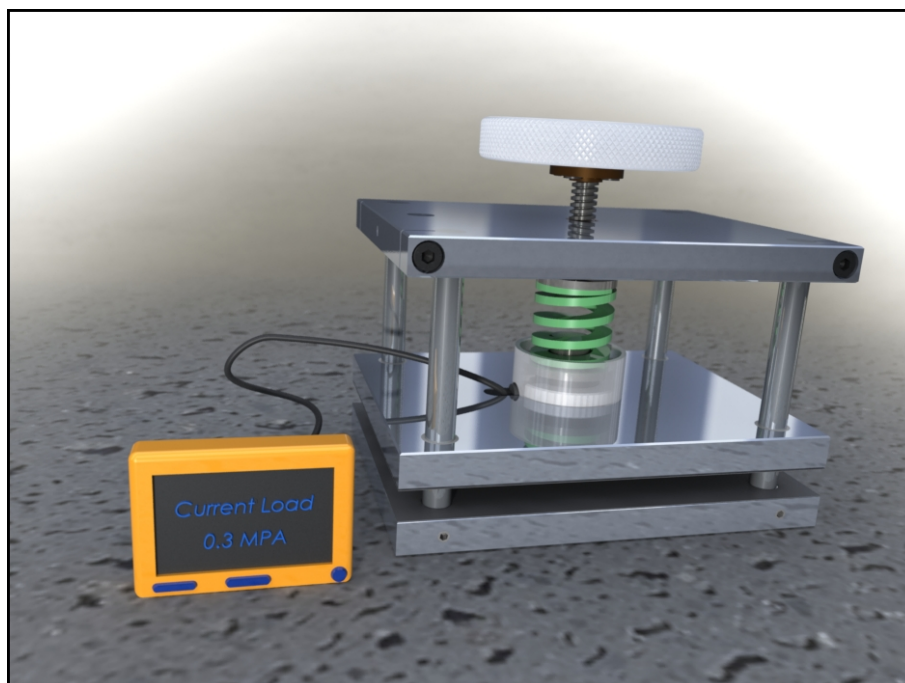


Henrik Wiik Berg  
Martine Vik Byrkjeland  
Peter Hermansen Ulvestad  
Eivind Melby

## Rigg for test av battericeller

Produktutvikling av rigg for å påføre trykk på  
battericeller under testing

Bacheloroppgave i Ingeniørfag, Maskin  
Veileder: Rolf Alexander Skar  
Mai 2023





Henrik Wiik Berg  
Martine Vik Byrkjeland  
Peter Hermansen Ulvestad  
Eivind Melby

## **Rigg for test av battericeller**

Produktutvikling av rigg for å påføre trykk på  
battericeller under testing

Bacheloroppgave i Ingeniørfag, Maskin  
Veileder: Rolf Alexander Skar  
Mai 2023

Norges teknisk-naturvitenskapelige universitet  
Fakultet for ingeniørvitenskap  
Institutt for maskinteknikk og produksjon



Kunnskap for en bedre verden





Kunnskap for en bedre verden

## Rigg for test av battericeller

*Produktutvikling av rigg for å påføre trykk på  
battericeller under testing*

Henrik Wiik Berg  
Martine Vik Byrkjeland  
Peter Hermansen Ulvestad  
Eivind Melby

Gradering: Åpen

Bachelor i ingeniørfag: Maskin  
Innlevert: Mai 2023  
Veileder: Rolf Alexander Skar

Norges teknisk-naturvitenskapelige universitet  
Institutt for vareproduksjon og byggingsteknikk

Oppgavens tittel:	Dato: 22.05.2023		
Produktutvikling av rigg for mekanisk test av battericeller	Antall sider:		
	Masteroppgave	Bacheloroppgave	X
Navn: Henrik Wiik Berg, Eivind Melby, Peter Hermansen Ulvestad og Martine Vik Byrkjeland			
Veileder: Rolf Alexander Skar			
Eventuelle eksterne faglige kontakter/ veiledere: Asbjørn Ulvestad (IFE)			

### Sammendrag:

**Hensikten** var å utvikle en justerbar rigg for statisk påføring av trykk på battericeller, hvor man skulle kunne påføre en gitt mekanisk kraft og lese av trykket som en battericelle utsettes for i sanntid. Oppgaven artet seg som ett produktutviklingsløp, hvor det etter IFE sine spesifikasjoner, skulle designes og produseres en fullverdig prototype av en rigg for test av battericeller. Styrende faktorer ville i stor grad omhandle volumbegrensning, kostnad, funksjon og brukervennlighet.

**Metode** ble tolket som et standard produktutviklingsløp. I forsknings-sammenheng har det ikke blitt foretatt markedsanalyse verken før eller etter oppgavens start. Dette lå utenfor oppgavens rammer, og med det faktum at det leveres ett produkt etter spesifikasjoner gitt fra en enkelt oppdragsgiver, hvor hovedfokuset lå i utviklingsløpet fra ide til prototype. Det ble tatt utgangspunkt i de store spørsmålene for å ligge grunnlag for en god metode, nemlig hvorfor og hvordan.

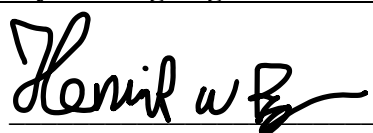
**Hovedfokus** i oppgaven tok for seg det å lage en rigid og funksjonell rigg som er i henhold til IFE sine krav. Under hele prosessen skulle det være sterkt fokus på evne til omstilling dersom det oppstod en utfordring, samt kreativitet og optimalisering for både produkt og prosess. Det ble lagt sterkt vekt på dokumentasjon i form av vedlegg for å kunne produsere riggen basert på underlaget i denne rapporten, det skulle også foreligge monteringsmanual og en god beskrivelse for bruk og feilsøking.

**Resultat** ble den ferdige prototypen som ble produsert på rapportens dokumentasjonsunderlag og imøtekom samtlige krav satt fra oppdragsgiver relatert til riggens funksjon, nøyaktighet og brukervennlighet. Riggens kalibrert og funksjoner demonstrert ovenfor oppdragsgiver. Det var derimot ingen komplett prototype uten drøfting og forbedringsforslag, dette fremkommer godt gjennom rapportens drøfting og endelige besvarelse av problemstilling. Rapporten fremstår som godt dokumentert ved at den gir produksjonsunderlag og detaljert beskrivelse ned på komponentnivå, samtidig som den ivaretar dokumentasjon knyttet til riggen sin funksjon som et produkt.

**Konklusjonen** er at riggen som ble produsert basert på dokumentasjonsunderlaget i denne rapporten imøtekom IFE sine kravspesifikasjoner med tanke på både funksjon og brukervennlighet. Det foreligger mer diskusjon på detaljnivå, som omhandler forbedringspotensial på flere komponenter i konstruksjonen, men som har vist seg å ikke være kritiske men veiledende for en mer komplett rigg. Det gjenstår arbeid for mer eksakte kostnader knyttet til masseproduksjon og langtidstesting med faktiske celler, men dette ligger ikke innenfor rammene for denne rapporten.

Stikkord:


Brukermanual
Bærekraft
CAD modellering
Design
Innovasjon
Produksjon
Produksjonsunderlag
Produktutvikling
Programmering
Prototype
Styrkeberegning



Henrik Wiik Berg



Peter Hermansen Ulvestad



Eivind Melby



Martine Vik Byrkjeland

Microsoft teams 22.05.2023

# Forord

Rapporten har som hensikt å dokumentere et produktutviklingsløp hvor det er utviklet en rigg for testing av battericeller på oppdrag for Institutt for energiteknikk, representert ved forsker Asbjørn Ulvestad.

Gruppen av medforfattere ble etablert ved studiestart høst 2019, og har siden fungert som en gruppe ved oppgaver og innleveringer der deltakerne har tatt samme kurs. Høsten 2022 kontaktet Peter Ulvestad sin bror Asbjørn, ansatt ved IFE, for å forhøre seg om Asbjørns arbeidsgiver kunne ha en interessant oppgave. Gruppen ble da tildelt ideen om å utvikle en testrigg for battericeller, som ble raskt godtatt i gruppen som oppgave for bachelor.

Rapporten går gjennom hele produktutviklingsløpet for å utvikle en mekanisk rigg for testing av battericeller i samsvar med IFE sine kravspesifikasjoner. Den er begrenset til selve utviklingen og produksjon av prototype, hvor markedsføring og fullskala produksjon ikke ligger innenfor rammene til denne rapporten. Det nevnes bare innledningsvis.

Alle gruppemedlemmer har deltatt aktivt i ide og designfasen. Peter har utarbeidet det endelige designet, Henrik har utarbeidet tegninger og stått for produksjon av prototype, Eivind har koblet elektronikk og programmert Arduino, mens Martine har jobbet med materialvalg og bærekrafts aspekter knyttet mot rigg for celledtester. Alle utgifter knyttet til produktutviklingen er 100% selvfinansiert av gruppemedlemmene.

Gruppen gir en stor takk til IFE for ett godt samarbeid og for å ha gitt nøyaktige kravspesifikasjoner, som har hjulpet til ved å sitte tydelige rammer for prosjektet i en tidlig fase. Gruppen gir også veileder ved NTNU Rolf Alexander Skar en stor takk for sine innspill og for å føre arbeidet i riktig retning med god oppfølging under hele prosessen. Øvrig vil gruppen også takke til gruppemedlemmers familie for deres tålmodighet, og kolleger for deres faglige hjelp og faglige innspill underveis.

Denne rapporten avslutter herved med dette et 4-årig utdanningsløp for gruppen bestående av fire nettstudenter ved NTNU Gjøvik.



# Abstract

The **purpose** of this thesis was to develop an adjustable rig for applying static pressure onto battery cells, where one should be able to get real time reading of the force applied to the cell at any given time. This thesis ranges through the process of product development, where the team will design and produce a prototype of a rig for battery cell testing according to IFE's specifications. The governing factors will mostly consist of factors such as volume limitation, cost, functionality, and user friendliness.

The **method** will appear as a standard line of product development. In the context of the rig being developed for research purposes, a market analysis has not been conducted during the process. This is because it's being delivered a product based on a single customer's specifications, where the focus is based on the process of an idea to a prototype. The report is based on the method of the simple questions such as why and how.

The **main focus** of the thesis is to describe the process of developing and production of both solid and functional rig according to IFE's specifications. During the whole process there has been a strict focus on re-design when needed, creativity and process/product optimisation. The report's documentation, in the form of attachments, will have a strong focus on making the reader to be able to reproduce an identical product by using the report.

The **result** is a complete prototype, which is produced on this thesis documentation and analysis. A prototype that is in line with the specifications, such as function, accuracy, and ease of use. On the other hand, it is not a complete prototype without the discussion of improvement proposals which is occurring frequently in this report. The report stands are a well-documented product development process where verifiability and production description at component level, at the same time as the report account for the rig as a product itself.

The **conclusion** is that the prototype produced has met the superior specifications from IFE and is during tests showing satisfactory results both on functionality and user friendliness. There is an own chapter where the components and functionality are more discussed at component level, this gives a clue to potential improvements, but non improvements that are critical for the rig to be operative pr now. It is some work left before an eventual mass production related to calculate the costs and long-term testing of the rig, but this is not in the scope for this thesis and must be done on a later stage.

# Innholdsfortegnelse

Forord.....	iv
Abstract.....	v
Innholdsfortegnelse.....	vi
1 Innledning.....	1
1.1 Bakgrunn.....	1
1.2 Problemstilling.....	2
1.3 Mål med oppgaven.....	2
1.4 Tilnærming og begrensning.....	3
1.5 Representanter.....	4
2 Teori.....	6
2.1 Spesifikasjoner fra IFE.....	6
2.1.1 Kravspesifikasjoner fra oppdragsgiver.....	6
2.2 Bærekraft.....	10
2.3 Økonomi.....	11
2.3.1 Overordnet kostnadsbilde.....	11
2.3.2 Utvikling og produksjonskostnad.....	12
3 Metode.....	14
3.1 Designprosessen.....	14
3.2 Beregninger.....	19
3.2.1 Resulterende krefter i konstruksjon.....	20
3.2.2 Spenninger og moment for bevegelse skrue.....	22
3.2.3 Beregning av pasning for bunnplate og glidebolter.....	24
3.2.4 Beregning av nødvendig temperatur i bunnplate for montering av glidebolter.....	28
3.2.5 Utbøying av topp-plate som følge av totalkraft.....	30
3.2.6 Moment for låsing av glidebolter i topp-plate.....	31
3.3 Komponenter.....	33
3.3.1 Bunnplate (1).....	34
3.3.2 Midtplate (2).....	35
3.3.3 Topplate (3).....	35
3.3.4 Materialvalg for plater (1,2 og 3).....	35

3.3.5	Rod/Guidebolter (4) .....	36
3.3.6	Deksel/rør (5) .....	36
3.3.7	Lastcelle (6).....	37
3.3.8	Fjær (7).....	37
3.3.9	Spring guide topp, bunn og spacer (8,13,16) .....	38
3.3.10	TR8x2, Mutter og skrue (10,12).....	38
3.3.11	Bearing, Iglidur (14).....	39
3.3.12	Hand Wheel (15).....	39
3.3.13	Arduino Mega med komponenter (19,20,21).....	39
3.3.14	Termisk matte/isolator (17).....	40
3.3.15	Øvrige komponenter og skruer (9,11,18).....	41
3.4	Produksjon/montering mekanisk .....	41
3.4.1	Deler som inngikk i produksjon.....	41
3.4.2	Maskinering av plater .....	42
3.4.3	Maskinering av guidebolter og fjærholdere .....	44
3.4.4	3D print av øvrige maskinelementer .....	45
3.4.5	Montering prototype.....	46
3.5	Produksjon trykkovervåking.....	48
3.5.1	Trykkovervåking .....	48
3.5.2	Elektroniske komponenter.....	52
3.5.3	Pseudokode og kildekode.....	55
3.5.4	Usikkerhet ved målinger .....	56
3.6	Beskyttelse av elektronisk utstyr .....	57
3.6.1	Boks for Arduino.....	57
3.6.2	Boks for ADC.....	58
3.6.3	3D-printing av bokser for Arduino og ADC .....	59
3.7	HMS.....	60
4	Resultater.....	61
4.1	Endelig løsning .....	61
4.2	Testing .....	63
4.2.1	Testing av lastceller.....	63
4.2.2	Mekanisk testing.....	68
5	Diskusjon.....	70

5.1	Innkjøpskostnad.....	70
5.1.1	Direkte kostnader .....	70
5.1.2	Indirekte kostnader .....	71
5.2	Beregninger.....	72
5.3	Innhenting og logging av data .....	73
5.4	Mulige forbedringsområder .....	74
6	Konklusjon .....	78
	Litteraturliste .....	81
	Vedlegg .....	85

# Figurliste

Figur 1: Organisasjonskart .....	4
Figur 2: Skissert celle .....	6
Figur 3: Kompakt design .....	14
Figur 4: Negativ fjær design .....	15
Figur 5 : Plunger design .....	15
Figur 6 : Tannhjul design .....	16
Figur 7 : Gir design .....	16
Figur 8: Sammenstilling 1 .....	17
Figur 9: Sammenstilling 3 .....	18
Figur 10: Sammenstilling 2 .....	18
Figur 11: Resulterende reaksjonskraft mellomplate (1 MPa).....	20
Figur 12: Spenning presspasning .....	26
Figur 13: Trykk presspasning.....	26
Figur 14: Spenning presspasning 2 .....	27
Figur 15: Trykk presspasning.....	27
Figur 16: Varmeutvidelse elementmetode 1 .....	29
Figur 17: Varmeutvidelse elementmetode 2 .....	29
Figur 18: Håndberegning.....	30
Figur 19: Utbøying elementmetoden.....	31
Figur 20: Innfestning elementmetoden 1.....	32
Figur 21: Innfestning elementmetode 2.....	32
Figur 22: Oversikt over komponenter som inngår direkte i sammenstillin.....	33
Figur 23: Solidworks-modell av bunnplate.....	34
Figur 24: Solidworks-modell av midtplate.....	35
Figur 25: Solidworksmodell av topplate .....	35
Figur 26: Guidebolter .....	36
Figur 27: Deksel/rør .....	36
Figur 28: Innmat fjær .....	38
Figur 29: TR8X2 Mutter og skrue.....	38
Figur 30: Iglidur bearing .....	39
Figur 31: Håndratt .....	39
Figur 32: Termisk matte i grafitt .....	40
Figur 33: G-kode for test-skjæring og gjennomført test i MDF.....	42
Figur 34: Slicer for aluminium plater.....	42
Figur 35: Ferdig maskinerte plater .....	43
Figur 36: Manuell Dreieing.....	44
Figur 37: Slicer for SLS .....	45
Figur 38: Slicer for FFD.....	46
Figur 39: Identifisering av slaver/sensorer ved hjelp av adresse (Campbell, u.å.).....	49
Figur 40: Kobling av SDA og SCL til ekstern strømkilde (Campbell, u.å.).....	50
Figur 41: PINOUT for Raspberry Pi 4 (Raspberry Pi Foundation. u.å.).....	51

Figur 42: PINOUT for Arduino Uno (The Robotics Back-End, u.å.).....	51
Figur 43: 1,8’’ TFT Display Breakout and Shield fra Adafruit (Adafruit, u.å.a.).....	53
Figur 44: DS1307 Real Time Clock fra Adafruit (Adafruit u.å.b.).....	54
Figur 45: Pinout Arduino Mega (Arduino.cc u.å.b).....	54
Figur 46: Load Cell Amplifier – HX711 fra Sparkfun.com (Sparkfun.com, u.å.).....	55
Figur 47: Grafisk oversikt over lineariteten for lastcellen fra TE Connectivity.....	56
Figur 48: Arduino Mega med shield ferdig modellert .....	58
Figur 49: Ferdig modellert boks tilpasset Arduino Mega og shield.....	58
Figur 50: 3D modell av ADC kortet.....	58
Figur 51: Snitt som viser hvordan delene er festet sammen.....	59
Figur 52: Ferdig sammensatt boks .....	59
Figur 53: Topp av boks .....	59
Figur 54: Bunn av boks .....	59
Figur 55: ADC-boks.....	59
Figur 56: Ferdig design og prototype.....	61
Figur 57: Ferdig montert testjigg.....	68
Figur 58: Klassediagram .....	85
Figur 59: Koblings skjema.....	86
Figur 60: Testing av lastcelle fra TE Connectivity .....	86
Figur 61: Test av lastcelle fra Sparkfun .....	87
Figur 62: Dimensjoner for lastcellen fra TE Connectivity (TE Connectivity, 2020).....	140
Figur 63: Dimensjoner for lastcellen kjøpt fra Sparkfun (HTC-Sensor.com, u.å.),.....	140
Figur 64: U-bolt med popnagle .....	140
Figur 65: Oppsett av lastcelle test .....	142

## Tabelliste

Tabell 1: Estimert kostnadsbilde .....	12
Tabell 2: Inngangsdata for bevegelseskrue .....	22
Tabell 3: Inngangsdata for beregning av pasninger .....	24
Tabell 4: Inngangsdata for beregning av press-montering.....	28
Tabell 5: Oversikt over komponenter.....	34
Tabell 6: Spesifikasjoner for FX29 og TAS606.....	37
Tabell 7: Oversikt over ulike Arduino modeller (Arduino.cc, u.å.b).....	52
Tabell 8: Oversikt over minste og største avvik og i tillegg hvor mye avvik det er i kg. ....	64
Tabell 9: Resultatene viser avvik ved hvert forsøk og det største avviket i %.....	65
Tabell 10: Test for å finne lavest kalibreringsvekt.....	66
Tabell 11: Nullfaktorens påvirkning på resultatet.....	67
Tabell 12: Kalibreringsfaktorens påvirkning på resultatet .....	67
Tabell 13: Innkjøpskost materiell.....	71
Tabell 14: Oversikt over måleinstrumenter for vekt. ....	141
Tabell 15: Oversikt over alle vektskiver og lignende som er brukt .....	141

# Begrepsliste

Begrep	Forklaring
3D	3 Dimensjonal
ADC	Analog to digital converter
AM	Additive manufacturing- Additiv tilvirkning, 3D print etc.
ATC	Automatic tool changer
CAD	Computer Aided Design
CAM	Computer aided manufacturing -Data assistert produksjon
CNC	Computer Numerical Control – datastyrt bevegelse i maskiner
CP	Contact pressure – Kontakt trykk
ERC	Europeisk forskningsråd
FEA/FEM	Finite element analysis –Numerisk dataassistert regnemetode
G-KODE	Programmeringsspråk brukt for å kontrollere CNC-maskiner
HIG	Høyskolen I Gjøvik, Senere NTNU
HMS	Helse Miljø og sikkerhet
IFE	Institutt for energiteknikk
MDF	Medium densityfiberboard, finerplatet med middels tetthet
MESH	Strukturell oppbygning av 3D model bestående av polygoner
NFR	Norsk forskningsråd
NTNU	Norges teknisk-naturvitenskapelige universitet
PA GFR	Polyamid Glass reinforced polyester
Posecelle	«Pouch cell» på engelsk
RTC	Real time clock
SI/C	Silisiumkarbid, Materiale I battericeller
Slicer	Programvare for å konvertere 3D objektfiler
SOLIDWORKS	CAD software
STL	Stereolitografi Filformat for 3D-modeller





# 1 Innledning

## 1.1 Bakgrunn

Formålet med oppgaven var å gjennomgå et produktutviklingsløp på oppdrag fra Institutt for energiteknikk. Oppdraget gikk ut på å utvikle og produsere en rigg for mekanisk testing av battericeller i henhold til IFE sine spesifikasjoner. De underliggende formål var å gjennomgå et komplett produktutviklingsløp, hvor fasene ide, design, analyse, prototype, produksjon, produkt og konklusjon skal fremlegges i denne rapporten.

Celletesteren har som hensikt å komprimere battericeller over lengre tid. Det skal være en konstant kraft tilført batteri-cellens overflate, dette med muligheter for å tilføre forskjellige kraftpåkjenninger og overvåke disse digitalt. Den skulle også ha mulighet for å teste flere celler i samme syklus ved behov. Rigger skal brukes for å undersøke mekanismene i SI/C batteri-celler som er påført en konstant kraft over flere sykluser i perioder på 3-12 måneder. Fokuset skal spesifikt dreie seg om hva som skjer i forhold til delaminering og reduksjon av kapasitet over tid ved forskjellige kraftpåkjenninger.

En av flaskehalsene til moderne batterier er knyttet til at ekspansjon av battericeller fører til en irreversibel partikkel til partikkel brytning, delaminering og partikkelbrudd, noe som igjen fører til redusert kapasitet og økt mekanisk degradering. (Müller, *et al.*, 2019)

For å imøtekomme markedets stadig voksende etterspørsel etter bærekraftige batterier med lenger levetid og stabil effekt over tid, er forskning og testing av battericeller avgjørende for å skulle imøtekomme markedets behov. Om riggen for celletesting viser seg å være vellykket vil dette kunne bidra til det grønne-skiftet, samt at det kan forsterke Norges posisjon som en energiaktør i det internasjonale markedet.

Utover battericellens betydning for samfunn og funksjon, vil oppgaven avgrenses til produktutvikling hvor IFE vil være videre ansvarlig for testing etter at rigg er overlevert.

Gruppen er innforstått med kravspesifikasjoner og forventninger knyttet til rigg for celletesting, og har utelukkende arbeidet for å levere et produkt som tilfredsstill

oppdragsgivers kravspesifikasjoner. Dette ble utført hvor overordnede mål var å ta stilling til bærekraft, reproduserbarhet, brukervennlighet, lav kostnad og funksjonalitet.

## 1.2 Problemstilling

Etter en utbredt gjennomgang av dagens marked for tilsvarende produkter og en sammenligning med IFE sine gitte kravspesifikasjoner og ønsker, er det konkludert med 4 aktuelle delproblemstillinger:

1. Hvordan er det mulig å utvikle en rigg for celletesting som imøtekommer kravspesifikasjoner, som samtidig har et økonomisk fortrinn sammenlignet andre rigger som har tilsvarende kapabiliteter.
2. Hva skal man prioritere mest med tanke på produktets kvalitet, innovasjon og bærekraft når det dukker opp problemstillinger underveis i prosessen?
3. Hvor vil problemer og forbedringsområder hovedsakelig ligge i en prototype?
4. Hvordan skal denne riggen bli et førstevalg i forskningssammenheng på celler?

Når de 4 delproblemstillingene ble summert, så ble det enighet om en hovedproblemstilling: Hvordan utvikle og produsere en rigg for celletesting som imøtekommer kravspesifikasjoner fra IFE, samtidig som den ikke er for komplisert for produsent og forbruker?

## 1.3 Mål med oppgaven

Hovedmål med oppgaven var å:

1. Designe en prototype rigg for testing av battericeller, som skal ha funksjon i henhold til IFE sine kravspesifikasjoner, også for fremtidige forsøk.
2. Kontinuerlig analysere og optimalisere design ved bruk elementmetoden, håndberegninger og brukeropplevelser.

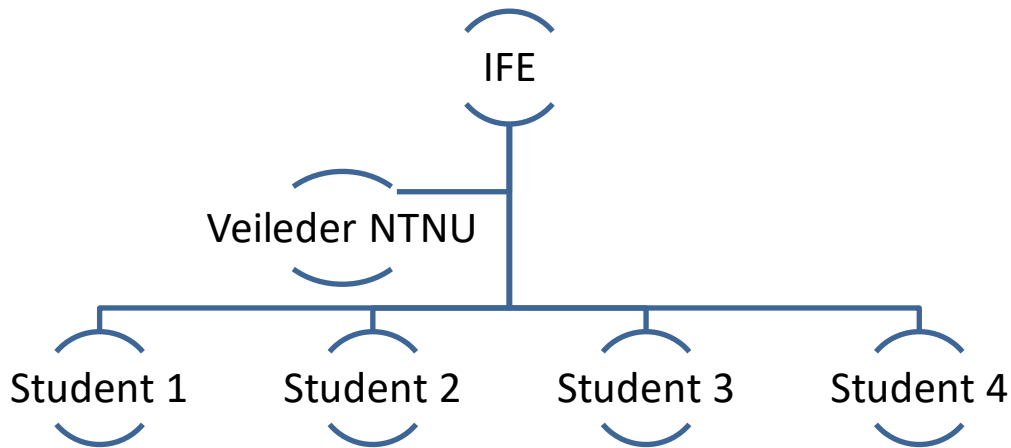
3. Kontinuerlig utfordre kostnad og balansere dette opp mot bærekraft, funksjonalitet og pålitelighet.
4. Dokumentere produktutviklingsløp hvor produksjonsoptimalisering og dokumentasjon for å produsere og montere en identisk test rigg for celler er vektlagt høyt.
5. Utvikle et brukervennlig produkt som enkelt kan monteres og feil-søkes på.
6. Unngå et for unikt og spesialtilpasset design, for å være innovative, og tenke langsiktig med tanke på eventuelle bruksområder. Samtidig skal produktet være standardisert for oppgaven det er tiltenkt.

## **1.4 Tilnærming og begrensning**

Tilnærmingen var basert på studentenes utdanningsløp og IFE sine forventninger. Erfaring og kunnskap på tvers i gruppen var viktig for en god og fornuftig tilnærming. Rapporten ble utarbeidet i henhold til tilgjengelig teknisk underlag. Oppgavens kompleksitet ble holdt innenfor utdanningens rammer, samtidig som gruppen aktivt oppsøkte kunnskap utenfor læreplan for å øke rapportens estetiske og teoretiske kvalitet. Oppgaven hadde en stor faglig og teknisk bredde hvor det underveis måtte sattes begrensninger for holde tidsplan, og det at oppgavens avsluttende rapport skulle imøtekomme IFE sine kravspesifikasjoner på en fornuftig og hensiktsmessig måte.

Begrensningene ble lagt til et produktutviklingsløp hvor hovedansvaret var fordelt på konsept, design, produksjon og funksjonstest av ferdig prototype. Utover dette var det en felles forståelse for hva produktet skal brukes til og at de overordnede målene i avsnitt 1.3 ble overholdt, samtidig som IFE sine kravspesifikasjoner ble imøtekommet. Det var også begrensninger for utarbeidelse av en markedsanalyse, hvor det i denne rapporten ble utført spesifikt på oppdrag for IFE. Det ville da på et senere tidspunkt eventuelt bli foretatt en grundig markedsanalyse, om produktet viser seg å fungere tilfredsstillende i forhold til tiltenkt funksjonen.

## 1.5 Representanter



*Figur 1: Organisasjonskart*

Organisasjonskartet presenterer representerte involverte i oppgaven, der IFE fungerte som en linjemyndighet og veileder hadde en rolle som rådgivende myndighet. Øvrig hadde studentene i oppgaven en flat struktur seg imellom.

Studentene har gjennom store deler av studieløpet fungert som en gruppe ved flere kurs. Det ble dermed et naturlig valg for studentene å skrive denne oppgaven sammen, uavhengig av at det fremstod som kontroversielt med 4 studenter på en enkelt bacheloroppgave ifølge fakultetets retningslinjer.

Gruppen består av nettstudenter med stor variasjon i geolokalitet. Arbeidsfordeling, planlegging og utførelse ble gjort i samsvar med forprosjektplan. Eventuelle endringer underveis ble tatt opp i plenum, hvor alle studentene deltok i avgjørelser som hadde direkte og indirekte innvirkning på oppgavens omfang og resultat.

### **IFE:**

Institutt for energiteknikk har 720 ansatte fordelt på 35 land med en omsetting på omtrent 1.3 milliarder årlig. De jobber blant annet med å utvikle lønnsom, sikker og miljøvennlig teknologi innen fornybar energi, petroleumsutvinning CO<sub>2</sub>-håndtering (CCS), kreftmedisin, nukleær medisin, materialforskning og opprettholdelse med utvikling av nasjonal nukleær

kompetanse rettet mot dekommisjonering av reaktorer. (IFE,2023) IFE sin representant inn mot dette prosjektet/oppgaven er forsker Asbjørn Ulvestad.

### **Veileder NTNU:**

Norges tekniske-naturvitenskapelige universitet, representert av førsteamanuensis for vareproduksjon og byggeteknikk Rolf Alexander Skar. Han er utdannet sivilingeniør innen elektrokjemi og doktorgrad innen uorganisk kjemi. Han har arbeidserfaring fra aluminiumsindustri i ved Husnes og videre som førsteamanuensis på HiG før og etter HiG ble innlemmet i NTNU.

### **Student 1**

Henrik Wiik Berg, født: 29.09.1992. Nettstudent ved NTNU Gjøvik som maskiningeniør. Har 12 års erfaring fra petroleumsindustrien som mekaniker på roterende maskineri og i senere tid som vedlikeholds-ingeniør, spesifikt mot roterende maskineri ved Equinor sitt raffineri på Mongstad. Primær interesse: Roterende maskineri, CAD, AM og maskinering.

### **Student 2**

Peter Ulvestad, født: 03.03.1986. Nettstudent ved NTNU Gjøvik som maskiningeniør. Er tidligere utdannet matematikk- og naturfagslærer. Deleier i et lite firma som har laget skreddersydde produkter for bruk i forskning, hvor hovedoppgaver har vært CAD/CAM og produksjon på en selvlaget CNC-fres.

### **Student 3**

Eivind Melby, født 24.08.1979. Nettstudent ved NTNU Gjøvik som maskiningeniør. Utdannet trafikklærer og arbeider som avdelingsleder i Wright Trafikkskole AS. Programmeringserfaring fra tidligere studier som dataingeniør.

### **Student 4**

Martine Byrkjeland, født 07.10.1994. Nettstudent ved NTNU Gjøvik som maskiningeniør. Arbeidserfaring fra Elkems smelteverk med prosessanlegg som knuser og fraksjonerer metall, som videre blir solgt til kunder over hele verden.

# 2 Teori

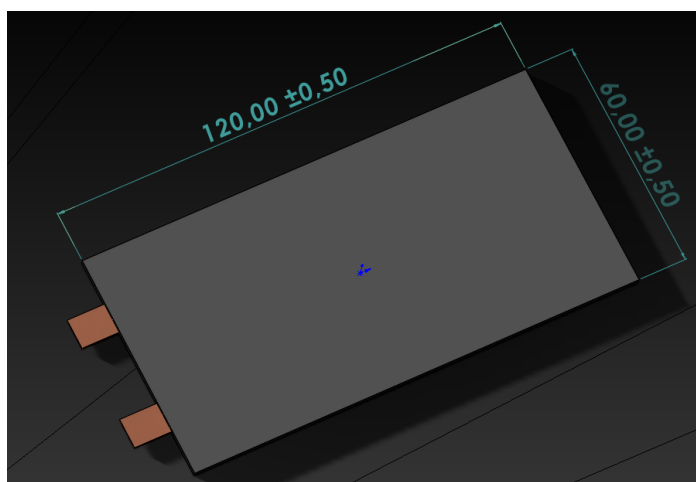
## 2.1 Spesifikasjoner fra IFE

I møte med IFE, representert av Asbjørn Ulvestad den 22.02.2023, ble det fremlagt og diskutert kravspesifikasjoner i forbindelse med rigg som skulle utvikles. Møtet var hovedsakelig for å diskutere spesifikasjoner i detalj og fremlegge skisserte modeller for IFE for en elimineringsprosess hvor det ble gitt tydelige indikasjoner på hvilket design som skulle arbeides videre med. Kravspesifikasjoner gitt i møte 22.02.2023 kommer frem i avsnitt 2.1.1

### 2.1.1 Kravspesifikasjoner fra oppdragsgiver

#### Design

Det skal være en konstant kraftpåkjenning på en rektangulær battericelle, som festes mellom to plater i et tidsrom mellom 3-12 måneder. Det må tas høyde for at antall battericeller og at kraftpåkjenning kan økes ved et senere tidspunkt, uten at dette medfører store og kostbare endringer på test-rigg. Cellene som rigg skal påføre kraftpåkjenning har en overflate på 120mmx60mm og tykkelse lik 1.8 mm. Det er problematikk knyttet til denne typen test jigger i dag, da det stort sett er spesialtilpassede design som er på markedet, som gjerne ikke er standardisert og fullskala-testet til akkurat dette formålet ifølge IFE.



Figur 2: Skissert celle

Det er ett ønske at riggens volum er lavt, da det er begrenset plass i test-skap hvor testene skal foregå. Det er også behov for å kontinuerlig overvåke om batteriet utvider seg, og dette kan gjøres med ulike sensorer. Det vil også være fordelaktig å plassere avleser for overvåkning av trykk og forskyvning på utside av skap, da det er kabelgjennomganger i test-skap. Ett annet viktig kriterium er også at tamper for batteri-celle skal være lett tilgjengelig og på ingen måte skal kunne komme i kontakt med selve riggen. Det kan være flere forskjellige metoder for tilkobling for tamper, og dette må tas høyde for i designprosessen.

Det må i tillegg være brukervennlig og ha en lav vekt. Det skal være mulig for bruker selv å gjennomføre enkel feilsøking, samt reparasjoner eller utbytting av komponenter. Det må medfølge brukermanual på norsk og engelsk, og i tillegg en manual for hvordan man logger data.

IFE ønsker også mulighet for å skru rigg fast i hyller i test-skap, for da å kunne operere med eventuelle glideskuffer som øker effektivisering av arbeidet med test av batteri-cellene.

IFE ønsker også at test-jiggen lages slik at varme ledes vekk fra batteriet, og at det samtidig isoleres mot kortslutning av batteripoler mot hverandre eller jord.

Det må vurderes hvorvidt testtrigg skal leveres ferdig montert eller som et monteringssett, på grunn av spørsmål knyttet til kostnad og testtriggens kompleksitet. Det er et krav til å kunne produsere rigg basert på rapportens dokumentasjon. Det vil også være en fordel at produsent også foretar montering, da det ved IFE sine institusjoner vil det være mange forskjellige personer involvert underveis i tester av celler.

Sikkerhetsfaktor oppgis ikke av IFE, men i designfasen vil det tas høyde for 1,5 x von Mises som sikkerhetsfaktor innledningsvis, hvor dette kan justeres under designprosessen.

### **Kraftoverføring**

Spenningene som batteriene skal utsettes for i utgangspunktet inklusivt 1,5 x von Mises er mindre enn 0,3MPa, men det skal i design tas høyde for en spenning opp mot 1 MPa. Med en battericelles areal lik  $7200\text{mm}^2$  tilsvarer det en kraft og vekt på henholdsvis 2160 N og maksimalt 7200 N, og en vekt tilsvarende 220 Kg ved førstegangstest og 734 Kg som maksvekt. Kraften cellen utsettes for skal være konstant til tross av cellens kan ekspandere og kontrahere igjennom mange sykluser.

Lokaler som rigg skal brukes i har ikke tilstrekkelig ventilasjon for å bruke gass som arbeidsmedium. Dette kan i tillegg lede til problematikk i form av hyppigere etterfylling og overvåking av rigg. Temperatur vil også variere noe som kan gi stor uforutsigbarhet med gass. Gass har også ekstra sikkerhetsmessige og praktiske utfordringer som risikoer knyttet til personell eksponert for gass, samt faren for lekkasjer.

Væske også uaktuelt som følge av sikkerhetsmessige og rent praktiske årsaker, da væsker kan ha en høy elektrisk ledningsevne. Det kan også gi høy uforutsigbarhet som arbeidsmedium og er en veldig fordyrende faktor i designet, med tanke på komponenter og volumetriske restriksjoner.

En applikasjon hvor væsker og gass kan egne seg er ved gjentatte oppladninger og utladninger batterier som går over noen få dager, på fagspråket kalt «forming». En prosess hvor hurtighet ved battericellebytte vektlegges og trykk ikke vil tape seg i stor grad på grunn av det korte tidsrommet hver battericelle utsettes for trykk. Men dette inngår ikke rammene for denne oppgaven, men nevnes av IFE som en mulig applikasjon for noen av designene som innledningsvis ble skissert av gruppen.

Foretrukket kraftoverføringsmetodikk vil være en ren statisk mekanisk overføring, med bruk av fjærer for å sikre konstant trykk på batteri-cellen. Tanken er å sette en kraftpåkjenning jevnt fordelt på battericellens overflate med hjelp av forspenningskraft og en stabiliserende fjærkraft. Fjærer kan gi en ekstra usikkerhet i avlesning av kraft. IFE ønsker en enkel oppspenning av celle i rigg uten behov for hyppig justering, dette er for å ikke binde personell opp mot celletester i den grad av at de må justere rigger ofte.

I dette designet sitter IFE en minimumsgrense til levetid for applikasjon til 5 år, noe som må tas høyde for med tanke materialvalg og robusthet.

## **Materialer**

Materialer som inngår i batteri-celle tester, skal primært oppfylle 3 kriterier

1. Det skal være lav vekt og unngå stål og tykt gods.
2. Det skal være lav-kost.
3. Det skal lede varme vekk fra batteri-celle effektivt.



Rigg vil bli jordet, så det er ikke nødvendig å ta hensyn til antistatiske materialer, sett vekk fra matter som skal inngå i direkte kontakt med batteri-celle. Denne matten må være av et materiale som leder varme samtidig som det ikke leder strøm.

Det må også av sikkerhetsmessige årsaker unngå materialer og smøremidler med høyt innhold av fluorforbindelser, da batteri-celle kan utvikle varme, som igjen kan få materialer til å avgi hydrogenfluorid gass. Dette kan være meget helseskadelig.

Det må tas hensyn til hvordan ulike materialer agerer med hverandre i riggen i forhold til spenningsrekken for materialer, det må også tas hensyn til ledningsevne for både elektrisitet og varme. Test-skapet som riggene skal monteres i har en temperatur variasjon på -15 til 100 grader, hvor IFE opplyser at testene vil i all hovedsak utføres mellom 15-30 grader. Maksimal temperatur vil være opp mot 60-70 grader.

## **Data**

For å overvåke trykkendringer og eventuell utvidelse av battericellene, kan det benyttes både en sensor for å måle trykket og en avstandssensor for å måle utvidelsen. Man kan eventuelt bruke trykksensor for begge måltyper, for å kutte kostnads og spare plass. For å finne eventuell utvidelse må man vite fjærkonstanten for fjæra.

Data fra sensorene må minimum kunne lastes inn på et SD kort, men det er svært fordelaktig å kunne laste ned data via WiFi. IFE ser på det som fordelaktig å kunne overvåke test-celler utenfra test-skapet, dette er fordelaktig for volum og av praktiske årsaker. Dette blir igjen et spørsmål om kostnad.

Data må konverteres til ønskelig format/enhet for å kunne tolkes og det er høyt krav til nøyaktige toleranser og riktig data.

En eventuelt burde også produseres deksel til styreenheten, for å unngå smuss og minimere fare for brukerfeil.

Utvidelsessensor kan kobles til skapet og trenger ikke gå innom styringsenheten.

## **Kostnad**

IFE ser for seg kostnader på mellom 1000kr og 5000kr per rigg ferdig montert. Det vil være avgjørende videre da IFE anslår noen 10-talls til 200 testtrigger. I prisen ferdig montert skal

det følge med materialer, materialbearbeiding, sensorer, prosessor, montering og bruksanvisning.

Det må vurderes kostnader i forhold til frakt, produsent og produksjonssted i løpet av oppgaven. En kostnadsanalyse må utarbeides, samt en omtale av miljø-fotavtrykk knyttet til produktets komponenter.

## **HMS og sikkerhet**

Minimumskrav er at sikkerhetsrisikoer er markert på produkt, slik som risikoer for klemfare, brannfare også videre.

Førstegangsmontering og førstegangsbruk må kvalitet sikres. Det må utarbeides en prosedyre for dette med relevante sjekklister.

Det burde foreligge en punktleksjon over fremgangsmåte og ekstrahering av data, samt vedlagt datablad på norsk og engelsk for involverte kjemikalier og festemidler tilgjengelig til enhver tid.

## **2.2 Bærekraft**

En av de største utfordringene for moderne produksjon av nye produkter er bærekraft. Veldig mange produkter kan ikke gjenvinnes da de er produsert på en slik måte som gjør det vanskelig. Enten er det umulig å ta dem fra hverandre eller for dyrt å gjøre det. Prosessen med å designe et bærekraftig produkt starter derfor allerede under planleggings- og designfasen. Det er mange parametere som skal tas hensyn til når materialene faktisk skal være bærekraftige. Det trengs god kvalitet, lang levetid og bestanddeler som kan tas fra hverandre og gjenvinnes. Materialene skal også transporteres, og da må miljøbelastningen tas med i regnestykket.

De siste årene har sirkulær økonomi blitt en stor del av klimadebatten, og det med god grunn. Å innføre sirkulær økonomi i større grad er kanskje det viktigste tiltaket for at verden skal nå FNs bærekraftsmål innen 2030 (FN-Sambandet, 2023). Sirkulær økonomi handler om å beholde og bruke ressursene våre så lenge som mulig, slik at minst mulig ender opp som avfall. For å oppnå en sirkulær økonomi er det helt avgjørende at materialene i produktene kan gjenvinnes, og brukes på nytt etter endt levetid (Miljømerking Norge, (u.å.)). Aluminium

er et sirkulært materiale som inngår i et lukket produksjonskretsløp. Når aluminium først er produsert, kan det nemlig smeltes om uendelig mange ganger uten at det påvirker materialets egenskaper. Hele 75 prosent av alt aluminium som noen gang har blitt produsert er fortsatt i bruk (The International Aluminium Institute, 2018). Aluminium er i tillegg et svært lett materiale, noe som har stor betydning for materialets bærekraft. Fraktkostnadene reduseres betraktelig når materialenes vekt er lav, og det gjelder både for råvarer som kjøpes inn og ferdig produkt som sendes ut til kunde. Fraktselskapet som frakter materiale og ferdigvare sparer drivstoff med lettere gods, og slik reduseres klimautslipp knyttet til transporten. Det medfører at produktet også blir mer tiltalende og kostnadseffektivt for kunden.

## **2.3 Økonomi**

### **2.3.1 Overordnet kostnadsbilde**

IFE er en selveid stiftelse, men finansieres gjennom forskjellige kanaler og grunnbevilgning (<10%). IFE er i all hovedsak prosjektfinansiert. Prosjektene kan være finansiert av Norsk forskningsråd alene, et samarbeid mellom NFR og industripartner, eller direkte fra industripartner. Prosjektene kan også finansieres av det Europeisk forskningsråd og lokale organisasjoner som regionale forskningsfond.

I dette tilfellet med battericeller er det penger tildelt fra NFR for pilotlinje av batteriproduksjon som ligger til grunn. Dette fører til at riggen vil bli finansiert som laboratorie-drift, eller en del av forskningsinfrastruktur som batteriproduksjon.

Kostnadsrammen er skal være mellom 1000kr og 5000kr per rigg, med en total anskaffelseskostnadsramme på 75.000-200.000 kr, avhengig av endelig behov. Dette må IFE kartlegge under pilotlinjen for prosjektet, som et spørsmål om tilgjengelig plass til rigger og kostnad knyttet til en rigg. Det vil under piloten bli avgjort om dette er «kjekt å ha», eller viser seg å være essensielt for videre forskning. Dette kan igjen være avgjørende for eventuelle fremtidige kostnadsrammer og en total-leveranse.

IFE og NTNU er primærinteressenter i oppgaven, da produktet kommer IFE sitt prosjekt direkte til gode, imens det er flere sekundærinteressenter i form av mulige forbrukere, leverandører og andre aktører i forskningsmiljøet.

