

Design og implementering av GPS-mottaker for bruk i vanskelige omgivelser

Prosjektoppgave
Trondheim desember 2005

Tor Ivar Hystad
Tore Johan Høilund

NTNU

Norges teknisk-naturvitenskaplige universitet
Fakultet for Informasjonsteknologi, matematikk og elektroteknikk
Institutt for Elektronikk og telekommunikasjon



1 Forord

Denne rapporten beskriver prosjektoppgaven som er en obligatorisk del av masterstudiet. Prosjektet er en del av fordypningsemnet som gjennomføres i løpet av høstsemesteret i 5. årstrinn (9.semester). Vi har vært to studenter som har samarbeidet på dette prosjektet og denne rapporten er en felles besvarelse.

Vi studerer begge elektronikk med fordypning innen navigasjon og fjernmåling.

Prosjektet har blitt gjennomført ved NTNU i samarbeid med Forsvarets forskningsinstitutt (FFI). Rapporten tar for seg bakgrunnsinformasjon, utvikling og resultater relatert til prosjektet.

Vi vil i denne anledning takke vår veileder ved NTNU, Börje Forssell for all nødvendig hjelp. Ønsker også å takke alle ved FFI som har vært involvert i utviklingen av systemet. Spesielt vår veileder og kontaktperson, Leif Hanssen, som vi har hatt en nærmest kontinuerlig dialog med gjennom hele prosjektet. Leif Hanssens kunnskaper og erfaring innen maskinvare og programvare utvikling samt tilgjengelighet gjennom hele prosjektperioden har vært sterkt medvirkende til prosjektets resultat. Vi vil også takke institutt for Elektronikk og telekommunikasjon, og institutt for Teknisk kybernetikk for utlån av alt teknisk utstyr benyttet under utvikling og testing.

Tor Ivar Hystad

Tore Johan Høilund

Trondheim, desember 2005

2 Innholdsfortegnelse

2 Innholdsfortegnelse	3
3 Sammendrag	5
4 Innledning	7
4.1 Bakgrunn for prosjektet	7
5 GPS bakgrunnsteori	8
5.1 Satellittkonstellasjon	8
5.3 Signaler	9
5.4 Prinsipiell virkemåte	9
5.5 Feilkilder i GPS.....	11
5.6 Kort om modernisering av GPS	12
6 Bussteori	13
6.1 SPI.....	13
6.2 USART	15
6.3 USB.....	16
7 NMEA-0183 protokollen	17
7.1 Andre NMEA standarder	21
8 Valg av komponenter	22
8.1 GPS-modul.....	22
8.2 Mikrokontroller (MCU)	28
8.3 Minne for lagring av posisjoner.....	30
8.4 Andre komponenter	32
9 Design og implementering	34
9.1 Systembeskrivelse.....	34
9.2 Maskinvare.....	35
9.3 Programvare	39
9.4 Brukergrensesnitt	46
9.5 Bilder av GPS-mottakeren	48
10 Test av den ferdige GPS-mottakeren	50
10.1 Elektrisk test før tilkobling av drivspenning.....	50

2 Innholdsfortegnelse

10.2 Elektrisk test med tilkoblet drivspenning.....	50
10.3 Programmering av mikrokontrolleren.....	51
10.4 USB-kommunikasjon.....	51
10.5 Lading av batteri	52
10.6 Enkel test av programmet	52
10.7 Strømforbruk.....	54
10.8 Utfyllende test av GPS-mottakeren.....	55
11 Videre arbeid.....	57
12 Resultat og konklusjon.....	58
13 Kilder og referanser	59
 Vedlegg A, kretsskjema	 61
Produksjonsunderlag.....	63
Komponentplassering	64
Komponentliste	65
Vedlegg B, kretsskjema for utviklingskretser.....	69

3 Sammendrag

Denne prosjektrapporten tar for seg utviklingen av maskinvare og programvare for en GPS-mottaker. Det er i tillegg tatt med noe bakgrunnsinformasjon i rapporten etter ønske fra Leif Hanssen. Den utviklede GPS-mottakeren er ment som en forberedelse til masteroppgaven, der GPS-mottakeren skal integreres med treghetsnavigasjon.

Dette prosjektet kan deles inn i fire hovedområder:

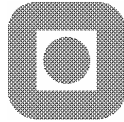
- Valg av komponenter.
- Design av maskinvare.
- Design av programvare.
- Test av ferdig utviklet GPS-mottaker.

Viktige poeng i denne oppgaven er at GPS-mottakeren har mulighet for å lagre posisjonsdata over tid og opprettholde et lavt strømforbruk på grunn av batteridrift. GPS-mottakeren skal også ha en liten fysisk størrelse. Rett valg av komponenter er avgjørende for å oppnå disse designspesifikasjonene. Det benyttes kommersielt tilgjengelige komponenter. Det er utviklet et enkelt brukergrensesnitt i form av et display og trykknapper. De lagrede posisjonsdata kan i ettertid overføres til PC via USB for å etterprosessere.

Den resulterende GPS-mottakeren fungerer som ønsket. Etter enkle tester ble det avduket enkelte feil i maskinvaren, uten at dette gikk på bekostning av systemets funksjon.

Oppgaveteksten er gjengitt på neste side.

NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET
FAKULTET FOR INFORMATIKK, MATEMATIKK OG ELEKTROTEKNIKK



PROSJEKTOPPGAVE

Kandidatenes navn: Tor Ivar Hystad og Tore Johan Høilund

Fag: Navigasjon og fjernmåling

Oppgavens tittel (norsk): **Design og implementering av GPS-mottaker for bruk i vanskelige omgivelser.**

Oppgavens tittel (engelsk): **Development of a GPS-receiver for use in difficult surroundings.**

Oppgavens tekst: GPS-mottakere sammenbygget med annen elektronisk utrustning blir alt vanligere. I den forbindelse er det ønskelig at posisjonsbestemmelsen skal kunne utføres også når det er begrenset sikt til satellittene, f.eks. innendørs eller i skyggen av hus m.m. Forskjellige teknikker muliggjør utnyttelse av GPS også ved kraftig demping av satellittsignalene, men for å øke sannsynligheten for en akseptabel posisjonsbestemmelse i slike situasjoner kan GPS-løsningen støttes av f.eks. magnetkompass og/eller treghetsbasert utstyr. Prosjektoppgaven går ut på å utvikle en GPS-mottaker med tilleggsfunksjoner ved bruk av kommersielt tilgjengelige komponenter, inkludert en GPS-modul. Mottakeren skal lagre posisjonsdata som i ettertid skal behandles på en PC. I den sammenhengen skal det utarbeides et hensiktsmessig lagringsformat for posisjonsdataene. Data overføres fra mottaker til PC ved hjelp av USB. Mottakeren utstyres med display og knapper for å oppnå et enkelt brukergrensesnitt. Ettersom mottakeren drives av batteri, er lavt strømforbruk en viktig faktor. Det skal utføres tester og evaluering av resultatene.

Oppgaven utføres i samarbeid med FFI.

Oppgaven gitt: 23. august 2005

Besvarelsen leveres innen: 16. desember 2005

Utført ved: Institutt for elektronikk og telekommunikasjon

Veileder: Forsker Leif Hanssen, FFI.

Trondheim, 23. august 2005

faglærer

4 Innledning

Dette prosjektet består av å designe og implementere en GPS-mottaker av liten fysisk størrelse som skal lagre posisjoner i et lagringsmedium. GPS-mottakeren skal også ha et enkelt brukergrensesnitt i form av et display og trykknapper. Ved siden av å utvikle maskinvaren, så må også programvaren utvikles. Det skal benyttes rimelige, kommersielt tilgjengelige komponenter.

Et viktig poeng i oppgaven er at GPS-mottakeren skal ha batteridrift, slik at lavt strømforbruk har vært et viktig poeng i valg av komponenter. Kommunikasjon med PCen foregår ved hjelp av USB. Testing av den ferdige enheten inngår også i oppgaven.

Oppgaven videreføres i masteroppgave, da skal GPS-mottakeren videreutvikles med treghtsnavigasjon.

4.1 Bakgrunn for prosjektet

FFI jobber blant annet med et prosjektet ”ad hoc sensor nettverk”. Dette er små noder med innebygd prosessor, kommunikasjonsenhet og sensorer. Disse skal kunne legges ut i stort antall i et geografisk område og konfigurere seg selv i et nettverk. En av problemstillingene i forbindelse med dette prosjektet er posisjonering og lokalisering av nodene. En løsning på dette er posisjonering ved hjelp av GPS.

En sensornode har begrenset strømkapasitet, og en GPS-mottaker bruker generelt mye strøm (se avsnitt 8.1 og 10.7), derfor er det interessant å finne ut om det kan benyttes treghtsnavigasjon til å følge bevegelsen til sensornodene, både for å spare effekt og når noden er utenfor GPS-dekning.

5 GPS bakgrunnsteori

Den fulle betegnelsen på GPS er NAVSTAR-GPS, **NAV**igation **S**ystem with **T**iming **A**nd **R**anging, **G**lobal **P**ositioning **S**ystem.

GPS er et satellittbasert navigasjonssystem som er finansiert og kontrollert av U. S. Department of Defense (DoD). I utgangspunktet var det laget for militært bruk, men i dag er det blitt det mest utbredte navigasjonssystemet også for sivil bruk.

GPS betraktes som et 3-delt system bestående av satellittene (space segment), kontrollstasjonene (control segment) og navigasjonsmottakerene (user segment). Det finnes to hoved GPS-tjenester, en for sivile brukere (SPS, **S**tandard **P**ositioning **S**ervice) og en for autoriserte brukere (PPS, **P**recise **P**ositioning **S**ervice).

Kontrollsegmentet har sitt hovedkvarter på Schriever Air Force Base i Colorado, hvor de får inn data fra en rekke kontrollstasjoner rundt om i verden. Her blir blant annet satellittenes baner og nøyaktigheten til klokken ombord i satellittene nøye overvåket. Kontrollsegmentet inneholder hovedklokka som gir sann GPS tid.

5.1 Satellittkonstellasjon

Den normale konstellasjonen av GPS satellitter består av 24 satellitter, men det er vanligvis også noen reservesatellitter tilgjengelig (29 satellitter totalt per 12.12.05, [34]). De 24 aktive satellittene er tilnærmet uniformt fordelt over 6 baneplan med 4 satellitter i hver. Satellittbanene er tilnærmet sirkulære og har en inklinasjon på 55° relativt til ekvatorplanet.

Inklinasjon på 55° ble valgt av tre årsaker:

1. Satellittene ble opprinnelig planlagt skutt opp ved hjelp av romferge.
2. To satellitter i bane i motsatt retning skal ha skjæring på 90°, noe som gir god geometri for bruker.
3. Denne inklinasjonen gir noenlunde jevn global dekning.

Banene er ikke-geostasjonære med en radius på 26 560 km og omløpstid på et halvt stjernedøgn som tilsvarer ca 11 timer og 58 min. Teoretisk sett skal minst 4 satellitter være synlig fra hvilket som helst sted på jorda uansett tid på døgnet. For å oppnå en entydig posisjonsbestemmelse kreves minimum 4 satellitter (se avsnitt 5.4).

5.3 Signaler

Hver satellitt sender ut to spredt spektrum-signaler på L-båndet. Bærebølgene L_1 og L_2 , har bærefrekvensene $f_1 = 1575,42$ MHz og $f_2 = 1227,6$ MHz. Disse to frekvensene er et multiplum av grunnfrekvensen $f_0 = 10,23$ MHz slik at $f_1 = 154f_0$ og $f_2 = 120f_0$. De to bærebølgene fra hver satellitt bruker BPSK (Binary Phase-Shift Keying) modulert i fase kvadratur. L_1 og L_2 er modulert med to halvtilfeldige koder, også kalt PRN-koder (PseudoRandom Noise). De to kodene er kalt C/A-kode og P-kode. L_1 inneholder begge kodene i tillegg til data signalet, mens L_2 inneholder kun P-koden og datasignal (Det er også mulig å sende C/A-kode på L_2). Signalene er høyrehåndss sirkulær polariserte. C/A-koden har en lengde på 1023 bit, og gjentar seg med en frekvens på 1 KHz. P-koden har en full periode på 267 dager, hver enkelt satellitt sender kun en unik del av denne fulle perioden som er på 7 dager før koden gjentas.

C/A-koden (Clear eller Coarse/Acquisition) er den som benyttes i sivile mottakere, mens P-koden (Precision eller Protected) er kun for autoriserte brukere.

5.4 Prinsipiell virkemåte

PRN-kodene er kjent i GPS-mottakerene, slik at det er mulig for flere satellitter med samme bærefrekvens å gi individuell informasjon til en og samme mottaker.

Idet en mottaker mottar signal fra en satellitt genererer den en egen kode som sammenlignes med den mottatte, slik at den kan låse seg inn på satellittens signal. GPS-signalet inneholder navigasjonsinformasjon for den aktuelle satellitten og en almanakk for alle satellitter, med koeffisienter for å beregne ionosfærisk forsinkelse, og forholdet mellom satellitt-tid og sann GPS tid. GPS-dataene blir sendt på en datarate på 50 bps. Det er helt nødvendig med en nøyaktig klokke om bord i satellittene, siden posisjonsberegning er basert på en måling av hvor lang tid signalet har brukt fra satellitt til mottaker. Derfor har alle satellittene innebygde cesium og/eller rubidium atomur.

GPS-mottakeren må ha signaler fra minst 4 satellitter for å kunne beregne en entydig 3-dimensjonal posisjon. Siden GPS-mottakeren ikke kan være nøyaktig synkronisert med satellitt systemtid betraktes tiden som en ukjent. Mottakeren beregner posisjon ut fra 4 ligninger med 4 ukjente, hvorav 1 av de ukjente er tid og de 3 andre er posisjoner [23]:

$$\psi_1 = \sqrt{(X_1 - x)^2 + (Y_1 - y)^2 + (Z_1 - z)^2} + c\Delta t \quad (\text{Ligning 5.1})$$

$$\psi_2 = \sqrt{(X_2 - x)^2 + (Y_2 - y)^2 + (Z_2 - z)^2} + c\Delta t \quad (\text{Ligning 5.2})$$

$$\psi_3 = \sqrt{(X_3 - x)^2 + (Y_3 - y)^2 + (Z_3 - z)^2} + c\Delta t \quad (\text{Ligning 5.3})$$

$$\psi_4 = \sqrt{(X_4 - x)^2 + (Y_4 - y)^2 + (Z_4 - z)^2} + c\Delta t \quad (\text{Ligning 5.4})$$

ψ_i = støyfri avstand til satellitt.

$[X_i, Y_i, Z_i]^T$ = kartesisk posisjons koordinat til satellitt i .

$[x, y, z]^T$ = kartesisk posisjons koordinat til GPS-mottaker.

Δt = mottaker tidsforskjell fra satellitt systemtid.

c = lys hastighet.

Mottakerposisjonen $[x, y, z]^T$ er bundet til referanse koordinatsystemet brukt av satellittene. I GPS' tilfelle er dette WGS-84 (World Geodetic System 1984), som er sentrert i jordens sentrum. WGS-84 definerer også ellipsoiden som grovt beskriver jordens overflate.

