



XMOTIONPLUS IMU by PXL

PWO_HEALTHCARE_DEV_1

Abstract

Een IMU-module met visualisatie software. Dit werk stand-alone met Bluetooth

Glenn Kerselaers & Devlin Voets

<http://xmotionplus.co.nf/>

INHOUD

1. Woord vooraf	3
2. Omschrijving van de opdracht	4
3. Doel	4
4. Verwerking	5
5. Resultaat	6
6. Opmerking	6
7. Prioriteiten	6
8. Kritische succesfactor	6
9. Onzekerheden	6
10. Risico's en valkuilen	6
11. Tijdplanning	7
12. Deelprojecten	7
13. Informatieplan	8
14. Communicatieplan	8
15. IMU-module	9
15.1. Inleiding	9
15.2. Componentenlijst	9
15.3. Schema	10
15.4. PCB	11
15.5. Blokschema	13
15.6. De arduino	14
15.7. De vibro-motor	14
15.8. Bluesmirf	15
15.9. IMU-sensor	15
15.9.1. Accelerometer	16
15.9.2. Gyroscop	16
15.10. Logic level converter	16
15.11. LiPO batterij en charger/booster	18
15.12. I ² C	19
15.12.1. Inleiding	19
15.12.2. Eigenschappen van i ² c	19
15.12.3. Werking van i ² c	19
15.12.4. Kloksynchronisatie	20
15.12.5. Arbitrage	20
15.12.6. Adresseren van i ² c componenten algemeen	20

15.12.7.	7-bit adressering	21
15.12.8.	10-bit adressering	21
15.12.9.	Conditie	21
15.12.10.	Start-conditie	22
15.12.11.	Stop-conditie	22
15.12.12.	Adres-pconditie	23
15.12.13.	Acknowledge van een slave.....	23
15.13.	Software	26
16.	Visualisatie software	27
16.1.	GUI	27
16.2.	Software	28
17.	Bronnen	29
17.1.	Software	29
17.2.	Arduino bibliotheken	29
17.3.	Processing bibliotheken.....	29
17.4.	Project websites	29

1. WOORD VOORAF

Wij hebben dit project de voorkeur gegeven omdat we het onderwerp en doel aantrekkelijk vonden. De zorgsector heeft altijd ons geïnteresseerd. We willen de mensen die ons geholpen, gesteund hebben bedanken. Vooral de leerkrachten die ons hebben geholpen als we vastzaten, Ing. Vincent Claes en Ing. Dieter Vanrykel.

2. OMSCHRIJVING VAN DE OPDRACHT

Van het departement Healthcare hebben we een opdracht gekregen. Deze houdt in dat de bewegingen van een patiënt, bijvoorbeeld arm- of beenbeweging, geanalyseerd worden. Dit zal worden gedaan met sensoren. Deze worden bevestigd op de patiënt en vervolgens zullen de gegevens draadloos naar de computer gestuurd worden. Hierdoor heeft men de mogelijkheid om de beweging te visualiseren of analyseren op de computer. We zullen daarom twee IMU (Inertial Measurement Unit) modules ontwerpen. Deze IMU modules worden verbonden aan een interfacekaart met Bluetooth. Hiermee is het mogelijk om gegevens draadloos te versturen. De IMU's worden gebruikt om de snelheid, oriëntatie en zwaartekracht te meten. Er kan ook feedback worden gegeven aan de gebruiker door vibrerende motoren in de IMU. Er is ook de mogelijkheid om meerdere modules te gebruiken.

De kern van dit project is de kennis rond de IMU en Bluetooth waaronder:

1. Een rapport
2. Hardware
3. Software
4. Demo-opstelling

Als een klant zich bij ons meldt voor advies of met een ontwerpopdracht, dan moeten wij uiteraard een helder beeld hebben van:

- Welke opties zijn er?
- Waar moeten we op letten?
- Welke hardware-bouwstenen zijn nodig/beschikbaar?
- Welke software-bouwstenen zijn nodig/beschikbaar?
- etc...

Het eindproduct is bestemd voor een omgeving waarbij twee of meerdere apparaten op afstand gemeten en/of bestuurd worden via Bluetooth. De opdrachtgevers van dit project zijn Mr. Dieter Vanrykel en Mr. Vincent Claes. De teamleden van dit project zijn Devlin Voets en Glenn Kerselaers. Het product zal worden gebruikt medische wereld. De gebruikers kunnen docenten, studenten, dokters, patiënten,... Het product moet simpel te bedienen en gebruiken sinds dat de eindgebruiker geen of weinig technische kennis heeft. Het is een totaal pakket dus hard- en software.

3. DOEL

Het doel van dit project is het vergaren van kennis en ervaring met betrekking tot remote monitoring en controlling van microcontrollers over Bluetooth. Hiernaast specifieke kennis met betrekking tot de Arduino microcontroller, Inter-Integrated Circuit Protocol, IMU-module en de mogelijkheden met deze IMU. De gebruikte IMU-module zal uitvoerig worden behandeld wat betreft zijn inzetbaarheid.

4. VERWERKING

Wij zijn begonnen met onze opdracht te bestuderen en nadenken hoe we het gaan aanpakken? Wat zijn de streefdatums?...

Om succesvol naar ons einddoel te werken, hebben we de opdracht gesplitst in fases en op deze fases een tijdschema aangebonden. Hierna vind je een overzicht van de fases met de overeenkomstige activiteiten.

FASE	ACTIVITEIT	STREEFDATUM
Haalbaarheidsfase	<ul style="list-style-type: none"> • Bluetooth overdrachtssnelheid hoog genoeg? • Resolutie van sensoren groot genoeg? • Voeding. • Compatible microcontroller. • Bevestiging van de modules. • I²C meest voorkomende problemen voorkomen. 	26 januari 2014
Definitiefase	<ul style="list-style-type: none"> • Twee IMU's waarin een accelerometer en gyroscoop zit. • Een interfacekaart met Bluetooth (BLE) module. • Microcontroller voor gegevensverwerking en overdracht naar PC. • Eventueel visualisatie software voor PC. 	19 januari 2014
Realisatiefase	<ul style="list-style-type: none"> • Schema opstellen, printontwerp en print montage. • Software ontwikkelen. • Module bevestiging systeem. 	4 mei 2014
Testfase	<ul style="list-style-type: none"> • Sensoren uitlezing. • I²c gegevens overdracht. • Bluetooth gegevens overdracht. • Interpretatie gegevens met PC. 	18 mei 2014
Nazorgfase	<ul style="list-style-type: none"> • Demo-presentatie. • Bundel. • Website. • Documentatie project. 	1 juni 2014

5. RESULTAAT

We ontwerpen een demonstratie-toepassing, die gebruikt maakt van het I2C protocol dat wordt toegepast door de I/O poorten van de IMU-module. Deze kunnen uitgelezen worden via een client applicatie in combinatie met Bluetooth. De ingelezen data wordt gevisualiseerd op de gebruiker zijn computer. Er wordt software voor de microcontroller en client geschreven.

Ook zullen we een rapport opstellen waarin de beperkingen en mogelijkheden van ons product vermeld staat.

- Een rapport met daarin heldere antwoorden op de volgende vragen:
- Wat zijn de beperking van Bluetooth?
- Wat zijn de beperkingen van de IMU-module
- Mogelijkheden van de IMU-model
- Software
- Een set functies om gegevens op te vragen
- visualisatie software
- Een demo-opstelling
- Beweging van de arm op het scherm laten zien

6. OPMERKING

Een bijproduct van dit project is de extra kennis van I2C. De toepassingsgebieden van I2C. Hoe het een ontwerp kan helpen om zo minimaal mogelijk te worden.

7. PRIORITEITEN

Data gegevens van de IMU-modules via Bluetooth naar de computer sturen.

8. KRITISCHE SUCCESFACTOR

- Genoeg procesvermogen.
- Nauwkeurigheid.
- Gebruiksgemak.
- Betrouwbaarheid.
- Goede Bluetooth verbinding.

9. ONZEKERHEDEN

We hebben nog nooit met Bluetooth gewerkt. Maar er is genoeg informatie beschikbaar online. We hebben geen kennis over digitale signaal verwerking. Dus we moeten op eigen houtje zoeken naar oplossing voor een vloeiend stabiel signaal.

10. RISICO'S EN VALKUILEN

Het risico dat we misschien hebben is met het gebruik van meerdere IMU-module. Misschien is de Arduino niet snel genoeg om de data te verwerken.

11. TIJDPLANNING

Dit is een ruwe tijdplanning waar we naar trachtte te werken.

Tijd Fasen	T 1	T 1	T 1	T 1	T 1	T 1	T 1	T 1	T 1	T 2	T 2	T 2	T 2	T 2	T 2	T 2	T 2	T 2
	W e e k 1	W e e k 2	W e e k 3	W e e k 4	W e e k 5	W e e k 6	W e e k 7	W e e k 8	W e e k 9	W e e k 1	W e e k 2	W e e k 3	W e e k 4	W e e k 5	W e e k 6	W e e k 7	W e e k 8	W e e k 9
Definiëren inhoud van project																		
Proef toepassing																		
Plan van Aanpak																		
Proef-toepassing																		
Afwerken Rapport																		
Documentatie																		

12. DEELPROJECTEN

- Proef-toepassing.
- IMU-module uitlezen.
- Data versturen met Bluetooth.
- Gegevens visualiseren.
- Arduino programmeren.
- Digitale signaal verwerking.

