

# BEERTRESS

## Analyse

Semesterprojekt 3, Gruppe 8

Peter Wann  
201907121

August Hjerrild Andersen  
201907251

Simon Phi Dang  
201705957

Henry Pham  
201606071

Alexander Flarup Wodstrup  
201810602

Lucas Friis-Hansen  
201811527

Jim Sørensen  
201602614

Shynthavi Prithviraj  
201807198

17. december 2020

# Indhold

<b>1</b>	<b>Risk assessment</b>	<b>2</b>
1.1	Wireless . . . . .	2
1.1.1	WebSocket . . . . .	2
1.1.2	AdHoc . . . . .	2
1.1.3	Konklusion . . . . .	2
1.1.4	Simpel prof of concept . . . . .	3
1.2	Vægtsensor . . . . .	3
1.2.1	Load cell 1 kg . . . . .	3
1.2.2	Load cell 10 kg . . . . .	3
1.2.3	Køkkenvægt . . . . .	3
1.2.4	Konklusion . . . . .	4
1.3	Køretøj . . . . .	4
1.4	Line tracker sensor . . . . .	4
1.4.1	IR-sensor . . . . .	4
1.4.2	TCS230 & TCS3200 . . . . .	5
1.4.3	Konklusion . . . . .	5
1.4.4	Proof of concept . . . . .	5
1.5	Afstandssensor . . . . .	5
1.5.1	Ultrasonisk afstandssensor (HC-SR04) . . . . .	6
1.5.2	Infrarød afstandssensor (2Y0A21YK) . . . . .	6
1.5.3	Konklusion . . . . .	6
1.6	Data kommunikation . . . . .	7
1.6.1	UART . . . . .	7
1.6.2	I2C . . . . .	7
1.6.3	SPI . . . . .	8
1.6.4	Konklusion . . . . .	8
1.6.5	Proof of context . . . . .	8

# 1 Risk assessment

## 1.1 Wireless

For at tilkalde Beertress skal kunderne have en måde at kommunikere med robotten. Der kunne tilsluttes knapper på hvert bord som var bundet med ledning til Beertress. Men dette ville ikke være en optimal løsning, da der skal tages højde for mange nye forhold. Blandt andet at sikre at ledningerne ikke sætter sig fast eller vikles ind i Beertress' hjul, samt at kunder ikke snubler over dem. Dette vil ikke være en elegant løsning. Derfor har vi valgt at benytte en form for trådløs kommunikation mellem kunden og Beertress. Igennem undersøgelse på nettet har vi fundet frem til to mulige løsninger: WebSocket og AdHoc.

### 1.1.1 WebSocket

WebSocket er en full-duplex kommunikations vej imellem to enheder over en webserver. Dette vil give os mulighed for at kunder kan hidkalde Beertress igennem en hjemmeside, som de evt. kan åbne fra telefonen. En lille ulempe ved WebSocket er, at det kræver en internet forbindelse for at fungere. Heldigvis er vi i det 21. århundrede, og der er internet overalt. En fordel ved WebSocket er, at det returnerer et event-driven svar, så vi ikke bliver nødt til at aflæse fra serveren hele tiden.

### 1.1.2 AdHoc

AdHoc tillader at oprette direkte forbindelse mellem to RPi'er. Denne løsning vil kunne benyttes, hvis vi anvender en RPi og en knap på hvert bord til at hidkalde Beertress. Kunderne kan derfra trykke på knappen, når de gerne vil have betjening, hvorefter der vil blive sendt et signal fra f.eks. bord 4's RPi til Beertress. En fordel ved AdHoc er, at det kan benyttes alle steder, da det ikke kræver, at der er en internet forbindelse i forvejen. Tilgængelighed er dens rækkevidde begrænset til 100m, da den selv opretter sit eget hotspot. Men eftersom der ikke er mange bar på over 100m, er dette stadig en mulig løsning.

### 1.1.3 Konklusion

Begge muligheder ville være en fin løsning på en wireless kommunikation. Vi har dog valgt at gå med WebSocket, eftersom det er event driven og vil give os mere frihed til at udvikle et lækkert brugerinterface til kunden, som kan tilgå det fra deres smartphone. Derudover vil vi modtage un-

dervisningen i WebSocket, som vil give os en klar fordel til at komme i gang med projektet i forhold til AdHoc, som vil kræve meget selvstudie uden mulighed for at søge hjælp hos undervisere.

