

Univerzitet u Sarajevu

Elektrotehnički fakultet

Ugradbeni sistemi 2023 / 24

Izvještaj za laboratorijsku vježbu br. 1

Razvojni sistem picoETF i MicroPython

Ime i prezime: **Ivona Jozić**

Broj indeks-a: **19357**

12. mart 2024.

Sadržaj

1 Pseudokod i / ili dijagram toka.....	3
1.1 Zadatak 4	3
1.2 Zadatak 5	3
1.3 Zadaci za dodatne bodove	3
1.3.1 Zadatak 2	3
2 Analiza programskog rješenja	4
1.3 Zadatak 1	4
1.4 Zadatak 2	4
1.5 Zadatak 3	4
1.6 Zadatak 4	4
1.7 Zadatak 5	4
2.6 Zadaci za dodatne bodove	5
2.6.1 Zadatak 1	5
2.6.2 Zadatak 2	5
3 Korišteni hardverski resursi.....	6
4 Zaključak	6
5 Prilog	6
5.1 Zadatak 4 / izvorni kod.....	6
5.2 Zadatak 5 / izvorni kod.....	6
5.3 Zadaci za dodatne bodove	7
5.3.1 Zadatak 2 / izvorni kod.....	7

1 Pseudokod i / ili dijagram toka

Kako su od postavljenih zadataka u sklopu Laboratorijske vježbe 1, samo 4. i 5. zadatak, te 2. zadatak u Zadatacima za dodatne bodove zahtijevali programsku implementaciju, to ćemo u nastavku predložiti pseudokodove za spomenute 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)
        postavi_brojac_na_nulu()
        blokiraj_tastere()

    if(taster4 aktiviran)
        postavi_brojac_na_255()
        blokiraj_tastere()
```

1.3 Zadaci za dodatne bodove

1.3.1 Zadatak 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

1.3 Zadatak 1

U prvom zadatku je bilo potrebno izvršiti konfiguraciju mikrokontrolera Raspberry Pi Pico W (RP2040). Za početak smo povezali mikrokontroler sa računarom, koristeći USB kabal, pri čemu je tokom povezivanja bilo neophodno držati pritisnut bijeli taster na mikrokontroleru. Nakon što je povezivanje završeno, na mass storage je pohranjen *flash_nuke.uf2* koji je obrisao sav pohranjeni sadržaj sa mikrokontrolera, a zatim je na mikrokontroler instalirana najnovija verzija MicroPython firmware za RP2040, čime je okončan postupak konfigureacije.

1.4 Zadatak 2

Cilj drugog zadatka je bio upoznavanje sa rasporedom pinova na mikrokontroleru, te razvojnim okruženjem Thonny koje se može koristiti u radu sa picoETF sistemom. Koristili smo REPL komandnu liniju kako bismo upravljali sa GPIO pinovima mikrokontrolera. U ovom zadatku su izvršavane naredbe za kontrolu fizičkog tastera na pinu GP0, a koje su trebale utvrditi logičko stanje tastera. Kreiran je objekat „t1“ preko kojeg smo dobivali informacije o logičkom stanju tastera.

1.5 Zadatak 3

Kako razvojni sistem picoETF pored tastera posjeduje i 8 LED dioda, cilj 3. zadatka je bio upoznavanje sa njihovim radom u kombinaciji sa već ranije korištenim tasterima. Dioda su povezane na pinove GP4 – GP11, te se one mogu paliti i gasiti kreiranjem odgovarajućih objekata i postavljanjem njihovih logičkih vrijednosti za svaki od pinova. U ovom konkretnom zadatku je bilo zahtijevano da se stanje dioda mijenja u odnosu na logičku vrijednost tastera T1, tj. da je dioda upaljena ukoliko je on pritisnut, odnosno ugašena ukoliko on nije pritisnut.

1.6 Zadatak 4

U četvrtom zadatku je bilo potrebno kreirati skriptu main.py, koja će kontrolisati stanje LED7 diode u zavisnosti od stanja tastera T4, tj. čiji programski kod treba dovesti do paljenja LED7 ukoliko je T4 pritisnut i njenog svijetljenja dokle god je taster pritisnut. Prvo smo inicijalizirali varijable koje odgovaraju modulima za pinove GP3 (T4) i GP10 (LED7), te zatim u while petlji postigli paljenje LED7 u zavisnosti od stanja T4.

1.7 Zadatak 5

U petom zadatku je bila zahtijevana implementacija 8-bitnog brojača uz pomoć LED dioda na mikrokontroleru. Tasteri T1 – T4 trebaju upravljati brojačem, na način da pritisak na taster T1 uvećava brojač za 1, pritisak na taster T2 umanjuje brojač za 1, dok pritisci na tastere T3 i T4 postavljaju brojač na 255 (sve LED diode upaljenje), odnosno na 0 (sve diode ugašene), respektivno. Implementirano rješenje se zasniva na organizaciji objekata koji odgovaraju pinovima za tastere i LED diode u liste, kako bi manipulacija bila lakša. Brojač je implementiran kao decimalna varijabla koja je u svojoj binarnoj reprezentaciji na 8 bita odgovara diodama LED0 – LED7. Shiftanjem brojača udesno, te vršenjem logičke operacije AND bit po bit, brojač se inkrementira. Cjelokupna implementacija ovog problema data je u odjeljku 5.2.

2.6 Zadaci za dodatne bodove

2.6.1 Zadatak 1

U ovom zadatku je bio cilj da se pomoću komadi u REPL promptu kontroliše rad RGB LED diode. Ona ima 3 različite boje čijim kombinovanjem možemo dobiti ukupno 7 različitih boja. Varijable su postavljene kao pojedinačne boje: red, green i blue, te se kombiniranim uključivanjem i isključivanjem dobijaju različite boje koje RGB LED može pružiti:

- Crvena + plava = ljubičasta

- Plava + zelena = tirkizna

- Crvena + zelena = žuta

- Crvena + zelena + plava = bijela.

Na ovaj način možemo dobiti svaku od boja, a kao primjer je dat način dobivanja bijele i ljubičaste boje:

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

Sada ukoliko želimo ljubičastu, dovoljno je samo ugaziti zelenu:

