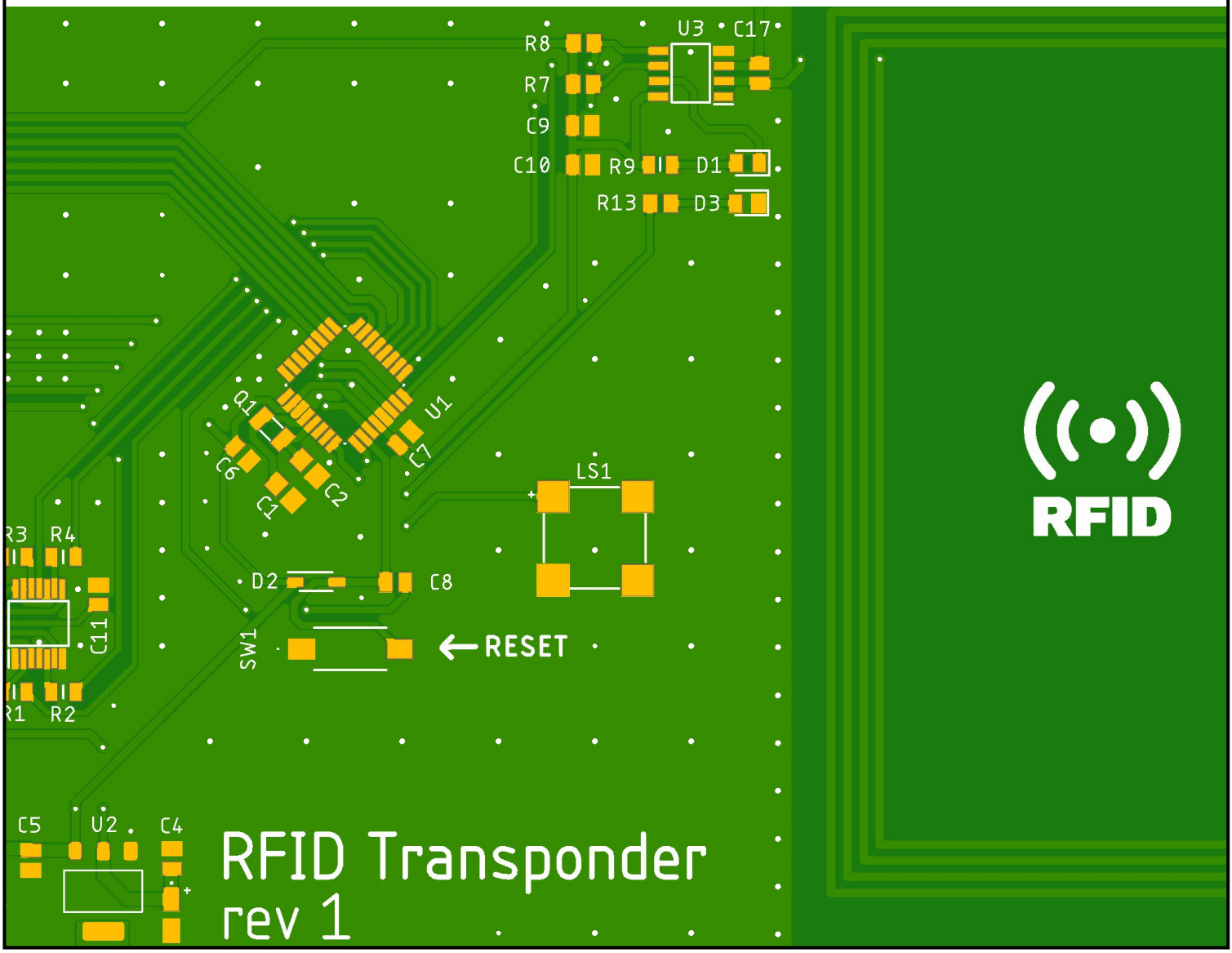
RFID Transponder

Ruben Schoonbaert



Inhoudsopgave

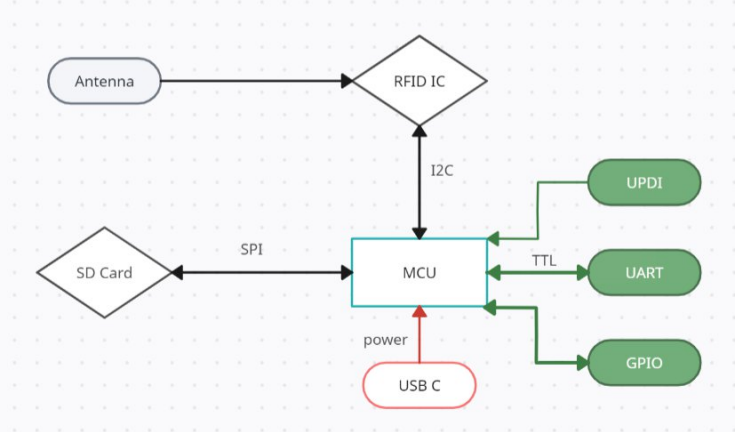
1) *Inleiding*

De aanleiding voor de keuze van dit project was de opgave die ik gekregen heb voor mijn bachelorsproef. Hierin willen ze dat ik een systeem onderzoek om adhv RFID stickers een digitaal volgsteem te creëeren voor hun retourlijn. Daarom dat ik voor het project geavanceerd printontwerp me al eens wilde verdiepen in RFID.

Het originele projectvoorstel hield de volgende parameters in:

- *RFID spoel op de printplaat.*
- *RFID lezer IC met bijhorende componenten*
- *Micro SD kaart houder om data naar weg te schrijven*
- *Toepasselijke MCU*
- *Connectoren voor UPDI en UART, eventueel GPIO break-out*
- *USB C poort en toepasselijke IC & componenten voor stroomverdeling*

1.1) Bespreking blokschema



De Schakeling wordt bestuurd door een MCU van Atmega deze IC bestuurd de andere IC's in de schakeling.

De RFID IC zal worden bestuurd via het I2C dataprotocol. Deze zal een signaal ontvangen uit de Antenne die op de printplaat zal geëtst zijn. Als het een RFID Signaal binnenkrijgt stuurt de IC het terug naar de MCU.

Figuur 1: Blokschema

De MCU zal dit signaal omzetten van een binair signaal naar een leesbare string. Hierna zal het deze string sturen naar de Micro SD kaart adhv het SPI dataprotocol. (Hierbij zal het ofwel een nieuw bestand creëeren of een bestaand bestand bewerken).

