

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

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

Ime i prezime: **Mirza Mahmutović**

Broj index-a: **19320**

19. mart 2024. godine

# Sadržaj

1	Pseudokod i/ili dijagram toka .....	1
1.1	Zadatak 1 .....	1
1.2	Zadatak 2 .....	1
1.3	Zadatak 3 .....	2
1.4	Zadatak 4 .....	2
1.5	Zadatak 5 .....	3
2	Analiza programskog rješenja .....	5
2.1	Zadatak 1 .....	5
2.2	Zadatak 2 .....	5
2.3	Zadatak 3 .....	5
2.4	Zadatak 4 .....	5
2.5	Zadatak 5 .....	6
3	Korišteni hardverski resursi .....	6
4	Zaključak .....	7
5	Prilog .....	7
5.1	Zadatak 1/izvorni kod .....	7
5.2	Zadatak 2/izvorni kod .....	8
5.3	Zadatak 3/izvorni kod .....	9
5.4	Zadatak 4/izvorni kod .....	9
5.5	Zadatak 5/izvorni kod .....	11

# 1 Pseudokod i/ili diagram toka

## 1.1 Zadatak 1

```
while (true) do  
    LED1.on()  
    sleep(1)  
    LED1.off()  
  
    LED2.on()  
    sleep(1)  
    LED2.off()  
  
    LED3.on()  
    sleep(1)  
    LED3.off()  
  
    LED4.on()  
    sleep(1)  
    LED4.off()
```

```
end_while
```

## 1.2 Zadatak 2

```
counter = 0  
increment = true  
while (true) do  
    leds = counter  
    if(Taster_1.pressed()) then  
        increment = !increment  
    end_if  
    if(increment) then  
        counter += 1  
    else  
        counter -= 1  
    end_if  
    if(counter < 0) then  
        counter = 255
```

```

        end_if
        if(counter > 255) then
            counter = 0
        end_if
        sleep(1)
    end_while

1.3 Zadatak 3
counter = 1
increment = true
while(true) do
    leds = counter
    if(increment && counter == 64) then
        counter = 255
        increment = false
    else_if(!increment && counter == 1) then
        increment = true
    else_if(increment) then
        counter *= 2
    else
        counter /=2
    end_if
    if(counter < 0) then
        counter = 255
    end_if
    if(counter > 255) then
        counter = 0
    end_if
    sleep(1)
end_while

```

#### 1.4 Zadatak 4

```

counter = 1
loop = false
increment = true
while(true) do
    led1.toggle()
    if(Taster_1.pressed()) then

```

```

        loop = true
        vrijeme = 0.1
else_if(Taster_2.pressed()) then
        loop = true
        vrijeme = 0.5
end_if
while(loop) do
    leds = counter
    if(increment && counter == 64) then
        counter = 255
        increment = false
    else_if(!increment && counter == 1) then
        increment = true
        loop = false
        vrijeme = 0.5
        leds = 0
    else_if(increment) then
        counter *= 2
    else
        counter /=2
    end_if
    if(counter < 0) then
        counter = 255
    end_if
    if(counter > 255) then
        counter = 0
    end_if
    sleep(vrijeme)
end_while
    sleep(vrijeme)
end_while

```

## 1.5 Zadatak 5

```

t = 0.005
onTime = offTime = t, step = ((1.9 * t) - t)/15
counter = 0, limit = 32
increment = true

```

```

while(true) do
    if(counter == 0 && increment == false) then
        increment = true
    else_if(counter == limit && increment == true) then
        increment = false
        limit = 62
        counter = limit
    else_if(counter % 2 == 0 && counter != 0) then
        if(increment) then
            onTime += step
            offTime -= step
        else
            onTime -= step
            offTime += step
        end_if
    end_if
    led1.toggle()
    if(increment) then
        counter++
    else
        counter--
    end_if
    if(counter % 2 != 0) then
        sleep(onTime)
    else
        sleep(offTime)
    end_if
end_while

```

## 2 Analiza programskog rješenja

### 2.1 Zadatak 1

U zadatku je bilo potrebno modificirati početni kod, tako da se redom pale i gase LED1, LED2, LED3 i LED4. Dovoljno je deklarirati 3 dodatne DigitalOut varijable, povezati ih sa odgovarajućim LED diodama, te u *while* petlji dodati isti kod za te 3 diode. Pored toga, da bi se program ispravno izvršavao na LPC1114ETF sistemu, potrebno je uključiti enable signal (LED\_ACT).

