

Styresystem for fremdrift av Shell-Eco-marathon-kjøretøy

Utvikling og test av cruise control og menneske-maskin-interaksjon

Jon Martin Harstad Bakken

Master i teknisk kybernetikk

Oppgaven levert: Juni 2009

Hovedveileder: Jo Arve Alfredsen, ITK

Oppgavetekst

NTNU deltar våren 2009 for andre gang i Shell Eco-marathon med en bil i Urban Concept-klassen. Bilen som ble utviklet og bygget våren 2008 skal forbedres og målet er å øke rekkevidden fra 729 km til 1000 km på en energimengde tilsvarende én liter bensin. Oppgaven inngår i et samarbeid med øvrige studenter i Shell Eco-marathon-gruppen ved NTNU. Denne delen av prosjektet omfatter utvikling av et styresystem for bilen med fokus på følgende funksjoner:

- Gjøre det mulig for føreren å gi motorpådrag både gjennom gasspedal og cruise control
- Gi fører informasjon gjennom en skjermløsning
- Styre lys, horn og andre elektriske komponenter i bilen
- Åpne for muntlig kommunikasjon mellom fører og resten av laget

Generelt inkluderer oppgaven omfatter følgende punkter:

- Utarbeide kravspesifikasjon og designspesifikasjon for ovennevnte styresystem
- Detaljspesifikasjon og implementasjon av dedikert maskin- og programvare
- Test, diskusjon av resultater og dokumentasjon

Oppgaven gitt: 12. januar 2009

Hovedveileder: Jo Arve Alfredsen, ITK

Forord

Jeg føler meg heldig som har kunnet delta på NTNUs trolig mest tverrfaglige masteroppgave noensinne. Det har vært en svært lærerik prosess hvor jeg har fått innblikk i andre fagfelt i tillegg til arbeidet på mitt eget felt. Det har vært givende å kunne kombinere teori og praksis og sitte igjen med et fysisk produkt som bidro som en brikke i spillet for å ta verdensrekord.

Jeg ønsker å takke alle i gruppa for et godt samarbeid og innsatsen som er lagt ned for å nå våre felles mål. Det har vært et utrolig spennende og lærerikt semester sammen med dere!

Emilie Vallée har vært uvurdelig med gode innspill og erfaringer fra løpet. Veileder, Jo Arve Alfredsen, har bidratt med gode tips underveis. Personellet på komponentlageret har hjulpet til med bestillinger, etsing av kretskort og komponenter. Detlef Blankenburg, Knut Einar Aasland og Asbjørn Solberg ved Institutt for produktutvikling og materialer har bidratt med god veiledning både ved kontoret i Perleporten og under løpet i Tyskland.

En spesielt stor takk til Camilla Aabakken som fortjener ros for å ha vært tålmodig og forsåelsesfull når kjæresten har jobbet til langt på natt.

Jon Martin Harstad Bakken

Trondheim, 26. juni 2009.

Innholdsfortegnelse

Forord	I
Innholdsfortegnelse	II
Sammendrag	VI
1 Innledning	1
1.1 Bakgrunn	1
1.2 Årets gruppe	1
1.3 Arbeidsprosessen	2
1.4 Målesetting	3
1.5 Strategi	4
1.6 Rapporten	4
2 Drivlinje	5
2.1 Akselerasjon av motor under trilling	6
2.2 Motorstyring	7
3 Fremdriftstyring	10
3.1 Gasspedal	10
3.2 Cruise Control	11
3.3 Skjermløsning	13
4 Brukergrensesnitt	15
4.1 Definerings av bruker	15
4.2 Heuristiske brukergrensesnittprinsipper	15
4.3 Test av brukergrensesnitt	16
4.4 Kommunikasjon med føreren	19
5 Implementering	22
5.1 Design av kretskort	22
5.2 Demping av knappespenningssvingninger	26
5.3 Produksjon av kretskort	28
5.4 Lys og horn	30
5.5 Batteriløsning	34
5.6 Kabling	35
6 Resultater og diskusjon	37
6.1 Fremdriftstyringen	37
6.2 Kommunikasjon med fører	37

6.3	Batteriløsningen	37
6.4	Konkurransen	37
6.5	Brukergrensesnitt	38
7	Konklusjon	40
8	Videre arbeid	41
9	Referanser og kilder	43
10	Vedlegg	45
10.1	Vedlegg A: Pakker for testing av motor.....	45
10.2	Vedlegg B: Check list før start	46
10.3	Vedlegg C: Utdrag av kildekode	47
10.4	Vedlegg D: Skjemategning for fremdriftstyring	55
10.5	Vedlegg E: Innholdsfortegnelse for vedlagt CD	56

Figurliste

Figur 1 Skisse på hvordan mine oppgavene har hengt sammen med de andres.	3
Figur 2 Endelig design på drivlinje og støttesystemer	5
Figur 3 Prinsippskisse for gasspedal.	10
Figur 4 Cruise Control i Jeep Grand Cherokee Laredo.....	11
Figur 5 Tilstandsmaskin for fremdriftssystemet.....	12
Figur 6 Skjerm under testing, EA DOG-M.	13
Figur 7 Papirprototyping	17
Figur 8 Rattet slik det så ut før oppgraderingen.....	23
Figur 9 Beregning av fysisk størrrelse på kretskortet.....	23
Figur 10 Utlegg av kretsen. Ringen nederst er staget til rattet.	24
Figur 11 Utlegg av kretsen for skjermen.....	24
Figur 12 Kretskortet montert på rattet i bilen.....	25
Figur 13 Forsiden av rattet.	25
Figur 14 Spenning over knapp uten debounce	26
Figur 15 Skjemategning for debouche-krets	27
Figur 16 Kompakt debounce-krets for cruise-control-knappene.	27
Figur 17 Lysbord for belysning av kretskort.....	28
Figur 18 Lutbad for å vaske bort belyst belegg.....	29
Figur 19 Oppvarmet syrebad for å etse vekk kobber	29
Figur 20 Ferdig etset og vasket kretskort	30
Figur 21 Oppkobling av horn, relé og bryter på rattet.	31
Figur 22 Fysisk oppkobling av LEDs, motstand, BuckPuck og LOGO!.....	32
Figur 23 Logisk styring implementert på Siemens LOGO! er beholdt fra 2008.	33
Figur 24 Krets for lysstyring. Motstanden til høyre var ikke i bruk.	33
Figur 25 Batteri på 12 V og 2,2 Ah.....	35
Figur 26 Kretsskjema for batteri, hovedbryter og sikringer.....	35

Tabelliste

Tabell 1 Pakker i PECC-COM	8
Tabell 2 Pedalvinkel ved manuell kjøring.....	10
Tabell 3 Pedalvinkel i cruise control-modus.	12
Tabell 4 Vurdering av egenskapene til kommunikasjonsalternativene.....	20
Tabell 5 Strømforbruk i komponentene	34
Tabell 6 Farger på ledere til lys i halen	36

Sammendrag

Shell Eco-marathon er en årlig konkurranse for lag fra hele Europa. Denne rapporten følger arbeidet med å bygge et styresystemet i bilen for NTNUs deltagelse i 2009. Dette er NTNUs andre deltagelse. Bilens karosseri har blitt beholdt og det meste av innmaten er nytt. Den deltagende gruppen har bestått av ti studenter fra fem forskjellige institutter.

Det har blitt laget et fremdriftssystem som sender hastighetsreferanser til en motorstyring. Dette systemet består av gasspedal, styrekort, skjerm og batterier. Fremdriftssystemet kan kjøres manuelt eller ved hjelp av en cruise control. Sjåføren får informasjon fra systemet på en skjerm og kommuniserer via mobil til hjelpemannskap ved banen. Det har vært fokusert på brukervennlighet gjennom utviklingsprosessen.

I løpet av semesteret har bilen blitt designet og bygget. Alle komponenter har blitt forbedret og virkningsgraden på systemene har økt markant. Dette medførte at Shell Eco-marathon 2009 ble en stor suksess for NTNU.

Nettsiden www.fuelfighter.no inneholder mye informasjon om prosjektet.

1 Innledning

Shell Eco-marathon er en årlig konkurranse som arrangeres for høyskoler og universitet fra hele Europa. Det konkurreres i to klasser: Prototype og Urban Concept. I Prototype står lagene fritt til å velge design og komponenter. I Urban Concept stilles det strengere krav til bilens dekk, lys, komfort, fysiske størrelse, bagasjeplass og kjørestil. Det var 147 deltagende biler i 2009, hvor rundt 30 deltok i Urban Concept.

1.1 Bakgrunn

NTNU deltok for første gang i 2008 med bilen Pure Choice og tok en sterk andreplass ved å kjøre 729 km/l¹ som var rett bak vinneren De Haagse Hogeschool fra Nederland med 789 km/l. Konkurransen foregikk ved Nogaro Racing Circuit i Frankrike 21. – 24. mai. Pure Choice var en hydrogenbil med et selv bærende karbonfiberkarosseri.

Fjorårets lag gjorde en formidabel jobb med å utvikle bilen fra bunnen. På grunn av tidspress ble bilen ikke ferdig før rett før konkurransen og det var blitt gjort lite testing på aerodynamikk og energieffektivitet i totalsystemet. Fokuset var hele tiden vekt og det resulterte i tidenes letteste bil i Urban Concept med 69 kg uten fører.

Urban Concept-konkurransen går ut på å kjøre 22 km på maksimalt 53 minutter. Dette gir en minimums gjennomsnittshastighet på 25 km/h. Underveis skal det gjennomføres tre stopp - hvert på 10 sekunder. Konkurransen skal fra og med 2009 holdes på EuroSpeedway Lausitz i Tyskland. Banen er 3173 meter lang og det blir kjørt syv runder.

1.2 Årets gruppe

Shell Eco-marathon er et unikt prosjekt ved NTNU siden det går på tvers av instituttgrensene. Prosjektet er styrt av studentene selv, men organiseres av Institutt for produktutvikling og materialer. Gruppesammensettingen er basert på en kombinasjon av behov og lyst. Det ble sendt ut forespørsler til aktuelle linjer og interesserte meldte seg. Gruppen har i 2008/2009 bestått av:

¹ Forbruket av hydrogen omregnes til tilsvarende energimengde i bensin og oppgis i antall kilometer per liter bensin. Dette er for å ha en felles og lett forståelig måleenhet i konkurransen uansett drivstoff.

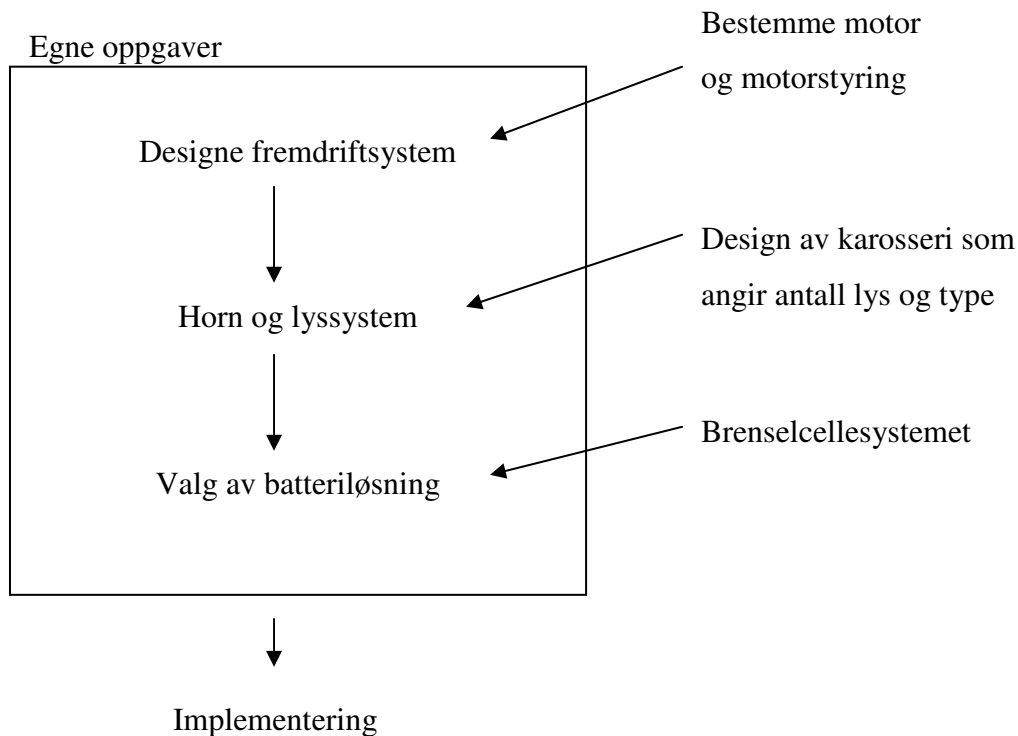
Navn	Linje	Ansvarsområde
Audun Storaker	Produktutvikling og materialteknikk	Hjuloppheng
Anders Gellein	Energi & Miljø	Brenselcellestyring og motorstyring
Magnus Christian Ebbesen	Ingeniørvitenskap og IKT	Prosjektleder og 3D-modellering
Hanne Flåten Andersen	Industriell kjemi og bioteknologi	Brenselcelle
Stian Christensen	Produktutvikling og produksjon	Hjuloppheng
Daniel Vik Skogen	Produktutvikling og Materialer	Kompositt og felger
Rolv Marius Faleide	Energi & Miljø	Testing og produksjon av motor
Bård Stærkebye Pettersen	Produktutvikling og materialer	Bremser og hjul
Øyvind Frotjold Birkeland	Produktutvikling og produksjon	Hjul, ventilasjon og 3D modellering
Jon Martin Harstad Bakken	Teknisk Kybernetikk	Styresystemer for fremdrift

- Sjøfører og reservesjøfører, Emilie Vallee og Hege Bache, har kommet med synspunkter og vært testpersoner under arbeidet.
- Tre studenter ved Industriell design har laget grafisk profil og standmateriell.
- Prosjektet har også en egen EiT-gruppe som har bidratt mer perifere oppgaver, som tilhenger og animasjoner. Gruppeoppgavene er nærmere forklart i grupperapporten *DNV Fuel Fighter - Technical and organizational support systems* [6]

1.3 Arbeidsprosessen

Det har hele tiden vært et tett samarbeid i gruppa, mange av oppgavene er overlappende og alle har vært avhengige av hverandre. Figur 1 viser overordnet hvordan mine oppgaver har

hengt sammen med de andres. Gruppen har hatt en flat organisering hvor alle har hatt konkrete ansvarsområder med bilen og mer perifere og organisatoriske oppgaver.



Figur 1 Skisse på hvordan mine oppgavene har hengt sammen med de andres.

Januar og februar var preget av teoretisk arbeid med modelleringer, design og planlegging. Mars og april ble mye tid brukt til å lage de fysiske komponentene. Tiden etter påske var satt av til testing, men ble spist opp av pr-event og iherdig jobbing med å få bilen ferdig. Arbeidsoppgavene overlappet hverandre, og på det mest intense stod for eksempel hele gruppa og pusset støpeformen til den nye halen.

1.4 Målesetting

Førrige gruppe satt ut en utfordring om å passere 1000 km/l. Denne utfordringen ble tatt. Da det er vanskelig å si hvordan andre lag vil gjøre det i konkurransen, ble det unngått å si noe om endelig plassering. Det har allikevel hele tiden vært et ønske om pallplassering.

Anders Gellein utførte i sin prosjektoppgave [4] høsten 2008 en detaljert simulering som tok høyde for ny baneprofil, økt vekt og planlagte forbedringer i aerodynamikk og drivlinje. Modellen ble verifisert ved å sammenligne med fjorårets bil og løp. Det ga en teoretisk

kjørelengde på 1200 - 1250 km/l. På grunn av usikre parametre i modellen og for å ikke gi falske forhåpninger ovenfor sponsorer, ble 1000 km/l det endelige målet.

