Оглавление

Лабораторная работа № 12

Лабораторная работа № 220

Лабораторная работа № 334

Лабораторная работа № 447

ЛАБОРАТОРНАЯ РАБОТА №1.

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

1. Цель работы

Ознакомиться с общими принципами программирования микроконтроллеров.

1. Задание

В рамках лабораторной работы № 1 требуется:

1. Настроить схему синхронизации (тактирование) на основе внутреннего тактового генератора, разрешить тактирование для нужных компонентов МК;
2. Запрограммировать плату на мигание светодиодом;
3. Запрограммировать переключение режимов мигания с помощью джойстика.
4. Краткое пояснение к заданию
   1. Тактирование

Любая цифровая компонента системы может работать только тогда, когда на нее поступает тактовый сигнал – так осуществляется ее синхронизация с остальной системой. Например, работать с неким абстрактным портом ввода/вывода «X» микроконтроллера мы можем лишь тогда, когда он получает тактовый сигнал, т.е. синхронизирован с остальной системой

В целях энергосбережения многие компоненты микроконтроллеров отключаются от тактирования. То есть, если наш порт «X» по умолчанию отключен от тактирования, чтобы начать с ним работу, мы должны включить для него тактирование, включить его в схему синхронизации.

В этой работе Вам предстоит использовать светодиоды и джойстик, которые «запитаны» на порты ввода/вывода GPIOE и GPIOF соответственно. Значит, для того, чтобы использовать их в работе, Вам для начала нужно включить в схему тактирования порты GPIOE и GPIOF.

* 1. Тактовый генератор и корректировка частоты

Тактовый сигнал генерируется тактовым генератором (ТГ). В МК, который Вы будете программировать (STM32F3Disco), возможно использовать 4 вида:

* HSI (High Speed Internal), высокочастотный внутренний ТГ;
* LSI (Low Speed Internal), низкочастотный внутренний ТГ;
* HSE (High Speed External), высокочастотный внешний ТГ;
* LSE (Low Speed External), низкочастотный внешний ТГ.

Вам предстоит использовать HSI, который по умолчанию работает на частоте 8МГц.

Частота работы системы может корректироваться системой предделителей частоты (в англоязычной терминологии – «prescalers») и умножителей частоты (в англ. – «multipliers»). Так, если ТГ генерирует сигнал 8МГц, а нами используется 2x предделитель и 5x умножитель на блоке Фазовой Автоподстройки Частоты (ФАПЧ, PLL), то выходная частота будет составлять 8/2\*5=20МГц.

* 1. Программирование

После настройки схемы синхронизации можно начинать подавать напряжение на светодиоды и считывать манипуляции с джойстиком.

Мигание светодиодами в этой работе Вам предстоит реализовать с помощью цикла.

|  |
| --- |
| #include “stm32f3xx.h”  const uint32\_t RESET\_VALUE = 200000;  uint32\_t LED\_ON\_VALUE;  uint32\_t counter;  void init\_timing() {/\*Настраивает тактирование\*/;};  void led\_on() {/\*Включает светодиод\*/;}  void led\_off() {/\*Выключает светодиод \*/;}  int main(void) {  init\_timing();  LED\_ON\_VALUE = 50000;  while(1) {  counter = RESET\_VALUE;  led\_off();  while(counter--) {  if (counter == LED\_ON\_VALUE)  led\_on();  }  }  { |

*Листинг 1.1 – принципиальная структура кода для мигания светодиодами*

Поскольку исполнение кода осуществляется очень быстро, входное напряжение с джойстиков можно считывать непосредственно в основном цикле. В этом случае для смены режимом можно, например, менять значение «LED\_ON\_VALUE», чтобы изменить отношение времени простоя ко времени активности светодиода. Но ни в коем случае не используйте такой подход к работе с периферией в профессиональной практике – пользуйтесь *прерываниями*, если нужно обрабатывать вход с внешних устройств (их мы рассмотрим в лабораторной работе №2).

1. Общие сведения

Программирование микроконтроллера (МК) STM32 производится на языке C в среде Keil µVision. Суть программирования сводится к изменению значений в служебных регистрах МК для настройки и управления, плюс написанию Вашего собственного пользовательского кода.

|  |
| --- |
| #include "stm32f3xx.h" // Device header  void enable {  RCC -> AHBENR |= 0x00200000; // Enable GPIOF  }  int main(void) {  enable();  }  // ... |

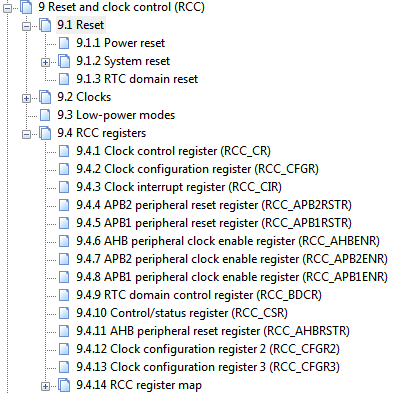
*Листинг 1.2 – принципиальная схема программирования МК STM32 в среде Keil* *µVision*

Обратите внимание на подключаемый файл «stm32f3xx.h». Он содержит в себе список имён регистров, сопоставленных со своими машинными адресами. Без его использования Вам требовалось бы обращаться к регистрам по их машинным адресам.

|  |
| --- |
| // Именование регистров строится примерно по такому  // принципу  #define RCC\_BDCR\_LSEDRV\_0 ((uint32\_t)0x00000008) // bit0  #define RCC\_BDCR\_LSEDRV\_1 ((uint32\_t)0x00000010) // bit1 |

*Листинг 1.3 – принцип именования регистров*

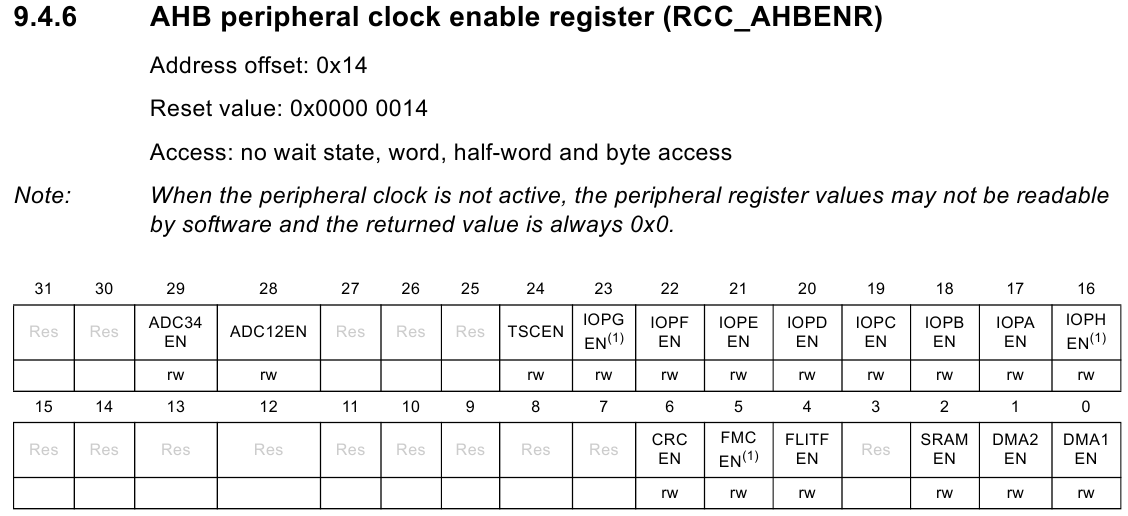
В документации, прилагаемой к МК STM32, для удобства пользователя регистры объединены в группы, и схема именования, предлагаемая файлом «stm32f3xx.h», в точности отвечает схеме, предлагаемой документацией (см. прилагаемый файл «STM32Disco\_ReferenceManual.pdf»).



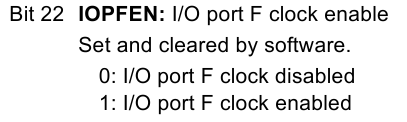
*Рисунок 1.1 – отрывок из содержания документации*

Сопоставьте листинг 1.1 с данным рисунком. В нём (листинге) мы использовали регистр «AHBENR» из группы регистров «RCC». Точно по такому же принципу производится обращение ко всем регистрам.

В том же листинге (листинг 1.1) мы присвоили регистру «RCC\_AHBENR» значение «0x00200000», что соответствует команде «включить порт ввода/вывода F». Это понятно из карты битов для регистра RCC\_AHBENR и приводимого к ней описания (сопоставьте рисунки 1.2 и 1.3, скриншоты описаний, взятых из документации). Устанавливая значения битов служебных регистров, мы управляем микроконтроллером.



*Рисунок 1.2 – карта битов для регистра RCC\_AHBENR*

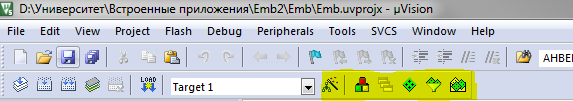


*Рисунок 1.3 – пояснение функциональности бита №22 регистра RCC\_AHBENR*

Ниже для всех необходимых для выполнения работы регистров будет дано полное табличное описание их конфигурирующих битов.

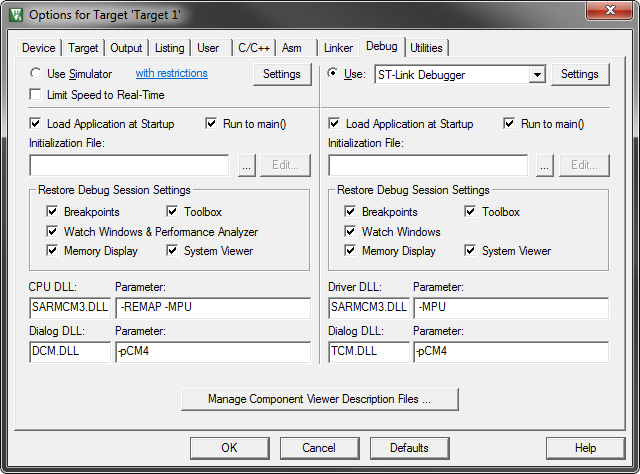
1. Настройка среды Keil µVision

После того, как Вы откроете среду разработки, обратите внимание на панель, выделенную на рисунке 1.4.



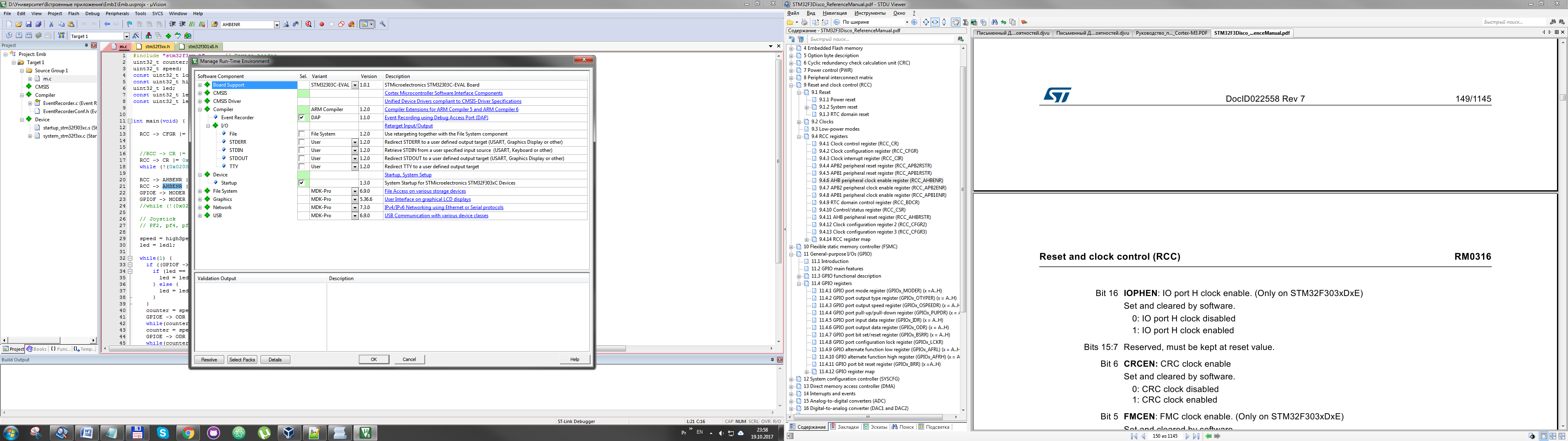
*Рисунок 1.4 – панель настройки*

Нажмите на первую иконку в выделенном фрагменте, перейдите на вкладку «Debug» и сконфигурируйте свой проект в точности так, как показано на рисунке 1.5.



*Рисунок 1.5 – панель настройки «Options for Target»*

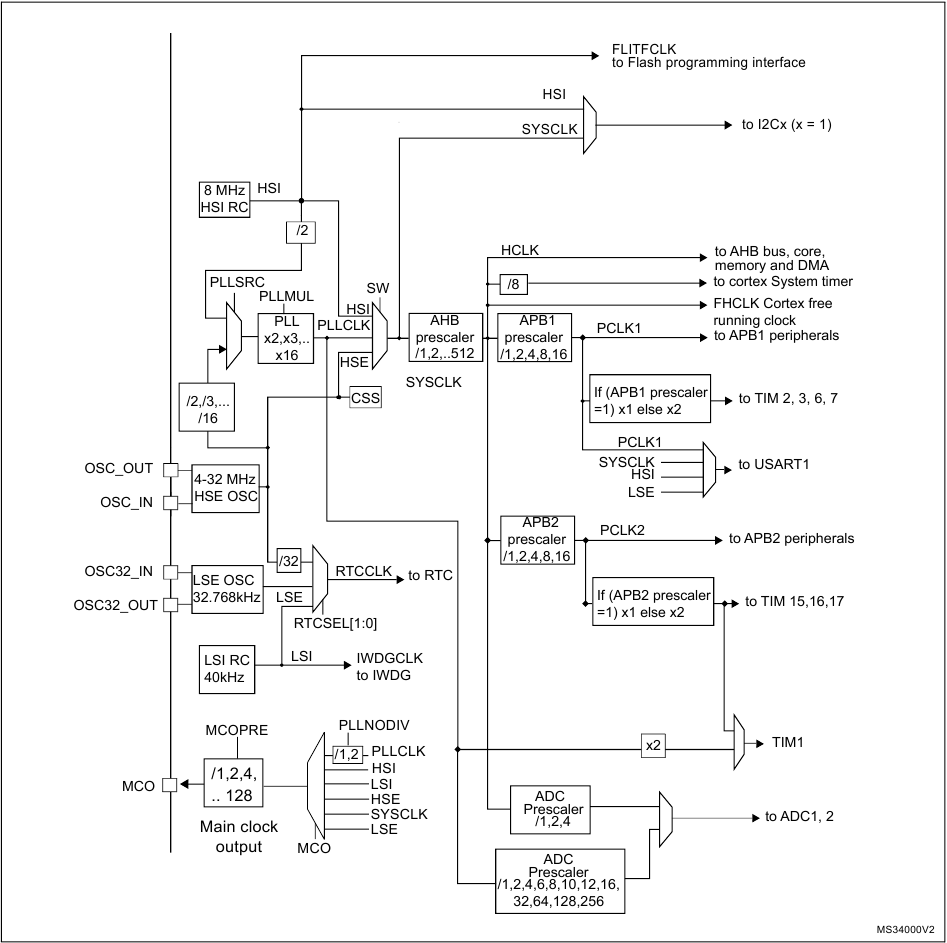
Перейдите в панель «Manage Run-Time Environment» (рисунок 1.4, выделенный фрагмент, 4-я иконка слева) и настройте свой проект по образцу, указанному на рис. 1.6.



