



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
TMIT Tanszék

Aradi Péter István

# **LÉPTETŐMOTOROK AZ OKOS OTTHONOKBAN**

Sötétítő vezérlése léptetőmotorral

KONZULENS

**Dr. Fehér Gábor**

BUDAPEST, 2023

# Tartalomjegyzék

<b>Kivonat.....</b>	<b>5</b>
<b>Abstract.....</b>	<b>6</b>
<b>1 Bevezető .....</b>	<b>7</b>
<b>2 Irodalomkutatás.....</b>	<b>8</b>
2.1 Léptetőmotorok működése .....	8
2.2 Léptetőmotorok típusai .....	9
2.2.1 Unipoláris motorok .....	9
2.2.2 Bipoláris motorok .....	10
2.2.3 Hibrid motorok .....	10
2.3 Általam használt driverek .....	11
2.3.1 Az ULN2003A alapú driver .....	11
2.3.2 Az A4988 driver .....	12
2.4 Általam használt mikrokontroller .....	14
2.5 A programozáshoz használt szoftver .....	15
2.6 Pozíció meghatározása.....	15
2.6.1 A kódban számolós módszer .....	15
2.6.2 Az abszolút encoder .....	16
2.6.3 Ellenáramerősség-érzékelés módszere .....	17
2.7 Végállások .....	18
2.7.1 A mechanikai végállás .....	18
2.7.2 Az optikai végállás.....	19
2.7.3 A mágneses végállás .....	20
2.8 Az okos otthon sötétítő vezérlési megoldások felépítésének és funkcióinak vizsgálata .....	21
2.8.1 A Lutron Serena Okos Sötétítő .....	21
2.8.2 A SmarterHome MySmartRollerShade .....	21
2.8.3 Az IKEA Fyrtur .....	22
2.8.4 A Yoolax motorizált árnyékoló .....	22
2.8.5 A Graywind motorizált napelemes sötétítő .....	23
<b>3 Tervezés .....</b>	<b>24</b>
3.1 A motor kiválasztása.....	24

3.2 A driver kiválasztása.....	24
3.3 A mikrokontroller kiválasztása .....	24
3.4 A programozáshoz használt szoftver kiválasztása.....	24
3.5 A pozíció meghatározásának kiválasztása .....	25
3.6 A végállás kiválasztása .....	25
3.7 A működést segítő további alkatrészek választása .....	26
3.8 A rendszerem összeszerelésének megtervezése.....	27
3.9 A kódom terveinek bemutatása.....	28
3.10 A webszerver terveinek bemutatása.....	33
3.11 A weboldal terveinek bemutatása .....	33
<b>4 Implementálás .....</b>	<b>35</b>
4.1 A hardver összeszerelése .....	35
4.2 A kód és a weboldal elkészítése .....	35
4.3 A végállás és a hozzá tartozó kód megvalósítása .....	36
4.4 A 3D nyomtatott fogaskerék .....	37
4.5 Az áttétel .....	38
4.6 A megvalósítás képekkel illusztrálva .....	39
<b>5 Tesztelés .....</b>	<b>42</b>
5.1 A rendszer használhatóságának bizonyítása .....	42
5.2 A helyes működés tesztelése.....	43
5.3 Hogyan tovább? .....	44
<b>6 Összefoglalás.....</b>	<b>46</b>
<b>7 Irodalomjegyzék.....</b>	<b>47</b>
<b>8 Képek forrásai.....</b>	<b>50</b>

# HALLGATÓI NYILATKOZAT

Alulírott **Aradi Péter István**, 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. 12. 08.

.....  
Aradi Péter István

# Kivonat

Manapság az életünkben egyre nagyobb szerepet kapnak azok a technológiai megoldások, amelyek könnyebbé és komfortosabbá teszik életünket. Gyakran találkozunk már az okos otthonokban is új technológiákkal. A felokosított lámpákon túl az otthoni eszközök motorizációja tovább fokozza a kényelmet, a léptetőmotorok felhasználása ideális választás lehet erre a feladatra, hiszen precízen, halkan, megbízhatóan működnek.

A félévben a feladatom egy WiFin keresztül vezérelhető sötétítő mozgató rendszer megtervezése és megvalósítása volt okos otthonok számára. A probléma a gyöngylánc kézi mozgatása, amely a motor segítségével helyettesíthető.

A dolgozat célja egy olyan műszaki megoldás megtervezése és megvalósítása, amely mindenki számára egyszerűen kezelhető, kényelmes és energiatakarékos megoldást nyújt.

A végeredmény egy olyan sötétítő mozgató rendszer lett, amely egy weboldalon található vezérlő panelről irányítható, végül az általam tervezett, megvalósított és tesztelt rendszer a feladatkiírásban szereplő funkciókat megfelelően kivitelezi.

Ennek a projektnek a megvalósítása, azért volt hasznos, mert egy számomra ismeretlen témakörben rengeteg elméleti és gyakorlati tudásra tettem szert és nagy a valószínűsége annak, hogy a jövőben a tanulmányaim során, vagy a munkahelyemen valamilyen formában ezt a tudást hasznosítani fogom.

# **Abstract**

Nowadays, technology is playing an increasingly important role in our lives, making our lives easier and more comfortable. We often see new technologies in smart homes. In addition to smart lights, the motorisation of home appliances can further enhance convenience, and the use of stepper motors can be the ideal choice for this task, because they are precise, quiet and reliable.

This semester my task was to design and implement a WiFi-controlled smart shade system for smart homes. The problem is the manual movement of the bead chain, which can be replaced by a stepper motor.

The aim of this thesis is to design and implement a technical solution that is convenient, energy efficient and easy to use for everyone.

The final result was a smart shade system that can be controlled from a control panel on a website. In the end, the system I designed, implemented and tested performs the features as specified in the task assignment.

Creating this solution was useful, because I have gained a lot of theoretical and practical knowledge on a topic that was unknown to me until now, and there is a high probability that I will use this knowledge in some form in my future studies or in my future workplace.

# 1 Bevezető

Manapság az életünkben egyre nagyobb szerepet kapnak azok a technológiai megoldások, amelyek könnyebbé és komfortosabbá teszik életünket. Gyakran találkozunk már az okos otthonokban is új technológiákkal. A felokosított lámpákon túl az otthoni eszközök motorizációja tovább fokozza a kényelmet, a léptetőmotorok felhasználása ideális választás lehet erre a feladatra, hiszen precízen, halkan, megbízhatóan működnek. Az okos otthonokban, azokon a helyeken, ahol kötött pályán való mozgásról beszélünk találkozhatunk léptetőmotorokkal. Ezek közé tartozik az ajtók és az ablakok nyitása és zárása, ételliftek mozgatása, vagy éppen a sötétítők fel- és lehúzása. Ezek a motorok lehetővé teszik a rendszerek számára a pontos irányítást és pozicionálást, ezzel hozzájárulva a felhasználói élmény javításához és az energiahatékonyság növeléséhez.

A félévben a feladatom egy a fentebb leírt sötétítő mozgató motor és vezérlőjének elkészítése volt okos otthonok számára. A probléma a gyöngylánc kézi mozgatása, amely a motor segítségével helyettesíthető. Elsőre nem tűnhet nehézségnek, azonban, ha jobban belegondolunk ez a kis „apróság” is okozhat gondokat az ott lakóknak. Ha a gyöngylánc nehezen hozzáférhető helyen van, vagy ha egy gyermek az, aki használni szeretné, nem biztos, hogy eléri a láncot, az átlagember számára pedig elsősorban kényelmi funkciót lát el ez a megoldás. Ezen indokok tanúskodnak a probléma fontosságáról.

A továbbiakban először a léptetőmotorok működését és meghajtását fogom ismertetni, valamint a pozíció és a végállások meghatározásának lehetőségeit. Bemutatom a vezérlési megoldások felépítését és funkcióit. Majd prezentálom a terveimet, a meghozott döntéseimet és ezeket meg is magyarázom. Ezután megmutatom, hogy készült el a megtervezett eszköz. Végül bemutatom, hogy milyen tesztek végeztem és a tanulságokat, amikkel gazdagodtam.

A léptetőmotorok által vezérelt sötétítők megtervezése és megvalósítása olyan terület, ahol a kreativitás és a műszaki precizitás egyaránt fontos szerepet játszik, ugyanis kreativitás, vagy éppen ötlet nélkül sem és alapos, pontos munka nélkül sem lehetséges egy ilyen összetettségű projekt helyes megoldása.

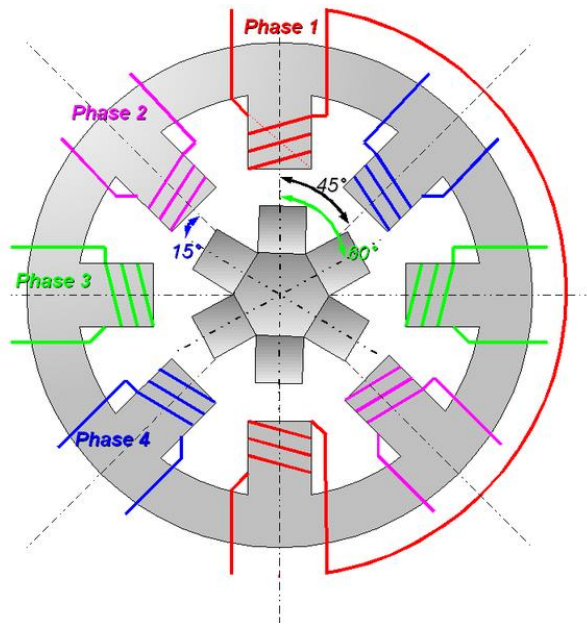
A dolgozat célja, egy olyan műszaki megoldás megtervezése és megvalósítása, amely mindenki számára egyszerűen kezelhető és kényelmes megoldást nyújt.

## 2 Irodalomkutatás

Munkám kezdetekor nem ismertem a léptetőmotorok működését, ezért utána kellett járnom és mivel a témám megértésében érdemes tisztában lenni alapvető működési elvükkel, ezért az alábbiakban részletezem, amit a témakörben elsajátítottam.

### 2.1 Léptetőmotorok működése

A léptetőmotor egy olyan szénkefe nélküli szinkron egyenáramú motor, amely sok alapvető villanymotorral ellentétben nem csupán folyamatosan forog körbe, amíg tápfeszültséget kap, hanem ennél sokkal precízebb. Úgy van felépítve, hogy a rajta átfolyó áram áthalad több, sorba kötött tekercsen, amik fázisokba vannak rendezve és gyorsan lehet őket be- és kikapcsolni sorban, így megengedve a motornak, hogy egy fordulatnak csak törtrészt tegye meg. Ezekre az egyedi, előre meghatározott fázisokra hivatkozunk „lépések”-ként. A léptetőmotorok úgy vannak tervezve, hogy egy fordulatot sok kicsi lépésre bontsanak le és ennek eredményeként rendkívül precízek.



**2.1. ábra A léptetőmotorok keresztmetszete,  
illetve belseje Forrás: [1]**

A különböző típusú léptetőmotorok belső felépítése bizonyos mértékben eltér egymástól, azonban abban megegyeznek, hogy forgórészük mágnesből van és ezt veszi körül az állórész, amely tekercsekből, azaz elektromágnesekből áll. Az egymással szemközt elhelyezkedő tekercsek össze vannak kötve, így egyszerre kapnak áramot. A forgórész

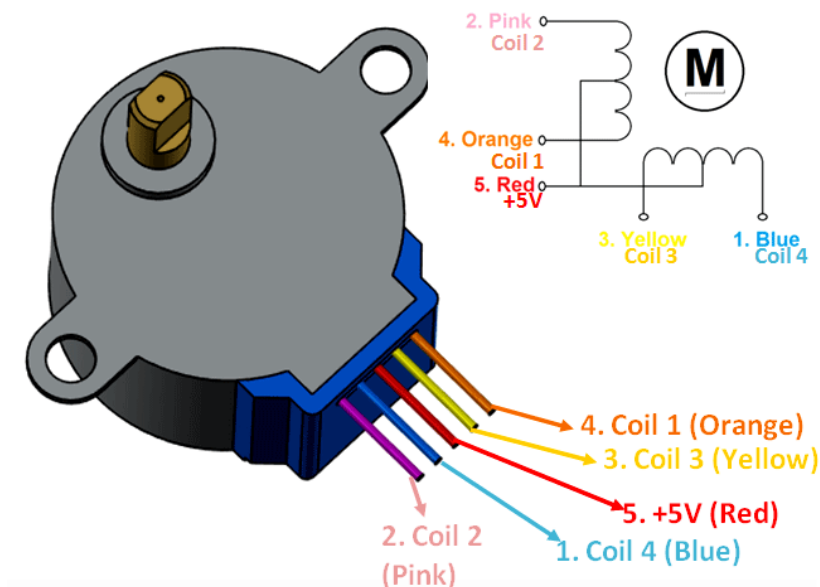


mindig úgy áll be, hogy párhuzamos legyen az egymással szemben elhelyezkedő, áram járta tekercsek által meghatározott vonallal. A 2.1. ábrán látható, hogy ott éppen az első fázis tekercsei kapnak áramot és az is látszik, hogy a következő fázisokban hány fokot fog fordulni a rotor. Ez a mágneses erőhatás miatt van és ennek köszönhetően működtethető a motor. A tekercseken megfelelő sorrendben folytatva az áramot a motor precízen léptethető és vezérelhető. A léptetőmotorok talán legnagyobb előnye a többi motorral szemben, hogy gyorsan el tudnak mozogni egy ismert és ismételhető pozícióba és ott addig tudnak maradni, amíg szükség van rá. [1]

## 2.2 Léptetőmotorok típusai

### 2.2.1 Unipoláris motorok

Munkám kezdetekor kaptam egy 28BYJ-48 típusú léptetőmotort és ehhez egy ULN2003A típusú interfésszel ellátott drivert. Ez a motor és driver, ebben az állapotában még képtelen a redőny mozgására nyomaték hiánya miatt. Ezeket az alkatrészeket azért kaptam, mert ez egy unipoláris motor, amely a legegyszerűbb típusú léptetőmotor azok közül, amiket ismerek és ennek köszönhetően viszonylag egyszerűen vezérelhető. Ezen a motoron tudtam gyakorolni és első ötleteimet kipróbálni. Ennek a motornak az egyszerűsége mellett a további előnye az, hogy kevés és egyszerű átalakítással bipoláris motor hozható létre belőle. Ehhez a motor belsejében található és a 2.2 ábrán látható két tekercs közepét összekötő piros vezetékot meg kell szakítani. A megszakítás két módon



2.2. ábra Az unipoláris motorok felépítése Forrás: [2]

lehetséges: a piros vezeték elvágásával, vagy a motorban található nyákon lévő középső vezeték megszakításával. Ekkor azonban már az eddigi driver nem képes meghajtani a motort, ezért új driverre van szükségünk, ami bipoláris motorokat is képes meghajtani. Az átalakításról az irodalomjegyzékben található bővebb információ. [2]

### **2.2.2 Bipoláris motorok**

További munkám során a konzulensemtől kapott A4988 driver-rel operáltam. Az A4988 egy „H-Bridge” típusú driver, amivel a bipoláris motorok képesek megfordítani az áram folyásának irányát. A fázisok feszültség alá helyezésével és a polaritás váltakoztatásával az összes tekercset egyszerre tudják használni, hogy forgassák a motort. Amíg az unipoláris motorok csak a tekercsek 50 százalékát használják egy időben, addig a bipoláris motorok az összeset, ezért a bipoláris motorok erősebbek és hatékonyabbak.

