

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

**IZVJEŠTAJ ZA LABORATORIJSKU VJEŽBU BR. 2**

SISTEMI FRDM-KL25Z, LPC1114ETF I MBED OS

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

Sarajevo, 24.3.2025.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Postavka zadatka</b>	<b>1</b>
2.1	Zadatak 1 i 1.1 . . . . .	1
2.2	Zadatak 2 . . . . .	2
2.3	Zadatak 3 . . . . .	2
2.4	Zadatak 4 . . . . .	3
2.5	Zadatak 5 . . . . .	4
2.6	Dodatni zadatak . . . . .	5
<b>3</b>	<b>Implementacija</b>	<b>7</b>
3.1	Zadatak 1 i 1.1 . . . . .	7
3.2	Zadatak 2 . . . . .	7
3.3	Zadatak 3 . . . . .	8
3.4	Zadatak 4 . . . . .	8
3.5	Zadatak 5 . . . . .	9
3.6	Dodatni zadatak . . . . .	10
<b>4</b>	<b>Testiranje i rezultati</b>	<b>11</b>
<b>5</b>	<b>Zaključak</b>	<b>12</b>
<b>6</b>	<b>Prilog</b>	<b>13</b>
6.1	Zadatak 1 i 1.1 . . . . .	13
6.2	Zadatak 2 . . . . .	14
6.3	Zadatak 3 . . . . .	15
6.4	Zadatak 4 . . . . .	16
6.5	Zadatak 5 . . . . .	18
6.6	Dodatni zadatak . . . . .	19

# 1 Uvod

Cilj ove laboratorijske vježbe bio je upoznavanje s alatima i razvojnim sistemima koji se koriste u izradi programa za ugradbene sisteme. Korišteni razvojni alati uključivali su Online Mbed simulator i ARM Keil Studio, dok su korišteni hardverski sistemi bili LPC1114ETF i FRDM-KL25Z.

U okviru vježbe realizovani su zadaci koji uključuju upravljanje LED diodama, implementaciju binarnog brojača, trčećeg svjetla i interakciju s tasterima. Programski kod razvijen je prvo u Mbed simulatoru, a zatim prenesen na stvarni hardver korištenjem ARM Keil Studija.

Korišteni hardver i softver:

Komponenta	Opis
Mbed simulator	Alat za simulaciju Mbed aplikacija u web browseru
ARM Keil Studio	Razvojno okruženje za programiranje ARM mikrokontrolera
LPC1114ETF	Razvojni sistem baziran na mikrokontroleru LPC1114FN28
FRDM-KL25Z	Razvojni sistem baziran na mikrokontroleru KL25Z128VLK

## 2 Postavka zadataka

### 2.1 Zadatak 1 i 1.1

#### Mbed simulator:

Napisati kod tako da se pale i gasi redom LED1, LED2, LED3 i LED4, na način da se prvo upali LED1 u trajanju od 1s, zatim ugasi LED1 u trajanju od 1s. Nakon toga se upali LED2 u trajanju od 1s i ugasi LED2 u trajanju od 1s, itd.

#### LPC1114ETF

Izvršiti obje verzije programa i na razvojnom sistemu LPC1114ETF.

---

#### Zadatak 1.1

---

```
1: Inicijalizacija:
2: leds[0 : 3] ← izlazi                                ▷ LED1 do LED4
3: cnt ← 0
4:
5: while True do
6:   leds[cnt % 4] ← 1                                  ▷ Upali trenutnu LED diodu
7:   Sačekaj 1 sekundu
8:   leds[cnt mod 4] ← 0                                ▷ Ugasi trenutnu LED diodu
9:   cnt ← (cnt % 4) + 1                                  ▷ Povećaj brojač s modulo operacijom
10:  Sačekaj 1 sekundu
11: end while
```

---

## 2.2 Zadatak 2

### Mbed simulator:

Dodati 8 LED dioda i korištenjem tih dioda realizovati binarni brojač, koji se uvećava za jedan svakih 200ms. Svakim klikom na BUTTON1 smjer brojanja se treba mijenjati. Modificirati kod tako da brojač broji unazad kada je BUTTON2 pritisnut (obratiti pažnju na napomenu o načinu simuliranja pritisnutog tastera)

### LPC1114ETF

Isti program izvršiti na razvojnom sistemu LPC1114ETF.

