



MINISTARSTVO ZNANOSTI,
OBRAZOVANJA I MLADIH
REPUBLIKE HRVATSKE



AGENCIJA ZA ODGOJ
I OBRAZOVANJE



HRVATSKA ZAJEDNICA
TEHNIČKE KULTURE



NATJECANJE
MLADIH
TEHNIČARA

Natjecanje mladih tehničara RH

Područje automatika

Pripremni zadaci za školsku/klupsku i županijsku razinu
natjecanja

NATJECANJE
MLADIH
TEHNIČARA

Autor zadataka:

Hrvoje Vrhovski, Hrvatska zajednica tehničke kulture

Sadržaj

62. natjecanje mladih tehničara; 2019./2020., školska/klupska razina	4
Model upravljanja trčecim svjetlom.....	4
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	4
Zadatak:.....	4
Montažna shema:.....	5
Jedno od mogućih programskih rješenja:	6
63. natjecanje mladih tehničara; 2020./2021., školska/klupska razina	8
Model primopredajnika Morseovog koda.....	8
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	8
Priprema elemenata:.....	9
Zadatak:.....	9
Montažna shema:.....	10
Jedno od mogućih programskih rješenja:	11
64. natjecanje mladih tehničara; 2021./2022. , školska/klupska razina	12
Igra svjetla	12
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	12
Zadatak:.....	12
Montažna shema:.....	13
Jedno od mogućih programskih rješenja:	13
65. natjecanje mladih tehničara; 2022./2023., školska/klupska razina	15
Model prometnog raskrižja	15
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	15
Zadatak:.....	15
Montažna shema:.....	16
Jedno od mogućih programskih rješenja:	17
66. natjecanje mladih tehničara; 2023./2024., školska/klupska razina	19
Model upravljanja elektromotornim pogonom vozila	19
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	19
Zadatak:.....	19
Priprema elemenata:.....	19
Izrada programa za model upravljanja elektromotornim pogonom:.....	19
Montažna shema:.....	20
Jedno od mogućih programskih rješenja:	20



62. natjecanje mladih tehničara; 2019./2020., županijska razina.....	22
Model digitalne vage	22
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	22
Zadatak:.....	22
Montažna shema:.....	23
Jedno od mogućih programskih rješenja:	23
63. natjecanje mladih tehničara; 2020./2021., županijska razina.....	27
Model mosne vage	27
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	27
Priprema elemenata:.....	27
Opis funkcija elemenata:.....	28
Zadatak:.....	28
Montažna shema:.....	29
Jedno od mogućih programskih rješenja:	29
64. natjecanje mladih tehničara; 2021./2022., županijska razina.....	32
Model senzora za pomoć pri parkiranju vozila.....	32
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	32
Zadatak:.....	32
Montažna shema:.....	33
Jedno od mogućih programskih rješenja:	33
65. natjecanje mladih tehničara; 2022./2023., županijska razina.....	38
Model automatskih vrata	38
Materijal, oprema i alat potrebni za izradu zadatka:	38
Zadatak	38
Montažna shema:.....	40
Jedno od mogućih programskih rješenja:	40
66. natjecanje mladih tehničara; 2023./2024., županijska razina.....	44
Model elektromotornog pogona automobila	44
Oprema, pribor i materijal potrebni za izvršenje zadatka:.....	44
Zadatak:.....	45
Montažna shema:.....	45
Jedno od mogućih programskih rješenja:	47

62. natjecanje mladih tehničara; 2019./2020., školska/klupska razina

Model upravljanja trčecim svjetlom

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Mikroupravljačko sučelje prema vlastitom izboru	1 kom
Ekperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti - duljine 10 cm do 20cm	1 kom
Potenciomtar 10 kOhma	1 kom
Fotootpornik	1 kom
Svjetleće diode iste boje promjera 5mm	8 kom
Otpornik 330 Ohma	8 kom
Otpornik 10 kOhma	1 kom
Tipkalo	2 kom

Zadatak:

Tvoj zadatak je na eksperimentalnoj pločici izraditi model upravljanja trčecim svjetlom. Trčecim svjetlom upravlja se pomoću potencimetra i to zavisno o svjetlosnim uvjetima.

Na eksperimentalnu pločicu postavi 8 (osam) svjetlećih dioda i pripadajuće otpornike, fotootpornik s pripadajućim otpornikom, potenciomtar i dva tipkala. Zatim sve elemente spoji s tvojim mikrokontrolerskim sučeljem. Provjeri jesi li svi elemente ispravno povezani s mikrokontrolerskim sučeljem i tek tada poveži sučelje s računalom i ako je potrebno s izvorom napajanja.

Napiši program za upravljanje radom trčeceg svjetla prema sljedećim zahtjevima:

Program se pokreće pritiskom na prvo tipkalo, a pritiskom na drugo tipkalo počinje se izvršavati iznova (program se pritiskom na drugo tipkalo resetira).

Tempo i način rada trčeceg svjetla određuje se potenciomtrom u zavisnosti od uvjeta koje očitava fotootpornik. Fotootpornik treba očitavati samo dva stanja osvijetljenosti, stanje jače osvijetljenosti i stanje slabije osvijetljenosti.

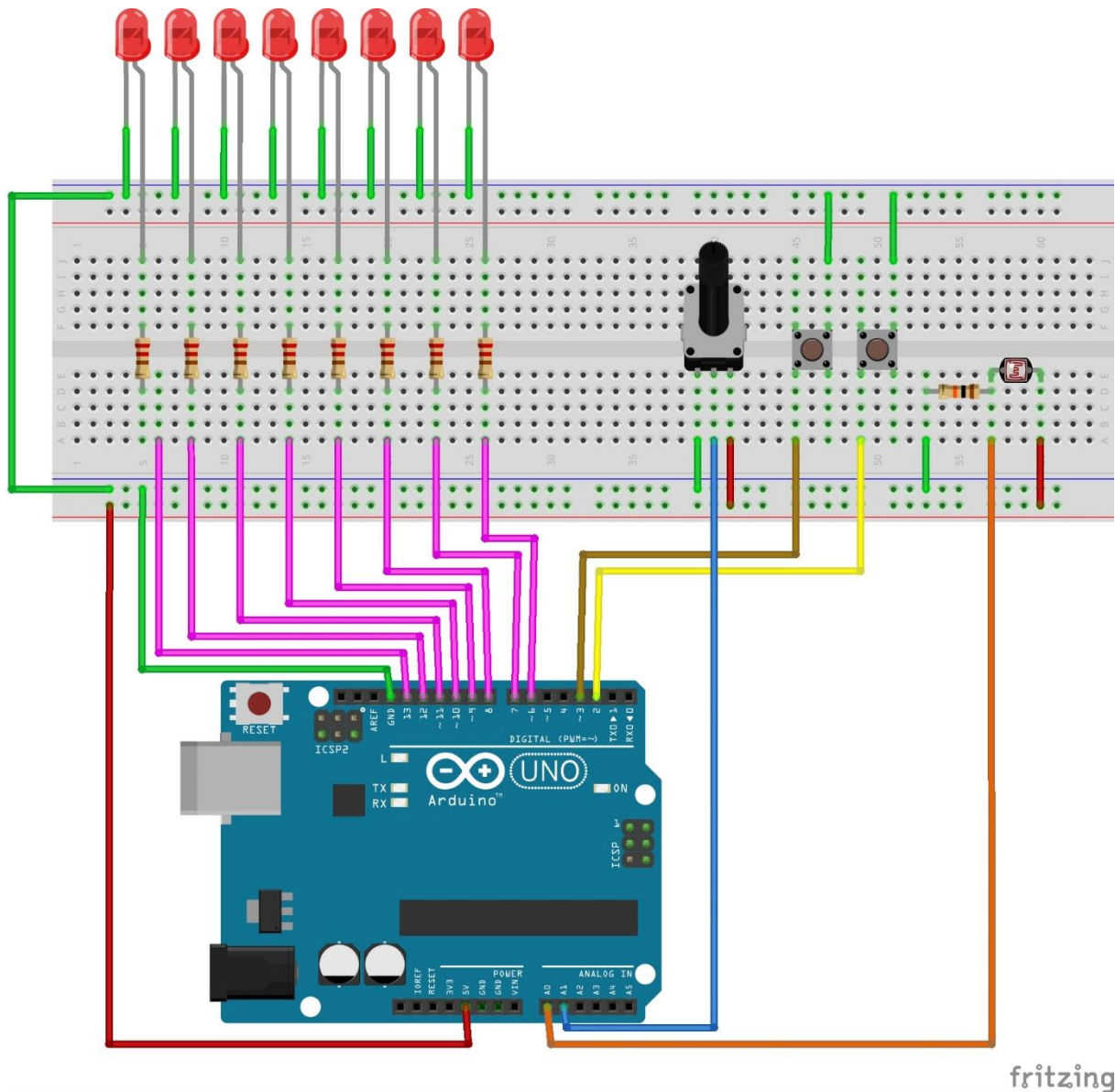
U slučaju kada fotootpornik očitava stanje jače osvijetljenosti, tvoj uređaj je u modu rada trčeceg svjetla čiji se tempo određuje potenciomtrom. U svakom trenutku mora biti upaljena samo jedna svjetleća dioda.

U slučaju kada fotootpornik očitava stanje slabije osvijetljenosti, tvoj uređaj je u modu treptanja. U ovom modu rada sve svjetleće diode su istovremeno upaljene ili ugašene, a tempo njihovog treptanja određuje se potenciomtrom.

Kad dovršiš program i ako je sve u redu pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.



Montažna shema:



TEHNICARA



Jedno od mogućih programskih rješenja:

```
int tipka1=2;
int tipka2=3;
int foto=A0;
int pot=A1;
int stanjefoto=0;
int stanjepot=0;
int x=0;

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(tipka1,INPUT_PULLUP);
  pinMode(tipka2,INPUT_PULLUP);
  pinMode(foto,INPUT);
  pinMode(pot,INPUT);
  Serial.begin(9600);
}

void loop() {
  while(digitalRead(tipka1)==1){
    stanjefoto=analogRead(foto);
    while(digitalRead(tipka2)==1){
      stanjefoto=analogRead(foto);
      Serial.println(stanjefoto);
      if (stanjefoto>600){
        stanjefoto=analogRead(foto);
        stanjepot=analogRead(pot)+1;
        Serial.print(stanjefoto);
        Serial.print(".....");
        Serial.println(stanjepot);
        for (x=6;x<14;x++){
          digitalWrite(x,HIGH);
          delay(stanjepot);
          digitalWrite(x,LOW);
          stanjefoto=analogRead(foto);
```



```
    stanjepot=analogRead(pot)+1;
    Serial.print(stanjefoto);
    Serial.print(".....");
    Serial.println(stanjepot);
}
}
else{
    stanjefoto=analogRead(foto);
    stanjepot=analogRead(pot)+1;
    Serial.print(stanjefoto);
    Serial.print(".....");
    Serial.println(stanjepot);
    for (x=6;x<14;x++){
        digitalWrite(6,HIGH);
        digitalWrite(7,HIGH);
        digitalWrite(8,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(10,HIGH);
        digitalWrite(11,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(13,HIGH);
        delay(stanjepot);
        digitalWrite(6,LOW);
        digitalWrite(7,LOW);
        digitalWrite(8,LOW);
        digitalWrite(9,LOW);
        digitalWrite(10,LOW);
        digitalWrite(11,LOW);
        digitalWrite(12,LOW);
        digitalWrite(13,LOW);
        delay(stanjepot);
        stanjefoto=analogRead(foto);
        stanjepot=analogRead(pot)+1;
        Serial.print(stanjefoto);
        Serial.print(".....");
        Serial.println(stanjepot);
    }
}
}
```



63. natjecanje mladih tehničara; 2020./2021., školska/klupska razina

Model primopredajnika Morseovog koda

Morseov kod je način prenošenja informacija upotrebom standardiziranih sekvenci (nizova) kraćih i duljih impulsa. Originalno je kreiran za potrebe električnog telegrafa sredinom 1830-tih, a upotrebljavao se i u radio komunikaciji. Razvojem komunikacijske tehnologije, Morseov kod je zastario, no i danas se koristi kao komunikacija u slučaju opasnosti, kao navigacijski radiosignal, u radioamaterstvu, kao identifikacija signala mobilnih telefona, a koriste ga naravno i izviđači.

Morseov kod je zamišljen da omogućí operaterima "očitanje i dešifriranje" oznake sa papirne vrpce. Prvotna ideja je bila da se šalju samo brojevi koji bi označavali riječi koje bi operater potražio u knjizi, međutim uskoro se razvila potreba za nekim specijalnim znakovima i složenijim porukama, tako da su sva slova i brojevi dobili svoju oznaku u crticama i točkicama. U pisanju Morseovog koda koriste se dva simbola "-" (dah) i "." (dot). Točkica ili dot je mjerna jedinica vremena i oznaka tempa slanja poruke. Jedna crtica obično traje tri točkice.

Morseov kod može biti emitiran na razne načine. Originalno je emitiran kao električni impuls duž telegrafске žice, ali isto kao i audio zvuk, radio signal, kao mehanički ili vizualni signal (npr. bljeskovi svijetla). Budući da se za emitiranje koda koriste samo dva stanja (on -off, 0 -1, da -ne...) to je ustvari rana primjena binarnog koda.

Internacionalno prihvaćeni standard za Morseov kod se sastoji od šest elemenata:

- Kratki znak, ili točka
- Dugi znak, ili crta
- Razmak među elementima istog slova - traje koliko i jedna točkica
- Razmak između slova - traje koliko i 3 točkice
- Razmak između riječi - traje koliko i 5 točkica
- Razmak između rečenica

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Mikroupravljačko sučelje prema vlastitom izboru	1 kom
Programator (ako je zaseban)	1 kom
Kabel za spajanje mikroupravljačkog sučelja s računalom	1 kom
Računalo s instaliranim softverskim paketom za programiranje odabranog sučelja	1 kom
Univerzalni mjerni instrument	1 kom
Produžni kabel s minimalno 3 utičnice, minimalne duljine 3 m	1 kom
Eksperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti – razne duljine (od 1 do 10 cm)	1 kom
Fotootpornik	1 kom
Bijela svjetleća dioda promjera 5 mm	1 kom
Crvena svjetleća dioda promjera 5 mm	1 kom
Otpornik 330 ohma	2 kom
Otpornik 10 Kohma	1 kom
Tipkalo	1 kom
Zujalica	1 kom
Mala sjekača kliješta za elektroniku	



Kliješta za uklanjanje izolacije s električnih vodiča	
Odvijač ravni	
Modelarski nožić – skalpel	
Obična ili tehnička olovka	
Ravnalo	

Priprema elemenata:

Za izradu tehničke tvorevine će ti trebati eksperimentalna pločica, mikroupravljačko sučelje, tipkalo, bijela svjetleća dioda, crvena svjetleća dioda, dva otpornika od 330 ohma, fotootpornik, otpornik od 10 Kohma i zujalica.

