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](#_Toc69742063)

[2. Hardware 3](#_Toc69742064)

[**1.1.** **Blokschema** 3](#_Toc69742065)

[**1.2.** **Schema** 5](#_Toc69742066)

[1.2.1. Communicaties 7](#_Toc69742070)

[**1.3.** **PSpise** 11](#_Toc69742071)

[**1.4.** **Bestukkingsplan** 12](#_Toc69742072)

[**1.5.** **PCB lay-out** 12](#_Toc69742073)

[**1.6.** **Stuklijst** 12](#_Toc69742074)

[**1.7.** **Aansluitgegevens** 12](#_Toc69742075)

[**1.8.** **Datasheets** 12](#_Toc69742076)

[3. Software 12](#_Toc69742077)

[**1.1.** **Flowchart software** 12](#_Toc69742078)

[**1.2.** **Listing (de code)** 12](#_Toc69742079)

[4. Bediening handleiding 12](#_Toc69742080)

[5. Besluit 12](#_Toc69742081)

[6. Opgezochte sites 12](#_Toc69742082)

# **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**

**BOX:** staat voor de boxen die aan de uitgang komen.

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

**5V omvormen naar +12V :** hier hebben we een step-up voorzien die onze 5V omzet naar +12 V en met de die 5V kunnen we ook -12V maken. Dit dient voor onze opamps te voorzien van de nodige spanning. Dit dient voor onze opamps te voorzien van de nodige voor de positieve spanning.

**5V omvormen naar –12V:** met de 5V kunnen we ook -12V maken. Dit dient voor onze opamps te voorzien van de nodige voor de negatieve spanning.

**5V omvormen naar 3.3V:** hier verlagen we de spanning van 5 V naar 3.3 V met een spanningsregelaar, dit is voor onze CPU en STM32F103CBT6 voeding te geven.

**STM32F103CBT6:** met deze kunnen we communiceren van de USB poort naar de CPU. Want onze CPU kan geen rechtstreekse signalen uitlezen van de computer daar is er geen poort voor voorzien.

**CPU:** onze CPU is van de SMT32F4x familie en die wordt gevoed met 3.3V en we willen ook ongeveer 50 I/O.

**LED’s:** deze led’s tonen aan hoe luid en hoe hoog of laag onze tonen zijn geregeld.

**Touchpad’s:** 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.

**Sturen van weerstanden voor het geluid:** wij hebben gezocht naar een oplossing om onze potentiometers te kunnen sturen. Zo kwamen

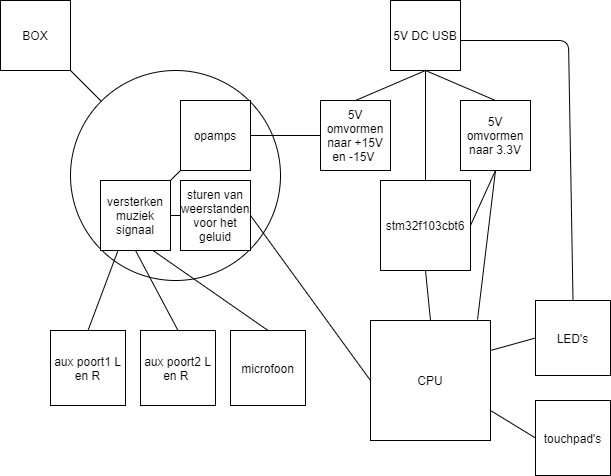
we uit op gestuurde potentiometers.

Dit hoort samen

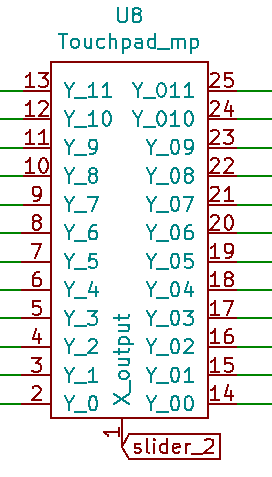
**Versterken muziek signaal:** wij gaan onze muziek versterken met opamps en ook nog met de nodige andere componenten.

