

Računalni sustavi stvarnog vremena

SVJETLO ZA BIKIKL

PROJEKTNI ZADATAK

Grupa:

Ilija Jazvić

Ivana Ribičić

Osijek, 2024.

Sadržaj

1.	Opis projektnog zadatka	1
2.	Prijedlog rješenja	2
2.1.	Arduino IDE	2
2.2.	Croduino Basic	2
2.3.	Fotootpornik	3
2.4.	Akcelerometar + žiroskop MPU-6050 (GY-521).....	3
2.5.	LED diode	4
2.6.	Shema spoja.....	5
3.	Opis rješenja	6
4.	Zaključak	11

1. Opis projektnog zadatka

Zadatak je bio napraviti svjetlo za bicikl na kojemu se crvena LED automatski pali prilikom kočenja, dok narančaste LED predstavljaju žmigavce te se pale kako se bicikl naginje. Potrebno je upravljati jačinom prednjeg svjetla na temelju trenutne osvijetljenosti, a informacije o stanju bicikla se dobivaju sa žiroskopa. Za realiziranje ovog zadatka potrebno je od sklopovlja koristiti akcelerometar (žiroskop) za dobivanje podataka o stanju bicikla odnosno da li se kreće, da li je u stanju mirovanja, te da li skreće i u koju stranu. Zatim je potrebno koristiti fotootpornik za dobivanje podataka o vanjskoj osvijetljenosti, te LED koje nam prikazuju i signaliziraju određeno stanje bicikla, te sami mikrokontroler koji je sami mozak projekta. Što se tiče potrebne programske podrške korišten je Arduino IDE koji je opisan u nastavku, biblioteka Wire.h koja služi za komunikaciju s žiroskopom.

2. Prijedlog rješenja

Svjetlo za bicikl treba raditi tako da se u ovisnosti o osvijetljenosti okoline intenzitet prednjeg svjetla povećava, odnosno smanjuje. Crveno svjetlo pali se prilikom kočenja bicikla, dok se žmigavci pale prilikom skretanja.

Korištene komponente:

- Croduino Basic pločica
- Fotootpornik
- Akcelerometar + žiroskop MPU-6050 (GY-521)
- LED diode

2.1. Arduino IDE

Programski kod napisan je u Arduino IDE razvojnom okruženju. Arduino je ime za otvorenu računalnu i softversku platformu koja omogućava dizajnerima i konstruktorima stvaranje uređaja i naprava koje omogućuju spajanje računala s fizičkim svijetom tj. stvaranje interneta stvari. Arduino je stvorila talijanska tvrtka SmartProjects 2005. rabeći 8-bitne mikrokontrolere Atmel AVR, da bi stvorili jednostavnu, malu i jeftinu platformu s kojom bi mogli lakše povezivati računala s fizičkim svijetom.

Svaki Arduino program sastoji se od dvije osnovne petlje, *loop()* i *setup()*. Kod pisan u *setup* dijelu izvršava se samo jednom, odmah nakon uključivanja mikrokontrolera. Ovaj dio koda služi za inicijalizaciju mikrokontrolera. Odmah nakon izvršenja *setup* dijela izvršava se kod koji se nalazi u *loop* dijelu programa. Sve naredbe napisane u ovom dijelu programa ponavljaju se sve dok se mikrokontroler ne resetira.

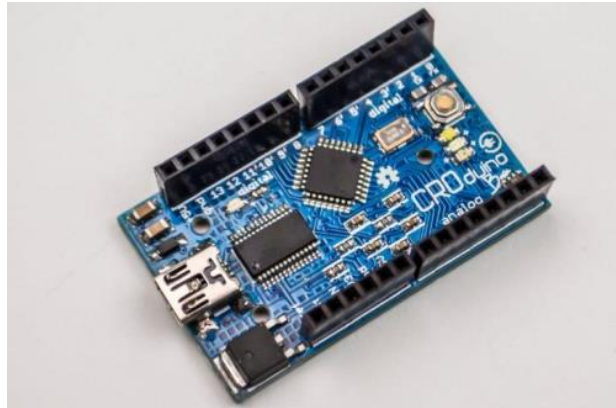
Za potrebe rješenja zadatka korištena je biblioteka *Wire.h* kako bi se omogućila komunikacija sa žiroskopom preko I2C protokola.

