

Foranalyse
for
Lavfrekvensstyring af Aeroquad

10893, Rasmus Bækgaard

08830, Rasmus Berg Kloster

18. december, 2013

Versionshistorie

1.0 | 05/6 -13 | RR | Første version af kravspecifikationen afleveret til Torben

Godkendelsesformular

Forfatter(e):	10893, Rasmus Bækgaard og 08830, Rasmus Berg Kloster
Godkendes af:	Torben Gregersen
Projektnummer:	13056
Antal side:	15
Kunde:	Ingeniørhøjskolen i Aarhus

Sted og dato:

Torben Gregersen

Rasmus Bækgaard

Rasmus B. Kloster

Indhold

1	Indledning	5
1.1	Formål	5
1.2	Valg af trådløs kommunikation	5
1.2.1	Regler og valg af Frekvens	5
1.2.2	Valg af sender	6
1.2.3	Valg af Microcontroller til fjernbetjening	7
2	Indledning	8
2.1	Formål	8
3	Hardware	9
3.1	Drone	9
3.2	Trådløs teknologi	9
3.3	Controller	9
3.4	Batteri	9
3.5	Sonar	9
4	Software	10
4.1	Tekstbehandlingsprogram	10
4.2	Versionskontrol	10
4.3	Valg af editor	11
4.4	Filopdeling	11
4.5	Videreprogrammering af Aeroquad's kode	11
4.6	Styring af dronen	12
4.7	•	12
5	Diverse	13

5.1	Love og regler	13
5.2	Sikkerhed	13
6	Konklusion	14
	Litteratur	15

Rettelser

Note: hvad hedder disse? 12

Kapitel 1

Indledning

1.1 Formål

1.2 Valg af trådløs kommunikation

Der er flere krav til den trådløse sender;

- Regler
- Frekvens
- Data bus
- Hastighed

1.2.1 Regler og valg af Frekvens

Der er mange regler inden for trådløs kommunikation, så hvilke frekvenser er lovlige at udnytte uden licens?

Det er de fleste i ISM-båndet, da de er lavet til Industriel, videnskabelig og medicinsk bånd. Grunden til det ikke er alle er at der er lokale krav, alt efter hvor man er i verden. Eks har USA meget komplekse krav til 433MHz. [1, s. 32]

De meste gængse frekvenser der er kan benyttes i industrien og til hobby brug er derfor:

- 433 MHz
- 868 MHz
- 915 MHz

- 2.4 GHz
- 5.8 GHz

Så der skal vælges en frekvens af disse. Den frekvens der egner sig bedst dette projekt, er den frekvens der rækker længst.

Factor	433 MHz	868 MHz	2.4 GHz
	Attenuation	Attenuation	Attenuation
Open office	0 dB	0 dB	0 dB
Window	< 1 dB	1 – 2 dB	3 dB
Thin wall (plaster)	3 dB	3 – 4 dB	5 – 8 dB
Medium wall (wood)	4 – 6 dB	5 – 8 dB	10 – 12 dB
Thick wall (concrete)	5 – 8 dB	9 – 11 dB	15 – 20 dB
Armoured wall (reinforced concrete)	10 – 12 dB	12 – 15 dB	20 – 25 dB
Floor or ceiling	5 – 8 dB	9 – 11 dB	15 – 20 dB
Armoured floor or ceiling	10 – 12 dB	12 – 15 dB	20 – 25 dB
Rain and/or Fog	20 – 25 dB	25 – 30 dB	?? *

Figur 1.1: Frekvens dæmpning - Figuren viser tests udfør af Telit wireless solutions et stort italiensk firma med speciale i trådløs teknologi.

Som man det ses på figures over så er der mindre modstand i objekter, desto længere man kommer ned i frekvens. I og med at næsten alle hobby entusiaster inden for RC fly, bruger 2.4 GHz. Bliver valget for dette projekt 433 MHz.

1.2.2 Valg af sender

Når det var valgt at projektet skulle benytte 433MHz. Skulle der findes en sender/modtager. Valget faldt på en chip fra Texas Instruments ccl101, Grundet dens mange muligheder. Nogle af de ting som projektet kunne drage fordel af var;

- Flere kanaler
- Variable datahastighed (0.6 to 600 kbps)
- Gode pakkehåndterings muligheder
- Flere frekvenser, hvor 433MHz er en af dem
- Programmerbar power output
- Spi interface

Denne chip var også nemt tilgængelig på diverse online sider som ebay. Hvilket betyder at den chip bliver brugt i vid udstrækning og dermed er prisen fornuftig.