---

#### Zadatak 2

---

```
1: Inicijalizacija:
2: leds[0 : 7] ← izlazi                                ▷ p7 do p14
3: button1, button2 ← ulazi                             ▷ Tasteri
4: cnt ← 0
5: dir ← tačno                                          ▷ Smijer kretanja
6:
7: while tačno do
8:   if button1 == 1 then
9:     dir ← ¬dir                                       ▷ Promijeni smjer
10:  end if
11:
12:   if button2 == 1 then
13:     cnt ← cnt - 1                                   ▷ Dekrementiraj
14:   else
15:     cnt ← cnt + (dir?1 : -1)                       ▷ Inkrementiraj ili dekrementiraj
16:   end if
17:
18:   cnt ← cnt&0xFF                                    ▷ Ograniči na 8 bita
19:   leds ← cnt                                        ▷ Prikaži na LED diodama
20:   Sačekaj 200 ms
21: end while
```

---

## 2.3 Zadatak 3

### Mbed simulator:

Korištenjem 8 LED dioda iz prethodnog zadatka realizovati “trčeće svjetlo” na LED diodama. Prvo se uključi prva dioda, zatim druga dioda i tako redom (prethodno uključene diode ostaju uključene). Dolaskom na kraj (uključeno svih 8 dioda), potrebno je realizirati trčeće svjetlo u obrnutom smjeru gašenjem u svakom koraku po jednu LED. Brzina izmjene je 0.1s. Nakon što se isključe sve LED, proces se ponavlja.

### LPC1114ETF

Isti program izvršiti na razvojnom sistemu LPC1114ETF.

---

### Zadatak 3

---

```
1: Inicijalizacija:
2: leds[0 : 7]  $\leftarrow$  izlazi ▷ p7 do p14
3:
4: while tačno do
5:   for  $i \leftarrow 0$  do 7 do
6:     leds  $\leftarrow$  (leds  $\ll$  1) + 1 ▷ Pomijeri lijevo i dodaj 1
7:     Sačekaj 100 ms
8:   end for
9:
10:  for  $i \leftarrow 0$  do 7 do
11:    leds  $\leftarrow$  (leds  $\gg$  1) ▷ Pomijeri desno
12:    Sačekaj 100 ms
13:  end for
14: end while
```

---

## 2.4 Zadatak 4

### Mbed simulator:

Modificirati prethodni zadatak tako da se nakon pokretanja sistema svakih 0.5s pali i gasi LED0, što označava da se čeka pritisak na taster. Ako se pritisne Taster 1, aktivira se trčeće svjetlo kao u prethodnom zadatku, sa vremenom izmjene od 0.1s. Ukoliko se pritisne Taster 2, tada je vrijeme izmjene 0.5s. Nakon što se završi jedan ciklus, ponovo se počinje paliti i gasiti LED0, sve do novog pritiska na taster.

### LPC1114ETF

Isti program izvršiti na razvojnom sistemu LPC1114ETF.

---

#### Zadatak 4

---

```
1: Inicijalizacija:
2: leds[0 : 7] ← izlazi                                ▷ p7 do p14
3: led0 ← p7
4: button1, button2 ← ulazi                             ▷ Tasteri
5:
6: function TRCECE(t)
7:   for i ← 0 do 7 do
8:     leds ← (leds << 1) + 1                            ▷ Pomijeri lijevo i dodaj 1
9:     Sačekaj t ms
10:  end for
11:
12:  for i ← 0 do 7 do
13:    leds ← (leds >> 1)                                ▷ Pomijeri desno
14:    Sačekaj t ms
15:  end for
16: end function
17:
18: while tačno do
19:   while button1 == 0 i button2 == 0 do
20:    led0 ← 0
21:    Sačekaj 500 ms
22:    led0 ← 1
23:    Sačekaj 500 ms
24:   end while
25:
26:   if button1 == 1 then
27:     trcece(100)                                         ▷ Brz efekat
28:   else
29:     trcece(500)                                         ▷ Spor efekat
30:   end if
31: end while
```

---

## 2.5 Zadatak 5

### Mbed simulator:

Napisati i izvršiti program koji omogućava promjenu trajanja osvjetljenosti crvene LED diode na sljedeći način:

- Na početku dioda je uključena  $T$  sekundi, a nakon toga se isključuje  $T$  sekundi.
- Postepeno se povećava vrijeme uključenosti, te u toku  $20T$  dostiže vrijeme uključenja od  $2T$
- Istovremeno se smanjivanje vrijeme isključenosti do  $0.2T$ .
- Nakon toga vrijeme isključenja raste do  $2T$ , a vrijeme uključenja se smanjuje do  $0.2T$ .
- Proces se nakon toga ponavlja.

