

Ontwikkeling van een robotische handprothese

Jelle van Bost

Nathan Guirado

Yochem van Rosmalen

Profielwerkstuk

11 januari 2018

Informatica en biologie

Ds. Pierson College, 's-Hertogenbosch

Begeleid door
Dhr. P. Cools &
Dhr. W. Vrouwenfelder

Inhoudsopgave

1 Inleiding	2
2 Theorie	3
2.1 Hedendaagse protheses	3
Soorten Protheses	3
Beenprothese	3
Implantaat	4
Armprothese	4
Overige protheses	5
2.2 Anatomie van de hand	6
Beenderen	6
Spieren en pezen	6
Zenuwen en aderen	7
Gouden ratio in de anatomie van de hand	7
2.3 Materialen en software	9
Materialen aan de buitenkant van de hand	9
Materialen in de hand	9
Software	10
2.4 Nabootsing menselijke bewegingen	12
Welke bewegingen	12
De materialen	12
Combineren	12
3 Documentatie	13
3.1 Ontwerpen en printen	13
3.2 De werking van de hand	15
Clone	15
Overige hardware	15
Het aansturen van de hand	16
Elektriciteitsschema	17
3.3 Programmeren	18
Ontwikkelingsomgeving	18
Versiebeheersysteem	18
Toelichting op code	19
4 Conclusie	22
5 Reflectie	23
Referenties	24
Bronvermelding bij figuren	27
Bijlage	30

1 Inleiding

Jaarlijks ondergaan er ongeveer 3200 mensen in Nederland alleen al een beenamputatie (Feenstra, z.j.). Sommige van deze mensen krijgen na de beenamputatie een prothese, dit is een kunstmatige vervanging van een lichaamsdeel. Protheses worden veel en al lang gebruikt. Ook wordt er nog veel onderzoek gedaan naar protheses. Zo zijn veel protheses tegenwoordig mechanisch en lijken ze goed op het oorspronkelijke lichaamsdeel. Een nadeel aan protheses is dat deze vaak erg duur zijn. De kosten komen door de ontwikkeling die veel tijd in beslag neemt en er geen massaproductie mogelijk is, omdat protheses op maat gemaakt moeten worden voor de gebruiker.

Het doel van dit profielwerkstuk om een zo goedkoop mogelijke mechanische handprothese te ontwikkelen. In dit profielwerkstuk wordt daarom onderzocht of het mogelijk is om een robotisch aangestuurde 3D-geprinte armprothese te maken. Er wordt gekeken naar de keuze van materialen, andere manieren van produceren en een verbeterde algemene werking. Ook zien we graag dat mensen bij kunnen dragen aan onze prothesehand. Alles wat wij maken zal dus openbaar en gratis beschikbaar zijn.

Allereerst wordt de anatomie van de hand uitgelegd. Vervolgens de gebruikte materialen en software en daarna hoe de materialen en software de menselijke bewegingen kunnen nabootsen. Vervolgens wordt het onderzoek beschreven en daaruit volgt een conclusie. Als laatste wordt gereflecteerd op het gemaakte profielwerkstuk.

2 Theorie

2.1 Hedendaagse protheses

Protheses zijn een erg goede vervanging voor ledematen die zijn beschadigd bij mensen of zelfs missen. Dit geldt voor erg veel delen van het lichaam, die elk kort zullen worden behandeld. Er bestaan verschillende soorten protheses. Van simpele nabootsingen van het desbetreffende ledemaat tot volledig robotisch aangedreven protheses. Voor organen bestaan er ook protheses, dit worden meestal implantaten genoemd, omdat de prothese helemaal is weggewerkt in het lichaam.

Soorten Protheses

Je hebt veel verschillende soorten protheses, bijvoorbeeld een beenprothese, borstprothese, armprothese, oogprothese, voetprothese en handprothese. Ook zijn er hele bekende protheses, die in de volksmond geen protheses meer worden genoemd. Denk aan een gehoorprothese (gehoorapparaatje), een oogprothese (bril of lenzen) en tandprotheses (kunstgebit). Elk van de eerdergenoemde protheses heeft een ander doel en worden dus ook op een andere manier gemaakt, en meestal ook met andere materialen.

Beenprothese

In de categorie beenprotheses bestaan er ook verschillende protheses, zo is er een voetprothese, een onderbeenprothese met eventueel een vervangende enkel, een bovenbeenprothese met een vervangende knie en enkel en een totale beenprothese. Bij een amputatie onder de knie wordt een onderbeenprothese het meeste toegepast. De onderbeenprothese bestaat uit een koker die de stomp (waar de amputatie is verricht) nauwsluitend omvat. De koker staat weer in verbinding met een prothesevoet, met eventuele adapters. Er zijn veel verschillende soorten prothesevoeten, hele simpele prothesevoeten, van simpel materiaal, tot prothesevoeten van veerkrachtig materiaal die de energie van een stap opslaan om die te gebruiken als je je weer afzet voor krachtigere stappen. Ook bestaan er volledige mechanische beenprotheses, werkend met meerdere microcontrollers. Een totale beenprothese wordt pas aangeraden als er een totale beenamputatie heeft ondervonden. Het principe is hetzelfde als hierboven is beschreven bij de onderbeenprotheses, alleen is er nu nog een extra koker, voor het bovenbeen en een knie. De kunstknie heeft een belangrijke taak, namelijk voorkomen dat de kokers onbedoeld kunnen buigen en de gebruiker omvalt. Er zijn hiervoor speciale kunstknieën ontworpen met een hydraulische of pneumatische vertraging, bij hele geavanceerde knieën wordt het bijna elk aspect van een stap elektronisch geregeld.



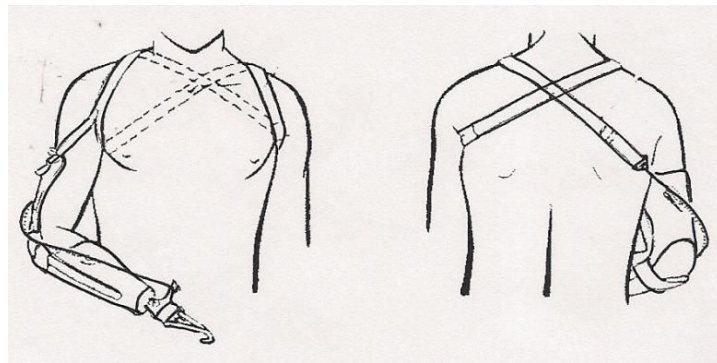
Figuur 1: Simpele beenprothese

Implantaat

Een implantaat is een prothese dat helemaal is weggewerkt in het lichaam (Implantaat, z.j.). Enkele voorbeelden hiervan zijn een hart-, heup- en knie-implantaat. Ook zijn er bijvoorbeeld elektrische implantaten, zoals pacemakers of een implanteerbare cardioverter-defibrillator. De cardioverter-defibrillator kan een elektrische schok geven aan het hart als het een ritmestoornis heeft. Dit zijn enkele voorbeelden van implantaten.

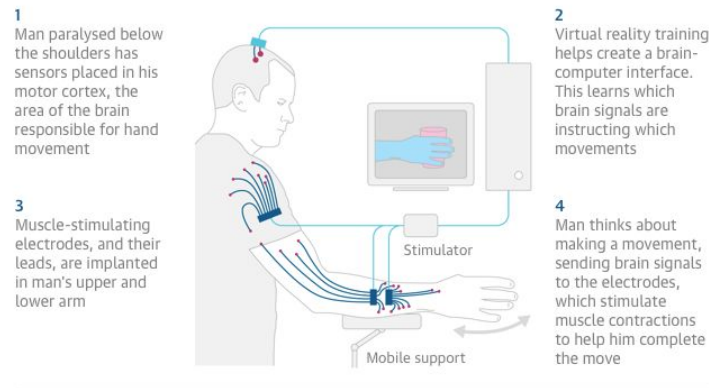
Armprothese

Ook bij armprothesen bestaan er verschillende soorten prothesen. Van vingerprothesen tot volledige arm prothesen. De meest gebruikte techniek om een armprothese te laten passen is om nadat de stomp een stabiel volume heeft gekregen er een gipsafdruk van te maken. Daaraan wordt er een stompkous gemaakt die om de stomp heen past en die men vast kan maken aan de prothese (Armprothese, z.j.). Een van de vereisten van een prothese is dat deze bruikbaar is in het dagelijks leven. Dit houdt in dat men er voorwerpen mee kan vastpakken, vasthouden, een vinger op kan steken en dergelijke acties. Dit moet aangestuurd worden. Er zijn verschillende manieren om dit te doen, bijvoorbeeld door het gebruik van de torso. Door deze in bepaalde richtingen te bewegen, trek je via een harnas de connectie strak en beweegt de hand. De 'Body-powered Prosthesis' is goed geïllustreerd in figuur 2, waar het harnas en de connectie met de hand te zien is.



Figuur 2: Een body-powered prothese met harnas

Ook is de hand aan te sturen via de elektrische signalen uit het brein. Dit noemt men een neuroprothese. Hier moet de gebruiker mee leren werken. Ook moet de microcontroller van de hand leren welke acties bij welke inputs horen. De werking hiervan is omschreven in figuur 3.



Figuur 3: Leren een neuroprothese te gebruiken

De derde vorm van het aansturen van een handprothese is via de armspieren. Deze protheses worden *Myoelectric Controlled Prostheses* genoemd. Deze protheses meten elektromyografie (EMG) signalen vanuit de bovenarm. Hier lopen spieren die eigenlijk de hand aan zouden moeten sturen. De buigspieren, waar de sensoren worden geplaatst, zitten voornamelijk aan de anteriore¹ kant van de arm, met hun antagonisten aan de posterioire² kant van de arm (Taylor, 2017). Deze spieren zijn verlengden van de hand en vingers en lopen als lange, dunne stroken door de arm. Als hier sensoren op worden geplaatst, zijn de elektrische signalen die deze spieren afgeven te meten. Als de signalen versterkt worden, kan de microcontroller het versterkte signaal gebruiken om de vingers op juiste wijze aan te sturen.



