

Cover page

Exam information

29013PE208 - MIBAD-01 Build a Device

Handed in by


Gustav Jentzch Seeberg
201808905@post.au.dk

Exam administrators

Helle Elbæk Alfastsen
hea@au.dk
 +4541893064

Assessors

Claus Melvad
Examiner
cme@mpe.au.dk
 +4523244761

Mikkel Bo Nielsen
Internal co-examiner
mbn@mpe.au.dk
 +4593508781

Hand-in information

Profiling system for freshwater lakes

MIBAD F21

Af Gustav Seeberg - 201808905

Vejleder: Claus Melvad



Indhold

Indledning.....	4
Baggrund for problemstillingen.....	4
Problemformulering	5
Problem analyse	5
Problemformulering	5
Kravspecifikationer	5
Problem afgrænsning	5
Koncept beskrivelse.....	6
Morphologi diagram	6
Beskrivelse af ideen	7
Del på bøjen.....	8
Del under vandet	8
Canistor og kabel	8
Design overvejelser	9
Montage på CB150	9
Dimensionering og valg af komponenter	11
Opdrifts budget	11
Dimensionering af kabel spil	12
Design af kabel spil	12
Dimensionering af motor	15
Power Budget	16
Drift tider	16
Modstand i ledning.....	17
Wirering på bøjen.....	18
Wirering i Canistor.....	19
Modul-tests og forsøg	20
Aflæsning af temperatur- og tryk sensor	20
Afstands test med LORA.....	21
Aflæsning af CH4 sensoren.....	23
Bestemmelse af peak current til steppermotor	24

Datakommunikation mellem UNO og MEGA	25
Beskrivelse af koden	26
Flow diagram for Uno	26
Flow diagram for Mega	27
Trådløs indstilling af stepper motor	27
Økonomi og opskalering	28
Diskussion	29
Konklusion	29
Perspektivering	30
Appendiks	31
A1 - Subsystemer og point givning	31
A2 - Aflæsning af temperatur og tryk sensor (I2C)	33
A3 - Klient og server kode til trådløs kommunikation med LORA	34
A4 - Aflæsning af CH4 sensor med MAX232	36
A5 - Bestemmelse af peak current for stepper motor	36
A6 - UART kommunikation mellem UNO og MEGA	38
A7 - Trådløs indstilling af stepper motor	40
A8 - Bom liste med komponenter	48

Indledning

Baggrund for problemstillingen

Det ses at klimaforandringerne har stor betydning for de naturlige processer der finder sted over alt på jorden. Her påtænkes særligt den naturlige lagring og udledning af drivhusgasser fra ferskvandssøer. Udledningen fra ferskvandssøer er særligt interessant da data tyder på at metan (CH₄) udledningen er mindst lige så stor som CH₄ udledningen fra husdyr/kreaturer. Hvor de store og dybe søer er bedre undersøgt, er data fra mindre lavvandede søer mere sparsomme. Lavvandede søer er af særlig interesse at få undersøgt bedre, da de kun udgør 9 % af det totale areal, mens de vurderes at udlede 15% af CO₂ og 40 % af CH₄ udledningen fra alle søer.

På grund af de lave vandstande, er udledningerne meget omskiftelige. De lavvandede søer har desuden tendens til lagdeling, hvori gasserne ophobes for efterfølgende at blive udledt, når lagdelingen brydes. De omskiftelige udledninger betyder, at en hyppig dataopsamling er nødvendig for at opnå et repræsentativt billede på udledningen af drivhusgasser. Almindelig manuel data opsamling vil ofte være utilstrækkelig, da man ikke kan forudsige hvornår udledningen finder sted og derved sjældent får den dokumenteret.

Problemformulering

Problem analyse

Der skal designes og konstrueres en robot, som kan foretage målinger i de forskellige dybdeniveauer af ferskvandssøer. Målingerne omfatter CO₂, CH₄, temperatur og tryk. Robotten skal integreres på en eksisterende bøjle af typen CB150 eller CB450. Robotten skal benyttes af forskere inden for limnologi, med henblik på at beskrive udledningen af drivhusgasser. De dybeste søer i Danmark er 30 meter, mens den ønskede sø i Silkeborg, hvor forskerne fra AU holder til, er 5 meter dyb.

Problemformulering

I forbindelse med projektet, skal følgende bestemmes:

- Hvordan sænkes sensorerne ned i søen?
- Design og konstruktion af op-/nedsænkings system.
- Dimensionering og valg af passende motorer.
- Hvordan monteres op-/nedsænkings systemet på bøjlen?
- Design og konstruktion af casing til sensorerne/elektronikken.
- Hvordan gøres casing til sensorerne/elektronikken vandtæt?
- Udarbejdelse af power-budget for de valgte komponenter.
- Design og konstruktion af opspolings system til data- og strøm kabler.

Kravspecifikationer

Der er sat følgende krav til robotten:

- Sensorerne skal kunne sænkes ned til 5 meters dybde.
- Datakommunikation og indstilling af robotten skal foregå trådløst vha. LORA.
- Antallet af samples pr. dag skal kunne reguleres.
- Tidsintervallerne i målingerne skal kunne reguleres.
- Nedsænkningen af sensorerne skal kunne reguleres. Dette involverer antallet af stop undervejs, den fysiske afstand imellem stoppene og tiden et stop tager.
- Robotten skal kunne integreres på bøjlerne: CB450 og CB150.

Problem afgrænsning

Der er afsat 10 ECTS-point til kurset. Der er et budget på 10.000 kr. Systemet vil blive kodet og styret vha. Arduino. Der fokuseres på proof of concept og arbejdes hen imod at lave en funktions dygtig prototype. Prototypens funktioner er valideret vha. modul-test. Det er ikke sikkert at al software og kodning fungerer 100 % optimalt, da C++ og Arduino programmering er ukendt ved projektets start. Der vil blive valgt en stepper motor fra skolens lager. CO₂ sensoren er ikke medtaget i løsningen, da denne endnu ikke var købt af forskerne.

Koncept beskrivelse

I dette afsnit vil den indledende designfase og de brugte metoder blive gennemgået.

Morphologi diagram

I forbindelse med udarbejdelse af robottens opbygning er der gjort brug af morphologi diagram og den design proces som følger heraf. Denne metode består af 7 steps og vil kort blive beskrevet:

I det første step er robottens opbygning blevet opdelt i forskellige subsystemer. Dette indebar:

1. Op/ned-sænkningssystem af sensorer (casing)
2. Motorvalg
3. Op/ned-sænkningssystem til data og strømkabler
4. Casing til elektronikken under vand

I step 2 er løsninger til de forskellige subsystemer blevet brainstormet ukritisk. Disse fremgår af appendiks A1.

I step 3 er der opsat relevante kriterier for de enkelte subsystemer. Som det fremgår af subsystem 1, ved op/nedsænkningssystemet til sensorerne, var det eksempelvis ønsket at opnå en vandtæt konstruktion som kunne styres med Arduino og samtidig ikke blev for kompleks. Dette fremgår af nedenstående tabel 1.

Subsystem 1 (Op-/nedsænkningssystem af sensorer)

	Ideer	Vægtning	Kompleksitet	Vandtæt	Styring med Arduino	SUM
1.1	Kravle på kabel		1	1	1	10
1.2	Kabel tromle (bøje)		5	3	3	36
1.3	Kabel tromle (sensorpakke)		3	1	3	20

Tabel 1: Subsystem 1

I step 4 blev der tildelt point og vægtning til de forskellige kriterier. Den samlede vægtning er 10. Der gives point mellem 1 og 5, hvor 5 er bedst. Ved subsystem 1 ses det eksempelvis at vandtæthed vægtes højest, da dette er kritisk for konstruktionen. I appendiks A1 ses den samlede vægtning og pointtildeling for subsystemerne.

I step 5 er selve morphologi diagrammet opsat. Ideerne fremgår af tabel 2, mens diagrammet ses på tabel 3.

Ide nr.	Beskrivelse
1.2	1. Kabeltromle sidder på bøjen
1.3	1. Kabeltromle sidder på casing
2.1	Stepmotor
3.2	2. Kabeltromle sidder på bøjen
3.3	2. Kabeltromle sidder på casing
4.1	3D-printet
4.2	Casing i PVC-rør
4.3	Indkøbt del (Canistor Blue Robotics)

Tabel 2: Oversigt over ideer

Subsystem 1 (Op-/ned sænkningssystem af sensorer)	1.2		1.3
Subsystem 2 (Motorvalg)	2.1		
Subsystem 3 (Op-/ned sænkningssystem af data/strømkabler)		3.2	3.3
Subsystem 4 (Casing til elektronik under vandet)	4.1	4.2	4.3

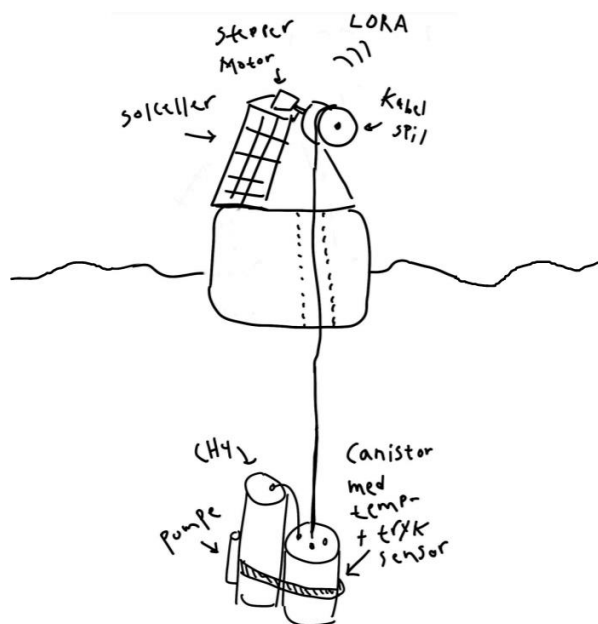
Tabel 3: Subsystemer og ideer

I step 6 er der fundet sammenhænge. I stedet for at have to op-/nedsænkningssystemer, blev muligheden for at slå disse sammen undersøgt. Det viste sig her, at man godt kunne nøjes med ét system hvor man hæver og sænker casingen vha. data/power kablerne.

Spørgsmålet gik nu på hvorvidt dette "op-/nedsænkningssystem" skulle sidde på bøjen eller under vandet på casingen. Ulemperne ved at have det under vandet består i at der dels opstår en masse modstand, ved rotation af ens spil. Desuden er det kompliceret at sikre vandtæt casing til motoren, samt at løsningen ikke er særlig energi effektiv da man ville skulle trække en høj strøm igennem ens kabel ned til motoren. Det oplagte valg blev derfor at montere denne på bøjen.

Beskrivelse af ideen

I dette afsnit vil løsningen kort blive gennemgået.



Figur 1: Overblikstegning af ideen

Løsningen som morfologi diagrammet endte ud i, kan bedst beskrives som en to delt anordning, hvor man har en del på bøjen og en del under vandet. Over vandet styres kabelspillet samt kommunikationen ind til land, mens der under vandet foretages målinger med sensorerne.

Del på bøjen

På bøjen kan tilgås 12 V batteri som lades af eksisterende solcelle paneler. Der er 28 ampere timer til rådighed ved CB150 bøjen og 56 ampere timer ved CB450 bøjen. Delen på bøjen består af en Arduino MEGA som er tilkoblet et LORA-modul således at denne kan tilgås med den eksisterende LORAWAN som er på sitet i Silkeborg. Arduino'en er ligeledes opkoblet til en motordriver, således at steppermotoren med kabelspillet kan indstilles. Kablet består af 8 ledninger og fungerer både som strømforsyning, samt datakommunikation. Motoren er dimensioneret efter at have et minimums moment på 1 N*m, således at der kan løftes 1 kg under vandet i en situation med mest kabel på tromlen. Som standard vil der ikke være strøm til stepper driveren/motor og canistoren. Det er først når MEGA'en modtager en gyldig instruks i form af en 3 cifret kode at der åbnes for relæerne til de to. På denne måde opnås en energi effektiv løsning.