13. INFORMATIEPLAN

De informatie wordt opgeslagen in een Cloud systeem. Vanaf het moment dat er een goede versie beschikbaar is wordt deze op GitHub geplaatst. De informatie over het product bevat schema's, voorbeeldcodes,... De informatie over het gebruik zal bestaan uit een korte handleiding. Hierin wordt vermeld hoe de sensoren te bevestigen en aansturen. De informatie over het project bevat de planning, voortgang, eventuele problemen en afspraken.

14. COMMUNICATIEPLAN

- Elke woensdag van 11:45 tot 16:45 komen de groepsleden samen om te vergaderen, overleggen en elkaar in te lichten over de voortgang van het project.
- Elke vrijdag om 8:30 zal er een kort overleg zijn met de opdrachtgevers.
- Elke keer als er gewerkt wordt aan het project wordt dit geblogd.
- Bij belangrijke gebeurtenissen of problemen wordt dit opgenomen in de tijdlijn van de website.
- Na school wordt er bijna dagelijks contact opgenomen met elkaar via Skype of andere media.

15. IMU-MODULE

15.1. INLEIDING

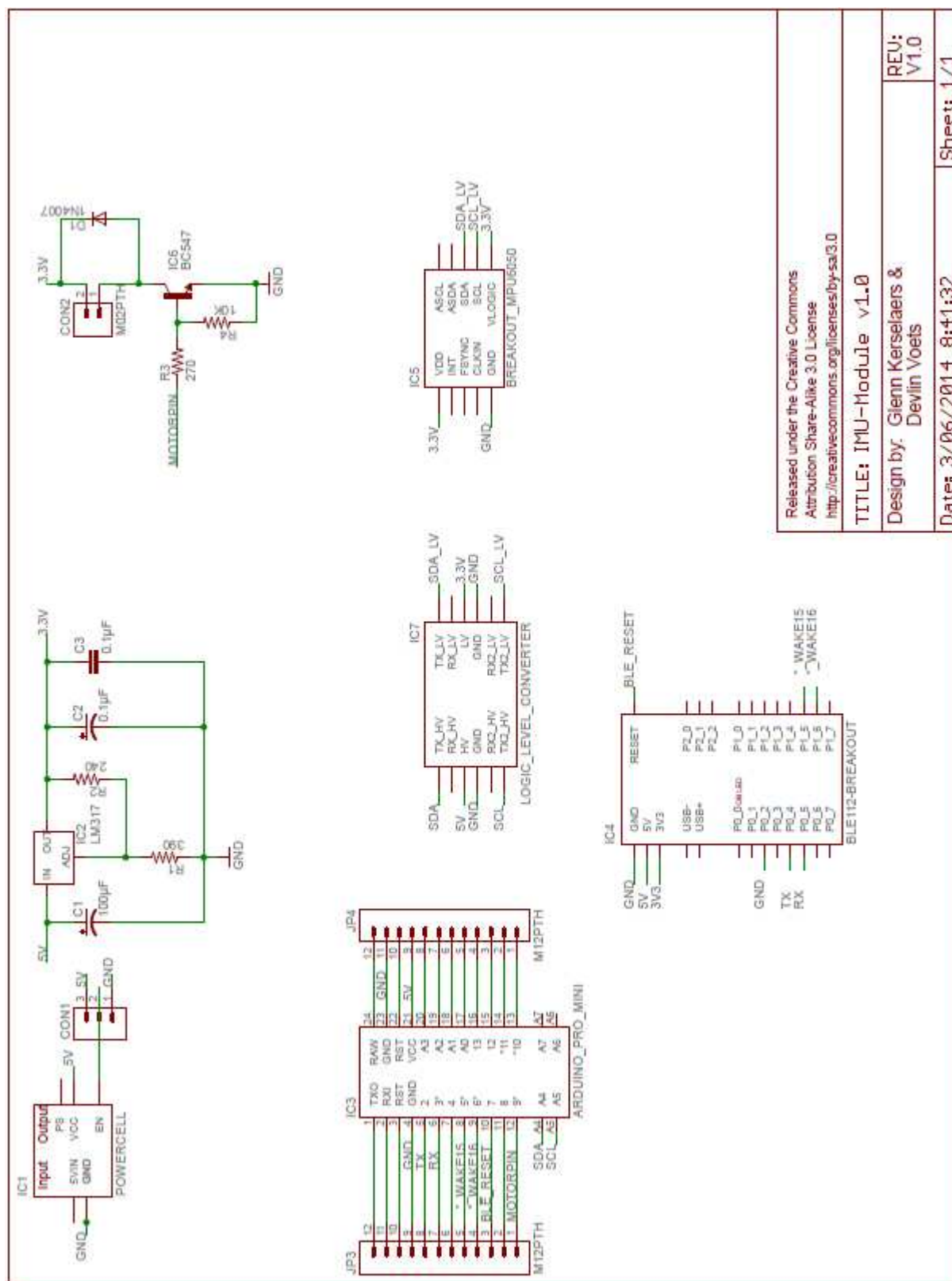
De IMU-module moest voldoen aan de eisen van dit project. We hebben ervoor gekozen om een gemoduleerd systeem te maken. Dit wil zeggen dat de PCB voorzien is van headers voor de modules op in te steken zodat ze na het project hergebruikt kunnen worden. Door deze keuze is de PCB groter dan normaal, maar de componenten zijn herbruikbaar.

15.2. COMPONENTENLIJST

Onze componenten komen vooral van Antratek en Reichelt. De onderstaande componentenlijst is voor één IMU-module.

AANTAL	BESCHRIJVING PRODUCT	PRIJS/ST
1	Arduino Pro Mini 328 - 5V/16MHz	€ 9,56
1	Logic Level Converter Bi-Directional	€ 2,89
1	Power Cell - LiPo Charger/Booster	€ 19,24
1	BAT-LIPO1400MAH	€ 7,20
1	FTDI Basic Breakout 5V	€ 14,40
1	BlueSmirf Silver	€ 30,25
	Accelerometer & Gyro – MPU-6050	€ 33,76
1	SMD-Multilayer ceramic capacitor 100N, 10%	€ 0,05
1	Subminiature-Elko, radial, 0,1µF/63Volt	€ 0,05
1	Subminiature-Elko, radial, 100µF/16Volt	€ 0,05
1	SMDChip Resistor, package 1206, 10 K-Ohm	€ 0,10
1	SMDChip Resistor, package 1206, 240 Ohm	€ 0,10
1	SMDChip Resistor, package 1206, 270 Ohm	€ 0,10
1	SMDChip Resistor, package 1206, 390 Ohm	€ 0,10
1	3-Terminal Adjustable Regulator	€ 0,83
1	Transistor NPN TO-92 45V 0,1A 0,5W	€ 0,04
1	Rectifier diode, DO41, 1000V, 1A	€ 0,02
1	Vibratie Motor disk 100% omsloten	€ 3,95
	Totaal	€ 122,69

