



DON BOSCO
T.I. HELCHTEREN

Fusor

Robin Peeten

6 ICT

Mentor: Dhr. Bertho Kevin

Schooljaar 2018-2019

Inhoud

Voorwoord	4
Inleiding.....	5
Abstract	6
Blokschema	7
1. Hardware.....	9
1.1 Hoogspanningsgenerator.....	9
1.1.1 Ingang hoogspanningsgenerator	10
1.1.2 Stroommeting.....	10
1.1.3 Transformatoren	10
1.1.4 Spanning Verdubbelen	11
1.1.5 Afvlakken	12
1.1.6 Hoogspanningsmeting.....	12
1.1.7 Temperatuurmeting	12
1.1.8 Hoogspanningsveiligheid	13
1.1.9 Hoogspanningsgenerator specificaties.....	13
1.2 Geiger counter	13
1.2.1 Geiger Muller tube	14
1.2.2 Geiger counter hoogspanningsgenerator.....	14
1.2.3 Geiger counter hoogspanningsgenerator kalibreren	15
1.2.4 Geiger counter puls detectie	15
1.2.5 Geiger counter behuizing	16
2 Hardware elektronica.....	17
2.1 Relais schakeling	17
2.2 Spanning/stroom meting ingang	17
2.3 Elektronica voedingspanningen.....	18
2.4 Veiligheidscontrole	19
2.5 MCP3008 ADC.....	19
2.5.1 Metingen.....	19
2.5.2 Spanningsreferentie.....	20
2.5.3 SPI.....	20
2.5.4 SPI Isolatie	20
2.6 Vacuümsensor.....	21
2.6.1 Druksensor	21

2.6.2	Load cell amplifier	21
2.6.3	HX711 communicatie	22
3	Software	22
3.1	Attiny85 safety controller code.....	22
3.2	Attiny85 Vacuüm sensor code.....	22
3.3	Arduino mini geiger counter code.....	23
3.4	Raspberry pi C# code.....	23
3.4.1	C# startup	24
3.4.2	C# Dataopslag ophalen.....	25
3.4.3	C# threads	26
3.4.4	C#-communicatie	27
3.4.4.1	ADC SPI.....	27
3.4.4.2	Safety controller pin status.....	28
3.4.5	C# Conversie/checksum	29
3.4.5.1	C# Conversie/checksum spanning/stroom en temperatuur	29
3.4.5.2	Checksum	30
4	Fusieproces en de reactorkamer.....	32
4.1	Basisprincipe fusie met een elektrisch veld	32
4.2	Reactor grid	33
4.3	Reactorkamer	34
4.4	Stralingen	35
4.5	Brandstof.....	35
	Besluit.....	36
	MCP3008 datasheet belangrijke pagina's	37
	LM4040 Datasheet	40
	SBM20 Geiger muller tube specificaties	42
	PCB elektronica schema's.....	43
	PCB Silk layer	45
	PCB Top layer.....	46
	PCB bottom layer.....	47
	Elektronica Geiger counter.....	48
	Hoogspanning schema	49

Voorwoord

Mijn GIP omvat een groot deel van mijn studie en is een vakoverschrijdend project. Het is een zeer uitdagend project met gevaarlijke onderdelen. Hierdoor heb ik er veel tijd en energie in gestoken. Er werd veel aandacht besteed aan veiligheid en afwerking. Het doel was een demonstratie fusiereactor te maken die bestuurd en gecontroleerd wordt door zelfgemaakte elektronica en software waarbij alle informatie wordt weergegeven op een display. De reactor maakt een plasma dat een mooi visueel effect geeft. Deze keuze werd gemaakt door mijn interesse in elektronica en kernfysica. Er waren heel wat obstakels te overwinnen zowel in hardware als software.

Gelukkig dat ik van mijn leerkrachten en mijn ouders toestemming heb gekregen om dit project te bouwen. Mvr. Lorina Vandenbroeck heeft goed geholpen met het nalezen en de opmaak van mijn GIP-bundel. Mijn ouders hebben mij financieel bijgestaan en geholpen waar ze konden, mijn teksten nagelezen om hierbij te helpen en te verbeteren. Deze wil ik dus allemaal ook zeer veel bedanken!

Hieraan werken was zeer boeiend en leerrijk. Dit project brengt mijn interesse in nucleaire fysica en elektronica echt naar boven en leerde me veel bij.

Inleiding

Kernenergie is onze belangrijkste schone energiebron, momenteel in de vorm van kernsplitsing en in de toekomst misschien op basis van fusie. Met mijn jarenlange interesse in het gebied van kernfysica en kernenergie loop ik al bijna drie jaar lang met het idee om mijn eigen demonstratie fusiereactor te bouwen. Nu had ik dus de kans om dit te doen via mijn GIP.

Ik zal laten zien hoe dit te verwezenlijken is, hoe het werkt en hoe ik het uiteindelijk uitgevoerd heb op een redelijk goedkope manier door creatief te zijn met de onderdelen en werking.

Abstract

Het doel van mijn GIP is een demonstratie fusiereactor te maken met nodige elektronica en software voor de metingen en sturing en het displayen van de informatie.

Hiermee laat ik de werking van een fusiereactor zien op basis van een elektrisch veld dat ionen met hitte doet fuseren. Door een zeer hoge negatieve spanning op een metalen grid te zetten zal er een plasma gecreëerd worden en zullen de ionen tot het grid worden toegetrokken. Veel van deze zullen door de aantrekkingskracht met een snelheid in het midden van het grid tegen elkaar botsen waar ze een kans tot fusie hebben indien ze genoeg energie bevatten. Dit gebeurt in een vacuüm om elektrische overslag te voorkomen en de hoeveelheid ionen te verminderen tot een werkbare hoeveelheid.

De hoogspanningsgenerator bestaat uit 2 hoogspanningstransformatoren (2kv AC) waarvan de primaire spoelen parallel staan en de secundaire spoelen in serie staan om zo de spanning tot 4kv AC te brengen. Hierna wordt de AC spanning nog eens verdubbeld en gelijkgericht met 2 condensatoren in serie en een diode. Het principe van deze spanningsverdubbeling is hetzelfde als in microgolfovens waar deze onderdelen ook vandaan komen. De spanning wordt dan afgevlakt en geeft uiteindelijk een uitgangsspanning van 13KV DC met een ingang van 230V AC. Dit zou in theorie ook een vermogen van meer dan 1500W kunnen leveren. De uitgangsspanning meet ik met behulp van speciale hoogspanningsweerstand die een laagspanning creëren wat gemeten wordt door een ADC in de elektronica. Het geheel is opgesteld in een aquarium, gevuld met olie voor de hoogspanningsisolatie en koeling van de componenten. Deze temperatuur wordt gemeten door de elektronica. De ingangsspanning hiervan wordt geregeld met behulp van een variac (variabele transformator) om zo ook de uitgangsspanning te kunnen regelen.

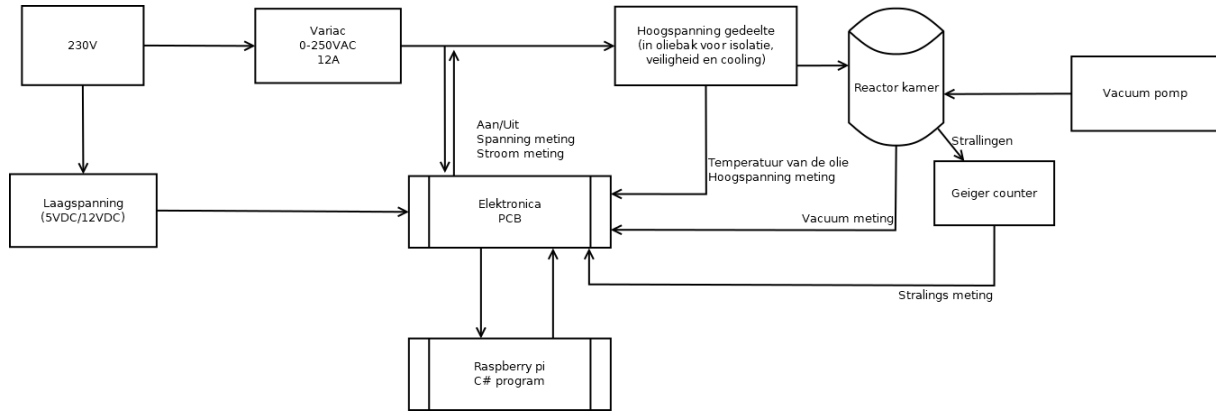
De elektronica bestaat uit een raspberry pi en een touchscreen display met software geschreven in C#, deze communiceert met eigen ontworpen elektronica op een pcb. De elektronica dient om alle data en informatie door te geven naar de raspberry pi, en met behulp van een attiny 85 zelf ook te beslissen of de meest belangrijke signalen binnen de parameters zijn, zodat wanneer alles veilig is de hoogspanning wordt aangezet.

De data bestaat uit spanning- en stroomwaarde aan de ingang van de hoogspanningsgenerator, de uitgangsspanning, temperatuur van de hoogspanningsgenerator, stralingsmeting met een zelfgemaakte Geiger counter, vacuüm niveau en spanningen nodig voor de elektronica.

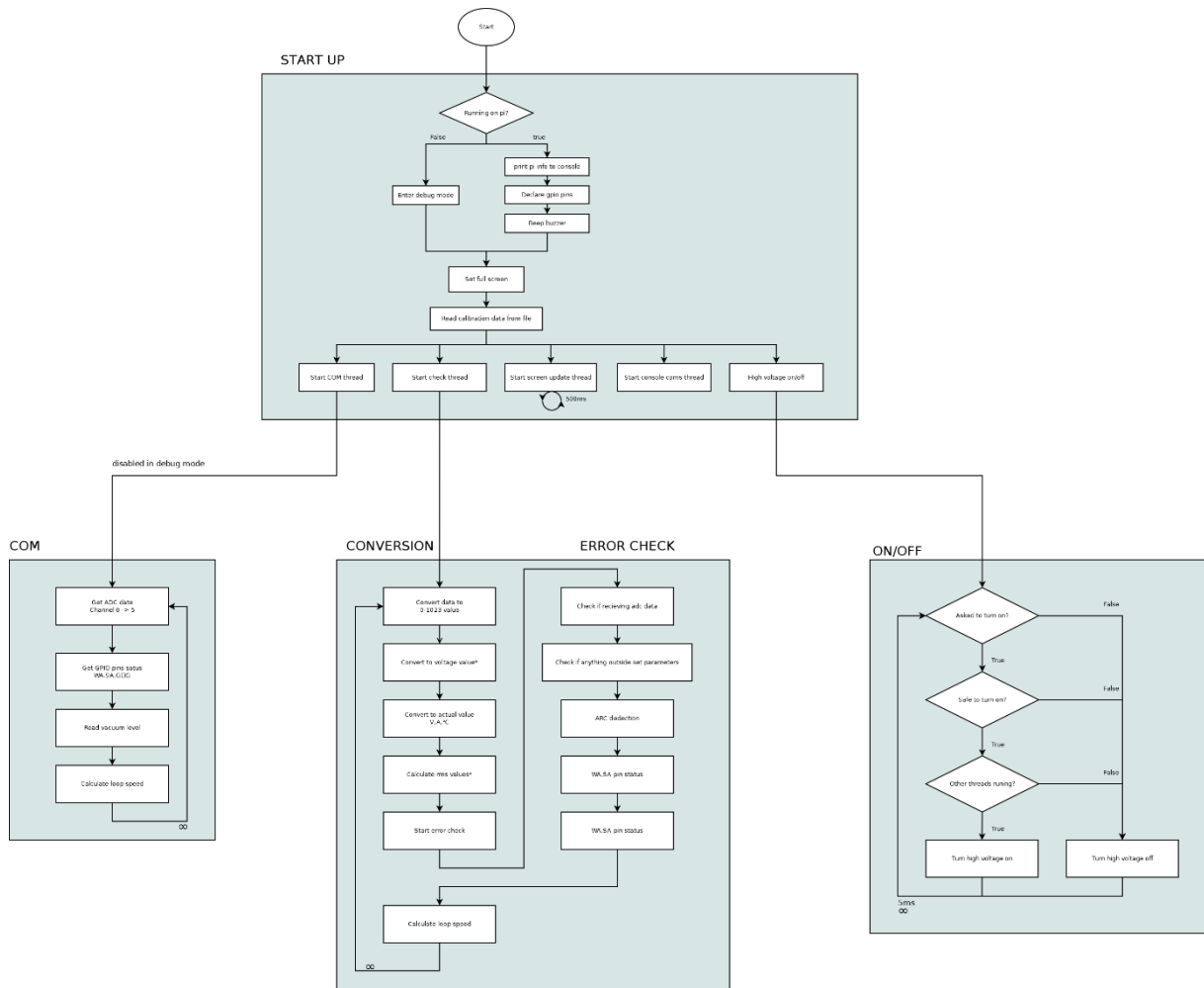
Dit wordt dan allemaal weergegeven op een bedienbaar touchscreen display waarmee de hoogspanning wordt aangezet en eventuele fouten gemeld.

Blokschema

Algemeen blokschema



Algemeen blokschema



Blokschema C# code

1. Hardware

De uitdaging bestaat erin om reactor kamer te bouwen, een vacuümruimte met een hoogspanningsbron. Hierin vindt een fusiereactie plaats. Hier worden allerlei metingen verricht en het proces gecontroleerd. Dit doe ik op een gebruiksvriendelijke manier.

1.1 Hoogspanningsgenerator

De reactorkamer heeft een zeer hoge negatieve spanning nodig met redelijk wat vermogen. Eerst had ik hiervoor gekeken naar transformators voor neonverlichting. Deze zijn moeilijk te vinden omdat ik de oude transformatoren nodig heb. De nieuwe elektronische niet geschikt zijn voor deze applicatie. Dit zou ook te duur zijn geweest. Mijn tweede keuze en eigenlijk de beste is een oude X-Ray transformator. Die werden vroeger gebruikt voor de X-ray lampen te voeden met een hoogspanning van $\sim 30\text{KV}$. Maar deze kon ik niet vinden of niet tot in België laten komen. Dan had ik het idee om het zelf proberen te maken. Ik gebruikte dan maar de hoogspanningstransformators in microgolfovens, maar deze geven niet genoeg spanning. Door te zoeken hoe het werkt kwam ik erop uit dat microgolfovens ook gebruik maken van een spanningsverdubbeling. Hier kon ik dan ook gebruik van maken. Maar dan was de spanning nog niet hoog genoeg. Ik kreeg het idee om twee systemen in serie te zetten. Ik maakte gebruik van oude onderdelen, wat mij een heel goedkope hoogspanningsgenerator levert van -12KV met een vermogen van 1500W .

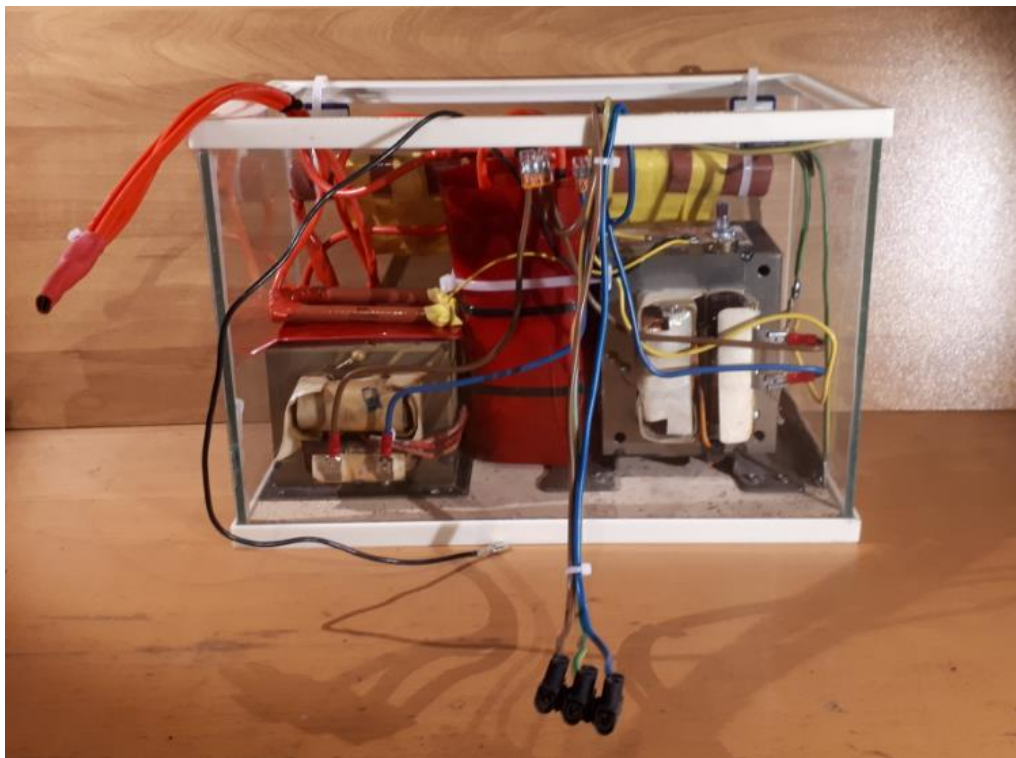
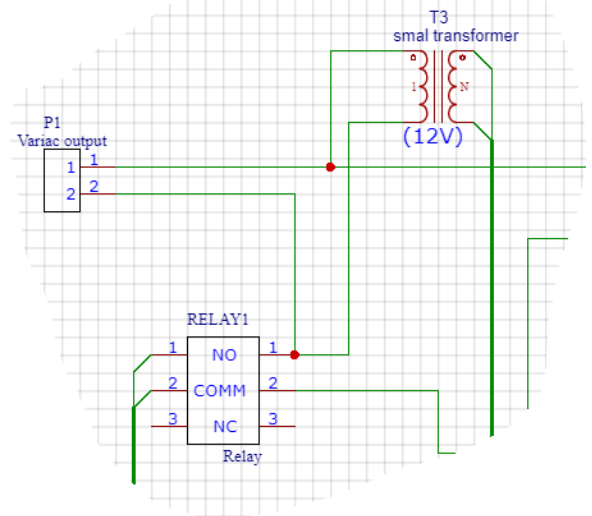


Foto hoogspanning generator

1.1.1 Ingang hoogspanningsgenerator

Voor de ingang van de hoogspanningsgenerator zal een variac (variabele transformator) aanwezig zijn van 12A met een automaat voor kortsluit- en overbelastingsbeveiliging. De variac dient om de spanning van 0 tot +230V om zo ook de uitgangsspanning van de generator te regelen van 0 tot 12KV. Direct daarna is een kleine transformator aangesloten om de uitgang van de variac te meten. Dit dient voor de veiligheid zodat de spanning traag van 0 tot de maximum spanning kan worden gebracht en niet dat deze direct op zijn maximumwaarde is. Een relais op het pcb bij de elektronica schakelt de opstelling.



