

**ÓBUDAI EGYETEM**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar**

**Szabaduló szoba vezérlő szoftver tervezése**

**OE-BGK** Hallgató neve: **Kovács Árpád**

**2021.** Neptun kód: BPJZ56

**Tartalomjegyzék**

[1.](#_30j0zll) Bevezetés 4

[2.](#_1fob9te) Szakirodalom elemző feldolgozása 5

[2.1](#_3znysh7) Hivatkozások 5

[2.1.1](#_2et92p0) Hivatkozások sorrendje 6

[3.](#_tyjcwt) Saját munka 7

[3.1](#_3dy6vkm) Rajzok, képek, diagramok, ábrák 7

[3.2](#_1t3h5sf) Képletek 8

[3.3](#_4d34og8) Felsorolás 10

[4.](#_2s8eyo1) Összefoglalás 12

[5.](#_17dp8vu) Idegen nyelvű tartalmi összefoglaló 13

[6.](#_3rdcrjn) Felhasznált irodalom vagy Irodalomjegyzék 14

[7.](#_26in1rg) Mellékletek 15

# Bevezetés

A feladat egy olyan szabaduló szoba vezérlés létrehozása, mellyel akár a programot távolról is lehessen futattni. A programnak támogatnia kell, az ESP32 mikrovezérlőt, valamint tartalmazza, ábrázolja, és engedélyezze a szobák szoftverének szerkesztését.

A feladatot úgy kell megoldani, hogy Raspberry PI mikroszámítógépen is futtatható legyen.

A dokumentum célja a diplomamunka dokumentálása és leírása, a szakdolgozat fő célkitűzése volt, egy olyan szoftver készítése, mely a szabaduló szobáknak a létrehozásában és menedzselésében segít. A szoftvernek támogatnia kell, új szabaduló szobák létrehozását, ESP32 mikrovezérlő szoftverének vezérlését HTTP-protokolon valamint a szobát fenntartó személyzet segítését, a csapatok nyomon követését, és utoljára de nem utolsósorban a szobák programozását.

A feladat eredménye egy olyan alkalmazás létrejötte mely távolról is engedi a felhasználónak, hogy távolról is dolgozhasson.

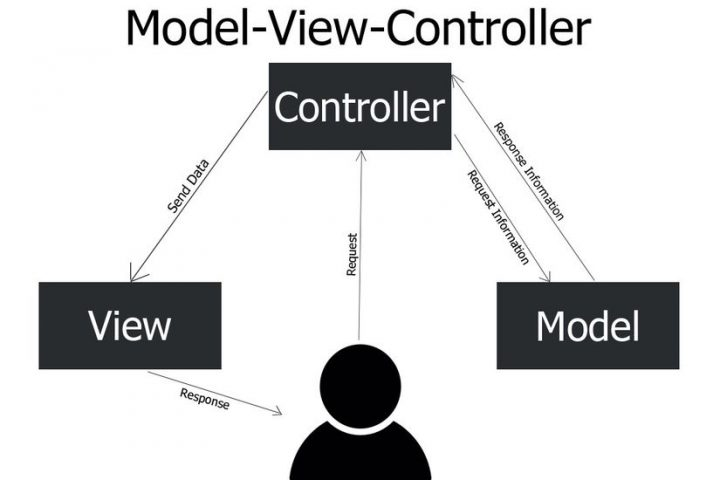
# Szakirodalom elemző feldolgozása

A feladathoz az esp32 típusú mikrovezérlőt lett választva, a feladatot úgy oldottam meg, hogy a program, egy közös csatornán keresztül kommunikál a szerverként üzemelő számítógéppel, és ez úgy lett megvalósítva, hogy azonos hálózathoz, van kötve az összes kliensként dolgozó Raspberry PI 400.

A feladat megoldásához elsődlegesen natív PHP-t terveztem felhasználni, de mivel a naptári év immáron 2021 úgy gondoltam, hogy ez túlságosan abszurd lenne ezért választottam a Laravel nevezetű PHP keretrendszert, mivel biztonságosabb.

