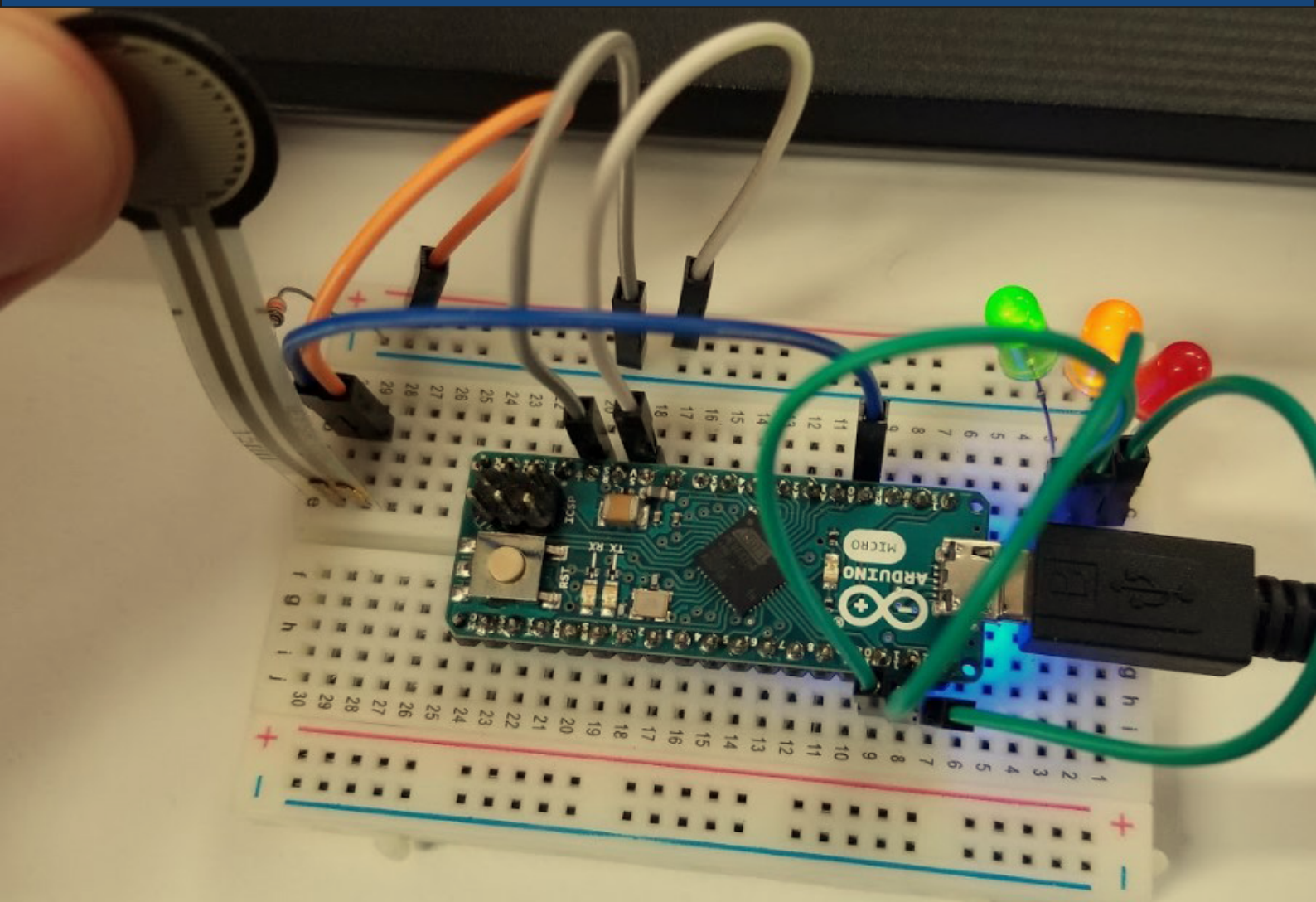




PRAKTIKUM ZA RADIONICU ELEKTROTEHNIKE I ROBOTIKE ZA UČENIKE SREDNJIH ŠKOLA U BOSNI I HERCEGOVINI



SADRŽAJ

ARDUINO MICRO.....	3
HARDVER.....	4
Ultrazvučni senzor za mjerenje udaljenosti (HC-SR04).....	6
Senzor za mjerenje pritiska (Force sensitive resistor).....	7
LCD ekran 16x2.....	7
SOFTVER.....	8
PODEŠAVANJE OKRUŽENJA.....	9
VJEŽBA BR. 1	
LED BLINK.....	11
VJEŽBA BR. 2	
LED FADE.....	12
VJEŽBA BR. 3	
SEMAFOR.....	14
VJEŽBA BR. 4	
SEMAFOR V2.....	15
VJEŽBA BR. 5	
LED BLINK V2.....	17
VJEŽBA BR. 6	
TRČEĆE SVIJETLO.....	18
VJEŽBA BR. 7	
SEMAFOR BONUS.....	19
VJEŽBA BR. 8	
MJERENJE RAZDALJINE.....	20
VJEŽBA BR. 9	
MJERENJE PRITISKA.....	22



WHO ARE WE AND WHAT DO WE DO?



**BH Futures
Foundation**

ARDUINO MICRO

Radionica Arduino Micro pruža priliku učenja i upoznavanja sa osnovnim komponentama u elektronici kroz realne i zanimljive primjere.

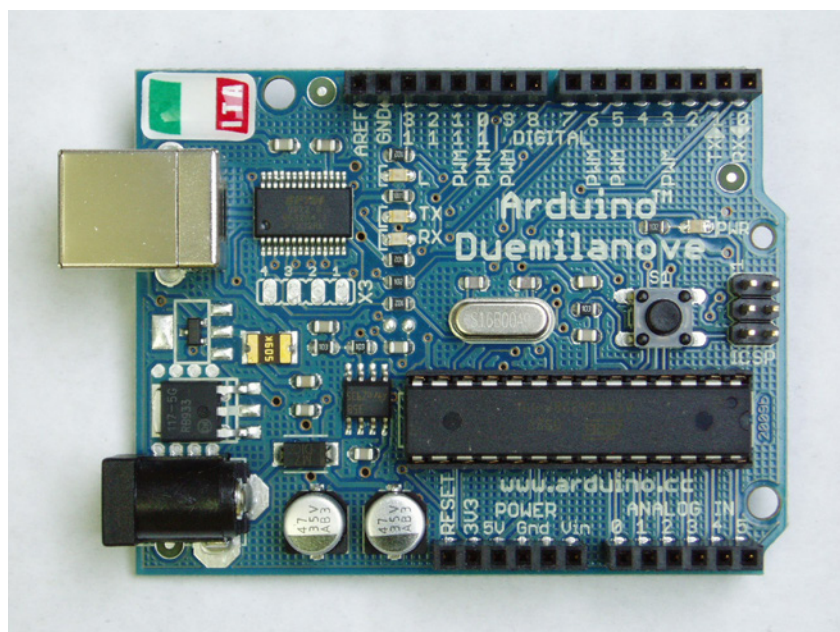
Za potrebe ove radionice kao mikrokontrolerska komponenta je korišten mikrokontroler Arduino micro koji se na jednostavan način može programirati da upravlja ostatkom električnog kola na željeni način. Ujedno ovaj mikrokontroler predstavlja osnovnu komponentu cijelog sistema. Osim mikrokontrolera za izvođenje pokaznih primjera neophodne su i komponente:

- Matador ploča
- Otpornici različitih otpornosti
- Prekidačke komponente
- Kablovi (Jumperi)
- Diode (LED)

Mikrokontroleri (skraćeno μC , uC ili MCU) su integrisani el. uređaji koji imaju sve što im je potrebno da budu računari: memoriju, procesor, ulaze i izlaze. Memorija u formi NOR flash ili OTP ROM je često uključena na čipu kao i standardna mala količina RAM-a. Mikrokontroleri se koriste u automatski upravljanim proizvodima i uređajima, kao što su kontrolni sistem motora automobila, medicinskim uređajima, uređajima na daljinsko upravljanje itd.