## 2.3.2 Utvikling og produksjonskostnad

Det ble satt opp ett kostnadsestimat i forprosjektet (prosjektidentifiseringsfasen), med en typisk usikkerhet på ca. 30 %, som for andre mulighets-/prosjektutviklingsstudier.

(Rolstandås, A, et al. 2014). Dette var for å kartlegge den tekniske og økonomiske gjennomførbarheten. I dette avsnittet ble de allerede fastsatte kostnadene til deler og estimerte kostnader for produksjon knyttet mot prosjektets identifiserings-fase. Alle tall ble hentet direkte fra innkjøp eller generelle produksjonsrater fra industrien. Det ble en aksept grense for usikkerhet lik 10 % av det totale kostnadsbildet før rapport overleveres til IFE for vurdering i sin pilot. Men det må da i etterkant utsendes forespørsler til faktisk potensielle produsenter for å innhente et mer reelt kostnadsbilde for en stor skala produksjon.

Det ble tatt høyde for at råemne som plater, resin og øvrige polymer skal kjøpes som en komplett enhet fra leverandør. Det ikke lagt opp for noe rest/kapp, og dette kan påvirke totalprisen. Det ble også tatt med i kalkulasjoner at det trengs 8 Arduinoer (styringsenhet) for å kunne drifte 50 testtrigge. Dette var ett av flere kostnadseffektiviserende forbedringer i design og utvikling.

Tabell 1:  
Estimert kostnadsbilde

Estimert Kostnadsbilde for Utvikling & Produksjon																	
KOSTNAD MATERIELL				ANTALL RIGGER				KOSTNAD UTVIKLING OG PRODUKSJON				ANTALL RIGGER					
NR	Beskrivelse	Innkjøp	Enhet	NOK	PR.RIGG	Enhet	1	50	NR	Beskrivelse	Antall	Enhet	NOK	PR.RIGG	1	50	
1	Lastcelle	1,00	STK	682,00	1,00	STK	682,00	34100,00	1	Design	1	H	300	100	30000	30000	
2	Iglus Foringer	4,00	STK	119,00	1,00	STK	119,00	5950,00	2	Beregning	1	H	300	40	12000	12000	
3	M4 skruer	1,00	STK	1,00	8,00	STK	8,00	400,00	3	Koding	1	H	250	100	25000	25000	
4	M3 skruer	1,00	STK	1,00	8,00	STK	8,00	400,00	4	Prototype	1	H	300	120	36000	36000	
5	Fjær	1,00	STK	23,00	1,00	STK	23,00	1150,00	5	Uttesting	1	H	150	20	3000	3000	
6	TRx2, Mutter	1,00	STK	50,00	2,00	STK	100,00	5000,00	6	Dokumentasjon	1	H	100	100	10000	10000	
7	Akslinger Ø8	0,3	m	110,00	1,33	m	146,52	7326,00	7	Frakt	1	STK	1000	0,1	100	5000	
8	Akslinger TR8x2	0,3	m	80,00	0,43	m	34,40	1720,00	8	Produksjon	1	H	400	6	2400	120000	
9	Rundbolt 316 L	1,00	m	800,00	0,04	m	32,00	1600,00	9	3D print	1	H	500	1	500	25000	
10	Alu Plate	2,88	m^2	6000,00	0,06	m^2	360,00	18000,00	10	Montering	1	H	100	2	200	10000	
11	Arduino Mega	1,00	STK	549,00	1,00	STK	549,00	3921,43	11	Administrativ kost	1	H	300	0,5	150	7500	
12	Resin	0,95	L	1070,00	0,0130	L	13,91	695,50									
13	Polymer 3D-print	0,50	Kg	623,00	0,0101	Kg	6,26	313,06									
14	Grafitt flak	1,00	STK	112,00	2,00	STK	224,00	11200,00									
15	ADC	1,00	STK	147,00	1,00	STK	147,00	1050,00									
16	RTC	1,00	STK	85,00	1,00	STK	85,00	607,14									
17	Loctite 243	1,00	STK	180	0,1	STK	18,00	900,00									
							2556,09	94333,13									
<b>3. Nøkkeltall</b>												<b>1. Kostnad materiell</b>					
		Pris NOK	Enhetskostnad			Kvantumrabatt %		0,00		0,03							
Totalkostnad 1 stk		5906,1	5906,1			Sum		2556,09		91503							
Totalkostnad 50 stk		102330,1	3840,1			Enhetspris		2556,09		1830,1							
Kostnad ink. utvikling 1 stk		121906,1	121906,1			<b>2. Produksjon og utvikling</b>		SUM		119350		283500					
Kostnad ink. utvikling 50 stk		221680,1	5232,1			Kvantumrabatt %		0		0,4							
						SUM		119350		170100							
						Enhetspris		119350		3402							
						Enhetspris eks utvikl		3350		2010							

Tabell 1 gir et grovt estimat over kostnader knyttet til innkjøp av materiell, samt en oversikt over mulig utvikling og produksjonskostnad. Det var tatt høyde for at noen emner ble kjøpt inn

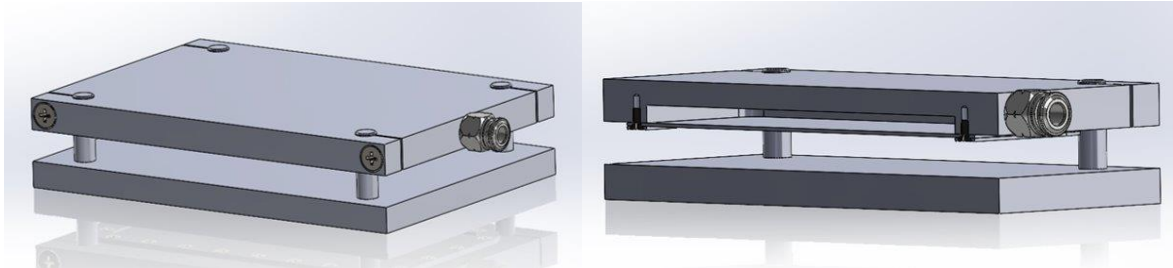
som hele emner fra leverandør, for å unngå kostnad ved kapp og bearbeiding. Man kan se et anslag av utviklingskostnad og hva som blir ansett som en logisk kvantumsrabatt, da prisene er hentet fra direktebestilling per komponent for en rigg.

# 3 Metode

## 3.1 Designprosessen

Det første stadiet i designprosessen var en idémyldring hvor alle deltagerne laget enkle skisser av forskjellige idéer på grunnlag av rammene beskrevet av oppdragsgiver. I første omgang ble arbeidet gjort individuelt for å få et utvalg med størst mulig bredde. Idéer og skisser ble så presentert for gruppen, og de forskjellige konseptene ble fordelt for videre arbeid. Underveis i arbeidet ble de forskjellige designene diskutert, og forslag til forbedringer/endringer kom ved hvert møte. Noen forslag ble forkastet i løpet av prosessen og andre fikk større eller mindre endringer.

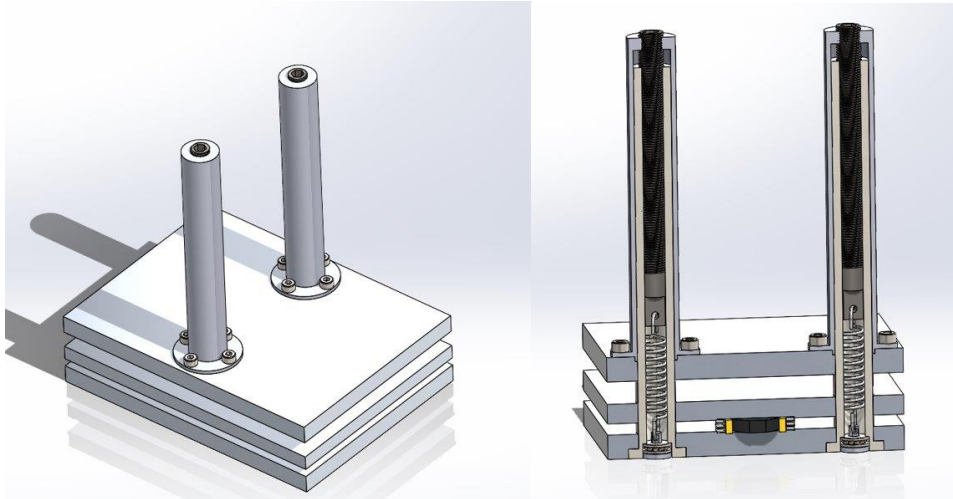
Noen av forslagene og idéene som ble arbeidet med:



*Figur 3: Kompakt design*

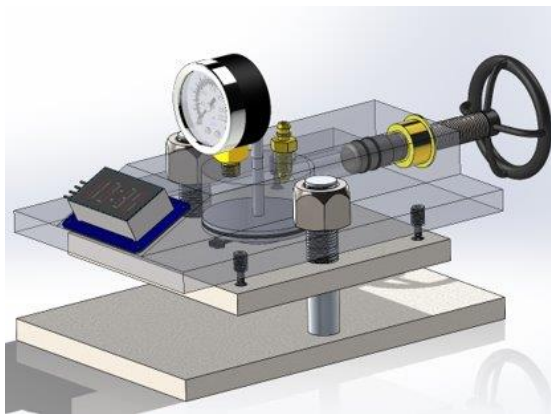
Figur 3 viser en gass- eller væskefylt blære bygget inn i topplaten. Gode egenskaper er blant annet at den er veldig kompakt og påfører jevnt trykk over hele flaten. Negative egenskaper er muligheten for lekkasje og mange slanger inn i skapet. Det brukes nitrogen under trykk i testlabben. Faren er tilknyttet lekkasjer, og det at å holde alt trykksatt i måneder av gangen gjorde at denne idéen ble forkastet.

Det ble nevnt under møte angående kravspesifikasjoner med IFE at de designene som hadde gass eller væsker som arbeidsmedium, kunne egne seg godt for en annen type applikasjon knyttet til testing celler som gikk ut på formering av celler.



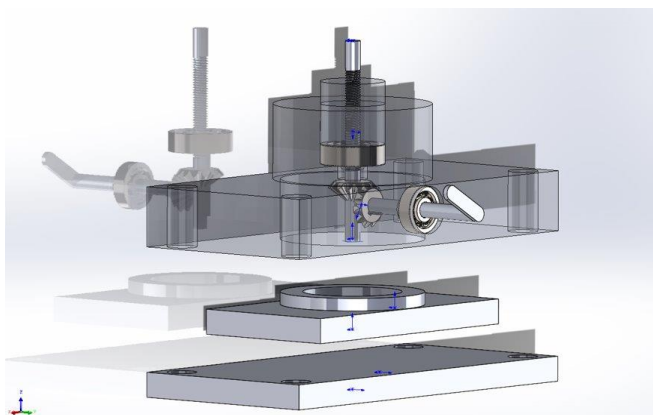
*Figur 4: Negativ fjær design*

Figur 4 viser styrestenger med innbygget trekkefjær. Dette var i hovedsak et forsøk på å lage et forslag med lange trekkefjærer for å kunne bruke noe med lav fjærkonstant, og ha mulighet for veldig store eller stabler med celler. Grunnen til at denne falt bort var at den er veldig plasskrevende, samt at det kreves mange kompliserte deler for å få den til å fungere.



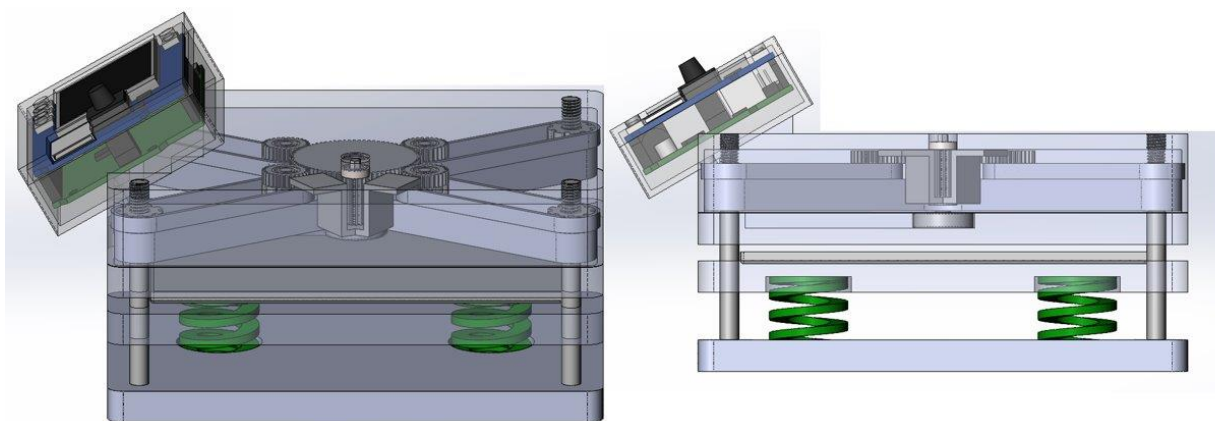
*Figur 5 : Plunger design*

Figur 5 viser et nytt forsøk på å bruke væske. En god løsning med et helt tett kammer, og mulighet for trykkjustering og overvåking direkte på testtriggen. Denne falt også bort, både på grunn av komplekse deler, og at det var ønskelig å unngå bruk av væsker inne i test-skapene.



*Figur 6 : Tannhjul design*

Figur 6 viser et forsøk på å lage en versjon som ga mulighet for å justere trykket fra siden, slik at tilgang fra toppen ikke var nødvendig. Dette ble ganske fort forkastet på grunn av unødvendig kompleksitet og begrenset plass.



*Figur 7 : Gir design*

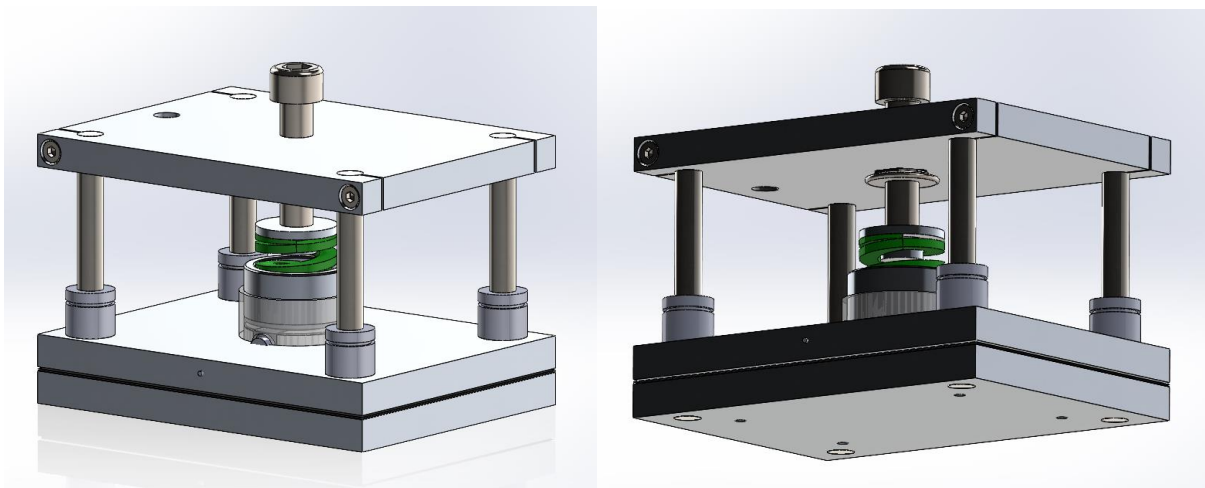
Figur 7 viser et forslag som inneholder en Arduino, som får data fra en lastcelle i midten. Det er en av måtene for trykkovertvåking som var med i vurderingen. Trykket er tenkt påført ved hjelp av gjengestenger i hjørnene, og fjærer i bunnen for å unngå trykkspikes ved eventuell utvidelse av cellen.

Avslutningen av idemyldringen var et møte med IFEs representant Asbjørn Ulvestad (22.02.2023), de forskjellige idéene ble presentert og diskutert. Det kom frem at det var ønskelig å unngå væsker og gasser, men mindre fordelene var veldig store. Samtidig var enkelhet i bruk og veldig lav sannsynlighet for feil de viktigste prioriteringene. Videre arbeid ble derfor gjort på de enkleste designidéene.



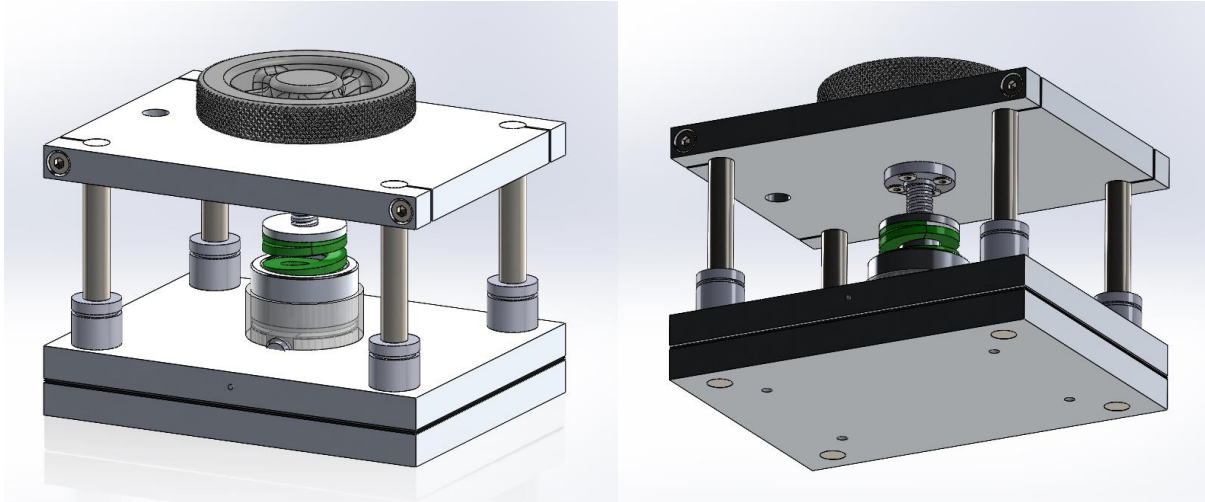
Antallet mulige design var kort tid etter møtet begrenset til ett, med en skrue som påførte trykket via en fjær. Valget på fjær kom av at det vil gi noe fleksibilitet og dermed ingen ekstreme trykkøkninger ved eventuell utvidelse av cellen. Trykket blir påført midt på battericellen for å unngå ujevn last eller trykkspikes på kantene/hjørnene på cellen. Da dette var bestemt begynte arbeidet med å bestemme hva slags lager-, skrue- og fjærtyper som skulle brukes. Det ble også arbeidet med overvåking av trykket cellene var utsatt for, og cellenes eventuelle utvidelse under testing.

Valgene som ble gjort var på grunnlag av tilgjengelighet av komponenter, som stenger og lager. Mens andre komponenter ble valgt på grunn av fleksibilitet, som standardiserte fjærer. Beregninger var også en stor del av beslutningsprosessen, de ble også brukt til å kontrollere at beslutninger gjort på annet grunnlag også var tilfredsstillende. Beregningene er beskrevet i kapittel 3.2.



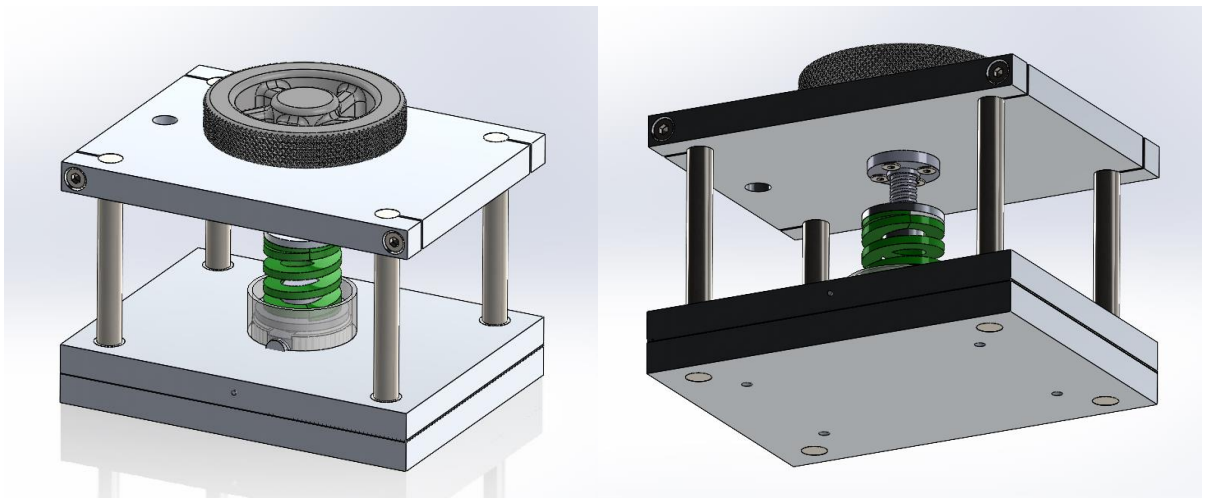
*Figur 8: Sammenstilling 1*

Figur 8 viser den første iterasjonen av det valgte designet med vanlig m10 skrue og kulelager.



*Figur 9: Sammenstilling 3*

Figur 9 viser den andre iterasjonen. Der er m10 socket head skruer og kulelager byttet til TR8x2 skruer og mutter.



*Figur 10: Sammenstilling 2*

Figur 10 viser den tredje iterasjonen der lagrene er byttet til IGUS glidelager

Det var i dette detaljarbeidet det aller meste av tiden til designarbeid ble brukt. Parallelt ble arbeidet med et system for overvåkning og logging av trykket gjort. Dette kan leses mer om i kapittel 3.5. Det ble designet et forslag med stor fleksibilitet bygget inn, og hver gang nye beslutninger ble tatt ble designet av selve riggen oppdatert, eller det ble laget en ny konfigurasjon i Solidworks.

En endelig endring kom under produksjonen, da det i utgangspunktet var tenkt en ren krympepassning mellom glidebolter og bunnplaten. Denne var for trang, og under trykk ble

aluminium fra bunnplaten skåret bort av glidebolten som er i stål. Det ble derfor besluttet pasningen skulle åpnes noe og at det skulle legges til en skrue og et spor i glidebolten for å forsikre at den forble på plass. Dette resulterte i to forskjellige løsninger for festing av glideboltene i prototypen og diskuteres i kapittel 5.4.

## 3.2 Beregninger

Beregningene ble utført i henhold til de spenningene som ble oppgitt i IFE sine spesifikasjoner for normal og maksimal spenning ved celletesting. Cellens overflateareal i planet er  $7200\text{mm}^2$ , og den normale og maksimale spenningen er angitt som  $1,5 \times$  von Mises sikkerhetsfaktor utgjør som utgjør  $0,3\text{MPa}$  og  $1\text{MPa}$ .

Formlene brukt i beregningene er hentet fra bøkene konstruksjonselementer (Hartvigsen *et al.*, 2006) og verkstedshåndboka (Dahlvig *et al.*, 1991), materiell fra maskindeler og mekanikk kurs, samt statiske elementmetode beregninger utført ved hjelp av Solidworks sin egen simulerings modul.

Utrekningene hadde som formål å avdekke følgende:

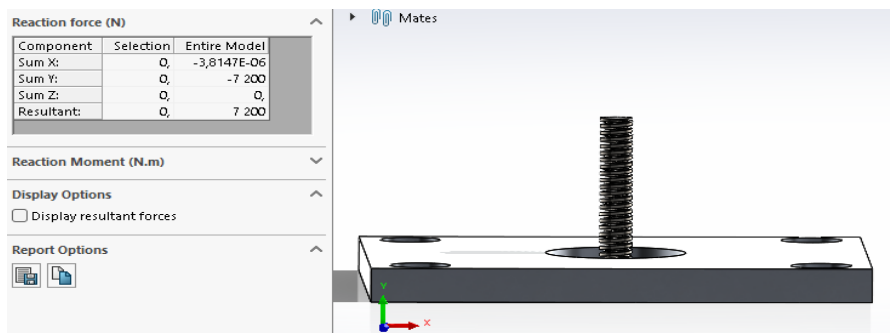
- Kraft og spenning som oppstår i konstruksjon som følge av påført moment: Elementmetoden & håndberegning
- Sikkerhetsfaktor og moment for dynamisk stillskrue: Håndberegning
- Pasninger og sikkerhetsfaktor for låsing av guidebolter til bunnplate: Håndberegning
- Nødvendig monteringsstemperatur for guidebolter til bunnplate: Håndberegning
- Nedbøyning av plate som følge av kraftpåkjenning: Elementmetoden & håndberegning
- Nødvendig moment for låsebolter i topp-plate: Elementmetoden

Det ble utført håndberegninger ved bruk av generelle formler og simuleringer, med bruk av elementmetoden for enkeltkomponenter, og forenklede sammenstillinger av komponenter som rigg.

Den mest omfattende beregningen var å beregne moment for låsebolter til topp-plate. Utregningene måtte baseres på kontaktrykk (CP) hvor materialene som inngår i forbindelsen har ulike egenskaper. Det ble konkludert med at denne typen utregning best ble løses ved bruk av elementmetoden, da håndberegning hadde gitt et for grovt resultat.

Øvrig ble det også gjort forenkling for bunnplatens varmeutvidelse for montering av guidebolter, hvor det ble tatt i beregning for en tykk sylinder som utvendig materiale. Dette ble etterprøvd med elementmetoden og det viste seg å være pålitelige tall, i samsvar med håndberegningene.

En mer strukturert og beskrivende utførelse av beregninger beskrives i resten av kapittel 3.2 sine underkapitler. Beregninger er ble utført i samarbeid med designprosess avsnitt 3.1 og ble kontinuerlig oppdatert ved designendringer.



Figur 11: Resulterende reaksjonskraft mellomplate (1 MPa)

### 3.2.1 Resulterende krefter i konstruksjon

Beregningene ligger til grunn for selve designprosessen som beskrives i kapittel 3.1:

#### Styrende kraft i konstruksjon som følge av spenning påført battericelle.

Battericellens areal ble basert på oppdragsgivers målsetting av celler

$$A = B * L$$

$$A = 120\text{mm} * 60\text{mm} = 7200\text{mm}^2$$

Design skulle forholde seg til spenningene 0,2 MPa som operativ last og mulighetsstudie for 0,67 MPa som maksimal last. I samarbeid med IFE ble det besluttet å designe og beregne for en sikkerhetsfaktor FOS = 1.5. Det ble da kalkulert ut ifra gitte verdier fra IFE følgende:

$$FOS = \frac{Q_{max}}{Q}$$

$$Q_{max} = FOS * Q_{max}$$

$$Q_{max1} = 1,5 * 0,2Mpa$$

$$Q_{max2} = 1,5 * 0,67Mpa$$

$$Q_{max1} = 0,3 Mpa \cap Q_{max2} = 1,0Mpa$$

Trykk kraften som bevegelse skrue da må overføre til mellomplate blir:

$$F = Q * A$$

$$F1 = 0,30 \frac{N}{mm^2} * 7200mm^2 = 2160N$$

$$F2 = 1,0 \frac{N}{mm^2} * 7200mm^2 = 7200N$$

Forenklet er da kraft som da igjen fordeles til per guidebolt lik:

$$Fb = \frac{F_{aksial}}{n}$$

$$Fb1 = \frac{2160N}{4} = 540N$$

$$Fb2 = \frac{7200 N}{4} = 1800N$$

### 3.2.2 Spenninger og moment for bevegelse skrue.

Bevegelse skruen som brukes er trapesgjenget med betegnelse TR8X2, med 30 graders profilvinkel og en gjengeinngang. (NS271,1943)

Tabell 2:  
Inngangsdata for bevegelseskrue

Data	Mutter	Bolt
Materialkvalitet	BS1400LG2 (Bronse)	AISI 304
D1 (lille diameter)	6,5 mm	6,2 mm
D2 (stigningsdiameter)	7,25 mm	7,18 mm
D3 (Store diameter)	8,3 mm	8 mm
u (friksjonskoeffisient)	0,173	0,173
P (stigning)	2	2

#### 1. Utregning av areal av boltens tverrsnitt

$$A_s = \left(\frac{\pi}{4}\right) * \left(\frac{d_2 + d_3}{2}\right)^2$$

$$A_s = \frac{\pi}{4} * \left(\frac{7,18\text{mm} + 8\text{mm}}{2}\right)^2 = 45,35\text{mm}^2$$

#### 2. Utregning av tiltrekkingsmoment $M_v$ for bolt

$$M_v = F * r_m * \tan(\varepsilon_1 + \varphi)$$

$$r_m = \frac{d_2}{2} = \frac{7,18\text{mm}}{2} = 3,59\text{mm}$$

$$\tan \varepsilon_1 = \tan^{-1}\left(\frac{u}{\cos 30}\right) = \tan^{-1}\left(\frac{0,173}{\cos 30}\right) = 11,29$$

$$\tan \varphi = \tan^{-1}\left(\frac{P}{\pi * d_m}\right) = \tan^{-1}\left(\frac{2}{\pi * 7,18\text{mm}}\right) = 5,06$$

$$M_{v1} = 2160\text{N} * 3,59\text{mm} * \tan(11,29 + 5,06) = 2278\text{Nmm} = 2,28\text{Nm}$$

$$M_{v2} = 7200\text{N} * 3,59\text{mm} * \tan(11,29 + 5,06) = 7582,9\text{Nmm} = 7,58\text{Nm}$$

### 3. Utregning av spenninger som oppstår i skruen ved maksimalt moment og sikkerhet.

$$Fa1 = \frac{F}{As} = \frac{2160N}{45,35mm^2} = 47,63N/mm^2$$

$$Fa2 = \frac{F}{As} = \frac{7200N}{45,35mm^2} = 158,76 \frac{N}{mm^2}$$

$$\tau_{v1} = \frac{Mv1 * d3}{\pi * d1^3} = \frac{2278Nmm * 8mm}{\pi * (6,2mm)^3} = \frac{24,34N}{mm^2}$$

$$\tau_{v2} = \frac{Mv2 * d3}{\pi * d1^3} = \frac{7200Nmm * 8mm}{\pi * (6,2mm)^3} = \frac{76,93N}{mm^2}$$

$$\sigma_{jf1} = \sqrt{\sigma_1 + 3 * \tau^3_1} = \sqrt{\frac{47,63N}{mm^2} + 3 * \left(\frac{24,34N}{mm^2}\right)^2} = 42,71 \frac{N}{mm^2}$$

$$\sigma_{jf2} = \sqrt{\sigma_2 + 3 * \tau^3_2} = \sqrt{\frac{158,76N}{mm^2} + 3 * \left(\frac{76,93N}{mm^2}\right)^2} = 133,84 \frac{N}{mm^2}$$

$$\sigma_{flyt Aisi 304} = \frac{205 N}{mm^2}$$

$$N1 = \frac{\sigma_f}{\sigma_{jf1}} = \frac{205 \frac{N}{mm^2}}{42,71 \frac{N}{mm^2}} = 4,8$$

$$N2 = \frac{\sigma_f}{\sigma_{jf2}} = \frac{205 \frac{N}{mm^2}}{133,84 \frac{N}{mm^2}} = 1,53$$

Utregningene gir et overslag over det som kan forventes å måtte påføre i form av moment for å oppnå ønsket spenning fordelt på cellens overflate, samtidig som det er god margin mot flyt i materialet. Merk at bolt vil utsettes hovedsakelig for trykk, så risiko for flyt som følge av strekk er minimal.

### 4. Utregning av margin mot eventuell knekklast:

$$P_{kr} = \frac{\pi^2 * E * I}{L^2}$$

$$I = \frac{\pi * r^4}{4} = \frac{\pi * 3,1mm^4}{4} = 72,53mm^4$$

$$P_{kr} = \pi^2 * 193 * \frac{10^3N}{mm^2} * 72,53mm^4 / 80mm^2 = 22480 N$$

$$P_{sf} = \frac{P_{kr}}{P} = \frac{22480N}{2160N} = 10,4$$

### 3.2.3 Beregning av pasning for bunnplate og glidebolter

Beregninger for å finne riktig pressmonn for sammenstilling av bunnplate og glidebolter, samt kontroll av spenningsnivå i glidebolt og bunnplate ved en gitt presspasning

Tabell 3:  
Inngangsdata for beregning av pasninger

Data	Glidebolt	Bunnplate
Areal (As)	50,26 mm <sup>2</sup>	NA
Diameter (d)	Ø 8,00 mm	Ø 8,00 mm
Materiale	AISI 304	AA 6061 T6
Poisson (tverrkontraksjon)	Vg = 0,33	Vp = 0,29
E-modul	193 GPA	68,9 GPA

$$u_1 = 0.1$$

$$F1 = \frac{F_{a1}}{n} = \frac{2160N}{4} = 540N$$

$$F2 = \frac{F_{a2}}{n} = \frac{7200N}{4} = 1800 N$$

#### 1. Utregning av nødvendig flate-trykk basert på aksialkraft

$$F_1 = p_{max} * \pi * d * l * u_1$$

$$p_{max1} = \frac{F_1}{\pi * d * l * u_1} = \frac{540N}{\pi * 8mm * 10mm * 0,1} = 21,48 Mpa$$

$$p_{max1} = \frac{F_1}{\pi * d * l * u_1} = \frac{1800N}{\pi * 8mm * 10mm * 0,1} = 71,62 Mpa$$

#### 2. Utregning av forventet pressmonn som følge av flate-trykk

Formelen forenkles som følge av massiv aksling, r-indre for bolt går bort.



$$\Delta d = 1000 * d * p_{max} * \left( \frac{1}{E2} * \left( \frac{1 + x_2^2}{1 - x_2^2} + v_p \right) + \frac{1}{E1} * \left( \frac{1 + x_1^2}{1 - x_2^2} - v_g \right) \right)$$

$$x = \frac{d_{indre}}{d_{ytre}} = x_1 = \frac{0}{8} \cap x_2 = \frac{8}{20}$$

$$\frac{1 + x_2^2}{1 - x_2^2} = \frac{1 + 0,4^2}{1 - 0,4^2} + 0,29 = 1,67$$

$$\frac{1 + x_1^2}{1 - x_1^2} = \frac{1 + 0^2}{1 - 8^2} + 0,33 = 0,3141$$

$$\Delta d_1 = 1000 * 8,0mm * 21,48 \frac{N}{mm^2} * (2,48 * 10^{-5} + 3,33 * 10^{-6}) = 4,83 \mu m$$

$$= 0,0048mm$$

$$\Delta d_2 = 1000 * 8,0mm * 71,62 \frac{N}{mm^2} * (2,48 * 10^{-5} + 3,33 * 10^{-6}) = 16,21 \mu m$$

$$= 0,01621mm$$

Dette gir et pressmonn på henholdsvis:

0,0048mm for 540N

0,01621 mm for 1800N

### 3. Utregning av toleranse

Glidebolter kommer målsatt til Ø 8,00 mm fra fabrikk, som betyr at muligheten for å etablere en pasning som oppfyller minimums pressmonn må forekomme i bunnplate.

Passende toleransevalg i henhold til NS-ISO-286-2: (Hartvigsen *et al.*,2006)

For F1 = 540 N blir pasning for boring bunnplate:

Basismål = 8,000mm, pasning N6  $\text{Ø } 8,000mm \frac{-0.005}{-0.013}$

Øg = 7,995

Ng = 7,987

For F2 = 1800N blir pasning for boring bunnplate

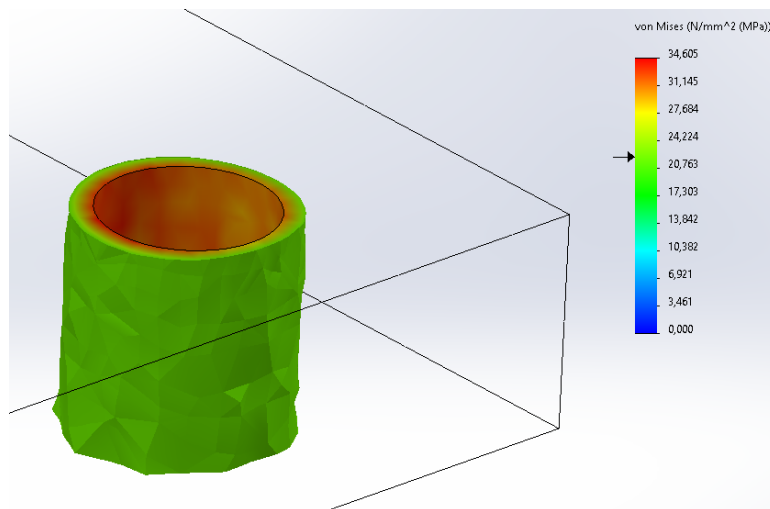
Basismål = 8,000mm, pasning R5  $\text{Ø } 8,000mm \frac{-0.017}{-0.023}$

Øg = 7,983

Ng = 7,977

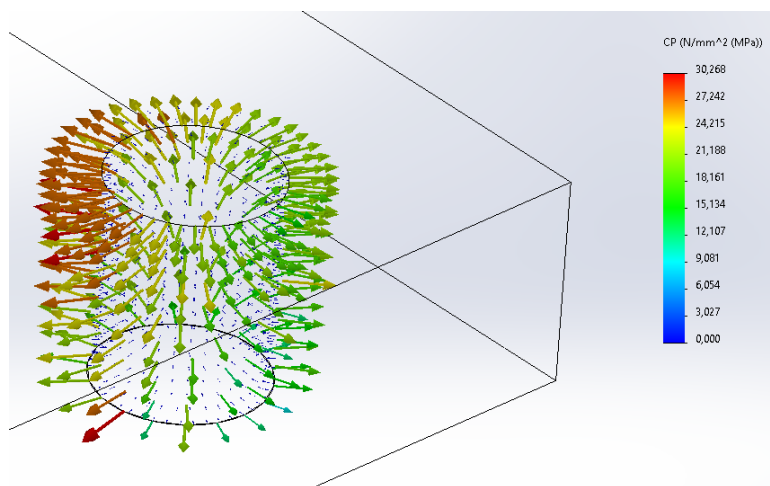
#### 4. Etterprøving av hånd-regning ved hjelp av elementmetoden

Først ble aksling gjort isometrisk. Deretter ble krympepassing med 0,1mm etablert og mesh kontroll satt på flater som inngikk i presspassing.



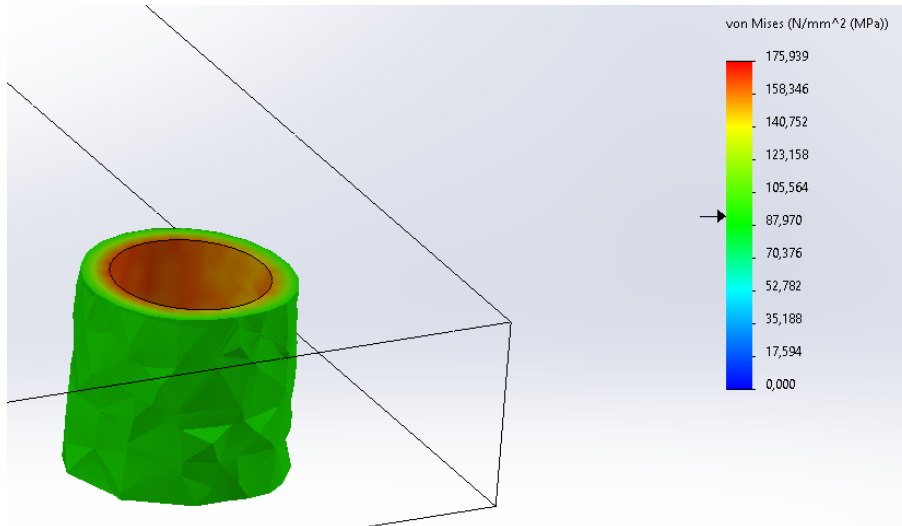
Figur 12: Spenning presspassing

Figur 12 viser von Mises spenning som følge av boring  $\text{\O} 7.995 \text{ mm}$



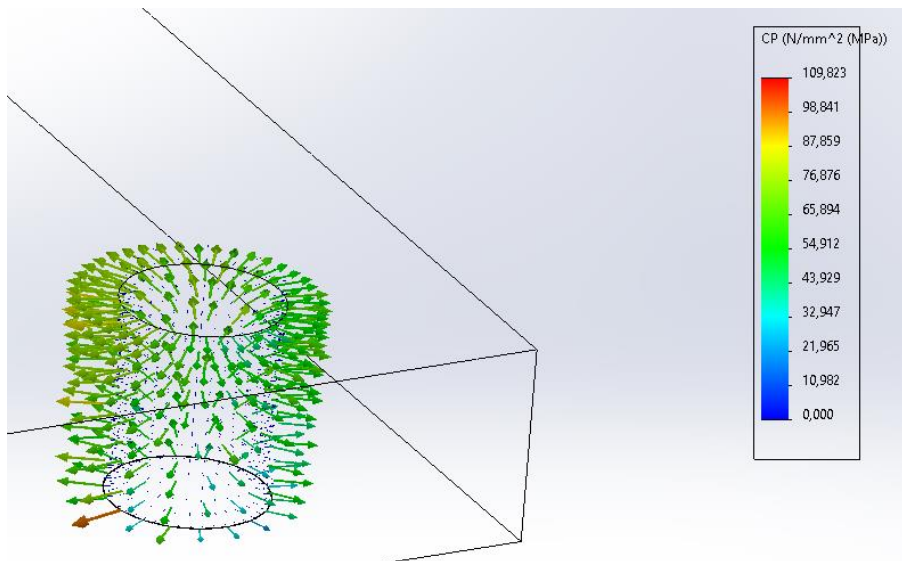
Figur 13: Trykk presspassing

Figur 13 viser kontakt-trykk som følge av boring  $\text{\O} 7.995 \text{ mm}$ . Som forventet vil flate-trykk avta mot kortsider med mindre volum. Forskyvning ligger for hele hjørnet på 0.001 mm.



Figur 14: Spenning presspasning 2

Figur 14 viser von Mises spenning som følger av boring lik  $\text{\O} 7.983 \text{ mm}$



Figur 15: Trykk presspasning

Figur 15 viser kontaktrykk som følge av boring  $\text{\O} 7.983 \text{ mm}$ . Dette stemte godt overens med håndberegninger.

### 3.2.4 Beregning av nødvendig temperatur i bunnplate for montering av glidebolter

Tabell 4:

Inngangsdata for beregning av press-montering

Data	Bunnplate (2)	Aksling (1)
Nominell diameter (D)	Ø 8,00mm	Ø 8,00mm
Utvidelseskoeffisient ( $\alpha$ )	$23,6 * (10^{-6}) \mu m / m . k$	$17,3 * 10^{-6} (\mu m / m . K)$
Grunntemperatur (C)	20 rader	20 grader
Nødvendig utvidelse (mm)	Min. 0,005 mm & 0,016mm	0 mm & 0 mm
Elastisitet modul	68,9 GPA	193 GPA

$$\Delta d = T_1 * D * \Delta T_1(0) - \alpha_2 D * \Delta T_2$$

$$\Delta d = -\alpha_2 * D * T_2$$

$$T_2 = \frac{\Delta d}{-\alpha_2 * D}$$

$$T_2 = \frac{0,005mm}{-22,2 * \frac{(10^{-6}) \mu m}{m} . k * 8,00mm} + 20 = 46,48 \text{ } ^\circ\text{C}$$

$$T_2 = \frac{0,015mm}{-22,2 * \frac{(10^{-6}) \mu m}{m} . k * 8,00mm} + 20 = 99,45 \text{ } ^\circ\text{C}$$

Begge temperaturer er under temperaturer som kan påvirke materialets egenskaper på permanent basis.

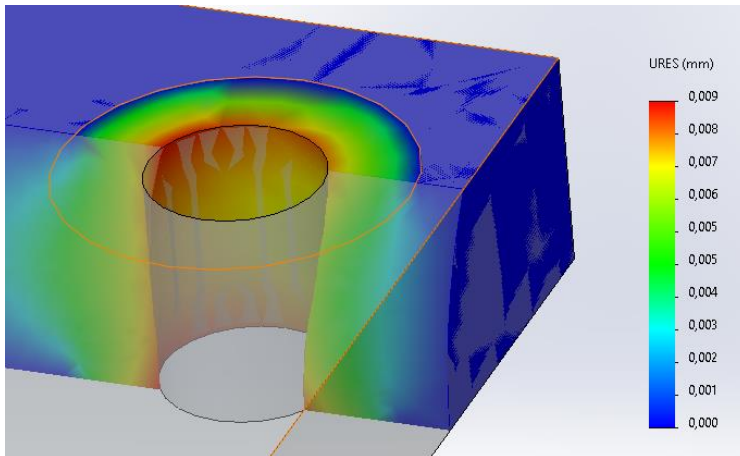
Radielle spenninger som oppstår som følge av varmeutvidelse

$$\sigma_r = \frac{E * \Delta D}{4 * a} * \left(1 - \frac{a^2}{b^2}\right)$$

$$\sigma_{r1} = \frac{68,9 * \frac{10^3 N}{mm^2} * 0,005mm}{4 * 8mm} * \left(1 - \frac{8mm^2}{16mm^2}\right) = 8,07N/mm^2$$

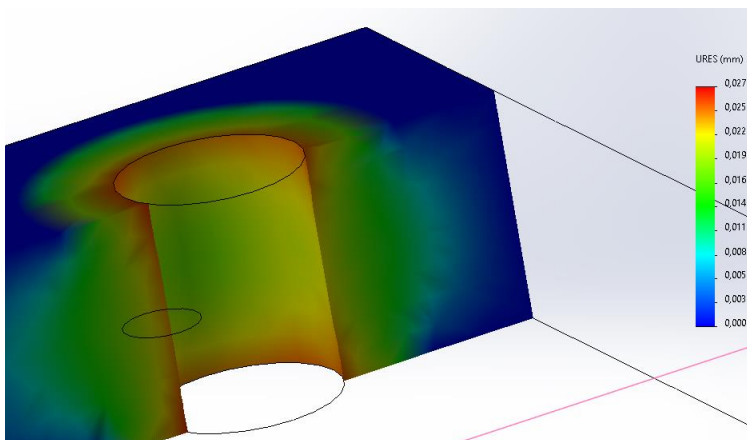
$$\sigma_{r2} = \frac{68,9 * \frac{10^3 N}{mm^2} * 0,015mm}{4 * 8mm} * \left(1 - \frac{8mm^2}{16mm^2}\right) = 24,22N/mm^2$$

Kontroll av varmeutvidelse og spenninger med elementmetoden.