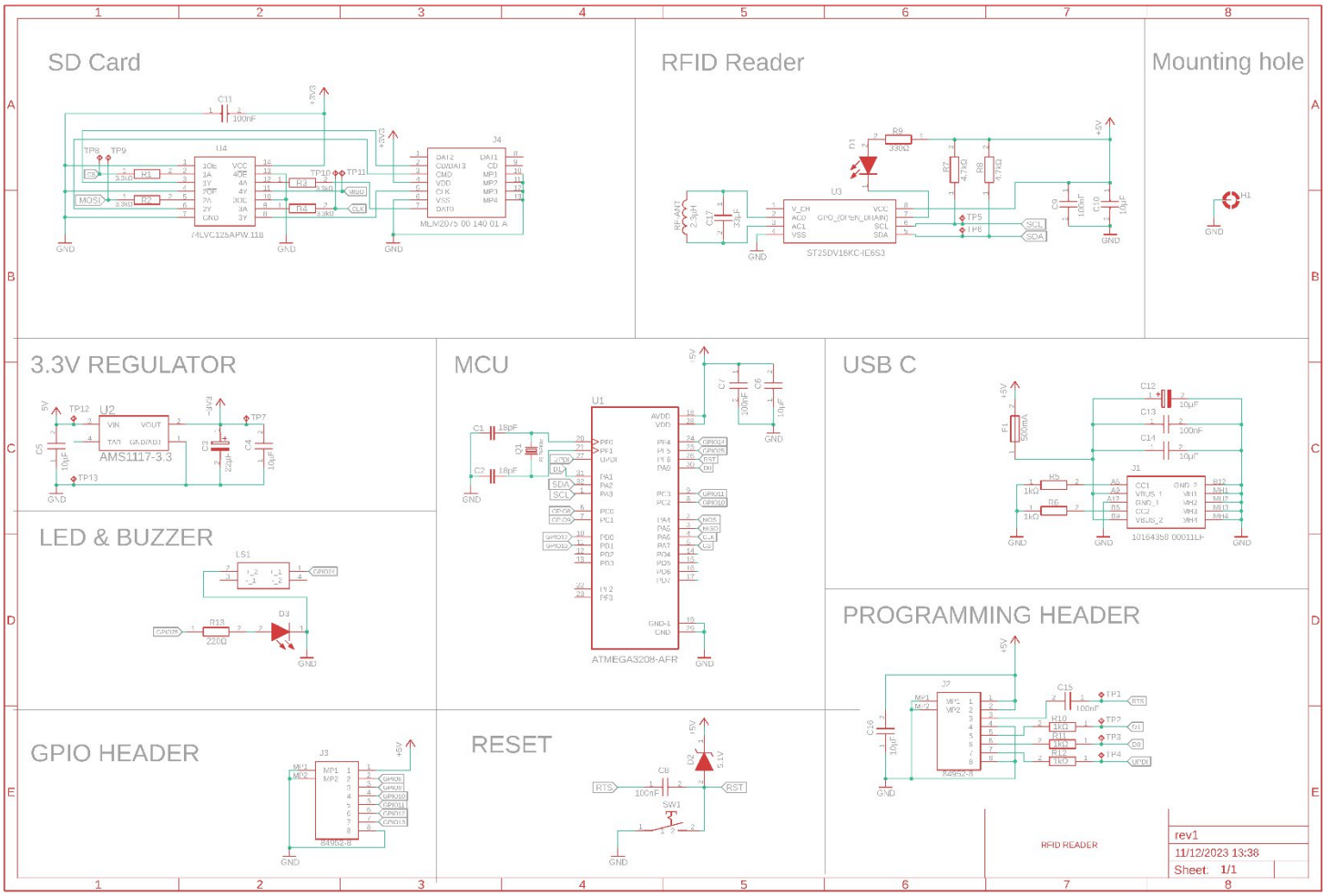
Via een geïntegreerde UPDI poort kan de bootloader geprogrammeerd worden op de MCU. Zo kan het geprogrammeerd worden met een gekende IDE (bv. Platformio of Arduino).

De UART poort kan de Microcontroller programmeren adhv een CH340C(of vergelijkbare IC) – USB schakeling.

Eventuele uitbreidingen of implementaties van het project kunnen worden verbonden met de schakeling door gebruik van de GPIO poort.

Heel deze schakeling wordt gevoed door middel van een standaard USB-C poort.

2) Bespreking Schema

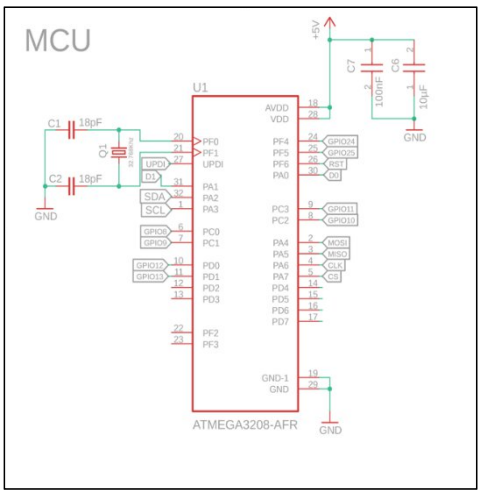


Figuur 2: Algemeen Schema

De schakeling is opgedeeld in ongeveer 7 verschillende sub-schakelingen. De complete schakeling telt zo een ~60 componentenen.

Hierin zitten er 3 verschillende IC's en 3 connectoren. Het schema is voorzien van een FFC connector voor te programmeren en een FFC connector die aangesloten is op verschillende GPIO pinnen om te dienen als een verbindingspunt voor eventuele uitbreidingen.

2.2) Bespreking MCU schema



Figuur 3: MCU-Schema

De gekozen MCU voor dit project is de AtmegaXX08 reeks van microchip technology (Specifiek de Atmega3208). Deze IC is dus vervangbaar met andere IC's van dezelfde reeks & package.

Alhoewel de keuze voor een lagere flash capaciteit (Bv. de 1608) de werking van eventuele library's zou kunnen hinderen.

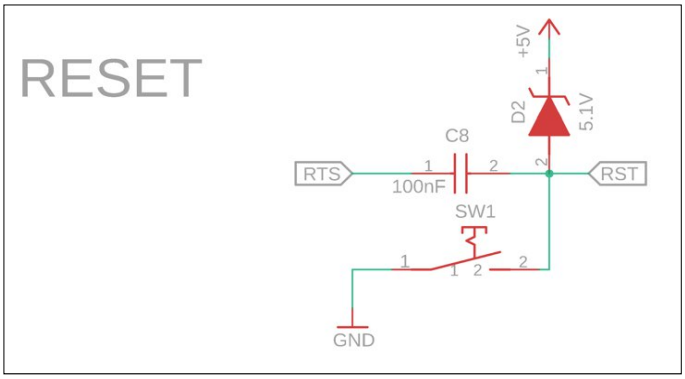
De Atmega3208 chip heeft 32KB flash, 4KB SRAM, I2C en SPI bussen, en 32 pinnen (waarvan 25 GPIO).(bron 1:p.2-4)

De logica van de chip is 5 V dus er moet gebruik gemaakt worden van logica omvormers om met IC's die op 3,3 V werken te interfacen.

Alhoewel de Atmegaxx08 reeks IC's een interne oscillator hebben heb ik geopteerd om toch de mogelijkheid te voorzien om een externe 32,768kHz kristaloscillator toe te voegen (Q1). Deze oscillator is voor meeste toepassingen redundant, maar het toevoegen van deze externe oscillator zorgt voor een stabielere werkingsconditie en een lager stroomverbruik dan met het gebruik van de interne oscillator.(bron 2)

Om de ingebouwde RSTCTRL te gebruiken moet de externe $\overline{RESET}$ pin getrokken worden naar de massa.(bron 1: p. 108)