Model ćeš izraditi na eksperimentalnoj pločici. Na pločicu postavi tipkalo, crvenu svjetleću diodu i pripadajući otpornik, bijelu svjetleću diodu i pripadajući otpornik, zujalicu te fotootpornik s pripadajućim otpornikom. Fotootpornik postavi tako da ga bijela svjetleća dioda izravno osvjetljava.

Funkcije tipkala i bijele svjetleće diode se ne programiraju, nego je tipkalo spojeno serijski sa svjetlećom diodom i njenim zaštitnim otpornikom. Strujni krug bijele svjetleće diode izgleda ovako:

- pozitivni pol napajanja sa mikrokontrolerskog sučelja ili bilo kojeg drugog izvora napajanja,
- tipkalo,
- zaštitni otpornik (ako je izvor napajanja 3,3 V, tada ne treba postaviti zaštitni otpornik),
- bijela svjetleća dioda,
- negativni pol napajanja.

Svakim pritiskom na tipkalo, zatvara se strujni krug bijele svjetleće diode i ona se pali. Otpuštanjem tipkala, bijela svjetleća dioda se gasi. Dakle, svakim pritiskom na tipkalo odašilje se svjetlosni signal. Svaki pritisak na tipkalo, ujedno i odašiljanje svjetlosnog signala sa bijele svjetleće diode, možemo shvatiti kao jedan impuls. Taj impuls traje onoliko dugo koliko dugo je tipkalo pritisnuto.

Zadatak:

Izradi program kojim ćeš pomoću fotootpornika očitavati Morseov kod koji svjetlosnim signalima odašilje bijela svjetleća dioda.

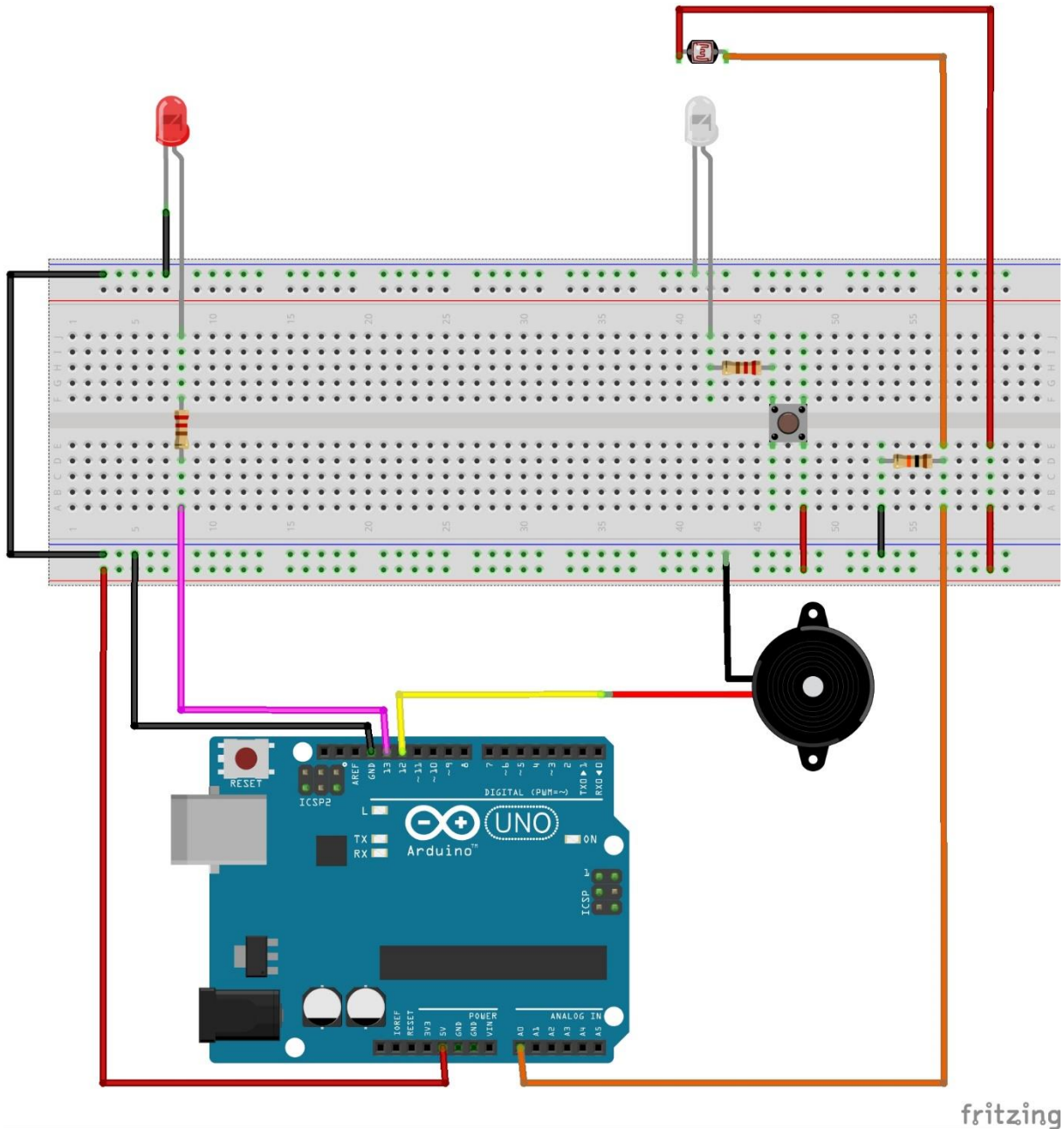
Kad se pritisne tipkalo, bijela svjetleća dioda svijetli. Bijela svjetleća dioda svijetli dok god je tipkalo pritisnuto. Kad se tipkalo otpusti bijela svjetleća dioda se gasi. Taj dio primopredajnika se ne programira, nego se spaja direktno na napajanje i uzemljenje mikrokontrolerskog sučelja.

Fotootpornik očitava svjetlosne signale koje odašilje bijela svjetleća dioda te se, zavisno od njegovog očitavanja, u istom tempu odašiljanja signala aktiviraju ili deaktiviraju crvena svjetleća dioda i zujalica. Bez obzira na duljinu trajanja pritiska na tipkalo, maksimalna duljina trajanja prikazanog signala može biti jedna sekunda. Drugim riječima, crvena svjetleća dioda i zujalica ne smiju biti aktivni dulje od jedne sekunde u pojedinom impulsu, bez obzira na to koliko dugo je pritisnuto tipkalo. S druge strane, bijela svjetleća dioda mora svijetliti dok god je tipkalo pritisnuto. Ako se tipkalo drži pritisnutim dulje od jedne sekunde, na serijskom se monitoru mora ispisivati poruka „Tipkalo predugo pritisnuto...“.



Kad dovršiš program i ako je sve u redu, pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.

Montažna shema:



Jedno od mogućih programskih rješenja:

```
long vrijeme=millis();
long pocvr=vrijeme;
int foto=analogRead(A0);

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  foto=analogRead(A0);
  Serial.println(foto);
  if(foto>800){ // podesiti prema vrijednosti koju očitava fotootpornik
    foto=analogRead(A0);
    if((pocvr+1000)>vrijeme){
      vrijeme=millis();
      digitalWrite(13,HIGH);
      digitalWrite(12,HIGH);
    }
    else{
      vrijeme=millis();
      pocvr=0;
      foto=0;
      digitalWrite(13,LOW);
      digitalWrite(12,LOW);
      Serial.println("Predugo stisnuto tipkalo...");
    }
  }
  else{
    vrijeme=millis();
    foto=0;
    pocvr=vrijeme;
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
  }
}
```



64. natjecanje mladih tehničara; 2021./2022. , školska/klupska razina

Igra svjetla

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Računalo	1 kom
Mikroupravljačko sučelje prema vlastitom izboru	1 kom
Eksperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti - duljine 10 cm do 20cm	1 kom
Svjetleće diode iste boje promjera 5mm	2 kom
Otpornik 330 Ohma	2 kom
Tipkalo	2 kom
Univerzalni mjerni instrument	1 kom

Zadatak:

Tvoj zadatak je na eksperimentalnoj pločici izraditi elektronički uređaj za upravljanje jačinom svjetla svjetlećih dioda.

Na eksperimentalnu pločicu postavi 2 (dvije) svjetleće diode i pripadajuće otpornike te dva tipkala. Zatim sve elemente spoji s tvojim mikroupravljačkim sučeljem. Provjeri jesu li svi elementi ispravno povezani s mikroupravljačkim sučeljem i tek tada poveži sučelje s računalom i ako je potrebno s izvorom napajanja.

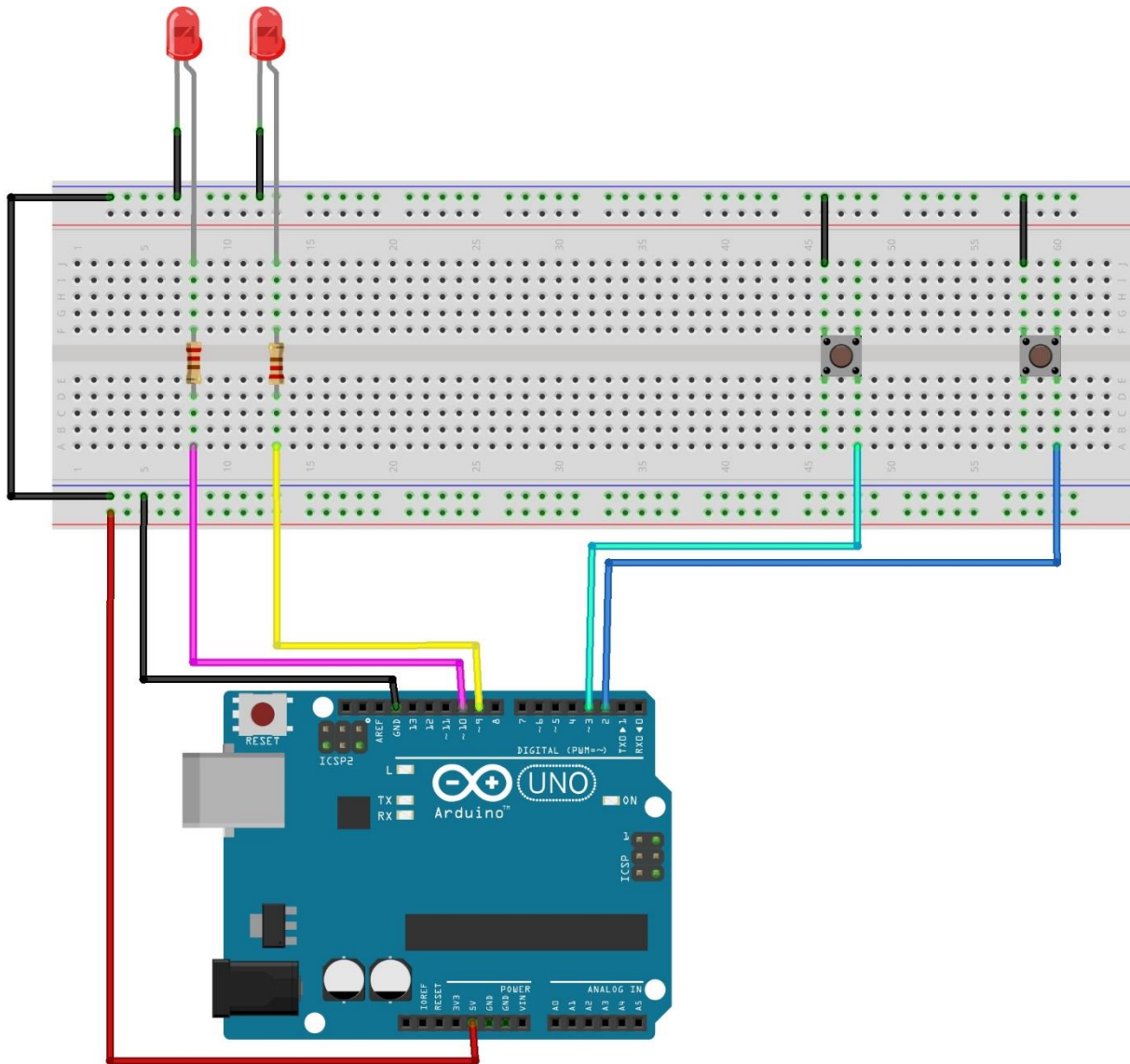
Napiši program za upravljanje tvojim elektroničkim uređajem prema sljedećim zahtjevima:

- Program se pokreće pokretanjem mikroupravljačkog sučelja.
- Za vrijeme dok je prvo tipkalo pritisnuto, a drugo otpušteno, druga svjetleća dioda je ugašena, a prva svjetleća dioda mijenja jačinu svjetlosti od nule do najveće moguće vrijednosti koja se može postaviti na tom izvodu mikroupravljačkog sučelja.
- Korak promjene jačine svjetlosti iznosi 1, a vremenski razmak između koraka promjene jačine svjetlosti iznosi 10 milisekundi.
- Kad jačina svjetlosti dosegne najveću moguću vrijednost za taj izvod mikroupravljačkog sučelja, jačina svjetlosti se smanjuje do 0, također korakom 1 i vremenskim razmakom promjene koraka od 10 milisekundi.
- Nakon što je jačina svjetlosti pala na 0, ponovo se povećava. I to se ponavlja dok god je prvo tipkalo pritisnuto.
- Ako je drugo tipkalo pritisnuto, a prvo ne, prva svjetleće dioda se gasi, a druga nastavlja s promjenama jačine svjetla s istim vrijednostima koje su vrijedile i za prvo tipkalo u trenutku pritiska na tipkalo.
- Ako su oba tipkala pritisnuta ili ako niti jedno nije pritisnuto, obje svjetleće diode su ugašene.