**Opamps:** Dit zijn gewoon onze opamps.

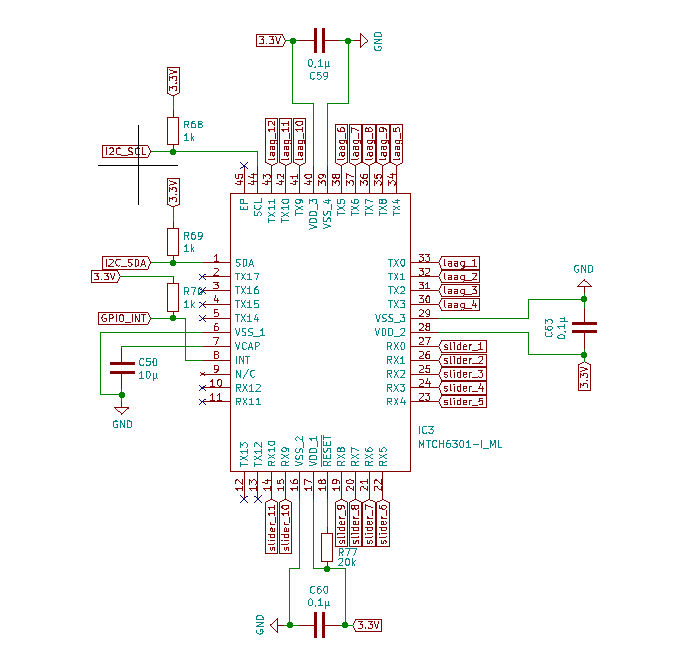
**De AUX en microfoon poorten:** dit zijn onze ingangspoort die je kan inpluggen meteen een jacket(AUX).