*Рисунок 1.6 – панель настройки «Manage Run-Time Environment»*

Среди прилагаемых к методическим указаниям архивов имеется архив «en.stwe-link-009.rar» с драйверами STM32 для Windows. Для того чтобы подключенный МК опознавался Вашей системой, установите драйвер из этого архива. Обратите внимание, что их (драйверов) два: для архитектур «x86» и «amd64».

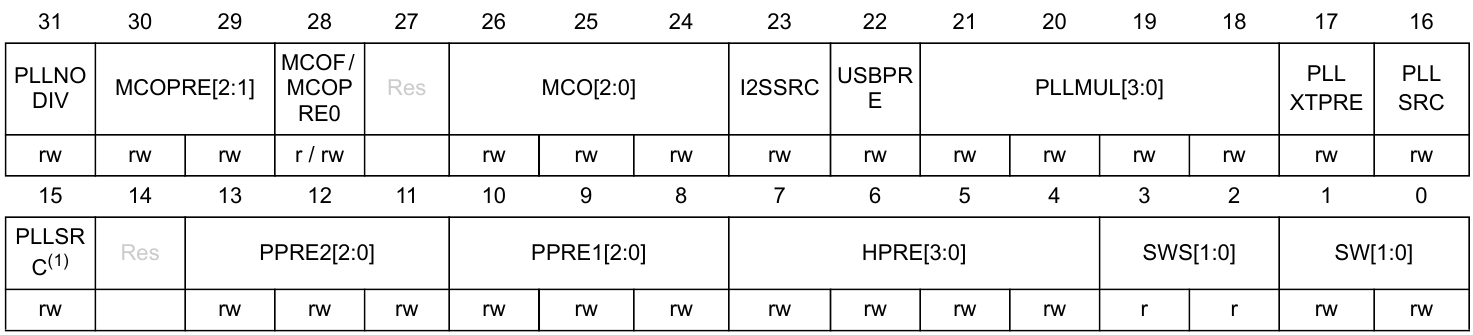
1. RCC



*Рисунок 1.7 – схема тактирования*

Приведённая схема служит иллюстрацией для регистров группы RCC, подробное описание которых даётся ниже.

* 1. RCC\_CFGR

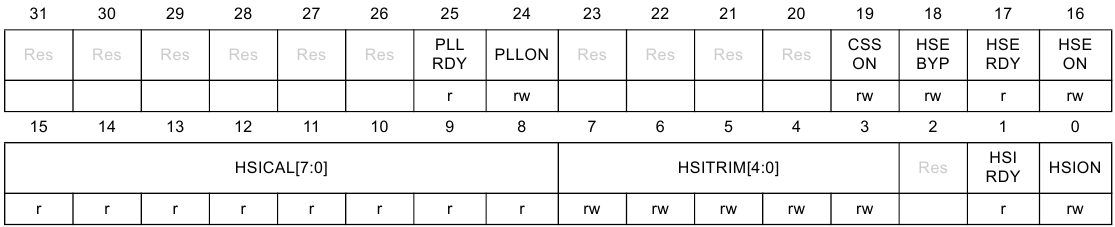


*Рисунок 1.8 – карта битов для регистра RCC\_CFGR*

*Таблица 1.1 – описание битов для регистра RCC\_CFGR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31 | Только для STM32F303x6/8, STM32F328x8, STM32F303xDxE, STM32F398xE.  PLLNODIV: не использовать предделение при выходе с блока Фазовой автоподстройки частоты (ФАПЧ, PLL) на выход схемы тактирования МК (ВСТ МК, MCO).  Значения устанавливаются программно. Используется для отключения двукратного предделителя (out/2) с выхода ФАПЧ на выход СТ.   * 0: установить предделение с ФАПЧ на ВСТ на 2; * 1: не устанавливать предделение с ФАПЧ на ВСТ. |
| 30:28 | Только для STM32F303x6/8, STM32F328x8, STM32F303xDxE, STM32F398xE.  MCOPRE: предделитель выхода схемы тактирования (ВСТ).  Значения устанавливаются программно. Рекомендуется устанавливать их до активации ВСТ.   * 000: предделитель ВСТ на 1; * 001: предделитель ВСТ на 2; * 010: предделитель ВСТ на 4; * 011: предделитель ВСТ на 8;   …   * 111: предделитель ВСТ на 128. |
| 28 | Только дляSTM32F303xB/C, STM32F358xC.  MCOF: флаг ВСТ.  Значения устанавливаются аппаратно.   * 0: в поле MCO (см. биты 26:24) записано новое значение; * 1: источник тактирования, указанный в поле MCO, готов к работе. |
| 27 | Зарезервирован. |
| 26:24 | MCO: выход схемы тактирования (ВСТ).  Значения устанавливаются программно. Используется для выбора источника тактового сигнала.   * 000: ВСТ неактивен, источник тактирования не указан; * 001: зарезервировано; * 010: Низкочастотный внутренний тактовый генератор (LSI clock); * 011: Низкочастотный внешний тактовый генератор (LSE clock); * 100: Системный тактовый генератор (системый ТГ, SYSCLK); * 101: Высокочастотный внутренний тактовый генератор (HSI clock); * 110: Высокочастотный внешний тактовый генератор (HSE clock); * 111: Блок фазовой автоподстройки частоты (ФАПЧ, PLL), возможно деление на 2 в зависимости от значения бита PLLNODIV (31-й бит, см. выше). |
| 23 | Только для STM32F303xB/C/D/E, STM32F358xC, STM32F398xE.  I2SSRC: выбор источника тактирования для порта I2S.  Значения устанавливаются программно. Используется для тактирования портов I2S2и I2S3 с помощью внешнего ТГ. Значение этого бита должно быть установлено до того, как ТГ для I2S2(3) будет активирован.   * 0: I2S2 и I2S3 тактируются системным ТГ; * 1: I2S2 и I2S3 тактируются внешним ТГ. |
| 22 | USBPRE: предделитель USB.  Значения устанавливаются программно. Используются для генерации 48 МГц тактового сигнала для USB. Значение этого бита должно быть установлено до того, как источник тактового сигнала для USB будет активирован.   * 0: сигнал с ФАПЧ делится на 1,5; * 1: сигнал с ФАПЧ не делится. |
| 21:18 | PLLMUL: множитель ФАПЧ (PLL).  Значения устанавливаются программно. Используются для задания коэффициента умножения выходного сигнала для блока Фазовой автоподстройки частоты (ФАПЧ). Значения этих битов должны быть установлены до того, как блок ФАПЧ будет включён.  Внимание: выходная частота с блока ФАПЧ не должна превышать 72 МГц.   * 0000: частота входного сигнала ФАПЧ x 2; * 0001: частота входного сигнала ФАПЧ x 3;   …   * 1110: частота входного сигнала ФАПЧ x 16; * 1111: частота входного сигнала ФАПЧ x 16. |
| 17 | PLLXTPRE: предделитель с высокочастотного внешнего ТГ (HSE) на ФАПЧ (PLL).  Значения устанавливаются программно. Используется для выбора коэффициента деления частоты с выхода HSE на вход ФАПЧ.  Замечание: этот бит означает то же, что и поле PREDIV в регистре RCC\_CFGR2 (см. документацию): используется для совместимости с другими МК STM32.   * 0: предделение отключено; * 1: предделение на 2 включено. |
| 16:15 | Только для STM32F303xD/E, STM32F398xE.  PLLSRC: источник тактирования для ФАПЧ (PLL).  Значения устанавливаются программно. Используются для задания источника тактирования блока ФАПЧ. Значения этих битов должны быть установлены до того, как блок ФАПЧ будет включён.   * 00: высокочастотный внутренний ТГ, частота делится на 2 (HSI/2); * 01: высокочастотный внутренний ТГ (HSI), делитель устанавливается полем PREDIV в рг. RCC\_CFGR2 (см. документацию); * 10: высокочастотный внешний ТГ (HSE), делитель устанавливается полем PREDIV в рг. RCC\_CFGR2; * 11: зарезервировано. |
| 16 | Только для STM32F303xB/C, STM32F358xC, STM32F303x6/8, STM32F328x8.  PLLSRC: источник тактирования для ФАПЧ (PLL).  Значения устанавливаются программно. Используются для задания источника тактирования блока ФАПЧ. Значения этих битов должны быть установлены до того, как блок ФАПЧ будет включён.   * 0: высокочастотный внутренний ТГ, частота делится на 2 (HSI/2); * 1: высокочастотный внешний ТГ, делитель устанавливается полем PREDIV в рг. RCC\_CFGR2. |
| 15 | Зарезервирован для устройств STM32F303xD/E. |
| 14 | Зарезервирован. |
| 13:11 | PPRE2: предделитель для шины APB2.  Значения устанавливаются программно. Используется для задания делителя частоты на шине APB2.   * 0xx: деление не используется; * 100: деление на 2; * 101: деление на 4; * 110: деление на 8; * 111: деление на 16. |
| 10:8 | PPRE1: предделитель для шины APB1.  Значения устанавливаются программно. Используется для задания делителя частоты на шине APB2.   * 0xx: деление не используется; * 100: деление на 2; * 101: деление на 4; * 110: деление на 8; * 111: деление на 16. |
| 7:4 | HPRE: предделитель шинного преобразователя AHB.  Значения устанавливаются программно. Используется для задания делителя частоты на выходе шинного преобразователя.   * 0xxx: деление не используется; * 1000: деление на 2; * 1001: деление на 4; * 1010: деление на 8; * 1011: деление на 16; * 1100: деление на 64; * 1101: деление на 128; * 1110: деление на 256; * 1111: деление на 512. |
| 3:2 | SWS: переключение системного ТГ.  Значения устанавливаются аппаратно. Используется для определения, какой из ТГ используется в качестве системного.   * 00: Высокочастотный внутренний ТГ (HSI); * 01: Высокочастотный внешний ТГ (HSE); * 10: ФАПЧ (PLL); * 11: не используется. |
| 1:0 | SW: переключение системного ТГ.  Значения устанавливаются программно для определения системного ТГ.  Значения устанавливаются аппаратно, автоматически, в пользу высокочастотного внутреннего ТГ (HSI clock) при работе в режиме «Stop» или «Standby» или в случае отказа внешнего высокочастотного ТГ (HSE), если включена система защиты схемы тактирования (Clock Security System, CSS).   * 00: Высокочастотный внутренний ТГ (HSI); * 01: Высокочастотный внешний ТГ (HSE); * 10: ФАПЧ (PLL); * 11: не используется. |

* 1. RCC\_CR

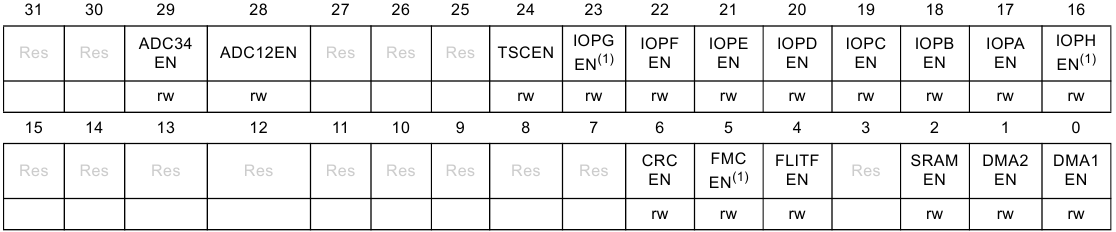


*Рисунок 1.9 – карта битов для регистра RCC\_CR*

*Таблица 1.2 – описание битов для регистра RCC\_CR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:26 | Зарезервировано. |
| 25 | PLLRDY: флаг готовности блока ФАПЧ (PLL).  Значения устанавливаются аппаратно. Используется для индикации блокировки ФАПЧ.   * 0: ФАПЧ разблокирован; * 1: ФАПЧ заблокирован. |
| 24 | PLLON: бит включения блока ФАПЧ (PLL).  Значения устанавливаются программно. Используется для включения блока ФАПЧ.  Значения могут быть установлены аппаратно при входе МК в режимы «Stop» или «Standby». Эти значения не могут быть изменены, если блок ФАПЧ используется в качестве системного ТГ.   * 0: ФАПЧ выключен; * 1: ФАПЧ включен. |
| 23:20 | Зарезервировано. |
| 19 | CSSON: система защиты схемы тактирования (CSS, Clock Security System).  Значения устанавливаются программно. Используется для включения системы защиты схемы тактирования.  После включения системы защиты значение данного поля используется как индикатор работы внешнего высокочастотного тактового генератора (внешний ВЧТГ, HSE), и в случае его (HSE) отказа сбрасывается на 0.   * 0: система защиты отключена / произошла ошибка внешнего ВЧТГ (HSE); * 1: система защиты включена / внешний ВЧТГ функционирует нормально. |
| 18 | HSEBYP: разрешение на использование внешнего ВЧТГ (HSE).  Значения устанавливаются программно. Используется для разрешения использования внешнего ТГ. Значения могут быть изменены только в том случае, если поле «HSEON» сброшено.   * 0: использование внешнего ВЧТГ запрещено; * 1: использование внешнего ВЧТГ разрешено. |
| 17 | HSERDY: флаг готовности внешнего ВЧТГ (HSE).  Значения устанавливаются аппаратно. Используется для индикации готовности внешнего ВЧТГ.   * 0: внешний ВЧТГ (HSE) не готов; * 1: внешний ВЧТГ (HSE) готов. |
| 16 | HSEON: включение внешнего ВЧТГ (HSE).  Значения устанавливаются программно. Используется для включения внешнего ВЧТГ.  Значения устанавливаются аппаратно при входе МК в режимы «Stop» или «Standby». Значение этого бита не может быть изменено, если внешний ВЧТГ используется в качестве системного ТГ.   * 0: внешний ВЧТГ отключен; * 1: внешний ВЧТГ включен. |
| 15:8 | HSICAL: частотные значения внутреннего ВЧТГ (HSI).  Значения устанавливаются аппаратно, автоматически при запуске МК. |
| 7:3 | HSITRIM: корректировка частоты внутреннего ВЧТГ (HSI).  Значения устанавливаются программно. Установленное значение прибавляется к значению поля «HSICAL», что позволяет корректировать частоту, нивелируя воздействие напряжения и температуры окружающей среды.  По умолчанию, значение равняется 16. Будучи добавленным к значению поля «HSICAL», оно должно довести выходную частоту внутреннего ВЧТГ до 8 МГц (± 1%). Шаг подстройки между двумя последовательными шагами HSICAL примерно 40 кГц. |
| 2 | Зарезервировано |
| 1 | HSIRDY: флаг готовности внутреннего ВЧТГ (HSI).  Значения устанавливаются аппаратно. Используется для индикации готовности внутреннего ВЧТГ. После того, как значение поля «HSION» сбрасывается, «HSIRDY» сбрасывается вслед за ним по прошествии 6 тактов внутреннего ВЧТГ.   * 0: внутренний ВЧТГ не готов; * 1: внутренний ВЧТГ готов. |
| 0 | HSION: включение внутреннего ВЧТГ (HSI).  Значения устанавливаются программно.  В принудительном порядке (аппаратно) устанавливается значение 1 при выходе из режимов «Stop» или «Standby» и в случае сбоя в работе внешнего ВЧТГ (HSE), если тот использовался в качестве системного ТГ.  Значение этого бита не может быть изменено, если внутренний ВЧТГ (HSI) установлен в качестве системного ТГ.   * 0: внутренний ВЧТГ выключен; * 1: внутренний ВЧТГ включен. |