1.5 Strategi

For å nå målsetningen måtte bilen forbedres på alle områder. Spesielt viktig var aerodynamikk og drivlinjen. Andre fokusområder inkluderte hjuloppheng, bremsesystem, felger og kjørestrategi. Styring, ratt og frontlykter skulle beholdes, mens karosseriet skulle modifiseres. Ved nyanskaffelser ble de prioritert på følgende måte:

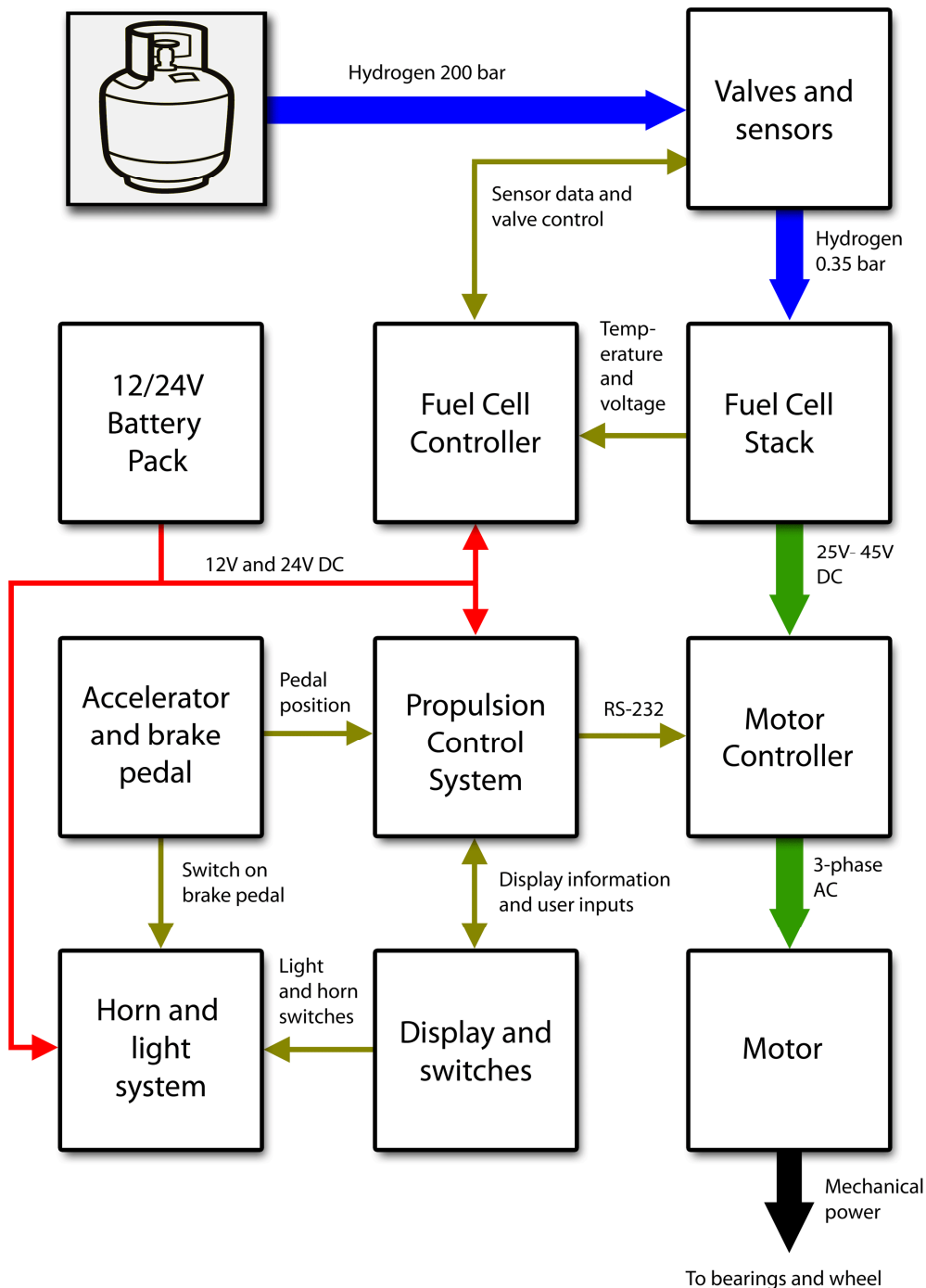
1. Effektivitet
2. Vekt
3. Pris

Effektiviteten ble prioritert på bekostning av vekt fordi bilen allerede var utrolig lett; det som manglet var høy effektivitet i drivlinja. Gode sponsorer ga gode økonomiske forhold og man stod fritt til å velge kostbare løsninger, selvsagt innenfor rimelighetens grenser.

1.6 Rapporten

Rapporten skal fungere som en dokumentasjon av arbeidet som er gjort, en bruksanvisning av bilen og hjelp på veien for fremtidige grupper. Diplomoppgaven har gått ut på å lage et grensesnitt mellom fører og bil som sørger for fremdrift, lys/horn og tilbakemeldinger om hastighet. Rapporten tar i hovedsak for seg dette arbeidet, men arbeidet med gruppa har også inkludert mange andre oppgaver. Viktigst har det vært å lage og drifte nettside og planlegge logistikk på arrangementene i løpet av semesteret.

2 Drivlinje



Figur 2 Endelig design på drivlinje og støttesystemer

Energikretsen som begynner med hydrogen og ender med at bilen kjører framover, kalles *powertrain*, eller drivlinje på norsk. Høsten 2008 ble det valgt å beholde samme grunnprinsipp i bilen, med hydrogen brenselcellestakk og elektromotor. Dette er beskrevet i Anders Gelleins prosjektoppgave [4]. Bilens drivlinje består av:

1. Hydrogensystem med tank, ventiler og målinger.
2. Brenselcellestack og styring av denne
3. Lastrelé og diode
4. Motorstyringsenhet
5. Motor
6. Mekanisk overføring til hjul

Siden konkurransen går ut på å bruke minst mulig drivstoff, spiller punktene 2 til 6 en viktig rolle. Denne rapporten tar for seg styringen av motorstyringsenheten. En optimal styring av motorstyringen vil ha stort utslag på totalvirkningsgraden. Totalsystemet, som vist i Figur 2, inneholder også styresystem for brenselcelle. Brenselcellestyringen er dekket av Anders Gelleins masteroppgave [3].

2.1 Akselerasjon av motor under trilling

Den mekanisk svakeste komponenten i fremdriftssystemet til bilen er frihjulet. Det er plassert mellom motor og hjul for å kunne ha en tilnærmet friksjonsfri trilling. Motoren har et tomgagstap på 0,5 Nm (Rolv Marius Faleides masteroppgave [2]) ved at magnetene sleper med seg et magnetfelt som går gjennom jernet. Tapet er et virvelstrømstap og hysterestap i jernet under rotasjon.

Etter avsluttet trilling vil ikke frihjulet gripe før motoren har større hastighet enn hjulet. Dersom akselerasjonen av motoren er høy når frihjulet griper, blir det en høy momentpuls og man risikerer at frihjulet ryker. Det er to måter å unngå dette; enten å redusere akselerasjonen rett før hjulet griper eller å innføre en akselerasjonsbegrensning. For å kunne redusere akselerasjonen rett før hjulet griper, må styresystemet vite hjulhastigheten. Hjulene har ingen gjennomgående aksling og hastighetsmåling må gjøres enten på bremseskivene eller direkte på dekket. Det er imidlertid et tøft miljø i hjulbuene med vibrasjoner, fuktighet og skitt, som gjør dette vanskelig.

Toleransen på frihjulet i bilen ble maksimert ved å gå for FKNN 6206 fra GMN, som tåler et moment på 148 Nm. Rolv Marius Faleide gjorde en Simulink-simulering som viste at man oppnår tilstrekkelig akselerasjon ved å ha et motormoment på 20 Nm.

2.2 Motorstyring

Motorstyring eller *inverter*, gjør om DC-strøm fra brenselcellestacken til en tidsvarierende strøm (AC) hvor frekvensen og spenningen får motoren til å gå rundt med ønsket turtall og moment.

Valg av enhet

Valget av motorstyringsenhet ble gjort samtidig med valget av motor. Denne delen av prosjektet er dokumentert i Rolv Marius Faleides masteroppgave [2] og blir her bare kort oppsummert:

Motoren er laget for AUV-en Hugin og har med sine 17 polpar en rotasjonshastighet som tillater den å være tilkoblet hjulakslingen direkte uten gir. Motoren ble viklet om for å ha riktig antall viklinger og maksimal kobbertetthet i stator. Resultatet ble en stor effektivitetsøkning i forhold til fjorårets løsning hvor en mindre og raskere motor var tilkoblet via gir på navet.

SmartMotor leverte motorstyringen SMIK48V sammen med Hugin-motoren. Styringen var en relativt kompakt enhet, og tilbød en rekke funksjoner som har vært nyttige under testingen av systemet.

- Temperaturmåling på kjøleelement og PCB
- RS-232-kommunikasjon
- Regenerativ bremsing.

Styringsenheten ble omprogrammert av SmartMotor for å oppfylle våre krav. Maksimalt dreiemoment ble satt til 20 Nm for å begrense strømmen fra brenselcella og å beskytte frihjulslageret mot for store påkjenninger. I tillegg ble spenningsarbeidsområdet økt fra 32-48V til 18-48V, da det under testing ble oppdaget store spenningsfall fra brenselcellestacken.

Den regenerative bremsingen viste seg å være utført i hardware og kunne ikke skrues av via software. Dermed ble det nødvendig å gjeninnføre dioden mellom brenselcella og motorstyringa for å beskytte brenselcella mot strømmen som kommer når bilen dyttes bakover. Denne dioden har et spenningsfall på 0,7-1V. Dette gir et tap på

$$P_{\text{tap}} = U_{\text{diode}} \cdot I_{\text{avg}} = 1 \text{ V} \cdot 2 \text{ A} = 2 \text{ W}. \quad (1)$$

Med en antatt totaleffekt ved kjøring på 100 W utgjør tapet omtrent 2 %. Planen var å kjøre et av løpene uten denne dioden, noe som ikke ble gjort da vi bare fikk ett løp.

Styring av motorstyring

SMIK48V kommuniserer med omverdenen gjennom RS-232 og protokollen PECC-COM. Protokollen er dårlig dokumentert og skriftlig materiale var utdatert, så det trengtes mye hjelp fra SmartMotor. Skriftlig dokumentasjon finnes i databladet [14]. Enheten både mottar og sender pakker. Pakkene, som vist i Tabell 1, består av data innkapslet med en header og feilforhindrende koder.

Tabell 1 Pakker i PECC-COM

Start	Lengde N	Checksum	Data1	...	DataN	CRC-16
-------	----------	----------	-------	-----	-------	--------

Pakkene starter alltid med 0xFF og en header. Alle andre steder 0xFF opptre, blir 0xFF kodet som to etterfølgende bytes med verdi 0xFF. Lengden N sier hvor mange bytes data som sendes. Headeren avsluttes med en checksum på 8 bits som beregnes etter formelen

$$\text{Checksum} = \text{!} (0\text{xFF} + \text{Lengde}) + 1 \quad (2)$$

Ved mottak summeres *Start*, *Lengde N* og *Checksum*. Bli summen noe annet enn 0x00, er header ikke inntakt². Etter headeren kommer selve dataene og til slutt en CRC-16-verdi for payloaden.

Kommandoer

Datadelen av pakkene inneholder kommandoen som skal utføres. Formatet på disse er beskrevet i *Hugin motor software user's guide* [15] og dokumentasjonen til PECC-COM [14].

² En korrekt sjekksum er ingen garanti for inntakte data, men den er med på å sannsynliggjøre en korrekt transmisjon.

Verdier er angitt i Q.20 format. Det vil si at de 20 laveste bitene brukes til desimaldelen og resten brukes til heltallsdelen og fortegn. Beregning av Q-20-verdier gjøres etter formelen

$$\text{Q.20-verdi} = \text{pu-verdi} * 2^{20} \quad (3)$$

For eksempel 1,45 blir da

$$\text{Q.20-verdi} = 1,45 * 2^{20} = 1520435 \quad (4)$$

Representert heksadesimalt med little-endian blir verdien {0x33, 0x33, 0x17, 0x00}.

Mye brukte kommandoer er:

- Ping motor: {0x01, 0x00}
- Start: {0x02, 0x00}.
- Sett hastighetsreferanse til 0 pu³: {0x10, 0x02, 0x3F, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}. De siste fire bytene angir antall pu i Q.20-format og little-endian.
- Stopp: {0x04, 0x00}.

For hver av kommandoene sender SMIK48V et svar som inneholder informasjon om hvilken kommando som er utført og eventuelle feilmeldinger.

Testoppsett

Terminalprogrammet *Br@ys terminal* [22] ble brukt under testing av motoren. Pakkene ble laget manuelt og sendt fra en pc via en D-SUB-kabel til motorstyringen. Nettjenesten *On-line CRC Calculator* [19] ble benyttet for å beregne CRC-16-verdier. En typisk test går ut på å starte motorstyringens interne program, starte motor og angi hastighetsreferanser. Pakkene som har blitt brukt er angitt i Vedlegg A.

³ pu-verdi: Per Unit. Vanlig uttrykk innen kraftelektronikk hvor verdiene angis i en faktor av base-verdier. 1.0pu tilsvarer dermed merkeverdien.

3 Fremdriftstyring

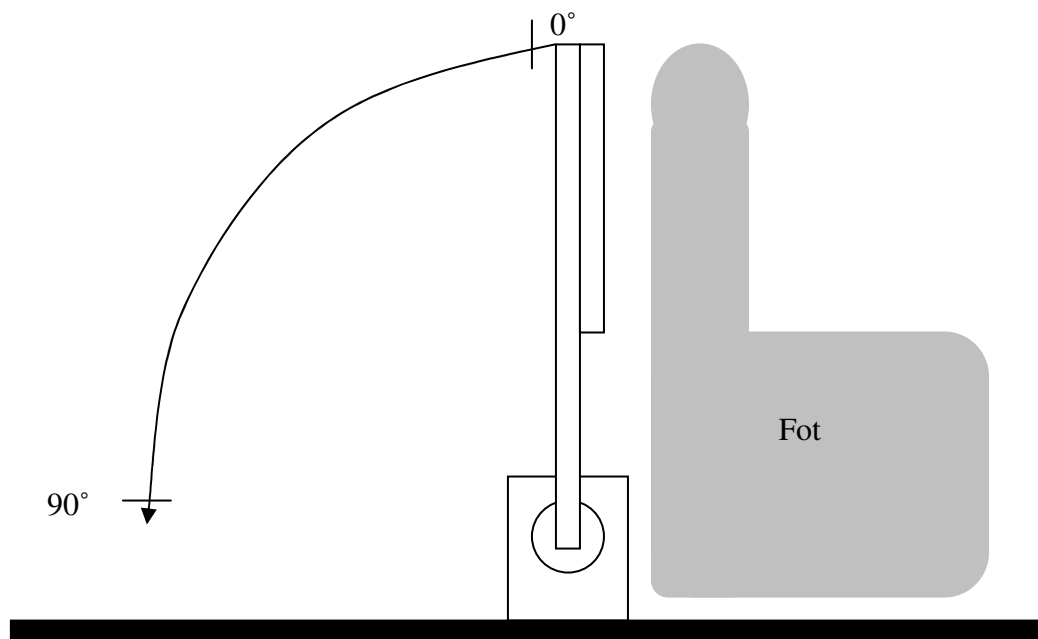
3.1 Gasspedal

Føreren kjører bilen ved hjelp av en gasspedal. En skisse for denne er gitt i Figur 3. I akslingen mot gulvet sitter et potmeter som forsynes med 5 V. En spenning kan da leses ut og tilsvarer vinkelen til pedalen.

Siden motorstyringen tar inn hastighet som referanse og ikke et moment må fremdriftstyringen integrere opp verdiene som sendes fra gasspedalen. Dette er utført ved å dele opp vinkelområdet i flere deler. Hvert område har sin funksjonalitet, som vist i Tabell 2

Tabell 2 Pedalvinkel ved manuell kjøring

Omtrentlig vinkel på gasspedal	Funksjon
0° - 10°	Hastighetsreferansen nullsettes
10° - 20°	Hastighetsreferansen senkes omvendt proporsjonalt med vinkelen.
20° - 40°	Hastighetsreferansen holdes stabil
40° - 70°	Hastighetsreferansen øker proporsjonalt med vinkelen med 40° som bias.



Figur 3 Prinsippskisse for gasspedal.

3.2 Cruise Control

Cruise controlen er den sentrale delen i fremdriftsystemet. For å gjøre betjeningen intuitiv ble virkemåten inspirert av cruise controlen i en vanlig bil. Der er det knapper som skrur av og på cruise controlen og knapper for å øke og senke innstillt hastighet. Se Figur 4.



Figur 4 Cruise Control i Jeep Grand Cherokee Laredo. (Bildet hentet fra Wikipedia, og er under GNU Free Documentation License.)

