

# Arduino i SRF04 ultrazvučni senzor

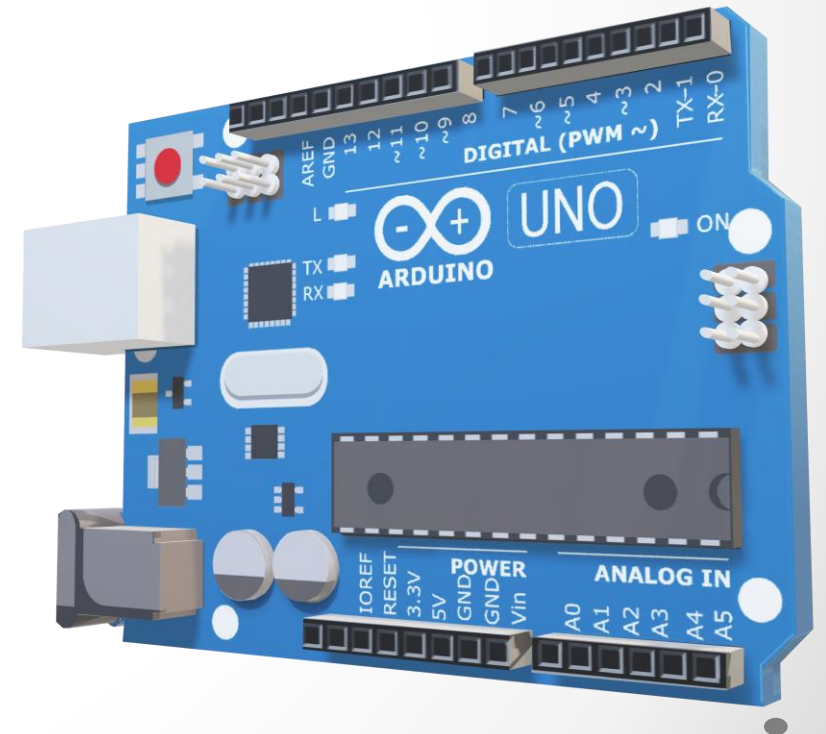
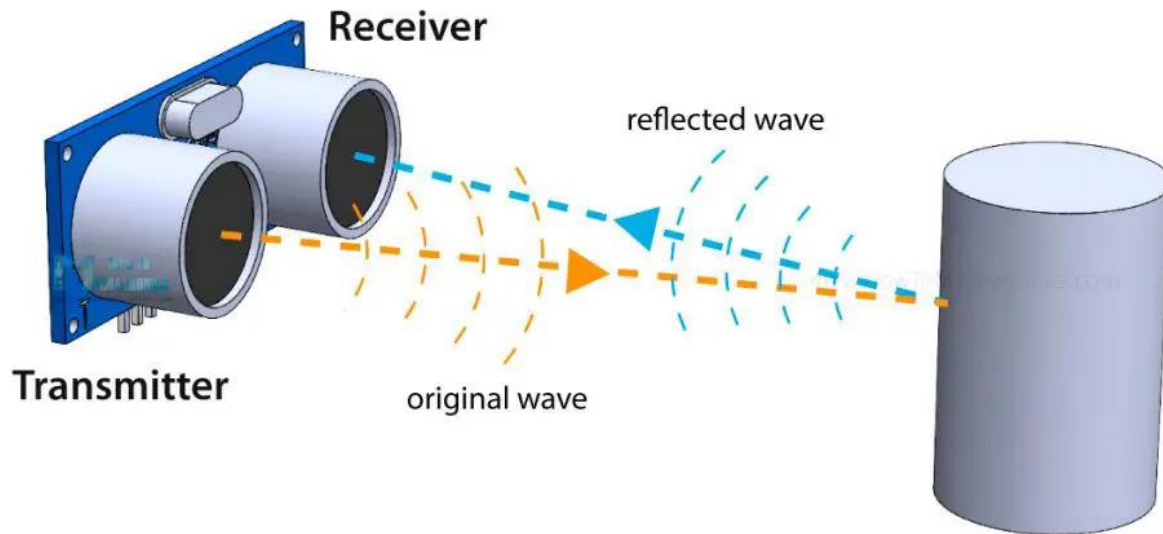
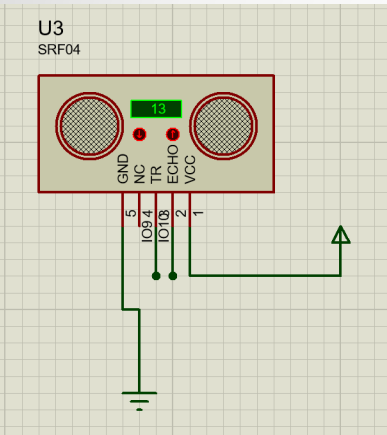
## Internet stvari 2023. - V termin (1 čas)



