SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1867

INTEGRACIJA SENZORA NA AUTONOMNI KAJAK

Tomislav Maras

Zagreb, lipanj 2011.

Sadržaj

Uvod

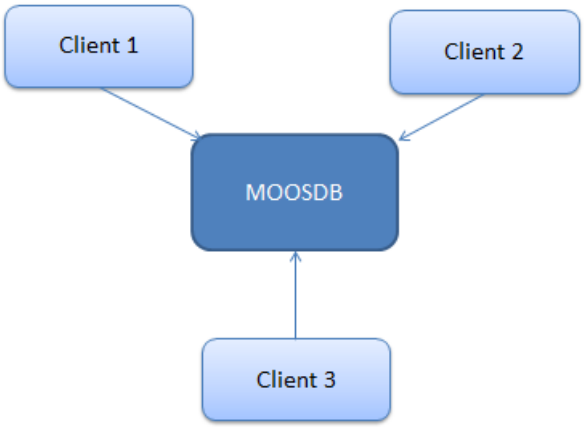
Svi smo bar jednom poželjeli da dan traje duže od 24 sata. Životni ritam prosječnog čovjeka je užurban i prenatrpan. S jedne su strane želje, obaveze i planovi, a s druge su mogućnost i vrijeme. Vrijeme teče i nikog ne čeka. Često čujemo ljude kako traže više vremena. Zato se u današnje vrijeme puno ulaže u procese i sustave koji će smanjiti opterećenje čovjeka, pružiti nam manje brige i više vremena.

Tu ulaze i autonomni sustavi koji se još uvijek istražuju i njihove se mogućnosti preispituju. Postoji i čitav niz natjecanja koja su vezana uz autonomne sustave i na kojima sudjeluju studenti i mladi znanstvenici. Takva se natjecanja održavaju u Europi i SAD-u. U razvijenijim svjetskim državama vlada praksa da svako tehničko sveučilište ima svoje autonomnom vozilo. Na tom vozilo rade studenti zajedno sa profesorima i često se takvo vozilo generacijski nadograđuje i poboljšava. U takav se projekt upustio u Fakultet elektrotehnike i računarstva iz Zagreba. Cilj je osmisliti i napraviti autonomni kajak koji će biti temelj za daljnje osmišljavanje, istraživanje i unprijeđivanje autonomnog sustava.

Zadatak ovog Završnog rada je bio uklopiti i podesiti sve potrebne senzore za osnovnu funkcionalnost autonomnog kajaka. Predviđeno je postavljanje kompasa, GPS-a i video kamere koja se koristi za slanje video signala sa autonomnog kajaka.

Sustav MOOS

MOOS je nastao kao platforma za robotska istraživanja. Otvorenog je koda i pisan je u programskom jeziku C++. Njegove biblioteke olakšavaju rješavanje mnogih problema koji se javljaju pri komunikaciji više različitih elemenata.
Sam sustav ima zvjezdastu topologiju (slika 1). U središtu je poslužiteljska aplikacija, a ostale klijentske aplikacije svaku komunikaciju obavljaju preko središnje. Središnja aplikacija se naziva MOOS baza podataka („MOOS Database“), dok su klijentske aplikacije MOOSApp.



Slika 1 Primjer strukture MOOS-a sa 2 klijenta

Ovakva mreža ima sljedeća svojstva:

- Nema izravne komunikacije između klijenata
- Svaku komunikaciju započinje klijent
- Svaki klijent ima jedinstveno ime
- Pojedini klijent ne treba znati da postoje ostali klijenti
- Klijent ne može poslati podatke drugom klijentu, može ih poslati samo središnjoj poslužiteljskoj aplikaciji
- Mreža može biti rasprostranjena na bilo kojim računalima i na bilo kojem operacijskom sustavu

Iako takva struktura usporava komunikaciju zbog poslužitelja posrednika, ona ima velike prednosti. Vrlo važna prednost je jednostavnost mreže neovisno o broju klijenata, poslužitelj ima uvid u sve aktivne konekcije i može alocirati resurse

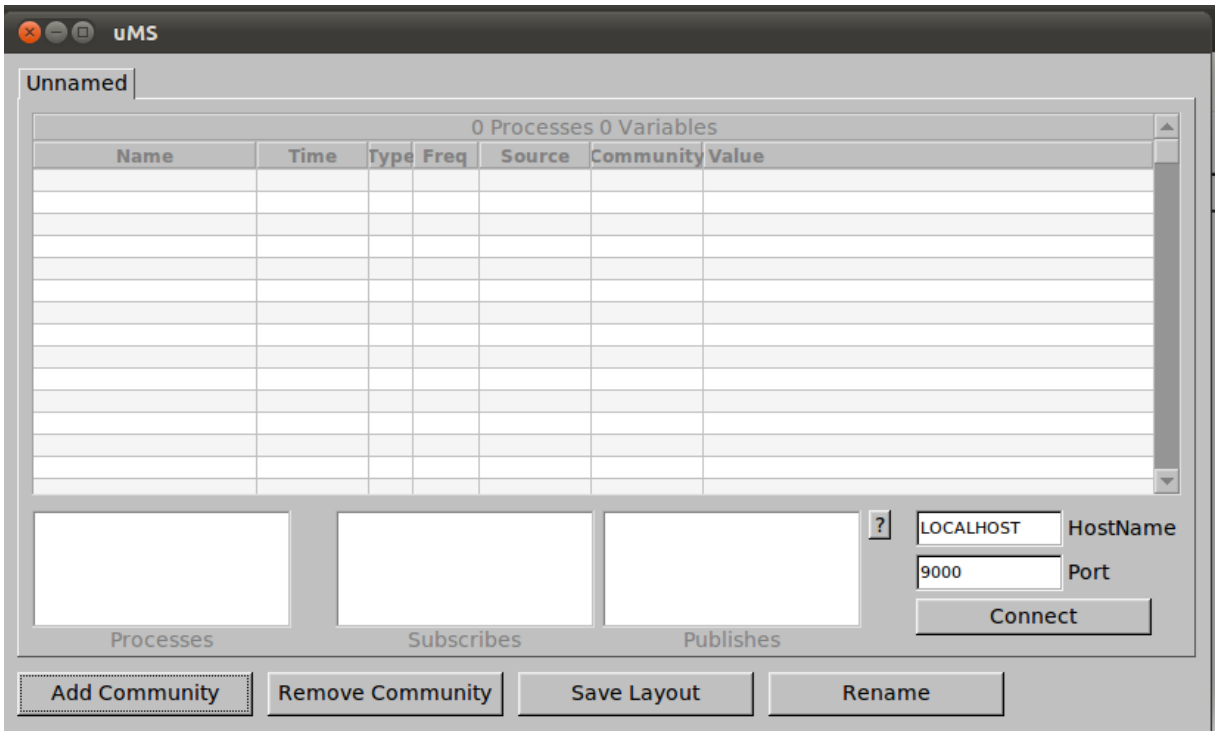
prema potrebi. Druga prednosti je to da ukoliko jedan klijent ima grešku, on ne može izravno utjecati na ostale klijente.
Još jedna od prednosti MOOS sustava je ta što koristi procese umjesto dretve. Takva odluka povećava stabilnost sustava, ukoliko jedan klijent postane nestabilan, samo on se gasi, ostali ostaju raditi. Moguće je koristiti i različita računala, operacijske sustave pošto je svaka komponenta svoja aplikacija i sprječava se interferencija podataka.
Poslužitelj MOOSDB glavni je dio MOOS sustava. On služi kao posrednik između svih komunikacija. MOOSDB se može gledati kao kontrolna ploča u kojoj je spremljeno trenutno stanje sustava. Njemu imaju pristup svi klijenti koji su se prethodno spojili na sustav. MOOSDB ima mogućnost da ne pamti samo trenutna stanja, već pamti i prošla stanja.
Sustav MOOS omogućuje klijentima da se pretplate na određene podatke. Ukoliko jedan klijent koristi podatke drugog klijenta on se na njih pretplaćuje pozivom metode Register kojom specificira ime podatka koji želi primati i kojom brzinom, u sustavu za autonomni kajak koristi se vrijednost 0 kojom se specificira da se podaci šalju maksimalnom brzinom.