### **2.2.3 Hibrid motorok**

Miután átalakítottam a 28BYJ-48 motort bipolárisra, kaptam egy másik motort is. Egy NEMA17-es motornak a 17HS3001-20B modelljét, amely a 2.3 ábrán látható. Ezzel el is jutottunk a harmadik, egyben az utolsó általam tanulmányozott léptetőmotor típushoz, ugyanis ez a motor egy hibrid típusú léptetőmotor. A hibrid motorok a bipoláris motoroknál is pontosabbak, mert képesek fél lépésekre és mikrolépésekre is. Ennek köszönhetően egyenletesebben és pontosabban tudnak működni. A lépések és a fordulatok pontosabban vezérelhetők, a működés halkabb és jóval nagyobb a nyomaték-



**2.3 ábra A NEMA17 17HS3001-20B motor**

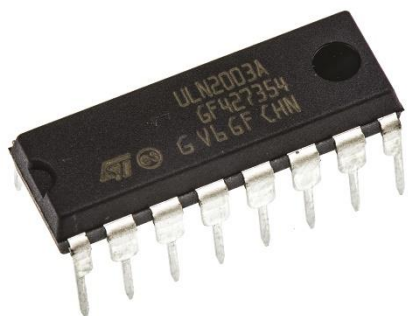
**Forrás:[3]**

méret arány. Ez annak köszönhető, hogy az állórész belső oldala és a forgórész külső oldala fogazott és a két tekercsen felváltva folyik áram, valamint váltakozik a polaritásuk is. A motor adatlapja is arról tanúskodik, hogy egy lépés 1,8 fokos, ami azt jelenti, hogy „full step” módban 200 lépés egy teljes fordulat. A munkám során ennek mélyebb tanulmányozására nem volt szükség, mivel nem használtam „full step” mód mellett más módokat. [3] A léptetőmotorok működéséről és típusairól a [4] forrásban lehet tájékozódni. [5]

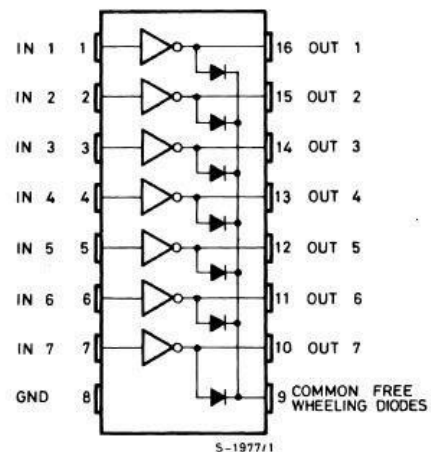
## 2.3 Általánosan használt driverek

### 2.3.1 Az ULN2003A alapú driver

Az ULN2003A unipoláris motorhoz tartozó IC, amely 16 lábbal rendelkezik, ahogy az a 2.4 ábrán is látszik. 7 bemenete, 7 kimenete, egy föld pinje van. Továbbá van egy olyan lába, amelyen a feszültséghatároló diódáknak a katódjai vannak egyesítve. Ezt a lábat általában a tápfeszültségre kell kötni. A IC, amely a 2.5 ábrán látható 7 invertert és 7 diódát tartalmaz, amely azért felel, hogy visszafelé ne tudjon áram folyni, így nagyon egyszerű a felépítése. Ennek az IC-nek a segítségével jön létre az ULN2003A alapú driver. A driver, ahogy az a 2.6 ábrán látható, rendelkezik 7 bemenettel és 5 kimenettel, valamint egy 5-től 12 V-ig terjedő táppal és egy föld pinnel. A bemenetek közül nekem csak 4-re volt szükségem. A kimenetek közül a piros vezeték pedig eltér a



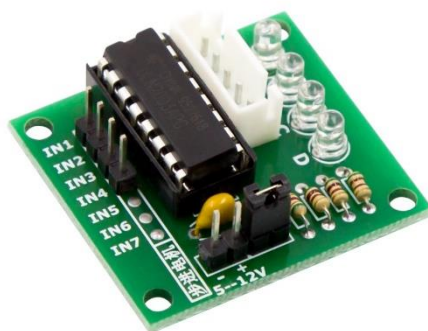
2.4 ábra Az ULN2003A IC Forrás: [4]



2.5 ábra Az ULN2003A IC műszaki rajza  
Forrás: [5]

többtől, ugyanis ez a motorban található két tekercs közepét köti össze. Az így kapott 4 tekercsrész közül

egyszerre csak egyen folyik áram. A driveren megtalálható továbbá 4 darab LED, amik felváltva villognak és mindig az világít, amelyiken éppen áram folyik. Ezen kívül még 4 darab ellenállás található meg a driveren, amelyek a bekötött bemenetekre vannak kötve. Az ULN2003A adatlapja a [6] forrásban található.



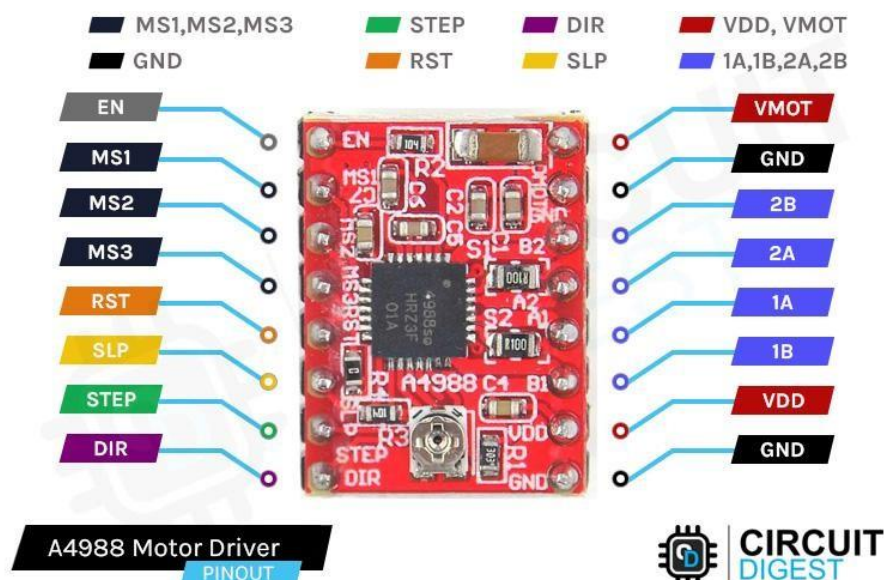
**2.6 ábra Az ULN2003A alapú driver**

**Forrás: [6]**

### **2.3.2 Az A4988 driver**

A fent leírtak szerint, miután átalakítottam a 28BYJ-48 motort bipolárisra már csak az A4988 drivert használtam. A továbbiakban erről a driverről fogok írni bővebben. Ez a driver beépített „translator”-rel, vagyis fordítóval rendelkezik, ami azt jelenti, hogy csupán két pin segítségével könnyedén vezérelhető a motor. Az egyik pinnel a forgás iránya állítható be, amíg a másikkal lehet léptetni a motort. A driver öt különböző lépésméretre képes, amelyek közül a legnagyobb a „full step”, vagyis a teljes lépés és a legkisebb a „sixteenth-step”, vagyis a tizenhatod lépés. Ezek között található még 3 méret, amelyek a legkisebbtől duplázódnak, amíg el nem érik a teljes lépést. Tartalmaz egy potmétert, amely segítségével állítható a kimeneti áram nagysága. A beállítás úgy történik, hogy egy ampermérőt sorosan bekötünk az egyik tekercs egyik vége és a driver közé és addig tekergetjük a potmétert, amíg a mért érték meg nem egyezik az adatlapon megadott áramértékkel, ami az én esetemben 1,2 A volt. Hűtés nélkül maximum 1 Ampert bír el tekercsenként, de hűtőborda segítségével, akár a dupláját is. Én nem használtam hűtőbordát, de nem is volt rá szükség az esetemben. Az előző driverhez hasonlóan, ennek

a drivernek is 16 pinje van, azonban itt teljesen más az elrendezés. A könnyebb megértés érdekében a 2.7 ábrán látható a driver a feliratozott pinekkel. A jobb alsó sarkában található egy föld pin és egyből fölötte egy tápláló pin, amelyek segítségével a driver az áramot kapja. Ugyanezen az oldalon a következő 4 pin a motor tekercseire csatlakozik rá. Az 1A és 1B az egyik tekercs, míg a 2A és 2B a másik tekercs két végére kapcsolódik. Ezen az oldalon az utolsó két pin a motor áramellátását biztosítja. A jobb felső pin a táplálásért, az alatta lévő pin pedig a földelésért felelős. A bal oldalon is alulról indulunk és a legelső pin a dir pin, ami a motor forgásának irányát határozza meg. A fölötte lévő step pin pedig minden alkalommal, amikor „HIGH” állásba lép, léptet egyet a motoron. Ezután várunk egy általunk meghatározott időt, majd „LOW” állásba tesszük és ismét várunk. A lépések olyan sebességgel követik egymást, ahogy azt a várakozások nagyságával állítjuk. A következő a sleep pin, ami, ha logikai 0-t kap, akkor alvó módba teszi az eszközt. A következő pin a reset pin, amely, ha logikai 0-t kap egy előre definiált állapotba lép. A fentebb lévő 3 pin az MS1, MS2 és MS3 az 5 lépésméret közül egy kiválasztására való. Ehhez számtalan igazságtáblát találhatunk az interneten. Az én esetemben azonban, mivel „full step” módban használtam, nem volt szükség ezen pinek bekötésére. Ha „half step” módba állítom, a tekercsek nem teljesen felváltva kapnak áramot, hanem először kap az egyik tekercs, utána mindkettő egyszerre és végül csak a másik. A „microstep” módok használatához a driver több különböző nagyságú áramot



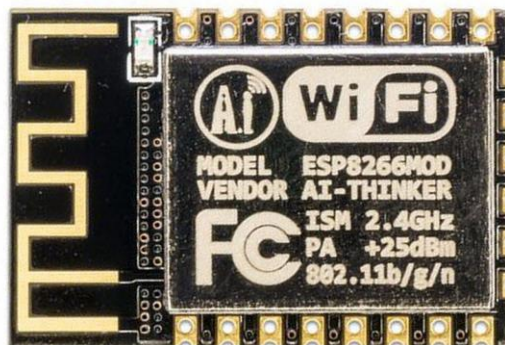
2.7 ábra Az A4988 motor driver Forrás: [7]

szolgáltat a motornak, így léptetve azt. Az utolsó pin az enable pin, vagyis az engedélyező pin, amely logikai 1 hatására tiltja a kimeneteket.

Bár a felépítése ennek a drivernek jóval bonyolultabb, mint az unipolárisnak, azonban éppen emiatt sokkal könnyebben vezérelhető. Továbbá az A4988 driver mellett szól az is, hogy ez csak bipoláris és hibrid, amíg az ULN2003A driver csak unipoláris motor vezérlésére képes. A motorok összehasonlításakor pedig fontos szempont, hogy a Nema17 motor robusztusabb és precízebb még az átalakított 28BYJ-48 motornál is. Az adatlapja megtalálható a [7] forrásban.

## 2.4 Általam használt mikrokontroller

A mikrokontrollerek lényegében kicsi számítógépek, melyek processzorral, memóriával, valamint ki- és bemenetekkel rendelkeznek. Beágyazott rendszerekben széles körben alkalmazzák őket, de az alapelveik mindig azonosak: a mikrokontroller a rá telepített szoftver segítségével végrehajt különböző feladatokat. Képes válaszolni a bemeneti jelekre kimeneti jelekkel, értelmezni és továbbítani a szenzorok adatait, valamint mozgatni a rákapcsolt léptetőmotorokat. Általában kis méretűek, alacsony energiafogyasztásúak, de teljesítményük és memóriájuk limitált.



**2.8 ábra Az ESP-12F mikrokontroller**

**Forrás: [8]**

A 2.8 ábrán látható ESP-12F mikrokontrollert használtam a motor vezérléséhez, Wi-Fi kapcsolat kialakításához és az interneten keresztüli eléréshez és a vezérlési parancsok kiadásához szükséges weboldal futtatásához.[8]

## 2.5 A programozáshoz használt szoftver

A mikrokontroller programozásához a nagy népszerűségnek örvendő Arduino IDE szoftvert használtam. Ennek kezelése egyszerű és könnyen átlátható. A programban alapvetően C++ nyelven írhatunk kódokat, de számtalan könyvtár és beépített függvényt használhatunk, továbbá tömérdek előre megírt hasznos példakód van az IDE-ben, hogy munkánkat támogassa. Minden kódnak van egy alapértelmezett felépítése, ami két függvényből áll. A mögöttes elképzelés az, hogy az egyik függvény az eszköz bekapcsolásakor vagy újraindításakor egyszer lefut és beállítja a hardverre rákapcsolt pinek pozícióját, azokat a változókat, amiket a későbbiekben használni fogunk és további konfigurációkat az eszközzel kapcsolatban. Ez a függvény „void setup(){...}” néven található meg minden működő kódban. A másik függvény a „void loop(){...}” nevet viseli és a „void setup(){...}” után fut le. Ez a függvény egy végtelen ciklus és addig fut le újra és újra, amíg a rendszert nem áramtalanítjuk, vagy el nem alszik. Ebben a függvényben történik minden, amit szeretnénk, hogy az eszközünk csináljon. Az én esetemben ez most a motor shaft-jának forgatása és ezzel a sötétítő mozgatása és vezérlése. Ha kész van a kódunk, akkor a megfelelő gombra kattintva a szoftver ellenőrzi a kódunkat és ha nem talál hibát elkezd feltölteni azt a mikrokontrollerre.

## 2.6 Pozíció meghatározása

Ahhoz, hogy majd a motort pontosan tudjuk vezérelni, minden pillanatban tudnunk kell, hogy éppen hol jár a forgásban és hány lépést tett meg eddig és melyik irányba. Ennek számontartására számos lehetőség van. Ezek közül fogok most néhányat kiemelni, amelyek szóba jöhetnek a pozíció meghatározásakor.

### 2.6.1 A kódban számolás módszer

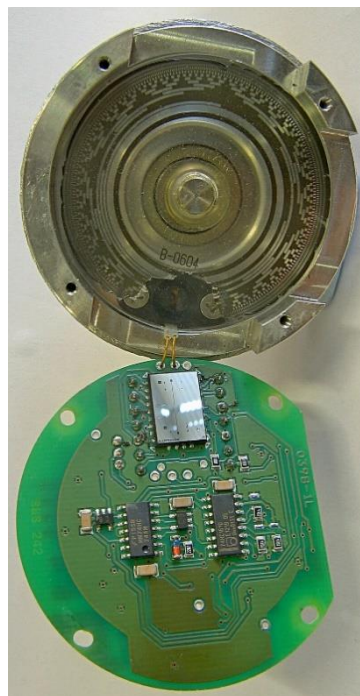
Az első legészserűbbnek tűnő megoldás, ha egyszerűen számoljuk egy változóval a lépéseket és minden lépés után növeljük, vagy csökkentjük eggyel a változó értékét attól függően, hogy éppen melyik irányba forog a motor. Ennek a hátránya, hogy mindig ugyanarról a pontról kell indítanunk a motort és a számlálást és ahhoz, hogy a pozitív számok halmazán maradjunk ennek a pontnak az egyik végpontnak kell lennie, ami sötétítő húzása esetében a legfelső, vagy a legalsó pontot jelenti. Ennél a megoldásnál problémát jelenthet viszont, ha működés közben elveszti a tápot áramszünet, vagy bármilyen egyéb okból kifolyólag, mert így, amikor újra áramot kap, visszaállítja az



összes változót eredeti állapotába. Ezáltal elveszti, hogy hol jár és azt a pozíciót, ahol éppen tartózkodik, fogja a beállításoktól függően legalsónak, vagy legfelsőnek tekinteni. Ez veszélyes, mert a sötétítőt és magát a motort is megrongálhatja, vagy akár tönkre is teheti. Erre a problémára jelentenek megoldást a végállások, amikről majd később lesz szó. Ennél a megoldásnál probléma lehet, ha a motor kihagy lépéseket, ugyanis attól, hogy ez történik a számolás nem áll meg és úgy lesz számolva, mintha minden rendben zajlott volna. Ez a nehézség akkor jön szóba, ha a motor túl van terhelve és már nem bírja gond nélkül működtetni a rendszert. Erre a hibára is megoldást jelentenek a végállások, azonban van egy másik megoldás, ami az előző problémát is megoldja és ez nem más, mint a fizikai pozíció encoder.