Deel schema hoogspanningsgenerator

1.1.2 Stroommeting

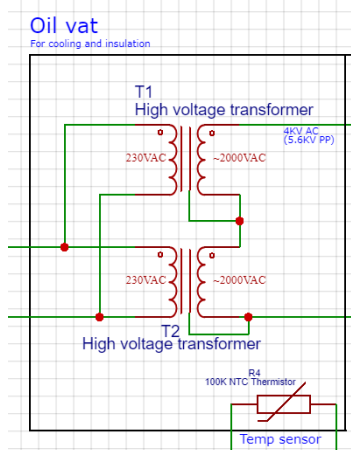
In serie is ook een stroommeter aangesloten. Dit is een ACS712, een klein breakoutboard met een microchip die de stroom meet door het opgewekt magnetisch veld. Dit geeft de waarde door op de signaalpin die bij 0A op 2.5V zal staan. Het zal 100mv/A zakken of dalen naargelang de richting van de stroom. Deze stroommeter kan een maximumstroom van 20A meten wat meer dan genoeg is voor deze applicatie. De waarde van de uitgang wordt doorgegeven naar de elektronica.



Foto ACS712

1.1.3 Transformatoren

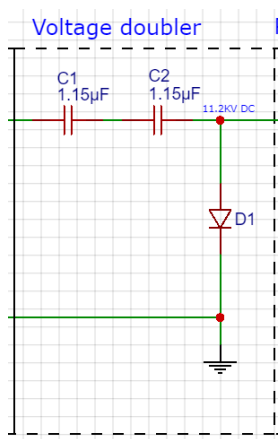
De ingangsspanning komt binnen op twee hoogspanningstransformatoren die uit oude microgolfovens zijn gehaald. Standaard krijgen deze 230V-AC binnen en leveren meer dan 2KV-AC. Ik heb hiervan de primaire spoelen in parallel gezet en de secundaire spoelen in serie om zo de totale uitgangsspanning te verdubbelen tot meer dan 4KV-AC.



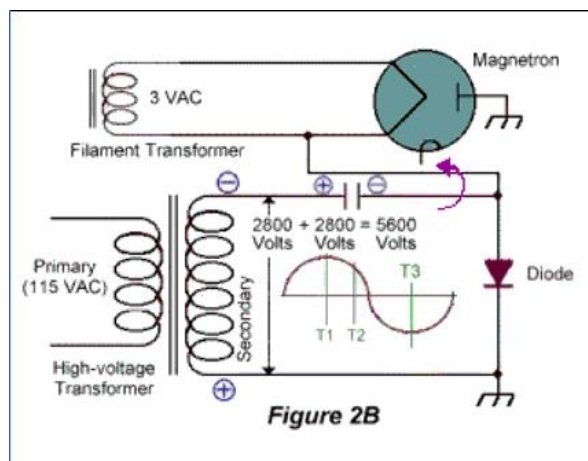
Deel schema hoogspanningsgenerator

1.1.4 Spanning verdubbelen

Om de spanning nog te verhogen heb ik ook hetzelfde principe toegepast als in microgolfovens. Hiervoor gebruik ik twee condensators in serie met een hoogspanningsdiode. In de ene helft van de AC-cyclus zullen de condensators normaal worden opgeladen maar in de andere richting houdt de diode dit tegen en wordt de spanning over de diode verdubbeld. Met de ingang van 4KV-AC (5.6KV PP) wordt de uitgang nu 11.2KV-DC in theorie. In praktijk is dit 13KV wanneer er geen belasting is. Wat net op de maximumwaarde is van de diode.



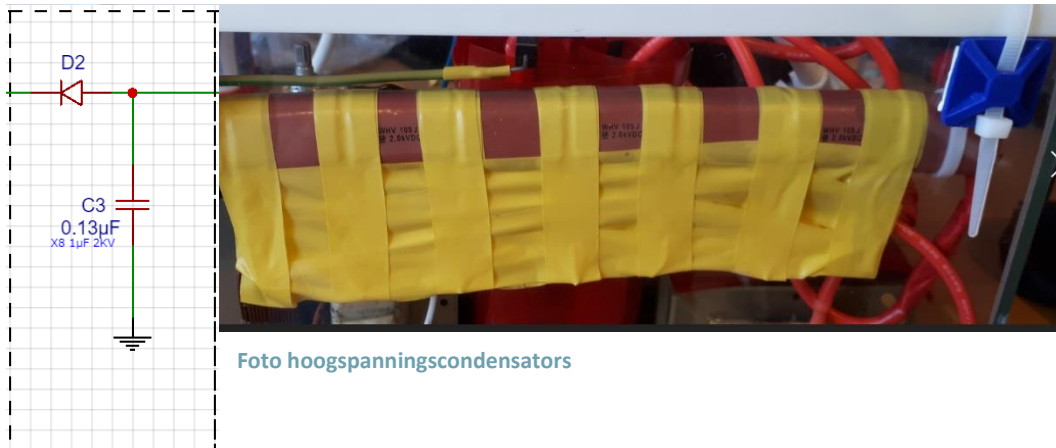
Deel schema hoogspanning generator



Spanning verdubeler in microgolfoven

1.1.5 Afvlakken

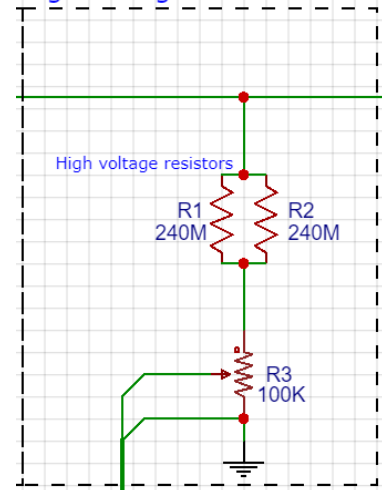
Voor het afvlakken is ook eerst weer een hoogspanningsdiode gezet zodat de lading in de condensators blijft. Omdat één grote hoogspanningscondensator niet veel voorkomt en daardoor duur zijn om te kopen, heb ik er acht van 2kv en 1 μ F elks gebruikt. Zo kom ik tot een maximum spanning van 16kv. De totale capaciteit is 0.13 μ F en kan dan ongeveer 11 joules opslaan. Dit zal de uitgang niet perfect mooi vlak maken, maar is vooral voor het opnemen van het vermogen van de reactor te verhogen.



1.1.6 Hoogspanningsmeting

Om de hoogspanning direct te kunnen meten, is een spanningsverdeler gebruikt met weerstanden. Dit is een simpel design maar door de hoge spanning kan dit niet zomaar gebeuren. Hiervoor gebruik ik speciale hoogspanningsweerstand die elk 20KV aankunnen. Deze zijn elk 240M ohm en er worden er twee in parallel gebruikt. Ze zijn in parallel geschakeld om de meting van de ADC in de elektronica gemakkelijker te maken. Bij elke meting is er namelijk een kleine stroom aanwezig en door de hoge weerstandswaarde zakt de gemeten spanning en wordt de meting inaccuraat. Bij de maximumspanning van 12KV vloeit er maar een stroom van 100 μ A.

high voltage measurement



Schema hoogspanningsmeting

1.1.7 Temperatuurmeting

Bepaalde componenten kunnen vrij warm worden, zeker onder belasting. Eén van de transformatoren wordt al +80°C zonder belasting. Hiervoor wordt alles in olie gezet en wordt de temperatuur hiervan gemeten. De sensor hiervoor is een ntc-thermistor van 100K. Dat is een

weerstand waarvan de weerstandswaarde zal dalen met een toenemen van de temperatuur. De elektronica kan deze waarde meten en omzetten in een temperatuur.



Foto thermistor

1.1.8 Hoogspanningsveiligheid

Voor personenbescherming is de hele opstelling opgebouwd in een klein aquarium en heeft de bovenkant een geaarde metalen draadwerk zodat geen hoogspanningselement aangeraakt kan worden. Binnenin is een speciale hoogspanningsdraad gebruikt met een doorslagvastheid van 40 KV en zijn de condensators en hoogspanningsweerstanden ook geïsoleerd met plastic. Bij andere onderdelen en verbindingen is plakband en krimpkous gebruikt. Het aquarium wordt gevuld met olie om koeling van de componenten en elektrische isolatie en ionisatie van de lucht te vermijden.

1.1.9 Hoogspanningsgenerator specificaties

Ingangsspanning = 230V AC

Uitgangsspanning = -12KV DC max

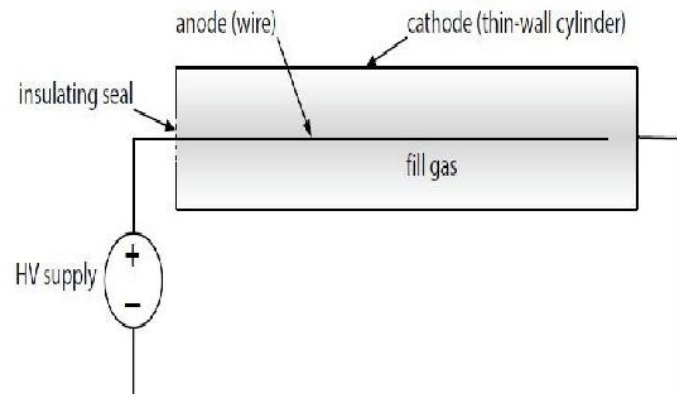
Vermogen = +1500W

1.2 Geiger counter

De reactorkamer zal ook een hoeveelheid X-stralen afgeven. Hiervoor heb ik mijn eigen Geiger counter gemaakt om dit te kunnen meten. De hoeveelheid stralingen zal niet aan gevaarlijke hoeveelheden komen maar is wel detecteerbaar.

1.2.1 Geiger Muller tube

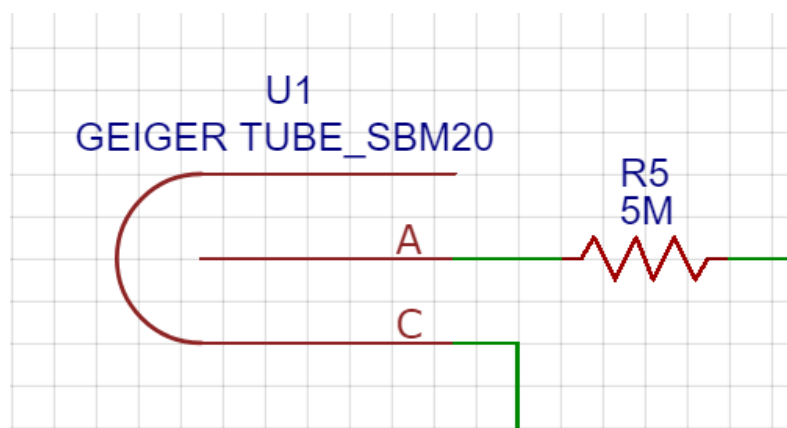
De detector zelf (Geiger muller tube) is een klein, hol afgesloten buisje met daarin een gas of mengeling van gassen die inert zijn. Meestal gebruikt men helium, argon en neon. Door het buisje loopt ook een kleine metalen staaf, geïsoleerd van de metalen buitenrand. Hierop staat een hoge spanning. Wanneer het gas geïoniseerd wordt, is er een overslag tussen de binnenstaaf en het buitenomhulsel en vloeit er een stroom. Die stroompulsen zijn de detectie van stralingen.



Basisschema geiger Muller tube

De tube die ik gebruik is een SBM20, dit is een oud Russisch model, dit maakt het goedkoop en het is zeker goed genoeg voor mijn toepassing.

Tussen de staaf van de tube en de 400V is een 5Mohm weerstand om de stroom te beperken. Een grotere weerstandswaarde zorgt voor een langere dead-time na elke pulse dat er geen detectie mogelijk is; een lagere weerstandswaarde zorgt voor meer slijtage. De 5Mohm waarde is aangeraden door het datasheet.



Deel schema geiger counter

1.2.2 Geiger counter hoogspanningsgenerator

De SBM20 heeft een spanning van 400V nodig. Hiervoor heb ik een kleine hoogspanningsgenerator gekocht van China die ik kan kalibreren tot op 400V.

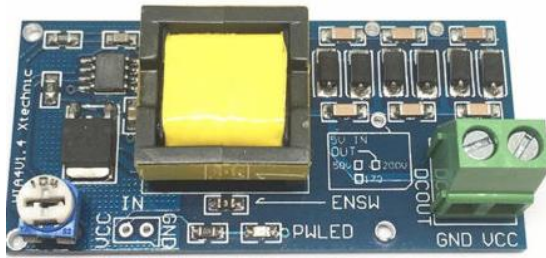
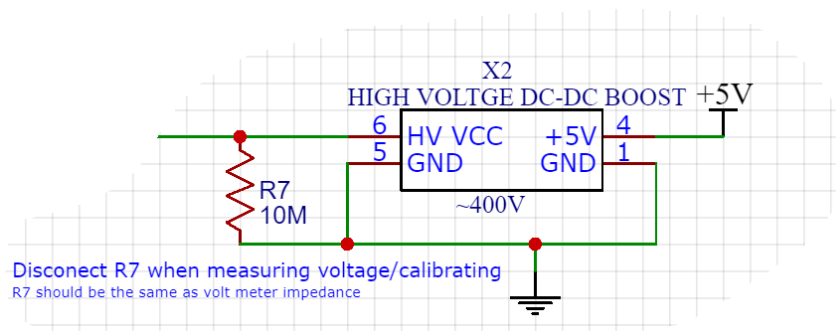


Foto hoogspanning generator

1.2.3 Geiger counter hoogspanningsgenerator kalibreren

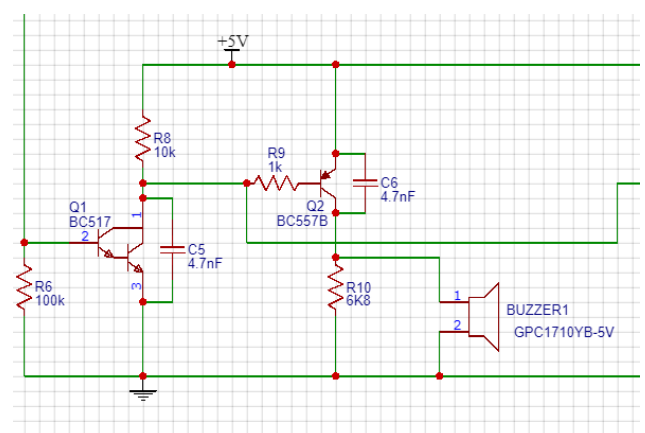
Het kalibreren gebeurt simpel door een potentiometer aan te passen met een voltmeter op de uitgang, Maar....! Doordat een voltmeter ook zijn eigen weerstandswaarde heeft van meestal 10Mohm, en deze generator helemaal niet krachtig is, zakt de spanning al bij het kalibreren. Daarom wordt na het kalibreren ook een 10Mohm weerstand parallel over de bron gezet.



