

UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
UGRADBENI SISTEMI (RI/TK) 2024/25

IZVJEŠTAJ ZA LABORATORIJSKU VJEŽBU BR. 1

RAZVOJNI SISTEM PICOETF I MICROPYTHON

Ime i prezime: **Vedad Gaštan**
Broj indexa: **19685**

17.03.2025.

Sadržaj

1	Uvod	1
2	Postavka zadatka	1
2.1	Zadatak 5	1
2.2	Zadatak 5-1	2
2.3	Zadatak 5-2	3
3	Implementacija	5
3.1	Zadatak 5	5
3.2	Zadatak 5-1	5
3.3	Zadatak 5-2	6
4	Testiranje i rezultati	6
5	Zaključak	7
6	Prilog	8
6.1	Zadatak 5	8
6.2	Zadatak 5-1	8
6.3	Zadatak 5-2	9

1 Uvod

Cilj ove laboratorijske vježbe je upoznavanje sa razvojnim sistemom picoETF, koji se temelji na mikrokontroleru RP2040, te njegovim programiranjem u MicroPython-u koristeći Thonny IDE. Fokus vježbe je rad sa digitalnim ulazima i izlazima, uključujući tastere, LED diode i RGB LED.

Korišteni hardver i softver:

Komponenta	Opis
picoETF	Razvojni sistem baziran na RP2040
Thonny IDE	Razvojno okruženje za MicroPython
MicroPython	Programski jezik prilagođen mikrokontrolerima
Wokwi	Simulator korišten za simuliranje picoETF-a

2 Postavka zadatka

2.1 Zadatak 5

U okviru ovog zadatka je potrebno napisati programski kod koji realizira binarni brojač na diodama LED0-LED7. Tastere T1-T4 je potrebno koristiti za upravljanje brojačem, na sljedeći način:

- pritisnut taster T1 povećava stanje brojača- brojač naprijed (vrijeme promjene stanja svake sekunde),
- pritisnut taster T2 umanjuje stanje brojača- brojač nazad (vrijeme promjene stanja svake sekunde),
- ako nije pritisnut ni jedan taster zadržava trenutno stanje
- pritisak na taster T3 postavlja stanje brojača na 0x00,
- pritisak na taster T4 postavlja stanje brojača na 0xFF.
- stanje brojača u binarnom formatu prikazuje se na LED, početno stanje brojača je 0x00, vrijeme između dva stanja implementirati pomoću softverskog kašnjenja, te nije dozvoljeno korištenje sistema prekida.

Inicijalizacija:

```
leds[4:12] ← izlazi  
t1, t2, t3, t4 ← ulazi  
cnt ← 0x00
```

while True do

```
    if t1 pritisnut then  
        cnt ← (cnt + 1) & 0xFF  
    else if t2 pritisnut then  
        cnt ← (cnt - 1) & 0xFF
```

```

else if t3 pritisnut then
    cnt ← 0x00
else if t4 pritisnut then
    cnt ← 0xFF
end if

```

```

for i = 0 to 7 do
    if cnt bit i je 1 then
        led[i] ← 1
    else
        led[i] ← 0
    end if
end for

```

Sačekaj 1 sekundu

```

end while

```

2.2 Zadatak 5-1

U ovom zadatku potrebno je implementirati binarni brojač na LED diodama LED0-LED7, pri čemu se promjena stanja brojača dešava samo na klik tastera. To znači da se stanje brojača mijenja samo pri promjeni stanja tastera sa otpušteno na pritisnuto, dok držanje tastera pritisnutim ne mijenja stanje brojača. Funkcije tastera Tasteri imaju sljedeće funkcije:

- T1: Svaki klik povećava stanje brojača (inkrement).
- T2: Svaki klik umanjuje stanje brojača (dekrement).
- T3: Postavlja brojač na 0x00.
- T4: Postavlja brojač na 0xFF.
- Ako nijedan taster nije aktivan, brojač zadržava trenutno stanje.

Dakle, potrebno je implementirati detekciju promjene stanja tastera ("edge detection") umjesto kontinuirane provjere da li je taster pritisnut. Implementaciju uraditi bez korištenja sistema prekida.

