



Szkenner megvalósítása egér szenzorral

Készítette

Bodnár Máté

Programtervező informatikus BSc

Témavezető

Dr. Geda Gábor

Egyetemi docens

EGER, 2024

Tartalomjegyzék

Bevezetés	3
1. Bevezető	4
1.1. Motiváció	4
1.2. Célkitűzés	4
2. Felhasznált technológiák	5
2.1. Arduino	5
2.1.1. Arduino platform bemutatása	5
2.1.2. Miért választottam az Arduino UNO-t?	6
2.1.3. Arduino UNO	6
2.1.4. Az Arduino alkalmazási területei	8
2.2. Visual Studio	8
2.3. Github	8
3. Hardveres megvalósítás	9
3.1. ADNS-9800 szenzor	9
3.1.1. Működése	9
3.1.2. Adatok beolvasása	10
3.1.3. Adatok továbbítása a Visual Studio felé	13
3.1.4. Szenzor mozgása	13
3.2. Hardveres bekötés	13
4. Szoftveres megvalósítás	14
4.1. Beolvasott értékek tárolása 2 dimenzós mátrixban	14
4.2. Bikubik interpoláció	14
4.2.1. Működése	14
4.3. Mátrix átalakítása képpé	14
Összegzés	15
Irodalomjegyzék	17

Bevezetés

A digitalizálás egyre nagyobb szerepet játszik az életünkben, különösen a dokumentumok kezelésében és tárolásában, hiszen a sok ideig tárolt papír dokumentumok elveszhetnek vagy könnyen sérülhetnek.

Manapság a családi fotókat vagy régebbi képeket is célszerű digitalizálni, hogy tovább megmaradjon. A digitalizálással lehetőségünk nyílik arra is hogy rendszerezzünk hivatalos iratokat, így könnyebben tudunk majd hozzájuk férni és akár keresni is közöttük.

Bár számos szkennert van a piacon, viszont ezek drágák és nagy helyigényűek. Ezért egy saját készítésű szkennert jobban illeszkedik majd a mi igényeinkhez, valamint sokkal olcsóbb mint egy boltban vett.

Ebben a projektben megmutatom hogy lehet egy egérszenzort felhasználni hivatalos iratok vagy egyéb más papír alapú dokumentumok szkennelésére. A szenzor két sínnel vízszintesen és függőlegesen fog mozogni, és soronként készíti el a képkockákat. Ezután egy C# alkalmazás feldolgozza azokat és ki alakít belőlük egy nagy dokumentumot.

1. fejezet

Bevezető

1.1. Motiváció

Az ötletem mögött több tényező is áll. Elsősorban szerettem volna egy olyan eszközt létrehozni, amely megfizethető alternatívát nyújt a drága szkennerekkel szemben. Az egérszenzorok könnyen beszerezhetők és viszonylag olcsók, ezért jó választásnak tündtek egy saját fejlesztésű szkennerekhez. Ez különösen hasznos lehet olyan helyeken, ahol a költségek csökkentése nagyon fontos, például iskolákban vagy kisebb cégeknél.

Mindig is érdekelt, hogyan lehet egy meglévő egyszerű technológiát új és kreatív módon felhasználni. Az egérszenzorokat alapvetően mozgásérzékelésre tervezték, de ebben a projektben megszeretném mutatni, hogy dokumentumok szkennelésére is alkalmasak.

Emellett fontos számomra, hogy egy olyan eszközt alkossak, amelyet egy egyszerű felhasználó is használhat, otthon vagy akár a munkahelyén anélkül, hogy drága berendezésekre kellene költenie.

Valamint kihívást látok ebben a projektben, hogy hogyan is tudom ezt megvalósítani. Izgalmas feladat az, hogy ötvözzem az informatikát az elektronikával. Ez nem csak a szakmai tudásomat fejleszti, hanem egy olyan eredményt ad, amelyre büszke lehetek.