Kad dovršiš program i ako je sve u redu pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.



Montažna shema:



fritzing

Jedno od mogućih programskih rješenja:

```
int led1 = 9;  
int led2 = 10;  
int tipk1 = 2;  
int tipk2 = 3;  
int pauza = 10;  
int svjetlo = 0;  
int korak = 1;
```

```
void setup() {  
  pinMode(led1, OUTPUT);
```



```
pinMode(led2, OUTPUT);  
pinMode(tipk1, INPUT_PULLUP);  
pinMode(tipk2, INPUT_PULLUP);  
Serial.begin(9600);  
}  
  
void loop() {  
while((digitalRead(tipk1)==LOW)&&(digitalRead(tipk2)==HIGH)){  
  analogWrite(led1, svjetlo);  
  analogWrite(led2, 0);  
  Serial.println(svjetlo, korak);  
  svjetlo = svjetlo + korak;  
  if (svjetlo <= 0 || svjetlo >= 255) {  
    Serial.println(svjetlo, korak);  
    korak = -korak;  
  }  
  delay(pauza);  
}  
while((digitalRead(tipk1)==HIGH)&&(digitalRead(tipk2)==LOW)){  
  analogWrite(led1, 0);  
  analogWrite(led2, svjetlo);  
  Serial.println(svjetlo, korak);  
  svjetlo = svjetlo + korak;  
  if (svjetlo <= 0 || svjetlo >= 255) {  
    korak = -korak;  
  }  
  delay(pauza);  
}  
analogWrite(led1, 0);  
analogWrite(led2, 0);  
}
```

NATJECANJE
MLADIH
TEHNIČARA



65. natjecanje mladih tehničara; 2022./2023., školska/klupska razina

Model prometnog raskrižja

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Mikroupravljačko sučelje prema vlastitom izboru	1 kom
Ekperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti - duljine 5 do 10 cm	1 kom
Crvena svjetleća dioda promjera 5mm	2 kom
Žuta svjetleća dioda promjera 5 mm	1 kom
Zelena svjetleća dioda promjera 5 mm	2 kom
Bijela svjetleća dioda promjera 5 mm	1 kom
Otpornik 330 Ohma (po potrebi, zavisno od mikroupravljačkog sučelja)	6 kom
Fotootpornik	1 kom
Tipkalo	1 kom

Zadatak:

Tvoj zadatak je na eksperimentalnoj pločici izraditi model prometnog raskrižja. Model raskrižja sastoji se od dva semafora, jednog za vozila i jednog za pješake te jednom uličnom svjetiljkom. Uz to, na modelu raskrižja se nalaze tipkalo i fotootpornik.

Na eksperimentalnu pločicu, jednu pokraj druge, postavi tri svjetleće diode, crvenu, žutu i zelenu. One predstavljaju semafor za vozila. Zatim, jednu pokraj druge, postavi dvije svjetleće diode, crvenu i zelenu. One predstavljaju semafor za pješake. Zasebno postavi bijelu svjetleću diodu. Ta dioda predstavlja rasvjetu prometnog raskrižja. Sada postavi na eksperimentalnu pločicu fotootpornik. Na kraju, na eksperimentalnu pločicu postavi i tipkalo. Sve elemente spoji s mikroupravljačkim sučeljem.

Provjeri ispravnost spojeva i, ako je sve u redu, spoji sučelje s računalom.

Napiši program za upravljanje radom modela prometnog raskrižja prema sljedećim zahtjevima: Model semafora naizmjenično propušta automobile i pješake. Na semaforu za pješake smije svijetliti zeleno svjetlo samo onda kada je na semaforu za vozila upaljeno crveno svjetlo. U svim ostalim kombinacijama na semaforu za pješake mora svijetliti crveno svjetlo!

Ovdje su navedena vremena svijetljenja određenog svjetla na semaforu za vozila:

- Crveno svjetlo -> 5 sekundi
- Crveno i žuto svjetlo -> 1 sekunda
- Zeleno svjetlo -> 5 sekundi
- Samo žuto svjetlo -> 2 sekunde

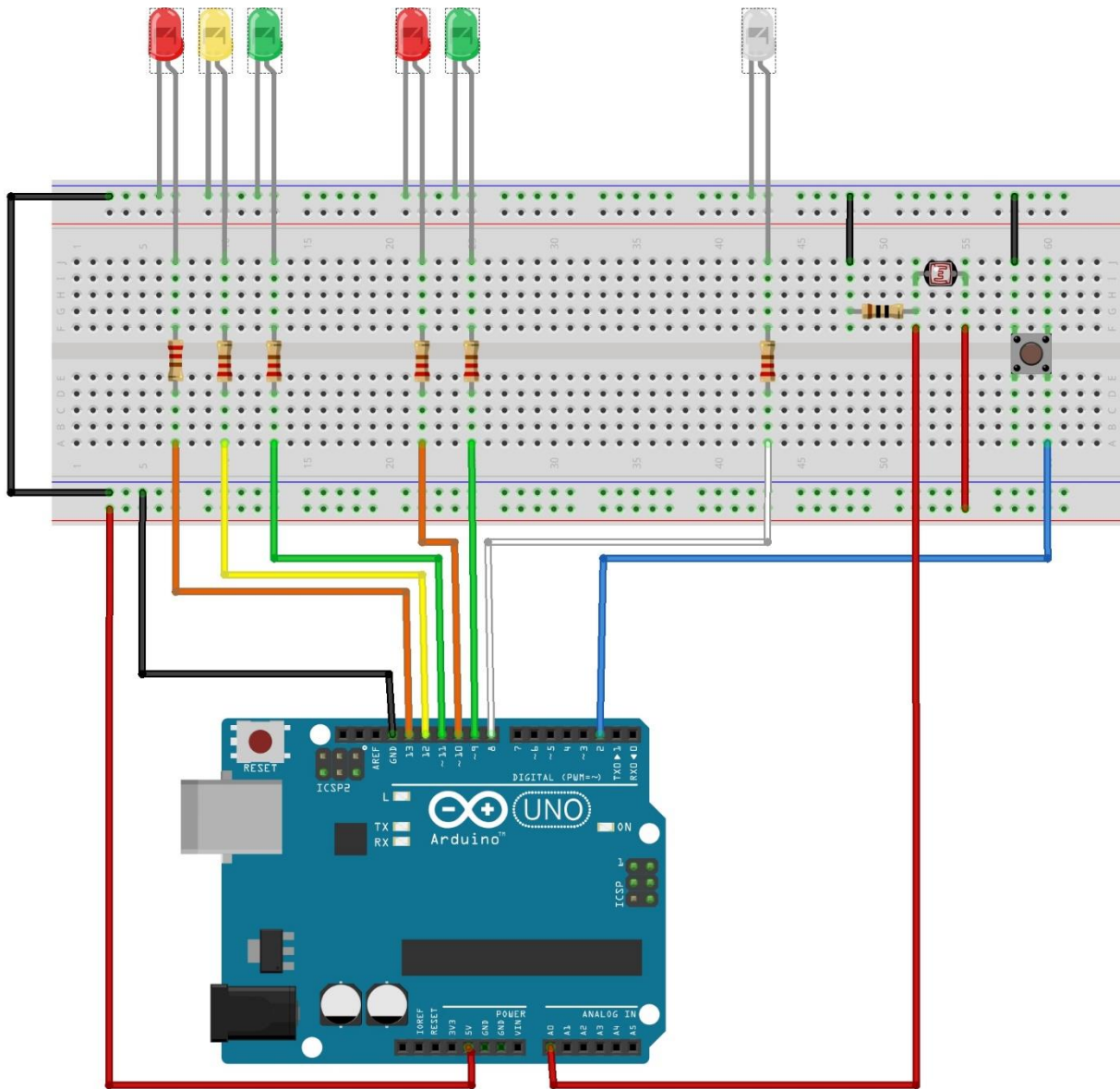
Svjetla na semaforu za pješake ovise o radu semafora za vozila.

Bijela svjetleća dioda predstavlja rasvjetu prometnog raskrižja i upaljena je kada je raskrižje u mraku. U našem slučaju, bijela se svjetleća dioda pali kada je fotootpornik zasjenjen. Nivo zasjenjenosti odredi svojevrijedno, tako da svjetleća dioda zasvijetli kada fotootpornik zakloniš rukom. Program se pokreće pritiskom na tipkalo i nakon prvog pritiska na tipkalo neprestano se ponavlja. Program se ponovo pokreće tek ponovnim pokretanjem mikroupravljačkog sučelja i pritiskom na tipkalo.

Kad dovršiš program i ako je sve u redu pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.



Montažna shema:



fritzing

TEHNIČARA



Jedno od mogućih programskih rješenja:

```
long vrijeme=millis();  
long pocvr=vrijeme;  
int crv_a=13;  
int zut_a=12;  
int zel_a=11;  
int crv_p=10;  
int zel_p=9;  
int bij=8;  
int foto=analogRead(A0);  
int prag=50; //podesiti prema vrijednostima očitavanja foto senzora
```

```
void setup() {  
  pinMode(crv_a,OUTPUT);  
  pinMode(zut_a,OUTPUT);  
  pinMode(zel_a,OUTPUT);  
  pinMode(crv_p,OUTPUT);  
  pinMode(zel_p,OUTPUT);  
  pinMode(bij,OUTPUT);  
  pinMode(2,INPUT_PULLUP);  
  digitalWrite(crv_a,LOW);  
  digitalWrite(zut_a,LOW);  
  digitalWrite(zel_a,LOW);  
  digitalWrite(crv_p,LOW);  
  digitalWrite(zel_p,LOW);  
  digitalWrite(bij,LOW);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  pocvr=millis();  
  while((pocvr+5000)>millis()){  
    digitalWrite(crv_a,HIGH);  
    digitalWrite(zut_a,LOW);  
    digitalWrite(zel_a,LOW);  
    digitalWrite(crv_p,LOW);  
    digitalWrite(zel_p,HIGH);  
    foto=analogRead(A0);  
    if(foto<prag){  
      digitalWrite(bij,HIGH);  
    }  
    else{  
      digitalWrite(bij,LOW);  
    }  
  }  
}
```

```
pocvr=millis();  
while((pocvr+1000)>millis()){  
  digitalWrite(crv_a,HIGH);
```



```
digitalWrite(zut_a,HIGH);  
digitalWrite(zel_a,LOW);  
digitalWrite(crv_p,HIGH);  
digitalWrite(zel_p,LOW);  
foto=analogRead(A0);  
if(foto<prag){  
    digitalWrite(bij,HIGH);  
}  
else{  
    digitalWrite(bij,LOW);  
}  
}  
  
pocvr=millis();  
while((pocvr+5000)>millis()){  
    digitalWrite(crv_a,LOW);  
    digitalWrite(zut_a,LOW);  
    digitalWrite(zel_a,HIGH);  
    digitalWrite(crv_p,HIGH);  
    digitalWrite(zel_p,LOW);  
    foto=analogRead(A0);  
    if(foto<prag){  
        digitalWrite(bij,HIGH);  
    }  
    else{  
        digitalWrite(bij,LOW);  
    }  
}  
  
pocvr=millis();  
while((pocvr+2000)>millis()){  
    digitalWrite(crv_a,LOW);  
    digitalWrite(zut_a,HIGH);  
    digitalWrite(zel_a,LOW);  
    digitalWrite(crv_p,HIGH);  
    digitalWrite(zel_p,LOW);  
    foto=analogRead(A0);  
    if(foto<prag){  
        digitalWrite(bij,HIGH);  
    }  
    else{  
        digitalWrite(bij,LOW);  
    }  
}  
}  
}
```



66. natjecanje mladih tehničara; 2023./2024., školska/klupska razina

Model upravljanja elektromotornim pogonom vozila

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Mikroupravljačko sučelje prema vlastitom izboru	1 kom
Ekperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti - duljine 5 do 10 cm	1 kom
Crvena svjetleća dioda promjera 5mm	1 kom
Zelena svjetleća dioda promjera 5 mm	1 kom
Bijela svjetleća dioda promjera 5 mm	1 kom
Otpornik 330 Ω (ako je potrebno, zavisno od mikroupravljačkog sučelja)	3 kom
Fotootpornik	1 kom
Otpornik 10 K Ω	1 kom
Tipkalo	2 kom
Otpornik 10 K Ω (uz tipkala, ako je potrebno)	2 kom

Zadatak:

Tvoj zadatak je izraditi model upravljanja elektromotornim pogonom vozila. Model se sastoji od dva tipkala, tri svjetleće diode (bije, zelene i crvene) te fotootpornika. Crvena i zelena svjetleća dioda predstavljaju smjer vrtnje elektromotora. Bijela svjetleća dioda predstavlja svjetla vozila.

Priprema elemenata:

Na eksperimentalnu pločicu, jednu pokraj druge, postavi bijelu, crvenu i zelenu svjetleću diodu. One će predstavljati smjer vrtnje elektromotora (ili smjer kretanja vozila) i svjetla vozila. Nakon toga postavi pripadajuće otpornike za svaku svjetleću diodu. Zatim, jedno pored drugog postavi dva tipkala i po potrebi zaštitne otpornike za tipkala. Na kraju, postavi na eksperimentalnu pločicu fotootpornik i pripadajući zaštitni otpornik.

Sve elemente spoji s mikroupravljačkim sučeljem.

Provjeri ispravnost spojeva i, ako je sve u redu, spoji sučelje s računalom.