*Figur 16: Varmeutvidelse elementmetode 1*

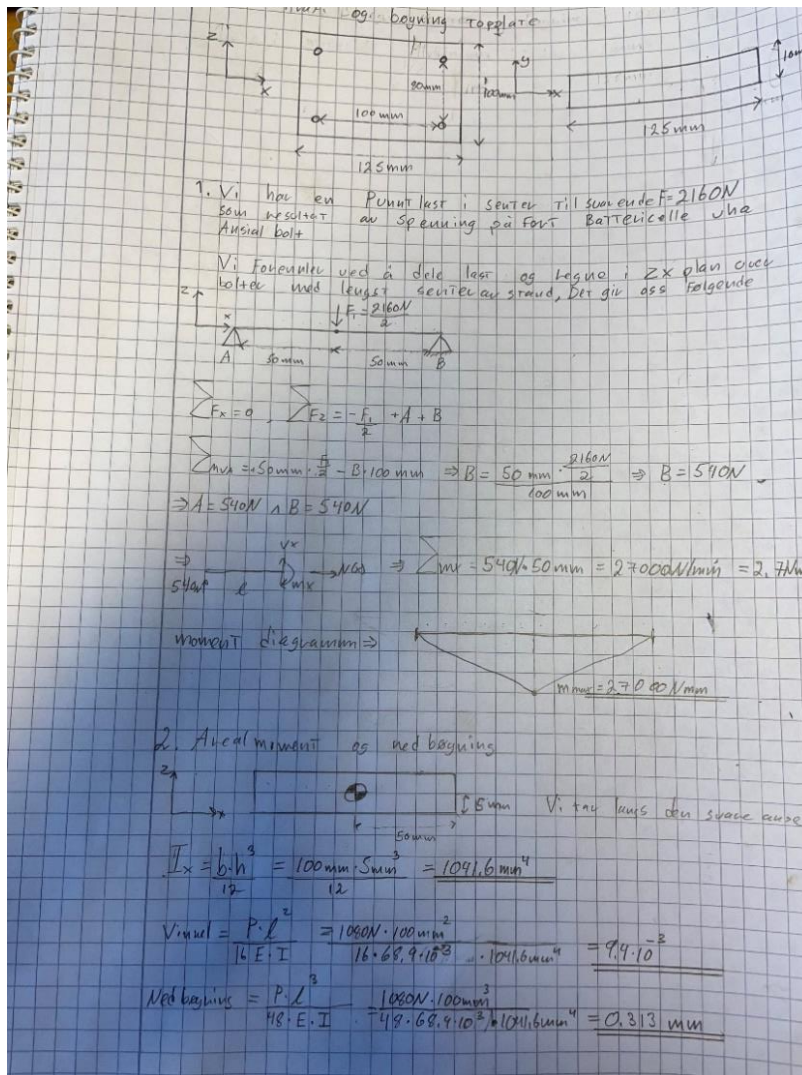
Figur 16 termisk ekspansjon ved romtemperatur 20 grader og oppvarmet til 48 grader



*Figur 17: Varmeutvidelse elementmetode 2*

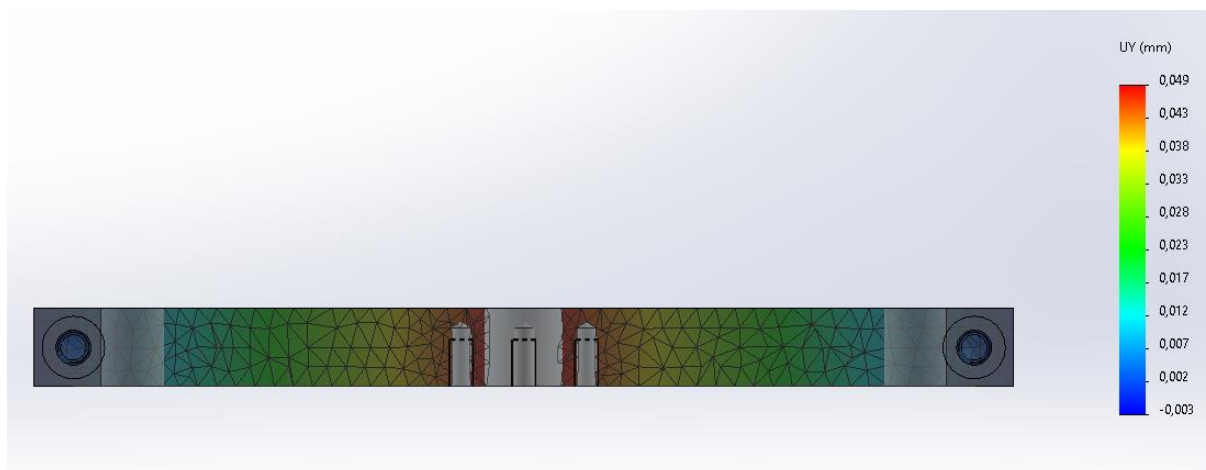
Figur 17 viser termisk ekspansjon ved romtemperatur 20 grader oppvarmet til 99.5 grader

### 3.2.5 Utbøyning av topp-plate som følge av totalkraft



Figur 18: Håndberegning

Kraft som følge av 0,3 MPa på batteri gir en utbøyning lik 0,313 mm sett fra et 2-dimensjonalt-utsnitt.



Figur 19: Utbøyning elementmetoden

Utbøyning 2D snitt som følge av 0,3 MPa overført til senter-mutter og fordelt på mutterflate.  
0,049 mm

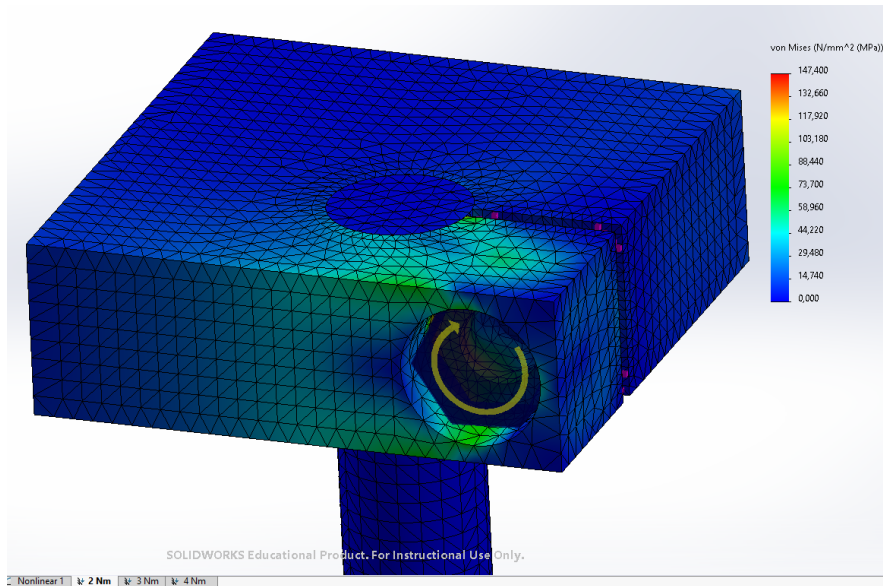
### 3.2.6 Moment for låsing av glidebolter i topp-plate

Håndberegning for låsebolter var ikke hensiktsmessig. Det kunne heller ikke gi ett nøyaktig moment for låsebolter med tanke på at man må kombinere flere begrep fra mekanikken i noe som ikke eksisterer i ett standard formelverk. Det ble av den årsak hovedsakelig utregnet med elementmetoden ved hjelp av Solidworks.

Minimal kontaktspenning for låsing av bolt:

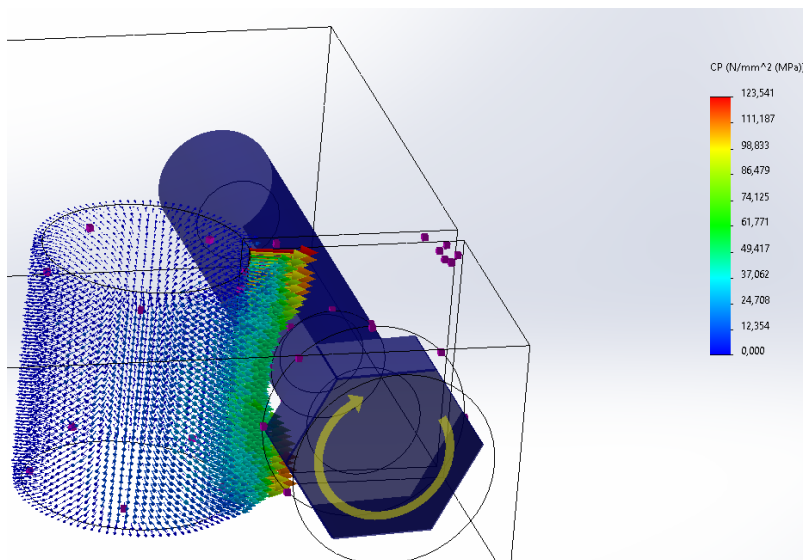
$$P_{min1} = 21,48 \text{ Mpa} = Cp_1$$

$$P_{min2} = 71,62 \text{ Mpa} = Cp_2$$



Figur 20: Innfestning elementmetoden 1

Figur 20 viser oss Mesh som ble kjørt med manuelt påført mesh kontroll til alle skarpe overganger og flater involvert i pasning ved samtlige simuleringer. Ingen spenningskonsentrasjoner gikk over flytegrense, selv uten tilstrekkelig med kjervsnitt i bolthull.



Figur 21: Innfestning elementmetode 2

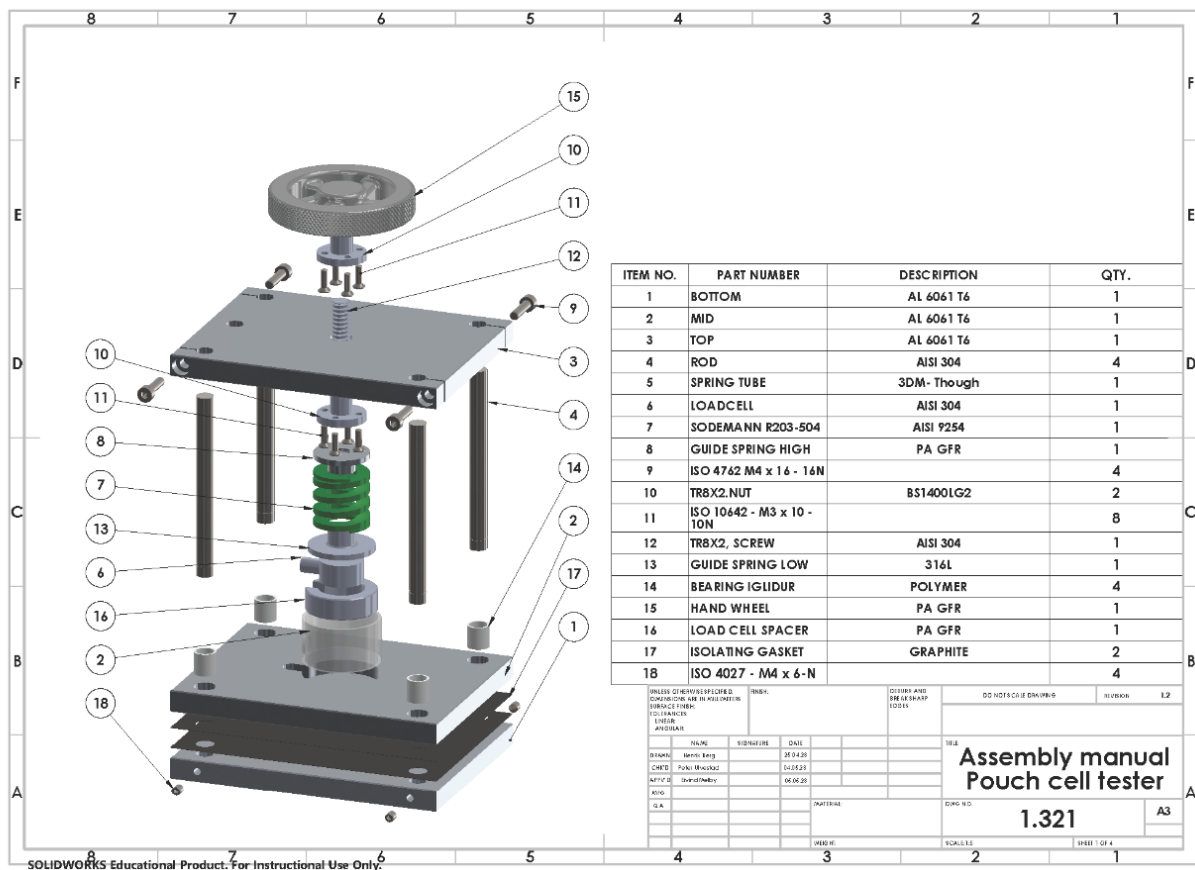
Figur 21 viser at ved 2 Nm tilført bolten for låsing av topp plate mot guidebolt, så oppnås det tilstrekkelig kontaktrykk i boring mot guidebolter. Dette var med friksjonskoeffisient lik 2 og med en lokal kontakt mellom bolt/topplate, samt låsing av glidebolt. Det vil forekomme en svak forspenning som følge av at hullene vil ha 0.001 mm press før tiltrekning av bolt.



### 3.3 Komponenter

Dette avsnittet ble etablert for å definere det unike med hver komponent som ble tilvirket eller bearbeidet, og kort om hvorfor de ble valgt i det ferdige designet. Komponenter som ble ansett som hyllevare ble ikke inkludert som egne avsnitt av praktiske årsaker.

Detaljerte dimensjoner av komponenter kommer frem i vedlegg 7 «maskintegninger». En mer detaljer sammenstillings-beskrivelse kommer frem i vedlegg 6 «montasjeanvisning».



Figur 22: Oversikt over komponenter som inngår direkte i sammenstillin

Tabell 5:  
Oversikt over komponenter

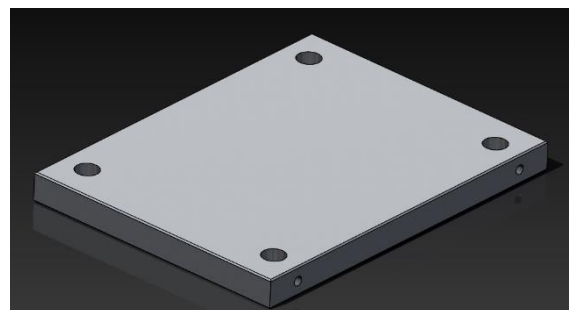
KOMPONENTNR	KOMPONENTNAVN	MATERIALE	ANTALL	LAGES	BEARBEIDES	FERDIG
1	BOTTOM PLATE	AL 6061 T6	1	X		
2	MID PLATE	AL 6061 T6	1	X		
3	TOP PLATE	AL 6061 T6	1	X		
4	ROD/GUIDEBOLT	AISI 304	4		X	
5	SPRING TUBE	3DM though	1	X		
6	LOADCELL	AISI 304	1			X
7	SODEMANN, R203-504	AISI 9254	1			X
8	SPRING GUIDE, TOP	GFR PA	1	X		
9	ISO 4762, M4x16 -16N		4			X
10	TR8x2, NUT	BS1400 LG2	2			X
11	ISO4762, M4x16		4			X
12	TR8x2, Screw	AISI 304	1		X	
13	SPRING GUIDE, LOW	GFR PA	1	X		
14	BEARING, IGLIDUR	NYLON	4		X	
15	HAND WHEEL	3DM-Though	1	X		
16	LOAD CELL, SPACER	PA GFR	1	X		
17	ISOLATIONG GASKET	Graphite	2		X	
18	ISO 4027, M4x6-N		4			X
19	ARDUINO MEGA		1		X	
20	RTC		1			X
21	ADC		1			X

### 3.3.1 Bunnplate (1)

Bunnplaten består av aluminiumslegeringen 6061 T6 og produseres på basismålene 120mmx95mmx10mm. Det er utregnet en presspassning i avsnitt 3.2.3 for bolthull som låses mot guidebolter, dette er for å slippe usikkerhet og ekstra kostnad med det å ha en gjenget

festemetode. For ekstra sikkerhet mot løsrivning

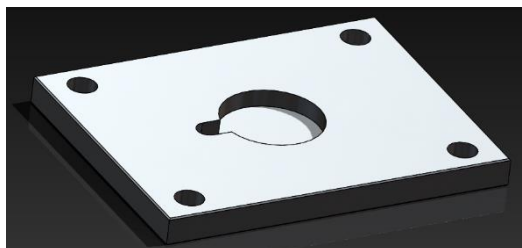
som følge av aksialkraft, må det maskineres ett hull for hver guidebolt som låses av ved bruk av spissede M4 settskruer mot spor i aksling (18). Platen har i stor-skala produksjon svært kort produksjonstid og har få til ingen operasjoner som innehar høy verktøyslitasje. Man så for seg at utvendige basismål og hull skjæres ut ved hjelp av vannkutter, for så å foreta passninger og låsebolter i en ny oppspenning som en god materialbearbeidings metode.



Figur 23: Solidworks-modell av bunnplate

### 3.3.2 Midtplate (2)

Midtplaten består av aluminiumslegeringen 6061 T6 og er tenkt produsert på basismålene 120mmx95mmx10mm. Det er 4 hull hvor det skal maskineres en pasning i henhold til leverandør av de lineære lagrenes spesifikasjoner. Det er også en

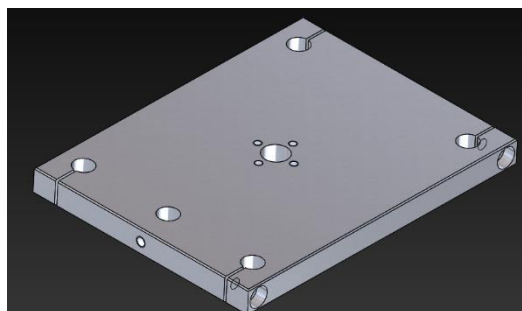


Figur 24: Solidworks-modell av midtplate

forsenkning for sentrering av lastcelle og styring av riggens innmat, samt et spor for kabelgjennomføring til lastcelle. Denne platen krever bare en oppspenning og er derfor en rimelig del å produsere i en potensiell storskala/serie produksjon.

### 3.3.3 Topplate (3)

Topplaten består av aluminiumslegeringen 6061 T6 og produseres på basismålene 120mmx95mmx10mm. Det skal maskineres 4 hull for guidebolter, 1 hull for TR8x2 mutter, 1 hull for en eventuell avstandsmåler, samt hull for å montere låsebolter til låsing av guidebolter og topplate.



Figur 25: Solidworksmodell av topplate

Denne platen innehar høyest produksjonstid og

dermed høyest kostnad av platene. Det vil kreve 3-4 oppspenninger med kontrollmåling og nullstilling underveis.

### 3.3.4 Materialvalg for plater (1,2 og 3)

Bunnplaten, midtplaten og topplaten utgjør en stor del av produktet. IFE sitt ønske var at produktet skulle være så lett som mulig, og at tyngre metaller burde unngås. I tillegg skal materialet stå under statisk trykk over lengre tidsperioder, så det er svært viktig at det ikke oppstår siging i materialet. Temperaturforholdene har mye å si for siging, og for de vanligste metallene må temperaturen overstige 250°C for at det skjer (G.S. Wang, K. Liu og S.L. Wang, 2018, 1. Introduction avsnitt 3). Skapene som riggene skal stå i har en makstemperatur på 100°C, så det antas at siging ikke blir et problem.

Aluminium 6061 T6 regnes for å være svært godt egnet til strukturelle applikasjoner på grunn av legeringens spesielle egenskaper. Hovedelementene som er tilsatt i legeringen er magnesium og silisium (*Aluminum 6061-T6*, u.å.), der magnesiumet bidrar til å øke styrken og silisiumet reduserer metallens smeltetemperatur (*How and why alloying elements are added to aluminum*, u.å.). I tillegg går denne legeringen gjennom en kunstig aldriingsprosess for å oppnå maksimal styrke. Legeringen oppnår derfor høy styrke, stor formbarhet, gode mekaniske egenskaper og stor sveisbarhet. Denne legeringen er også svært enkel å maskinere, og det gir fordeler i forhold til produksjonstid og verktøyslitasje. Den kan egne seg for eloksering om det skulle bli aktuelt (Rajaa *et al*, 2018).

### 3.3.5 Rod/Guidebolter (4)

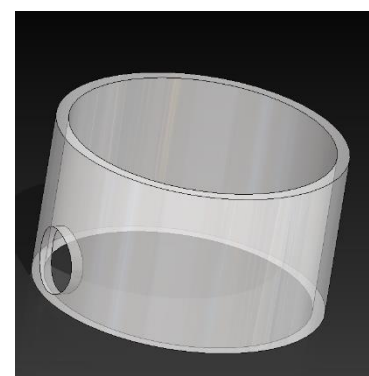
Guideboltene er laget av rustfritt stål type AISI 304. Dette materialet er svært mye brukt innen produksjon av sveisede rør og til store og små konstruksjoner. Det består av 18% krom og 8% nikkell, og det er denne sammensetningen som gir stålet dets spesielle egenskaper (*AISI Type 304 Stainless Steel*, u.å.). Sammensetningen sørger for at stålet blant annet har høyt flytepunkt, god sveisbarhet og gode mekaniske egenskaper. Guideboltene krever svært lite maskinering. Det er kun kapping, plandreining og dreining av spor for låseskruer mot bunnplate som er nødvendig etter mottak av guideboltene. Guideboltene kommer på mål  $\text{Ø } 8.00 \text{ mm } \pm 0.01$ .



Figur 26: Guidebolter

### 3.3.6 Deksel/rør (5)

Røret har som hovedfunksjon å gi fjærpakke med guide for fjør en støtte når det ikke er lastpåkjenning og for å sentrere kabel mot utfresing i mid-plate. Denne blir av praktiske og estetiske årsaker printet i et resin som er transparent og sterk. Man står relativt fritt til materialvalg som følge av at den ikke utsettes for noen krefter. Funksjonen er å gi montør/bruker mulighet til å overvåke dynamiske komponenter og tilse at trim i riggen er riktig montert på linje. De har også en veldig kort produksjonstid, og kan fremstilles lokalt.



Figur 27: Deksel/rør

### 3.3.7 Lastcelle (6)

To typer lastceller har blitt brukt i dette prosjektet. Den ene er produsert av TE Connectivity og er av type FX29 og den andre er produsert av HT Sensor Technology CO., Limited og er av type TAS606. Tabell 6 viser spesifikasjonene for FX29 (TE Connectivity, 2020) og TAS606 (HTC-Sensor.com, u.å).

Tabell 6:  
Spesifikasjoner for FX29 og TAS606.

Type	H x D	Min kg	Maks kg	Oppløsning	Usikkerhet
FX29	(5,45x19,4) mm	0 kg	90,7 kg	20 mV/V	±1%
TAS606	(12 x 20) mm	0 kg	200,0 kg	1,5 mV/V	±0,3%

### 3.3.8 Fjær (7)

Ved opplading og utlading av batterier vil de kunne utvide seg. En undersøkelse viser at de kan utvide seg 300 – 320 %, og dette kan medføre at silisiumpartiklene i batteriet og at bindemidlet bryter ned. En løsning på dette er å sette batteriene under forspenningstrykk. Enten ved å utøve statisk eller dynamisk trykk med fjærer. (Müller, *et al.*, 2019).

For å ha litt spillerom slik at fjæra ikke deformeres må det legges til en sikkerhetsfaktor. Dette er beskrevet i kapittel 3.2.1. Her er maksimalspenning som kan forventes, inkludert sikkerhetsfaktorer lik 1.0 MPa. Fjæra må derfor ha en høyere flytegrense  $R_e$  enn 1,0 Mpa.

Fjæra som skal brukes må komme med en oppgitt fjærkonstant. Da kan man beregne ut ifra hvor mange mm fjæra er trykket sammen hvor høyt trykk som utøves til enhver tid. Det mest praktiske er å benytte fjærer beskrevet i en standard (International Standard, 2019). Fjærene beskrevet i standarden er fargekodede, og de viser hvor kraftige de er. Grønn er lett, blå er medium, rød er sterk og gul er ekstra sterk fjær. Disse kan da byttes ut dersom man ønsker mer kraft, uten å endre fjærlengde og diameter. Utfordringen med disse fjærene er at de har ± 10 % variasjon i fjærkonstanten. Fordelen med å velge fjærer beskrevet i denne standarden er at så lenge fjæren ikke blir strukket eller klempt sammen mer enn det som er grensen oppgitt i standarden, så skal man ikke gå over flytegrensen. Etersom fjæra blir satt under relativt konstant trykk der den ikke blir påført svingninger, kan problemer med materialtretthet sees bort ifra.

### 3.3.9 Spring guide topp, bunn og spacer (8,13,16)

Fjærguider (8,13) har som hensikt å overføre kraft som går gjennom fjær (7) til lastcelle og sørge for at fjær har en rett vertikal kraftretning. Disse produseres i materialet 316 L for prototypen, men det burde gjøres en studie på om de kan produseres på restkapp av plater (1,2,3) i aluminium eller ved bruk av samme materiale som for spacer(16). De er lite komplekse komponenter å maskinere, men trykk ekstrudering burde vurderes til alternativ produksjonsmetode ved serieproduksjon. Spacer (16) er for å sentrere lastcelle (6).



Figur 28: Innmat fjær

Spacer er ikke utsatt for krefter og man står relativt fritt i forhold til materialvalg med tanke på styrke, det er i dette tilfellet valgt glassfiberarmert polyamid av rent praktiske årsaker som at man kan printe spacer på stedet og innen kort tid.

### 3.3.10 TR8x2, Mutter og skrue (10,12)

Tr8x2 med en gjengeinngang for skrue (12) og mutter (10). Delene er i utgangspunktet tiltenkt som 3D maskinelementer i printere, noe som gjør dette til en god økonomisk løsning på grunn av tilgang i markedet, samt at det er utskiftbare komponenter. Trapesgjenger er til det formålet å omdanne et dreiemoment fra rattet til aksialkraft i bolten. Skruen er av AISI 304 og mutter er av bronse for å sikre gode glide-egenskaper og unngå mulige helseskadelige smøremidler. Trapesgjenger har også



Figur 29: TR8X2 Mutter og skrue

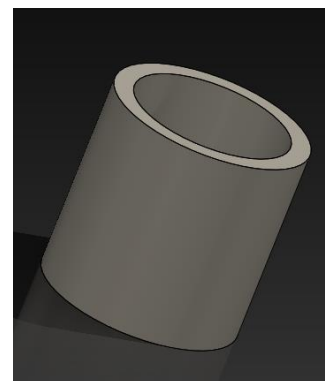
gode selvlåsende egenskaper som reduserer en eventuell kontra/låse mutter ved bruk av vanlige 60 graders gjenger (Dahlvig *et al.*1991). I henhold til kalkulasjonene er skrue og mutter godt under flytegrenser, og begrensninger knyttet til kraftpåkjenninger og spenninger. Bruken av denne typen gjenger åpner også muligheter for fremtidig dreiemoment ved hjelp av en el-motor.

### 3.3.11 Bearing, Iglidur (14)

Bearing Iglidur (JSM-0810-10) (Iglus, 2023) kommer med standard mål på  $d1 = \text{Ø } 8 \text{ mm}$   $d2 = \text{Ø } 10 \text{ mm}$  og lengde på 10 mm direkte-bestilt med en H7 pasning for montering i mellomplate.

Den har en lav kostnad, vekt og byggehøyde i forhold til de først tiltenkte lineære kulelagrene som var tiltenkt oppgaven. Iglidur sitt materiale er en høy-ytelses polymer kombinert med en matrise av polymer med fibre og fillers. Den har gode mekaniske egenskaper knyttet til UV og høy kjemisk resistens, samt at den ivaretar

mekaniske egenskaper helt opp til Tg på 90 grader celsius og har en friksjonskoeffisient ned mot 0.01. Den inngår som en av få slidedeler på rigg, men har en forventet levetid utover kravspesifikasjoner fra IFE på 5 år, basert på hvor statisk riggen er og miljøet den står i.

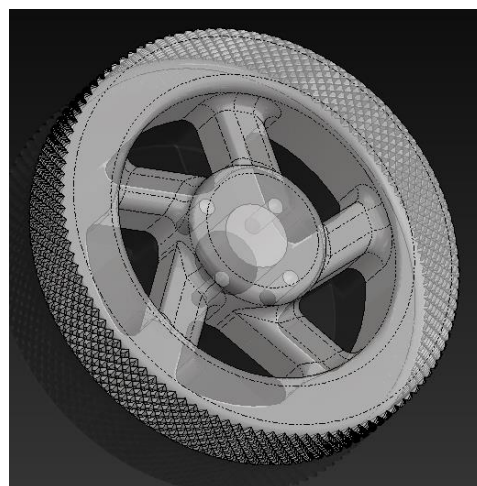


Figur 30: Iglidur bearing

### 3.3.12 Hand Wheel (15)

Hånd-ratt for å tilføre ett moment og dermed aksialkraft tilført batteri-celle er konstruert med utgangspunkt i funksjon og estetikk. Funksjons faktorer er å ha tilstrekkelig diameter for å kunne danne et ønsket moment uten å måtte påføre rattet for mye kraft i form av moment. Materialvalget er et transparent sterkt resin, som utfyller funksjonen å kunne tilse at det er ønsket inngrep mellom mutter og ratt, samt tilse at bolteforbindelser er riktig montert.

Estetisk er det et moderne design med et materialvalg som er både moderne og innovativt.



Figur 31: Håndratt

### 3.3.13 Arduino Mega med komponenter (19,20,21)

Arduino Mega, RTC (Real Time Clock) og ADC (Analog-to-Digital Converter) er komponenter brukt spesifikt til trykkovertvåking og blir beskrevet i dybde i kapittel 3.4 Produksjon trykkovertvåking.

### 3.3.14 Termisk matte/isolator (17)

#### Materialvalg

Det var nødvendig med et elektrisk isolerende og varmeledende materiale mellom battericellene og aluminiumsplatene, slik at strøm ikke ledes og varme ledes bort fra battericellene. I dette tilfellet vil man ha et material med høy

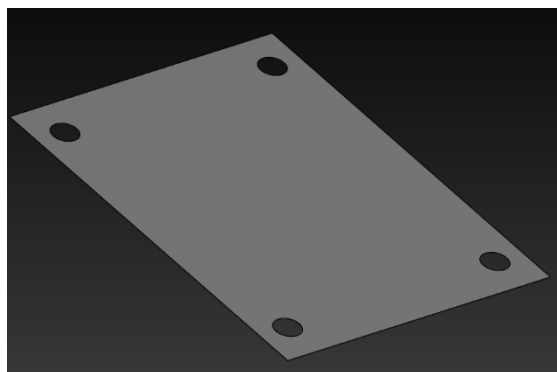
varmeledningsevne og stor elektrisk motstand. Et

materials varmeledningsevne er antall watt ledet per meter tykkelse av materialet, per grad av temperaturforskjell mellom den ene og den andre siden. Det skrives ofte som  $W/mK$ , og jo høyere tallet er jo bedre leder materialet varme (Grøn, 2022).

Som regel har materialer med høy varmeledningsevne også høy elektrisk ledningsevne, og materialer med lav varmeledningsevne har ofte også lav elektrisk ledningsevne (Helseth, 2023). Plast og gummi er materialer som ofte brukes til elektrisk isolering, men de har dårlig varmeledningsevne. Det finnes flere typer grensesnittmaterialer som brukes mellom elektriske enheter og det tilstøtende materialet, og de kommer også i ulike former. Disse materialene er spesielt sammensatt av flere komponenter, slik at de oppnår både god varmeledningsevne og lav/ingen elektrisk ledningsevne. Grensesnittmaterialene brukes som regel i form av termisk pasta eller termiske plater. For denne oppgaven er det mest hensiktsmessig å bruke termiske matter. Mattene lages som regel av silikon eller grafitt, og begge disse er gode alternativer.

Det er et hav av matter å velge mellom fra ulike produsenter, der varmeledningsevne, hardhet og mattenes tykkelse er det som varierer mest. Det var viktig at mattene ikke var tykkere enn 0.25mm, og helst enda tynnere om mulig. Det viste seg at silikonmattenes varmeledningsevne blir dårligere jo tynnere matta er. Slik er det ikke for grafittmattene, og i tillegg har disse generelt mye høyere varmeledningsevne og hardhet enn silikonmattene.

Valget falt derfor på en grafittmatte som er illustrert i form av en solidworks-modell i figur 32. Matten har svært god varmeledningsevne på  $10 W/mK$ , en tykkelse på kun 0.16 mm og hardhet på hele 80 Shore A (Thermal Interface Sheet, 2023). Grafittmatten egner seg altså svært godt til formålet.



Figur 32: Termisk matte i grafitt



### **3.3.15 Øvrige komponenter og skruer (9,11,18)**

Øvrige komponenter og skruer er valgt fra Solidworks sin verktøykasse ifra ISO katalog. Dette er for å i høyest mulig grad standardisere både skruevalg og materialvalg for senere innkjøp eller endringer. M4 og M3 skruer er valgt ut ifra dimensjonering og kraftpåkjenninger som konstruksjonen utsettes for, samt det å ha en minst mulig påvirkning på konstruksjons bygge mål og oppnå en helhetlig samt gjennomført estetikk.

## **3.4 Produksjon/montering mekanisk**

Dette avsnittet er en beskrivelse av produksjonen og montering av to rigger for celletesting. Prosessen beskrives stegvis for produksjon og montering, som det ble gjort ved fremstilling av prototype. Produksjon og montering ble utført med det utstyr som var tilgjengelig der og da. Det er da ikke den mest lønnsomme og effektive produksjonsmetoden for en rigg, men en beskrivelse av hvordan det ble utført i forbindelse med fremstilling av en prototype for denne rapporten.

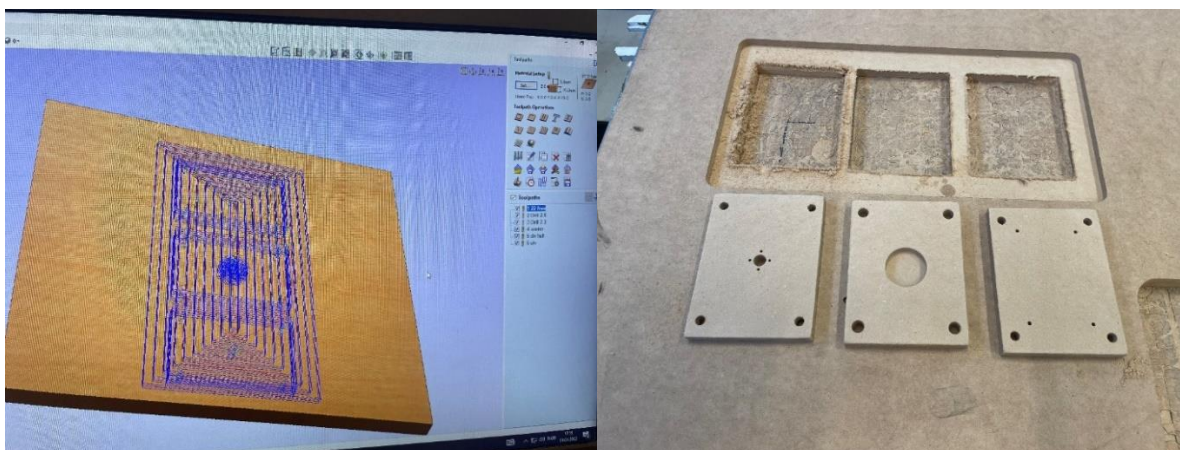
### **3.4.1 Deler som inngikk i produksjon**

I løpet av april ble følgende materiell mottatt for produksjon/maskinering/montering

1. Aluminiumsplate 6061 T6 410mmx1270mmx10mm (restkapp fra Astrup stål)
2. 2 stykk gjengestenger TR8x2 og 6 stykk muttere TR8x2
3. 6 stykk 150mmx150mmx 0.15mm grafitt plater med heft
4. 3 stykk fjører av forskjellig stivhet og 2 stykk 500 mm guidebolter AISI 304
5. 900Ml resin 3D-M tough og en rull med F3 PA GF Pro polymer
6. Restkapp Ø30x120 mm 316L bolt.
7. Øvrige komponenter ble kjøpt lokalt gjennom Tess-partnershop.
8. Restkapp MDF plate 1200mm x 2400mm x 20 mm

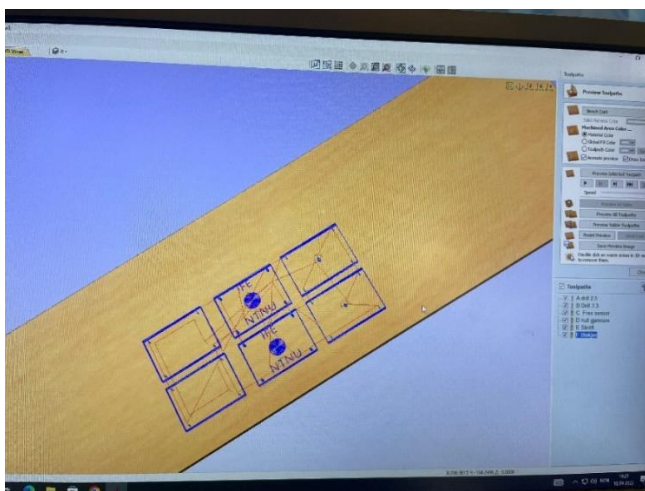
### 3.4.2 Maskinering av plater

Det ble tatt ut et dxf (2D-snitt) fra plater i Solidworks. Disse ble importerte til CAM programvaren Aspire for en cp1325 ATC cncPRO 3-akse maskin. Deretter ble det satt opp ett program for utskjæring av de tre platene i designet, som skulle skjæres ut av en MDF-plate. Dette var for å teste programmet som ble laget og luke ut eventuelle feil knyttet til hvordan programmet kjøres på maskin. Disse ble deretter eksporterte til maskinen på ett numerisk kontroll språk for maskiner kalt G-code Tap, via en postprocessor for den gitte maskinen.



Figur 33: G-kode for test-skjæring og gjennomført test i MDF

Utfresingen i MDF platen viste seg å være feilfri. Det ble derfor utarbeidet G-kode i Aspire for to komplette sett med aluminiumsplater til rigg.



Figur 34: Slicer for aluminium plater

Teksten for IFE og NTNU ble lagt til av estetiske grunner da dette blir produsert 2 stykk prototype rigger. Turtall og mating ble delvis utregnet for hånd ved hjelp av

verkstedshåndboka (Dahlvig *et al.*, 1991), og kontrollert ved bruk av nettsiden machining doctor (Spencer, 2019, Milling Parameters). Parameterne måtte reduseres noe med hensyn på at maskinen ikke har vannkjøling, men bare tilgang på luftkjøling.

Verktøy som inngikk i produksjonen, var følgende:

- Ø 3,3 mm & Ø 2,5mm hardmetall bor
- Ø 6,0 mm 2-skjærs carbid fres
- Ø 2,0 mm konisk 2-skjærs carbid fres
- T-spor klemmer 10 mm
- Maskinstikke og 3D taster
- Skivefres 1 mm, M4 og M3 Gjengetapp
- Brotsj Ø8 H7, H7 Ø10, og N6 Ø8



*Figur 35: Ferdig maskinerte plater*

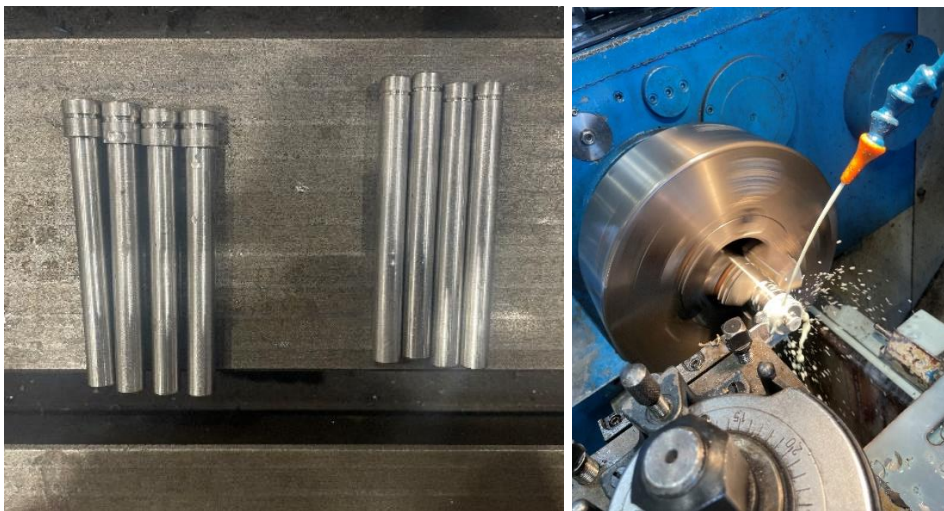
Utskjæringer av plater ble utført med dybde 9,95mm mot 10 mm nominell tykkelse, for å unngå at emnet eventuell løsner eller får sperringer ved siste kutt. De kunne da lett dunkes ut med gummihammer etter maskinering. Platene ble så 3D scannet med en 3D-skanneren Creaform Handyscan black elite og eksportert som stereolitografi fil til Polyworks inspector for å utarbeide målebrev (vedlegg 5). Dette var hovedsakelig for å kontrollere opp mot utvendige basismål og eventuell nedbøyning som følge av spenningsfrigjøring som resultat av

maskinering. Avspenningsgløding ble vurdert i forkant, men ikke utført da det ved målebrev viste seg å ikke være behov for i dette tilfellet.

Emnene ble så spent opp i maskinskrustikke og hull med gitte pasninger ble brotsjet, samt at gjenger ble lagd manuelt med gjengetapp. Deretter ble platene slipt på 25Um papir på planbord, for så å bli pusset over med auto-sol for bedre utsende.

De termiske grafittpakningene ble også skjært ut ved hjelp av drag-knife i samme cnc maskin. Øvrig maskinering av plater ble utført i ved hjelp av samme dxf snitt som for platene.

### 3.4.3 Maskinering av guidebolter og fjærholdere



*Figur 36: Manuell Dreining*

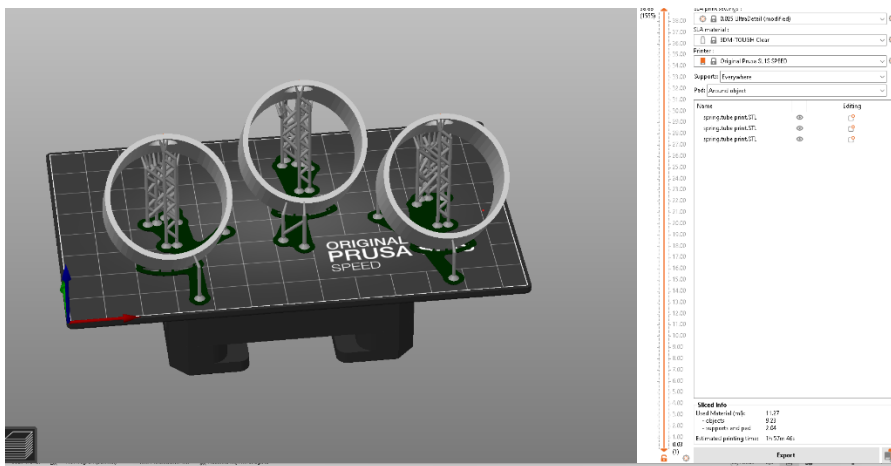
Guideboltene kom målsatt til  $\text{Ø}8,00\text{mm} \pm 0,01$ , dette gjorde diametral maskinering unødvendig. Stengene ble kappet med kappstål i manuell dreiebenk, for så å utføre plandreining av ender, deretter ligge kanten 0,5 mm med 45 graders stål. Til slutt ble guidebolt oppspent med lavt ut-stikk for å maskinere spor for låse-skrue.

Deretter ble de polert med 3M Scotch-brite pussematte med parafinholdig væske, for en god overflate. Det kommer frem fra figur 36 at 4 av guideboltene har større endediameter, dette som følge av en prøvemaskinering for låsespor.

Gjengestang TRx8 ble også kappet ved hjelp av dreiebenk for høyest mulig nøyaktig og av-gradet og plandreid i endene.

