Univerzitet "Džemal Bijedić"

Fakultet Informacijskih Tehnologija

Završni rad

Samobalansirajući robot

elektronički bazirani projekat za održavanje robota na dva točka uspravnim

Profesor:
Doc.dr. Elmir Babović

Student:
Adi Šoše, IB160034

Akademska godina: 2018/2019

Mostar, 2019. godina

IZJAVA O AUTORSTVU

Ja, **ADI (AMIR) ŠOŠE**, student Fakulteta informacijskih tehnologija, Univerziteta "Džemal Bijedić" u Mostaru, pod punom moralnom, materijalnom i krivičnom odgovornošću,

Izjavljujem

da je rad pod naslovom

SAMOBALANSIRAJUĆI ROBOT

u potpunosti rezultat sopstvenog istraživanja, gdje su korišteni sadržaji (tekst, ilustracije, tabele itd.) drugih autora jasno označeni pozivanjem na izvor i ne narušavaju bilo čija vlasnička ili autorska prava.

U Mostaru, 17.09.2019.

Adi Šoše, IB160034

Sadržaj

1 Sazetak

2 Uvod

3 Arduino

Arduino je elektronički bazirana platforma otvorenog koda[1]. Radi se o ploči koja na sebi najčešće ima Atmel-ov 8-bitni AVR mikrokontroler. Iako postoji više vrsta Arduino ploča (Uno, Nano, Mega, Leonardo itd.), koncept programiranja njihovog ponašanja je isti. Ono što međutim čini razlike među ovim verzijama, je kiličina radne memorije, kao i broj ulazno/iznaznih pinova. Ono što čini ovu platformu veoma popularnom jeste njena pristupačnost, kako cijenom, tako i stepenom potrebnog predznanja iz polja elektronike i integralnih kola.

Arduino, kao platforma, nije namijenjen za rješenja u produkciji i masovnu proizvodnju, već za izradu prototipa uređaja ili projekte koji spadaju u kategoriju hobija. Osobina, koja za to ima najveći značaj, je opća namjenjenost Arduina što ga u proizvodnji čini skupljim od ploča koje su napravljene da služe samo jednoj svrsi.

Pošto sam kroz razvoj ovog projekta koristio Arduino Mega, sve buduće reference će se odnositi na Mega model.

3.1 Proces kompajliranja

Programski jezik u kojem se pišu Arduino datoteke koje sadrže izvorni kod je C++. Međutim većina standardnih biblioteka su preuzete iz C programskog jezika, zbog male količine radne memorije kontrolera. Arduino datoteke je lako moguće prepoznati po njihovoj ".ino" eksteniziji. Ove datoteke se još nazivaju i skicama.

Nakon što se pokrene proces kompajliranja projekta, Arduino okruženje pravi male promjene u kodu, kako bi se nakon toga mogao proslijediti gcc i g++ kompajleru. U ovoj fazi se sve datoteke u direktoriju kombinuju u jedanu i na njen početak se dodaje `#Include<Arduino.h>` zaglavlje. Zatim slijedi povezivanje koda sa standardnim Arduino bibliotekama, a nakon toga i sa ostalim bibliotekama uključenim u direktorji skice. Kako bi bila uključena u proces prevođenja, biblioteka (njen .cpp i .h dokument) se mora nalaziti na `libraries\{NazivBiblioteke}` putanji.

Kada je skica povezana i prevedena, vrijeme je da takva bude prebačena u Arduino memoriju gdje će se izvršavati.

[1] https://github.com/arduino

3.2 Pisanje koda

Kao što je spomenuto u sekciji 3.1, Programski jezik koji koristimo pri programiranju Arduino kontrolera je C++. Najjednostavnija ".ino" datoteka se sastoji iz dvije funkcije:

1. Setup
2. Loop

Funkcija "Setup" se izvršava samo jednom pri pokretanju kontrolera i služi za inicijalizaciju komponenti i objekata. Druga funkcija, pod nazivom "Loop" je sama srž načina rada ovih ploča. Kod koji se nalazi unutar ove funkcije će se ciklično izvršavati sve dok je Arduino upaljen. Taj kod je sekvenca koja predstavlja i ponašanje isprogramirane ploče u upotrebi.

