# Introductie

## Woord vooraf

Dit is het eindverslag van het jaarproject voor het vak Practice Enterprise.

In het 2de bachelor jaar staat opnieuw een project op het programma voor het vak Practice Enterprise, wat ook de enige evaluatievorm is voor dit vak. Dit project heeft als doel om de kennis die we het voorbije semester hebben opgedaan, te kunnen toepassen in de praktijk. Het is een jaarproject waarin we zoveel mogelijk zelfstandig werken en dat is meteen ook een goede voorbereiding op de bachelor proef in ons afstudeerjaar.

## Informatie auteur

Mijn naam is Thomas Leemans en ik studeer aan de Thomas More Hogeschool campus De Nayer te Sint-Katelijne-Waver. Ik zit momenteel in de 2de bachelor Elektronica/ICT.

Ik woon in Grimbergen en ben geboren op 7 juni 2000.

Emailadres: [thomasleemans1@gmail.com](mailto:thomasleemans1@gmail.com)

Mijn hobby’s zijn:

* Tennis
* Met de hond wandelen
* Games
* Gezelschapsspel met vrienden/familie

## Taakverdeling

Voor dit project hadden we de keuze om alleen of in duo te werken. Gezien er niet meteen iemand dezelfde interesse had om aan dit project te werken en gezien het door de Corona maatregelen moeilijker is om samen aan iets te werken, heb ik ervoor gekozen om dit project alleen te maken.

Hierdoor waren er dus geen taken te verdelen, maar was het vooral belangrijk om goed mijn tijd te verdelen om zeker de deadline te behalen.

Inleiding

Sinds het uitbreken van de Corona pandemie en de bijhorende lockdown, zijn mensen veel meer hun inkopen online beginnen doen. Steeds meer worden er thuis pakjes geleverd, vaak zonder aanbellen door de bezorger. De pakjes worden regelmatig zomaar in de brievenbus achtergelaten, zonder de koper (meteen) op de hoogte te brengen van de levering. Voor een aantal bestellingen kan het nadelig zijn om langere tijd in een te warme of te koude brievenbus te blijven liggen.

Dit bracht mij op het idee om een brievenbusmelder of mailalert te ontwikkelen. Het doel ervan is: telkens wanneer er iets in je brievenbus wordt gedropt, zal er een mail verstuurd worden naar het ingestelde mailadres (= dat van de eigenaar van de brievenbus). De brievenbusmelder moet dus eerst verbonden worden met het persoonlijk netwerk van de eigenaar. Wanneer dit is gebeurd, kan het systeem worden ingebouwd aan een vlakke kant van de brievenbus.

In dit verslag kan u lezen hoe dit project stap voor stap tot stand is gekomen. Ook kan u terugvinden welke componenten ik heb gekozen en waarom ik deze gekozen heb. Ten slotte geef ik ook mee wat het kostenplaatje van het totale project is.

# **Hoofdstuk 1 : probleem- en doelstelling**

Februari 2020 kwam ons land stilaan in de greep van het Coronavirus. In maart 2020 volgde de eerste lockdown, waarbij onder andere alle niet essentiële winkels de deur moesten sluiten en dit voor vele weken.

Plots kon men niet meer gaan shoppen voor de nieuwe lentecollectie, kon er geen nieuwe electronica worden gekocht of kon men geen cadeautjes meer gaan kopen wanneer iemand jarig was.

De oplossing: online winkelen! Winkels werden creatief met het opstarten van een eigen webshop waarbij hun artikelen online werden aangeboden.

Deze bestellingen moesten natuurlijk geleverd worden. B-post kon de plotse verhoging van de bestellingen niet aan en ook de pakjesdiensten bleken al snel onderbemand. Pakketjes werden (en worden nog steeds) gewoon in de brievenbussen gedropt, zonder aan te bellen bij de koper om te melden dat er iets is geleverd.

Het doel van mijn project is om mensen te verwittigen wanneer er een pakketje is geleverd in de brievenbus. Hiermee kan vermeden worden dat pakketjes te lang in ongunstige omstandigheden worden bewaard of dat erg dure pakketjes onbemand in de brievenbus achterblijven.

## Gantt Chart

*Fig 1.0 Gantt chart*

(ghantt chart aanpassen)

De fig. 1.0 is een ghantt chart waarbij ik dus een schatting heb gemaakt van de tijd die nodig zou zijn voor elk onderdeel.

Ik ben gestart met nadenken hoe ik het zou aanpakken en welke componenten ik voor dit project zou nodig hebben. Ik heb dan van de gekozen componenten de datasheets gelezen om zo te weten te komen wat de precieze werking is van elk component en hoeveel stroom elk component verbruikt.

Na het lezen en het verwerken van de datasheets, kon ik een meer gerichte keuze maken in de componenten. Zo heb ik onder andere gekozen voor een andere spanningsregelaar.

Daarna ben ik begonnen met een schema te tekenen. Mijn PCB design was vrij snel klaar. Die hoort ook bij de hardware gedeelte 1.