Programsko ostvarenje MOOS klijenata

MOOS nudi klasu CMOOSApp koja olakšava programsko ostvarenje MOOS klijenata. Sama CMOOSApp klasa je osmišljena kao beskonačna petlja koja stalno poziva metodu Iterate() koja po postavkama ne radi ništa. Zadaća programera je u tu metodu napisati ono što želi da se izvršava. Osim što stalno poziva metodu Iterate() klasa CMOOSApp provjerava ako postoje svježi podaci na koje je MOOS klijent pretplaćen. Ukoliko postoje svježi podaci CMOOSApp poziva metodu OnNewMail(). U OnNewMail() metodu programer smješta kod koji obrađuje pristigle podatke. Osim navedenih metoda postoje i metode OnConnectToServer() i OnStartup(). Metoda OnConnectToServer() se poziva prilikom spajanja klijenta na poslužitelj, a metoda OnStartup() se poziva prilikom pokretanja aplikacije.

Uvid u rad aplikacije

Najjednostavniji način za testiranje i uvid u rad aplikacije je pokretanje alata uMS. uMS se spaja na MOOSDB poslužitelj kao i ostali MOOS klijenti, i omogućuje uvid u trenutno stanje na poslužitelju, odnosno omogućuje pregled svih varijabli koje se trenutno nalaze na poslužitelju. Unutar uMS-a moguće je mijenjati i dodavati nove varijable. Izgled uMS sučelja nalazi se na sljedećoj slici.



Slika 2 Izgled uMS sučelja

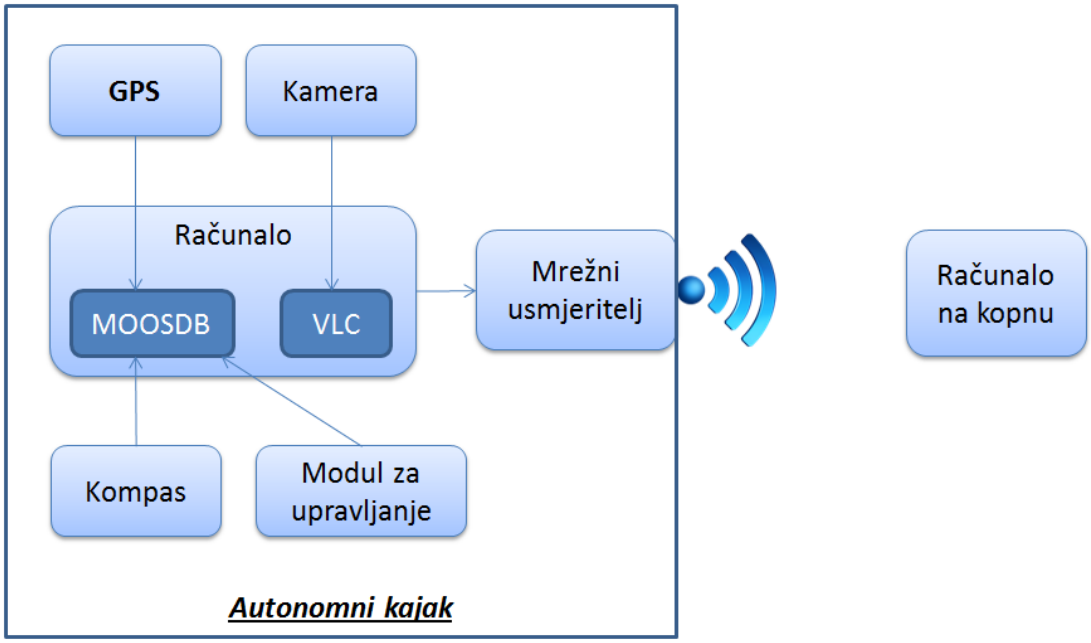
Nakon što je pokrenut MOOSDB, pokrene se uMS i odabere se opcija „Connect“ kako bi se dohvatile sve varijable iz sustava. Pritom je potrebno podesiti parametre "HostName" i "Port". Kada se uMS spoji na MOOSDB počinje prikazivati sve varijable i njihove trenutne vrijednosti.

Ugradnja senzora

Podaci sa senzora koji su ugrađeni na autonomni kajak trebaju biti jednostavno dostupni na višim razinama upravljanja. Zbog toga se podaci sa GPS uređaja i kompasa dohvaćaju preko sustava MOOS, dok je slanje video signala ostvaren uz pomoć programskog alata VLC[1]. Cijeli sustav komunikacije na autonomnom kajaku je ostvaren koristeći sustav MOOS. Upravo zbog toga je podacima sa GPS-a i kompasa najlakše pristupati ako su oni integrirani sa MOOS sustavom. Tako dohvaćeni podaci spremaju se u MOOSDB i lako im se pristupa.

Podešavanje i postavljanje senzora, način povezivanja, kao i problemi na koje se nailazi biti će opisani u nastavku rada.

Na sljedećoj slici nalazi se shema koja prikazuje povezanost svih modula potrebnih za rad autonomnog kajaka. Vidimo da je centralni dio računalo na kojeg se spajaju svi moduli. Računalo obrađuje dobivene podatke i obavlja potrebne akcije. Pomoću mrežnog usmjeritelja uvedena je i mogućnost spajanja sa računalom na kopnu kako bi se dobio trenutni uvid u rad autonomnog kajaka.



Slika 3 Shema povezanosti elemenata na autonomnom kajaku

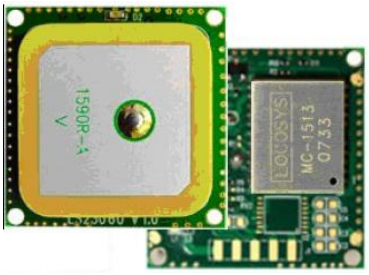
[1] VideoLan – besplatni alat otvorenog koda za emitiranje i prikazivanje multimedijskog sadržaja

GPS uređaj

U samoj navigaciji autonomnog kajaka, najbitniji senzor je GPS. Pozicija na kojoj se autonomni kajak nalazi izrazito nam je potrebna i bez tog podatka nikakva se misija ne bi mogla isplanirati. Određivanje pozicije treba biti što brže i što preciznije.