#### **1.1.4 Simpel prof of concept**

Et simpel prof of concept kan være en test, der får en RPi til at aflæse et input fra en hjemmeside.

### **1.2 Vægtsensor**

For at kunne veje noget og give det en konkret vægt i en enhed som kg, er der umiddelbart ikke mange muligheder andet end en load cell, som er en strain gauge. For at måle kraften der bliver givet på en strain gauge, bliver der oftes anvendt en wheatstone bro. Ved hjælp af en wheatstone bro er det muligt at måle spændingsforskellen, der kan konverteres om til et digital signal. Der er så forskellige muligheder for typer af load cells, som beskrives i nedenstående afsnit.

#### **1.2.1 Load cell 1 kg**

En simpel strain gauge har en rækkevide imellem 0 og 1 kg. Uden at påvirke denne load cell på vægt, er det muligt, at den giver positive signaler, hvilket kunne give et offset i vægten der udregnes. For denne load cell er det også muligt, at den kan være ikke linear i måling af små vægte, hvor det kan være nødvendigt at give den noget vægt i forvejen ved at "pre-load" den.

#### **1.2.2 Load cell 10 kg**

Der findes også en 10 kg load cell, som kan måle imellem 0 og 10 kg. Denne load cell er dog en half-bridge strain gauge. Det vil så være nødvendigt at færdiggøre kredsløbet, så der kan måles på spændingsforskellen. Dette gøres ved at tilføje 2 modstande med ens værdier. Denne load cell kan også have problemer med positive værdier, selvom der er intet vægt på den, samt non-linearitet i starten, når man tilføjer små vægte.

#### **1.2.3 Køkkenvægt**

En almindelig køkkenvægt, som allerede er bygget og alt er klar kan give bedre precision og sikre at vægten, der måles, er korrekt. Den har dog det problem, at den skal skilles ad og tilsluttes på vores PSoC for at kunne give signalet til de andre moduler, i stedet for at display signalet på et display.

#### **1.2.4 Konklusion**

Køkkenvægten, der allerede har alt samlet, er meget præcis, men vil give problemer, da den ikke er fleksibel med at give vægten, der måles automatisk videre til vores system. Load cellen, der kan måle op til 10 kg, giver mere mulighed for at udvide vores kapacitet af øl, der kan måles og har lignende mulige problemer, som load cellen, der kan måle op til 1 kg. Der er dog også lidt mere setup for load cellen, der kan måle op til 10 kg, da det er en half-bridge strain gauge. Da Beertress robotten ikke er stor nok til kunne bære 10 kg, var der ikke behov for en vægt, der kan måle op til 10 kg. Der vælges derfor at gå videre med load cellen, som kan måle op til 1 kg.

### **1.3 Køretøj**

Da ingen af os har den rette ekspertise til at bygge køretøjet fra bunden, valgte vi at kigge på hvilke muligheder der var på Embedded Stock. Her fandt vi en færdigbygget tank med motor, som består af ét chassis med to larvefødder, der hver styres af to styk DC-motor. De andre muligheder i Embedded Stock var ikke interessant, da de var for små til den vægtbelastning, vi ville påføre dem.

### **1.4 Line tracker sensor**

Det er blevet besluttet, at Beertress skal anvende farvesensoren 'TCS230' for at køre rundt. Sensoren kan detektere op til fire farver, herunder rød, blå, grøn og en clear farve. Sensoren fungerer ved at sende hvidt lys ud og opfange den reflekterede farve ved hjælp af fotodioderne. Udover farvesensoren var ir-sensor og farvesensoren 'TCS3200' også med i overvejelserne, da der skulle vælges en sensor.

#### **1.4.1 IR-sensor**

En ir-sensor fungerer ved, at sensoren lyser infrarød lys ud til en overflade via en LED. Sensoren måler heraf den reflekterede infrarøde stråling. Ud fra intensiteten af strålingen, kan sensoren bestemme hvor lyst eller mørkt overfladen under sensoren er. Lyse overflader vil reflektere mere infrarød lys end mørke overflader og vil derfor fremkomme lysere for sensoren. Dette gør, at sensoren kan detektere en mørk linje på lys overflade, og lys linje på mørk overflade. Denne sensor ville være god, hvis det blot var en bane, den skulle følge uden stoppesteder. Men da Beertress skal stoppe ved de enkelte borde og slutpositionen, ville denne sensor ikke være den mest optimale.

