Thomas More

Campus De Nayer

Jan De Nayerlaan 5

2860 Sint-Katelijne-Waver

Tel. (015) 31 69 44

*Practice Enterprise Electronics*

*Schooljaar 2020-2021*

**NAAM: Hannes Hugaert & Stijn Van den Bossche**

#### **ONDERWERP: Muziekversterker**

**AFDELING: Elektronica – Embedded Hardware 2 | 2EE2**

**Elektronica – Embedded Software 2 |**

[1. Woord vooraf 3](#_Toc71034737)

[2. Hardware 3](#_Toc71034738)

[**1.1.** **Blokschema** 3](#_Toc71034739)

[**1.2.** **Schema** 6](#_Toc71034740)

[1.2.1. Communicaties 8](#_Toc71034744)

[1.2.2. Spanningsregelaars 10](#_Toc71034749)

[1.2.3. Digitale potentiometer(AD5204BRUZ10-REEL7) 11](#_Toc71034750)

[1.2.4. 8 digit LED display driver (MAX7219CNG+) 12](#_Toc71034751)

[1.2.5. Versterking 15](#_Toc71034759)

[**1.3.** **PSpise** 15](#_Toc71034760)

[**1.4.** **Bestukkingsplan** 15](#_Toc71034761)

[**1.5.** **PCB lay-out** 15](#_Toc71034762)

[**1.6.** **Datasheets** 15](#_Toc71034763)

[3. Software 15](#_Toc71034764)

[**1.1.** **Flowchart software** 15](#_Toc71034765)

[**1.2.** **Listing (de code)** 15](#_Toc71034766)

[4. Bediening handleiding 16](#_Toc71034767)

[5. Besluit 16](#_Toc71034768)

[6. Opgezochte sites 16](#_Toc71034769)

# **Woord vooraf**

Wij gaan een muziekversterker maken die voorzien is van 2 channels en een microfoon. Op deze versterker kunnen we de tonen regelen (trable, medium en bass) maar we kunnen ook het volume regelen. Onze sliders zijn niet de gewone die we tegenkomen op een gewone versterker maar wij gaan onze voorzien van touchpad’s als sliders. Om ook de weergave van het volume of de tonnen te laten zien voorzien we ledjes naast de touchpad’s om die verandering te kunnen waarnemen.

# **Hardware**

## **Blokschema**

**Audio output:** staat voor de boxen die aan de uitgang komen, deze boxen moet voorzien worden van een inwendige versterking omdat onze versterking amper stroom zal hebben aan de uitgang.

**2 AUDIO inputs:** Wij hebben ons mengpaneel voorzien van 2 lines waar wij muziek op kunnen laten spelen. + Wij kunnen op line 1 eventueel ook een microfoon steken. Zodat we kunnen zingen op muziek en terwijl het liedje horen.

**5V DC:** micro USB, dit is onze input voor onze versterker te voeden.

**5V DC naar 3.3 V DC:** dit is de naam van de component 173950336, het is een step-down voorzien die onze 5V omzet naar +3.3 V DC. Dit dient voor onze Microcontroller en onze capacitieve tastsensor te voeden.

**Microcontroller:** onze Microcontroller is eenSTM32F411CEU7, we kunnen communiceren via een st-link V2 met een computer, zodat we er een programma kunnen naar sturen. Want onze CPU kan geen rechtstreekse signalen uitlezen van de computer daar is er geen poort voor voorzien. Onze Microcontroller is van de SMT32F4x familie en die wordt gevoed met 3.3V en we willen ook ongeveer 40 I/O. En onze heeft er 49 dus dat is zeer goed.

**8-digit LED displaydrivers:** deze chip heet MAX7219CNG+. Dit is een chip voor 8 segment displays zou aansturen maar wij hebben een matrix gemaakt met onze led’s waardoor dit ook met deze chip lukt om het aan te sturen.

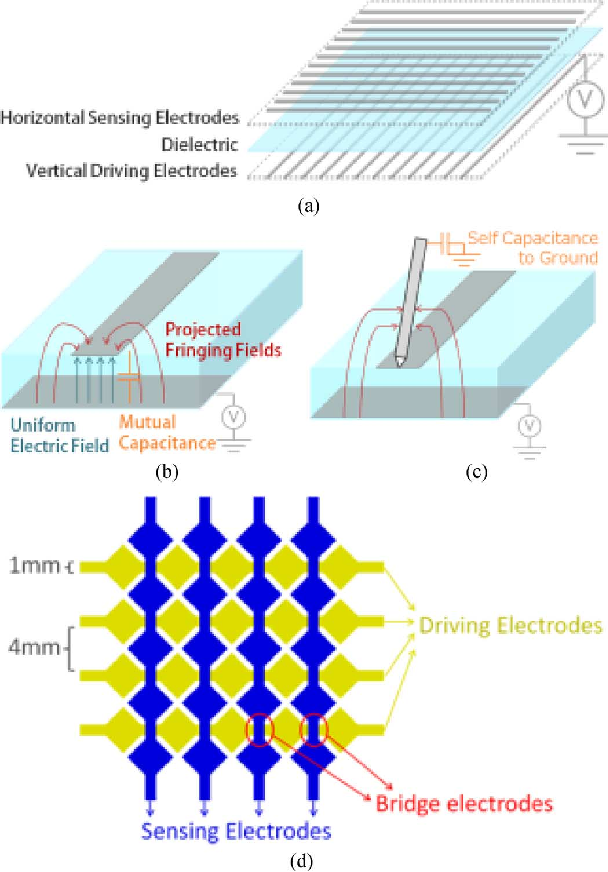
Dit hoort samen

**LED’s:** deze led’s tonen aan hoe luid en hoe hoog of laag onze tonen zijn geregeld. Wij hebben 132 LED’s voorzien zodat echt al de standen heel duidelijk zijn.

**Capacitieve tastsensor:** deze sensor heet de MTCH6301-I\_ML, Het leest capacitieve signalen binnen en die zet dat om naar een I²C signaal zodat onze Microcontroller ons aanrakingen verstaat. We hebben ook gekozen voor capacitief omdat dat geen corrosie geeft op de sliders.