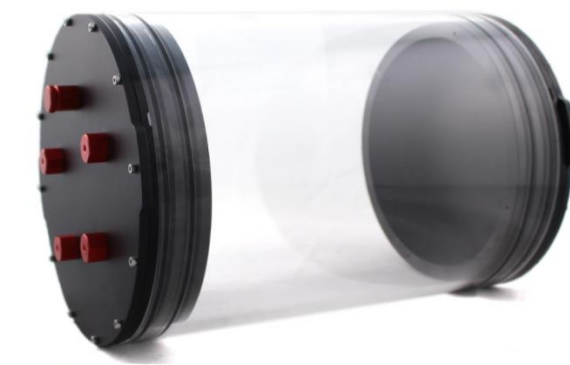
Del under vandet

Delen under vandet består af en canistor fra Blue Robotics. Inde i canistoren er en Arduino UNO som er tilkoblet de ønskede sensorer. Sensorerne indebærer temperatur, tryk og CH4. Arduinoen og sensorerne forsynes med strøm gennem kablet. Når data aflæses af UNO'en sendes disse videre til MEGA'en vha. UART kommunikation. På bøjen sendes data retur til land gennem LORA-modulet.

Det smarte ved denne løsning er at den er skalerbar samt fleksibel, forstået på den måde at man kan tilføje eller ændre udvalget af sensorer efter behov. Ender man i en situation, hvor UNO'en ikke har nok inputs, må denne udskiftes til en MEGA. Begrænsningerne består af strømforbrug og canistorens vægt. I og med at man hæver og sænker under vand, kan der vindes en del på opdrift. Et sådant opdriftsbudget vil blive gennemgået senere.

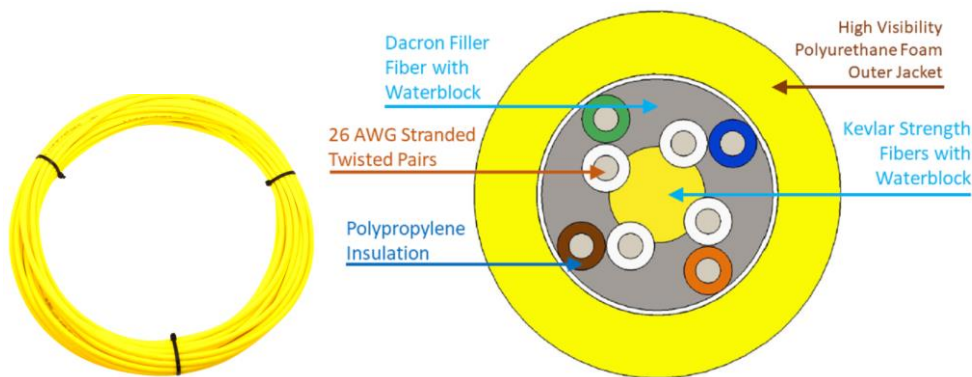
Canistor og kabel

På nedenstående billede ses canistoren fra Blue Robotics, der er valgt. Fordelen ved dette valg er at man opnår en vandtæt konstruktion, samt at der findes propper til kabel gennemføring. Desuden er temperatur og tryksensorerne special lavet til canistoren, således at de kan monteres i låget.



Figur 2: Cannister fra Blue Robotics (ROV8)

Kablet fra Blue Robotics er tiltænkt undervands applikationer og har derfor neutral opdrift. Det består af 4 ledningspar af 26 AWG standard og kan derfor teoretisk holde til 2,2 A pr. ledning. Af de 4 par, vil 2 blive brugt til datakommunikation og de sidste 2 vil blive brugt til strømforsyning. Kablet har en brudstyrke på 155 kg, mens arbejdsstyrken (working strength) er 35 kg. Selv over vand, vil applikationen ikke blive så tung, hvorfor kablet er egnet. Modstanden gennem kablet er 0,127 ohm pr. meter.



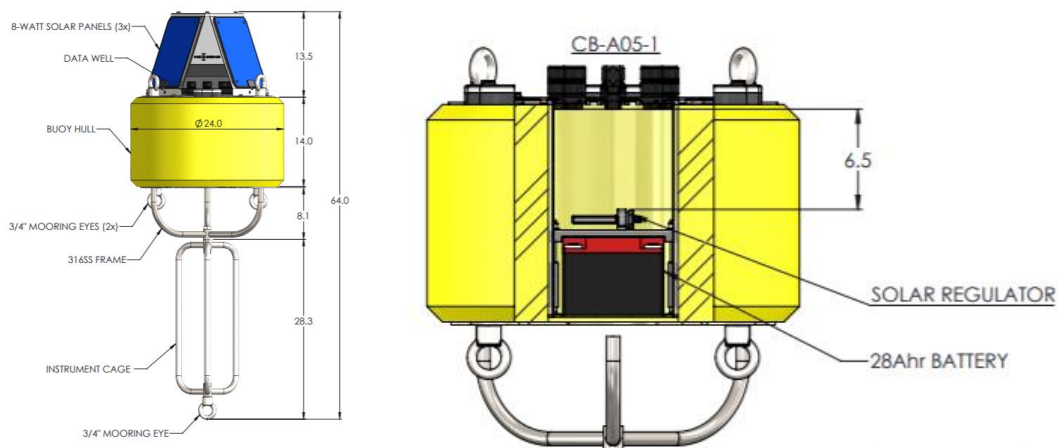
Figur 3: Teather fra Blue Robotics

Design overvejelser

I dette afsnit beskrives design overvejelserne i forbindelse med montage af elektronik og kabel spil.

Montage på CB150

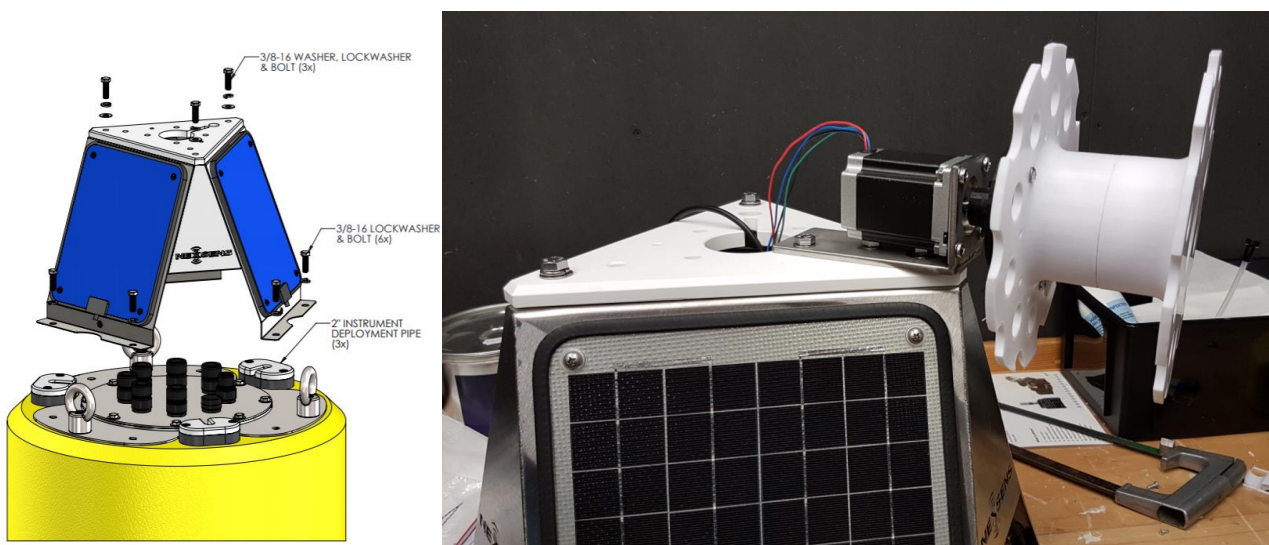
Forskerne i Silkeborg har tilsendt bøjen CB150, så montage kan afprøves. Bøjen ser ud på følgende måde:



Figur 4: Overblik over bøjen CB150

Bøjen har et vandtæt rum, hvor elektronikken placeres og tilkobles 12 V batteriet. Kabelføring til stepper motoren, samt eventuel antenne til forøgelse af LORA-modulets rækkevidde kan trækkes igennem kabelpropperne.

Der er lavet et beslag, således at stepper motoren kan monteres på oversiden af bøjen. Der skal printes et hus til motoren, så denne kan skånes fra vind og vejr. Se nedenstående:



Figur 5: Montage af prototype

Kablet kan da fires ned igennem en af de 3 sensor huller på bøjen. Se nedenstående:



Figur 6: Overblik over nedsæknings valget

Ideen med at nedsænke igennem sensor hullet er at man har styr på sit kabel, samt at udhænget bliver så småt som muligt. Hvis kabel blev sænket ned på siden af bøjen, ville det kunne flyde væk og hive i kabeltromlen fra alle sider.

Dimensionering og valg af komponenter

I dette afsnit vil fremgangsmåden til bestemmelse af motoren blive beskrevet. Der vil blive gennemgået opdriftsbudget og dimensionering af kabelspil.

Opdrifts budget

Der udarbejdes opdrifts budget på baggrund af canistoren og sensorernes vægt og volumen. Kablet er ikke medtaget, da dette har neutral opdrift.

Komponent	Masse	Volumen
Methan sensor	12,5 kg	$V_{CH_4} = \pi \cdot \left(\frac{136}{2} mm\right)^2 \cdot 494 mm = 7,176 L$
Pumpe til methan sensor	0,7 kg	$V_{pumpe} = 0,4 L$
CO2 sensor	0,5 kg	$V_{CO_2} = \pi \cdot \left(\frac{53,4}{2} mm\right)^2 \cdot 280 mm = 0,627 L$
Canistor (ROV 8 serie)	4,25 kg	$V_{ROV8} = \pi \cdot \left(\frac{216}{2} mm\right)^2 \cdot 322 mm = 11,799 L$
Temperatur sensor	20 g	Ubetydeligt
Tryk sensor	20 g	Ubetydeligt
Elektronik indvendigt i canistor	1 kg	
Total:	$m_{tot} = 17,49 kg$	$V_{tot} = V_{CH_4} + V_{pumpe} + V_{ROV8} = 19,375 L$

Tabel 4: Opdrifts budget

Det påpeges at CO2 sensoren ikke indgår i udregningen, da denne ikke var modtaget i Silkeborg. Der antages en densitet for vand på $1 \frac{g}{cm^3}$, svarende til massefylden ved 5 C. Den samlede opdrift er følgende:

$$m_{opdrift} = m_{tot} - V_{tot} \cdot 1 \frac{g}{cm^3} = -885 g$$

Det ses at der skal tilføres ca. 1 kg til canistoren før denne vil kunne synke.

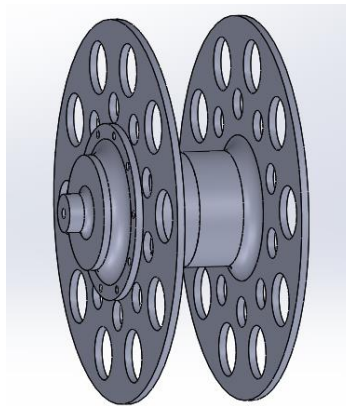
Der tilføjes ekstra vægt indvendigt i canistoren således at den totalt kommer til at veje max 1 kg under vandet.

Dimensionering af kabel spil

Kabelspilet tåntænkes i en sådan størrelse, at det ville kunne benyttes til samtlige søer i Danmark. For at kunne dække samtlige søer, skal kablet da være 30 meter langt. Kablet som benyttes har en diameter på 7,6 mm, hvilket giver følgende volumen:

$$V_{kabel} = \pi \cdot \left(\frac{7,6}{2} \text{ mm}\right)^2 \cdot 30 \text{ m} = 1,361 \text{ L}$$

Kablet har en minimum bukkeradius på 75 mm. Kabelspillet tænkes udført som en tromle med sider til at forhindre kablet i glide af. Et udkast til kabelspillet ses på nedenstående figur:



Figur 7: Udkast til kabelspil

Den indre tromle tænkes udført med en ydre radius på 80 mm. Dette er lidt mere end minimumsværdien, og vurderes passende. Siderne på kabelspillet ønskes med så stor diameter som muligt, for at undgå en meget lang kabeltromle og dermed mere vrid på motorakslen.

Spillet er opbygget af 3 dele og tænkes 3D printet for hurtig prototyping. Grundet bed-size på printerne er sidernes diameter begrænset til 200 mm. Der vælges en passende bredde på spillet, således at man tager højde for at kablet aldrig vil pakke sig perfekt om tromlen:

$$V_{spil} = \pi \cdot (100 \text{ mm})^2 \cdot 80 \text{ mm} - \pi \cdot (40 \text{ mm})^2 \cdot 80 \text{ mm} = 2,111 \text{ L}$$

Det ses at man har ca. halvanden gange kablets volumen på tromlen. Dette antages passende, da det ikke vil pakke sig helt tæt ved oprulning.

$$\frac{V_{spil}}{V_{kabel}} = 1,55$$

Efter at have printet kabeltromlen ses det at 30 meter kabel kun lige kan vikles omkring. En endnu større tromle ville da være at foretrække. Til søen i Silkeborg, hvor dybden kun er 5 meter er kabelspillet rigeligt stor, såfremt at man kun monterer 5 meter kabel.