Schema hoogspanning generator

1.2.4 Geiger counter puls detectie

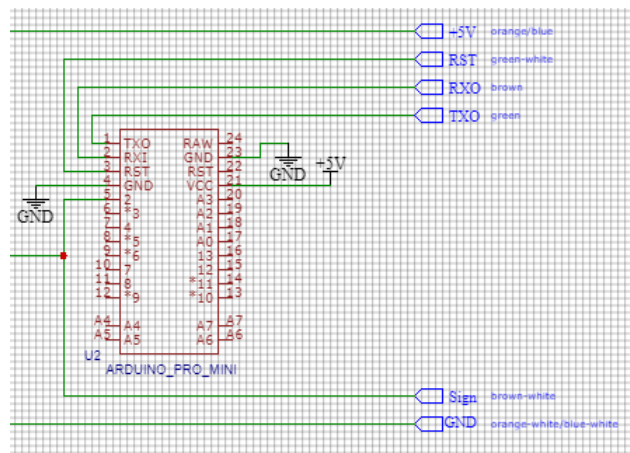
Het omhulsel van de geiger muller tube gaat naar een 100K weerstand zodat wanneer er een stroompuls is, via de spanningsverdeling een deel van de stroom de transistor aanstuurt. Het is een kleine stroom door de grote weerstand voor de geiger tube. Daarvoor wordt een darlington transistor gebruikt die de stroom 30.000 keer versterkt. Ik gebruik een bc517 wat een npn transistor is, hiervoor is een pul up weerstand zodat wanneer er een pulse is de spanning laag wordt getrokken. De Falling edge wordt doorgegeven naar een arduino mini voor het tellen, en naar een pnp (BC557B) transistor die het signaal omdraait en een hoog signaal geeft bij elke puls voor een buzzer. Deze kan een klik geven bij elke puls.



Schema puls detectie 1

De arduino mini krijgt een laag signaal bij elke pulse en gebruikt een interrupt om deze elke keer te kunnen dedecteren. Elke interrupt wordt geteld en elke 5 seconden worden de aantal gedetecteerde pulsen via seriële communicatie doorgegeven naar de raspberry pi.

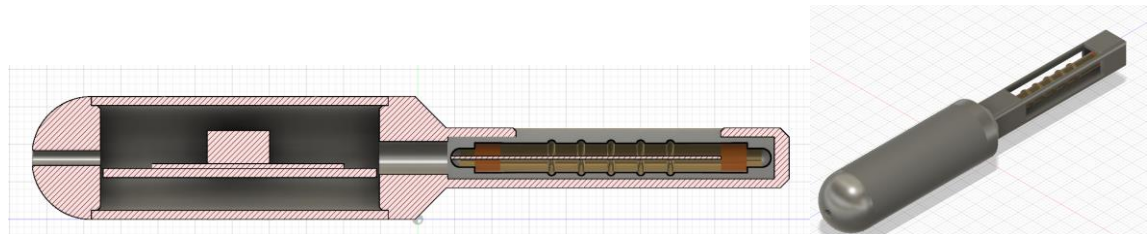
Hiervoor wou ik eerst een attiny85 gebruiken maar door softwareproblemen is dit niet gelukt. De attiny85 heeft namelijk maar één interrupt en geen hardwareseriële communicatie. De softwareserial library gebruikt de enige interrupt die ik niet meer kan gebruiken voor het detecteren van de pulsen.



Deel schema geiger counter

1.2.5 Geiger counter behuizing

Voor de geiger counter heb ik mijn eigen behuizing ontworpen in fusion 360 en met mijn eigen 3D printer in pla geprint.



Foto's geiger counter handvat design 1

De top waarin de geiger muller tube zit is hier goed beschermd en de aansluitingen zijn ook verborgen, maar de tube is niet helemaal verstopt om zo minder penetrerende stralingen te kunnen meten en de meting algemeen gevoeliger te maken.

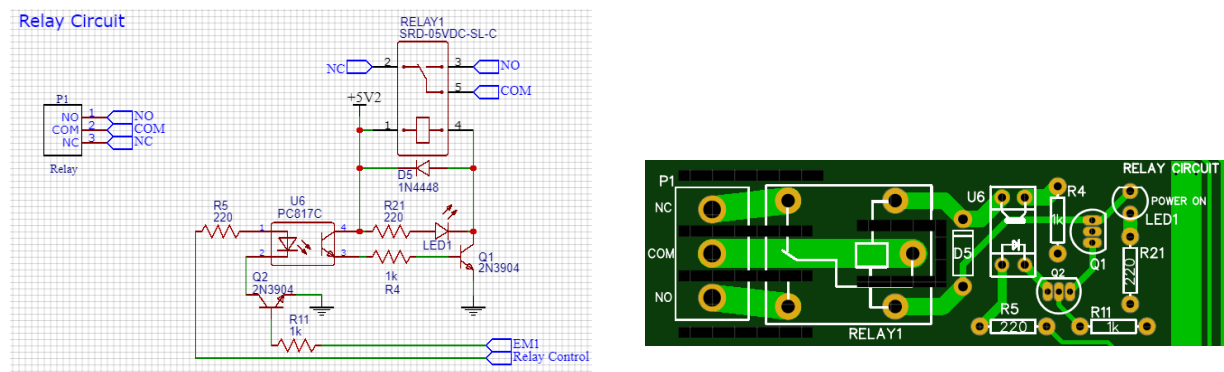
Het handvat is naar de grootte van de elektronica gemaakt zodat deze hier netjes inpast. Het is ook in een “fuzzy” mode geprint om ze het gevoel en grip van het handvat te verbeteren. Mijn fusor zal alleen meer X-stralen afgeven die ik kan dedecteren, maar mijn geiger counter kan ook gama en beta stralen dedecteren.

2 Hardware elektronica

Voor de verschillende metingen en de hoogspanningsbesturing te maken heb ik mijn eigen elektronica en pcb ontworpen en deze laten fabriceren in China bij JLCPCB. Het doet verschillende analoge metingen van spanningen, stromen en temperatuur. Het meten van vacuüm niveau en een relais schakelen voor de hoogspanning aan/uit te zetten en al deze data doorsturen naar een raspberry pi. Hierbij is veel aandacht besteed aan veiligheid met galvanische scheidingen, fatsoenlijke isolatie bij relais en een secundaire microcontroller die zijn eigen metingen doet parallel met een raspberry pi.

2.1 Relais schakeling

De relais (SRD-05VDC-SL-C) wordt veelgebruikt bij arduino's en dergelijke voor het schakelen van 230V. Het kan een stroom van 10A schakelen en ik zou hier net onder moeten blijven. De relais is optisch geïsoleerd van de raspberry pi, en slechts enkel als er een hoog signaal komt van de raspberry pi (relay control) en van de attiny85 (EM1), kan de relais schakelen. Op deze plaats van de pcb is geen ground plane gemaakt door de 230V die hier aanwezig is, en er zijn ook sloten in de pcb gemaakt tussen de 230V traces.



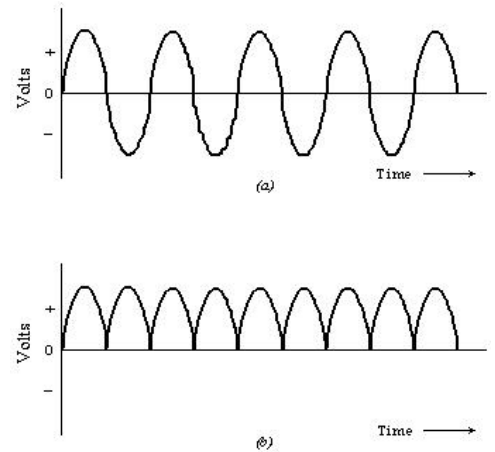
Schema relais schakeling

2.2 Spanning/stroom meting ingang

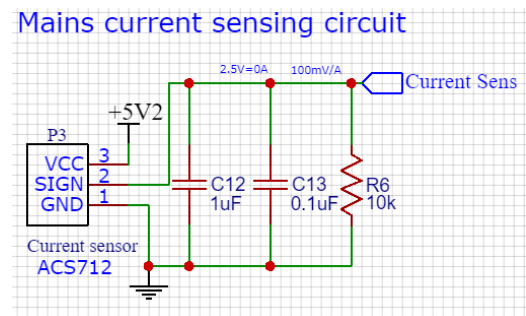
Extern wordt de spanning die uit de variac komt, de uitgang spanning van de hoogspanning generator en de ingang stroom van de hoogspanning generator gemeten.

Spanning meting ingang golf

De spanning van de variac wordt eerst verlaagd met een kleine transformator, wordt dan gelijkgericht met 4 diodes en dan wordt nog eens de spanning verlaagd met weerstanden en een potentiometer voor kalibratie. Hier zijn ook ontstoringcondensators met een weerstand die ervoor zorgt dat de condensators niet afvlakken. Dit zorgt voor een gelijkgerichte golf waar de faseverschuiving nog berekend kan worden als er al geweten is of die voor of na gaat ijlen. De faseverschuiving is wel niet afgemaakt in software door planningsproblemen en te weinig tijd om nog af te maken.

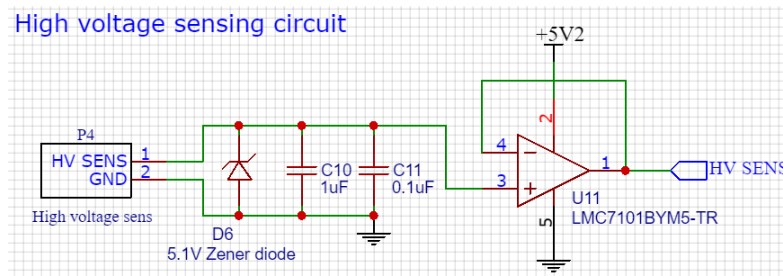


Voor het signaal van de stroommeter zijn ook ontstoringcondensators voorzien, en een weerstand om afvlakking met de condensators te voorkomen.



Stroommeting schema

Het signaal van de hoogspanningsmeting is ook voorzien op ontstoringcondensators en een zenerdiode om te voorkomen dat de spanning boven 5V komt om de ADC en attiny85 te beschermen. Maar door de grote impedantie bij de hoogspanningsverdeler is er reeds zeer snel een spanningsval bij zeer kleine stromen. Een de ADC (mpc3008) en de ADC van de attiny85 trekken bij elke meting wat stroom en dit deed de spanning en sterk zakken. Daarom is de zenerdiode weggelaten doordat die een hoge lekstroom heeft en is er een opamp tussen gezet die als spanningsvolger is opgesteld. Zo blijft de ingang impedantie bij de metingen constant en is deze ook hoger. Het kalibreren gebeurt dan volledig in hardware.



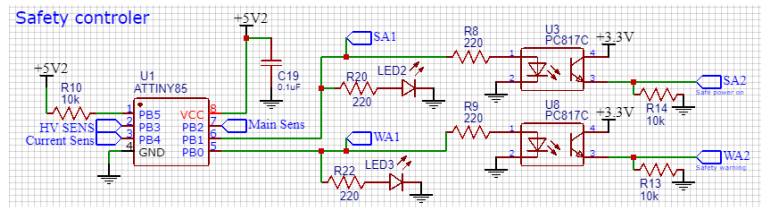
Hoogspanning meting ingang schema

2.3 Elektronica voedingsspanningen

Voor de elektronica componenten zijn er 3 rails. 2X 5V en 1X 10V. Er komt 5V van de raspberry pi die gebruikt wordt voor de buzzer en de capacitieve isolator van de ADC. Voor de veiligheid, indien er een fout optreedt, is er een secundaire geïsoleerde 5V spanningbron die de componenten voedt die verbonden zijn met de hoge spanningen componenten. De 5V secundaire komt simpel binnen via een usb micro aansluiting. De secundaire 5V voedt ook een DC-DC-converter die 10V aanmaakt wat gebruikt word bij de ADC spanningsreferentiediode. Hier zijn ook meerdere en grote condensatoren aangesloten wegens de grote hoeveelheid elektrische storingen.

2.4 Veiligheidscontrole

Indien de raspberry pi foute data binnenkrijgt of vastloopt en de hoogspanning niet kan afzetten met onveilige spanningen/stromen aanwezig, dan is er ook nog een back-up microcontroller. Een attiny85 microcontroller meet de ingang spanning en stroom van de hoogspanning generator en de uitgangspanning. Deze is geprogrammeerd om enkel een hoog signaal naar de relais te sturen indien de spanning-/stroomwaarden binnen parameters zijn. De maximumwaarden zijn hier hoger dan de raspberry pi omdat dit de back-up is indien de raspberry pi niet werkt naar behoren. Indien de spanning/stroom de maximumwaarde nadert, zal er ook een waarschuwingssignaal worden gegeven naar de raspberry pi. De waarschuwing en het veilig signaal zijn beide optisch geïsoleerd van de raspberry pi.



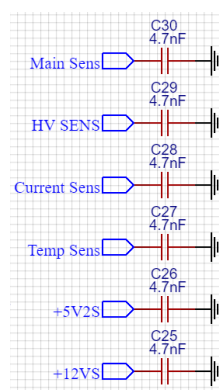
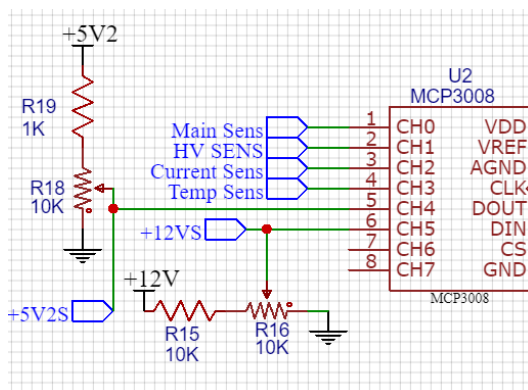
Veiligheid controller schema

2.5 MCP3008 ADC

Omdat de raspberry pi geen interne ADC heeft heb ik een ADC3008 gebruikt om de analoge signalen digitaal door te geven naar de raspberry pi.

2.5.1 Metingen

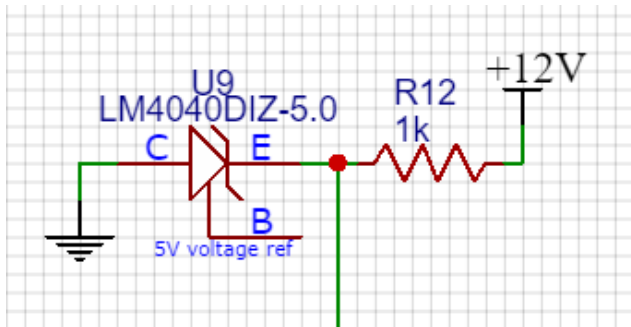
De ADC meet de ingang spanning/stroom van de hoogspanningsgenerator en de uitgangspanning, de temperatuur van de hoogspanningsgenerator en de voedingspanningen van de elektronica. De voedingspanningen worden verlaagd met een weerstand, een spanningverdeler en een potentiometer voor de kalibratie. Elke analoge ingang heeft ook zijn eigen ontstoringcondensator, bij elke meting vloeit er ook een stroom en dit kan de spanning doen zakken wat voor onnauwkeurige meting zorgt. De condensators dienen dan ook voor dit effect te minderen.



ADC gedeeltelijk schema

2.5.2 Spanningsreferentie

Er is ook een 5V spanningsreferentie aanwezig voor betere nauwkeurigheid van de metingen. Dit bestaat uit een zenerdiode die de 10V van de DC-DC converter accuraat verlaagt tot 5V, met een weerstand om de stroom te beperken door de diode. Bij temperatuursverandering zal de spanning ook minimaal variëren. Er is ook een kleine offset aanwezig van 4mV die in software is bijgerekend voor nog betere nauwkeurigheid. (LM4040DIZ-5.0 datasheet in bijlagen)



Spanning referentie schema

2.5.3 SPI

Voor communicatie tussen de raspberry pi en de ADC wordt het SPI-protocol gebruikt. Dit werkt mij een verbinding van 4 draden.

SCLK: clock frequentie

MOSI: Master Out Slave In = data van de raspberry pi naar de adc

MISO: Master In Slave Out = data van ADC naar de raspberry pi

SS: Select = duidt aan voor welk apparaat de communicatie is als er meerdere zijn, hier is er enkel de adc

Het exacte protocol en hoe het gebruikt wordt is uitgelegd bij de software.

2.5.4 SPI Isolatie

Tussen de ADC(mcp3008) en de raspberry pi is ook een capacitieve isolator (Si8441BB). Deze heeft een dubbele functie. Ten eerste als isolatie zoals bij optocouplers (deze zijn te traag voor communicatie) om de raspberry pi te beschermen als er spanningspieken van de mcp3008 zouden optreden door een fout. De tweede functie is de 5V-spanningscommunicatie omzetten naar 3.3V voor de raspberry pi en

visa versa. De Si8441BB is hiervoor gekozen omdat deze perfect geschikt is voor SPI communicatie en isolerend. (Si8441BB datasheet in bijlagen)

2.6 Vacuümsensor

De vacuümsensor dient niet om het niveau ervan te meten maar om te detecteren of de druk laag genoeg is zodat men de reactorkamer niet kan openmaken om te vermijden dat de hoogspanning genaakbaar wordt.

