

Upravljanje mobilnim robotom korištenjem BLE na iOS platformi

Slobodan Lubura, Sumedin Nišić

Elektrotehnički fakultet
Univerzitet u Istočnom Sarajevu
Bosna i Hercegovina

slobodan.lubura@etf.unsa.rs.ba, sume@bih.net.ba

Jasmin Nišić

Fakultet elektrotehnike
Univerzitet u Tuzli
Bosna i Hercegovina

jasmin.nisic@gmail.com

Sadržaj – Tablet računari i smartphone telefoni danas zauzimaju značajno mjesto, kako u privatnoj kućnoj upotrebi, tako i u industrijskim aplikacijama u vidu inteligentnih mobilnih sistema, ali i servisnoj robotici u koju defakto spadaju sve mobilne platforme od mars rovera do robotics cleanera. U kombinaciji sa bluetooth vezom omogućavaju izradu efikasnih sistema bežično upravljanih mobilnih robota. U okviru ovog rada razvijen je ugrađeni mobilni sistem koji se može slobodno kretati u 2D prostoru, baziran na dva podsistema koji međusobno komuniciraju BLE vezom: mobilni robot na Arduino Uno platformi, te iOS aplikacija za upravljanje istim. Kao uređaj za kontrolu iskorišten je pametni telefon Apple kompanije - iPhone 6, a mobilni robot je Parallax Shield Bot.

Ključne riječi – *smartphone; mobilni robot; Arduino; iOS; iPhone; bluetooth; BLE; Parallax Shield-Bot*

I. UVOD

Mobilni roboti predstavljaju automatizovani stroj [1] koji se može kretati kroz zadanu okolinu. To je ustvari mehatronički sistem koji se sastoji iz niza podsistema: mehaničkih, pogonskih, energetskih, upravljačkih, senzorskih, komunikacijskih i sl.

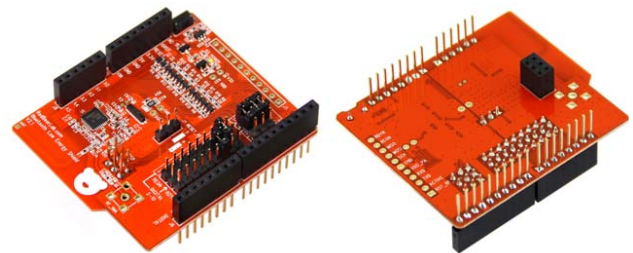
Robotski sistemi dobivaju signale iz radne sredine preko senzora i djeluju na istu pomoću pogona (aktuatora). Veza između opažanja i djelovanja može biti ostvarena jednostavnom obradom signala ili pak može uključivati složene algoritme odlučivanja. Mobilnost se može načiniti upotrebom točkova, nogu, peraja ili drugih pogona [2].

Današnji personalni računari sve više mjesta ustupaju pametnim telefonima i tablet računarima, kako u privatnoj, kućnoj upotrebi, tako i u industrijskim aplikacijama u vidu inteligentnih mobilnih sistema. Ovakvi uređaji u kombinaciji sa Bluetooth vezom niske potrošnje energije (Bluetooth Low Energy - BLE) [3], omogućavaju izradu efikasnih i jeftinih sistema koji nezavisno operiraju i komuniciraju u mobilnom okruženju. Jedna od praktičnih primjena ovakvih sistema jeste primjena u izradi bežično upravljanih mobilnih robota koji se mogu iskoristiti u različitim okruženjima (npr. potrošačka elektronika i daljinski upravljana vozila za zabavu, ali i roboti za operacije na mjestima na kojima ljudima pristup nije moguć ili bezbjedan).

U okviru ovog rada razvijen je ugrađeni mobilni sistem koji se može slobodno kretati u 2D prostoru (ekstenzija na 3D prostor za uređaje kao što su dronovi je moguća uz relativno

jednostavne zahvate na postojećem sistemu). Baziran je na dva podsistema koji međusobno komuniciraju BLE vezom: mobilni robot na Arduino Uno platformi [4], te iOS aplikacija za upravljanje istim. Kao uređaj za kontrolu iskorišten je pametni telefon Apple kompanije - iPhone 6, a mobilni robot je Parallax Shield-Bot [5].

