

# REMOTE CONTROL AND LCD



# CILJEVI

- Upoznati se sa osnovnim osobinama IR predajnika i prijemnika
- Znati koristiti Arduino IRremote biblioteku za rad sa IR prijemnikom i predajnikom.
- Upoznati se se osnovnim osobinama LCD-a i načinima povezivanja sa Arduino Uno razvojnom pločom
- Znati koristiti LiquidCrystal biblioteku za rad sa LCD-om.

# REFERENCE

Informacije o Arduino IRreceive biblioteci i IR receiveru :

<https://github.com/z3t0/Arduino-IRremote>

<https://arduino-info.wikispaces.com/file/view/IR-Receiver-AX-I838HS.pdf/264668680/IR-Receiver-AX-I838HS.pdf>

Informacije o Arduino LCD biblioteci:

<https://www.arduino.cc/en/Reference/LiquidCrystal>

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

# IR KOMUNIKACIJA

IR komunikacija je u širokoj upotrebi.

Jednostavna za upotrebu.

Ima puno korisnih primjena.

Najpoznatija svakodnevna upotreba je u daljinskim upravljačima.

Sa jednostavnim IR predajnikom i prijemnikom može se upravljati:

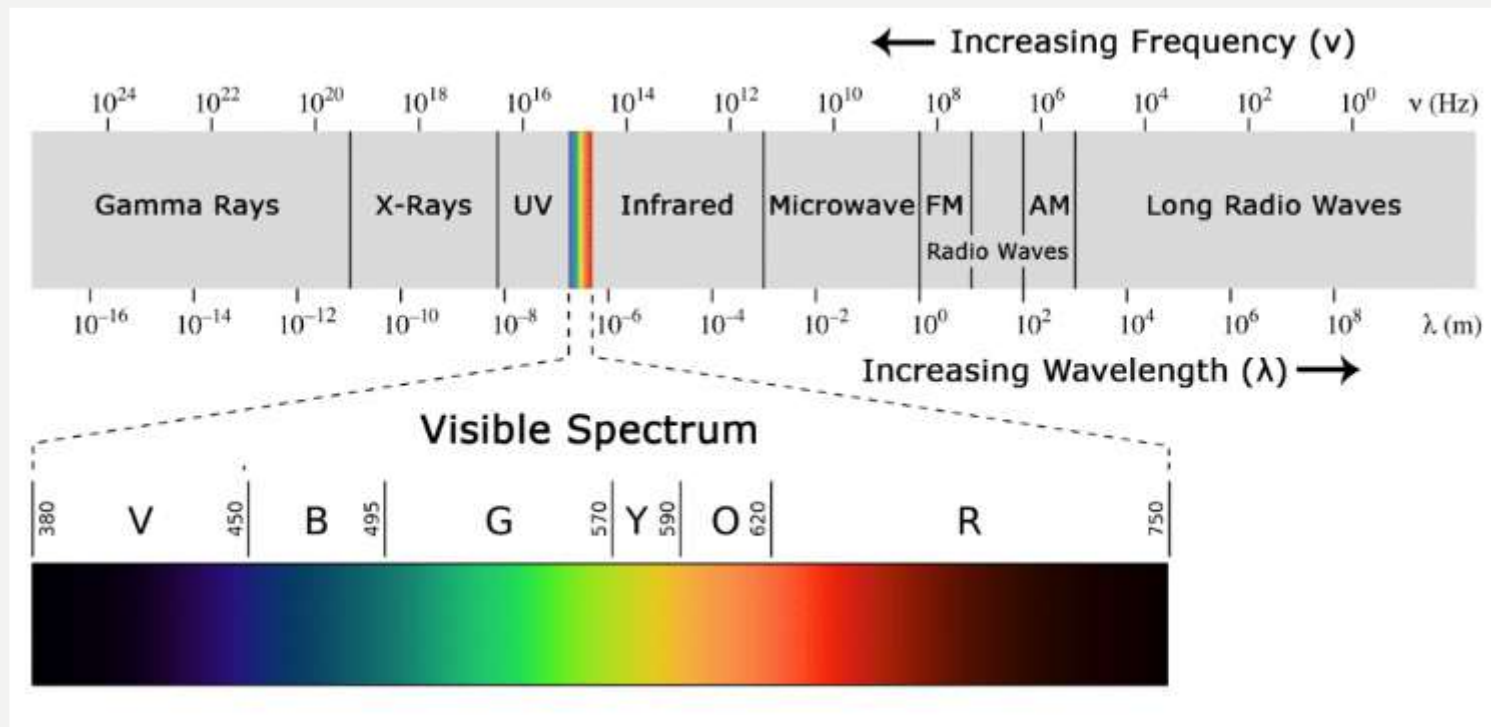
- AV uređajima kao TV, VCR, CD, DVD, etc.
- kućnim aparatima kao klima, ventilator, etc.
- multimedijском opremom

# ŠTO JE IR (INFRARED)?

IR je oblik svjetlosti sličan onom koji vidimo oko nas.

Jedina razlika je u frekvenciji, odnosno talasnoj dužini.

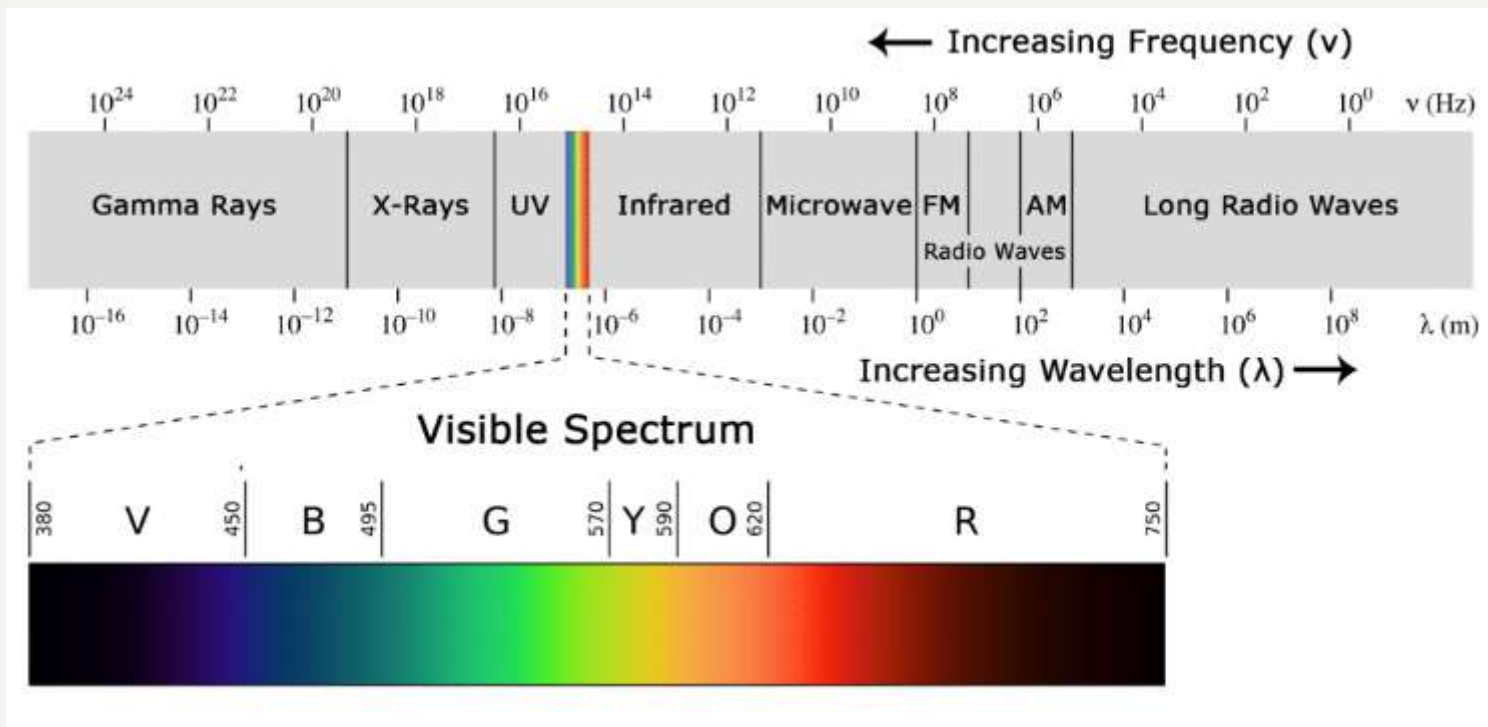
IR zračenje leži izvan opsega vidljive svjetlosti, tako da ga ljudi ne mogu vidijeti.



