

Test

AR.Drone control via DevKit8000



Titel: Test
Forfattere: Mads Havgaard Mikkelsen
Kasper Kirkeby Jacobsen
Vejleder: Torben Gregersen
Projektnummer: 12042
Institution Aarhus Universitet Ingeniørhøjskolen
Sider: 14
Dato: 19-12-2012

Versionshistorie

Ver.	Dato	Initialer	Beskrivelse
1.0	12/11 2012	09636	Enhedstest
1.1	26/11 2012	09636	Tilføjelse accepttest

Godkendelsesformular

Forfatter(e):	Kasper Kirkeby Jacobsen og Mads Mikkelsen
Godkendes af:	Torben Gregersen
Projektnummer:	12042
Dokument-id: (filnavn)	Test.pdf
Antal sider:	14
Kunde:	Aarhus Universitet Ingeniørhøjskolen

Sted og dato:

Torben Gregersen

Kasper Kirkeby Jacobsen (09588)

Mads Havgaard Mikkelsen (09636)

Indholdsfortegnelse

1	Test.....	3
1.1	Enhedstest	3
1.1.1	Xbox controller.....	3
1.1.2	Autonavigation.....	5
1.1.3	AT Commands	7
1.1.4	Navigationsdata	7
1.1.5	Styring vha. skærm input	7
1.1.6	DevKit8000.....	7
1.2	Integrationstest.....	10
1.3	Accepttest	11
1.3.1	Use Case 1.: System initialisering	11
1.3.2	Use Case 2.: Opdater video feed.....	11
1.3.3	Use Case 3.: Skift videokamera	11
1.3.4	Use Case 4.: Manøvrering – touch skærm	12
1.3.5	Use Case 5.: Opdater navigations data	12
1.3.6	Use Case 6.: Skift settings	13
1.3.7	Use Case 7.: Manøvrering – Xbox controller	13
1.3.8	Use Case 8.: Auto-navigation efter linje på gulvet.....	14

1 Test

Dette afsnit beskriver de test der er lavet i projektet. Der er lavet 3 former for test:

- Enhedstest
- Integrationstest
- Accepttest.

Enhedstest er test af forskellige enheder såsom klasser eller dele af programmet. En enhedstest er lavet for at finde ud af om den enkelte enhed har den funktionalitet, som er krævet af den.

Integrationstest er lavet for at teste om systemet virker som helhed, efter de enkelte enheder er blevet integreret i systemet.

Accepttesten er en test af det færdige projekt, for at se om det opfylder de use cases der er sat op som krav til systemets funktionalitet.

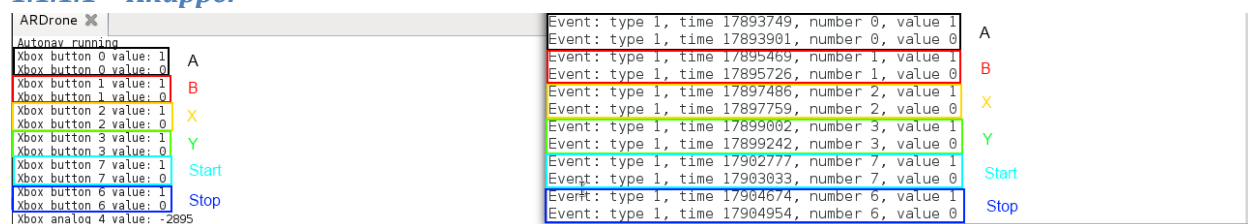
1.1 Enhedstest

1.1.1 Xbox kontroller

Xbox kontrollerens driver blev testet ved at afvikle Linux programmet "jstest", der viser outputtet fra kontrolleren. Xbox kontroller delen i programmet sammenlignes med hvad "jstest" gav som output.

De knapper og analogpinde der benyttes i projektet blev testet. Xbox kontrollerens knapper og analogspinde indeks nummer, samt deres output noteres. Da Xbox360 kontrolleren ikke kommer med nogen dokumenteret driver til Linux, var det nødvendigt at lave disse tests af deres indeks og værdi for at kunne genkende de forskellige knapper og analoge fra hinanden, samt for de analoges vedkommende at kende deres maksimum og minimum værdier.

