

Foranalyse

AR.Drone control via DevKit8000



Titel: Foranalyse
Forfattere: Mads Havgaard Mikkelsen
Kasper Kirkeby Jacobsen
Vejleder: Torben Gregersen
Projektnummer: 12042
Institution: Aarhus Universitet Ingeniørhøjskolen
Sider: 8
Dato: 19-12-2012

Versionshistorie

Ver.	Dato	Initialer	Beskrivelse
1.0	06/09 2012	09636	
1.1	13/11 2012	09588	Tilføjelser af Autonavigation

Godkendelsesformular

Forfatter(e):	Kasper Kirkeby Jacobsen og Mads Mikkelsen
Godkendes af:	Torben Gregersen
Projektnummer:	12042
Dokument-id: (filnavn)	Foranalyse.pdf
Antal sider:	8
Kunde:	Aarhus Universitet Ingeniørhøjskolen

Sted og dato:

Torben Gregersen

Kasper Kirkeby Jacobsen (09588)

Mads Havgaard Mikkelsen (09636)

Indholdsfortegnelse

1	Introduktion	3
2	Hardware platform	3
3	Software platform.....	3
3.1	Problem.....	3
3.2	Løsningsmodel	3
4	WiFi	4
4.1	Problem.....	4
4.2	Løsningsmodel	4
5	GUI toolkit	5
5.1	Problem.....	5
5.2	Løsningsmodel	5
6	Joystick.....	5
6.1	Problem.....	5
6.2	Løsningsmodel	5
7	USB indgange	6
7.1	Problem.....	6
7.2	Løsningsmodel	6
8	Autonavigation.....	6
8.1	Problem.....	6
8.2	Løsningsmodel	6
9	AR.Drone	7
9.1	Kommunikation.....	7
9.2	Manøvrering og tilstande.....	7
9.3	Video	8

1 Introduktion

Formålet med foranalysen er at undersøge, om de problemstillinger der har været i projektoplægget har været mulige at løse, og hvordan vi har sikret os den bedste løsning.

2 Hardware platform

Hardware platformen er et DevKit8000 med tilhørende LCD touchskærm. DevKit8000 vælges som hardware platform, da det som krav fra skolen blev opstillet som platformen til projektet.

DevKit8000 er en single board computer, og benytter sig af en processor med ARM arkitektur.

Ydermere benyttes AR.Dronen som quadcopter drone og hardware platform til at blive styret fra DevKit8000.

3 Software platform

3.1 Problem

Der skal findes en kernel og operativ system der er kompatibel og kan kompileres til processorer med ARM arkitektur, og som understøtter vores behov for:

- WiFi USB kompatibilitet
- Grafisk brugerflade
- Joystick USB kompatibel
- LCD touch skærm kompatibel

3.2 Løsningsmodel

De kerner og operativ systemer der blev overvejet (i deres respektive embeddede udgaver):

1. Linux kernel

- a. Angström
- b. Ubuntu
- c. ArchLinux
- d. Android

2. Windows kernel

- a. Windows Embedded Compact 7

Det blev bestemt at kernen skulle være en Linux kerne, da Windows koster penge, og at det ikke kommer som Open-source og vi derved ikke selv har muligheden for at kompilere det med de tilvalg der ønskes.

Der findes mange Linux distributioner til embeddede processorer¹, vi valgte kun at fokusere på 4 af disse distributioner for at gøre processen nemmere. Disse 4 Linux distributioner blev valgt på grundlag af popularitet, vores kendskab til dem og mængden af support og hjælp der er at finde på internettet. Android vælges fra, da vi ikke er interesseret i at skulle udvikle en Android applikation og miste en del af Linux fleksibilitet som Android fratager en, samt vi ingen erfaring har med at porte Android til embeddede enheder.

Tilbage står vi med 3 almindelige Linux distributioner, Ubuntu, Angström og ArchLinux.

Af disse 3 Linux distributioner vælges Angström, og det er udelukkende pga. erfaring med distributionen samt dens nemme online distribution builder², der giver en den fordel at man kan bygge et standard image op med kun de pakker man ønsker sit styresystem at have.

4 WiFi

4.1 Problem

AR.Dronen er en quadrokopter hvis primære formål er at kunne bevæge sig frit omkring. Derfor skal der oprettes trådløs kommunikation mellem DevKit8000 og AR.Dronen over dens WiFi. DevKit8000 er et embedded board, der kommer uden integreret WiFi modul. Da der vælges at installere en Linux kerne og operativsystem på DevKit8000, er der en begrænset mængde WiFi USB enheder der er kompatible, da ikke alle fabrikanter har drivere der understøtter Linux kerner.

