

Projektdokumentation  
AU2  
*Den intelligente bil*  
Gruppe 1

4. Semesterprojekt E4PRJ4  
Ingeniørhøjskolen, Aarhus Universitet  
Vejleder: Arne Justesen

26. oktober 2015

Navn	Studienummer	Underskrift
Kristian Thomsen	201311478	
Philip Krogh-Pedersen	201311473	
Lasse Barner Sivertsen	201371048	
Henrik Bagger Jensen	201304157	
Kenn Hedegaard Eskildsen	201370904	
Karsten Schou Nielsen	201370045	
Jesper Pedersen	201370530	



# Indhold

<b>Indhold</b>	<b>iii</b>
<b>1 Projektformulering</b>	<b>1</b>
1.1 Problemformulering .....	2
1.2 Projektbeskrivelse .....	2
1.3 Ordforklaring .....	3
<b>2 Kravspecifikation</b>	<b>5</b>
2.1 Systemoversigt .....	6
2.2 Aktør-kontekstdiagram .....	9
2.3 Aktørbeskrivelser .....	9
2.4 Funktionelle krav .....	10
2.5 Ikke-funktionelle krav .....	10
2.6 Use Cases .....	11
2.6.1 Use Case beskrivelser - Initiering og Formål .....	12
2.6.2 Fully Dressed Use Cases .....	14
<b>3 Systemarkitektur</b>	<b>27</b>
3.1 Indledning .....	27
3.2 BDD for AU2 .....	27
3.3 Bil .....	28
3.3.1 Diagrammer for bil .....	28
3.3.2 Fremdrift .....	33
3.3.3 Styretøj .....	35
3.3.4 Pi .....	36
3.3.5 Sensorer .....	38
3.3.6 MPU-6050 Accelerometer/Gyroskop .....	39
3.4 PC .....	41
3.4.1 BDD for PC .....	41
3.4.2 IBD for PC .....	42
3.4.3 Signalbeskrivelse for PC .....	42
3.5 Protokolbeskrivelse .....	44
3.5.1 Kamera .....	44
3.5.2 GUI .....	44
<b>4 Accepttest</b>	<b>47</b>
4.1 Funktionelle Krav .....	48
4.2 Ikke-funktionelle krav .....	62



# 1 Projektformulering

## Version

Dato	Version	Initialer	Ændring
29. september	1	Alle	Første udkast.
26. oktober	2	PKP, KT og JEP	Mindre rettelser efter review
	3		
	4		

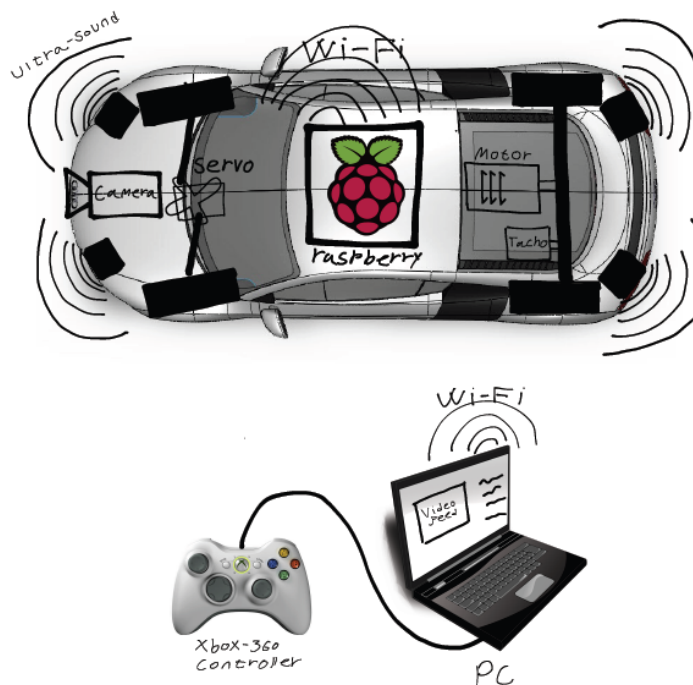
## 1.1 Problemformulering

Ifølge Niklas Alexander Chimirri, forsker inden for områder som barndom, psykologi og teknologi ved Roskilde Universitet, er leg en vigtig del af børns opvækst. Det er essentielt for deres fremtid da det gør børnene sociale, robuste, kreative og ikke mindst nysgerrige. Med til at skabe rammerne for børns leg er legetøj, og i dag er det vigtigt at børn har mulighed for at anvende den teknologi der er til rådighed i dagens Danmark. Dette bekræftes i en artikel der er udgivet på Roskilde Universitets hjemmeside i november 2014. Han konkluderer at der er for stor forskel imellem den virkelighed børnene møder i, og uden for børnehaven ift. den teknologi der i dag er til rådighed.

## 1.2 Projektbeskrivelse

Projektet skal bidrage til eller i det mindste sætte fokus på, at det er vigtigt at børn har muligheden for at lege... og gerne med moderne teknologi. Derfor omhandler projektet design og implementering af en fjernstyret bil. Det skal ikke være en almindelig fjernstyret bil - den skal være intelligent og den får navnet "AU2". En skitse af bilen er vist på figur 1.

Den intelligente del består af sensorer samt en kommunikationsenheder, som gør det muligt at styre bilen over et trådløst netværk. Brugeren har hermed mulighed for at navigere bilen ved at betragte en computerskærm, der viser et live-stream med video fra et kamera monteret på bilen. Er bilen inden for synsfeltet kan den selvfølgelig også styres ved at se direkte på den. For at undvige forhindringer på kørebanen, implementeres et anti-kollisionssystem bestående af afstandssensorer på bilen, placeret sådan at de kan detektere om bilen nærmer sig en forhindring. Således kan bilen selv kan standse eller undvige, hvis den nærmer sig en forhindring hastigt. Anti-kollisionssystemet har til formål at forhindre en evt. kollision og derved beskadigelse af bilen eller dens omgivelser.



Figur 1: Rigt billede af systemet i sin helhed

## 1.3 Ordforklaring

### System

Det totale system indeholder bil, software på PC og kommunikation mellem Bil og PC.

### HID (Human Interface Device)

Et interface som en bruger anvender til at interagere med en computer fx. tastatur og mus. I dette projekt anvendes desuden en Xbox-360 controller, med følgende funktionalitet:

- Right Trigger (RT)
- Left Trigger (LT)
- Flere knapper her.

### Hovedvindue

Hovedvinduet i software på PC indeholder videostream, status på bilen samt muligheder for at konfigurere og kalibrere systemet.

### Bil

Med bil menes den hardware der fysisk er placeret på bilen, dette være sig bla. bilens controllerenhed, her et Raspberry Pi 2 board, afstandssensorer, tachometer samt accelerometer.

### Pi (Raspberry Pi 2 B)

En Raspberry Pi er en single board computer i kreditkortstørrelse. Den anvendes i dette system som en controller til at styre bilen med.

### Wi-Fi netværk

Trådløst netværk af standarden "IEEE 802.11", som Bil og PC kommunikerer over. Dette netværk sættes op lokalt til brug udelukkende for kommunikationen imellem Bil og PC.

### AKS (Anti-kollisionssystem)

Et system på bilen bestående af fire afstandssensorer, samt signalbehandling- og reguleringssoftware som er i stand til at forhindre en kollision ved at overtage styring fra Bruger i tilfælde af forestående kollision. Der differentieres mellem "Undvig forhindring" og "Tænd/Sluk AKS".

- Tænd/Sluk bruges i forbindelse med at koble AKS til eller fra, således at bilen ikke vil undgå en kollision hvis AKS er slukket, men vil undgå en kollision hvis AKS er tændt.
- Undvig forhindring bruges i forbindelse med en forestående kollision. Her overtager AKS styring af bilen indtil forhindringen er undvejet.

**Afstandssensorer**

Afstandssensorerne er de 4 ultralydssensorer der er påmonteret bilen. Disse kan herefter benævnes som følgende:

- Front Left (FL)
- Front Right (FR)
- Rear Left (RL)
- Rear Right (RR)

**UC (Use Case)**

En use case er en standard for et brugsmønster til at afdække funktionalitet for et system.



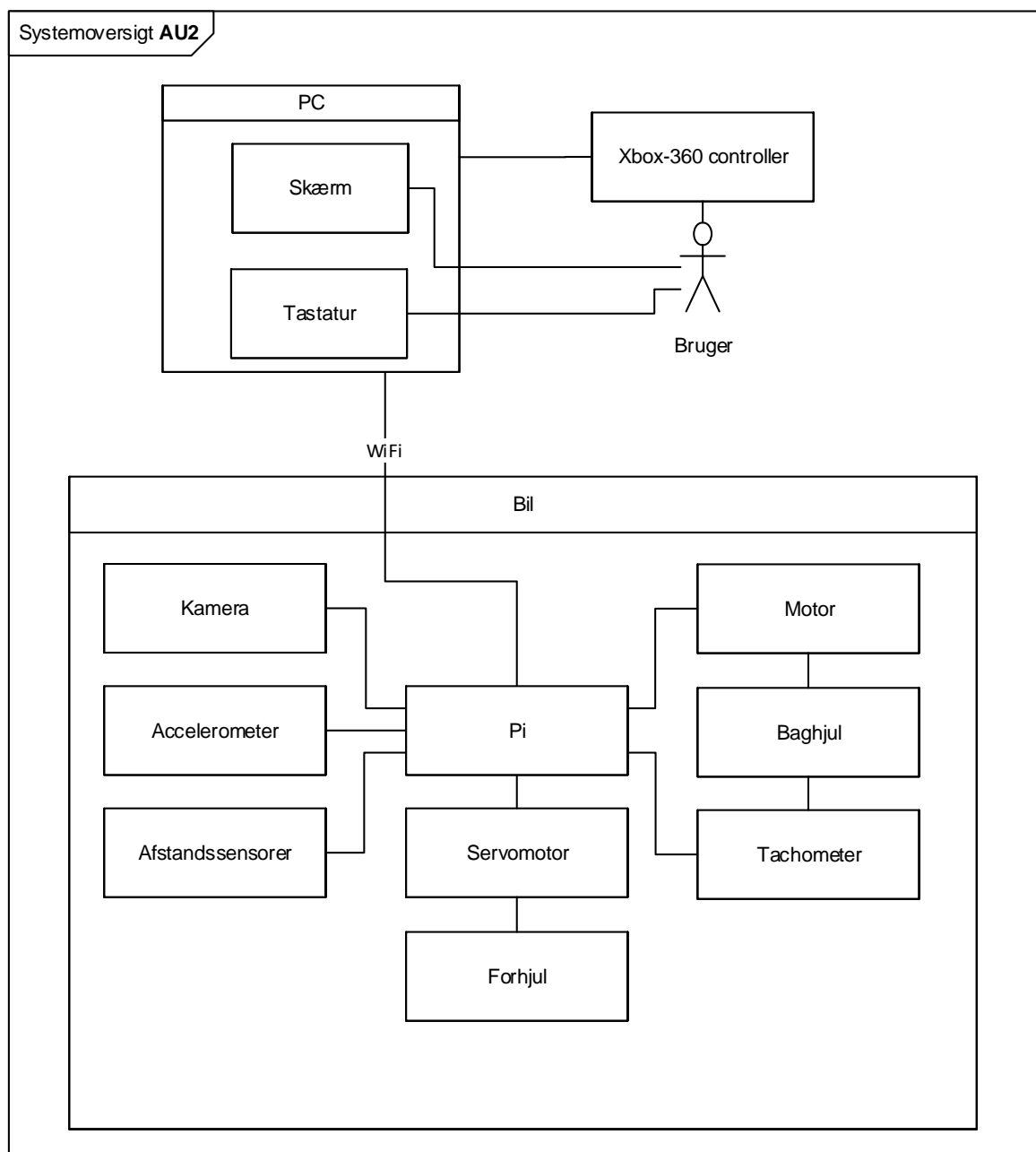
## 2 Kravspecifikation

### Version

Dato	Version	Initialer	Ændring
29. september	1	Alle	Første udkast.
26. oktober	2	PKP, KT og JEP	Mindre rettelser efter review
	3		
	4		

## 2.1 Systemoversigt

På figur 2 ses den overordnede systemoversigt med kommunikationsveje og mekaniske forbindelser. Diagrammet skal give læseren et hurtigt overblik over det samlede system. I afsnittet beskrives blokke og kommunikationsveje mere detaljeret. Under figur 2 er blokkene kort beskrevet.