Nenad Petrović

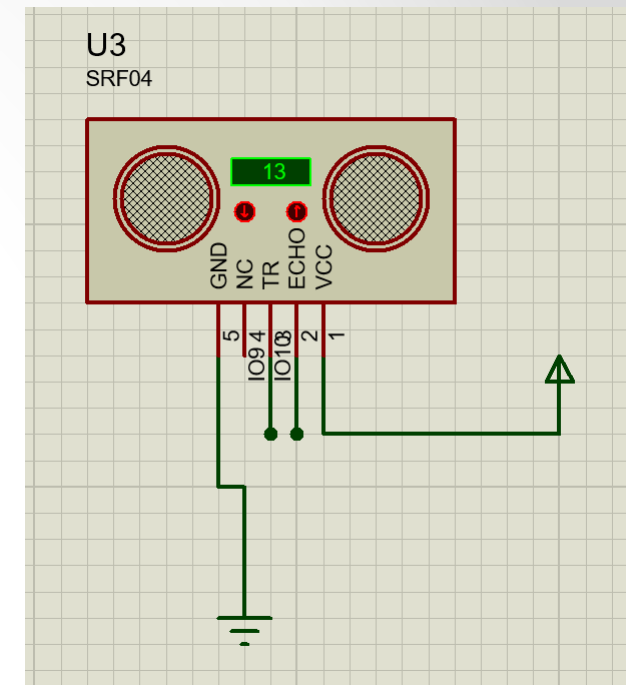
Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



# Uvod o SRF04 ultrazvučnom senzoru

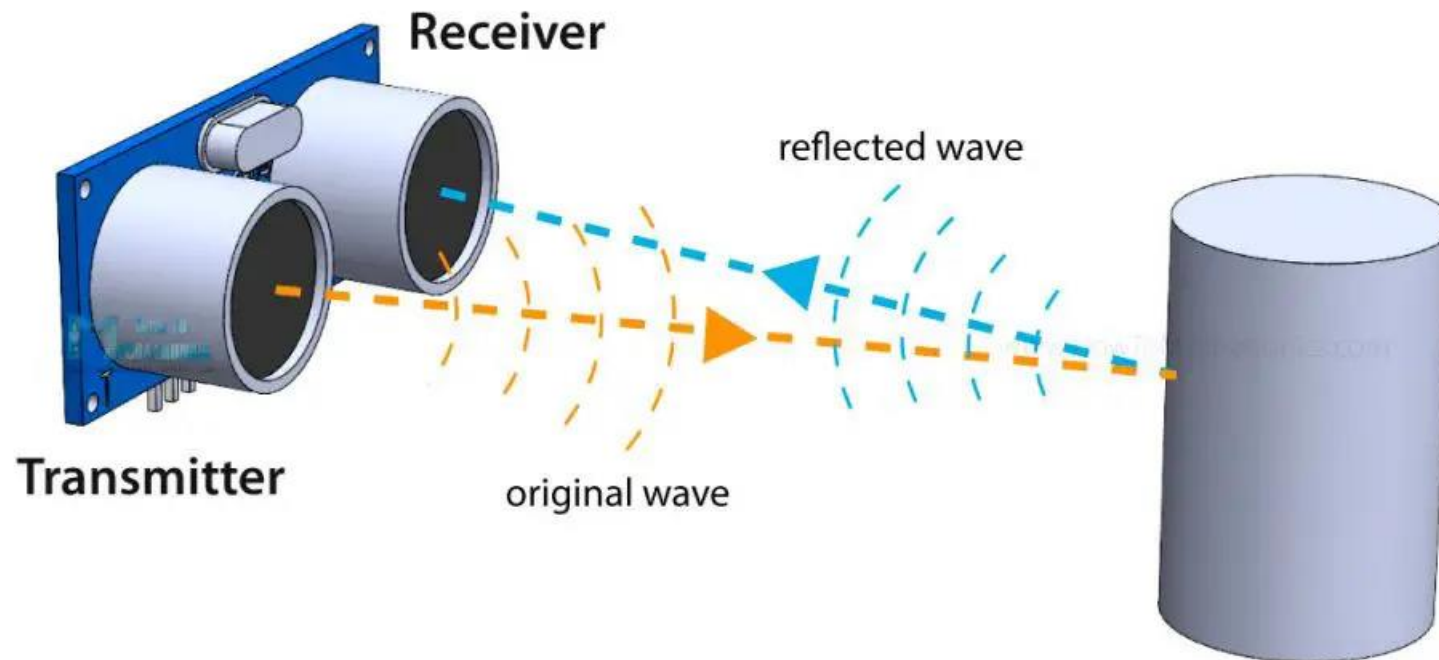
- Priuštiv i lak za upotrebu senzor čija se svrha merenje rastojanja
- Opseg od 2 do 400 cm
- Sastoji se od dva ultrazvučna transducer-a.
  - Prvi je transmitter koji šalje ultrazvučne impulse
  - Drugi je receiver koji osluškuje prisustvo reflektovanih talasa
- U suštini, praktično je sličan SONAR-u koji podmornice koriste za detekciju objekata.
- Posедуje 4 pina
- VCC
  - Ide na 5V pin Arduina
- GND
  - GND pin na Arduinu
- Trig
  - Povezuje se na bilo koji digitalni pin
  - Aktivacijom šalje ultrazvučni talas preko transmittera
- Echo
  - Osluškuje reflektovani signal