5.5 Feilkilder i GPS

Feilkilder	Forventet feil i avstandsmåling [m RMS]	
	P-kode	C/A-kode
Satellitt klokkefeil	1-3	1-3
Efemeridefeil	2,5-7	2,5-7
Ionosfære forsinkelse	0,4-2	2-15
Tropesfære forsinkelse	0,4-2	0,4-2
Mottakerfeil	0.1-0.3	1-2
Forskjell i mottakerkanaler	0-0,1	0-0,2
Feil på grunn av flerveistransmisjon	1-2	2-4
Resulterende avstandsfeil	3-8	4-18
Resulterende posisjonsfeil		
Horisontalt (HDOP = 1,5)	4.5-12	6-27
Vertikalt (VDOP = 2,5)	7.5-20	10-45

Tabell 5.1; Feilkilder i GPS [21].

I tabell 5.1 er feilkildene i dagens GPS listet opp. For eksempel ser man at ionosfæreforsinkelsen går betraktelig ned hvis man har mulighet til å benytte to frekvenser. Dette er på grunn av at ionosfæren er dispersiv, og man kan da utnytte dette ved å benytte to-frekvens metoden for å redusere denne feilen. HDOP og VDOP står for henholdsvis “Horizontal Dilution of Precision” og “Vertical Dilution of Precision”. Forklaring og utledning av DOP tas ikke med i denne rapporten, for utfyllende forklaring av disse begrepene vises det til for eksempel [21], kapittel 9.

5.6 Kort om modernisering av GPS

GPS skal oppdateres for blant annet å oppnå høyere posisjonsnøyaktighet. Her er noen tiltak som skal til for høyne posisjonsnøyaktighet [24]:

- Nytt sivilt signal på L2.
- En ny bærebølge (L5, 1176MHz, QPSK modulert), med et nytt sivilt signal.
- Forbedring av satellittene.
- Flere bakkestasjoner.

6 Bussteori

Sammenkobling mellom forskjellig komponenter er gjort på forskjellige måter. En indirekte del av oppgaven er å forstå hvordan de aktuelle kommunikasjonsgrensesnittene fungerer.

6.1 SPI

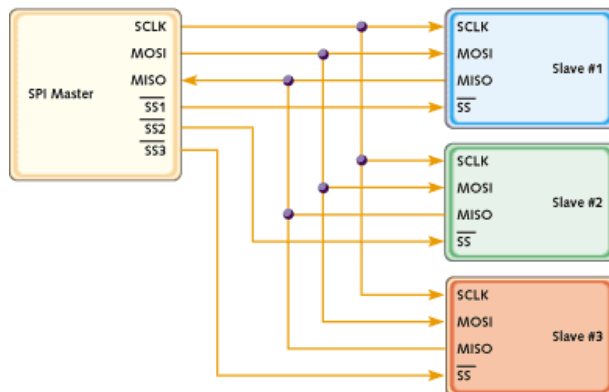
SPI (Serial Peripheral Interface) er en standard for seriell overføring innen digital elektronikk. Denne standarden ble grunnlagt av Motorola. Overføringen skjer synkronisert ved at et klokkesignal legges ved. Flere enheter kan kobles til på bussen, overføringen er full dupleks.

For hver enhet som tilkobles bussen er følgende signaler nødvendig:

- Data inn.
- Data ut.
- Klokkesignal.
- Et signal som forteller hvilken krets det skal overføres data til ("chip select").

6.1.1 Virkemåte

Data overføres mellom "master" og "slave". "Master" er den aktive delen i systemet, den generer klokkesignalet, klokkesignalet genereres kun når data skal overføres. "Slave" kan kun sende og motta data når "master" generer et klokkesignal. "Master" må sende data til "slave" for å kunne motta data fra denne. Mange komponenter kan kun brukes som "slave".



Figur 6.1; Eksempeloppkobling av SPI bussen [31].

Figur 6.1 viser et eksempel på sammenkobling av enheter ved hjelp av SPI bussen, her er det en "master" og de resterende er "slave". I SPI har man muligheten til å koble flere enheter på bussen, dette betinger at kun en "master" og en "slave" er aktive om gangen, portene på de andre enhetene blir da høyohmig ("tristate").

MISO: Master In Slave Out, data fra "slave" til "master".

MOSI: Master Out Slave In, data fra "master" til "slave".

SS: Slave Select, tilsvarer "chip select".

SCK: Shift Clock.

SPI bussen har fire moduser (0, 1, 2, 3), disse sier noe om hvordan data er "klokke" inn eller ut av en SPI enhet. Philips' I²C (Inter Integrated Circuit) bussen er nært beslektet med SPI. SPI bussen har i dette prosjektet også en annen funksjon, da den brukes for å programmere mikrokontrolleren mens den er loddet på kretskortet (In System Programming, ISP). [30] [17] [31]

6.2 USART

USART eller UART er seriell dataoverføring fra en sender til en (vanligvis) mottaker. UART sender data bitvis, og mottakeren samler så sammen disse bitene til bytes. UART står for Universal Asynchronous Receiver/Transmitter, mens USART står for Universal Synchronous- Asynchronous Receiver/Transmitter. Det vil si at UART kun kan overføre data asynkront (data som ikke er synkronisert med et klokkesignal), mens med USART har man mulighet til å overføre data som er synkronisert til et klokkesignal.

6.2.1 Virkemåte

Når senderen overfører dataord til UARTen for asynkron transmisjon, legges et startbit til i begynnelsen av hvert dataord, dette brukes for at mottakerens klokke skal synkroniseres med senderens. Klokken kan ikke ha en drift på mer enn 10 % i løpet av hele dataordet (dagens utstyr har ingen problemer med dette kravet). Etter startbitet sendes LSB (Least Significant bit) av dataordet først, etter dataordet kan det legges til et bit for paritetssjekk. På slutten legges det til minst ett stoppbit.

Serieporten på en PC fungerer på samme måte, da kalles logisk '1' for "mark", og logisk '0' for "space".

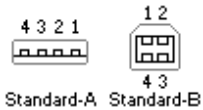
6.3 USB

USB (Universal Serial Bus) er et system med en vert og flere tilkoblede enheter, maksimalt kan 127 enheter kobles til. USB tillater ”hot swap”, det vil si til/fra koble mens PCen er oppe og går. Dette er i dag en av de mest populære grensesnittene på en PC.

Finnes i versjoner for hastigheter fra 1,5Mbit til 480Mbit.

Hver tilkoblet enhet har tilhørende ”pipes”, som er logiske kanaler. Dette er informasjonskanalene mellom vert og tilkoblede enheter. Data overføres serielt og differensielt. Hver ”pipe” har en maksimal lengde på typisk 2^n bytes. ”Pipes” er delt opp i fire kategorier avhengig overføringstype:

- *Kontroll overføring*; Brukt for korte enkle overføringer, som kommandoer og status forespørsler.
- *Isokron overføring*; Brukes ofte for sanntidsoverføring av audio/video.
- *Avbrudds overføring*; Brukes til enheter som trenger en kjapp respons, for eksempel mus.
- *Bulk overføring*; Brukes til overføringer som trenger stor kapasitet, for eksempel filoverføring.

Pin	Beskrivelse	USB kontakt
1	Buss-spenning	 Standard-A Standard-B
2	D-	
3	D+	
4	Jord	

Tabell 6.1; Oversikt over signalene i en USB kontakt [30].

Buss-spenningen er 5V nominelt, D- og D+ er differensiell overføring av data. USB grensesnittet kan levere opptil 500mA.

7 NMEA-0183 protokollen

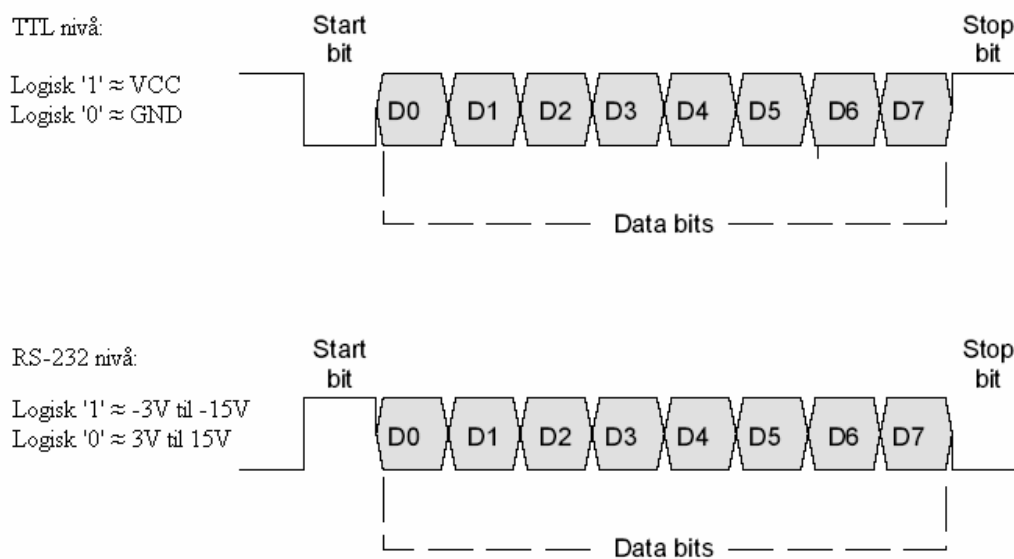
NMEA (National Maritime Electronics Association) er et organ for rådgivning og utvikling innen maritim elektronikk, en "non-profit" organisasjon. NMEA ble dannet av elektronikkforhandlere som møttes på New York Boat Show i 1957 for å diskutere forholdet til produsentene av maritim elektronikk. I begynnelsen av 1980 årene lagde NMEA en standard for digital overføring av informasjon mellom maritime enheter. Mange GPS-mottakere, ekkolodd, radarer og autopiloter for kommersielt bruk og andre maritime instrumenter som finnes i dag, benytter seriekommunikasjon basert på NMEA og RS-232.

I NMEA snakker man om forskjellige setninger eller meldinger som er ASCII strenger. Det er spesifisert forskjellige datasett til forskjellige brukerområder, for eksempel Loran-C, Omega, Transit, GPS og også produsentspesifikke datasatt. NMEA kilder sender ut utvalgte meldinger, på bakgrunn av hvordan de er konfigurert. Ofte sendes en gruppe NMEA meldinger ut med ett sekunds intervall. I NMEA-0183 er det en sender, og en eller flere mottakere.

Egentlig er NMEA-0183 basert på EIA-422 (tidligere RS-422), men i de fleste sammenhenger kan man regne det som RS-232 kompatibel. Mange NMEA kilder sender på TTL/CMOS nivå. NMEA-0183 enheter er konfigurert som enten sendere eller mottakere [25].

NMEA-0183 standardisert utstyr har et asynkront serielt grensesnitt med følgende parametere om ikke annet er sagt:

Baud rate: 4800 bps.
Data bits: 8 (mest signifikante bit er D0, se figur 7.1).
Paritet: Ingen.
Stoppbit: 1 eller flere.



Figur 7.1; Illustrasjon NMEA protokollen, TTL- og RS232-nivå [26].

Startbit er alltid logisk '0', mens stoppbit alltid er logisk '1'. De tilhørende spenningsnivåene kommer frem av figur 7.1. Det er ikke noen form for "handshake" i NMEA. Alle NMEA meldinger bruker 8 bits ASCII karakterer.

En NMEA-0183 melding ser slik ut:

\$IDIDX,data1,data2,data3,...,dataN*CS<CR><LF>

Datafelt	Forklaring
\$	Starten på meldinga, identifiserer senderen.
ID	Angir maskinvare type
IDX	Identifiserer type melding.
data1 til dataN	Selve informasjonen i melding, nummerert her fra 1 til N. Eksempel på slike data kan være: Hastighet, klokkeslett, posisjonsdata.
*	Markør for å separere sjekksummen fra meldinga.
CS	Checksum. Sjekksum for å sjekke hele datasettet.
<CR>	Slutten på datasettet. <CR>: Carriage Return.
<LF>	Slutten på datasettet. <LF>: Line Feed.

Tabell 7.1; NMEA-0183 melding [19].

Legg merke til at det brukes komma for å skille de forskjellige delene i meldinga. Hvis noen data i meldingen ikke er tilgjengelig, så blir de aktuelle dataene neglisjert, det vil si en tom meldingsplass, men kommane blir likevel sent, uten mellomrom. Noen produsenter følger ikke denne reglen og sender ut nuller istedenfor tomrom [32]. Siden informasjonen i meldingene (dataN) har variabel lengde, bør mottakeren basere seg på å telle antall komma for å trekke ut riktig informasjon fra meldingen. I motsetning til å telle posisjon til de aktuelle dataene.

Sjekksummen CS representerer "eksklusiv eller" (XOR) operasjon som involverer alle 8 databit (uten start- og stoppbit), mellom alle ASCII-karakterene mellom "\$" og "*". Det 8 bits resultatet er så delt opp i to 4 bits deler, og konvertert til den tilhørende heksadesimale verdien, som blir representert på ASCII form.

Meldinga kan ikke overstige 82 tegn, da regnes \$, <CR> og <LF> med.

NMEA standarden gir hver elektronikkprodusent mulighet til å ha egendefinerte meldinger. Da starter hver melding med "\$P", så kommer tre bokstaver som representere produsenten, deretter de ønskede data, resten av meldingen er standard NMEA-0183 (se tabell 7.1).

Eksempler på NMEA kilder:

Kilde:	ID:
LORAN-C mottaker	LC
OMEGA mottaker	OM
Integrert instrumentasjon	II
GPS-mottaker	GP
Integrert navigasjon	IN
Kartplottere	EC
DECCA	DE
Radar	RA
Ekkolodd	SS eller SD
Autopilot	AG eller AP
Vær instrumenter	WI

Tabell 7.2; Eksempler på NMEA sendere.

Et eksempel på NMEA meldinger ut fra en GPS-mottaker kan se slik ut:

```
14:11:04 $GPGSV,3,1,10,20,79,119,43,01,56,185,36,04,34,300,,25,33,067,43*72
14:11:04 $GPGSV,3,2,10,11,24,158,43,07,30,260,,13,18,212,,30,09,019,*7C
14:11:04 $GPGSV,3,3,10,05,07,341,,24,01,312,*7B
14:11:04 $GPGLL,5917.49861,N,01105.73975,E,141104.00,A,A*6A
14:11:04 $GPZDA,141104.00,01,05,2004,00,00*65
14:11:05 $GPRMC,141105.00,A,5917.49868,N,01105.73964,E,0.037,295.62,010504,,A*6B
14:11:05 $GPVTG,295.62,T,M,0.037,N,0.069,K,A*3C
14:11:05 $GPGGA,141105.00,5917.49868,N,01105.73964,E,1,4,11.00,44.4,M,40.4,M,,*67
14:11:05 $GPGSA,A,3,20,01,25,11,,,,,,15.05,11.00,10.27*33
14:11:05 $GPGSV,3,1,10,20,79,119,44,01,56,185,37,04,34,300,,25,33,067,43*74
14:11:05 $GPGSV,3,2,10,11,24,158,43,07,30,260,,13,18,212,,30,09,019,*7C
14:11:05 $GPGSV,3,3,10,05,07,341,,24,01,312,*7B
14:11:05 $GPGLL,5917.49868,N,01105.73964,E,141105.00,A,A*62
14:11:05 $GPZDA,141105.00,01,05,2004,00,00*64
14:11:06 $GPRMC,141106.00,A,5917.49872,N,01105.73963,E,0.048,297.73,010504,,A*6E
14:11:06 $GPVTG,297.73,T,M,0.048,N,0.088,K,A*39
14:11:06 $GPGGA,141106.00,5917.49872,N,01105.73963,E,1,4,11.00,44.3,M,40.4,M,,*6F
14:11:06 $GPGSA,A,3,20,01,25,11,,,,,,15.06,11.00,10.28*3F
```

Figur 7.2; Eksempel på NMEA meldinger ut fra en GPS-mottaker [26].

