

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и кибербезопасности
Высшая школа технологий искусственного интеллекта
Направление Математика и Компьютерные науки

Отчёт по дисциплине Программирование
микроконтроллеров.

Лабораторная работа №3.

Работу выполнила:
Михайлова Алёна
студент группы 5130201/30002
Проверила:
Вербова Н. М.

Санкт-Петербург - 2025 г.

Тема :

События и прерывания.

Цель :

Ознакомится с основными методами обработки событий и прерываний.

Постановка задачи:

Разработать программу для микроконтроллера (МК) STM32F200 мигающую светодиодом PG7 и регистрирующую и обрабатывающую с разным приоритетом замыкание кнопок "WAKEUP" и "USER". При нажатии кнопки "WAKEUP" на некоторое время должен подключаться светодиод PG6, а при нажатии кнопки "USER" светодиод PG8 (Рис. 1).

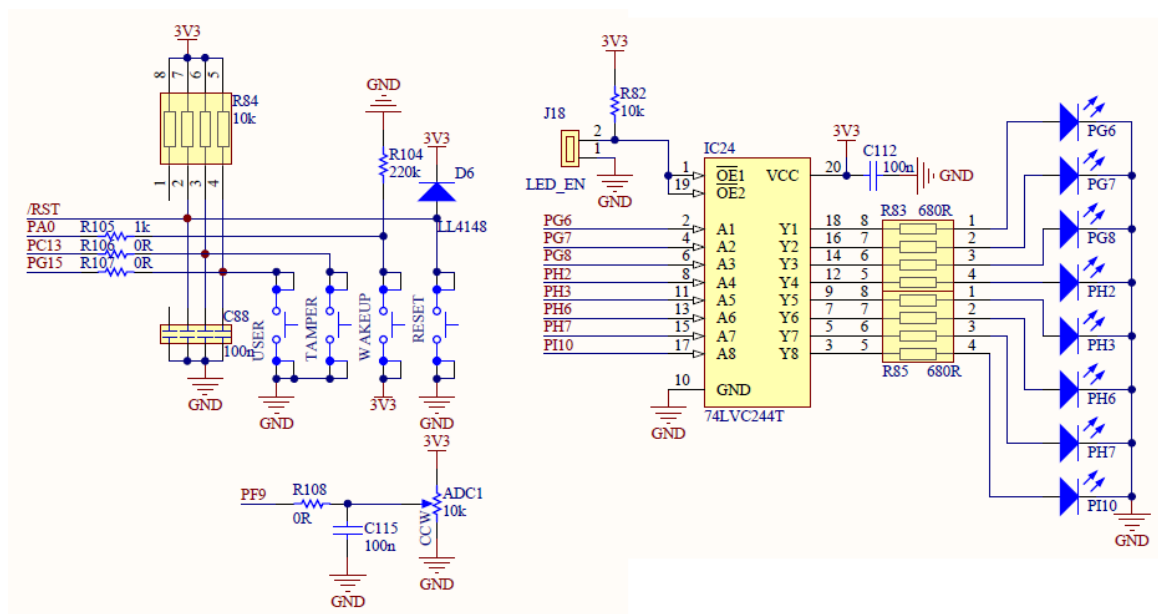


Рис. 1 Схема блоков кнопок и светодиодов

Теоретические данные:

Обработка прерываний в ядре Cortex-M3 микроконтроллера STM32 выполняется диспетчером вложенных векторных прерываний NVIC (Nested Vectored Interrupt Controller). Диспетчер прерываний – часть ядра Cortex-M3.

Контроллер NVIC поддерживает вложенность прерываний и их приоритеты. Каждому прерыванию при настройке NVIC присваивается свой приоритет. Если во время обработки низкоприоритетного прерывания возникает высокоприоритетное, то оно, в свою очередь, прервет обработчик низкоприоритетного прерывания.

Чтобы по окончании обработки прерывания возобновить работу с того места, в котором прерывание произошло, необходимо запомнить текущие результаты работы ядра для обеспечения возможности их последующего восстановления. При инициации прерывания NVIC переключает ядро в режим обработки прерывания. После перехода в режим обработки прерывания значения регистров ядра помещаются в стек.

Непосредственно во время записи значения регистров в стек осуществляется выборка начального адреса функции обработки прерывания.

