

Univerzitet u Sarajevu
Elektrotehnički fakultet
Ugradbeni sistemi 2023/24.

Izvještaj za laboratorijsku vježbu 1
Razvojni sistem picoETF i MicroPython

Ime i prezime: **Kanita Kadušić**
Broj index-a: **19327**

Sarajevo, mart 2024.

Sadržaj

1	Pseudokod.....	1
1.1	Zadatak 4.....	1
1.2	Zadatak 5.....	1
1.3	Dodatni zadatak 1.....	2
1.4	Dodatni zadatak 2.....	2
2	Analiza programskog rješenja.....	3
2.1	Zadatak 4.....	3
2.2	Zadatak 5.....	3
2.3	Dodatni zadatak 1.....	4
2.4	Dodatni zadatak 2.....	4
3	Korišteni hardverski resursi.....	5
4	Zaključak.....	6
5	Prilog.....	7
5.1	Zadatak 4: Izvorni kôd.....	7
5.2	Zadatak 5: Izvorni kôd.....	8
5.3	Dodatni zadatak 1: Izvorni kôd.....	9
5.4	Dodatni zadatak 2: Izvorni kôd.....	10

1 Pseudokod

1.1 Zadatak 4

```
button4 ← input(3)  
led7 ← output(11)  
  
while true  
    value(led7) ← value(button4)
```

1.2 Zadatak 5

```
button1..4 ← input(0..3)  
led0..7 ← output(4..11)  
  
counter ← 0  
  
while true  
    if value(button1) = 1 and all(value(button2,3,4) = 0) and counter < 255  
        counter ← counter + 1  
    else if value(button2) = 1 and all(value(button1,3,4) = 0) and counter > 0  
        counter ← counter - 1  
    else if value(button3) = 1 and all(value(button1,2,4) = 0)  
        counter ← 0  
    else if value(button4) = 1 and all(value(button1,2,3) = 0)  
        counter ← 255  
  
    bits ← binary(counter)  
  
    for i in 0..7  
        value(ledi) ← bits[i]  
  
    while any(value(button1,2,3,4) = 1)  
        continue
```

1.3 Dodatni zadatak 1

$rgb_{1,2,3} \leftarrow output(14, 12, 13)$

$value(rgb_{1,2,3}) \leftarrow 0, 0, 0$

$value(rgb_{1,2,3}) \leftarrow 0, 0, 1$

$value(rgb_{1,2,3}) \leftarrow 0, 1, 0$

$value(rgb_{1,2,3}) \leftarrow 0, 1, 1$

$value(rgb_{1,2,3}) \leftarrow 1, 0, 0$

$value(rgb_{1,2,3}) \leftarrow 1, 0, 1$

$value(rgb_{1,2,3}) \leftarrow 1, 1, 0$

$value(rgb_{1,2,3}) \leftarrow 1, 1, 1$

1.4 Dodatni zadatak 2

$rgb_{1,2,3} \leftarrow output(14, 12, 13)$

$colors \leftarrow \{001, 010, 011, 100, 101, 110, 111\}$

$time, step \leftarrow 0.1$

while *true*

for *i* **in** 0..6

$value(rgb_{1,2,3}) \leftarrow colors[i][2, 1, 0]$

$pause(time)$

$time \leftarrow time + step$

if $time = 1.0$ **and** $step > 0$ **or** $time = 0.1$ **and** $step < 0$

$step \leftarrow -step$

2 Analiza programskog rješenja

2.1 Zadatak 4

[1] uvoz klase *Pin* iz modula *machine*

[2] uvoz funkcije *sleep* iz modula *time*

[4] kreiranje objekta za očitavanje i postavljanje vrijednosti digitalnog ulaza **GP3** na kojem se nalazi taster **T4**

[5] kreiranje objekta za očitavanje i postavljanje vrijednosti digitalnog izlaza **GP11** na kojem se nalazi dioda **LED7**

[7-8] neprekidno očitavanje stanja tastera i postavljanje stanja diode na očitanu vrijednost

[9] pauza u izvršavanju programa radi sprečavanja blokiranja online simulatora

2.2 Zadatak 5

[1-16] analogni komentari na osnovu Zadatka 4

[18] kreiranje niza objekata radi lakše manipulacije u nastavku

[20-23] funkcija koja prima string i cijeli broj i vraća proslijeđeni string sa dovoljnim brojem nula dodanim na početak stringa kako bi isti imao dužinu jednaku odgovarajućem parametru

[25-26] funkcija koja vraća logičku istinu ukoliko su vrijednosti svih digitalnih ulaza povezanih sa tasterima jednake odgovarajućim parametrima – u suprotnom vraća logičku neistinu

