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: **Ismar Muslić**
Broj index-a: **19304**

19. mart 2024. godine

Sadržaj

1 Pseudokod

U narednom segmentu izvještaja dati su pseudokodovi za redovne zadatke sa LV2. Iako je Zadatak za dodatne bodove 1 implementiran, uspješno kompajliran i pokrenut na razvojnom sistemu FRDM – KL25Z, s obzirom na to da je njegova tačna implementacija izvan okvira trenutnog znanja studenata, pseudokod za ovo programsko rješenje je u ovom dijelu izostavljen. U prilogu je dat kod implementacije ovog programskog rješenja sa kojom se predmetni asistent složio, međutim, s obzirom na to da ta implementacija nije korektna implementacija koja je od studenata zahtijevana, to je još jedan dodatni razlog zašto je pseudokod izostavljen.

1.1 Zadatak 1

```
while(true)
      for(i = 1 to 4)
        led_i = 1;
        sleep(1s)
        led_i = 0;
        sleep(1s);
```

1.2 Zadatak 2

```
brojac = 0;
increment = true;
while(true)
      diode = brojac;

      if(taster pritisnut)
        increment.toggle();
        blokiraj_taster();

      if(increment = true)
        brojac++;
      else brojac--;

      if(brojac izvan opsega)
        vrati_brojac_u_opseg();

      sleep(1s);
```

1.3 Zadatak 3

```
Bus diode;

while(true)
      for(i = 0 to 8)
        diode = 2^i;
        sleep(0.1s)

      diode = 255;
      sleep(0.1s)

      for(i = 8 to 0 step -1)
        diode -= 2^i;
        sleep(0.1s)
```

1.4 Zadatak 4

```
Bus diode;
diode = 0;

trajanje1 = 0.1s;
trajanje2 = 0.5s;
trajanje = 0;

while(true)
      while(taster1 iskljucen AND taster2 iskljucen)

            diode = 1;
            sleep(0.5s);
            diode = 0;
            sleep(0.5s);

            if(taster1 pritisnut)
              trajanje = trajanje1;
            else trajanje = trajanje2;

            for(i = 0 to 8)
              diode = 2^i;
              sleep(trajanje);

            diode = 255;
            sleep(trajanje);

            for(i = 8 to 0 step -1)
              diode -= 2^i;
              sleep(trajanje);
```

1.5 Zadatak 5

```
Bus diode;

T = 0.005;
vrijeme_on = T;
smjer = true;

while(true)
      diode = 1;
      sleep(vrijeme_on);
      diode = 0;
      sleep(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 += 3/50*T;
      else vrijeme_on -= 3/50*T;
```

2 Analiza programskog rješenja

2.1 Zadatak 1

Prvi zadatak na ovoj vježbi je zahtijevao sekvencijalno paljenje i gašenje dioda LED0 – LED3 sa međutrajanjem od 1 sekunde u kojem su diode ugašene. Konkretna implementacija je urađena pomoću BusOut klase u kojoj se LED diode inicijaliziraju, te for – petlje u kojoj se množeći varijablu sa 2, led bus efektivno shifta ulijevo, što rezultuje paljenjem i gašenjem dioda u zahtijevanom redoslijedu.

2.2 Zadatak 2

U zadatku 2 traženo rješenje je implementacija binarnog brojača koji se automatski inkrementira svake sekunde, uz dodatnu funkcionalnost da se na pritisak tastera brojač prebaci u mod dekrementiranja. Ovo rješenje je implementirano ponovno pomoću objekta BusOut klase u kojem su inicijalizirane diode, te jedne dodatne varijable increment, koja funkionira kao logički toggle koji se mijenja između dvije vrijednosti ukoliko je taster pritisnut. Konkretno, program kreće inkrementiranjem brojača, te se na pritisak tastera 1 ovaj smjer mijenja i program tada umanjuje brojač za 1. Ponovnim pritiskom taster, program se vraća u mod inkrementiranja brojača.

2.3 Zadatak 3

Zadatak 3 traži implementaciju „trčećeg svjetla“ koje u jednom smjeru „trči“ od početka do kraja. Kada stigne do kraja, potrebno je upaliti sve LED diode te ih zatim u suprotnom smjeru gasiti sve dok ne budu sve ugašene. Zahtijevano međutrajanje je 0.1s. Implementacija se temelji na binarnoj reprezentaciji brojeva koje LED diode prikazuju, te je stoga program na elegantan način iskodiran ponovno koristeći BusOut objekat. Dvije for – petlje su korištene u implementaciji od kojih prva služi za „trčeće“ svjetlo u uzlaznom smjeru, gdje se korišteni Bus objekat „shifta“ kao stepen broja 2, da bi se redom palile jedna po jedna LED dioda u uzlaznom smjeru. Kada se stigne do zadnje diode, BusOut objekat se postavlja na vrijednost 255, što znači da će sve diode biti upaljene. Zatim se od 255 redom oduzimaju stepeni broja 2 u drugoj for – petlji da bi se dobio efekat postepenog gašenja jedne po jedne diode u silaznom smjeru. Konkretna mbed OS implementacija je u prilogu.