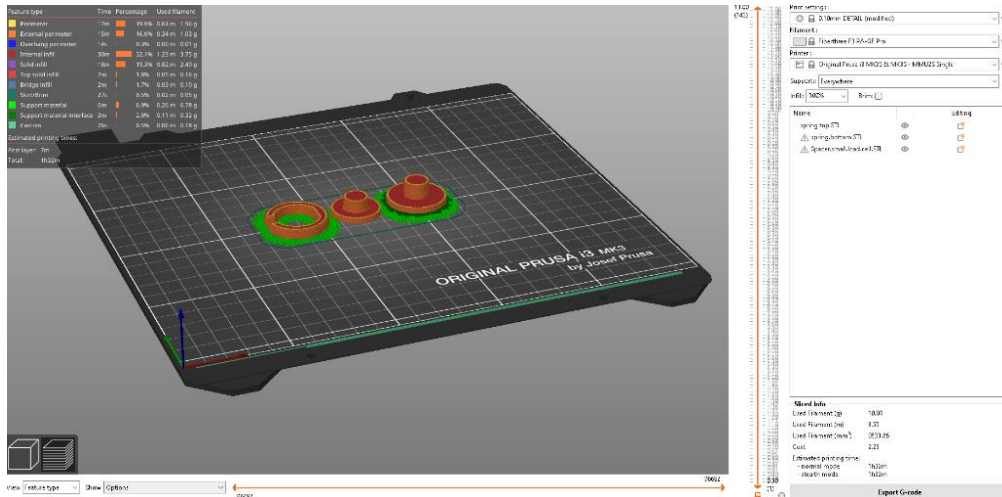
Topp og bunnholder for fjær ble maskinert i dette tilfellet i manuell dreiebenk, som følge av at det ikke var tilgang på CNC-benk. De ble dreid i 2 oppspenninger ved bruk av hensiktsmessig turtall på rundt 600 RPM med erfaringsbasert innstilling av mating. Vanlige verktøy som stikkstål, planstål ble i hovedsak brukt, men en Ø 8mm fres med utkast måtte brukes for lav-profil i ende av øvre fjærholder. Alle grader ble tatt og delene ble kontrollmålt opp mot maskintegning (vedlegg 7).

### 3.4.4 3D print av øvrige maskinelementer



Figur 37: Slicer for SLS

3D printing av rør for beskyttelse av fjær ble gjort på en SLS1Speed printer. Tegningene ble hentet fra Solidworks som STL fil og importert inn i Prusa slicer deretter printet i materialet 3DM- Clear med ekstra fin oppløsning (0,025 mm lag). Materialet har hovedsakelig en minimal støttestruktur i form av dets styrke, men en klar fordel med å være gjennomsiktig, så man kan overvåke de dynamiske komponentenes vertikale linje i riggen. De ble printet vinklet, da direkte-berøring på plate medfører komplikasjoner i forhold til heftelse og fordeling av resin.



Figur 38: Slicer for FFD

Komponentene fjærholdere og senterings ring for lastcelle ble 3D printet ved hjelp av Prusa MK3S printer, ved bruk av GFR PA. Dette var som følge av at komponenter har behov for noe styrke og påkjennes kun trykk-krefter. Fjærholdere ble også maskinert manuelt i materialet 316 L for å utføre tester i samsvar med materialbeskrivelse gitt i maskintegninger (vedlegg 7).

### 3.4.5 Montering prototype

Montering av ferdig produkt startet når alle komponenter var ferdig produserte, kontrollmålt opp mot maskintegning i (vedlegg 7), og klarert i henhold til montasjeanvisning (vedlegg 6) og målebrev (vedlegg 5)

Utstyr brukt i forbindelse med montering:

- Varmeskap 0-400 grader 1000x1000X1200mm samt varmhansker
- CRC -40 Spray
- Håndpresse
- Momentnøkkel 0-5 Nm
- Rettholt
- Loctite 243
- Unbrakonøkkel 3 mm & 2mm

Platene ble lagt i varmeskap og satt til 120 grader (+25 grader for sikkerhetsmargin) i ca. 60 min. Akslinger ble kjølt ned med CRC -40 spray for å sikre en glidepasning ved montering,

da aluminium har høy varmeledningsevne ved direkte kontakt med planbord i håndpresse ved montering. Platene ble så fraktet til presse hvor akslinger lå ferdig nedkjølt, for så å montere ved hjelp av håndpresse. Da det fortsatt var noe motstand måtte håndpresse brukes varsomt for å presse akslingene ned i den plane overflaten for å sikre god og parallell heft med plate. Når kontroll er gjort for planhetsavvik mellom bunnplate og akslingens ende, så monteres M4 settskruer for låsing av bolter med et moment tilsvarende 2Nm.

Deretter ble iglus foringer presset for hånd i mellomplate, for så å kontrollmåles for eventuelle deformasjoner. Det ble utført en enkel brotsjring med justerbar brotsj  $\text{\O} 7-8$  mm for å sikre glideklaring til akslinger.

Deretter ble grafitt-ark montert og riktig heftet for å unngå luftbobler og rifter på bunndel og mellomdel på sidene, som presser mot cellen. Så ble mellomplate montert på guidebolter og kontroll at den fører fint og friksjonsfritt mellom guidebolter og iglus foringer.

Innmat (fjær, celle, holdere, rør og mellomstykke) ble satt på topplate og korrigert på linje. Deretter ble TR8x2 muttere montert til topplate og ratt, for så å feste med M3 skruer med et moment tilsvarende 2 Nm. Gjengestang TR8x2 ble montert i toppdel og toppdel ble deretter montert på guidebolter og låst av med 4 stykk M4 skruer med moment på 2Nm. Til slutt ble rattet festet ved hjelp av Loctite 243 (gjengelåsing) for å sikre gunstig heft mot TR8x2 gjengestang

En mer detaljert og grafisk monteringsanvisning finner man i Vedlegg 6

Det ble så utført flere mekaniske funksjonstester for å etterleve sikkerhetsmargin for kraftpåkjenninger, også for å sikre en riktig og gunstig montasje før videre testing av lastcelle og kalibrering av lastcelle.

## 3.5 Produksjon trykkovervåking

### 3.5.1 Trykkovervåking

#### Bakgrunn

Det finnes en rekke måter å finne ut om trykket endrer seg i riggen, som vil si at batteriet har utvidet seg. En måte er å montere en sensor som måler avstanden fra toppen av riggen til toppen av batteriet. Disse målerne er veldig nøyaktige, men kan bli svært dyre. Bruk av slike sensorer gir ett godt bilde av utvidelsen, men det gir ingen løsning på hvor mye trykk cellen har blitt utsatt for. Utvidelsessensoren sier også bare noe om dette trykket har endret seg, ikke hva det er i utgangspunktet. En løsning på dette kan være å regne ut hvor høyt moment man kan påføre skruen på toppen av riggen, men dette kan fortsatt gi unøyaktigheter, ved for eksempel at ett av guideboltene har mer motstand enn annen. En mer anvendelig løsning kan derfor være å benytte en sensor for å måle nettopp trykket. Det er selvsagt ikke noe i veien for å benytte moment på skruen på toppen, dersom man gjennom testing av flere rigger finner ut at trykksensorene gir samme utslag på flere rigger ved samme moment på skruene. Riggen designes også for å ikke trenge trykksensor dersom man ønsker å bruke moment.

#### Lastceller

En trykksensor eller lastcelle er en type transduser som konverterer en form for energi til en annen. I dette tilfellet er det en innretning som konverterer kinetisk energi i form av trykk over til noe man kan måle med et elektrisk signal (Trent, u.å.b). Videre må dette elektriske signalet måles og bearbeides. Det finnes en rekke utstyr for å måle disse signalene, men det som trengs i dette tilfellet er noe som kan logge signalet over lang tid og som kan plote resultater på en måte som en bruker kan overvåke. Det er et krav fra IFE at resultatene av trykkendringer skal logges slik at de kan analyseres ved et senere tidspunkt.

Lastceller har ulike måle intervall. Det finnes celler som måler trykk fra få gram til ett par kilo, som for eksempel en kjøkkenvekt, til store industrielle lastceller som måler mangfoldige tonn. Den cellen som skal brukes i dette prosjektet har en maksvekt på 200kg. Dette fordi testene som skal kjøres i starten er på under 0,2MPa. Da lastcellenes nøyaktighet minker ved

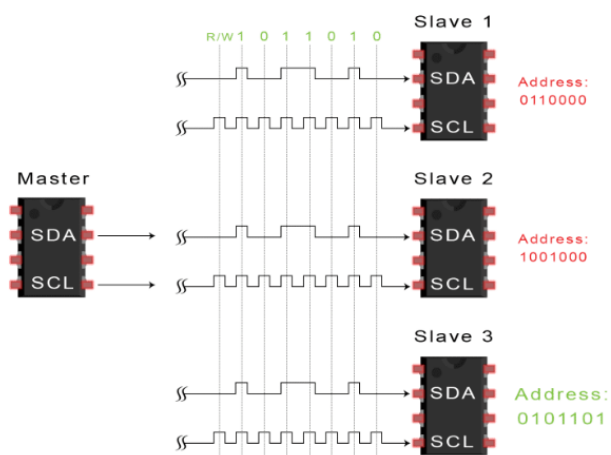


økt målespenn, er det valgt å bruke en celle med målespenn på 0-200kg. Det må legges til rette for også kunne bruke lastceller som måler høyere trykk, da IFE også har større battericeller.

## Kommunikasjon

Spesifikasjonene på lastceller beskrives i cellens datablad. Her finnes informasjon over hvor mye strøm cellen skal kobles opp med og hvordan utsignalet er satt opp. Mange celler har mulighet til å kobles opp med ulik strømsørrelse og dette vil igjen påvirke hvor nøyaktig utsignalet blir. For eksempel kan databladet til en celle oppgi at det elektroniske signalet har en gitt følsomhet oppgitt i mV/V (Trent, u.å.a). Den kan også oppgis i for eksempel 0 – 5V der 0 V er der cellen er ubelastet og 5V der cellen er fullt belastet. I noen tilfeller kan de også komme med I2C digitalt brukergrensesnitt (TE Connectivity, 2020).

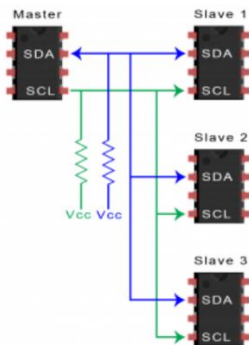
I2C, også kalt Inter-Integrated Circuit, er en protokoll som gjør at elektronisk utstyr kan kommunisere direkte med for eksempel mikrokontrollere som Arduino eller Raspberry Pi. Dette settes opp i en form for master og slave konfigurasjon der en mikrokontroller kan kommunisere med mange sensorer som er koblet sammen. Hver sensor/celle har en egen adresse som er unik for den. Det gjør at mikrokontrolleren kan identifisere hver enkelt sensor og skille de fra hverandre (Campbell, u.å.).



Figur 39: Identifisering av slaver/sensorer ved hjelp av adresse (Campbell, u.å.)

Det som er fordelen med I2C er at det kun kreves to ledninger koblet til mikrokontrolleren. Normalt må hver sensor kobles til en unik inngang på mikrokontrolleren, men med I2C kobles alle sensorer sammen og kobles til slutt til mikrokontrolleren med to ledninger. De to

inngangene kalles SDA og SCL. SDA står for Serial Data og det er her datasignalene overføres. SCL står for Serial Clock som er ett klokkesignal som synkroniserer datastrømmen mellom alle enhetene. Noen I2C sensorer krever at SDA og SCL er koblet til 5V. I tilfeller som dette må det kobles til en motstand på 4,7 K $\Omega$ . Uten denne motstanden vil SDA og SCL signalet alltid være høyt (det vil si 5V), men både mikrokontrolleren og alle I2C sensorene må kunne sette signalet lavt (det vil si 0V) for å kunne kommunisere.



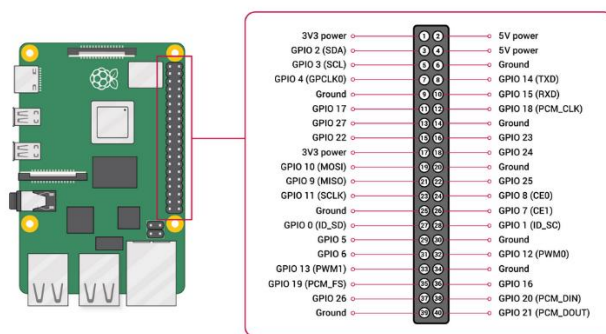
Figur 40: Kobling av SDA og SCL til ekstern strømkilde (Campbell, u.å.)

## Mikrokontollere

Mikrokontrollere finnes i en rekke størrelser og med mange forskjellige bruksområder. For at en bruker skal kunne se hvor høyt trykk batteriet er utsatt for, er det naturlig å bruke en mikrokontroller. Mikrokontroller er en veldig liten datamaskin, som kan tolke signalene fra lastcellen, bearbeide data og presentere de på en måte som en bruker kan forstå.

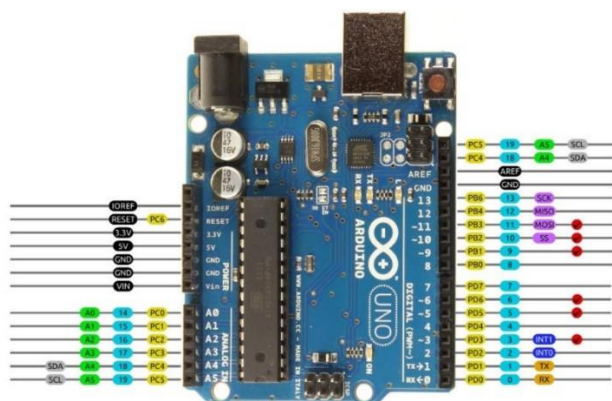
Raspberry Pi er en liten mikrokontroller med dimensjoner på 85,6mm x 56,5mm og kommer i en rekke forskjellige typer. Den kan settes opp med ulike operativsystem som for eksempel Linux. Den kan kobles opp med en tv-skjerm eller monitor og kan kobles til med tastatur og mus. Som en ordinær datamaskin kan den kobles opp til nettet via for eksempel WiFi eller med nettverkskabel (Raspberry Pi Foundation, u.å.b).

Raspberry Pi kan benyttes til programmering og kan blant annet brukes med programmeringsspråket Python. I tillegg har den en rekke inn og ut porter som kan kobles til en rekke forskjellige sensorer, motorer, aktuatorer og lignende. Lastceller kan kobles til denne med I2C protokollen, eller ved at mikrokontrolleren leser 0-3.3V signaler inn (Raspberry Pi Foundation, u.å.a).



Figur 41: PINOUT for Raspberry Pi 4 (Raspberry Pi Foundation, u.å.)

Arduino er også en liten mikrokontroller med dimensjoner på 68,6mm x 53,4mm og kommer i en rekke forskjellige størrelser. Denne skiller seg fra Raspberry Pi ved at den ikke kobles opp med skjerm, mus og tastatur. I stedet programmeres denne ved at programmet lastes opp fra en datamaskin over på mikrokontrolleren via en USB-kabel. Når programmet er lastet over kjøres programmet kontinuerlig så lenge Arduinoen er koblet til strøm. Man trenger med andre ord ikke starte noen programmer manuelt, dette skjer av seg selv så lenge den er koblet til strøm. Den kan på samme måte som Raspberry Pi lese og sende digitale og analoge signaler til og fra sensorer, motorer og lignende. Softwaren som følger med Arduinoen kan programmeres med ett språk som er basert på C++ (Arduino.cc, 2023).



Figur 42: PINOUT for Arduino Uno (The Robotics Back-End, u.å.)

Både Raspberry Pi og Arduino kan konvertere et analogt signal (f.eks. 0-5V) til digitalt signal. Raspberry Pi har en begrensning at maksimal strøm er 3.3V (Raspberry Pi Foundation, u.å.a), mens Arduino kan konverterer fra 0 til 5V. Denne konvertering skjer ved hjelp av en innebygget ADC. ADC står for Analog-to-digital converter (ElectronicsTutorials, u.å.) En ADC oppgis gjerne med en oppløsning gitt i bits. For eksempel en 10 bits eller 12 bits ADC. Dersom strømmen som skal brukes er fra 0-5V og ADCen som skal brukes er 10 bits, vil

oppløsningen fungerer slik at 0 V tolkes som 0 og 5 V tolkes som  $2^{10}$ , som vi si 1024. Målt volt kan derfor leses av som en verdi mellom 0 og 1024. Ved en 12 bits ADC vil maksimal verdi være  $2^{12}$  som er 4096. En 12 bits ADC vil derfor lese av verdien mer nøyaktig da oppløsningen er høyere.

Målet ved å bruke en mikrokontroller og at dette skulle kunne leveres i ett så kompakt format som mulig. I tillegg skal enheten være brukervennlig og gi brukeren den informasjonen han trenger. Valget i dette prosjektet falt på Arduino, da denne type mikrokontroller ikke trenger å startes opp manuelt, men kun kobles til strøm.

### 3.5.2 Elektroniske komponenter

#### Arduino

Arduino kommer i en rekke forskjellige fysiske størrelser og med mange ulike oppsett. Dette er ett utvalg av muligheter:

*Tabell 7:  
Oversikt over ulike Arduino modeller (Arduino.cc, u.å.b).*

Navn	Lengde	Vidde	Minne	Digital I/O	I2C	WiFi	Pris
Arduino Giga	101mm	53mm	2 Mb	76	Ja	Ja	720,-
Arduino Leonardo	68,6mm	53,3mm	32 Kb	20	Ja	Nei	225,-
Arduino Mega	101,52mm	53,3mm	255 Kb	54	Ja	Nei	440,-
Arduino Nano	45,0mm	18mm	32 Kb	8	Nei	Nei	225,-
Arduino Uno	68,6mm	53,4mm	32 Kb	14	Ja	Nei	240,-

Prisen vil selvsagt endre seg i forhold til dagens kurs, men tabellen viser prisen kontrollene har i forhold til hverandre.

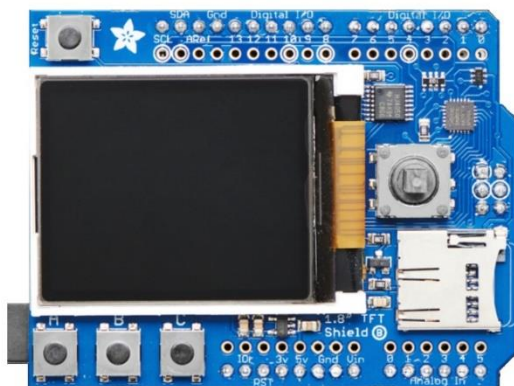
Valget av Arduino var i utgangspunktet Arduino Uno, da denne var forholdsvis liten, men samtidig hadde plass på toppen til å montere et shield, som er en type ekspansjonskort som kan monteres på toppen av Arduinoen, som ikke krever noen form for lodding. Disse ekspansjonskortene er lette å sette på og billige å produsere (University Libraries, 2022). Utfordringen startet under programmering av Arduinoen. Arduino Uno har en maksimal minnekapasitet på 32Kb. Etter at alle bibliotek og kode var lastet opp til minne under

programmeringen, viste det seg raskt at 32Kb var for lite. En mulighet var å effektivisere koden, eller bruke en Arduino med mer minne.

## Skjerm og SD-kortleser

Det er ønskelig at brukeren skal kunne se informasjon fra lastcellen i hver rigg. Dette kan løses ved å bruke en skjerm med Arduinoen. Det skal også være mulig å hente ut en logg over data med ett SD kort. Det som behøves da er en skjerm som kan kobles opp, ett utvidelseskort med SD kort leser, eller en kombinasjon som har begge delene.

Valget falt på et breakout shield med både skjerm og SD-kort leser. 1,8 '' TFT Display Breakout And Shield fra Adafruit, har en skjerm med 128x160 pixler og en microSD-kortleser. I tillegg har den en rekke knapper og en joystick som kan programmeres.



Figur 43: 1,8'' TFT Display Breakout and Shield fra Adafruit (Adafruit, u.å.a.)

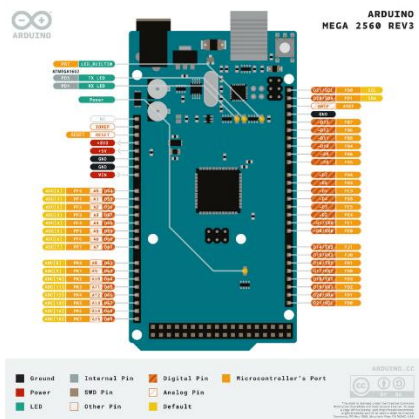
## Loggføring av data

Arduino har en svakhet i det at den ikke er presis ved logging av tid over lengre perioder. I tillegg til dette vil den miste oversikten over tiden dersom strømmen blir borte. Loggføring av tid er ganske viktig med tanke på at det er ønskelig å logge tidspunktet hvor trykket i cellen endrer seg. Løsningen på dette er å bruke ett tilleggs kort kalt en RTC. Dette står for Real Time Clock. Dette er ett kort med innebygget klokke og plass til ett lite batteri. Med andre ord vil den ikke miste oversikten over tiden selv om strømmen fra Arduinoen blir borte. Valget falt på DS1307 RTC breakout board fra Adafruit. Dette kortet kommuniserer over I2C protokollen.



Figur 44: DS1307 Real Time Clock fra Adafruit (Adafruit u.å.b.)

Arduinoen som hadde vært brukt fram til nå var en Arduino Uno, men uforutsette problemer oppsto da RTC kortet skulle kobles opp. TFT shielden bruker blant annet pin 4 for datakommunikasjon. Dette er tilfeldigvis også sammen pin som brukes til SDA i I2C protokollen. På noen skjema så er også pin 18 satt som SDA på Arduino Uno, men disse er egentlig koblet sammen med 4, så det fungerte heller ikke. Det som skjedde var at så fort RTC kortet ble slått på, så kuttet skjermen på shielden. Løsningen ble derfor å bruke Arduino Mega, da denne har SDA på pin 20. Dette løste også de utfordringene som hadde oppstått på grunn av for lite minne.



Figur 45: Pinout Arduino Mega (Arduino.cc u.å.b)

## Valg av lastcelle

Lastcellene som ble vurdert var en lastcelle fra TE Connectivity, av type FX29 og en lastcelle fra Sparkfun, av type TAS606. Lastcellen fra TE Connectivity er en celle som er relativt billig og tar lite plass. Den kommer med muligheter for dataoverføring ved I2C protokollen, ved analog utgang fra 0.5-4.5V eller fra 16-24 mV/V. Den kan også måle vekt opp mot 200lb som er ca. 90kg. Det er også greit å spesifisere at cellen er tilnærmet lineær med tanke på signalene ut i forhold til målt vekt. Usikkerheten er  $\pm 1\%$  av pålagt kraft (TE Connectivity, 2020). Lastcellen fra Sparkfun er en del dyrere og har dataoverføring ved analog utgang fra 1-2

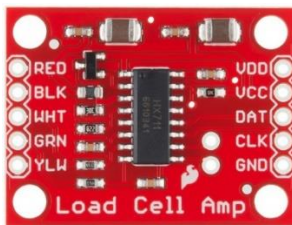
mV/V. Den kan måle vekt opp mot 200kg. Usikkerheten er  $\pm 0.3\%$  av pålagt kraft (HT Sensor Technology CO., Limited, u.å.)

Det som hadde vært ønskelig var å få tak i en celle med I2C protokoll, da denne hadde vært enklest å koble opp.

### Ekstern ADC

Det viser seg at Arduino ikke nødvendigvis er veldig nøyaktig å bruke for å lese mV/V. Løsningen på dette er da å bruke en ekstern signalforsterker. Valget falt på Load Cell Amplifier – HX711 fra Sparkfun (Sparkfun.com, u.å.c). Dette er en 24-bit ADC, som gir en oppløsning på  $2^{24}$  som vil si  $16,8 * 10^6$ . Den er også spesiallaget for lastceller.

Usikkerhet i målingen ved dette kortet sammen med lastcellen kan være  $\pm 5\%$  på grunn av temperatur, elektronisk støy og lignende. Mye av dette kan reduseres ved at lastcellen kalibreres ved hjelp av å programmere Arduinoen (Sparkfun.com, u.å.a).



Figur 46: Load Cell Amplifier – HX711 fra Sparkfun.com (Sparkfun.com, u.å.).

### 3.5.3 Pseudokode og kildekode

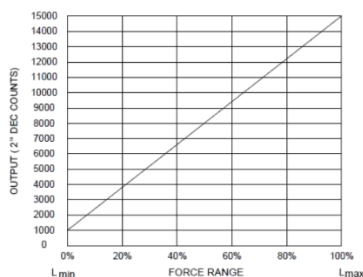
For at Arduinoen skal kunne utføre de oppgavene som trengs måtte den programmeres. Den programmeres i et språk som ligner C++. For å programmere og kommunisere med Arduinoen brukes en ordinær USB kabel som kobler en PC til Arduinoen. Programmet som brukes til programmeringen heter Arduino IDE og det kan også plote data Arduinoen sender over kabelen. Alle detaljer angående selve programmeringen ligger som vedlegg.

- Klassediagram som inneholder en oversikt over alle klasser med metoder og funksjoner ligger som vedlegg 1.
- Visuelt koblingsskjema av Arduino, RTC, ADC og lastcelle ligger som vedlegg 2.

- Pseudokode er en form for kode som er slik at selv om man ikke kan programmere i det språket som er benyttet, eller ikke kan programmere, skal man kunne forstå hvordan programmet skal fungere. Denne koden ligger som vedlegg 8.
- Komplet programeringskode ligger som vedlegg 9.
- Brukermanual for styringsenhet ligger som vedlegg 10.
- Brukermanual for behandling av loggført data ligger som vedlegg 11.
- Veiledning for programmeringen ligger som vedlegg 12.

### 3.5.4 Usikkerhet ved målinger

Det finnes en rekke faktorer som kan påvirke avlesning av trykket. Som nevnt tidligere har load cell amplifiser kortet og lastcellen som kobles til opp mot  $\pm 5\%$  usikkerhet med tanke på målingene. Mye av dette skal være mulig å redusere ved å kalibrere cellen. I tillegg har cellen fra TE Connectivity  $\pm 1\%$  usikkerhet med tanke på lineariteten, mens cellen fra Sparkfun har  $\pm 0.3\%$ . Tabell 47 viser lineariteten for lastcellen fra TE Connectivity (Te Connectivity, 2020)



Figur 47: Grafisk oversikt over lineariteten for lastcellen fra TE Connectivity.

Det er også ønskelig å finne ut hvor lite vekt som trengs for å kalibrere cellen. Dersom det ønskes ett trykk tilsvarende 60kg på cellen ved start av batteritesten, er det mest nøyaktige å kalibrere med en vekt på 60kg. Dette er dessverre lite hensiktsmessig rett og slett på grunn av vekten.

Load cell amplifiser kortet som lastcellen er koblet til i tillegg til programmeringskoden som er laget, oppgir to faktorer som brukes til kalibrering. Den ene kalles nullfaktor og det er faktoren eller verdien cellen programmeres med når den er helt avlastet. Det er for cellen skal vite hvor nullpunktet er. Den andre er en kalibreringsfaktor som kan settes når cellen er



belastet, slik at riktig vekt vises. Det som er ønskelig er at den kalibreringsfaktoren stemmer og er tilnærmet lik uansett hvilken vekt som legges på. Ta følgende eksempel. Si at cellen belastes med 5kg og kalibreringsfaktoren cellen programmeres med er 1000, slik at det avgir verdien 5,00. Dersom cellen belastes med 60kg så skal cellen gi fra seg verdien 60,00 dersom den er programmert med faktoren 1000. Dersom disse to faktorene er like, så kan cellen like godt kalibreres med 5kg som 60kg, og det skal den egentlig være ettersom cellen skal være ganske lineær. For å sjekke dette må det gjennomføres en rekke forsøk der cellen belastes med 60kg en rekke ganger. Faktoren må regnes ut og det må sjekke hvor stort målingsspenn det er i resultatene og få ut en gjennomsnittsfaktor for 60kg. Deretter må det forsøkes med mindre vekt for eksempel 30kg. Gjennomsnittsfaktoren bør være ganske lik som den var for 60kg. Målet er å finne ut hvor lite vekt det kan kalibreres med og likevel få tilnærmet lik faktor. Håpet er at kalibrering med en vekt på rundt 5kg skal være nok.

Beskrivelsen av gjennomføring av testen og hvilken type utstyr som ble brukt under testen ligger som vedlegg 13. Resultatene av testen finnes i kapittel 4.2.1.

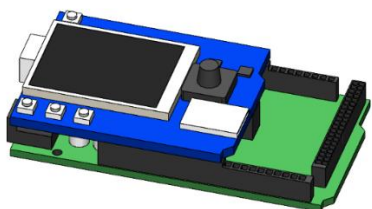
## **3.6 Beskyttelse av elektronisk utstyr**

### **3.6.1 Boks for Arduino**

Elektronisk utstyr bør beskyttes fra omgivelsene med tanke på fuktighet, støv og lignende. Både Arduinoen og shielden har kretskortet eksponert, så det måtte designes en boks (111,0mm x 69,3mm x 28,0mm) som kan beskytte disse, og i tillegg sørge for at ledninger ikke dras ut eller løsner fra kretskortet eller kontakter. Kortslutninger eller dårlig kontakt kan føre til unøyaktige avlesninger. Tanken var å 3D-modellere Arduino, shield og RTC slik at en boks kan designes rundt denne. Denne boksen kan deretter 3D-printes. Temperaturen Arduino kan brukes under er fra -40 til 85<sup>0</sup>C, så materialet som skal 3D-printes må tåle minst 85<sup>0</sup>C. Valget faller da på å lage prototype i PLA (Polyactic acid). PLA er en termoplast som har et smeltepunkt på rundt 165 C, men har GT temperatur på 55<sup>0</sup> C. Det betyr at PLA kan starte å deformeres dersom temperaturen når denne grensen (Anderson, Farah og Langer, 2016). PLA lages blant annet fra plantestivelse og kan komposteres etter bruk, så det er nok et mer miljøvennlig valg. (Boissonneault, 2022) Dersom temperatur blir et problem, kan boksen

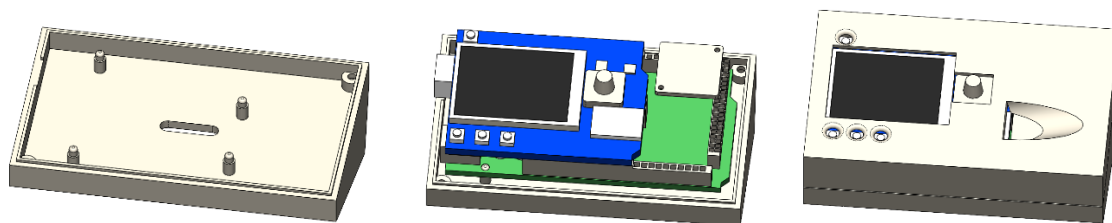
printes i for eksempel PET (Polyethylene Terephthalate). PET har en GT-temperatur på 70-78<sup>0</sup> C og en smeltetemperatur 200-260<sup>0</sup> C (Matweb.com, u.å).

For å designe boksen som Arduinoen skal være i, må Arduinoen og shielden 3D-modelleres med riktige mål og dimensjoner. Disse kan mest sannsynligvis lastes ned fra diverse sider med ferdige modeller, men i dette prosjektet kreves det bare at dimensjonene er riktige, ikke at alle små detaljer må være med.



*Figur 48: Arduino Mega med shield ferdig modellert*

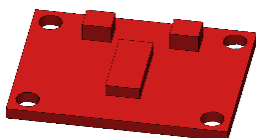
Etter at modellen er ferdig, kunne boksen modelleres:



*Figur 49: Ferdig modellert boks tilpasset Arduino Mega og shield.*

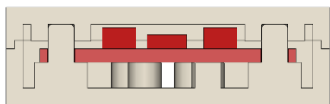
### 3.6.2 Boks for ADC

ADC kortet må også beskyttes med en boks. En grov modell av kortet ble modellert slik at dimensjonene var riktige.

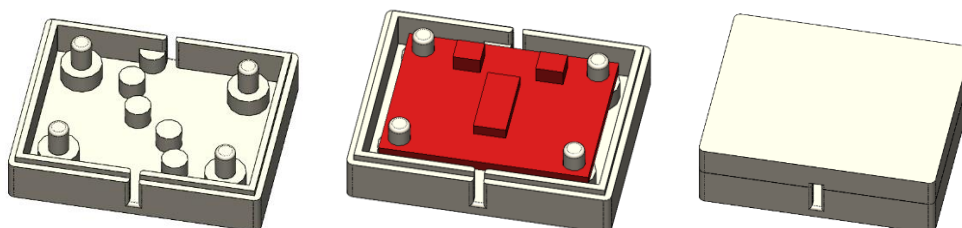


*Figur 50: 3D modell av ADC kortet*

Deretter kunne boksen modelleres. Etersom boksen er liten og veier lite er den kun festes sammen med friksjon. Det vil si at ved å sette tappene som stikker opp fra bunnen med en diameter på 3mm og hullene i toppen på 3mm er dette nok til at den ikke åpner seg.



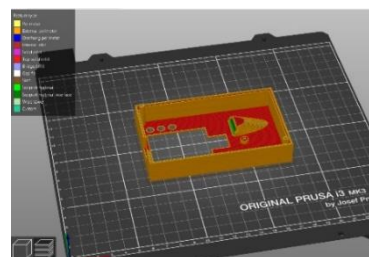
Figur 51: Snitt som viser hvordan delene er festet sammen.



Figur 52: Ferdig sammensatt boks

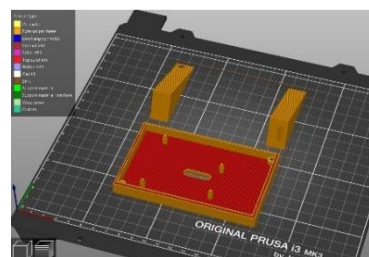
### 3.6.3 3D-printing av bokser for Arduino og ADC

Målet med printingen var å printe boksene med minimalt bruk av support og for å få en så pen overflate som mulig. Materialet som ble testet først var PLA med 15% infill. Toppdelen av boksen ble printet med oversiden ned mot platen med support der hvor SD kortet skal sitte.



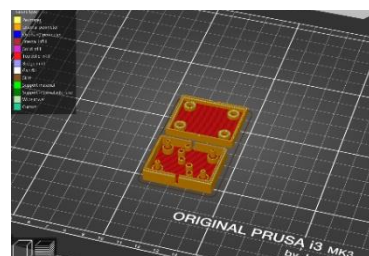
Figur 53: Topp av boks

Bunnen ble delt i to slik at føttene printes separert. Dette gjøres igjen for å bruke minimalt med support. Føttene kunne siden limes på med superlim.



Figur 54: Bunn av boks

ADC boksen ble printet slik at toppen av toppdelen og bunnen av bunndelen printes ned mot platen.



Figur 55: ADC-boks

## 3.7 HMS

Produksjon og tilvirkning av deler ble utført i Equinor sine lokaler og da ble lokalt verneregime ivaretatt, som igjen omfatter verneutstyr tilpasset arbeidsoperasjon. Sikkerhetsinstruksjoner for hver individuell maskin ble også etterlevd til enhver tid. Det ble ikke registrert noen hendelser eller forløp til hendelser under produksjon og tilvirkning av komponenter.

Under maskinering ble det brukt dobbelt-hørselsvern, tette vernebriller, vernesko, kutt-sikre hansker og ekstra støymatter i produksjonslokalet. Det inngår også i HMS og til enhver tid holde en oversiktlig og ryddig arbeidsplass, hvor man spør fagarbeidere underveis i prosess ved usikkerhet knyttet til arbeidsoperasjon.

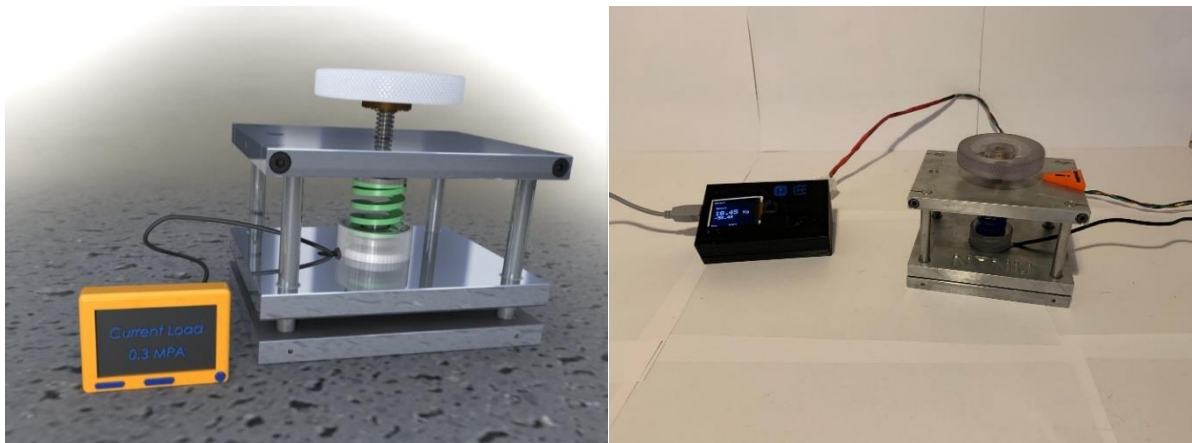
Ved 3D-print ble printerens egen håndbok for sikkerhet etterfulgt, dette spesielt ved tanke på bruk av Resin, hvor man skal unngå innånding og direkte kontakt. Dette ble løst ved bruk av kjemikalieresistente engangshansker og enkelt åndedrettsvern.

Ved montering ble lokalt reglement ivaretatt ved bruk av vernesko, vernebriller, varmhansker og kutt-sikre hansker brukt. Det er selvsagt en klemfare ved montering som må ivaretas underveis i montering, og faremomenter knyttet til feil verktøybruk. Følger man montasjeanvisning i vedlegg 5 så vil dette bidra til en risikofri montering, gitt at man tar forhåndsregel i forhold til personlig verneutstyr og bruk av kjemikalier, samt har datablad for kjemikalier tilgjengelig om uhell skulle oppstå.

# 4 Resultater

## 4.1 Endelig løsning

Det endelige designet ble en forholdsvis kompakt rigg, tatt i betraktning kravet fra IFE om en rigg som tar minst mulig plass, og som samtidig ivaretar tiltenkt funksjon. Riggens ble utstyrt med lastcelle for overvåking av trykk, samt en skjerm for avlesning av det aktuelle trykket cellen blir utsatt for. Riggens er solid og kompakt samtidig som den består av relativt få komponenter, noe som gir en rigg med få feilmoder og lav sannsynlighet for svikt.



*Figur 56: Ferdig design og prototype*

### Fysisk størrelse

Riggens grunnflate er på 95mm x 125mm, og høyden er fleksibel. På prototypen er den 105mm. Den endelige versjonen fikk ganske mye mindre grunnflate pga. tynne IGUS lager (10mm ytre diameter), som gjorde det mulig å ha guideboltene nært kanten av platene, samt nært battericellen.

### Fleksibilitet i designet

Det er mange muligheter for å endre riggen etter eventuelle endringer i behov. Fjæren er en standard verktøyfjær, og kan byttes uten noen endringer til 3 forskjellige stivheter avhengig av hvor mye trykk som er ønsket. Samtidig er lavere fjærkonstant gunstig for å holde trykket

mest mulig stabilt på tross av battericellens potensielle utvidelse. Fjærene finnes også med samme indre og ytre diameter, med forskjellige lengder. Om det skulle være ønskelig kan det også kjøres tester helt uten fjær.

Makshøyden for battericellen kan også justeres forholdsvis enkelt. Det eneste som må gjøres er å endre lengden på guideboltene i hjørnene. Er det ønskelig med mye lavere fjærkonstant eller høyere trykk må fjæren bli lengre. Som nevnt finnes fjærene med samme indre og ytre diameter, så dette kan også løses enkelt med lengre guidebolter og skrue. Begge disse delene er standard deler som er enkle å få tak i og bearbeide.

Plassen til lastcellen er laget til å passe de største standard typene, og man kan få kjøpt lastceller på flere tonn i det formatet. For å bruke mindre lastcelle legges det bare inn en 3D-printet avstandsstykke, slik som det er gjort i prototypen.

## **Operasjon**

Operasjon av riggen gjøres ved å plassere cellen mellom de to nederste platene. Platene er 5mm lengre enn delen av cellen som skal stå under trykk, dette for å gjøre det lett å få cellen på plass, og samtidig minke sannsynligheten for å få kontakt mellom polene på batteriet og riggen. Når battericellen er plassert skrues håndtaket til ønsket trykk. Det er 2mm gjengestigning på skruen, noe som gir god kontroll over hvor mye trykk battericellen blir utsatt for. Da den trapesgjengede skruen også har mye mindre variasjon i friksjon og mye bedre overflatekvalitet enn en vanlig m10 skrue er operasjonen veldig jevn og fin.

Kalibrering av lastcellen kan gjøres på flere måter. Da riggen skal brukes i en lab hvor de har tilgang på nøyaktig måleutstyr er det mest nærliggende å bruke en annen lastcelle på plassen til battericellen, og så kalibrere lastcellen i riggen ved hjelp av den. Ellers kan skruen tas helt ut av topplaten, erstattes med en rundbolt på 6mm som passer gjennom mutteren. Så kan det påføres en gitt vekt på rundt 5kg (Se kapittel 4.2.1) på denne. Det kan være effektivt å få laget en slik vekt hvis mange rigger skal kalibreres.

## 4.2 Testing

### 4.2.1 Testing av lastceller

#### Testing av lastcellen fra Sparkfun

Lastcellen fra TE Connectivity var billigere en cellen fra Sparkfun, men cellen fra Sparkfun har høyere maksimalbelastning og var mer nøyaktig. Valget falt derfor på bruk av cellen fra Sparkfun. En test av cellen fra TE Connectivity er likevel vedlagt som vedlegg 3.

Testen av lastcellen fra Sparkfun ble delt opp i tre delmål:

1. Finn nullfaktor og kalibreringsfaktoren ble ved stadig høyere vekt og finn ut om de er jevne. Bakgrunnen for dette målet var å sjekke om målingene var lineære, noe det var kritisk at de var.
2. Hvor lav vekt kan benyttes for å kalibrere cellen. Bakgrunnen var at det er ønskelig å kunne kalibrere lastcellen med så lite vekt som mulig og likevel få riktig resultat. Mindre vekt er mer håndterlig for en bruker.
3. Hvor stor innvirkning nullfaktor har på høy og lav vekt og hvor stor innvirkning kalibreringsfaktoren har på høy og lav vekt. Bakgrunnen for denne testen var å finne ut om begge faktorene var like viktige.

#### Første deltest

Mål: Finn nullfaktor og kalibreringsfaktoren ble ved stadig høyere vekt og finn ut om de er jevne.

Fremgangsmåte ved den første delen var at først ble nullfaktor funnet ved avlastet vekt. Deretter ble kalibreringsfaktoren funnet ved at den kalibreres med en pålagt vekt. Vekten som ble brukt under forsøkene var 5,55kg, 10,45kg, 20,5kg, 30,38kg, 40,45kg, 50,35kg, 60,2kg og 70,25kg. Under forsøk nummer en ble cellen kalibrert med 5,55kg og deretter ble den belastet med alle de åtte vektene i tur og orden. Avlest vekt ble så logget og avviket regnet ut. Det ble utført i alt åtte forsøk med alle tilgjengelige vekter.

Det bør også nevnes at cellen kan påvirkes av svinging. I databladet for cellen er det oppgitt sig på  $\pm 0,1\%$  av pålagt last de første 30 minuttene (HTC-Sensor.com u.å.). Det er ikke tatt hensyn til i denne testen, ettersom utslaget vil være ganske minimalt.

Tabell 6 gir en oversikt over alle forsøkene og alle resultat ligger i vedlegg nummer 4. Både nullfaktor og kalibreringsfaktor er tatt med, i tillegg til hvilke vekter som viste størst og minst utslag i forhold til forventet vekt.

*Tabell 8:  
Oversikt over minste og største avvik og i tillegg hvor mye avvik det er i kg.*

Forsøksnr	Belastet vekt	Nullfaktor	K. faktor	Minst avvik ved:	Avvik	Størst avvik ved:	Avvik
1	5,55 kg	1884	13400	5,55 kg	0,00	30,85 kg	-0,59
2	10,45 kg	1937	13350	5,55 kg	0,02	60,20 kg	0,37
3	20,5 kg	1938	13350	5,55 kg	-0,01	70,25 kg	-0,36
4	30,85 kg	2426	13150	30,85 kg	0,04	70,25 kg	1,90
5	40,45 kg	3015	13350	5,55 kg og 10,45 kg	0,05	30,85 kg	-0,31
6	50,3 kg	3333	13350	5,55 kg	0,03	40,45 kg	0,33
7	60,2 kg	3453	13350	60,20 kg	0,01	40,45 kg	0,29
8	70,25 kg	3645	13350	5,55 kg	0,02	30,85 kg	-0,49

Forsøk fire skiller seg ut ved at avviket ved 70,25kg ble 1,9kg mer enn forventet. Her er også kalibreringsfaktoren lavere enn ved de andre forsøkene. Mye kan tyde på at det er den menneskelige faktoren som spiller inn her. Det kan være at u-bolten har kommet inntil bordet eller inntil cellen slik at riktig faktor ikke har blitt stilt inn. På de resterende forsøkene er faktoren tilnærmet det samme, så mye kan tyde på at forsøk fire ikke er representativt i forhold til hvordan cellen presterer.