Oversikt over noen NMEA-0183 meldinger:

- RMC; Informasjon om klokkeslett, dato, posisjon, fart og kurs.
- GLL; Informasjon om posisjon og klokkeslett.
- GSV; Informasjon om satellittene (det vil si antall, satellitt ID, satellittenes elevasjon og azimuth, satellittenes signal-til-støy-forhold).
- VTG; Informasjon om kurs og fart.
- GSA; Informasjon om målenøyaktighet (DOP), antall satellitter og målemodus (2D eller 3D).
- ZDA; Informasjon om klokkeslett og dato.

7.1 Andre NMEA standarder

Det finnes en versjon som heter 0183-HS (HS = High Speed), som bruker et 3 leders grensesnitt og en baud rate på 38400.

Tidligere standarder: NMEA-0180 og NMEA-0182.

NMEA-2000 er en nyere standard som muliggjør 2 veis kommunikasjon med flere sendere og mottakere. Det vil si at det er flere ”mastere”, uten noen sentral kontroller. Dette nettverket er selvkonfigurerende.

8 Valg av komponenter

Valg av komponenter er en viktig del av denne oppgaven. I dette kapittelet gis det en bakgrunn for hvorfor de enkelte komponentene er valgt.

Prosjektgruppen gikk tidlig i gang med å se på valg av komponenter. Dette fordi det kan ta lang tid å produsere en ferdig krets. I tillegg må det være mulig å få tak i komponentene innen en rimelig tid for å unngå forsinkelser i prosjektgangen. Det ble brukt mye tid på internett for å finne ut hva slags komponenter som finnes på markedet. Et viktig poeng i denne oppgaven er at det skal være en liten håndholdt enhet, med batteridrift, derfor veier kriterier som liten fysisk størrelse og lite strømforbruk tungt. Spenningsnivåer for drivspenning og logiske nivåer er viktig med tanke på valg av komponenter. Det ble valgt en drivspenning på 3,3V for hele kretsen.

8.1 GPS-modul

Det skal brukes en ferdig GPS-modul som sender ut ferdig prosesserte data. De ferdig prosesserte dataene kommer som regel ut i form av NMEA meldinger, eller en annen protokoll. Prosjektgruppen skal gjøre bruk av NMEA protokollen.

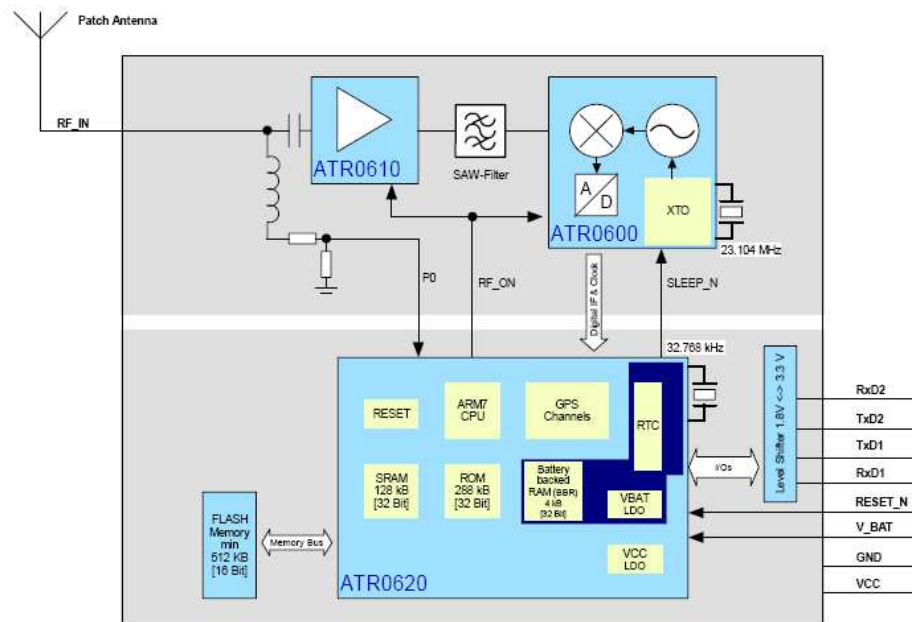
Det finnes GPS-moduler både med og uten ferdig påmontert antenne. Ved å bruke moduler med ferdig påmontert antenne, unngås problemer relatert til HF-signaler som skal gå fra antenna til GPS-modul, det er derfor ønskelig å benytte en GPS-modul med påmontert antenne. I de fleste tilfeller er det snakk om en lappantenne (patchantenne).

Etter vurderinger ble to GPS-moduler nærmere studert:

- SAM-LS fra ublox.
- uPatch100 fra Fastrax.

8.1.1 Kort presentasjon av SAM-LS

- Innebygd lappantenne.
- 16 parallelle følgekanaler.
- Frekvens: L1 (1575,41 MHz), C/A-kode.
- 2 stk. serieporter (UART) på TTL/CMOS nivå.
- 4 Mb flashminne.
- Spennning: 2,7V til 3,3V (drift og back-up spenning).
- Strømforbruk:
 - o Typisk: 56mA.
 - o Maksimalt: 125mA.
- Forskjellig operasjonsmodus, ettersom man ønsker kontinuerlig posisjonsangivelse eller lavere strømforbruk.
- Støtter følgende protokoller: NMEA, UBX (ublox sitt format), RTCM.
- Serieport1 er ferdig konfigurert til å sende følgende NMEA meldinger: GGA, GLL, GSA, GSV, RMC, VTG, ZDA, TXT. Sendes ved 9600bps, ellers standard NMEA protokoll.
- Serieport2 er ferdig konfigurert til å sende UBX pakker. Sendes ved 57600bps, 8bits data, ingen paritet og ett stoppbit.
- DGPS, SBAS/WAAS/EGNOS mulighet.
- Fysisk størrelse: 31.5 x 47 x 9.5 [mm].
- Vekt: 23 [g].
- Oppstartstider (se avsnitt 8.1.3):
 - o Kald start: 34 sekunder.
 - o Varm start: 33 sekunder.
 - o Het start: <3,5 sekunder.
- Sensitivitet:
 - o Innlåsing ("acquisition"): -138dBm.
 - o Sporing ("tracking"): -147dBm.
- Produsert av sveitsiske ublox.



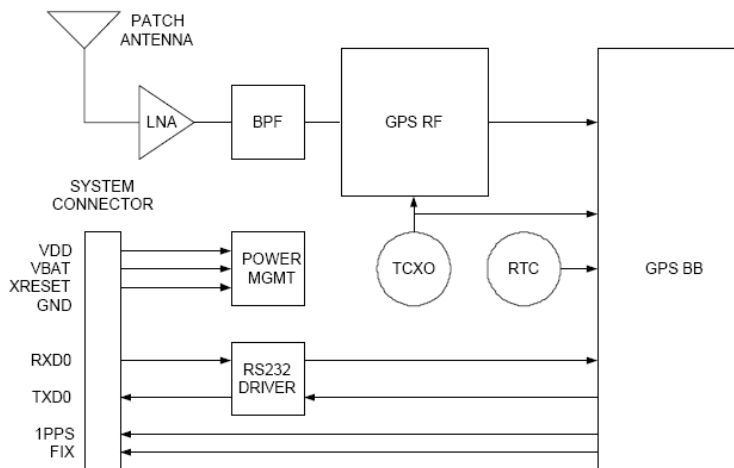
Figur 8.1; Blokkskjema av SAM-LS [14].

SAM-LS er basert på en ANTARIS GPS søkemotor (utviklet i samarbeid med Atmel).

I figur 8.1 er ATR0610 en RF LNA (Radio Frequency Low Noise Amplifier) IC, ATR0600 er en RF Front-end IC, det vil si en nedkonverter og en A/D omformer. ATR0620 er basisbåndenheten.

8.1.2 Kort presentasjon av uPatch100

- Innebygd lappantenne.
- 12 parallelle følgekanaler.
- Frekvens: L1 (1575,42 MHz), C/A-kode.
- Spenning: 3,3V til 5,5V (drift og back-up spenning).
- Strømforbruk:
 - o Innlåsing ("acquisition"): 76 mA.
 - o Sporing/navigasjon ("tracking"): 44mA.
- Fysisk størrelse: 28 x 28 x 7,0 [mm].
- Vekt: 20 [g].
- 1 UART som er ferdig konfigurert for å sende ut følgende NMEA meldinger:
GGA, GSA, GSV, RMC. Disse sendes med 4800bps, standard NMEA.
- uPatch100 kan konfigureres til å sende andre NMEA meldinger.
- Strømsparemodus kan muliggjøres ved å fjernedrivspenning, og kun opprettholde back-upspenning.
- 1 PPS (Puls per Sekund) utgang tilgjengelig, synkronisert til GPS-tid.
- Utgang som indikerer en gyldig posisjonsangivelse.
- Sensitivitet:
 - o Innlåsing ("acquisition"): -139dBm.
 - o Sporing ("tracking"): -152dBm.
 - o Navigasjon: -150dBm.
- Oppstartstid (se avsnitt 8.1.3):
 - o Kald start: 38 sekunder.
 - o Varm start: 32 sekunder.
 - o Het start: <8 sekunder.
- Ikke DGPS, SBAS/WAAS/EGNOS mulighet.
- Har en fast oppdateringsrate på 1 Hz.
- Produsert av finske Fastrax.



Figur 8.2; Blokkskjema av uPatch100 [5].

uPatch100 er basert på SONY mottakerarkitektur, med en CXA3355 RF chip, og en CXD2956 basisbåndenhet. Figur 8.2 illustrerer oppbyggingen av uPatch100.

8.1.3 Forklaring

Kald start er når mottakeren har kjennskap til almanakken (det vil si grov informasjon om banedata som funksjon av tid) men ikke til sin egen posisjon og tid.

Varm start er når mottakeren vet omtrent hvor den er, men mottakeren har vært avslått i lengre tid og kan være transportert til et annet sted.

Het start er når mottakeren har nesten full kjennskap til egen posisjon og tid.

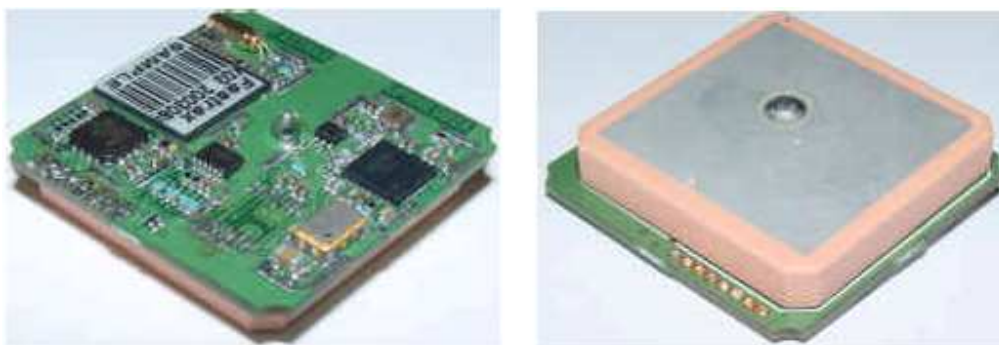
I tillegg finnes det en "*Factory start*" for GPS-mottakere, det vil si at mottakeren ikke har noe kjennskap til almanakken i hele tatt, denne tar det i overkant av 12 minutter å laste inn.

8.1.4 Drøfting

Det som er viktig for valg av GPS-modul i denne oppgaven er liten fysisk størrelse, lavt strømforbruk, tilgjengelighet, samt god følsomhet/sensitivitet.

Ut fra de oppgitte spesifikasjonen for uPatch100 og SAM-LS, så gir det klare indikasjoner på at uPatch100 har et mindre strømtrekk og bedre følsomhet. I tillegg har uPatch100 en mindre fysisk størrelse enn SAM-LS.

Det ble derfor, ut fra ovenstående kriterier, bestemt at uPatch100 versjon C4 skulle benyttes.



Figur 8.3; Bilder av uPatch100 [5].

8.2 Mikrokontroller (MCU)

Veileder hos FFI, Leif Hanssen, hadde uttrykt et ønske om at det skulle benyttes en 8-bits RISC (Reduced Instruction Set Computer) mikrokontroller fra Atmel. Ettersom prosjektgruppen hadde noe erfaring med programmering av mikrokontrollere fra Atmel, ble det vedtatt å benytte en av disse.

Kriteriene for valg av mikrokontroller er at den har egnede grensesnitt mot andre enheter, liten fysisk størrelse og lite strømforbruk. Uansett valg av mikrokontroller, så er typisk strømforbruk for en mikrokontroller, mindre enn strømforbruket for en GPS-mottaker.

Nødvendig grensesnitt:

- 2 stk. UART, for bruk mot GPS-modulen og USB konverteringskretsen.
- 2 stk. I/O porter for bruk mot display, det vil si 16 I/O pinner.
- SPI for bruk mot flashminne, og eventuelle treghetskomponenter.
- 4 stk. I/O pinner for bruk til trykknapper.
- 1 stk. I/O port med ADC (analog til digital omformer) mulighet, for bruk mot eventuelle treghetskomponenter som har analog utgang, det vil si 8 pinner.
- 2 stk. I/O pinner for styring av bakgrunnsbelysning på display og strømsparemodus for GPS-modulen.
- Ekstra I/O porter for bruk sammen med SPI bussen, og til treghetskomponenter.

Prosjektgruppen har enda ikke bestemt seg for treghetskomponenter, derfor tas det med noen ekstra I/O porter slik at valg av treghetskomponenter ikke begrenses av tilgjengelig I/O porter på mikrokontrolleren.

ATmega128 ble valgt som mikrokontroller. Det finnes en versjon av denne mikrokontrolleren som takler en drivspenning på 3,3V, denne heter ATmega128L. Ved å benytte 3,3V driftsspenning oppnås det et lavere strømforbruk på mikrokontrolleren [1]. ATmega128L tilhører AVR familien av mikrokontrollere.

8.2.1 Utvalgte fakta om ATmega128L

- Drivspenning: 2,7V til 5,5V.
- Klokkehastighet: Opptil 8 MHz.
- Intern RC oscillator.
- Forskjellige "sleepmodes" for strømsparing og støyreduksjon.
- Grensesnitt: 2 stk. USART, 8 stk. 10 bits ADC, 8 PWM (puls bredde modulerte) kanaler, JTAG, SPI, TWI, ISP (In System Programming) programmering via SPI, Analog komparator.
- Teller/"timere": 2 stk. 8bits tellere og 2 stk. 16bits tellere.
- 128KB reprogrammerbart flashminne.
- 4KB EEPROM.
- 4KB SRAM.
- 8 bits Harvard RISC arkitektur.
- Innebygd "pull-up-motstander".
- Strømtrekk: Typisk 10mA ved 3,3V og 8MHz klokkefrekvens.



Figur 8.3; ATmega128 [34].

8.3 Minne for lagring av posisjoner

I prosjektet skal det lagres posisjoner for senere å overføres til en PC, derfor trengs det et ikke-flyktig minne, det skal brukes et minne som basert på flash-teknologi.

Ved valg av flashminne er lagringskapasitet, tilgjengelighet, fysisk størrelse og grensesnitt opp mot mikrokontroller viktige kriterier. Leif Hanssen nevnte på et tidlig stadium at det er ønskelig å lagre langt mer enn 10000 "tracking-punkter", det vil si lagrede posisjoner. Atmel sin AT45DBxxxx DataFlash serie ble studert nærmere. Disse har mulighet for et serielt grensesnitt mot mikrokontrolleren. AT45DB321C, på 32Mbit, er tilgjengelig i små pakker. Denne kan kobles til mikrokontrolleren via SPI grensesnittet.