Izrada programa za model upravljanja elektromotornim pogonom:

Na početku programa sve tri svjetleće diode se upale na jednu sekundu i zatim se ugase. To se odvija samo jednom. Nakon toga kreće izvršavanje programa prema sljedećim zahtjevima:

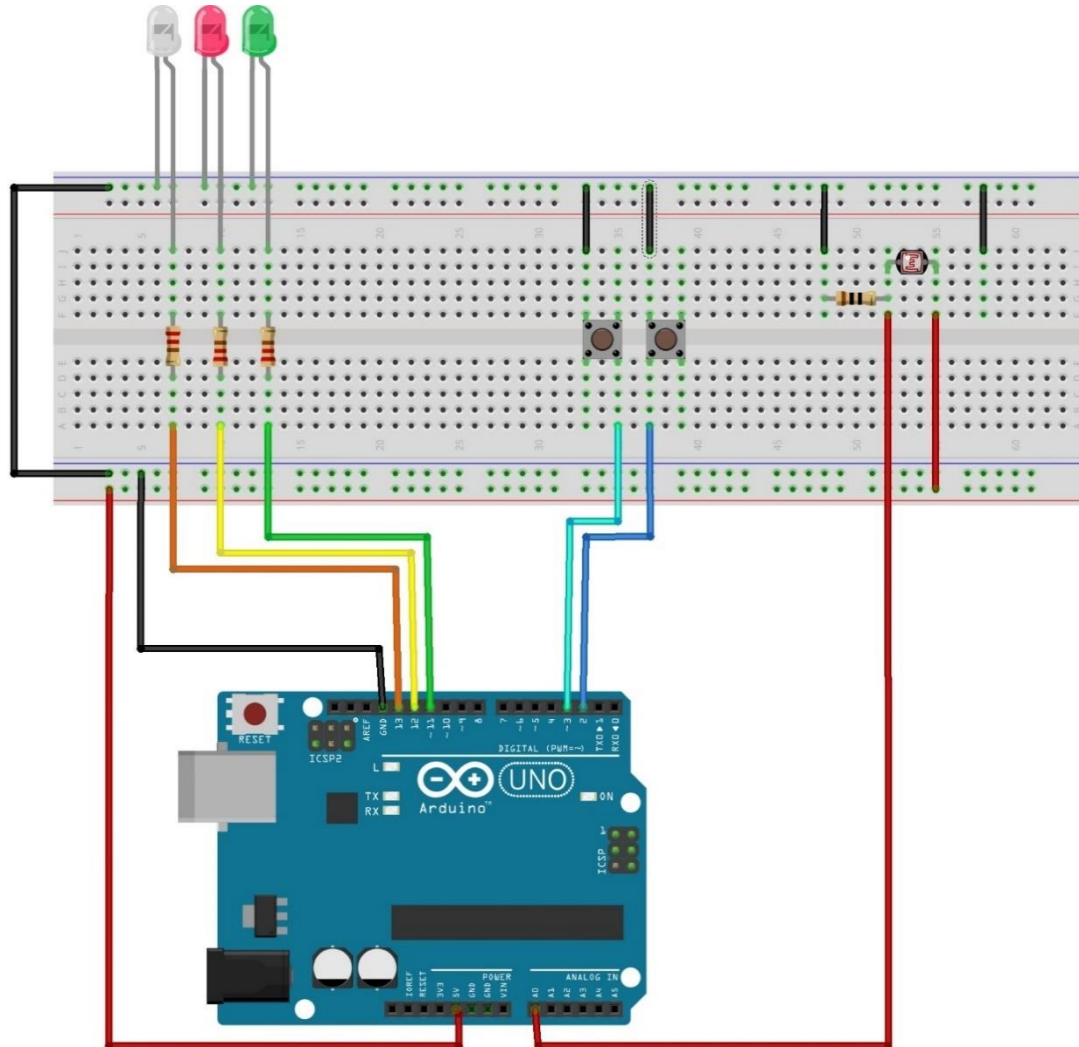
- Ako ni jedno tipkalo nije pritisnuto, ili ako su oba tipkala pritisnuta, crvena i zelena svjetleća dioda ne svijetle. *Sada elektromotor ne bi radio i vozilo bi mirovalo.*
- Ako je pritisnuto samo prvo tipkalo, crvena svjetleća dioda svijetli, a zelena je ugašena. *Sada bi se vozila kretalo prema naprijed.*
- Ako je pritisnuto samo drugo tipkalo, zelena svjetleća dioda svijetli, a crvena je ugašena. *Sada bi se vozila kretalo unazad.*
- Bez obzira na stanje tipkala, ako je fotootpornik osvjetljen, bijela svjetleća dioda je ugašena. *To predstavlja dnevno svjetlo, glavna svjetla na vozilu su ugašena.*
- Bez obzira na stanje tipkala, ako je fotootpornik zaklonjen od svjetla, bijela svjetleća dioda svijetli. *To predstavlja noć, glavna svjetla na vozilu su upaljena.*

Program se pokreće pokretanjem mikroupravljačkog sučelja i neprestano se ponavlja. Program se ponovo pokreće tek ponovnim pokretanjem mikroupravljačkog sučelja.



Kad dovršiš program i ako je sve u redu pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.

Montažna shema:



fritzing

Jedno od mogućih programskih rješenja:

```
int bij=13;
int mot_l=12;
int mot_d=11;
int tipk_1=2;
int tipk_2=3;
int foto=analogRead(A0);
int prag=50; //podesiti prema vrijednostima očitavanja foto senzora
```

```
void setup() {
  pinMode(bij,OUTPUT);
  pinMode(mot_l,OUTPUT);
  pinMode(mot_d,OUTPUT);
}
```



```
pinMode(tipk_1,INPUT_PULLUP);
pinMode(tipk_2,INPUT_PULLUP);
digitalWrite(bij,HIGH);
digitalWrite(mot_l,HIGH);
digitalWrite(mot_d,HIGH);
delay(1000);
digitalWrite(bij,LOW);
digitalWrite(mot_l,LOW);
digitalWrite(mot_d,LOW);
delay(1000);
}

void loop() {
if ((digitalRead(tipk_1)==LOW) && (digitalRead(tipk_2)==HIGH)){
digitalWrite(mot_l,HIGH);
digitalWrite(mot_d,LOW);
foto=analogRead(A0);
if(foto<prag){
digitalWrite(bij,HIGH);
}
else{
digitalWrite(bij,LOW);
}
}
else if((digitalRead(tipk_1)==HIGH) && (digitalRead(tipk_2)==LOW)){
digitalWrite(mot_l,LOW);
digitalWrite(mot_d,HIGH);
foto=analogRead(A0);
if(foto<prag){
digitalWrite(bij,HIGH);
}
else{
digitalWrite(bij,LOW);
}
}
else{
digitalWrite(mot_l,LOW);
digitalWrite(mot_d,LOW);
foto=analogRead(A0);
if(foto<prag){
digitalWrite(bij,HIGH);
}
else{
digitalWrite(bij,LOW);
}
}
}
```



62. natjecanje mladih tehničara; 2019./2020., županijska razina

Model digitalne vage

Oprema, pribor i materijal potrebni za izvršenje zadatka:

- Mikroupravljačko sučelje prema vlastitom izboru
- Programator (ako je zaseban)
- Računalo s instaliranim softverskim paketom za programiranje odabranog sučelja
- Eksperimentalna pločica
- LCD 16*2 ili 20*4 sa zalemljenim kontaktima za učvršćivanje na eksperimentalnu pločicu
- Potencijometar 10 kOhma
- Senzor mase (20 kg) i AD pretvornik
- Tipkala, 2 komada
- Otpornik 330 Ohma
- Spojne žice za spajanje komponenti - duljine otprilike 10 cm i 20 cm, jednih i drugih po 20 kom
- Produžni kabel s minimalno 3 utičnice, minimalne duljine 3 m
- Mala sjekača kliješta za elektroniku
- Kliješta za uklanjanje izolacije s električnih vodiča
- Odvijač ravni

Zadatak:

Tvoj zadatak je na eksperimentalnoj pločici izraditi model digitalne vage.

Na eksperimentalnu pločicu postavi LCD modul s pripadajućim elektroničkim dijelovima (potencijometar i otpornik), 2 (dva) tipkala i AD pretvornik. Zatim sve elemente i module spoji s tvojim mikrokontrolerskim sučeljem, a AD pretvornik sa senzorom mase. Provjeri jesu li svi elementi ispravno povezani a posebno obrati pažnju na ispravan polaritet elemenata i modula Tek ako je sve ispravno povezano, poveži sučelje s računalom i ako je potrebno s izvorom napajanja.

Napiši program za model digitalne vage prema sljedećim zahtjevima:

Model digitalne vage prikazuje podatke na zaslonu LCD modula. Digitalna vaga očitava 5 mjerenja te izračunava ukupnu masu, prosječnu masu, najmanju i najveću masu.

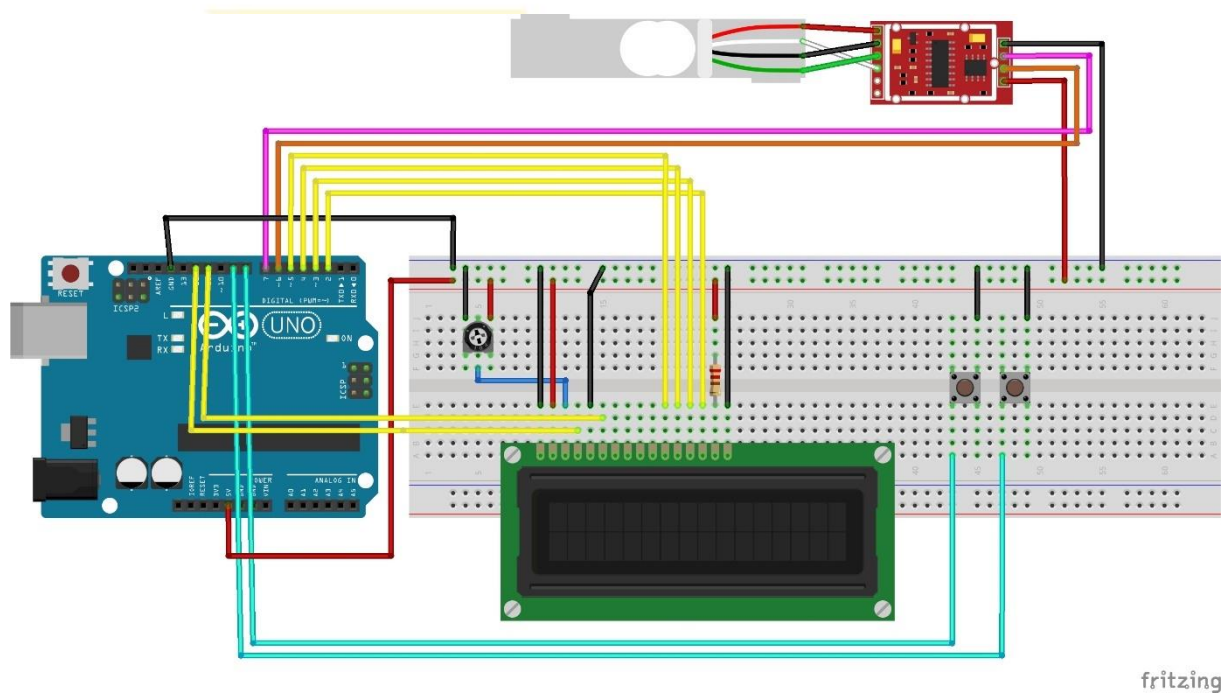
Kada se program pokrene vaga se mora moći umjeriti (kalibrirati) na vrijednost 0, a na zaslonu LCD modula mora pisati uputa za rukovatelja, u prvom redu ekrana tekst „**Tipka 1**“ a u drugom redu „**Tara=0**“ i „**Start**“. Pritiskom na tipku 1 procedura za mjerenje mase se pokreće i umjerava. Na zaslonu LCD modula mora u prvom redu pisati uputa: „**Tpk2->mjerenje**“, a u drugom redu **redni broj mjerenja**. Procedura se sastoji iz 5 uzastopnih mjerenja mase. Da bi se očitala masa potrebno je svaki puta pritisnuti tipku 2. Poslije svakog pritiska na tipku 2 procedura izvršava mjerenje mase i na zaslon LCD modula u prvom redu ispisuje samo **trenutno izmjerenu masu**. Nakon jedne sekunde pauze, procedura nastavlja sa sljedećim mjerenjem, a na zaslonu LCD modula prikazuje se opet poruka za mjerenje (u prvom redu tekst „**Tpk2->mjerenje**“, a u drugom redu **redni broj mjerenja**). Nakon petog mjerenja procedura na zaslonu LCD modula mora ispisati **ukupnu, prosječnu, minimalnu i maksimalnu masu**. Ispis mora biti pregledan kako bi bilo jasno vidljivo o kojim se podacima radi. Ako koristiš dvoredni LCD modul tada neka procedura u prvom prikazu ispiše ukupnu masu u prvom redu, prosječnu u drugom te nakon jedne sekunde pauze u drugom prikazu minimalnu masu u prvom a maksimalnu u drugom



redu. Ako koristiš četveroredni LCD modul, neka procedura svaki podatak ispiše u zasebnom redu, ukupnu masu u prvom redu, prosječnu u drugom, minimalnu u trećem i maksimalnu u četvrtom redu zaslona.

Sve vrijednosti masa se ispisuju na 2 decimale s oznakom za grame.