3.3 Razvojno okruženje

3.3.1 Arduino IDE

Arduino posjeduje svoje oficijelno okruženje pod nazivom "Arduino IDE". Ono dolazi u dvije verzije, online[2], i verzija koju instaliramo na lokalni računar. U ovom tekstu ću se fokusirati na offline okruženje, jer se uz njegovu instalaciju automatski instaliraju i Windows driveri potrebni za prebacivanje koda na Arduino..

Pri kreiranju prve skice ispred korisnika se nalazi "prazna" datoteka.

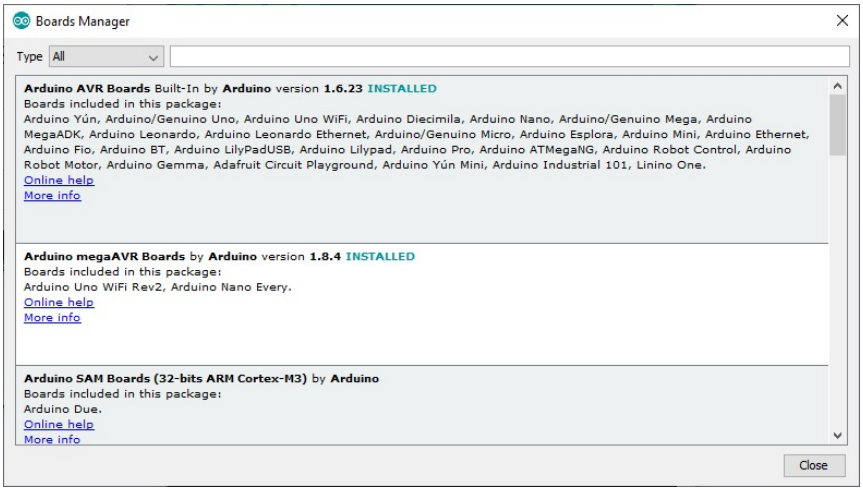
```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Alati

Boards manager je opcija koja se nalazi u alatnoj traci pod stavkom "Tools". Koristeći ovaj alat, biramo trenutnu arhitekturu ili model ploče na koju ćemo postaviti kod. S obzirom da je program otvorenog koda, kroz ovaj proces se mogu insatlirati i

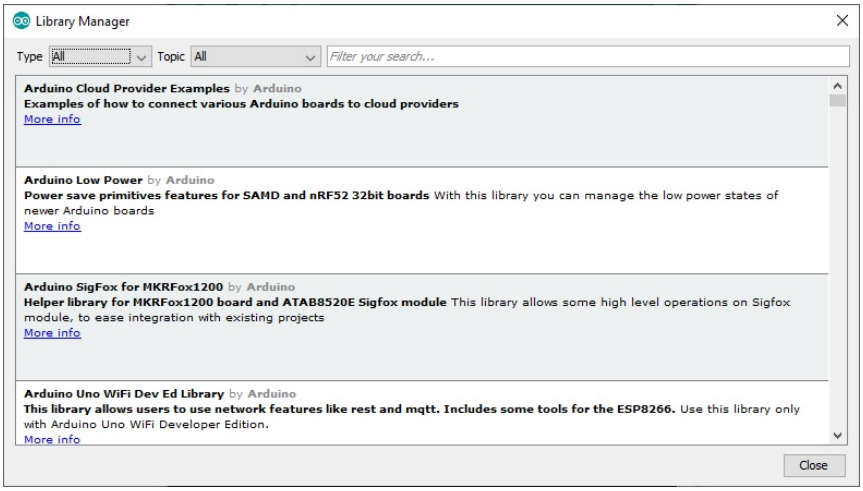
[2] https://create.arduino.cc/editor

datoteke koje nisu ugrađene u IDE, s tim da je u tim slučajevima potrebno obratiti pažnju na sigurnost. Na ovaj način moguće je koristiti ovo razvojno okruženje u svrhu pisanja koda i za druge platforme, ili klonove Arduino ploča. Iz prozora je moguće preuzeti i instalirati nove ploče, kao i nadograditi već postojeće



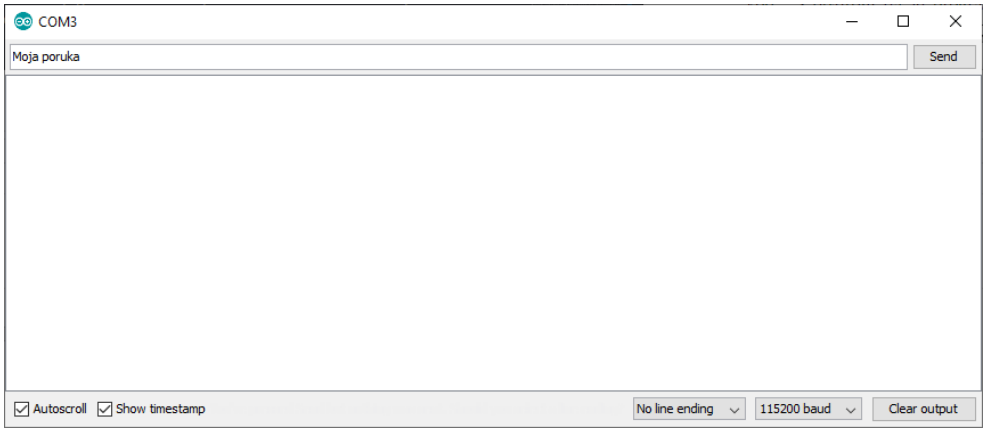
Slika 1: Boards manager prozor

Library manager , ovaj alat se također nalazi pod stavkom "Tools". Arduino IDE već dolazi sa predefinisanim repozitorijem biblioteka koje se nude za preuzimanje i instaliranje. Ukoliko korisnik želi preuzeti biblioteke iz drugih izvora to je moguće uraditi na dva načina: Instalacijom direktno iz ".zip" datoteke ili dodavanjem putanje repozitorija u postavkama okruženja. Brisanje se mora raditi ručno na putanji `%HOMEPATH%\Documents\Arduino\lib`



Slika 2: Library manager prozor

Serial monitor je jedan od načina da se uspostavi komunikacija sa Arduino uređajem dok je isti pokrenut. Kada se ovaj alat pokrene, korisnik može pomoću serijske komunikacije razmjenjivati tekstualne poruke sa Arduinom. Pored na dnu prozora se nalazi padajući izbornik, čija trenutno odabrana stavka definiše brzinu komunikacije u bitovima po sekundi.