Radni napon	5V DC
Radna struje	15mA
Radna frekvencija	40KHz
Minimalno rastojanje	2cm
Maksimalno rastojanje	400cm
Preciznost	3mm
Dimenzije	45 x 20 x 15mm

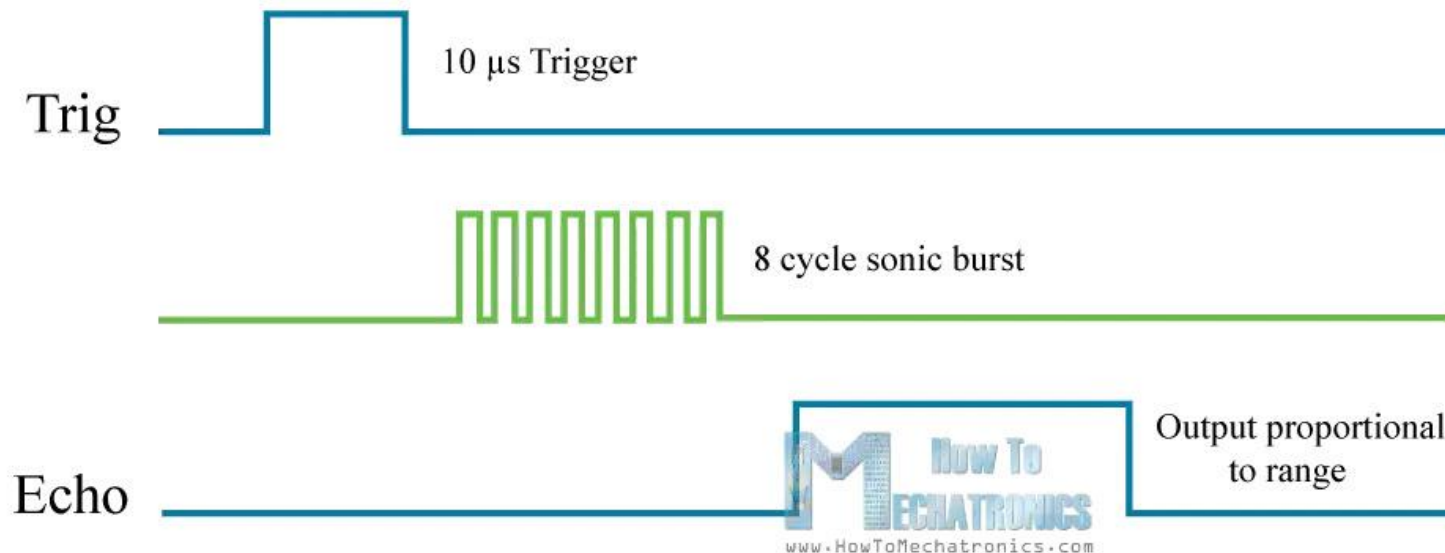
# Princip rada SRF04 ultrazvučnog senzora

- Transmitter emituje ultrazvuk na frekvenciji od 40 000 Hz
- Talas putuje kroz vazduh i ako naiđe na objekat ili prepreku na svojoj putanji, vraća se nazad ka receiver-u
- Razmatrajući vreme putovanja talasa i brzinu zvuka, moguće je jednostavno izračunati rastojanje



