Univerzitet u Sarajevu
Elektrotehnički fakultet
**Ugradbeni sistemi 2023 / 24**

# Izvještaj za laboratorijsku vježbu br. 1

Razvojni sistem picoETF i MicroPython

Ime i prezime: **Ismar Muslić**
Broj index-a: **19304**

**12. mart 2023. godine**

# Sadržaj

# 1 Pseudokod

S obzirom na to da su samo 4. i 5. zadatak od redovnih, kao i 2. zadatak za dodatne bodove zahtijevali programsko rješenje, pseudokod će u ovom izvještaju biti dat samo za te zadatke.

## 1.1 Zadatak 4

```
while(true)
      if(taster4 aktiviran)
        led7 = 1
```

## 1.2 Zadatak 5

```
while(true)
      if(taster1 aktiviran)
        uvećaj_brojač()
        blokiraj_tastere()

      if(taster2 aktiviran)
        umanji_brojač()
        blokiraj_tastere()

      if(taster3 aktiviran)
        ugasi_sve_diode()
        blokiraj_tastere()

      if(taster4 aktiviran)
        upali_sve_diode()
        blokiraj_tastere()
```

## 1.3 Zadatak za dodatne bodove 2

```
while(true)
      while(trajanje < 1.1)
        for(boja in boje[])
          RGB(boja)
          Sleep(trajanje)
        Trajanje += 0.1

      while(trajanje > 0.1)
        for(boja in boje[])
          RGB(boja)
          sleep(trajanje)
        trajanje -= 0.1
```

# 2 Analiza programskog rješenja

## 2.1 Zadatak 1

U zadatku 1 bilo je potrebno srušiti stari i flashati novi firmware mikrokontrolera Raspberry Pi Pico W (RP2040). To se postiglo konekcijom kontrolera na PC uz držanje tastera „BOOTSEL“ koji je na istom ugrađen, čime se Raspberry Pi povezuje u mass storage načinu rada. Prvo je na mass storage device prebačen fajl `flash_nuke.uf2` koji je očistio firmware particiju, a zatim je na isti način novi, „čist“ firmware flashan na uređaj. Verzija flashanog firmware-a je v1.22.2.

## 2.2 Zadatak 2

U zadatku 2, bilo je potrebno upoznati se sa razvojnim okruženjem Thonny, koje se koristi za rad s picoETF sistemom. Potrebno je bilo koristiti REPL komandnu liniju, te izvršavati komande kako bi se upravljalo sa GPIO pinovima mikrokontrolera. Izvršavane su naredbe za kontrolu fizičkog tastera na pinu GP0. Naredbama se moglo utvrditi logičko stanje tastera, a kreirani objekat „t1“ je služio kao varijabla koja svjedoči o tom stanju.

## 2.3 Zadatak 3

Zadatak 3 zahtijevao je kontroliranje integriranih LED dioda na razvojnom sistemu picoETF. Ove diode su povezane na pinove GP4 – GP9, te se iste pale i gase kreiranjem odgovarajućeg objekta koji odgovara pinu na koji je povezana dioda, te postavljanjem njegove vrijednosti na logičko 1 ili 0, respektivno. Također se stanje dioda može mijenjati u odnosu na drugu varijablu, pa je tako realizirano da se prilikom izvršavanja naredbe u REPL promptu LED0 pali ukoliko je pritisnut taster T1.

## 2.4 Zadatak 4

U ovom zadatku potrebno je bilo kreirati skriptu main.py koja kontroliše stanje diode LED7 na osnovu stanja tastera T4. Ukoliko je taster pritisnut, tada je potrebno da se LED7 upali i svijetli dok god je pritisnut i taster. To se postiglo na način da se incijaliziraju varijable koje odgovaraju modulima LED7 i T4 na njihovim respektivnim pinovima, te je u kontinualnu while(true) petlju upisan kod koji će paliti LED7 na osnovu stanja T4.

## 2.5 Zadatak 5

Ovaj zadatak tražio je implementaciju 8-bitnog binarnog brojača uz pomoć integriranih LED dioda LED0 – LED7. Također je traženo da tasteri T1 – T4 imaju funkcionalnosti za inkrementiranje, dekrementiranje, postavljanje svih dioda na stanje logičke jedinice i na stanje logičke nule. Implementirano rješenje zasniva se na organizaciji datih objekata koji predstavljaju pinove za tastere i LED diode u liste, s kojima se daljnjim manipulacijama postiže željeni efekat. Brojač je implementiran kao decimalna varijabla koja je u svojoj binarnoj predstavi na 8 bita raspoređena po diodama LED0 – LED7. Shiftanjem brojača udesno, te izvršavanjem logičke operacije & bit po bit, brojač se inkrementira. Detaljna implementacija je data u prilogu 3.1.

## 2.6 Dodatni zadatak 1

Potrebno je bilo u REPL promptu slati komande za kontrolu RGB LED – a. Ova dioda, kao što je očigledno, ima 3 različite boje, s kojima se u kombinaciji može dobiti 7 različitih

režima. Varijable su postavljene kao svaka od pojedinačnih boja, red, green i blue, te su kombiniranim uključivanjem i isključivanjem dobijene različite boje koje RGB LED pruža, na slijedeći način:

- Crvena + plava = ljubičasta
- Plava + zelena = tirkizna
- Crvena + zelena = žuta
- Crvena + zelena + plava = bijela.