Dit hoort samen

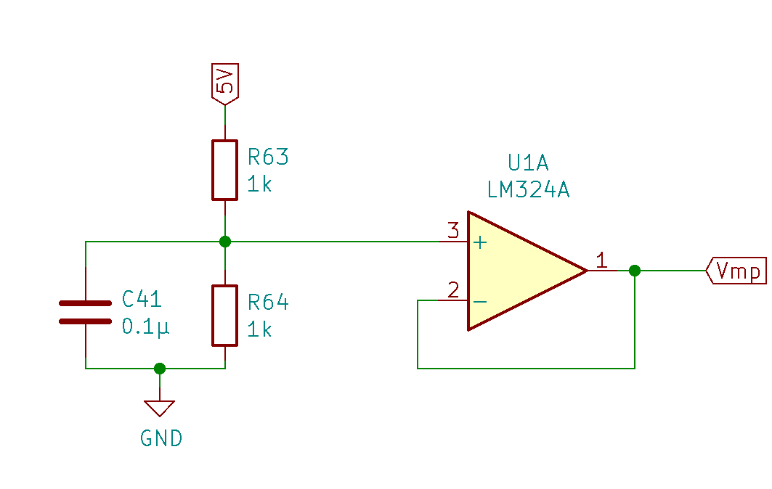
**Touchpads:** dit zijn geleiders die op een print door middel van aanraking een capacitief signaal doorsturen. Deze touchpad’s worden onze sliders van ons paneel. Deze gaan wij zelf maken zo besparen we 3 euro per slider uit en kunnen we onze vorm en grote zelf bepalen.

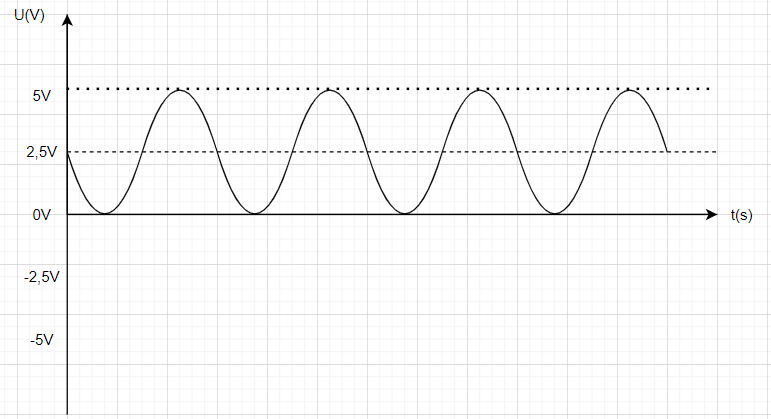


**digitale potentiometers:** onze potentiometers noemen AD5204BRUZ10-REEL7. Wij hebben gezocht naar een oplossing om onze potentiometers te kunnen sturen. Zo kwamen

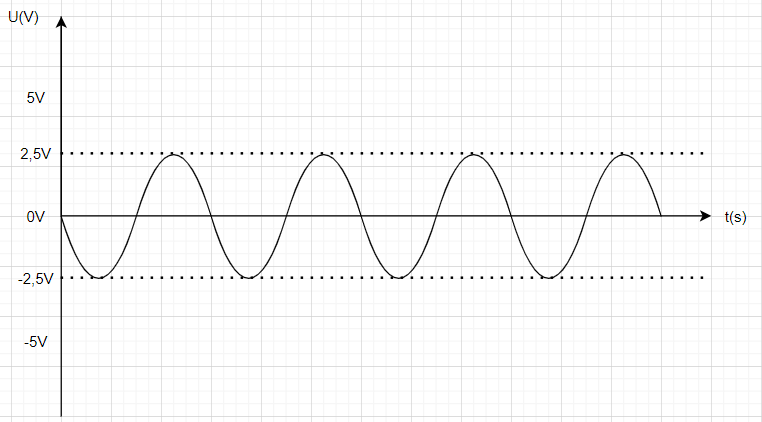
we uit op digitale potentiometers potentiometers.

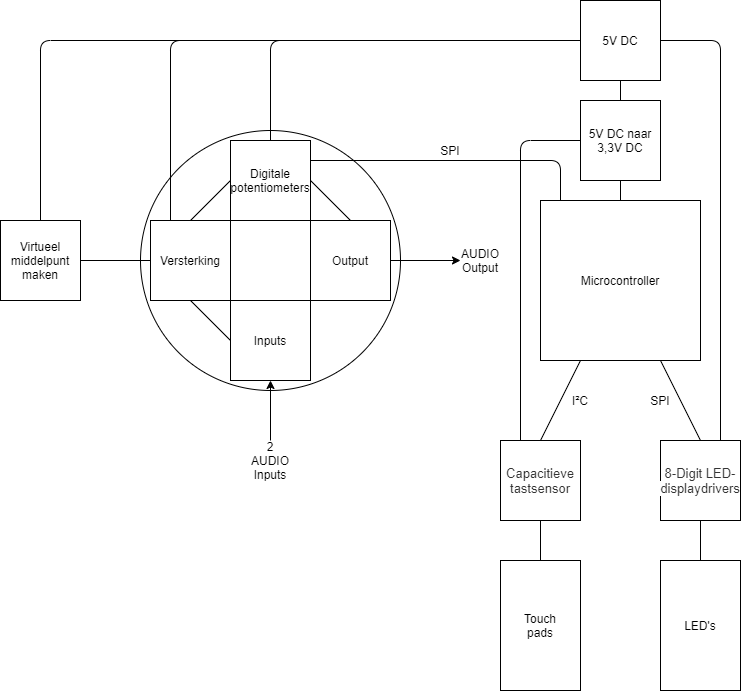
**Versterken:** wij gebruiken de opamps LM324A.We gaan onze muziek versterken met opamps en ook nog met de nodige andere componenten.

**Virtueel middelpunt:** omdat we problemen hadden met de potentiometers en zo in de schakeling hebben we een virtueel middelpunt moeten maken dit is aan de hand van een spanningsdeler een opamp.

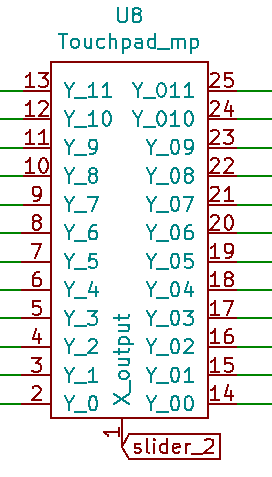


5V/2 =2,5V is dus gewoon een spanningsdeller gedeeld door 2 waardoor we gewoon twee gelijke weerstanden moeten nemen. De docent stelde ons voor om 1kΩ weerstanden te gebruiken.

