

Report Week 45, Group Sort-Bot

IE303812 Real Time Programming

Student 1 Inga Lill Bjølstad

Student 2 Malén Barstad

Student 3 Demy Patrick Gielesen

Student 4 Aleksander Solheim

Project idea

Sorterings-Robot. Vi skal ha satt opp ei GUI som representerer et fast område (kart) der en robot på hjul skal kunne kjøre. På det området som roboten befinner seg vil det vere objekter som roboten skal kunne gjenkjenne enten ved fasong eller farge. Både roboten og objektene (foreløpig kalt “klosser”) er feldig plassert på det angitte området.

Funksjonen alt de blir at vi skal kunne si roboten hvor hver av klossene skal bli plassert ved å markere område på kartet via GUIen. Roboten skal da kunne kjenne igjen og finne disse klossene og se dem på angitt sted.

Detailed Project Description

Koden vår er delt opp i to deler (java projects). Vi har ei del som er dedikert til kjøre på odroid og en del som er dedikert til å kjøre på ei pc/mac som en GUI. Hver av disse har egne oppsett med hensyn på tråder og kommuniserer kun via Wifi. Gui skal kunne vise fram det roboten ser, samt med at den kan endre på robotens “modus” og kunne manuel-kjøre roboten.

Gui er oppsett som en klient for en server som kjører på odroiden. Client-klassa setter opp en ny tråd som kommuniserer med serveren og kommer deretter til å vente på “triggers” frå GUI før den sender en kommando. Samtidig tar client-klassa imot en konstant strøm av bilder som brukes til å vise en videofeed frå odroid. Videofeeden er fullstendig uendret og blir sendt direkte fra odroiden så fort som den klarer (det kan hende at vi endrer dette i etterkant til å bli ei bestemt FPS for å minske load på odroid). Ettersom alt på GUI skal kun bli kontrollert av en enkel konstant prosess, valgte vi å ha ingen trådar på GUI. GUI kjører allerede på ei pc, som kan bli tenkt på som sin egen ekstern tråd i forhold til odroid. Med unntak av de få bakliggende prosesser som er tilstede som feks GUI runnable-vindu og client sockets.

Odroid delen er bygd på 3 grunnleggende prosesser som kjører samtidig. Disse er:

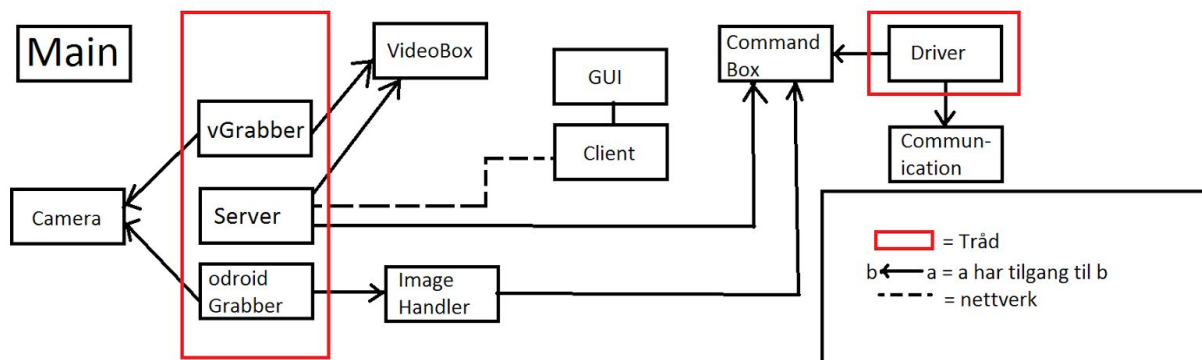
- **Arduino kontroll delen;** Vi har en hovedklasse som styrer alt som har med arduino å gjøre. Den leser fra en såkalt “CommandBox” som vi bruker til å lagre kommandoene som GUI har gitt odroiden. Alt som har med oversettelse frå GUI “triggers” til kommandoer som arduinoen

kan forstå har ArduinoDriver-klassa i oppgave å fullføre. Den sender deretter kommandoen via ArduinoCommunication-klassa som er den klassa som oppnår kommunikasjon og skriver/leser fra Serial streamen.