## **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.

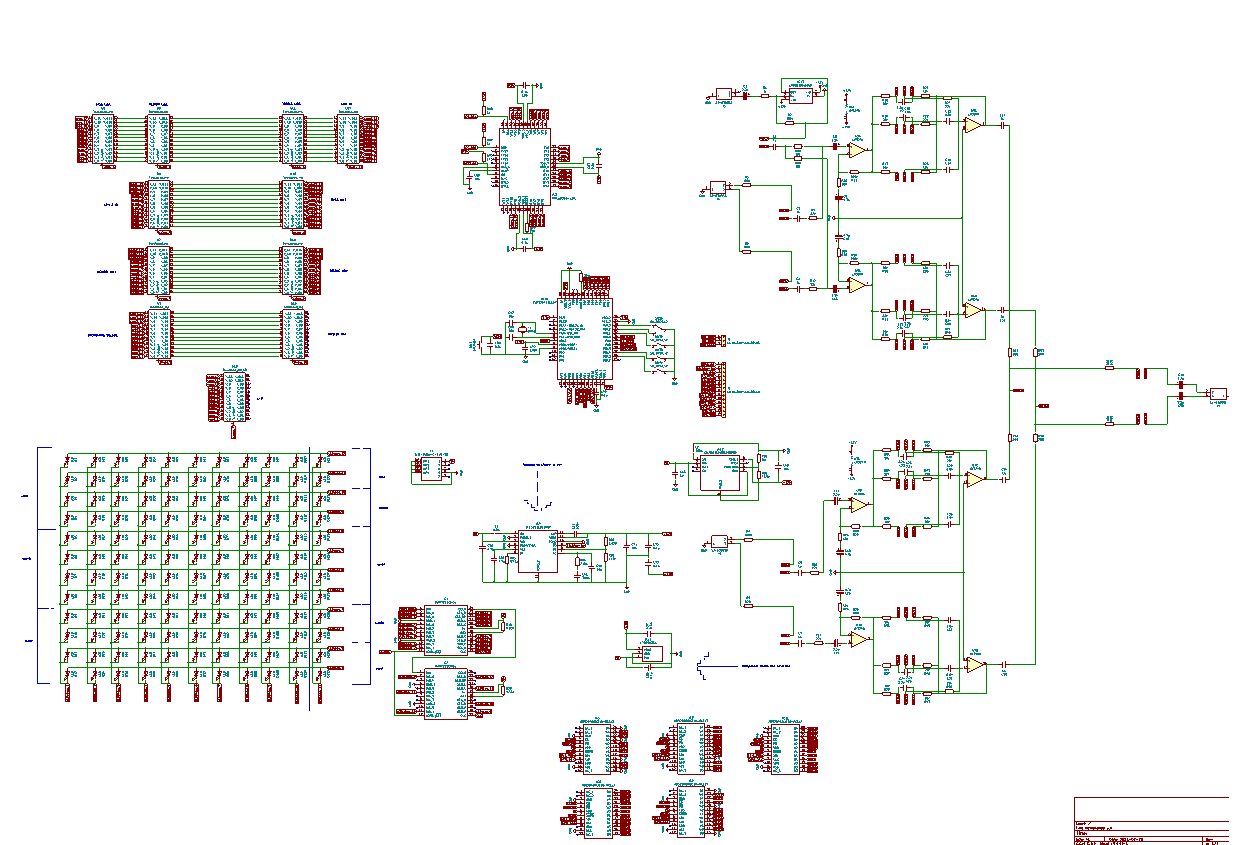


**MTCH6301-I\_ML:** dit is de IC die de signalen op de touchpad’s uitleest en om zet in data die over I²C naar de microcontroller gaat.

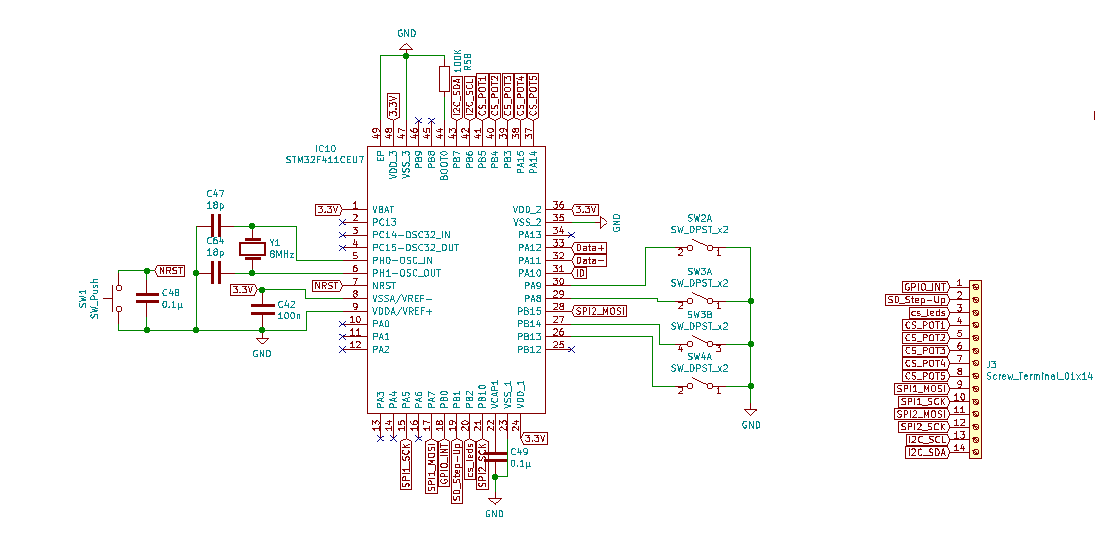
Deze condensatoren zorgen er voor dat er geen storing op de 3.3V zit, zo blijf je een ongeveer een mooie rechte 3.3V hebben.

