

Nastavni predmet:	MIKROUPRAVLJAČI
Vježba: 13	Arduino – Timeri i njihov prekid
Cilj vježbe:	Rješavanje upravljačkih zadataka – rad s timer-ima/brojačima

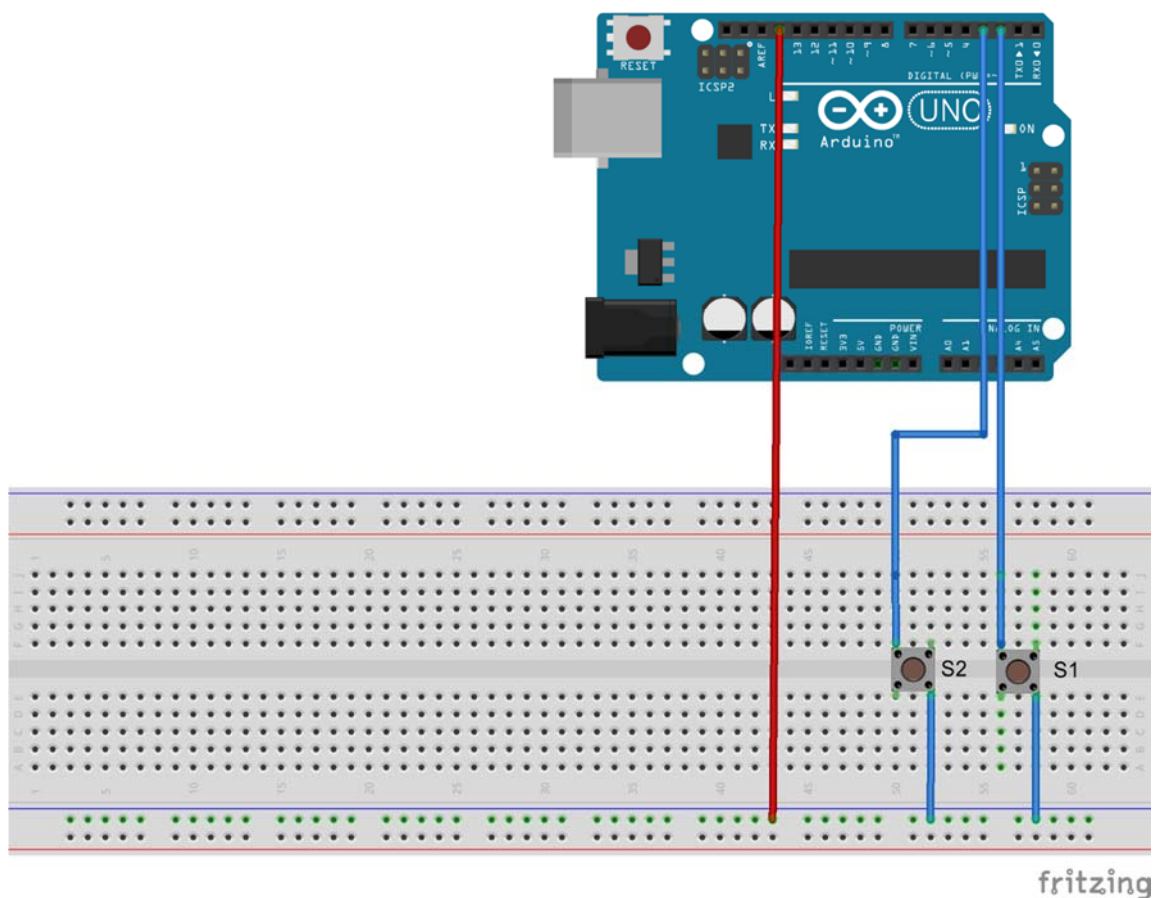
Upute

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

Zadatak 1. Koristeći TimerOne library postavi 16-bitni Timer1 te pomoću tipki S1 (start) i S2 (stop) mjeri vrijeme koje je prošlo između dva pritiska na tipke. Ispiši rezultat u milisekundama putem serijskog monitora.

Grafička shema:



Kòd zadatka

```
#include <TimerOne.h>          // Ova biblioteka sadrži kolekciju funkcija za
                                // konfiguriranje 16-bitnog hardverskog tajmera
                                // naziva Timer1 na ATmega168/328 mikrokontroleru

void setup()
{
    // Inicijaliziraj digitalni pin kao izlaz
    // Na većini Arduino pločica Pin 13 spojen na ugrađenu LED diodu
    pinMode(13, OUTPUT);
    pinMode(2, INPUT_PULLUP);
    pinMode(3, INPUT_PULLUP);
    Serial.begin(9600);
    Timer1.initialize(1000); //Početak korištenja tajmera. Argument (1000)
                            //je period tajmera mikrosekundama

    interrupts();            //Omogućuje rad s prekidima
    attachInterrupt(digitalPinToInterrupt(2), start, FALLING);
    attachInterrupt(digitalPinToInterrupt(3), Stop, FALLING);
}
uint16_t msec=0;
void loop()
{

}

/// -----
/// Prekidna rutina koju poziva tajmer
/// -----
void timerIsr(){
    msec++;
}

void start(){
    Timer1.attachInterrupt( timerIsr ); // Navođenje servisne rutine tajmera
    Serial.println("Poceo brojati...");
}

void Stop(){
    Timer1.detachInterrupt();
    Serial.print("Proteklo je: ");
    Serial.print(msec);
    Serial.println("ms");
    msec=0;
}
```

Zadatak 2. Ponovi prethodni zadatak tako da se izmjereno vrijeme umjesto na serijski monitor ispisuje na LCD zaslon (I2C). Vrijeme umjesto u milisekundama ispisati u sekundama.