### 2.6.2 Az abszolút encoder

Az absolute rotary position encoder vagy más néven az abszolút forgó pozíció encoder egy érzékelő, amely pontosan követi a léptetőmotor rotorjának aktuális pozícióját, így információval szolgál a rendszer számára és lehetővé teszi, hogy a vezérlés pontosan szabályozza a motor forgását és pozícionálását. Továbbá az encoder a motor és a driver áramtalanításakor se veszti el az adatokat és onnan számol tovább, ahol abbahagyta. Így encoder használatakor nincs szükség végállásokra sem és mivel a rotort



2.9 ábra Az absolute rotary encoder

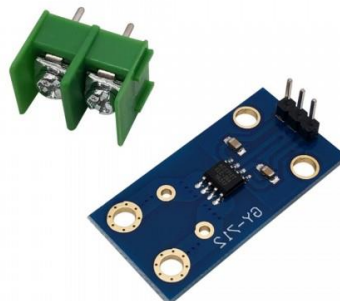
Forrás: [9]



figyeli, ezért, ha átugrik egy lépést a motor, észreveszi és nem számolja, így nem lesz eltolódva a számolás a rotor pozíciójától. A 2.9 ábra egy ilyen encodert szemléltet. Bár sok előnye van, ugyanis fontos problémákat kiküszöböl, de az előnyök mellett hátrányai is vannak. Egy ilyen eszköz drága, sok helyet foglal és még a rendszert is jóval komplexebbé teszi, így nem feltétlenül ez a legjobb megoldás, ami szóba jöhet. Ebben az oktatóvideóban az encoderek működéséről esik szó [9].

### 2.6.3 Ellenáramerősség-érzékelés módszere

Encoder nélkül is meg lehet oldani a problémát a lépések kihagyásáról, ugyanis áramerősség érzékelés segítségével meghatározható, ha egy lépés kimaradt. Ha olyan rotort használunk, amelyben a szemközti pólusok megegyeznek, de nem egyenlő nagyságúak, tehát a rotor pólusai nem szabályosak, akkor a rotor bármely módon meghajtható. Ez azonban Lenz törvénye szerint indukál egy ellenáramot, mivel a vezető, azaz a tekercsek körül változik a mágneses tér a szabálytalan rotornak köszönhetően. Amikor a rotor forog, ezeknek az áramoknak a 2.10 ábrán látható áramerősség-érzékelő szenzorok érzékelésével meg lehet határozni az abszolút pozíciót. A mért áramokat diszkrét Fourier transzformációval átalakíthatjuk komplex együtthatókká. E-ad-osos alakban meghatározható a rotor szöge. Amennyiben a becsült és mért szögeknek vesszük a különbségét megkapjuk a deltákat, amik különbségének a fele egy viszonylag pontos mérés a hibáról. Ha úgy vezérlünk, hogy ez a hiba 0-hoz tartson a motor kalibrálható lesz és a kihagyott lépések észlelhetővé és javíthatóvá válnak. Így a motor encoder nélkül megoldja a problémát és a motor nyomatéka is csak 7%-kal csökken a Lenz törvény



**2.10 ábra Áramerősség-érzékelő szenzor**

**Forrás: [10]**

alapján fellépő ellenáram miatt. A twentei egyetem oktatóvideójából megismerhetjük részletesebben az ellenáramerősség-érzékelés módszerét [10].

## 2.7 Végállások

A továbbiakban a végállásokról fogok írni. A végállások, olyan szenzorok, amelyek, ha érzékelik a mozgás fizikai határát, vagy végpontját, jeleznek számunkra kimenetük megváltoztatásával. Végállásra azért van szükség, mert, ha pozicionáláshoz nem használunk abszolút encoder-t, akkor a rendszer csak a bekapcsolásnál lévő pozícióhoz képest tud lépéseket számolni és ilyenkor semmi nem garantálja, hogy mindig ugyanabban a pozícióban lesz bekapcsoláskor a motor. Ha így használjuk sötétítővel, akkor létrejöhet olyan helyzet, hogy a motor kilép a működési tartományból és ebben az esetben sérülhet vagy tönkre is mehet a sötétítő és a motor is. Ebben az esetben sok rendszer használja azt a vezérlési stratégiát, hogy a motort lassan mozgatja egy irányba, amíg egy végállás szenzor nem aktiválódik. Ekkor megváltozik a motor iránya és addig lép, amíg vissza nem tér a működési tartományba. Amint a végállás deaktiválódik, újra biztonságosan vezérelhető a rendszer és ebben az esetben mindig ugyanabból a pontból fog indulni a számolás, ahol már a végállás éppen nincs aktiválva, így kiváltható egy abszolút encoder használata és csak akkor kényszerülünk rá, ha lépéskihagyásra gyanakszunk működés során.

Munkám során 3 fajta végállással találkoztam, melyek itt találhatóak meg [11].

### 2.7.1 A mechanikai végállás

Az első a mechanikai végállás, amely a legegyszerűbb formája a végállásoknak. Ez az eszköz egy kapcsolóból és két vezetékből áll, ahogy a 2.11 ábra is szemlélteti. Ha



2.11 ábra Mikrokapcsoló

Forrás: [11]

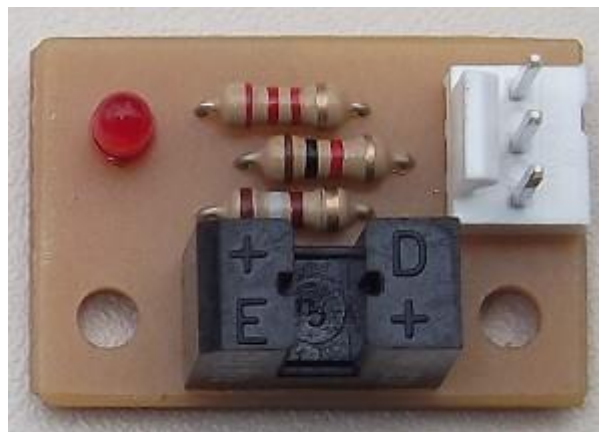


2.12 ábra Reed kapcsoló Forrás: [12]

a kapcsoló benyomódik a végállás kimenete megváltozik és csak akkor változik vissza, ha a kapcsolót felengedik. Általában akkor triggerelik, ha az alkatrész eléri a mozgásteret végét. Mechanikai végállások között is megkülönböztetünk két csoportot: az érintéssel és az érintés nélkül. Az érintéssel mechanikai végállás működésbe hozásához nevéből következik, hogy kontaktra van szükség. Ezeket a végállásokat mikrokapcsolóknak hívjuk. Ezeknek a végállásoknak a hátránya, hogy a jelenlegi sötétítő húzós projektben nem igazán van két olyan mozgó alkatrész, amely fizikai kontaktust hoz létre. Az érintés nélküli mágneses kapcsolókat Reed kapcsolóknak hívjuk. A 2.12 ábrán látható, hogy néznek ki a Reed szenzorok. Ezek a kapcsolók akkor váltanak kimenetet, ha egy mágnes a közelükbe ér. Ez a végállás működőképes lehet, mivel nem kell kontakt a triggereléshez, de hátrányt jelent, hogy túlságosan közel, a cikk szerint 1 mm-en belülre, kell kerülni egy mágnes a Reed kapcsolóhoz. A mechanikai végállásról bővebben itt lehet olvasni [12].

### 2.7.2 Az optikai végállás

A második fajta végállás, amit tanulmányoztam az optikai végállás, amely a 2.13 ábrán megfigyelhető. Ezek az eszközök megfigyelik a fényszintet és hirtelen változás esetén váltanak kimenetet. Akkor működnek jól, ha a fény útjának elzárásakor magas jelet adnak ki a kimenetükön. Összeépítéséhez több dologra van szükség, mint az előző végállásnál. Szükség van egy 180, egy 1000 és egy 2200 Ohm-os ellenállásra, egy piros LED-re, kábelcsatlakozókra és egy fénygátra. Használatkor nekem hátrányt jelentene, hogy nem biztos, hogy a sötétítő alja gond nélkül bemenne a fénygát U betűjének két szára közé és mivel ablakban lenne, valószínű, hogy az ablakon át beszűrődő fény megzavarná a szenzort.



2.13 ábra Az optikai végállás Forrás: [13]

### 2.7.3 A mágneses végállás

A harmadik típusú végállás a mágneses végállás, más néven Hall effect szenzor, amely a 2.14 ábrán látható. A Hall szenzorok jelátalakítóként működnek, amelyek a kimenet feszültségének nagyságát változtatják a közelükben lévő mágneses tér nagyságának függvényében. Használják közelség érzékeléshez, pozicionáláshoz és sebesség méréséhez is. Három csoportba oszthatóak a Hall szenzorok működésük alapján. Az első csoport a bipoláris Hall szenzor, amely megfelelő intenzitású északi vagy déli polaritású mágneses mező érzékelése esetén megváltoztatja kimenetét és addig tartja ezt a kimenetet, amíg egy ellenkező polaritású mágneses mező meg nem változtatja azt. Ezeket angolul „latching Hall sensor”-nak, vagyis reteszkimenettel rendelkező Hall szenzornak hívják. A második csoport az Unipoláris pozitív Hall szenzor, amely kimenete déli polaritású mágneses mező jelenlétében megváltozik, de ennek a megszűntekor nem tartja a jelet, hanem visszavált. A harmadik csoport az Unipoláris negatív Hall szenzor, amely kimenete északi polaritású mágneses mező jelenlétében megváltozik, de ennek a megszűntekor nem tartja a jelet, hanem visszavált. Elektronikai kapcsolóként használva kevésbé jellemző rájuk a mechanikai hiba létrejötte, mivel nem lépnek fizikai kapcsolatba semmivel. Gyengeségük, hogy használhatatlanná válnak, olyan helyeken, ahol alapvetően nagy mágneses tér van jelen. Bekötésük általában eszköztől függ, de általában, ha a felirat velünk szemben van, akkor a bal oldali vezeték a tápfeszültség, a középső a föld és a jobb oldali a jel, amely Unipoláris pozitív Hall szenzor esetében, ha a szenzor nem érzékel mágneses teret, vagy a mágnes északi pólusát érzékeli, akkor logikai 1-et ad ki, de ha a mágnes déli pólusát tesszük közel a szenzorhoz, akkor logikai 0 lesz a kimeneten. A mágneses végállások elemzéséhez a következő tanulmányt használtam: [13].



**2.14 ábra Hall effekt szenzor**

**Forrás: [14]**

## **2.8 Az okos otthon sötétítő vezérlési megoldások felépítésének és funkcióinak vizsgálata**

Az okos otthonok számára motorizált sötétítőkre nem csak DIY (Do It Yourself, tükörfordításban „csináld magad”, saját kezűleg végzett munkára utal) módszerrel lehet szert tenni, ugyanis számos ilyen eszközt lehet venni áruházakban, de akár az internetről is, azonban egyes termékeket tovább lehet fejleszteni házilag DIY módszerrel. A továbbiakban különböző cikkek [14], [15], [16] által legjobbnak, leggazdaságosabbnak és legkörnyezetbarátabbnak vélt termékeket fogom vizsgálni felépítés és funkció szempontjából.

### **2.8.1 A Lutron Serena Okos Sötétítő**

A Lutron Serena Okos Sötétítő erős és halk motort, könnyen használható mobil applikációt és könnyű beállítást ígér, valamint méretre is szabják a sötétítőt a megrendelő számára. Az applikáció számos funkcióval látja el a felhasználókat, mint az állítható ütemezés vagy az energiatakarékos beállítások. Akkumulátorral működik, így a vezetékek nem rontják az esztétikát és az elem a sötétítőt tároló tokban van elhelyezve. Lehet automatizálni, rutinokat létrehozni, vagy akár beszéddel kiadott parancsokkal is vezérelhető. Hátránya azonban, hogy mellé a sötétítő működéséhez további kiegészítőket kell vásárolni. Azonban annak, hogy több cikk szerint is a legjobb termék a piacon és azt mondják róla, hogy a vetélytársai nem érnek fel hozzá, van egy árnyoldala, ugyanis messze az egyik legdrágább termék a maga 600\$-os árával és emellé a kötelező kiegészítők még 150\$-ba kerülnek, így 750\$ körüli árával vonja magára a vásárlók figyelmét.

### **2.8.2 A SmarterHome MySmartRollerShade**

A következő a listában a SmarterHome MySmartRollerShade nevű terméke. Ez a megoldás emberi beavatkozás nélküli töltéssel áll a felhasználóknak a rendelkezésére és ezt úgy éri el, hogy napelem panelek vannak beleépítve a sötétítőbe. Ennek köszönhetően nem kell más energiaforrás a működtetéséhez, azonban valószínűleg ennek tudható be, hogy a motor az átlagosnál lassabban operál. Távirányítóval nem irányítható, csak bluetooth-on keresztül, vagy hangparancsokkal, ezért egy Wi-Fi hub készülék is

szükséges hozzá. Azonban az extra kiegészítő ellenére is meglepően olcsó hasonló kategóriájú társaihoz képest. Kiegészítővel együtt 380\$-ba kerül.

### 2.8.3 Az IKEA Fyrtur

Az IKEA Fyrtur egy olcsóbb megoldás az okos otthonok előnyeit élvezni vágyók számára. Ennél az árucikknél is gondoltak a vezetékek helyettesítésének előnyeire, ezért ebben az esetben is akkumulátorról beszélünk, amely könnyen nyitható kialakításának köszönhetően egyszerű töltést biztosít. Ennél a terméknel viszont nem mi szabhatjuk méretre a sötétítőt, hanem különböző standard méretek vannak, amik közül választani lehet. A vezérlése távirányítóval, beszédparancssal, vagy az áru saját applikációján



**2.15 ábra Az IKEA Fyrtur működés közben**

**Forrás: [15]**

keresztül történhet, azonban az applikáció sokak szerint nem funkcionál megfelelően, így a legtöbben harmadik féltől származó applikációval használják. Bár a termékhez jár távirányító, mégis szükség van az IKEA Dirigera Hub nevű eszközre az okos funkciók használatához. A 2.15 ábrán a terméket láthatjuk működés közben. A sötétítő a kiegészítővel együtt 150\$-ba kerül, ami vetélytársaihoz képest nagyon olcsónak számít.

### 2.8.4 A Yoolax motorizált árnyékoló

A Yoolax motorizált árnyékolója ismét egy jó választás az okos otthonok számára. Ez a sötétítő több népszerű platformmal is kompatibilis. Ezek közé tartozik az Amazon

Alexa és a Google Assistant is. A termék vásárlása esetén opcionálisan beszerezhető egy napelemes akkumulátor töltő is, amely használata esetén nem kell manuális töltéssel bajlódni. A termékhez két motor közül lehet választani. Az egyik a yl23 bluetooth motor, amíg a másik a Z425 „okos akkumulátor” motor. Az első motor használatához szükség van egy bluetooth hubra is, a második viszont kiegészítő nélkül is működtethető. Az applikációján keresztül időzítőket lehet állítani, hogy mikor mennyire húzza le, vagy fel a sötétítőt. Az is beállítható fényérzékelő szenzorok segítségével, hogy napkeltekor és napnyugtakor változzon a sötétítő állása. Ennél a terméknél a vásárló határozhatja meg a kívánt szélességét és magasságát az árunak. Ezek az előnyök azonban azzal járnak, hogy az ára kicsit magasabb, ugyanis a termék 300\$ körüli értékben kapható.

### **2.8.5 A Graywind motorizált napelemes sötétítő**

Az általam utoljára vizsgált termék pedig nem más, mint a Graywind-nek a motorizált napelemes sötétítője. Ez a termék könnyedén használható, minden ismertebb hang asszisztenssel és okos otthon eszközzel. Úgy lett tervezve, hogy a modern technológiai megoldásokkal is működjön, így időzítőn kívül fel van szerelve napkeltét és napnyugtát érzékelő szenzorokkal és mivel programozható is, ezért lehet rá további funkciókat fejleszteni és írni. Egy távirányítóval maximum 15 ilyen sötétítőt lehet vezérelni külön-külön, de akár egységesen is. Ez a termék 150-200\$-ig terjedő értékben található meg az áruházakban vagy online webáruházakban.

