

Universiteit Twente

Bacheloropdracht

Ontwerp en Implementatie van
een Meetsysteem voor
Tractiekrachtregistratie van
Veer-gebaseerde Halo-Gravity
Traction

Auteur

L.J. (Luuk) Baten
S2626810

Begeleiders:

ir. E.E.G. Hekman
Prof.dr.ir. Gabriëlle Tuijthof
prof.dr. Moyo Kruyt

FACULTEIT ENGINEERING TECHNOLOGY
DEPARTMENT BIOMECHANICAL ENGINEERING (BE)

11 februari 2024

Abstract

Abstract Nederlands

Bij mensen met een ernstige misvorming van de wervelkolom zoals bij scoliose, kyfose of lordose kan de wervelkolom pre-operatief deels worden gecorrigeerd door middel van Halo-Gravity Traction. Dit is een methode waarbij de wervelkolom door middel van een gewicht of veer in de goede stand wordt getrokken. Voor onderzoek naar de effectiviteit van verschillende veer-gebaseerde Halo-Gravity Traction strategieën is een meetsysteem nodig welke in deze bacheloropdracht is ontworpen, gerealiseerd en geïmplementeerd.

Het definitieve ontwerp bestaat uit een Arduino-gebaseerd modulair meetsysteem in een 3D geprint omhulsel. De gebruikte hardware componenten zijn: een powerbank, loadcell met meetversterker, RTC module en een micro SD kaart module. Het systeem is te bedienen door middel van knoppen met feedback door LED's.

Het ontwerp is geanalyseerd met een risicoanalyse en de opgestelde functies en eisen zijn geëvalueerd. Deze hebben uitgewezen dat het systeem geen significante risico's vormt voor de gebruiker, maar dat niet aan alle opgestelde eisen is voldaan. Wel voldoet het systeem aan alle opgestelde functionaliteiten.

De conclusie is daarom dat het waarschijnlijk lijkt dat het systeem in de toekomst effectief kan worden ingezet voor onderzoek naar veer-gebaseerd Halo-Gravity Traction.

Vervolgonderzoek kan focussen op het samenstellen van optimalere hardware componenten en zal uit moeten wijzen of aan alle eisen voldaan kan worden.

Abstract English

In people with a serious deformity of the spine such as scoliosis, kyphosis or lordosis, the spine can be partly corrected pre-operatively using Halo-Gravity Traction. This is a method in which the spine is pulled into the correct position by means of a weight or spring. Research into the effectiveness of various spring-based Halo-Gravity Traction strategies requires a measurement system that was designed, realized and implemented in this bachelor's assignment.

The final design consists of an Arduino-based modular measurement system in a 3D printed enclosure. The hardware components used are: a power bank, load cell with measuring amplifier, RTC module and a micro SD card module. The system can be operated using buttons with feedback through LEDs.

The design has been analyzed with a risk analysis and the established functions and requirements have been evaluated. These have shown that the system does not pose any significant risks to the user, but that not all requirements have been met. The system does meet all established functionalities.

Therefore, the conclusion is that it appears likely that the system can be effectively used for future research on spring-based Halo-Gravity Traction.

Further research could focus on assembling more optimal hardware components and determining if all requirements can be met.

Inhoudsopgave

Abstract	1
Abstract Nederlands	1
Abstract English	1
1 Inleiding	3
2 Theorie	4
2.1 Misvormingen van de wervelkolom	4
2.2 Halo-Gravity Traction	4
3 Analyse	6
3.1 Probleemstelling	6
3.2 Ontwerpdoel	6
3.3 Functies en eisen	6
3.3.1 Functies	6
3.3.2 Eisen	6
4 Concept	8
4.1 Hardware	8
4.1.1 Besturingssysteem	8
4.1.2 Stroomvoorziening	9
4.1.3 Kracht meten	9
4.1.4 Datum en tijd bijhouden	10
4.1.5 Data verwerken	11
4.1.6 Bediening	11
4.2 Software	12
4.2.1 Algemene code	12
4.2.2 Databeheer	13
4.2.3 Gebruikersfouten	13
4.3 Integratie in huidige HGT systeem	14
5 Realisatie	16
5.1 Ontwerp 1	16
5.2 Ontwerp 2	16
5.3 Ontwerp 3	17
5.4 Risicoanalyse	18
5.5 Evaluatie	19
6 Discussie	20
7 Conclusie	22
8 Referenties	23
Bijlage A. Aansluiting hardware	25
Bijlage B. Faalmodus en effectanalyse	26
Bijlage E. Schema code	28
Bijlage D. Code	29

1 Inleiding

Voor mensen met een misvormde wervelkolom, zoals in het geval van *scoliose*, *kyfose* of *lordose* kan orthopedische chirurgie een uitkomst bieden. Hevige verkrommingen van de wervelkolom zijn echter nog altijd een uitdaging voor orthopedisch chirurgen. Snelle correctie van ernstige scoliose vergroot namelijk het risico op neurologische problemen. [1] Er is gerapporteerd dat *Halo-Gravity Traction (HGT)* hierbij succesvol kan assisteren door het verminderen van de vervormingen zonder significante complicaties. [2]

Over het algemeen wordt HGT uitgevoerd door middel van het aanbrengen van een tractiekracht met gewichten. Deze methode, hoewel effectief gebleken, brengt echter een variëteit aan veiligheidsrisico's en andere nadelen met zich mee. Hierom zijn in 2020 door Yu et al. en in 2021 door B.A.M. Bruin HGT methoden gerealiseerd, waarbij gebruik wordt gemaakt van een veer om een tractiekracht aan te brengen. Deze laatste wordt momenteel gebruikt in het Universitair Medisch Centrum Utrecht (UMC) en is te zien in figuur 1. [3, 4]

Volgens orthopedisch chirurg M. Kruyt van het UMC verschilt de draagtijd van het HGT systeem momenteel tussen patiënten. Dit komt onder andere doordat patiënten de mogelijkheid hebben zichzelf los te koppelen van het systeem wanneer de tractiekracht te oncomfortabel wordt en wanneer ze de rolstoel niet kunnen gebruiken.

Er is gebleken dat de eerste twee weken van de tractie het meest effectief zijn. [1], maar de optimale draagtijd en patroon is nog onbekend. Daarnaast is het effect van beweging van de patiënt op de tractiekracht ook nog niet geheel duidelijk.

Hierom is het belangrijk dat er onderzoek gedaan wordt naar de constantheid van de tractiekracht en de optimale draagtijd.

Op verzoek van M. Kruyt zal in deze bacheloropdracht daarom een systeem ontworpen, gerealiseerd en geïmplementeerd worden voor het meten en opslaan van de tractiekracht van het huidige veer-gebaseerde HGT systeem.

In hoofdstuk 2 wordt de theorie met betrekking tot de wervelkolom en Halo-Gravity Traction beschreven. In hoofdstuk 3 worden de probleemstelling, ontwerpdoel en de functies en eisen uiteengezet. Hoofdstuk 4 biedt een overzicht van de gebruikte hardware en software van het meetsysteem en de integratie hiervan in het HGT systeem. Vervolgens beschrijft hoofdstuk 5 de op basis hiervan gerealiseerde ontwerpen en wordt hier een risicoanalyse en evaluatie van het definitieve ontwerp gepresenteerd. De discussie en conclusie worden gepresenteerd in opeenvolgend hoofdstuk 6 en 7.



Figuur 1: Het huidige HGT systeem van B.A.M. Bruin. [3]

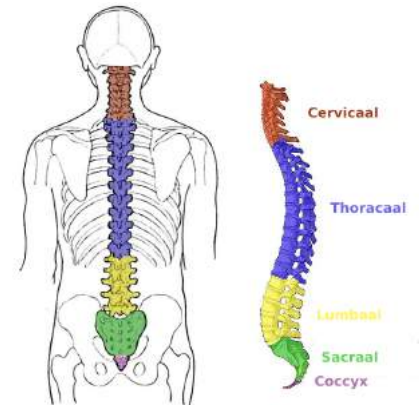
2 Theorie

2.1 Misvormingen van de wervelkolom

Bij de menselijke wervelkolom wordt onderscheid gemaakt tussen vijf delen: zie figuur 2. Vanaf *craniale* zijde: zeven *cervicale* wervels, twaalf *thoracale* wervels, vijf *lumbale* wervels, vijf gefuseerde *sacrale* wervels en de *coccyx*. De stijfheid en vorm van de wervelkolom bieden structuur aan de romp en het hoofd. [6]

Zoals te zien is in figuur 2 is in de wervelkolom van een gezond persoon een natuurlijke kromming aanwezig in het saggitale vlak. Deze S-vorm werkt schokabsorberend en zorgt ervoor dat het hoofd boven de pelvis is gepositioneerd. [7] Met *kyfose* wordt de natuurlijke voorwaartse kromming van bijvoorbeeld de thoracale wervels bedoeld en *lordose* is de natuurlijke achterwaartse kromming van bijvoorbeeld de cervicale en lumbale wervels.

De normale hoeveelheid thoracale kyfose is 20° - 45° en de normale hoeveelheid lumbale lordose is 25° - 60° . In het frontale vlak is de wervelkolom van een gezond persoon recht. Een kromming tot en met 10° wordt hierbij echter nog beschouwd als normaal. [8, 9]



Figuur 2: Anatomie van de wervelkolom. [5]

Wanneer de kromming van de wervelkolom buiten dit bereik valt is er sprake van aandoeningen zoals *hypo/hyper-kyfose*, *hypo/hyper-lordose*, *scoliose* of een combinatie hiervan.

Elk van deze aandoeningen kan veroorzaakt worden door een variëteit aan redenen waaronder geboorteafwijkingen, veroudering, degeneratie, tumoren en trauma. [10]

De gevolgen van een abnormale stand van de wervelkolom omvatten pijn in de rug en benen, neurale symptomen waaronder spierzwakte en gevoelloosheid van de benen en functionele beperkingen waaronder moeite met rechtop staan en/of beweging. Ook kunnen significantere klachten ontstaan zoals cardiale en gastro-intestinale complicaties en een verminderde ademhalingsfunctie. [11, 12]

Huidige behandelingen kunnen opgedeeld worden in twee groepen: chirurgische en niet-chirurgische behandelingen. Chirurgische ingrepen worden vaak bij ergere gevallen gebruikt en blijken veelal effectief. Niet chirurgische ingrepen zijn over het algemeen minder effectief, maar ook minder belastend voor de patiënt. Dit zijn onder andere fysiotherapie, medicijninjectie en brace-orthopedie. [7, 8, 13]

2.2 Halo-Gravity Traction

Vervormingen van de wervelkolom kunnen ook met een combinatie van behandelingen verminderd worden. Een voorbeeld hiervan is de combinatie van HGT met een orthopedische ingreep. Bij HGT wordt een halo ring met pinnen vastgezet aan de schedel van de patiënt waarna deze omhoog wordt getrokken. Gedurende een aantal dagen wordt deze kracht daarna geleidelijk verhoogd tot maximaal 50% van het lichaamsgewicht van de patiënt. Dit zorgt ervoor dat de abnormale kromming in de wervelkolom al gedeeltelijk wordt verminderd, waardoor in de operatie die volgt een kleinere hoek overbrugd hoeft te worden. [2]

Door de grote hoek die normaliter bij een orthopedische ingreep verbeterd moet worden is vaak een hoge mate van osteomie vereist. Een procedure waarbij een deel van de wervelkolom wordt verwijderd om de houding hiervan te verbeteren. Dit verhoogt de kans op neurologische schade. Ook beschikken patiënten vaak niet over voldoende longcapaciteit, waardoor bij een operatie cardio-respiratoire stoornissen kunnen ontstaan.

