МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ: МИКРОКОНТРОЛЛЕРЫ STM8S

*Рекомендовано в качестве учебного пособия
Редакционно-издательским советом
Томского политехнического университета*

Издательство
Томского политехнического университета
2014

УДК 681.322(075.8)
ББК 32.973.26-04я73
О-75

Авторы
С.Н. Торгаев, И.С. Мусоров, Д.С. Чертихина,
М.В. Тригуб, А.Н. Рыбаков

О-75 **Основы микропроцессорной техники: микроконтроллеры STM8S**: учебное пособие / С.Н. Торгаев, И.С. Мусоров, Д.С. Чертихина и др.; Томский политехнический университет. − Томск: Изд-во Томского политехнического университета, 2014. – 130 с.

В пособии изложено описание работы основных периферийных устройств микроконтроллеров STM8S, а также служебных регистров, необходимых для работы с данным микроконтроллером. Приводится большое количество примеров программ (написанных в программе IAR Embedded Workbench), осуществляющих настройку периферийных устройств.

Пособие предназначено для студентов, обучающихся по направлениям 11.03.04 «Электроника и наноэлектроника», 12.03.04 «Биотехнические системы и технологии».

УДК 681.322(075.8)
ББК 32.973.26-04я73

Рецензенты

Кандидат физико-математических наук, доцент ТГУ
Н.Г. Кушик

Кандидат физико-математических наук
старший научный сотрудник Института оптики атмосферы СО РАН
Д.В. Шиянов

Содержание

Введение

На сегодняшний день одними из наиболее распространённых восьми разрядных микроконтроллеров являются *AVR*-микроконтроллеры. Однако в последнее время все большую популярность, в данном классе микроконтроллеров, набирают микроконтроллеры компании *STMicroelectronics*. Это связано с тем, что микроконтроллеры *STM*8 обладают рядом преимуществ. В частности, они имеют больший объём *flash*-памяти, являются достаточно недорогими, а также имеют большее количество периферийных устройств (таймеры, АЦП, *UART* и т.д.).

Микроконтроллеры *STM*8 во всем модельном ряде сохраняется *pin-to-pin* совместимость, т.е. если разработчику не хватает возможностей какого-то контроллера есть возможность замены его на более старшую модель, имеющую «богатую» периферию и/или больший объемом памяти. Все регистры данных микроконтроллеров также совместимы между разными моделями. В некоторых случаях могут быть небольшие расхождения, но этих случаев можно полностью избежать если при программировании использовать стандартную библиотеку от *STMicroelectronics*. Для сохранения совместимости в *STM*8 использован подход, при котором создается набор периферийных устройств на весь модельный ряд, вся периферия нумеруется, и для каждого микроконтроллера из этого набора ставится определенный комплект с сохранением нумерации. Таким образом в микроконтроллере вполне может быть *UART*2 но не быть *UART*1.

Данное учебное пособие посвящено изучению основных блоков микроконтроллеров серии *STM*8*S*. В пособии рассмотрены принципы работы и описание регистров основных периферийных устройств (порты ввода/вывода, таймеры, контроллер прерываний, АЦП). Также представлено большое количество программ, содержащих примеры настройки периферийных устройств на различные режимы работы.

Глава 1. Архитектура микроконтроллеров STM8

1.1. Особенности архитектуры микроконтроллеров *STM*8

В основе микроконтроллеров семейства ***STM*8** (рис 1.1) лежит *CISC*-ядро собственной разработки *STMicroelectronics*, разработанное на основе более ранних 8-разрядных контроллеров ***ST*7**. Ядро спроектировано по гарвардской архитектуре, подразумевающей разделение данных и команд.

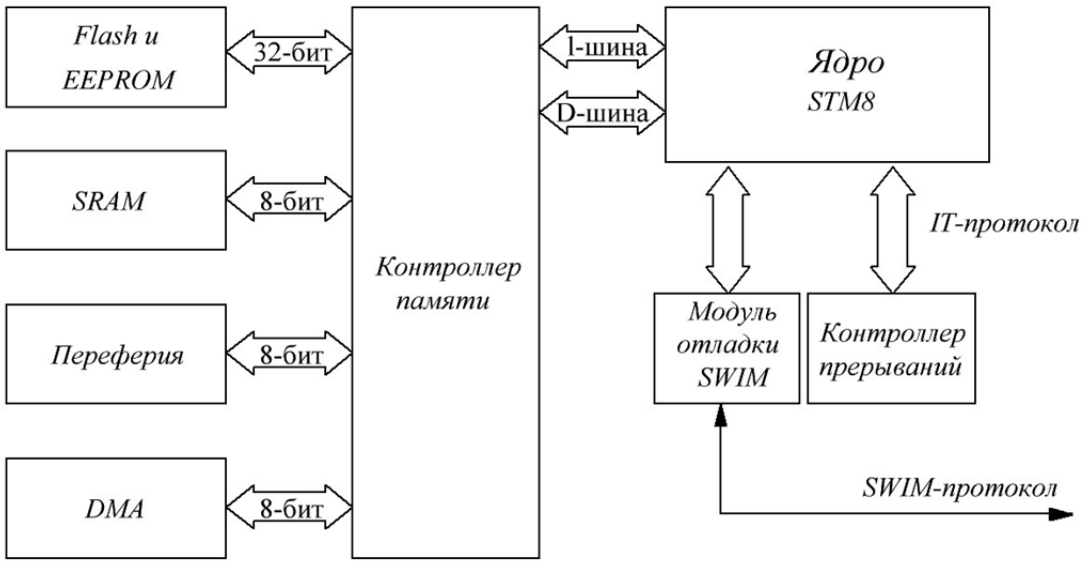

Рис. 1.1. Блок схема микроконтроллеров STM8

Существенным преимуществом микроконтроллеров ***STM*8** является наличие трехуровневого конвейера, позволяющего значительно повысить производительность. Конвейер организует выполнение команды за три шага: выборка (извлечение команды из памяти), декодирование (декодирование команды и чтение данных из памяти) и выполнение. Кроме этого, для увеличения производительности используется адресное пространство 16 Мбайт, 32-битная шина доступа к *Flash*-памяти, 16-битные индексные регистры, ядро имеет аппаратную поддержку знаковых операций сложения, умножения и деления.

Все эти новшества позволяют получить производительность процессора до 16 *MIPS* при работе на тактовой частоте 16 МГц. Весомый вклад в функциональность новой архитектуры внес контроллер вложенных прерываний с четырьмя уровнями приоритетов, детерминированным временем входа в обработчик, автоматическим сохранением и выгрузкой контекста стека. Доступна технология «*tail-chaining*», позволяющая при возникновении очереди прерываний не возвращаться обратно

в основную программу для переходя к следующему прерыванию, а сразу переходить в него.

В микроконтроллерах ***STM*8** доступно до 32 прерываний – как минимум, по одному на периферийный модуль; пять внешних прерываний - по одному прерыванию на порт ввода-вывода с программируемыми условиями реакции; три немаскируемых прерывания - ***RESET, TRAP*** (программное прерывание) и ***TLI*** (аппаратное прерывание по переднему или заднему фронту на специально выделенной для этого ножке).

Неоспоримым плюсом для разработчика является наличие у ***STM8*** возможности отладки с помощью системы ***SWIM*** (***Single Wire Interface Module***) и ***Debug Module***. Как следует из названия, для отладки и программирования потребуется всего одна ножка ***SWIM*** микроконтроллера (плюс задействуется еще ножка для сброса - ***NRST***). В дальнейшем, в случае необходимости, она может быть задействована как порт общего назначения и использована в проекте. Максимальная скорость передачи данных по ***SWIM***-интерфейсу составляет 145 байт/мс. Модуль отладки позволяет производить чтение/запись всей памяти и регистров периферии в режиме реального времени, обеспечивает доступ к данным без остановки ***CPU*** и предоставляет неограниченное количество точек останова команд.

1.2. Обзор семейства микроконтроллеров *STM*8

Семейство микроконтроллеров ***STM*8** делится на четыре линейки в зависимости от сферы применения:

- для промышленного использования - ***STM*8*S***;
- для применения в автомобильном сегменте - ***STM*8*A***;
- для решений с низким потреблением - ***STM*8*L***;
- для различных сенсорных приложений - ***STM*8*T***.

На сегодняшний день общее количество контроллеров ***STM*8** насчитывает уже более 120 наименований. Такое разнообразие позволяет точно подобрать нужный для разработки контроллер. Разработчику доступна вся стандартная периферия:

- 10/12-битные АПЦ;
- 12-битные ЦАП;
- большое количество 8/16-разрядных таймеров;
- встроенный *LCD*-драйвер;
- интерфейсы *CAN*, *USART*, *SPI*, *I2C*;
- до 128 кбайт *Flash*-памяти.

Рассмотрим подробно два семейства микроконтроллеров: ***STM8S*** и ***STM8L***.

Микроконтроллеры *STM8S*

Микроконтроллеры включают в себя четыре серии микроконтроллеров с различной производительностью, размером встроенной памяти и набором периферийных устройств на борту (рис. 1.2).

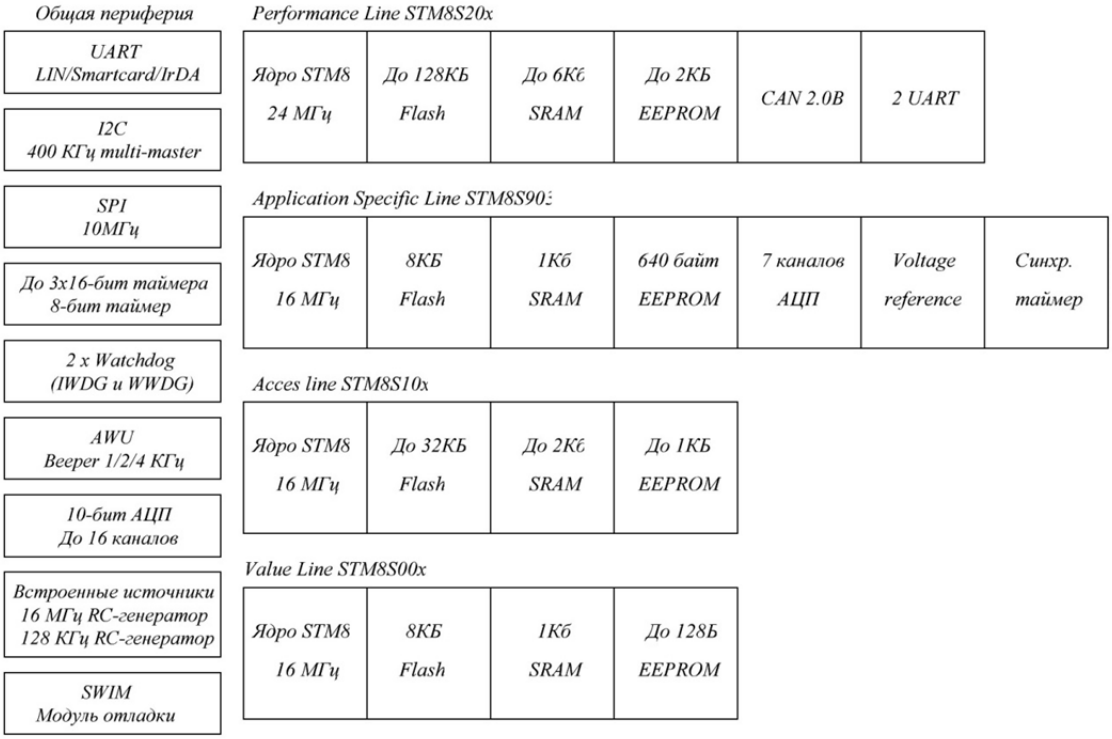

Рис. 1.2. Характеристики микроконтроллеров STM8S

Серия ***STM8S003/005/007 Value Line*** представляет собой контроллеры начального уровня с базовым набором функций и оптимальной ценой.

Серия ***STM8S103/105 Access Line*** имеет больше возможностей, большее разнообразие корпусов и сервис фабричного программирования.

Серия ***STM8S207/208 Performance Line*** имеет широкий набор периферийных устройств, более высокую производительность и подходит для средних и высокопроизводительных приложений.

Серия ***STM8S Application Specific Line*** обеспечивает решения задач по построению систем обработки аналоговых сигналов.

Ниже представлены основные характеристики микроконтроллеров семейства ***STM8S Value Line***:

- ядро *STM*8 с гарвардской архитектурой;

- максимальная частота работы- 16МГц/16*MIPS*;
- до 64 кбайт *Flash*-памяти;
- до 6 кбайт *SRAM*-памяти;
- 28 байт *EEPROM*-памяти;
- 10-битное АЦП (до 16 каналов);
- 6 таймеров (16 и 8 бит);
- коммуникационные интерфейсы: *I2C*, *UART*, *SPI*;
- расширенный температурный диапазон: -40...85°C;
- корпуса *UFQFN*20, *TSSPO* 20, *LQFP*32, *LQFP*48.

Интерфейсы микроконтроллеров *STM8S*

I2C-интерфейс. Микроконтроллеры имеют на борту один модуль ***I2C*** со стандартным набором параметров. Модуль может работать в режиме «Ведущий» (*Master*) или «Ведомый» (*Slave*). Поддерживается режим «*MultiMaster*». Доступны стандартные (*Standard*) скорости передачи данных до 100 кГц и быстрая (*Fast*) передача данных на частотах до 400 кГц. Возможна 7- или 10-битная адресация. Реализовано вывод микроконтроллера из режима пониженного энергопотребления при опознавании своего адреса в посылке.

SPI-интерфейс. Микроконтроллеры содержат один модуль ***SPI***. Модуль может работать в режиме «*Master*» или «*Slave*» и поддерживать полнодуплексную, полудуплексную и симплексную передачу данных. Максимальная скорость передачи данных - до 10 Мбит/с. Данные могут быть переданы старшим либо младшим битом вперед. Полярность и фаза тактового сигнала может быть программно изменена. Аналогично интерфейсу ***I2C***, реализовано вывод микроконтроллера из режима пониженного энергопотребления.

UART-интерфейс. Микроконтроллеры ***STM8S Value Line*** содержат до двух модулей ***UART***. Модули ***UART*** могут работать на скоростях до 1 Мбит/с. Данные предаются 8/9-битными словами с одним или двумя стоповыми битами и контролем четности. Возможно подключение смарт-карты в соответствии со стандартом *ISO* 7618-3.Также ***UART*** может работать в ***SPI***-режиме. При этом микроконтроллер выступает в роли ведущего ***SPI***-устройства.

Аналогово-цифровой преобразователь

Микроконтроллеры ***STM8S Value Line*** содержат один аналогово-цифровой преобразователь (АЦП) с разрешающей способностью 10 бит. Максимальное количество входных аналоговых каналов - 16. Система настроек встроенного аналогового мультиплексора позволяет задавать

режимы однократного или непрерывного преобразования с возможностью использования режима сканирования. Доступен аналоговый сторожевой таймер с работой по нижнему и верхнему порогу с возможностью генерации прерывания. Для запуска преобразования по внешнему сигналу к триггеру АЦП подключена одна из ножек микроконтроллера.

Питание микроконтроллеров *STM8S*

Для питания микроконтроллера потребуется источник питания с напряжением 2.95÷5.5 В. Такой диапазон питающих напряжений очень удобен в связи с тем, что контроллер легко интегрируется в систему с питанием как 3.3 В, так и 5 В. Для питания ядра микроконтроллера используется встроенный преобразователь напряжения. Подача питания на аналоговую периферию происходит через специально выделенные для этой цели ножки.

После подачи питания или сброса микроконтроллер начинает работать в активном режиме (***Run Mode***). В этом режиме потребление микроконтроллера максимально. Но благодаря гибкой системе тактирования и питания периферии, есть возможность сократить потребление, снижая скорость работы контроллера и задействовав только необходимую периферию.

При реализации приложений, критичных по потреблению, микроконтроллер поддерживает четыре режима пониженного энергопотребления:

- Режим «***Wait Mode***». В этом режиме центральный процессор остановлен, но периферийные устройства продолжают работать. Все регистры и содержимое оперативной памяти сохраняется. Пробуждение выполняется с помощью внутреннего или внешнего прерывания, либо общего сброса.
- Режим «***Active Halt mode with regulator on***». В этом режиме центральный процессор и тактирование периферии остановлено. “Пробуждения” генерируются от внутреннего программируемого таймера (***AWU***). Основной регулятор напряжения остается включенным, поэтому потребление тока выше, чем в режиме «***Active Halt Mode with regulator off***», но времени на “пробуждение” требуется меньше. “Пробуждение” также возможно от внешнего прерывания или сброса.
- Режим «***Active Halt Mode with regulator off***». Этот режим похож на режим «***Active Halt mode with regulator on***», за исключением того, что внутренний регулятор напряжения выключен и “пробуждение” происходит медленнее.

• Режим «***Halt Mode***». В этом режиме микроконтроллер потребляет меньше всего энергии. Центральный процессор и периферийное тактирование остановлены, главный регулятор напряжения выключен. Вывод из режима осуществляется внешним прерыванием или сбросом.

Микроконтроллеры *STM8L*

Линейка микроконтроллеров ***STM8L*** включает в себя четыре серии малопотребляющих микроконтроллеров с различной производительностью, размером встроенной памяти, набором периферийных устройств на борту и, собственно, потреблением (рис. 1.3).

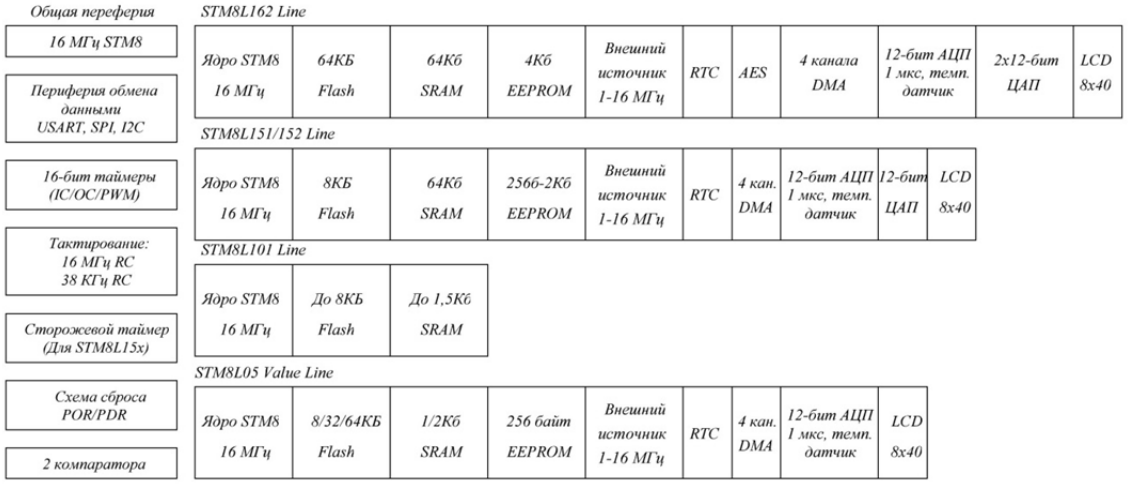

Рис. 1.3. Характеристики микроконтроллеров STM8L

Серия ***STM8L*051/052 *Value Line*** представляет собой малопотребляющие контроллеры с базовым набором функций и лидирующим соотношением цена/производительность.

Серия ***STM8L*101** предлагает контроллеры начального уровня с минимальным набором периферии.

Серия ***STM8L*151/152** имеет широкий набор периферийных устройств и подходит для средних и высокопроизводительных малопотребляющих приложений.

Серия ***STM8L*162** отличается от предыдущей серии наличием модуля шифрования. Основная особенность данной серии – это низкое потребление и направленность этих микроконтроллеров на применение в недорогих устройствах с батарейным питанием.

По сравнению с ***STM8S*** данная линейка имеет усиленную аналоговую периферию: 12-битное АЦП, 12-битный ЦАП и компараторы. Для управления несложными жидкокристаллическими дисплеями доступен *LCD*-контроллер. Преимуществом данных микроконтроллеров

является наличие ***DMA***-контроллера, редко встречающегося в 8-битных микроконтроллерах.

Глава 2. Порты ввода/вывода общего назначения

2.1. Режимы работы портов ввода/вывода

Порты ввода/вывода общего назначения используются для обмена данными между микроконтроллером и внешними устройствами. Каждый вывод порта может быть индивидуально запрограммирован как цифровой вход или цифровой выход. Кроме того, некоторые порты имеют альтернативные функции, такие как: аналоговый вход, внешнее прерывание, вход/выход периферийных устройств микропроцессора и т.д. В процессе работы ножка порта может выполнять только одну альтернативную функцию. Определенный порт будет вести себя как вход или выход в зависимости от состояния регистра направления данных порта. На рис. 1.1 показана блок схема портов ввода/вывода общего назначения микроконтроллера ***STM8S*** [XX].

