



11-10-2017

# Arkitektur og Kravspecifikation

PRJ gruppe 15  
SEMESTERPROJEKT 3

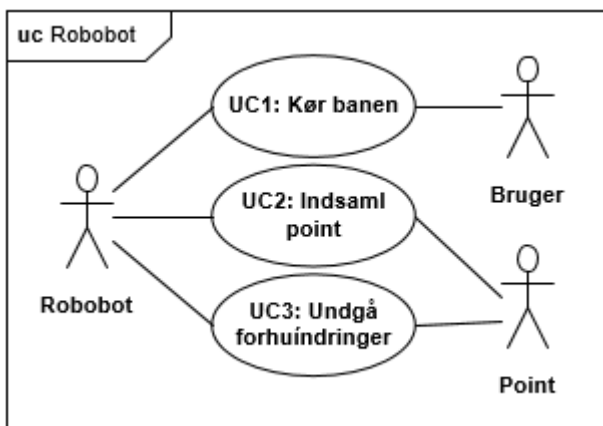
## Indhold

Kravspecifikation .....	2
Use cases for RoboBot.....	2
UC1: Kør banen.....	2
UC2: Indsaml point .....	2
UC3: Undgå forhindringer .....	3
Sekvensdiagram for hele systemet, udkast.....	3
Use case 1, fully dressed.....	4
Use case 2, fully dressed.....	5
Use case 3, fully dressed.....	5
Aktør beskrivelse .....	6
MoSCoW .....	6
Accepttest for funktionelle krav .....	6
Accepttest for ikke-funktionelle krav .....	7
HW arkitektur .....	9
BDD .....	10
Blokbeskrivelse .....	10
IBD .....	12
Grænseflader .....	13
Signalbeskrivelse .....	14
Signaltypebeskrivelse .....	15
SW arkitektur .....	16
Domænenemodell.....	17
Klassediagrammer .....	17
SD for UC1.....	19
SD for PSoC (slave).....	19
SD for RPi (master) .....	20
SD for UC2.....	21
SD for PSoC (slave).....	21
SD for RPi (master) .....	22
SD for UC3.....	23
SD for PSoC (slave).....	23
SD for RPi (master) .....	23
UML for hele systemet, udkast.....	24

## Kravspecifikation

- Use Cases, udkast
- Samlet SD for alle UC, udkast
- UC 1 fully dressed
- UC 2 fully dressed
- UC 3 fully dressed
- Aktørbeskrivelse
- MoSCoW
- Accepttest for funktionelle krav
- Accepttest for ikke-funktionelle krav

### Use cases for RoboBot



Billede 1: UC diagram

#### UC1: Kør banen

Aktører: Robobot (primær), bruger (sekundær)

Initialisering: Bruger

Precondition: Der er en tom/ledig bane med streger af sort tape

Banen har en startpunkt (hvor robotten starter) og et mål, hvor robotten skal nå hen

Postkondition: robotten har kørt banen og er kommet i mål

Hovedscenarie:

1. Brugeren placerer robotten på punkt 1 og tænder for den
2. Robotten afspiller en start-lyd
3. Robotten kører med hastigheden 0,1-0,2 m/s, mens den følger den sorte tape
4. Robotten finder frem til målet vha. LSTM (long short term memory)

#### UC2: Indsaml point

Aktører: Robobot (primær), point (sekundær)

Initialisering: Robobot

Precondition: Der er en tom/ledig bane med streger af sort tape og hvide prikker, på tapen, som repræsenterer point

Postkondition: Robotten har kørt banen og er kommet i mål efter af have indsamlet et X antal point

Hovedscenarie:

1. Robotten kører på banen og følger den sorte tape
2. Robotten passerer en hvid prik
3. Robotten laver en lyd og tænder for en LED
4. Når robotten har alle LEDer tændt, kører den i mål

### UC3: Undgå forhindringer

Aktører: Robobot (primær), point (sekundær)

Initialisering: Robobot

Precondition: Der er en tom/ledig bane med streger af sort tape, hvide prikker og brikker med størrelsen 0,25x0,25x0,25, som repræsenterer forhindringer

Postkondition: Robotten har kørt banen og er kommet i mål efter af have indsamlet et X antal point, mens den undgår forhindringer

Hovedscenarie:

1. Robotten kører på banen og følger den sorte tape
2. Robotten afspiller en positiv lyd, og tænder for en af dens LEDer, når den passerer en hvid prik
3. Robotten drejer eller bakker, eller gør ingenting hvis den ser en forhindring [extension 1], [extension 2]
4. Robotten drejer for at undgå forhindringen
5. Robotten kører videre, indsamler point og kommer i mål

Extension 1: Robotten bakker

1. Robotten bakker, når den ser forhindringen i mangel på en forgrening af banen
2. En af robottens LEDer slukker og den afspiller en negativ lyd
3. Robotten kan ikke samle point op de næste 10 sekunder

Extension 2: Robotten gør ingenting

1. Robotten kører ind i forhindringen
2. Robotten stopper, alle dens LEDer slukker og den laver en stop-lyd

Positiv lyd: en lyd på to noder, som bliver højere

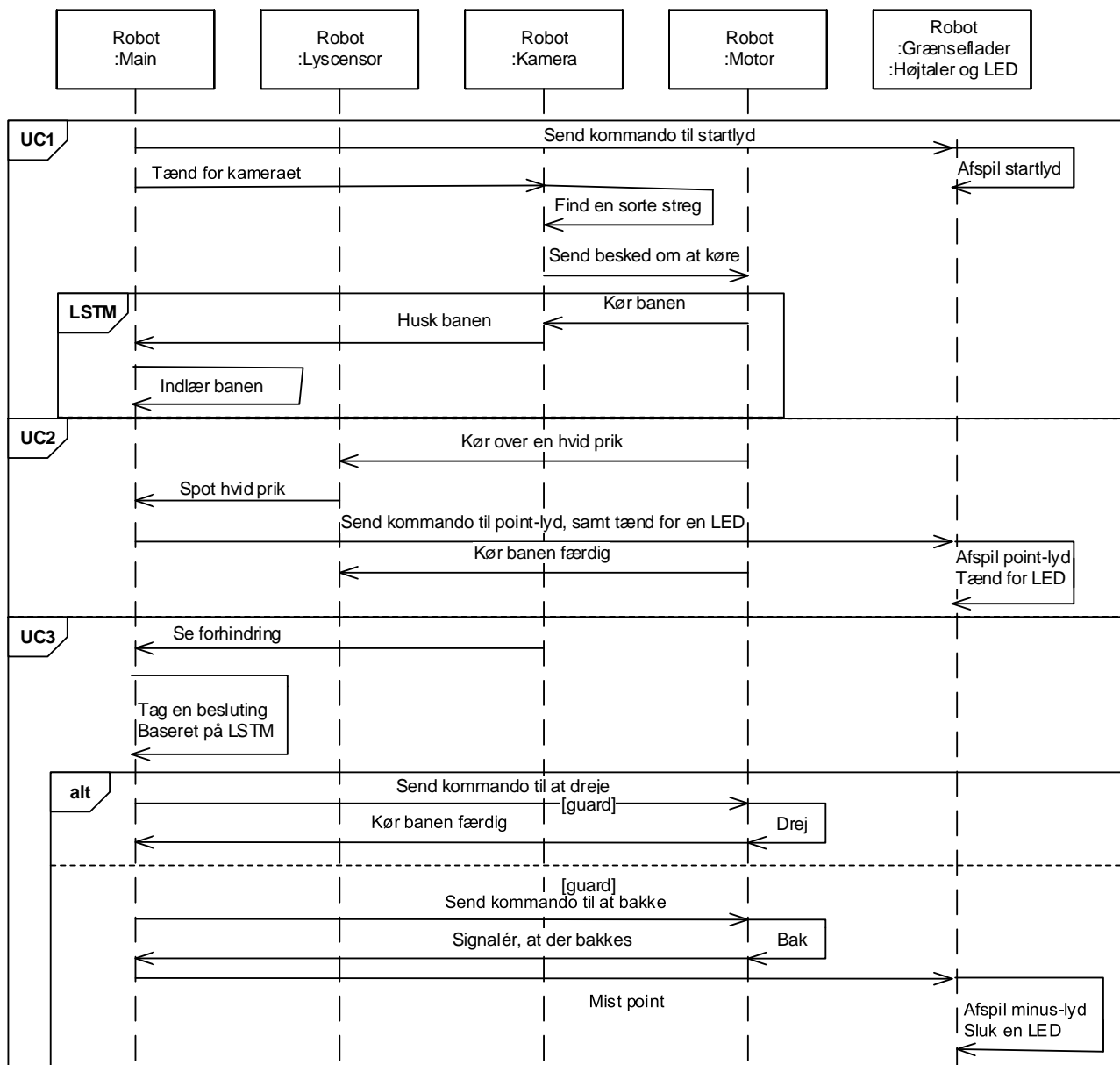
Negativ lyd: en lyd på to noder, som bliver lavere

Disse kan f.eks. være check-ind og fejl ved check in på rejsekort-lyde

Start-lyd og stop lyd: en kort melodi (der er forskellige fra hinanden og positive/negative lyde)

### Sekvensdiagram for hele systemet, udkast

Dette SD bruges kun som grundlag for fully dressed use cases og dele af SW arkitektur.