A Laravel keretrendszer egy Symfony támogatottságra épülő szoftver mely imáron 8.0-ás verziót is túllépte. A keretrendszer multiplatformos, szóval ugyanúgy fel lehet használni Raspberry PI-ra mint Windows számítógépre.

A keretrendszer csomagoló rendszere Composer névre hallgat, majd felraktam a 8.0-ás Laravelt, és létrehoztam a kezdő fileokat.



Ezzek végeztével elkezdtem kidolgozni a feladathoz szükséges táblázati elemeket, később feltelepítettem a Laravel Jetstram nevezetű rendszert a keretrendszerbe, e program segítségével elkezdtem a felhasználói felületet mely Vue.JS Javascript támogatásával üzemel.

Vue 3.0 segítségével elkezdtem az egy oldalas alkalmazás létrehozzását. Mely csak Ajax használatával kommunikál a rendszerrel, a többit egy előre fordított Javascript-ten keresztül tölt be.

Mivel a Babel könyvtár nem támogatja Bootstrap verziót ezért szükséges volt a Tailwind CSS könyvtár használatát igénybe vennem.

A modulokat létrehoztam, majd elkezdtem a weboldalhoz tartozzó weboldalakat létrehozzni. Ezek egy Vue.js könyvtárban találhatóak, melyeknek direkt kommunikáció-val rendelkeznek a weboldalhoz. Ezt egy úgy nevezett Axios.JS csomagon keresztül intézi.

Mivel a Laravel egy MVC rendszer ezért szükségesnek érzem el is magyarázni, hogy is dolgozik egy Model-View-Controller struktúra.

A modellek

A modellek amik tárolják az adatokat, amelyek segítségével kommunikálunk az adatbázzissal, valamint ezeket az adatokat írjuk bele.

Erre példa amit használtam programok tárolásánál. Majd ennek a modell-hoz szükséges kidolgozni az adatbázis felőli kezelő felületét is.

View vagyis nézetek

Ebben a projektben ezek az útvonalaknak felelnek meg amit a rendszer felhasznál az előre fordított javascript fileoknál. Normál esetben a lekért útvonalat a szerver maga abban a pillanatban generálja le, és adja át a böngészőnek.

Controller vagyis a kezelő

Itt lehetséges a modellek viselkedését meghatározni eseményekre, pl. a lekért modellhez rakjon össze útvonalat és jelenítse meg. Ebben az esetben az Inertia. JS felel a nézet és kontroller közötti kommunikációért.

Babel fordító könyvtár

Ez a könyvárt lehetővé teszi újabb alkalmazások pl (Vue, React, vagy Angular) típusú alkalmazások felhasználást régebbi böngészőre, valamint a kódott optimalizálja. Az így létrehozott Stílus és Javascript fájl nem több pár megabytenál ezzel levéve a súlyos fileok mozgatását a szerverről. Ezt nevezik még WebPack technológiának is.

A Blockly könyvtár

Google által fejlesztett grafikus programozási program, mely képes több nyelvre is lefordítani a lehelyezett elemeket támogatott a php, javascript, go stb.

Az itt felhasznált része csak annyiban merül ki, hogy a felhasználó ebben rakja össze az ő általa írt programokat, és majd ezeket lementi és lefutattja a kezelő felület.

Ehhez szükséges volt egy külön konzol létrehozni mely lehetővé teszi a felhasználóknak a blockly által támogatott blockok futattását, valamint azok reakcióját a valós rendszerrel.

Raspberry PI 400

A szolgáltatás futattására én a Raspberry PI 400-as mikroszámítógépet használtam fel. Ezt használja a felhasználó az eszköz kezelésére.

Manjaro ARM

Manjaro ARM változatta ez az operációs rendszer fut a Raspberry PI 400-on. A Manjaro egy Arch típusú Linux disztrubició.

MySQL

A MySQL egy relációs adatbázis kezelő szoftver.