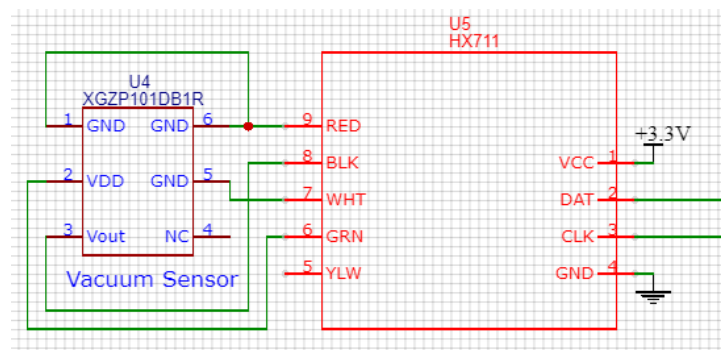
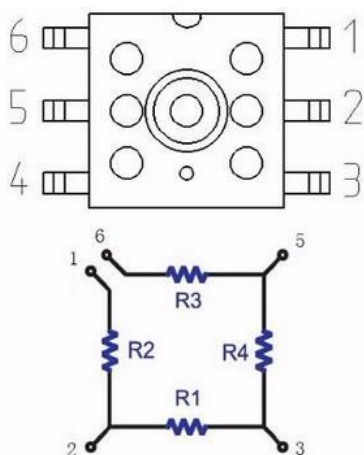
2.6.1 Druksensor

De sensor is een XGZP101DB1R, deze kan ruwweg het drukniveau meten maar is niet geschikt voor het nauwkeurig meten van de zeer lage druk die de vacuüm pomp creëert.

2.6.2 Load cell amplifier

De HX711 is een load cell amplifier die meestal gebruikt wordt om gewicht te meten met load cells, maar doordat deze hetzelfde principe gebruiken heb ik het ook gebruikt voor mijn drukmeting.

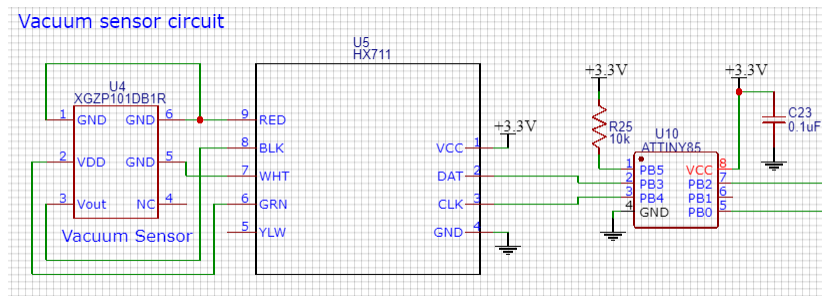
Het werkt op het principe dat weerstanden van waarde veranderen als deze druk belast worden. Bij deze sensor gebeurt dat omdat de weerstanden die op een membraam zitten dat van vorm verandert bij een drukverandering. Deze verandering is zeer klein waardoor er een speciale versterker voor nodig is, namelijk dus de HX711.



Schema vacuüm sensor en HX711

2.6.3 HX711 communicatie

Door problemen met het communicatieprotocol nodig om met de HX711 uit te werken heb ik een extra Attiny85 gebruikt. Deze heeft de nodige library om hiermee te communiceren.



Schema vacuum sensor en HX711

Door problemen met mijn vacuümpomp en de retour daarvan had ik geen tijd om deze sensor te kalibreren en gebruik ik deze nu enkel om te meten of er een onderdruk aanwezig. (code in bijlagen)

3 Software

Voor de werking van het geheel is er code geschreven voor drie microcontrollers: twee keer Attiny85 (safety controller, vacuüm sensor) en één keer arduino mini (geiger counter) in arduino. Voor een raspberry pi 3A+ in C# met een windows form en console mogelijkheden.

3.1 Attiny85 safety controller code

De code voor de safety controller is basic gehouden, het moet enkel drie ADC berekeningen maken, kijken of die nog binnen bepaalde waarden zijn. Zo niet, dan wordt er eerst een waarschuwingssignaal gestuurd naar de raspberry pi. Buiten de veilige waarden is er geen signaal meer naar de raspberry pi en de relais wordt aangestuurd waardoor de relais afschakelt.

3.2 Attiny85 Vacuüm sensor code

De code voor de vacuüm sensor was bedoeld om enkel een ruwe meting te doen voor detecteren of er een onderdruk aanwezig was dat ervoor zorgt dat de hoogspanning niet kan aanstaan wanneer er geen onderdruk aanwezig is. Dit zorgt ervoor dat de hoogspanning enkel kan aanstaan als de vacuüm kamer gesloten is met een onderdruk.

Door problemen met mijn vacuümpomp en het zeer late ontvangen van een nieuwe pomp heb ik niet voldoende tijd gehad om dit volledig te laten werken.

3.3 Arduino mini geiger counter code

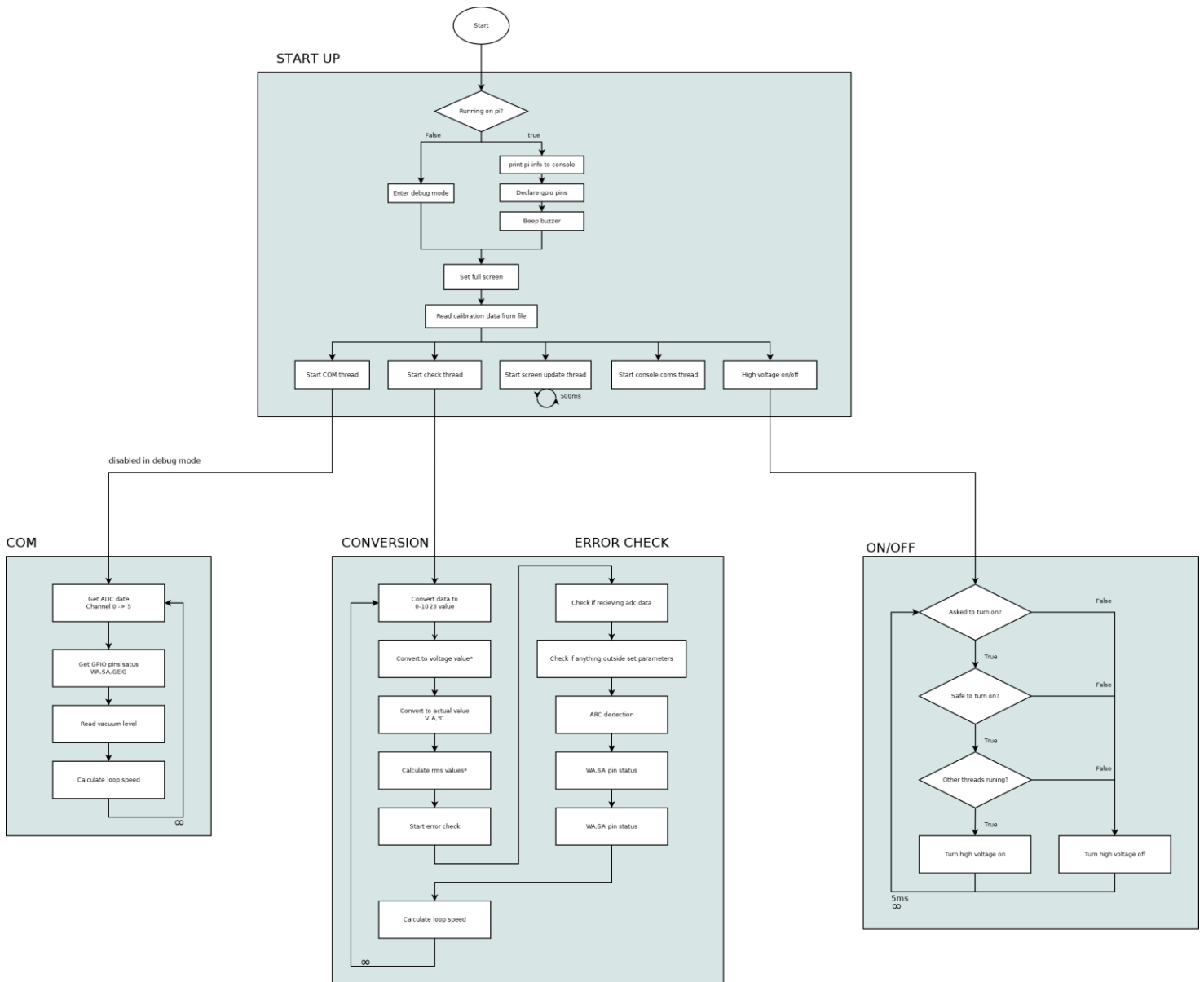
De code voor de geigercounter is ook eenvoudig . Het heeft een Interrupt wat betekent dat als het signaal op de puls pin van hoog naar laag verandert er een code voorrang krijgt op de loop code. De interrupt zal een getal met één waarde vermeerderen voor elke puls. Elke 15 seconden wordt dat getal terug op nul gezet en wordt via seriële communicatie die waarde eerst doorgestuurd naar de raspberry pi.

De counts per minute waarde kan dan simpelweg berekend worden door de pulsen van de laatste 15 seconden maal 4 te vermenigvuldigen. Voor de sbm-20 kan je dan het aantal uSv/h berekenen door het te vermenigvuldigen met 0.0057. (zie de raspberry pi code)

3.4 Raspberry pi C# code

Meestal wordt voor raspberry pi's code geschreven in python. Maar omdat ik een GUI wou maken, wat in python niet gemakkelijk is, heb ik gekozen te werken met visual studio en C# . Hiermee heb ik meer ervaring. Ik heb dan ook gekozen het in C# te gebruiken, maar.C# en .NET code werken standaard niet op een raspberry pi noch met linux. Ik heb een programma gevonden dat dit wel mogelijk maakt: Mono. Dit maakt het dan mogelijk C# code op linux te gebruiken, en dan met UnoSquare libraries kan ik de GPIO pinnen sturen en de communicatieprotocollen gebruiken zoals SPI en serieel.

Doordat de code wel langer dan 1600 lijnen is geworden heb ik daar ook een blokschema voor gemaakt om het overzichtelijk te maken. Met behulp van het blokschema kan ik de belangrijke code tonen en uitleggen.



3.4.1 C# startup

Bij het opstarten probeert de raspberry pi een beetje code. Als dit slaagt, zal al de informatie van de raspberry pi in de console worden geschreven. Indien dit faalt, verschijnt het in de console en wordt het ook aangeduid in de rest van het programma doordat delen code alleen werken wanneer het runt op een raspberry pi. Ook worden hier de locaties van de tekstbestanden bepaald omdat op windows “\” en Linux “/” gebruikt bij de onderverdeling. Dit is alleen gedaan om het ook te kunnen runnen op een windows computer voor het maken en het testen van het programma.


```

//check if running on raspberry pi
try
{
    //print raspberry pi info to console
    string Info = Pi.Info.ToString();
    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine("Raspberry pi environment detected");
    Console.WriteLine(Info);
    Console.WriteLine("RUN MODE");
    Console.ResetColor();

    //set location of settings file
    SettingsFile = @"DATA/Settings data.txt"; //Because it couldnt be the same for windows and linux could it!?
    ThermistorTableFile = @"DATA/100K thermistor table.txt";

    //SET buzzer PINOUT
    var buzzerPin = Pi.Gpio[WiringPiPin.Pin01]; //bcm18
    buzzerPin.PinMode = GpioPinDriveMode.Output;

    //beep buzzer
    buzzerPin.Write(true);
    Thread.Sleep(1000);
    buzzerPin.Write(false);

    Pi.Gpio.Pin01.Write(true);

    //make clear program is running normal on raspberry pi
    RunningOnRaspberry = true;
    label1.ForeColor = System.Drawing.Color.Green;
    label1.Text = ("RUN MODE");
}
catch
{
    //make clear program is not fully functional because not no raspberry pi hardware is found
    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine("Outside raspberry pi environment");
    Console.WriteLine("Running in DEBUG MODE");
    Console.ResetColor();
    Console.ForegroundColor = ConsoleColor.White;

    //set location of settings file
    SettingsFile = @"DATA\Settings data.txt"; //fucks sake
    ThermistorTableFile = @"DATA\100K thermistor table.txt";

    RunningOnRaspberry = false;
    label1.ForeColor = System.Drawing.Color.Red;
    label1.Text = ("DEBUG MODE");
}

```

Deel code bij opstart

3.4.2 C# Dataopslag ophalen

Er is een tekstdocument met alle belangrijke informatie die het programma nodig heeft voor de werking.

Om nauwkeurig te kalibreren worden de waarden van de elektronica nagemeten met een nauwkeurige multimeter (0.3%) en kan zo een vermenigvuldigingsfactor berekend worden wat in de kalibratiedata wordt gezet. Samen met de actuele spanning van de spanning referentie kunnen de spanning-/stroomwaarden zeer accuraat berekend worden.

De minimum- en maximumwaarden worden ook in dit bestand gezet en zijn zo ook gemakkelijk te veranderen

```
ADCreferenceVoltage=5.0035
```

```
//first digit always place in array  
//Second digit, divide input voltage by voltage at adc pin  
CalMainsVoltage=0,1  
CalMainsCurrent=1,1  
CalHVsens=2,1  
CalTemp=3,1  
Cal+5V2=4,1.9898060765340660215824473380321  
Cal+12V=5,2.494251724
```

```
//Parameters  
  
ParSpeedDelay=5  
  
ParMainsVoltageMIN=-1  
ParMainsVoltageLOW=-1  
ParMainsVoltageHIGH=100  
ParMainsVoltageMAX=200  
  
ParMainsCurrentMIN=-10  
ParMainsCurrentLOW=-7  
ParMainsCurrentHIGH=8  
ParMainsCurrentMAX=10  
  
ParHighVoltageHIGH=7000  
ParHighVoltageMAX=10000  
  
ParTempMIN=0  
ParTempLOW=10  
ParTempHIGH=70  
ParTempMAX=80  
  
ParV5MIN=4.5  
ParV5LOW=4.7  
ParV5HIGH=5.35  
ParV5MAX=5.7  
  
ParV12MIN=6  
ParV12LOW=8  
ParV12HIGH=13  
ParV12MAX=15  
  
ParARCVoltage=200
```

```
//Get ADC reference voltage  
if (Line.Contains("ADCreferenceVoltage="))  
{  
    ADCreferenceVoltage = Convert.ToSingle(Line.Substring(Line.LastIndexOf("=") + 1, Line.Length - Line.LastIndexOf("=") - 1));  
}  
  
//read calibration data  
if (Line.Contains("Cal"))  
{  
    CalibrationData[Convert.ToInt16(Line.Substring(Line.LastIndexOf("=") + 1, 1))] =  
        Convert.ToSingle(Line.Substring(Line.LastIndexOf(",") + 1, Line.Length - Line.LastIndexOf(",") - 1));  
}  
  
//read MIN MAX parameters data  
if (Line.Contains("Par"))  
{  
    //par mains voltage  
    if (Line.Contains("ParMainsVoltageMIN"))  
    {  
        MainsVoltageMIN = Convert.ToSingle(Line.Substring(Line.LastIndexOf("=") + 1, Line.Length - Line.LastIndexOf("=") - 1));  
    }  
    else if (Line.Contains("ParMainsVoltageLOW"))  
    {  
        MainsVoltageLOW = Convert.ToSingle(Line.Substring(Line.LastIndexOf("=") + 1, Line.Length - Line.LastIndexOf("=") - 1));  
    }  
    else if (Line.Contains("ParMainsVoltageHIGH"))  
    {  
        MainsVoltageHIGH = Convert.ToSingle(Line.Substring(Line.LastIndexOf("=") + 1, Line.Length - Line.LastIndexOf("=") - 1));  
    }  
    else if (Line.Contains("ParMainsVoltageMAX"))  
    {  
        MainsVoltageMAX = Convert.ToSingle(Line.Substring(Line.LastIndexOf("=") + 1, Line.Length - Line.LastIndexOf("=") - 1));  
    }  
}
```

Deel code data invoer

Settings file 1

3.4.3 C# threads

Om het programma sneller bepaalde delen geïsoleerd te laten werken, worden er meerdere threads aangemaakt. Threads zijn delen code die op zichzelf werken en niet op andere delen code moeten wachten. Zo is er een thread voor de hardwarecommunicatie, het checken van alle data, het scherm updaten, console commands en voor besturing van hoogspanningrelais.

```
//start screen update thread  
Thread screenUpdateThread = new Thread(screenUpdate);  
screenUpdateThread.IsBackground = true;  
screenUpdateThread.Start();  
  
Console.WriteLine("ScreenUpdate thread started");
```

Voorbeeld starten van thread

3.4.4 C#-communicatie

De communicatie thread moet via SPI de ADC data verkrijgen, serieel de data van de geiger counter. Er wordt gekeken worden welke pinnen hoog/laag zijn van de safetycontroller en de vacuüm sensor.