Inicijalizacija:

```

leds[4:12] ← izlazi
t1, t2, t3, t4 ← ulazi
p_t1, p_t2, p_t3, p_t4 ← 0
cnt ← 0x00

```

```

while True do
    if t1 pritisnut and not p_t1 then
        cnt ← (cnt + 1) & 0xFF
    else if t2 pritisnut and not p_t2 then
        cnt ← (cnt - 1) & 0xFF
    end if
end while

```

```

else if t3 pritisnut and not p_t3 then
    cnt ← 0x00
else if t4 pritisnut and not p_t4 then
    cnt ← 0xFF
end if

```

```

p_t1 ← t1.value()
p_t2 ← t2.value()
p_t3 ← t3.value()
p_t4 ← t4.value()

```

```

for i = 0 to 7 do
    leds[i].value((cnt » i) & 1)
end for

```

Sačekaj 0.01 sekundu

```

end while

```

2.3 Zadatak 5-2

Modifikovati zadatak 5 dodavanjem sljedećih funkcionalnosti:

- indikacija smjera brojača pomoću RGB diode:
 - crvena- aktivan brojač naprijed (Rgb)
 - plava- aktivan brojač nazad (rgB)
 - zelena- kada se setuje ili resetuje brojač. (rGb)
 - indikacija parnosti/neparnosti brojača:
 - ako je aktivan brojač naprijed i stanje brojača parno, pored crvene aktivirati i zelenu (RGb),
 - ako je aktivan brojač naprijed i stanje brojača neparno, pored crvene aktivirati i plavu (RgB), 7
 - ako je aktivan brojač nazad, a stanje brojača parno pored plave aktivirati i zelenu (rGB),
 - ako je aktivan brojač nazad, a stanje brojača neparno pored plave aktivirati i crvenu i zelenu (RGB).
-

Inicijalizacija:

```

leds[4:12] ← izlazi
t1, t2, t3, t4 ← ulazi
r, g, b ← izlazi
cnt ← 0x00

```

```

while True do

```

```

if t1 pritishnut then
  cnt ← (cnt + 1) & 0xFF
  g.value(0)
  b.value(0)
  r.value(1)
  g.value(not (cnt & 1))
  b.value(cnt & 1)
else if t2 pritishnut then
  cnt ← (cnt - 1) & 0xFF
  r.value(0)
  g.value(0)
  b.value(1)
  g.value(not (cnt & 1))
  r.value(cnt & 1)
else if t3 pritishnut then
  cnt ← 0x00
  r.value(0)
  b.value(0)
  g.value(1)
else if t4 pritishnut then
  cnt ← 0xFF
  r.value(0)
  b.value(0)
  g.value(1)
end if

for i = 0 to 7 do
  leds[i].value((cnt » i) & 1)
end for

  Sačekaj 1 sekundu
  r.value(0)
  b.value(0)
  g.value(0)
end while

```

3 Implementacija

3.1 Zadatak 5

Koraci implementacije:

1. **Inicijalizacija pinova:** Za 8 LED dioda koristi se pinovi od 4 do 11, postavljeni kao izlazi (`Pin.OUT`). Četiri tastera su povezana sa pinovima od 0 do 3, postavljeni kao ulazi (`Pin.IN`).
2. **Postavke brojača:** Varijabla `cnt` se inicijalizira na 0 (0x00).
3. **Glavna petlja:** U beskonačnoj petlji (`while True`), provjerava se stanje svakog tastera (`t1`, `t2`, `t3`, `t4`):
 - Ako je pritisnut taster `t1`, broj se povećava za 1.
 - Ako je pritisnut taster `t2`, broj se smanjuje za 1.
 - Ako je pritisnut taster `t3`, broj se resetuje na 0.
 - Ako je pritisnut taster `t4`, broj se postavlja na 255 (0xFF).
4. **Prikaz na LED diodama:** Vrijednost `cnt` se prikazuje na 8 LED dioda, tako da svaki bit broja `cnt` odgovara jednoj LED diodi.
5. **Odgoda:** U petlji se dodaje kašnjenje od 1 sekunde korištenjem `time.sleep(1)`.

Korištene funkcije:

- `Pin()`: Inicijalizacija pinova za ulaze i izlaze.
- `value()`: Čitanje ili postavljanje vrijednosti na pin.
- `time.sleep()`: Dodavanje kašnjenja u petlji.

3.2 Zadatak 5-1

Koraci implementacije:

1. **Inicijalizacija pinova:** Pinovi za LED diode i dugmadi ostaju isti kao u prvom zadatku.
2. **Praćenje prethodnog stanja:** Dodane su varijable `p_t1`, `p_t2`, `p_t3`, `p_t4` koje prate prethodno stanje dugmadi, kako bi se detektovali promjene stanja (debouncing).
3. **Glavna petlja:** U glavnoj petlji, program provjerava stanje svakog tastera:
 - Ako je taster `t1` pritisnut, a prethodno nije bilo pritisnuto, broj se povećava.
 - Ako je taster `t2` pritisnut, a prethodno nije bilo pritisnuto, broj se smanjuje.
 - Ako je taster `t3` pritisnut, a prethodno nije bilo pritisnuto, broj se resetuje na 0.
 - Ako je taster `t4` pritisnuto, a prethodno nije bilo pritisnuto, broj se postavlja na 255 (0xFF).
4. **Ažuriranje prethodnog stanja:** Na kraju svake iteracije, stanje svakog tastera se sprema u odgovarajuće varijable (`p_t1`, `p_t2`, `p_t3`, `p_t4`) kako bi se pratilo da li je taster prethodno bio pritisnut.