S obzirom na funkcionalne zahtjeve, robot bi trebao doći u bilo koju tačku 2D prostora koristeći dva servo motora, a da sve instrukcije za kretanje robot dobija kao inpute od iPhone aplikacije putem BLE veze. Takođe, bitna stavka je sinhronizacija ova dva podsistema tako da u slučaju prekida Bluetooth veze mobilni robot treba prvi dobiti informaciju da treba zaustaviti svako kretanje. Što se tiče drugih zahtjeva, iPhone aplikacija treba biti energetski efikasno dizajnirana, te mora omogućiti čuvanje energije u toku rada sa BLE čipom, a upravljačka logika aplikacije treba biti striktno odvojena od direktnog pristupa hardveru, kako bi se omogućila fleksibilnost aplikacije, omogućila eventualna portabilnost na druge platforme, te olakšao proces nadogradnje i debugiranja. Takođe, ne bi smjelo biti primjetnog kašnjenja između davanja instrukcija na displeju iPhonea i reakcije upravljanog objekta. Zbog visokog nivoa apstrakcije operativnog sistema [6], koji pokreće iPhone (iOS 8.1) i njegovog visokog nivoa, te same prirode Arduino platforme i Atmel mikrokontrolera, za komunikaciju između ova dva sistema potrebno je instalirati interfejs za BLE komunikaciju, koji može prevesti odgovarajuće instrukcije iz domena jednog podsistema u drugi. Za tu svrhu iskorišten je BLE Shield [3] kompanije RedBearLab (sl. 5).

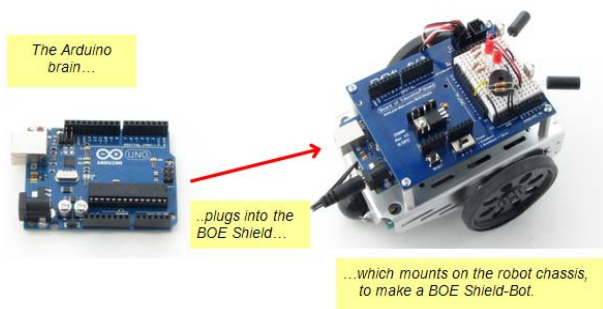


Slika 1. BLE Shield

II. PRINCIP RADA

A. Arduino Uno

Arduino je open-source platforma [4] za kreiranje elektroničkih prototipova bazirana na hardveru i softveru koji je fleksibilan i jednostavan za korištenje. To je ustvari platforma koja predstavlja skup elektroničkih i softverskih komponenti koje se mogu jednostavno povezivati u složenije cjeline s ciljem izrade poučnih elektroničkih sklopova. Osnovu Arduina čini mikrokontroler. Arduino okruženje najčešće koristi 8 bitne mikrokontrolere koje proizvodi tvrtka ATMEL. Najrasprostranjeniji model je ATMEGA328P koji se koristi na osnovnoj Arduino prototipnoj pločici koja je prikazana na sl.1.



Slika 2. Parallax BOE Shield-Bot

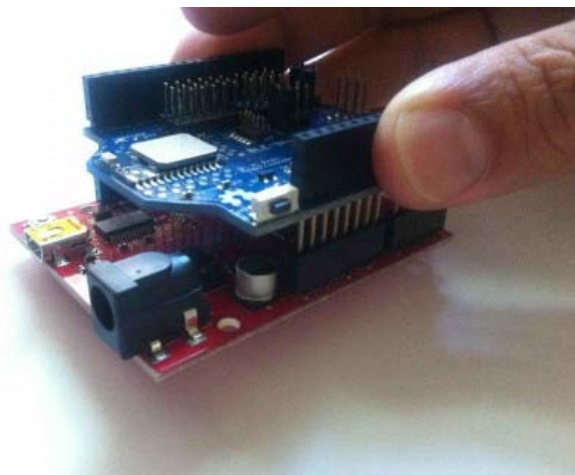
U osnovi, sve Arduino kompatibilne pločice sastoje se od mikrokontrolera, integriranog sklopa za komunikaciju s računarom, te perifernih elektronskih dijelova za osiguravanje mogućnosti rada mikrokontrolera – stabilizatori napona, kvarcni oscilator za generisanje frekvencije takta i slično.