4.2 Løsningsmodel

Der findes lister online med hvilke WiFi chipsets der er kompatible med Linux. Vi har valgt at benytte os af denne liste³, for at finde en kompatibel WiFi enhed.

DevKit8000 giver mulighed for at købe udvidelsesmoduler bla. et WiFi modul. Dette modul var en mulighed at tilføje til DevKit8000, da det allerede er designet til DevKit8000. Dette WiFi modul kan købes på Embest hjemmeside⁴.

IHA havde en gammel WiFi USB enhed med Ralink rt73 chipset, der er kompatibel med Linux. Denne WiFi USB, D-Link DWL-G122, tages i brug i projektet. Alternativet hvis ikke skolen havde haft en kompatibel WiFi USB, var at købe en med et kompatibelt chipset.

¹ http://en.wikipedia.org/wiki/ARM_architecture#Operating_systems

² <http://narcissus.angstrom-distribution.org/>

³ <http://linuxwireless.org/en/users/Devices/USB>

⁴ <http://www.embedinfo.com/english/product/WF8000-U.asp>

5 GUI toolkit

5.1 Problem

Der skal vælges et graphic user interface toolkit der skal bruges til at designe programmet. GUI toolkittet skal vælges til at kunne virke til embeddede processorer og til Linux.

5.2 Løsningsmodel

1. FLTK

FLTK er et letvægt krydsplatform GUI toolkit der er baseret på C++. FLTK er afhængigt af X11 (X-Window server) for at kunne vise en grafisk brugerflade.

2. WxWidgets

WxWidgets er i den tungere ende af krydsplatform GUI toolkits, da det er et framework der har mange biblioteker m.m. til at krydsplatform udvikling på tvært af forskellige platforme (Mac, Windows, Linux m.m.)

3. GTK

GTK er et krydsplatform GUI toolkit der er afhængig af X11. GTK er et af de populære toolkits til at designe GUI's. GTK er udviklet i C, og understøtter mange forskellige script- og programmerings sprog.

4. Qt

Qt er sammen med GTK de mest populære GUI toolkits. Qt er et krydsplatform GUI toolkit, og et komplet framework. Qt er uafhængig af X11, og kan køre på dens egen vindue server. Qt tilbyder som del af dens framework en masse funktioner og libraries, der sørger for at det krydskompileres korrekt til kompileres platform.

Qt blev valgt som GUI toolkit. Qt blev valgt udelukkende baseret på tidligere erfaringer med at programmere i det, samt erfaringen med at portere det til ARM processor baserede systemer. Samt at Qt er meget enkelt at programmere i.

6 Joystick

6.1 Problem

At finde et joystick der er kompatibelt med Linux, og hvis protokol eller API vi kan kommunikere med, for at vi igennem programmet kan styre AR.Dronen med et joystick.

6.2 Løsningsmodel

Da vi havde en Xbox controller med tilhørende Xbox Wireless Receiver til rådighed, og den viste sig at være kompatibel med Linux blev denne valgt.

7 USB indgange

7.1 Problem

Både WiFi'et, Xbox Wireless Receiver og evt. andre USB tilslutninger (Keyboard m.m.), benytter USB interface til DevKit8000. DevKit8000 har en USB Type A indgang (USB 2.0) samt en USB Type Mini-B indgang. USB Type A bruges af både WiFi, Xbox Wireless m.m., og der skal derfor skaffes flere USB Type A indgange.

Derudover kan DevKit8000 ikke give nok strøm til at drive USB tilslutninger der trækker meget strøm.

7.2 Løsningsmodel

Da det pga. strømmangel ikke er muligt at tilslutte en alm. USB Type A hub med indgange, skal der benyttes en USB Type A hub med ekstern strømforsyning.

8 Autonavigation

8.1 Problem

AR.Dronen skal kunne følge en linje på gulvet under den, og skal flyve over linjen indtil linjen ikke længere er under den. AR.Dronen har kun en sensor der giver mulighed for at følge en linje på gulvet under den, og dette er kameraet i bunden af den. Kameraet giver en 15 fps video som kan bruges til at følge linjen på gulvet.

8.2 Løsningsmodel

På internettet kan man finde mange eksempler på robotter der følger linjer med forskellige metoder at detektere dem på. Vi ikke finde nogen eksempler på flyvende robotter der bruger et kamera til dette.

Der blev brugt lidt til på at finde den bedste måde at følge en linje på gulvet, men der var ingen metode vi fandt der bare kunne implementeres. Vi fandt dog et enkelt forum post hvor der er en der spørger om ideer til detektering af en linje vha. kamera.⁵ Ideen er at tage billedet fra et kamera der peger ned af, og inddele i et 3x3 grid og derudfra kunne styrer robotten til at køre frem dreje osv., alt efter hvilke af de 9 felter der er sorte, og holde feltet midt i robotens synsfelt sort.