Fysisk størrelse er 8 x 6 mm. for en CASON 8 pakke.

8.3.1 Grovt overslag for nødvendig lagringskapasitet

Det er ønskelig å lagre 2D-posisjon, tid og muligens høyde. Dette hentes blant annet fra NMEA meldinger som er 8 bits ASCII-tegn. Vi lagrer da lengde- og breddegrader, klokkeslett, dato, og muligens høyde.

Lagrings format:

Lengdegrad (longitude):	XXXX.XXXX	9 Byte
Breddegrad (latitude):	XXXXX.XXXX	10 Byte
Klokkeslett:	XXXXXX	6 Byte
Dato:	XXXXXX	6 Byte
Eventuell høyde:	XXXX	4 Byte

I tillegg trengs det markører som separere datasettene, totalt ca. 40 Byte, som tilsvarer 320 bit.

$$\frac{32Mbit}{320bit} = 100000 \text{ posisjoner.} \quad (\text{Ligning 8.1})$$

En lagringskapasitet på 10000 posisjoner ses på som et minimum. 32Mbit lagringskapasitet som gir 100000 posisjoner eller ”tracking-punkt”, ses derfor på som tilstrekkelig.

På bakgrunn av dette velges AT45DB321C fra Atmel som minne for lagring av posisjoner. Andre flashminner har vært vurdert, blant annet M29W320DB70N6E og M25P80 fra ST.

Det viste seg vanskelig å få tak i AT45DB321C i tide, derfor ble M25P80 implementert. Denne har en lagringskapasitet en 8Mbit.

$$\frac{8Mbit}{320bit} = 25000 \text{ posisjoner} \quad (\text{Ligning 8.2})$$

Dette gir kun 25000 lagrede posisjoner, men dette er tilstrekkelig i en prototyp.

Kretskortet er i midlertidig konstruert slik at det på et senere tidspunkt kan byttes til AT45DB321C.

8.4 Andre komponenter

8.4.1 Spenningsregulator

Det satses på å bruke et batteri med spenning 3,7V, og det skal brukes en drivspenning på 3,3V på kretskortet. Det trengs derfor en LDO (Low DropOut) regulator med 3,3V utspenning. Liten fysisk størrelse og mulighet for å levere nok strøm er viktige poeng i valg av spenningsregulator. LP3985 fra National Semiconductor ble valgt, denne kan levere 150mA, og har en ”dropout spenning” (spenningsfall over regulatoren) på typisk 60mV og maksimalt 100mV. Maksimum inngangsspenning på denne regulatoren er 6V. Andre regulatorer som ble sett på som passende: KF33BD fra ST, TPS79430 fra Texas Instruments og LP3997 fra National Semiconductor.

8.4.2 Batteri

Et 3.7V Lithium-Polymer batteri fra VARTA er valgt på bakgrunn av ønske fra FFI. Batteriet har en kapasitet på 900mAh.

8.4.3 Krets for konvertering til USB grensesnitt

Det var ønskelig fra oppdragsgiver sin side at grensesnittet mot PCen er USB. Utvikling og implementering av en USB-driver i programvaren på mikrokontrolleren er omfattende og tidkrevende, for å unngå dette ble det valgt å benytte en krets som konverterer et UART-grensesnitt til USB. FT232BM fra FTDI er en slik krets. Tore Johan Høiland har erfaring med bruk av FT232BM, derfor ble denne valgt.

8.4.4 Display

FFI hadde et egnet display som det var ønskelig at ble benyttet.

MG1060A-SGL-4 er et grafisk display (100 x 60 punkter) fra Everbouquet som det kan skrives parallelt eller serielt til. Displayet har en egen displaykontroller som sitter på displayet, SED1560A.

8.4.5 Diverse

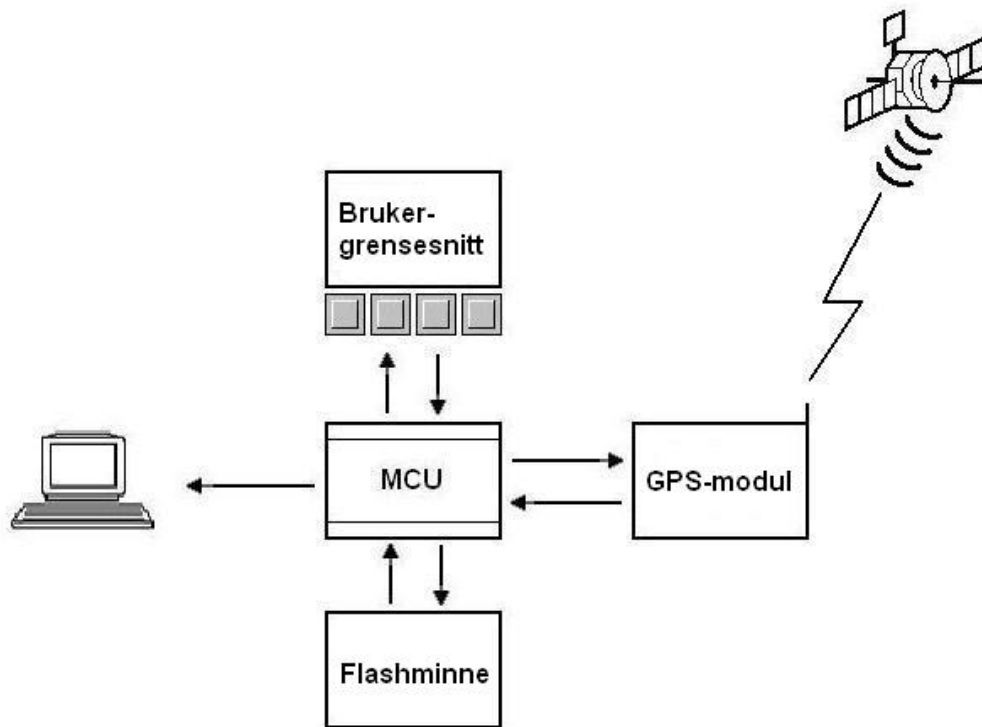
Bakgrunnsbelysning på display styres ved å bruke en P-kanal D-MOS transistor, BSS84. For å kjøre GPS-modulen i strømsparemodus fjernes driftsspenningen (VDD) ved hjelp av en egen spenningsregulator, LP3985 fra National Semiconductor, samme type som systemets hovedregulator. Denne styres av mikrokontrolleren.

For å kunne lade batteriet ved hjelp av strømmen fra USB-bussen, benyttes det egen ladekrets. Valg av ladekrets (BQ24013 fra Texas Instruments) ble gjort av Leif Hanssen. Ladestrømmen fra BQ24013 styres av mikrokontrolleren ved hjelp av en N-kanal D-MOS transistor, BST82.

I tillegg er det valgt komponenter som tilkoblingskontakter, trykknapper, krystall, avkoblingskondensatorer, motstander og spoler.

9 Design og implementering

9.1 Systembeskrivelse



Figur 9.1; Blokkdiagram for systemet.

Figur 9.1 er et blokkdiagram som viser hovedtrekkene i løsningen på prosjektet. Retningen på pilene viser informasjonsflyten. Mikrokontrolleren (MCU) tar imot informasjon fra GPS (som NMEA meldinger), analyserer denne informasjonen, viser posisjon og tid på displayet, og lagrer posisjon og tid ("tracking-punkt") i flashminnet. Lagring av "tracking-punkt" i flashminnet skjer hvert sekund. Det er kun nye posisjoner som lagres, det vil si at hvis mottakeren befinner seg på en plass over lengre tid, så lagres ikke posisjonen før man får inn ny posisjon (nye lengde- og breddegrader). Brukergrensesnittet er beskrevet i avsnitt 9.4.

9.2 Maskinvare

Her følger en forklaring på de løsningene av prosjektet som omhandler maskinvaren. Det er kun hovedkomponentene som blir behandlet i dette avsnittet. For detaljert informasjon om oppkobling av de forskjellige enhetene vises det til kretsskjema som er lagt ved som vedlegg A. Betegnelser i parentes referer til kretsskjemaet.

9.2.1 Mikrokontroller (MCU)

Følgende enheter er tilkoblet ATmega128 (U9):

- Eksternt 8 MHz krystall for drift av MCU (OS3).
- Eksternt 32768 Hz krystall for drift av sanntidsklokke, og styring av lagringstidspunkt i flashminnet (OS2).
- GPS-modul, uPatch100 (U3).
- Flashminne via SPI bussen, M25P80 (U10).
- USB konverteringskrets, FT232BM (U4).
- Krets for lading av batteriet, BQ24013 (U8).
- Display, MG1060A-SGL (P2).
- ISP grensesnitt for programmering av mikrokontrolleren (J1).
- Tilkoblinger for utvidelse til treghetsnavigasjon (J2 og J3).
- 4 knapper for brukergrensesnitt (SW1, SW2, SW3 og SW4).

Avcc (pinne 64) er koblet til et filter mot støy ved bruk av den innebygde analog til digital konverteren. Knapper for brukergrensesnitt trenger ikke ha "pull-up" motstander siden ATmega128 har innebygde "pull-up" motstander.

9.2.2 USB konvertering

FT232BM (U4) er en krets for konvertering mellom UART grensesnitt og USB.

Dokumentasjon fra FTDI er god, både med tanke på datablad og "application notes". Det henvises til kretsskjema (vedlegg A) for implementering av USB-konverteringskretsen.

Et EEPROM minne (U5) er lagt til kretsen for å lagre data som identifiserer kretsen for PCen. I tillegg er det lagt til 3 statuslysdioder for å indikere følgende tilstander:

- Kretsen er tilkoblet en USB-buss, Lysdiode D2 på kretsskjemaet.
- Data sendes til PC, lysdiode D3 på kretsskjemaet.
- Data mottas fra PC, lysdiode D1 på kretsskjemaet.

For å konfigurere FT232BM benyttes programmet Mprog 2.3, tilgjengelig fra FTDI sin hjemmeside. For å benytte Mprog 2.3, må man laste inn en såkalt D2XX driver på PCen, dette gir Mprog 2.3 mulighet til aksessere FT232BM via DLL funksjoner, henviser til [29] for mer detaljert beskrivelse av dette.

Det benyttes en VCP-driver (Virtual Com Port) på PCen, dette gjør at FT232BM ser ut som en vanlig COM-port på PCen. Da kan et enkelt terminalprogram benyttes på en PC for motta lagrede "tracking-punkt" fra systemet, for eksempel HyperTerminal.

FT232BM kobles til UART1 på mikrokontrolleren. System og PC kommuniserer med baudrate på 115200bps.

9.2.3 Lading av batteriet

BQ24013 (U8) er en krets for lading av batteriet. Denne er koblet opp mot mikroprosessen slik at ladingen kan overvåkes og styres av mikrokontrolleren.

Transistor Q2, som styres av mikrokontrolleren, brukes får kunne velge to forskjellige maksimale ladestrømmer, ca. 100 mA og ca. 480 mA, refererer til ligning 12 i [7].

Ladespenningen er på 4,2V. Under lading av batteriet benyttes spenningen fra USB-porten på en PC.

STAT1 (pinne 3) og STAT2 (pinne 4) på BQ34013 indikerer ladestatus, se tabell 9.1.

Ladestatus	STAT1	STAT2
Holder på å lade	ON	OFF
Lading ferdig	OFF	ON
Ingen batteri tilkoblet, ladefeil, sleepmode.	OFF	OFF

Tabell 9.1; Ladestatus for BQ24013 [7].

For mer detaljert virkemåte av BQ24013, henvises det til tilhørende datablad.

9.2.4 GPS-modul

GPS-modul (U3) kobles til UART0 på mikrokontrolleren, i tillegg kobles utgangene 1PPS og GPIO9 til mikrokontrolleren. 1PPS gir 1 puls per sekund ut. GPIO9 angir om den aktuelle posisjonen gitt i NMEA meldingen er gyldig (logisk '1'), ved innlåsningsmodus ("acquisition") gir den ut et signal 0,5Hz, i sporemodus ("tracking") gir den ut 1Hz. Den har to strømforsyningstilkoblinger, VDD og Vbat. Ved fjerning av VDD går GPS-modulen i søvnmodus, mens Vbat er back-upspenning. Søvnmodus er aktuelt i forbindelse med strømsparing. Spenningen inn på VDD har en egen regulator (U7) som kontrolleres av mikrokontrolleren slik at uPatch100 kan gå i søvnmodus. uPatch100 kan konfigureres over UART0 ved hjelp av Sony ASCII Protocoll, denne muligheten er ikke benyttet i denne delen av prosjektet, men kan være nyttig senere med tanke på utvidelse av prosjektet i masteroppgaven. GPS-modulen og mikrokontrolleren kommuniserer med en baudrate på 4800bps.

9.2.5 Display

Displayet (MG1060A-SGL) kobles til PORTA og PORTC på mikrokontrolleren. PORTA brukes for kontrolldata til displayet, mens PORTC brukes for å sende data som skal vises på displayet. Bakgrunnsbelysningen kan trekke 50mA til 100mA, noe som er uakseptabelt mye i denne sammenheng, det er gjennomført to tiltak for å redusere dette strømforbruket:

- Det er det lagt inn en motstand (R16) for å begrense strømmen til bakgrunnsbelysningen.
- En transistor (Q1), som styres av mikrokontrolleren, brukes for å skru av og på bakgrunnsbelysningen på displayet.

9.2.6 Kommentarer til design av maskinvare

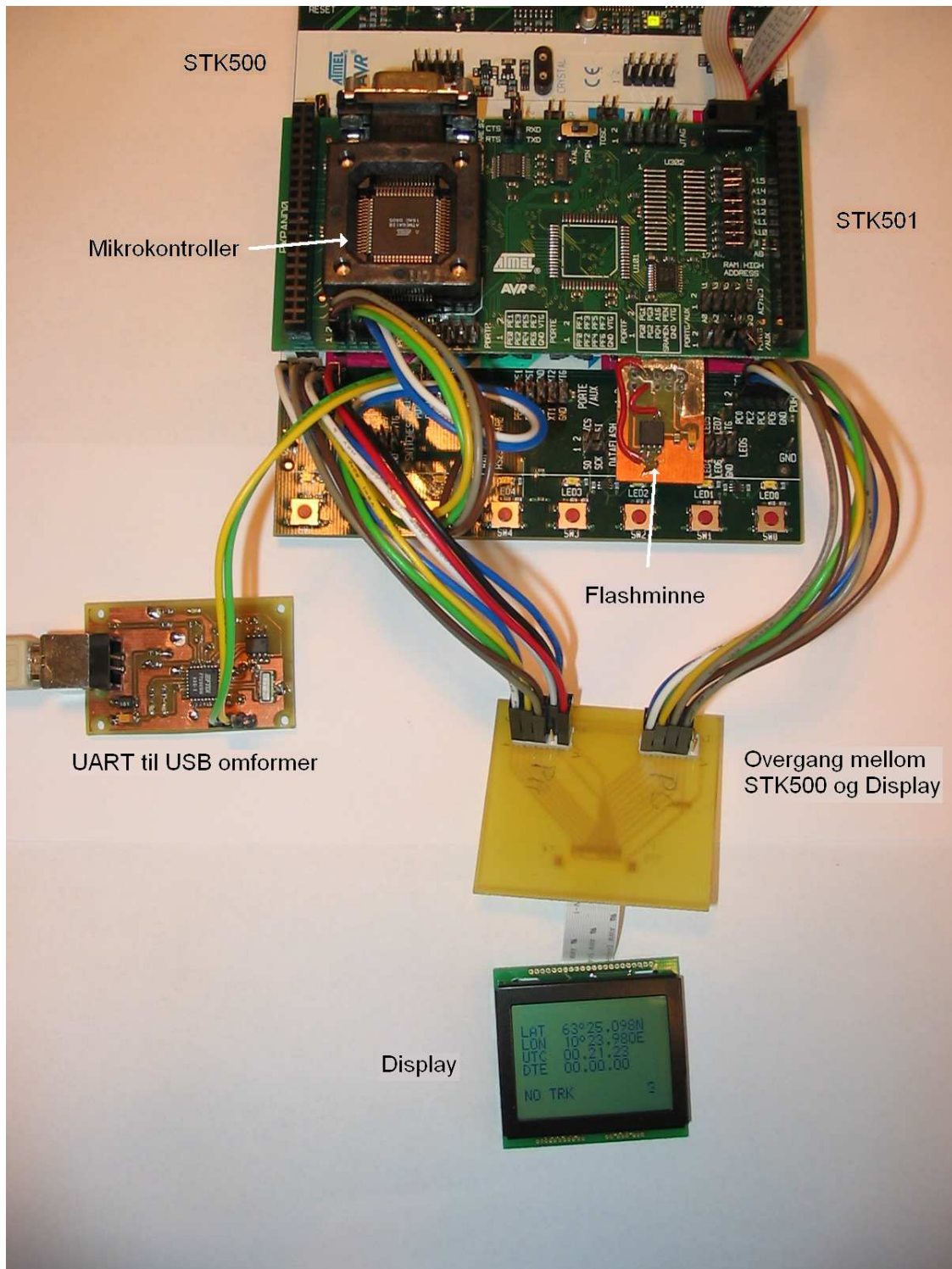
Thorkild Kåsa ved FFI designet kretskortet etter prosjektgruppens ønsker. Kretskortet ble etset hos Elprint. Det meste av loddearbeidet ble utført av FFI. Kretsskjema, produksjonsgrunnlag og komponentplassering legges ved som vedlegg A. Det er lagt inn avkoblingskondensatorer. Kretskortet er forøvrig et to-lags kort av typen FR4. Valg av komponenter og utvikling av kretsskjema var en prosess som tok lengre tid en planlagt.

