

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: **Ivona Jozić**

Broj indeks-a: **19357**

19. mart 2024.

Sadržaj

1 Pseudokod i / ili dijagram toka.....	3
1.1 Zadatak 1	3
1.2 Zadatak 2	3
1.3 Zadatak 3	3
1.4 Zadatak 4	3
1.5 Zadatak 5	4
1.6 Zadaci za dodatne bodove	4
1.6.1 Zadatak 1	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
2.5 Zadatak 5	6
2.6 Zadaci za dodatne bodove	6
2.6.1 Zadatak 1	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.....	7
5.3 Zadatak 3 / izvorni kod.....	8
5.4 Zadatak 4 / izvorni kod.....	9
5.5 Zadatak 5 / izvorni kod.....	10
5.6 Zadaci za dodatne bodove	10
5.6.1 Zadatak 1 / izvorni kod.....	10

1 Pseudokod i / ili dijagram toka

1.1 Zadatak 1

```
while(1)
    for i in 1..4
        upali(led_i)
        sacekaj(1 sekunda)
        ugasi(led_i)
        sacekaj(1 sekunda)
```

1.2 Zadatak 2

Led podrazumijeva BusOut tip objekta

brojac -> 0

```
while(1)
    led -> brojac
    if(taster1 pritisnut)
        promijeni_smjer
        sacekaj(vrijeme)
    if(smjer=povecavaj)
        brojac ++
    else
        brojac --
    if(brojac_izvan_opsega)
        podesi_ispravan_opseg()
```

1.3 Zadatak 3

```
while(1)
    for i in 0..8
        led -> 2^i
        sacekaj(0.1 sekundu)

    led->255
    sacekaj(0.1 sekundu)

    for i in 8..0, step -1
        led -> led - 2^i
        sacekaj(0.1 sekundu)
```

1.4 Zadatak 4

```
while(1)
    while(taster1_nije_pritisnut AND taster2_nije_pritisnut)
        led -> 1
        sacekaj(0.5 sekundi)
        led -> 0
        sacekaj(0.5 sekundi)
```

```

        if(taster1 pritisnut)
            trajanje -> 0.1 sekundu
        if(taster2 pritisnut)
            trajanje -> 0.5 sekundi

    for i in 0..8
        led -> 2^i
        sacekaj(trajanje)

    led -> 255
    sacekaj(trajanje)

    for i in 8..0, step -1
        led -> led - 2^i
        sacekaj(trajanje)

```

1.5 Zadatak 5

```

T -> 0.005 sekundi
vrijeme_on -> T
smjer -> true

while(1)
    led -> 1
    sacekaj(vrijeme_on)
    led -> 0
    sacekaj(2 * T - vrijeme_on)

    if(smjer = true AND |vrijeme_on - 1.9 * T| < 10-16)
        smjer -> false
    else if(smjer = false AND |vrijeme_on - 0.1 * T| < 10-16)
        smjer -> true

    if(smjer = true)
        vrijeme_on -> vrijeme_on + 3/50*T
    else
        vrijeme_on -> vrijeme_on - 3/50*T

```

1.6 Zadaci za dodatne bodove

1.6.1 Zadatak 1

U postavci zadatka se zahtijeva implementacija koja traži umijeće paralelnog programiranja što još uvijek nismo u mogućnosti pružiti, pa je za ovaj zadatak samo uvršten kod u prilogu pod oznakom 5.6.1.

2 Analiza programskog rješenja

2.1 Zadatak 1

U prvom zadatku je bilo potrebno modificirati priloženi kod iz postavke vježbe tako da se naizmjenično pale prve četiri diode na način da je uvijek upaljena samo jedna u trajanju od 1s, a zatim su prije paljenja sljedeće sve diode ugašene u trajanju od 1s. Dioda se pale u redoslijedu LED1, LED2, LED3, te LED4. Zahtijevano rješenje je implementirano uz pomoć jednog objekta tipa BusOut, beskonačne while petlje i jedne for petlje čiji se brojač multiplicira u stepene dvice, a zatim prikazuje na diodama.