Cellen har en usikkerhet på  $\pm 0,3\%$  av pålagt last (HT Sensor Technology CO., Limited, u.å.) og ADC kortet har en usikkerhet på  $\pm 5\%$  sammen med lastcellen. (Sparkfun.com, u.å.c).

Tabell 7 viser hvor nærme resultatene optimalt sett kan forventes å være og hvor nærme forsøkene var.



Usikkerheten er regnet ut slik:

$$\frac{\text{Pålagt vekt} * \pm 0,3}{100}$$

Tabell 9:

Resultatene viser avvik ved hvert forsøk og det største avviket i %

Forsøksnr	Størst avvik ved:	Avvik	$\pm 0,3\%$	Faktisk usikkerhet
1	30,85 kg	-0,59	-0,12	-1,91
2	30,85 kg	-0,27	-0,12	-0,88
3	30,85 kg	-0,23	-0,12	-0,75
4	50,35 kg	1,40	0,15	2,78
5	30,85 kg	-0,31	-0,12	-1,00
6	40,45 kg	0,33	0,12	0,82
7	20,50 kg	0,17	0,06	0,83
8	30,85 kg	-0,49	-0,12	-1,59

Avviket er en del over de  $\pm 0,3\%$  som er oppgitt som cellens maksimale nøyaktighet. Andre faktorer som kan spille inn er temperatur, svinging, menneskelig feil ved bruk også videre. Håpet var at den faktiske usikkerheten skulle ligge lavere enn  $\pm 5\%$  som blir oppgitt som maksimal usikkerhet ved bruk av ADC kort og lastcelle. Forsøksnummer fire var det forsøket som hadde størst negativt utslag, men til og med verdiene for dette forsøket er innenfor forventet usikkerhet. Det kan også være interessant å legge merke til at det er ved vekten 30,38kg det er størst utslag i prosent i fem av åtte forsøk. Det var den vekten forsøk 4 ble kalibrert med, der kalibreringsfaktoren ble lavere enn normalen.

### Andre deltest

Mål: Hvor lav vekt kan benyttes for å kalibrere cellen.

Den andre delen av forsøket gikk ut på å finne ut hvor liten vekt man kan kalibrere cellen med og likevel få en kalibreringsfaktor som fungerer også ved høyere vekter. Fremgangsmåten var å finne først nullfaktoren ved avlastet celle og så finne kalibreringsfaktoren ved belastet celle. Vektene som ble brukt under forsøket var 1, 2, 3 og 4kg. Faktoren det siktes inn på er 13350 ettersom det er den faktoren som opptrer oftest i de forrige forsøkene slik tabell 4 viser. Hver vekt ble kalibrert fem ganger og gjennomsnittet ble regnet ut.

Tabell 10:  
Test for å finne lavest kalibreringsvekt

Forsøksnr.	Belastet vekt	Gjennomsnittlig nullfaktor	Gjennomsnittlig kalibreringsfaktor
9 til 13	1,0 kg	3565,0	13500
14 til 18	2,0 kg	3541,5	13400
19 til 23	3,0 kg	3547,2	13380
24 til 28	4,0 kg	3493,6	13350

Fra forsøk en til åtte steg nullfaktoren jevnt og trutt og jevnet ut rundt ca. 3500 ifølge tabell 6. Fra forsøk ni til tjueåtte stabiliserte nullfaktoren seg rundt samme verdi. Det kan muligens tyde på at det var en form for innkjøringsperiode. Det kan også være fordi cellen nok ble utsatt for en del støt ved pålegg av de aller tyngste vektene ettersom de begynte å bli uhåndterlige.

Gjennomsnittlig kalibreringsfaktor stabiliserte seg ved rundt 4,0kg. Dette er den eneste av de fire belastningsvektene der kalibreringsfaktoren ble 13350 i alle fem forsøkene.

### Tredje deltest

Mål: Hvor stor innvirkning nullfaktor har på høy og lav vekt og hvor stor innvirkning kalibreringsfaktoren har på høy og lav vekt.

Den laveste nullfaktoren som ble registrert i forsøket var 1884 og den høyeste 3635. Den laveste kalibreringsfaktoren som ble registrert i forsøket var 13150 og den høyeste 13650. En laveste vekten som ble brukt i forsøket var 5,55kg og den høyeste 70,25kg. For å teste ut alle kombinasjoner trengs  $2^3$  forsøk. Nullfaktoren og kalibreringsfaktoren ble satt manuelt ved at verdiene ble skrevet til filene på SD kortet ved å åpne de i notepad.

Nullfaktorens påvirkning på vekt ga følgende resultater:

Tabell 11:  
Nullfaktorens påvirkning på resultatet

Forsøksnr	Nullfaktor	K. faktor	Belastet vekt	Avvik	Faktisk usikkerhet	Differanse
29	1884	13150	5,55 kg	0,22	3,96	
31	3645	13150	5,55 kg	0,07	1,26	2,70
30	1884	13650	5,55 kg	0,03	0,54	
32	3645	13650	5,55 kg	-0,11	-1,98	2,52
33	1884	13150	70,25 kg	1,29	1,84	
35	3645	13150	70,25 kg	1,15	1,64	0,2
34	1884	13650	70,25 kg	-1,35	-1,92	
36	3645	13650	70,25 kg	-1,45	-2,06	0,14

Resultatet tyder på at nullfaktoren har mer å si for lave vekter enn høye vekter. Differansen er mellom 2,52 og 2,70% ved 5,5kg ved sammenlignbare kalibreringsfaktorer. Den er mellom 0,14 og 0,2% ved 70,25kg ved sammenlignbare kalibreringsfaktorer.

Tabell 12:  
Kalibreringsfaktorens påvirkning på resultatet

Forsøksnr	Nullfaktor	K. faktor	Belastet vekt	Avvik	Faktisk usikkerhet	Differanse
29	1884	13150	5,55 kg	0,22	3,96	
30	1884	13650	5,55 kg	0,03	0,54	3,42
31	3645	13150	5,55 kg	0,07	1,26	
32	3645	13650	5,55 kg	-0,11	-1,98	3,24
33	1884	13150	70,25 kg	1,29	1,84	
34	1884	13650	70,25 kg	-1,35	-1,92	3,76
35	3645	13150	70,25 kg	1,15	1,64	
36	3645	13650	70,25 kg	-1,45	-2,06	3,7

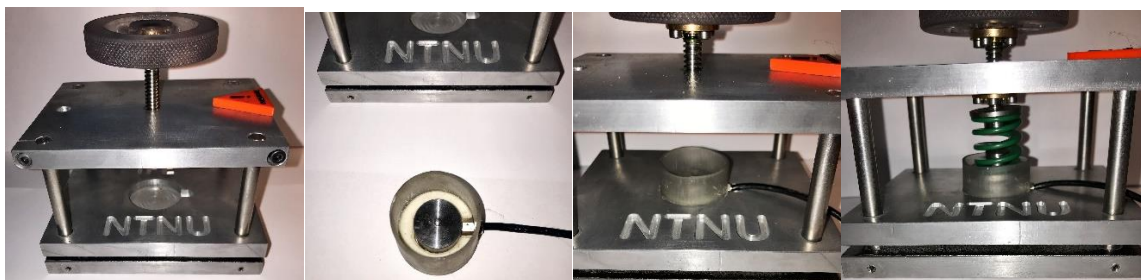
Resultatet tyder på at kalibreringsfaktoren har mye å si uansett vekt. Prosentpoengdifferansen er over 3 i alle tilfellene.

## 4.2.2 Mekanisk testing

Målene med de mekaniske testene var å finne ut:

- Passer lastcellen i testjiggen og hvor enkelt er den å montere.
- Kan nødvendig trykk på lastcellen oppnås ved kun håndkraft og hvor nøyaktig kan trykket økes.
- Vil fjæra glipe av lastcellen under belastning eller komme bort i kanten av hylsen.
- Kan strøm gå fra batteriet over i testjiggen.

**Passer lastcellen i testjiggen og hvor enkelt er den å montere.**



*Figur 57: Ferdig montert testjigg*

Holderen til lastcellen var litt trang og måtte pusses ned litt. Så snart den var pusset, gikk monteringen greit. Det var ikke nødvendig å skru av toppen av testjiggen for å montere cellen.

**Kan nødvendig trykk på lastcellen oppnås ved kun håndkraft og hvor nøyaktig kan trykket økes.**

Skruen var litt kort så det maksimale trykket som kunne settes var rundt 70kg før den nådde bunnpunktet. Kraften som trengtes for å sette trykket var ganske lite og det gjorde at det var forholdsvis enkelt å fininnstille trykket.

**Vil fjæra glipe av lastcellen under belastning eller komme bort i kanten av hylsen.**

Ved å holde fjæra sentrert over lastcellen med to fingre (sørge for at fjæra ikke vrir seg), holdt den seg helt stabilt og kom ikke inntil koppen.

**Kan strøm gå fra batteriet over i testjiggen.**

For å teste dette ble ett multimeter med ringefunksjon brukt. Det var ingen forbindelse mellom den midterste og den nederste delen av platen. Dette betyr at mattene som ligger mellom platene ikke fører strøm mellom platene. Det må nevnes at ved å teste overflaten til mattene viser det seg at selve overflaten leder strøm, men den leder ikke strøm igjennom mattene. Det betyr at strøm ikke vil gå fra batteri til testtrigg, men det hindrer ikke kortslutning av batteriet.

### **Videre testing**

På grunn av begrensninger når det kommer til tid, da dette prosjektet har en kort tidsramme, så har riggen ikke blitt testet over lengre perioder. IFE må selv stå for videre testing.

IFE har planlagt å kjøre i gang pilot for testing av battericeller i løpet av sommeren 2023, og det er derfor viktig de gjennom testing av rigg kan gi tilbakemelding på følgende punkter:

- Klarer riggen å holde jevnt trykk over lange perioder uten manuell justering.
- Klarer brukere som ikke kjenner riggen å sette opp ett forsøk med å bruke brukermanualene.
- Må alle riggene kalibreres, eller er cellene såpass like at man kun trenger å skrive inn kalibreringsfaktorene i filene på SD-kortet dersom denne faktoren allerede er kjent.
- Hvordan håndterer Arduinoen logging og styring av flere rigger.
- Om de oppdager feil knyttet direkte til komponenter, eller har forbedringsforslag knyttet mot montering og selve designet av riggen.

# 5 Diskusjon

## 5.1 Innkjøpskostnad

Det er her en beskrivelse direkte kostnadene ved produksjon av prototype. Det må tas i betraktning at disse kostnadene ikke på noen måte gjenspeiler det reelle kostnadsbildet av en faktisk produksjon. Det er kun direktebestilt materiell uten kostnader knyttet til bearbeidelse som man kan anse som reelle kostnader. Det vil kreve en grundig analyse av alle komponenter i markedet med tanke på innkjøp, noe som også vil gi muligheter for prisreduksjon på direktebestilt materiell. Dette vil også skape en diskusjon i ettertid for enkelte komponenter som eventuelt kan erstattes av billigere tilsvarende komponenter.

### 5.1.1 Direkte kostnader

Totalkostnad knyttet til utvikling av prototype og produkt har gruppen selv dekket med private midler. Maskiner og utstyr har man hatt tilgang på privat eller gjennom arbeidsplass. Gruppen ble på forhånd enig om at gruppen tar kost knyttet til prosjektet, av noen inngående komponenter som «soft-parts» ble det bestilt inn ekstra sett med komponenter som følge av mulige design-endringer eller komplikasjoner knyttet til produksjon, montering og testing. Det er ikke estimert timepris eller timeantall i forbindelse med design, programmering, produksjon eller montering. For estimerte kostnader knyttet til ferdig produksjon se kapittel 2.3

Følgende direkte kostnader er knyttet til oppgaven:

Tabell 13:  
Innkjøpskost materiell

Innkjøpskost for prosjekt					
Element nr	beskrivelse	Kostnad NOK	Antall	Totalkostnad	Hvem
1	Aluminiumsplate	1207,00	1	1207,00	Henrik
2	Resin	1260,00	1	1260,00	Henrik
3	Polymer	600,00	1	600,00	Henrik
4	Loctite	178,00	1	178,00	Henrik
5	Grafitt plater	112,00	6	672,00	Henrik
6	Iglus lager	119,00	4	476,00	Henrik
7	M4 og M3 skruer	2,00	50	100,00	Henrik
8	TR8X2 mutter og skrue	160,00	4	640,00	Henrik
9	Arduino Mega	549,00	1	549,00	Eivind
10	Arduino Uno	350,00	1	350,00	Eivind
11	TFT Shield	465,00	1	465,00	Eivind
12	ADC	199,00	3	597,00	Eivind
13	Koblingsbrett m/ledninger	171,00	1	171,00	Eivind
14	FX29 load cell	322,00	1	322,00	Eivind
15	TAS606 load cell	883,00	1	883,00	Eivind
16	RTC	94,00	1	94,00	Eivind
17	Fjær	60,00	10	600,00	Peter
18	Guidebolter	75,00	4	300,00	Peter
19	Lastcelle test	300,00	1	300,00	Peter
<b>Totalkostnad materiell til oppgave</b>				<b>9764,00</b>	

Dette er en relativt høy kost for et utdanningsprosjekt finansiert av gruppen, men deltagere er i full jobb ved siden av studier så kostnadsnivået har i liten grad gitt negative konsekvenser for deltagerens personlige økonomi.

### 5.1.2 Indirekte kostnader

Arbeidstimer i forbindelse med utvikling anses som engangsinvestering hvor dette subsidieres av gruppen selv som i form av tapt inntekt, som følge av fri og avspasering i forbindelse med prosjektet

Øvrige kostnadsdrivere foruten om innkjøp av direkte materiell er knyttet til produksjon av komponenter som ikke kommer ferdig levert. Her har det gjennom prosess blitt brukt

maskiner og metoder som grenser mellom CNC styrte og helt manuelle prosesser. I realiteten vil en CNC-styrt vannskjæringsprosess av plater og CNC dreiebenk for maskinering av akselformede komponenter være relevant både med tanke på kvalitet og kostnad. Her må det sendes ut maskintegninger og antall, til flere leverandører for så å hente inn og vurdere de beste tilbudene.

En siste kostnadsdriver kommer til montering og kalibrering, som vi har utarbeidet montasjeanvisning vedlegg (5) og kalibreringsprosedyre vedlegg (10) som i tråd med IFE sine ønsker, monteres og kalibreres de av IFE personell. Om det skulle vise seg at IFE har tilstrekkelig med tid eller kompetanse for sammenstilling må dette utføres av en leverandør, mulig som pakkepris i forbindelse med maskinering. Montering og kalibrering utført av underleverandør er ikke noe denne oppgaven har som hensikt å anskaffe, da vi følger kravspesifikasjoner fra IFE.

## **5.2 Beregninger**

Det eneste kravet som gruppen utarbeidet i samsvar med IFE var at vi skulle ta høyde for en 1,5 von Mises sikkerhetsfaktor for konstruksjonen. Dette var muligenset noe tynt grunnlag å basere samtlige kalkulasjoner på. Samtlige utrekninger er basert på generelle formler og ved simuleringen er det gjort forenklinger med tanke på effektivitet og nødvendige komponenter som inngår i simulering. Det er behov for u-lineær simulering av hele sammenstillingen ved bruk av FEA, for å kunne gå god for at det er tatt høyde for alle krefter som inngår i konstruksjon, da det reelt opptrer spenninger utover det 2 dimensjonale planet i praksis. Men beregninger som er utført gir et godt overslag av krefter som inngår, og konstruksjonens sikkerhet mot flyt i materialer og katastrofale brudd.

Det vil muligens ikke være et godt nok beregningsunderlag for enkelte kunder om de skulle være interessert i å kjøpe og ta produktet i bruk. Det vil ved en storstilt produksjon blitt foretatt flere tester for konstruksjon for å finne svakheter knyttet til kraft, temperatur, utmatting og kombinasjonen av materialer over tid. Dette er selvsagt ikke utført da det er utviklet en prototype med nettopp denne hensikt, å kunne teste om den imøtekommer bruksområde over lang tid og ikke bare ved enkeltpåkjenninger.



Potensielle brukere vil da ofte ha sine egne kravspesifikasjoner som rigg skal kunne operere etter. For å imøtekomme disse burde det utarbeides en generell simulering av konstruksjonen for å kunne gi en kunde et raskt og konkluderende svar. Dette ligger ikke innenfor denne rapportens rammer, men burde gjennomføres ved et senere tidspunkt dersom det skulle bli aktuelt for andre brukere enn IFE.

### 5.3 Innhenting og logging av data

Kravspesifikasjonene fra IFE bestod i at batteriene skulle kunne settes under trykk ved forspenning og at dette trykket skulle kunne måles, eventuelt at utvidelse av batteriet under test også skulle kunne måles. IFE så det også som fordelaktig å plassere avleser for overvåking og trykk utenfor testskapet for å spare plass. Løsningen som er valgt gjør at brukeren kan sette batteriet under trykk ved å vri på hjulet på toppen av testtriggen. Avlesing av faktisk forspenning vises i displayet på styringsenheten og starttrykket kan lagres slik at endringer kan enkelt leses av. En styringsenhet kan overvåke opptil sju rigger og styringsenheten kan stå utenfor skapet, ettersom ledningen fra lastcellen er ganske lang. Når det kommer til overvåking av utvidelse, kommer alle fjærene oppgitt med fjærkonstanter. Utrekning av utvidelsen kan da regnes ut med Hookes lov.

Avlesning av data skulle være brukervennlig og det skulle følge med brukermanualer. Løsningen som er valgt, skal være brukervennlig med få menyer og de fleste knapper og menyer skal være intuitive. Det følger med brukermanualer for styringsenheten, loggføring av data og i tillegg en brukermanual for hvordan programmeringen kan settes opp. Manualene er på norsk og engelsk.

Loggføring av data måtte minimum kunne lastes inn på ett SD-kort, men det var også fordelaktig å kunne laste data ned via WiFi. Data fra sensorene måtte også konverteres til ønsket benevnelse og det stilles krav til nøyaktige toleranser og riktig data. Løsningen som er valgt gjør at all data kan hentes ut ved hjelp av ett SD-kort. Det er ikke valgt en løsning med WiFi da dette krever nok et utvidelseskort. Dataene kommer i en datafil som kan åpnes i Excel. Dataene som er logget er dato, klokkeslett og hvilket trykk som er lagret oppgitt i kg. Benevnelsene som kan vises på styringsenheten er i hovedsak newton og kg. Den kan også vises i MPa, men denne har noen mangler. Benevnelsene på styringsenheten er satt i kg som

standard, men dette kan endres av brukeren enten ved å endre den i styringsenheten, eller ved å endre den i kildekoden og last opp på nytt. Når det kommer til nøyaktighet av oppgitt data så har lastcellen en usikkerhet på  $\pm 0,3\%$  og ADC kortet og lastcellen skal til sammen være under  $\pm 5\%$ . Ved testing var det maksimale avviket 2,78%. Dette kan nok forbedres en hel del ved å gå over til lastceller som kommuniserer via I2C.

For at elektronikken ikke skulle skades var det ønskelig å produsere deksel for styringsenheten og dette er produsert for både styringsenheten og for ADC kortet.

## 5.4 Mulige forbedringsområder

Det er i løpet av design og produksjonsprosess avdekket flere forbedringsområder med tanke på design, maskinelementer og kostnad. De som ikke er beskrevet og utbedret i designkapittelet avsnitt 3.1 designprosess, vil beskrives nærmere i dette avsnittet.

### 1. Innfestningsmetode for guidebolter til bunnplate

Innfestningsmetode er pr nå basert på presspasning mellom aluminium og stål utregnet med pasning i avsnitt 5.1.3 og låsing ved M4x6 sett-skrue. Dette er ikke en gunstig løsning med tanke på materialkombinasjon for presspasning, samt at det medfører økt montasjetid og er avhengig av utstyr som til dømes en varmeovn og presse. Det er også knyttet økt risiko til feilmontering og økte sikkerhetsrisikoer knyttet til bruk av varme. Dette burde utbedres med en gjenget løsning mellom aksling bunnplate eller med bruk av samme type låsing som mellom topp plate og guidebolt.

### 2. Nedsenkning av bolthull for TR2X8 muttere

Her må det ses på en annen løsning for forsenkning, basert på tilgjengelig gods på mutter. Det lar seg ikke løse på en estetisk eller praktisk måte pr i dag med tanke på bruk av umbrakobolter, da dette vil føre til økte bygge mål og mindre vandring for konstruksjon. Det burde derfor ses på mulighet for en TR2X8 mutter som er ferdigforsenket, eller med andre dimensjoner, eventuelt regne over hvor stort tiltreknings-behovet er med tanke på å bruke flat-skruer

### **3. Kostnad knyttet til grafitt matter**

Grafittmatter er valgt spesifikt på grunnlag av tekniske årsaker og som resultat av kort leveringstid. Det er en relativ høy pris å betale for denne typen grafittmatter. Her burde det gjøres kostnadsomslag og vurdering av innkjøp av større kvantum med grafittflak for ved masseprodusering, istedenfor 150mm x 150mm som er brukt pr i dag. Her må man ut i markedet å gjøre ytterligere undersøkelser for å få redusert pris, men samtidig ivareta de tekniske spesifikasjonene.

### **4. Kostnad knyttet til produksjon og bearbeiding av deler**

Produksjon og bearbeiding er utført delvis ved bruk av CNC maskiner og delvis ved bruk av manuelle verktøymaskiner. Som nevnt i avsnitt 5.2.2, burde det tas en egen studie på effektivisering av produksjon og monteringslinje ved eventuelle større anskaffelser. Dette er ikke gjort i forbindelse med denne oppgaven, men burde utføres ved eventuell fullskala produksjon av jigg. Samt reelle og oppdaterte priser på en stor-skala produksjon vil veie tungt for riggen sin fremtid løsning i forskning på celler.

### **5. Kostnad knyttet til bruk av lastceller med I2C brukergrensesnitt.**

Slik oppsettet av testenheten er nå kan hver Arduino kobles opp til syv lastceller. I tillegg må det kjøpes inn ett ADC kort per lastcelle. Lastceller med I2C kan være noe dyrere, men til gjengjeld kan man i teorien koble opp så mange celler som helst, så lenge de alle har unike adresser. Da trenger man heller ikke ADC kort. Målingene vil nok også bli mer nøyaktige ettersom man tar vekk en faktor (ADC kortet) i usikkerhetsberegningen.

### **6. Grafisk brukergrensesnitt**

Det finnes muligheter for å koble flere Arduino sammen med for eksempel en Raspberry Pi ved hjelp av for eksempel I2C. Raspberry Pi kommer med mulighet til å koble til en PC-skjerm og i tillegg muligheter til å koble opp til nettet. Da kan man lage et grafisk brukergrensesnitt over alle sensorene og vise de på en stor skjerm. I tillegg vil man da ha tilgang til live data fra sensorene.

## **7. Bruke test jiggen uten lastceller**

Jiggen skal også kunne brukes uten lastceller. Tanken er at dersom man gjennom flere forsøk finner ut at man klarer å finne riktig trykk ved å finne riktig moment for skruen på toppen, kan lastcellen elimineres og man kan gå over til å kun måle utvidelsen av cellen ved en utvidelsessensor.

## **8. Visning av MPa**

Arduinoen skal vise riktig trykk i MPa når brukere setter programmet til det, men tallet som viser som lagret trykk er ikke riktig oppgitt. Dette er ikke kritisk. MPa kan enten fjernest fra programmet eller utbedres.

## **9. Eloksering av Aluminiumsplater**

Eloksering av aluminiumsplater ble ikke prioritert som følge av tidsplan, men kunne vært gjennomført for å oppnå et mer estetisk riktig produkt. Det kunne virket fordelaktig i et marked hvor mye av teknologien er lik, men de mest profesjonelle designet/utseendet kan være avgjørende sett ved siden av faktorer som pris og konstruksjonens begrensinger.

## **10. Laste ned data ved hjelp av WiFi**

Arduino Uno kan komme med Wifi, mens Arduino Mega ikke har denne muligheten. En mulighet for å koble en Arduino Mega opp med WiFi kan være å koble opp Arduinoen med en WiFi modul.

## **11. Grafittplater**

Som nevnt i kapittel 4.2.2 så leder grafittplatene strøm over overflaten, men ikke igjennom selve platen. Den vil da virke isolerende mellom batteri og rigg, men vil ikke hindre eventuell kortslutning av polene på batteriet. Det kan være ønskelig å finne en matte som ikke leder strøm over overflaten.

## **12. Kostnadsrammer fullskala produksjon**

Det skulle vært utarbeidet en nøye og oppdatert kalkyle knyttet til kostnaden ved produksjon og montering av flere rigger, hvor burde anbud føres ut i det reelle markedet, innhentet ferske tall og fremstilte disse for IFE. Tallene som kommer frem i rapport gir nøyaktige overslag

knyttet til kostnad for direktemateriell, men det er for mye usikkerhet knyttet til kostnader som frakt, produksjon og sammenstilling. Det mangler altså et konkret tilbud man kan ligge frem for IFE som beskriver en ferdig pris per rigg.

### **13. Prissammenligning med andre rigger i markedet**

Det er ikke utført noe prismatch med tilsvarende produkter i markedet, dette burde utføres for å gi et grovt estimat på hvordan rigger kommer ut i forhold til andre rigger. Selv om det er vanskelig å vurdere en celletesters kvalitet uten å ha prøvd produktet selv fysisk. Her burde det vert hentet inn informasjon basert på erfaringer fra tidligere forsøk fra anerkjente aktører. Det vil ikke være moralsk forsvarlig å tilby IFE en rigg som har riktige forutsetninger til funksjon, men for høy pris.

## 6 Konklusjon

Gruppen har i løpet av denne oppgaven og gjennom denne rapporten utført en produktutvikling i henhold til tidsplanen som ble utarbeidet i forprosjektet og innenfor rammene som ble satt innledningsvis i rapporten. Det ble tydelig tidlig i prosessen at en produktutvikling, selv med de rammene som ble satt, ville kreve svært mye av studentene som deltok i oppgaven. Det har derfor vært en kontinuerlig utfordring å prioritere riktig i forhold til rapportens innhold, med hensyn på hvilke deler av prosessen som skal ivaretas mest..

Tar man hovedproblemstillingen i betraktning «*Hvordan utvikle og produsere en rigg for celledtesting som imøtekommer kravspesifikasjoner fra IFE, samtidig som den ikke er for komplisert for produsent og forbruker?*», så fremstår det som hovedproblemstillingen er besvart i rapporten. Etter flere tester av rigg konkluderes det med at riggen oppfyller tiltenkt funksjon, også i praksis. Tilbakemelding fra IFE på den endelige iterasjonen av produktet bekrefter at riggen oppfyller spesifikasjonene beskrevet under seksjon 2.1.1. Det vektlegges som spesielt positivt at designet hensyntar mulig utvidelse av cellen under syklisk opp- og utlading, ved å implementere både løpende måling av faktisk trykk for å overvåke effekten av utvidelsen, og fjærbelastning for å minimere denne effekten. Dette samtidig som at riggen likevel er brukervennlig, lite kompleks, og med et design som er fleksibelt med hensyn på mulig fremtidig modifikasjon for andre cellegeometrier.

Ser man på rapportens besvarelse av de fire delproblemstillingene (gitt i avsnitt 1.2 problemstillinger), så er samtlige spørsmål besvart gjennom rapporten. Hvor alle delproblemstillinger er besvart på en god måte, men samtidig krever enkelte en liten drøfting for å kunne gi et fullgodt svar, som for delproblemstilling nummer en som er knyttet mot økonomi. Besvarelser er vinklet fra et ingeniørperspektiv, som har som hensikt å besvare de teoretiske og praktiske problemstillingene knytte til utvikling og produksjon av riggen.

Rapportens svar for delspørsmål nummer en, «*Hvordan er det mulig å utvikle en rigg for celledtesting som imøtekommer kravspesifikasjoner, som samtidig har et økonomisk fortrinn sammenlignet andre rigger som har tilsvarende kapabiliteter*». Inneholder rapporten ikke andre økonomiske konklusjoner eller sammenligninger enn egen utvikling og produksjonskost, dette er et stort tema som burde ses videre på før en eventuell

masseproduksjon. Mangelen på sammenligningsgrunnlag er også et naturlig resultat av at produktet oppfyller en skreddersydd rolle i et spesifikt laboratorium, og derfor ikke har et etablert og transparent marked. IFEs vurdering av kostnadsestimatet beskrevet under seksjon 2.3.2, er at det ligger godt innenfor betalingsviljen for et produkt med disse kapabilitetene. Det skal påpekes at en markedsanalyse ikke var innenfor oppgavens rammer.

For delproblemstilling nummer to «*Hva skal man prioritere mest med tanke på produktets kvalitet, innovasjon og bærekraft når det dukker opp problemstillinger underveis i prosessen?*». Her har man lært i løpet av prosessen at det å samkjøre faktorer som kvalitet, kost, innovasjon og bærekraft ikke er enkelt til enhver tid. Men at en god løsning av den ene faktoren ofte bærer ofte påvirker de andre faktorene positivt, eksempelvis som å 3D-printe komponenter som ikke en bærende del av konstruksjon, eller å samkjøre flere lastceller på samme arduino. Dette drar opp kvalitet, øker bærekraft og reduserer kost med tanke på frakt og produksjonstid, samt det viser til innovativ tankegang gjennom det å utforske og samtidig implementere forskjellige kombinasjoner nye teknologier.

For delproblemstilling nummer tre «*Hvor vil problemer og forbedringsområder hovedsakelig ligge i en prototype?*», er det erfart at problemer og forbedringsområder ofte oppstår i en veldig tidlig fase ved design og som følge av løsninger knyttet til riggens primære funksjoner. Dette ble løst veldig effektivt med å lage hver sine ulike design og sammenligne og drøfte dem i møte med IFE, for så å kjøre en elimineringsprosess hvor gruppen og IFE i samråd velger ut det designet som videre skal optimaliseres i henhold til kravspesifikasjoner. Men de mest aktuelle forbedringsområdene oppstår når prototype er produsert og testet, først da ble det utarbeidet veldig spesifikke forbedringsområder som burde tas høyde for ved en senere masseproduksjon.

For delproblemstilling nummer fire «*Hvordan skal denne riggen bli et førstevalg i forskningssammenheng på celler?*», så besvares dette med at vurderingen av hvordan denne riggen kan bli et førstevalg i forskningssammenheng på poseceller henger tett sammen med vurderingen av hovedproblemstillingen. I tillegg er IFEs tilbakemelding på dette punktet at pålitelighet, målbarhet og reproduserbarhet vil være spesielt utslagsgivende for valg av utstyr til forskningsøyemed. Dette fordi verdien i ethvert eksperiment er avhengig av at relevante parametere i omgivelsene er kontrollert og/eller blir målt, for å kunne evaluere effekten tilsiktede endringer. For riggen i denne oppgaven oppnås dette hovedsakelig ved 1) Høy brukervennlighet og lav kompleksitet, som reduserer risiko for bruker- og utstyrfeil; 2)

Fjærbelastning, som bidrar til å opprettholde jevnt trykk gjennom eksperimentet; og 3) Måling av faktisk trykk, som både øker reproduserbarhet mellom tester og plukker opp eventuelle avvik underveis i testingen. IFEs vurdering er at denne graden av kontroll generelt er forbeholdt langt mer avansert og kostbart utstyr, og at denne riggen derfor vil være meget attraktiv for bruk i forskningssammenheng.

Når det kommer til hovedmålene som ble satt for oppgaven er alle punkter tatt høyde for i rapport, noen mer enn andre. Det foreligger tilstrekkelig dokumentasjon i rapport og vedlegg for å kunne produsere og montere en identisk rigg, samt at riggen har en langsiktig tankegang og skal ved behov være forholdsvis enkel å modifisere.

Gruppens personlige refleksjoner over oppgaven er at det som innledningsvis skulle virke som en veldig overkommelig oppgave med tanke på arbeidsmengde, utviklet seg til å bli en solid mengde arbeid. Dette resulterte i at det var vanskelig å utarbeide en rapport hvor gruppen følte at alle deler av prosessen fikk like mye oppmerksomhet som de fortjente. Det er derfor en stor mengde vedlegg utover rapporten, som absolutt er relative og oppsummerende for oppgaven.

IFE har uttalt at pilot hvor prototype skal testes, har oppstart i løpet av sommeren 2023. Det vil med dette konkludere om riggen er tilstrekkelig for oppgaven som den er i dag, eller om den eventuelt må re-designes etter mulige forbedringsområder (avsnitt 5.2) og IFE sine egne erfaringer fra piloten.



# Litteraturliste

Adafruit (u.å.a), *Adafruit 1,8'' Color TFT Shield w/microSD and joystick – V2*. Tilgjengelig fra: <https://www.adafruit.com/product/802> (Hentet 1.5.2023)

Adafruit (u.å.b), *Adafruit DS1307 Real Time Clock Assembled Breakout Board*. Tilgjengelig fra: <https://www.adafruit.com/product/3296> (Hentet 1.5.2023)

*AISI Type 304 Stainless Steel*, (u.å.) Tilgjengelig fra: <https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=mq304a> (Hentet 18.05.2023)

*Aluminum 6061-T6* (u.å.), Tilgjengelig fra: <https://www.matweb.com/search/datasheet.aspx?MatGUID=b8d536e0b9b54bd7b69e4124d8f1d20a&ckck=1> (Hentet 18.05.2023)

Anderson, D., Farah, S. og Langer, R. (2016) *Physical and mechanical properties of PLA, and their functions i widespread applications – A comprehensive review*. Tilgjengelig fra: [https://dspace.mit.edu/bitstream/handle/1721.1/112940/Anderson\\_Physical and mechanical properties.pdf?sequence=1](https://dspace.mit.edu/bitstream/handle/1721.1/112940/Anderson_Physical%20and%20mechanical%20properties.pdf?sequence=1) (Hentet 1.5.2023)

Arduino.cc (u.å.a) *Arduino Mega 2560 REV3*. Tilgjengelig fra: <https://docs.arduino.cc/static/2de2c8ff00fc05065634e3823a9266c4/A000067-pinout.png> (Hentet 1.5.2023)

Arduino.cc (u.å.b) *Boards*. Tilgjengelig fra: <https://store.arduino.cc/collections/boards> (Hentet 1.5.2023)

Arduino.cc (2023) *What is Arduino?* Tilgjengelig fra: <https://docs.arduino.cc/learn/starting-guide/whats-arduino> (Hentet 26.04.2023).

Boissonneault, T (2022) *PLA recycling: Can PLA 3D Printing Filament be Recycled?* Tilgjengelig fra: <https://www.wevolver.com/article/pla-recycling-can-pla-3d-printer-filament-be-recycled> (Hentet 16.05.2023)

Campbell, S. (u.å.) *Basics of the I2C communication protocol*. Tilgjengelig fra: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/> (Hentet 26.04.2023)

Dahlvig, G. et al. (1991) *Konstruksjonselementer, 2 utgave*  
OSLO: Gyldendal Norsk forlag AS

ElectronicsTutorials, (u.å.), *Analogue to Digital Converter*. Tilgjengelig fra:  
<https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html> (Hentet 1.5.2023))

FN-Sambandet (2023), *Ansvarlig forbruk og produksjon*. Tilgjengelig fra:  
<https://www.fn.no/om-fn/fns-baerekraftsmaal/ansvarlig-forbruk-og-produksjon> (Hentet 18.05.2023)

Grøn, Ø. (2022) *termisk konduktivitet*. Tilgjengelig fra: [https://snl.no/termisk\\_konduktivitet](https://snl.no/termisk_konduktivitet)  
(Hentet 18.05.2023)

Hartvigsen, H. et al. (2006) *Verkstetshåndboka, 6 utgave*  
Tromsø: Gyldendal Norsk Forlag AS

Helseth, L.E. (2023) *Varmeledningsevne*. Tilgjengelig fra: <https://snl.no/varmeledningsevne>  
(Hentet 18.05.2023)

*How and why alloying elements are added to aluminum*, (u.å.). Tilgjengelig fra:  
<http://www.alcotec.com/us/en/education/knowledge/qa/How-and-why-alloying-elements-are-added-to-aluminum.cfm> (Hentet 18.05.2023)

HTC-Sensor.com (u.å.), *TAS606 Miniature Compression Type Force Sensor*. Tilgjengelig fra: [https://cdn.sparkfun.com/assets/6/e/b/c/3/DS-13332-Load\\_Cell\\_-\\_200kg\\_\\_Disc\\_\\_TAS606\\_.pdf](https://cdn.sparkfun.com/assets/6/e/b/c/3/DS-13332-Load_Cell_-_200kg__Disc__TAS606_.pdf) (Hentet 1.5.2023)

HT Sensor Technology CO., Limited, (u.å.), *TAS606 Button type compression load cell*.  
Tilgjengelig fra: <http://www.htc-sensor.com/products/151.html> (Hentet 1.5.2023)

IFE (2023) *Om IFE*. Tilgjengelig fra: <https://ife.no/om-ife/> (Hentet: 17. Januar 2023).

Igus (2023) *product iglide J, sleeve bearing*. Tilgjengelig fra:  
<https://www.igus.com/product/3?artNr=JSM-0810-10> (Hentet: 20. Februar 2023).

International Standard (2019) *ISO 10243: Tools for pressing – Compression springs with rectangular section – Housing dimensions and colour coding*. Tilgjengelig fra:

<https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1037307> (Hentet 2.5.2023)

Matweb.com, (u.å.) *Overview of materials for Polyethylene Terephthalate (PET), Unreinforced*. Tilgjengelig fra:

<https://www.matweb.com/search/DataSheet.aspx?MatGUID=a696bdcdf6f41dd98f8eec3599eaa20&ckck=1> (Hentet 1.5.2023)

Miljømerking Norge (u.å.), *Sirkulær økonomi*. Tilgjengelig fra:

<https://svanemerket.no/miljo/sirkulaer-okonomi/> (Hentet 18.05.2023)

Müller, *et al.*, (2019) Effects of Mechanical Compression on the Aging and the Expansion Behavior of Si/C-Composite|NMC811 in Different Lithium-Ion Battery Cell Formats, *Journal of The Electrochemical Society*, 166, doi: 10.1149/2.1121915jes

Rjaa *et al.* (2018) Aging Time Effects on the Mechanical Properties of Al 6061-T6 Alloy. *Engineering, Technology and Applied Science Reseach*. DOI: 10.48084/etasr.2102

Raspberry Pi Foundation (u.å.a) *Raspberry Pi hardware*. Tilgjengelig fra:

<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (Hentet 26.04.2023)

Raspberry Pi Foundation (u.å.b) *What is a Raspberry Pi?* Tilgjengelig fra:

<https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/> (Hentet 26.04.2023)

Rolstandås, A, et al. (2014) *Praktisk prosjektledelse, 2 utgave*

Bergen: Fagbokforlaget

Soedemann industrifjerde A/S (2020) *ISO verktøyfjærer, CAD modell*.

Tilgjengelig fra: <https://www.fjaer.net/produkter/verktoyfjaerer/> (Hentet 05.02.2023)

Spencer, Eric (2019) *Machining parameters for milling*. Tilgjengelig fra :

[www.machiningdoctor.com/calculators/#milling-calculators](http://www.machiningdoctor.com/calculators/#milling-calculators) (Hentet 07.04.2023).