Figur 2: Overordnet systemoversigt

**Pi**

Systemets kerne er et Raspberry Pi 2 board. Pi'en står for at processere data fra afstandssensorene, og håndtere streaming af video. Derudover afvikles regulering til motor, samt styring af servo også fra Pi'en.

**Servomotor**

Servomotor har til opgave at omsætte signal fra Pi'en til mekanisk styring af bilens forhjul.

**Afstandssensor**

Bilens 2 fremadrettet og 2 bagudrettet afstandssensorer har til formål at indsamle data om eventuelle forhindringer i bilen kørebane.

**Accelerometer**

Der er påmonteret et accelerometer der anvendes til regulering af hastighed.

**Kamera**

Bilens kamera streamer video til PC'ens skærm så Bruger har mulighed for at navigere på baggrund af visuel feedback

**PC**

PC afvikler den software hvorigennem bilen kontrolleres, konfigureres og kalibreres. Det er ligeledes via computeren at Bruger får visuel feedback fra bilens kamera.

**Xbox-360 Controller**

Til at kontrollere bilen, benyttes en Xbox-360 controller. vha. en række trykknapper og styrepinde kan bilens hastighed, såvel som retning bestemmes.

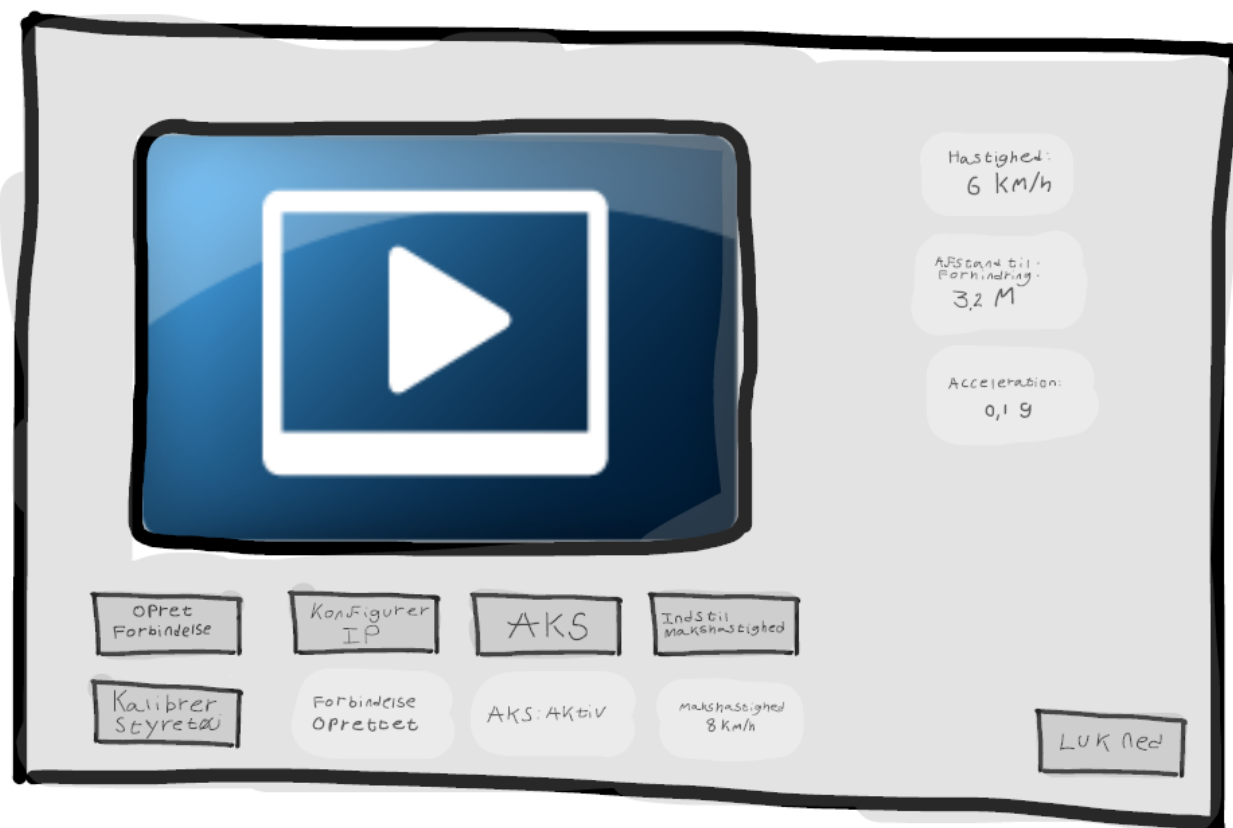
**Motor**

Motoren omsætter data, herunder regulering fra Pi'en til mekanisk styring af bilens hastighed.

**Tachometer**

Motorens omdrejningshastighed kan via tachometeret aflæses og herefter benyttes til databehandling og regulering.

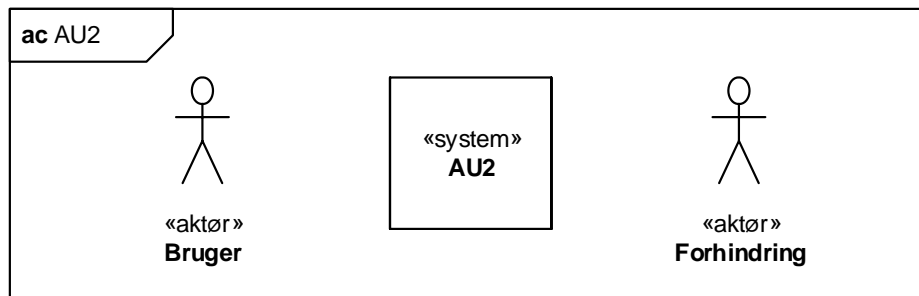
I figur 3 vises en skitse af hovedmenuen i softwaren på PC.



Figur 3: Skitse af hovedmenu

## 2.2 Aktør-kontekstdiagram

På figur 4 ses aktørkontekstdiagram over systemet.



Figur 4: Aktør kontekst diagram for AU2.

## 2.3 Aktørbeskrivelser

Som figur 4 viser, er der 2 aktører til systemet. *Bruger* og *Forhindring*.

### Bruger - Primær Aktør

Brugeren vil typisk være et barn med alder over 8 år, men kan også være en voksen med interesse for fjernstyrede biler.

Bruger kan:

- Starte og stoppe systemet
- Styre bilen over et Wi-Fi netværk.
- Konfigurere og kalibrere system.

### Forhindring - Sekundær Aktør

*Forhindring* er objekter i det miljø bilen kører i, og som der dermed er risiko for at bilen kan kollideres med.

## 2.4 Funktionelle krav

Ambitionen for dette projekt er som absolut minimum at realisere nedenstående punkter under ”skal”. Det forventes desuden at punkterne under ”bør” realiseres, men de har lavere prioritet. Punkterne under ”kan” forventes ikke realiseret, og punkterne under ”vil ikke...” realiseres med sikkerhed ikke. Sidstnævnte punkter kan ses som udviklingsmuligheder i forhold til senere versioner af systemet.

Systemet...

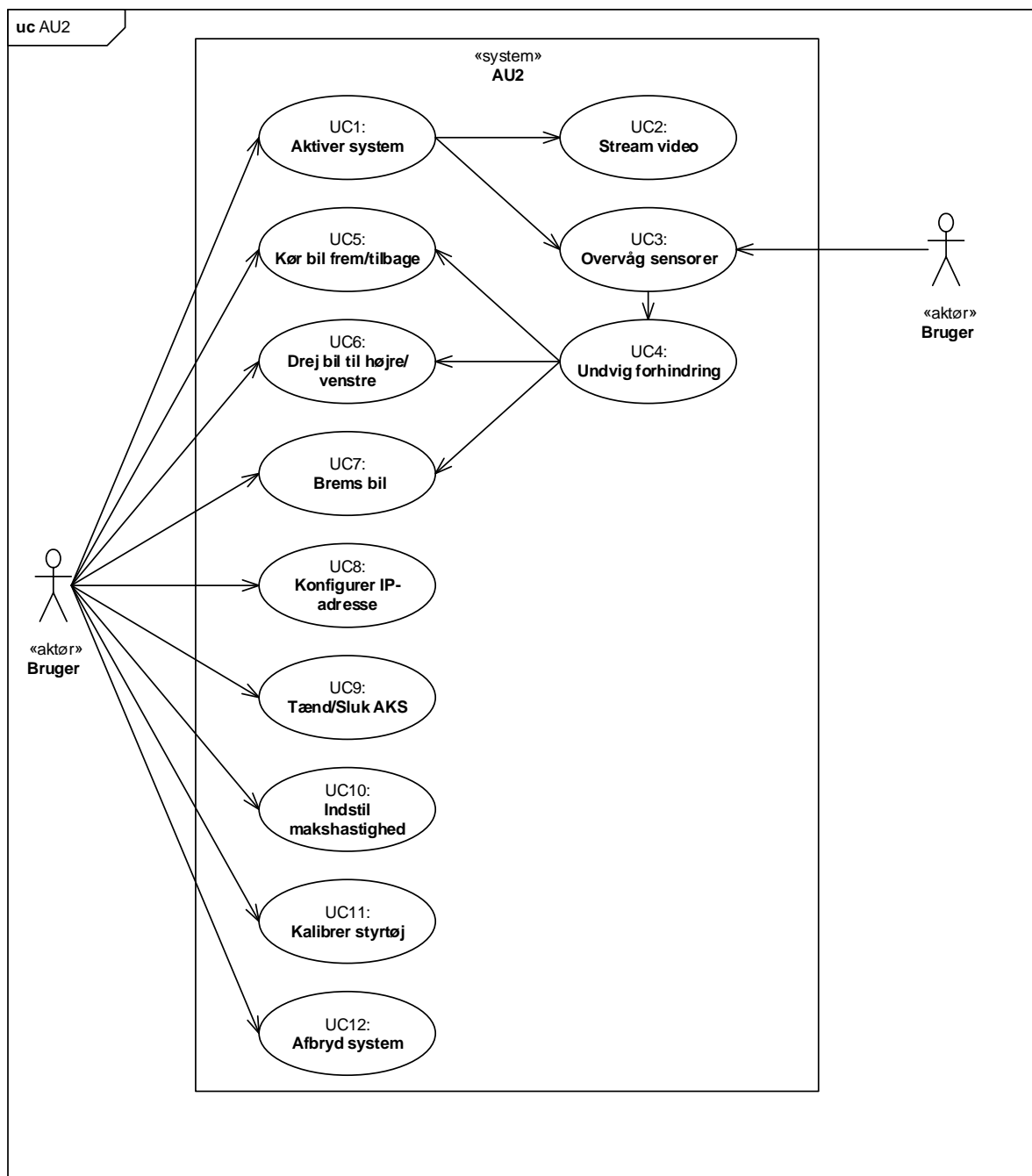
1. ... *Skal* kunne køre frem og tilbage.
2. ... *Skal* kunne dreje.
3. ... *Skal* kunne regulere hastigheden på bilen.
4. ... *Skal* give Bruger mulighed for at begrænse maksimumshastighed.
5. ... *Skal* give Bruger mulighed for manuel styring via Xbox-360 controller af hastighed og retning.
6. ... *Skal* via Wi-Fi netværk kunne kommunikere mellem bil og PC.
7. ... *Skal* kunne identificere forhindringer foran og bag bilen.
8. ... *Skal* indeholde et anti-kollisionssystem baseret på afstandssensorer.
9. ... *Skal* via. anti-kollisionssystem kunne undvige og/eller stoppe før kollision.
10. ... *Skal* indeholde et kamera til at streame video.
11. ... *Bør* give Bruger mulighed for at aktivere/deaktivere anti-kollisionssystemet på bilen.
12. ... *Bør* have bremselys, som aktiveres når bilen bremser.

## 2.5 Ikke-funktionelle krav

1. Bilens maksimumshastighed uden begrænsning er  $10km/t \pm 1km/t$
2. Bilens bremselængde ved maksimumshastighed uden begrænsning må ikke overstige 1 m.
3. Bilen skal kunne accelerere fra  $0km/t$  til maksimumshastighed uden begrænsning på højst 6 s.
4. Forsinkelse fra brugerinput til at bilen reagerer må ikke overstige 50ms.
5. Afstandssensorerne skal kunne identificere en forhindring i form af et kvadrat med en sidelængde på  $S = \sqrt{K \times L}$ , hvor  $K = 0.015m$  og  $L$  er afstanden og det gælder at  $0.20m < L < 6.00m$ . Kvadratet skal være vinkelret på bilen, således at fladen på kvadratet vender direkte mod bilen. På afstande over  $6m$  er det ikke et krav at systemet kan detektere forhindringen.
6. Mister bilen forbindelsen med PC i mere end  $50ms$ , standser bilen automatisk.
7. Kameraet skal minimum have en opdateringshastighed på 15 billeder i sekundet.
8. Systemet skal vise video-stream med en opløsning på  $640 \times 480$  pixels i hovedvinduet.
9. PC skal som minimum sende kommandoer til bilen 60 gange i sekundet.
10. HID skal bestå af en Xbox-360 controller, tastatur og mus.