### 2.2 Zadatak 2

U zadatku je bilo potrebno realizovati binarni brojač, koji se uvećava za jedan svake sekunde, pri čemu se svakim klikom na taster smjer brojanja treba mijenjati. Kao u prethodnom zadatku, pošto se radi sa LED diodama, prije svega je potrebno uključiti enable signal. Ovdje je također znatno praktičnije koristiti jednu BusOut varijablu za svih 8 LED dioda, umjesto 8 nezavisnih DigitalOut varijabli. Postavlja se početna vrijednost brojača na 0, te se ulazi u beskonačnu *while* petlju. Unutar petlje se neprekidno provjerava da li je pritisnut taster *Taster\_1*, te se na osnovu toga mijenja smjer kretanja brojača. Pored toga, potrebno je provjeravati da li se brojač nalazi u dozvoljenom opsegu vrijednosti.

### 2.3 Zadatak 3

U zadatku je bilo potrebno pomoću 8 LED dioda realizovati tzv. „trčeće svjetlo“. Za lakšu realizaciju postavljenog zadatka, opet se koristi BusOut varijabla koja predstavlja svih 8 LED dioda. Također su potrebne varijable za predstavljanje brojača i smjera kretanja. U beskonačnoj *while* petlji se zatim na osnovu trenutne vrijednosti brojača podešava njegova sljedeća vrijednost. Da bi se upalila sljedeća LED dioda, a ugasila trenutna, dovoljno je brojač pomnožiti, odnosno podijeliti sa 2, u zavisnosti od smjera kretanja. Pri tome, treba imati dodatan uslov za regulisanje paljenja svih dioda odjednom, te mijenjanja smjera.

### 2.4 Zadatak 4

Zadatak 4 predstavlja nadogradnju zadatka 3. Da bi se omogućilo paljenje i gašenje LED0 diode dovoljno je brojač postaviti na vrijednost 1, te u beskonačnoj *while* petlji mijenjati stanje LED0 diode. Da bi se registrovao pritisak tastera, potrebne su dvije DigitalIn varijable. Prepoznavanjem pritiska tastera se *loop* varijabla postavlja na vrijednost *true*, što označava da je vrijeme za realizaciju „trčećeg svjetla“, te program ulazi u unutrašnju *while* petlju. U zavisnosti od pritisnutog tastera postavlja se vrijednost za vrijeme izmjene. Kada se završi

„trčeće svjetlo“, *loop* varijabla se postavlja na *false*, što predstavlja izlaz iz unutrašnje *while* petlje, a vrijeme izmjene na početno. Logika kretanja „trčećeg svjetla“ je identična kao u prethodnom zadatku.

## 2.5 Zadatak 5

Za rješavanje petog zadatka potrebno je na osnovu početnog vremena  $T$  izračunati korak po kojem se trebaju mijenjati  $t1$  (vrijeme uključenosti) i  $t2$  (vrijeme isključenosti). Korištene su i 3 varijable *counter*, *limit*, *increment*. Ove varijable redom predstavljaju: broj puta koliko se prošlo kroz *while* petlju, broj puta koliko se treba proći kroz *while* petlju, te da li se brojač smanjuje (smanjuje se vrijeme uključenosti, a povećava vrijeme isključenosti) ili povećava (smanjuje se vrijeme isključenosti, a povećava vrijeme uključenosti). Dva prolaska kroz *while* petlju traju vrijeme  $T$ , jedan prolazak predstavlja uključeno stanje, dok drugi prolazak predstavlja isključeno stanje. U svakom drugom prolasku kroz petlju (nakon  $T$  sekundi),  $t1$  i  $t2$  mijenjaju svoje vrijednosti. Ovim se osigurava da se nakon  $30T$  sekundi dostigne vrijeme uključenosti od  $1.9T$  sekundi, a vrijeme isključenosti od  $0.1T$  sekundi. Kada se dostigne *limit*, *limit* se povećava za 30, te se *increment* postavlja na *false*. Ovo je potrebno uraditi da bi se nakon dostignutog limita u suprotnom smjeru (vrijednost varijable *counter* dostigne 0) postiglo vrijeme uključenosti od  $0.1T$  sekundi, a vrijeme isključenosti od  $1.9T$  sekundi. Nakon završetka ove faze, vrijednost varijable *limit* se više ne mijenja, te se samo mijenja stanje varijable *increment*.

## 3 Korišteni hardverski resursi

Na ovoj vježbi su korišteni sljedeći razvojni sistemi:

- *LPC1114ETF*, baziran na mikrokontroleru *NXP LPC1114FN28*
- *Freescale FRDM-KL25Z*, baziran na mikrokontroleru *Freescale KL25Z128VLK*.

Sistem *LPC1114ETF* posjeduje razne elemente, a za ovu vježbu su korišteni:

- 8x LED
- 2x TASTERA

Na sistemu *FRDM-KL25Z* je korištena 1x RGB LED dioda.