1.2.3 Valg af Microcontroller til fjernbetjening

Det var allerede forudbestemt at denne skulle benytte en atmel AVR chip, da der var kendskab til denne type chips.

Det der skulle bestemmes var hvilken Chip i atmels serie.

- Spi
- I2C
- external interrupts pins

Det viste sig at de mindste chips som levede op til disse krav er;

- ATtiny88 [2]
- atmega8 [3]
- atmega16 [4]
- atmega32 [5]

Disse chips opfyldte kravene til chippen der skulle bruges. Så der skulle ses på andre faktorer, for at vælge en. Det blev pris, der afgjorde dette.

ATtiny88 - 46,91 kr.

Atmega8 - 7,71 kr.

Atmega16 - 21,40 kr.

Atmega32 - 36,29 kr.

(prisen - I skrivende stund, på ebay, for dip versionen, inc. porto)

Derfor blev valget en atmega8.

Kapitel 2

Indledning

2.1 Formål

Kapitel 3

Hardware

3.1 Drone

3.2 Trådløs teknologi

3.3 Controller

3.4 Batteri

3.5 Sonar

Kapitel 4

Software

Dette afsnit beskriver hvilke overvejelser der er gjort mht. software. Det indbefatter hvordan kommunikationen mellem fjernbetjeningen skulle være, hvordan dronens styring blev overtaget uden en egentlig driver, hvordan arkitekturen skulle bygges op for optimalt design og de tidsmæssige krav kunne overholdes, samt hvad der er muligt at implementere foruden den nuværende.

4.1 Tekstbehandlingsprogram

Som tekstbehandler lå Microsoft Word som en nem faktor, men da det ene medlem af gruppen ikke havde Microsofts Office-pakken og grundet Word's meget dårlige håndtering af billeder, tabeller og skriftformatering faldt valget på open source programmet LaTeX.

I modsætning til Word har LaTeX et fokus på, at brugeren af programmet skriver noget med indhold og bekymrer sig om udseendet senere. Med den meget simple måde at skrive LaTeX-dokumenter på kommer også, at det er meget nemt at flette flere dokumenter sammen til ét og det med versionskontrol er meget nemt at merge konflikter.

4.2 Versionskontrol

For at sikre at filer ikke forsvandt i udviklingsprocessen, skulle der vælges versionskontrol, der vil gøre det muligt at trække gamle filer frem.

Muligheden at bruge Dropbox som versionskontrol blev udelukket, da tidligere versioner kun gemmes i 30 dage¹ og 2 personer arbejdende i samme projekt medfører uhensigtsmæssige overskrivninger.

Versionskontrol fra Subversion blev ligeledes udelukket, da det tager længere tid at løse konflik-

¹<https://www.dropbox.com/help/113/en>

ter og holde styr på gamle version, for slet ikke at nævne mængden af plads der skal bruges. Som filstyring blev Github² valgt – et værktøj der er designet til versionsstyring for programmører. RAB har i et år brugt dette og Github tillader nemt og kontrolleret, at smelte dokumenter sammen.

4.3 Valg af editor

Dronens software er skrevet i open source kode, da dette giver mulighed for programmering i Linux og Windows³. Til dette kan Arduino's standard editor bruges, men denne har ingen farver på keywords, hvilket gør koden meget svær at læse. Dette kan omgås ved at bruge en ekstern editor, så som Notepad++ eller Sublime2, som giver farve, men som kræver at du skifter frem og tilbage imellem vinduerne for at compile.

For helt at løse dette bruges programudvidelsen Visual Micro⁴ til Microsoft Visual Studio 2012, hvilket giver mulighed for både at se projektets opbygning og compile koden direkte i vinduet til dronen, samt se output fra dronen.

4.4 Filopdeling

Dronens software er kun skrevet som header-filer og er opdelt således, at hver klasse har sin egen undermappe til projektmappen. For at det virker skal mapperne have det samme navn som filerne og de skal have en bestemt placering⁵.

Da alle filer styres igennem Github skal de ligge i samme filtræ, hvilket ikke lader sig gøre med Arduino's standardopbygning.