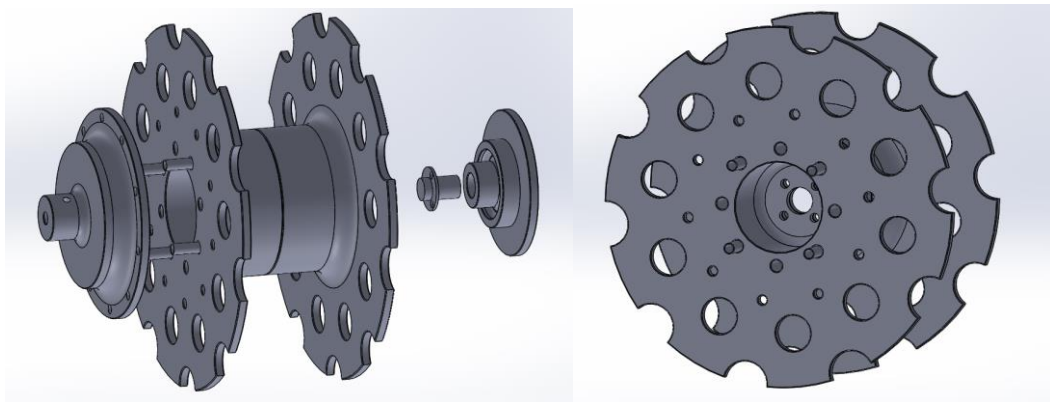
Design af kabel spil

I forbindelse med design af kabel spillet, har der været fokus på at opnå en stiv konstruktion, samt sikre god kraftoverførsel fra motoren. Desuden skulle det forhindres at kablet blev snoet, når spillet hæver og

sænker canistoren. Dette er løst med en slæbering med 8 ledninger - samme antal ledere som i kablet fra Blue Robotics. Slæberingen er rated til 1,5 til 2 A og burde derfor være tilstrækkelig:

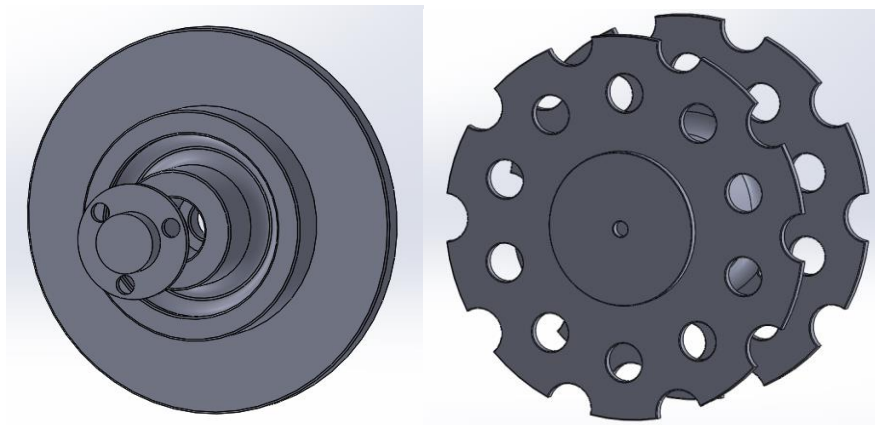
<https://www.fruugo.dk/black-metal-8-ledninger-kapsel-slip-ring-240v-til-vindmolle-vindmolle-vindmollegenerator/p-55787695-113039983>

Første prototype består af fire 3d-printede dele. Motor flange, en prop til slæberingen og to dele bestående af en halv tromle med tilhørende side, til at forhindre kablet i at løbe af. Der er lavet gennemgående huller til aluminiums rør på 6 mm, som afstiver konstruktionen. På motor flangen og den bagerste tromle del, er hullerne ikke gennemgående, for at forhindre rørene i at falde ud. Alle delene boltes sammen med M5 bolte. Motorflangen fastgøres til motorakslen med pinol skruer.



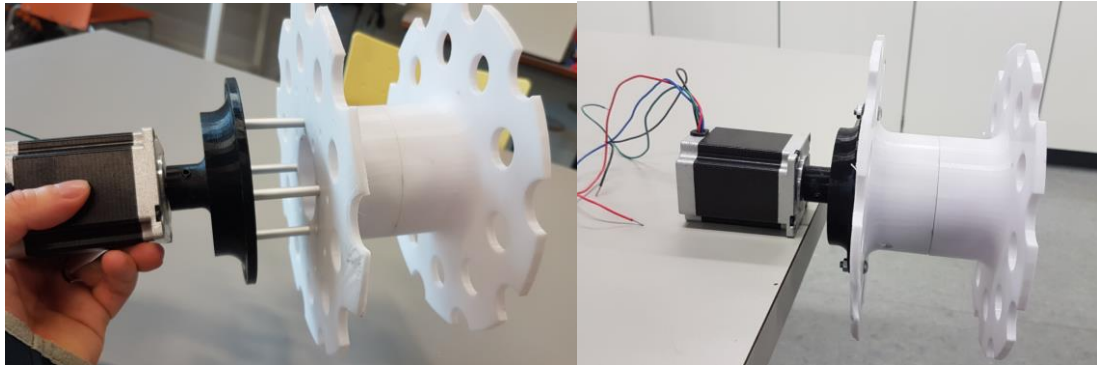
Figur 8: CAD-tegninger af prototype

Slæberingen har 3 monterings huller og skures fast i proppen, der ligeledes skrues fast i tromlen:



Figur 9: CAD-tegninger af prototype

Den printede prototyper ser ud på følgende måde:



Figur 10: Printet prototype

På billedet nedenfor ses det hvor meget 30 meter kabel fylder på tromlen. Da applikationen i Silkeborg, hvor bøjen befinder sig, kun skal bruge 5 meter, er den dog rigeligt stor.



Figur 11: Prototype med 30 meter kabel oprullet

Prototypen er printet i PLA, hvilket kun kan bruges til proof of concept og tør test. Når den skal befinde sig i et vådt miljø, vil det være nødvendigt at printe i PETG. Skolen havde PLA til rådighed, samt at der godt kan opstå problemer med store print i PETG - hvorfor dette valg blev truffet.

Dimensionering af motor

I forbindelse med dimensionering og valg af stepper motoren, skal der tages særligt højde for følgende:

- Holding torque
- Moving torque
- Strømforbrug

Der tages udgangspunkt i worst case scenariet, hvor kabeltromlen er fuld og man derved har det største moment. I forhold til opdrifts budget, sigtes der efter at canistoren med sensorer kommer til at veje max 1 kg under vand. Da kabeltromlen har en ydre diameter på 200 mm, bliver det nødvendige moment:

$$0,1 \text{ m} \cdot 1 \text{ kg} \cdot 9,81 \frac{\text{m}}{\text{s}^2} = 0,981 \text{ Nm}$$

Stepper motoren skal kunne holde canistoren stille i den rigtige position. Dette er givet af holding torque. Samtidig skal den være stærk nok til at kunne rotere kabeltromlen fra stillestående position, givet ved moving torque.

Moving torque kan ved overslags beregning bestemmes som holding torque minus to gange detent torque. Detent torque er det moment som skal til at dreje akslen, når der ikke sendes spænding gennem spolerne.

Da kabelspillet kommer til at sidde på en bøjle med begrænset strømforbrug (12 V batteri med 28/56 ampere timer ved fuld opladning) er det vigtigt at den ikke kræver så stor en strøm. Der er fundet en 2-faset stepper motor, der trækker 1 A ved 3,4 V for hver spole. Denne har holding torque på 1,76 N*m, detent torque på 400 g*cm og moving torque på 1,68 N*m. Motoren er derfor rigeligt stærkt til formålet.

<https://let-elektronik.dk/shop/1290-stepper/601082-bipolar-stepper-motor-with-planet-gear-box-18kgcm/>

I stedet for at købe denne motor hjem, vælges en Nema 23 stepper motor fra skolen lager. Dette er ligeledes nævnt i problemafgørelsen. Motoren er af et ukendt mærke, hvorfor dens specifikke specs er ukendte. Den er dog ligeledes stærk nok til formålet.

Power Budget

I dette afsnit gennemgås power budgettet.

Komponent:	Supply voltage (V)	Peak Current (mA)	Power (Whr)	Standby current (mA)	Standby power (Whr)	Power ved (1 måling Whr)
CO2 sensor	12	55	0,66	?	0	0,495
CH4 sensor	12	690	8,28	0 mA	0	6,21
Pumpe til CH4 sensor (Seabird SBE 5T)	12	640	7,68	0 mA	0	1,92
Temperatur sensor	5	1,4	0,007	0,14 µA	0	0,00525
Tryk sensor	5	1,25	0,00625	0,1 µA	0	0,0046875
Arduino UNO	12	42	0,504	30,8 µA	0	0,378
Arduino MEGA	12	74	0,888	1,5 mA	0,018	0,666
Stepper driver (DQ860MA) + stepper motor (??)	24	600,00	14,4	?	0	3,6
DC-DC boost converter (CN6009)	?	?		?	0	0
DC-DC step down (LM2596)	?	?		?	0	0
Relæ	5	90	0,45	?	0,45	0,3375
LORA SX1276	3,3	28	0,0924	8,4 mA	0,0924	0,0693
RS232 converter (MAX232)	5	5	0,025	?	?	0,01875
Total		2226,65	32,99	9,93	0,56	13,70

Tabel 5: Power Budget

Øverste sensor er markeret med rød, da denne ikke var modtaget i Silkeborg og derfor ikke er med i løsningen. Den er medtaget i power budget, såfremt den bliver tilføjet i fremtiden. Det ses at methan sensoren og pumpen dertil, samt stepper motoren har det største effektforbrug.

Det har ikke været muligt at finde informationer på samtlige komponenter, hvorfor nogle er markeret med " ?".

I den endelige løsning slukkes der for power supply til canistoren, når der ikke foretages målinger. Dette er derfor "Standby Power" er sat til 0 W for komponenterne som ikke befinder sig på bøjlen.

Det ses at man ved kontinuerlig drift, hvor alting er tændt, har et forbrug på ca. 33 Whr. I kolonnen helt til højre, tages der udgangspunkt i forbruget ved én måling. En måling indebærer 30 min warm up af methan sensoren, efterfulgt af ca. 15 min målinger. Dette betyder at alle komponenter er tændt i 45 min, på nær motoren og pumpen som er tændt 15 min. En måling har et forbrug på ca. 14 Whr.

Drift tider

De to bøjler CB150 og CB450 er udstyret med henholdsvis et og to 12-volts batterier.

Givet at der er 28 ampere timer (CB150) og 56 ampere timer (CB450), får man:

	Amp timer (batteri)	Watt timer	Oppe tid ved konstant drift (Timer)	Antal målinger	Aktive døgn (4 målinger)
CB150:	28	336		10,18	24,52
CB450:	56	672		20,4	49,04
					6,13
					12,26

Tabel 6: Drift tider

Det ses at man vil kunne opnå 6 aktive døgn på CB150 og 12 aktive døgn på CB450, givet at man laver 4 målinger af 45 min varighed pr. døgn. Solcellerne på bøjen vil oplade batterierne i mellemtiden, hvilket forlænger drift tiden yderligere. Bøjerne er udstyret med 3 x 13 watt solceller, hvor meget batteriet oplades, er ikke medregnet.

Modstand i ledning

I dette afsnit vil der blive set på tabet igennem kablet ned til canistoren. Der vil blive set på 2 scenarier, et for en løsning med 30 meter kabel og et for en løsning med 5 meter kabel. Modstanden gennem kablet er 0,127 ohm pr. meter. Da der benyttes to lednings par til strømforsyning, halveres modstanden.

Den samlede strøm som skal trækkes til elektronikken i canistoren er:

$$I_{canistor} = 1,4 \text{ A}$$

Modstanden i kablet på 30 meter er:

$$R_{30} = 0,5 \cdot 0,127 \frac{\Omega}{m} \cdot 60 \text{ m} = 3,81 \Omega$$

Dette giver følgende spændings fald:

$$\Delta U_{30} = I_{canistor} \cdot R_{30} = 5,33 \text{ V}$$

Arduino Uno, CH4 og CO2 sensorerne skal forsynes med 12 V, hvorfor spændingen er nødt til at blive øget vha. en DC-DC boost converter. Der benyttes en reguleret boost converter, så output spændingen altid er konstant. Der vælges at bruge en CN6009, da denne var på skolen.