PHP

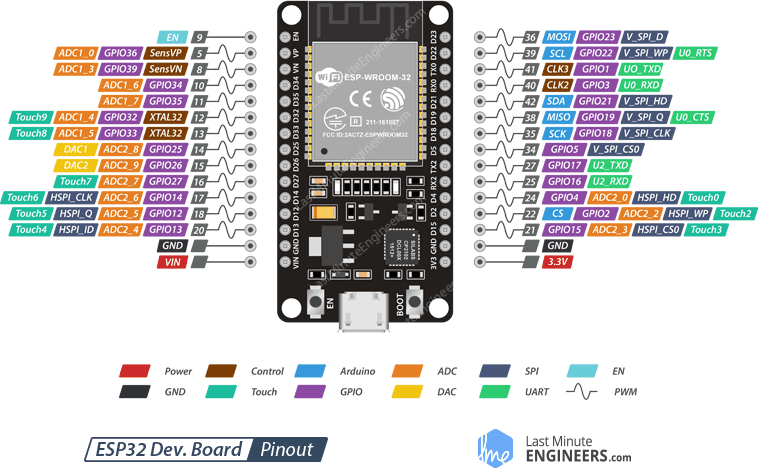
PHP vagyis prehypertext szkript nyelv ebben íródott a Laravel rendszer.

NPM/Node.JS

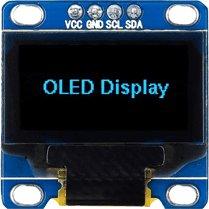
A rendszer az npm könyvártkezelő segítségevel építí fel magát.

ESP32

Az esp32 az espressif által tervezett és gyártott mikrokontroller. Ezt használtam fel a feladat megoldásához, ez az eszköz kommunikál a Raspberry Pi/vel a Laravel API csatornáján és állítja be a szoftvert.

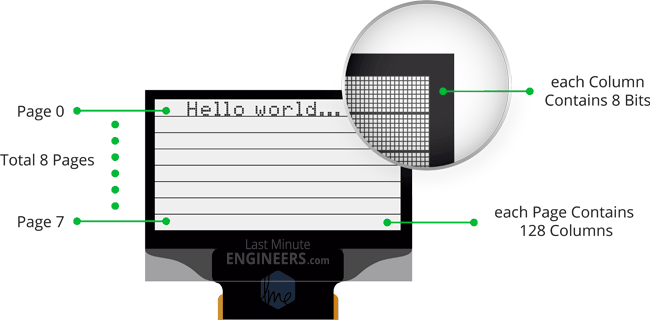


Az ESP32 támogat több -féle protokkolt egyenlőre 3 dolog lett kidolgozva képes relé üzemmódban dolgozni, érzékeli azt és elküldi az RFID kártya ID-jét a szervernek, valamint képes bemenetti módban is dolgozni azáltal, hogy az megjelölt () portott rövidre zárjuk a földdel. Utoljára és nem utolsósorban képes az eszköz megjelenítő eszközként is dolgozni amennyiben egy oled kijelzőtt kötünk rá.



Zerotier szolgáltatás

Egy nyílt forráskódú VPN szolgáltatás, melynek segítségével a fejlesztéshez használt szervergéppen lévő MySQL szerver valamint a Raspberry PI 400 kommunikál. Ehhez létre kellet hozni az ő hálozatukon egy Zerotier network-ot. Majd mind a szervergépet, és a Raspberry PI-t is rácsatlakoztattam.



