



**DE HOGESCHOOL
MET HET NETWERK**

www.pxl.be

PXL-TECH

Bachelor in de Elektronica-ICT

Multisensor

Vak: Engineering Project

Docent: Stukken B. & Vanrykel D.

Dries Brans
Tim Steensels



I. Inhoudsopgave

I.	Inhoudsopgave	1
II.	Inleiding	2
III.	Multisensor	3
IV.	Besluit	22
V.	Verbeteringen	22

II. Inleiding

De multisensor is een sensor bord die je kan plaatsen op de PSoC 4200(M). Het is Arduino compatibel. De multisensor heeft een aantal sensoren om data in te lezen. De communicatie word verwezenlijkt met SPI, I2C en UART. De data die je verzamelt kan ook verzonden worden over wifi met de ESP8266 module. De data kan eveneens op de SD-kaart worden gezet. Elke sensor die gebruikt word, word uitgelegd van zijn kunne en hoe je de sensor moet gebruiken aan de hand van een voorbeeldprogramma.

III. Multisensor

A. Componenten

Elk component wordt besproken wat het kan of doet. Daaronder vind je een voorbeeld code van hoe je het component aanstuurt. De volledige codes vind je in een .rar bestand op Github. Alle componenten die gebruikt zijn komen van Sparkfun. Daar kan je de originele schema's vinden. Verder vind je meer informatie op <https://github.com/Dries11400702/MultiSensor>.

Si7021 sensor

De Si7021 sensor is een sensor die **temperatuur** kan meten. Het is één van de 3 sensoren die temperatuur kan meten. Deze sensor kan ook **luchtvochtigheid** meten. We sturen de sensor aan via I2C. Een communicatie protocol om data door te sturen.

PSoC 4200 pins : 4.0; 4.1

PSOC 4200M pins : 4.0; 4.1

Files :

https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/Temp_Humi_Sensor

I2CMaster.h

In deze file vind u de belangrijkste commands van de sensor, onder andere de slave address. Deze is zeer belangrijk om de communicatie vast te leggen met de sensor. Verder ziet u nog wat commands om te achterhalen wat je wil opvragen.

```
#define MHumi      (0xE5) // Measure Relative Humidity, Master Mode
#define MTemp      (0xE3) // Measure Temperature, Master Mode
#define MHumiNH    (0xF5) // Measure Relative Humidity, No Hold Master Mode
#define MTempNH    (0xF3) // Measure Temperature, No Hold Master Mode
```

Verder staat er de Humi_Temp_ReadReg() functie. Deze staat in de h-file omdat we de functie gaan gebruiken in de main.c. We hebben alleen een read functie gemaakt omdat we alleen maar data gaan uitlezen en niet gaan sturen.

I2CMaster.c

In de c-file staat de functie Humi_Temp_ReadReg() beschreven.

```
uint16 Humi_Temp_ReadReg(uint8 Reg) {
    uint8 Write_Buf = 0;           // buffer dat de register gaat lezen
    Write_Buf = Reg;

    uint16 Read_Buf = 0;
    uint16 Read_Buf_1 = 0;
    uint16 Read_Buf_2 = 0;

    //Step 1
    I2C_I2CMasterWriteBuf(HUMI_TEMP_ADDR, &Write_Buf, 1,
        I2C_I2C_MODE_NO_STOP);
```

```

while(0u == (I2C_I2CMasterStatus() & I2C_I2C_MSTAT_WR_CMPLT)) {}

//Step 2
I2C_I2CMasterReadBuf(HUMI_TEMP_ADDR, &Read_Buf, 2,
I2C_I2C_MODE_REPEAT_START);
while(0u == (I2C_I2CMasterStatus() & I2C_I2C_MSTAT_RD_CMPLT)) {}

Read_Buf_1 = Read_Buf << 8;
Read_Buf_2 = Read_Buf >> 8;

Read_Buf = Read_Buf_1 ^ Read_Buf_2;

return Read_Buf;
}

```

In deze functie laten we zien hoe we communiceerden met de sensor. In stap 1 schrijven we als master op de buffer het commando. Door de functie I2C_I2CMasterWrite_Buf() wordt dit uitgevoerd. Deze functie komt uit de component block die je plaats bij het design. Wanneer de sensor dit ontvangen heeft, krijgen we een ACK.

Na de ACK begint stap 2. De functie I2C_I2CMasterRead_Buf() leest wat er in de buffer staat. Die buffer is gevuld met data die de sensor daar in heeft gezet. Let op, je moet hier 2 bytes inlezen (zie datasheet). Bij de andere I2C sensoren moet je maar 1 byte inlezen. Bij deze sensor is het ook niet nodig om een control register aan te zetten.

Main.c

In de main vraag ik de functie op en stuur het commando door van wat ik wil meten. Wanneer je de data hebt gekregen, moet de data nog omgevormd worden in het gevraagde. De formules kan je vinden in de datasheet.

