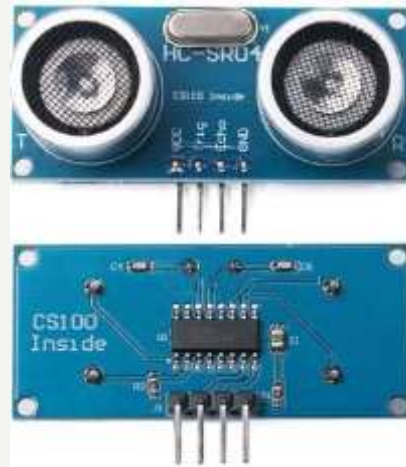


LCD I ULTRASONIČNI SENZOR



HC-SR04

LCD

Standard



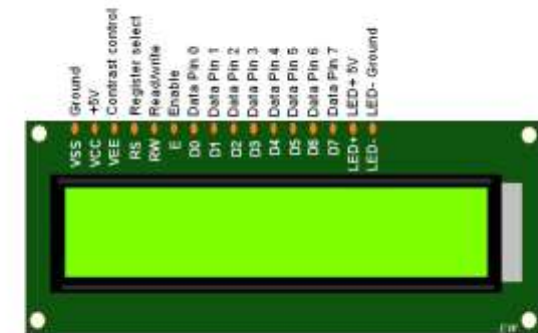
I2C (two wire)



STANDARD LCD

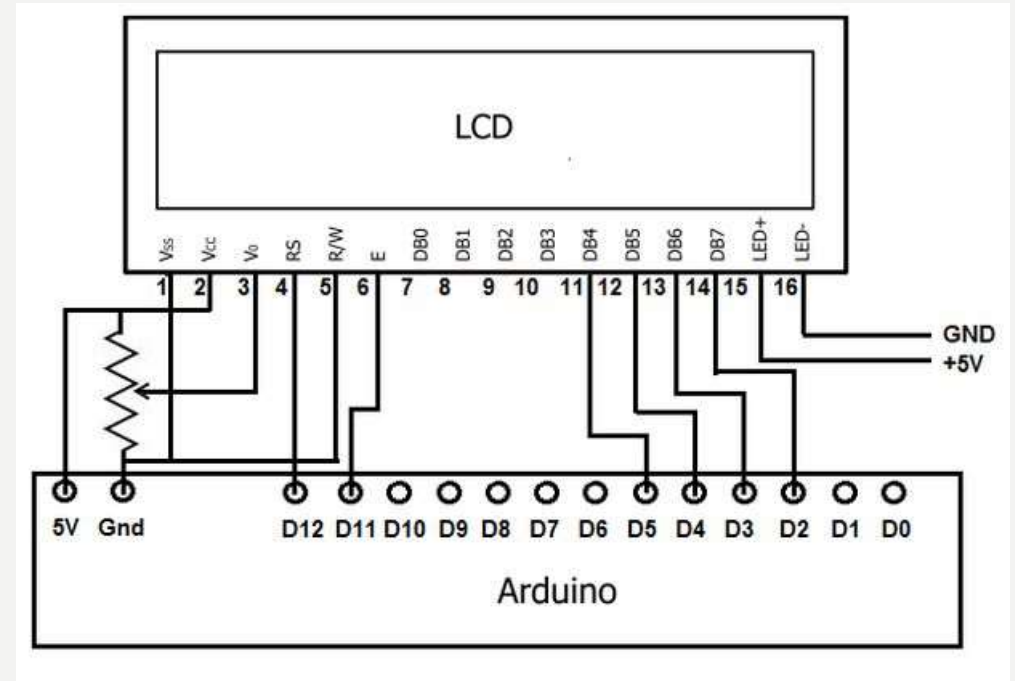
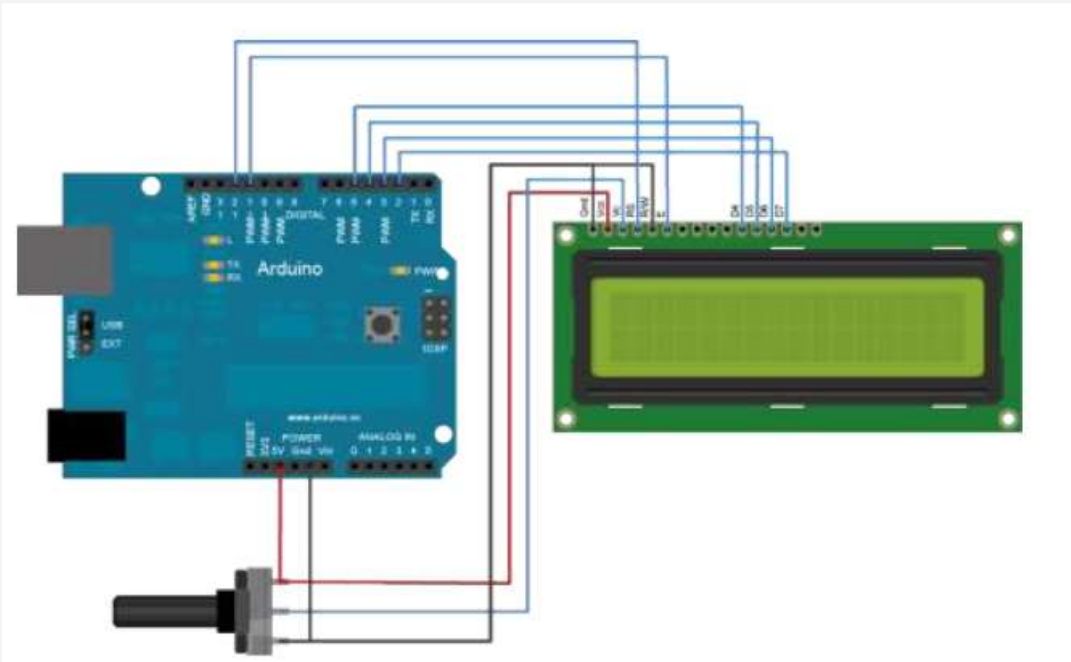
Pinovi standardnog LCD-a

