**Debreceni Szakképzési Centrum**

**Brassai Sámuel Gimnáziuma és Műszaki Szakgimnáziuma**

**ZÁRÓdolgozat**

Ambilight LED világítás

**Horváth Károly**

**54 481 05 Műszaki informatikus**

**Debrecen**

**2020**

**ZÁRÓDOLGOZAT FELADATKIÍRÁS**

|  |  |  |  |
| --- | --- | --- | --- |
| Név: | Horváth Károly | OKJ szám: | 54 481 05 |
| Tagozat: | nappali | Szakképesítés: | műszaki informatikus |

Témavezető neve: Kányádi Zoltán

|  |
| --- |
| A dolgozat témája:  Speciális ledekből álló mikrovezérelt szalag elkészítése, amely egy megjelenítő  (pl. számítógép-monitor) aktuális színeit feltérképezi és harmonizálva kivetíti az eszköz hátterére   * a dolgozat témájának kiindulási pontja, háttere, célja * az eszköz megépítéséhez szükséges kellékek, alkatrészek és fontos jellemzőik * a kapcsolási rajz értelmezése; az eszköz vezérlésének a leírása * az eszköz megépítésének lépései, az egységek közötti együttműködés formái * a megvalósításhoz szükséges szoftver kódja és lényeges részeinek kibontása, elemzése * további fejlesztési irányok, kiegészítő lehetőségek |
|  |

|  |  |
| --- | --- |
| A témavezető tanár: | |
|  | ....................................................  Kányádi Zoltán |

Kérem a szakdolgozat témájának jóváhagyását.

Debrecen, 2020. január 31.

|  |  |
| --- | --- |
|  | ....................................................  Horváth Károly |

A záró dolgozat témáját az iskolavezetés jóváhagyta.

Debrecen, 2020. január 31.

|  |  |
| --- | --- |
|  | ....................................................  Szilágyi Sándor  szakképzési intézményvezető helyettes |

Kötelező konzultációk igazolása:

|  |  |  |  |
| --- | --- | --- | --- |
| Alkalom: | Készültség | Dátum: | Témavezető aláírása |
| 1. | 30% | 2020.02.28 |  |
| 2. | 65% | 2020.03.20 |  |
| 3. | 95% | 2020.04.15 |  |

Tartalomjegyzék

[Bevezetés 2](#_Toc38018157)

[1.ALKATRÉSZEK, KELLÉKEK BEMUTATÁSA, FONTOS JELLEMZŐIK 3](#_Toc38018158)

[1.1 Arduino 3](#_Toc38018159)

[Története 3](#_Toc38018160)

[Hardver 3](#_Toc38018161)

[Szoftver 12](#_Toc38018162)

[1.2 Led szalag 14](#_Toc38018163)

[Általánosságban 14](#_Toc38018164)

[WS2812B LED szalag 16](#_Toc38018165)

[1.3 Billenőkapcsoló 17](#_Toc38018166)

[1.4 Próbapanel 18](#_Toc38018167)

[1.5 470Ω-os ellenállás 18](#_Toc38018168)

[1.6 Áramellátás 19](#_Toc38018169)

[2.A KAPCSOLÁSI RAJZ ÉRTELMEZÉSE, AZ ESZKÖZ VEZÉRLÉSÉNEK LEÍRÁSA 20](#_Toc38018170)

[2.1 A kapcsolási rajz értelmezése 20](#_Toc38018171)

[2.2 Az eszköz vezérlésének leírása 20](#_Toc38018172)

[3.AZ ESZKÖZ MEGÉPÍTÉSE 21](#_Toc38018173)

[3.1 Az eszköz megépítésének lépései 21](#_Toc38018174)

[3.2 Problémák az építésnél 23](#_Toc38018175)

[4.A SZOFTVERES RÉSZE A PROJEKTNEK 24](#_Toc38018176)

[4.1 Prismatik nevű program 24](#_Toc38018177)

[4.2 Az eszköz programkódja, lényeges része 25](#_Toc38018178)

[ÖSSZEFOGLALÁS 26](#_Toc38018179)

[FELHASZNÁLT IRODALOM 27](#_Toc38018180)

[MELLÉKLETEK 28](#_Toc38018181)

# Bevezetés

Záródolgozatom témája egy Arduino-val vezérelt LED szalag, mely a monitor hátuljára körbe fel van ragasztva, melynek célja, hogy lemásolja a monitoron megjelenő színeket és ezt kivetítse. Ezt Ambilight világításnak hívják, ez a neve a Philips által kidolgozott eredeti technológiának, melyet a cég 2002 óta rak bele a saját készülékeibe. Ez abszolút egy “látványelem”, a kivetített színek optikailag megnövelik a kijelzőt, és egy nagyon szép végeredményt kaphatunk. Azért választottam ezt a témát, mert mindig is vonzottak az ilyen dizájnelemek, nagyon fel tudják dobni egy-egy dolog hangulatát. Segítségemre szolgált az is, hogy nagyon sok információt sikerült összegyűjtenem az internetről, amik miatt sokkal egyszerűbb volt a dolgom.

Nos, mi is van akkor, ha a mi készülékünk nem rendelkezik ezzel a technológiával?!

Nagyon sok féle házi megvalósítás létezik, de nyilván egyik sem lesz olyan pontos és tökéletes, mint az eredeti. Alapból két féle lehetőség van:

-az első a HDMI jelét figyeli, és az alapján vezérli a ledeket

-a második pedig Arduino által vezérelt led szalag

Előbbi előnye, hogy szinte mindenhol működik, (pl. videóknál, játékoknál, tv-adásoknál), de viszont hátránya az, hogy nem lehet teljesen személyre szabni, és meglehetősen drága kivitelezni.

Utóbbi viszont egy elég jó megoldásnak bizonyul, magam is ezt választottam. Itt két legfontosabb elemünk van, egy Arduino mikrovezérlő és egy WS2812B LED szalag, ami azért különleges, mert mindegy egyes LED-et külön-külön lehet vezérelni, erről majd részletesebben a későbbiekben fogok írni. Egyik legnagyobb előnye, hogy viszonylag olcsó kivitelezni, a led szalagon kívül csak egy mikrovezérlőre van szükség. Hátránya viszont, hogy az Arduino miatt mindenképp szükség van egy PC/HTPC-re, ami a jelet küldi a vezérlőre, szóval itt már a különböző TV-adások nem fognak működni, de játékoknál és filmeknél tökéletesen fog dolgozni.

Azért is találtam ezt a projektet megfelelőnek a szakdolgozatomhoz, mert egyaránt elektronikai és programozói tudásomat is fel tudtam benne használni. Az alkatrészek összeállítása elég nagy előkészületekkel járt, sorban végig kellett gondolnom, hogy mit és hogyan szeretnék megvalósítani. Át kellett gondolnom, hogy mekkora LED szalagra lesz szükségem, milyen tápellátást fogok neki biztosítani, milyen mikrovezérlő lesz a legmegfelelőbb számomra.

Célom az, hogy részletesen bemutassam munkámat, minden egyes nehézséggel és megoldással, valamint kedvet csinálni másoknak, akik szintén kedvelik ezeket a látványos technológiákat.

# 1.ALKATRÉSZEK, KELLÉKEK BEMUTATÁSA, FONTOS JELLEMZŐIK

Az alábbiakban részletezni fogom a felhasznált alkatrészeket, valamint ismertetni szeretném a projekt működését, összeállítását.

## 1.1 Arduino

Az egyik leglényegesebb alkatrész, mely a legtöbb feladatot látja el, a mikrovezérlő. A mikrovezérlők, vagy más néven mikrokontrollerek egyetlen lapkára integrált, általában vezérlési feladatokra optimalizált cél-számítógép. A mikrokontroller egy mikroprocesszor kiegészítve az áramköri lapkájára integrált perifériákkal. Megtalálható benne:

* CPU - azaz központi egység - a processzor
* RAM - azaz adatmemória - a ram
* ROM - azaz programmemória - ez olyan, mint egy pendrive, csak kisebb
* be-kimeneti vezérlők – ezekkel tudja a jeleket érzékelni, vagy a jeleket kiküldeni

Manapság az élet minden területén találkozunk ilyen vezérlőkkel, akár digitális órákban, járművekben és még sokáig sorolhatnánk.

### Története

Az egyik legelterjedtebb ilyen mikrokontrollert használó vezérlő, az Arduino. 2002-ben egy olasz egyetemen a diákok kezdtek el dolgozni a fejlesztésén, mert amit akkor használtak gyakorlásra sok tanulónak nem volt megfizethető, ezért egy olyan eszközt akartak létrehozni, amelyet könnyen beszerezhet bárki, és mellé még ne legyen túl bonyolult se. Céljuk az volt, hogy egy nagyon sokszínű platformot hozzanak létre, rengeteg vezérelhető szenzorral és további alkatrészekkel. Mivel a fejlesztők azt akarták, hogy az Arduino nagy áttörés legyen a vezérlők versenyében, úgy döntöttek, hogy nyílt-forráskódúvá teszik. Ezzel bebiztosították az érdeklődést felé, és nagyon sokan kezdtek el rajta dolgozni, így rohamosan elterjedt és 2013-ra már kb. 700.000 db vezérlő volt a felhasználók kezében.