MPL3115A2 sensor

De MPL3115A2 sensor is een sensor die **temperatuur** kan meten. Het is één van de 3 sensoren die temperatuur kan meten. Deze sensor kan ook **hoogte** meten t.o.v. de zeespiegel. Daarbij kan je ook **luchtdruk** meten. We sturen de sensor aan via I2C. Een communicatie protocol om data door te sturen

PSoC 4200 pins : 4.0; 4.1

PSOC 4200M pins : 4.0; 4.1

Files :

https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/Alti_Press_Sensor

I2CMaster.h

In deze custom h-file staan de commands en de functie declaraties. Als je luchtdruk (altitude) wilt meten gebruik je de PRESS_OUT_*SB commands. In deze h-file staat een custom **write** functie. Deze functie is om de control register van de sensor te activeren. Dit is nodig om communicatie te leggen tussen sensor (SLAVE) en PSoC (MASTER).

```
#define TEMP_OUT_MSB    (0x04) // Temperature data MSB byte
#define TEMP_OUT_LSB    (0x05) // Temperature data LSB byte

#define PRESS_OUT_MSB    (0x01) // Pressure data MSB byte
#define PRESS_OUT_CSB    (0x02) // Pressure data CSB byte
#define PRESS_OUT_LSB    (0x03) // Pressure data LSB byte

void Alti_Press_WriteReg(uint8 Reg, uint8 value);
```

I2CMaster.c

In de c-file staan de definities van de functies. De Alti_Press_ReadReg() functie staat uitgelegd bij de Si7021 sensor (Humi_Temp_ReadReg()).

```
void Alti_Press_WriteReg(uint8 Reg, uint8 value){
    uint8 Write_Buf[2] = {0};
    Write_Buf[0] = Reg;           // register dat wordt opgevraagd
    Write_Buf[1] = value;        // value dat je in het register zet

    I2C_I2CMasterWriteBuf(ALTI_PRESS_ADDR, (uint8 *) Write_Buf, 2,
I2C_I2C_MODE_COMPLETE_XFER);
    while(0u == (I2C_I2CMasterStatus() & I2C_I2C_MSTAT_WR_CMPLT)) {}

    return;
}
```

In de Alti_Press_WriteReg() functie gaan we alleen schrijven naar de sensor. Deze functie is bedoeld om de control register actief te zetten.

Main.c

Bij deze sensor staan er nog extra functies om de gelezen data van de sensor om te zetten in een werkelijke waarde (Trans* functies). Voordat we in een oneindige loop terecht komen, waar je de sensor waardes gaat inlezen. Moet je eerst de control register aanzetten.

De DeviceID is een test om de communicatie tussen master en slave te testen. Dit werkt met het WHO_AM_I register. Als je deze opvraagt en hij geeft de data die staat in het WHO_AM_I register, deze vind je in de datasheet van de sensor. Dan weet je dat de I2C communicatie werkt.

Daarna lees je de data in van de sensoren. De rode LED is mee geprogrammeerd als een kleine test. Als de temperatuur boven 25°C is. Zal de rode LED gaan branden.

LSM9DS1 sensor

De LSM9DS1 sensor kan 3 features meten. Het bevat een **3D accelerometer**, **3D gyroscope** en een **3D magnetometer**. Vandaar dat de sensor 9 Degrees of Freedom (9DoF) wordt genoemd. Deze is wel wat specialer dan de rest. Meestal moet je maar 1 control register opzetten voor de communicatie tot stand te brengen. Bij deze heb je meerdere control registers. Deze zijn bedoeld voor het instellen van de gevoeligheid van de sensor. Je kan ook nog temperatuur mee inlezen.

PSoC 4200 pins : 4.0; 4.1

PSOC 4200M pins : 4.0; 4.1

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/9DOF>

I2CMaster.h

In de h-file vind je de commands die nodig zijn om de sensor aan te spreken. Daarbij staan de zelfgemaakte write en read functie declaraties. Wat bij deze sensor speciaal is. Dat er veel control registers zijn. Deze zijn voor de sensor gevoeligheid in te stellen.

```
#define CTRL_REG1_G      (0x10) // control register
#define CTRL_REG2_G      (0x11) // control register
#define CTRL_REG3_G      (0x12) // control register
#define CTRL_REG4        (0x1E) // control register
#define CTRL_REG5_XL      (0x1F) // control register
#define CTRL_REG6_XL      (0x20) // control register
#define CTRL_REG7_XL      (0x21) // control register
```

I2CMaster.c

Zie I2CMaster.c bij MPL3115A2 sensor.

Main.c

Wat je eerst moet doen voor je in de oneindige lus komt te zitten. Zijn de control registers instellen. Als je dit niet doet, krijg je een 0x00 bij het op vragen van de waarde.

```
DoF_WriteReg(CTRL_REG5_XL, 0x38);
DoF_WriteReg(CTRL_REG6_XL, 0x20);
DoF_WriteReg(CTRL_REG7_XL, 0x00);
```


Proximity sensors

Op de multisensor staan 4 **proximity sensoren**. Deze kan je terug vinden op elk uiteinde van het PCB. Ze staan zo, zodat je bewegingen met de hand kan detecteren. Op github staat een programma waar je één sensor mee aanstuurt om iets te detecteren.

