

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

**Izvještaj za laboratorijsku vježbu 2**  
Sistemi FRDM-KL25Z, LPC1114ETF i Mbed OS

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

Sarajevo, mart 2024.

# Sadržaj

<b>1</b>	<b>Pseudokod</b>	<b>1</b>
1.1	Zadatak 1	1
1.1.1	Zadatak 1 – prvi dio	1
1.1.2	Zadatak 1 – drugi dio	1
1.2	Zadatak 2	2
1.2.1	Zadatak 2 – prvi dio	2
1.2.2	Zadatak 2 – drugi dio	2
1.3	Zadatak 3	3
1.4	Zadatak 4	4
1.5	Zadatak 5	5
1.6	Dodatni zadatak 1	5
<b>2</b>	<b>Analiza programskog rješenja</b>	<b>6</b>
2.1	Zadatak 1	6
2.1.1	Zadatak 1 – prvi dio	6
2.1.2	Zadatak 1 – drugi dio	6
2.2	Zadatak 2	7
2.2.1	Zadatak 2 – prvi dio	7
2.2.2	Zadatak 2 – drugi dio	7
2.3	Zadatak 3	7
2.4	Zadatak 4	7
2.5	Zadatak 5	8
2.6	Dodatni zadatak 1	8
<b>3</b>	<b>Korišteni hardverski resursi</b>	<b>9</b>
3.1	LPC1114ETF	9
3.2	FRDM-KL25Z	9

<b>4</b>	<b>Zaključak.....</b>	<b>10</b>
<b>5</b>	<b>Prilog.....</b>	<b>11</b>
5.1	Zadatak 1 .....	11
5.1.1	Zadatak 1 – prvi dio: Izvorni kôd .....	11
5.1.2	Zadatak 1 – drugi dio: Izvorni kôd .....	11
5.2	Zadatak 2 .....	12
5.2.1	Zadatak 2 – prvi dio: Izvorni kôd .....	12
5.2.2	Zadatak 2 – drugi dio: Izvorni kôd .....	12
5.3	Zadatak 3: Izvorni kôd .....	13
5.4	Zadatak 4: Izvorni kôd .....	14
5.5	Zadatak 5: Izvorni kôd .....	15
5.6	Dodatni zadatak 1: Izvorni kôd .....	16

# 1 Pseudokod

## 1.1 Zadatak 1

### 1.1.1 Zadatak 1 – prvi dio

```
act ← output(LED_ACT)  
led ← output(LED1)
```

```
value(act) ← 0
```

```
while true
```

```
    value(led) ← 1  
    pauseMicro( $10^6$ )
```

```
    value(led) ← 0  
    pauseMicro( $10^6$ )
```

### 1.1.2 Zadatak 1 – drugi dio

```
act ← output(LED_ACT)  
led ← {output(LED1), output(LED2), output(LED3), output(LED4)}
```

```
value(act) ← 0
```

```
while true
```

```
    for i in 0..3  
        value(led[i]) ← 1  
        pauseMicro( $10^6$ )
```

```
        value(led[i]) ← 0  
        pauseMicro( $10^6$ )
```

## 1.2 Zadatak 2

### 1.2.1 Zadatak 2 – prvi dio

```
act ← output(LED_ACT)
diodes ← outputCollection(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7)
button ← input(Taster_1)
```

```
value(act), value(diodes) ← 0
counter, step ← 0, 1
```

```
while true
```

```
  if value(button) = 1
    step ← -step
```

```
  if (counter < 255 or step = -1) and (counter > 0 or step = 1)
    counter ← counter + step
```

```
  value(diodes) ← counter
  pauseMicro(106)
```

### 1.2.2 Zadatak 2 – drugi dio

```
act ← output(LED_ACT)
diodes ← outputCollection(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7)
button ← input(Taster_1)
```

```
value(act), value(diodes) ← 0
counter ← 0
```

```
while true
```

```
  value(diodes) ← counter
```

```
  if value(button) = 0 and counter < 255
    counter ← counter + 1
```

```
  else if value(button) = 1 and counter > 0
    counter ← counter - 1
```

```
  pauseMicro(106)
```

### 1.3 Zadatak 3

```
act ← output(LED_ACT)  
diodes ← outputCollection(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7)  
  
value(act), value(diodes) ← 0  
  
while true  
  while i ← 1 ≠ 256 factor 2  
    value(diodes) ← i  
    pauseMicro(105)  
  
  value(diodes) ← 255  
  pauseMicro(105)  
  
  while i ← 128 ≠ 0 factor 0.5  
    value(diodes) ← i − 1  
    pauseMicro(105)  
  
  value(diodes) ← 0  
  pauseMicro(105)
```