# ŠTO JE IR (INFRARED)?

Kako je IR svjetlost, IR komunikacija zahtijeva liniju direktne vidljivosti između predajnika i prijemnika.

Za razliku od WiFi ili Bluetooth, IR komunikacija se ne može obavljati kroz zidove i druge materijale.



# IR DALJINSKI

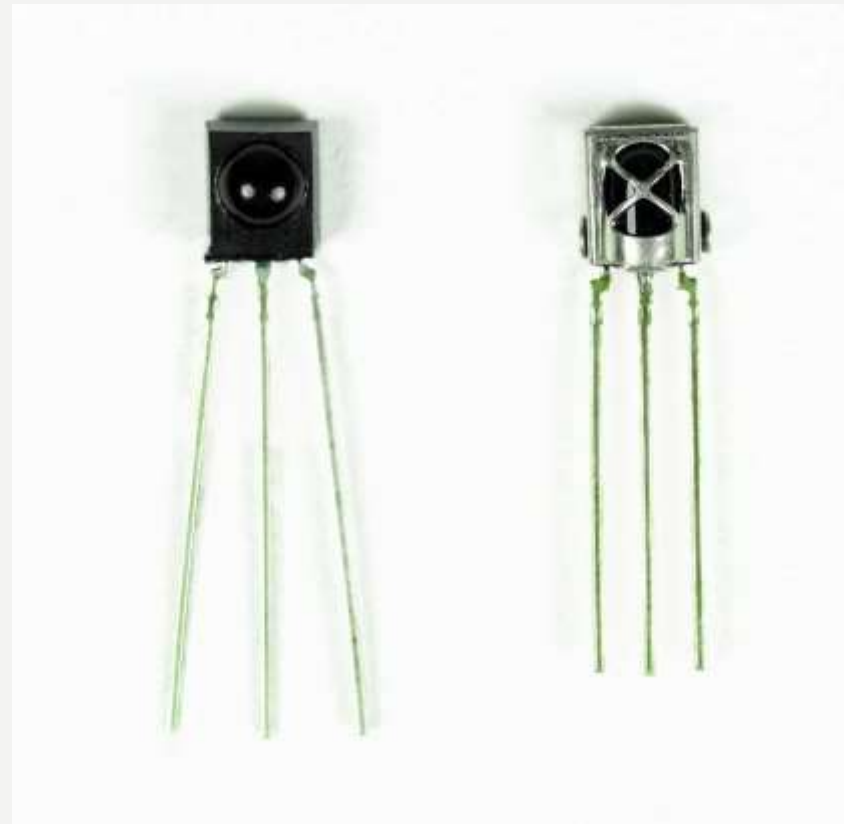
Tipični IR komunikacioni sistem zahtijeva IR predajnik i IR prijemnik.

Predajnik sadrži LED koja emituje svjetlost u IR spektru umjesto u vidljivom spektru.



# IR PRIJEMNIK

IR prijemnik je fotodioda, pojačavač i dekođer koji konvertuju IR svjetlost u električni signal.





# MODULACIJA IR SIGNALA

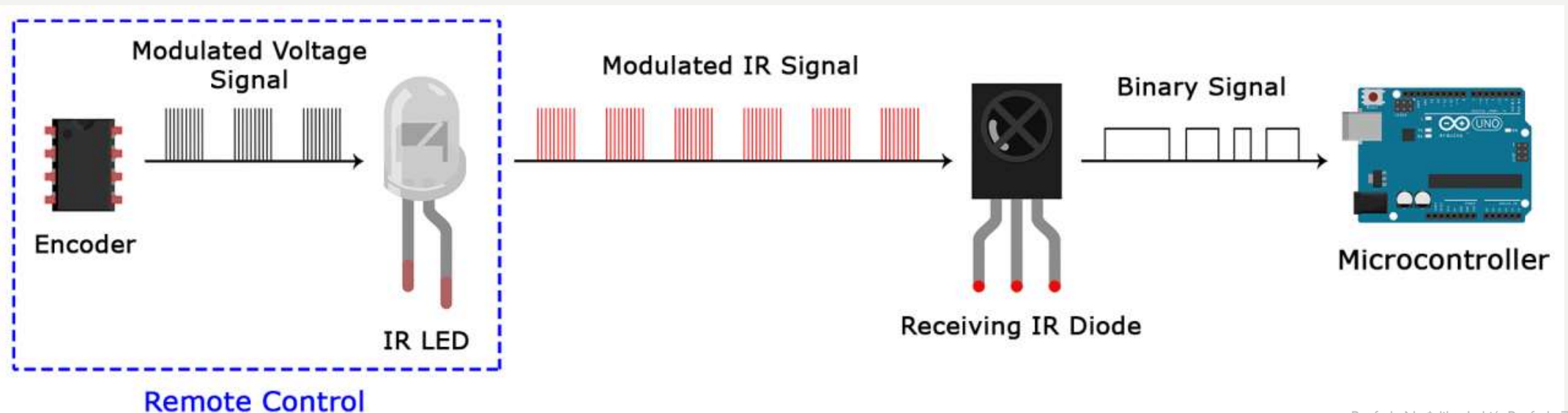
IR svjetlost smituje sunce, sijalice i sve drugo što proizvodi toplotu. To znači da postoji puno IR svjetlosnog šuma svuda oko nas.

Da bi se preduprijedila interferencija šuma i IR signala, IR signal je modulisan.

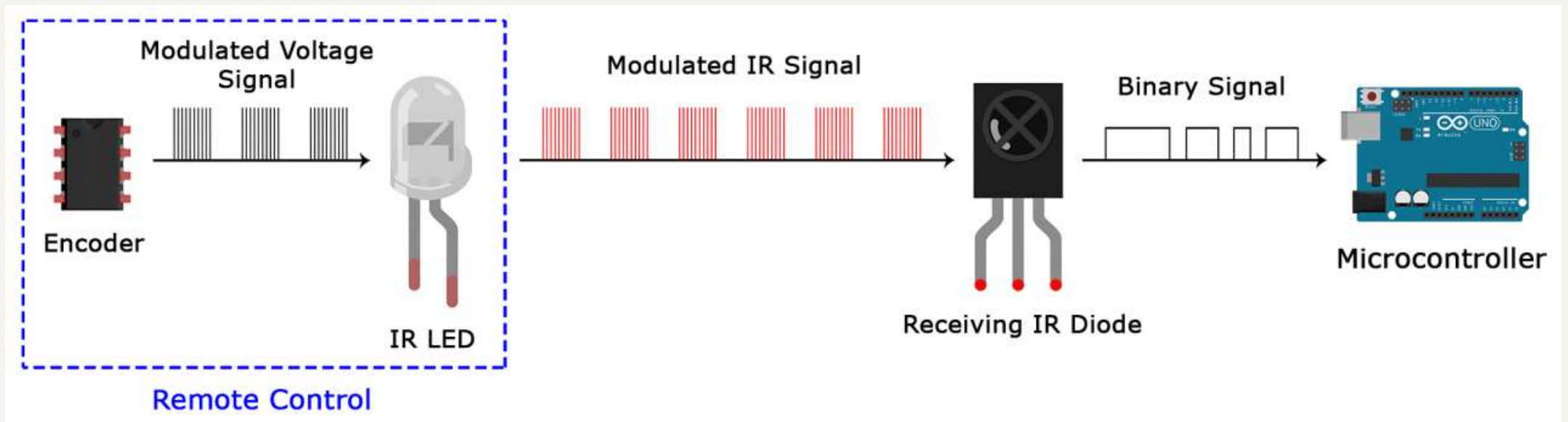
Modulacija IR signala obavlja se pomoću enkodera u IR daljinskom, koji binarni signal konvertuje u modulisani električni signal, koji se šalje ka predajnoj LED.