Odabran je GPS uređaj tvrtke Locosys, model LS20030. Njegove karakteristike su:

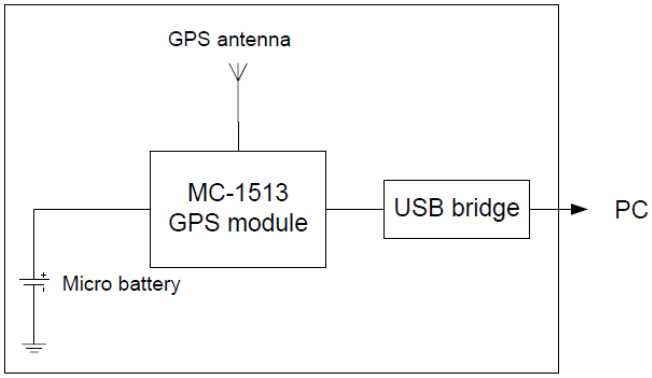
- 32 kanala
- USB priključak za spajanje
- Osvježenje podataka od 1 – 5 Hz
- Ugrađena baterija u sam uređaj radi što bržeg dohvaćanja podataka
- Greška u horizontalnom pozicioniranju ~ 3m
- Ukoliko radi na 5V potreban je stabilizator napona
- LED[2] indikator postavljenosti GPS-a



Slika 4 GPS Locosys LS20030, fizički izgled

GPS uređaj spaja se na računalo putem USB priključka. Nakon spajanja potrebno je pričekati nekoliko trenutaka kako bi se uređaj pozicionirao i moga slati ispravne podatke.

[2] Light-emmiting diode



Slika 5 GPS Locosys LS20030, sistemski blok dijagram

Obrada podataka iz GPS-a

Nakon spajanja računala i GPS uređaja potrebno je pričekati da uređaj uspostavi vezu sa satelitima. To nam signalizira crvena LED bljeskalica. Kada ona počne bljeskati, može se pokrenuti klijent MOOSNavi. To je klijent koji obrađuje podatke pristigle sa GPS uređaja. Podaci sa čitaju sa USB vrata na koji je priključen GPS putem klase SimpleSerial. GPS uređaj šalje više tipova poruka koje su zapisane u obliku NMEA[3].

Tablica 1 NMEA poruke

Kod poruke	Opis
GGA	Fiksirani podaci za globalno pozicioniranje.
GLL	Geografska pozicija – geografska širina i geografska dužina.
GSA	GNSS[4] DOP[5] i aktivni sateliti
GSV	GNSS dostupni sateliti
RMS	Preporučeni minimalni specificirani GNSS podaci
VTG	Kurs i brzina

[3] Specifikacija za komunikaciju između morskih električnih uređaja
[4] Globalni navigacijski satelitski sustavi
[5] Dilution of Precision

Za potrebe autonomnog kajaka dovoljan je GGA zapis. U njemu se nalaze svi potrebni podaci, odnosno geografska širina i dužina. GGA poruke se obrađuju i dobiveni podaci spremaju se na MOOS poslužitelju. Time je pristup podacima sa GPS-a omogućen i drugim klijentima.

Primjer GGA poruke je:

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Tablica 2 Dijelovi GGA poruke

Ime	Primjer	Opis
ID poruke	$GPGGA	GGA protokol
UTC vrijeme	123519	hhmmss
Geografska širina	4807.038	ddmm.mmm
Indikator N/S	N	N=north; S=south
Geografska dužina	01131.000	dddmm.mmm
Indikator E/W	E	E=east; W=west
Indikator pozicije	1	
Broj korištenih satelita	08	Od 0 do 12
HDOP	0.9	Horizontal Dilution Of Precision
Nadmorska visina	545.4	
Jedinica	M	Metri
Visina geoida	46.9	
Jedinica	M	Metri
Prazno polje		
Kontrolna suma	*47	

Kada MOOSNavi klijent primi GGA poruku, kreće njena obrada. Poruka se parsira i odvajaju se geografska dužina, geografska širina te potrebni indikatori. Dobiveni podaci spremaju se u varijable Longitude, Latitude, Indic_EW i Indic_NS. Sadržaj metode Iterate() koji demonstrira obradu podataka nalazi se na sljedećoj slici.

```
29  bool CSimulator::Iterate() {
30      size_t longitudeStart;
31      size_t longitudeEnd;
32      size_t latitudeStart;
33      size_t latitudeEnd;
34      size_t pom;
35      string temp;
36
37      string usbData = serial.readString();
38      if (usbData.substr(0, 6) == "$GPGGA") {
39          pom = usbData.find(",");
40          temp = usbData.substr(pom + 1);
41          latitudeStart = temp.find(",") + 1;
42          temp = temp.substr(latitudeStart);
43          latitudeEnd = temp.find(",");
44          latitude = temp.substr(0, latitudeEnd);
45          temp = temp.substr(latitudeEnd + 1);
46          indic_NS = temp.substr(0, 1);
47          temp = temp.substr(2);
48          longitudeEnd = temp.find(",");
49          longitude = temp.substr(0, longitudeEnd);
50          temp = temp.substr(longitudeEnd + 1);
51          indic_EW = temp.substr(0, 1);
52      }
53      m_Comms.Notify("Latitude", latitude, 0);
54      m_Comms.Notify("Longitude", longitude, 0);
55      m_Comms.Notify("Indic_EW", indic_EW, 0);
56      m_Comms.Notify("Indic_NS", indic_NS, 0);
57      return true;
58  }
```

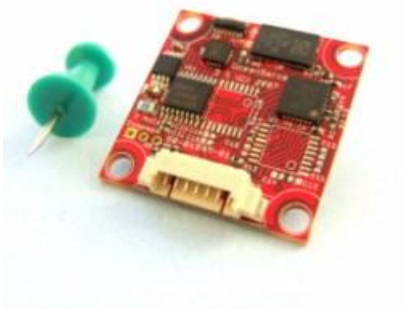
Slika 6 MOOSNavi, metoda Iterate()

Kompas

Još jedan od senzora kojim će biti opremljen autonomni kajak je kompas. Korišten je kompas OceanServer OS5000 koji se sa računalom spaja putem USB priključka. Podaci koje daje kompas se obrađuju i spremaju kako bi bili dostupni na višim razinama.