Analizirati razliku između izvršavanja programa za  $T=0.005$  i  $T=1s$

### FRDM-KL25Z ili LPC1114ETF:

Isti program izvršiti na razvojnom sistemu po izboru studenta

---

#### Zadatak 5

---

```
1: Inicijalizacija:
2: led  $\leftarrow$  LED1
3: T  $\leftarrow$  5.0                                ▷ Osnovni vremenski period u ms
4:
5: function CIKLUS(on, off)
6:     led  $\leftarrow$  1                                ▷ Upali LED
7:     Sačekaj on ms
8:     led  $\leftarrow$  0                                ▷ Ugasi LED
9:     Sačekaj off ms
10: end function
11:
12: while tačno do
13:     on  $\leftarrow$  T, off  $\leftarrow$  T                    ▷ Početne vrijednosti
14:
15:     for i  $\leftarrow$  0 do 20 do                        ▷ Faza povećanja širine impulsa
16:         ciklus(on, off)
17:         on  $\leftarrow$  on + T/20
18:         off  $\leftarrow$  off - 0.8  $\times$  T/20
19:     end for
20:
21:     for i  $\leftarrow$  0 do 20 do                        ▷ Faza smanjenja širine impulsa
22:         ciklus(on, off)
23:         off  $\leftarrow$  off + T  $\times$  0.09
24:         on  $\leftarrow$  on - T  $\times$  0.09
25:     end for
26: end while
```

---

## 2.6 Dodatni zadatak

### FRDM-KL25Z:

Napisati i izvršiti program koji simulira promjenu stanja RGB diode koristeći samo on/off režim, na sljedeći način:

- Početni trenutak: dioda svijetli crvenom bojom T sekundi.
- Nakon tog perioda, dioda se isključuje T/2 sekundi, a zatim se uključuje i svijetli plavom bojom T sekundi.
- Nakon T sekundi, dioda se isključuje opet na T/2 sekundi, a zatim se uključuje zelena LED koja svijetli T sekundi.
- Nakon što je zelena boja isključena T/2 sekundi, LED dioda prelazi u "miješani mod" tako što se naizmjenično pali crvena i plava boja, a trajanje uključenosti je T/4 sekunde u ukupnom trajanju od 2T sekundi.

- Kada se završi ovaj ciklus, proces se ponavlja.

Analizirati ponašanje sistema za  $T=2$  i  $T=0.2s$

---

Dodatni zadatak

---

```

1: Inicijalizacija:
2: red  $\leftarrow$  PTB18                                 $\triangleright$  Crvena LED
3: blue  $\leftarrow$  PTB19                              $\triangleright$  Plava LED
4: green  $\leftarrow$  PTD1                              $\triangleright$  Zelena LED
5: T  $\leftarrow 2 \times 10^5$                              $\triangleright$  Osnovno vrijeme u mikrosekundama
6:
7: while tačno do
8:   red  $\leftarrow$  1
9:   Sačekaj T  $\mu s$ 
10:  red  $\leftarrow$  0
11:  Sačekaj T/2 $\mu s$ 
12:
13:  blue  $\leftarrow$  1
14:  Sačekaj T  $\mu s$ 
15:  blue  $\leftarrow$  0
16:  Sačekaj T/2 $\mu s$ 
17:
18:  green  $\leftarrow$  1
19:  Sačekaj T  $\mu s$ 
20:  green  $\leftarrow$  0
21:  Sačekaj T/2 $\mu s$ 
22:
23:  for i  $\leftarrow$  0 do 3 do
24:    red  $\leftarrow$  1
25:    Sačekaj T/4 $\mu s$ 
26:    red  $\leftarrow$  0
27:    blue  $\leftarrow$  1
28:    Sačekaj T/4 $\mu s$ 
29:    blue  $\leftarrow$  0
30:  end for
31: end while

```

---



## 3 Implementacija

### 3.1 Zadatak 1 i 1.1

Koraci implementacije:

#### 1. Inicijalizacija pinova:

- Za **Zadatak 1**, koristi se `DigitalOut` za upravljanje jednom LED (`LED1`).
- Za **Zadatak 1.1**, koristi se `BusOut` za upravljanje četiri LED (`LED1` do `LED4`).
- Na `LPC1114ETF` dodatno se mora aktivirati LED izlaz (`LED_ACT`) postavljanjem `ledE = 0` i preimenovati pinove za LED izlaze.