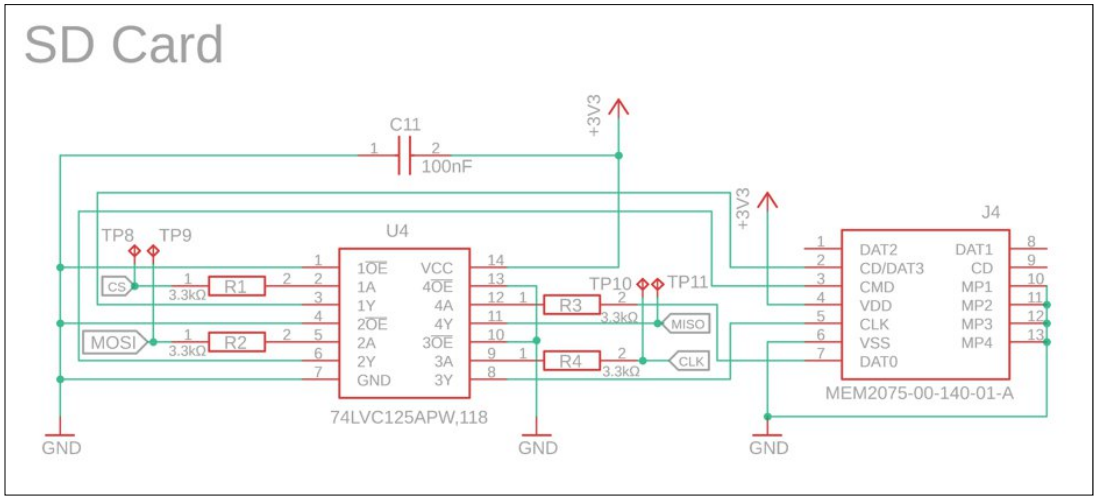
In dit het reset-schema kan dit gedaan worden met behulp van een manuele knop (SW1). Maar dit kan ook softwarematig gedaan worden door de RTS pin. Die geeft een signaal, deze gaat door de condensator C8 en geeft een korte lage puls op de RST pin



Figuur 4: Reset-Schema

De zenerdiode (D2) fungeert hier als voltage klem om de reset pin te beschermen tegen overstroom. Door dit schema kan de MCU robuust manueel en automatisch gereset worden.

2.3) Bespreking SD-Schema

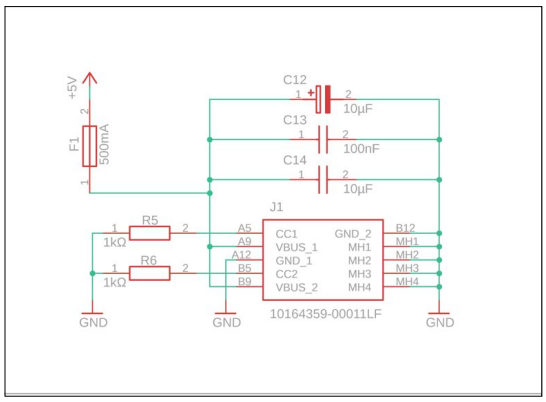


Figuur 5: SD-Card Schema

Zoals eerder vermeld is de logica van de MCU 5V, dit zorgt er voor dat het niet rechstreeks kan interfacen met de Micro SD-Kaart via SPI, omdat de Micro SD-Kaart werkt met een interne logica van 3,3V[(bron 3)]

Daarom wordt er in dit schema gebruikt gemaakt van de “74LVC125 IC” (U4). Deze chip heeft 4 “bus buffer gates”, die geschikt zijn om directioneel de logica om te zetten van 3,3V naar 5V en omgekeerd[(bron 4)]. Zo kan de MCU communiceren met de Micro SD-Kaart via SPI zonder dat de SD-Kaart schade oploopt.

2.4) Bespreking voeding schema

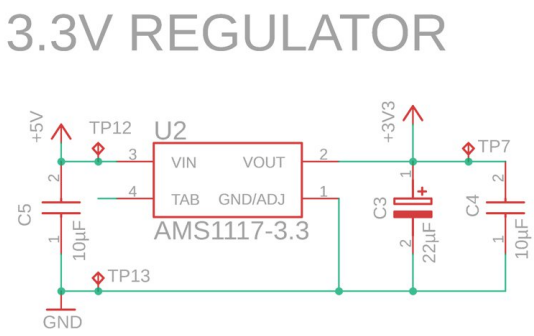


De schakeling wordt gevoed door een standaard 6 pin USB-C poort van 5V. Om de USB poort succesvol in te stellen moeten de “CC” pinnen verbonden worden aan de massa met behulp van 1kΩ weerstanden.

De voeding wordt ontkoppeld door 2 keramische condensatoren en 1 tantaal elektrolytische condensator. Om voltagepieken weg te werken en een stabielere spannings voeding te garanderen.

Figuur 6: USB-C Schakeling

Als bescherming voor de voeding tegen eventuele kortsluitingen of overstromen is deze schakeling voorzien van een PPTC-zekering van 500mA. Deze zekering is zelf herstellend en moet dus niet vervanger worden in het geval van een kortsluiting.



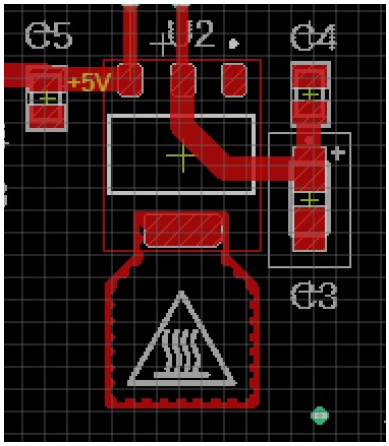
Figuur 7: 3,3V regulator schema

Om de schakeling te voorzien van een 3,3V voeding wordt er gebruik gemaakt van een AMS1117 lineaire voltage regulator (U2).

Deze voltage regulator wordt voorzien in een SOT-2223 package.

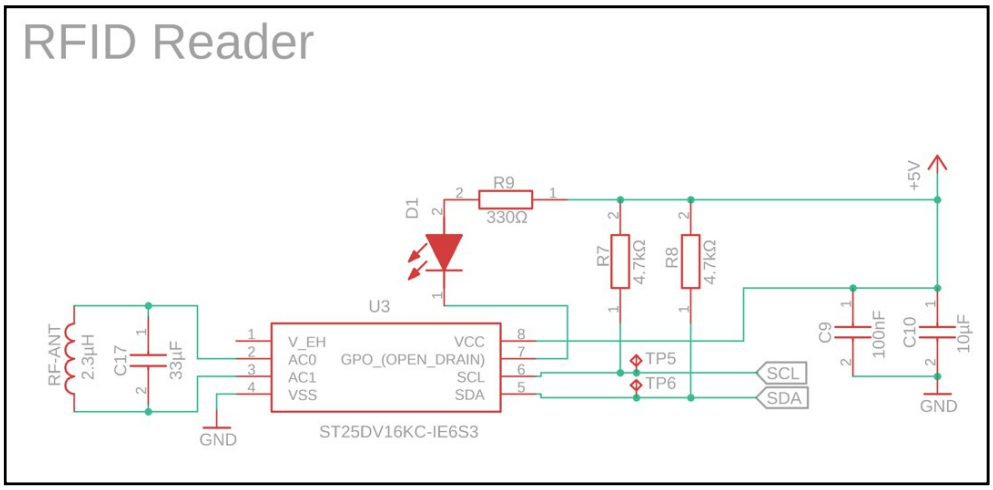
De spanning uit de regulator wordt ontkoppeld door een elektrolytische tantaal condensator van 22µF (C3) en een keramische condensator van 10µF (C4) zoals beschreven in de datasheet.(bron 5)

Omdat lineaire voltage regulators wel wat warmte kunnen produceren is op de anode van de regulator een koelvlak ingebouwd op de print voorzien. Zo kan eventuele extra warmte sneller gedissipeerd worden. En wordt deze warmte weg gehouden van de temperatuur-gevoelige elektrolytische condensator (C3).



Figuur 8: Printontwerp voltage regulator

2.4) Bespreking RFID schema



Figuur 9: Bespreking RFID Schema

Het RFID schema functioneert rond de ST25D-reeks chip (ST25DV04KC) (U3), deze IC is een dynamische NFC/RFID tag IC met een 16kb EEPROM met een I2C bus.