Karakteristike kompasa:

- Točnost određivanja smjera je
    - 0.5° dok je kompas u ravnini
    - 1° za nagib manji od ± 30°
    - 1.5 za nagib manji od ± 60°
- Kutovi nagiba naprijed - nazad od +/- 90° i kutovi i nagib lijevo – desno od +/- 180°
- Niska potrošnja energije, manje od 30 mA
- RS232 i USB priključak
- Čvrsti dizajn
- Male dimenzije
- Težina ~ 2 g
- Radna temperatura od -40 °C do 80 °C
- Brzina osvježavanja podataka od 40H
- Podesiva brzina očitavanja od 4800 do 115000 (zadana je 19200)



Slika 7 Kompas OceanServer OS 5000

Podešavanje kompasa

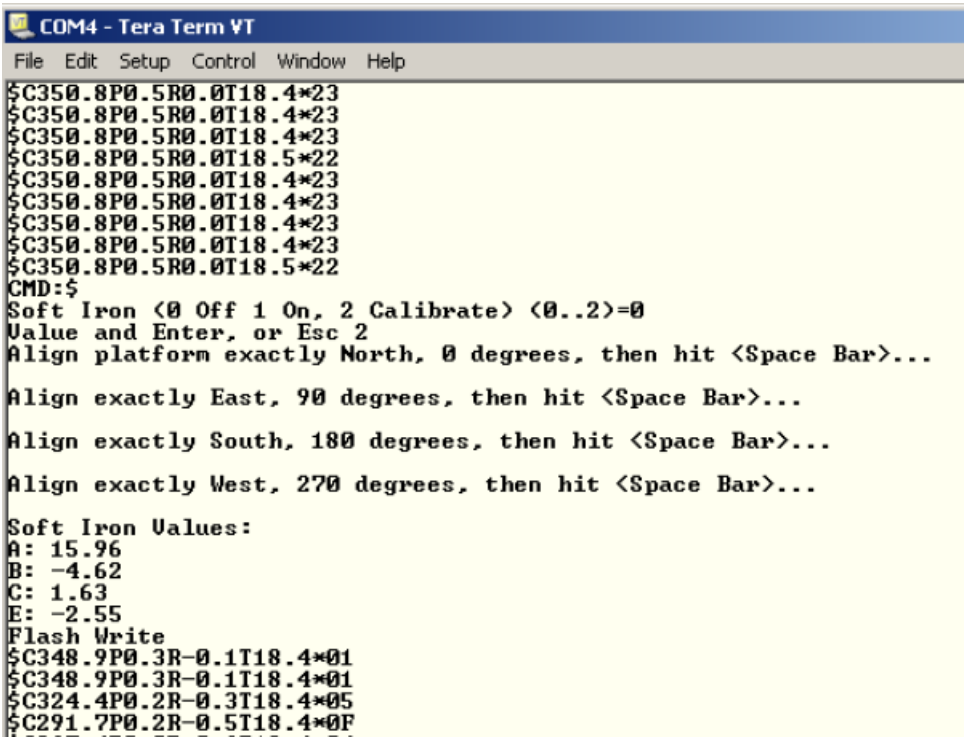
Nakon postavljanja kompasa na autonomni kajak potrebno ga je podesiti, odnosno izvršiti kalibraciju. Na taj ćemo način podesiti kompas i biti sigurni u točnost rezultata. Ovaj kompas podržava dvije vrste kalibracije, tako zvane „Soft iron“ i „Hard iron“ kalibracije.

Nakon što se kompas smjesti na autonomni kajak može se započeti sa procesom kalibracije.

1) Kajak zajedno sa kompasom postaviti u okolinu gdje je malo magnetskog zračenja koje bi moglo ometati kompas
2) Potrebno je postaviti kajak u ravninu i pratiti da podaci koje kompas daje o nagibu (roll i pitch) budu manju od 1°
3) Kalibraciju pokrećemo tako da stisnemo tipku „ESC“ i potom veliko slovo „C“
4) Kompas je potrebno polako okretati i napraviti puni krug, pri tome treba obratiti pažnju da se nagib ne mijenja, odnosno da ne bude veći od 1°
5) Nakon napravljenog punog kruga prekidamo kalibraciju X-Y osi pritiskom razmaknice
6) Sada je potrebno postaviti kompas pod kutom od 90° kako bi se obavila kalibracija Z osi
7) Pritiskom tipke „ESC“ i potom velikog slova „Z“ pokreće se kalibracija Z osi
8) Kompas je ponovno potrebno okrenuti barem jedan puni krug (360°) te potom prekinuti kalibraciju pritiskom razmaknice

Ukoliko tokom kalibracije nagibi (pitch i roll) prijeđu 1° i kalibracija ne uspije, cijeli se postupak može ponoviti.

Slijedi nam „Soft iron“ kalibracija koja nije nužna. Pokreće se pritiskom tipke „ESC“ i potom „$“. Nakon toga potrebno je pratiti uputstva sa ekrana. Prikaz se nalazi na sljedećoj slici.



Slika 8 Soft iron kalibracija

Obrada podataka sa kompasa

Nakon što se kompas spoji sa računalom putem USB-a, može se pokrenuti MOOS klijent koji omogućuje komunikaciju sa kompasom. Kompas šalje poruke u točno određenom formatu. Obradom dobivenih poruka dohvaćamo podatke koji nama trebaju. Svaka poruka koju kompas šalje počinje znakovima $C, nakon toga slijede podaci koji su međusobno odvojeni definiranim graničnicima. Primjer poruke je : $Chhh.hPpp.pRrr.rTtt.t*cc.

Tablica 3 Podaci koje šalje kompas

Oznaka	Pojašnjenje
hhh.h	Nalazi se između graničnika „C“ i „P“ te definira smjer kretanja
pp.p	Nalazi se između graničnika „P“ i „R“ te definira pitch, to jest nagib naprijed-nazad
rr.r	Nalazi se između graničnika „R“ i „T“ te definira roll, to jest nagib lijevo-desno
tt.t	Nalazi se između graničnika „T“ i „*“ te definira temperaturu
cc	Nalazi se na kraju poruke, nakon graničnika „*“ i označava kontrolnu sumu, računa se kao heksadekadska xor suma svih znakova između „$“ i „*“

Podatak koji je nama potreban je smjer kretanja. MOOS klijent ima metodu Iterate koju se nadjača i u njoj se obavlja obrada poruka koje šalje kompas. Jednostavnim parsiranjem dobivenih poruka dohvaća se podatak koji definira smjer kretanja. Taj se podatak sprema i redovito osvježava ovisno o promjenama.

Na sljedećoj se slici nalazi sadržaj metode Iterate() iz klijenta MOOSKompas koji prikazuje način obrade ulaznih podataka.

```
60  bool CSimulator::Iterate() {
61      size_t compassStart;
62      size_t compassEnd;
63
64      string usbData = serial.readString();
65      if (usbData.substr(0,2)=="$C"){
66      compassStart = usbData.find("$C") + 2;
67      compassEnd = usbData.find("P");
68      compassData=usbData.substr(compassStart, compassEnd - compassStart);
69      }
70      m_Comms.Notify("CompassData",compassData,0);
71      cout << usbData << '\n';
72      cout<<"poslano"<<std::endl;
73      return true;
74  }
```

Slika 9 MOOSKompas, metoda Iterate()

Video kamera

Predviđeno je da će autonomni kajak biti opremljen sa jednostavnom kamerom koja je usmjerena na prostor ispred kajaka. Bilo je potrebno uspostaviti prijenos video signala sa kamere koja se nalazi na kajaku i operatera koji sa kopna nadzire rad kajaka. U tu je svrhu korištena jednostavna video kamera tvrtke Logitech, model QuickCam Express. Pri slanju i primanju signala koristi se besplatni program otvorenog koda za prikazivanje i emitiranje multimedijskih sadržaja, VLC.

