

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ВТ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Операционные системы реального времени»
Тема: ОБЗОР ОС FREERTOS, ПОДГОТОВКА ПРОЕКТА.

Студенты гр. 6492

Преподаватель

Мурашко А. С.

Огурецкий Д. В.

Гречухин М. Н

Санкт-Петербург

2020

Цель: изучить использование ОС FreeRTOS в среде Keil MDK-ARM.

Задание: сконфигурировать ОС FreeRTOS и создать задачу, которая будет мигать светодиодом на плате.

Ход работы.

Создадим проект в Keil MDK-ARM под названием Lab1.

В качестве целевого устройства выбираем микроконтроллер STM32F411RE компании «ST Microelectronics».

Добавим все необходимые модули к проекту с помощью Manage Run-Time Environment:

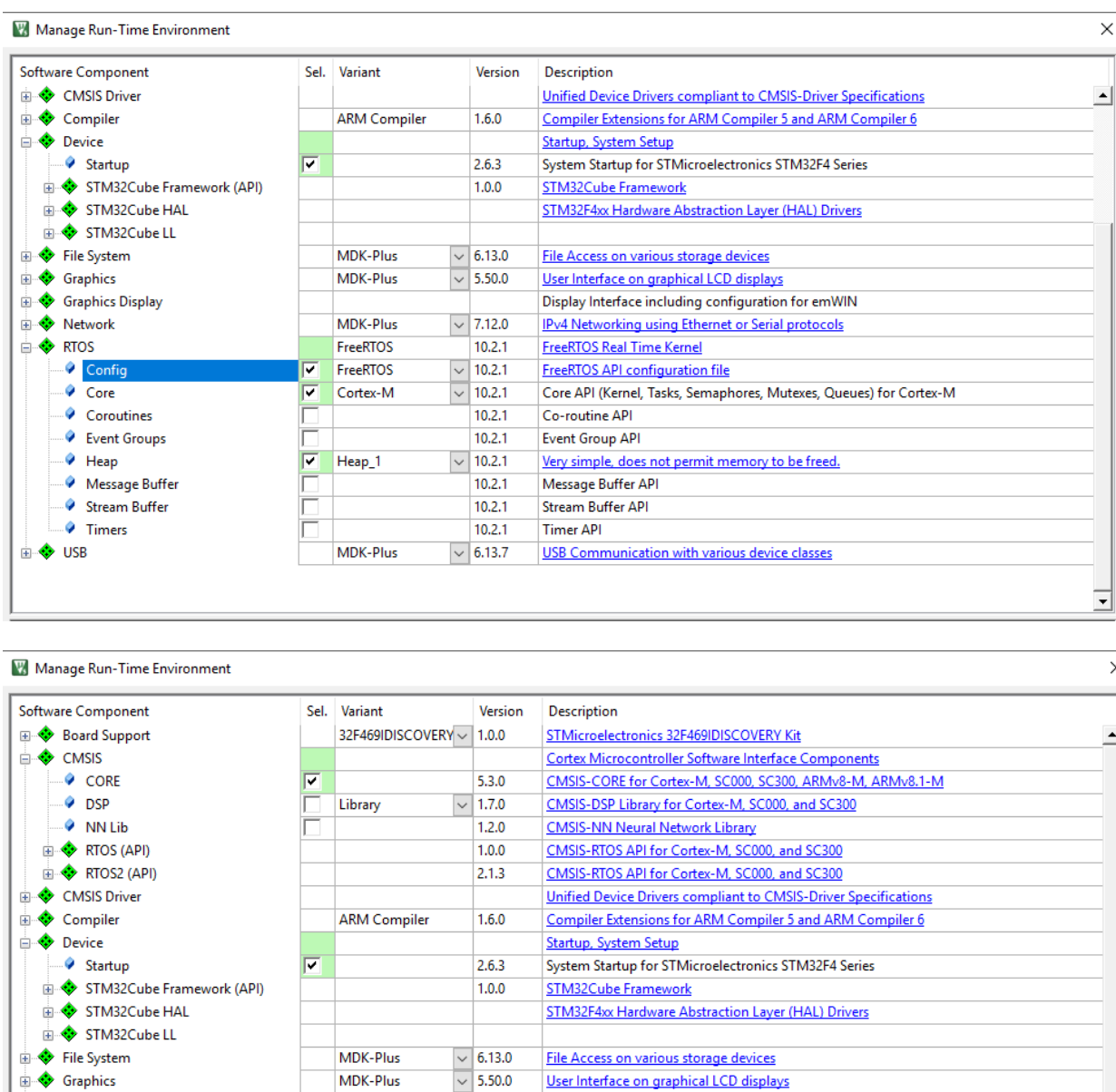


Рисунок 1 – Добавление модулей к проекту.