#### 2. Glavna petlja:

- U **Zadatku 1**, LED se pali (`myled = 1`), čeka 1 sekundu (`wait(1)`), gasi i ponovo čeka.
- U **Zadatku 1.1**, koristi se brojač `cnt` za ciklično paljenje LED (`leds[cnt % 4]`), tj. povećavanjem vrijednosti `cnt` i operacijom mod 4, ciklički se uvijek aktivira jedan LED, s pauzom od 1 sekunde između promjena (`wait(1)`).

Korištene funkcije:

- `DigitalOut`, `BusOut`: Upravljanje izlaznim pinovima.
- `wait()`, `ThisThread::sleep_for()`: Osiguravaju vremenske pauze.

### 3.2 Zadatak 2

Koraci implementacije:

#### 1. Inicijalizacija pinova:

- `BusOut` upravlja 8 LED (`LED0–LED7` na `LPC1114ETF` ili `p7–p14` na simulatoru).
- `DigitalIn` čita stanje tastera (`Taster_1`, `Taster_2` na `LPC1114ETF`, `p5` i `p6` u simulatoru).
- Na `LPC1114ETF`, `ledE = 0` aktivira LED izlaze.

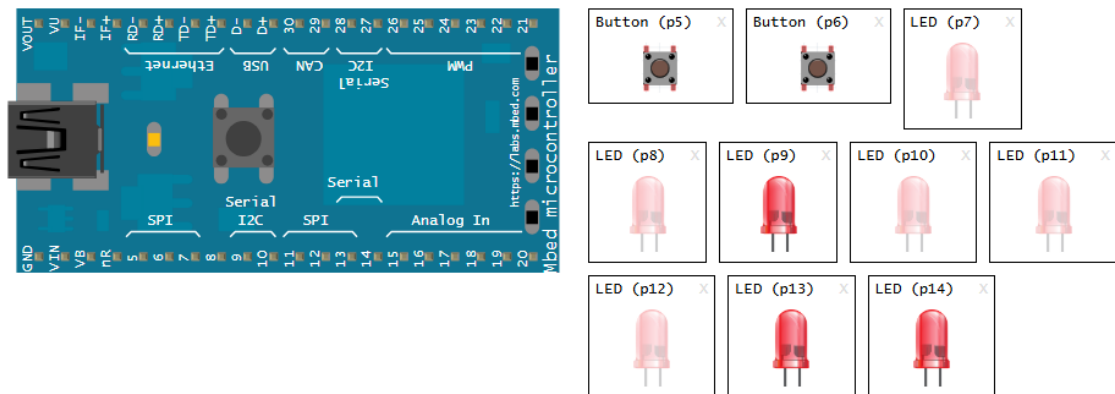
#### 2. Glavna petlja:

- Ako je `button1` pritisnut, smjer brojanja (`dir`) se obrće (`dir = !dir`).
- Ako je `button2` pritisnut, brojač se dekrementira (`cnt--`), inače se mijenja prema `dir`.
- Vrijednost `cnt` se maskira na 8 bita (`cnt &= 0xFF`) i prikazuje na LED (`leds = cnt`).

#### 3. Odgoda: Petlja ima pauzu od 200 ms (`wait_ms(200)`, i `ThisThread::sleep_for(200ms)`).

Korištene funkcije:

- `BusOut`, `DigitalIn`: Upravljanje izlazima/ulazima.
- `wait_ms`, `ThisThread::sleep_for()`: Kontrola brzine izvođenja.



### 3.3 Zadatak 3

Koraci implementacije:

#### 1. Inicijalizacija pinova:

- BusOut upravlja 8 LED.
- Na LPC1114ETF, ledE = 0 aktivira LED izlaze.

#### 2. Glavna petlja:

- Prva petlja (`for (int i = 0; i < 8; i++)`) pomiče svjetlo ulijevo (`leds = (leds << 1) + 1`).
- Druga petlja pomiče svjetlo udesno (`leds = (leds >> 1)`).
- Pauza između koraka je 100 ms (`ThisThread::sleep_for(100ms), wait_ms(100)`).

Korištene funkcije:

- BusOut: Upravljanje LED.
- Bit shift operacije (`<<`, `>>`): Kreiranje animacije.
- `wait_ms()`, `ThisThread::sleep_for()`: Kontrola brzine.