Doordat HGT ervoor zorgt dat de longfunctie pre-operatief wordt verbeterd, wordt de kans op post-operatieve complicaties verkleint. [2, 14]

Er zijn HGT systemen waarbij de patiënt in bed ligt, maar ook mobiele versies in de vorm van een looprek of rolstoel. Laatst genoemde wordt momenteel in het UMC gebruikt voor het uitvoeren van HGT, zie figuur 3. [15]

De duur van de HGT behandeling ligt meestal rond enkele weken, maar kan in sommige gevallen

oplopen tot meerdere maanden. Tijdens de behandeling hebben patiënten de mogelijkheid zichzelf los te koppelen van het systeem, wanneer de tractie te onaangenaam wordt of op momenten wanneer het HGT systeem niet gedragen kan worden. De patiënten zijn dus niet constant in tractie. [2, 14]



Figuur 3: Een patiënt in het UMC Utrecht gekoppeld aan een HGT systeem.

De originele vorm van de HGT behandeling is met een systeem van een gewicht en katrollen, maar er zijn inmiddels ook systemen ontwikkeld die gebruik maken van een veersysteem. De constantheid van de tractiekracht en effectiviteit van deze vorm van HGT zijn nog niet onderzocht. [3, 4] Daarnaast is er momenteel nog geen consensus over een meest effectieve strategie, draagtijd en draagpatroon voor deze HGT behandelmethode. [2] Voor onderzoek hiernaar is een meetsysteem nodig welke in deze bacheloropdracht zal worden ontworpen.

3 Analyse

3.1 Probleemstelling

In het UMC is er tot op heden een gebrek aan meetapparatuur om onderzoek te doen naar de effectiviteit van verschillende veer-gebaseerde HGT strategieën.

3.2 Ontwerpdoel

Het doel van deze bacheloropdracht is het ontwerpen, realiseren en implementeren van een meet-systeem voor het registreren van de tractiekracht van het huidige HGT systeem.

3.3 Functies en eisen

Om duidelijkheid te creëren over de functionaliteit en eigenschappen van het meetsysteem zullen in dit hoofdstuk de benodigde functies en eisen behandeld worden. Deze zijn opgesteld aan de hand van de parameters die gemeten moeten worden en de opdracht beschrijving vanuit M. Kruyt.

3.3.1 Functies

- [F01] Kracht van HGT meten
- [F02] Kracht van HGT uitlezen
- [F03] Datum en tijd bijhouden
- [F04] Kracht, datum en tijd opslaan
- [F05] Energie leveren
- [F06] Energiebron verwisselen
- [F07] Meting doen
 - [F07.1] Meting starten
 - [F07.2] Meting resetten
- [F08] Data verwerken
 - [F08.1] Data naar computer overzetten
 - [F08.2] Data weergeven

3.3.2 Eisen

Functionele vereisten:

- [E01] Het meetsysteem moet spatwaterdicht zijn.
- [E02] Het product mag maximaal 2 kg wegen.
- [E03] Het meetsysteem moet elke 10 seconden de huidige tractiekracht kunnen meten. [0.10 kg nauwkeurigheid]
- [E04] Het meetsysteem moet elke 10 seconden de huidige tractiekracht kunnen opslaan.
- [E05] Het meetsysteem moet elke 10 seconden de huidige datum en tijd kunnen opslaan. [1.0 seconde nauwkeurigheid]
- [E06] Het systeem moet minimaal 1 week aan data kunnen opslaan. (datum, tijd en kracht)
- [E07] Het product moet een energiebron bevatten waarmee het systeem minimaal 1 week kan blijven meten.
- [E08] Het product moet een tractiekrachten tot en met 45 kg kunnen meten.
- [E09] Moet minimaal 20 behandelingen kunnen meegaan.
- [E10] Het meetsysteem moet betrouwbare metingen kunnen geven bij temperaturen van -10°C tot en met $+40^{\circ}\text{C}$.

[E11] Het product moet een veiligheidsfactor hebben van ≥ 2.2

Gebruikers vereisten:

[E12] Het product mag de werking van de HGT niet beperken.

[E13] Het product mag de bewegingsruimte van de patiënt niet beperken.

[E14] De energiebron moet op een eenvoudige wijze verwisselbaar zijn.

[E15] De gemeten data moet eenvoudig naar een computer over te kunnen zetten zijn.

[E16] De gemeten data van minimaal een week moet op een eenvoudige manier duidelijk zichtbaar kunnen worden gemaakt.

[E17] Het systeem moet eenvoudig te gebruiken zijn. Binnen 5 minuten moeten zowel arts als patiënt kunnen weten hoe het product werkt.

4 Concept

4.1 Hardware

Om een meetsysteem te ontwerpen die aan de functies en eisen voldoet is een succesvolle combinatie van hardware componenten nodig. Daarom zal in dit hoofdstuk besproken worden welke componenten er in het systeem gebruikt zijn. Alle componenten en hoe deze zijn aangesloten zijn te zien in bijlage A.

4.1.1 Besturingssysteem

Om het meetsysteem aan te sturen wordt gebruikt gemaakt van Arduino. Dit was een van de onderdelen waarvan voor de bacheloropdracht al is vastgesteld dat dit gebruikt zou worden. Arduino is een *open-source* computerplatform waarmee elektronische componenten aangestuurd kunnen worden.

Om aan de combinatie van functies en eisen te voldoen zal het systeem data moeten kunnen opslaan en de datum en tijd moeten kunnen bijhouden. Er zijn Arduino's die een of beide van deze functies al ingebouwd hebben. Bij een Arduino waarbij dit niet het geval is zullen de missende functies met Arduino compatibele modules verwezenlijkt moeten worden.

Om een welafgewogen keuze te maken tussen een volledig modulair systeem en een Arduino waar een of beide functies al ingebouwd zijn zullen alle voor- en nadelen worden overwogen.

Volledig modulair systeem

- Voordelen:
 - Bij een modulair systeem kan voor elke benodigde functie een specifieke hardware component worden gekozen. Hierdoor kan een optimaal systeem gecreëerd worden.
 - Het is over het algemeen goedkoper om een eenvoudige Arduino te combineren met modules dan een Arduino te gebruiken met de hier benodigde functies al ingebouwd.
 - Wanneer de keuze voor een Arduino niet is gelimiteerd aan Arduino's met een ingebouwde datum en tijd registratie en/of dataopslag functie kan een Arduino gekozen worden met laag-stroomverbruik mogelijkheden, zoals een slaapmodus.
- Nadelen:
 - Een modulair systeem bevat meer bedrading en componenten die bij beschadiging de werking van het systeem kunnen hinderen.
 - Een modulair systeem is minder compact dan een Arduino met geïntegreerde functionaliteiten.

Arduino met ingebouwde functies

- Voordelen:
 - Een gespecialiseerde Arduino is compacter dan een Arduino met verschillende modules.
 - Geïntegreerde systemen maken het systeem eenvoudiger en de kans op fouten kleiner.
- Nadelen:
 - Arduino's met een van beide opties ingebouwd zijn relatief duurder dan andere Arduino's die in combinatie met modules dezelfde functionaliteit bereiken. Tegelijkertijd beschikken deze Arduino's niet over de hiervoor besproken slaapmodus, waardoor het stroomverbruik hoger zal zijn.
 - De enige Arduino's met beide opties ingebouwd zijn de Arduino's van de MKR familie. Ook deze Arduino's zijn relatief duurder.
 - De I/O pins van de Arduino MKR borden hebben een operationele spanning van 3.3V. De loadcell waarmee gewerkt zal worden en behandeld wordt in hoofdstuk 4.1.3 heeft een aangeraden minimale excitatiespanning van 5V. Werken met deze Arduino's kan daardoor de nauwkeurigheid van de loadcell aantasten. [16]

Een Arduino met ingebouwde functies is dus eenvoudiger en compacter. Dit weegt echter niet op tegen het modulaire systeem wat kosteneffectiever is en betere stroombesparingsopties biedt. De afweging van de voor- en nadelen laat daarom zien dat een volledig modulair systeem beter geschikt is voor het beoogde doel.

Het specifieke model waar het systeem origineel mee is uitgewerkt is de Arduino UNO R3 Rev. Deze is gekozen door de stabiele reputatie, vele documentatie en mogelijkheid tot stroombesparingsmodi. Dit laatste kan van pas komen bij het voldoen aan eis E07; het minimaal 1 week kunnen gebruiken van de stroombron.

In de fysieke versie is de Arduino UNO R3 Rev vervangen door de vergelijkbare Arduino UNO R4 Minima. Dit is gedaan omdat deze op de meeste vlakken beter presteert. De Arduino R4 Minima heeft een werkspanning van 5V en werkt betrouwbaar bij temperaturen van $-40\text{ }^{\circ}\text{C}$ tot $+85\text{ }^{\circ}\text{C}$. [17]

4.1.2 Stroomvoorziening

De functies en eisen met betrekking tot de stroomvoorziening zijn F05, F06, E06, E13 en E14. Als het systeem hieraan moet voldoen zal een stroombron gebruikt moeten worden die minimaal 1 week werkt, de bewegingsruimte van de patiënt niet beperkt en makkelijk verwisselbaar is. Hiervoor is dus een mobiele en compacte stroombron nodig. Ook zal deze compatibel moeten zijn met de Arduino. De volgende opties zijn meegenomen in de keuze:

- Batterij(en)
- Accu
- Powerbank
- Zonnepaneel
- Dynamo

Hierbij hebben enkele van deze opties eigenschappen die niet optimaal zijn voor het voldoen aan de functies en eisen. Batterijen en accu's leveren geen constante spanning wat wel nodig is voor integratie met de Arduino. Deze heeft namelijk een spanning tussen 4.8V en 5.5V nodig via de USB poort. [17] Hierdoor zou een extra component nodig zijn om dit te verzekeren waardoor extra kosten gemaakt moeten worden. Bovendien zijn batterijen niet herlaadbaar en dienen ze na verloop van tijd te worden vervangen, een aspect dat, gezien de kosten en milieuoverwegingen, niet voordelig is. Bij het gebruik van een zonnepaneel of dynamo als stroombron, ontstaat een afhankelijkheid van externe factoren, wat onzekerheden met zich meebrengt. Een toevoeging hierop kan zijn dat er een stroom opslag wordt gebruikt waarbij uiteindelijk alsnog een batterij, accu of powerbank nodig is.

Een powerbank heeft in vergelijking met voorgaande opties geen last van de genoemde nadelen. Deze is oplaadbaar, levert een constante spanning, heeft geen extra componenten nodig om te werken en is niet afhankelijk van externe factoren om het systeem van stroom te voorzien. Daarom is dit de optie waarmee gewerkt zal worden.

De powerbank deze bacheloropdracht is gebruikt is de Anker - PowerCore III Sense powerbank 10000 mAh. Volgens de fabrikant is het werkende temperatuurbereik hiervan $-10\text{ }^{\circ}\text{C}$ tot $+45\text{ }^{\circ}\text{C}$. Later in de bacheloropdracht werd vastgesteld dat de daadwerkelijke effectieve capaciteit van de powerbank 5500 mAh is bij een spanning van 5V.

4.1.3 Kracht meten

Naast de Arduino is de hardware component verantwoordelijk voor het meten van de tractiekracht de enige component die al vaststond aan het begin van deze bacheloropdracht. Dit is de S610 miniatuur loadcell. [16] Op zichzelf voldoet deze al aan de functies en eisen F01, E08 en E11, maar is deze nog niet compatibel met Arduino of in staat betrouwbare metingen uit te voeren. Om dus ook te voldoen aan de andere relevante functies eisen, namelijk F02, F04, E03 en E04 is een meetversterker nodig. Hier zijn verschillende redenen voor:

1. De signalen die een loadcell genereert zijn te zwak om goed uitgelezen te kunnen worden door een Arduino. Een meetversterker versterkt deze signalen tot het niveau waar ze door de Arduino uitgelezen kunnen worden.

2. Loadcellen zijn gevoelig voor ruis en interferentie van elektrische signalen, waardoor de signalen niet goed uitgelezen kunnen worden door een Arduino. Een meetversterker filtert dit en zorgt ervoor dat alleen het relevante signaal overblijft.
3. Meetversterkers bieden de mogelijkheid om de loadcell te kalibreren en de lineariteit van de loadcell aan te passen. Dit is cruciaal voor het verkrijgen van nauwkeurige en consistente resultaten.