* 1. RCC\_AHBENR



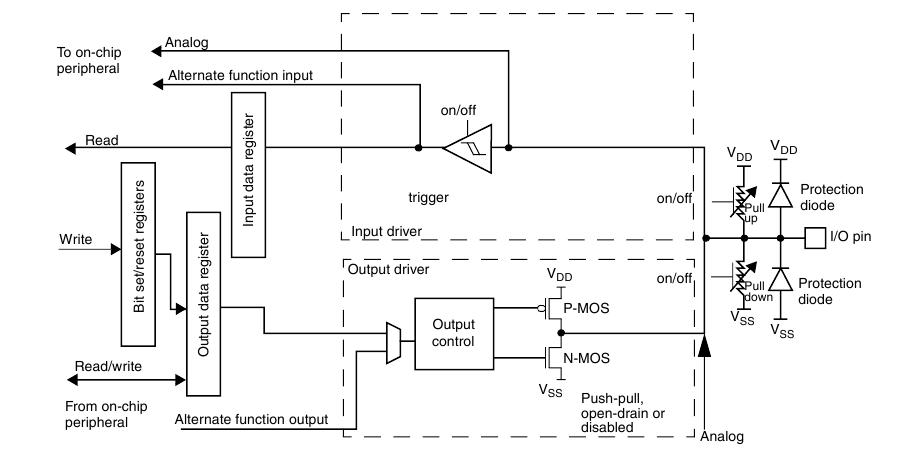
*Рисунок 1.10 – карта битов для регистра RCC\_AHBENR*

*Таблица 1.3 – описание битов для регистра RCC\_AHBENR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:30 | Зарезервировано. |
| 29 | Только для STM32F303xB/C/D/E, STM32F358xC, STM32F398xE.  ADC34EN: включение тактирования аналого-цифровых преобразователей (АЦП, ADC) 3 и 4.  Значения устанавливаются программно. Используется для включения АЦП3 и АЦП4   * 0: АЦП3 и АЦП4 выключены; * 1: АЦП3 и АЦП4 включены. |
| 28 | ADC12EN: включение тактирования аналого-цифровых преобразователей (АЦП, ADC) 1 и 2.  Значения устанавливаются программно. Используется для включения АЦП1 и АЦП2   * 0: АЦП1 и АЦП2 выключены; * 1: АЦП1 и АЦП2 включены. |
| 27:25 | Зарезервирован. |
| 24 | TSCEN: включение тактирования контроллера датчика прикосновений (TSC).  Значения устанавливаются программно.   * 0: TSC выключен; * 1: TSC включен. |
| 23 | IOPGEN: включение тактирования порта ввода/вывода G.  Значения устанавливаются программно.   * 0: тактирование порта G выключено; * 1: тактирования порта G включено. |
| 22 | IOPFEN: включение тактирования порта ввода/вывода F.  Значения устанавливаются программно.   * 0: тактирование порта F выключено; * 1: тактирование порта F включено. |
| 21 | IOPEEN: включение тактирования порта ввода/вывода E.  Значения устанавливаются программно.   * 0: тактирование порта E выключено; * 1: тактирование порта E включено. |
| 20 | IOPDEN: включение тактирования порта ввода/вывода D.  Значения устанавливаются программно.   * 0: тактирование порта D выключено; * 1: тактирование порта D включено. |
| 19 | IOPCEN: включение тактирования порта ввода/вывода C.  Значения устанавливаются программно.   * 0: тактирование порта C выключено; * 1: тактирование порта C включено. |
| 18 | IOPBEN: включение тактирования порта ввода/вывода B.  Значения устанавливаются программно.   * 0: тактирование порта B выключено; * 1: тактирование порта B включено. |
| 17 | IOPAEN: включение тактирования порта ввода/вывода A.  Значения устанавливаются программно.   * 0: тактирование порта A выключено; * 1: тактирование порта A включено. |
| 16 | IOPHEN: включение тактирования порта ввода/вывода H.  Значения устанавливаются программно.   * 0: тактирование порта H выключено; * 1: тактирование порта H включено. |
| 15:7 | Зарезервировано. |
| 6 | CRCEN: включение тактирования для блока генерации CRC-кода.  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирования включено. |
| 5 | FMCEN: включение тактирования блока FMC.  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирование включено. |
| 4 | FLITFEN: включения тактирования блока доступа к Flash (FLITF).  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирование включено. |
| 3 | Зарезервировано. |
| 2 | SRAMEN: включение тактирования интерфейса SRAM.  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирование включено. |
| 1 | DMA2EN: включение тактирования блока DMA2 (блок прямого доступа к памяти).  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирование включено. |
| 0 | DMA2EN: включение тактирования блока DMA1 (блок прямого доступа к памяти).  Значения устанавливаются программно.   * 0: тактирование выключено; * 1: тактирование включено. |

1. GPIOx

GPIO, General Puprose Input/Output, Порты ввода-вывода общего назначения.



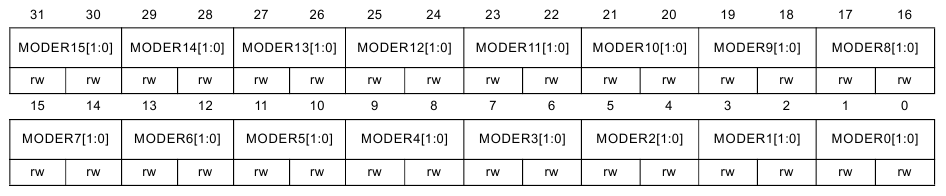
*Рисунок 1.7 – принципиальная схема порта ввода/вывода*

Каждый GPIOx-порт (GPIOA, GPIOB и т.д.) содержит:

* 4 32-битных конфигурационных регистра (MODER, OTYPER, SPEEDR, PUPDR);
* 2 32-битных регистра данных (IDR, ODR – Input/Output data register);
* 32-битный регистр установки/сброса (BSRR – Bit set/reset register);
* 32-битный регистр блокировки (LCKR);
* 2 32-битных регистра выбора альтернативных функций (Alternate function, регистры AFRH, AFRL).

В этой работе Вам понадобятся регистры MODER, IDR, ODR. Использоваться будут группы GPIOE и GPIOF.

* 1. GPIOx\_MODER

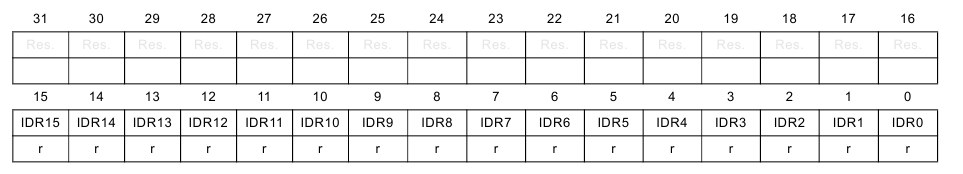


*Рисунок 1.11 – карта битов для регистра GPIOx\_MODER*

*Таблица 1.4 – описание битов для регистра GPIOx\_MODER*

|  |  |
| --- | --- |
| Бит | Описание |
| <x>:<x-1> | MODER: используются для конфигурирования режима ввода/вывода.  Значения устанавливаются программно.   * 00: режим ввода; * 01: режим вывода общего назначения; * 10: режим альтернативной функции; * 11: аналоговый режим. |

* 1. GPIOx\_IDR, регистр ввода

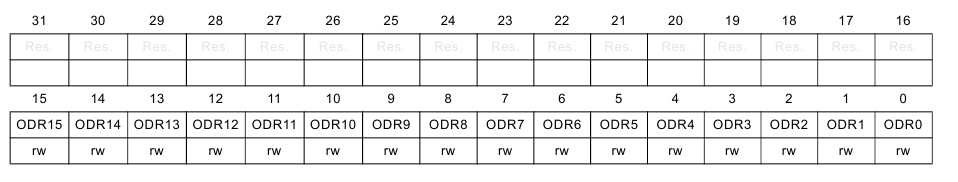


*Рисунок 1.12 – карта битов для регистра GPIOx\_IDR*

*Таблица 1.5 – описание битов для регистра GPIOx\_IDR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Зарезервированы. |
| 15:0 | IDR: для ввода битов данных. Хранят входное значение, поступающее от порта ввода/вывода.  Только для чтения. |

* 1. GPIOx\_ODR, регистр вывода



*Рисунок 1.13 – карта битов для регистра GPIOx\_ODR*

*Таблица 1.6 – описание битов для регистра GPIOx\_ODR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Зарезервированы. |
| 15:0 | ODR: хранят выходное значение.  Значения устанавливаются программно.  Для установки значений отдельных битов из этого набора можно использовать реигстры GPIOx\_BSRR и GPIOx\_BRR.  *Описание BSRR и BRR здесь не приводится. См. «STM32F3Disco\_ReferenceManual», стр. 240 и 242.* |

ЛАБОРАТОРНАЯ РАБОТА №2.

ПРЕРЫВАНИЯ

1. Цель работы

Ознакомиться с прерываниями и их обработкой

1. Задание

В рамках лабораторной работы № 2 требуется:

1. Настроить схему синхронизации;
2. разрешить прерывания для таймера, написать обработчик прерываний;
3. запрограммировать мигание светодиодом, пользуясь прерываниями по таймеру.
4. Общее представление о прерываниях
   1. Прерывания

В предыдущей лабораторной работе мы осуществляли мигание светодиодом, подавая напряжение на светодиод, используя в качестве «ориентира» значение переменной-счетчика. В этот раз управлять миганием светодиода мы будем с использованием прерываний от таймера.

Когда процессор получает сигнал прерывания, он приостанавливает выполнение основного кода и переходит к выполнению *кода обработчика* данного конкретного прерывания. Завершив обработку прерывания, он снова возвращается к основному коду.

Прерывания могут делиться по приоритету. В STM32 установлен NVIC/КВВП (Nested Vectored Interruprs Controller / Контроллер Вложенных Векторных Прерываний). Вложенность их характеризуется тем, что пользователь может назначить прерыванию приоритет, и тогда при срабатывании высокоприоритетного прерывания во время выполнения низкоприоритетного процессор перейдет к выполнению высокоприоритетного, а низкоприоритетное поставит в очередь (см. рисунок ниже).



*Рисунок 2.1 – вложенные прерывания*

В вашей работе Вам следует использовать обработчик прерывания таймера 2. Он задается такой функцией:

|  |
| --- |
| \_\_irq void TIM2\_IRQHandler (void) {  /\*Тело обработчика\*/;  } |

*Листинг 2.1 – тело обработчика прерываний*

Заметьте, что Вам точно предписывается объявление функции. Дело в том, что к каждому источнику прерываний жестко привязывается обработчик прерываний, адрес которого хранится в аппаратном векторе прерываний. Так, когда компилятор среды «Keil µVision» встречает функцию *«\_\_irq void TIM2\_IRQHandler (void)*», он размещает ее код в области памяти, указанной в векторе прерываний.

Однако в предыдущей работе мы не писали никаких обработчиков прерываний, хотя и работали с внешним устройством – джойстиком. Все потому, что нужно предварительно разрешить прерывание для конкретного потенциального источника прерываний (например, входа, который используется джойстиком), и только после этого станет возможно использование обработчика.

* 1. Особенности настройки прерываний в STM32

В STM32 используется усложненная терминология. В английской терминологии STM32 существуют понятия «событие» и «прерывание» («Event» и «Interrupt»). События, некие сигналы, генерируются аппаратным обеспечением микроконтроллера и могут порождать реакцию со стороны системы, в том числе – порождать прерывания. Так, сигнал от внешнего устройства (например, изменение значения бита по нарастающему фронту, от 0 до 1) – это событие, которое, в зависимости от настроек, может порождать ту или иную реакцию системы, в том числе – и прерывание.

Чтобы обрабатывать прерывания от внешних устройств, их выходы должны быть запитаны на так называемые «линии внешних прерываний». Именно для «линий внешних прерываний» назначаются разрешения и запреты на прерывание. Так, подключив устройство к «линии внешних прерываний», мы можем генерировать исключения при поступающем от линии сигнале.

«Линии внешних прерываний» обозначаются как «EXTI lines» или «External Interrupts and Events Lines» и настраиваются группой регистров EXTI IMR/RTSR/FTSR, а также регистром SYSCFG\_EXTICR.

С *внутренними* устройствами все проще – их не надо подключать к линиям внешних прерываний, поскольку для них уже зарезервированы биты, разрешающие и запрещающие прерывания. Поскольку мы с Вами используем *внутренний таймер*, а не внешние источники прерываний, все, что нам нужно сделать – сконфигурировать сам таймер (об этом позже) и разрешить для него прерывания непосредственно, минуя этап с «линиями внешних прерываний».



*Рисунок 2.2 – о разнице в назначении прерываний для внешних и внутренних источников прерываний*

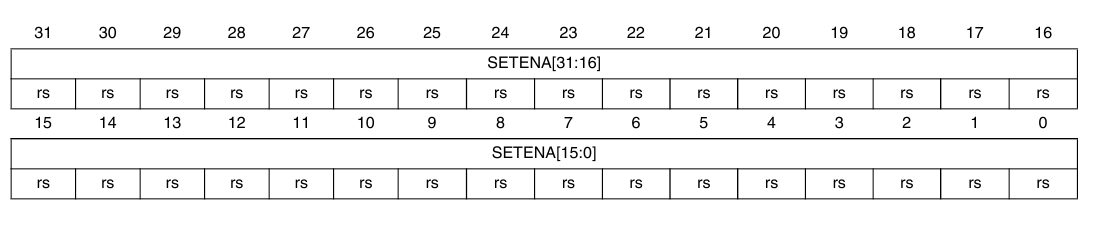
Пояснения к рис. 2.2. Для линии 1 – ножка A, прерывания запрещены, для линии 2 – ножка B, прерывания разрешены, для внутреннего таймера 2 – прерывания разрешены (дополнительно ничего указывать не надо).