- **Video prosessering;** Når odroid delen blir startet, så er det første den gjør er å sette opp kommunikasjon med et kamera. Webcam/kamera har den funksjonen at de leser informasjon fra omgivelsene når kameraet er på uansett hva. Dette vil si at selve om programmet ikke bruker bildene så har kameraet en konstant videostream. Ut fra dette lagde vi ei såkalt “framegrabber” som lager et bilde av video streamen akkurat når den har lyst. Framegrabber er derfor en tråd. Den framegrabberen som er brukt til å prosessere bilder har blitt satt på et intervall av 0.5sek (kommer helt sikkert til å bli endra til litt raskere i senere tid). Vi har gjort dette fordi at bildebehandling tar en god stund, og kan potensielt få odroid til å krasje eller andre rare ting. Ved å sette bildebehandling på et lite interval så slipper vi at prosessen hopper seg opp og likevel så vil odroid klare å stille hjulene riktig vei.
- Videostream for GUI; er bygd på samme måte som video prosesserings delen med unntak at den ikke skal endre på bildene i det hele tatt, men derimot sende dem til en lagringsplass “VideoBox” slik at serveren kan lese dem og sende til GUI. Her blir det igjen brukt et framegrabber objekt som leser fra kameraets-videostream så rask som den bare klarer (altså ikke oppsatt timer) for å generere en “video”.

Som på GUI delen så er det små tråder i bakgrunn som kjører som for ting til å fungere. Disse ser vi vekk ifra. Et eksempel på odroid bakgrunnsprosesser er Server sockets.

Class-diagram



Her er vGrabber og OdroidGrabber framegrabber objekter som tar bilder av videostreamen til kameraet. Driver er hovedklassen til arduino som leser kommandoene fra Commandbox og sender via Communication til arduino. VideoBox er lagrings boksa til bildene som blir sendt til GUI.

GUI og clienten er lagt inn i eget prosjekt som er koblet til samme port som serveren. Client og GUI er nå en klasse.

Previous week status

Fikk gjort ferdig manuell modus. Holder på å få sende kommandoer fra klienten til serveren som skal styre roboten.

Current status

Skal begynne å teste at automodus på roboten fungerer. Holder på med å mappe knappene på gui til String kommandoer som skal bli sendt til server.

Software

Begynt å opprette pakker som skal vise hva vi sender over serveren. Har satt inn ActionListener på hver knapp for å skille knappenes egenskaper fra hverandre.

Hardware

Har fått satt sammen roboten, så nå er den ferdig montert.

Problems encountered and next steps

Få serveren til å fungere med kommandoene for å styre roboten.

Få arduino til å fungere med Odroid.

Report Week 43, Group Sort-Bot

IE303812 Real Time Programming

Student 1 Inga Lill Bjølstad

Student 2 Malén Barstad

Student 3 Demy Patrick Gielesen

Student 4 Aleksander Solheim

Project idea

Sorterings-Robot. Vi skal ha satt opp ei GUI som representerer et fast område (kart) der en robot på hjul skal kunne kjøre. På det området som roboten befinner seg vil det vere objekter som roboten skal kunne gjenkjenne enten ved fasong eller farge. Både roboten og objektene (foreløpig kalt "klosser") er feldig plassert på det angitte området.

Funksjonen alt de blir at vi skal kunne si roboten hvor hver av klossene skal bli plassert ved å markere område på kartet via GUIen. Roboten skal da kunne kjenne igjen og finne disse klossene og se dem på angitt sted.

Detailed Project Description

Koden vår er delt opp i to deler (java projects). Vi har ei del som er dedikert til kjøre på odroid og en del som er dedikert til å kjøre på ei pc/mac som en GUI. Hver av disse har egne oppsett med hensyn

på tråder og kommuniserer kun via Wifi. Gui skal kunne vise fram det roboten ser, samt med at den kan endre på robotens “modus” og kunne manuel-kjøre roboten.

