

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

## IZVJEŠTAJ ZA LABORATORIJSKU VJEŽBU BR. 3

VIŠEBITNI DIGITALNI ULAZI I IZLAZI

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

7.4.2025.

# Sadržaj

|          |                               |          |
|----------|-------------------------------|----------|
| <b>1</b> | <b>Uvod</b>                   | <b>1</b> |
| <b>2</b> | <b>Postavka zadatka</b>       | <b>1</b> |
| 2.1      | Zadatak 1 . . . . .           | 1        |
| 2.2      | Zadatak 2 . . . . .           | 2        |
| 2.3      | Zadatak 3 . . . . .           | 3        |
| <b>3</b> | <b>Implementacija</b>         | <b>5</b> |
| 3.1      | Zadatak 1 . . . . .           | 5        |
| 3.2      | Zadatak 2 . . . . .           | 5        |
| 3.3      | Zadatak 3 . . . . .           | 6        |
| <b>4</b> | <b>Testiranje i rezultati</b> | <b>7</b> |
| 4.1      | Zadatak 1 . . . . .           | 7        |
| 4.2      | Zadatak 2 . . . . .           | 7        |
| 4.3      | Zadatak 3 . . . . .           | 7        |
| <b>5</b> | <b>Zaključak</b>              | <b>8</b> |
| <b>6</b> | <b>Prilozi</b>                | <b>9</b> |
| 6.1      | Zadatak 1 . . . . .           | 9        |
| 6.2      | Zadatak 2 . . . . .           | 10       |
| 6.3      | Zadatak 3 . . . . .           | 10       |

# 1 Uvod

Cilj ove laboratorijske vježbe je upoznavanje sa višebitnim digitalnim ulazima i izlazima, načinom korištenja matrične tastature i načinom korištenja višecifrenog 7-segmentnog displeja.

Korišteni hardver i softver:

| Komponenta                      | Opis   |
|---------------------------------|--|
| picoETF                         | Razvojni sistem baziran na RP2040                |
| Thonny IDE                      | Razvojno okruženje za MicroPython                |
| MicroPython                     | Programski jezik prilagođen mikrokontrolerima    |
| LPC1114ETF                      | Razvojni sistem baziran na LPC1114FN28           |
| Arm Keil IDE                    | Razvojno okruženje za mbed                       |
| mbed.h                          | Biblioteka za rad s mikrokontrolerima u C jeziku |
| Višecifreni 7-segmentni displej | Displej sa četiri 7-segmentne cifre              |
| Matrična tastatura              | Tastatura  |

## 2 Postavka zadatka

### 2.1 Zadatak 1

U ovom zadatku potrebno je povezati matričnu tastaturu s razvojnim sistemom LPC1114ETF prema šemi, tako da su prva tri pina u lijevom donjem uglu prazna, a peti pin se ne koristi. Manji, žuti konektor treba biti na desnoj strani, a veći ljubičasti konektor na lijevoj strani, s oznakama u boji okrenutim prema vani.

Funkcionalnost koja se treba implementirati je da sve LED diode sistema budu uključene. Kada se pritisne i drži pritisnuta cifra na tastaturi (1-8), potrebno je da se od 8 LED dioda razvojnog sistema isključi ona čiji je odgovarajući broj pritisnut. Pritiskom na A isključene diode se uključuju, nakon čega je ponovo moguće isključiti diode prema prethodno opisanoj proceduri.

---

```
1: Inicijalizacija:
2: ledE ← 0
3: leds ← BusOut(LED0..LED7)
4: col ← BusIn(dp9, dp10, dp11, dp13)
5: row ← BusOut(dp16, dp15, dp17, dp18)
6:
7: while tačno do
8:   for i ← 0 do 3 do
9:     row[i] ← 1
10:
11:   if row[0] = 1 ∧ col[0] = 1 then
12:     leds ← 255; leds[0] ← 0
13:   else if row[0] = 1 ∧ col[1] = 1 then
14:     leds ← 255; leds[1] ← 0
```