### Hardver

Az elkészített Arduino-kat egy nyomtatott áramkörként kell elképzelni, melyeket Arduino Board-nak neveznek. Többféle board van, ezek főleg a mikrokontroller típusában, méretben, belső memóriában és a ki- és bemenetek számában különböznek. A legtöbb Arduino-ban egy 8-bites, az Atmel cég által gyártott AVR mikrokontroller található. Az AVR egy mikrovezérlő család, melyet a cég 1996-ban fejlesztett ki. Ezek egy darab chip-el rendelkeznek, és RISC típusúak. A RISC egy, a számítógépek processzorának tervezésénél alkalmazott tervezési stratégia. Ezen technológia fő jellemzői:

* nincs olyan utasítás, ami a memóriaelérést az aritmetikával kombinálja
* minden műveletvégző utasítás regisztereket használ
* kis számú, egyszerűsített címzési mód
* egyetlen RISC processzor sem használ indirekt címzést
* az utasítások „huzalozott” megvalósítása (nem mikrokóddal, hanem hardveresen kerülnek megvalósításra)
* minden utasítás ugyanolyan hosszúságú
* az utasításokat lehetőleg 1 órajelciklus alatt hajtsa végre
* utasítás-futószalagos végrehajtás
* nagyszámú általános célú regiszter

Az egyszerűbb kialakítás miatt több hely áll rendelkezésre az áramköri lapkán, amit a RISC processzorok tervezésekor gyakran regisztertárak, gyorsítótárak elhelyezésére használnak fel.

Azért különleges az AVR, mert ez volt az első mikrovezérlő-család, amelyben csipre integrált flash memóriát kezdtek használni a programtárolásra, ellentétben más mikrovezérlőkkel, amelyeknél ekkor még egyszer programozható ROM, EPROM vagy EEPROM memóriák szolgáltak erre a célra.

Az Arduino-hoz többek közt ezeket a mikrovezérlőket használják:

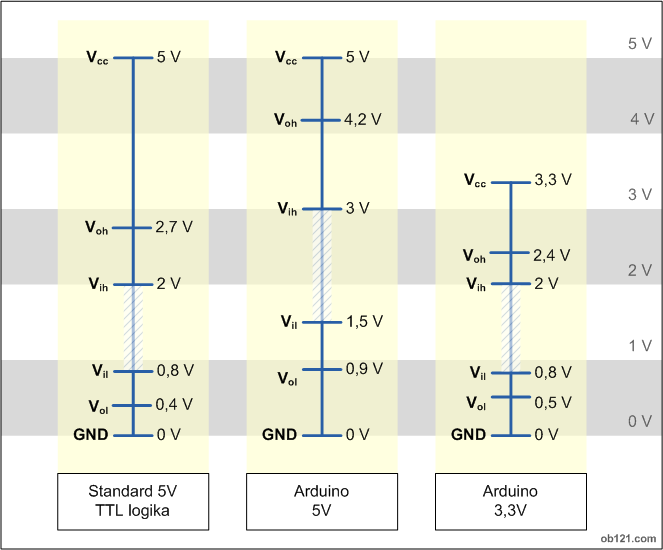
* ATmega8
* ATmega168
* Atmega328
* ATmega1280
* ATmega2560

Az Arduino-k általában több memóriatípussal rendelkeznek. A legfontosabb, a „fő” memória a flashmemória. Ez tárolja a board-ra feltöltött programkódot, és ezt kikapcsolás után is megőrzi, szóval elegendő csak egyszer feltölteni rá az aktuális kódot. Fontos megjegyezni, hogy nem az egész flashmemória terület erre használódik fel, hanem a beépített letöltőprogram (bootloader) és a különböző kommunikációk is ebből vesznek el területet, valamint a letöltött könyvtárak is sok helyet tudnak foglalni. Ennek eredményeként, ha például a mi eszközünkben 32kb a flashmemória, akkor a programkód tárhelyének kb. 24-30kb lesz elérhető. Fontos dolog még az, hogy ez nem egy végtelenül-újraírható memória, a maximális írásciklus-számát 100.000-ben limitálták. Ez arra elég, hogy egy naponta 10-szer újraírt programot mintegy 27 évig gond nélkül tároljon.

Következő memóriatípus az SRAM (static random-access memory; statikus tetszőleges hozzáférésű memória). Ez a programban definiált belső változókat tárolja. Legnagyobb különbség a flashmemóriával szemben, hogy ez áramtalanítás után nem őrzi meg tartalmát, szóval minden egyes újraindításkor a program újradefiniálja a változókat és ezek kerülnek be az SRAM-ba.

A harmadik memóriatípus, amely megtalálható az Arduino-ban, az az EEPROM ((Electrically Erasable Programmable Read-Only Memory, programozható, törölhető, csak olvasható memória). Ez hasonlóan a flashmemóriához, kikapcsolás után is megőrzi tartalmát, és szintúgy 100.000 írásciklusra van hitelesítve. E memória kicsit lassabb, mint az SRAM. Ezekből a technikai jellemzőkből adódóan az EEPROM az alábbi funkciókra alkalmazható:

* konfiguráció(k) letárolása
* indulási alapbeállítások mentése
* újraindulásoktól független számlálók, értékek, gyüjtött értékek (pl. Üzemóraszámláló) mentése

A logikai magas (HIGH) és alacsony (LOW) szinteket alkalmazó (digitális) elektronikai berendezések többnyire az úgynevezett TTL logika (Transistor-Transistor Logic) alapján definiálják a magas/alacsony jelekhez köthető feszültségszinteket. Ezektől a szintektől az Arduino logikai szintjei némileg eltérnek:

1. ábra TTL logikai szintek [1]

* VOH: Minimális kimeneti tápfeszültségszint. Efölött a TTL eszköz HIGH jelet biztosít.
* VIH: Minimális bemeneti feszültségszint a magas (HIGH) jelhez.
* VIL: Maximális bemeneti feszültségszint az alacsony (LOW) jelhez.
* VOL: Maximális kimeneti feszültsége egy alacsony (LOW) jelhez.

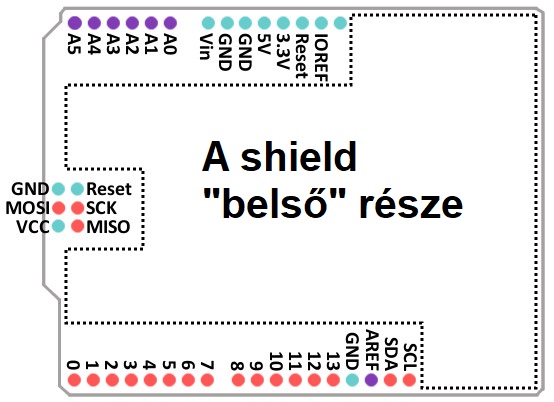
A letakart részeken a bemenet kiértékelése (HIGH vagy LOW) bizonytalanná válik.

Míg a program-logikai szinteket a true (igaz, 1) és false (hamis, 0) konstansokkal, addig a lábak állapotát a HIGH (magas, 1) illetve a LOW (alacsony, 0) konstansokkal lehet jellemezni.

* A HIGH állapot egy jel olvasása esetén az 5V-os bemeneteken 3V jelszint felett, míg a 3,3V bemeneteken 2,0V felett jelentkezik. A LOW állapotok 5-os board-oknál 1,5V alatt, 3,3V-os boardoknál kb. 1,0V alatt jelentkeznek.
* A köztes feszültségszinteket (5V-os boardoknál 1,5V - 3,0V, 3,3V-os board-oknál 1,0V - 2,0V) érdemes kerülni, mert a bemeneti állapot bizonytalanná válik.
* Amennyiben a bemeneti felhúzóellenállás előzőleg aktiválásra került az adott lábon, akkor onnan jó eséllyel soha nem fogunk HIGH állapotot olvasni.
* Kimenetek esetén a HIGH állapot kiadását követően a láb a maximális feszültségszintet veszi fel, azaz 3,3V-os board-oknál a 3,3V-ot, az 5V-osoknál az 5V-ot.
* Amennyiben előzően a belső felhúzóellenállást aktiváltuk, a kimeneti feszültség is HIGH jel esetén az 5V helyett kevesebb (kb. 3,3V) lesz.

Az Arduino mikrovezérlőket előzetesen beprogramozták egy indítóbetöltővel (boot loader), amely megkönnyíti a programok feltöltését a flash memóriába. A jelenlegi Arduino board-okat Universal Serial Bus (USB) segítségével programozzák.

Fontos eszközök ebben a témakörben az úgynevezett shield-ek. Az Arduino-k funkcionalitását könnyedén kiegészíthetjük ezekkel a Shield-ekkel, amelyek a Board-okhoz egyszerűen illeszthető elektronikai áramkörök. Ezek segítségével azt Arduino-t akár közvetlenül az internetre csatlakoztathatjuk, motorokat, LED-eket, vezérelhetünk vele, vagy Wi-Fi, Bluetooth hálózatra kapcsolódhatunk. Több shield is egymásra rakható, így nagyon sok érdekes projekt megvalósítható, mint példáúl egy olyan is, amelyet az iskolában egy tanóra keretében valósítottunk meg. Itt egy Ethernet Shield, egy Arduino és egy relé segítségével kapcsoltunk egy áramkört, webböngészőn keresztül. Ez egy nagyon érdekes projekt volt számomra, és ez a jövőben is hasznos lehet, ha valami ilyesmi munkába kezdek. Fontos tényező, hogy az összekapcsolni kívánt Arduino és Shield formailag teljesen megegyezzen. Az alábbi képen az általános kiosztás látható.