De gekozen meetversterker is de HX711. Dit is een veelgebruikte meetversterker in combinatie met loadcellen en Arduino. [18, 19] Deze heeft een voedingsspanning bereik van 2.8 tot 5.5V en het betrouwbare temperatuurbereik van de HX711 is -40 tot +85 °C. [20]

De loadcell heeft een laag stroomverbruik en een capaciteit van 0.9 tot 50kg. De aangeraden spanning om de loadcell mee te exciteren is 5 - 10V en het betrouwbare temperatuurbereik is 0 tot +80 °C. Twee belangrijke eisen waaraan de loadcell moet voldoen, E08 en E11, stellen dat de loadcell een tractiekracht van 45kg aan moet kunnen en een veiligheidsfactor moet hebben van ≥ 2.2 . Volgens de fabrikant van de loadcell is deze gemaakt om krachten tot 45kg te meten en heeft deze een veiligheidsfactor van 3. [16]

4.1.4 Datum en tijd bijhouden

Voor de hardware component verantwoordelijk voor het bijhouden van de datum en tijd zijn de volgende functies en eisen van toepassing: F03, F04, E05. Deze component zal dus de actuele datum en tijd moeten kunnen bijhouden en mogelijkheid bieden tot het opslaan ervan. Zoals bij de rest van de componenten zal deze dus ook compatibel moeten zijn met Arduino. De volgende opties zijn in overweging genomen:

- Interne klok van de Arduino
- Real-Time Clock (RTC) module
- GPS module
- Wifi module

Voor het gebruiken van de interne klok van de Arduino is vereist dat deze een constante stroomtoevoer heeft, omdat anders de actuele datum en tijd verloren gaat. Aangezien dit met een powerbank niet gegarandeerd kan worden valt deze optie af. De GPS module en de wifi module gebruiken respectievelijk een GPS verbinding en wifi verbinding om de datum en tijd bij te houden. Hierbij is het van belang dat er een constante verbinding met deze signalen is wat niet altijd gegarandeerd kan worden. Ook verbruiken deze modules relatief veel stroom.

RTC-modules verbruiken over het algemeen minder stroom in vergelijking met GPS- en wifi-modules, omdat ze specifiek zijn ontworpen voor een laag stroomverbruik, in tegenstelling tot de laatstgenoemden, die meestal niet worden gebruikt in systemen met een laag stroomverbruik. Dit is dan ook de optie welke in het uiteindelijke systeem is gebruikt.

De specifieke module die is gebruikt is de DS3231 RTC module. De module beschikt over een knoopcel batterij zodat ook na afsluiting van de stroomtoevoer naar het systeem, zoals wanneer de powerbank wordt losgekoppeld, de datum en tijd zullen kloppen.

De originele versie van de module beschikt over een circuit wat in staat is de knoopcel batterij op te laden. Dit zou een goede toevoeging kunnen zijn aan het systeem, omdat je er daarmee voor kunt zorgen dat de batterij wordt opgeladen door de powerbank en daardoor nooit vervangen hoeft te worden. Er is echter gedocumenteerd dat het opladen van een dergelijke batterij enkele aanpassingen vergt. Zo zal de origineel meegeleverde CR2032 vervangen moeten worden door een oplaadbare LIR2032 batterij. Ook zal de spanning naar deze module gereduceerd moeten worden naar maximaal 4.7V in vergelijking met de 5V die gebruikt wordt door de rest van het systeem. Wanneer dit niet gedaan wordt kan de batterij ontploffen. [21]

Volgens de datasheet van de DS3231 RTC module is het maximale stroomverbruik van de batterij tijdens het bijhouden van de tijd zonder externe stroomtoevoer maximaal 3.0 μ A bij een spanning van 3.63V. [22] Met een CR2032 met een capaciteit van 220 mAh levert dit een theoretische levensduur op van minimaal $\frac{220}{3.0 \cdot 10^{-3} \cdot 24 \cdot 365.25} = 8.37$ jaar.

Daarom is bepaald dat de originele batterijduur lang genoeg is om de voordelen van een oplaadbare batterij in de RTC module met de daarbij komende kosten van extra onderdelen of eventuele veiligheidsrisico's te compenseren.

Om het oplaad circuit uit te schakelen is er een kleine knip gemaakt in de aansluiting met de rechter VCC pin welke in figuur 4 in het rood te zien is. Ook belangrijk om te vermelden is dat de LED waar de DS3231 RTC module standaard mee is uitgerust hier vanaf is gehaald om stroom te besparen.

Het temperatuurbereik waarin de RTC module nauwkeurig datum en tijd kan bijhouden is -40° tot $+85^{\circ}$.

4.1.5 Data verwerken

Voor het verwerken van de data zijn meerdere stappen nodig waarbij de volgende functies en eisen van toepassing zijn:

1. De data moet (tijdelijk) kunnen worden opgeslagen: F04, E04 en E05.
2. De data moet naar een computer overgezet kunnen worden: F08.1 en E15.
3. De data moet op eenvoudige wijze zichtbaar kunnen worden gemaakt: F08.2 en E16.

De volgende opties zijn hiervoor overwogen:

- micro SD kaart module
- Wifi module
- Bluetooth module
- Seriële communicatie (USB)

Om data te verwerken met de wifi- en bluetooth module zal de data tijdelijk opgeslagen moeten worden om deze op een later moment in zijn geheel naar een computer over te zetten. Zoals eerder vermeld kan het behouden van een constante verbinding om daarmee data continu data te verzenden namelijk niet gegarandeerd worden.

Ook de optie om seriële communicatie te gebruiken om de data naar een computer te verzenden zal gebruik moeten maken van een tijdelijke opslag. De gekozen Arduino beschikt over 256 kilobyte aan opslagruimte. Er is geobserveerd dat een enkele regel data in een tekstbestand 24 bytes aan opslag is. Om de datum, tijd en kracht iedere tien seconden op te slaan zouden maximaal $\frac{256000}{24} = 10666$ metingen gedaan kunnen worden. Dit komt neer op een theoretische maximale meetperiode van $\frac{10666}{6 \cdot 60} = 29,6$ uur, wat niet voldoende is om aan de opgestelde eisen te voldoen.

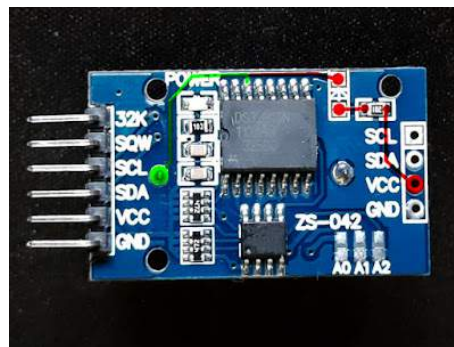
Om een wifi module, bluetooth module of seriële communicatie te gebruiken voor data verwerking zal dus een externe opslag gebruikt moeten worden. In dat geval is de optie om een micro SD kaart module te gebruiken logischer. Op die manier is er op dit gebied namelijk slechts behoefte aan een enkele hardware component wat voor zowel de kosten, fysieke omvang als het stroomverbruik voordeliger is.

4.1.6 Bediening

Het systeem moet een mogelijkheid bieden om wanneer dit gewenst is op eenvoudige wijze een meting te starten en op een later moment deze meting ook weer te kunnen resetten met tegelijkertijd duidelijke feedback. Op die manier kan namelijk voldaan worden aan functies en eisen F07.1, F07.2 en E17.

De volgende opties zijn in overweging genomen:

- Fysieke knoppen en LED's
- Bluetooth verbinding



Figuur 4: Het oplaadcircuit van de DS3231 RTC module. [21]

- Wifi verbinding

De combinatie van fysieke knoppen en LED's zal de gebruiker een eenvoudige manier bieden om het systeem te bedienen. Deze optie is daarnaast ook relatief goedkoop en laag in stroomverbruik.

Met een bluetooth of wifi verbinding zou het mogelijk zijn om een meting te starten met een mobiele applicatie of via een webbrowser, waarmee duidelijkere feedback geboden kan worden. Dit maakt het voor de gebruiker inzichtelijker wanneer een meting is gestart of gereset. Bij deze opties zullen wel extra beveiliging stappen ondernomen moeten worden zodat onbevoegden geen toegang krijgen tot de bediening van het systeem. Daarnaast gebruiken modules die een dergelijke verbinding tot stand moeten brengen veel stroom zoals eerder vermeld.

Omdat de combinatie van fysieke knoppen en LED's relatief meer voordelen heeft is deze optie gekozen. In het systeem wordt gebruik gemaakt van twee tactiele schakelaars. Dit zijn kleine drukschakelaars die eenvoudig op de Arduino aangesloten kunnen worden. De schakelaar waarmee de meting gestart kan worden is zwart van kleur en degene die de meting weer stop zet is rood van kleur. Zo kan eenvoudig onderscheid gemaakt worden.

Het systeem zal ook gebruik maken van twee LED's; een gele en een rode. Deze geven bij het drukken op een van de schakelaars feedback zodat het voor de gebruiker duidelijk is in welke staat het systeem is. Een meer gedetailleerde uitwerking hiervan wordt behandeld in hoofdstuk 4.2. Voor aansluiting op de Arduino zijn beide knoppen gecombineerd met een 10000Ω weerstand en beide LED's met een 220Ω weerstand. De aansluiting van deze componenten is te zien in bijlage A.

4.2 Software

Om de hardware componenten aan te sturen is de juiste software nodig. Hoe deze software opereert zal hier beschreven worden.

4.2.1 Algemene code

Om alle hardware componenten aan te sturen is een enkele code gebruikt welke naar de Arduino is geüpload. De code is geschreven in Arduino IDE, de programmeertaal van Arduino. Een overzicht van de code is te zien in bijlage C en de volledige code is te zien in bijlage D. Het primaire deel van de code bestaat uit een *setup* welke bij het aansluiten van het systeem op een stroombron eenmaal uitgevoerd zal worden en een *loop* welke hierna constant opnieuw uitgevoerd zal worden tot de stroom weer wordt afgesloten. Vanuit de loop worden andere functies aangeroepen met een specifieke rol welke op hun beurt ook weer functies kunnen aanroepen.

Er zijn 3 hoofdfuncties in de code:

- CheckKnoppenStatus
- MetingOpzet
- DataOpslaan

CheckKnoppenStatus

Deze functie is via de loop constant actief en controleert of de start- of stopknop wordt ingedrukt. Wanneer de stopknop als laatste is ingedrukt staat het systeem stand-by en gebeurt er verder niets. Wanneer de startknop als laatste is ingedrukt wordt eenmaal de MetingOpzet functie aangeroepen waarna elke 10 seconden de DataOpslaan functie wordt uitgevoerd. Ook worden de LED's via de CheckKnoppenStatus functie aangestuurd om feedback aan de gebruiker te geven over de status van het systeem.

MetingOpzet

Via deze functie wordt het systeem gereed gemaakt voor een nieuwe meting wanneer de startknop is ingedrukt. Deze functie controleert eerst of de micro SD kaart goed is aangesloten en geeft de gebruiker via de LED's feedback wanneer dit niet zo is. Hierna worden andere functies aangeroepen waarmee de loadcell wordt geïnitieerd, gecontroleerd wordt welke bestandsnaam gebruikt zal worden en een tekstbestand met deze naam wordt aangemaakt op de micro SD kaart. Ook past deze functie de datum en tijd aan naar 00/00/00 00:00:00 wanneer in de situatie van ontkoppelde

powerbank ook de batterij van de RTC module leeg is geraakt. Op die manier kan eenvoudig vastgesteld worden wanneer deze situatie heeft plaatsgevonden.

DataOpslaan

Via deze functie worden zowel de datum en tijd van de RTC module als de kracht van de loadcell gecontroleerd. Deze worden hierna op de micro SD kaart opgeslagen in het tekstbestand wat voor de meting is aangemaakt.

