|  |  |
| --- | --- |
|  | ELEKTRONICA-ICT  Elektronische systemen 2 - 2020-2021 |

**16 X 16 Image sensor**

|  |  |
| --- | --- |
| Author  Author  Lector | Tom Haels  Moris Damian  Frederik Vreys Wout Swinkels |

Content

[1 Introduction 2](#_Toc73624774)

[2 BOM 2](#_Toc73624775)

[3 Tools 3](#_Toc73624776)

[3.1 Assemblage 3](#_Toc73624777)

[3.2 Testing 3](#_Toc73624778)

[4 Electrical circuit 3](#_Toc73624779)

[5 PCB design 4](#_Toc73624780)

[6 PCB assembly 4](#_Toc73624781)

[7 Software 4](#_Toc73624782)

[8 Results 6](#_Toc73624783)

[9 Conclusion 6](#_Toc73624784)

[10 Reference list 6](#_Toc73624785)

[11 Attachment 6](#_Toc73624786)

# Introduction

Hoe kunnen we met behulp van elektronica een hoeveelheid licht waarnemen? Hoe kunnen we dit omzetten naar een meetbaar signaal? Welke componenten hebben we nodig om dit te verwezenlijken? Hoe kunnen deze waarden digitaal uitgelezen worden? Om een antwoord te krijgen op deze vragen hebben we zelf een 16 x 16 pixel image sensor ontwikkeld. Naast de image sensor zelf hebben we er ook voor gekozen om nog een kleine aparte test PCB te ontwerpen. Deze test PCB dient voor het testen van de analoge elektronica die ook op de volledige image sensor zit. Het is een makkelijkere manier om eventuele fouten in de versterkingsschakeling te kunnen debuggen. In deze application note gaan we bespreken welke stappen we hebben doorlopen om dit te bereiken. Als eerste zal er een elektrische schema moeten ontwikkeld worden. Op basis van dit schema kan een PCB design ontwikkeld worden. De waarde aan de output zal mogelijk uitgelezen kunnen worden via een microcontroller. Op deze manier krijgen we een beter inzicht hoe een camerabeeld voorgesteld kan worden op een OLED display. Als inspiratiebron hebben we een project online gevonden genaamd DigiObscura. In dit project wordt ook een self made image sensor ontwikkeld. <https://www.instructables.com/DIY-Image-Sensor-and-Digital-Camera/>

# BOM

|  |  |  |
| --- | --- | --- |
| BOM 16 x 16 image sensor | | |
| Component | Waarde | Aantal |
| Fototransistor | 1540601NBA500 | 256 |
| Opamp | LM358D | 2 |
| Mosfet | NDS7002A | 1 |
| Comparator | LM397MFX/NOPB | 1 |
| 555 Timer | LM555CMM/NOPB | 1 |
| Diode | 1N4004G | 2 |
| Multi/Demulti-plexer | CD74HC4067M96M4 | 2 |
| Header | 6-pin female | 2 |
| Header | 5-pin female | 1 |
| Weerstand | 1 kOhm  10 kOhm  100 kOhm | 5  2  1 |
| Condensator  (gepolariseerd) | 100 nF  1 nF  100 nF | 2  1  2 |

|  |  |  |
| --- | --- | --- |
| BOM test circuit | | |
| Component | Waarde | Aantal |
| Fototransistor | 1540601NBA500 | 1 |
| Opamp | LM358D | 2 |
| Mosfet | NDS7002A | 2 |
| Header | 5-pin male | 1 |
| Weerstand | 100 kOhm  1 kOhm | 2  1 |
| Condensator | 22 µF  1 nF | 1  1 |