I nedenstående tabel, kan længde, modstand, tab og nødvendig start spænding ses:

Længde på kabel	Modstand i kabel	Spændingsfald i kabel	Nødvendig startspænding (Fra CN6009)
30 meter	3,81 Ω	5,33 V	18 V
5 meter	0,635 Ω	0,89 V	13 V

Tabel 7: Modstand og spændingsfald i kabel

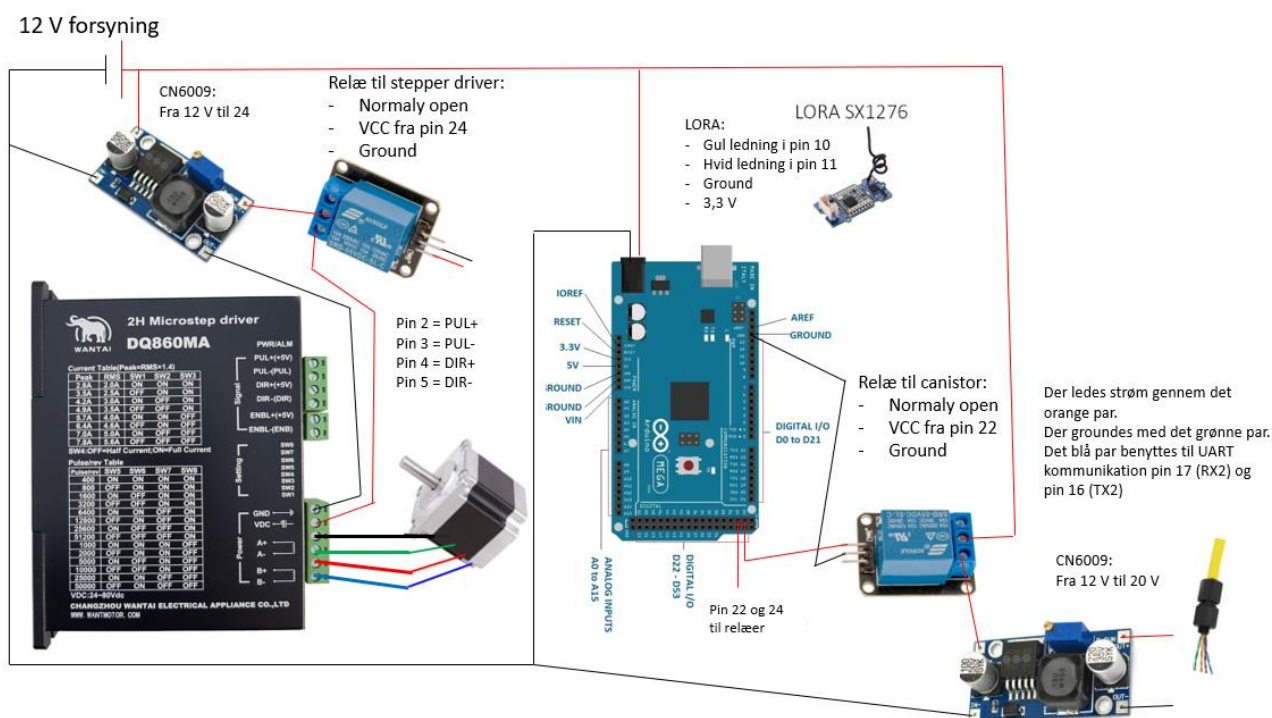
Dette er det teoretiske spændingsfald gennem kablerne. Der vælges derfor at booste spændingen til 20 V i begge tilfælde, for efterfølgende at sænke til 12 V i Canistoren vha. en reguleret step down converter (LM2596).

Wirering på bøjnen

Delen på bøjnen består af følgende elektriske komponenter:

- Arduino Mega
- Lora-modul (SX1276)
- Stepper driver (DQ860MA)
- Stepper motor (ukendt model)
- 5 V relæ
- DC-DC Boost converter (CN6009)

Komponenterne er forbundet i henhold til nedenstående billede:



Figur 12: Oversigt over wirering på bøjnen

Det bemærkes at wirering bærer præg af de udleverede komponenter fra skolen. Det anbefales at udskifte DQ860MA med TB6600, hvis der laves en fremtidig prototype, da denne kan forsynes direkte med 12 V fra batteriet. Derved kan den ene CN6009 fjernes.

Wirering i Canistor

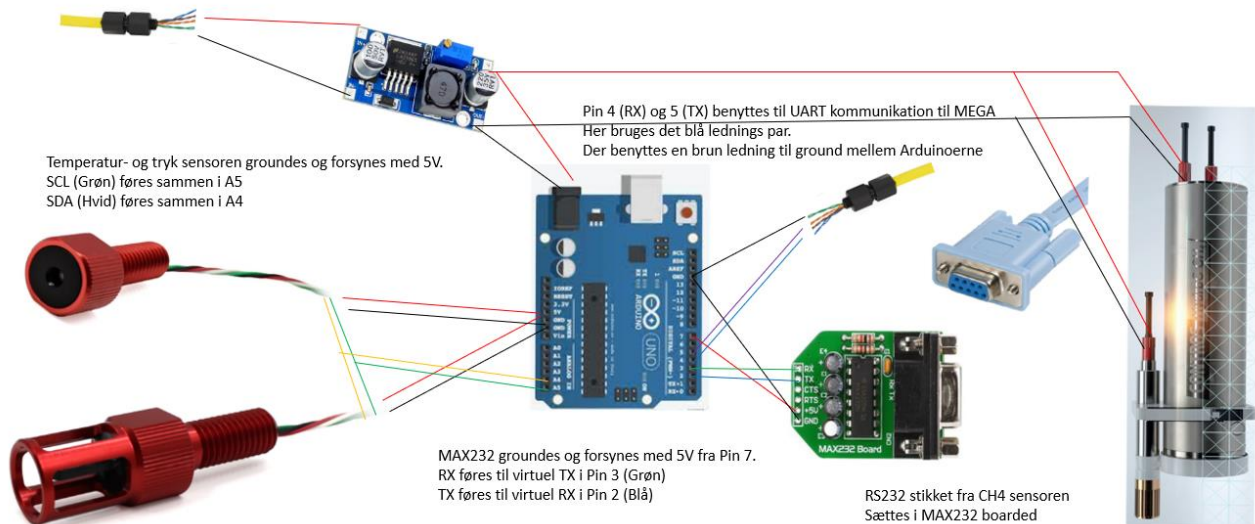
Delen i canistoren består af følgende elektriske komponenter:

- Arduino Uno
- Temperatur sensor - TSYS01 - *Celsius Fast-Response, $\pm 0.1^{\circ}\text{C}$ Temperature Sensor (I2C)*
- Tryk sensor - MS5837 - *Bar30 High-Resolution 300m Depth/Pressure Sensor*
- Methan sensor - CONTROS HydroC dissolved gas sensor
- Pumpe til methan sensor (SBE 5T)
- DC-DC step down converter (LM2596)
- RS232 converter (MAX232)

Komponenterne er forbundet i henhold til nedenstående billede:

Spændingen fra bøjlen sænkes til 12 V
Vha LM2596 DC-DC converteren.
Der ledes strøm gennem det orange par
Mens der groundes med det grønne par

Arduino Uno, CH₄-sensor og pumpe parallelkobles og forsynes med 12 V fra LM2596



Figur 13: Oversigt over wirering i canistor

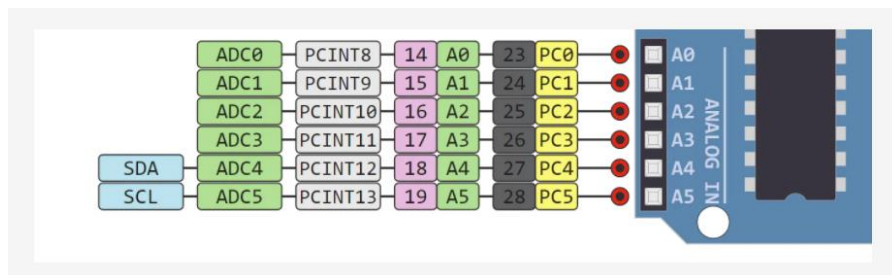
Modul-tests og forsøg

I dette afsnit vil de forskellige modul-tests blive gennemgået.

Aflæsning af temperatur- og tryk sensor

Til denne test benyttes én Arduino UNO, samt sensorerne "TSYS01 - Celsius Fast-Response, $\pm 0.1^{\circ}\text{C}$ Temperature Sensor (I2C)" og "MS5837 - Bar30 High-Resolution 300m Depth/Pressure Sensor" fra Blue Robotics. Formålet med testen er at aflæse flere sensorer med det samme input på Arduino'en.

Sensorerne kan tilgås med I2C eller SPI. Der er valgt at benytte I2C, for derved at kunne have begge sensorer på den samme bus. Disse tilgås vha. analogt input A4 og A5 på Arduino UNO:



Figur 14: A4 og A5 input på Arduino UNO

Sensorerne kobles op i henhold til følgende:

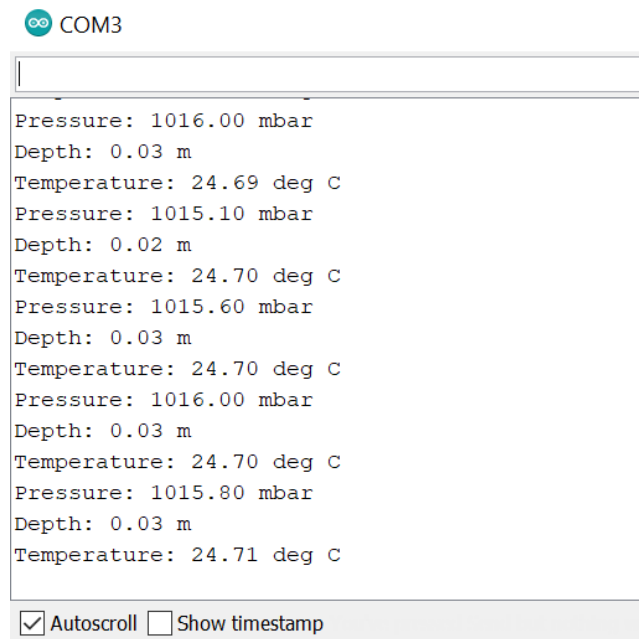


Figur 15: Ledninger fra temperatur- og tryk sensor

Altså med SCL (grøn) i A5 og SDA (Hvid) i A4. Sensorerne groundes og forsynes med 5 V fra UNO'en.

Koden der skrives til Arduino og benyttes til testen, kan ses i appendiks A2:

Temperatur, tryk og dybden, kan da aflæses i serial monitoren. Testen er foretaget ved stuetemperatur uden kalibrering af udstyret. De aflæste værdier fremgår af nedenstående:



Figur 16: Serial monitor fra aflæsning af sensorer

Tryk sensoren er lidt forkert på den, da den egentlig ligger på et bord, men siger den er 3 cm under vandoverfladen. Dette vurderes dog rimeligt og kan skyldtes den befinder sig i luften. Som det ses, er det intet problem at aflæse begge sensorer vha. I2C.

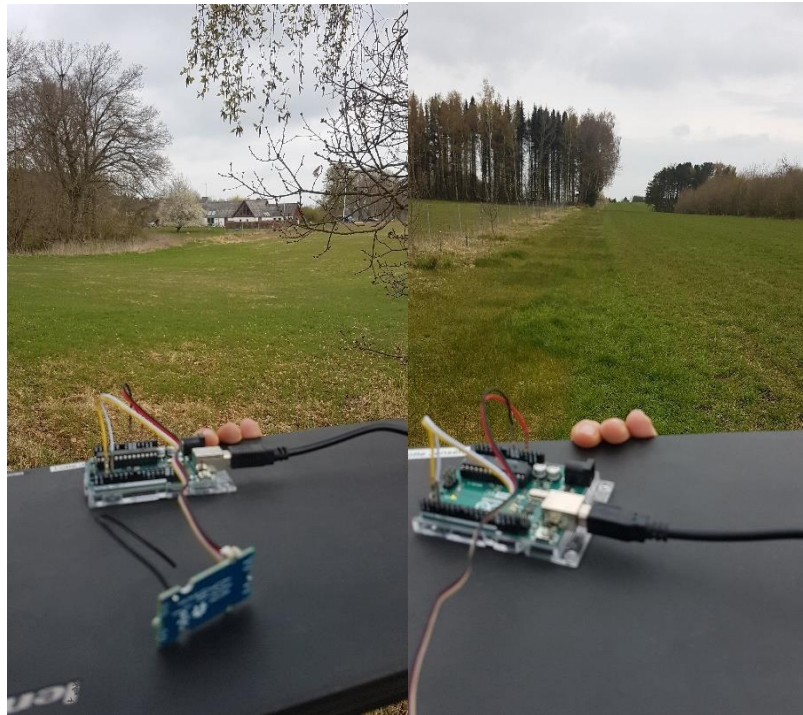
Afstands test med LORA

Til denne test benyttes en Arduino UNO, en Arduino MEGA og to LORA SX1276 moduler. Formålet med testen er at bestemme den maksimale afstand til udveksling af sensordata. I forsøget sendes en tekstbesked frem og tilbage mellem en client og en server.

