

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 10	Mjerenje udaljenosti UZV senzorom
Cilj vježbe:	Svladati osnovne principe i način mjerenja udaljenosti Arduinoom

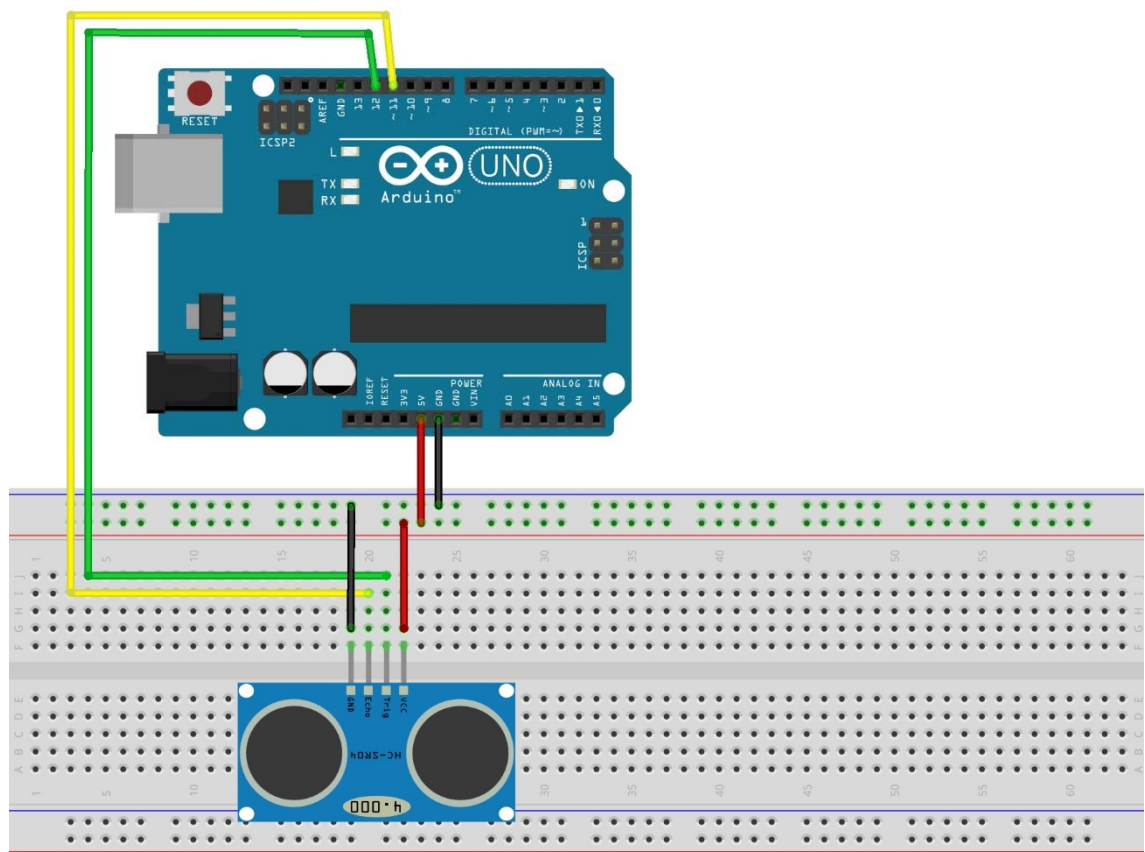
Upute

Sve zadatke spremi na USB, a u bilježnici za sve zadatke napiši:

- postupak izrade programa
- objašnjenje korištenih naredbi
- dobivene rezultate po točkama
- odgovoriti u bilježnicu na postavljena pitanja vezana uz ovu vježbu
- Ukoliko u kodu postoji greška, korigiraj i objasni!

Zadatak 1. Spoji ultrazvučni senzor prema slici te napiši program koji će na Serial monitor ispisivati vrijednosti udaljenosti predmeta od senzora! Promatraj da li je izračunata udaljenost jednaka i koliki je mjerni opseg ako se ultrazvuk odbija od tvrde prepreke ili mekane, na primjer odjeće!