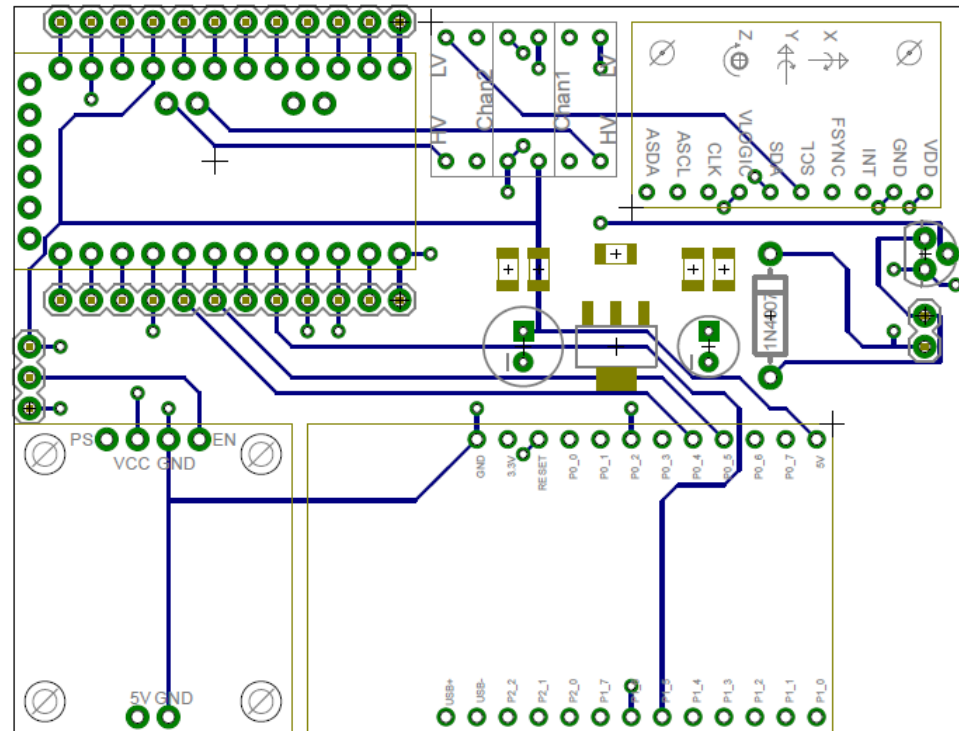
15.3. SCHEMA



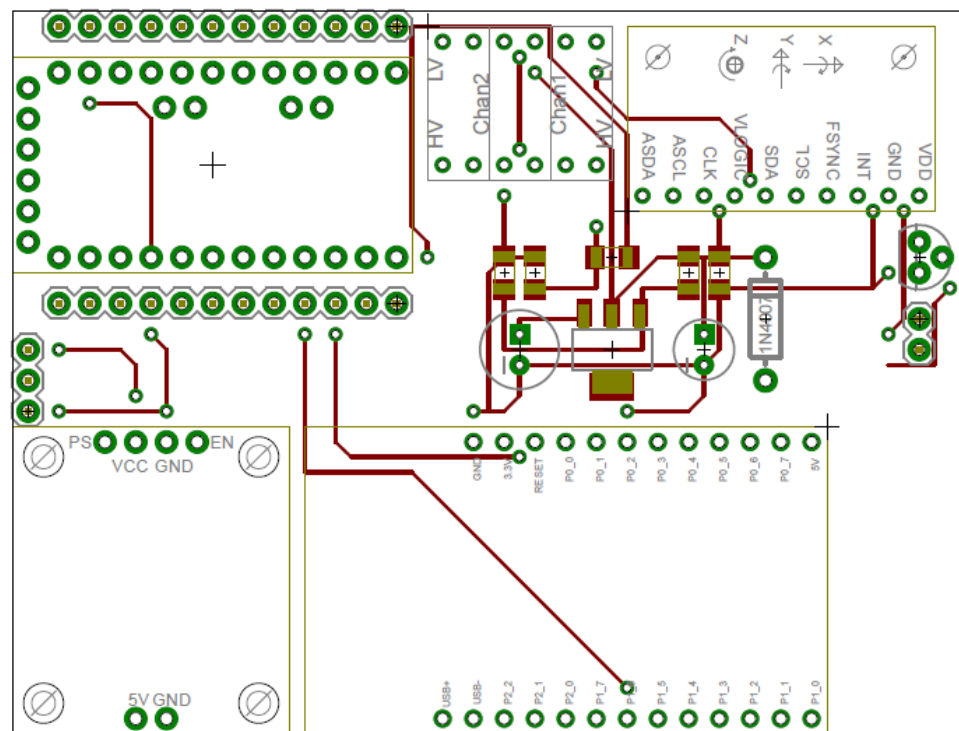
Figuur 1: Schema IMU-module

15.4. PCB

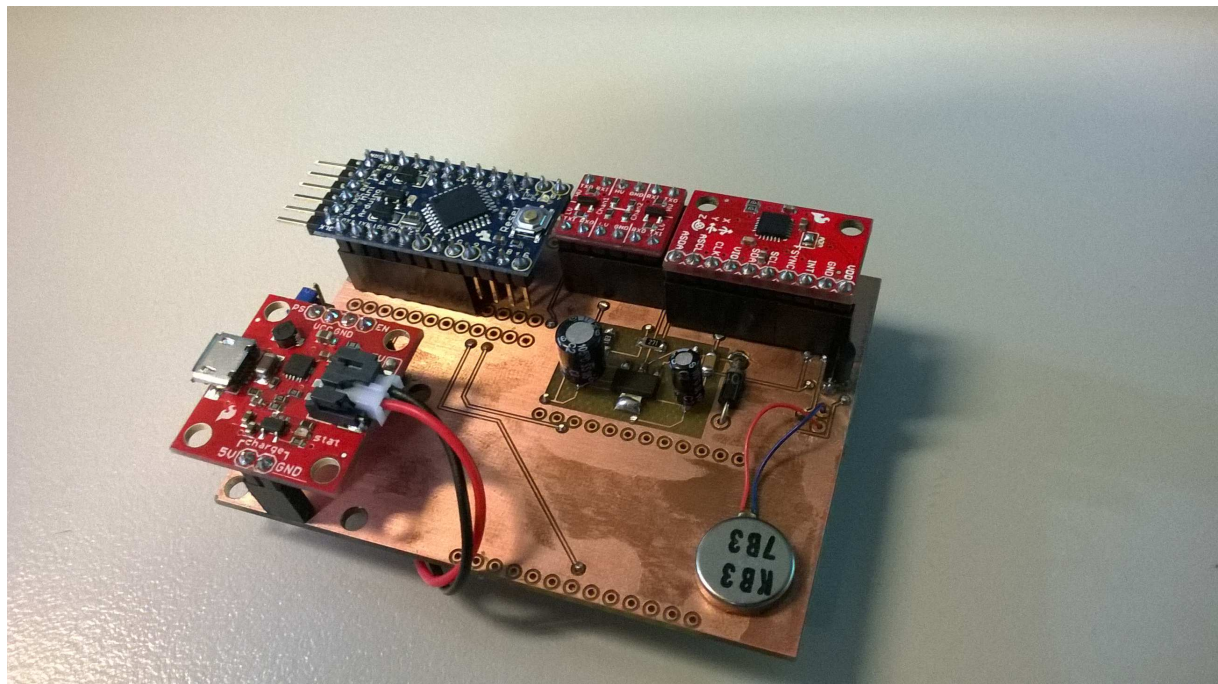
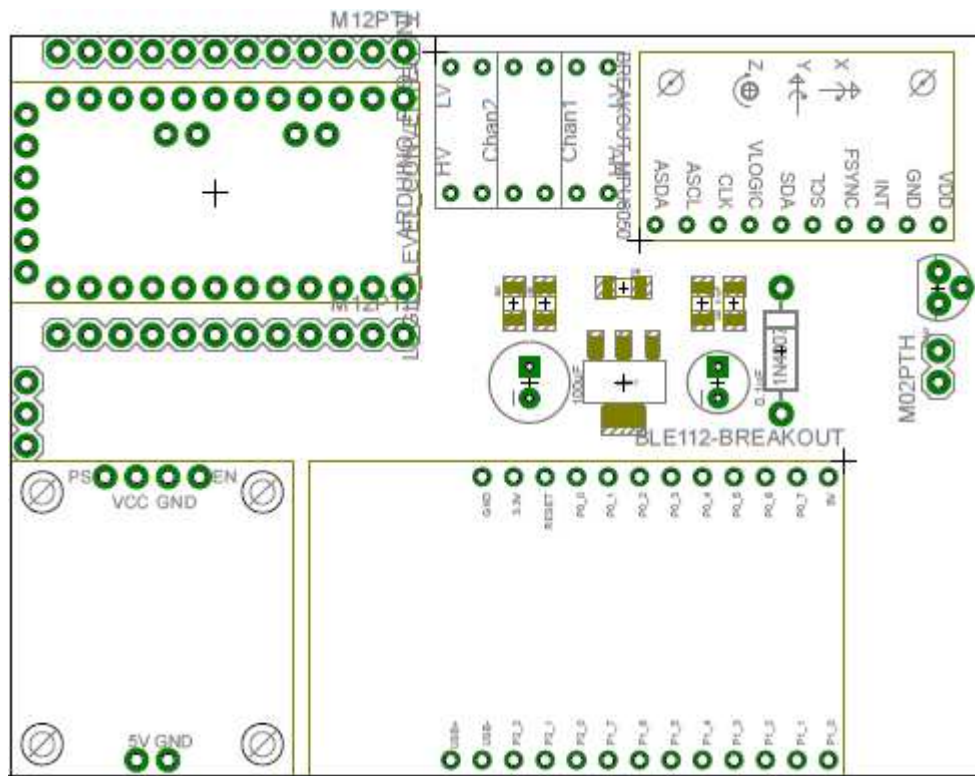
De PCB's werden plaatselijk gemaakt met het freesmachine.