Figuur 4: Moderne handprothese

Overige protheses

Onder overige protheses vallen gehoorprotheses, oogprotheses en tandprotheses. Deze protheses zijn meestal niet ter vervanging van het desbetreffende lichaamsdeel, maar omdat dit beschadigd is. Een gehoorapparaat versterkt de geluidsimpulsen waardoor iemand met een gehoorbeschadiging toch goed kan horen. Een bril breekt het licht op een bepaalde manier waarop je toch goed kan zien en een kunstgebit wordt over je tandvlees of resterende tanden geplaatst waardoor ze eruit zien als nieuw en het eten en spreken weer een stuk beter gaat. Onder overige protheses valt ook een borstprothese. De borstprothese wordt ingezet als iemand als gevolg van bijvoorbeeld borstkanker, haar borst heeft moeten laten amputeren. Deze protheses bestaan voor het merendeel uit siliconen.

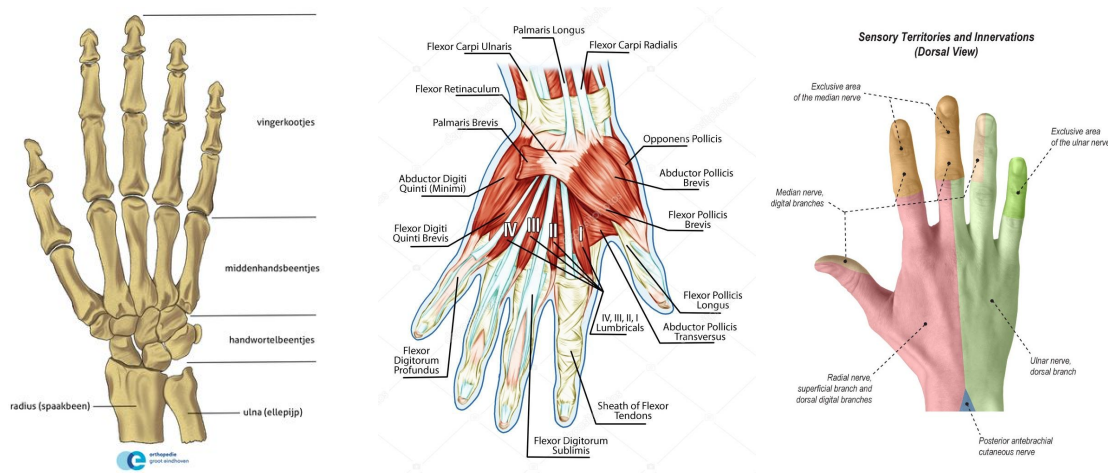
¹ Vooraanzicht lichaam

² Achteraanzicht lichaam

2.2 Anatomie van de hand

Handen zijn uiterst belangrijk voor de mens, omdat mensen met de hand in staat hun omgeving te manipuleren en ook essentiële levensbehoeften vervullen, zoals eten en drinken. Handen zijn erg kenmerkend voor de mens. De menselijke hand wijkt erg af van wat vele dieren hebben.

De menselijke hand wordt gekenmerkt door opponeerbare duimen, wat betekent dat mensen met het topje van hun duim hun andere vingers aan kunnen raken (Hand, in *Wikipedia*). Bij veel dieren is dit niet het geval.



Figuur 5: Anatomie van de hand. 5.1: Beenderen, 5.2: Spieren en pezen en 5.3: Zenuwen

Beenderen

De hand bestaat wel uit 27 kleine botjes (Natuurinformatie - Hand, z.j.). De botjes zorgen ervoor dat de handen, en met name de vingers ervan, erg beweegbaar zijn. De duim kan wel 90 graden worden gedraaid, in tegenstelling tot de andere vingers. Deze kunnen ongeveer 45 graden worden gedraaid. De onderarm eindigt met de ellepijp en het spaakbeen. Hier gaat het over naar de pols, deze bestaat uit acht handwortelbeentjes. Hier zijn alle vingers aan bevestigd. Boven de handwortelbeentjes, in de middenhand, bevinden zich de middenhandsbeentjes. Dit maakt de pols beweegbaar, zodat het makkelijker is om voorwerpen vast te grijpen. De vingers bestaan uit vingerkootjes. Bij elke vinger zijn dit er drie, behalve de duim, deze heeft er twee. Hierdoor is een vinger in drie delen te krommen. In de geneeskunde en anatomie worden deze vingers vaak aangeduid met romeinse cijfers. Dit loopt van digitus I (de duim) tot digitus V (de pink) (Alpenfels, 1955).

Spieren en pezen

De spieren in de hand worden verdeeld onder twee soorten: de intrinsieke en extrinsieke banden. Intrinsieke banden zijn kort en stijf en bevinden zich vooral in de handpalm. Deze spieren hoeven namelijk niet sterk uit te rekken. Extrinsieke banden zijn soepeler en een stuk langer. Zij verbinden de onderarmbotten met de middenhandsbeentjes (Mortel , 2009). Wanneer deze worden aangespannen, buigen de vingers.

Ook bevindt het CMC-gewricht zich in de hand. Dit gewricht maakt het mogelijk de duim in oppositie te bewegen, dit betekent dat de duim naar de pink wordt bewogen (Oosterbos, z.j.). Hierdoor kan men de duim in veel meer verschillende posities houden dan de andere vier vingers. Dit maakt de menselijke hand ook zo complex.

In de hand bevinden zich ook pezen die het te ver terugbuigen van de vingers voorkomen. Als deze er niet waren, zouden de vingers makkelijk de verkeerde kant op buigen, hetgeen zorgt voor het scheuren van andere spieren.

Zenuwen en aderen

De hand is voorzien van drie gemengde zenuwen: de nervus ulnaris, nervus medianus en nervus radialis. Deze zenuwen sturen de spieren in de hand en onderarm aan. De zenuwen hebben alle drie zowel motorische ('bewegende') als sensibele ('voelende') functie. Deze sensibele functie laat een hoge haptische waarneming toe, wat betekent dat je erg veel kan voelen met de hand. Bijvoorbeeld druk-, bewegings- en vibratiestimulansen. De bloedtoevoer wordt verzorgd door twee slagaders: de arteria radialis en arteria ulnaris.

Gouden ratio in de anatomie van de hand

De gouden ratio of gulden snede is een speciale wiskundige verdeling van een lijnstuk. De gouden ratio is al vanaf de oudheid bestudeerd. Dit omdat het vaak in de natuur voor blijkt te komen. De gouden ratio op lijnstuk AB (figuur 6) met S als de gouden ratio is te definiëren als:

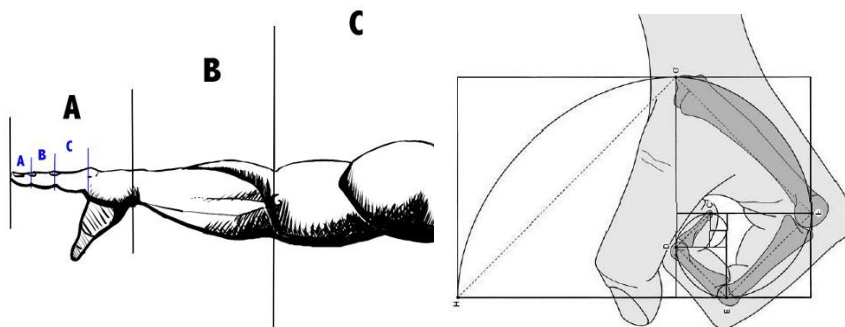
$$\phi_1 = \frac{1+\sqrt{5}}{2} \approx 1,618...$$



Figuur 6: Lijnstuk AB

Dit ratio blijkt massaal voor te komen in de natuur. De gouden ratio wordt door velen gezien als de meest esthetische verhouding in zowel kunst, natuur en design (Osborne, 1999). In de natuur kan men denken aan de groente romanesco en verschillende bloemblaadjes.

Maar ook in de anatomie blijkt de gulden snede voor te komen. Bijvoorbeeld in de arm. Zie figuur 7. Onze vingers bestaan uit 3 delen. De eerste twee delen (blauwe B en C in figuur 7). Zijn in gouden ratio met de totale lengte van de vinger. Ook staan de middelvinger en pink in deze verhouding met elkaar. (Murali, 2012)



Figuur 7: Gouden ratio in anatomie

De rij van Fibonacci houdt sterk verband met de gulden snede. Deze interessante reeks begint met 0 en 1, waarbij elk volgend element de som van beiden elementen hiervoor is. Dit geeft voor de eerste tien getallen de volgende reeks:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... ("Fibonacci", in *Wikipedia*)

Als men dit vervolgens plot op een (oneindig) rechte lijn is het gouden ratio duidelijk te visualiseren. Ook nadert de reeks het gouden ratio als zijnde een limiet.

2.3 Materialen en software

Een hand is erg complex. Een nabootsing van een hand kost dan ook onwijs veel tijd, energie en materialen om te maken. Bij het maken van een prothese moet er veel rekening worden gehouden met de materiaalkeuze. Ook is het beperken van de onderdelen belangrijk.

Materialen aan de buitenkant van de hand

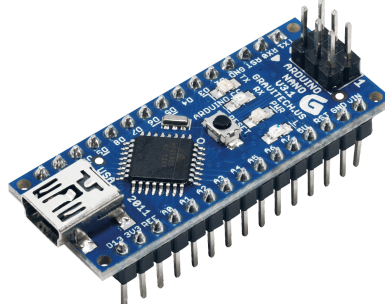
Bij het kiezen van een printmateriaal, zijn er een paar overwegingen die er moeten worden gemaakt. Je wilt dat het materiaal sterk is. Het moet sterk genoeg zijn om niet zomaar kapot te gaan maar tegelijkertijd zacht genoeg dat je het nog wel bewerkbaar is. Ook moet het 3D printbaar zijn. Dit zijn best wat vereisten voor het materiaal. Het belangrijkste voor ons is dat het 3D printbaar is, omdat het doel is dat de prothese ge-3D-print wordt. De materialen hierin zijn beperkt. ABS is beter hittebestendig en sterker. Het enige nadeel is dat het moeilijker printen is (Chilson, 2013).