1.1.1.1 Knapper



Figur 1 Xbox kontroller knap test. Venstre side viser output fra fjernstyringssoftwaren, og højreside viser output fra "jstest"

Figur 1 viser tests af inputs af Xbox kontrollerens knapper. Venstre del af Figur 1 viser debug-outputtet af programmet, og højre del viser outputtet af "jstest". Farverne på Figur 1 viser sammenhængen mellem debugger output og jstest, hvor teksten med tilhørende farve viser hvilken knap der blev trykket. Ud fra testen ses at indekseringen af Xbox kontrollerens knapper er som vist i Figur 2.

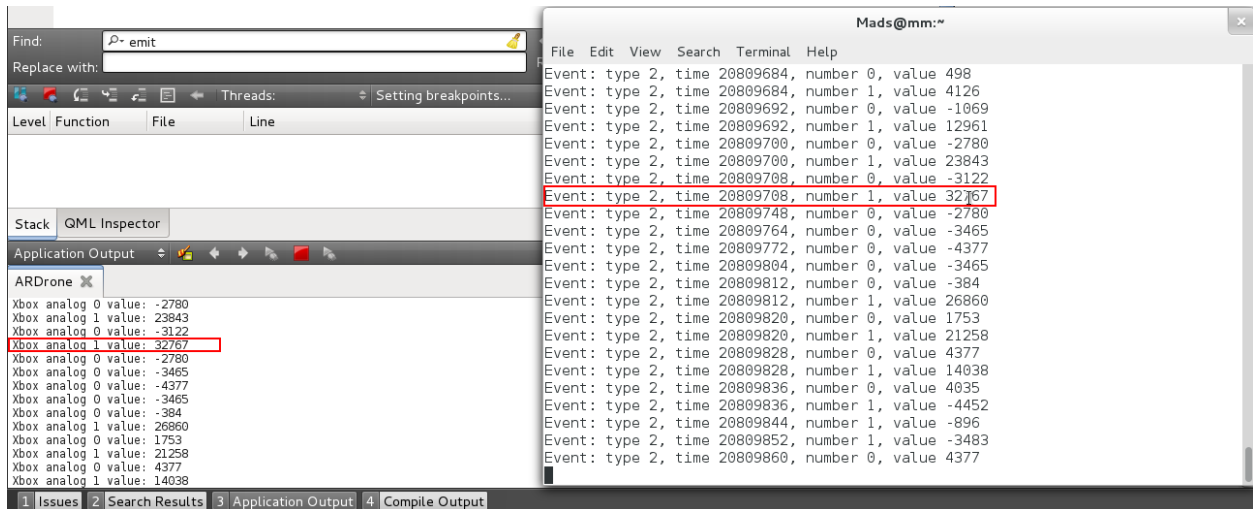
Xbox kontroller input	Indeksnummer	Deaktiv værdi	Aktiv værdi
A	0	0	1
B	1	0	1

X	2	0	1
Y	3	0	1
Start	7	0	1
Stop	6	0	1

Figur 2 Indeksering af Xbox kontroller knap input

1.1.1.2 Analoge pind

Der benyttes den højre og venstre analogpind på Xbox kontrolleren. De testes ved at bevæge dem ud i deres yder positioner for at læse deres maksimum og minimum værdier.



Figur 3 Output fra venstre analog pind

Figur 3 viser fjernstyringssoftwarens debugger og "jstest" output for den venstre analogpind på Xbox kontrolleren. Det røde markerede på Figur 3 viser den maksimale værdi en analogpind kan have. Ved at studere outputs fra "jstest" og fjernstyringssoftwarens debugger ses der at en analog pind har 2 outputs, henholdsvis for X og Y akse. Efter tests og evaluering af outputs, kunne Xbox kontrollerens analoge pinde indekseres og skaleres som vist i Tabel 1.