2.2. Croduino Basic

Croduino Basic je standardna hrvatska Arduino kompatibilna pločica. Pločica kompatibilna s Arduinom (Nano). Neke od karakteristika pločice:

- koristi Atmega328 mikroupravljač
- 14 digitalnih ulaza/izlaza (od toga 6 s PWM opcijom)
- 8 analognih ulaza
- FTDI RS232 se koristi kao USB pretvornik (potrebni FTDI driveri)
- Minijturnih dimenzija 50x30mm
- Ugrađen 5V regulator napona s ciljem da se omogući korištenje vanjskog napajanja (7-20V, preporučeno 9 ili 12V). Pristupa mu se putem Vin pina
- Ugrađena tipka za reset
- Narančasta LEDica na pinu broj 13

- Plava LED kao indikator napajanja, bijela i crvena kao TX/RX indikatori



Slika 2.1. Arduino Basic pločica

2.3. Fotootpornik

Fotootpornik (eng. photoresistor ili light dependent resistor- LDR) je otpornik, čiji se električni otpor smanjuje s povećanjem intenziteta ulazne svjetlosti. Promjenom otpora fotootpornika mijenja se napon na točki koja je spojena na mikrokontroler pa očitavanjem tog analognog izvoda možemo odrediti koliko je fotootpornik osvijetljen.

Ova vrijednost korištena je za postavljanje jačine intenziteta prednjeg svjetla bicikla.



Slika 1.2. Fotootpornik

2.4. Akcelerometar + žiroskop MPU-6050 (GY-521)

Senzor MPU-6050 kompletan je 6-osni uređaj za praćenje kretanja. Kombinira 3-osni žiroskop, 3-osni akcelerometar i digitalni procesor pokreta. Koristi I2C sučelje sabirnice za komunikaciju s mikrokontrolerima. 3-osni žiroskop s tehnologijom mikroelektro-mehaničkog sustava (MEMS) koristi se za detekciju brzine rotacije duž x, y i z osi.

Ove informacije korištene su za računanje kuteva u smjeru x i y. Kočenje je predstavljeno kao promjena kuta u smjeru x osi, a lijevi i desni žmigavac kao promjena veličine kuta u smjeru y osi.



Slika 2.3. Žiroskop

2.5. LED diode

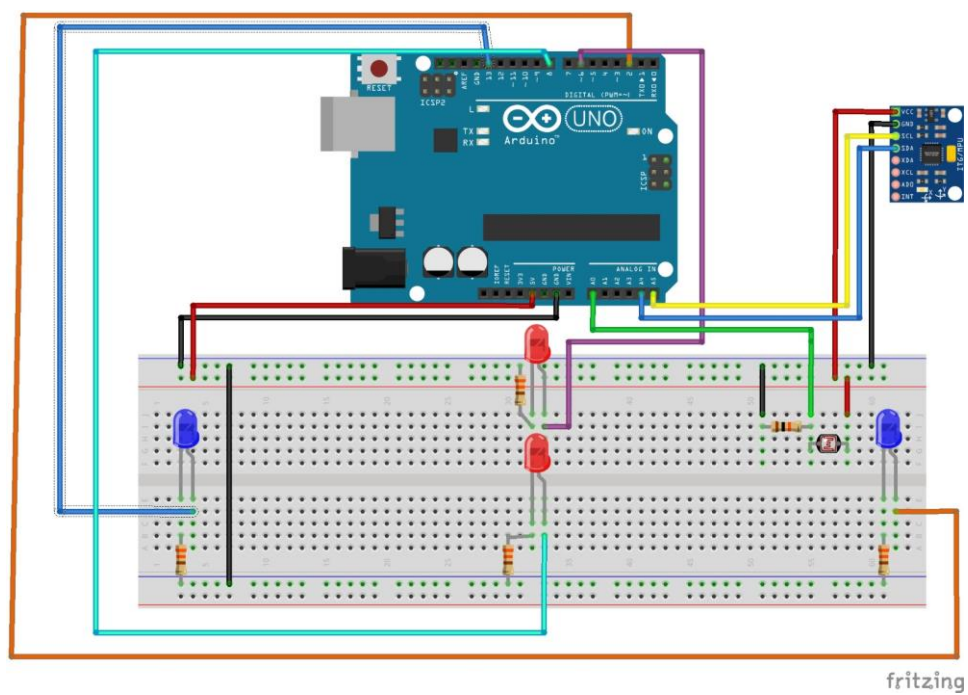
Svjetleća dioda ili LED (skr. od engl. light-emitting diode) je poluvodički elektronički element koji pretvara električni signal u optički (svjetlost). Propusno polarizirana svjetleća dioda emitira elektromagnetsko zračenje na način spontane emisije uzrokovane rekombinacijom nosilaca električnoga naboja (elektroluminiscencija). Elektroni prelazeći iz vodljivog u valentni pojas, oslobađaju energiju, koja se dijelom očituje kao toplina, a dijelom kao zračenje. Boja emitirano­g svjetla ovisi o poluvodiču, kao i o primjesama u njemu i varira od infracrvenog preko vidljivog do ultraljubičastog dijela spektra.