▷ Deaktiviraj aktivni LED indikator  
▷ 8-bitni izlaz za LED diode  
▷ Ulazi za stupce tipkovnice  
▷ Izlazi za redove tipkovnice  
▷ Prođi kroz sve 4 reda  
▷ Aktiviraj trenutni red  
▷ Upali sve LED osim LED0  
▷ Upali sve LED osim LED1

|     |   |                            |
|-----|---|----------------------------|
| 15: | <b>else if</b> row[0] = 1 $\wedge$ col[2] = 1 <b>then</b> |                            |
| 16: | leds $\leftarrow$ 255; leds[2] $\leftarrow$ 0             | ▷ Upali sve LED osim LED2  |
| 17: | <b>else if</b> row[1] = 1 $\wedge$ col[0] = 1 <b>then</b> |                            |
| 18: | leds $\leftarrow$ 255; leds[3] $\leftarrow$ 0             | ▷ Upali sve LED osim LED3  |
| 19: | <b>else if</b> row[1] = 1 $\wedge$ col[1] = 1 <b>then</b> |                            |
| 20: | leds $\leftarrow$ 255; leds[4] $\leftarrow$ 0             | ▷ Upali sve LED osim LED4  |
| 21: | <b>else if</b> row[1] = 1 $\wedge$ col[2] = 1 <b>then</b> |                            |
| 22: | leds $\leftarrow$ 255; leds[5] $\leftarrow$ 0             | ▷ Upali sve LED osim LED5  |
| 23: | <b>else if</b> row[2] = 1 $\wedge$ col[0] = 1 <b>then</b> |                            |
| 24: | leds $\leftarrow$ 255; leds[6] $\leftarrow$ 0             | ▷ Upali sve LED osim LED6  |
| 25: | <b>else if</b> row[2] = 1 $\wedge$ col[1] = 1 <b>then</b> |                            |
| 26: | leds $\leftarrow$ 255; leds[7] $\leftarrow$ 0             | ▷ Upali sve LED osim LED7  |
| 27: | <b>else if</b> row[0] = 1 $\wedge$ col[3] = 1 <b>then</b> |                            |
| 28: | leds $\leftarrow$ 255                                     | ▷ Upali sve LED diode      |
| 29: | <b>else if</b> row[3] = 1 $\wedge$ col[1] = 1 <b>then</b> |                            |
| 30: | leds $\leftarrow$ 0                                       | ▷ Ugasi sve LED diode      |
| 31: | <b>end if</b>   |                            |
| 32: |   |                            |
| 33: | row[z] $\leftarrow$ 0                                     | ▷ Deaktiviraj trenutni red |
| 34: | <b>end for</b>  |                            |
| 35: |   |                            |
| 36: | Čekaj 100 ms  | ▷ Pauza između skeniranja  |
| 37: | <b>end while</b>  |                            |

---

## 2.2 Zadatak 2

Za picoETF razvojni sistem, potrebno je napisati testni program za provjeru ispravnosti povezivanja i rada 7-segmentnog displeja sa zajedničkom anodom, tako što će se segmenti i cifre uključivati sistematski. Testni program treba uključivati:

- Na svakoj cifri (D1, D2, D3, D4) redom uključivati segmente A, B, C, D, E, F, G i DP. Svaki segment uključen po 0.2s, nakon čega se isključuje. Kada se završi provjera na jednoj cifri prelazi se na sljedeću.
- Kada se završi prethodna provjera, na svim ciframa se prikazuje broj 8, i to sljedećim redoslijedom: prvo je prikaz na cifri D1 (prikaz traje 1s, nakon čega se isključuje), zatim se uključuje prikaz na D2, pa na D3, i konačno na D4.
- Prethodno opisani proces se ciklično ponavlja.
- Također, potrebno je testirati ispis cifre 8 s brzinom izmjene od 20ms, umjesto prvobitnih 1s, te uočiti eventualne razlike.