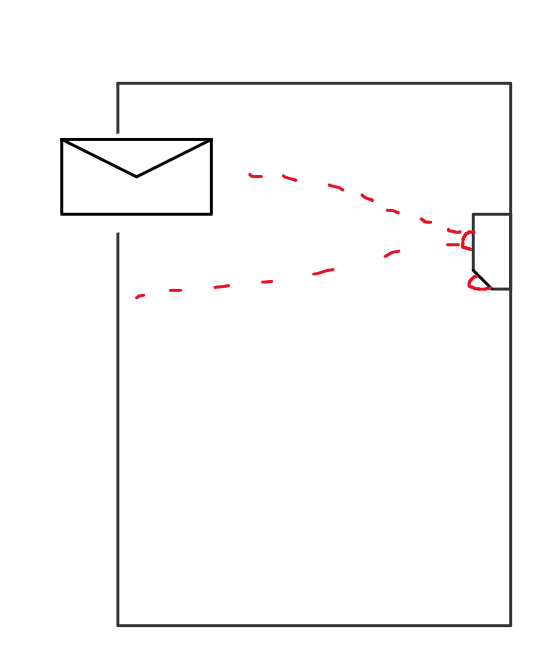
Het solderen duurde vrij lang en ging niet zo vlot. Het was voor mij de eerste keer dat ik zo’n kleine componenten moest solderen. Ik heb hierbij ook hulp gekregen van de docent.

Zoals u in mijn ghantt chart ook kan zien, ben ik gelijktijdig ook begonnen met programmeren. Ik had een ESP32 breakout gekocht om te kunnen testen of ik acces-point en wifi zou kunnen opstellen. Door het programmeren leerde ik stilaan Micropython kennen. Dit zou ik dan later nodig hebben om alles definitief te programmeren.

Eens al mijn componenten gesoldeerd, kon ik mijn programma helemaal uitwerken.

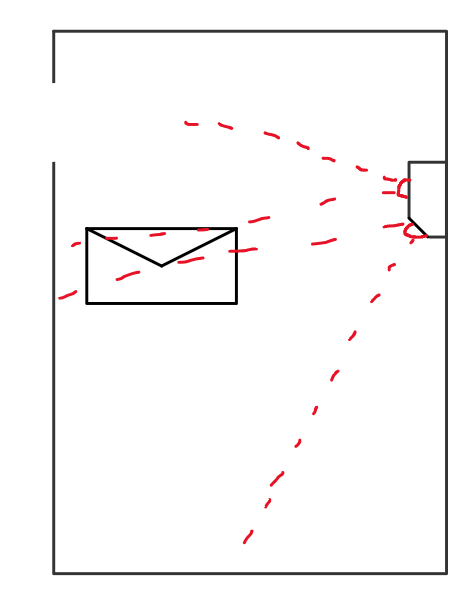
# **Hoofdstuk 2: Schetsen**

**Situatie 1:**

****

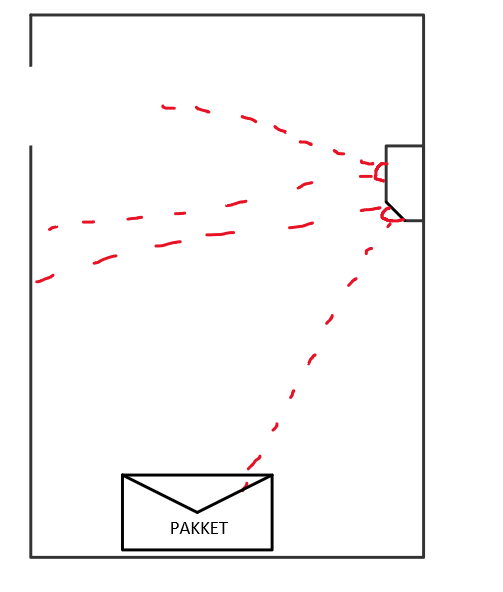
De post blijft hangen in de gleuf van de brievenbus. De 1ste naderingssensor gaat dit detecteren.

**Situatie 2:**



De post valt neer in de brievenbus. Beide naderingssensoren gaan dit detecteren.

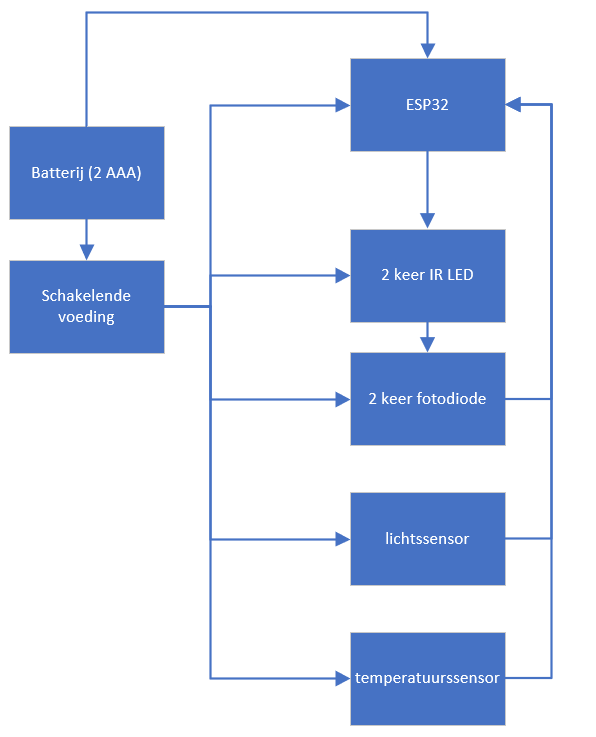
**Situatie 3:**



Een pakketje wordt via het deurtje achteraan de brievenbus in de brievenbus gelegd. Dit kan enkel bij brievenbussen die groot genoeg zijn. Enkel de 2de naderingssensor gaat het pakket detecteren.