2.2 Zadatak 2

Cilj drugog zadatka je bio implementacija binarnog brojača koji se inkrementira za 1 svake sekunde, a čije se stanje prikazuje na LED diodama LED0 – LED7. Dodatna funkcionalnost koju ovaj binarni brojač treba imati je promjena smjera brojanja, odnosno mogućnost dekrementiranja za 1 u slučaju da dođe do pritiska na taster. U slučaju da je brojač bio u dekrement modu, ponovni pritisak na taster ga vraća u inkrement modul rada. Ovo rješenje je, kao prethodno, implementirano pomoću objekta tipa BusOut klase u kojem su inicijalizirane diode, te jedne dodatne varijable increment, koja je zamišljena da funkcionira kao logički toggle koji se mijenja između dvije vrijednosti ukoliko je taster pritisnut.

2.3 Zadatak 3

U trećem zadatku je implementirano „trčeće“ svjetlo, što podrazumijeva paljenje po jedne od LED dioda u smjeru sa lijeva na desno, zatim paljenje svih dioda, te postepeno gašenje po jedne LED diode u smjeru sa desna na lijevo. Period između dvije promjene treba trajati 0.1 sekundu. Implementacija je ponovno izvedena pomoću objekta tipa BusOut koje olakšava parsiranje i prikaz 8-bitnog broja pomoću LED dioda. Pored spomenutog objekta, korištene su i dvije for petlje gdje prva služi za „trčeće“ svjetlo sa lijeva na desno i unutar koje se objekat tipa BusOut „shifta“ kao stepen broja 2, da bi se redom palile jedna po jedna LED dioda. Nakon što se upali posljednja LED dioda, objekat BusOut se postavlja na vrijednost 255, što za posljedicu ima paljenje svih LED dioda. Sljedeći korak je od broja 255 redom oduzimati stepeni broja 2 u opadajućem poretку kroz drugu for petlju da bi se postigao efekt postepenog gašenja jedne po jedne diode sa desna na lijevo.

2.4 Zadatak 4

U četvrtom zadatku je postavljeni zahtjev bio veoma sličan onom postavljenom u trećem zadatku, tako da je osnovna ideja ostala ista. Modifikacija u odnosu na prethodni zadatak je u tome što je implementaciju potrebno proširiti sa dva dodatna tastera čiji pritisak će odrediti vrijeme izmjene LED dioda. U slučaju da je pritisnut taster 1 izmjena se dešava na svakih 0.1 sekundu, dok u slučaju pritiska na taster 2, izmjena dolazi nakon svakih 0.5 sekunde. Još jedna od izmjena u odnosu na prethodni zadatak je to što prije nego se pokrene „trčeće“ svjetlo sa nekim od perioda izmjene, LED0 dioda treba da se pali i gasi sa naizmjeničnim trajanjem od 0.5 sekunde, a tek nakon pritiska na neki od tastera „trčeće“ svjetlo treba da se aktivira.

Kako se zadatak idejno ne razlikuje u odnosu na treći, tako ni implementaciono ne postoje velike razlike. Naime, jedina razlika je u tome što for petljama prethodi jedna while petlja koja

pali i gasi LED0 dok se ne desi pritisak na jedan od tastera, a zatim u zavisnosti od toga koji taster je pritisnut postavlja vrijeme izmjene na odgovarajuću vrijednost. Nakon što „trčeće“ svjetlo prođe jedan ciklus paljenja sa lijeva na desno i gašenja sa desna na lijevo, ponovno počinje da blinka LED0 čekajući na ponovni pritisak jednog od tastera.

2.5 Zadatak 5

U petom zadatku je bila zahtijevana implementacija paljenja i gašenja diode u trajanju od dva ciklusa po 30 perioda, čije je trajanje varijabilno. U prvom ciklusu je zahtijevano da se vrijeme u kojem dioda gori postepeno povećava od T (period) do $1.9T$, a vrijeme ugašenosti diode isto tako postepeno smanjuje od T do $0.1T$. U drugom ciklusu je trebalo postići suprotno, to jeste da se vrijeme u kojem dioda gori postepeno smanjuje, a ono u kojem je dioda ugašena postepeno povećava. Za dovoljno mali period, na LED diodi se primjećuje efekt „disanja“. Implementirano rješenje je zasnovano na podjeli perioda od $30T$ na ΔT vrijeme za koje se dužina trajanja upaljene diode treba povećavati, odnosno smanjivati. Ove akcije dodatno ovise o vrijednosti varijable „smjer“ koja definira da li se vrijeme koje dioda provodi upaljena treba povećavati ili smanjivati.

