

## **1. Указание мер безопасности.**

К работе допускаются лица, изучившие настоящее описание, инструкцию по технике безопасности при работе с измерительными приборами, а также прошедшие инструктаж по безопасности труда на рабочем месте.

Прежде чем приступить к работе, внешним осмотром оборудования убедитесь в отсутствии механических повреждений его элементов, отсутствии торчащих и оборванных проводов, следов горения электрооборудования. О выявленных недостатках сообщите преподавателю.

Приступайте к работе только после разрешения и в присутствии преподавателя.

Производитель, выпустивший автомобиль,  
не обязан учесть вас его вождению.

Из интернет форума

### Лабораторная работа №3

**Тема:** События и прерывания.

**Цель:** Ознакомиться 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 (см. рис. 1).

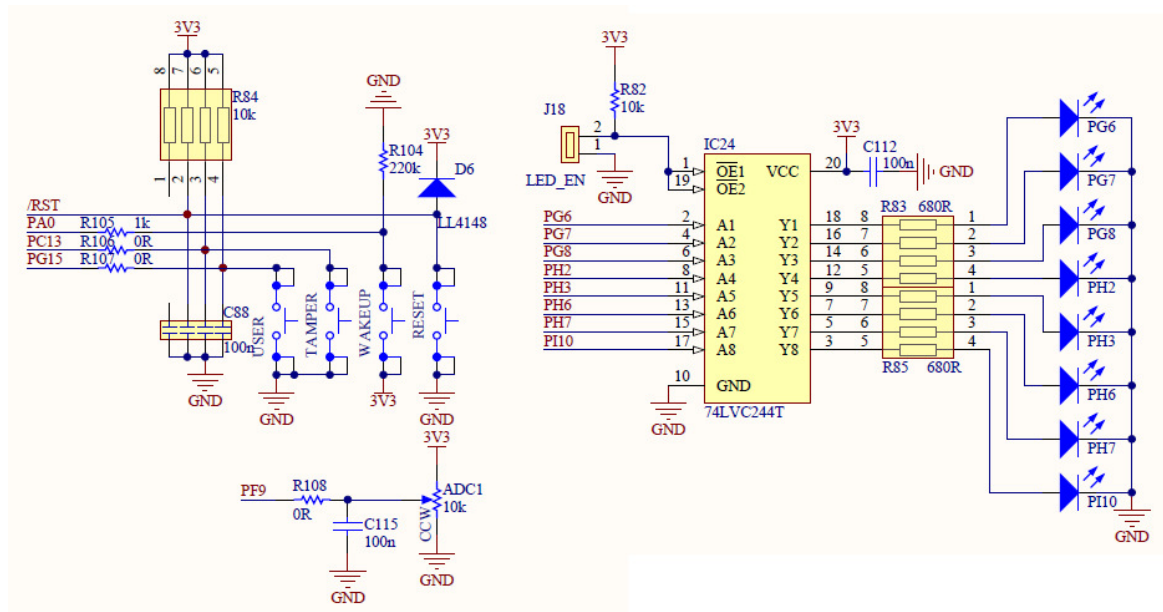


Рис. 1 Схема блоков кнопок и светодиодов

#### Материал для предварительного изучения:

1. Понятие и организация событий и прерываний.
2. Реализация стека.
3. Средства языка C для работы с событиями и прерываниями.

#### Краткие теоретические сведения

В процессе работы системы время от времени возникают некоторые внешние по отношению к исполняемой программе условия, требующие обработки

(обслуживания). Это события (Events). На произошедшее событие (аппаратное или программное) могут реагировать ядро или периферийные блоки. Одним из вариантов реакции может быть – *прерывание* (Interrupt). Взаимосвязь между событием и прерыванием заключается в следующем: каждое прерывание вызывается событием, но не каждое событие вызывает прерывание, помимо прерываний, события могут обрабатываться с помощью опроса по готовности.

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

Из самого названия видно, что 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.

*Стек* – это структура данных или устройство памяти, организованное по принципу “последним вошел – первым вышел (LIFO)”, т.е. данные поступившие в стек, будут извлекаться из него в обратном порядке.

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.

Как уже было сказано, NVIC поддерживает прерывания с различными приоритетами. В связи с этим различные прерывания могут прерывать друг друга. При этом могут возникать разные ситуации, обработка которых оптимизирована по-разному:

### **1. Приостановка низкоприоритетного прерывания**

В этой ситуации, обработка низкоприоритетного прерывания прекращается. Выполняется сохранение в стек набора данных низкоприоритетного прерывания и запускается обработка высокоприоритетного прерывания. После его обработки, содержимое стека автоматически извлекается и возобновляется обработка низкоприоритетного прерывания.