Arduino je fizičko-računarska platforma (razvojni sistem) otvorenog koda. Hardver se sastoji od jednostavnog otvorenog hardverskog dizajna Arduino ploče sa procesorom i pratećim ulazno-izlaznim elementima, tačnije, na sebi poseduje mikrokontroler. Softver se sastoji od razvojnog okruženja koje čine standardni kompajler i bootloader koji se nalazi na samoj ploči.

Arduino hardver se programira koristeći programski jezik zasnovan na Wiring jeziku (sintaksa i biblioteke). U osnovi je sličan C++ programskom jeziku sa izvesnim pojednostavljenjima i izmenama. Integrisano razvojno okruženje je zasnovano na Processing-u.



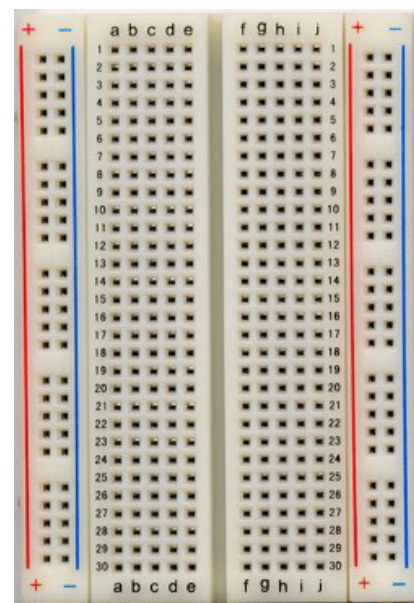
HARDVER

Arduino ploču čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom. Bitan aspekt Arduino projekta je standardizovan raspored konektora koji omogućava lako povezivanje sa dodatnim modulima, poznatijim kao štitovi. Ove dodatne module, štitove, proizvode razni proizvođači širom sveta. Većina ploča poseduje 5V linearni naponski regulator i 16MHz kristalni oscilator (ili keramički rezonator u nekim verzijama). Arduino mikrokontroleri se isporučuju sa programiranim bootloader-om koji pojednostavljuje postupak prebacivanja prevedenog koda u fleš memoriju na čipu. Drugi mikrokontroleri obično zahtevaju zaseban programator.

Da bi bilo moguće uspešno izvesti ovakvu vježbu koristićemo matador ploču koja pruža mogućnost konektovanja svih ostalih komponenti sistema na jednostavan način. Takođe vrlo je praktična za demonstraciju jer se jednostavno vrši postavljanje/uklanjanje pinova.

Otpornici su pasivne elektroničke komponente koje pružaju otpor protjecanju električne struje. Drugim riječima, apsorbiraju dio električne energije koju pretvaraju u toplinu. Možemo ih pronaći u gotovo svim elektroničkim sklopovima gdje se koriste za stvaranje željenog strujno-naponskog odnosa, ograničenje struje, smanjenje napona i struje.

Prekidač je jednostavni elektromehanički uređaj koji zatvara ili otvara strujno kolo. Sa time omogućuje ili onemogućuje tok struje, i kod jednostavnih sklopova, uključuje ili isključuje uređaj. Za razliku od tastera, ostvaruje trajan, a ne trenutni električni spoj.



Video and Website © 2004 ClarkZapper.net

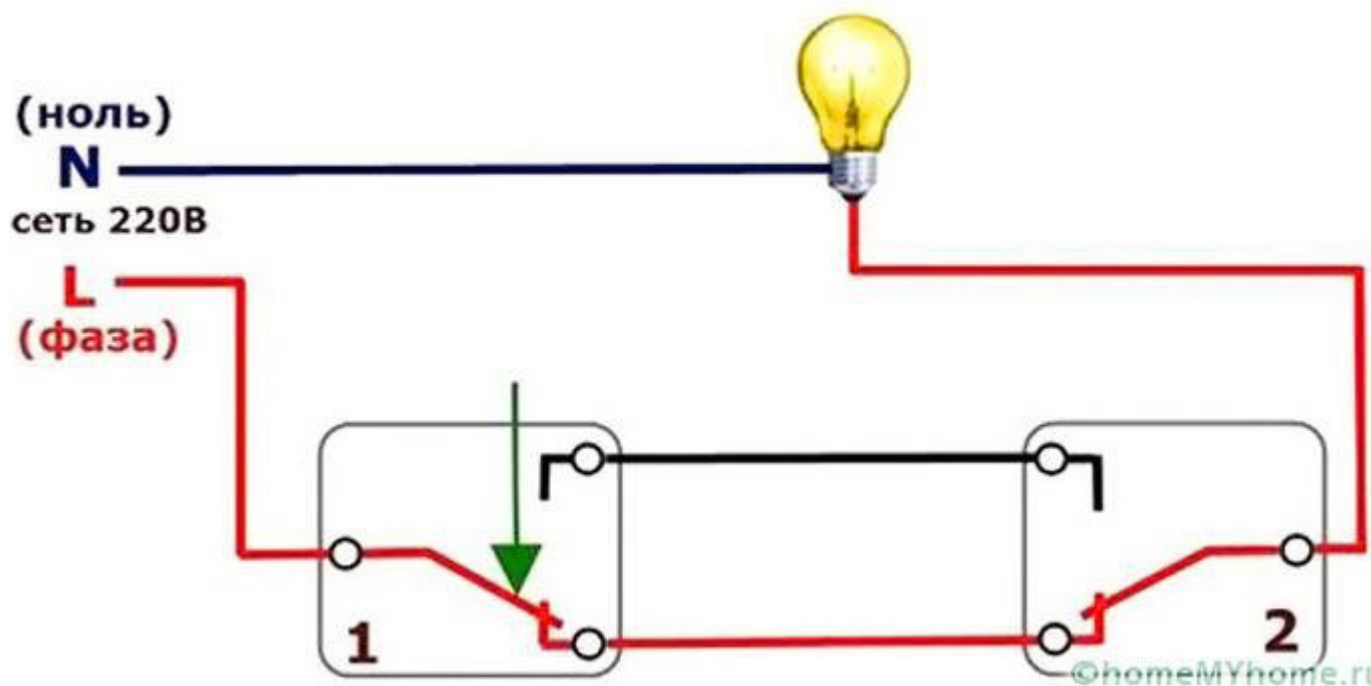
Matador ploča



Otpornici

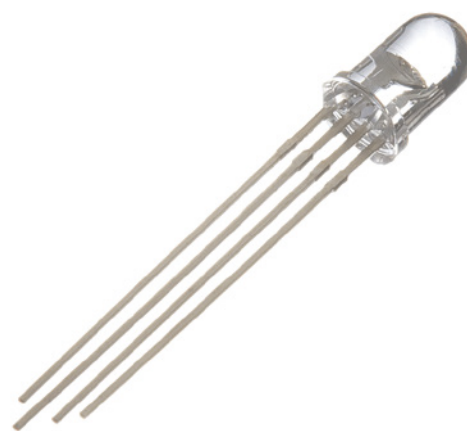
Tehnički posmatrano, prekidači moraju da isključuju i struje kratkog spoja, pa obični prekidači malih snaga kućnih električnih uređaja ne spadaju u tu grupu. Međutim, naziv prekidač je u tako raširenoj upotrebi da se koristi za obje vrste.

U elektronici, prekidačima se često nazivaju i druge elektronske komponente koje vrše ulogu prekidanja struje. Tako se može govoriti o tranzistoru, releju, ili elektronskoj cijevi kao prekidaču.



Šema prekidača

Svjetleća dioda ili LED (skr. od engl. Light Emitting Diode) je poluprovodnički element koji pretvara električni signal u optički signal (svjetlost). Propusno polarizovana dioda (svjetleća dioda) emituje elektromagnetsko zračenje kao spontane emisije uzrokovane rekombinacijom nosilaca električnoga naboja (elektroluminiscencija). Elektroni prelazeći iz vodljivog u valentni pojas, oslobađaju energiju, koja se dijelom emituje u vidu toplotne energije kao toplota, a dijelom kao zračenje. Boja emitovanog svjetla zavisi od poluprovodnika, kao i o primjesama u njemu i varira od infracrvenog preko vidljivog do ultraljubičastog dijela spektra.