Figuur 8: Verschil PLA en ABS

Materialen in de hand

Er is ook ander materiaal nodig voor de prothesehand. Bijvoorbeeld motortjes, draadjes, stroomdraadjes en een microcontroller of microcomputer, deze sturen alles in de hand aan. Er zijn erg veel verschillende motors op de markt. In een prothesehand heeft men deze het liefst zo klein mogelijk, met toch aardig wat kracht. De ruimte is namelijk minimaal en motors nemen al gauw flink wat ruimte in. Ook de draad moet krachtig zijn. Visdraad is erg sterk en dun, wat het een perfect materiaal maakt. De twee meest voorkomende microcontrollers tegenwoordig zijn de Raspberry Pi en Arduino. Dit zijn de enige twee controllers die zijn onderzocht, omdat het grootste marktaandeel uit deze twee controllers bestaat. Raspberry Pi is een microcomputer en past in je handpalm. Arduino heeft ook meerdere microcontrollers met deze grootte, tevens hebben zij ook een nog kleinere microcontroller: de Arduino Nano. Deze is de grootte van een pink. Dit maakt haar de perfecte microcontroller voor een prothesehand. Het grote verschil is dat de Nano geen poorten (USB, HDMI...) heeft. Er kunnen dus niet makkelijk USB-apparaten aan worden gekoppeld. Alles moet hierdoor via pins, de metalen pinnetjes aan de onderkant in figuur 9.

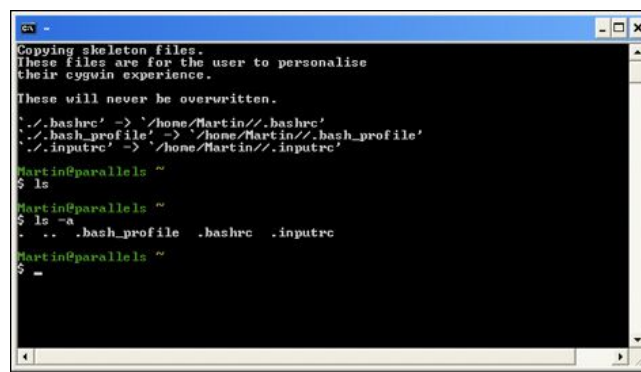


Figuur 9: Arduino Nano v3

Software

De software voor het productieproces wordt vooral bepaald door de microcontroller. Vaak is een bepaalde microcontroller afhankelijk van één programmeertaal en met een bepaald ontwikkelproces uitgedacht. Om het simpel te houden gaat het nu verder in op de twee meest voorkomende microcontrollers: de Raspberry Pi en de Arduino.

De Raspberry Pi is een microcontroller die draait op een speciale versie van Linux (een open source³ besturingssysteem), genaamd Raspbian. Dit besturingssysteem of operating system (OS) is speciaal ontwikkeld voor de Pi (Raspbian, z.j.). Gezien het feit dat het een Unix-systeem is, werk je vooral met de terminal. Een herkenbaar aspect van een dit systeem is de centrale rol van het bestandssysteem (Unix, in *Wikipedia*). Om met dit bestandssysteem te werken maakt men gebruik van kleine programma's die gecombineerd worden in een shell (figuur 10). Unix wordt vaak geassocieerd met een 'Command Line Interface'. Dit betekent dat de computer wordt bestuurd met korte getypte opdrachten in plaats van het klikken op knoppen en slepen van bestanden.



```
GN -
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

These will never be overwritten.
'./bashrc' -> '/home/Martin/./bashrc'
'./bash_profile' -> '/home/Martin/./bash_profile'
'./inputrc' -> '/home/Martin/./inputrc'

Martin@parallels ~
$ ls
Martin@parallels ~
$ ls -a
.  ..  .bash_profile  .bashrc  .inputrc
Martin@parallels ~
$
```

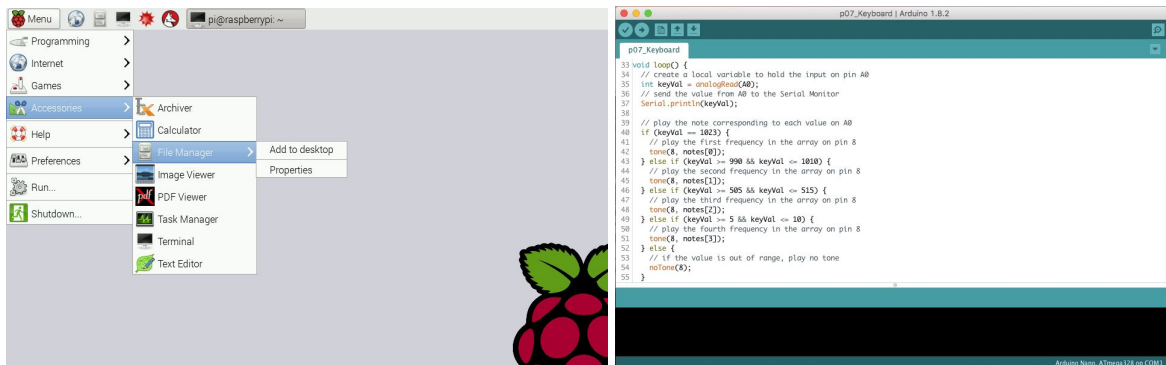
Figuur 10: de Shell

Verder ziet de Raspberry Pi er uit als een normale Windows computer, met een bureaublad, applicaties, een browser en een dock. Over het algemeen programmeer je Raspberry Pi's in Python. Dit is een in Nederland ontwikkelde programmeertaal met een erg overzichtelijke syntax.

De Arduino is daarentegen geen microcomputer, maar een microcontroller (Di Justo, 2015). Dit verschil uit zich in het niet hebben van een user interface (UI). Als men een Arduino dus via een hdmi-kabel aan een extern beeldscherm hangt, gebeurt er niks. Een Arduino is enkel bedoeld voor dingen als robotica en home automation (het automatisch besturen van elektronische onderdelen van je huis). De Arduino is daarentegen wel weer erg simpel in gebruik, vooral gecombineerd met de Arduino IDE⁴ (gebaseerd op Processing) en de Arduino programmeertaal (gebaseerd op Wiring) (Arduino Foundation, 2017). Doordat de software, programmeertaal en hardware zo op elkaar is afgestemd, gaat het proces een stuk makkelijker over het algemeen.

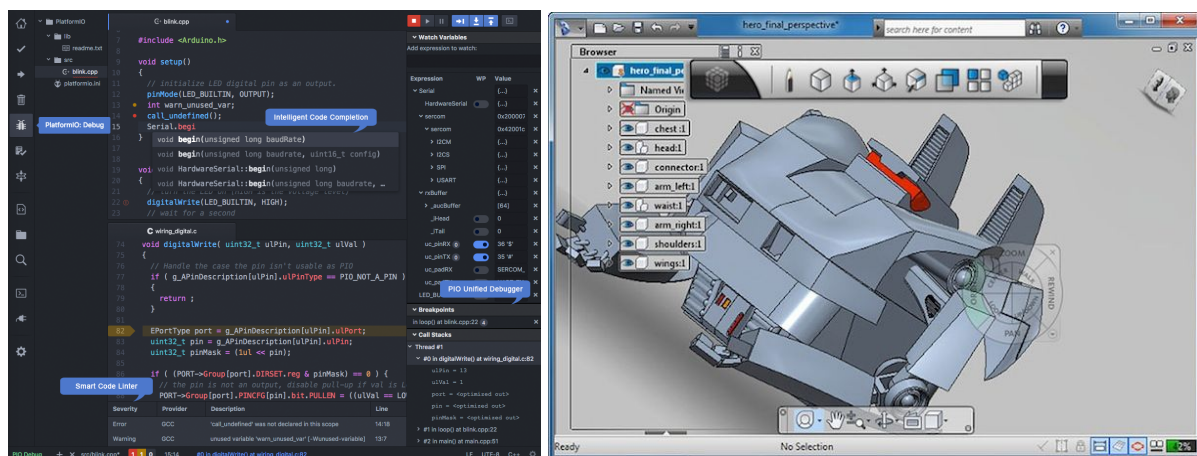
³ Software waarvan iedereen toegang heeft tot het bronmateriaal (de 'source')

⁴ Integrated Development Environment; de software die men gebruikt om software te ontwikkelen



Figuur 11: De Raspberry Pi UI | Figuur 12: De Arduino IDE

Voor programmeren is een goede, functionele IDE erg handig. Dit is software om het ontwikkelen van software te vereenvoudigen. Een veelgemaakte keuze hiervoor is de Atom text editor (Henry, 2014). Dit is enkel een text editor, maar door de vele packages die downloadbaar zijn, is Atom inmiddels volledig om te bouwen tot uitgebreide IDE. Atom is ontwikkeld door Github, en daarom ook volledig open source. Echter is Atom nog niet gebouwd voor microcontrollers, zoals de Arduino IDE is. Hiervoor zijn packages beschikbaar. PlatformIO (Kravets, 2017) is hier een van. Deze voegt verschillende tools en packages toe die het programmeren van een microcontroller een stuk makkelijker maken. Zo kan men builden (de code testen op fouten en omzetten naar C++) en uploaden (van de computer naar de geheugenkaart van de microcontroller plaatsen), rechtstreeks vanuit Atom.



