

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Mateja Tošić, 0038/2020

GPS lokator sa funkcijama merenja brzine,
pređenog puta i vremena

projekat iz predmeta Principi modernih telekomunikacija

mentor:
prof. dr Milan Bjelica

Beograd, jun 2023

Sažetak

Cilj ovog projekta je izrada uređaja koji ima mogućnost određivanja GPS koordinata, merenja brzine, pređenog puta i vremena korišćenjem Arduino UNO mikrokontrolera.

Ključne reči: Arduino, GPS, debouncing, mikrokontroler, brzina, softverski UART, paralelni interfejs, prekid

Sadržaj

1	Uvod	3
2	Povezivanje modula	4
2.1	Šematski prikaz komponenti	4
2.2	Komunikacija između modula	5
2.2.1	Komunikacija između Arduina i GPS-a	5
2.2.2	Komunikacija između Arduina i LCD-a	6
2.2.3	Komunikacija između Arduina i tastera	6
3	Softverska implementacija	7
4	Prikaz karakterističnih situacija	9
5	Način komunikacije korisnika sa uređajem	11
6	Zaključak	12
	Literatura	13
	Prilozi	14

Spisak slika

2.1	<i>Šematski prikaz.</i>	5
3.1	<i>Dijagram toka.</i>	7
4.1	<i>Karakteristične situacije.</i>	9

Glava 1

Uvod

Cilj ovog projekta je izrada GPS uređaja uz pomoć Arduino UNO mikrokontrolera. Uređaj je u stanju da na osnovu GPS modula dobije informacije, parsira ih i predstavi ih u obliku (geografska širina, geografska dužina). Pored toga, sam uređaj na osnovu GPS lokacije ima mogućnost merenja brzine kretanja uređaja kao i izračunavanje pređenog puta i trajanja kretanja.