PSoC 4200 pins : 4.2; 1.0

PSOC 4200M pins : 4.2; 1.0

Files :

https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/Proximity_sensor

Het test programma bestaat uit het detecteren met een proximty sensor die een LED aanstuurt.

```
if (CapSense_IsBusy() == 0)
{
    // LED on

    LED_Write(!CapSense_CheckIsWidgetActive(CapSense_PROXIMITYSENSOR0__PROX));
    //update baseline en start de volgende scan
    CapSense_UpdateEnabledBaselines();
    CapSense_ScanEnabledWidgets();
}
```

Wanneer er iets wordt gedetecteerd komt het in deze if-loop. De LED gaat aan wanneer er iets wordt gedetecteerd op Proximitysensor0. Deze maak je aan in het top design in het blokje CapSenseCSD. Nadat je iets gescand hebt wordt de baseline geüpdatet en kan de volgende scan van start gaan.

LDR sensor

Met deze sensor kan je de **lichtinval** meten. Het is een LDR dus de weerstand verandert van waarde a.d.h.v. lichtinval. Daardoor meten we na de LDR de spanning om zodanig te weten of er veel licht aanwezig is of niet.

PSoC 4200 pins : 2.5

PSOC 4200M pins : 2.5

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/LDR>

Op Github staat een test programma waarmee je de LDR kan testen.

```
result = ADC_GetResult16(0);
LDR_V = result * 3.3 / 2047;           // Voltage
LDR_R = 4700 * ((3.3 / LDR_V) - 1.0); // Resistance

// Printen van spanning met 2 gehele achter de komma.
LDR_Vp = LDR_V * 100;
LDR_printV[0] = (LDR_Vp / 100) + 48;
LDR_printV[1] = ',';
LDR_printV[2] = ((LDR_Vp / 10) % 10) + 48;
LDR_printV[3] = (LDR_Vp % 10) + 48;

for(i = 0; i < 4; i++){
    UART_UartPutChar(LDR_printV[i]);
}
UART_UartPutString("\n\r");

CyDelay(1000);
```

We lezen het resultaat in van de ADC. Dit is een 16 bit waarde dat je binnenkrijgt. Met dat resultaat berekenen we de spanning en de weerstand die er op dat moment over de LDR staat. Daarbij kan je de spanning ook bekijken op de seriële monitor tot 2 gehele achter de komma.

ESP8266

De ESP8266 is een **wifi-module**. Hiermee kunnen we data doorsturen en ontvangen via internet. Als je wilt werken met putty moet je extern nog extra verbinden. Je moet dan UART van de PSoC5 verbinden met de PSoC4.

PSoC 4200 pins : 0.4; 0.5

PSoC 4200M pins : 3.0; 3.1

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/ESP8266>

Voor het testen van de ESP8266 kan je op Github een programma vinden. Daar sturen we simpele commands. Wanneer de blauwe LED op de ESP8266 blinkt, weten we dat het is ontvangen en uitgevoerd.

```
UART_UartPutString("AT");  
UART_UartPutChar(13);           // Enter (CR)  
UART_UartPutChar(10);          // Line Feed (LF)  
  
CyDelay(1000);  
  
for(;;)  
{  
    UART_UartPutString("AT+CWLAP");  
    UART_UartPutChar(13);        // Enter (CR)  
    UART_UartPutChar(10);        // Line Feed (LF)  
    CyDelay(5000);  
}
```

AT command geeft weer of er connectie is. Als je via putty meekijkt moet je een OK ontvangen na de command. Daarna komen we in de oneindige lus om zo om de 5s te kijken welke netwerken zich bevinden rond de ESP8266. De "Enter" en "Line Feed" is zeer belangrijk. Hierdoor stuur je het commando werkelijk door. Je kan dit ook vervangen door `UART_UartPutString("AT+CWLAP\n\r")`.

OLED + SD-kaart

Deze 2 componenten werken met SPI. We hebben de communicatie met SPI niet kunnen verwezenlijken. Maar de connectie met het component en de pinnen van de PSoC liggen wel vast. De OLED is een scherm waarop je je gemeten data of tekst kan weergeven. Het is net als een LCD scherm, maar alleen complexer. De SD-Kaart dient om je data in op te slaan of data uit te lezen.

OLED

PSoC 4200 pins : 3.1; 3.0; 0.6; 3.4; 2.6

PSOC 4200M pins : 6.0; 6.1; 6.2; 2.7; 0.2

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/OLED>

De OLED werkt met een driver. De SD1306 datasheet kan je terug vinden op Github. Deze datasheet heb je nodig om de OLED aan te sturen.

SD-Kaart

PSoC 4200 pins : 3.1; 3.0; 0.6; 3.6

PSOC 4200M pins : 6.0; 6.1; 6.2; 0.3

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/SD-Kaart>

IR LED

Met dit component kan je **IR** signalen **uitsturen**. Je kan een IR code ontvangen met de IR receiver en de code doorsturen via de IR led. Ook kan je je eigen codes maken en dat doe je in het programma.

PSoC 4200 pins : 2.3

PSOC 4200M pins : 2.3

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/IR%20LED>

IR Receiver

Met dit component kan je **IR** signalen **ontvangen**. Deze signalen kan je dan bewerken in je programma. De receiver die we gebruiken hebben is de TSOP38238.