### Use case 1, fully dressed

Navn	Use case 1: Kør banen
Mål	Robot kører autonomt rundt på banen og ender i målområde
Initiering	Bruger
Aktører	Robot (primær) bruger(sekundær)
Antal samtidige forekomster	1
Prækondition	Robot står på banen og er tændt og brugergrænsefladen er åben
Postkondition	Robot har gennemført banen og står i målområde
Hoved scenarie	<ol style="list-style-type: none"> <li>Bruger starter robot via brugergrænseflade.</li> <li>Robot kører rundt på banen.</li> </ol> <p>[ext. 1.]</p> <ol style="list-style-type: none"> <li>Robottens kamera finder point punkt på banen.</li> </ol> <p>[ext. 2.]</p>

	4. Robottens kamera finder mål område. [ext. 3.] 5. Robotten kører ind i målområde. 6. Robotten kommunikerer til brugergrænseflade om at have nået i mål 7. Brugergrænsefladen viser brugeren, at robotten er nået i mål.
<b>Udvidelser / undtagelser</b>	[ext. 1.] 1. Robottens sensor måler afstand til en forhindring 2. kør UC 3. [ext. 2.] 1. kør UC 2. [ext. 3.] 1. Robotten når målområdet uden nok point. 2. Robottens kamera anser opmåling af målområde som en del af banen. 3. Robotten fortsætter fra UC1. punkt 3.

### Use case 2, fully dressed

Navn	Use case 2: Indsaml Point
<b>Mål</b>	Robotten læser point punkt på bane og tænder en diode per point samlet ind
<b>Initiering</b>	Brugeren har startet robotten
<b>Aktører</b>	Robot (primær)
<b>Antal samtidige forekomster</b>	1
<b>Prækondition</b>	Robot er startet og har ikke indsamlet alle pointene
<b>Postkondition</b>	Robot har indsamlet alle pointene på banen.
<b>Hoved scenarie</b>	8. Starter fra UC1 punkt 3. 9. Robot kører over pointpunkt. 10. Robottens kamera læser punkt. [Ext 1.] Pointpunktet er allerede læst. 11. Robotten tænder lysdiode. 12. Robotten kommunikerer til brugergrænsefladen om antal point indsamlet. 13. Robotten forsætter indtil alle point er indsamlet. [Ext 2.] Alle point indsamlet
<b>Udvidelser / undtagelser</b>	[Ext 1.] 1. Robottens kamera ignorerer punktet. 2. Kør UC2 fra punkt 2. [Ext 2.] 1. Kør UC1 fra punkt 6.

### Use case 3, fully dressed

Navn	Use case 3: undgå forhindring
<b>Mål</b>	Robottens sensor møder forhindring, finder en ny rute for at undgå forhindring
<b>Initiering</b>	Robotten kører på banen
<b>Aktører</b>	Robot (primær)
<b>Antal samtidige forekomster</b>	1
<b>Prækondition</b>	Robot kører UC 1.
<b>Postkondition</b>	Robot har mødt forhindring og undgået den ved at finde ny rute.

<b>Hoved scenarie</b>	14. Robottens sensor finder forhindring på banen. 15. Robotten udregninger ny rute. [ext 1.] Robotten bliver nødt til at bakke 16. Robotten fortsætter med UC1 & UC2
<b>Udvidelser / undtagelser</b>	[ext 1.] 1. Robotten bliver fratrasket point. 2. Robotten slukker for lysdiode. 3. Robotten kommunikerer til brugergrænsefladen at point er blevet fratrasket.

### Aktør beskrivelse

**Bane:** Banen vil være et opmarkeret område af sort tape på en hvid baggrund. Banen vil være vinklet i 90 grader. Målområdet vil blive markeret i den bestemt farve, f.eks. Grøn tape. Som vil være et firkantet område i midten af banen forbundet fra flere sider.

**Point Punkt:** Et point punkt, vil være en bestemt farve for at kunne blive registreret af robotten. Punkterne vil bestå af unikke farver for at robotten kan skelne imellem de forskellige punkter.

**Forhindring:** En forhindring, vil være et objekt som blokere banen, som gør at robotten må finde en ny rute, f.eks. En kasse som fysisk blokker for robotten.

### MoSCoW

MoSCoW	Beskrivelse
Must	Et enkelt kamera bruges til at se striben på jorden.
Must	Robotten er drevet af et batteri. Robotten får strøm udelukkende fra et batteri.
Must	Robotten har 2 hjul som styres individuelt for at dreje.
Must	DC-DC omformer til at forsyne RaspberryPi, PSoC mm.
Should	Farvesensor til at registrer mål og point.
Should	Robotten har 3 LEDer som kan indiker point.
Should	Robotten har en højttaler, som kan afspille en start lyd
Should	Kompassensor. En sensor som fortæller hvilken retning robotten peger. Ved retning forstås der: Hvor mange grader fra nord, altså en enkelt rotationsakse.
Should	Have en brugergrænseflade, hvorfra man kan starte robotten og holde øje med dens rute, antal point etc
Could	Batteristatus. Gør det muligt at se hvor meget strøm der er tilbage på batteriet.
Could	LSTM. Dette vil gøre det muligt for robotten at indlære labyrinten over flere generationer

### Accepttest for funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt
1	UC1 gennemført	Robotten placeres ved start på banen. Brugeren interagerer med robotten via en brugergrænseflade	Robotten holder i mål. Brugergænsefladen fortæller brugeren hvor lang tid det tog at gennemføre		

2	UC2 gennemført	Robotten kører over en farvet prik og indsamler point	Robotten indsamler point ved at afspille en positiv lyd og tænde en LED. Efter gennemførelsen vises antal point via brugergrænsefladen		
3	UC3 gennemført	Der er en forhindring på banen. Robotten undgår forhindringen ved enten at dreje eller bakke og dermed miste point	Hvis robotten drejer, kører den videre og indsamler point uden konsekvenser. Bakker robotten mister den point, slukker en LED og siger en negativ lyd		

### Accepttest for ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/kommentar
1.a	Use case 1 "Kør bane" gennemføres på under 10 minutter.	Robotten placeres ved punkt 1 på banen. En person trykker på start knappen på robotten og et stopur startes.	Robotten holder i mål før der er gået 10 minutter.		
1.b	Robotten kan følge en streg på 5 cm +/- 1 cm. I bredden.	Robotten placeres oven på cirkel kanten. Cirklen er optegnet på hvidt underlag og har en radius på 50 cm +/- 1cm. Cirklen er påtegnet med en sort streg, der har en tykkelse på 5cm +/- 1cm. En person tænder for robotten efter 30 sekunder stoppes robotten.	Robotten holder på cirkel kanten med strengen imellem de to trækkende hjul i samme retning som den startede.		