Slika 3: Serial monitor prozor

3.3.2 PlatformIO

Pored Arduino IDE, programerima na izboru stoji još razvojnih okruženja u kojima mogu razvijati svoje Arduino kodove. Jedno od tih okruženja je PlatformIO.

PlatformIO se instalira u vidu nadogradnje za Visual Studio Code, koji i sam spada u kategoriju programa otvorenog koda. Kao i Arduino IDE, nudi izbor ploče koja će se prorgamirati, među kojima je oko 700 ponuđenih[3], pored samog Arduina. S obzirom da je sada razvojno okruženje Visual Studio Code, to dolazi sa svim njegovim prednostima(Markiranje sintakse, Intellisense, Personalnim postavkama okruženja itd.).

Razlog, pored već navedenih, zbog kojeg sam ja izabrao PlatformIO je njegova arhitektura projekta. Naime, u ovoj platformi se koristi standardna arhitektura cpp-a, što je nekome ko dolazi iz tog svijeta, daleko lakše za održavati. Još jedna minorna razlika između ova dva okruženja je to da glavna datoteka sa izvornim kodom u PlatformIO nije .ino, već "main.cpp" u kojoj je obavezno uključiti `Arduino.h` biblioteku.

[3] https://platformio.org/

3.4 Tipovi varijabli

Pošto ploča ima ograničenu radnu memoriju. Izbor tipova podataka je veoma bitan, zbog njihovog predefinisanog memorijskog prostora koji zauzimaju. Neki od najčešće korištenih tipova su:

- bool (8 bita) - može sadržati jednu od dvije vrijednosti (da ili ne)
- byte (8 bita) - cijeli broj od 0 do 255, bez predznaka
- char (8 bita) - cijeli broj između -127 i 127 kojeg će kompajler pokušati prevesti u karakter
- word (16 bita) - cijeli broj bez predznaka između 0 i 65 535
- int (16 bita) - cijeli broj između -32 768 i 32 767
- long (32 bita) - cijeli broj između 2 147 483 648 i 2 147 483 647
- unsigned long (32 bita) - cijeli broj između 0 4 294 967 295, bez predznaka.