3.4.4.1 ADC SPI

De communicatie met de ADC (MCP3008) via SPI was wat moeilijker om uit te werken. Via arduino is dat zeer gemakkelijk via een library, maar ik vond geen werkende library om dat in C# te doen met de MCP3008. Daarom heb ik het zelf uitgezocht hoe ik hier exact mee moet communiceren en mijn data verkrijgen van de juiste kanalen. Dit heb ik kunnen uitzoeken met behulp van de datasheet van de MCP3008.

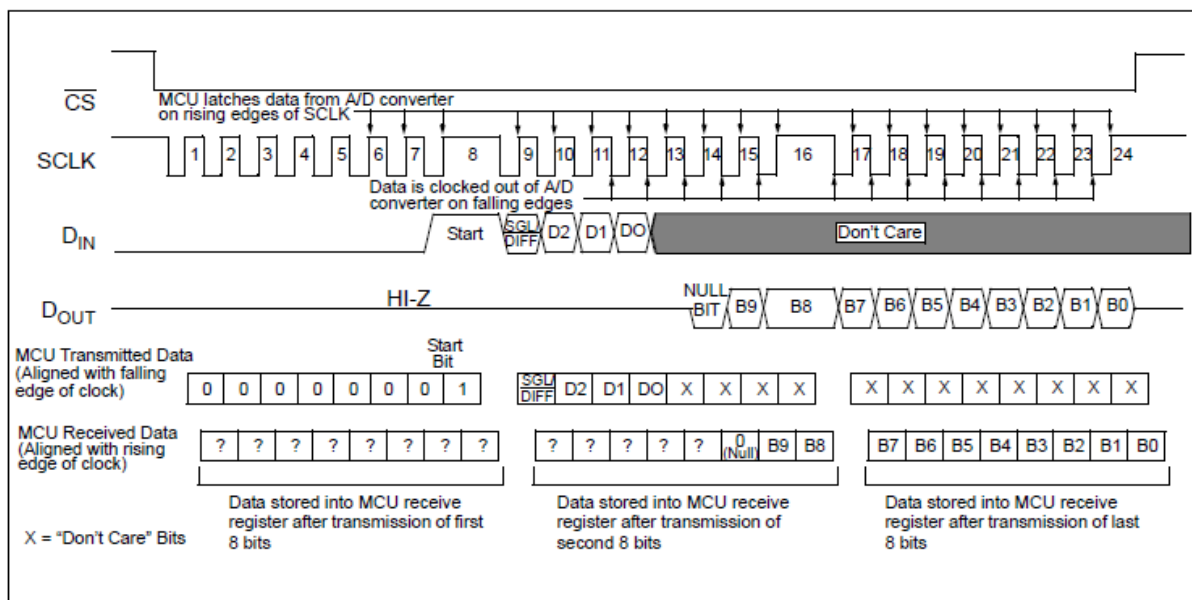


FIGURE 6-2: SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 1,1: SCLK idles high).

Daar was een diagram, hoe te communiceren met de mcp3008, een microcontroller met 8bit segmenten. Ik kon er dan uithalen welk getal overeenkwam met het juiste kanaal en hoe ik dan terugkomende data ontving.

```

/* for more information
 * mcp3008 datasheet 6.1
 * https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf#G1.1036393
 *
 * bin comm example (ring buffer)
 * 0000 0001 0000 XXXX XXXX XXXX (sent data single channel)
 * ??? ???? ???? ???? 0000 0000 (received data)
 * (C=channel select)
 * (D=adc data 0-->1024)
 * (X=dont care bits)
 * (?=0)
 */

//note! data clocked out at rising edge en clocked in at falling edge

//128 +16 per channel
byte[] mosiData = { 1, 128, 0 }; //CH0

while (true)
{
    Stopwatch SPIstopWatch = new Stopwatch();
    Int32 SPITicks = 0;

    if (RunningOnRaspberry)
    {
        SPIstopWatch.Start();

        //set frequency in hertz (500Khz min / 32Mhz max) (3.6Mhz on +5V / 1Mhz on +3.3V max freq for mcp3008)
        Pi.Spi.Channel0.Frequency = 1000000;

        mosiData[1] = 128; //CH0
        MainsSens = Pi.Spi.Channel0.SendReceive(mosiData);

        mosiData[1] = Convert.ToByte(mosiData[1] + 16); //CH1
        HVsens = Pi.Spi.Channel0.SendReceive(mosiData);

        mosiData[1] = Convert.ToByte(mosiData[1] + 16); //CH2
        CurrentSens = Pi.Spi.Channel0.SendReceive(mosiData);

        mosiData[1] = Convert.ToByte(mosiData[1] + 16); //CH3
        TempSens = Pi.Spi.Channel0.SendReceive(mosiData);

        mosiData[1] = Convert.ToByte(mosiData[1] + 16); //CH4
        VCC52 = Pi.Spi.Channel0.SendReceive(mosiData);

        mosiData[1] = Convert.ToByte(mosiData[1] + 16); //CH5
        VCC12 = Pi.Spi.Channel0.SendReceive(mosiData);
    }
}

```

C# code SPI communicatie

Hier wordt zo de data per kanaal opgevraagd en wordt het in een 2 byte array gezet wat verder verwerkt wordt in de conversie en check thread.

3.4.4.2 Safety controller pin status

Hier worden twee booleanse waarden veranderd naargelang het signaal op de pin hoog/laag is.

```

//get wa and sa pin status
saStatus = Pi.Gpio.Pin21.Read();
waStatus = Pi.Gpio.Pin22.Read();

```

3.4.5 C# Conversie/checksum

Conversie van de data van de ADC gebeurt in drie stappen. Eerst wordt de data van de 2byte array in een 16bit integer gezet, dan wordt dit in actuele spanning omgevormd (behalve bij temperatuurmeting) en wordt het omgevormd met de kalibratie data naar de echte waarde.

3.4.5.1 C# Conversie/checksum spanning/stroom en temperatuur

```
//Mains voltage conversion
ConversionData[ConversionChannel, 0] = MainsSens[2] + (MainsSens[1] * 256);
ConversionData[ConversionChannel, 1] = ConversionData[ConversionChannel, 0] * (ADCreferenceVoltage / 1023);

MainsVoltage = ConversionData[ConversionChannel, 1] * (CalibrationData[ConversionChannel]);

ConversionChannel++;
```

Code spanningberekening

Hier worden de stappen uitgevoerd zoals hierboven uitgelegd. De actuele spanning berekenen is een eenvoudige berekening.

```
//Current conversion
//+ Offset calc
ConversionData[ConversionChannel, 0] = CurrentSens[2] + (CurrentSens[1] * 256);
ConversionData[ConversionChannel, 1] = ConversionData[ConversionChannel, 0] * (ADCreferenceVoltage / 1023);

MainsCurrent = Convert.ToSingle((ConversionData[ConversionChannel, 1] - 2.5) / 0.1);
```

Code stroomberekening

De stroomberekening is een beetje anders doordat de nulwaarde 2.5V (5V ref) is en de stroom 1A per 100mv is.

Maar doordat dit een wisselspanning is, moet de RMS spanning worden berekend . Dit wordt gedaan door alle waarden voor 40ms op te slaan in een list en de maximumwaarde in die lijst te vermenigvuldigen met de vierkantswortel van 2.

```

//RMS CALC
//add data to list
ListMainsVoltage.Add(MainsVoltage);
ListMainsCurrent.Add(MainsCurrent);
// Get the elapsed time as a TimeSpan value.
TimeSpan ts2 = RMStimingStopWatch.Elapsed;
//capture data for 40ms (2 50hz ac cycles) to calculate rms then throw out to capture next run
if (ts2.Milliseconds >= 40)
{
    RMStimingStopWatch.Reset();

    float MaxVoltage = ListMainsVoltage.Max();
    //float MinVoltage = ListMainsVoltage.Min();
    float MaxCurrent = ListMainsCurrent.Max();
    //float MinCurrent = ListMainsCurrent.Min();

    rmsVoltage = Convert.ToSingle(MaxVoltage / Math.Sqrt(2));
    rmsCurrent = Convert.ToSingle(MaxCurrent / Math.Sqrt(2));

    ListMainsVoltage.RemoveRange(0, ListMainsVoltage.Count);
    ListMainsCurrent.RemoveRange(0, ListMainsCurrent.Count);
}

```

Code rms berekening

Bij het berekenen van de temperatuur wordt geen actuele spanning berekend aangezien het niet nodig is. Online vond ik een lijst met veel voorbeelden van welke temperatuur overeenkomt met de weerstandswaarde. Door twee willekeurige punten te pakken kan je een formule uitwerken ($Y2-Y1/X2-X1$) Hiermee kwam ik dan op $Y=-10/36509X + 1912725/36509$. Waarbij Y de temperatuur is en X de thermistor weerstand. In de code is de formule verder uitgewerkt maar het betekent exact hetzelfde.

```

//Temperature sensing conversion
ConversionData[ConversionChannel, 0] = TempSens[2] + (TempSens[1] * 256);
//ConversionData[ConversionChannel, 1] = ConversionData[ConversionChannel, 0] * (ADCReferenceVoltage / 1023); //Voltage out of thermistor divider not used.
ThermistorResistance = (TempResistor / (1023 / ConversionData[ConversionChannel, 0] - 1)) / 10; //calculate resistance of thermistor
OilTemperature = Convert.ToSingle(-0.000273905 * ThermistorResistance + 52.390506450464269084335369360979); //Calculate temperature with equation

```

Code voor het berekenen van temperatuur 1

3.4.5.2 Checksum

Elke keer dat de data wordt omgevormd, wordt ook gekeken of alle waarden nog wel binnen veilige waarden zijn. Indien er iets buiten waarden valt, wordt dit in een lijst gezet om op het GUI weer te geven. Het wordt op de console omgezet met een tijdstip van de error en indien het een belangrijke waarden heeft, wordt aangeduid dat het niet veilig is om de hoogspanning aan te zetten.

```

//ERROR 1
if (VS > VS MAX)
{
    if (!ErrorList.Contains("5V2 too high"))
    {
        ErrorList.Add("5V2 too high");
        tsTime = Time.Elapsed;
        Console.WriteLine("5V2 too high " + tsTime.TotalSeconds);
    }
    HVsafe = false;
}
else if (VS > VS HIGH)
{
    if (!ErrorList.Contains("5V2 high"))
    {
        ErrorList.Add("5V2 high");
        tsTime = Time.Elapsed;
        Console.WriteLine("5V2 high " + tsTime.TotalSeconds);
    }
}
else if (VS < VS MIN)
{
    if (!ErrorList.Contains("5V2 too low"))
    {
        ErrorList.Add("5V2 too low");
        tsTime = Time.Elapsed;
        Console.WriteLine("5V2 too low " + tsTime.TotalSeconds);
    }
    HVsafe = false;
}
else if (VS < VS LOW)
{
    if (!ErrorList.Contains("5V2 low"))
    {
        ErrorList.Add("5V2 low");
        tsTime = Time.Elapsed;
        Console.WriteLine("5V2 low " + tsTime.TotalSeconds);
    }
}
}

```

Code binnen parameters check van 5V

Een controle die ik heb bijgevoegd is de mogelijke overslag. Dit heb ik toegevoegd nadat ik bij een test van de reactor een te laag vacuüm vaststelde. Hierdoor ontstond een overslag. Dit kan zeer gevaarlijk zijn; wanneer dit gebeurt ontstaat een grote stroom getrokken en daalt de spanning hierdoor. Om dit te detecteren, wordt er in mijn programma elke 100ms gekeken of de spanning te veel is gezakt ten opzichte van de laatste meting. Indien dit zo is, kan er overslag plaatsvinden en wordt de hoogspanning uitgezet.

```

//ERROR 8
TimeSpan ts = ARCTimingStopWatch.Elapsed;
//Check every 100ms for sudden voltage drop wich can indicate a high voltage arc
if (ts.Milliseconds >= ARCTime)
{
    ARCTimingStopWatch.Reset();
    if (HVlast >= HighVoltage + ARC)
    {
        if (!ErrorList.Contains("Possible ARC dedected"))
        {
            ErrorList.Add("Possible ARC dedected");
            tsTime = Time.Elapsed;
            Console.WriteLine("Big voltage drop on high voltage line dedected - posible ARC! " + tsTime.TotalSeconds);
        }
        HVsafe = false;
    }
    HVlast = HighVoltage;
}
}

```

Code voor overslag dedectie

Hier worden alle spanningstroomwaarden en de temperatuur gecheckt, en de status van de safety controller en of de snelheid van de conversie en het controleren wel snel genoeg werkt. Dit moet minsten een paar duizend keer per seconde plaatsvinden voor een nauwkeurige meting (vooral voor de wissel spanning/stroom).

```
//ERROR 11
tsTime = Time.Elapsed;
if (ConversionTickcount < 10000 && tsTime.TotalSeconds > SpeedDelay)
{
    if (!ErrorList.Contains("Low speed"))
    {
        ErrorList.Add("Low speed");
        tsTime = Time.Elapsed;
        Console.WriteLine("Thread low speed error (" + ConversionTickcount + "/s) " + tsTime.TotalSeconds);
        HVsafe = false;
    }
}
```

Code voor snelheid check

4 Fusieproces en de reactorkamer

De bedoeling van dit project is het tonen van een fusieproces. Dit doe ik dit op basis van een elektrisch veld, een hoge negatieve spanning dat een plasma gaat maken en de protonen aantrekt. Dit proces vindt plaats in een grid. Hier krijgen ze kans om tegen elkaar te botsen en als ze genoeg energie hebben, zullen ze fuseren tot een nieuw element.

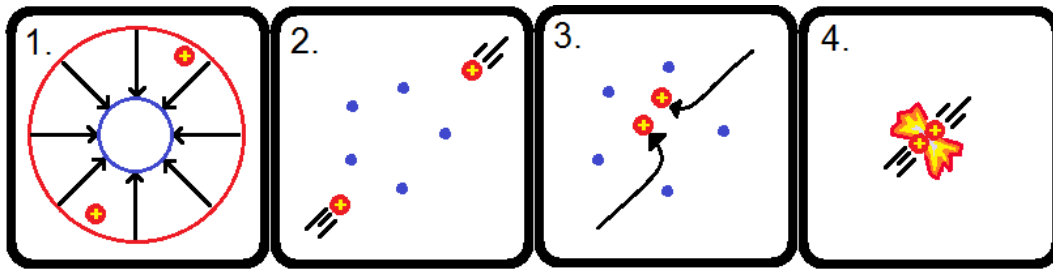
4.1 Basisprincipe fusie met een elektrisch veld

Figuur 1: De buitenste cirkel is het glas, neutraal geladen. Hierin is een vacuüm gecreëerd, wat ervoor zorgt dat er nog maar een beperkt aantal atomen aanwezig zijn. De binnenste cirkel is van metaal waarop een zeer grote negatieve elektrische spanning wordt gezet. Hier zijn veel elektronen aanwezig zijn. De protonen van de atomen worden hierdoor aangetrokken tot de middelste cirkel, maar de elektronen die bij de atomen aanwezig zijn worden hiervan weggestoten. Uiteindelijk zullen de atomen worden gescheiden in protonen en elektronen en creëert men een plasma.

Figuur 2: Nu de protonen en elektronen van elkaar gescheiden zijn, kunnen de protonen verder naar het elektrisch geladen metaal. Dit metaal is een bol gevormd met draden. Op de doorsnede is te zien dat er veel plaats tussen de draden zit en de protonen kans hebben om daardoor te vliegen wanneer ze van de juiste kant komen. Veel zullen tegen de draden botsen, wat verloren energie is en dit dus ook zo een inefficiënt design maakt.

Figuur 3: Wanneer de protonen tussen de draden door gaan, hebben ze heel wat snelheid en energie opgebouwd. Wanneer dit niet genoeg is, zullen ze andere protonen wel tegenkomen, maar stoten ze elkaar nog te veel af. Dit is vergelijkbaar met twee magneten met dezelfde polen. Als je deze tegen elkaar wilt duwen, kost het heel wat energie.

Figuur 4: Wanneer je echter wel genoeg energie hebt om deze afstotingskracht te overkomen, zullen de protonen fuseren tot een nieuw element.

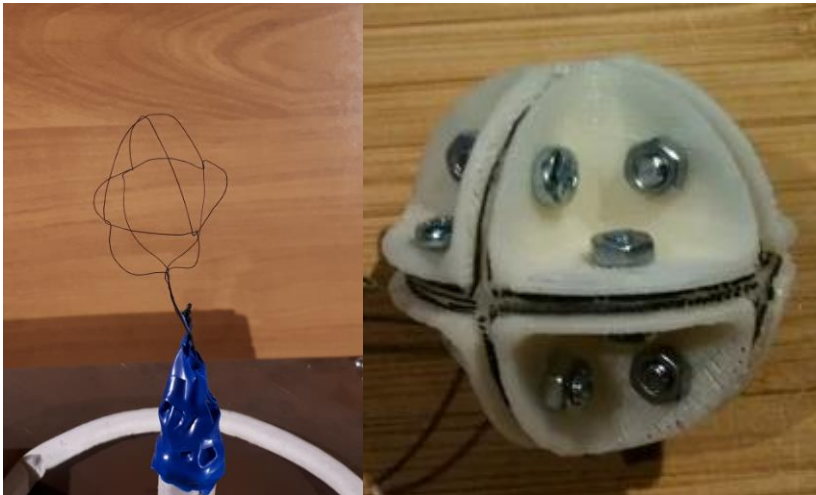


- "Protonen" correcte benaming is nucleus als er neutronen enz aanwezig zijn. Maar dit is maar een simpel voorbeeld om het proces proberen uit te leggen.