### **2. Непрерывная обработка прерываний**

Эта ситуация может возникнуть в двух случаях: если два прерывания имеют одинаковый приоритет и возникают одновременно и если низкоприоритетное прерывание возникает во время обработки высокоприоритетного.

В этом случае, промежуточные операции над стеком не производятся. Происходит только загрузка адреса обработчика низкоприоритетного прерывания и оно ставится в очередь.

### **3. Запоздывание высокоприоритетного прерывания**

Ситуация возникает, если высокоприоритетное прерывание происходит во время перехода к обработке низкоприоритетного.

Помимо простой установки приоритета прерываний, NVIC реализует возможность группировки приоритетов.

Прерывания в группе с более высоким приоритетом могут прерывать обработчики прерываний группы с более низким приоритетом. Прерывания из одной группы, но с разным приоритетом внутри группы не могут прерывать друг друга. Если же были активизированы оба события, приоритет внутри группы определяет только порядок вызова обработчика.

Значение приоритета прерывания задается в регистрах Interrupt Priority Registers. При этом часть бит отвечает за приоритет группы, в которой находится прерывание, а часть – за приоритет внутри группы.

Для каждого прерывания устанавливается программируемый уровень приоритета от 0 по 15. Более высокому значению уровня приоритета соответствует более низкий приоритет, таким образом, уровень 0 обладает самым высоким приоритетом прерывания.

Настройка распределения бит на приоритет группы или приоритет внутри группы осуществляется с помощью регистра Application Interrupt and Reset Control Register. Настройка приоритетов и группировка приоритетов зависят от конкретной реализации ядра.

Из возможных 8 бит приоритета используются только 4. Для большинства задач этого вполне достаточно.

Маскирование прерывания осуществляется с помощью регистров Interrupt Set-enable Registers.

Маскирование прерываний позволяет включать/выключать различные вектора прерываний. Если прерывание замаскировано, то при генерировании события периферией NVIC не вызывает обработчика данного события.

Все возможные прерывания, поддерживаемые NVIC, записываются в таблицу векторов прерываний. По своей сути, таблица векторов прерываний является

списком адресов функций обработчиков прерываний. Номер в списке соответствует номеру (вектору) прерывания. NVIC поддерживает до 240 различных векторов прерываний. В описании ядра стандартизованы только прерывания исключений ядра:

Reset  
NMI  
HardFault  
MemManage  
BusFault  
UsageFault  
SVCall  
PendSV  
SysTick

Остальные прерывания определяются конкретным производителем.

В таблице векторов, находящейся в начале адресного пространства флэш-памяти как минимум должны находиться:

- адрес вершины стека
- местоположение программы сброса
- местоположение NMI ISR
- местоположение Hard Fault ISR.

Ядро считывает из начала флэш-памяти значение указателя стека SP (адрес вершины стека) и значение PC (местоположение программы сброса). Таким образом, автоматически начинается выполнение функции с адресом, считанным в регистре PC. Это может быть, например функция `int main ()`.

Следом за обязательными четырьмя компонентами, может находиться дальнейшая таблица векторов прерываний. При желании, можно поместить таблицу векторов прерываний в другой области памяти, при этом следует сообщить NVIC, место нового размещения таблицы. За такое смещение таблицы векторов отвечает регистр Vector Table Offset Register (регистр смещения таблицы векторов).

### **Программа работы**

#### **1. Настроить порты ввода/вывода общего назначения GPIO.**

Настройте выходы PG6, PG7 и PG8 на вывод цифровых данных. Используйте конструкции из “Bits definition for GPIO\_MODER register”, “Bits definition for GPIO\_OTYPER register” и “Bits definition for GPIO\_OSPEEDR register” файла `stm32f207xx.h`.

Настройте выходы PA0 и PG15 на ввод цифровых данных. Используйте конструкции из “Bits definition for GPIO\_MODER register”, “Bits definition for GPIO\_OTYPER register” и “Bits definition for GPIO\_OSPEEDR register” файла `stm32f207xx.h`.

2. Подключите тактирование контроллера конфигурации системы SYSCFG. Используйте конструкцию “Bit definition for RCC\_APB2ENR register” файла stm32f207xx.h.