2.6 Zadaci za dodatne bodove

2.6.1 Zadatak 1

Implementacija ovog zadatka zasnovana je na veoma sličnoj ideji kao i implementacija petog zadatka, s tim što je došlo do promjene razvojnog sistema (korišten je FRDM-KL25Z) i njegova RGB LED dioda. Za razliku od petog zadatka, sekvencijalno su korištene boje R, G, B, te neke od njihovih kombinacija, sa različitim trajanjima perioda T . Iako ovo nije korektno rješenje zadanog problema kakvo je traženo kroz postavku laboratorijske vježbe, s obzirom da ono zahtijeva umijeće paralelnog programiranja, i dalje se mogu uočiti neki efekti i promjene stanja RGB LED diode.

3 Korišteni hardverski resursi

Za potrebe laboratorijske vježbe 2 korišteni su razvojni sistemi LPC1114ETF i FRDM-KL25Z. Svi korišteni elementi su integrisani na ovim sistemima, a oni podrazumijevaju:

Za sistem LPC1114ETF:

- 8 LED dioda
- 2 fizička tastera

Za sistem FRDM – KL25Z:

- RGB LED dioda

4 Zaključak

Prilikom izvođenja Laboratorijske vježbe 2 nije bilo problema osim sporijeg rada zbog susreta sa novim razvojnim okruženjima, LPC1114ETF i FRDM-KL25Z. Prilikom same implementacije rješenja nije bilo poteškoća zahvaljujući ranijem radu u programskom jeziku C++. Kompaktnosti i pojednostavljenju rješenja je doprinijelo i postojanje objekta tipa BusOut. Cilj vježbe je bio upoznavanje sa dodatnim alatima i sistemima poput Mbed simulatora, ARM Keil razvojnog okruženja, te samih LPC1114ETF i FRDM-KL25Z, što je uspješno postignuto.

5 Prilog

U prilogu su dati kodovi koje se izvršavaju na razvojnim okruženjima LPC1114ETF i FRDM-KL25Z (zadatak za dodatne bodove). U slučaju da želimo kodove pokretati u Mbed simulatoru, potrebno je samo promijeniti deklaraciju varijabli (dodati LED diode i tastere, te podesiti odgovarajuće pinove prema simulatoru) i izostaviti varijablu E tipa DigitalOut koja je enable signal za diode.

5.1 Zadatak 1 / 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;
    leds=0;

    while(1) {
        for(int i=1; i<16; i*=2){
            leds=i;
            wait_us(1e6);
            leds = 0;
            wait_us(1e6);
        }
    }
}
```

5.2 Zadatak 2 / izvorni kod

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

BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
DigitalIn taster1(Taster_1);
```

```

int main(){
    int brojac=0;
    bool povecaj=1;

    while(1){
        leds=brojac;

        if(taster1==1){
            povecaj=!povecaj; //promjena smjera brojanja

            wait_us(200000);
        }
        if(povecaj) brojac++;
        else brojac--;

        if(brojac>255) brojac=0;
        if(brojac<0) brojac=255;

        wait_us(1000000);
    }
}

```

5.3 Zadatak 3 / izvorni kod

```

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

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

int main() {
    E = 0;
    leds = 0;

    while (true) {
        for(int i=0; i<=8; ++i) { //petlja za trcece svjetlo sa lijeva na desno
            leds = static_cast<int>(std::pow(2, i));
            wait_us(100000);
        }
        leds = 255; //paljenje svih dioda
        wait_us(100000);
        for(int i=8; i>=0; --i) { //petlja za trcece svjetlo sa desna na lijevo
            leds = leds - static_cast<int>(std::pow(2, i));
            wait_us(100000);
        }
    }
}