Основные особенности портов ввода/вывода:

- биты порта могут быть настроены индивидуально;
- возможные режимы входа: ***floating input*** (дифференциальный вход) или ***input with pull-up*** (вход с подтягивающим резистором);
- возможные режимы выхода: ***push-pull output*** (двухтактный выход) или ***pseudo-open-drain output*** (выход с или псевдо-открытым стоком);
- отдельные регистры для ввода и вывода данных;
- внешние прерывания могут быть включены или отключены индивидуально;
- триггер Шмитта может быть отключен на аналоговых входах для уменьшения энергопотребления;
- вход до 5 В.

К каждому порту привязаны следующие регистры: выходных данных (***ODR***), регистр входных данных (***IDR***), регистр направление данных (***DDR***), регистр управления 1 (***CR*1**) и регистр управления 2 (***CR*2**). Порт настраивается на ввод/вывод с помощью соответствующих битов в регистрах ***DDR, CR*1** и ***CR*2**. Каждый ***N***-ый бит в регистрах соответствует ***N***-ому выводу порта. Возможные конфигурации порта приведены в табл. 1.1.

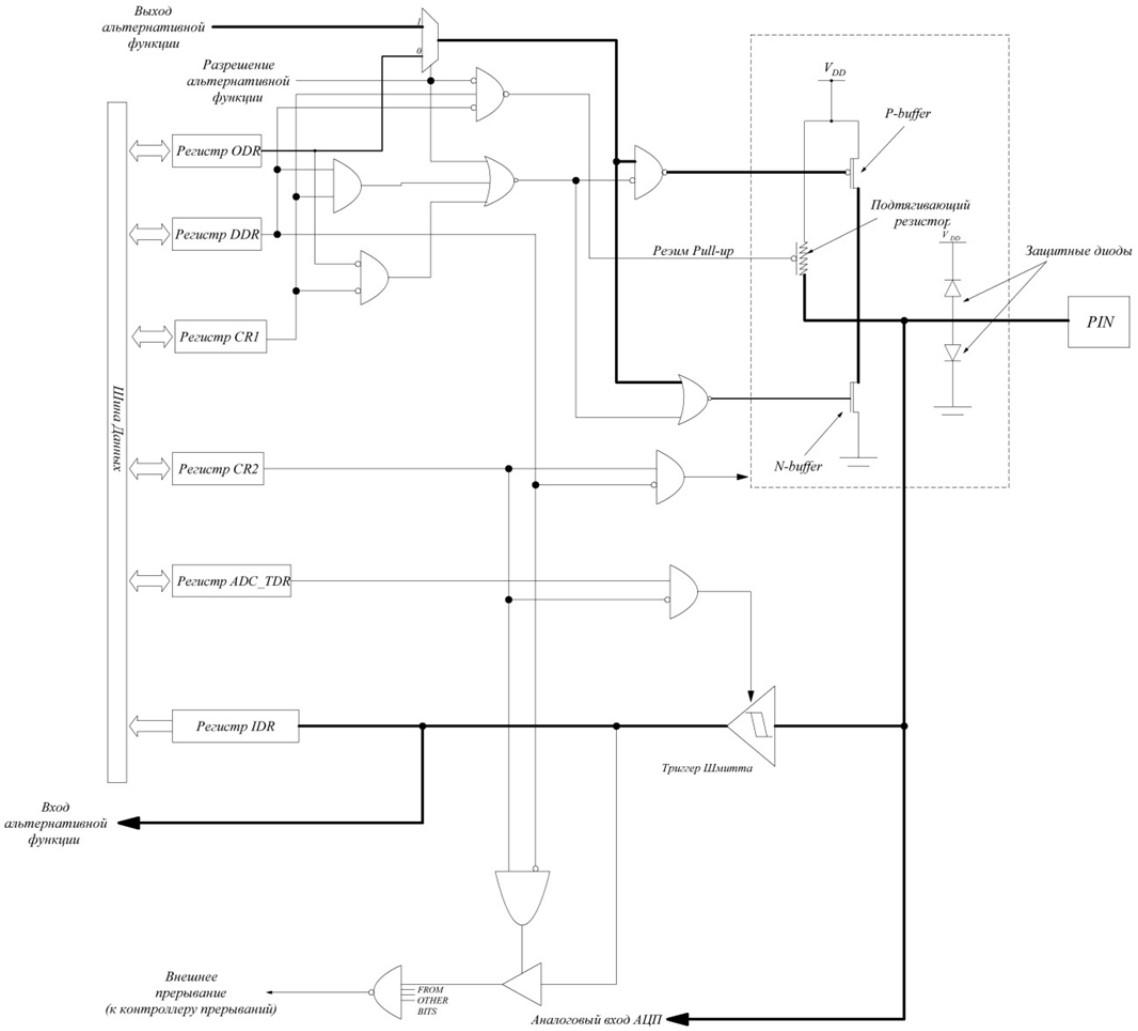

Рис. 1.1. Блок схема портов ввода-вывода

Таблица 1.1

Конфигурации портов ввода/вывода общего назначения

Режим	Бит ***DDR***	Бит ***CR*1**	Бит ***CR*2**	Функция	*Pull-up*	Верхний транзистор	Диоды	
Ввод	0	0	0	Без подтяжки, без прерываний	*Off*	*off*	*on*	*on*
	0	1	0	С подтяжкой, без прерываний	*On*			
	0	0	1	Без подтяжки, с прерываниями	*Off*			
	0	1	1	С подтяжкой, с прерываниями	*on*			
Вывод	1	0	0	Открытый сток	*off*	*Off*		
	1	1	0	Двухтактный выход		*On*		

Окончание табл. 1.1

	1	x	1	Скорость переключений до 10МГц		Зависит от ***CR*1**		
	1	x	x	Реальный открытый сток	Отсутствует			

Режим ввода

Настройка вывода порта на работу в режиме ввода осуществляется очисткой соответствующего бита в регистре ***DDR***. В этом режиме чтение регистра ***IDR*** возвращает значение соответствующего вывода. Как показано в таблице 1.1, выводы порта могут быть сконфигурированы на четыре различные режима ввода, однако, на практике, не все порты имеют возможности прерывания или подтяжки. После сброса, все выводы портов находятся в состоянии ***floating input*** (дифференциальный вход).

Альтернативная функция

Некоторые порты ввода/вывода могут быть использованы для выполнения альтернативной функции входа. Например, порт может использоваться в качестве входа захвата таймера и т.д. Альтернативные функции порта выбираются автоматически в регистрах соответствующего периферийного устройства. Для альтернативной функции ввода, необходимо сконфигурировать вывод порта в режим ***floating input*** или ***pull-up input*** в регистрах ***DDR*** и ***CR*1**.

Возможность прерывания

Порт можно настроить на вход с прерыванием, путем установки битов в регистре ***CR*2**. В данном случае микроконтроллер будет реагировать на прерывание по перепаду или по уровню входного сигнала. Чувствительность к переднему или заднему фронту программируется для каждого вектора прерывания в регистре ***EXTI_CR[*2:1*].*** Внешние прерывания возможны только если порт находится в режиме ввода.

Режимы вывода

Установка бита в регистре ***DDR*** выбирает режим вывода на соответствующей ножке порта. Устанавливая или сбрасывая бит в регистре ***ODR*** можно установить на соответствующем выводе значение логической «1» или «0», соответственно. Чтение регистра ***IDR*** возвращает значение соответствующего вывода. С помощью регистров контроля (***CR*1**

и ***CR*2**) возможно настроить следующие режимы выхода: ***push-pull output, open-drain output***.

2.2. Регистры портов ввода/вывода

Регистр направления данных

Px_DDR (где ***x*** – имя порта):

7	6	5	4	3	2	1	0
***DDR*7**	***DDR*6**	***DDR*5**	***DDR*4**	***DDR*3**	***DDR*2**	***DDR*1**	***DDR*0**
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Данный регистр служит для настройки портов на ввод или вывод данных. Все биты данного регистра устанавливаются и очищаются программно:

0: ввод;

1: вывод.

Регистр управления портов 1

Px_CR1 (где ***x*** – имя порта):

7	6	5	4	3	2	1	0
***C*17**	***C*16**	***C*15**	***C*14**	***C*13**	***C*12**	***C*11**	***C*10**
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Все биты данного регистра устанавливаются и очищаются программно. Они определяют функции порта в зависимости от выбранного режима (ввод/вывод):

- Входной порт (***DDR*=0**):
  0: дифференциальный вход (***floating input***);
  1: ввод с подтягивающим резистором (***pull-up***).
- Выходной порт (***DDR*=1**):
  0: с открытым стоком (***open drain***);
  1: двухтактный (***push-pull***).

Регистр управления портов 2

Px_CR2 (где ***x*** – имя порта):

7	6	5	4	3	2	1	0
***C*27**	***C*26**	***C*25**	***C*24**	***C*23**	***C*22**	***C*21**	***C*20**
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Биты данного регистра определяют различные функции порта в зависимости от выбранного режима (ввод/вывод). Эти биты устанавливаются и сбрасываются программно.

- Входной порт (***DDR*=0**):
  0: запретить внешнее прерывание;
  1: разрешить внешнее прерывание.
- Выходной порт (***DDR*=1**):
  0: выходная частота до 2 МГц;
  1: выходная частота до 10 МГц.

Регистр выходных данных

Px_ODR (где ***x*** – имя порта):

7	6	5	4	3	2	1	0
***ODR*7**	***ODR*6**	***ODR*5**	***ODR*4**	***ODR*3**	***ODR*2**	***ODR*1**	***ODR*0**
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Биты данного регистра определяют выходное состояние порта при работе в режиме вывода. Регистр может быть использован, как для чтения, так и для записи данных.

Регистр входных данных порта

Px_IDR (где ***x*** – имя порта):

7	6	5	4	3	2	1	0
***IDR*7**	***IDR*6**	***IDR*5**	***IDR*4**	***IDR*3**	***IDR*2**	***IDR*1**	***IDR*0**
r	*r*	*r*	*r*	*r*	*r*	*r*	*r*

Биты данного регистра хранят текущее состояние соответствующего вывода порта. Регистр может быть использован для чтения, вне зависимости от того как настроен порт на ввод или на вывод данных.

2.3. Примеры настройки портов ввода/вывода

Пример настройки порта микроконтроллера на вывод

Представленная ниже программа реализует мерцание светодиода, подключенного к нулевой ножке порта ***D***. Данный светодиод зажигается уровнем логического нуля на порте.

```
#include "iostm8s003k3.h"      //подключение заголовочного файла с объ-
                               явлениями регистров, масок и битов
int i;                         //объявление переменной
int main( void )               //основная программа
{
//Настройка нулевого разряда порта D
PD_DDR_bit.DDR0=1;             //0-вход; 1-выход
```

```
PD_CR1_bit.C10=1;          //0-выход с открытым стоком; 1-выход push-pull
PD_CR2_bit.C20=0;          //скорость переключения: 0-2 MHz; 1-10MHz
PD_ODR_bit.ODR0= 0;        //зажигаем светодиод
  for ( ; ; )              //бесконечный цикл
  {
     for (i=0;i<=5000;i++); //временная задержка
     {
     }
//инверсия нулевого разряда порта D
PD_ODR_bit.ODR0 =! PD_ODR_bit.ODR0;
  }
}
```

Пример настройки порта микроконтроллера на ввод

Представленная ниже программа реализует инверсию нулевой ножки порта ***D*** при нажатии кнопки, подключенной к седьмой ножкой порта ***B***. В разомкнутом состоянии на ножку порта подана логическая “1”, при нажатии кнопки на входе формируется сигнал низкого уровня.

```
#include " iostm8s003k3.h"  //подключение заголовочного файла
int i,j;                   //объявление переменных
int main( void )           //основная программа
{
//Настройка нулевого разряда порта D
PD_DDR_bit.DDR0=1;         //0-вход; 1-выход
PD_CR1_bit.C10=1;          //0-выход с открытым стоком; 1-выход push-pull
PD_CR2_bit.C20=0;          //скорость переключения: 0-2 MHz; 1-10MHz
PD_ODR_bit.ODR0= 0;        //зажигаем светодиод
//Настройка седьмого разряда порта B
PB_DDR_bit.DDR7=0;         //0-вход; 1-выход
PB_CR1_bit.C17=0;          //0-дифференциальный вход; 1-вход с подтягивающим резистором
PB_CR2_bit.C27=0;          //0-прерывания запрещены; 1-прерывания разрешены
  for (; ;)                //бесконечный цикл
  {
    if (PB_IDR_bit.IDR7!=1) //проверка нажатия кнопки
```

```
    {
    //Инверсия нулевого разряда
    PD_ODR_bit.ODR0 =!PD_ODR_bit.ODR0;
      for (i=0;i<=30000;i++);        //временная задержка
      {
        for (j=0;j<=30000;j++);
        { }
      }
    }
  }
}
```

Глава 3. Прерывания микроконтроллера *STM8S*

3.1. Контроллер прерываний. Вектора прерываний

В микроконтроллерах семейства ***STM8S*** поддерживаются два вида прерываний:

- немаскируемые прерывания: ***RESET, TLI*** и ***TRAP;***
- маскируемые прерывания: внешние прерывания или прерывания от внутренней периферии микроконтроллера.

Контроллер прерываний микроконтроллера выполняет следующие функции:

- Управляет аппаратными прерываниями:
  - Внешними прерываниями, которые возможны на большинстве портов ввода/вывода. Все они имеют свой заданный вектор прерывания. Чувствительность к уровню сигнала может быть изменена;
  - Прерываниями от внутренней периферии.
- Управляет программными прерываниями (***TRAP***).
- Управляет вложенными прерываниями.

Для работы с прерываниями необходимо разрешить все прерывания. Глобальное разрешения прерываний осуществляется командой ***asm("rim")***, а запрещение командой ***asm("sim")***. Так же необходимо создать обработчики прерываний. Обработчики прерываний определяются следующим образом:

#pragma vector=<номер прерывания> (табл. 3.1)

__interrupt void <имя прерывания>,

<номер прерывания> – номер прерывания в ***hex*** формате. Берется значение из технической документации микроконтроллера и к нему прибавляется 2.

<имя прерывания> – имя, которое будет именем подпрограммы обработчика прерывания (определяется пользователем).

Таблица 3.1

Вектора прерываний микроконтроллера STM8S

№	Источник прерывания	Описание	Выход из режима останова	Вектор прерывания
	RESET	Сброс	Да	0x00 8000
	TRAP	Программное прерывание	-	0x00 8004
0	*TLI*	Прерывание верхнего уровня	-	0x00 8008
1	*AWU*	Автоматический выход из состояния останова	-	0x00 800*C*
2	*CLK*	Контроллер тактирования (*clock controller*)	-	0x00 8010

Окончание табл. 3.1

3	*EXTI*0	Внешнее прерывание порта *A*	Да	0x00 8014
4	*EXTI*1	Внешнее прерывание порта *B*	Да	0x00 8018
5	*EXTI*2	Внешнее прерывание порта *C*	Да	0x00 801*C*
6	*EXTI*3	Внешнее прерывание порта *D*	Да	0x00 8020
7	*EXTI*4	Внешнее прерывание порта *E*	Да	0x00 8024
8		Зарезервирован	-	0x00 8028
9		Зарезервирован	-	0x00 802*C*
10	*SPI*	Окончание передачи	Да	0x00 8030
11	*TIM1*	*TIM*1 обновление/переполнение сверху (*overflow*)/переполнение снизу *(underflow*)/запуск	-	0x00 8034
12	*TIM*1	*TIM*1 захват/сравнение	-	0x00 8038
13	*TIM*2	*TIM*2 обновление/переполнение сверху (*overflow*)	-	0x00 803*C*
14	*TIM*2	*TIM*2 захват/сравнение	-	0x00 8040
15		Зарезервирован	-	0x00 8044
16		Зарезервирован	-	0x00 8048
17	*UART*1	*Tx* завершение отправки	-	0x00 804*C*
18	*UART*1	Заполнение регистра данных (прием окончен)	-	0x00 8050
19	I^2C	Прерывание I^2C	Да	0x00 8054
20		Зарезервирован	-	0x00 8058
21		Зарезервирован	-	0x00 805*C*
22	*ADC*1	*ADC*1 окончание преобразования	-	0x00 8060
23	*TIM*4	*TIM*4 захват/сравнение	-	0x00 8064
24	*Flash*	*EOP/WR_PG_DIS*	-	0x00 8068
Reserved				0x00 806*C to* 0x00 807*C*

3.2. Регистры прерываний

Регистры настройки приоритета прерываний

Регистры настройки приоритета прерываний (***ITC_SPRx***).

	7	6	5	4	3	2	1	0
***ITC_SPR*1**	***VECT*3*SPR***[1:0]		***VECT*2*SPR***[1:0]		***VECT*1*SPR***[1:0]		***VECT*0*SPR***[1:0]	
***ITC_SPR*2**	***VECT*7*SPR***[1:0]		***VECT*6*SPR***[1:0]		***VECT*5*SPR***[1:0]		***VECT*4*SPR***[1:0]	
***ITC_SPR*3**	***VECT*11*SPR***[1:0]		***VECT*10*SPR***[1:0]		***VECT*9*SPR***[1:0]		***VECT*8*SPR***[1:0]	
***ITC_SPR*4**	***VECT*15*SPR***[1:0]		***VECT*14*SPR***[1:0]		***VECT*13*SPR***[1:0]		***VECT*12*SPR***[1:0]	
***ITC_SPR*5**	***VECT*19*SPR***[1:0]		***VECT*18*SPR***[1:0]		***VECT*17*SPR***[1:0]		***VECT*16*SPR***[1:0]	
***ITC_SPR*6**	***VECT*23*SPR***[1:0]		***VECT*22*SPR***[1:0]		***VECT*21*SPR***[1:0]		***VECT*20*SPR***[1:0]	
***ITC_SPR*7**	***VECT*27*SPR***[1:0]		***VECT*26*SPR***[1:0]		***VECT*25*SPR***[1:0]		***VECT*24*SPR***[1:0]	
***ITC_SPR*8**	Зарезервировано				***VECT*3*SPR***[1:0]		***VECT*3*SPR***[1:0]	
	rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты [7:0] ***VECTxSPR***[1:0]: Биты настройки приоритета вектора ***x:***

00: средний приоритет;

01: низкий приоритет;

10: не оказывает ни какого влияния;

11: высокий приоритет.

Данные восемь регистров служат для определения приоритета прерываний. Номера всех векторов прерываний приведены в таблице 3.1. В регистре ***ITC_SPR*1** биты 1:0 принадлежат вектору прерывания ***TLI***.

Регистр настройки внешних прерываний 1

Регистр настройки внешних прерываний 1 (***EXTI_CR*1**).

7	6	5	4	3	2	1	0
V	-	***I*1**	***H***	***I*0**	***N***	***Z***	***C***
r	r	rw	r	rw	r	r	r

• Биты 5 и 3 ***I***[1:0]: Биты приоритета прерываний.

Данные биты показывают приоритет текущего запроса на прерывание. Когда происходит запрос на прерывание, приоритет соответствующего вектора автоматически загружается из регистра ***ITC_SPRx***.

Регистр настройки внешних прерываний 1

Регистр настройки внешних прерываний 1 (***EXTI_CR*1**).

7	6	5	4	3	2	1	0
PDIS[1:0]		***PCIS***[1:0]		***PBIS***[1:0]		***PAIS***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:6 ***PDIS***[1:0]: Биты чувствительности внешнего прерывания порта ***D***:

00: прерывания по низкому уровню и спадающему фронту сигнала;

01: прерывания только по переднему фронту сигнала;

10: прерывания только по спадающему фронту сигнала;

11: прерывания по высокому уровню и переднему фронту сигнала.

Данные биты могут быть изменены только в случае, если биты ***I*1** и ***I*2** в регистре ***CPU_CCR*** установлены*.

• Биты 5:4 ***PCIS***[1:0]: Биты чувствительности внешнего прерывания порта ***C***.

Описание аналогично описанию битов ***PDIS.***

• Биты 3:2 ***PBIS***[1:0]: Биты чувствительности внешнего прерывания порта ***B***.

Описание аналогично описанию битов ***PDIS.***

• Биты 1:0 ***PAIS***[1:0]: Биты чувствительности внешнего прерывания порта ***A***.

Описание аналогично описанию битов ***PDIS.***

*** – Загрузку данного регистра необходимо осуществлять до команд глобального разрешения и запрещения прерываний.**

Регистр настройки внешних прерываний 2

Регистр настройки внешних прерываний 2 (***EXTI_CR2***).

7	6	5	4	3	2	1	0
Зарезервировано					***TLIS***	***PEIS***[1:0]	
					rw	*rw*	*rw*

• Биты 7:3 Зарезервированы.

• Бит 2 ***TLIS:***Чувствительность прерываний верхнего уровня:

0: Прерывание по спадающему фронту;

1: Прерывание по переднему фронту.

Данные биты могут быть изменены программно при условии, что внешнее прерывание, на соответствующем порту (***PD***7), запрещено.

• Биты 1:0 ***PEIS***[1:0]: Биты чувствительности внешнего прерывания порта ***E***.