9.3 Programvare

Programvareutvikling er en stor og tidkrevende del av prosjektet. For å komme tidlig i gang med programvareutvikling, ble det satt opp ett testoppsett for å kunne utvikle programvaremoduler slik at de forskjellige komponentene kunne utnyttes best mulig i prosjektet.

Utstyr som har vært brukt i testoppsettet for utvikling og testing av programvaren:

- STK500 og STK501 (simuleringskit fra Atmel).
- AVRButterfly (demokort fra Atmel).
- Trimble Ensign XL GPS (GPS-mottaker for å generere NMEA meldinger).
- MG1060A-SGL-4 (display).
- Kretskort for å kunne koble STK500 til displayet. Se vedlegg B.
- Kretskort for konvertering mellom UART og USB. Se vedlegg B.
- Kretskort med M25P80 flashminne. Se vedlegg B.
- Oscilloskop og multimeter.
- PC med C-kompilator.
- Strømforsyning.



Figur 9.2; Utstyr oppkoblet for utvikling og testing av programvare.

Figur 9.2 er en illustrasjon som viser hvordan utstyret er koblet opp under utvikling av programvare. Kretser for UART til USB omformer, flashminne og overgang mellom STK500 og display er designet av FFI. Under utvikling av programvare ble forskjellige NMEA meldinger brukt, men RMC meldingen er den som benyttes i den ferdigutviklede versjonen.

Programmeringen av mikrokontrolleren foregår i programmeringsspråket C, dette fordi mikrokontrolleren i utgangspunktet er designet for kompilert C-kode (i likhet med resten av AVR familien). I tillegg har begge prosjektmedlemmene noe erfaring med C-programmering. Leif Hanssen anbefalte kompilatoren CodeVisionAVR fra HPinfotech, versjon 1.23.7 Standard benyttes. Programmering av mikrokontrolleren skjer via SPI bussen mens mikrokontrolleren er montert i den aktuelle kretsen, dette kalles In System Programming (ISP). For å overføre programmet fra PCen til mikrokontrolleren benyttes AVR Studio versjon 4.11, Build 410, Service Pack 3.

Bakgrunnsinformasjon for det programmeringstekniske er innhentet fra [1], [2], [5], [6], [7], [8], [11], [12], [16], [17], [18], [20], [27], [26], [28] samt diverse eksempler på kildekode innhentet fra internett. I tillegg har Leif Hanssen bidratt med råd og hjelp under programmeringen.

9.3.1 Lagringsformat

Et hensiktsmessig lagringsformat for data ("tracking-punkt") som lagres i flashminnet ble utarbeidet. Lagringsformatet i flashminnet er:

Breddegrad(N/S),Lengdegrad(E/W),klokkeslett,dato;<CR><NL>

<CR> = Carriage Return, ASCII tegn: D (heksadesimalt).

<NL> = New Line, ASCII tegn: A (heksadesimalt).

Klokkeslett er Universal Time Coordinated (UTC) tid. Komma brukes for skille de forskjellige dataene, mens semikolon brukes for indikere slutten på et datasett, sammen med <CR> og <NL>.

Lengde- og breddegrad er angitt i WGS-84 (World Geodetic System 1984) datumet, ECEF (Earth Centered Earth Fixed) koordinater.

<CR> og <NL> er i først og fremst tatt med på grunn av det gir en oversiktlig fremstilling av dataene i et terminalprogram på PCen. Informasjon om høyde er ikke tatt med.

Eksempel på et "tracking-punkt":

6325.1640N,01024.0000E,143850,071105;

Som betyr:

- Breddegrad: 63° 25,1640 minutter Nord
- Lengdegrad: 10° 24,0000 minutter Øst
- Klokkeslett: 14:38:50
- Dato: 07.11.05

9.3.2 Overvåkning av batterispenning

Det er lagt til en funksjon som gir brukeren informasjon om batteriets tilstand.

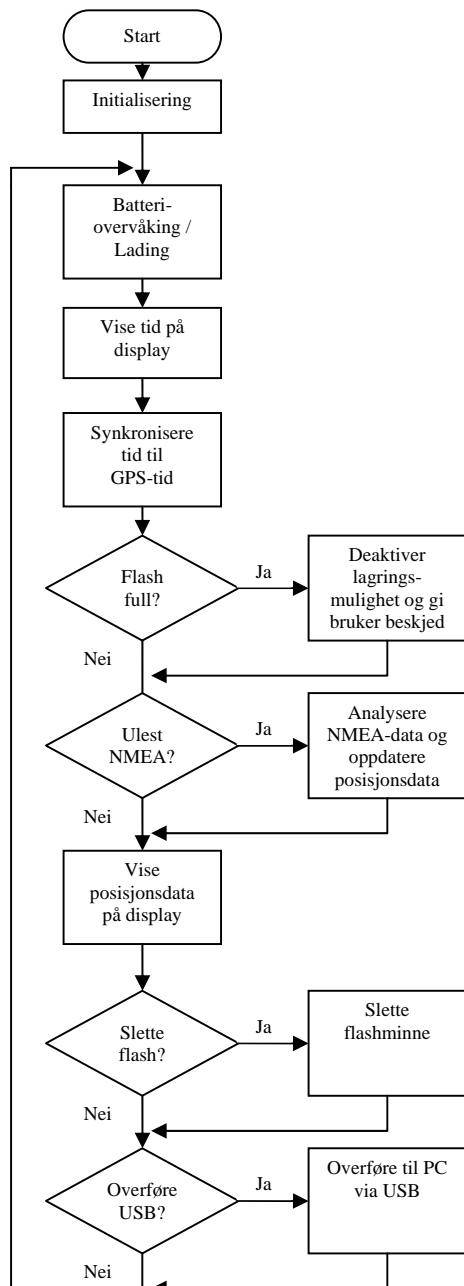
Overvåkningen av batteriet er realisert ved hjelp av en analog til digital omformer på mikrokontrolleren. Batterispenningen vises i displayet.

9.3.3 Kildekoden

Kildekoden er ikke lagt ved rapporten, men det er laget et forenklet flytdiagram for å illustrere hovedløkka i programmet, se figur 9.3. Den komplette kildekoden er for øvrig på ca. 3600 linjer.

Programmet er delt opp i forskjellige filer for å gi en god oversikt:

- *main.c*; Innholder hovedløkka i programmet, alle avbruddsrutiner, og funksjoner for lading/overvåking av batteri og konvertering fra tall til BCD (Binary Coded Decimal). I tillegg ligger det meste av initialiseringen her.
- *display.c* og *display.h*; Innholder driver for display, det vil si initialisering av display, skriving av ASCII tegn til displayet, og sletting av tegn på displayet.
- *nmea.c* og *nmea.h*; Innholder funksjoner for analysering av NMEA strenger, vise posisjon på displayet, og finne ut om et nytt "tracking-punkt" er registrert.
- *time.c* og *time.h*; Innholder sanntidsklokka, funksjon for synkronisering av sanntidsklokka opp mot GPS (UTC tid), og funksjoner for å vise klokkeslett og dato på displayet.
- *flash.c* og *flash.h*; Innholder funksjoner for kommunikasjon med flashminnet over SPI bussen. Det vil si lagring, sletting, og innhente statusinformasjon fra flashminnet.



Figur 9.3; Flytdiagram for hovedløkka.

Figur 9.3 er et flytdiagram som i grove trekk illustrerer hovedløkka i kildekoden. Legg merke til at funksjoner som innhenting av data fra GPS, innhenting data fra brukergrensesnitt, lagring i flashminne og sanntidsklokke ikke inngår i hovedløkka. Dette er realisert i avbruddsrutiner ("interrupt").

9.3.4 Kommentarer til programmeringen

Programmeringsdelen var, som forspeilet, tidkrevende. Spesielt utvikling av displaydriveren, og utarbeidelse av en fornuftig algoritme for å motta og analysere NMEA meldinger.

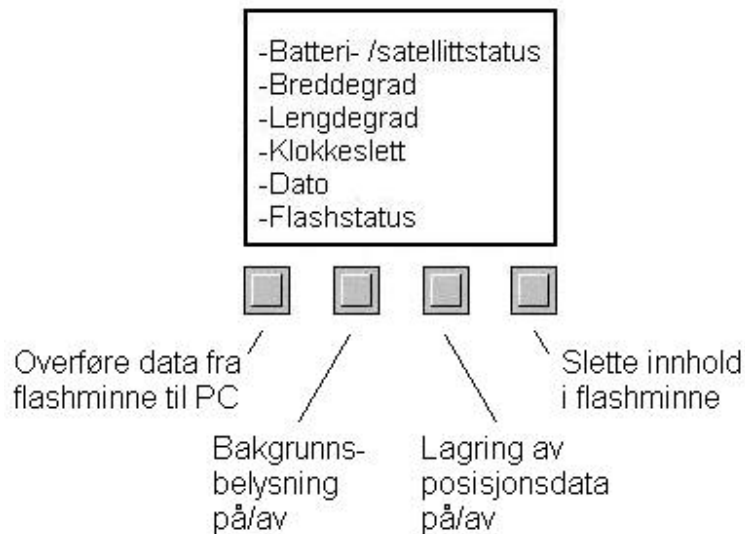
Under utvikling av programvaren var det opprinnelig tenkt at tid (klokkeslett og dato) på displayet kun ble baserte på data direkte fra NMEA meldinger. I midlertidig viste det seg at tiden på displayet (UTC), kun ble oppdatert cirka hvert 3. sekund. Det er ønskelig at tiden oppdateres jevnlig hvert sekund på displayet. Dette ble løst ved at det ble implementert en sanntidsklokke i programvaren, basert på "TIMER/COUNTER0" i mikrokontrolleren. Sanntidsklokka (kalles "RTC" i kildekoden) blir jevnlig synkronisert med GPS (UTC tid).

Posisjonsdata ("tracking-punkt") lagres, som beskrevet tidligere, hvert sekund. Hvis det er ønskelig at det skal lagres med en lavere frekvens enn hvert sekund, kan dette enkelt implementeres i kildekoden, for eksempel hvert 5, 10. eller 30. sekund. Dette kan være nyttig hvis man har behov for lagring av "tracking-punkt" over lengre tid.

For mer konkret beskrivelse av programvaren, henvises det til kildekoden. I kildekoden er det forsøkt lagt til gode kommentarer for å beskrive virkemåten.

9.4 Brukergrensesnitt

Mottakeren er utstyrt med et grafisk display og 4 knapper som grensesnitt mot bruker. Displayet har et skjermbilde med 6 linjer som gir forskjellig informasjon om GPS-mottakerens status. De 4 knappene registrerer brukerens kommandoer (se figur 9.4).



Figur 9.4; Brukergrensesnitt for GPS-mottaker.

Det er lagt til en egen knapp for å kunne slå på bakgrunnsbelysningen når dette er nødvendig. De tre resterende knappene er til funksjoner knyttet til håndtering av posisjonsdata ("tracking-punkt"). Når data lagres, ikke lagres, overføres til PC eller slettes vil dette vises i "Flashstatus" linjen henholdsvis som "TRK", "NO TRK", "USB" og "DELETE".



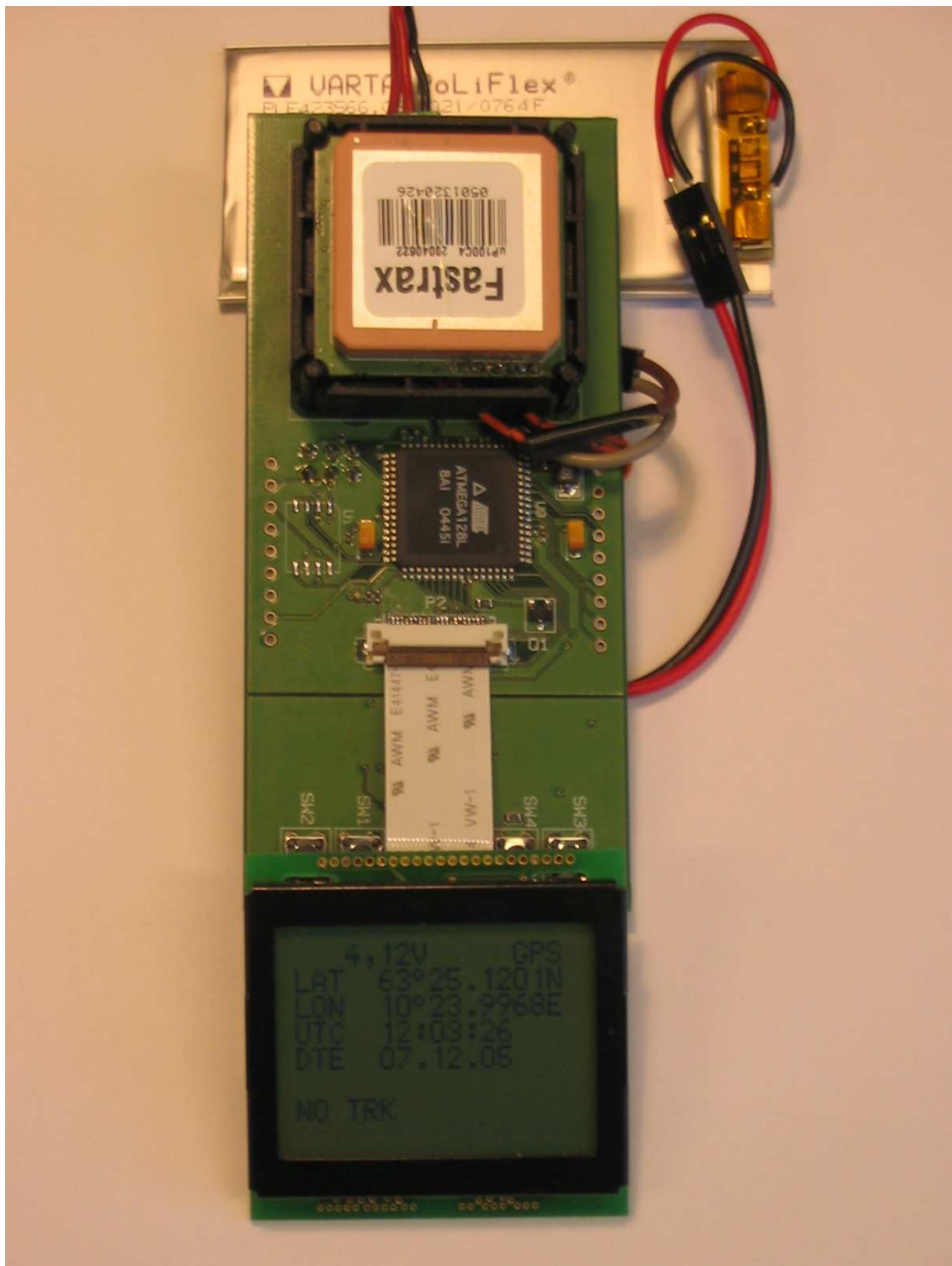
Figur 9.5; Bilde av display med navigasjonsdata.