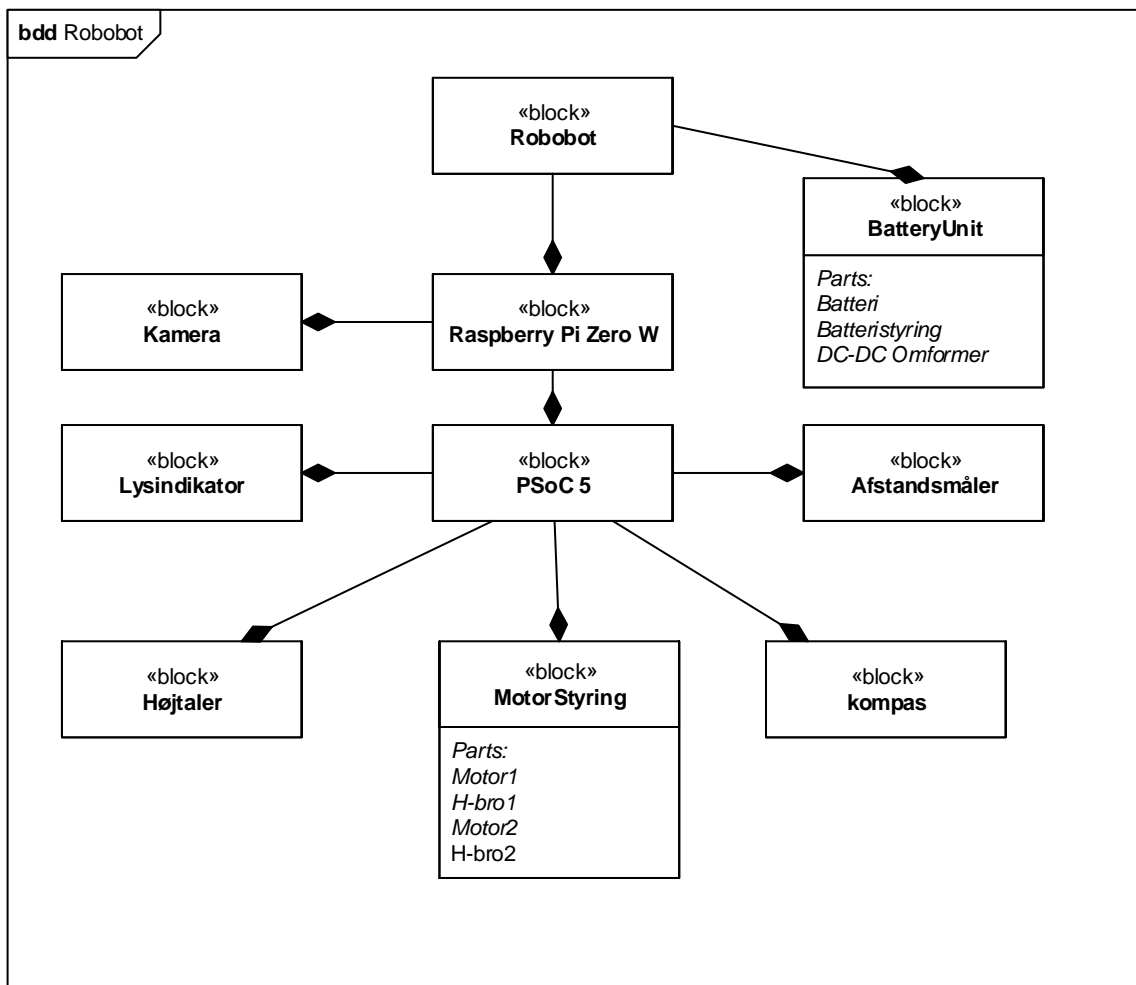
1.c	Startlyden skal starten inden for 5 sek. efter at robotten er blevet tændt	Et stopur startes når en person trykker på tænd på robotten.	Funktionen igangsættes inden for 5 sek. efter at der er trykket på tænd-knappen.		
1.d	Maks. 2 sekunder efter robotten er kørt over en prik skal den afspille en positiv lyd* og tænde en LED	Robotten sætte på en streg med en prik forude*. Robotten tændes, så den kører frem over prikken.	Robotten afspiller positiv-lyd* inden for 2 sekunder fra prikken passeres.		
2.a	Robottens rotation kan bestemmes med en præcision på +/- 10°	robotten sættes på en streg med et 90° sving forude. Robotten tændes, så den kører frem og tager svinget.	På web-interfacet ud for kompas ses en 80°-100° ændring i svingets retning.		
2.b	Batterilevetid: Use case 1 skal kunne gennemføres minimum 6 gange før batteriet er dødt	Sikre at batteriet er fuldt opladt(12.6V). Gennemfør use case1 seks gange.	Robotten har gennemført use case 1 seks gange og batterispændingen er over 3.4V.		
2.c	Tophastighed er min 0.2 m/s	Robotten sættes på en 3 meter lang lige streg og startes så den kører langs strengen. Ved afstandende 1m. og 2m. sættes to tjekpunkter, hvorved et stopur hhv. startes og stoppes, når robotten passerer.	Tiden mellem de to tjekpunkter er maks. 5 sek.		

3.a	MTBF (mean time between failure): Use case 1 skal kunne gennemløbes 5 gange i træk uden at der forekommer fejl.	Gennemløb use case 1 fem gange.	Robotten har i alle gennemløb fulgt use casen afvigelse eller sammenbrud.		
3.b	Der foretages minimum 3 målinger i sekundet med kameraet.	Kameraet rettes mod en flade hvis farve skiftes mellem sort og hvis med en frekves på 3Hz.	Der er i kameraets log registreret 3 farveskift pr. sek.		
3.c	Kompassensoren kan aflæses med en frekvens på minimum 10 Hz	Robotten roteres.	Kompas sensorens log viser min. 10 målinger pr. sek.		

## HW arkitektur

- BDD
- Blokbeskrivelse
- IBD
- Grænseflader
- Signalbeskrivelse
- Signaltypebeskrivelse

## BDD



## Blokbeskrivelse

### Raspberry Pi Zero W

En lille computer som kører linux distributionen Lubuntu. Koden til at styre robotten kører her. Information modtages fra PSoCen og kommandoer sendes ud til PSoCen. Desuden kører koden til billedbehandling af video fra kameraet også her.

### BatteryUnit

Har til formål at forsyne hele systemet med 5VDC og 11.1VDC til motoren. BatteryUnit indeholder batteri og batteristyring samt en DC-DC omformer.

DC-DC omformer består af en lineær low drop out spændingsregulator, LM317. DC-DC omformeren bruges til at forsyne de moduler, som skal bruge 5 Volt forsyning. Se grænseflade beskrivelse.

Batteri er et 3 cell LI-PP på 11.1 Volt. Batteri er power storage for robotten.

Batteristyring sikre at spændingen på batteriet ikke kommer under 9V. For at sikre batteriet ikke tager skade.

## **Kamera**

Officielt raspberry pi kamera: "Camera module V1". Kameraet er forbundet direkte til raspberry pi via CSI forbindelse. Bruges til at orientere robotten i forhold til stregen.

## **Lysindikator**

3 LEDer som styres af PSoC 5, Lysindikator indikerer antallet opsamlede point på banen og mål. De 3 LEDer tændes en ad gangen efterhånden som der samles point og blinker når målet er nået.

## **PSoC 5**

PSoC (Programmable system on chip) består af en microcontroller samt fpga. PSoC 5 bruges til at kommunikere med sensorer og styre forskellige moduler ud fra kommandoer som modtages fra Raspberry pi.

## **Afstandsmåler**

Afstandsmåleren består af et lille PCB board. Afstandsmåleren måler afstanden hen til nærmeste forhindring på banen og er forbundet til PSoCen via I2C.

## **Motorstyring**

Motorstyringen består af 2 H-broer og 2 DC motor. Motorstyring har skal få robotten til køre, dreje og ændre hastighed ud fra hvad PSoCen giver besked på.

H-bro1 består af en L298, der har indbygget et logik kredsløb der sikre at H-broen ikke kan kortslutte. H-bro1 er forbundet til motor1 og PSoC 5. H-bro1 modtager 2 logik, 1 PWM signal fra PSoC 5, de 2 logik signaler bestemmer hvilken retning H-broen skal få motoren til at dreje. H-bro1 er også forbundet til BatteryUnit, for at kunne få energi nok til at få motoren til at dreje.

H-bro2 består af en L298, der har indbygget et logik kredsløb der sikre at H-broen ikke kan kortslutte. H-bro2 er forbundet til motor2 og PSoC 5. H-bro2 modtager 2 logik, 1 PWM signal fra PSoC 5, de 2 logik signaler bestemmer hvilken retning H-broen skal få motoren til at dreje. H-bro2 er også forbundet til BatteryUnit, for at kunne få energi nok til at få motoren til at dreje.

Motor1 er motoren i bagbords siden.

Motor2 er motoren i styrbords siden.