Описание аналогично описанию битов ***PDIS.***

3.3. Пример настройки порта микроконтроллера для работы по прерыванию

Представленная ниже программа реализует инверсию нулевой ножки порта ***D*** при нажатии кнопки по внешнему прерыванию (кнопка соединена с седьмой ножкой порта ***B***).

```
#include " iostm8s003k3.h"        //подключение заголовочного файла
int i,j;                          //объявление переменных
void interrupt_init(void);        //Объявление подпрограммы настройки
                                  прерываний
#pragma vector=0x06
__interrupt void EXTI_PB7(void);  //Имя вектора внешнего прерыва-
ния
int main (void)                   //Основная программа
{
PB_DDR_bit.DDR7=0;                //0-вход; 1-выход
PB_CR1_bit.C17=0;                 //0-дифференциальный вход; 1-вход с под-
                                  тягивающим резистором
```

```
PB_CR2_bit.C27=1;          //0-прерывания запрещены; 1-прерывания разрешены
PD_DDR_bit.DDR0=1;         //0-вход; 1-выход
PD_CR1_bit.C10=1;          //0-выход с открытым стоком; 1-выход типа Push-pull
PD_CR2_bit.C20=0;          //Скорость переключения: 0-до 2 MHz; 1-до 10MHz
PD_ODR_bit.ODR0= 0;        //Зажигаем светодиод
interrupt_init();          //Вызов подпрограммы настройки прерываний
for (;;)
{}
 }

//подпрограмма настройки прерываний
void interrupt_init(void)
{
asm("rim");                //Глобальное разрешение прерываний
}

//Программа обработки прерывания
__interrupt void EXTI_PB7(void)
{
//Инверсия нулевого бита порта D
PD_ODR_bit.ODR0 =! PD_ODR_bit.ODR0;
  for (i=0;i<=30000;i++);  //временная задержка
  {
    for (j=0;j<=30000;j++);        //временнаязадержка
    {}
  }
}
```

Глава 4. Таймеры микроконтроллера *STM*8*S*

Микроконтроллеры семейства ***STM*8*S*** могут быть оснащены таймерами трех различных типов:

- таймер с расширенным управлением (***advanced control***) ***TIM*1**;
- таймер общего назначения (***general purpose***) ***TIM*2, *TIM*3**, ***TIM*5**;
- базовый таймер (***basic***) ***TIM*4, *TIM*6**.

Основные свойства таймеров приведены в табл. 4.1.

Таблица 4.1

Основные характеристики таймеров

Таймер	Разрядность	Тип счета	Предделитель	Каналы захвата/ сравнения	Комплементарные выходы	Повторитель счетчика	Вход внешнего запуска	Вход внешнего сброса
***TIM*1**	16 бит	Вверх/ вниз	Любое целое от 1 до 65536	4	3	Да	1	1
***TIM*2**		Вверх	Степень числа 2 от 1 до 32768	3	Нет	Нет	0	0
***TIM*3**				2				
***TIM*4**	8 бит		Степень числа 2 от 1 до 128	0				
***TIM*5**	16 бит	Вверх	Степень числа 2 от 1 до 32768	3	Нет	Нет	1	0
***TIM*6**	8 бит		Степень числа 2 от 1 до 128	0			0	

4.1. 16-разрядный расширенный таймер (*TIM*1)

***TIM*1** состоит из 16-разрядного авто-перезагружаемого счетчика, который может считать как вверх, так и вниз. Данный таймер управляется программируемым предварительным делителем и может использоваться для различных целей, таких как:

- отсчет времени;
- измерение длительности внешних импульсов;
- формирование выходных сигналов;
- генерация прерываний для различных целей;

• синхронизация с таймерами ***TIM5/TIM6*** или с внешними сигналами.

Этот таймер идеально подходит для контроля и может работать как от внутреннего тактового генератора, так и от внешнего источника тактовых сигналов. 16-битный программируемый делитель позволяет поделить тактовую частоту на любое число в диапазоне от 1 до 65536. Таймер имеет 4 независимых канала, которые могут быть поочередно сконфигурированы как:

• вход захвата;
• выход сравнения;
• широтно-импульсная модуляция (ШИМ);
• 6-ступенчатая ШИМ;
• импульсный режим вывода;
• дополнительные выходы на трех каналах, с программируемыми вставками запаздывания.

Вход сброса нужен для того чтобы установить выходной сигнал таймера в исходное состояние. Генерация прерывания таймера может происходить при следующих событиях:

• по переполнению счетчика;
• событие запуска (начало/конец счета, инициализация или счет на внутренний/внешний запуск);
• вход захвата;
• выход сравнения;
• вход сброса.

Предварительный делитель частоты, 16-битный счетчик, авто-перезагружаемый регистр и регистр повторений счетчика могут быть загружены или считаны программно. Автоматически перезагружаемый регистр состоит из перезагружаемого и теневого регистров. Существует два режима работы с авто-перезагружаемым регистром:

• Предварительная загрузка разрешена (бит ***ARPE*** в регистре ***TIM1_CR*1** установлен) в этом режиме, когда информация записана в авто-перезагружаемый регистр, она хранится в предварительно загружаемом регистре и переносится в теневой регистр на следующем событии;

• Предварительная загрузка запрещена (бит ***ARPE*** в регистре ***TIM1_CR*1** сброшен) в этом режиме, когда информация записана в авто-перезагружаемый регистр, она немедленно переносится в теневой регистр.

На рис. 4.1 представлена блок схема Таймера 1 микроконтроллера.

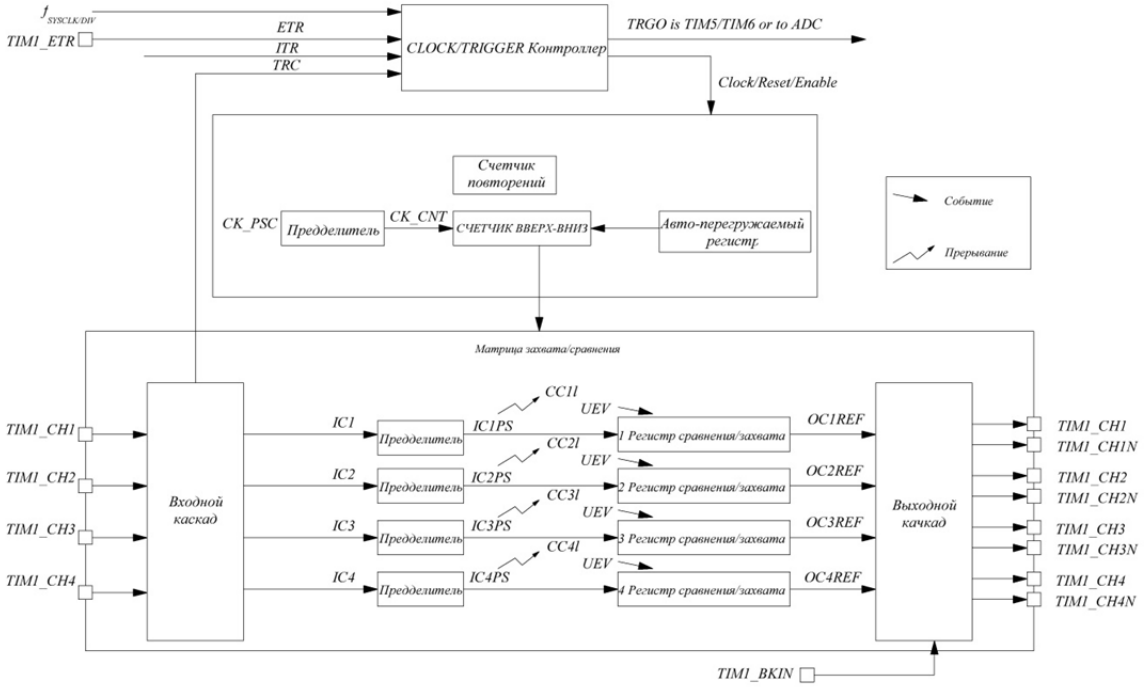

Рис. 4.1. Блок схема Таймера 1

Событие обновления (***Update event***) генерируется в следующих случаях:

- переполнение счетчика;
- программно, путем установки бита ***UG*** в регистре ***TIM*1_*EGR***;
- по событию на ***clock/trigger*** контроллере (рис. 4.1).

Когда предварительная загрузка разрешена (***ARPE*=1**), при генерации события, теневой регистр обновляется значением предварительной загрузки (***TIM1_ARR***), а также буфер предварительного делителя частоты обновляется значением из регистра ***TIM*1_*PSCR***.

Событие обновления (***Update event***) может быть запрещено, путем установления бита ***UDIS*** в регистре ***TIM1_CR*1**.

При запуске счётчика он тактируется с выхода предварительного делителя ***CK_CNT***, если бит ***CEN*** в регистре ***TIM*1_*CR*1** установлен.

Чтение и запись в 16-битный счетчик

Регистр ***TIM*1_*CNTR*** (*Timer 1 Counter*) является счетным регистром Таймера 1. Это 16-битный регистр, в котором хранится текущее значение Таймера 1. Доступ к этому регистру производится через регистры ***TIM*1_*CNTRH*** (*Counter high*) и ***TIM*1_*CNTRL*** (*Counter low*), причем, сначала, обращение (как в режиме чтения, так и в режиме записи) должно производиться к старшему регистру (***TIM*1_*CNTRH***), а потом –

к младшему (***TIM*1_*CNTRL***). Данное правило относится и к другим 16-битным регистрам микроконтроллера ***STM*8*S***.

Предварительный делитель частоты

Регистр ***TIM*1_*PSCR*** (***Timer* 1 *Prescaler Register***) является регистром предварительного делителя Таймера 1. Как и ***TIM*1_*CNTR***, это 16-битный регистр, обращение к которому происходит аналогичным образом, через регистры ***TIM*1_*PSCRH*** (*high byte*) и ***TIM*1_*PSCRL*** (*low byte*). Значение регистра ***TIM*1_*PSCR*** определяет величину, на которую делится входная частота, по формуле

$$f_{CK\_CNT} = \frac{f_{CK\_PCS}}{PSCR[15:0]+1},$$

где f_{CK_CNT} – частота, поступающая на таймер, f_{CK_PCS} – частота, поступающая на делитель. Новое значение делителя учитывается в следующем периоде (после очередного обновления счетчика событий).

Режим счета вверх

В данном режиме счета, таймер/счетчик считает от 0 до определенного пользователем значения (значение регистра ***TIM*1_*ARR***). Затем счет возобновляется с 0 и генерируется событие переполнение (***overflow event***) и событие обновления (если бит ***UDIS*** в регистре ***TIM*1_*CR*1** равен 0). На рис. 4.2 показан принцип счета вверх таймера.

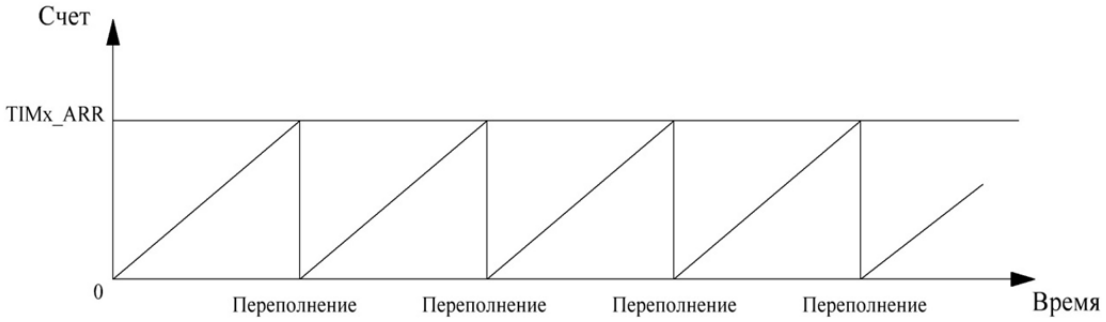

Рис. 4.2. Режим счета вверх таймера микроконтроллера

События обновления (***UEV***) также могут быть получены путем установки бита ***UG*** в регистре ***TIM*1_*EGR*** и могут быть запрещены программно, путем установления бита ***UDIS*** в регистре ***TIM*1_*CR*1**. Это позволит избежать обновления теневого регистра на этапе записи значения в предварительно загружаемый регистр. ***UEV*** отсутствует до тех пор, пока бит ***UDIS*** в регистре ***TIM*1_*CR*1** не сброшен. Если бит ***URS*** (запрос на обновление) в ***TIM*1_*CR*1** установлен, то установка бита ***UG*** генерирует ***UEV*** без установки флага ***UIF*** (регистр ***TIM*1_*SR*1**), соответственно, запроса на прерывание не будет.

Если происходит событие обновления, то все регистры обновляется, и флаг обновления (бит ***UIF*** в регистре ***TIM*1_*SR*1**) устанавливается (в зависимости от бита ***URS***):

- теневой регистр обновляется предварительно загруженным значением регистра ***TIM*1_*ARR***;
- буфер предварительного делителя частоты перезагружается значением регистра ***TIM*1_*PSCR***.

На рис. 4.3 и 4.4 показаны два примера работы счетчика с различной частотой, когда регистр ***TIM*1_*ARR*=0x36**.

На рис. 4.3 предварительный делитель установлен на 2, поэтому частота работы счетчика (***CK_CNT***) равна половине тактирующей частоты (***CK_PSC***) и предварительная загрузка запрещена (***ARPE=0***). Следовательно, теневой регистр немедленно изменяется и переполнение счетчика происходит, когда счет вверх достигает **0x36** и генерируется ***UEV***.

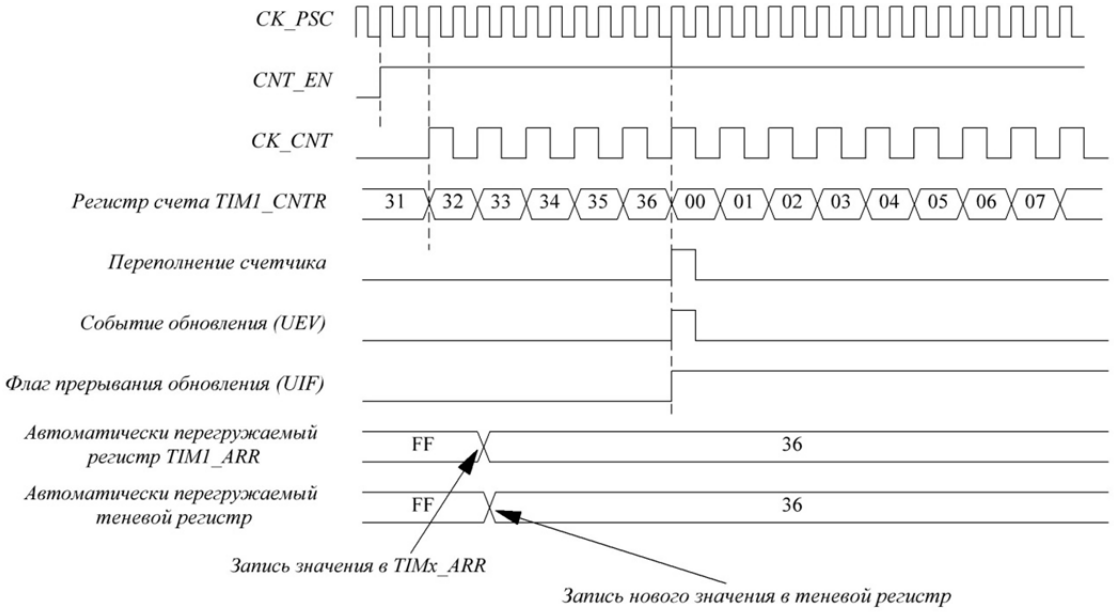

Рис. 4.3. Пример работы счетчика с предварительным делителем частоты равным 2 и запрещенной предварительной загрузкой

На рис. 4.4 показана последовательность действий таймера при условии, что предварительный делитель установлен в 1, поэтому частота работы счетчика (***CK_CNT***) равна тактирующей частоте (***CK_PSC***). В данном случае разрешена предварительная загрузка (***ARPE*=1**), поэтому следующее переполнение счетчика происходит, когда счет вверх достигает значение **0x36**. Новое значение автоматически перезагружаемого регистра равное **0x36** учитывается после переполнения, которое генерирует ***UEV***.

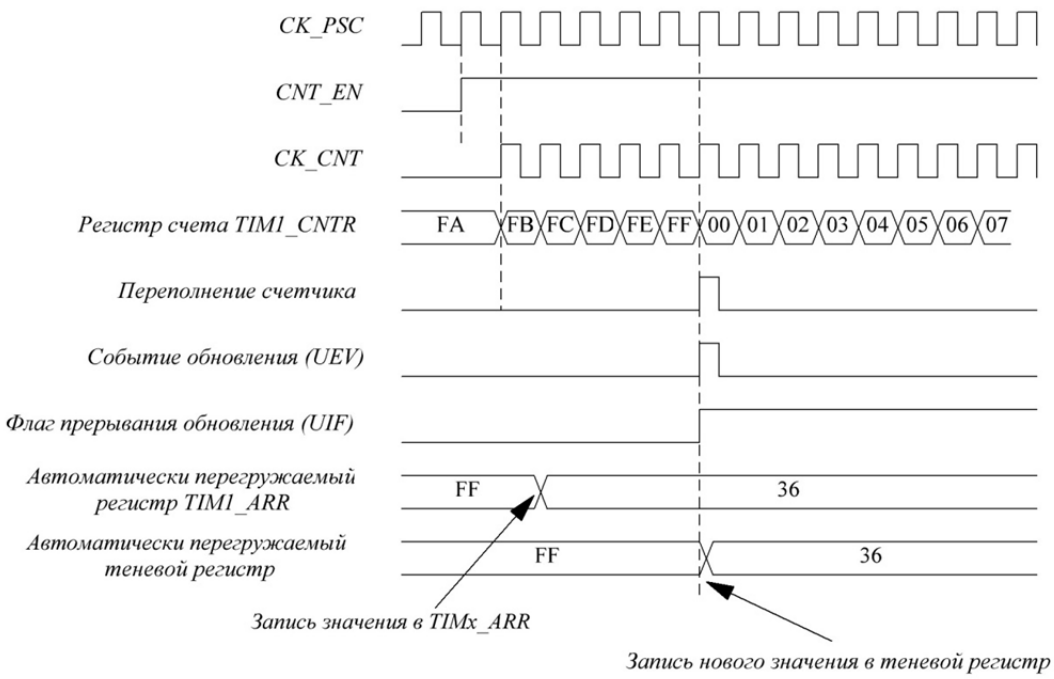

Рис. 4.4. Пример работы счетчика с предварительным делителем частоты равным 1 и разрешенной предварительной загрузкой

Режим счета вниз

В данном режиме счетчик считает с предустановленного значения (регистр ***TIM*1_*ARR***) до 0. Затем счет возобновляется с предустановленного значения и генерируется событие переполнения (***underflow event***) и ***UEV*** (если бит ***UDIS*** в регистре ***TIM*1_*CR*1** равен 0) (рис. 4.5).

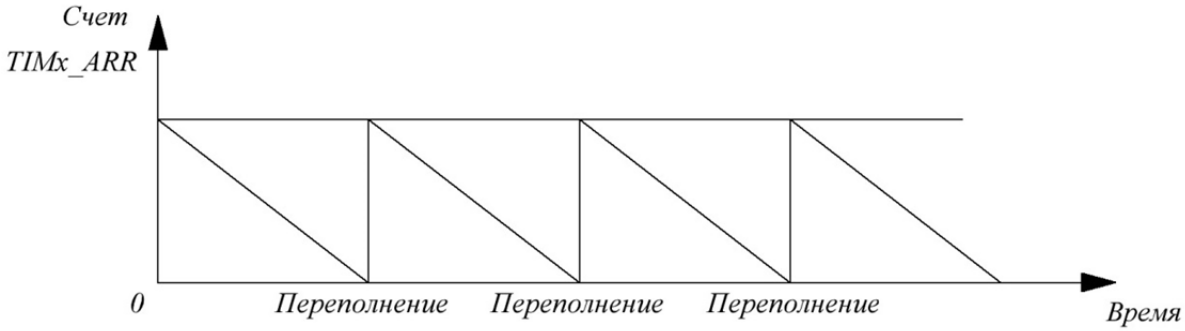

Рис. 4.5. Режим счета вниз таймера микроконтроллера

События обновления также могут быть получены путем установки бита ***UG*** в регистре ***TIM*1_*EGR***. ***UEV*** и может быть запрещено программно, путем установления бита ***UDIS*** в регистре ***TIM*1_*CR*1**. Это позволит избежать обновления теневого регистра, пока записывается

значение в регистр предварительной загрузки. ***UEV*** отсутствует до тех пор, пока бит ***UDIS*** в регистре ***TIM*1_*CR*1** не сброшен.

Если бит ***URS*** (запрос на обновление) в регистре ***TIM*1_*CR*1** установлен, установка бита ***UG*** генерирует ***UEV*** без установки флага ***UIF***. Соответственно, запроса на прерывание не будет. Это позволяет избежать генерацию события обновления и прерывания захвата, при очистке счетчика на событие захвата.