Tabel 1 Indeksering af Xbox kontroller analog pinde

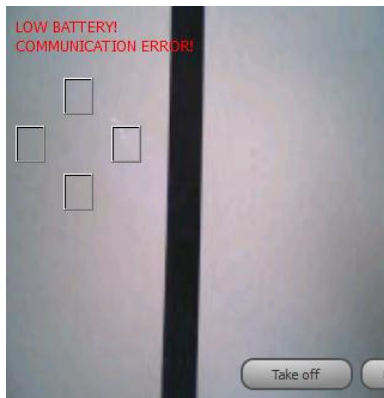
Xbox kontrollers analog input	Indeksnummer	Maksimum værdi	Minimum værdi
Analog venstre pind X-akse	0	32767	-32767
Analog venstre pind Y-akse	1	-32767	32767
Analog højre pind X-akse	3	32767	-32767
Analog højre pind Y-akse	4	-32767	32767

Fra Tabel 1 ses der at Y-aksen er invers på begge analogpinde, så når en analogpind er i bund, er outputtet maksimum, og når den analogpind er i top er outputtet minimum. Hver analogpind har derudover to indekseringer, en der viser outputtet for X-aksen og en for Y-aksen.

1.1.2 Autonavigation

Autonavigation er blevet testet ved at sætte programmet i debugging mode og manuelt holde AR.Dronen hen over en linje på gulvet, så man ved hjælp af bundkameraet kan se linjen på skærmen. Derefter er der trykket på Autonavigationsknappen i programmet. Der er sat et breakpoint i programmet, efter grid'et er blevet opdateret med databehandling fra billedet, så programmet viser det billede der er blevet behandlet. Fra Qt Creators watch vindue kan grid'ets variabler aflæses, og dette kan sammenlignes med billedet fra kameraet, for at verificere at billedbehandlingen virker efter hensigten.

Eksempler på billeder med tilhørende aflæst grid data, kommer herefter:

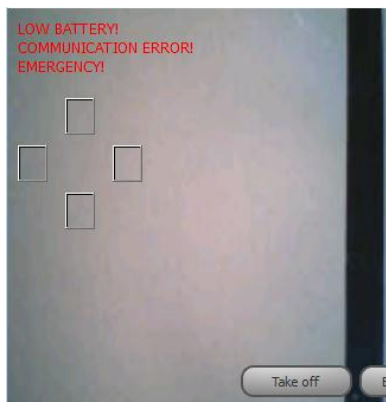


Figur 4

Hvis billedet AR.Dronen ser, er som i Figur 4, vil programmet sende en flyv fremad kommando til AR.Dronen.

Grid'er aflæses til følgende:

0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0



Figur 5

Grid Output:

0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1

Figur 5 vil føre til at programmet sender en flyv fremad kommando og flyv sidelæns til venstre samtidigt.



Figur 6

Grid Output:

0	0	0	1	1
0	0	1	1	0
0	0	1	1	0
0	1	1	0	0
0	1	1	0	0

Figur 6 vil føre til at programmet sender en flyv fremad kommando og drej til højre samtidigt.

Som det kan ses af de 3 test giver programmet et grid tilbage der repræsenterer en linje på gulvet godt. Dette giver et godt input til autonavigationsalgoritmen.

1.1.3 AT Commands

Til test af AT Commands og UDP forbindelsen er der blevet lavet et lille testprojekt [Bilag 2]. Testprojektets formål er at teste metoden til at sende kommandoer til AR.Dronen.

Når programmet startes, sender det en række kommandoer, der får AR.Dronen til at lette og svæve i 5 sek. for derefter at lande igen.

Testen blev kørt fra en computer der er logget på AR.Dronens WiFi netværk. Når programmet startes letter AR.Dronen og flyver op i ca. 1 m højde hvor den svæver i et par sekunder, hvorefter den lander igen. For at gentage processen skal programmet lukkes og startes igen.

1.1.4 Navigationsdata

Til at teste navigationsdataen blev en "AR.Drone dummy", der er et stykke software [Bilag 4] der blev udviklet til at simulere AR.Dronen benyttet. AR.Drone dummyen sættes op til at sende navigations data ud, hvorefter navigationsdata klassen testes ud fra denne navigationsdata.