Slika 10 Korištena video kamera

Postavljanje kamere i slanje video signala

Video kamera ima USB priključak pomoću kojeg se spaja na računalo na kajaku. Nakon što su računalo i kamera povezani, može se pokrenuti slanje video signala. Računalo na kajaku je opremljeno sa vanjskim mrežnim usmjeriteljem. On nam omogućava slanje signala prema operateru na kopnu.

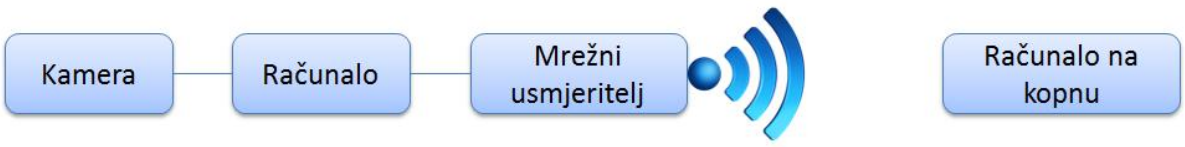
Za sam proces slanja video signala iskorišten je program VLC, koji podržava mrežno slanje video signala. Pri tome je potrebna povezanost na mrežu i poznavanje IP adresa odredišta na koje se signal šalje.

Radi jednostavnijeg pokretanja i uspostave video prijenosa napisane su dvije skripte:

- run.sh
- receive.sh

Prvo se iz terminala sa računala na kajaku pokreće skripta run.sh koja pokreće slanje video signala na unaprijed definiranu IP adresu i poznata vrata. U run.sh skripti moguće je podesiti IP adresu i vrata na koja se šalje video signal, te veličinu slike. U postojećoj skripti postavljena su vrata 5004 i veličina slike 640x480.

Nakon što se video signal sa kajaka počne emitirati, operater za računalom na kopnu pokreće skriptu receive.sh te na taj način prima video signal i prikazuje se na ekranu. U skripti recive.sh potrebno je samo podesiti vrata na kojima se osluškuje signal (5004).



Slika 11 Slanje video signala

Testiranje

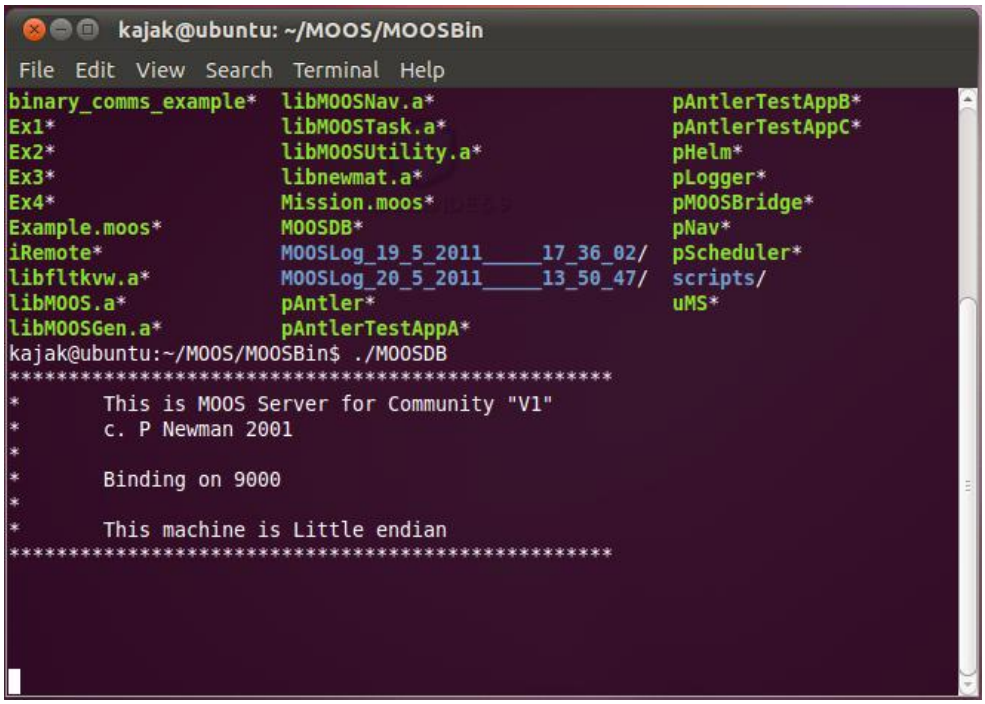
Nakon pojedinačnog postavljanja senzora i obrade njihovih podataka, bilo je potrebno isprobati kako sustav radi u cjelini. Za tu je priliku obavljeno testiranje koje je dokumentirano i opisano.

Sa računalom je potrebno spojiti GPS uređaj, kompas, video kameru i mrežni usmjeritelj te priključiti napajanje. Nakon paljenja računala moguće je započeti testiranje.

Slika 12 Dijelovi sustava

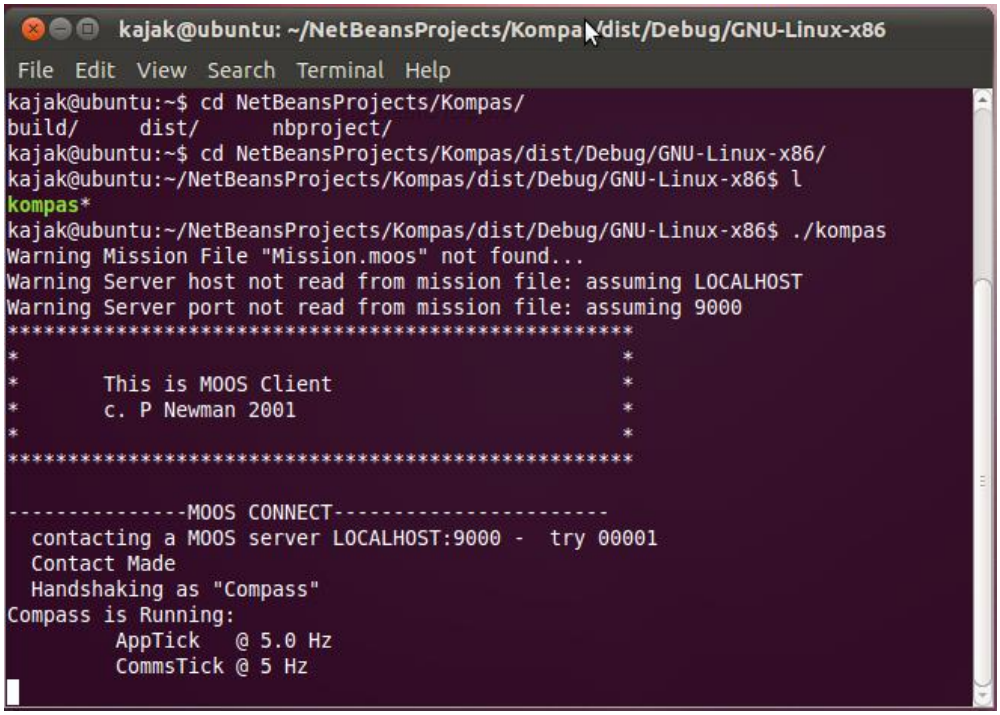
1) računalo
2) napajanje
3) video kamera
4) GPS uređaj
5) kompas
6) mrežni usmjeritelj

Potrebno je prvo pokrenuti poslužitelj, odnosno MOOSDB. Primjer pokretanja nalazi se na sljedećoj slici.

