

Университет ИТМО

Лабораторная работа №1
по предмету
“Встроенные системы”

Выполнил:
Студент группы
Р3410
Глушков Дима

Санкт-Петербург,
2020 г.

Введение:

Данная лабораторная работа предназначена для получения практических навыков работы с портами ввода-вывода на виртуальном стенде на базе микрон роллера STM32F407VGT с использованием HAL функций.

Описание используемого виртуального стенда:

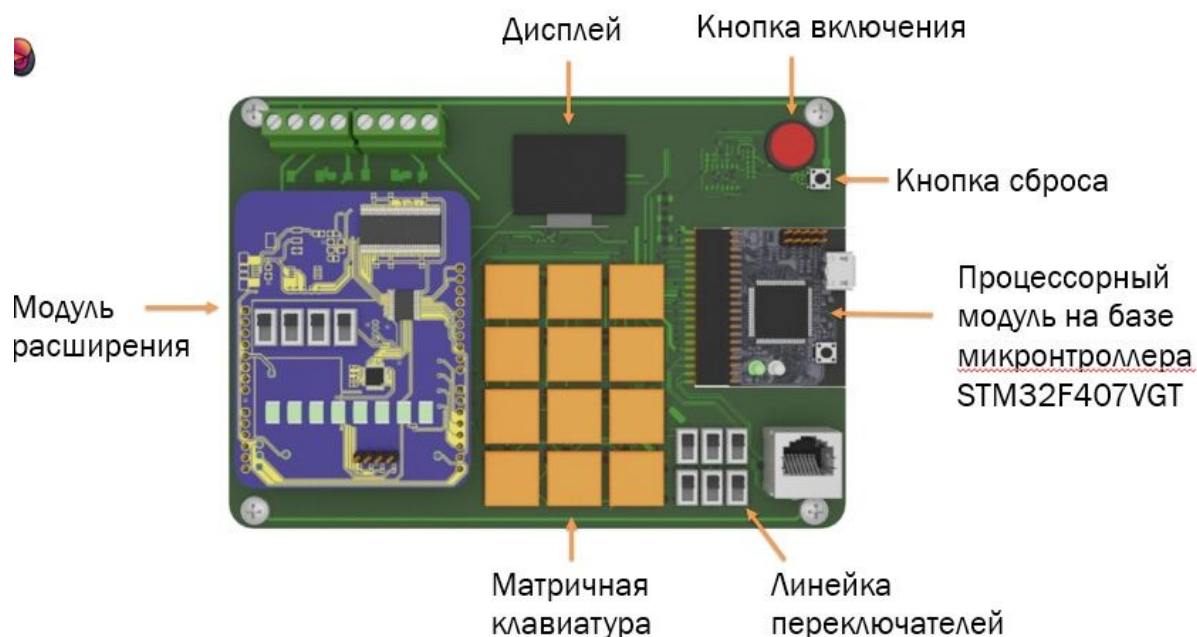


Рис. 1. Виртуальный стенд с описанием доступных модулей.

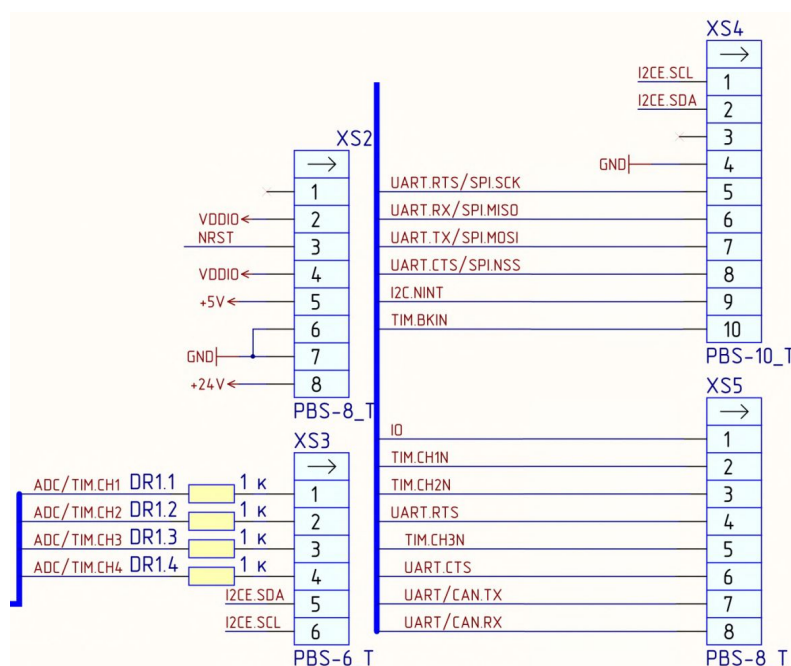


Рис. 2. Схема модуля расширения.