## 4 Zaključak

Ova vježba je za cilj imala da se studenti upoznaju sa Mbed OS, te razvojem aplikacija za mikrokontrolere sa nekim od *ARM Cortex-M* jezgri. U tu svrhu su korišteni sljedeći razvojni alati:

- Online Mbed simulator
- ARM Keil Studio (cloud verzija)

Pored toga studenti su upoznati sa sljedećim razvojnim sistemima:

- LPC1114ETF, koji ima jezgro ARM Cortex-M0
- FRDM-KL25Z, koji ima jezgro ARM Cortex-M0+

Uz programski zahtjevnije zadatke, upoznali smo se sa načinom programiranja na ovim sistemima, te sa njihovim raspoloživim resursima. Svi zadaci su uspješno urađeni u predviđenom vremenu.

## 5 Prilog

### 5.1 Zadatak 1/izvorni kod

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

DigitalOut E(LED_ACT);
DigitalOut led1(LED0);
DigitalOut led2(LED1);
DigitalOut led3(LED2);
DigitalOut led4(LED3);

int main() {
    E=0;
    while(1) {
        led1 = 1;
        wait_us(1000000);
        led1 = 0;
        wait_us(1000000);

        led2 = 1;
```

```

        wait_us(1000000);

        led2 = 0;

        wait_us(1000000);

        led3 = 1;

        wait_us(1000000);

        led3 = 0;

        wait_us(1000000);

        led4 = 1;

        wait_us(1000000);

        led4 = 0;

        wait_us(1000000);
    }
}

```

## 5.2 Zadatak 2/izvorni kod

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut E(LED_ACT);
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
DigitalIn btn(Taster_1);
int main() {
    E=0;
    int counter = 0;
    bool increment = true;
    while (1) {
        leds = counter;
        if(btn == 1)
            increment = !increment;
        if(increment)
            counter++;
        else
            counter--;
        if(counter < 0) counter = 255;
        if(counter > 255) counter = 0;
        wait_us(1000000);
    }
}

```

```

    }
}

```

### 5.3 Zadatak 3/izvorni kod

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut E(LED_ACT);
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
int main() {
    E=0;
    int counter = 1;
    bool increment = true;
    while (1) {
        leds = counter;
        if(increment && counter == 64) {
            counter = 255;
            increment = false;
        }
        else if(!increment && counter == 1)
            increment = true;
        else if(increment)
            counter *= 2;
        else
            counter /= 2;
        if(counter < 0) counter = 255;
        if(counter > 255) counter = 0;
        wait_us(100000);
    }
}

```

### 5.4 Zadatak 4/izvorni kod

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut E(LED_ACT);
DigitalOut led1(LED0);
DigitalIn taster_1(Taster_1);
DigitalIn taster_2(Taster_2);
BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);

```

```

int vrijeme = 500000;
int main() {
    E=0;
    int counter = 1;
    bool loop = false, increment = true;
    while (1) {
        led1 = !led1;
        if(taster_1) {
            loop = true;
            vrijeme = 100000;
        }
        else if(taster_2) {
            loop = true;
            vrijeme = 500000;
        }
        while(loop) {
            leds = counter;
            if(increment && counter == 64){
                counter = 255;
                increment = false;
            }
            else if(!increment && counter == 1) {
                increment = true;
                loop = false;
                vrijeme = 500000;
                leds = 0;
            }
            else if(increment)
                counter *= 2;
            else
                counter /= 2;
            if(counter < 0) counter = 255;
            if(counter > 255) counter = 0;
            wait_us(vrijeme);
        }
        wait_us(vrijeme);
    }
}

```

```

    }
}

```

## 5.5 Zadatak 5/izvorni kod

```

#include "mbed.h"
#include "lpc1114etf.h"
DigitalOut E(LED_ACT);
DigitalOut led1(LED0);
BusOut leds(LED1, LED2, LED3, LED4, LED5, LED6, LED7);
int main() {
    E=0;
    double t = 5000;
    double t1 = t, t2 = t, step = ((1.9 * t) - t)/15;
    int counter = 0, limit = 32;
    bool increment = true;
    while (1) {
        if(counter == 0 && increment == false) increment = true;
        else if(counter == limit && increment == true) {
            increment = false;
            limit = 62;
            counter = limit;
        }
        else if(counter % 2 == 0 && counter != 0) {
            if(increment) {
                t1 += step;
                t2 -= step;
            }
            else {
                t1 -=step;
                t2 +=step;
            }
        }
        led1 = !led1;
        if(increment) counter++;
        else counter--;
        if(counter % 2 != 0) wait_us(t1);
        else wait_us(t2);
    }
}

```

}  
}