## 2.6 Use Cases

På figur 5 ses use case diagram over de funktionelle krav.



Figur 5: Use case diagram for AU2.

### 2.6.1 Use Case beskrivelser - Initiering og Formål

#### UC1: Aktiver system

Initieres af: Bruger

Denne UC giver Bruger mulighed for at aktivere systemet. Bruger åbner software på PC, og sætter bilens "ON/OFF"-knap til "ON" for at tilslutte batteriet. Herefter konfigureres bilen, UC2 + UC3 initieres og PC'en viser hovedvinduet.

#### UC2: Stream Video

Initieres af: UC1: Aktiver system

Denne UC initierer videostream fra kameraet, og forbindelsen over Wi-Fi netværket oprettes.

#### UC3: Overvåg sensorer

Initieres af: UC1: Aktiver system

Denne UC initierer overvågning af bilens sensorer, herunder, de 4 afstandssensorer, tachometer, samt accelerometer. Use casen kører kontinuerligt og henter løbende data fra sensorerne.

#### UC4: Undvig forhindring

Initieres af: UC3: Overvåg sensorer

Denne UC har til formål at lade AKS overtage styring af bilen under kørsel hvis en forhindring detekteres enten foran eller bagved bilen. Når forhindringen er undvejet overgives styringen igen til Bruger.

#### UC5: Kør bil frem/tilbage

Initieres af: Bruger

Denne UC har til formål at give Bruger mulighed for at ændre hastighed på bilen via de trykfølsomme "LT" og "RT"-knapper på Xbox-360 controlleren. Bruger trykker på "LT" og bilen kører fremad, eller Bruger trykker på "RT" og bilen bakker.

#### UC6: Drej bil til højre/venstre

Initieres af: Bruger

Denne UC har til formål at lade Bruger ændre bilens retning. Bruger benytter venstre styrepind på Xbox-360 controlleren. Føres styrepinden til venstre, drejer bilens forhjul til venstre. Føres styrepinden til højre, drejer bilens forhjul til højre. Det har ingen betydning hvis styrepinden samtidig føres lidt opad eller nedad.

#### UC7: Brems bil

Initieres af: Bruger

Denne UC har til formål at lade Bruger sænke bilens hastighed. Bruger trykker "X" på Xbox-360 controlleren, jo længere tid knappen holdes nede jo mere sænkes bilens hastighed. Deaccelerationen er konstant.



**UC8: Konfigurer IP-adresse**

Initieres af: Bruger

Denne UC har til formål at lade Bruger konfigurere PC'ens IP-adresse således at der kan opnås forbindelse til bilen.

**UC9: Tænd/sluk AKS**

Initieres af: Bruger

Denne UC har til formål at give Bruger mulighed for at vælge om AKS skal være tændt eller slukket. Bruger kan via "Hovedvindue" på PC'en vælge status for AKS.

**UC10: Indstil makshastighed**

Initieres af: Bruger

Denne UC har til formål at give Bruger mulighed for at indstille en maksimumhastighed på bilen. Hastigheden indstilles via PC'ens "Hovedvindue".

**UC11: Kalibrer styretøj**

Initieres af: Bruger

Denne UC har til formål at give Bruger mulighed for at kalibrere bilens styretøj, så den kører ligeud når styrepinden ikke påvirkes. Bruger indtaster via menuen "Kalibrer styretøj" en værdi der angiver center for styretøjet.

**UC12: Afbryd system**

Initieres af: Bruger

Denne UC har til formål at lade Bruger afbryde hele systemet. Bruger afslutter software på PC, og sætte bilen "ON/OFF"-knap til "OFF" for at afbryde forbindelse til batteriet.

## 2.6.2 Fully Dressed Use Cases

### Use Case 1: Aktiver system

<b>Navn:</b>	UC1: Aktiver system
<b>Mål:</b>	At aktivere system
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	UC2: Stream video, UC3: Overvåg sensor, UC8: Konfigurer system
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	Netværksforbindelse er opsat og fungerende
<b>Resultat:</b>	Bil er initieret, PC viser Hovedvindue, UC2 og UC3 er initieret
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bilens "ON/OFF"-switch sættes til "ON".</li> <li>2. Bruger starter software på PC.</li> <li>3. Hovedvindue fremkommer på skærmen.</li> <li>4. Bruger trykker på "Opret forbindelse"</li> <li>5. PC opretter forbindelse til bilen. <ul style="list-style-type: none"> <li>• [Ext 5.a : Forbindelse kan ikke oprettes]</li> </ul> </li> <li>6. UC2: Stream video initieres af System. <ul style="list-style-type: none"> <li>• [Ext 6.a : Initiering af UC2 fejler]</li> </ul> </li> <li>7. UC3: Overvåg sensorer initieres af System. <ul style="list-style-type: none"> <li>• [Ext 7.a : Initiering af UC3 fejler]</li> </ul> </li> <li>8. PC prompter "Forbindelse oprettet"</li> <li>9. UC1 afsluttes</li> </ol>
<b>Udvidelser:</b>	<p><b>[Ext 5.a : Forbindelse kan ikke oprettes]</b></p> <ol style="list-style-type: none"> <li>1. System prompter "Forbindelse kan ikke oprettes".</li> <li>2. UC8: Konfigurer IP.</li> <li>3. Systemet fortsætter fra punkt 3 i hovedscenariet.</li> </ol> <p><b>[Ext 6.a : Initiering af UC2 fejler]</b></p> <ol style="list-style-type: none"> <li>1. System prompter "Videostream kan ikke oprettes".</li> <li>2. UC1 fortsætter fra punkt 3.</li> </ol> <p><b>[Ext 7.a : Initiering af UC3 fejler]</b></p> <ol style="list-style-type: none"> <li>1. System prompter "Initiering af sensorer fejlet".</li> <li>2. UC1 fortsætter fra punkt 3.</li> </ol>

Tabel 1: UC1: Aktiver system

**Use Case 2: Stream Video**

<b>Navn:</b>	UC2: Stream video
<b>Mål:</b>	At starte videostreamen
<b>Initiering:</b>	UC1: Aktiver system
<b>Aktører:</b>	Ingen
<b>Reference:</b>	UC1
<b>Antal samtidige forekomster:</b>	Én
<b>Forudsætning:</b>	UC1 frem til punkt 6 er fuldført
<b>Resultat:</b>	Videostream er initieret og kørende
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"><li>1. Bilen initierer kameraet.<ul style="list-style-type: none"><li>• [Ext 1.a: Initiering af kamera fejler]</li></ul></li><li>2. Bilen streamer video fra kamera til PC via Wi-Fi netværket.</li></ol>
<b>Udvidelser:</b>	<b>[Ext 1.a : Initiering af kamera fejler]</b> <ol style="list-style-type: none"><li>1. System prompter PC med "Kamera-initiering fejlet".</li><li>2. UC2 afsluttes.</li></ol>

Tabel 2: UC2: Stream video

**Use Case 3: Overvåg sensor**

<b>Navn:</b>	UC3: Overvåg sensorer
<b>Mål:</b>	At overvåge sensorer
<b>Initiering:</b>	UC1: Aktiver system
<b>Aktører:</b>	Ingen
<b>Reference:</b>	UC1
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1 frem til punkt 7 er fuldført
<b>Resultat:</b>	Sensorer overvåges løbende
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bilen initierer tachometer <ul style="list-style-type: none"> <li>• [Ext 1.a: Initiering af tachometer fejler]</li> </ul> </li> <li>2. Bilen initierer accelerometer <ul style="list-style-type: none"> <li>• [Ext 2.a: Initiering af accelerometer fejler]</li> </ul> </li> <li>3. Bilen initierer afstandssensorer. <ul style="list-style-type: none"> <li>• [Ext 3.a: Initiering af afstandssensorer fejler]</li> </ul> </li> <li>4. Bilen overvåger sensorer.</li> <li>5. Sensordata sendes til softwaren på PC'en.</li> </ol>
<b>Udvidelser:</b>	<p><b>[Ext 1.a : Initiering af tachometer fejler]</b></p> <ol style="list-style-type: none"> <li>1. Systemet prompter PC med "tachometer-initiering fejlet".</li> <li>2. UC2 afsluttes.</li> </ol> <p><b>[Ext 2.a : Initiering af accelerometer fejler]</b></p> <ol style="list-style-type: none"> <li>1. Systemet prompter PC med "accelerometer-initiering fejlet".</li> <li>2. UC2 afsluttes.</li> </ol> <p><b>[Ext 3.a : Initiering af afstandssensorer fejler]</b></p> <ol style="list-style-type: none"> <li>1. Systemet prompter PC med "afstandssensor-initiering fejlet".</li> <li>2. UC2 afsluttes.</li> </ol>

Tabel 3: UC3: Overvåg sensorer

**Use Case 4: Undvig forhindring**

<b>Navn:</b>	UC4: Undvig forhindring
<b>Mål:</b>	At bilen undviger en evt. kollision med en forhindring.
<b>Initiering:</b>	UC3: Overvåg sensor
<b>Aktører:</b>	Forhindring
<b>Reference:</b>	UC3, UC6: Drej bil til højre/venstre, UC7: Brems bil
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1 er gennemført, UC3 er gennemført, bilen er på vej mod en forhindring.
<b>Resultat:</b>	UC5, UC6 og/eller UC7 gennemføres og UC3 fortsætter.
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bilen analyserer indsamlet data fra afstandssensorer, kører den fremad analyseres de forreste sensorer ditto bagud.</li> <li>2. AKS overtager styring fra Bruger midlertidigt.</li> <li>3. <ul style="list-style-type: none"> <li>• [ALT a: UC6: Drej bil til højre/venstre aktiveres, hvis en enkelt sensor registrerer en forhindring]</li> <li>• [ALT b: UC7: Brems bil aktiveres hvis begge sensorer registrer en forhindring]</li> </ul> </li> <li>4. Bilen giver igen styring tilbage til brugeren.</li> <li>5. UC4 afsluttes.</li> </ol>
<b>Udvidelser:</b>	

Tabel 4: UC4: Undvig forhindring

**Use Case 5: Kør bil frem/tilbage**

<b>Navn:</b>	UC5: Kør bil frem/tilbage
<b>Mål:</b>	At få bilen til at køre frem eller tilbage.
<b>Initiering:</b>	Bruger
<b>Reference:</b>	Ingen
<b>Antal samtidige forekomster:</b>	Én
<b>Forudsætning:</b>	UC1: Aktiver system er fuldført og systemet er operationelt.
<b>Resultat:</b>	Bilens hastighed er ændret.
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger ændrer position af RT på Xbox-360 controlleren. <ul style="list-style-type: none"> <li>• [Ext 1.a: Bruger ændrer position af LT.]</li> </ul> </li> <li>2. Controllerens input streames til bilen.</li> <li>3. Bilen ændrer fremadgående hastighed i henhold til brugerens input. Et hårdere tryk resulterer i en højere hastighed og et lettere tryk resulterer i en lavere hastighed.</li> <li>4. UC5 afsluttes.</li> </ol>
<b>Udvidelser:</b>	<p><b>[Ext 1.a : Bruger ændrer position af LT.]</b></p> <ol style="list-style-type: none"> <li>1. Controllerens input streames til bilen.</li> <li>2. Bilen ændrer bagudgående hastighed i henhold til brugerens input. Et hårdere tryk resulterer i en højere hastighed og et lettere tryk resulterer i en lavere hastighed.</li> <li>3. Systemet fortsætter fra punkt 4 i hovedscenariet.</li> </ol>