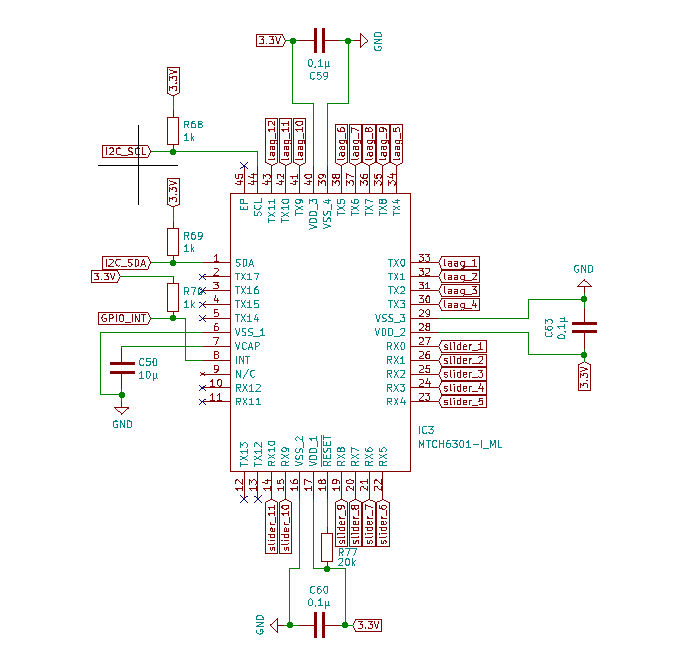
We willen dat het nulpunt op 2,5 V ligt. Zodat we de opamps kunnen fluctueren tussen 0V en 5V. Zo hebben we zekerheden met het signaal en de potentiometers en dan kunnen we gewoon de 5V van de usb gebruiken wat heel handig. Anders moesten we de spanning verlagen van 5V naar 2,5V en ook naar -2,5V en dat gaat ons veel meer kosten dan dat we het middelpunt gaan verleggen. We blijven ook het zelfde uitgangssignaal horen.



## **Schema**



**Touchpad:** Dit is het symbool van onze sliders, deze hebbe zelf ontworpen om dat we deze wouden verwerken in onze PCB, hier hebbe we heel veel moeten voor opzoeken want zoe een raster van een slider moet aan een bepaalde afmetingen voldoen zodat het zeker voldoende afstand heeft van el vlakje en ook een deftige connectie kunt maken als je met je vinger op de touchpad komt.

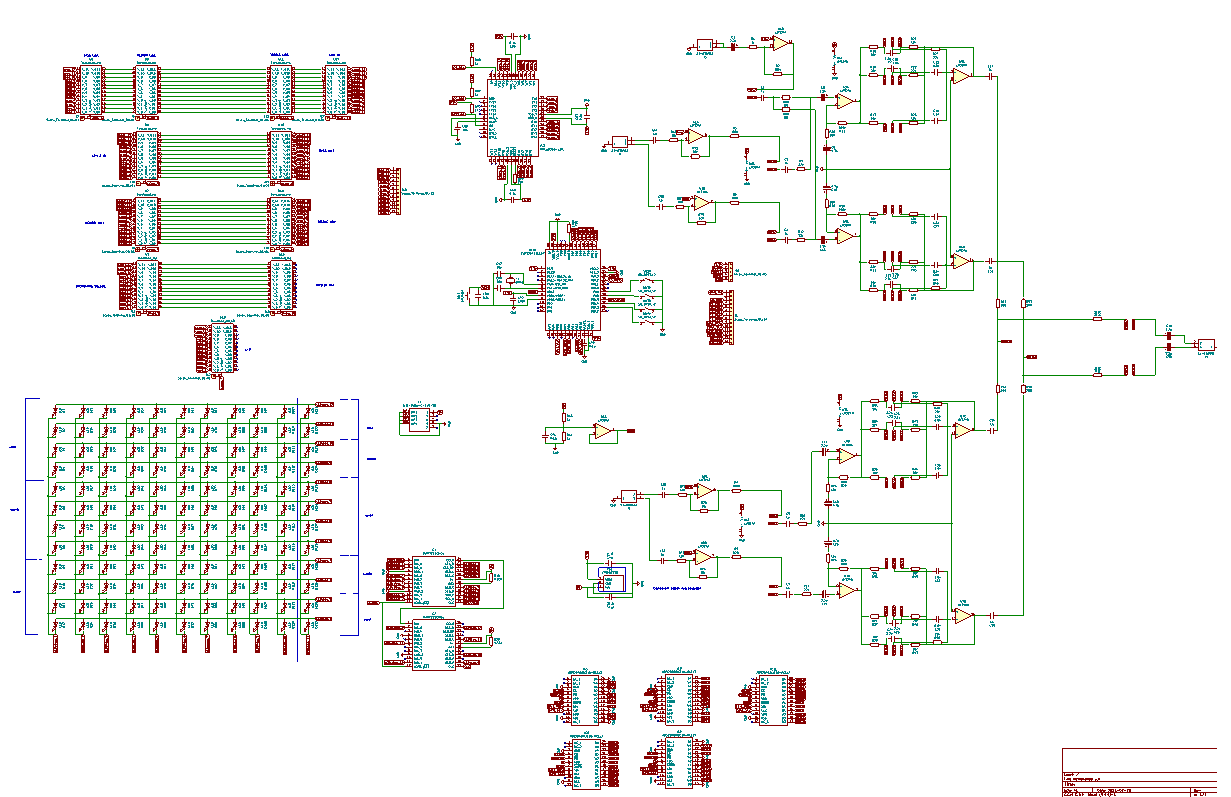


**MTCH6301-I\_ML:** dit is de IC die de signalen op de touchpad’s uitleest en om zet in data die over I²C naar de microcontroller gaat.

