

UNIVERZITET U SARAJEVU



ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

## Laboratorijska vježba 5

---

# Prekidi i tajmeri

---

**Bazdar Faris**

*Predmet: Ugradbeni sistemi*

Akadska godina: 2019/2020

Sarajevo, april 2020.

## 1 Zadatak 1 alternativa

```
#include "mbed.h"
#include "C12832.h"
#include "stdio.h"

C12832 lcd(SPI_MOSI, SPI_SCK, SPI_MISO, p8, p11);

Ticker t1;
InterruptIn dugmeSS(p8);
InterruptIn dugmeR(p9);

int zastavica = 0;
float stopanje = 0;

void STARTSTOP() {
    zastavica = !(zastavica);
}

void RESET() {
    lcd.cls();
    stopanje = 0;
    zastavica = 0;
}

void BROJANJEPRIKAZ() {
    if(zastavica == 1 && stopanje < 99.8) stopanje += 0.1;
    else if(zastavica == 1 && stopanje >= 99.8) {stopanje = 0;
        lcd.cls();}
    lcd.locate(0,3);
    lcd.printf("\%04.1f", stopanje);
}

int main() {
    dugmeSS.rise(&STARTSTOP);
    dugmeR.rise(&RESET);
    t1.attach(&BROJANJEPRIKAZ, 0.1);
    wait_ms(100);
}
```

**Opis koda:** Na početku se inicijalizira "ticker" t1 koji će služiti da ponavlja izvršavanje neke funkcije svakih n sekundi. Definirane su pomoćne funkcije koje se zovu pri uzlaznim ivicama na pinovima p8, p9. Funkcija "STARTSTOP()" mijenja vrijednost globalne varijable "zastavica". Svrha te varijable je da kontroliše kada treba, a kada ne mjeriti vrijeme. Funkcija "RESET()" služi da vrati globalnu varijablu "stopanje" koja predstavlja trenutnu vrijednost prikaza na štoperici na nulu i da vrati vrijednost zastavice na nulu. Ukoliko je zastavica na jedinici, onda je potrebno da program mjeri vrijeme, a ukoliko je zastavica na nuli onda ne mjeri. Funkcija "BROJANJEPRIKAZ()" vrši ažuriranje varijable "stopanje" ukoliko je to potrebno, te prikazuje vrijednost na sam display. Unutar maina se nalazi umjesto tajmera funkcija "wait", pored toga na uzlazne ivice se zovu funkcije "STARTSTOP i RESET", a periodično se poziva funkcija "BROJANJEPRIKAZ", periodom  $T =$

0.1s. Pošto je u postavi rečeno da display treba sadržavati "dvije cifre za sekunde + jedna decimala za desetinke" uvedeno je resetovanje ukoliko bi štopanje vremena prešlo preko 99.9 s.

## 2 Zadatak 2 alternativa

```
#include "mbed.h"

Ticker t1;

BusOut prikaz1 (p8, p7, p6, p5);
BusOut prikaz2 (p12, p11, p10, p9);

InterruptIn taster(BUTTON1);

const float T(2);
int brojac1(0);
int brojac2(0);

int zastavica = 0;

void STARTSTOP(){
    zastavica = !(zastavica);
}

void PromjenaPrikaz(){
    if(zastavica == 1){
        brojac2 = (brojac2 + 1) % 16;
        prikaz2 = brojac2;
    }
    brojac1 = (brojac1 + 1) % 16;
    prikaz1 = brojac1;
}

int main () {
    prikaz1 = brojac1 ;
    prikaz2 = brojac2 ;
    taster.rise(&STARTSTOP);
    taster.fall(&STARTSTOP);
    t1.attach(&PromjenaPrikaz , T);
    wait(T);
}
```

### Opis koda i dodatni komentari traženi u izvještaju:

1. U 2 BusOut-a postavljeni su LED diode sa svrhom da rade kao binarni brojač (0-15), "prikaz1" prikazuje svakih T vrijednost broja koji se povećava nakon svakog T. Ukoliko se taster drži to povećava broj za jedan i prikazuje na "prikaz2". Ne koristi se programiranje vođeno događajima.
2. Ukoliko se promijeni konstanta T na 2.0 sve više dolazi u izražaj mana kod izbjegavanja programiranja vođenog događajima. Sada su periodične promjene "prikaz1" dosta sporije i držanje tastera nije pravilno registrirano zbog toga što se očitava da li je pritisnuto dugme svake 2 sekunde, a ne odma kada se dugme pritisne. Ovo je

sve moguće pravilno realizirati korištenjem prekida.

3. U prilogu je dat kod koji je realiziran korištenjem prekida i tickera. Postavljena je zastavica pomoću koje se može zaključiti da li je dugme pritisnuto. Promjena stanja na LED diodama se vrši pomoći tickera koji se periodično izvršava.