Reglene i konkurransen spesifiserer at bilen må kjøres ved hjelp av en gasspedal. For å oppfylle denne regelen, vil gasspedalen under kjøring med cruise control fungere som en dødmannsbytter. Når føreren slipper pedalen, vil motoren stanse. Bilens cruise control avviker dermed fra bransjestandarden hvor gasspedalen vil aksellere bilen også når cruise controlen er aktivert.

To verdier angir hastighet:

- **Motorhastigheten:** Dette er den faktiske referansen som sendes motoren
- **Cruise-hastigheten:** Når cruise controlen er aktivert settes motorhastigheten lik denne.

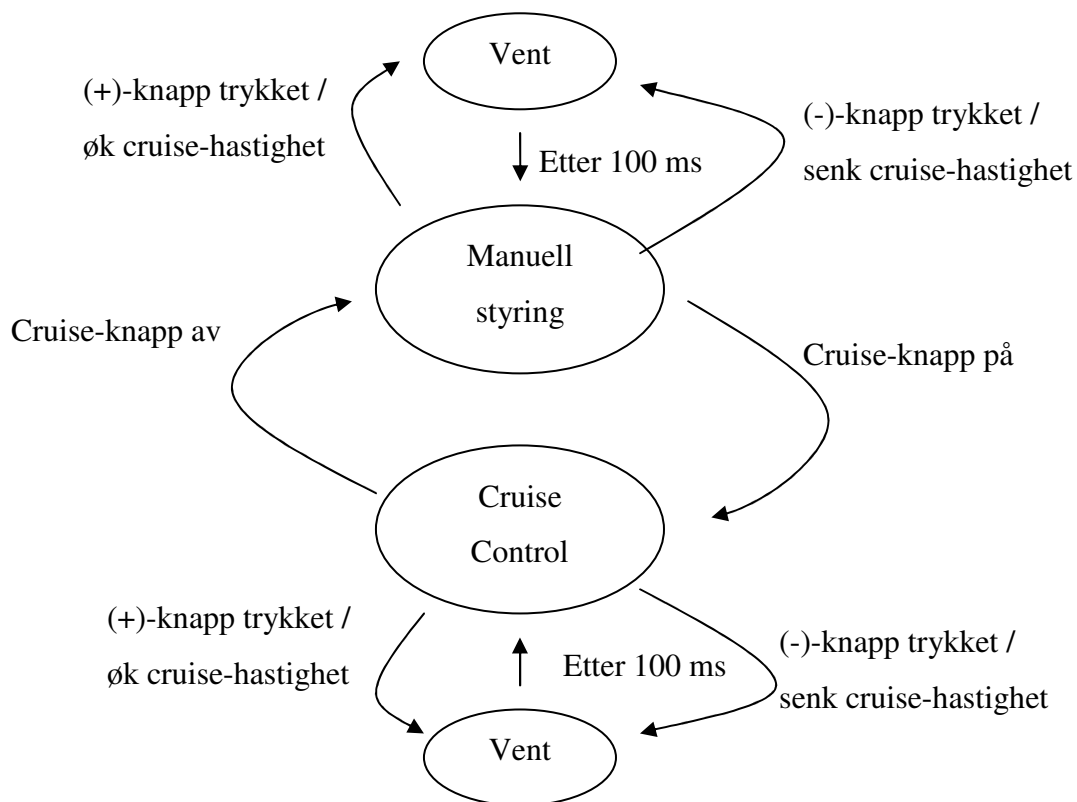
En knapp på hver side av rattet gir føreren mulighet til å endre den innstilte cruise-hastigheten. I et tidlig utkast av systemet tar cruise controlen motorhastigheten i øyeblikket

idét den blir aktivert og holder denne verdien. Siden cruise controlen stort sett brukes i området 20-30 km/h, er det bedre at cruise-verdien holdes fast, uansett om cruise kontroll er aktivert eller ikke. Da kan føreren ved start skru på cruise controlen innstillt på for eksempel 25 km/t og la den stå for aksellering, fremfor å aksellere ved hjelp av gasspedal.

Tabell 3 Pedalvinkel i cruise control-modus.

Omtrentlig vinkel på gasspedal	Funksjon
0° - 10°	Hastighetsreferans settes til null
10° - 90°	Hastighetsreferansen tilsvarer cruise-hastighet

Tilstandsmaskin for å endre hastighet og aktivere/deaktivere cruise controlen er vist i Figur 5.



Figur 5 Tilstandsmaskin for fremdriftssystemet

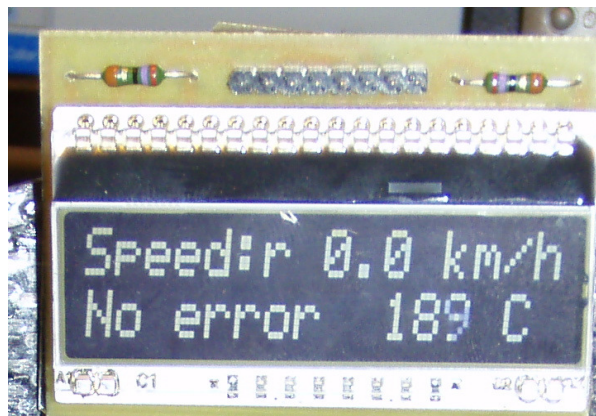
3.3 Skjerm løsning

En stor mangel i fjor var tilbakemelding fra systemet om gitt hastighet. Føreren var overlatt til kun GPSen for å avlese motorhastighet. Det ble utarbeidet en prioritert liste over informasjon som var ønskelig å presentere for brukeren:

1. Hjulhastighet
2. Motorhastighet
3. Innstilt cruise-hastighet.
4. Feilmeldinger og cruise-tilstand.
5. Tidtaking.

For å begrense informasjonen som gis brukeren ble tidtaking fjernet. GPSen beholdes siden den gir en god og uavhengig hastighet på bilen. Å vise hjulhastighet på skjermen er dermed overflødig.

Valget falt på en EA DOG-M-skjerm (Figur 6) på grunn av tilgjengelighet og spesifikasjoner. Skjermen har to linjer med 16 tegn og kobles til en mikroprosessor via SPI eller prallel. Skjermen leveres i flere fargekombinasjoner. Hvit tekst på svart bakgrunn ble valgt for høyest mulig kontrast.



Figur 6 Skjerm under testing, EA DOG-M.

På bakgrunn av MMI⁴-testing ble det besluttet at skjermen skal vise motorhastighet og cruise-hastighet. Ledig plass i høyre hjørne ble brukt til å vise vinkelposisjon til gasspedal og følgende systemtilstander:

⁴ Menneske-Maskin-interaksjon

- **D:** Decrease, senke-knappen for cruise controlen er trykket inn
- **C:** Cruise controlen er aktivert ved at vippebryter er i opp-posisjon.
- **I:** Increase, øke-knappen er trykket inn.

4 Brukergrensesnitt

Det har blitt lagt vekt på et godt brukergrensesnitt overfor føreren. Bevisste valg i utviklingsprosessen bidrar til å gjøre kjøreopplevelsen enklere for sjåføren.

4.1 Definerings av bruker

Brukeren i et MMI-system er ofte vanskelig å konkretisere fordi det er mange forskjellige typer brukere. Slik er det ikke i denne bilen. Her er en konkret sjåfør med kjente egenskaper og kunnskapsnivå.

Emilie Vallée:

- 23 år, fra Frankrike, snakker engelsk godt, litt norsk.
- Studerer Biofysikk og medisinsk teknologi.
- Har tidligere vært sjåfør for et fransk lag i Shell Eco-marathon og kjenner dermed konkurransen og kjøreforholdene godt.

Miljøvariable er forhold som opptrer når produktet skal brukes i et konkret miljø. I bilen gjelder disse forholdene:

- Fører må bruke hansker: Knapper og håndtak må være designet for dette.
- Føreren bruker hjem: Hjemmen bidrar til å minske utsikten og mest mulig av instrumenter og paneler må plasseres i området under frontruta. Bilen er lydløs, men hjelmen vil sammen med støy fra andre biler med forbrenningsmotor gjøre det vanskelig å høre lydsignaler. Man må derfor belage seg på visuelle indikasjoner fremfor lydbaserte.
- Føreren sitter i en lukket bil med varm kjøredress: Bevegeligheten er nedsatt og kontrollpaneler bør være plassert nær hverandre.
- Innsiden av bilen er tilnærmet svart og kombinert med sterkt sollys må panelene ha høye kontraster. LCD-skjermen må ha bakgrunnsbelysning.

4.2 Heuristiske brukergrensesnittprinsipper

Jakob Nielsen, presentert som en ekspert innen brukergrensesnitt i emnet TDT4180 MMI, har utarbeidet en liste over ti heuristiske brukergrensesnittprinsipper [20]. Disse ble brukt for å vurdere skjerm- og rattløsningen. Prinsippene er:

1. Vise systemstatus

Systemet bør alltid vise brukeren hva som skjer og gi tilbakemeldiger.

2. Koblingen mellom systemet og den reelle verden

Systemet bør være tilpasset brukeren og snakke brukerens språk.

3. Brukerkontroll og brukerfrihet

Brukere gjør innimellom feil, og må ha en enkel mulighet til å ”angre”, og gå tilbake til en tidligere tilstand.

4. Konsistens og standarder

Systemet bør følge plattformkonvensjonen og følge gjeldende standarder, slik at brukeren kjenner seg igjen.

5. Unngå feil

Det som er enda bedre enn å gi gode feilmeldinger, er å unngå at feilene skjer.

6. Gjenkjennelse fremfor pugging

Gjør objekter, hendelser og valg synlige. Brukeren skal slippe å huske informasjon. Instruksjoner for bruk av systemet bør være tilgjengelig til en hver tid.

7. Fleksibilitet og effektiv bruk.

Ekspertbrukere bør få snarveier som gjør det raskere å bruke systemet

8. Estetisk og minimalistisk design.

Dialoger bør ikke inneholde informasjon som er irrelevant, da den viktige informasjonen lettere forsvinner i mengden.

9. Hjelp brukeren til å gjenkjenne, diagnostisere og fortsette etter feil

Feilmeldinger bør bruke et enkelt språk, presist si hva som er feil og foreslå en løsning.

10. Hjelp og dokumentasjon

Systemet bør leveres med dokumentasjon og mulighet for hjelp underveis.

4.3 Test av brukergrensesnitt

For å verifisere at rattløsningen er brukbar, ble det gjennomført en MMI-test. Det finnes mange måter å teste et brukergrensesnitt avhengig av hvilken fase produksjonen er i. Ideelt sett skal utviklingsprosessen inkludere tester mellom hver versjon for å peke ut kursen. Det ble det ikke tid til i dette prosjektet, men det er heller ikke nødvendig med tanke på det begrensede antallet brukere og applikasjonens omfang.

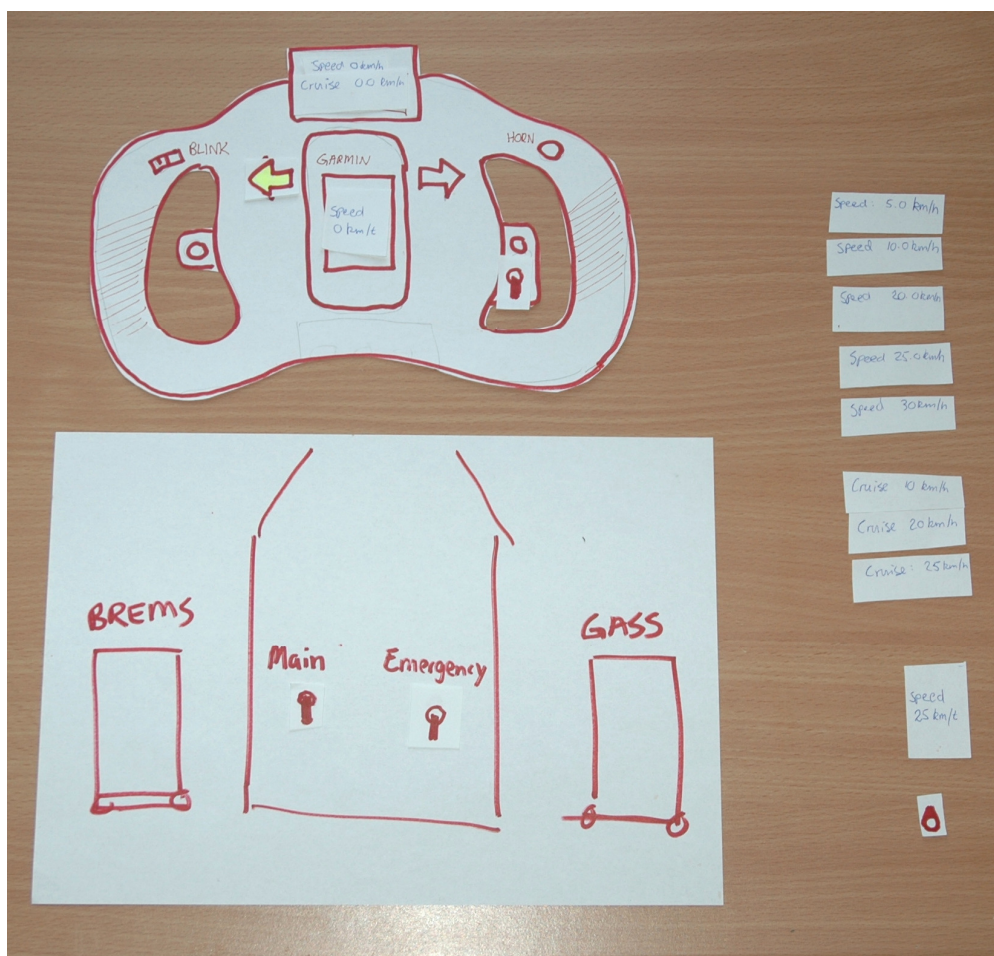
Papirprototyping

Papirprototyping er et godt valg i en tidlig fase før et fysisk produkt er klart eller når man ønsker en rask og effektiv test. Den går ut på å bruke papirmodell av produktet. Testpersonen ”trykker” på papiret der knapper er tegnet inn. Ved å bruke papirlapper som legges over skjermområdene, endrer produktet seg etterhvert.

Testplan

Testpersonen bør matche den virkelige brukeren. Personen får ved ankomst informasjon om hva testen går ut på, hva produktet skal brukes til og instruksjoner om å hele veien forklare hva man tenker. Ideelt sett skal det være tre personer som utfører testen i tillegg til testpersonen:

- En testleder som gir brukeren instruksjoner.
- En som holder orden på lapper og endrer skjermbildet utfra hendelsene.
- En referent som noterer seg hva brukeren gjør og tenker.



Figur 7 Papirprototyping

Oppgaver som ble testet:

1. Cruise Control-test 1

- Sett bilen i manuell modus og aksellerer til 25 km/t.
- Aktiver cruise-controll
- Senk hastigheten til 20 km/t
- Skru av motor for å trille.
- Brems ned og stans.

2. Cruise Control test-2

- Sett inn cruise-hastighet på 25 km/t og start motoren
- Når hastigheten er nådd, kjør på cruise-control i 15 sekunder
- Skru av motor for å trille.

3. Test av lys og horn

- Sett på hovedlys
- Blink til venstre i 5 sekunder
- Blink til høyre i 5 sekunder
- Aktiver hornet i 2 sekunder
- Skru av hovedlys
- Sett på nødblink i 10 sekunder

4. Multitasking

- Akseller bilen til 20 km/h ved hjelp av cruise control.
- Forbikjøring. Skru på blinklys til høyre og bruk hornet.
- Øk farten til 25 km/h
- Skru av blinklyset
- Senk farten til 10 km/h

Resultat

Det ble høstet flere erfaringer av testen:

- Brukeren foretrakk å slippe gasspedalen for å deaktivere cruise kontrollen fremfor å først vippe ned cruise-bryter og så slippe pedalen.
- Det tar tid å øke cruise-hastigheten til 25 km/t før start. Cruise-hastigheten bør dermed være ferdiginnstilt på rundt 25 km/t.
- I multitasking med forbikjøring ble det mye på en gang. Sjøføren har bedre trening i dette og det anses ikke som et problem.

- Manuell modus er ikke nødvendig. Føreren klarer fint å kjøre bilen kun på cruise-control.
- Styring av lys og horn fungerer fint.
- Føreren trenger ikke slippe taket på rattet under kjøring for å betjene knappene bortsett fra hoved- og nødlys. Disse er ikke i bruk under normal kjøring.