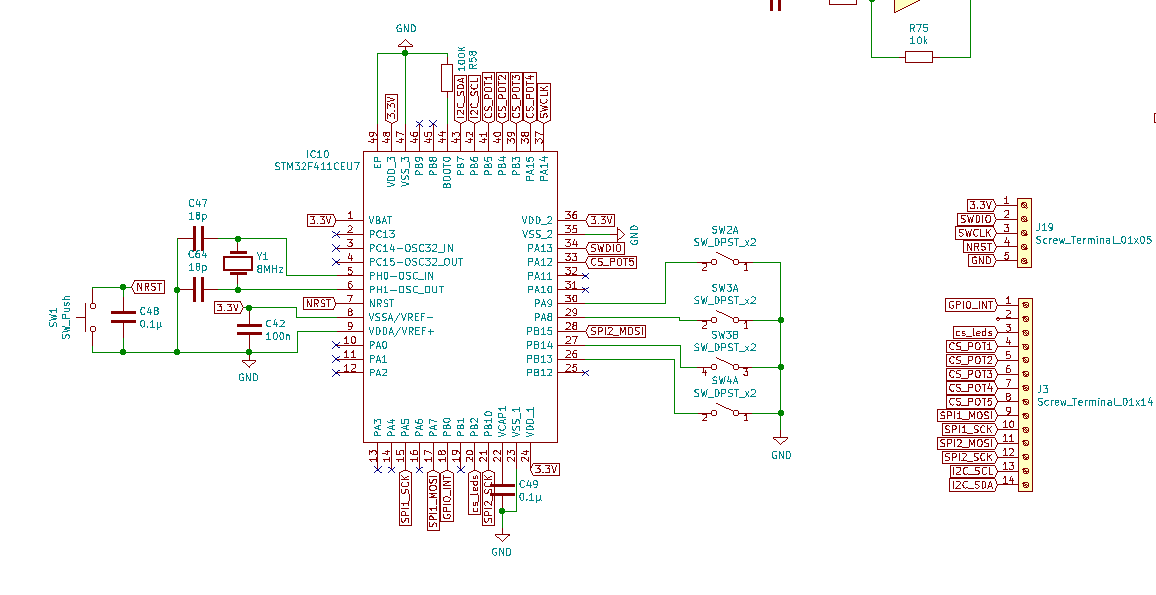
Deze condensatoren zorgen er voor dat er geen storing op de 3.3V zit, zo blijf je een ongeveer een mooie rechte 3.3V hebben.

Als de interrupt van de microcontroller spanning verliest of de spanning is hoger dan 3.3V zorgt de pull up weerstand dat de nodige 3.3V aan de IC komt.

De RESET moet laag zijn om een zichzelf te resetten maar wij willen niet dat dat we deze chip resetten dus daarom zetten we ze aan de 3.3V zodat de chip zijn data niet verliest door de reset.



Micro usb



**STM32F411CEU7:** Deze microcontroller is voorzien van 512KB Flash en 128 KB RAM en is een ARM cortex die op 32 bit werkt. Deze microcontroller is voorzien van 49 pinnen wat ruim genoeg is voor onze uitwerking. Deze microcontroller moet gevoed worden op een spanning tussen 1.7 en 3.6 dus hebben we gekozen voor het midden ongeveer 3.3V.

We hebben ons bord voorzien van externe pinnen stel dat er iet met onze microcontroller gebeurd of we willen de signalen mooi en gemakkelijk willen lezen kunnen we deze uitlezen op de ze pinnen.

Wij hebben een reset knop voorzien op de microcontroller zodat we altijd op nieuw het programma kunnen laten afspelen, stel dat er een is kunnen we dit direct uitlezen door de pinnen die hierboven vermeld worden. De reset moet voor zien worden van condensator zodat we geen bounce krijgen zie oscillatie.

De microcontroller moet een kristal oscillatie hebben tussen 4 a 26 MHz. Wij hebben de onze voorzien met een kristal van 8 MHz, waardoor deze perfect tussen 4 a 26 MHz oscillatie past die onze microcontroller nodig heeft.

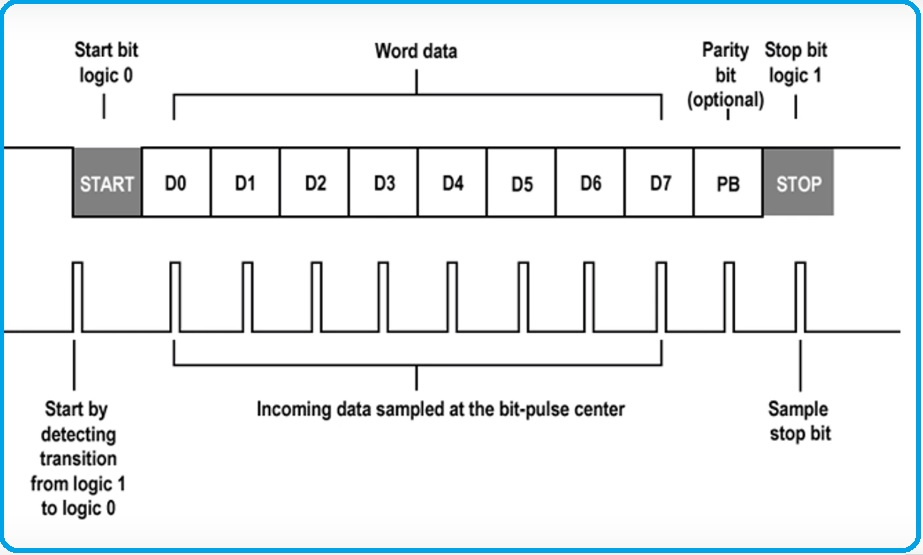
Op de microcontroller zijn er twee poorten voorzien voor een oscillatie en wij hebben gekozen voor PH0 en PH1 omdat de andere een oscillatie maar van 32 kHz max mag voorzien en dat met het gebruik van een RTC, deze bevind zich op pinnen PC14 en PC15. Door dat we een de kristal gebruiken krijgen we een interne klok van 16 MHZ en als we de RTC gingen gebruiken was het een interne klok van 32 kHz.



## Communicaties

**UART**

UART hebben we voorzien omdat we ons programma moeten inlezen in de microcontroller. Aan de pinnen van de UART voorzien we een “USB to ttl” zodat we seriële communicatie van de usb op die PCB naar UART kunnen omzetten. Dit kan de microcontroller perfect uitlezen.



