

# STM32F103. Контроллер прерываний

**Прерывание** (англ. *interrupt*) — сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается и управление передаётся обработчику прерывания, который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.

# STM32F103. Контроллер прерываний

В зависимости от источника прерывания делятся на:

- внешние (от периферийных устройств);
- внутренние ( деление на ноль, переполнение стека);
- программные (инициируются самой программой для использования функций firmware, драйверов, операционной системы).

# STM32F103. Контроллер прерываний

Внешние прерывания делятся на:

- маскируемые – прерывания, которые можно запрещать установкой соответствующих разрядов;
- немаскируемые – прерывания, которые исполняются всегда.

Кроме того, важной особенностью набора прерываний является приоритезация.

# STM32F103. Контроллер прерываний

**Nested vectored interrupt controller (NVIC).**

**Контроллер вложенных векторизованных прерываний.**

Вектор прерывания – указатель на область памяти, которая содержит указатель на обработчик прерывания.

Векторы объединяются в таблицу векторов.

# STM32F103. Контроллер прерываний

Position	Priority	Type of priority	Acronym	Description	Address
	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000_0008
	-1	fixed	HardFault	All class of fault	0x0000_000C
	0	settable	MemManage	Memory management	0x0000_0010
	1	settable	BusFault	Pre-fetch fault, memory access fault	0x0000_0014
	2	settable	UsageFault	Undefined instruction or illegal state	0x0000_0018
	-	-	-	Reserved	0x0000_001C - 0x0000_002B
	3	settable	SVCall	System service call via SWI instruction	0x0000_002C
	4	settable	Debug Monitor	Debug Monitor	0x0000_0030
	-	-	-	Reserved	0x0000_0034
	5	settable	PendSV	Pendable request for system service	0x0000_0038
	6	settable	SysTick	System tick timer	0x0000_003C
0	7	settable	WWDG	Window Watchdog interrupt	0x0000_0040
1	8	settable	PVD	PVD through EXTI Line detection interrupt	0x0000_0044
2	9	settable	TAMPER	Tamper interrupt	0x0000_0048
3	10	settable	RTC	RTC global interrupt	0x0000_004C

# STM32F103. Контроллер прерываний

При написании программы в среде Keil  
таблица векторов прерываний  
находится в файле

**startup\_stm32f10x\_hd.s;**

Для обработки прерывания надо  
написать обработчик:

```
void EXTI3_IRQHandler(void)  
{  
}
```

# STM32F103. Контроллер прерываний

*Контроллер прерываний является частью ядра Cortex-M3. Для допуска обработки прерывания его надо разрешить, описания управляющих регистров находятся в документации на библиотеку CMSIS.*

*Кроме того, надо разрешить генерацию прерывания соответствующей периферией.*

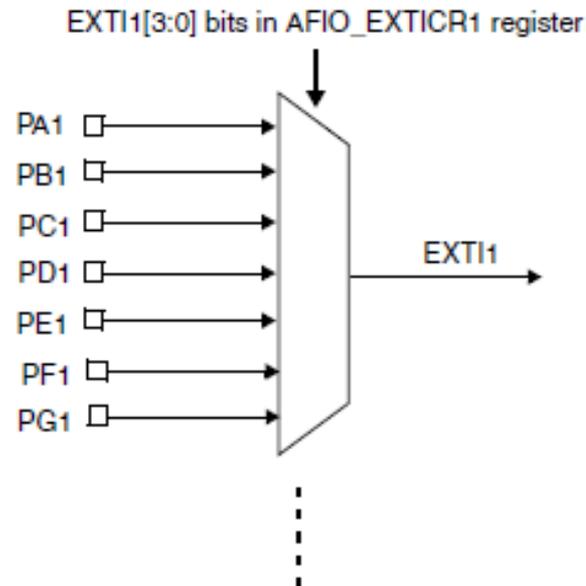
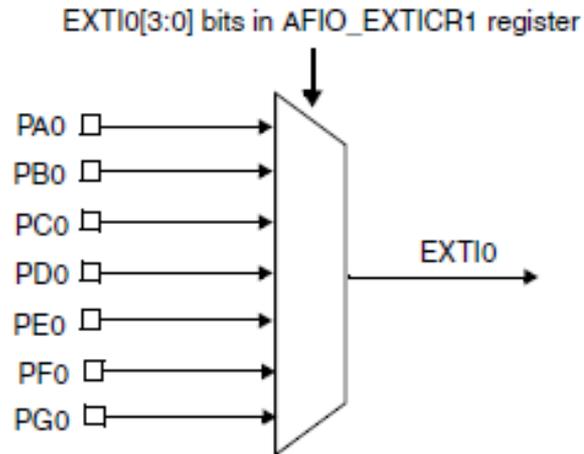
*Поэтому в программе рекомендуется использовать функции StdPeriphLib, HAL library.*

# STM32F103. Контроллер прерываний

Для расширения управления линиями прерывания фирма STM дополнила стандартный контроллер прерываний контроллером внешних событий/ прерываний **External interrupt/event controller (EXTI)**.