2.4 Zadatak 4

Zadatak 4 predstavlja nadogradnju zadatka 3, gdje je potrebno dodati funkcionalnosti za 2 tastera. Sistem treba da čeka na aktivaciju prvog ili drugog tastera u odnosu na šta diode rade isti efekat kao u prethodnom zadatku, sa međutrajanjem od 0.1s ili od 0.5s, respektivno. Implementirano rješenje se zasniva na dodatnoj while() petlji koja predstavlja waiting mode u kojem se dioda LED0 pali i gasi, te se čeka na na pritisak tastera. Na osnovu toga koji je taster pritisnut, izvršava se jednak kod kao u prethodnom zadatku, sa modificiranim međutrajanjem.

2.5 Zadatak 5

U zadatku 5 bilo je potrebno implementirati paljenje i gašenje diode u zadanim vremenskim periodima, sa varijabilnom brzinom promjene vremena za koje je dioda upaljena, odnosno ugašena. Za dovoljno malen period T, na LED diodi se dobija efekat

„disanja“. Implementirano je rješenje zasnovano na podjeli perioda od 30T (koji je specificiran tekstom zadatka) na deltaT vrijeme za koje se dužina trajanja upaljene diode treba uvećavati, odnosno smanjivati. Postepenim najprije uvećavanjem trajanja upaljenog svjetla, a skraćivanjem trajanja za koje je svjetlo ugašeno, a zatim izvršavanjem obrnutog slučaja, dobija se željeni efekat. Ove akcije dodatno ovise o vrijednosti varijable „smjer“ koja definira da li treba da se izvodi prvi ili drugi od navedenih slučajeva. U daljnje detalje nećemo ulaziti s obzirom da je kod u mbed – u dat u prilogu izvještaja.

2.6 Dodatni zadatak 1

Implementacija Zadatka za dodatne bodove 1 je urađena na sličan način kao u zadatku 5, s tim da su sekvencijalno korištene boje R, G, B, te neke od njihovih kombinacija, sa različitim trajanjima perioda T. Iako ovo nije korektno rješenje zadanog problema, predmetni asistent se sa njim složio s obzirom na to da je korektno rješenje izvan okvira znanja studenata, jer ono zahtijeva znanje iz područja paralelnog programiranja. Kod implementacije je također u prilogu.

3 Korišteni hardverski resursi

U okviru ove laboratorijske vježbe korišteni su razvojni sistemi LPC1114ETF i FRDM – KL25Z. Korišteni integrisani elementi u ovim sistemima uključuju:

Za sistem LPC1114ETF:

- 8x LED diode
- 2 fizička tastera

Za sistem FRDM – KL25Z:

- RGB LED.

4 Zaključak

U okviru ove vježbe upoznali smo se sa novim razvojnim sistemima LPC1114ETF, te FRDM – KL25Z. Nije bilo poteškoća prilikom implementacije programskih rješenja, s obzirom na to da je mbed OS header file za programski jezik C++, što je studentima veoma blizak programski jezik. Još jedno od olakšanja u odnosu na prošlu vježbu jeste mogućnost korištenja sabirnice kao objekta klase BusOut, da bi se kao višebitni digitalni izlaz lakše koristilo više LED – ova u nizu. Cilj vježbe je bio upoznavanje sa dodatnim alatima i sistemima kao što su Mbed simulator, ARM Keil razvojno okruženje, te sami sistemi LPC1114ETF i FRDM – KL25Z, što je uspješno postignuto.

5 Prilog

Napomenuo bih da je kod za Mbed simulator jednak kao onaj dat u nastavku, s tim da je jedina razlika u imenovanju digitalnih ulaza i izlaza, kao i potrebi za definisanjem LED_ACT varijable na sistemu LPC1114ETF, kako bi se pinovi prebacili u mod rada za LED diode, umjesto za digitalni I/O.

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 taster(Taster_1);

DigitalOut E(LED_ACT);

int main() {

    E = 0;
    leds = 0;

    int brojac = 0;
    bool increment = true;

    while(1) {
        leds = brojac;

        if(taster == 1) {
            increment = !increment;
            wait_us(20000);
        }
```

```
        if(increment)
            ++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) {
            leds = static_cast<int>(std::pow(2, i));
            wait_us(100000);
        }
        leds = 255;
        wait_us(100000);
        for(int i=8; i>=0; --i) {
            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) {
            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) {
            leds = static_cast<int>(pow(2, i));
            wait_us(trajanje);
        }

        leds = 255;
        wait_us(trajanje);
        for(int i=8; i>=0; --i) {
            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 Zadatak za dodatne bodove 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;
        }
    }
}
```