#### **1.4.2 TCS230 & TCS3200**

Color sensor TCS230 og TCS3200 er to farvesensorer af samme type, men ikke samme model. De har stortset de samme specifikationer, og output fra begge sensorer er ens i form af en registreret strøm-til-frekvens output. Sensorerne er bygget op på samme princip, i og med at de begge består af et 8x8 fotoarray, som kan anvendes på fire forskellige måder, alt afhængig af hvilken farve (rød, grøn, blå, clear) man ønsker at kende frekvensen på. Denne type sensor vil være en oplagt løsning, i forhold til at kunne navigere rundt på en bane med stop undervejs, som det er tænkt i dette projekt.

#### **1.4.3 Konklusion**

Undervejs som de forskellige muligheder er blevet undersøgt, er der kommet frem til, at farvesensoren er smartest, således at banen, bordene og slutposition har hver deres farve. Når det kommer til valget af farvesensor, så var der to muligheder ved Embedded stock på Katrinebjerg. Disse to sensorer var af samme type, men hver deres model. Valget stod mellem TCS230 og TCS3200. De to sensorer er på mange måder ens, hvorfor valget mellem dem var forholdsvis let, da der ikke syntes at være ulemper ved at vælge den ene frem for den anden. Desuden var der kun en TCS230 ledig på daværende tidspunkt i Embedded stock, derfor blev denne valgt.

#### **1.4.4 Proof of concept**

Et eksempel på proof of concept kunne være en test, hvor farvesensoren detekterer rød farve, som udskrives på et konsolvindue. Der observeres, at konsolvinduet udskriver 'RED', når der sættes et stykke rødt karton under sensoren.

### **1.5 Afstandssensor**

Der var to afstandssensorer til rådighed via Embedded Stock. Det var to forskellige typer, en ultrasonisk sensor og en infrarød afstandssensor. For at finde den rette afstandssensor blev de to undersøgt grundigere.

Før de to sensorer blev undersøgt, blev der sat prioriteter op. Her blev der fundet frem til de 2 vigtigste prioriteter:

Prioritet 1 var, at den kunne detektere objekter i mørke, da idéen bag Beertress bl.a. er den skal køre rundt på restauranter og evt. mørke barer.

Prioritet 2 var, at kunne detektere objekter så langt som muligt. Til dette projekt er en kortere detektionsafstand acceptabel, da Beertress ikke forventes at køre hurtigt. Men til en evt. fremtidig skalering, hvor Beertress kan køre hurtigere, er det vigtigt at kunne detektere objekter fra en længere afstand.

#### 1.5.1 Ultrasonisk afstandssensor (HC-SR04)

Denne sensor benytter lydbølger til at detektere afstand. Sensoren sender en lydbølge ud. Lydbølgen ”rammer” et objekt, og bliver kastet tilbage, som sensoren modtager. Derefter kan der ud fra tidsperioden, hvor lydfrekvensen har været ”væk”, beregnes afstand mellem objekt og sensor.

Nogle af fordelene ved denne sensor, er at den ikke påvirkes af støv, regn og sne. Derudover kan den detektere objekter op til 400 cm væk.[3]

Nogle af dens begrænsninger er dog, at den har svært ved at detektere ”bløde”, kurvede, tynde og små objekter.[1] Temperatursvingninger kan også påvirke præcisionen, da lydbølger bliver langsommere i koldere vejr.[4]

#### 1.5.2 Infrarød afstandssensor (2Y0A21YK)

Afstanden på den infrarøde (IR) afstandssensor aflæses ved at kigge på spændingsoutputtet. Den stiger, desto tættere et objekt er på sensoren. IR-sensoren har ikke problemer med at detektere kurvede, tynde og små objekter.

Når detektoren modtager direkte sollys eller møder noget reflekterende, har den svært ved at måle den præcise afstand. Derudover skal linsen til sensoren holdes ren, da støv, vand og olie kan forringe præcisionen. Den kan detektere objekter op til 80 cm væk. Sensoren anvender en JST Connector. [2]

#### 1.5.3 Konklusion

**Lys** påvirker ikke den ultrasoniske afstandssensor. IR-sensoren kan påvirkes af reflekterende objekter, og det må antages at der ved indendørs brug er risiko for at køre forbi vinduer.

**Detektionsafstanden** for den ultrasoniske sensor er op til 400 cm, altså 5 gange større end for IR-sensoren. Det vil derfor være nemmere at skalere Beertress med den ultrasoniske sensor.