Montažna shema:



Jedno od mogućih programskih rješenja:

```
//Županijsko natjecanje MT 2019/2020
#include "HX711.h"
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define DOUT 7
#define CLK 6
int reset_tpk = 8;
int vaga_tipk = 9;
float minimum = 1000;
float maksimum = 0;
float ukupno=0;
float prosjek = 0;
int brojac = 0;
int kontrola = 0;
HX711 scale;
float kal_f = 205; // Faktor kalibracije ovisi o odabranom senzoru mase
float masa;
```

```
void setup() {  
  pinMode(reset_tpk,INPUT_PULLUP);  
  pinMode(vaga_tipk,INPUT_PULLUP);  
  lcd.begin(16,2);  
  Serial.begin(9600);  
  scale.begin(7, 6);  
  lcd.setCursor(0,0);  
  while(digitalRead(reset_tpk)==1){  
    lcd.setCursor(5,0);  
    lcd.print("Tipka 1");  
    lcd.setCursor(0,1);  
    lcd.print("Tara=0 && Start");  
  }  
  scale.set_scale(kal_f);  
  scale.tare();  
  lcd.clear();  
}  
void loop() {  
  minimum = 1000;  
  maksimum = 0;  
  ukupno=0;  
  prosjek = 0;  
  brojac = 0;  
  kontrola = 1;  
  for(brojac=1;brojac<6;brojac++){  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Tpk2->mjerenje");  
    lcd.setCursor(0,1);  
    lcd.print(brojac);  
    lcd.setCursor(1,1);  
    lcd.print(".");  
    lcd.setCursor(10,1);  
    lcd.print(ukupno,2);  
    kontrola==1;  
    while(digitalRead(vaga_tipk)==1){  
      kontrola=1;  
    }  
    if((digitalRead(vaga_tipk)==0)&&(kontrola==1)){  
      kontrola==0;  
      masa = scale.get_units(10), 5;  
      if (masa < 0)  
      {  
        masa = 0.00;  
      }  
      ukupno=ukupno+masa;  
      if(minimum>masa){  
        minimum=masa;  
      }  
      if (maksimum<masa){
```



```
    maksimum=masa;
  }
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Masa: ");
  Serial.print("Masa: ");
  lcd.setCursor(8,0);
  lcd.print(masa,2);
  Serial.print(masa,2);
  lcd.setCursor(14,0);
  lcd.print("g");
  Serial.println(" g");
  delay(1000);
  while(digitalRead(reset_tpk)==0){
    lcd.clear();
    Serial.println("Reset");
    lcd.setCursor(0,0);
    lcd.println("Reset");
    scale.tare();
    lcd.clear();
    lcd.println(ukupno,2);
    brojac=0;
    lcd.setCursor(0,1);
    lcd.println(brojac);
  }
}
else{
  kontrola=1;
}
delay(1000);
}
lcd.clear();
lcd.setCursor(0,0);
lcd.println("Ukupno: ");
lcd.setCursor(10,0);
lcd.println(ukupno,2);
lcd.setCursor(15,0);
lcd.println("g");
lcd.setCursor(0,1);
lcd.println("Prosjek: ");
lcd.setCursor(10,1);
lcd.println(ukupno/5,2);
lcd.setCursor(15,1);
lcd.println("g");
while(digitalRead(vaga_tipk)==1){
}
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.println("Minimum: ");
```



```
lcd.setCursor(10,0);  
lcd.println(minimum,2);  
lcd.setCursor(15,0);  
lcd.println("g");  
lcd.setCursor(0,1);  
lcd.println("Maksimum: ");  
lcd.setCursor(10,1);  
lcd.println(maksimum,2);  
lcd.setCursor(15,1);  
lcd.println("g");  
while(digitalRead(vaga_tipk)==1){  
}  
delay(1000);  
kontrola=0;  
delay(1000);  
scale.set_scale(kal_f);  
scale.tare();  
}
```



NATJECANJE
MLADIH
TEHNIČARA



63. natjecanje mladih tehničara; 2020./2021., županijska razina

Model mosne vage

Mosna vaga je vaga na kojoj se mjeri masa vozila. Takve vage imaju nekoliko primjena. Najčešće se koriste prilikom kontrole utovarene robe na skladištu. No primjenjuju se i kod osobnih automobila. Naime, osobni automobili moraju imati pravilno raspoređeno opterećenje na obje osovine, prednjoj i stražnjoj. To se opterećenje provjerava mosnom vagom. Najprije se izvažuje ukupna masa automobila, a zatim se važe opterećenje na prednjoj te na stražnjoj osovini. Dobivena masa ukupnog automobila mora biti jednaka zbroju masa na prednjoj i na stražnjoj osovini. Također, mase na prednjoj i stražnjoj osovini moraju biti jednake, uz dozvoljeno malo odstupanje. Takva se mjerenja moraju provoditi u slučaju da su na automobilu izvedene veće preinake koje bi mogle rezultirati nejednakim opterećenjem na osovinama. Na primjer, prilikom ugradnje plinskih boca.

Oprema, pribor i materijal potrebni za izvršenje zadatka:

- Mikroupravljačko sučelje prema vlastitom izboru	1 kom
- Programator (ako je zaseban)	1 kom
- Kabel za spajanje mikroupravljačkog sučelja s računalom	1 kom
- Računalo s instaliranim softverskim paketom za programiranje odabranog sučelja	1 kom
- Univerzalni mjerni instrument	1 kom
- Produžni kabel s minimalno 3 utičnice, minimalne duljine 3 m	1 kom
- Eksperimentalna pločica	1 kom
- Set spojnih žica za spajanje komponenti – razne duljine (od 1 do 10 cm)	1 kom
- Crvena svjetleća dioda promjera 5 mm	1 kom
- Zelena svjetleća dioda promjera 5 mm	1 kom
- Otpornik 330 ohma	2 kom
- Senzor mase s AD pretvornikom	1 kom
- Tipkalo (po potrebi i po jedan otpornik za svako tipkalo)	2 kom
- Model auta	1 kom
- Mala sjekača kliješta za elektroniku	
- Kliješta za uklanjanje izolacije s električnih vodiča	
- Odvijač ravni	
- Modelarski nožić – skalpel	
- Obična ili tehnička olovka	
- Ravnalo	

Priprema elemenata:

Za izradu tehničke tvorevine će ti trebati eksperimentalna pločica, mikroupravljačko sučelje, dva tipkala, crvena i zelena svjetleća dioda, pripadajući otpornici te senzor mase s AD pretvornikom. Zavisno od podešavanja načina rada tipkala, možda ćeš morati upotrijebiti i otpornike za ta dva tipkala. Uz to će ti trebati i model auta. Prema veličini modela izraditi ćeš ploču za vaganje.

Na senzor mase pričvrsti ploču za vaganje koja mora biti dovoljno velika da automobil koji koristiš stane na nju. Pokraj ploče za vaganje postavi neko postolje u visini ploče za vaganje. To



može biti bilo što, nekakva kutijica, tanjurić, komadić drvene letvice, bilo što. Bitno je da to postolje ne dotiče ploču za vaganje i da je u istoj visini s njom.

To postolje služi kao oslonac za automobil kad se važe opterećenje na osovina. Kad mjeriš opterećenje na osovina, postavi automobil s jednom osovinom na ploču za vaganje a s drugom na to postolje.

Elektronski dio modela ćeš izraditi na eksperimentalnoj pločici. Na pločicu postavi tipkala (po potrebi i njihove otpornike) i svjetleće diode s pripadajućim otpornicima. Zatim spoji senzor mase s AD pretvornikom te AD pretvornik poveži s mikrokontrolerskim sučeljem. Spoji tipkala i svjetleće diode s mikrokontrolerskim sučeljem, Spoji sučelje s računalom i to je to.

Opis funkcija elemenata:

Prvo tipkalo služi za postavljanje tara vrijednosti (vrijednost koju senzor mase očitava kad je vaga prazna). Pritiskom na to tipkalo program postavlja početnu vrijednost vaganja na 0. Drugo tipkalo služi za pohranjivanje očitanih masa. Svako očitavanje popraćeno je kratkim paljenjem zelene i istovremenim gašenjem crvene svjetleće diode.

Zadatak:

Izradi program kojim ćeš očitavati mase vozila te ih usporediti kako bi se vidjelo je li vozilo ujednačenih opterećenja na obje osovine.

Na početku programa, na ekranu računala ispisuje se poruka "**Pritisni tipku1 za postavljanje tara mase na 0**". Pritiskom na prvo tipkalo postavlja se masa prazne vage na 0. Na ekranu računala se ispisuje ta vrijednost u gramima.

Nakon jedne sekunde se pali crvena svjetleća dioda, zelena ostaje ugašena, a na ekranu se ispisuje poruka "Izmjeri masu vozila - tipkalo 2". Tada postavi vozilo na vagu te pritiskom na drugo tipkalo pohrani izmjerenu masu u varijablu predviđenu za to. Prilikom pritiska na tipkalo gasi se crvena svjetleća dioda, a pali zelena.

Nakon jedne sekunde gasi se zelena svjetleća dioda, a pali crvena. Istovremeno se na ekranu ispisuje poruka "**Izmjeri masu na prvoj prednjoj - tipkalo 2**". Ponovnim pritiskom na drugo tipkalo očitana se vrijednost pohranjuje u za to određenu varijablu. Prilikom pritiska na tipkalo gasi se crvena svjetleća dioda, a pali zelena.

Nakon jedne sekunde gasi se zelena svjetleća dioda, a pali crvena.

Sada se na ekranu ispisuje poruka "**Izmjeri masu na stražnjoj osovini - tipkalo 2**". Ponovnim pritiskom na drugo tipkalo očitana se vrijednost pohranjuje u za to određenu varijablu. Prilikom pritiska na tipkalo gasi se crvena svjetleća dioda, a pali zelena.

Sada imaš pohranjene tri vrijednosti ili tri mase, masu ukupnog vozila te mase na prednjoj i stražnjoj osovini.

Na kraju ispiši sve te tri mase na ekranu računala.

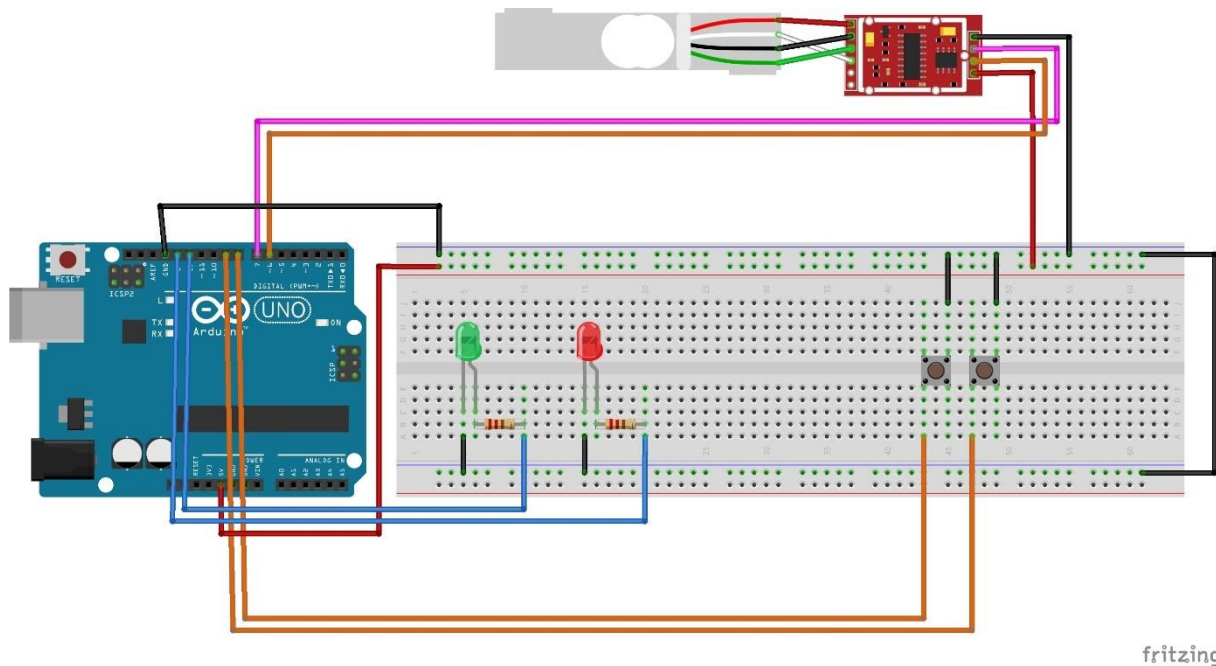
Ako je masa ukupnog vozila jednaka zbroju masa na prednjoj i stražnjoj osovini na ekranu računala ispiši poruku "**Mase se podudaraju**".

Ako je masa ukupnog vozila različita od zbroja masa na prednjoj i stražnjoj osovini na ekranu računala ispiši poruku "**Mase različite**".



Nakon 5 sekundi zelena svjetleća dioda se gasi, a pali crvena. Na ekranu se ispisuje tekst **"Novo mjerenje"** te se pritiskom na prvo tipkalo proces mjerenja ponavlja... Kad dovršiš program i ako je sve u redu, pozovi ocjenjivačko povjerenstvo. Nakon ocjenjivanja zadatka, ako imaš još vremena, pokušaj napraviti funkcionalnu nadogradnju uređaja.

Montažna shema:



Jedno od mogućih programskih rješenja:

```
#include "HX711.h"
#define DOUT 7
#define CLK 6
int reset_tpk = 8;
int tipk=9;
int ukupno=0;
int prva_osovina=0;
int druga_osovina=0;
int crv_led=13;
int zel_led=12;
```

```
HX711 scale;
float kal_f = 199; // Faktor kalibracije ovisi o odabranom senzoru mase
int masa;
```

```
void setup() {
  pinMode(reset_tpk,INPUT_PULLUP);
  pinMode(tipk,INPUT_PULLUP);
  pinMode(crv_led,OUTPUT);
  pinMode(zel_led,OUTPUT);
```




```
Serial.begin(9600);
scale.begin(7, 6);
while(digitalRead(reset_tpk)==1){
  Serial.println("Reset");
  delay(100);
  Serial.print("Tpk1->tara=0");
  Serial.println("Pritisni tipku1 za postavljanje tara mase na 0");
}
scale.set_scale(kal_f);
scale.tare();
}

void loop() {
  digitalWrite(crv_led,LOW);
  digitalWrite(zel_led,LOW);
  masa = scale.get_units(), 5;
  if (masa < 0)
  {
    masa = 0.00;
  }
  Serial.print(masa);
  Serial.println(" g");
  delay(1000);
  digitalWrite(crv_led,HIGH);
  while(digitalRead(tipk)==1){
    Serial.println("Izmjeri masu vozila");
    ukupno=scale.get_units(), 5;
  }
  digitalWrite(crv_led,LOW);
  digitalWrite(zel_led,HIGH);
  delay(1000);
  digitalWrite(crv_led,HIGH);
  digitalWrite(zel_led,LOW);
  while(digitalRead(tipk)==1){
    Serial.println("Izmjeri masu na prvoj osovini");
    prva_osovina=scale.get_units(), 5;
  }
  digitalWrite(crv_led,LOW);
  digitalWrite(zel_led,HIGH);
  delay(1000);
  digitalWrite(crv_led,HIGH);
  digitalWrite(zel_led,LOW);
  while(digitalRead(tipk)==1){
    Serial.println("Izmjeri masu na drugoj osovini");
    druga_osovina=scale.get_units(), 5;
  }
  digitalWrite(crv_led,LOW);
  digitalWrite(zel_led,HIGH);
  delay(1000);
  Serial.print(ukupno);
```



```
Serial.print(" ");  
Serial.print(prva_osovina);  
Serial.print(" ");  
Serial.println(druga_osovina);  
if(ukupno==(prva_osovina+druga_osovina)){  
  Serial.println("Mase se podudaraju");  
}  
else{Serial.println("Mase različite");}  
delay(5000);  
digitalWrite(crv_led,HIGH);  
digitalWrite(zel_led,LOW);  
while(digitalRead(reset_tpk)==1){  
  Serial.println("Novo mjerenje");  
  scale.tare();  
}
```



64. natjecanje mladih tehničara; 2021./2022., županijska razina

Model senzora za pomoć pri parkiranju vozila

Oprema, pribor i materijal potrebni za izvršenje zadatka:

Prijenosno računalo	1 kom
Mikroupravljačko sučelje prema vlastitom izboru s kabelom za spajanje na računalo	1 kom
Programator (ako je zaseban)	1 kom
Eksperimentalna pločica	1 kom
Set spojnih žica za spajanje komponenti - duljine 10 cm do 20 cm	1 kom
Svjetleće diode crvene boje promjera 5mm	8 kom
Svjetleća dioda zelene boje promjera 5 mm	1 kom
Otpornik 330 Ohma	9 kom
Tipkalo	1 kom
Ultrazvučni senzor	1 kom
Univerzalni mjerni instrument	1 kom
Produžni kabel s minimalno tri utičnice	1 kom

Zadatak:

Na eksperimentalnoj pločici izradi model senzora za pomoć pri parkiranju vozila.