4.2.2 Databeheer

Om de data overzichtelijk weer te geven en makkelijk toegankelijk te maken voor de gebruiker is rekening gehouden met de manier van dataopslag.

Zoals hiervoor vermeld zal de data opgeslagen worden op een micro SD kaart. Deze data moet hierna op een computer eenvoudig zichtbaar kunnen worden gemaakt. Hierom is ervoor gekozen om de data op te slaan in tekstbestanden. Dit is namelijk het enige bestandstype waar de Arduino data op kan schrijven en ook eenvoudig te openen is in Microsoft Excel waarmee de data makkelijk verwerkt kan worden.

De bestandsnamen zijn gelimiteerd tot 8 tekens doordat er gewerkt wordt met Arduino. Op de micro SD-kaart wordt voor elke dag waarop een meting wordt gestart, een specifieke map gemaakt. De naam van deze map is de volledige datum van die dag. Binnen deze map wordt voor elke meting die op die dag begint, een nieuw tekstbestand gecreëerd. De bestandsnaam bestaat uit de dag, de maand en een numeriek label dat aangeeft welke meting het is, zoals weergegeven in tabel 1.

Daarnaast is er een map genaamd '*FILES*' aanwezig op de micro SD kaart. In deze map bevindt zich een tekstbestand genaamd '*UITLEG0*' welke gebruikt wordt om in elke aangemaakte map een tekstbestand te kopiëren waarin wordt uitgelegd hoe de data verwerkt kan worden en hoe het systeem bestuurd kan worden.

Tabel 1: Voorbeeld van beheer van bestanden op de micro SD kaart.

micro SD kaart mappen	metingmappen bestanden
FILES	DataVerwerking.xlsx UITLEG0.txt
01012024	0101-1.txt 0101-2.txt UITLEG.txt
02012024	0201-1.txt 0201-2.txt 0201-3.txt UITLEG.txt

4.2.3 Gebruikersfouten

De kans dat de gebruiker van het systeem altijd exact de juiste handeling zal uitvoeren is miniem. Daarom is in de software rekening gehouden met elke ongebruikelijke situatie die plaats zouden kunnen vinden. In tabel 2 zijn alle systeemhandelingen te zien bij de druk op een van de knoppen in verschillende situaties. Er zijn slechts twee situaties, aangeduid met groen, waarin een knop invoer leidt tot handeling van het systeem in combinatie met feedback van de LED's. In overige situaties zal een knop invoer enkel worden beantwoord met feedback van de LED's.

Het zou ook kunnen gebeuren dat tijdens een meting de powerbank leeg raakt of deze wordt ontkoppelt en daarnaast zou de micro SD kaart tijdens de meting kunnen worden ontkoppelt. In deze gevallen zal een meting niet verder kunnen gaan. Om echter te voorkomen dat de tot dan toe

gemeten data beschadigd raakt of verdwijnt wordt na elk specifiek meetmoment de data opgeslagen op de micro SD kaart zodat deze data achteraf wel beschikbaar is.

Tabel 2: Systeemhandelingen bij knopinvoer in mogelijke situaties.

Powerbank opgeladen en aangesloten?	micro SD kaart goed aangesloten?	Is er een meting bezig?	Knop invoer	Actie
Ja	Ja	Ja	Start	Gele LED blinkt drie keer
Ja	Ja	Ja	Stop	Meting stopt en rode LED blinkt eenmaal
Ja	Ja	Nee	Start	Meting start en gele LED blinkt eenmaal
Ja	Ja	Nee	Stop	Rode LED blinkt drie keer
Ja	Nee	Nee	Start	Beide LED's blinken drie keer
Ja	Nee	Nee	Stop	Rode LED blinkt drie keer
Nee	-	Nee	Start/Stop	Er gebeurt niets

4.3 Integratie in huidige HGT systeem

In het UMC zijn er momenteel twee rolstoelmodellen welke worden gebruikt voor HGT. Over het algemeen wordt voor volwassenen de QUICKIE Life i gebruikt en voor kinderen en kleinere patiënten de ZIPPIE Youngster 3. [3]

Het ontwerp van de HGT constructie is op deze rolstoelen verschillend en daarnaast worden er momenteel per rolstoelmodel ook nog verschillende HGT constructies gebruikt. Zo verschilt tussen constructies de lengte van de horizontale balk en de grootte van de schoor die de horizontale en verticale balk ondersteunt. Om ervoor te zorgen dat de hardware van het meetsysteem op iedere versie van het HGT systeem past zal gekeken moeten worden naar de overeenkomsten tussen de modellen.

Alle verschillende HGT modellen hebben in ieder geval twee dingen gemeen:

1. Ze bevatten twee balken die loodrecht op elkaar staan met een schoor ter versteviging, zie figuur 5.
2. Het koord en de katrollen bevinden zich aan de rechterkant van het frame.

Daarom is er gekozen voor een hardware omhulsel dat een rechte hoek bevat en open kan aan de linkerkant. De vorm van het omhulsel zal de vorm van de schoor volgen en daarmee semi-driehoekig zijn. Op die manier wordt de ingenomen ruimte zoveel mogelijk geconcentreerd buiten de bewegingsruimte van de patiënt.

Omdat de loadcell moet worden gemonteerd tussen het koord van het HGT systeem zal dit de enige component zijn welke zich buiten het hardware omhulsel bevind. Het huidige koord zal intact gelaten worden. Op deze manier kan namelijk verzekerd worden dat in het geval dat de loadcell



Figuur 5: De bovenconstructie van het HGT frame met de horizontale balk, verticale balk en de schoor.

of de bevestiging hiervan met het koord breekt, de tractiekracht op de patiënt niet plotseling verdwijnt.

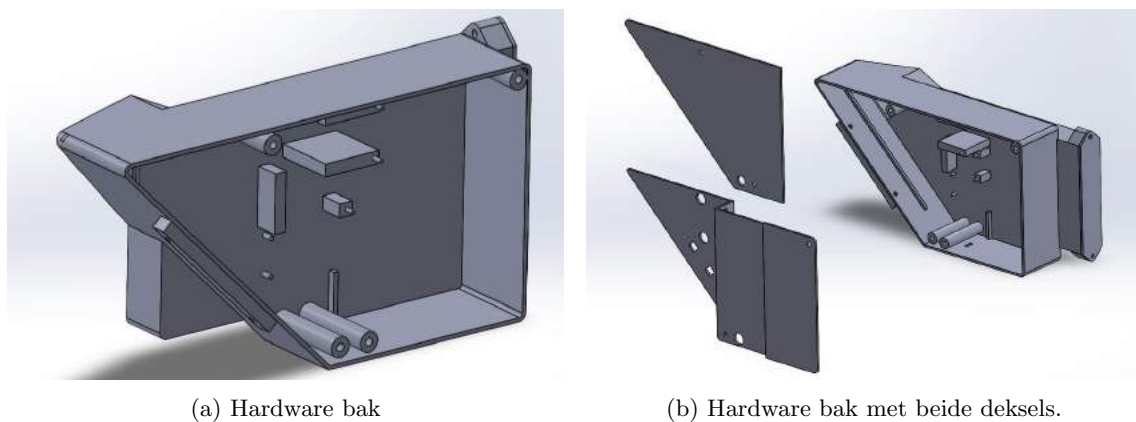
5 Realisatie

Om de hardware aan het HGT-frame te monteren en geschikt te maken voor gebruik, is een omhulsel nodig. Op basis van de gekozen hardwarecomponenten, functies en eisen zijn hieruit drie opeenvolgende ontwerpen voortgekomen, waarbij elk ontwerp een verbetering is ten opzichte van het vorige. Alle ontwerpen en hun bijbehorende eigenschappen worden hier uiteengezet.

5.1 Ontwerp 1

Om alle negen hardware componenten (Arduino, RTC module, micro SD kaart module, meetversterker, powerbank, twee LED's en twee knoppen) in het ontwerp te verwerken is er bij het eerste ontwerp voor gekozen om een nieuw ontwerp te creëren in tegenstelling tot een al bestaand omhulsel te zoeken. Hiermee kan namelijk een ontwerp verzekerd worden wat optimaal bij het huidige HGT frame past en tegelijk een overzichtelijke uitstraling behoudt voor de gebruiker.

Dit ontwerp is gemaakt met het 3D CAD tekenprogramma SolidWorks. Het ontwerp, zie figuur 6, is zo ontworpen dat het geprint kon worden met een 3D printer. 3D ontwerpen zijn namelijk lichtgewicht, sterk genoeg voor deze toepassing en er zijn eenvoudig specifieke en unieke ontwerpen mee te maken.



Figuur 6: 3D SolidWorks tekening van ontwerp 1.

Zoals in het figuur is te zien heeft dit ontwerp twee deksels. Op deze manier is het mogelijk om tijdens een meting alle hardware componenten af te schermen en bij het starten of stoppen van een meting enkel de componenten zichtbaar te hebben die daarbij nodig zijn. Zo is er minder kans dat een meting onopzettelijk wordt onderbroken, maar is het voor de gebruiker wel duidelijk welke onderdelen gebruikt kunnen worden voor bediening van het systeem.

Met ditzelfde idee kan uitgelegd worden waarom de deksels gemonteerd zijn met bouten en moeren. Hiermee zijn de deksels eenvoudig genoeg om open te maken door de gebruiker, maar niet eenvoudig genoeg om onbevoegden moeiteloos toegang te geven tot de besturing.

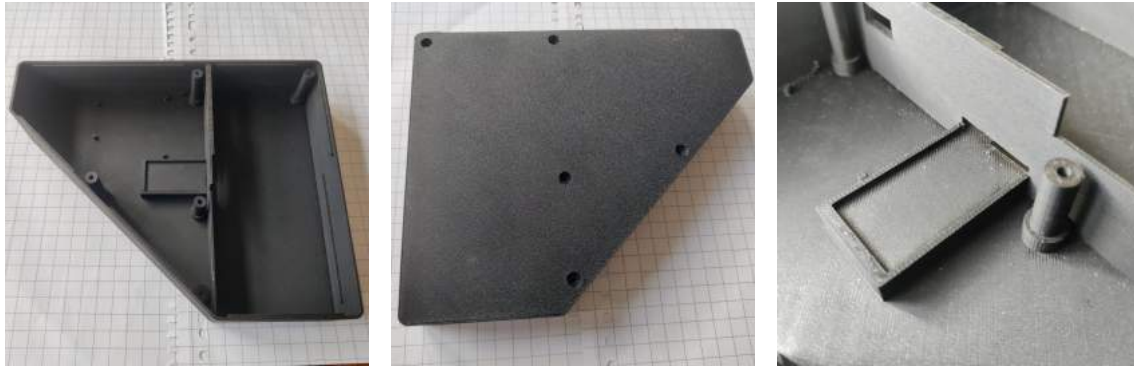
Ook is in dit ontwerp te zien dat er zich enkele pinnen in de bak bevinden. Hieraan kunnen de hardware componenten vastgeklit worden zodat ze niet bewegen tijdens het gebruik wat de kans op schade minimaliseert.

Verder is belangrijk om te benoemen dat dit ontwerp met zijn fysieke vorm vastgemaakt zou kunnen worden aan het HGT frame. Er kan namelijk een balkje aan het frame vastgeschroefd worden waardoor het omhulsel in de open driehoek van het HGT frame vast komt te zitten. Deze manier van bevestigen is ontworpen op het moment waarop de veronderstelling heerste dat er voor het HGT systeem één uniforme maat bestond. Later is duidelijk geworden dat er momenteel nog gebruik wordt gemaakt van meerdere maten waardoor deze bevestigingsmethode niet optimaal is.