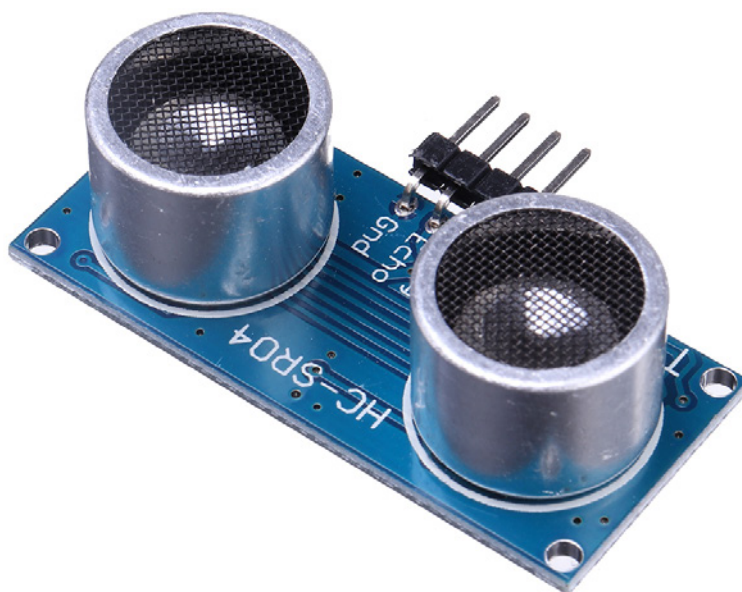


LED dioda

Ultrazvučni senzor za mjerenje udaljenosti (HC-SR04)

Mnogi mjerači udaljenosti danas se baziraju ovoj mjernoj metodi kod koje mjerni impuls dolazi iz ultrazvučnog, optičkog ili RF izvora energije. Parametri impulsa koji se emituje bitni su za proračun udaljenosti, tako na primjer brzina zvuka u vazduhu (343m/s pri 20°C), brzina svjetlosti ($299\,792\,458\text{ m/s}$ odnosno $0,3\text{m/ns}$). Koristeći elementarnu fiziku dobijamo $\text{udaljenost} = \text{brzina} \cdot \text{vrijeme}$, gdje udaljenost mjerimo indirektno mjereći vrijeme uz poznatu brzinu prostiranja impulsa kroz vazduh ili neki drugi medij. Greške koje mogu nastati kod primjene TOF metode mjerenja udaljenosti nisu zanemarljive i zato se ovi senzori koriste najviše u pokazne i edukativne svrhe jer nisu pouzdani za precizno mjerenje distance iz više razloga a neki od njih su:

- Brzina zvuka kroz vazduh mijenja se sa promjenom temperature i vlažnost vazduha pa te promjene utiču na mjerni podatak
- Detekcijska nesigurnost ili (time-walk errors) uslovljena različitim površinama predmeta od kojih se odbija impuls. Uzrokuje promjenu u frekvencijskoj karakteristici impulsa.
- Kada se gleda razlika u brzini prostiranja zvučnih i svjetlosnih impulsa kroz vazduh tada vrijeme koje mjerimo kod svjetlosti je 10 na 6 puta kraće nego kod zvuka, što poskupljuje izvedbu uređaja.
- Površinska interakcija javlja se kad se emitovani impuls iz senzora višestruko odbije od raznih površina i na kraju ga detektuje prijemnik senzora. Takav signal nije vjerodostojan podatak o udaljenosti te ga se mora smatrati nevaljanim. Takva pojava naziva se (crosstalk).



*Ultrazvučni senzor za mjerenje udaljenosti
(HC-SR04)*

Senzor za mjerenje pritiska (Force sensitive resistor)

Ovo je senzor čiji se otpor mijenja prilikom primjene mehaničke sile ili pritiska na glavu senzora.

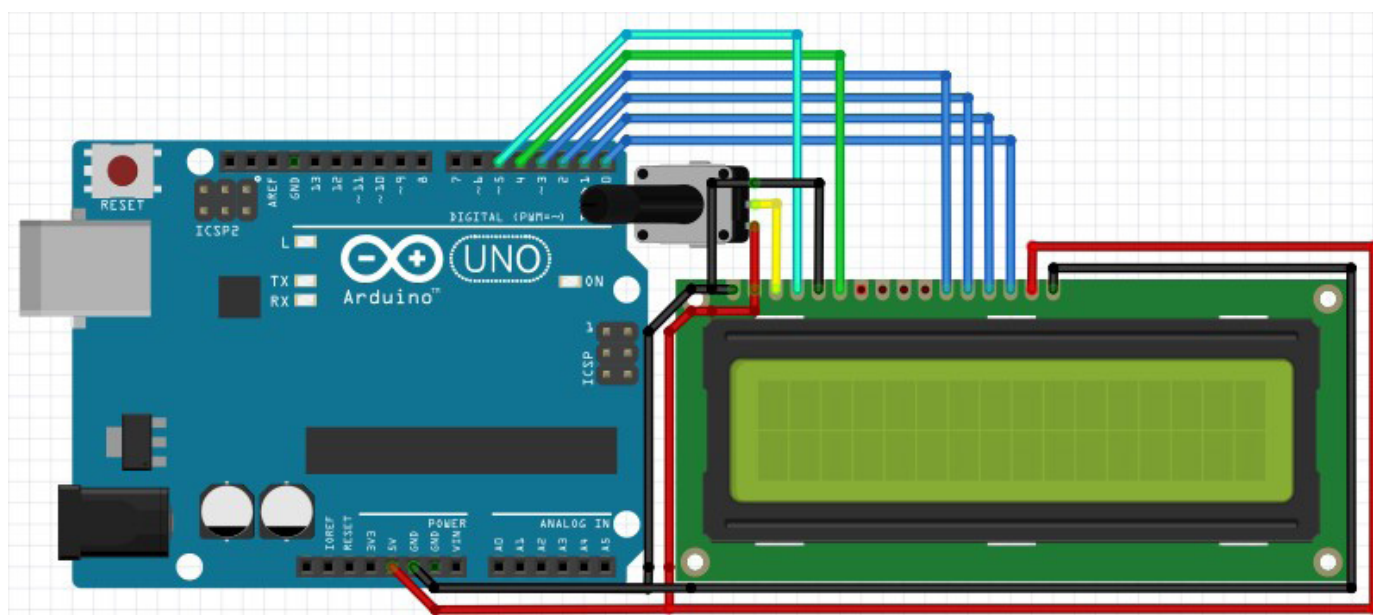
Senzorski otpornici sastoje se od provodnog polimera, koji na predvidiv način mijenja otpor nakon primjene sile na njegovu površinu. Obično se isporučuju kao polimerni list ili mastilo koje se može primeniti sitotiskom. Senzorski film se sastoji od električno provodnih i nevodljivih čestica suspendovanih u matrici. Čestice su veličine ispod mikrometra i formulisane su tako da smanjuju temperaturnu zavisnost, poboljšavaju mehanička svojstva i povećavaju trajnost površine. Primjenom sile na površinu osjetljivog filma, čestice dodiruju provodne elektrode, mijenjajući otpornost filma. Kao i sa svim senzorma baziranim na otpornicima, otpornici sa senzorom sile imaju jednostavan izgled i funkciju a takođe mogu raditi zadovoljavajuće u umjereno neprijateljskim okruženjima.



*Senzor za
mjerenje pritiska
(Force sensitive re-*

LCD ekran 16x2

Ovo je komponenta koja omogućava prikazivanje teksta, simbola, vrijednosti ili bilo čega drugoga što dolazi s Arduino uređaja. Sposoban je pokazati 16 znakova u 2 reda te je koristan u mnoštvu projekata koji trebaju poslati jasno vidljivu i čitku izlaznu informaciju.