5. **Prikaz na LED diodama:** Vrijednost `cnt` se prikazuje na 8 LED dioda.
6. **Odgođeno:** U petlji se dodaje kašnjenje od 0.01 sekunde korištenjem `time.sleep(0.01)`.

Korištene funkcije:

- `Pin()`: Inicijalizacija pinova za ulaze i izlaze.
- `value()`: Čitanje ili postavljanje vrijednosti na pin.

3.3 Zadatak 5-2

Koraci implementacije:

1. **Inicijalizacija pinova:** Pinovi za LED diode (od 4 do 11) su postavljeni kao izlazi (`Pin.OUT`). Također, dodani su pinovi za RGB LED (`r`, `g`, `b`) kako bi se upravljalo bojama.
2. **Glavna petlja:** U glavnoj petlji, program provjerava stanje svakog tastera:
 - Ako je taster `t1` pritisnut, brojač se povećava, a pali se crvena RGB lampica, te se pali zelena boja ako je brojač paran, a plava ako je brojač neparan.
 - Ako je taster `t2` pritisnuto, brojač se smanjuje, a pali se plava RGB lampica, te se pali zelena boja ako je brojač paran, a crvena ako je brojač neparan.
 - Ako je taster `t3` pritisnut, brojač se resetuje na 0, a boja se postavlja na zelenu.
 - Ako je taster `t4` pritisnut, brojač se postavlja na 255, a boja se također postavlja na zelenu.
3. **Prikaz na LED diodama:** Vrijednost broja `cnt` se prikazuje na 8 LED dioda.
4. **Odgođeno:** U petlji se dodaje kašnjenje od 1 sekunde korištenjem `time.sleep(1)`.
5. **Resetovanje boja:** Na kraju svake iteracije boje RGB LED dioda se isključuju (`r.value(0)`, `g.value(0)`, `b.value(0)`).

Korištene funkcije:

- `Pin()`: Inicijalizacija pinova za ulaze, izlaze i RGB LED.
- `value()`: Čitanje ili postavljanje vrijednosti na pin.
- `time.sleep()`: Dodavanje kašnjenja u petlji.

4 Testiranje i rezultati

Testiranje je izvršeno pomoću Thonny IDE-a i picoETF razvojnog sistema.

Rezultati testiranja:

- T1 pravilno povećava broj prikazan na LED diodama.
- T2 pravilno smanjuje broj.
- T3 i T4 resetuju i postavljaju brojač na odgovarajuće vrijednosti.
- RGB LED pravilno prikazuje boje prema smjeru brojača i parnosti/neparnosti.

5 Zaključak

U okviru implementacije svih zadataka postignuti su ključni ciljevi vezani za kontrolu i prikaz broja pomoću LED dioda, uz dodatnu interaktivnost putem četiri tastera. Svaki zadatak se temeljio na osnovnim principima rada sa mikrokontrolerom, kao što su čitanje stanja pinova, upravljanje izlazima i implementacija osnovnih petlji.

Implementirani su svi zadaci u skladu sa zahtjevima:

- U prvom zadatku implementirano je osnovno brojanje i prikaz broja na LED diodama, gdje je broj povećavan, smanjivan, resetovan ili postavljen na maksimalnu vrijednost, ovisno o pritisku tastera.
- Drugi zadatak obuhvatio je edge detection, što je omogućilo da se broj mijenja samo prilikom klika na taster, a ne prilikom kontinualnog držanja tastera.
- U trećem zadatku dodana je funkcionalnost RGB LED dioda, čime je rad sistema postao vizualno predstavljiv kroz promjene boja LED dioda.

Iako je implementacija uspješno izvršena, postoje mogućnosti za dalja poboljšanja:

- **Optimizacija debouncinga:** Trenutna implementacija debouncinga koristi odgodu (kašnjenje) kako bi se spriječile višestruke promjene broja. Dodatno poboljšanje moglo bi uključivati hardverski debouncing sa kondenzatorima ili korištenje naprednijih algoritama za softverski debouncing.
- **Raznovrsniji prikazi na LED diodama:** Umjesto binarnog prikaza broja, mogli bi se implementirati razni vizualni efekti na LED diodama koji bi omogućili bolju prezentaciju broja ili drugih podataka.
- **Dodavanje interfejsa za korisničko praćenje:** Moguće je dodati LCD ekran ili serijski interfejs koji bi omogućio korisniku da vidi trenutni broj i stanje svih pinova, što bi olakšalo debugiranje i interakciju.