Когда происходит событие обновления, все регистры обновляются, и флаг обновления (***UIF*** бит в регистре ***TIM1_SR*1**) устанавливается (в зависимости от бита ***URS***):

- теневой регистр обновляется предварительно загруженным значением регистра ***TIM*1_*ARR***;
- буфер предварительного делителя частоты перезагружается предварительно загруженным значением регистра ***TIM*1_*PSCR***.

На рис. 4.6 и 4.7 показаны два примера работы таймера/счетчика с различной частотой, при условии, что регистр ***TIM*1_*ARR*=0x36**.

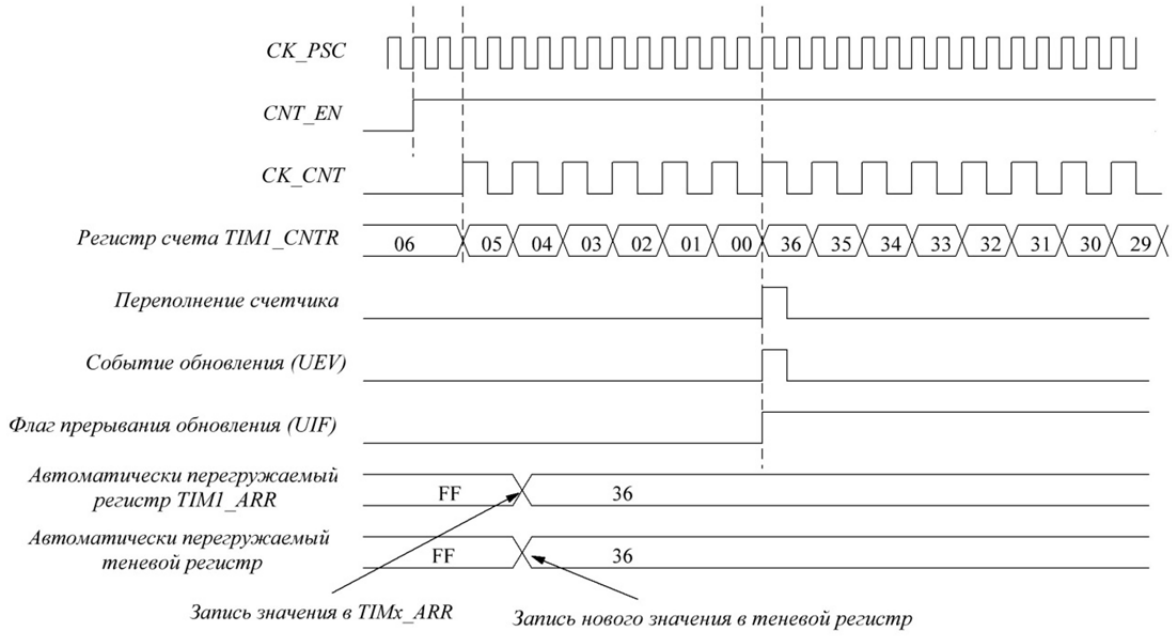

Рис. 4.6. Пример работы счетчика с предварительным делителем частоты равным 2 и запрещенной предварительной загрузкой

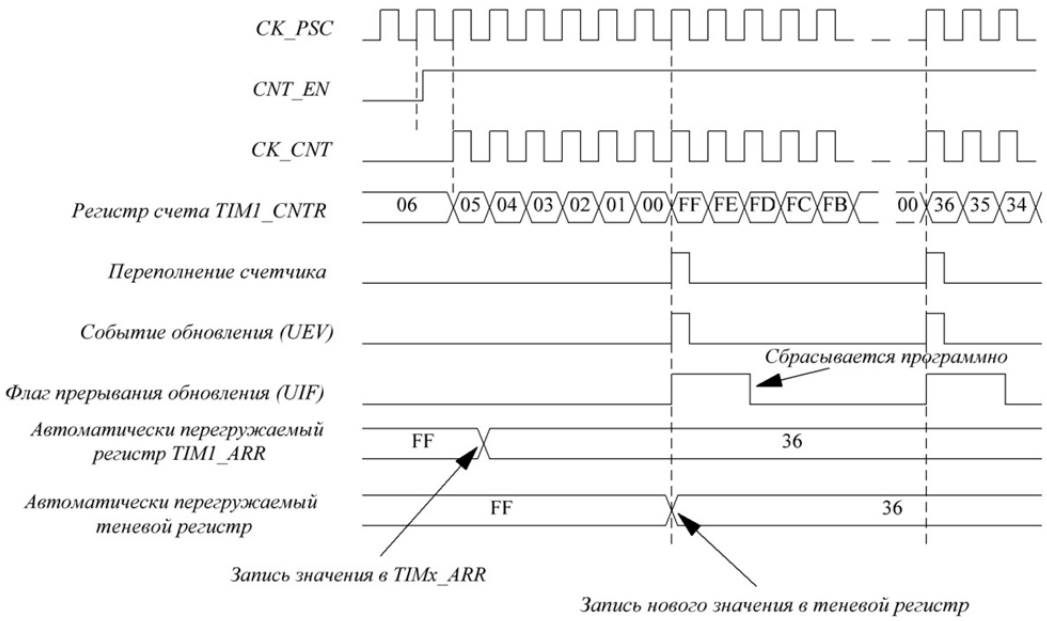

Рис. 4.7. Пример работы счетчика с предварительным делителем частоты равным 1 и разрешенной предварительной загрузкой

В режиме счета вниз, предварительная загрузка обычно не используется. Следовательно, новое значение принимается во внимание в следующем периоде.

Режим выравнивания по центру (счет вверх/вниз)

В данном режиме таймер/счетчик считает от 0 до предустановленного значения (содержание регистра ***TIM*1*_ARR***) минус 1. После этого генерируется событие переполнения сверху (***overflow event***). Далее счетчик считает до 0 и генерируется событие переполнения снизу (***underflow event***). После этого счетчик начинает считать с 0 и т.д. Если у таймера есть счетчик повторений, то ***UEV*** генерируется после каждого счета вверх/вниз и повторяется ***N*** раз. Число повторений задается с помощью регистра счетчика повторений (***repetition counter register TIM*1*_RCR***). Иначе ***UEV*** генерируется на каждое переполнение (***underflow*** и ***overflow***).

В режиме выравнивания по центру бит ***DIR*** (***direction bit*** – бит направления) в регистре ***TIM*1*_CR*1** не может быть записан. Он обновляется аппаратно и показывает направление счета.

На рис. 4.8 приведен пример работы счетчика в режиме счета вверх/вниз.

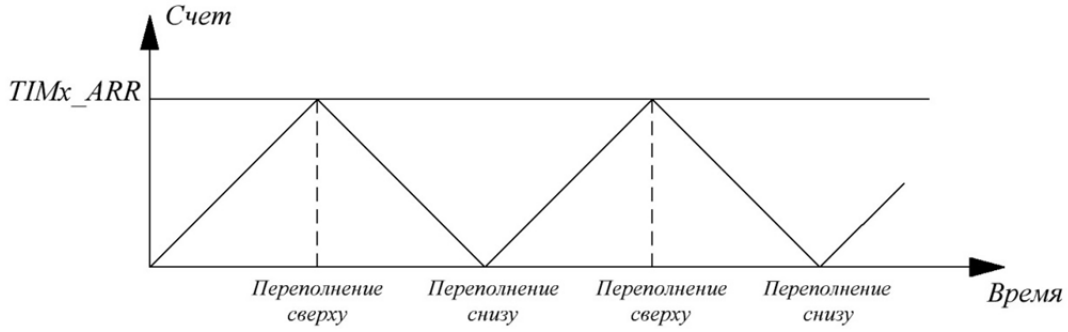

Рис. 4.8. Режим выравнивания по центру таймера микроконтроллера

На рис. 4.9 приведен пример поведения счетчика с различной частотой.

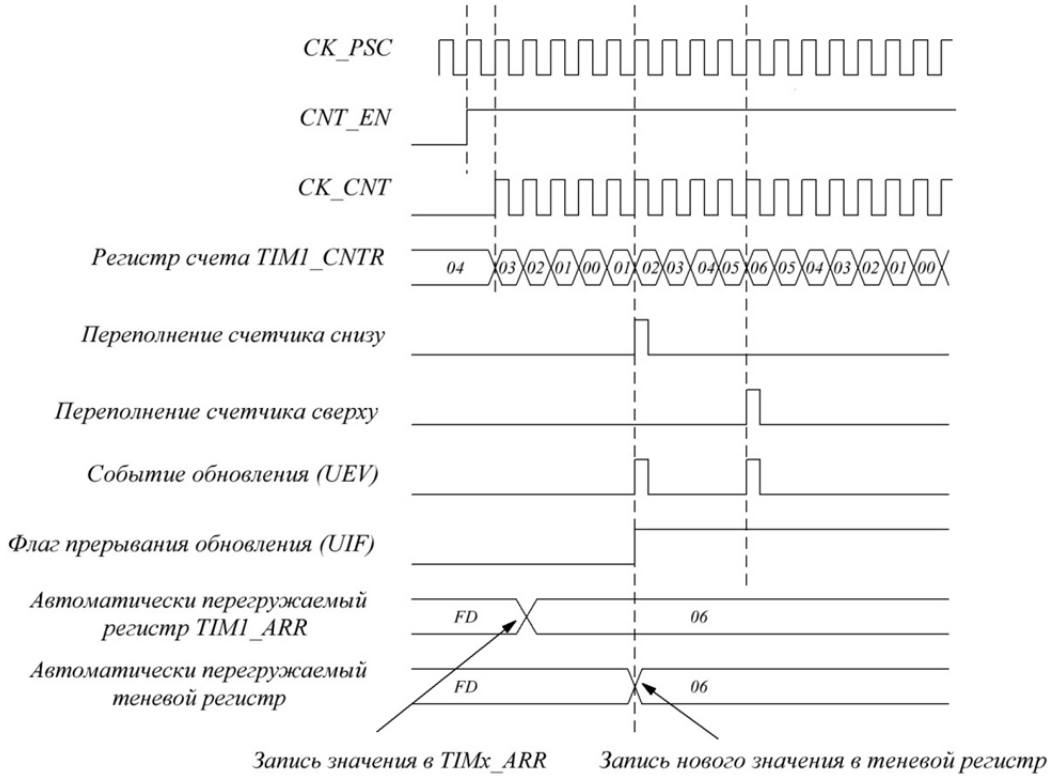

Рис. 4.9. Пример работы счетчика, при $f_{CK\_CNT} = f_{CK\_PSC}$, ***TIM*1_*ARR*=06*h*, *ARPE*=1.

Представим ряд советов по использованию режима выравнивания по центру:

- В режиме выравнивания по центру используется настройка направления счета (вверх/вниз). Это означает, что счетчик начинает счет вверх или вниз в зависимости от уровня, записанного в бите ***DIR*** в

регистре ***TIM*1_*CR*1**. Причем, бит ***DIR*** и ***CMS*** не могут быть изменены одновременно.

• Запись в счетчик во время работы в режиме выравнивания по центру не рекомендуется, так как это может привести к неожиданным результатам. В частности:

✓ направление не обновится, если значение, записанное в счетчик, больше автоматически перезагружаемого (***TIM*1_*CNT*>*TIM*1_*ARR***). Например, если счетчик считает вверх, то он продолжит делать это;

✓ направление обновляется, если 0 или значение ***TIM*1_*ARR*** записаны в счетчик, но ***UEV*** не сгенерировано.

• Безопасный способ использования режима выравнивания по центру – это генерирование ***UEV*** программно (устанавливая бит ***UG*** в регистре ***TIM*1_*EGR***), перед запуском счетчика.

Режим повтора счета вниз

В режиме повтора вниз ***UEV*** генерируется только при достижении счетчиком 0. Это может быть полезно при генерации сигналов широтно-импульсной модуляции. При этом данные переносятся из предварительно загружаемого регистра в теневой регистр после каждого «***n***» переполнения (***overflow/underflow***), где ***n*** это число повторений в регистре повторений счетчика ***TIM*1_*RCR***.

Число повторений счета вниз уменьшается:

• С каждым переполнением (***overflow***) в режиме счета вверх.

• С каждым переполнением (***underflow***) в режиме счета вниз.

• С каждым переполнением (***overflow/ underflow***) в режиме выравнивания по центру. Хотя это ограничивает максимальное количество повторений до 128 циклов ШИМ.

Число повторений определяется в регистре ***TIM*1_*RCR*** (как показано на рис. 4.10). Когда ***UEV*** генерируется программно (установка бита ***UG*** в регистре ***TIM*1_*EGR***) или аппаратно, оно происходит сразу же, независимо от значения повторения. Повторение перезагружается значением регистра ***TIM*1_*RCR***.

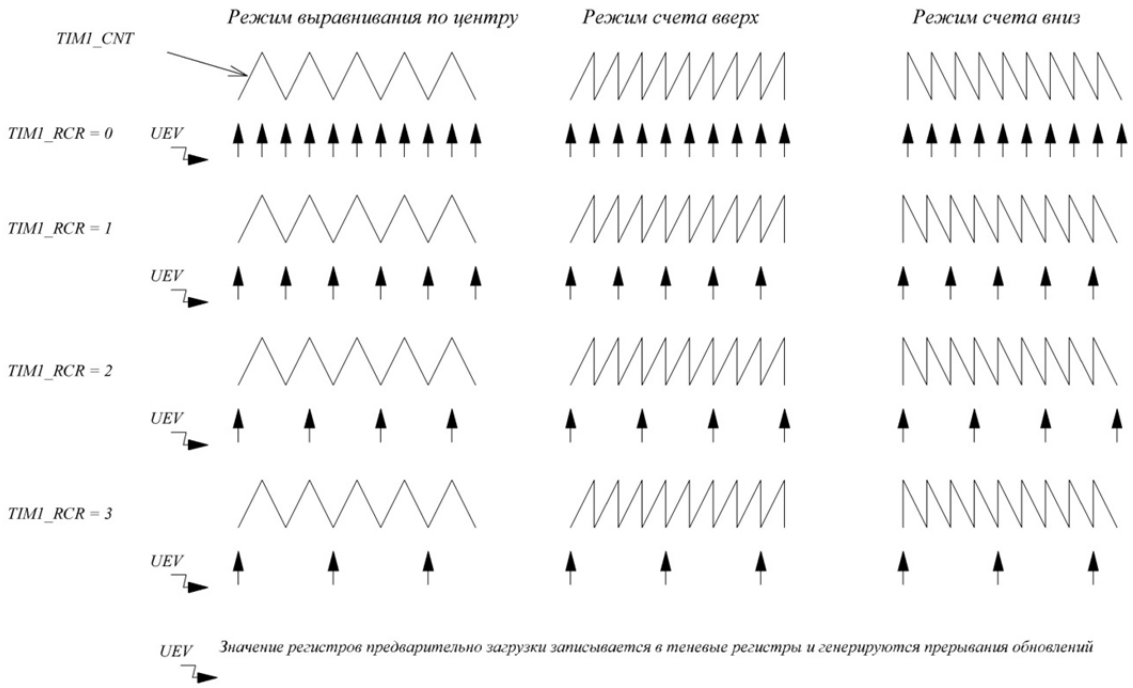

*Рис. 4.10. Пример частоты обновления в зависимости от режима и конфигурации регистра **TIM1_RCR**.*

Clock/trigger контроллер

Clock/trigger контроллер позволяет настраивать источник тактирования таймера, входы и выходы запуска**..**

В качестве источника тактирования могут применяться:

- внутренний источник тактирования (***f**MASTER*)*;
- внешний источник тактирования, режим 1: внешний вход таймера (***TIx***);
- внешний источник тактирования, режим 2: внешний запускающий вход (***ETR***);
- внутренние входы запуска (***ITRi***): использование одного таймера, в качестве предварительного делителя для другого таймера.

Внутренний источник тактирования (*f**MASTER*)

Если ***clock/trigger*** контроллер и вход внешнего запуска запрещены (***SMS***=000 в регистре ***TIM1_SMCR*** и ***ECE***=0 в регистре ***TIM1_ETR***), биты ***CEN***, ***DIR*** (***TIM1_CR1***) и ***UG*** (***TIM1_EGR***) ведут себя как управляющие биты и могут быть изменены только программно (кроме ***UG***, который обнуляется автоматически). Как только бит ***CEN*** устанавливается, делитель синхронизируется с внутренним источником тактирования (***Internal clock***) (рис. 4.11).

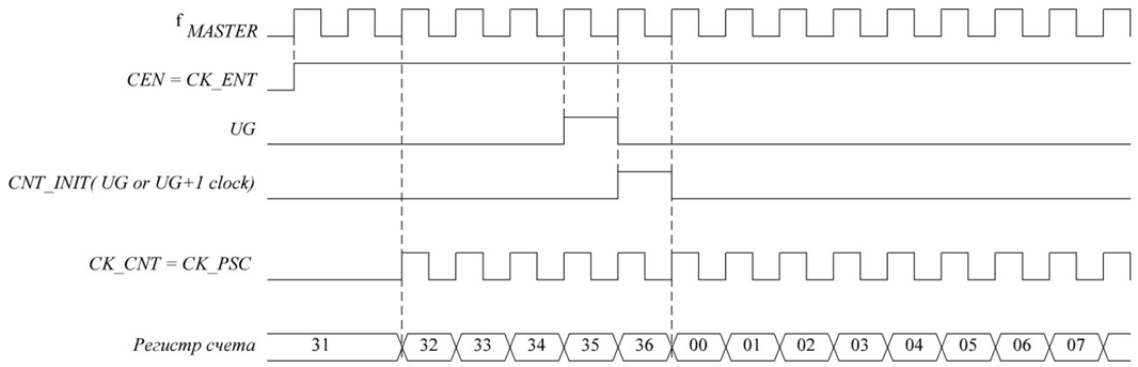

Рис. 4.11. Пример работы таймера при внутреннем источнике тактирования $f_{CK\_PSC}=f_{VASTER}$.

Внешний источник тактирования, режим 1

Счетчик может считать при поступлении импульса на выбранный вход таймера по переднему или заднему фронту. Данный режим выбирается, когда биты ***SMS***=111 в регистре ***TIM*1_*SMCR***.

Для настройки счета вверх и, например, по переднему фронту сигнала на входе ***TI*2** необходимо используйть следующий алгоритм:

1) настройте канал 2 для обнаружения нарастающих фронтов на входе ***TI*2**, записав в биты ***CC*2*S***=01 регистра ***TIM*1_*CCMR*2**;

2) настройте входной фильтр длительности, задав значение битов ***IC*2*F*** [3:0] в регистре ***TIM*1_*CCMR*2** (Если фильтр не нужен, биты необходимо оставлять равными 0);

3) задайте чувствительность к переднему фронту сигнала, сбросив бит ***CC*2*P*** в регистре ***TIM*1_*CCER*1**;

4) настройте таймер на тактирование от внешнего источника, режим 1, задав биты ***SMS***=111 в регистре ***TIM*1_*SMCR***;

5) выберите вход ***TI*2**, задав биты ***TS***=110 в регистре ***TIM*1_*SMCR***;

6) запустите таймер, установкой бита ***CEN*** в регистре ***TIM*1_*CR*1**.

При поступлении переднего фронта сигнала на вход ***TI*2**, счетчик считает один раз и устанавливается флаг запуска (бит ***TIF*** в регистре ***TIM1_SR*1**) и будет послан запрос на прерывание, если прерывания разрешены (зависит от бита ***TIE*** в регистре ***TIM*1_*IER***). Задержка между передним фронтом сигнала на входе ***TI*2** и сигналом со счетчика обусловлена ресинхронизацией (сопряжением двух цифровых систем, имеющих различные структуры циклов и независимые скорости передачи) на входе ***TI*2** (рис. 4.12).

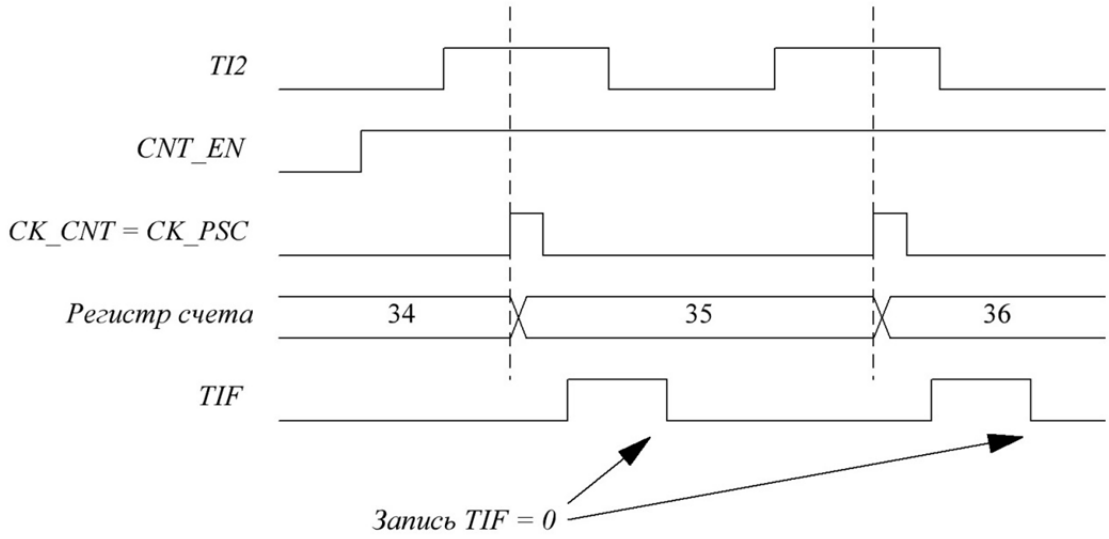

Рис. 4.12. Пример работы таймера при работе схемы управления в режиме внешнего источника тактирования 1

Внешний источник тактирования, режим 2

Счетчик может считать при поступлении импульса на ***ETR*** (***external trigger)*** по переднему или заднему фронту. Данный режим выбирается, когда биты ***ECE***=111 в регистре ***TIM*1_*ETR***.