## 1.4 Zadatak 4

```
act ← output(LED_ACT)
led ← output(LED1)
diodes ← outputCollection(LED0,LED1,LED2,LED3,LED4,LED5,LED6,LED7)
button1,2 ← input(Taster_1,Taster_2)

value(act), value(diodes) ← 0

while true
  value(led) ← 1
  pauseMicro(5 · 105)

  if value(button1) and time ← 105 or value(button2) and time ← 5 · 105
    while i ← 1 ≠ 256 factor 2
      value(diodes) ← i
      pauseMicro(time)

    value(diodes) ← 255
    pauseMicro(time)

    while i ← 128 ≠ 0 factor 0.5
      value(diodes) ← i - 1
      pauseMicro(time)

    value(diodes) ← 0
    pauseMicro(time)

  value(led) ← 0
  pauseMicro(5 · 105)
```

## 1.5 Zadatak 5

```
act ← output(LED_ACT)
led ← output(LED0)

value(act) ← 0
period, factor, step ←  $T \cdot 10^5, 300, 9$ 

while true
  if factor = 30 or factor = 570
    step ← -step

  value(led) ← 1
  pauseMicro(factor · period ÷ 30)

  value(led) ← 0
  pauseMicro((600 - factor) · period ÷ 30)

  factor ← factor + step
```

## 1.6 Dodatni zadatak 1

```
rgb ← {output(LED_RED), output(LED_GREEN), output(LED_BLUE)}

period ← { $T_{red} \cdot 10^5, T_{green} \cdot 10^5, T_{blue} \cdot 10^5$ }
factor, step ← 300, 9

while true
  if factor = 30 or factor = 570
    step ← -step

  for i in 0..2
    value(rgb[i]) ← 1
    pauseMicro(factor · period[i] ÷ 30)

    value(rgb[i]) ← 0
    pauseMicro((600 - factor) · period[i] ÷ 30)

  factor ← factor + step
```



## 2 Analiza programskog rješenja

Napomena: Ponavljajući dijelovi kôda u različitim zadacima će se analizirati samo prilikom prvog pojavljivanja.

### 2.1 Zadatak 1

#### 2.1.1 Zadatak 1 – prvi dio

[2] uključivanje *header* fajla radi lakšeg korištenja izvoda i funkcija

[4] kreiranje objekata za očitavanje i postavljanje vrijednosti digitalnih izlaza **P1\_5** i **P0\_1** na kojima se nalaze pin **LED\_ACT** i dioda **LED1** respektivno

[7] postavljanje pina **LED\_ACT** na 0V kako bi se omogućio prikaz na LED diodama

[10/13] postavljanje stanja diode na 1/0

[11/14] pauza u izvršavanju programa u trajanju od 1 sekunde

#### 2.1.2 Zadatak 1 – drugi dio

[5] kreiranje niza objekata za očitavanje i postavljanje vrijednosti digitalnih izlaza **P0\_1-P0\_4** na kojima se nalaze diode **LED1-LED4**

[11-18] manipulacija nizom digitalnih izlaza analogno prvom dijelu Zadatka 1

## 2.2 Zadatak 2

### 2.2.1 Zadatak 2 – prvi dio

[6] kreiranje instance klase za manipulaciju nizom digitalnih izlaza za jednostavno očitavanje i postavljanje vrijednosti digitalnih izlaza **P0\_0-P0\_7** na kojima se nalaze diode **LED0-LED7**

[7] kreiranje objekta za očitavanje i postavljanje vrijednosti digitalnog ulaza **P0\_8** na kojem se nalazi taster **Taster\_1**

[11] postavljanje stanja *bus*-a, tj. svih dioda na 0 jer su početna stanja 1 na sistemu LPC1114ETF

[15] promjena predznaka vrijednosti koraka za izmjenu smjera brojanja pritiskom na taster

[16-17] uvećanje vrijednosti brojača ukoliko je ista manja od 255, odnosno smanjenje vrijednosti brojača ukoliko je ista veća od 0

[19] postavljanje stanja *bus*-a na vrijednost brojača čime se realizuje odgovarajući prikaz

### 2.2.2 Zadatak 2 – drugi dio

[17-18] uvećavanje vrijednosti brojača dok je taster otpušten, odnosno umanjivanje vrijednosti brojača dok je taster pritisnut

## 2.3 Zadatak 3

[15-18] realizacija sekvence brojeva 1, 2, 4, 8, 16, 32, 64, 128 (stepeni dvojke)

[23-26] realizacija sekvence brojeva 127, 63, 31, 15, 7, 3, 1, 0 (stepeni dvojke umanjeni za 1)

[17/21/25/29] pauza u izvršavanju programa u trajanju od 0.1 sekunde

## 2.4 Zadatak 4

[14-15/37-38] realizacija paljenja i gašenja diode **LED0** svakih 0.5 sekundi

[19] varijabli koja čuva vrijeme izmjene trčućeg svjetla se dodjeljuje vrijednost 0.1 odnosno 0.5 pritiskom tastera **Taster\_1** odnosno tastera **Taster\_2** respektivno, čime se aktivira trčće svjetlo