Grafička shema:



fritzing

Kòd zadatka

```
const int trigPin = 12;    // deklaracija Trig pina
const int echoPin = 11;    // deklaracija Echo pina

long duration;             // deklaracija varijable za trajanje
long distance;             // impulsa i varijable za udaljenost

void setup() {
  pinMode(trigPin, OUTPUT); // definira trigPin kao izlaz
  pinMode(echoPin, INPUT);  // definira echoPin kao ulaz
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW); // trigPin postavljamo u
  delayMicroseconds(2);       // logičku 0

  digitalWrite(trigPin, HIGH); // postavljamo trigPin
  delayMicroseconds(10);       // u logičku 1 u trajanju
  digitalWrite(trigPin, LOW);  // 10 µs

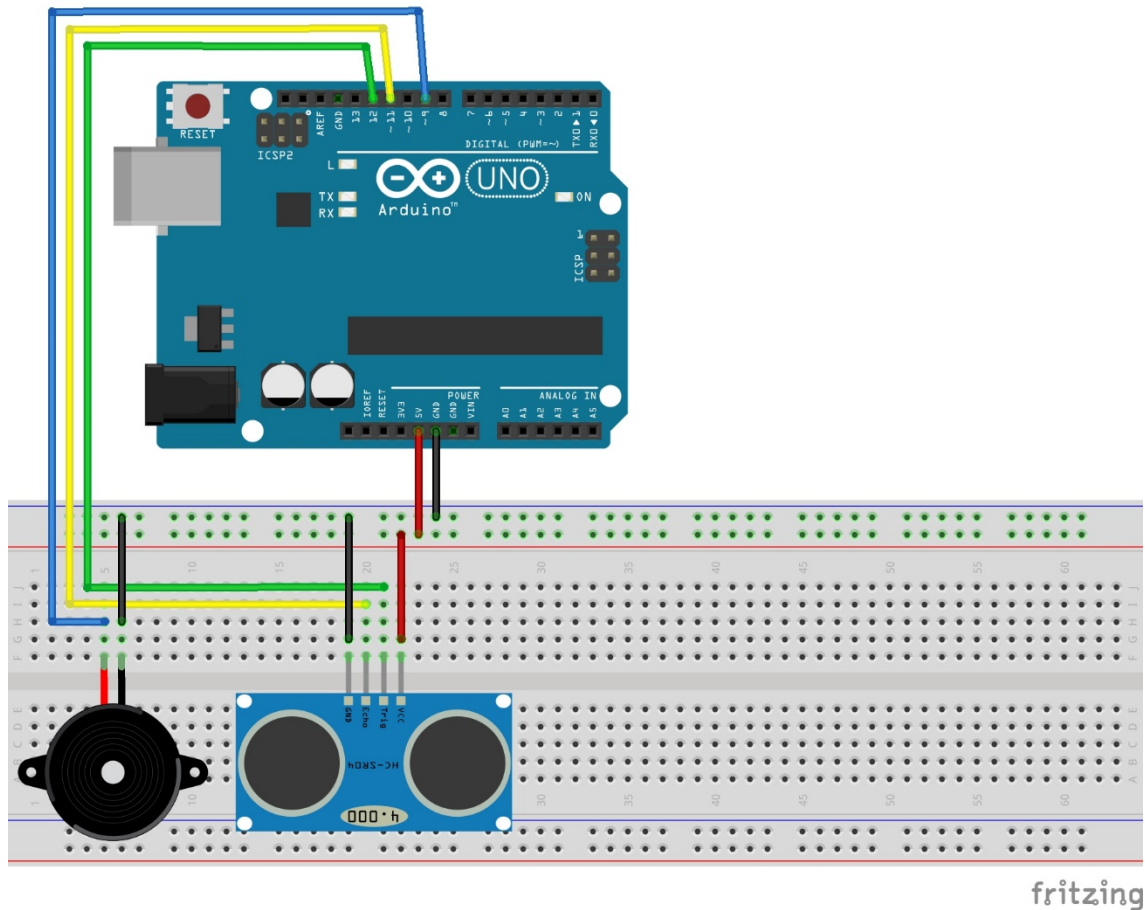
  duration = pulseIn(echoPin, HIGH); // funkcija za određivanje trajanja
                                     // logičke 1 na echoPin što odgovara
                                     // trajanju putovanja UZV u µs

  distance= duration*0.034/2;        // računanje udaljenosti

  if(distance>332 || distance<2)     // vrijednost udaljenosti mora biti
                                     // između 2.4 cm i 330 cm
  {
    Serial.println("Udaljenost je izvan dozvoljenih granica!");
  }
  else
  {
    Serial.print("Udaljenost je: "); // ispis udaljenosti
    Serial.println(distance);        // na Serial monitor
    Serial.println(" cm.");
  }
  delay(50);
}
```

Zadatak 2. Proširi prethodni zadatak tako da spojiš piezo zujalicu koja će davati kontinuirani zvučni signal kada je udaljenost manja od 10 cm. Odaberi frekvenciju tona zujalice u skladu s brojem tvog radnog mjesta prema formuli $f = (200 + 50 \cdot \text{BrojRadnogMjesta})$ u labosu. Pozitivni pin zujalice spoji na pin 9 Arduina.