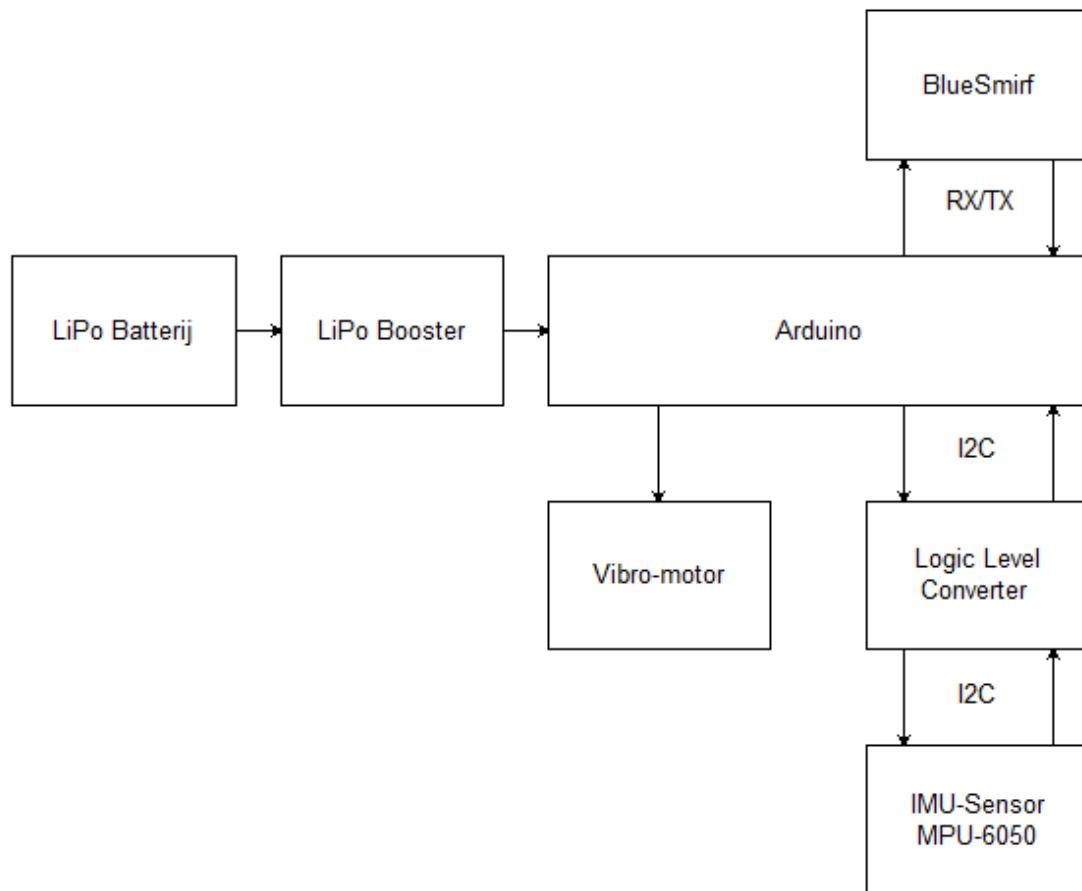


Figuur 2: PCB bottom layer



Figuur 3: PCB Top layer





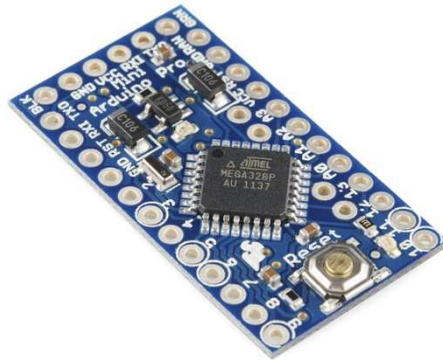
Figuur 6: Blokschema IMU-module

De IMU-module wordt gevoed door een LiPo-batterij. Sinds dat deze maximaal 3,7V bedraagt hebben we een LiPo booster aan de batterij gehangen. Dit zorgt ervoor dat de 3,7V opgekrikt wordt naar 5V. Deze 5V is geschikt voor het voeden van de IMU. We hebben een Arduino Pro Mini gekozen als microcontroller. Deze stuurt de componenten aan en verwerkt gegevens. De Arduino kan de vibro-motor aansturen en als ook gegevens opvragen van de IMU-sensor. De gegevens van de IMU-sensor worden opgevraagd d.m.v. I2C. Omdat de sensor een 3,3V apparaat is hebben we er een Logic Level Converter tussen gezet. Deze vormt het 5V signaal om naar een 3,3V signaal. Vervolgens voert hij een aantal bewerkingen uit om zo nauwkeurige juiste gegevens te bekomen. Deze gegevens kunnen verstuurd worden via de BlueSmirf, een Bluetooth module. De Arduino kan ook gegevens krijgen van de BlueSmirf. Er is ook een motorsturing en een 3,3V spanningsstabilisator schakeling aanwezig. De 3,3V spanningsstabilisator is nodig omdat de voeding enkel 3,7V of 5V uitgeeft.

15.6. DE ARDUINO

De Arduino Pro Mini is een microcontroller bord dat gebaseerd is op de ATmega328. Het heeft 14 digitale I/O aansluitingen waarvan 6 gebruikt kunnen worden als PWM uitgang, 8 analoge ingangen, on-board resonator, een reset knop. Om deze te programmeren is een FTDI kabel of een breakout bord nodig.

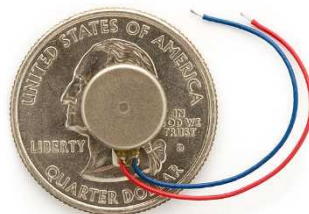
De Arduino wordt geprogrammeerd met een eigen taal genaamd Arduino. Arduino-taal is afgeleid van de programmeer taal wiring. Wiring is een afleiding van C. Standaard wordt de Arduino geprogrammeerd met het programma van Arduino zelf. Je kunt ook sommige andere code editors gebruiken.



Figuur 7: Arduino Pro Mini [Sparkfun]

15.7. DE VIBRO-MOTOR

Dit is een simpele DC-motor. Typisch aan dit soort motoren is dat ze klein zijn. Deze vindt je onder ander terug in gsm's en andere mobiele technologie. De motor werkt met een maximale spanning van 3,3V. De snelheid van de motor kan geregeld worden d.m.v. PWM.



Figuur 8: Vibration Motor [Sparkfun]

15.8. BLUESMIRF

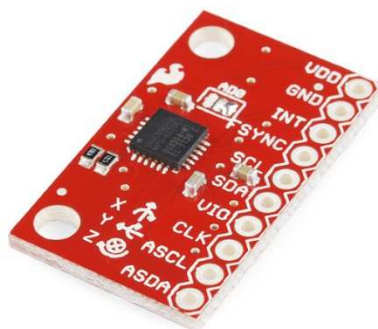
Dit is een Bluetooth apparaat waarmee je gegevens kunt versturen en ontvangen. Het functioneert als een seriële tunnel. Hierdoor is het zeer gemakkelijk om je bekabelde seriële verbinding te vervangen. Deze modules kunnen communiceren met elk Bluetooth apparaat dat SPP ondersteunt. Onze BlueSmirf maakt gebruik van de RN-42 Bluetooth SMD module. Dit is een klasse 2 Bluetooth apparaat dat gelimiteerd is tot 10 meter. We zijn niet dieper op theorie van Bluetooth gegaan sinds dat wij zelf niet direct aan het protocol moesten werken.



Figuur 9: BlueSMiRF Silver [Sparkfun]

15.9. IMU-SENSOR

De IMU-sensor is een MPU-6050. Deze bevat een 3-assige accelerometer, een 3-assige gyroscoop, een I2C interface, en een microcontroller. De microcontroller wordt gebruikt omdat de sensor DMP ondersteunt. Dit zijn waarden die bekomen worden door complexe berekeningen. Echter, wordt er weinig informatie verschaft hoe de DMP waarden opgevraagd en gebruikt kunnen worden.



Figuur 10: MPU-6050 Breakout [Sparkfun]

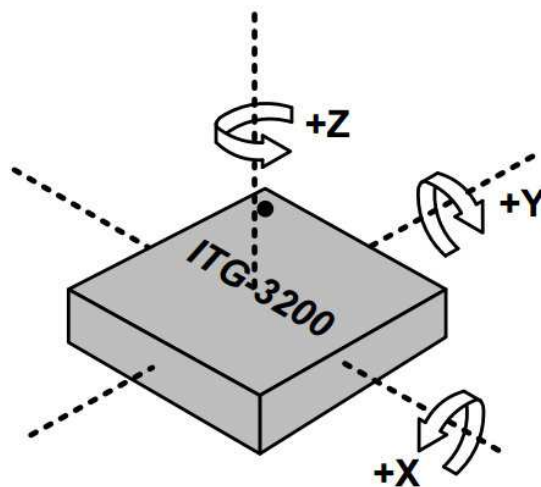
15.9.1. ACCELEROMETER

Een Accelerometer is een elektromechanisch apparaat dat versnellende krachten meet. Deze krachten kunnen statisch zijn, zoals de constante zwaartekracht die aan je lichaam trekt, of ze kunnen dynamisch, veroorzaakt door beweging of vibratie. Door het meten van de statische versnelling, wegens de zwaartekracht, kan men de hoek weergeven waaronder het apparaat zich bevindt t.o.v. de aarde. Door het waarnemen van de hoeveelheid dynamische versnelling kan men de weg van het apparaat volgen. Een Accelerometer kan het project helpen om de omgeving beter te kennen. Rijdt het omhoog? Gaat het omver vallen? Vliegt het horizontaal of maakt het een duikvlucht? Door middel van de data van de Accelerometer kan hier een antwoord op worden verschaft. Er zijn verschillende manieren om een Accelerometer te maken. Sommige Accelerometers maken gebruik van het piezo-elektrisch effect. Deze bevatten microscopisch kleine kristal structuren die op elkaar worden gedrukt door versnellende krachten hierdoor wordt er een spanning opgewekt. Een andere manier is door waarnemen van veranderingen in capacitantie. Twee microstructuren naast elkaar hebben een zekere capacitantie tussen elkaar. Als de versnellende krachten één van de structuren verplaatsen dan zal die capacitantie veranderen. Er is ook een schakeling toegevoegd om de capacitantie om te zetten naar spanning.

Er zijn nog talrijk andere methodes; piezo-resistief effect, hete lucht ballen en licht.

15.9.2. GYROSCOOP

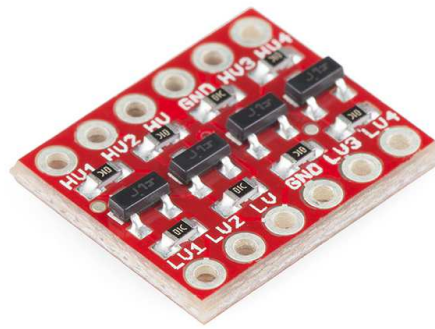
Een gyroscoop is een apparaat dat gebruikt wordt om een ronddraaiende beweging te meten. Wij gebruiken een gyroscoop om de beweging van de IMU te meten. De Gyroscoop is in staat om rotaties rond drie assen te bepalen: X, Y en Z.



Figuur 11: accelerometer assen

15.10. LOGIC LEVEL CONVERTER

De logic level converter wordt gebruikt om het logisch niveau van I²C te veranderen. Sinds dat de IMU-sensor een 3.3v sensor is, is het ook nodig dat het inkomende I²C-signaal van niveau veranderd. Deze LLC is bidirectioneel. Het maakt gebruik van MOSFET technologie.