Tabel 5: UC5: Kør bil frem/tilbage

**Use Case 6: Drej bil til højre/venstre**

<b>Navn:</b>	UC6: Drej til højre/venstre
<b>Mål:</b>	At få bilen til at dreje mod højre eller venstre
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	UC3
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1: Aktiver system er fuldført og systemet er operationelt
<b>Resultat:</b>	Retningen på bilens forhjul er ændret
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger ændrer position på den venstre styrepind på xbox-360 controlleren. <ul style="list-style-type: none"> <li>• [Ext 1.a: AKS bliver anvendt.]</li> </ul> </li> <li>2. Controllerens input streames til bilen.</li> <li>3. Bilen behandler input fra Bruger, hvis styrepinden føres til venstre drejes forhjulene til venstre, hvis styrepinden føres til højre drejes forhjulene ligeledes til højre.</li> <li>4. UC6 afsluttes.</li> </ol>
<b>Udvidelser:</b>	<p><b>[Ext 1.a : AKS bliver anvendt.]</b></p> <ol style="list-style-type: none"> <li>1. Bilen analyserer input fra UC3.</li> <li>2. Bilen drejer til højre, hvis sensor FL registrerer en forhindrer, ditto venstre og FR.</li> <li>3. Bilen undviger forhindringen.</li> <li>4. Systemet fortsætter fra punkt 3 i hovedscenariet.</li> </ol>

Tabel 6: UC6: Drej til højre/venstre

## Use Case 7: Brems bil

<b>Navn:</b>	UC7: Brems bil
<b>Mål:</b>	At få bilen til at bremse
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	UC3
<b>Antal samtidige forekomster:</b>	Én
<b>Forudsætning:</b>	UC1: Aktiver system er fuldført og systemet er operationelt
<b>Resultat:</b>	hastigheden på bilen er sænket
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger trykker på "X" knappen på Xbox-360 controlleren. <ul style="list-style-type: none"> <li>• [Ext 1.a: AKS er anvendt.]</li> </ul> </li> <li>2. Controllerens input streames til bilen.</li> <li>3. Bilen tjekker input, hvis bremsekommando modtages sænker bilen hastigheden.</li> <li>4. UC7 afsluttes.</li> </ol>
<b>Udvidelser:</b>	<b>[Ext 1.a : AKS er anvendt]</b> <ol style="list-style-type: none"> <li>1. Systemet initierer UC4</li> <li>2. Systemet fortsætter fra punkt 3 i hovedscenariet</li> </ol>

Tabel 7: UC7: Brems bil



**Use Case 8: Konfigurer IP-adresse**

<b>Navn:</b>	UC8: Konfigurer IP-adresse
<b>Mål:</b>	At konfigurere bilens IP-adresse til PC'en
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	Ingen
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1: Aktiver system er udført til punkt 3, bilen og PC er på samme netværk, systemet viser "Hovedvindue" samt at systemet er operationelt
<b>Resultat:</b>	IP adressen på bilen er indstillet
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger trykker på "Konfigurer IP".</li> <li>2. Konfigurationssmenuen for IP-adressen vises, og der er mulighed for at indtaste en IP-adresse.</li> <li>3. Bruger indtaster bilens IP-adresse.</li> <li>4. Bruger trykker "Gem" og system viser "Hovedvindue".</li> <li>5. Bruger trykker på "Opret forbindelse".</li> <li>6. Hovedvindue viser "Forbindelse oprettet". <ul style="list-style-type: none"> <li>• [Ext 6.a Hovedvindue viser "Forbindelse ikke oprettet".]</li> </ul> </li> <li>7. UC8 afsluttet.</li> </ol>
<b>Udvidelser:</b>	[Ext 6.a Hovedvindue viser "Forbindelse ikke oprettet"] <ol style="list-style-type: none"> <li>1. Bruger gentager fra punkt 2 i hovedscenarie.</li> </ol>

Tabel 8: UC8: Konfigurer IP-adresse

**Use Case 9: Tænd/sluk AKS**

<b>Navn:</b>	UC9: Tænd/sluk AKS
<b>Mål:</b>	At tænde eller slukke for AKS på bilen
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	UC11: Kalibrer styretøj
<b>Antal samtidige forekomster:</b>	Én
<b>Forudsætning:</b>	UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt
<b>Resultat:</b>	
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger trykker på "AKS".</li> <li>2. System viser "AKS-menu".</li> <li>3. AKS-menu giver Bruger mulighed for at tænde/slukke for AKS .</li> <li>4. AKS-menu viser status for AKS.</li> <li>5. Bruger trykker tænd/sluk efter ønske.</li> <li>6. Bilen tænder/slukker for AKS systemet efter brugerens ønske.</li> <li>7. "Hovedvindue" indikerer nuværende status af AKS.</li> </ol>
<b>Udvidelser:</b>	

Tabel 9: UC9: Tænd/sluk AKS

**Use Case 10: Indstil makshastighed**

<b>Navn:</b>	UC10: Indstil makshastighed
<b>Mål:</b>	At konfigurere bilens makshastighed
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	UC8: Konfigurer IP-adresse
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt
<b>Resultat:</b>	
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger trykker på "Indstil makshastighed".</li> <li>2. Systemet præsenterer menu makshastighed med mulighed for indtastning af makshastighed fra 1-10 km/t.</li> <li>3. Menuen indikerer bilens nuværende makshastighed.</li> <li>4. Bruger indtaster bilens nye makshastighed.</li> <li>5. Bruger trykker på "Opdater".</li> <li>6. "Hovedvindue" viser den nye værdi som makshastighed. <ul style="list-style-type: none"> <li>• [Ext 1. "Hovedvindue" viser ikke den nye makshastighed]</li> </ul> </li> </ol>
<b>Udvidelser:</b>	[Ext 1. Menuen indikerer ikke den nye makshastighed] <ol style="list-style-type: none"> <li>1. Bruger går til UC8</li> </ol>

Tabel 10: UC10: Indstil makshastighed

## Use Case 11: Kalibrer styretøj

<b>Navn:</b>	UC11: Kalibrer styretøj
<b>Mål:</b>	At kalibrere systemet så bilen kører ligeud når brugeren slipper styrepinden på Xbox-360 controlleren
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	Ingen
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedmenu", at systemet er operationelt samt bilen holder stille
<b>Resultat:</b>	Bilens styretøj er kalibreret
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger vælger "Kalibrer styretøj".</li> <li>2. Systemet viser menu for Kalibrering med mulighed for indtastning af værdi mellem -50 og 50, hvor -50 svarer til fuldt udslag til venstre og 50 vil fuldt udslag til højre.</li> <li>3. Bruger indtaster værdi. <ul style="list-style-type: none"> <li>• [Ext 3.a : Bruger indtaster ugyldig værdi]</li> </ul> </li> <li>4. Bruger trykker på "Gem".</li> <li>5. Systemet gemmer værdien på bilen.</li> <li>6. Forhjulene drejer en absolut værdi mod enten, højre eller venstre: positiv værdi oversættes til højre, og negativ værdi oversættes venstre.</li> <li>7. Systemet returnerer til "Hovedvindue"</li> </ol>
<b>Udvidelser:</b>	<p>[Ext 3.a : Bruger indtaster en ugyldig værdi]</p> <ol style="list-style-type: none"> <li>1. Systemet prompter: "Ugyldig værdi, indtast en gyldig værdi."</li> <li>2. Systemet fortsætter fra punkt 2 i hovedscenariet.</li> </ol>

Tabel 11: UC11: Kalibrer styretøj

**Use Case 12: Afbryd system**

<b>Navn:</b>	UC12: Afbryd system
<b>Mål:</b>	At lukke systemet ned
<b>Initiering:</b>	Bruger
<b>Aktører:</b>	Bruger
<b>Reference:</b>	Ingen
<b>Antal samtidige forekomster:</b>	En
<b>Forudsætning:</b>	UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt
<b>Resultat:</b>	Systemet er lukket sikkert ned og forsyning til batteriet er afbrudt
<b>Hovedscenarie:</b>	<ol style="list-style-type: none"> <li>1. Bruger lukker ned for softwaren på PC'en. <ul style="list-style-type: none"> <li>• [Ext 1. Bilen er løbet tør før strøm]</li> </ul> </li> <li>2. Bruger skubber kontakten "ON/OFF" på undersiden af bilen til position "OFF".</li> <li>3. Strømmen til bilen afbrydes.</li> </ol>
<b>Udvidelser:</b>	[Ext 1. Bilen er løbet tør før strøm] <ol style="list-style-type: none"> <li>1. Systemet lukker korrekt ned og forbindelse afbrydes til softwaren på PC'en.</li> <li>2. Uce Case 12 fortsætter fra punkt 3 i hovedscenariet.</li> </ol>

Tabel 12: UC12: Afbryd system



## 3 Systemarkitektur

### Version

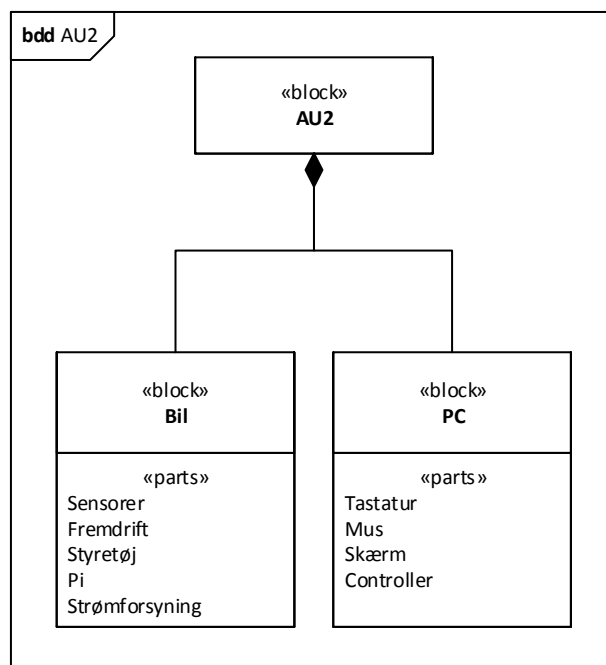
Dato	Version	Initialer	Ændring
26. oktober	2	Alle	Første udkast
	3		
	4		

### 3.1 Indledning

Følgende afsnit beskriver arkitekturen for hardware- og software delene af projektet. BDD og IBD er lavet mhp. forståelse og indblik i systemet, således alle grænseflader og interne dele af systemet bliver forklaret. Til hvert diagram vil der være en kort forklaring, som beskriver det yderligere.

### 3.2 BDD for AU2

På figur 6 ses det overordnede blokdiagram for AU2. Der vises tilhørsforhold og sammenhæng med det samlede system, som senere bliver uddybet mere.



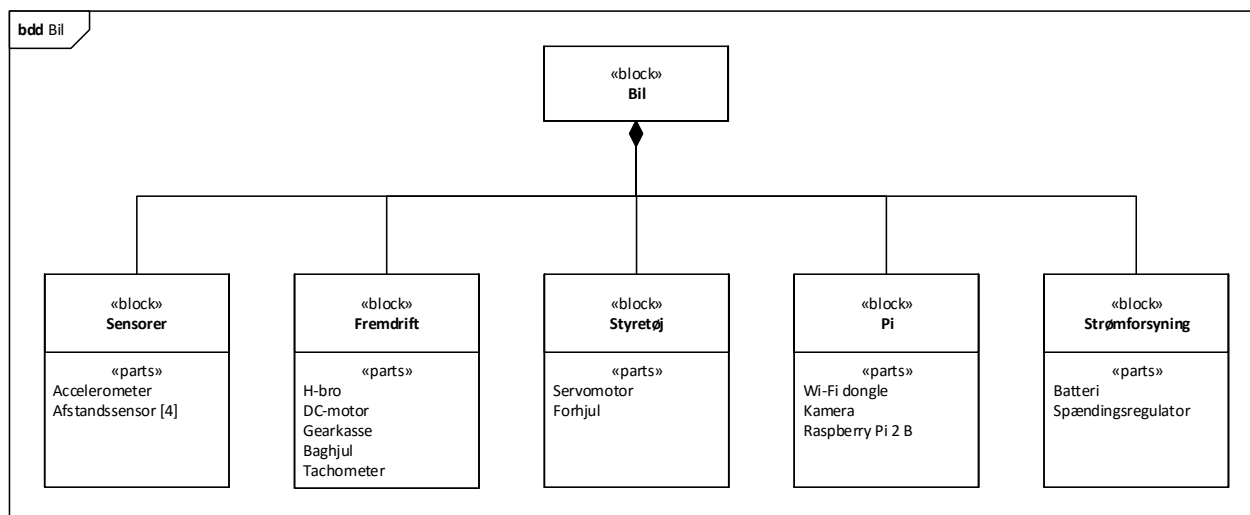
Figur 6: Overordnet BDD for AU2

## 3.3 Bil

### 3.3.1 Diagrammer for bil

#### BDD for bil

Dette diagram viser blokken bil fra figur 6. Blokken 'bil' skal forstås som alle mekaniske og elektriske som er fastgjort på køretøjet. Således er