- float (32 bita) - broj sa plutajućom tačkom između -3.4028235E38 i 3.4028235E38

Niz je indeksirana kolekcija varijabli bilo kojeg tipa.

String se Pored već navedenih tipova, u praksi se veoma često koristi, kao tip podatka. Ovaj tip podatka nema fiksnu veličinu koju zauzima u memoriji i sastoji se od niza karaktera. Ono što ga čini korisnim su funkcije koje su ponuđene za rad sa stringovima kao što su:

- CharAt(n) - vraća karakter na poziciji n
- IndexOf(c) - vraća prvu poziciju karaktera c u stringu
- Concat(val) - dodaje vrijednost iz varijable val na kraj stringa
- Replace(sub1, sub2) - vrši zamjenu svih instanci sub1, instancom sub2 u stringu
- Substring(n, k) - vraća komad stringa između n i k pozicija
- Lenght() - vraća dužinu stringa

Pokazivači su tipovi podataka koji upućuju na adresu. Diferenciraju se pomoću znaka *, ako je varijabla x, onda je &x adresa te varijable.

3.5 Konstante

`Arduino.h` biblioteka dolazi sa korisnim konstantama.

- INPUT | OUTPUT - ova konstanta upravlja električno ponoašanje pina, i priprema pin da prihvata ulazne, ili šalje izlazne signale.
- HIGH | LOW - odnosi se na pinove i ima može imati različito ponašanje
    1. Kada je pin postavljen kao INPUT, onda je ovo povratna vrijednost koju vraća očitavanje tog pina, ukoliko je struja u njemu veća od 3.0V ili 2.0V u zavisnosti od ploče koja se koristi, vratit će se HIGH, a ukoliko je manje, LOW.
    2. Kada je pin postavljen kao OUTPUT, sa HIGH će se njen izlaz postavilja na 3.3V ili 5V u zavisnoti od ploče koja se koristi, a sa LOW na 1.5V ili 1.0V.
- e | E - koriste se zbog lakše čitljivosti koda, predstavljaju eksponencijalni dio izraza 10^n koji se množi sa brojem koji mu prethodi.

3.6 Funkcije

Ono što omogćava kontrolisanje ulaza i izlaza Arduino ploče jesu funkcije `Arduino.h` biblioteke.

delayMicroseconds(value) pravi zastoj vremenske dužine value u mikrosekundama.