4.4 Kommunikasjon med føreren

Det er svært viktig å ha kommunikasjon med føreren under både testløp og konkurranse. Feil kan identifiseres og noen ganger rettes opp, men viktigst er å avlaste føreren med tidtaking og holde orden på stoppene underveis. To personer (*hjelpemannskapet*) kan under løpet stå i nærheten av banen og det er naturlig at det er disse som har kontaktet med føreren. Reglene gir ingen føringer på hvilken teknologi som benyttes, men radioforbindelser må følge tysk lov og man må søke om tillatelse for frekvensene man ønsker. Special Rules [17] sier i Article 27:

The use of hand-held communications devices is forbidden in all vehicles. However the use of fully voice-activated “hands-free” kits are allowed.

I hovedsak finnes det to muligheter; kommunikasjon med radio eller over mobilnettet. Tre alternativer ble vurdert:

Walkietalkie

Fjorårets gruppe gikk for denne løsningen. Fører og hjelpemannskapet har hver sin radio og kommunikasjonen er halv-duplex⁵. Radioene ble den gangen leid. Et headsett med ørepropp og mikrofon ble brukt av fører for å oppfylle kravene i Article 27 og samtidig gjøre det mulig å høre under hjelmen. Ulempene er noe lav lyd kvalitet og et komplisert tysk byråkrati for å få tillatelser i orden. Man risikere at flere bruker samme frekvens. En stor fordel er at fører ikke blir forstyrret unødvendig siden kommunikasjonen må initieres fysisk fra hjelpemannskapet.

Mobiltelefon med data og tale

Denne løsningen tok inspirasjon fra PAC-Car II som har tatt flere verdensrekorder i Prototype-klassen. Laget fra Eidgenössische Technische Hochschule i Zürich har dokumentert arbeidet sitt i boka ” The World’s Most Fuel Efficient Vehicle” [13]. Der benyttes Qtec 9090,

⁵ Halv-duplex. Toveiskommunikasjon, men bare en vei av gangen.

en kombinert mobiltelefon og PDA⁶ for å sende både data og kommunisere muntlig med føreren. Dette er en spennende løsning som åpner for diagnose og i noen tilfeller også fjernstyring av delsystemer i bilen. KISS-prinsippet⁷ taler mot denne løsningen sammen med en stor jobb med implementering og testing. Man vil måtte bruke datatrafikk i mobilnettet, noe som medfører en viss kostnad.

Mobiltelefon med tale

Bruk av vanlige mobiltelefoner i begge ender er en svært enkel og også anbefalt løsning fra arrangøren. Førerens mobil må ha handsfree-sett med ørepropp og mikrofon. I tillegg må mobilene ha batteritid for minst en time. Ved å kjøpe to tyske kontantkort kan utgiftene minimeres.

Tabell 4 Vurdering av egenskapene til kommunikasjonsalternativene

	Vekting	Walkie-talkie	Mobiltelefon med data og tale	Mobiltelefon med tale
Pris for innkjøp	10 %	90 %	0 %	100 %
Pris for bruk	5 %	100 %	0 %	50 %
Brukervennlighet	40 %	70 %	60 %	100 %
Krav til lisenser	15 %	0 %	100 %	100 %
Lav utviklingstid	20 %	95 %	0 %	100 %
KISS-faktor	10 %	95 %	20 %	100 %
SUM	100 %	71 %	41 %	98 %

Vurdering

Egenskapene til de tre løsningene ble veid opp mot hverandre som vist i Tabell 4. Hver egenskap fikk en prosentvis vektning som tilsier hvor viktig denne egenskapen er.

Alternativene fikk en poengsum hvor 100 % er best. Poengene og vektningen er multiplisert og til slutt summert. Egenskapene som ble vurdert er pris for innkjøp og bruk. Med prosjektets gode økonomiske rammer utgjorde dette kun 15 % til sammen. Brukervennlighet er svært viktig, da føreren bør holde fokus på kjøringen. Utviklingstid ble vektet hele 20 %, da tidsforbruk var en viktig faktor hele veien. KISS-faktoren er tatt med for å holde kompleksiteten nede. Jo mer komplekst system, jo mer kan gå galt.

⁶ PDA - Personlig Digital Assistent, en lommedatamaskin.

⁷ KISS-prinsippet – Keep It Simple, Stupid. Prinsipp for å holde ting enkelt.

Konklusjon

Mobiltelefon med tale ble en klar vinner og eneste store ulempene med denne løsningen er om samtalen brytes. Da må føreren svare på oppringningen på nytt under kjøring, men dette er lite sannsynlig.

5 Implementering

Atmels AVR-serie består av mange mikrokontrollere som er egnet for å styre fremdriftssystemet. Brikkene har UART, ADC og SPI innebygget. AVR-biblioteket har innebygde funksjoner for å beregne CRC-16 i <util/crc16.h>.

5.1 Design av kretskort

Skjemategninger og kretsutlegg har blitt laget i Cadsoft Eagle. Dette programmet ble valgt på grunn av et enkelt brukergrensesnitt og stor utbredelse. Dette sikrer enkel tilgang til ferdige komponentbibliotek. Skjemategninger finnes i Vedlegg D og på CD. Kretsen inneholder:

- 5 V spenningsregulator.
- Plass for 3.3 V spenningsregulator for alternativ skjerm.
- Atmega168 mikrokontroller.
- Knapp for reset.
- ISP-header for programmering.
- MAX232 for RS-232-kommunikasjon med motorstyring.
- 9-pins D-SUB-kontakt kabel til motorstyring.
- 8-pins header for tilkobling av skjerm.
- Header for tilkobling av gasspedal og brytere.
- Tre lysdioder for å vise:
 - 5 V virker.
 - 3.3 V virker.
 - Mikrokontroller kjører.

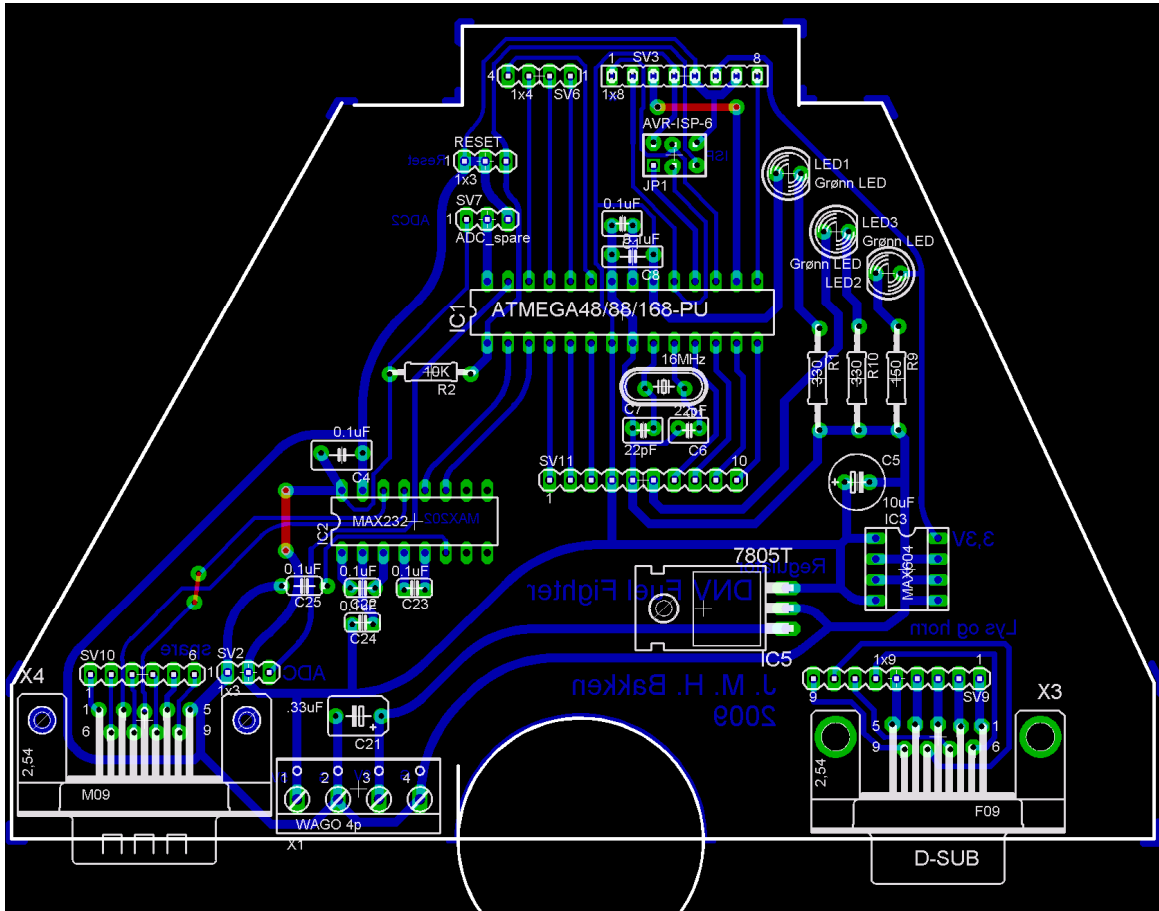
Et hjørne av kretskortet er forbeholdt inngang knappene for lys og horn. Signalene sendes videre via en egen 9-pins D-SUB-kabel til LOGO!-en.



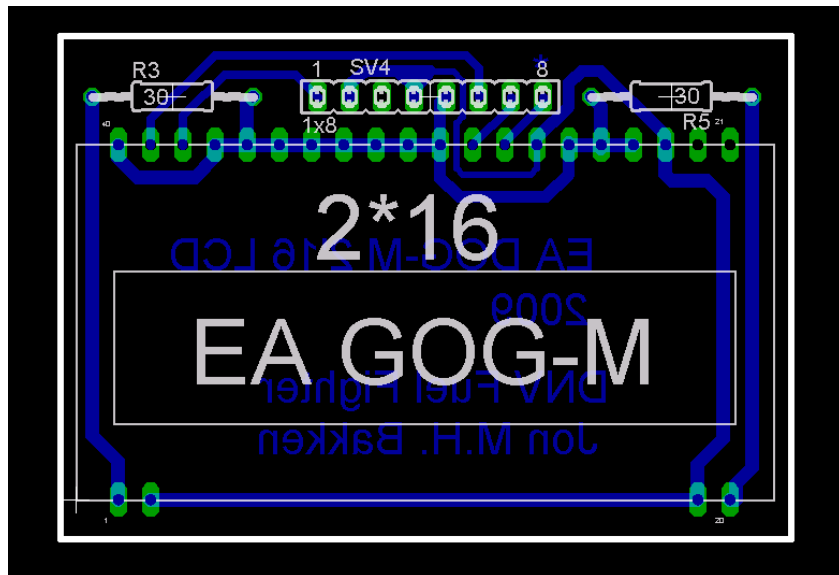
Figur 8 Rattet slik det så ut før oppgraderingen



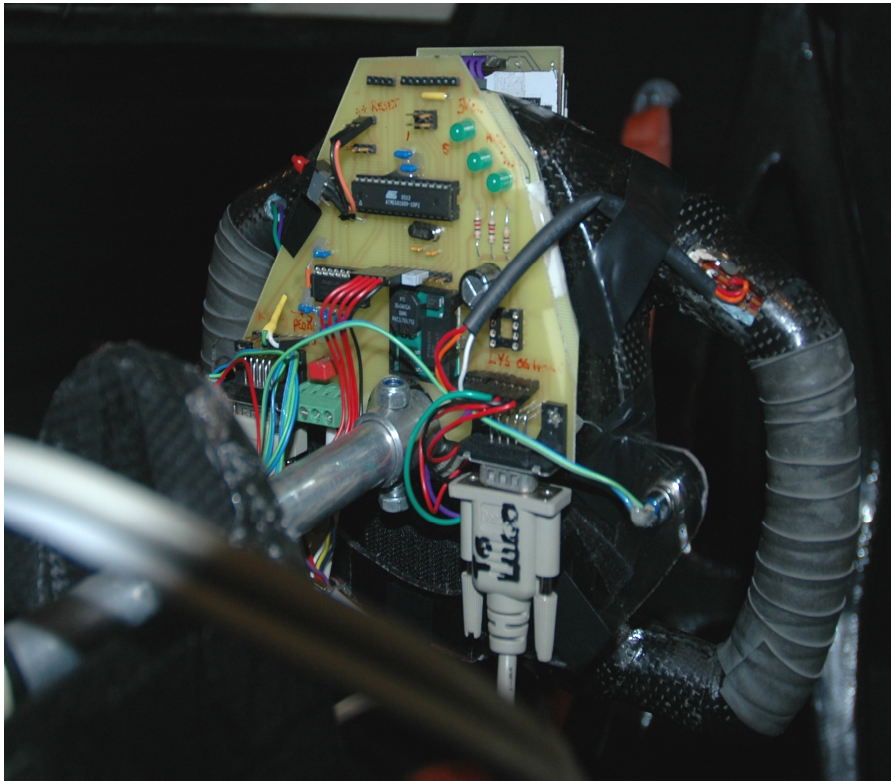
Figur 9 Beregning av fysisk størrelse på kretskortet



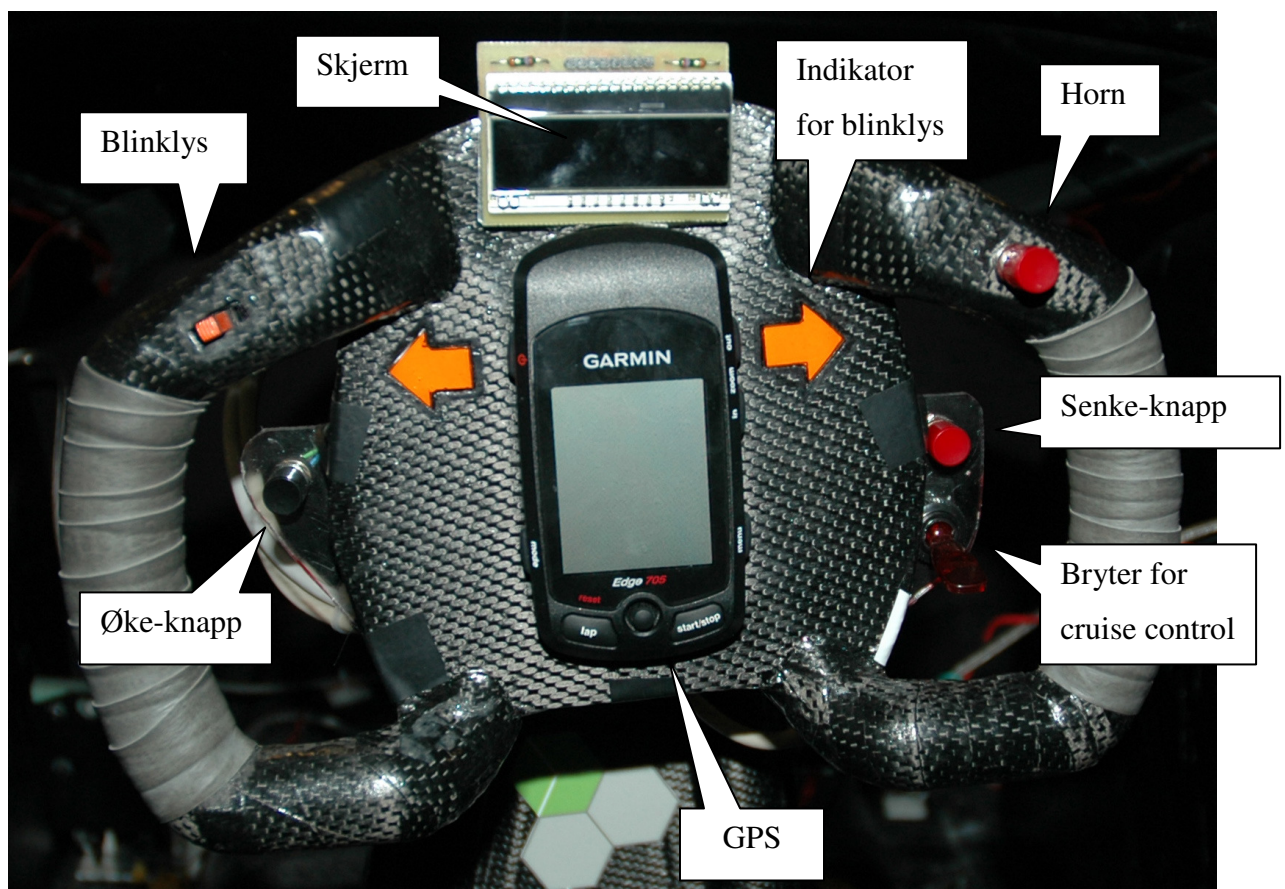
Figur 10 Utlegg av kretsen. Ringen nederst er staget til rattet.