Настройки FreeRTOSConfig.h для корректной работы программы:

Требуется задать следующие параметры: частота CPU – «16MHz», частота тиков – «1KHz».

```
46
47 extern uint32_t SystemCoreClock;
48 #endif
49
50 /* Constants that describe the hardware and memory usage. */
51 #define configCPU_CLOCK_HZ 16000000
52 #define configTICK_RATE_HZ ((TickType_t)1000)
53 #define configTOTAL_HEAP_SIZE ((size_t)4096)
54 #define configMINIMAL_STACK_SIZE ((uint16_t)256)
55 #define configSUPPORT_DYNAMIC_ALLOCATION 1
56 #define configSUPPORT_STATIC_ALLOCATION 0
```

Рисунок 2 – Настройка частоты в файле FreeRTOSConfig.h

Необходимо применить патч для работы freeRTOS на STM32:

- 1) В STM32 используются 4 бита регистра приоритетов прерываний.
- 2) Поскольку используется 4 бита регистра приоритета прерываний, а 0 всегда является наиболее высоким приоритетом, то логично, что минимальный будет 15, то есть 0x0F в шестнадцатеричной системе.

```
102 /* Cortex-M specific definitions. */
103 #ifdef __NVIC_PRIO_BITS
104 /* __NVIC_PRIO_BITS will be specified when CMSIS is being used. */
105 #define configPRIO_BITS __NVIC_PRIO_BITS
106 #else
107 /* 7 priority levels */
108 #define configPRIO_BITS 4
109 #endif
110
111 /* The lowest interrupt priority that can be used in a call to a "set priority" function. */
112 #define configLIBRARY_LOWEST_INTERRUPT_PRIORITY 0x0F
```

Рисунок 3 – Настройки приоритета прерываний в файле FreeRTOSConfig.h

Создадим файл main.c, в котором и напишем программу для мигания диодом.

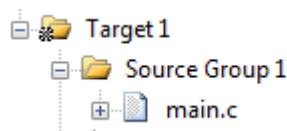


Рисунок 4 – Добавленный файл main.c

Проверять работу будем в эмуляторе. Для этого выберем соответствующий режим в настройках Flash.

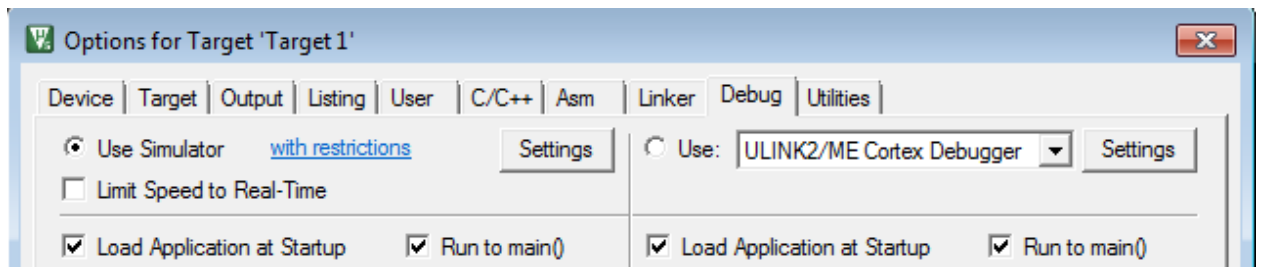


Рисунок 5 – Выбор эмулятора для работы.

Код программы.

```
#include "stm32f4xx.h"           // Device header
#include "FreeRTOSConfig.h"      // ARM.FreeRTOS::RTOS:Config
#include "FreeRTOS.h"           // ARM.FreeRTOS::RTOS:Core
#include "task.h"               // ARM.FreeRTOS::RTOS:Core

void task1(void* pvParameters)
{
    // blinking of diode
    while(1)
    {
        GPIOA->ODR |= GPIO_ODR_ODR_5;
        for(int i = 0; i < 1000000; i++);
        GPIOA->ODR &= ~GPIO_ODR_ODR_5;

        vTaskDelay(500); // delay 1 sec
    }
}

int main(void)
{
    // we use diode A5
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN; // turn on GPIOA
    GPIOA->MODER |= GPIO_MODER_MODER5_0; // set A5 output

    xTaskCreate(task1, "task1", configMINIMAL_STACK_SIZE, NULL, 4, NULL);
    vTaskStartScheduler();

    // in case of error in Scheduler
    while(1)
    {
    }
}
```

Результаты работы.