Glava 2

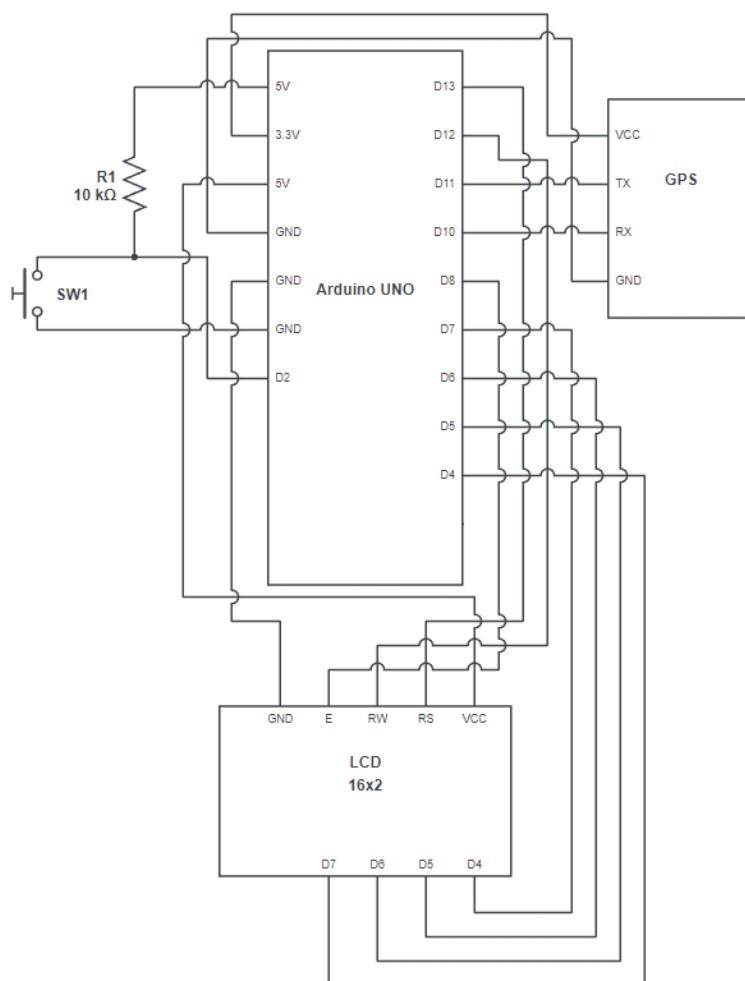
Povezivanje modula

2.1 Šematski prikaz komponenti

Šematski prikaz povezivanja dat je na slici 2.1. Pri povezivanju modula, veoma je bitno da se taster(SW1) poveže na pinu D2 (alternativno na pinu D3) kako bi Arduino UNO bio u mogućnosti da generše prekide izazvane pritiskom na taster. Alternativno ukoliko bi se koristio drugi Arduino mikrokontroler, pinovi koji omogućuju prekide mogu se naći u dokumentaciji Arduino mikrokontrolera. LCD ekran povezan je sa Arduinoom koristeći paralelni interfejs na pinovima D4,D5,D6,D7. GPS modul povezan je sa Arduinoom koristeći softverski UART(*Universal Asynchronous Receiver-Transmitter*) na pinovima D11 i D10 koji je na odgovarajući način implementiran Arduino bibliotekom SoftwareSerial.

Za sam uređaj, potrebne su sledeće komponente:

1. Arduino UNO(1 kom.)
2. GPS-NEO-6M(1 kom.)
3. LCD 16x2(1 kom.)
4. Taster(1 kom.)
5. Otpornik $10k\Omega$ (1 kom.)
6. Kratkospojne žice(16 kom.)



Slika 2.1: Šematski prikaz.

2.2 Komunikacija između modula

2.2.1 Komunikacija između Arduina i GPS-a

GPS modul i Arduino mikrokontroler komuniciraju preko softvertskog UART-a. Arduino preko pina D11 neprestano prima informacije koje dolaze sa linije TX GPS modula. Koristeći Arduino biblioteku TinyGPSPlus vrši se parsiranje ulaznih informacija i dobijanje informacije o geografskoj širini i geografskoj dužini. Kasnije, te informacije koriste se za računanje pređenog puta i brzine. Takođe biblioteka TinyGPSPlus obezbeđuje blokirajući poziv funkcije koja prima GPS informacije sve dok GPS modul ne dobije signal za-

dovoljavajućeg kvaliteta. Tek nakon toga biblioteka(funkcija) vrši parsiranje pristiglih informacija.

2.2.2 Komunikacija između Arduina i LCD-a

LCD ekran i Arduino mikrokontroler komuniciraju preko paralelnog interfejsa. Koristeći digitalne pinove D7, D6, D5, D4, šalju se informacije koje treba da budu ispisane na sam LCD ekran. Radi lakše komunikacije mikrokontrolera i LCD ekrana koristi se Arduino biblioteka LiquidCrystal. Funkcijom *write()* te biblioteke vrši se ispis teksta.

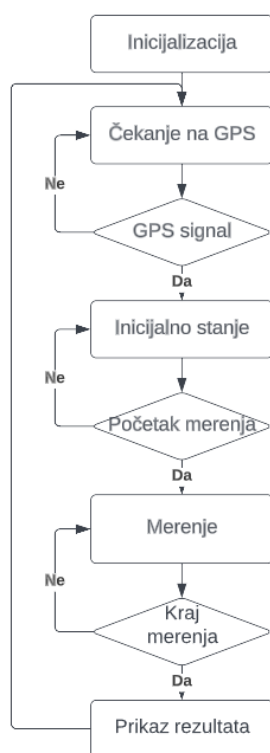
2.2.3 Komunikacija između Arduina i tastera

Kako Arduino UNO mikrokontroler ima 2 digitalna pina preko kojih je moguće primiti spoljašnji prekid(D2,D3), taster se mora povezati na neki od njih. Pritiskom na taster po liniji D2 Arduinu stiže prekid koji se obrađuje u prekidnoj rutini. Prekidna rutina obrađuje prekid na osnovu stanja u kome se tada uređaj nalazi i biće detaljnije objašnjena kasnije.