Figur 11 Utlegg av kretsen for skjermen



Figur 12 Kretskortet montert på rattet i bilen

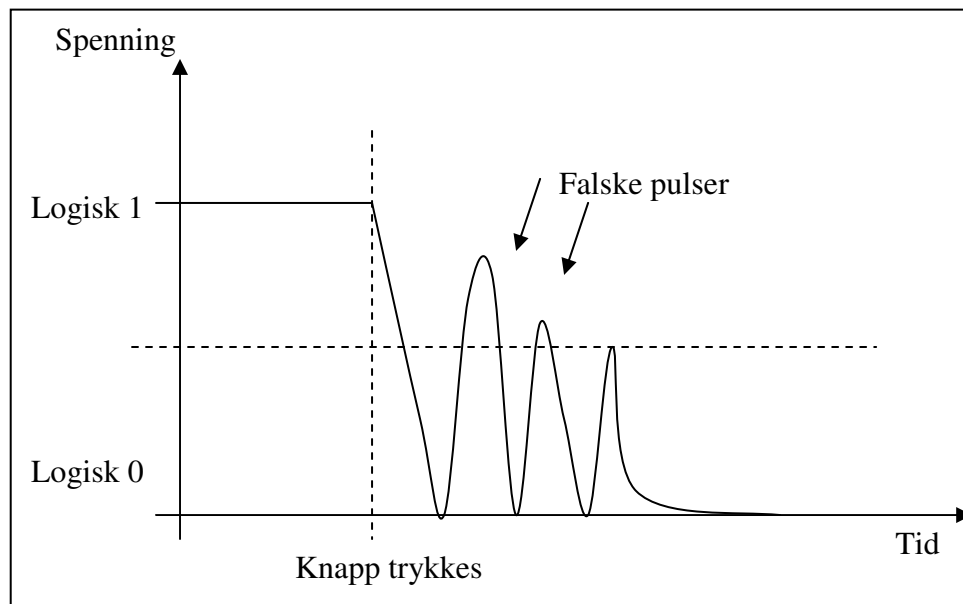


Figur 13 Forsiden av rattet.

5.2 Damping av knappespenningssvingninger

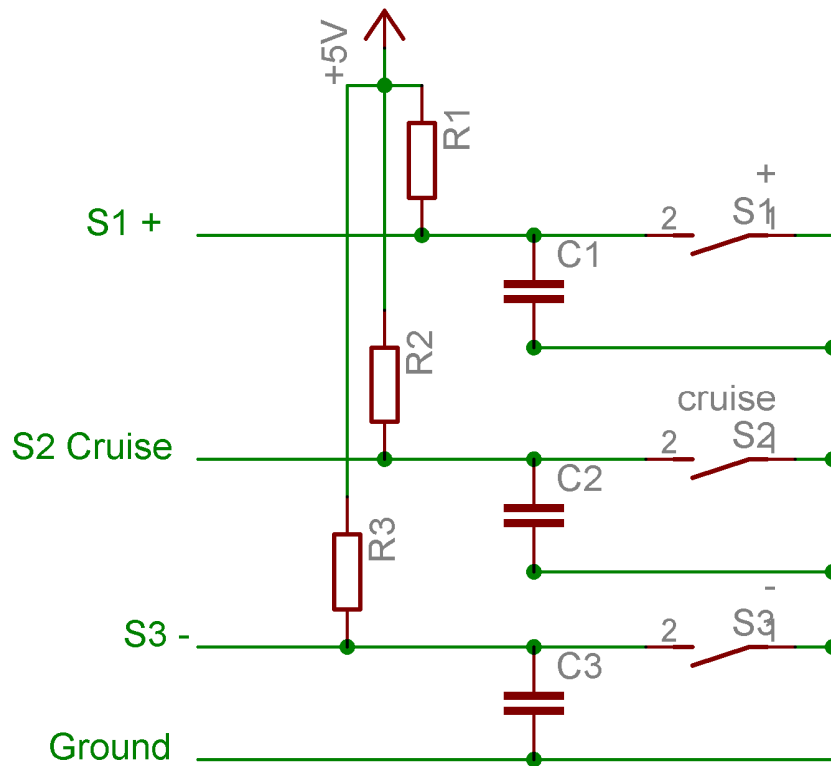
Enhver mekanisk bryter som trykkes inn fungerer som en liten induktans. Dette medfører svingninger på signalet, som vist i Figur 14. Signalet hopper mellom å være over og under grensa for logisk 0 og 1. I en avbruddsbasert programkode eller ved høy samplingfrekvens vil et trykk registreres som flere korte trykk.

Teknikker for å unngå problemer kalles *debouncing*. *Designing Embedded Hardware* [18] foreslår å løse dette i software siden brukeren som oftest holder knappen inne et halvt sekund. Når softwaren først registrerer en logisk 0, venter den i noen hundre millisekund og sjekker inngangen på nytt. Er det fremdeles lavt, er det et gyldig knappetrykk og neste gang den registrerer logisk 0 er det et nytt knappetrykk.

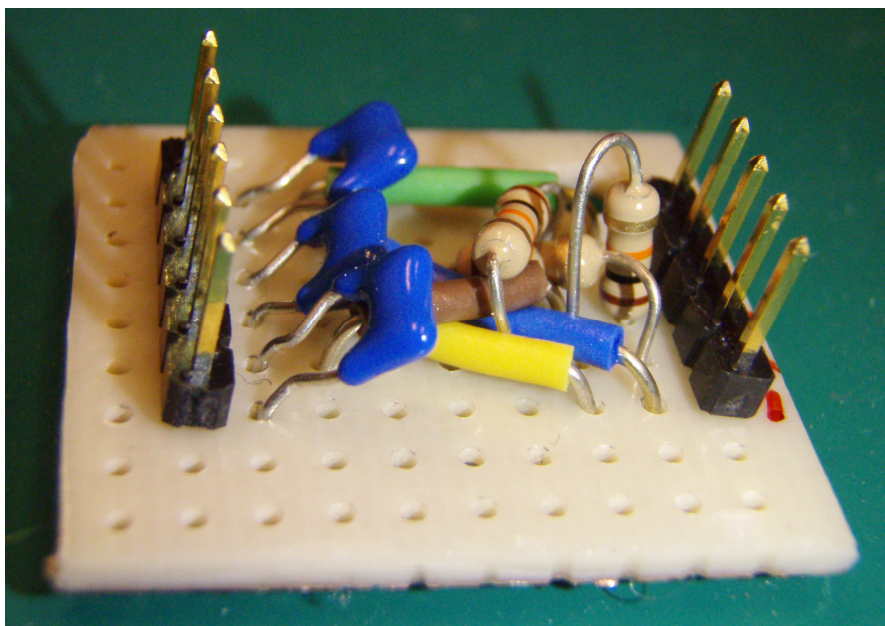


Figur 14 Spenning over knapp uten debounce

For å minimere kompleksiteten i software, ble problemet løst elektronisk i kombinasjon med pull-up-motstander. Dette viste seg å bli en både enkel og effektiv løsning, som vist i Figur 15 og Figur 16. En keramisk kondensator på $0,1 \mu\text{F}$ ble plassert i parallellell med bryteren og en motstand på $10 \text{ k}\Omega$ trekker signalet opp når knappen ikke er trykket. Signalet er dermed invertert ved at knapp trykket gir logisk 0.



Figur 15 Skjemategning for debouche-krets

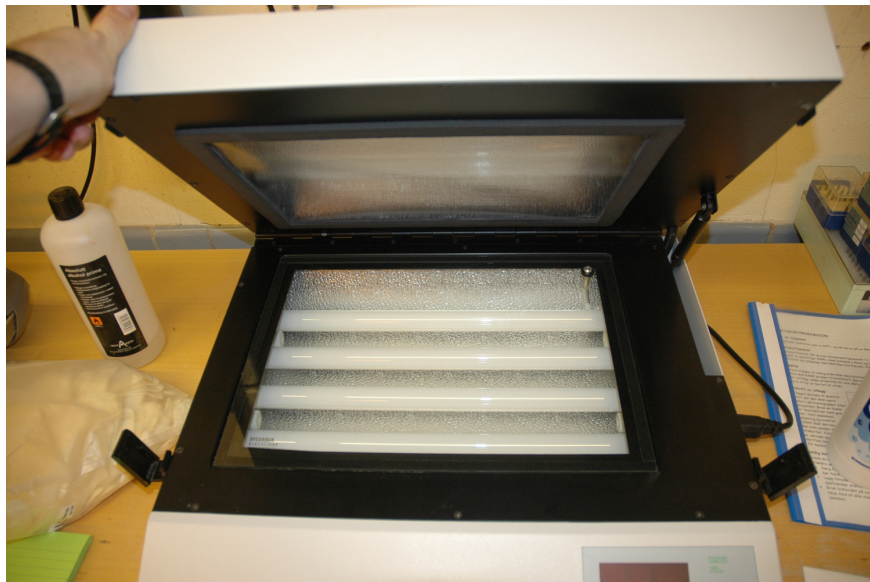


Figur 16 Kompakt debounce-krets for cruise-control-knappene.

5.3 Produksjon av kretskort

Kretskort kan bestilles fra PCB-leverandører eller man kan gjøre jobben selv. Institutt for Teknisk Kybernetikk har utstyr for egenproduksjon. Prosessen er som følger:

1. Kretsen designes i et egnet program og skrives ut på folie i svart-hvitt. På svarte områder beholdes kobberet og på gjennomsiktige områder skal det etses bort.
2. Folien festes på et blankt kretskort med et syreresistent UV-belegg.
3. Kortet belyses i et lysbord med ultrafiolett lys i to minutter. Belegget på kortet brytes ned de stedene kobberet skal bort. Hadde det vært et tosidig kretsdesign, måtte begge sider hatt folie og blitt belyst.



Figur 17 Lysbord for belysning av kretskort

4. Folien fjernes og belyst belegg vaskes bort i et lutbad.



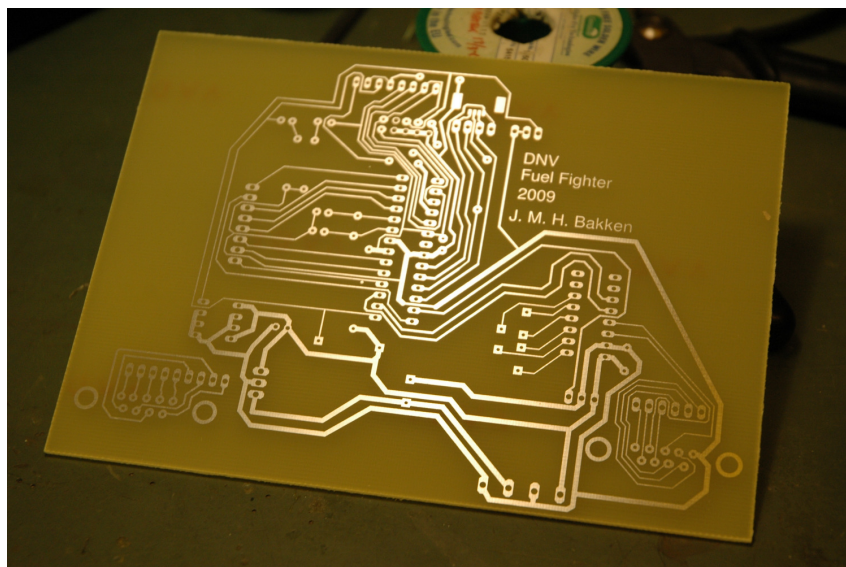
Figur 18 Lutbad for å vaske bort belyst belegg

5. Kortet etses i et oppvarmet syrebad i ca tre minutter. I syrebadet forsvinner kobberet der belegget er fjernet, resten blir værende. Nå vil kun kretsbanene stå igjen.



Figur 19 Oppvarmet syrebad for å etse vekk kobber

6. Kortet vaskes først i vann og så med rødsprit for å fjerne restene av belegget.



Figur 20 Ferdig etset og vasket kretskort

7. For å få en bedre overflate på kobberbanene, bør kortet fortinnes. Dette gjøres ved å helle en sur tinnløsning over kortet og la det ligge badet i væsken.
8. Det borres hull der komponentbein og ledninger skal festes.
9. Komponentene loddes på.

5.4 Lys og horn

Arrangøren av konkurransen stiller strenge krav til bilene og spesielt til bilene i Urban Concept. Bilene skal være utrustet ganske likt som vanlige personbiler både når det gjelder bagasjeplass og sikkerhet. Lys og horn er i tillegg viktig under kjøringen av løpet på bane. Ved forbikjøring skal føreren gi signal med hornet. Ved bremsing og stans er bremselys påkrevet.

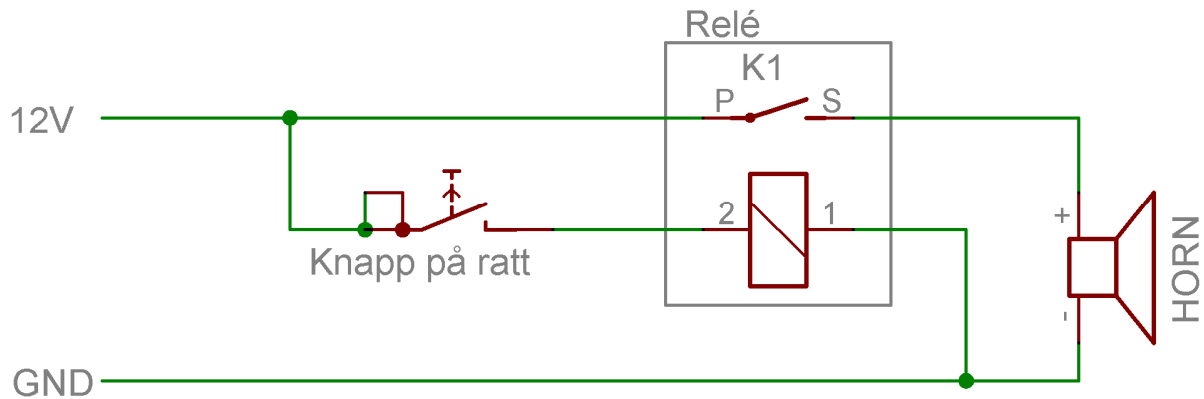
Horn

Reglene [16] sier følgende om horn:

Each vehicle must be equipped with the authorised horn that can be purchased on the Shell Eco-marathon Website's e-shop centre.

Dette er et standard bilhorn fra Bosch og har ifølge databladet [9] og produktbladet [10] en effekt på 50 Watt ved 12 V. Etter en lydtest ble det konkludert med at lyden var svært høy og

hornet ble dermed plassert i høyre hjulbue for å være best mulig dempet for føreren, men samtidig godt hørbar på utsiden av bilen. Et standard bilrelé ble plassert foran hornet for å begrense strømmen som går gjennom bryteren på rattet, som vist i Figur 21.



Figur 21 Oppkobling av horn, relé og bryter på rattet.

Lyssystemet

Om lys sier reglene [16]:

The vehicle must have a lighting system in proper working order for on- road use, including:

1. *Two front headlights*
2. *Two front turn indicators*
3. *Two amber rear turn indicators*
4. *Two red brake indicators lights in the rear*
5. *Two red rear lights (may be combined with the brake lights)*
6. *The centre of each headlight beam must be located at least 30cm to each side of the longitudinal axis of the vehicle*
7. *The mandatory red indicator light for the self starter operation must be separate from any of the above (Article 78)*

Punkt 5 og 7 var endret siden 2008 og punkt 7 gjelder ikke hydrogenbiler. Det eksisterende lyssystemet i bilen var i tråd med de nye reglene. Det bestod av totalt 14 LED⁸-lamper på 350 mA for hoved- og blinklys samt 9 røde SMD⁹-LED på 20 mA. Lampene var parallellkoblede

⁸ LED – Light-emitting diode. En lysdiode.