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.

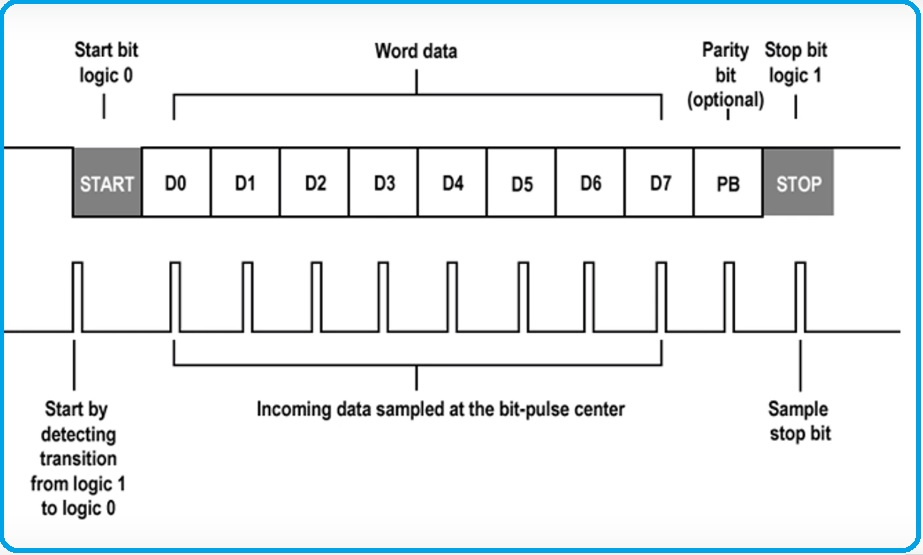
Op de microcontroller zijn er twee poorten voorzien voor een oscillatie en wij hebben gekozen voor PH0 en PH1 omdat de andere een oscillatie maar van 32 kHz max mag voorzien en dat met het gebruik van een RTC, deze bevind zich op pinnen PC14 en PC15. Door dat we een de kristal gebruiken krijgen we een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