Glava 3

Softverska implementacija

U ovom odeljku biće objašnjene glavne funkcionalne celine koda koji se nalazi na mikrokontroleru. Ceo kod dat je u prilogu. Dijagram toka dat je na slici 3.1.



Slika 3.1: *Dijagram toka.*

Blok **inicijalizacija** pretstavljen je funkcijom *setup()* u kojoj se inicijalizuje komunikacija preko softverskog UART-a, inicijalizuje biblioteka koja upravlja LCD ekranom, a zatim se za digitalni pin D2 vezuje prekidna rutina koja je napisana funkcijom *taster()*.

Deo dijagrama koji obuhvata blok **čekanje na GPS i proveru GPS signala** određen je funkcijom *obradiGpsSignal()*. Ova funkcija u sebi sadrži blokirajući poziv funkcije za prijem informacije od GPS-a iz biblioteke TinyGPSPlus. Prema tome, poziv same funkcije *obradiGpsSignal()* je takođe blokirajući sve dok se ne dobije zadovoljavajuća GPS informacija (ona iz koje je moguće dobiti podatke o geografskoj širini i geografskoj dužini).

Inicijalno stanje predstavlja završetak funkcije *setup()* koja u sebi sadrži ispis geografske dužine i geografske širine na LCD ekranu. Uređaj ostaje u inicijalnom stanju sve dok se ne okine prvi prekid uz pomoć tastera.

U bloku **merenje** ulazi se nakon okidanja prekida uz pomoć tastera. Nakon prekidne rutine, poziva se funkcija *stanjeMerenja()* u kojem se uređaj nalazi sve dok ga korisnik ne prekine drugim prekidom, odnosno drugim pritiskom tastera. U funkciji *stanjeMerenja()* vrši se računanje pređene distance uz pomoć funkcije *distanceBetween()* biblioteke TinyGPSPlus. Funkcija *distanceBetween()* na osnovu dve zadate tačke određene geografskom dužinom i geografskom širinom vraća razdaljinu između njih u metrima. Nakon okidanja drugog prekida pomoću tastera računa se proteklo vreme i brzina, zatim se funkcijom *ispisiRezultate()* prikazuje vreme, brzina i pređena razdaljina na LCD ekranu.

Nakon završetka merenja, uređaj ostaje u stanju **prikaz rezultata**. U tom stanju prikazuju se rezultati prethodnog merenja. Okidanjem trećeg prekida, uređaj se vraća u inicijalno stanje u kojem se na LCD ekranu ispisuju trenutne GPS koordinate.

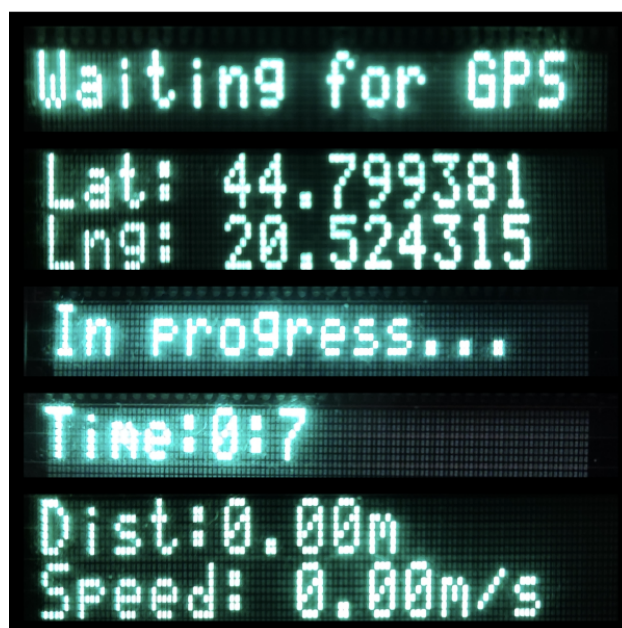
Prekidna rutina koja je određena funkcijom *taster()* poziva se svaki put kada se pritisne taster, tj okine prekid. U njoj se na osnovu flag-ova, uređaj prebacuje u određeno stanje. Zbog nesavršenosti tastera, dešava se da se jednim pritiskom na taster, zbog prisustva šuma koji se tom prilikom stvara, prekid okine više puta. Ovaj problem (debouncing) rešen je tako što se pri prvom ulasku u prekidnu rutinu započne merenje vremena. Ukoliko se zbog postojanja šuma desi da se prekid okine još nekoliko puta od trenutka prve pojave prekida(*t*) do trenutka *t*+200ms, ovi prekidi neće biti prihvaceni, već će biti tretirani kao nevalidni, odnosno smatraće se da su nastali kao posledica šuma tastera. Interval od 200ms određen je eksperimentalno.