Arduino MEGA opsættes som client, mens Arduino UNO opsættes som server. Clienten sættes til at sende en besked afsted til serveren, der så modtager og sender en bekræftelse retur. Clienten placeres indenfor i et hus, tilkoblet en computer således at bekræftelsen kan aflæses i serial monitor. Serveren tilkobles strøm fra en anden computer og vil så gradvist blive flyttet længere og længere væk indtil forbindelsen imellem de to LORA-moduler afbrydes.

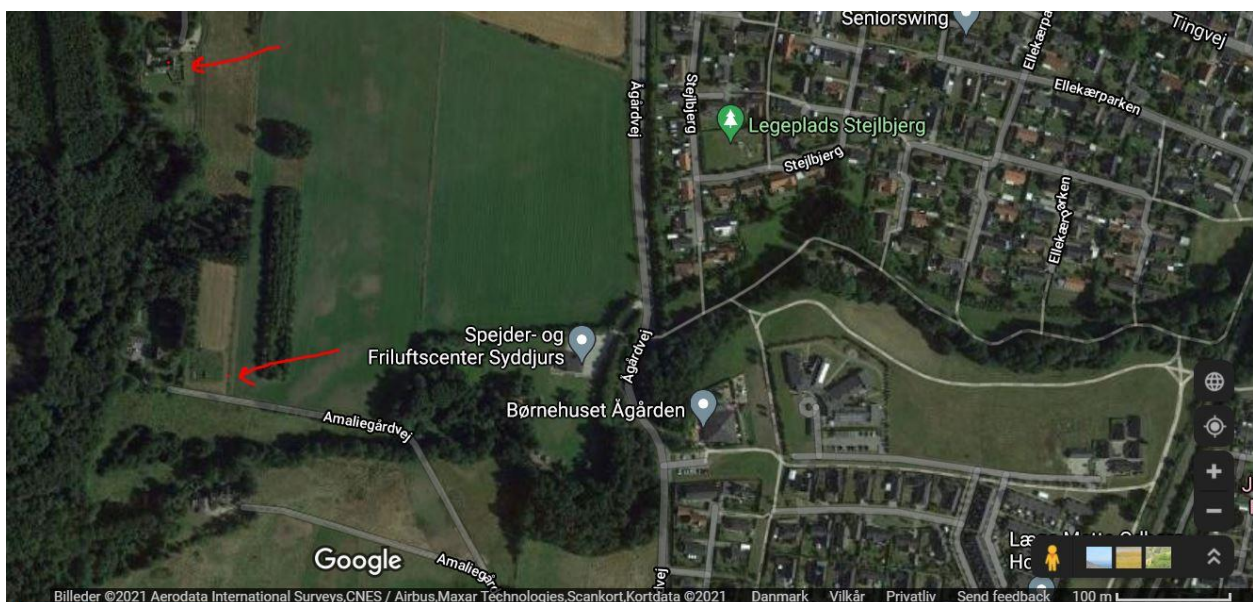
Client koden og Server koden findes i appendiks A3

Det lykkedes at opnå en afstand på ca. 250 meter. Billeder fra forsøget fremgår af nedenstående:



Figur 17: Afstands test med LORA

På billedet til venstre ses afstanden til huset fra skel af. På billedet til højre ses afstanden til skel (høje grantræer i enden af marken på venstre side.



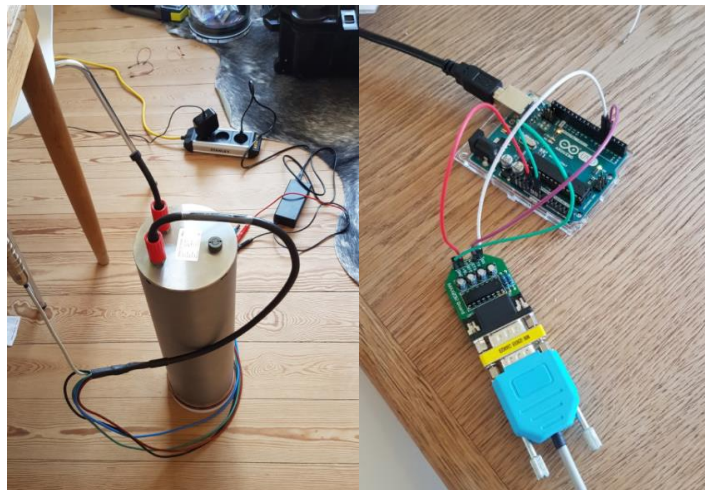
Figur 18: Opnået afstand til kommunikation mellem LORA-moduler (ca. 250 m)

På ovenstående billede ses placeringen af de to LORA moduler, hvor det lykkedes at opnå kommunikation. Afstanden er ca. 250 meter.

Aflæsning af CH₄ sensoren

Sensoren kommer fra det tyske firma 4H Jena Engineering og aflæses almindeligvis vha. en RS232 til USB-adapter, samt medfølgende software. Den udsender 8 data bit og 1 stop bit. Størrelsen på den samlede data-pakke som udsendes er bestemt vha. programmet "HyperTerminal". Data-pakken er bestemt til 200 bits. Sensoren er sat til at udsende data konstant, hvilket betyder at den ikke skal have en start kommando.

Den forsynes med den tilhørende strømforsyning og kobles op til Arduino gennem MAX232 boardet. Boardet er grounded og forsynet med 5V gennem Arduino. Ligeledes er RX forbundet til TX og omvendt imellem de to.



Figur 19: Aflæsning af CH₄-sensor

Sensorens båndbredde indstilles til henholdsvis 9600 og 115200 for at matche Arduino'en. De modtagne data er følgende:

CH4T-1120-001		
Date	28-05-2021	-
Time	13:20:04	-
Weekday	5	-
P_pump	0,442	W
p_TDLAS	1035,29	mbar
p_in	1016,65	mbar
I_total	287,75	mA
U_supply	19,064978	V
external_pump	1	-
Runtime	467910	s
Conc_estimate	7,1	ppm
pCH ₄	7,058	µatm
xCH ₄	7,034	ppm
T_control	33,993	°C
T_gas	32,369	°C
%rH_gas	43,143	%

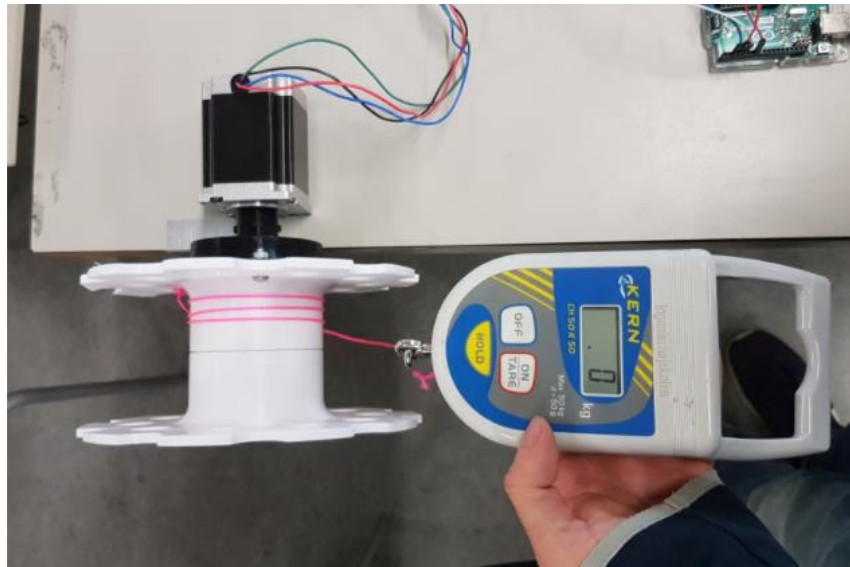
Figur 20: Data fra CH₄-sensor

Der lader til at være problemer med båndbredden (både 9600 og 115200) giver forkerte karakterer ved aflæsningen i serial monitoren på Arduino. Det er ikke lykkedes at aflæse data i samme grad som ved brug

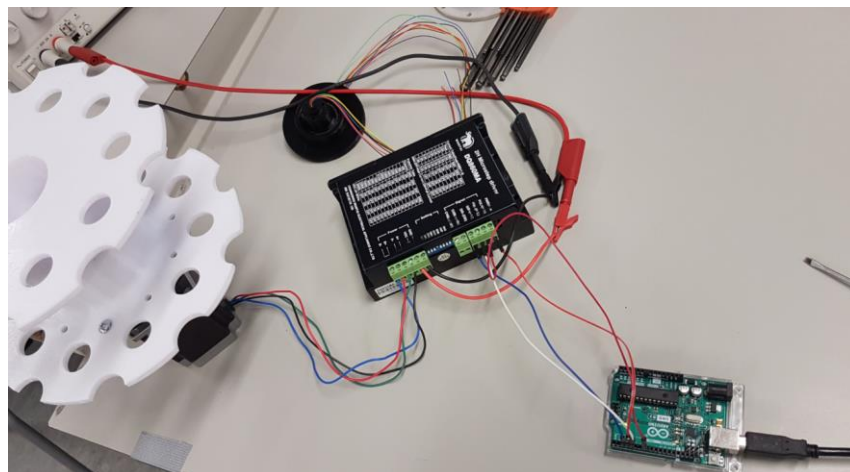
af medfølgende adapter og software. Dette burde kunne løses ved at skrive bedre software til Arduinoen. I appendiks A4 ses forsøget.

Bestemmelse af peak current til steppermotor

I denne test, vil strømmen stepper motoren trækker ved maksimal belastning blive bestemt. Formålet med testen er at fastlægge hvilke motordrivere som kan benyttes. Til testen benyttes driveren DQ860MA, Arduino UNO og en kuffertvægt.



Figur 21: Bestemmelse af peak current vha. kuffert vægt



Figur 22: Overblikks billede fra forsøg med bestemmelse af peak current

Driveren forsynes med 24 V fra strømforsyning. Der skrives et mindre program til Arduino til styring af stepper motoren. Dette program kan ses i appendiks A5. Der bindes snor fast i siden, som derefter kan vikles om tromlen. Stepper motoren sættes i gang, hvorefter den holdes i venstre hånd og der trækkes i kuffertvægten med højre hånd. Der hives i kuffertvægten indtil motoren ikke længere kan rotere tromlen. På strømforsyningen kan strømmen aflæses. Denne blev aflæst til ca. 550 mA, ved en belastning på ca. 2,95 kg. Givet at tromlen har en radius på 40 mm, får man følgende moment:

$$2,95 \text{ kg} \cdot g \cdot 40 \text{ mm} = 1,157 \text{ N} \cdot \text{m}$$

Ved en fuldt oprullet kabeltromle med 30 meter kabel, vil man da kunne løfte ca. 1,18 kg.

$$\frac{1,157 \text{ N} \cdot \text{m}}{g \cdot 100 \text{ mm}} = 1,18 \text{ kg}$$

Til opsætningen i Silkeborg, hvor der bare ønskes 5 meter kabel, vil momentet være betydeligt mindre.

Da der kun trækkes små 600 mA, kan der bruges mindre drivere end DQ680MA. Dette kunne eksempelvis være "TB6600". Dette ville være at foretrække, da den kan forsynes med 12 V, som kan tilgås på bøjlen.

Datakommunikation mellem UNO og MEGA

Der er valgt at benytte UART som kommunikations protokol mellem Arduino'erne. Der oprettes virtuelle RX- og TX-kanaler på UNO'en vha. kommandoen "*SoftwareSerial*". På denne måde undgås fejl ved upload til boarded. De almindelige RX og TX (pin 0 og 1) er nemlig reserveret til kommunikationen gennem USB, når denne er forbundet til computeren.