4.2 Reactor grid

Zoals in vorige uitleg werd gezegd, is er een metalen grid van draden nodig voor de werking. Doordat dit zeer hoge temperaturen kan tegenkomen, kan dit niet zomaar met elk metaal worden gedaan; de smeltemperatuur moet hoog genoeg zijn. Koper smelt al bij iets meer dan 1000C en ijzer smelt al bij 1500C. Veel andere metalen komen hier niet ver boven. Hiervoor heb ik het element gebruikt dat het hoogste smeltpunt heeft van het hele periodiek systeem, namelijk wolfram of Tungsten (Engelse benaming). Dit smelt pas bij een temperatuur van meer dan 3400C!

Om de vorm te verkrijgen heb ik een soort mal online gevonden en in 3D geprint. Hier kan ik mijn wolframdraad omheen draaien tot de juiste vorm en daarna de mal uit elkaar halen.



Wolfram grid

Grid mal

4.3 Reactorkamer

Mijn reactorkamer bestaat uit een metalen plaat die geaard is en waar de elektronica zich op bevindt. De reactorkamer is gemaakt van een glazen stolp luchtdicht op de plaat. Deze vorm wordt gebruikt omdat het ideaal is om drukverschil te weerstaan. Er is een aansluiting voor een silicone tube in de plaat die naar de vacuümmeter gaat die dan verder aansluit aan de vacuümpomp. De hoogspanning komt langs onder binnen en is extra geïsoleerd met een pvc-buis en. Aan de hoogspanningsdraad is dan het grid bevestigd.

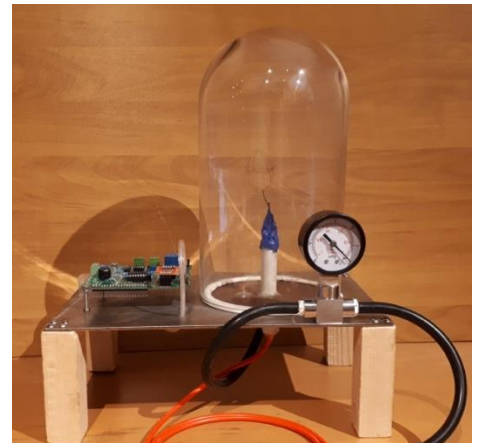
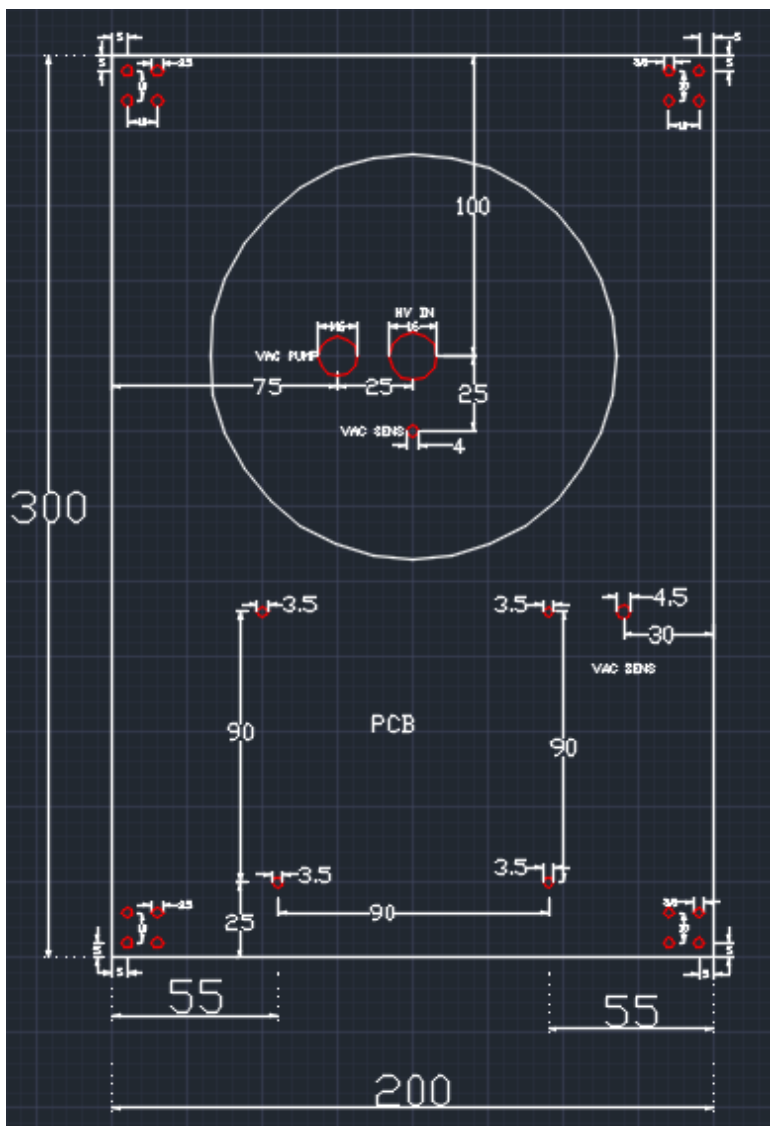


Foto reactor en elektronica



Schema reactor metalen plaat boorgaten

4.4 Stralingen

De wolk van protonen en elektronen (plasma) in de reactorkamer zullen rondbewegen en zullen energie verliezen in de vorm van elektromagnetische stralingen. Deze zullen in het zichtbaar spectrum zijn en als blauw licht te zien zijn, ultraviolet en in de vorm van X-Stralen. De hoeveelheid X-Stralen is niet gevaarlijk, toch niet voor blootstelling op lange duur.

Indien er daadwerkelijk een fusiereactie is zullen er ook neutronenstralen zijn. Fusors kunnen dus ook gebruikt worden als neutronbron voor experimenten en onderzoek, en ook voor de medische wereld.

4.5 Brandstof

De beste soort "brandstof" voor dit soort reactor is een combinatie van deuterium en tritium. Door de kosten van deze stoffen gebruik ik deze niet, wat een werkelijke fusie ook moeilijker maakt

Deuterium is een waterstofatoom met een extra neutron in de nucleus. Deze stof kost meer dan €100 per 100ml en was dus te duur voor mij. Deze stof is ook zelf te verkrijgen doordat het natuurlijk voorkomt in water, maar om de concentraties hoog genoeg te krijgen is er enorm veel elektriciteit nodig, dat maakt het duurder dan online kopen.

Tritium is een waterstofatoom met twee extra neutronen in de nucleus. Door de hoeveelheid neutronen is deze stof ook radioactief met een halfleven van 12 jaar en geeft beta stralen af. Niet alleen is dit te duur voor mij, maar dit is ook een gevaarlijke stof om te gebruiken voor dit project.

Besluit

Mijn GIP was een grote uitdaging. Het toont dat veel onderdelen van elkaar afhangen voor de volledige werking van het geheel. Niet alleen voor de werking ervan, maar ook voor de veiligheid is hier belangrijk. Het vroeg veel opzoekwerk en onderzoek, het aanpassen van niet werkende delen. Het blijkt ook niet gemakkelijk te zijn een fusiereactor te bouwen met elektronische besturing.

Er waren onderdelen waar ik lang naar een oplossing heb moeten zoeken, en ik ben ook veel tijd verloren door kapotte onderdelen en het lange proces, onderhandelingen met bedrijven om dit in orde te krijgen. Ik heb hier veel uit geleerd door eigen opzoekwerk, zelfstudie en doorzetting. Verder dank ik iedereen die mij hierin heeft ondersteund.



MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface

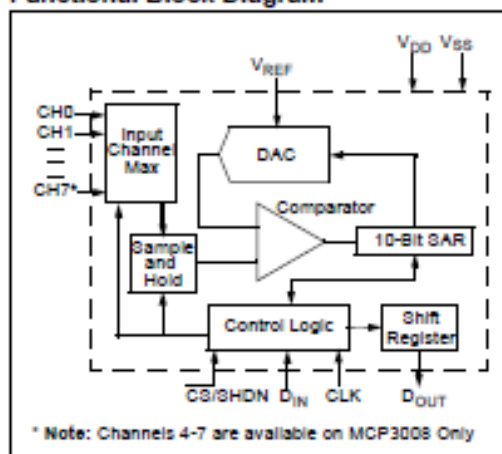
Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog Inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 kbps max. sampling rate at $V_{DD} = 5V$
- 75 kbps max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
- 5 nA typical standby current, 2 μA max.
- 500 μA max. active current at 5V
- Industrial temp range: $-40^{\circ}C$ to $+85^{\circ}C$
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

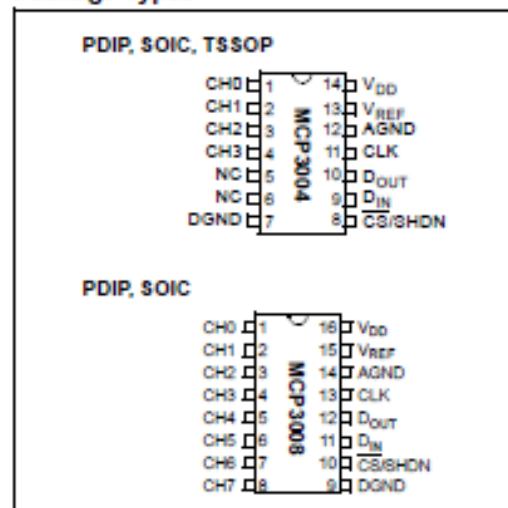
Functional Block Diagram



Description

The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ± 1 LSB. Communication with the devices is accomplished using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200 kbps. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby currents of only 5 nA and typical active currents of 320 μA . The MCP3004 is offered in 14-pin PDIP, 150 mil SOIC and TSSOP packages, while the MCP3008 is offered in 16-pin PDIP and SOIC packages.

Package Types



5.0 SERIAL COMMUNICATION

Communication with the MCP3004/3008 devices is accomplished using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the \overline{CS} line low (see Figure 5-1). If the device was powered up with the \overline{CS} pin low, it must be brought high and back low to initiate communication. The first clock received with \overline{CS} low and D_{IN} high will constitute a start bit. The $\overline{SGL/DIFF}$ bit follows the start bit and will determine if the conversion will be done using single-ended or differential input mode. The next three bits (D_0 , D_1 and D_2) are used to select the Input channel configuration. Table 5-1 and Table 5-2 show the configuration bits for the MCP3004 and MCP3008, respectively. The device will begin to sample the analog input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

Once the D_0 bit is input, one more clock is required to complete the sample and hold period (D_{IN} is a "don't care" for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 10 clocks will output the result of the conversion with MSB first, as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 10 data bits have been transmitted and the device continues to receive clocks while the \overline{CS} is held low, the device will output the conversion result LSB first, as is shown in Figure 5-2. If more clocks are provided to the device while \overline{CS} is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring \overline{CS} low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.1 "Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports", "Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports", for more details on using the MCP3004/3008 devices with hardware SPI ports.

TABLE 5-1: CONFIGURE BITS FOR THE MCP3004

Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	D2*	D1	D0		
1	X	0	0	single-ended	CH0
1	X	0	1	single-ended	CH1
1	X	1	0	single-ended	CH2
1	X	1	1	single-ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

* D2 is "don't care" for MCP3004

TABLE 5-2: CONFIGURE BITS FOR THE MCP3008

Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Because communication with the MCP3004/3008 devices may not need multiples of eight clocks, it will be necessary to provide more clocks than are required. This is usually done by sending 'leading zeros' before the start bit. As an example, Figure 6-1 and Figure 6-2 shows how the MCP3004/3008 can be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0, which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1, where the clock idles in the 'high' state.

As is shown in Figure 6-1, the first byte transmitted to the A/D converter contains seven leading zeros before the start bit. Arranging the leading zeros this way induces the 10 data bits to fall in positions easily manipulated by the MCU. The MSB is clocked out of the A/D converter on the falling edge of clock number 14. Once the second eight clocks have been sent to the device, the MCU receive buffer will contain five unknown bits (the output is at high-impedance for the first two clocks), the null bit and the highest order 2 bits of the conversion. Once the third byte has been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Employing this method ensures simpler manipulation of the converted data.

Figure 6-2 shows the same thing in SPI Mode 1,1, which requires that the clock idles in the high state. As with mode 0,0, the A/D converter outputs data on the falling edge of the clock and the MCU latches data from the A/D converter in on the rising edge of the clock.

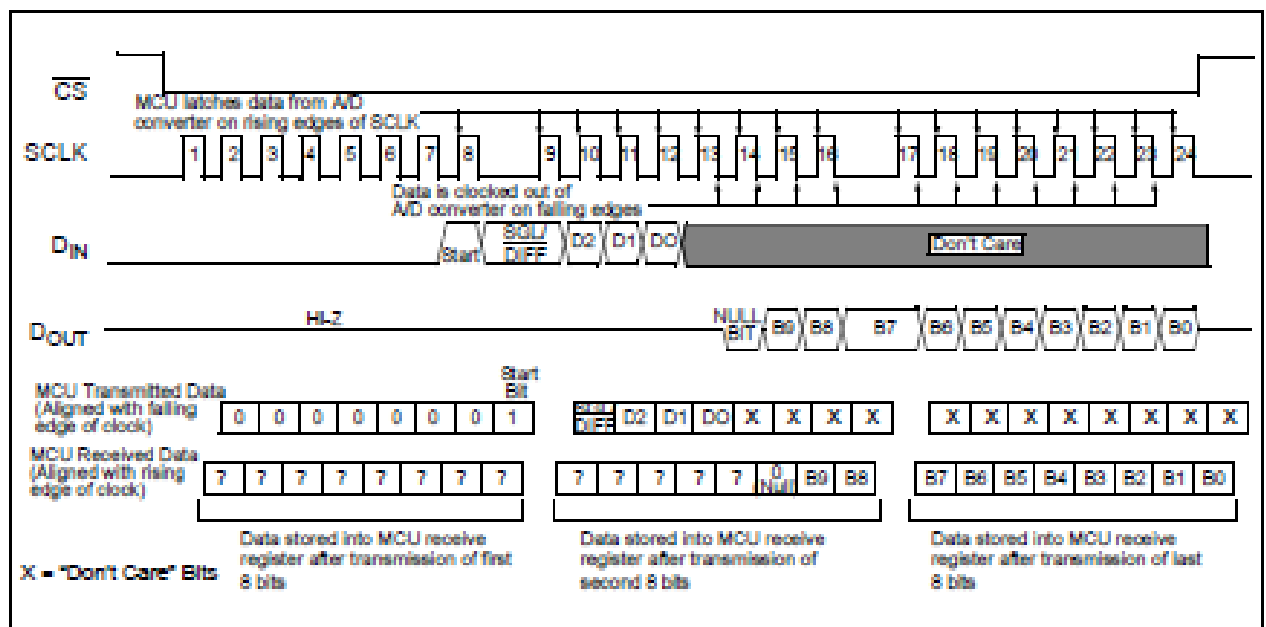


FIGURE 6-1: SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 0,0; SCLK idles low).

LM4040-N/-Q1 Precision Micropower Shunt Voltage Reference

1 Features

- SOT-23 AEC Q-100 Grades 1 and 3 Available
- Small Packages: SOT-23, TO-92, and SC70
- No Output Capacitor Required
- Tolerates Capacitive Loads
- Fixed Reverse Breakdown Voltages of 2.048 V, 2.5 V, 3 V, 4.096 V, 5 V, 8.192 V, and 10 V
- Key Specifications (2.5-V LM4040-N)
 - Output Voltage Tolerance (A Grade, 25°C): $\pm 0.1\%$ (Maximum)
 - Low Output Noise (10 Hz to 10 kHz): $35 \mu\text{V}_{\text{rms}}$ (Typical)
 - Wide Operating Current Range: 60 μA to 15 mA
 - Industrial Temperature Range: -40°C to $+85^\circ\text{C}$
 - Extended Temperature Range: -40°C to $+125^\circ\text{C}$
 - Low Temperature Coefficient: 100 ppm/ $^\circ\text{C}$ (Maximum)

2 Applications

- Portable, Battery-Powered Equipment
- Data Acquisition Systems
- Instrumentation
- Process Controls
- Energy Management
- Product Testing
- Automotives
- Precision Audio Components

3 Description

Ideal for space-critical applications, the LM4040-N precision voltage reference is available in the sub-miniature SC70 and SOT-23 surface-mount package. The advanced design of the LM4040-N eliminates the need for an external stabilizing capacitor while ensuring stability with any capacitive load, thus making the LM4040-N easy to use. Further reducing design effort is the availability of several fixed reverse breakdown voltages: 2.048 V, 2.5 V, 3 V, 4.096 V, 5 V, 8.192 V, and 10 V. The minimum operating current increases from 60 μA for the 2.5-V LM4040-N to 100 μA for the 10-V LM4040-N. All versions have a maximum operating current of 15 mA.