---

### 1: Inicijalizacija:

|  |                     |
|--|---------------------|
| 2: segments $\leftarrow$ [Pin(8)..Pin(15)] | ▷ Segmenti A-G i DP |
| 3: digits $\leftarrow$ [Pin(4)..Pin(7)]    | ▷ Cifre D1-D4       |

```

4:
5: for seg ∈ segments do
6:     seg.value(1)                                ▷ Inicijalno ugasi sve segmente
7: end for
8:
9: for dig ∈ digits do
10:    dig.value(1)                                ▷ Inicijalno deaktiviraj sve cifre
11: end for
12:
13: while tačno do                                ▷ Prva petlja - animacija po segmentima
14:    for digit ∈ digits do
15:        digit.value(0)                            ▷ Aktiviraj trenutnu cifru
16:
17:        for seg ∈ segments do
18:            seg.value(0)                            ▷ Upali segment
19:            Čekaj 0.2 s
20:            seg.value(1)                            ▷ Ugasi segment
21:        end for
22:
23:        digit.value(1)                            ▷ Deaktiviraj cifru
24:    end for
25:                                ▷ Druga petlja - paljenje svih segmenata po ciframa
26:    for digit ∈ digits do
27:        digit.value(0)                            ▷ Aktiviraj trenutnu cifru
28:
29:        for i ← 0 do 6 do                            ▷ Prođi kroz segmente A-G
30:            segments[i].value(0)                    ▷ Upali segment
31:        end for
32:
33:        Čekaj 1 s
34:
35:        for i ← 0 do 6 do
36:            segments[i].value(1)                    ▷ Ugasi segment
37:        end for
38:
39:        digit.value(1)                            ▷ Deaktiviraj cifru
40:    end for
41: end while

```

---

## 2.3 Zadatak 3

Za razvojni sistem picoETF potrebno je spojiti četverocifreni 7-segmentni displej prema priloženoj šemi i napisati program koji implementira brojač na tasterima 1 i 2. Pritiskom na taster 1 prikaz na 7-segmentnom displeju se treba uvećavati (npr. 0000, 0001, 0002, itd.), dok se pritiskom na taster 2 prikaz smanjuje. Pritisak na taster 3 resetuje brojač na 0000, a pritisak na taster 4 pokreće/zaustavlja automatsko brojanje (promjena brojača svaku sekundu).

Brojanje treba realizovati na uzlaznu ivicu, odnosno ako se stanje tastera promijeni iz otpušten u pritisnut (logička nula u logičku jedinicu). To znači da ukoliko korisnik zadrži taster duže vrijeme, brojanje ne bi smjelo da se nastavi sve dok se taster ne otpusti, pa ponovo pritisne. Odnosno, broje se samo "klikovi", a ne dužina trajanja pritiska na taster.

---

```

1: Inicijalizacija:
2: button1..button4 ← Pin(0..3, IN)                                ▷ Tasteri na pinovima 0-3
3: nums ← matrica kodova za cifre 0-9                             ▷ 8-bitni zapis za svaku cifru
4: digits ← [Pin(7), Pin(6), Pin(5), Pin(4)]                      ▷ Cifre D1-D4
5: segments ← [Pin(8)..Pin(15)]                                   ▷ Segmenti A-G i DP
6: prev ← [False, False, False, False]                           ▷ Prethodna stanja tastera
7: auto ← False                                                  ▷ Automatski inkrement mod
8: cnt ← 0                                                        ▷ Brojač
9: last_auto_increment ← time.time()                             ▷ Vrijeme posljednjeg automatskog povećanja
10:
11: function POSTAVISEGMENT(cifra)
12:   for  $i \leftarrow 0$  do 7 do
13:     segments[i].value(nums[cifra][i])
14:   end for
15: end function
16:
17: function ISPISIBROJ(tmp)
18:   for  $i \leftarrow 0$  do 3 do
19:     postaviSegment(tmp%10)
20:     digits[i].value(0)
21:     Čekaj 0.002 s
22:     digits[i].value(1)
23:     tmp ← tmp//10
24:   end for
25: end function
26:
27: while tačno do
28:   ispisiBroj(cnt)
29:   current_time ← time.time()
30:
31:   if button1.value()  $\wedge$   $\neg$ prev[0] then
32:     cnt ← cnt + 1
33:   else if button2.value()  $\wedge$   $\neg$ prev[1] then
34:     cnt ← cnt - 1
35:     if cnt < 0 then
36:       cnt ← 0
37:     end if
38:   else if button3.value()  $\wedge$   $\neg$ prev[2] then
39:     cnt ← 0
40:   else if button4.value()  $\wedge$   $\neg$ prev[3] then
41:     auto ←  $\neg$ auto
42:   end if
43:

```

```

44:   prev ← [button1.value(), ..., button4.value()]
45:
46:   if auto ∧ current_time − last_auto_increment ≥ 1.0 then
47:       cnt ← cnt + 1
48:       last_auto_increment ← current_time
49:   end if
50: end while

```

---

## 3 Implementacija

### 3.1 Zadatak 1

U ovom zadatku implementiran je kod za LPC1114ETF sistem koji omogućava kontrolu LED dioda pomoću matrične tastature. Glavni koraci implementacije su:

#### 1. Inicijalizacija pinova:

- Za LED diode korišten je BusOut objekat koji omogućava kontrolu više izlaznih pinova kao grupu.
- Za kolone tastature korišten je BusIn objekat.
- Za redove tastature korišten je BusOut objekat.

#### 2. Skeniranje tastature:

- U glavnoj petlji, implementirano je skeniranje matrične tastature postavljanjem jednog po jednog reda u stanje 1 i provjerom stanja kolona.
- Za svaku kombinaciju reda i kolone koja predstavlja određeni taster, implementirana je odgovarajuća akcija.

#### 3. Implementacija funkcionalnosti:

- Kada je detektovan pritisak na tastere 1-8, odgovarajuća LED dioda se isključuje, a ostale ostaju uključene.
- Kada je detektovan pritisak na taster A, sve LED diode se uključuju.
- Dodatno je implementirano da se pritiskom na taster 0 isključe sve LED diode.

### 3.2 Zadatak 2

U ovom zadatku implementiran je testni program za provjeru ispravnosti povezivanja i rada 7-segmentnog displeja na picoETF razvojnom sistemu. Glavni koraci implementacije su:

#### 1. Inicijalizacija pinova:

- Definirani su pinovi za segmente (A-G, DP) i cifre (D1-D4) 7-segmentnog displeja.
- Svi segmenti su inicijalno postavljeni na visoko stanje (isključeni, jer displej koristi zajedničku anodu).
- Sve cifre su inicijalno postavljene na nisko stanje (isključene).

## 2. Testiranje segmenata:

- Implementirana je beskonačna petlja koja:
  - Za svaku cifru pojedinačno aktivira svaki segment na 0.2s i zatim ga deaktivira.
  - Nakon testiranja svih segmenata na jednoj cifri, prelazi na sljedeću.

## 3. Testiranje prikaza broja 8:

- Nakon testiranja segmenata, implementiran je test prikaza broja 8 na svakoj cifri pojedinačno, s trajanjem od 1s po cifri.
- Isti test je implementiran i s brzinom izmjene od 20ms umjesto 1s, radi poređenja.

## 3.3 Zadatak 3

U ovom zadatku implementiran je brojač koji se kontroliše pomoću tastera, a prikazuje na 7-segmentnom displeju. Glavni koraci implementacije su:

### 1. Inicijalizacija:

- Definisani su pinovi za tastere, segmente displeja i cifre displeja.
- Kreirana je lista prikaza cifara od 0 do 9 na 7-segmentnom displeju.
- Inicijalizirane su varijable za praćenje stanja tastera, vrijednosti brojača i automatskog režima rada.

### 2. Implementacija funkcija za prikaz:

- Implementirana je funkcija `postaviSegment` koja postavlja odgovarajuće segmente za prikaz određene cifre.
- Implementirana je funkcija `ispisiBroj` koja prikazuje četverocifreni broj na displeju, koristeći tehniku multipleksiranja.

### 3. Glavna petlja:

- U glavnoj petlji programa implementirana je logika brojača:
  - Taster 1: Povećava vrijednost brojača za 1.
  - Taster 2: Smanjuje vrijednost brojača za 1 (ne ispod 0).
  - Taster 3: Resetuje brojač na 0.
  - Taster 4: Uključuje/isključuje automatski režim brojanja.
- Implementirana je detekcija promjene stanja tastera (edge detection) kako bi brojač reagovao samo na "klik", a ne na kontinuirani pritisak.
- U automatskom režimu, brojač se povećava za 1 svake sekunde.



## 4 Testiranje i rezultati

### 4.1 Zadatak 1

Testiranje matične tastature i LED dioda na LPC1114ETF sistemu dalo je sljedeće rezultate:

- Pritiskom na tastere 1-8 odgovarajuća LED dioda se isključuje, dok ostale ostaju uključene.
- Pritiskom na taster A sve LED diode se ponovno uključuju.
- Pritiskom na taster 0 sve LED diode se isključuju.
- Skeniranje tastature radi pouzdano, s dovoljno brzim odzivom na pritisak tastera.