Za realizaciju svjetla za bicikl korištene su četiri LED diode. Jedna predstavlja prednje svjetlo čiji intenzitet ovisi o vrijednosti dobivenoj s fotootpornika. Druga predstavlja svjetlo koje se pali prilikom kočenja bicikla što je predstavljeno promjenom kuta duž x osi žiroskopa. Preostale dvije su lijevi i desni žmigavac koje trepere u ovisnosti o tome na koju stranu biciklo skreće, odnosno ovisno o vrijednosti kuta očitano preko žiroskopa.



Slika 2.4. LED dioda

2.6. Shema spoja



Slika 2.5. Shema spoja

3. Opis rješenja

Predloženo rješenje izrađeno je korištenjem prethodno navedenih komponenti te je napisan odgovarajući programski kod.

```
#include<Wire.h>

const int photoResistor = A0; // Fotootpornik, analogni pin A0
const int frontLight = 6;      // Prednje svjetlo bicikla, PWM pin
const int brakes = 8; // Svjetlo koje se pali prilikom kočenja
const int left = 13; // Lijevi žmigavac
const int right = 2; // Desni žmigavac

int value; // Vrijednost fotootpornika (0-1023)

const int MPU_addr = 0x68; // MPU6050 I2C adresa
int16_t AcX, AcY, AcZ;

int minVal = 265;
int maxVal = 402;

double x;
double y;
double z;

void setup(){

    // Definiranje ulaznih i izlaznih pinova
    pinMode(frontLight, OUTPUT);
    pinMode(brakes, OUTPUT);
    pinMode(left, OUTPUT);
    pinMode(right, OUTPUT);
    pinMode(photoResistor, INPUT);

    Wire.begin(); // Inicijalizacija komunikacije s MPU6050
    Wire.beginTransmission(MPU_addr); // Početak komunikacije
    Wire.write(0x6B); // Komunikacija s registrom 6B
    Wire.write(0); // Pisanje 0 u registar 6B
    Wire.endTransmission(true); // Kraj prijenosa
}

void loop(){

    value = analogRead(photoResistor); // Čitanje vrijednosti
    fotootpornika

    // Postavljanje vrijednosti prednjeg svjetla u ovisnosti o
    vrijednosti fotootpornika
```



```

if(value > 400 && value < 500){
    analogWrite(frontLight, 255);
}
else if(value >= 500 && value < 600){
    analogWrite(frontLight, 212);
}
else if(value >= 600 && value < 700){
    analogWrite(frontLight, 170);
}
else if(value >= 700 && value < 800){
    analogWrite(frontLight, 127);
}
else if(value >= 800 && value < 900){
    analogWrite(frontLight, 85);
}
else if(value >= 900 && value < 1000){
    analogWrite(frontLight, 42);
}
else if(value >= 1000){
    analogWrite(frontLight, 0);
}

// Čitanje vrijednosti akcelerometra
Wire.beginTransaction(MPU_addr);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr, 14, true);

// Vrijednosti osi
AcX = Wire.read()<<8|Wire.read(); // x-os
AcY = Wire.read()<<8|Wire.read(); // y-os
AcZ = Wire.read()<<8|Wire.read(); // z-os

// Računanje kuteva
int xAng = map(AcX, minVal, maxVal, -90, 90);
int yAng = map(AcY, minVal, maxVal, -90, 90);
int zAng = map(AcZ, minVal, maxVal, -90, 90);

// Pretvorba radijana u stupnjeve
x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI);
y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);
z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);

// Kočenje
if(x >= 20 && x <= 90){
    digitalWrite(brakes, HIGH);
}
else{
    digitalWrite(brakes, LOW);
}

```

```

    }

    // U ovisnosti o vrijednosti kuteva paliti lijevi ili desni
    žmigavac
    if(y >= 10 && y <= 100){
        digitalWrite(left, HIGH);
        digitalWrite(right, LOW);
        delay(200);
        digitalWrite(left, LOW);
    }
    else if(y >= 260 && y <= 350){
        digitalWrite(right, HIGH);
        digitalWrite(left, LOW);
        delay(200);
        digitalWrite(right, LOW);
    }
    else{
        digitalWrite(right, LOW);
        digitalWrite(left, LOW);
    }

    delay(500);
}

```

Na početku je bilo potrebno postaviti ulazne i izlazne pinove za Croduino, zatim inicijalizacija komunikacije s MPU6050, koristi se Wire biblioteka za uspostavljanje I2C veze s MPU6050 akcelerometrom. Prvo se šalje adresa uređaja putem Wire.beginTransmission(), a zatim se piše u registar 0x6B kako bi se konfigurirao akcelerometar.