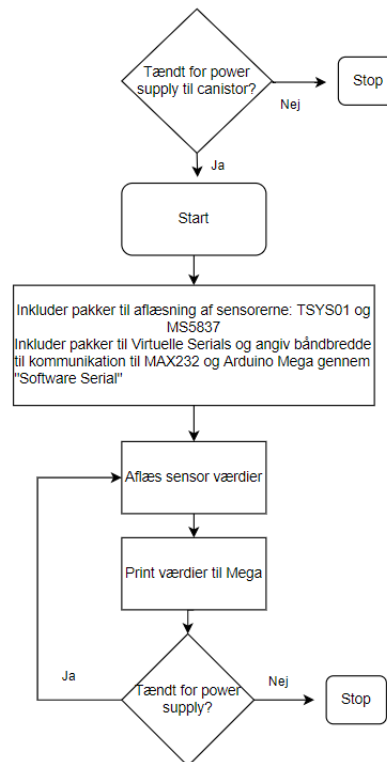
Når UNO'en har foretaget en måling på sensorerne i canistoren, skriver den resultatet videre til MEGA'en. Det er lykkedes at sende data gennem 30 meter kabel, hvorfor dette valg findes egnet. Kommunikationen er ikke synkroniseret ved UART, men da man kun har én vejs kommunikation (fra UNO til MEGA), tænkes dette ikke at være et problem. På UNO benyttes pin 4 (RX) og 5 (TX), mens der for MEGA benyttes 17 (RX) og 16 (TX). Koden til dette kan ses i appendiks A6.

Beskrivelse af koden

I dette afsnit vil koden på de 2 Arduino'er blive beskrevet

Flow diagram for Uno

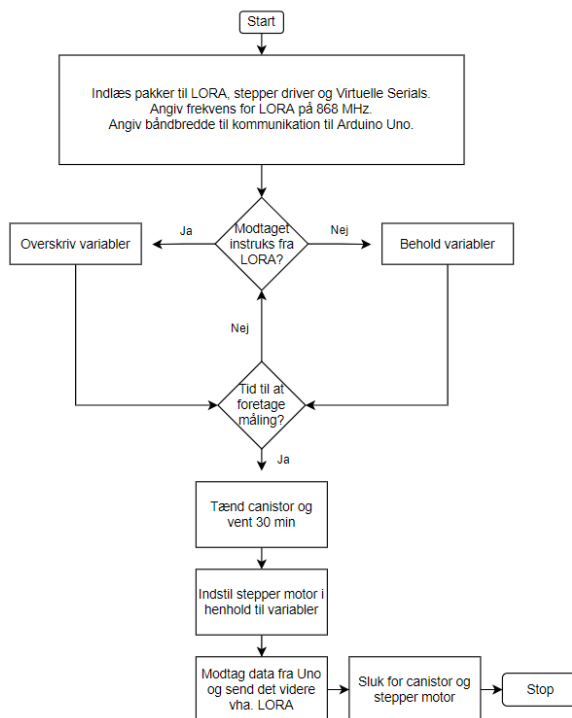
Dennes opgave er at foretage målinger på sensorerne og sende resultatet videre til Arduino Mega via UART kommunikation. Flow diagrammet ser ud på følgende måde:



Figur 23: Flow diagram for kode på Arduino UNO

Flow diagram for Mega

Dennes opgave er at modtage instrukser til indstilling af kabelspillet trådløst over LORA. Samtidig skal den modtage data fra Uno og videresende det retur over LORA. Flow diagrammet ser ud på følgende måde:



Figur 24: Flow diagram for Arduino MEGA

Trådløs indstilling af stepper motor

Ideen er at der sendes en 3 cifret kode som modtages af LORA-modulet. Koden skal da gemmes som en variabel på Arduino Mega'en. Ud fra den modtagne kode, skal de forskellige 3 cifre sammenlignes med prædefinerede indstillinger. Hvor hvert cifre svarer til en indstilling ala. nedenstående:

Antal målinger pr. dag	Afstand mellem stop	Pause pr. stop
0	0 = Passiv	0 = Passiv
1	A = 10 cm	A = 30 sek.
2	B = 20 cm	B = 1 min
3	C = 30 cm	C = 1,5 min
4	D = 40 cm	D = 2 min
5	E = 50 cm	E = 5 min

Figur 25: Indstillings valgmuligheder

Modtager man eksempelvis koden "4EE", skal bøjnen foretage 4 målinger den pågældende dag. Hvor hver måling indebærer et stop på 5 min for hver 50 cm. Denne variabel med indstilling af målinger vil være gældende indtil, at man sender en ny indstilling. MEGA vil sende den samme kode retur som bekræftelse på den valgte indstilling.

Skal bøjnen sættes i passiv mode, pga. årstiden og det begrænsede behov for målinger, kan koden "000" sendes. Sådan som koden fungerer nu, vil Arduino'en starte i passiv mode med slukket motor og

strømforsyning til canistoren. Først når den modtager en reel kommando, vil den tænde relæet til canistor og stepper driver og foretage målinger. Den nuværende prototype kan kun holde kabel spillet stillestående, så længe at der er strøm til stepper motorens spoler. Dette betyder at kabel spillet naturligt vil rulle ud, når den sidste måling er foretaget. Konsekvensen af dette er at canistoren med sensorerne synker mod bunden indtil at alt kablet er rullet ud. Dette var tænkt som en energi effektiv løsning, da man da sparede den strøm som motoren ville bruge på at rulle spillet ud igen. I koden er der defineret en start dybde. Hver gang motoren hæver canistoren et stop, opdateres dybden. Når "overfladen" nås, her defineret ved 1 meter, sender den en bekræftelse på at målingen er færdig hvorefter at relæerne slukkes. På denne måde vil den synke til bunds og derved være klar til en ny måling, når den ønskede tid er gået.

Koden til trådløs indstilling af stepper motorerne kan ses i appendiks A7.

Økonomi og opskalering

Som tidligere nævnt er løsningen med en sensor pakke under vandet skaler- og justerbar. Udvalget og mængden af sensorer kan skiftes efter behov. Dette betyder at applikationen rammer et bredere marked og altså ikke kun er forbeholdt CH4-, CO2- og temperatur målinger.

Nedenfor ses en bom liste med komponenter til en lignende løsning. BEMÆRK at der ikke er medregnet prisen til CO2 og CH4 sensoren. Desuden gøres der opmærksom på at ikke alle komponenterne er ens med den beskrevne løsning/prototype. Skolen lå eksempelvis inde med stepper motor som så blev brugt i stedet for at købe nye. I appendiks A8 kan linket til forhandlere ses.

nr	Navn	Antal	Pris	Fragt	Shipping land	Input V	Leverings tid
1	Deph/Pressure sensor (Blue Robotics)	1	451	500	Tyskland	2,5/5,5 V	7 dage
2	Arduino MEGA	1	247	80	DK	7/12 V	2 dage
3	LORA modul SX1276	2	330		DK	3,5/5 V	2 dage
4	Bipolar Stepper motor, 2 faset	1	383	80	DK	3,4 V pr. fase	3-5 dage
5	Stepper motor driver TB660	1	149	80	DK		3-5 dage
6	Temperature sensor (Blue Robotics)	1	376		Tyskland	3,3/5,5 V	7 dage
7	ROV Teather (Blue Robotics) - 6 m	1	183		Tyskland		7 dage
8	ROV Enclosure (Blue Robotics)	1	2712		Tyskland		7 dage
9	DC-DC Step up boost converter	1	99	80	DK		3-5 dage
10	DC-DC Step down converter	1	99		DK		3-5 dage
11	Slæbering	1	119	42,5	China		14 dage
12	MAX 232 (RS-232 converter)	1	68		DK	5 V	2 dage
13	Arduino Uno	1	169		DK	7/12 V	2 dage
14	5V relæ	2	50		DK	5 V	3-5 dage

tot 6297,5

Figur 26: BOM-liste til indkøb af en lignende løsning

Det ses at de indkøbte dele koster ca. 6.300 kr. Hertil skal tillægges materialeomkostninger til motor beslag og 3D-printet tromle.

Diskussion

Den udviklede prototype har vist at den godt ville kunne løse behovene for forskerne i Silkeborg med en sø på 5 meter. Der er udført en række test som understøtter de valgte komponenter og prototypen som *proff of concept*. Før løsningen, kan implementeres på bøjen, skal der skrives bedre software. Det er ikke lykkedes at få fornuftige data ud af CH4 sensoren uden brug af medfølgende software. Dette problem skal ligeledes løses.

Konklusion

Der er designet og testet en prototype til profilering af tryk, temperatur og CH4 målinger i de forskellige dybde-niveauer af søer. Den designede prototype fungerer som *proff of concept*. Prototypen kan opdeles i 2 dele, hvor man har en del på bøjen bestående af en Arduino MEGA, stepper driver, stepper motor og et LORA-modul. Den anden del befinder sig under vand i en canistor, hvor man har en Arduino UNO samt temperatur-, tryk- og CH4 sensor.

Sensorerne sænkes ned i søen vha. et kabelspil som er monteret på CB150 bøjens overside. Kablet benyttes både til datakommunikation og strømforsyning. Prototypen til kabelspillet er printet i PLA og egner sig kun til proff of concept. Der er fundet en stepper motor som kan yde det nødvendige moment på minimum 1 N*m til at hæve og sænke canistoren med tilhørende sensorer under vand. I løsningen er der brugt en stepper motor fra skolens lager.

Al elektronik under vandet befinder sig inden i en købt canistor, som er tiltænkt undervandsbrug. Ledninger kan trækkes gennem kabelgennemføringer som skrues fast i canistorens låg. Elektronikken på bøjen befinder sig i et vandtæt rum i midten af bøjen.

Det udarbejdede power-budget viser, at der kan opnås drift tider på 6 dage for CB150 og 12 dage for CB450, givet at man foretager 4 målinger pr. dag af 45 minutters varighed.

Sensorerne kan sænkes ned til 5 meters dybde.

Såfremt at forholdene er ens med den udførte test mellem de to LORA-moduler, kan datakommunikation og indstilling af robotten foretages op til 250 meter væk.

Der er skrevet Arduino software som trådløst kan regulere stepper motoren vha. LORA. Dette gælder antallet af stop, afstanden i mellem dem og tiden et stop varer. Indstillingen af stepper motoren foregår ved at sende en 3 cifret kode til Arduino MEGA, som da sammenlignes med prædefinerede indstillinger.

Perspektivering

Fremadrettet kan der ses på en dybde justering, hvor man indstiller kabelspillet i forhold til tryk målingen. Den indstillede dybde vil da komme tættere på det ønskede, da afstanden som canistoren hæves og sænkes med, vil variere i takt med mængden af kabel på tromlen. Den nuværende prototype ruller spillet ud, når der slukkes for strømmen. Dette betyder at kabellængden skal tilpasses alt efter hvor man placerer bøjen. I forhold til en mere general løsning, som kan bruges flere steder, kan der laves en bremsemekanisme eller monteres gearkasse på motoren, således at den kan holde sensorpakken i slutpositionen, når der slukkes for strømmen. Softwaren er tilpasset til at der startes i den samme dybde, dette ville ligeledes skulle ændres.

Projektet har vækket interessen for elektronik og Arduino programmering, samt vist at projektarbejde er sjovest i grupper af flere personer.

Appendiks

A1 - Subsystemer og point givning

Her ses løsningsideerne til de forskellige subsystemer

Op/nedsænkningssystem til sensorer (Subsystem 1)

- Kravle på kabel
- Kabel tromle system
 - o Placeret på bøjen
 - o Placeret på sensor pakken

Valg af motor til op/nedsæknings system (Subsystem 2)

- DC-motor
- Servo motor
- Stepper motor

Motor mount (Dette subsystem fjernes)

- 3D printet
- CNC bearbejdet
- Laser skåret

Valg af sensor til at registrere når man er på bunden (Dette subsystem fjernes)

- Tryksensor
 - o Piezo
 - o Mekanisk knap
 - o Tryk måling
- Afstands sensor
 - o IR
 - o Sonar
- Vægt måling
 - o Ophæng i fjeder (kuffert vægt)

System til stabilisering af sensor pakke under op/nedsænkning (Dette subsystem fjernes)

- Ophæng i 3 kabler
- Vægt under sensor pakke
- Trisse system

System til opspoling af data- og strømkabler (Subsystem 3)

- Kabel tromle system
- Automatisk oprul som set i en støvsuger

Case til sensorer (Subsystem 4)

- 3D print
- PVC rør
- Canistor

Montage på bøjerne CB150 og CB450 (Dette subsystem fjernes)

- Fastgør motor på beslag på toppen af bøjen

Her ses pointdelingen for de forskellige subsystemer:

Subsystem 1 (Op-/ned sænkningssystem af sensorer)

			Kompleksitet	Vandtæt	Styring med Arduino	SUM
	Ideer	Vægtning	3	5	2	
1.1	Kravle på kabel		1	1	1	10
1.2	Kabel tromle (bøje)		5	3	3	36
1.3	Kabel tromle (sensorkasse)		3	1	3	20

Det giver bedst mening at have kabel tromlen på bøjen.