Testen af navigationsdata klassen køres i Qt Creator, hvor programmet køres i Debug tilstand. Der sættes breakpoints de nødvendige steder, og variablerne aflæses i Qt Creators watch vindue.

Alt navigationsdata, der blev sendt fra AR.Drone dummyen, blev læst korrekt.

1.1.5 Styring vha. skærm input

Styring vha. skærmen testes ved at oprette forbindelse til AR.Dronen og derefter starte programmet op.

Først testes om man kan skifte kamera feed ved at trykke på knappen "Toggle camera". Derefter lettes AR.Dronen ved at trykke på knappen "Take off". Når AR.Dronen er i luften, testes de forskellige manøvrerings knapper.

Alle knapperne virkede efter hensigten.

1.1.6 DevKit8000

1.1.6.1 Linux kerne

1.1.6.1.1 Fremgangsmåde

Linux kernen kompileres på host computer, og det nye ulmage (kerne image) placeres på SD kortets boot partitionen. De nye kernemoduler kompileres ligeledes, og lægges over på SD kortets filsystems partitionen.

Hvis ikke boot parametrene er opsat korrekt, gøres dette.

1.1.6.1.2 Resultat

Den nye Linux kerne starter op, med de nye moduler tilsluttet og aktive.

1.1.6.2 WiFi på DevKit8000

1.1.6.2.1 Fremgangsmåde

Linux kernelen kompiles med den medfølgende kernel opsætning [Bilag 10], hvor kernen er konfigureret til at have wireless interface drivere, deriblandt RT73USB modulet, som WiFi USB'en benytter sig af.

Fremgangsmåden som beskrevet i afsnittet 1.1.6.1.1 Fremgangsmåde, benyttes til at opdatere kernen på SD kortet.

Efter DevKit8000 er startet op, bruges følgende kommando for at se om der registreres et trådløst interface.

```
[root@DevKit8000 ~]$ iwconfig
```

1.1.6.2.2 Resultat

Terminal på DevKit8000 giver følgende output:

```
[root@DevKit8000 ~]$ iwconfig  
eth0    no wireless extensions.  
usb0    no wireless extensions.  
wlan0    IEEE 802.11abgn ESSID:off/any  
          Mode:Managed Access Point: Not-Associated Tx-Power=off  
          Retry long limit:7 RTS thr:off Fragment thr:off  
          Power Management:off
```

Hvilket viser at der er et trådløst interface på wlan0.

1.1.6.3 Xbox kontroller

Linux kernelen kompiles med den medfølgende kernel opsætning [Bilag 10], hvor kernen er konfigureret til at have joystick drivere, deriblandt Microsoft Xbox kernel driver.

Fremgangsmåden som beskrevet i afsnittet 1.1.6.1.1 Fremgangsmåde, benyttes til at opdatere kernen på SD kortet.

Efter DevKit8000 er startet op, bruges følgende kommando for at se om DevKit8000 har registreret Xbox Wireless Receiver som et joystick input.

```
[root@DevKit8000 ~]$ dmesg | grep Xbox
```

Har Linux kernelen registreret tilslutningen af Xbox Wireless Receiver giver den output lign.:

```
[36449.989599] usb 1-1.5: Product: Xbox 360 Wireless Receiver for Windows
```

```
[36450.684014] input: Xbox 360 Wireless Receiver as  
/devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.5/1-1.5:1.0/input/input19  
[36450.684597] input: Xbox 360 Wireless Receiver as  
/devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.5/1-1.5:1.2/input/input20  
[36450.684868] input: Xbox 360 Wireless Receiver as  
/devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.5/1-1.5:1.4/input/input21  
[36450.685270] input: Xbox 360 Wireless Receiver as  
/devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.5/1-1.5:1.6/input/input22
```

Derefter findes dens characters device files ved kommandoen:

```
[root@DevKit8000 ~]$ ls /dev/input/ | grep js
```

Hvis der findes characters device files for Xbox Wireless Receiver bliver outputtet som følgende:

```
js0  
js1  
js2  
js3
```

Med 4 character device files, da Xbox Wireless Receiver kan have op til 4 trådløse Xbox kontrollers tilsluttet. Da Linux kernelen har lavet 4 character device files til Xbox kontrollerne, kan det konkluderes at det virker. Xbox driveren opretter Xbox kontrollerne i logisk rækkefølge, 0 til 3.