[20-34] analogni komentari na osnovu Zadatka 3

## 2.5      Zadatak 5

[6] deklaracija i inicijalizacija početnog vremena uključenosti i isključenosti u sekundama

[15] deklaracija perioda inicijaliziranog na osnovu vrijednosti početnog vremena

[21] deklaracija i inicijalizacija faktora kojim se množi period i koraka kojim se povećava faktor

[28] promjena predznaka vrijednosti koraka ukoliko je faktor dostigao granične vrijednosti

[30/34] postavljanje stanja diode na 1/0

[32/36] pauza u izvršavanju programa sa „suprotnim“ trajanjem

[38] odgovarajuće povećanje odnosno smanjenje veličine faktora u ovisnosti od predznaka vrijednosti koraka

## 2.6      Dodatni zadatak 1

Napomena: Zahtjevi definisani u ovom zadatku su djelimično izmijenjeni na laboratorijskoj vježbi.

[3-5] deklaracija i inicijalizacija početnih vremena uključenosti i isključenosti svih komponenti u okviru RGB LED diode u sekundama

[7] kreiranje niza objekata za očitavanje i postavljanje vrijednosti digitalnih izlaza **PTB18**, **PTB19** i **PTD1** na kojima se nalaze diode **LED\_RED**, **LED\_GREEN** i **LED\_BLUE** respektivno

[10-25] manipulacija nizom digitalnih izlaza analogno Zadatku 5 za realizaciju sekvencijalnog uključivanja i isključivanja svih komponenti u okviru RGB LED diode

### 3 Korišteni hardverski resursi

#### 3.1 LPC1114ETF

	<i>Komponenta</i>	<i>Opis</i>	<i>Količina</i>
1	LPC1114FN28	mikrokontroler	1
2	taster	digitalni ulaz	2
3	LED dioda	digitalni izlaz	8
4	USB A kabal	napajanje i komunikacija	1
5	PGM taster	pokretanje <i>bootloader</i> -a	1
6	LED_ACT pin	uključivanje LED dioda	1

#### 3.2 FRDM-KL25Z

	<i>Komponenta</i>	<i>Opis</i>	<i>Količina</i>
1	KL25Z128VLK	mikrokontroler	1
2	RGB LED dioda	digitalni izlaz	1
3	USB A – USB Mini B	napajanje i komunikacija	1

## 4      **Zaključak**

Osim blage nesigurnosti u početku zbog rada s novim sistemima, izrada laboratorijske vježbe je protekla bez problema.

Klasa „BusOut“ je veoma pojednostavila implementaciju!

Zadatak 5 je izuzeo nešto vremena kako bi se osmislila realizacija koja minimizira rad s realnim brojevima kako bi se izbjegle greške nastale nagomilavanjem manjih grešaka prilikom iteriranja.

S obzirom na rad s novim hardverom, većina sadržaja laboratorijske vježbe čini novostečeno znanje. Naravno, temeljni koncepti su zadržani.

## 5 Prilog

### 5.1 Zadatak 1

#### 5.1.1 Zadatak 1 – prvi dio: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03:
04: DigitalOut act(LED_ACT), led(LED1);
05:
06: int main() {
07:     act = 0;
08:
09:     for (;;) {
10:         led = 1;
11:         wait_us(1e6);
12:
13:         led = 0;
14:         wait_us(1e6);
15:     }
16: }
```

#### 5.1.2 Zadatak 1 – drugi dio: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03:
04: DigitalOut act(LED_ACT);
05: DigitalOut led[] = {LED1, LED2, LED3, LED4};
06:
07: int main() {
08:     act = 0;
09:
10:     for (;;) {
11:         for (int i = 0; i < 4; i++) {
12:             led[i] = 1;
13:             wait_us(1e6);
14:
15:             led[i] = 0;
16:             wait_us(1e6);
17:         }
18:     }
19: }
```

## 5.2 Zadatak 2

### 5.2.1 Zadatak 2 – prvi dio: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03: #include "BusOut.h"
04:
05: DigitalOut act(LED_ACT);
06: BusOut diodes(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
07: DigitalIn button(Taster_1);
08:
09: int main() {
10:     act.write(0);
11:     diodes.write(0);
12:     int counter = 0, step = 1;
13:
14:     for (;;) {
15:         if (button.read()) step *= -1;
16:         if ((counter < 255 || step == -1) && (counter > 0 || step == 1))
17:             counter += step;
18:
19:         diodes.write(counter);
20:         wait_us(1e6);
21:     }
22: }
```

### 5.2.2 Zadatak 2 – drugi dio: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03: #include "BusOut.h"
04:
05: DigitalOut act(LED_ACT);
06: BusOut diodes(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
07: DigitalIn button(Taster_1);
08:
09: int main() {
10:     act.write(0);
11:     diodes.write(0);
12:     int counter = 0;
13:
14:     for (;;) {
15:         diodes.write(counter);
16:
17:         if (!button.read() && counter < 255) counter++;
18:         else if (button.read() && counter > 0) counter--;
19:
20:         wait_us(1e6);
21:     }
22: }
```