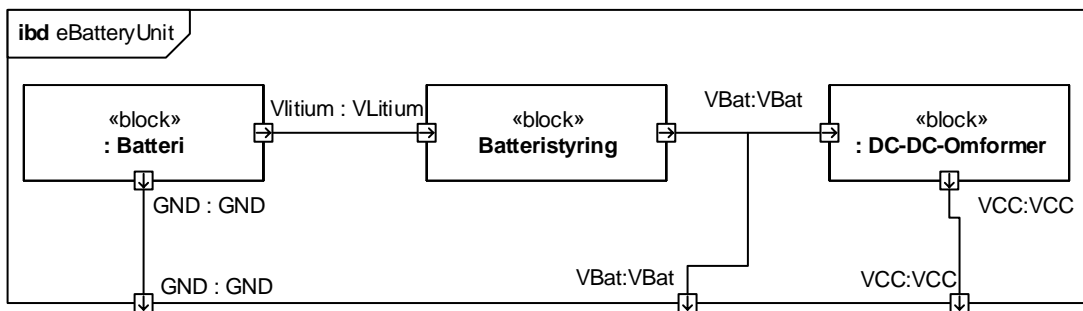
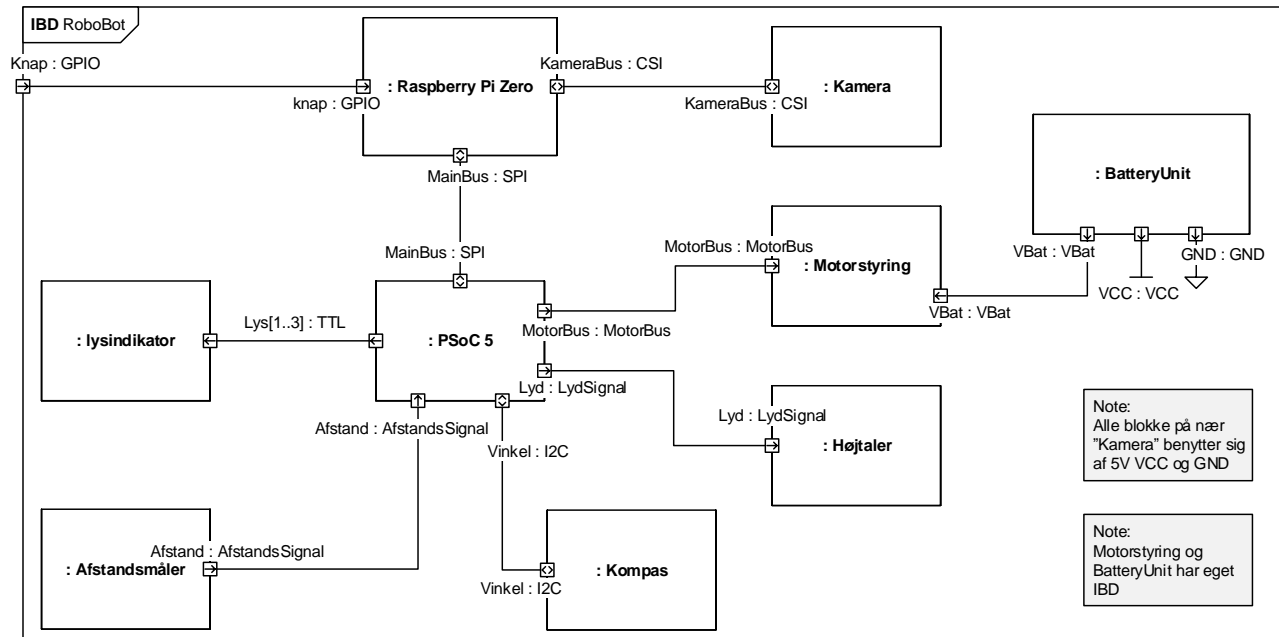
## **Højttaler**

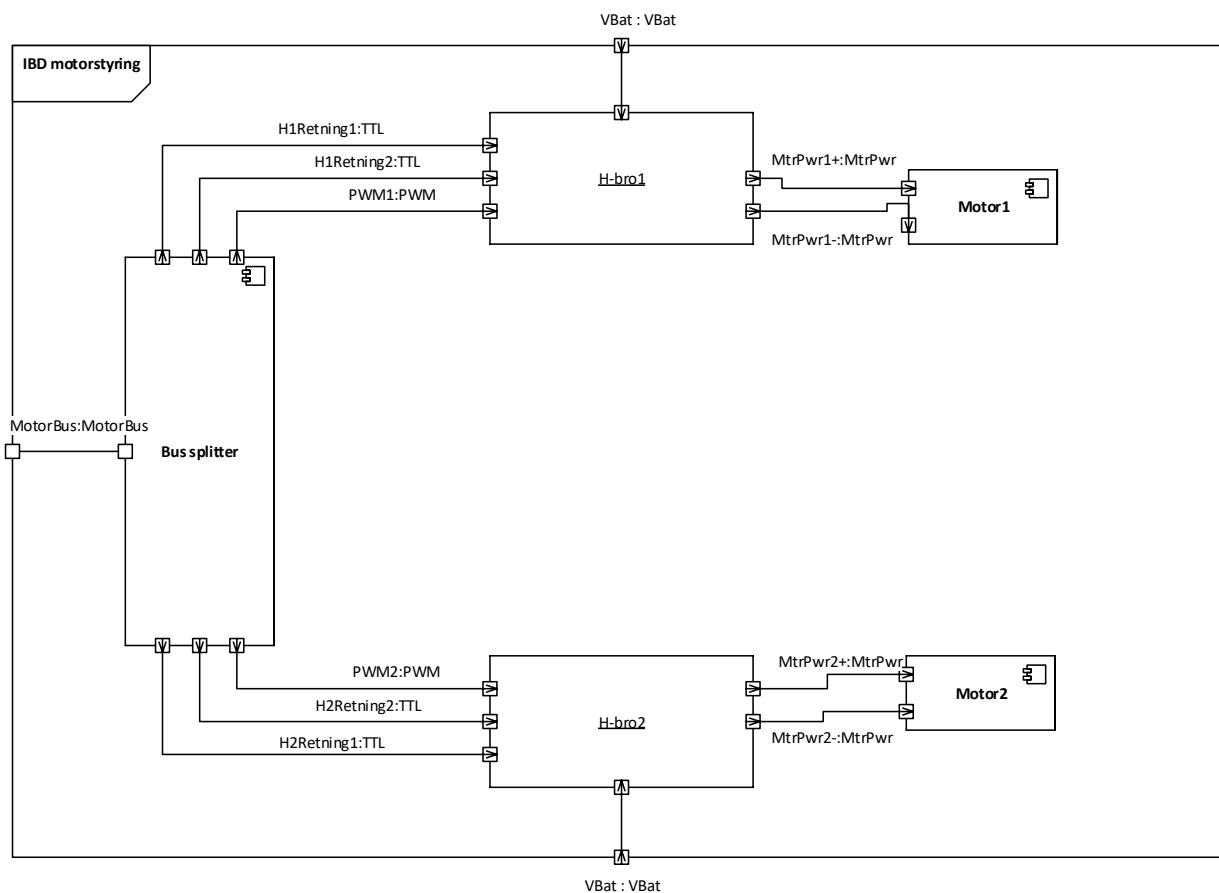
Simple højttaler som kan afpille nogle enkelte toner, for eksempel når der opsamles et point.

## **Kompas**

Kompas består af en HMC6352. Kompasset er en sensor som måler robottens retning ved hjælp af magnetometre. PSoC 5 spørger HMC6352 om retning ved at sende 'A' over I2C hvorefter HMC6352 sender resultatet som en vinkel i grader i forhold til nord.