U glavnoj petlji programa *loop()*, se primenom tehnike polling-a osluškiju, odnosno proveravaju ispunjeni uslovi i na osnovu njih se prelazi u određena stanja.

Glava 4

Prikaz karakterističnih situacija

Na slici 4.1 prikazane su četiri moguće karakteristične situacije koje se mogu naći na LCD ekranu uređaja.



Slika 4.1: Karakteristične situacije.

Prva situacija je ispis *Waiting for GPS* na LCD ekranu. Ova situacija prikazuje da je uređaj u stanju čekanja na GPS signal. Po dobijanju kvalitetnog GPS signala ovaj ispis sa ekran nestaje.

Druga karakteristična situacija je ispis podataka o geografskoj širini i geografskoj dužini *Lat:... Long:...*. U ovoj situaciji se uređaj nalazi po dobijanju GPS signala i u njemu ostaje dok se ne otpočne proces merenja.

Treća karakteristična situacija je ispis *In progress....* Ovaj ispis označava da se uređaj trenutno nalazi u procesu merenja. U njega se ulazi pritiskom na taster, kojim se otpočinje proces merenja, a iz njega se izlazi ponovnim pritiskom na taster.

Četvrta karakteristična situacija je ispis *Time ...*, a zatim i ispis *Dist:... Speed:...,*. U ovoj situaciji se uređaj nalazi nakon završetka procesa merenja. Najpre se u intervalu od tri sekunde na ekranu prikazuje dužina trajanja merenja *Time:...*, a nakon toga se na LCD displeju prikazuju podaci o pređenom putu i vremenu *Dist:... Speed:....* U ovoj situaciji uređaj ostaje sve dok korisnik ne pritisne taster i vrati uređaj u karakterističnu situaciju broj 2.

Glava 5

Način komunikacije korisnika sa uređajem

Nakon pokretanja uređaja na ekranu prikazaće se poruka *Waiting for GPS* koja označava da GPS modul nije još dobio odgovarajući signal. Nakon što modul dobije odgovarajući signal na ekranu prikazaće se trenutna geografska širina i geografska dužina. U ovom trenutku korisnik je u mogućnosti da interaguje sa uređajem uz pomoć tastera. Pritiskom na taster pokreneće se proces merenja, na ekranu će se ispisati poruka *In progress....* U ovom trenutku uređaj se nalazi u stanju merenja i iz njega izlazi onog trenutka kada korisnik želi da završi merenje. To može učiniti ponovnim pritiskom na taster. Nakon toga na ekranu prikazuju se vreme u formatu mm:ss, gde mm označava protekle minute od početka merenja, a ss sekunde, zatim prikazuje se pređena razdaljina izražena u metrima i na kraju brzina. Nakon prikazivanja rezultata korisnik je u mogućnosti da ponovnim pritiskom na taster resetuje uređaj. Samim tim uređaj se vraća na inicijalno stanje u kome se na ekranu prikazuju geografska širina i geografska dužina.

Glava 6

Zaključak

U ovom projektu prikazan je proces kreiranja uređaja koji se sastoji iz više modula. Prikazan je način povezivanja modula preko odgovarajućih interfejsa kao i način komunikacije između tih modula. Takođe dat je i softverski deo implementacije, u kojoj je prikazan način obrade GPS signala, računanje razdaljine između tačaka koje su opisane geografskom širinom i geografskom dužinom i prikaz rezultata na LCD ekranu. Softverska implementacija dodatno je uključivala i softversko rešavanje problema šuma koji se javlja pri pritisku na taster *eng. debouncing* koji je rešen uz pomoć tajmera.