2. ábra Egy shield kiosztása [2]

* az 5V, 3,3V, GND pinek a kép felső részén látható, kék színnel jelzett helyre kerülnek, ez 8 darab pint jelent (régebbi típusoknál ez csak 6 darab volt)
* lila színnel jelölve, az analóg pinek állnak, 6 darab van belőlük
* piros színnel jelölve a másik oldalon pedig a digitális pinek láthatók, 14 db van

Pinek amikről még nem esett szó:

* Az úgynevezett SPI (Serial Peripheral Interface) pinek, amely egy szinkron soros kommunikációs interfész specifikáció, melyet rövid távú kommunikációhoz használnak, elsősorban a beágyazott rendszerekben. Az SPI eszközök full duplex módban kommunikálnak, a master-slave architektúrák egy mastert tartalmazhatnak. Az SPI egy master, multi-slave kommunikáció.

- MISO (Master In Slave Out): ezen keresztül kapja a jelet a master eszköz

-MOSI (Master Out Slave In): ezen keresztül küldi a jelet a master eszköz

- SCK (Serial Clock): egy órajel, melyet a master eszköz generál az adatok szinkronizálásához

* Az I²C (Inter-Integrated Circuit) pinek, amely egy multi-master, multi-slave, csomagkapcsolt soros busz. Az I²C jellemzően kis távolságú, viszonylag alacsony sebességű IC és fedélzeti rendszerek közötti kommunikációra szokás alkalmazni. Ez két darab pint jelent, az SCL és az SDA.

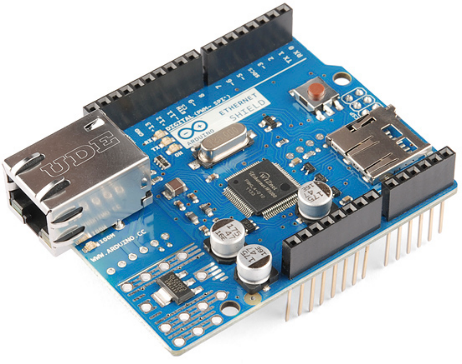
-SCL: ez az órajel, adatátvitel-szinkronizálást végez

-SDA: ez az adatjel, adatátvitelért felelős

Valamennyi shield az összes pint használja összekapcsolásnál, de olyan is előfordul, hogy csak párat (SPI, I²C, soros pinek, analóg pinek) valamelyikét.

Rengeteg shield van a piacon, túl sok ahhoz, hogy mindről szót ejtsünk. Alább a legtöbbet használt shieldeket fogom részletezni.

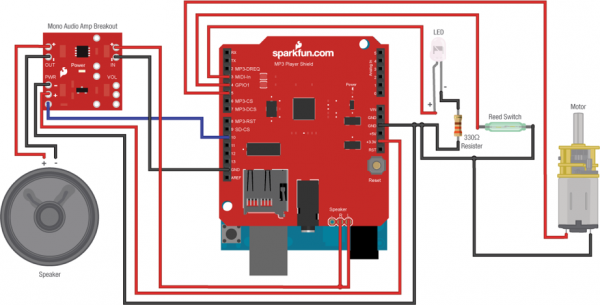
Ethernet, WiFi, Wireless, GPS



3. ábra Ethernet Shield [3]

* *Arduino Ethernet Shield:* Ez az egyik legelterjedtebb shield, segítségével az Arduino kapcsolódni tud az internethez, de csak kábel segítségével
* *Arduino WIFI Shield:* Ez tulajdonképpen az Ethernet Shield „vezetékmentes” változata. Az Arduino képes lesz csatlakozni WIFI routerhez, így az internethez
* *Cellular Shield w/ SM5100B:* Ezzel a shielddel az Arduino-t egy valóságos mobiltelefonként használhatjuk, üzenetet küldhetünk, vagy mikrofon segítségével beszélgethetünk is
* *GPS Shield:* ezzel egyszerűen az Arduino mindig „tudni” fogja, hogy hol van helyileg

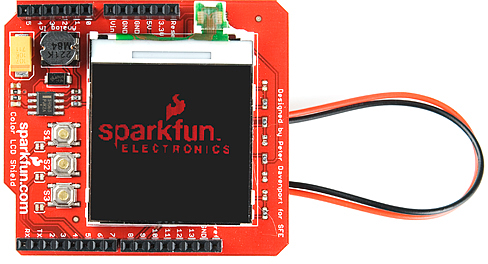
Zene és hang



4. ábra MP3 Shield [4]

* *MP3 Player Shield:* Egy microSD kártya és hangszóró segítségével saját MP3 lejátszót hozhatunk létre
* *Music Instrument Shield:* Hangszer effekteket tud kiadni (dob, zongora, fafúvósok, rézfúvósok, és még sok más hangot)

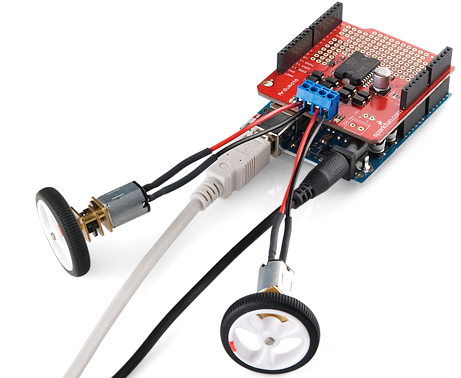
Kijelzők és kamerák



5. ábra Color LCD Shield [5]

* *Color LCD Shield:* Egy 128x128-as LCD kijelzővel lesz ellátva az Arduino
* *CMUcam Shield:* Ez egy kamerával ellátott shield, robotoknál szokták használni, akadályok kikerülése végett

Motor vezérlők

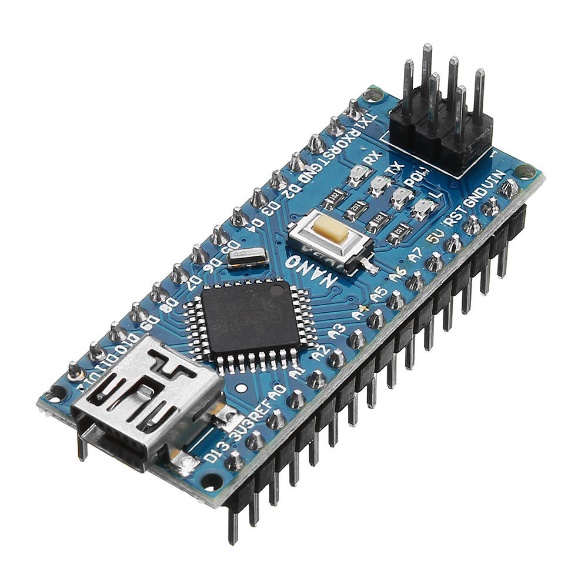


6. ábra Ardumoto Motorvezérlő Shield [6]

* *Ardumoto Motorvezérlő Shield:* Ez a klasszikus motorvezérlő shield, kettő darab DC motort tud kezelni
* *Monster Moto Shield:* Ez az előbbi erősebb változata, nagyobb teljesítményre képes, de funkiójuk ugyanaz

Mint említettem, több típusú Arduino létezik. Én a továbbiakban a három legsűrűbben előforduló típusról fogok beszélni, a Nano-ról részletesebben, mivel én is azt használom.