Den ultrasoniske afstandssensor benyttes, da den opfylder begge prioriteter bedre end IR afstandssensoren. Derudover er der ikke mulighed for hverken i elektronikværkstedet eller i Embedded Stock at få en JST Connector, som IR-sensoren skal bruge. En lille ulempe er dog, at den ultrasoniske sensors præcision påvirkes en smule af temperaturen. Dette forventes dog ikke at blive et problem for Beertress, da den er beregnet til indendørs brug.

## **1.6 Data kommunikation**

Der er mange forskellige måder at kommunikere med data på. Medlemmerne i gruppen er hidtil blevet præsenteret for tre forskellige - UART, I2C og SPI.

### **1.6.1 UART**

UART står for Universal Asynchronous Receiver-Transmitter og fungerer ved at sende først en startbit, dernæst dataframen, så en eventuel paritets bit og til sidst en slut bit. Det er som sagt en asynkron kommunikationsmetode, så det er kun når der sendes en start bit, at den begynder at læse.

Fordele ved UART er, at det er forholdsvis nemt at implementere og ret let at forstå. Der er få pins der skal forbindes og det kan køre på en høj hastighed.

En ulempe ved UART er, at det er noget vi har beskæftiget os med før, nemlig i Semesterprojekt 2. Rent personligt udviklingsmæssigt, vil det derfor ikke være så spændende at arbejde med igen, fremfor at undersøge nye kommunikationsmetoder. Derudover kan UART max sende 9 bits i dataframe. Det kunne muligvis også blive et problem.

### **1.6.2 I2C**

I2C er en seriel, synkron kommunikationsform, som består af en master og mindst en slave. Der kan godt være op til flere slaver, hvis dette er ønsket. I2C sender data i beskeder på en byte af gangen, og er afhængig af signaler kaldet ACK eller NACK efter hver byte, til at vide, om den kan sende eller læse mere til eller fra slaven.

Fordele ved I2C er, at den kun består af 2 pins, SDA og SCL. Derudover vil den kunne kommunikere rigelig hurtigt, i forhold til hvad vi har behov for. I2C er også forholdsvis let at forstå. Master/slave forholdet giver også god mening i vores projekt, da RPi'en kommer til at sende ordre ud, når den har en. Derudover er der implementeret en I2C-driver i linux man vil kunne benytte.



Ulemper ved I2C kan være, at den kun kan sende beskeder på 8 bit af gangen. Derudover skal der noget logik til for at masteren kan vide, hvornår den kan læse eller skrive til slaven. Hvis Beertress er i gang med en ordre, må masteren jo ikke sende en ny ordre.

### 1.6.3 SPI

SPI er en seriel, synkron kommunikationsform, som består af en master og mindst en slave. SPI sender data i full-duplex, og hver gang der sendes noget modtages der også noget. SPI sender/modtager informationer via shift registers, hvor der for hver clock sendes og modtages en bit. Den samlede byte er fra starten placeret i shift registeret.

Fordele ved SPI er, at det super hurtigt. Det er derudover muligt at sende og modtage på samme tid, hvilket i nogle tilfælde kan være smart.

Ulemper ved SPI er, at det består af 4 pins modsat UART og I2C, som har 2 pins. Da der i forvejen skal bruges mange ledninger i dette projekt kan dette være u hensigtsmæssigt. Derudover vil det ikke give speciel god mening i vores projekt, at få informationer hver gang man sender noget. Det er altså en feature, som ikke rigtig er nyttig i dette projekt.

### 1.6.4 Konklusion

Det lette valg ville være UART, da det er noget vi har brugt før, og derfor let ville kunne implementere igen. Vi har dog valgt at bruge I2C, da dette virker mere logisk for os end SPI, da vi ikke har behov for at sende og læse på samme tid. Og vi er ikke interesseret i u hensigtsmæssigt mange pins, da vi kun skal bruge en slave.

### 1.6.5 Proof of context

Læs en byte fra en temperatur måler, som også skal laves i faget HAL. Det observeres, at der kan læses 2 bytes.

## Litteratur

- [1] *Advantages and Limitations of Ultrasonic sensors.* <https://www.maxbotix.com/articles/advantages-limitations-ultrasonic-sensors.htm/>.
- [2] *Datablad for 2Y0A21YK.* [https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf).

[3] *Datablad for HC-SR04.* <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.

[4] *Speed of sound.* [https://www.engineeringtoolbox.com/speed-sound-d\\_82.html](https://www.engineeringtoolbox.com/speed-sound-d_82.html).