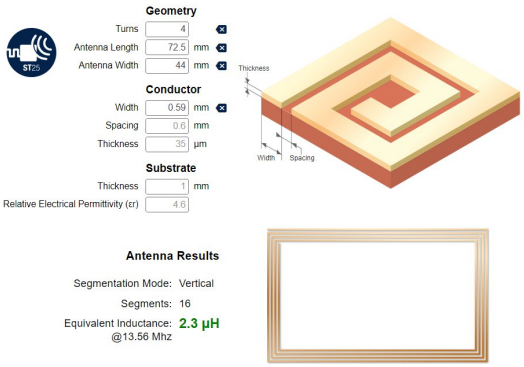
Het nadeel aan deze chip is dat ze niet de mogelijkheid heeft om andere niet-gevoede-RFID tags te lezen (Bv. Een RFID sticker). Er kan echter wel data op de chip geschreven worden met behulp van een ander RFID toestel (Zoals een smartphone).

Het voordeel is wel dat er data kan geschreven worden naar de IC vanuit de microcontroller en dat deze data dan kan gelezen worden door een andere Reader. Hierdoor werkt het toestel als een soort transponder, het kan namelijk data “lezen” en schrijven.

De IC heeft 1 GPO pin die adhv. het aanpassen van het intern register ingesteld kan worden voor verschillende toepassingen.(bron 6) Hier hangt er een “open drain” Ledschakeling aan.

3) RFID Antenne

De ingebouwde spoel op de printplaat is ontworpen met gebruik van de eDesignsuite calculator gemaakt door ST-Engineering (De fabrikant van het RFID IC).



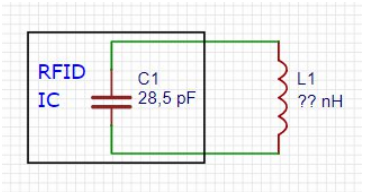
Figuur 10: eDesignsuite tool

Parallel met aansluitpinnen voor deze spoel is er een inwendige “tuning” condensator voorzien van 28,5pF in het IC(bron 6), dit vormt een LC-schakeling. Door het feit dat we de formule om de resonantiefrequentie van een LC-keten te berekenen kennen:

$$Fr = \frac{1}{\sqrt{2 \cdot \pi \cdot LC}}$$

En dat we weten dat de resonantiefrequentie 13,56MHz moet bekomen, kunnen we hieruit de gewenste antenne-inductie berekenen.

Echter is het niet altijd mogelijk om met enkel deze parameter en het gebruikt van de eDesignsuite tool goede waardes uit te komen, in dit geval is de gewenste waarde voor de spoelwaarde 4,8µH.

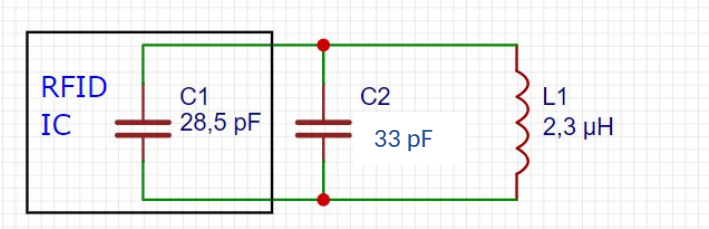


$$Fr = \frac{1}{\sqrt{2 \cdot \pi \cdot LC}} <==> L = \frac{1}{4\pi^2 \cdot f^2 \cdot C}$$

$$L = \frac{1}{4\pi^2 \cdot 13{,}56Mhz^2 \cdot 28{,}5pF} = 4{,}8\mu H$$

Figuur 11: LC Condensator berekening

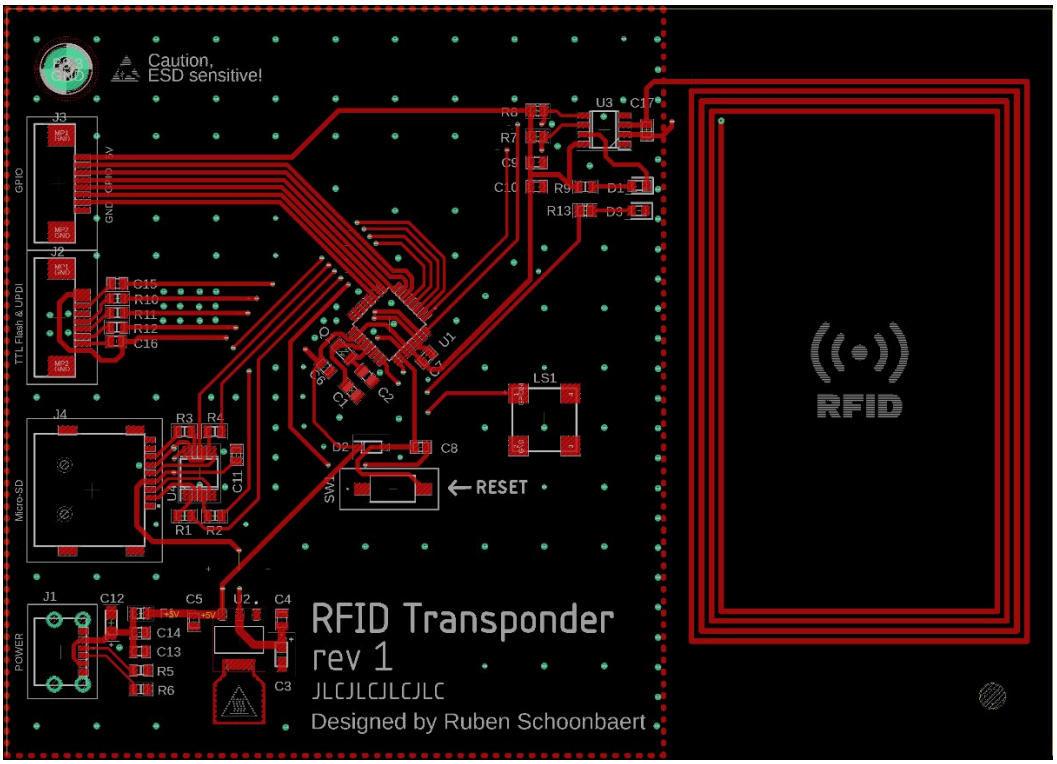
Deze waarde is te hoog voor deftig te kunnen maken op deze printplaat. Om dit op te lossen kan er een condensator parallel worden geplaatst met de spoel.(bron 7) Zo zal de gewenste spoelwaarde dalen. Door met deze waardes te werken is er op dit schema een spoel van 2,3µH gekozen met een parallelcondensator van 33pF (C17).



Figuur 12: Bekomen antenneschakeling.