PSoC 4200 pins : 2.4

PSOC 4200M pins : 2.4

Files :

<https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/IR%20Receiver>

Over de IR LED en receiver hebben we geen voorbeeldprogramma dat samenhangt met het bordje. Er zijn wel voorbeeldprogramma's op Github om er van af te leiden hoe je ze moet gebruiken.

Sounddetector

De sounddetector **detecteert geluid**. De geluidsdetector levert niet alleen een audio-uitgang, maar ook een binaire indicatie van de aanwezigheid van geluid en een analoge representatie van de amplitude. De 3 uitgangen zijn gelijktijdig en onafhankelijk, dus je kan elke uitgang samen of apart gebruiken wanneer je wilt. De envelope-uitgang kunt u gemakkelijk uitlezen via amplitude van het geluid door simpelweg het meten van de analoge spanning.

PSoC 4200 pins : 2.0; 2.1; 2.2

PSOC 4200M pins : 2.0; 2.1; 2.2

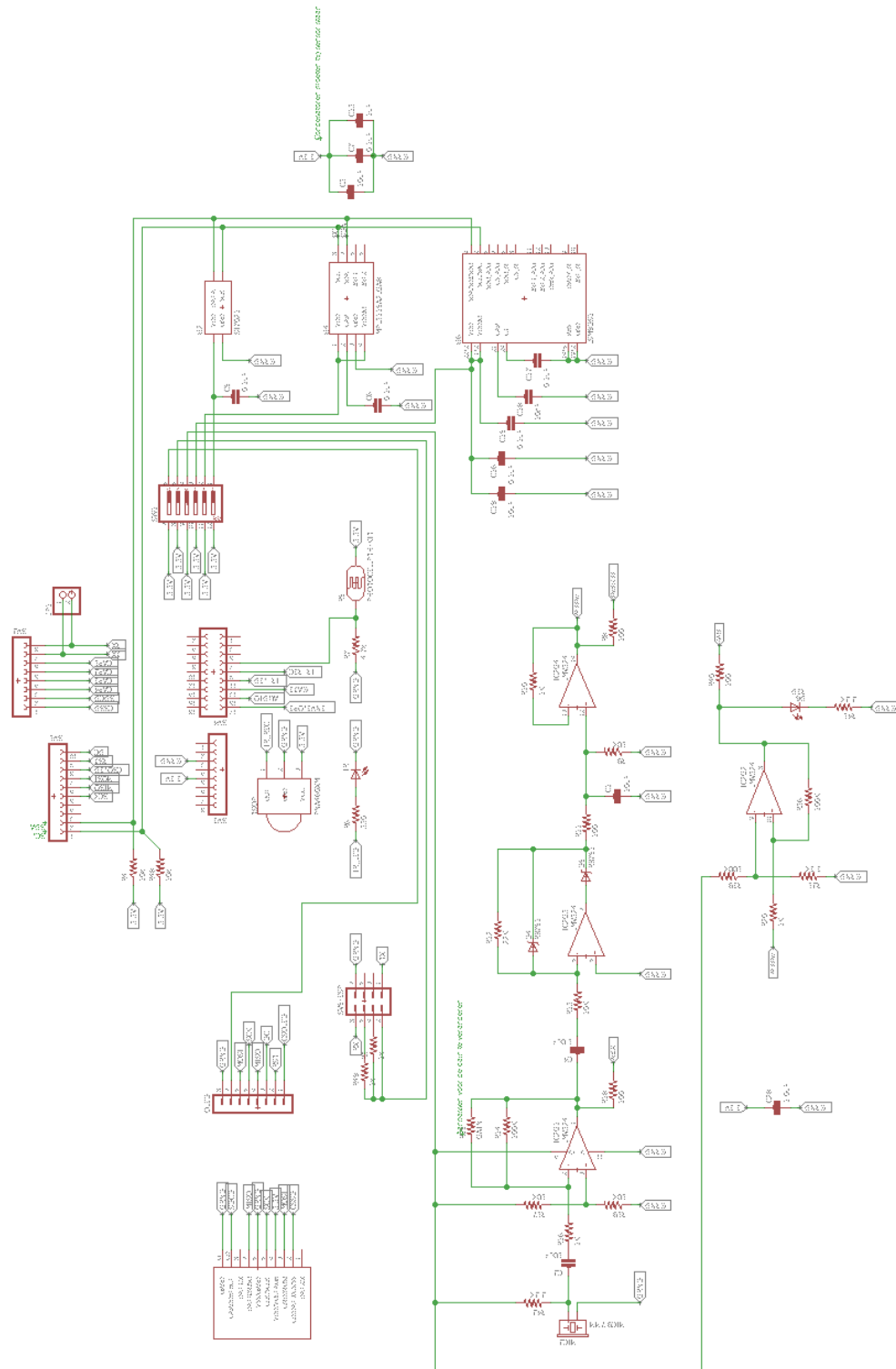
Files :

<https://github.com/Dries11400702/MultiSensor/tree/master/Componenten/Sounddetector>

B. Schema

Het schema is opgebouwd in Eagle. We hebben de schema's genomen die op Sparkfun stonden en samen gevoegd tot één groot schema.