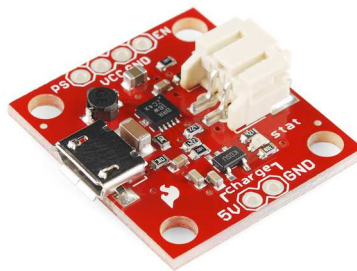
Figuur 12: Logic Level Converter [Sparkfun]

15.11. LIPO BATTERIJ EN CHARGER/BOOSTER

Het apparaat moest stand-alone werken en natuurlijk ook licht en compact. Daarom hebben we gekozen voor LiPo batterijen. De batterij levert 3,7V maar onze toepassing heeft 5V nodig. Als oplossing hebben we een booster en charger in één gevonden. Deze laadt de batterij op via micro-usb. Het is ook uitgerust met diverse beveiligingen zoals ontlading. Het kan ook geïntegreerd worden op de print. Enige nadeel is dat de oplader niet beveiligd is tegen omgekeerde polariteit. De batterij is beschermd tegen overspanning, overstroom en minimum spanning.



Figuur 13: LiPo batterij [Sparkfun]



Figuur 14: LiPo charger/booster [Sparkfun]

15.12.1. INLEIDING

In de computertechnieken worden bussen gebruikt om verschillende onderdelen met elkaar te verbinden. Ze orden op alle niveaus gebruikt: op lokaal niveau, op regionaal niveau en op globaal niveau. Op eenzelfde printplaat kan er sprake zijn van een bus systeem. Het is ook mogelijk dat verschillende printen binnen hetzelfde apparaat met elkaar verbonden zijn door hetzelfde bussysteem. Er zijn twee verschillende soorten bussen, de parallelle en de seriële bus. Bij parallelle bussen worden meerdere databits tegelijkertijd verzonden. Het aantal parallelle bits bedraagt 8, 16, 32 en ook 64. Bij een seriële bus worden alle bits apart verzonden. Bij parallelle bussen is de dataoverdracht snelheid over het algemeen sneller. Dit komt omdat er meerdere bits tegelijk worden verzonden. Parallelle communicatie wordt gebruikt voor relatief korte afstanden. Het maakt hierbij niet uit of de bus onderdelen op één print staan of verschillende printen. Bij dataoverdrachten tussen aparte apparaten zie je dit type bus minder vaak. De Wii remote en nunchuk is hier een uitzondering op. Om een dataoverdracht tussen aparte apparaten te verkrijgen wordt een seriële bus gebruikt. I²C is in het begin van de jaren 80 ontwikkeld door Philips semiconductors. De oorspronkelijke bedoeling was een gemakkelijke verbinding tussen een CPU en chips in een TV toestel te maken. Voorheen werd gebruikt gemaakt van Byte Wide. Dit systeem bestaat uit een 8 bit brede bus. Dit heeft gevolg dat er veel koperbanen nodig zijn voor de adres- en datalijnen. Doordat het veel besturingslijnen heeft betekend het dat het gevoeliger is voor storingen door EMC (Elektromagnetische Compatibiliteit) en ESD (Electro Static Discharge). Daarom werd door Philips onderzoek gedaan naar een 2-draadsbus, het resultaat I²C bus. De I²C-bus is een seriële bus. Het is ontwikkeld voor de verbinding van microprocessoren en andere IC's op één en dezelfde print. De I²C-bus heeft eigenschappen die de parallelle bus niet heeft. De IC's beschikken over ingangsfilters, de kleine pieken op de signaalleiding wegfilteren. Langzame chips kunnen een wachttijd afdwingen. Aan de wachteisen kunnen de parallelle systeem bussen niet voldoen. Interessant aan I²C is dat het Multi-master principe mogelijk is. Op deze wijze kunnen meerdere controllers dezelfde leidingnetten gebruiken en zelfs onderling communiceren.

15.12.2. EIGENSCHAPPEN VAN I²C

I²C heeft enkel twee lijnen nodig, een seriële datalijn (SDA) en een seriële kloklijn (SCL). Elk onderdeel op de bus heeft een eigen uniek adres. Er is op elk moment een master en slave relatie. Meerdere masters zijn mogelijk, er is dus geen centrale master. I²C werkt op verschillende snelheden, 100 kbit/s, 400 kbit/s en 3.4 Mbit/s. Er is bidirectionale data overdracht tussen masters en slaves.

15.12.3. WERKING VAN I²C

De I²C-bus is een systeembus waarbij tweedraden gebruikt worden voor data (SDA) en de systeemklok (SCL). Beide leidingen zijn met een pull-up weerstand met de VCC verbonden. In de niet-actieve toestand is het niveau op deze twee communicatielijnen dus hoog. Het aantal IC's dat op deze bus mag worden aangesloten is in principe onbeperkt. Wel is de maximale capaciteit van de leiding 400 pF.

Eerst wordt er een startconditie gegenereerd zodat andere IC's weten dat er een IC iets wil gaan verzenden. Vervolgend volgen er 8 bits voor het adres en een veelvoud van 8 bits voor de informatie naar de betreffende IC. Iedere reek van 8 bits wordt afgesloten door een acknowledge bit. Deze dient ter controle of de data goed is aangekomen. Als alle informatie is aangekomen wordt er een stopconditie gestuurd. Dit laat toe dat andere IC's ook data kunnen overdragen. Als een IC een reeks van 8 bits en een acknowledge-bit heeft ontvangen stuurt de IC een acknowledge-bit terug zodat de verzender weet dat de informatie goed is aangekomen.

Er zijn enkele regels/definities die opgesteld zijn om de bus naar behoren te laten functioneren, deze zijn als volgt:

- **Master:** de IC die bepaalt wanneer er in welke richting dataoverdracht plaatsvindt. Dit IC genereert ook als enige de klokpulsen op de SCL-lijn. Zijn er meerdere masters, dan is er sprake van een Multi-master systeem.
- **Slave:** elke IC die met de I²C-bus verbonden is, maar geen klokpulsen kan genereren. Deze IC's ontvangen commando's en kloksignalen van een master.
- **Vrije bus:** de bus is vrij als zowel de SDA- als de SCL-lijn hoog zijn. Een master kan een bus alleen in beslag nemen als deze vrij is.
- **Start:** een master neemt de vrije bus in beslag door een startconditie te sturen. Dit houdt in dat de SDA-lijn laag wordt gemaakt terwijl de SCL-lijn hoog is.
- **Stop:** een master kan een bus weer vrij geven door een stopconditie te sturen. De stopconditie is vergelijkbaar met de startconditie. Nu wordt alleen de SDA-lijn hoog gemaakt terwijl de SCL-lijn hoog is.
- **Ongeldige data:** de data op de SDA-lijn moet stabiel zijn zolang de SCL-lijn hoog is. Hieruit blijkt al direct dat de start- en stopcondities uniek zijn.
- **Dataformaat:** elke keer wordt over de I²C-bus acht databits (een byte) verzonden. Elke byte wordt gevolgd door een negende klokpuls tijdens dat de ontvangende IC (master of slave) een acknowledge (ACK) te sturen.
- **IC-adres:** elk IC dat geschikt is voor gebruik op de I²C-bus heeft een eigen, uniek zeven-bits adres. Gewoonlijk is dit adres in het IC vastgelegd, soms is een gedeelte van het adres extern instelbaar. Deze laatste optie maakt het gebruik van meerdere exemplaren van hetzelfde soort IC op de bus mogelijk. Adres 00 is het zogenaamde 'General Call Address'. Op dit adres reageren alle aangesloten IC's.
- **R/W-bit:** ieder IC heeft een zeven-bits adres. Het achtste bit (LSB) dat bij een adressering wordt verzonden, is het zogenaamde R/W bit. Is dit bit '1' dan wil de master de slave uitlezen. Bij een '0' wil hij data naar de slave schrijven.

15.12.4. KLOKSYNCHRONISATIE

Alle masters generen hun klokpulsen zelfstandig. Door de speciale configuratie van de buslijnen bestaat de mogelijkheid dat concurrerende masters die klokpulsen met elkaar kunnen synchroniseren. Een master die een '0' op de SCL-lijn zet zal ooit daarvan weer een '1' moeten maken. Vanaf dat tijdstip leest hij herhaaldelijk de SCL-lijn totdat deze '1' wordt. Dat betekent dat, zolang een ander apparaat de SCL-lijn nog laag houdt, de master zich daar op synchroniseert. Het traagste apparaat heeft voorrang.

15.12.5. ARBITRAGE

Onder arbitrage wordt het proces verstaan dat de toekenning van de bus aan de diverse masters regelt. Op de I²C-bus is een speciale procedure met de naam CSMA/CD (Carrier Sense, Multiple Acces with Collision Avoidance) geïmplementeerd. Carrier Sense betekent dat een master die iets wil lezen of schrijven, de lijnen scant om vast te stellen of de bus vrij is of niet. De I²C-bus wordt als bezet gezien als één van beide lijnen laag is. Als de master een vrije bus heeft gevonden. Genereert hij een startconditie en verzend de eerste byte, deze byte is het adres van de ontvanger. Als een andere master dit gelijktijdig zou willen doe, dan worden eerst de pulsen op de SCL-lijn uitgelijnd. Gedurende dat de gelezen klok hoog is zullen de master het verzonden SDA-bit vergelijken met het door hen teruggelezen bit.

Een master die een '1' verzendt en een '0' ontvangt, merkt dan op dat een ander apparaat de bus bedient en breekt de overdracht af. Deze procedure heet collision avoidance. Dit in tegenstelling tot een andere procedure met de naam collision detection, waarbij beide masters de bus moeten vrij geven. Door collision avoidance wordt het transport vermogen verhoogd, omdat er altijd een master is die de bus krijgt.