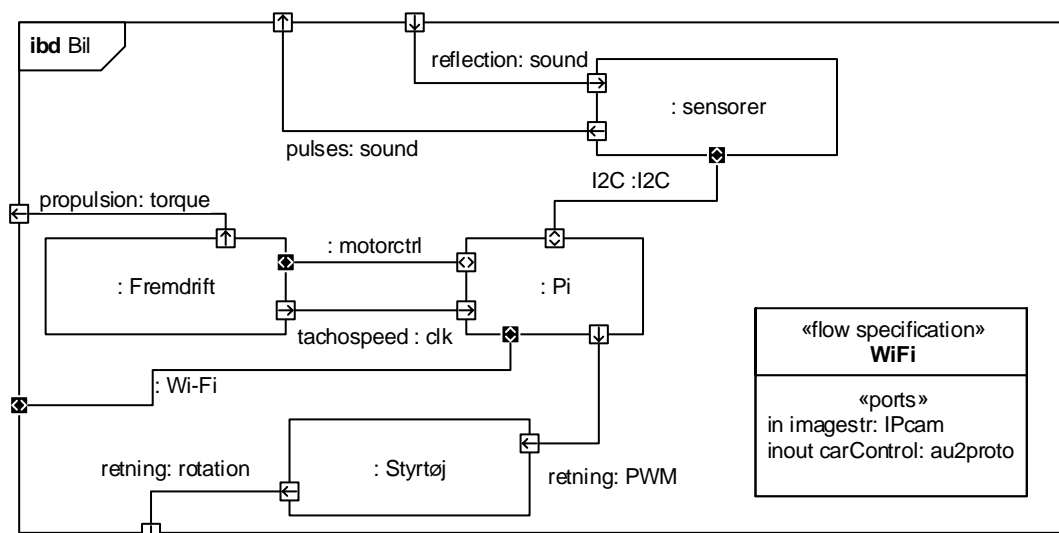


Figur 7: BDD for bil



### IBD for signaler i bil

På figur 8 ses de interne forbindelser for figur 7. Diagrammet skaber et overblik over hvilke signaler der sendes og modtages. Signalerne bliver beskrevet yderligere i signalbeskrivelsen på side ???. Bemærk at alle forsyningerne ikke er taget med på diagrammet, men istedet er lavet i et diagram for sig. Forsyningerne kan ses på figur 9.



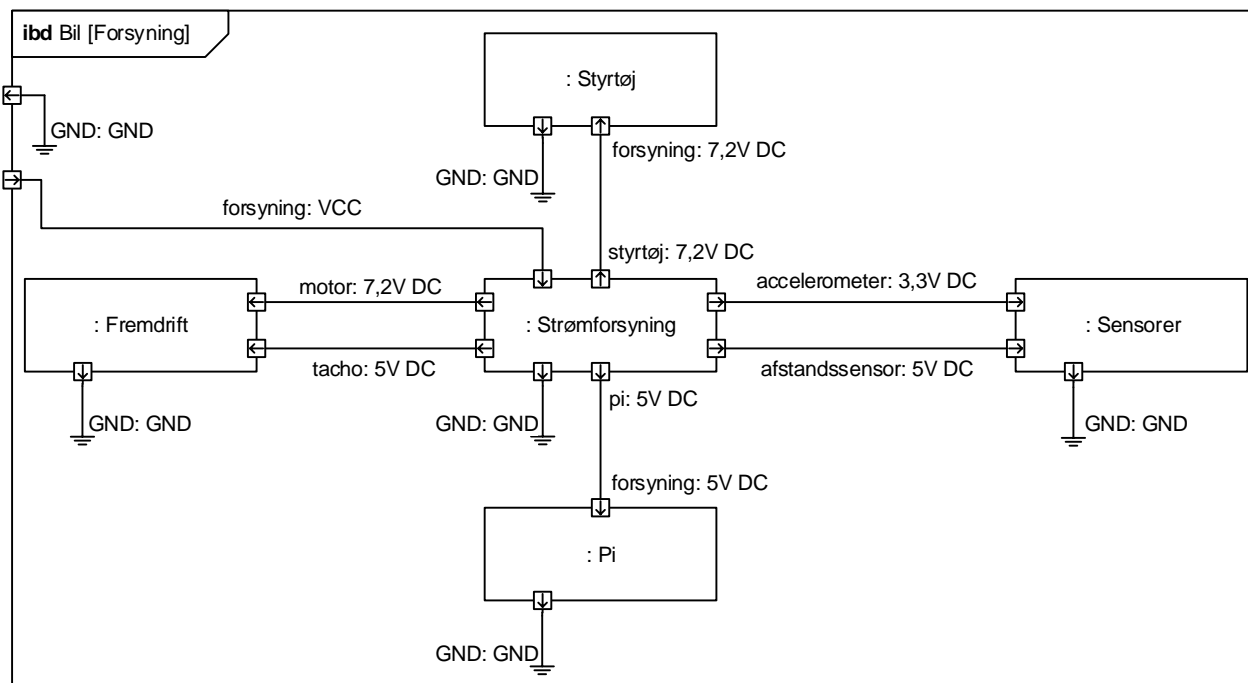
Figur 8: IBD for bil

### Signalbeskrivelser for bilens signaler

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
Propulsion: torque	Baghjulenes torque til underlaget.	-	
motorSpeed: PWM	Et PWM signal der bestemmer motorhastigheden.	Frekvens: 30kHz +/- 1kHz 0-5V +/- 0.2V	Logisk signal: Lav = 0V +/- 0.2V Høj = 5V +/- 0.2V
forward: bool	Kontrolsignal til H-bro.	0-5V ± 0.2V	Lav = 0V +/- 0.2V 'idle' Høj = 5V +/- 0.2V 'frem'
reverse: bool	Kontrolsignal til H-bro.	0-5V ± 0.2V	Lav = 0V +/- 0.2V 'idle' Høj = 5V +/- 0.2V 'tilbage'
tachoSpeed: freq	Digitalt signal med varierende frekvens afhængig af baghjulenes omdrejningshastighed.	-	Vejledende: 64Hz = 10Km/t Logisk signal: Lav = 0V +/- 0.2V Høj = 5V +/- 0.2V
Inout SDA: bool	I <sup>2</sup> C dataline til sensorer herunder accelerometer og afstandssensorer.	0-5V ± 0.5V	Logisk signal: Lav = 0V ± 0.5V Høj = 7.2V ± 0.5V
Inout SCL: bool	I <sup>2</sup> C clockline til sensorer herunder accelerometer og afstandssensorer.	0-5V ± 0.5V	Logisk signal: Lav = 0V ± 0.5V Høj = 7.2V ± 0.5V
Pulses: sound	Ultralydsbølger afsendt af sensor.	Jfv. Datablad (henvisning kommer senere)	
reflection: sound	Refleksionsbølge af udsendte ultralydsbølger.	Jfv. Datablad (henvisning kommer senere)	
retning: PWM	PWM signal der vha pulsbredden angiver hvilken retning servomotoren skal dreje og dermed hvilken retning bilen skal dreje.	Pulsbredde: 0.5ms - 2.5ms Freq = 360Hz 0.5ms = 18% Duty cycle (Venstre) 2.5ms = 90% Duty cycle (Højre)	
retning: rotation	Får bilen til at dreje.	30 grader til hhv. venstre og højre ± 5 grader	

## Forsyninger

Diagrammet på figur 9 tilsvarende direkte figur 8, blot med beskrivelsen af forsyning. Dette giver forbedret overblik da de to diagrammer sat sammen bliver uoverskueligt.



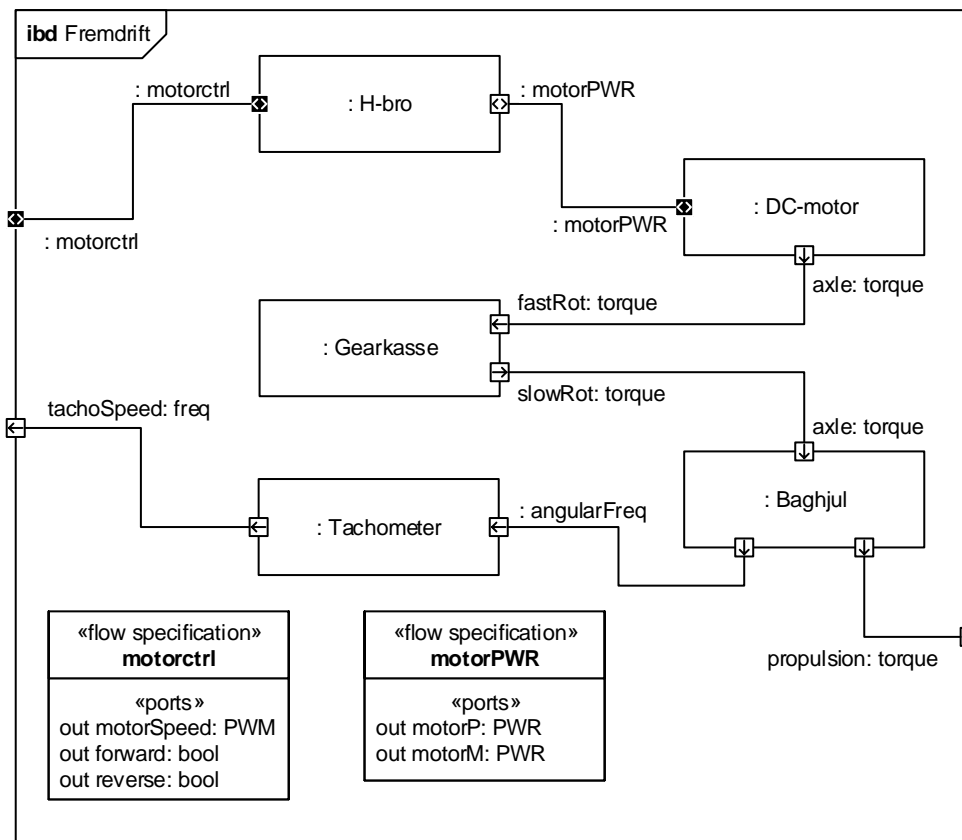
Figur 9: IBD for bilens forsyninger

## Signalbeskrivelse for bilens forsyning

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
forsyning: VCC	Forsyningsspænding fra det tilkoblede batteri.	7.2V DC +/- 1V max. 20A	
GND: GND	Reference.	0V	
styrtøj: 7.2V DC	Forsyningsspænding til styrtøj herunder servomotor.	7.2V DC $\pm$ 0,5V	
accelerometer: 3.3V DC	Forsyningsspænding til accelerometeret.	3,3V DC +/- 0,2V	
afstandssensor: 5V DC	Individuel forsyningsspænding til afstandssensorerne.	5V DC +/- 0,5V	
PI: 5V DC	Individuel forsyningspænding til PI.	5V DC +/- 0,5V	
Tacho: 5V DC	Individuel forsyningspænding til tachometeret.	5V DC +/- 0,5V	
motor: 7.2V DC	Individuel forsyningspænding til motoren.	5V DC +/- 0,5V	

### 3.3.2 Fremdrift

Bilens fremdrift forårsages af motoren samt tilhørende elektronik, hvilket er beskrevet på figur 10. Det skal igen noteres at forsyningen til H-broen ikke er på diagrammet, men findes på figur 9. Motoren trækker altså ikke sin strøm fra signalet motorCtrl.