Predajna LED konvertuje modulisani električni signal u modulisani IR svjetlosni signal.

IR prijemnik demoduliše IR svjetlosni signal i, prije prosljeđivanja informacije mikrokontroleru, konvertuje ga u binarni.



# MODULACIJA IR SIGNALA



Modulisani IR signal je serija IR svjetlosnih impulsa visoke frekvencije, poznate kao frekvencija nosioca (PCM - Pulse Code Modulation frekvencija).

Frekvencija nosioca najčešće iznosi 38kHz, zato što je u prirodi rijetka i stoga se može razlikovati od ambijentnog šuma.

Na ovaj način IR prijemnik će znati da je signal od 38kHz poslat sa predajnika. Nije pokupljen iz okruženja.

Dioda u IR prijemniku detektuje sve frekvencije IR svjetlosti, ali postoji filter propusnik opsega, koji propušta samo IR signal od 38kHz. Prijemnik dalje pojačava modulisani signal i konvertuje ga u binarni.

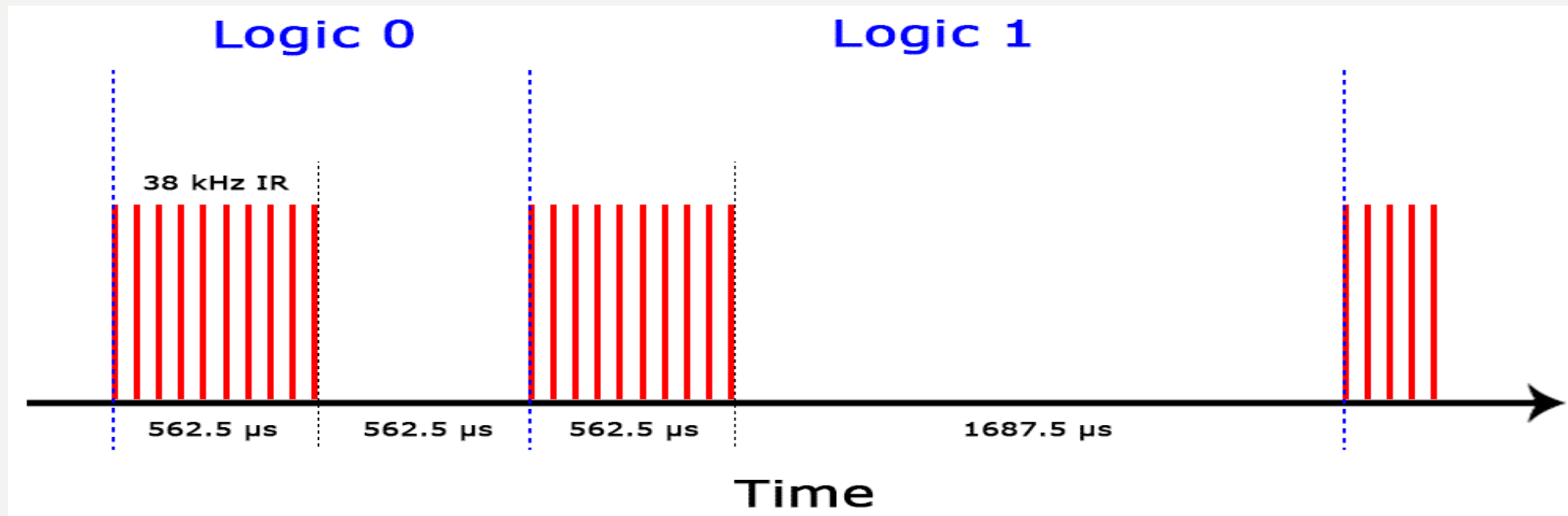
# PROTOKOL IR KOMUNIKACIJE

Obrazac po kojem se modulirani IR signal konvertuje u binarni i obrnuto.

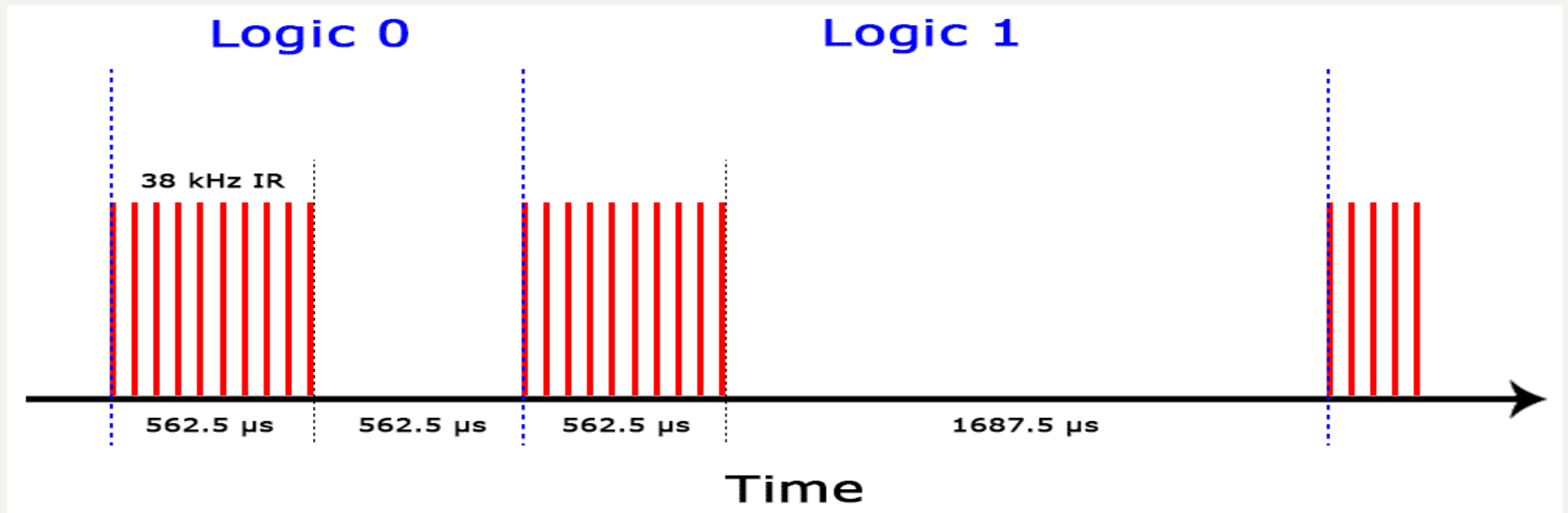
Postoje mnogi protokoli. Sony, Matsushita, NEC i RC5 su najpoznatiji.

NEC protokol je najčešće zastupljen u Arduino projektima, te će poslužiti kao primjer konvertovanja moduliranog IR signala u binarni i obrnuto.

Logička jedinica se predstavlja povorkom impulsa frekvencije 38kHz, trajanja 562.5 $\mu$ s, iza koje slijedi impuls niskog naponskog niva trajanja 1687.5 $\mu$ s. Logička nula se predstavlja povorkom impulsa frekvencije 38kHz, trajanja 562.5 $\mu$ s, iza koje slijedi impuls niskog naponskog nivoa, trajanja 562.5 $\mu$ s.



# PROTOKOL IR KOMUNIKACIJE



Na ovaj način NEC protokol kodira i dekodira binarne podatke u/iz modulisanog signala.

Drugi protokoli se razlikuju samo u trajanju povorke impulsa i trajanju impulsa niskog naponskog nivoa.

# IR KODOVI

Svaki put kada se pritisne odgovarajuće dugme na daljinskom upravljaču, generiše se jedinstveni heksadecimalni kod.

Ta informacija se dalje moduliše i šalje, preko IR predajnika, prema IR prijemniku.