Deklaracijom varijabli kao što je dato ispod, dobija se recimo ljubičasta boja na sljedeći način:

```
>>> red = Pin(14, Pin.OUT)
>>> green = Pin(12, Pin.OUT)
>>> blue = Pin(13, Pin.OUT)
>>> red.value(1)
>>> blue.value(1)
```

### 2.6 Dodatni zadatak 2

Potrebno je implementirati program koji reguliše RGB diodu na način da sve njene kombinacije boja promijeni u trajanju svake od po 0.1s, te da se to trajanje postepeno povećava do 1s nakon što je dioda prošla kroz sve kombinacije. Zatim trajanje svake boje opada postepeno od 1s nazad do 0.1s, te se uvećanje i smanjenje trajanja repetitivno izvršava u programu. Željeni efekat je postignut implementacijom while(true) petlje koja se neprestano izvršava na kontroleru, te liste pinova koji odgovaraju R, G i B vrijednostima RGB diode. Ove varijable su u for – petlji mijenjane proizvoljnim redoslijedom, tako da dioda prođe kroz sve boje, a varijabilno trajanje svake iteracije niza boja je implementirano pomoću jedne dodatne floating – point varijable koja se kroz for – petlje kontinutalno uvećava za 0.1s, a zatim, kada dostigne vrijednost od 1s, umanjuje za 0.1s sve do početnog trajanja od 0.1s, te se tada glavna while(true) petlja ponavlja.

## 3 Korišteni hardverski resursi

Za potrebe laboratorijske vježbe 1 korišten je razvojni sistem picoETF u kojem su integrisani svi potrebni elementi.

Ovi elementi u kontekstu prve vježbe uključuju:

- Tasteri (4 kom.)
- LED (8 kom.)
- RGB LED (1 kom.)

Kao što je navedeno, svi hardverski resursi su integrisani u razvojni sistem picoETF, te su kao takvi i korišteni u sklopu vježbe.

# 4 Zaključak

Osim manjih prepreka prilikom korištenja programskog jezika Python kao novine, nije bilo problema pri izvođenju laboratorijske vježbe 1. Python kao novi programski jezik nije nešto s čime smo se do sada susretali u kontekstu sintakse i upotrebljivosti. Njegova sintaksa je dosta liberalnija u smislu deklaracije i korištenja varijabli i njihovih tipova, te se suštinski po tome razlikuje od sintakse programskih jezika kao što su C/C++, na šta se bilo potrebno navići. Cilj vježbe je bio upoznavanje sa radom u laboratoriji i razvojnim sistemom picoETF, što je i postignuto.

# 5 Prilog

## 5.1 Zadatak 5/Izvorni kod

```
 1. import time
 2. from machine import Pin
 3. time.sleep(0.1) # Wait for USB to become ready
 4.
 5. leds = [Pin(i, Pin.OUT) for i in range(4, 12)] #deklracija LED-ova
 6.
 7. tasteri = [Pin(i, Pin.IN) for i in range(0, 4)] #deklaracija tastera
 8.
 9. def prikazi_brojac(broj): #funkcija za update-anje stanja LED-ova
10.   for i in range(8):
11.     leds[i].value((broj>>i)&1)
12.   time.sleep(0.2)
13.
14. brojac = 0
15.
16. while True:
17.
18.   if brojac > 255:
19.     brojac = 0
20.   elif brojac < 0:
21.     brojac = 255
22.
23.   if tasteri[0].value():
24.     brojac += 1
25.     prikazi_brojac(brojac)
26.     while tasteri[0].value():  #onemogući unos pri držanju tastera
27.       time.sleep(0.1)
28.
29.   elif tasteri[1].value():
30.     brojac -= 1
31.     prikazi_brojac(brojac)
32.     while tasteri[1].value():
33.       time.sleep(0.1)
34.
35.   elif tasteri[2].value():
36.     brojac = 0
37.     prikazi_brojac(brojac)
38.     while tasteri[2].value():
39.       time.sleep(0.1)
40.
41.   elif tasteri[3].value():
42.     brojac = 255
43.     prikazi_brojac(brojac)
44.     while tasteri[3].value():
45.       time.sleep(0.1)
46.
```

## 5.2 Zadatak za dodatne bodove 2/Izvorni kod

```
 1. import time
 2. from machine import Pin
 3. time.sleep(0.1) # Wait for USB to become ready
 4.
 5. red = Pin(14, Pin.OUT)
 6. green = Pin(12, Pin.OUT)    #deklaracija R, G i B pinova
 7. blue = Pin(13, Pin.OUT)
 8.
 9. pins = [red, green, blue]   #niz za lakše upravljanje vrijednostima
10.
11. trajanje = 0.1              #početno trajanje
12.
13. while True:
14.     while trajanje < 1.1:   #uvećavanje trajanja
15.         for lRGB in [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 0, 1], [0, 1, 1], [1, 1, 0], [1, 1,
1]]:
16.             for pin, value in zip(pins, lRGB):
17.                 pin.value(value)
18.             time.sleep(trajanje)
19.         trajanje += 0.1
20.
21.     while trajanje > 0.1:   #smanjenje trajanja
22.         for lRGB in [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 0, 1], [0, 1, 1], [1, 1, 0], [1, 1,
1]]:
23.             for pin, value in zip(pins, lRGB):
24.                 pin.value(value)
25.             time.sleep(trajanje)
26.         trajanje -= 0.1
27.
```