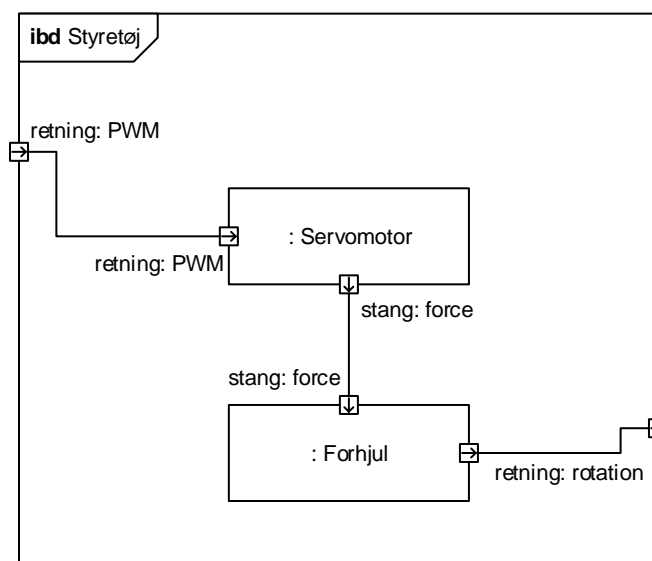
Figur 10: IBD for blokken fremdrift

### Signalbeskrivelse for fremdrift

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
motorSpeed: PWM	Et PWM signal der bestemmer motorhastigheden.	Frekvens: 30kHz +/- 1kHz 0-5V +/- 0.2V	Logisk signal: Lav = 0V +/- 0.2V Høj = 5V +/- 0.2V
forward: bool	Kontrolsignal til H-bro.	0-5V ± 0.2V	Lav = 0V +/- 0.2V 'idle' Høj = 5V +/- 0.2V 'frem'
reverse: bool	Kontrolsignal til H-bro.	0-5V ± 0.2V	Lav = 0V +/- 0.2V 'idle' Høj = 5V +/- 0.2V 'tilbage'
motorP: PWR	Et PWM med frekvens som motorSpeed, dog med mulighed for højere effekt. Dette signal forsyner motoren.	Frekvens: 30kHz +/- 1kHz 0-7,2V +/- 0.5V	Logisk signal: Lav = 0V +/- 0.5V Høj = 7,2V +/- 0.5V
motorM: PWR	Reference til motorP.	0V ± 0.5V	
fastRot: torque	Kraft der overføres fra motor til gearkasse via drivaksel.	-	
slowRot: torque	Kraft der overføres fra gearkasse til baghjul via drivaksel.	-	
: angularFreq	Hjulenes omdrejningshastighed.	-	
tachoSpeed: freq	Digitalt signal med varierende frekvens afhængig af baghjulets omdrejningshastighed.	-	Vejledende: 64Hz = 10Km/t Logisk signal: Lav = 0V +/- 0.2V Høj = 5V +/- 0.2V
Propulsion: torque	Baghjulets torque til underlaget.	-	

### 3.3.3 Styretøj

De interne signaler for blokken styretøj er beskrevet nedenfor i figur 11.



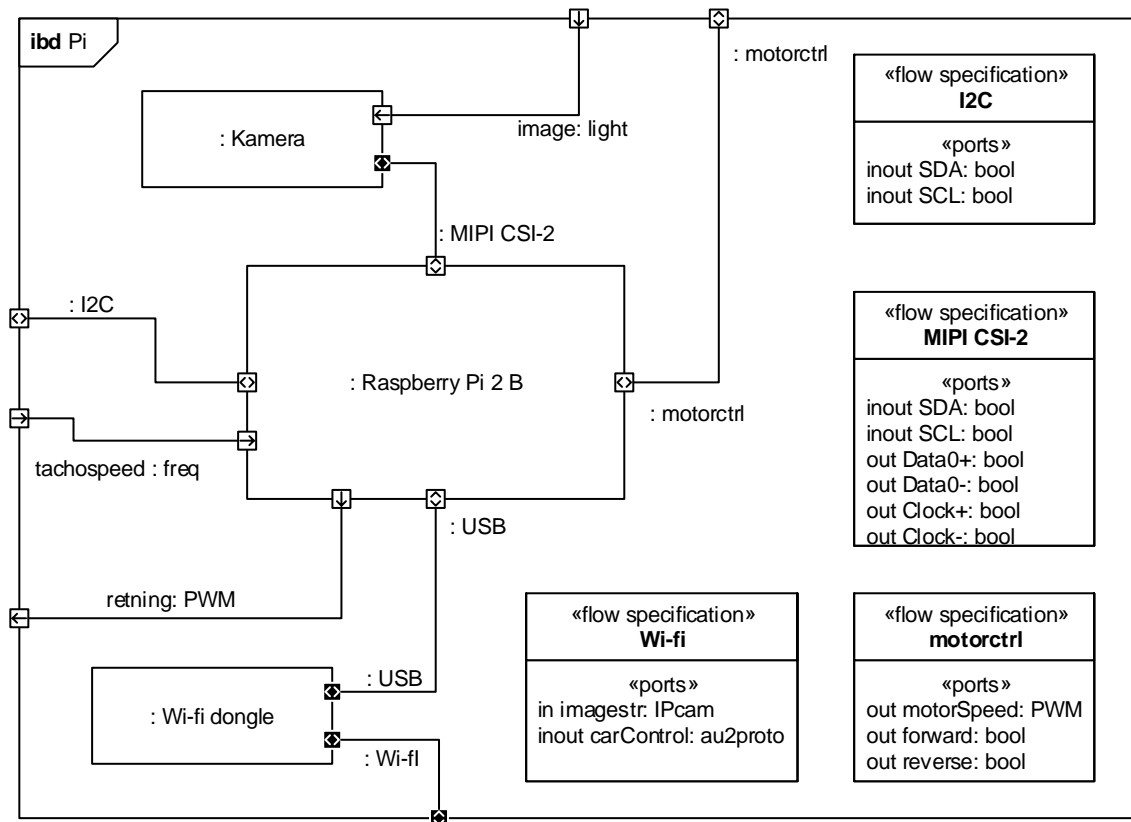
Figur 11: IBD for blokken styretøj

#### signalbeskrivelse for styretøj

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
retning: PWM	PWM signal der vha pulsbredden angiver hvilken retning servomotoren skal dreje og dermed hvilken retning bilen skal dreje.	Pulsbredde: 0.5ms – 2.5ms Freq = 360Hz 0.5ms = 18% Duty cycle (Venstre) 2.5ms = 90% Duty cycle (Højre)	
stang: force	Skal overføre kraften fra servomotoren til forhjulene. Dette sker via en stang.	-	
retning: rotation	Får bilen til at dreje.	30 grader til hhv. venstre og højre $\pm$ 5 grader	

### 3.3.4 Pi

Her beskriver intern kommunikation for controlleren PI i systemet.



Figur 12: IBD for blokken PI

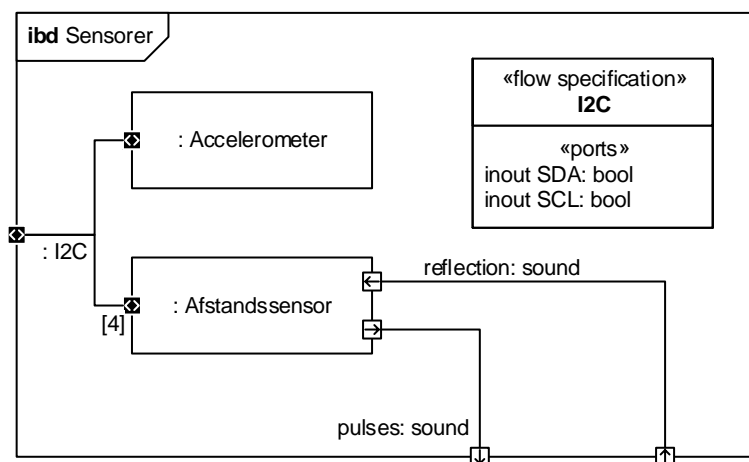
### Signalbeskrivelse for Pi



Signal (navn: type)	Funktion	Tolerancer	Kommentarer
SDA: bool	I2C dataline til sensorer herunder accelerometer og afstandssensorer	0-5V $\pm$ 0.5V	Logisk signal: Lav= 0V $\pm$ 0.5V Høj= 7.2V $\pm$ 0.5V
SCL: bool	I2C clockline til sensorer herunder accelerometer og afstandssensorer	0-5V $\pm$ 0.5V	Logisk signal: Lav= 0V $\pm$ 0.5V Høj= 7.2V $\pm$ 0.5V
Image: light	Lysindfald til kamerasensor	-	-
SDA: bool	I2C dataline til sensorer herunder accelerometer og afstandssensorer	0-5V $\pm$ 0.5V	Logisk signal: Lav= 0V $\pm$ 0.5V Høj= 7.2V $\pm$ 0.5V
SCL: bool	I2C clockline til sensorer herunder accelerometer og afstandssensorer	0-5V $\pm$ 0.5V	Logisk signal: Lav= 0V $\pm$ 0.5V Høj= 7.2V $\pm$ 0.5V
motorSpeed: PWM	Et PWM signal der bestemmer motorhastigheden.	Frekvens: 30kHz $\pm$ 1kHz 0-5V $\pm$ 0.2V	Logisk signal: Lav= 0V $\pm$ 0.2V Høj= 5V $\pm$ 0.2V
forward: bool	Kontrolsignal til H-bro	0-5V $\pm$ 0.2V	Logisk signal: Lav= 0V $\pm$ 0.2V "idle" Høj= 5V $\pm$ 0.2V "forward"
reverse: bool	Kontrolsignal til H-bro	0-5V $\pm$ 0.2V	Logisk signal: Lav= 0V $\pm$ 0.2V "idle" Høj= 5V $\pm$ 0.2V "back"
:USB	Serielforbindelse mellem Wi-fi dongle og Pi	VBUS = 5V $\pm$ 0.2V D- = 5V $\pm$ 0.2V D+ = 5V $\pm$ 0.2V GND = 0V	VBUS for Low-power port: Diff "1" (D+)-(D-) > 200 mV og D+ > VIH (min)  Diff "0" (D-)-(D+) > 200 mV og D- > VIH (min)

### 3.3.5 Sensorer

På figur 13 ses de interne signaler for blokken Sensorer.



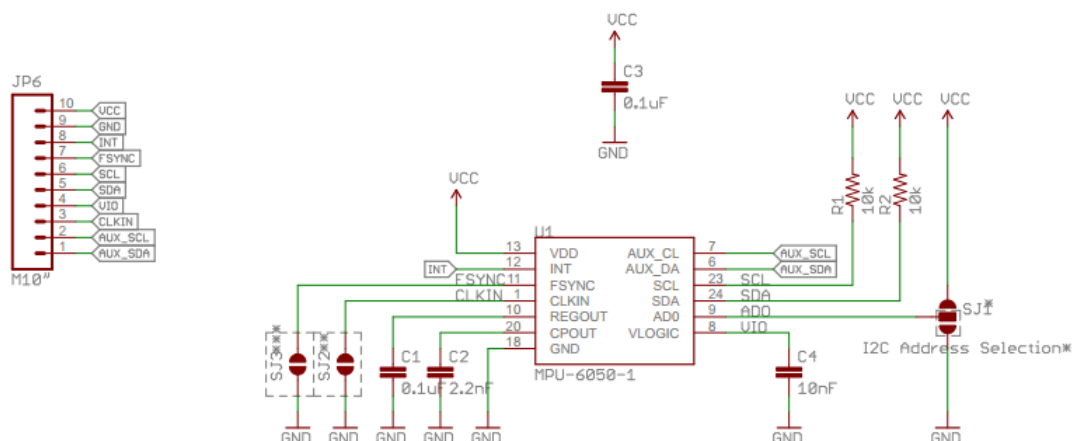
Figur 13: IBD for blokken sensorer

#### signalbeskrivelse for sensorer

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
Inout SDA: bool	I <sup>2</sup> C dataline til sensorer herunder accelerometer og afstandssensorer.	0-5V ± 0.5V	Logisk signal: Lav = 0V ± 0.5V Høj = 7.2V ± 0.5V
Inout SCL: bool	I <sup>2</sup> C clockline til sensorer herunder accelerometer og afstandssensorer.	0-5V ± 0.5V	Logisk signal: Lav = 0V ± 0.5V Høj = 7.2V ± 0.5V
Pulses: sound	Ultralydsbølger afsendt af sensor.	Jfv. Datablad (henvisning kommer senere)	
reflection: sound	Refleksionsbølge af udsendte ultralydsbølger.	Jfv. Datablad (henvisning kommer senere)	

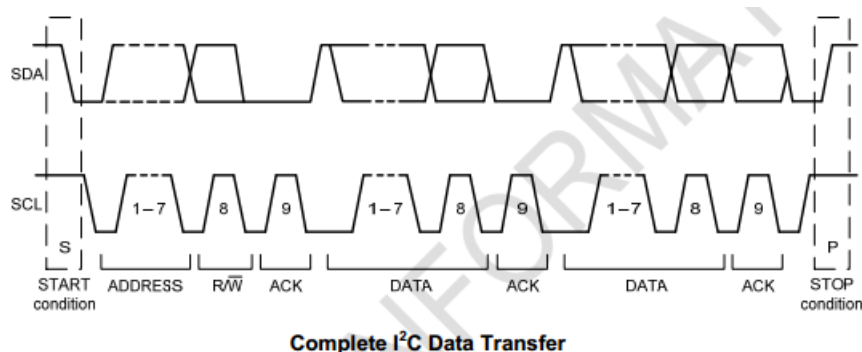
### 3.3.6 MPU-6050 Accelerometer/Gyroskop

MPU-6050 er en kombination af et accelerometer, et gyroskop, begge med 3 akser og et termometer. Det betyder for systemet at det er i stand til at registrere en ændring i acceleration og/eller orientering i alle retninger. Sensoren er blevet valgt til projektet på baggrund af et I<sup>2</sup>C interface, som derved kan tilkobles en samlet bus sammen med andre sensorer på bilen. Udover dette giver det konstruerede breakoutboard mulighed for nem tilslutning, men fortsat lille størrelse på sensoren. Sensoren fungerer altid som en slave med adressen `0b110100X`, hvor X bliver bestemt af det logiske niveau på pin AD0, der som standard er lav.