Функции HAL, которые могут потребоваться при выполнении лабораторной работы:

```
void HAL_GPIO_TogglePin(uint gpio_port, uint pin);
GPIO_PinState HAL_GPIO_ReadPin(uint gpio_port, uint pin);
void HAL_GPIO_WritePin(uint gpio_port, uint pin, GPIO_PinState pin_state);
void HAL_Delay(uint delay_ms);
```

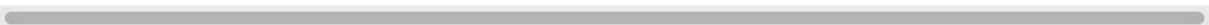
Задание:

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

<  >

N = 0x1 (шестнадцатеричное значение).

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

```
#include "hal.h"

int i = 0;
int delay = 500;
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};

int animation_check()
{
    GPIO_PinState switch1state = HAL_GPIO_ReadPin(GPIOE, sw_num[0]);
    GPIO_PinState switch2state = HAL_GPIO_ReadPin(GPIOE, sw_num[1]);
    GPIO_PinState switch3state = HAL_GPIO_ReadPin(GPIOE, sw_num[2]);
    GPIO_PinState switch4state = HAL_GPIO_ReadPin(GPIOE, sw_num[3]);
    if ((switch1state != GPIO_PIN_SET) && (switch2state != GPIO_PIN_SET)
    && (switch3state != GPIO_PIN_SET) && (switch4state == GPIO_PIN_SET))
        return 0;
    else
        return 1;
}

void pause()
{
    GPIO_PinState state;
    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 = GPIO_PIN_SET;
    while ((state == GPIO_PIN_SET))
    {
        state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
        HAL_Delay(delay);
        if (animation_check())
            break;
    }
    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);
    return;
}

void animation()
{
    GPIO_PinState state = GPIO_PIN_SET;

    // 1st iteration
    HAL_GPIO_WritePin(GPIOD, leds_num[0], GPIO_PIN_SET);
```

```

HAL_GPIO_WritePin(GPIOD, leds_num[1], GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOD, leds_num[7], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)
pause();
if (animation_check())
return;

// 2nd iteration
HAL_GPIO_WritePin(GPIOD, leds_num[0], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[7], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[2], GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOD, leds_num[6], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)
pause();
if (animation_check())
return;

// 3d iteration
HAL_GPIO_WritePin(GPIOD, leds_num[1], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[6], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[3], GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOD, leds_num[5], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)
pause();
if (animation_check())
return;

// 4th iteration
HAL_GPIO_WritePin(GPIOD, leds_num[2], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[5], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[4], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)
pause();
if (animation_check())
return;

// 5th iteration
HAL_GPIO_WritePin(GPIOD, leds_num[5], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)
pause();
if (animation_check())
return;

// 6th iteration
HAL_GPIO_WritePin(GPIOD, leds_num[3], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[4], GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOD, leds_num[2], GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOD, leds_num[6], GPIO_PIN_SET);
state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
HAL_Delay(delay);
if (state != GPIO_PIN_SET)

```

```

    pause();
    if (animation_check())
        return;

    // 7th iteration
    HAL_GPIO_WritePin(GPIOD, leds_num[2], GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, leds_num[5], GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, leds_num[1], GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, leds_num[7], GPIO_PIN_SET);
    state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
    HAL_Delay(delay);
    if (state != GPIO_PIN_SET)
        pause();
    if (animation_check())
        return;

    // 8th iteration
    HAL_GPIO_WritePin(GPIOD, leds_num[1], GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, leds_num[6], GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, leds_num[0], GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, leds_num[7], GPIO_PIN_SET);
    state = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);
    HAL_Delay(delay);
    if (state != GPIO_PIN_SET)
        pause();
    if (animation_check())
        return;

    return;
}

int umain()
{
    while (1)
    {
        GPIO_PinState switch1state = HAL_GPIO_ReadPin(GPIOE, sw_num[0]);
        GPIO_PinState switch2state = HAL_GPIO_ReadPin(GPIOE, sw_num[1]);
        GPIO_PinState switch3state = HAL_GPIO_ReadPin(GPIOE, sw_num[2]);
        GPIO_PinState switch4state = HAL_GPIO_ReadPin(GPIOE, sw_num[3]);
        if (!((switch1state != GPIO_PIN_SET) && (switch2state != GPIO_PIN_SET)
        && (switch3state != GPIO_PIN_SET) && (switch4state == GPIO_PIN_SET)))
        {
            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);

            HAL_GPIO_WritePin(GPIOD, leds_num[0], switch1state);
            HAL_GPIO_WritePin(GPIOD, leds_num[1], switch2state);
            HAL_GPIO_WritePin(GPIOD, leds_num[2], switch3state);
            HAL_GPIO_WritePin(GPIOD, leds_num[3], switch4state);
        }
        else
        {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);

            // HAL_GPIO_WritePin(GPIOD, leds_num[3], GPIO_PIN_RESET);

            animation();

```

```
        for (i = 0; i < 8; i++)
            HAL_GPIO_WritePin(GPIOD, leds_num[i], GPIO_PIN_RESET);

        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);
    }
}
return 0;
}
```