Satellittstatus vises i form av "GPS" oppe i høye hjørne når mottakeren har gyldig utregnet posisjon. For batteristatus er det laget et voltmeter for å kunne overvåke batterispenningen. Batteristatus kan alternativt vises som en enkel status melding, for eksempel "BATTERY OK", "BATTERY LOW" eller "CHARGING". Dette kan gjøres i forbindelse med en overvåkingsfunksjon som legger de forskjellige komponentene i søvnmodus når spenningen faller under en gitt grense.

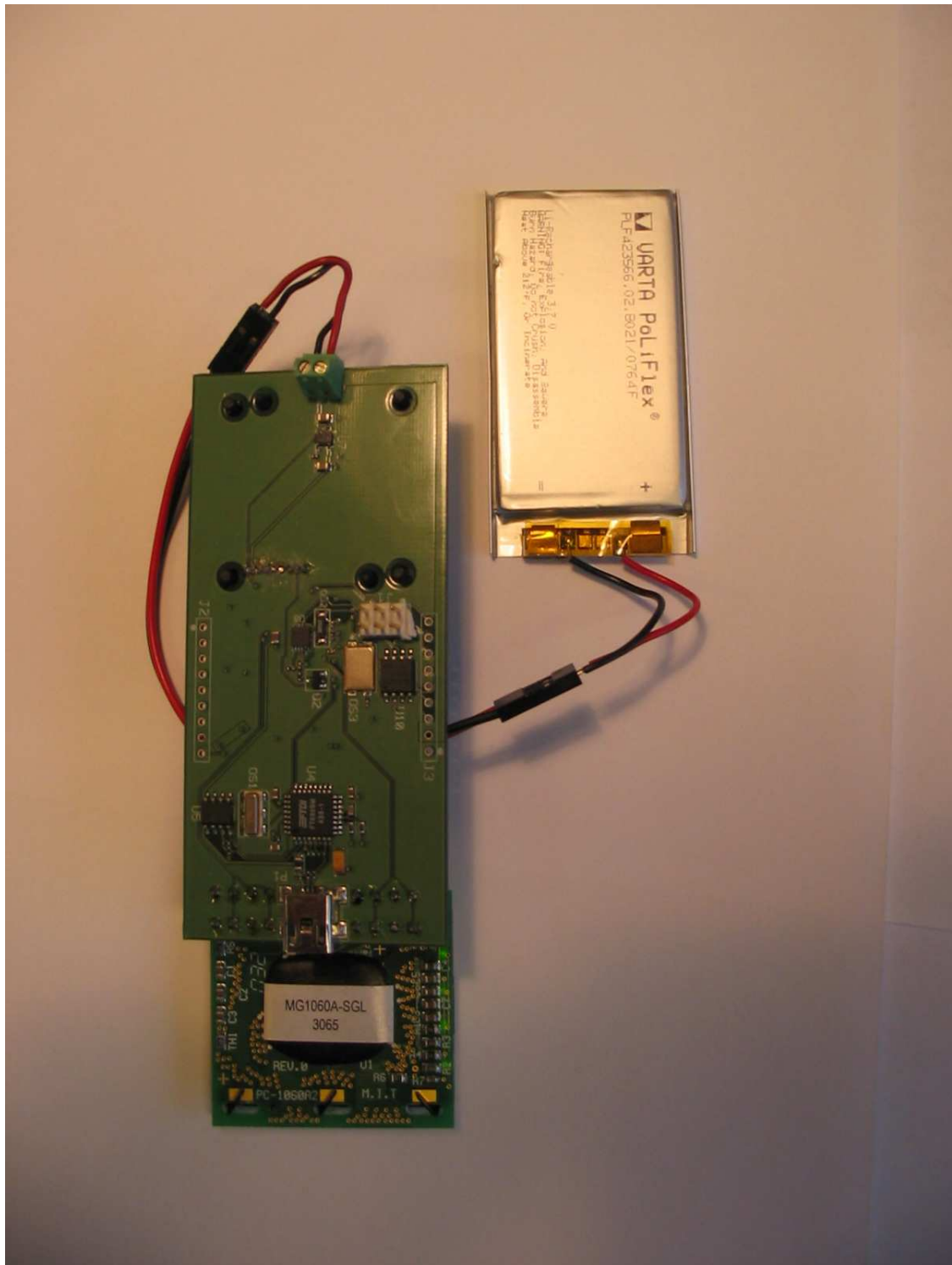
Det er ikke laget noen topptekst til de forskjellige knappene i displayet, det er imidlertid mulighet til dette, da det er ubenyttede linjer igjen i bunnen av displayet. Dette kan være nyttig hvis systemet utvides slik at samme knapp kan ha flere funksjoner. Ettersom det er et grafisk display er valgfriheten stor, med tanke på størrelse på bokstaver og mengde informasjon som vises. Displayet har 100 x 60 piksler. Display driveren er laget slik at hver bokstav er 8 piksler høy og 6 piksler bred (inkludert mellomrom). Dette gir 7 linjer av 16 bokstaver. I tillegg kan symboler og spesielle tegn lages, for eksempel en pil til visning av retning. Noe som kan være nyttig ved videre utvikling.

Lagring i flashminnet kan skrus på når GPS-mottakeren enda ikke har utregnet en gyldig posisjon, men ingenting blir lagret i flashminnet før en gyldig posisjon har blitt utregnet. Hvis flashminnet er fullt, må flashminnet slettes for å kunne lagre nye posisjoner.

9.5 Bilder av GPS-mottakeren



Figur 9.6; Bilde av GPS-mottaker, forside.



Figur 9.7; Bilde av GPS-mottaker, bakside.

10 Test av den ferdige GPS-mottakeren

Ferdig produsert kretskort ble mottatt den 22.11.05, som var noe senere enn først planlagt. Dette ført til at det ikke ble like mye tid til testing av den ferdige GPS-mottakeren som det var ønsket tidlig i prosjektfasen. Det var ønskelig at hele november kunne brukes til utesting av den ferdige GPS-mottakeren. Testingen er delt opp som de etterfølgende avsnitt viser.

10.1 Elektrisk test før tilkobling av drivspenning

Det første som ble gjort etter at kortet var mottatt, var en visuell kontroll av kortet samt en elektrisk test med et multimeter for å lokalisere eventuelle feil som for eksempel kortslutninger.

Det ble oppdaget loddetinn mellom pinne 63 og 64 på mikrokontrolleren, det vil si full kortslutning mellom drivspenning (+3V3) og jord (GND). Dette ble fjernet.

Kontakten for tilkobling til display var montert speilvendt. Dette ble løst ved å modifisering av flatkabelen mellom display og kontakten for tilkobling av display på kretskortet.

10.2 Elektrisk test med tilkoblet drivspenning

Det ble brukt en ekstern strømforsyning (Mascot type 719) istedenfor batteridrift under testingen. Et multimeter ble benyttet for å sjekke spenningsnivåer på forskjellige steder i kretsen. Alle spenningsnivåer var tilfredsstillende med unntak av VCC-USB på kretsskjemaet, denne ble målt til 2,18V uten at USB-kontakten var tilkoblet en PC, det skulle ikke vært noe spenning her når ikke USB-kontakten var tilkoblet. Senere viste det seg at denne ”feilspenningen” ikke resulterer i noen funksjonell feil, annet enn at lysdiode D2 på kretsskjemaet lyser konstant (denne skal kun lyse når USB-kontakten tilkobles en PC). Kilden til denne feilspenningen er sannsynligvis ladekretsen BQ24013 (U8), men det kommer ikke tydelig frem av databladet til BQ24013 at den har et spenningsnivå på

inngangen (pinne 1 og 2 på BQ24013), når den har en spenningskilde tilkoblet på utgangen (pinne 9 og 10 på BQ24013). En annen kilde til feilspenningen kan være USB-konverteringskretsen FT232BM (U4). På grunn av liten tid og at systemet faktisk fungerer ble det ikke brukt mye tid på feilsøking relatert til dette problemet.

10.3 Programmering av mikrokontrolleren

Overføring av programmet fra PC til mikrokontroller gjøres, som tidligere nevnt, med programmet AVRstudio. Under første overføring av programvaren ble det ikke oppnådd kontakt mellom AVRstudio og mikrokontrolleren. Etter diverse informasjonssøk på internett, viste det seg at ISP kontakten på den ferdig GPS-mottakeren var tilkoblet feil pinner på mikrokontrolleren. Den riktige oppkobling er:

- Pinne 1 (MISO) på ISP-kontakten skal tilkobles pinne 3 på mikrokontrolleren.
- Pinne 4 (MOSI) på ISP-kontakten skal tilkobles pinne 2 på mikrokontrolleren.

Disse pinnene på mikrokontrolleren er tilkoblet GPS-modulen (uPatch100), slik at ved å bryte forbindelsen mellom uPatch100 og mikrokontrolleren, og benytte disse pinnene under programmering av mikrokontrolleren løste problemet seg. Under vanlig drift av kretsen kobles så disse tilbake til uPatch100.

Etter denne modifikasjonen fungerte overføringen av programvaren som ønsket.

En fremtidig løsning på dette kan være å legge inn en "bootloader" på mikrokontrolleren, dette gir mulighet for å overføre programvaren fra PC til mikrokontrolleren via UART-grensesnittet. Da slipper man ISP-kontakten, og siden det ene UART grensesnittet på mikrokontrolleren er tilkoblet FT232BM (UART til USB konvertering), kan man da programmere mikrokontrolleren via USB.

10.4 USB-kommunikasjon

Lederne i USB-kontakten (P1) var ikke loddet på kretskortet. Disse ble loddet på, og i tillegg ble også pinne 4 og 5 i kontakten loddet sammen. Etter disse modifikasjonene fungerte kommunikasjonen mellom mikrokontroller og PC via USB bra.

10.5 Lading av batteri

Når kretsen er tilkoblet USB grensesnittet så skal batteriet kunne lades opp ved hjelp av ladekretsen BQ24013. Lading kan skrues av og på ved hjelp av CE ("Chip Enable", pinne 7 på BQ24013). Statusen på lading skal kunne leses av på STAT1 og STAT2, se avsnitt 9.2.3.

Det ble foretatt en testopplading av batteriet. Under denne testen ble ladestrømmen overvåket. Under starten av ladesyklusen ble ladestrømmen målt til 166mA, etter cirka 3 timer var ladestrømmen sunket til 4 mA, i følge datablad for batteriet [13], er batteriet full-ladet når ladestrømmen har sunket til under 10mA.

Styring av ladingen ved hjelp av CE fungerte, men det var i midlertidig ikke mulig å få noe informasjon fra STAT1 og STAT2, verken på mikrokontrolleren eller ved bruk av oscilloskop.

Funksjonen for å forandre ladestrømmen ut av BQ24013, oppkoblingen mellom Iset (pinne 6 på BQ24013) og PORTG1 (pinne 34 på mikrokontrolleren), fungerte tilsynelatende ikke, ladestrømmen var den samme hele tiden.

På grunn av liten tid ble det ikke gjort noen større undersøkelser/feilsøking på hvorfor STAT1, STAT2 og kontrollering av ladestrømmen ikke fungerte som ønsket, dette er heller ikke funksjoner som er essensielle for prosjektet, man får uansett ladet opp batteriet.

10.6 Enkel test av programmet

Etter at mikrokontroller var programmert ble det gjort en enkel test av programvaren for å verifisere at slik ting som GPS-modul (uPatch100), lagring av posisjon i flashminnet, sletting av flashminnet, visning av korrekte data på display, knapper (SW1, SW2, SW3 og SW4) og overføring av data til PC fungerte tilfredsstillende.

Denne testen ble utført på samme geografiske sted som utvikling av programvaren (rom C-551, elektrobygget, NTNU avdeling Gløshaugen), på bakgrunn av at dette ga en viss indikasjon av ytelsen på uPatch100 sammenlignet GPS-mottakeren som ble benyttet under utvikling av programvaren (Trimble Ensign XL).

Under denne testen kom følgende momenter frem:

- Knappene var i første omgang uhyre sensitive for berøring, dette skyldtes at de interne ”pull-up-motstandene” på de tilhørende inngangene ikke var skrudd på i mikrokontrolleren. Etter at dette var rettet på fungerte knappene helt fint.
- uPatch100 fungerte bedre med tanke på posisjonsinnhenting enn GPS-mottakeren som ble brukt under utvikling av programvaren (Trimble Ensign XL).
- Under overføring av posisjonsdata (”tracking-punkt”) fra mikrokontroller til PC ble det avslørt en logisk feil i programvaren. Mikrokontrolleren stoppet aldri overføringen til PCen. Den feilen ble rettet på.
- Bakgrunnsbelysning på display lyser svakt når den er skrudd av, når bakgrunnsbelysningen skrues på, lyser det sterkere. Dette skjer høyst sannsynligvis på bakgrunn av valget av transistor (Q1). Retting av denne feilen ble ikke prioritert på grunn av liten tid.

Selv om denne enkle testen ga en indikasjon på at uPatch100 fungerte bedre enn Trimble Ensign XL i vanskelige omgivelser (innendørs), så er det ikke grunnlag for å konkludere med at den ene er bedre enn den andre. Det må mer utfyllende tester til for å kunne komme med en konklusjon.

10.7 Strømforbruk

Som beskrevet tidligere i rapporten er lavt strømforbruk en viktig faktor. Derfor måles det totale strømforbruk for kretsen under forskjellige operasjonsmodus.

Operasjonsmodus	Totalt strømforbruk [mA]
Kun visning av informasjon på displayet, uPatch100 har ingen gyldig utregnet posisjon. Ingen bruk av flashminnet.	103
Kun visning av informasjon på displayet, uPatch100 har gyldig utregnet posisjon. Ingen bruk av flashminnet.	103
Sletting av flashminnet. uPatch100 har gyldig utregnet posisjon.	119
Lagring av posisjon i flashminnet. uPatch100 har gyldig utregnet posisjon.	110

Tabell 10.1; Totalt strømtrekk.