3. Настройте контроллер внешних событий/прерываний EXTI (External interrupt/event controller).

В соответствии с рекомендациями стр. 169 справочного руководства (см. файл CD00225773.pdf) необходимо поступить следующим образом:

### **Выбор аппаратных прерываний**

Для конфигурирования 23 линий как источников прерываний, используйте следующую процедуру:

- Сконфигурируйте маскирующие биты для требуемых линий из возможных 23 линий прерываний (EXTI\_IMR). Используйте конструкции из “Bit definition for EXTI\_IMR register” файла stm32f207xx.h.

- Сконфигурируйте биты регистров выбора фиксации линий прерывания по нарастающему и падающему фронтам (соответственно EXTI\_RTSR и EXTI\_FTSR). Используйте конструкции из “Bit definition for EXTI\_RTSR register” и “Bit definition for EXTI\_FTSR register” файла stm32f207xx.h.

- Сконфигурируйте биты подключения и маски, управляющие каналом NVIC IRQ, закрепленным за контроллером внешнего прерывания (EXTI) так, чтобы прерывание, поступающее от одной из 23 линий, могло быть правильно распознано. Этот пункт реализуйте чуть позже при настройке NVIC.

А сейчас следует определиться с тем, к какому порту будут относиться выбранные линии EXTI. Это делается при помощи находящихся в контроллере конфигурации системы SYSCFG регистров конфигурации внешних прерываний контроллера конфигурации системы SYSCFG\_EXTICR. Используйте конструкции из “EXTI0 configuration” и “EXTI15 configuration” файла stm32f207xx.h.

Теперь можно вернуться к отложенному ранее пункту, т.е. настройке NVIC.

Настройте ресурсы, выделяемые под групповые приоритеты прерываний. Настройка ресурсов выполняется с помощью регистра прерывания приложений и управления сбросом SCB\_AIRCR. Используйте для настройки функцию “NVIC\_SetPriorityGrouping(uint32\_t PriorityGroup)”.

Установите приоритеты прерываний. Установка приоритетов осуществляется с помощью регистров приоритета прерывания NVIC\_IPR (см. файл CD00228163.pdf). Используйте для каждой линии функцию “void NVIC\_SetPriority(IRQn\_Type IRQn, uint32\_t priority)” из строки 1464 файла core\_cm3.h.

Активируйте прерывания. В соответствии со справочным руководством по программированию (см. файл CD00228163.pdf) активация обработки определенного вектора прерывания осуществляется с помощью регистров NVIC\_ISER. Используйте для каждой линии функцию “void NVIC\_EnableIRQ(IRQn\_Type IRQn)” из строки 1382 файла core\_cm3.h.

4. Наберите основное тело программы, используя предоставляемые возможности Keil µVision5.

5. Постройте проект, нажав F7 или выбрав пункт Build из меню Project.



6. Запустите отладочный режим, выбрав из меню Debug пункт Start/Stop Debug Session.

7. В появившемся окошке Evaluation Mode нажмите кнопку ОК.

8. Находясь в режиме отладки, выберите из меню Debug пункт Run.

9. Программа загрузится в оценочную плату и светодиод PG7 начнет мигать.

10. Нажимая на кнопки “WAKEUP” и “USER” в разном порядке наблюдайте за очередностью обработки вызванных ими прерываний регистрируемой соответственно светодиодами PG6 и PG8.

10. Поменяйте приоритеты прерываний в программе наоборот. Наблюдайте за тем, что изменилось в порядке обработки прерываний. Сделайте приоритеты одинаковыми и отметьте произошедшие изменения в порядке обработки прерываний.

11. Прodelайте подобные изменения с приоритетами групп прерываний. Зафиксируйте изменения в порядке обработки прерываний групп и внутри группы.

12. Продемонстрируйте полученный результат преподавателю.

#### **Содержание отчета:**

1. Тема и цель работы.

2. Задание на лабораторную работу.

3. Алгоритм программы.

4. Полученные результаты.

5. Выводы по работе с анализом реализованной программы.

#### **Литература:**

1. RM0033 Reference manual. STM32F205xx, STM32F207xx, STM32F215xx and STM32F217xx advanced ARM-based 32-bit MCUs. Доступно (2015, Июнь): <http://www.st.com/>

2. PM0056 Programming manual. STM32F10xxx/20xxx/21xxx/L1xxxx Cortex-M3 programming manual. Доступно (2015, Июнь): <http://www.st.com/>

3. Cortex™-M3 DevicesGeneric User Guide. Доступно (2015, Июнь): <http://www.st.com/>

## Регистр маски прерываний

### 8.3.1 Interrupt mask register (EXTI\_IMR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	MR18	MR17	MR16
									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
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
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 31:23 Reserved, must be kept at reset value (0).