15.12.6. ADRESSEREN VAN I²C COMPONENTEN ALGEMEEN

Ieder apparaat dat verbonden is met de bus moet kunnen aangesproken kunnen worden (geadresseerd). Ter onderscheiding heeft elk apparaat een uniek adres: het slave adres. Apparaten die uitsluitend functioneren als master hebben geen adres nodig. Oorspronkelijk voorzag het I²C concept 7-bits adressen. Hiermee zijn maximaal

128 bouwstenen aan te sturen. Er zijn echter bepaalde adressen gereserveerd. Philips heeft het ook mogelijk gemaakt om 10-bits adressen te gebruiken. Omdat het gebruik van 7 bits en 10 bits adressen gelijktijdig mogelijk is, is de 10 bit modus geforceerd. De adressering voor de 10-bit modus is dus wat omslachtig.

15.12.7. 7-BIT ADRESSERING

Als een master met een slave wil communiceren, moet hij eerst de bus opvragen via een start conditie. Dan zendt hij als eerste een byte waarvan de eerste 7 bits het adres vormen. Het laatste bit legt de transportrichting vast ('0' = schrijven, '1' = lezen). Na een startconditie moeten alle aangesloten slaves de bus bewaken om dit byte te lezen hierbij kunnen langzame slaves de master tot wachten dwingen. Nadat alle 8 bits zijn ontvangen vergelijkt iedere slave het ontvangen adres met zijn eigen adres. Dit adres ligt vast of het kan handmatig zijn ingesteld. IC's met een afwijkend adres luisteren niet meer naar de bus voorlopig en wachten op een volgende startconditie. De IC met het juiste adrs stuurt een acknowledge-bit.

Nu weet de master dat de geadresseerde slave tot dataoverdracht in staat is. Komt er geen acknowledge-bit dan is de IC beschadigd of is er geen IC aanwezig op dat adres.

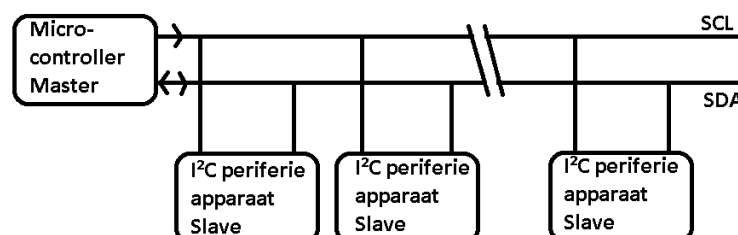
15.12.8. 10-BIT ADRESSERING

10-bit adressen zijn in het leven geroepen omdat de meeste adressen binnen de 7-bits adresruimte al bezet zijn, daarom is deze modus gemaakt. Deze modus is compatibel met de 7-bit adresmodus. Zodat 7-bit en 10-bit adressen op dezelfde bus kunnen aangesloten worden. Het idee van het 10-bit adres is dat er twee adresbytes worden gebruikt. De eerste adresbyte bevat een combinatie die gereserveerd is voor de wijze van adres uitbreiding.

Daarna volgt een tweede adresbyte dat de verdere adresbits bevat. De richtingsbit krijgt hierbij een andere betekenis. De 7-bits in het eerste adresbyte hebben de vorm 11110XX. XX stelt de twee hoogst waardige adresbits van het 10-bits adres voor. De andere acht bits volgen in het tweede adresbyte. De geadresseerde slave moet beide bytes bevestigen met een acknowledge bit. Bij de verwerking van het general call address en de startbyte procedure moet de slave reageren als een 7-bit bouwsteen. Dat betekend echter geen beperking en is gewoon logisch.

15.12.9. CONDITIES

De I²C-bus heeft twee actieve lijnen en een massa verbinding. De actieve lijnen zijn SDA en SCL. Beide lijnen zijn bidirectioneel. SDA is de Serial Data lijn en SCL is de Serial CLock lijn. Elk component die aan de lijn verbonden is heeft een uniek adres. Elk component kan afhankelijk van zijn functie zowel als zender al ontvanger werken. De master is het ic dat de commando's op de bus plaatst. In de I²C protocol specificatie staat dat het IC dat de dataoverdracht begint, de master is. Elk ander IC is dan een slave. De I²C bus is een Multi master bus. Dit houdt in dat er meer dan één IC de mogelijkheid bezit om de dataoverdracht te beginnen. De master IC's zijn meestal microcontroller die het geheel besturen.

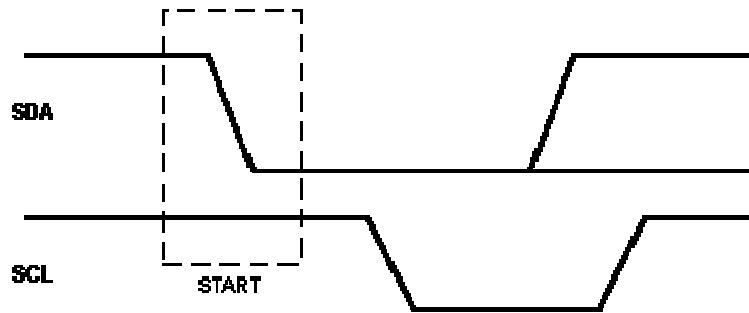


Figuur 15: blokschema I²C

15.12.10. START-CONDITIE

Om dataoverdracht te verkrijgen op de bus moet er een start conditie voorafgaan. De start conditie werkt als een attentie signaal aan alle IC's op de bus dat er dataoverdracht gaat komen. Het resultaat hiervan is dat alle IC's luisteren naar de data op de bus.

De start conditie wordt altijd gegeven door de master. Na de start conditie wordt de bus als bezet beschouwd. De bus komt weer vrij na een stop conditie.

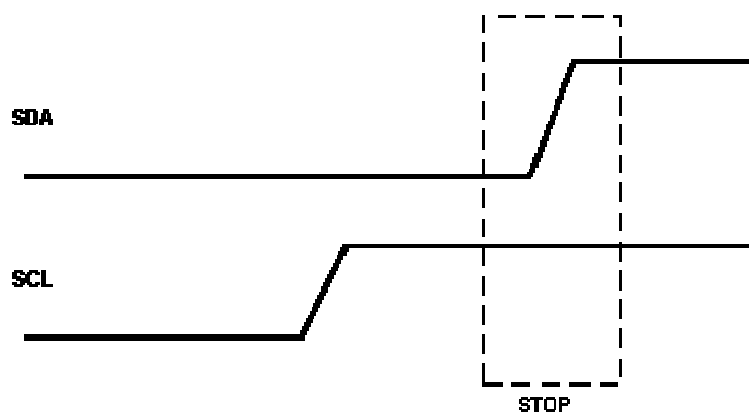


Figuur 16: Start-conditie

Merk op: de startconditie is een overgang van hoog naar laag op de SDA lijn terwijl de SCL lijn hoog blijft. Een boodschap kan meerdere start condities bevatten. Het gebruik van deze herhaalde start is algemeen in I²C.

15.12.11. STOP-CONDITIE

De stop conditie wordt door de master gegenereerd. Met de stop conditie wordt de bus weer vrijgegeven.



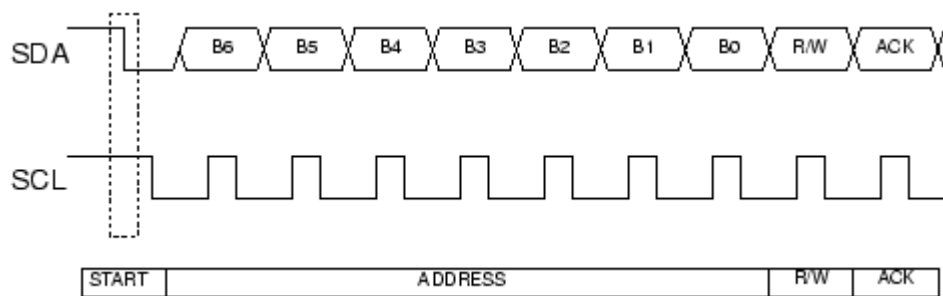
Figuur 17: stop-conditie

Merk op: bij een stop conditie gaat de SDA lijn over van laag naar hoog terwijl de SCL lijn hoog blijft.

Een stop conditie geeft altijd het einde van een boodschap aan. Na een stop conditie wordt er door de IC's geen data meer ingelezen, maar wachten alle IC's op een nieuwe start conditie. Een bijzonderheid is de zogenaamde repeated start condition. Hierbij beëindigt de master de dataoverdracht niet door een stopconditie, maar maakt gewoon weer een nieuwe startconditie. Dat zorgt ervoor dat het programma sneller verloopt.