1.2. Célkitűzés

A szakdolgozatom célja, egy olyan szkennerek létrehozása, amely egy egyszerű egérszenzort használ a dokumentumok fekete-fehér beolvasására. Az eszköz működésének alapja, hogy a szenzor monokróm felvételeket készít a dokumentumról, majd ezeket egy számítógépes programmal feldolgozom és összeállítom egy nagy dokumentummá. Mivel a szenzor csak egy 30 x 30-as képet rögzít így azt interpolációval felnövelem. Ezek az algoritmusok lehetővé teszik, hogy a képet megnöveljük kevés minőség veszteséssel.

A dolgozat eredményeként egy egyszerű és költséghatékony szkennert szeretnék létrehozni, amely hasznos és megkönnyíti az emberek életét.

2. fejezet

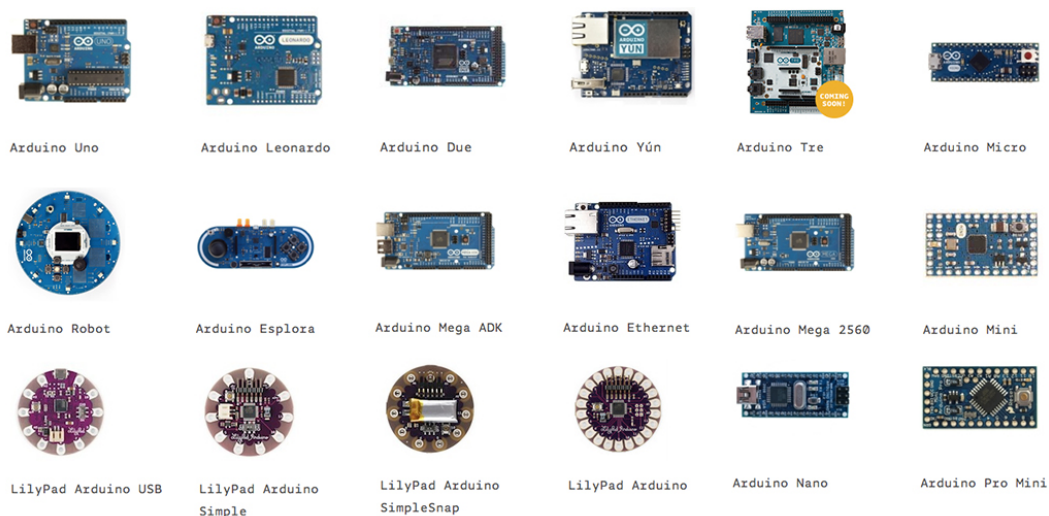
Felhasznált technológiák

Ebben a fejezetben a szakdolgozatomban használt technológiákról és azok előnyeiről, fogok beszámolni.

2.1. Arduino

2.1.1. Arduino platform bemutatása

Az Arduino[1] egy nyílt forráskódú platform amiket az elektronikai projektekhez találtak ki, majd bekerült az oktatásba is, oktatási céllal. Sokan használják egyszerűbb feladatok automatizálására vagy akár okos otthon rendszerek kialakítására. Ezek mellett manapság már az ipari alkalmazásuk sem ritka. A működéséhez szükség van egy mikrokontrollerre valamint egy fejlesztő környezetre az Arduino IDE-re, amivel általában USB kábelben keresztül tudjuk átküldeni a programot a fizikai eszközre.



2.1. ábra. Arduino család néhány fajtája

2.1.2. Miért választottam az Arduino UNO-t?

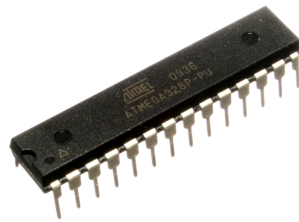
Szakdolgozatom során az Arduino UNO mellett döntöttem mivel több olyan előnye van ami fontos a projektem során.

