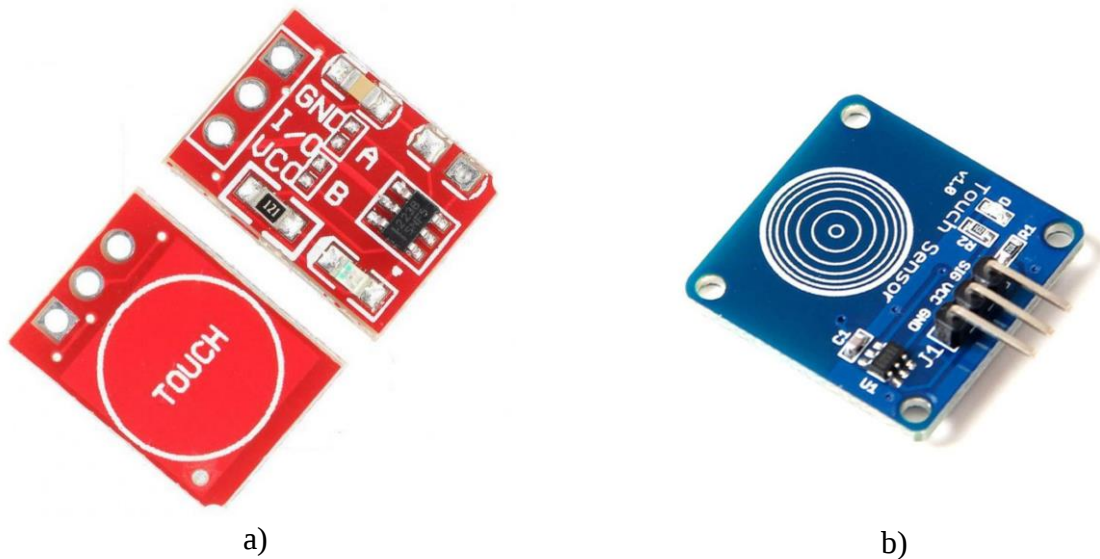


Senzor dodira – *Touch* senzor

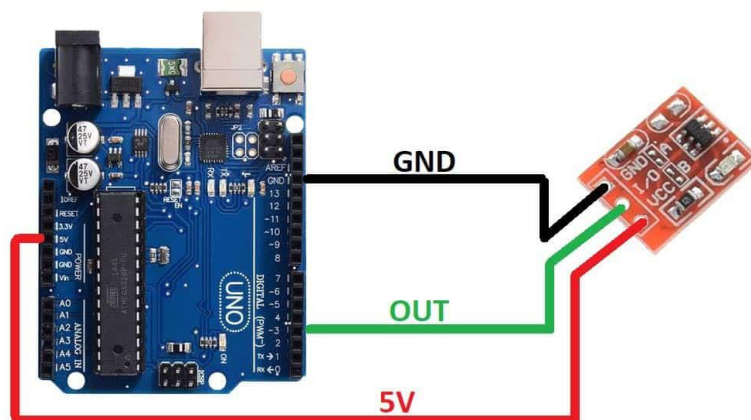
Touch senzori su uređaji koji služe za detektovanje fizičkog dodira ili pritiska na svojoj površini. Na vežbama se koristi tip senzora koji radi na osnovu kapacitivnosti (kapacitivni senzori dodira): detektuju promenu kapacitivnosti nastalu usled prisustva prsta ili drugog provodnog objekta na svojoj površini.



Slika 1. Senzori dodira

Senzor dodira ima tri pina, koji se sa Arduinoom povezuju na sledeći način:

1. **GND** pin – povezuje se na GND pin Arduinoa
2. **VCC** pin – povezuje se na VCC pin Arduinoa (5V)
3. **SIGNAL** pin – povezuje se na jedan od digitalnih pinova Arduinoa



Slika 2. Povezivanje senzora dodira sa Arduinoom

Kada senzor nije dodirnut *SIGNAL* pin ima vrednost logičke nule (*LOW*). U suprotnom na datom pinu je vrednost logičke jedinice (*HIGH*).

Za rad sa senzorom nije potrebna nikakva dodatna biblioteka. Nakon povezivanja senzora potrebno je samo odgovarajući pin inicijalizovati kao ulazni. Nakon toga se po potrebi čita vrednost na datom pinu.