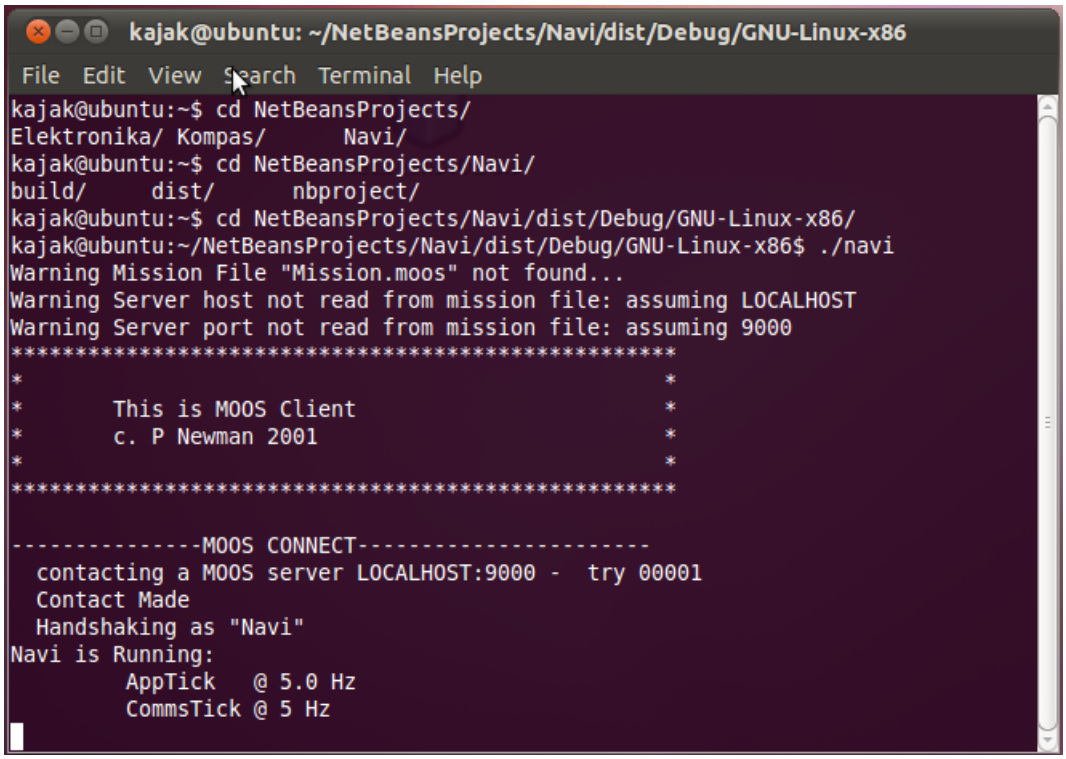


Slika 13 Pokretanje MOOSDB

Sada možemo aktivirati klijente; kompas i GPS uređaj. Postupak se nalazi na sljedećim slikama.