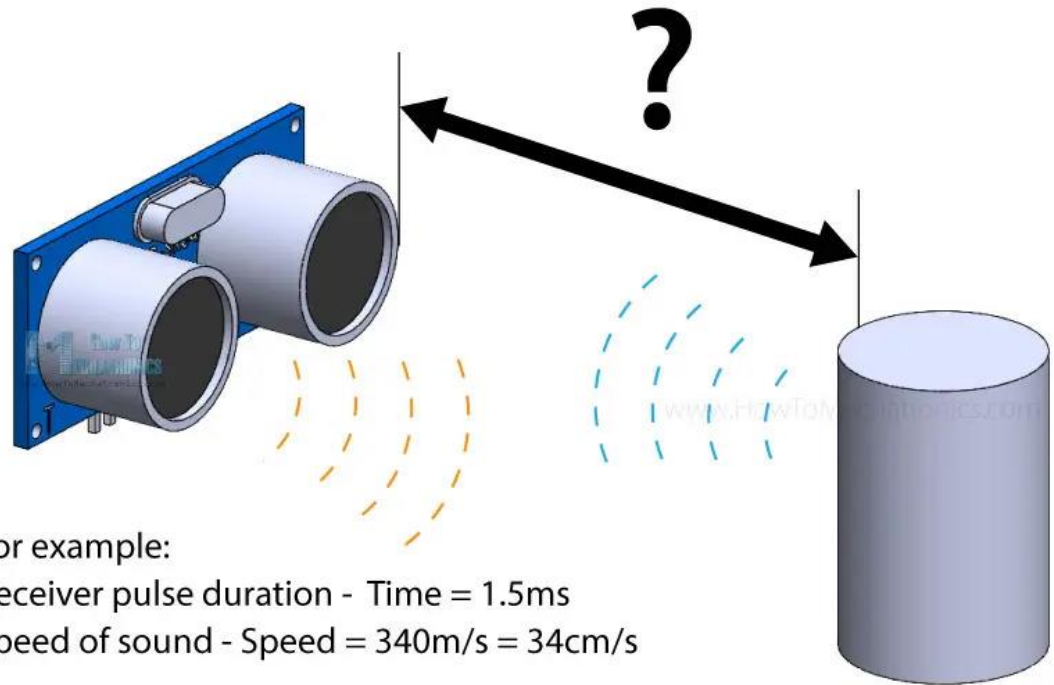
# Princip rada SRF04 ultrazvučnog senzora

- Da bismo generisali ultrazvučni talas, potrebno je da postavimo TRIGGER pin na stanje visokog napona u trajanju od 10  $\mu$ s.
  - Kao rezultat, šalje se ultrazvučni talas od 8 ciklusa koji se kreće brzinom zvuka kroz vazduh
- ECHO pin se stavlja u HIGH stanje nakon 8 ciklusa, pa nakon toga počinje da čeka reflektovani talas
  - [Ako nema objekta] ECHO pin će se vratiti u LOW stanje nakon time out perioda od 38ms
  - [Primljen reflektovani talas] ECHO pin će preći u LOW stanje brže od 38ms. Rastojanje koje je zvuk prešao računamo na osnovu vremena tokom kog je bio ECHO pin u stanju HIGH u tom slučaju



# Računanje rastojanja

- Vreme
  - Trajanje kretanje talasa do objekta i nazad
- Brzina
  - Brzina zvuka  $340\text{m/s} = 34\text{cm/ms}$
- Rastojanje =  $(\text{Brzina} \times \text{Vreme}) / 2$ 
  - Zato što obuhvata put do objekta i nazad
- Primer
  - Izmereno  $1.5\text{ms}$
  - Znači da je ECHO pin imao visok naonski nivo HIGH u trajanju od  $1.5\text{ms}$ , što merimo pulseIn() funkcijom
  - Rastojanje računamo
    - $(34\text{cm/ms} \times 1.5\text{ms}) / 2 = 51\text{cm} / 2 = 25.5\text{cm}$
    - $(0.034\text{cm/us} \times 1500\text{ us}) / 2 = 51\text{cm} / 2 = 25.5\text{cm}$



For example:

Receiver pulse duration - Time =  $1.5\text{ms}$

Speed of sound - Speed =  $340\text{m/s} = 34\text{cm/s}$

$$\text{Distance} = (\text{Speed} \times \text{Time}) / 2$$

$$\text{Distance} = (34\text{cm/ms} \times 1.5\text{ms}) / 2 = 25.5\text{cm}$$



# Pomoćna funkcija za merenje trajanja impulsa

- Poziv funkcije
  - `pulseIn(pin, value)`
  - `pulseIn(pin, value, timeout)`
- Parametri
  - **Pin** (int): broj Arduino pina na kom čitamo impuls
  - **Value** (int): tip pročitanoog impulsa – HIGH ili LOW
  - **Timeout** [opciono] (unsigned long): broj mikrosekundi koji se čeka da impuls počne, podrazumevano je jedna sekunda
- Povratna vrednost – unsigned long
  - Dužina impulsa u **mikrosekndama** ili 0 ako je istekao time out
- Primer merenja dužine trajanja HIGH
  - Pin u ulaznom režimu

```
int pin = 7;
unsigned long duration;

void setup() {
    pinMode(pin, INPUT);
}

void loop() {
    duration = pulseIn(pin, HIGH);
}
```