## IBD





## Grænseflader

Grænseflader - Hele system IBD				
Blok	Blok-beskrivelse	Signalnavn : type	Fysisk-pin	Kommentar
Raspberry Pi Zero (RPI)	Central-modul som styrer robotten	Knap : GPIO		
		MainBus : SPI		
		KameraBus : CSI		
Kamera	Kamera som filmer vejen foran robotten	KameraBus : CSI		
Psoc 5	Styring til hardware.	MainBus : SPI		
		MotorBus : MotorBus		
		Vinkel : I2C		
		Lyd : LydSignal		
		AfstandsSignal:Afstand		
		Lys[1..3] : TTL		
Lysindikator	Indikator til at tælle point	Lys[1..3] : TTL		
Højttaler	Bruges til at spille en startlyd	Lyd : LydSignal		
Kompas	Bruges af robotten til retningsbestemmelse	Vinkel : I2C		

Afstandsmåler	Bruges til at se point og mål	AfstandsSignal:Afstand		
Motorstyring	Bruges til styring af de to motorer på robotten	MotorBus: MotorBus VBat : VBat		
BatteryUnit	Bruges til sikring af batteri og til at lave 5V til system	VBat : VBat		
Grænseflader - Motorstyring IBD				
Bus splitter	Er blot en visuel repræsentation af MotorBus 'signalets' indhold. Motorbus består af 6 signaler.	MotorBus: MotorBus		
		H1Rening1 : TTL		
		H1Rening2 : TTL		
		PWM1 : PWM		
		H2Rening1 : TTL		
		H2Rening2 : TTL		
		PWM2 : PWM		
H-bro1	H-bro til motor 1	H1Rening1 : TTL		
		H1Rening2 : TTL		
		PWM1 : PWM		
		VBat : VBat		
		MtrPwr1+:MtrPwr		
		MtrPwr1-:MtrPwr		
Motor1	Motor i venstre side	MtrPwr1+:MtrPwr		
		MtrPwr1-:MtrPwr		
H-Bro2	H-bro til motor 2	H2Rening1 : TTL		
		H2Rening2 : TTL		
		PWM2 : PWM		
		VBat : VBat		
		MtrPwr2+:MtrPwr		
		MtrPwr2-:MtrPwr		
Motor2	Motor i højre side	MtrPwr2+:MtrPwr		
		MtrPwr2-:MtrPwr		
Grænseflader - BatteryUnit IBD				
Batteri	11.1 V litium batteri	VLitium : VLitium		
		GND : GND		
Batteristyring	Sikring til afladning af batteri	VLitium : VLitium		
		VBat:VBat		
DC-DC-Omformer	Omformer Batteriets spænding til 5V DC.	VBat : VBat		
		VCC : VCC		

## Signalbeskrivelse

<b>SignalBeskrivelse - Hele system IBD</b> (rækkefølge ud fra grænseflader)					
Signalnavn	Type	Indre-signal	Signal-Beskrivelse	Fra blok → til blok	Fra pin → til pin
Knap	GPIO		Reset af robot, grænseflade til bruger	[Bruger] → Rpi	
MainBus	SPI	MOSI	Master-out-slave-in	Rpi → Psoc 5	X → Y

		MISO	Master-in-slave-out	Rpi → Psoc 5	
		SCLK	Clocksignal fra master	Rpi → Psoc 5	
KameraBus	CSI		Camera Serial Interface	Rpi ↔ Kamera	
MotorBus	MotorBus	H1Rening1	Styring af H-Bro 1	Psoc 5 → Ms*(H-bro1)	
		H1Rening2		Psoc 5 → Ms(H-bro1)	
		H2Rening1	Styring af H-Bro 2	Psoc 5 → Ms(H-bro2)	
		H2Rening2		Psoc 5 → Ms(H-bro2)	
		PWM1	PWM til H-Bro 1	Psoc 5 → Ms(H-bro1)	
		PWM2	PWM til H-Bro 2	Psoc 5 → Ms(H-bro2)	
Vinkel	I2C	SDA	Data line	Psoc 5 ↔ kompas	
		SCL	Clock line	Psoc 5 ↔ kompas	
Lyd	LydSignal		Frekvens til højttaler		
AfstandsSignal	Afstand		Afstand foran robot	Afstandsmåler → Psoc 5	
Lys[1..3]	TTL		On/off signal til LED	Psoc 5 → lysindikator	
VBat	Batteri-spænding		Spænding direkte fra batteristyring	Batteristyring → Ms(H-bro1) Batteristyring → Ms(H-bro2)	
<b>SignalBeskrivelse - Motorstyring IBD</b>					
MtrPwr1+	MtrPwr		+ Forbindelse til motor 1	Ms(H-bro1) → Motor 1	
MtrPwr1-	MtrPwr		- Forbindelse til motor 1	Ms(H-bro1) → Motor 1	
MtrPwr2+	MtrPwr		+ Forbindelse til motor 2	Ms(H-bro2) → Motor 2	
MtrPwr2-	MtrPwr		- Forbindelse til motor 2	Ms(H-bro2) → Motor 2	
<b>SignalBeskrivelse - BatteryUnit IBD</b>					
VLitium	VLitium		Spænding direkte fra batteriet	Batteri → Batteristyring	
GND	GND		Systemets common-ground	Batteri → GND	
VCC	VCC		5 Volts forsyning til system	DC-DC-Omformer → [VCC]	

\* **Ms = Motorstyring**

## Signaltypebeskrivelse

### Signaltypebeskrivelse

#### I2C

I2C (inter-integrated circuit) er en kommunikationsstandard som bruger 2 ledninger, SCL og SDA. SCL er clocken og SDA er data. Ved I2C kommunikation er der en master og en eller flere slaver, og full duplex er muligt. Hver slave har en adresse, som sendes ud først for at specificere hvilken slave der skal kommunikeres med. For hver byte sendes der også en ACK (acknowledge) eller NACK bit som bestemmer om der skal sendes yderligere data.

#### SPI

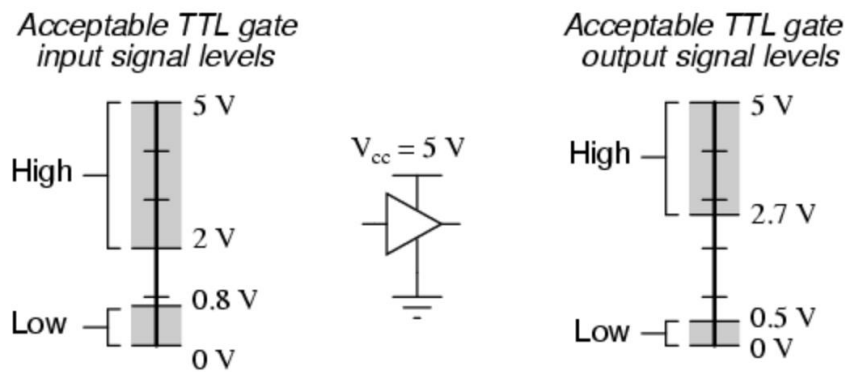
Serial peripheral interface-bus er en kommunikationsstandard med mulighed for full duplex. Der er minimum 4 ledninger i en SPI-bus.

- MOSI: data sendes fra master til slave.
- MISO: data fra slave til master
- Clock
- Slave select: Der er en ledning til slave select for hver slave. Slave select aktiveres for den slave der snakkes med.



## TTL

En standard for hvilke spændinger der tolkes som high, og hvilke der tolkes som low. Acceptable spændinger for input og output ses her:



## GPIO

En pin som via software kan sættes til enten høj eller lav.