В стек перемещаются данные из регистра статуса программы (Program Status Register (PSR)), счетчика программы (Program Counter (PC)) и регистра связи (Link Register (LR)). Благодаря этому запоминается состояние, в котором находилось ядро перед переходом в режим обработки прерываний.

Кроме того, сохраняются значения регистров R0 – R3 и R12. Регистры R0 – R3 используются в инструкциях для передачи параметров, поэтому помещение их значений в стек делает возможным их дальнейшее использование в функции обработки прерывания, а регистр R12 часто выступает в роли рабочего регистра программы.

По завершении обработки прерывания все действия выполняются в обратном порядке: извлекается содержимое стека и, параллельно с этим, осуществляется выборка адреса возврата.

С момента инициации прерывания до выполнения первой команды обработчика прерываний проходит 12 тактов, такое же время необходимо для возобновления основной программы после завершения обработки прерывания.

Код программы:

```
#include <stm32f207xx.h>
#include <stm32f2xx.h>
#include <core_cm3.h>

void
delay (unsigned long t)
{
    unsigned long i;
    for (i = 0; i < t; ++i);
}

int
```

```

main (void)
{
    RCC->AHB1ENR |= (RCC_AHB1ENR_GPIOEN | RCC_AHB1ENR_GPIOAEN); //включаем
    тактирование для портов GPIOG, GPIOA
    GPIOG->MODER |= (GPIO_MODER_MODE6_0 |
                    GPIO_MODER_MODE7_0 |
                    GPIO_MODER_MODE8_0);

    GPIOG->OTYPER = 0; //тип порта, 0 - обычный выход, который можно перевести в 0
или 1
    GPIOG->OSPEEDR = 0; //устанавливаем низкую скорость переключения порта
    GPIOA->MODER = 0; //настраиваем вывод на вход
    GPIOA->OTYPER = 0;
    GPIOA->OSPEEDR = 0;

    RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN; //SYSCFGEN тактирование контроллера
    конфигурации системы
    EXTI->IMR |= (EXTI_IMR_MR0 | EXTI_IMR_MR15); // EXTI - контроллер внешних
    прерываний, EXTI_IMR - регистр маски прерываний, 1 - прерывание включено
    EXTI->RTSR |= EXTI_RTSTR_TR0; //EXTI_RTSTR - Регистр включения детектора
    нарастающего края импульса
    EXTI->FTSR |= EXTI_FTSR_TR15; //EXTI_FTSR - Регистр включения детектора
    спадающего края импульса

    SYSCFG->EXTICR[0] |= SYSCFG_EXTICR1_EXTI0_PA; //прикрепляем PA к
    зарезервированной ранее линии
    SYSCFG->EXTICR[3] |= SYSCFG_EXTICR4_EXTI15_PG; //прикрепляем PG к
    зарезервированной ранее линии

    NVIC_SetPriorityGrouping (0x5UL); //задаем разбиение
    //IRQn - прерывание. Принимает значения из перечисления IRQn_Type, которое
    определено в файле stm32xxxx.h для конкретного микроконтроллера.
    //PreemptPriority - группа прерывания.
    NVIC_SetPriority (EXTI0_IRQn, 2 | 2 << 2); //задаем группу и подгруппы для
    конкретного прерывания
    NVIC_SetPriority (EXTI15_10_IRQn, 1 | 1 << 2);

    NVIC_EnableIRQ (EXTI0_IRQn); //активируем прерывание
    NVIC_EnableIRQ (EXTI15_10_IRQn); //активируем прерывание

    //мигаем диодом
    for (;;) {
        GPIOG->ODR ^= GPIO_ODR_OD7;
        delay (200000);
    }
}

//обработчик прерывания
void EXTI0_IRQHandler (void)
{
    unsigned int i;
    for (i = 0; i < 60; ++i) {
        GPIOG->ODR ^= GPIO_ODR_OD6; //включаем светодиод
        delay (100000);
    }
    EXTI->PR |= EXTI_PR_PR0; //сбрасываем прерывание
}

void EXTI15_10_IRQHandler (void)
{
    unsigned int i;
    for (i = 0; i < 30; ++i) {
        GPIOG->ODR ^= GPIO_ODR_OD8; //включаем светодиод
        delay (300000);
    }
    EXTI->PR |= EXTI_PR_PR15; //сбрасываем прерывание
}

```

Алгоритм программы:

Для начала в данной лабораторной работе необходимо настроить выходы PG6, PG7 и PG8 на вывод цифровых данных аналогично предыдущим лабораторным работам: подключить тактирование, с целью снижения энергопотребления МК после подачи на него питания, так как до этого момента практически все периферийные блоки выключены. Включение/выключение блока производится подачей/прекращением на него тактового сигнала.