Sparkfun.com (u.å.a), *Load Cell Amplifier HX711 Breakout Hookup Guide*. Tilgjengelig fra:

[https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?\\_ga=2.175351266.1503137913.1681123666-1224697633.1677188976](https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?_ga=2.175351266.1503137913.1681123666-1224697633.1677188976).# (Hentet 1.5.2023)

Sparkfun.com (u.å.b), *Load Cell – 200kg. Disc (TAS606)*. Tilgjengelig fra:  
<https://www.sparkfun.com/products/13332> (Hentet 1.5.2023)

Sparkfun.com (u.å.c), *Sparkfun Load Cell Amplifier – HX711*. Tilgjengelig fra:  
<https://www.sparkfun.com/products/13879> (Hentet 1.5.2023)

TE Connectivity. (2020) *FX29 Compact compression load cell*. Tilgjengelig fra:  
<https://docs.rs-online.com/e996/A700000007467685.pdf> (Hentet 26.04.2023)

The International Aluminium Institute (2018), *Sustainability*. Tilgjengelig fra:  
<https://recycling.world-aluminium.org/review/sustainability/> (Hentet 18.05.2023)

The Robotics Back-End (u.å.) *Arduino Uno Pins – A Complete Practical Guide*. Tilgjengelig fra: <https://roboticsbackend.com/arduino-uno-pins-a-complete-practical-guide/> (Hentet 26.04.2023)

Trent. D (u.å.a) *Load cell specifications & datasheets*. Tilgjengelig fra:  
<https://www.800loadcel.com/white-papers/377.html> (Hentet 26.04.2023)

Trent. D (u.å.b) *Strain gauge load cell basics*. Tilgjengelig fra:  
<https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html> (Hentet 26.04.2023)

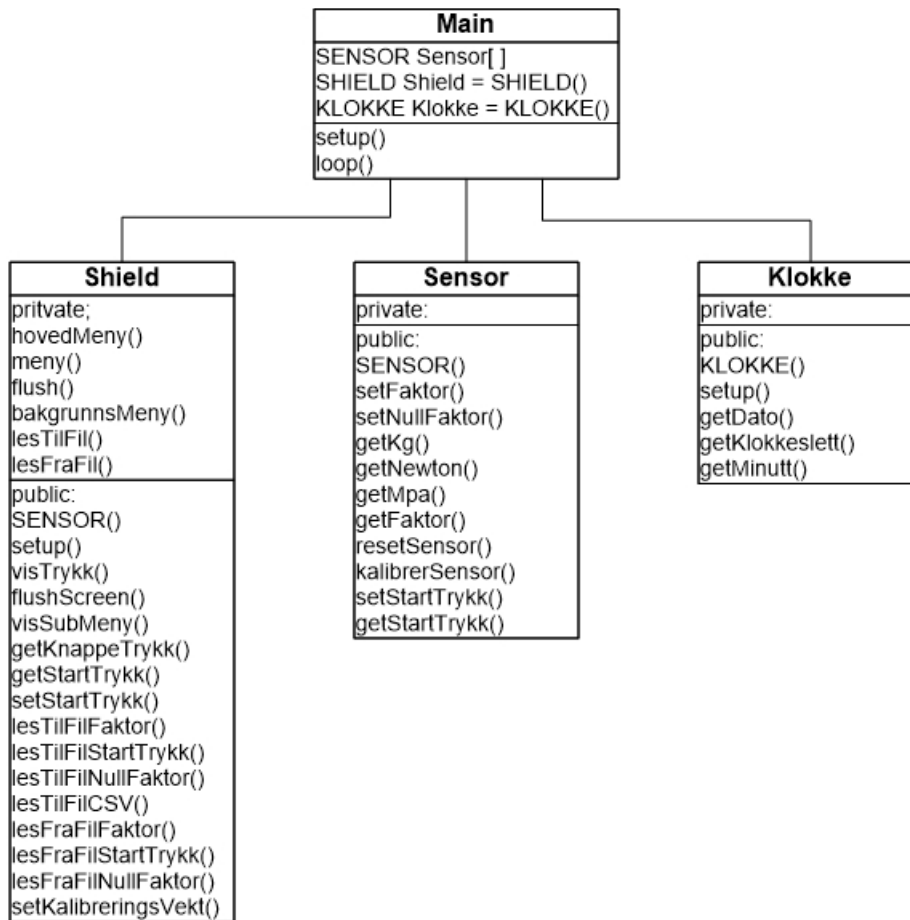
University Libraries, (2022), *Introduction to Shields*. Tilgjengelig fra:  
<https://guides.temple.edu/c.php?g=419841&p=2908632> (Hentet 1.4.2023)

Wang G. S., Liu K. og Wang S. L., (2018), *Evolution of Elevated-Temperature Strength and Creep Resistance during Multi-Step Heat Treatments in Al-Mn-Mg Alloy*. Tilgjengelig fra:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6073777/#:~:text=Generally%20the%20dislocation%20creep%20occurs,22%2C23%2C24%5D> (Hentet: 16.05.2023)

Thermal Interface Sheet, (2023). Tilgjengelig fra: [https://no.rs-online.com/web/p/thermal-pads/7943973?fbclid=IwAR3GTlo\\_nnuvZuejKCsVwBnXPEdzcQQiOF797cJLcRUa98qKPSDT73a7C6I](https://no.rs-online.com/web/p/thermal-pads/7943973?fbclid=IwAR3GTlo_nnuvZuejKCsVwBnXPEdzcQQiOF797cJLcRUa98qKPSDT73a7C6I) (Hentet: 16.05.2023)

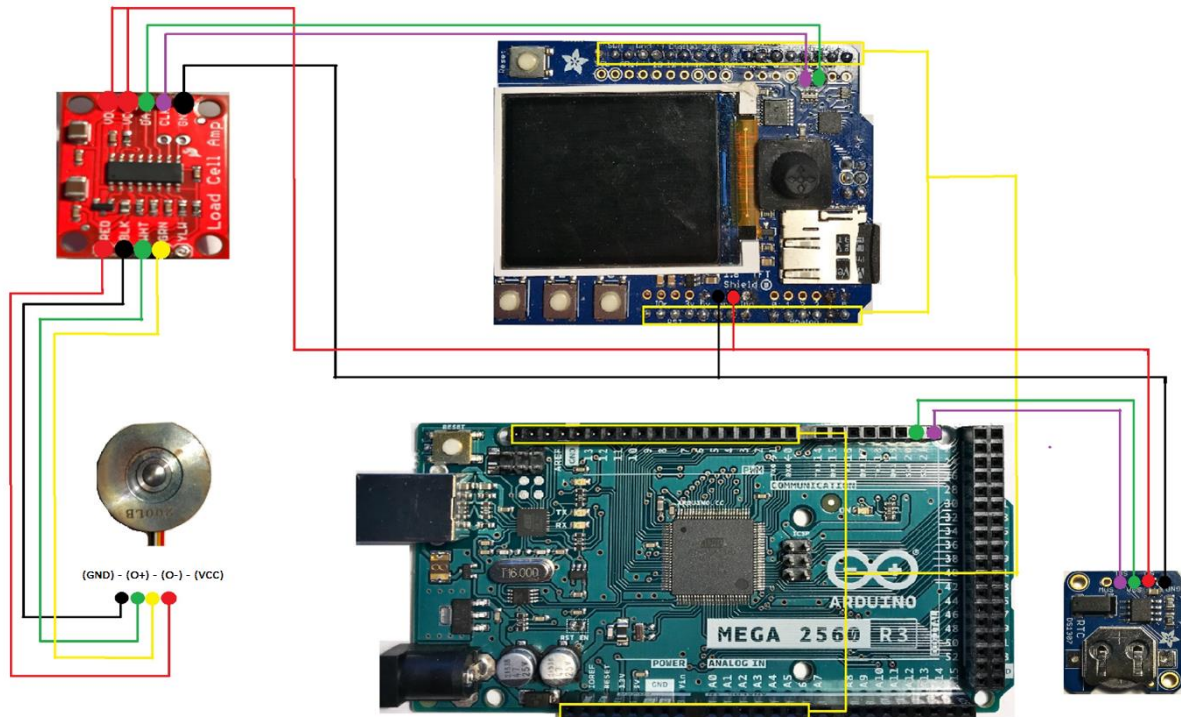
# Vedlegg

## Vedlegg 1: Klassediagram



Figur 58: Klassediagram

## Vedlegg 2: Koblingskjema Arduino Mega med shield



Figur 59: Koblingskjema

## Vedlegg 3: Testing av lastcelle fra TE Connectivity

	A	B	C	D	E	F	G	H	I	J	K
1		Kalibrering					Testing				
2	Forsøksnummer	Avlastet kg	Nullfaktor	Vektskivenr.	Belastet kg	Faktor	Testvekt	Vektskivenr	Forventet vekt	Avlest vekt	Avvik
3	1	0	1476204	1 og 5	5,55	529500	5,55	1 og 5	5,55	5,51	-0,04
4							5,6	1 og 4	5,6	5,59	-0,01
5							10,45	1 og 2	10,43	10,37	-0,06
6							20,5	1 og 6	20,5	13,35	-7,15
7											

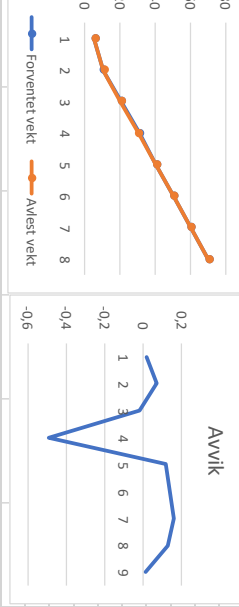
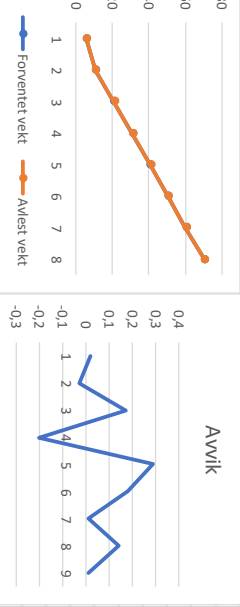
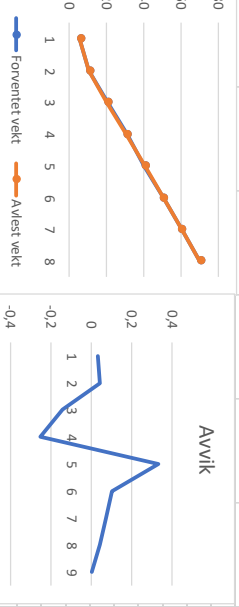
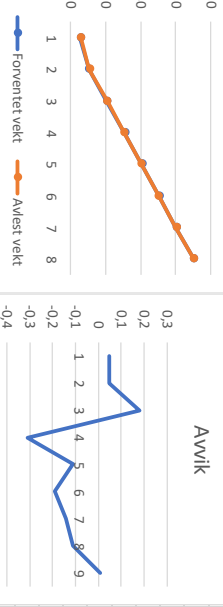
Figur 60: Testing av lastcelle fra TE Connectivity

## Vedlegg 4: Testing av lastcelle fra Sparkfun (Excel-filer er vedlagt innlevering)

Forsøksnr.	Kalibrering	Avlastet kg	Nullfaktor	Vektstivemr.	Balastet kg	Faktor	13400	Testing					
								Testvekt	Vektstivemr.	Forventet vekt	Avlest vekt	Avvik	
1	0	1884	1 og 5	5,55	5,55	13400	5,55	1 og 5	5,55	0,0	0,00		
							10,45	1 og 2	10,43	-0,02	-0,19		
							20,5	1 og 6	20,51	0,01	0,05		
							30,85	1, 2 og 6	30,26	-0,59	-1,91		
							40,45	1,6 og 7	40,23	-0,22	-0,54		
							50,35	1,2,6 og 7	50,63	0,28	0,56		
							60,2	1,2,3,6 og 7	60,2	59,93	-0,27		-0,45
							70,25	1,2,3,4,5,6 og 7	70,25	69,85	-0,4		-0,57
							0	Avlastet	0	0	0		
							0	Avlastet	0	0	0		
2	0	1937	1 og 2	10,45	10,45	13350	5,55	1 og 5	5,57	0,02	0,36		
							10,45	1 og 2	10,48	0,03	0,29		
							20,5	1 og 6	20,55	0,05	0,24		
							30,85	1, 2 og 6	30,58	-0,27	-0,88		
							40,45	1,6 og 7	40,59	0,14	0,35		
							50,35	1,2,6 og 7	50,42	0,07	0,14		
							60,2	1,2,3,6 og 7	60,2	60,57	0,37		0,61
							70,25	1,2,3,4,5,6 og 7	70,25	70,34	0,09		0,13
							0	Avlastet	0	0,01	0,01		
							0	Avlastet	0	0,01	0,01		
3	0	1938	1 og 6	20,5	20,5	13350	5,55	1 og 5	5,54	-0,01	-0,18		
							10,45	1 og 2	10,42	-0,03	-0,29		
							20,5	1 og 6	20,55	0,05	0,24		
							30,85	1, 2 og 6	30,62	-0,23	-0,75		
							40,45	1,6 og 7	40,39	-0,06	-0,15		
							50,35	1,2,6 og 7	50,22	-0,13	-0,26		
							60,2	1,2,3,6 og 7	60,2	60,12	-0,08		-0,13
							70,25	1,2,3,4,5,6 og 7	70,25	69,89	-0,36		-0,51
							0	Avlastet	0	0,03	0,03		
							0	Avlastet	0	0,03	0,03		
4	0	2426	1, 2 og 6	30,85	30,85	13150	5,55	1 og 5	5,7	0,15	2,70		
							10,45	1 og 2	10,64	0,19	1,82		
							20,5	1 og 6	21,00	0,5	2,44		
							30,85	1, 2 og 6	30,89	0,04	0,13		
							40,45	1,6 og 7	40,45	41,1	0,65		1,61
							50,35	1,2,6 og 7	50,35	51,75	1,4		2,78
							60,2	1,2,3,6 og 7	60,2	61,36	1,16		1,93
							70,25	1,2,3,4,5,6 og 7	70,25	72,15	1,9		2,70
							0	Avlastet	0	0,02	0,02		
							0	Avlastet	0	0,02	0,02		

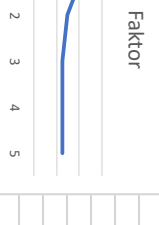
Figur 61: Test av lastcelle fra Sparkfun

Forsøksnr.	Avlastet kg	Nulffaktor	Vektstkvendr.	Belastet kg	Faktor	Testvekt	Vektstkvendr.	Forventet vekt	Avlest vekt	Avvik	% Avvik
5	0	3015	1,6 og 7	40,45		5,55	1 og 5	5,55	5,6	0,05	0,90
						10,45	1 og 2	10,45	10,5	0,05	0,48
						20,5	1 og 6	20,5	20,68	0,18	0,88
						30,85	1, 2 og 6	30,85	30,54	-0,31	-1,00
						40,45	1,6 og 7	40,45	40,34	-0,11	-0,27
						50,35	1,2,6 og 7	50,35	50,16	-0,19	-0,38
						60,2	1,2,3,6 og 7	60,2	60,06	-0,14	-0,23
						70,25	1,2,3,4,5,6 og 7	70,25	70,14	-0,11	-0,16
						0	Avlastet	0	0,01	0,01	
<b>Forsøksnr.</b>	<b>Avlastet kg</b>	<b>Nulffaktor</b>	<b>Vektstkvendr.</b>	<b>Belastet kg</b>	<b>Faktor</b>	<b>Testvekt</b>	<b>Vektstkvendr.</b>	<b>Forventet vekt</b>	<b>Avlest vekt</b>	<b>Avvik</b>	<b>% Avvik</b>
6	0	3333	1,2,6 og 7	50,35		5,55	1 og 5	5,55	5,58	0,03	0,54
						10,45	1 og 2	10,45	10,49	0,04	0,38
						20,5	1 og 6	20,5	20,36	-0,14	-0,68
						30,85	1, 2 og 6	30,85	30,6	-0,25	-0,81
						40,45	1,6 og 7	40,45	40,78	0,33	0,82
						50,35	1,2,6 og 7	50,35	50,45	0,1	0,20
						60,2	1,2,3,6 og 7	60,2	60,27	0,07	0,12
						70,25	1,2,3,4,5,6 og 7	70,25	70,29	0,04	0,06
						0	Avlastet	0	0	0	
<b>Forsøksnr.</b>	<b>Avlastet kg</b>	<b>Nulffaktor</b>	<b>Vektstkvendr.</b>	<b>Belastet kg</b>	<b>Faktor</b>	<b>Testvekt</b>	<b>Vektstkvendr.</b>	<b>Forventet vekt</b>	<b>Avlest vekt</b>	<b>Avvik</b>	<b>% Avvik</b>
7	0	3453	1,2,3,6 og 7	60,2		5,55	1 og 5	5,55	5,57	0,02	0,36
						10,45	1 og 2	10,45	10,42	-0,03	-0,29
						20,5	1 og 6	20,5	20,67	0,17	0,83
						30,85	1, 2 og 6	30,85	30,65	-0,2	-0,65
						40,45	1,6 og 7	40,45	40,74	0,29	0,72
						50,35	1,2,6 og 7	50,35	50,53	0,18	0,36
						60,2	1,2,3,6 og 7	60,2	60,21	0,01	0,02
						70,25	1,2,3,4,5,6 og 7	70,25	70,39	0,14	0,20
						0	Avlastet	0	0,01	0,01	
<b>Forsøksnr.</b>	<b>Avlastet kg</b>	<b>Nulffaktor</b>	<b>Vektstkvendr.</b>	<b>Belastet kg</b>	<b>Faktor</b>	<b>Testvekt</b>	<b>Vektstkvendr.</b>	<b>Forventet vekt</b>	<b>Avlest vekt</b>	<b>Avvik</b>	<b>% Avvik</b>
8	0	3645	1,2,3,4,5,6 og 7	70,25		5,55	1 og 5	5,55	5,57	0,02	0,36
						10,45	1 og 2	10,45	10,52	0,07	0,67
						20,5	1 og 6	20,5	20,48	-0,02	-0,10
						30,85	1, 2 og 6	30,85	30,36	-0,49	-1,59
						40,45	1,6 og 7	40,45	40,57	0,12	0,30
						50,35	1,2,6 og 7	50,49	50,49	0,14	0,28
						60,2	1,2,3,6 og 7	60,2	60,36	0,16	0,27
						70,25	1,2,3,4,5,6 og 7	70,25	70,38	0,13	0,19
						0	Avlastet	0	0,01	0,01	





Forsøksnr.		Nullfaktor		Vektstivener.		Belastet kg		Faktor	
	Avlastet kg								
9	0	3477		8	1,00		13650		
10	0	3578		8	1,00		13500		
11	0	3590		8	1,00		13450		
12	0	3591		8	1,00		13450		
13	0	3589		8	1,00		13450		
	Gj.snitt	3565		Gj.snitt			13500		
14	0	3557		9	2,00		13350		
15	0	3533		9	2,00		13450		
16	0	3494		9	2,00		13450		
17	0	3547		9	2,00		13350		
18	0	3577		9	2,00		13400		
	Gj.snitt	3541,6		Gj.snitt			13400		
19	0	3527		10	3,00		13400		
20	0	3600		10	3,00		13350		
21	0	3498		10	3,00		13350		
22	0	3573		10	3,00		13350		
23	0	3538		10	3,00		13450		
	Gj.snitt	3547,2		Gj.snitt			13380		
24	0	3521		11	4,00		13350		
25	0	3459		11	4,00		13350		
26	0	3503		11	4,00		13350		
27	0	3476		11	4,00		13350		
28	0	3509		11	4,00		13350		
	Gj.snitt	3493,6		Gj.snitt			13350		
	Nullfaktor								
	Faktor								
Forsøksnr.		Manuelt innsett		Vektstivener.		Belastet kg		Faktor	
29		1884	1 og 5			5,55	13150		Testing
30		1884				5,55	13650		
31		3645				5,55	13150		
32		3645				5,55	13650		
	Nullfaktor								
	Faktor								
Forsøksnr.		Manuelt innsett		Vektstivener.		Belastet kg		Faktor	
33		1884	1,2,3,4,5,6 og 7			70,25	13150		Testing
34		1884				70,25	13650		
35		3645				70,25	13150		
36		3645				70,25	13650		

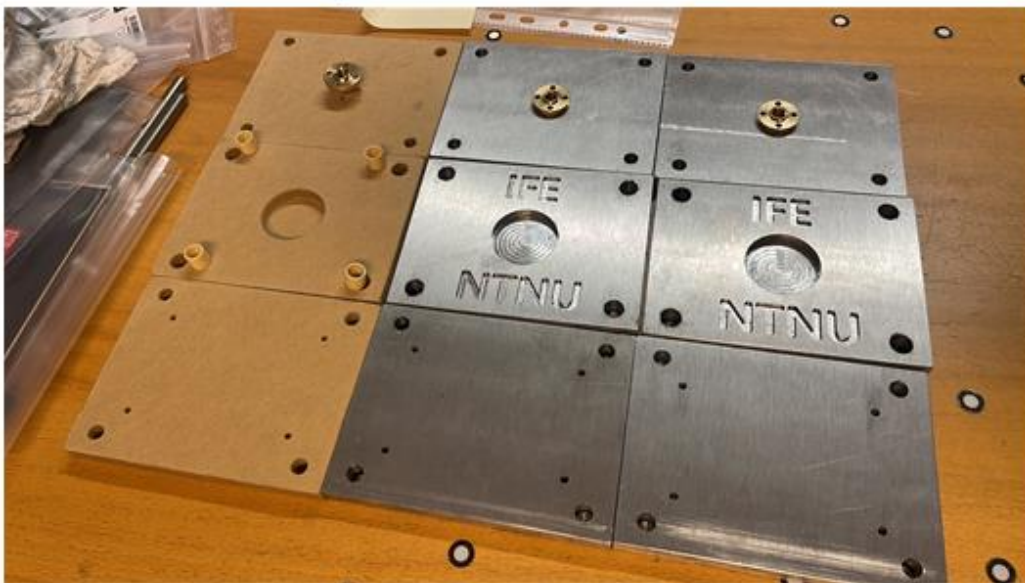


## Vedlegg 5: Målebrev av plater (Org. filer lagt ved innlevering)



**innovmetric**

Report Author: Henrik Berg  
Date: 24.04.2023



3D scan utført med Handyscan Black elite  
Konvertert til STL i VX elements  
Deretter målebrev utarbeidet i Polyworks inspector

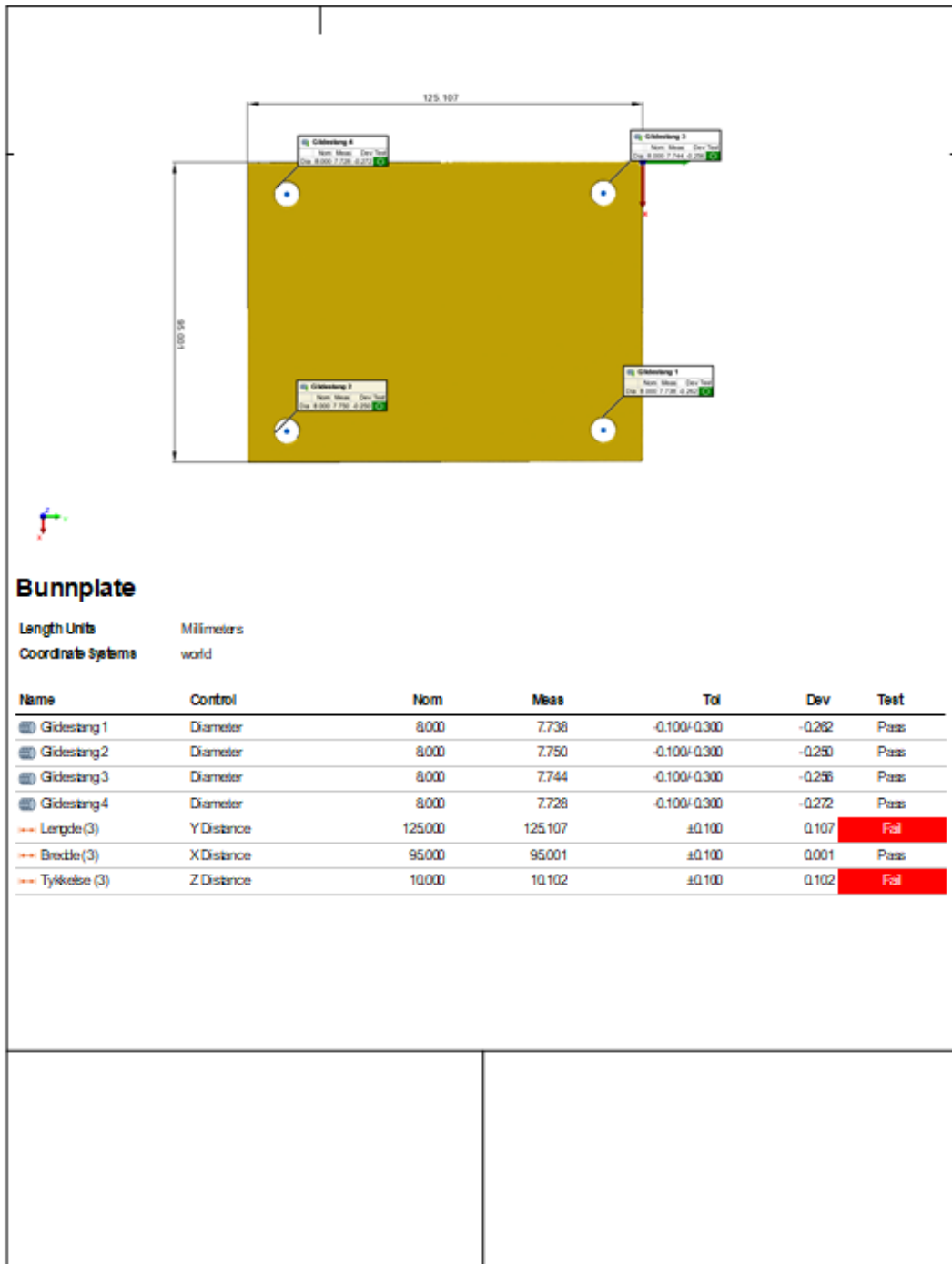
Alle mål av interesse kontrollert og avvik finnes i rapport.  
Som følge av dette målebrevet kan vi korrigere eventuelle utfordringer knyttet til senere produksjon av batteri-celle tester.

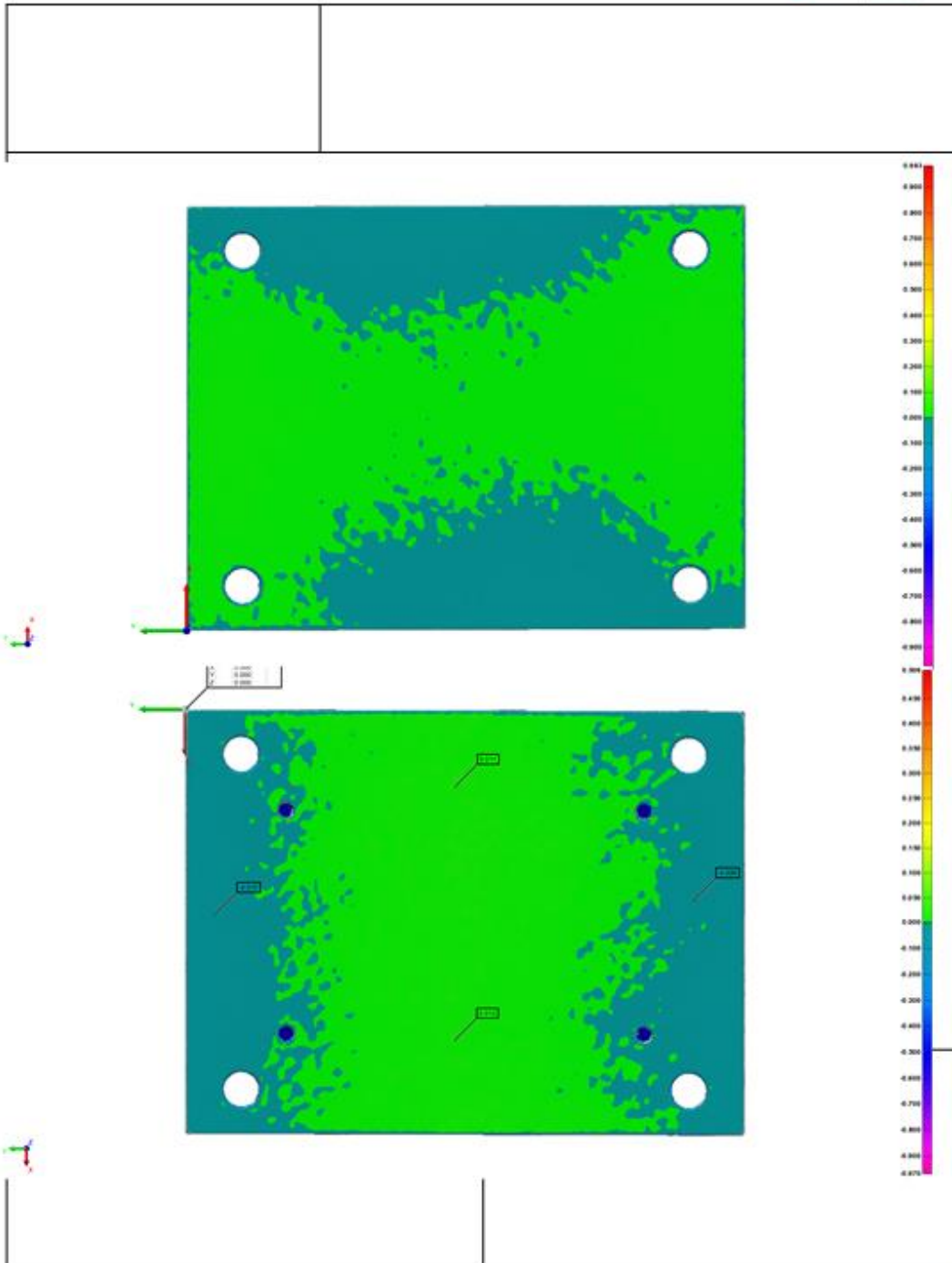
Obs ingen direkte link bortsett fra navn knyttet mot maskintegninger  
Standard for ikke toleransesatte mål er iht NS-EN-2768-1

<b>Organization:</b> NTNU	<b>Part name:</b> Puch cell tester
<b>Operator:</b>	<b>Part number:</b>
<b>E-mail:</b> Henrikwb@ntnu.no	<b>Drawing #:</b> 1-1 Assem b
<b>Workspace:</b>	<b>Serial #:</b>
<b>Project:</b> Målebrev Puch cell tester	<b>Device:</b> C reaform Handys can black

1/7

Målebrev 1/7





3/7

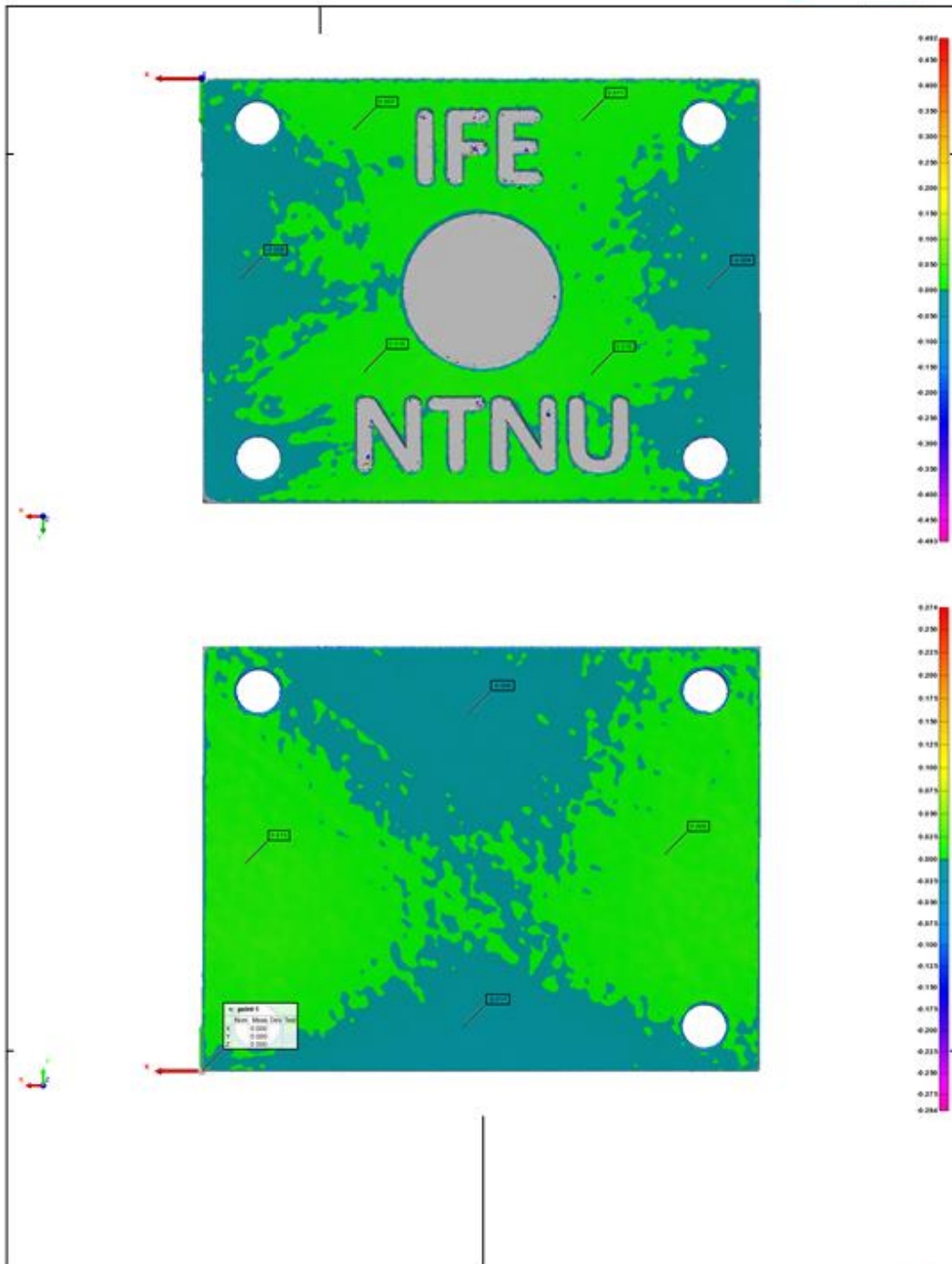
Målebrev 3/7



### Mellom Plate

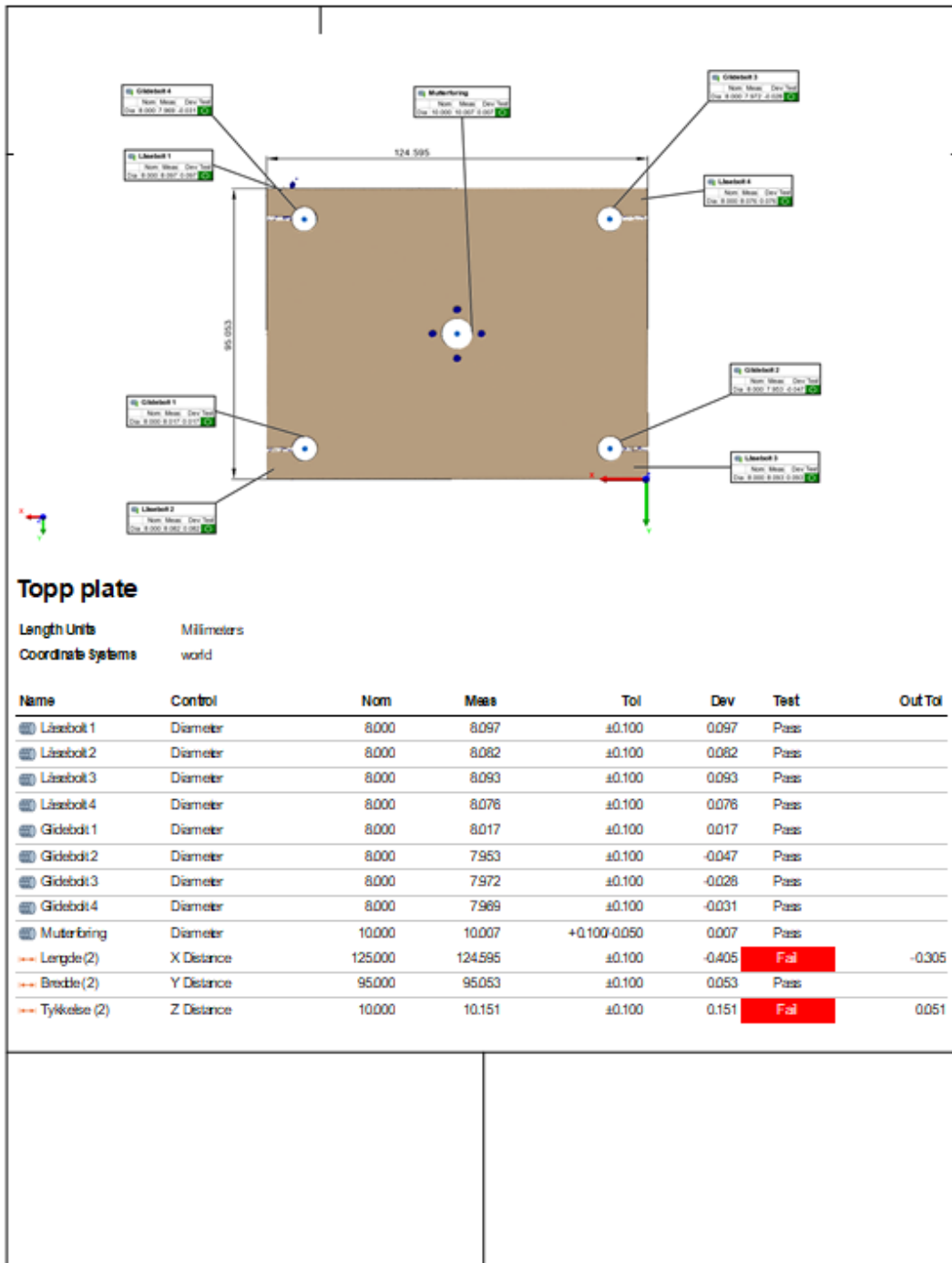
Length Units: Millimeters  
Coordinate Systems: world

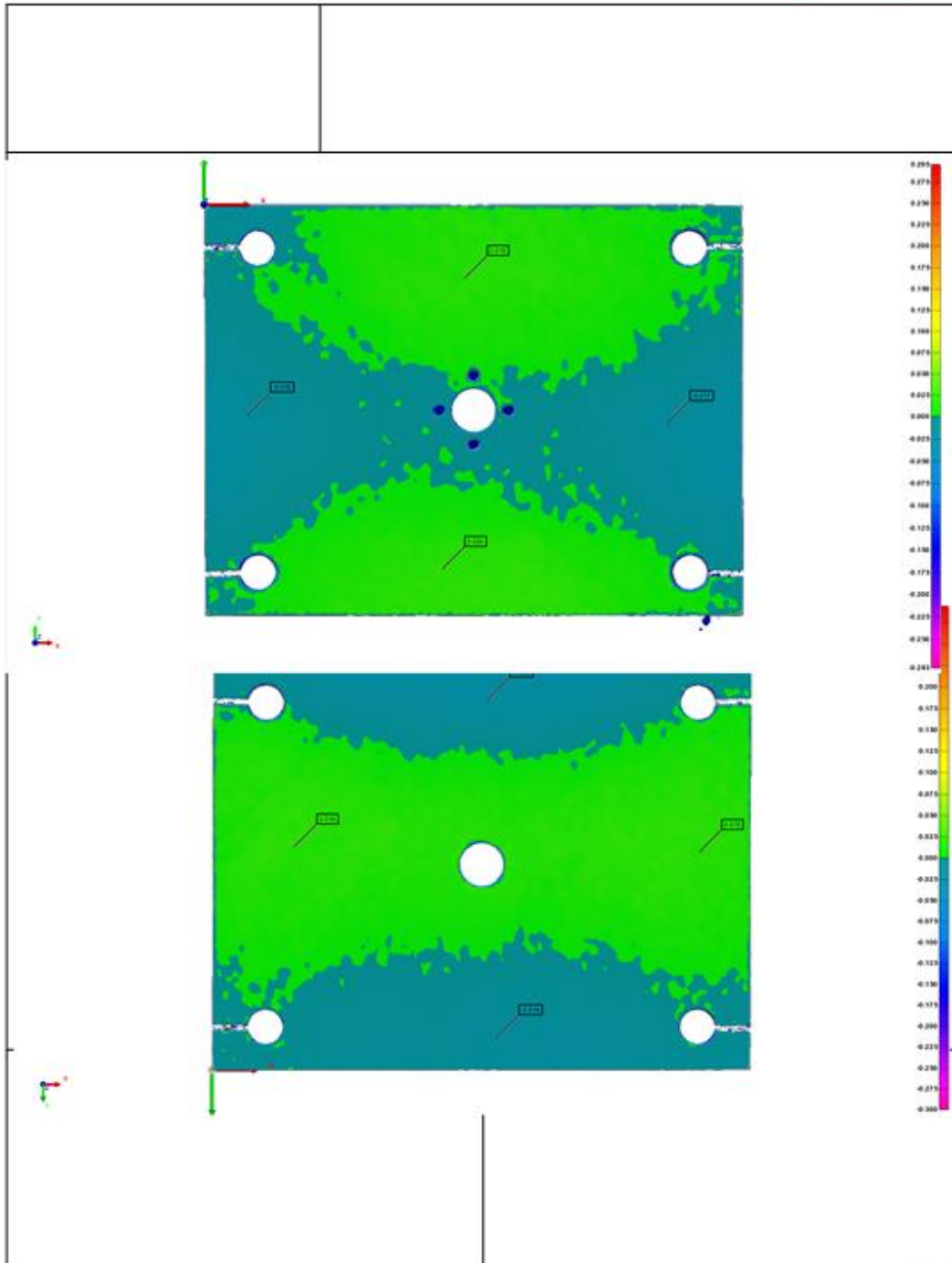
Name	Control	Nom	Meas	Tol	Dev	Test	Out Tol
Foring 1	Diameter	9.800	9.644	0.000/-0.200	-0.156	Pass	
Foring 2	Diameter	9.800	9.609	0.000/-0.200	-0.191	Pass	
Foring 3	Diameter	9.800	9.626	0.000/-0.200	-0.174	Pass	
Foring 4	Diameter	9.800	9.623	0.000/-0.200	-0.177	Pass	
Utspringcalle	Diameter	35.000	34.859	+0.100/-0.150	-0.141	Pass	
Lengde	X Distance	125.000	124.975	±0.100	-0.025	Pass	
Bredd	Y Distance	95.000	95.237	±0.100	0.237	Fail	0.137
Tykkelse	Z Distance	10.000	10.133	±0.100	0.133	Fail	0.033



5/7

Målebrev 5/7





7/7

Målebrev 7/7



## Vedlegg 6: Monteringsanvisning (Org. filer lagt ved innlevering)

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	BOTTOM	AL 6061 T6	1
2	MID	AL 6061 T6	1
3	TOP	AL 6061 T6	1
4	ROD	ALSI 304	4
5	SPRING TUBE	3DM- Tough	1
6	LOAD CELL	ALSI 304	1
7	SODEMANN R203-504	ALSI 9254	1
8	GUIDE SPRING HIGH	PA GFR	1
9	ISO 4762 M4 x 16 - 14N		4
10	TR8X2 NUT	8514001G2	2
11	ISO 10642 - M3 x 10 - 10N		8
12	TR8X2, SCREW	ALSI 304	1
13	GUIDE SPRING LOW	316L	1
14	BEARING IGLIDUR	POLYMER	4
15	HAND WHEEL	PA GFR	1
16	LOAD CELL SPACER	PA GFR	1
17	ISOLATING GASKET	GRAPHITE	2
18	ISO 4027 - M4 x 6-N		4

ASSEMBLY AND INSPECTION CHECKLIST

DO NOT SCALE DIMENSIONS

SCALE: 1:1

DATE: 05/24/20

REV: 1

DESCRIPTION: Assembly manual Pouch cell tester

ITEM NO: 1.321

REV: A3

SHEET: 01 of 4

SOLIDWORKS Educational Product. For Instructional Use Only.

ITEM NO.	Description	MATERIAL	QTY.
1	BOTTOM	AL 6061 T6	1
2	MID	AL 6061 T6	1
3	ROD	ALSI 304	4
14	BEARING IGLIDUR	POLYMER	4
5	ISOLATING GASKET	GRAPHITE	2
6	ISO 4027 - M4 x 6-N		4

ASSEMBLY AND INSPECTION CHECKLIST

DO NOT SCALE DIMENSIONS

SCALE: 1:1

DATE: 05/24/20

REV: 1

DESCRIPTION: Assembly Part 1-2

ITEM NO: 1.321-1

REV: A3

SHEET: 01 of 4

SOLIDWORKS Educational Product. For Instructional Use Only.

**A** - Mount Iglus bearing (14) to mid plate (2) by applying a vertical force to the Controlle measure inner diameter of iglus bearing (14) to be less than Ø8.00mm

**B** - Pre-heat Bottom plate (1) to 120 degrees in an oven or similar. Do not use open flames, when this leads to poor heat controlle. Then fit rod(s)-(3) to the holes. Make sure flush surface to bottom plate(14) If necessary apply vertical Force to rod(S)(3) to make sure full contact. Then insert set screws (6) and apply an torque equal to 3 Nm

**C** - Remove glue cover from Isolating gasket(5) and assemble with glued surface against bottom (1) and mid (2). Make sure it is evenly attached across surface.

**3**

**4**

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	BOTTOM	AL 6061 T6	1
2	MID	AL 6061 T6	1
3	TOP	AL 6061 T6	1
4	ROD	AISI 304	4
5	SPRING TUBE	3DM- Tough	1
6	LOADCELL	AISI 304	1
7	SODEMANN R203-504	AISI 9254	1
8	GUIDE SPRING HIGH	PA GFR	1
9	ISO 4742 M4 x 16 - 16N		4
10	TR8X2.NUT	BS1400LG2	1
11	ISO 10442 - M3 x 10 - 10N		4
12	GUIDE SPRING LOW	316L	1
13	BEARING IGLIDUR	POLYMER	4
14	LOAD CELL SPACER	PA GFR	1
15	ISOLATING GASKET	GRAPHITE	2
16	ISO 4027 - M4 x 6-N		4

E- Mount item 2-6-12-7-8 in order as per view, be sure to get item 6-14-2 aligned in order to get outline of Loadcell (6) correct, there should not be applied any excessive force.

D- Screw M3x10 (11) to guide spring high (8) with torque = 1.8 Nm  
The fit between TOP(3) and guide spring high(8) should be glide fitted

F- Mount top(3) to the rod(s)(4) and slide down to suitable height due to thickness of battery cell, double check parallelness between (2) & (3) with parallel block. Then tight M4x16 to top (3) with torque 2-2,5 Nm

**Assembly Part 3-4**  
1-321.2 A3

SOLIDWORKS Educational Product. For Instructional Use Only.

**5**

**6**

Part no.	Description	Material	QTY.
15	HAND WHEEL	3DM- THOUGH	1
10	TR8X2.NUT	BL1200GS	1
11	ISO 10442 - M3 x 10 - 10N		4
12	TR8X2, SCREW	AISI 304	1

G- Mount item (10) gently to item (15), then insert screws (11) and torque with 1.5 Nm  
Do NOT overtight due to brittle material

H- Install thread bar (12) to Nut(10) and gently screw in touch with upper spring holder  
Check that the device runs smoothly and there's no rubbing.  
Connect cable from loadcell to Arduino.

I- Calibrate Load cell, this procedure is covered in Load cell document.

**Assembly Part 5-6**  
1.321.3 A3

SOLIDWORKS Educational Product. For Instructional Use Only.

# Vedlegg 7 : Maskintegninger (Org. filer lagt ved innlevering)

ITEM 6, 9, 11, 12, 18 are not included in drawings, ISO drawings available.