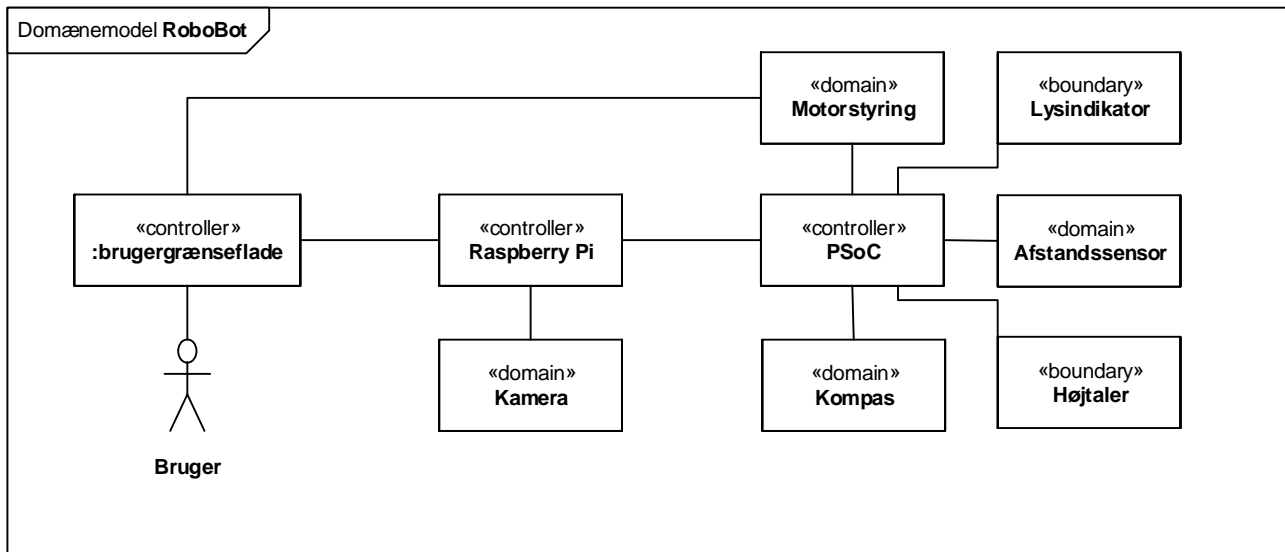
## Kilder

<https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/>

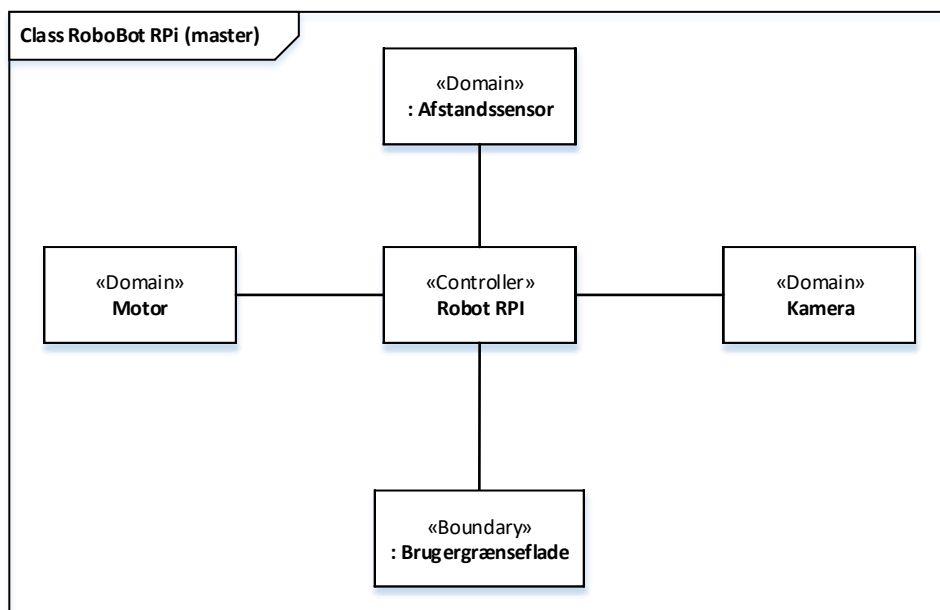
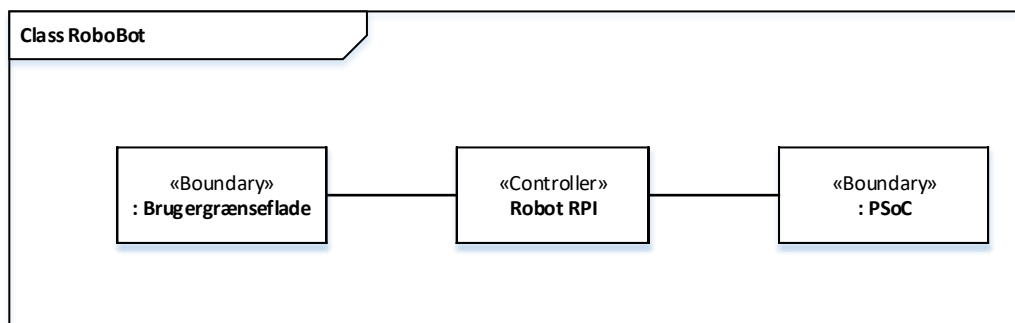
## SW arkitektur

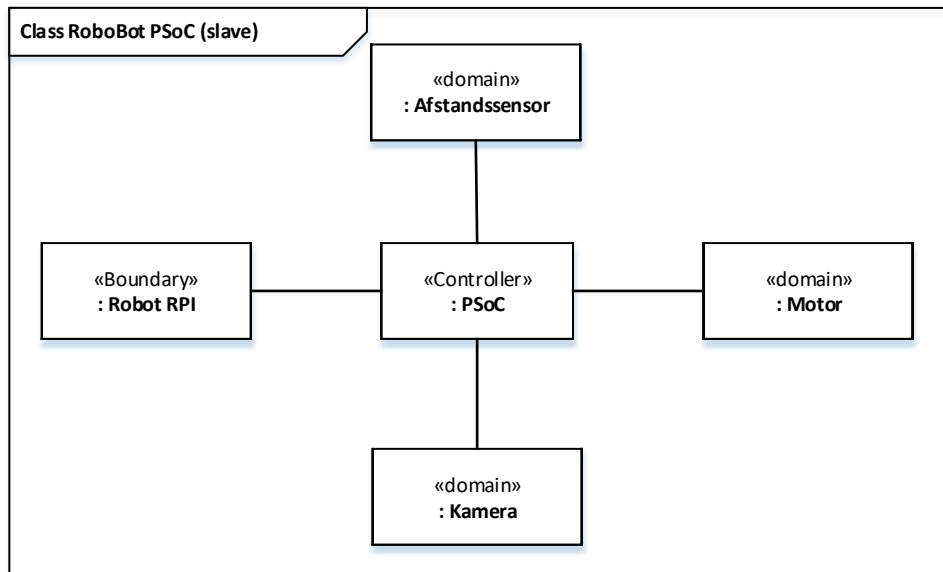
- Domænemodel
- Klassediagrammer
- SD for UC 1
- SD for UC 2
- SD for UC 3
- UML diagram for hele systemet