LCD ekran 16x2



SOFTVER

Arduino integrisano razvojno okruženje je aplikacija napisana u Java programskom jeziku. Kreirano je tako da uvede u programiranje učenike, studente i ostale početnike koji nisu upoznati sa načinom razvoja softvera. Sastoji se od uređivača koda sa mogućnostima kao što su označavanje koda, uparivanje zagrada, automatsko uvlačenje linija. Ovaj uređivač može da prevede kôd a zatim ga i prebaci u čip jednom komandom. U ovom slučaju nije potrebno podašavati parametre prevođenja koda ili pokretati programe iz komandne linije.

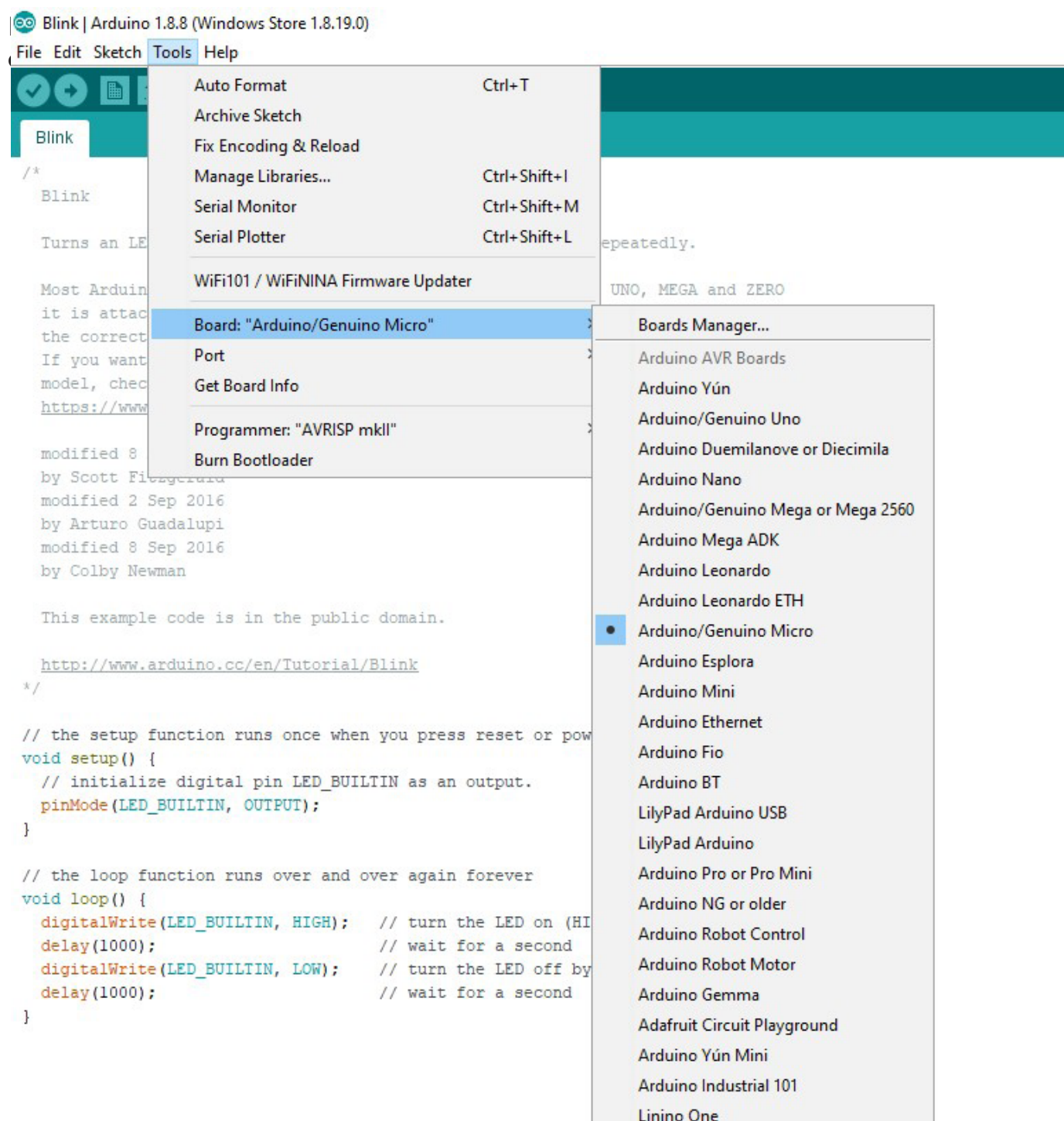
Arduino integrisano razvojno okruženje dolazi sa C/C++ bibliotekom zvanom "Wiring" koja čini uobičajene ulazno-izlazne operacije veoma jednostavnim. Arduino programi se pisu u C/C++ programskom jeziku, mada korisnici moraju da definišu samo dve funkcije kako bi napravili izvršni program. Te funkcije su:

- `setup()` - funkcija koja se izvršava jednom na početku i služi za početna podešavanja
- `loop()` - funkcija koja se izvršava u petlji sve vreme dok se ne isključi ploča

Naziv vježbe	Komponente	Vrijeme izrade	Težina
LED blink	LED, Otpornik, Arduino, Matador ploča, jumper	5-10 min	Početnik
LED fade	LED, Otpornik, Arduino, Matador ploča, jumper	5-10 min	Početnik
Semafor	3xLED, 3xOtpornik, Arduino, Matador ploča, jumper	10-15 min	Srednji
Semafor v2.	5xLED, 3xOtpornik, Arduino, Matador ploča, jumper	10-15 min	Srednji
LED blink v2.	LED, Otpornik, Arduino, Matador ploča, jumper	10-15 min	Napredni
Trčeće svijetlo	3xLED, 3xOtpotnik, Potenciometar, Arduino, Matador ploča, jumper	10-15 min	Napredni
Semafor (bonus)	5xLED, 5xOtpornik, Arduino, Matador ploča, jumper	10-15 min	Srednji
Mjerenje razdaljine	Ultrazvučni sensor, Arduino, Matador ploča, jumper, otpornici	5-10 min	Srednji
Mjerenje pritiska	Senzor mjerenja pritiska, 3x LED, Arduino, Matador ploča, jumper, otpornici	5-10 min	Srednji