Pin No.	Symbol	Function
1	V _{SS}	Ground
2	V _{CC}	Power supply
3	V ₀	Contrast
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.

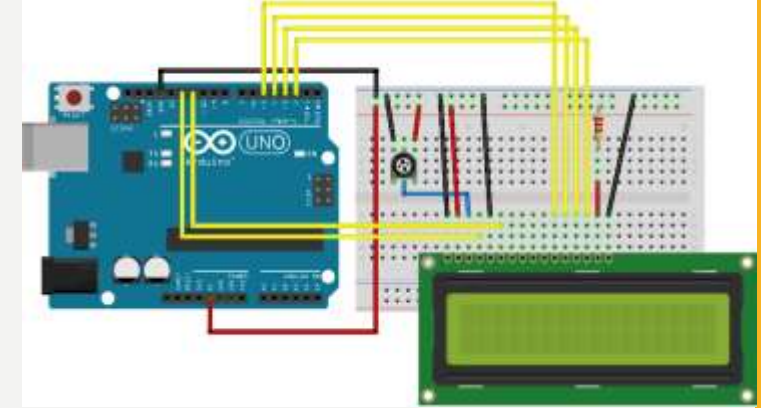
The circuit:

- * 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

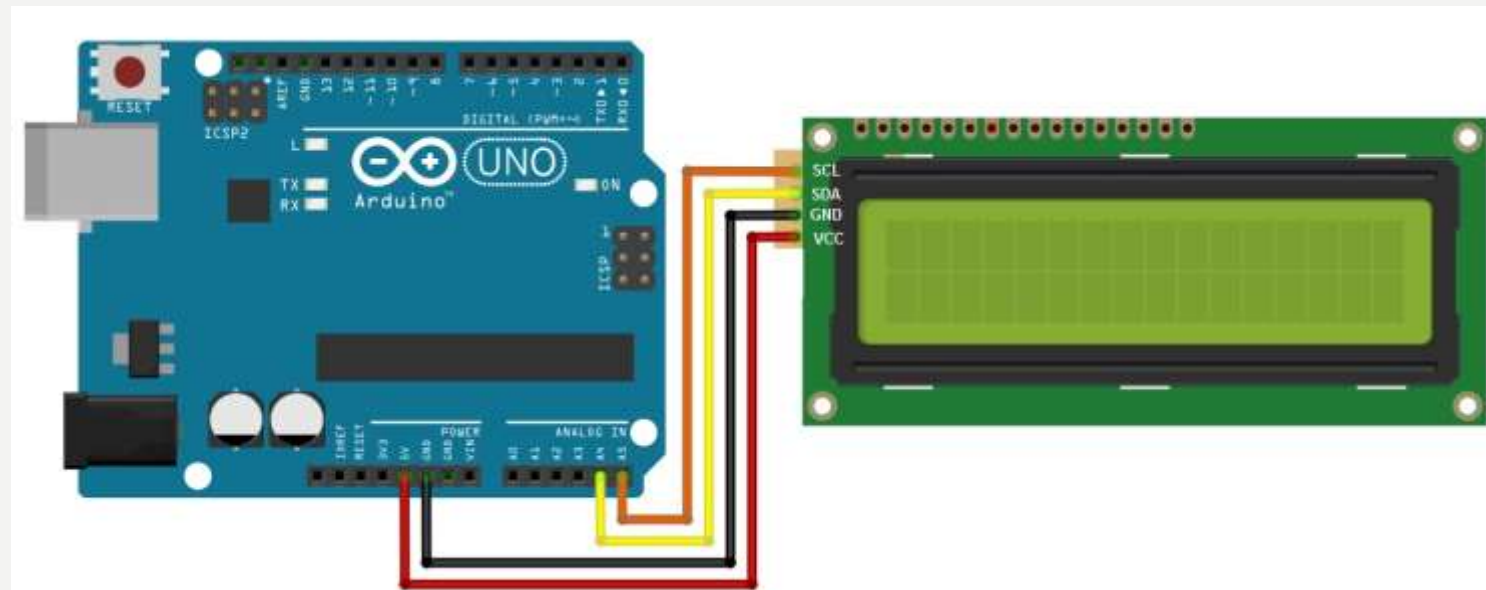
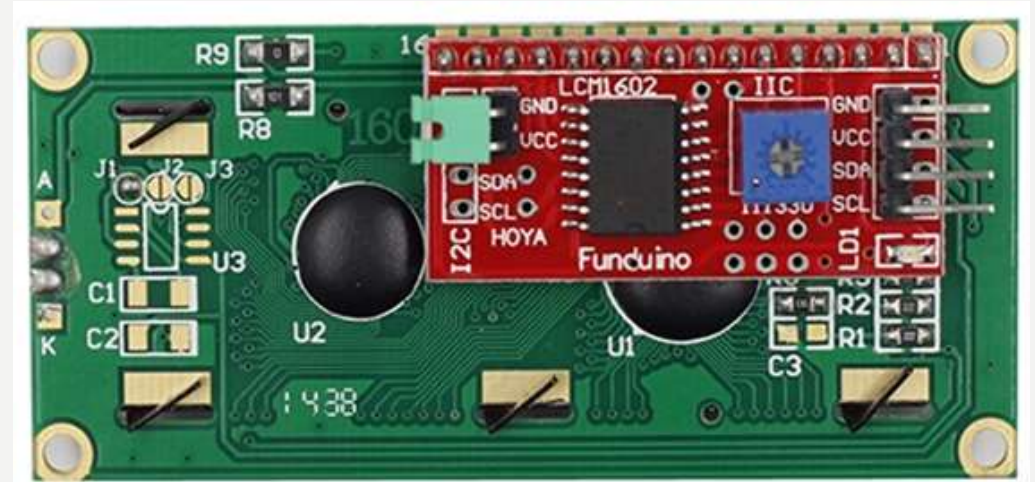
GND: masa