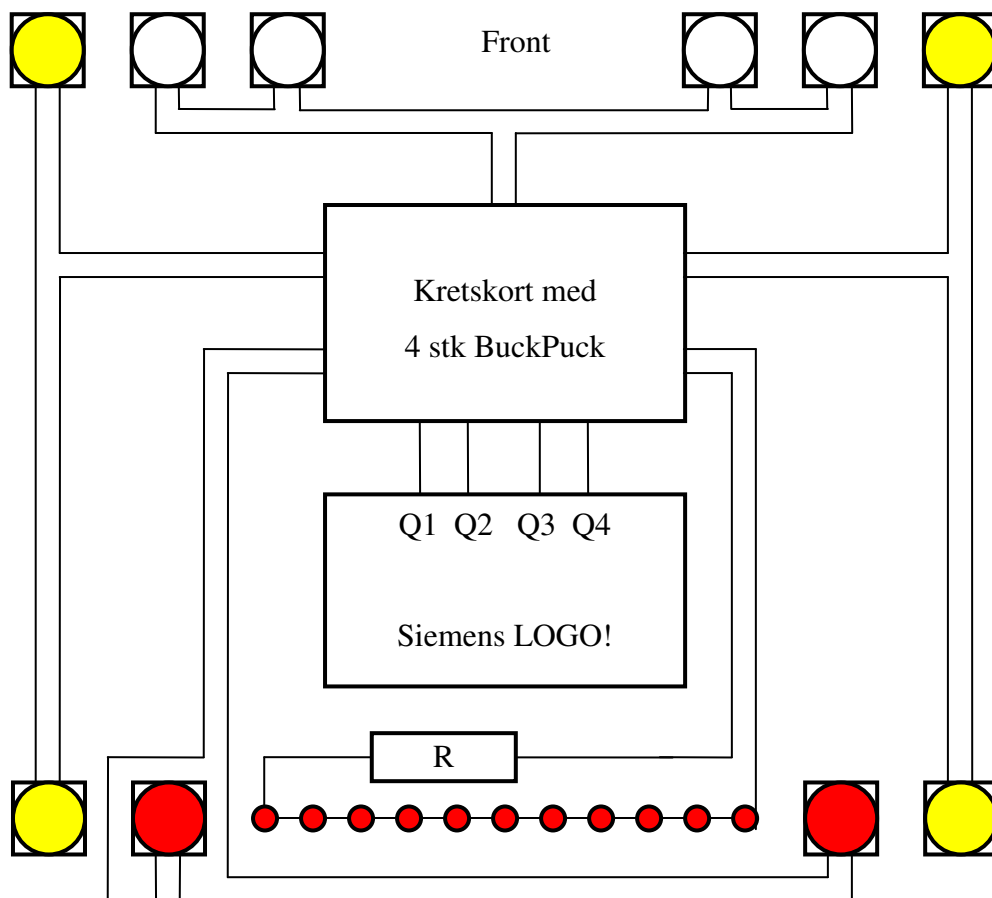
⁹ SMD – Surface-mount device. Overflatemontert komponent.

uten motstand for å spare vekt. Det eneste som begrenset strømmen til lampene var batteriene. Lysene ble kontrollert av brytere på ratt og rattstammen og en Siemens LOGO! ble brukt for den logiske styringen.

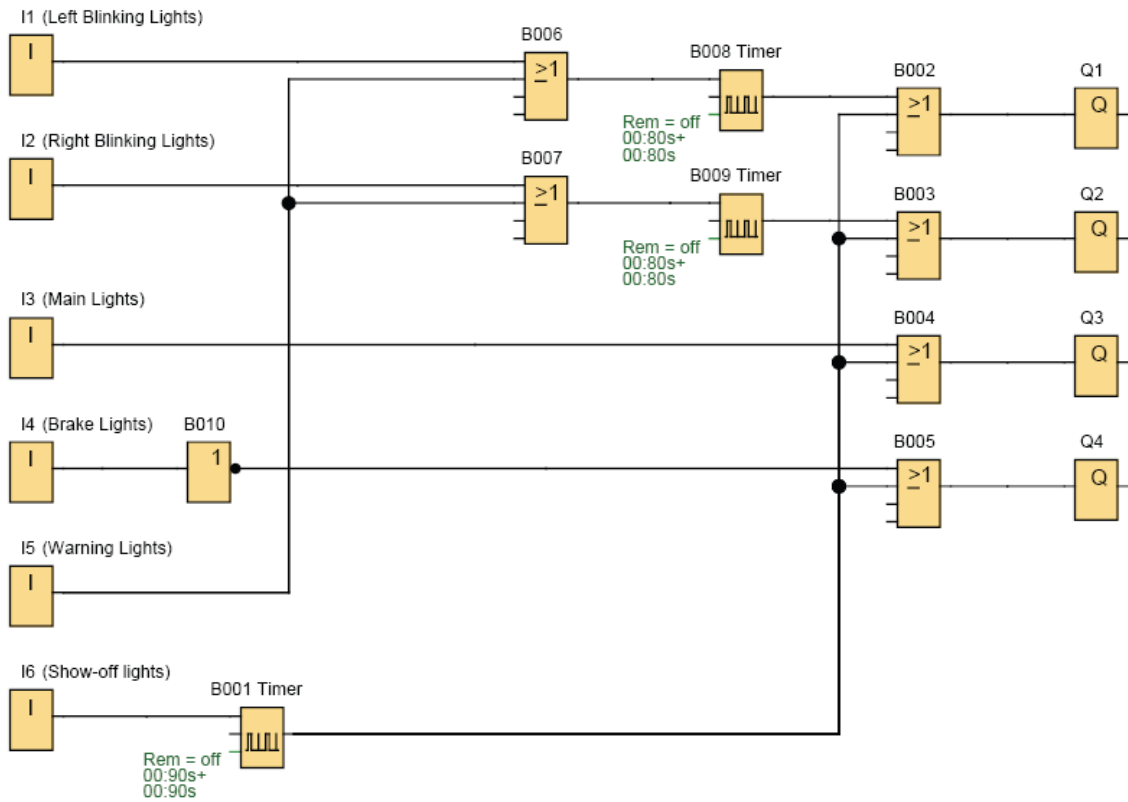
Det ble tidlig klart at fjorårets løsning kunne beholdes med et par forbedringer.

Parallellkoblingen av lampene var ikke mulig sammen med en ny batteriløsning og lysene bak måtte redesignes på grunn av en mulig forlengelse av bilen. De aerodynamiske egenskapene til forlengelsen var usikre, og dermed måtte lysene i den opprinnelige bakenden beholdes.

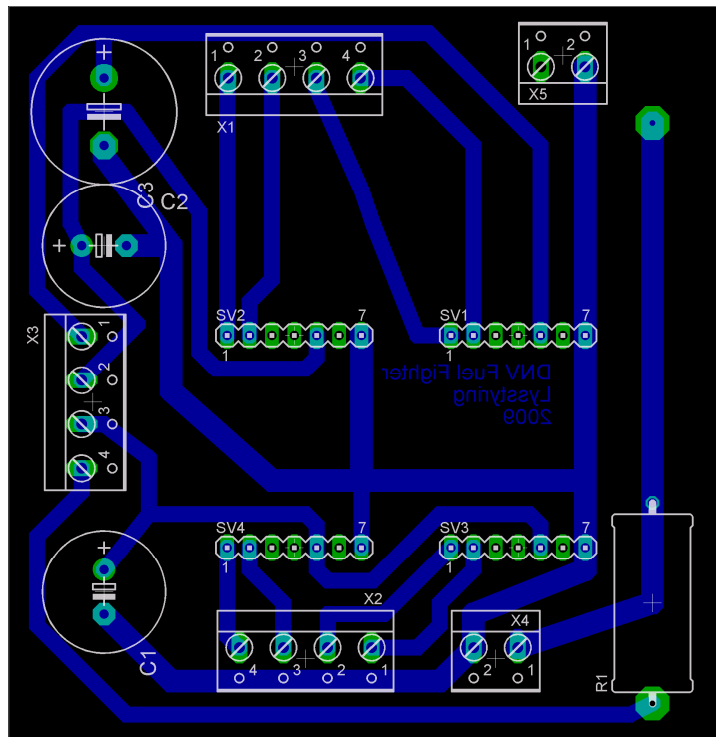
Strømmen kan lettest begrenses ved hjelp av motstander. Dette resulterer i at overflødig energi blir brent bort. Ulempene er at lysstyrken vil variere med batterispenningen og man får motstander som sender ut mye tapt energi i form av varme. Et alternativ var å bruke LED-drivere som gir en konstant strøm uansett spenning. Valget falt på BuckPuck. Den lille modulen på 2x2 cm tåler en inn-spenning på opptil 32 V og leverer ut en fast strøm på 350mA.



Figur 22 Fysisk oppkobling av LEDs, motstand, BuckPuck og LOGO!



Figur 23 Logisk styring implementert på Siemens LOGO! er beholdt fra 2008.



Figur 24 Krets for lysstyring. Motstanden til høyre var ikke i bruk.

5.5 Batteriløsning

Fjorårets bil bruke lithium-ionbatterier. Tolv 3,7 V batterier var koblet sammen til en batteripakke. Batteriene måtte lades én og én og dette var både tungvindt og tidkrevende. I tillegg var sikkerhetsmarginen på kapasiteten lav.

Kalkulering av batterikapasitet

Tabell 5 Strømforbruk i komponentene

Forbruk	Spenning	Strøm	Varighet	Forbruk	Kommentar
Styring brenselcelle	24 V	200 mA	100 %	4,8 Wh	1)
Ventiler	24 V	150 mA	99 %	3,6 Wh	
Fremdriftsystem	24 V	10 mA	100 %	0,2 Wh	2)
└ Skjerm	24 V	2 mA	100 %	0,0 Wh	
LOGO!	24 V	20 mA	100 %	0,5 Wh	
Lys	24 V	150 mA	5 %	0,2 Wh	3)
Horn	12 V	4166 mA	1 %	0,5 Wh	4)
			SUM	9,8 Wh	

- 1) Maksimalforbruket til sbRIO er 6,0 W uten eksterne komponenter, ifølge databladet
- 2) Fremdriftsystemet har en innebygget Switching Regulator som senker spenningen til 5V.
- 3) Lysene brukes kun ved forbikjøring og stans.
- 4) Hornet er det eneste i systemet som trenger 12V og har en effekt på 50W.

Det var bare ett godkjent horn og det gikk på 12 V. Det ble dermed nødvendig å designe batteriløsningen for både 12 og 24 V.

Kravspesifikasjon

Absolutte krav til batteripakken:

- Det er kun lov med en batteripakke ifølge reglene.
- Skal kunne gi ut 12 og 24 V
- 12 V-delen skal kunne gi 5 A i pulser på opptil tre sekunder til hornet.
- 24 V-delen skal kunne gi 0,5 A kontinuerlig.
- Nok kapasitet til å kunne kjøre en time.

Ønskede egenskaper:

- Lav vekt.
- Enkelt å lade.
- Brukervennlig: Enkelt å ta og inn av bilen.

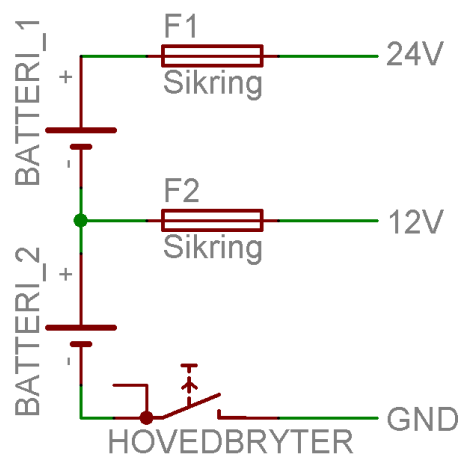
Konklusjon

En god sammenligning mellom aktuelle batterityper finnes i Elvein Grudics masteroppgave [1]. Valget falt på to 12 V drillbatterier fra Biltema, som vist i Figur 25. Batteriene er svært enkle å lade, klarer høye strømpulser fra hornet og er relativt billig i innkjøp. Vekten på hver av batteriene er på 581 gram og tilsvarer omtrent en fordobling av 2008-batteriene. Kapasiteten ligger på 52,8 Wh. Dette er godt innenfor minstekravet på 9,8 Wh og gir en trygghet om å ikke gå tom for strøm.

Batteriholder fra drillene ble demontert og brukt i bilen. Montering og demontering av batteriene er dermed gjort med et håndgrep. Det ble kjøpt inn to ekstra batterier slik at et par kan brukes i bilen mens det andre paret er til lading.



Figur 25 Batteri på 12 V og 2,2 Ah.



Figur 26 Kretsskjema for batteri, hovedbryter og sikringer

5.6 Kabling

2008-laget prioriterte vekt svært høyt i alle faser under byggingen av bilen. En bieffekt av dette ble underdimensjonering i kablingen, spesielt til horn og lys, med tanke på strømstyrke og mekanisk belastning.

En full utskifting av brenselcellestak, motor, motorstyring og styresystem gjorde det naturlig å bytte ut all kabling i bilen. Det ble lagt vekt på å bruke flertrådede ledninger med tilstrekkelig tverrsnitt for å forhindre energitap.

Signalkabler fra ratt til motorstyring og lysstyring er 9-leders D-SUB og ble valgt på grunn av enkel tilgang på kabler og kontakter, samt solide kontakter med skruer som er lette å montere og demontere.

Det meste av kablingen er utført med enkeltledere. Et unntak er kabelene til lys i halen, hvor oppsettet er vist i Tabell 6.

Tabell 6 Farger på ledere til lys i halen

Funksjon	Farge, positiv leder	Farge, negativ leder
Kjørellys	Lilla	Blå
Bremselys	Grønn	Brun
Høyre blinklys	Gul	Hvit
Venstre blinklys	Svart	Rød

6 Resultater og diskusjon

Her blir resultater fra de forskjellige delene av oppgaven presentert og diskutert.

6.1 Fremdriftstyringen

Systemet virket smertefritt. Tilbakemeldingen fra føreren på selve cruise kontrollen taler for seg selv: "Awsome".

6.2 Kommunikasjon med fører

Det var relativt enkelt å kjøpe kontantkort. Prisen ble ca 300 kr for to sim-kort og ringeminutter. Løsningen med mobiltelefon fungerte svært godt. Vind mot mikrofonen på hjelpemannskapets telefon var et lite problem, men ble løst ved å oppsøke vindstille steder ved banen.

6.3 Batteriløsningen

Batteriløsningen fungerte svært godt. Batteriene gikk aldri tom som strøm ved fordi man var påpasselig med å lade etter hvert løp og testløp.

6.4 Konkurransen

Under testløpene før konkurransen virket bilen smertefritt. En alvorlig feil dukket opp etter at bilens lys var kjørt i "show-off mode" i omtrent 10 minutter. I denne modusen vil alle bilens lys blinke i takt. Det medførte trolig at batterispenningen pulserte med frekvens på 0,55 Hz. sbRIO-kortet til brenselcellestyringen inneholder en relativt stort kondensator som var med å kompensere for spenningspulsene. Foran kondensatoren ligger en termistor som skal virker som sikring. Den pulserende strømmen gjennom denne forårsaket høy varme og komponentet røk. Til alt hell var et ekstra kort med til konkurransen og problemet ble løst etter noen timer. Det medførte at flere sterkt ønskede testrunder ble avlyst.

Under løp nummer 1 ble bilen først hindret i å starte på grunn av lekkasje fra hydrogentanken. Da det viste seg at feilen lå i arrangørens utstyr, fikk bilen starte i løpet til en annen gruppe umiddelbart etter. Bilen kom bare omtrent 400 meter på grunn av at hydrogenflaskeventilen hadde for liten åpning.

Løp nummer 2 gikk bedre. Denne dagen var preget av mye vind og motoren ble kjørt hardere enn normalt. Etter fire runder stanset motoren og føreren gikk i gang med feilsøking over telefon med hjelpemannskapet. Før feilen ble funnet ble bilen vinket av banen og løpet var over. Feilen antas å ha vært overoppheting i motorrommet. Motorstyringen har da slått seg av. Etter løpet ble ventilasjonen i motorrommet forbedret ved å lage hull i karosseriet.

Det tredje og siste løpet startet smertefritt. Alle tre stoppene ble gjennomført og bilen kjørte i mål litt under to minutter før tidsfristen. Etter en halv time forelå resultatet som tilsa et forbruk på 1246 km/l.