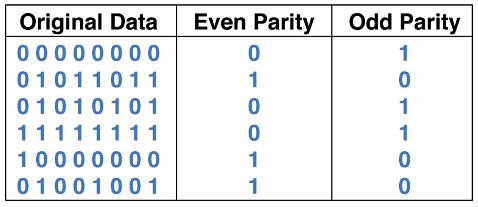
Binnen komende data

CLK die hoog word

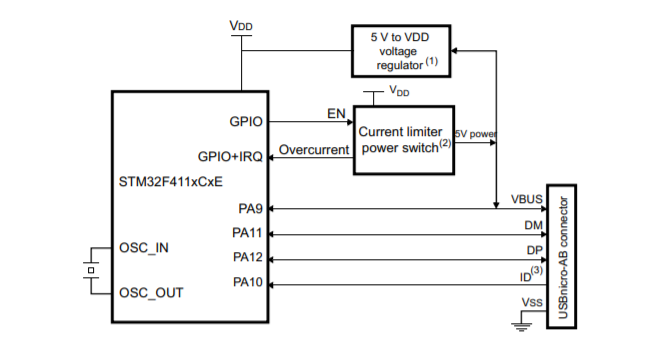
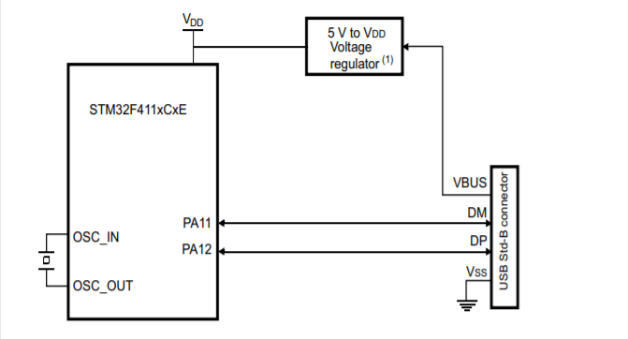
als je data moet

uitlezen

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

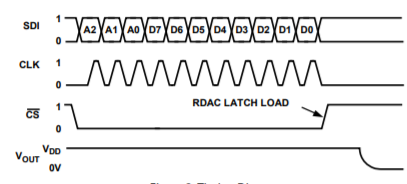
De data van UART word altijd maar uitgelezen als de binnenkomende data eerst laag wordt dit is de start bit en dan heb je de 8 bit of 1byte die zowel hoog of laag kunnen zij maar natuurlijk niet tegelijk natuurlijk. Na de 8 bit komt de parity bit deze is er niet altijd maar het detecteert altijd 1 fout 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.

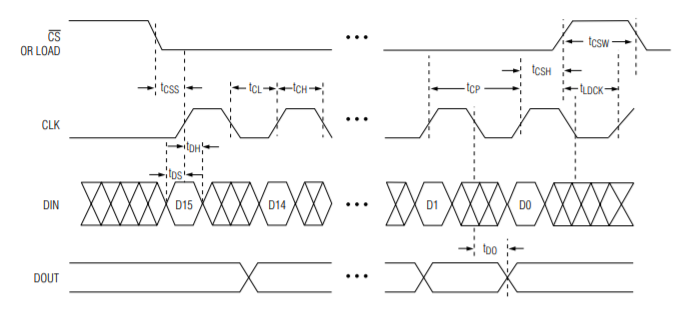
Doordat we de STM32F411CEU7 kunnen we niet programmeren met de “USB to ttl” dus dan programmeren we niet met het protocol Uart maar wel met seriële communicatie. De datasheet verteld dat we onze pinnen van de micro USB connector AB op pinnen PA9 tot en met PA12 of micro USB connector B op de pinnen PA11 en PA12 moeten verbinden. Wij hebben gekozen om een connector B te gebruiken omdat we die al hadden liggen.