ASSEMBLY MANUAL 1.321 DESCRIBES STEPWISE MOUNTING INSTRUCTIONS  
ASSEMBLY MANUAL 1.321.1-2-3 DESCRIBES TORQUE, FORCES AND ASSEMBLY TEMPERATURES

ITEM NO.	Description	Material	QTY.
1	BOTTOM	AL 6061 T6	1
2	MID	AL 6061 T6	1
3	TOP	AL 6061 T6	1
4	ROD	AISI 304	4
5	SPRING TUBE	3DM- Tough	1
6	LOADCELL	AISI 304	1
7	SODEMANN R203-504	AISI 9254	1
8	GUIDE SPRING HIGH	PA GFR	1
9	ISO 4762 M4 x 16 - 16N		4
10	TRBX2.NUT	BS1400LG2	2
11	ISO 10642 - M3 x 10 - 10N		8
12	TRBX2, SCREW	AISI 304	1
13	GUIDE SPRING LOW	316L	1
14	BEARING IGLIDUR	POLYMER	4
15	HAND WHEEL	PA GFR	1
16	LOAD CELL SPACER	PA GFR	1
17	ISOLATING GASKET	GRAPHITE	2
18	ISO 4027 - M4 x 6-N		4

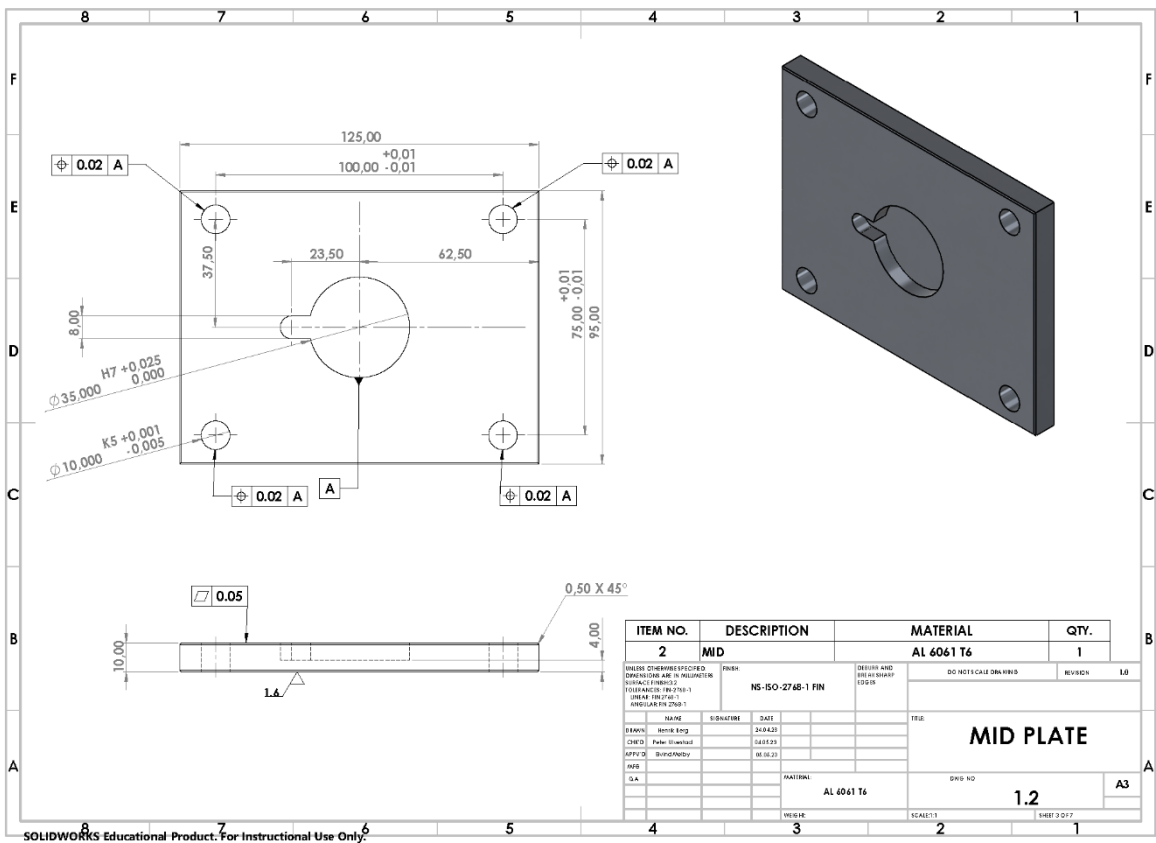
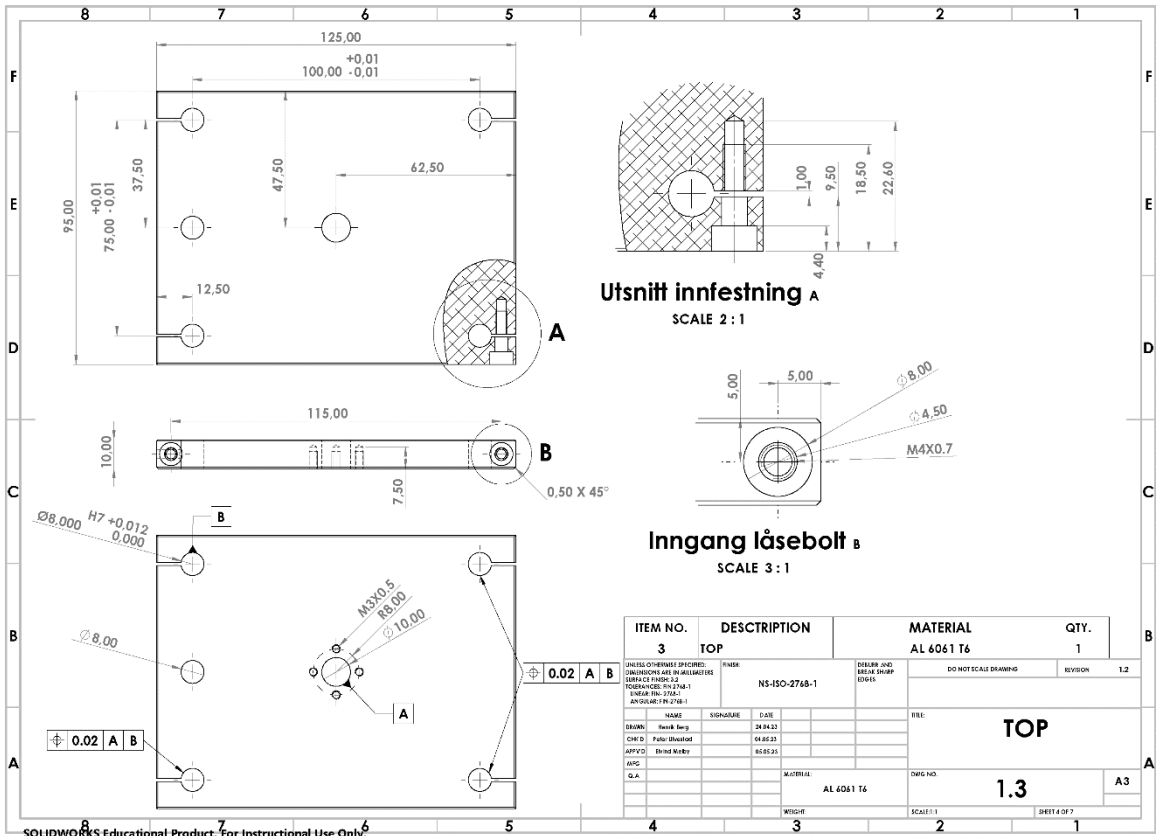
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILLIMETERS TOLERANCES: FINISHES: 1 SURFACE FINISH: 1 ANGULAR FINISH: 1	FINISH NS-ISO-2748-1 FIN	DESIGN AND ISSUE SHARP UPPER	DO NOT SCALE DRAWINGS	REVISION	LS
NAME	SIGNATURE	DATE	TITLE		
BRUNN	Bohrer Edg	24.04.23	ASSEMBLY		
CHW	Peter Strickland	03.05.23	Pouch cell tester		
APV	Bruce Andby	01.06.23	1.0		
MS			A3		
SA			REF: EGM		
SCALE: 1:1	SHEET 1 OF 7				

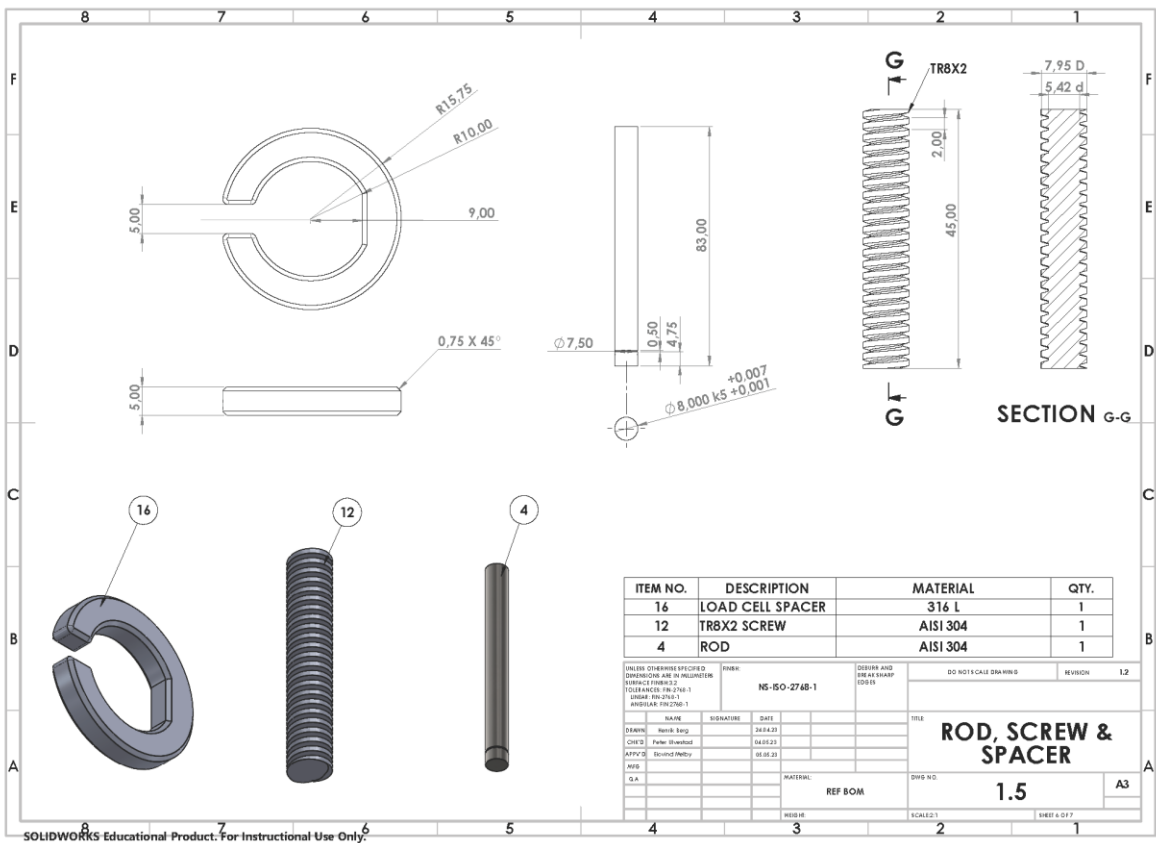
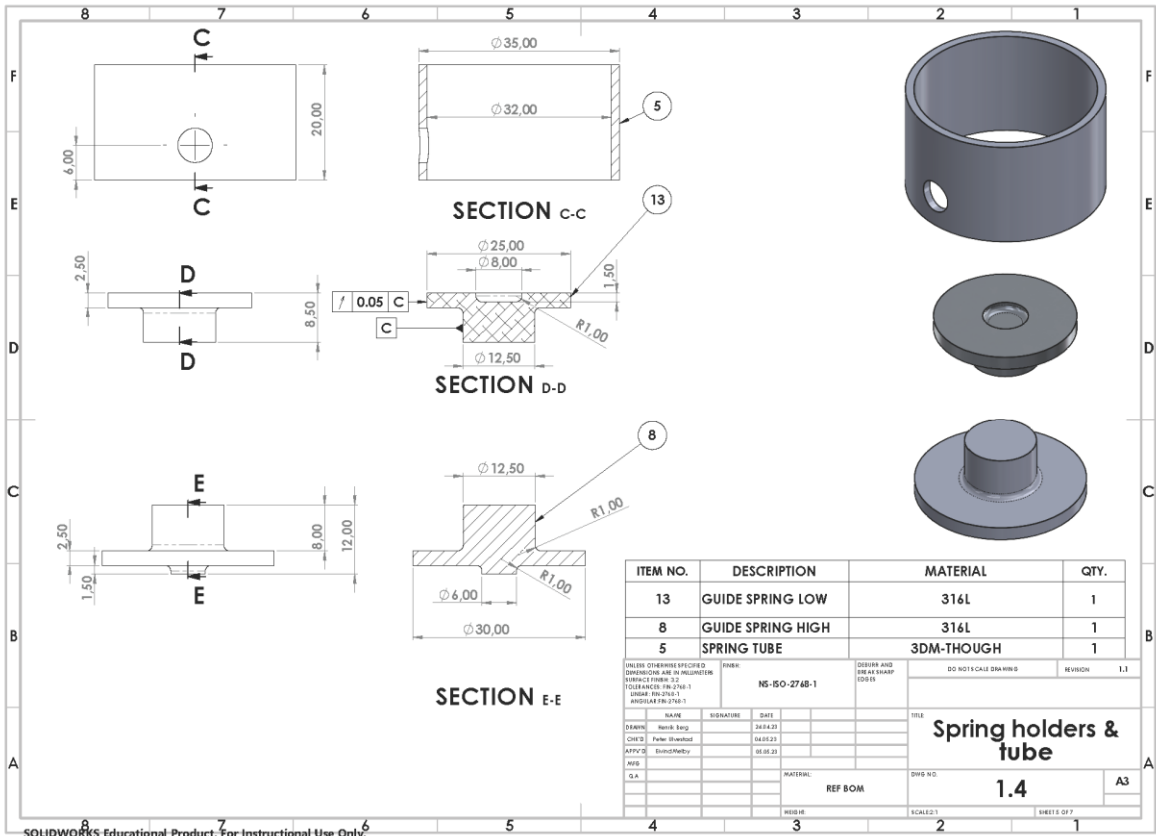
SOLIDWORKS Educational Product. For Instructional Use Only.

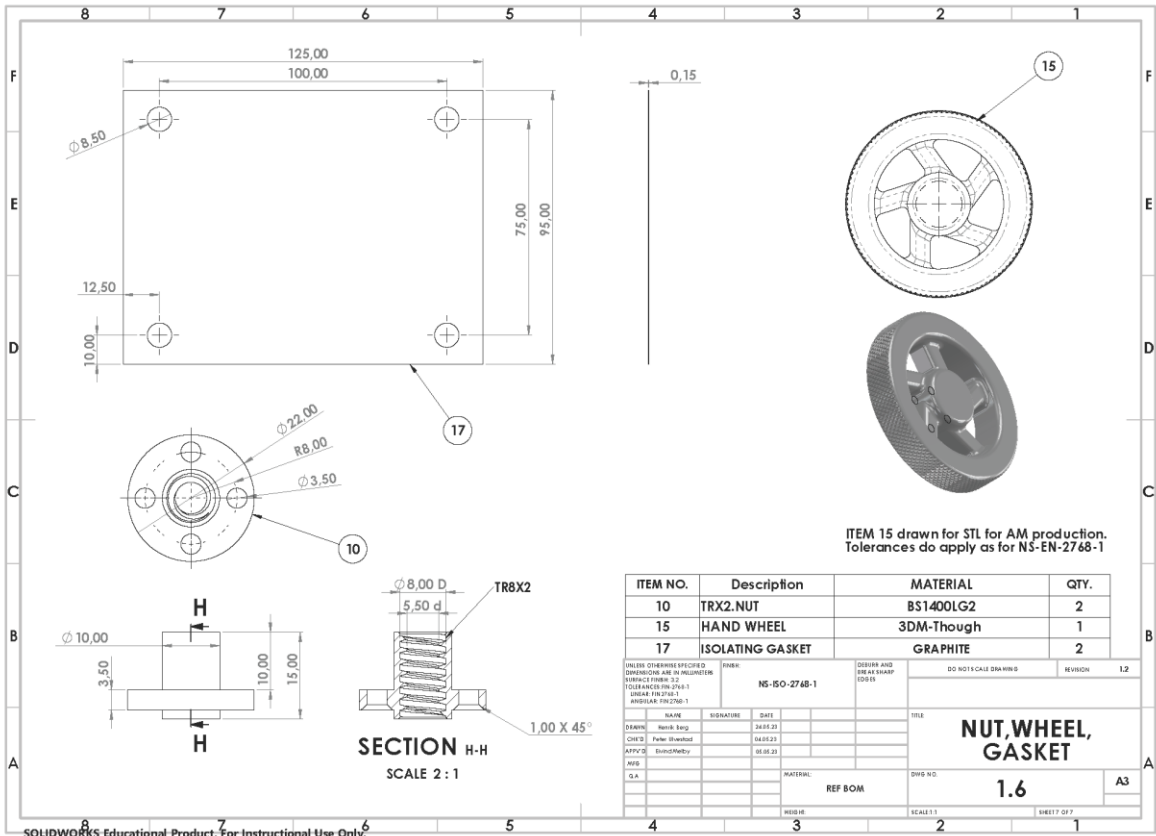
ITEM NO.	DESCRIPTION	MATERIAL	QTY.
1	BOTTOM	AL 6061 T6	1

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILLIMETERS TOLERANCES: FINISHES: 1 SURFACE FINISH: 1 ANGULAR FINISH: 1	FINISH NS-ISO-2748-1	DESIGN AND ISSUE SHARP UPPER	DO NOT SCALE DRAWINGS	REVISION	LS
NAME	SIGNATURE	DATE	TITLE		
BRUNN	Bohrer Edg	24.04	Bottom plate		
CHW	Peter Strickland	03.05.23	1.1		
APV	Bruce Andby	04.06.23	A3		
MS			REF: EGM		
SA			AL 6061 T6		
SCALE: 1:1	SHEET 2 OF 7				

SOLIDWORKS Educational Product. For Instructional Use Only.







## Vedlegg 8: Pseudokode

```

/*****
/
/           Hovedprogram - main
/*****
/ Programmet består av en setup() funksjon som starter alle sensorer og
/ lignende og en loop() funksjon som kjører hele tiden og utfører alle
/ oppgaver.
/*****

setup() {
  Start seriell kommunikasjon.
  Start opp shield.
  Start opp klokke.
  Start opp riktig antall sensorer.
}

/*****
/ loop() funksjonen kjører kontinuerlig så lenge arduinoen er på.
/ Dette er hovedprogrammet.
/ Programmet består av en hovedmeny som vises på skjermen. Her vises
/ den aktive sensoren og hvilket trykk den har nå i riktig benevnelse.
/ I tillegg til dette vises ett tall som er differansen mellom det
/ trykket cellen har nå og det trykket den opprinnelig ble satt opp
/ med. Dette er for å se om trykket har øket eller sunket siden
/ batteriet ble satt inn. Dette starttrykket kan lagres ved å trykke
/ lagre. Dette lagres så til fil.
/ Dersom man trykker på meny, kommer man til en undermeny der man kan
/ enten velge mellom å endre benevnelse eller kalibrere lastcellen
/ Vil man kalibrere cellen, trykke man på valg av kalibrering. Da vises
/ først en meny der cellen nullstilles uten vekt, deretter vises en
/ meny der cellen kalibreres ved å legge på en vekt. Kalibreringen
/ tar ca 1,5 minutt.
/ Velger man benevnelse kan man velge mellom Newton, Kg eller MPA
/ I tillegg lagrer programmet trykket fra cellene i en .csv fil som
/ kan åpnes i for eksempel Excel.
/*****
loop() {

  Les startrykk lagret i fil og lagre det i sensorklassen i den sensoren som vises på skjerm.

  if aktiv meny == "hovedmeny"{
    if målt trykk har endret seg{
      Skriv ut det nye trykket til skjerm.
    }

    Sjekk etter knappetrykk.

    if bruker har trykket på knappen meny{

```

```

    Sett aktiv meny til "meny"
  }
  if brukeren har trykket på knappen lagre{
    Lagre det nåværende trykket til fil, da dette er nytt startpunkt.
  }
  if brukeren har trykket joysticken opp{
    Vis informasjon fra neste sensor.
  }
  if brukeren har trykke joysticken ned{
    Vis informasjon fra forrige sensor.
  }
}

if aktiv meny == "meny"{
  Vis undermeny der brukeren kan velge mellom kalibrering og benevnelse.

  Sjekk etter knappetrykk.

  if brukeren har trykket på knappen tilbake{
    Gå tilbake til hovedmenyen.
  }
  if brukeren har trykket på knappen velg{
    Brukeren kan velge mellom kalibrering og benevnelse.
    Sett ny meny til valget brukeren tar.
  }
  if brukeren trykker enten opp eller ned på joysticken{
    Alternere mellom valgene brukeren kan velge.
    Valget som vises med grønn skrift på
    skjermen er det aktive valget.
    Aktiv meny settes enten til "benevnelse" eller til "kalibrering".
  }
}

if aktiv meny == "kalibrering"{
  Vis undermeny der brukeren skal nullstille sensoren.

  Sjekk etter knappetrykk.

  if brukeren trykker på knappen tilbake{
    sett aktiv meny til "meny"
  }
  if brukeren trykker på knappen velg{
    Vis menyen der brukeren får beskjed om å legge en
    kalibrert vekt på sensoren.
    Lagre faktoren sensoren oppgir når den helt
    avlastet til fil.
    Nullstill sensoren.

    while venter på knappetrykk{
      if brukeren trykker på knappen tilbake{

```



```

        gå tilbake til forrige skjerm.
    }
    if brukeren trykker på knappen velg{
        Kalibrer sensoren.
        Skriv ny kalibreringsfaktor til fil
    }
}
}
}

if aktiv meny == "benevnelse"{
    Vis undermeny med valg av benevnelse.

    Sjekk etter knappetrykk.

    if brukeren trykker på knappen tilbake{
        gå tilbake til menyen der brukeren kan velge mellom
        benevnelse eller kalibrering.
    }
    if brukeren trykker på knappen velg{
        benevnelsen settes til det valget brukeren
        har gjort og viser så fram hovedmenyen igjen.
    }
    if brukeren trykker på knappen opp på joysticken{
        vis grønn farge på den neste benevnelse.
    }
    if brukeren trykker på knappen ned på joysticken{
        vis grønn farge på forrige benevnelse.
    }
}

if minutt på klokken == valg satt av bruker. F.eks. 00,15,30 eller 45{
    Lagre det målet trykket til en CSV fil.
    Det som skal lagres er dato, klokkeslett og trykk.
}

if det oppdages en plutselig økning/senkning i trykk{
    Lagre det målet trykket til en CSV fil.
    Det som skal lagres er dato, klokkeslett og trykk.
}
}
}

```

## Vedlegg 9: Kildekode (Ligger vedlagt som fil)

### Hovedprogram - Main

```
1  /******  
2  /*          Hovedprogram - main          */  
3  /******  
4  /* Programmet består av en setup() funksjon som starter alle sensorer og */  
5  /* lignende og en loop() funksjon som kjører hele tiden og utfører alle */  
6  /* oppgaver.          */  
7  /* Nedenfor finnes alle variable som kan endres fritt av brukeren,    */  
8  /* innenfor de kriteriene som beskrives bak.          */  
9  /******  
10 /* Variable som kan endres av brukeren:          */  
11 #define ANTALL_SENSORER 1 // Antall sensorer som er koblet til arduino  
12 #define LOADCELL1_DOUT_PIN 2 //datapin ut for sensor1  
13 #define LOADCELL1_SCK_PIN 3 //klokkepin for sensor1  
14 //#define LOADCELL2_DOUT_PIN 7 //eksempel for sensor2. Hver sensor trenger  
15 //#define LOADCELL2_SCK_PIN 6 //to #define setninger med korrekt pin.  
16  
17 /*Forteller programmet hvilke pin hver sensor er koblet til. Det er lagt  
18 til eksempel på hvordan man kan legge til sensor 2          */  
19 SENSOR Sensor[ANTALL_SENSORER] = {SENSOR(LOADCELL1_DOUT_PIN, LOADCELL1_SCK_PIN)};  
20                                     //, SENSOR(LOADCELL2_DOUT_PIN, LOADCELL2_SCK_PIN)};  
21  
22 /*Sensornavnet er navnet som brukes ved lagring av filer. Kan ikke  
23 overskride 7 bokstaver, ettersom maks filnavn er 8 bokstaver*/  
24 String sensorNavn[ANTALL_SENSORER] = {"Sensor1"}; //, "Sensor2"};  
25  
26 /*Bestemmer klokkeslett når data skal skal lagres til SD kortet. Default  
27 er hel, kvart over, halv og kvart på. Fire lagringer per time. Dersom  
28 hyppigere lagring ønskes, så kan dette bare fylles inn, eks: {0,10,15,30,45}*/  
29 int lagreVedMinutt[] = {0,15,30,45};  
30  
31 /*Bestemmer at ekstra verdier til SD kortet skal lagres dersom trykk nivået  
32 overskrides raskt. Default er 0.1 som er 100g. Endres trykket i cellen mer enn  
33 100g på kort tid, lagres dette*/  
34 float threshold=0.1;  
35  
36 /*Bestemmer hvor ofte LCD skjermen skal oppdateres. Default er 0.005 og det  
37 vil si at dersom den verdien som er skrevet til skjerm er 0.005 forskjellig  
38 fra ny målt verdi, oppdateres skjermen. Det er for at skjermen ikke skal  
39 stå og blinke hver gang loop() løkken starter*/  
40 float sensitivitet = 0.005;  
41  
42 /*Setter hvilken benevnelse som skal brukes som default. 0 er Newton, 1 er kg  
43 og 2 er MPA          */  
44 int aktuellBenevnelse = 1;  
45  
46 /*Beskriver arealet på den aktuelle battericellen, oppgitt i meter          */  
47 float batteriAreal = 0.06 * 0.12;  
48  
49 /*Beskriver kalibreringsvekten som skal benyttes i kg          */  
50 float kalibreringsVekt=4.00;  
51  
52 /*Beskriver hvor lang pause arduinoen får etter ett knappetrykk. Ved lav          */  
53 /*verdi kan dobbeltrykking av knapper forekomme. 1000 = 1 sek          */  
54 int pause = 500;  
55  
56 /*Dette er hvor mye platen som ligger under lastcellen veier oppgitt i          */  
57 /*kg. Denne vekten må være med ettersom cellen kun måler vekt over denne          */  
58 /*platen          */  
59 float plate = 0.00;  
60  
61
```

```

61
62  /******
63  /* Globale variable som ikke skal endres */
64  /******
65  #define HOVEDMENY 0
66  #define MENY 2
67  #define KALIBRERING 3
68  #define BENEVNELSE 4
69  #define UNDERMENY 5
70  #define TILBAKE 6
71  #define LAGRE 7
72  #define VELG 8
73  #define OPP 9
74  #define NED 10
75  #define KALIBRERING2 11
76  #define TID 12
77  #define NEWTON 13
78  #define KG 14
79  #define MPA 15
80  String Benevnelse[] ={"N", "Kg", "Mpa"};
81  int sisteMinutt=0; //Tar vare på siste klokkeslett som ble lagret til SD kortet
82  int aktuellSensor=0;
83  int underMenyValg=BENEVNELSE;
84  int i=0;
85  int meny =HOVEDMENY; //Startmenyen er HOVEDMENY som er 0
86  float sensorTrykk[ANTALL_SENSORER]; //Tar vare på siste målte sensortrykk
87  //initialiserer ett objekt av type SHIELD. Denne inneholder all kode for
88  //shield kortet med lcd skjerm og kortleser.
89  SHIELD Shield = SHIELD();
90  //initialiserer ett objekt av type KLOKKE. Denne inneholder all kode for
91  //RTC kortet
92  KLOKKE Klokke = KLOKKE();
93
94
95
96  /******
97  /* setup() funksjonen kjøres en gang når arduinoen starter opp. Den */
98  /* setter alle startparametere og lignende */
99  /******
100 void setup() {
101     //Starter seriell kommunikasjon. Kan benyttes dersom man ønsker å kommunisere
102     //med arduinoen gjennom en usb kabel
103     Serial.begin(9600);
104     while (!Serial);
105
106     //kjører oppsett av Shield og Klokke klassene
107     Shield.setup();
108     Klokke.setup();
109
110     //Starter sensorene ved å lese kalibrert faktor og nullfaktor fra fil,
111     //og så starte sensoren
112     for(i; i < ANTALL_SENSORER; i++){
113         Sensor[i].setFaktor(Shield.lesFraFilFaktor(sensorNavn[i]));
114         Sensor[i].setNullFaktor(Shield.lesFraFilNullFaktor(sensorNavn[i]));
115
116         //Denne settes til 100 for å være sikker på at lcd skjermen skal starte
117         //utskrift
118         sensorTrykk[i] = 100.0;
119     }
120     Shield.setKalibreringsVekt(kalibreringsVekt);
121 }
122

```

```

122
123 /*****
124 /* loop() funksjonen kjører kontinuerlig så lenge arduinoen er på. */
125 /* Dette er hovedprogrammet. */
126 /* Programmet består av en hovedmeny som vises på skjermen. Her vises */
127 /* den aktive sensoren og hvilket trykk den har nå i riktig benevnelse. */
128 /* I tillegg til dette vises ett tall som er differansen mellom det */
129 /* trykket cellen har nå og det trykket den opprinnelig ble satt opp */
130 /* med. Dette er for å se om trykket har øket eller sunket siden */
131 /* batteriet ble satt inn. Dette starttrykket kan lagres ved å trykke */
132 /* lagre. Dette lagres så til fil. */
133 /* Dersom man trykker på meny, kommer man til en undermeny der man kan */
134 /* enten velge mellom å endre benevnelse eller kalibrere lastcellen */
135 /* Vil man kalibrere cellen, trykke man på valg av kalibrering. Da vises */
136 /* først en meny der cellen nullstilles uten vekt, deretter vises en */
137 /* meny der cellen kalibreres ved å legge på en vekt. Kalibreringen */
138 /* tar ca 1,5 minutt. */
139 /* Velger man benevnelse kan man velge mellom Newton, Kg eller MPA */
140 /* I tillegg lagrer programmet trykket fra cellene i en .csv fil som */
141 /* kan åpnes i for eksempel Excel.
142 /*****
143 void loop() {
144     int knappeTrykk;
145
146     //Starttrykket er det trykket cellen ble initiert med når brukeren trykker
147     //Lagre på hovedmenyen dette ble lagret til fil. Starttrykket hentes på nytt
148     //når arduinoen starter og den hentes fra en fil på SD kortet.
149     Shield.setStartTrykk(Shield.lesFraFilStartTrykk(sensorNavn[aktuellSensor]));
150
151     //Programmet starter med hovedmenyen
152     if(meny == HOVEDMENY){
153         //Serial.println("Hovedmeny");
154
155         //If setningen sjekker hvor mye differanse det er mellom forrige avlesning
156         //av trykk og det trykket som leses nå. Dersom trykket er større en
157         //definert sensitivitet, skrives den ut til skjerm. Dette er for at skjermen
158         //ikke skal stå å blinke hver gang loopen tar en runde.
159         if(sensorTrykk[aktuellSensor]-Sensor[aktuellSensor].getKg() >= sensitivitet ||
160            Sensor[aktuellSensor].getKg()-sensorTrykk[aktuellSensor]>=sensitivitet){
161
162             //Skriver til skjerm med benenvnensen i newton pluss platen under cellen.
163             if(aktuellBenevnelse == 0){
164                 Shield.visTrykk(Sensor[aktuellSensor].getNewton() , Benevnelse[aktuellBenevnelse],
165                    sensorNavn[aktuellSensor], plate, NEWTON, 0.0);
166                 sensorTrykk[aktuellSensor] = Sensor[aktuellSensor].getKg();
167             }
168             //Skriver til skjerm med benenvnensen i kg pluss platen under cellen.
169             if(aktuellBenevnelse == 1){
170                 Shield.visTrykk(Sensor[aktuellSensor].getKg() , Benevnelse[aktuellBenevnelse],
171                    sensorNavn[aktuellSensor], plate, KG, 0.0);
172                 sensorTrykk[aktuellSensor] = Sensor[aktuellSensor].getKg();
173             }
174             //Skriver til skjerm med benenvnensen i Mpa.
175             if(aktuellBenevnelse == 2){
176                 Shield.visTrykk(Sensor[aktuellSensor].getMpa(batteriAreal),
177                    Benevnelse[aktuellBenevnelse], sensorNavn[aktuellSensor], plate, MPA, batteriAreal);
178                 sensorTrykk[aktuellSensor] = Sensor[aktuellSensor].getKg();
179             }
180         }
181     }

```

```

181
182 //Sjekker om noen av knappene er trykket inn.
183 //Hvis noen er trykket inn, lagres dette i knappetrykk.
184 //Sjekker knappetrykk som hører til HOVEDMENY
185 knappeTrykk = Shield.getKnappetrykk(HOVEDMENY);
186 //Serial.println(knappeTrykk);
187
188 //Hvis brukeren trykker meny, endres menyen fra HOVEDMENY til MENY og
189 //undermenyen med benevnning og kalibrering skal vises videre.
190 if(knappeTrykk == MENY){
191     Serial.println("Meny");
192     meny = MENY;
193     delay(pause);
194 }
195 //Hvis brukeren trykker lagre, lagres trykket til fil. Filen får
196 //navnet "Sensornavn" + T.txt
197 //For eksempel Sensor1T.txt
198 else if(knappeTrykk == LAGRE){
199     Serial.println("Lagre");
200     Shield.lesTilFilStartTrykk(sensorNavn[aktuellSensor],Sensor[aktuellSensor].getKg())
201     sensorTrykk[aktuellSensor] =100;
202 }
203 //Hvis brukeren trykker opp på joysticken vises verdiene fra neste sensor
204 else if(knappeTrykk == OPP){
205     if(aktuellSensor < ANTALL_SENSORER-1){
206         aktuellSensor++;
207     }
208     else{
209         aktuellSensor=0;
210     }
211     delay(pause);
212 }
213 //Hvis brukeren trykker opp på joysticken vises verdiene fra forrige sensor
214 else if(knappeTrykk == NED){
215     if(aktuellSensor > 0){
216         aktuellSensor--;
217     }
218     else{
219         aktuellSensor = ANTALL_SENSORER-1;
220     }
221     delay(pause);
222 }
223 }
224
225 //Dersom meny = MENY skal menyen med benevnelse og kalibrering vises på skjermen
226 if(meny == MENY){
227     bool lagre = true;
228     Serial.println("Meny Skjerm");
229
230     //Viser skjermbildet med benevnelse og kalibrering. underMenyValg kan være enten
231     //BENEVNELSE eller KALIBRERING og dette avgjør hvilken av valgene som skal framstå
232     //med grønn farge.
233     Shield.visSubMeny(MENY, underMenyValg);
234

```

```

224
225 //Dersom meny = MENY skal menyen med benevnelse og kalibrering vises på skjermen
226 if(meny == MENY){
227     bool lagre = true;
228     Serial.println("Meny Skjerm");
229
230     //Viser skjermbildet med benevnelse og kalibrering. underMenyValg kan være enten
231     //BENEVNELSE eller KALIBRERING og dette avgjør hvilken av valgene som skal framstå
232     //med grønn farge.
233     Shield.visSubMeny(MENY, underMenyValg);
234
235     //Så lenge lagre = true sjekkes det for knappetrykk.
236     while(lagre){
237
238         //Leter etter knappetrykk som hører til menyen MENY
239         knappeTrykk = Shield.getKnappetrykk(MENY);
240         //Serial.println(knappeTrykk);
241
242         //Dersom brukeren trykker tilbake, settes meny til HOVEDMENY og underMenyValg til
243         //BENEVNELSE (slik at denne blir grønn dersom brukeren trykker meny igjen).
244         //Skjermen resettes for å fjerne all skrift. Sensortrykket settes til 100, slik at
245         //sensitivitetskriteriet i første if-setning skal slå til.
246         if(knappeTrykk == TILBAKE){
247             Serial.println("Tilbake");
248             Shield.flushScreen();
249             sensorTrykk[aktuellSensor] =100;
250             meny = HOVEDMENY;
251             underMenyValg = BENEVNELSE;
252             lagre = false;
253             delay(pause);
254         }
255         //Dersom brukeren trykker velg vil meny enten settes BENEVNELSE menyen eller
256         //KALIBRERING menyen,
257         else if(knappeTrykk == VELG){
258             if(underMenyValg == BENEVNELSE){
259                 meny = BENEVNELSE;
260                 underMenyValg=NEWTON;
261             }
262             else if(underMenyValg == KALIBRERING){
263                 meny = KALIBRERING;
264             }
265             lagre=false;
266             Serial.println("Velg");
267             delay(pause);
268         }
269         //Dersom brukeren trykker opp eller ned endres valgene fra BENEVNELSE til KALIBRERING
270         //og omvendt.
271         else if(knappeTrykk == OPP || knappeTrykk == NED){
272             if(underMenyValg == BENEVNELSE){
273                 underMenyValg = KALIBRERING;
274             }
275             else if(underMenyValg == KALIBRERING){
276                 underMenyValg = BENEVNELSE;
277             }
278             delay(pause);
279             lagre = false;
280             Serial.println("Opp");
281         }
282     }
283 }
284 }
285

```

```

285
286 //Dersom meny = KALIBRERING skal menyen der lastcellen skal nullstilles vises på
287 //skjermen
288 if(meny == KALIBRERING){
289     bool lagre = true;
290     Serial.println("Kalibreringsmeny");
291     underMenyValg = KALIBRERING;
292
293     //Viser skjermbildet der cellen skal nullstilles. Menyene er KALIBRERING og
294     //og underMenyValg er lik KALIBRERING.
295     Shield.visSubMeny(KALIBRERING, underMenyValg);
296
297     //Så lenge lagre = true
298     while(lagre){
299
300         //Sjekker knappetrykk som gjelder for menyen KALIBRERING
301         knappetrykk = Shield.getKnappetrykk(KALIBRERING);
302
303         //Dersom brukeren trykker tilbake, skal skjermen vise menyen med valg av
304         //kalibrering og benevnelse. underMenyValg settes til BENEVNELSE slik
305         //at denne teksten blir grønn.
306         if(knappetrykk == TILBAKE){
307             Serial.println("Tilbake");
308             meny = MENY;
309             underMenyValg = BENEVNELSE;
310             lagre = false;
311             delay(pause);
312         }
313
314         //Dersom brukeren trykker Velg skal lastcellen nullstilles. Nullfaktoren skal lagres til
315         //fil
316         else if(knappetrykk == VELG){
317
318             //Nullfaktoren finnes og leses til fil. Filnavnet blir sensornavn + N.txt.
319             Shield.lesTilFilNullFaktor(sensorNavn[aktuellSensor], Sensor[aktuellSensor].resetSensor());
320             bool lagreUndermeny = true;
321             underMenyValg = KALIBRERING2;
322
323             //Neste undermeny vises. Dette er menyen der lastcellen skal kalibreres med en vekt.
324             //Menyen er KALIBRERING og underMenyValg er KALIBRERING2
325             Shield.visSubMeny(KALIBRERING, underMenyValg);
326             delay(pause);
327
328             while(lagreUndermeny){
329
330                 //Sjekker knappetrykk forbundet med menyen KALIBRERING2
331                 knappetrykk = Shield.getKnappetrykk(KALIBRERING2);
332
333                 //Hvis brukeren trykker tilbake, settes meny til KALIBRERING igjen og menyen
334                 //med nullstilling av cellen vises.
335                 if(knappetrykk == TILBAKE){
336                     meny = KALIBRERING;
337                     lagreUndermeny = false;
338                     lagre = false;
339                     delay(pause);
340                 }
341

```

```

341
342 //Dersom brukeren trykker velg, regnes faktoren ut og lagres til fil. Deretter settes
343 //cellen opp med den nye faktoren. Menyen settes så til HOVEDMENY og underMenyValg
344 //settes til BENEVNELSE slik at den blir grønn når brukeren trykker videre.
345 if(knappeTrykk == VELG){
346     Sensor[aktuellSensor].kalibrerSensor(kalibreringsVekt);
347     Shield.lesTilFilFaktor(sensorNavn[aktuellSensor], Sensor[aktuellSensor].getFaktor());
348     Serial.println(Sensor[aktuellSensor].getFaktor());
349     lagreUndermeny=false;
350     lagre=false;
351     Shield.flushScreen();
352     meny=HOVEDMENY;
353     underMenyValg=BENEVNELSE;
354
355 }
356 }
357 lagre=false;
358 Serial.println("Velg");
359 }
360 }
361 }
362
363 //Dersom meny = BENEVNELSE vises menyen med valg av benevnelse
364 if(meny == BENEVNELSE){
365     Serial.println("Benevnelsesmeny");
366     bool lagre = true;
367
368     //Viser undermeny med BENEVNELSE og underMenyValg lik NEWTON. NEWTON ble satt da
369     //brukeren trykket velg i første meny.
370     Shield.visSubMeny(BENEVNELSE, underMenyValg);
371
372     //Så lenge lagre = true
373     while(lagre){
374         knappeTrykk = Shield.getKnappeTrykk(BENEVNELSE);
375
376         //Hvis brukeren trykker Tilbake, settes meny lik MENY og underMenyValg
377         //lik BENEVNELSE slik at teksten blir grønn når brukeren klikker videre
378         if(knappeTrykk == TILBAKE){
379             Serial.println("Tilbake");
380             //Shield.flushScreen();
381             meny = MENY;
382             underMenyValg = BENEVNELSE;
383             lagre = false;
384             delay(pause);
385         }
386         //Hvis brukeren trykker velg, endres benevnelse og man returneres til
387         //HOVEDMENYen
388         else if(knappeTrykk == VELG){
389             if(underMenyValg == NEWTON){
390                 meny = HOVEDMENY;
391                 aktuellBenevnelse=0;
392                 lagre = false;
393                 Shield.flushScreen();
394                 sensorTrykk[aktuellSensor] =100;
395                 delay(pause);
396             }

```



```

397     else if(underMenyValg == KG) {
398         meny = HOVEDMENY;
399         aktuellBenevnelse=1;
400         lagre = false;
401         Shield.flushScreen();
402         sensorTrykk[aktuellSensor] =100;
403         delay(pause);
404     }
405     else if(underMenyValg == MPA){
406         meny = HOVEDMENY;
407         aktuellBenevnelse=2;
408         lagre = false;
409         Shield.flushScreen();
410         sensorTrykk[aktuellSensor] =100;
411         delay(pause);
412     }
413 }
414
415 //Her endres hvilken tekst som skal bli grønn i undermenyen BENEVNELSE og
416 //brukeren trykker opp
417 else if(knappeTrykk == OPP){
418     if(underMenyValg == NEWTON){
419         underMenyValg = MPA;
420     }
421     else if(underMenyValg == MPA){
422         underMenyValg = KG;
423     }
424     else if(underMenyValg == KG){
425         underMenyValg = NEWTON;
426     }
427     delay(pause);
428     lagre = false;
429     Serial.println("Opp");
430 }
431 //Her endres hvilken tekst som skal bli grønn i undermenyen BENEVNELSE og
432 //brukeren trykker ned
433 else if(knappeTrykk == NED){
434     if(underMenyValg == NEWTON){
435         underMenyValg = KG;
436     }
437     else if(underMenyValg == KG){
438         underMenyValg = MPA;
439     }
440     else if(underMenyValg == MPA){
441         underMenyValg = NEWTON;
442     }
443     delay(pause);
444     lagre = false;
445     Serial.println("Ned");
446 }
447 }
448
449 }

```

```

450 //resetter variabelen med lagrede knappetrykk
451 knappeTrykk =1;
452
453 //for løkken løper like mange ganger som der er elementer i lagreVedMinutt[].
454 for(int i =0; i < (sizeof(lagreVedMinutt)/sizeof(int)); i++){
455 //Dersom nåtiden (nåværende minutt) er lik ett av elementene i lagreVedMinutt[]
456 //og den ikke allerede er skrevet til fil, skal dato, klokkeslett og avlest vekt
457 //lagres til en .csv fil. Filen får navnet sensornavn + .csv
458 if(Klokke.getMinutt() == lagreVedMinutt[i] & lagreVedMinutt[i] != sisteMinutt){
459     Shield.lesTilFilCSV(sensorNavn[aktuellSensor], Klokke.getDato(),
460         Klokke.getKlokkeslett(), Sensor[aktuellSensor].getKg() + plate, KG);
461     sisteMinutt = lagreVedMinutt[i]; //lagrer siste lagrede minutt, slik at vi ikke får duplikater i filen.
462 }
463 }
464
465 //dersom nåværende trykk endrer seg med høyere verdi en threshold skal dette lagres til fil.
466 //Dette er for å pukke opp eventuelle spikes i trykk
467 if(sensorTrykk[aktuellSensor]-Sensor[aktuellSensor].getKg() >= threshold ||
468     Sensor[aktuellSensor].getKg()-sensorTrykk[aktuellSensor] <= -threshold){
469     Shield.lesTilFilCSV(sensorNavn[aktuellSensor], Klokke.getDato(), Klokke.getKlokkeslett(),
470         Sensor[aktuellSensor].getKg() + plate, KG);
471 }
472 }

```

## Klasser

```

1 #include <HX711.h>
2 #include <RTClib.h>
3 #include <SPI.h>
4 #include <SD.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_ST7735.h>
7 #include <Adafruit_seesaw.h>
8 #include <Adafruit_TFTShield18.h>
9
10
11 /*****
12 /* Deklarasjon av klassen Shield */
13 /*****
14 /* Klassen inneholder alle metoder for å styre Adafruit 1.8" TFT shield. */
15 /* Bibliotek som må installeres er: */
16 /* Adafruit GFX library. */
17 /* Adafruit ST7735 and ST7789 library. */
18 /* Adafruit Seesaw library. */
19 /* Eksempelkode følger med bibliotekene. Eksempelkoden som er brukt mest */
20 /* er seesaw_shield18_test. */
21 /*****
22 /* Kodene er veldig modifisert, men forfatteren av den originale */
23 /* seesaw_shield18_test.ino krever at følgende tekst legges ved: */
24 /*****
25 This is an example sketch for the Adafruit 1.8" TFT shield with joystick
26 This example is for the Seesaw version
27 ----> http://www.adafruit.com/products/802
28
29 Check out the links above for our tutorials and wiring diagrams
30 These displays use SPI to communicate, 4 pins are required to
31 interface
32 One pin is also needed for the joystick, we use analog 3
33 Adafruit invests time and resources providing this open source code,
34 please support Adafruit and open-source hardware by purchasing
35 products from Adafruit!
36
37 Written by Limor Fried/Ladyada for Adafruit Industries.
38 MIT license, all text above must be included in any redistribution
39 *****/

```

```

40 class SHIELD {
41     private:
42         Adafruit_TFTShield18 seesaw;
43         // TFT display and SD card will share the hardware SPI interface.
44         // Hardware SPI pins are specific to the Arduino board type and
45         // cannot be remapped to alternate pins. For Arduino Uno,
46         // Duemilanove, etc., pin 11 = MOSI, pin 12 = MISO, pin 13 = SCK.
47         #define SD_CS 4 // Chip select line for SD card on Shield
48         #define TFT_CS 10 // Chip select line for TFT display on Shield
49         #define TFT_DC 8 // Data/command line for TFT on Shield
50         #define TFT_RST -1 // Reset line for TFT is handled by seesaw!
51         #define HOVEDMENY 0
52         #define MENY 2
53         #define KALIBRERING 3
54         #define BENEVNELSE 4
55         #define UNDERMENY 5
56         #define TILBAKE 6
57         #define LAGRE 7
58         #define VELG 8
59         #define OPP 9
60         #define NED 10
61         #define KALIBRERING2 11
62         #define TID 12
63         #define NEWTON 13
64         #define KG 14
65         #define MPA 15
66
67         //Variable for å ta vare på verdier som allerede
68         //har blitt skrevet til displayet
69         float lagret_trykk = 0.0;
70         String lagret_benevning="";
71         String lagret_endring="";
72         String lagret_sensornavn="";
73         String lagret_Knapp1="Omstart";
74         String lagret_Knapp2="Meny";
75         String lagret_Knapp3="Lagre";
76         float start_trykk = 80.0;
77         float kalibreringsVekt=5.0;
78
79
80         //initsialiserer TFT shield og setter riktig pins for kommunikasjon
81         Adafruit_ST7735 shield = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
82
83