5.2 Ontwerp 2

Om een omhulsel te creëren wat op alle versies van het HGT frame past is een tweede ontwerp uitgewerkt welke te zien is in figuur 7. Hierbij is afgestapt van het fysieke klemsysteem en een ontwerp bedacht waarbij het omhulsel met klittenband aan het frame bevestigd kan worden. Dit is sterk genoeg om het omhulsel in normale omstandigheden aan het frame te houden, maar kan ook makkelijk losgehaald worden wanneer dit nodig is. Zoals in figuur 7b is te zien, zijn bouten en

moeren verzonken in het omhulsel zodat de zijkanten plat zijn en niet in de weg zitten voor een klittenband verbinding.



(a) De bak van het hardware omhulsel zonder deksels.

(b) Zijkant van het omhulsel met gaten voor de moeren.

(c) De sleuf voor de SD kaart module en een boutbuis.

Figuur 7: Verschillende onderdelen van het fysieke ontwerp van ontwerp 2.

Bij dit ontwerp is er ook een omhulsel voor de loadcell bijgekomen. Deze moet extra bescherming bieden voor de loadcell welke gevoelig is voor beschadigingen en niet gebruikt mag worden in vochtige omgevingen. [23] Om de loadcell tussen het koord te bevestigen zonder het koord op te splitsen is hier gekozen om gebruik te maken van eivormige staaldraad klemmen, welke het koord aan beide kanten van de loadcell in een lus kunnen houden. Deze lussen kunnen dan via stalen ringen aan de loadcell verbonden worden.

Van dit tweede ontwerp is, zoals in het figuur te zien is, een fysiek versie gerealiseerd. Hierbij zijn enkele tekortkomingen van het ontwerp aan het licht gekomen:

1. De voornaamste tekortkoming was dat de powerbank bij toevoeging van de USB kabel niet meer in het ontwerp paste.
2. Er zijn op meerdere punten exacte maten gebruikt. Hierdoor pasten de bouten en de SD kaart module niet goed in de hiervoor bestemde gaten, welke in figuur 7c te zien zijn.
3. De pinnen waaraan de hardware componenten bevestigd kunnen worden bleken fragiel.

5.3 Ontwerp 3

In het derde en tevens ook definitieve ontwerp zijn de tekortkomingen van de eerste ontwerpen opgelost. Het omhulsel is vergroot en er is speling aangebracht op de plaatsen waar dit nodig was. Ook zijn de pinnen voor de hardware componenten uit dit ontwerp gehaald en zijn deze vervangen voor kleine stukken klittenband die de componenten tijdens gebruik van het systeem op hun plek houden. Dit ontwerp is te zien in figuren 8a en 8b.



(a) Volledige hardware omhulsel met bovendeksel losgemaakt.

(b) Binnenkant van de bak met bediening zichtbaar.

(c) Loadcell in 3D geprint omhulsel aan het koord geknoopt.

Figuur 8: Fysieke ontwerp van ontwerp 3.

Bij het testen van dit ontwerp werd een niet eerder vertoond obstakel duidelijk. De powerbank schakelt zichzelf uit wanneer het stroomverbruik onder een drempelwaarde komt waar het huidige meetsysteem wel onder komt. Hierdoor was de maximale meettijd van dit systeem anderhalve minuut.

Om dit obstakel te overkomen is een combinatie van hardware en software gebruikt op basis van personen die een vergelijkbaar probleem ervaren. [24] Aan de hardware is een *metal-oxide-semiconductor field-effect transistor (MOSFET)* toegevoegd in combinatie met twee weerstanden van respectievelijk 20Ω en 10000Ω . Deze zijn te zien in bijlage A.

Via de software zorgt deze MOSFET ervoor dat elke minuut een verhoging in het stroomverbruik aan de powerbank wordt gevraagd gedurende een periode van honderd milliseconden waardoor deze aan blijft staan. Dit resulteert in een verhoogd stroomverbruik van het systeem, maar is essentieel om de maximale meettijd te verhogen.

Het gemiddelde stroomverbruik van het meetsysteem is hierna vastgesteld met een USB volt- en Ampèremeter. Dit is 70 mA. Met de eerder genoemde effectieve powerbank stroomcapaciteit van 5500 mAh is de theoretische maximale meettijd $\frac{5500}{70 \cdot 24} = 3.27$ dagen.

Bij het bevestigen van de loadcell inclusief omhulsel tussen het koord werd duidelijk dat de eerder bedachte constructie met klemmen en ringen in het horizontale vlak teveel ruimte in zou nemen om de HGT behandeling goed uit te kunnen voeren. Daarom is gebruik gemaakt van knopen rechtstreeks aan de loadcell wat significant minder ruimte in beslag nam en uitstekend functioneerde.

5.4 Risicoanalyse

Een faalmodus en effectanalyse (FMEA) is een gestructureerde methode om zwakheden in systemen en processen te analyseren voordat ze plaatsvinden. Deze methode wordt veelal in de gezondheidszorg gebruikt om potentiële risico's op faalmodi te verminderen. Een faalmodus wordt in deze context gedefinieerd als een potentiële manier waarop een systeem of proces kan falen. [25, 26]

Voor het analyseren van deze faalmodi worden drie eigenschappen van fouten gebruikt:

- Ernst (**S**): Dit geeft de mate aan waarin de mogelijke fout een effect heeft op de patiënt, systeem of proces en hoe significant de consequenties zijn.
- Kans op voorkomen (**P**): Dit geeft aan hoe vaak de mogelijke fout waarschijnlijk zal plaatsvinden.
- Detecteerbaarheid (**D**): Dit geeft aan hoe waarschijnlijk het is dat de mogelijke fout gedetecteerd zal worden.

De mogelijke scores van de fouteigenschappen en de bijbehorende definities hiervan zijn te zien in tabel 3.

Tabel 3: Scoreschaal voor ernst, kans op voorkomen en detecteerbaarheid van fouten.

	Definitie	Score
Ernst (S)	<u>Geen effect</u> (Fout is niet merkbaar en heeft geen invloed op de patiënt of het systeem of proces.)	1
	<u>Licht effect</u> (Fout veroorzaakt kleine gevolgen of is een probleem voor de patiënt, systeem of proces, zonder dat er letsel of een hoger zorgniveau nodig is.)	2
	<u>Matig effect</u> (Fout veroorzaakt enig prestatieverlies voor de patiënt, kan het zorgniveau verhogen of is volledig catastrofaal voor het systeem of proces.)	3
	<u>Groot effect</u> (Fout veroorzaakt een hoge mate van prestatieverlies, met blijvende gevolgen voor de patiënt.)	4
	<u>Ernstig of catastrofaal effect</u> (Fout veroorzaakt de dood of groot, permanent verlies van functie.)	5
Kans op voorkomen (P)	Jaarlijks of minder vaak	1
	Maandelijks	2
	Wekelijks	3
	Dagelijks	4
	Ieder uur of vaker	5
Detecteerbaarheid (D)	(Bijna) altijd	1
	Waarschijnlijk	2
	Niet waarschijnlijk	3
	(Bijna) nooit	4

Potentiële faalmodi zijn geanalyseerd voor elke systeemcomponent en de criticiteit ervan is bepaald aan de hand van het Risico Prioriteit Nummer (RPN). Dit is een numerieke score welke berekend wordt aan de hand van de eigenschappen van de fouten ($RPN = S \cdot P \cdot D$). De theoretische minimale RPN is 1 en het theoretische maximum is 100.

Voor de grens van een acceptabele RPN score is bepaald dat dit is bij $S = 2$, $P = 3$ en $D = 2$. Dit geeft $RPN = 2 \cdot 3 \cdot 2 = 12$. Faalmodi met een hogere RPN score vormen een te groot risico en hier zal naar moeten worden gekeken.

De volledige FMEA is te zien in bijlage B en laat zien dat er geen significante risico's aanwezig zijn in het systeem. Belangrijk om te vermelden is dat dit een inschatting is aangezien het systeem tijdens deze bacheloropdracht niet is getest.

5.5 Evaluatie

Als laatste stap van de realisatie is het wezenlijk om te evalueren of het definitieve systeem voldoet aan de opgestelde functies en eisen.

In tabel 4 is te zien aan welke functies en eisen het systeem voldoet, aan welke niet en van welke functies en eisen in verder onderzoek bepaald kan worden of het systeem hieraan voldoet. De evaluatie wijst erop dat het systeem voldoet aan alle functies en de meeste eisen. Eisen E01, E07 en E10 echter niet. Van eisen E09, E14, E15, E16 en E17 zal in nader onderzoek de voldoening bepaald moeten worden.

Tabel 4: Functies en eisen evaluatie; ✓ = voldaan, ✗ = niet voldaan, ✧ = nader te bepalen.

Functies				Eisen			
[F01]	✓	[F07]	✓	[E01]	✗	[E10]	✗
[F02]	✓	[F07.1]	✓	[E02]	✓	[E11]	✓
[F03]	✓	[F07.2]	✓	[E03]	✓	[E12]	✓
[F04]	✓	[F08]	✓	[E04]	✓	[E13]	✓
[F05]	✓	[F08.1]	✓	[E05]	✓	[E14]	✧
[F06]	✓	[F08.2]	✓	[E06]	✓	[E15]	✧
				[E07]	✗	[E16]	✧
				[E08]	✓	[E17]	✧
				[E09]	✧		

[E01] Het systeem moet spatwaterdicht zijn.

In verband met tijdsbeperkingen is deze specifieke eis opzij gezet om te kunnen voldoen aan eisen met een hogere prioriteit. Dit is namelijk een eis die in vergelijking met de andere eisen eenvoudiger door de gebruiker zelf op te lossen valt wanneer dit nodig wordt geacht. In het fysieke ontwerp is daarom geen rekening gehouden met het spatwaterdicht maken van het hardware omhulsel.

[E07] Het product moet een energiebron bevatten waarmee het systeem minimaal 1 week kan blijven meten.

Zoals eerder vermeld kan de huidige powerbank het systeem naar verwachting ongeveer 3.27 dagen van stroom voorzien. Er zal dus een aanpassing gedaan moeten worden als er een meting van minimaal een week gedaan moet worden.

[E10] Het meetsysteem moet betrouwbare metingen kunnen geven bij temperaturen van -10°C tot en met +40°C.

Het gekozen temperatuurbereik is op basis van gebruikelijke Nederlandse temperaturen tot stand gekomen, zodat het systeem bij patiënten van het UMC Utrecht doorgaans ook buiten gebruikt kan worden. Bij het analyseren van de hardwarecomponenten blijkt dat alleen de gebruikte loadcell niet voldoet aan de vereiste betrouwbaarheid voor het volledige bereik. De resultaten hiervan kunnen volgens de fabrikant niet als betrouwbaar worden beschouwd onder de 0 °C.

6 Discussie

Het definitieve ontwerp is geanalyseerd aan de hand van de opgestelde functies en eisen en hierbij is enkel gebruik gemaakt van theoretische evaluatie. Er is tijdens deze bacheloropdracht namelijk geen praktijkonderzoek of andere evaluatiemethode uitgevoerd. De belangrijkste functies en eisen hebben betrekking tot het meten en opslaan van de tractiekracht, het bijhouden en opslaan van de datum en tijd en het voorzien van stroom aan het systeem.

Er zijn drie eisen waar het systeem niet aan heeft voldaan. De eerste, eis E01, stelt dat het systeem spatwaterdicht moet zijn. Zoals vermeld is deze eis in verband met tijdsbeperkingen buiten beschouwing gelaten, zodat andere eisen geprioriteerd konden worden. Spatwaterdichtheid van het systeem is belangrijk, omdat de gebruikte hardware componenten kunnen beschadigen van contact met water. Gebruikers zouden dit zelf op kunnen lossen door bijvoorbeeld een hoes over het hardware omhulsel te gebruiken, de dekselranden af te plakken of het systeem niet in situaties te gebruiken waar spatwater een mogelijk gevaar kan zijn, maar dit is niet optimaal.