Управление режимами работы разрядов портов ввода/вывода общего назначения выполняется с помощью регистра режимов, на рис.2. В данном случае нам следует установить значение 01 для вывода цифровых данных в соответствующие биты.

6.4.1 GPIO port mode register (GPIOx_MODER) (x = A..I)

Address offset: 0x00

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 2y:2y+1 **MODERy[1:0]**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

Рис. 2

Выходы PA0 и PG15 настраиваются на ввод цифровых данных. Для этого также пользуемся таблицей на Рис. 2. Исходя из этого устанавливаются значения 00, после подключения тактирования недостающих регистров.

После чего подключается тактирование контроллера конфигурации системы SYSCFG. Для ее подключения необходимо использовать конструкцию "Bit definition for RCC_APB2ENR register" файла stm32f207xx.h. А для установки значения воспользоваться документацией из

файла CD00225773.pdf, в файле была найдена информация о том, что для подключения тактирования необходимо перевести бит по номеру 14 в состояние со значением 1, что и было выполнено непосредственно в самой программе.

Далее настраивается контроллер внешних событий/прерываний EXTI (External interrupt/event controller). Всего есть 23 линии для конфигурирования источников прерываний, однако мы будем использовать только линии 0 и 15.

Конфигурируются маскирующие биты для требуемых линий, то есть 0 и 15. Для этого обратимся к файлу stm32f207xx.h и использовать конструкции из "Bit definition for EXTI_IMR register".

На данном этапе произошла резервация двух линий под прерывания, далее необходимо сконфигурировать биты регистров выбора фиксации линий прерывания по нарастающему и падающему фронтам (EXTI_RTSR и EXTI_FTSR). Для этого обратимся к конструкциям "Bit definition for EXTI_RTSR register" и "Bit definition for EXTI_FTSR register". В данной лабораторной работе происходит фиксация в одном случае нарастающего фронта (PA0), а в другом – падающего (PG15).

После этого прикрепляются порты к зарезервированным линиям. Это делается при помощи находящихся в контроллере конфигурации системы SYSCFG регистров конфигурации внешних прерываний контроллера конфигурации системы SYSCFG_EXTICR. Для подключения следует воспользоваться конструкцией из "EXTI0 configuration" и "EXTI15 configuration" файла stm32f207xx.h.

После настройки контроллера внешних прерываний следует установить приоритеты прерываний. Установка приоритетов осуществляется с помощью регистров приоритета прерывания NVIC_IPR (см. файл CD00228163.pdf). Для каждой линии использовалась функция "void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)" из строки 1464 файла core_cm3.h. В данной функции на первом месте стоит тип прерывания, то есть его позиция в таблице прерываний, а на втором непосредственно приоритет.

И наконец активируется прерывание. Для этого необходимо воспользоваться справочным руководством по программированию (см. файл CD00228163.pdf) активация обработки определенного вектора прерывания осуществляется с помощью регистров NVIC_ISET. Для каждой линии использовалась функция "void NVIC_EnableIRQ(IRQn_Type IRQn)" из строки 1382 файла core_cm3.h.

Сам обработчик прерываний написан отдельной переопределенной функцией. Имя этой функции найдено в предустановленном файле StartUp. Для каждой линии обработчик прерываний индивидуален. Он включает и выключает прикрепленный к нему светодиод.

Вывод:

Были изучены основные методы обработки событий и прерываний.

Разработана программа для микроконтроллера (МК) STM32F200 мигающую светодиодом PG7 и регистрирующую и обрабатывающую с разным приоритетом замыкание кнопок "WAKEUP" и "USER". При нажатии кнопки "WAKEUP" на некоторое время подключается светодиод PG6, а при нажатии кнопки "USER" светодиод PG8.