**SPI**

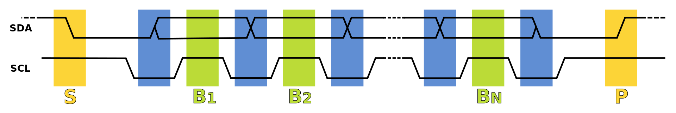
Als de Chip Select (CS) een laag signaal heeft en die doorstuurt naar een bepaalde slave, want wij gebruiken er meerdere. Zo kan de data alleen naar de bepaalde slave’s verzonden worden die een laag CS binnen krijgen.

* **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**

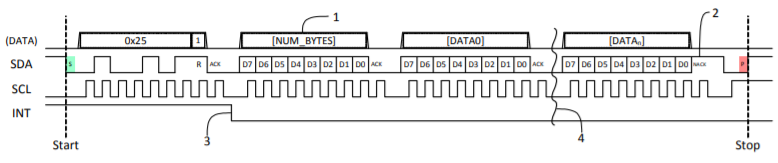
De IC voor onze touchpad’s worden aangestuurd met I²C. Dit protocol wordt over 2 lijnen gestuurd de lijn SDA (data line) en SCL (clock line). De lijn data is geen data die binnenkomt op de chip maar toont aan wat voor data er binnenkomt.



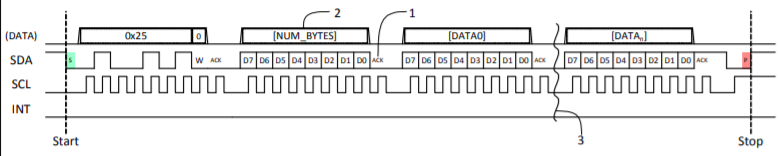
**Start:** De chip begint te starten met lezen van data als de SDA lag wordt en SCL hoog is dit is dan de start bit, daarna moet SCL eens laag worden en dan kan het data beginnen ingelezen worden. Per keer dat SCL hoog wordt de data op SDA uitgelezen.

**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



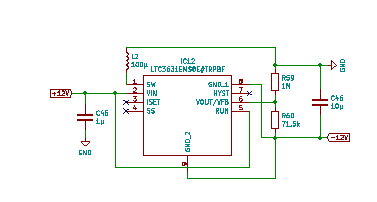
**Write:** De eerste data dat binnen gelezen wordt is het adres + een write bit zodat de master zegt dat hij de data wilt uitlezen. De INT blijft hier laag zodat het altijd kan dat verzenden.



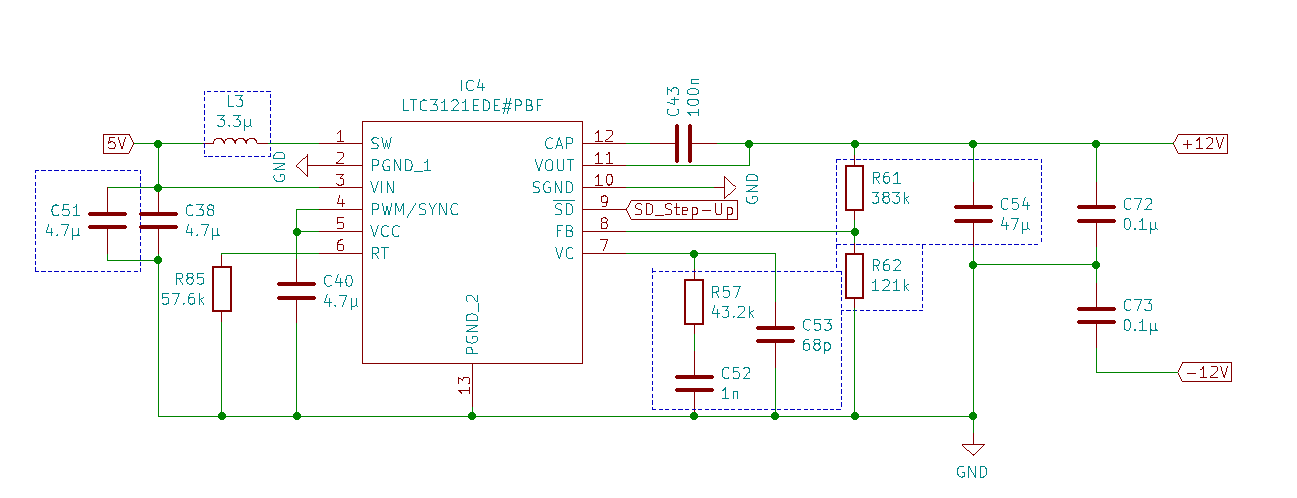
**Stop:** De stop bit wordt aangeduid als de SCL al eventjes hoog is en daarna SDA hoog wordt.