VCC: Napajanje

SDA: DATA signal of I2C-bus sistem

SCL: CLOCK signal of I2C-bus system

Potenciometar za podešavanje kontrasta



I2C LCD – INSTALIRANJE BIBLIOTEKE

Potrebno je preuzeti ZIP fajl odavde:

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.

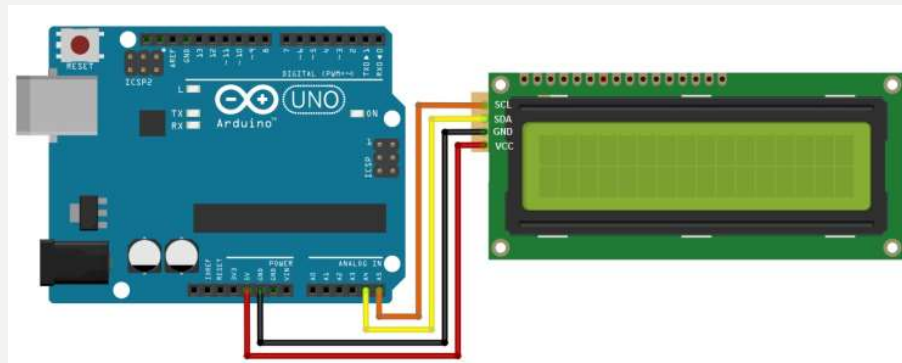
Kliknite na "Serial Monitor" komandno dugme u gornjem desnom uglu.

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.

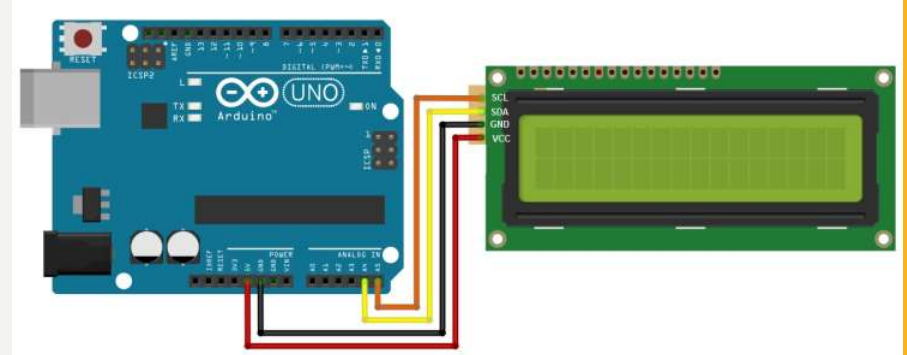
The circuit:

- * 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);  
}
```

HC-SR04 ULTRASONIC SENSOR

- jeftin sensor rastojanja
- jednostavan za upotrebu,
- domet mu se kreće od 2 do 400cm,
- u robotici se koristi za izbjegavanje prepreka,
- koristi se i u raznim projektima automatizacije,
- automobili koriste sličan sensor u sistemima za pomoć pri parkiranju.
- često je potrebno koristiti vodootporni sensor kao JSN-SR04T ili IR senzor koji nije osjetljiv na promjene temperature.



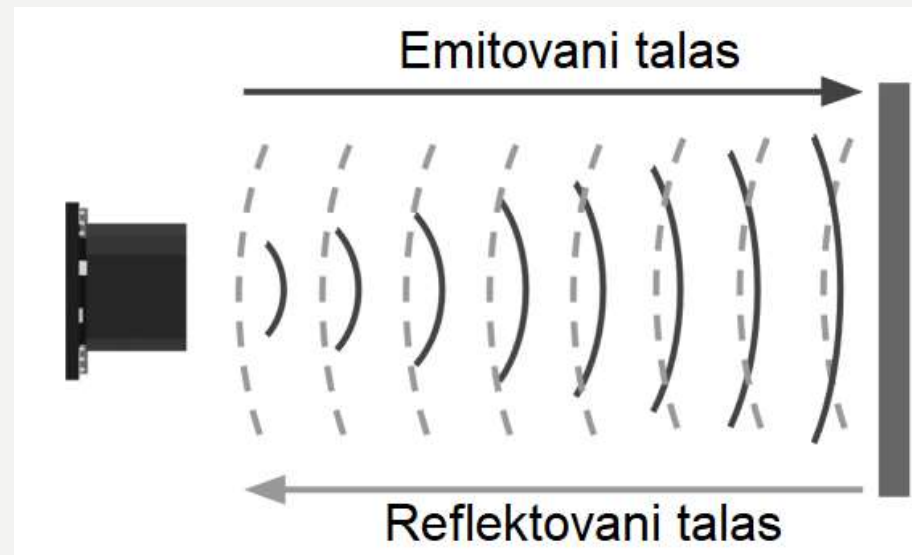
KAKO SENZOR RADI?

Ultrasonični senzor emituje zvučni talas previsoke frekvencije da bi ga ljudsko uho čulo.

Talas putuje kroz vazduh, brzinom, približno 343 m/s.

Ako postoji objekat ispred senzora, zvučni talas će se reflektovati ka prijemniku koji će ga detektovati.

Mjerenjem proteklog vremena od emitovanja do prijema talasa, može se izračunati rastojanje između senzora i objekta.



ODREĐIVANJE RASTOJANJA?

- Na 20°C brzina zvuka je 343 m/s ili 0.034 cm/μs. Neka je proteklo vrijeme 2000μs. Ako se pomnoži brzina prostiranja i vrijeme putovanja talasa, dobija se rastojanje koje je talas prešao.

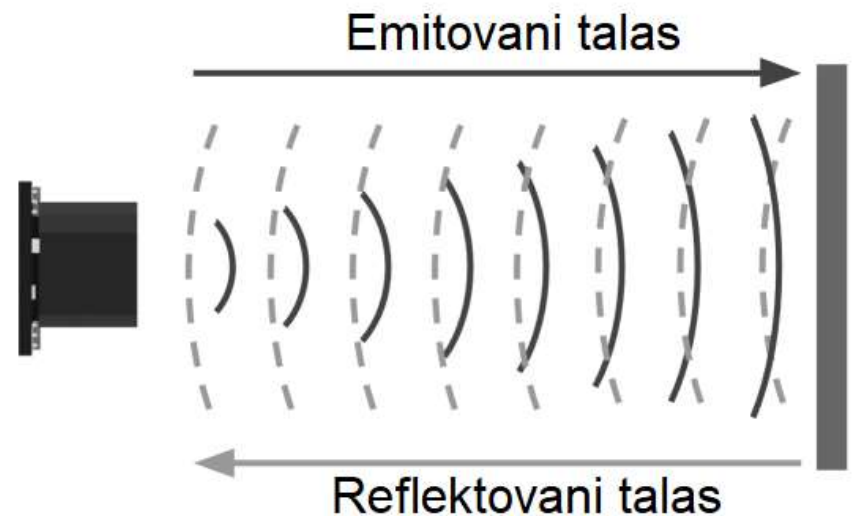
Rastojanje = Brzina x Vrijeme

- Međutim to nije rastojanje senzora i objekta. Ono je samo polovina toga.

Rastojanje (cm) = Brzina zvuka (cm/μs) × Vrijeme (μs) / 2

- Na kraju se ima:

Rastojanje (cm) = 0.0343 (cm/μs) × 2000 (mple this μs) / 2 = 34.3 cm



ZAVISNOST BRZINE PROSTIRANJA ZVUKA OD TEMPERATURE VAZDUHA