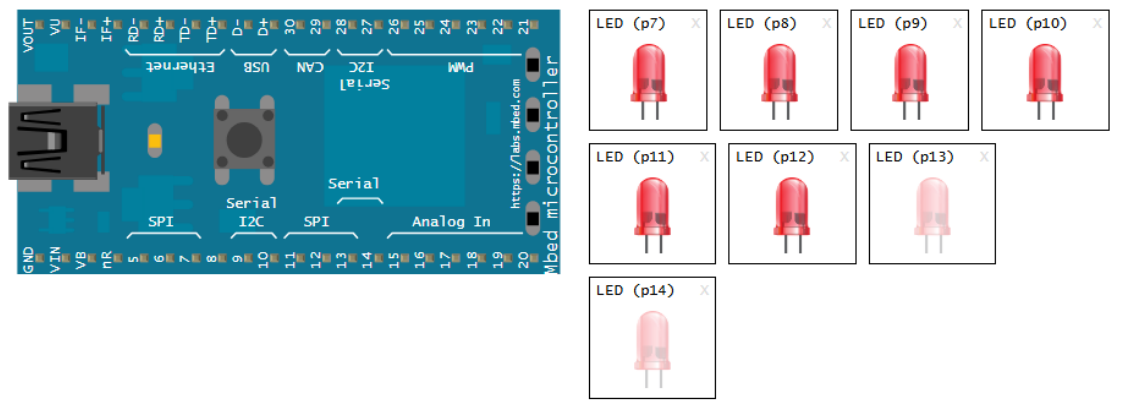
### 3.4 Zadatak 4

Koraci implementacije:

#### 1. Inicijalizacija pinova:

- BusOut za 8 LED, DigitalOut za pojedinačnu LED (LED0 na LPC1114ETF, p7 na simulatoru).
- Na LPC1114ETF, ledE = 0 aktivira LED izlaze.
- DigitalIn čita stanje tastera (Taster\_1, Taster\_2 na LPC1114ETF, p5 i p6 u simulatoru).

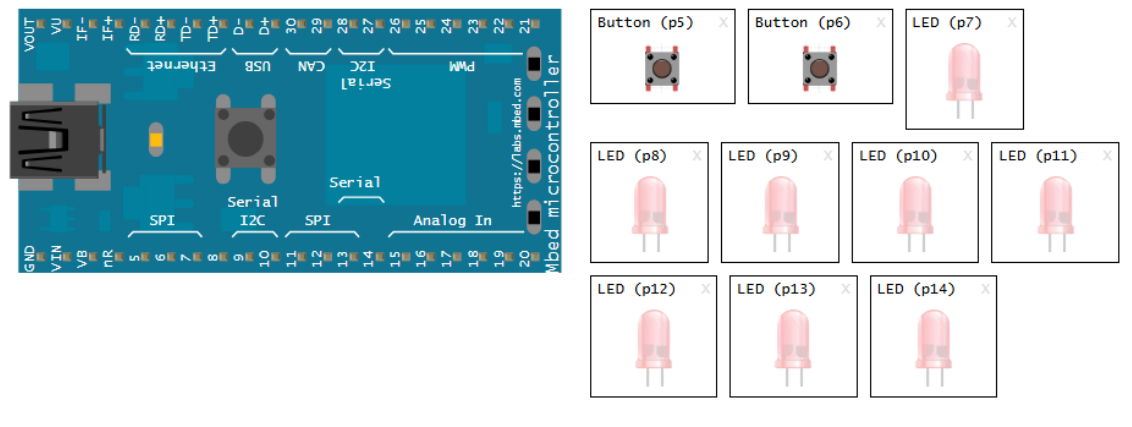
#### 2. Glavna petlja:



- Dok su obje tipke otpuštene (`button1 == 0 && button2 == 0`), led0 treperi (`led0 = !led0`) s intervalom od 500 ms.
- Ovisno o pritisnutoj tipci, poziva se funkcija `trcece()`, implementirana u prethodnom zadatku, s različitim vremenima (100 ms ili 500 ms).

#### Korištene funkcije:

- `BusOut`, `DigitalOut`: Upravljanje LED.
- `wait_us()`: Precizno kontroliranje vremena.



### 3.5 Zadatak 5

#### Koraci implementacije:

##### 1. Inicijalizacija pinova:

- `DigitalOut` za pojedinačnu LED (LED0 na LPC1114ETF, LED1 na simulatoru).

- Na LPC1114ETF, `ledE = 0` aktivira LED izlaze.