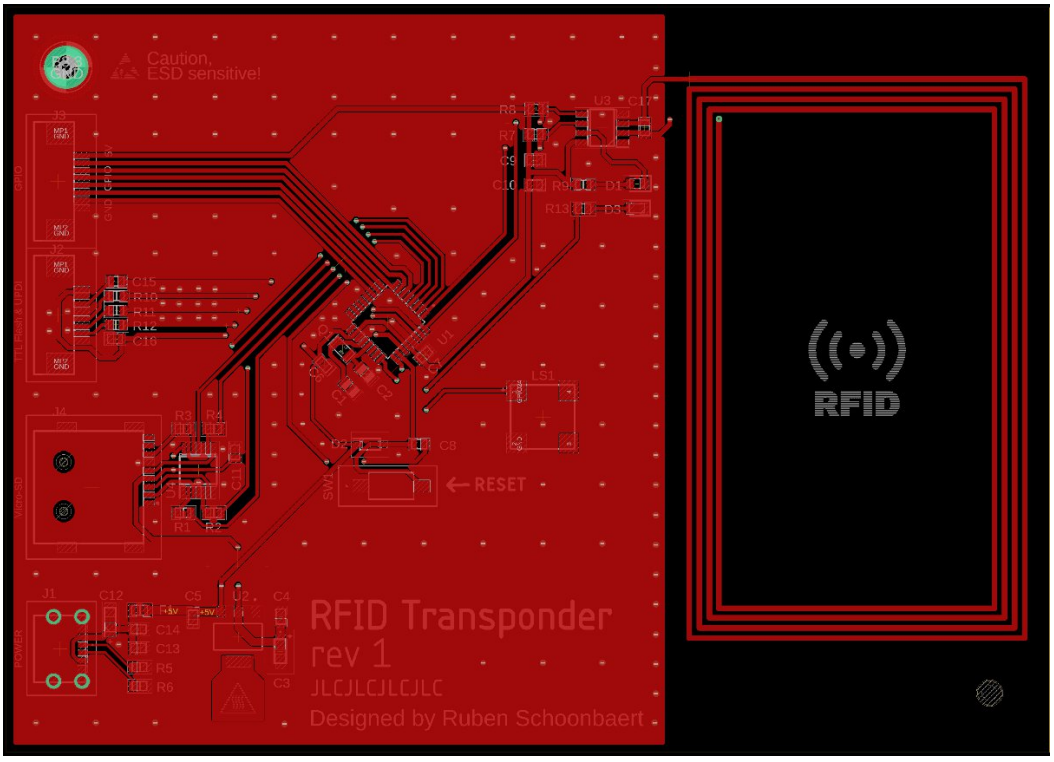
4) Koperlaag top

4.1) Zonder polygons:



Figuur 13: Koperlaag-top zonder polygons

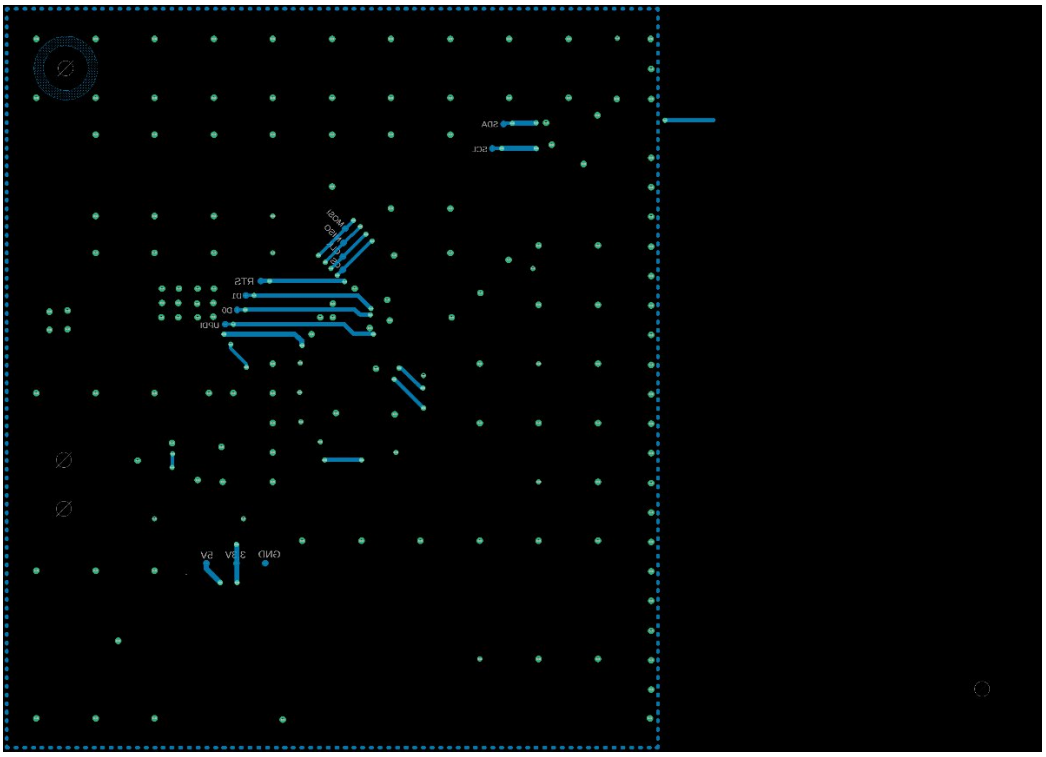
4.2) Met Polygons:



Figuur 14: Koperlaag top met polygons

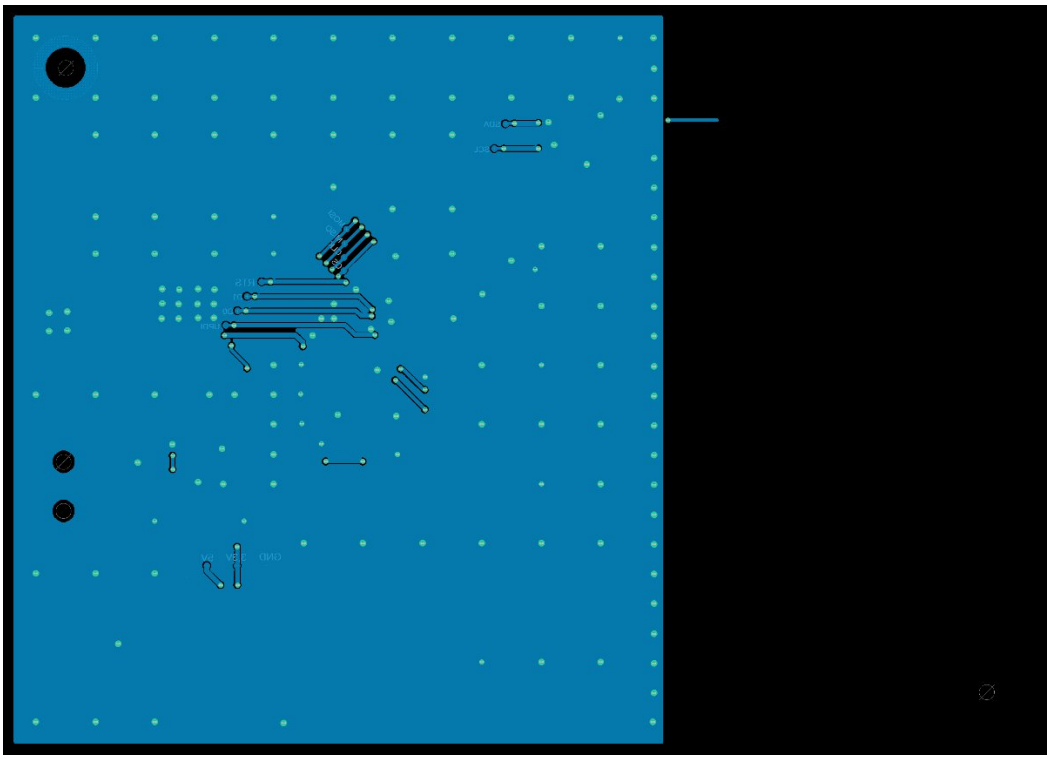
5) Koperlaag bottom

5.1) Zonder polygons



Figuur 15: Koperlaag-bottom zonder polygons

5.2) Met polygons



Figuur 16: Koperlaag-bottom met polygons

6) Test & uitbreidingsmogelijkheden

Om een mogelijkheid te voorzien om de schakeling te “debuggen” in geval van een probleem met de schakeling. En om eventuele verdere uitbreidingen op de print te ondersteunen is de schakeling voorzien van test pads.

