

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Кафедра вычислительной техники

Лабораторная работа № 2 по дисциплине
"Информационно-управляющие системы"

Вариант 4

Выполнили:

Айтуганов Д. А.

Чебыкин И. Б.

Группа: Р3401

Проверяющий: Пинкевич В. Ю.

Санкт-Петербург, 2017

Задание

Контроллер SDK-1.1 циклически проигрывает нисходящую гамму нот второй октавы (длительность каждой ноты – 1 секунда) и на линейку светодиодов выводит количество замыканий входа Т1 (счетный вход таймера 1 на рис. 5). Подсчет количества замыканий входа должен быть реализован с помощью таймера-счетчика 1. В результате выполнения работы должны быть разработаны драйверы системного таймера, таймера-счетчика, звукового излучателя, светодиодных индикаторов.

Модель взаимодействия

Исходный код

```

#include <timer.h>
#include <aduc812.h>
#include <stdint.h>
#include <max.h>
#include <led.h>

#define OCTAVE 2
#define FREQ 11059000
#define CYCLE 12 // pulses
#define PERIOD (FREQ / CYCLE)
#define PERIOD2 (FREQ / CYCLE / 2)

const uint16_t notes[] = {
493 * OCTAVE,
440 * OCTAVE,
391 * OCTAVE,
349 * OCTAVE,
329 * OCTAVE,
293 * OCTAVE,
261 * OCTAVE,
};

unsigned long ms_count;
unsigned long last_note_swap;

uint8_t current_ena;
uint8_t current_note;

uint8_t note_delay[2];
uint8_t time_delay[2];

void set_vector(unsigned char xdata* address, void* vector);
void set_ena(uint8_t value);

void compute_note_delay(uint8_t index);
void compute_timer_delay();

void set_vector(uint8_t xdata* address, void* vector) {
    unsigned char xdata* tmp;

    *address = 0x02;

    tmp = (unsigned char xdata*)(address + 1);
    *tmp = (unsigned char)((unsigned short)vector >> 8);
    ++tmp;

    *tmp = (unsigned char)vector;
}

void set_ena(uint8_t value) {
    write_max(ENA, value);
}

void time2_handler() interrupt(5) {
    TH2 = time_delay[1];
    TL2 = time_delay[0];

    ms_count++;

    if(DTimeMs(last_note_swap) >= 1000) {
        leds(TL1);
        last_note_swap = GetMsCounter();
        compute_note_delay(current_note++);
        if(current_note == 7) {
            current_note = 0;
        }
    }
}

```

```

TF2 = 0;
}

void note_handler() interrupt(1) {
    TH0 = note_delay[1];
    TL0 = note_delay[0];
    current_ena = ~current_ena;
    set_ena(current_ena);
}

void InitTimer() {
    last_note_swap = 10;
    current_ena = 0x18;

    compute_note_delay(0);
    compute_timer_delay();

    ms_count = 0;

    TH0 = 0xFF;
    TL0 = 0xFF;

    TH1 = 0x0;
    TL1 = 0xFE;

    TH0 = note_delay[1];
    TL0 = note_delay[0];

    TCON = 0x50;
    TMOD = 0x41;

    TH2 = time_delay[1];
    TL2 = time_delay[0];

    RCAP2L = TL2;
    RCAP2H = TH2;
    T2CON = 0x84;

    set_vector(0x200B, (void*)note_handler);
    set_vector(0x202B, (void*)time2_handler);

    ET0 = 1;
    ET1 = 1;
    ET2 = 1;
    EA = 1;
}

unsigned long GetMsCounter(void) {
    return ms_count;
}

unsigned long DTimeMs(unsigned long t0) {
    return ms_count - t0;
}

void DelayMs(unsigned long t) {
    unsigned long target = t + ms_count;
    while(ms_count != target);
}

void compute_note_delay(uint8_t index) {
    uint16_t delay = PERIOD2 / notes[index];
    uint8_t* ptr = (uint8_t*)&delay;
    note_delay[0] = 0xFF - ptr[0];
    note_delay[1] = 0xFF - ptr[1];
}

void compute_timer_delay() {
    uint16_t delay = PERIOD / 1000;
    uint8_t* ptr = (uint8_t*)&delay;

```

```
time_delay[0] = 0xFF - ptr[0];  
time_delay[1] = 0xFF - ptr[1];  
}
```

Вывод

В результате выполнения лабораторной работы был получен опыт работы с таймерами и системой прерываний в SDK-1.1.