# Arduino COVID-19 zaštita

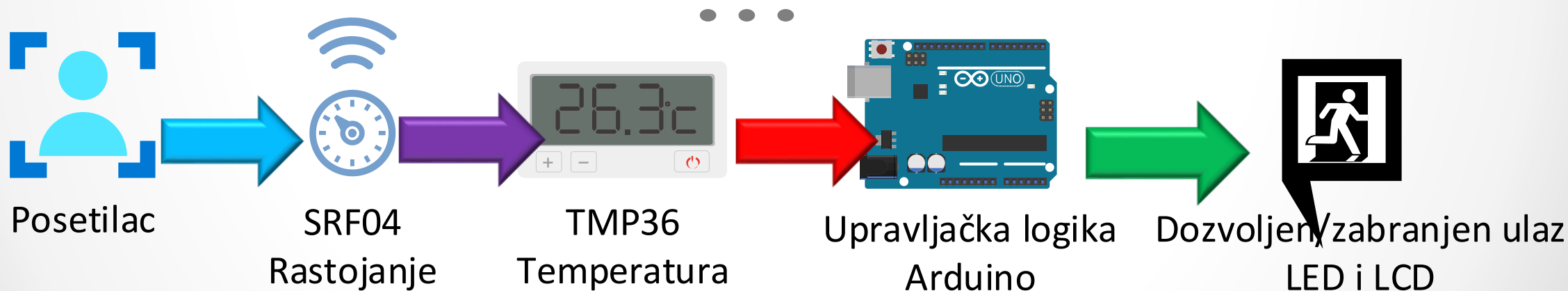
## SRF04 ultrazvučni i TMP36 temperaturni senzori

### *Internet stvari 2023. - V termin*

Nenad Petrović

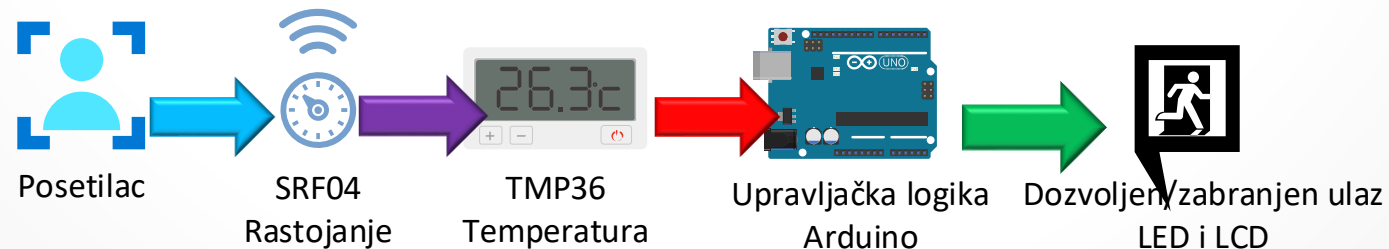
Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



# Primer 5.1

- Realizovati sistem zaštite od COVID-19 na ulazu zatvorenih prostorija korišćenjem Arduino Mega 2560 i pomoćnih uređaja:
  - **SRF-04 ultrazvučni senzor za merenje rastojanja**
  - TMP-36 analogni temperaturni senzor
  - LCD displej LM-016L
  - crvena i zelena LED dioda
- Senzor za merenje rastojanja svake polovine sekunde proverava da li je osoba došla do ulaza - ukoliko je izmereno rastojanje manje od 5 cm.
- Ukoliko je ovaj uslov ispunjen, pribavlja se trenutna vrednost temperature sa analognog senzora.
- Ukoliko je izmerena vrednost temperature manja od 37 stepeni celzijusovih, na LCD displeju se prikazuje poruka "DOBRODOSLI!", a svetli zelena LED dioda. U suprotnom, svetli crvena LED dioda i štampa poruka "VISOKA TEMPERATURA".