De tweede eis waar niet aan is voldaan is E07. Deze eis stelt dat het product een energiebron moet bevatten waarmee het systeem minimaal 1 week kan blijven meten. In hoofdstuk 5.3 is echter berekend dat de theoretische maximale meetduur 3.27 dagen is. Hiervoor zijn drie redenen:

1. Bij het initiële onderzoek naar de geschikte hardware componenten kwam de Arduino UNO R3 Rev naar voren als beste optie voor het systeem. Bij bestelling van de onderdelen is echter de Arduino UNO R4 minima besteld, omdat deze toen, naar verwachting, een betere optie zou zijn. Later werd duidelijk dat deze nieuwe Arduino niet beschikt over de stroombesparingsmodi die de originele Arduino wel heeft.
2. De powerbank heeft een lagere effectieve stroomcapaciteit dan waar van tevoren mee rekening was gehouden.
3. In latere stadia van de bacheloropdracht werd bekend dat er verschillende Arduino's zijn die, ondanks dat ze niet over stroom besparende modi beschikken, wel energiezuinig zijn. Dit type Arduino is niet meegenomen in de keuze.

Hierdoor zal de energiebron tijdens een week waarin de patiënt in tractie is verwisseld of opgeladen moeten worden.

Een vervolgonderzoek zou een grondige vergelijking moeten omvatten van alle beschikbare Arduino-modellen, met als specifiek doel het identificeren en kiezen van het meest energie-efficiënte systeem. Ook zal een powerbank met een grotere capaciteit gebruikt moeten worden. Vaak betekent dit ook dat de powerbank groter zal zijn, waardoor ook het hardware omhulsel aangepast zal moeten worden.

E10 is de derde eis waar in deze bacheloropdracht niet aan voldaan is. Deze stelt dat het meetsysteem betrouwbare metingen moet kunnen geven binnen een temperatuurbereik van $-10\text{ }^{\circ}\text{C}$ tot $+40\text{ }^{\circ}\text{C}$. De loadcell is de enige hardware component waarvan de fabrikant niet kan garanderen dat deze betrouwbaar functioneert in het volledige gegeven temperatuurbereik. Dit heeft als gevolg dat wanneer de patiënt zich tijdens tractie buiten begeeft bij temperaturen onder de $0\text{ }^{\circ}\text{C}$, verkeerde meetresultaten gegeven kunnen worden. Bij een verbetering van het systeem zal dus een loadcell benodigd zijn welke ook lagere temperaturen betrouwbare metingen kan geven.

Naast de hiervoor genoemde eisen zijn er ook vijf eisen waarvan in later onderzoek bepaald zal moeten worden of het huidige systeem hieraan heeft voldaan.

De eerste hiervan, eis E09, stelt dat het product minimaal 20 behandelingen mee zal moeten kunnen gaan. Het product is ontworpen om hieraan te voldoen, maar praktijkonderzoek zal moeten uitwijzen of dit ook daadwerkelijk het geval is.

Eisen E14, E15, E16 en E17 hebben allemaal betrekking op gebruikerservaringen. Aangezien het product nog niet in de praktijk is getest kan hier nog geen uitspraak over gedaan worden.

Naast het behalen van de functies en eisen kwamen er na het implementeren van het definitieve ontwerp nog enkele verbeterpunten naar boven:

- Het huidige ontwerp maakt gebruik van bouten en moeren om de deksels op de hardware bak de monteren. Omdat deze verzonken zijn in het ontwerp bleek het moeilijk om grip op de moeren te krijgen en de bouten los te kunnen schroeven. In een verbeterd ontwerp zal

hier een nieuw systeem op moeten worden bedacht waarbij de moeren bijvoorbeeld in het omhulsel zitten geklemd of gelijmd. Ook zou schroefdraad in het ontwerp kunnen worden aangebracht zodat er geen moeren gebruikt hoeven te worden.

- De powerbank wordt momenteel in het ontwerp gehouden door een bak verbonden aan de bovendeksel. Deze bak vertoonde na montage enkele scheuren. In een verbeterd ontwerp zal deze bak dus sterker gemaakt moeten worden.

7 Conclusie

In deze bacheloropdracht is een meetsysteem ontworpen voor het registreren van de tractiekracht van een veer-gebaseerd HGT systeem.

Het uiteindelijke systeem heeft niet voldaan aan alle eisen, maar beschikt desondanks over alle opgestelde functionaliteiten. Het is in staat de tractiekracht te registreren met bijbehorende datum en tijd en deze data kan naar een computer overgezet worden. Daarnaast is vastgesteld dat er geen significante risico's zijn en het systeem dus veilig is om mee te werken.

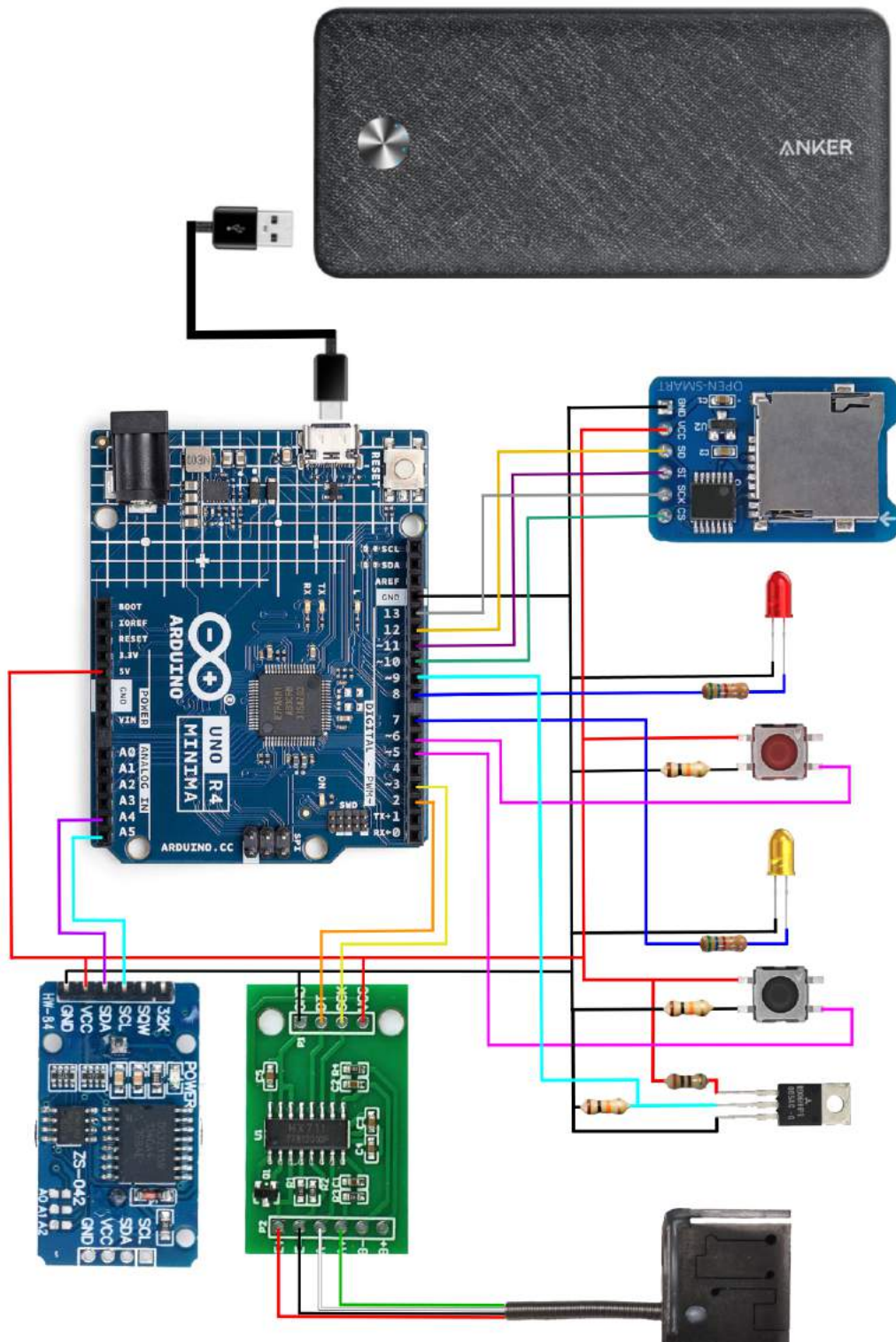
Hiermee is de basis van het systeem gelegd en lijkt het waarschijnlijk dat het systeem in de toekomst effectief kan worden ingezet voor onderzoek naar veer-gebaseerde HGT.

8 Referenties

1. Park DK, Braaksma B, Hammerberg KW en Sturm P. The efficacy of preoperative halo-gravity traction in pediatric spinal deformity the effect of traction duration. *J. Spinal Disord. Tech.* 2013 May; 26:146–54. DOI: [10.1097/bsd.0b013e318237828c](https://doi.org/10.1097/bsd.0b013e318237828c)
2. Yang C, Wang H, Zheng Z, Zhang Z, Wang J, Liu H, Kim YJ en Cho S. Halo-gravity traction in the treatment of severe spinal deformity: a systematic review and meta-analysis. *Eur. Spine J.* 2017 Jul; 26:1810–6. DOI: [10.1007/s00586-016-4848-y](https://doi.org/10.1007/s00586-016-4848-y)
3. Bruin B. Design and Implementation of a Spring-based Halo-Gravity Traction System for Complex Pediatric Spinal Deformity. 2021
4. Yu H, Kim E en Garg S. Development of a spring-based weight system for halo gravity traction for complex pediatric spinal deformity. *Spine Deform.* 2020 Oct; 8:879–84. DOI: [10.1007/s43390-020-00117-1](https://doi.org/10.1007/s43390-020-00117-1)
5. The Vertebral Column - Joints - Vertebrae - Vertebral Structure. [Online; accessed 3. Jan. 2024]. 2024 Jan. Available from: <https://teachmeanatomy.info/back/bones/vertebral-column>
6. Devereaux MW. Anatomy and Examination of the Spine. *Neurol. Clin.* 2007 May; 25:331–51. DOI: [10.1016/j.ncl.2007.02.003](https://doi.org/10.1016/j.ncl.2007.02.003)
7. Scoliosis – Symptoms, Diagnosis and Treatment. [Online; accessed 3. Jan. 2024]. 2024 Jan. Available from: <https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Scoliosis>
8. Boachie-Adjei O en Lonner B. SPINAL DEFORMITY. *Pediatr. Clin. North Am.* 1996 Aug; 43:883–97. DOI: [10.1016/S0031-3955\(05\)70440-5](https://doi.org/10.1016/S0031-3955(05)70440-5)
9. Abrisham SMJ, Ardekani MRS en Mzarch MAB. Evaluation of the Normal Range of Thoracic Kyphosis and Lumbar Lordosis Angles Using EOS Imaging. *Mædica* 2020 Mar; 15:87. DOI: [10.26574/maedica.2020.15.1.87](https://doi.org/10.26574/maedica.2020.15.1.87)
10. Spinal Deformity. [Online; accessed 3. Jan. 2024]. 2022 Oct. Available from: <https://www.yalemedicine.org/conditions/spinal-deformity>
11. Wang Y, Wang D, Zhang G, Ma B, Ma Y, Yang Y, Xing S, Kang X en Gao B. Effects of spinal deformities on lung development in children: a review. *J. Orthop. Surg. Res.* 2023 Mar; 18:246. DOI: [10.1186/s13018-023-03665-0](https://doi.org/10.1186/s13018-023-03665-0)
12. Ames CP, Scheer JK, Lafage V, Smith JS, Bess S, Berven SH, Mundis GM, Sethi RK, Deinlein DA, Coe JD, Hey LA en Daubs MD. Adult Spinal Deformity: Epidemiology, Health Impact, Evaluation, and Management. *Spine Deformity* 2016 Jul; 4:310–22. DOI: [10.1016/j.jspd.2015.12.009](https://doi.org/10.1016/j.jspd.2015.12.009)
13. Jia Y, Peng Z, Qin Y en Wang G. Surgical versus Nonsurgical Treatment for Adult Spinal Deformity: A Systematic Review and Meta-Analysis. *World Neurosurgery* 2022 Mar; 159:1–11. DOI: [10.1016/j.wneu.2021.12.017](https://doi.org/10.1016/j.wneu.2021.12.017)
14. Liu D, Yang J, Sui W, Deng Y, Li F, Yang J en Huang Z. Efficacy of Halo-Gravity Traction in the Perioperative Treatment of Severe Scoliosis and Kyphosis: A Comparison of Adolescent and Adult Patients. *World Neurosurgery* 2022 Oct; 166:e70–e76. DOI: [10.1016/j.wneu.2022.06.087](https://doi.org/10.1016/j.wneu.2022.06.087)
15. Halo-Gravity Traction | Boston Children’s Hospital. [Online; accessed 3. Jan. 2024]. 2024 Jan. Available from: <https://www.childrenshospital.org/treatments/halo-gravity-traction>
16. S610 Tension Compression Mini Load Cell. Strain Measurement Devices. Available from: <https://www.smdsensors.com/wp-content/uploads/2019/12/S610-Miniature-Load-Cell-sensor.pdf>
17. Arduino® UNO R4 Minima, Product Reference Manual. Arduino®. 2024. Available from: <https://docs.arduino.cc/resources/datasheets/ABX00080-datasheet.pdf>
18. Arduino with Load Cell and HX711 Amplifier (Digital Scale) | Random Nerd Tutorials. [Online; accessed 25. Jan. 2024]. 2022 Sep. Available from: <https://randomnerdtutorials.com/arduino-load-cell-hx711>
19. Instructables. Arduino Scale With 5kg Load Cell and HX711 Amplifier. *Instructables* 2020 Oct. Available from: <https://www.instructables.com/Arduino-Scale-With-5kg-Load-Cell-and-HX711-Amplifi>