6.5 Brukergrensesnitt

Nielsens heuristiske brukergrensesnittprinsipper har blitt oppfylt i varierende grad:

1. **Vise systemstatus.** Skjermen viser innstillt cruise-hastighet, noe som ikke er normalt i vanlige personbiler. Dette er svært nyttig i bilen hvor hastigheten bør holdes på bestemte nivåer underveis. Brytere som veksler mellom to eller flere posisjoner er utført med vippebrytere. Føreren kan både se og føle hvilken posisjon bryterne er i.
2. **Kobling mellom systemet og den reelle verden.** Senke-knappen til cruise kontrollen er plassert på venstre side av rattet og øke-knappen på høyre. Dette er i tråd med at verdier som oftest øker mot høyre.
3. **Brukerkontroll og brukerfrihet.** Dødmannsknappen gjør at brukeren nårsomhelst kan slippe gassen og motorpådraget settes til null. Totalsystemet er dessverre tungt å restarte og det er mange steg som må gjennomføres, som vist i Oppstartprosedyren i vedlegget.
4. **Konsistens og standarder.** Cruise kontrollen følger stort sett standarden fra standard biler bortsett fra at gasspedalen fungerer som en dødmannsbryter.
5. **Unngå feil.** Programkoden er skrevet uten avbrudd og all funksjonalitet utføres i en big-while-loop. Det gir ekstremt gode sanntidsegenskaper og hindrer at mikroprosessen overbelastes.
6. **Gjenkjennelse fremfor pugging.** Dette prinsippet er til en viss grad oppfylt. Informasjonen på skjermen kunne vært bedre og knappene kunne vært bedre merket. På en annen side blir brukeren en ekspert og dermed nødt til å vite hvor knappene sitter uten å se på dem. Under kjøring må det fokuseres på det som skjer utenfor bilen.

7. **Fleksibilitet og effektiv bruk.** Systemet åpner for to måter å senke hastighet på under kjøring med cruise control. Føreren kan bruke senke-knappen eller rett og slett slå av cruise-controllen eller motoren og trille til den ønskede hastigheten er nådd. Bortsett fra dette er det ingen andre bevisst designede snarveier, men det er heller ikke nødvendig i denne sammenhengen.
8. **Estetisk og minimalistisk design.** Det har vært et bevisst valg å ha så få knapper som mulig. Start av motor utføres under oppstarten av mikrokontrolleren. Knapp for start er da i praksis den samme som for å restarte kontrolleren. Skjermen viser vinkelposisjon på gasspedalen uten at dette er av interesse for føreren. Allikevel er det en god løsning da informasjonen på skjermen både kan brukes under løpet og når gasspedalposisjonene tunes. Estetisk står det heller dårlig til. Rattet bærer preg av å være en utvidelse av fjorårets løsning utført under tidspress.
9. **Hjelp brukeren til å gjenkjenne, diagnostisere og fortsette etter feil.** Dette ble løst ved å ha hjelpepersonell tilgjengelig over telefon under løpet. Lysdioder på rattet og panelet for brenselcellestacken ga en indikasjon på hvor eventuelle feil lå. Ideelt sett burde feilmeldinger fra motorstyringen kommet opp på skjermen, men dette ble ikke implementert.
10. **Hjelp og dokumentasjon.** Knappene på rattet og rattstammen burde vært merket bedre. Systemet er godt dokumentert både i denne og andre av gruppemedlemmenes rapporter.

7 Konklusjon

Konkurransen ble stor suksess for NTNU. Laget vant førsteplassen i Urban Concept med et resultat på 1246 km/l. Dette er det beste resultatet i denne klassen i verden noensinne.

Resultatet ble mulig gjennom å forbedre alle systemene i en allerede god bil. Denne rapporten har tatt for seg fremdriftssystemet. Det foreligger ingen måledata på denne delen av systemet, men virkningen var stor blant annet ved hjelp av en cruise control som akselererer bilen hurtig opp i hastighet og holder hastigheten på et jevnt nivå. Føreren fikk god informasjon om hastigheten gjennom en skjerm-løsning og en GPS. Under løpet har føreren holdt kontakt med hjelpepersonell ved banen over mobilkommunikasjon. Dette ble vurdert som den beste løsningen før løpet og fungerte utmerket.

Bilen har også fått oppgradert lys- og hornsystemet. Lyssystemet har blitt mer robust og energieffektivt. Fremdriftssystemet og styresystemet av brenselcellestacken drives på batterier. Batteriløsningen ble oppgradert og er lette å bruke.

8 Videre arbeid

Det vil være to hovedstrategier for neste års gruppe; Beholde bilen og forbedre den ytterligere eller bygge en ny bil fra bunnen av. I en ny bil vil det være helt åpent hvilke områder man prioriterer. Velger man derimot å beholde bilen vil følgende punkter være verdt å se på:

- Bilen er dimensjonert for å akkurat fullføre tre løp i konkurransen. Derfor er vekt og effektivitet prioritert fremfor holdbarhet. Av komponenter som berøres av denne rapporten er det rattet som lider mest av tidens tann etter utallige prøvekjøringer med gjestesjåfører. I arbeidet med et nytt ratt bør både skjerm og knapper tilpasses føreren.
- Gasspedalen plukker opp elektronisk støy og har svak mekanisk utforming. Overføring basert på strøm fremfor spenning vil bedre dette, men denne delen trenger et nytt design.
- Styling av horn og lys fungerer godt. LED-ene til bremselyset er dog noe i svakeste laget og bør skiftes ut med noen sterkere. Reglene sier ingenting om lysstyrke, men LED-ene bør skiftes både med tanke på sikkerhet og det estetiske.
- Styresystemet har mulighet for å vise feilmeldinger på skjermen, men benytter seg ikke av dette. Feilmeldinger og måledata fra motorstyringa kan dekodes og gi nyttige testdata. Programkoden har åpnet for dette, men funksjonaliteten er ikke ferdigimplementert.
- Cruise Control-en styres ved hjelp av trykknapper. Det bør utredes om det er mer intuitivt å styre den ved hjelp av for eksempel en roterende skive med samme utforming som en lysdimmer.
- Styresystemet starter ikke motorstyringa med mindre den resettes. En bedre løsning vil være at styresystemet poller motorstyringa og starter den automatisk hvis nødvendig.
- Mye av kablinga i bilen er hastverksarbeid og har stort forbedringspotensial. Spesielt bør det bli lettere å koble fra enkeltenheter ved å legge inn kontakter i stedet for (eller i tillegg til) å bruke skruetilkoblingene på komponentene. Dette gjelder mest instrumenteringa hvor et spenningsfall over kontaktene ikke påvirker fremdriftseffektiviteten.
- Fjernavlesning av systemparametre og systemdata er verdifullt under testing. Et slikt system bør være en selvstendig enhet som ikke påvirker kjøringen av de andre

systemene, men har mulighet for å hente ut data. Dermed vil bilen være upåvirket av kommunikasjonsbrudd.

Generelle tips:

- Sette seg tidlig inn i tidligere arbeid, både ved NTNU og andre universiteter/høyskoler. Spesielt boka om PAC-Car II anbefales.
- Bestill deler tidlig. Det er bedre å kjøpe for mye enn for lite.
- Begynn det fysiske arbeidet på bilen tidlig.
- Legg opp til å bruke mye tid på testing før konkurransen. Det blir garantert mindre tid til slikt enn planlagt, men da har man en buffer for å få alt klart før avreise.
- Det er lurt å gjøre seg minst mulig avhengig av andre gruppemedlemmer siden det motsatte ofte medfører at ting ikke blir ferdig før i siste liten.
- Sett bort arbeid før det er for sent. Man trenger ikke (og rekker ikke) å gjøre alt selv. Det finnes gode verksteder både på Institutt for Teknisk Kybernetikk og andre steder på Gløshaugen som kan være behjelpelige med metall- og plastarbeider.
- Lag detaljerte og gjennomtenkte pakkelister før avreise til konkurransen. Dermed unngår man å glemme utstyr og man vil være i stand til å ta ting på sparket.
- Ha alltid en ”Plan B”. Hva gjør man et komponent ryker? Hvor får man tak i utstyr i Tyskland?
- Planlegg også reisen godt. Ha også en pakkliste for personlig utstyr. Det anbefales å bo på campen fremfor hotell. Dette gir nærhet til banen og det sosiale som skjer på kveldene. Her treffer man likesinnede fra hele verden.
- Legg opp til å ha god tid i Tyskland. Det dukker garantert opp transportskader. I tillegg tar forflytning alltid lenger tid enn planlagt.
- Lausitz er et vindutsatt område. Ta det med i beregningene for kjøling i bilen. Husk også telt som tåler vind.

9 Referanser og kilder

- [1] Diplomoppgave Elvedin Grudic: *Electric Propulsion System for the Shell Eco-marathon PureChoice Vehicle*, Institutt for elkraftteknikk, NTNU, 2008.
- [2] Diplomoppgave Rolv Marius Faleide: *Electric Drive for World's Most Fuel Efficient Vehicle*, Institutt for elkraftteknikk, NTNU, 2008.
- [3] Diplomoppgave Anders Gellein: *Motordrift og elektrisk system til Eco-marathon-kjøretøy*, Institutt for elkraftteknikk, NTNU, 2009.
- [4] Prosjektoppgave Anders Gellein: *Framdriftsystem for Shell Evo-marathon-bil*, Institutt for elkraftteknikk, NTNU, 2008.
- [5] Storaker, Pettersen, Christensen, Skogen, Birkeland, Ebbesen: *DNV Fuel Fighter - The world's most fuel efficient car*, Institutt for produktutvikling og materialer, 2009.
- [6] Storaker, Pettersen, Christensen, Skogen, Birkeland, Ebbesen: *DNV Fuel Fighter - Technical and organizational support systems*, Institutt for produktutvikling og materialer, 2009.
- [7] Storaker, Pettersen, Christensen, Skogen, Birkeland, Ebbesen: *DNV Fuel Fighter - Technical overview*, Institutt for produktutvikling og materialer, 2009.
- [8] Storaker, Pettersen, Christensen, Skogen, Birkeland, Ebbesen: *DNV Fuel Fighter - Technical sub-systems*, Institutt for produktutvikling og materialer, 2009.
- [9] Datablad for Bosch bilhorn: <http://www.eshopsem.com/images/Horn.pdf>
- [10] Produktblad for Bosch bilhorn: <http://www.eshopsem.com/images/Horn2.pdf>
- [11] Datablad for Atmel AVR ATmega168.
- [12] Datablad for skjermen EA DOG-M.
- [13] J.J. Santin, C.H. Onder, J. Bernard, D. Isler, P. Kobler, F. Kolb, N. Weidmann, L. Guzzella, *The World's Most Fuel Efficient Vehicle, PAC-Car II*, ETH Zurich, Institut für Mess- und Regeltechnik.
- [14] Tore Skjellnes, *Pecc communication*, 2000.
- [15] SmartMotor, *Hugin motor software user's guide*, 2003, revised 2009.
- [16] Shell Eco-marathon, Official Rules 2009.
- [17] Shell Eco-marathon, Special Rules and Regulations - Europe 2009
- [18] *Designing Embedded Hardware (2nd Edition)*, John Catsoulis, 2005.
- [19] *On-line CRC Calculator*, <http://www.lammertbies.nl/comm/info/crc-calculation.html>
- [20] *Ten Usability Heuristics*, Jakob Nielsen, http://www.useit.com/papers/heuristic/heuristic_list.html

[21] Datablad for National Instruments sbRIO-961x/963x/964x.

[22] Br@ys terminal, <http://braypp.googlepages.com/terminal>

10 Vedlegg

10.1 Vedlegg A: Pakker for testing av motor

Funksjon	Pakke
Start	\$FF\$02\$FF\$FF\$02\$00\$60\$01
Start motor	\$FF\$05\$FC\$10\$01\$55\$00\$00\$2F\$D0
Set speed referance 0,00pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$00\$00\$61\$1E
Set speed referance 0,25pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$04\$00\$A1\$1C
Set speed referance 0,50pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$08\$00\$A1\$19
Set speed referance 1,00pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$10\$00\$A1\$13
Set speed referance 1,50pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$18\$00\$61\$14
Set speed referance 2,00pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$20\$00\$A1\$07
Set speed referance -1pu	\$FF\$0A\$F7\$10\$02\$3F\$04\$00\$00\$00\$00\$F0\$FF\$21\$1A
Read speed ref	\$FF\$06\$FB\$10\$03\$55\$04\$00\$00\$9D\$16
Read temp PCB	\$FF\$06\$FB\$10\$03\$55\$04\$03\$00\$6D\$16
Read temp heat sink	\$FF\$06\$FB\$10\$03\$55\$04\$04\$00\$5D\$14
Stop motor	\$FF\$04\$FD\$10\$01\$01\$00\$90\$54
Stopp	\$FF\$02\$FF\$FF\$04\$00\$C0\$02
Ping	\$FF\$02\$FF\$FF\$01\$00\$90\$01
Reset	\$FF\$02\$FF\$FF\$05\$00\$50\$03

10.2 Vedlegg B: Check list før start

Det var kort tid fra bilen ble kjørt ut på banen til startskuddet gikk. På denne tiden må alle systemer startes og sjekkes. Det ble utarbeidet en detaljert prosedyre for oppstart av bilen. Rekkefølgen er gitt av systemet, men også for å kunne gjøre flere oppgaver samtidig. For eksempel skrur føreren på GPSen mens hjelpemannskapet monterer døra og skrur på hovedstrømbryter.

Under start

1. I motorrommet:
 - a. Sjekke at hydrogenflaskeventilen er helt åpen.
 - b. Sjekke at trykket er riktig.
2. Setebelte og hjelm på.
3. Døra monteres på.
4. Ringe opp mobilen i bilen.
5. Sikkerhetsbryter i førerhuset av.
6. Skru på GPS.
7. Skru på knappen på taket. (hovedstrømbryter)
8. Test av cruise control-knappene.
9. Skru på brenselcelle.
10. Vente noen sekunder på statuslyset fra brenselcella.
11. Skru på auto-purge.
12. Vri om "skulderknappen" for motorkretsen.
13. Restarte hastighetskontrolleren med knapp på baksiden av rattet.
14. Fører trykker lett på gassen for å teste motoren.

Avslutning

1. Vri om "skulderknappen" for å bryte motorkretsen.
2. Skru av brenselcelle og auto-purge.
3. Skru av knappen på taket.

10.3 Vedlegg C: Utdrag av kildekode

Her følger et utdrag av kildekoden til fremdriftssystemet. Fullstendig kode finnes på vedlagt CD.