Arduino Nano



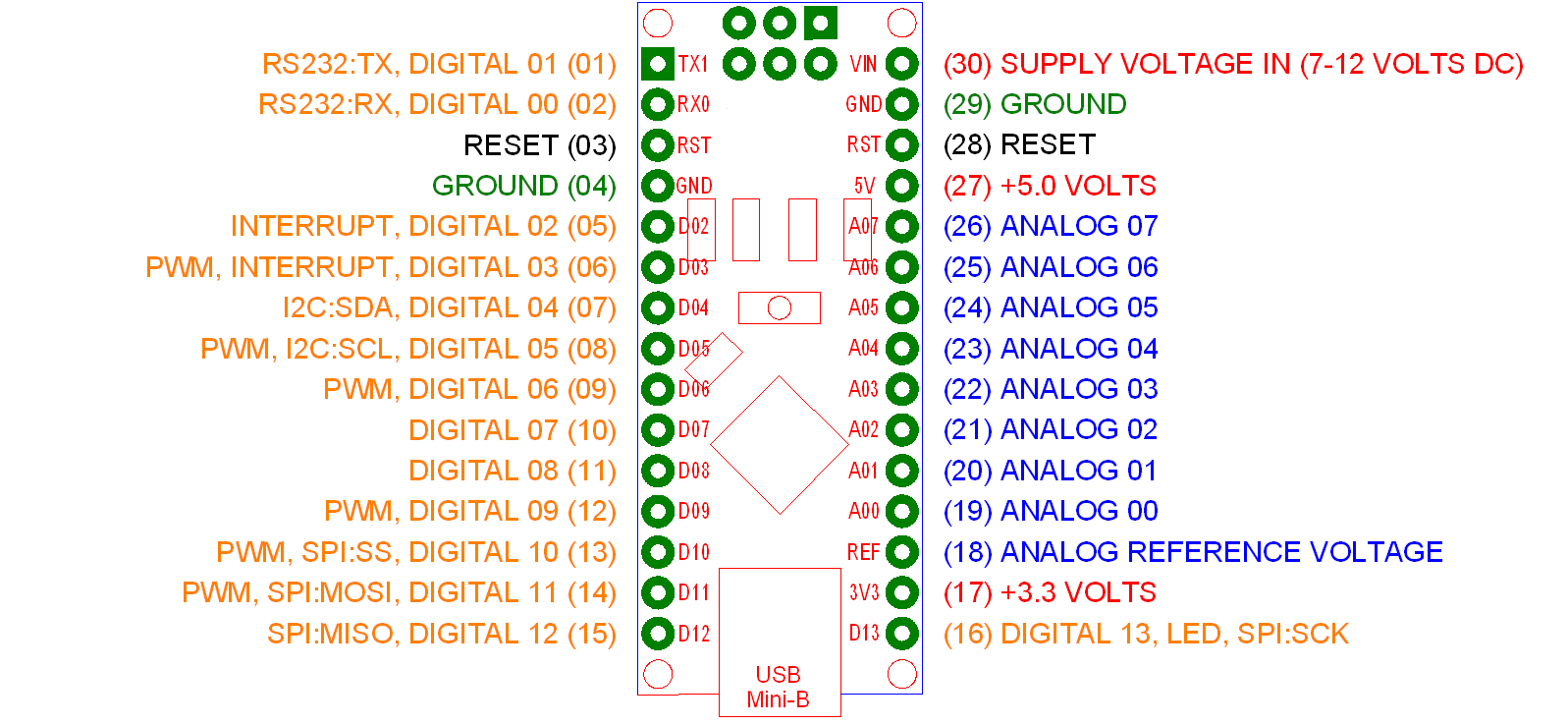
7. ábra Arduino NANO [7]

Ez a legkisebb, mindössze 18x45 mm nagyságú Arduino. Előnye, hogy hogy kevés helyet foglal, könnyedén lehet próbapenelhez csatlakoztatni, valamint nem túl drága, 1.000 Ft és 2.000 Ft között már hozzájuthatunk. Mindössze egy darab tápellátási lehetőség van rajta, egy Mini-B USB port. Én ezt választottam a projektem elkészítéséhez.

Specifikációk:

* Microkontroller: ATmega328
* Üzemi feszültség: 5V
* Bemeneti feszültség (ajánlott): 7V ~ 12V
* Bemeneti feszültség (határ): 6V ~ 20 V
* Digitális ki/bemenet pinek: 14
* Analóg bemenet pinek: 8
* DC feszültség egy darab ki/bemenet pinre: 40mA
* Flash Memória: 32KB (ebből 2kb-ot a már említett bootloader foglal le)
* SRAM: 2KB
* EEPROM: 1KB
* Órajel sebesség: 16MHz

Be és kimenetek a board-on:



8. ábra Arduino Nano pinkiosztás [8]

Mind a 14 digitális pin hasznáható ki és –bemenetekként is egyaránt, ezeket a programkódban lehet definiálni, de erre a későbbiekben ki fogok térni. Több pin speciális funkcióval rendelkezik, mint példáúl az RX0 (logikai 0) és a TX1 (logikai 1), melyek a TTL adatokat kezelik. A PWM pinek, mint ahogy a fenti ábrán is látszik (5.ábra) a 3, 5, 6, 9, 10 és 11-es pinek. Ezek arra szolgálnak, hogy a digitális jeleket analóg jelekké alakítsák impulzusszélesség-moduláció segítségével. Ez olyankor hasznos, amikor egy olyan elemet akarunk működtetni, amelyhez analóg jel szükséges, példáúl stepper vagy servo motorok, led fényerő szabályozása. Az SPI pinek, melyekről már szó esett, a 10(SS), 11(MOSI), 12(MISO) és 12(SCK)-es pineken érhetők el. A 13-as digitális pinhez egy beépített LED van csatlakozatva, mely fel vagy le van kapcsolva a csatlakozás függvényében. Az I²C pinek a 4(SDA), és 5(SCL) digitális pineken helyezkednek el. A 18-as pinen megtalálható az AREF, melyen az analóg pinek referencia feszültségét lehet szabályozni programkódon belül.

Mint említettem, a két másik Arduino ami elterjedtebb, az Arduino UNO és az Arduino MEGA2560. Összehasonlításképp készítettem egy táblázatot (1.táblázat), melyből kiderül a különbség a típusok között, és az hogy nekem miért volt a legelőnyösebb az Arduino NANO. Teljesítmény szinten nem látható különbség, mindhárom ugyanolyan feszültségszinten működik, kivéve a MEGA, egy ki- bemenetre jutó DC feszültsége, mely nem 40mA, hanem 20mA. Ez amiatt van, mert a MEGA-n sokkal több be- kimenet pin található, de ugyanolyan üzemi feszültségen működik. Memória szinten a két kisebb megegyezik, csak a MEGA memóriája nagyobb. A legnagyobb különbségek a méretekben és az árban vannak. Ezek voltak ezek a szempontok, amik miatt én a Nanot választottam. Mivel projektemben nekem csak 1db digitális pin-re van szükség, ezért a Nano és az UNO is egyaránt megfeleltek volna, de itt gondoltam át, hogy nekem minél kisebb kellene, hogy esztétikusan el tudjam helyezni a monitorom hátuljára. És nem utolsó sorban feleslegesnek tartottam több pénzt költeni egy UNO-ra, mivel ugyanarra a teljesítményre képesek.

|  |  |  |  |
| --- | --- | --- | --- |
| Modell | **Arduino NANO** | **Arduino UNO** | **Arduino MEGA 2560** |
| Mikrokontroller | ATmega328 | ATmega328P | ATmega2560 |
| Üzemi feszültség | 5V | 5V | 5V |
| Bemeneti feszültség (ajánlott) | 7V ~ 12V | 7V ~ 12V | 7V ~ 12V |
| Bemeneti feszültség (határ) | 6V ~ 20V | 6V ~ 20V | 6V ~ 20V |
| Digitális be -kimenetek | 14 | 14 | 54 |
| Analóg bemenetek | 8 | 6 | 16 |
| DC feszültség egy ki/bemenet pinre 40mA | 40 mA | 40 mA | 20mA |
| Flashmemória | 32 KB | 32 KB | 256 KB |
| SRAM | 2 KB | 2 KB | 8 KB |
| EEPROM | 1 KB | 1 KB | 4 KB |
| Órajel sebesség | 16 MHz | 16 MHz | 16 MHz |
| Méret | 45mm x 18mm | 69mm x 53mm | 101mm x 53mm |
| Ár | ~1.200 Ft | ~7.000 FT | ~15.000Ft |

**1**. táblázat Három Arduino összehasonlítása

### Szoftver

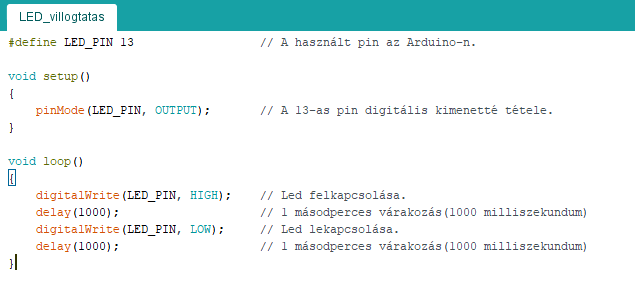
Az Arduino hardver programját bármilyen programozási nyelven lehet írni olyan fordítóval, amely bináris gépi kódot állít elő a célprocesszor számára, de a cég egy saját beépített fejlesztői környezetet is létrehozott. Ez az úgynevezett Arduino IDE(integrated development environment=integrált fejlesztői környezet), amely segítségével Arduino programokat készíthetünk, tesztelhetünk, majd az Arduino Board-okra tölthetjük.. Ez egy cross-platform program, ami azt jelenti, elérhető minden operációs rendszerre (Windows, Linux, macOS) egyaránt. A programot Java programozási nyelven írták. Ez magában foglal egy kódszerkesztőt, olyan funkciókkal, mint a szöveg vágása és beillesztése, a szöveg keresése és cseréje, az automatikus behúzás, a zárójelek illesztése és a szintaxis kiemelése. Tartalmaz továbbá egy szöveges konzolt, egy eszköztárt a legtöbbet használt funkciókhoz és a működési menük hierarchiáját. Az Arduino IDE a C és a C++ nyelveket támogatja.

Fontos elemek ebben a témában a könyvtárak. Ezekkel kiterjeszthetjük a program funkcionalitását, szükségesek lehetnek, ha például LCD kijelzőt, szervo motort, vagy esetleg ledeket akarunk vezérelni. Nagyon sok ilyen könyvtár van, a legfontosabbak már alapból be vannak építve az IDE-be, de ha mást akarunk, akkor egyszerűen be tudjuk tölteni őket. Az én projektemhez is használok ilyen könyvtárat.

A programkódot egy szöveges fájlként mentjük el, .ino fájlkiterjesztéssel, ezt hívják vázlatnak. A vázlat két alap eleme:

* setup(): Ezt egyszer hívjuk meg, amikor a program betölt, vagy újraindul. Itt adjuk meg a változókat, definiáljuk a ki és -bemeneti pinek-et, valamint a szükséges könyvtárakat.
* loop(): Miután a setup() funkció véget ér, ezen belül vezéreljük az Arduinot. Ezt többször meghívhatjuk, ezek alkotják a program ,,szívét”.