### 4.2 Zadatak 2

Testiranje 7-segmentnog displeja na picoETF sistemu pokazalo je:

- Segmenti A-G i DP se pravilno aktiviraju i deaktiviraju na svakoj cifri.
- Prikaz broja 8 na svakoj cifri pojedinačno radi ispravno s trajanjem od 1s.
- Prilikom testiranja s brzinom izmjene od 20ms umjesto 1s, primijećena je perzistencija vida - displej izgleda kao da svi segmenti svijetle istovremeno, što potvrđuje da je multipleksiranje uspješno implementirano.

### 4.3 Zadatak 3

Testiranje brojača na picoETF sistemu dalo je sljedeće rezultate:

- Taster 1 uspješno povećava vrijednost brojača.
- Taster 2 uspješno smanjuje vrijednost brojača, s ograničenjem da ne ide ispod 0.
- Taster 3 uspješno resetuje brojač na 0.
- Taster 4 uspješno uključuje i isključuje automatski režim brojanja.
- Detekcija promjene stanja tastera radi ispravno - brojač reaguje samo na "klik", a ne na kontinuirani pritisak.
- Automatsko brojanje u intervalima od 1 sekunde radi pouzdano.
- Multipleksiranje 7-segmentnog displeja radi ispravno, s dovoljno brzim osvježavanjem da se postiže iluzija istovremenog prikaza svih cifara.

## 5 Zaključak

Kroz ovu laboratorijsku vježbu uspješno smo implementirali nekoliko ključnih koncepata rada s višebitnim digitalnim ulazima i izlazima:

1. **Matrična tastatura:** Naučili smo principe rada s matričnom tastaturom, uključujući tehniku skeniranja redova i kolona za detekciju pritisnutih tastera.
2. **Višecifreni 7-segmentni displej:** Savladali smo tehniku multipleksiranja za kontrolu višecifrenog 7-segmentnog displeja, što omogućava prikaz različitih cifara uz ograničen broj pinova mikrokontrolera.
3. **Detekcija promjene stanja:** Implementirali smo detekciju promjene stanja tastera (edge detection), što je ključno za mnoge praktične primjene.

Ova laboratorijska vježba pružila nam je praktično iskustvo u radu s komponentama koje se često koriste u ugradbenim sistemima. Naučene tehnike imaju široku primjenu u različitim projektima, od jednostavnih korisničkih interfejsa do složenijih sistema za kontrolu i praćenje. Implementacija za LPC1114ETF sistem pokazala je kako se može efikasno koristiti matrična tastatura za kontrolu LED dioda, dok su implementacije za picoETF sistem demonstrirale fleksibilnost MicroPython-a za rad s različitim periferijama.

Za buduća poboljšanja sistema, mogli bismo razmotriti optimizaciju koda za smanjenje potrošnje energije prilikom skeniranja tastature i osvježavanja displeja. Također bi bilo korisno dodati funkcije poput debouncing-a za tastere i implementirati tranzicijske efekte na displeju.

## 6 Prilozi

### 6.1 Zadatak 1

```
#include "mbed.h"
#include "lpc1114etf.h"

DigitalOut ledE(LED_ACT);
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
BusIn col(dp9, dp10, dp11, dp13);
BusOut row(dp16, dp15, dp17, dp18);

int main(){
    ledE=0;
    while(1){
        for(int i = 0; i < 4; i++){
            row[i] = 1;
            if(row[0]==1 && col[0]==1){
                leds=255;
                leds[0]=0;
            } else if(row[0]==1 && col[1]==1){
                leds=255;
                leds[1]=0;
            } else if(row[0]==1 && col[2]==1){
                leds=255;
                leds[2]=0;
            } else if(row[1]==1 && col[0]==1){
                leds=255;
                leds[3]=0;
            } else if(row[1]==1 && col[1]==1){
                leds=255;
                leds[4]=0;
            } else if(row[1]==1 && col[2]==1){
                leds=255;
                leds[5]=0;
            } else if(row[2]==1 && col[0]==1){
                leds=255;
                leds[6]=0;
            } else if(row[2]==1 && col[1]==1){
                leds=255;
                leds[7]=0;
            } else if(row[0]==1 && col[3]==1){
                leds = 255;
            } else if(row[3]==1 && col[1]==1){
                leds=0;
            }
            row[i]=0;
        }
        ThisThread::sleep_for(100ms);
    }
}
```