- Brzina zvuka je veoma zavisna od temperature i u manjoj mjeri od vlažnosti vazduha. Zavisnost od temperature je približno 0.6 m/s po stepenu Celsius-a. Na 20°C može se uzeti da je brzina 343 m/s, ali ako se želi tačniji rezultat, potrebno je izračunati brzinu zvuka po formuli:

$$V \text{ (m/s)} = 331.3 + (0.606 \times T)$$

V = Brzina zvuka (m/s)

T = Temperatura vazduha (°C)

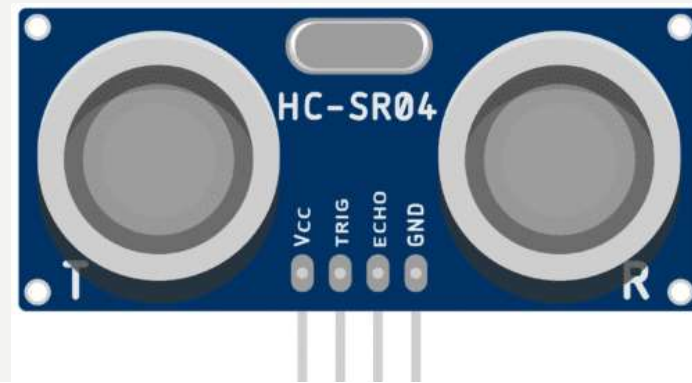
- Nije uzeta u obzir vlažnost, jer je njen uticaj relativno mali.

KAKO SE UPRAVLJA HC-SR04 SENZOROM?

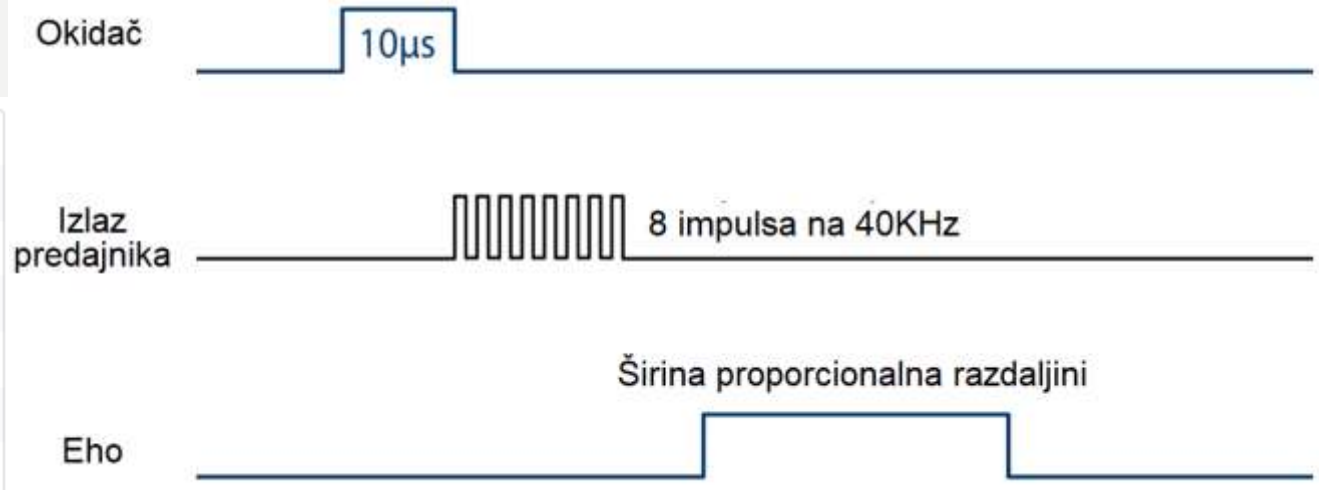
- Dva srebrna cilindra (ultrasonični primopredajnik)
- Jedan emituje zvučni talas a drugi ga prima.
- Za start emitovanja: na Trig pin impuls trajanja najmanje 10 μ s.
- Senzor zatim kreira 8 perioda zvučnog talasa na 40 kHz.
- Talas putuje, odbija se od prepreke i vraća nazad ka prijemu.
- Na Eho pin-u se zatim pojavljuje impuls čije je trajanje proporcionalno rastojanju senzora od objekta.

Može se koristiti `pulseIn()` funkcija u Arduino kodu za očitavanje dužine impulse.

Nakon toga već datom formulom se izračunava rastojanje.

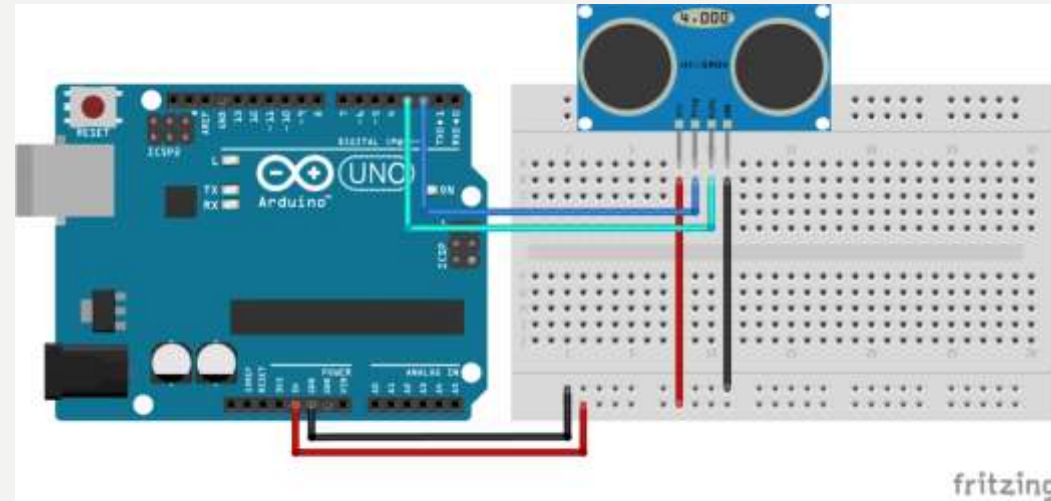


Operating voltage	5 V
Operating current	15 mA
Frequency	40 kHz
Measuring range	2 – 400 cm
Resolution	3 mm
Measuring angle	15 degrees
Trigger input signal	10 μ s high pulse



POVEZIVANJE SENZORA SA ARDUINO UNO

Povezivanje senzora sa Arduino UNO razvojnom pločom je dato na slici:



Na slici su upotrijebljeni pinovi 2 and 3 za Trig i Echo pin, ali se može koristiti i bilo koji drugi digitalin pin.

HC-SR04	Arduino
VCC	5 V
Trig	Pin 2
Echo	Pin 3
GND	GND

PRIMJER – ODREĐIVANJE RASTOJANJA OBJEKTA