[28] deklaracija i inicijalizacija brojača

[31-38] očitavanje stanja tastera i izvršavanje pripadajućih akcija vodeći računa o neželjenom istovremenom pritiskanju tastera i opsegu vrijednosti brojača

[40-41] kreiranje objekta koji čuva vrijednost brojača u binarnom zapisu dužine 8 bita

[43-44] postavljanje stanja svih 8 dioda na vrijednosti odgovarajućih bita

[48-49] osiguranje okidanja samo na uzlaznu ivicu sata, tj. sprečavanje višestrukih akcija u logičkoj jedinici

[51] pauza u izvršavanju programa radi sprečavanja blokiranja online simulatora

2.3 Dodatni zadatak 1

[1-5] analogni komentari na osnovu Zadatka 4

[8-10] postavljanje vrijednosti digitalnih izlaza **GP14**, **GP12** i **GP13** na 0, 0 i 0 respektivno kako bi RGB dioda bila ugašena

[13-15] postavljanje vrijednosti digitalnih izlaza **GP14**, **GP12** i **GP13** na 0, 0 i 1 respektivno kako bi RGB dioda svijetlila plavom bojom

[18-45] analogan rezon prethodnom za postizanje redom boja: zelena, cijan, crvena, roza, žuta i bijela

2.4 Dodatni zadatak 2

[1-8] analogni komentari na osnovu Zadatka 4

[12] kreiranje niza sa svim mogućim kombinacijama emitovanih svjetlosti uz izostavljanje slučaja ugašene RGB diode

[17-18] deklaracija i inicijalizacija varijabli za dužinu pauze i veličinu koraka kojim se povećava odnosno smanjuje pauza

[22-26] prolazak kroz sve moguće kombinacije emitovanih svjetlosti i dodjela odgovarajućih vrijednosti digitalnim izlazima **GP12-14**

[30] pauza u izvršavanju programa za postizanje funkcionalnosti zahtijevane u zadatku

[35] odgovarajuće povećanje odnosno smanjenje dužine pauze u ovisnosti od predznaka vrijednosti koraka

[40-41] promjena predznaka vrijednosti koraka u ovisnosti od toga da li se dostigla dužina pauze u izvršavanju programa od 0.1 sekunde ili 1.0 sekunde

3 Korišteni hardverski resursi

	<i>Komponenta</i>	<i>Opis</i>	<i>Količina</i>
1	RP2040	mikrokontroler	1
2	taster	digitalni ulaz	4
3	LED	digitalni izlaz	8
4	RGB LED	digitalni izlaz	1
5	USB A – USB Micro	napajanje i komunikacija	1

4 **Zaključak**

Osim blage nesigurnosti u početku zbog rada s novim sistemima, izrada laboratorijske vježbe je protekla bez problema. Nešto više vremena je provedeno u istraživanju funkcija za rad sa stringovima dostupnim u MicroPython-u.

Zadatak 5 je izuzeo nešto vremena kako bi se osmislila realizacija u kojoj dugi pritisak tastera ne uzrokuje višestruke akcije.

Dodatni zadatak 2 je zahtijevao razmišljanje u cilju postizanja što efikasnije implementacije uz što manje zauzeće memorije, pritom vodeći računa o reprezentaciji realnih brojeva u računarskoj memoriji.

S obzirom da se radi o prvoj laboratorijskoj vježbi, kompletan njen sadržaj čini novostečeno znanje.

5 Prilog

5.1 Zadatak 4: Izvorni kôd

```
01: from machine import Pin
02: from time import sleep
03:
04: T4 = Pin(3, Pin.IN)
05: LED7 = Pin(11, Pin.OUT)
06:
07: while True:
08:     LED7.value(T4.value())
09:     sleep(0.2)
```