Subsystem 2 (Motorvalg)

			Position	Moment	Styring med Arduino	SUM
	Ideer	Vægtning	4	4	2	
2.1	Steppmotor		5	5		40
2.2	DC		x		x	
2.3	Servo		x		x	

Subsystem 3 (Op-/ned sænkningssystem af data/strømkabler)

			Kompleksitet	Vandtæt	Styring med Arduino	SUM
	Ideer	Vægtning	3	5	2	
3.1	Autooprul (støvsuger)	x		0	0	x
3.2	Kabel tromle (bøje)		5	3	3	36
3.3	Kabel tromle (sensorkasse)		3	1	3	20

Det kan ikke lade sig gøre med auto oprul i de længder (30 m). 5 meter ville nok godt kunne lade sig gøre, men det er komplekst. Ift at gøre det vandtæt, giver det bedre mening oppe på bøjen.

Subsystem 4 (Casing til elektronik under vandet)

			Kompleksitet	Vandtæt	Funktion/Fleksibilitet	SUM
	Ideer	Vægtning	2	5	3	
4.1	3D printet		1	1	2	12
4.2	PVC rør		2	2	3	22
4.3	Købt Canister		5	5	5	50

Det er meget mere fleksibelt med en købt canistor, da man her kan ændre på sensorer efter behov. Ved de andre skal man lave en ny hver gang. Man kan ligeledes lave det helt vandtæt, hvilket ligeledes giver en del opdrift som muliggør en mindre motor.

A2 - Aflæsning af temperatur og tryk sensor (I2C)

Kode til aflæsning af temperatur og tryksensor

```
// SCL (grøn) i A5 og SDA (Hvid) i A4. Sensorerne groundes og forsynes med 5 V fra UNO'en
#include <Wire.h>
#include "MS5837.h" // Henter bibliotek til Tryk sensor
#include "TSYS01.h" //Henter bibliotek til Temp sensor
TSYS01 sensorTemp; //Definere
MS5837 sensorTryk; //Deninere
void setup() {
  Serial.begin(9600);
  Serial.println("Starting");
  Wire.begin();
  sensorTemp.init(); // Starter temperatur sensoren
  // Starter tryk sensor og tester om denne starter korrekt op
  while (!sensorTryk.init()) {
    Serial.println("Init failed!");
    Serial.println("Are SDA/SCL connected correctly?");
    Serial.println("Blue Robotics Bar30: White=SDA, Green=SCL");
    Serial.println("\n\n");
    delay(5000);
  }
  sensorTryk.setModel(MS5837::MS5837_30BA);
  sensorTryk.setFluidDensity(997); // kg/m^3 (freshwater, 1029 for seawater)
}
void loop() {
  sensorTryk.read(); // Aflæser tryk sensoren
  Serial.print("Pressure: ");
  Serial.print(sensorTryk.pressure());
  Serial.println(" mbar");
  Serial.print("Depth: ");
  Serial.print(sensorTryk.depth());
  Serial.println(" m");
  sensorTemp.read(); //Aflæser temperatur sensoren
  Serial.print("Temperature: ");
  Serial.print(sensorTemp.temperature());
  Serial.println(" deg C");
  delay(1000); // Dette er bare for test. Behøver ik ske så ofte
}
```

A3 - Klient og server kode til trådløs kommunikation med LORA

Client kode til Arduino MEGA

//Lora modulet groundes og forsynes med 3,3 V. Gul sættes i pin 10, mens hvid ledning sættes i Pin 11.

```
#include <SoftwareSerial.h>
#include <RH_RF95.h>
//Dette setup er lavet på Arduino MEGA'en
// Singleton instance of the radio driver
SoftwareSerial ss(10, 11); // (Gul i Pin 10 og hvid i Pin 11)
RH_RF95 rf95(ss);
void setup()
{
  pinMode(7,OUTPUT);
  Serial.begin(9600);
  Serial.println("RF95 client test.");
  if (!rf95.init())
  {
    Serial.println("init failed");
    while(1);
  }
  rf95.setFrequency(868.0);
}
void loop()
{
  digitalWrite(7, HIGH);
  Serial.println("Sending to rf95_server");
  // Send a message to rf95_server
  uint8_t data[] = "Hello World!";
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
  // Now wait for a reply
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if(rf95.waitAvailableTimeout(3000))
  {
    // Should be a reply message for us now
    if(rf95.recv(buf, &len))
    {
      Serial.print("got reply: ");
      Serial.println((char*)buf);
    }
  }
}
```

```
    }
    else
    {
        Serial.println("recv failed");
    }
}
else
{
    Serial.println("No reply, is rf95_server running?");
}
delay(1000);
}
```

Server kode til Arduino UNO

// Lora modulet groundes og forsynes med 3,3 V. Gul sættes i pin 2, mens hvid ledning sættes i Pin 3.

```
#include <SoftwareSerial.h>
#include <RH_RF95.h>
//Setup på Arduino UNO
// Singleton instance of the radio driver
SoftwareSerial ss(2, 3 ); // RX og TX (Gul i Pin 2 og hvid i Pin 3)
RH_RF95 rf95(ss);
int led = 7; //Bruges udelukkende som en respons test
void setup()
{
    Serial.begin(9600);
    Serial.println("RF95 server test.");
    pinMode(led, OUTPUT);
    if(!rf95.init())
    {
        Serial.println("init failed");
        while(1);
    }
    rf95.setFrequency(868.0);
}
void loop()
{
    if(rf95.available())

    {
        // Should be a message for us now
        uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        if(rf95.recv(buf, &len))
```

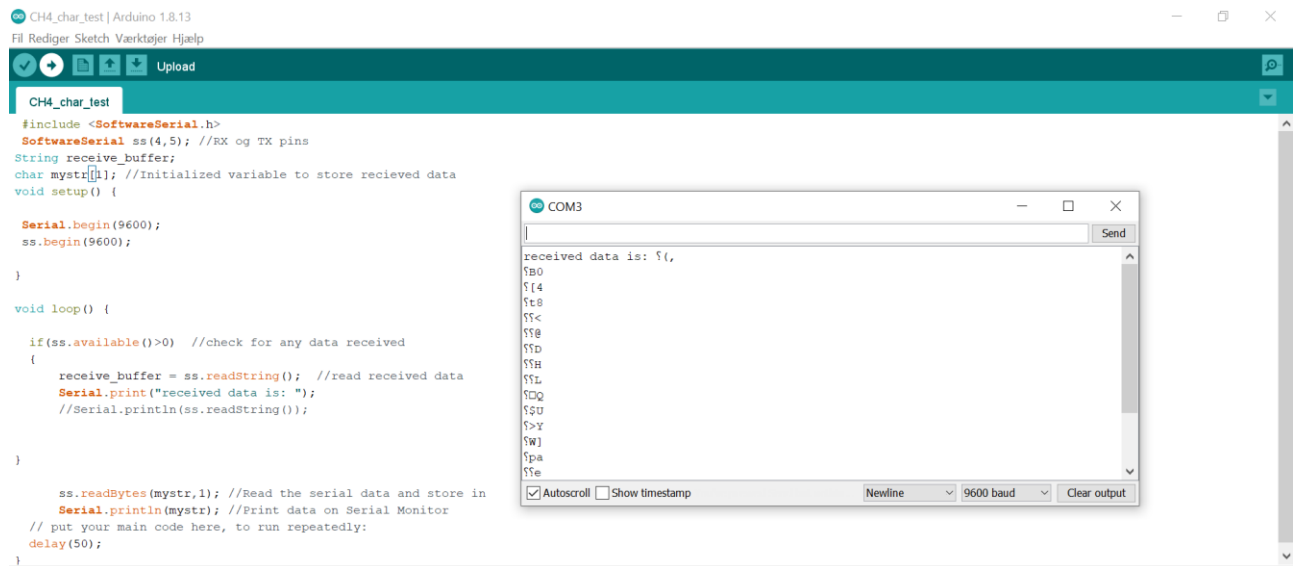
```

{
  digitalWrite(led, HIGH);
  Serial.print("got request: ");
  Serial.println((char*)buf);
  // Send a reply
  uint8_t data[] = "And hello back to you";
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
  Serial.println("Sent a reply");
  digitalWrite(led, LOW);
}
else
{
  Serial.println("recv failed");
}
}
}

```

A4 - Aflæsning af CH4 sensor med MAX232

Forsøg på aflæsning af CH4 sensor vha. Arduino.



A5 - Bestemmelse af peak current for stepper motor

Nedenfor ses den skrevne kode:

```

/*
* Mellem stepper driver og motor har man:
* Sort i A+
* Grøn i A-

```

```
* Rød i B+
* Blå i B-
*/

void setup() {

  // put your setup code here, to run once:

  pinMode(7, OUTPUT); // Denne styrer "PUL -" - Hvis man gerne vil holde motoren i den samme position,
  kan denne sættes til low

  pinMode(2, OUTPUT); //Denne styrer "DIR-"

  //Pins med 5V

  pinMode(4, OUTPUT); //Denne styrer "Pul+"

  pinMode(3, OUTPUT); // Denne styrer "DIR+"

}

void loop() {

  digitalWrite(4, HIGH); //Sørger for 5V

  digitalWrite(3, HIGH); //Sørger for 5V

  // put your main code here, to run repeatedly:

  digitalWrite(2, HIGH); //Sætter ens retning for at teste. retningen vil være givet af hvordan kablerne er
  trukket til "DIR + og DIR -"

  //Kører med ens hastighed givet ved delayet

  digitalWrite(7,HIGH);

  delay(25);

  digitalWrite(7,LOW);

  delay(25);

}

//Noter fra forsøg

//Half current (switch 4 off på stepper driver) - Man fik samme resultat med full current (switch 4 on)

//Ved testen kan hives 2,95 kg før motoren stopper. Dette er helt inde ved tromlen og ved delay på 25: 2,95
kg * g * 40 mm = 1,157 N*m (Moving torque)
```

//Strømforsyning trækker 0,55 A ved denne belastning

//Detent torque er ca. $10 \text{ g} * \text{g} * 40 \text{ mm} = 0,004 \text{ N*m}$ - bestemt ved at dreje tromlen med kuffert vægten

A6 - UART kommunikation mellem UNO og MEGA

Her ses koden til UART-kommunikation mellem de to Arduinoer til aflæsning af sensorværdier:

Opsætning på UNO

```
/*  
 * SCL (grøn) i A5 og SDA (Hvid) i A4. Sensorerne groundes og forsynes med 5 V fra UNO'en  
 */  
  
#include <Wire.h>  
  
#include "TSYS01.h"  
  
TSYS01 sensor;  
  
#include <SoftwareSerial.h>  
  
SoftwareSerial ss(4, 5); //RX og TX  
  
char mystr[5] = "12345"; //String data  
  
void setup() {  
    // Begin the Serial at 9600 Baud  
    Serial.begin(9600);  
    ss.begin(9600);  
  
    // Igen kode fra temp:  
    Wire.begin();  
    sensor.init();  
}  
  
void loop() {  
    //Serial.write(mystr,5); //Write the serial data  
    delay(5000);  
  
    //ss.write(mystr,5); //Write the serial data  
    sensor.read();  
}
```

```
ss.print("\n"); //Starter med at sende ny linje

ss.print("Temperature: ");

Serial.print("Temperature: ");

ss.print(sensor.temperature());

Serial.print(sensor.temperature());


ss.println(" deg C");

Serial.println(" deg C");

}
```