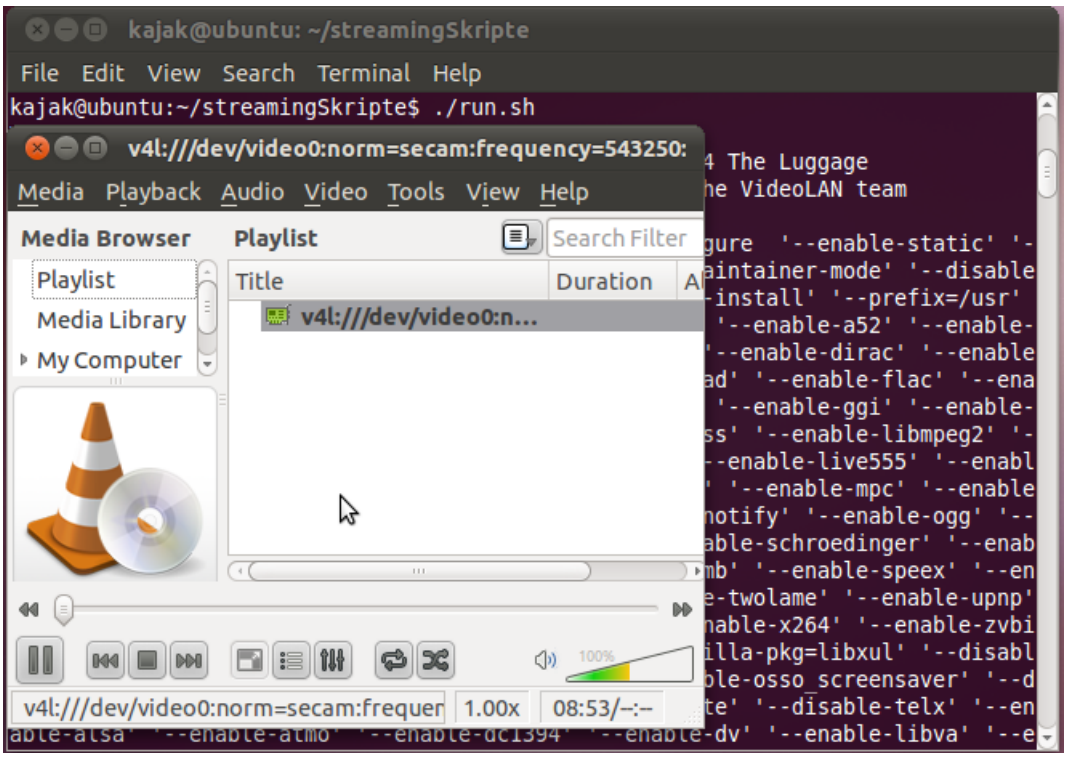


Slika 14 Pokretanje klijenta Compass



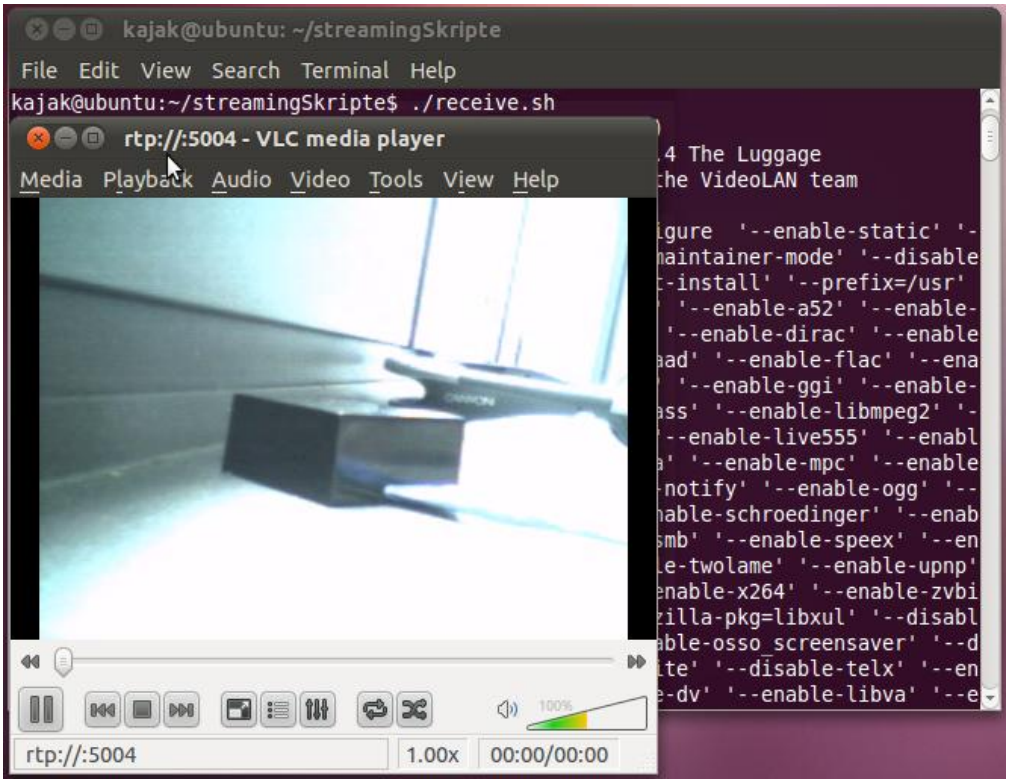
Slika 15 Pokretanje klijenta Navi

Preostalo je još pokrenuti slanje video signala. To radimo pozivajući run.sh skriptu nakon čega nam se otvara sljedeći prozor.



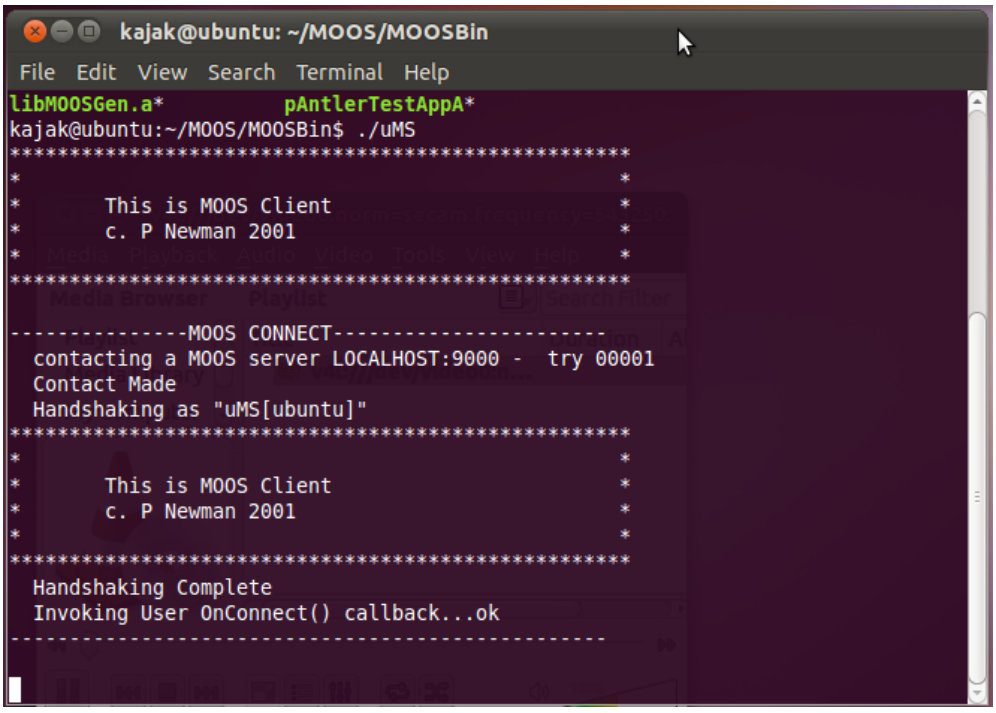
Slika 16 Slanje video signala

Sa udaljenog računala na koji se šalje video signal potrebno je pokrenuti receive.sh skriptu. Na taj način primamo vidio signal koji se šalje i on se prikazuje u posebnom prozoru.



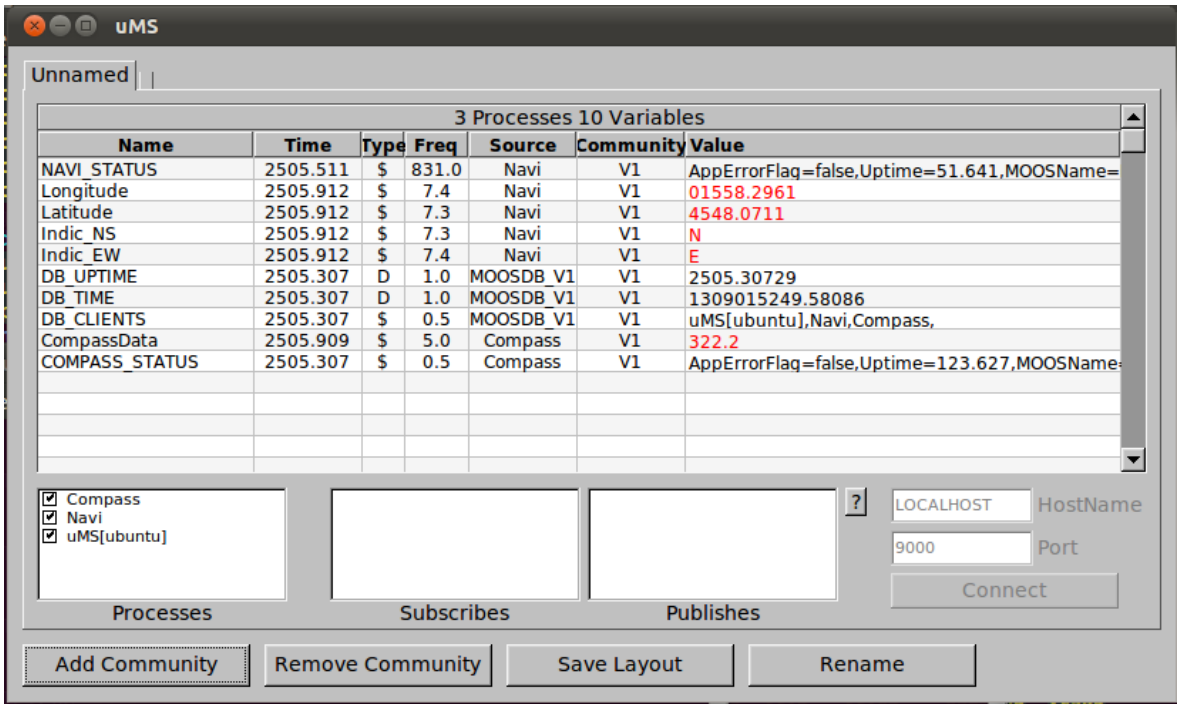
Slika 17 Emitiranje dobivenog video signala

Za detaljniji uvid u rad sustava poslužit ćemo se uMS-om.