B. BLE (Bluetooth Low Energy)

Tradicionalna Bluetooth veza je "connection oriented", što znači da nakon prvog uspostavljanja veze, ona biva održavana cijelo vrijeme do njenog prekida, čak i ako nema protoka podataka putem iste, što značajno troši energetske resurse (baterije u slučaju mobilnih uređaja). BLE (*Bluetooth Low Energy*) uveden je 2011. godine [3] i predstavlja znatno unaprijeđen radio prenos u odnosu na tradicionalni bluetooth, prvenstveno sa aspekta potrošnje energije, a u zadnje vrijeme i brzine prenosa podataka. Bluetooth uređaji niske potrošnje energije najveći dio vremena funkcionišu u *sleep* režimu i uređaji se bude samo prilikom eksplicitnog iniciranja veze, tj. prenosa podataka. Ovakav način rada omogućava neometano funkcionisanje uređaja 5 do 10 godina sa običnom CR2032 baterijom. Funkcionalno, BLE je zadržao osnovne principe tradicionalnog bluetootha (AFH - *adaptive frequency hopping*, sigurnost i enkripciju, te autentifikaciju korisnika), što ga čini fleksibilnim i jednostavnim za ugradnju u sisteme koji su ranije funkcionisali sa starijim generacijama Bluetootha. BLE tehnologija je prvi put u pametnim telefonima iskorištena 2011. godine u uređaju iPhone 4S Apple kompanije, što i predstavlja minimalan hardverski zahtjev podsistema za upravljanje realizovanog u okviru ovog rada.

C. BLE Shield

BLE (Bluetooth Low Energy) Shield je dizajniran od strane kompanije RedBearLab [3] za upotrebu sa Arduino pločama. (Arduino Uno, Mega 2560, Leonardo, Due) i omogućava povezivanje ovih ploča sa BLE centralnim uređajima (što mogu biti tableti ili pametni telefoni), te razvoj iOS i Android aplikacija za komunikaciju sa Arduino sistemima. U svojoj osnovnoj konfiguraciji ima integriranu PCB antenu, ali pruža mogućnost korištenja eksterne antene preko SMA konektora. U potpunosti implementira Bluetooth® v4.0 standard i sadrži integrirane kola za povezivost sa niskom potrošnjom energije, te implementiran kompletan sloj veze (*link layer*) OSI referentnog modela. Funkcioniše u kombinaciji sa iPhoneom 4S i novijim, te zahtijeva komunikaciju sa iOS verzijama 6.0 ili novijim.



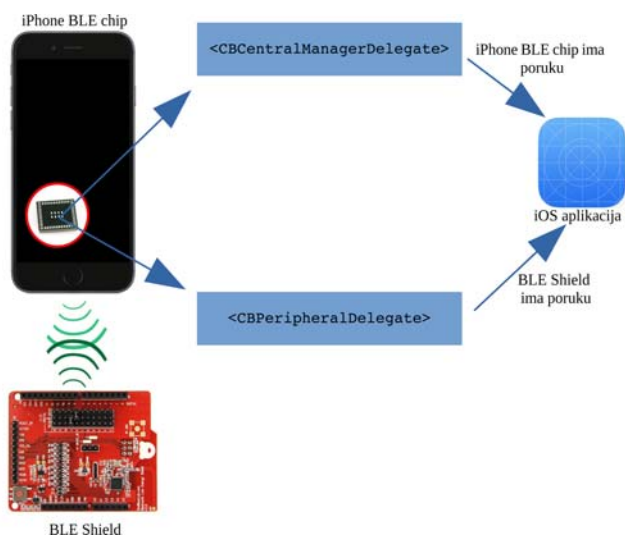
Slika 3. Spajanje BLE Shield na Arduino Uno