For at komme rundt om dette blev mappen med filerne forsøgt linket til Arduino's foretrukne placering. Desværre virker dette ikke og alle filer havnede derfor ved siden af *.ino*-filen, projektets hovedfil, der må ligge frit.

4.5 Videreprogrammering af Aeroquad's kode

Open source projekter har som regel deres egne kodestandarder og den følges strikt. Desværre har hovedudvikleren af systemet valgt kun at arbejde i header-filer og ingen source-filer, hvilket

²<https://github.com/>

³<http://aeroquad.com/showwiki.php?title=AeroQuad+Software>

⁴<http://www.visualmicro.com/>

⁵C:/Users/<brugernavn>/Documents/Arduino/libraries/

er anderledes end hvad IHA træner sine studerende i.

Valget mellem pæn og overskuelig kode, frem for fungerende og grim kode, er derfor ikke nemt, da koden som udgangspunkt ikke danner baggrund for ens karakter og omskrivning, så koden ville virke, ville være for stor.

Som udgangspunkt blev koden derfor udvidet i header-filer, men det endte hurtigt i flere erklæringer af variabler der skulle deles imellem filerne. For at overvinde dette blev koden delt ud i *.h*-filer og *.cpp*-filer og selvom størstedelen af koden er skrevet i C kan en C-compiler ikke bruge Arduino's Serial-port, hvilket filendelsen *.cpp* omgår.

4.6 Styring af dronen

For at styre dronen fra Aeroquad, bruges der normalt en fjernbetjening med 2 pinde⁶, der styrer dronens tiltning, hældning, rotation og opdrift. Eftersom vi ikke havde en sådan, var det nødvendigt at lave et modul som kunne overskrive det, således input fra et program ville blive registreret som en fjernbetjenings output.

Den første mulighed var at skrive signalet der blev modtaget af selve senderen til den respektive driver. Dette krævede dog, at vi kendte algoritmen det blev sendt igennem, hvilket er et bachelorprojekt i sig selv.

Den anden mulighed var at lade modtagermodulet undgå et analogt signal, men derimod læse hvad der ville komme ud af signalet. Ved at vælge den sidste løsning og oprette et tilsvarende array, som de originale data blev sendt til, kunne der overskrives med egne værdier, uden at ændre mange steder i den eksisterende kode. Herefter var det blot et spørgsmål om, at forstå hvad der blev sendt til dronen.

4.7 •

⁶FiXme Note: hvad hedder disse?

Kapitel 5

Diverse

5.1 Love og regler

5.2 Sikkerhed

Kapitel 6

Konklusion

Litteratur

- [1] K. L. Larsen, “Bestemmelser om luftfart med ubemandede luftfartøjer, som ikke vejer over 25 kg,” tech. rep., Statens Luftfartsvæsen, januar 2004. [http://selvbetjening.trafikstyrelsen.dk/civilluftfart/Dokumenter/Love%20og%20bestemmelser/Bestemmelser%20for%20Civil%20Luftfart%20\(BL\)/BL%2009-serien/9_4_ud3.pdf](http://selvbetjening.trafikstyrelsen.dk/civilluftfart/Dokumenter/Love%20og%20bestemmelser/Bestemmelser%20for%20Civil%20Luftfart%20(BL)/BL%2009-serien/9_4_ud3.pdf).
- [2] Atmel, “Tiny88.” <http://www.atmel.com/devices/attiny88.aspx?tab=parameters>, 25. november 2013.
- [3] Atmel, “Mega8.” <http://www.atmel.com/devices/attiny8.aspx?tab=parameters>, 25. november 2013.
- [4] Atmel, “Mega16.” <http://www.atmel.com/devices/ATMEGA16.aspx?tab=parameters>, 25. november 2013.
- [5] Atmel, “Mega32.” <http://www.atmel.com/devices/ATMEGA32.aspx?tab=parameters>, 25. november 2013.
- [6] © Arduino, “Download the arduino software.” <http://arduino.cc/en/Main/Software>, 22. november 2013.
- [7] E. Chatzikyriakidis, “QueueList library for arduino.” <http://playground.arduino.cc/Code/QueueList>, 13. oktober 2013.
- [8] AeroQuad, “Aeroquad software.” <http://aeroquad.com/showwiki.php?title=AeroQuad+Software>, 3. juli 2013.
- [9] Wikipedia, “Fifo.” <http://en.wikipedia.org/wiki/FIFO>, 22. november 2013.