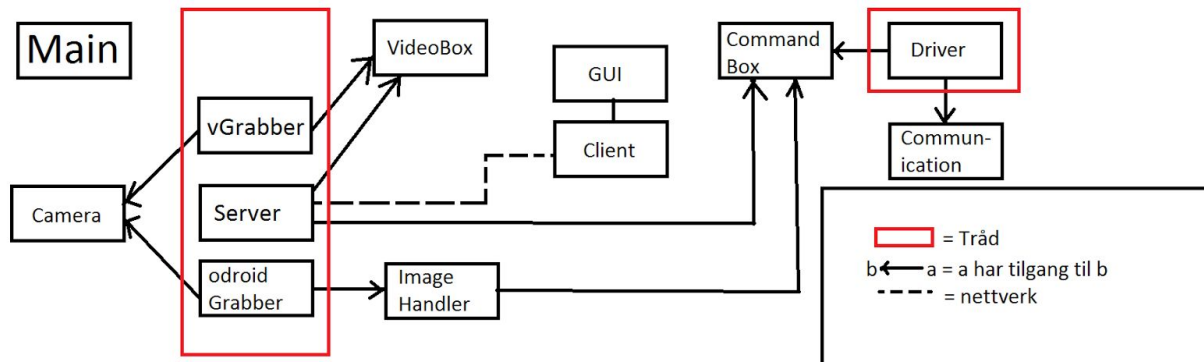
Gui er oppsett som en klient for en server som kjører på odroiden. Client-klassa setter opp en ny tråd som kommuniserer med serveren og kommer deretter til å vente på “triggers” fra GUI før den sender en kommando. Samtidig tar client-klassa imot en konstant strøm av bilder som brukes til å vise en videofeed fra odroid. Videofeeden er fullstendig uendret og blir sendt direkte fra odroiden så fort som den klarer (det kan hende at vi endrer dette i etterkant til å bli ei bestemt FPS for å minske load på odroid). Ettersom alt på GUI skal kun bli kontrollert av en enkel konstant prosess, valgte vi å ha ingen trådar på GUI. GUI kjører allerede på ei pc, som kan bli tenkt på som sin egen ekstern tråd i forhold til odroid. Med unntak av de få bakliggende prosesser som er tilstede som feks GUI runnable-vindu og client sockets.

Odroid delen er bygd på 3 grunnleggende prosesser som kjører samtidig. Disse er:

- **Arduino kontroll delen;** Vi har en hovedklasse som styrer alt som har med arduino å gjøre. Den leser fra eni såkalt “CommandBox” som vi bruker til å lagre kommandoene som GUI har gitt odroiden. Alt som har med oversettelse fra GUI “triggers” til kommandoer som arduinoen kan forstå har ArduinoDriver-klassa i oppgave å fullføre. Den sender deretter kommandoen via ArduinoCommunication-klassa som er den klassa som oppnår kommunikasjon og skriver/leser fra Seriel streamen.
- **Video prosessering;** Når odroid delen blir startet, så er det første den gjør er å sette opp kommunikasjon med et kamera. Webcam/kamera har den funksjonen at de leser informasjon fra omgivelsene når kameraet er på uansett hva. Dette vil si at selve om programmet ikke bruker bildene så har kameraet en konstant videostream. Ut fra dette lagde vi ei såkalt “framegrabber” som lager et bilde av video streamen akkurat når den har lyst. Framegrabber er derfor en tråd. Den framegrabberen som er brukt til å prosessere bilder har blitt satt på et intervall av 0.5sek (kommer helt sikkert til å bli endra til litt raskere i senere tid). Vi har gjort dette fordi at bildebehandling tar en god stund, og kan potensielt få odroid til å krasje eller andre rare ting. Ved å sette bildebehandling på et lite interval så slipper vi at prosessen hopper seg opp og likevel så vil odroid klare å stille hjulene riktig vei.
- Videostream for GUI; er bygd på samme måte som video prosesserings delen med unntak at den ikke skal endre på bildene i det hele tatt, men derimot sende dem til en lagringsplass “VideoBox” slik at serveren kan lese dem og sende til GUI. Her blir det igjen brukt et framegrabber objekt som leser fra kameraets-videostream så rask som den bare klarer (altså ikke oppsatt timer) for å generere en “video”.

Som på GUI delen så er det små tråder i bakgrunn som kjører som for ting til å fungere. Disse ser vi vekk ifra. Et eksempel på odroid bakgrunnsprosesser er Server sockets.

Class-diagram



Her er vGrabber og OdroidGrabber framegrabber objekter som tar bilder av videostreamen til kameraet. Driver er hovedklassen til arduino som leser kommandoene fra Commandbox og sender via Communication til arduino. VideoBox er lagrings boksa til bildene som blir sendt til GUI.