## 5.3 Zadatak 3: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03: #include "BusOut.h"
04:
05: DigitalOut act(LED_ACT);
06: BusOut diodes(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
07:
08: int main() {
09:     act.write(0);
10:     diodes.write(0);
11:
12:     const int microseconds = 1e5;
13:
14:     for (;;) {
15:         for (int i = 1; i != 256; i *= 2) {
16:             diodes.write(i);
17:             wait_us(microseconds);
18:         }
19:
20:         diodes.write(255);
21:         wait_us(microseconds);
22:
23:         for (int i = 128; i != 0; i /= 2) {
24:             diodes.write(i - 1);
25:             wait_us(microseconds);
26:         }
27:
28:         diodes.write(0);
29:         wait_us(microseconds);
30:     }
31: }
```



## 5.4 Zadatak 4: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03: #include "BusOut.h"
04:
05: DigitalOut act(LED_ACT), led(LED1);
06: BusOut diodes(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
07: DigitalIn button1(Taster_1), button2(Taster_2);
08:
09: int main() {
10:     act.write(0);
11:     diodes.write(0);
12:
13:     for (;;) {
14:         led.write(1);
15:         wait_us(5e5);
16:
17:         int micro;
18:
19:         if ((button1.read() && (micro = 1e5)) || (button2.read() && (micro = 5e5))) {
20:             for (int i = 1; i != 256; i *= 2) {
21:                 diodes.write(i);
22:                 wait_us(micro);
23:             }
24:
25:             diodes.write(255);
26:             wait_us(micro);
27:
28:             for (int i = 128; i != 0; i /= 2) {
29:                 diodes.write(i - 1);
30:                 wait_us(micro);
31:             }
32:
33:             diodes.write(0);
34:             wait_us(micro);
35:         }
36:
37:         led.write(0);
38:         wait_us(5e5);
39:     }
40: }
```

## 5.5 Zadatak 5: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03:
04: DigitalOut act(LED_ACT), led(LED0);
05:
06: const double T = 0.005;
07:
08: int main() {
09:     act.write(0);
10:
11:     // regularna vrijednost: period = T
12:     // 1] množenje s 1e6 (zbog funkcije "wait_us")
13:     // 2] dijeljenje s 10 (zbog [*])
14:     // dakle: T * 1e6 / 10 => T * 1e5
15:     double period = T * 1e5;
16:
17:     // regularne vrijednosti: factor = 1, step = 0.9 / 30
18:     // 1] množenje s 10 (kako bi se riješili decimale u 0.9) [*]
19:     // 2] množenje s 30 (kako bi se riješili dijeljenja s 30) [**]
20:     // dakle: 1 * 10 * 30, 0.9 / 30 * 10 * 30 => 300, 9
21:     int factor = 300, step = 9;
22:
23:     for (;;) {
24:         // regularna provjera: factor == 0.1 || factor == 1.9
25:         // 1] množenje s 10 (zbog [*])
26:         // 2] množenje s 30 (zbog [**])
27:         // dakle: 0.1 * 10 * 30 || 1.9 * 10 * 30 => 30 || 570
28:         if (factor == 30 || factor == 570) step *= -1;
29:
30:         led.write(1);
31:         // dijeljenje s 30 zbog [**]
32:         wait_us(factor * period / 30);
33:
34:         led.write(0);
35:         // dijeljenje s 30 zbog [**]
36:         wait_us((600 - factor) * period / 30);
37:
38:         factor += step;
39:     }
40: }
```

## 5.6 Dodatni zadatak 1: Izvorni kôd

```
01: #include "mbed.h"
02:
03: const double T_red = 0.05;
04: const double T_green = 0.05;
05: const double T_blue = 0.05;
06:
07: DigitalOut rgb[] = {LED_RED, LED_GREEN, LED_BLUE};
08:
09: int main() {
10:     double period[] = {T_red * 1e5, T_green * 1e5, T_blue * 1e5};
11:     int factor = 300, step = 9;
12:
13:     for (;;) {
14:         if (factor == 30 || factor == 570) step *= -1;
15:
16:         for (int i = 0; i < 3; i++) {
17:             rgb[i].write(1);
18:             wait_us(factor * period[i] / 30);
19:
20:             rgb[i].write(0);
21:             wait_us((600 - factor) * period[i] / 30);
22:         }
23:
24:         factor += step;
25:     }
26: }
```