# Merenje vremena

- Preskaler podešen tako da generiše TMR0 prekid svake milisekunde
- Cilj nam je da izmerimo 0.5s
- $0.5s = 500ms$ , pa zato brojimo do 500
- Dve promenljive
  - Counter
    - Povećava se svakim ulaskom u tajmer0 prekid
    - Svake milisekunde se inkrementira
  - Half\_\_seconds
    - Povećava se za 1 svaki put kada prođe 500 ulazaka u prekid tajmera0
    - Označava koliko polovina sekundi je prošlo

```
...
volatile int counter = 0;
volatile int half_seconds = 0;
...
void setup() {
    ...
    cli(); //Onemoguciti prekide
    ...
    //Preskaler 1/64 takta : CS02 = 0, CS01 = 1, CS00 = 1
    TCCR0B|=(1<<CS01);
    TCCR0B|=(1<<CS00);

    TCCR0A=(1<<WGM01);    //Podesiti rezim poklapanja sa OCR0A

    OCR0A=0xF9; //Za 1ms: (16*10^6/64)-1 = 1000/4 = 250 - 1 = 249 (F9 hex)

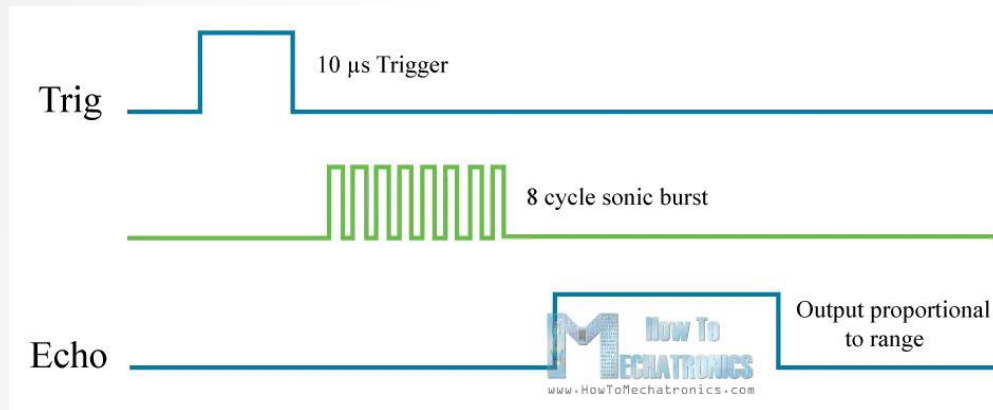
    TIMSK0|=(1<<OCIE0A); //Omoguciti izvor prekida - poklapanje sa OCR0A

    sei(); //Globalno omoguciti prekide
}
```

```
//Rutina koja se izvrsava kada Timer 0 ima vrednost OCR0A
ISR(TIMER0_COMPA_vect){
    counter++;
    if(counter==500){
        half_seconds++;
        counter=0;
    }
}
```

# Merenje rastojanja i temperature

- $\text{Rastojanje} = (\text{Brzina} \times \text{Vreme}) / 2$



```
float measureDistance(int trigger, int echo){
    long duration = 0;
    int distanceCm = 0;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distanceCm = duration * 0.034 / 2;
    return distanceCm;
}
```

```
float measureTemp(int sensorPin){
    int sensorValue = 0;
    float temp = 0;
    lcd.clear();
    sensorValue = analogRead(sensorPin); //procitati vrednost napona u opsegu do 1024
    temp = (double)sensorValue / 1024; //prevesti na procentualnu vrednost
    temp = temp * 5; //mozenje sa 5V da dobijemo napon
    temp = temp - 0.5; //Oduzeti offset
    temp = temp * 100; //Prebaci u stepene celzijusa
    lcd.print(temp);
    return temp;
}
```

# Glavna petlja

- Svake polovine sekunde
  - Izmeriti rastojanje
  - Ako je izmereno rastojanje manje od 5
    - Jeste
      - Izmeriti temperaturu
        - Temperatura manja od 37
          - Zeleni LED
          - LCD poruka DOBRODOSLI!
        - Temperatura veća/jednaka od 37
          - Crveni LED
          - LCD poruka VISOKA  
TEMPERATURA
    - Nije
      - LCD poruka CEKAM...

```
void loop() {  
    if(half_seconds == 1){  
        measuredDistance = measureDistance(trigPin, echoPin);  
        if(measuredDistance<5){  
            measuredTemperature = measureTemp(tempPin);  
            if(measuredTemperature<37){  
                digitalWrite(greenPin, HIGH);  
                digitalWrite(redPin, LOW);  
                lcd.clear();  
                lcd.print("DOBRODOSLI!");  
            }  
            else{  
                digitalWrite(redPin, HIGH);  
                digitalWrite(greenPin, LOW);  
                lcd.clear();  
                lcd.setCursor(0,0);  
                lcd.print("VISOKA");  
                lcd.setCursor(0,1);  
                lcd.print("TEMPERATURA");  
            }  
        }  
        else{  
            lcd.clear();  
            lcd.setCursor(0,0);  
            lcd.print("CEKAM...");  
        }  
        half_seconds = 0;  
    }  
}
```

# Arduino i servo

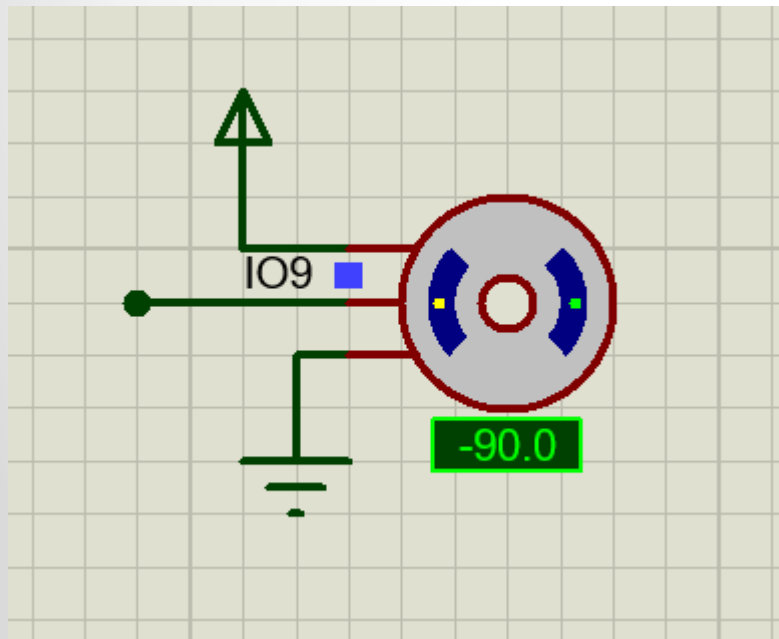
*Internet stvari 2023. - V termin (2 čas)*