Mindenképp szerepelnie kell mindkét funkciónak a kódban, még akkor is ha az egyiket nem használjuk semmire sem. Egy egyszerű példa segítségével bemutatom a működését:



9. ábra Beépített led villogtatása [9]

Mint említettem az Arduino Nano pin kiosztásánál, a 13-as pin-hez csatlakozik egy board-ra integrált led. A példán szereplő kód segítségével ez a led 1 másodpercenként fel, majd le kapcsol.

## 1.2 Led szalag

Az Arduino után a következő fontos alkatrész a WS2812B LED szalag, hisz ez adja az egész projekt látványi lényegét.

### Általánosságban

A LED szalag egy felületre szerelt fénykibocsátó diódák (SMD LED-ek) és más alkatrészek által lakott rugalmas áramköri lap, amelynek általában öntapadós a hátsó része. Eleinte csak háttérvilágításként vagy dekoratív világításként használták, de ahogy fejlődött a technológia, egyre nagyobb teljesítményű ledek voltak elérhetőek, így azok fényereje is a sokszorosára nőtt. Ez azt eredményezte, hogy elkezdték felhasználni nagyobb teljesítményű világításként, UV-fények kibocsájtására, ipari előállításban, de még a növénytermesztésben is.

Több jellemző szerint csoportosítjuk ezeket a LED sorokat:

* vízállóság
* szín
* ragasztófelület
* beépített alkatrészek
* vezető feszültség, állandó feszültség
* vezérlés fajtája

Néhány ledsornál nincs figyelembe véve a vízállóság, de még így is IP20-as védettségük lehet a fizikai veszélyek miatt. Ezek általában védettek azt áramütéssel szemben, de azért előfordulhat, hogy valamilyen finom fém rövidre zárja, ezért kapja ezt a védettséget. A vízálló ledsorok általában hővezető epoxi vagy szilikon borítást kapnak. Akármilyen is legyen a led szalag borítása, mindenképpen rendelkezik az egyik oldalán egy két-oldalú ragasztóval, mellyel az objektumokhoz rögzíthetjük.

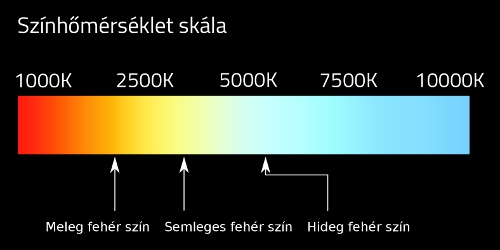
A leggyakoribb különbségek a led sor fajták között, hogy:

* az egyes LED-eket hogyan vezéreljük,
* milyen a ledsor színe
* címezhetőek-e vagy sem az egyes LED-ek

Négy fajta ledsort különböztetünk meg:

* Egyszínű, nem címezhető: A szálon lévő minden LED kizárólag fehér színű, jellemzően 2200K - 6500K (kelvin) a színhőmérséklet.

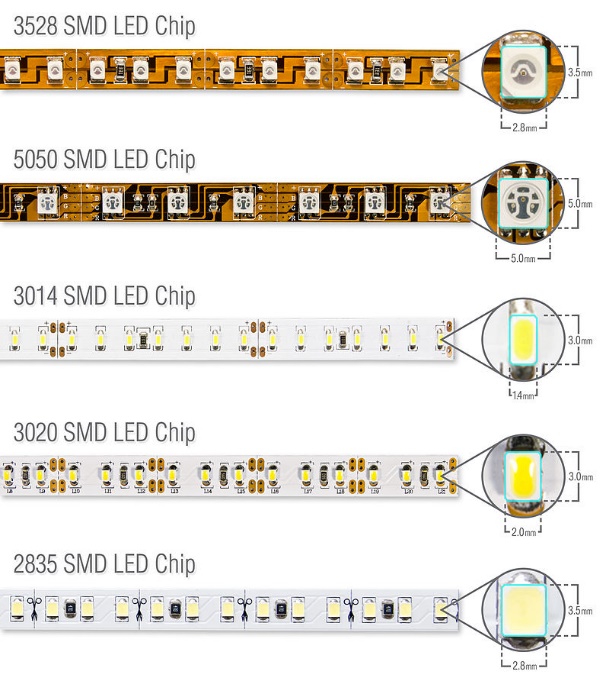
Egy fényforrás színhőmérsékletét az általa okozott színérzet és egy hipotetikus feketetest-sugárzó által létrehozott színérzet alapján határozzák meg.



. ábra Színhőmérséklet skála [10]

* Dinamikusan hangolható fehér, nem címezhető: Lehetővé teszi a felhasználó számára, hogy a színhőmérsékletet szabályozza. A led szalagon lévő ledek egyik fele magas színhőmérsékletű, másik fele pedig alacsony színhőmérsékletű. Ennek köszönhetően a két hőmérséklet közötti sávban bármilyen speciális színhőmérsékletet létre tud hozni.
* Többszínű, nem címezhető: Minden egyes led képes piros, zöld, kék, vagy egyben a három (fehér) szín kibocsájtására. Feszültség állítással tudjuk szabályozni, hogy mikor melyik színt jelenítse meg.
* RGB, címezhető: Mindegy egyes lednek van egy saját chip-je, ezzel mindegyiket külön-külön tudjuk vezérelni (színt váltani, effekteket váltani). Ide tartozik az általam használt WS2812B led szalag is.

Fontos különbség továbbá ezen led szalagok között, hogy milyen SMD led chipek (felületbe ágyazott chip) találhatók rajta.



. ábra Leggyakoribb LED szalag fajták, SMD chipek [11]

Több fajta ilyen chip van (8. ábra), lehetnek más színűek, címezhetőek, nem címezhetőek, de más-más a formájuk, méretük és a teljesítményük is. A leggyakoribb SMD típusok: 3528 (egyszínű, nem címezhető, nagyon alacsony teljesítményű); 5050 (RGB, címezhető, nagy teljesítményszint); 2835 (újabb egyszínű, hasonló elrendezés a 3528-hoz, de ez nagyobb teljesítményszinteket is bír).

Általában 12V vagy 24V feszültségszinten működnek, egyenárammal, valamely tápegység segítségével.

### WS2812B LED szalag

A WS2812B egy intelligensen vezérelhető led szalag, melyben a vezérlő áramkör és az RGB chip 5050-es SMD komponensekbe vannak integrálva. Az adatátviteli protokoll egyetlen NZR kommunikációs módot használ. A pixel bekapcsolása és visszaállítása után a DIN-port fogadja az adatokat a vezérlőtől. Az első 24 bites adatot az első pixel összegyűjti, majd elküldi a belső adatrögzítőre. A többi adatot a belső jelátalakító erősítő áramkör alakítja át, és a DOUT porton keresztül továbbítja a következő kaszkád pixelhez.



12. ábra Led-pinkiosztás, funkciók [12]

Legfőbb tulajdonságok és előnyök:

* a vezérlő áramkör és a ledek között oszlik el csak a tápfeszültség
* beépített jel-átalakító áramkör
* minden led 256-os fényerősséget bír, 16777216 db színt tud megjeleníteni, a beolvasási frekvencia nem kevesebb, mint 400Hz/s
* 5 méteres szalagon belül bármilyen jel átvihető a ledek között, áramkör fejlesztés nélkül
* az adattovábbítás 800Kbps sebességű

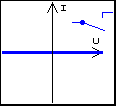


## 1.3 Billenőkapcsoló

13. ábra Szögletes billenőkapcsoló [13]

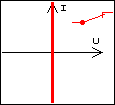
Az elektromos kapcsoló egy olyan áramköri elem, amely az elektromos áramot vagy átengedi, vagy nem. A kapcsolók nagyon különbözőek lehetnek a kapcsolható áram, és feszültség nagysága, a működtetés módja, illetve esztétikai kialakításuk szerint. A kapcsolók általában finommechanikai szerkezetek.

Az ideális kapcsoló:

Nyitott állapotban:

* vezetése 0
* ellenállása végtelen
* 0 áramú áramgenerátorként modellezhető,

14. ábra Kapcsoló nyitott állapotban [14]

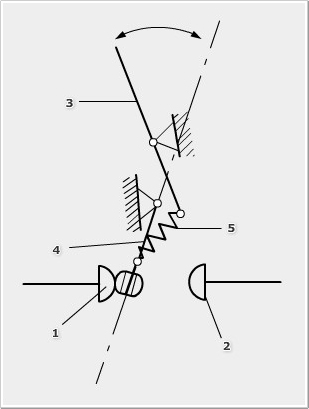
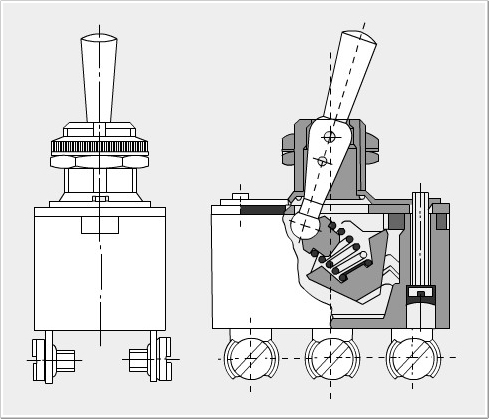
Zárt állapotban:

* vezetése végtelen
* ellenállása 0
* 0 elektromotoros erejű feszültséggenerátorral modellezhető.

