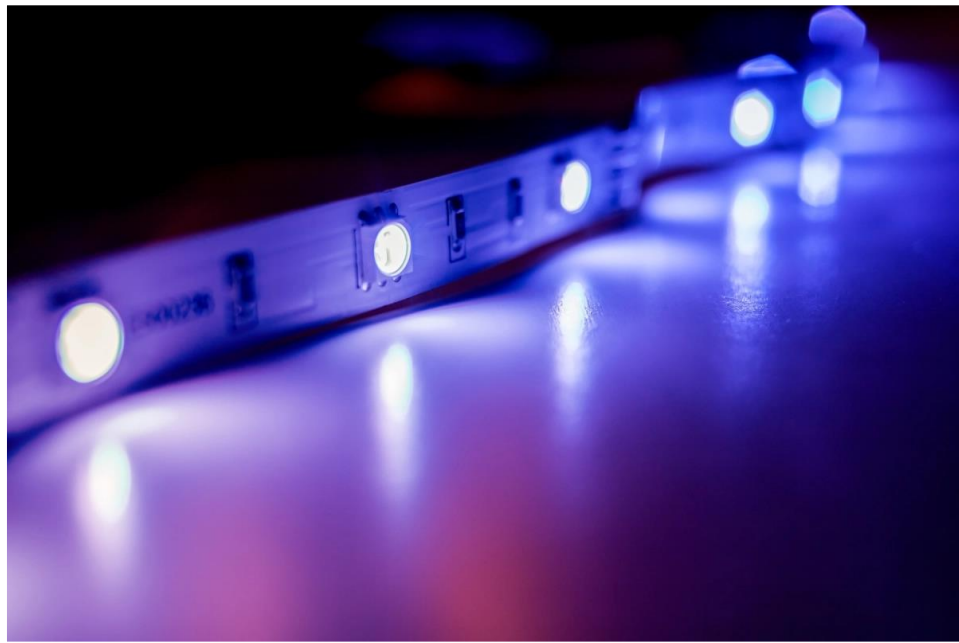


Eline Biernaux



2021-2022

Geïntegreerde proef

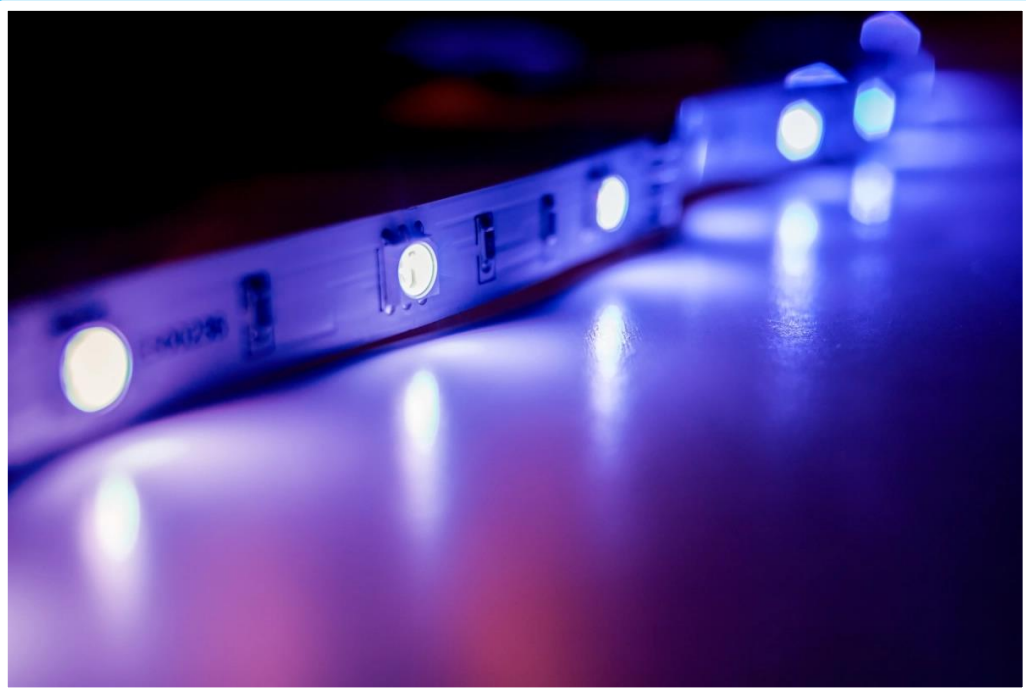
Studierichting: 6ICT

Begeleiders:

*De heer Meyers, de heer Grates-
solle, mevrouw Vos*

PROVIL
UNIEK IN STEM & SPORT!

Eline Biernaux



2021-2022

Geïntegreerde proef

Studierichting: 6ICT

Begeleiders:

*De heer Meyers, de heer Grates-
solle, mevrouw Vos*

PROVIL
UNIEK IN STEM & SPORT!

Voorwoord

Deze paper is tot stand gekomen naar aanleiding van de geïntegreerde proef aan het Provinciaal Instituut Lommel. Deze bevat verschillende vakken die aangeboden worden op school zoals programmeren, Nederlands, Engels of elektronica. Dit project zal getuigen van de competenties die opgebouwd werden in de richting Industriële Informatiecommunicatietechnologie.

De interesse in programmeren en de liefde voor muziek brachten het idee een hulpmiddel te ontwikkelen om op een eenvoudige manier piano te leren spelen. Dit is mogelijk door middel van een ledstrip en een Windowsapplicatie die doorheen het jaar verwezenlijkt werd.

In de eerste plaats wil ik graag mijn vader, de heer B. Biernaux, bedanken voor zijn geduld en inzicht. Hij heeft mij stap voor stap leren omgaan met moeilijkheden die ik tijdens het ontwikkelingsproces ondervond. Mijn dank gaat ook uit naar de heer F. Meyers en de heer C. Gratesolle die dit project tijdens de schooluren hebben begeleid. Natuurlijk betuig ik ook mijn dank aan al de andere betrokkenen die mij hebben bijgestaan in dit uitdagende, maar fijne eindwerk.

Eline Biernaux
Juni 2020

Abstract

Eline Biernaux, Industriële Informatie Communicatietechnologie, Provinciaal Instituut Lommel, Abstract of research paper, submitted 24 June 2022: How to learn people play the piano through technology

The aim of this project is to develop a means which enables the user to learn how to play the piano without any knowledge of sheet music. The Piano Aid is a tool for all ages to learn playing the piano in an innovative, but above all simple way. The appliance will work with LED lighting that will show the user which keys to press in the precise moment. This will be supported by a software package that visually stimulates the user in the learning process. In this way, learning is not only efficient but also fun from the beginning onwards.

In the first chapter of this paper, the problems that may occur will be analyzed. The solutions to those problems will be described in detail in the next chapters. The Piano Aid will consist of an LED strip that is attached to a frame. This frame will be made of a light material and will be further explained in the next chapter. In chapter three, the LED strip will be connected to an extra power supply and to an Arduino that will control it. The code to control the Arduino will be described in chapter four. Furthermore, this entire project is supported by a self-written Windows application which will take care of the midi files, will do all the calculations and will make everything visible for the user in chapter five.

If this product may be prepared for the market, then it is important that all cabling is concealed in the frame. The frame must therefore be made out of a stronger material. The design of the application needs to be improved and could later appear on a tablet or smartphone. To make the application more user-friendly, it would certainly be a surplus if staves are shown on the screen while playing a piece of music. This way, the user can also learn notes while playing the piano.

If I would do this project again, I would start researching faster and especially start writing this paper sooner because of the amount of work it was. But I have learned that no project is too big with the right motivation and passion. Everything has its ups and downs and sometimes you just have to get through it!