Контроллер имеет 19 линий с детекторами фронта.

# STM32F103. Контроллер прерываний



- EXTI line 16 is connected to the PVD output
- EXTI line 17 is connected to the RTC Alarm event
- EXTI line 18 is connected to the USB Wakeup event
- EXTI line 19 is connected to the Ethernet Wakeup event (available only in connectivity line devices)

# STM32F103. Контроллер прерываний

На 19 линиях могут генерироваться прерывания (interrupt) и события (events).

Отличия прерывания от события:

прерывание всегда обрабатывается программным кодом, например, в

обработчике прерывания по линии EXTI2 можно запустить АЦП;

событие при нужной настройке периферии, например АЦП, может напрямую запустить АЦП.

# STM32F103. Контроллер ПДП

Контроллер прямого доступа к памяти (DMA).

Устройство для пересылок между областями памяти без использования процессора.

Так как STM32 имеет единое адресное пространство, то разрешены следующие пересылки:

- память – память;
- периферия – память;
- память – периферия;
- периферия – периферия.

Основная задача – разгрузка процессора от часто повторяющихся операций.

# STM32F103. Контроллер ПДП

## Пример 1.

На ножку МК подаётся тактовая частота.

АЦП настроен на запуск по фронту сигнала.

Требуется организовать сбор данных АЦП в массив без использования процессора.

*Решение.*

Настраивается КПДП на работу с соответствующим АЦП, причем данные от АЦП собираются с тактовой частотой и пересылаются в кольцевой буфер.

При заполнении первой или второй половины буфера генерируются прерывания, при обработке которых процессор забирает данные.

# STM32F103. Контроллер ПДП

## Пример 2.

На ножку МК подаётся тактовая частота.

ЦАП настроен на запуск по фронту тактового сигнала.

Требуется организовать пересылку данных из массива в ЦАП без использования процессора.

*Решение.*

Настраивается КПДП на работу с соответствующим каналом ЦАП, причем данные в ЦАП пересылаются из кольцевого буфера с тактовой частотой.

При пересылке первой или второй половины буфера генерируются прерывания, при обработке которых процессор обновляет данные.

# STM32F103. Контроллер ПДП

## DMA

- Два контроллера DMA1 и DMA2;
- программирование приоритетов запросов:
  - *very high, high, medium, low;*
  - *при одинаковом приоритете запрос канала 1 имеет более высокий приоритет, чем запрос канала 2, и т.д.;*
- доступ ко флэш, SRAM и любой периферии;
- программируемый размер пересылки до 65536.

# STM32F103. Контроллер ПДП

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1						
SPI/I <sup>2</sup> S		SPI1_RX	SPI1_TX	SPI2/I2S2_RX	SPI2/I2S2_TX		
USART		USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I <sup>2</sup> C				I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG	
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP

# STM32F103. Интерфейсы

Интерфейсы МК:

- USART – 5;
- SPI – 3;
- I2S – 2;
- I2C – 2;
- USB – 1;
- CAN – 1;
- SDIO – 1.

# STM32F103. USART

**Универсальный синхронный-асинхронный приёмопередатчик**  
(*Universal Synchronous-Asynchronous Receiver-Transmitter, U(S)ART*).

Устройство для коммуникации между устройствами по последовательному протоколу.

# STM32F103. USART

Физический уровень.

Со стороны МК: логические уровни 0 – Vdd.

Следовательно, со стороны другого устройства должны быть такие же уровни.

Нельзя подключать к USART МК устройства с портом RS-232 или RS-485 без преобразователя уровней.

RS-232:

Лог. 0: от +5 до +15 В;

Лог. 1: от -5 до -15 В.

# STM32F103. USART

Где используется UART.

Например, связь МК-компьютер.

Преимущества:

- Легкая организация протокола обмена;
- Нужно всего две линии Tx и Rx;
- Организация гальваноразвязки.

Недостатки:

- Требуются преобразователи интерфейса USB-UART, но они дешевы и доступны.

# STM32F103. USART

Преобразователи интерфейса USB-UART:

- FT232, CP2102.