# Hoofdstuk 3 : Toepassen

## 3.1 Blokdiagram hardware:



Het toestel gaat gebruik maken van 3 AAA batterijen zodat de batterijen niet te snel vervangen moeten worden.

Om met wifi te kunnen werken heb ik gekozen voor een ESP32. De ESP32 heeft 5 inputs en 2 outputs.

De 2 outputs zijn:

* 2 keer de IR LED: deze zorgen ervoor dat infrarood licht wordt uitgezonden.

De 5 inputs zijn:

* 2 keer de fotodiode: Deze detecteren het infrarood licht van de outputs en merken veranderingen in de afstand tot de gleuf wanneer een brief in de brievenbus wordt geplaatst.
* De lichtsensor: Die gaat detecteren of het deurtje van de brievenbus open is geweest.
* Temperatuursensor: Die gaat meten hoe warm het in de brievenbus is.
* 3 AAA batterijen: Deze zorgen voor voldoende spanning om het systeem te doen werken.

### **3.1.1 De componenten**

Om dit systeem te kunnen maken, heb ik volgende componenten gebruikt:

* ESP32-WROOM-32
* VCNL4035
* VCNL3030
* TMP36GSZ
* OPT3002
* Schakelende voeding: ADP1607ACPZN00-R7
* Weerstanden
* Spoel
* Condensatoren
* Kabels

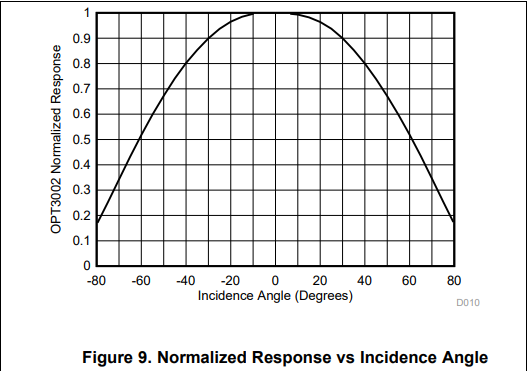
Dit systeem verbruikt veel voeding eenmaal die uit het slaapmodus zit. Gelukkig werkt dit systeem grotendeels in slaapmodus. Die gaat alleen wakker worden als er iets gedetecteerd wordt.(aantal verbruik nog zeggen)

### ESP32-S2-WROOM:

### OPT3002

OPT3002 is een licht naar digitaal sensor. Deze sensor heeft een effectief dynamisch bereik van 23-bits. Deze sensor is low-power, dit wil zeggen dat hij heel weinig verbruikt. Het verbruik van OPT3002 is: (1 waarde beschrijven en de meting)

Volgende grafiek geeft de invalshoek van deze sensor weer:



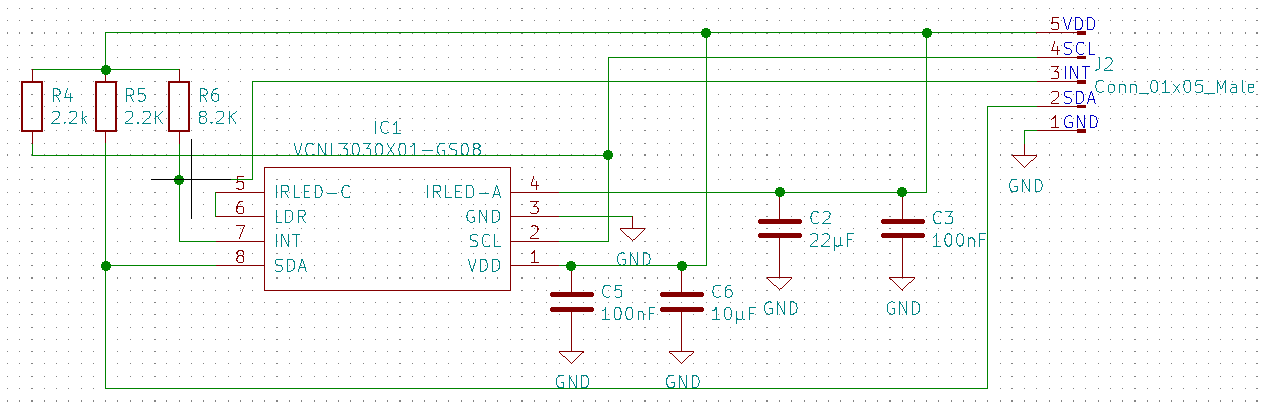
Deze sensor heeft een brede invalshoek waardoor hij bij het openen van de brievenbus nog altijd de fotonen(lichtdeeltjes) kan ontvangen.