20. 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales. AVIA semiconductor. Available from: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
21. Battery charging circuit of DS3231 module. [Online; accessed 7. Jan. 2024]. 2024 Jan. Available from: <https://www.onetransistor.eu/2019/07/zs042-ds3231-battery-charging-circuit.html>
22. Datasheet DS3231, Extremely Accurate I2C Integrated RTC/TCXO/Crystal. Maxim Integrated Products, Inc. 2015. Available from: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf>
23. Thin Film Strain Gauge Sensor Handling. Strain Measurement Devices. Available from: <https://www.smdsensors.com/wp-content/uploads/2019/08/Load-Cell-Handling-instructions.pdf>
24. Instructables. USB External Battery Packs on Arduino Turns OFF! Instructables 2017 Oct. Available from: <https://www.instructables.com/USB-External-Battery-Packs-on-Arduino-turns-OFF>
25. Anjalee JAL, Rutter V en Samaranayake NR. Application of failure mode and effects analysis (FMEA) to improve medication safety in the dispensing process - a study at a teaching hospital, Sri Lanka. BMC Public Health 2021 Jul; 21:1430. DOI: [10.1186/s12889-021-11369-5](https://doi.org/10.1186/s12889-021-11369-5)
26. Ullah E, Baig MM, GholamHosseini H en Lu J. Failure mode and effect analysis (FMEA) to identify and mitigate failures in a hospital rapid response system (RRS). Heliyon 2022 Feb; 8. DOI: [10.1016/j.heliyon.2022.e08944](https://doi.org/10.1016/j.heliyon.2022.e08944)

Bijlage A. Aansluiting hardware



Figuur 9: De aansluiting van de hardware componenten.

Bijlage B. Faalmodus en effectanalyse

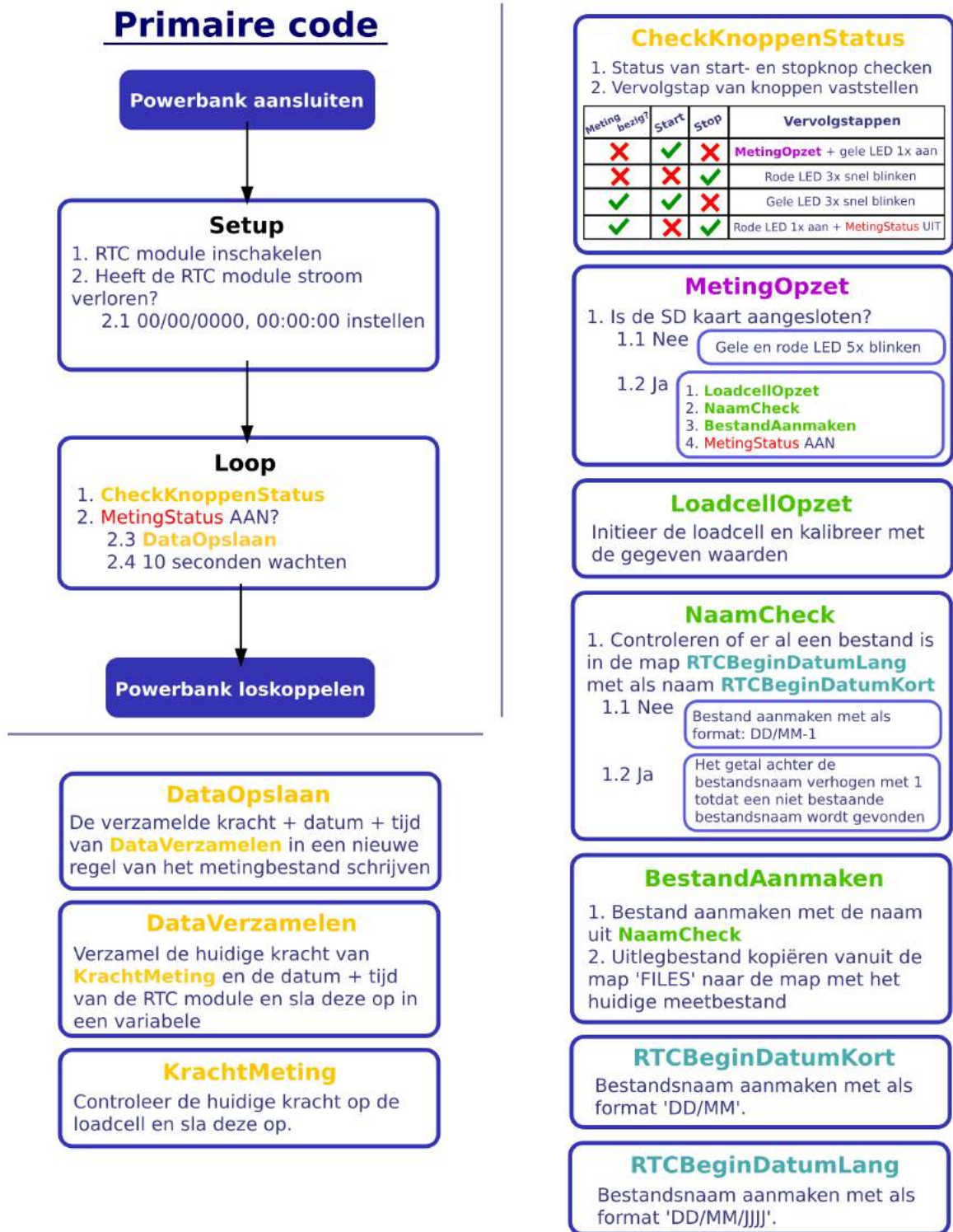
Tabel 5: FMEA deel 1

Onderdeel	Faalmodus	Effecten	S	Oorzaken	P	D	Ontwerp overwegingen	RPN
Knoppen	Startknop werkt niet	1. Defect meetsysteem	3	<ul style="list-style-type: none"> Slijtage Mechanische overbelasting Beschadigde jumper kabels 	1	1	<ul style="list-style-type: none"> Goede kwaliteit knop gebruiken Aansluiting jumperkabels zekeren 	3
	Stopknop werkt niet	1. Defect meetsysteem	3	<ul style="list-style-type: none"> Slijtage Mechanische overbelasting Beschadigde jumper kabels 	1	1	<ul style="list-style-type: none"> Goede kwaliteit knop gebruiken Aansluiting jumperkabels zekeren 	3
LED's	Gele LED werkt niet	1. Onvoldoende feedback	2	<ul style="list-style-type: none"> Schade door impact Defect door veroudering Beschadigde jumper kabels 	1	1	<ul style="list-style-type: none"> Vervangbare LED gebruiken Goede kwaliteit LED gebruiken Aansluiting jumperkabels zekeren 	2
	Rode LED werkt niet	1. Onvoldoende feedback	2	<ul style="list-style-type: none"> Schade door impact Defect door veroudering Beschadigde jumper kabels 	1	1	<ul style="list-style-type: none"> Vervangbare LED gebruiken Goede kwaliteit LED gebruiken Aansluiting jumperkabels zekeren 	2
Loadcell	Loadcell verbinding met tractiekoord beschadigd	1. Defect meetsysteem	3	<ul style="list-style-type: none"> Knoop glijdt van connectie met loadcell af Schade door impact 	1	1	<ul style="list-style-type: none"> Contactoppervlak tussen loadcell en koord verhogen Verbinding tussen koord en loadcell beschermen met krimpkous 	3
		2. Tractiekracht verdwijnt korte tijd en komt daarna weer terug	2		1	1		2
	Kabel van loadcell naar meetversterker raakt defect	1. Defect meetsysteem	3	<ul style="list-style-type: none"> Kabel blijft ergens achter hangen Materiaalmoetheid na veel gebruik 	1	2	<ul style="list-style-type: none"> Oprolkatrol gebruiken om loshangende kabel te minimaliseren Geleidendstukken toevoegen waar kabel doorheen loopt 	6
Loadcell beschadigd	2. Patiënt krijgt stroomschok	3	1		1	3		
		1. Defect meetsysteem	3	<ul style="list-style-type: none"> Schade door impact 	1	2		6

Tabel 6: FMEA deel 2

		2. Onbetrouwbare meetresultaten	2	1	3	6
				<ul style="list-style-type: none"> • Tractiekracht te hoog geworden • Vocht bij de loadcell gekomen 		<ul style="list-style-type: none"> • Opening tussen loadcell en omhulsel bedekken met zacht materiaal • Slipsysteem toevoegen welke garandeert dat de maximale belasting niet wordt overschreden
Powerbank	Oververhitting	1. Defect meetsysteem	3	1	1	3
		2. Brandgevaar	3	1	1	3
	Kortsluiting	1. Defect meetsysteem	3	1	1	3
		2. Brandgevaar	3	1	1	3
Omhulsel	Klittenband verbinding met frame beschadigd	1. Omhulsel valt van frame	2	1	1	2
		2. Kabel van loadcell wordt kapot	3	1	1	3
	Omhulsel beschadigd	1. Vergrootte kans op waterschade	2	1	1	2
		2. Vergrootte kans op systeem schade	2	1	1	2
Overige hardware componenten	HX711 werkt niet	1. Onbetrouwbare meetresultaten	2	1	1	2
		2. Defect meetsysteem	3	1	1	3
	SD kaart module werkt niet	1. Defect meetsysteem	3	1	1	3
		RTC module werkt niet	3	1	1	3

Bijlage C. Schema code



Figuur 10: Schematische weergave van de code.