Bits 22:0 **MRx**: Interrupt mask on line x

0: Interrupt request from line x is masked

1: Interrupt request from line x is not masked

Для демаскирования линии прерываний, например 15 в регистр EXTI\_IMR с адресом 0x40013C00 необходимо записать значение 0x8000.

## Регистр выбора фиксации линий прерывания по падающему фронту

### 8.3.4 Falling trigger selection register (EXTI\_FTSR)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									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
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
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 31:23 Reserved, must be kept at reset value (0).

Bits 22:0 **TRx**: Falling trigger event configuration bit of line x

0: Falling trigger disabled (for Event and Interrupt) for input line

1: Falling trigger enabled (for Event and Interrupt) for input line.

Позволяет настроить контроллер прерываний так, чтобы прерывания возникали при переходе потенциалов на выводах, например PG15 или PC13 (см. рис. 1) из состояния логической 1 в состояние логического 0. Для этого в регистр EXTI\_FTSR с адресом 0x40013C0C необходимо записать значение 0xA000.



## Регистр выбора фиксации линий прерывания по нарастающему фронту

### 8.3.3 Rising trigger selection register (EXTI\_RTSTR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 Reserved, must be kept at reset value (0).

Bits 22:0 **TRx**: Rising trigger event configuration bit of line x

0: Rising trigger disabled (for Event and Interrupt) for input line

1: Rising trigger enabled (for Event and Interrupt) for input line

Позволяет настроить контроллер прерываний так, чтобы прерывания возникали при переходе потенциалов на выводах, например выводе PA0 (см. рис. 1) из состояния логического 0 в состояние логической 1. Для этого в регистр EXTI\_RTSTR с адресом 0x40013C08 необходимо записать значение 0x0001.

## Регистр ожидания

### 8.3.6 Pending register (EXTI\_PR)

Address offset: 0x14

Reset value: undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PR22	PR21	PR20	PR19	PR18	PR17	PR16
									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits 31:23 Reserved, must be kept at reset value (0).

Bits 22:0 **PRx**: Pending bit

0: No trigger request occurred

1: selected trigger request occurred

This bit is set when the selected edge event arrives on the external interrupt line. This bit is cleared by writing a 1 to the bit or by changing the sensitivity of the edge detector.

Регистр ожидания EXTI\_PR, активирует приоритеты прерываний и используется для сброса прерывания. Адрес этого регистра 0x40013C14.

## Регистр конфигурации внешних прерываний контроллера конфигурации системы

### 7.2.6 SYSCFG external interrupt configuration register 4 (SYSCFG\_EXTICR4)

Address offset: 0x14

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved

Bits 15:0 **EXTIx[3:0]**: EXTI x configuration (x = 12 to 15)

These bits are written by software to select the source input for the EXTIx external interrupt.

0000: PA[x] pin

0001: PB[x] pin

0010: PC[x] pin

0011: PD[x] pin

0100: PE[x] pin

0101: PF[x] pin

0110: PG[x] pin

0111: PH[x] pin

*Note: PI[15:12] are not used.*

Регистры конфигурации внешних прерываний контроллера конфигурации системы SYSCFG\_EXTICR позволяют определиться с тем, к какому порту будут относиться выбранные линии EXTI. Например, если необходимо чтобы линия 15 принадлежала порту G в регистр SYSCFG\_EXTICR4 с адресом 0x40013814 необходимо записать G значение 0x6000.

## Регистры установки/сброса прерываний

### 4.2.2 Interrupt Set-enable Registers

The NVIC\_ISER0-NVIC\_ISER7 registers enable interrupts, and show which interrupts are enabled. See the register summary in [Table 4-2 on page 4-3](#) for the register attributes.

The bit assignments are:

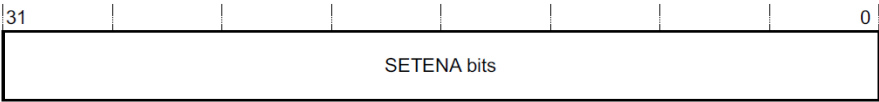


Table 4-4 ISER bit assignments

Bits	Name	Function
[31:0]	SETENA	Interrupt set-enable bits. Write: 0 = no effect 1 = enable interrupt. Read: 0 = interrupt disabled 1 = interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

С помощью регистров NVIC\_ISER с базовым адресом 0xE000E100 осуществляется активация обработки определенного вектора прерывания. В соответствии с таблицей векторов прерываний МК линии, например с 10 по 15 генерируют одно прерывание на всех с номером 40.