Ik heb de OPT3002 een adres gegeven, namelijk 0x44

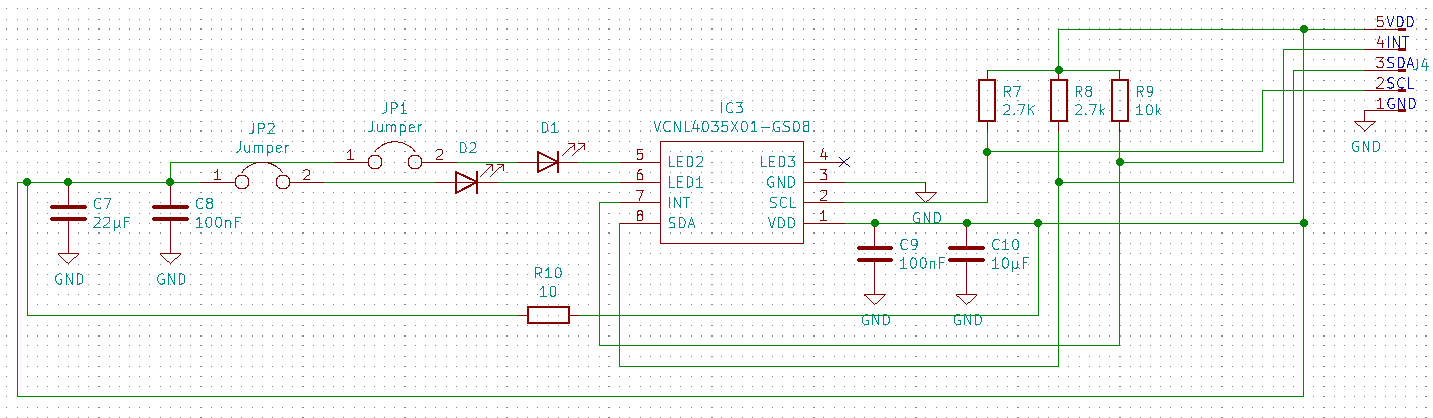
Ik gebruik deze sensor om na te kijken of de brievenbus geopend is. Zo weet ESP32 dat iemand een pakket in de brievenbus legt of iemand de brieven en/of de pakket uit de brievenbus haalt.