Opsætning på MEGA

```
/*Modtager setup på MEGA

* Bruger pin 17 (RX) og 16 (TX)

* Der skal laves en string for hver sensor. Disse kan så sendes over som én samlet "liste" af strings

*/

#include <string.h>

#include <SoftwareSerial.h>

SoftwareSerial ss(17, 16); //RX og TX

char mystr[30]; //Initialized variable to store recieved data

void setup() {

    // Begin the Serial at 9600 Baud

    Serial.begin(9600);

    ss.begin(9600); //Ligesom ovenstående, bare for pin 17 og 16

}

void loop() {

    // Serial.readBytes(mystr,5); //Read the serial data and store in var

    //Serial.println(mystr); //Print data on Serial Monitor
```



```
ss.readBytes(mystr,30); //Read the serial data and store in  
Serial.println(mystr); //Print data on Serial Monitor  
delay(1000);
```

A7 - Trådløs indstilling af stepper motor

Her ses koden til trådløs indstilling af stepper motorerne

Arduino Uno med LORA-modul (Afsender af instruks)

```
//Afsender LORA modul  
  
#include <SoftwareSerial.h>  
  
#include <RH_RF95.h>  
  
  
//Setup på Arduino UNO  
  
// Singleton instance of the radio driver  
  
SoftwareSerial ss(2, 3 ); // RX og TX (Gul i Pin 2 og hvid i Pin 3)  
  
RH_RF95 rf95(ss);  
  
int led = 7;  
  
void setup()  
{  
  pinMode(7,OUTPUT);  
  
  Serial.begin(9600);  
  
  Serial.println("RF95 client test.");  
  
  if (!rf95.init())  
  {  
    Serial.println("init failed");  
  
    while(1);  
  }  
  
  rf95.setFrequency(868.0);  
}
```

```
void loop()
{
    digitalWrite(7, HIGH);

    Serial.println("Sending to Arduino MEGA");

    // Send a message to rf95_server

    //uint8_t data[] = '\n';

    uint8_t data[] = "4AA"; //Her sendes instruksen til Mega som indstiller stepper motor

    rf95.send(data, sizeof(data));


    rf95.waitPacketSent();

    // Now wait for a reply

    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];

    uint8_t len = sizeof(buf);

    if(rf95.waitAvailableTimeout(3000))
    {
        // Should be a reply message for us now

        if(rf95.recv(buf, &len))
        {
            Serial.print("got reply: ");

            Serial.println((char*)buf);

        }

        else

        {

            Serial.println("recv failed");

        }

    }
}
```

```
else  
  
{  
  
    Serial.println("No reply, is Arduino MEGA running?");  
  
}  
  
delay(1000);  
  
}
```

Arduino Mega med LORA-modul og stepper driver + motor (Modtager)

Det bemærkes at værdierne benyttet skal tilpasses de enkelte cases. Disse ses udelukkende som proff of concept.

//Modtager LORA modul til styring af stepper

```
#include <SoftwareSerial.h> // Virtuelle RX og TX
```

```
#include <RH_RF95.h> // LORA modul
```

```
#include <Stepper.h> // Stepper
```

```
long day = 86400000; // 86400000 milliseconds in a day
```

```
long hour = 3600000; // 3600000 milliseconds in an hour
```

```
long minute = 60000; // 60000 milliseconds in a minute
```

```
long second = 1000; // 1000 milliseconds in a secon
```

```
//Dette setup er lavet på Arduino MEGA'en
```

```
SoftwareSerial ss(10, 11); // (Gul i Pin 10 og hvid i Pin 11)
```

```
RH_RF95 rf95(ss);
```

```
/*På driveren DQ860MA (SW1=1, SW2=1, SW3=1, SW4=0, SW5=1, SW6=1, SW7=1, SW8=1), hvor 1=on og 0=off
```

```
* Pin 2 = PUL+, Pin 3 = PUL-, Pin 4 = DIR+, Pin 5 = DIR-
```

```
*/
```

```
const float stepsPerRevolution = 1600; // Steps pr. revolution

Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5); //Pins til styring af DQ860MA

float dybde = 5; //Defineret start dybde (5 m)

float omkreds = 0.25; //Denne værdi er brugt til testen

// Givet at man ruller fra start til slut, må den gennemsnitslige diameter på tromlen kunne bruges (0.44 m
ved 30 m kabel) og (0.25 m med snor på tromle)

//Pins til relæ styring af stepper og canistor strømforsyning

const int releC = 22; // Relæ til canistor

const int releM = 24; // Relæ til stepper motor

void setup()

{

    // Definere pins til relæ

    pinMode(releC, OUTPUT);

    pinMode(releM, OUTPUT);

    // Sætter hastigheden til 60 rpm

    myStepper.setSpeed(60);

    Serial.begin(9600);

    Serial.println("RF95 client test.");

    if (!rf95.init())

    {

        Serial.println("init failed");

        while(1);

    }

    rf95.setFrequency(868.0);

}

void loop()

{
```

```
Serial.println("Sending to rf95_server");

// Send a message to rf95_server

uint8_t data[] = "Arduino MEGA is connected";

rf95.send(data, sizeof(data));

rf95.waitPacketSent();

// Now wait for a reply

uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];

uint8_t len = sizeof(buf);


if(rf95.waitForAvailable(3000))
{
    // Should be a reply message for us now

    if(rf95.recv(buf, &len))
    {
        Serial.print("got reply: ");

        Serial.println((char*)buf);

        // Gemmer input som String

        String tal = ((char*)buf);

        Serial.println(tal); //Denne printes som bekræftelse

        // Sammenligner strings og vælger en setting ud fra dette

        if (tal == "4EE")

        { Serial.println("Command 4EE");

            //Sætter motoren i gang i henhold til

            digitalWrite(releC, HIGH);

            //delay(0.5*hour); // 30 minutters opvarmning til CH4 sensoren

            digitalWrite(releM, HIGH); //Åbner relæ til stepper
```

```
myStepper.step(1*stepsPerRevolution);

dybde = dybde - 1*omkreds;

Serial.println(dybde); //Test
}

if (tal == "4BB")

{ Serial.println("Command 4BB");

//Sætter motoren i gang i henhold til

digitalWrite(releC, HIGH);

//delay(0.5*hour); // 30 minutters opvarmning til CH4 sensoren

digitalWrite(releM, HIGH); //Åbner relæ til stepper

myStepper.step(0.8*stepsPerRevolution);

dybde = dybde - 0.8*omkreds;

Serial.println(dybde); //Test
}

if (tal == "4AA")

{ Serial.println("Command 4AA");

//Sætter motoren i gang i henhold til

digitalWrite(releC, HIGH);

//delay(0.5*hour); // 30 minutters opvarmning til CH4 sensoren

digitalWrite(releM, HIGH); //Åbner relæ til stepper

myStepper.step(0.4*stepsPerRevolution);

dybde = dybde - 0.4*omkreds;

Serial.println(dybde); //Test
}

if (tal == "000") { Serial.println("Pausing motor");

//Sætter motoren i gang i henhold til
```

```
        myStepper.step(0*stepsPerRevolution);
    }

    else {Serial.println("Unknown command"); }

//Send koden retur, for bekræftelse på indstilling tal.length()

    uint8_t retur[tal.length()];

    tal.toCharArray(retur, tal.length());

    rf95.send(retur, sizeof(tal));

    }

    else

    {

        Serial.println("recv failed");

    }

}

else

{

    Serial.println("No reply, is rf95_server running?");

}

/*Foretag måling i det ønskede tidsinterval givet ved den modtagne kommando

* Lav en for-løkke som

*/

// // Tjek om canistor er løftet op til overfladen

if (dybde < 1) {uint8_t data3[] = "Messurement complete 1 meter reached";

rf95.send(data3, sizeof(data3));

Serial.println("Messurement done");

// Herefter slukkes for motor og canister, mens der ventes indtil næste måling
```

```
digitalWrite(releC, LOW);
```

```
digitalWrite(releM, LOW);
```

```
/* Hvor lang tid der skal ventes er givet af antallet af målinger pr. dag
```

```
 * Det skal bemærkes at ikke alle indstillinger er skrevet ind, da koden kan laves smartere og mere  
 fungerer som proff of concept
```

```
 * Der skal tages højde for warmup tid (30 min) for CH4 og tiden en måling varer
```

```
 */
```

```
String tal = ((char*)buf);
```

```
Serial.println(tal);
```

```
if (tal == "1EE")
```

```
{
```

```
    delay(23.5*hour); }
```

```
if (tal == "4AA")
```

```
{ Serial.println("Hold pause for 4AA");
```

```
    delay(10*second);
```

```
    //delay(11.5*hour);}
```

```
}
```

```
if (tal == "4BB")
```

```
{ Serial.println("Hold pause for 4BB");
```

```
    delay(10*second);
```

```
    //delay(7.5*hour);
```

```
}
```

```
if (tal == "4EE")
```

```
{ Serial.println("Hold pause for 4EE");
```

```
    delay(10*second);
```

```
    //delay(7*hour);}
```


}

dybde = 5; // Når motoren slukkes ruller spillet ud og dybden resettes

}

}

A8 - Bom liste med komponenter

Her ses bom listen til indkøb af komponenter til en lignende løsning:

nr	Navn	Antal	Pris	Frugt	Shipping land	Input V	Leverings tid
1	Deph/Pressure sensor (Blue Robotics)	1	451	500	Tyskland	2,5/5,5 V	7 dage
2	Arduino MEGA	1	247	80	DK	7/12 V	2 dage
3	LORA modul SX1276	2	330		DK	3,5/5 V	2 dage
4	Bipolar Stepper motor, 2 faset	1	383	80	DK	3,4 V pr. fase	3-5 dage
5	Stepper motor driver TB660	1	149	80	DK		3-5 dage
6	Temperature sensor (Blue Robotics)	1	376		Tyskland	3,3/5,5 V	7 dage
7	ROV Teather (Blue Robotics) - 6 m	1	183		Tyskland		7 dage
8	ROV Enclosure (Blue Robotics)	1	2712		Tyskland		7 dage
9	DC-DC Step up boost converter	1	99	80	DK		3-5 dage
10	DC-DC Step down converter	1	99		DK		3-5 dage
11	Slæbering	1	119	42,5	China		14 dage
12	MAX 232 (RS-232 converter)	1	68		DK	5 V	2 dage
13	Arduino Uno	1	169		DK	7/12 V	2 dage
14	5V relæ	2	50		DK	5 V	3-5 dage

tot 6297,5

Her ses links til forhandlere:

nr	Navn	Link
1	Deph/Pressure sensor (Blue Robotics)	https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar30-sensor-r1/
2	Arduino MEGA	https://dk.rs-online.com/web/p/arduino/7154084/
3	LORA modul SX1276	https://dk.rs-online.com/web/p/kommunikation-og-tradlos-udvikling/1845117/
4	Bipolar Stepper motor, 2 faset	https://let-elektronik.dk/shop/1290-stepper/601082-bipolar-stepper-motor-with-planet-gear-box-18kgcm/
5	Stepper motor driver TB660	https://minielektro.dk/tb6600-stepper-driver.html?gclid=Cj0KCQjwwLKFBhDPARIsAPzPI-LomsobMmRgKVHfw221kfHUItSc8Vc6rIFaoX87qAPSS2MuGZXacZ8aAgLaEALw_wcB
6	Temperature sensor (Blue Robotics)	https://bluerobotics.com/store/sensors-sonars-cameras/sensors/celsius-sensor-r1/
7	ROV Teather (Blue Robotics) - 6 m	https://bluerobotics.com/store/cables-connectors/cables/cab-nbpuf-4utp-26awg/
8	ROV Enclosure (Blue Robotics)	https://bluerobotics.com/store/watertight-enclosures/8-series/wte8-asm-r1/
9	DC-DC Step up boost converter	https://elektronik-lavpris.dk/p139824/modu0111-dc-dc-step-up-boost-4-35v-converter/
10	DC-DC Step down converter	https://elektronik-lavpris.dk/p139823/modu0106-dc-dc-step-down-konverter-in-4-35vdc-out-13-35vdc-2a-print/
11	Slæbering	https://www.fruugo.dk/black-metal-8-ledninger-kapsel-slip-ring-240v-til-vindmølle-vindmølle-vindmøllegenerator/p-55787695-113039983?language=da&ac=croud&gclid=CjwKCAjwjuqDBh
12	MAX 232 (RS-232 converter)	https://dk.rs-online.com/web/p/kommunikation-og-tradlos-udvikling/8829042/
13	Arduino Uno	https://dk.rs-online.com/web/p/arduino/7697409/
14	5V relæ	https://minielektro.dk/relae-modul-5v-1-kanal.html?gclid=Cj0KCQjw78yFBhCZARIsAOxgSx1jpm-o50yJCZTIBD_A6tzBYFyTUNv-O_Q9iWlhH6KyaXQeSiO6ZvYaAuTKEALw_wcB