- **1. Egyszerű és széles körűen elterjedt és támogatott.** Széles körben elérhetőek könyvtárak és példakódok amik megkönnyítik a fejlesztést. Valamint ez a legjobban dokumentált mikrokontroller a felhasználók között.
- **2. Megfelelő teljesítmény és I/O portok.** Ez a mikrokontroller egy ATmega328P mikrovezérlőt használ mely megfelelő arra hogy kezeljem az egérszenzort és az RGB LED-et, valamint az adatok egyszerű továbbítására.
- **3. Megbízhatóság és stabilitás.** A stabilitás a projektem szempontjából nagyon fontos hisz hosszabb távon kell hiba mentes működést nyújtania, hiszen dokumentumok szkennelése közben folyamatos adatgyűjtésre és adatátvitelre van szükségem.
- **4. Költséghatékonyság.** Ez az egyik legolcsóbb fejlesztői eszköz, amely a kellő igényeket kielégíti a projektemhez.
- **5. Egyszerű programozhatóság valamint egyszerű kapcsolat.** Könnyen programozható mikrokontroller az Arduino IDE szoftverrel, és egyszerűen USB-n keresztül lehet áttölteni rá programokat.

2.1.3. Arduino UNO

Ez volt az egyik legelső és leginkább elterjedt mikrokontroller. Ezeknek az eszközöknek a szíve és lelke egy ATmega328P mikrovezérlő melyben található:

- Processzor
- Memória
- Különböző perifériák:
 - Időzítő áramkörök
 - Analóg és digitális be és kimenetek
 - Kommunikációs perifériák és még sok egyéb

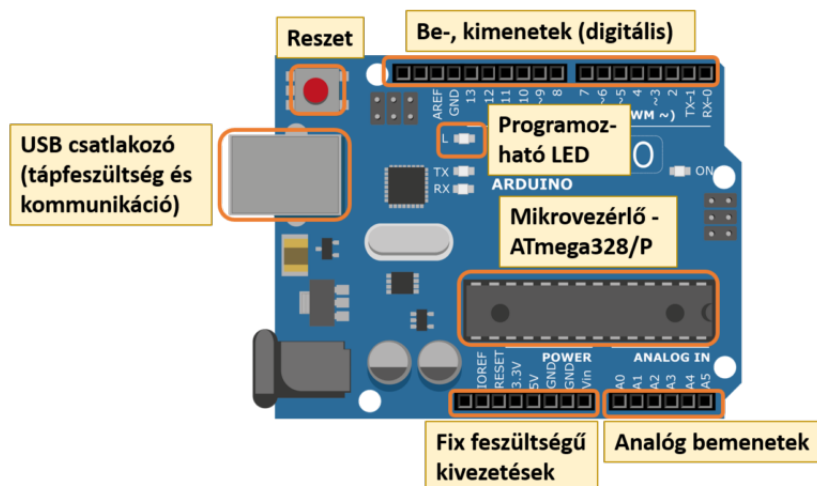


2.2. ábra. ATmega328P mikrovezérlő

Ezek segítségével tudunk szenzorjeleket mérni, nyomógombok vagy más beviteli eszközök állapotát beolvasni. Az UNO áramkör szerepe hogy a mikrovezérlőnek a lábait kivezesse. Így kényelmesebben és egyszerűbben rá tudjuk kötni a különböző eszközöket amiket vezérelni szeretnénk, vagy értékeket beolvasni róluk.

A mikrokontrollereken általában nem szokott futni operációs rendszer, ezért minden erőforrást a feladatra összpontosít és egy garantált maximális idő alatt képes végrehajtani a feladatokat.

Ahhoz hogy külső eszközöket és áramköröket rátudjunk csatlakoztatni, ismernünk kell az UNO lábkiosztását, amikre a kódból tudunk hivatkozni és vezérelni őket. Az Arduino UNO áramköri lapon fel van tüntetve, hogy melyik lábat melyik számmal érjük el. A kimenetek lehetnek digitálisak 0–13-ig vagy analóg bemenetek A0–A5-ig. Az analóg bemenetek működhetnek digitális ki vagy bemenetként, ez attól függ, hogy hogy állítjuk be a kódban. Van fix feszültséget leadó kivezetés is amelyen akár 3.3 V-ot vagy 5 V-ot is leadhatunk. Ezeket szemlélteti a 2.3. számú ábra.



