

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: **Haris Mališević**
Broj Indeksa: **19328**

12.03.2024.

Sadržaj

1 Pseudokod / dijagram toka.....	2
1.1 Zadatak 1.....	2
1.2 Zadatak 2.....	2
1.3 Zadatak 3.....	3
1.4 Zadatak 4.....	4
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
3 Korišteni hardverski resursi.....	6
4 Zaključak.....	6
5 Prilog.....	7
5.1 Zadatak 1.....	7
5.2 Zadatak 2.....	7
5.3 Zadatak 3.....	9
5.4 Zadatak 4.....	10

1 Pseudokod / dijagram toka

1.1 Zadatak 1

main():

Beskonačna petlja:

 Za svaki LED od 0 do 3:

 Upali LED[i].

 Sačekaj 1 sekundu.

 Ugasli LED[i].

 Sačekaj 1 sekundu.

1.2 Zadatak 2

funkcija **increment():**

 Carry = 1.

 Za svaki pin u nizu LED-ova:

 Current = pin.value().

 Ako je Current jednako 1 i carry jednako 1:

 Postavi pin na 0.

 Inače:

 Definiši integer tmp i postavi ga na zbir carry i current.

 Postavi pin na tmp.

 Postavi carry na 0.

funkcija **decrement():**

 Definiši integer index i postavi ga na 0.

 Dok je index manji od 8:

 Ako je leds[index] jednako 1:

 Prekini petlju.

 Inače:

 Inkrementiraj index.

 Ako index nije jednak 8:

 Postavi leds[index] na 0.

 Za svaki brojač i od 0 do index (isključujući index):

 Postavi leds[i] na 1.

funkcija **changeDirection():**

 Promeni vrednost countUp na suprotnu vrednost.

main():

Beskonačna petlja:

Ako je countUp istinito:

Pozovi funkciju increment().

Inače:

Pozovi funkciju decrement().

Sačekaj 1 sekundu.

Ako je pritisnut taster btn:

Pozovi funkciju changeDirection().

1.3 Zadatak 3

funkcija **runningLight** (pauseTime_ms):

Definiši index i postavi ga na -1.

Dok je index + 1 manje od NUM_LEDS:

Isključi LED na poziciji index.

Uključi LED na poziciji index + 1.

Inkrementiraj index.

Sačekaj pauseTime_ms mikrosekundi.

Postavi index na NUM_LEDS - 1.

Za svaki brojčani indeks i od 0 do NUM_LEDS - 1:

Uključi LED na poziciji i.

Sačekaj pauseTime_ms mikrosekundi.

Dok je index veći ili jednak 0:

Isključi LED na poziciji index.

Dekrementiraj index.

Sačekaj pauseTime_ms mikrosekundi.

main():

Beskonačna petlja:

runningLight(200 000)

1.4 Zadatak 4

Koristimo istu **runningLight** funkciju iz zadatka 3, uz izmjenu **main** funkcije:

main():

Beskonačna petlja:

Dok su pritisnuti taster1 i taster2:

Vrti prazno.

Ako se pritisne taster1:

runningLight(100 000)

Inače Ako se pritisne taster2:

runningLight(500 000)

Inače:

LED[0].toggle()

Sačekaj 0.5 sekundi.

2 Analiza programskog rješenja

2.1 Zadatak 1

Ovaj zadatak je iziskivao jednostavno ponavljanje paljenja i gašenja LED-ova u ciklusu i rješenje je dosta jednostavno. Prolazi se kroz svaki od 4 pina te se pale na 1 sekund pa gase na 1 sekund.

2.2 Zadatak 2

Drugi zadatak zahtjeva implementaciju binarnog inkrementiranja na 8 LED-ova uz mogućnost dekrementiranja nakon pritiska tastera. Pogodno je iskoristiti funkcije *increment()* i *decrement()* sa prethodne laboratorijske vježbe. U beskonačnoj petlji, provjerava se u kojem stanju se nalazi program *countUp* boolean varijablom te se poziva odgovarajuća funkcija. Ova varijabla mijenja stanje kad god je taster pritisnut.

2.3 Zadatak 3

Zadatak broj tri je tražio implementaciju trčećeg svjetla. Funkcija koja provodi ovo ponašanje prima dužinu pauze kao parametar i provodi jedan ciklus *trčanja*. Main funkcija beskonačnom petljom kontinuirano poziva trčanje sa traženom pauzom.

2.4 Zadatak 4

Ovaj zadatak traži korisnički unos kroz mogućnost pritiska na dva tastera. Jedan taster poziva trčeće svjetlo sa pauzom od 0.1 sekundi dok drugi taster poziva trčeće svjetlo sa pauzom od pola sekunde.

Treptajući LED0 koji obavještava korisnika da treba izvršiti unos bio je nova funkcionalnost sa kojom smo se susreli i iziskuje preciznost prilikom klika na taster. Potrebno je isprobati još nekoliko načina da pritisak na neki od tastera ima intuitivniji odziv.

3 Korišteni hardverski resursi

Za ovu laboratorijsku vježbu korišteni razvojni sistemi FRDM-KL25Z i LPC1114ETF. Oni imaju integrisane sve elemente koji su korišteni.

