

Университет ИТМО

Встроенные системы
Лабораторная работа №1

Выполнили: Калугина Марина
 Саржевский Иван
 Группа: Р3402

г. Санкт-Петербург

2020 г.

Описание виртуальной лабораторной

Виртуальная лаборатория основана на модели реального лабораторного стенда SDK-1.1M.

Предоставляет возможность ознакомиться с основами разработки ПО для встроенных систем. Клиентская часть выполнена в виде веб-приложения, расположенного по адресу <http://itmo.vpswork.ru/esim>.

Для начала рассмотрим окружение.

В интерфейсе пользователя есть интерактивная модель стенда, на которой можно переключать переключатели, нажимать на кнопки и смотреть индикацию на светодиодах, поле для вывода сообщений консоли и терминала в процессе работы системы, поле для загрузки архива с ПО и кнопка для компиляции проекта.

Для работы необходимо написать исходный код на языке C++, положить его в архив и отправить на компиляцию на сервер. Там объектный файл пользовательского кода объединяется с объектным файлом модели и получается исполняемая модель системы с программой пользователя, которая будет запускаться на виртуальном стенде.

Функция `main` скрыта в ядре моделирования, мы используем функцию `umain`. Необходимо подключить заголовочный файл `hal.h`, который предоставляет объявление функции для использования стенда, подключается он на сервере. Все исходники кладутся в один `zip`-архив и загрузить его в интерфейс приложения и нажать кнопку `compile`. В консоль выведется статус компиляции. При успешной компиляции можно подать питание на виртуальный стенд и программа начнет исполняться.

Можно сохранять результаты моделирования в качестве лога и вывод консоли, для этого в интерфейсе предусмотрены специальные кнопки.

Теперь рассмотрим подробнее сам виртуальный лабораторный стенд.

Состав:

- * Материнская плата
- * Процессорный модуль на базе микроконтроллера STM32F407VGT
- * Кнопка включения
- * Кнопка сброса
- * Дисплей
- * Матричная клавиатура
- * Линейка переключателей
- * Модуль расширения с линейкой светодиодов и переключателями

Отличия от реального стенда:

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

Схема подключения портов ввода-вывода аналогична реальному стенду для большинства устройств.

Модуль расширения содержит 4 переключателя (SW1...SW4) и 8 светодиодов (LED1...LED8), подключается к разъемам XS2, XS3, XS4, XS5.

Работа с портами ввода-вывода осуществляется путем чтения и записи в специальные области памяти, куда отображается пространство ввода-вывода. У микроконтроллера есть 5 16-рядных IO-портов общего назначения: GPIOA, GPIOB, GPIOC, GPIOD, GPIOE.

На микросхеме пины портов называются PA0-PA15...PE0-PE15. В виртуальном стенде порты, подключенные к светодиодам автоматически настроены как выходные порты, а подключенные к переключателям и кнопкам - как входные. Для работы с ними существует абстракция, предоставляемая библиотекой HAL.

Задание

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 анимации по кнопке nBTN	Не горит	Горит красным
Режим вывода состояний SW переключателей	Не горит	Горит желтым

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

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.

Исходный код

Далее представлен код для виртуальной лабораторной.

В функции `main` находится основная логика: вывод анимации на лампочках при включенном втором переключателе и дублирование значение на переключателях, если второй переключатель выключен. При этом первом случае должен гореть зеленый светодиод, во втором – желтый.

В функции `checknBTN` после каждого переключения анимации происходит проверка кнопки. Если кнопка нажата, то анимация должна приостановиться до повторного нажатия кнопки. При этом должен гореть красный светодиод.

```
#include "hal.h"
```

```
GPIO_PinState state;
```

```

// По нажатию на кнопку должна останавливаться анимация и загораться светодиод
void checknBTN(){
    // Если кнопка сначала имела один стейт, а потом другой -- значит ее нажали
    GPIO_PinState statel = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
    if (statel != state) {
        // светодиод VD7 загорается красным, остальные гореть не должны
        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);
        state = statel;
        HAL_Delay(100);
        statel = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
        HAL_Delay(100);
        // Ожидание повторного нажатия кнопки
        while((state == statel)) {
            statel = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
            HAL_Delay(100);
        }
        // После повторного нажатия кнопки возвращаем зеленый светодиод и выключаем
        красный
        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);
    }
}

int umain(){
    int i = 0, delay = 500;
    // Описание лампочек и переключателей
    int leds_num[] =
{GPIO_PIN_3,GPIO_PIN_4,GPIO_PIN_5,GPIO_PIN_6,GPIO_PIN_8,GPIO_PIN_9,GPIO_PIN_11,GPIO
_PIN_12};
    unsigned int sw_num[] = {GPIO_PIN_4, GPIO_PIN_8, GPIO_PIN_10, GPIO_PIN_12};

    // Сохранение изначального стейта
    state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);

    // Начало основной логики программы
    while(1){

        // Если включен второй переключатель - проигрывается анимация
        if(HAL_GPIO_ReadPin(GPIOE, sw_num[1])){

            // Зажечь зеленый светодиод
            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);

            // Начало анимации: попарно загораются лампочки, проверяется, не нажата
            ли кнопка, ставится задержка в 500 мс, гаснут лампочки, задержка, проверка кнопки
            for (i = 0; i < 4; i++){
                HAL_GPIO_WritePin(GPIOD, leds_num[i%4], GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOD, leds_num[i%4 + 4], GPIO_PIN_SET);
                checknBTN();
                HAL_Delay(delay);
                checknBTN();
                HAL_GPIO_WritePin(GPIOD, leds_num[i%4], GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOD, leds_num[i%4 + 4], GPIO_PIN_RESET);
            }
        }
    }
}

```

```

        checknBTN();
        HAL_Delay(delay);
        checknBTN();
    }

    // Вторая часть анимации: тоже самое, только лампочки зажигаются в
    обратном порядке
    for (i = 2; i >= 0; i--){
        HAL_GPIO_WritePin(GPIOD, leds_num[i%4], GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOD, leds_num[i%4 + 4], GPIO_PIN_SET);
        checknBTN();
        HAL_Delay(delay);
        checknBTN();
        HAL_GPIO_WritePin(GPIOD, leds_num[i%4], GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOD, leds_num[i%4 + 4], GPIO_PIN_RESET);
        checknBTN();
        HAL_Delay(delay);
        checknBTN();
    }

    } else {

        // Если не включен второй переключатель, на лампочках надо дублировать
        переключатели

        // Загорается желтый светодиод
        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);

        // Проверяем все переключатели, кроме второго
        for (i = 0; i < 4; ++i) {
            // Если второй переключатель - пропускаем
            if (i == 1) {
                continue;
            }

            // Проверяем значение на переключателях и зажигаем/гасим лампочку в
            зависимости от значения
            if(HAL_GPIO_ReadPin(GPIOE, sw_num[i])){
                HAL_GPIO_WritePin(GPIOD, leds_num[i], GPIO_PIN_SET);
            } else {
                HAL_GPIO_WritePin(GPIOD, leds_num[i], GPIO_PIN_RESET);
            }
        }

    }

    }

    return 0;
}

```

Вывод

В ходе выполнения лабораторной работы были изучены основные возможности виртуального стенда. Была произведена работа над лампочками, переключателями, pBTN кнопкой и светодиодами виртуального лабораторного стенда.