Inhoud

Voorwoord	3
Abstract	4
Inhoud	5
Inleiding	7
1 Probleemstelling	8
2 Behuizing	10
2.1 <i>Frame</i>	10
2.1.1 Uitmeten en tekenen van het frame	10
2.1.2 Eerste frame uitwerken	10
2.1.3 Ledstrip bevestigen	11
3 Elektronica	12
3.1 <i>Leds</i>	12
3.1.1 Ledstrip kiezen	12
3.1.2 Ledstrip aansluiten	12
3.1.2.1 Ledstrip aansluiten op Arduino	12
3.1.2.2 Externe bron	14
3.1.3 Ledstrip testen	15
3.1.4 Testprogramma	15
3.2 <i>Communicatie</i>	16
3.2.1 Arduino of RaspberryPI	16
3.2.2 Website	16
3.2.3 Web-USB	16
3.2.4 Applicatie	17
4 Applicatie	18
4.1 <i>Midifile</i>	18

4.2	Werking	19
4.2.1	Songloader.....	19
4.2.2	PlaySong.....	20
5	Ledsturing	21
	Besluit	23
	Afkortingen en vakjargon	24
	Verwijzingen.....	I
	Bijlagen 1	III
	Bijlagen 2	IV
	Bijlagen 3	V

Inleiding

Muziek is een heel belangrijk deel van ons leven. Ze bepaalt onze gemoedstoestand en kan een treurige dag weer opbeuren. Zou het echter niet fijn zijn zelf muziek te kunnen maken? Natuurlijk! Dat wilt toch iedereen? Nochtans heeft niemand zin om eerst jaren te spenderen in een muziekschool om dan na een jaar pas 'Broeder Jakob' te kunnen spelen op de piano. Gelukkig is daar een hele goede oplossing voor en daar gaat deze geïntegreerde proef dan ook over.

De Piano Aid is een hulpmiddel om jong en oud op een innovatieve, maar vooral eenvoudige manier piano te leren spelen. Het toestel zal werken met ledverlichting die de gebruiker laat zien op welke toetsen hij of zij de vingers moet plaatsen op het juiste moment. Dit wordt verder ondersteund door vernieuwende software die de gebruiker visueel meeneemt op het beeldscherm. Op deze manier is het leren niet alleen efficiënt, maar ook leuk vanaf het begin.

In het eerste hoofdstuk worden de problemen die zich kunnen voordoen geanalyseerd. In de hoofdstukken die volgen worden de oplossingen voor deze problemen uitvoerig beschreven. De Piano Aid zal bestaan uit een ledstrip die bevestigd is aan een frame. Dit frame zal vervaardigd worden uit een licht materiaal en zal verder toegelicht worden in het hoofdstuk Behuizing. In hoofdstuk drie zal de ledstrip aangesloten worden op een extra voeding en de Arduino die deze ledstrip zal aansturen. Deze aansturing wordt door simpele Arduinocode gedaan en zal worden beschreven in hoofdstuk vier. Verder wordt dit hele project ondersteund door een zelfgeschreven Windowsapplicatie die alle berekeningen maar ook visualisatie voor zijn rekening zal nemen. Dit deel zal verder toegelicht worden in hoofdstuk vijf.

In deze bundel zult u verder meegenomen worden in het ontwikkelingsproces. Zal dit project tot een goed einde gebracht worden of eindigt het in een kakofonie? Daar komt u zo achter.

1 Probleemstelling

Dit eindwerk bestaat uit vier onderdelen die nauw met elkaar samen zullen werken om een mooi geheel te kunnen vormen. Allereerst wordt er een frame voorzien waar een ledstrip op gemonteerd wordt. De ledstrip dient aangestuurd te worden met een microcontroller die op zijn beurt aangestuurd wordt door Windowssoftware. Deze toepassing zal alles visueel aan de gebruiker laten zien en zal ook alle berekeningen doen.

De volgende problemen zullen in deze bundel een oplossing krijgen

- Behuizing
 - Frame/prototype 3D-tekenen
 - Frame uitwerken
 - Ledstrip bevestigen
- Elektronica
 - Arduino aansturen onderzoeken
 - Circuit aansluiten en tekenen
 - Extra voeding voor de ledstrip aansluiten
- Sturing leds
 - Leds aansturen
 - Noten koppelen aan leds
 - Leds per noot laten schijnen
 - Leds in akkoorden laten schijnen
 - Verschillende kleuren per hand
- Applicatie
 - Twee standaard muziekstukken kunnen afspelen (later misschien database met meer muziek)
 - Simpel Design front-end
 - Knoppen (vooruit, achteruit, pauze, Play, sneller, trager)
 - Programmeren met Visual studio
 - Noten visueel laten zien (notenbalken?)
 - Website op pc (later op tablet of zelfs app ontwikkelen)

2 Behuizing

De Piano Aid zal bestaan uit een ledstrip die de gebruiker laat zien waar hij of zij de vingers moet plaatsen. Maar de leds blijven niet vanzelf boven de piano hangen. Daarom wordt er een frame voorzien die de ledstrip op zijn plaats zal houden.

2.1 Frame

De snelste en eenvoudigste manier om dit te kunnen verwezenlijken, is om gebruik te maken van maquettekarton. Het lichte maar toch stevige materiaal is ideaal om het eerste prototype uit te vervaardigen en bij te stellen waar nodig.

2.1.1 Uitmeten en tekenen van het frame

Om het frame zo gemakkelijk mogelijk te kunnen fabriceren en later uit ander materiaal te kunnen maken, is het belangrijk een sjabloon te tekenen dat goed past op de toetsen. De toetsen meten blijkt niet de juiste oplossing te zijn omdat de piano meten een lastige opdracht is. Een meetlat past niet tussen de toetsen en een flexibel meetinstrument is niet nauwkeurig genoeg. Dit zorgt ervoor dat het frame uiteindelijk niet mooi op de toetsen past.

De tweede poging is echter wel raak. Door een strook papier op de toetsen te leggen en streepjes te trekken tussen de toetsen krijgen we een realistischer beeld van waar de gleuven moeten komen (figuur 1). De Aid moet vier octaven¹ lang worden om handig te blijven. Zo is hij niet te lang om te kunnen vervoeren, maar wel lang genoeg om verschillende muziekstukken te kunnen spelen.

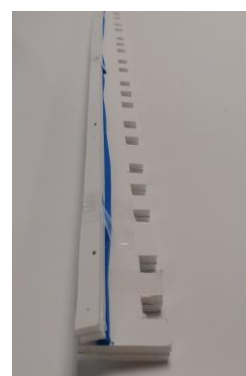


Figuur 1: tweede poging uitmeting piano

2.1.2 Eerste frame uitwerken

Als het sjabloon dan eindelijk klaar is en past op de piano, wordt het frame tweemaal overgetekend op maquettekarton, waarna het wordt uitgesneden en op elkaar wordt geplakt met plakband. Zo kan het later nog makkelijk gedemonteerd kan worden.

Ook twee extra stroken van 1cm breed worden uitgesneden en bevestigd aan de bovenkant met maquettenaalden en plakband. Deze worden later gebruikt om de leds op te bevestigen.



Figuur 2: eerste prototype

¹Een **octaaf** (van Latijn: octavus, de achtste) is in de muziektheorie het interval in een diatonische toonladder tussen een eerste toon en de daarboven liggende achtste. Bv. Van 'do' tot 'do'.

Later bleek dat de leds een beetje schuin moesten staan om duidelijk op de toetsen te kunnen schijnen. Daar zijn de blauwe streken voor op figuur 2. Hier wordt meer over weergegeven tijdens het testen van de leds.