Bijlage D. Code

```
1 #include "Arduino.h"
3 #include <Wire.h>
#include <RTClib.h>
5 #include <SPI.h>
#include <SD.h>
7 #include "HX711.h"

9 // Modules
RTC_DS3231 rtc;
11 HX711 scale;

13 // Bestanden
File MetingBestand;
15 File UitlegOrigineel;
File UitlegKopie;
17

// Variabelen
19 char DatumTijdKracht[30];
char BeginDatumKort[20];
21 char BeginDatumLang[20];
char ExtraBestandNaam[20];
23 char HuidigeKracht[20];
char Kracht[20];
25

// SD kaart module bedrading
27 const int chipSelect = 10;

29 // HX711 versterker bedrading
const int LOADCELL_DOUT_PIN = 2;
31 const int LOADCELL_SCK_PIN = 3;

33 //Powerbank puls bedrading
const int POWERBANK_PULS_PIN = 9;
35

// Kalibratiefactoren loadcell
37 #define kalibratie_factor -96000
#define nul_factor -195000
39

// Knoppen en LED's
41 static const int buttonPin1 = 5; // Pin voor startknop
static const int buttonPin2 = 6; // Pin voor stopknop
43 static const int LEDpin1 = 7;
static const int LEDpin2 = 8;
45

bool MetingStatusAAN = false; // Houdt bij of de vorige invoer
de startknop was
47 bool MetingStatusUIT = false; // Houdt bij of de vorige invoer
de stopknop was

49 int StartKnopStatus = 0;
int lastButtonState1 = 0;
51

int StopKnopStatus = 0;
53 int lastButtonState2 = 0;

55 // Aanpasbare variabelen
static unsigned long Meettijd = 10000;
57 static unsigned long PulsMoment = 60000;

59
void setup() {
61 pinMode(13, OUTPUT);
digitalWrite(13, LOW); // Zet de Arduino onboar LED uit voor
stroombesparing
63

if (rtc.lostPower()) {
65 rtc.adjust(DateTime(0, 0, 0, 0, 0, 0)); // Verander de datum naar 00/00/00
00:00:00 als de RTC module stroom heeft verloren
}

67 pinMode(buttonPin1, INPUT);
69 pinMode(buttonPin2, INPUT);
pinMode(LEDpin1, OUTPUT);
```

```

71 pinMode(LEDpin2, OUTPUT);
   pinMode(POWERBANK_PULS_PIN, OUTPUT);
73 }

75
void MetingOpzet() {
77   if (!SD.begin(chipSelect)) {
       digitalWrite(LEDpin1, HIGH);
79       digitalWrite(LEDpin2, HIGH);
       delay(200);
81       digitalWrite(LEDpin1, LOW);
       digitalWrite(LEDpin2, LOW);
83       delay(200);
       digitalWrite(LEDpin1, HIGH);
85       digitalWrite(LEDpin2, HIGH);
       delay(200);
87       digitalWrite(LEDpin1, LOW);
       digitalWrite(LEDpin2, LOW);
89       delay(200);
       digitalWrite(LEDpin1, HIGH);
91       digitalWrite(LEDpin2, HIGH);
       delay(200);
93       digitalWrite(LEDpin1, LOW);
       digitalWrite(LEDpin2, LOW);
95       delay(200);
       digitalWrite(LEDpin1, HIGH);
97       digitalWrite(LEDpin2, HIGH);
       delay(200);
99       digitalWrite(LEDpin1, LOW);
       digitalWrite(LEDpin2, LOW);
101      delay(200);
       digitalWrite(LEDpin1, HIGH);
103      digitalWrite(LEDpin2, HIGH);
       delay(200);
105      digitalWrite(LEDpin1, LOW);
       digitalWrite(LEDpin2, LOW);
107      return;
   }
109   else {
       NaamCheck();
111       BestandAanmaken();
       LoadcellOpzet();
113
       MetingStatusAAN = true;
115       MetingStatusUIT = false;

117       digitalWrite(LEDpin1, HIGH);
       delay(3000);
119       digitalWrite(LEDpin1, LOW);
   }
121 }

123
125 char* RTCBeginDatumKort() {
       DateTime now = rtc.now();
127       snprintf(BeginDatumKort, sizeof(BeginDatumKort), "%02d%02d", now.day(), now.month
       ());
       return BeginDatumKort;
129 }

131
char* RTCBeginDatumLang() {
133   DateTime now = rtc.now();
       snprintf(BeginDatumLang, sizeof(BeginDatumLang), "%02d%02d%02d", now.day(), now.
       month(), now.year() % 100);
135   return BeginDatumLang;
   }
137

139 char* NaamCheck() {
       strcpy(BeginDatumKort, RTCBeginDatumKort());
141       strcpy(BeginDatumLang, RTCBeginDatumLang());

143   int ExtraFileNumber = 1;

```

```

145 // Controleer of het meetbestand al bestaat
146 while (SD.exists(String(BeginDatumLang) + "/" + String(BeginDatumKort) + "-" +
147 String(ExtraFileNumber) + ".txt")) {
148     ExtraFileNumber++;
149 }
150 // Creer de naam voor het meetbestand
151 snprintf(ExtraBestandNaam, sizeof(ExtraBestandNaam), "%s/%s-%d.txt",
152     BeginDatumLang, BeginDatumKort, ExtraFileNumber);
153 return ExtraBestandNaam;
154 }
155
156 void BestandAanmaken() {
157     RTCBeginDatumLang();
158
159     SD.mkdir(BeginDatumLang);
160
161     if (!SD.exists(ExtraBestandNaam)) {
162         MetingBestand = SD.open(ExtraBestandNaam, FILE_WRITE);
163         MetingBestand.println("Datum,Tijd,Gewicht");
164         if (MetingBestand) {
165             MetingBestand.close();
166         }
167     }
168
169     if (SD.exists("FILES/UITLEGO.txt")) {
170         SD.remove(String(BeginDatumLang) + "/UITLEG.txt");
171         UitlegOrigineel = SD.open("FILES/UITLEGO.txt", FILE_READ);
172         UitlegKopie = SD.open(String(BeginDatumLang) + "/UITLEG.txt", FILE_WRITE);
173
174         while (UitlegOrigineel.available()) {
175             UitlegKopie.write(UitlegOrigineel.read());
176         }
177
178         UitlegOrigineel.close();
179         UitlegKopie.close();
180     }
181 }
182
183
184 void LoadcellOpzet() {
185     scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
186     scale.set_scale(kalibratie_factor);
187     scale.set_offset(nul_factor);
188 }
189
190 char* KrachtMeting() {
191     float krachtValue = scale.get_units();
192
193     dtostrf(krachtValue, 6, 1, HuidigeKracht);
194     return HuidigeKracht;
195 }
196
197
198 char* DataVerzamelen() {
199     DateTime now = rtc.now();
200
201     KrachtMeting();
202
203     char tempKracht[20]; // Tijdelijke buffer
204     strcpy(tempKracht, HuidigeKracht);
205
206     snprintf(DatumTijdKracht, sizeof(DatumTijdKracht), "%02d-%02d-%04d,%02d:%02d:%02d
207     ,%s", now.day(), now.month(), now.year(), now.hour(), now.minute(), now.second
208     (), tempKracht);
209
210     return DatumTijdKracht;
211 }
212
213 void DataOpslaan() {
214     strcpy(DatumTijdKracht, DataVerzamelen());
215
216     MetingBestand = SD.open(ExtraBestandNaam, FILE_WRITE);

```



```

217 |
218 |     if (MetingBestand) {
219 |         MetingBestand.println(DatumTijdKracht);
220 |         MetingBestand.close();
221 |     }
222 | }
223 |
224 |
225 | void CheckKnoppenStatus() {
226 |     // Lees de digitale status van de start- en stopknop
227 |     int StartKnopStatus = digitalRead(buttonPin1);
228 |     int StopKnopStatus = digitalRead(buttonPin2);
229 |
230 |     lastButtonState1 = StartKnopStatus;
231 |     lastButtonState2 = StopKnopStatus;
232 |
233 |     delay(20);
234 |
235 |     // Knoppen menu
236 |     if (MetingStatusAAN == false && MetingStatusUIT == false && StartKnopStatus ==
237 |         HIGH) { // Systeem is net opnieuw opgestart en START knop wordt
238 |         ingedrukt
239 |         MetingOpzet();
240 |     }
241 |     else if (MetingStatusAAN == false && MetingStatusUIT == false && StopKnopStatus
242 |         == HIGH) { // Systeem is net opnieuw opgestart en STOP knop wordt
243 |         ingedrukt
244 |         digitalWrite(LEDpin2, HIGH);
245 |         delay(200);
246 |         digitalWrite(LEDpin2, LOW);
247 |         delay(200);
248 |         digitalWrite(LEDpin2, HIGH);
249 |         delay(200);
250 |         digitalWrite(LEDpin2, LOW);
251 |         delay(200);
252 |         digitalWrite(LEDpin2, HIGH);
253 |         delay(200);
254 |         digitalWrite(LEDpin2, LOW);
255 |         delay(200);
256 |         digitalWrite(LEDpin2, HIGH);
257 |         delay(200);
258 |         digitalWrite(LEDpin2, LOW);
259 |
260 |         MetingStatusAAN = false;
261 |         MetingStatusUIT = true;
262 |     }
263 |     else if (MetingStatusAAN == true && MetingStatusUIT == false && StartKnopStatus
264 |         == HIGH) { // Er is een meting bezig en de START knop wordt ingedrukt
265 |         digitalWrite(LEDpin1, HIGH);
266 |         delay(200);
267 |         digitalWrite(LEDpin1, LOW);
268 |         delay(200);
269 |         digitalWrite(LEDpin1, HIGH);
270 |         delay(200);
271 |         digitalWrite(LEDpin1, LOW);
272 |         delay(200);
273 |         digitalWrite(LEDpin1, HIGH);
274 |         delay(200);
275 |         digitalWrite(LEDpin1, LOW);
276 |         delay(200);
277 |         digitalWrite(LEDpin1, HIGH);
278 |         delay(200);
279 |         digitalWrite(LEDpin1, LOW);
280 |         delay(200);
281 |         digitalWrite(LEDpin1, HIGH);
282 |         delay(200);
283 |         digitalWrite(LEDpin1, LOW);
284 |
285 |         MetingStatusAAN = true;
286 |         MetingStatusUIT = false;
287 |     }

```

```

else if (MetingStatusAAN == true && MetingStatusUIT == false && StopKnopStatus ==
HIGH) { // Er is een meting bezig en de STOP knop wordt ingedrukt
289 digitalWrite(LEDpin2, HIGH);
delay(3000);
291 digitalWrite(LEDpin2, LOW);
MetingStatusAAN = false;
293 MetingStatusUIT = true;
}
295 else if (MetingStatusUIT == true && MetingStatusAAN == false && StopKnopStatus ==
HIGH) { // Er is geen meting bezig en de STOP knop wordt ingedrukt
digitalWrite(LEDpin2, HIGH);
297 delay(200);
digitalWrite(LEDpin2, LOW);
299 delay(200);
digitalWrite(LEDpin2, HIGH);
301 delay(200);
digitalWrite(LEDpin2, LOW);
303 delay(200);
digitalWrite(LEDpin2, HIGH);
305 delay(200);
digitalWrite(LEDpin2, LOW);
307 delay(200);
digitalWrite(LEDpin2, HIGH);
309 delay(200);
digitalWrite(LEDpin2, LOW);
311 delay(200);
digitalWrite(LEDpin2, HIGH);
313 delay(200);
digitalWrite(LEDpin2, LOW);
315
MetingStatusAAN = false;
317 MetingStatusUIT = true;
}
319 else if (MetingStatusUIT == true && MetingStatusAAN == false && StartKnopStatus
== HIGH) { // Er is geen meting bezig en de START knop wordt ingedrukt
MetingOpzet();
321 }
}
323
void loop() {
325 CheckKnoppenStatus();
327 static unsigned long PulsTijd = 0;
PulsTijd = millis();
329
if (millis() - PulsTijd >= PulsMoment) {
331 digitalWrite(POWERBANK_PULS_PIN, HIGH);
delay(100);
333 digitalWrite(POWERBANK_PULS_PIN, LOW);
PulsTijd = millis();
335 }
337 static unsigned long lastRunTime = 0;
static bool isRunning = false;
339
if (MetingStatusAAN && !MetingStatusUIT && !isRunning) {
341 isRunning = true;
lastRunTime = millis();
343 }
345 if (isRunning && (millis() - lastRunTime >= MeetTijd)) {
DataOpslaan();
347 lastRunTime = millis();
}
349
if (MetingStatusUIT && !MetingStatusAAN) {
351 isRunning = false;
}
353 }

```

Code 1: Volledige Arduino IDE code