## Таблица векторов прерываний

### 8.1.3 Interrupt and exception vectors

[Table 20](#) is the vector table for the STM32F20x and STM32F21x devices.

Table 20. Vector table

Position	Priority	Type of priority	Acronym	Description	Address
	-	-	-	Reserved	0x0000_0000
	-3	fixed	Reset	Reset	0x0000_0004
...	...	...	...	...	...
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000_00E0

В соответствии с картой регистров NVIC прерывание 40 активируются в регистре NVIC\_ISER1 с адресом 0xE000E104.

## Карта регистров NVIC и значения сброса

### 4.3.11 NVIC register map

The table provides shows the NVIC register map and reset values. The base address of the main NVIC register block is 0xE000E100. The NVIC\_STIR register is located in a separate block at 0xE000EF00.

Table 44. NVIC register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	NVIC_ISER0	SETENA[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	NVIC_ISER1	SETENA[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Для активации 40 прерывания в регистр NVIC\_ISER1 необходимо записать 0x0100.

## Регистр прерывания приложений и управления сбросом

### 4.4.5 Application interrupt and reset control register (SCB\_AIRCR)

Address offset: 0x0C

Reset value: 0xFA05 0000

Required privilege: Privileged

The AIRCR provides priority grouping control for the exception model, endian status for data accesses, and reset control of the system.

To write to this register, you must write 0x5FA to the VECTKEY field, otherwise the processor ignores the write.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
r					rw	rw	rw						w	w	w

Bits 31:16 **VECTKEYSTAT[15:0]/ VECTKEY[15:0]** Register key

Reads as 0xFA05

On writes, write 0x5FA to VECTKEY, otherwise the write is ignored.

Bit 15 **ENDIANESS** Data endianness bit

Reads as 0.

0: Little-endian

Bits 14:11 Reserved, must be kept cleared

Bits 10:8 **PRIGROUP[2:0]**: Interrupt priority grouping field

This field determines the split of group priority from subpriority, see [Binary point on page 135](#).

Bits 7:3 Reserved, must be kept cleared

Bit 2 **SYSRESETREQ** System reset request

This is intended to force a large system reset of all major components except for debug.

This bit reads as 0.

0: No system reset request

1: Asserts a signal to the outer system that requests a reset.

Bit 1 **VECTCLRACTIVE**

Reserved for Debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is unpredictable.

Bit 0 **VECTRESET**

Reserved for Debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is unpredictable.

Позволяет настраивать ресурсы, выделяемые под групповые приоритеты. Настройка ресурсов групповых приоритетов осуществляется в соответствии с таблицей группировки приоритетов.

**Таблица группировки приоритетов**

**Table 45. Priority grouping**

PRIGROUP [2:0]	Interrupt priority level value, PRI_N[7:4]			Number of	
	Binary point <sup>(1)</sup>	Group priority bits	Subpriority bits	Group priorities	Sub priorities
0b011	0bxxxx	[7:4]	None	16	None
0b100	0bxxx.y	[7:5]	[4]	8	2
0b101	0bxx.yy	[7:6]	[5:4]	4	4
0b110	0bx.yyy	[7]	[6:4]	2	8
0b111	0b.yyyy	None	[7:4]	None	16

1. PRI\_n[7:4] field showing the binary point. x denotes a group priority field bit, and y denotes a subpriority field bit.

Например, для настройки ресурсов групповых приоритетов таким образом, чтобы за приоритет группы и за приоритет внутри группы отвечало по 2 бита (см. среднюю строку в таблице группировки приоритетов) в регистр прерывания приложений и управления сбросом, необходимо записать значение 0x05FA0500.

Адрес самого регистра вычисляется как сумма смещения его адреса и базового адреса блока регистров SCB.

#### **Сведения о базовом адресе блока SCB из справочного руководства по программированию**

##### **4.4.15 SCB register map**

The table provides shows the System control block register map and reset values.

The base address of the SCB register block is 0xE000 ED00 for register described in [Table 48](#).

Адресом регистра прерывания приложений и управления сбросом является 0xE000ED0C.

