

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

### **Izvještaj za laboratorijsku vježbu br. 3**

Višebitni digitalni ulazi i izlazi

Ime i prezime: **Mirza Mahmutović**

Broj index-a: **19320**

26. mart 2024. godine

# Sadržaj

1	Pseudokod i/ili dijagram toka .....	1
1.1	Zadatak 1 .....	1
1.2	Zadatak 2 .....	2
1.3	Zadatak 3 – izbor 1 .....	4
2	Analiza programskog rješenja.....	6
2.1	Zadatak 1 .....	6
2.2	Zadatak 2 .....	6
2.3	Zadatak 3 – izbor 1 .....	7
3	Korišteni hardverski resursi .....	8
4	Zaključak.....	8
5	Prilog.....	9
5.1	Zadatak 1/izvorni kod .....	9
5.2	Zadatak 2/izvorni kod .....	10
5.3	Zadatak 3 – izbor 1/izvorni kod .....	12

# 1 Pseudokod i/ili dijagram toka

## 1.1 Zadatak 1

```
char keys[16] = {'1','2', '3','A',  
                '4','5', '6','B',  
                '7','8', '9','C',  
                '*', '0', '#','D'}  
  
char keyInput()  
    for (i = 0; i < 4; i++) do  
        rows[i].on()  
        int col = cols  
        if (col == 1) then  
            return keys[i * 4 + 0]  
        else_if (col == 2) then  
            return keys[i * 4 + 1]  
        else_if (col == 3) then  
            return keys[i * 4 + 2]  
        else_if (col == 4) then  
            return keys[i * 4 + 3]  
        end_if  
    end_for  
  
while (true) do  
    char n = keyInput()  
    if (n.isBetween(0,9)) then  
        leds[n.toNumber()] = 1  
        while (keyIsPressed()) do  
            sleep()  
        end_while  
    else_if (n == 'C') then  
        leds = 0  
        while (keyIsPressed()) do  
            sleep()  
        end_while  
    end_if
```

**end\_while**

## 1.2 Zadatak 2

```
cifre = {
    0: 0b1000000,
    1: 0b1111001,
    2: 0b0100100,
    3: 0b0110000,
    4: 0b0011001,
    5: 0b0010010,
    6: 0b0000010,
    7: 0b1111000,
    8: 0b0000000,
    9: 0b0010000,
    10: 0b1111111
}

mjesto = [Pin(4, Pin.OUT), Pin(5, Pin.OUT), Pin(6, Pin.OUT), Pin(7, Pin.OUT)]
segmenti = [Pin(8, Pin.OUT), Pin(9, Pin.OUT), Pin(10, Pin.OUT), Pin(11, Pin.OUT), Pin(12, Pin.OUT), Pin(13, Pin.OUT), Pin(14, Pin.OUT)]
tasteri = [Pin(0, Pin.IN), Pin(1, Pin.IN), Pin(2, Pin.IN), Pin(3, Pin.IN)]

prikazi_cifru(cifra)
    binarno = cifre[cifra]
    for (i = 0; i < 7; i++) do
        segmenti[i].value(binarno & 1)
        binarno = binarno >> 1
    end_for

prikazi_poziciju(pozicija)
    for (i = 0; i < 4; i++) do
        mjesto[i].value(0)
    end_for
    mjesto[pozicija].value(1)

prikazi_broj(broj)
    n = 3
    cifra = daj_cifru(broj, n)
    while (n >= 0) do
```

```

        prikazi_cifru(10)
        prikazi_poziciju(3-n)
        prikazi_cifru(cifra)
        n -= 1
        cifra = daj_cifru(broj, n)
        sleep()

    end_while
daj_cifru(broj, n)
    return broj // 10 ** n % 10
auto = false
broj = 0
while (true) do
    leds = counter
    if(tasteri[0].pressed()) then
        broj += 1
        while (tasteri[0].pressed()) then
            sleep()
        end_while
    else_if (tasteri[1].pressed()) then
        broj -= 1
        while (tasteri[1].pressed()) then
            sleep()
        end_while
    else_if (tasteri[2].pressed()) then
        broj = 0
        while (tasteri[2].pressed()) then
            sleep()
        end_while
    else_if (tasteri[3].pressed()) then
        auto = !auto
        count = 0
        while (tasteri[3].pressed()) then
            sleep()
        end_while
    end_if
    if (auto) then