15. ábra Kapcsoló zárt állapotban [15]

A kapcsolóknak az is feladatuk, hogy lehetőleg ne jöjjön létre bizonytalan helyzet, azaz a kapcsoló mozgó alkatrésze ne legyen képes megállni az érintkezők között. Ezért finommechanikai billenőszerkezeteket alkalmazunk, aminek tipikus példáját a 16. ábra mutatja. A 3. működtető kar elmozdításával az 5. nyomórugó még jobban összenyomódik, és amikor a működtető kar túllendült a semleges helyzeten (a semleges helyzetnél a három csapágyazási pont egybeesik), a 4. mozgó érintkező átugrik a 2. álló érintkezőre. Az érintkező nyomást áttételesen a nyomórugó biztosítja.

Egy tipikus billenő kapcsoló szerkezetét a 17. ábra mutatja. Ebbe a családba tartoznak a háztartási villanykapcsolók is, azzal a megjegyzéssel, hogy a gyártók gyakran nagyobb figyelmet szentelnek a külső megjelenésre, mint a billenő tulajdonságok biztosítására. A jó billenő tulajdonság azt jelenti, hogy ilyenkor a finommechanikai szerkezet súrlódási veszteségei kicsik, a csapágyazásoknál megjelenik az élágyazás és a rugalmas csapágyazás.



17. ábra Tipikus billenőkapcsoló szerkezete [17]

16. ábra Finommechanikai billenőszerkezeteket [16]

## C:\Users\karik\Desktop\zarodoga\1280px-400_points_breadboard.jpg1.4 Próbapanel

18. ábra Közönséges próbapanel [18]

A próbapanel egy áramköri lap, mely lehetővé teszi az áramkör prototípusának elkészítését. Fejlődéstörténeti szempontból a légszerelés utódja, de a nyomtatott huzalozású lemez közvetlen elődje.

## 1.5 470Ω-os ellenállás

Elektromos ellenállásnak (röviden ellenállásnak) nevezzük az elektromos vezető két pontjára kapcsolt feszültség és a vezetőn áthaladó áram erősségének a hányadosaként értelmezett fizikai mennyiséget. Jele a latin resistentia (=ellenállás) szó alapján R. Az ellenállások alapanyaga szén és fém vegyületek; ötvözetek. Jellemzőek a tömör anyagú és a huzalellenállások. Használt hordozó anyagok: műanyag és kerámia.



## 1.6 Áramellátás

-Tápegység

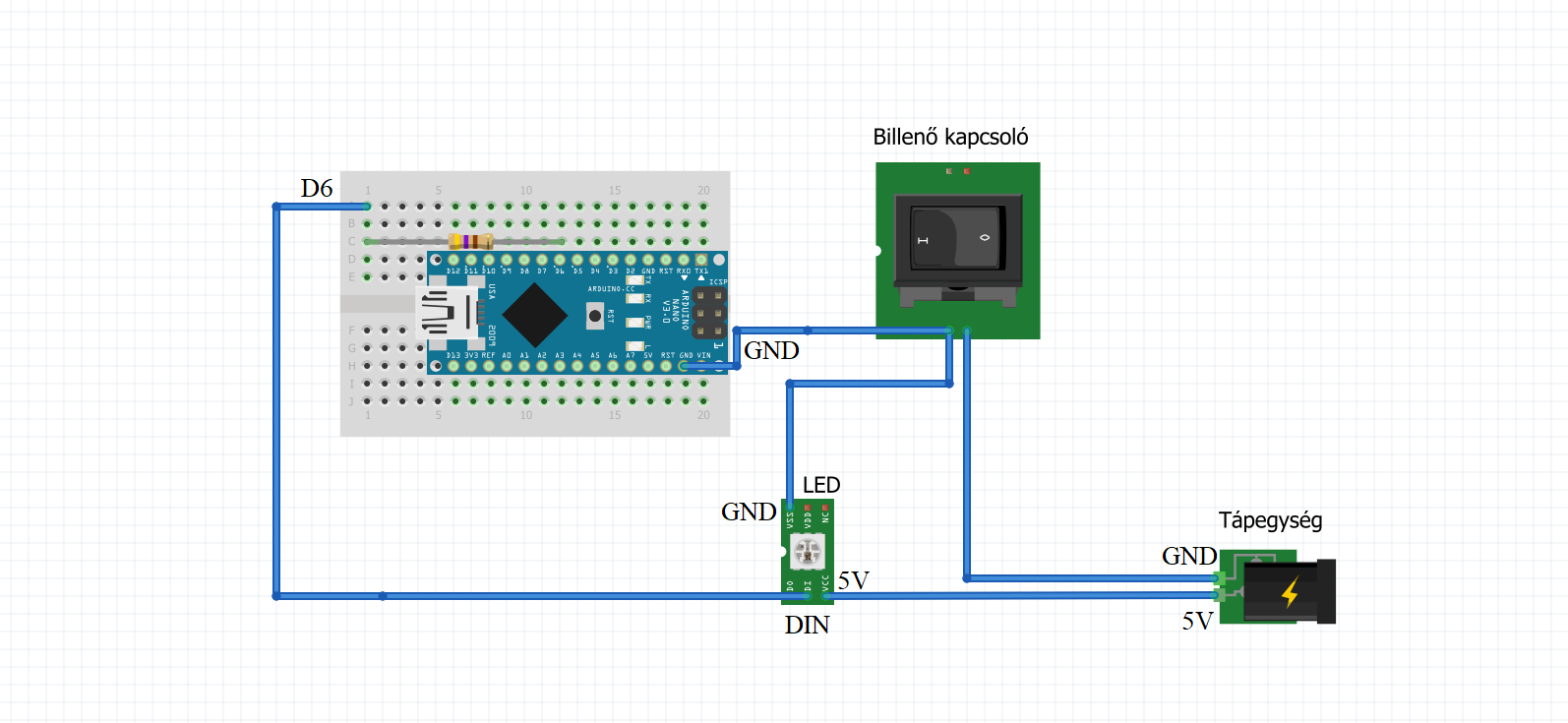
Egy 5V 3A-es tápegységet használok a LED szalag áramellátásához.

-Kábelek

4db egyszerű jumper kábelt használok az Arduino-LED szalag összekötéséhez, valamint a összedarabolt LED szalagokat is ilyenekkel kötöm össze.

# 2.A KAPCSOLÁSI RAJZ ÉRTELMEZÉSE, AZ ESZKÖZ VEZÉRLÉSÉNEK LEÍRÁSA

## 2.1 A kapcsolási rajz értelmezése



19. ábra Kapcsolási rajz

A 19. ábrán látható a kapcsolási rajz, mely mutatja, hogy az alkatrészek milyen kapcsolatban állnak egymással, és hogy hogyan vannak összekötve. A tápegységtől elindulva látható, hogy a LED és a tápegység 5V pinje össze van kapcsolva, így kap áramot a LED. Szintén a tápegységen, a GND pin össze van kötve a kapcsoló egyik lábával, amivel azt érjük el, hogy a kapcsoló rövidre zárja majd a tápegységet, így megszüntetve az áramforrást a ledhez. Ez úgy folytatódik, hogy a LED-en lévő GND pin össze van kötve a kapcsoló másik lábával. Ez azt eredményezi, ha a kapcsoló fel van kapcsolva, akkor zavartalanul folyik az áram a tápegységből a LED felé, ha viszont lekapcsoljuk, akkor megszakítjuk az áramellátást ugyanezen a szakaszon. Mint amiről már beszéltünk, a LED-en van egy DIN pin, ezen fog kommunikálni az Arduinoval. Ezt rákötjük az Arduino D6 pinjére (ez általunk választható, arra a digital pinre kötjük, amelyiket definiáltunk a programkódban).

## 2.2 Az eszköz vezérlésének leírása

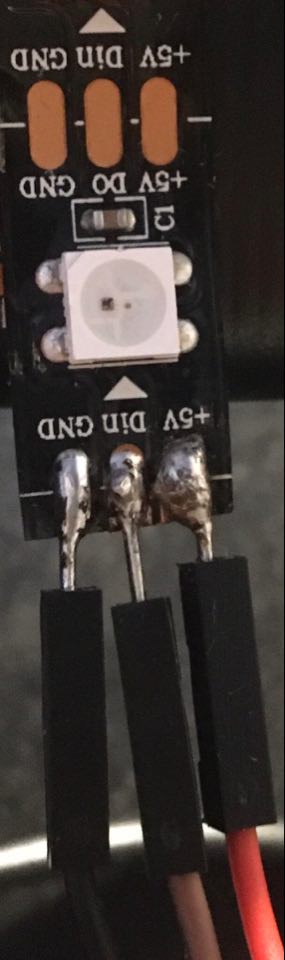
A vezérlést az Arduino végzi. Az általam megírt programon kívül, egy másik, előre megtervezett és bárki által letölthető program is szükséges. Ez fogja a képernyőt a ledekhez igazítani, lebontani szegmensekre, de erről később fogok írni a szoftveres részen.