2.1.3 Ledstrip bevestigen

Een ledstrip heeft aan de onderkant standaard dubbelzijdige tape hangen, maar deze wordt niet gebruikt vanwege de simpele reden dat dit nog niet het definitieve frame is. Daarom wordt de ledstrip met gewone plakband bevestigd aan de onderkant van het frame, zodat iedere led op een pianotoets kan schijnen.

3 Elektronica

De elektronica van de Piano Aid bestaat uit een ledstrip die aangestuurd wordt door een microcontroller. De ledstrip heeft ook een extra voeding nodig zodat alles veilig kan werken. Deze drie onderdelen worden dan ook aangesloten en getest.

3.1 Leds

3.1.1 Ledstrip kiezen

De juiste ledstrip kiezen is zeer belangrijk. Er zijn namelijk veel varianten op de markt waarvan de afstand tussen de leds, de lichtsterkte of het aantal kleuren kan verschillen.

Voor dit project is het erg van belang dat iedere led afzonderlijk aanstuurbaar is. Zo kan elke noot apart belicht worden. De kleur van de leds is niet zo belangrijk. Nochtans zijn RGB leds wel aangeraden als er een specifieke kleur gewenst is. Het is dus gewoon een leuke extra.

Zoals al eerder vermeld is het van belang dat de ledstrip het juiste aantal leds heeft. Iedere piano heeft zo'n 73 witte en zwarte toetsen over een afstand van een meter. Dat wilt dus ook zeggen dat de ledstrip zoveel leds per meter moet bevatten. De beste keuze is een ledstrip van 144 leds per meter, want zo worden alle leds beschenen op een kleine afwijking na. Deze ledstrip was in het begin van het jaar echter niet aanwezig, dus zijn alle tests van de ledsturing verlopen met een ledstrip van 60 leds per meter.

3.1.2 Ledstrip aansluiten

De ledstrip van 60 leds per meter heeft drie aansluitpinnen. Dat zijn de bronzige streepjes (figuur 3). In het midden van deze streepjes kunnen de leds geknipt worden om de strip korter te maken of meer leds aan te bevestigen.



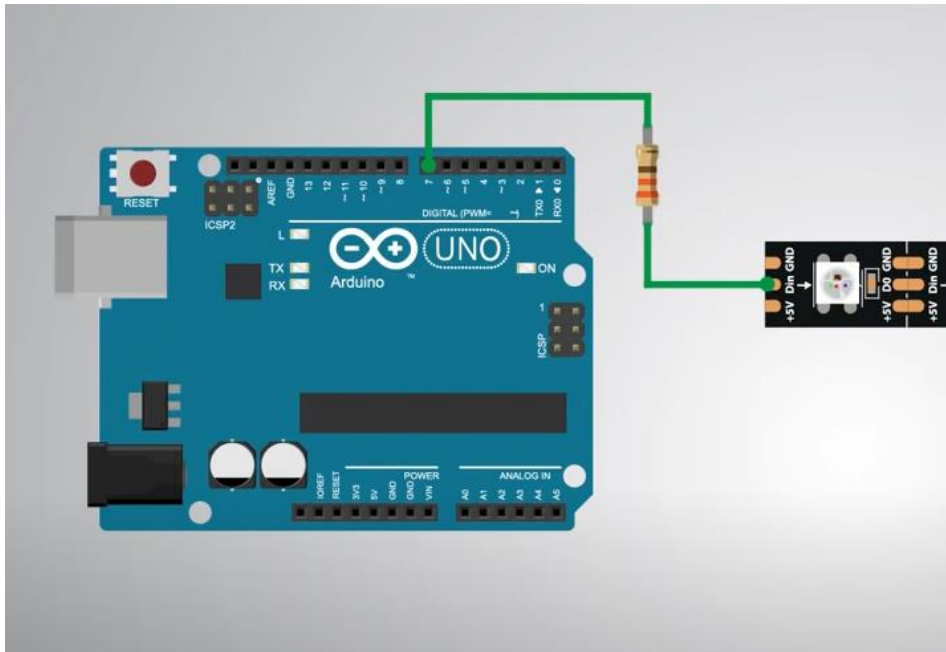
Figuur 3: led pinnen

Bij de bovenste pin staat GND. Dit staat voor Ground of de aarding en hangt dus aan de 0 Volt. De tweede of middelste pin is de datapin. Hier worden de signalen door gestuurd zodat iedere led weet of hij wel of niet moet branden. De laatste pin is de 5 Volt pin.

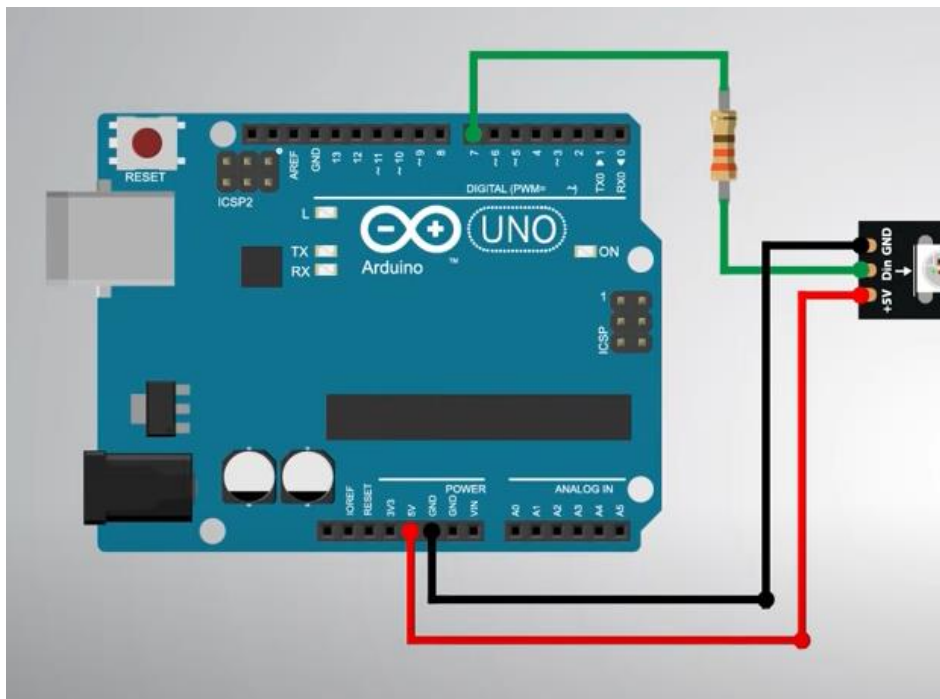
3.1.2.1 Ledstrip aansluiten op Arduino

De datapin wordt aangesloten op één van de uitgangen van de Arduino, bijvoorbeeld op pin 7 (zie figuur 4). Voor de zekerheid wordt er een weerstand van 220Ω tussen geplaatst. Dit is een standaardwaarde voor dit soort ledstrips. In principe is een weerstand enkel nodig als de kabel tussen de Arduino en de ledstrip erg lang is. Zo zouden er storingen door reflecties kunnen ontstaan (Majenko, 2020). In dit geval is de kabel slechts enkele centimeters lang. Hij is dus niet echt van belang.

Vervolgens worden ook de GND en de 5V aangesloten op de Arduino (Figuur 5).