Zatim imamo glavnu petlju u kojoj je prvo realizirano prednje svjetlo. Jačina prednjeg svjetla ovisi o vrijednosti dobivenoj s fotootpornika. Što je jača osvijetljenost okoline, vrijednost dobivena preko fotootpornika je veća, a prednje svjetlo tada svijetli slabijim intenzitetom. Kako bi se podešavala jačina svijetljenja LED diode koja predstavlja prednje svjetlo bicikla, korištena je pulsno-širinska modulacija PWM (engl. *pulse-width modulation*). Pulsno širinska modulacija radi na način da vrlo brzo uključuje i isključuje izvod mikrokontrolera, otprilike 1000 puta u sekundi. Pojačanje i smanjivanje intenziteta svjetlosti ovisi o tome koliki je dio vremena izvod uključen, a koliko isključen.

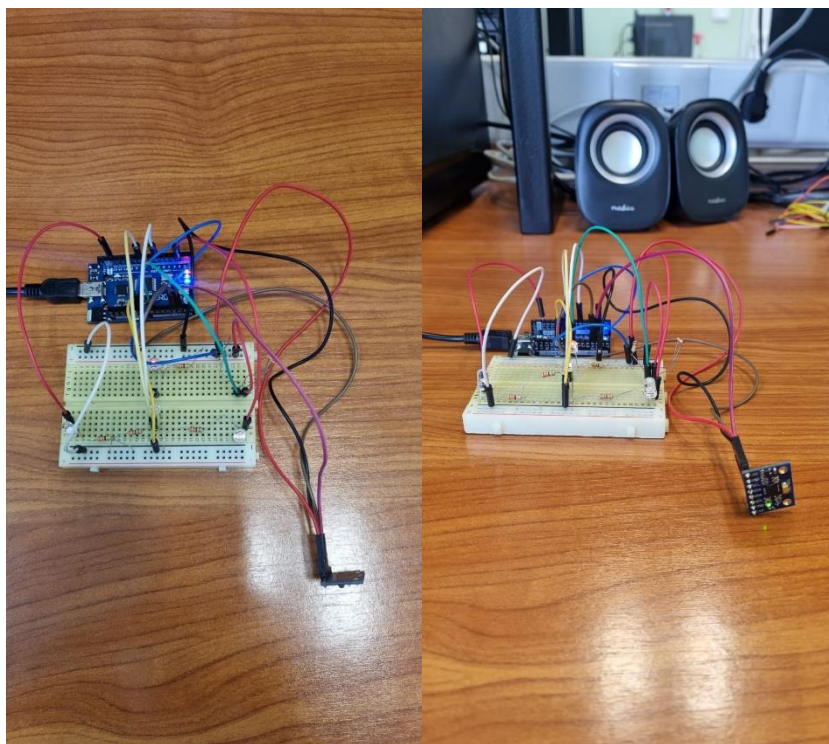
Nakon toga smo čitali podatke s akcelerometra, podaci o vrijednostima osi akcelerometra se čitaju iz uređaja i pohranjuju u varijable. Vrijednosti akcelerometra se mapiraju na kuteve koristeći map() funkciju, a zatim se pretvaraju iz radijana u stupnjeve te smo pomoću stupnjeva eksperimentirali (rotirali akcelerometar u različite osi) i ispisivali vrijednosti akcelerometra te smo zaključili rubne vrijednosti za određivanje stanja bicikla. X os akcelerometra nam predstavlja os koja će označavati da li je biciklo u pogonu ili biciklo koči, to se također može vidjeti u kodu. Y os nam predstavlja nagetost bicikla odnosno skretanje bicikla u jednu ili drugu stranu, također možemo vidjeti u kodu rubne uvjete koje

smo postavili kako bi razlikovali da li biciklo ide prema naprijed, skreće lijevo ili skreće desno.