Figuur 13: Atom met PlatformIO package. | Figuur 14: Autodesk 123D software voor Windows

Het ontwerpen van 3D-afbeeldingen van de hand vereist ook speciale software. Hiervoor zijn er erg veel verschillende applicaties (Duann, 2012). Zo zijn er computerapplicaties, maar ook webapplicaties. Deze applicaties hebben wat weg van Adobe Photoshop. Het zijn over het algemeen programma's met veel tools, zoals te zien is in figuur 14. Ook hebben deze programma's vaak voorontworpen basisvormen, om het productieproces te versnellen. Het kunnen zowel desktop applicaties als webapps zijn. Een voorbeeld van een webapp hiervoor is Tinkercad. Het doel van Tinkercad is het ontwerpen van 3d-afbeeldingen zo eenvoudig mogelijk te maken. Tinkercad is dan ook geen complexe, professionele tool. Toch valt hier bijna alles mee te maken.

2.4 Nabootsing menselijke bewegingen

Menselijke bewegingen vallen op meerdere manieren na te bootsen. Een keuze is om de duim alleen in en uit te laten bewegen. Echter is dit niet hoe het werkelijk zit, de duim kan als een soort bal rondbewegen op de handpalm. Er moet met meerdere dingen rekening worden gehouden als er een menselijke beweging wordt geïmiteerd.

Welke bewegingen

Allereerst moet er over de beweging nagedacht worden. Hoe ziet deze er natuurlijk uit, hoe gaan we deze modelleren en hoe is dat mogelijk. Hierbij moet er nagedacht worden over hoe de knie moet buigen, welke hoek de voet continu moet staan en wanneer de voet weer op de grond moet worden geplaatst. Dit is vooral van belang voor de balans. Ook moet de beweging er zo natuurlijk mogelijk uitzien.

De materialen

Er is ook iets nodig voor de beweging van de vingers. Ook heb je jumper wires⁵ nodig om alles met elkaar te verbinden. De microcontroller is nodig voor de aansturing en de motors voor de uiteindelijke beweging. Ook is er een sensor nodig die deze motoren aanstuurt. Alles in de arm heeft stroom nodig, er zijn dus ook batterijen nodig.

Het streven blijft uiteindelijk het ontwikkelen van een zo'n realistisch mogelijke prothese, zowel op esthetisch als mechanisch gebied.

Combineren

Als de microcontroller werkt en datgene dat de menselijke beweging moet nabootsen kan je met elkaar verbinden, zodat de pulsen vanuit de microcontroller kunnen worden omgezet in een menselijke beweging. Als dit ook gedaan is, is het tijd om de code die is hebt geschreven, uit te testen. Door je code via de microcontroller te laten lopen kan die er voor zorgen dat de menselijke beweging wordt uitgevoerd.

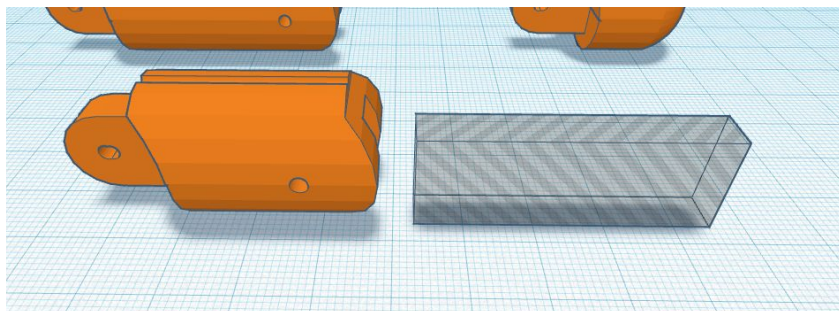
⁵ De draadjes die elektronische componenten met elkaar verbinden

3 Documentatie

In dit onderdeel documenteren wij de verschillende keuzes, fouten en vooruitgangen die wij hebben gemaakt. We beschrijven de ontwikkeling van onze robotische prothesehand zo goed mogelijk. Het eindproduct is ruwweg in drie verschillende onderdelen te verdelen: het ontwerpen en printen, het programmeren en de mechanische werking. Deze drie onderdelen overlappen elkaar continu. Voor de oriëntatie begonnen we met het kijken naar bestaande 3d-print handen en prothese handen. Hoe zijn deze ontworpen en/of hoe werken deze? Vervolgens is er online research gedaan naar verschillende software voor het ontwerpen van 3d modellen.

3.1 Ontwerpen en printen

De meest voor de hand liggende optie voor ons was Tinkercad⁶. Dit is een webapplicatie en hierdoor op elke computer met een internetverbinding te gebruiken. Het ontwerpen gaat door bepaalde vormen aan je document toe te voegen en die te vervormen. Bepaalde vormen kunnen ook in negatieve stand, waardoor die vorm uit de andere vorm gehaald kan worden.



figuur 15: Normale en negatieve vorm

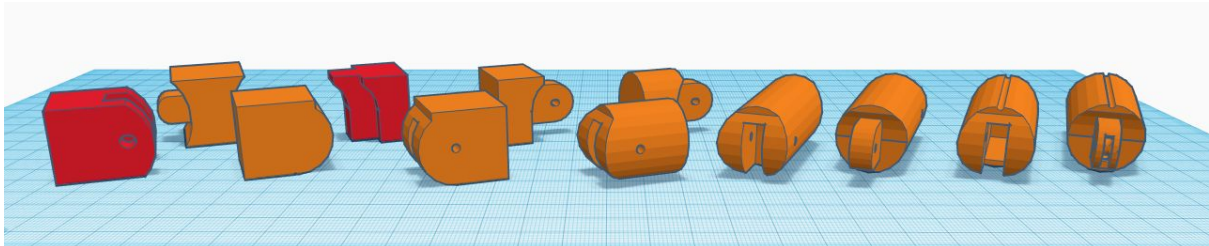
Voor het printen hebben wij de MakerBot Replicator (figuur 16) gebruikt. Dit is een desktop 3d printer met een applicatie voor de mac. Bij deze printer zijn er heel veel instellingen te veranderen, iets dat wij dan ook uitgebreid hebben gedaan. Hiervoor hadden wij een preset gemaakt waarin instellingen zoals de laagdikte, reissnelheid van de kop, printtemperatuur en dikte ondersteuningsstructuur ingesteld staan.



Figuur 16: De Makerbot Replicator

⁶ www.tinkercad.com

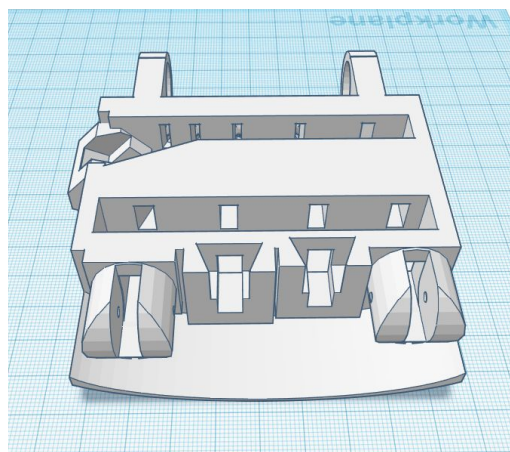
In de loop van de tijd zijn wij meerdere problemen tegengekomen in het ontwerp, deze hebben we telkens opgelost door zowel grote als kleine aanpassingen te maken. Zo had de eerste versie van de vinger nog geen ruimte in de binnenkant voor de mechaniek (het visdraad). Bij het ontwerp hebben we ook rekening gehouden met de verhouding van Fibonacci, de “gulden snede”. Door deze reeks te gebruiken hebben we geprobeerd een hand te maken die zoveel mogelijk overeenkomt met een normale mensenhand. Hier zie je het proces van een deel van de vinger:



Figuur 17: Proces van het middenkootje

Na het printen van een deel van de vingers of van de palm zagen wij steeds weer nieuwe leermomenten. Nadat wij deze dan fouten hadden gezien gingen wij bespreken hoe wij het dan beter hadden kunnen doen of welke printinstellingen wij moesten veranderen. Hierdoor ontwikkelde ons ontwerp steeds meer met de tijd en hebben wij uiteindelijk ons uiteindelijke product kunnen ontwerpen en printen.

Op de plek waar alle vingers bij elkaar komen, de handpalm, trad een nieuw probleem op. Je kan immers niet de binnenste twee vingers vastzetten, aangezien het koppelstuk van de buitenste twee vingers hierbij in de weg zat. Om deze 2 te kunnen plaatsen moesten wij een schuifmechanisme bedenken dat stevig genoeg zou zijn dat het niet los zou schieten, maar ook makkelijk te verwijderen is, aangezien je alleen op deze manier de vingers aan de knokkels vast kon maken met een pin.



Figuur 18: Palm van onderen met vertoning knokkelschuifjes

In figuur 18 kun je zien dat de vingers met pinnetjes aan de knokkels vast worden gemaakt en dat dat niet mogelijk zou zijn bij de ring- en middelvinger als deze niet verwijderd konden worden.

3.2 Elektronica

Het programmeren was erg afhankelijk van de hardware keuzes. Als microcontroller hebben we uiteindelijk gekozen voor de Arduino Nano. Toen wij begonnen met ons te verdiepen in het onderwerp, raakten we steeds meer bekend met de Raspberry Pi. Toch ontdekten we later dat dit niet de beste microcontroller is voor een prothesehand. Omdat dit een microcomputer is, en dus geen microcontroller, is het veel ingewikkelder om ons project ermee uit te werken. Ook is de Raspberry Pi vrij groot. Het voordeel aan de Raspberry Pi is wel dat je hiermee veelal Python scripts runt. Deze taal beheersten we alledrie al vrij goed, in tegenstelling tot C++, waar Arduino op gebaseerd is.