Чтобы разрешить прерывания для таймера, сделайте следующее. В таблице 82 документа «STM32Disco\_ReferenceManual» на странице 291 найдите в столбце «Description» второй таймер (TIM2). Запомните его позицию («Position»). После этого в регистре NVIC\_ISER0 для бита, позиция которого соответствует найденной нами позиции для TIM2, установите значение 1. Тем самым вы разрешите прерывания для таймера 2.

Таким образом, таблица 82 из «ReferenceManual» устанавливает соответствие между вектором прерываний битами, отвечающими за разрешение прерываний для каждого из источников.

Регистр NVIC\_ISERx («ISER0») описан в документе «STM32F3Disco\_ProgrammingManual», перевод ниже.

1. NVIC\_ISERx (ISER0 в рамках данной работы)



*Рисунок 2.3 – карта битов для регистра NVIC\_ISERx*

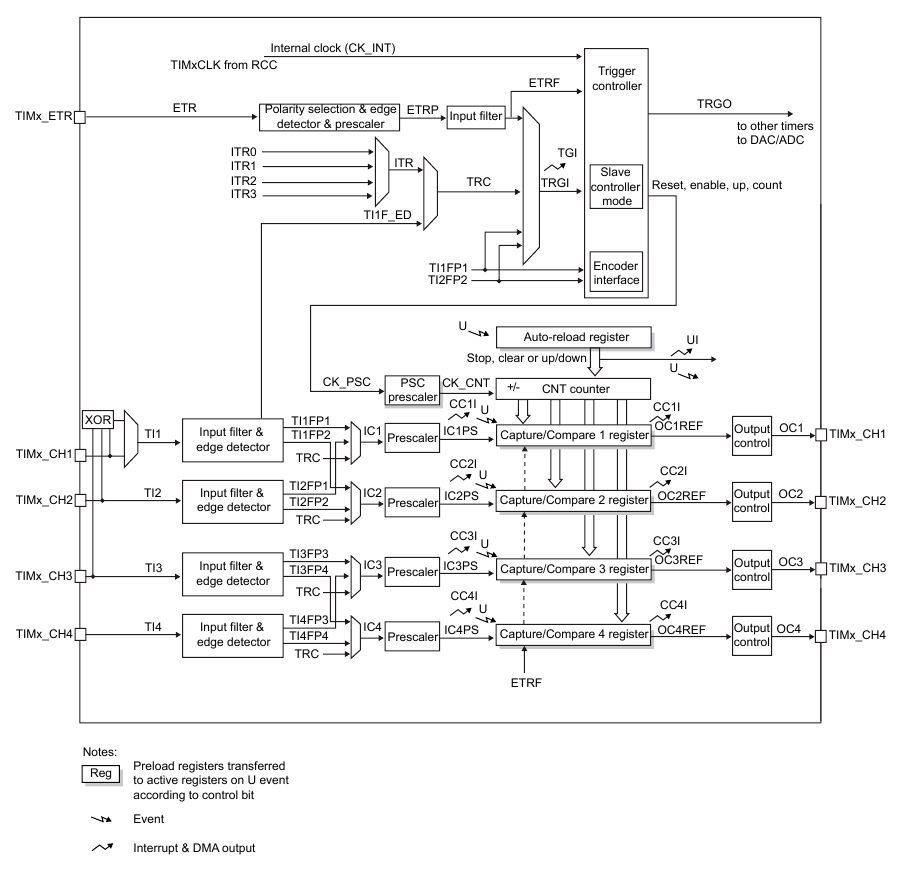
*Таблица 2.1 – описание битов для регистра NVIC\_ISERx*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:0 | SETENA: используется для активации прерываний для выбранного источника.  Устанавливается программно.  При записи:   * 0: нет действия; * 1: активация прерываний.   При чтении:   * 0: прерывания отключены; * 1: прерывания включены. |

Для нашей с Вами платы определен 81 источник прерываний. Так, при обращении к ISER0 мы получаем доступ к 0:31 источникам, к ISER1 – к источникам 32:63, к ISER2 – 64:81.

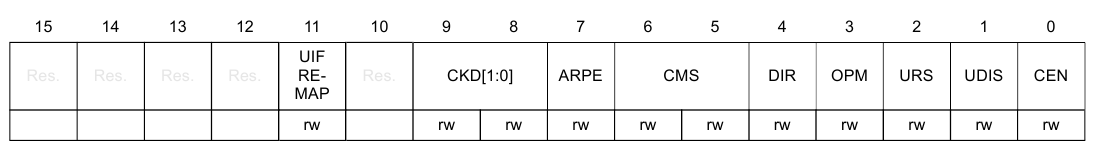
Обратите внимание. В таблице 2.1 указано, что при записи 0 в соответствующий бит ничего не происходит, хотя логично предположить, что запись «0» отключает прерывание. Для отключения прерывания используется регистр NVIC\_ICERx, который имеет точно такую же структуру, как и «ISERx». В рамках данной работы отключать прерывания не потребуется, но имейте в виду эту особенность. Для подробных сведений обратитесь к документу «STM32F3Disco\_ProgrammingManual», раздел 3.4, «NVIC».

1. TIMx (TIM2 в рамках данной работы)



*Рисунок 2.4 – таймер общего назначения*

* 1. TIM2\_CR1



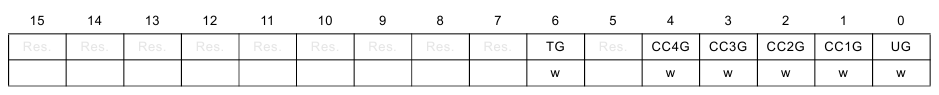
*Рисунок 2.5 – карта битов для регистра TIM2\_CR1*

*Таблица 2.2 – описание битов для регистра TIM2\_CR1*

|  |  |
| --- | --- |
| Бит | Описание |
| 15:12 | Зарезервировано |
| 11 | UIFREMAP: переназначение бита статуса UIF.  Значения устанавливаются программно.   * 0: нет переназначения, бит статуса UIF не копируется в бит 31 регистра TIMx\_CNT; * 1: переназначение включено, бит статуса UIF копируется в бит 31 регистра TIMx\_CNT. |
| 10 | Зарезервировано |
| 9:8 | CKD: предделитель частоты.  Устанавливает величину отношения частоты таймера CK\_INT к частоте счетчиков, используемых цифровыми фильтрами (ETR, TIx)  Значения устанавливаются программно.   * 00: tDTS = tCK\_INT; * 01: tDTS = 2\*tCK\_INT; * 10: tDTS = 4\*tCK\_INT; * 11: Зарезервировано. |
| 7 | ARPE: загрузка автоперезагружаемого значения таймера  Значения устанавливаются программно.   * 0: регистр TIMx\_ARR не буферизуется; * 1: регистр TIMx\_ARR буферизуется. |
| 6:5 | CMS: режим выравнивания по центру.  Значения устанавливаются программно.   * 00: выравнивание «по грани». Счетчик увеличивается или уменьшается в зависимости от значения бита DIR; * 01: выравнивание по центру 1. Счетчик поочередно увеличивается и уменьшается. Выходные флаги сравнения значений каналов, сконфигурированные на выходе (CCxS в регистре TIMx\_CCMRx установлены в «00»), устанавливаются только когда счетчик уменьшается; * 10: выравнивание по центру 2. Счетчик поочередно увеличивается и уменьшается. Выходные флаги сравнения значений каналов, сконфигурированные на выходе (CCxS в регистре TIMx\_CCMRx установлены в «00»), устанавливаются только когда счетчик увеличивается; * 11: выравнивание по центру 3. Счетчик поочередно увеличивается и уменьшается. Выходные флаги сравнения значений каналов, сконфигурированные на выходе (CCxS в регистре TIMx\_CCMRx установлены в «00»), устанавливаются в обоих случаях. |
| 4 | DIR: направление изменения счетчика.  Значения устанавливаются программно.   * 0: Счетчик увеличивается; * 1: Счетчик уменьшается. |
| 3 | OPM: генерация единичного импульса (режим одновибратора)  Значения устанавливаются программно.   * 0: Счетчик не останавливается при прерывании; * 1: счетчик останавливается при прерывании, очищая при этом бит CEN. |
| 2 | URS: Источник прерываний  Значения устанавливаются программно.   * 0: Любое из перечисленных ниже событий заставляет таймер генерировать сигнал прерывания или запрос DMA/ПДП (Direct Memory Access, Прямой Доступ к Памяти).   + Переполнение счетчика;   + установка бита UG;   + сигнал от подчиненного контроллера; * 1: Только переполнение счетчика генерирует сигнал прерывания или DMA-запрос. |
| 1 | UDIS: отключение сигналов прерываний.  Значения устанавливаются программно.   * 0: Сигнал прерываний генерируется по наступлению следующих изменений:   + Переполнение счетчика;   + установка бита UG;   + сигнал от подчиненного контроллера. * 1: события отключены. Событие не генерируется, теневые регистры сохраняют свои значения (ARR, PSC, CCRx). Тем не менее, счетчик и предделитель обновляются, если установлен бит UG или от подчиненного контроллера получен сигнал перезагрузки. |
| 0 | CEN: включение счетчика.  Значения устанавливаются программно.   * 0: Счетчик отключен; * 1: счетчик включен.   Внешний таймер, стробированный режим и режим энкодера могут работать только в том случае, если бит CEN был предварительно установлен пользователем. Тем не менее, CEN может быть установлен в режиме триггера, аппаратно (т.е. автоматически).  CEN автоматически обнуляется в режиме одновибратора, при наступлении изменений, приводящих к генерации сигнала прерывания. |

* 1. TIMx\_EGR

Используется для генерации сигналов об изменениях, прямо или косвенно приводящих к генерации сигнала прерывания.

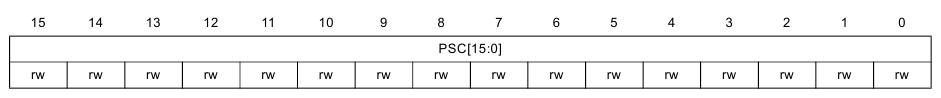


*Рисунок 2.6 – карта битов для регистра «TIMx\_EGR»*

*Таблица 2.3 – описание битов для регистра «TIMx\_EGR»*

|  |  |
| --- | --- |
| Бит | Описание |
| 15:7 | Зарезервировано |
| 6 | TG. «Триггеринг» изменений  Значения устанавливаются программно.   * 0: нет действия; * 1: устанавливает флаг «TIF» в регистре TIMx\_SR. Используется как логический атрибут при принятии системой решения о генерации сигнала прерывания. Используется также при обмене через DMA/ПДП. |
| 5 | Зарезервировано |
| 4 | CC4G: генерация сигнала «захват/сравнение». См. СС1G. |
| 3 | СС3G. Аналогично. |
| 2 | СС2G. Аналогично. |
| 1 | CC1G. Генерация сигнала «захват/сравнение».  Значение устанавливается программно. Значение этого может быть очищено аппаратно.   * 0: нет действия; * 1: генерирует сигнал «захват/сравнение» на канале 1.   Если канал CC1 настроен как выход, CC1IF установлен в «1», отправляется соответствующий сигнал прерывания или запрос ПДП.  Если канал CC1 настроен как вход, текущее значение счетчика запоминается в регистре TIMx\_CCR1. |
| 0 | UG: генерация сигнала обновления  Значения устанавливаются программно, обнуляются аппаратно.   * 0: нет действия; * 1: переинициализирует счетчик и генерирует обновление регистров. Обратите внимание, что счетчик предделителя также обнуляется, при этом его коэффициент остается нетронутым. Счетчик сбрасывается, если выбран режим выравнивания по центру или бит DIR установлен в «0», иначе если DIR=1, он принимает значение регистра TIMx\_ARR. |

TIMx\_PSC

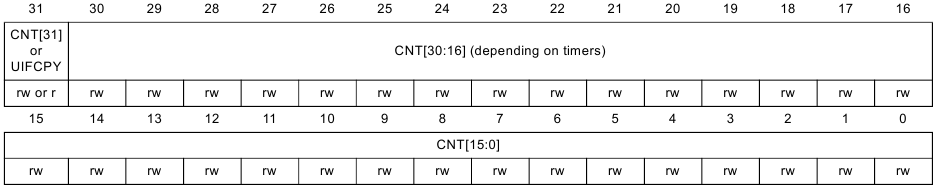


*Рисунок 2.6 – карта битов для регистра TIMx\_PSC*

*Таблица 2.3 – описание битов для регистра TIMx\_PSC*

|  |  |
| --- | --- |
| Бит | Описание |
| 15:0 | PSC: значение предделителя.  Частота счетчика равняется fCK\_PSC / (PSC[15:0] + 1). PSC хранит значение, которое загружается в активный регистр-предделитель при каждом новом сигнале, приводящем к появлению сигнала прерывания («событии» в терминологии STM). Это относится и к тем случаям, когда счетчик сбрасывается через бит UG регистра TIMx\_EGR или посредством триггера, когда тот установлен в режим сброса (reset mode). |

* 1. TIMx\_CNT

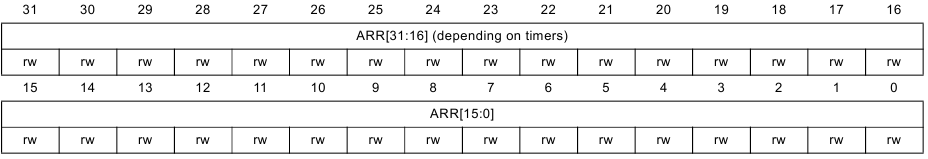


*Рисунок 2.7 – карта битов для регистра TIMx\_CNT*

*Таблица 2.4 – описание битов для регистра TIMx\_CNT*

|  |  |
| --- | --- |
| Бит | Описание |
| 31 | Смысловое значение зависит от значения бита UIFREMAP таймера TIMx\_CR1.  Если UIFREMAP=0:  CNT[31]: старший бит счетчика (при использовании TIM2). Для остальных счетчиков – зарезервированное поле.  Если UIFREMAP=1:  UIFCPY: копия бита UIF из регистра TIMx\_ISR, только для чтения. |
| 30:16 | Старшие биты счетчика (при использовании TIM2). |
| 15:0 | Младшие биты счетчика. |

* 1. TIMx\_ARR

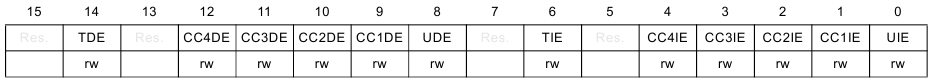


*Рисунок 2.8 – карта битов для регистра TIMx\_ARR*

*Таблица 2.5 – описание битов для регистра TIMx\_ARR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Старшая часть автоперезагружаемого значения. |
| 15:0 | Младшая часть автоперезагружаемого значения. |

* 1. TIMx\_DIER



*Рисунок 2.9 – карта битов для регистра TIMx\_DIER*

*Таблица 2.6 – описание битов для регистра TIMx\_DIER*

|  |  |
| --- | --- |
| Бит | Описание |
| 15 | Зарезервировано |
| 14 | TDE: включение опроса триггера DMA (ПДП) |
| 13 | Зарезервировано |
| 12 | CC4DE: см. CC1DE |
| 11 | CC3DE: аналогично. |
| 10 | CC2DE: аналогично. |
| 9 | CC1DE: включение обращения к DMA через регистр «захват/сравнение 1»   * 0: Обращение к DMA по CC1 отключено; * 1: Обращение к DMA по CC1 включено. |
| 8 | UDE: Включение запросов к DMA на обновление   * 0: запросы отключены; * 1: запросы включены. |
| 7 | Зарезервировано. |
| 6 | TIE: включение прерываний триггера.   * 0: прерывания триггера отключены; * 1: прерывания триггера включены. |
| 5 | Зарезервировано. |
| 4 | CC4IE: включение прерываний от блока захвата/сравнения 4.   * 0: прерывания отключены; * 1: прерывания включены. |
| 3 | CC3IE: Аналогично |
| 2 | CC2IE: Аналогично |
| 1 | CC1IE: Аналогично |
| 0 | UIE: включение прерываний.   * 0: прерывания отключены; * 1: прерывания включены. |

ЛАБОРАТОРНАЯ РАБОТА №3.