# Tools

## Assemblage

* Multisim
* Altium Designer
* Soldeerbout
* Soldeeroven

## Testing

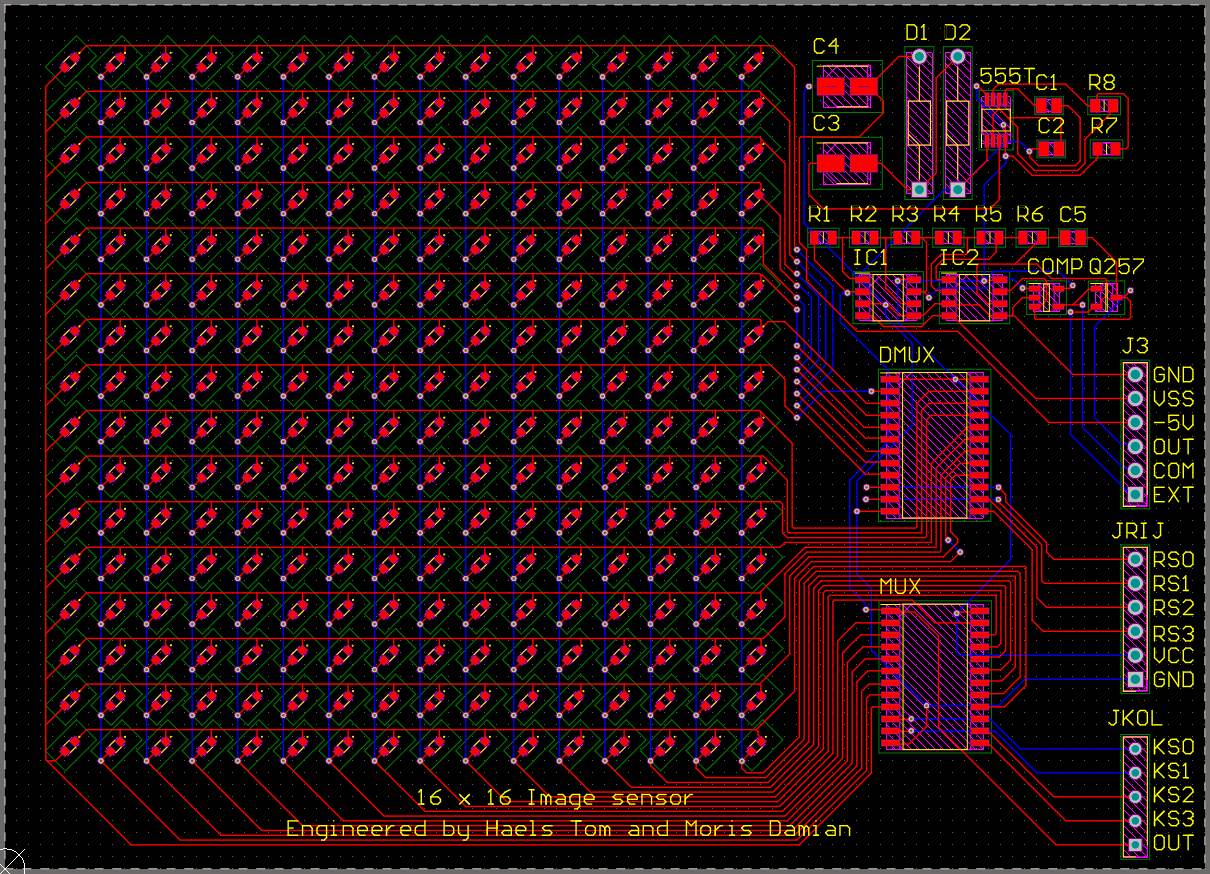
* Multimeter
* Oscilloscoop
* Labovoeding
* Arduino Nano
* Arduino Software