III. IOS APLIKACIJA

Funkcionalnost aplikacije za upravljanje robotom se može podijeliti na tri dijela: upravljanje Bluetooth konekcijom, upravljanje korisničkim interfejsom i reagovanje na korisničke komande, te upravljanje listom uređaja. Središnji dio aplikacije se zasniva na funkcijama *CoreBluetooth* frameworka. To je skup biblioteka iOS i Mac OS X operativnih sistema [6] za komunikaciju sa bluetooth uređajima niske potrošnje energije i principijelno predstavlja apstrakciju Bluetooth 4.0 standarda za BLE. To znači da sakriva velik broj detalja niskog nivoa od developera i olakšava interakciju sa BLE uređajima. U BLE komunikaciji generalno postoje dvije osnovne uloge: centralna i periferna, što se može analogno povezati sa tradicionalnim klijent-server odnosom (gdje bi centralni uređaj u BLE sistemu činio neki vid servera u server-klijent komunikaciji). Periferni uređaj prikuplja podatke iz okoline, a centralni te podatke obrađuje. U kontekstu iOS aplikacije, centralni uređaj je predstavljen objektom klase *CBCentralManager*, a periferni *CHPeripheral*, te za upravljanje ovim objektima iz

realizovane aplikacije neophodno je iskoristiti Objective-C *delegate* klase iz pomenutog frameworka: `CBCentralManagerDelegate` i `CBPeripheralDelegate`. Navedeni delegati sadržavaju metode koji se pozivaju na tačno određene događaje (tzv. *event-driven* metodi) i generalno služe za izvještavanje aplikacije o promjenama koje se dešavaju na BLE uređajima u okolini ili BLE čipu unutar samog telefona.

Metod `centralManagerDidUpdateState` iz `CBCentralManagerDelegate` klase obavještava aplikaciju o promjenama stanja bluetooth hardvera telefona i iskorišten je za pravilno prekidanje ili uspostavljanje konekcije sa mobilnim robotom, odnosno skeniranje uređaja pri svakom uključanju bluetooth veze. U slučaju pronalaska perifernog uređaja u blizini, on BLE čipu telefona odgovara određenom sekvencom podataka koji se aplikaciji šalju u okviru parametara metoda `didDiscoverPeripheral`. Između ostalog, šalje se i UUID uređaja koji ga jedinstveno određuje i na osnovu kojeg aplikacija može upravljati tačno određenim uređajem u svakom trenutku, ukoliko ih više bude pronađeno. Takođe, aplikacije dobije i informacije o karakteristikama pronađenog uređaja, tj. adresi RX/TX pinova za serijsku komunikaciju. Podaci upisani na TX pin na strani BLE Shield modula će iOS aplikaciji biti dostupni na RX pinu i suprotno - aplikacija BLE Shieldu podatke šalje na TX pin, dok ih on čita sa RX pina. Maksimalna količina podataka koja se može poslati odjednom je 20 bajta zbog BLE112 restrikcija. Za slanje podataka koristi se metod `writeValue` u čijem se pozivu specificiraju podaci koji se šalju, te adresa na koju se šalje.



Slika 4. Princip rada iOS aplikacije

IV. ARDUINO APLIKACIJA

Servo motor koji se kontroliše putem iOS aplikacije čini osnov kretanja mobilnog robota kojim se upravlja. Servo motor je motor sa promjenljivom rotacionom pozicijom i

kontroliše se slanjem električnih impulsa različitom frekvencijom na njegovo kontrolnu liniju. iOS aplikacija vrši izračun dužine trajanja impulsa na osnovu zadanih parametara od strane korisnika, te vrijednosti iz opsega 1.3 ms do 1.6 ms šalje Arduino aplikaciji, koja dalje konfigurise servo motore sa tim podacima. Dužina trajanja impulsa od 1.3 ms za korištene servo motore znači kretanje punom brzinom u smjeru kazaljke na satu, a impuls dužine 1.7 ms znači kretanje punom brzinom u smjeru suprotnom od kretanja kazaljke na satu, dok impulsi od 1.5 ms na upravljačkoj liniji servo motora znače instrukciju za njegovo mirovanje [4]. Globalne varijable i predprocesorske direktive iskorištene za konfiguraciju servo motora su:

```
#include <Servo.h>
```

```
int LED = 13;
Servo servoLeft;
Servo servoRight;
```