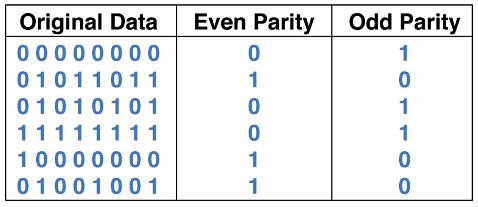
Binnen komende data

CLK die hoog word

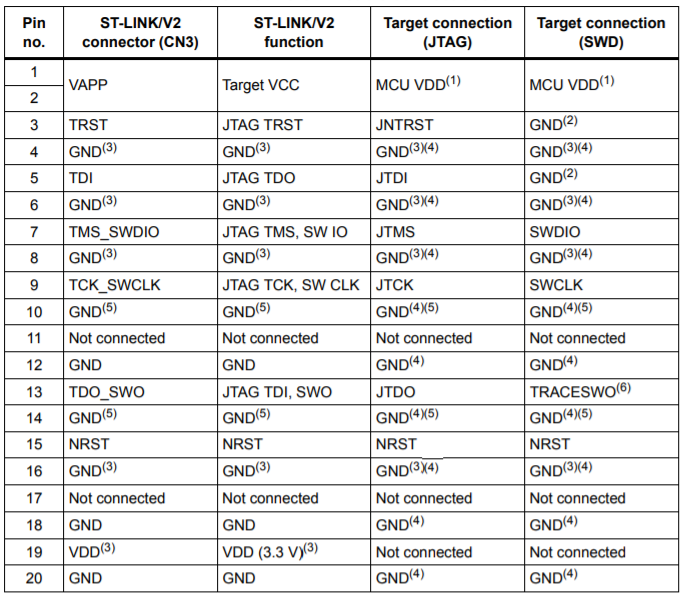
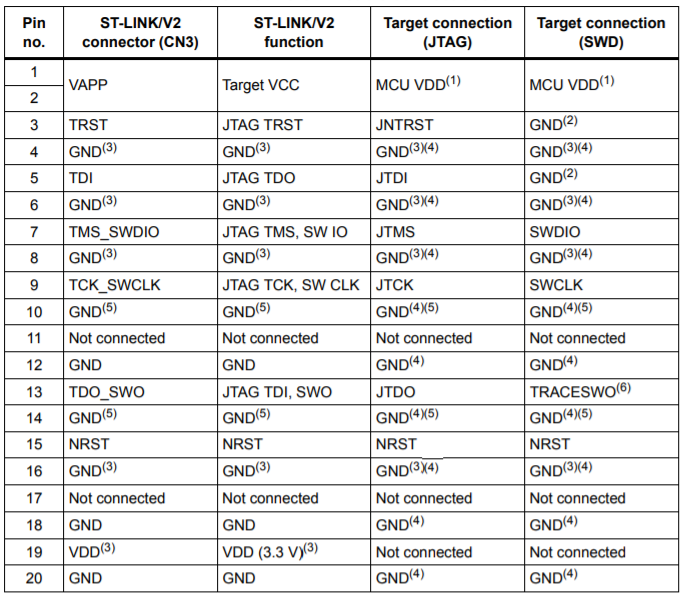
als je data moet

uitlezen

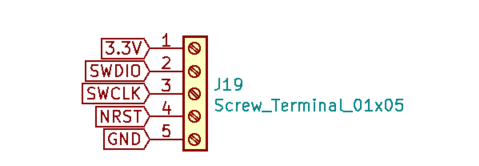
Tijd voor uitlezing: 1.5 bit, 1 bit, …..(de rest is ook 1 bit)

De data van UART word altijd maar uitgelezen als de binnenkomende data eerst laag wordt dit is de start bit en dan heb je de 8 bit of 1byte die zowel hoog of laag kunnen zij maar natuurlijk niet tegelijk natuurlijk. Na de 8 bit komt de parity bit deze is er niet altijd maar het detecteert altijd 1 fout en zegt of het oneven is of even aan de hand van een 1(even) of 0(oneven) te zetten in deze bit. Zoals deze teken hiernaast. De laatste bit moet terug hoog worden bij de UART dit wordt de stop bit genoemd en dan stopt de TX met uitlezen.

Doordat we de STM32F411CEU7 kunnen we niet programmeren met de “USB to ttl” dus dan programmeren we niet met het protocol Uart maar wel met seriële communicatie. De datasheet van de st link V2 verwijst er naar om de pinnen zo aan te sluiten zoals op de op de tabel en tekening hieronder.



2. Maak verbinding met GND voor ruisonderdrukking op het lint.

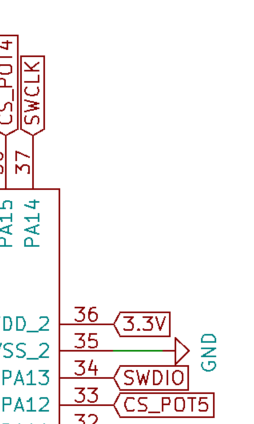


3. Alleen beschikbaar op ST-LINK / V2, niet aangesloten op ST-LINK / V2-ISOL.

4. Ten minste één van deze pennen moet met de grond worden verbonden voor correct gedrag (ze allemaal verbinden is

aanbevolen).

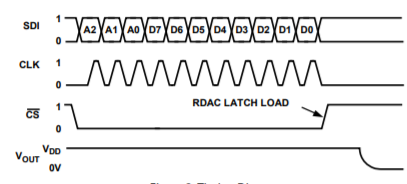
5. GND op ST-LINK / V2, gebruikt door SWIM op ST-LINK / V2-ISOL

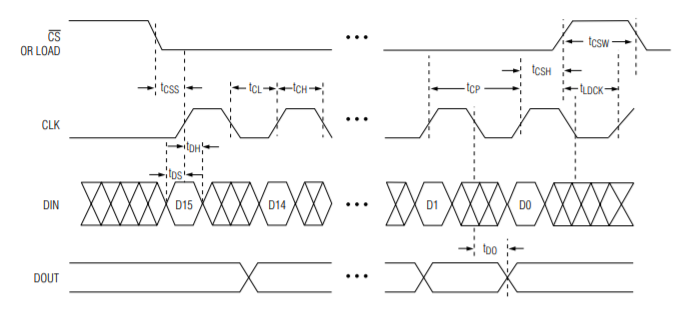


De pin SWCLK van de ST link V2 moeten op pin PA14 van de STM32F411CEU7 komen. De pin SWDIO van de ST link V2 moeten op pin PA13 van de STM32F411CEU7 komen. Stel deze in op de microcontroller en dan kan je beginnen met programmeren van de microcontroller op ons mengpaneel.