Для настройки счета вверх и, например, чтобы считать один раз на каждые два передних фронта сигнала на ***ETR*** используйте следующий алгоритм:

1) если не требуется фильтр, то биты ***ETF***=0000 в регистре ***TIM*1_*ETR***;

2) настройте предварительный делитель, установив биты ***ETPS***[1:0]=01 в регистре ***TIM*1_*ETR***;

3) задайте чувствительность к переднему фронту сигнала на ***ETR***, сбросив бит ***ETP*** в регистре ***TIM*1_ *ETR***;

4) разрешите внешний источник тактирования, в режиме 2, установив биты ***ECE*** в регистре ***TIM*1_*ETR***;

5) запустите таймер, установкой бита ***CEN*** в регистре ***TIM*1_*CR*1**.

Счетчик считает один раз на каждые два передних фронта сигнала на ***ETR*** (рис. 4.13). Задержка между передним фронтом сигнала на входе ***ETR*** и сигналом со счетчика обусловлена ресинхронизацией на входе ***ETR*** (***ETRP***).

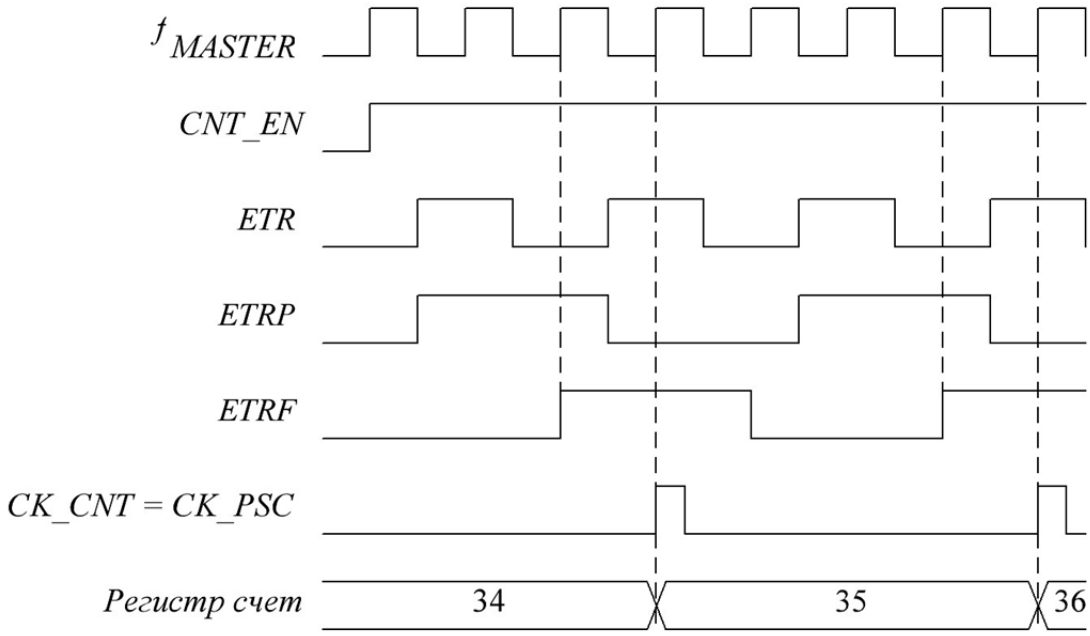

Рис. 4.13. Пример работы таймера при работе схемы управления в режиме внешнего источника тактирования 2

Синхронизация запуска

Существуют четыре входа запуска:

- ***ETR***;
- ***TI*1**;
- ***TI*2**;
- ***TRGO*** с таймеров ***TIM*5**/***TIM*6**.

Таймер 1 может быть синхронизован с внешним запуском в трех режимах: стандартный режим запуска, запуск в режиме сброса и запуск в закрытом режиме.

В стандартном режиме запуска счетчик может начать счет в ответ на событие на выбранном входе.

Для настройки счета вверх и, например, по переднему фронту сигнала на входе ***TI*2** используйте следующий алгоритм:

1) Настройте канал 2 на работу по переднему фронту сигнала на входе ***TI*2** (бит ***CC*2*P***=0 в регистре ***TIM*1_*CCER*1**). Так как в данном примере фильтр не требуется, настройте входной фильтр на продолжительность 0 (***IC*2*F***=0000). Предварительный делитель захвата не используется для запуска и поэтому не подлежит настройке. Биты ***CC*2*S*** выбирают источник входа захвата и также не настраиваются.

2) Настройте таймер в стандартном режиме запуска записав в биты ***SMS***=110 в регистре ***TIM*1_*SMCR***. В качестве источника выберите вход ***TI*2**, записав в биты ***TS***=110 в регистре ***TIM*1_*SMCR***.

Когда на входе ***TI2*** появится передний фронт сигнала, счетчик начнет счет по внешнему источнику тактовых импульсов и установится флаг ***TIF*** (рис. 4.14). Задержка между передним фронтом сигнала на входе ***TI2*** и сигналом со счетчика обусловлена ресинхронизацией на входе ***TI2***.

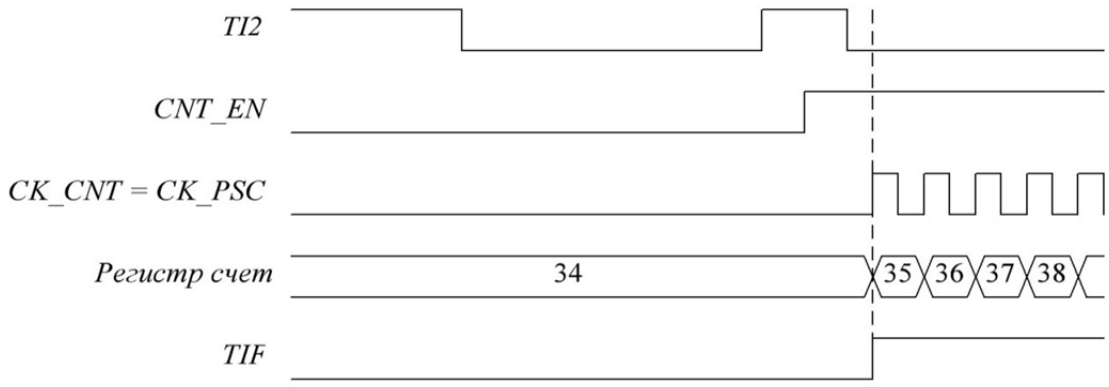

Рис. 4.14. Пример работы таймера при работе схемы управления в режиме запуска

При работе в режиме сброса счетчик и его предварительный делитель могут быть повторно инициализированы, в ответ на событие на входе запуска. Кроме того, если бит ***URS*** в регистре ***TIM1_CR1*** сброшен, генерируется ***UEV***. Затем все предварительно загружаемые регистры (***TIM1_ARR***, ***TIM1_CCRi***) обновляются.

Для настройки счета вверх и, например, по переднему фронту сигнала на входе ***TI1*** используйте следующий алгоритм:

1) Настройте канал 1 на работу по переднему фронту сигнала (бит ***CC1P***=0 в регистре ***TIM1_CCER1***) на входе ***TI1***. Так как в данном примере фильтр не требуется, настройте входной фильтр на продолжительность 0 (***IC2F***=0000). Предварительный делитель захвата не используется для запуска и поэтому не подлежит настройки. Биты ***CC1S*** выбирают источник входного захвата и также не настраиваются.

2) Настройте таймер в режиме запуска сброса записав в биты ***SMS***=100 в регистре ***TIM1_SMCR***. В качестве источника выберите вход ***TI1***, записав в биты ***TS***=101 в регистре ***TIM1_SMCR***.

3) Запустите таймер, установкой бита ***CEN*** в регистре ***TIM1_CR1***.

Таймер считает от внешнего источника тактирующих импульсов, до поступления переднего фронта сигнала на входе ***TI1***. Как только появляется сигнал на ***TI1***, таймер сбрасывается и начинает счет с нуля. В тоже время устанавливается флаг запуска (бит ***TIF*** в регистре ***TIM1_SR1***) и, если прерывание разрешено (зависит от бита ***TIE*** в регистре ***TIM1_IER***), будет сформирован запрос на прерывание.

Рис. 4.15 демонстрирует работу таймера, когда в автоперезагружаемом регистре ***TIM*1_*ARR***=36*h*. Задержка между передним фронтом сигнала на входе ***TI*1** и сигналом со счетчика обусловлена ресинхронизацией на входе ***TI*1**.

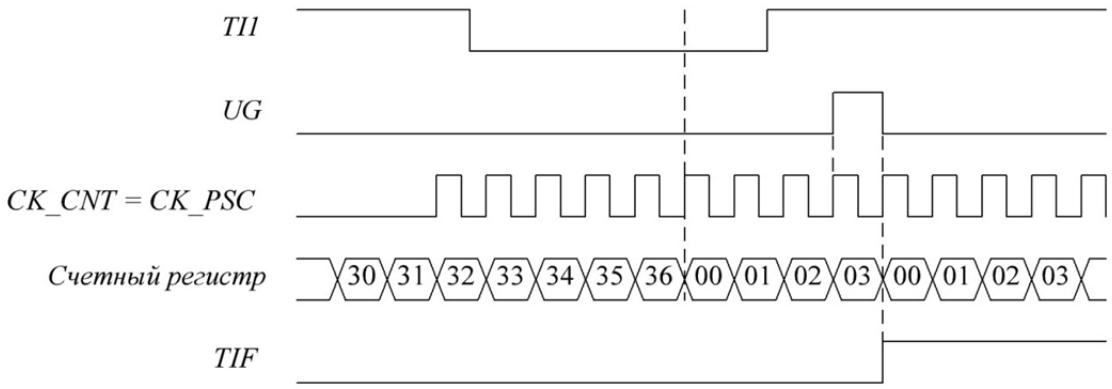

Рис. 4.15. Пример работы таймера при работе схемы управления в режиме запуска сброса

При работе в закрытом режиме счетчик может быть включен, в зависимости от уровня сигнала на выбранном входе.

Для настройки счета вверх при низком уровне сигнала на входе ***TI*1** используйте следующий алгоритм:

1) Настройте канал 1 на работу по низкому уровню сигнала (бит ***CC*1*P***=1 в регистре ***TIM*1_*CCER*1**) на входе ***TI*1**. Так как в данном примере фильтр не требуется, настройте входной фильтр на продолжительность 0 (***IC*2*F***=0000). Предварительный делитель захвата не используется для запуска и поэтому не подлежит настройки. Биты ***CC*1*S*** выбирают источник входного захвата и также не настраиваются.

2) Настройте таймер в закрытом режиме запуска записав в биты ***SMS***=101 в регистре ***TIM*1_*SMCR***. В качестве источника выберите вход ***TI1***, записав в биты ***TS***=101 в регистре ***TIM*1_*SMCR***.

3) Запустите таймер, установкой бита ***CEN*** в регистре ***TIM*1_*CR*1**(в закрытом режиме запуска счетчик не начнет счет, если бит ***CEN***=0, вне зависимости от уровня сигнала на входе запуска).

Таймер работает от внутреннего источника тактирования, до тех пор, пока на входе ***TI*1** низкий уровень сигнала. Как только на входе ***TI*1** появляется высокий уровень сигнала, таймер останавливается. Флаг ***TIF*** устанавливается каждый раз, когда таймер начинает или прекращает работать (рис. 4.16). Задержка между передним фронтом сигнала на входе ***TI*1** и сигналом со счетчика обусловлена ресинхронизацией на входе ***TI*1**.

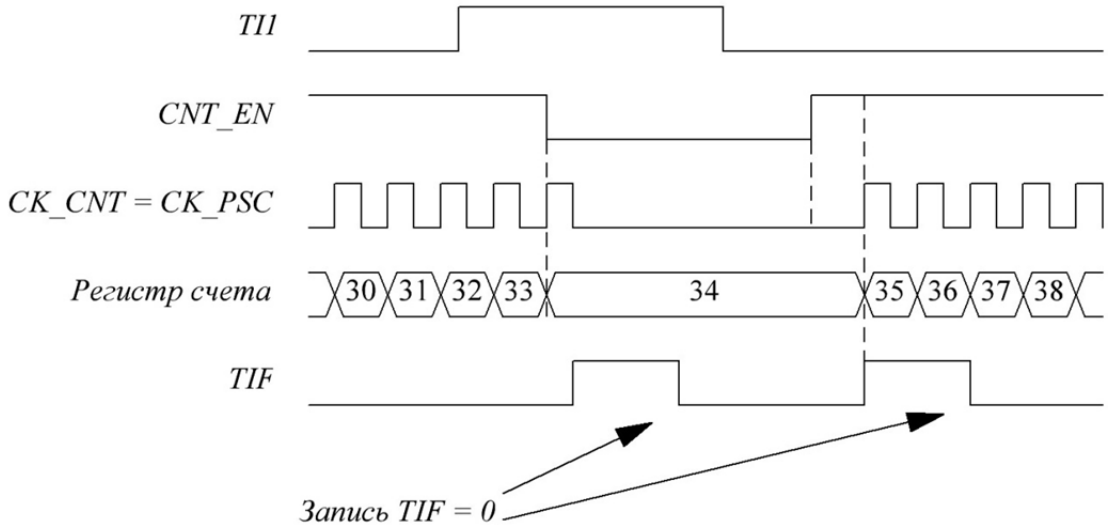

Рис. 4.16. Пример работы таймера при работе схемы управления в режиме закрытого запуска.

Синхронизация между таймерами (*TIM*1, *TIM*5 и *TIM*6).

В некоторых микроконтроллерах таймеры связаны вместе внутри для синхронизации таймеров или соединения их последовательно. Когда один из таймеров настроен в режиме ведущего, он может служить выходом запуска (***TRGO***), тем самым запускать, останавливать, перезапускать или тактировать любой другой таймер, который настроен на режим ведомого (рис. 4.17).

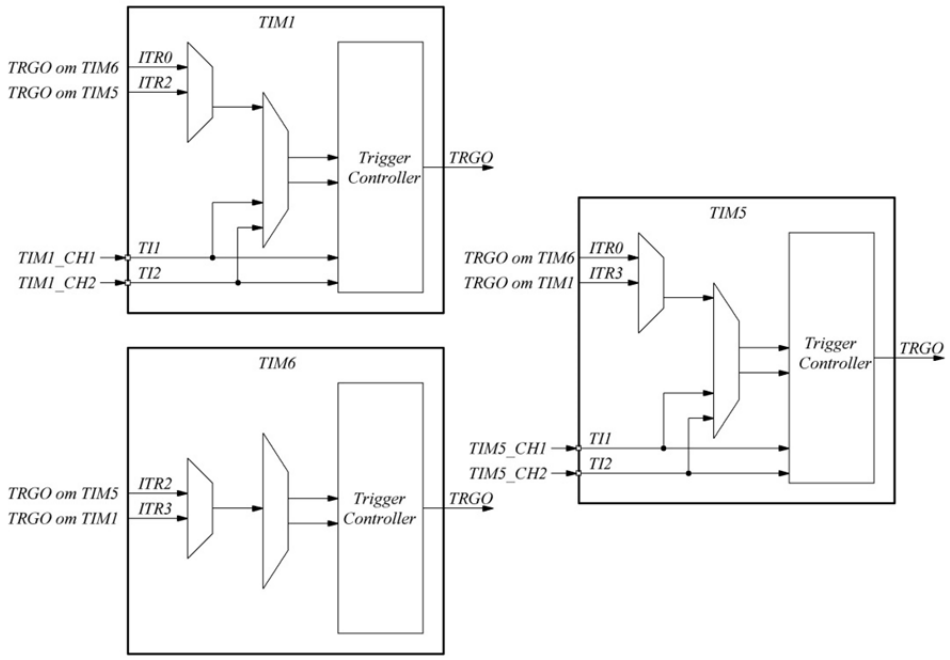

Рис. 4.17. Пример реализации системы последовательной работы таймеров

Также таймера можно использовать в качестве предварительного делителя для другого таймера.

На рис. 4.18 показано, как таймер ***A*** может быть использован в качестве предварительного делителя для таймера ***B***.

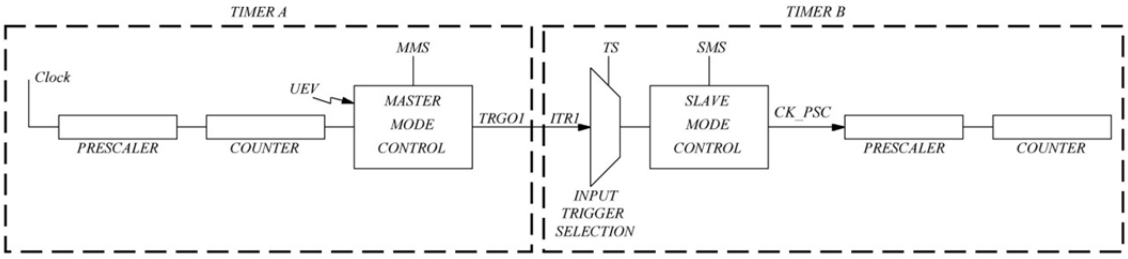

Рис. 4.18. Пример использования одного таймера в качестве предварительного делителя другого

Приведем пример настройки таймеров:

1) Настройте таймер ***A***, как ведущий таймер, чтобы он вырабатывал периодический сигнал запуска на каждое ***UEV***. Для того чтобы на выходе ***TRGO*1**, при каждой генерации ***UEV***, возникал передний фронт сигнала, запишите в биты ***MMS***=010 в регистре ***TIMx_CR*2**.

2) Соедините выход ***TRGO*1** таймера ***A*** с таймером ***B***, таймер ***B*** должен быть в режиме ведомого, использующего вход запуска ***ITR*1**. Выбрать это можно, записав в биты ***TS***=001 в регистре ***TIMx_SMCR*** (x-номер таймера).

3) Настройте ***clock/trigger*** контроллер на внешний источник тактирования, в режиме 1, записав в биты ***SMS***=111 в регистре ***TIMx_SMCR***. Это позволяет тактировать таймер ***B***, по переднему фронту сигнала запуска с таймера ***A*** (который соответствует переполнению счетчика таймера ***A***).

4) Запустите оба таймера установив биты ***CEN*** в соответствующих регистрах ***TIMx_CR*1**.

Режим ШИМ

Режим широтно-импульсной модуляции позволяет генерировать импульсный сигнал с частотой, определяемой значением регистра ***TIM*1_*ARR***, а коэффициент заполнения определяется величиной регистров ***TIM*1_*CCRi***.

Режим ШИМ может быть выбран, независимо для каждого канала (один ШИМ на выход ***OCI***), записав 110 (режим ШИМ 1) или 111 (режим ШИМ 2) в биты ***OCiM*** регистров ***TIM*1_*CCMRi***. Соответствующий регистр предварительной загрузки должен быть включен путем установки ***OCiPE*** бита регистров ***TIM*1_*CCMRi***. Автоматически загружаемый регистр предварительной загрузки (в режиме счета вверх или вы-

равнивания по центру) можно при желании подключить установкой бита ***ARPE*** в регистре ***TIM*1_*CR*1**.

Так как значения регистров предварительной загрузки копируются в теневые регистры только при возникновении ***UEV***, все регистры должны быть инициализированы путем установки бита ***UG*** в регистре ***TIM*1_*EGR*** перед запуском счетчика.

Полярность ***OCi*** изменяется программно, используя биты ***CCiP*** в регистрах ***TIM*1_*CCERi***. Они могут быть запрограммированы на активный высокий уровень или активный низкий уровень. Выход ***OCi*** разрешается комбинацией битов ***CCiE***, ***MOE***, ***OISi***, ***OSSR*** и ***OSS*** (в регистрах ***TIM*1_*CCERi*** и ***TIM*1_*BKR***).

В режиме ШИМ (1 или 2), ***TIM*1_*CNT*** и ***TIM*1_*CCRi*** всегда сравниваются, чтобы определить, является ли ***TIM*1_*CCRi*** ≤ ***TIM*1_*CNT*** или ***TIM*1_*CNT*** ≤ ***TIM*1_*CCRi*** (в зависимости от направления счетчика).