2.3. ábra. Az Arduino UNO részei

2.1.4. Az Arduino alkalmazási területei

- **Kezdő projektekhez.** Az Arduino Board-ok tökéletesek a kezdők számára akik az elektronikát és az informatikát szeretnék ötvözni. A fejlesztő környezetet egyszerű kezelni valamint a könyvtárak és a példa kódok is nagyon sokat segítenek azoknak az embereknek akik elkezdnek érdeklődni az ilyen dolgok iránt.
- **Oktatási platform.** Könnyen kezelhetősége miatt szokták alkalmazni, hogy ezekkel az eszközökkel tanítsák meg az elektronika és az informatika működését.
- **Robotikában.** A nagy vállalatok vezetői is felfedezték ezt a technológiát, és gyakran Arduino Board-okat használnak a robotok megvalósításához és vezérléséhez.
- **Zene és művészet.** Az Arduino Board-okat szokták egyszerű hangszerek létrehozására is alkalmazni, vagy már meglévő hangszerekbe beépíteni elektronikus alkatrészként.
- **IoT - Internet of Things.** Legtöbb esetben okos otthon rendszerekbe szokták beépíteni, mert sok szenzort rá tudunk csatlakoztatni amikkel könnyen vezérelhetjük a saját otthonunkat.
- **Viselhető eszközök. (Wearables)** Kompakt méretük miatt könnyen beépíthetőek ruhákba, akár ékszerekbe vagy más hordozható eszközökbe. Amelyekkel mozgásokra reagálhatunk mérhetünk testhőmérsékletet és még sok más.

2.2. Visual Studio

A projekt szoftveres részét Visual Studio[2] fejlesztői környezettel készítettem el, mert ez áll hozzám a legközelebb. Itt valósítottam meg az interpolációt a kép minőségvesztés nélküli növelését. Valamint a program irányító felületét is itt programoztam le.

2.3. Github

A verziókövetéshez a Githubot[3] használtam azon belül is a Github Desktop-ot, mellyel könnyen tudtam több platformon is dolgozni a projekten, valamint ha valamit elrontottam könnyen vissza tudtam állítani a projektet egy korábbi verzióra.

3. fejezet

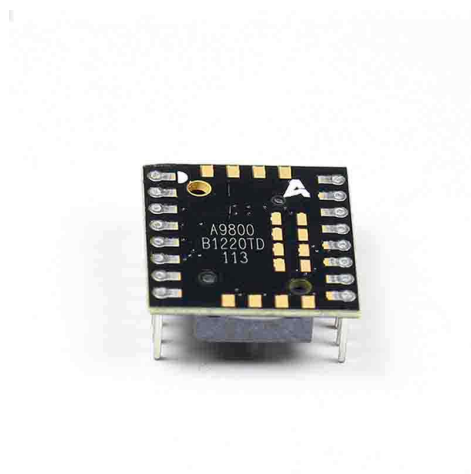
Hardveres megvalósítás

3.1. ADNS-9800 szenzor

Az általam készített szkennernek a lelke a kicsi és olcsó ADNS-9800 lézeres optikai érzékelő. Az eredeti felhasználási módja ennek az eszköznek az volt, hogy egerek mozgását kövessék. Ennek a szenzornak az a különlegessége, és azért volt megfelelő számomra mivel úgy követi a mozgást, hogy folyamatosan képeket készít. Ezáltal fel tudtam használni a projektembe mint egy kis kamera. Valamint gyors képfeldolgozási képességgel rendelkezik, ezért alkalmas dokumentumok beolvasására.

3.1.1. Működése

Egy beépített lézer meg világítja a szenzor alatt lévő felületet majd a lencse készít egy képet. A beépített kis processzor dolgozza fel ezeket az adatokat és számítja ki, hogy az eszköz milyen irányba mozdult el.