Figuur 17: Test / uitbreiding pads

Er zijn test/uitbreidings mogelijkheden voorzien op:

- De 5 en 3,3 volt dc-voedingen
- De seriële datapoort (RX & TX)
- De reset & UPDI pinnen
- De I2C bus
- De SPI bus

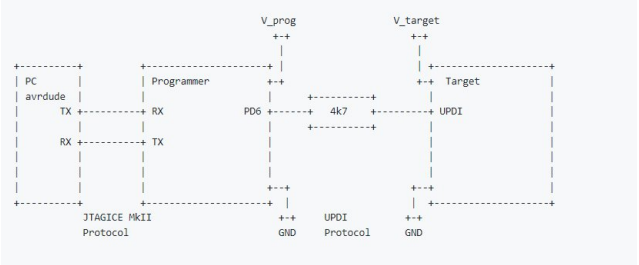
Al deze pads zitten op de onderzijde van de print en zijn soldeerdbaar.

7) Flash & UPDI Programmeren

7.1 UPDI programmeren

Voordat de MCU kan geprogrammeerd worden met een IDE (Arduino of platformio) via flash, moet er eerst een bootloader op de MCU worden gezet. Hiervoor moeten we gebruik maken van de jtag2updi firmware(bron 8) . Deze code is verantwoordelijk voor het initialiseren van de bootloader op de microcontroller.

Om dit te doen werken, moet de jtag2updi code op een standaard arduino worden gezet. Hierna moet de arduino verbonden worden met de UPDI pin van de PCB volgens de deze schakeling(figuur 18):



Figuur 18: UPDI circuit.

*De 4,7kΩ weerstand is voorzien op de print.

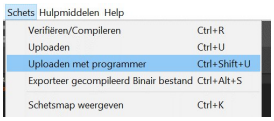
Nu kan de bootloader op de Atmega3208 worden gezet met de volgende stappen:

- Installeer de MegaCoreX core (bron9)
- Selecteer het juiste bord (“MegaCoreX -> ATmega3208”)
- Ga naar " Hulpmiddelen " en stel de volgende bord-instellingen in:
  - Clock: "Internal 20 MHz"
  - Pinout: "32 pin standard"
  - Reset pin: "Reset"
  - Bootloader: "Optiboot (UART0 default pins)"
  - Programmer: "jtag2updi(megaTinyCore)"
- Onder “hulpmiddelen” duw op “burn bootloader”

Deze UPDI-schakeling zit in een ge-3D-printe behuizing en kan verbonden worden met de PCB via de FFC connector J2.

Figuur 19: UPDI-programeer toestel

Via deze schakeling kan er ook code geüpload worden naar de atmega3208 via de “uploaden met programmer” functie.



Er kan dan wel geen gebruik gemaakt worden van de seriële poort.

7.2) Flash programmeren

Nu de print via flash kan worden geprogrammeerd wordt er gebruik gemaakt van een CH340C programmeer-bord die speciaal voor deze print ontworpen is. Ik heb de keuze gemaakt om dit extern van de voornaamste schakeling te maken omdat ik graag zelf een verwisselbaar programmeer-bord wilde hebben die ik kan implementeren op andere komende projecten.

Figuur 20: CH340C programeer bord

Dit bord heeft de mogelijkheid om 5 en 3,3V logica te gebruiken. Het wordt verbonden met de print aan de hand van een FFC kabel. En kan verbonden worden met een computer door middel van een micro usb kabel.

Om dit bord te kunnen gebruiken moet de host computer voorzien van de CH340C driver. Anders is er de mogelijkheid dat op sommige besturingssystemen de CH340C chip niet wordt gedetecteerd.

8) Componentenlijst

Part	Value	Description	Package	Price (€)	RoHS compliant
C1	1←þF	Capacitor	C0←05	?	?
C2	1←þF	Capacitor	C0←05	?	?
C3	22µF	Tantalum Capacitor	CAPC2012	0,15↵	1
C4	10µF	Capacitor	C0←05	0,034	1
C5	10µF	Capacitor	C0←05	0,034	1
C6	10µF	Capacitor	C0←05	0,034	1
C7	100nF	Capacitor	C0←05	0,027	2
C↵	100nF	Capacitor	C0←05	0,027	2
C9	100nF	Capacitor	C0←05	0,027	2
C10	10µF	Capacitor	C0←05	0,15↵	1
C11	100nF	Capacitor	C0←05	0,027	2
C12	10µF	Tantalum Capacitor	CAPC2012	0,15↵	1
C13	100nF	Capacitor	C0←05	0,027	2
C14	10µF	Capacitor	C0←05	0,15↵	1
C15	100nF	Capacitor	C0←05	0,027	2
C16	10µF	Capacitor	C0←05	0,15↵	1
C17	10µF	Capacitor	C0←05	0,15↵	1
D1		LED	D0←05	?	?
D2	MMSZ5231B-7-F	Diode	D0←05	0,0↵	1
D3		LED	D0←05	?	?
F1	500mA	PPTC Fuse	FUSC2114X125N	0,094	1
J1	USB C	connector	1016435900011LF	0,3←↵	1
J2		FFC-connector	←4952↵	?	?
J3		FFC-connector	←4952←↵	?	?
J4		SD-card slot	MEM20750014001A	?	?
LS1	4kHz	Buzzer	CMT-←540S-SMT-TR	2,92	1
Q1	32.76←Khz	Kristal oscillator	3.2X1.5	0	1
R1	3,3kΩ	Resistor	R0←05	0,512	2
R2	3,3kΩ	Resistor	R0←05	0,014	2
R3	3,3kΩ	Resistor	R0←05	0,014	2
R4	3,3kΩ	Resistor	R0←05	0,014	2
R5	10kΩ	Resistor	R0←05	0,009	2
R6	10kΩ	Resistor	R0←05	0,009	2
R7	4,7kΩ	Resistor	R0←05	0,014	2
R↵	4,7kΩ	Resistor	R0←05	0,014	2
R9	330Ω	Resistor	R0←05	?	?
R10	1kΩ	Resistor	R0←05	0,017	1
R11	1kΩ	Resistor	R0←05	0,017	1
R12	1kΩ	Resistor	R0←05	0,017	1
R13	220Ω	Resistor	R0←05	?	?
SW1		TACTWITCH	PTS636SM50SMTRLFS	?	?
U1	ATMEGA320←AFR	MCU IC	QFP←0P900X900X120-32N	1,53	1
U2	AMS1117-3,3V	Voltage regulator	SOT229P700X1←0-4N	?	?
U3	ST25DV16KC-IE6S3	RFID IC	SOIC127P600X175-←N	1,03	1
U4	74LVC125APW,11↵	IC	SOP65P640X110-14N	0,2←7	1
			total =	←,192	

RoHs
1 = RoHS compliant
2 = RoHS compliant through exemption

Alle '?' zijn onderdelen die ik al in bezit had waarvan ik de waarden niet terugvond

9) Conclusie

9.1) Algemeen

Over het algemeen vind ik dit project zeker geslaagd, de werking van de print is exact zoals verwacht en is zeer robuust. Elk onderdeel die beschreven staat in het origineel projectvoorstel is functioneel. Er zijn geen defecten in het printontwerp en de layout van de componenten zorgde niet voor problemen.

9.2) Wat kan beter?