Kiemelném a fent említett funkciók közül a napelemes töltést, melynek segítségével nem szükséges vezetékeket használnunk és a naplemente és napkelte érzékelésére használt fényérzékelő szenzorokat, amelyek remek ötletek lehetnek továbbfejlesztésre.

A termékek részletesebb leírása a hivatkozásokban érhető el. [14], [15], [16]

## **3 Tervezés**

### **3.1 A motor kiválasztása**

A tervezésem kezdetén tudni szerettem volna, hogy milyen alkatrészek lesznek szükségesek a kapcsolás létrehozásához. A legfontosabb alkatrész, vagyis a motor kiválasztása volt az első számomra. Itt két motor jött szóba: a 28BYJ-48 már a bipoláris módosítással és a NEMA17 17HS3001-20B. Úgy döntöttem, hogy a NEMA17 jobb választás lesz számomra, mert mivel nagyobb a nyomatéka, így nagyobb erőt lesz képes kifejteni a sötétítőn. Más szavakkal robusztusabb, mint a 28BYJ-48 és mivel a NEMA17 hibrid motor a másik pedig bipoláris, így a NEMA17 precízebb lesz a munka során, ezért egyenletesebben és pontosabban fog működni.

### **3.2 A driver kiválasztása**

A drivert nem volt bonyolult kiválasztani, ugyanis a műhelyben két drivert kaptam. Az egyik az ULN2003A, a másik pedig az A4988 volt. Mivel az ULN2003A csak unipoláris motorokat képes meghajtani és én egy hibrid motort választottam, ezért kénytelen voltam az A4988 mellett letenni a voksomat, mivel ez bipoláris és hibrid motorokat is képes meghajtani.

### **3.3 A mikrokontroller kiválasztása**

A mikrokontroller kiválasztásánál a konzulensem ajánlását megfogadva döntöttem el, hogy melyiket fogom használni a projektem során. Ez pedig nem volt más, mint az ESP-12F. Mivel nem volt más lehetőségem, így ezt választottam, de ha visszamehetnék és lenne más lehetőségem, akkor is ezt választanám, mert egy egyszerű mikrokontroller és könnyen működtethető.

### **3.4 A programozáshoz használt szoftver kiválasztása**

A szoftver kiválasztásában a döntésemet több ember véleményére alapoztam. Megkérdeztem a konzulensemét és egy a témában jártas ismerősömet, hogy mit ajánlanának a programozáshoz és mindketten az Arduino IDE-t ajánlották. Továbbá különböző fórumoldalakon is megnéztem mások véleményét és mindenki ezt javasolta és



tartotta a legjobb döntésnek a feladathoz. Az egyöntetű vélemény az volt, hogy egyszerűen működtethető és jól átlátható a felülete, így ezt választottam.

### **3.5 A pozíció meghatározásának kiválasztása**

A pozíció meghatározására három módszert ismertem meg, amiket korábban már leírtam. Ezek név szerint a „kódban számolás módszer”, az „abszolút encoder” és az „ellenáramerősség-érzékelés módszer”. Ezek közül az abszolút encoder használata esetén olyan problémákba ütközünk, hogy drága, sok helyet foglal és még a rendszert is jóval komplexebbé teszi, így nem feltétlenül ez a legjobb megoldás, ami szóba jöhet. Az ellenáramerősség-érzékeléshez szükségünk lenne ilyen áramerősség-szenzorokra és egy szabálytalan póluselosztású rotorra. Ezeknek a szenzoroknak a használata bonyolult és rendkívül időigényes, továbbá egy ilyen rotor beszerzése sem lett volna gyors és egyszerű. Ha fél évnél több idő állna rendelkezésre a megoldás megvalósítására, akkor ez jó megoldás lehetne, de így sajnos el kellett vetnem ezt a lehetőséget. Utolsó megoldásként maradt a „kódban számolás módszere”, de ezzel a módszerrel, mint már fentebb leírtam, olyan is előfordulhat, hogy tönkremegy a motor, vagy a sötétítő. Azonban végállás használatával ez elkerülhető és így már tökéletes döntésnek tűnik, mivel egyszerűen kivitelezhető és a végállás megvédi a motort és a sötétítőt egyaránt, valamint az a probléma sem jöhet elő, hogy a motor kihagy lépéseket, mert az általam választott motor erős és könnyedén meg képes birkózni a függöny mozgásánál fellépő erőhatásokkal.

### **3.6 A végállás kiválasztása**

A végállás kiválasztásánál összesen 4 lehetőségem volt. Ezek közül kettő mechanikai végállás: a mikrokapcsoló és a Reed kapcsoló. Egy optikai végállás és egy mágneses végállás: a Hall szenzor. Először a mikrokapcsolót szerettem volna használni, mivel működése rendkívül egyszerű, azonban fizikai kontaktust igényel és ezt a sötétítő mozgatása közben nem tudtam volna biztosítani, ugyanis, ha felrögzítettem volna valahová, akkor sem lett volna semmi, ami megnyomja a kapcsolóját, így be kellett látnom, hogy más végállás kell a rendszerembe. Ezután megvizsgáltam az optikai végállást. Ennek a működését úgy képzeltem el, hogy a szenzort a sötétítő tetejénél rögzítem és az eszköz U alakú részébe helyezem a sötétítőt. Amikor a szenzor nem látja a másik oldalt a sötétítőtől, akkor normálisan működik, azonban, ha túlhúzódik a sötétítő, akkor megáll a felhúzás és a rendszer visszatér alapvető működési tartományába. Ezzel a

tervvel több probléma is felmerül. Először is a szenzor U alakú része nagyon keskeny, ezért a sötétítő alján levő vastagabb rész nem férne keresztül rajta és összeütközne vele, amitől a szenzor könnyen elromolhat. Szintén az U alakú rész keskenysége miatt van az a probléma is, hogy ha a sötétítő kilép a működési tartományból, akkor lehetséges, hogy visszatéréskor nem „talál bele” az U alakú részbe és ez anyagi károkhoz vezetne, mivel ekkor addig húzná a motor lefelé a sötétítőt, amíg az egyik el nem romlik. A harmadik probléma az a szenzorral, hogy fényt érzékelve működik. Mivel a sötétítő valószínűleg az ablakban lesz, ezért könnyen lehet, hogy bezavarja a szenzort. Ebben az esetben a motor ugyanúgy addig húzná lefelé a sötétítőt, amíg egyikük el nem romlik. A Reed kapcsoló, bár mechanikai végállás, mégis mágneses úton működik. Ez a megoldás jónak tűnik, mivel nincs a környezetében, ami megzavarná. Egyetlen hátránya az, hogy nagyon közel, 1 milliméteren belülre kell kerülnie egy mágnesnek a kapcsolóhoz ahhoz, hogy átbillentse. Ez a megoldás megvalósítható, de nagyon körülményes, mivel, ha túl távol van a mágnes, akkor ahogy fentebb leírtam, vagy a sötétítő, vagy a motor romlik el. Ha túl közel van a mágnes, akkor összeérhetnek és kontaktusnál elromolhat a kapcsoló. Mivel itt 1 millimétről beszélünk, ekkora eltérés két húzás között megtörténhet. Így elérkeztünk utolsó végállásunkhoz: a Hall szenzorhoz. Ennél az eszköznél azt találtam ki, hogy a mágneset a sötétítő aljára erősítem, a sötétítő tetejénél pedig elhelyezem a Hall szenzort. Itt nagyobb távolságból képes érzékelni a mágneses teret az eszköz így a Reed kapcsolónál felbukkanó problémák megoldódnak. Amint érzékeli a szenzor a mágneset, a sötétítő visszatér a működési tartományába. Ugyanez történik minden indításkor automatikusan. Mivel itt semmi olyan problémára nem tudtam gondolni, ami anyagi kárt jelentene és emellett ez a végállás megfelelően elvégzi a dolgát, ezért ezt a megoldást választottam és vásároltam egy Unipoláris pozitív Hall szenzort.

### **3.7 A működést segítő további alkatrészek választása**

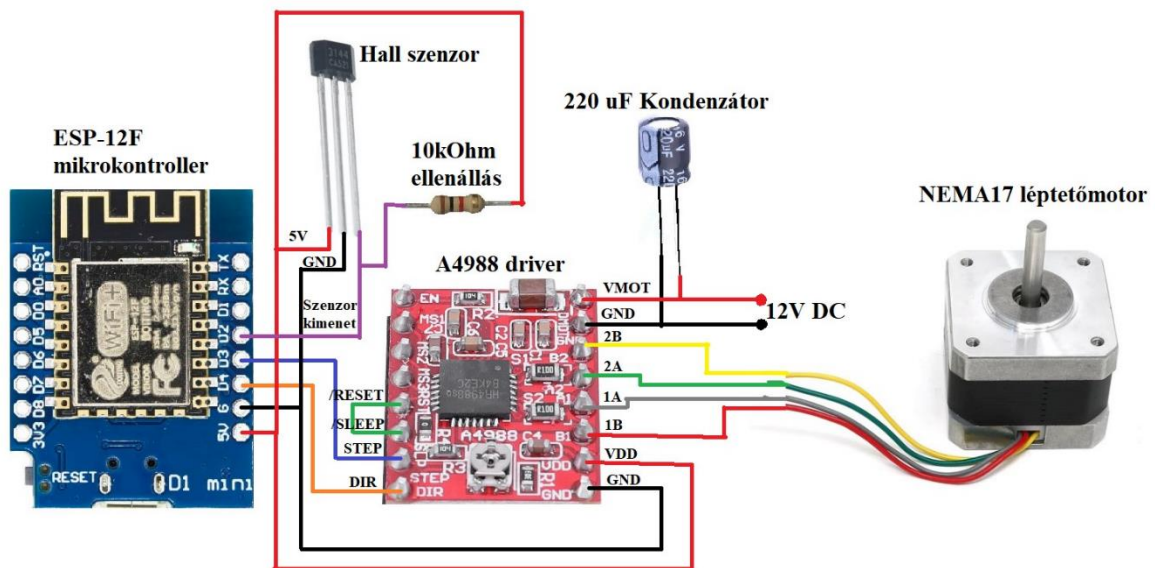
Információgyűjtés céljából az interneten megnéztem néhány oldalt, amelyen hasonló áramkört építenek össze és megvizsgáltam mikre lesz még szükségem a rendszer megépítéséhez. A motor feszültség alá helyezéséhez szükségem volt egy 12 V-os adapterre. Továbbá egy 220 uF-os kondenzátorra, hogy megvédje a drivert a feszültségtüskéktől. Szükségem volt még egy breadboard-ra és vezetékekre az összekötésekhez, valamint egy power jack DC csatlakozóra ahhoz, hogy az adaptert a breadboardba köthessem. Végül szükségem volt áttételre is, hogy a motor biztosan gondmentesen felhúzza a sötétítőt és kellett 3D nyomtatott fogaskerék is, ami abban volt

egyedi, hogy a sötétítő gyöngysorára volt méretezve. A gyöngysor és a motor paramétereinek lemérésére szükségem volt egy tolómérőre, az áramerősségi határérték beállításához pedig egy multiméterre is. Ezekkel az alkatrészekkel a konzulensem és a műhely támogatták a munkámat.

### 3.8 A rendszerem összeszerelésének megtervezése

Miután összegyűjtöttem, a szükséges alkatrészeket a rendszerem megépítéséhez, az a fázis következett, hogy megtervezzem, hogyan legyenek összekötve és összeszerelve az alkatrészek. Végül a 3.1 ábrán látható tervet hoztam létre. A továbbiakban ennek kitalálása során hozott döntéseim magyarázata következik.

A tervem középpontjában a driver volt, mivel ez hajtja majd meg a motort. Először összekötöttem a driver jobb alsó GND pinjét a mikrokontroller föld pinjével, hogy a két pin azonos potenciálon legyen. Ezután a driver VDD pinjére rákötöttem a mikrokontroller



3.1 ábra A rendszerem összeszerelési terve

Forrás: [16], [17], [18], [19], [20], [21]

5 V pinjét, hogy ez a két pin is azonos potenciálon legyen és ezáltal a két alkatrész azonos feszültségen operáljon. Ezután az 1A, 1B, 2A és 2B pinekre úgy kötöttem rá a motorban lévő tekercsek végződésének vezetékét, hogy az 1A és 1B az egyik tekercs két végére csatlakozzon, a 2A és 2B pedig a másik tekercs két végére. Az én motoromon, ahogy az ábrán látható motoron is volt egy piros, egy sárga, egy zöld és egy szürke vezeték. A motor adatlapjáról megállapítottam, hogy a sárga és zöld vezeték csatlakozik az egyik

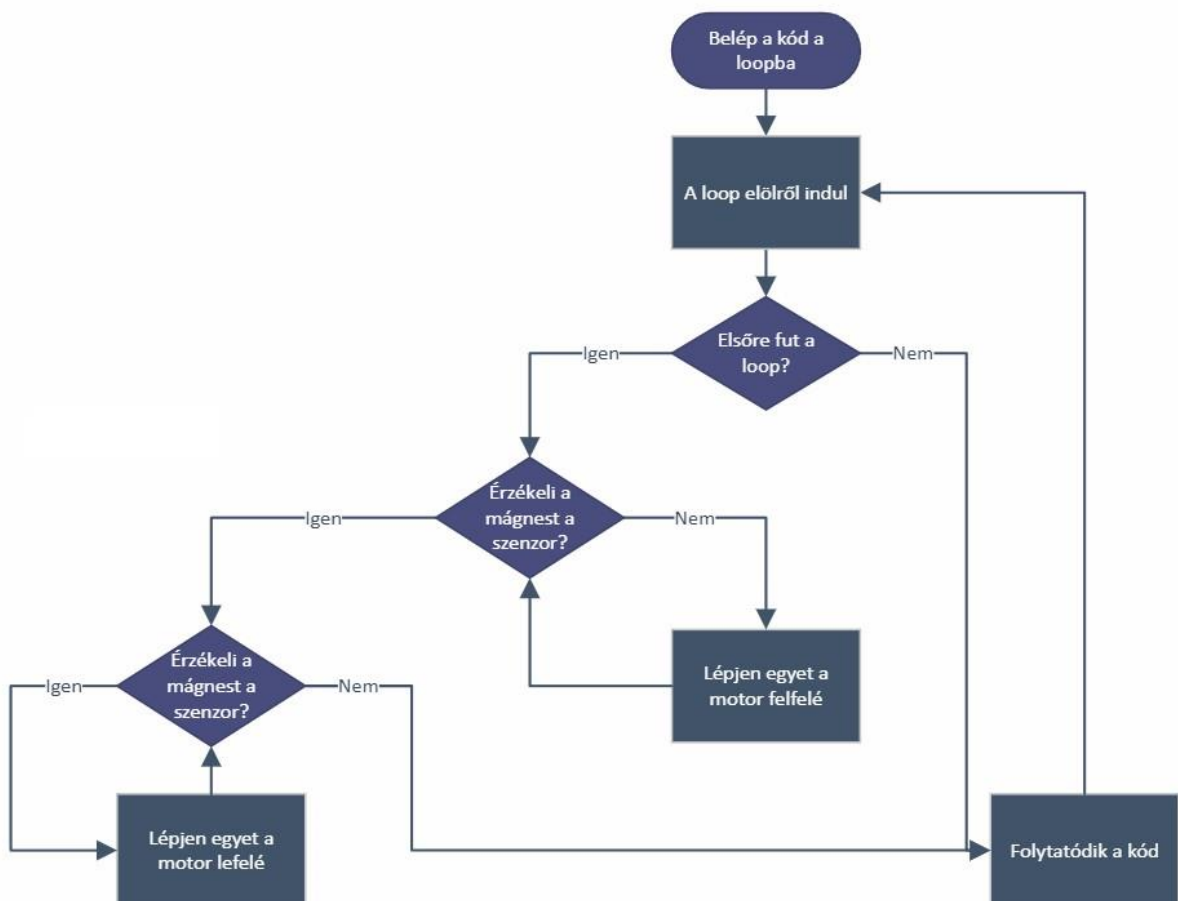
tekercsre, a szürke és piros vezeték pedig a másik tekercsre a motorban. Ez alapján az információ alapján az ábrán látható módon csatlakoztattam a motor vezetékeit. Ezután a motor tápellátását kellett megoldanom, mivel a motornak nem elég a mikrokontroller pinjéből szerezhető 5 V. A motor működéséhez 8-35 V-os tápfeszültségre van szükség, ezért egy adapter által biztosított 12 V feszültséget egy power jack DC csatlakozó segítségével két vezetéken keresztül adtam rá a VMOT és GND pinekre, így biztosítva a motor tápellátását. A két pin közé beraktam egy 220 uF-os kondenzátort, hogy megvédje a drivert az esetleges feszültségtüskéktől. A driver másik oldalán a /reset és a /sleep pineket összekötöttem, hogy felső állásban legyen a két pin és működés közben ne kerüljön át alvó állapotba, vagy induljon újra. A STEP pint rákötöttem a mikrokontroller D3-as kimenetére és a DIR pint a mikrokontroller D4-es kimenetére, mert a kódban ezeket a kimeneteket definiáltam úgy, hogy a STEP és DIR pinekre csatlakoznak. Ekkor már csak a Hall szenzor bekötése volt hátra. A Hall szenzornak három pinje van, ezek balról jobbra a VCC, a GND és a digitális kimenet. A VCC-re rákapcsoltam a mikrokontroller 5 V pinjét, a GND-re pedig a mikrokontroller Ground pinjét, így 5 V-tal táplálva a Hall szenzort. A szenzor digitális kimenetét a mikrokontroller D2-es pinjére kötöttem, mert a kódban ezt a pint definiáltam a Hall szenzortól kapható adatok fogadására. Így amíg egy mágnes déli polaritású mágneses mezejét érzékeli a szenzor, addig a D2-es pin „LOW”, vagyis alsó állapotba kerül és ha elvesszük a mágnest, akkor visszakerül felső állapotba. Összekötöttem a Hall szenzor 5 V-os és digitális kimeneti pinjeit és közéjük helyeztem egy 10 kOhm-os ellenállást. Erre azért volt szükség, hogy a szenzor kimenetét felhúzzuk 5 V-ra, tehát felső állásba. A terv létrehozásában a következő cikk segítette a munkámat: [17].