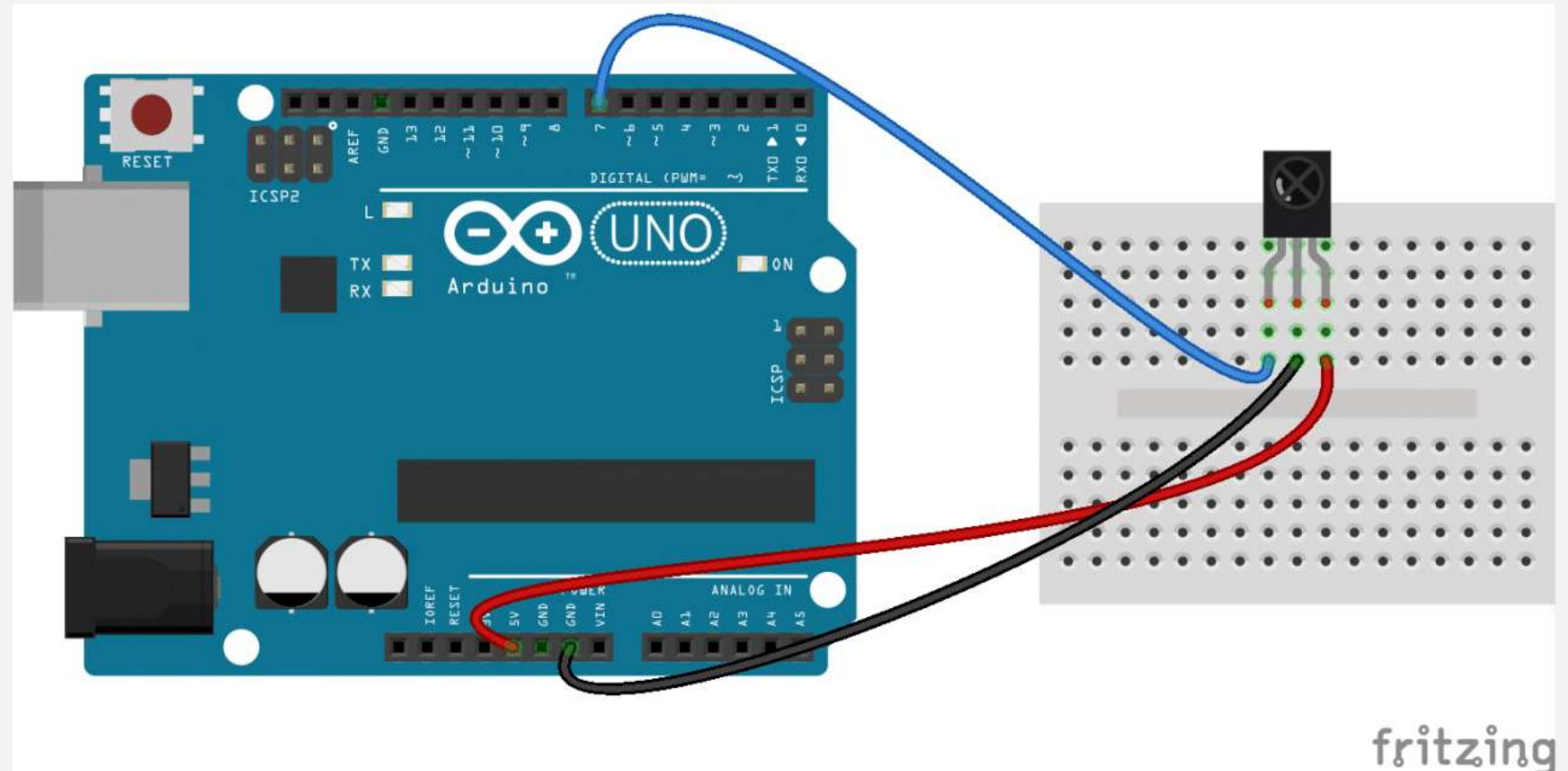
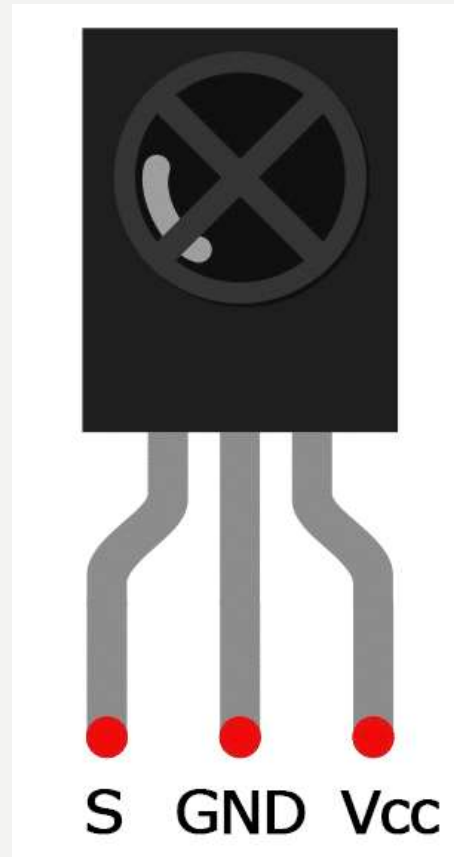
Da bi mogao da dešifruje koje dugme je pritisnuto, prijemni mikrokontroler treba da zna koji kod odgovara svakom dugmetu na daljinskom upravljaču.

Različiti daljiski šalju različite kodove za komandna dugmad, te je stoga potrebno odrediti koji kod generiše pojedino dugme na daljinskom kojime raspolazete.

Ukoliko uspijete pronaći datasheet za Vaš daljiski, podaci o kodovima će biti izlistani u njemu.

Ukoliko ne, postoji Arduino skeč, koji će, za najpopularnije daljiske upravljače, kad pritisnete pojedino dugme, informaciju o kodu oštampati na serijskom monitoru.

# KAKO POVEZATI IR PRIJEMNIK I ARDUINO



# INSTALIRANJE IRREMOTE BIBLIOTEKE

Potrebno je preuzeti ZIP fajl odavde:

<https://www.arduinolibraries.info/libraries/i-rremote>

ili <https://github.com/z3t0/Arduino-IRremote>

Otvoriti Arduino IDE, otići na Sketch -> Include Library -> Add .ZIP Library

Selektovati Irremote ZIP fajl koji ste preuzeli sa linka iznad.

Klik na Open

# PR: PRONALAZENJE KODOVA

```
#include <IRremote.h>
```

```
const int RECV_PIN = 7;  
IRrecv irrecv(RECV_PIN);  
decode_results results;
```

```
void setup(){  
  Serial.begin(9600);  
  irrecv.enableIRIn();  
  irrecv.blink13(true);  
}
```

```
void loop(){  
  if (irrecv.decode(&results)){  
    Serial.println(results.value,  
  HEX);  
    irrecv.resume();  
  }  
}
```

**IRrecv irrecv(receivePin)** – kreira receiver objekt (ime po izboru).

**irrecv.enableIRIn()** – započinje sa prijemom (osluškivanjem). Ova funkcija omogućuje tajmerski interapt, koji uzima malo procesorskog vremena svakih 50 µs.

**irrecv.decode(&results)** – Pokušava da primi IR kod. Vraća TRUE ako je kod primljen i FALSE ako nije. Kada je kod primljen, informacija je smještena u "results".

**results.decode\_type:** Neke vrijednosti: **NEC**, **SONY**, **RC5**, **RC6**, **UNKNOWN**, ...

**results.value:** Pročitani IR kod

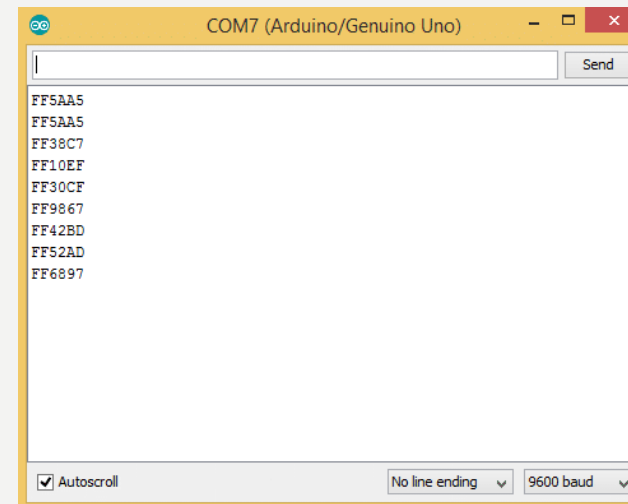
**results.bits:** Broj bita u pročitanoj kodu

**results.rawbuf:** Niz koji sadrži trajanja IR impulsa

**results.rawlen:** Broj elemenata rawbuf niza

**irrecv.resume()** – Ova funkcija se mora pozvati nakon prijema IR koda. Njome se resetuje prijemnik i priprema za prijem sljedećeg koda.