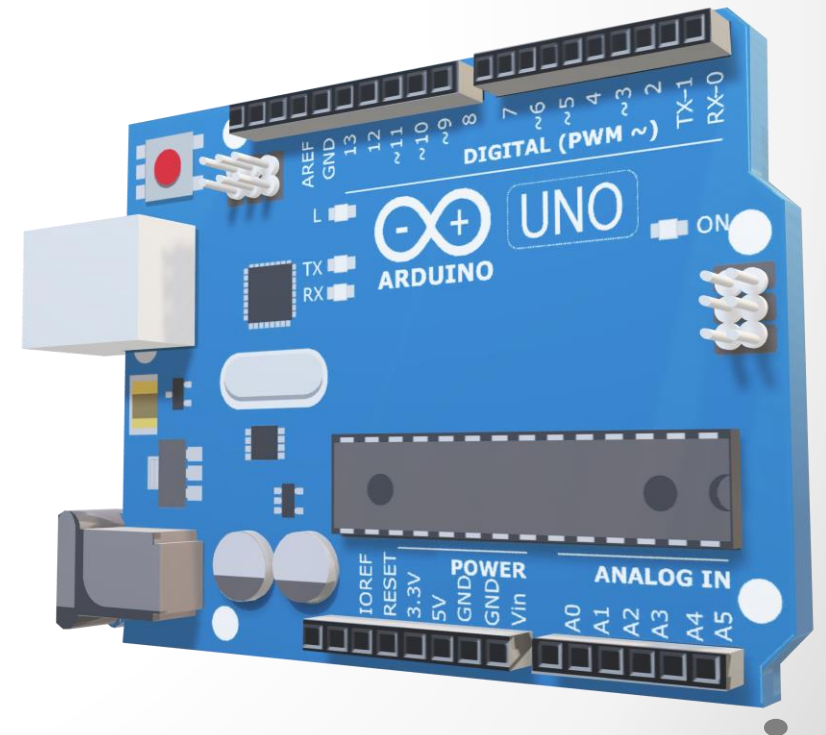
Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



...



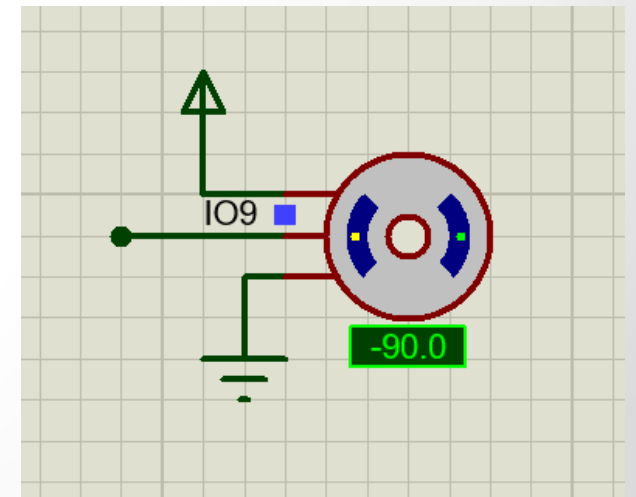


# Uvod o upravljanju servo motorima

- Biblioteka Servo
  - `#include <Servo.h>`
- Daje mogućnost Arduino pločama da upravljaju raznovrsnim servo motorima
- Omogućava rad sa više servo motora
  - Obratiti pažnju na tajmere
  - 12 servo jedinica se mogu upravljati samo jednim tajmerom
  - Arduino Due – do 60 servo jedinica
  - Arduino Mega – do 48
- Namenjena hobi RC servo motorima (igračke, prototipi)
- Servo se sastoji od zupčanika i osovine koji se mogu precizno kontrolisati
  - Osovnina se može okretati pod različitim uglovima od 0 do 180
  - Kontinualna rotacija serva utiče na brzinu
- Na svim pločama koje nisu Mega, upotreba biblioteke onemogućava PWM analogni upis na pinovima 9 i 10, bez obzira da li je tu servo ili ne
- Kod Mega, do 12 serva bez interference sa PWM funkcionalnostima
  - 12-23 serva će onemogućiti PWM na pinovima 11 i 12