Таймер способен генерировать ШИМ в режиме выравнивания по краю или выравниванию по центру в зависимости от битов ***CMS*** в реестре ***TIM*1_*CR*1**.

Режим счета вверх активен, когда бит ***DIR*** в регистре ***TIM*1_*CR*1** сброшен.

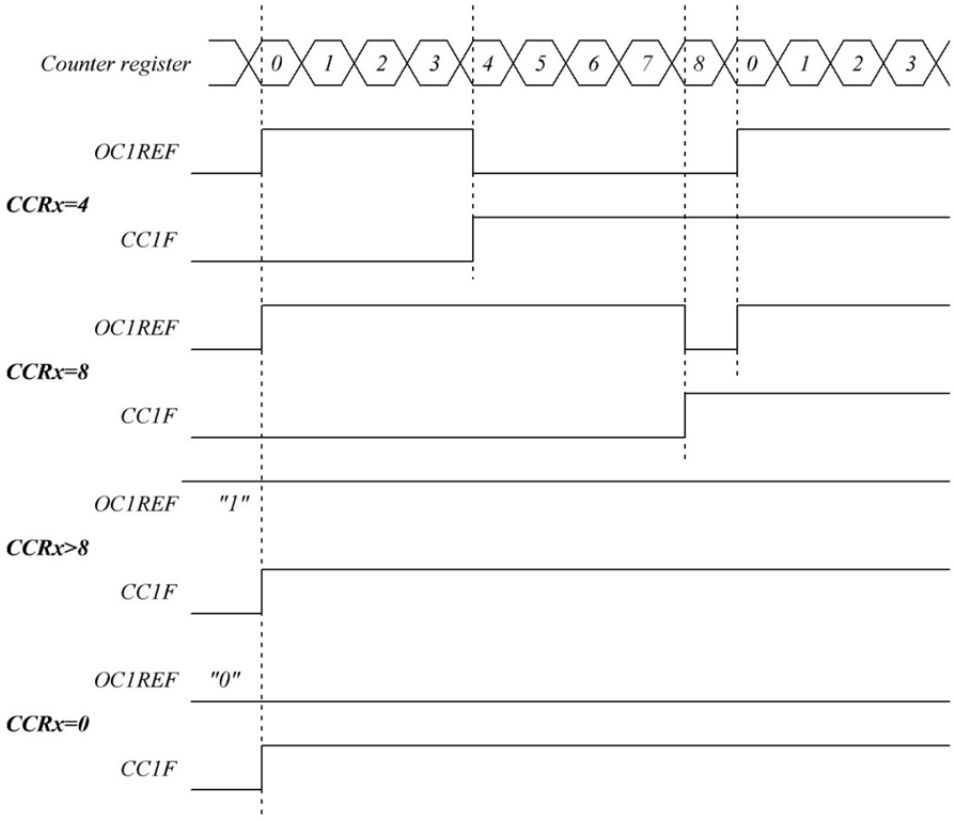

*Рис. 4.19. ШИМ (режим 1) в режиме выравнивания по краю (**ARR**=8)*

Пример настройки ШИМ при работе в режиме счета вверх:

В этом примере используется режим ШИМ 1. Опорный сигнал ШИМ, ***OCiREF,*** сигнал высокого уровня, пока ***TIM*1_*CNT*** <***TIM*1_*CCRi***. В противном случае, он становится сигналом с низким уровнем. Если сравниваемое значение в ***TIM*1_*CCRi*** больше, чем значение авто загружаемого регистра (***TIM*1_*ARR***), то ***OCiREF*** равен 1. Если сравниваемое значение равно 0, ***OCiREF*** равен 0. Рис. 4.19 показывает несколько примеров сигналов ШИМ в режиме выравнивания по краю, где ***TIM*1_*ARR***=8.

Режим счета вниз активен, когда бит ***DIR*** в реестре ***TIM*1_*CR*1** установлен.

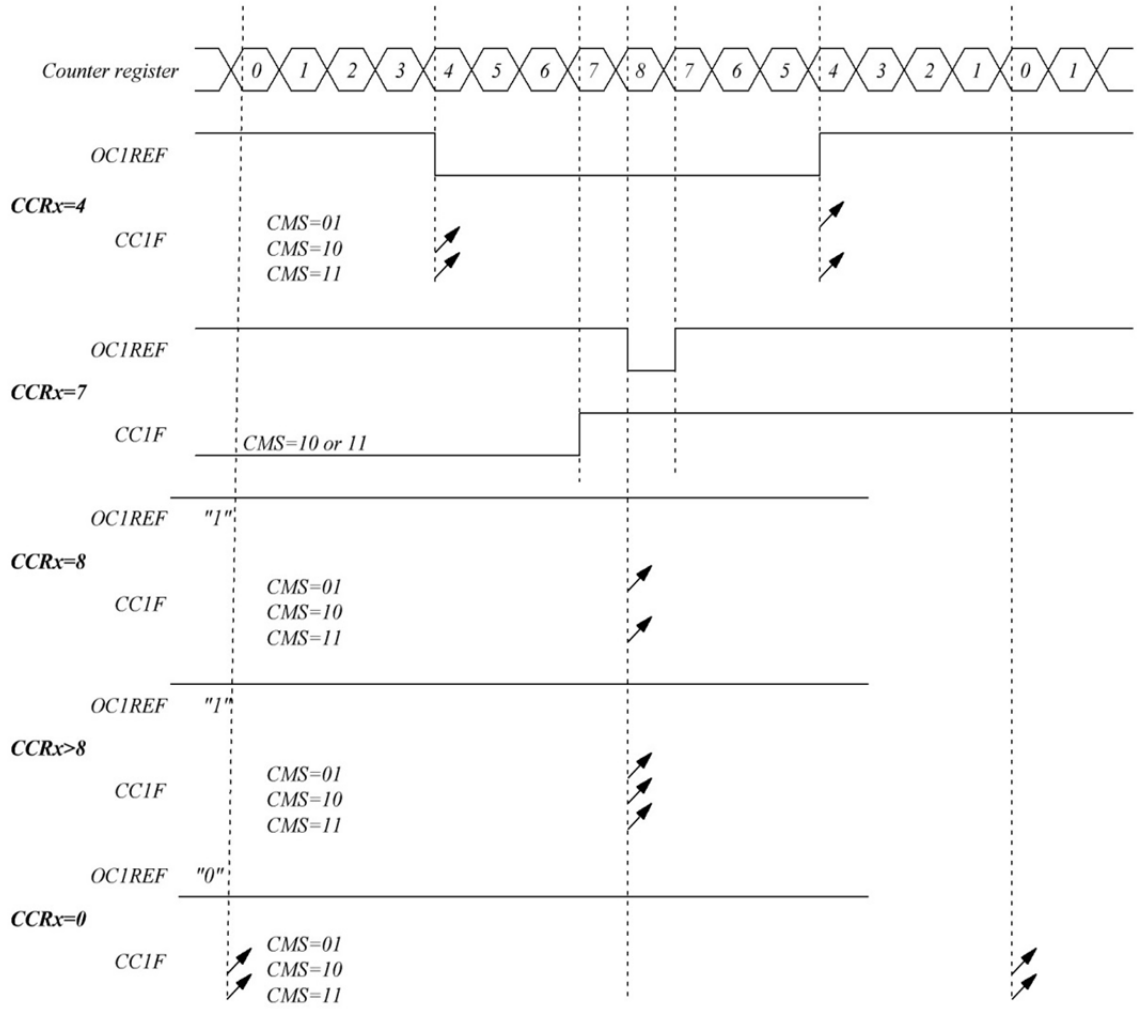

*Рис. 4.20. Пример режима ШИМ с выравниванием по центру (**ARR**=8)*

В режиме ШИМ 1, опорный сигнал ***OCiREF*** низкого уровня, пока ***TIM*1_*CNT*> ***TIM*1_*CCRi***. В противном случае он становится сигналом высокого уровня. Если сравниваемое значение в ***TIM*1_*CCRi***

больше, чем значение в автоматически перезагружаемом регистре ***TIM*1_*ARR***, ***OCiREF*** равен 1. Нулевой сигнал ШИМ не возможен в этом режиме.

Режим выравнивания по центру активен, когда биты ***CMS*** в регистре ***TIM*1_*CR*1** отличаются от 00 (все остальные конфигурации имеют такое же влияние на сигналы ***OCiREF***/***OCI***).

Флаг сравнения устанавливается, когда счетчик считает вверх, вниз, или вверх и вниз, в зависимости от конфигурации битов ***CMS***. Бит направления (***DIR***) в регистре ***TIM*1_*CR*1** в этом режиме обновляется аппаратно и доступен только для чтения.

На рис. 4.20 показан пример работы ШИМ в режиме выравнивания по центру сигналов, где:

- ***TIM1_ARR***=8;
- Режим ШИМ – 1;
- Флаг устанавливается в трех различных случаях:
  - Счетчик считает вниз (***CMS***=01);
  - Счетчик считает вверх (***CMS***=10);
  - Счетчик считает вверх и вниз (***CMS***=11).

Режим одиночного импульса

Режим одиночного импульса (***OPM***) является частным случаем предыдущих режимов. В этом режиме счетчик может быть запущен внешним импульсом и генерировать импульс с программируемой длиной после программируемой задержки.

Запуском счетчика можно управлять с помощью ***clock***/***trigger*** контроллера. Генерация сигнала может быть в режиме выхода или в режиме сравнения ШИМ. Для выбора данного режима работы необходимо установить бит ***OPM*** в регистре ***TIM*1_*CR*1**. Импульс будет генерироваться правильно, только если сравниваемое значение отличается от начального значения счетчика. Перед началом (когда таймер ждет запуска), должна быть выполнена следующая настройка:

- В режиме счета вверх: $CNT<CCRi\leq ARR$ (в частности, $0<CCRi$);
- В режиме счета вниз: $CNT>CCRi$.

На рис. 4.21 показана диаграмма работы счетчика в режиме одиночного импульса.

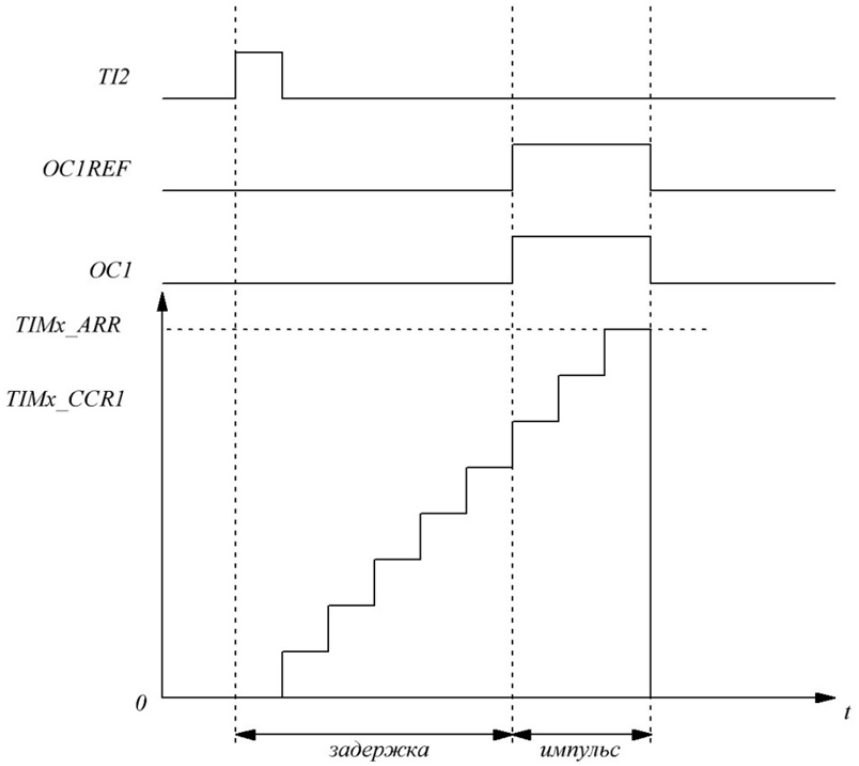

Рис. 4.21. Пример реализации режима одиночного импульса

Приведем пример, который показывает, как генерировать положительный импульс на ***OC*1** с длиной t_{PULSE} и после задержки t_{DELAY}, как только положительный фронт появляется на входном контакте ***TI*2**.

Выполните следующие действия, чтобы использовать ***IC*2**, в качестве однократного запуска:

- Установите ***IC*2** на вход ***TI*2**, записав в биты ***CC*2*S***=01 в регистре ***TIM*1*_CCMR*2**.
- ***IC*2** должен работать по переднему фронту сигнала. Для этого необходимо записать в бит ***CC*2*P***=0 в регистре ***TIM*1*_CCER*1**;
- Настройте ***IC*2**, в качестве запуска для ***clock*/*trigger*** контроллера, записав в биты ***TS***=110 в регистре ***TIM*1*_SMCR***;
- ***IC*2** используется для запуска счетчика, если в биты ***SMS*** записать 110 в регистре ***TIM*1*_SMCR***.

Форма сигнала в режиме ***OPM*** определяется, путем записи в регистры сравнения (с учетом тактовой частоты и счетчик делителя), следующим образом:

- t_{DELAY} (время задержки) определяется значением записанным в регистре ***TIM*1*_CCR*1**;

• t_{PULSE} (время импульса) определяется разницей между автоматически перезагружаемым значением и сравниваемым значением (***TIM*1_*ARR* – *TIM*1_*CCR*1**);

• Для создания сигнала, с переходом из 0 в 1, после того как произошло сравнение, и из 1 в 0, по достижению счетчиком предварительно загружаемой величины, необходимо разрешить ШИМ в режиме 2, записав в биты ***OCiM***=111 в регистре ***TIM_CCMR*1**. Существует и альтернативный вариант, для этого, необходимо разрешить регистр предварительной загрузки, установив бит ***OC*1*PE*** в регистре ***TIM*1_*CCMR*1** и сбросив бит ***ARPE*** в регистре ***TIM*1_*CR*1**. В данном случае запишите сравниваемое значение в регистр ***TIM*1_*CCR*1** и предварительно загружаемое значение запишите в регистр ***TIM*1_*ARR***. Затем, сгенерируйте обновление, установив бит ***UG***, и ждите события внешнего запуска на ***TI*2**. В данном примере бит ***CC*1*P*** сброшен.

В примере, приведенном выше, биты ***DIR*** и ***CMS***, в регистре ***TIM*1_*CR*1**, должны быть сброшены.

4.2. Регистры таймера 1

Регистр контроля 1

Регистр контроля 1 **(*TIM*1_*CR*1).**

7	6	5	4	3	2	1	0
ARPE	***CMS***[1:0]		***DIR***	***OPM***	***URS***	***UDIS***	***CEN***
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7 ***ARPE***:

0: регистр ***TIM*1_*ARR*** может быть записан непосредственно;

1: регистр ***TIM*1_*ARR*** записывается через предварительно загружаемый регистр.

• Биты 6 и 5 ***CMS***:

00: счетчик считает вверх или вниз в зависимости от бита направления (***DIR***);

01: счетчик считает вверх и вниз поочередно. Флаги прерывания устанавливаются, только когда счетчик считает вниз;

10: счетчик считает вверх и вниз поочередно. Флаги прерывания устанавливаются, только когда счетчик считает вверх;

11: счетчик считает вверх и вниз поочередно. Флаги прерывания устанавливаются, в обоих случаях, когда счетчик считает вверх и вниз.

• Бит 4 ***DIR***:

0: счетчик считает вверх;

1: счетчик считает вниз.

• Бит 3 ***OPM***:

0: счетчик не останавливается при возникновении события обновления (***update event***);

1: счетчик останавливается при следующем событии обновления (очистка бита ***CEN***).

• Бит 2 ***URS***:

0: прерывание, кроме обычного способа, может также вызываться модулем внешнего тактирования, или программно, установкой бита ***UG*** в регистре ***TIM*1*_EGR***;

1: прерывание возникнет только при переполнении счетчика.

• Бит 1 ***UDIS***:

0: прерывания ***TIM*1** разрешены;

1: прерывания ***TIM*1** запрещены.

• Бит 0 ***CEN***:

0: счет запрещен;

1: счет разрешен.

Регистр контроля 2

Регистр контроля 2 (***TIM*1*_CR*2**).

7	6	5	4	3	2	1	0
Reserved	***MMS***[2:0]			***Reserved***	***COMS***	***Reserved***	***CCPC***
	rw	*rw*	*rw*		*rw*		*rw*

• Бит 7. Зарезервирован.

• Биты 6:4 ***MMS***. Выбор режима ведущего:

000: сброс – бит ***UG*** в регистре ***TIM*1*_EGR*** используется как выход запуска (***TRGO***). Если вход запуска формирует сигнал сброса, то сигнал на ***TRGO*** задерживается и сравнивается с фактическим сигналом сброса;

001: разрешение – сигнал разрешения счетчика используется в качестве выхода запуска (***TRGO***).Это используется для того, чтобы запустить несколько таймеров или АЦП и контролировать окно, в котором ведомые таймеры или АЦП запущены. Сигнал разрешения счетчика генерируется с помощью логического сложения контрольного бита ***CEN*** и входа запуска, когда он настроен в режиме закрытого запуска;

010: обновление – ***Update event*** выбирается в качестве выхода запуска (***TRGO***);

011: импульс сравнения (***MATCH*1**) – Выход запуска отправляет положительный импульс когда флаг ***CC*1*IF*** будет установлен;

100: сравнение – сигнал на ***OC*1*REF*** is используется в качестве выхода запуска (***TRGO***);

101: сравнение – сигнал на ***OC2REF*** используется в качестве выхода запуска (***TRGO***);

110: сравнение – сигнал на ***OC3REF*** используется в качестве выхода запуска (***TRGO***);

111: сравнение – сигнал на ***OC4REF*** используется в качестве выхода запуска (***TRGO***).

- Бит 3. Зарезервирован.
- Бит 2 ***COMS***:

0: когда управляющие биты захвата/сравнения предустановлены (***CCPC***=1), они обновляются, установив бит ***COMG***;

1: когда управляющие биты захвата/сравнения предустановлены (***CCPC***=1), они обновляются, установив бит COMG или на нарастающем фронте на ***TRGI***.

- Бит 1. Зарезервирован.
- Бит 0 ***CCPC***:

0: биты ***CCiE***, ***CCiNE***, ***CCiP***, и ***CCiNP*** в регистре ***TIM1_CCERi*** и бит ***OCiM*** в регистре ***TIM1_CCMRi*** не установлены;

1: биты ***CCIE***, ***CCiNE***, ***CCIP***, ***CCiNP*** и ***OCiM*** предварительно установлены и обновляются только при установке бита ***COMG*** в регистре ***TIM1_EGR***.

Регистр управления режимом ведомого

Регистр управления режимом ведомого **(*TIM1_SMCR*).**

7	6	5	4	3	2	1	0
MSM	***TS***[2:0]			***Reserved***	***SMS***[2:0]		
rw	*rw*	*rw*	*rw*		*rw*	*rw*	*rw*

- Бит 7 ***MSM***. Режим ведущий/ведомый:

0: нет действий;

1: событие на входе запуска (***TRGI***) задерживается и разрешает синхронизацию между ***TIM1*** и другим таймером (через ***TRGO***).

- Биты 6:4 ***TS***[2:0]. Выбор запуска.

В данных битах выбирается вход запуска (***TRGI***), который будет использоваться для синхронизации счетчика:

000: внутренний запуск ***ITR0*** соединяется с ***TIM6 TRGO***;

001: зарезервировано;

010: внутренний запуск ***ITR2*** соединяется с ***TIM5 TRGO***;

011: зарезервировано;

100: ***TI1*** детектор фронта (***TI1F_ED***);

101: отфильтрованный 1 вход таймера (***TI1FP1***);

110: отфильтрованный 2 вход таймера (***TI1FP2***);

111: внешний вход запуска (***ETRF***).

• Бит 3. Зарезервирован, всегда равен «0».

• Биты 2:0 ***SMS***[2:0]. Выбор режима ***clock***/***trigger***/***slave***:

000: контроллер ***clock***/***trigger*** запрещен, если бит ***CEN***=1, предварительный делитель тактируется от внутреннего источника;

001: режим 1 – счетчик считает вверх или вниз, полярность фронта на ***TI2FP2*** зависит от уровня сигнала на ***TI2FP1***;

010: режим 2 – счетчик считает вверх или вниз, полярность фронта на ***TI2FP1*** зависит от уровня сигнала на ***TI2FP2***;

011: режим 3 – счетчик считает вверх или вниз, полярность фронта на ***TI2FP1*** и ***TI2FP2*** зависит от уровня сигнала на другом входе;

100: режим сброса – передний фронт выбранного сигнала запуска (***TRGI***) перенастраивает счетчик и вызывает обновление регистров;

101: закрытый режим запуска – тактирование счетчика разрешено пока уровень сигнала запуска (***TRGI***) высокий. Счетчик останавливается (но не сбрасывается), как только уровень запускающего сигнала становится низким. И запуском, и остановкой счетчика возможно управлять;

110: стандартный режим запуска – счетчик запускается при появлении переднего фронта запускающего сигнала ***TRGI*** (но бит не сбрасывается). Возможно управление только запуском счетчика;

111: режим внешнего источника тактирования 1 – счетчик тактируется передним фронтом сигнала, на выбранном сигнале запуска (***TRGI***).