- Na eksperimentalnu pločicu postavi 8 (osam) crvenih i jednu (1) zelenu svjetleću diodu te pripadajuće otpornike, ultrazvučni senzor i tipkalo.
- Zatim sve elemente spoji s mikroupravljačkim sučeljem.
- Provjeri jesu li svi elementi ispravno povezani s mikroupravljačkim sučeljem.
- Ako je sve u redu, poveži sučelje s računalom i, ako je potrebno, s izvorom napajanja.

Napiši program za upravljanje radom modela prema sljedećim zahtjevima:

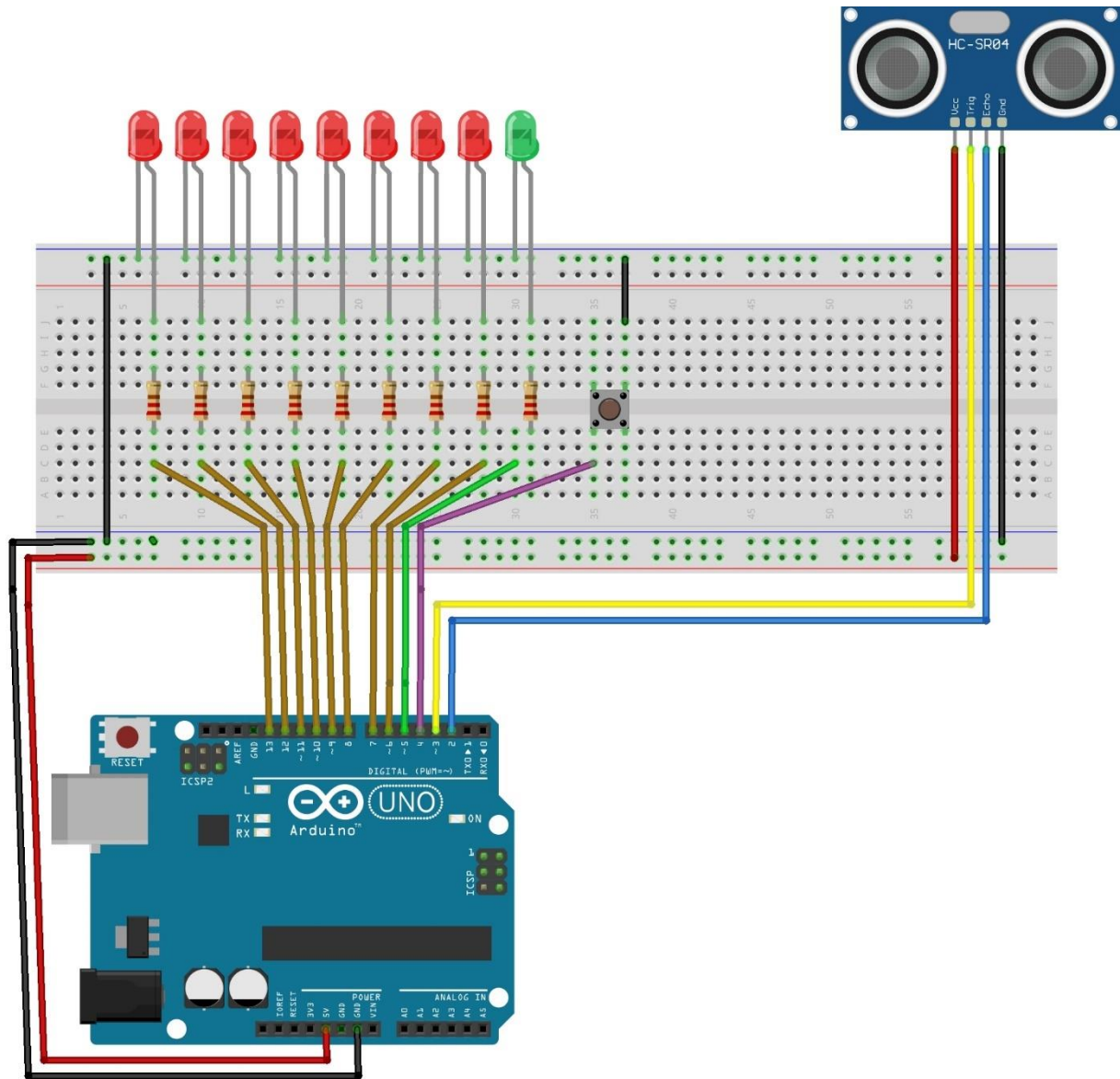
- Na početku programa se pale sve svjetleće diode i nakon jedne sekunde sve se gase. To se odvija samo jednom. *Ovaj dio programa služi kao kontrola ispravnosti spajanja svjetlećih dioda.*

Tada započinje očitavanje udaljenosti prepreke od ultrazvučnog senzora (ovaj dio programa odvija se neprestano):

- Ako je udaljenost neke prepreke od ultrazvučnog senzora veća od 50 cm, sve crvene svjetleće diode su ugašene, a zelena svjetleća dioda je upaljena.
- Pri udaljenosti od 50 cm gasi se zelena svjetleća dioda, a pali se prva crvena svjetleća dioda.
- Kako se udaljenost od prepreke smanjuje, redom se pale crvene svjetleće diode, jedna po jedna. Za svakih 5 cm udaljenosti, pali se jedna sljedeća svjetleća dioda.
- Ako se udaljenost povećava, diode se redom gase.
- Ako udaljenost prijeđe 50 cm, sve se crvene svjetleće diode gase, a pali se zelena.
- Ako je pritisnuto tipkalo, udaljenost od prepreke se ne očitava, sve crvene svjetleće diode su ugašene, a zelena svjetleća dioda treperi tempom od jedne desetinke sekunde (jednu desetinku sekunde je upaljena, jednu desetinku sekunde je ugašena) i to se ponavlja sve dok se ne otpusti tipkalo.



Montažna shema:



fritzing

Jedno od mogućih programskih rješenja:

```
int echo = 2;
int trig = 3;
int tipka = 4;
int vrijeme = 0;
int udaljenost= 0;
```

```
void setup() {
  pinMode(echo, INPUT);
```



```
pinMode(trig, OUTPUT);  
pinMode(13,OUTPUT);  
pinMode(12,OUTPUT);  
pinMode(11,OUTPUT);  
pinMode(10,OUTPUT);  
pinMode(9,OUTPUT);  
pinMode(8,OUTPUT);  
pinMode(7,OUTPUT);  
pinMode(6,OUTPUT);  
pinMode(5,OUTPUT);  
pinMode(tipka,INPUT_PULLUP);  
digitalWrite(5,HIGH);  
digitalWrite(6,HIGH);  
digitalWrite(7,HIGH);  
digitalWrite(8,HIGH);  
digitalWrite(9,HIGH);  
digitalWrite(10,HIGH);  
digitalWrite(11,HIGH);  
digitalWrite(12,HIGH);  
digitalWrite(13,HIGH);  
delay(1000);  
digitalWrite(5,LOW);  
digitalWrite(6,LOW);  
digitalWrite(7,LOW);  
digitalWrite(8,LOW);  
digitalWrite(9,LOW);  
digitalWrite(10,LOW);  
digitalWrite(11,LOW);  
digitalWrite(12,LOW);  
digitalWrite(13,LOW);  
delay(1000);  
Serial.begin(9600);  
}
```

```
void loop() {  
  while(digitalRead(tipka)==LOW){  
    digitalWrite(5,HIGH);  
    digitalWrite(6,LOW);  
    digitalWrite(7,LOW);  
    digitalWrite(8,LOW);  
    digitalWrite(9,LOW);  
    digitalWrite(10,LOW);  
    digitalWrite(11,LOW);  
    digitalWrite(12,LOW);  
    digitalWrite(13,LOW);  
    delay(100);  
    digitalWrite(5,LOW);  
    delay(100);  
  }  
  digitalWrite(trig, LOW);  
}
```



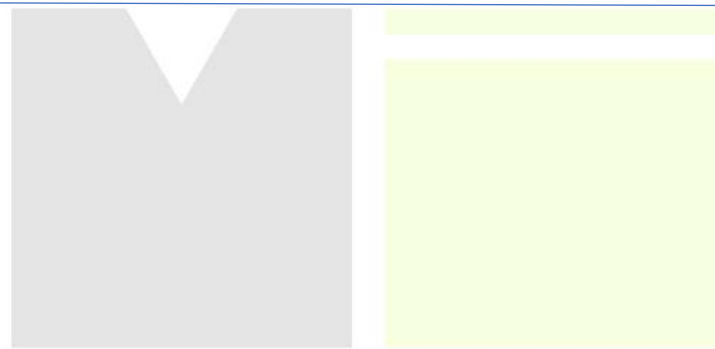
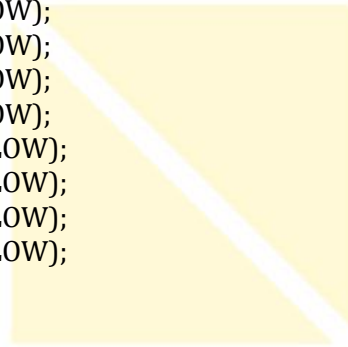
```
delayMicroseconds(2);  
digitalWrite(trig, HIGH);  
delayMicroseconds(5);  
digitalWrite(trig, LOW);  
vrijeme = pulseIn(echo, HIGH);  
udaljenost= vrijeme*0.034/2;  
Serial.println(udaljenost);  
if (udaljenost>50){  
  digitalWrite(5,HIGH);  
  digitalWrite(6,LOW);  
  digitalWrite(7,LOW);  
  digitalWrite(8,LOW);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
  digitalWrite(12,LOW);  
  digitalWrite(13,LOW);  
}  
else if(udaljenost==50){  
  digitalWrite(5,LOW);  
  digitalWrite(6,HIGH);  
  digitalWrite(7,LOW);  
  digitalWrite(8,LOW);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
  digitalWrite(12,LOW);  
  digitalWrite(13,LOW);  
}  
else if((udaljenost<50)&&(udaljenost>=45)){  
  digitalWrite(5,LOW);  
  digitalWrite(6,HIGH);  
  digitalWrite(7,HIGH);  
  digitalWrite(8,LOW);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
  digitalWrite(12,LOW);  
  digitalWrite(13,LOW);  
}  
else if((udaljenost<45)&&(udaljenost>=40)){  
  digitalWrite(5,LOW);  
  digitalWrite(6,HIGH);  
  digitalWrite(7,HIGH);  
  digitalWrite(8,HIGH);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
  digitalWrite(12,LOW);  
  digitalWrite(13,LOW);
```




```
}  
else if((udaljenost<40)&&(udaljenost>=35)){  
    digitalWrite(5,LOW);  
    digitalWrite(6,HIGH);  
    digitalWrite(7,HIGH);  
    digitalWrite(8,HIGH);  
    digitalWrite(9,HIGH);  
    digitalWrite(10,LOW);  
    digitalWrite(11,LOW);  
    digitalWrite(12,LOW);  
    digitalWrite(13,LOW);  
}  
else if((udaljenost<35)&&(udaljenost>=30)){  
    digitalWrite(5,LOW);  
    digitalWrite(6,HIGH);  
    digitalWrite(7,HIGH);  
    digitalWrite(8,HIGH);  
    digitalWrite(9,HIGH);  
    digitalWrite(10,HIGH);  
    digitalWrite(11,LOW);  
    digitalWrite(12,LOW);  
    digitalWrite(13,LOW);  
}  
else if((udaljenost<30)&&(udaljenost>=25)){  
    digitalWrite(5,LOW);  
    digitalWrite(6,HIGH);  
    digitalWrite(7,HIGH);  
    digitalWrite(8,HIGH);  
    digitalWrite(9,HIGH);  
    digitalWrite(10,HIGH);  
    digitalWrite(11,HIGH);  
    digitalWrite(12,LOW);  
    digitalWrite(13,LOW);  
}  
else if((udaljenost<25)&&(udaljenost>=20)){  
    digitalWrite(5,LOW);  
    digitalWrite(6,HIGH);  
    digitalWrite(7,HIGH);  
    digitalWrite(8,HIGH);  
    digitalWrite(9,HIGH);  
    digitalWrite(10,HIGH);  
    digitalWrite(11,HIGH);  
    digitalWrite(12,HIGH);  
    digitalWrite(13,LOW);  
}  
else if(udaljenost<20){  
    digitalWrite(5,LOW);  
    digitalWrite(6,HIGH);  
    digitalWrite(7,HIGH);  
    digitalWrite(8,HIGH);  
}
```




```
digitalWrite(9,HIGH);  
digitalWrite(10,HIGH);  
digitalWrite(11,HIGH);  
digitalWrite(12,HIGH);  
digitalWrite(13,HIGH);  
}  
else {  
  digitalWrite(5,LOW);  
  digitalWrite(6,LOW);  
  digitalWrite(7,LOW);  
  digitalWrite(8,LOW);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
  digitalWrite(12,LOW);  
  digitalWrite(13,LOW);  
}  
}
```



NATJECANJE
MLADIH
TEHNIČARA



65. natjecanje mladih tehničara; 2022./2023., županijska razina

Model automatskih vrata

Svi povremeno odlazimo u velike trgovačke centre. Uvijek su na ulazima u takve centre postavljena automatska vrata, klizna ili okretna. Takva se vrata postavljaju iz praktičnih i higijenskih razloga. Tvoj je zadatak na eksperimentalnoj pločici izraditi pojednostavljen model automatskih vrata.

