

# Projekt AutoCar

# Rapport

Diplomingeniør Elektronik  
4. Semesterprojekt forår 2016

Ingeniørhøjskolen Aarhus Universitet  
Vejleder: Lars G. Johansen

27. maj 2016

---

Jonas Baatrup  
Studienr. 201405146

---

Troels Ringbøl Brahe  
Studienr. 20095221

---

Nicolai H. Fransen  
Studienr. 201404672

---

Jesper Kloster  
Studienr. 201404571

---

Rasmus Harboe Platz  
Studienr. 201408608

---

Nicolai Bonde  
Studienr. 201404519

---

Emil Jepsen  
Studienr. 20092013

**Ansvarsområder** Tabellen nedenfor viser de primære ansvarsområder for hver af gruppens medlemmer.

Navn	Ansvarsområder
Jonas Baatrup	Motorstyring, Motor-design, Regulering
Nicolai Bonde	RPi kode, Bil-design, Forsyning
Jesper Kloster	Servo, Vejbanesensorer, Tachometer
Nicolai Fransen	Servo, Vejbanesensorer, Tachometer
Troels Brahe	Sonar, Regulering, RPi kode
Emil Jepsen	Motorstyring, Regulering, SPI
Rasmus Platz	TCP, GUI

Tabel over ansvarsfordeling i projektet

## Resumé

Denne rapport beskriver udviklingen af et 4. semester-projekt på IHA. Problemstillingen omhandler design og implementering af en selvkørende bil, AutoCar. AutoCar har en DC-motor til fremdrift, en servo-motor til retningsstyring, en sonar-sensor til afstandsbedømmelse, en vejbanesensor i hver side til detektion af vejbanestriber, to batterisensorer til at vise batteriniveauerne, et tachometer til at finde AutoCar's fart, og intern logik, regulering, og kommunikation for at få det hele til at fungere sammen. Brugeren kan via interfacet starte og stoppe AutoCar, få udskrevet status fra sensorerne, sætte farten og anmode om overhaling. AutoCar skal selv sørge for at køre inden for vejbanestriberne.

AutoCar er udviklet med en PSoC 4 og en Raspberry Pi 2b, der tilsammen fungerer som kontrolenhed for AutoCar. GUI'et er designet med QT Creator i en Linux-terminal.

Udviklingsprocessen har båret præg af iterative værktøjer som SCRUM og V-modellen, og med fokus på design fra bunden og op. ASE-modellen for projektudvikling er også benyttet i den tidsmæssige planlægning. Brugen af disse modeller har hjulpet i udviklingen af projekt. Projektprocessen er mundet ud i en prototype hvor størstedelen af funktionaliteten er implementeret.

## Abstract

This report describes the development of a 4th semester project at IHA concerning the design and development of a self-driving car, AutoCar. AutoCar has a DC-motor for propulsion, a servo for directional control, a sonar sensor used for ranging, a road sensor in each side to detect road markings, two battery sensors to detect and display battery levels, a tachometer to measure the speed of AutoCar, and a lot of internal logic, regulation, and communication to make it all work. The user can use the interface to start and stop AutoCar, print the status from the sensors, set the speed, and a request overtaking. AutoCar is itself responsible for driving within the road markings.

AutoCar is developed with a PSoC 4 and a Raspberry Pi 2b, which collectively function as the control unit. The GUI is designed with QT Creator in a Linux terminal.

The development process has been influenced by iterative tools such as SCRUM and the V model, and a strong focus on a bottom-up approach to the design process. The ASE Development Model has been used for chronological planning, and redmine along with SmartGit has been used to keep track of project files and resources. The use of these models and tools has ensured a competently developed system.

# Indhold

<b>Indhold</b>	<b>2</b>
<b>1 Kravspecifikation</b>	<b>3</b>
1.1 Aktørbeskrivelse . . . . .	3
1.2 Fully dressed use cases . . . . .	5
1.3 Ikke-funktionelle krav . . . . .	13

# 1 Kravspecifikation

Kravene til produktet er prioriteret ved brug af MoSCoW metoden. Her er kravene for produktet inddelt i fire kategorier, hvor de vigtigste elementer er prioriteret højest. **Must** benævner de krav som er vigtigst at opfylde, og som er absolut nødvendigt for produktet. **Should** er de krav produktet bør opfylde. **Could** er kravene som produktet evt. kunne opfylde, hvis projektets tidsramme tillader det. **Won't** er krav som ikke vil blive opfyldt inden for projektets tidsrammer, men evt. kan tages med i senere iterationer.

Følgende opdeling viser kravene udvalgt for dette projekt:

- Must**
  - Holde konstant udgangsstrøm og -spænding
  - Have stabil regulering
  - Ikke påvirke andre moduler ved fejl
  - Konstrueres med EEE komponenter
- Should**
  - Have programmerbar udgangsstrøm og -spænding
- Could**
  - Have overstrømsbeskyttelse på udgangen
- Won't**
  - Indeholde galvanisk adskillelse

## 1.1 Aktørbeskrivelse

I det følgende afsnit beskrives systemets aktører. Ved hver aktør angives typen, samt en kort beskrivelse af aktørens funktion og/eller hvordan de påvirker systemet.

### 1.1.1 Aktør: Bruger

**Type:**

Primær

**Beskrivelse:**

Brugeren interagerer med systemet via et interface.

Han kan indstille den ønskede fart, samt kontrollere den nuværende fart og spændingen på batterierne

## 1.2 Fully dressed use cases

### 1.2.1 Use case 1 - Start bil

**Mål:**

Initiere bilen så den er klar til kørsel og er klar til at modtage input

**Initiering:**

Brugeren

**Aktører:**

Brugeren (primær)

**Referencer:**

Ingen

**Samtidige forekomster:**

En

**Forudsætning:**

Bilen er slukket og der er forbindelse fra interface til bil

**Resultat:**

Bilens sensorer er tændt, motorer er klar, bilen holder stille

**Hovedscenarie:**

1. Brugeren vælger via interface "Start bil"
2. Bilen monitorerer sensorinputs og rapporterer status
3. Bilen udfører motortjek ved at køre bilen lidt frem og derefter tilbage
4. Bilen rapporterer status
5. Bilen tænder for- og baglys, blinker med blinklys hvis status er OK

**Extension 1: Status ikke OK**

6. Bilen afventer brugerinput

**Extensions:****Extension 1:** Status ikke OK

1. Bilen rapporterer fejl og forsøger at angive hvilken sensor og/eller motor der fejler

### 1.3 Ikke-funktionelle krav

I dette afsnit beskrives de ikke-funktionelle krav. Her opstilles f.eks. krav om præcision, brugervenlighed samt produktets dimensioner.

- Bilens længde og bredde må ikke overskride 50cm x 30 cm
- Brugeren skal have mulighed for at kommunikere med bilen via tekst-terminal og / eller grafisk brugergrænseflade
- Bilen skal detektere objekter indenfor intervallet 20cm til 2,5m
- Bilen skal kunne køre med en fart på 13 km/t  $\pm$  0,5
- Bilen skal have en maksimal vægt på 2kg
- Bilen skal måle sin fart i intervallet 0-13 km/t  $\pm$  0,5
- Bilen skal kunne indstille sin fart i intervallet 0-13 km/t med en opløsning på 0,5 km/t