

Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1
по дисциплине «Встроенные системы»

Выполнил: Ларочкин Г.
Шуст И.
Тарасенко Д.
Группа: Р3400
Преподаватель: Ключев А.О.

Санкт-Петербург
2020 г.

Описание виртуальной лабораторной

Виртуальная лаборатория основана на модели реального лабораторного стенда SDK-1.1M, которая позволяет познакомиться с основами разработки ПО для встроенных систем. Сам лабораторный стенд расположен в виде веб-приложения <http://itmo.vpswork.ru/esim>.

У виртуальной модели стенда есть возможность переключать switch, нажимать на кнопки, наблюдать за светодиодами. Для работы необходимо написать код на языке C++, при этом начало программы должно находиться в функции main, положить его в архив и загрузить на виртуальный стенд. Далее происходит компиляция и линковка разработанной программы с объектным файлом виртуального стенда.

Состав:

- Процессорный модуль на базе STM32F407VGT
- Материнская плата
- Кнопка включения
- Кнопка сброса
- Дисплей
- Матричная клавиатура
- Линия переключателей
- Линия светодиодов
- Цветные светодиоды

Отличия от реального стенда

- Упрощена работа с портами ввода/вывода процессорного модуля
- Скорость реакции на события пользователя - от 100мс.
- Включение/выключение питания выполнено в виде кнопки, вместо шнура питания

Описание задачи

В случае установки на SW переключателях кода N (см. вариант задания) на светодиодные индикаторы LED1 ... LED8 должна выводиться анимация согласно варианту задания. Во всех остальных случаях светодиодные индикаторы отражают значение, выставленное на SW переключателях.

По кнопке nBTN процессорного модуля должна осуществляться приостановка анимации. Одно нажатие – анимация приостанавливается, следующее нажатие – анимация продолжается с того же момента. Когда анимация не выводится на светодиоды, нажатие на кнопку игнорируется.

Состояния светодиодов процессорного модуля в процессе работы программы:

Условие	Светодиод VD6	Светодиод VD7
Вывод анимации на линейку светодиодов модуля расширения	Горит зеленым	Не горит
Остановка по кнопке nBTN	Не горит	Горит красным
Режим ввода состояний SW переключателей	Не горит	Горит желтым

N = 0x8 (шестнадцатеричное значение)

Кадр	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8
1								
2								
3								
4								
5								
6								
7								
8								

Зеленым отмечено состояние, когда соответствующий светодиод горит. В противном случае – не горит, то есть выключен.

Анимация выводится циклически, то есть после вывода последнего кадра анимации она начинается сначала – с первого кадра. Время показа одного кадра – 0.5 с.

Выполнение

Далее представлен код для виртуальной лабораторной работы. Основная логика выполняется в функции *umain*, где происходит проверка на нажатие кнопки, подсветка светодиодов в указанной в задании комбинации, анимация на светодиодном индикаторе согласно варианту.

В функции *read_sw* проверяет состояние переключателей и возвращает его значение.

В функции *check_button* происходит проверка на нажатие кнопки pBTN. Важно отметить, что для корректной работы данной функции пришлось добавить механизм для *button threshold* для того, чтобы избежать повторное нажатие, если долго держать нажатой кнопку.

В функции *animate* происходит изменение анимации на светодиодном индикаторе в течение 0.5 секунд, при этом происходит переход анимации в следующее состояние. Если анимация находится в последнем состоянии, то происходит переход в первое состояние.

```

#include "hal.h"
enum btn_state
{
    STOP,
    START,
};

unsigned int read_sw();
void animate(unsigned int &animation_state);
void check_button(btn_state &state_btn);

unsigned int led_num[] = {GPIO_PIN_3, GPIO_PIN_4, GPIO_PIN_5, GPIO_PIN_6};
unsigned int sw_num[] = {GPIO_PIN_4, GPIO_PIN_8, GPIO_PIN_10, GPIO_PIN_12};
unsigned int led_state[8][2] = {
    {GPIO_PIN_3, GPIO_PIN_12},
    {GPIO_PIN_3, GPIO_PIN_4},
    {GPIO_PIN_4, GPIO_PIN_5},
    {GPIO_PIN_5, GPIO_PIN_6},
    {GPIO_PIN_6, GPIO_PIN_8},
    {GPIO_PIN_8, GPIO_PIN_9},
    {GPIO_PIN_9, GPIO_PIN_11},
    {GPIO_PIN_11, GPIO_PIN_12}};

int umain()
{
    unsigned int animation_state = 0;
    btn_state state_btn = START;
    while (1)
    {
        unsigned int state_sw = read_sw();
        if (state_sw == GPIO_PIN_4 /* 0x8 code */)
        {
            check_button(state_btn);
            if (state_btn == STOP)
            {
                // red light
                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);

                for (int i = 0; i < 4; i++)
                {
                    HAL_GPIO_WritePin(GPIOD, led_num[i], GPIO_PIN_RESET);
                }
                continue;
            }

            // green light
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);

```

```

        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
        animate(animation_state);
    }
    else
    {
        // yellow light
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
        for (int i = 0; i < 4; i++)
        {
            if (state_sw & sw_num[i])
                HAL_GPIO_WritePin(GPIOD, led_num[i], GPIO_PIN_SET);
            else
                HAL_GPIO_WritePin(GPIOD, led_num[i], GPIO_PIN_RESET);
        }
    }
}

return 0;
}

unsigned int read_sw()
{
    unsigned int result = 0;
    for (int i = 0; i < 4; i++)
    {
        GPIO_PinState state = HAL_GPIO_ReadPin(GPIOE, sw_num[i]);
        if (state == GPIO_PIN_SET)
            result |= sw_num[i];
    }

    return result;
}

void animate(unsigned int &animation_state)
{
    if (animation_state == 8)
        animation_state = 0;

    unsigned int first_pin = led_state[animation_state][0];
    unsigned int second_pin = led_state[animation_state][1];
    animation_state++;

    HAL_GPIO_WritePin(GPIOD, first_pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, second_pin, GPIO_PIN_SET);
    HAL_Delay(500); // delay 0.5s
    HAL_GPIO_WritePin(GPIOD, first_pin, GPIO_PIN_RESET);

```

```

        HAL_GPIO_WritePin(GPIOD, second_pin, GPIO_PIN_RESET);
    }

void check_button(btn_state &state_btn)
{
    GPIO_PinState state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
    if (state == GPIO_PIN_RESET)
    {
        if (state_btn == START)
            state_btn = STOP;
        else if (state_btn == STOP)
            state_btn = START;

        // button tresholding
        while (state == GPIO_PIN_RESET)
        {
            state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
        }
    }
}

```

Вывод

В ходе данной лабораторной работы были изучены основные возможности виртуального стенда *esim*. Также были получены базовые навыки работы со светодиодами, переключателями и кнопками nBTN на модели стенда SDK-1.1M.