Grafička shema:



Kòd zadatka

```
const int trigPin = 12;    // deklaracija Trig pina
const int echoPin = 11;    // deklaracija Echo pina
const int piezoPin = 9;    // deklaracija Piezo pina

long duration;             // deklaracija varijable za trajanje
long distance;             // impulsa i varijable za udaljenost

void setup() {
  pinMode(trigPin, OUTPUT); // definira trigPin kao izlaz
  pinMode(echoPin, INPUT);  // definira echoPin kao ulaz

  pinMode(piezoPin, OUTPUT);

  beep_init();              // inicijalni zvučni signal za početak rada
  delay(1000);

  Serial.begin(9600);
}
```

```
void loop() {
digitalWrite(trigPin, LOW); // trigPin postavljamo u
delayMicroseconds(2);      // logičku 0

digitalWrite(trigPin, HIGH); // postavljamo trigPin
delayMicroseconds(10);      // u logičku 1 u trajanju
digitalWrite(trigPin, LOW); // 10 µs

duration = pulseIn(echoPin, HIGH); // funkcija za određivanje trajanja
                                     // logičke 1 na echoPin što odgovara
                                     // trajanju putovanja UZV u µs

distance= duration*0.034/2; // računanje udaljenosti

if(distance>332 || distance<2) // vrijednost udaljenosti mora biti
                               // između 2,4 cm i 330 cm
{
    Serial.println("Udaljenost izvan dozvoljenih granica!");
}
else
{
    Serial.print(„Udaljenost je: "); // ispis udaljenosti
    Serial.println(distance);         // na Serial monitor
    Serial.println(„ cm.");
}
delay(50);
}

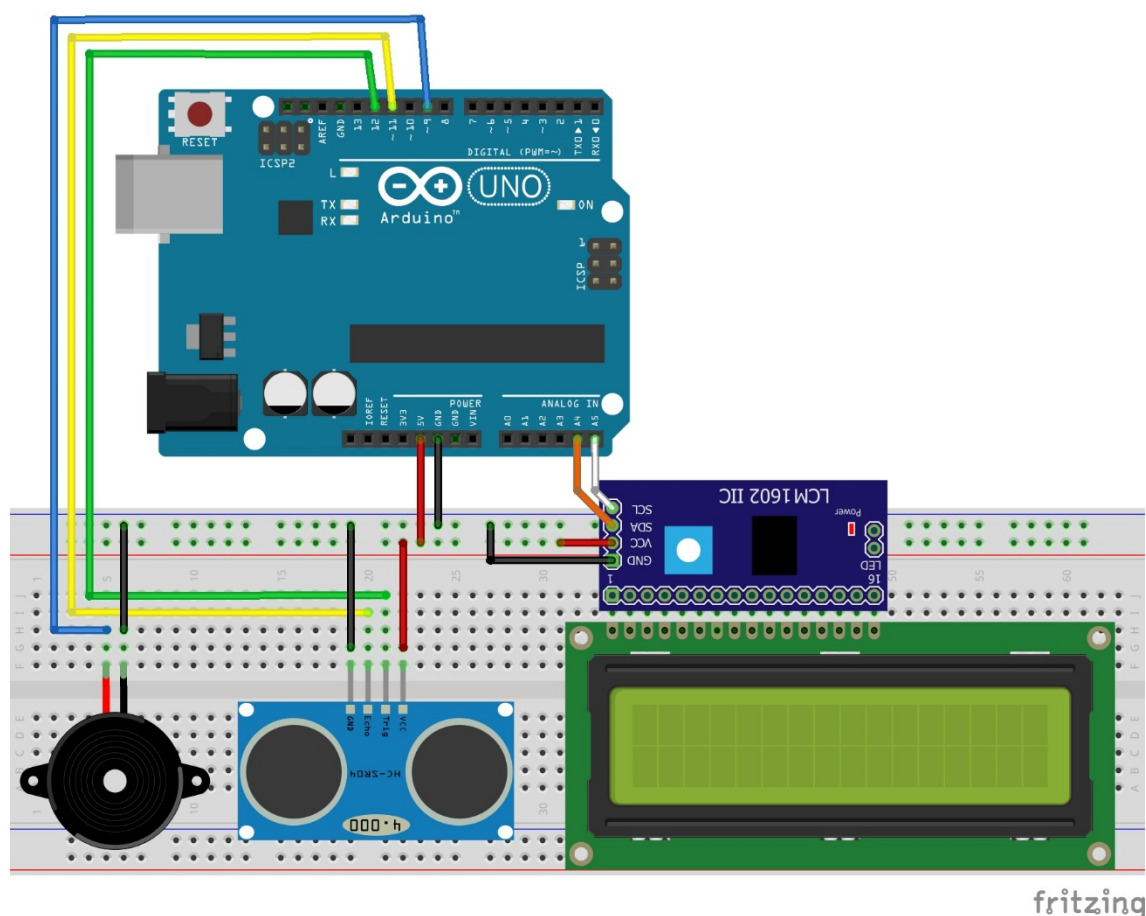
if(distance<10)
{
    beep(distance);
}
else
{
    noTone(piezoPin);
}
delay(50);
}

void beep(unsigned char distance){
    tone(piezoPin, frequency);
}

void beep_init(){
tone(piezoPin, 500, 300); // cjelokupna sintaksa funkcije tone:
delay(300);              // tone(pin, frequency, duration)
tone(piezoPin, 500, 300);
delay (300);
tone(piezoPin, 500, 300);
delay (300);
}
```