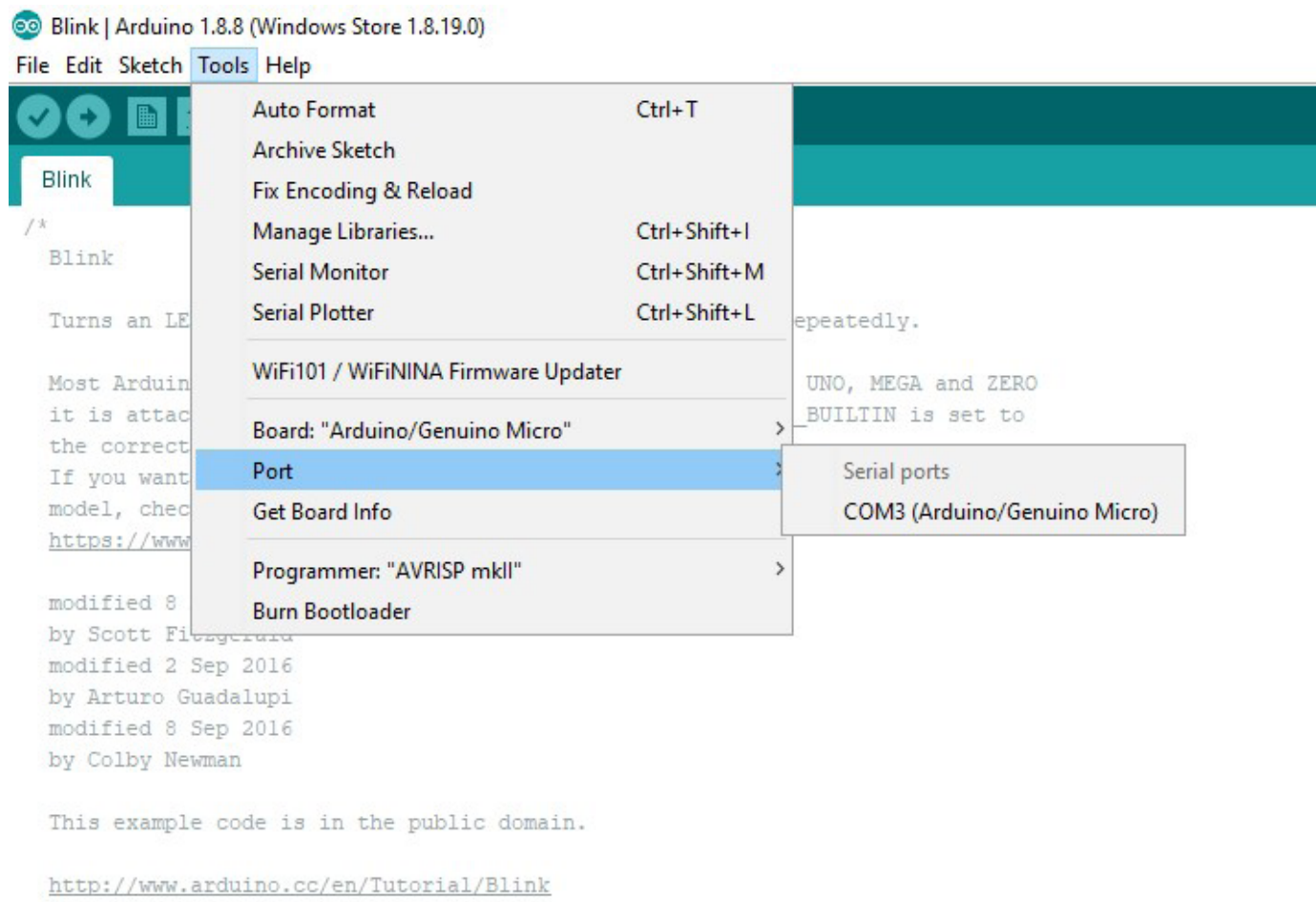
PODEŠAVANJE OKRUŽENJA

Da bi komunikacija između hardvera (Arduino uređaja) i softvera (Razvojno okruženje Arduino) bila ostvarena podrebnno je izvršiti podešavanje okruženja. Povezivanje se obavlja pomoću USB kabela koji se sa USB porta na računaru priključuju na USB Micro-b ulaz na strain Arduino uređaja (Napomena: Za različite vrste Arduino uređaja koriste se različite vrste kabela. Za Arduino Micro koji je korišten u ovoj radionici koriste se gorenavedeni tip kabela).



Izbor Arduino uređaja u programskom okruženju

Nakon pravilno izabranog Arduino uređaja potrebno je odabrati port na kome je konektovan naš Arduino da bi računar znao sa kojim USB portom treba da razmjenjuje podatke. To ćemo uraditi na sljedeći način:



Izbor željenog porta za komunikaciju

Sada je sve spremno za kodiranje. Nakon završetka pisanja koda potrebno je izvršiti validaciju koda i prebacivanje istog na Arduino uređaj. To se vrši klikom na dugme označeno br.1 i ukoliko je validacija uspješna prebacivanje koda na Arduino klikom na dugme br. 2. Nedugo nakon toga Arduino uređaj je u funkciji.



Validacija i kompajliranje koda

VJEŽBA BR. 1 LED BLINK

Jednostavna vježba koja za cilj ima da uključi i isključi LED diodu naizjenu. Iako je to jedna od najjednostavnijih primjera upotrebe koje možemo obaviti sa Arduino mikrokontrolerom za početnike u elektronici to može izgledati izazovno i zabavno. Cilj vježbe je kako je već navedeno "natjerati" LED diodu da nakon proizvoljno odabranog vremenskog intervala (koji je prilikom kodiranja izabran) svijetli a zatim je isključiti. U Arduino okruženju, korisnik bi mogao da napiše ovakav program:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

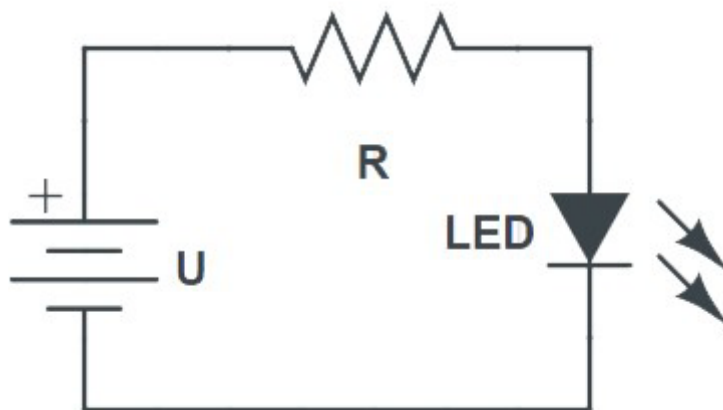
Izgled najjednostavnijeg koda pisanog za Arduino mikrokontroler (LED blink)

Izgled električnog kola koje odgovara napisanom kodu nalazi se na sljedećoj slici:

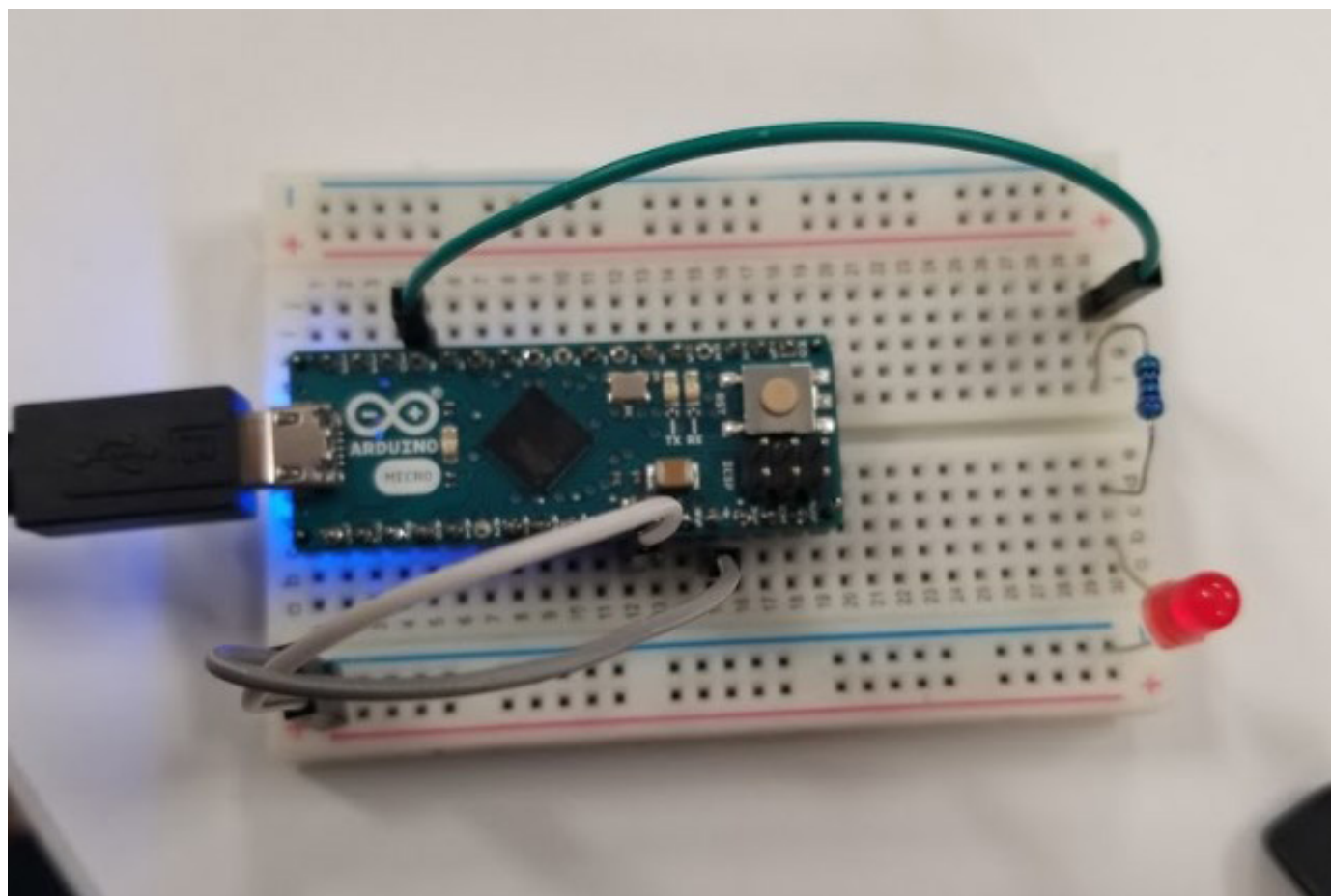
U - Naponski izvor, u slučaju Arduino mikrokontrolera to je najčešće 5V

R – Odgovarajući otpornik

LED – (Light Emiting Diode) Dioda željene boje



Jednostavno el. kolo sa LED diodom



Izgled vježbe 1. i 2.