micros() vraća unsigned long koji predstavlja broj mikrosekundi koje su prošle od paljenja Arduino ploče.

pinMode(pin, INPUT | OUTPUT) upravlja ponašanjem pina i na osnovu konstante INPUT ili OUTPUT određuje da li će pin očitavati ili slati struju na svom kraju.

digitalRead(pin) očitava trenutno stanje pina čije je ponašanje postavljeno na očitavanje i vraća LOW ili HIGH konstantu u zavisnosti od voltaže koju očitava.

digitalWrite(pin, HIGH | LOW) postavlja jačinu protoka struje na izlazu pina na visoko ili nisko.

analogRead(pin) vraća integer vrijednost između 0 i 1024, koji je direktno proporcionalan jačini struje na pinu.

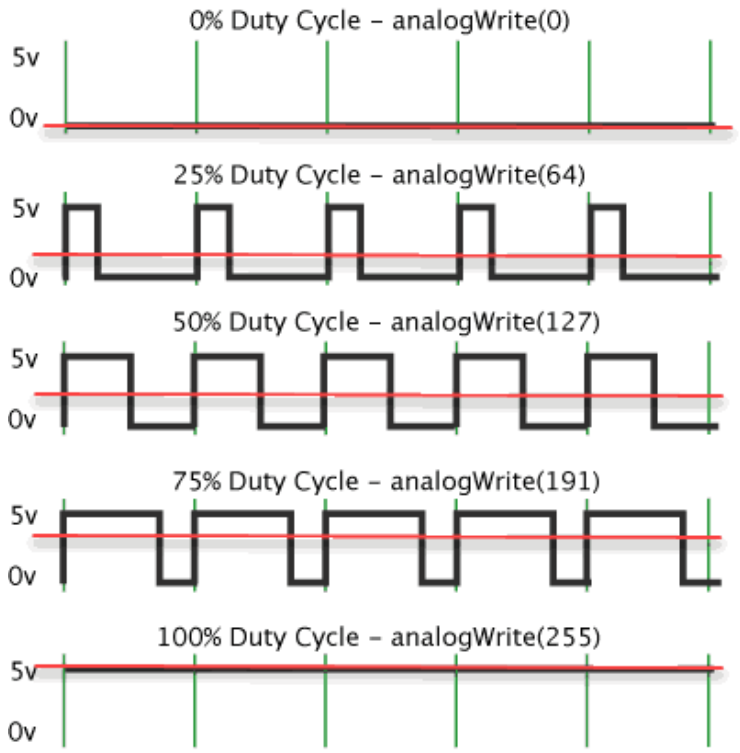
analogWrite(pin, value) radi modulaciju širine pulsa u što simulira jačinu struje na izlazu pina, pošto analogni izlaz na Arduinu ne postoji. Value ima raspon između 0 i 255 što je proporcionalno prividnoj jačini struje koju će pin proizvesti.

Serial.Begin(rate) inicijalizira serijski protok podataka, gdje je rate brzina prenosa podataka u bitima po sekundi.

Serial.Read() vraća prvi bajt serijske komunikacije na ulazu.

Serial.Write(String) šalje podatke u binarnom obliku na serijski izlaz.

attachIntterupt(digitalPinToInterrupt(pin), ISR, mode) Povezuje stanje pina mode (CHANGE, LOW, RISING, FALLING) sa prekidom u izvršenju loop funkcije i poziva ISR (rutina interapcije) koji je funkcija koja ne prima niti vraća parametre.



Slika 4: Primjer modulacije širine pulsa

3.7 Pinovi

Svaka verzija na sebi ima različit broj i mogućnosti pinova, ali svaka posjeduje barem po jedan pin svake vrste,sa izuzecima, kako bi se omogućile jednake funkcionalnosti svih ploča.

Dakle tipovi pinova koji postoje su:

- Digitalni - mogu biti ulazni i izlazni, postavljaju se i očitavaju samo 2 stanja, pomoću funkcije pinMode() (poglavlje 3.6) se odabire režim u kojem će raditi, a nakon toga se može ili slati struja jačine 1.0V/1.5V ili 3.3V/5V postavljajući digitalWrite() (poglavlje 3.6) HIGH ili LOW retrospektivno, ili očitavati jačina struje koristeći digitalRead() (poglavlje 3.6)