## 3.2 Schema’s

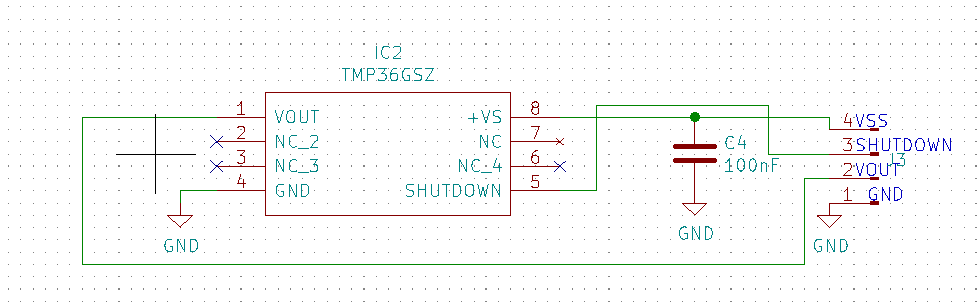
### VCNL3030



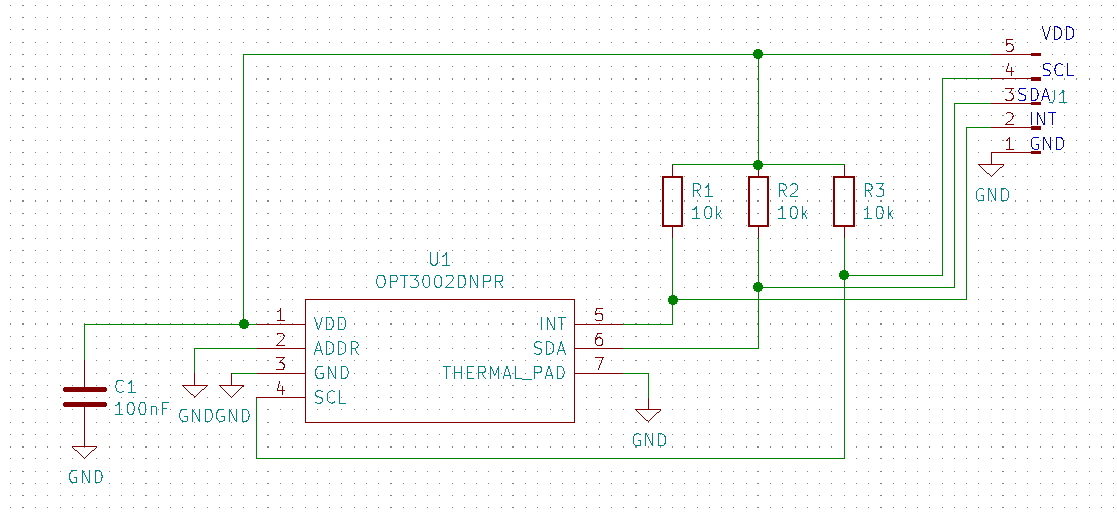
### VCNL4035



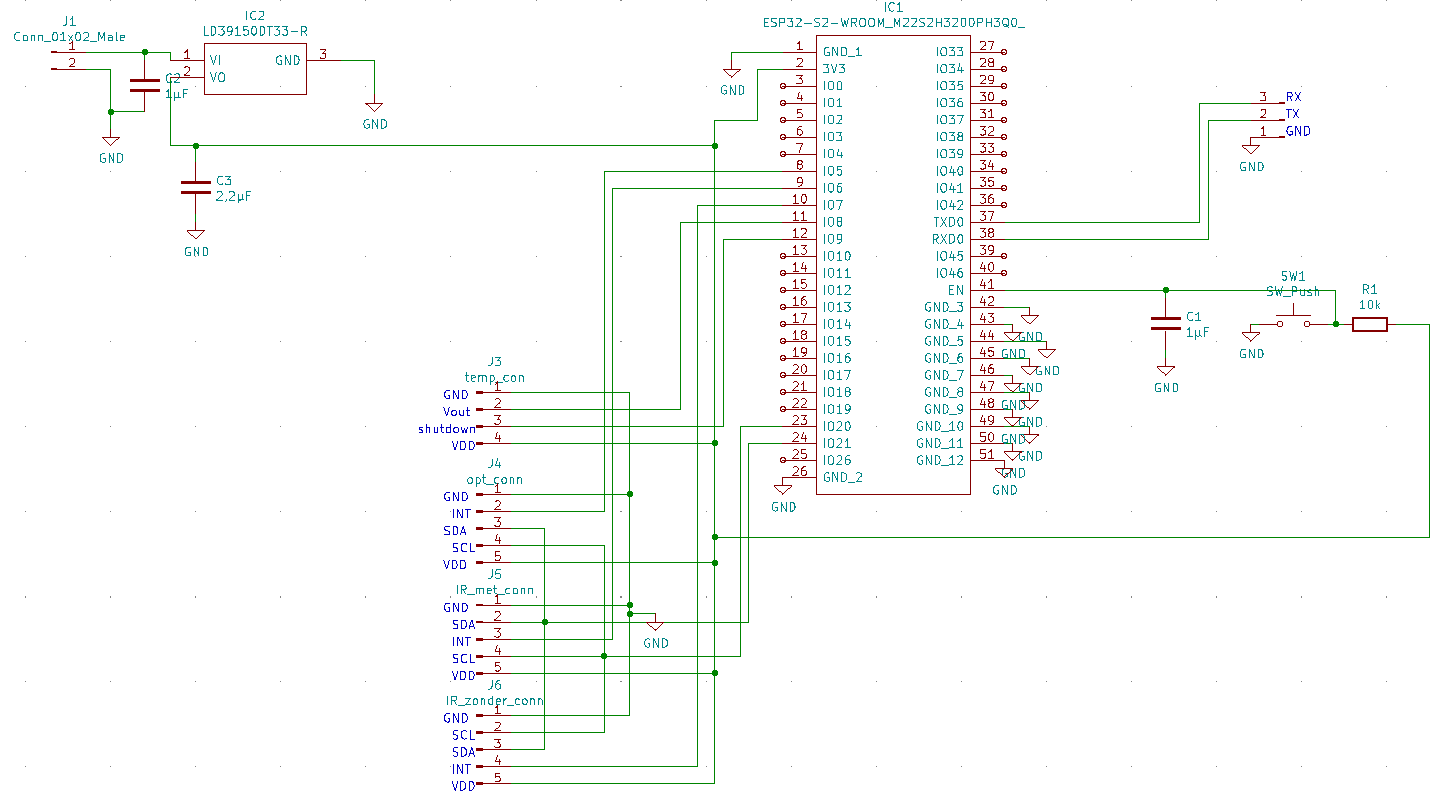