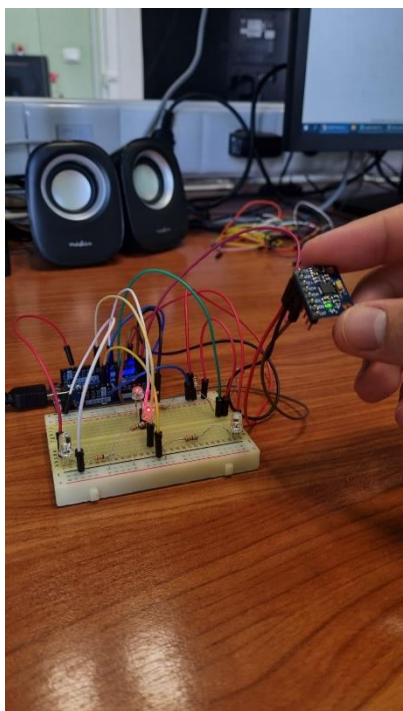
Prvo je bio realiziran dio za kočenje samog bicikla, svjetlo koje se pali prilikom kočenja pali se kada kut duž x osi poprimi vrijednost u definiranom intervalu, to jest kada se žiroskop iz nekog ravnog položaja pomjeri prema naprijed, s obzirom da bi definiranje negativnog ubrzanja bilo prilično nezgodno. Također imamo i interval vrijednosti x-a kada biciklo ne koči, a to je interval između 91 i 19 stupnjeva, gdje je maksimalna vrijednost $x = 360$.

Svjetla žmigavca su napravljena na način da se pale u ovisnosti na koju je stranu nagnut žiroskop što bi trebalo predstavljati nagib bicikla na određenu stranu. Vidimo da smo prvo ispitivali da li je vrijednost y u rasponu od 10 do 100 što nam predstavlja skretanje u lijevo. Te smo vrijednosti dobili pomenutim eksperimentiranjem sa žiroskopom kako bi definirali koje vrijednosti žiroskopa i koje osi će predstavljati stanje bicikla. Te je na kraju također ispitan uvjet da li biciklo skreće u desno. Vidimo da je razlika u intervalima 90 stupnjeva u oba slučaja. Isto tako je bilo područje u kojem smo predstavili da biciklo ide naprijed a to je područje između y vrijednosti 351 do 9, gdje je maksimalna vrijednost $y = 360$. Dodan je i delay koji predstavlja treperenje led kao što je i u stvarnosti na biciklu predstavljeno kočenje.

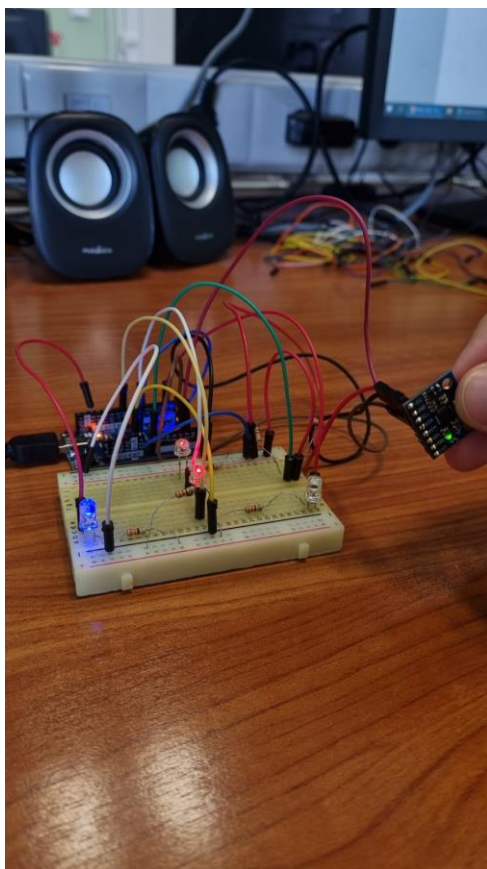
Na slikama u nastavku prikazan je izgled spoja.



Slika 3.1. Izgled projekta spojen na breadboard



Slika 3.2. Upaljeno crveno svjetlo zbog „kočenja“



Slika 3.3. Upaljen lijevi žmigavac

4. Zaključak

Ovaj Arduino kod omogućuje kontrolu biciklističkih svjetala i automatsko pokazivanje smjera na temelju okoline i nagiba bicikla. Kroz kombinaciju čitanja senzora fotootpornika i MPU6050 akcelerometra, uređaj prilagođava svjetlinu prednjeg svjetla ovisno o razini okolne svjetlosti te aktivira kočnice i pokazivače smjera prema nagibu bicikla. Ovaj sustav donosi dodatnu sigurnost i praktičnost biciklistima, omogućujući im da se bolje prilagode prometnim uvjetima i budu bolje vidljivi u prometu. Naravno riješeni projekt (kod) se može još unaprijediti tako da se još detaljnije specificiraju rubne vrijednosti za određeno stanje bicikla npr. kada skreće lijevo ili desno.