# 3.AZ ESZKÖZ MEGÉPÍTÉSE

## 3.1 Az eszköz megépítésének lépései

Mindenekelőtt végig kellett gondolnom, hogy a saját monitoromhoz mekkora LED szalag fog kelleni. Egy 24”-os monitorról van szó, lemértem az oldalakat és megterveztem, hogy melyik oldalra hány darab led fog kelleni. Nagyon fontos, hogy a szemközti oldalakon meg kell egyezni a LED-ek számának. Esetemben a felső és alsó részre 31-31 darab led került, a két szélső oldalra pedig 17-17 darab. Az így összesen 96 darab lednek megfelelő hosszúságú led szalagot szereztem be (a biztonság kedvéért 1 méterrel hosszabbat). Ezután azt kellett kiszámolnom, hogy milyen tápegységre lesz szükségem. Mivel egy led fogyasztása 60mA (teljes fényerőn), így nekem megfelelő egy 5V 3A-es tápegység. Az 5V adott, mivel a led szalag ezen a feszültségszinten működik.

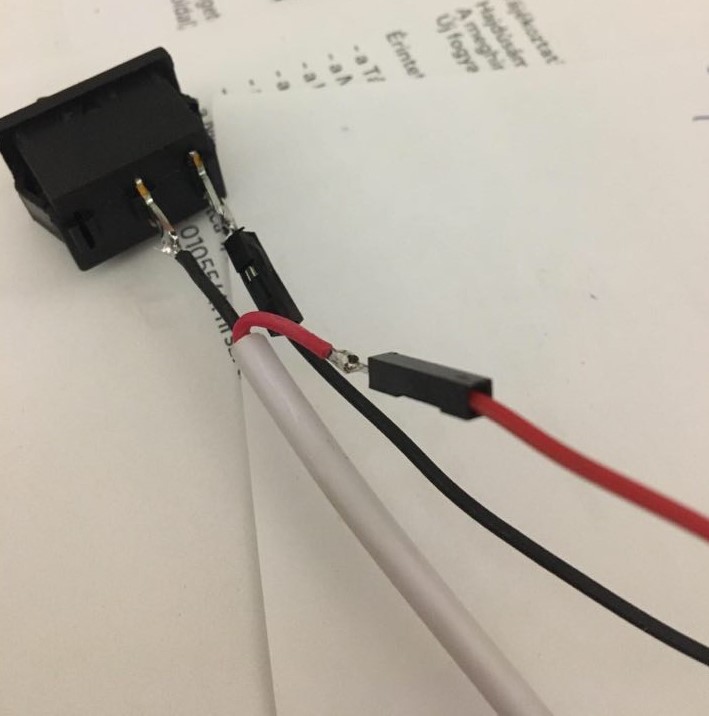
Miután beszereztem ezeket, az első dolgom az volt, hogy leteszteljem a LED szalagot, hogy biztos tökéletesen működik e. Először is ráforrasztottam a szalag 3 pinjére a kábeleket (20. ábra).



20. ábra Led pinek forrasztása

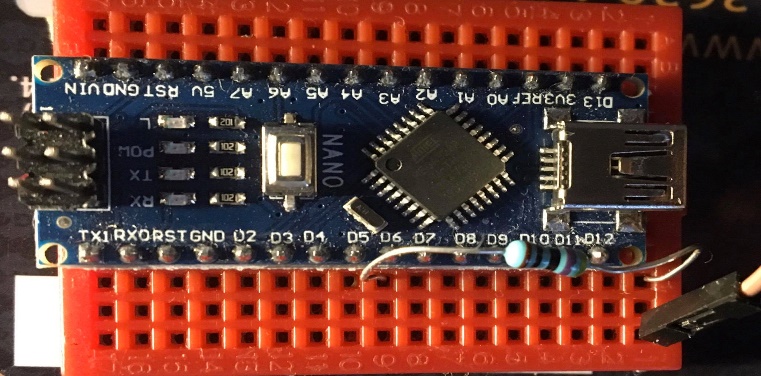
Rákötöttem a LED szalagot az Arduinora, és futtattam egy teszt kódot. Az eredmény sajnos nem a várt lett, a szalag egy része nem világított, mivel vagy szállítás közben megtört, vagy gyárilag hibás volt. Szerencsére (mivel hosszabb szalagot rendeltem), könnyen ki tudtam vágni a hibás darabot, így a többi részét tökéletes működésre tudtam bírni. Miután sikeresen teszteltem a szalagot, megfelelő darabokra vágtam és sorban elkezdtem a felhelyezést és a forrasztást. Az alsó részen kicsit nehezebb dolgom volt, mivel a monitor talpa miatt nem tudtam egy darab szalaggal megoldani, ezért 3 részből kellett összehoznom az alsó részt. Fontos dolog, hogy a szalagon találhatók nyilak, melyeket mindenképp azonos irányban, folytatólagosan kell felrakni. Az alsó rész után már csak a két felső sarkot kellett összeforrasztanom, de mégis itt nehezebb dolgom volt, mivel sokkal kisebb kábeleket kellett forrasztanom, sokkal kisebb helyen. Ezután újra teszteltem ugyanazzal a kóddal, amivel a legelején, hogy megbizonyosodjak arról, hogy minden kábel tökéletesen át viszi e a jelet. Aprólékos munka volt, valahol nem sikerült jól a forrasztás és újra kellett forrasztani.

Végül mindenhol jól működött, hozzá láthattam a tápegység és a szalag összekötéséhez, valamint a kapcsoló közéjük építéséhez. Szerencsére találtam itthon egy pont ideillő átalakítót, ezért magához a tápegységhez nem kellett hozzányúlni. Az átalakító egyik végébe bele lehet dugni a tápegységet, a másik végét pedig megblankoltam. A két benne lévő vezeték rákötése következett a kapcsolóra és a szalagra. A fekete színű vezetéket a kapcsoló egyik lábára forrasztottam rá, a piros színűt pedig a szalag 5V pinjére. A kapcsoló másik lábára a szalag GND pinjét forrasztottam rá. Így ezzel elértem, hogy kapcsolni tudjam a szalag áramellátását (21.ábra).



21. ábra Kapcsoló beépítése

Miután ezzel végeztem, belehelyeztem a próbapanelbe az Arduinot és rákötöttem a szalag DIN pinjét az Arduino D6 pinjére. Ezt egy 470Ω-os ellenálláson keresztül tettem meg, hogy védjem a szalagot (22. ábra).



22. ábra A panel összeállítása

## 3.2 Problémák az építésnél

Miután mindent beállítottam, és azt hittem kész az eszköz, kipróbáltam. Sajnos azzal szembesültem, hogy a legelső LED zölden villog, valamint működés közben véletlenszerű színek villogtak, teljesen máshogy nézett ki, mint ahogy kellett volna. Hosszas keresés és gondolkodás után sikerült rájönnöm, hogy mi okozta a hibát. Az Arduino GND pinje nem volt rákötve a szalag GND pinjére, ez okozta a villogást és a rossz színek megjelenítését. Ezt úgynevezett Ground Potenciálnak nevezzük. A mellékletben csatolt egyik videón látható, hogy milyen jelenség jött elő ekkor.

# 4.A SZOFTVERES RÉSZE A PROJEKTNEK

## 4.1 Prismatik nevű program

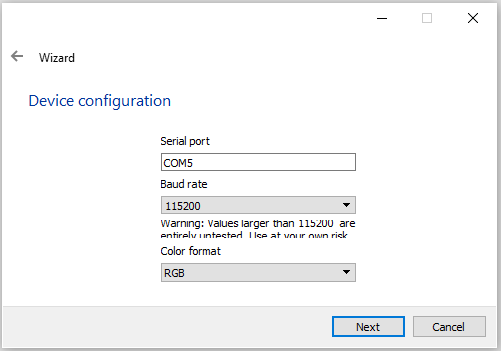
A Woodenshark nevű programozó csapat fejlesztette ki a programot. Ez segítséget nyújt két különböző LED technológia megvalósítására.