```

```

        if (count == 9) then
            broj += 1
            count = 0
        end_if
        count += 1
    end_if
    if (broj > 9999) then
        broj = 0
    end_if
    if (broj < 0) then
        broj = 9999
    end_if
    prikazi_broj(broj)
    sleep()
end_while

1.3 Zadatak 3 – izbor 1

keys[16] = {'1','2', '3','A', '4','5', '6','B',
            '7','8', '9','C', '*', '0', '#','D'}

digit = [[0,0,0,0,0,0,1],[1,0,0,1,1,1,1],[0,0,1,0,0,1,0],[0,0,0,0,1,1,0],
          [1,0,0,1,1,0,0],[0,1,0,0,1,0,0],[0,1,0,0,0,0,0],[0,0,0,1,1,1,1],
          [0,0,0,0,0,0,0],[0,0,0,0,1,0,0]]

period = [0.001,.002,.003,.004,0.005,.006,.007,.008,.009,.01]

t=.01

digits = [Pin(4, Pin.OUT), Pin(5, Pin.OUT), Pin(6, Pin.OUT), Pin(7, Pin.OUT)]

segments = [Pin(8, Pin.OUT), Pin(9, Pin.OUT), Pin(10, Pin.OUT), Pin(11,
Pin.OUT), Pin(12, Pin.OUT), Pin(13, Pin.OUT), Pin(14, Pin.OUT)]

display(x)

    for (i = 0; i < 4; i++) do
        for (segment in segments) do
            segment.on()
        end_for
        digits[i].off()
        for (j = 0; j < 7; j++) do
            if (digit[int(x[i])][j]) then
                segments[j].on()
            else

```

```

        segments[j].off()

    end_if
end_for
digits[i].on()
end_for
sleep()

keyInput()
for (i = 0; i < 4; i++) do
    rows[i].on()
    for (j = 0; j < 4; j++) do
        if (cols[j].on()) then
            rows[i].off()
            return keys[i * 4 + j]
        end_if
    end_for
    rows[i].off()
end_for
return ''

out=Pin(28,Pin.OUT)
out.off()
DP.on()
curr=9
while(true) do
    c = keyInput()
    while (keyIsPressed()) do
        display(curr)
        sleep()
    end_while
    if (c.isBetween(0,10)) then
        curr = c
        t = period[curr]
    else_if(c == 0) then
        curr = 10
        t = period[curr]
    else_if(c == 'C' && curr < 10) then

```

```

        curr += 1
        t = period[curr]
    else_if (c == 'D' && curr > 1) then
        curr -= 1
        t = period[curr]
    end_if
    if(out.isOn()) then
        out.off()
        display(curr)
    else
        out.on()
        display(curr)
    end_if
end_while

```

## 2 Analiza programskog rješenja

### 2.1 Zadatak 1

Zadatak 1 predstavlja prvi zadatak u radu sa tastaturom (keypad-om). Prvo je potrebno navedenu tastaturu povezati sa *LPC1114ETF* sistemom, upravo onako kako je u postavci objašnjeno. Zatim je potrebno implementirati traženu funkcionalnost. Pritiskom cifre na tastaturi treba da se upali odgovarajuća LED dioda, te ista da ostane upaljena sve dok se ne pritisne tipka 'C', čime se sve upaljene LED diode gase. Za rad sa tastaturom, potrebno je deklarirati niz znakova koji predstavljaju sve moguće tipke na tastaturi. Potrebno je i definisati funkciju, koju smo nazvali *keyInput()*. Zadatak ove funkcije je da pri svakom pozivu prolazi kroz redove tastature, i provjerava da li jedna od kolona signalizira pritisnutu tipku na toj poziciji. Znači, suština ove funkcije jeste da provjeri ima li koja tipka pritisnuta, te ukoliko ima, da vrati odgovarajući znak iz prethodno definisanog niza znakova. U *main* funkciji se postavlja početna vrijednost LED dioda, tj. gase se sve LED diode. Zatim se u beskonačnoj *while* petlji čita pritisnuti znak, te ako je validan znak, izvršava se funkcionalnost koja je s njim definisana. Ukoliko je znak jedan od brojeva u zatvorenom intervalu [1,8], pali se odgovarajuća dioda, a ukoliko to znak 'C', gase se sve prethodno upaljene diode.