**irrecv.blink13(true)** – Omogućuje blinkanje LED tokom prijema IR koda.



Key	Code
CH-	0xFFA25D
CH	0xFF629D
CH+	0xFFE21D
<<	0xFF22DD
>>	0xFF02FD
>	0xFFC23D
-	0xFFE01F
+	0xFFA857
EQ	0xFF906F
100+	0xFF9867
200+	0xFFB04F
0	0xFF6897
1	0xFF30CF
2	0xFF18E7
3	0xFF7A85
4	0xFF10EF
5	0xFF38C7
6	0xFF5AA5
7	0xFF42BD
8	0xFF4AB5
9	0xFF52AD



# PR: PRONALAZENJE PROTOKOLA

```
#include <IRremote.h>

const int RECV_PIN = 7;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();
  irrecv.blink13(true);
}

void loop(){
  if (irrecv.decode(&results)){
    Serial.println(results.value, HEX);
    switch (results.decode_type){
      case NEC: Serial.println("NEC"); break ;
      case SONY: Serial.println("SONY"); break ;
      case RC5: Serial.println("RC5"); break ;
      case RC6: Serial.println("RC6"); break ;
      case DISH: Serial.println("DISH"); break ;
      case SHARP: Serial.println("SHARP"); break ;
      case JVC: Serial.println("JVC"); break ;
      case SANYO: Serial.println("SANYO"); break ;
      case MITSUBISHI: Serial.println("MITSUBISHI"); break ;
      case SAMSUNG: Serial.println("SAMSUNG"); break ;
      case LG: Serial.println("LG"); break ;
      case WHYNTER: Serial.println("WHYNTER"); break ;
      case AIWA_RC_T501: Serial.println("AIWA_RC_T501"); break ;
      case PANASONIC: Serial.println("PANASONIC"); break ;
      case DENON: Serial.println("DENON"); break ;
      default:
        case UNKNOWN: Serial.println("UNKNOWN"); break ;
    }
    irrecv.resume();
  }
}
```

Poznavati protokol koji vaš daljinski koristi može biti upotrebljivo ako želite da radite na nekim naprednijim projektima.

Ili ste možda samo radoznali.

Dati program će identifikovati koji protokol koristi Vaš daljinski.

Trebalo bi da radi čak i sa daljinskima iz vaše kuće.

# PR: ŠTAMPANJE KOJE DUGME

```
#include <IRremote.h>
```

```
const int RECV_PIN = 7; IRrecv irrecv(RECV_PIN);  
decode_results results; unsigned long key_value = 0;
```

```
void setup(){  
  Serial.begin(9600);  
  irrecv.enableIRIn(); irrecv.blink13(true);  
}
```

```
void loop(){  
  if (irrecv.decode(&results)){
```

```
    if (results.value == 0xFFFFFFFF)  
      results.value = key_value;
```

```
    switch(results.value){  
      case 0xFFA25D: Serial.println("CH-"); break;  
      case 0xFF629D: Serial.println("CH"); break;  
      case 0xFFE21D: Serial.println("CH+"); break;  
      case 0xFF22DD: Serial.println("|<<"); break;  
      case 0xFF02FD: Serial.println(">>|"); break ;  
      case 0xFFC23D: Serial.println(">|"); break ;  
      case 0xFFE01F: Serial.println("-"); break ;  
      case 0xFFA857: Serial.println("+"); break ;  
      case 0xFF906F: Serial.println("EQ"); break ;  
      case 0xFF6897: Serial.println("0"); break ;  
      case 0xFF9867: Serial.println("100+"); break ;  
      case 0xFFB04F: Serial.println("200+"); break ;  
      case 0xFF30CF: Serial.println("1"); break ;  
      case 0xFF18E7: Serial.println("2"); break ;  
      case 0xFF7A85: Serial.println("3"); break ;  
      case 0xFF10EF: Serial.println("4"); break ;  
      case 0xFF38C7: Serial.println("5"); break ;  
      case 0xFF5AA5: Serial.println("6"); break ;  
      case 0xFF42BD: Serial.println("7"); break ;  
      case 0xFF4AB5: Serial.println("8"); break ;  
      case 0xFF52AD: Serial.println("9"); break ;
```

```
    }  
    key_value = results.value;  
    irrecv.resume();
```

```
  }  
}
```

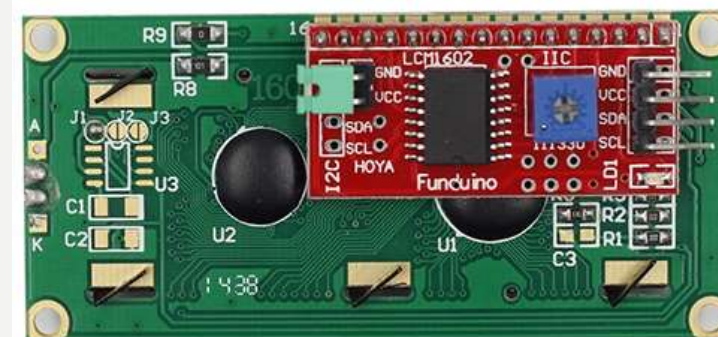
Proširenje koda iz prvog primjera tako da štampa oznaku dugmeta, umjesto hexadecimalnog koda.

# LCD

Standard



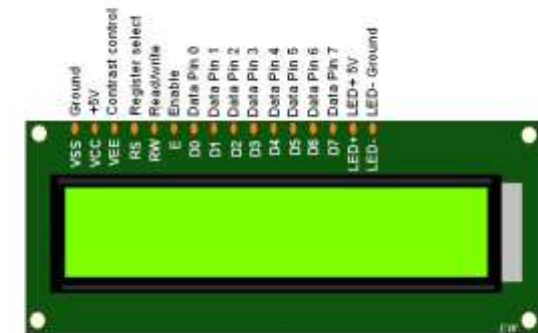
I2C (two wire)