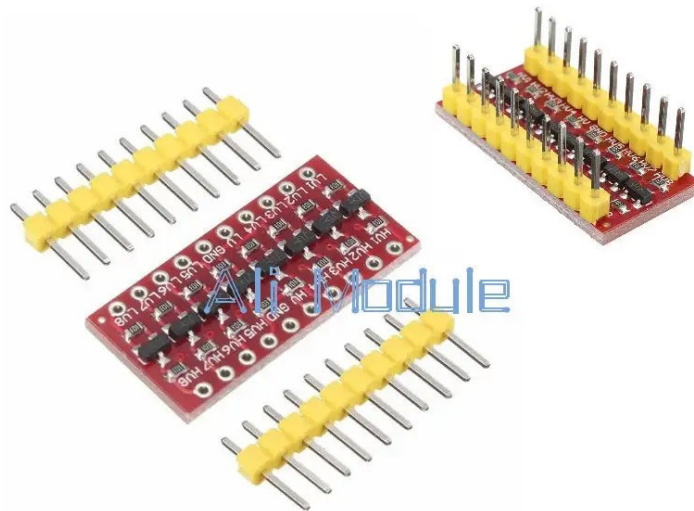
3.1. ábra. ADNS-9800 szenzor

Az általam választott szenzor már nyáklappal volt ellátva így azzal nem volt teendőm. Ezáltal egyszerűbb volt a bekötése is, és az elhelyezése is.



3.2. ábra. ADNS-9800 szenzor nyák lapra szerelve

Csak egy problémám volt, hogy a nyáklappal szerelt verzió 3.3 V-os logikai szinttel működött viszont az Arduino UNO 5 V-os szinttel. Így be kellett szereznem egy két irányú logikai szintillesztő modult ami átalakította a 3,3 V-os jelet 5 V-ossá. Ehhez a projekthez egy 8 csatornás logikai szintillesztő modult használtam.



3.3. ábra. Két irányú logikai szintillesztő

3.1.2. Adatok beolvasása

Az adatok beolvasása közben a szenzor egy sorokból és egy oszlopokból álló képkockát készít, amely egy 30 x 30-as pixel rács. Ezáltal egy dokumentum szkennelése lehet akár több ezer képkocka is. A működése a következő lépésekből áll:

1. **Feltöltjük a firmware-t a szenzorra.** Erre azért van szükség, mert enélkül a szenzor csak egy hardveres érékelő, amely a lézerrel megvilágított felületről képeket készít. Ahhoz, hogy ezt az információt feldolgozza és továbbítsa szüksége van egy vezérlőprogramra, azaz firmware-re. Ezt a firmware-t minden bekapcsolás után fel kell tölteni mert a szenzornak nincs állandó memóriája. A firmware feltöltését a 3.1. számú metódus végzi. A regiszterek megfelelő beállítása után a firmwaret soronként továbbítja az SPI buszon keresztül a szenzornak. Ebben a metódusban fontos az időzítés hogy a szenzor helyesen értelmezze az adatokat.

kód 3.1. Firmware feltöltése

```
1 void adns_upload_firmware() {
2     adns_write_reg(REG_Configuration_IV, 0x02);
3     adns_write_reg(REG_SROM_Enable, 0xd);
4     delay(10);
5     adns_write_reg(REG_SROM_Enable, 0x8);
6     adns_com_begin();
7     SPI.transfer(REG_SROM_Load_Burst | 0x80);
8     delayMicroseconds(15);
9     unsigned char c;
10    for(int i = 0; i < firmware_length; i++){
11        c = (unsigned char)pgm_read_byte(firmware_data + i);
12        SPI.transfer(c);
13        delayMicroseconds(15);
14    }
15    adns_com_end();
16 }
```

2. Majd a **sendFrame()** metódust meghívva készítünk egy képet a szenzorral. Amit egy nagy sebességű, full-duplex kommunikációs protokollal (Serial Peripheral Interface – SPI) kiolvasunk és eltároljuk, ahol a kommunikáció egy órajel vezérelt mechanizmussal működik.

kód 3.2. sendFrame metódus

```
1 void sendFrame() {
2     adns_write_reg(REG_Power_Up_Reset, 0x5A);
3     delay(50);
4     adns_write_reg(REG_LASER_CTRL0, 0x00);
5     adns_write_reg(REG_Frame_Capture, 0x93);
6     delayMicroseconds(120);
7     adns_write_reg(REG_Frame_Capture, 0xc5);
8     delayMicroseconds(120);
9     delay(1);
10    adns_com_begin();
11    delayMicroseconds(100);
12    byte readys = 0;
13    while(readys == 0){
14        SPI.transfer(REG_Motion & 0x7f);
15        delayMicroseconds(100);
16        readys = SPI.transfer(0);
17        readys = readys & 1;
18        delayMicroseconds(20);
19    }
20    SPI.transfer(REG_Pixel_Burst & 0x7f);
21    delayMicroseconds(100);
22    Serial.print("FRAME:");
23    for (int i = 0; i < 900; i++){
24        byte pixelValue = SPI.transfer(0);
25        Serial.print(pixelValue);
26        Serial.print("_");
27    }
28    delayMicroseconds(15);
29    adns_com_end();
30    delayMicroseconds(5);
31    Serial.println();
32 }
```