**SPI**

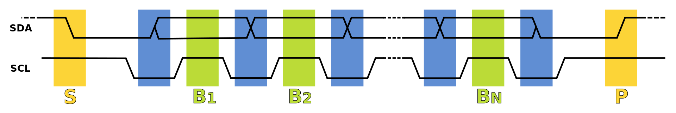
Als de Chip Select (CS) een laag signaal heeft en die doorstuurt naar een bepaalde slave, want wij gebruiken er meerdere. Zo kan de data alleen naar de bepaalde slave’s verzonden worden die een laag CS binnen krijgen.

* **Digitale potentiometer:** Als je CS laag gemaakt hebt kan je de 16 bit data beginnen inlezen. De digitale potentiometers beginnen de adressen en de data in te lezen als het CLK hoog is. Als alle adressen en data is ingelezen moet CS teru goog worden en zal de weerstands waarde van de poteniometer veranderen.
* **8-Digit LED Display Driver:** Hier gebeurt exact het zelfde als bij de digitale potentiometer, behalve dat als de data wordt ingelezen en iets nadat de CLK laag is verander de output.



**I²C**

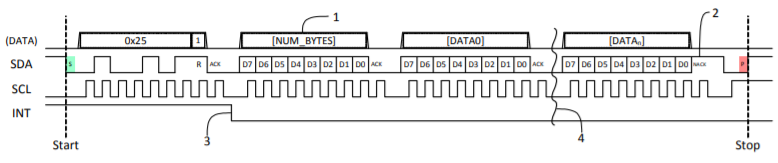
De IC voor onze touchpad’s worden aangestuurd met I²C. Dit protocol wordt over 2 lijnen gestuurd de lijn SDA (data line) en SCL (clock line). De lijn data is geen data die binnenkomt op de chip maar toont aan wat voor data er binnenkomt.



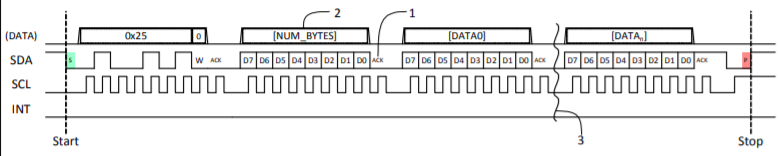
**Start:** De chip begint te starten met lezen van data als de SDA lag wordt en SCL hoog is dit is dan de start bit, daarna moet SCL eens laag worden en dan kan het data beginnen ingelezen worden. Per keer dat SCL hoog wordt de data op SDA uitgelezen.

**Read:** De eerste data dat binnen gelezen wordt is het adres + een read bit zodat de master zegt dat hij de data wilt uitlezen. Als dit gebeurd is zal de INT laag worden zodat het kan beginnen met de data uit te lezen, als INT hoog stond wil dat zeggen dat het adres niet machte.

Master read



**Write:** De eerste data dat binnen gelezen wordt is het adres + een write bit zodat de master zegt dat hij de data wilt uitlezen. De INT blijft hier laag zodat het altijd kan dat verzenden.



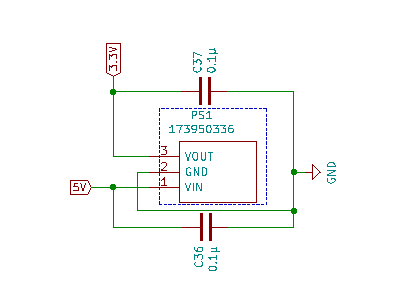
**Stop:** De stop bit wordt aangeduid als de SCL al eventjes hoog is en daarna SDA hoog wordt.



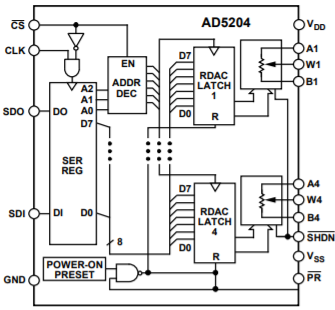
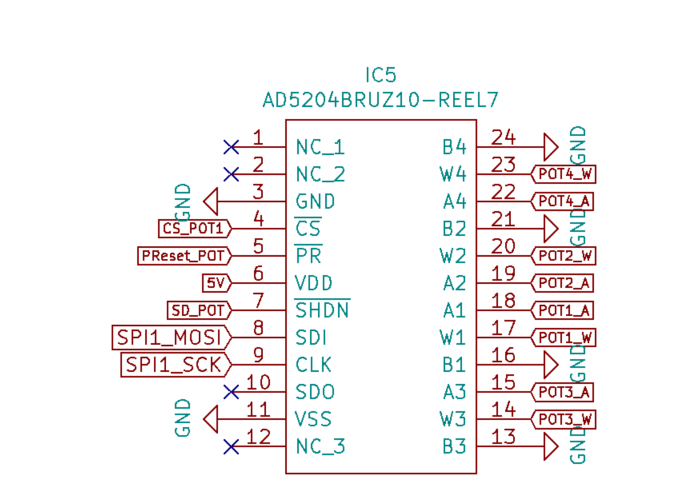
## Spanningsregelaars

**173950336:** is een Step Down die 5V DC omzet naar 3,3V DC en heeft een uitgangsstroom van 500 mA.

De condensatoren dienen voor spanningen af te vlakken zo zitten we zonder ruis op de beide spanningen.



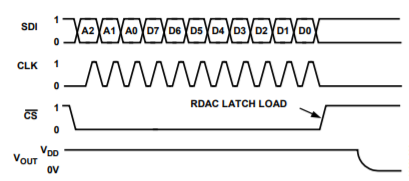
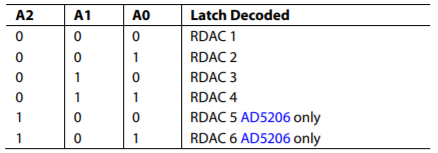
## Digitale potentiometer(AD5204BRUZ10-REEL7)



**CS:** Zorgt er voor dat dat de waardes maar gelezen wordt als deze uitgelezen worden als deze laag is.

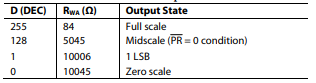
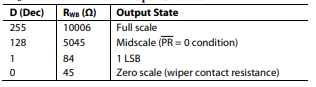
**CLK:** Als de Clock hoog is zal de data doorgegeven worden in het shiftregister. Door de AND poort in de chip zorgt de CS en de clock er voor de data wordt door gegeven en uitgelezen kan worden.