Literatura

- [1] Ofcijalni sajt Arduina:
<https://www.arduino.cc/>
- [2] Dokumentacija GPS modula:
<https://datasheetspdf.com/pdf/866235/u-blox/NEO-6M/1>
- [3] Dokumentacija LCD ekrana:
<https://datasheetspdf.com/datasheet/GU112x16G-7806A.html>

Prilozi

Programski kod koji se nalazi na mikrokontroleru dat je u prilogu:

```
#include <TinyGPS++.h>
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

volatile float gSirina = 0;
volatile float gDuzina = 0;
const int BUTTON_PIN = 7;
boolean pocetak = false;
boolean kraj=false;
boolean uToku=false;
unsigned long vreme = 0;
volatile float rastojanje = 0;

LiquidCrystal lcd(13, 12, 8, 7, 6, 5, 4);
TinyGPSPlus gps;
SoftwareSerial sUart(11, 10);

void setup()
{
  Serial.begin(9600);
  sUart.begin(9600);
  lcd.begin(40, 2);
  pinMode(2,INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(2), taster, RISING);
  lcd.print("Waiting for GPS");
  obradiGpsSignal();
  ispisiKoordinate();
}
```



```

void taster() {
    static unsigned long poslPrekid=0;
    unsigned long trenPrekid=millis();
    if(trenPrekid-poslPrekid>200){
        if(!pocetak and !uToku){
            pocetak=true;
        }else if(uToku and pocetak){
            uToku=false;
            pocetak=false;
        }
        else{
            kraj=true;
            uToku=false;
        }
    }
    poslPrekid=trenPrekid;
}

```

```

void obradiGpsSignal() {
    while (1) {
        sUart.listen();
        while (sUart.available() > 0) {
            gps.encode(sUart.read());
        }
        if (gps.location.isUpdated()) {
            gDuzina = gps.location.lng();
            gSirina = gps.location.lat();
            break;
        }
    }
}

```

```

void ispisiKoordinate() {
    lcd.clear();
    lcd.print("Lat: ");
    String gS = String(gSirina, 6);
    lcd.print(gS);
    lcd.setCursor(0, 1);
    lcd.print("Lng: ");
    String gD = String(gDuzina, 6);
}

```

```

    lcd.print(gD);
}

void ispisiRezultate(float brzina, unsigned long interval, float rastojanje) {
    lcd.clear();
    unsigned long seconds = interval/ 1000;
    unsigned long minutes = seconds / 60;
    lcd.print("Time:");
    lcd.print(minutes);
    lcd.print(":");
    lcd.print(seconds);
    delay(3000);
    lcd.clear();
    lcd.print("Dist:");
    String rst = String(rastojanje, 2);
    lcd.print(rst);
    lcd.print("m");
    lcd.setCursor(0,1);
    lcd.print("Speed: ");
    String v=String(brzina,2);
    lcd.print(v);
    lcd.print("m/s");
}

void stanjeMerenja(){
    while(1){
        if(!uToku){
            unsigned long interval=millis()-vreme;
            unsigned long seconds = interval/ 1000;
            float brzina= (rastojanje*1.0)/seconds;
            ispisiRezultate(brzina,interval,rastojanje);
            uToku=true;
            return;
        }
        obradiGpsSignal();
        float oldlat=gSirina;
        float oldlong=gDuzina;
        delay(1000);
        obradiGpsSignal();
        double pom=TinyGPSPlus().distanceBetween(oldlat,oldlong,gSirina,gDuzina);
        //zbog greske koju pravi GPS pri mirovanju uvodi se ovo ogranicenje
    }
}

```

```

    if(pom>1){
        rastojanje+=pom;
    }
}

void loop()
{
    if(pocetak and !uToku){
        uToku=true;
        vreme=millis();
        rastojanje=0;
        lcd.clear();
        lcd.print("In progress...");
        stanjeMerenja();
    }
    if(kraj){
        kraj=false;
        obradiGpsSignal();
        ispisiKoordinate();
    }
}

```