ШИМ-СИГНАЛ

1. Цель работы

Освоить дополнительный функционал таймеров, генерацию ШИМ-сигнала

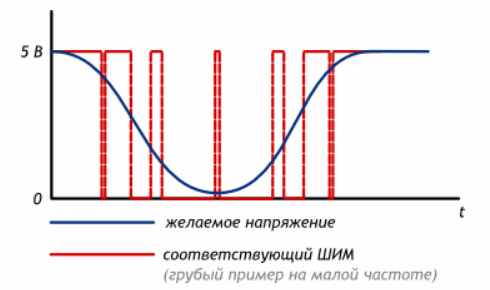
1. Задание

В рамках лабораторной работы № 3 требуется:

1. Настроить схему синхронизации;
2. Настроить таймер на генерацию ШИМ-сигнала;
3. Настроить выводы таймера на LED (GPIOx)
4. Краткое пояснение к заданию
   1. Широтно-импульсная модуляция (ШИМ)

Регулировка выходного напряжения – сложная задача, и многие микроконтроллеры (особенно из недорогих) не имеют аппаратного обеспечения, которое позволяло бы им точно задавать напряжение на выходе. Однако те же самые микроконтроллеры используются для управления системами, напряжение в которых варьируется.

Это возможно потому, что МК используют ШИМ сигналы (в англоязычной литературе – PWM), имитируя некую величину выходного напряжения.



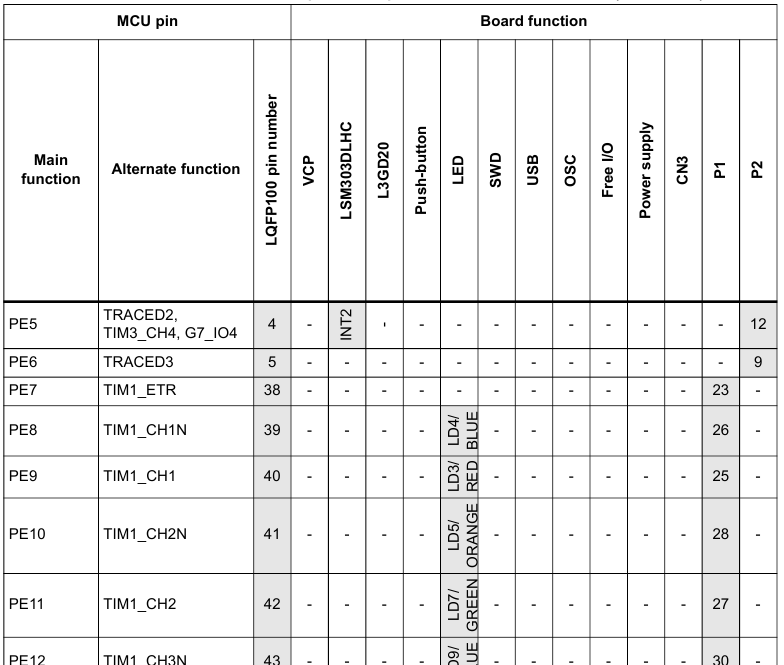
*Рисунок 3.1 – иллюстрация ШИМ*

Так, к примеру, скорость, с который разгоняется громоздкий электродвигатель, не сопоставима со скоростью работы процессора МК, который им управляет, и если нам требуется заставить двигатель работать вполсилы, мы можем «подталкивать» его в определенном темпе, попеременно в равные промежутки времени то подавая, то сбрасывая напряжение на выходе микроконтроллера. Аналогично, чтобы заставить двигатель работать в четверть силы, нам нужно четверть периода работы МК подавать напряжение, а оставшуюся треть – сбрасывать. Продолжительность времени включения к периоду называют скважностью.

Аналогичный подход работает и с менее инертными системами, например, с LED. Разумеется, светодиоды не инертны и моментально откликаются на входное напряжение, и в этом случае вместо постоянного свечения возникает мерцание, однако при такой высокой частоте нам все равно будет казаться, что светодиод светится тусклым светом. Именно это Вам предстоит реализовать в рамках данной ЛР.

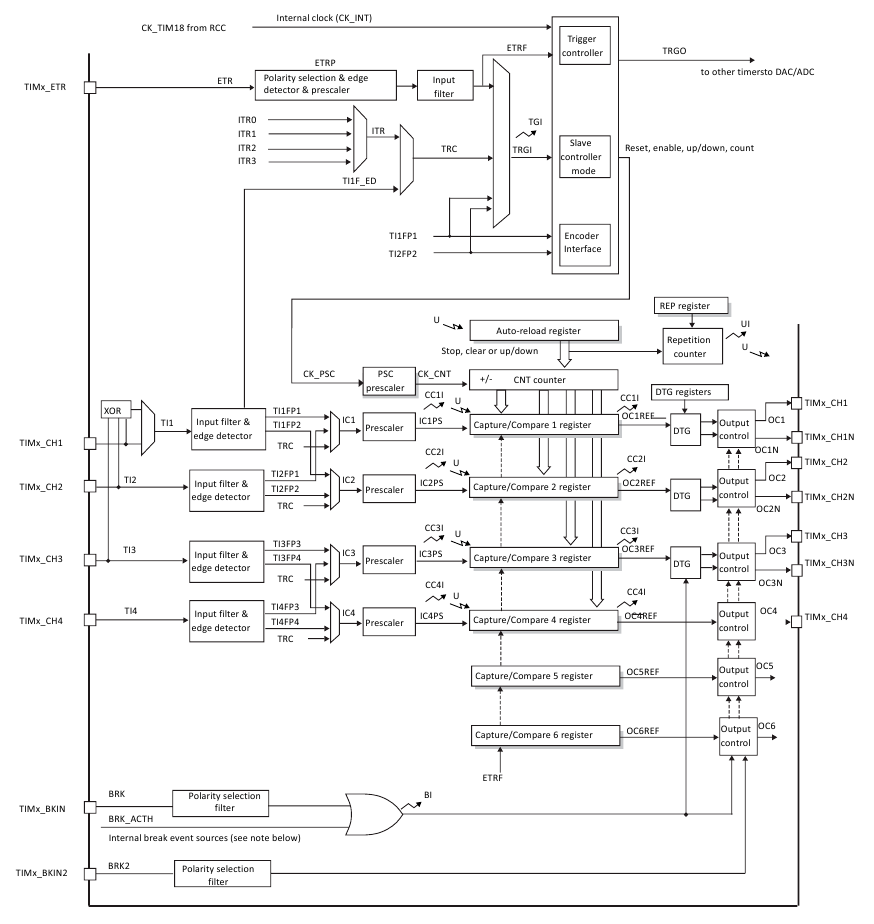
* 1. Альтернативные функции GPIOx.

Как мы знаем из первой ЛР, порты GPIOx могут быть сконфигурированы на вход и на выход. Однако этим портам также может быть назначена альтернативная функция. В документе «STM32F3Disco\_UserManual» есть таблица альтернативных функций.



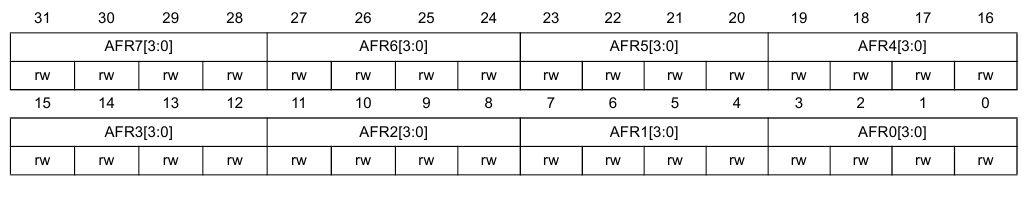
*Рисунок 3.2 – фрагмент таблицы альтернативных функций, стр. 30*

Из таблицы видно, что для светодиодов, прикрепленных к порту «E», альтернативная функция связывается с таймером 1 (плюс, в этой мнемонике есть обозначение соответствующего выходного канала). Таким образом, нам нужно задать альтернативную функцию для нужных нам LED (порты GPIOx) и настроить таймер на использование нужных нам каналов.



*Рисунок 3.3 – диаграмма блока расширенного таймера*

1. GPIOx
   1. GPIOx\_AFRL

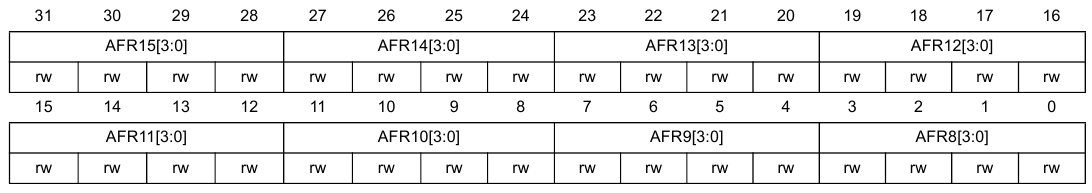


*Рисунок 3.3 – карта битов для регистра GPIOx\_AFRL*

*Таблица 3.1 – описание битов для регистра GPIOx\_AFRL*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:0 | AFRy: выбор альтернативной функции для порта x, ножки y.   * 0000: AF0; * 0001: AF1;   ...   * 1111:AF15 |

* 1. GPIOx\_AFRH

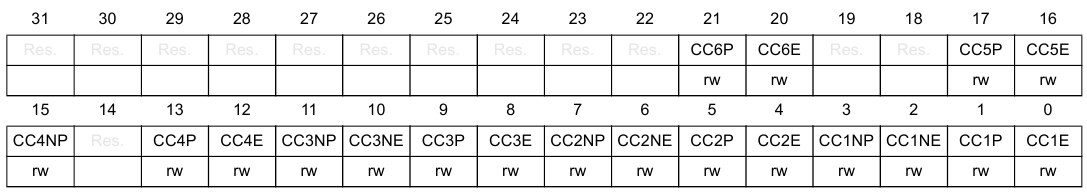


*Рисунок 3.4 – карта битов для регистра GPIOx\_AFRH*

*Таблица 3.2 – описание битов для регистра GPIOx\_AFRL*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:0 | AFRy: выбор альтернативной функции для порта x, ножки y.   * 0000: AF0; * 0001: AF1;   ...   * 1111:AF15 |

1. TIM1
   1. TIM1\_CCER

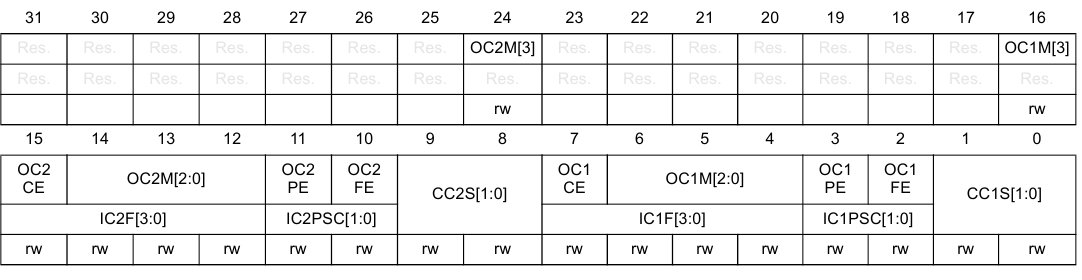


*Рисунок 3.5 – карта битов для регистра TIM1\_CCER*

*Таблица 3.3 – описание битов для регистра TIM1\_CCER*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:22 | Зарезервировано |
| 21 | CC6P: полярность блока захвата/сравнения 6  См. описание CC1P. |
| 20 | СС6E включение выхода блока захвата/сравнения 6.  См. описание CC1E. |
| 19:18 | Зарезервировано |
| 17 | CC5P. Аналогично. |
| 16 | CC5E. Аналогично. |
| 15 | CC4NP: дополнительная полярность блока захвата/сравнения 4.  См. описание CC1NP. |
| 14 | Зарезервировано. |
| 13 | CC4P. Аналогично. |
| 12 | CC4E. Аналогично. |
| 11 | CC3NP. Аналогично. |
| 10 | CC3NE. Включение дополнительного выхода блока захвата сравнения 3. |
| 9 | CC3P. Аналогично. |
| 8 | CC3E. Аналогично. |
| 7 | CC2NP. Аналогично. |
| 6 | CC2NE. Аналогично. |
| 5 | CC2P. Аналогично. |
| 4 | CC2E. Аналогично. |
| 3 | CC1NP. Дополнительная полярность выхода блока захвата/сравнения 1.  Если канал CC1 настроен как выход.   * 0: OC1N активен, высокий; * 1: OC1N активен, низкий.   Если канал CC1 настроен как вход.  Этот бит используется вместе с CC1P для того, чтобы определить полярность TI1FP1 и TI1FP2. См. описание CC1P.  Этот бит перестает быть доступным для записи, когда поле LOCK регистра TIMx\_BDTR имеет значение, отличное от значения по умолчанию, и CC1S=”00” (что означает, что канал настроен на выход).  На каналах с дополнительным выходом этот бит предварительно установлен. Если бит CCPC установлен регистром TIMx\_CR2, тогда активный бит CC1NP берет новое значение из предустановленного бита (только когда генерируется «событие» Commutation, Соединение). |
| 2 | CC1NE: включает дополнительный выход OC1N блока захвата/сравнения 1   * 0: дополнительный выход OC1N не активен. Уровень OC1N определяется содержанием MOE, OSSI, OSSR, OIS1, OIS1N и CC1Eю * 1: дополнительный выход активен, и соответствующий «пин» выхода определяется содержанием MOE, OSSI, OSSR, OIS1, OIS1N и CC1E.   На каналах с дополнительным выходом значение этого бита предварительно установлено. Если бит CCPC установлен регистром TIMx\_CR2, тогда активный бит CC1NE получает новое значение из предустановленного бита (только когда генерируется «событие» Commutation). |
| 1 | CC1P: полярность выхода блока захвата/сравнения 1  Если канал CC1 настроен как выход:   * OC1 активен, высокий; * OC1 активен, низкий.   Если канал CC1 настроен как вход.  Биты CC1NP/CC1P выбирают активную полярность TI1FP1 и TI2FP1 для операций триггеринга или захвата.   * 00: Не инвертированный / по восходящему фронту. Плата реагирует на восходящий фронт TIxFP1 (операции триггеринга или захвата в режимах сброса (reset), внешнего тактирования или режиме триггеринга), TIxFP1 инвертирован (операции триггеринга в режимах энкодера или стробированном режиме); * 01: инвертированный / по падающему фронту. Плата реагирует на падающий фронт TIxFP1, TIxFP1 инвертирован; * 10: зарезервировано, не используйте эту конфигурацию; * 11: не инвертирован / по обоим фронтам. |
| 0 | CC1E: включение выхода блока захвата/сравнения 1.  Если канал CC1 сконфигурирован как выход.   * 0: выключен. Канал OC1 не активен; * 1: включен. Выход OC1 на соответствующий «пин» определяется битами MOE, OSSI, OSSR, OIS1, OIS1N и CC1NE.   Если канал CC1 сконфигурирован как вход.  Этот бит опередляет, может ли захват счетчика быть выполнен в блоке сравнения захвата 1 (TIMx\_CCR1) или нет.   * 0: захват отключен; * 1: захват включен.   На каналах с дополнительным выходом этот бит предустановлен. Если бит CCPC регистра TIMx\_CR2 установлен, активный бит CC1NE берет новое значение из предустановленного бита (но только когда генерируется «событие» Commutation). |