## Domænemodel



## Klassediagrammer

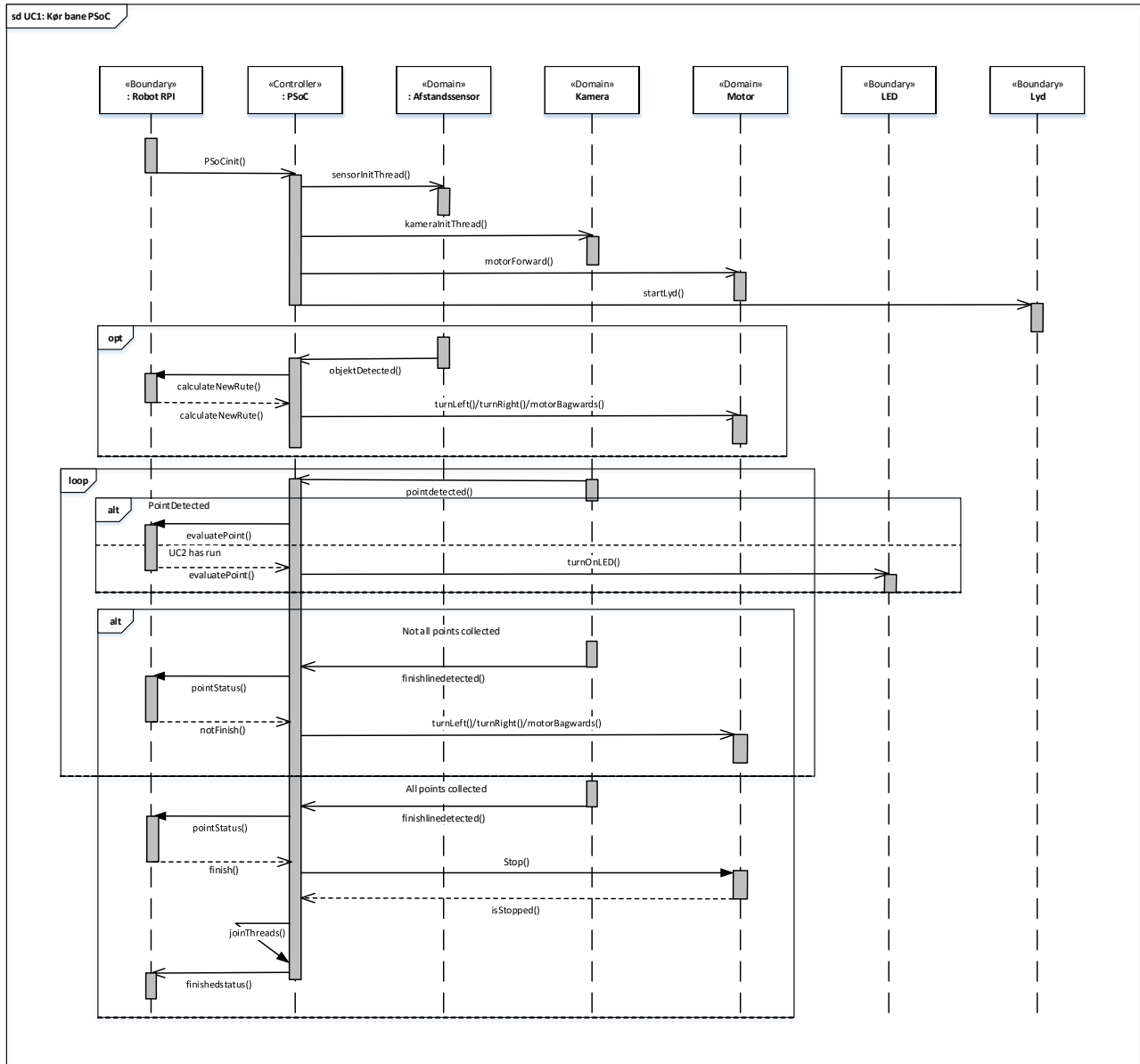




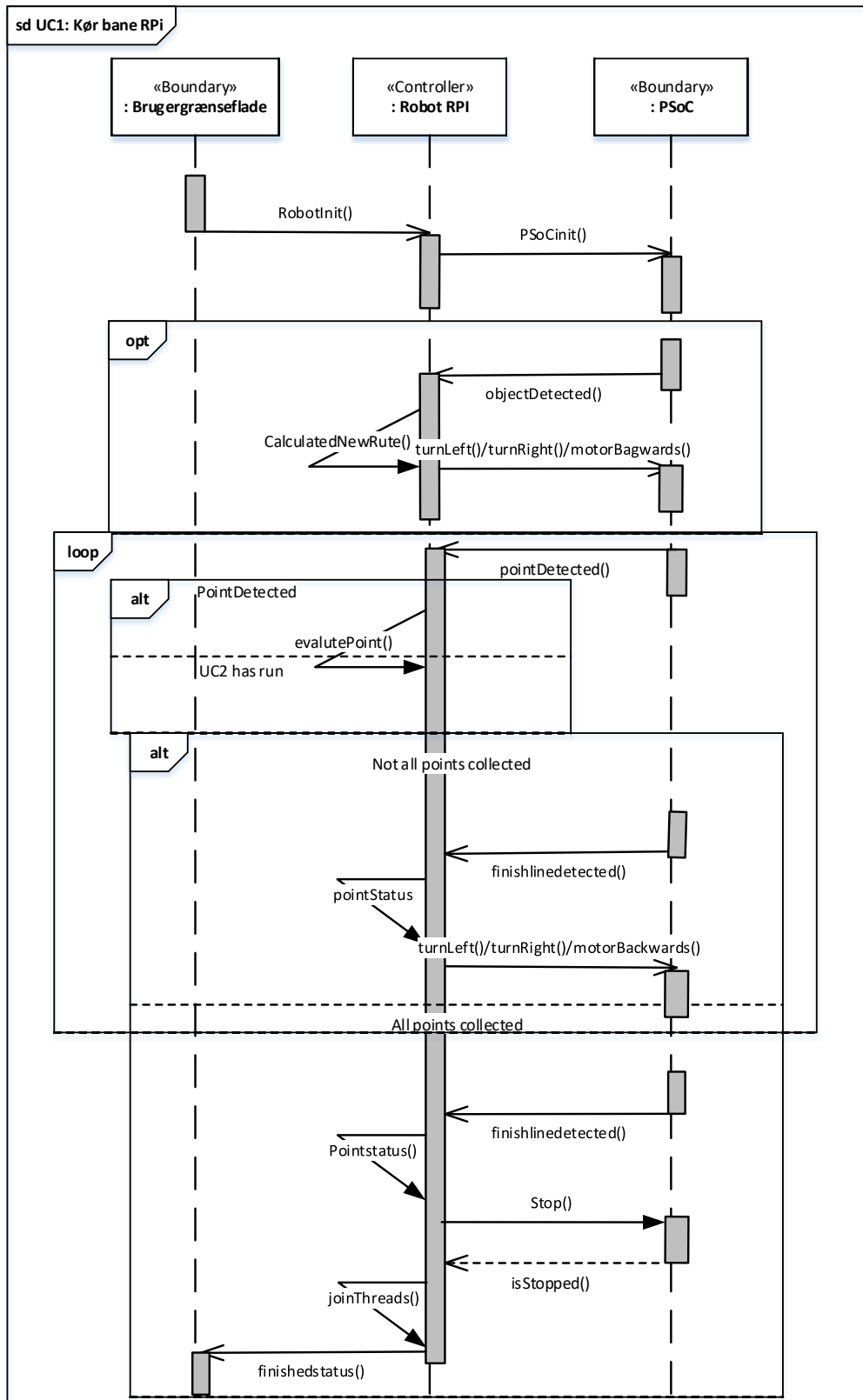
## SD for UC1

SD for PSoC (slave)

Der skal zoomes ind på diagrammet



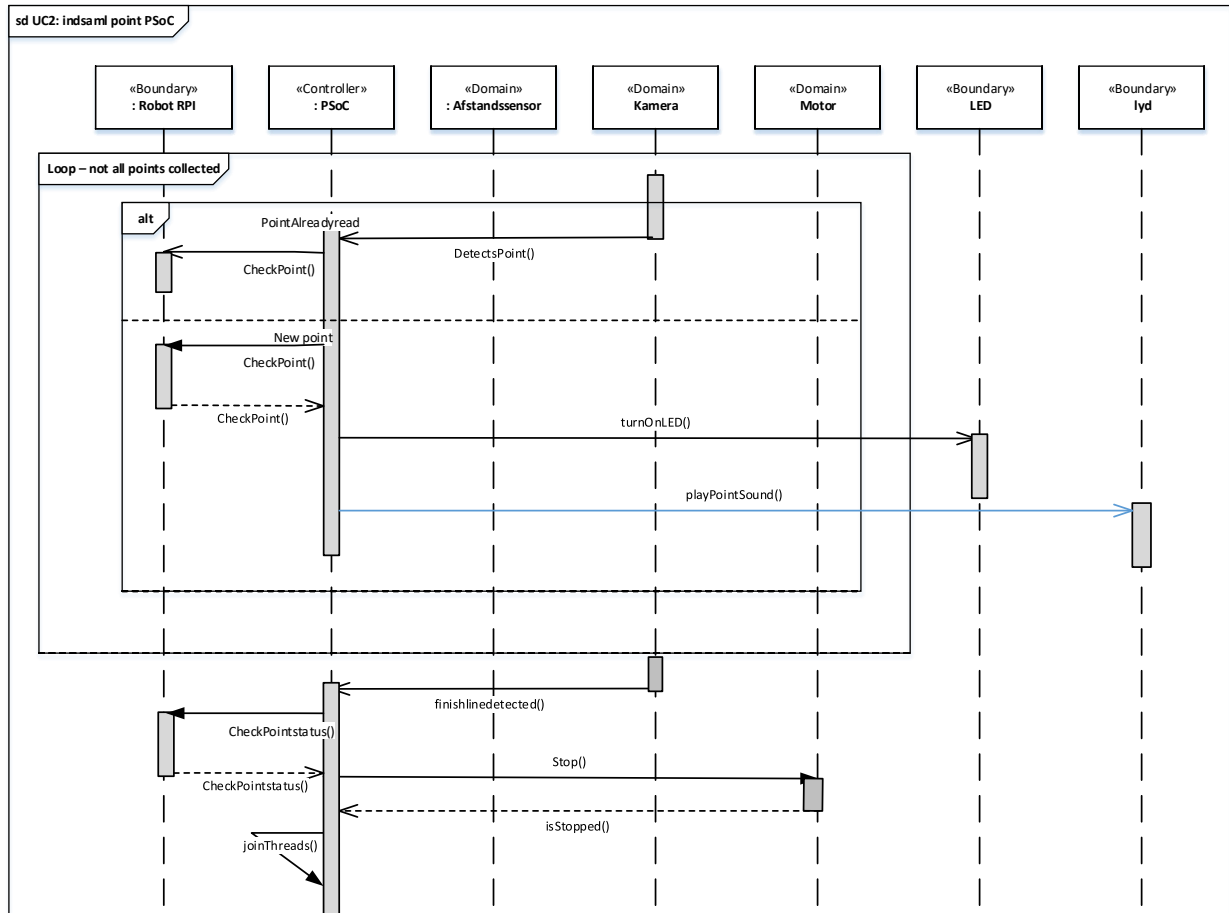
## SD for RPi (master)