```


5.4 Zadatak 4 / izvorni kod

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

BusOut leds(LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7);
DigitalIn taster1(Taster_1);
DigitalIn taster2(Taster_2);

DigitalOut E(LED_ACT);

int main() {
    E = 0;
    leds = 0;

    int trajanje_t1 = 100000;
    int trajanje_t2 = 500000;
    int trajanje=0;

    while(true) {
        while(taster1 == 0 && taster2 == 0) { //led0 blinka do pritiska na neki od tastera
            leds = 1;
            wait_us(500000);
            leds = 0;
            wait_us(500000);
            if(taster1==1) trajanje=trajanje_t1;
            else trajanje=trajanje_t2;
        }

        for(int i=0; i<=8; ++i) { //petlja za trcece svjetlo sa lijeva na desno
            leds = static_cast<int>(pow(2, i));
            wait_us(trajanje);
        }

        leds = 255; //paljenje svih dioda
        wait_us(trajanje);
        for(int i=8; i>=0; --i) { //petlja za trcece svjetlo sa desna na lijevo
            leds = leds - static_cast<int>(pow(2, i));
            wait_us(trajanje);
        }
    }
}
```

5.5 Zadatak 5 / izvorni kod

```
#include "mbed.h"
#include "lpc1114etf.h"
#include <cmath>

DigitalOut led(LED0);
DigitalOut E(LED_ACT);

int main() {
    E = 0;
    double T = 0.005;
    double on = T;
    bool smjer = true;

    while(1) {
        led = 1;
        wait_us(on * 1e6);
        led = 0;
        wait_us((2*T-on)*1e6);

        if(smjer && std::abs(on-1.9*T) < 1e-16) smjer = false;
        else if(!smjer && std::abs(on - 0.1*T) < 1e-16) smjer = true;

        if(smjer) on += 3./50*T;
        else on -= 3./50*T;
    }
}
```

5.6 Zadaci za dodatne bodove

5.6.1 Zadatak 1 / izvorni kod

```
#include "mbed.h"

#define T_red 1
#define T_green 0.5
#define T_blue 0.3

DigitalOut red(PTB18);
DigitalOut green(PTB19);
DigitalOut blue(PTB1);

int main() {
    float red_on = T_red;
    float red_off = T_red;

    float green_on = T_green;
    float green_off = T_green;
```

```
float blue_on = T_blue;
float blue_off = T_blue;

const float deltaT_red = 0.9/(30*T_red);
const float deltaT_green = 0.9/(30*T_green);
const float deltaT_blue = 0.9/(30*T_blue);

while(1) {
    red_on = red_off = T_red;
    green_on = green_off = T_green;
    blue_on = blue_off = T_blue;

    while(red_on <= 1.9*T_red) {
        red = 1;
        wait_us(red_on*(1e6));
        red = 0;
        wait_us(red_off*(1e6));

        red_on += deltaT_red;
        red_off -= deltaT_red;
    }

    while(green_on <= 1.9*T_green) {
        green = 1;
        wait_us(green_on*(1e6));
        green = 0;
        wait_us(green_off*(1e6));

        green_on += deltaT_green;
        green_off -= deltaT_green;
    }

    while(blue_on <= 1.9*T_blue) {
        blue = 1;
        wait_us(blue_on*(1e6));
        blue = 0;
        wait_us(blue_off*(1e6));

        blue_on += deltaT_blue;
        blue_off -= deltaT_blue;
    }

    while(red_off <= 1.9*T_red) {
        red = 1;
        wait_us(red_on*(1e6));
        red = 0;
        wait_us(red_off*(1e6));
```

```
        red_on -= deltaT_red;
        red_off += deltaT_red;
    }

    while(green_off <= 1.9*T_green) {
        green = 1;
        wait_us(green_on*(1e6));
        green = 0;
        wait_us(green_off*(1e6));

        green_on -= deltaT_green;
        green_off += deltaT_green;
    }

    while(blue_off <= 1.9*T_blue) {
        blue = 1;
        wait_us(blue_on*(1e6));
        blue = 0;
        wait_us(blue_off*(1e6));

        blue_on -= deltaT_blue;
        blue_off += deltaT_blue;
    }
}
```