Eszköz regisztráció

Az eszközön lévő kód

|  |
| --- |
| if(esp32\_pass=="")  {  JSONVar tmp1;  tmpaddress=web\_server+"?ajax&esp32\_welcome=reg";  Serial.println(tmpaddress);  res=http\_sys(tmpaddress);  tmp1=JSON.parse(res);  Serial.println(JSON.stringify(tmp1));  configobj["esp32pass"]=tmp1["pass"];  Serial.println(JSON.stringify(configobj));  save\_config();  ESP.restart();}    } |

### Hivatkozások sorrendje

A Felhasznált irodalomban szereplő publikációkat nem betűrendi sorrendbe, nem a megjelenés dátumának sorrendjében kell felsorolni, hanem a szövegben történő hivatkozás sorrendje szerint. Tehát a szövegben az első hivatkozás száma [1] és a dolgozat további részeiben egymás után, ahogy következnek. Természetesen, ha a későbbiekben ismét hivatkozni kell egy már számozott publikációra, megtehető.

https://github.com/google/blockly[1]

# Saját munka

A Laravel keretrendszert választottam majd, elkezdtem létrehozni a kezdő struktúrát, kezdve a felhasználóékéval.

Ehhez a laravel starterkit-jét telepítettem mely támogatja a CRFS védelmet valamint a botok elleni védelmet is ennek a telepítő kódja:

|  |
| --- |
| composer require laravel/breeze –dev  php artisan breeze:install  npm install  npm run dev  php artisan migrate |

Ez a starter kit támogatja az Inertia.JS felhasználását, mely egy lapos webalkalmazást lehet vele létrehozni.

Majd folytatva a beállításokat, beállítottam a levelező szervert, redis szervert.

Majd folytatva elkezdtem a modelleket létrehozni.

|  |
| --- |
| php artisan make:model Room -a |

Ez a parancs létrehozza:

* Modellt,
* Migrációs fájlt,
* Gyártó fájlt,
* Vezérlőt.

Mivel a Laravel az egy MVC kategóriába tartozik . ezért szükséges létrehozni a kezelő fájlokat.

## Adatbázis:

Rooms table:

|  |
| --- |
| Schema::create('rooms', function (Blueprint $table) {  $table->id();  $table->string('name');  $table->bigInteger('user\_id')->unsigned()->index();  $table->foreign('user\_id')->references('id')->on('users');  $table->timestamps();  }); |

Devices tábla:

|  |
| --- |
| Schema::create('devices', function (Blueprint $table) {  $table->id();  $table->string('name');  $table->string('password');  $table->string('ip\_address',45);  $table->enum('mode',array(["relay","rfid"]));  $table->integer('status');  $table->timestamp('last\_online');  $table->bigInteger('room\_id')->unsigned()->index()->default('1');  $table->foreign('room\_id')->references('id')->on('rooms');  $table->timestamps();  }); |

Program tábla:

|  |
| --- |
| Schema::create('programs', function (Blueprint $table) {  $table->id();  $table->string('name');  $table->boolean('active');  $table->text('xml\_block');  $table->text('javascript\_block');  $table->bigInteger('room\_id')->unsigned()->index()->default('1');  $table->foreign('room\_id')->references('id')->on('rooms');  $table->timestamps();  }); |

Kamera tábla:

|  |
| --- |
| Schema::create('cameras', function (Blueprint $table) {  $table->id();  $table->string('name')->default('Camera');  $table->string('url')->default('http://localhost');  $table->bigInteger('room\_id')->unsigned()->index()->default('1');  $table->foreign('room\_id')->references('id')->on('rooms');  $table->timestamps();  }); |

Teams tábla:

|  |
| --- |
| Schema::create('teams', function (Blueprint $table) {  $table->id();  $table->string('name');  $table->timestamps();  }); |

Runs tábla:

|  |
| --- |
| Schema::create('runs', function (Blueprint $table) {  $table->id();  $table->bigInteger('room\_id')->unsigned()->index();  $table->foreign('room\_id')->references('id')->on('rooms');  $table->bigInteger('program\_id')->unsigned()->index();  $table->foreign('program\_id')->references('id')->on('programs');  $table->bigInteger('team\_id')->unsigned()->index();  $table->foreign('team\_id')->references('id')->on('teams');  $table->timestamp('start\_time')->default(now());  $table->timestamp('finish\_time')->nullable();  $table->timestamps();  }); |

Ez által létrehoztuk a struktúrát.