## 2. Glavna petlja:

- Prva petlja povećava **on** vrijeme i smanjuje **off** vrijeme. (potreban korak za povećavanje **on**:  $(2T - T)/20 = T/20$ , a za smanjenje **off** vremena:  $(0.2T - T)/20 = -0.8T/20$ )
- Druga petlja smanjuje **on** vrijeme i povećava **off** vrijeme. (potreban korak za smanjenje **on**:  $(0.2T - 2T)/20 = -0.09T$ , a za povećavanje **off** vremena:  $(2T - 0.2T)/20 = 0.09T$ )
- Funkcija `ciklus(on, off)` pali/gasi LED s definiranim vremenima.

### Korištene funkcije:

- `DigitalOut`: Upravljanje LED.
- `wait_us()`: Postavljanje trajanja impulsa.

## 3.6 Dodatni zadatak

### Koraci implementacije:

#### 1. Inicijalizacija pinova:

- `DigitalOut` za crvenu (PTB18), plavu (PTB19), i zelenu (PTD1) LED.

#### 2. Glavna petlja:

- Redom se pali crvena, plava, i zelena LED s pauzom od  $T/2$ .
- Zatim se izvodi brza sekvenca crvena-plava (4 puta).

### Korištene funkcije:

- `DigitalOut`: Upravljanje bojama.
- `wait_us()`: Kontrola trajanja svjetljenja.

## 4 Testiranje i rezultati

Svi zadaci testirani su u dvije faze:

1. **Simulacija u Mbed okruženju:** Programi su prvo implementirani i testirani u online Mbed simulatoru kako bi se osigurala ispravnost logike prije prelaska na stvarni hardver.
2. **Testiranje na stvarnom hardveru:** Nakon uspješne simulacije, programi su preneseni na odgovarajuće razvojne ploče (LPC1114ETF i FRDM-KL25Z) korištenjem ARM Keil Studija.

Testiranje je pokazalo da osnovna funkcionalnost svih zadataka radi kako je zamišljeno, uz neke karakteristične razlike između simulacije i stvarnog hardvera. U zadacima s LED diodama, sekvence paljenja i gašenja su se jasno vidjele na stvarnom hardveru, iako su neki vizualni efekti izgledali malo drugačije nego u simulatoru.

Programi za trčeće svjetlo su na hardveru zahtijevali fino podešavanje vremenskih parametara kako bi efekti izgledali kako treba. RGB dioda na FRDM-KL25Z ploči je jasno prikazivala sve boje, iako su brze promjene bile teže za praćenje golim okom.

Glavni rezultati testiranja pokazuju da:

- Osnovna funkcionalnost svih zadataka je ispravno implementirana
- Vizualni efekti su jasno vidljivi na stvarnom hardveru
- Vremenski parametri se moraju fino podesiti za optimalan izgled efekata

## 5 Zaključak

U okviru ove laboratorijske vježbe postignuti su ključni ciljevi vezani za upravljanje LED diodama i interakciju s tasterima koristeći Mbed simulator i stvarne razvojne sisteme (LPC1114ETF i FRDM-KL25Z). Svaki zadatak temeljio se na osnovnim principima rada s mikrokontrolerima, uključujući upravljanje digitalnim izlazima, implementaciju vremenskih kašnjenja i simulaciju interaktivnih scenarija.

Implementirani su svi zadaci u skladu sa zahtjevima:

- U prvom zadatku uspješno je demonstrirano upravljanje pojedinačnim LED diodama, što je prošireno na sekvencijalno paljenje i gašenje više LED dioda.
- Drugi zadatak obuhvatio je implementaciju binarnog brojača s promjenom smjera brojanja ovisno o stanju tastera.
- U trećem i četvrtom zadatku realizirano je "trčeće svjetlo" s kontrolom brzine putem tastera, što je omogućilo dinamičku interakciju s sistemom.
- Peti zadatak i dodatni zadatak proširili su funkcionalnost na promjenu trajanja osvjetljenosti LED dioda i upravljanje RGB LED-om, čime je demonstrirana fleksibilnost Mbed okruženja.

Iako su svi zadaci uspješno riješeni, postoje mogućnosti za daljnja poboljšanja:

- **Poboljšanje odziva tastera:** Implementacija debounce algoritama za pouzdanije detektiranje pritisaka
- **Preciznije vremenske kontrole:** Korištenje timer modula i prekida umjesto blokirajućih funkcija čekanja
- **Dinamička kontrola parametara:** Mogućnost promjene brzine i načina rada direktno na ploči, bez potrebe za ponovnim prevođenjem programa

Kroz ovu vježbu stečena su sljedeća iskustva:

- Praktično korištenje Mbed simulatora za brzu verifikaciju koda prije prelaska na stvarni hardver.
- Prilagodba koda za različite platforme (LPC1114ETF i FRDM-KL25Z), uz pažnju na specifičnosti pinova i resursa.
- Razumijevanje ograničenja simulacije u odnosu na stvarni hardver, posebno u pogledu vremenske preciznosti i interakcije s komponentama.
- Važnost dokumentacije i referenciranja datasheeta prilikom rada s mikrokontrolerima.