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Eagle%20files>

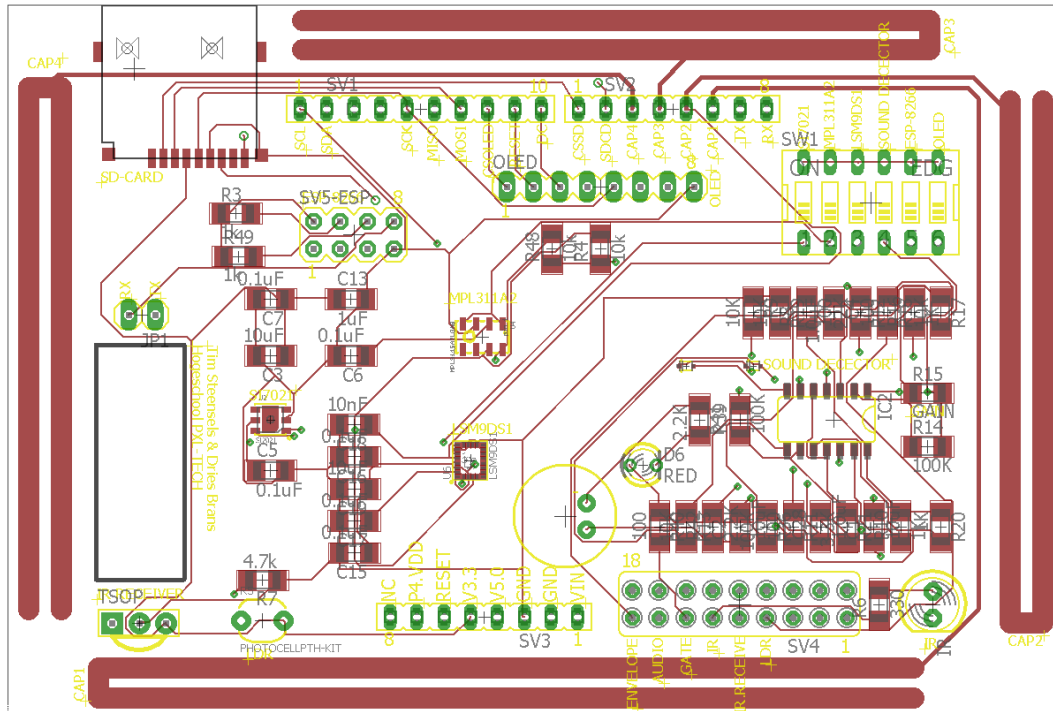


C. Board design

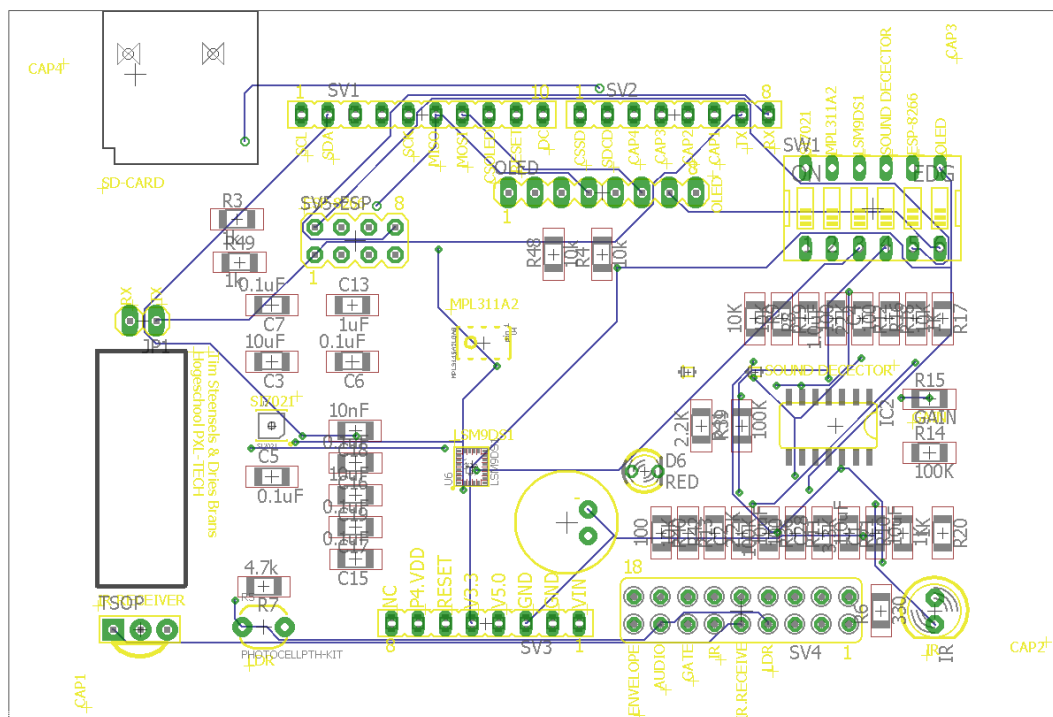
Het board design is gedaan in Eagle.