Na listingu 1 prikazan je program kojim se usled dodira putem serijske komunikacije ispisuje 1, a u suprotnom 0. Podrazumeva se da je *SIGNAL* pin povezan na digitalni pin 4.

```
#define TOUCH_PIN 4

bool touch;

void setup()
{
    Serial.begin(9600);
    pinMode(TOUCH_PIN, INPUT);
    touch = 0;
}

void loop()
{
    touch = digitalRead(TOUCH_PIN);
    Serial.println(touch);
}
```

Listing 1. Program za detekciju dodira senzora

Softversko diferenciranje

Ukoliko je cilj da se jednim dodirom izvrši jedinstvena aktivnost, odnosno da se data aktivnost izvrši isključivo pri promeni stanja ($0 \rightarrow 1$ ili $1 \rightarrow 0$), potrebno je primeniti tehniku zvanu “softversko diferenciranje”. Za implementaciju ove tehnike potrebne su dve promenljive kojima se označavaju prethodno i trenutno stanje. Ideja je da se ta dva stanja porede i u slučaju njihovog razlikovanja izvršava se izvesna aktivnost.

Na listingu 2 prikazan je program kojim se implementira brojač čija se vrednost uvećava i ispisuje samo na početku dodira senzora.

```
#define TOUCH_PIN 4

bool touch_old, touch_new;
unsigned int counter;

void setup()
{
```

```

    Serial.begin(9600);
    pinMode(TOUCH_PIN, INPUT);

    touch_old = 0;
    touch_new = 0;
    counter = 0;

    counterPrint();
}

void loop()
{
    touch_new = digitalRead(TOUCH_PIN);
    if (touch_new != touch_old)
    {
        if (touch_new)
        {
            counter++;
            counterPrint();
        }

        touch_old = touch_new;
    }
}

void counterPrint()
{
    Serial.print("Brojac: ");
    Serial.println(counter);
}

```

Listing 2. Program za implementaciju brojača

Funkcija *millis()*

Ova funkcija vraća broj milisekundi od kako je Arduino počeo da izvršava program. Ova vrednost će prekoračiti opseg (odnosno vratiti se na nulu) nakon približno 50 dana. Povratna vrednost funkcije je tipa *unsigned long*.

Na listingu 3 je prikazan je program kojim se ispisuje poruka svakih 5 sekundi.

```

#define DESIRED_INTERVAL 5000

unsigned long time_old, time_new;

void setup()
{
    Serial.begin(9600);
    Serial.println("Program je započeo sa radom!");
}

```

```

    time_old = millis();
}

void loop()
{
    time_new = millis();
    if (time_new - time_old >= DESIRED_INTERVAL)
    {
        Serial.println("Proslo je 5 sekundi!");

        time_old = time_new;
    }
}

```

Listing 3. Program za detekciju željenog vremenskog intervala

Servo motor

Standardni servo motori su aktuatori koji dozvoljavaju precizno upravljanje pozicijom. Njegova karakteristika je da je ugao motora u intervalu 0 – 180 stepeni. Drugim rečima, može da napravi polovinu rotacije.

Servo motor, kao i drugi motori, je u osnovi samo DC motor, ali sa dodatnim karakteristikama:

- **kontrolno kolo** za kontrolisanje motora, odnosno podešavanje ugla,
- **zupčanici** za pretvaranje brzine u obrtni moment i
- **potenciometar** za praćenje ugla. Ovo omogućava servu „da zna gde se nalazi“.

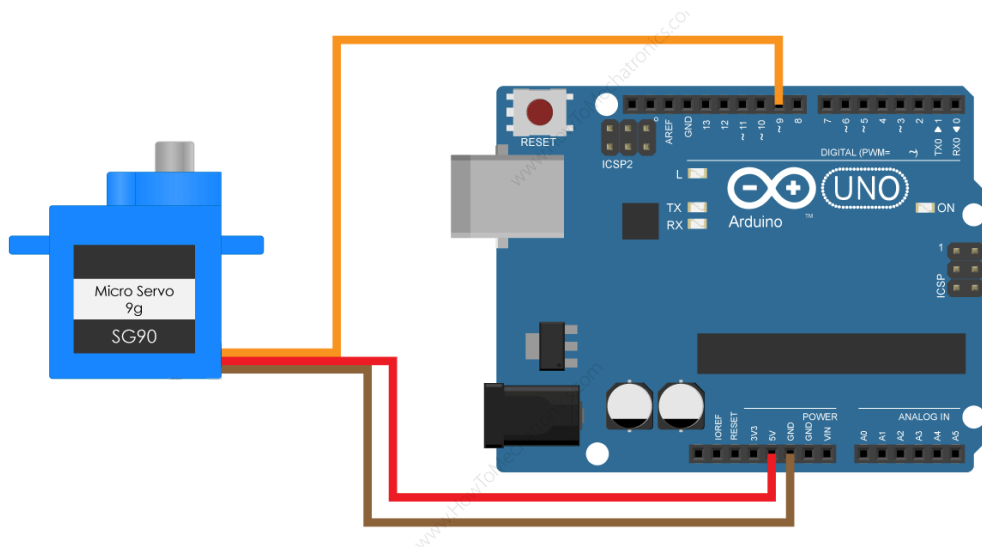


Slika 3. Servo motor

Servo motor ima tri žice, koje se sa Arduinoom povezuju na sledeći način:

1. **GND** žica (**BRAON**) – povezuje se na GND pin Arduinoa

2. **VCC** žica (**CRVENA**) – povezuje se na VCC pin Arduina (5V)
3. **SIGNAL** žica (**NARANĐASTA**) – povezuje se na jedan od digitalnih pinova Arduina koji podržava *PWM*.