**SDI:** Hier worden alle bits ingelezen die je doorstuurt op deze pin. Het shiftregister leest een serieel signaal in de eerste 3 bits zijn de adres bits de rest zijn de 8 data bits (of 1 byte). Als alle data is uitgelezen en CS wordt terug hoof zal de data uitgelezen worden.



**SDO:** gaan wij niet gebruiken omdat je dan dn bv meerdere potentiometers er kunt aan hangen en dan heb je een hele slinger van componenten die in serie aan elkaar hangen. Maar wij willen dat als SPI programmeren deze potentiometers wat natuurlijk ook zeker kan want als je deze potentiometers hebt met 6 potentiometers dan heb je deze pin niet maar voor de rest is alles het zelfde dus je kan dit perfect als SPI protocol programmeren.

**An, Wn, Bn:** zijn de pinnen van de potentiometer.



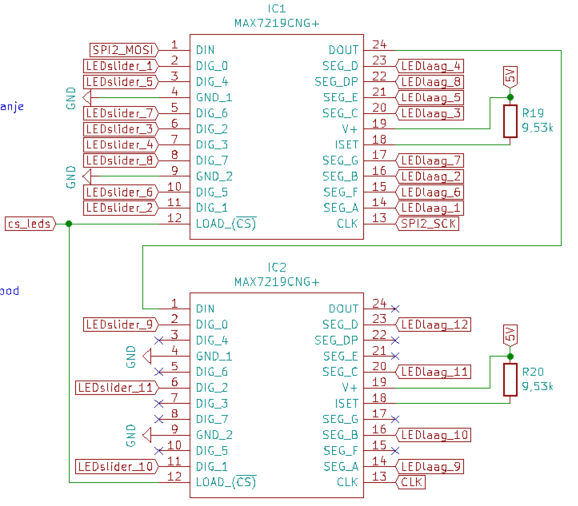
**PR:** Zorgt ervoor dat de waardes resetten als het laag is en terug op de begin waarden komen te staan.

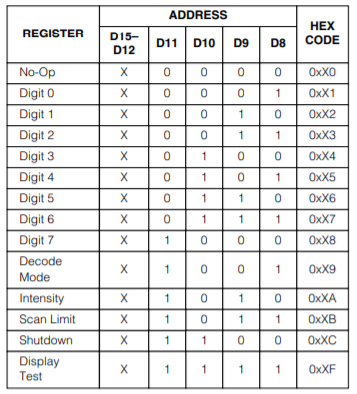
**SHDN:** is de shutdown van de potentiometers zelf. Het zet de potentiometers af als het signaal laag is.

**Vdd en Vss:** De Vdd is de spanning die de potentiometer gaat voeden en is ook de spanning die er maximum door de potentiometers mag vloeien en Vss is de minimum spanning die er door de potentiometers mag vloeien. Maar waarde Vdd moet tussen de 3V a 5V DC vloeien en Vss moet tussen de 0V a -2,7V DC vloeien en het verschil tussen Vdd + Vss mag niet hoger zijn dan 5,5V.

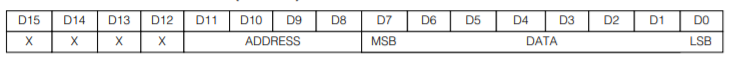
## 8 digit LED display driver (MAX7219CNG+)

Deze IC wordt vak gebruik voor 7 segment displays, maar je kan het ook voor andere displays of matrixen gebruiken. Zo hebben wij zelf onze LED matrix zelf gemaakt om zo de niveaus aan te duiden want zo kan je zien op welk niveau de potentiometers zich bevinden. Je moet de anodes en de kathodes samen steken om een matrix te maken.

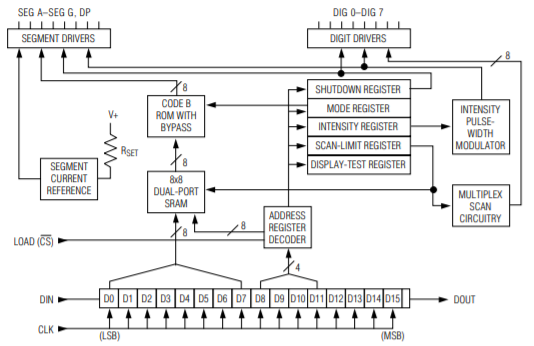


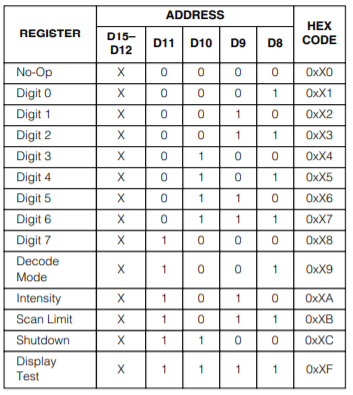


De communicatie naar deze chip werkt het zelfde als de potentiometers, er moeten 4 bits voorzien worden om adressen door te geven dit is bit 11, 10 , 9 en 8. Daarna heb je ook de data die verzend we met 8 bit dit is bit 7(MSB) tot bit 0(LSB).

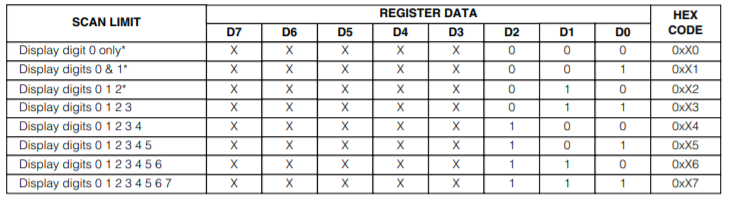


Elke bit voor de adressen heeft een eigen functie zo zie je in de tabel hieronder welke binaire waarde voor welke functie gebruikt wordt. Zo zie je ook dat de eerste 4bit geen functie heeft D15 tot D12.





De tabel hierboven verteld over de digits. De onderste tabel is voor de segmenten. Zo laat de tabel zien dat niet alle data bruikbaar is om de segmenten te doen op te lichten. Natuurlijk zou deze code niet goed gaan bij een 7 segment display maar daar hebben ze een andere codering voor. Maar die hebben wij niet nodig, wij gebruiken deze code.