### 3.9 A kódom terveinek bemutatása

Ebben a fejezetben a „void loop(){...}” függvényben található folyamatokat fogom bemutatni.

A függvény egy „if”-fel kezdődik, aminek a feltétele, hogy u legyen 0. Fontos tudni, hogy u egy olyan globális változó, amelyet létrehozásakor 0-val tettem egyenlővé. Erre azért van szükség, hogy ez az „if” csak egyszer fusson le minden áramtalanítás után, mert ebben a részben húzzuk fel a sötétítőt a kezdő pozíciójába, amely minden indítás után ugyanaz lesz. Amíg a sötétítő el nem éri a kezdőpontját, addig nem lehet beavatkozni WiFi-n keresztül a sötétítő mozgásába, így véletlenül sem lehet olyan, hogy más

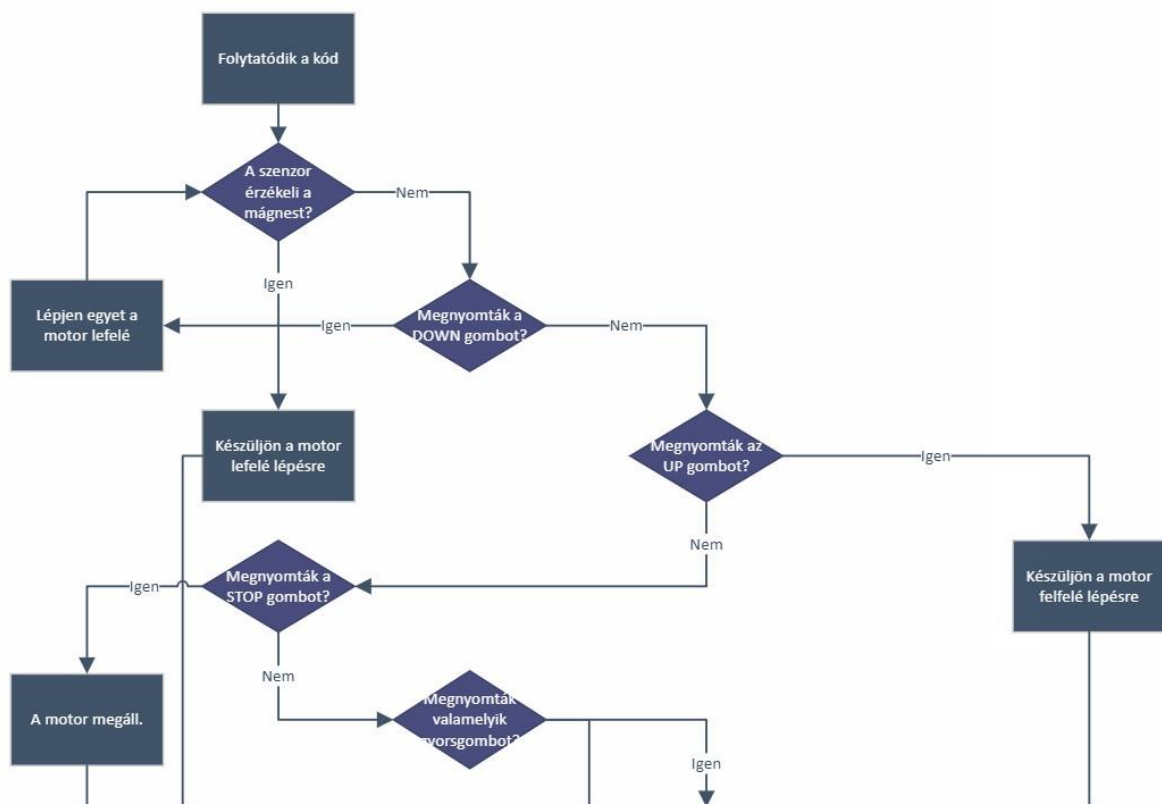
pozícióból indul. Ha először fut a loop függvény bekapcsolást követően, akkor egy „hall” nevű változóban eltárolom a Hall szenzor kimenetén lévő értéket. A driver „dir” pinjét magas állásba teszem. Ezután egy „while” ciklus következik, ami addig fog futni, amíg a „hall” változó magas állásban van, vagyis nem érzékel a szenzor elég nagy mágneses teret. A cikluson belül 1-et lépek a motorral és frissítem a „hall” változó értékét. Amint a szenzor érzékeli a mágnest, kimenetét alacsony állásba teszi, ezért a „hall” változó is alacsony lesz, így nem fut tovább a „while” ciklus. Ekkor megváltoztatom a „dir” pin kimenetét alacsonyra és újabb „while” ciklus kezdődik, amiben 1-et léptetek a motoron és addig tart, amíg visszatér a motor a működési tartományba, vagyis a szenzor ismét nem érzékeli a mágnest. Amint véget ér a „while” ciklus az u változót 1-el teszem egyenlővé, hogy többet ne fusson le ez a rész. A 3.2 ábra szemlélteti a fentebb leírt kódrészletet.



**3.2 ábra A loop első részén lévő automatikus pozícióba mozgatás szemléltetése**

Ezután egy biztonsági funkció következik, amely minden lépés után megvizsgálja, hogy a szenzor érzékeli-e a mágneket, mert ha igen, az azt jelenti, hogy a motor valahol kihagyott egy lépést és kifutott a működési tartományból, ezért lépésenként vissza kell léptetnem. Ezt a funkciót azért hoztam létre, hogy meghibásodás esetén se essen kár a motorban, vagy a sötétítőben, de szerencsére gyakorlatban nem volt rá szükség. A 3.3 ábrán a biztonsági funkciót szemléltetem, valamint a gombok működési algoritmusának első fele is látható.

Itt kezdődik a gombok általi vezérlés egy „if”-fel. Ha a DOWN gombot megnyomjuk, akkor a driver dir pin-je alacsony állásba kerül és egy számláló segítségével

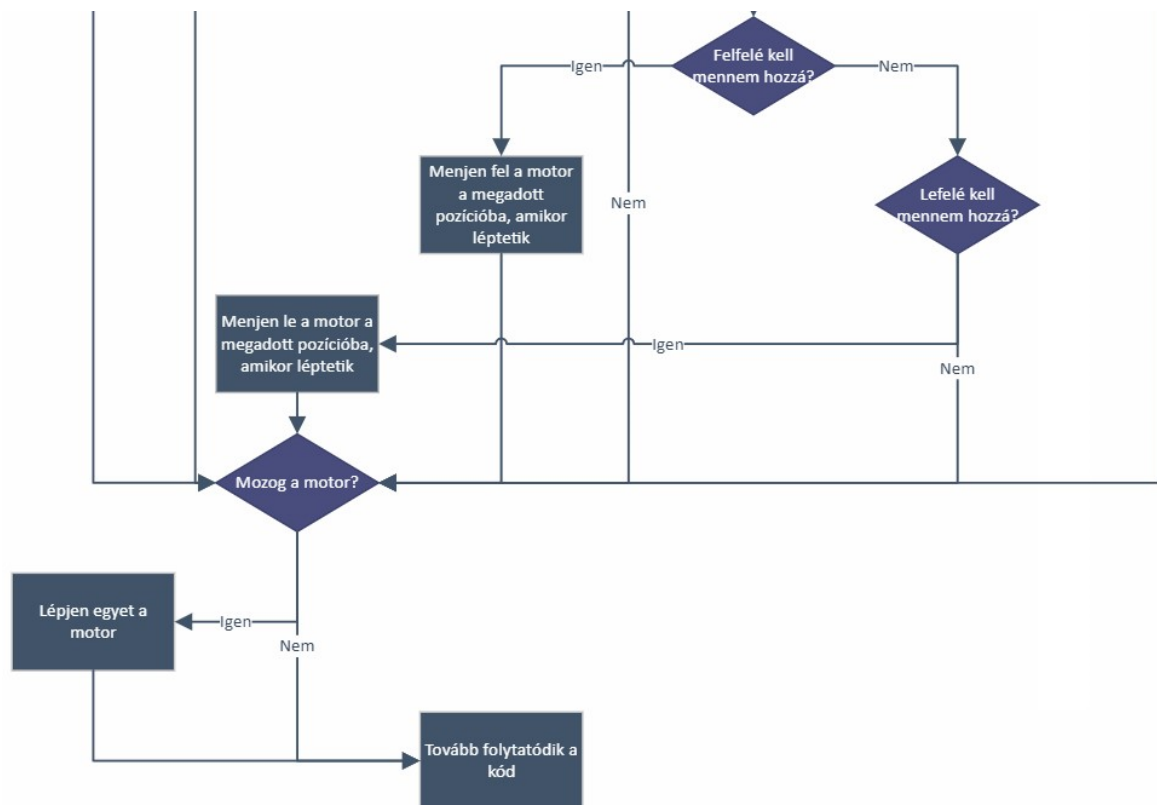


**3.3 ábra A biztonsági funkció és a gombok működési algoritmusának első része**

megadom a motornak, hogy mennyi lépést kell megtennie, amíg a sötétítő teljesen lehúzódik. Beállítom az állapotot tároló változókat, hogy eltárolják az általam kívánt állapotot: a sötétítő nem áll, hanem mozog, nem használunk gyorsgombot és a sötétítő lefelé mozog. Az „if” után egy „else if” következik, amiben, ha az UP gombot nyomjuk meg, akkor a dir pin-t magas állásba teszem és a teljes húzáshoz szükséges lépésszámból kivonva a pozíciónkat megkapjuk, hány lépést kell megtenni, hogy a sötétítő a felső állásába érkezen. Ahogy előzőleg, itt is beállítjuk az állapotváltozókat, annyi

különbséggel, hogy itt azt tároljuk el, hogy felfelé húzzuk a sötétítőt. Ezután újabb „else if” következik, ami azt vizsgálja, hogy megnyomtam-e a STOP gombot. Ha igen, akkor megáll a motor és az állapotváltozókat frissítem. Egyben eltárolom, hogy álló helyzetben van a motor és egyben eltárolom, hogy nem használok gyorsgombot.

Ezután a gyorsgombok kódrészlete fut le. Egy „else if”-fel kezdődik, amiben azt vizsgálom, hogy valamelyik gyorsgomb be lett-e nyomva. Ha igen, akkor az állapotváltozót, amely a gyorsgomb használatát tárolja frissítem arra, hogy gyorsgombot használok. Ekkor egy „if”-ben megvizsgálom, hogy a jelenlegi pozícióm kisebb-e, mint



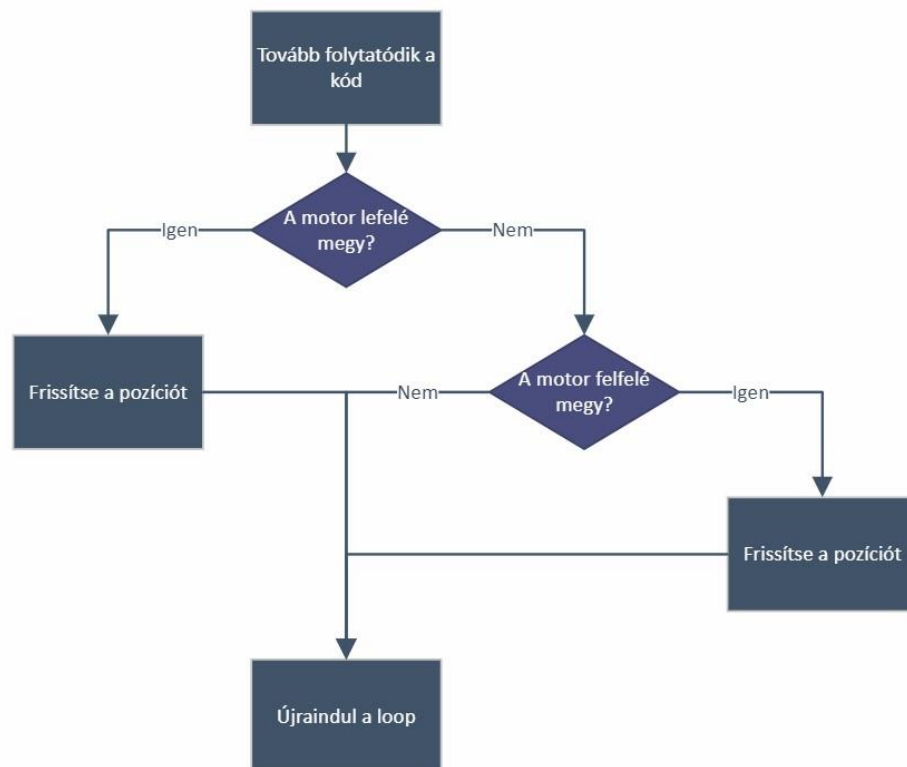
az a pozíció, amit a gyorsgombokkal kiválasztottam. Ha igen, akkor a dir pin-t magas

### 3.4 ábra a gombok működési algoritmusának második része folyamatábrán feltüntetve

állásba teszem és a gyorsgomb által megadott százaléérték, a teljes húzáshoz szükséges lépésszám és a jelenlegi pozíció segítségével a program kiszámolja, hogy hány lépést kell tennie a motornak, hogy a kiválasztott pozícióba kerüljön és megadja ezt a lépésszámot a motornak. Ekkor frissülnek az állapotváltozók és eltárolják, hogy nem áll a motor és felfelé megy. Az „if” után jön egy újabb, ami azt vizsgálja, hogy a jelenlegi pozíció nagyobb-e, mint ahova a gyorsgombokkal beállítottam. Ha igen, akkor ugyanaz történik, mint az előző „if”-ben, csak lefelé fogja húzni a redőnyt. A 3.4 ábrán a gombok működési algoritmusának második része látható folyamatábrán feltüntetve. A gyorsgombokkal

érhető el, hogy a sötétítőt köztes állapotokba vezéreljük. Ezzel a terveim szerint a teljes mozgástér 25, 50 és 75 százalékára tudunk navigálni a gyorsgombokkal. A gyorsgombok és a gyorsfunkciók számának egyedül a sötétítő teljes felhúzásához szükséges motor által lépések száma szab felső határt.