15.12.12. ADRES-PCONDITIE

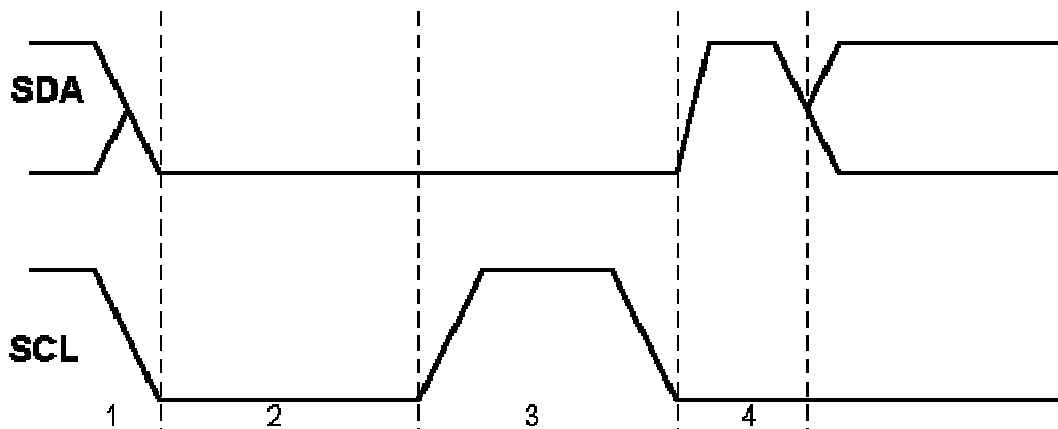
Na een start conditie gaat de master een byte op de bus plaatsen met het adres van het gewenste IC.



Figuur 18: Adres

15.12.13. ACKNOWLEDGE VAN EEN SLAVE

Als een master een byte naar een slave zendt, dan moet er na elke byte een acknowledge bit van de slave komen. Door deze bit verteld de slave dat hij de overdracht ontvangen heeft en erkend. Elke byte die op de SDA lijn gezet wordt moet 8 bits lang zijn. Het aantal bytes dat in een boodschap verzonden wordt is onbeperkt, maar elke byte moet gevolgd worden door een acknowledge bit.

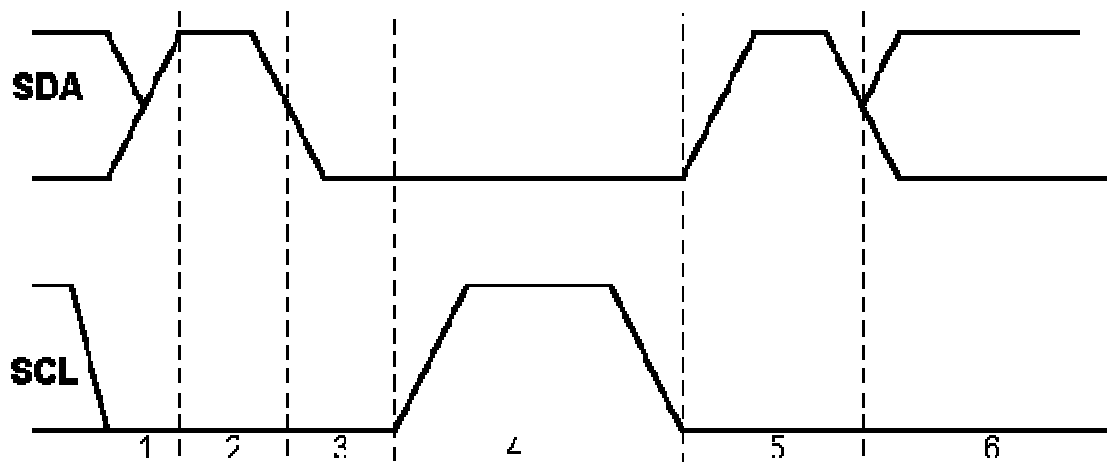


Figuur 19: situatie

Zodra de master de SCL-lijn laag trekt aan het einde van de bit (1) zal de slave de SDA-lijn ook laag trekken (2). De master zet dan een klokpuls op de SCL-lijn (3). Gedurende deze klokpuls moet de SDA-lijn laag blijven. Na de klokpuls laat de slave de SDA lijn terug vrij (4).

ACKNOWLEDGE VAN EEN MASTER

Als een master een byte van een slave ontvangt, dan moet hij een acknowledge bit aan de slave geven. De master controleert zowel de SDA als de SCL lijn.

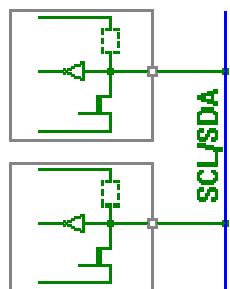


Figuur 20: situatie

Als de slave de laatste bit naar de master heeft verzonden (1), dan laat hij de SDA-lijn los. De SDA-lijn moet dan hoog worden (2). De master trekt de SDA-lijn nu opnieuw laag (3). Daarna genereert de master een klokpuls op de SCL-lijn (4). Als de klokpuls afgelopen is, laat de master de SDA-lijn terug vrij (5). De slave heeft nu opnieuw controle over de SDA-lijn (6).

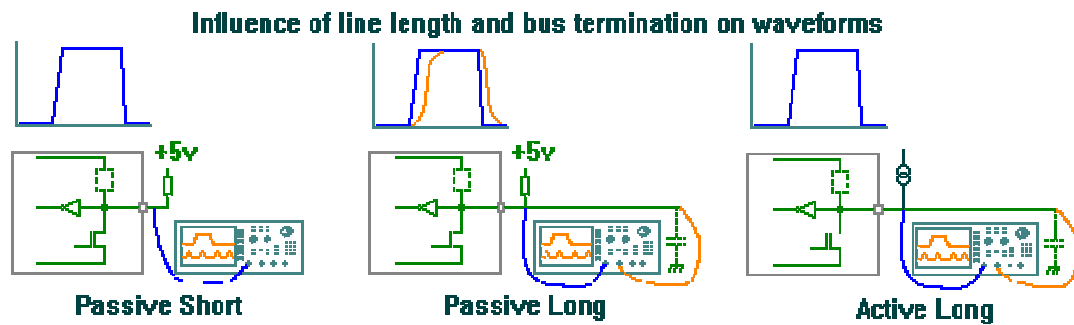
HARDWARE

I²C maakt gebruik van twee actieve lijnen, SCL en SDA. Beide lijnen zijn bi-directioneel. Om te voorkomen dat een IC doorbrandt, gebruiken deze signalen een open collector op een open drain output. Dit hangt af van de gebruikte techniek.



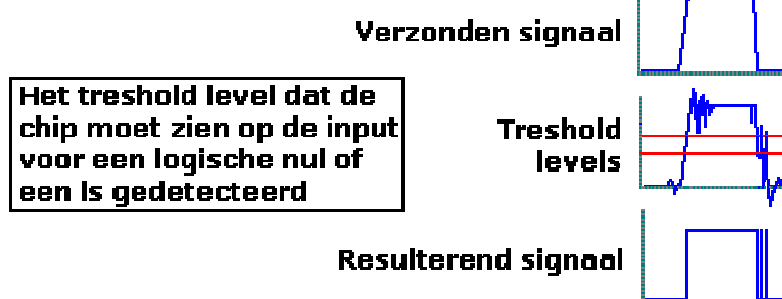
De businterface is opgebouwd rond een input buffer en een open drain of een open collector transistor. Als er niet gebeurt op de bus zijn de lijnen logisch '1'. Een veel voorkomende fout is dat externe pull-up weerstanden vergeten worden. Deze zijn nodig om te voorkomen dat het niveau van de logische 1 onder de minimale waarde komt en aangezien wordt voor een logische '0'. Om iets op de bus te zetten, trekt de IC met de transistor de bus naar een logische '0'.

Deze techniek wordt bus mastering genoemd. Als de bus bezet is door een IC die een logische '0' uitzend verliezen alle andere chips hun mogelijkheid tot het zenden van informatie op de bus.

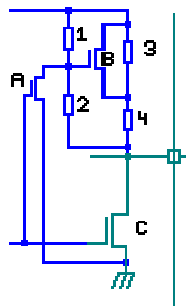


Een nadeel van de open collector techniek is dat bij een lange bus de snelheid achteruit gaat. Lange lijnen geven een capacitieve belasting aan de uitgang. Omdat de pull-up weerstanden passief zijn, krijg je een RC constante die zich manifesteert in de vorm van de signalen. De signalen worden door deze constante minder scherp. Des te hoger deze RC constante is, hoe langzamer de bus wordt. Op een gegeven moment kan de IC niet meer duidelijk de overstap tussen de logische 0 en 1 onderscheiden.