Er zijn veel verschillende soorten microcontrollers van Arduino. Wij hebben hierbij voor de Nano gekozen, aangezien deze erg klein is. Toch bezit deze controller genoeg pins⁷ om alles op aan te sluiten. Ook de Arduino programmeren gaat gepaard met groot gebruiksgemak: Je schrijft wat code op je computer, *upload* deze code naar de Arduino met de IDE en de Arduino blijft deze code uitvoeren.

Clone

Arduino is geheel open-source. Dit betekent dat alle broncode vrij op het internet beschikbaar is. Deze vrijheid zorgt ervoor dat er vele clones⁸ van de Arduino controllers bestaan. Dit zijn, veelal chinese, namaak Arduino's. Hierdoor zijn er Arduino's die worden verkocht voor ongeveer één vierde van de originele prijs. Er zijn echter niet alleen maar voordelen. De Nano die wij hadden gekocht had een OEM driver⁹ voor de CH34X chipset. Dit is een andere driver/chipset dan het origineel. Dit zorgde er helaas voor dat de Arduino crashte in combinatie met macOS. Hierdoor komt de Arduino niet tussen de beschikbare apparaten te staan. Dit is nodig voor het programmeren, meer hierover in §3.3. Na wat zoeken op het internet kwamen wij op een repository¹⁰ van Adrian Mihalko. Hij omschrijft het probleem als volgt: "*Version 1.3 (...) of the OEM driver for the CH34x chipset currently causes a kernel panic (a.k.a. crash) when installed on macOS Sierra.*" (Mihalko, 2017). Hij heeft dit opgelost door een bestandje, dat standaard zit inbegrepen bij macOS, te herschrijven. Hierdoor werd onze clone weer herkend door Mac computers.

Overige hardware

Natuurlijk heb je nog veel meer hardware nodig om een hand uiteindelijk te laten bewegen. Bijvoorbeeld de motortjes. Deze zijn grofweg in drie categorieën in te delen: DC-motoren, stappenmotoren en servomotoren.

Een DC¹¹ draait continu zo lang als er stroom op staat. Ze zijn tweedradig (power en ground) en draaien meestal met een hoge RPM¹².

Een stappenmotor werkt een stuk ingewikkelder, met meerdere elektromagneten die om een centraal tandwiel staan. De stappenmotor werkt in stapjes, en is dus erg precies te besturen. Het RPM is een stuk lager dan bij een DC-motor, maar de torque (draaikracht) is erg hoog. Een stappenmotor is echter wel vijfdradig.

Een servo is een snelle, precieze elektromotor. Meestal kan deze maar in 180 graden bewegen. Werkt ongeveer hetzelfde als een stappenmotor.

(ModMyPi, 2013)

⁷ Een bepaald soort aansluiting bij elektronica; kleine geleidende pinnetjes.

⁸ Precies nagemaakte Arduino controllers

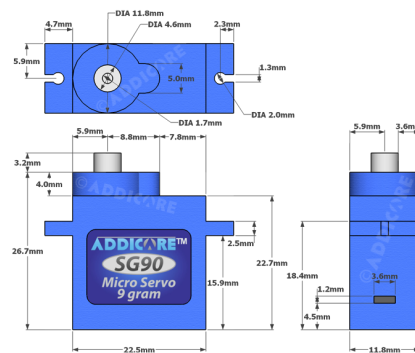
⁹ Een driver of stuurprogramma legt een verbinding tussen de hardware en het besturingssysteem.

¹⁰ Een pagina op Github.com met open-source code.

¹¹ DC = Direct Current.

¹² RPM = Revolutions Per Minute. Het aantal rondjes per minuut die de motor draait.

Wij hebben uiteindelijk gekozen voor de servomotor omdat deze het best bij een robotarm past. De motor beweegt snel, heeft maar een verbinding naar de Arduino en heeft genoeg torque. Echter is het allerbelangrijkste aan servomotoren dat ze feedback naar de Arduino kunnen sturen. We kunnen dus weten in welke positie hij staat. Dit is belangrijk bij het aansturen van de motor. Als deze bijvoorbeeld al volledig naar rechts is gedraaid, kunnen we niet nogmaals de opdracht geven om verder naar rechts te draaien. In ons geval zou de vinger dan door buigen. Wij zijn uiteindelijk gekomen op de SG90. Dit is een zeer kleine, maar krachtige servomotor.



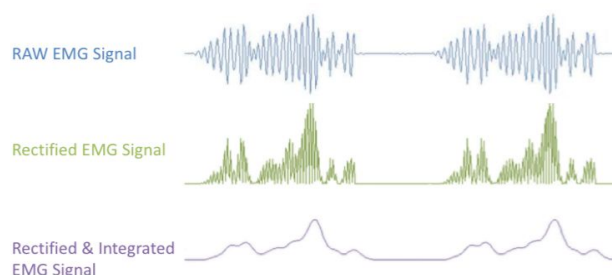
Figuur 17: De afmetingen van SG90 Mini Servo

Het aansturen van de hand

Zoals omschreven in de theorie, is het mogelijk om de hand aan te sturen door het kijken naar elektrische signalen in de armspieren. Dit wordt gedaan met behulp van een EMG sensor. Een voorbeeld van een EMG sensor is de MyoWare Muscle Sensor van Sparkfun. Deze kleine sensor is zeer eenvoudig in gebruik en gemaakt voor microcontrollers.

Bij het plaatsen van deze sensor is het belangrijk hoe deze geplaatst wordt. De sensoren moeten in het midden van de spier en in dezelfde richting als de vezels zitten.

De output van de sensor is echter geen ruw EMG signaal. Het is versterkt en de vele korte pieken zijn samengetrokken (Advancer Tech., 2015). Dit verschil is geïllustreerd in figuur 18, waar raw al versterkt is, rectified rechtgetrokken is en integrated de samengetrokken pieken zijn.



Figuur 18: Verschillende EMG signalen

Ook de bedrading en signaalgeving werkt erg goed bij de MyoWare. Er zijn maar drie draden voor nodig: V^+ ¹³, SIG ¹⁴ en GND ¹⁵.

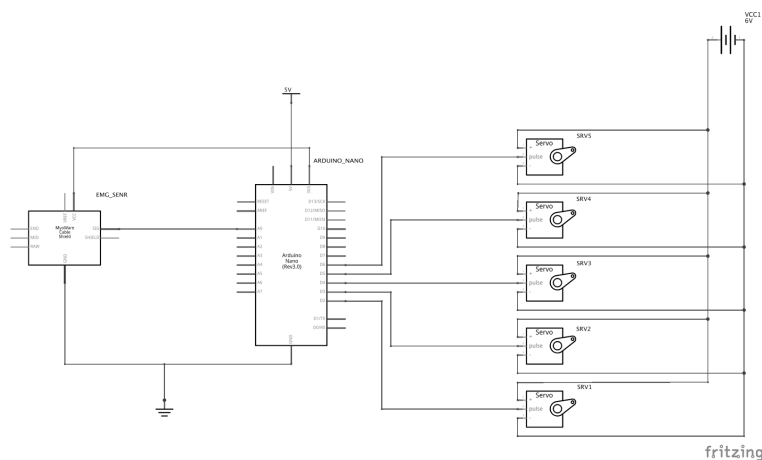
¹³ V^+ geeft de stroomtoevoer aan

¹⁴ SIG geeft de signaaldraad aan. Hierdoor loopt het signaal van de sensor naar de microcontroller

¹⁵ GND is de ground. Dit is de stroomafvoer.

Elektriciteitsschema

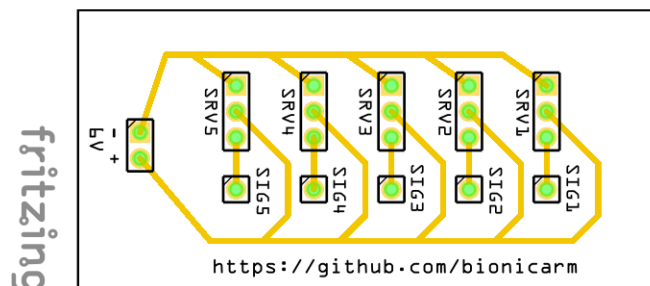
Voor je begint met het werken met circuits, is het belangrijk dat de stroomkringen altijd goed uitgedacht zijn. Als dit niet het geval is, kan er bijvoorbeeld kortsluiting optreden. Het is hierom dan ook belangrijk om een elektriciteitsschema van te voren te maken. Dit deden wij met Fritzing. Fritzing is een gebruiksvriendelijke applicatie voor het maken van zulke schema's. Fritzing ondersteunt drie verschillende soorten weergaves: Breadboard, Schematic en PCB. Breadboard is erg visueel. Je gebruikt hier dan ook getekende versies van echt bestaande onderdelen. Bij de Schematic view is dit heel anders. Schematic view geeft, zoals de naam al zegt, een schematische weergave de stroomkring. Breadboard wordt gebruikt voor het maken van prototypes, schematische weergave kan zowel bij prototypes als bij het uiteindelijke gesoldeerde product gebruikt worden. De stroomkring van onze prothesehand zag er dan als volgt uit:



Figuur 19: Schematische weergave van stroomkring.

Een grotere versie van dit elektriciteitsschema is te vinden in de bijlage.

In het schema is te zien dat de MyoWare sensor verbonden is aan de Arduino. Er komt stroom in vanaf de 3V3 pin van de Arduino. Vervolgens geeft de sensor signaal af aan Arduino pin A0. De Arduino en sensor hebben een gemeenschappelijke ground. Vijf digitale pins (D2-D6) geven signaal af aan de vijf servomotoren. Deze hebben een losse stroomkring, van 6 volt. Dit wordt geleverd door vier in serie geschakelde AA batterijen. De 5 servomotoren zijn aan de Arduino en bronspanning gekoppeld via een zelf ontworpen printplaat. Dit is eigenlijk een soort driver voor de servo's.