Uključivanje *Servo.h* biblioteke omogućava jednostavan interfejs za konfiguraciju i upravljanje servo motorima. Funkcija `void setup()` vrši inicijalno podešavanje Arduino podsistema:

```
void setup()
{
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);

    servoLeft.attach(8);
    servoRight.attach(9);
    // Povezivanje objekata servo motora sa
    // pinovima 8 i 9 na ploči
    servoLeft.writeMicroseconds(1500);
    servoRight.writeMicroseconds(1500);
    // Inicijalizacija početnih brzina servo
    // motora
}
```

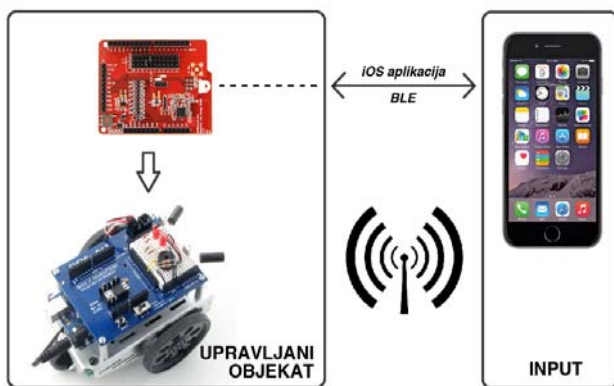
Uključivanje biblioteke *SoftwareSerial.h* omogućava upotrebu biblioteke za serijsku komunikaciju, a prije nego Arduino i BLE modul mogu početi sa razmjenom podataka potrebno je inicijalizirati *SoftwareSerial* objekat sa adresama RX i TX pinova, kao i podrazumijevani *baud rate* (koji je 9600 za BLE Shield module). Glavninu koda za upravljanje motorom, tj. za čitanje vrijednosti poslanih od strane iOS aplikacije, te prenesene putem BLE Shield modula se koristi kod u okviru `void loop()` funkcije, a svodi se na provjeru dostupnosti podataka, te u potvrđnom slučaju slanje primljenih podataka na kontrolne linije servo motora. Metod `available()` vraća broj vrijednosti koje se trenutno nalaze u *bufferu* BLE Shield uređaja, a metod `read()` vrši čitanje zadnje vrijednosti iz *buffera*, nakon čega vrši njeno brisanje iz istog (što je brza i jednostavna operacija s obzirom da je *buffer* implementiran kao *stack*).

```

if (BLE_Shield.available()==2) {
  servoLeft.
    writeMicroseconds(BLE_Shield.read());
  servoRight.
    writeMicroseconds(BLE_Shield.read());
}

```

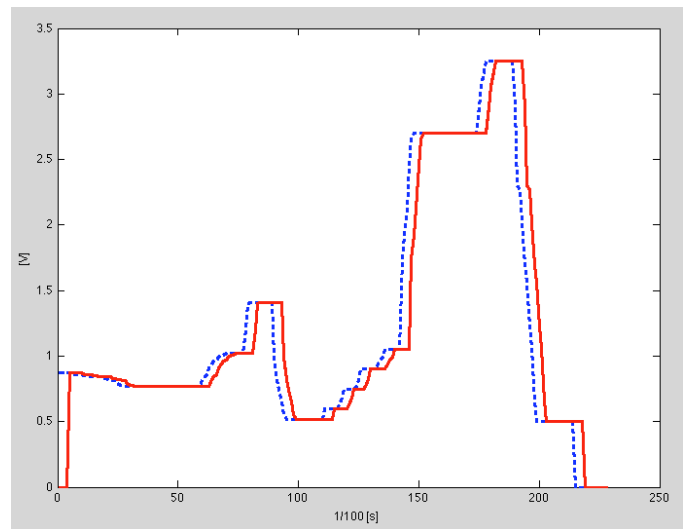
Cjelokupne proračune pozicije na osnovu *UISlider* kontrola iz *CocoaTouch* biblioteke iskorištene pri kreiranju iOS aplikacije vrši upravo iOS aplikacija, te mobilnom robotu dostavlja konkretne vrijednosti pozicije koje se samo trebaju primijeniti na kontrolisane servo motore. Moguća je i suprotna implementacija – da Arduino aplikacija vrši normalizaciju i proračun vrijednosti na osnovu dobijenih pozicija slidera, ali to nije preporučljivo iz razloga što se taj proces mnogo brže odvija na iPhoneu zbog znatno većih CPU resursa takvog podsistema u odnosu na Arduino sa Atmel mikrokontrolerom.