Materijal, oprema i alat potrebni za izradu zadatka:

- Mikroupravljačko sučelje prema vlastitom izboru
- Programator (ako je zaseban)
- Računalo s instaliranim softverskim paketom za programiranje odabranog sučelja
- Eksperimentalna pločica
- Ultrazvučni senzor
- Svjetleća dioda 5 mm, crvena
- Svjetleća dioda 5 mm, zelena
- Svjetleća dioda 5 mm, žuta
- Otpornik 330 Ohma, 3 komada
- Tipkala, 2 komada
- Spojne žice za spajanje komponenti - duljine otprilike 10 cm, 20 kom
- Produžni kabel s minimalno 3 utičnice, minimalne duljine 3 m
- Mala sjekača kliješta za elektroniku
- Kliješta za uklanjanje izolacije s električnih vodiča

Zadatak

U prvom dijelu, prema priloženim uputama, sastavi jednostavan elektronički sklop za upravljanje modelom automatskih vrata. Na eksperimentalnu pločicu montiraj crvenu, žutu i zelenu svjetleću diodu s pripadajućim otpornicima, dva tipkala i ultrazvučni senzor. Sve poveži s mikroupravljačkim sučeljem.

Nakon toga izradi program za upravljanjem tvojim modelom, prema sljedećim zahtjevima:

- Pokretanjem programa sve tri svjetleće diode se pale na jednu sekundu i zatim se gasi. Tim se procesom utvrđuje radi li sve tri diode ispravno.
- Potom se uključuje crvena svjetleća dioda što signalizira da su vrata zaključana. Zelena i žuta svjetleća dioda su ugašene.
- Pritiskom na prvo tipkalo započinje automatski način rada vrata. Crvena svjetleća dioda se gasi, a pali se zelena.
- U ovom, automatskom načinu rada, kada ultrazvučni senzor detektira da je neki predmet („osoba“) na udaljenosti manjoj od 10 cm, vrata se počinju otvarati, što se signalizira treptanjem žute svjetleće diode. Žuta dioda se pali i gasi tempom od po pola sekunde, pola sekunde je upaljena, pola ugašena. Taj se proces izvršava tri puta. Nakon toga se žuta svjetleća dioda pali, a zelena se gasi. To znači da su vrata sad potpuno otvorena. Proces otvaranja vrata traje 3 sekunde.
- Vrata ostaju otvorena dok god je neka „osoba“ ispred ultrazvučnog senzora, na udaljenosti manjoj od 10 cm. Za to vrijeme stanje svjetlećih dioda se ne mijenja.



- Kada se „osoba“ udalji od ultrazvučnog senzora na udaljenost veću od 10 cm, započinje proces zatvaranja vrata. Vrata se zatvaraju istim tempom kojim se i otvaraju, ali sada se žuta svjetleća dioda najprije gasi na pola sekunde, pa se zatim pali na pola sekunde, i to se izvršava tri puta. Nakon toga, žuta se svjetleća dioda gasi, a zelena se pali.
- Ponovnom detekcijom „osobe“ na udaljenosti manjoj od 10 cm proces otvaranja vrata kreće iznova.
- Pritiskom na drugo tipkalo vrata prelaze u zaključano stanje, upaljena je samo crvena svjetleća dioda i senzor ne očitava nikakve promjene.
- Ponovni pritisak na prvo tipkalo pokreće automatski rad vrata.

Sažetak

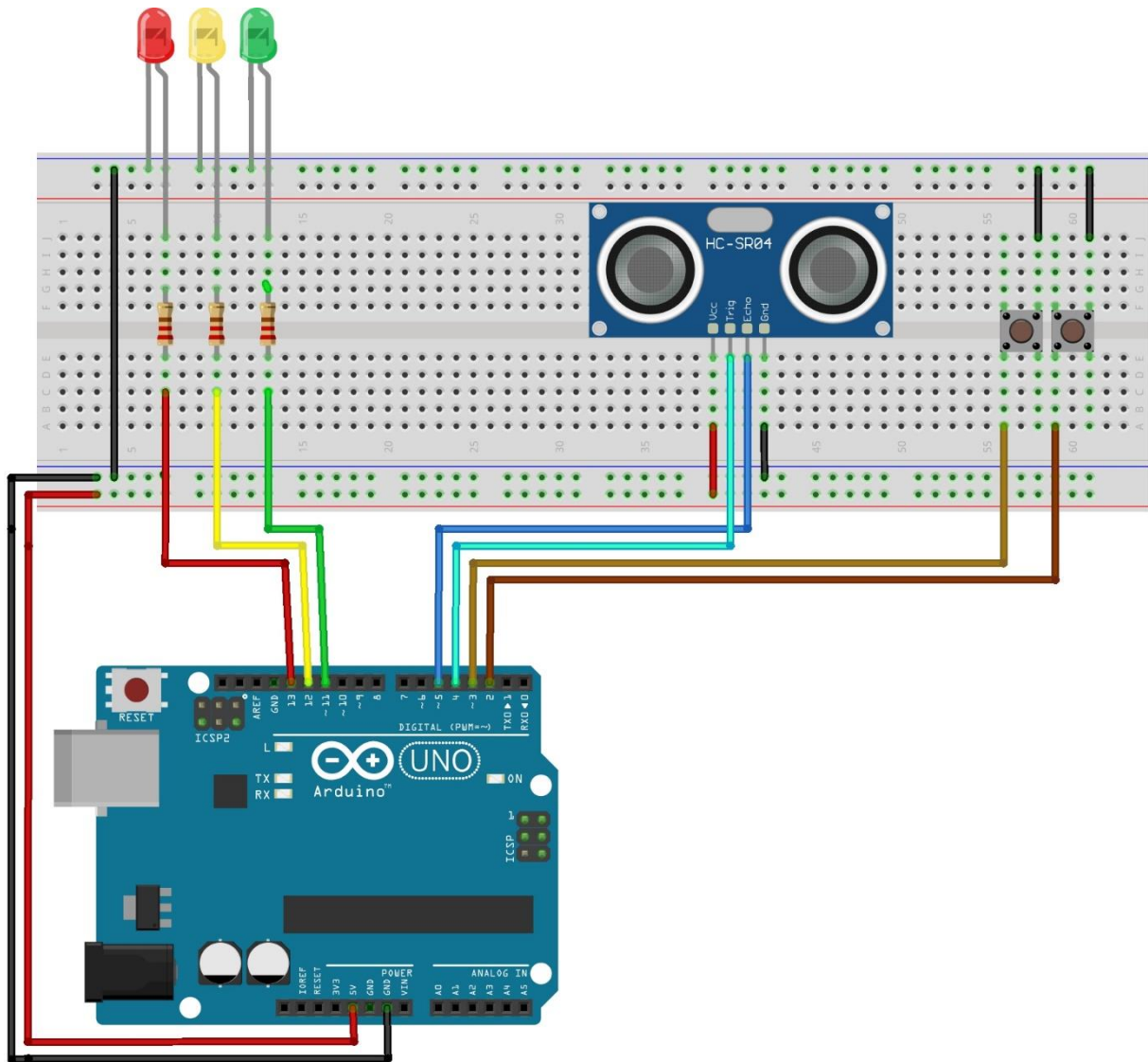
- Na početku programa vrata su „zatvorena“.
- Pritisak na prvo tipkalo „otključava“ vrata.
- Pritisak na drugo tipkalo „zaključava“ vrata.
- Očitavanje udaljenosti manje od 10 cm od ultrazvučnog senzora pokreće proces otvaranja vrata.
- Očitavanje udaljenosti veće od 10 cm od ultrazvučnog senzora pokreće proces zatvaranja vrata.
- Otvaranje ili zatvaranje vrata traju po 3 sekunde.



NATJECANJE
MLADIH
TEHNIČARA



Montažna shema:



fritzing

Jedno od mogućih programskih rješenja:

```
int tipk1=2;  
int tipk2=3;  
int trig=4;  
int echo=5;  
int led_zel=11;  
int led_zut=12;  
int led_crv=13;  
long vrijeme;  
int razmak;
```



```
void setup()
{
  pinMode(trig,OUTPUT);    //UZV okidanje
  pinMode(echo,INPUT);    //UZV očitavanje
  pinMode(led_zel,OUTPUT); //zelena LED
  pinMode(led_zut,OUTPUT); //žuta LED
  pinMode(led_crv,OUTPUT); //crvena LED
  pinMode(tipk1,INPUT_PULLUP); //tipkalo 1
  pinMode(tipk2,INPUT_PULLUP); // tipkalo 2
  digitalWrite(led_zel,HIGH);
  digitalWrite(led_zut,HIGH);
  digitalWrite(led_crv,HIGH);
  delay(1000);
  digitalWrite(led_zel,LOW);
  digitalWrite(led_zut,LOW);
  digitalWrite(led_crv,LOW);
  delay(1000);
  digitalWrite(led_crv,HIGH);
  Serial.begin(9600);
  Serial.println("Početak glavne petlje");
}

void loop()
{
  if(digitalRead(tipk1)==LOW)
  {
    autom_vrata();
  }
  zakljuc_vrata();
}

void zakljuc_vrata()
{
  digitalWrite(led_zel,LOW);
  digitalWrite(led_zut,LOW);
  digitalWrite(led_crv,HIGH);
  Serial.println("Zaključana vrata");
  while(digitalRead(tipk1)==1){}
}

void autom_vrata()
{
  Serial.println("Automatski rad vrata");
  digitalWrite(led_crv,LOW);
  digitalWrite(led_zel,HIGH);
  while(digitalRead(tipk2)==1)
  {
    prepreka();
  }
}
```



```
if((razmak>0)&&(razmak<10)){  
  Serial.println(razmak);  
  otvaranje();  
}  
digitalWrite(led_zut,LOW);  
}  
}
```

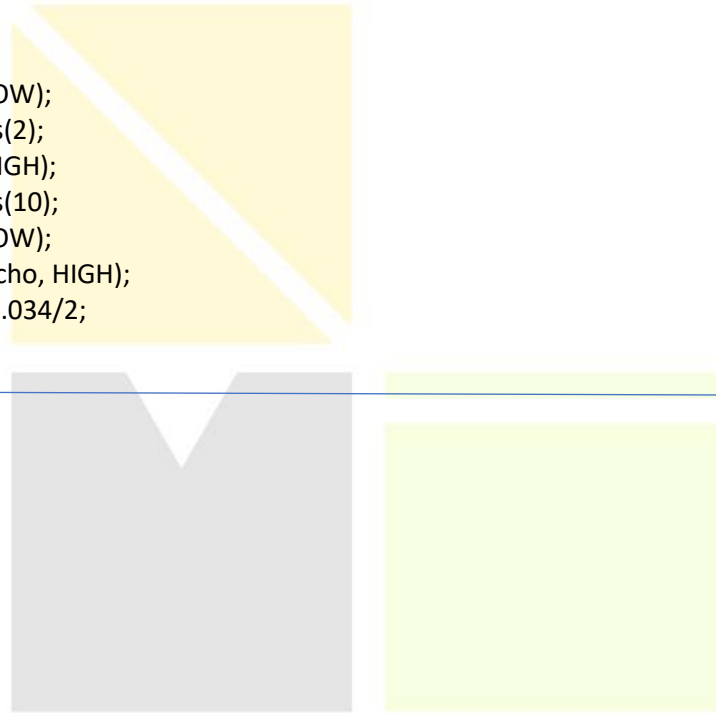
```
void otvaranje(){  
  Serial.println("Otvaranje: ");  
  digitalWrite(led_zut,HIGH);  
  delay(500);  
  digitalWrite(led_zut,LOW);  
  delay(500);  
  digitalWrite(led_zut,HIGH);  
  delay(500);  
  digitalWrite(led_zut,LOW);  
  delay(500);  
  digitalWrite(led_zut,HIGH);  
  delay(500);  
  digitalWrite(led_zut,LOW);  
  delay(500);  
  digitalWrite(led_zut,HIGH);  
  digitalWrite(led_zel,LOW);  
  Serial.println(razmak);  
  while(razmak<10)  
  {  
    Serial.println(razmak);  
    prepreka();  
  }  
  zatvaranje();  
}
```

```
void zatvaranje(){  
  Serial.println("Zatvaranje: ");  
  digitalWrite(led_zut,LOW);  
  delay(500);  
  prepreka();  
  digitalWrite(led_zut,HIGH);  
  delay(500);  
  prepreka();  
  digitalWrite(led_zut,LOW);  
  delay(500);  
  prepreka();  
  digitalWrite(led_zut,HIGH);  
  delay(500);  
  prepreka();  
  digitalWrite(led_zut,LOW);  
}
```




```
delay(500);  
prepreka();  
digitalWrite(led_zut,HIGH);  
delay(500);  
digitalWrite(led_zut,LOW);  
digitalWrite(led_zel,HIGH);  
prepreka();  
}
```

```
void prepreka(){  
digitalWrite(trig, LOW);  
delayMicroseconds(2);  
digitalWrite(trig, HIGH);  
delayMicroseconds(10);  
digitalWrite(trig, LOW);  
vrijeme= pulseIn(echo, HIGH);  
razmak= vrijeme*0.034/2;  
}
```



NATJECANJE
MLADIH
TEHNIČARA



66. natjecanje mladih tehničara; 2023./2024., županijska razina

Model elektromotornog pogona automobila

Svi znamo što su automobili. Znamo, vjerojatno, nabrojati i mnoge dijelove ili sklopove od kojih se sastoje moderni automobili. Na primjer, karoserija, kotači, motor, ovjes itd...