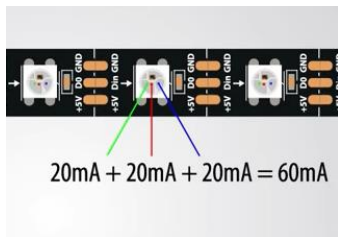


Figuur 4: aansluiten Digital Pin (Nedelkovski, 2018)



Figuur 5: GND en 5V aansluiten op Arduino (Nedelkovski, 2018)

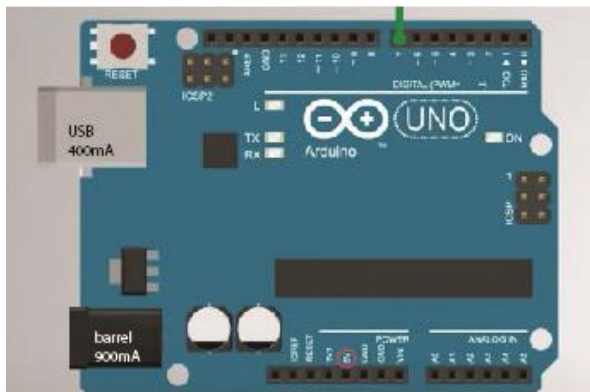
3.1.2.2 Externe bron



Figuur 4: RGB led (Nedelkovski, 2018)

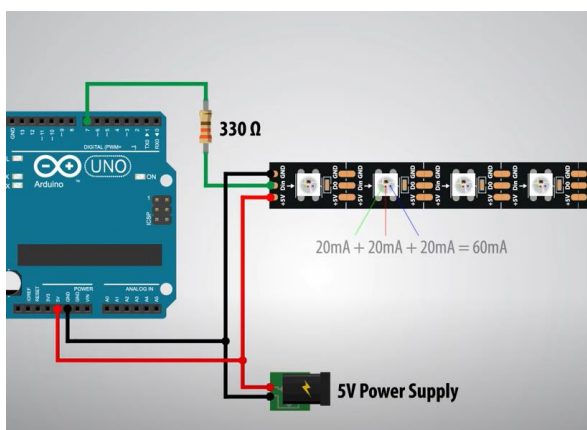
In iedere led van de ledstrip zitten drie aparte deelledjes: een rode, een groene en een blauwe led. Vandaar de naam RGB. Elk deelledje heeft 20 milliampères nodig om optimaal te kunnen branden. Elke led heeft dus met andere woorden 60 mA nodig (zie figuur 6).

Als de Arduino gevoed wordt met de USB-connector kan de 5V-pin ongeveer 400 milliampères leveren terwijl de “barrel”-voedingskabel zo een 900mA kan leveren (zie figuur 7). De voedingslijn van de strip is in parallel schakeling aangesloten. (Zie datasheet ws2812b). Dat wilt zeggen dat de totaalstroom die de ledstrip nodig heeft gelijk is aan de som van alle deelstromen die door de leds vloeien. Er kunnen maximum 15 leds tegelijkertijd op volle helderheid branden. Daarom zal de maximumstroom ongeveer 3 ampère bedragen. Het probleem is bijgevolg dat de Arduino zelf de ledstrip niet zal kunnen voeden en zelfs misschien zou kunnen doorbranden als er teveel leds tegelijk branden.

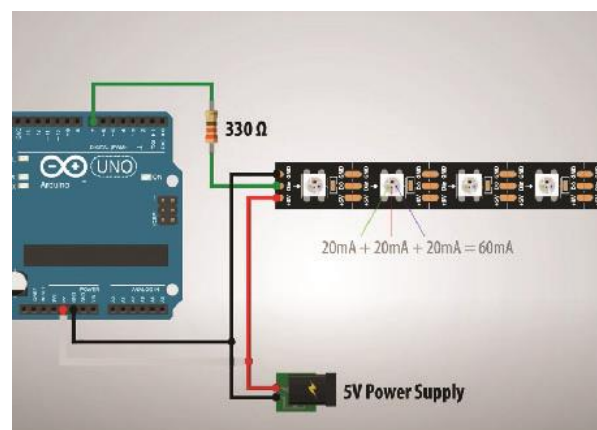


Figuur 5: connectoren

Om dit probleem op te lossen wordt er een externe bron toegevoegd aan de schakeling om extra stroom te kunnen sturen naar de leds (zie figuur 8). Om de Arduino te beveiligen op overbelasting wordt later de verbinding tussen de externe bron en de 5V pin weggehaald zodat de bron enkel de ledstrip voorziet van stroom en spanning (zie figuur 9).



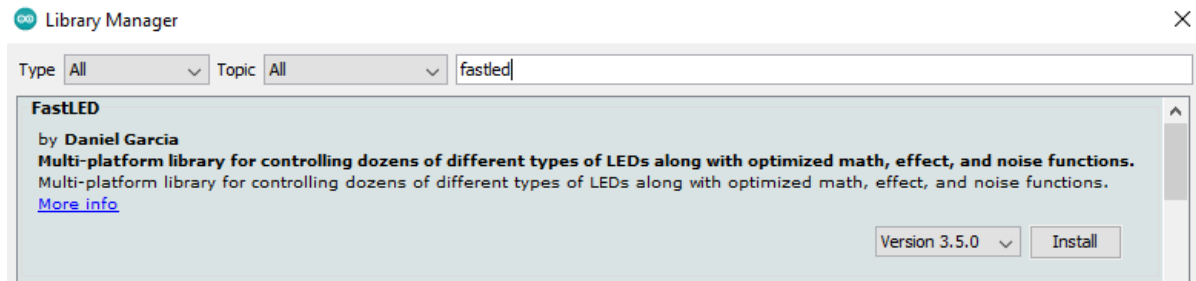
Figuur 7: toegevoegde externe bron



Figuur 6: Arduino gescheiden van externe bron

3.1.3 Ledstrip testen

Nu de ledstrip en de Arduino aangesloten zijn, kan het testen beginnen. De Arduino-app biedt allerlei handige voorgeprogrammeerde libraries die terug te vinden zijn onder 'sketch' -> 'Include Library' -> 'Manage Libraries' zoals te zien op figuur 8.



Figuur 8: Instalatie FastLED Library

FastLED is een makkelijke, efficiënte en snelle Arduino library voor het programmeren van adresseerbare ledstrips en pixels zoals de WS2810, WS2812, Neopixel en nog veel meer. Deze library wordt gebruikt door duizenden developers, voor hobbyprojecten en commerciële producten. Deze wordt dan ook geïnstalleerd.

3.1.4 Testprogramma

Onderstaand testprogramma wordt gebruikt om te testen of de ledstrip juist is aangesloten. Led 3 en 5 zullen in een bepaalde RGB-kleur branden.

```
test1 | Arduino 1.8.16
File Edit Sketch Tools Help
test1
#include <FastLED.h>
#define LED_PIN 7
#define NUM_LEDS 22
CRGB leds[NUM_LEDS];
void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
}
void loop() {
  leds[3] = CRGB(255, 0, 0);
  FastLED.show();
  delay(500);
  leds[5] = CRGB(0, 0, 255);
  FastLED.show();
  delay(500);
}
```

Figuur 9: test ledstrip met FastLED (Nedelkovski, 2018)

3.2 Communicatie

De Piano Aid wordt aangestuurd door hardware en software. De communicatie tussen deze twee onderdelen moet dus vlot verlopen aangezien de muziek-data snel tot bij de ledstrip moet komen zodat de gebruiker een fijne leerervaring heeft.

3.2.1 Arduino of RaspberryPI

De Arduino en de RaspberryPI zijn de meest geschikte microprocessors² voor kleinschalige projecten als deze. In tegenstelling tot de Arduino kan de RaspberryPI veel data verwerken. Je kan hem als het ware vergelijken met een computer op broekzakformaat. Zo kan hij sensors uitlezen en moeilijke berekeningen doen. Toch is de Arduino een betere optie voor dit eindwerk omwille van het feit dat een ledstrip aansturen niet veel rekenkracht vraagt. Hij is daarbij ook makkelijker te programmeren dan een RaspberryPI.