A metódus működése:

1. **Szenzornak kiadjuk, hogy készítse egy képet.** Ekkor bele írjuk a megfelelő értéket a REG_Frame_Capture regiszterbe. A szenzor ekkor egy 30 x 30 pixeles képkockát olvas be.
2. **Várunk amíg a szenzor elkészíti a képet.** Az Arduino egy ciklusban folyamatosan ellenőri a szenzor állapotát ezért amíg a szenzor nem jelzi, nem olvassuk ki az értéket.
3. **A szenzor továbbítja a képkockát.** A Pixel Burst regiszteren keresztül történik a képkocka eljuttatása a szenzortól az Arduino-ig SPI kapcsolattal. Az SPI kapcsolat négy vezetéken keresztül történik:
 - **MOSI** (Master Out, Slave In): Az adatok a mikrokontroller felől a szenzor felé mennek.
 - **MISO** (Master In, Slave Out): Az adatok a szenzor felől mennek az Arduino felé.
 - **SCLK** (Serial Clock): Az órajel, amelyet az Arduino generál.
 - **CS** (Chip Select): Az adott szenzort választja ki a kommunikációhoz.
4. **Az Arduino továbbítja az adatokat a számítógépre.** Az adatok soros kommunikáción keresztül kerülnek a számítógépre ahol feldolgozom az adatokat. Minden képkocka "FRAME:" előtaggal kezdődik, hogy könnyebben elkülöníthetőek legyenek a képkockák.

3.1.3. Szenzor mozgása

A szenzort két motor fogja mozgatni az egyik vízszintesen a másik függőlegesen.

3.2. Hardveres bekötés

smartdraw, circuitikz

4. fejezet

Szoftveres megvalósítás

kell még egy az arduinohoz az arduino szenzor kezelés és szenzor mozgató egy alkalmazás amiről tudom kezelni a szkennelést

4.1. Beolvasott értékek tárolása 2 dimenzós mátrixban

adatszerkezet amiben a beolvasott képet tároljuk

4.2. Bikubik interpoláció

4.2.1. Működése

Működésének alapjai, Matematikai leírása

4.3. Mátrix átalakítása képpé

Összegzés

Tapasztalatok amiket szereztem a projekt megvalósítása közben Továbbfejlesztési gondolatok

színes vagy szürke képet szeretne beolvasni soros porton küldök egy bitet hogy színes vagy szürke legyen a kép a studiobol felbontásra vonatkozóan például feles átfedéssel

Források

- 2.1. ábra: <https://predictabledesigns.com/wp-content/uploads/2017/10/HeroImage.png>
- 2.2. ábra: <https://techfun.hu/wp-content/uploads/2017/09/11.jpg>
- 2.3. ábra: <https://www.inf.u-szeged.hu/~makan/tananyagok/wp-content/uploads/2020/07/ArduinoParts3-768x462.png>
- 3.1. ábra: <https://www.sicstock.com/cdn/shop/products/551554.jpg?v=1544514465>
- 3.3. ábra: https://www.elektrobot.hu/items/3607_1.webp

Irodalomjegyzék

- [1] Arduino: *<https://www.arduino.cc/en/Guide/Introduction>*
- [2] Microsoft Visual Studio: *<https://visualstudio.microsoft.com/>*
- [3] Github: *<https://github.com/>*

Nyilatkozat

Alulírott *Bodnár Máté*, büntetőjogi felelősségem tudatában kijelentem, hogy az általam benyújtott, *Szkenner megvalósítása egér szenzorral* című szakdolgozat önálló szellemi termékem. Amennyiben mások munkáját felhasználtam, azokra megfelelően hivatkozom, beleértve a nyomtatott és az internetes forrásokat is.

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

Eger, 2025. március 3.

aláírás