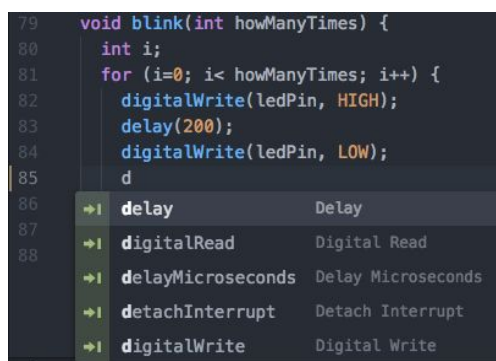
Figuur 20: Servo driver printplaat (ook op bijlage)

3.3 Programmeren

Nu we weten dat we gaan programmeren in de arduino-taal, de Nano volledig naar toebehoren werkt en de juiste hardware binnen is, kunnen we beginnen met programmeren. Hierin liepen we echter ook tegen belangrijke keuzes en vele leermomenten aan. In deze paragraaf wordt beschreven hoe dit proces te werk ging, en wat we daarvoor nodig hadden.

Ontwikkelingsomgeving

Wij hebben ervoor gekozen om de Arduino IDE niet te gebruiken in ons ontwikkelproces. De IDE van Arduino mist vele functionaliteiten die het ontwikkelen net iets makkelijker maken. Zo kun je bijvoorbeeld maar aan een bestand werken en mist het een terminal. Maar het allerbelangrijkste, er is geen autocompletion. Dit zorgt ervoor dat je maar een deel van de functie hoeft te typen, zoals te zien in figuur 21. Dit werkt sneller en zorgt voor een kleinere kans op kleine structuurfouten.



Figuur 21: Autocomplete-arduino package

Toch miste Atom een aantal belangrijke functies voor gebruik bij robotica projecten. Zo is er geen seriële monitor (voor de tekst output van de Arduino), geen compiler (de arduino-code omzetten in C++ code) en is het niet mogelijk direct te builden en uploaden vanuit Atom. Dit zou betekenen dat je eerst alle code in de Arduino IDE zou moeten plakken, om het vanuit daar te uploaden. Gelukkig is hier een package voor, genaamd PlatformIO. Dit voegt al deze functies, en nog veel meer, toe aan Atom. Hier is nog een kleine toevoeging voor nodig, voor het builden en uploaden naar de Arduino (Fatsi, 2015). Dit is mogelijk als de Arduino is aangesloten en de computer hem herkent. Bij dit proces wordt de code vanaf de computer naar de Arduino geladen.

Nu is Atom helemaal klaar voor ons project, met volledige functionaliteit.

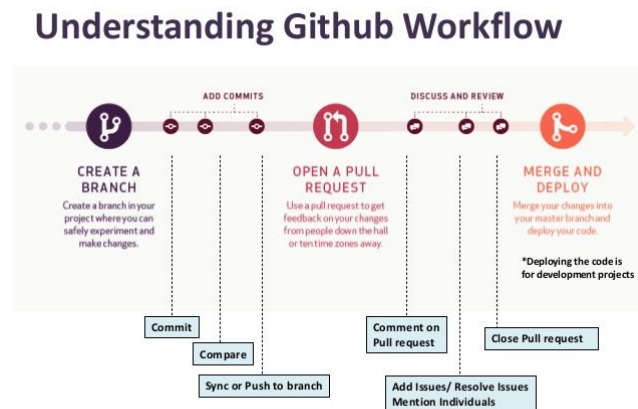
Om optimaal gebruik te maken van PlatformIO, moet je aan een bepaalde folder structuur voldoen. Zo moeten er bepaalde bestanden aanwezig zijn, en moeten bijvoorbeeld alle libraries¹⁶ in de map `lib/` staan.

Versiebeheersysteem

Wij hebben alle code met behulp van een version control system (VCS) geschreven. Hierdoor kan een van ons drieën een nieuwe functie of verbeterde code probleemloos toevoegen. Dit kan dan herzien worden, alvorens het vanuit een zijtak in de hoofdtak kan worden samengevoegd. Een completere beschrijving van dit proces is te zien in figuur 22. Dit zorgt ervoor dat er niet per ongeluk foutieve code geupload wordt, en de oudere versies dan overschreven worden. Dit werkt bijvoorbeeld zo bij Dropbox of een andere cloud opslag service. Door deze manier van werken is al onze broncode ook

¹⁶ Bibliotheken: een verzameling algemene functies/routines, waardoor je geen nieuwe code hoeft te schrijven voor iets dat erg algemeen is of vaak voorkomt.

volledig open-source, evenals alle andere bestanden die met dit project te maken hebben. Deze zijn te vinden op onze Github pagina, <https://github.com/bionicism/bionicism>.



Figuur 22: De Github workflow

Toelichting op code

Bij het schrijven van arduino-code maak je altijd gebruik van twee verplichte functies: **void setup() {}** en **void loop() {}**. Zoals de naam misschien al verhuut, gebruik je de setup functie om alles in op te zetten en de pins hun taak te geven. De loop functie is vervolgens het deel dat continu (in een ‘loop’) gaat lopen. Dit blijft net zolang doorgaan als dat er stroomtoevoer is. Dit ziet er als volgt uit:

```
void setup() {  
    // Hier plaats je alle setup-code  
}  
  
void loop() {  
    // En hier alles dat uitgevoerd moet worden  
}
```

Voor het gebruik van servomotoren heb je de Servo bibliotheek (library) nodig. Deze plaatsten we in `lib/Servo-master`, hierdoor konden we in het arduino-bestand er makkelijk naar verwijzen. Daarna moeten de motoren worden geïnitieerd met behulp van de library. Dit is hieronder te zien. De structuur bij het gebruiken van een library in Arduino is als volgt: “Library Variabelenaam;”. Hieraan wordt vervolgens in de `setup{}` functie een pinnetje aangekoppeld met `attach()`.

```
#include <Servo.h>  
  
Servo m1;  
Servo m2;  
Servo m3;  
Servo m4;  
Servo m5;  
// SETUP
```

```

void setup() {
    m1.attach(2);
    m2.attach(3);
    m3.attach(4);
    m4.attach(5);
    m5.attach(6);
}

```

Vervolgens kan de motor worden aangestuurd. Bij een servo motor doe je dit met `m1.write(n)`; waarin n een getal tussen de 0 en 180 is. Dit komt overeen met het aantal graden die de ring op de motor draait.

Vervolgens hadden we een aantal functies om bepaalde handposities aan te nemen. Zoals bijvoorbeeld een duim opsteken:

```

void thumbsUp() {
    m1.write(0);
    m2.write(180);
    m3.write(180);
    m4.write(180);
    m5.write(180);
}

```

In deze functie staat de duim helemaal uit en de vier andere vingers naar binnen.

De sensor moet ook worden ingesteld in de code. Dit deden wij als volgt:

```

void setup() {
    // Set EMG pin to input
    pinMode(A0, INPUT);
}

void loop() {
    // read value of analog pin 0 and give emg the value
    int emg = analogRead(A0);
}

```

Om nu de handeling met de sensorwaarde te koppelen, werd er gebruik gemaakt van een if-statement in de loop functie.

```
if (emg < 50) {  
    allFingers(700);  
} else if (emg > 50 && emg < 200) {  
    pinch();  
} else if (emg > 700) {  
    allFingers(2300);  
}
```

Hier bepaald de range van de waarde voor de emg variabele welke functie wordt uitgevoerd. In het geval dat het een bepaald getal is, wordt dat deel van de code uitgevoerd. Zo wordt er bij `emg > 700` de functie `allFingers(2300);` uitgevoerd.

De overige volledige broncode staat in de bijlage, aangevuld met kort commentaar.

4 Conclusie

Ons doel was om een goedkopere, makkelijker bewerkbare prothesehand te ontwikkelen, die mechanisch naar toebehoren werkt. Door de hand te printen met een 3D-printer, is deze prothesehand vele malen goedkoper dan prothese handen op de markt van nu. Ook werkt de hand volledig op aansturing van armspieren.

Voor deze prothesehand zijn de volgende materialen nodig:

Product	Aantal	Prijs per stuk (€)	Totale prijs (€)
Arduino Nano De microcontroller	1	4,60	4,60
Servo Driver Hier wordt de stroom en het signaal van de servo's verdeeld	1	12,66	12,66
SG90 Mini Servo De motors die de vingers laten bewegen	5	3,75	18,75
Visdraad Zitten aan de motortjes en de vinger vast zodat de vingers bewegen	1	6,95	6,95
Batterijhouder voor 4x AA Zorgt dat de AA batterijen op hun plek blijven	1	12,00	12,00
AA batterijen Zorgen voor stroom	4	6,35 (per 4)	6,35
MyoWare Muscle Sensor Spier sensor voor aansturing	1	50,95	50,95
Elektrodes Voor het opvangen van elektrische signalen	3	1,16	3,50
Jumper wires Zorgen voor stroomovergang tussen verschillende onderdelen in de arm	1	ong 3,50	3,50
3D-printer filament Het plastic waarmee geprint wordt	1 rol	22,50	22,50
Totaalprijs prothesehand	-	-	136,13

De prothesehand heeft in totaal 136,13 euro gekost.

Deze prijs is erg variabel. Zo kunnen er duurdere motoren worden gebruikt, zodat er meer kracht beschikbaar is. De keuze en hoeveelheid filament is ook zeer belangrijk voor de prijs en kan erg variëren.

5 Reflectie

Wij denken dat dit profielwerkstuk in vele opzichten erg goed heeft bijgedragen aan ons leren. Niet alleen met het schrijven van verslagen, maar ook over elektronica, design, 3d-printers en projectmatig werken. De taakverdeling was erg duidelijk en de communicatie verliep goed.

Aan het begin van het project hadden we bepaalde ideeën over de prothese die we wilden maken. Wij hebben er alles aan gedaan om dit voor elkaar te krijgen. We kunnen de arm aansturen met onze spieren.