## 6 Prilog

### 6.1 Zadatak 1 i 1.1

Mbed simulator:

```
#include "mbed.h"
DigitalOut myled(LED1); // Upravljanje jednom LED
int main() {
    while(1) {
        myled = 1; // Upali LED
        wait(1);
        myled = 0; // Ugasi LED
        wait(1);
    }
}
```

```
#include "mbed.h"
BusOut leds(LED1,LED2,LED3, LED4); // Grupa od 4 LED
int main() {
    int cnt = 0; // Brojac za odabir LED
    while(1) {
        leds[cnt % 4] = 1; // Upali trenutnu LED
        wait(1);
        leds[cnt % 4] = 0;
        cnt=cnt % 4 + 1; // Sljedeca LED
        wait(1);
    }
}
```

LPC1114ETF:

```
#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT);
DigitalOut myled(LED0);
int main() {
    ledE = 0; myled = 0;
    while(1) {
        myled = 1;
        ThisThread::sleep_for(1s);
        myled = 0;
        ThisThread::sleep_for(1s);
    }
}
```

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT); // Pin za aktivaciju LED izlaza
BusOut leds(LED0,LED1,LED2, LED3);
int main() {
    ledE = 0; leds = 0; // Aktivacija LED izlaza i gasenje svih LED-ova
    int cnt = 0;
    while(1) {
        leds[cnt % 4] = 1;
        ThisThread::sleep_for(1s);
        leds[cnt % 4] = 0;
        cnt=cnt % 4 + 1;
        ThisThread::sleep_for(1s);
    }
}

```

## 6.2 Zadatak 2

Mbed simulator:

```

#include "mbed.h"
BusOut leds(p7, p8, p9, p10, p11, p12, p13, p14);
DigitalIn button1(p5); // Promjena smjera
DigitalIn button2(p6); // Smanjenje broja
int cnt = 0;
bool dir = true; // Smjer brojanja

int main() {
    while(1) {
        if (button1 == 1) {
            dir = !dir; // Promijeni smjer
        }
        if (button2 == 1) {
            cnt--;
        } else {
            cnt += dir ? 1 : -1; // Brojanje po smjeru
        }

        cnt &= 0xFF; // Ogranicenje na 8 bita
        leds = cnt; // Prikaz na LED diodama
        wait_ms(200);
    }
}

```



## LPC1114ETF:

```
#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT); // Pin za aktivaciju LED izlaza
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
DigitalIn button1(Taster_1);
DigitalIn button2(Taster_2);
int cnt = 0;
bool dir = true;

int main() {
    ledE = 0; leds = 0; // Aktivacija LED izlaza i gasenje svih LED-ova
    while(1) {
        if (button1 == 1) {
            dir = !dir;
        }
        if (button2 == 1) {
            cnt--;
        } else {
            cnt += dir ? 1 : -1;
        }

        cnt &= 0xFF;
        leds = cnt;
        ThisThread::sleep_for(200ms);
    }
}
```

## 6.3 Zadatak 3

Mbed simulator:

```
#include "mbed.h"
BusOut leds(p7, p8, p9, p10, p11, p12, p13, p14);
int main() {
    while (1) {
        // Trcece svjetlo ulijevo
        for (int i = 0; i < 8; i++) {
            leds = (leds << 1) + 1; // Pomak lijevo i dodavanje LED
            wait_ms(100);
        }
        // Trcece svjetlo udesno
        for (int i = 0; i < 8; i++) {
            leds = (leds >> 1); // Pomak udesno
            wait_ms(100);
        }
    }
}
```

## LPC1114ETF:

```
#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT); // Pin za aktivaciju LED izlaza
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
int main() {
    ledE = 0; leds = 0; // Aktivacija LED izlaza i gasenje svih LED-ova
    while (1) {
        for (int i = 0; i < 8; i++) {
            leds = (leds << 1) + 1;
            ThisThread::sleep_for(100ms);
        }

        for (int i = 0; i < 8; i++) {
            leds = (leds >> 1);
            ThisThread::sleep_for(100ms);
        }
    }
}
```

## 6.4 Zadatak 4

### Mbed simulator:

```
#include "mbed.h"
BusOut leds(p7, p8, p9, p10, p11, p12, p13, p14);
DigitalIn button1(p5);
DigitalIn button2(p6);

DigitalOut led0(p7); // Indikator cekanja
void trcece(int t){
    // Funkcija za trcece svjetlo
    for (int i = 0; i < 8; i++) {
        leds = (leds << 1) + 1;
        wait_us(t*1000);
    }
    for (int i = 0; i < 8; i++) {
        leds = (leds >> 1);
        wait_us(t*1000);
    }
}
int main() {
    while(1) {
        // Treptanje dok se ceka na taster
        while(button1==0 && button2==0){
            led0 = 0;
            wait_us(500*1000);
            led0 = 1;
            wait_us(500*1000);
        }
        // Pokretanje efekta ovisno o tasteru
    }
}
```

```

        if(button1 == 1){
            trcece(100); // Brza verzija
        } else {
            trcece(500); // Spora verzija
        }
    }
}

```

### LPC1114ETF:

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT); // Pin za aktivaciju LED izlaza
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
DigitalOut led0(LED0);
DigitalIn button1(Taster_1);
DigitalIn button2(Taster_2);

void trcece(int t){
    for (int i = 0; i < 8; i++) {
        leds = (leds << 1) + 1;
        wait_us(t*1000);
    }

    for (int i = 0; i < 8; i++) {
        leds = (leds >> 1);
        wait_us(t*1000);
    }
}

int main() {
    ledE = 0; leds = 0; // Aktivacija LED izlaza i gasenje svih LED-ova
    while(1) {
        while(button1==0 && button2==0){
            led0 = 0;
            wait_us(500*1000);
            led0 = 1;
            wait_us(500*1000);
        }
        if(button1 == 1){
            trcece(100);
        } else {
            trcece(500);
        }
    }
}

```

## 6.5 Zadatak 5

Mbed simulator:

```
#include "mbed.h"
DigitalOut led(LED1);
const float T = 5; // Osnovno vrijeme

void ciklus(int on, int off){
    led = 1; wait_us(on*1000); // LED upaljena
    led = 0; wait_us(off*1000); // LED ugasena
}

int main() {
    while(1) {
        float on = T, off = T;
        for (int i = 0; i <= 20; i++) {
            ciklus(on, off);
            on += T / 20; // Povecaj vrijeme upaljeno
            off -= 0.8 * T / 20; // Smanjuje vrijeme ugaseno
        }
        for (int i = 0; i <= 20; i++) {
            ciklus(on, off);
            off += T * 0.09; // Povecaj vrijeme ugaaseno
            on -= T * 0.09; // Smanjuje vrijeme upaljeno
        }
    }
}
```

LPC1114ETF:

```
#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut ledE(LED_ACT); // Pin za aktivaciju LED izlaza
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7); // Samo u svrhu
    inicijalnog gasenja svih LED-ova
DigitalOut led(LED0);
const float T = 5;

void ciklus(int on, int off){
    led = 1;
    wait_us(on*1000);
    led = 0;
    wait_us(off*1000);
}

int main() {
    ledE = 0; leds = 0; // Aktivacija LED izlaza i gasenje svih LED-ova
    while(1) {
        float on = T, off = T;
        for (int i = 0; i <= 20; i++) {
            ciklus(on, off);
            on += T / 20;
            off -= 0.8 * T / 20;
        }
    }
}
```

```

    }

    for (int i = 0; i <= 20; i++) {
        ciklus(on, off);
        off += T * 0.09;
        on -= T * 0.09;
    }
}
}

```

## 6.6 Dodatni zadatak

### FRDM-KL25Z:

```

#include "mbed.h"
DigitalOut red(PTB18); // Crvena LED
DigitalOut blue(PTB19); // Plava LED
DigitalOut green(PTD1); // Zelena LED

int main() {
    float T = 2*100000; // Osnovno vrijeme ciklusa
    while(1) {
        // Crvena faza
        red = 1; wait_us(T); red = 0;
        wait_us(T / 2);

        // Plava faza
        blue = 1; wait_us(T); blue = 0;
        wait_us(T / 2);

        // Zelena faza
        green = 1; wait_us(T); green = 0;
        wait_us(T / 2);

        // Mjesovita faza (crvena+plava)
        for (int i = 0; i < 4; i++) {
            red = 1; wait_us(T / 4); red = 0;
            blue = 1; wait_us(T / 4); blue = 0;
        }
    }
}

```