Ekkor egy „if” megnézi az állást tároló állapotváltozót és ha a motor a változó alapján nem áll, akkor léptet rajta egyet. Ezután jön a kódnak az a része, amiben frissül a motor jelenlegi pozíciója. Ez a rész egy „if”-fel kezdődik, amely megvizsgálja, hogy amióta utoljára lefutott az „if”, azóta lépett-e a motor, hogy az állapotváltozó alapján mozog-e a motor és hogy az irányt tároló állapotváltozó alapján lefelé fut-e. Ha ez mind igaz és az állapotváltozó szerint gyorsgombot használok, akkor egy „if”-en belül frissítem a pozíciót és az előző pozíciót tároló változót is. Ha nem használtam gyorsgombot, akkor ugyanez történik, csak máshogy kell számítani a pozíció eltárolására használt változó értékét. Jön egy „if”, amiben ugyanazt a három feltételt vizsgálom, mint az előbb, de most akkor fut le az „if”, ha az állapotváltozó szerint felfelé megy a sötétítő. Ebben az esetben is külön frissítem a pozíciót és az előző pozíciót tároló változót is attól függően, hogy a gyorsgombokat használtam-e az állapotváltozó szerint. Ezután újraindul a loop és kezdődik az egész előlről. Ezt mutatja be a 3.5 ábrán látható folyamatábra is.



**3.5 ábra A pozíció frissítésének kódjáról folyamatábra**



### 3.10 A webszerver terveinek bemutatása

A terveim szerint a motort egy weboldaltól lehet majd vezérelni, de a weboldal eléréséhez webszerverre lesz szükségem, amelyet a mikrokontroller biztosít a számomra. A weboldal terveiről a következő pontban lehet majd olvasni. A WiFi-re csatlakozáshoz létrehozok két konstans globális változót és az egyikbe a WiFi nevét a másikba pedig a jelszavát írom. Létrehozom a webszervert a 80-as porton, majd létrehozok egy „notFound()” nevű függvényt, amely, ha nem sikerült elérni a webszervert visszajelez, hogy nem találta meg. Majd a „setup()” függvényben beállítom a WiFi módot és elindítom a WiFi-t a globális változók adatainak segítségével. Létrehozok egy „if”-et, hogy ha nem tud csatlakozni, akkor Serial Monitoron jelezze nekem és térjen vissza. Ha viszont sikeres a kapcsolat, akkor ezt is jelezze nekem a Serial Monitoron és írja ki a weboldal IP címét is. A kliens és a szerver között létrejön egy TCP (Transmission Control Protocol) kapcsolat. A kliens a szerverhez kapcsolódik a szerver IP-címével és a használt porttal. Ezután a kliens HTTP (Hypertext Transfer Protocol) kérést küld a szervernek, amire HTTP válaszban a webszerver elküldi a weboldalt a kliensnek. A weboldalon minden gomb megnyomása után kap a szerver egy HTTP GET kérést, amit, ha megkap a szerver, akkor egy loopban használt globális változót megváltoztatok és válaszként küld a szerver egy „ok” üzenetet a kliensnek, hogy megkapta és feldolgozta a kérést. Miután a válasz elküldésre került a TCP kapcsolatot lezárják. A „setup()” végére szükséges két sor. Az egyikben, ha nem találta meg a szervert, meghívom a már fentebb említett „notFound()” függvényt. A következő sorban pedig elindítom a webszervert.

### 3.11 A weboldal terveinek bemutatása

Terveim szerint a weboldalt majd egy konstans változóban tárolom és amikor a kliens elküldi az első kérést, arra válaszul a szerver elküldi a változót, amiben az oldalt tárolom. A weboldal kinézetét úgy képzelem el, hogy 6 egységesen kinéző gomb van két sorban és feliratok különböztetik meg őket egymástól. Az első sorban helyezkedik el a felhúzó, lehúzó és megállító gomb. A második sor a gyorsgomboké, így itt található a 25, 50 és 75 százalék gomb, amelyekkel a köztes állapotok érhetők el. A gombok lenyomásakor szeretném, ha a kliens elküldené a kérést a szervernek és felengedéskor nem történne semmi. Valamint azt is szeretném megvalósítani, hogy amikor az egér a gomb fölé érkezik megváltozzon a gomb színe. Szeretném, hogy a weboldaltól vezérelhető legyen a motor, attól függetlenül, hogy éppen milyen eszköztől vezérlem.

Irányítható legyen PC-ről, laptopról és akár mobiltelefonról is. Továbbá ne függjön a működés attól, hogy éppen melyik böngészőt használom, tehát helyes működést várok el Microsoft Edge-ben, Google Chrome-ban, Mozilla Firefox-ban, vagy éppen Operában.

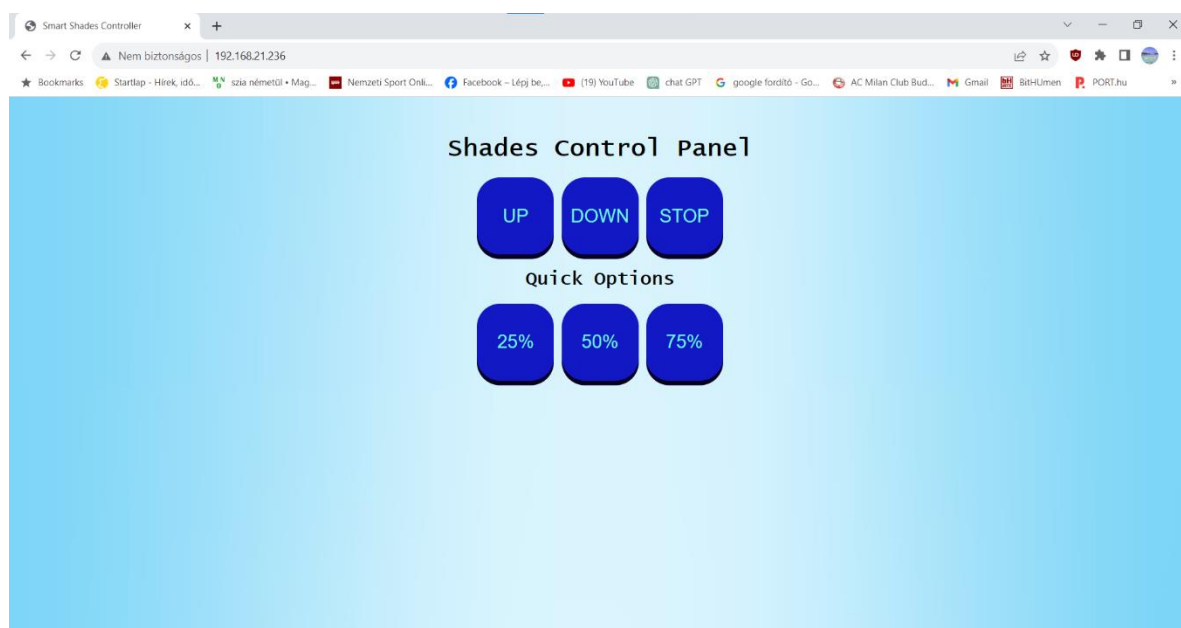
## 4 Implementálás

### 4.1 A hardver összeszerelése

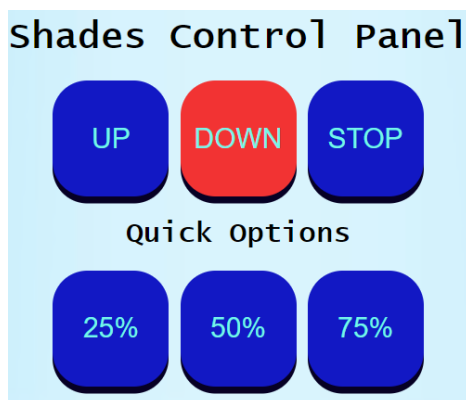
Úgy kezdtem el a rendszerem megvalósítását, hogy a 3.8 fejezetben megtervezett rendszert megépítettem, de még a Hall szenzor nélkül. Az alkatrészek megfelelő összekötése után elkezdhettem a NEMA17 motor kipróbálását. A próbákhoz az interneten talált egyszerű teszt kódokat használtam [18], melyeket kisebb-nagyobb módosításokkal is teszteltem. Miután a tesztek során megfigyeltem, hogy a motor megfelelően hajtja végre a mikrokontrollerre feltöltött kódokat megbizonyosodtam róla, hogy az összeszerelési tervem helyes volt és az összeköttetések megteremtésekor sem követtem el hibát. Ezután elkezdtem megírni a megtervezett kódot.

### 4.2 A kód és a weboldal elkészítése

A tervezés során kitalált algoritmusomon végül szinte semmit sem kellett változtatnom, bár a 3.2-es ábrán látható rész és a 3.3-as ábra első fele ekkor még nem került megírásra, ugyanis ekkor még nem volt rendelkezésemre álló Hall szenzor.



4.1 ábra A sötétítő vezérlő panel



**4.2 ábra** Kurzort a gomb fölé helyezve megváltozik a gomb színe

Ezután megírtam a weboldalt is, amelyhez fel kellett frissítenem kicsit a HTML-ről és CSS-ről tanult ismereteimet. Végül a weboldal is úgy nézett ki, ahogy megterveztem. A weboldal és a kód megírásában nagy segítséget nyújtottak a következő oldalak: [19], [20], [21], [22]. A 4.1 ábrán a kész sötétítő vezérlő panel látható. Kinézetén a tervekhez képest annyit módosítottam, hogy színessé tettem a gombokat és a hátteret is, mert kissé unalmas felület volt fehérén, de szerintem a változtatás elérte a kívánt hatást. Továbbá azt is sikerült megvalósítanom, hogy a kurzort a gomb fölé húzva megváltozzon az adott gomb színe. Ezt a 4.2 ábrán szemléltetem. Sajnos nem látszik jól, mert a kurzort eltávolítja a képről a képernyőkép készülőkor.

### 4.3 A végállás és a hozzá tartozó kód megvalósítása

A weboldal elkészülte után már meg tudtam nézni, hogy a motort meg tudom-e forgatni a weboldal használatával és a terveimnek megfelelően működött minden funkciója. Azonban ekkor még nem volt a rendszeremben végállás, így a konzulensemmelel való megbeszélés után arra jutottunk, hogy a mágneses végállás lenne a legjobb megoldás a rendszerembe, ezért vettem egy Hall szenzort. Miután leteszteltem, hogy működik-e bekötöttem a rendszerembe és megírtam hozzá a kódot, amelynek algoritmusát a 3.2 ábrán és a 3.3 ábra első felén lehet látni. Az újonnan megírt kód és a végállás is remekül működött, azonban volt egy kisebb probléma.

A mágnesnek, amelynek ekkor birtokában voltam, túl kicsi volt a mágneses mezeje, így ahhoz, hogy a szenzor érzékelni tudja, túlságosan közel kellett hozzá vinni. Ez pedig azt a problémát vetette fel, hogy mivel így a szenzor nehezen érzékeli a mágneset, mi lesz, ha egyszer nem érzékeli és tönkremegy a motor. Ezt mindenképpen meg kellett előzni és első gondolatom az volt, hogy szükségem lesz egy erősebb mágnesre.

Szerencsére kaptam kölcsönbe egy az előzőnél jóval erősebb mágnezt. Ezt a mágnezt a szenzor nagyjából 20-25-ször távolabbról is érzékelte, így a végállás is készen állt arra, hogy élesben kipróbáljam. A végállás bekötéséhez a következő cikkből informálódtam: [23].

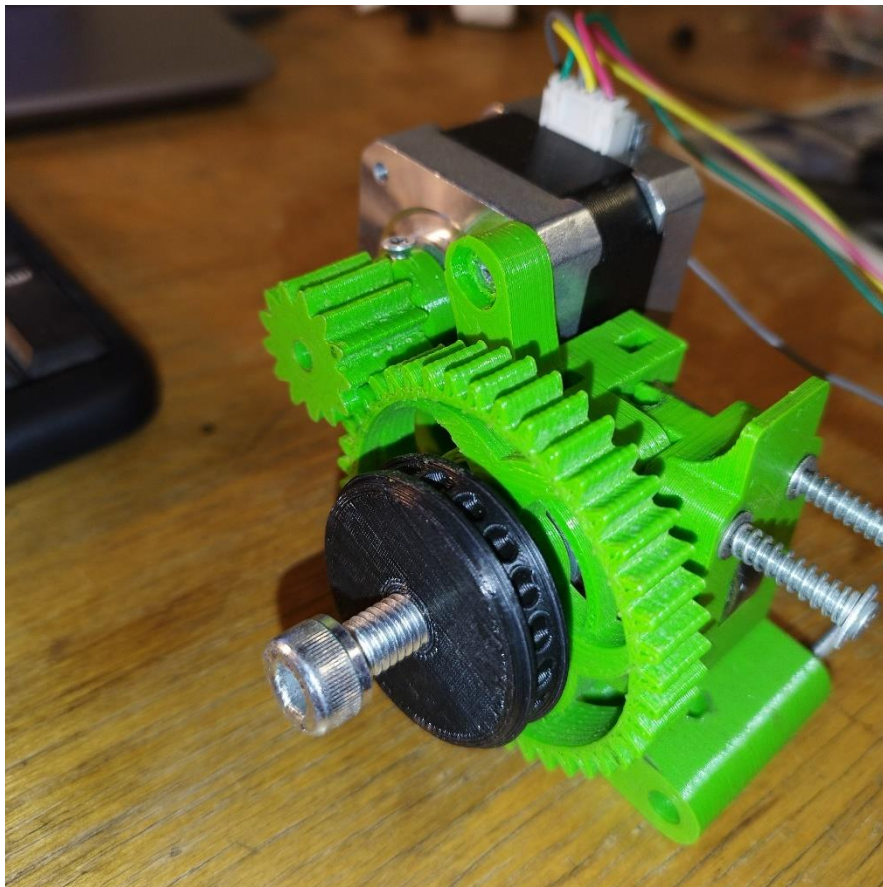
#### 4.4 A 3D nyomtatott fogaskerék

Egy valamire volt szükségem ahhoz, hogy gyakorlatban is kipróbálhassam a motort. Ez nem volt más, mint egy fogaskerék. Egy olyan speciális fogaskerékre volt szükségem, amelynek a közepén lévő lyukba tökéletesen illeszkedik a motor D alakú shaftja, vagyis tengelye és amelynek kerületén lévő félgömb alakú lyukakba illeszkednek a sötétítő gyöngyláncának gyöngyei. A konzulensem, amikor közöltem vele, mire van szükségem, felajánlotta, hogy a műhelyben található 3D nyomtató segítségével nyomtat egy ilyen fogaskereket nekem. Azonban ahhoz, hogy ki tudjuk nyomtatni, tudnunk kellett mind a motornak, mind a gyöngyláncnak a pontos paramétereit. Tudnom kellett a gyöngyláncról, hogy mekkora egy gyöngy átmérője és milyen hosszú az a szakasz, ami két gömböt összeköt. A motor tengelyéről is két adatot kellett tudnom. Mivel nem szabályos kör alapú, hanem D alapú, ezért szükségem volt a legnagyobb és legkisebb átmérőjére. A legkisebb átmérőjét, úgy mértem le, hogy a tolómérő egyik szárát a D egyenes oldalának közepéhez tettem, amíg a másikat a D másik oldalára. A legnagyobb átmérőt, úgy mértem le, hogy az előző méréshez képest elforgattam a tolómérőt 90 fokkal a tengely körül. A legkisebb átmérő hossza szerepelt a motor adatlapján is, így azt csak azért mértem le, hogy biztosra menjek. Mivel nekem ahhoz, hogy meg tudjam tervezni a fogaskereket a sugarára volt szükségem, ezért először meghatároztam, hogy mekkora kerületű fogaskereket akarok. Úgy döntöttem, hogy 10 golyó nagyságú elég lesz. Így a kerület a golyó átmérőjének és a két golyó közötti távolság összegének 10-szerese. A kerületet elosztva 2-szer pi-vel megkaptam a tervhez szükséges fogaskerék sugarát. A konzulensem segítségével megadtuk a szoftvernek a megfelelő paramétereket és kinyomtattuk a fogaskereket. A fogaskereket ráhúztam a motor tengelyére és a fogaskerékre rátettem a gyöngyláncot. Azonban a motor nem bírta felfelé húzni a sötétítőt. Először azt hittem, hogy a számítások során rontottam el valamit és a lyukak nem pontosan ott vannak, ahol a gyöngyök, de nem ez volt a baj. Nem volt elég nagy a motor nyomatéka ahhoz, hogy felhúzza a sötétítőt. Ekkor két ötletem támadt a probléma megoldására. Az első, hogy áttételt alkalmazok, hogy nagyobb erőt tudjon kifejteni a motor. A másik pedig az volt, hogy beszerелек még egy motort és együttes erővel húzzák

fel a sötétítőt. Végül az első megoldást valósítottam meg, mert a másodikban nem voltam 100 százalékgig biztos, hogy működne-e és a konzulensem is inkább az első megoldást ajánlotta.