```
// Define Trig and Echo pin:
```

```
#define trigPin 2
```

```
#define echoPin 3
```

```
// Define variables:
```

```
long duration;
```

```
int distance;
```

```
void setup() {
```

```
  // Define inputs and outputs:
```

```
  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
```

```
  //Begin Serial communication at a baudrate of 9600:
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  //
```

```
  digitalWrite(trigPin, LOW);
```

```
  delayMicroseconds(5);
```

```
  // Generisanje Trig impulsa
```

```
  digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
```

```
  digitalWrite(trigPin, LOW);
```

```
  // Očitavanje trajanja impulse na echoPin upotrebom pulseIn() u microsec
```

```
  duration = pulseIn(echoPin, HIGH);
```

```
  // Izračunavanje distance
```

```
  distance= duration*0.034/2;
```

```
  // Štampanje distance na serijskom monitoru (Ctrl+Shift+M):
```

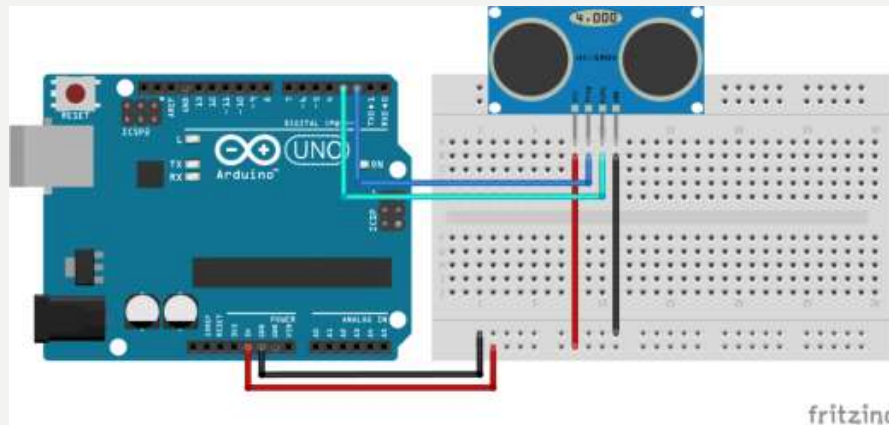
```
  Serial.print("Distance = ");
```

```
  Serial.print(distance);
```

```
  Serial.println(" cm");
```

```
  delay(50);
```

```
}
```



PRIMJER – NEWPING LIBRARY

NewPing (autor Tim Eckel) može se koristiti za mnoge ultrasonic senzore.

Kod je kraći nego u predhodnom primjeru.

Osim toga, NewPing biblioteka ima i drugih korisnih odobina.

Omogućuje postavljanje maksimalnog dometa, te neće čekati čitavu sekundu ako ne primi eho.

Ima ugrađen median filter.

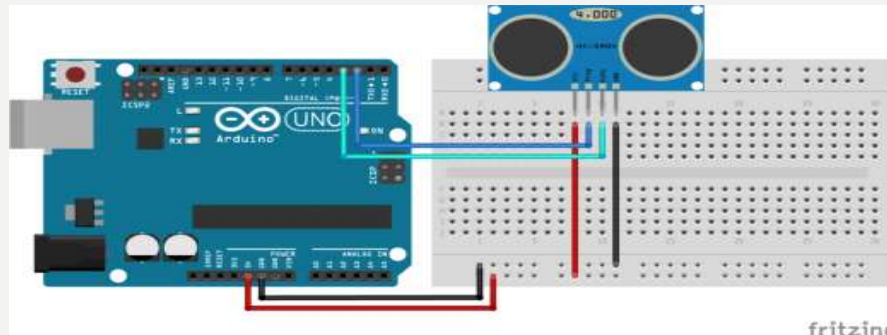
Može se koristiti `distance = sonar.ping_cm()` ili `distance = sonar.ping_in()`

Koji vraćaju mjereno rastojanje u cm ili inčima.

Upotrebom ovih funkcija nije potrebno izračunavanje kao u predhodnom primjeru.

Biblioteku možete preuzeti s linka:

https://github.com/eliteio/Arduino_New_Ping



```
// Uključivanje biblioteke
#include <NewPing.h>
```

```
// Definisane pinova i maksimalnog rastojanja
```

```
#define trigPin 2
```

```
#define echoPin 3
```

```
#define MAX_DISTANCE 350 // Maksimalno rastojanje (u centimetrima). Maksimalan domet --
// senzora je između 400-500cm.
```

```
// NewPing setup
```

```
NewPing sonar(trigPin, echoPin, MAX_DISTANCE);
```

```
void setup() {
```

```
  Serial.begin(9600); // Inicijalizacija serijske komunikacije.
```

```
}
```

```
void loop() {
```

```
  delay(50); // Čekanje 50ms između dva ping-a (oko 20 ping/sec). 29ms je najkraće vrijeme
```

```
  duration = sonar.ping();
```

```
  distance = (duration / 2) * 0.0343;
```

```
  Serial.print("Distance = ");
```

```
  Serial.print(distance); // Rastojanje će biti 0 kada je objekat izvan maksimalnog dometa.
```