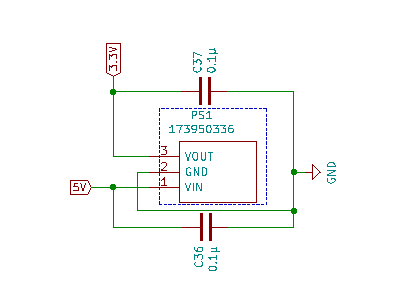
## Spanningsregelaars

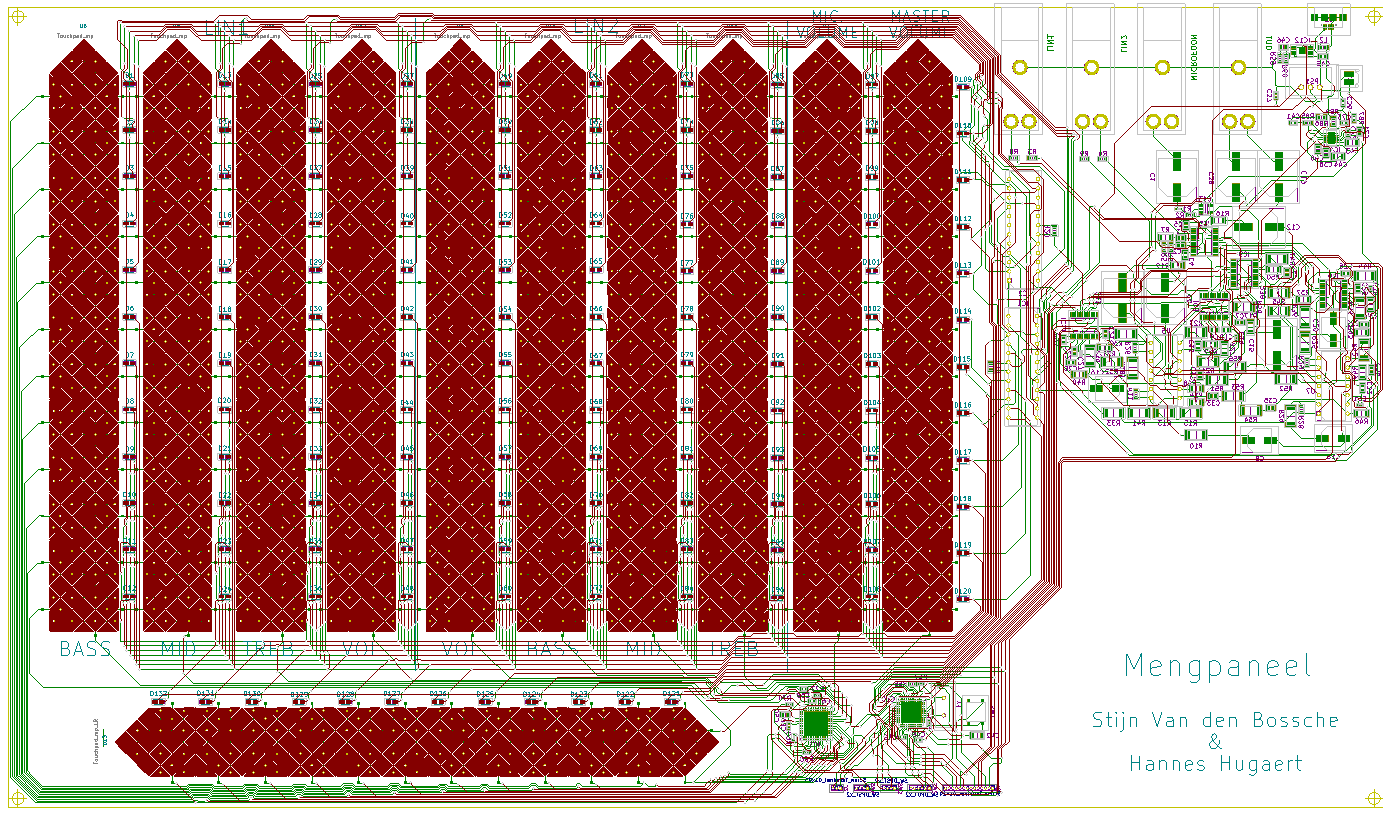
**LTC3631EMS8E#TRPBF** is een step-down die ons -12V gaat geven voor de opamps. We hebben deze Step-down aan de +12V gehangen zodat we 50mA hebben die we uit de -12V kunnen halen.