## 6.2 Zadatak 2

```
1 from machine import Pin
2 import time
3
4 segments = [
5     Pin(8, Pin.OUT), # A
6     Pin(9, Pin.OUT), # B
7     Pin(10, Pin.OUT), # C
8     Pin(11, Pin.OUT), # D
9     Pin(12, Pin.OUT), # E
10    Pin(13, Pin.OUT), # F
11    Pin(14, Pin.OUT), # G
12    Pin(15, Pin.OUT) # DP
13 ]
14 digits=[
15     Pin(4, Pin.OUT), # D1
16     Pin(5, Pin.OUT), # D2
17     Pin(6, Pin.OUT), # D3
18     Pin(7, Pin.OUT) # D4
19 ]
20
21 for seg in segments:
22     seg.value(1)
23 for dig in digits:
24     dig.value(1)
25
26 while True:
27     for digit in digits:
28         digit.value(0)
29         for seg in segments:
30             seg.value(0)
31             time.sleep(0.2)
32             seg.value(1)
33         digit.value(1)
34
35     for digit in digits:
36         digit.value(0)
37         for i in range(7):
38             segments[i].value(0)
39         time.sleep(1)
40         for i in range(7):
41             segments[i].value(1)
42         digit.value(1)
```

## 6.3 Zadatak 3

```
1 import time
2 from machine import Pin
3
4 # Definicija tastera
5 button1 = Pin(0, Pin.IN)
6 button2 = Pin(1, Pin.IN)
7 button3 = Pin(2, Pin.IN)
8 button4 = Pin(3, Pin.IN)
9
10 nums = [
11     [0, 0, 0, 0, 0, 0, 1, 1], # 0
```

```

12     [1, 0, 0, 1, 1, 1, 1, 1], # 1
13     [0, 0, 1, 0, 0, 1, 0, 1], # 2
14     [0, 0, 0, 0, 1, 1, 0, 1], # 3
15     [1, 0, 0, 1, 1, 0, 0, 1], # 4
16     [0, 1, 0, 0, 1, 0, 0, 1], # 5
17     [0, 1, 0, 0, 0, 0, 0, 1], # 6
18     [0, 0, 0, 1, 1, 1, 1, 1], # 7
19     [0, 0, 0, 0, 0, 0, 0, 1], # 8
20     [0, 0, 0, 1, 1, 0, 0, 1], # 9
21 ]
22 digits = [Pin(7, Pin.OUT), Pin(6, Pin.OUT), Pin(5, Pin.OUT), Pin(4, Pin.OUT)]
23 segments = [Pin(i, Pin.OUT) for i in range(8, 16)]
24
25 def postaviSegment(cifra):
26     for i in range(8):
27         segments[i].value(nums[cifra][i])
28
29 def ispisiBroj(tmp):
30     for i in range(4):
31         postaviSegment(tmp % 10)
32         digits[i].value(0)
33         time.sleep(0.002)
34         digits[i].value(1)
35         tmp //= 10
36
37 for seg in segments:
38     seg.value(1)
39 for dig in digits:
40     dig.value(1)
41
42 prev = [False, False, False, False]
43 auto = False
44 last_auto_increment = time.time()
45 cnt = 0
46
47 while True:
48     ispisiBroj(cnt)
49     current_time = time.time()
50     if button1.value() and not prev[0]:
51         cnt += 1
52     elif button2.value() and not prev[1]:
53         cnt -= 1
54         if cnt < 0:
55             cnt = 0
56     elif button3.value() and not prev[2]:
57         cnt = 0
58     elif button4.value() and not prev[3]:
59         auto = not auto
60
61 prev = [button1.value(), button2.value(), button3.value(), button4.value()]
62
63 if auto:
64     if current_time - last_auto_increment >= 1.0:
65         cnt += 1
66         last_auto_increment = current_time

```