# STANDARD LCD

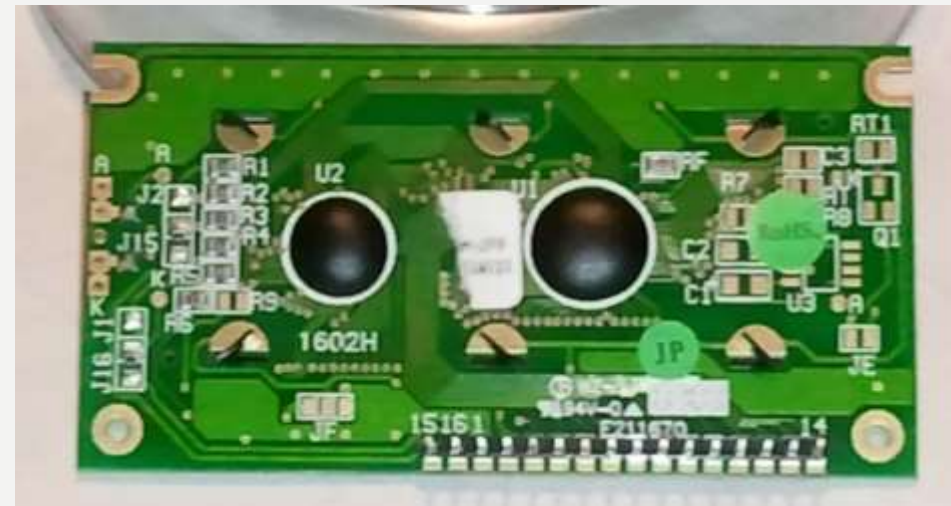
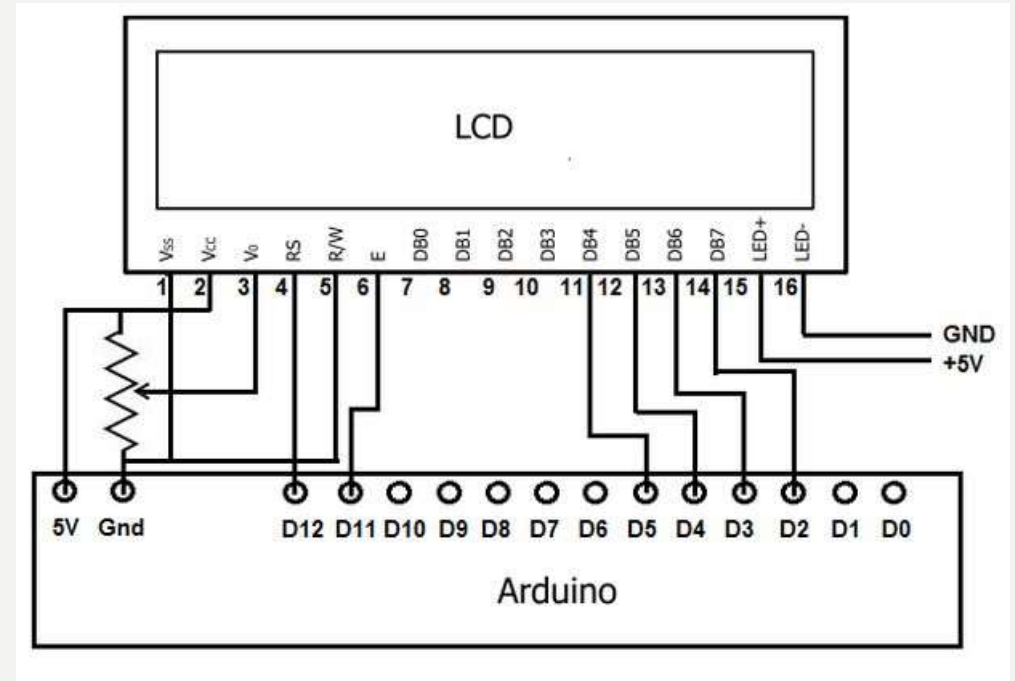
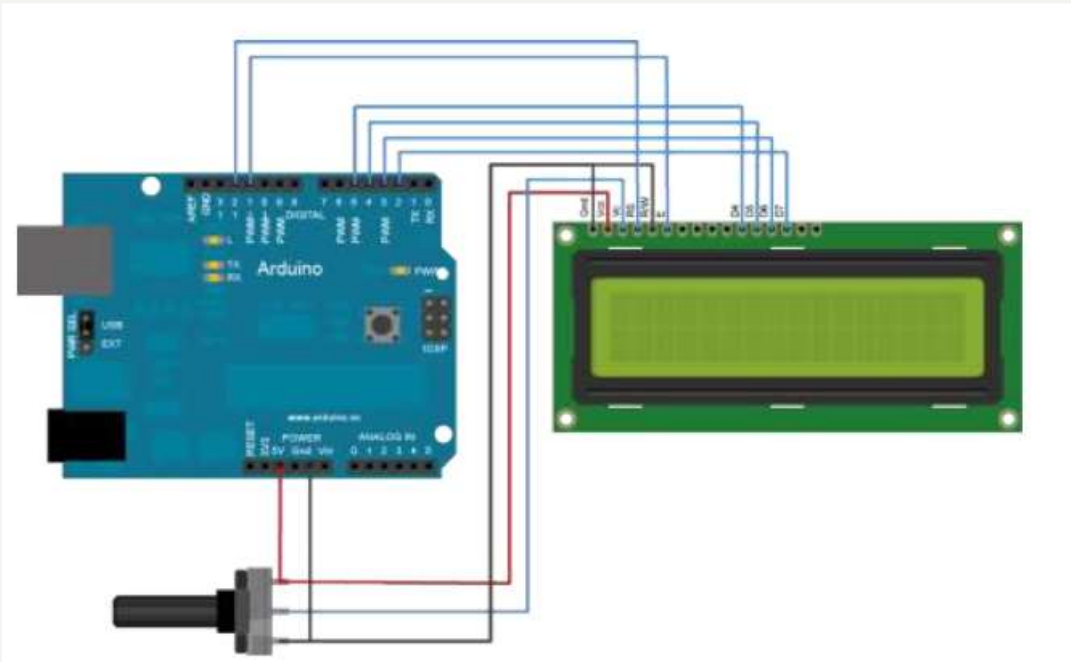
Pinovi standardnog LCD-a

Pin No.	Symbol	Function
1	V <sub>SS</sub>	Ground
2	V <sub>CC</sub>	Power supply
3	V <sub>0</sub>	<b>Contrast</b>
4	RS	Select Display Data("H") or Instructions("L")
5	R/W	Read or Write Select Signal
6	E	Read/Write Enable Signal
7	DB0	Display Data Signal
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED + (A)	Please also refer to 6.1 PCB drawing and description.
16	LED - (K)	Please also refer to 6.1 PCB drawing and description.



# STANDARD LCD

Povezivanje LCD-a sa Arduino Uno razvojnom pločom





# HELLO WORLD!

```
/*  
LiquidCrystal Library - Hello World
```

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and shows the time.

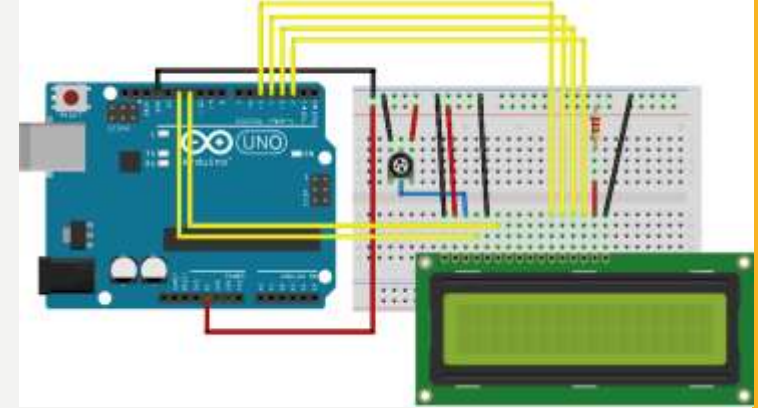
The circuit:

- \* LCD RS pin to digital pin 12, \* LCD Enable pin to digital pin 11
- \* LCD D4 pin to digital pin 5, \* LCD D5 pin to digital pin 4
- \* LCD D6 pin to digital pin 3, \* LCD D7 pin to digital pin 2
- \* LCD R/W pin to ground, \* LCD VSS pin to ground
- \* LCD VCC pin to 5V, \* 10K resistor:
  - \* ends to +5V and ground
  - \* wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis  
library modified 5 Jul 2009 by Limor Fried (<http://www.ladyada.net>)  
example added 9 Jul 2009 by Tom Igoe  
modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/LiquidCrystal>  
\*/



```
// include the library code:  
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 2);  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}
```

```
void loop() {  
  // set the cursor to column 0, line 1  
  // (note: line 1 is the second row, since counting begins with 0):  
  lcd.setCursor(0, 1);  
  // print the number of seconds since reset:  
  lcd.print(millis() / 1000);  
}
```