Slika 5. Princip rada sistema

V. ANALIZA ODZIVA

Kako je jedan od funkcionalnih zahtjeva postavljenih pri izradi ovog sistema bilo i što manje vrijeme odziva robota na zadane instrukcije, izvršeno je snimanje zadanih vrijednosti zajedno sa vremenskim otiskom na predajnoj strani, kao i na prijemnoj kako bi se imao uvid u realno transportno kašnjenje. Slika 6 grafički prikazuje to kašnjenje koje u realnim uslovima i udaljenostima između predajnika i prijemnika iznosi između 20 i 40 ms.



Slika 6. Vrijeme odziva robota

VI. ZAKLJUČAK

U okviru ovog rada razvijen je ugrađeni mobilni sistem koji se može slobodno kretati u 2D prostoru, baziran na dva podsistema koji međusobno komuniciraju BLE vezom: mobilni robot na Arduino Uno platformi, te iOS aplikacija za upravljanje istim. Kao uređaj za kontrolu iskorišten je pametni telefon Apple kompanije - iPhone 6, a mobilni robot je Parallax Shield Bot.

Pogon robota ostvaren je korištenjem dva upravljiva servo motora. Servo motor koji se kontroliše putem iOS aplikacije čini osnov kretanja mobilnog robota kojim se upravlja. Instrukcije za kretanje robot dobija kao inpute od iPhone aplikacije putem BLE veze, slanjem električnih impulsa različitim frekvencijom na kontrolnu liniju servo motora. U slučaju prekida Bluetooth veze mobilni robot dobiva informaciju da treba zaustaviti svako kretanje do ponovne uspostave veze.

Za eventualnu modifikaciju mjernog sistema moguće je ići u pravcu izrade softverskih rješenja koja, naravno uvažavajući i daljnji razvoj hardvera mobilnih uređaja, će omogućiti bolje čuvanje energije u toku rada sa BLE čipom, kraći vremenski odziv robota na zadani input, te širenje aplikacije i na druge operativne sisteme.

LITERATURA

- [1] Jasmin Velagić, Sarajevo, 2012. Mobilna robotika,
- [2] Z.Kovačić, Zagreb, 2008. Praktikum robotike
- [3] J. ReadBear Lab official documentation: Getting started with BLE Shield (www.redbearlab.com)
- [4] John Baichtal, Arduino for Beginners: Essential Skills Every Maker Needs, Amazon, 2013.
- [5] <http://learn.parallax.com/ShieldRobot>
- [6] Alasdair Allan, iOS Sensor Apps with Arduino, O'Reilly Media, septembar 2011

ABSTRACT

**MOBILE ROBOT CONTROL USING BLE ON iOS
PLATFORM**

Sumedin Nisic, Slobodan Lubura, Jasmin Nisic

Tablet computers and smartphones today have an important role in everyday home use, as well as in industrial applications in the form of intelligent mobile systems. Combined with bluetooth connectivity they allow us, inter alia, to build effective systems of wirelessly controlled mobile robots. This paper describes embedded mobile system that can move freely in 2D space, based on two subsystems that communicate with each other using BLE connection: one system is a mobile robot based on an Arduino Uno and another one is iOS application for its control and management. Apple iPhone 6 has been used as a control device, and controlled object was Parallax Shield-Bot mobile robot.