* 1. TIM1\_CCMR1



*Рисунок 3.6 – карта битов для регистра TIM1\_CCMR1*

Каналы могут использоваться как выходы (режим захвата) и как входы (режим сравнения). Направление канала определяется конфигурацией соответствующих битов (CCxS). Другие биты имеют различное значение в зависимости от того, на вход или на выход работает канал. Для определенного бита OCxx описывает его функционал, когда канал работает на выход, а ICxx – функционал, когда канал работает на вход. Поэтому Вам придется учитывать, что тот же самый бит может иметь различное значение для режима входа и режима выхода.

*Таблица 3.4 – описание битов для регистра TIM1\_CCMR1,*

*ДЛЯ ВЫХОДА (СРАВНЕНИЯ)*

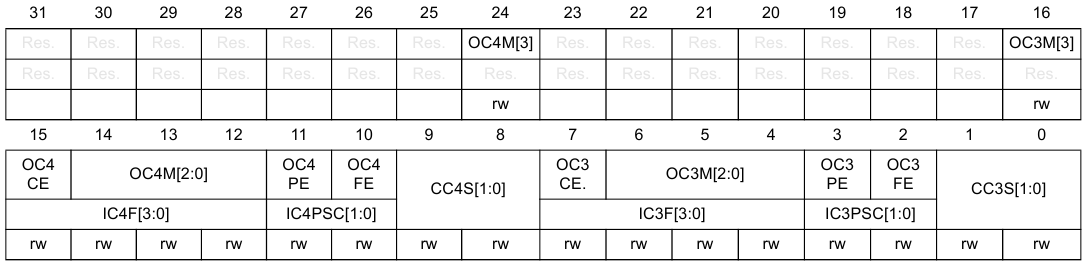
|  |  |
| --- | --- |
| Бит | Описание |
| 31:25 | Зарезервировано |
| 24 | OC2M[3]: Режим сравнения 2 – бит 3 (см. описание 14:12). |
| 23:17 | Зарезервировано. |
| 16 | OC1M[3]: Режим сравнения 1 – бит 3 (см. описание 6:4) |
| 15 | OC2CE: Включить очистку сравнения 2. |
| 14:12 | OCM2M[2:0]: Режим сравнения 2. |
| 11 | OC2PE: Включение предустановки сравнения 2. |
| 10 | OC2FE: Включение быстрого сравнения 2. |
| 9:8 | CC2S[1:0] Выбор захвата/сравнения 2  Это битовое поле определяет направление канала (выход/вход) и используемый вход.   * 00: канал CC2 сконфигурирован как выход; * 01: канал CC2 сконфигурирован как вход, IC2 отображается на TI2; * 10: канал CC2 сконфигурирован как вход, IC2 отображается на TI1; * 11: канал CC2 сконфигурирован как вход, IC2 отображается на TRC. Этот режим работает, только битом TS (регистр TIMx\_SMCR) выбран внешний триггер входа.   Биты CC2S доступны для записи только тогда, когда канал отключен (в регистре TIMx\_CCRE бит CC2E=0). |
| 7 | OC1CE: Включить очистку сравнения 1.   * OC1REF не затрагивается сигналом ocref\_clr\_int; * OC1REF очищается, как только обнаруживается высокий уровень сигнала ocref\_clr\_signal (вход OCREF\_CLR или ETRF). |
| 6:4 | OC1M: Режим сравнения 1.  Эти биты определяют поведение выходного сигнала с OC1 или с OC1N. OC1REF активен, в то время как активность OC1 и OC1N зависит от битов CC1P и CC1NP.   * 0000: Замороженное состояние. В результате сравнения регистров TIMx\_CCR1 и TIMx\_CNT (счетчик) не производится никакого выхода; * 0001: При совпадении установить канал 1 на активный уровень. Когда значение регистра TIMx\_CNT достигает значения TIMx\_CCR11, выходной сигнал OC1REF усиливается; * 0010: При совпадении установить канал 1 на неактивный уровень. Когда значение регистра TIMx\_CNT достигает значения TIMx\_CCR11, выходной сигнал OC1REF ослабляется; * 0011: переключение. OC1REF переключается, когда TIMx\_CNT=TIMxCCR1; * 0100: ослабить сигнал на OC1REF; * 0101: усилить сигнал на OC1REF; * 0110: режим ШИМ-модуляции 1. Канал активен, пока TIMx\_CNT<TIMx\_CCR1; это справедливо как для уменьшающегося, так и для увеличивающегося счетчика; * 0111: режим ШИМ-модуляции 2. Если TIMx\_CNT<TIMx\_CCR1, канал неактивен. Это справедливо как для уменьшающегося, так и для увеличивающегося счетчика; * 1000: Триггерируемый режим одновибратора 1. При увеличении счетчика канал будет оставаиться активным, пока не поступит сигнал от триггера (TRGI). После этого выполнится сравнение, как в режиме ШИМ модуляции 1, а при следующем «обновлении» каналы снова станут активными.   При уменьшении счетчика канал будет оставаться неактивным до тех пор, пока не поступит сигнал от триггера. После этого произойдет сравнение, как в режиме ШИМ-модуляции 1, а при следующем «обновлении» каналы снова станут неактивными;   * 1001: Триггерируемый режим одновибратора 2. При увеличении счетчика канал будет оставаться неактивным, пока не поступит сигнал от триггера (TRGI). Затем выполнится сравнение, как в решиме ШИМ-модуляции 2, а при следующем «обновлении» каналы снова станут неактивными.   При уменьшении счетчика канал будет оставаться активным, пока не поступит сигнал от триггера (TRGI). Затем выполнится сравнение, как в режиме ШИМ-модуляции 1, и при следующем «обновлении» каналы снова станут активными.   * 1010/1011: Зарезервировано; * 1100: комбинированный режим ШИМ-модуляции 1. OC1REF ведет себя так же, как в режиме ШИМ-модуляции 1. OC1REFC – это логическое «ИЛИ» для OC1REF и OC2REF; * 1101: комбинированный режим ШИМ-модуляции 2. OC1REF ведет себя так же, как в режиме ШИМ-модуляции 2. OC1REFC – это логическое «И» для OC1REF и OC2REF; * 1110: Асимметричный режим ШИМ-модуляции 1. OC1REF ведет себя так же, как в режиме ШИМ-модуляции 1. OC1REFC выводит OC1REF, когда счетчик увеличивается, и OC2REF – когда уменьшается; * 1111: Асимметричный режим ШИМ-модуляции 2. OC1REF ведет себя так же, как в режиме ШИМ-модуляции 2. OC1REFC выводит OC1REF, когда счетчик увеличивается, и OC2REF – когда уменьшается.   Эти биты не могут быть изменены, пока биты LOCK в регистре TIMx\_BDTR отличаются от своего значения по умолчанию и CC1S=00.  В режиме ШИМ модуляции уровни OCREF меняются тлько тогда, когда меняется результат сравнения или когда режим выхода сравнения переключается из замороженного режима в режим ШИМ-модуляции.  На каналах с дополнительными выходами это битовое поле предустановленно. Если бит CCPC регистра TIMx\_CR2 установлен, то активные биты OC1M берут свое новое значение из предустановленных битов (только когда происходит событие «COM»). |
| 3 | OC1PE: Включение предустановки сравнения 1.   * 0: предварительная загрузка TIMx\_CCR1 отключена. TIMx\_CCR1 может быть изменен в любое время, новое значение вступает в силу немедленно; * 1: предварительная загрузка TIMx\_CCR1 включена. Операции чтения/записи применяются к загружаемому значению. Значение TIMx\_CCR1 загружается в активный регистр при каждом «обновлении события».   Эти биты не могут быть перезаписаны, пока значение битов LOCK регистра TIMx\_BDTR отличается от первоначального и CC1S=00 (канал настроен на выход).  Режим ШИМ-модуляции может быть использовать без валидации загружаемого регистра только в режиме одновибратора (когда в регистре TIMx\_CR1 бит OPM=1). Иначе поведение непредсказуемо. |
| 2 | OC1FE: Быстрое сравнение 1 включено.  Этот бит используется для ускорения сравнения.   * 0: CC1 ведет себя как обычно, зависит от значения счетчика и CCR1 (даже когда триггер включен). Минимальная задержка подачи сигнала на CC1 составляет 5 тактов; * 1: Активный фронт на входе триггера передается как сигнал совпадения при сравнении на выход CC1. Далее OC устанавливается на уровень сравнения вне зависимости от результатов сравнения. Задержка между запоминанием входа триггера и активацией выхода CC1 сократилась до 3 тактов. OCFE работает только тогда, когда канал функционирует в режиме ШИМ-модуляции 1 или 2. |
| 1:0 | CC1S: выбор захвата/сравнения 1.  Это битовое поле определяет направление канала (вход/выход), а также используемый выход.   * 00: Канал CC1 сконфигурирован как выход; * 01: Канал CC1 сконфигурирован как вход, IC1 отображается на TI1; * 10: Канал CC1 сконфигурирован как вход, IC1 отображается на TI2; * 11: Канал CC1 сконфигурирован как вход, IC1 отображается на TRC. Этот режим работает, только если вход внутреннего триггера установлен посредством бита TS регистра TIMx\_SMCR).   Биты CC1S доступны для записи, только когда канал отключен (в регистре TIMx\_CCER CC1E=0). |

*Таблица 3.5 – описание битов для регистра TIM1\_CCMR1,*

*ДЛЯ ВХОДА (ЗАХВАТА)*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Зарезервировано |
| 15:12 | IC2F: фильтр захвата 2. |
| 11:10 | IC2PSR[1:0]: предделитель захвата 2. |
| 9:8 | CC2S Выбор захвата/сравнения 2.  Это битовое поле определяет направление канала (вход/выход), а также используемый вход.   * 00: Канал CC2 настроен как выход; * 01: Канал CC2 настроен как вход, IC2 отображается на TI2; * 10: Канал CC2 настроен как вход, IC2 отображается на TI1; * 11: Канал CC2 настроен как вход, IC2 отображается на TRC. Этот режим работает только тогда, когда вход внутреннего триггера установлен посредством бита TS регистра TIMx\_SMCR.   Биты CC2S доступны для записи только тогда, когда канал отключен (в регистре TIMx\_CCER бит CC2E=0). |
| 7:4 | IC1F[3:0] Фильтр захвата 1.  Это битовое поле определяет частоту, используемую для запоминания (сэмплирования) входа TI1 и длины цифрового фильтра, применнного к TI1. Цифровой фильтр состоит из счетчика «событий», в котором N последовательных событий ожидают своей очереди на выход.   * 0000: Нет фильтра, запоминание осуществляется на частоте fDTS; * 0001: fSAMPLING=fCK\_INT, N=2; * 0010: fSAMPLING=fCK\_INT, N=4; * 0011: fSAMPLING=fCK\_INT, N=8; * 0100: fSAMPLING=fDTS/2, N=6; * 0101: fSAMPLING=fDTS/2, N=8; * 0110: fSAMPLING=fDTS/4, N=6; * 0111: fSAMPLING=fDTS/4, N=8; * 1000: fSAMPLING=fDTS/8, N=6; * 1001: fSAMPLING=fDTS/8, N=8; * 1010: fSAMPLING=fDTS/16, N=5; * 1011: fSAMPLING=fDTS/16, N=6; * 1100: fSAMPLING=fDTS/16, N=8; * 1101: fSAMPLING=fDTS/32, N=5; * 1110: fSAMPLING=fDTS/32, N=6; * 1111: fSAMPLING=fDTS/32, N=8. |
| 3:2 | IC1PSC: предделитель захвата 1.  Это битовое поле определяет отношение предделителя, применяемого ко входу CC1. Предделитель сбрасывается вместе с установкой CC1E=0 (регистра TIMx\_CCER).   * 00: предделение не используется, захват выполняется каждый раз, когда на входе блока захвата генерируется сигнал; * 01: захват выполняется каждое 2-е появление сигнала; * 10: захват выполняется каждое 4-е появление сигнала; * 11: захват выполняется каждое 8-е появление сигнала. |
| 1:0 | CC1S: выбор захвата/сравнения 1.  Это битовое поле определяет направление канала (вход/выход), а также используемый выход.   * 00: Канал CC1 сконфигурирован как выход; * 01: Канал CC1 сконфигурирован как вход, IC1 отображается на TI1; * 10: Канал СС1 сконфигурирован как вход, IC1 отображается на TI2; * 11: Канал CC1 сконфигурирован на вход, IC1 отображается на TRC. Этот режим работает, только если вход внутреннего триггера установлен битом TS (регистр TIMx\_SMCR).   Биты CC1S доступны для записи, только когда канал отключен (В регистре TIMx\_CCER CC1E=0). |

* 1. TIM1\_CCMR2



*Рисунок 3.7 – карта битов для регистра TIM1\_CCMR2*

Примечание, которое дано для регистра CCMR1 (см. выше), справедливо и для этого регистра).