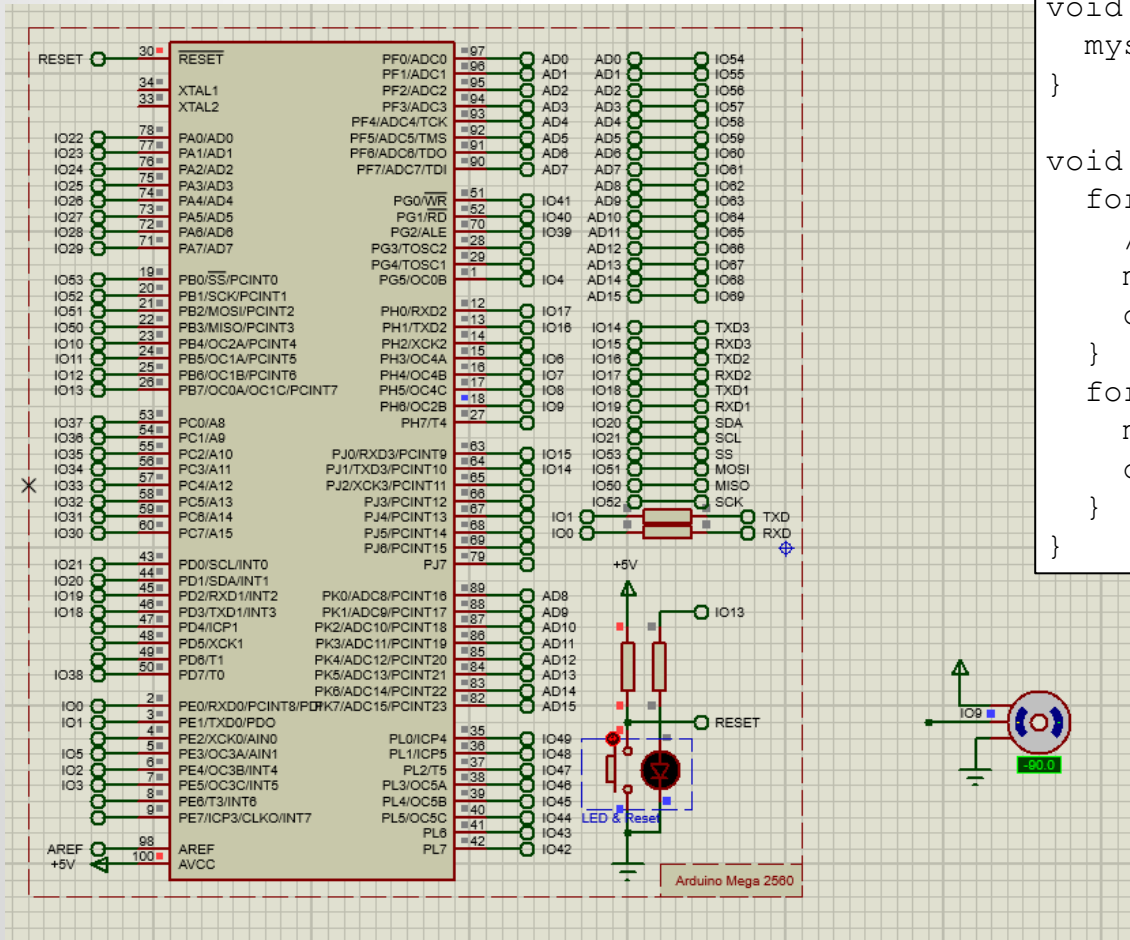
# Povezivanje

- Servo motor ima tri žice
  - Power
  - Ground
  - Signal
- Power je obično crven i vezuje se na 5V pin Arduina
- Ground je crn ili braon, povezuje se na uzemljenje
- Signal je žut, beo ili narandžast i povezuje se na digitalni pin Arduino ploče
- Napomena
  - Servo zahteva dosta struje, pa u slučaju više od jednog, može biti neophodno dodatno napajanje (nije dovoljan 5V pin Arduina)
  - Povezati ground Arduina i eksternog napajanja zajedno



# Primer 5.2

- Realizovati Arduino-zasnovani sistemi za upravljanje servo osovinom – od 0 do 180 i nazad.



```
#include <Servo.h>
```

```
Servo myservo; // kreiramo servo objekat za upravljanje motorom
```

```
int pos = 0; // trenutna servo pozicija
```

```
void setup() {  
    myservo.attach(9); // povezujemo servo na IO9 pin  
}
```

```
void loop() {  
    for (pos = 0; pos <= 180; pos += 10) { // okreće se od 0 do 180 stepeni  
        //korak 10 stepeni  
        myservo.write(pos); // postaviti servo u zeljenu poziciju  
        delay(100); // cekamo da dodje fizicki gde treba  
    }  
    for (pos = 180; pos >= 0; pos -= 10) { // okretanje nazad  
        myservo.write(pos);  
        delay(100);  
    }  
}
```