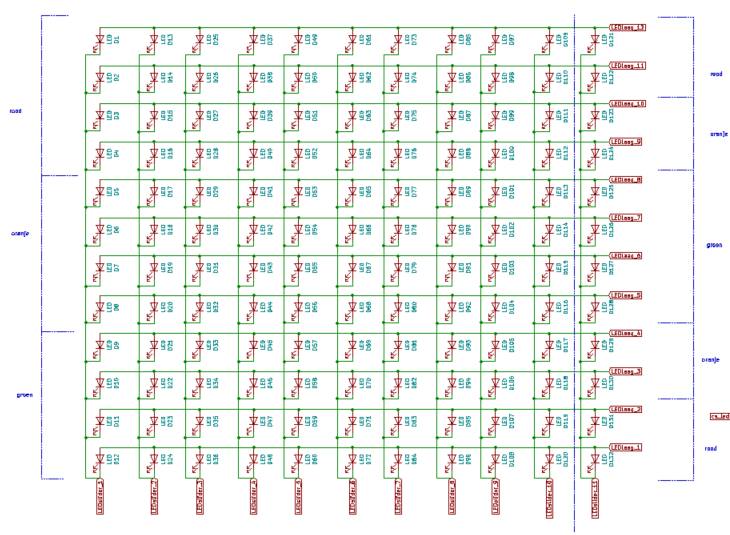


Onze led matrix is voorzien van 132 LED’s dus komen we met 1 8digit-LED display driver niet toe dus moesten we dit met 2 8digit-LED display driver doen. Deze kunnen we achter elkaar zetten zodat we via seriële communicatie kunnen doorlussen. Omdat wij ook verschillende kleuren gebruiken om aan te duiden op welke volume we zitten hebben we gekozen voor de kleuren groen oranje en rood te gebruiken.

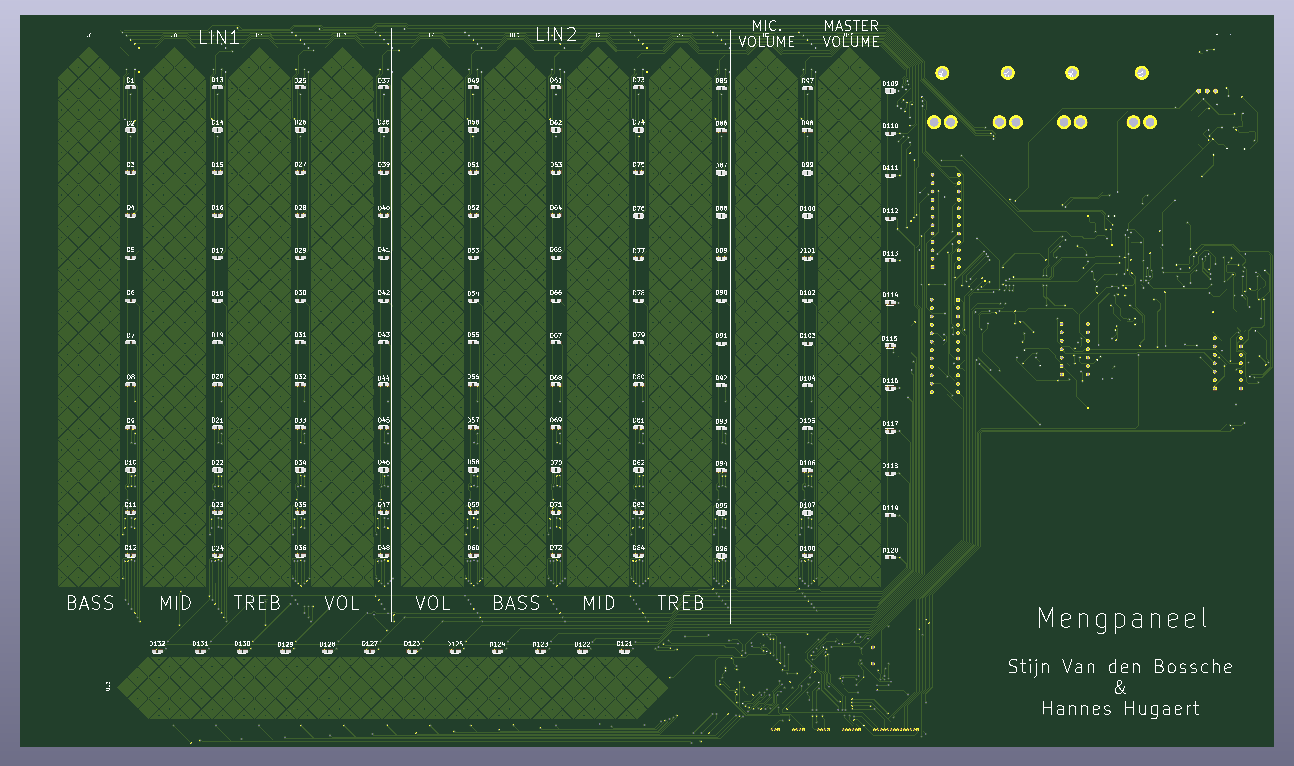
Groen betekent dat het geluid of regel circuit nog rustig staat.

Oranje betekent dat het volume of regel circuit al luider en zwaarder wordt.

Rood betekent dat het volume of regel circuit op het luidste en zwaarste staat.



Elke LEDslider heeft zijn eigen slider waar het naast komt te staan de leds staan mooi allemaal geschikt naast de sliders zoals ze hierboven staan. Door het schema te zien van hierboven kan je op de PCB zien welke LEDslider lijn voor welke slider dient.





## Versterking

Onze PCB is ook nog voorzien van een volledig circuit om de volumes, Bassen, mediums en trables te regelen op 2 lijnen waar muziek op komt en ook op lijn 1 waart je ook kan opteren om een microfoon te voorzien. Ik ga aan de hand van grafieken laten zien hoe elke versterking werkt laten zien.

Dit laat ik zien in het hoofdstuk PSpise.

## **PSpise**

## **Bestukkingsplan**

## **PCB lay-out**

## **Datasheets**

# **Software**

## **Flowchart software**

## **Listing (de code)**

# **Bediening handleiding**

# **Besluit**

# **Opgezochte sites**