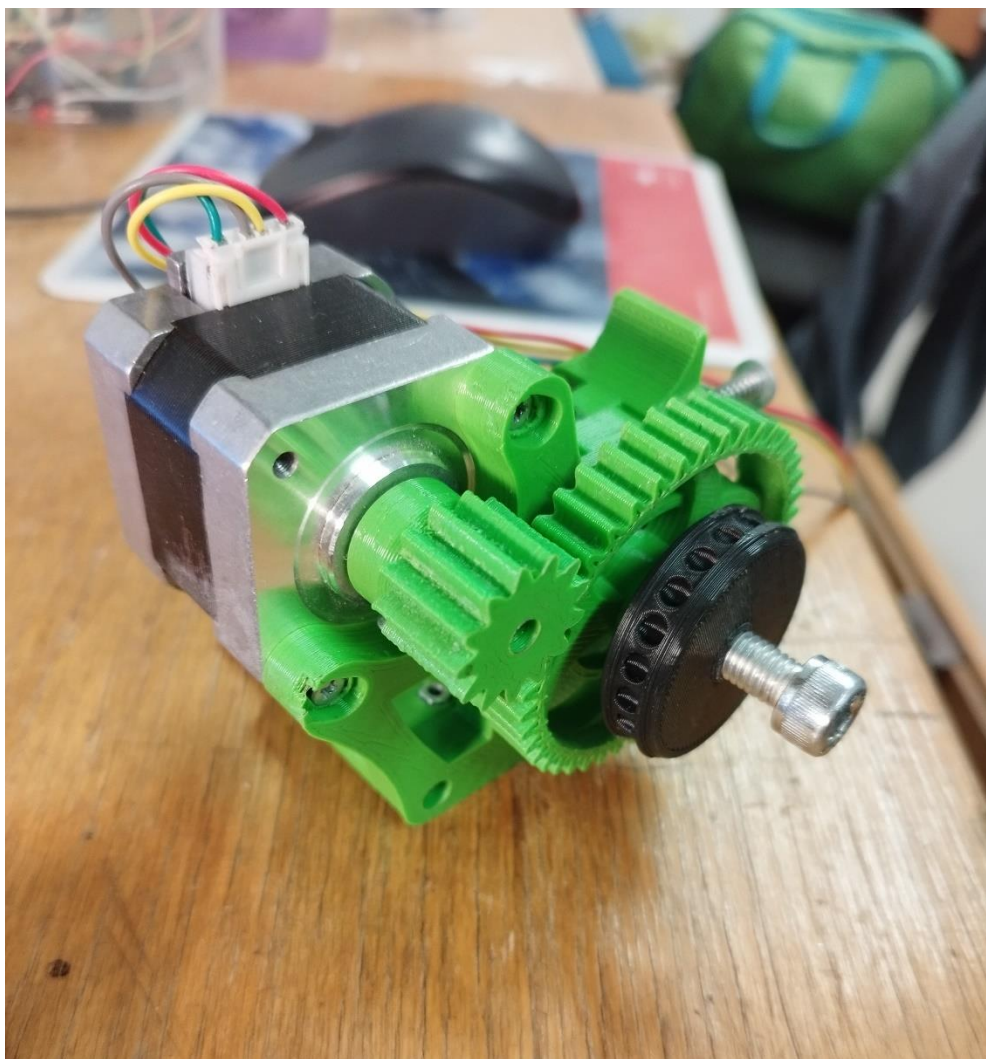
## 4.5 Az áttétel

A motor, amit használok a projektemhez egy régi szétszerelt 3D nyomtatóból lett kivéve. A nyomtató még a műhelyben volt és még ott volt az az áttétel, ami még akkor volt rajta amikor a nyomtatóban működött. Ezt az áttételt kiszedtem a nyomtatóból és rátettem a motorra. Az áttétel fogaskerekei tökéletesen működtek. Úgy döntöttem, hogy az előzőleg nyomtatott fogaskerék helyett egy nagyobbat használok, amibe egyszerre több gyöngy is belefér. Így újra kiszámoltam a fogaskerék sugarát, de most 20 gyöngyre. Ezután a szoftverbe beírtuk az új fogaskerék paramétereit és kinyomtattuk a fogaskereket. Ennek a fogaskeréknek a közepén nem egy D alak volt, hanem egy 8-as csavar menetei, ugyanis a nyomtatott fogaskerék egy olyan csavarra lett rácsavarva, amely az áttétel nagy fogaskerekét tartotta. A fentebb leírt szerkezetet szemlélteti a 4.3 ábra.



**4.3 ábra A nyomtatott fogaskerék, az áttétel és a motor**

A 4.4 ábra ugyanezt mutatja be a motor oldaláról.



**4.4 ábra A nyomtatott fogaskerék, az áttétel és a motor a motor oldaláról**

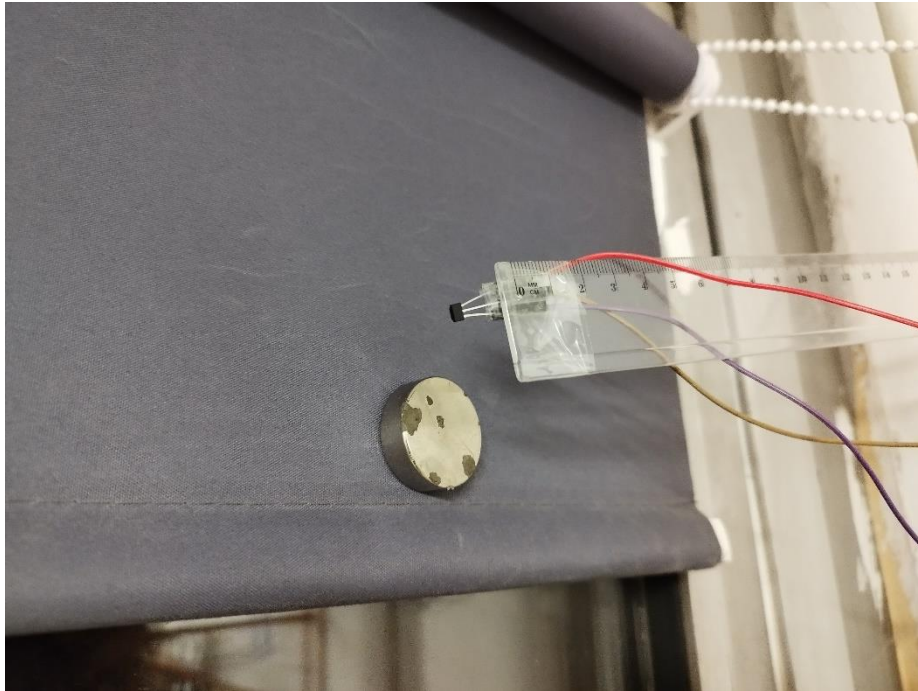
Amikor már a rendelkezésemre állt a képen látható szerkezet, ismét rátettem a fogaskerekre a gyöngyláncot, kipróbáltam és tökéletesen működött a rendszerem, azonban a tesztekéről a következő fejezetben beszélek bővebben.

## **4.6 A megvalósítás képekkel illusztrálva**

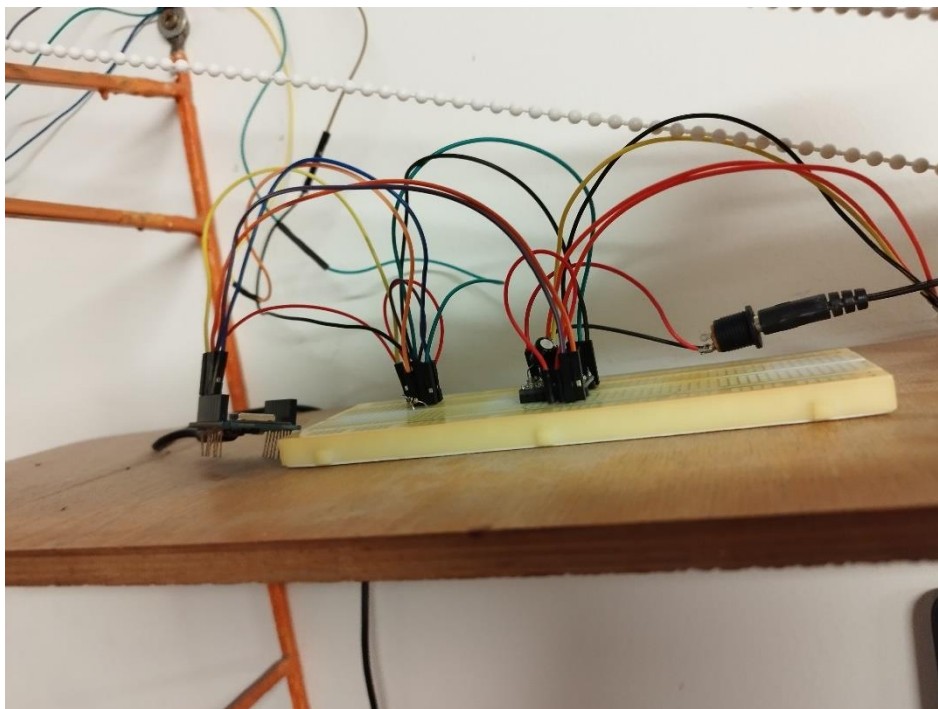
A megvalósítást a kollégiumi szobámban található, ablakra szerelhető sötétítővel végeztem. A Hall szenzort úgy tudtam a sötétítő tetejénél rögzíteni, hogy egy vonalzó egyik végére rögzítettem a szenzort, amíg a másik végét egy fémcsőhöz rögzítettem. A 4.5 ábra a szenzort szemlélteti működés közben, amikor a mágnes a közelébe ér. A többi alkatrészt, beleértve ebbe a motort is, az ablakra merőleges falon található polc tetejére tettem és ott rögzítettem. A 4.6 ábrán szintén működés közben láthatóak az összekötések a polc tetején. A breadboard-tól balra látszik a mikrokontroller, a breadboard bal szélén



van bekötve az ellenállás és a Hall szenzor, amelynek vezetékei a fal mentén húzódnak. Ezek a vezetékek a kép tetején látszanak.



**4.5 ábra A Hall szenzor működés közben**

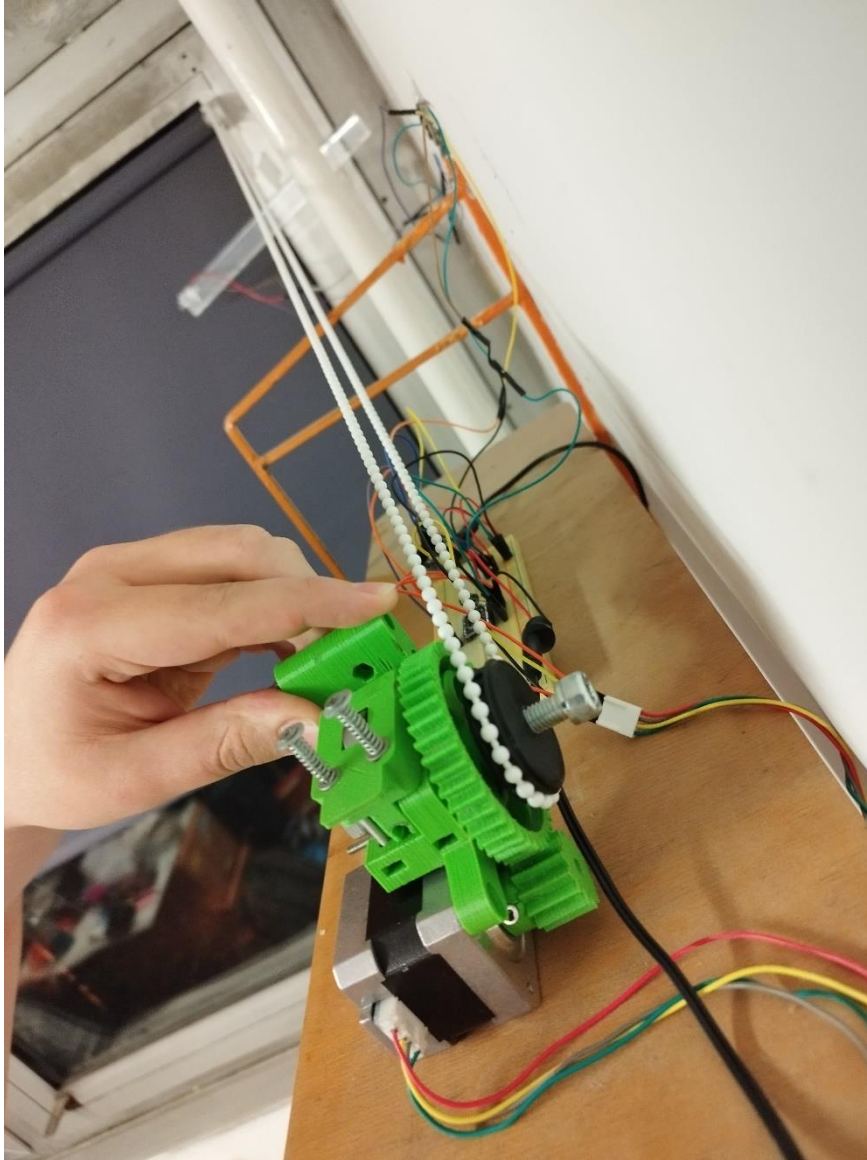


**4.6 ábra A mikrokontroller, a driver és az összekötések**

A 4.6 ábrán a breadboard közepén látható a driver és a kép jobb oldalán láthatjuk a power jack DC csatlakozót, amely 12V feszültséggel látja el a motort. A kép tetején



láthatjuk a feszülő gyöngyláncot, amely a 4.7 ábrán látható és a breadboard-tól jobbra elhelyezkedő motor által hajtott fekete fogaskereket veszi körül.



**4.7 ábra A teljes rendszer a motorral az előtérben működés közben**

A 4.7 ábrával zárom ezt a fejezetet, amelyen a teljes létrehozott rendszer látszik. Az előtérben található a motor, amely össze van csavarozva az áttétellel és a gyöngyláncot húzza, a háttérben pedig a sötétítő, amely kellemes tempóban mozog felfelé, miközben ennek a projektnek sikerességét szimbolizálja.