*Примечание: Закрытый режим запуска не должен использоваться, если в качестве запускающего входа выбран **TI1F_ED** (**TS**=100). **TI1F_ED** выводит 1 импульс для каждого перехода на **TI1F**, в то время как закрытый режим запуска проверяет уровень сигнала запуска.*

Регистр настройки внешнего запуска

Регистр настройки внешнего запуска **(*TIM*1_*ETR*)**.

7	6	5	4	3	2	1	0
ETP	***ECE***	***ETPS*** [1:0]		***ETF*** [3:0]			
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7 ***ETP***. Полярность внешнего запуска.

Этот бит выбирает, какой сигнал будет использоваться для запуска:

0: ***ETR*** не инвертированный, активным является высокий уровень или передний фронт;

1: ***ETR*** инвертированный, активным является низкий уровень или задний фронт.

• Бит 6 ***ECE***. Разрешение внешнего источника тактирования:

Этот бит разрешает внешний источник тактирования в режиме 2.

0: внешний источник тактирования в режиме 2 запрещен;

1: внешний источник тактирования в режиме 2 разрешен. Счетчик тактируется любым активным уровнем сигнала на ***ETRF***.

*Примечание: Установка бита **ECE** имеет тот же эффект, что и выбор внешнего источника тактирования в режиме* 1*, когда **TRGI** соединен с **ETRF** (**SMS**=*111 *и **TS**=*111 *в регистре **TIM1_SMCR**). Возможно одновременное использование внешнего источника тактирования в режиме* 2 *со следующими режимами: стандартный режим запуска, запуск в режиме сброса и запуск в закрытом режиме. Однако, **TRGI** в данном случае не должен быть соединен с **ETRF** (биты **TS**=*111 *в регистре **TIM1_SMCR**). Если режимы* 1 *и* 2 *внешнего источника тактирования разрешены одновременно, то входом внешнего источника тактирования будет **ETRF**.*

• Биты 5:4 ***ETPS***. Предварительный делитель частоты внешнего запуска. Частота ***ETPR*** должна быть, по крайней мере равна ¼ от частоты f_{MASTER}. Предварительный делитель позволяет снизить частоту ***ETPR***. Это полезно в случае приема импульсов с внешнего источника тактирования с высокой частотой. Деление частоты задается следующим образом:

00: предварительный делитель частоты отключен;

01: частота ***ETPR*** делится на 2;

10: частота ***ETPR*** делится на 4;

11: частота ***ETPR*** делится на 8.

• Биты 3:0 ***ETF***. Фильтр внешнего запуска.

Это битовое поле определяет частоту, используемую для канала ***ETPR***, сигнал и длину цифрового фильтра, применяемого к нему. Цифровой фильтр выполнен из счетчика событий (***N***), необходимых для проверки переключения на выходе:

0000: Нет фильтра, $f_{SAMPLING}=f_{MASTER}$;

0001: $f_{SAMPLING}=f_{MASTER}$, N=2;

0010: $f_{SAMPLING}=f_{MASTER}$, N=4;

0011: $f_{SAMPLING}=f_{MASTER}$, N=8;

0100: $f_{SAMPLING}=f_{MASTER}/2$, N=6;

0101: $f_{SAMPLING}=f_{MASTER}/2$, N=8;

0110: $f_{SAMPLING}=f_{MASTER}/4$, N=6;

0111: $f_{SAMPLING}=f_{MASTER}/4$, N=8;

1000: $f_{SAMPLING}=f_{MASTER}/8$, N=6;

1001: $f_{SAMPLING}=f_{MASTER}/8$, N=8;

1010: $f_{SAMPLING}=f_{MASTER}/16$, N=5;

1011: $f_{SAMPLING}=f_{MASTER}/16$, N=6;

1100: $f_{SAMPLING}=f_{MASTER}/16$, N=8;

1101: $f_{SAMPLING}=f_{MASTER}/32$, $N=5$;
1110: $f_{SAMPLING}=f_{MASTER}/32$, $N=6$;
1111: $f_{SAMPLING}=f_{MASTER}/32$, $N=8$.

Регистр разрешения прерывания

Регистр разрешения прерывания (***TIM*1_*IER***).

7	6	5	4	3	2	1	0
BIE	***TIE***	***COMIE***	***CC*4*IE***	***CC*3*IE***	***CC*2*IE***	***CC*1*IE***	***UIE***
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Бит 7 ***BIE***. Разрешение прерываний по входу останова:
  0: запретить прерывание по входу останова;
  1: разрешить прерывание по входу останова.
- Бит 6 ***TIE***. Разрешение прерываний запуска:
  0: запретить прерывание запуска;
  1: разрешить прерывание запуска.
- Бит 5 ***COMIE***. Разрешение прерываний ***Commutation***:
  0: запретить прерывание ***Commutation***;
  1: разрешить прерывание ***Commutation***.
- Биты 4:1 ***CCiIE***[3:0]. Разрешение прерываний по захвату-сравнению:

0: запретить прерывание по захвату-сравнению на соответствующем канале;

1: разрешить прерывание по захвату-сравнению на соответствующем канале.

- Бит 0 ***UIE***. Разрешение прерываний по событию обновления:
  0: запретить прерывание по событию обновления;
  1: разрешить прерывание по событию обновления.

Регистр статуса 1

Регистр статуса 1 (***TIM*1_*SR*1**).

7	6	5	4	3	2	1	0
BIF	***TIF***	***COMIF***	***CC*4*IF***	***CC*3*IF***	***CC*2*IF***	***CC*1*IF***	***UIF***
rc w0	*rc w0*	*rc w0*	*rc w0*	*rc w0*	*rc w0*	*rc w0*	*rc w0*

- Бит 7 ***BIF***. Флаг прерывания по входу останова. Этот флаг устанавливается аппаратно, как только вход останова становится активным. Он может быть очищен программно, если вход останова не активен:

0: событие останова не произошло;

1: на входе останова был обнаружен активный уровень.

• Бит 6 ***TIF***. Флаг прерывания запуска. Этот флаг устанавливается аппаратно по событию запуска (активный фронт сигнала обнаружен на входе ***TRGI***, если выбран закрытый режим запуска, то обнаруживаются оба фронта сигнала). Флаг очищается программно:

0: событие запуска не произошло;

1: ожидание прерывания запуска.

• Бит 5 ***COMIF***. Флаг прерывания ***Commutation***. Этот флаг устанавливается аппаратно на ***COM*** (когда контрольные биты захвата/сравнения – ***CCiE***, ***CCiNE***, ***OCiM*** – обновляются). Флаг очищается программно:

0: событие на ***COM*** не обнаружено;

1: ожидается прерывание ***COM***.

• Бит 4 ***CC4IF***. Флаг прерывания захвата/сравнения 4.

Аналогично описанию ***CC1IF***.

• Бит 3 ***CC3IF***. Флаг прерывания захвата/сравнения 3.

Аналогично описанию ***CC1IF***.

• Бит 2 ***CC2IF***. Флаг прерывания захвата/сравнения 2.

Аналогично описанию ***CC1IF***.

• Бит 1 ***CC1IF***. Флаг прерывания захвата/сравнения 1.

Если канал ***CC1*** настроен как выход, то этот флаг устанавливается аппаратно, когда счетчик достигает сравниваемой величины, с некоторым исключением, в случае если счетчик работает в режиме выравнивания по центру (см. описание битов ***CMS*** в регистре ***TIM1_CR1***). Флаг очищается программно:

0: нет соединения;

1: содержание регистра ***TIM1_CNT*** стало равно содержанию регистра ***TIM1_CCR1***.

Примечание: В режиме выравнивания по центру, счетчик считает вверх, когда его значение равно 0*, и считает вниз когда оно равно значению в* ***ARR*** *(он считает вверх, от* 0 *до* ***ARR***-1*, и считает вниз от* ***ARR*** *до* 1*). Эти два значения не попадают ни в одно из состояний битов* ***CMS****. Тем не менее, бит* ***CC1IF*** *устанавливается, если* ***CNT*** *достигает значения* ***ARR****, когда сравниваемое значение больше, чем значение автоматической перезагрузки (****CCR1*** *>* ***ARR****).*

Если канал ***CC1*** настроен как вход, то этот бит устанавливается аппаратно по захвату. Он очищается программно, чтением регистра ***TIM1_CCR1L***:

0: ни на одном входе захвата не было обнаружено;

1: значение счетчика было захвачено в регистр ***TIM1_CCR1*** (выбранная полярность фронта сигнала была обнаружена на входе ***IC1***).

• Бит 0 ***UIF***. Флаг прерывания по событию обновления. Этот флаг устанавливается аппаратно по переполнению. Он очищается программно:

0: переполнения не произошло;

1: ожидается прерывание по переполнению. Этот бит устанавливается аппаратно, когда регистры обновляются: по переполнению при счете вверх и вниз, если ***UDIS***=0 в регистре ***TIM*1_*CR*1**; когда ***CNT*** перенастраивается программно, используя бит ***UG*** в регистре ***TIM*1_*EGR***, если биты ***URS***=0 и ***UDIS***=0 в регистре ***TIM*1_*CR*1**; когда ***CNT*** перенастраивается по событию запуска, если биты ***URS***=0 и ***UDIS***=0 в регистре ***TIM*1_*CR*1**.

Регистр статуса 2

Регистр статуса 2 **(*TIM*1_*SR*2)**.

7	6	5	4	3	2	1	0
Зарезервировано			***CC*4*OF***	***CC*3*OF***	***CC*2*OF***	***CC*1*OF***	Зарезер-
			rc_w0	*rc_w0*	*rc_w0*	*rc_w0*	вирован

• Биты 7:5. Зарезервированы.

• Бит 4 ***CC*4*OF***. Флаг прерывания перезахвата захвата/сравнения 4.
Аналогично описанию ***CC*1*OF***.

• Бит 3 ***CC*3*OF***. Флаг прерывания перезахвата захвата/сравнения 3.
Аналогично описанию ***CC*1*OF***.

• Бит 2 ***CC*2*OF***. Флаг прерывания перезахвата захвата/сравнения 2.
Аналогично описанию ***CC*1*OF***.

• Бит 1 ***CC*1*OF***. Флаг прерывания перезахвата захвата/сравнения 1.

Этот флаг устанавливается аппаратно, только в том случае, если соответствующий канал настроен на вход в режиме захвата. Он очищается программно, записью в него 0:

0: не было обнаружено перезахвата.

1: значение счетчика было захвачено в регистр ***TIM*1_*CCR*1**, в то время как флаг ***CC*1*IF*** уже был установлен.

• Бит 0. Зарезервирован.

Регистр генерации событий

Регистр генерации событий **(*TIM*1_*EGR*)**.

7	6	5	4	3	2	1	0
BG	***TG***	***COMG***	***CC*4*G***	***CC*3*G***	***CC*2*G***	***CC*1*G***	***UG***
w	*w*	*w*	*w*	*w*	*w*	*w*	*w*

• Бит 7 ***BG***. Генерация события входа останова.

Этот бит устанавливается программно и генерирует событие. Он очищается аппаратно (автоматически):

0: ни какого действия;

1: генерируется событие по входу останова. Бит ***MOE*** очищается и устанавливается флаг ***BIF***. Если прерывание разрешено битом ***BIE***, то оно генерируется.

• Бит 6 ***TG***. Генерация события запуска.

Этот бит устанавливается программно и генерирует событие. Он очищается аппаратно (автоматически):

0: ни какого события;

1: в регистре ***TIM*1_*SR*1** устанавливается флаг ***TIF***. Если прерывание разрешено битом ***TIE***, то оно генерируется.

• Бит 5 ***COMG***. Контроль обновления генерации захвата/сравнения.

Этот бит может быть установлен программно и очищен аппаратно (автоматически):

0: ни какого события;

1: когда бит ***CCPC*** в регистре ***TIM*1_*CR*2** установлен, это позволяет обновить биты ***CCiE***, ***CCiNE***, ***CCiP***, ***CCiNP*** и ***OCiM***.

Примечание: Этот бит действует только на каналы, которые имеют дополнительный выход.

• Бит 4 ***CC*4*G***. Генерация захвата/сравнения 4.

Аналогично описанию ***CC*1*G***.

• Бит 3 ***CC*3*G***. Генерация захвата/сравнения 3.

Аналогично описанию ***CC*1*G***.

• Бит 2 ***CC*2*G***. Генерация захвата/сравнения 2.

Аналогично описанию ***CC*1*G***.

• Бит 1 ***CC*1*G***. Генерация захвата/сравнения 1.

Этот бит может быть установлен программно, чтобы сгенерировать событие. Он очищается аппаратно (автоматически).

0: ни какого действия;

1: событие захвата/сравнения генерируется на 1 канале.

Если канал ***CC*1** настроен в режиме выхода, то флаг ***CC*1*IF*** устанавливается и формируется соответствующее прерывание, если оно разрешено. Если канал ***CC*1** настроен в режиме входа, то текущее значение счетчика фиксируется в регистре ***TIM*1_*CCR*1**. Флаг ***CC*1*IF*** устанавливается и, если разрешено, то посылается запрос на соответствующее прерывание. Устанавливается флаг ***CC*1*OF***, если на данный момент уже установлен флаг ***CC*1*IF***.

• Бит 0 ***UG***. Генерация обновления.

Этот бит может быть установлен программно и очищается аппаратно (автоматически):

0: ни какого действия;

1: счетчик перенастраивается и генерируется обновление его регистров. Обратите внимание, что предварительный делитель счетчика также очищается. Счетчик очищается, если он работает в режиме выравнивания по центру или бит ***DIR***=0 (режим счета вверх). В другом случае, в него загружается значение регистра ***TIM*1_*ARR***, если бит ***DIR***=1 (режим счета вниз).

Регистр режима захвата сравнения 1

Регистр режима захвата сравнения 1 **(*TIM*1_*CCMR*1)**.

Этот канал может быть использован в качестве входа (режим захвата) или в качестве выхода (режим сравнения). Направление канала определяется путем настройки битов ***CC*1*S***. Все остальные биты данного регистра, в режиме входа и выхода, отвечают за различные функции. Каждый бит ***OCii*** описывает свою функцию, когда канал настроен на выход, ***ICii*** описывает свою функцию, когда канал настроен на вход. Поэтому следует помнить, что один и тот же бит может иметь различные функции.

Канал настроен как выход.

7	6	5	4	3	2	1	0
OC*1*CE	***OC*1*M***[2:0]			***OC*1*PE***	***OC*1*FE***	***CC*1*S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7 ***OC*1*CE***. Разрешение очистки выхода сравнения 1.

Этот бит используется для разрешения очистки выходного сигнала сравнения канала 1 (***OC*1*REF***), внешним событием на ножке ***TIM*1_*TRIG***:

0: ***OC*1*REF*** не зависит от входного сигнала ***ETRF*** (который приходит с ножки ***TIM*1_*TRIG***);

1: ***OC*1*REF*** очищается, как только высокий уровень сигнала обнаружен на ***ETRF*** (который приходит с ножки ***TIM*1_*TRIG***).

• Биты 6:4 ***OC*1*M***[2:0]. Режим выхода сравнения 1.

Эти биты определяет поведение выходного опорного сигнала, ***OC*1*REF***, от которого происходит ***OC*1**. Активный уровень сигнала на ***OC*1*REF*** – высокий уровень, в то время как, активный уровень на ***OC*1** зависит от бита ***CC*1*P***.

000: заморожен – сравнение между выходным регистром сравнения ***TIM*1_*CCR*1** и регистром счетчика ***TIM*1_*CNT*** не имеет не влияет на состояние выходов;

001: установка на канале 1 активного уровня – сигнал на ***OC*1*REF*** становится высоким, когда регистр счетчика ***TIM*1_*CNT*** соответствует регистру захвата сравнения 1 (***TIM*1_*CCR*1**);

010: установка на канале 1 неактивного уровня – сигнал на ***OC*1*REF*** становится низким, когда регистр счетчика ***TIM*1_*CNT*** соответствует регистру захвата сравнения 1 (***TIM1_CCR1***);

011: переключение – ***OC*1*REF*** переключается, когда ***TIM*1_*CNT*=*TIM*1_*CCR*1**;

100: вынужденный неактивный уровень – сигнал на ***OC*1*REF*** принудительно низкий;

101: вынужденный активный уровень – сигнал на ***OC*1*REF*** принудительно высокий;

110: режим ШИМ 1 – в режиме счета вверх, канал 1 активен (***OC*1*REF***=1) до тех пор, пока ***TIM*1_*CNT*<*TIM*1_*CCR*1**, в противном случае канал неактивен (***OC*1*REF***=0). В режиме счета вниз, канал 1 неактивен (***OC*1*REF***=0) до тех пор, пока ***TIM*1_*CNT*>*TIM1_CCR1***, в противном случае канал активен (***OC*1*REF***=1);

111: Режим ШИМ 2 – в режиме счета вверх, канал 1 неактивен (***OC*1*REF***=0) до тех пор, пока ***TIM*1_*CNT*<*TIM*1_*CCR*1**, в противном случае канал активен (***OC1REF***=1). В режиме счета вниз, канал 1 активен (***OC*1*REF***=1) до тех пор, пока ***TIM*1_*CNT*>*TIM*1_*CCR*1**, в противном случае канал неактивен (***OC*1*REF***=0).

*Примечание: Эти биты не могут быть изменены пока биты **LOCK** в регистре **TIM1_BKR** равны **LOCK**=11 и **CC1S**=00 (канал настроен на выход). В режимах ШИМ 1 или 2 уровень сигнала на **OCiREF** изменяется только когда изменяется результат сравнения или когда происходит переключение между режимами «заморожен» и «ШИМ». У каналов, которые имеют комплементарный выход, это битовое поле является предварительно загружаемым. Если в регистре **TIM1_CR2** биты **CCPC** установлены, то в биты **OCM** загружается новое значение из битов предварительной загрузки, только когда генерируется **COM**.*

- Бит 3 ***OC*1*PE***. Разрешение предварительной загрузки выхода сравнения 1:

  0: предварительный регистр ***TIM*1_*CCR*1** запрещен. ***TIM*1_*CCR*1** может быть записан в любое время. Новое значение учитывается немедленно;

  1: предварительный регистр ***TIM*1_*CCR*1** разрешен. Доступны операции чтения/записи регистра предварительной загрузки. Предварительно загруженное значение регистра ***TIM*1_*CCR*1** загружается в теневой регистр по каждому ***UEV***.

*Примечание: Эти биты не могут быть изменены пока биты **LOCK** в регистре **TIM1_BKR** равны **LOCK**=11 и **CC1S**=00 (канал настроен на выход). Для правильной работы, регистры предварительной загрузки должны быть разрешены, когда таймер работает в режиме ШИМ. Это не является обязательным в режиме одиночного импульса (в регистре **TIM1_CR1** бит **OPM** установлен).*

• Бит 2 ***OC1FE***. Быстрое разрешение выхода сравнения 1:

0: ***CC1*** ведет себя стандартно в зависимости от счетчика и значений ***CCR1***, даже когда включен запуск. Минимальная задержка активации выхода ***CC1***, с того момента как на входе запуска обнаружен фронт сигнала, составляет 5 циклов;

1: активный фронт на входе запуска действует на выход ***CC1*** как совпадение сравнения. Если это происходит, то ***OC*** устанавливается независимо от результата сравнения. Задержка для сравнения входа запуска и активации выхода ***CC1*** составляет 3 цикла. ***OCFE*** действует, только если канал настроен в режимах ШИМ 1 или 2.

• Биты 1:0 ***CC1S***[1:0]. Выбор захвата/сравнения 1.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC1*** настроен как выход;

01: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TI1FP1***;

10: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TI2FP1***;

11: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC1S** доступны только для записи, когда канал выключен (**CC1E**=0 в регистре **TIM1_CCER1**).*

Канал настроен как вход.

7	6	5	4	3	2	1	0
IC1F[3:0]				***IC1PSC***[1:0]		***CC1S***[1:0]	
rw	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***

• Биты 7:4 ***IC1F***[3:0]. Фильтр входа захвата 1.