### 2.2 Zadatak 2

U drugom zadatku se prvi put susrećemo sa 7-segmentnim displejem, kojeg je potrebno povezati sa *picoETF* razvojnim sistemom. Za rad sa 7-segmentnim displejem, potrebno je za



svaku cifru [0-9], definisati vrijednosti za 7 segmenata koji se koriste za prikaz iste. Znači, potrebno je za svaku cifru definisati koji segmenti trebaju biti upaljeni, a koji ugašeni. Za pregledniji kod, definisali smo i četiri funkcije, koje služe za prikaz brojeva na ovom displeju. Pa tako, funkcija *prikazi\_cifru()*, za zadanu cifru, na osnovu prethodno definisanog niza vrijednosti segmenata, pali i gasi segmente, što u suštini prikazuje cifru na displeju. Funkcijom *prikazi\_poziciju()* određujemo na kojem mjestu će se sljedeća proslijeđena cifra prikazati. Ova funkcionalnost nam je potrebna, pošto radimo sa četverocifrenim 7-segmentnim displejem. Funkcija *daj\_cifru()* daje traženu cifru za zadani broj. Naprimjer, ukoliko imamo broj 345 i tražimo njegovu drugu cifru, ova funkcija će nam vratiti vrijednost 4. I na kraju, funkcija *prikazi\_broj*, uz pomoć prethodno definisanih funkcija prikazuje zadani broj na displeju.

Zadatak je tražio da se na osnovu pritisnutih tastera mijenja vrijednost na displeju, na sljedeći način: pritiskom na *taster 1* potrebno je uvećati vrijednost na displeju za 1, pritiskom na *taster 2* potrebno je vrijednost umanjiti za 1, pritiskom na *taster 3* vrijednost se postavlja na 0, a pritiskom na *taster 4* pokreće se, odnosno zaustavlja, automatsko brojanje. Za implementaciju ovoga, potrebno je postaviti početnu vrijednost broja na 0, te u beskonačnoj *while* petlji provjeravati da li je neki taster pritisnut. Ukoliko je pritisnut *taster 1*, broj se treba uvećati te se program blokira sve dok je taj taster pritisnut. Na analogan način su implementirane funkcionalnosti za *taster 2* i *taster 3*. Za *taster 4* korištena je boolean varijabla *auto* čija inicijalna vrijednost je *false*. Pritiskom na *taster 4* se vrijednost *auto* varijable mijenja, te se pokreće/zaustavlja automatsko brojanje. Potrebno je još na kraju provjeravati da li je vrijednost broja u opsegu reprezentativnih brojeva.

### 2.3 Zadatak 3 – izbor 1

Zadatak 3 koristi i tastaturu iz prvog zadatka, te 7-segmentni displej iz drugog zadatka. Njihove funkcionalnosti su implementirane na sličan način kao u prethodnim zadacima. U zadatku je potrebno implementirati „generator četvrtki“, pri čemu se pritiskom na taster 'A' signal pokreće i zaustavlja, dok se pritiskom na tastere 0-9 mijenja period signala (1 do 10ms). Pritisak na taster 'C' se period signala povećava za 1ms, dok se pritiskom na taster 'D' period umanjuje za 1ms. Na 7-segmentnom displeju treba biti prikazan trenutni period signala. Za realizaciju napravljen je niz pod nazivom *period* u kojem su smještene vrijednosti mogućih perioda signala. Za početak je postavljen period od 10ms, a signal je ugašen. U beskonačnoj *while* petlji se učitava pritisnuti znak, te se kroz if uslove provjerava koji je znak pritisnut. Na osnovu toga se mijenja trenutni period signala. Također, program se blokira dok se tipka drži pritisnutom. Na kraju petlje se provjerava koja je bila prethodna vrijednost signala, tj. da li je signal bio

upaljen i ugašen, te na osnovu toga mijenja njegova vrijednost. Signal se pali/gasi, te se na displeju prikazuje trenutni period. U funkciji *display* je postavljen *sleep*. Mana ovog rješenja je ta što period nije uvijek upotpunosti precizan, iz razloga što se dio vremena troši na ostatak koda, a ne samo na čekanje. Kao rezultat ovoga, period signala je često neznatno veći od traženog. Elegantno rješenje ovog problema bi bilo korištenje tajmera, međutim kako tajmeri nisu prerađeni do vježbe tako nisu ni korišteni.