A következő paranccsal meg lementjük a struktúrát az adatbázisba.

|  |
| --- |
| php artisan migrate |

MVC modell:.

[kép]

Ez szerint elkezdtem kidolgozni az MVC modellt.

Modellekel kezdtem.

Szoba tábla modellje:

|  |
| --- |
| class Room extends Model  {  use HasFactory;  protected $fillable=['name','user\_id'];  } |

Teams tábla modellje:

|  |
| --- |
| class Teams extends Model  {  use HasFactory;  protected $fillable=['name'];  } |

Runs tábla modellje:

|  |
| --- |
| class Runs extends Model  {  use HasFactory;  protected $fillable=['team\_id','program\_id','start\_time'];  } |

Device tábla modellje:

|  |
| --- |
| class Device extends Model  {  use HasFactory;  protected $fillable = [  'name',  'mode',  'status',  'room\_id'  ];  protected $hidden=[  'ip\_address',  'password'  ];  } |

User tábla modellje:

|  |
| --- |
| class User extends *Authenticatable*  {  use HasFactory, Notifiable;  protected $fillable = [  'name',  'email',  'password',  ];  protected $hidden = [  'password',  'remember\_token',  ];    protected $casts = [  'email\_verified\_at' => 'datetime',  ];  } |

Programs tábla modellje:

|  |
| --- |
| class Programs extends Model  {  use HasFactory;  protected $fillable = [  'name',  'active',  'javascript\_block',  'room\_id'  ];  protected $hidden=[  'xml\_block',  ];  } |

Camera tábla moddelje:

|  |
| --- |
| class Camera extends Model  {  use HasFactory;  protected $fillable = [  'name',  'url',  'room\_id'  ];  } |

Ezek végeztével elkezdtem kidolgozni a nézeteket is. Ezeknek a használatához igénybe vettem a Vue.JS könyvtárat. Majd párhuzamosan kidolgoztam az útvonalat (Route-t) és a hozzá tartozó nézetet. Mivel a cél az egy oldalas alkalmazás készítése volt ezért segítségül vettem megint az Inertia.JS könyvtárát és az ahhoz tartozó tábla kezelő modulokat is.

|  |
| --- |
| npm install inertia-table |

Ezt a modult feltelepítve létrehoztam a következő rendszert.

|  |
| --- |
| <?php  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Route;  use App\Models\Device;  */\**  *|--------------------------------------------------------------------------*  *| API Routes*  *|--------------------------------------------------------------------------*  *|*  *| Here is where you can register API routes for your application. These*  *| routes are loaded by the RouteServiceProvider within a group which*  *| is assigned the "api" middleware group. Enjoy building your API!*  *|*  *\*/*  Route::middleware(['middleware'=>'device'],function () {  Route::get('add-device',function (Request $request){  *// return '';*  $id=Device::*create*(['name'=>'Default','ip\_address'=>$request->getClientIp(),'mode'=>'relay','status'=>'0','last\_online'=>now(),'room\_id'=>1,'password'=>''])->id;  Device::*findOrFail*($id)->update(['password'=>(string)*md5*('salt'.$id.'salt')]) ;  return Device::*findOrFail*($id);  });  Route::get('status/{device\_password}',function ($device\_password){  return Device::*findOrFail*($device\_password);  });  Route::get('store',function (Request $request){  return Device::*latest*()->orderBy('created\_at','desc')->get();  });  });  Route::middleware('auth:api')->get('/user', function (Request $request) {  return $request->user();  }); |

Hálózatkezelés:

Az alapötlete a szervernek és a használatának a peertől-peerig való használat. Ez magába foglalja az adatok védelmét is mivel az adatok 80-as porton utaznak ezért szükséges volt védelmet használni ezért szükséges volt bevezeti a Raspberry Pi-t mint az esp32 hálózatnak a „router-ét” amivel eléri a szervert, ehhez én a zerotier nevezetű nyílt forrású vpn szolgáltatás vettem fel, melyel a felhasználó hozzáférést az eszközhöz LAN kapcsolaton keresztül a hálózathoz, így könnyedén a többi eszköz felcsatlakozhat a zerotier vpn-jéhez, amin keresztül az esp32 mikrovezérlők hozzáférnek a szerverhez. Valamint szintén hálózati hídat létrehozva lehet kapcsolódni a vezérlő weboldalhoz is melynek a címe: [http://10.147.17.234](http://10.147.17.234/).