10.7.1 Kommentarer til måling av strømforbruk

Det ble brukt et vanlig multimeter som amperemeter. Bakgrunnsbelysning er "av" ved samtlige operasjonsmodus. Strømtrekket varierte hele tiden, verdiene i tabell 10.1 er de høyeste verdiene ved forskjellige modus. De minste verdiene ligger på ca. 78 mA for de fleste operasjonsmodus. Disse variasjonene skyldes antakelig GPS-mottakeren.

Strømtrekk ved overføring til PC via USB er ikke tatt med på grunn av at kretsen da får strøm fra USB-bussen. Den komponenten som trekker desidert mest strøm er GPS-modulen (uPatch100), se for øvrig kapittel 8.1.2 for GPS-modulens strømtrekk. Ut fra datablad til uPatch100 skulle man tro at GPS-modulen trakk mindre strøm når en gyldig posisjon er utregnet, men det viser målingene ingen tendenser til, dette bør i midlertidig undersøkes nærmere i masteroppgaven.

Det var vanskelig å finne noen god gjennomsnittsverdi for strømtrekket, derfor er høyeste strømtrekk oppgitt i tabell 10.1.

På bakgrunn av observasjoner kan det antas et tidsgjennomsnittelig strømtrekk på ca. 100 mA ved normal bruk av GPS-mottakeren.

10.8 Utfyllende test av GPS-mottakeren

NTNU har en GPS simulator som kan simulere forskjellige posisjoner med forskjellige parametere til feilkilder som flerveistransmisjon, ionosfære og troposfære. Slik testing hadde vært interessant, men det ble det ikke tid til i prosjektoppgaven. En slik test er og mer rettet mot selve GPS-modulen (uPatch100) og dens evne til å regne ut gyldige posisjoner, og sier derfor ikke så mye om resten av kretsen som prosjektgruppen har utviklet.

Det ble utført en enkel praktisk test av GPS-mottakeren den 04.12.05. GPS-mottakeren ble tatt med på en fottur rundt om i Trondheim sentrum. Dette for å verifisere at GPS-mottakeren virker, og avsløre eventuelle praktiske feil eller mangler i programvaren.

Følgende momenter/feil kom frem:

1. Adressen til lagringsposisjon i flashminnet lagres i et flyktig minne på mikrokontrolleren, derfor settes denne adressen til 0 ved strømbrudd. Dette vil si at hvis man vil lagre nye "tracking-punkt" etter et strømbrudd, så lagrer man over gamle "tracking-punkt".
2. Selv om GPS-mottakeren er stasjonær, endres posisjonen (lengde- og breddegrad) noe, slik at disse lagres i flashminnet.

Kommentarer:

1. Denne feilen skyldtes logisk feil i programvaren. Adressen til lagringsposisjon i flashminnet lagres nå på mikrokontrollerens EEPROM (Electrical Erasable Programable Read-Only Memory) minne, som er et ikke-flyktig minne. Den sist brukte adressen lagres i EEPROM, og ved neste oppstart av mikrokontrolleren brukes denne adressen som startadresse for lagring i flashminnet.
2. Hvis posisjonsdataene ("tracking-punkt") er de samme over tid skal ikke disse lagres for å spare plass i flashminnet. Posisjonsresultatet vil alltid fluktuere på grunn av mange varierende årsaker, dette kan løses ved kun å sjekke på de mest signifikante verdiene i posisjonsdataene

Mer utfyllende testing vil bli utført i masteroppgaven.

11 Videre arbeid

Prosjektoppgaven er i hovedsak en forberedelse til masteroppgaven, hvor det skal lages et system som skal kunne operere under vanskelige forhold, det vil si når GPS-signalene til tider uteblir.

Systemet som er utviklet under prosjektoppgaven er ikke blitt testet så mye som ønsket, en del av masteroppgaven blir derfor å videreføre testingen, og disse testene skal sammenlignes med tilsvarende tester av utvidet/forbedret mottaker utviklet gjennom masteroppgaven.

Arbeid videre under masteroppgave er å utvide mottakeren med et treghetsnavigasjons system, eng.: Inertial Navigation System (INS). Hvilke komponenter INS-modulen skal bestå av bestemmes under masteroppgaven. Det som forløping er klart er at det blir bruk av MEMS-type (Micro Electronic Mechanical System) eller lignende komponenter. Gyroskop, akselerometere og elektronisk kompass er eksempler på aktuelle komponenter. I den forbindelse skal det undersøkes hensiktsmessige måter å kombinere GPS og INS data på.

Det skal ses på muligheter for å minimere strømforbruk, ettersom systemet er batteridrevet. Et eksempel er å kun navigere ved hjelp av INS og kjøre GPS-modulen i strømspare modus i perioder hvor dette er mulig.

Som beskrevet tidligere i rapporten er det avdekket enkelt feil i maskinvaren, derfor bør det vurderes om det er nødvendig å utvikle et nytt kretskort.

12 Resultat og konklusjon

I dette prosjektet er det utviklet en ren GPS-mottaker med en hensiktsmessig rutine for posisjonslagring og overføringsmulighet til PC. Hele systemet kan overvåkes og kontrolleres gjennom et enkelt brukergrensesnitt. Utviklingen bestod av valg av komponenter og design av maskinvare med tilhørende programvare.

Det har blitt utført en begrenset mengde med testing som har gitt en indikasjon på at systemet i hovedsak fungerer som det skal, men det er i midlertidig avdekket enkelte feil ved maskinvaren.

Prosjektoppgaven har gitt ett godt grunnlag for videre utvikling i masteroppgaven, i form av et system som fungerer som plattform for utvidelse, men også i form av tilegnet kunnskap om elektronikkutvikling.

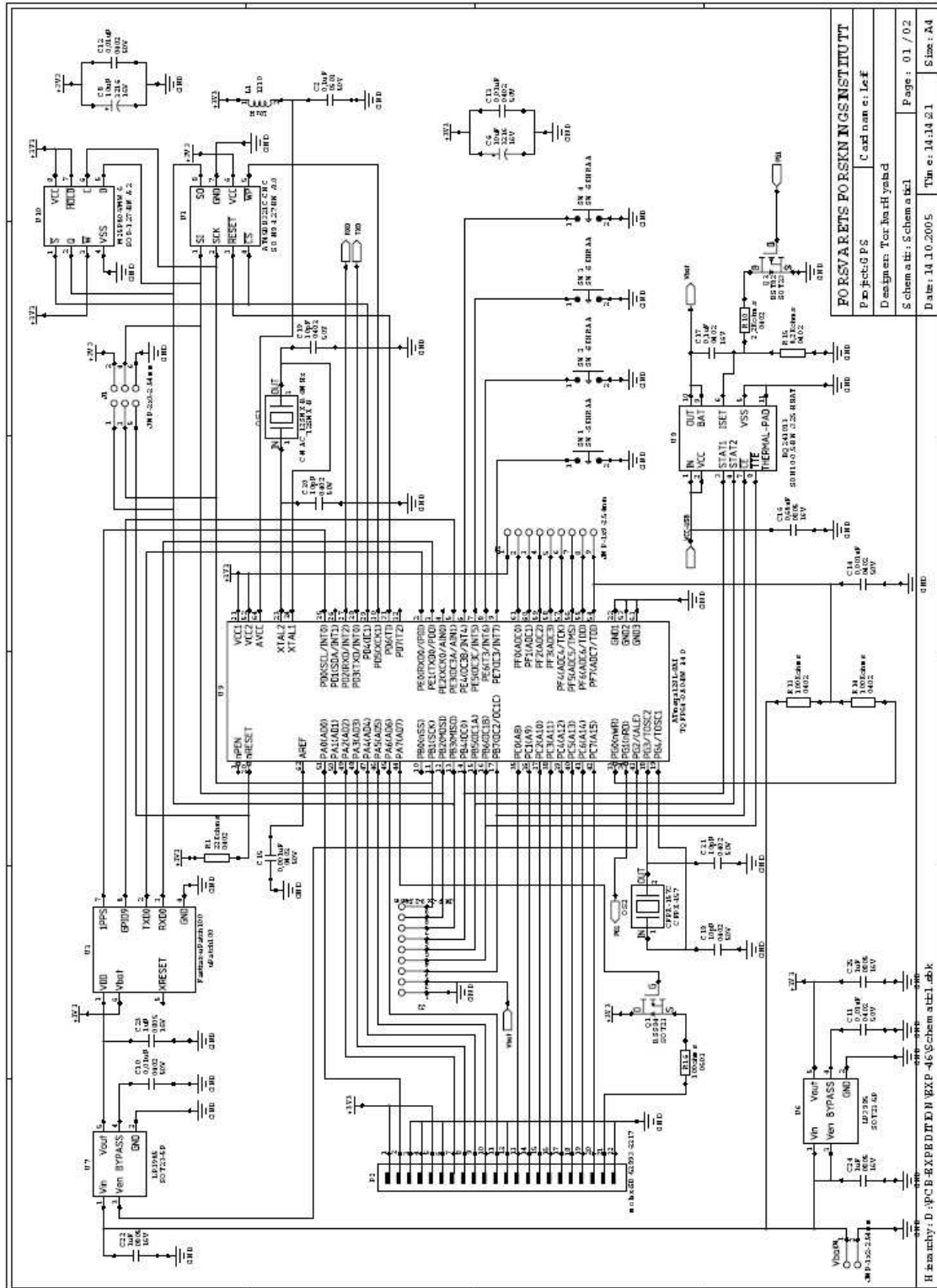
13 Kilder og referanser

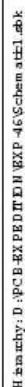
- [1] Datablad for AVR ATmega128 fra Atmel.
- [2] Datablad for M25P80 fra ST.
- [3] Datablad for FT232BM fra FTDI.
- [4] Datablad for uPatch100 fra Fastrax.
- [5] uPatch100 interface description.
- [6] Datablad for AT45DB321C fra Atmel.
- [7] Datablad for BQ24013 fra Texas Instruments.
- [8] Datablad for LP3985 fra National Semiconductor.
- [9] Datablad for BSS84.
- [10] Datablad for BST82.
- [11] Datablad for MG1060A-SGL fra Everbouquet.
- [12] Datablad for SED1560 (displaydriver).
- [13] Datablad for batteri.
- [14] Datablad for SAM-LS fra ublox.
- [15] Application notes fra FTDI.
- [16] Application note AVR306 fra Atmel.
- [17] Application note AVR151 fra Atmel.
- [18] Application note AVR134 fra Atmel.

- [19] GPS Basics av Jean-Marie Zogg.
- [20] Introduksjon til C, kompendie av Åge T. Johansen.
- [21] Radionavigation Systems av Börje Forssell.
- [22] Global Positioning Systems, Inertial Navigation, and Integration av Mohinder S. Grewal, Lawrence R. Weil og Angus P. Andrews.
- [23] Introduction to Random Signals and Applied Kalman Filtering, 3. utgave, av Robert Grover Brown og Patrick Y.C. Hwang.
- [24] GPS Modernization PowerPoint presentasjon av Börje Forssell.
- [25] The NMEA protocol av Klaus Betke.

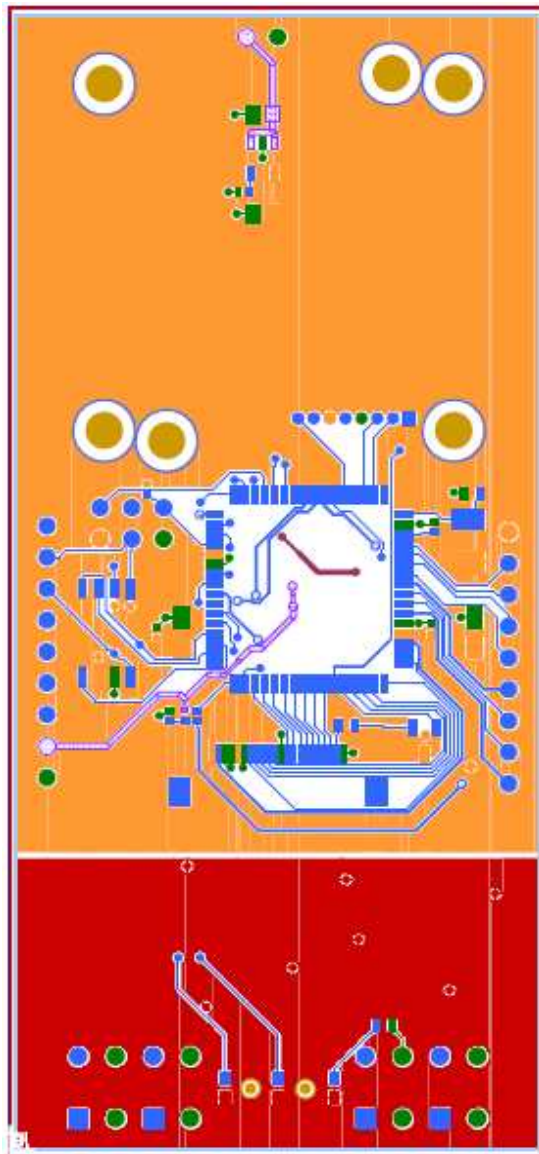
- [26] Hovedprosjekt (høgskole) av Ermin Lepenica, Anders Walmann, Ole Christian Mustorp og Tor Ivar Hystad.
- [27] Manualen til CodeVisionAVR.
- [28] <http://www.avrfreaks.com/>
- [29] <http://www.ftdichip.com/>
- [30] <http://www.wikipedia.org/>
- [31] <http://www.embedded.com/>
- [32] <http://gpsd.berlios.de/NMEA.txt>
- [33] <http://www.pocketgpsworld.com/ttfcomparisons.php>
- [34] <http://www.sechangsemicon.com/>
- [35] <http://tycho.usno.navy.mil/gpscurr.html>

Vedlegg A, kretsskjema

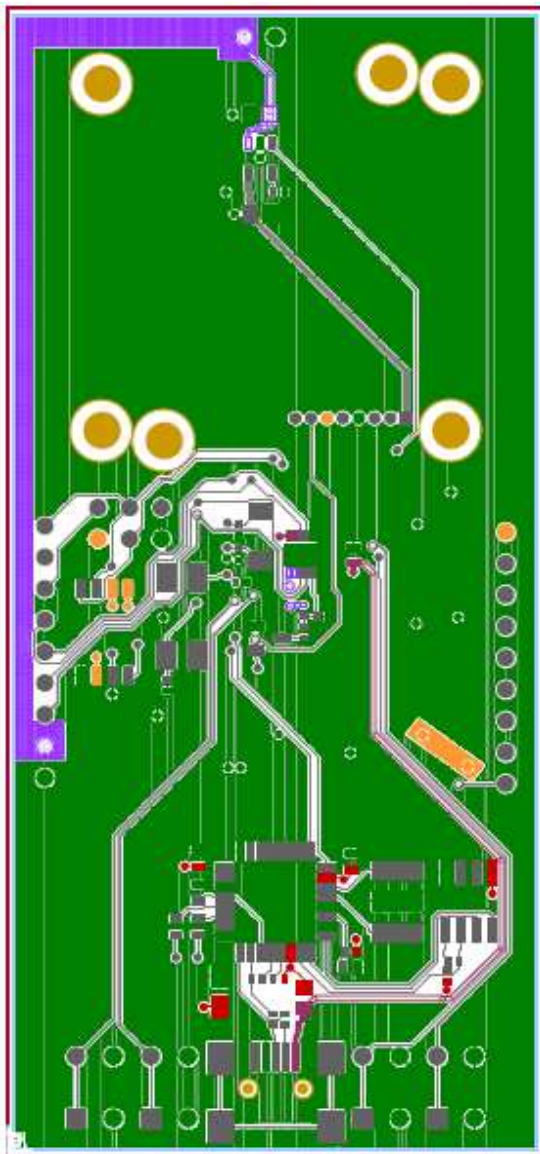




Produksjonsunderlag

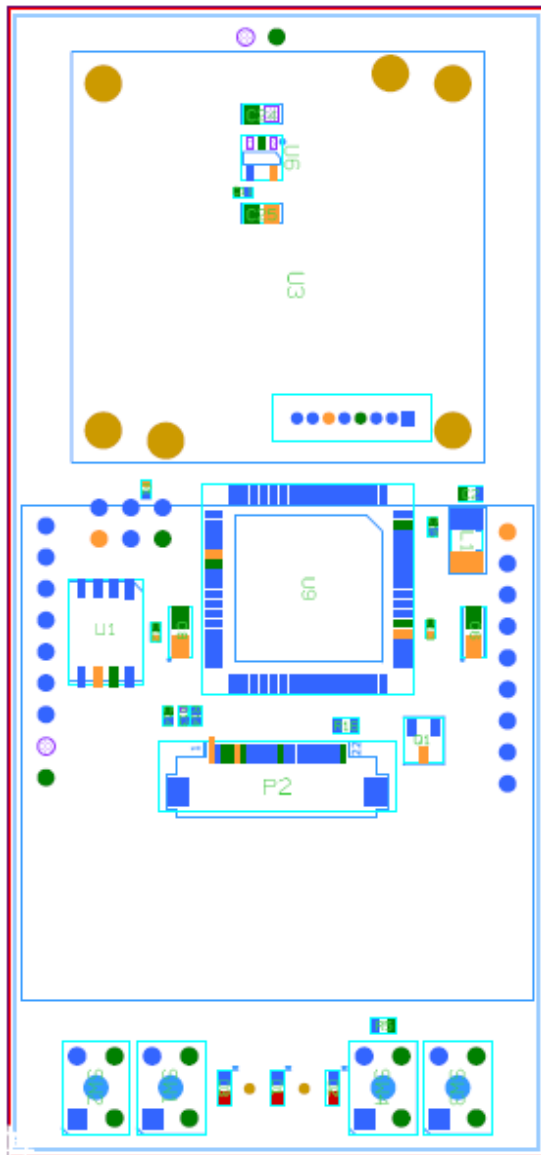


Overside av kretskort.