VJEŽBA BR. 2

LED FADE

Za ovu vježbu će nam biti potrebne iste komponente kao i za prethodnu (vježbu br. 1) s' tim da će biti korišćene druge funkcije u programu Arduino studio. Cilj vježbe je konstruisati električno kolo na identičan način kao u prethodnoj vježbi a zatim napisati kod koji će postepeno pojačavati svjetlost koju LED dioda emituje a kada dostigne maksimalnu vrijednost ugasi se i napravi kratku pauzu prije nego što ponovo kreće da se postepeno pojačava nivo osvjjetljenja. Da bi uspješno odradili zadatu vježbu moraćemo obraditi funkciju AnalogWrite kao i PWM modul. PWM služi za digitalni prikaz analognih vrijednosti.

Kodiranje je veoma jednostavno a primjer kako to može biti urađeno nalazi se na slici ispod:



The analogWrite() function uses PWM, so if you want to change the pin you're using, be sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/Fade>

**/*

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Kod za vježbu "LED FADE"

Suština napisanog koda je da inkrementalno povećava osvjjetljenost za datu vrijednost fadeAmount (u ovom slučaju 5, možemo staviti i drugu vrijednost), Kada prekorači maksimalnu vrijednost od 255 tada se vrijednost fadeAmount postavlja na -5 i zatim se svjetlost inkrementalno smanjuje dok ne dođe do 0. Proces se zatim obrće i kreće ponovno pojačavanje svjetlosti.



VJEŽBA BR. 3 SEMAFOR

Spojite na mikrokontroler tri LED diode kako biste napravili mali model jednostavnog semafora za automobile. Na početku na semaforu svijetli samo crveno svijetlo tri sekunde. Potom se uključuje žuto svijetlo i svijetli zajedno s crvenim dodatnu jednu sekundu. Nakon toga svijetli samo zeleno svijetlo četiri sekunde. Nakon zelenog svijetli samo žuto svijetlo jednu sekundu, te ciklus kreće ispočetka.

```
int ledcrvena = 13;
int ledzuta = 12;
int ledzelena = 11;

void setup() {
  pinMode(ledcrvena, OUTPUT);
  pinMode(ledzuta, OUTPUT);
  pinMode(ledzelena, OUTPUT);
}

void loop() {

  digitalWrite(ledcrvena, HIGH);
  delay(3000);

  digitalWrite(ledzuta, HIGH);
  delay(1000);

  digitalWrite(ledcrvena, LOW);
  digitalWrite(ledzuta, LOW);

  digitalWrite(ledzelena, HIGH);
  delay(4000);
  digitalWrite(ledzelena, LOW);

  digitalWrite(ledzuta, HIGH);
  delay(1000);
  digitalWrite(ledzuta, LOW);

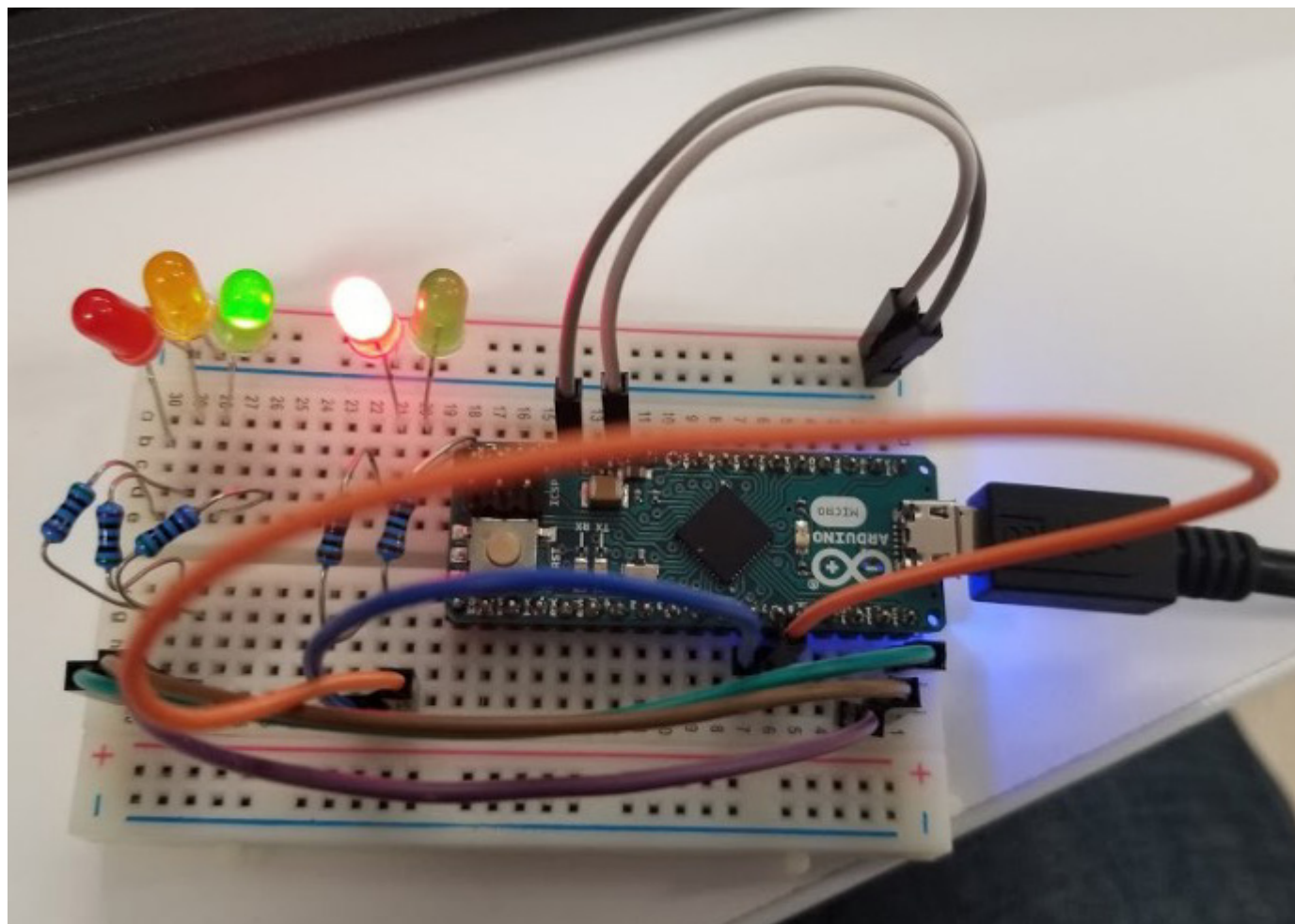
}
```

Kod za vježbu "Semafor"

Napomena: LED diode moguće je spojiti na bilo koji digitalni izvod mikrokontrolera, prema Vašoj želji. Izvodi 13, 12 i 11 u ovom primjeru odabrani su nasumično.