Figur 14: MPU-6050 diagram

Diagrammet for sensoren er vist i figur 14. Maksimal bushastighed for MPU-6050 er 400kHz, men da der er andre sensorer som arbejder langsommere end MPU-6050 i systemet, passer den fint ind. Accelerometeret er en MEMS-type, hvor der er bygget mikroskopiske kondensatorer ind i chippen, som kan fjedre og bevæge sig, hvilket registreres som en ændring i kapacitans. Denne ændring kan omregnes til nogle brugbare værdier, og kan herfra anvendes til bl.a. retningsbestemmelse. Der er i alt 7 16-bits registre i sensoren, som hver især er tilknyttet til en ADC på hver akse, med undtagelse af register nr. 7, som er tilknyttet termometeret. Protokollen for kommunikation med sensoren ser således ud:



Figur 15: I<sup>2</sup>C protokol for MPU-6050

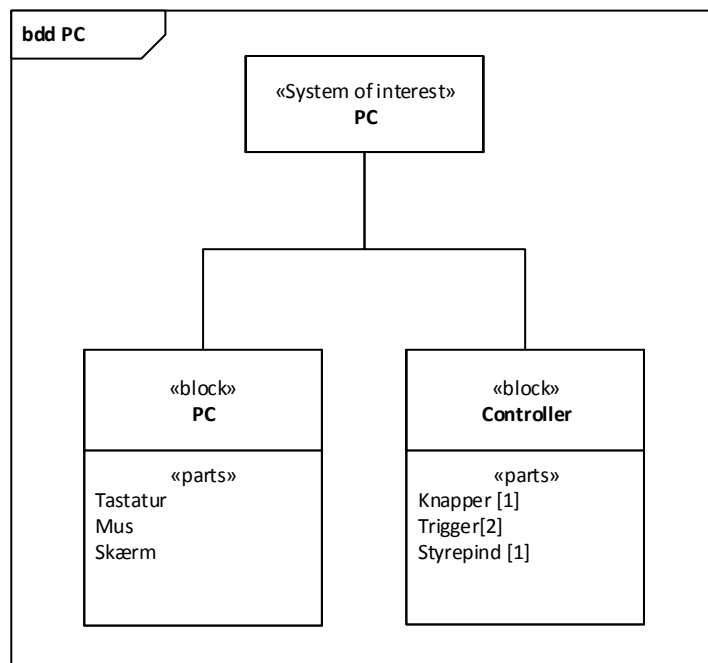
Som det ses i figur 15, starter masteren med at sætte en startsekvens ud på SDA (HIGH-to-LOW),

mens SCL er høj. Herefter betragtes bussen som optaget, indtil der bliver sendt en stopsekvens på SDA (LOW-to-HIGH) af masteren, mens SCL ligeledes er høj. Efter startsekvensen bliver der sendt en 7-bits adresse og en R/W bit. Data der bliver transmitteret over I<sup>2</sup>C bliver sendt i pakker af 8 bits. Når der først er sendt en startsekvens, er der ingen begrænsning på hvor meget data der må sendes, udover at der efter hver pakke, skal registreres en acknowledge. MPU-6050 indeholder desuden en DMP (Digital Motion Processor), som har til opgave at håndtere noget af dataprocesseringen fra selve MPU-6050.

## 3.4 PC

### 3.4.1 BDD for PC

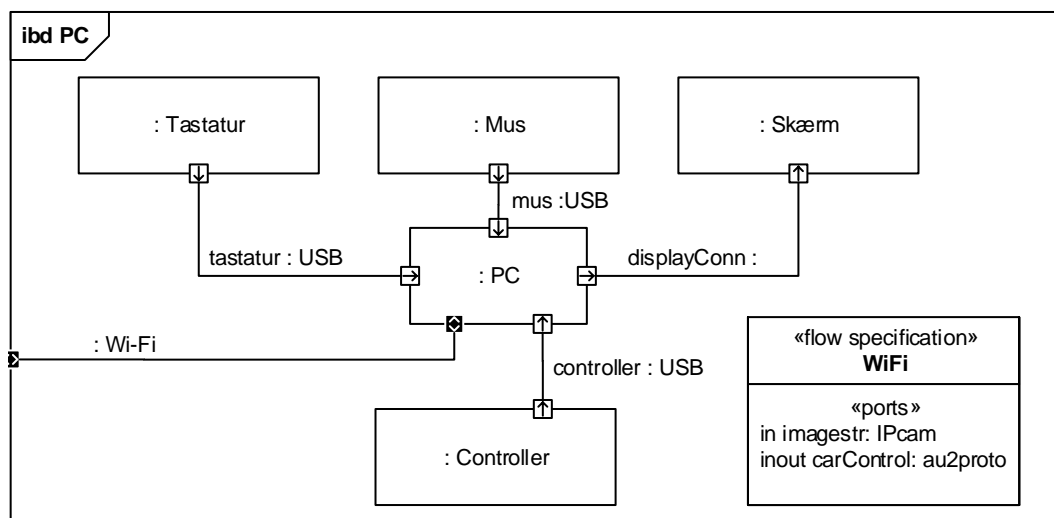
På figur 16 ses et overordnet blokdiagram for PC'en og dens interagerende dele.



Figur 16: BDD for PC kontekst

### 3.4.2 IBD for PC

På figur 17 ses alle interne signaler og en yderligere specificering af blokdiagrammet for pc på figur 16



Figur 17: IBD for blokken PC

### 3.4.3 Signalbeskrivelse for PC

Signal (navn: type)	Funktion	Tolerancer	Kommentarer
Imagestr: IPcam	TODO Karsten		
carControl: au2proto	TODO Karsten		
Tastetur: USB	Seriellkommunikation fra tastetur til PC	VBUS = 5V min. 4.40V max 5.25V GND = 0V D- = 5V +/- 0.2V D+ = 5V +/- 0.2V	VBUS for Low power port: Diff "1" (D+) - (D-) > 200 mV and D+ > VIH (min) Diff "0" (D-) - (D+) > 200 mV and D- > VIH (min)
Mus: USB	Seriellkommunikation fra mus til PC	VBUS = 5V min. 4.40V max 5.25V GND = 0V D- = 5V +/- 0.2V D+ = 5V +/- 0.2V	VBUS for Low power port: Diff "1" (D+) - (D-) > 200 mV and D+ > VIH (min) Diff "0" (D-) - (D+) > 200 mV and D- > VIH (min)
displayConn:			
Controller: USB	Seriellkommunikation fra Controller til PC	VBUS = 5V min. 4.40V max 5.25V GND = 0V D- = 5V +/- 0.2V D+ = 5V +/- 0.2V	VBUS for Low power port: Diff "1" (D+) - (D-) > 200 mV and D+ > VIH (min) Diff "0" (D-) - (D+) > 200 mV and D- > VIH (min)

## 3.5 Protokolbeskrivelse

### 3.5.1 Kamera

Kameraet er påmonteret bilen og er tilsluttet til Pi'en hvor programmet motion er blevet installeret. Motion har den fordel at det kan lave en socketforbindelse hvorved Pi'en kan streame videodata den vej igennem. På PC'en modtages videoen igennem GUI'en ved hjælp af opensource biblioteker fra VLC media player. Herved laves der også en socket til at modtage videoen i.

Kommando	Svar	Beskrivelse	Bitmønstre
openPlayer	Kontinuert stream fra motion	Motion streamer video data til PC'en	UDP socket forbindelse

Tabel 13: Kamera Protokol

Motion streamer video ligeså snart der er tændt for bilen. Der er derfor ikke nogen direkte kommunikation mellem PC'en og Pi'en igennem Kameraprotokollen.

### 3.5.2 GUI

GUI'en kommunikerer mellem PC'en og bilen igennem to TCP forbindelser. Controlleren har sin egen hvor data bliver sendt kontinuert. updateData og shutDown har også den samme forbindelse updateData opdateres kontinuert. shutDown sendes kun når brugeren trykker på "Lukned" på GUI'en.



Kommando	Svar	Beskrivelse	Bitmønstre
updateData	sendStatus	GUI'en sender makshastighed, status på AKS og kalibrering af styretøj til bilen. Bilen svarer tilbage med hastighed, afstand til forhindring og acceleration	Ved updateData sendes data som char i rækkefølgen: status, AKS, kalibrering. Ved sendStatus sendes data også som char i rækkefølgen: hastighed, afstand, acceleration
shutDown	shuttingDown	GUI'en sender besked til bilen om at den skal lukke dens software ned	shutDown sendes som en string "shutdown" shuttingDown sender det samme tilbage
controllerFrem	<i>Intet svar</i>	Bruger har trykket på RT på controlleren	Char fra 0-255 afhængig af hvor hårdt der trykkes på knappen
controllerTilbage	<i>Intet svar</i>	Bruger har trykket på LT på controlleren	Char fra 0-255 afhængig af hvor hårdt der trykkes på knappen
controllerVenstre	<i>Intet svar</i>	Bruger har ændret positionen af venstre styrepind, til venstre, på controlleren	Char "V"
controllerHøjre	<i>Intet svar</i>	Bruger har ændret positionen af venstre styrepind, til højre, på controlleren	Char "H"

Tabel 14: GUI Protokol



## 4 Accepttest

### Version

Dato	Version	Initialer	Ændring
29. september	1	Alle	Første udkast.
26. oktober	2	PKP, KT og JEP	Mindre rettelser efter review
	3		
	4		

## 4.1 Funktionelle Krav

Fremgangsmåden for test af funktionelle krav er generelt taget udgangspunkt i Use Cases. I tabel 15 er vist en matrice der sammenholder Use Cases med funktionelle krav, der sikrer at alle krav bliver testet ved test af Use Cases. Der henvises til kravnumre i afsnit 2.4 på side 10.

Krav	1	2	3	4	5	6	7	8	9	10	11	12
UC1						X				X		
UC2										X		
UC3							X					
UC4	X	X	X				X	X	X			
UC5	X		X		X							
UC6		X			X							
UC7			X									X
UC8						X						
UC9											X	
UC10			X	X								
UC11		X										
UC12												

Tabel 15: Use Case-krav matrice

### Use Case 1: Aktiver system

Use case under test		UC1: Aktiver system		
Scenarie		Hovedscenarie		
Forudsætning		Netværksforbindelse er opsat og fungerende		
Step	Handling	Forventet Resultat	Resultat	Godkendt / Kommentar
1.1	Bruger sætter bilens "ON/OFF"-switch til "ON".	Visuel test: Lampe på bilens strømforsyning lyser.		
1.2	Bruger starter software på PC.	Visuel test: Hovedvinduet vises på skærmen.		
1.3	Bruger trykker på "Opret forbindelse".	Visuel test: Hovedvindue viser "Forbindelse oprettet".		
1.4	Bruger observerer hovedvinduet.	Visuel test: Videostream vises i hovedvinduet.		
1.5	Bruger observerer hovedvinduet.	Visuel test: Bilens aktuelle hastighed vises i hovedvinduet.		

1.6	Bruger observerer hovedvinduet.	Visuel test: Bilens aktuelle tyngdeacceleration vises i hovedvinduet.		
1.7	Bruger observerer hovedvinduet.	Visuel test: Data fra bilens afstandssensorer vises i hovedvinduet.		