Automobil se kreće na kotačima, a kotače pokreće motor (benzinski, dizel, električni...).

Tvoj je zadatak na eksperimentalnoj pločici izraditi pojednostavljeni model elektromotornog pogona automobila. Uz to, tvoj će automobil imati osnovne svjetlosne sklopove (glavna svjetla, kočiona svjetla i pokazivače smjera), detektor nivoa rasvjete i radar za udaljenost od vozila ispred.

Oprema, pribor i materijal potrebni za izvršenje zadatka:

- Mikroupravljačko sučelje prema vlastitom izboru,
- Programator (ako je pogrebno),
- Računalo s instaliranim softverskim paketom za programiranje odabranog sučelja,
- Eksperimentalna pločica,
- Servomotor 360°, 2 kom,
- Ispravljač ili baterije za napajanje servomotora,
- Ultrazvučni senzor,
- Fotootpornik,
- Otpornik 10 K Ω ,
- Svjetleća dioda 5 mm, crvena, 2 kom,
- Svjetleća dioda 5 mm, bijela, 2 kom,
- Svjetleća dioda 5 mm, žuta, 2 kom,
- Otpornik 330 Ohma, 6 komada,
- Tipkalo, 4 komada,
- Spojne žice za spajanje komponenti – raznih duljina (5 cm, 10 cm, 15 cm), 40 kom,
- Produžni kabel s minimalno 3 utičnice, minimalne duljine 3 m,
- Mala sjekača kliješta za elektroniku,
- Kliješta za uklanjanje izolacije s električnih vodiča.

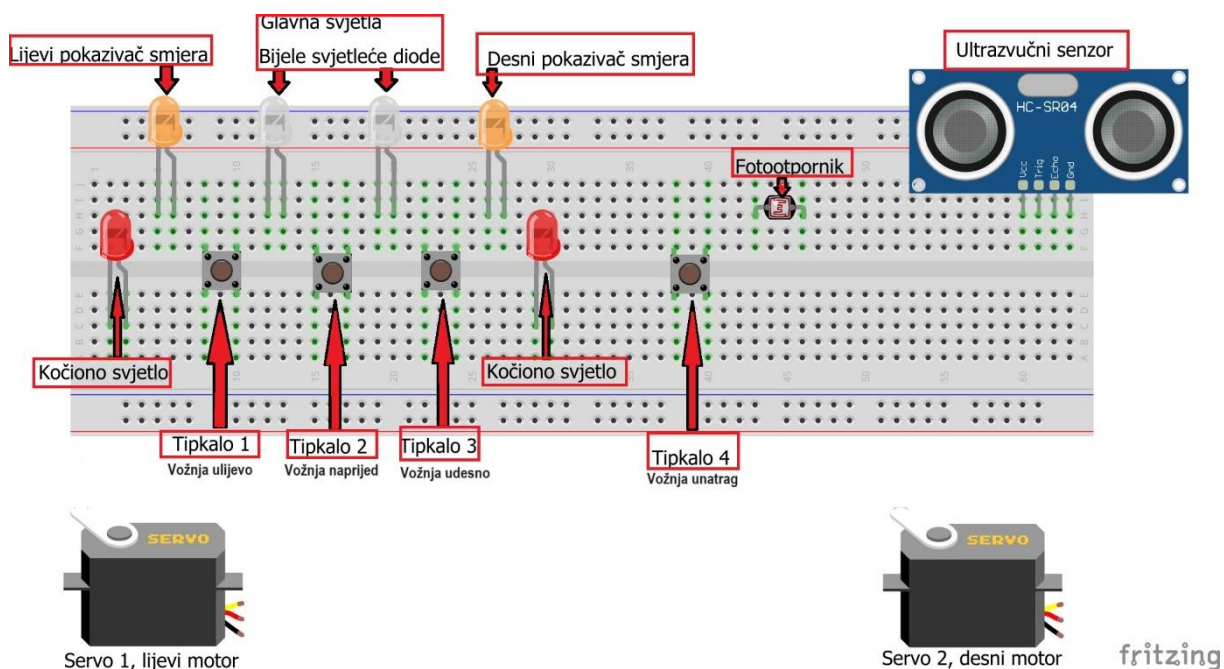


Zadatak:

U prvom dijelu, prema priloženim uputama, sastavi jednostavan elektronički sklop za upravljanje modelom elektromotornog pogona automobila. Na eksperimentalnu pločicu montiraj crvene, bijele i žute svjetleće diodu s pripadajućim otpornicima, četiri tipkala, fotootpornik sa zaštitnim otpornikom i ultrazvučni senzor.

Tipkala rasporedi tako da su tri tipkala postavljena blizu jedno drugom, a četvrto tipkalo postavi malo dalje od njih.

Montažna shema:



Na eksperimentalnu pločicu spoji „+“ i „-“, pol s mikroupravljačkog sučelja. Sve elemente poveži s napajanjem i mikroupravljačkim sučeljem. Servomotore poveži s napajanjem i mikroupravljačkim sučeljem.

Nakon toga izradi program za upravljanjem tvojim modelom, prema sljedećim zahtjevima:

- Pokretanjem programa sve svjetleće diode se pale na jednu sekundu i zatim se gase. Tim se procesom utvrđuje radi li sve tri diode ispravno. Taj dio programa se izvodi samo jednom, nakon pokretanja programa.

Sada počinje glavna petlja programa:

Fotootpornik detektira nivo rasvjete. Po svojoj želji odredi vrijednost koja će predstavljati granicu između „noći“ i „dana“. Ako je vrijednost očitana s fotootpornika manja od te tvoje granice, to znači da je pala noć i obje bijele svjetleće diode svijetle. Ako je očitana vrijednost veća od tvoje granice, to znači da je trenutno dan i obje su bijele svjetleće diode ugašene. (Pojednostavljeno, kad zakloniš fotootpornik rukom, to znači da je pala noć...)

- Tipkala imaju sljedeću funkciju:
 - o Pritisak samo na srednje tipkalo (tipkalo 2) pokreće automobil naprijed. Obje bijele svjetleće diode svijetle;
 - o Pritisak samo na lijevo tipkalo (tipkalo 1) pokreće automobil u lijevo (samo desni motor radi, pali se lijeva žuta svjetleća dioda);
 - o Pritisak samo na desno tipkalo (tipkalo 3) pokreće automobil u desno (samo lijevi motor radi, pali se desna žuta svjetleća dioda);
 - o Istovremeni pritisak na lijevo i desno tipkalo (tipkala 1 i 3), zaustavlja motore;
 - o Pritisak samo na četvrto tipkalo (tipkalo sa strane) pokreće automobil unatrag, oba pokazivača smjera su upaljena.
- Kada ultrazvučni senzor detektira da je neki predmet („osoba“) na udaljenosti manjoj od 10 cm, motori se zaustavljaju i obje crvene svjetleće diode su upaljene. Kada je udaljenost od „prepreke“ veća od 10 cm, crvene svjetleće diode se gase i motori nastavljaju s radom.

Napomene:

Motori smiju raditi samo kada je pritisnuto samo jedno tipkalo. Bilo koja kombinacija pritisaka na više od jednog tipkala uzrokuje zaustavljanje automobila!

Poželjno je da fotootpornik detektira razinu rasvjete neprestano tijekom izvršavanja programa i da se, zavisno od očitane vrijednosti, glavna svjetla pale ili gase, bez obzira na to rade li motori u tom trenutku, ili ne.



Jedno od mogućih programskih rješenja:

```
#include <Servo.h>

int trig = 2;      // Izvod UZV senzora za okidanje signala
int echo = 3;     // Izvod UZV senzora za očitavanje signala
int tipk1 = 4;    // Tipkalo 1
int tipk2 = 5;    // Tipkalo 2
int tipk3 = 6;    // Tipkalo 3
int tipk4 = 7;    // Tipkalo 4
int lijevi_PS = 8; // Lijevi pokazivač smjera
int desni_PS = 9; // Desni pokazivač smjera
int glavno = 12;  // Glavna svjetla
int zadnje = 13; // Kočiona svjetla
int foto=A0;     // Fotootpornik
int svjetlo = 800; // Razina koja označava razliku između dana i noći. Podesiti prema
vrijednostima očitavanja s fotootpornika

float trajanje; // Očitavanje s UZV senzora
float udaljenost; // Izračunata vrijednost udaljenosti u centimetrima

Servo servo_l;
Servo servo_d;

void setup() {
  servo_l.attach(10); // Lijevi motor
  servo_d.attach(11); // Desni motor
  pinMode(tipk1,INPUT_PULLUP);
  pinMode(tipk2,INPUT_PULLUP);
  pinMode(tipk3,INPUT_PULLUP);
  pinMode(tipk4,INPUT_PULLUP);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(lijevi_PS,OUTPUT);
  pinMode(desni_PS,OUTPUT);
  pinMode(glavno,OUTPUT);
  pinMode(zadnje,OUTPUT);

  Serial.begin(9600);

  digitalWrite(lijevi_PS,HIGH);
  digitalWrite(desni_PS,HIGH);
  digitalWrite(zadnje,HIGH);
  digitalWrite(glavno,HIGH);
```



```
delay(1000);  
digitalWrite(lijevi_PS,LOW);  
digitalWrite(desni_PS,LOW);  
digitalWrite(zadnje,LOW);  
digitalWrite(glavno,LOW);  
delay(1000);  
}
```

```
void loop() {  
  if ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==LOW) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==HIGH)){  
    naprijed();  
  }  
  else if ((digitalRead(tipk1)==LOW) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==HIGH)){  
    lijevo();  
  }  
  else if ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==LOW)&&  
(digitalRead(tipk4)==HIGH)){  
    desno();  
  }  
  else if ((digitalRead(tipk1)==LOW) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==LOW)&&  
(digitalRead(tipk4)==HIGH)){  
    stop();  
  }  
  else if ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==LOW)){  
    natrag();  
  }  
}
```

```
void naprijed(){  
  delay(100);  
  while ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==HIGH)){  
    //Serial.println("Naprijed");  
    prepreka();  
    servo_l.write(150);  
    servo_d.write(30);  
    digitalWrite(lijevi_PS,LOW);  
    digitalWrite(desni_PS,LOW);  
    digitalWrite(zadnje,LOW);  
    foto = analogRead(A0);  
    Serial.println(foto);  
  }
```




```
if (foto < svjetlo){  
    digitalWrite(glavno,HIGH);  
}  
else{  
    digitalWrite(glavno,LOW);  
}  
}  
}
```

```
void lijevo(){  
    delay(100);  
    while ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==HIGH)){  
        //Serial.println("Lijevo");  
        prepreka();  
        servo_l.write(90);  
        servo_d.write(30);  
        digitalWrite(lijevi_PS,HIGH);  
        digitalWrite(desni_PS,LOW);  
        digitalWrite(zadnje,LOW);  
        foto = analogRead(A0);  
        Serial.println(foto);  
        if (foto < svjetlo){  
            digitalWrite(glavno,HIGH);  
        }  
        else{  
            digitalWrite(glavno,LOW);  
        }  
    }  
}
```

```
void desno(){  
    delay(100);  
    while ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&  
(digitalRead(tipk4)==HIGH)){  
        //Serial.println("Desno");  
        prepreka();  
        servo_l.write(150);  
        servo_d.write(90);  
        digitalWrite(lijevi_PS,LOW);  
        digitalWrite(desni_PS,HIGH);  
        digitalWrite(zadnje,LOW);  
        foto = analogRead(A0);  
        Serial.println(foto);  
    }  
}
```



```
if (foto < svjetlo){
    digitalWrite(glavno,HIGH);
}
else{
    digitalWrite(glavno,LOW);
}
}
}

void natrag(){
    delay(100);
    while ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&
(digitalRead(tipk4)==HIGH)){
        //Serial.println("Natrag");
        servo_l.write(30);
        servo_d.write(150);
        digitalWrite(lijevi_PS,HIGH);
        digitalWrite(desni_PS,HIGH);
        digitalWrite(zadnje,LOW);
        foto = analogRead(A0);
        Serial.println(foto);
        if (foto < svjetlo){
            digitalWrite(glavno,HIGH);
        }
        else{
            digitalWrite(glavno,LOW);
        }
    }
}

void stop(){
    delay(100);
    while ((digitalRead(tipk1)==HIGH) && (digitalRead(tipk2)==HIGH) && (digitalRead(tipk3)==HIGH)&&
(digitalRead(tipk4)==HIGH)){
        //Serial.println("Stop");
        digitalWrite(lijevi_PS,LOW);
        digitalWrite(desni_PS,LOW);
        digitalWrite(zadnje,LOW);
        digitalWrite(glavno,HIGH);
        servo_l.write(90);
        servo_d.write(90);
        foto = analogRead(A0);
        Serial.println(foto);
        if (foto < svjetlo){
```



```
    digitalWrite(glavno,HIGH);  
  }  
  else{  
    digitalWrite(glavno,LOW);  
  }  
}  
}
```

```
void prepreka(){  
  digitalWrite(trig, LOW);    // smirivanje odašiljača  
  delayMicroseconds(2);     // smirivanje odašiljača  
  digitalWrite(trig, HIGH);  // odašiljanje signala  
  delayMicroseconds(10);    // odašiljanje signala  
  digitalWrite(trig, LOW);  // smirivanje odašiljača  
  trajanje = pulseIn(echo, HIGH); // očitavanje proteklog vremena  
  udaljenost = trajanje * 0.034/2; // diljeljenje očitano g vremena s brzinom zvuka i još kroz 2 da bi se  
  dobila udaljenost do prepreke  
  while (udaljenost < 10){  
    digitalWrite(trig, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trig, LOW);  
    trajanje = pulseIn(echo, HIGH);  
    udaljenost = trajanje * 0.034/2;  
    servo_l.write(90);  
    servo_d.write(90);  
    digitalWrite(lijevi_PS,LOW);  
    digitalWrite(desni_PS,LOW);  
    digitalWrite(zadnje,HIGH);  
    digitalWrite(glavno,LOW);  
    //Serial.println(udaljenost);  
  }  
  //Serial.println(udaljenost);  
}
```

