ESP AT-Command dokumentáció: <https://docs.espressif.com/projects/esp-at/en/release-v2.2.0.0_esp8266/index.html>
30. ESP8266 felhasznált firmware: <https://github.com/CytronTechnologies/esp-at-binaries/blob/main/Cytron_ESP-01S_AT_Firmware_V2.2.0.bin>
31. SHT31 SEN0385 Szenzor adatlap: <https://dfimg.dfrobot.com/nobody/wiki/88b31350da4f54d00989c74c6fa392f7.pdf>
32. Dfrobot WIKI leírás az SHT31 SEN0385 Szenzorról: <https://wiki.dfrobot.com/SHT31_Temperature_Humidity_Sensor_Weatherproof_SKU_SEN0385>
33. „Query” fogalom magyarázat <https://www.techtarget.com/searchdatamanagement/definition/query#:~:text=A%20database%20query%20is%20either,other%20forms%20of%20data%20manipulation>.
34. Python mysql-connector könyvtár dokumentációja: <https://pypi.org/project/mysql-connector-python/>
35. Python socket könyvtár dokumentációja: <https://docs.python.org/3.11/library/socket.html>
36. Python flask könyvtár dokumentációja: <https://flask.palletsprojects.com/en/3.0.x/>
37. Python gunicorn HTTP szerver hivatalos honlapja: <https://gunicorn.org>

# Függelék

A függelék szövege.