1.1.6.4 Qt på DevKit8000

1.1.6.4.1 Fremgangsmåde

Programmet som er vedlagt [Bilag 11] med en GUI, kompileres i Qt Creator med Qt versionen beregnet til ARM arkitekturen.

Programmet ligges over SD kortets filsystem partition, hvorefter DevKit8000 startes op og følgende kommando afvikles:

```
[root@DevKit8000 ~]$ ./Test_Qt -qws
```

1.1.6.4.2 Resultat

Programmet, Test_Qt, starter op, og en GUI med en knap dukker op.

1.1.6.5 Touchskærm på DevKit8000

1.1.6.5.1 Fremgangsmåde

DevKit8000 startes op. Efter opstart af DevKit8000 startes touchskærms kalibrering ved følgende kommando:

```
[root@DevKit8000 ~]$ ts_calibrate
```

1.1.6.5.2 Resultat

Et program åbnes, hvori det er muligt enten at kalibrere touchskærmen eller tegne.

1.2 Integrationstest

Den første enhed der virkede var videofeed fra AR.Dronen til programmet. Det er oven på dette program resten af projektet er bygget. For hver ny enhed der er blevet tilføjet programmet er det blevet testet om både enheden og den eksisterende funktionalitet i programmet virker. Integrationen af de forskellige enheder er ikke det samme som iterationerne, da en iteration godt kan indeholde mere end en enhed. Integrationstesten bruges for at finde fejl i implementeringen mellem programmet og nye enheder.

Programmet har virket hver gang der er blevet tilføjet en ny enhed.

1.3 Accepttest

1.3.1 Use Case 1.: System initialisering

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at DevKit8000 er slukket og har WiFi dongle, Xbox Wireless dongle og USB Hub tilsluttet samt strømforsyningen til DevKit8000 og USB hub er tilsluttet elnettet. Xbox kontrolleren skal tændes.

Handling	Udfald	Check box	Kommentar
Tænd DevKit8000 på power knappen imens boot knappen holdes nede.	Systemet startes op	✓	
Vent til programmet er startet (ca. 1min).	Systemet vil starte op i applicationen til at styrer AR.Dronen.	x	Systemet starter op i kommando prompt. Skriv: <code>"/ARDrone - qws"</code> for at starte programmet
Hvis Xbox kontroller ikke allerede er tændt hold Xbox logo -knappen nede indtil der er lys omkring den.	Xbox kontroller er sluttet til Xbox wireless receiver.	✓	
Tryk på settings knappen i højre tophjørne i programmet.	Indstillingsmenuen åbner	✓	Indstillingsmenuen åbne men med meget lang reaktionstid.

1.3.2 Use Case 2.: Opdater video feed

Startbetingelser:

AR.Dronen og fjernstyringssystemet er tændt.

Handling	Udfald	Check box	Kommentar
Flyt rundt på AR.Dronen.	Verificer at videofeedet opdateres direkte på skærmen.	✓	Test udført på bærbar computer da programmet kører meget langsom på DevKit8000.

1.3.3 Use Case 3.: Skift videokamera

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at fjernsystemet er initieret, og skærmen viser et af de 4 kamera video feeds.

Handling	Udfald	Check box	Kommentar
----------	--------	-----------	-----------

Tryk på camera knappen på skærmen.	Videoen på skærmen ændrer sig til den næste i rækken (der er 4 forskellige video feeds).	✓	Test udført på bærbar computer da programmet kører meget langsom på DevKit8000. Test udført på både skærm og Xbox kontroller.
------------------------------------	------------------------------------------------------------------------------------------	---	----------------------------------------------------------------------------------------------------------------------------------

1.3.4 Use Case 4.: Manøvrering – touch skærm

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at fjernsystemet er initieret.