Kòd zadatka

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <TimerOne.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // definiranje I2C adrese LCD-a i
                                     // veličine LCD-a 16 znakova u 2 retka

bool startLCDv = 0; //Deklaracija boolean varijabli potrebne za ispis na LCD
bool stopLCDv = 0;

void setup()
{
    lcd.begin();           // inicializiranje LCD-a
    lcd.backlight();       // uključivanje pozadinskog osvjetljenja LCD-a

    pinMode(2, INPUT_PULLUP); //Inicijalizacija pinova na kojima se nalaze
                              //tipkala
    pinMode(3, INPUT_PULLUP);
    Timer1.initialize(1000); //Inicijalizacija timera s periodom 1000
                              //mikrosekundi
    interrupts();           //Pokretanje prekida
    attachInterrupt(digitalPinToInterrupt(2), Start, FALLING); //Definiranje
                                                                //prekidne rutine na pin
                                                                //2 za Start funkciju
    attachInterrupt(digitalPinToInterrupt(3), Stop, FALLING); //Definiranje
                                                                //prekidne rutine na pin
                                                                //3 za Stop funkciju
}
uint16_t msec=0;
void loop()
{
    if (startLCDv == 1){
        lcd.clear();
        lcd.setCursor(0,0);           // postavljanje kursora na
                                      // lokaciju 0,0 (stupac, redak)
        lcd.print("Poceo brojati.."); // ispisivanje poruke
        startLCDv = 0;
    }

    if (stopLCDv==1){
        lcd.clear();
        lcd.setCursor(0,0);           // postavljanje kursora na
                                      // lokaciju 0,0 (stupac, redak)
        lcd.print("Proteklo je: ");
        lcd.setCursor(0,1);          // postavljanje kursora na
                                      // lokaciju 0,1 (stupac, redak)
```

```

    lcd.print(msec / 1000);           // prikaz proteklog vremena u sekundama

    lcd.println("ms");
    msec=0;
    stopLCDv=0;
}

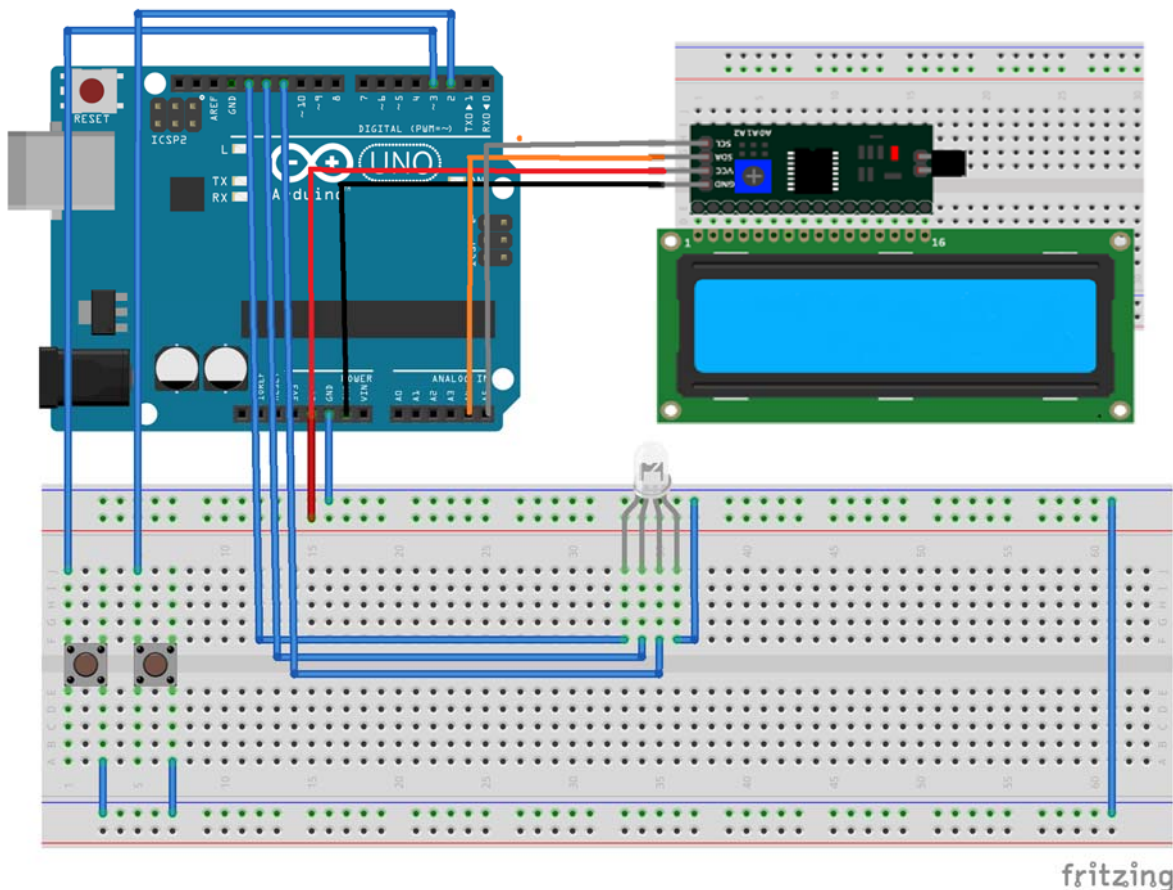
/// -----
/// Prekidne rutine koje pozivamo tipkalima
/// -----
void timerIsr(){ //Prekidna rutina koju poziva tajmer i koja broji vrijeme
    msec++;
}