# I2C LCD

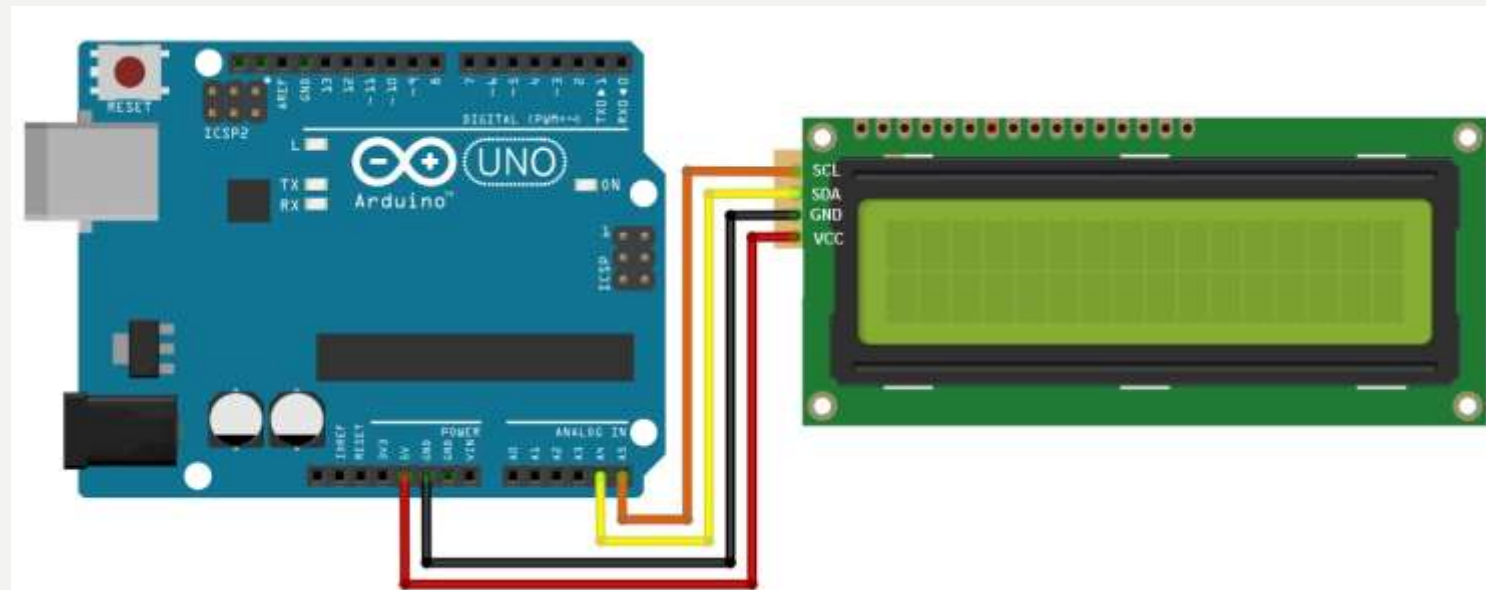
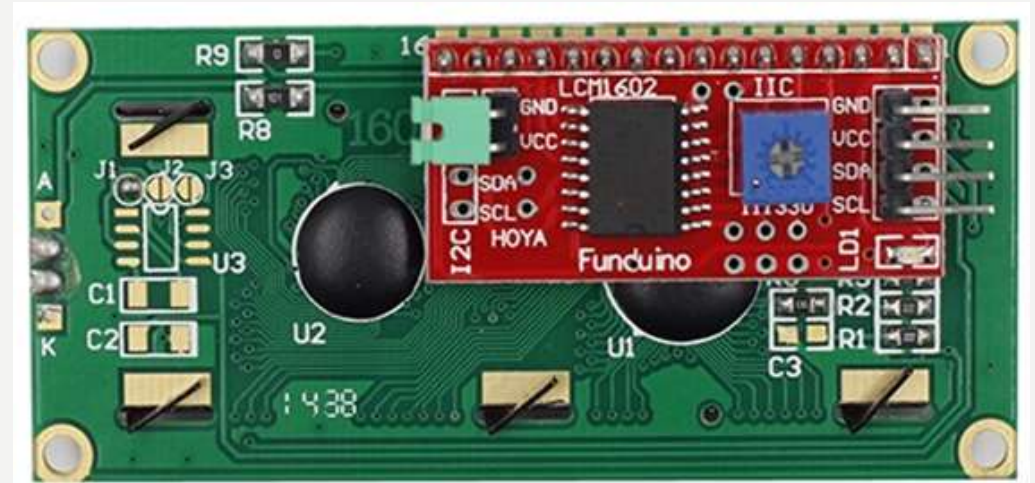
GND: masa

VCC: Napajanje

SDA: DATA signal of I2C-bus sistem

SCL: CLOCK signal of I2C-bus system

Potenciometer za podešavanje kontrasta



# I2C LCD – INSTALIRANJE BIBLIOTEKE

Potrebno je preuzeti ZIP fajl odavde:

[https://github.com/johnrickman/LiquidCrystal\\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C)

Otvoriti Arduino IDE, otići na Sketch -> Include Library -> Add .ZIP Library

Selektovati Irremote ZIP fajl koji ste preuzeli sa linka iznad.

Klik na Open



# I2C SCANNER

Ovim primjerom skenira se I2C adresa upotrijebljena za Vaš displej.

Povežite displej u skladu sa slikom ispod.

Upišite prikazani skeč u Vaš Arduino.

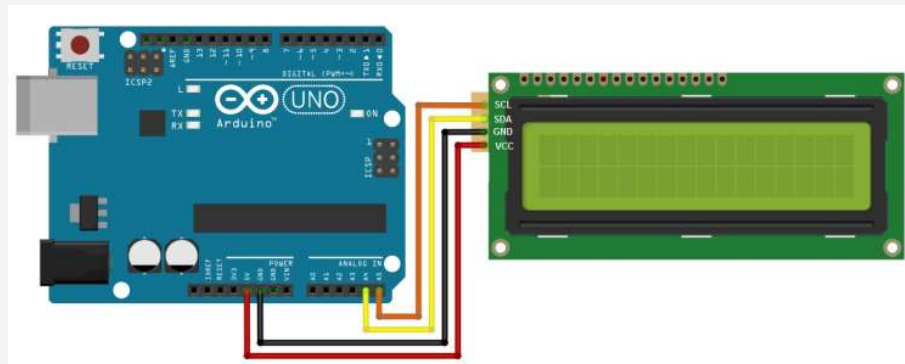
Kliknite na "Serial Monitor" komandno dugme u gornjem desnom uglu.

Podesite brzinu na 115200.

Pritisnite Reset dugme na Arduinou.

Rezultat može da izgleda kao prikazano ispod:

```
I2C scanner. Scanning ...  
Found address: 39 (0x27)  
Done.  
Found 1 device(s).
```



```
// I2C Scanner  
#include <Wire.h>  
  
void setup() {  
  Serial.begin (115200);  
  
  // Leonardo: wait for serial port to connect  
  while (!Serial) {}  
  
  Serial.println (); Serial.println ("I2C scanner. Scanning ...");  
  byte count = 0;  
  
  Wire.begin();  
  for (byte i = 8; i < 120; i++)  
  {  
    Wire.beginTransmission (i);  
    if (Wire.endTransmission () == 0)  
    {  
      Serial.print ("Found address: ");  
      Serial.print (i, DEC);  
      Serial.print (" (0x");  
      Serial.print (i, HEX);  
      Serial.println ("");  
      count++;  
      delay (1); // maybe unneeded?  
    } // end of good response  
  } // end of for loop  
  Serial.println ("Done.");  
  Serial.print ("Found ");  
  Serial.print (count, DEC);  
  Serial.println (" device(s).");  
} // end of setup  
  
void loop() {}
```

# HELLO WORLD I2C!

```
/*  
LiquidCrystal Library - Hello World
```

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and shows the time.

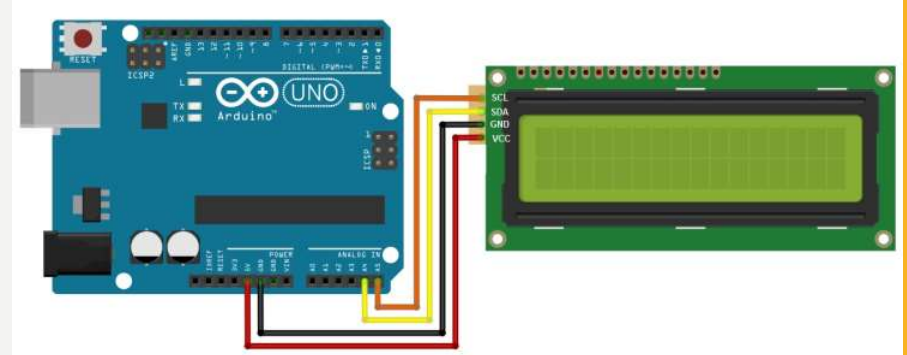
The circuit:

- \* LCD RS pin to digital pin 12, \* LCD Enable pin to digital pin 11
- \* LCD D4 pin to digital pin 5, \* LCD D5 pin to digital pin 4
- \* LCD D6 pin to digital pin 3, \* LCD D7 pin to digital pin 2
- \* LCD R/W pin to ground, \* LCD VSS pin to ground
- \* LCD VCC pin to 5V, \* 10K resistor:
  - \* ends to +5V and ground
  - \* wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by David A. Mellis  
library modified 5 Jul 2009 by Limor Fried (<http://www.ladyada.net>)  
example added 9 Jul 2009 by Tom Igoe  
modified 22 Nov 2010 by Tom Igoe