Ova implementacija je omogućila bolje razumijevanje rada sa mikrokontrolerima i osnovama upravljanja hardverskim komponentama. Kroz ovu laboratorijsku vježbu naučeno je:

- Kako pravilno koristiti pinove za čitanje i pisanje podataka.
- Razumijevanje osnovnih principa debouncinga i njegovog uticaja na tačnost sistema.
- Postavljanje i upravljanje RGB LED diodama, kao i njihovo korištenje za vizualno predstavljanje podataka.
- Korištenje osnovnih funkcija mikrokontrolera u stvaranju interaktivnih sistema.

6 Prilog

6.1 Zadatak 5

```
1 import time
2 import machine
3 from machine import Pin
4
5 leds = []
6 for i in range(4, 12):
7     leds.append(Pin(i, Pin.OUT))
8
9 t1 = Pin(0, Pin.IN)
10 t2 = Pin(1, Pin.IN)
11 t3 = Pin(2, Pin.IN)
12 t4 = Pin(3, Pin.IN)
13
14 cnt = 0x00
15
16 while True:
17     if t1.value():
18         cnt = (cnt + 1) & 0xFF # Uveca cnt ciklicno
19     elif t2.value():
20         cnt = (cnt - 1) & 0xFF # Smanji cnt ciklicno
21     elif t3.value():
22         cnt = 0x00
23     elif t4.value():
24         cnt = 0xFF
25
26     for i in range(0, 8):
27         leds[i].value((cnt >> i) & 1) # i-ta LED dioda ima vrijednost i-tog bita cnt-
28         time.sleep(1)
```

6.2 Zadatak 5-1

```
1 import time
2 import machine
3 from machine import Pin
4
5 leds = []
6 for i in range(4, 12):
7     leds.append(Pin(i, Pin.OUT))
8
9 t1 = Pin(0, Pin.IN)
10 p_t1 = 0 # Prethodna vrijednost t1 tastera
11 t2 = Pin(1, Pin.IN)
12 p_t2 = 0 # Prethodna vrijednost t2 tastera
13 t3 = Pin(2, Pin.IN)
14 p_t3 = 0 # Prethodna vrijednost t3 tastera
15 t4 = Pin(3, Pin.IN)
16 p_t4 = 0 # Prethodna vrijednost t4 tastera
17
18 cnt = 0x00
19
20 while True:
21     # Vrsi se rad samo ako je taster sad aktivan, a prethodni ciklus nije bio
```

```

22     if t1.value() and not p_t1:
23         cnt = (cnt + 1) & 0xFF # Uveca cnt ciklicno
24     elif t2.value() and not p_t2:
25         cnt = (cnt - 1) & 0xFF # Smanji cnt ciklicno
26     elif t3.value() and not p_t3:
27         cnt = 0x00
28     elif t4.value() and not p_t4:
29         cnt = 0xFF
30     p_t1 = t1.value()
31     p_t2 = t2.value()
32     p_t3 = t3.value()
33     p_t4 = t4.value()
34     for i in range(0, 8):
35         leds[i].value((cnt >> i) & 1) # i-ta LED dioda ima vrijednost i-tog bita cnt-
36         a
37     time.sleep(0.01) # Debouncing delay

```

6.3 Zadatak 5-2

```

1  import time
2  import machine
3  from machine import Pin
4
5  leds = []
6  for i in range(4, 12):
7      leds.append(Pin(i, Pin.OUT))
8
9  t1 = Pin(0, Pin.IN)
10 t2 = Pin(1, Pin.IN)
11 t3 = Pin(2, Pin.IN)
12 t4 = Pin(3, Pin.IN)
13
14 r = Pin(14, Pin.OUT)
15 g = Pin(12, Pin.OUT)
16 b = Pin(13, Pin.OUT)
17
18 cnt = 0x00
19
20 while True:
21     if t1.value():
22         cnt = (cnt + 1) & 0xFF # Uveca cnt ciklicno
23         g.value(0)
24         b.value(0)
25
26         r.value(1)
27         g.value(not (cnt & 1)) # Pali ako neparan cnt
28         b.value(cnt & 1) # Pali ako paran cnt
29
30     elif t2.value():
31         cnt = (cnt - 1) & 0xFF # Smanji cnt ciklicno
32         r.value(0)
33         g.value(0)
34
35         b.value(1)
36         g.value(not (cnt & 1)) # Pali ako neparan cnt
37         r.value(cnt & 1) # Pali ako paran cnt
38     elif t3.value():

```

```

39         cnt = 0x00
40         r.value(0)
41         b.value(0)
42         g.value(1)
43     elif t4.value():
44         cnt = 0xFF
45         r.value(0)
46         b.value(0)
47         g.value(1)
48
49     for i in range(0, 8):
50         leds[i].value((cnt >> i) & 1) # i-ta LED dioda ima vrijednost i-tog bita cnt-
51         a
52         time.sleep(1)
53         r.value(0)
54         b.value(0)
55         g.value(0)

```