```
  Serial.println(" cm");
```

```
}
```

KAKO KORISTITI MEDIAN DIGITALNI FILTER?

Jedna od važnijih novina NewPing biblioteke je realizacija median filtera.

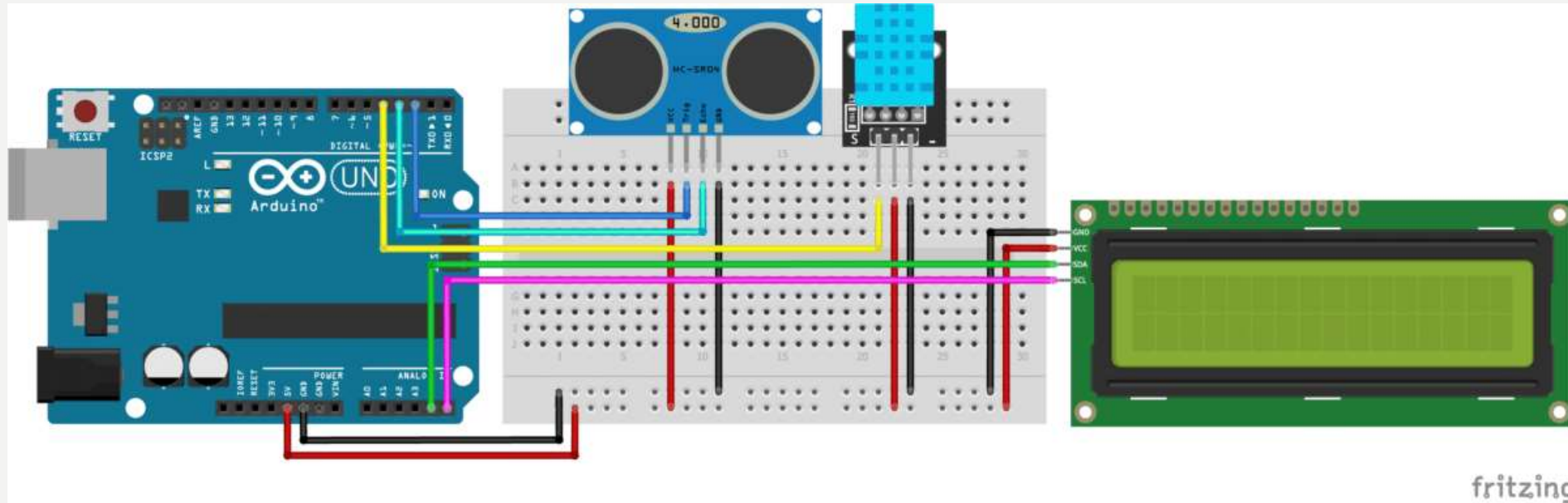
Ovaj filter značajno poboljšava tačnost mjerenja rastojanja.

ping_median() funkcija uzima puno mjerenja trajanja, tako da se kao zaključna uzima srednja vrijednost.

Najčešće se uzima 5 čitanja ali može se zadati i drugi broj.

```
int iterations = 5;  
duration = sonar.ping_median(iterations);
```

PRIMJER - HC-SR04 SA DHT11 I I2C LCD



U ovom primjeru koristi se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit_Sensor** biblioteka, ona se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka **LiquidCrystal_I2C** može se preuzeti sa [ovog linka \(GitHub\)](#). Uključena je i **Wire.h** biblioteka.

PRIMJER - HC-SR04 SA DHT11 I I2C LCD

```
#include <Adafruit_Sensor.h> // https://github.com/adafruit/Adafruit_Sensor
#include <DHT.h> // https://github.com/adafruit/DHT-sensor-library
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
```

```
// Define Trig pin, Echo pin and DHTPin:
#define trigPin 2
#define echoPin 3
#define DHTPin 4
```

```
// Define SDA and SCL pin from LCD:
#define SDAPin A4 // Data pin
#define SCLPin A5 // Clock pin
```

```
// Connect to LCD via i2c, default address 0x27 (A0-A2 not jumpered):
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27,20,4);
```

```
// Define DHT sensor type:
#define DHTType DHT11
```

```
// Define variables:
long duration; int distance; float speedofsound;
```

```
// Create a DHT sensor object:
DHT dht = DHT(DHTPin,DHTType);
```

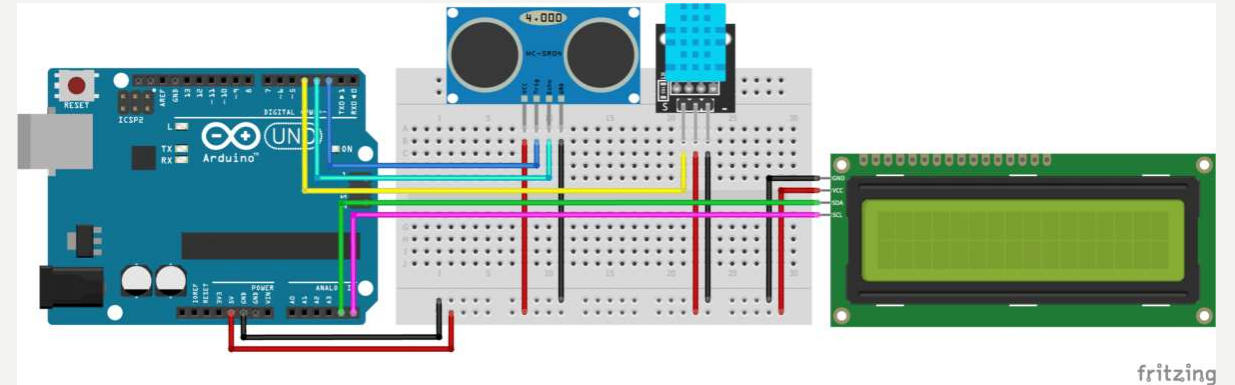
```
void setup() {
  // Define inputs and outputs:
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```
  dht.begin();
```