The LM4040-N uses a fuse and Zener-zap reverse breakdown voltage trim during wafer sort to ensure that the prime parts have an accuracy of better than $\pm 0.1\%$ (A grade) at 25°C. Bandgap reference temperature drift curvature correction and low dynamic impedance ensure stable reverse breakdown voltage accuracy over a wide range of operating temperatures and currents.

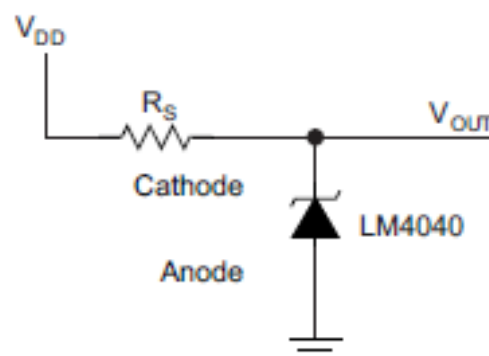
Also available is the LM4041-N with two reverse breakdown voltage versions: adjustable and 1.2 V. See the LM4041-N data sheet (SNOS841).

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM4040-N	TO-92 (3)	4.30 mm × 4.30 mm
	SC70 (5)	2.00 mm × 1.25 mm
	SOT-23 (3)	2.92 mm × 1.30 mm
LM4040-N-Q1	SOT-23 (3)	2.92 mm × 1.30 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Shunt Reference Application Schematic





LOW-POWER QUAD-CHANNEL DIGITAL ISOLATOR

Features

- High-speed operation
 - DC to 150 Mbps
- No start-up initialization required
- Wide Operating Supply Voltage: 2.70–5.5 V
- Wide Operating Supply Voltage: 2.70–5.5V
- Ultra low power (typical)
 - 5 V Operation:
 - < 1.6 mA per channel at 1 Mbps
 - < 6 mA per channel at 100 Mbps
 - 2.70 V Operation:
 - < 1.4 mA per channel at 1 Mbps
 - < 4 mA per channel at 100 Mbps
- High electromagnetic immunity
- Up to 2500 V_{RMS} isolation
- 60-year life at rated working voltage
- Precise timing (typical)
 - <10 ns worst case
 - 1.5 ns pulse width distortion
 - 0.5 ns channel-channel skew
 - 2 ns propagation delay skew
 - 6 ns minimum pulse width
- Transient Immunity 25 kV/μs
- AEC-Q100 qualified
- Wide temperature range
 - –40 to 125 °C at 150 Mbps
- RoHS-compliant packages
 - SOIC-16 wide body
 - SOIC-16 narrow body

Applications

- Industrial automation systems
- Hybrid electric vehicles
- Isolated switch mode supplies
- Isolated ADC, DAC
- Motor control
- Power inverters
- Communications systems

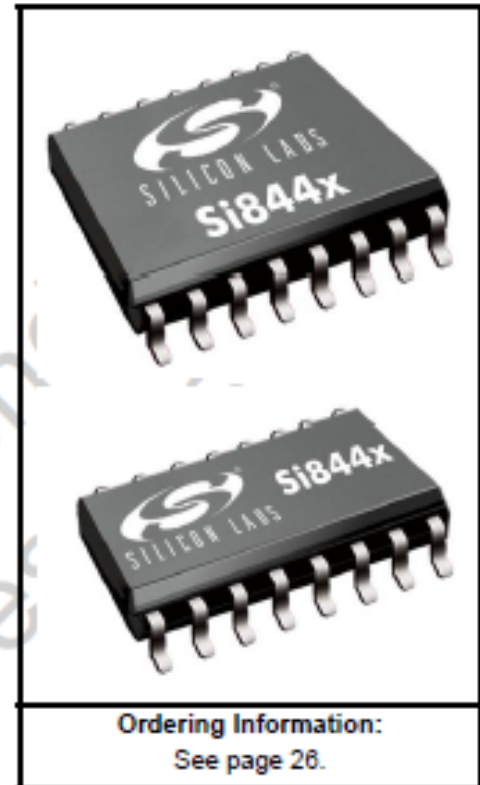
Safety Regulatory Approvals

- UL 1577 recognized
 - Up to 2500 V_{RMS} for 1 minute
- CSA component notice 5A approval
 - IEC 60950-1, 61010-1 (reinforced insulation)
- VDE certification conformity
 - IEC 60747-5-2 (VDE0884 Part 2)

Description

Silicon Lab's family of ultra-low-power digital isolators are CMOS devices offering substantial data rate, propagation delay, power, size, reliability, and external BOM advantages when compared to legacy isolation technologies. The operating parameters of these products remain stable across wide temperature ranges throughout their service life. For ease of design, only VDD bypass capacitors are required.

Data rates up to 150 Mbps are supported, and all devices achieve worst-case propagation delays of less than 10 ns. All products are safety certified by UL, CSA, and VDE and support withstand voltages of up to 2.5 kVrms. These devices are available in 16-pin wide- and narrow-body SOIC packages.



Ordering Information:
See page 26.

SBM20 Geiger muller tube specificaties

GENERAL SPECIFICATIONS

Parameters	SBM-20 / SBM-20U
Gas Filling	Ne + Br2 + Ar
Cathode Material	Stainless Steel, 50 mikm
Maximum Length (mm)	108 / 101
Effective Length (mm)	91.0 / 83.5
Maximum Diameter (mm)	11
Effective Diameter (mm)	10
Connector	Pin
Operating Temperature Range °C	-60 to +70

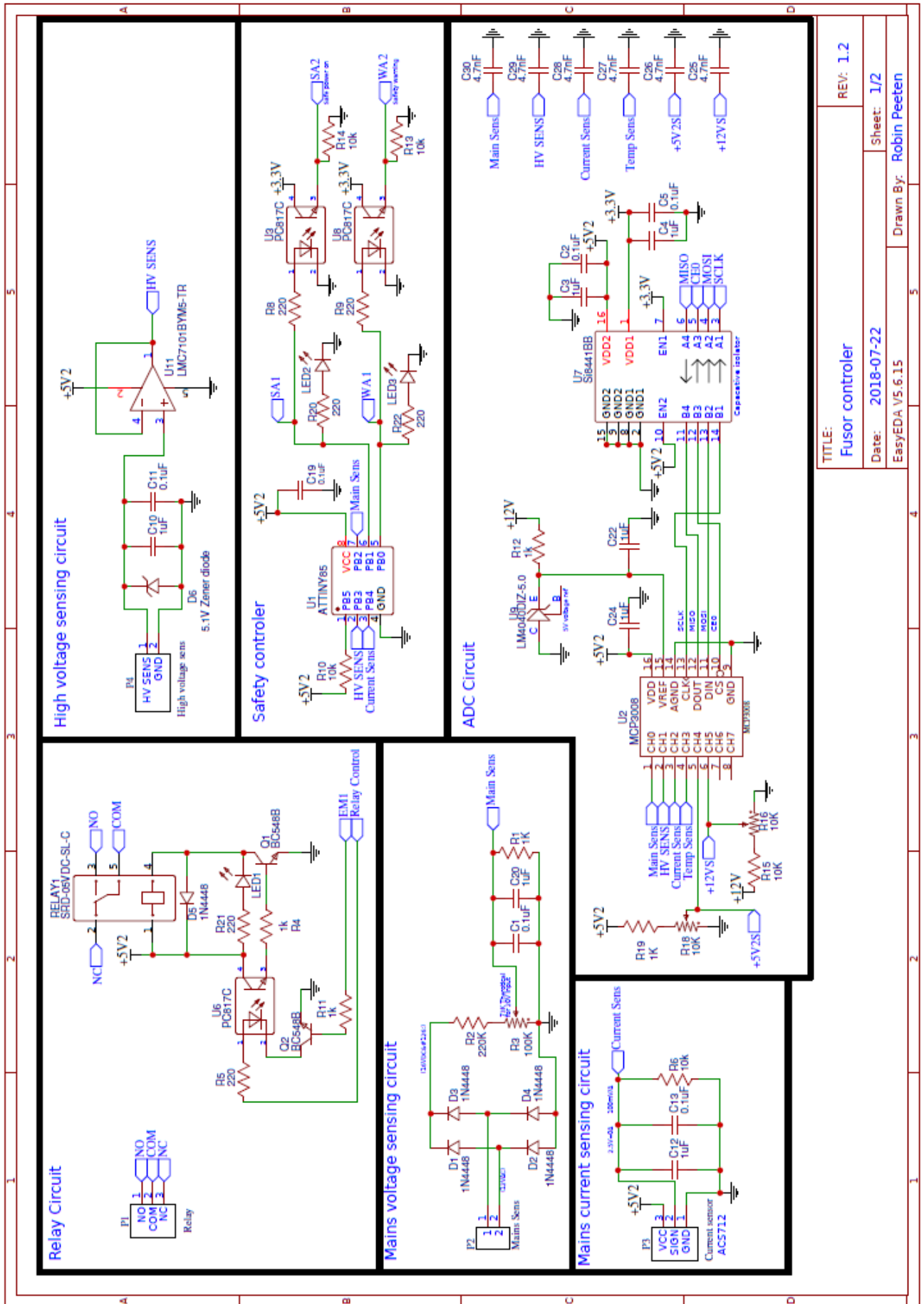
WALL SPECIFICATIONS

Parameters	SBM-20 / SBM-20U
Areal Density (mg/cm ²)	40
Thickness (mm)	0.05

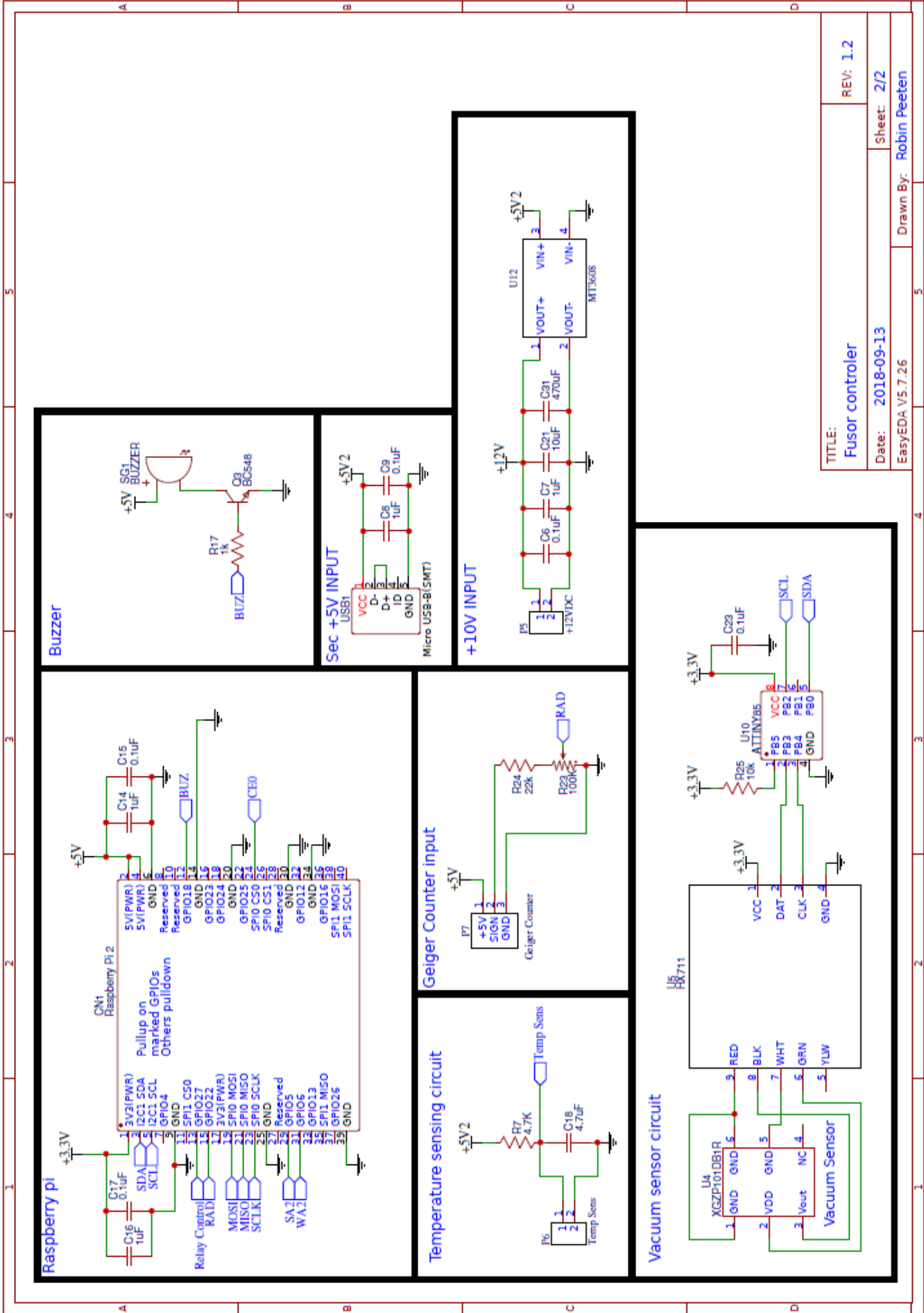
ELECTRICAL SPECIFICATIONS

Parameters	SBM-20 / SBM-20U
Minimum Anode Resistor (meg ohm)	1.0
Recommended Anode Resistor (meg ohm) circuit diagram	5.1
Recommended Operating Voltage (volts)	400
Operating Voltage Range (volts)	350 - 475
Initial voltage (volts)	260 - 320
Plateau length (volts)	at least 100
Maximum Plateau Slope (%/100 volts)	10
Minimum Dead Time (at U=400V, micro sec)	190
Working range (mR/s)	0.004 - 40
Working range (mR/h)	0.014 - 144
Gamma Sensitivity Ra ²²⁶ (cps/mR/hr)	29
Gamma Sensitivity Co ⁶⁰ (cps/mR/hr)	22
Inherent counter background (cps)	1
Tube Capacitance (pf)	4.2
Life (pulses)	at least 2*10 ¹⁰
Weight (grams)	10 / 9