Slika 18 Pokretanje uMS-a

Na taj način imamo izravan uvid u aktivnosti svih klijenata i stanja njihovih varijabli.



Slika 19 uMS sučelje sa prikazom podataka

Zaključak

Cilj ovog Završnog rada je bio postaviti osnovne senzore na autonomni kajak i omogućiti slanje video signala. Glavna je ideja bila osposobiti temeljni model koji će kasnije biti lako nadograđivati i poboljšavati. Podatke iz senzora trebalo je obraditi i spremiti kako bi bili lako dostupni na višim razinama upravljanja.

Pri rješavanju tih problema iskorištene se neke gotove aplikacije i sustavi koji su olakšali rješavanje. Sustav MOOS olakšao je spremanje podataka dobivenih od senzora, a mogućnosti VLC-a iskorištene su pri slanju i primanju video signala.

U procesu rješavanja ovog Završnog rada pružena mi je mogućnost za rad u realnom okruženju i sa stvarnim komponentama. Smatram to veoma korisnim i pozitivnim iskustvom. Dok je na papiru sve super i sve radi, kad se rješenje sa papira prenese na sustav, događaju se razne pogreške. Upravo su te pogreške i problemi sastavni dio realnog radnog okruženja. Sa tim će se problemima svaki budući inženjer susresti. Zato je potrebno steći naviku i rutinu u rješavanju usputnih poteškoća na koje se nailazi. Ovaj je Završni rad poslužio i kao izvrsni uvod u upoznavanje problematike i mogućnosti autonomnih sustava.

Tomislav Maras

Literatura

1) Šribar J., Motik B. Demistificirani C++. Zagreb: Elemental, 2008.
2) Jakopec R. C++ programiranije za apsolutne početničke. Varaždin: PRO-MIL, 2006.
3) Paul Newman, Under the Hood of the MOOS Communications API, 17.03.2009. http://www.robots.ox.ac.uk/~pnewman/MOOSDocumentation/CommsArchitecture/latex/CommsArchitecture.pdf
4) Paul Newman, Introduction to programming with MOOS., 04.06.2009. http://www.robots.ox.ac.uk/~pnewman/MOOSDocumentation/ProgrammingWithMOOS/latex/ProgrammingWithMOOS.pdf
5) Prof. dr. sc. Mario Žagar, UNIX i kako ga okoristiti, 2007. http://docbook.rasip.fer.hr/ddb/public/index.php/publication/html/rasipbook/id/1?=&tts=0&css=original
6) Digital Compass User's guide, OS500 series, 2007. http://www.ocean-server.com/download/OS5000_Compass_Manual.pdf
7) Datasheet of GPS samrt antena moduleM LS20030, 25.10.2007. http://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3_datasheet_v1.0.pdf
8) VideoLan Documentation, 18.02.2011. http://wiki.videolan.org/Documentation:Documentation
9) Logitech QuickCam Express Manual, http://www.fixya.com/support/p305666-logitech_quickcam_express_personal_web/manual-19310

Naslov: Integracija senzora na autonomni kajak

Sažetak

Izrada i osmišljavanje autonomnog kajaka je složen i zahtjevan proces. Potrebno je osmisliti dizajn, sustav upravljanja, sustav komunikacije i podesiti potrebne senzore. Kako bi svi podaci sa senzora bili lako dostupni na višim razinama upravljanja potrebno je to dobro povezati i postaviti ispravan sustav komunikacije.

Za svu komunikaciju na autonomnom kajaku iskorišten je sustav MOOS koji djeluje na principu poslužitelja i klijenata. Postoji jedan poslužitelj – MOOSDB na kojem su pohranjeni svi podaci i na njega se spajaju svi ostali klijenti koji na taj način imaju uvid u podatke.

Kompas i GPS služe pri navigaciji i pozicioniranju autonomnog kajaka. Ti su podaci od velike važnosti pri planiranju autonomne misije. Podatke sa kompasa i GPS-a potrebno je obraditi i spremiti u obliku koji su jednostavni i lako dohvatljivi za daljnju upotrebu. Zbog toga su oni integrirani u MOOS sustav.

Jednostavna video kamera omogućuje nam uvid u misiju i bolji doticaj sa samim autonomnim kajakom dok on plovi. Video signal koji se snima potrebno je proslijediti operateru koji je na kopnu kako bi se autonomna misija mogla nadgledati.

Svi podaci koje autonomni kajak prikuplja i generira tijekom svoje misije dostupni su i operateru na kopnu. Pomoću vanjskog mrežnog usmjeritelja koji je smješten na kajaku moguće je povezati se sa kajakom i na taj način imati uvid u trenutno stanje, poziciju, smjer i primati sliku putem video signala.

Ključne riječi: autonomni sustavi, autonomni kajak, senzori, GPS uređaj, kompas, video kamera, prijenos video signala, sustav MOOS

Title: Sensor integration for the autonomous kayak

Abstract

Production and designing of autonomous kayak is a complex and challenging process. It is necessary to develop the design, management system, communication system and to set up the necessary sensors. In order to all the data from the sensor should be easily available at higher levels of management, it is necessary to properly connect the elements and set a proper system of communication.

For all communication on the autonomous kayak is used MOOS system that operates on the principle of servers and clients. There is one server - MOOSDB where all data are stored. The customers who want to have access to the data can easily connect to the server.

Compass and GPS are used in navigation of autonomous kayak. These data are of great importance in the planning of an autonomous mission. Data from compass and GPS must be processed and stored in a format that is simple and easily attainable for future use. This is why they are integrated into the MOOS system.

A simple video camera allows us insight into the mission and better contact with autonomous kayak while he sails. Video signal that is recorded on kayak should be forwarded to the operator on land.

All data collected and generated by autonomous kayak during its mission are available to the operator on the mainland. Using an external network router that is located on the kayak, operator's computer and computer on kayak can be linked. This is how operator can have a look at the current state, position, course and receive a picture via the video signal.

Key words: autonomous systems, autonomous kayak, sensors, GPS, compass, video camera, video streaming, the MOOS system