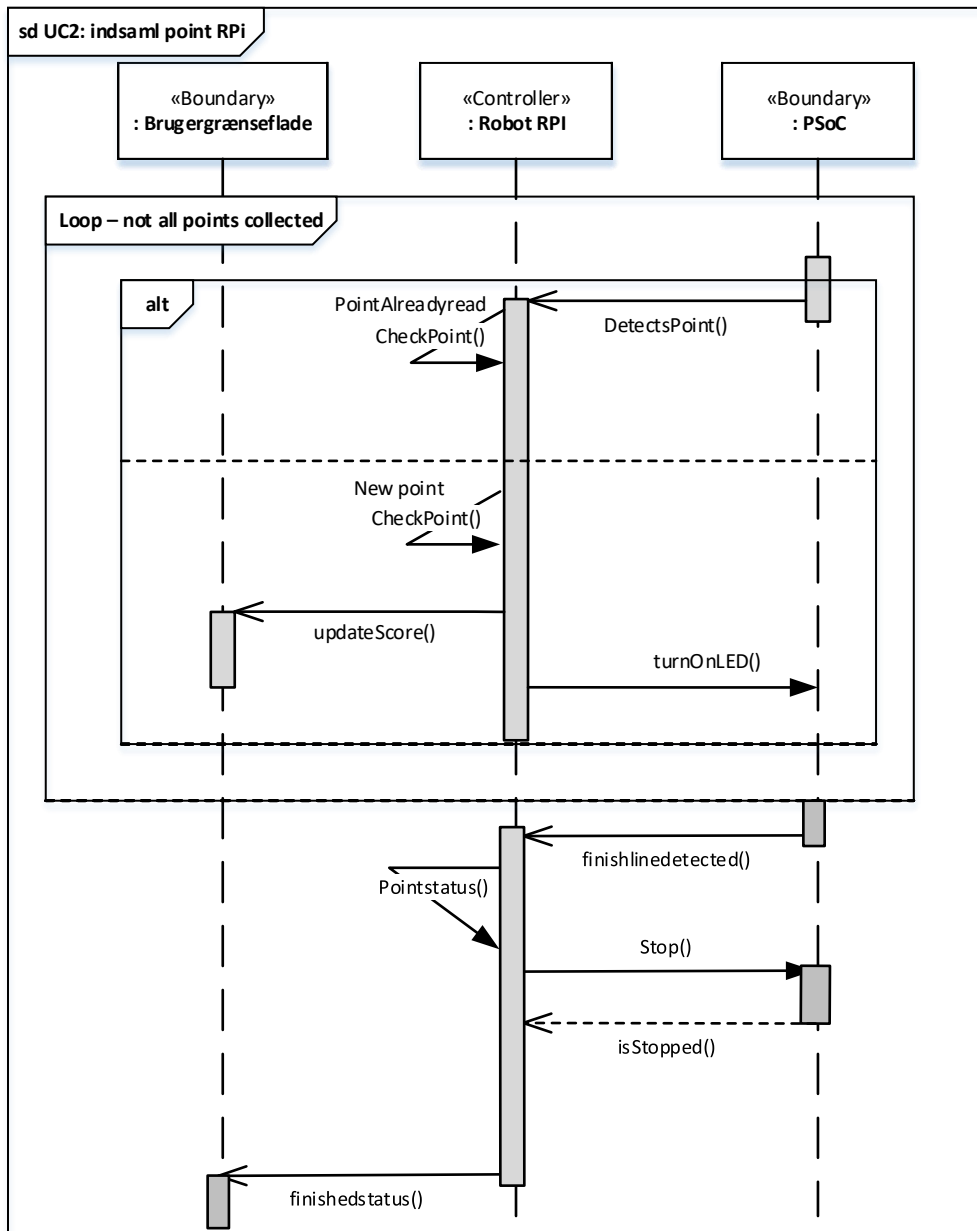
SD for UC2

SD for PSoC (slave)

Der skal zoomes ind på diagrammet



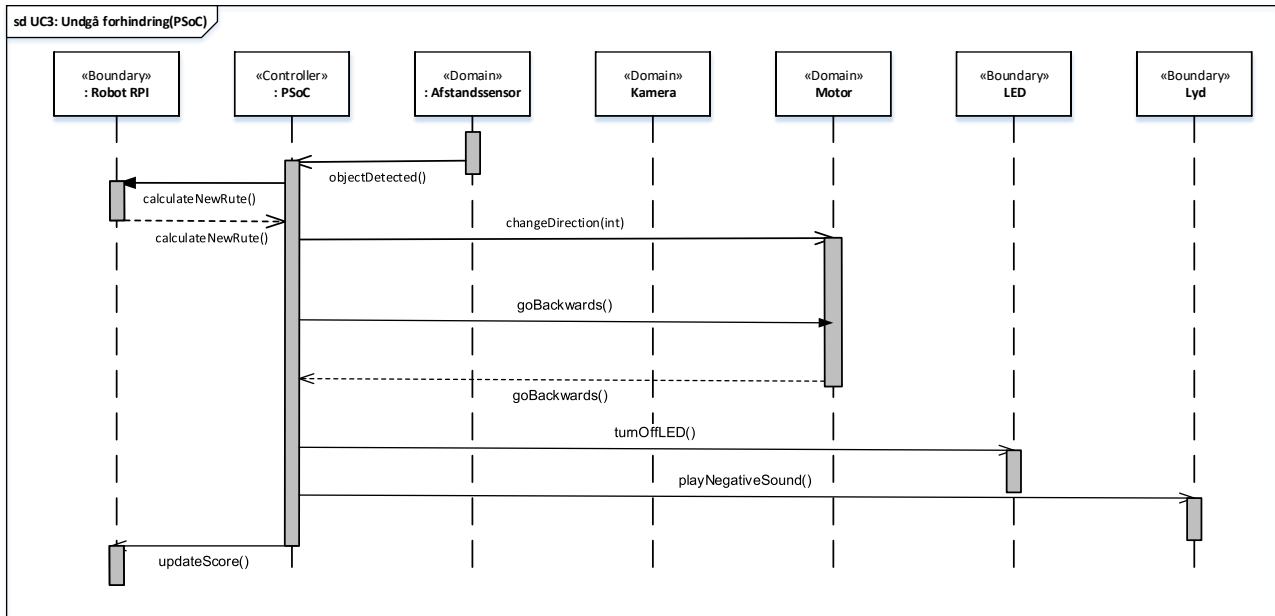
## SD for RPi (master)



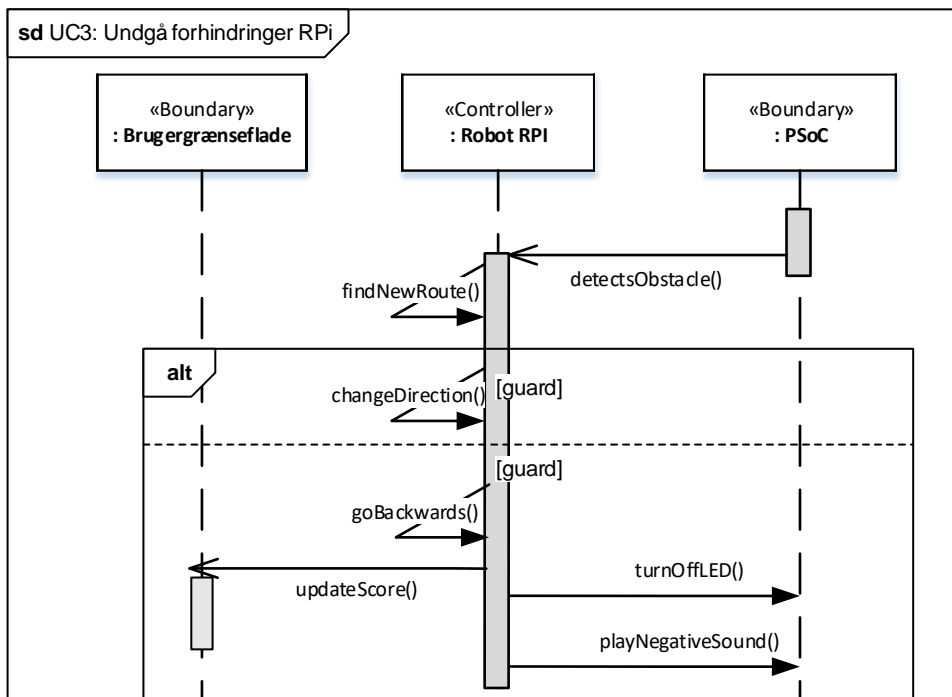
## SD for UC3

### SD for PSoC (slave)

Der skal zoomes ind på diagrammet



### SD for RPi (master)





## UML for hele systemet, udkast