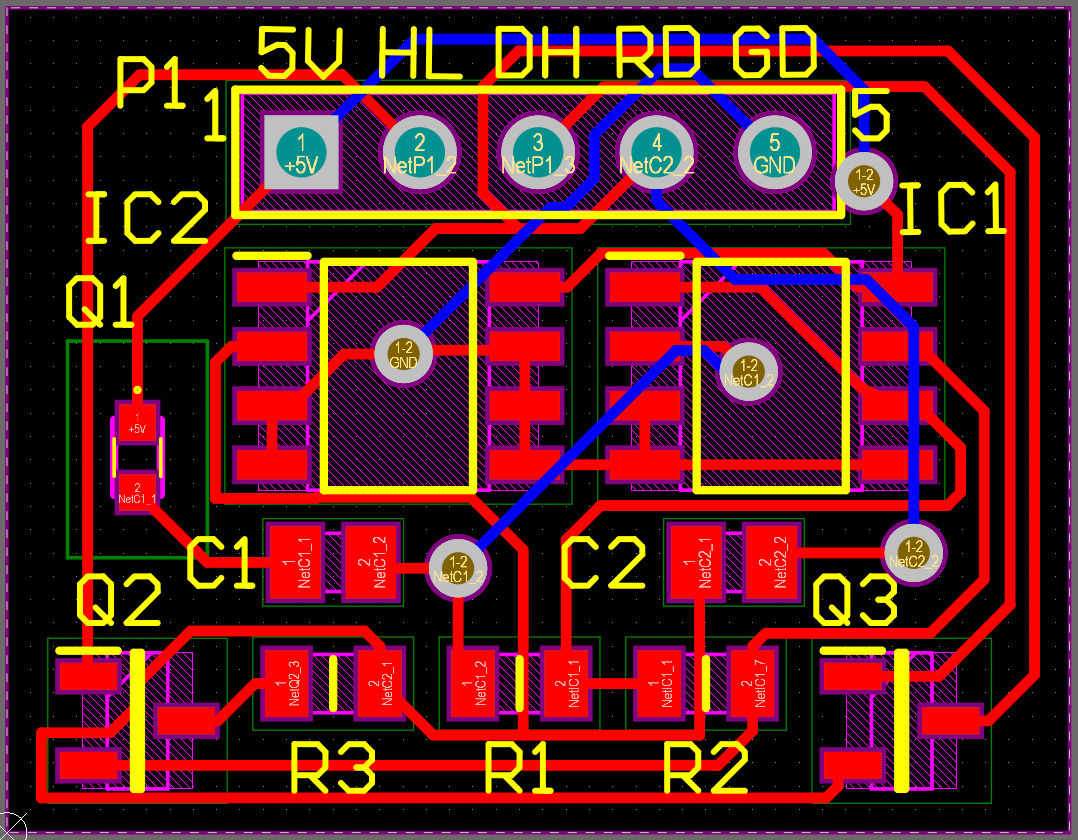
# Electrical circuit

* Het elektrische schema!
* Hoe werkt de schakeling?
  + Voeg berekeningen m.b.t de schakeling hiertoe

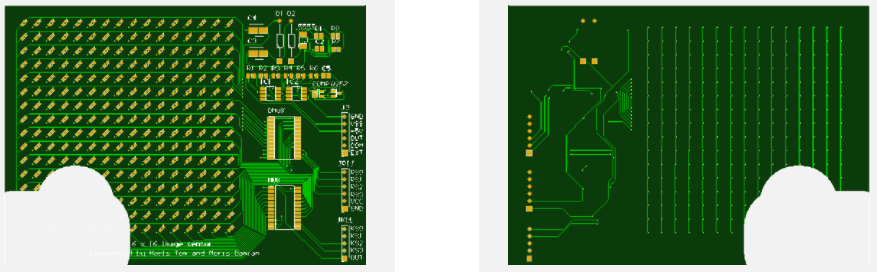
# PCB design

Nadat het elektrisch schema getest was in Multisim en overgenomen in Altium Designer kon de PCB getekend worden. De PCB is ook volledig ontworpen in Altium Designer met de footprints die bij de componenten zitten op Mouser. Dit bespaart veel tijd en moeite omdat de footprints niet zelf getekend hoeven worden en je weet dan ook zeker dat de footprints correct zijn aangezien dat deze van de fabrikant zijn. Voor de fototransistors is het belangrijk om ze in een regelmatige vorm te plaatsen zodat de lichtinval gelijkmatig gemeten kan worden en uiteindelijk dus ook duidelijker kan gevisualiseerd worden. Hier is gekozen voor een vierkante configuratie, deze leek het meest logisch te zijn. Voor de rest maken de posities van de componenten niet echt uit voor de werking van het schema. Maar er is wel op gelet dat de PCB er verzorgd uitziet door een propere lay-out van componenten te behouden.





De test PCB is besteld op JLCPCB maar de image sensor is besteld op PCBWay aangezien dat de gerber files van de image sensor op JLCPCB op onverklaarbare reden niet correct inladen.