Запускаем Debug mode нажатием на иконку Start Debug Session.

Нет доступа к памяти, так как debugger считает, что выбран STM с MCU (защитой памяти). Соответственно ему нужно разрешить работать с памятью.

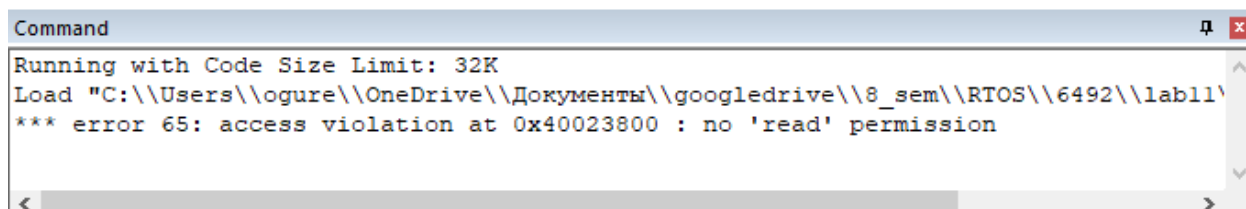


Рисунок 6 – Окно ошибок проекта.

Выбираем карту памяти:

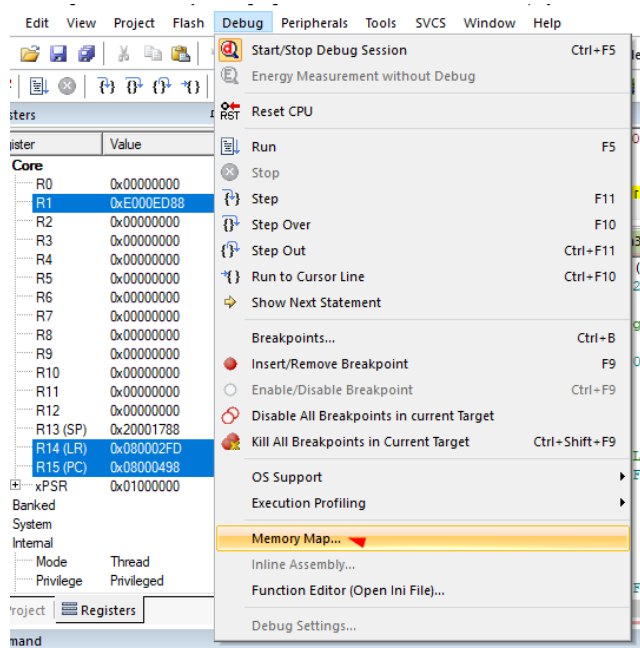


Рисунок 7 – Окно настройки Debug.

Выставляем разрешение на следующие области памяти:

0x40000000, 0x400FFFFF

и нажимаем Map Range.

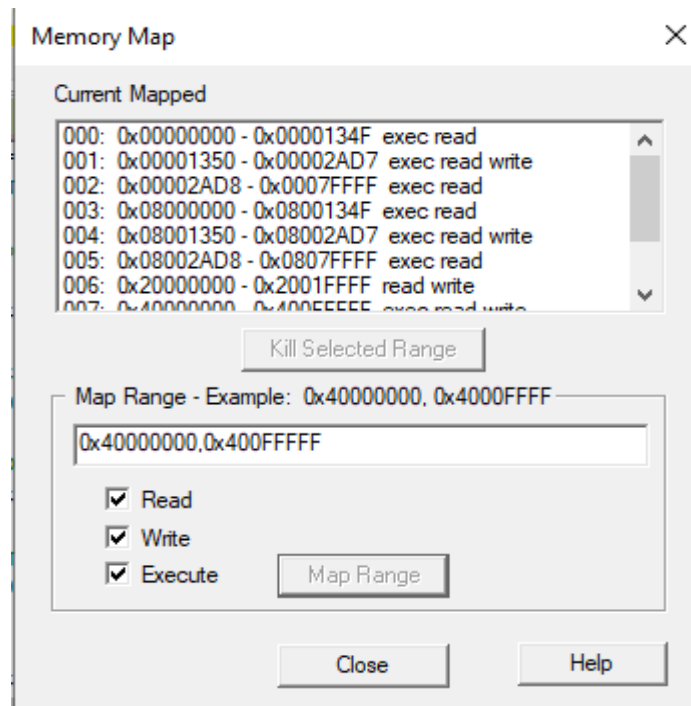


Рисунок 8 – Настройки Memory Map.