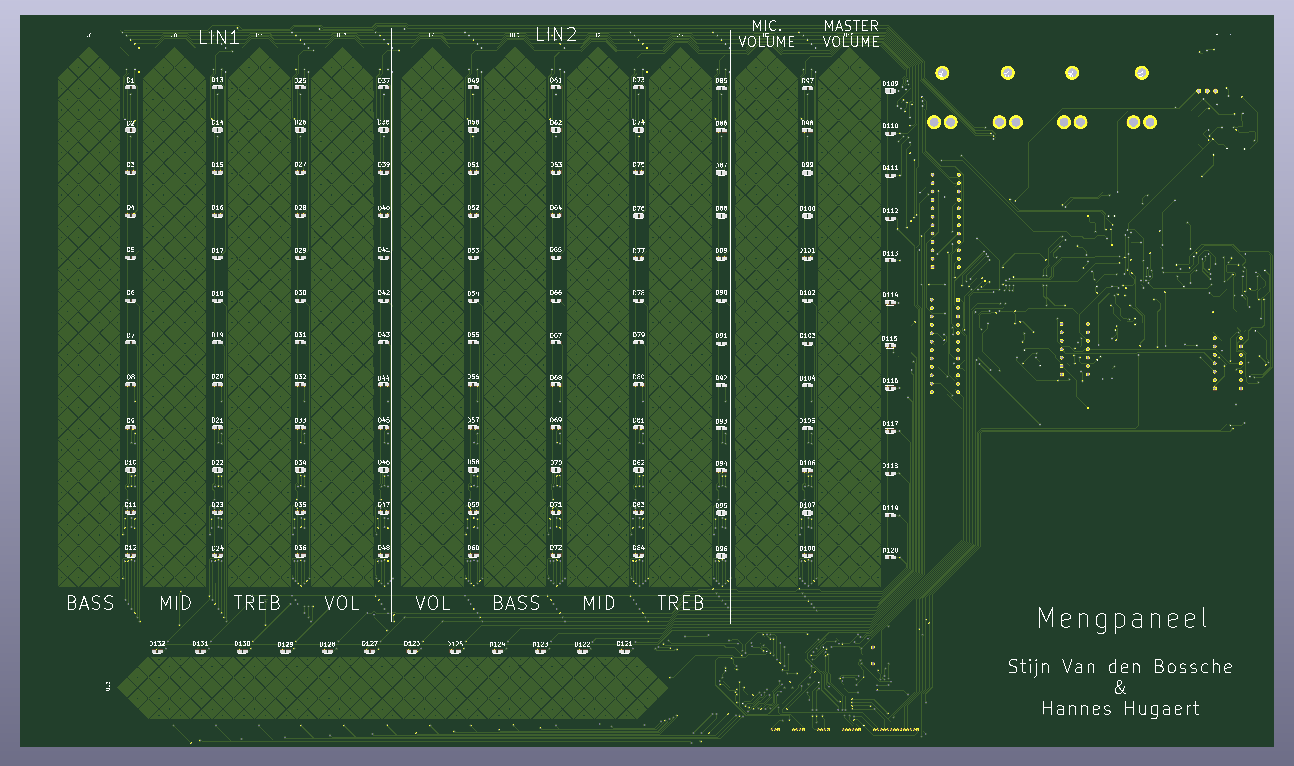


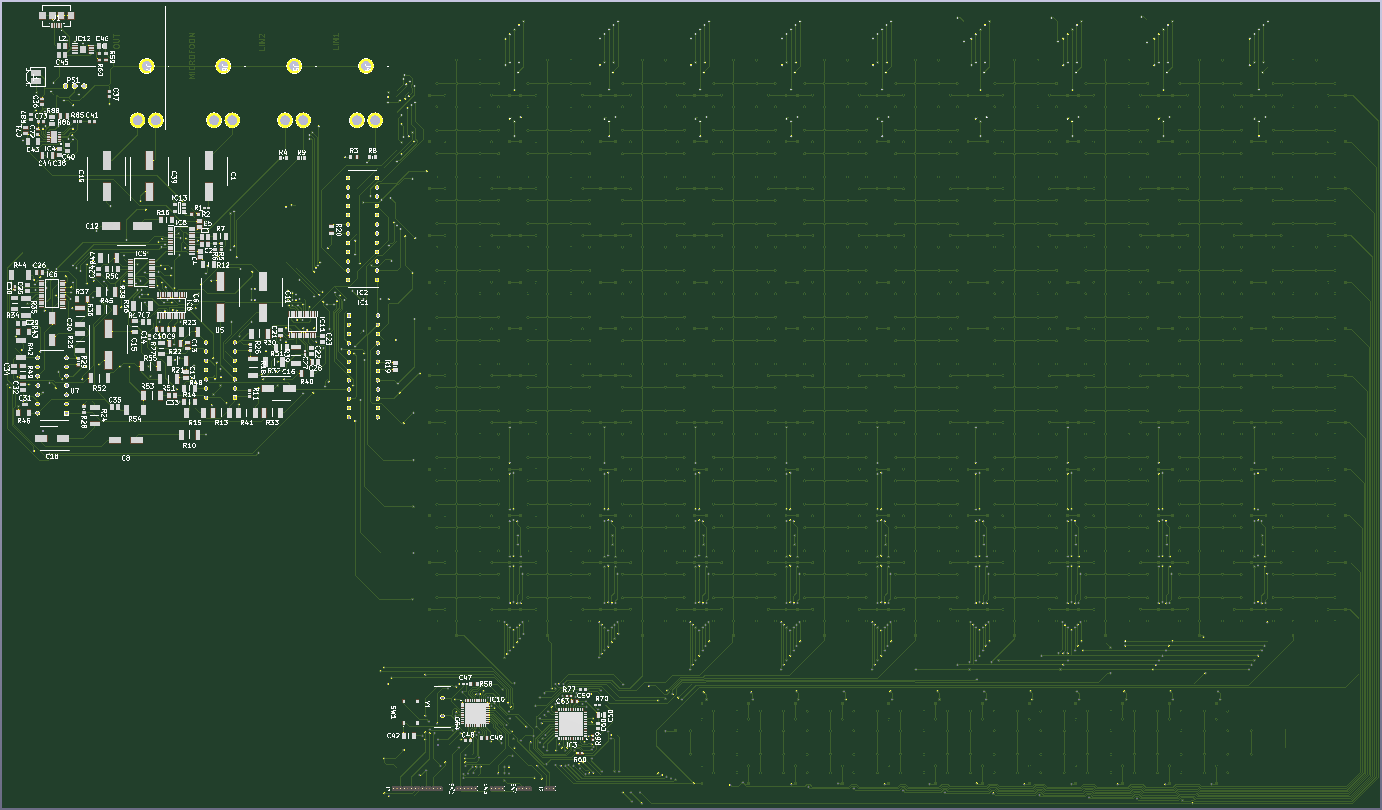
**LTC3121EDE#PBF** is een step-up die onze binnenkomende voltage omzet naar +12V om











## **PSpise**

## **Bestukkingsplan**

## **PCB lay-out**

## **Stuklijst**

## **Aansluitgegevens**

## **Datasheets**

# **Software**

## **Flowchart software**

## **Listing (de code)**

# **Bediening handleiding**

# **Besluit**

# **Opgezochte sites**