Files : <https://github.com/Dries11400702/MultiSensor/tree/master/Eagle%20files>

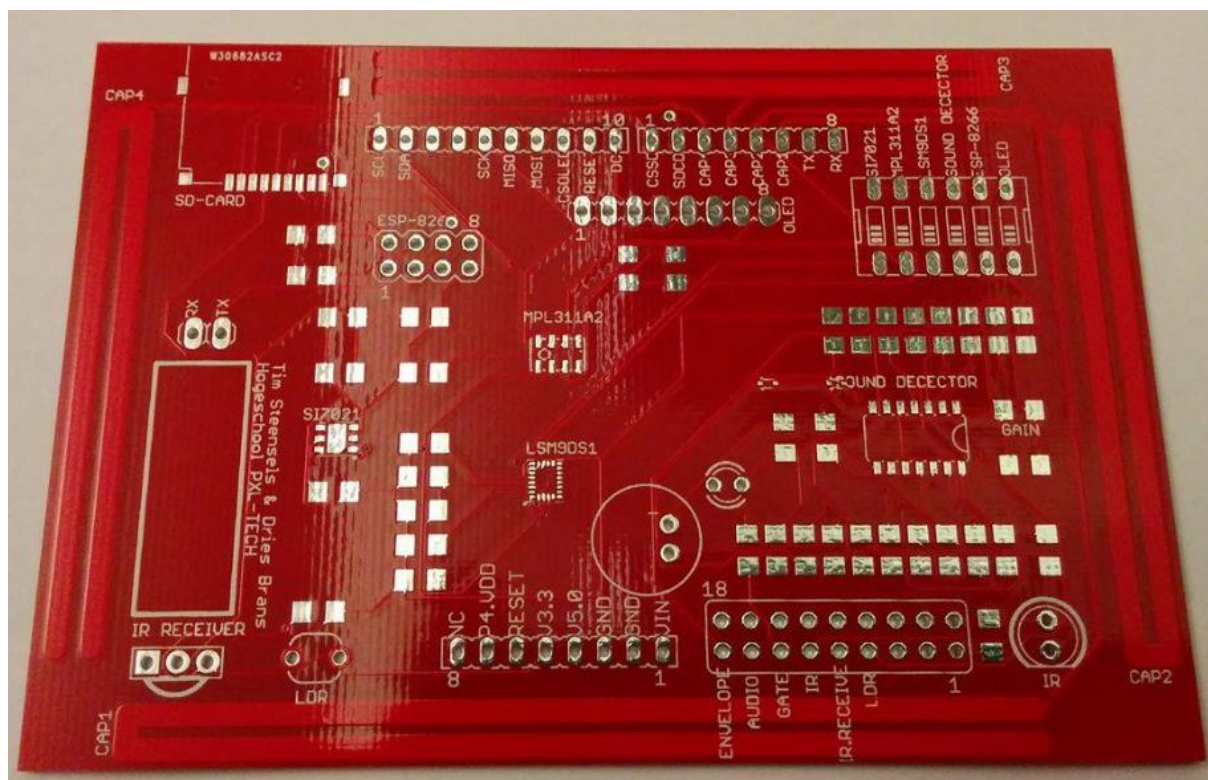
TOP



BOT



Resultaat



D. Bestuklijst

Dit is de lijst van componenten die op de PCB staan. Bij elk component vind je de site voor het bekijken van de datasheet. Als je componenten dubbel besteld (safe) kost het 75,22 euro. De werkelijke prijs van als je fixed aankoopt bedraagt 51,47 euro. Deze kosten zijn zonder de bestelling van de PCB's.

Bestellijst (safe)					
Component	#	Leverancier	url	Prijs/stuk	Probleem
DIP-Switch 6x	2	TME	http://www.tme.eu/nl/details/a6d-6100/di	€ 1,52	
SD-kaart	2	TME	http://www.tme.eu/nl/details/mcc-sdmicr	€ 1,40	
Mini photocell	2	TME	http://www.tme.eu/nl/details/pgm5526/fo	€ 0,25	
MPL3115A2 (altitude sensor)	2	Farnell	http://be.farnell.com/ntp/mpl3115a2/pres	€ 2,97	Moeilijk solderen
SI7021 (Humidity and Temp. Sensor)	2	Farnell	http://be.farnell.com/silicon-labs/si7021-a2	€ 5,53	Moeilijk solderen
LSM9DS1TR (3x accelerometer)	2	Farnell	http://be.farnell.com/stmicroelectronics/ls	€ 9,04	Moeilijk solderen
LM324M	2	Farnell	http://be.farnell.com/fairchild-semiconduc	€ 0,62	
RB751 schottky diode	2	Farnell	http://be.farnell.com/ntp/rb751v40/diode	€ 0,20	
10nF condensator	2	Farnell	http://be.farnell.com/multicomp/mc0805y	€ 0,65	
330 weerstand	4	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,02	
10K weerstand	10	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
100 weerstand	5	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
1K weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
2K2 weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
3K3 weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
22K weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
100K weerstand	3	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
4k7 weerstand	4	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,02	
10µF condensator	10	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
0,1µF condensator	12	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
1µF condensator	4	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
Rode LED	1	Reichelt	http://www.reichelt.com/LEDs-3-mm/LED-	€ 0,09	
male pin headers	1	Reichelt	https://www.reichelt.com/MPE-087-1-050	€ 0,89	
Microfoon	2	Reichelt	https://www.reichelt.com/MCE-100/3/inde	€ 0,66	
OLED	1	Antratek	https://www.antratek.be/micro-oled-break	€ 18,09	
ESP8266	1	Antratek	https://www.antratek.be/wifi-serial-transc	€ 8,41	
TME				€ 6,34	
Farnell				€ 38,08	
Reichelt				€ 4,30	verzendingkosten
Antratek				€ 26,50	
Totaal				€ 75,22	