Previous week status

Vi har fullført basic-kommunikasjon mellom client og server klasser og holder på å teste ut med å sende et enkelt bilde over. Arduino delen er på god gang til å kunne sende kommandoer til arduinoen ved å forstå enkle bool kommandoer fra commandbox, men sliter litt med å få koda til å vente på svar fra arduino(Seriel kommunikasjon har ei viss delay som java må ta høyde for). GUI er nærmere ferdig på layout delen og skal snart bli implementert som et eget netbeans prosjekt, ettersom nå har GUI på odroid for debugging. Bildebehandling er så å si ferdig med unntak for support for å velge hvilken farge odroid skal filtrere mot.

Current status

- Kommunikasjon mellom ei client og ei server klasse er blitt fullført og vi skal dermed lage client delen som et eget prosjekt der vi kan videre utvikle på GUI.
- Fastmontere flere deler
- Lage support for å velge hvilken farge odroiden skal følge etter
- Forbedre GUI
- Printe deler

Software

I programmet kan vi nå kjøre video og bildebehandling, velge farge på objektet og gjort endringer slik at bildebehandling ikke tar like mye prosessering kraft.

Koden fra netbeans til arduino kan nå velge kjøreretning og velge fart via netbeans.

Hardware

Vi har fått inn det meste av utstyr nå, med unntak av batterier som er på vei. Den første blokka vi har printa ut har en litt for general farge (blå) og må sannsynligvis bli malt med ei anna farge for at vi skal kunne detektere den uten støy. Skal kanskje printe ut enda en blokk, eller gå for en grønn tennisball som neste objekt som vi kan detektere.

Problems encountered and next steps

Arduino har hatt litt problemer med at kommandoene vi har sendt til den ikke alltid gir tilbakemelding, noe som får Driveren til å vente i uendelig tid. Har også funnet ut at viss roboten er i gang på "auto-modus" så vil vi nok ikke klare å stoppe programmet før det har blitt fullført. Vi skal se nærmere på dette senere.

GUI, slik som den har blitt satt opp fram til nå, setter verdier til true/false alt ettersom ka vi trykker på. Dette blir feil i forhold til socket kommunikasjon. Vi skal prøve å få til triggers frå GUI slik at det blir kun send kommandoer til odroid kun når vi har endra noke i GUI.

Report Week 41, Group Sort-Bot

IE303812 Real Time Programming

Student 1 Inga Lill Bjølstad

Student 2 Malén Barstad

Student 3 Demy Patrick Gielesen

Student 4 Aleksander Solheim

Project idea

Sorterings-Robot. Vi skal ha satt opp ei GUI som representerer et fast område (kart) der en robot på hjul skal kunne kjøre. På det området som roboten befinner seg vil det vere objekter som roboten skal kunne gjenkjenne enten ved fasong eller farge. Både roboten og objektene (foreløpig kalt "klosser") er tilfeldig plassert på det angitte området.

Funksjonen SI alt deHe blir at vi skal kunne si SI roboten hvor hver av klossene skal bli plassert ved å markere eH område på kartet via GUIen. Roboten skal da kunne kjenne igjen og finne disse klossene og seHe dem på angitt sted.

Previous week status

Gjenkjenne objekter med kamera er godt i gang og kan lett finne objektet vi ønsker og kun objektet vi ønsker.

Kommunikasjon mellom arduino er oppe å går og kan sende kommandoer til arduino via netbeans. Satt på avstandssensorer.

Satt opp GUI og gjort små endringer på knappe konfigurasjon i GUI

OpenCV og RXTX bibliotek er satt opp og fungerende på alle maskiner i gruppen.

Current status

- Begynne å sette opp server/klient
- Fastmontere flere deler
- Utvide bildebehandlingen
- Forbedre GUI
- Printe deler

Software

I programmet kan vi nå kjøre video og bildebehandling, velge farge på objektet og gjort endringer slik at bildebehandling ikke tar like mye prosessering kraft.

Koden fra netbeans til arduino kan nå velge kjøreretning og velge fart via netbeans.

Hardware

Bestilt opp noe mer hardware, som batteri til Odroid og lader, hentet noen avstands sensorer som skal kobles på.

Satt flere deler 3D deler på print, plog og en cube i annen farge.