Slika 4. Povezivanje servo motora sa Arduinoom

Za rad sa servom motorom u programu je potrebno uključiti postojeću Servo biblioteke i kreirati objekat klase Servo. Za inicijalizaciju serva na određenom pinu koristi se metoda *attach*, kojoj se prosleđuje vrednost datog pina kao parametar. Kasnije se pozicija serva određuje pozivanjem metodom *write*, kojoj se prosleđuje željena pozicija kao parametar.

Na listingu 4 prikazan je program kojim se servo u punom opsegu rotira naizmenično u jednu, pa u drugu stranu. Podrazumeva se da je *SIGNAL* pin povezan na digitalni pin 9.

```
#include <Servo.h>

#define SERVO_PIN 9

Servo servo;
int pos;

void setup()
{
    servo.attach(SERVO_PIN);
    pos = 0;
    servo.write(pos);
}

void loop()
{
    while(pos < 180)
    {
        servo.write(pos);
```

```
        delay(20);  
        pos++;  
    }  
  
    while(pos > 0)  
    {  
        servo.write(pos);  
        delay(20);  
        pos--;  
    }  
}
```

Listing 4. Program za rotiranje servo motora

Zadaci za vežbanje

1. Napisati program koji proverava brzinu korisnika. Od korisnika se očekuje da u roku od 3 sekunde što više puta pritisne senzor dodira, nakon čega se putem serijske komunikacije ispisuje rezultat.
2. Napisati program koji proverava reflekse korisnika. U prozvoljnom trenutku potrebno je da se upali LED dioda koja je ugrađena na ploči, nakon čega korisnik treba što pre da odreaguje i pritisne senzor dodira. Postupak je potrebno izvršiti ukupno 10 puta, nakon čega se kao rezultat ispisuje prosečno vreme reakcije korisnika.
3. Napisati program koji ima tri funkcionalnosti: mirovanje servo motora, njegovo rotiranje u jednu stranu i rotiranje u drugu stranu. Funkcionalnosti se menjaju pritiskom senzora dodira.
4. Napisati program za kontrolisanje ugla servo motora pomoću senzora dodira. Ukoliko je interval dodira kraći od pola sekunde servo se zarotira za 15 stepeni u jednu stranu, a u slučaju dužeg dodira neophodno je da se zarotira za istu vrednost u drugu stranu.
5. Napisati program kojim se implementira funkcionalnost štoperice. Štoperica je inicijalno zaustavljena. Pokretanje i zaustavljanje štoperice se vrši putem kraćeg pritiska senzora dodira (kraćeg od pola sekunde). Kada je štoperica pokrenuta, svake sekunde se serijskim putem ispisuje vreme u formatu „hh:mm:ss“. Pri tome, servo motor ima funkciju kazaljke za sekunde, tako da se tokom 60 sekundi rotira u celom opsegu u jednu stranu. Nakon tog perioda, odnosno nakon uvećanja minute, brzo se vraća u drugu stranu i nastavlja osnovno rotiranje. U slučaju dužeg pritiska senzora dodira, štoperica se resetuje i zaustavlja.
6. Napisati program kojim se implementira funkcionalnost sata sa dodatnom funkcionalnošću alarma. Vreme na satu se ispisuje na istini način kao i u prethodnom zadatku, takođe sa funkcionalnošću kazaljke za sekunde. Alarm se podešava tako što se u proizvoljnom trenutku serijskim putem unosi vreme u formatu „hh:mm:ss“ (24-časovni format). Kada se vreme sata podudara sa unesenim vremenom, pali se alarm tako što ugrađena dioda počne da treperi. Treperenje traje dok se god alarm ne ugasi putem kratkotrajnog pritiska senzora dodira (kraće od pola sekunde). I tokom treperenja sat funkcioniše na isti način. U slučaju dužeg pritiska tastera dodira, vreme se pomera za 1 sat unapred (sa 23:xx:xx se pomera na 00:xx:xx). Ukoliko želimo da poništimo alarm potrebno je serijskim putem uneti poruku “ALARM_STOP”.