Tabel 16: Accepttest for UC1: Aktiver system

**Use Case 2: Stream Video**

<b>Use case under test</b>		UC2: Stream Video		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1 frem til punkt 5 er fuldført		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
2.1	Bruger har Wireshark åbent på samme computer. Wireshark er opsat til at overvåge det pågældene netværk.	Visuel test: I Wireshark observeres der for overføring af pakker fra bilens IP-adresse til computerens IP-adresse.		
2.2	Bruger åbner AU2 softwaren på PC'en og trykker "Operet forbindelse".	Visuel test: Der vises et live-feed fra bilens kamera.		

Tabel 17: Accepttest for UC2: Stream Video

**Use Case 3: Overvåg sensorer**

Use case under test		UC3: Overvåg sensor		
Scenarie		Hovedscenarie		
Forudsætning		UC1 frem til punkt 6 er fuldført		
Step	Handling	Forventet Resultat	Resultat	Godkendt / Kommentar
3.1	Bruger observerer terminalen i Raspberry Pi 2en, igennem en SSH forbindelse.	Visuel test: Terminalen udskriver "Tachometer initializing... Done".		
3.2	Bruger observerer terminalen i Raspberry Pi 2en, igennem en SSH forbindelse.	Visuel test: Terminalen udskriver "Accelerometer initializing... Done".		
3.3	Bruger observerer terminalen i Raspberry Pi 2en, igennem en SSH forbindelse.	Visuel test: Terminalen udskriver "Distancesensors initializing... Done".		
3.4	Bruger kører en tur med bilen og observerer hovedmenu i softwaren på PC.	Visuel test: Bruger observerer at data for bilens hastighed, afstand til forhindring og acceleration fremgår af brugerfladen.		

Tabel 18: Accepttest for UC3: Overvåg sensor

**Use Case 4: Undvig forhindring**

Use case under test		UC4: Undvig forhindring		
Scenarie		Hovedscenarie		
Forudsætning		UC1 er gennemført, UC3 er gennemført.		
Step	Handling	Forventet Resultat	Resultat	Godkendt / Kommentar
4.1	Bruger styrer bilen fremad mod en forhindring på min. (30cm × 30cm) vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen er umiddelbart til venstre for objektet.	Visuel test: Bruger observerer at bilen ændrer kurs til højre på trods af brugerinput.		
4.2	Bruger tester om det er muligt at styre bilen igen med Xbox-360 controlleren.	Visuel test: Bruger observerer at bilen reagerer på brugerinput.		
4.3	Bruger styrer bilen fremad mod en forhindring på (30cm × 30cm) vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen er umiddelbart til højre for objektet.	Visuel test: Bruger observerer at bilen ændrer kurs til venstre på trods af brugerinput.		
4.4	Bruger styrer bilen fremad mod en forhindring på (30cm × 30cm) vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen har retning lige mod objektet.	Visuel test: Bruger observerer at bilen standser på trods af brugerinput.		



4.5	Bruger bakker mod en forhindring på $(30cm \times 30cm)$ vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen er umiddelbart til venstre for objektet.	Visuel test: Bruger observerer at bilen ændrer kurs til venstre på trods af brugerinput.		
4.6	Bruger bakker bilen mod en forhindring på $(30cm \times 30cm)$ vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen er umiddelbart til højre for objektet.	Visuel test: Bruger observerer at bilen ændrer kurs til højre på trods af brugerinput.		
4.7	Bruger bakker bilen mod en forhindring på $(30cm \times 30cm)$ vinkelret på bilens kørebane vha. Xbox-360 controlleren, således at bilen har retning lige mod objektet.	Visuel test: Bruger observerer at bilen standser på trods af brugerinput.		

Tabel 19: Accepttest for UC4: Undvig forhindring

**Use Case 5: Kør bil frem/tilbage**

<b>Use case under test</b>		UC5: Kør bil frem/tilbage		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er fuldført og systemet er operationelt.		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
5.1	Bruger holder "RT" på Xbox-360 controlleren halvt nede.	Visuel test: Bruger observerer at bilen accelererer fremad til halv makshastighed og holder denne.		
5.2	Bruger holder "RT" på Xbox-360 controlleren helt nede.	Visuel test: Bruger observerer at bilen accelererer fremad til makshastighed og holder denne.		
5.3	Bruger holder "RL" på Xbox-360 controlleren halvt nede.	Visuel test: Bruger observerer at bilen accelererer bagud til halvdelen af makshastighed og holder denne.		
5.4	Bruger holder "RL" på Xbox-360 controlleren helt nede.	Visuel test: Bruger observerer at bilen accelererer bagud til makshastighed og holder denne.		

Tabel 20: Accepttest for UC5: Kør bil frem/tilbage

**Use Case 6: Drej bil til højre/venstre**

Use case under test		UC6: Drej bil til højre/venstre		
Scenarie		Hovedscenarie		
Forudsætning		UC1: Aktiver system er fuldført og systemet er operationelt		
Step	Handling	Forventet Resultat	Resultat	Godkendt / Kommentar
6.1	Bruger ændrer position af den venstre styrepind på Xbox360-controlleren til venstre yderposition.	Visuel test: Bilens forhjul drejer 30° til venstre i forhold til udgangspunktet.		
6.2	Bruger ændrer position af den venstre styrepind på Xbox360-controlleren halvvejs til venstre yderposition.	Visuel test: Bilens forhjul drejer 15° til venstre fra udgangspunktet.		
6.3	Bruger ændrer position af den venstre styrepind på Xbox360-controlleren til højre yderposition.	Visuel test: Bilens forhjul drejer 30° til højre fra udgangspunktet.		
6.4	Bruger ændrer position af den venstre styrepind på Xbox360-controlleren halvvejs til højre yderposition.	Visuel test: Bilens forhjul drejer 15° til højre fra udgangspunktet.		
6.5	Bruger ændrer position af den venstre styrepind på Xbox360-controlleren til center mellem højre og venstre yderposition.	Visuel test: Bilens forhjul går tilbage til udgangsposition.		

Tabel 21: Accepttest for UC6: Drej bil til højre/venstre

**Use Case 7: Brems bil**

<b>Use case under test</b>		UC7: Brems Bil		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er fuldført og systemet er operationelt.		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
7.1	Bruger trykker på "X" knappen på Xbox-360 controlleren.	Visuel test: Bruger observerer at bilens hastighed sænkes hvis i fart, ellers tændes bilens bremselys blot.		

Tabel 22: Accepttest for UC7: Brems Bil

**Use Case 8: Konfigurer IP-adresse**

<b>Use case under test</b>		UC8: Konfigurer IP-adresse		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt.		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
8.1	Bruger trykker på "Konfigurer IP".	Visuel test: Konfigurations menuen for IP-adressen vises, og der er mulighed for at indtaste en IP-adresse.		
8.2	Bruger trykker på bilens IP-adresse. Og trykker "Gem".	Visuel test: Systemet viser "Hovedvindue".		
8.3	Bruger trykker på "Opret forbindelse".	Visuel test: System viser at der er "Forbindelse oprettet".		

Tabel 23: Accepttest for UC8: Konfigurer IP-adresse

**Use Case 9: Tænd/Sluk AKS**

<b>Use case under test</b>		UC9: Tænd/sluk AKS		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt.		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
9.1	Bruger trykker på "AKS".	Visuel test: System viser "AKS-menu" med mulighed for at tænde/slukke for AKS og status for AKS.		
9.2	Bruger trykker på "Tænd AKS".	Visuel test: "Hovedvindue" indikerer status af AKS for tændt.		
9.3	Bruger trykker på "AKS".	Visuel test: System viser "AKS-menu" med mulighed for at tænde/slukke for AKS og status for AKS.		
9.4	Bruger trykker på "Sluk AKS".	Visuel test: "Hovedvindue" indikerer status af AKS for slukket.		

Tabel 24: Accepttest for UC9: Tænd/sluk AKS

**Use Case 10: Indstil maksimalhastighed**

<b>Use case under test</b>		UC10: Indstil makshastighed		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedvindue" samt at systemet er operationelt.		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
10.1	Bruger trykker på "Indstil makshastighed".	Visuel test: Hovedvindue viser menu med mulighed for at indtaste makshastighed fra 1-10 km/t.		
10.2	Menuen viser bilens nuværende makshastighed.	Den nuværende makshastighed vises.		
10.3	Bruger indtaster bilens ønskede makshastighed.	Menuen viser den ønskede makshastighed.		
10.4	Bruger trykker på "Opdater".	Systemet viser den nye makshastighed.		
10.5	Bruger holder "RT" inde på Xbox 360 controlleren.	Bilen accelererer til den angivne maksimalhastighed.		

Tabel 25: Accepttest for UC10: Indstil makshastighed

**Use Case 11: Kalibrer styretøj**

<b>Use case under test</b>		UC11: Kalibrer styretøj		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er udført, bilen og PC er på samme netværk, at systemet viser "Hovedmenu", at systemet er operationelt samt bilen holder stille		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
11.1	Bruger vælger "Kalibrer styretøj"	Visuel test: Menu med mulighed for kalibrering fremkommer.		
11.2	Bruger indtaster en værdi mellem 50 og -50 for kalibrering.	Den ønskede værdi vises.		
11.3	Bruger trykker på "Gem".	Systemet gemmer værdien på bilen og forhjulene drejer en absolut værdi mod enten højre eller venstre: positiv værdi giver udslag til højre, og negativ værdi giver udslag venstre.		
11.5	Systemet returnerer til "Hovedvindue"	Visuel test: "Hovedvindue" fremkommer		

Tabel 26: Accepttest for UC11: Kalibrer styretøj



**Use Case 12: Afbryd system**

<b>Use case under test</b>		UC12: Afbryd system		
<b>Scenarie</b>		Hovedscenarie		
<b>Forudsætning</b>		UC1: Aktiver system er fuldført, bilen holder stille og systemet er operationelt		
<b>Step</b>	<b>Handling</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
12.1	Bruger lukker ned for softwaren på PC'en.	Visuel test: Hovedvinduet forsvinder fra skærmen.		
12.2	Bruger skubber kontakten "ON/OFF" på undersiden af bilen til position "OFF"	Visuel test: Lampe på strømforsyning slukker.		

Tabel 27: Accepttest for UC12: Afbryd system

## 4.2 Ikke-funktionelle krav

Krav	Test	Forventet Resultat	Resultat	Godkendt/ kommentar
1.	Der udmåles en strækning, på en vandret overflade, på 10 meter. Bilen startes således at maksimumshastighed er nået når den passerer startpunktet for den udmålte strækning. Der tages tid på bilen fra startpunkt til slutpunkt af strækningen. Herefter omregnes disse data til en hastighed.	Der måles en maksimumshastighed på 10 km/t $\pm$ 10%.		
2.	Der udmåles en strækning, på en vandret overflade, på 1 meter. Bilen startes således at maksimumshastighed er nået når den passerer startpunktet for den opmålte strækning. Bilen sættes til at bremse indtil at bilen er i stilstand. Der observeres om bilen er kommet ud over slutpunkt for den udmålte strækning.	Bil er i stilstand inden for den udmålte stræknings start- og slutpunkt.		

3.	Bilen sættes til at accelerere ligeud på en vandret overflade. Der tages tid fra start af bilens acceleration. Ved 6 sekunders mærket tages der to billeder af bilen med et kamera med en maks. lukketid på $1/100s$ . Disse billeder bruges til at finde hastigheden af bilen, ved at aflæse længden bilen har flyttet sig og dividere det med tiden der er gået mellem de to billeder.	De beregnede data viser at bilen har nået maksimumhastighed på $10 \text{ km/t} \pm 10\%$ .		
4.	Bruger trykker på "B"-knappen på Xbox-360 controlleren. Der måles en tid fra tryk på knap til test-LED på bilen lyser.	Den målte tid overstiger ikke $50ms$ .		
5.	Der placeres en genstand der opfylder givne krav for forhindring ( $30cm \times 30cm$ ) i afstanden $6m$ fra sensoren og der måles om sensoren detekterer pågældene genstand.	Der observeres at sensoren detekterer genstanden.		
6.	Bruger slukker for program på PC.	Der observeres at bilens bremse-LED lyser indenfor $50ms$ .		
7.	Datablad for kamera aflæses.	Kameraet er detekteret til at have en opdateringshastighed på minimum 15 billeder i sekundet.		

8.	Der tages et screenshot af hovedvinduet. Selve videofeedet beskæres i mspaint.exe og måles.	Den målte opløsning er $640 \times 480$ pixels.		
9.	Wireshark anvendes til at aflæse antal kommandoer sendt per sekund.	Den aflæste mængde kommandoer er minimum 60.		
10.	Bruger kigger på HID.	HID består af Xbox-360 controller og tastatur.		

Tabel 28: Ikke funktionelle krav