```
>>> green.value(0)
```

*Zadatak je implementiran i testiran u simulatoru.

2.6.2 Zadatak 2

U ovom zadatku je bilo potrebno implementirati program koji reguliše rad RGB LED diode, na način da se sve moguće kombinacije izmjenjuju u jednakom trajanju krenuvši od 0.1s do 1s sa korakom od 0.1s, a zatim se to trajanje treba smanjivati od 1s do 0.1s sa istim korakom. Zahtijrvi efekat se postiže implementacijom while(true) petlje koja se neprestano izvršava na kontroleru, te liste pinova koji odgovaraju R, G i B vrijednostima RGB LED diode. Ove varijable se u for – petlji mijenjaju u proizvoljnom redoslijedu, tako da dioda poprmi sve boje, a varijabilno trajanje svake iteracije niza boja je implementirano pomoću jedne dodatne varijable koja se kroz for – petlje kontinuirano 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 vanjska while(true) petlja ponavlja.

*Zadatak je implementiran i testiran u simulatoru.

3 Korišteni hardverski resursi

Za potrebe laboratorijske vježbe 1 korišten je razvojni sistem picoETF u sklopu kojeg su integrirani svi potrebni elementi.

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

4 Zaključak

Prilikom izvođenja Laboratorijske vježbe 1 nije bilo problema osim sporijeg rada zbog susreta sa novim razvojem okruženjem, kako picoETF-om, tako i MicroPythonom, s obzirom da do sada nismo imali priliku implementirati programska rješenja u programskom jeziku Python. Cilj vježbe je bio upoznavanje sa radom u laboratoriji i razvojnim sistemom picoETF, što je i postignuto.

5 Prilog

5.1 Zadatak 4 / izvorni kod

```
1. from machine import Pin
2.
3. taster4 = Pin(3, Pin.IN)
4. led7 = Pin(10, Pin.OUT)
5.
6. while True:
7.     led7.value(taster4.value())
```

5.2 Zadatak 5 / izvorni kod

```
1. import time
2. from machine import Pin
3. time.sleep(0.1) #Cekanje na pripravnost USB
4.
5. leds = [Pin(i, Pin.OUT) for i in range(4, 12)] #deklracija LED dioda
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 dioda
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(): #onemogućiti 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(): #onemogućiti unos pri držanju tastera
33.             time.sleep(0.1)
34.
35.     elif tasteri[2].value():
36.         brojac = 0
37.         prikazi_brojac(brojac)
38.         while tasteri[2].value(): #onemogućiti unos pri držanju tastera
39.             time.sleep(0.1)
40.
41.     elif tasteri[3].value():
42.         brojac = 255
43.         prikazi_brojac(brojac)
44.         while tasteri[3].value(): #onemogućiti unos pri držanju tastera
45.             time.sleep(0.1)

```

5.3 Zadaci za dodatne bodove

5.3.1 Zadatak 2 / izvorni kod

```

1. import time
2. from machine import Pin
3. time.sleep(0.1)           #Cekanje na pripravnost USB
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): #kombiniraju se vrijednosti iz nizova pins i 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.

```