void Start(){ // Rutina koja započinje brojanje
    startLCDv=1; // Postavljanje boolean varijable za ispisivanje poruke na LCD
    Timer1.attachInterrupt( timerIsr ); // Definiramo ISR timerIsr koju
                                         //poziva tajmer
}

void Stop(){ // Rutina koja prekida brojanje
    Timer1.detachInterrupt(); // Odspajanje timerIsr rutine s interrupta
    stopLCDv=1; // Postavljanje boolean varijable za ispisivanje vremena na LCD
}

```

Zadatak 3. Prethodni zadatak proširi dodavanjem RGB LED diode. Također proširi program tako da dioda svijetli crveno ako je vrijeme između pokretanja i zaustavljanja brojanja kraće od 1000ms, zeleno ako je vrijeme između 1000 i 2000 ms te plavo ako je vrijeme dulje od 2000ms.



Kòd zadatka

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <TimerOne.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // definiranje I2C adrese LCD-a i
                                     // veličine LCD-a 16 znakova u 2 retka

bool startLCDv = 0; //Deklaracija boolean varijabli potrebne za ispis na LCD
bool stopLCDv = 0;

int R = 13;
int G = 12;
int B = 11;

void setup()
void setup()
{
    lcd.begin();           // inicializiranje LCD-a
    lcd.backlight();       // uključivanje pozadinskog osvjetljenja LCD-a

    pinMode(2, INPUT_PULLUP); //Inicijalizacija pinova na kojima se nalaze
                              //tipkala
    pinMode(3, INPUT_PULLUP);
    Timer1.initialize(1000); //Inicijalizacija timera s periodom 1000
                              //mikrosekundi
    interrupts();           //Pokretanje prekida
    attachInterrupt(digitalPinToInterrupt(2), Start, FALLING); //Definiranje
                                                                //prekidne rutine na pin
                                                                //2 za Start funkciju
    attachInterrupt(digitalPinToInterrupt(3), Stop, FALLING); //Definiranje
                                                                //prekidne rutine na pin
                                                                //3 za Stop funkciju
}
uint16_t msec=0;
void loop()
{
    if (startLCDv == 1){
        lcd.clear();
        lcd.setCursor(0,0);           // postavljanje kursora na
                                      // lokaciju 0,0 (stupac, redak)
        lcd.print("Poceo brojati.."); // ispisivanje poruke
        startLCDv = 0;
    }

    if (stopLCDv==1){
        lcd.clear();
        lcd.setCursor(0,0);           // postavljanje kursora na
                                      // lokaciju 0,0 (stupac, redak)
        lcd.print("Proteklo je: ");
        lcd.setCursor(0,1);           // postavljanje kursora na
                                      // lokaciju 0,1 (stupac, redak)
```

```

lcd.print(msec / 1000);           // prikaz proteklog vremena u sekundama
lcd.println("ms");

if(msec < 1000){
    Serial.println("R");
    digitalWrite(R, 1);
    delay(500);
    digitalWrite(R, 0);
    delay(500);
}
if(msec > 1000 && msec < 2000){
    Serial.println("G");
    digitalWrite(G, 1);
    delay(500);
    digitalWrite(G, 0);
    delay(500);
}
if(msec > 2000){
    Serial.println("B");
    digitalWrite(B, 1);
    delay(500);
    digitalWrite(B, 0);
    delay(500);
}

msec=0;
stopLCDv=0;
}
}

/// -----
/// Prekidne rutine koje pozivamo tipkalima
/// -----
void timerIsr(){ //Prekidna rutina koju poziva tajmer i koja broji vrijeme
    msec++;
}

void Start(){ // Rutina koja započinje brojanje
    startLCDv=1; // Postavljanje boolean varijable za ispisivanje poruke na LCD
    Timer1.attachInterrupt( timerIsr ); // Definiramo ISR timerIsr koju
                                         //poziva tajmer
}

void Stop(){ // Rutina koja prekida brojanje
    Timer1.detachInterrupt(); // Odspajanje timerIsr rutine s interrupta
    stopLCDv=1; // Postavljanje boolean varijable za ispisivanje vremena na LCD
}

```