Bestellijst (fixed)					
Component	#	Leverancier	url	Prijs/stuk	Probleem
DIP-Switch 6x	1	TME	http://www.tme.eu/nl/details/a6d-6100/di	€ 1,52	
SD-kaart	1	TME	http://www.tme.eu/nl/details/mcc-sdmicr	€ 1,40	
Mini photocell	1	TME	http://www.tme.eu/nl/details/pgm5526/fo	€ 0,25	
MPL3115A2 (altitude sensor)	1	Farnell	http://be.farnell.com/ntp/mpl3115a2/pres	€ 2,97	Moeilijk solderen
SI7021 (Humidity and Temp. Sensor)	1	Farnell	http://be.farnell.com/silicon-labs/si7021-a2	€ 5,53	Moeilijk solderen
LSM9DS1TR (3x accelerometer)	1	Farnell	http://be.farnell.com/stmicroelectronics/ls	€ 9,04	Moeilijk solderen
LM324M	1	Farnell	http://be.farnell.com/fairchild-semiconduc	€ 0,62	
RB751 schottky diode	1	Farnell	http://be.farnell.com/ntp/rb751v40/diode	€ 0,20	
10nF condensator	1	Farnell	http://be.farnell.com/multicomp/mc0805y	€ 0,65	
330 weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,02	
10K weerstand	5	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
100 weerstand	5	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
1K weerstand	1	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
2K2 weerstand	1	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
3K3 weerstand	1	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
22K weerstand	1	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
100K weerstand	3	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,03	
4k7 weerstand	2	Reichelt	https://www.reichelt.com/SMD-1206-from	€ 0,02	
10µF condensator	5	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
0,1µF condensator	6	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
1µF condensator	2	Reichelt	https://www.reichelt.com/Aluminum-Elect	€ 0,04	
Rode LED	1	Reichelt	http://www.reichelt.com/LEDs-3-mm/LED-	€ 0,09	
male pin headers	1	Reichelt	https://www.reichelt.com/MPE-087-1-050	€ 0,89	
Microfoon	1	Reichelt	https://www.reichelt.com/MCE-100/3/inde	€ 0,66	
OLED	1	Antratek	https://www.antratek.be/micro-oled-break	€ 18,09	
ESP8266	1	Antratek	https://www.antratek.be/wifi-serial-transc	€ 8,41	
TME				€ 3,17	
Farnell				€ 19,04	
Reichelt				€ 2,76	verzendingkosten
Antratek				€ 26,50	
Totaal				€ 51,47	

E. Realisatie

Tijdens het proces is er een schema opgesteld. Bij deze gaan we dit proces overlopen om te bekijken wat er voltooid is en wat niet en waarom niet. Dit wordt verwezenlijkt met het plan van aanpak erbij te nemen. Onder elke taak staat uitleg of dat wij vonden dat de taak goed was uitgevoerd of niet. Er is uitleg bij wanneer we vonden dat de taak niet goed in orde was.

Plan van aanpak kan je vinden op deze link :

<https://github.com/Dries11400702/MultiSensor/tree/master/PVA>

Taak componenten

De componenten kiezen werd redelijk snel verwezenlijkt. De componenten lijst kan u eerder in de paper bekijken. Naargelang het semester vielen er een paar componenten weg. Wegens dat ze niet veel nut hebben op de multisensor. De componenten en sensoren kwamen wel wat later aan dan verwacht en heeft wat voor problemen gezorgd.

OK

Taak Hoogte/Druk Sensor (MPL3115A2)

Nadat we de sensor wat hebben bestudeerd wat het juist allemaal kan, wouden we de sensor testen. Dit heeft veel tijd in beslag genomen wegens I2C. Het was moeilijk om de communicatie tussen sensor en PSoC vast te leggen. We kunnen vaststellen van het “testen van gevoeligheid” is weggevallen omdat het tijd heeft gekost voor de I2C communicatie. Het plaatsen op de PCB was vrijwel direct gedaan wanneer we begonnen aan de PCB.

OK maar,

lang zoeken naar de I2C communicatie.

Taak Fotocel (LDR)

LDR ging redelijk snel. De sensor is basiskennis en het heeft niet lang geduurd voor die af was. Het programmeren is een analoge waarde inlezen en vandaaruit verder werken. Het plaatsen op de PCB was snel gedaan.

