Управление прерываниями в STM32F2xx.

Используемый модуль стандартной библиотеки драйверов периферии – MISC.

**void NVIC\_Init(NVIC\_InitTypeDef \*NVIC\_InitStruct);**

Описание:

Инициализирует периферийный модуль контроллера вложенных векторов прерываний NVIC в соответствии с специальными параметрами в структуре NVIC\_Init\_Struct.

Примечание: для корректной настройки приоритетов прерываний сперва необходимо вызвать функцию NVIC\_PriorityGroupConfig().

Параметры:

*NVIC\_InitStruct* – указатель на структуру NVIC\_InitTypeDef, которая содержит конфигурационную информацию для определенного периферийного NVIC. Содержит следующие поля:

uint8\_t NVIC\_IRQChannel, - определяет канал запроса прерывания для разрешения/запрещения. Этот параметр может быть перичеслением IRQn\_Type (список перечисления смотреть в справочнике).

uint8\_t NVIC\_IRQChannelPreemptionPriority, - определяет приоритет вытеснения для канала запроса прерывания IRQ, выбранного в NVIC\_IRQChannel. Этот параметр может принимать значения от 0 до 15, как описано в таблице MISC\_NVIC\_Priority\_Table. Самое низкое значение приоритета означает наивысший приоритет.

uint8\_t NVIC\_IRQChannelSubPriority, - определяет уровень под-приоритета для канала запроса прерывания IRQ, выбранном в NVIC\_IRQChannel. Этот параметр может принимать значения от 0 до 15, как описано в таблице MISC\_NVIC\_Priority\_Table. Самое низкое значение приоритета означает наивысший приоритет.

FunctionalState NVIC\_IRQChannelCmd, - определяет, разрешен, или запрещен канал запроса прерывания IRQ, определенный в поле NVIC\_IRQChannel. FunctionalState – перечисление, имеющее два значения: DISABLE, ENABLE.

Возвращаемое значение:

Нет.

**void NVIC\_PriorityGroupConfig(uint32\_t NVIC\_PriorityGroup);**

Описание:

Настраивает групповой приоритет: приоритет вытеснения и под-приоритет.

Параметры:

*NVIC\_PriorityGroup*, - определяет длину битов группирующего приоритета. Этот параметр может принимать одно из следующих значений:

NVIC\_PriorityGroup\_0: 0 битов для приоритета вытеснения, 4 бита для под-приоритета.

NVIC\_PriorityGroup\_1: 1 бит для приоритета вытеснения, 3 бита для под-приоритета.

NVIC\_PriorityGroup\_2: 2 бита для приоритета вытеснения, 2 бита для под-приоритета.

NVIC\_PriorityGroup\_3: 3 бита для приоритета вытеснения, 1 бит для под-приоритета.

NVIC\_PriorityGroup\_4: 4 бита для приоритета вытеснения, 0 битов для под-приоритета.

Примечание: когда выбран NVIC\_PriorityGroup\_0, IRQ вытеснение больше невозможно. Ожидающий запрос прерывания будет обслужен только по под-приоритету.

Возвращаемое значение:

Нет.

**void NVIC\_SetVectorTable(uint32\_t NVIC\_VectTab, uint32\_t Offset);**

Описание:

Устанавливает положение векторной таблицы и смещение.

Параметры:

*NVIC\_VectTab*, - определяет, где располагается векторная таблица: в RAM, или Flash памяти. Этот параметр может принимать одно из следующих значений:

NVIC\_VectTab\_RAM: векторная таблица во внутренней SRAM.

NVIC\_VectTab\_FLASH: векторная таблица во внутренней FLASH.

*Offset*, - поле смещения базового адреса векторной таблицы. Это значение должно быть кратно числу 0x200.

Возвращаемое значение:

Нет.

**void SysTick\_CLKSourceConfig(uint32\_t SysTick\_CLKSource);**

Описание:

Настраивает тактовый источник системного таймера.

Параметры:

SysTick\_CLKSource, - определяет тактовый источник SysTick. Этот параметр может принимать одно из следующих значений:

SysTick\_CLKSource\_HCLK\_Div8: AHB частота, поделенная на 8 выбрана как тактовый источник SysTick.

SysTick\_CLKSource\_HCLK: AHB тактовая частота выбрана как тактовый источник SysTick.

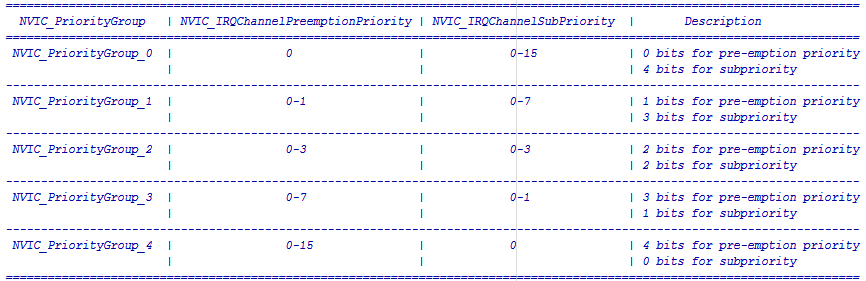
Возвращаемое значение:

Нет.

**Как настроить прерывания, используя драйвер misc.**

Исключения Cortex-M3 управляются функциями CMSIS.

1. Настроить группировку приоритета, используя функцию NVIC\_PriorityGroupConfig() в соответствии с следующей таблицей. В таблице ниже приведены допустимые значения ширины битового поля приоритета вытеснения и подприоритета в соответствии с желаемой настройкой группируемого приоритета, выполняемой функцией NVIC\_PriorityGroupConfig().



2. Разрешить и настроить приоритет выбранных каналов IRQ, используя NVIC\_Init().

Замечание: когда выбран NVIC\_PriorityGroup\_0, вытеснение другими IRQ больше невозможно. Ожидающий запрос прерывания будет обслужен только в соответствии с его подприоритетом.

Замечание: порядок следования приоритетов, отсортированный от наивысшего к наинизшему:

- Низший приоритет вытеснения.

- Низший подприоритет.

- Низший номер IRQ.

3. Добавить в проект файл-шаблон с заглушками обработчиков исключений (stm32f2xx\_it.c, stm32f2xx\_it.h). Добавить требуемые обработчики прерываний от периферии (гдн - там показано, в смысле, в файле). Имя обработчика должно соответствовать заданному в стартапе! (смотри файл startup\_stm32f2xx.s. В файле стартапа прописана векторная таблица.