5.2 Zadatak 5: Izvorni kôd

```
01: from machine import Pin
02: from time import sleep
03:
04: T1 = Pin(0, Pin.IN)
05: T2 = Pin(1, Pin.IN)
06: T3 = Pin(2, Pin.IN)
07: T4 = Pin(3, Pin.IN)
08:
09: LED0 = Pin(4, Pin.OUT)
10: LED1 = Pin(5, Pin.OUT)
11: LED2 = Pin(6, Pin.OUT)
12: LED3 = Pin(7, Pin.OUT)
13: LED4 = Pin(8, Pin.OUT)
14: LED5 = Pin(9, Pin.OUT)
15: LED6 = Pin(10, Pin.OUT)
16: LED7 = Pin(11, Pin.OUT)
17:
18: LED = [LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7]
19:
20: def addZeros(string, length):
21:     while len(string) < length:
22:         string = "0" + string
23:     return string
24:
25: def buttonValues(a, b, c, d):
26:     return T1.value() == a and T2.value() == b and T3.value() == c and T4.value() == d
27:
28: counter = 0
29:
30: while True:
31:     if buttonValues(1, 0, 0, 0) and counter < 255:
32:         counter = counter + 1
33:     elif buttonValues(0, 1, 0, 0) and counter > 0:
34:         counter = counter - 1
35:     elif buttonValues(0, 0, 1, 0):
36:         counter = 0
37:     elif buttonValues(0, 0, 0, 1):
38:         counter = 255
39:
40:     binary = bin(counter)
41:     bits = addZeros(binary[2:], 8)
42:
43:     for i in range(0, 8):
44:         LED[i].value(int(bits[i]))
45:
46:     # kako bi se osiguralo okidanje samo na uzlaznu ivicu sata
47:     # tj. sprijecile visestruke akcije u logickoj jedinici
48:     while T1.value() or T2.value() or T3.value() or T4.value():
49:         pass
50:
51:     sleep(0.2)
```

5.3 Dodatni zadatak 1: Izvorni kôd

```
01: from machine import Pin
02:
03: R = Pin(14, Pin.OUT)
04: G = Pin(12, Pin.OUT)
05: B = Pin(13, Pin.OUT)
06:
07: # (R, G, B) = (0, 0, 0) => RGB dioda ugasena
08: R.off()
09: G.off()
10: B.off()
11:
12: # (R, G, B) = (0, 0, 1) => PLAVA
13: R.off()
14: G.off()
15: B.on()
16:
17: # (R, G, B) = (0, 1, 0) => ZELENA
18: R.off()
19: G.on()
20: B.off()
21:
22: # (R, G, B) = (0, 1, 1) => CIJAN
23: R.off()
24: G.on()
25: B.on()
26:
27: # (R, G, B) = (1, 0, 0) => CRVENA
28: R.on()
29: G.off()
30: B.off()
31:
32: # (R, G, B) = (1, 0, 1) => ROZA
33: R.on()
34: G.off()
35: B.on()
36:
37: # (R, G, B) = (1, 1, 0) => ZUTA
38: R.on()
39: G.on()
40: B.off()
41:
42: # (R, G, B) = (1, 1, 1) => BIJELA
43: R.on()
44: G.on()
45: B.on()
```

5.4 Dodatni zadatak 2: Izvorni kôd

```
01: from machine import Pin
02: from time import sleep
03:
04: # kreiranje objekata R, G i B za postavljanje vrijednosti
05: # odgovarajucih digitalnih izlaza GP14, GP12 i GP13
06: R = Pin(14, Pin.OUT)
07: G = Pin(12, Pin.OUT)
08: B = Pin(13, Pin.OUT)
09:
10: # niz sa svim mogucim kombinacijama emitovanih svjetlosti
11: # u nizu je izostavljen slucaj ugasene RGB diode (tj. 000)
12: colors = ["001", "010", "011", "100", "101", "110", "111"]
13:
14: # segment [0.1, 1.0] je bolje posmatrati kao [1, 10]
15: # jer su na prvobitnom segmentu samo 0.5 i 1.0 tacno zapisani u racunarskoj memoriji
16: # a zelimo izbjeći gresku nastalu nagomilavanjem manjih gresaka
17: sleepTime = 1
18: step = 1
19:
20: while True:
21:     # prolazak kroz sve moguće kombinacije emitovanih svjetlosti
22:     for lights in colors:
23:         # postavljanje odgovarajucih vrijednosti digitalnih izlaza GP12-14
24:         R.value(int(lights[2]))
25:         G.value(int(lights[1]))
26:         B.value(int(lights[0]))
27:
28:         # postavljanje broja sekundi zaustavljanja
29:         # dijeljenje s 10 zbog inicijalnog prelaska na segment [1, 10]
30:         sleep(sleepTime / 10)
31:
32:     # povećanje/smanjenje "usporavanja" (sleep) u ovisnosti od varijable step
33:     # ako je step = 1, usporavanje se povećava do 10 (tj. do 1.0s)
34:     # ako je step = -1, usporavanje se smanjuje do 1 (tj. do 0.1s)
35:     sleepTime = sleepTime + step
36:
37:     # ako je vrijeme postalo 10 (tj. 1.0s) uzlaznom putanjom (tj. step = 1) ili
38:     # ako je vrijeme postalo 1 (tj. 0.1s) silaznom putanjom (tj. step = -1)
39:     # potrebno je promijeniti znak koraka (step)
40:     if (sleepTime == 10 and step > 0) or (sleepTime == 1 and step < 0):
41:         step = -1 * step
```