1) De USB C connector: iets waar ik tijdens het ontwerpproces niet had bij stilgestaan is dat de dikte van de printplaat zelf de werking van de USB-C kabel zou kunnen hinderen. De 3mm rand voor de USB-C connector zorgt er voor dat de kabel niet zomaar kan gebruikt worden. Om dit op te lossen heb ik een USB kabel ietwat moeten verminken om ze te doen passen.

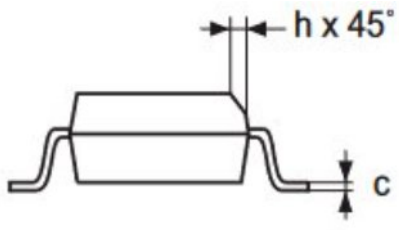
Figuur 21: USB-C connector

2) De Test pads: Moest ik een volgende revisie maken van dit ontwerp zou ik de test pads nog eens opnieuw bekijken. De plaatsing van de pads is in de praktijk niet zo handig. En de grootte van de pads is bij nader inzien ook wat te klein, dit zorgt er voor dat het moeilijk is om ze te “proben” en er kabels op te solderen.

9.3) Waar heb ik me aan mispakt?

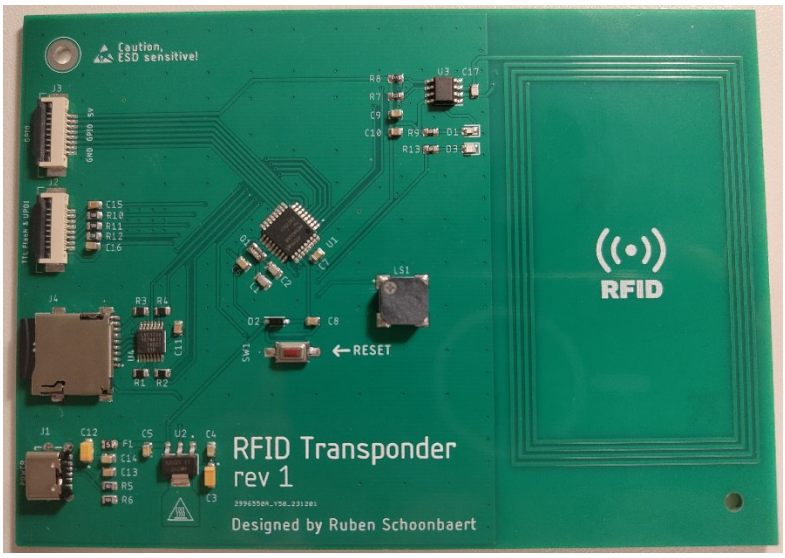
Tijdens het eerste soldeerproces had ik (foutief) de conclusie getrokken dat de polariteit van de ST25DV chip te zien was aan de hand van de oriëntering van de tekst op de package en een notch op de bovenkant van de package. Dit was omdat er hier in de datasheet quasi niets over is terug te vinden. Dit bleek dus niet te kloppen en zorgde voor een kortsluiting van de IC.

Na verder onderzoek bleek dat de polariteit van chip te zien valt door het feit dat 1 kant van de chip een vellingskant heeft van 45°. Deze kant heeft de pinnen (1-4).



Gelukkig heeft de PPTC zekering (F1) de print en de chip beschermd van schade op te lopen. En werkten al de componenten nog na het omdraaien van deze chip

Figuur 22: beschrijving polariteit.



Figuur 23: Bestukte print

10) Afkortingen Lijst

- ***EEPROM*** *- Electrically erasable programmable read-only memory*
- ***FFC*** *– Flat flex cable*
- **GPIO –** *General Purpose Input/Output*
- ***GPO*** *– General Purpose Output*
- **I2C –** *Inter IC*
- **IC –** *Integrated Circuit*
- ***IDE*** - integrated development environment
- **MCU –** *Microcontroller Unit*
- **NFC –** *Near field communication*
- **PPTC -** *Polymeric Positive Temperature Coefficient*
- **RFID –** *Radio Frequency Identification*
- ***RoHS –*** *Restriction of Hazardous Materials*
- **RST -** *Reset*
- **RSTCTRL –** *Reset Controller*
- **RTS -** *Ready To Send*
- **SD –** *Secure Digital*
- ***SOT –*** *Small outline transistor*
- ***SOP -*** *Small outline package*
- **SPI –** *Serial Peripheral Interface*
- **SRAM –** *Static Random Accesible Memory*
- ***UPDI*** *- Unified Program and Debug Interface*

11) Bronnen lijst

1. https://www.microchip.com/en-us/product/atmega3208
2. https://www.ti.com/video/6313368697112
3. https://www.engineersgarage.com/interfacing-sd-card-with-avr-microcontroller-part-38
4. https://www.ti.com/product/SN74LVC125A
5. http://www.advanced-monolithic.com/pdf/ds1117.pdf
6. https://eu.mouser.com/datasheet/2/389/st25dv04kc-2450072.pdf
7. https://www.st.com/resource/en/application_note/an2866-how-to-design-a-1356-mhz-customized-antenna-for-st25-nfc--rfid-tags-stmicroelectronics.pdf
8. https://github.com/ElTangas/jtag2updi
9. https://github.com/MCUdude/MegaCoreX