Raspberry PI beállítása:.

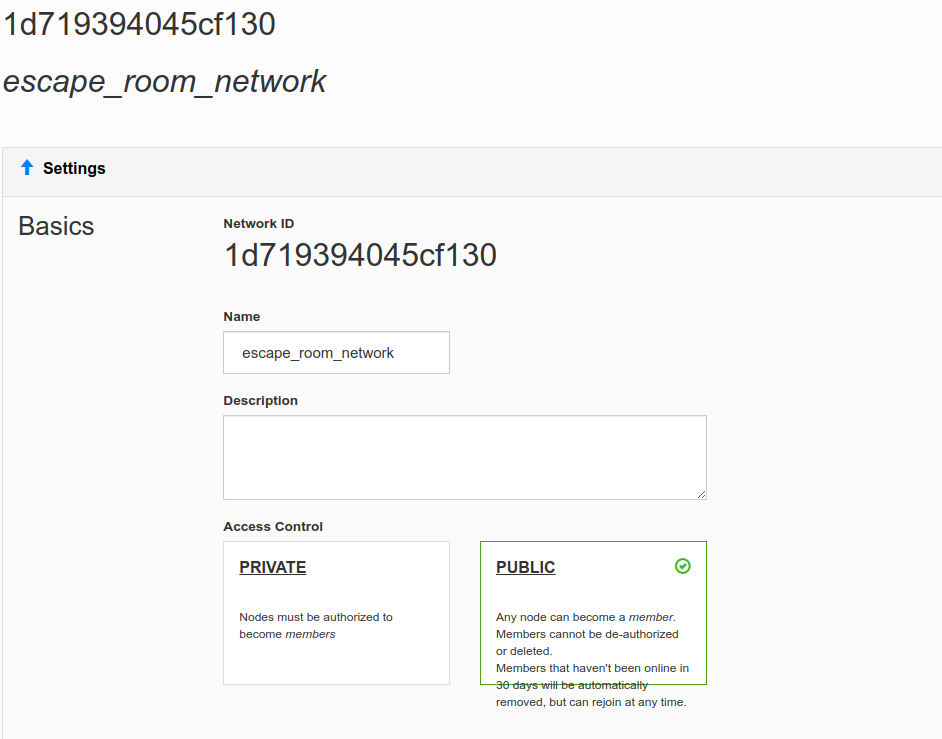
Zerotier alkalmazást feltelepítve beállítottam, hogy induláskor először lecsatlakozon a hálózatról, majd visszacsatlakozon azért, mert a képfájl ami a github tárolóban fent, van nejelentsen ütközést, a legelső indulásnál.

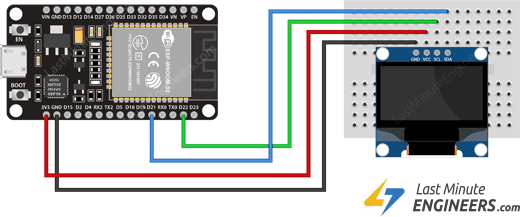
|  |
| --- |
| zerotier-cli leave 1d719394045cf130  zerotier-cli join 1d719394045cf130 |

Rasp Ap nyílt hozzáférésű szoftver felraktam linux operációs rendszerre. Majd beállítottam, hogy a Zerotier hálózatra adjon hozzáférést a beállított SSID, és wifi jelszó segítségével.

|  |
| --- |
| SSID: Escape Room  Jelszó: escape\_room |

Majd felállítottam egy Webmin szolgáltatást is a rendszeren. Ahol átírányítottam kéréseket a VPN szolgáltatásra.