```
  // Initiate the LCD:
  lcd.init(); lcd.backlight();
```

```
  // Begin Serial communication at a baudrate of 9600:
  Serial.begin(9600);
```

```
}
```



```
void loop() {
  // Clear the trigPin by setting it LOW:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);

  // Trigger the sensor by setting the trigPin high for 10 microseconds:
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echoPin. This returns the duration (length of the pulse) in microseconds:
  duration = pulseIn(echoPin, HIGH);

  // Read the temperature:
  int temperature = dht.readTemperature();

  // Calculate speed of sound in m/s:
  speedofsound = 331.3+(0.606*temperature);

  // Calculate the distance in cm:
  distance = duration*(speedofsound/10000)/2;

  // Print the distance and temperature on the Serial Monitor:
  lcd.setCursor(0,0); lcd.print("T: "); lcd.print(temperature); lcd.print("\xDF" "C ");
  lcd.print("D: "); lcd.print(distance); lcd.print("cm ");
  lcd.setCursor(0,1); lcd.print("Spd: "); lcd.print(speedofsound); lcd.print("m/s ");
  delay(100);
}
```

PRIMJER ZA VJEŽBU

Štoperica.

Mjeriti vrijeme od trenutka kada se objekat udalji od ultrasoničnog senzora na rastojanje veće ili jednako 10cm, do trenutka kada se ponovo vrati na rastojanje manje od 10cm.

U gornjem redu LCD-a, s lijeva ispisivati "DISTANCE: ", a s desna ispisivati izmjereno rastojanje u cm, npr.: "24cm".

U desnom uglu donjeg reda LCD-a, ispisivati proteklo vrijeme od kada se objekat udaljio (VUO), na rastojanje veće od 10cm. Vrijeme ispisivati u obliku: mm:ss.d

Nakon povratka objekta na rastojanje manje od 10cm, vrijeme proteklo od udaljavanja do povratka (IV), ispisati u lijevom uglu donjeg reda, u obliku: mm:ss.d, dok u desnom uglu donjeg reda ispisati: 00:00.0 i resetovati VUO (vratiti na 0).

IV ispis zadržati dok je objekat prisutan na ostojanju manjem od 10cm, kao i nakon što se objekat udalji, sve do njegovog ponovnog povratka na rastojanje manje od 10cm, kada treba ispisati novo IV.

VUO zadržati na 0 dok je objekat blizu senzora, na rastojanju manjem od 10cm. Čim se objekat udalji VUO poprima vrijednost proteklog vremena od udaljavanja i osvježava se ispis u desnom uglu donjeg reda LCD-a.



(4-2-1 bod)

PRIMJER ZA VJEŽBU

Naplatni punkt na autoputu.

Ultrasonični senzor detektuje nailazak vozila i omogućuje uključenje rasvjete ispred naplatnog šaltera. Rasvjetu predstaviti sa dvije istobojne LED. Rasvjetu uključivati samo ako je mračno, odnosno ako se pomoću fotootpornika detektuje nizak nivo svjetlosti.

Ultrasonični sensor je postavljen iznad vozila na naplatnom punktu i mjeri minimalno rastojanje od vozila koje prolazi. Na osnovu tog podatka vrši klasifikaciju vozila na putničke automobile i teretna vozila (autobuse, kamione, kombije, ...).

Rasvjetu držati uključenu sve dok vozilo ne prođe ispod ultrazvučnog senzora ili ne protekne maksimalno vrijeme predviđeno za prolazak (ne koristiti delay!!). Maximalno vrijeme zadavati putem seriskog monitora (predefinisati 15sek).

Informaciju o broju putničkih i teretnih vozila koja su prošla od početka nadzora, ispisati na LCD-u i serijskom monitoru.



(4-2-1 bod)

VJEŽBE SA AUTOMOBILOM

1. Objekat ispred automobila

Kada se ispred automobila postavi objekat, na rastojanju između 20-40cm, automobil ga pomoću robotske ruke prihvati i odnese ga na 10 cm odstojanja od zida.

(3-1)

2. Parkiranje – objekat kao orjentir

Kada se ispred automobila postavi objekat, on tu informaciju iskoristi da se parkira u garažu.

(4-2)

Više informacija o automobilu na:

https://www.aliexpress.com/item/1005004799385277.html?spm=a2g0o.order_list.order_list_main.218.47d61802QypXXk

(Poeni ovih vježbi se sabiraju)