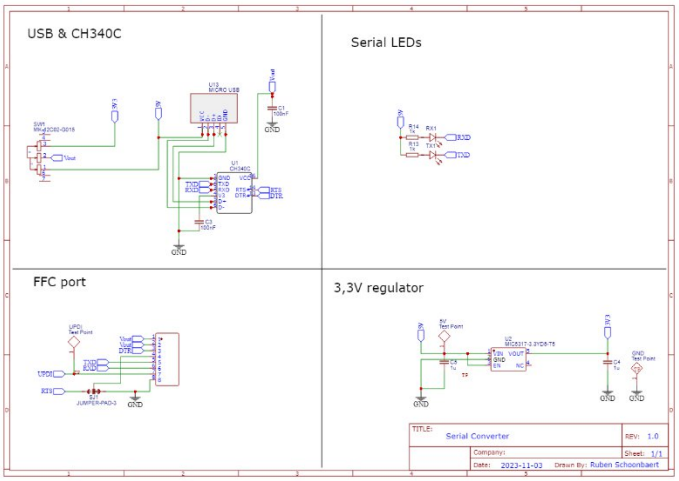
12) Extra

12.1 Code)

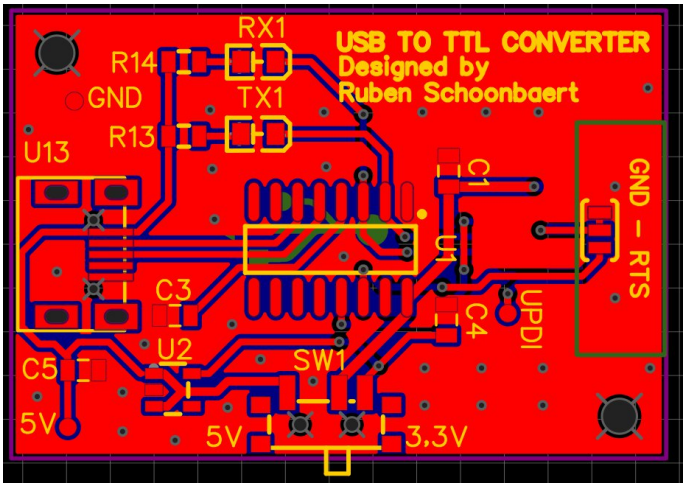
```
#include "ST25DVSensor.h"
#include <SPI.h>
#include <SD.h>

#if defined(ARDUINO_B_L4S5I_IOT01A)
  #define GPO_PIN PE4
  #define LPD_PIN PE2
  #define SDA_PIN PB11
  #define SCL_PIN PB10
  #define WireNFC MyWire
  TwoWire MyWire(SDA_PIN, SCL_PIN);
  ST25DV st25dv(12, -1, &MyWire);
#else
  #define DEV_I2C         Wire
  ST25DV st25dv(12, -1, &DEV_I2C);
#endif
String last_read;
bool cleared = false;
const int chipSelect = 7;

void beep(int pitch, int duration, int loudness){
  tone(24, pitch, loudness);
  delay(duration);
  noTone(24);
  delay(10);
}

void clearFile(const char* filename) {
  if (SD.begin(chipSelect)) {
    // Open the file in write mode without appending
    File myFile = SD.open(filename, FILE_WRITE | O_TRUNC);

    if (myFile) {
      myFile.close();
      Serial.println("File cleared!");
    }
  }
}

void printLine(){
  for(int i = 0; i < 50; i++){
    Serial.print("-");
  }
  Serial.println();
}

void printStart(){
  Serial.println(F("                                                "));
  Serial.println(F("                                                "));
  Serial.println(F("                                                "));
  Serial.println(F("                                                "));
  Serial.println(F("                                                "));
  Serial.println(F("                                                "));
  printLine();
  Serial.println(F("By Ruben Schoonbaert"));
  Serial.println(F("Written 29/12/2023"));
  printLine();
  delay(1000);
  Serial.println(F("COMMANDS:"));
  Serial.println(F("$printFile -> prints all records on the SD card"));
  Serial.println(F("$clearFile -> clears all records on the SD card"));
  Serial.println(F("$clearRFID -> removes last entry from RFID tag"));
  Serial.println(F("$addRecord <record> -> Adds new record to SD & RFID Tag"));
  Serial.println(F("$testBuzzer -> Tests integrated piezo buzzer"));
  Serial.println(F("$testLED -> Tests integrated LED"));
  printLine();
}

void printFile(const char* filename) {
  if (SD.begin(chipSelect)) {
    // Open the file in read mode
    File myFile = SD.open(filename, FILE_READ);
    printLine();
    Serial.println(F("    Printing SD-Card content    "));
    printLine();
    if (myFile) {
      while (myFile.available()) {
        Serial.write(myFile.read());
      }

      myFile.close();
      printLine();
    } else {
      Serial.println(F("Error opening file for reading."));
      digitalWrite(25, HIGH);
    }
  }
}

bool isDuplicate(const char* filename, const String& newString) {
  File file = SD.open(filename);
  if (file) {
    while (file.available()) {
      String line = file.readStringUntil('\n');
      line.trim();
      if (line.equals(newString)) {
        file.close();
        return true; // Duplicate found
      }
    }
    file.close();
  }
  return false; // No duplicate found
}

void addRecord(String record){
  const char uri_write_protocol[] = URI_ID_0x01_STRING;
  if(st25dv.writeURI(uri_write_protocol, record, "")) {
    Serial.println(F("Write to RFID Tag failed!"));
    digitalWrite(25, HIGH);
    while(1);
  } else {
    if(record != "!ignore"){
    Serial.print(F("Written : \""));
    Serial.print(record);
    Serial.print(F("\" to RFID tag succesfully"));
    Serial.println();
    }
  }
}
```

```
void setup() {
  Serial.begin(9600);
  pinMode(24, OUTPUT);
  pinMode(25, OUTPUT);
  digitalWrite(25, HIGH);
  printStart();
  if(st25dv.begin() == 0) {
    Serial.println(F("System Init done!"));
    beep(2000, 100, 50);
  } else {
    Serial.println("System Init failed!");
    digitalWrite(25, HIGH);
    while(1);
  }
  delay(500);
  digitalWrite(25, LOW);
}

void loop() {
  String uri_read;
   if (Serial.available() > 0) {
    String command = Serial.readStringUntil('\n');
    command.trim();

    if (command.equals("$clearFile")) {
      clearFile("records.txt");
      cleared = true;
      beep(2000, 100, 50);
      return;
    } else if (command.equals("$printFile")){
       beep(2000, 100, 50);
      printFile("records.txt");
    } else if (command.startsWith("$addRecord ")) {
      // Extract the record from the command
      String recordToAdd = command.substring(11);
      // Add the record to the file
      addRecord(recordToAdd);
    } else if (command.equals("$clearRFID")){
       addRecord("!ignore");
       Serial.println(F("Cleared RFID"));
         beep(2000, 100, 50);
    } else if (command.equals("$testLED")){
      bool state = false;
      for(int i = 0; i<10; i++){
        digitalWrite(25, state);
        state = !state;
        delay(100);
      }
    }
     else if (command.equals("$testBuzzer")){
     int melody[] = {262, 330, 392, 523, 392, 330, 262, 196, 262};
     for (int i = 0; i < sizeof(melody) / sizeof(melody[0]); i++) {
      delay(10);
      beep(melody[i], 200, 100);
     }

    } else {
      Serial.println("unrecognised command!");
      beep(500, 500, 500);
    }
  }

  if(st25dv.readURI(&uri_read)) {
    Serial.println("Read failed!");
    digitalWrite(25, HIGH);
    while(1);
  }
  if(uri_read != last_read){
     if (SD.begin(chipSelect)) {
      if(uri_read.startsWith("www.")){
        uri_read = uri_read.substring(4);
      }
      else if(uri_read.startsWith("http://www.")){
        uri_read = uri_read.substring(11);
      }
      else if(uri_read.startsWith("http://")){
        uri_read = uri_read.substring(7);
      }
      else if(uri_read.startsWith("https://")){
        uri_read = uri_read.substring(8);
      }
      if(uri_read != "!ignore"){
        if (!isDuplicate("records.txt", uri_read)) {
          if(!cleared){
            Serial.print(F("New record detected = "));
          } else {
            Serial.print(F("current RFID record = "));
          }
          cleared = false;
          Serial.print(uri_read);
          Serial.println();

          File myFile = SD.open("records.txt", FILE_WRITE);
          if (myFile) {
            myFile.println(uri_read);
            myFile.close();
            delay(100);
            beep(800, 100, 50);
            delay(10);
            beep(1000, 100, 50);
          }
         }
       }
      last_read = uri_read;
     }
  }
  delay(500);
  digitalWrite(25, LOW);
}
```

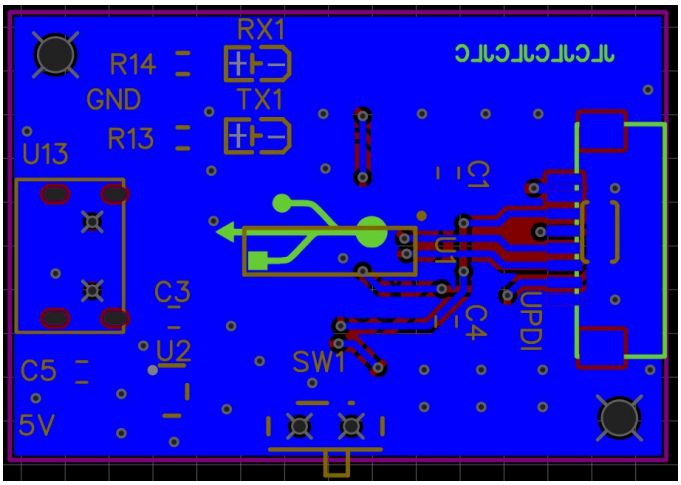
12.2 Programmeer bord)



Figuur 24: Programmeer bord schema



Figuur 25: Programmeer bord top



Figuur 26: Programmeer bord bottom