VJEŽBA BR. 4 SEMAFOR V2

Nadogradite prethodni zadatak tako da dodate još dvije LED diode, jednu crvenu i jednu zelenu. One predstavljaju semafor za pješake. Model semafora naizmjenično propušta automobile i pješake. Na semaforu za pješake uključeno je zeleno svjetlo samo onda kada je na semaforu za automobile uključeno crveno svjetlo. U svim ostalim kombinacijama na semaforu za pješake uključeno je crveno svjetlo.



Izgled vježbe 4.

Napomena: LED diode moguće je spojiti na bilo koji digitalni izvod mikrokontrolera, prema Vašoj želji. Izvodi 7 i 6 u ovom primjeru odabrani su nasumično.



```
int ledcrvena = 13;
int ledzuta = 12;
int ledzelena = 11;

int ledcrvenap = 7;
int ledzelenap = 6;

void setup() {
  pinMode(ledcrvena, OUTPUT);
  pinMode(ledzuta, OUTPUT);
  pinMode(ledzelena, OUTPUT);

  pinMode(ledcrvenap, OUTPUT);
  pinMode(ledzelenap, OUTPUT);
}

void loop() {

  digitalWrite(ledcrvena, HIGH);
  digitalWrite(ledzelenap, HIGH);
  delay(3000);

  digitalWrite(ledzelenap, LOW);
  digitalWrite(ledcrvenap, HIGH);
  digitalWrite(ledzuta, HIGH);
  delay(1000);

  digitalWrite(ledcrvena, LOW);
  digitalWrite(ledzuta, LOW);

  digitalWrite(ledzelena, HIGH);
  delay(4000);
  digitalWrite(ledzelena, LOW);

  digitalWrite(ledzuta, HIGH);
  delay(1000);
  digitalWrite(ledzuta, LOW);

  digitalWrite(ledcrvenap, LOW);
}
```

Kod za vježbu "Semafor" sa svjetlima za pješake



VJEŽBA BR. 5 LED BLINK V2

Ovo vježba je orijentisana na programiranje. Zadatak je podesiti LED diodu tako da se nasumično uključuje i isključuje. Modifikacija u odnosu na prvi zadatak je da je to potrebno uraditi bez korišćenja ugrađene funkcije delay(ms).

```
// constants won't change. Used here to set a pin number:
const int ledPin = LED_BUILTIN; // the number of the LED pin

// Variables will change:
int ledState = LOW; // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0; // will store last time LED was updated

// constants won't change:
const long interval = 1000; // interval at which to blink (milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the difference
  // between the current time and last time you blinked the LED is bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}
```

Kod za vježbu "Semafor" sa svjetlima za pješake



VJEŽBA BR. 6 TRČEĆE SVIJETLO

Napravite trčeće svijetlo sa tri LED diode po ugledu na vježbu “Semafor” ali s mogućnošću upravljanja brzinom efekta trčanja. Za promjenu brzine trčanja iskoristite jedan potenciometar spojen na analogni ulaz mikrokontrolera.

Napomena: Kako biste mijenjali brzinu efekta mijenjajte pauzu između uključivanja LED dioda, a za promjenu pause iskoristite očitane vrijednosti sa analognog ulaza.

```
int analogniulaz = A0;
int led1 = 13;
int led2 = 12;
int led3 = 11;
int pauza = 0;

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop() {
  pauza = analogRead(analogniulaz);

  digitalWrite(led1, HIGH);
  delay(pauza);
  digitalWrite(led1, LOW);

  digitalWrite(led2, HIGH);
  delay(pauza);
  digitalWrite(led2, LOW);

  digitalWrite(led3, HIGH);
  delay(pauza);
  digitalWrite(led3, LOW);
}
```

Kod za vježbu 6.



VJEŽBA BR. 7 SEMAFOR BONUS

```
int crvena=3;
int zuta=2;
int zelena=12;
int crvenaP=6;
int zelenaP=7;

void setup() {
  pinMode(crvena, OUTPUT);
  pinMode(zuta, OUTPUT);
  pinMode(zelena, OUTPUT);

  pinMode(crvenaP, OUTPUT);
  pinMode(zelenaP, OUTPUT);
}

void loop()
{
  digitalWrite(crvena, HIGH); // turn the LED on (HIGH)
  digitalWrite(zelenaP, HIGH);
  delay(5000);

  digitalWrite(zelenaP, LOW)
  delay(500);
  digitalWrite(zelenaP, HIGH)
  delay(500);
  digitalWrite(zelenaP, LOW)
  digitalWrite(crvenaP, HIGH);
  digitalWrite(zuta, HIGH);
  delay(2000);

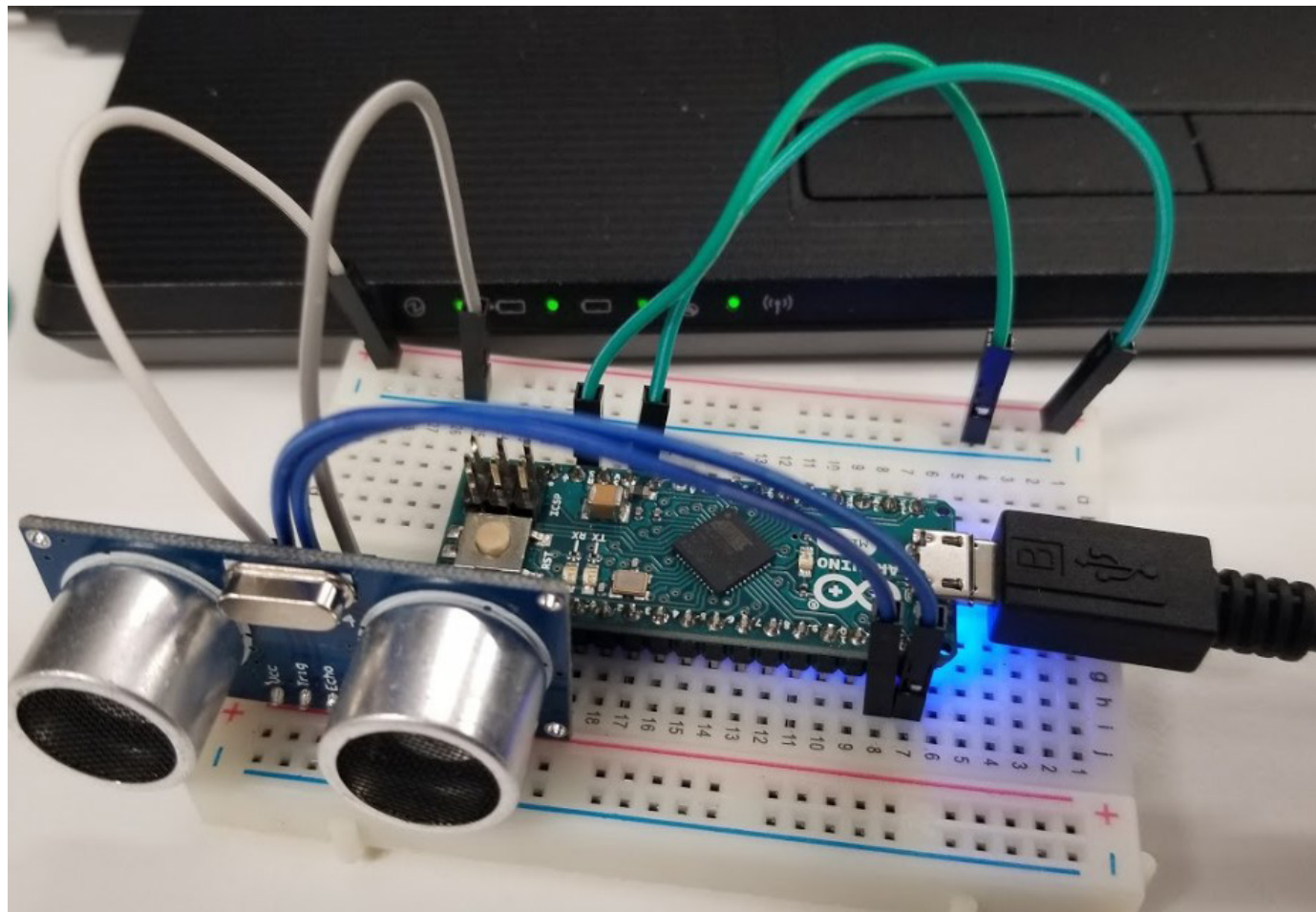
  digitalWrite(crvena, LOW);
  digitalWrite(zuta, LOW);
  digitalWrite(zelena, HIGH);
  delay(4000);
  digitalWrite(zelena, LOW);
  delay(500);
  digitalWrite(zelena, HIGH);
  delay(500);
  digitalWrite(zelena, LOW);
  delay(500);
  digitalWrite(zelena, HIGH);
  delay(500);
  digitalWrite(zelena, LOW);
  delay(500);
  digitalWrite(zelena, HIGH);
  delay(500);

  digitalWrite(zelena, LOW);
  digitalWrite(zuta, HIGH);
  delay(1000);
  digitalWrite(zuta, LOW);
  digitalWrite(crvenaP, LOW);
}
```

Kod za vježbu 7.