3.2.2 Website

Om alles visueel te maken zal er gebruik gemaakt worden van een website. Het is een makkelijke manier en daarbij ook een techniek die op school aangereikt wordt. De website zal commando's naar de Arduino sturen, zodat hij de juiste leds kan laten branden. Het is dus belangrijk dat de communicatie tussen deze twee onderdelen vlot verloopt.

3.2.3 Web-USB

De Arduino moet aangestuurd worden vanaf een website. Hiervoor wordt er gebruik gemaakt van de web-USB library. Het probleem is echter dat de Arduino UNO niet is uitgerust met een web-USB. Dat wilt zeggen dat de Arduino niet in staat is om met een server te connecteren.

Hier komt de Leonardo in beeld. Deze Arduino wordt volgens de Arduinosite wel ondersteund met een web-USB. Het punt is alleen dat hij geen aparte chip aan boord heeft die voor de communicatie tussen de USB-poort en de seriële interfase zorgt, zoals de uno. Dat wilt zeggen dat wanneer de geschreven code gecompileerd wordt om doorgestuurd te worden naar de Arduino, er automatisch een extra stukje code aan toegevoegd wordt zodat de Leonardo weet wat hij moet doen. (Pert, 2020)

Dit zorgt voor veel problemen. Zelfs het kleinste foutje in de geschreven code kan ervoor zorgen dat de seriële poort van de Leonardo 'dicht gemetst' wordt (Pert, 2020). Kortom, de Arduino wordt niet meer herkend door de computer en wil geen data meer ontvangen omdat de bootloader overschreven is. Het resetten of herbranden van de bootloader verandert niets.

Een website connecteren met een Arduino is dus niet zo makkelijk als het lijkt. Tijd voor wat anders.

3.2.4 Applicatie

Vandaag de dag is Windows het meest verkochte *operating system* op aarde. Vandaar de keuze om hier een app voor te schrijven. De populairste Windows ontwikkelingsframeworks zijn Universal Windows Platform, Windows Presentation Foundation en Windows Forms. Voor dit project zal er gebruikgemaakt worden van Windows Forms, simpelweg omdat hier vakgeleerden voor aanwezig zijn om dit project te begeleiden. De code zal verder toegelicht worden in hoofdstuk vijf.

De applicatie zal ontwikkeld worden (zoals heel veel andere desktopapps zoals Spotify of Netflix) met .NET framework. .NET is een framework waar applicaties op draaien en wordt standaard geïnstalleerd bij een Windowsinstallatie. Het heeft veel functionaliteiten en ondersteunt verschillende programmeertalen. Per programmeertaal heeft .Net een compiler die de code omzet naar machinetaal zodat de hardwaredesktop de code kan uitvoeren.

De software voor de Piano Aid wordt geprogrammeerd in Visual studio. VS is een Integrated Development Environment (IDE), wat niet meer wilt zeggen dat het een visuele omgeving is die af en toe een handje helpt bij het schrijven van code. Ook wordt er gekozen om in de programmeertaal C# te programmeren, omdat het ondertussen één van de meest gebruikte programmeertalen ter wereld is.



Figuur 10: Visual Studio logo

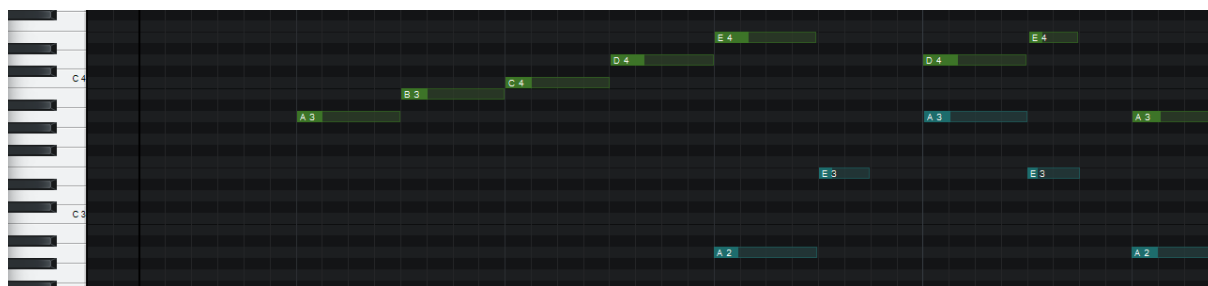
4 Applicatie

De applicatie is ontworpen om de gebruiker visueel te ondersteunen. Ze zal er ook voor zorgen dat de muziekbestanden uitgelezen en omgevormd worden zodat ze uiteindelijk tot bij de ledstrip geraken.

4.1 Midifile

Dit project kan alleen maar tot een goed einde gebracht worden door middel van de midifile. De midifile is een elektronisch muziekbestand waarin iedere noot in het muziekstuk wordt bijgehouden met een bepaalde tijd, lengte, toonhoogte en nog veel meer, wat maakt dat apparaten deze muziekstukken kunnen lezen en afspelen.

Alle midifiles die op het moment in de applicatie zitten zijn gemaakt met het programma Studio One 5. Om muziekstukken in te kunnen spelen, kan er een elektronische piano aangesloten worden op deze applicatie. De noten die dan gespeeld worden, worden dan visueel voorgesteld in een raster zoals op figuur 11. De balkjes stellen de noten voor en kunnen ook van lengte verschillen. In die noten staan ook letters. Deze letters zijn de toonhoogte of noten. Iedere noot die gespeeld kan worden krijgt een nummer beginnende bij de eerste noot op de piano namelijk C1. Deze nummers worden later uitgelezen door de applicatie.

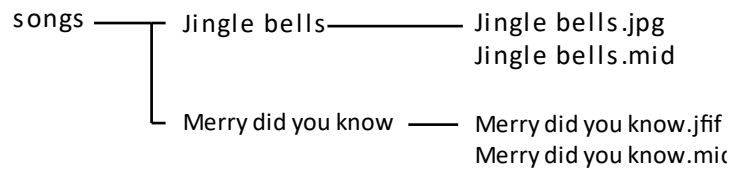


Figuur 11: visuele voorstelling van midifile

4.2 Werking

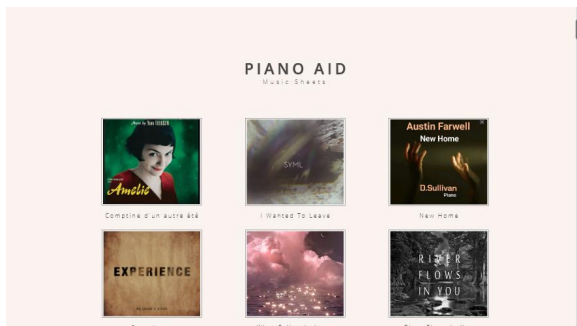
4.2.1 Songloader

Wanneer de applicatie opent, is het de bedoeling dat alle muziekstukken in de startpagina geladen worden zodat de gebruiker een liedje kan kiezen. Dit gebeurt aan de hand van de volgende mappenstructuur.