23. ábra A Prismatik logója

* Ambilight háttérvilágítás
* Hang alapú világítás

Ebben a programban kell beállítani, az általam összerakott eszköz paramétereit:

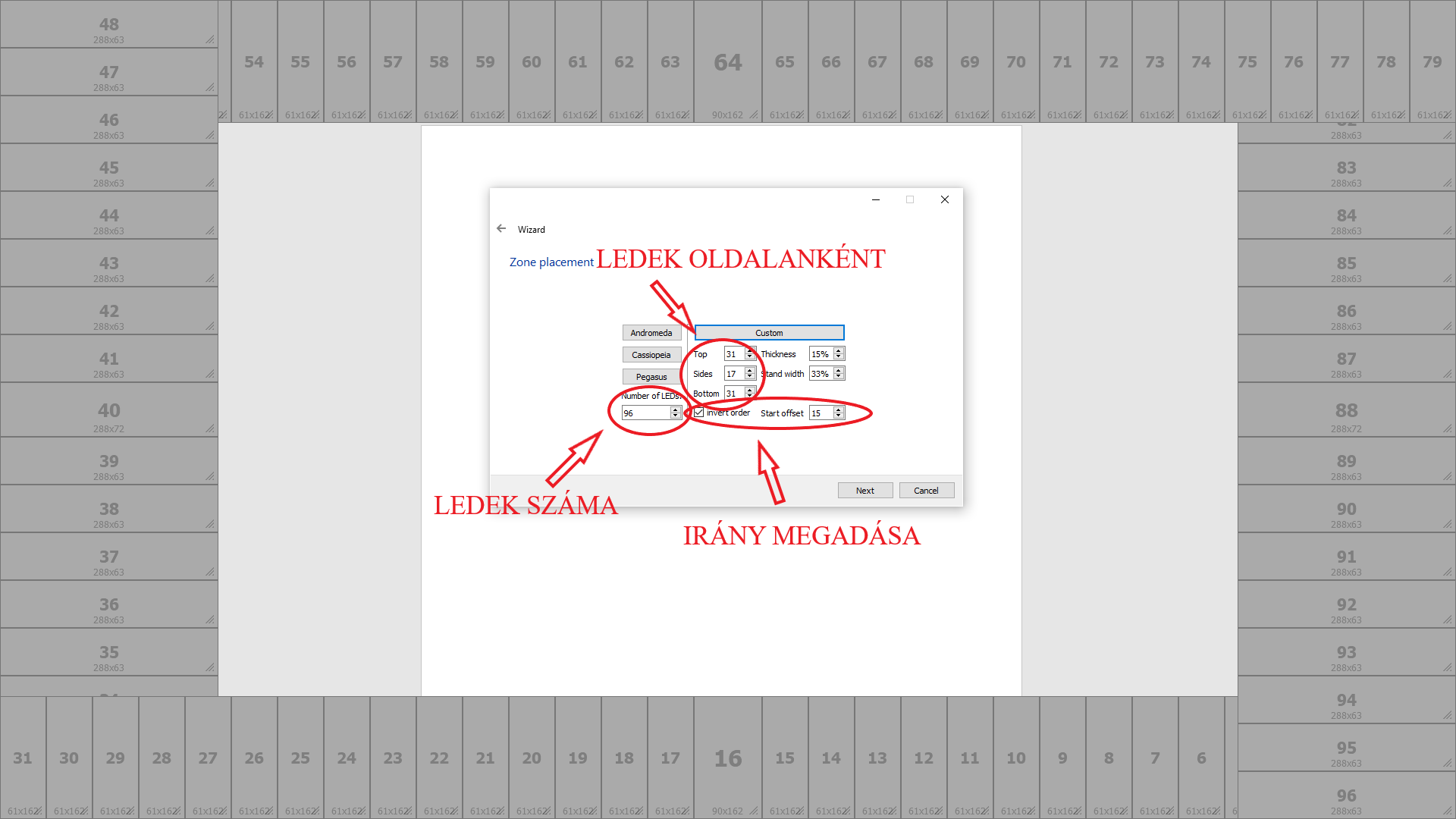
* melyik soros porton keresztül érjük el az Arduinot
* milyen baud rate-et állítunk be (hány jelet továbbítunk 1 másodperc alatt)
* milyen színformátumot állítunk be
* hány darab led van az egyes oldalakon
* milyen sorrendben és hol kezdtük el a felszerelést

Esetemben a COM5-ös soros portot kell kiválasztani, mert arra van csatlakoztatva az Arduino. A baud rate 115200 (teszteltem nagyobbaknál is, de változás nem volt, néha villogott). Fontos hogy az itt lévő baud rate-nek és a programban megadottnak meg kell egyezniük. Valamint a színformátum egyszerű RGB, mivel a szalag is ezt tudja (24. ábra).

24. ábra Paraméterek megadása

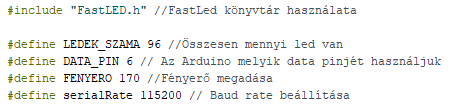
Ezután állítjuk be a felszerelt eszköz paramétereit. Vannak előre elkészített sablonok, de nekem a Custom (egyéni)-t kell használni. Itt adjuk meg, hogy hány darab led van, melyik oldalra mennyi jut. Az irányt is itt adjuk meg, hogy honnan indultunk. Esetemben 96 LED-ről van szó. Alsó-felső oldalakon 31-31 led van, szélső oldalakon pedig 17-17. Végül megadjuk, hogy honnan indulunk. Az ábrán látható, hogy a jobb alsó sarokból indul, és így megy körbe a led szalag. A képernyő oldalain megjelenő 96 darab kis szegmens a ledeket fogja szimbolizálni. Ezek a beállítások láthatók a 25. ábrán.

Összefoglalva, ez a program teszi lehetővé, hogy minden egyes lednek legyen egy saját kis szegmense a monitoron, és mindegyik azt a színt tudja mutatni, amit mi elvárunk.

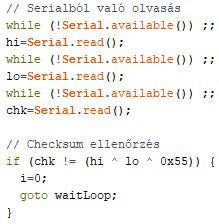


25. ábra Eszköz paraméterek beállítása

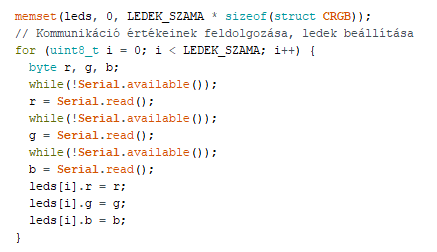
## 4.2 Az eszköz programkódja, lényeges része

Lényegében nem túl hosszú a programkód, mindössze 77 sorról van szó. Ez amiatt van, mivel a legnagyobb részét a Prismatik program végzi. A mellékletben csatolt kódban minden részt megjegyzésben jelöltem, hogy milyen feladatuk van az eszköz működésében. Alább csak a kód lényegesebb részéről írok.

Az elején meghívom a szükséges könyvtárakat, definiálom a fontosabb változókat és értéket adok nekik.



Itt a serial portról beolvassa, hogy milyen színek vannak a képernyőn, valamint ezeket ellenőrzi.

Itt a beolvasott értékeket elküldi a szalagra, és így kapjuk meg a megfelelő színeket.

# ÖSSZEFOGLALÁS

Mielőtt elkezdtem keresgélni, hogy milyen vonalon kellene elindulnom, nem igazán hallottam erről a technológiáról. Több Arduino projektet is átnéztem, de egyik se nyerte el úgy a tetszésem, mint az Ambilight. Komolyabban beleástam magam a témába, rengeteg megoldási módot átnéztem, de szerintem mindenképp sikerült a számomra legmegfelelőbbet kiválasztani. Szerettem volna nem teljesen ugyanazt megcsinálni, mint amik fent vannak az interneten, hanem kisebb módosításokat is végezni rajta. Külön sikerélménynek tartom, hogy meg tudtam oldani egy saját átalakítóval az áramellátást, valamint hogy beépítettem egy kapcsolót is, hogy kedvemre fel vagy le kapcsoljam a szalagot. Voltak nehézségek, az elején a hibás led szalag, a végén a ground pin bekötésére is nehéz volt rájönni, amely a villogást okozta. Nem volt egyszerű a forrasztás, főleg a sarkokon ahol nagyon kicsi hely volt rá. A végső eredménnyel nagyon meg vagyok elégedve, tökéletesen működik, nagyon feldobja egy-egy film vagy videó hangulatát, valamint játékoknál is jól néz ki. Szeretném fejleszteni a projektet, ha lesz egy kis szabadidőm, akkor TV-re is meg szeretném csinálni –mivel a mellékletben található videón látható, hogy most még csak a monitoromat ékesíti az eszköz- ezt az Ambilight világítást. Összességében szerintem egy nagyon érdekes és látványos projektet sikerült választanom.



. ábra A működő eszköz

# FELHASZNÁLT IRODALOM

<https://hu.wikipedia.org/wiki/Arduino#/media/F%C3%A1jl:Az_Arduino_TTL_szintjei.png> [1]

<https://learn.sparkfun.com/tutorials/arduino-shields/all> [2] [3] [4] [5] [6]

<https://imgaz2.staticbg.com/thumb/large/oaupload/banggood/images/4D/75/bfacef5e-9ca0-4650-85a3-87a23e196292.JPG> [7]

<https://www.keywild.com/arduino/gallery/Nano_PinOut.png> [8]

<https://en.wikipedia.org/wiki/Arduino> [9]

<http://promika.hu/index.php?s=LED&c=15> [10]

<https://en.wikipedia.org/wiki/LED_strip_light> [11]

<https://www.seeedstudio.com/document/pdf/WS2812B%20Datasheet.pdf> [12]

<http://pbcomputer.hu/termekeink/miniatur-szogletes-billenokapcsolo-14x8mm-1-aramkor-2-lab-2-allasu> [13]

<https://hu.wikipedia.org/wiki/Kapcsol%C3%B3> [14] [15]

<https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2010-0017_37_finommechanikai_epitoelemek/ch06.html#id525890> [16] [17]

<https://en.wikipedia.org/wiki/Breadboard> [18]

<https://www.instructables.com/id/DIY-Ambilight-Using-Arduino-Nano/>

<https://logout.hu/bejegyzes/bullzeye/ambilight.html>

<https://hu.wikipedia.org/wiki/Atmel_AVR>

<https://en.wikipedia.org/wiki/Arduino_IDE>

# MELLÉKLETEK

1. sz. melléklet: Programkód

2. sz. melléklet: Rosszul működő eszköz (videó)

3. sz. melléklet: Kész, működő eszköz (videó)