Invloed van slechte lijn afsluiting: Reflectie



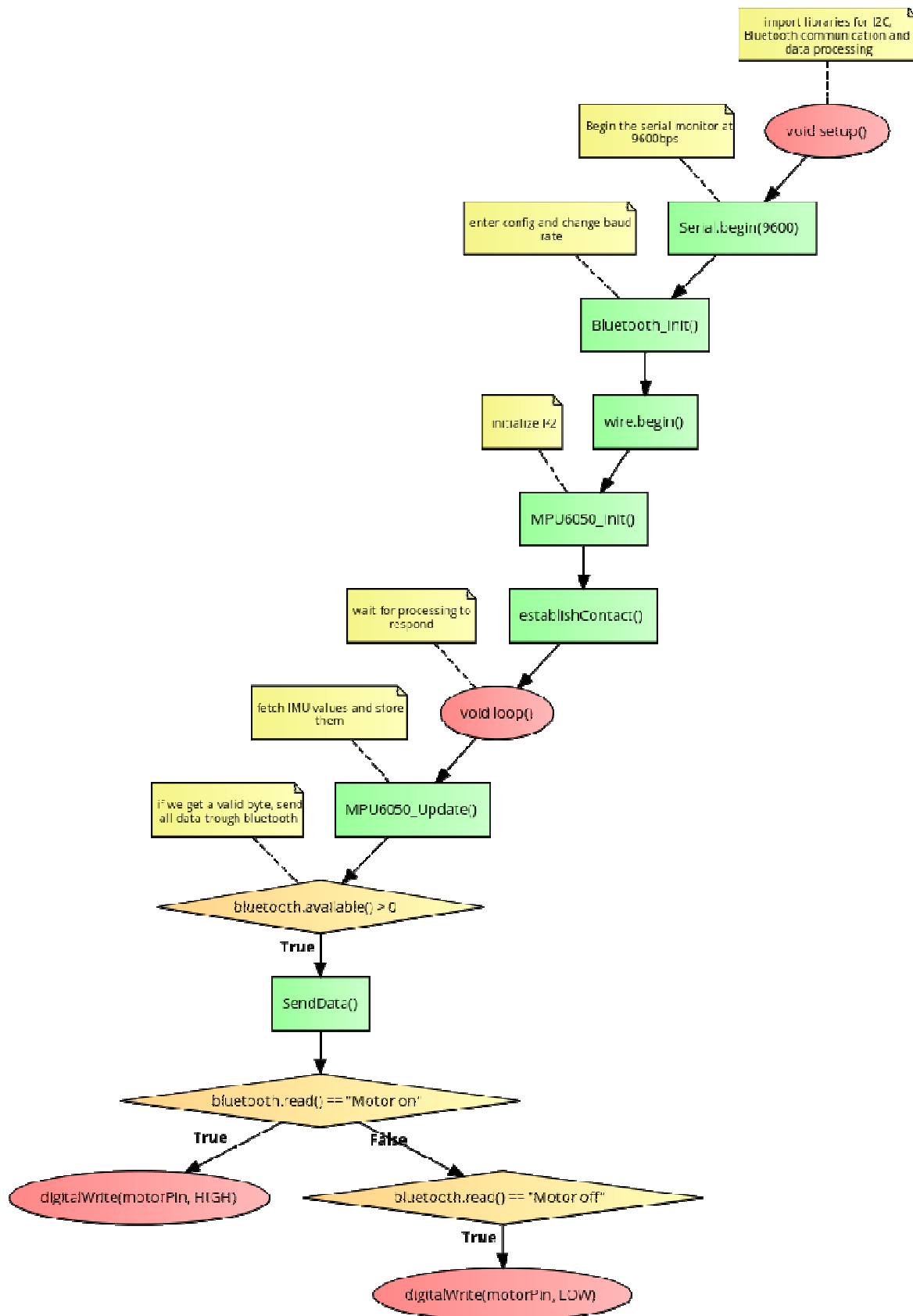
Een ander probleem is dat je bij zeer hoge snelheden last kan krijgen van reflecties. Dit kan zo erg zijn dat spooksignalen de transmissie kunnen verstoren en de data die verstuurd wordt, wordt beïnvloed. Zelfs de schmitt trigger aan de ingang van de chip helpt dan niet meer afdoende.



Om dit probleem op te lossen heeft Philips een actieve I²C terminator ontwikkeld. Deze bestaat uit twee charge pumps. Je kan dit apparaat zien als een dynamische weerstand. Op het moment dat de logische '1' een '0' wordt geeft het een grote stroom (lage dynamische weerstand) op de bus. Hierdoor kan de parasitaire capaciteit van de bus snel worden geladen. Zodra de spanning boven een bepaald niveau komt, wordt de hoge stroom uitgezet en de uitgangstroom valt scherp af.

15.13. SOFTWARE

De laatste nieuwe software is steeds beschikbaar op GitHub, volg deze link <http://goo.gl/R20y9v>.



16. VISUALISATIE SOFTWARE

We hebben besloten om ook visualisatie software te maken. We zijn eerst begonnen in Visual Studio met de programmeertaal Visual Basic .Net. Maar dit hebben we relatief snel opgegeven omdat we geen goede bibliotheken vonden voor onze doelen.

We zijn toen overgestapt naar Processing. Dit is een open-source programma met een grote gemeenschap. Hier vonden we wel enkele interessante bibliotheken. Voor de GUI (Graphical User Interface) hebben we G4P gebruikt. Voor deze bibliotheek is ook een tool gemaakt de GUI Builder. Hiermee kan de GUI gemaakt worden door objecten op het scherm te slepen i.p.v. alles te programmeren. Dit heeft het werk versneld. Voor de visualisatie hebben we een bibliotheek gekozen die verschillende grafieken ondersteunde, GWOPTICS. Echter, conflicteerde deze met de G4P bibliotheek omdat we externe schermen gebruiken. Als oplossing zijn we overgestapt naar Grafika. Dit is ook een grafieken bibliotheek maar deze heeft niet zo'n rijke tools. Men kan gegevens verkrijgen van de IMU via de seriële kabel of Bluetooth. De data kan gevisualiseerd worden d.m.v. grafieken, een geanimeerd arm of via gewone uitgeprinte gegevens. Het programma is zo gemaakt dat elke actie aan de GUI gelogd wordt in een bestand voor debug doelstellingen.

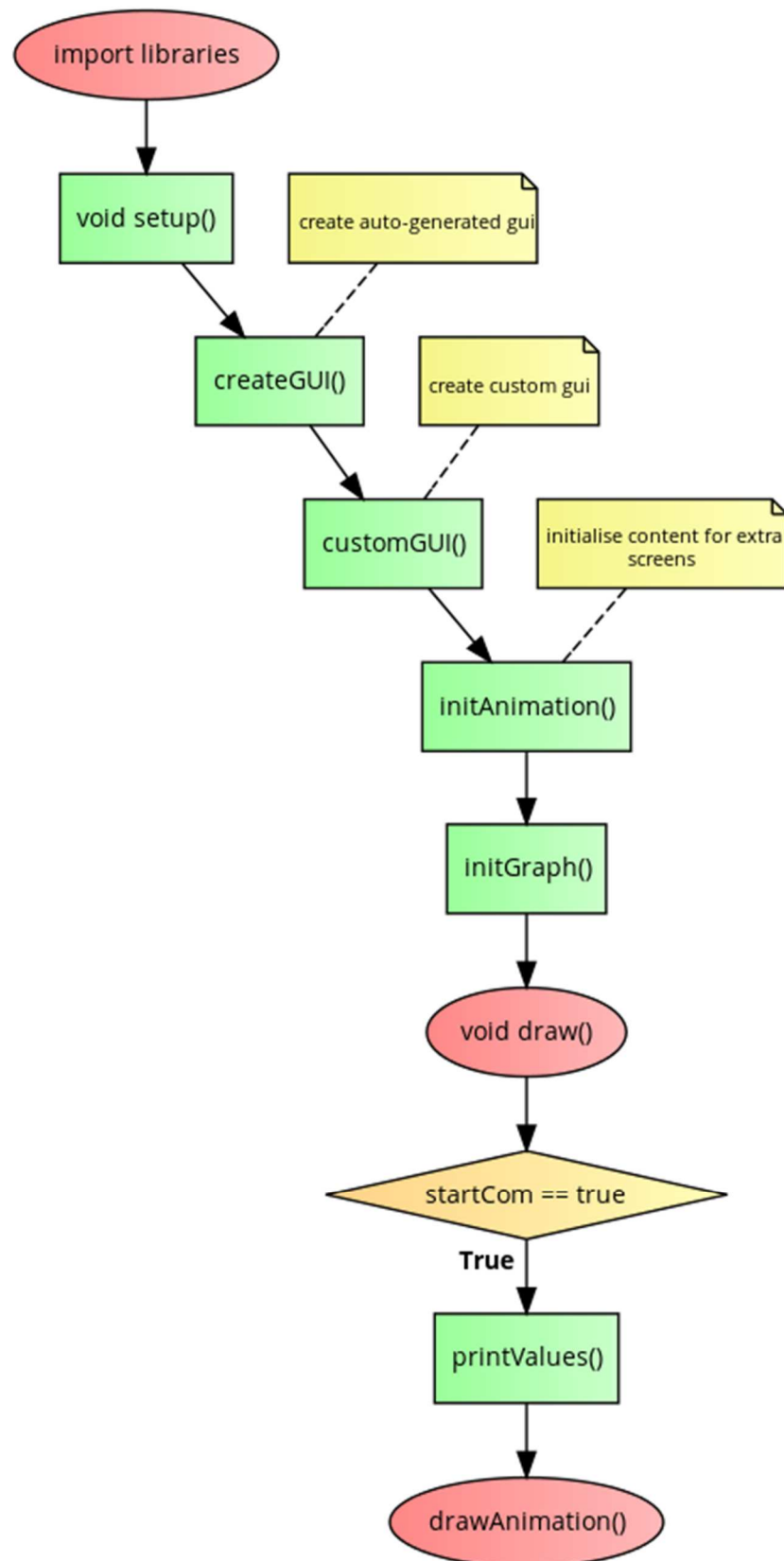
16.1. GUI

De Graphical User Interface werd gemaakt met G4P.



Figuur 21: Graphical User Interface

De laatste nieuwe software is steeds beschikbaar op GitHub, volg deze link <http://goo.gl/R20y9v>.



17. BRONNEN

17.1. SOFTWARE

- Arduino:
<http://goo.gl/Q9Tr>
- Processing:
<http://goo.gl/wh7Z>
- Code2flow
<http://goo.gl/Vk5Cxt>

17.2. ARDUINO BIBLIOTHEKEN

- Kalman:
<http://goo.gl/4Nuglj>

17.3. PROCESSING BIBLIOTHEKEN

- G4P:
<http://goo.gl/blloi3>
- Grafica:
<http://goo.gl/s2hHMY>

17.4. PROJECT WEBSITES

- Hoofd website:
<http://goo.gl/FvoF4U>
- GitHub:
<http://goo.gl/R20y9v>