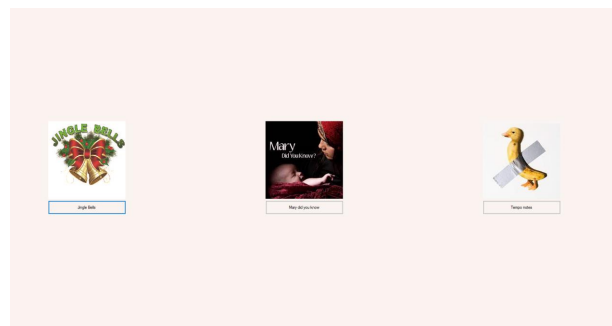


Figuur 12: voorstelling van de mappenstructuur

De SongLoader van de applicatie haalt alle songs op uit de directory en plaatst deze in een Dictionary. Deze Dictionary houdt voor ieder muziekstuk de midifile en de coverfoto bij. Op deze manier kan alle informatie meteen doorgegeven worden aan de startpagina. De startpagina of mainpage maakt voor ieder muziekstuk een nieuw panel aan waarin hij een knop en een PictureBox plaats. Deze *panels* worden dan per drie op een rij geplaatst zoals op figuur 13. Dit is hoe het scherm er zou moeten uitzien, maar op dit moment is de styling nog niet verder geraakt dan op figuur 14.



Figuur 13: design startscreen



Figuur 14: Huidige startscreen

4.2.2 PlaySong

Nu de startpagina geladen is, kan de gebruiker een muziekstuk kiezen. Op het moment dat hij of zij een liedje aanklikt, gaat de Form frmPlaysong open. De app haalt de midifile-data op waarin de pathTomidifile en de bpm-resolutie zitten. Hij haalt daarbij ook de huidige bpm op uit de midifile en de USB-poort waar de Arduino op aangesloten is, zoals te zien in de code hieronder.

```
1.     private void frmPlaySong_Load(object sender, EventArgs e)
2.     {
3.         LoadAvailableCommPorts();
4.
5.         midiData = GetMidiFileData(_pathToMidiFile, resolutionFactorToBeAbleToAugmentTimerIntervalToFixBpm);
6.         nudBPM.Value = midiData.Bpm;
7.     }
```

Vervolgens knipt de applicatie de midifile op. Dat wilt zeggen dat hij van alle noten die op dezelfde tijd ingedrukt moeten worden een commando maakt die hij vervolgens in een sequencelist zet. Visueel ziet deze lijst er zo uit, maar natuurlijk veel langer.

SequenceList
010405#>
02132425#>
0203#>

Zoals eerder vermeld bij de midifiles staat ieder nummer voor een noot. Zo bestaat het eerste commando in de tabel uit led één, vier en vijf.

Als de gebruiker op start klikt, begint een nieuwe timer te lopen. De applicatie stuurt dan aan de hand van deze timer en de tempomap van de midifile de *sequencelist* door naar de Arduino.

De gebruiker kan niet alleen maar op start drukken, maar hij of zij kan ook het muziekstuk pauzeren, herstarten of de snelheid ervan veranderen.

5 Ledsturing

De ledsturing is opgebouwd uit twee onderdelen, namelijk de beveiliging en de omvorming van de commando's die zij zal doorkrijgen van de applicatie.

Om de Arduino te kunnen monitoren wordt er gebruikgemaakt van een stukje code dat de Arduino een "hartslag" geeft. Dit stukje code laat een ledje op het moederbord knipperen, ongeacht of de Arduino berekeningen doet of niet. Deze hartslag zal een indicatie zijn of de Arduino nog reageert of niet.

Ook zit er een *whilebreaker* in het programma ingebouwd. Dit stukje code zorgt ervoor dat als de Arduino vast blijft hangen in een *loop*, deze toch onderbroken kan worden. Hiervoor werd er wel een extra kabeltje toegevoegd aan de schakeling. Dit is gewoon een verbinding tussen een inputpin en de 5 Voltpin. Als deze connectie onderbroken wordt, breaket de Arduino zijn programma meteen.

Wanneer de Arduino een commando ontvangt van de app, weet hij niet hoeveel ledjes hij door zal krijgen. Vandaar de keuze om ieder commando af te sluiten met "#>". Zo weet de Arduino welke leds hij tegelijkertijd moet laten branden. De Arduino haalt dit teken er na ieder commando af en verwijdert de overbodige nullen zoals bij ledje 02. Vooraleer hij het ontvangen commando doorstuurt naar de ledstrip, zet de Arduino alle leds ook heel kort uit, maar dit gebeurt zodanig snel dat het oog dit niet waarneemt.

De volgende code is de code die de commando's omvormt en doorstuurt naar de ledstrip.

```
1. void loop() {
2.   HB.beat();
3.
4.   if(inputStringComplete){
5.     inputStringComplete = false;
6.     printToSerial(inputString);
7.     String command = inputString;
8.     command.replace("#>", "");
9.     inputString = "";
10.    printToSerial(command);
11.    clearAllLeds();
12.
13.    int maxLeds = 100;
14.    int arrLedIds[maxLeds];
15.    int numberOfSetLeds = getLedArray(command, maxLeds, arrLedIds);
16.
17.    //printToSerial(String(numberOfSetLeds));
18.    for (int i = 0; i < numberOfSetLeds ;i++){
19.      leds[arrLedIds[i]] = CRGB(10, 10, 0);
20.    }
21.    FastLED.show();
22.  }
23. }
```

Op deze manier worden alle commando's na elkaar doorgestuurd zodat het liedje zichtbaar wordt op de toetsen van de piano.

Besluit

De ledstrip werd gemonteerd op het frame en aangesloten op een extra voeding van vijf Volt. De applicatie verwerkt de midifile data, laadt alle muziekstukken in het startscherm en stuurt de data succesvol door naar de Arduino die ze op zijn beurt verwerkt en doorstuurt naar de ledstrip. Er zitten twee verschillende muziekstukken in de applicatie, maar het design heeft nog niet zijn volle potentieel bereikt.

Stel dat dit product marktklaar gemaakt zou moeten worden, dan is het van belang dat alle bekabeling weggewerkt wordt in het frame. Het frame moet daarom uit sterker materiaal vervaardigd worden. Het design van de applicatie moet geoptimaliseerd worden en kan later op een tablet of smartphone verschijnen. Om de applicatie gebruiksvriendelijker te maken zou het zeker een meerwaarde zijn als er notenbalken op het scherm verschijnen tijdens het afspelen van een muziekstuk. Op deze manier kan de gebruiker ook noten leren lezen tijdens het pianospelen.

Als ik dit project opnieuw zou moeten doen, zou ik sneller beginnen met onderzoeken en vooral eerder beginnen met het schrijven van deze paper vanwege de hoeveelheid werk. Toch heb ik ook geleerd dat met de juiste motivatie en passie geen project te groot is. Alles gaat met ups en downs en soms moet je er gewoonweg doorheen!