*Таблица 3.6 – описание битов регистра TIM1\_CCMR2*

*ДЛЯ ВХОДА (СРАВНЕНИЯ)*

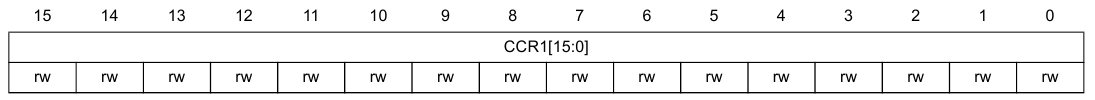
|  |  |
| --- | --- |
| Бит | Описание |
| 31:25 | Зарезервировано |
| 24 | OC4M[3]: режим сравнения 4, бит 3. |
| 23:17 | Зарезервировано. |
| 16 | OC3M[3]: режим сравнения 3, бит 3. |
| 15 | OC4CE: включить очистку сравнения 4. |
| 14:12 | OC4M: Режим сравнения 4. |
| 11 | OC4PE: Включение предзагрузки сравнения. |
| 10 | OC4FE: Включение быстрого сравнения 4 |
| 9:8 | CC4S: выбор захвата/сравнения 4  Это битовое поле определяет направление канала (вход/выход), а также используемый вход.   * 00: Канал CC4 используется как выход; * 01: Канал CC4 используется как вход, IC4 отображается на TI4; * 10: Канал CC4 используется как вход, IC4 отображается на TI3; * 11: Канал CC4 используется как вход, IC4 отображается на TRC. Этот режим работает, только когда вход внутреннего триггера установлен посредством бита TS (реигстра TIMx\_SMCR).   Биты CC4S доступны для записи, только когда канал выключен (в регистре TIMx\_CCER CC4E=0). |
| 7 | OC3CE: Включение сравнения 3. |
| 6:4 | OC3M: Режим сравнения 3. |
| 3 | OC3PE: Включение предзагрузки сравнения 3. |
| 2 | OC3FE: Включение быстрого сравнения 3. |
| 1:0 | Выбор захвата/сравнения 3.  Это битовое поле определяет направление канала (вход/выход), а также используемый выход.   * 00: Канал CC3 используется как выход; * 01: Канал CC3 используется как вход, IC3 отображается на TI3; * 10: Канал CC3используется как вход, IC3 отображается на TI4; * 11: Канал CC3 используется как вход, IC3 отображается на TRC. Этот режим работает, только когда вход внутреннего триггера установлен посредством бита TS (реигстра TIMx\_SMCR).   Биты CC4S доступны для записи, только когда канал выключен (в регистре TIMx\_CCER CC3E=0). |

*Таблица 3.7 – описание битов регистра TIM1\_CCMR2*

*ДЛЯ ВЫХОДА (ЗАХВАТА)*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Зарезервировано |
| 15:12 | IC4F: фильтр захвата 4. |
| 11:10 | IC4PSC: предделитель входа захвата 4. |
| 9:8 | CC4S: выбор захвата/сравнения 4.  Это битовое поле определяет направление канала (вход/выход), а также используемый вход.   * 00: Канал CC4 используется как выход; * 01: Канал CC4 используется как вход, IC4 отображается на TI4; * 10: Канал CC4 используется как вход, IC4 отображается на TI3; * 11: Канал CC4 используется как вход, IC4 отображается на TRC. Этот режим работает, только когда вход внутреннего триггера установлен посредством бита TS (реигстра TIMx\_SMCR).   Биты CC4S доступны для записи, только когда канал выключен (в регистре TIMx\_CCER CC4E=0). |
| 7:4 | IC3F: фильтр захвата 3. |
| 3:2 | IC3PSC: предделитель входа захвата 3. |
| 1:0 | CC3S: выбор захвата/сравнения 3.  Это битовое поле определяет направление канала (вход/выход), а также используемый выход.   * 00: Канал CC3 используется как выход; * 01: Канал CC3 используется как вход, IC3 отображается на TI3; * 10: Канал CC3используется как вход, IC3 отображается на TI4; * 11: Канал CC3 используется как вход, IC3 отображается на TRC. Этот режим работает, только когда вход внутреннего триггера установлен посредством бита TS (реигстра TIMx\_SMCR).   Биты CC4S доступны для записи, только когда канал выключен (в регистре TIMx\_CCER CC3E=0). |

* 1. TIM1\_CCR1



*Рисунок 3.8 – карта битов для регистра TIM1\_CCR1*

*Таблица 3.8 – описание битов для регистра TIM1\_CCR1*

|  |  |
| --- | --- |
| Бит | Описание |
| 15:0 | Значение захвата/сравнения 1  Если канал CC1 сконфигурирован как выход.  CCR1 – это значение, которое загружается в регистр захвата/сравнения 1 (предзагружаемое значение). Предзагрузка может быть отключена (регистр TIMx\_CCMR1, бит OC1PE), и тогда данное значение «рассматривается» микроконтроллером как актуальное, иначе оно копируется в активный регистр захвата/сравнения 1, где впоследствии генерируется сигнал «события».  Если канал CC1 сконфигурирован как вход.  CR1 – это счетчик, изменяемый последним «событием» захвата. |

ЛАБОРАТОРНАЯ РАБОТА №4.

АЦП

1. Цель работы

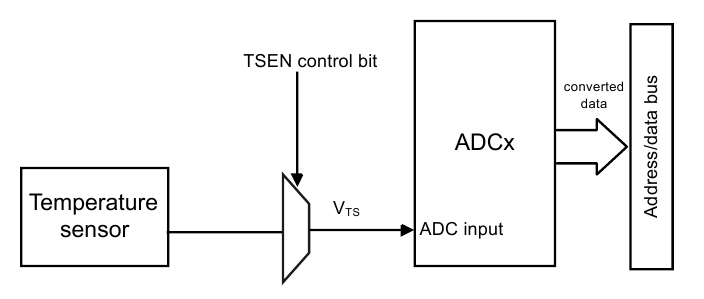
Освоить работу с АЦП.

1. Задание

В рамках лабораторной работы № 4 требуется:

1. Настроить схему синхронизации;
2. Настроить АЦП на работу со встроенным температурным датчиком;
3. Запрограммировать зависимость свечения/мигания LED от значения температуры.
4. Краткое пояснение к заданию

В плате, предоставленной Вам, имеется внутренний температурный датчик, запитанный на процессор, и аналогово-цифровой преобразователь (АЦП/ADC), который может считывать его значение.

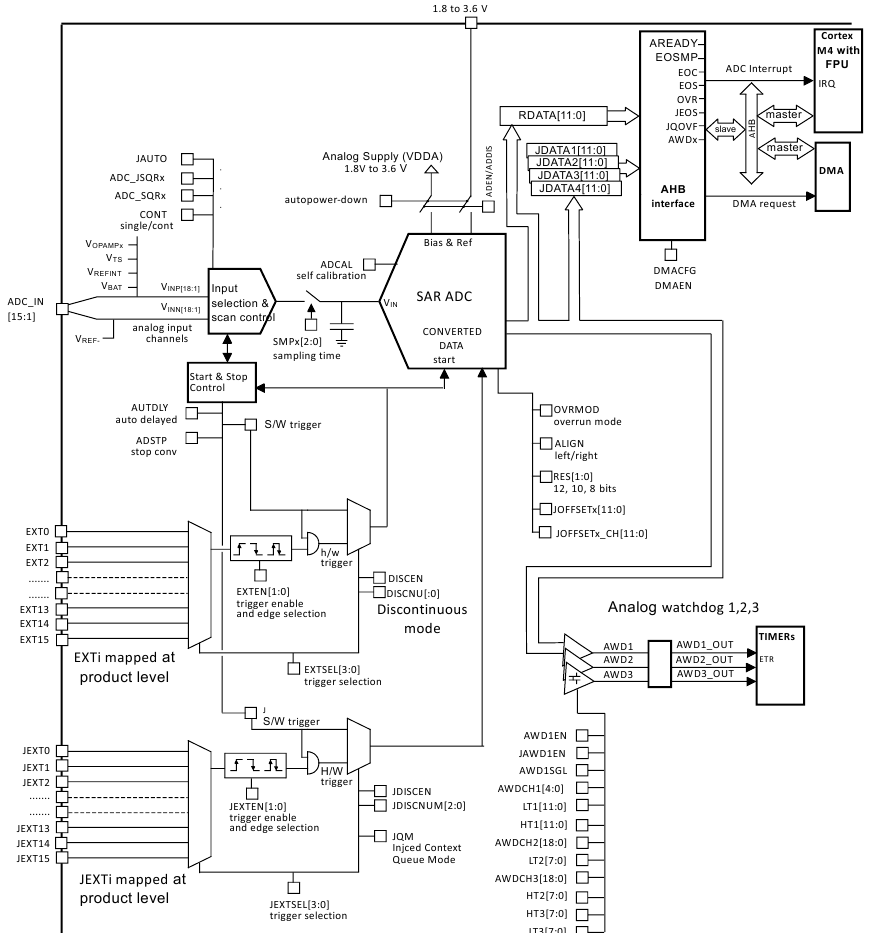


*Рисунок 4.1 – температурный датчик и АЦП*

Температура процессора зависит от частоты его работы. Следовательно, Вам нужно разогнать процессор, чтобы увеличить частоту его работы.

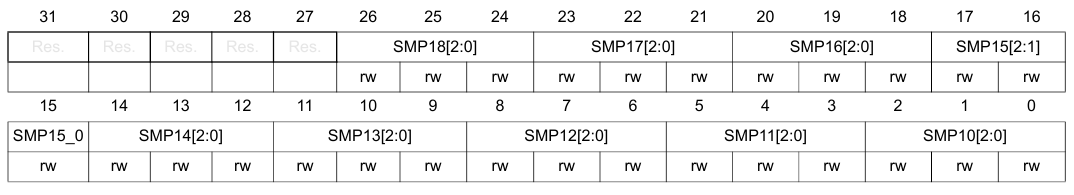
Вам предстоит сделать так, чтобы вывод LED соответствовал температуре процессора. Это может быть мигание (малая температура – редкие мигания, высокая – частые) или включение нескольких светодиодов (одного, двух трех – чем выше температура, тем больше включенных LED).

В плате предусмотрено 4 АЦП. С температурным датчиком может работать только АЦП1 (ADC1).



*Рисунок 4.2 – схема АЦП*

1. ADC1
   1. ADC1\_SMPR2

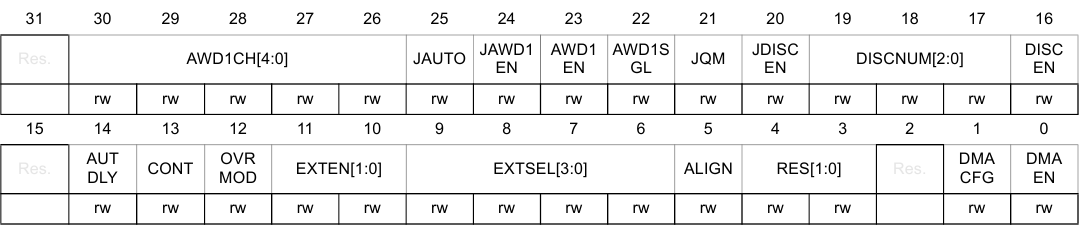


*Рисунок 4.2 – карта битов для регистра ADC1\_SMPR2*

*Таблица 4.1 – описание битов для регистра ADC1\_SMPR2*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:27 | Зарезервировано |
| 26:0 | SMPx[2:0]: выбор времени считывания («семплирования») для канала x.  Используется для выбора времени семплирования для каждого из каналов. Во время семплирования эти биты не должны изменяться.  Значения устанавливаются программно.   * 000: 1.5 времени такта АЦП; * 001: 2.5; * 010: 4.5; * 011: 7.5; * 100: 19.5; * 101: 61.5; * 110: 181.5; * 111 601.5.   Запись в эти биты возможна лишь тогда, когда биты ADSTART и JADSTART установлены в 0 (это значит, что никакого преобразования не происходит).  *Примечание: указываются такты АЦП, а не процессора, потому как тактирование АЦП может быть независимо от тактирования остальной системы.* |