Problems encountered

...

Next steps

Server/klient gjøres ferdig, få trødløs kommunikasjon mellom Odroid og pc,

Report Week 39, Group Sort-Bot

IE303812 Real Time Programming

Student 1 Inga Lill Bjølstad

Student 2 Malén Barstad

Student 3 Demy Patrick Gielesen

Student 4 Aleksander Solheim

Project idea

Sorterings-Robot. Vi skal ha satt opp ei GUI som representerer et fast område (kart) der en robot på hjul skal kunne kjøre. På det området som roboten befinner seg vil det vere objekter som roboten skal kunne gjenkjenne enten ved fasong eller farge. Både roboten og objektene (foreløpig kalt "klosser") er tilfeldig plassert på det angitte området.

Funksjonen SI alt deH blir at vi skal kunne si SI roboten hvor hver av klossene skal bli plassert ved å markere eH område på kartet via GUIen. Roboten skal da kunne kjenne igjen og finne disse klossene og seH dem på angitt sted.

Previous week status

Har fått roboten og Odroid. Har fått startet å seH på Odroid. Startet på å gjenkjenne objekter ved kamera. FåH printet ut firkant i mørkeblå som skal bli brukt SI å bli sortert. Disse klarer roboten å kjenne igjen.

På roboten har vi fått fikset opp med å seH ledningene på riktig plass, fått laget et lite program SI å styre roboten.

Current status

Få Arduino SI å fungere på netbeans. Begynt på jobbe med GUI og jobbe med å lage flere komponenter som skal 3D printes. FåH SI det meste med OpenCV for å begynne å skrive på det. SaH opp klssediagram for Odroid.

Software

På Koden har vi fått roboten SI å kjørt med Arduino. SI roboten har vi begynt å seH på OpenCV for å lære oss hvordan det fungerer for å kunne begynne med det. Begynt med installasjon av diverse program vi trenger.

Hardware

FåH hardwaren vi skal bruke, har seH den sammen og har begynt å koble det opp SI software.

Problems encountered

Problemer vi har møH på er et problem med macbook med nedlastning av openCV, deH er noe vi jobber med å prøve å få løst. Andre problemet er at det er vanskelig å få kjørt en Odroid på batterier, må finne en løsning der.

Next steps

Få ferdig pløgen som skal 3D printes, få opp kommunikasjon mellom komponentene, finne ut hva vi kan gjøre med batteriene til odroiden, få openCV på macbook.

Report Week 37, Group: Sort-Bot

IE303812 Real Time Programming

Student 1 Inga Lill Bjølstad

Student 2 Malén Barstad

Student 3 Demy Patrick Gielesen

Student 4 Aleksander Solheim

Project idea

Sorterings-Robot. Vi skal ha satt opp ei GUI som representerer et fast område (kart) der en robot på hjul skal kunne kjøre. På det området som roboten befinner seg vil det vere objekter som roboten skal kunne gjenkjenne enten ved fasong eller farge. Både roboten og objektene (foreløpig kalt "klosser") er tilfeldig plassert på det angitte området.

Funksjonen til alt dette blir at vi skal kunne si til roboten hvor hver av klossene skal bli plassert ved å markere ett område på kartet via GUIen. Roboten skal da kunne kjenne igjen og finne disse klossene og sette dem på angitt sted.

Previous week status

Fikk ikke gjort så mye siden flere personer på gruppen hadde konte eksamen. Har laget første utkast av rapport og funnet ut hva slags hardware vi trenger i første omgang.

Current status

Skal begynne å tegne klassediagram, finne ut hvordan vi skal sette opp skjelettet for applikasjon.

Software

Inntil videre har vi ikke begynt å kode. Vi har satt opp et prosjekt på github, fått testet hvordan det fungerer sammen med Netbeans.

Hardware

Det mekaniske på prosjektet er det vanskelig å planlegge inntil vi får roboten vår så vi får se hvordan vi kan bygge på den, og eventuelt hvor vi kan plassere delene. Kommet fram til vi vil bruke bluetooth.

Problems encountered

Ikke fått startet med prosjektet på grunn av konte eksamener.

Next steps

Neste uke planlegger vi å få sammen hardware og begynne få opp kommunikasjon mellom enhetene så lenge vi har fått delene.