Afkortingen en vakjargon

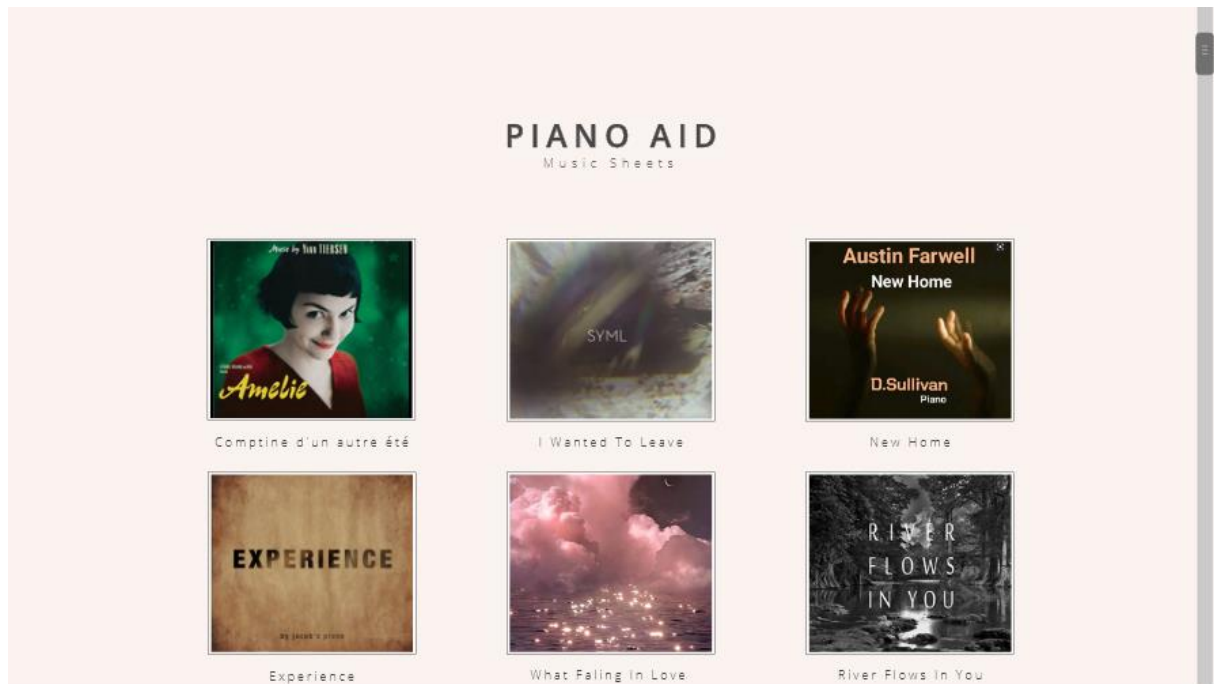
Bootloader	Een bootloader is een computerprogramma dat zorgt voor het starten van het besturingssysteem bij de opstart (bootstrap) van een computer.
C#	Fonetisch: [Sie charp], is ondertussen een van de meest gebruikte programmeertalen ter wereld.
Hardware	Hardware is een verzamelnaam voor alle fysieke onderdelen die samenhangen met het computersysteem.
Seriële interface	Een seriële interface verzend informatie tussen twee digitale systemen in de vorm van spanningspulsen (vertaald in 0 en 1).
Microcontroller	Een typische microcontroller bevat een processor, geheugen en input/output (I/O) randapparatuur op een enkele chip
Midifile	Elektronisch muziekbestand
Noten	De noten die beschreven worden in deze paper zullen benoemd worden met letters terwijl het algemeen bekend is dat toonladders bestaan uit do, re, mi, fa, sol, la, si, do. De letters staan in volgende volgorde: C, D, E, F, G, A, B, C waarvan de C komt overeen met de 'do' enzovoort.
Software	Software is alles wat op de computer wordt geïnstalleerd en vervolgens op het beeldscherm wordt getoond en gebruikt.

Verwijzingen

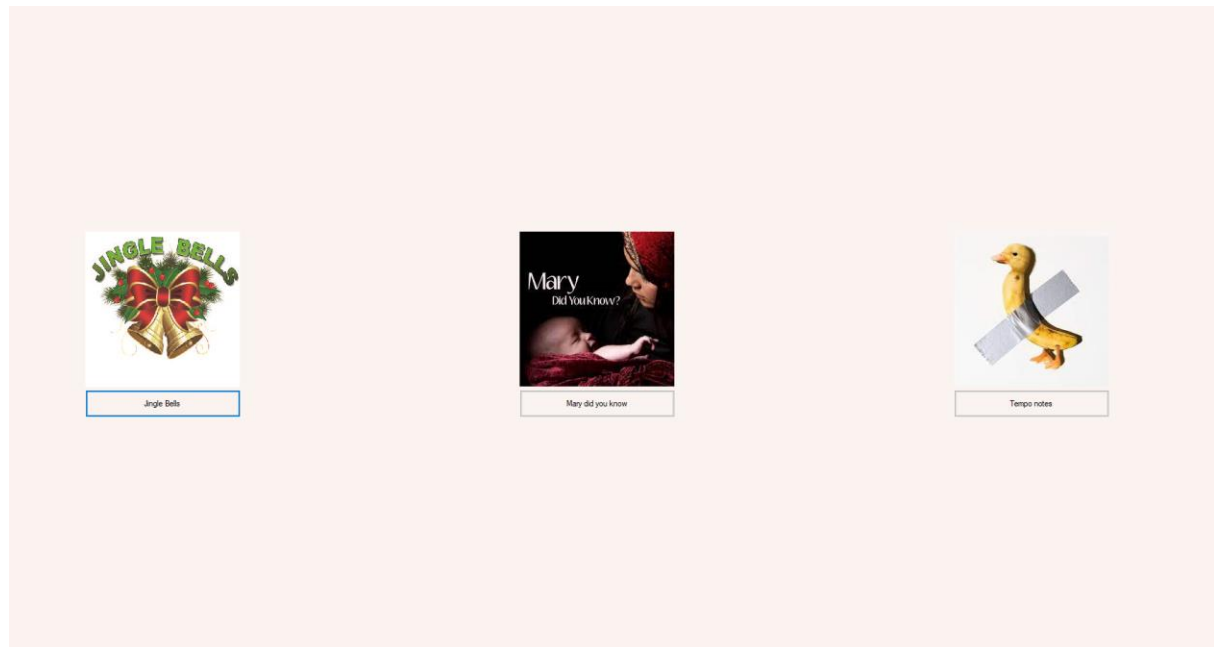
- Banzi, M. (2021). *Arduino.cc*. Opgehaald van Arduino: <https://www.arduino.cc/>
- Barela, A. (2021, 12 01). *Serial Communications Example*. Opgehaald van Adafruit: <https://learn.adafruit.com/using-webusb-with-arduino-and-tinyusb/serial-communications-example>
- Beaufort, F. (2021, 02 23). *Access USB Devices on the Web*. Opgehaald van Web.dev: <https://web.dev/usb/>
- educ8s.tv. (2017, 04 22). *Arduino Tutorial: C# to Arduino Communication*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=vHeG3Gt6STE&list=PLnURnGkHEZiWn4vFJ1PxCLL94M8xQH7PZ&index=8>
- Majenko. (2020, September 28). *How to determine what resistor value to use for LED strip*. Opgehaald van StackExchange: <https://arduino.stackexchange.com/questions/78232/how-to-determine-what-resistor-value-to-use-for-led-strip#:~:text=It's%20only%20the%20length%20of,impedance%20of%20the%20Arduino's%20output.>
- Nedelkovski, D. (2018, jan 13). *How To Control WS2812B Individually Addressable LEDs using Arduino*. Opgehaald van How To Mechatronics: <https://www.youtube.com/watch?v=UhYu0k2woRM&list=PLnURnGkHEZiWn4vFJ1PxCLL94M8xQH7PZ&index=5&t=246s>
- Pert. (2020, mei). *Arduino Leonardo - errors uploading programs after bad upload*. Opgehaald van Arduino.cc: <https://forum.arduino.cc/t/arduino-leonardo-errors-uploading-programs-after-bad-upload/656859/5>
- RobTillaart. (2019, juni 12). *HeartBeat*. Opgehaald van github.com: <https://github.com/RobTillaart/HeartBeat>
- TheONEMusicSchool. (2017, 09 18). *Introducing The ONE Piano Hi-Lite*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=g0M2vI9uqug&list=PLnURnGkHEZiWn4vFJ1PxCLL94M8xQH7PZ&index=4>