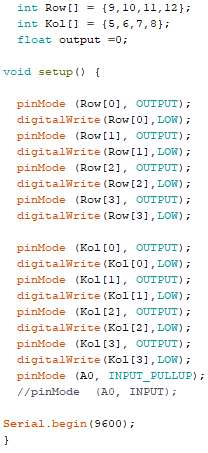


# PCB assembly

* Welke tools zijn gebruikt?
* Waar dien je rekening mee te houden? Zijn er componenten die moeilijk te solderen zijn?
* Tips voor het solderen van bepaalde componenten!

# Software

In dit hoofdstuk gaan we dieper in op de code die nodig is voor het aansturen van de multiplexer/demultiplexer en voor uitlezen van verkregen outputsignaal van elke fototransistor. De microcontroller die we hiervoor hebben gebruikt is een Arduino. In onderstaande afbeelding is de code ontleed in meerdere delen om elk aspect te bespreken.



De digitale input pinnen voor het aansturen van de demultiplexer worden in een array “Row” gedeclareerd. Voor de multiplexer is dat array “Kol”. Het outputsignaal gaan we analoog uitlezen op pin A0 die we output noemen.

Al de pinnen worden laag geschakeld en de analoge pin als pullup. Dit zorgt ervoor dat we geen zwevende waarden kunnen meten. Als er geen signaal wordt waargenomen zal de spanning op deze pin 5V bedragen. Moest deze gewoon als input gedeclareerd zijn zullen we ruis op deze pin waarnemen.

Onderaan de code vinden we “Serial.begin(9600)”. Dit zorgt ervoor dat we de waarden die geprint worden op of naar de seriële monitor kunnen lezen op een baud rate van 9600.



Het volgende deel van de code is opgedeeld in twee delen. Het bovenste deel is een functie genaamd BinaryOut en het tweede deel is de main code. De functie BinaryOut zal met behulp van de main code uitgelegd worden.

Het concept van de code is als volgt. De bedoeling is dat elke fototransistor apart van links boven naar rechts onder wordt uitgelezen. Met de demultiplexer zullen we elke rij beurtelings van een hoog signaal voorzien. Per rij zal elke kolom van links naar rechts uitgelezen worden met behulp van het aansturen van de multiplexer. Als al de 16 phototransistors zijn uitgelezen zal de rij gesloten worden en de volgende rij open geschakeld worden.

We maken gebruik van twee for loops. In de eerste for loop doorlopen we elke rij. Door middel van de functie BinaryOut[number,pins[]) geven we met number telkens mee de hoeveelste fototransistor dat moet uitgelezen worden in een bepaalde rij of kolom. Met pins[] geven we de array Row of Kol mee. Als we dieper in deze functie kijken kunnen we de functie bitread terug vinden. Bij bitread(number,i) geeft number een getal in binaire vorm weer dit in ons geval maximaal 15 is. I stelt de bitlocatie van number van rechts naar links voor. Door i in een for loop te laten oplopen van 0 tot en met 3 wordt elke bit die nodig is om een binaire waarde in number doorlopen en gecontroleerd op 0 of 1. Doordat Row en Kol ook een array met 4 elementen zijn representeert dit de locatie van elke bit. Afhankelijk van de binaire waarde zal de respectievelijke digitale pin hoog of laag geschakeld worden. Hierdoor wordt elke waarde in elke rij en kolom doorlopen. Als laatste onderdeel wordt de analoge waarde A0 in output uitgelezen. Bij een Arduino nano is de standaard analoge output tien bits. Dit is een waarde van maximaal 1024. We willen dat ons uitganssignaal een waarde representeert tussen 0 en 5. Dit kunnen we verkrijgen door een simpele berekening (output\*5)/1024. Deze berekening hebben we in variabele number geplaatst die we ook uitlezen door Serial.print(number).

Is dit duidelijk??? Moet het simpeler geschreven?

# Results

* Sensor kalibratie
* Sensor meetresultaten

# Conclusion

* Reflecteer hier over je resultaten
* Aanbevelingen
* Bevat geen nieuwe informatie!

# Reference list

The current file doesn't have any references.

# Attachment

* Informatie die relevant is maar niet binnen de AN past

Afgeprint kan bijlage zich beperken tot een opsomming die te raadplegen is digitaal.