Fremdrift.c

```
#include "Fremdrift.h"

//int cruise_rpm = 200;
char buttonInc = FALSE;
char buttonDec = FALSE;
char buttonCruise = FALSE;

int main (void){

    //Setup IO pins and defaults
    uartinit();
    SPI_MasterInit();
    delay_ms(500);
    DogmLiteInit();
    //ADC_init();

    DDRC = 0x2F;
    sbi(DDRD,5);

    //init button input on PD2-4
    cbi(DDRD,2);
    cbi(DDRD,3);
    cbi(DDRD,4);

    //speed variables
    int16_t speed_rpm = 0;
    int16_t speed_cruise = 255;
    uint16_t pedal_pos = 0;

    //Starter motor
    LCD_Clear();
    SkrivString("Starter motor");
    MotorStart();
    delay_ms(1000);
    delay_ms(1000);
    delay_ms(1000);
    MotorStartMotor();
    delay_ms(1000);
    delay_ms(1000);
    LCD_Clear();
    SkrivString("Motor started");
    delay_ms(1000);
    LCD_Clear();

    //Main loop
    while(1){

        //Status Blink og 0,1s delay
        sbi(PORTD,5);
        delay_ms(10);
        cbi(PORTD,5);
```

```

delay_ms(90);

//Poll buttons and pedal
buttonDec = ~PIND & (1<<2);
buttonCruise = PIND & (1<<3);
buttonInc = ~PIND & (1<<4);
pedal_pos = ADC_Read();

//Running cruise or manual speed regulation
speed_cruise = CruiseUpdateCruiseSpeed(speed_cruise, buttonDec, buttonInc);
if(buttonCruise) speed_rpm = CruiseAuto(speed_cruise, pedal_pos);
else speed_rpm = CruiseManual(speed_rpm, pedal_pos);

//Writing speeds to screen and send speed to SMIK
SkrivSpeed(speed_rpm, speed_cruise);
MotorSetSpeedRef(speed_rpm);

//Debugging to screen
char pos[5] = {0};
itoa(pedal_pos,pos,10);
SkrivString(" ");
LCD_SetPos(1,10);
SkrivString(pos);

//Write button state to LCD
LCD_SetPos(1,13);
if(buttonDec) SkrivString("D");
else SkrivString(" ");
if(buttonCruise) SkrivString("C");
else SkrivString(" ");
if(buttonInc) SkrivString("I");
}
return(0);
}

```

motor.c

```

#include "motor.h"
#include <util/crc16.h>

void MotorSend(int len, char data[]){
    uint8_t pos = 0; //Posisjon i utdata-tabellen

    char utdata[100];

    /*Pakkene er på formen:
    /
    /      0xFF | N | Header Checksum | data1 | ... | dataN | Data Checksum |
    /
    /      0xFF markerer start på pakke. Andre steder 0xFF inntreffer, blir
    /      den kodet som to påfølgende 0xFF.
*/

    //Header byte
    utdata[pos]=0xFF;
    pos++;

```

```

//N - number of decoded data bytes
utdata[pos]=len;
pos++;

//Header checksum
char check1 = ~(0xFF + (char)len) + 1;
utdata[pos]= check1;
pos++;
if(check1 == 0xFF){
    utdata[pos] = 0xFF;
    pos++;
}

//Data
int t;
uint16_t crc = 0;
for(t=0;t<len;t++){
    utdata[pos] = data[t];
    crc = _crc16_update(crc,utdata[pos]);
    pos++;
    if(data[t]==0xFF){
        utdata[pos]=0xFF;
        pos++;
    }
}

//Data checksum
char check2 = (char)(crc>>8)&0xFF;
char check3 = (char) crc&0xFF;

utdata[pos] = check2;
pos++;
if(check2 == 0xFF){
    utdata[pos] = 0xFF;
    pos++;
}
utdata[pos] = check3;
pos++;
if(check3 == 0xFF){
    utdata[pos] = 0xFF;
    pos++;
}

//Sender data
for(t=0;t<pos;t++){
    SendByte(utdata[t]);
}
}

void MotorPing(){
    char msg[2] = {0x01,0x00};
    MotorSend(2,msg);
}

void MotorStart(){
    char msg[2] = {0x02,0x00};
    MotorSend(2,msg);
}

void MotorStartWithError(){

```

```

        char msg[5] = {0x10,0x01,0x55,0x04,0x00};
        MotorSend(5,msg);
    }

void MotorStartMotor(){
    char msg[5] = {0x10,0x01,0x55,0x00,0x00};
    MotorSend(5,msg);
}

void MotorStop(){
    char msg[2] = {0x04,0x00};
    MotorSend(2,msg);
}

void MotorReset(){
    char msg[2] = {0x05,0x00};
    MotorSend(2,msg);
}

void MotorSetSpeedRef(int16_t speed_rpm){
    //Filling the message. Leaves the speed reference empty
    char msg[10] = {0x10,0x02,0x3F,0x04,0x00,0x00,0x00,0x00,0x00,0x00};

    //F7$10$02$3F$04$00$00$00$00$10$00

    //Converting RPM value into Q.20 PU value.
    int32_t speed_q20= ((int32_t)speed_rpm)*5958; //5958=2^20/176

    //Filling in the speed reference
    int t=0;
    for(t=0;t<4;t++){
        msg[9-t]= (char)(speed_q20>>(8*(3-t)))&0xFF;
    }

    //Sending the message
    MotorSend(10,msg);
}

int32_t MotorGetAksOL(){
    return 456;
}

int32_t MotorSetAksOL(int noe){
    char msg[10] = {0x10,0x02,0x3F,0x04,0x27,0x00,0x00,0x00,0x00,0x00};
    return 456;
}

void MotorGetSpeedRef(){
    //IKKE FERDIG
    char msg[2] = {0x01,0x00};
    MotorSend(2,msg);
}

void MotorGetSpeed(){
    //IKKE FERDIG
    char msg[6] = {0x10,0x03,0x56,0x02,0x01,0x00};
    MotorSend(6,msg);
}

void SkrivMotorError(uint8_t ec){

```

```

    if(ec==0x00) SkrivString("No error");
    else if(ec==0xFF) SkrivString("General error");
    else if(ec==0xFE) SkrivString("Timeout");
    else if(ec==0xFD) SkrivString("Break");
    else if(ec==0xFC) SkrivString("Not found");
    else if(ec==0xFB) SkrivString("Null pointer");
    else if(ec==0xFA) SkrivString("Wrong magic number");
    else if(ec==0xF9) SkrivString("Wrong data lenght");
    else if(ec==0xF8) SkrivString("Unknown command id");
    else if(ec==0xF7) SkrivString("Program is running");
    else if(ec==0xF6) SkrivString("Program is not running");
    else if(ec==0xF5) SkrivString("Unknown virtual address");
    else if(ec==0xF4) SkrivString("Permission denied");
}

```

cruise.c

```

#include "cruise.h"

//Common
#define HERTZ      10
#define MAX_SPEED  500

//CruiseManual
#define INC_FACTOR 4000
#define DEC_FACTOR 2000
#define RESET_LEVEL 200
#define HOLD_LEVEL 235
#define INC_LEVEL  270
#define MAX_LEVEL  650

//CruisePositionBased
#define CPB_BIAS 112

//CruiseAuto
#define CA_BIAS 200
#define CA_FACTOR 50

int16_t CruiseUpdateCruiseSpeed(int16_t speedCruise, char dec, char inc){
    //Increase and decrease speed
    if(inc) speedCruise += CA_FACTOR / HERTZ;
    if(dec) speedCruise -= CA_FACTOR / HERTZ;

    //Checking limits
    if(speedCruise>MAX_SPEED) speedCruise = MAX_SPEED;
    if(speedCruise<0) speedCruise = 0;

    return speedCruise;
}

int16_t CruiseManual(int16_t speedRpm, uint16_t pedalPos){
    //Manual drive, where the pedal works like in a normal car.

    if(pedalPos < RESET_LEVEL){
        //Stop motor
        speedRpm = 0;
    }else if(pedalPos < HOLD_LEVEL){
        //Decrease speed
    }
}

```

```

        speedRpm -= (int16_t)((HOLD_LEVEL-pedalPos) * DEC_FACTOR / HERTZ / 1000);
        if(speedRpm < 0) speedRpm = 0;
    }else if(pedalPos < INC_LEVEL){
        //Hold speed
        ;
    }else{
        //Increase speed
        speedRpm += (int16_t)((pedalPos-INC_LEVEL) * INC_FACTOR / HERTZ / 1000);
        if(speedRpm > MAX_SPEED) speedRpm = MAX_SPEED;
    }

    return speedRpm;
}

int16_t CruisePositionBased(uint16_t pedalPos){
    //Manual drive, where the position of the pedal is propotional to the speed reference.

    int16_t speedRpm = pedalPos - CPB_BIAS;
    return speedRpm;
}

int16_t CruiseAuto(int16_t speedCruise, uint16_t pedalPos){
    //Cruise Control

    //Checking the status of the pedal

    //OLD. Removed due to problems with the potmeter.
    //if(pedalPos < CA_BIAS) return 0;
    //else return speedCruise;

    return speedCruise;
}

```

adc.c

```

#include "adc.h"

//Read a single-ADC-value according to AREF
uint16_t ADC_Read(){
    uint16_t result = 0;
    PRR ^= (1<<PRADC);//Set the Power Reduction ADC-bit to 0

    //Set the adc-channel in use, and the reference voltage
    ADMUX = (0<<REFS1)|(1<<REFS0)|(0<<MUX3)|(1<<MUX2)|(0<<MUX1)|(0<<MUX0);//ADC4
and Internal AVcc voltage-reference
    DIDR0 = (1<<ADC4D);//Disable the digital buffer at this pin

    ADCSRA = (1<<ADEN)|(1<<ADSC)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);//Start conversion
while(ADCSRA & (1<<ADSC))//Wait until the conversion is finished
    ;
    result = (uint16_t)ADCL; //Get the result, first get the 8 lower bits
    result += (((uint16_t)ADCH)<<8); //Then the 2 higher bits(since this is a 10 bit conversion, only the
two lowest bits will ever be set)

    ADCSRA = 0x00;//Turn off the ADC-module
    PRR |= (1<<PRADC);
}

```

```
        return result;//Return
    }
}
```

dogm_liten.c

```
#include "dogm_liten.h"
#include <string.h>

#define DOGM_PORT PORTB
#define DOGM_DDR DDRB
#define CS_BIT 1
#define RS_BIT 2

void DogmLitenInit(void){
    sbi(DOGM_DDR,CS_BIT);
    sbi(DOGM_DDR,RS_BIT);
    sbi(DOGM_PORT,CS_BIT);
    cbi(DOGM_PORT,RS_BIT);

    SkrivCommand(0b00111001); //Function set
    SkrivCommand(0b00111001); //Function set
    SkrivCommand(0b00011100); //Bias set
    SkrivCommand(0b01011010); //Power
    SkrivCommand(0b01101001); //Follower control
    SkrivCommand(0b011110100); //Contrast
    SkrivCommand(0b00001100); //Turn on
    SkrivCommand(0b00000001); //Clear
    SkrivCommand(0b00000110); //Entry mode set
    SkrivCommand(0b00000010); //Home

    //SkrivString('Ready');
    delay_ms(100);
    SkrivCommand(0b00000001); //Clear
}

void SkrivChar(char inn){
    cbi(DOGM_PORT,CS_BIT);
    sbi(DOGM_PORT,RS_BIT);
    SPI_MasterTransmit(inn);
    sbi(DOGM_PORT,CS_BIT);
}

void SkrivCommand(char inn){
    cbi(DOGM_PORT,CS_BIT);
    cbi(DOGM_PORT,RS_BIT);
    SPI_MasterTransmit(inn);
    sbi(DOGM_PORT,CS_BIT);
    delay_ms(1);
}

void SkrivString(char inn[]){
    uint8_t i;
    for(i=0; i<strlen(inn);i++){
        SkrivChar(inn[i]);
    }
}
```

```

}
void SkrivSpeed(int16_t speed_rpm, int16_t speed_cruise){
    //this might be faster if the whole screen is written in one single SkrivString(), but the letters on
    screen will blink.

    LCD_SetPos(0,7);
    SkrivString(" ");
    LCD_SetPos(1,3);
    SkrivString(" ");

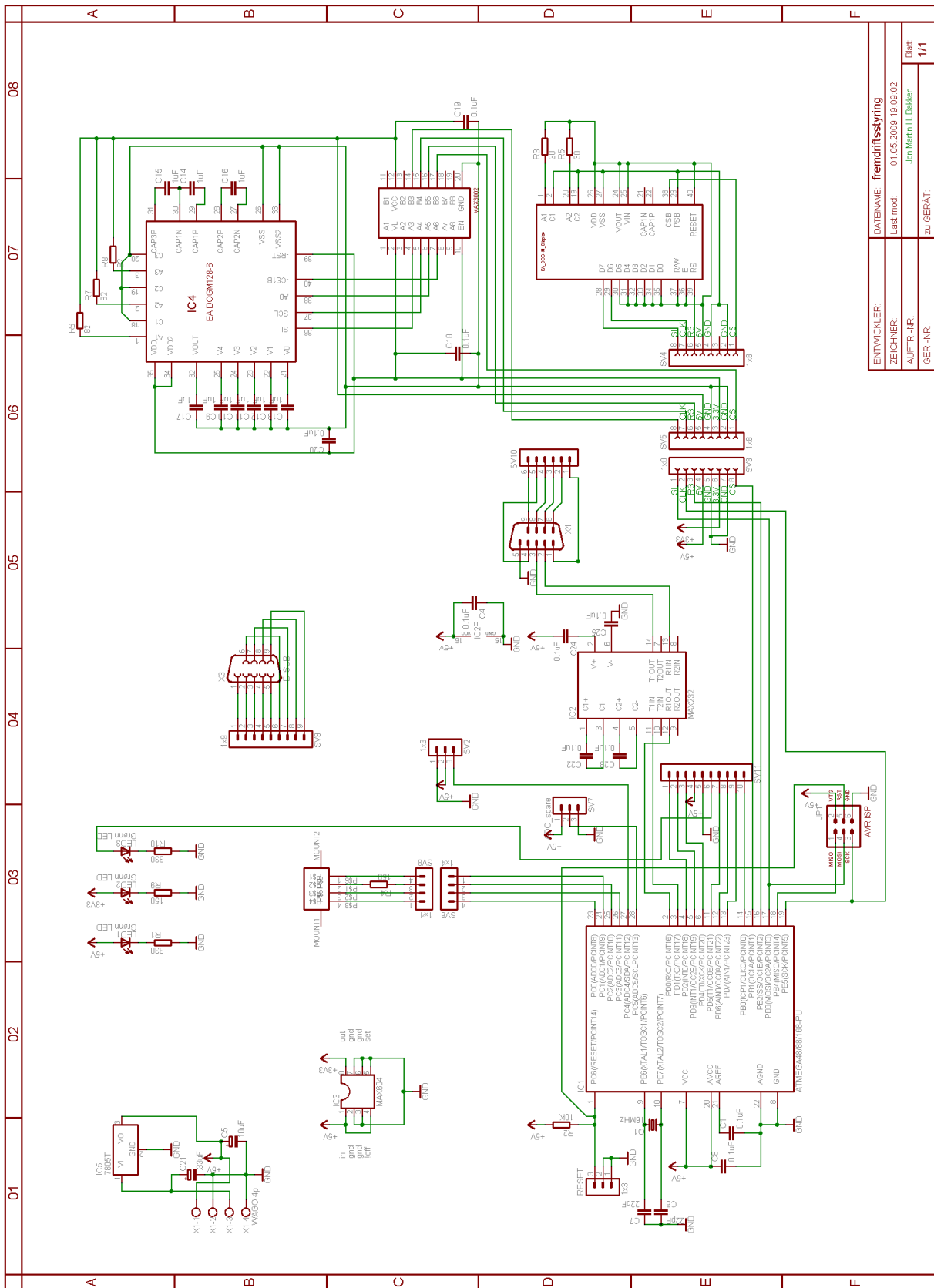
    //Translate speed_rpm to km/h string
    char fart_heltall[5];
    char fart_desimal[5];
    int speed_kmh_heltall = speed_rpm/10;
    int speed_kmh_desimal = speed_rpm%10;
    itoa(speed_kmh_heltall,fart_heltall,10);
    itoa(speed_kmh_desimal,fart_desimal,10);
    uint8_t fart_pos = 7;
    if(speed_rpm<100) fart_pos++;

    //Translate speed_cruise to km/h string
    char cruise_heltall[5];
    char cruise_desimal[5];
    int cruise_kmh_heltall = speed_cruise/10;
    int cruise_kmh_desimal = speed_cruise%10;
    itoa(cruise_kmh_heltall,cruise_heltall,10);
    itoa(cruise_kmh_desimal,cruise_desimal,10);
    uint8_t cruise_pos = 3;
    if(speed_cruise<100) cruise_pos++;

    //Write to screen
    LCD_SetPos(0,0);
    SkrivString("Speed:");
    LCD_SetPos(0,fart_pos);
    SkrivString(fart_heltall);
    SkrivString(".");
    SkrivString(fart_desimal);
    SkrivString(" km/h");
    LCD_SetPos(1,0);
    //SkrivMotorError(0);
    //SkrivString("No alerts");
    SkrivString("C: ");
    LCD_SetPos(1,cruise_pos);
    SkrivString(cruise_heltall);
    SkrivString(".");
    SkrivString(cruise_desimal);
}
void LCD_Clear(){
    SkrivCommand(0b00000001);
    delay_ms(5);
}
void LCD_Home()    {
    SkrivCommand(0b00000010);
}
void LCD_SetPos(int line, int pos){
    //Line: 0 eller 1
    //Position: 0:15
    SkrivCommand(0b10000000|(line<<6)|pos);
}

```


10.4 Vedlegg D: Skjemategning for fremdriftstyring



Tegningen finnes også vedlagt på CD.

10.5 Vedlegg E: Innholdsfortegnelse for vedlagt CD

På vedlagt CD finnes:

- Rapporter:
 - Prosjektrapport Anders Gellein
 - Diplomrapport Elvedin Grudic
 - Diplomrapport Bjørn Ola Wiik
 - Diplomrapport Håkon Rognstad
- Motorstyring:
 - Pecc-COM
 - Hugin motor software user's guide
 - Hardware summary
- Grupperapporter (med forside):
 - DNV Fuel Fighter - Technical and organizational support systems
 - DNV Fuel Fighter - Technical overview
 - DNV Fuel Fighter - Technical sub-systems
 - Brosjyre
- Datablader
 - Mikrokontroller Atmega168
 - Skjerm DOG-M
 - Bosch Bilhorn
 - National Instruments sbRIO
- Shell Eco-marathon
 - Official Rules
 - Special Rules
- Eagle-filer
- Kildekode