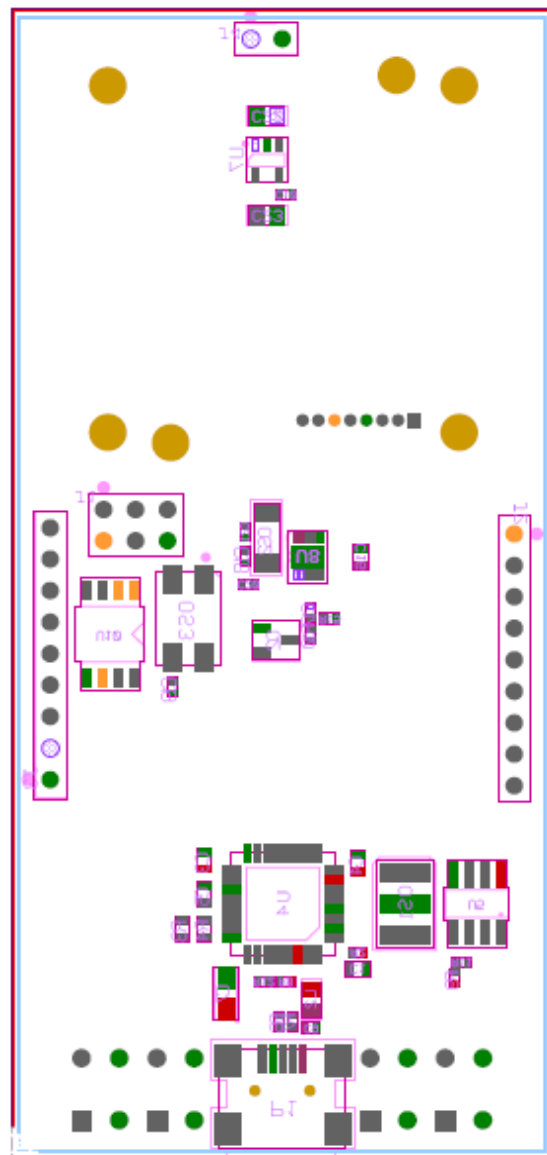


Underside av kretskort.

Komponentplassering



Komponentplassering,
overside av kretskort.



Komponentplassering,
underside av kretskort.

Komponentliste

Item	Qty	Reference Designator	Part Number	Description
1				
		U5	93LC46BSN	
	1			
2				
		U1	AT45DB321C-CNC	
	1			
3				
		U9	ATmega128L-8AI	
	1			
4				
		L2	BLM21P300S	
	1			
5				
		U8	BQ241013	
	1			
6				
		Q1	BSS84	
	1			
7				
		Q2	BST82	
	1			
8				
		C21	CAP-NPO-0402-10p-50V-5%	
		C18		
		C20		
		C19		
	1			
9				
		C8	CAP-T-3216-10u-16V-20%	
		C6		
		C7		
	3			
10				
		C9	CAP-X7R-0402-10n-50V-10%	
		C12		
		C13		
		C10		
		C11		
	5			
11				

Vedlegg A, kretsskjema

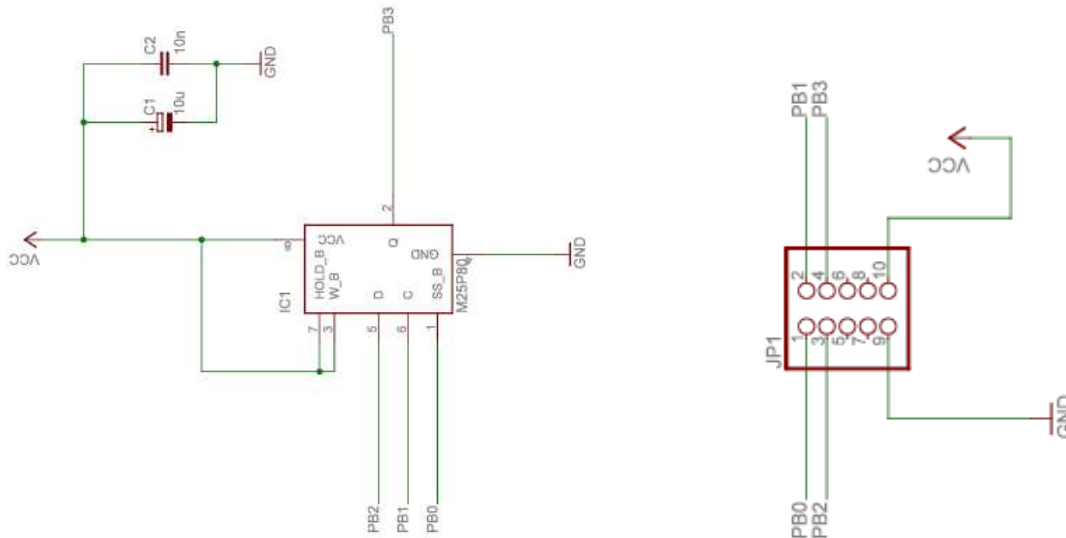
		C15	CAP-X7R-0402-1n-50V-10%	
		C14		
	2			
12				
		C2	CAP-X7R-0603-100n-50V-10%	
		C4		
		C1		
		C3		
	4			
13				
		C5	CAP-X7R-0603-33n-50V-10%	
	1			
14				
		C23	CAP-X7R-0805-1u-16V-10%	
		C22		
		C25		
		C24		
	4			
15				
		C16	CAP-X7R-0805-680n-16V-10%	
	1			
16				
		C17	CAP-Y5V-0402-0.1u-16V	
	1			
17				
		OS2	CFPX-157C	
	1			
18				
		OS3	C-MAC_12SMX-B-8MHz	
	1			
19				
		OS1	CSTCC6.00MG-TC	
	1			
20				
		U3	Fastrax-uPatch100	
	1			
21				
		U4	FT232BM	
	1			
22				
		L1	Hitano-SCI1210FT100	
	1			
23				
		J4	JMP-1x2-2.54mm	

	1			
24				
		J3	JMP-1x9-2.54mm	
		J2		
	2			
25				
		J1	JMP-2x3-2.54mm	
	1			
26				
		D2	LED-0603-GREEN	DIALIGHT LYSDIODE
		D1		
	2			
27				
		D3	LED-0603-RED	DIALIGHT LYSDIODE
	1			
28				
		U7	LP3985	NATIONAL
		U6		
	2			
29				
		U10	M25P80-VMW6	
	1			
30				
		P1	Mini-USB	
	1			
31				
		P2	molex-SD-52893-2217	
	1			
32				
		R8	RES-0402-0-1/16W-5%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
33				
		R12	RES-0402-1.5K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
34				
		R14	RES-0402-100K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
		R13		
	2			
35				
		R9	RES-0402-10K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
36				
		R10	RES-0402-2.2K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
		R11		

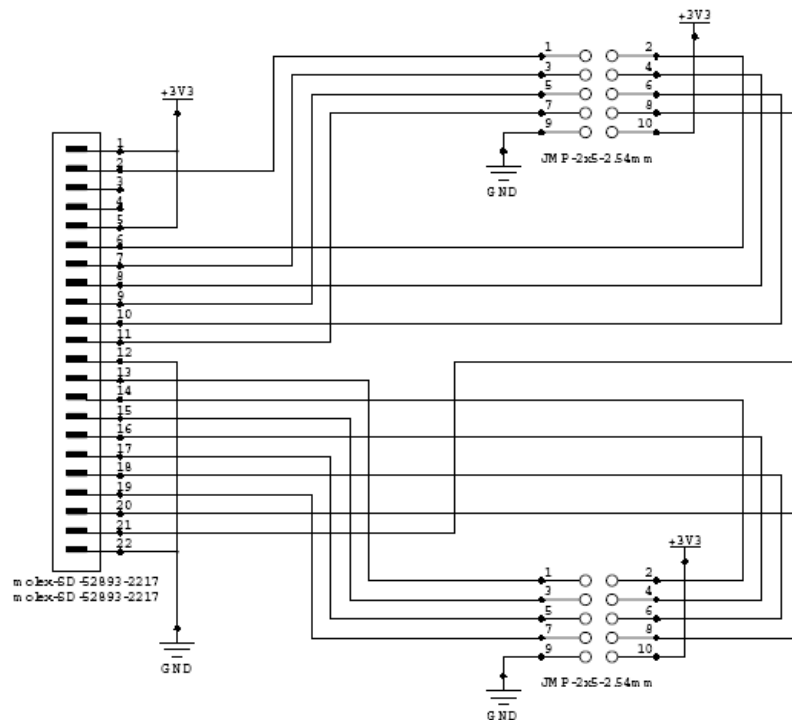
Vedlegg A, kretsskjema

	2			
37				
		R1	RES-0402-22K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
38				
		R7	RES-0402-27-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
		R6		
	2			
39				
		R2	RES-0402-470-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
40				
		R15	RES-0402-8.2K-1/16W-1%	Resistor, 1/16W, 0402, Thick Film, Low noise.
	1			
41				
		R16	RES-0603-100-1/10W-1%	Resistor, 1/10W, 0603, Thick Film. Low noise.
	1			
42				
		R3	RES-0603-300-1/10W-1%	Resistor, 1/10W, 0603, Thick Film. Low noise.
		R4		
		R5		
	3			
43				
		SW3	SW-SKHRAA	
		SW4		
		SW2		
		SW1		
	4			

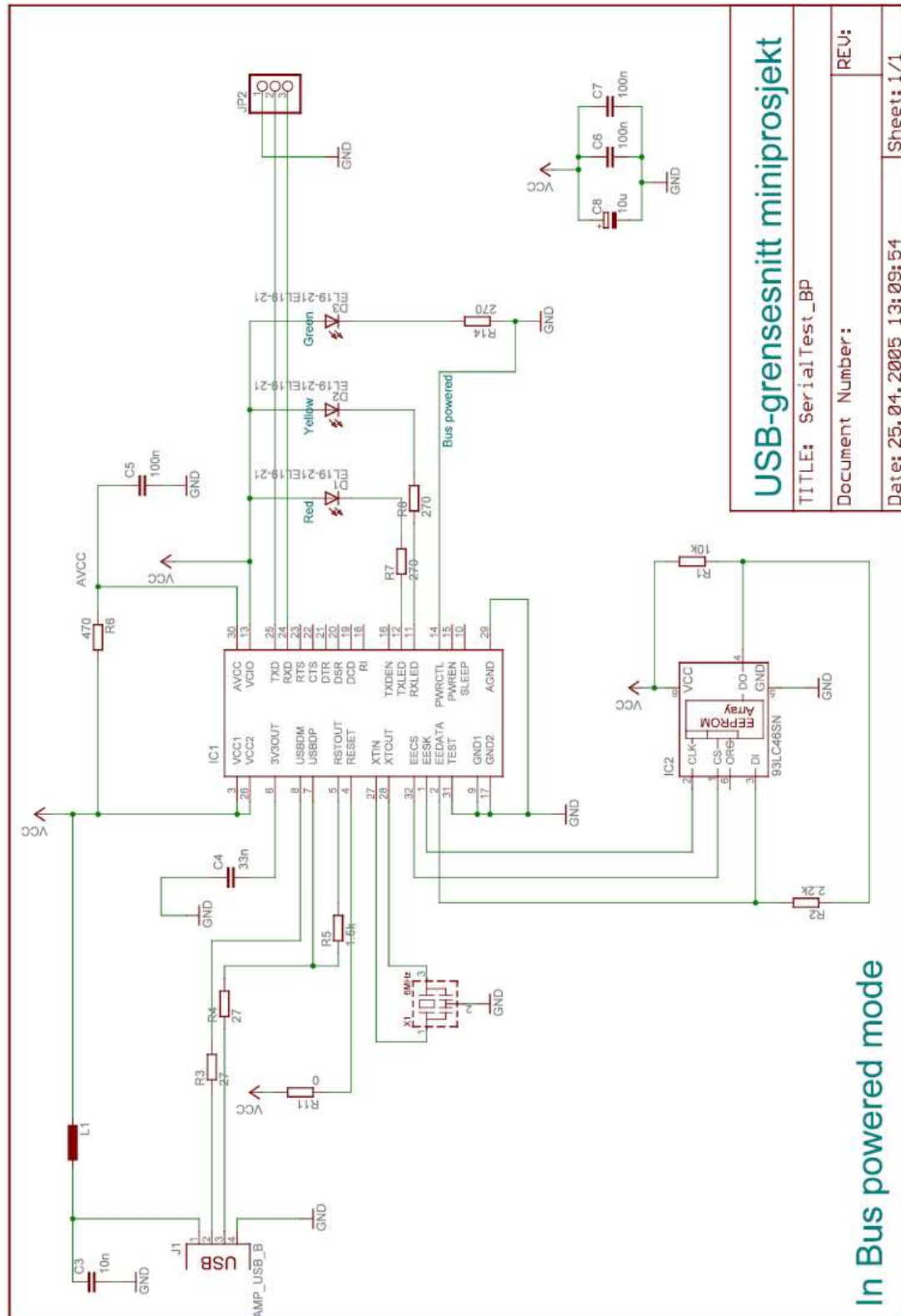
Vedlegg B, kretsskjema for utviklingskretser



Kretsskjema for test av flashminne M25P80



Kretsskjema for overgang mellom display og simuleringskit STK500/STK501.



Kretsskjema for test av USB konverteringskrets FTDI FT232BM.