* 1. ADC1\_CFGR

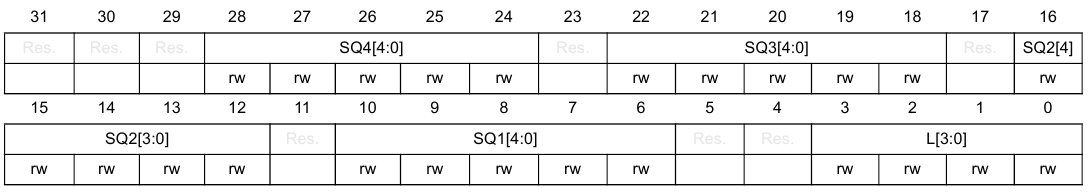


*Рисунок 4.3 – карта битов для регистра ADC1\_CFGR*

*Таблица 4.2 – описание битов для регистра ADC1\_CFGR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31 | Зарезервировано |
| 30:26 | AWD1CH: Выбор канала для сторожевого таймера 1.  Выбор входного канала для контроля сторожевым таймером  Значения устанавливаются программно.   * 00000: зарезервировано (канала 0 не существует); * 00001: канал 1 контролируется сторожевым таймером 1; * 00010: канал 2; * 00011: канал 3;   …   * 10010: канал 18.   Канал, выбранный здесь, также должен быть выбран в регистрах SQR или JSQR.  Запись в эти биты может производиться только тогда, когда ADSTART=0 и JADSTART=0. |
| 25 | JAUTO: автоматическое принудительное преобразование. Значения устанавливаются программно.   * Преобразование отключено; * Преобразование включено.   Примечание: АЦП на STM32 может работать в обычном и принудительном режимах («regular» и «injected group conversion»). В обычном режиме АЦП циклически выполняет преобразования входных каналов «в цифру». В принудительном режиме преобразование генерируется программно или по какому-либо внутреннему сигналу, «событию» (в терминологии STM32). |
| 24 | JAWDEN: включение сторожевого таймера 1 на каналах с принудительным преобразованием. Значения устанавливаются программно.   * 0: таймер включен; * 1: таймер отключен.   Можно записывать в этот бит, только если JADSTART=0. |
| 23 | AWD1EN: включение сторожевого таймера 1 на каналах с обычным преобразованием. Значения устанавливаются программно.   * 0: таймер вкл.; * 1: таймер выкл.   Записывать в этот бит можно тогда, когда ADSTART=0; |
| 22 | AWD1SGL: включение сторожевого таймера 2 на одном или нескольких каналах.   * 0: таймер включен на всех каналах; * 1: таймер включен на одном канале. |
| 21 | JQM: управление последовательностью принудительного преобразования (регистр JSQR).  Определяет, что как обрабатывать пустую последовательность. Значения устанавливаются программно.   * 0: JSQR, режим 0. Очередь никогда не остается пустой, сохраняется последняя записанная в JSQR конфигурация; * 1: режим1. Очередь может быть пустой, и когда это происходит, аппаратные и программные триггеры принудительного преобразования автоматически отключаются сразу после обработки последней валидной последовательности принудительного преобразования.   Писать в этот бит можно, только когда JADSTART=0.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 20 | JDISCEN: прерывающийся режим для каналах с принудительным преобразованием.   * 0: отключен; * 1: включен.   Бит доступен для записи, только когда JADSTART=0.  Нельзя использовать одновременно автоматический принудительный и прерывающийся режимы. Когда установлен бит JAUTO, биты DISCEN и JDISCEN должны быть сброшены.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 19:17 | DISCNUM: определение каналов для прерывающегося режима.  Используются для того, чтобы определить количество каналов с обычным преобразованием, для которого необходимо включить прерывающийся режим.   * 000: 1 канал; * 001: 2 канала;   …   * 111: 9 каналов.   Бит доступен для записи, только когда ADSTART=0.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 16 | DISCEN: прерывающийся режим для каналов с обычным преобразованием. В   * 0: отключен; * 1: включен.   Невозможно включить одновременно непрерывный и прерывающийся режим, т.е. когда биты DISCEN и CONT одновременно установлены в «1».  Нельзя использовать одновременно автоматический принудительный и прерывающийся режимы. Когда установлен бит JAUTO, биты DISCEN и JDISCEN должны быть сброшены.  Бит доступен для записи, только когда ADSTART=0.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 15 | Зарезервировано. |
| 14 | AUTDLY: Режим отложенного преобразования.   * 0: выключен; * 1: включен   Бит доступен для записи, только когда JADSTART=0 и ADSTART=0.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 13 | CONT: режимы единичного/непрерывного преобразования. При установке этого бита в 1 преобразование выполняется непрерывно, до тех пор, пока CONT не будет отключен.   * 0: Единичное преобрвазование; * 1: непрерывное преобразование.   Невозможно включить одновременно непрерывный и прерывающийся режим, т.е. когда биты DISCEN и CONT одновременно установлены в «1».  Бит доступен для записи, только когда ADSTART=0.  Когда установлен сдвоенный режим (ADC1\_CCR, бит DUAL != 0), бит JQM подчиненного АЦП блокируется для записи, и его значение приравнивается к значению ведущего АЦП. |
| 12 | OVRMOD: режим перезаписи.  Этот бит устанавливает способ обработки ситуаций, когда в регистре данных (DR) находятся старые данные, но уже получены новые результаты с АЦП.   * 0: остаются старые данные; * 1: записываются новые данные. |
| 11:10 | EXTEN: включение/выключение аппаратного триггера и выбор полярности для каналов с обычным преобразованием.   * 00: Аппаратный триггер выключен, преобразование запускается программно; * 01: аппаратный триггер включен, реагирует на восходящий фронт; * 10: аппаратный триггер включен, реагирует на падающий фронт; * 11: аппаратный триггер включен, реагирует как на восходящий, так и на падающий фронт.   Бит доступен для записи, только когда ADSTART=0. |
| 9:6 | EXTSEL: назначение «события» (в терминологии STM32), сигнала, триггеру группы обычного преобразования.   * 0000: событие 0; * 0001: событие 1;   …   * 1111 : событие 15.   Бит доступен для записи, только когда ADSTART=0. |
| 5 | ALIGN: выравнивание данных (результатов преобразования) в регистре данных ADC1\_DR.   * 0: правое выравнивание; * 1: левое выравнивание. |
| 4:3 | RES: точность преобразования.   * 00: 12 бит; * 01: 10 бит; * 10: 8 бит; * 11: 6 бит.   Бит доступен для записи, только когда ADSTART=0 и JADSTART=0. |
| 2 | Зарезервировано. |
| 1 | DMACFG. Режим прямого доступа к памяти (DMA/ПДП).   * 0: единичный режим; * 1: круговой режим.   Бит доступен для записи, только когда ADSTART=0 и JADSTART=0.  В сдвоенном режиме этот бит игнорируется, а настройка должна осуществляеться через бит DMACFG в регистре ADC\_CCR. |
| 0 | DMAEN: включение прямого доступа к памяти.  Включает или отключает генерацию ПДП-запросов. Позволяет использовать GP-DMA для автоматической записи преобразованных данных.   * 0: отключен; * 1: включен. |

* 1. ADC1\_SQR

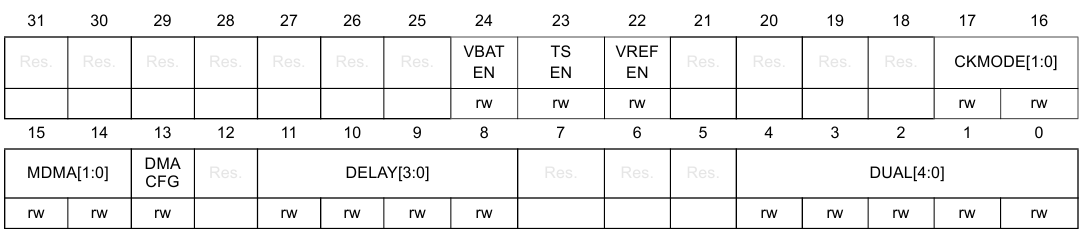


*Рисунок 4.4 – карта битов для регистра ADC1\_SQR*

*Таблица 4.3 – описание битов для регистра ADC1\_SQR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:29 | Зарезервировано |
| 28:24 | SQ4: Номера этих битов находятся в соответствии с номером канала. В последовательности обычного преобразования – это 4-й номер.  Бит доступен для записи, только когда ADSTART=0.  Примечание: каналы нумеруются с 1. Запись «00000» не имеет смысла. |
| 23 | Зарезервировано |
| 22:18 | SQ3: номер 3, полностью аналогично SQ4. |
| 17 | Зарезервировано. |
| 16:12 | SQ2: номер 3, полностью аналогично SQ4. |
| 11 | Зарезервировано. |
| 10:6 | SQ1: номер 2, полностью аналогично SQ4. |
| 5:4 | Зарезервировано. |
| 3:0 | Длина последовательности обычного преобразования:   * 0000: 1 преобразование; * 0001: 2 преобразования;   …   * 1111: 17 преобразований. |

* 1. ADC1\_CСR



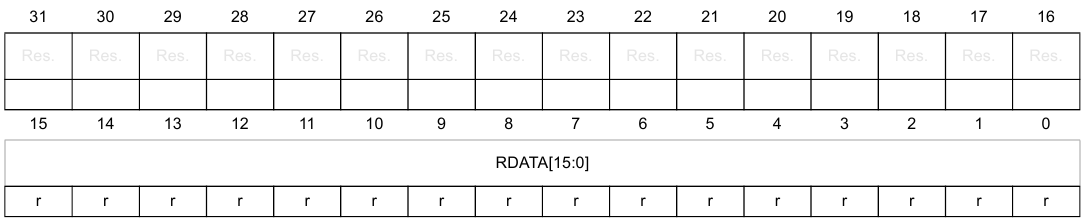
*Рисунок 4.5 – карта битов для регистра ADC1\_CCR*

*Таблица 4.4 – описание битов для регистра ADC1\_CCR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:25 | Зарезервировано |
| 24 | Включение канала VBAT   * 0: отключен; * 1: включен.   Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 23 | TSEN: включение температурного датчика.   * 0: отключен; * 1: включен.   Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 22 | Включение канала VREFINT.   * 0: отключен; * 1: включен.   Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 21:18 | Зарезервировано. |
| 17:16 | CKMODE: режим тактирования ADC.   * 00: CK\_ADCx(x=123). Асинхронный; * 01: HCLK/1 (синхронный). Эта конфигурация доступна только тогда, когда предделитель AHB установлен в значение «1» (регистр RCC\_CFGR, битовое поле HPRE), а системное тактирование занимает 50% рабочего цикла. * 10: HCLK/2 (синхронный); * 11: HCLK/4 (синхронный).   Для всех синхронных режимов отсутствует задержка между срабатыванием триггера и началом преобразования постоянна, погрешность отустствует.  Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 15:14 | MDMA: прямой доступ к памяти (ПДП, DMA) при сдвоенном режиме работе АЦП.   * 00: отключен; * 01: зарезервировано; * 10: включен для 12 и 10-битных преобразований; * 11: включен для 8 и 6-битных преобразований.   Бит доступен для записи, только когда ADSTART=0. |
| 13 | DMACFG: Режим прямого доступа к памяти (DMA/ПДП).   * 0: единичный режим; * 1: круговой режим.   Бит доступен для записи, только когда ADSTART=0. |
| 12 | Зарезервировано. |

|  |  |
| --- | --- |
| 11:8 | DELAY: задержка между двумя преобразованиями.  *Таблица 5.4.1 – таблица значений для бита ADC1\_CCR: DELAY*    Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 7:5 | Зарезервировано |
| 4:0 | DUAL: Выбор конфигурации сдвоенной работы   * 00000: независимый режим; * 00001: комбинированно. Обычный одновременный + принудительный одновременный; * 00010: комбинированно. Обычный одновременный + альтернативный триггерный; * 00100: зарезервировано. * 00101: только принудительный одновременный; * 00110: только обычный одновременный; * 00111: только чередующийся режим; * 01001: только альтернативный триггерный.   Все остальные комбинации зарезервированы.  Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |

* 1. ADC1\_DR

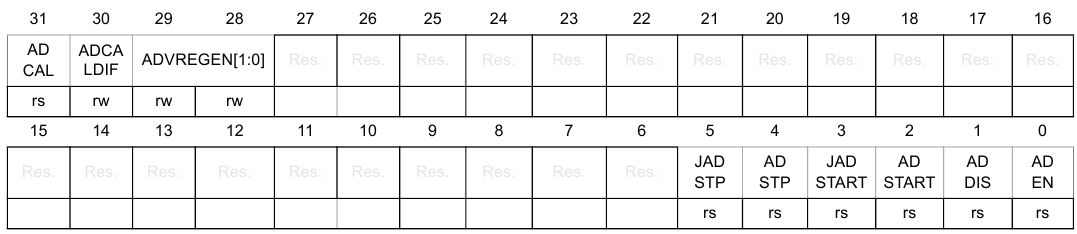


*Рисунок 4.6 – карта битов для ADC1\_DR*

*Таблица 4.5 – описание битов для регистра ADC1\_DR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31:16 | Зарезервировано |
| 15:0 | RDATA: Хранит результат преобразования. |

* 1. ADC1\_CR



*Рисунок 4.7 – карта битов регистра ADC1\_CR*

*Таблица 4.6 – описание битов для регистра ADC1\_CR*

|  |  |
| --- | --- |
| Бит | Описание |
| 31 | ADCAL: калибровка АЦП.  Перед установкой этого бита следует установить ADCALDIF, чтобы указать, для какого количества выводов производится калибровка.  Значения устанавливаются программно. Сбрасываются аппаратно.   * 0: калибровка окончена; * 1: начать калибровку.   Этот бит может быть изменен программно, только если ADEN=0.  Программно записывать битовое поле ADCx\_CALFACT только тогда, когда ADEN = 1 и ADSTART=0 (ADC включен, преобразование не осуществляется) |
| 30 | ADCALDIF: обозначает единственность или множественность калибрующихся выходов.   * 0: 1 выход, * 1: множество выход.   Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 29:28 | ADVREGEN: включение/выключение регулятора напряжения АЦП  Этот бит должен быть установлен перед началом калибровки. После установки нужно подождать, пока операция окончится, и только после этого настраивать калибровку.   * 00: Требуется промежуточное состояние перед включением или выключением АЦП; * 01: регулятор напряжения АЦП включен; * 10: регулятор напряжения АЦП выключен; * 11: зарезервировано.   Запись в данный бит возможна, если деактивированы следующие регистры: ADCAL=0, JADSTART=0, ADSTART=0, ADSTP=0, ADDIS=0 и ADEN=0. |
| 27:6 | Зарезервировано |
| 5 | JADSTP: команда прекращения принудительного преобразования.  Используется для того, чтобы остановить выполняемую операцию принудительного преобразования.  Значения устанавливаются программно, сбрасываются аппаратно.   * 0: значение по умолчанию; * 1: остановить преобразование.   Запись бита возможна, только если JADSTART=1 и ADDIS=0 (АЦП включен и выполняет принудительное преобразование, и при этом в очереди нет ожидающего напроса на отключение АЦП).  В автоматическом принудительном режиме (JAUTO=1) установка ADSTP приостанавливает и принудительное, и обычное преобразование (не используйте ADSTP вместо JADSTP). |
| 4 | ADSTP: команда прекращения принудительного преобразования.  Используется для того, чтобы остановить выполняемую операцию обычного преобразования.  Значения устанавливаются программно, сбрасываются аппаратно.   * 0: значение по умолчанию; * 1: остановить преобразование.   Запись бита возможна, только если JADSTART=1 и ADDIS=0 (АЦП включен и выполняет принудительное преобразование, и при этом в очереди нет ожидающего напроса на отключение АЦП).  В автоматическом принудительном режиме (JAUTO=1) установка ADSTP приостанавливает и принудительное, и обычное преобразование (не используйте ADSTP вместо JADSTP).  В сдвоенных режимах АЦП, одновременном и чередующемся), только бит ADSTP ведущего АЦП используется для остановки. |
| 3 | JADSTART: Начало принудительного преобразования на соответствующих каналах. Зависит от битов JEXTEN.  Преобразование начнется немедленно при поступлении программной команды (режим программного триггеринга, software trigger) или аппаратного сигнала, «события» (режим аппаратного триггеринга, hardware trigger).  Значения устанавливаются программно, сбрасываются аппаратно.   * 0: значение по умолчанию; * 1: начать операцию преобразования.   Запись бита возможна, только если JADSTART=1 и ADDIS=0 (АЦП включен и выполняет принудительное преобразование, и при этом в очереди нет ожидающего напроса на отключение АЦП).  В режиме автоматического принудительного преобразования и обычном режиме автоматические принудительные преобразования инициируются установкой бита ADSTART (при этом JADSTART должен быть сброшен). |
| 2 | ADSTART: начало обычного преобразования.  Аналогично JADSTART. |
| 1 | ADDIS: выключение АЦП.  Значения устанавливаются программно, сбрасываются аппаратно.   * 0: значение по умолчанию; * 1: выключить АЦП.   Бит может быть записан, только когда ADEN=0, JADSTART=0, ADSTART=0. |
| 0 | ADEN: включение АЦП.  Значения устанавливаются программно, сбрасываются аппаратно.   * 0: значение по умолчанию; * 1: включить АЦП.   Бит может быть установлен только тогда, когда все регистры CR сброшены в 0, кроме ADVREGEN, который должен быть сконфигурирован до включения АЦП (при этом нужно подождать, пока система закончит конфигурирование). |