Bijlagen 1

Startpagina website

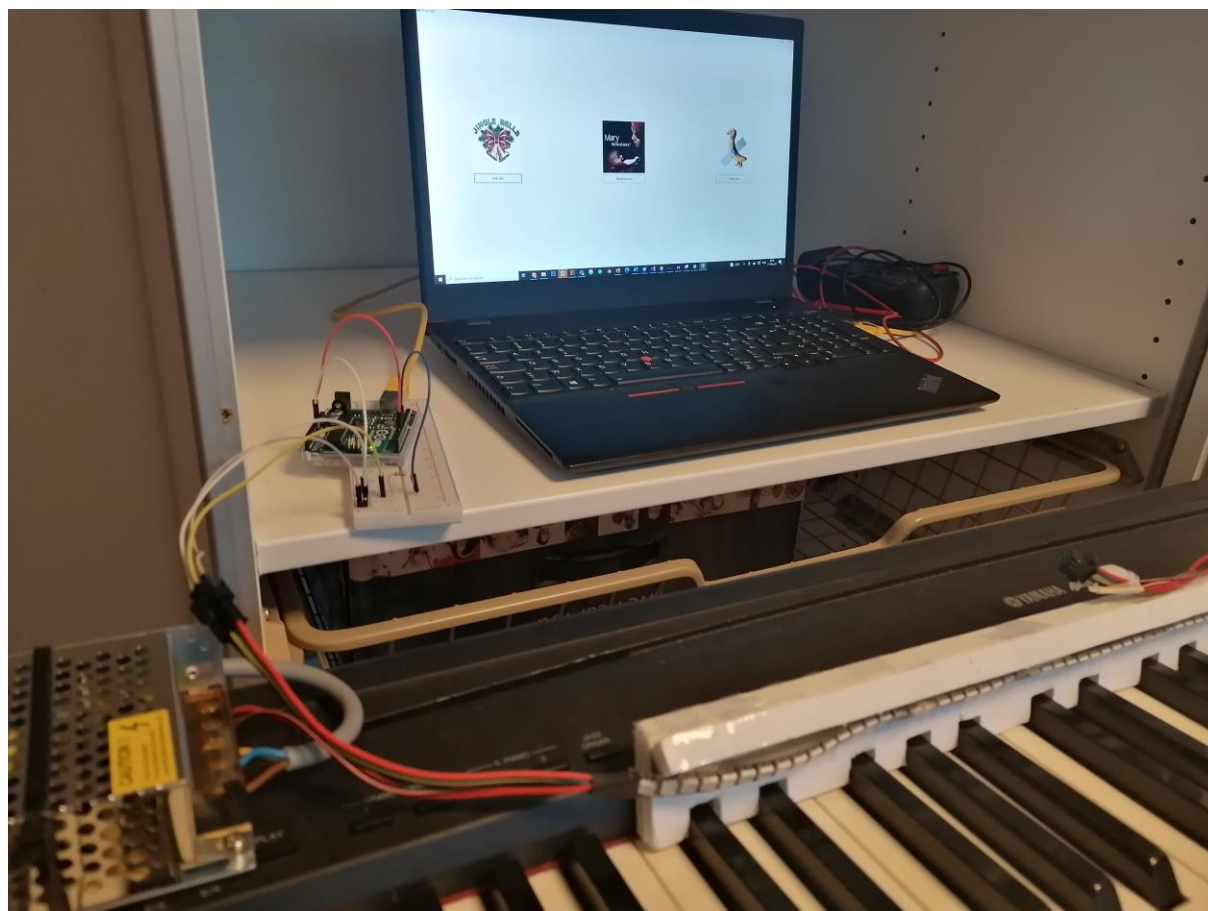


Huidige startpagina



Bijlagen 2

Piano Aid opstelling



Bijlagen 3

Volledige code ledsturing

```
24. #include "HeartBeat.h"
25. #include <FastLED.h>
26.
27. #define LED_PIN 7
28. // #define CLOCK_PIN 8
29. #define NUM_LEDS 140
30. #define WHILE_BREAKER_INPUT_PIN 4
31.
32. CRGB leds[NUM_LEDS];
33.
34. HeartBeat HB;
35.
36. void setup() {
37.     InitializeHeartBeat();
38.     InitializeUsbCommunication();
39.
40.     pinMode(WHILE_BREAKER_INPUT_PIN, INPUT);
41.
42.     //FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS); // <= voor WS2812B
43.     FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS); // <= ook voor WS2813
44.
45.     clearAllLeds();
46. }
47.
48. void InitializeHeartBeat() {
49.
50.     // Serial.begin(115200);
51.     // printToSerial(__FILE__);
52.     printToSerial(HEARTBEAT_LIB_VERSION);
53.
54.     pinMode(LED_BUILTIN, OUTPUT);
55.     HB.begin(LED_BUILTIN, 0.5); // frequency 0.5
56. }
57.
58. void InitializeUsbCommunication(){
59.     Serial.begin(9600); //initialize serial comm
60. }
61.
62. void HeartBeatFast(){
63.     HB.begin(LED_BUILTIN, 8);
64. }
65.
66. void HeartBeatSlow(){
67.     HB.begin(LED_BUILTIN, 2);
68. }
69. void HeartBeatIdle(){
70.     HB.begin(LED_BUILTIN, 0.5);
71. }
72.
73.
74. void printToSerial(String string){
75.     Serial.print(string);
76. }
77.
78. String inputString = "";
79. bool inputStringComplete = false;
80. String previousInChar = "";
81. void loop() {
82.     HB.beat();
83.
84.     if(inputStringComplete){
```

```

85.   inputStringComplete = false;
86.   printToSerial(inputString);
87.   String command = inputString;
88.   command.replace("#>", "");
89.   inputString = "";
90.   printToSerial(command);
91.   clearAllLeds();
92.
93.   int maxLeds = 100;
94.   int arrLedIds[maxLeds];
95.   int numberOfSetLeds = getLedArray(command, maxLeds, arrLedIds);
96.
97.   //printToSerial(String(numberOfSetLeds));
98.   for (int i = 0; i < numberOfSetLeds ;i++){
99.     leds[arrLedIds[i]] = CRGB(10, 10, 0);
100.  }
101.  FastLED.show();
102.  }
103. }
104.
105. int getLedArray(String command, int maxLeds, int arrLedIds[]){ //returns number of set Leds
106.
107.   int ledCounter = 0;
108.   String strLedId = "";
109.   //printToSerial(String(command.length()));
110.   for(int i = 0; i < command.length(); i++ ) {
111.     char c = command[i];
112.     if(i%2 == 0){
113.       strLedId = String(c);
114.       //printToSerial(strLedId);
115.     }
116.     else {
117.       strLedId += String(c);
118.       //printToSerial(strLedId);
119.       arrLedIds[ledCounter] = strLedId.toInt();
120.       //printToSerial(String(arrLedIds[ledCounter]));
121.       ledCounter++;
122.       if(ledCounter > maxLeds-1)
123.         break;
124.     }
125.   }
126.   return ledCounter;
127. }
128.
129. void serialEvent() {
130.   while (Serial.available()) {
131.     //HeartBeatFast();
132.     bool val = digitalRead(WHILE_BREAKER_INPUT_PIN);
133.     if(val == false)
134.       break;
135.
136.     // get the new byte:
137.     char inChar = (char)Serial.read();
138.     //Serial.println(inChar);
139.     inputString += inChar;
140.     // if the last incoming characters are #> set a flag
141.     // so the main loop can do something with it:
142.     if (previousInChar == "#" && inChar == '>') {
143.       inputStringComplete = true;
144.       //Serial.println("inputString Complete");
145.     }
146.     previousInChar = inChar;
147.     //delay(1);
148.   }
149.   //HeartBeatIdle();
150. }
151.
152. void clearAllLeds(){
153.

```



```
154. for(int i = 0;i<NUM_LEDS;i++){
155.     leds[i] = CRGB(0, 0, 0);
156. }
157. FastLED.show();
158. printToSerial("All Leds cleared");
159. }
160.
```