## 5 Tesztelés

### 5.1 A rendszer használhatóságának bizonyítása

Bár a rendszerem elkészült, azonban ennek használhatóságát bizonyítani képeken keresztül lehetetlen. Így az egyetlen dolog, amit tehetek az, hogy amellet, hogy képeket mutatok a működésről, tényekkel is bizonyítom, hogy a rendszerem kiválóan üzemel. A motornak a nyomatéka bekapcsolt állapotban 400 mNm az adatlapja szerint. A tényleges nyomatéka az áttétel miatt megnő közel 3,308-szorosára, vagyis 1323,2 mNm-re. A kicsi fogaskeréknek a sugarát 1 cm-nek állapítottam meg mérés után. Az erőt úgy számolhatjuk, hogy a nyomatékokat elosztjuk a sugárral.  $1,3232 \text{ Nm}$  elosztva  $0,01$  méterrel  $132,32 \text{ N}$  lesz. Tehát  $132,32$  Newton erőt képes kifejteni a motor sötétítőn. Ez annak felel meg, mintha egy  $13,49 \text{ kg}$ -os testet rögzítenék a gyöngylánc azon részéhez, amely lefelé mozgása esetén a sötétítő felfelé mozogna, majd rögzítés után magára hagynám a rendszert. Sajnos ezt a valóságban nem tudtam kipróbálni, de belátható az, hogy a magára hagyott rendszerben felhúzná a test a sötétítőt, tehát a motor az áttétellel képes lesz a sötétítő vezérlő rendszer működtetésére. Azonban a kiszámított erőérték megegyezik a megépített rendszerem terhelési határával, tehát ha úgy megnövelnénk a sötétítő tömegét, hogy annak felhúzásához több, mint  $132,32 \text{ Newton}$  erő kifejtése lenne szükséges, akkor a motor elkezdene kihagyni lépéseket és nem tudná felhúzni a sötétítőt.

Mivel a kollégiumban szereltem össze a rendszeremet, ezért a kollégium tulajdonát kellett használnom a megvalósításhoz, beleértve ebbe a sötétítőt és a polcot is. A motort csavarokkal lehetett volna megfelelően rögzíteni, de ezzel rongáltam volna a polcot, így inkább fogtam a motort tesztelés során, hogy ne mozduljon el. Azonban mivel a kép készítésekor elengedtem a motort, így a rajta lévő gyöngylánc, ami tesztelés közben feszes volt, kicsit lentebb ereszkedett. Az 5.1 ábrán látszik a teljes rendszer működés közben.



**5.1 ábra A sötétítő mozgató rendszerem működés közben**

## **5.2 A helyes működés tesztelése**

A tesztelés során a rendszerem minden funkcióját kipróbáltam és feljegyeztem a tapasztaltakat. A továbbiakban ezeket fogom ismertetni.

A teszt előtt egy a motor lépését számoló kóddal megszámloltam, hogy hány lépés a motor számára, amíg a legalsó pozícióból felhúzza a sötétítőt a legfelső pozícióba. Erre azért volt szükség, mert a kódban az egyik változó ezt a számot tárolja és ennek segítségével a motor számon tudja tartani, hogy éppen hol jár. A rendszeremben a teljes felhúzáshoz szükséges lépésszám 4800 volt.

A teszt előtti kiindulási helyzetben a sötétítő nagyjából félig volt felhúzva. Miután ráadtam a tápfeszültséget, elindult az indítások elején kötelező folyamat, ugyanis a motor elkezdte felhúzni a sötétítőt, addig amíg a Hall szenzor érzékelte a sötétítő aljára rögzített

mágnesnek a mágneses terét. Amikor ez megtörtént, a motor megállt és lassan leeresztette a sötétítőt addig, amíg a Hall szenzor már nem érzékelte a mágnest. Ezután a motor ismét megállt és innentől vált irányíthatóvá. Ez az állás a sötétítő legfelső pontja. Ennél feljebb nem megy helyes működés esetén. Amíg ebbe pozícióba beáll a sötétítő a mágnes segítségével, a weboldal vezérlő gombjai le vannak tiltva, így bármely gomb megnyomása ekkor még nem változtat a motor állásán. Ez ennek a funkciónak a helyes működését igazolja, ugyanis, ha lehetne vezérelni ekkor is a motort, akkor az a legfelső pont beállításában megzavarná a rendszert.

A következő teszt a sötétítő teljes lehúzása. Ehhez meg kell nyomni a weboldalon a „DOWN” gombot. Megnyomásra azonnal reagálva a sötétítő egyenletes mozgással lehúzódik és 4800 lépés után, pontosan az ablak alján megáll. Ekkor az „UP” gombot megnyomva a sötétítő egyenletes mozgással felhúzódik és megáll abban a pozícióban, amelyet a legfelső pozíciónak kineveztünk a bekapcsolás után. Ezután megnyomjuk újra a „DOWN” gombot, hogy a sötétítő ismét elinduljon lefelé és közben megnyomjuk a „STOP” gombot, melyre a motor és a sötétítő azonnal megáll. Tehát a három felső sorban lévő vezérlő gomb helyesen működik. Ekkor a köztes állapotok kipróbálásán volt a sor. Először a legfelső állásból mentem a 75%-os gyorsgombbal a legfelső köztes állásba. Ekkor a sötétítő úgy állt meg, hogy az ablak felső negyedét takarta. Ezután megnyomtam az 50% gombot, melyre megint megindult a motor és akkor állt meg, amikor a sötétítő az ablak felső felét takarta. Végül megnyomtam a 25% gombot és ennek eredményeként a sötétítő az ablak felső háromnegyedét takarta. Ezután a „DOWN” gombbal lementem a legalsó állásba és fordított sorrendben végigmentem lentről is a gyorsgombokon és ugyanazokat az eredményeket kaptam, amik ezen gombok helyes működésére utaltak. A weboldal hibátlanul működött telefonról és laptopról is Google Chrome-on, Mozilla Firefox-on és Operán egyaránt.

### **5.3 Hogyan tovább?**

Gondolkodtam azon, hogy a szakdolgozaton túl hogyan lehetne továbbfejleszteni a rendszert. Arra jutottam, hogy az ismert hibák kijavítása után, vagyis miután megfelelően rögzítettem a motort és a szoftver léptetési módszerét megváltoztattam, lehetne a rendszert teljesen automatizálni fényérzékelő szenzorok segítségével. A szenzorok által adott jel attól függne, hogy mennyire van világos odakint és a felhasználó

által megadott benti fényviszonyokat úgy tartaná állandó szinten, hogy a sötétítő mozgásával szabályozná a benti világosság szintet.

Meg lehetne oldani azt is, hogy a vezetékek ne rontsák a benti tér esztétikáját. Ezt az alkatrészek megfelelő tokozásával lehetne elérni. Továbbá csökkenteni lehetne a vezetékek és alkatrészek számát azzal, hogy napelemeket szerelek a sötétítő külső oldalára és ezekkel tölténém a tokba szerelt akkumulátorokat, amelyek a rendszer tápfeszültségét biztosítanák a továbbiakban. Létrehozhatnék a tokozásban fizikai gombokat, amelyekkel szintén irányítható lenne a sötétítő. Fejleszthetnék egy applikációt, amely leváltaná a weboldalas megoldást. Így nem lenne muszáj a mikrokontrollernek és a vezérlő eszköznek ugyanazon a WiFi hálózaton lennie, hanem a világ bármely pontjáról lehetne irányítani a sötétítőt interneteléréssel és egy jelszó birtokában, hogy ne lehessen bárkinek a sötétítőjét engedély nélkül irányítani.

Ezek a fejlesztések mind jobbra tennék a jelenlegi rendszert, azonban ezek az újítások már nem kerülnek megvalósításra ennek a szakdolgozatnak a keretein belül.

## 6 Összefoglalás

A feladatom egy olyan sötétítő mozgató rendszer megtervezése és megvalósítása volt, amit WiFi-n keresztül lehet vezérelni.

A projektem elején számos interneten található dokumentumból, cikkből, tanulmányból és oktatóvideóból tettem szert olyan tudásra, amely a projektem megtervezéséhez és megvalósításához elengedhetetlenül járult hozzá.

A tervemből minden megvalósításra került, beleértve ebbe a funkciókat, a weblapot és magát a rendszer összeépítését is. A rendszeremet látni működés közben szenzációs érzéssel töltött el és elképesztően büszke voltam magamra, ugyanis úgy működött, ahogy azt kitaláltam és ahogy a feladatkiírásban le volt írva.

A végeredmény egy olyan sötétítő mozgató rendszer lett, amely egy weboldalon található vezérlő panelről irányítható. Habár nem tökéletes, azonban a feladatkiírásban szereplő funkciókat megfelelően megvalósítja.

A tesztelés során azokat a tanulságokat vontam le a látottakból, hogy a motor segítségével jóval kényelmesebb a sötétítő üzemeltetése és egy okos otthonban remekül helytállna az én rendszerem is a motor megfelelő rögzítése esetén. Azt figyeltem meg, hogy az én rendszerem kicsivel hangosabb, mint amit eddig láttam, de ezt betudtam annak, hogy az áttétel fogaskerekei összekoccannak.

Ennek a projektnek a megvalósítása azért volt hasznos, mert egy számomra ismeretlen témakörben rengeteg elméleti és gyakorlati tudásra tettem szert és nagy a valószínűsége annak, hogy a jövőben a tanulmányaim során, vagy a munkahelyemen valamilyen formában ezt a tudást hasznosítani fogom. A világ pedig ezáltal gazdagabb lett egy gondosan megtervezett és kivitelezett projekttel, amelyet egyszer majd talán arra használ valaki, hogy a saját projektjéhez szerezzen pár ötletet.

## 7 Irodalomjegyzék

- [1] **Wright, Michael:** *How to Control Stepper Motors: The Most Comprehensive, Easy-to-Understand Advanced Guide for Hobbyists and Experts* (2016)
- [2] A 28BYJ-48 átalakításának leírása:  
<https://ardufocus.com/howto/28byj-48-bipolar-hw-mod/>
- [3] Különböző típusú léptetőmotorok működésének leírása:  
<https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/>
- [4] A léptetőmotorok működése és különböző típusai:  
<https://uk.rs-online.com/web/content/discovery/ideas-and-advice/stepper-motors-guide>
- [5] A NEMA17 17HS3001-20B adatlapja:  
<https://jkongmotor.en.made-in-china.com/product/jKBJzXuVAWhG/China-NEMA17-Square-40mm-Length-Stepper-Motor-with-Circuit-Board.html>
- [6] Az ULN2003A adatlapja:  
<https://datasheetspdf.com/pdf-file/366764/STMicroelectronics/ULN2003A/1>
- [7] Az A4988 driver adatlapja:  
[https://www.pololu.com/file/0J450/a4988\\_DMOS\\_microstepping\\_driver\\_with\\_translator.pdf](https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf)
- [8] A mikrokontroller adatlapja:  
[https://docs.ai-thinker.com/\\_media/esp8266/docs/esp-12f\\_product\\_specification\\_en.pdf](https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf)
- [9] Encoder működése:  
<https://www.youtube.com/watch?app=desktop&v=qT6FdvEsMs>
- [10] Twentei egyetem oktatóvideója ellenáram érzékelésével pozíciómeghatározásról:  
<https://www.youtube.com/watch?v=DGqvREtBnS4>
- [11] A végállásokról:  
<https://reprap.org/wiki/Endstop>

- [12] A mechanikai végállásról:  
[https://reprap.org/wiki/Mechanical\\_Endstop](https://reprap.org/wiki/Mechanical_Endstop)
- [13] A mágneses végállásokról bővebben:  
<https://www.tme.eu/hu/news/library-articles/page/42525/Hall-erzekelo-hogyan-ellenorizzuk-a-mukodeset/>
- [14] Már létező legjobbnak titulált okos otthon sötétítő megoldások:  
<https://www.nytimes.com/wirecutter/reviews/best-smart-blinds/>
- [15] Már létező 2023 legjobb okos otthon sötétítő megoldásai:  
<https://reviewed.usatoday.com/smarthome/best-right-now/best-smart-blinds#lutron-serena-smart-roller-shades-review>
- [16] Már létező megoldások, amik a legjobbak az otthonok hűvösen és sötétben tartásában:  
<https://www.popularmechanics.com/home/interior-projects/g44649431/best-smart-blinds/>
- [17] Alkatrészek összeszereléséről szóló cikk:  
<https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>
- [18] A cikkben talált kódokkal teszteltem az összeköttetések helyességét:  
<https://srituhobby.com/how-to-use-the-a4988-stepper-motor-driver-with-28byj-48-stepper-motor/>
- [19] A weboldal megírásában negy segítséget nyújtó cikk:  
<https://randomnerdtutorials.com/esp32-esp8266-input-data-html-form/>
- [20] A webszerver és a kliens közötti kommunikáció megteremtésében segítséget nyújtó cikk:  
<https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/>
- [21] Az AccelStepper könyvtárban lévő függvények és magyarázatuk, amelyet a kód megírásához használtam:  
<https://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html>
- [22] A weboldal szerkesztésében segítséget nyújtó weboldal:  
<https://www.w3schools.com/>



[23] A végállás bekötéséhez információt nyújtó cikk:

<https://maker.pro/arduino/tutorial/how-to-use-a-hall-effect-sensor-with-arduino>

Valamennyi link elérhető volt 2023. 12. 6-án.

## 8 Képek forrásai

- [1] [https://fr.wikipedia.org/wiki/Fichier:Moteur\\_pas\\_%C3%A0\\_pas\\_MRV.png](https://fr.wikipedia.org/wiki/Fichier:Moteur_pas_%C3%A0_pas_MRV.png)
- [2] <https://components101.com/motors/28byj-48-stepper-motor>
- [3] <https://market.samm.com/17hs3001-20b-3d-printer-stepper-motor-nema-17-model-en>
- [4] <https://uk.rs-online.com/web/p/darlington-pairs/1686450>
- [5] [https://www.researchgate.net/figure/RELAY-DRIVER-ULN2003A-The-ULN2003-is-a-high-voltage-high-current-Darlington-array\\_fig3\\_372788826](https://www.researchgate.net/figure/RELAY-DRIVER-ULN2003A-The-ULN2003-is-a-high-voltage-high-current-Darlington-array_fig3_372788826)
- [6] <https://robu.in/product/uln2003-driver-module-stepper-motor-driver-board/>
- [7] <https://circuitdigest.com/microcontroller-projects/interface-a4988-stepper-motor-driver-with-arduino>
- [8] <https://www.elektor.com/esp-12f-esp8266-based-wi-fi-module>
- [9] [https://en.wikipedia.org/wiki/Rotary\\_encoder](https://en.wikipedia.org/wiki/Rotary_encoder)
- [10] <https://www.tinytronics.nl/shop/en/sensors/current-voltage/acs712-current-sensor-module-20a>
- [11] <https://hu.farnell.com/omron/ss-5gl/microswitch-spdt-lever-125vac/dp/103444>
- [12] <https://www.sparkfun.com/products/8642>
- [13] <https://reprap.org/wiki/Endstop>
- [14] <https://www.addicore.com/products/ss49e-linear-hall-effect-sensor>
- [15] <https://www.ikea.com/us/en/p/fyrtur-black-out-roller-blind-smart-wireless-battery-operated-gray-90417462/>
- [16] <https://www.ebay.co.uk/itm/404552732738>
- [17] <https://thepihut.com/products/stepper-motor-nema-17-size-200-steps-rev-12v-350ma>
- [18] <https://www.ebay.co.uk/itm/191911031467>
- [19] <https://www.flyrobo.in/10k-ohm-resistor-other>

- [20] <https://www.microcontroller.hu/termekcimke/hall-sensor/>
- [21] <https://www.emag.hu/wifi-modul-esp8266-wemos-d1-mini-esp-12f-tobbsszinu-esp8266-wemos-d1-mini/pd/DW5798MBM/>

Valamennyi link elérhető volt 2023. 12. 6-án.