Ошибок больше нет, программа работает корректно. В результате получаем мигание «диода».

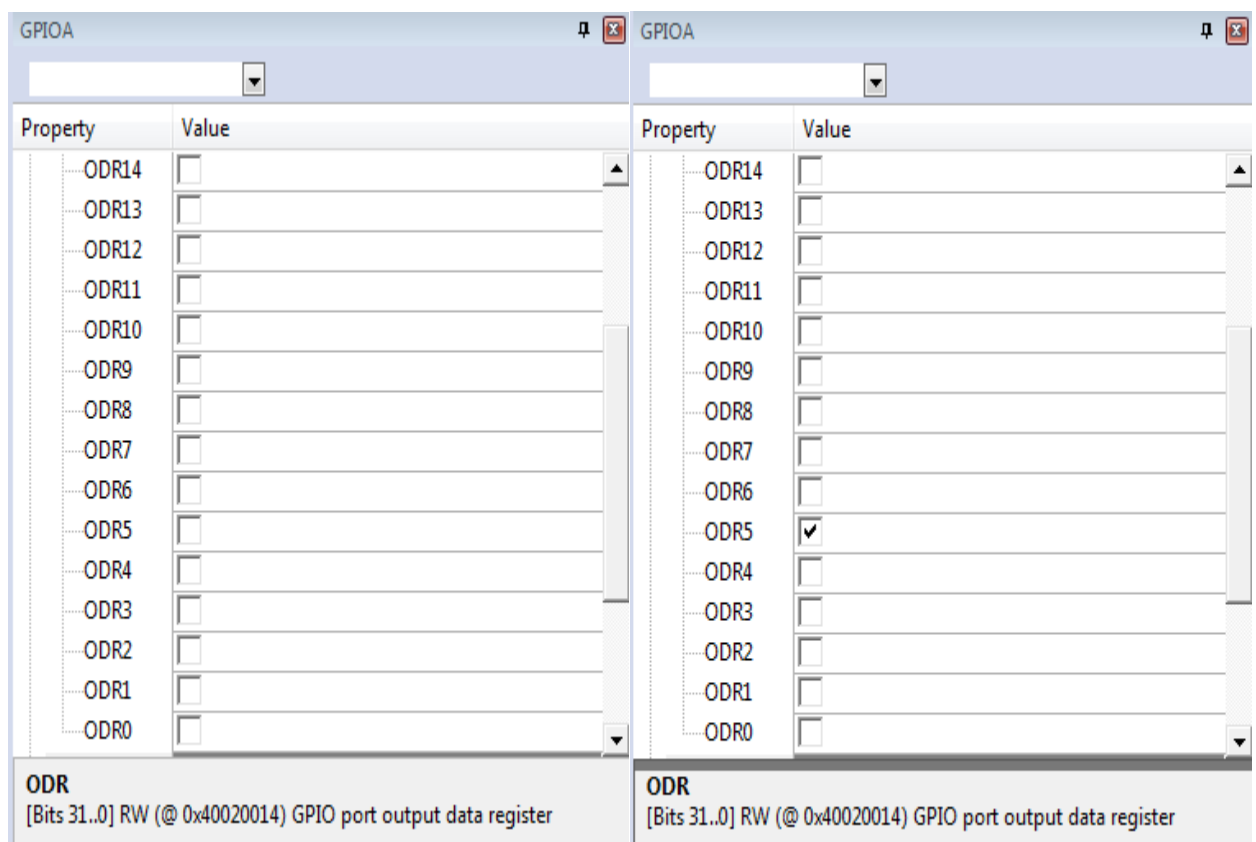


Рисунок 9 – Мигание диода в эмуляторе.

Вывод: в ходе работы было изучение использование ОС PV FreeRTOS, написана и отлажена программа, которая осуществляет мигание диода с периодом 1 секунда.