### TMP36GSZ



### Opt3002



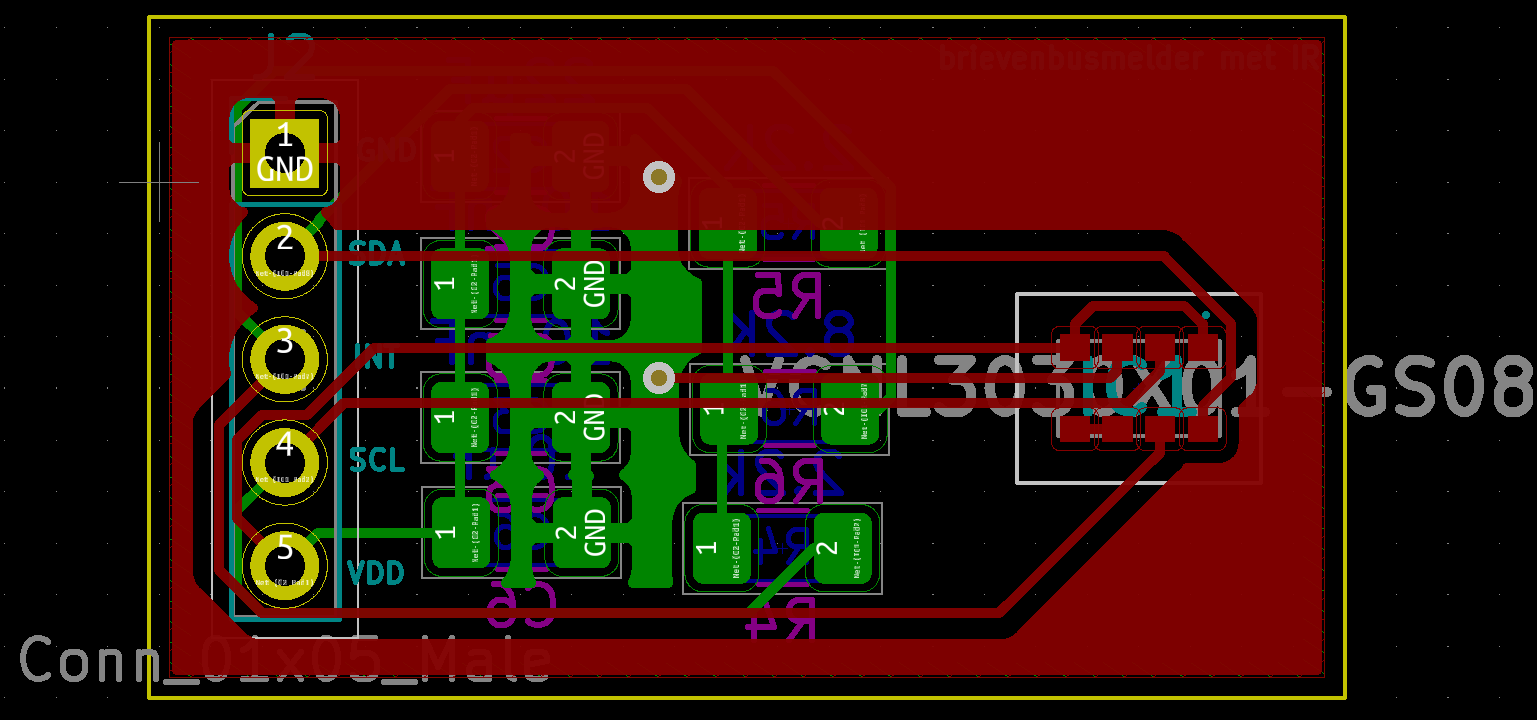
### ESP32 + LDO



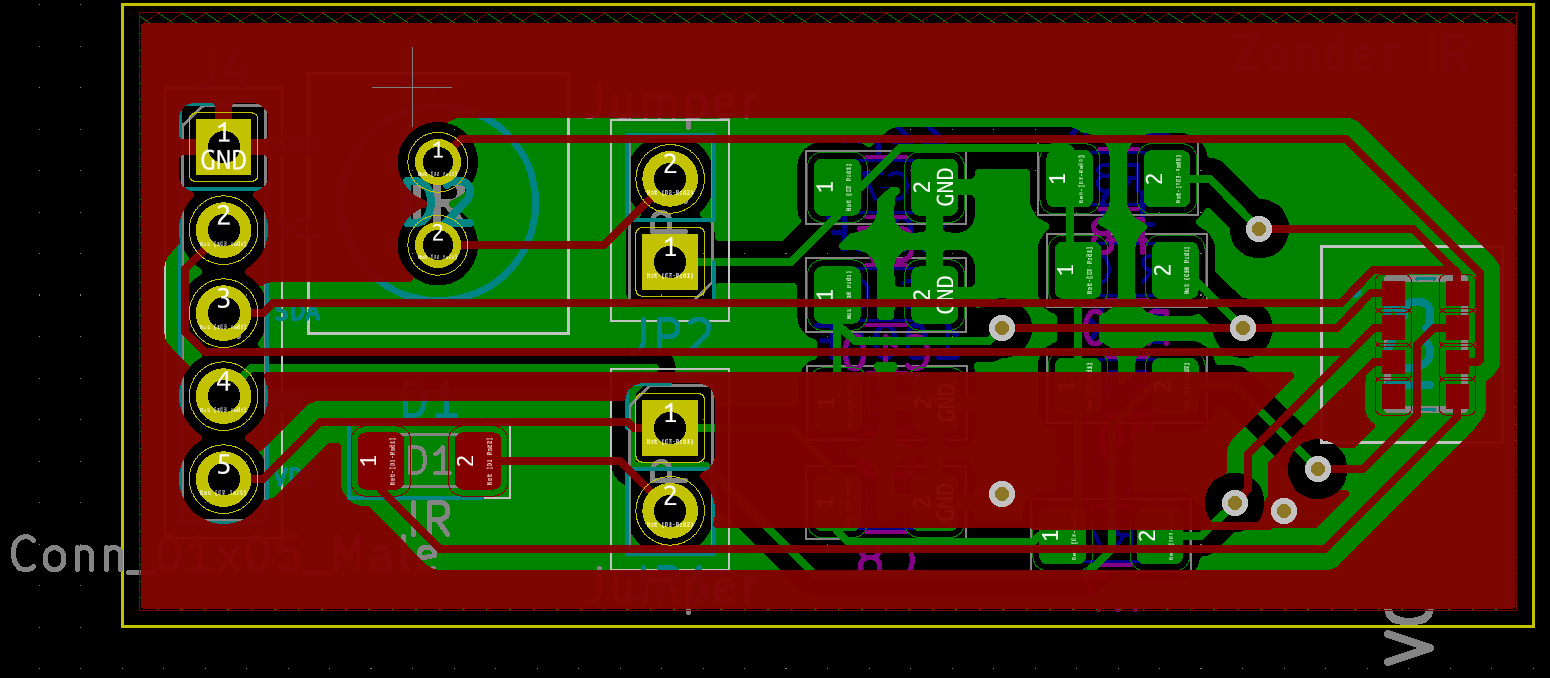
Ik heb voor elk van de 4 sensoren een aparte PCB gemaakt. Deze zullen allemaal verwerkt worden in een behuizing. Door te werken met aparte PCB’s kunnen de sensoren elk op een verschillende plaats worden ingewerkt zodat een invallende brief zeker gedetecteerd zal worden.