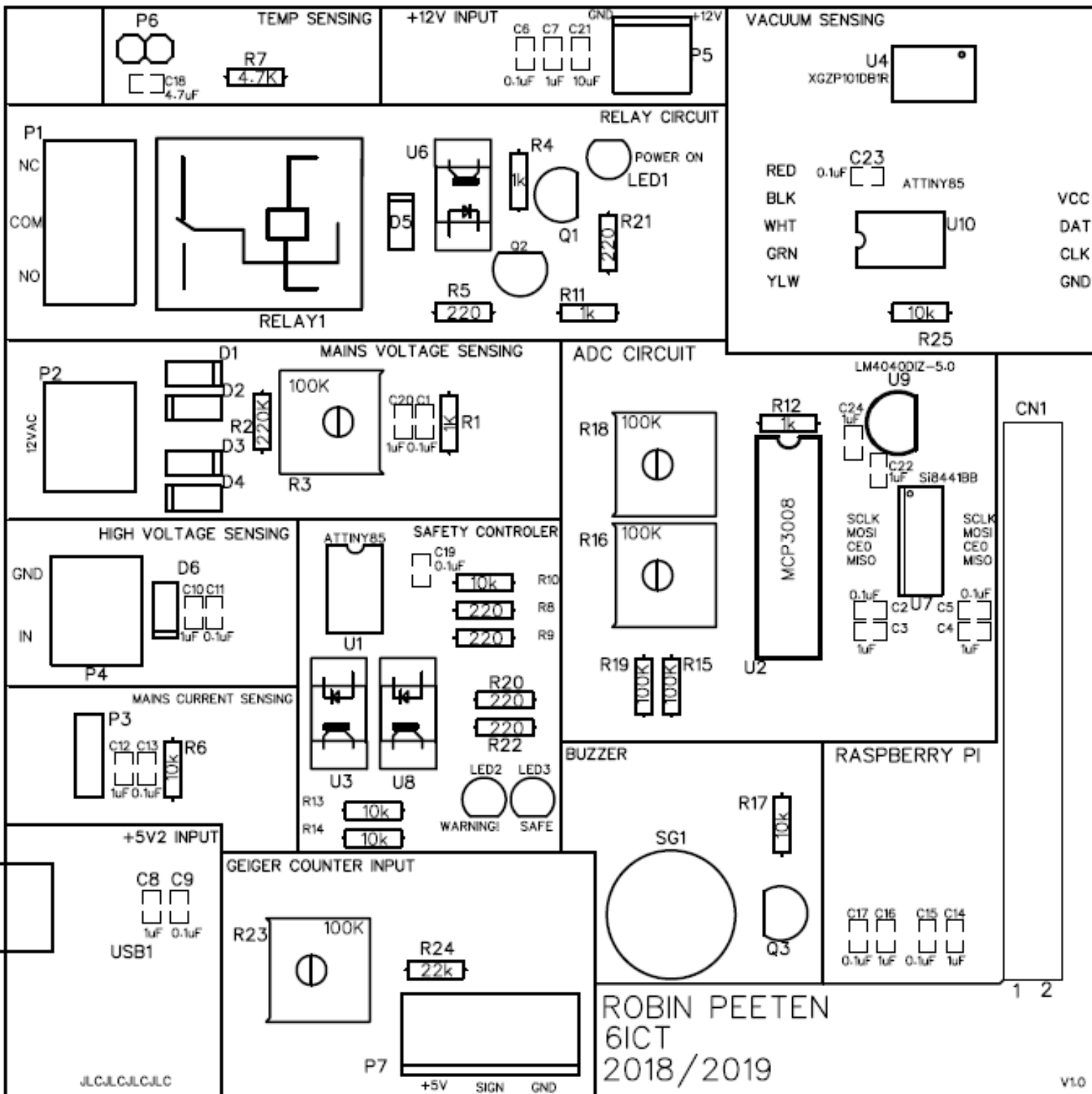
PCB elektronika schema's



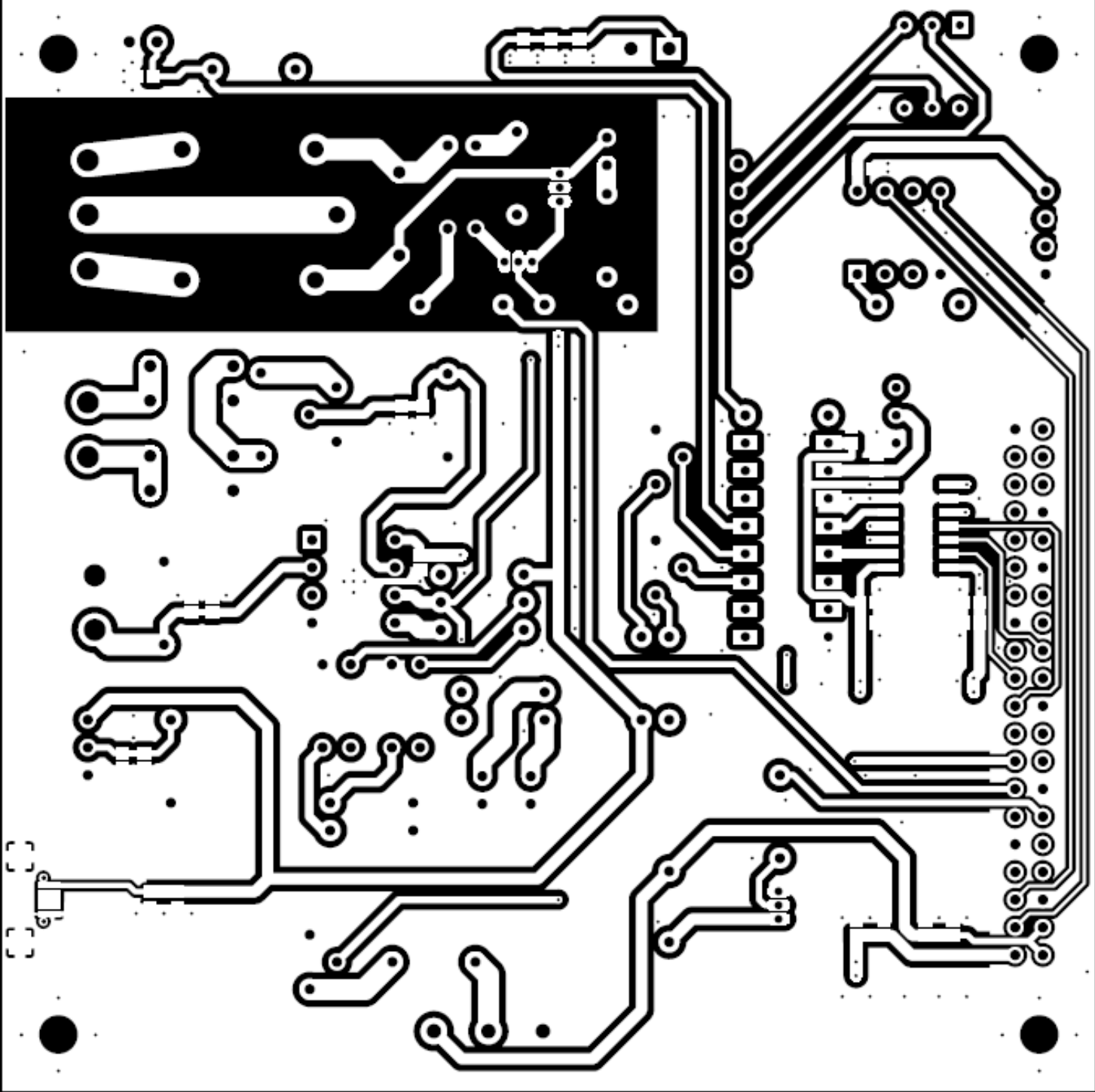
TITLE:	Fusor controller	REV: 1.2
Date:	2018-07-22	Sheet: 1/2
EasyEDA V5.6.15	Drawn By: Robin Peeten	



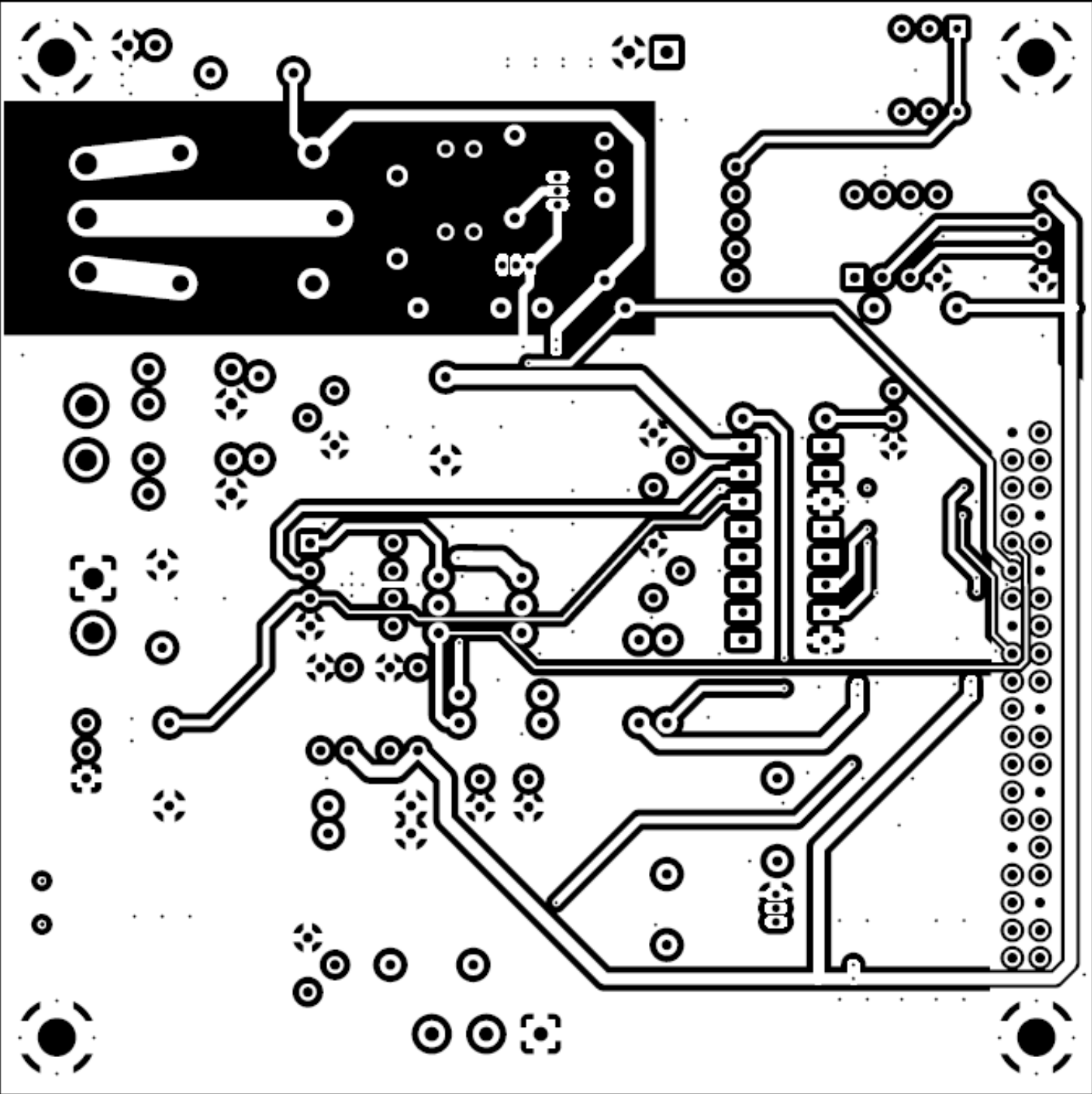
PCB Silk layer



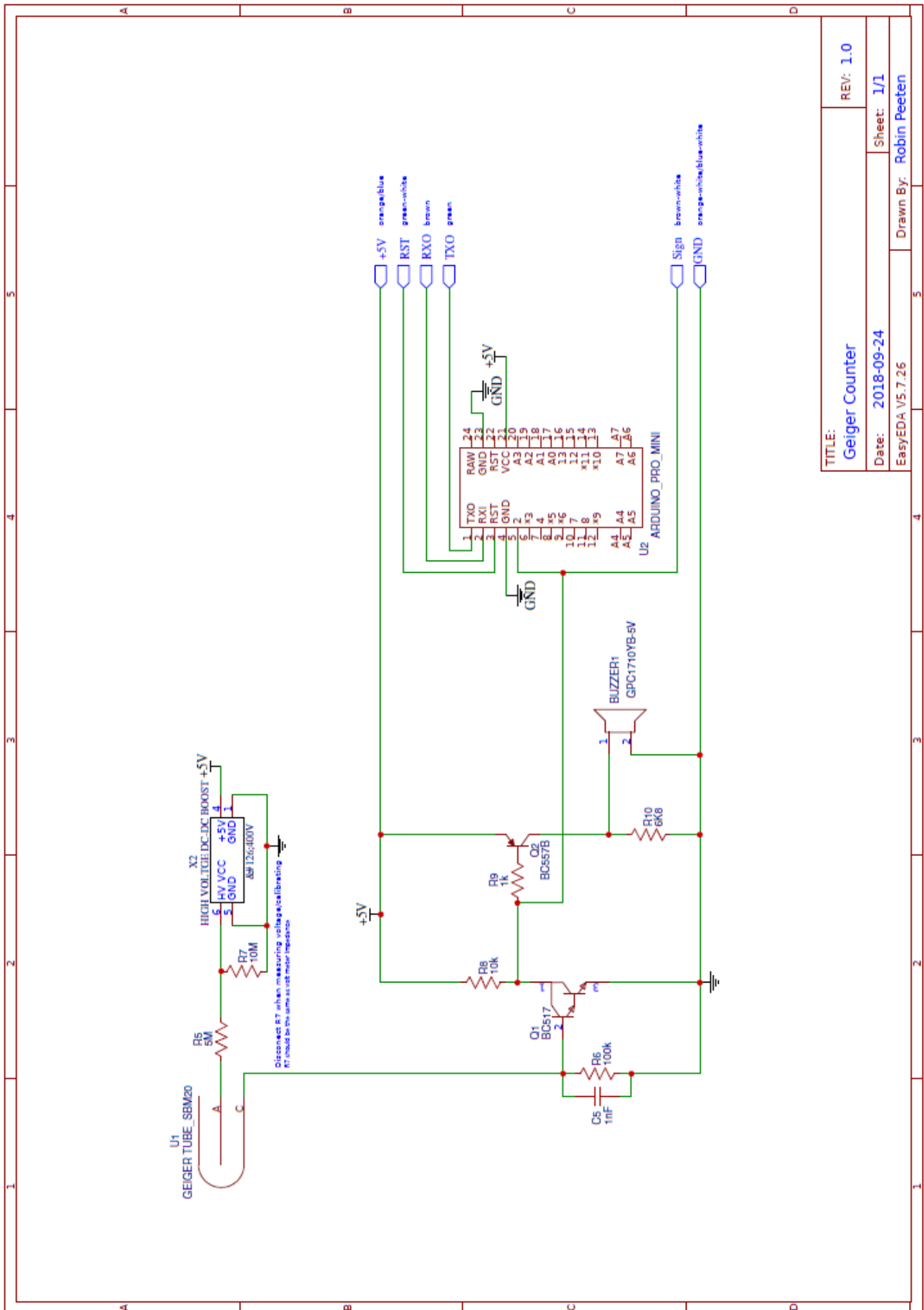
PCB Top layer



PCB bottom layer

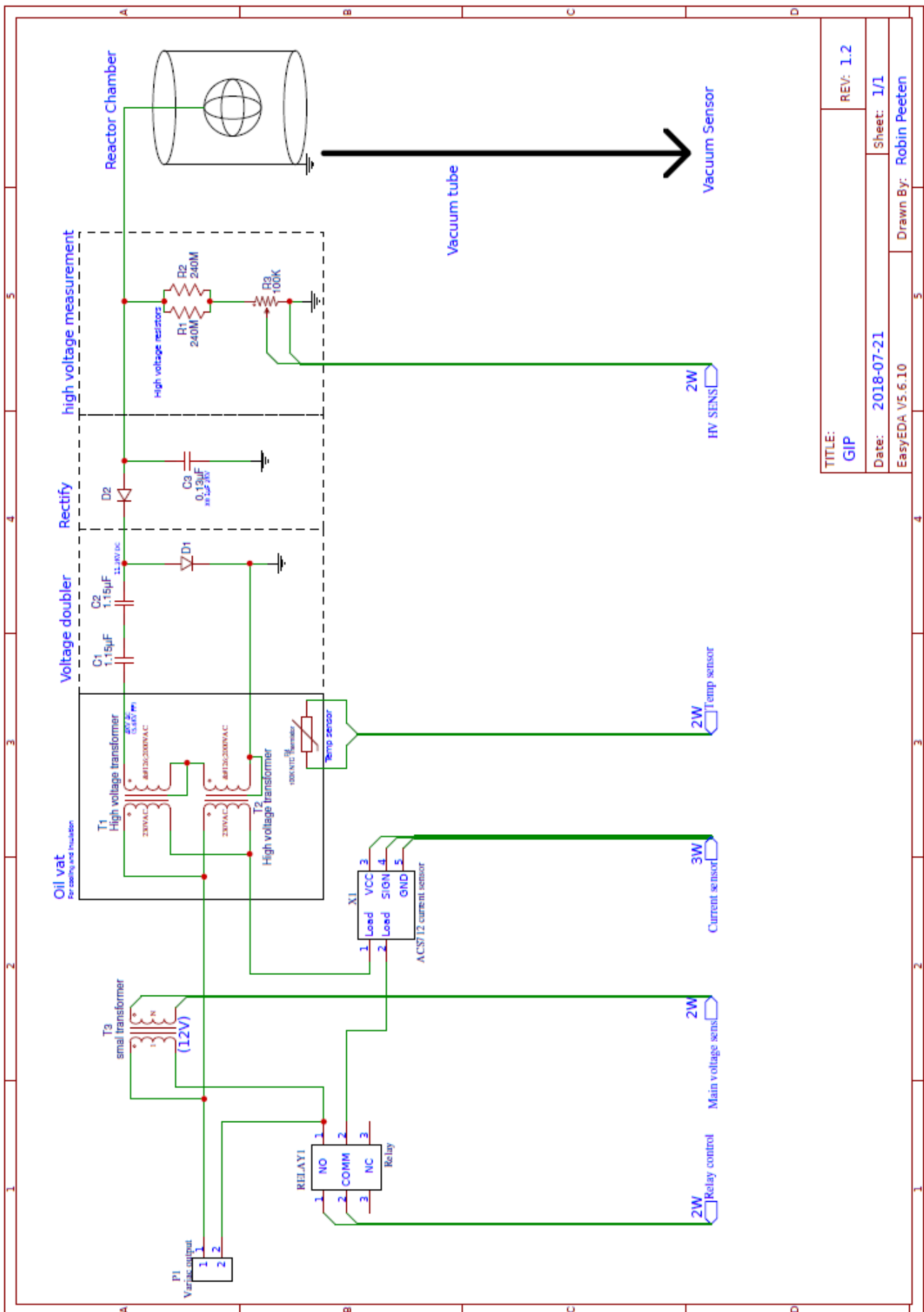


Elektronica Geiger counter



TITLE: Geiger Counter		REV: 1.0
Date: 2018-09-24	Sheet: 1/1	
EasyEDA V5.7.26		Drawn By: Robin Peeten

Hoogspanning schema



TITLE:	GIP	REV:	1.2
Date:	2018-07-21	Sheet:	1/1
EasyEDA V5.6.10		Drawn By:	Robin Peeten