Korišteni integrisani elementi su:

- LED (8x)
- Tasteri (2x)

Kako su navedeni elementi dio same ploče razvojnog sistema, ovdje se neće ulaziti u detalje njihove implementacije.

Korišteni alati uključuju:

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

4 Zaključak

Druga laboratorijska vježba za cilj je imala upoznavanje sa Mbed OS-om i novim razvojnim sistemima. U tom cilju smo bili uspješni te je većina ponuđenih zadataka uspješno urađena.

Mbed se čini kao veoma moćna C/C++ biblioteka i alat za razvoj ali dolazi u jak kontrast sa MicroPython-om sa prethodne vježbe. Keil Studio je veoma intuitivan i pristupačan za korištenje, za razliku od Online Mbed simulatora koji je dosta *grub* alat.

5 Prilog

5.1 Zadatak 1

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

DigitalOut E(LED_ACT);

DigitalOut led1(LED1);
DigitalOut led2(LED2);
DigitalOut led3(LED3);
DigitalOut led4(LED4);

DigitalOut leds[] = {led1, led2, led3, led4};

int main() {
    E = 0;
    while (1) {
        for (int i = 0; i < 4; i++) {
            leds[i] = 1;
            wait_us(1000000);
            leds[i] = 0;
            wait_us(1000000);
        }
    }
}
```

5.2 Zadatak 2

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

DigitalOut E(LED_ACT);

DigitalOut led1(LED0), led2(LED1), led3(LED2), led4(LED3),
led5(LED4), led6(LED5), led7(LED6), led8(LED7);

DigitalOut LEDs[] = {led1, led2, led3, led4, led5, led6, led7,
led8};
```



```

DigitalIn Btn(Taster_1);

bool countUp = true;

void increment() {
    int carry = 1;

    for (auto &pin : LEDs) {
        int current = pin;
        if (current == 1 && carry == 1) {
            pin = 0;
        } else {
            int tmp = carry + current;
            pin = tmp;
            carry = 0;
        }
    }
}

void decrement() {
    int index = 0;

    while (index < 8) {
        if (LEDs[index] == 1) {
            break;
        } else {
            index++;
        }
    }

    if (index != 8) {
        LEDs[index] = 0;
    }

    for (int i = 0; i < index; i++) {
        LEDs[i] = 1;
    }
}

void changeDirection() {

```

```

        countUp = !countUp;
    }

    int main() {
        E = 0;

        while (true) {
            if (countUp)
                increment();
            else
                decrement();

            wait_us(1000000);

            if (Btn == 1)
                changeDirection();
        }
    }
}

```

5.3 Zadatak 3

```

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

DigitalOut E(LED_ACT);

DigitalOut led1(LED0), led2(LED1), led3(LED2), led4(LED3),
led5(LED4), led6(LED5), led7(LED6), led8(LED7);

DigitalOut LEDs[] = {led1, led2, led3, led4, led5, led6, led7,
led8};

int NUM_LEDS = 8;

void runningLight(int pauseTime_ms) {
    int index = -1;
    while (index + 1 < NUM_LEDS) {
        LEDs[index] = 0;
        LEDs[index + 1] = 1;
        index++;
    }
}

```

```

        wait_us(pauseTime_ms);
    }

    index = NUM_LEDS - 1;

    // Svi LED-ovi su uključeni
    for (int i = 0; i < NUM_LEDS; i++) {
        LEDs[i] = 1;
    }

    wait_us(pauseTime_ms);

    while (index >= 0) {
        LEDs[index] = 0;
        index--;
        wait_us(pauseTime_ms);
    }
}

int main() {
    E = 0;
    while (true)
        runningLight(200000);
}

```

5.4 Zadatak 4

```

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

DigitalOut E(LED_ACT);

DigitalOut led1(LED0), led2(LED1), led3(LED2), led4(LED3),
led5(LED4),
        led6(LED5), led7(LED6), led8(LED7);

DigitalOut LEDs[] = {led1, led2, led3, led4, led5, led6, led7,
led8};

int NUM_LEDS = 8;

```

```

DigitalIn btn1(Taster_1);
DigitalIn btn2(Taster_2);

DigitalOut SignalLED(LED0);

void runningLight(int pauseTime_ms) {
    int index = -1;

    while (index + 1 < NUM_LEDS) {
        LEDs[index] = 0;
        LEDs[index + 1] = 1;
        index++;
        wait_us(pauseTime_ms);
    }

    index = NUM_LEDS - 1;

    // Svi LED-ovi su uključeni
    for (int i = 0; i < NUM_LEDS; i++) {
        LEDs[i] = 1;
    }

    wait_us(pauseTime_ms);

    while (index >= 0) {
        LEDs[index] = 0;
        index--;
        wait_us(pauseTime_ms);
    }
}

int main() {

    E = 0;

    while (true) {
        while (btn1 && btn2) {
            wait_us(100000);
        }
    }
}

```

```
    if (btn1) {  
        runningLight(100000);  
    } else if (btn2) {  
        runningLight(500000);  
    } else {  
        SignalledLED = !SignalledLED;  
        wait_us(500000);  
    }  
}  
}
```