## 3.3 bestukkingsplan

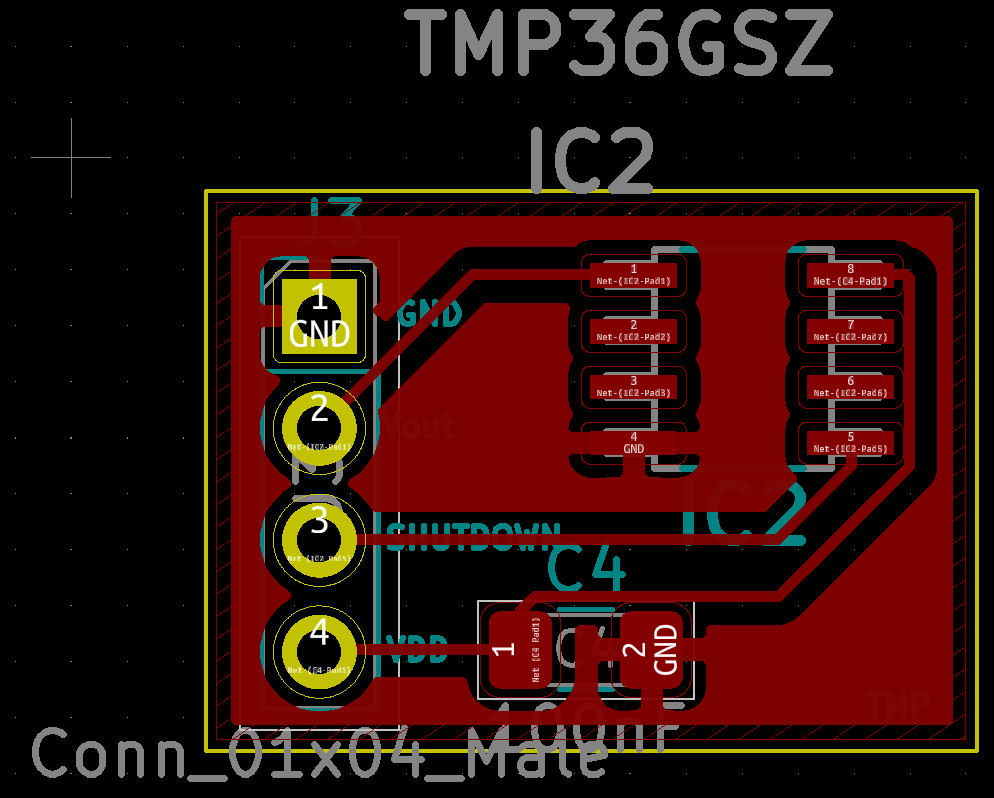
### VCNL3030



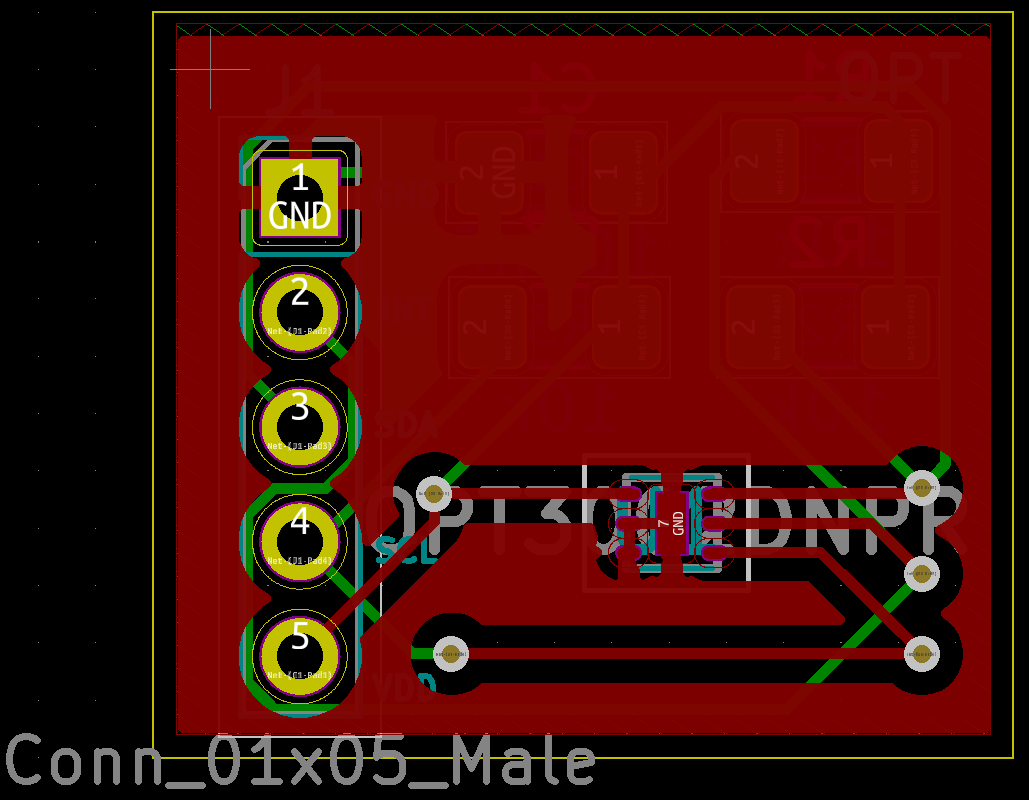
### VCNL4035



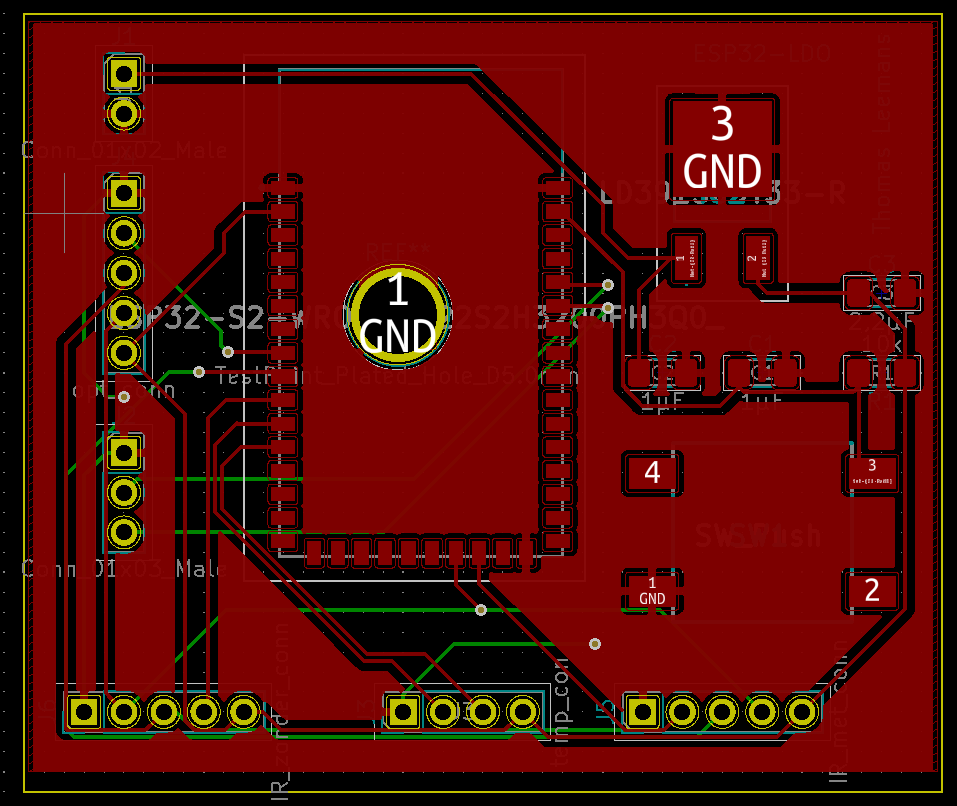
### TMP36GSZ



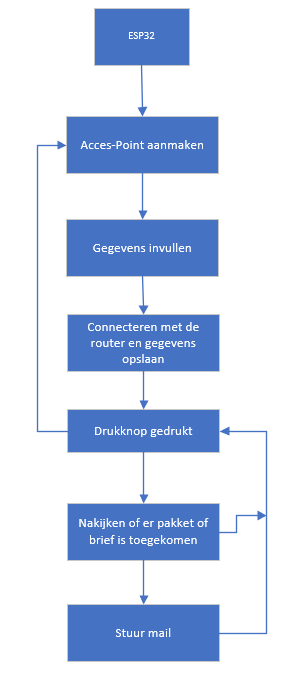
### OPT3002



### ESP32 + LDO



## 3.4 Blokdiagram software



## 3.5 flowchart

## 3.6 Software

## 3.7 kostenraming

## 3.8 bedieningshandleiding

## 3.9 metingen

## Besluit

## Bronnenlijst