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/LiquidCrystal>  
\*/



```
// include the library code:  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

```
// initialize the library with the numbers of the interface pins  
LiquidCrystal_I2C lcd(0x3F,16,2);
```

```
void setup() {  
  // initialize the lcd  
  lcd.init();  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 2);  
  //To power on backlight  
  lcd.backlight();  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  // set the cursor to column 0, line 1  
  // (note: line 1 is the second row, since counting begins with 0):  
  lcd.setCursor(0, 1);  
  // print the number of seconds since reset:  
  lcd.print(millis() / 1000);  
}
```

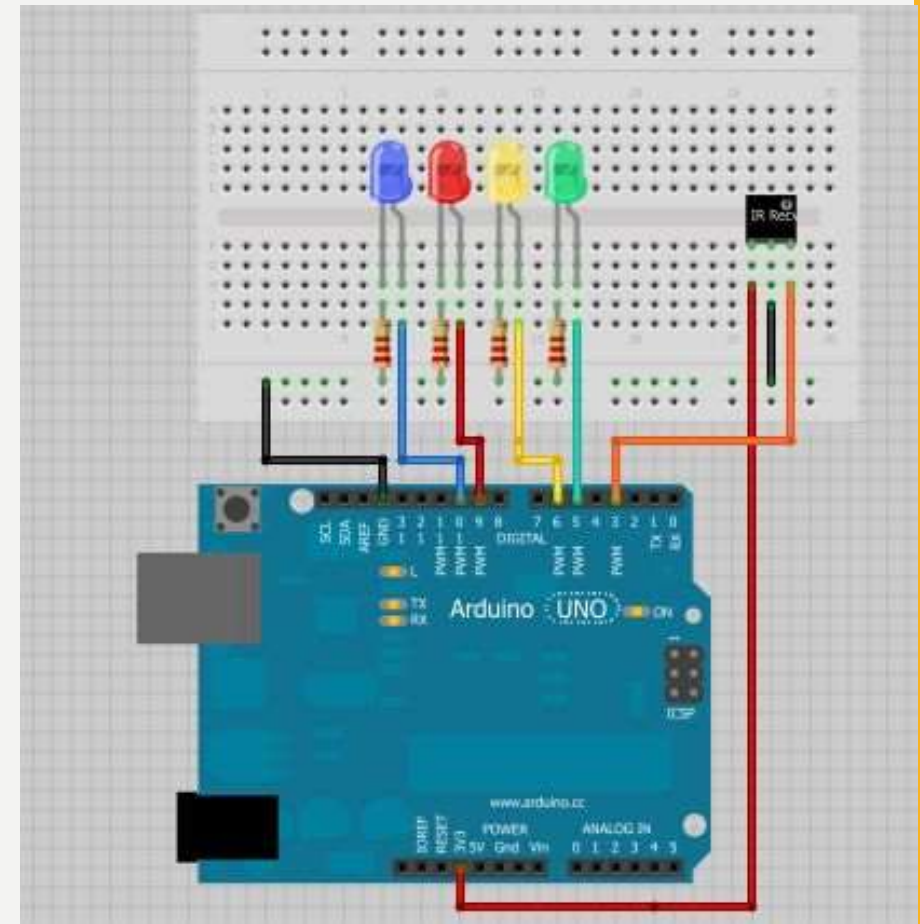
# PRIMJER ZA VJEŽBU

I. Kontrolisanje rada LED pomoću daljiskog upravljača.

Ako je dioda isključena pritisak na odgovarajući taster je uključuje i obrnuto. Za crvenu LED koristiti taster 1, za plavu LED koristiti taster 2, za žutu LED koristiti taster 3.

Na LCD-u ispisivati stanje diode (npr. R on, B off; G off).

Na startu su sve diode isključene. (3-2-1)



# PRIMJER ZA VJEŽBU

2. Kontrolisanje rada koračnog motora pomoću daljinskog upravljača.

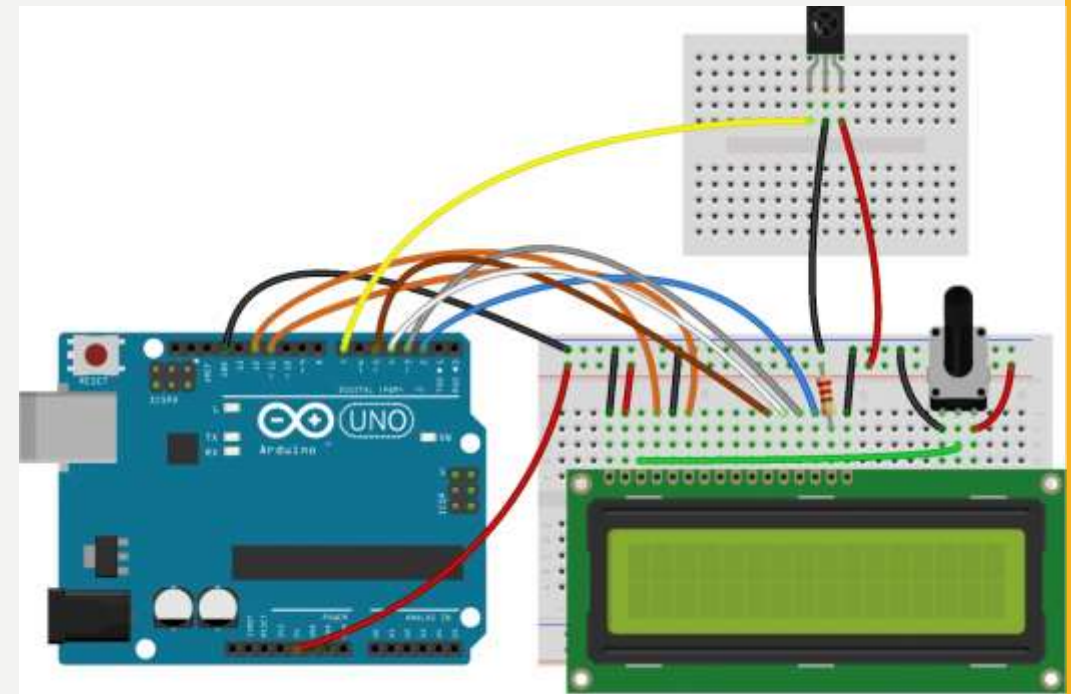
Pritiskom na taster >|| ukoliko motor miruje pokrenuti ga, ukoliko je motor pokrenut zaustaviti ga.

Pritiskom na taster + ubrzati okretanja koračnog motora, pritiskom na taster – usporiti okretanja koračnog motora.

Pritiskom na tastere >>| i |<< mijenjati smjer okretanja koračnog motora.

Informaciju o stanju motora ispisivati na LCD-u: "Motor miruje", "Udesno; Brzinom: 3kor/s", " Ulijevo; Brzinom: 12kor/s".

Na startu definisati početne vrijednosti brzine i smjera i obezbijediti da motor miruje prije prvog pritiskanja tastera >|| . Svaku promjenu stanja motora pr pratiti prigodnim zvučnim signalom. (5-3-2)



# PRIMJER ZA VJEŽBU

## 3. Mjerenje temperature i vlage.

Kada se pritisne taster 1 daljinskog upravljača, na LCD-u prikazati vrijednost temperature u °C i F u trajanju 5 sekundi.

Kada se pritisne taster 2, prikazati vrijednost vlažnosti u % i maksimalnu temperaturu u °C, u trajanju 5 sekundi.

U oba slučaja jednu vrijednost prikazati u gornjem a drugu u donjem redu LCD-a.

U ostalom vremenu na displeju treba da piše: u gornjem redu "1 Temp" u donjem redu "2 Hum". Pri desnom kraju donjeg reda displeja ispisati broj sekundi od startovanja aplikacije.

Promjenu ispisa na displeju prpratiti prigodnim zvučnim signalom.

Obezbijediti da se ispis može izmijeniti u svakom trenutku (i tokom trajanja ispisa od 5 sec). (7-5-3)