Voor een vervolg zou de prothesehand complexer kunnen zijn. Denk hierbij aan meerdere draairichtingen, verbeterde aansturing en versnelde reactietijd. Bij het produceren van een kunstmatige versie van een van de ingewikkeldste onderdelen van het lichaam, valt er altijd te verbeteren.

Referenties

Onze open-source bestanden zijn te vinden op Github:

<https://github.com/bionicarm/bionicarm>

<https://github.com/bionicarm/3dprints>

Advancer Technologies. (2015). *MyoWare Muscle Sensor Datasheet*. Binnengehaald 9 januari 2018 van <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf>

Arduino Foundation. (2017). *Introduction*. Binnengehaald 20 september 2017 van <https://www.arduino.cc/en/Guide/Introduction>

Alpenfels, E. J. (1955). The anthropology and social significance of the human hand. *Artificial limbs*, 2, 2-21

Anatomie van de hand. (z.j.). Binnengehaald 4 september 2017 van <https://www.hierhebigpijn.nl/anatomie/hand>

Armprothese. In *Wikipedia*. Binnengehaald 12 september 2017 van <https://nl.wikipedia.org/wiki/Armprothese>

Chilson, L. (26 januari 2013). The Difference Between ABS and PLA for 3D Printing. Binnengehaald 18 september 2017 van <https://www.protoparadigm.com/news-updates/the-difference-between-abs-and-pla-for-3d-printing/>

Danhof, N. (z.j.). Borstprothese. Binnengehaald 11 september 2017 van <http://www.rosesbodycare.nl/borstprothese.htm>

Di Justo, P. (4 december 2015). *Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board*. Binnengehaald 20 september 2017 van <https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/>

Duann. (7 november 2012). *Design for 3D Printing 101: Intro to Design for 3D Printing*. Binnengehaald 20 september 2017 van <https://www.shapeways.com/blog/archives/1678-design-for-3d-printing-101-intro-to-design-for-3d-printing.html>

Fatsi, E. (20 juli 2015). *Arduino Development in Atom Editor*. Binnengehaald 13 juni 2017 van <https://www.viget.com/articles/arduino-development-in-atom-editor>

Feenstra, M., & Woods, C. (z.j.). *Van amputatie tot prothese*.
https://richtlijnendatabase.nl/gerelateerde_documenten/f/2876/Patientenversie%20Amputatie%20tot%20Prothese.pdf

Gulden snede. In *Wikipedia*. Binnengehaald 7 september 2017 van https://nl.wikipedia.org/w/index.php?title=Gulden_snede&oldid=48684258

Henry, A. (20 april 2014). *Five best text editors*. Binnengehaald 20 september 2017 van <https://lifelhack.com/five-best-text-editors-1564907215>

Implantaat. In *Wikipedia*. Binnengehaald 11 september 2017 van <https://nl.wikipedia.org/wiki/Implantaat>

Kravets, I. (2017). *PlatformIO IDE for Atom*. Binnengehaald 20 september 2017 van <https://github.com/platformio/platformio-atom-ide>

Mihalko, A. (2017). *ch340g-ch34g-ch34x-mac-os-x-driver*. Binnengehaald 6 oktober 2017 van <https://github.com/adrianmihalko/ch340g-ch34g-ch34x-mac-os-x-driver/blob/master/README.md>

ModMyPi. (2013). What's the difference between DC, Servo & Stepper motors?.
Binnengehaald 29 december 2017 van <https://www.modmypi.com/blog/whats-the-difference-between-dc-servo-stepper-motors>

Mortelé, H., Dr. (2009). Anatomie van de hand. Binnengehaald 11 september 2017 van <http://www.mortele.info/patienteninfo/hand/anatomie.asp>

Murali, S. (2012). Golden ratio in human anatomy. (Unpublished master's thesis).
Government College Chittur. doi:10.13140/2.1.2265.9526

Natuurinformatie - Hand. (z.j.). Binnengehaald 4 september 2017 van <http://www.natuurinformatie.nl/nnm.dossiers/natuurdatabase.nl/i003843.html>

Oosterbos, K., Koot, H. Anatomie van pols en hand. (z.j.). Binnengehaald 4 september 2017 van <http://www.orthopedie-grooteindhoven.nl/anatomie-van-pols-en-hand>

Osborne, H. (1999). The Oxford companion to art (First ed.). Oxford: The Clarendon Press.

Prothese. In *Wikipedia*. Binnengehaald 11 september 2017 van <https://nl.wikipedia.org/wiki/Prothese>

Raspbian. (z.j.). *Welcome to Raspbian*. Binnengehaald 20 september 2017 van <https://www.raspbian.org>

Richard, F. ff. Weir. (2004). Design of artificial arms and hands for prosthetic applications. In Kutz, M. (red.) Standard handbook of biomedical engineering and design. New York: McGraw-Hill.

Taylor, T. (1999-2017). Muscles of the arm and hand. Binnengehaald 8 januari 2018 van <http://www.innerbody.com/anatomy/muscular/arm-hand>

Unix. In *Wikipedia*. Binnengehaald 25 september 2017 van <https://nl.wikipedia.org/w/index.php?title=Unix&oldid=49691124>

Vervloesem, K. & Overman, M. (2017). *Arduino voor iedereen* (5e druk). Haarlem: Reshift Digital.

Bronvermelding bij figuren

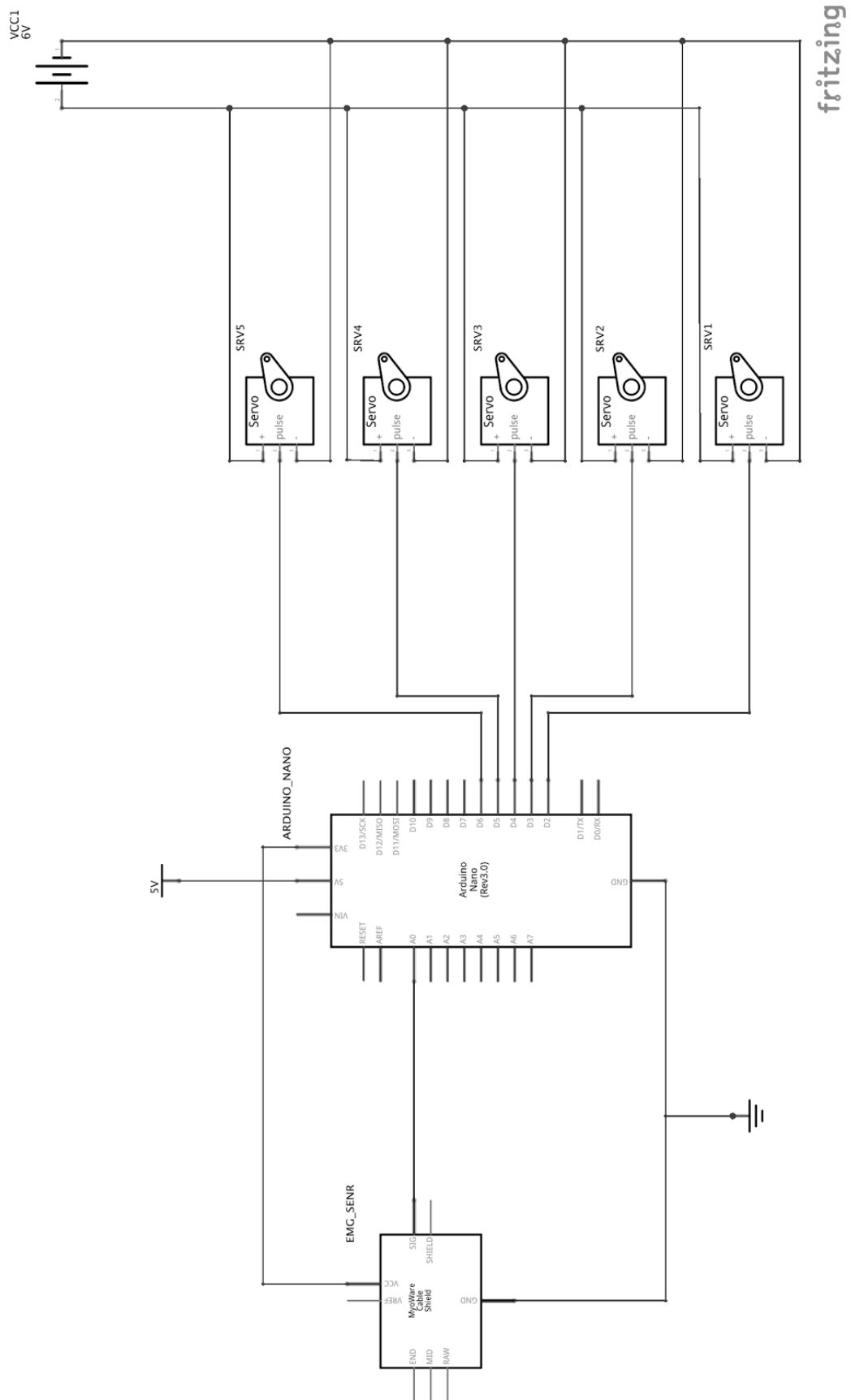
In volgorde van voorkomen.