OK

Taak RGB en gebarensensor of 4 cap. Sensoren

We hebben direct besloten om 4 proximity sensoren te plaatsen i.p.v. de dure RGB en gebarensensor. Met deze sensor kan je hand bewegingen detecteren. Dit hebben vervangen door simpel weg capacitieve velden aan te leggen op de uiteindes van de PCB. We hebben dit alleen getest als ze iets detecteren. Er is geen code geschreven die de handdetectie doet. Maar je kan wel vanuit de code werken die meegegeven is op Github om dit te realiseren.

OK maar,
geen code voor handdetectie.

Taak IR Receiver

De IR receiver had iets meer tijd in beslag genomen dan voorzien. De code is niet helemaal gelukt. Er zijn testen uitgevoerd met deze sensor maar het faalde steeds. Dit lag aan onze code. We hebben dit dan niet verder uitgevoerd om niet te veel tijd te verliezen.

NIET OK
Nog geen werkende code.

Taak IR LED

De IR LED heeft hetzelfde probleem als de IR Receiver. De code was een probleem en hebben dit achterwegen gelaten wegens tijdsgebrek.

NIET OK
Nog geen werkende code.

Taak Geluidsdetector

De geluidsdetector van Antretek is niet aangekomen. Dus we hebben dit component niet kunnen testen. Daardoor is er ook geen code die je kan terug vinden op Github. We gaan er vanuit dat de geluidsdetector werkt en hebben hem zonder testen geplaatst op de PCB.

OK maar,
geen code wegens probleem bij bestelling.

Taak SD-kaart

De SD-kaart werkt op SPI. Doordat I2C een tijd heeft geduurd. Hadden we minder tijd voor SPI. De SPI communicatie hebben we uiteindelijk niet gerealiseerd. We vonden het belangrijker om een PCB te hebben dan code voor 2 componenten (SD-kaart en OLED). De SD-kaart is wel geplaatst op de PCB en de code kan later nog altijd bijgezet worden.

NIET OK

SPI communicatie niet verwezenlijkt.

Taak Temp. en vochtigheid sensor (Si7021)

De Si7021 sensor werkt op I2C. Nadat I2C werd gerealiseerd was dit snel gedaan. De sensor plaatsen op de PCB gaf ook geen problemen.

OK

Taak 9 Degrees of Freedom (LSM9DS1)

Deze sensor communiceert via I2C. De sensor gaf geen problemen nadat de I2C communicatie werkte. De sensor staat onder de OLED op de PCB. Het is de moeilijkste component om te solderen. De soldeerpadjes zijn zeer klein.

OK

Taak ESP8266

De ESP8266 heeft een week langer in beslag genomen dan verwacht. De UART communicatie tussen PSoC4, PSoC5 en ESP8266 was een probleem. Dit is opgelost door de aansluitingen op de PCB.

OK maar,

te veel tijd verloren wegens aansluitingen met PCB.

Taak I2C

De I2C was een groot probleem. De communicatie tussen PSoC en sensor heeft veel tijd in beslag genomen. De studie over I2C was verkeerd opgevat. Daardoor was de code nooit correct om te communiceren over I2C. Nadat we de fout hadden opgemerkt, was de communicatie snel gelegd.

OK maar,

veel tijd verloren door fout van ons.

Taak SPI

De SPI is niet gelukt. Dit hebben we getest op de OLED. De communicatie voor SPI van PSoC naar sensor is zeer moeilijk met PSoC. We hebben het niet kunnen achterhalen om het te laten werken. We hebben dit dan laten vallen voor aan de PCB te beginnen.

NIET OK

Tijdsgebrek

Taak OLED

De OLED is niet gelukt om communicatie te leggen met de PSoC. Dit hebben we laten vallen voor tijdgebrek.

NIET OK

Tijdsgebrek

Taak PCB apart voeden

Aan dit zijn we niet begonnen wegens tijdgebrek.

NIET OK

Tijdsgebrek

Taak Stuurprogramma

We hebben gekozen om elk component apart een stuurprogramma te maken. Dit is overzichtelijker dan één groot programma. Deze programma's kan je allemaal terugvinden op Github. Niet elke sensor heeft een voorbeeldprogramma.

OK maar,

niet elke sensor heeft stuurprogramma.

Taak Extra

Omdat er geen extra tijd was zijn we aan dit nooit geraakt. Om dit te vervangen hebben we een Github gemaakt waar je de bestanden makkelijker kan vinden en opvragen.

IV. Besluit

- PCB is afgemaakt maar niet volledig getest.
- De sensoren brengen extra meer waarde toe aan de PSoC, vooral aan de PSoC4200.
- Te laat begonnen aan PCB ontwerp.
- Een groot voorbeeldprogramma was niet mogelijk om te maken.
- Niet alle componenten hebben een testprogramma.
- SPI is niet verwezenlijkt.
- I2C is verwezenlijkt.

V. Verbeteringen

- Planning beter opvolgen door een taak dat moeilijker is, op dat moment te laten vallen.
- PCB ontwerp kan een beter design hebben.
- Datasheets beter bekijken om zodanig geen foute code te schrijven.