Handling	Udfald	Check box	Kommentar
Tryk på takeoff knappen i bunden af skærbilledet.	AR.Dronen letter og svæver ca. 1m over punktet den littede fra.	✓	Denne og alle nedenstående test er udført på bærbar computer da programmet kører meget langsom på DevKit8000.
Tryk på flyv til højre knappen (Se manual for komplet oversigt over knapper [Appendix 1]).	AR.Dronen flyver sidelæns til højre så længe knappen holdes nede.	✓	
Tryk på flyv fremad knappen.	AR.Dronen flyver fremad så længe knappen holdes nede.	✓	
Tryk på flyv opad knappen.	AR.Dronen flyver højere op så længe knappen holdes nede.	✓	
Tryk på landing knappen.	AR.Dronen lander.	✓	

1.3.5 Use Case 5.: Opdater navigations data

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at fjernsystemet er initieret.

Handling	Udfald	Check box	Kommentar
Åben indstillingsmenuen og vælg show navigationdata. Luk indstillingsmenuen igen.	Navigationsdata vises under batteri indikator i venstre hjørne af skærmen.	✓	Denne og alle nedenstående test er udført på bærbar computer da programmet kører meget langsom på DevKit8000.

Flyt rundt på AR.Dronen med forskellige vinkler af AR.Dronen.	Navigationsdataen opdateres nu og burde vise forskellige værdier for vinkler.	√	
---------------------------------------------------------------	-------------------------------------------------------------------------------	---	--

1.3.6 Use Case 6.: Skift settings

Startbetingelser:

Fjernsystemet er initieret.

Handling	Udfald	Check box	Kommentar
Åben indstillingsmenu ved tryk på knappen i højre top hjørne på skærmen.	Indstillingsmenuen åbner.	√	Denne og alle nedenstående test er udført på bærbar computer da programmet kører meget langsom på DevKit8000.
Nu vælges at vise lavt batteri og ved siden af boxen skrives tallet 101	Lavt batteri alarm vises nu.	√	
Tryk på settings knap igen, og tryk save.	Indstillingsmenu lukkes igen.	√	

1.3.7 Use Case 7.: Manøvrering – Xbox kontroller

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at fjernstyringssystemet er initieret og Xbox Wireless receiver er tilsluttet og tændt.

Handling	Udfald	Check box	Kommentar
Tryk på start knappen på Xbox kontrolleren.	AR.Dronen letter og svæver ca. 1m over punktet den littede fra.	√	Denne og alle nedenstående test er udført på bærbar computer da programmet kører meget langsom på DevKit8000.
Hold den venstre analogpind til højre(Se manual for komplet oversigt over manøvrering [Appendisk 1]).	AR.Dronen flyver sidelæns til højre.	√	
Hold den venstre	AR.Dronen flyver	√	

analogpind til venstre.	sidelæns til venstre		
Hold den venstre analogpind opad.	AR.Dronen flyver højere opad.	✓	
Hold den venstre analogpind til venstre.	AR.Dronen snurrer til venstre rundt om sig selv.	✓	
Tryk på Back knappen.	AR.Dronen lander.	✓	

1.3.8 Use Case 8.: Auto-navigation efter linje på gulvet

Startbetingelser:

AR.Dronen er aktiveret og tændt samt at fjernstyringssystemet er initieret.

Handling	Udfald	Check box	Kommentar
Flyv AR.Dronen hen over linjen i ca. 1.8m højde og peg den i den retning den skal følge linjen. Brug use case 4 eller 7 til dette.	AR.Dronen er i startposition over linjen.	✓	Denne og alle nedenstående test er udført på bærbar computer da programmet kører meget langsom på DevKit8000.
Tryk på autonavigationsknappen i programmet.	AR.Dronen flyver efter linjen på gulvet, og vil stoppe når den ikke længere kan se linjen. På interfacet vil der stå Autoflying.	✓	
Tryk på autonavigationsknappen i programmet.	AR.Dronen stopper med autonavigation.	✓	