Zadatak 3. Proširi prethodni zadatak tako da spojiš LCD zaslon i udaljenost u cm, uz Serial monitor ispisuješ i na LCD zaslon.

Grafička shema:



Kòd zadatka

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int trigPin = 12;    // deklaracija Trig pina
const int echoPin = 11;    // deklaracija Echo pina
const int piezoPin = 9;    // deklaracija Piezo pina

long duration;             // deklaracija varijable za trajanje
long distance;             // impulsa i varijable za udaljenost

void setup() {
  pinMode(trigPin, OUTPUT); // definira trigPin kao izlaz
  pinMode(echoPin, INPUT);  // definira echoPin kao ulaz

  pinMode(piezoPin, OUTPUT);

  beep_init();              // zvučni signal za početak rada
  delay(1000);
}
```

```
lcd.begin();           // inicijaliziranje LCD-a
lcd.backlight();       // uključivanje pozadinskog osvjetljenja LCD-a
lcd.setCursor(0,0);    // postavljanje kursora na lokaciju 0,0 (stupac,redak)

Serial.begin(9600);
}

void loop() {
digitalWrite(trigPin, LOW); // trigPin postavljamo u
delayMicroseconds(2);      // logičku 0

digitalWrite(trigPin, HIGH); // postavljamo trigPin
delayMicroseconds(10);      // u logičku 1 u trajanju
digitalWrite(trigPin, LOW);  // 10 µs

duration = pulseIn(echoPin, HIGH); // funkcija za određivanje trajanja
                                     // logičke 1 na echoPinn što odgovara
                                     // trajanju putovanja UZV u µs

distance= duration*0.034/2;        // računanje udaljenosti

if(distance>332 || distance<2)     // vrijednost udaljenosti mora biti
                                     // između 2.4 cm i 330 cm
{
    Serial.println("Udaljenost van dozvoljenih granica!");

    lcd.setCursor(0,0);
    lcd.print("Udaljenost van dozvoljenih granica!");
}
else
{
    Serial.print(",Udaljenost je: "); // ispis udaljenosti
    Serial.println(distance);        // na Serial monitor
    Serial.println(", cm.");

    lcd.setCursor(0,0);
    lcd.print("Udaljenost je: ");    // ispis udaljenosti
    lcd.print(distance);             // na LCD display
    lcd.print(" cm.");
}
delay(50);
}

if(distance<10)
{
    beep(distance);
}
else
```



```

    {
        noTone();
    }
    delay(50);
}

void beep(unsigned char distance){
    tone(piezoPin, 500);
}

void beep_init(){
    tone(piezoPin, 500, 300); // cjelokupna sintaksa funkcije tone:
    delay(300);               // tone(pin, frequency, duration)
    tone(piezoPin, 500, 300);
    delay(300);
    tone(piezoPin, 500, 300);
    delay(300);
}

```

Zadatak 4. Modificiraj kôd iz prethodnog zadatka tako da spoju implementiraš funkcionalnost parkirnog senzora na modernim vozilima. Frekvenciju tona piezo zujalice podesi u skladu sa zadatkom 2, a tek pri završetku projekta postavi na 500 Hz.

- Na udaljenosti $d > 0,5$ m nema zvuka.
- Na udaljenosti $0,1 \text{ m} < d < 0,5$ m isprekidani zvuk, što je objekt bliže to je **frekvencija ponavljanja** sve veća.
- Na udaljenosti $d < 0,1$ m kontinuirani zvučni signal
- Na LCD zaslonu brojčano i vizualno prikaži udaljenost na način da kreiraš potrebne znakove za prikaz (pomoću 'bar graf skale', 'pomoću rastuće skale kao kod podešavanja glasnoće', sličica automobila, prepreke...) DATI VLASTITU IDEJU!



Kodove za kreiranje znakova i simbola moguće je provjeriti na <http://maxpromer.github.io/LCD-Character-Creator/> ili sličnoj.

Ideju za simbole i znakove pronaći na Google: „lcd custom characters“ ili sl.