## Регистры приоритета прерывания

### 4.3.7 Interrupt priority registers (NVIC\_IPRx)

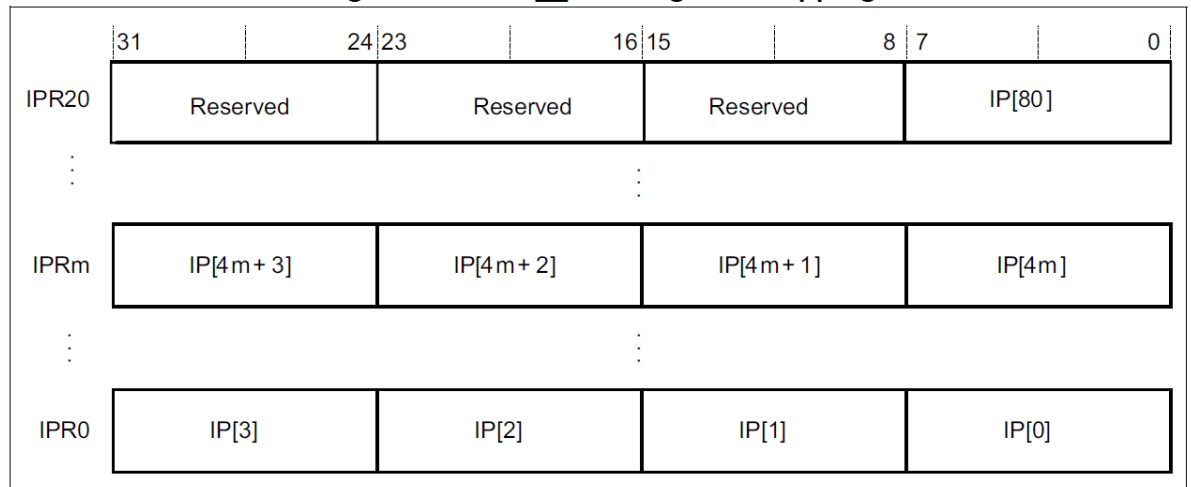
Address offset: 0x00- 0x0B

Reset value: 0x0000 0000

Required privilege: Privileged

The IPR0-IPR16 registers provide a 4-bit priority field for each interrupt. These registers are byte-accessible. Each register holds four priority fields, that map to four elements in the CMSIS interrupt priority array IP[0] to IP[67], as shown in [Figure 19](#).

**Figure 19. NVIC\_IPRx register mapping**



**Table 42. IPR bit assignments**

Bits	Name	Function
[31:24]	Priority, byte offset 3	Each priority field holds a priority value, 0-255. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[7:4] of each field, bits[3:0] read as zero and ignore writes.
[23:16]	Priority, byte offset 2	
[15:8]	Priority, byte offset 1	
[7:0]	Priority, byte offset 0	

See [The CMSIS mapping of the Cortex-M3 NVIC registers on page 119](#) for more information about the IP[0] to IP[67] interrupt priority array, that provides the software view of the interrupt priorities.

Find the IPR number and byte offset for interrupt  $N$  as follows:

- The corresponding IPR number,  $M$ , is given by  $M = N \text{ DIV } 4$
- The byte offset of the required Priority field in this register is  $N \text{ MOD } 4$ , where:
  - byte offset 0 refers to register bits[7:0]
  - byte offset 1 refers to register bits[15:8]
  - byte offset 2 refers to register bits[23:16]
  - byte offset 3 refers to register bits[31:24].

Позволяют осуществлять настройку непосредственно приоритетов.



Базовым адресом блока регистров NVIC является 0xE000E100. Согласно карте регистров NVIC смещение адреса регистра NVIC\_IPR0 равно 0x300. Таким образом, его адресом будет 0xE000E400. Для 80 прерывания, например, номер регистра IPR будет определяться как  $80/4 = 20$  и, следовательно, адреса регистра NVIC\_IPR20 будет 0xE000E420. Допускается побайтная адресация. В этом случае к адресу регистра добавляется смещение байта.

В байте значащими являются только старшие 4 бита приоритета.

### Фрагмент карты регистров NVIC и значений сброса

Table 44. NVIC register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x204	NVIC_IABR1	ACTIVE[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x208	NVIC_IABR2	Reserved																ACTIVE [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x300	NVIC_IPR0	IP[3]								IP[2]								IP[1]								IP[0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
:	:	:																															
0x320	NVIC_IPR20	Reserved																								IP[80]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SCB registers																																	