- PWM - digitalni pinovi koji podržavaju Pulse Width Modulation, tj. modulaciju širine pulsa (Slika 4)
- TX - digitalni pin, koristi se za serijsku transmisiju podataka, Serial.Write() (poglavlje 3.6)vrši ispis na ovaj pin.
- RX - digitalni pin, koristi se za serijsko čitanje podataka, Serial.read() (poglavlje 3.6) vrši učitavanje sa ovog pina.
- SCL - linija sata koja se koristi u I^2C komunikaciji !DODATI REF!
- SCL - linija podataka koja se koristi u I^2C komunikaciji
- INT - digitalni pinovi koji podržavaju prekide, aktiviraju se attachIntterupt() (poglavlje 3.6) funkcijom
- Analogni - pinovi koji imaju mogućnost analognog čitanja jačine struje pomoću analogRead() (poglavlje 3.6) funkcije
- Power - pinovi vezani za ulaz i izlaz struje u ploči
  - 5V - izlaz sa strujom jačine 5V
  - 3.3V - izlaz sa strujom jačine 3.3V
  - GND - uzemljenje
  - VIN - pin kroz koji se Arduino može napajati strujom od 9V
  - RESET - spajanjem na ground restartuje Arduino

3.8 MEGA

Arduino MEGA je jedna od verzija Arduino ploča. Na sebi ima ATMEGA2560 mikrokontroler.

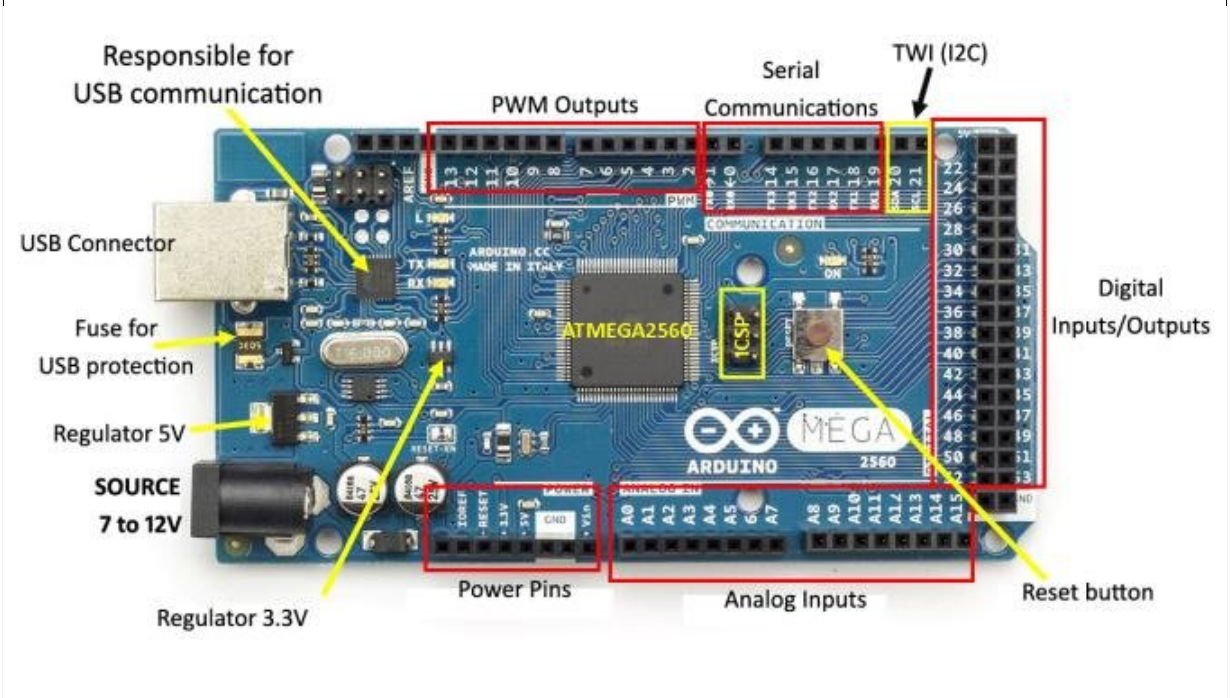
Ploča zahtjeva ulaznu struju od 6V do 20V, s tim da je preporučeno između 7V i 12V. Načini na koje se struja može dovesti u kontroler su kroz USB A interfejs, koji se ujedno

i koristi za povezivanje sa računarom, kroz 2.1mm priključak za napajanje sa pozitivnim centrom, ili direktno kroz VIN pin.

Flash memorija	128 KB
Veličina bootladera	8 KB
SRAM	8 KB
EEPROM	4 KB
Brzina sata	16 MHz
Jačina struje I/O pinova	40mA
Jačina struje na 3.3V pinu	60mA

Tabela 1: Arduino MEGA specifikcaije

Na slici 5 je prikazana Arduino MEGA ploča, te su pinovi označeni njihovim ulogama.



Slika 5: Šema pinova na ATMEGA2560

Dužina	101.6 mm
VŠirina	53.3 mm

Tabela 2: Arduino MEGA dimenzije

4 Android aplikacija

5 HC05

6 Koracni motori

7 MPU6050

8 Projekt

9 Zakljucak