```

```

84  /*****
85  /* hovedMeny() metoden tar imot trykk, endring, hvilken benevning og  */
86  /* navnet på sensoren som skal brukes og skriver ut til hovedmenyen.  */
87  /* Den skriver først ut de gamle tallene i svart for å resette displayet */
88  /* og deretter de nye i hvit tekst. Dette for at all tekst skal være ny  */
89  /* og oppdatert. Uten dette vil den gamle teksten henge igjen.      */
90  /*****
91  void hovedMeny(float trykk, float endring, String benevning, String sensornavn ){
92
93      int i=0;
94      shield.setTextColor(ST77XX_BLACK);
95
96      //While løkka går to ganger. Første gangen skal gammel tekst på skjermen
97      //fjernes. Dette gjøres ved å skrive den ut i svart. Den andre gangen
98      //skrives ny tekst.
99      while(i<=1){
100         shield.setTextSize(1);
101         shield.setCursor(10, 30);
102         shield.print(lagret_sensornavn);
103         shield.setTextSize(3);
104         shield.setCursor(10, 50);
105         shield.print(lagret_trykk);
106         shield.setTextSize(3);
107         shield.setCursor(120, 57);
108         shield.print(lagret_benevning);
109         shield.setTextSize(3);
110         shield.setCursor(10, 77);
111         shield.print(lagret_endring);
112         //Oppdaterer variablene med nye verdier slik at programmet
113         //skal huske hva som er skrevet til skjerm.
114         this->lagret_sensornavn = sensornavn;
115         this->lagret_endring = endring;
116         this->lagret_trykk = trykk;
117         this->lagret_benevning = benevning;
118         shield.setTextColor(ST77XX_WHITE);
119         i++;
120     }
121
122 }
123

```

```

123
124
125 /* *****/
126 /* meny() metoden tar imot to integer for å styre hvilken meny som skal */
127 /* skrives ut til skjerm. Disse integerne er fastsatt ved #define */
128 /* setningene overfor. Den skriver først ut de gamle tallene i svart for */
129 /* å resette displayet og deretter de nye i hvit tekst. Dette for at all */
130 /* tekst skal være ny og oppdatert. Uten dette vil den gamle teksten */
131 /* henge igjen. */
132 /* *****/
132 void meny(int menyvalg, int underMenyValg){
133     //Dersom man trykker på meny på hovedmenyen skrives denne menyen ut
134     //Den viser valg mellom Benevnelse og Kalibrering. Benevnelse skrives
135     //i grønt.
136     if(menyvalg == MENY & underMenyValg == BENEVNELSE){
137         shield.fillScreen(ST77XX_BLACK);
138         bakgrunnsMeny(2);
139         shield.setTextSize(2);
140         shield.setTextColor(ST77XX_GREEN);
141         shield.setCursor(10, 40);
142         shield.print("Benevnelse");
143         shield.setTextColor(ST77XX_WHITE);
144         shield.setCursor(10, 70);
145         shield.print("Kalibrering");
146     }
147     //Dersom brukeren trykker opp eller ned vil dette skrive ut denne
148     //menyen der kalibrering skrives i grønt.
149     if(menyvalg == MENY & underMenyValg == KALIBRERING){
150         shield.fillScreen(ST77XX_BLACK);
151         bakgrunnsMeny(2);
152         shield.setTextSize(2);
153         shield.setTextColor(ST77XX_WHITE);
154         shield.setCursor(10, 40);
155         shield.print("Benevnelse");
156         shield.setTextColor(ST77XX_GREEN);
157         shield.setCursor(10, 70);
158         shield.print("Kalibrering");
159     }
160     //Dersom brukeren har valgt menyen kalibrering, vises denne
161     //menyen der brukeren får beskjed om å ta av all vekt fra
162     //lastcellen.
163     if(menyvalg == KALIBRERING & underMenyValg == KALIBRERING){
164         shield.fillScreen(ST77XX_BLACK);
165         bakgrunnsMeny(2);
166         shield.setTextSize(1);
167         shield.setTextColor(ST77XX_WHITE);
168         shield.setCursor(10, 30);
169         shield.print("Nullstill.");
170         shield.setCursor(10, 50);
171         shield.print("Ta av all vekt.");
172         shield.setCursor(10, 70);
173         shield.print("Deretter trykk valg");
174     }

```

```

175 //Dersom brukeren har valgt menyen kalibrering, tatt av all vekt
176 //også trykket velg, vises denne menyen. Den forteller brukeren
177 //hvor mye vekt lastcellen skal belastes med.
178 if(menyvalg == KALIBRERING & underMenyValg == KALIBRERING2){
179     shield.fillScreen(ST77XX_BLACK);
180     bakgrunnsMeny(2);
181     shield.setTextSize(1);
182     shield.setTextColor(ST77XX_WHITE);
183     shield.setCursor(10, 30);
184     shield.print("Kalibrer");
185     shield.setCursor(10, 50);
186     shield.setTextColor(ST77XX_GREEN);
187     shield.print("Legg paa "+ String(kalibreringsVekt) + " Kg.");
188     shield.setTextColor(ST77XX_WHITE);
189     shield.setCursor(10, 70);
190     shield.print("Deretter trykk velg");
191 }
192 //Hvis brukeren har valgt benevnelse på den første undermenyen,
193 //skrives denne menyen ut der brukeren kan velge mellom benevnelserne
194 //Newton, kg eller MPa. Newton skrives i grønt.
195 if(menyvalg == BENEVNELSE & underMenyValg == NEWTON){
196     shield.fillScreen(ST77XX_BLACK);
197     bakgrunnsMeny(2);
198     shield.setTextSize(3);
199     shield.setTextColor(ST77XX_GREEN);
200     shield.setCursor(10, 30);
201     shield.print("Newton");
202     shield.setTextColor(ST77XX_WHITE);
203     shield.setCursor(10, 60);
204     shield.print("Kg");
205     shield.setTextColor(ST77XX_WHITE);
206     shield.setCursor(10, 90);
207     shield.print("Mpa");
208 }
209 //Hvis brukeren har valgt benevnelse på den første undermenyen,
210 //skrives denne menyen ut der brukeren kan velge mellom benevnelserne
211 //Newton, kg eller MPa. Kg skrives i grønt.
212 if(menyvalg == BENEVNELSE & underMenyValg == KG){
213     shield.fillScreen(ST77XX_BLACK);
214     bakgrunnsMeny(2);
215     shield.setTextSize(2);
216     shield.setTextColor(ST77XX_WHITE);
217     shield.setCursor(10, 30);
218     shield.print("Newton");
219     shield.setTextColor(ST77XX_GREEN);
220     shield.setCursor(10, 60);
221     shield.print("Kg");
222     shield.setTextColor(ST77XX_WHITE);
223     shield.setCursor(10, 90);
224     shield.print("Mpa");
225 }

```

```

226 //Hvis brukeren har valgt benevnelse på den første undermenyen,
227 //skrives denne menyen ut der brukeren kan velge mellom benevnelsene
228 //Newton, kg eller MPa. Mpa skrives i grønt.
229 if(menyvalg == BENEVNELSE & underMenyValg == MPA){
230     shield.fillScreen(ST77XX_BLACK);
231     bakgrunnsMeny(2);
232     shield.setTextSize(2);
233     shield.setTextColor(ST77XX_WHITE);
234     shield.setCursor(10, 30);
235     shield.print("Newton");
236     shield.setTextColor(ST77XX_WHITE);
237     shield.setCursor(10, 60);
238     shield.print("Kg");
239     shield.setTextColor(ST77XX_GREEN);
240     shield.setCursor(10, 90);
241     shield.print("Mpa");
242 }
243
244
245 }
246
247 /*****
248 /* flush() metoden sørger for å resette skjermen helt, uavhengig av hva */
249 /* som står der fra før. Skjermen blir helt svart ett sekund. */
250 /*****
251 void flush(void){
252     shield.fillScreen(ST77XX_BLACK);
253 }
254
255 /*****
256 /* bakgrunnsMeny() tar imot en integer for valg av bakgrunnsmeny. */
257 /* Bagrunnsmenyen er teksten som makerer knappene. D.v.s Omstart, Meny */
258 /* og Lagre også videre. Den skriver først ut gammel tekst i svart */
259 /* og deretter ny tekst i hvit */
260 /*****
261 void bakgrunnsMeny(int menyNummer){
262     seesaw.setBacklight(TFTSHIELD_BACKLIGHT_ON);
263     int i=0;
264     shield.setTextColor(ST77XX_BLACK);

```

```

265
266 //While løkken går to ganger. Første gang skrives gammel tekst ut med svart
267 //skrift slik at den blir borte. Deretter skrives ny tekst ut med hvit
268 //skrift
269 while(i<=1){
270     shield.setTextSize(1);
271     shield.setCursor(0, 0);
272     shield.print(lagret_Knapp1);
273     shield.setTextSize(1);
274     shield.setCursor(0, 120);
275     shield.print(lagret_Knapp2);
276     shield.setTextSize(1);
277     shield.setCursor(70, 120);
278     shield.print(lagret_Knapp3);
279     //Meny 1 er blandt annet menyen som vises sammen med hovedmenyen.
280     if(menyNummer==1){
281         //Oppdaterer variabelen med ny tekst
282         this->lagret_Knapp1 = "Omstart";
283         this->lagret_Knapp2 = "Meny";
284         this->lagret_Knapp3 = "Lagre";
285     }
286     //Meny 2 er menyen der brukeren skal velge mellom for eksempel
287     //benevnelse eller kalibrering, eller mellom N, Kg eller Mpa.
288     else if(menyNummer==2){
289         //Oppdaterer variabelen med ny tekst
290         this->lagret_Knapp1 = "Omstart";
291         this->lagret_Knapp2 = "Tilbake";
292         this->lagret_Knapp3 = "Velg";
293     }
294     shield.setTextColor(ST77XX_WHITE);
295     i++;
296 }
297 }
298
299 /*
300 /* lesTilFil() metoden tar imot en tekststring som skal være første del */
301 /* av navnet på en tekst fil, og hvilken verdi som skal lagres i denne. */
302 /*
303 void lesTilFil(String tekst, float verdi){
304     File tekstFil;
305     //Dersom filen allerede eksisteres, skal den slettes.
306     SD.remove(tekst + String(".txt"));
307
308     //Lager en ny fil med en tekststring metoden har tatt i mot. Den legger så på
309     //filekstensjonen .txt
310     tekstFil = SD.open(tekst + String(".txt"), FILE_WRITE);
311
312     //Dersom filen har blitt lagret, skrives mottatt verdi til filen også lukkes den
313     if (tekstFil){
314         tekstFil.print(verdi);
315         tekstFil.close();
316     }
317
318     //Hvis filen ikke eksisterer, skrives det ut melding om det.
319     else{
320         Serial.println(F("Kunne ikke skrive til fil"));
321     }
322 }
323

```



```

323
324
325 /* lesFrafil() metoden tar imot en tekststring som beskriver filnavnet */
326 /* på filen som skal åpnes. Deretter leses filen og informasjonen */
327 /* til en floatvariabel som så returneres. */
328 /******
329 float lesFraFil(String tekst){
330     File tekstFil;
331     float faktor =0.0;
332     //div brukes for å finne riktig sted for desimalene
333     float div=0.1;
334     bool bryter= false;
335
336     //Åpner fil med navn beskrevet i mottatt tekst + .txt
337     tekstFil = SD.open(tekst + String(".txt"), FILE_READ);
338     if (tekstFil) {
339
340         //Kjører så lenge det finnes innhold i filen
341         while (tekstFil.available()) {
342             //Leser en og en bokstav
343             char ch = tekstFil.read();
344
345             //Hvis vi har kommet til punktum, settes bryter til true,
346             //slik at vi kan finn ut hvor mange desimaler det er.
347             if (ch == '.'){
348                 bryter =true;
349
350             }
351             //Gjør om fra enkeltbokstaver til en float variabel
352             else if (ch >= '0' && ch <= '9') {
353                 faktor = (faktor * 10.00) + (ch - '0');
354             }
355             //Dersom bryter = true, øker vi div 10 ganger.
356             //Denne skal brukes til å dele faktoren slik at vi får
357             //riktig antall desimaler.
358             if(bryter){
359                 div = div * 10.0;
360             }
361         }
362         tekstFil.close();
363     }
364     //Hvis filen ikke kan åpnes
365     else {
366         Serial.println("Kan ikke åpne fil " + String(tekst));
367     }
368     //tar faktoren og deler med div. Div viser hvor mange desimaler
369     //vi har.
370     return faktor/div;
371
372 }
373
374

```

```

375 public:
376     SHIELD() {}
377
378     /*
379     /* setup() sørger for at TFT shield blir satt opp med riktig pin
380     /* konfigurasjon. Dette gjøres for skjermen og for SD kortet
381     /*
382     void setup () {
383         pinMode(TFT_CS, OUTPUT);
384         digitalWrite(TFT_CS, HIGH);
385         pinMode(SD_CS, OUTPUT);
386         digitalWrite(SD_CS, HIGH);
387
388
389         // Start seesaw helper chip
390         if (!this->seesaw.begin()){
391             Serial.println("seesaw kunne ikke startes!");
392             while(1);
393         }
394         Serial.println("seesaw started");
395
396         //Slår av belysningen for TFT shielden og resetter den.
397         seesaw.setBacklight(TFTSHIELD_BACKLIGHT_OFF);
398         seesaw.tftReset();
399
400         //Initsialiserer 1.8" TFT
401         shield.initR(INITR_BLACKTAB);
402         Serial.println("TFT OK!");
403
404         //Slår på belysningen
405         seesaw.setBacklight(TFTSHIELD_BACKLIGHT_ON);
406
407         //Setter bakgrunnsfargen til svart og roterer utskriften slik at den
408         //vises riktig vei
409         shield.fillScreen(ST77XX_BLACK);
410         shield.setRotation(1);
411
412         //Forsøker å starte SD kortmodulen
413         if (!SD.begin(SD_CS)) {
414             Serial.println(F("SD kortet feiler, eller er tatt ut!"));
415         }
416         else{
417             Serial.println(F("SD kortet er ok"));
418         }
419     }
420

```

```

420
421  /******
422  /* visTrykk() metoden tar imot trykk som skal vises (t), hvilken */
423  /* benevnelse som skal brukes og hva sensoren heter. Den regner ut hva */
424  /* differansen mellom vekten brukere har nullstilt trykket på ved å */
425  /* trykke på lagre på hovedmenyen og mottatt trykk. Dette vises sånn */
426  /* at brukeren kan se om trykket har endret seg etter at prosessen */
427  /* startet. */
428  /******
429  void visTrykk(float t, String benevning, String sensornavn, float plate, int trykktype, float areal) {
430  //regner ut differansen med hva starttrykket ble satt til og hva trykket
431  //er nå
432  float endring;
433
434  //viser bakgrunnsmenyen på skjermen
435  this->bakgrunnsMeny(1);
436
437
438  if(trykktype == NEWTON){
439  //regner ut differansen med hva starttrykket ble satt til og hva trykket
440  //er nå
441  endring= t - (start_trykk * 9.81);
442  //oppdaterer skjermbildet med oppdatert informasjon
443  this->hovedMeny(t + (plate * 9.81), endring, benevning, sensornavn);
444  }
445  else if(trykktype == MPA){
446
447  //regner ut differansen med hva starttrykket ble satt til og hva trykket
448  //er nå
449  endring= t - (((start_trykk * 9.81)/areal)/pow(10.0,6.0));
450  //oppdaterer skjermbildet med oppdatert informasjon
451  this->hovedMeny(t + (((plate * 9.81)/areal)/pow(10.0,6.0)), endring, benevning, sensornavn);
452
453  }
454  else{
455  //regner ut differansen med hva starttrykket ble satt til og hva trykket
456  //er nå
457  endring= t - start_trykk;
458  //oppdaterer skjermbildet med oppdatert informasjon
459  this->hovedMeny(t + plate, endring, benevning, sensornavn);
460  }
461
462
463
464
465  }
466
467  /******
468  /* flushScreen() resetter skjermen ved å slette all informasjon på den
469  /******
470  void flushScreen(void){
471  this->flush();
472  }
473

```

```

473
474
475 /* visSubMeny brukes for å vise fram undermenyene videre etter */
476 /* hovedmenyen. */
477
478 void visSubMeny(int menyvalg, int underMenyValg){
479     this->meny(menyvalg, underMenyValg);
480
481 }
482
483
484 /* getKnappetrykk() metoden tar imot info over hvilken meny som er aktiv */
485 /* på skjermen, for så å sjekke etter knappetrykk som gjelder for denne */
486 /* skjermen. Deretter returneres knappetrykket dersom den finnes. */
487
488 int getKnappetrykk(int meny){
489     //leser av hvilke knapper som brukeren har trykket inn
490     uint32_t buttons = seesaw.readButtons();
491
492     //Dersom aktiv meny er hovedmeny, er mulige knappetrykk
493     //meny, lagre, opp og ned på joysticken.
494     if(meny == HOVEDMENY){
495         if(! (buttons & TFTSHIELD_BUTTON_1)){
496             return MENY;
497         }
498         if(! (buttons & TFTSHIELD_BUTTON_3)){
499             return LAGRE;
500         }
501         if(! (buttons & TFTSHIELD_BUTTON_UP)){
502             return OPP;
503         }
504         if(! (buttons & TFTSHIELD_BUTTON_DOWN)){
505             return NED;
506         }
507     }
508     //Dersom aktiv meny er meny, dvs første undermeny, er
509     //mulige knappetrykk tilbake, velg, opp og ned på joysticken.
510     else if(meny == MENY){
511         if(! (buttons & TFTSHIELD_BUTTON_1)){
512             return TILBAKE;
513         }
514         if(! (buttons & TFTSHIELD_BUTTON_3)){
515             return VELG;
516         }
517         if(! (buttons & TFTSHIELD_BUTTON_UP)){
518             return OPP;
519         }
520         if(! (buttons & TFTSHIELD_BUTTON_DOWN)){
521             return NED;
522         }
523     }
524     //Dersom aktiv meny er kalibrering, er mulige knappetrykk tilbake
525     //og velg.
526     else if(meny == KALIBRERING){
527         if(! (buttons & TFTSHIELD_BUTTON_1)){
528             return TILBAKE;
529         }
530         if(! (buttons & TFTSHIELD_BUTTON_3)){
531             return VELG;
532         }
533     }

```

```

534 //Dersom aktiv meny er den andre menyen for kalibrering,
535 //er mulige knappetrykk tilbake og velg.
536 else if(meny == KALIBRERING2){
537     if(! (buttons & TFTSHIELD_BUTTON_1)){
538         return TILBAKE;
539     }
540     if(! (buttons & TFTSHIELD_BUTTON_3)){
541         return VELG;
542     }
543 }
544 //Dersom aktiv meny er benevnelse er mulige knappetrykk
545 //tilbake, velg, opp og ned på joysticken.
546 else if(meny == BENEVNELSE){
547     if(! (buttons & TFTSHIELD_BUTTON_1)){
548         return TILBAKE;
549     }
550     if(! (buttons & TFTSHIELD_BUTTON_3)){
551         return VELG;
552     }
553     if(! (buttons & TFTSHIELD_BUTTON_UP)){
554         return OPP;
555     }
556     if(! (buttons & TFTSHIELD_BUTTON_DOWN)){
557         return NED;
558     }
559 }
560 //dersom ingen knappetrykk er registrert returneres 1
561 return 1;
562 }
563
564 /******
565 /*  getStartTrykk() metoden returnerer trykket brukeren har satt ved      */
566 /*  å trykke lagre på hovedmenyen.                                         */
567 /******
568 float getStartTrykk(){
569     return start_trykk;
570 }
571
572 /******
573 /*  setStartTrykk() metoden lagrer trykket brukeren har satt ved        */
574 /*  å trykke lagre på hovedmenyen.                                         */
575 /******
576 void setStartTrykk(float trykk){
577     this->start_trykk = trykk;
578 }
579
580 /******
581 /*  lesTilFilFaktor() tar imot navnet på sensoren som er aktiv og hvilke  */
582 /*  faktor som skal lagres til fil.                                         */
583 /******
584 void lesTilFilFaktor(String sensornavn, float faktor){
585     this->lesTilFil(sensornavn, faktor);
586 }
587

```

```

587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644

```

```

/*****
/* lesTilFilStartTrykk() tar imot navnet på sensoren som er aktiv, legger*/
/* til en T på slutten av navnet og tilslutt hvilket trykk som skal */
/* lagres til fil. */
/*****
void lesTilFilStartTrykk(String sensornavn, float trykk){
    this->lesTilFil(sensornavn + String("T"), trykk);
}

/*****
/* lesTilFilNullFaktor() tar imot navnet på sensoren som er aktiv, legger*/
/* til en N på slutten av navnet og tilslutt hvilket faktor som skal */
/* lagres til fil. */
/*****
void lesTilFilNullFaktor(String sensornavn, long faktor){
    this->lesTilFil(sensornavn + String("N"), faktor);
}

/*****
/* lesTilFilCSV() tar imot navnet på den aktive sensoren, dato, */
/* klokkeslett, trykk og benevnelse og skal lagre dette i en CSV fil. */
/* CSV filen skal logge trykket på alle sensorene ved gjevne mellomrom. */
/* Innholdet i filen blir en linje for hver lagret trykk og kan se slik */
/* ut: */
/* 24.12.23;21:45;132.78 */
/*****
void lesTilFilCSV(String sensornavn, String dato, String tid, float trykk, int benevnelse){
    float trykkTilFil = trykk;
    String trykkString;
    String temp = "";

    //Dersom benvenelsen er Newton må dette regnes ut
    /*if(benevnelse == NEWTON){
        trykkTilFil = trykk * 9.81;
    }*/

    //lagrer trykket som er en float i et string array
    trykkString = String(trykkTilFil);

    //for løkken går så lenge det finnes bokstaver/tall i tekststrengel
    //når vi kommer til den 3. siste posisjonen må . byttes med ,
    //Det er fordi Excel bruker , i forbindelse med tall.
    for(int i = 0; i < trykkString.length(); i++){
        if(trykkString.length()-i == 3){
            temp = temp + ",";
        }
        else{
            temp = temp + trykkString[i];
        }
    }

    //Setter sammen en linje som består av dato, klokkeslett og trykk,
    //separert med semikolon
    String linje = dato + String(";") + tid + String(";") + temp + String("\n");

    File tekstFil;

```

```

644
645 //Åpner fil med riktig sensornavn .csv. Dersom den ikke eksisterer lages den
646 tekstFil = SD.open(sensornavn + String(".csv"), FILE_WRITE);
647
648 //Skriver en linje med målig til logfilen dersom filen eksisterer
649 if (tekstFil){
650     tekstFil.print(linje);
651     tekstFil.close();
652
653 }
654 //Hvis den ikke eksisterer
655 else{
656     Serial.println(F("Kunne ikke skrive til fil"));
657 }
658 }
659
660 /*****/
661 /* lesFraFilFaktor() metoden leser kalibreringsfaktoren fra fil. */
662 /* Sensornavnet benyttes for å finne riktig fil i forhold til spesifikk */
663 /* sensor. */
664 /*****/
665 float lesFraFilFaktor(String sensornavn){
666     return this->lesFraFil(sensornavn);
667 }
668
669 /*****/
670 /* lesFraFilStartTrykk() metoden leser starttrykket fra fil. */
671 /* Sensornavnet benyttes for å finne riktig fil i forhold til spesifikk */
672 /* sensor. */
673 /*****/
674 float lesFraFilStartTrykk(String sensornavn){
675     return this->lesFraFil(sensornavn + String("T"));
676 }
677
678 /*****/
679 /* lesFraFilNullFaktor() metoden leser nullfaktoren fra fil. */
680 /* Sensornavnet benyttes for å finne riktig fil i forhold til spesifikk */
681 /* sensor. */
682 /*****/
683 float lesFraFilNullFaktor(String sensornavn){
684     return this->lesFraFil(sensornavn + String("N"));
685 }
686
687 /*****/
688 /* setKalibreringsVekt() metoden setter riktig kalibreringsvekt. Dvs */
689 /* hvilken vekt som skal brukes til å kalibrere med. */
690 /*****/
691 void setKalibreringsVekt(float vekt){
692     this->kalibreringsVekt = vekt;
693 }
694 };
695

```

```

695
696  /******
697  /*           Deklarasjon av klassen Sensor           */
698  /******
699  /* Klassen inneholder alle metoder for å styre sensoren via SparkFuns */
700  /* load cell amplifier - HX711 kort. */
701  /* https://www.sparkfun.com/products/13879 */
702  /* Bibliotek som må installeres er: */
703  /* HX711 Arduino Library by Bogdan Necula. */
704  /* Eksempelkode følger med bibliotekene. Eksempelkoden som er brukt mest */
705  /* er eksempelkode fra Nathan Seidle: */
706  /* https://github.com/sparkfun/HX711-Load-Cell-Amplifier/tree/master/firmware
707  /******
708  class SENSOR {
709      private:
710
711      //pin på arduino for signaler
712      int dout_PIN;
713      //pin for HX711 klokke
714      int sck_PIN;
715      //initsiereing av klassen HX711
716      HX711 VektSensor;
717      float kalibreringsFaktor = 550000.0; //Ca riktig kalibreringsfaktor
718      long nullFaktor=0.0;
719      float start_trykk=0.0;
720      //float batteriAreal = 0.06 * 0.12;
721
722
723
724      public:
725      /******
726      /* SENSOR() konstruktøren tar imot verdier for hvilke pin som brukes på */
727      /* arduinoen. Dout er signaloverføring og SCK er klokkesignaler */
728      /* deretter settes riktig pin fra SparkFuns load cell amplifier kort */
729      /******
730      SENSOR(int DOUT, int SCK){
731          this->dout_PIN = DOUT;
732          this->sck_PIN = SCK;
733
734          //Starter loadcellen med riktig pinout
735          VektSensor.begin(dout_PIN , sck_PIN);
736      }
737
738      /******
739      /* setFaktor() metoden setter opp lastcellen med riktig faktor slik at */
740      /* signalene fra cellen settes opp riktig. */
741      /******
742      void setFaktor(long faktor){
743          //lagrer faktoren for senere bruk
744          this->kalibreringsFaktor = faktor;
745          //oppdaterer lastcellen med riktig faktor
746          VektSensor.set_scale(faktor);
747      }
748

```



```

748
749
750 /* setNullFaktor() metoden setter opp riktig faktor for å lese cellen */
751 /* uten vekt. Uten nullpunktet vil lastavlesningen forskyve seg. */
752 /*
753 void setNullFaktor(long faktor){
754     //lagrer faktoren for senere bruk
755     this->>nullFaktor = faktor;
756     //oppdaterer lastcellen med riktig nullfaktor
757     VektSensor.set_offset(faktor);
758 }
759
760
761 /* getKg() metoden returnerer avlest vekt i kg. Det er ingen konvertering*/
762 /* her ettersom lastcellen er kalibrert i kg. */
763 /*
764 float getKg(void){
765     return VektSensor.get_units();
766 }
767
768
769 /* getKg() metoden returnerer avlest vekt i Newton. Verdien må
770 /* konverteres ettersom lastcellen er kalibrert i kg. */
771 /*
772 float getNewton(){
773     return VektSensor.get_units() * 9.81;
774 }
775
776
777 /* getKg() metoden returnerer avlest vekt i Mpa. Verdien må
778 /* konverteres ettersom lastcellen er kalibrert i kg. */
779 /*
780 float getMpa(float areal){
781     return ((this->getNewton())/areal)/pow(10.0,6.0);
782 }
783
784
785 /* getFaktor() returnerer kalibreringsfaktoren som lastcellen er satt opp*/
786 /* med. */
787 /*
788 long getFaktor(void){
789     return kalibreringsFaktor;
790 }
791
792
793 /* resetSensor() metoden resetter lastcellen. Den må være helt avlastet */
794 /* for å vise riktig avlesning. */
795 /*
796 long resetSensor(void){
797     //nullstiller lastcellen
798     VektSensor.set_scale();
799     VektSensor.tare();
800
801     //leser av hvilken vekt/verdi lastcellen viser når den er helt avlastet
802     long faktor = VektSensor.read_average();
803     //setter ny nullfaktor
804     setNullFaktor(faktor);
805     Serial.println(faktor);
806     return faktor;
807 }

```

```

808
809
810 /* kalibrerSensor() tar imot vekten lastcellen skal kalibreres med. */
811 /* deretter beregnes riktig kalibreringsfaktor for cellen. */
812 /******
813 void kalibrerSensor(float kalibreringsVekt){
814     Serial.println("Starter kalibrering");
815     float faktorLav =0;
816     float faktorHoy;
817     float avlesning;
818
819     //Setter opp cellen med en forelopig for lav faktor
820     VektSensor.set_scale(faktorLav);
821     //Leser fra lastcellen
822     avlesning = VektSensor.get_units();
823
824     //Whileløkken går så lenge avlesningen er høyere en mottatt last.
825     //Det kan også legges til en ny while løkke der man sker med 100.0, for enda
826     //mer nøyaktig avlesning. Husk å trekke fra 1000 fra faktoren før ny while
827     //løkke startes. Det er usøyaktigheter i målingene så det er ikke hensiktsmessig
828     //å gjøre dette for nye.
829     while(avlesning > kalibreringsVekt){
830
831         //Forsøker med høyere faktor
832         faktorLav = faktorLav + 100.0;
833         //Setter ny faktor
834         VektSensor.set_scale(faktorLav);
835         //Leser ny avlesning
836         avlesning = VektSensor.get_units();
837         Serial.println(String(faktorLav) + " " + String(avlesning) );
838     }
839
840     /*faktorLav = faktorLav - 1000.0;
841     while(avlesning > kg){
842         //Forsøker med høyere faktor
843         faktorLav = faktorLav + 100.0;
844         //Setter ny faktor
845         VektSensor.set_scale(faktorLav);
846         avlesning = VektSensor.get_units();
847     }*/
848
849     //Vi har nå funnet lavre grense av faktoren som viser riktig vekt.
850     //Faktoren dobles før vi finner svre grense som viser riktig vekt
851     faktorHoy = faktorLav *2;
852
853     //Setter for høy faktor
854     VektSensor.set_scale(faktorHoy);
855
856     //løkken går så lenge avlesningen er mindre en kalibreringsvekten
857     while(avlesning < kalibreringsVekt){
858         //Forsøker med lavere faktor
859         faktorHoy = faktorHoy - 100.0;
860         //Setter ny faktor
861         VektSensor.set_scale(faktorHoy);
862         //Leser ny avlesning
863         avlesning = VektSensor.get_units();
864         Serial.println(String(faktorHoy) + " " + String(avlesning));
865     }
866

```

```

867 //Lagrer oppdatert kalibreringsfaktor. Regner ut gjennomsnittet
868 //mellom høy og lav faktor
869 setFaktor((faktorHoy + faktorLav)/2);
870 Serial.println((faktorHoy + faktorLav)/2);
871 }
872
873 /*****
874 /* setStartTrykk() metoden tar imot en verdi for starttrykk og lagrer */
875 /* denne verdien. */
876 /*****
877 void setStartTrykk(float trykk){
878     this->start_trykk = trykk;
879 }
880
881 /*****
882 /* getStartTrykk() metoden returnerer en verdi for starttrykk. */
883 /*****
884 float getStartTrykk(void){
885     return start_trykk;
886 }
887 };
888
889
890 /*****
891 /*           Deklarasjon av klassen Klokke           */
892 /*****
893 /* Klassen inneholder alle metoder for å styre klokken via Adafruits */
894 /* DS1307 Real Time Clock kort. */
895 /* https://www.adafruit.com/product/3296 */
896 /* Bibliotek som må installeres er: */
897 /* RTCLib by Adafruit */
898 /* Eksempelkode følger med bibliotekene. Eksempelkoden som er brukt mest */
899 /* er ds1307.ino */
900 /*****
901 class KLOKKE{
902     private:
903         RTC_DS1307 rtc;
904
905     public:
906         KLOKKE() {
907             }
908

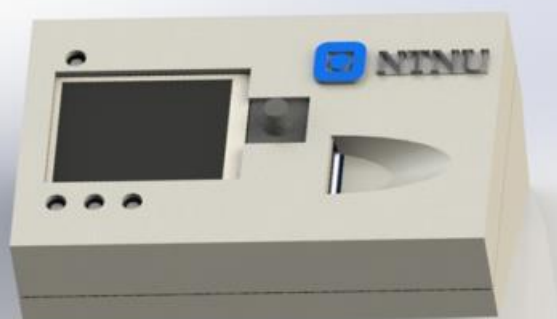
```

```

909  /* setup() kjører oppsett av klokkemodulen. Den stiller også klokken. */
910  /* Det vil si: Klokken stilles hver gang programmet lastes opp til */
911  /* arduinoen via usb kabelen. Den stilles etter klokken på pc'en som */
912  /* laster opp programmet. */
913  /* */
914  /* */
915  void setup(void){
916
917      //Forsøker å starte klokken. Hvis den ikke starter:
918      if (! rtc.begin()) {
919          Serial.println(F("Kan ikke starte klokke"));
920          Serial.flush();
921          while (1) delay(10);
922      }
923      //Hvis den starter normalt
924      else{
925          Serial.println("Klokken starter");
926      }
927      /*if (! rtc.isrunning() {
928          Serial.println("RTC is NOT running, let's set the time!");
929      }*/
930      //Denne setningen stiller RTC modulen med riktig tid fra pc'en
931      //som laster opp programmet.
932      rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
933  }
934
935  /* */
936  /* */
937  /* */
938  /* */
939  String getDate(void){
940      return String(rtc.now().day()) + String(".") + String(rtc.now().month()) + String(".") + String(rtc.now().year());
941  }
942
943  /* */
944  /* */
945  /* */
946  /* */
947  String getKlokkeslett(void){
948      return String(rtc.now().hour()) + String(":") + String(rtc.now().minute());
949  }
950
951  /* */
952  /* */
953  /* */
954  int getMinutt(void){
955      return rtc.now().minute();
956  }
957  };

```

## Vedlegg 10: Styringsenhet / Control unit



Meny Lagre

-3.77  
0.88 Kg

# Brukermanual

User manual

# Quick guide

## Oppstart / Startup



Start styringsenheten ved å koble den til en PC eller USB-Hub med en USB-kabel.

Start the control unit by hooking it up to a PC or USB-Hub using an USB-cable.

## Hovedmenyen / Start menu



Omstart	➡	Omstart av enhet / Restarts unit
Sensor1	➡	Nummer på aktiv sensor / Number of the active sensor
0.88 kg	➡	Vekt på lastcellen i kg / Weight on the load cell in kg
-3,77	➡	Endring av trykk fra start / Change in pressure from the start.
Meny	➡	Meny / menu Lagre* ➡ Lagre trykk / Save pressure.

\* Lagrer starttrykket ved starten av ny test / Saves the pressure at the start of a new test

## Velge aktiv sensor / Select active sensor



Aktiv sensor endres ved å trykke joysticken opp eller ned

Active sensor can be changed by pressing the joystick up or down

## Undermeny / Submenu



Trykk på meny for å komme til første undermeny. Velg kalibrering eller benevnelse ved å bruke joysticken og trykk velg.

Push the menu button to get to the submenu. Choose kalibrering to calibrate the loadcell or benevnelse to change units. Use the joystick to choose then press velg.

## Benevnelse / Units



Velg mellom Newton, Kg eller MPa ved å bruke joysticken. Trykk deretter velg.

Choose between Newton, Kg or MPa by using the joystick. Press velg to save.

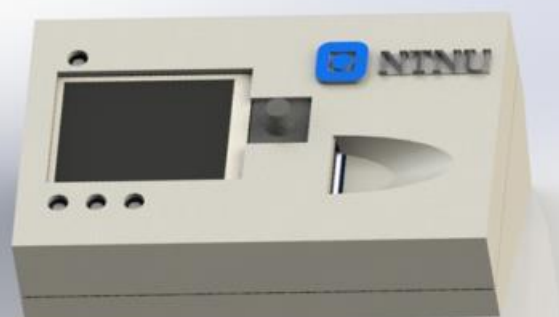
## Kalibrering / Calibrate



Ta av all vekt av lastcellen. Trykk velg. Legg på angitt kalibreringsvekt. Trykk velg.

Remove all weight from the load cell. Press velg. Add the weight stated in the display. Press Velg.

## Vedlegg 11: Loggføring av data / Data logging




# Brukermanual

User manual

# Quick guide

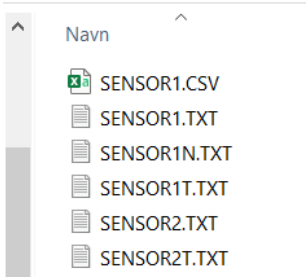




## SD-Kort / SD card

 Ta ut microSD kortet fra kontrollenheten. Sett det i en kortleser som kan leses av en PC.

Remove the microSD card from the control unit. Put it in a card reader that can be read from a PC.

---


## Filene på SD-kortet / The files on the SD-card

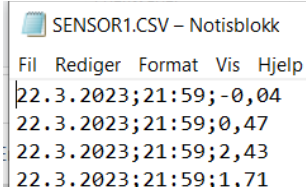
	<p>Sensor1.csv  Inneholder all logget data for sensor 1 / Contains all logged data from sensor 1.</p> <p>Sensor1.txt  Kalibreringsfaktor / Calibration factor</p> <p>Sensor1N.txt  Nullfaktor / Zero factor</p> <p>Sensor1T.txt  Lagret trykk / Saved pressure</p>
--	---

---

## .CSV – Comma Separated Values File

Innholdet i Sensor1.csv er logget data fra sensor 1. Dataen er i utgangspunktet logget kvart over, halv, kvart på og hver hele time. I tillegg er data lagret dersom de endrer seg mye over kort tid. All data er separert i filen med semikolon.

 Sensor1.csv contains all data logged from sensor 1. All of the data are logged fifteen past, half past, a quarter to and each whole hour. In addition, all data are logged if they change a lot in a short time. All data is separated by semicolon.



```
SENSOR1.CSV – Notisblokk
Fil Rediger Format Vis Hjelp
22.3.2023;21:59;-0,04
22.3.2023;21:59;0,47
22.3.2023;21:59;2,43
22.3.2023;21:59;1,71
```

---

## Excel

 .csv filen kan åpnes i Excel og bearbejdes videre.

The .csv file can be opened by Excel.



## Vedlegg 12: Programmering



# Brukermanual

User manual

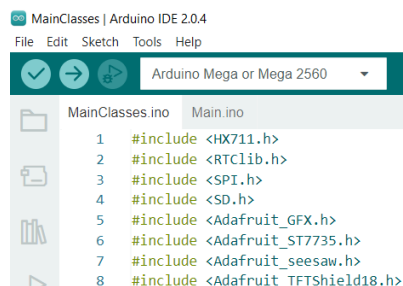
# Quick guide

---

## Arduino IDE



- Programmet Arduinoen er programmert i er Arduino IDE / The software used with the Arduino is Arduino IDE.
- Koble Arduinoen opp til en PC med en USB-kabel. / Hook Arduino to a PC using an USB cable.
- Åpne opp filene MainClasses.ino og Main.ino. Rekkefølgen er viktig. MainClasses.ino må lastes opp først og stå til venstre. / Open the files MainClasses.ino and Main.ino. The order is important. MainClasses.ino must be opened first and be on the leftmost pane.



```
MainClasses | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
MainClasses.ino Main.ino
1 #include <HX711.h>
2 #include <RTCLib.h>
3 #include <SPI.h>
4 #include <SD.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_ST7735.h>
7 #include <Adafruit_seesaw.h>
8 #include <Adafruit_TFTShield18.h>
```

---

## Bibliotek / Libraries



Følgende bibliotek er brukt av programmet / The following libraries are used by the program:

- **SD by Arduino/Sparkfun** – Verson 1.2.4
- **Adafruit BusIO by Adafruit** – Version 1.14.1
- **Adafruit GFX Library by Adafruit** – Version 1.11.5
- **Adafruit ST7735 and ST7789 library by Adafruit** – Version 1.10.0
- **Adafruit Seesaw library by Adafruit** – Version 1.6.3
- **HX711 by Rob Tillaart** – Version 0.3.6
- **HX711 Arduino Library by Bogdan Necula, Andreas Moti** – Verson 0.7.5
- **RTCLib by Adafruit** – Version 2.1.1

---

## Seriell kommunikasjon / Serial communication



Ved kalibrering av lastcellen vil Arduinoen skrive nullfaktoren og kalibreringsfaktoren til Arduino IDE ved å trykke Serial Monitor. Disse tallene finnes også i filene på SD kortet / While calibrating the load cell, the Arduino will print the zero factor and the calibrating factor to Arduino IDE by clicking Serial Monitor. This info can also be found on the SD card.

```
Output Serial Monitor x
Message (Enter to send message to 'Arduino Mega or Me

seesaw started
TFT OK!
SD kortet er ok
Klokken starter
Meny
Meny Skjerm
Opp
Meny Skjerm
Velg
Kalibreringsmeny
3805
```

Skjerm bilde fra Arduino IDE. Her er nullfaktoren 3805 / Screenshot from Arduino IDE. The zero factor is 3805

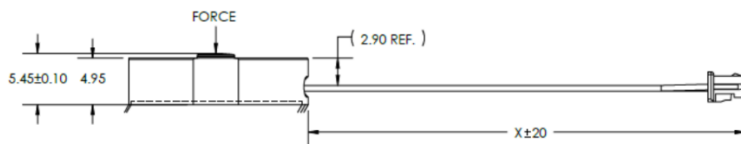
```
Output Serial Monitor x
Message (Enter to send message to 'Arc

15100.00 3.80
15000.00 3.82
14900.00 3.83
14800.00 3.86
14700.00 3.88
14600.00 3.92
14500.00 3.94
14400.00 3.96
14300.00 4.01
14250.00
14250
```

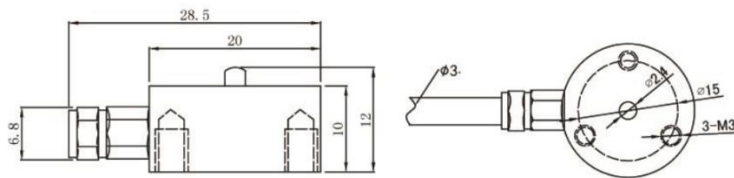
Skjerm bilde fra Arduino IDE. Her er kalibreringsfaktoren 14250 / Screenshot from Arduino IDE. The calibration factor is 14250

### Vedlegg 13: Oppsett av lastcelle test

For å starte testen trengtes en del utstyr. Det må finnes en metode der trykket kan settes på cellen og at den skal kunne ligge der uten at den må støttes. Det ble å henge vekten under cellen. Derfor må det konstrueres en form for krok eller lignende som kan overføre trykket til toppen av cellen, som i tillegg er så sterk at den tåler vekt uten å bøye seg. I tillegg er trykkplaten på cellen svært liten og vanskelig å sentrere vekten på, da den ikke må komme nær cellehuset.

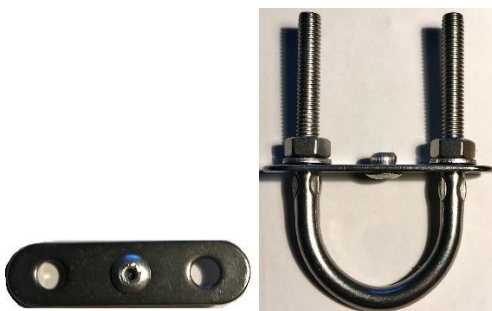


Figur 62: Dimensjoner for lastcellen fra TE Connectivity (TE Connectivity, 2020).



Figur 63: Dimensjoner for lastcellen kjøpt fra Sparkfun (HTC-Sensor.com, u.å.),

Løsningen på dette var å få tak i en u-bolt eller bøylebolt med muttere. Deretter kan det bores et hull i selve platen og deretter sette i en poppnagle. Hodet på poppnaglen skal passe ganske bra over lastcellen.



Figur 64: U-bolt med popnagle









Deretter kan vekter henges under lastcellen.




Vektene som skal brukes må veies først slik at målingene blir mest mulig nøyaktige. For å måle disse brukes to vekter. En baderomsvekt og en kjøkkenvekt.

Tabell 14:  
Oversikt over måleinstrumenter for vekt.

Type	Vektnavn	Aktuelt måleområde	Usikkerhet
Tanita RD-901	A	0-100kg	D=0.05kg
EK4150-S Camry	B	0-5kg	D=0.001kg

Tabell 15:  
Oversikt over alle vektskiver og lignende som er brukt

Vektnummer	Type	Målt med vekt	Vekt	Bilde
1	Ubolt med kjetting	A og B	0,55kg	
2	Vektskive i betong	A	9.90kg	
3	Vektskive i betong	A	9.85kg	
4	Vektskive i stål	A	5.05kg	
5	Vektskive i stål	A	5,00kg	
6	Vektskive i betong og stål	A	19.95kg	
7	Vektskive i betong og stål	A	19.95kg	
8	Kjetting og flaske med vann	B	1.00kg	

9	Kjetting og flasker med vann	B	2.00kg	
10	Kjetting og flasker med vann	B	3.00kg	
11	Kjetting og flasker med vann	B	4.00kg	

Oppsett av forsøket gjøres ved å bruke en skiftenøkkel som settes fast med en tvinge. Lastcellen tapes fast til skiftenøkkelen og ubolten med kjetting passer over denne.



Figur 65: Oppsett av lastcelle test

Ved testing leses nullfaktor av ved hjelp av Arduino IDE som nevnt i forrige kapittel. Dette kan også leses ved å ta ut SD kortet etter testing og lese av verdien i filene SENSOR1.txt (kalibreringsfaktor) og SENSOR1N.txt (nullfaktor).