VJEŽBA BR. 8 MJERENJE RAZDALJINE

Cilje vježbe je izmjeriti razdaljinu od senzora do proizvoljnih predmeta koristeći ultrazvučni senzor. Povezivanje komponente je veoma jednostavno. Senzor ima četiri pin ai to GND, VCC, TRIG, ECHO. Potrebno ih je povezati na uzemljenje(GND), 5V, OUTPUT , INPUT pinove respektivno.



Izgled vježbe 8.

Kod potreban za izvršenje ove vježbe nalazi se ispod. Prije kopiranja koda pokušajte samostalno da uradite ovaj zadatak.



```
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, cm, inches;

void setup()
{
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  // Convert the time into a distance
  cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343
  inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

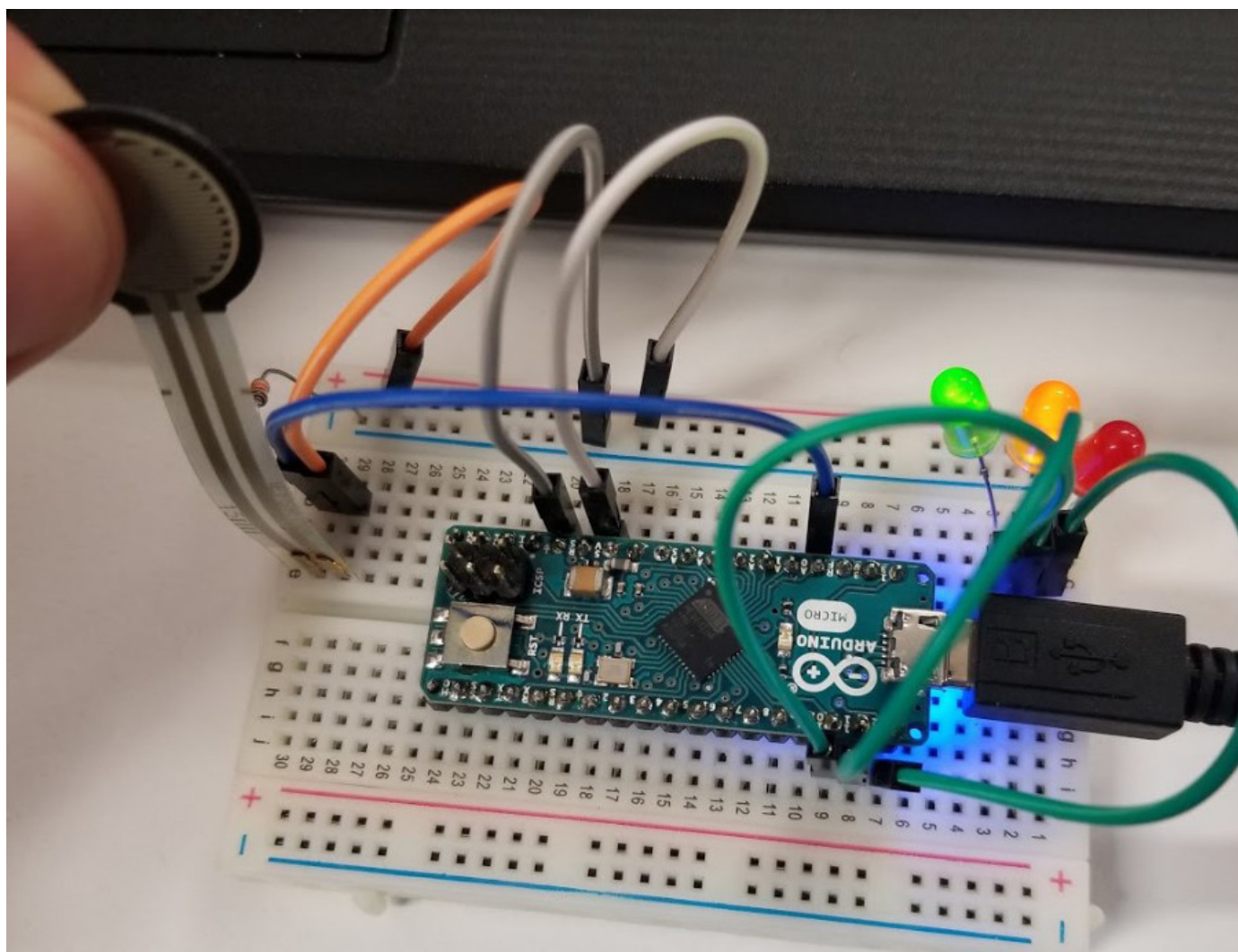
  delay(250);
}
```

Kod za vježbu 8.

VJEŽBA BR. 9 MJERENJE PRITISKA

Cilje vježbe je izmjeriti pritisak koji na senzor vrši spoljašnja sila ili u ovom slučaju pritisak naše ruke. Povezivanje komponente je veoma jednostavno. Senzor ima dva pina i to VCC, GND. Povezujemo VCC na 5V a GND pin na jedan od ulaznih pinova a zatim preko otpornik 10k na uzemljenje (GND). Zatim napišemo kod koji će da očitava vrijednost sa senzora i u zavisnosti od toga da uključuje i isključuje pojedine LED diode.

Na slici je prikazano kako je izvedena data vježba i kako se na osnovu pritika spoljašnje sile na senzor određene LED diode uključuju.



Vježba "Mjerenje pritiska mehaničke sile"



Kada smo spojili komponente kaon a slici pristupamo pisanju koda za datu vježbu. Treba voditi računa da se prije kompajliranja koda kabal ne priključuje na Arduino uređaj. Primjer koda za ovu vježbu nalazi se na sljedećoj slici.

```
int forcePin = A0;
int forceReading;
int red=5;
int yellow=6;
int green=7;
void setup(void)
{
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
void loop(void)
{
  digitalWrite(red,LOW);
  digitalWrite(yellow,LOW);
  digitalWrite(green,LOW);
  forceReading = analogRead(forcePin);
  if(forceReading<=300 and forceReading >0)
  {
    digitalWrite(red,LOW);
    digitalWrite(yellow,LOW);
    digitalWrite(green,HIGH);
  }
  if(forceReading<=600 and forceReading >300)
  {
    digitalWrite(red,LOW);
    digitalWrite(yellow,HIGH);
    digitalWrite(green,HIGH);
  }
  if(forceReading>600)
  {
    digitalWrite(red,HIGH);
    digitalWrite(yellow,HIGH);
    digitalWrite(green,HIGH);
  }
}
```

Kod za vježbu 9.

ZAJEDNIČKI PROJEKT FONDACIJE BUDUĆNOSTI U BOSNI I HERCEGOVINI I LA TROBE UNIVERZITETA

KONSTRUISAO I OBRADIO:

DIPL. ING. EL. MOMČILO AMOVIĆ,

KOORDINATOR VOLONTERSKOG PROGRAMA FONDACIJE
BUDUĆNOSTI U BOSNI I HERCEGOVINI