1. Livit. (2016). *Beenprothese*. [tekening]. Binnengehaald van
<https://www.livit.nl/wp-content/uploads/2016/09/Onderbeenprothese-2-239x300.png>
2. Hess, A. (2010). *Figure 8 Harness*. [schets]. Binnengehaald van
http://www.upperlimbprosthetics.info/index.php?p=1_9_Body-Powered
3. Guardian Graphic. (2017). *How neuro-prosthesis reconnects brain and muscles*. [frame].
Binnengehaald van
<https://www.theguardian.com/science/2017/mar/28/neuroprosthetic-tetraplegic-man-control-hand-with-thought-bill-kochevar>
4. Golem. (z.j.). *Ilimbpulse*. [foto]. Binnengehaald van
https://scr3.golem.de/screenshots/1005/ilimbpulse/ilimbpulse_3.jpg
- 5.1. Orthopedie Groot-Eindhoven. (z.j.). *Anatomie van de hand*. [tekening].
Binnengehaald van
http://www.orthopedie-grooteindhoven.nl/images/pages/s1/p1319/9495_1.png?1025
- 5.2. stihii. (z.j.). *Anatomy of muscle system hand*. [tekening]. Binnengehaald van
https://st.depositphotos.com/1752931/1415/i/950/depositphotos_14152005-stock-photo-anatomy-of-muscular-system-hand.jpg
- 5.3. School of medicine Washington. (z.j.). *Sensation hand dorsal*. [foto, bewerkt].
Binnengehaald van
[https://nervesurgery.wustl.edu/NerveImages/MISC%20-%20EM%20\(Images%20Require%20Captions\)/Sensation---Hand-Dorsal---IMG_6132.jpg](https://nervesurgery.wustl.edu/NerveImages/MISC%20-%20EM%20(Images%20Require%20Captions)/Sensation---Hand-Dorsal---IMG_6132.jpg)
6. Oliver, P C. (2005). *Secció àuria*. [vector]. Binnengehaald van
https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/Secció_àuria_-_Golden_section.png/350px-Secció_àuria_-_Golden_section.png

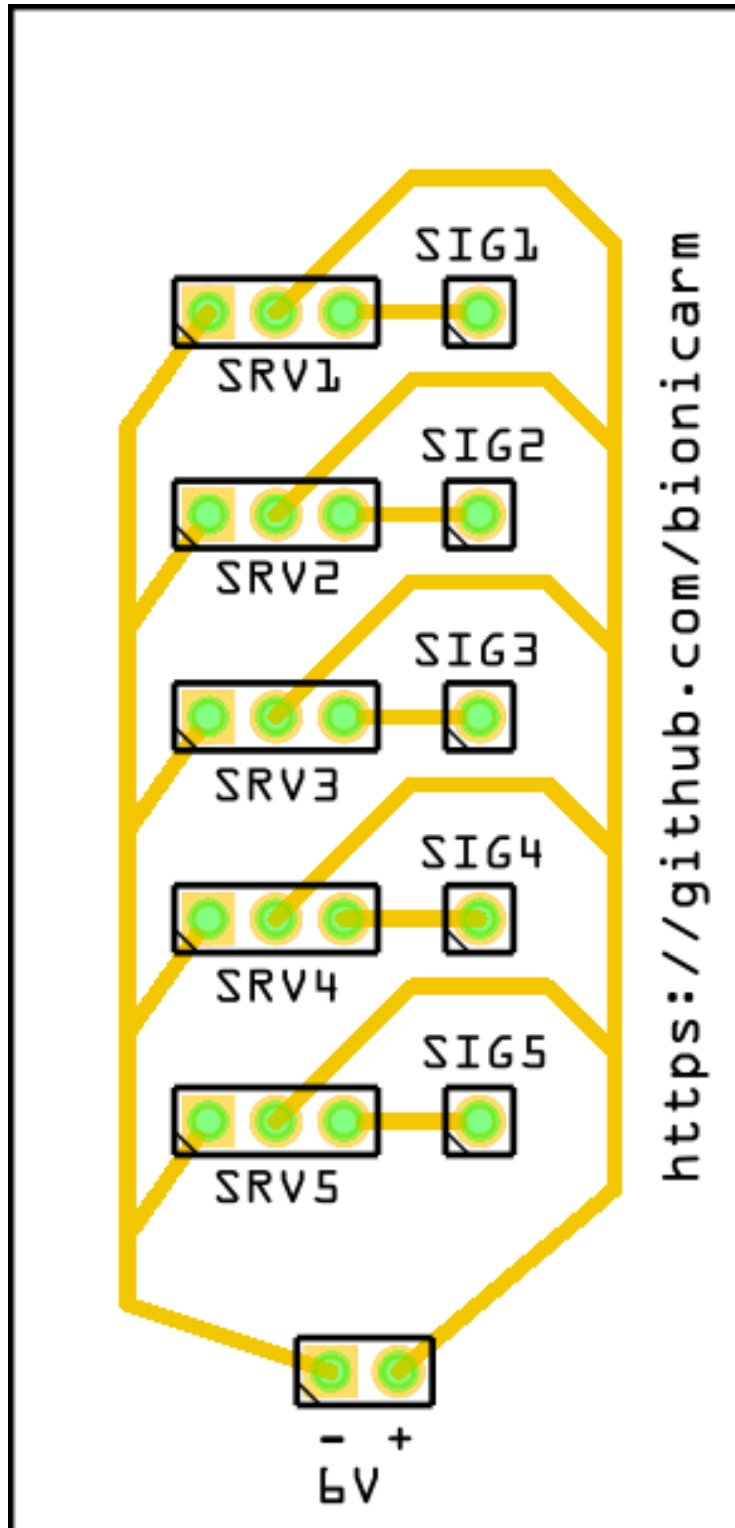
- 7.1. Eddie. (2010). *Golden ratio*. [tekening]. Binnengehaald van
<http://worldtruth.tv/wp-content/uploads/2012/01/goldenratio.png>
- 7.2. Pettit, R. (2016). *The fist of Fibonacci*. [tekening]. Binnengehaald van
<https://metadadajournal.files.wordpress.com/2016/02/image01-fist-of-fibonacci.jpg>
8. Olson, M. (2016). *PLA vs. ABS*. [foto]. Binnengehaald van
http://static.wixstatic.com/media/07075b_feeb742b74a84e7ca0596cf4396bae4e~mv2.png
9. core-electronics. (z.j.) *Nano V3.0 Board (Arduino Nano V3.0-Compatible)*.
Binnengehaald van
<https://core-electronics.com.au/nano-v3-0-board-arduino-nano-v3-0-compatible.html>
10. Streicher, M. (2008). *Opening Windows with Cygwin*. [schermabeelding].
Binnengehaald van
https://www.ibm.com/developerworks/aix/library/au-spunix_cygwin/launch.jpg
11. Long, S. (2014). *Changes to the Raspbian user interface*. [schermabeelding].
Binnengehaald van <https://www.raspberrypi.org/app/uploads/2014/12/addtod.jpg>
12. Eigen schermabeelding.
13. PlatformIO. (2017). *PlatformIO IDE for Atom*. [schermabeelding]. Binnengehaald
van http://docs.platformio.org/page/_images/ide-atom-platformio.png
14. Duann. (2012). *Autodesk's 123D App*. [schermabeelding]. Binnengehaald van
<https://www.shapeways.com/blog/wp-content/uploads/2012/11/autodesk-123d-36.jpg>
15. Eigen schermabeelding
16. Makerbot. (2016). *replicator_1*. [foto]. Binnengehaald van
http://3dprinterusa.com/wp-content/uploads/2016/02/replicator_1.jpg

17. Addicore. (z.j.). *Addicore SG90 Servo Dimensions Mechanical Drawing*. [tekening].
Binnengehaald van <https://www.addicore.com/Addicore-SG90-Mini-Servo-p/113.htm>
18. Advancer Technologies. (2015). *3-lead Muscle / Electromyography Sensor for Microcontroller Applications*. [grafiek]. Binnengehaald van
<https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf>
19. Eigen elektriciteitsschema.
20. Eigen printplaat ontwerp.
21. Eigen schermafbeelding.
22. Sukumar, R. (2015). *Understanding Github Workflow*. [tekening]. Binnengehaald van
<https://image.slidesharecdn.com/introductiontogithubslideshare-150913080822-lva1-app6892/95/introduction-to-github-slideshare-7-638.jpg?cb=1442131732>
23. Eigen schermafbeelding
24. Eigen schermafbeelding

Bijlage



Bijlage 1: Elektricitesschema



fritzing

Bijlage 2: De Servo Driver


```

// LIBRARIES //
#include <Servo.h>

// setting up servo's.
// m1 = finger 1 (thumb) etc.
Servo m1;
Servo m2;
Servo m3;
Servo m4;
Servo m5;

// SETUP //
void setup() {
  // Attach servo variables to digital pin 2-6
  m1.attach(2);
  m2.attach(3);
  m3.attach(4);
  m4.attach(5);
  m5.attach(6);

  // Serial monitor for debugging
  Serial.begin(9600);
}

// LOOP //
void loop() {
  // read the analog value of A4 and store it in variable 'emg'
  int emg = analogRead(A4);

  // if statement for defining which muscle activity corresponds
  // with which hand movement
  if (emg < 50) {
    allFingers(700);
  } else if (emg > 50 && emg < 200) {
    pinch();
  } else if (emg > 700) {
    allFingers(2300);
  }

  // delay to give the fingers time to move
  delay(1000);
}

```

```

// PRESETS //
void allFingers(int pos) {
    // get all fingers in same position.
    // With our servo's; 700 for open hand,
    // 2300 for fully closed hand
    m1.writeMicroseconds(pos);
    m2.writeMicroseconds(pos);
    m3.writeMicroseconds(pos);
    m4.writeMicroseconds(pos);
    m5.writeMicroseconds(pos);
    delay(100);
}

void pinch() {
    m1.writeMicroseconds(2300);
    m2.writeMicroseconds(2300);
    m3.writeMicroseconds(700);
    m4.writeMicroseconds(700);
    m5.writeMicroseconds(700);
    delay(100);
}

void thumbsUp() {
    m1.writeMicroseconds(700);
    m2.writeMicroseconds(2300);
    m3.writeMicroseconds(2300);
    m4.writeMicroseconds(2300);
    m5.writeMicroseconds(2300);
    delay(100);
}

void middleFinger() {
    m1.writeMicroseconds(2300);
    m2.writeMicroseconds(2300);
    m3.writeMicroseconds(700);
    m4.writeMicroseconds(2300);
    m5.writeMicroseconds(2300);
    delay(100);
}

```

Bijlage 3: Volledige broncode