Denne ide kan udvides til brug ved flyvning ved at gøre det grid linjen inddeles i større dvs. 5x5 eller 7x7 felter. Ud fra et sådant grid kan en algoritme beregne styringskommandoer til AR.Dronen. Det vil være smartest at have et grid med et ulige antal felter vandret på billedet da ideen er at holde linjen i midten af billedet.

⁵ <http://stackoverflow.com/questions/8463489/how-to-make-a-robot-follow-a-line-using-its-video-camera>

9 AR.Drone

Det første ved AR.Dronen der skal undersøges er om den har åbne protokoller da projektet afhænger af dette. Har AR.Dronen ikke åbne protokoller er det ikke muligt at udvikle en styring til den, da vi ikke ville kunne kende kommunikations protokollen til den.

Undersøgelsen viste at AR.Dronen har åbne protokoller, og de er beskrevet i [Bilag 1]

Eftersom AR.Dronen har åbne protokoller, skal der undersøges andre egenskaber, såsom hvilket interfaces den bruger til dens kommunikation (sockets, http eller lign.), hvilke flyveegenskaber den har, hvilke tilstande den kan befinde sig i, hvordan den streamer direkte video, m.m..

Al information til disse undersøgelser er tilgængelig i "ARDrone_SDK_1_7_Developer_Guide.pdf". I følgende afsnit vil vi komme ind på vores undersøgelser og resultater deraf.

9.1 Kommunikation

AR.Dronen benytter socket til kommunikation med dens klient. De sockets der oprettes imellem AR.Dronen og klienten, bruges til at sende styrings- og konfigurationsvariabler til AR.Dronen. Kommunikation fra AR.Dronen til klienten foregår på to forskellige porte, med henholdsvis video data og navigationsdata (flyve info m.m.).

AR.Dronen og klienten opretter forbindelse over en AD-HOC forbindelse hvor AR.Dronen har indbygget WiFi modul. AR.Dronens IP er 192.168.1.1, hvis ikke andet er konfigureret, og klienten får tildelt en IP i samme subnet som AR.Dronen mellem 2-5. Hvis AR.Dronens IP er 192.168.1.1, er klientens IP 192.168.1.2-5.

Alle AT-kommandoer sendes til AR.Dronen på port 5556, alt indkommende navigationsdata bliver sendt til klienten på port 5554, og alt video output fra AR.Dronen til klienten sendes på port 5555. Før AR.Dronen kender til klienten og dens IP, og kan broadcaste data, er det nødvendigt at klienten sender en vilkårlig besked på porten tilhørende video og navigationsdata.

9.2 Manøvrering og tilstande

AR.Dronen kan flyve X-Y-Z koordinater og rotere om sig selv. Alle disse flyveegenskaber og koordinater sendes over AT-kommandoer der bestemmer en retning og hastighed i X-Y-Z planet, eller en rotation om sig selv.

AR.Dronen har forskellige flyve tilstande der beskriver om den flyver, er lettet, er i nødstop osv..

AR.Dronen har 3 forskellige kommunikations tilstande, der beskriver den navigationsdata den sender på port 5554. Den første tilstand, som er initialiserings tilstanden kaldes 'bootstrap' tilstand, og i denne tilstand sendes der kun AR.Dronens tilstand. AR.Dronen kommer i denne tilstand hver gang den disconnecter dens WiFi forbindelse med en klient. Derudover kan AR.Dronen være i 'Debug' tilstand hvor den sender al information omkring AR.Dronen og dens tilstande på diverse hardware komponenter, software m.m.. Denne mængde navigationsdata er overflødig da der modtages mange informationer der ikke er nødvendige at modtage i dette projekt. Den sidste tilstand AR.Dronen kan

sende i, kaldes navigationsdata demo tilstand, hvor AR.Dronen kun sender de mest nødvendige information til klienten. Det drejer sig om bl.a. batteriniveau, højdemeter, vinkler og retninger AR.Dronen flyver. Navigationsdata demo tilstanden indeholder nok informationer til at tilfredsstille kravene for dette projekt.

9.3 Video

AR.Dronen kan sende dens video ud i to forskellige video kodeks, henholdsvis UVLC (VLIB) og P264 (H264). UVLC er et MJPG lignende kodeks hvor P264 er et H264 lignende kodeks. UVLC der er et kodeks lignende "Motion JPG" adskiller sig fra P264 ved at det ikke er komprimeret, men sender en serie af billeder til klienten. P264 derimod er komprimeret, og sender kun ændringen i billedet til klienten, og er derfor også et mere kompliceret kodeks, men mere effektivt da det ikke kræver så stor båndbredde og derfor egner sig bedst hvis AR.Dronen har dårligt signal til klienten.