### 3 Korišteni hardverski resursi

Na ovoj vježbi su korišteni sljedeći razvojni sistemi:

- *LPC1114ETF*, baziran na mikrokontroleru *NXP LPC1114FN28*
- *picoETF*, baziran na mikrokontroleru *RP2040*
- matrična tastatura
- četverocifreni 7-segmentni displej

Na sistemu *LPC1114ETF* su za ovu vježbu korišteni sljedeći elementi:

- 8x LED dioda

Na sistemu *picoETF* su korišteni sljedeći elementi:

- 4x tastera

### 4 Zaključak

Ova vježba je za cilj imala da se studenti upoznaju sa korištenjem višebitnih digitalnih ulaza i izlaza, te sa načinima korištenja matrične tastature i višecifrenih 7-segmentnih displeja. Zbog prvog praktičnog susreta sa matričnom tastaturom i višecifrenim 7-segmentnim displejem, shvatanje principa funkcionisanja i načina korištenja je predstavljao zahtjevniji dio zadataka. Unatoč raznim prepravkama koda, koje smo kroz ovu vježbu morali napraviti, rješenja su postignuta i postavljeni zadaci su uspješno urađeni u predviđenom vremenu. Kroz rješavanje zadataka uspješno su se shvatili načini korištenja ovih uređaja, te je sami cilj vježbe uspješno postignut.

## 5 Prilog

### 5.1 Zadatak 1/izvorni kod

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

BusOut leds (LED0,LED1,LED2,LED3,LED4,LED5,LED6,LED7);
DigitalOut E(LED_ACT);

BusOut rows(dp18,dp17,dp15,dp16);
BusIn cols(dp9,dp10,dp11,dp13);

char keys[16] = {'1','2','3','A',
                 '4','5','6','B',
                 '7','8','9','C',
                 '*','0','#','D'};

char keyInput(){
    for(int i=0;i<4;i++){
        rows=pow(2,i);
        int col=cols;

        if(col==1) return keys[i*4 + 0];
        else if(col==2) return keys[i*4 + 1];
        else if(col==4) return keys[i*4 + 2];
        else if(col==8) return keys[i*4 + 3];
    }
    return 0;
}

int main(){
    E=0;
    leds=0;
    while (true){
        char n=keyInput();
        if(n>'0' && n<'9'){
            leds[n-49]=1;
            while(cols>0){
```

```

        thread_sleep_for(100);
    }
}
else if(n=='C'){
    leds=0;
    while(cols>0){
        thread_sleep_for(100);
    }
}
thread_sleep_for(100);
}
}

```

## 5.2 Zadatak 2/izvorni kod

```

from machine import Pin
import time
time.sleep(0.1) # Wait for USB to become ready

cifre = {
    0: 0b1000000,
    1: 0b1111001,
    2: 0b0100100,
    3: 0b0110000,
    4: 0b0011001,
    5: 0b0010010,
    6: 0b0000010,
    7: 0b1111000,
    8: 0b0000000,
    9: 0b0010000,
    10: 0b1111111
}

mjesto = [Pin(m, Pin.OUT) for m in range(4, 8)]
segmenti = [Pin(s, Pin.OUT) for s in range(8, 15)]
tasteri = [Pin(i, Pin.IN) for i in range(4)]

```