Valamint ki



A szerkesztéshez Egyenletszerkesztő alkalmazását javasoljuk.

A szövegben az összefüggésre (3.2) történő hivatkozás, hasonlóan az ábrára, táblázatra való hivatkozáshoz történjen. A számozást fejezetenként újra kell kezdeni.

|  |
| --- |
| Alias /escape\_room "E:/Github/obudai\_diplomamunka/server\_part"  <Directory "E:/Github/obudai\_diplomamunka/server\_part">  AllowOverride AuthConfig  Require local  ErrorDocument 403 /error/XAMPP\_FORBIDDEN.html.var  </Directory> |

## Összefoglalás

A dolgozat utolsó része a rövid tartalmi összefoglaló, mely minimum egy oldal legyen, terjedelemben az egész Szakdolgozat maximum 5%-a lehet (1500-2500 karakter). Tartalmazza a kitűzött célok, felvetett problémák megoldását továbbá a következtetéseket. Térjen ki a gyakorlati alkalmazhatóságra.

# Idegen nyelvű tartalmi összefoglaló

Escape room management software which is based on ESP32, contains admin user interface, and multiple features for easier setup, reset, and for measuring time.

A magyar nyelvű összefoglalás idegen nyelven történő közlését tartalmazza. Az idegen nyelvű tartalmi összefoglaló elkészítése történhet: angolul, németül, oroszul vagy franciául.

# Felhasznált irodalom vagy Irodalomjegyzék

Példák az egyes publikációk megnevezésére.

Könyv esetében: Szerző (vagy Szerkesztő): Cím. Kiadó. Hely, Évszám. Oldalszám.

[1] M. Csizmadia Béla: Modell-alkotás. Nemzeti Tankönyvkiadó. Budapest, 2003. p. 21-24.

Szakcikk esetében: Szerző(k): Cím. Folyóirat megnevezése. Megjelenés száma/dátuma. Oldalszám.

[2] Hervay Péter – Csesznok Sándor: Marási környezet szimulációjának alkalmazása az oktatásban. Gépgyártás XLVII. évfolyam, 2007. 2-3. szám p. 15-17.

Jegyzet esetében: Szerző (vagy Szerkesztő): Cím. Oktatási Intézmény. Hely, Évszám. Oldalszám.

[3] Ambrusné dr. Alady Márta – Galla Jánosné – dr. Sipos Sándor: A Gépgyártástechnológia alapjai. BMF jegyzet. Budapest, 1995. p. 50-55.

CD-n megjelent publikáció esetében: Szerző (vagy Szerkesztő): Cím. Publikáció (Konferencia) neve. Hely, Évszám. CD. ISBN szám.

[4] Miko, B.: Hybrid inteligence system for mold cost estination. VI. International tools conference (ITC) 2007. Zlin, May 22-23, 2007. CD. ISBN-978-80-7318-572-5

Internetes publikáció esetében: Szerző (vagy Szerkesztő): Cím. Hely. Évszám. Honlap elérhetősége.

[5] Dr. Szabó László: Forgácsolás, hegesztés. Miskolc 2000.

<http://mek.oszk.hu/01200/01200/html/html>

# Mellékletek

Minden mellékletet sorszámmal, megnevezéssel kell ellátni. A szakdolgozat szövegi részében hivatkozni kell (1. sz. melléklet).

1. sz. melléklet: Az alkatrész összeállítási rajza.

2. sz. melléklet: A termékvisszahívás folyamata.

https://medium.com/@fmacedoo/standalone-application-with-electron-react-and-sqlite-stack-9536a8b5a7b9

https://lastminuteengineers.com/oled-display-esp32-tutorial/

<https://lastminuteengineers.com/multiple-ds18b20-esp32-web-server-tutorial/>

<https://medium.com/@dannyhuang_75970/learning-laravel-controllers-101-ad28d2bb5569>

https://github.com/RaspAP/raspap-webgui

https://medium.com/swlh/markup-images-with-vue-js-a4190d302005