```

def prikazi_cifru(cifra):
    binarno = cifre[cifra]
    for i in range(7):
        segmenti[i].value(binarno & 1)
        binarno = binarno >> 1

def prikazi_poziciju(pozicija):
    for i in range(4):
        mjesto[i].value(0)
    mjesto[pozicija].value(1)

def prikazi_broj(broj):
    n = 3
    cifra = daj_cifru(broj, n)
    while (n>=0):
        prikazi_cifru(10)
        prikazi_poziciju(3-n)
        prikazi_cifru(cifra)
        n -= 1
        cifra = daj_cifru(broj, n)
        time.sleep(0.001)

def daj_cifru(broj, n):
    return broj // 10 ** n % 10

auto = False
broj = 0

while True:
    if tasteri[0].value():
        broj += 1
        while (tasteri[0].value()):
            time.sleep(0.1)
    elif tasteri[1].value():
        broj -= 1
        while (tasteri[1].value()):

```

```

        time.sleep(0.1)
    elif tasteri[2].value():
        broj = 0
    elif tasteri[3].value():
        auto = not auto
        count = 0
        while(tasteri[3].value()):
            time.sleep(0.0001)
    if auto:
        if count == 9:
            broj += 1
            count = 0
            count += 1
    if broj > 9999:
        broj = 0
    if broj < 0:
        broj = 9999
    prikazi_broj(broj)
    time.sleep(0.1)

```

### 5.3 Zadatak 3 – izbor 1/izvorni kod

```

import time

from machine import Pin

time.sleep(0.1) # Wait for USB to become ready

R1=Pin(21,Pin.OUT)
R2=Pin(22,Pin.OUT)
R3=Pin(26,Pin.OUT)
R4=Pin(27,Pin.OUT)
rows = [R1,R2,R3,R4]

C1=Pin(0,Pin.IN, Pin.PULL_DOWN)
C2=Pin(1,Pin.IN, Pin.PULL_DOWN)
C3=Pin(2,Pin.IN, Pin.PULL_DOWN)
C4=Pin(3,Pin.IN, Pin.PULL_DOWN)
cols = [C1,C2,C3,C4]

D1=Pin(4, Pin.OUT)

```

```

D2=Pin(5, Pin.OUT)
D3=Pin(6, Pin.OUT)
D4=Pin(7, Pin.OUT)
A=Pin(8, Pin.OUT)
B=Pin(9, Pin.OUT)
C=Pin(10, Pin.OUT)
D=Pin(11, Pin.OUT)
E=Pin(12, Pin.OUT)
F=Pin(13, Pin.OUT)
G=Pin(14, Pin.OUT)
DP=Pin(15, Pin.OUT)

digits = [D1, D2, D3, D4]
segments = [A,B,C,D,E,F,G]

keys = ['1','2','3','A',
        '4','5','6','B',
        '7','8','9','C',
        '*', '0', '#', 'D']

digit = [[0,0,0,0,0,0,1],[1,0,0,1,1,1,1],
          [0,0,1,0,0,1,0],[0,0,0,0,1,1,0],
          [1,0,0,1,1,0,0],[0,1,0,0,1,0,0],
          [0,1,0,0,0,0,0],[0,0,0,1,1,1,1],
          [0,0,0,0,0,0,0],[0,0,0,0,1,0,0]]

period = [0.001,.002,.003,.004,0.005,.006,.007,.008,.009,.01]
t=.01

def keyInput():
    for i in range(0,4):
        rows[i].on()
        for j in range(0,4):
            if cols[j].value():
                rows[i].off()
                return keys[i*4+j]

```

```

        rows[i].off()
    return ''

def display(x):
    for i in range(0,4):
        for seg in segments:
            seg.on()
        digits[i].off()
        for j in range(0,7):
            if digit[int(x[i])][j]:
                segments[j].on()
            else:
                segments[j].off()
        digits[i].on()
    time.sleep(t*0.3)

out=Pin(28,Pin.OUT)
out.off()
DP.on()
curr=9

while True:
    c=keyInput()
    while keyInput()!='':
        display('{0:04d}'.format(int(curr+1)))
        time.sleep(t/2)
    if c>'0' and c<='9':
        curr=int(c)
        curr-=1
        t=period[curr]
    elif c=='0':
        curr=9
        t=period[curr]
    elif c=='C':
        if curr<9:

```



```

        curr+=1
        t=period[curr]
elif c=='D':
    if curr>0:
        curr-=1
        t=period[curr]
if out.value() == 1:
    out.off()
    display('{0:04d}'.format(int(curr+1)))
else:
    out.on()
    display('{0:04d}'.format(int(curr+1)))

```