Данное битовое поле определяет частоту $f_{SAMPLING}$, которая используется в качестве канала для входа ***TI1***. Цифровой фильтр выполнен из счетчика событий (***N***), необходимых для проверки перехода на выходе:

0000: Фильтра нет, $f_{SAMPLING}=f_{MASTER}$;

0001: $f_{SAMPLING}=f_{MASTER}$, $N=2$;

0010: $f_{SAMPLING}=f_{MASTER}$, $N=4$;

0011: $f_{SAMPLING}=f_{MASTER}$, $N=8$;

0100: $f_{SAMPLING}=f_{MASTER}/2$, $N=6$;

0101: $f_{SAMPLING}=f_{MASTER}/2$, $N=8$;

0110: $f_{SAMPLING}=f_{MASTER}/4$, $N=6$;

0111: $f_{SAMPLING}=f_{MASTER}/4$, $N=8$;
1000: $f_{SAMPLING}=f_{MASTER}/8$, $N=6$;
1001: $f_{SAMPLING}=f_{MASTER}/8$, $N=8$;
1010: $f_{SAMPLING}=f_{MASTER}/16$, $N=5$;
1011: $f_{SAMPLING}=f_{MASTER}/16$, $N=6$;
1100: $f_{SAMPLING}=f_{MASTER}/16$, $N=8$;
1101: $f_{SAMPLING}=f_{MASTER}/32$, $N=5$;
1110: $f_{SAMPLING}=f_{MASTER}/32$, $N=6$;
1111: $f_{SAMPLING}=f_{MASTER}/32$, $N=8$.

*Примечание: Даже на каналах, которые имеют комплементарные выходы, это битовое поле не является предварительно загружаемым и не зависит от содержания бита **CCPC** (в регистре **TIM1_CR2**).*

• Биты 3:2 ***IC1PSC***[1:0]. Предварительный делитель входа захвата 1.

Это битовое поле определяет коэффициент делителя, действующего на входе ***CC1*** (***IC1***). Предварительный делитель сбрасывается, как только ***CC1E***=0 (в регистре ***TIM1_CCER***):

00: нет предварительного делителя, захват производится каждый раз, при обнаружении фронта сигнала на входе захвата;
01: захват осуществляется 1 раз за каждые 2 события;
10: захват осуществляется 1 раз за каждые 4 события;
11: захват осуществляется 1 раз за каждые 8 событий.

• Биты 1:0 ***CC1S***[1:0]. Выбор захвата/сравнения 1.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC1*** настроен как выход;
01: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TI1FP1***;
10: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TI2FP1***;
11: канал ***CC1*** настроен как вход, ***IC1*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC1S** доступны только для записи, когда канал выключен (**CC1E**=0 в регистре **TIM1_CCER1**).*

Регистр режима захвата сравнения 2

Регистр режима захвата сравнения 2 **(*TIM1_CCMR2*)**.

Канал настроен как выход.

7	6	5	4	3	2	1	0
OC2CE	***OC2M***[2:0]			***OC2PE***	***OC2FE***	***CC2S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7 ***OC2CE***. Разрешение очистки выхода сравнения 2.

• Биты 6:4 ***OC2M***[2:0]. Режим выхода сравнения 2.

• Бит 3 ***OC2PE***. Разрешение предварительной загрузки выхода сравнения 2.
• Бит 2 ***OC2FE***. Быстрое разрешение выхода сравнения 2.
• Биты 1:0 ***CC2S***[1:0]. Выбор захвата/сравнения 2.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC2*** настроен как выход;

01: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TI1FP2***;

10: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TI2FP2***;

11: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC2S** доступны только для записи, когда канал выключен (**CC2E**=0 и **CC2NE** =0 в регистре **TIM1_CCER1**).*

Канал настроен как вход.

7	6	5	4	3	2	1	0
IC2F[3:0]				***IC2PSC***[1:0]		***CC2S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:4 ***IC2F***. Фильтр входа захвата 2.
• Биты 3:2 ***IC2PSC***[1:0]. Предварительный делитель входа захвата 2.
• Биты 1:0 ***CC2S***[1:0]. Выбор захвата/сравнения 2.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC2*** настроен как выход;

01: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TI1FP2***;

10: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TI2FP2***;

11: канал ***CC2*** настроен как вход, ***IC2*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC2S** доступны только для записи, когда канал выключен (**CC2E**=0 и **CC2NE** =0 в регистре **TIM1_CCER1**).*

Регистр режима захвата сравнения 3

Регистр режима захвата сравнения 3 **(*TIM1_CCMR3*):**

Канал настроен как выход.

7	6	5	4	3	2	1	0
OC3CE	***OC3M***[2:0]			***OC3PE***	***OC3FE***	***CC3S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7 ***OC3CE***. Разрешение очистки выхода сравнения 3.
• Биты 6:4 ***OC3M***[2:0]. Режим выхода сравнения 3.

- Бит 3 ***OC3PE***. Разрешение предварительной загрузки выхода сравнения 3.
- Бит 2 ***OC3FE***. Быстрое разрешение выхода сравнения 3.
- Биты 1:0 ***CC3S***[1:0]. Выбор захвата/сравнения 3.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход.

  00: Канал ***CC3*** настроен как выход;

  01: Канал ***CC3*** настроен как вход, ***IC3*** отображается на *TI3FP3*;

  10: Канал ***CC3*** настроен как вход, ***IC3*** отображается на ***TI4FP3***;

  11: Канал ***CC3*** настроен как вход, ***IC3*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

  *Примечание: Биты **CC3S** доступны только для записи, когда канал выключен (**CC3E**=0 и **CC3NE** =0 в регистре **TIM1_CCER2**).*

  Канал настроен как вход.

7	6	5	4	3	2	1	0
IC3F[3:0]				***IC3PSC***[1:0]		***CC3S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:4 ***IC3F***[3:0]. Фильтр входа захвата 3.
- Биты 3:2 ***IC3PSC***[1:0]. Предварительный делитель входа захвата 3.
- Биты 1:0 ***CC3S***[1:0]. Выбор захвата/сравнения 3.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC3*** настроен как выход;

01: канал ***CC3*** настроен как вход, ***IC3*** отображается на ***TI3FP3***;

10: канал ***CC3*** настроен как вход, ***IC3*** отображается на ***TI4FP3***;

11: канал ***CC3*** настроен как вход, ***IC3*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC3S** доступны только для записи, когда канал выключен (**CC3E**=0 и **CC3NE** =0 в регистре **TIM1_CCER2**).*

Регистр режима захвата сравнения 4

Регистр режима захвата сравнения 4 **(*TIM1_CCMR4*):**

Канал настроен как выход.

7	6	5	4	3	2	1	0
OC4CE	***OC4M***[2:0]			***OC4PE***	***OC4FE***	***CC4S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Бит 7 ***OC4CE***. Разрешение очистки выхода сравнения 4.
- Биты 6:4 ***OC4M***[2:0]. Режим выхода сравнения 4.

- Бит 3 ***OC4PE***. Разрешение предварительной загрузки выхода сравнения 4.
- Бит 2 ***OC4FE***. Быстрое разрешение выхода сравнения 4.
- Биты 1:0 ***CC4S***[1:0]. Выбор захвата/сравнения 4.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC4*** настроен как выход;

  01: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TI4FP4***;

  10: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TI3FP4***;

  11: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC4S** доступны только для записи, когда канал выключен (**CC4E**=0 и **CC4NE** =0 в регистре **TIM1_CCER2**).*

Канал настроен как вход.

7	6	5	4	3	2	1	0
IC4F[3:0]				***IC4PSC***[1:0]		***CC4S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:4 ***IC4F***[3:0]. Фильтр входа захвата 4.
- Биты 3:2 ***IC4PSC***[1:0]. Предварительный делитель входа захвата 4.
- Биты 1:0 ***CC4S***[1:0]. Выбор захвата/сравнения 4.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC4*** настроен как выход;

  01: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TI4FP4***;

  10: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TI3FP4***;

  11: канал ***CC4*** настроен как вход, ***IC4*** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM1_SMCR***).

*Примечание: Биты **CC4S** доступны только для записи, когда канал выключен (**CC4E**=0 и **CC4NE** =0 в регистре **TIM1_CCER2**).*

Регистр разрешения захвата/сравнения 1

Регистр разрешения захвата/сравнения 1 **(*TIM1_CCER1*):**

7	6	5	4	3	2	1	0
CC2NP	***CC2NE***	***CC2P***	***CC2E***	***CC1NP***	***CC1NE***	***CC1P***	***CC1E***
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Бит 7 ***CC2NP***. Полярность комплементарного выхода захвата/сравнения 2.

  Аналогично описанию бита ***CC1NP***.

- Бит 6 ***CC2NE***. Разрешение комплементарного выхода захвата/сравнения 2.

  Аналогично описанию бита ***CC1NE***.
- Бит 5 ***CC2P***. Полярность выхода захвата/сравнения 2.

  Аналогично описанию бита ***CC1P***.
- Бит 4 ***CC2E***. Разрешение выхода захвата/сравнения 2.

  Аналогично описанию бита *CC*1*E*.
- Бит 3 ***CC1NP***. Полярность комплементарного выхода захвата/сравнения1:

  0: активный уровень ***OC1N*** высокий;

  1: активный уровень ***OC1N*** низкий.

*Примечание: Этот бит не может быть изменен пока биты **LOCK** в регистре **TIM1_BKR** равны **LOCK**=11 или **LOCK**=10 и CC1S=00 (канал настроен на выход). У каналов, которые имеют комплементарный выход, этот бит является предварительно загружаемым. Если в регистре **TIM1_CR2** биты **CCPC** установлены, то в биты **CC1NP** загружается новое значение из битов предварительной загрузки, только когда генерируется **COM**.*

- Бит 2 ***CC1NE***. Разрешение комплементарного выхода захвата/сравнения 1:

  0: выключен – ***OC1N*** не активен;

  1: включен – сигнал на ***OC1N*** является выходом соответствующего контакта в зависимости от битов ***MOE***, ***OSSI***, ***OSSR***, ***OIS1***, ***OIS1N*** и ***CC1E***.

*Примечание: У каналов, которые имеют комплементарный выход, этот бит является предварительно загружаемым. Если в регистре **TIM1_CR2** биты **CCPC** установлены, то в биты **CC1NE** загружается новое значение из битов предварительной загрузки, только когда генерируется **COM**.*

- Бит 1 ***CC1P***. Полярность выхода захвата/сравнения 1.

  Канал *CC*1 настроен как выход:

  0: активный уровень ***OC1*** высокий;

  1: активный уровень ***OC1*** низкий.

  Канал *CC*1 настроен как вход для запуска:

  0: переключение по высокому уровню или по переднему фронту сигнала на ***TI1F***;

  1: переключение по низкому уровню или по заднему фронту сигнала на ***TI1F***.

  Канал *CC*1 настроен как вход в качестве входа захвата:

  0: захват по переднему фронту сигнала на ***TI1F*** или ***TI2F***;

  1: захват по заднему фронту сигнала на ***TI1F*** или ***TI2F***.

*Примечание: Этот бит не может быть изменен пока биты **LOCK** в регистре **TIM1_BKR** равны **LOCK**=11 или **LOCK**=10. У каналов, которые имеют комплементарный выход, этот бит является предварительно загружаемым. Если в регистре **TIM1_CR2** биты **CCPC** установлены, то в биты **CC1P** загружается новое значение из битов предварительной загрузки, только когда генерируется **COM**.*

• Бит 0 ***CC*1*E***. Разрешение выхода захвата/сравнения 1.

Канал *CC*1 настроен как выход:

0: выключен – ***OC*1** не активен;

1: включен – сигнал на ***OC*1** является выходом соответствующего контакта в зависимости от битов ***MOE***, ***OSSI***, ***OSSR***, ***OIS*1*N***, ***OIS*1*NE*** и ***CC*1*E***.

Канал *CC*1 настроен как вход:

Этот бит определяет, является ли захваченное значение счетчика входом регистра захвата/сравнения ***TIM*1_*CCR*1** или нет:

0: захват запрещен;

1: захват разрешен.

*Примечание: У каналов, которые имеют комплементарный выход, этот бит является предварительно загружаемым. Если в регистре **TIM*1_*CR*2** биты **CCPC** установлены, то в биты **CC*1*E** загружается новое значение из битов предварительной загрузки, только когда генерируется **COM**.*

Контрольные биты					Состояние выходов	
MOE	***OSSI***	***OSSR***	***CCiE***	***CCiNE***	***OCi***	***OCiN***
1	x	0	0	0	Выход запрещен (не управляется таймером)	Выход запрещен (не управляется таймером)
		0	0	1	Выход запрещен (не управляется таймером)	***OCiREF*** + полярность ***OCiN***= ***OCiREF xor CCiNP***
		0	1	0	***OCiREF*** + полярность ***OCi***= ***OCiREF xor CCiP***	Выход запрещен (не управляется таймером)
		0	1	1	***OCiREF*** + полярность + ***deadtime***	
		1	0	0	Выход запрещен (не управляется таймером)	Выход запрещен (не управляется таймером)
		1	0	1		
		1	1	0		
		1	1	1		
0	0	x	x	x	Выход запрещен (не управляется таймером)	
	0					
	0					
	0					
	1					
	1					
	1					
	1					

Регистр разрешения захвата/сравнения 2

Регистр разрешения захвата/сравнения 2 (***TIM*1_*CCER*2**):

7	6	5	4	3	2	1	0
Зарезервировано		***CC*4*P***	***CC*4*E***	***CC*3*NP***	***CC*3*NE***	***CC*3*P***	***CC*3*E***
		rw	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:6. Зарезервировано.
- Бит 5 ***CC*4*P***. Полярность выхода захвата/сравнения 4.
  Аналогично описанию бита ***CC*1*P***.
- Бит 4 ***CC*4*E***. Разрешение выхода захвата/сравнения 4.
  Аналогично описанию бита ***CC*1*E***.
- Бит 3 ***CC*3*NP***. Полярность комплементарного выхода захвата/сравнения 3.
  Аналогично описанию бита ***CC*1*NP***.
- Бит 2 ***CC*3*NE***. Разрешение комплементарного выхода захвата/сравнения 3.
  Аналогично описанию бита ***CC*1*NE***.
- Бит 1 ***CC*3*P***. Полярность выхода захвата/сравнения 3.
  Аналогично описанию бита ***CC*1*P***.
- Бит 0 ***CC*3*E***. Разрешение выхода захвата/сравнения 3.
  Аналогично описанию бита ***CC*1*E***.

Старший регистр счетчика

Старший регистр счетчика (***TIM*1_*CNTRH***):

7	6	5	4	3	2	1	0
CNT[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CNT***[15:8]. Значение счетчика (Старший байт).

Младший регистр счетчика

Младший регистр счетчика (***TIM*1_*CNTRL***):

7	6	5	4	3	2	1	0
CNT[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CNT***[7:0]. Значение счетчика (Младший байт).

Старший регистр предварительного делителя

Старший регистр предварительного делителя (***TIM*1_*PSCRH***):

7	6	5	4	3	2	1	0
PSC[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***PSC*** [15:8]. Значение предварительного делителя (Старший байт). Значение предварительного делителя делит частоту тактирования ***CK_PSC***. Частота тактирования счетчика f_{CK_CNT} равна f_{CK_PSC} / (***PSCR***[15:0]+1). ***PSCR*** содержит значения, которые загружаются в активный регистр предварительного делителя по каждому ***UEV***.

Младший регистр предварительного делителя

Младший регистр предварительного делителя (***TIM*1_*PSCRL*):**

7	6	5	4	3	2	1	0
PSC[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***PSC***[7:0]. Значение предварительного делителя (Младший байт).

Старший авто-перезагружаемый регистр

Старший авто-перезагружаемый регистр (***TIM*1_*ARRH*):**

7	6	5	4	3	2	1	0
ARR[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***ARR***[15:8]. Значение авто-перезагружаемого регистра (Старший байт). ***ARR*** это значение, которое будет загружено в авто-перезагружаемый регистр. Счетчик блокируется до тех пор, пока значение авто-перезагружаемого регистра равно нулю.

Младший авто-перезагружаемый регистр

Младший авто-перезагружаемый регистр (***TIM*1_*ARRL*):**

7	6	5	4	3	2	1	0
ARR[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***ARR***[7:0]. Значение авто-перезагружаемого регистра (Младший байт).

Регистр повторителя счетчика

Регистр повторителя счетчика (***TIM*1_*RCR*):**

7	6	5	4	3	2	1	0
REP[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***REP***[7:0]. Значение повторителя счетчика.

Когда регистры предварительной загрузки разрешены, эти биты позволяют пользователю настроить частоту обновления регистров сравнения, а так же частоту генерации прерываний по обновлению, если они разрешены (***UIE***=1). Каждый раз, когда ***REP_CNT***, связанный со счетчиком, считающим вниз, по достижению нуля, генерируется ***UEV*** и перезапускает счет со значения ***REP***. Как только регистр ***REP_CNT*** перезагружается значением ***REP*** (только по событию обновления повторителя ***U_RC***), любая запись в регистр ***TIM*1_*RCR*** игнорируется до следующего события обновления повторителя.

В режиме ШИМ (***REP***+1) соответствует:

•Числу периодов ШИМ в режиме выравнивания по краю;

•Половине периодов ШИМ в режиме выравнивания по центру.

Старший регистр захвата/сравнения 1

Старший регистр захвата/сравнения 1 **(*TIM*1_*CCR*1*H*):**

7	6	5	4	3	2	1	0
***CCR*1**[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*1**[15:8]. Значение захвата/сравнения 1 (Старший байт).

Если канал *CC*1 настроен как выход (биты ***CC*1*S*** в регистре ***TIM*1_*CCMR*1**):

Значение ***CCR*1** постоянно загружается в регистр захвата/сравнения 1, если предварительная загрузка разрешена (бит ***OC*1*PE*** в регистре ***TIMx_CCMR*1**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 1, когда обнаружено событие обновления. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*1**.

Если канал *CC1* настроен как вход (биты ***CC1S*** в регистре ***TIM*1_*CCMR*1**):

Значение ***CCR*1** является значением счетчика, загружается по последнему событию на входе захвата 1 (***IC*1**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 1

Младший регистр захвата/сравнения 1 **(*TIM*1_*CCR*1*L*):**

7	6	5	4	3	2	1	0
***CCR*1**[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*1**[7:0]. Значение захвата/сравнения 1 (Младший байт).

Старший регистр захвата/сравнения 2

Старший регистр захвата/сравнения 2 **(*TIM*1_*CCR*2*H*):**

7	6	5	4	3	2	1	0
***CCR*2**[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*2**[15:8]. Значение захвата/сравнения 2 (Старший байт).

Если канал *CC*2 настроен как выход (биты ***CC*2*S*** в регистре ***TIM*1_*CCMR*2**):

Значение ***CCR*2** постоянно загружается в регистр захвата/сравнения 2, если предварительная загрузка разрешена (бит ***OC*2*PE*** в регистре ***TIMx_CCMR*2**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 2, когда обнаружено ***UEV***. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*2**.

Если канал *CC2* настроен как вход (биты ***CC*2*S*** в регистре ***TIM*2_*CCMR*2**):

Значение ***CCR*2** является значением счетчика, загружается по последнему событию на входе захвата 2 (***IC*2**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 2

Младший регистр захвата/сравнения 2 **(*TIM*1_*CCR*2*L*):**

7	6	5	4	3	2	1	0
***CCR*2**[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*2**[7:0]. Значение захвата/сравнения 2 (Младший байт).

Старший регистр захвата/сравнения 3

Старший регистр захвата/сравнения 3 (***TIM*1_*CCR*3*H*):**

7	6	5	4	3	2	1	0
***CCR*3**[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*3**[15:8]. Значение захвата/сравнения 3 (Старший байт).

Если канал *CC*3 настроен как выход (биты ***CC*3*S*** в регистре ***TIM*1_*CCMR*3**):

Значение ***CCR*3** постоянно загружается в регистр захвата/сравнения 3, если предварительная загрузка разрешена (бит ***OC*3*PE*** в регистре ***TIMx_CCMR*3**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 3, когда обнаружено ***UEV***. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*3**.

Если канал *CC*3 настроен как вход (биты ***CC*3*S*** в регистре ***TIM1_CCMR*3**):

Значение ***CCR*3** является значением счетчика, загружается по последнему событию на входе захвата 3 (***IC*3**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 3

Младший регистр захвата/сравнения 3 (***TIM*1_*CCR*3*L*):**

7	6	5	4	3	2	1	0
***CCR*3**[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*3**[7:0]. Значение захвата/сравнения 3 (Младший байт).

Старший регистр захвата/сравнения 4

Старший регистр захвата/сравнения 4 (***TIM*1_*CCR*4*H*):**

7	6	5	4	3	2	1	0
***CCR*4**[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CCR*4**[15:8]. Значение захвата/сравнения 4 (Старший байт).

Если канал *CC*4 настроен как выход (биты ***CC*4*S*** в регистре ***TIM1_CCMR*4**):

Значение ***CCR*4** постоянно загружается в регистр захвата/сравнения 4, если предварительная загрузка разрешена (бит ***OC*4*PE*** в регистре ***TIMx_CCMR*4**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 4, когда обнаружено ***UEV***. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*4**.

Если канал *CC*4 настроен как вход (биты ***CC*4*S*** в регистре ***TIM1_CCMR*4**):

Значение ***CCR*4** является значением счетчика, загружается по последнему событию на входе захвата 4 (***IC*4**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 4

Младший регистр захвата/сравнения 4 **(*TIM*1_*CCR*4*L*):**

7	6	5	4	3	2	1	0
***CCR*4**[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CCR*4**[7:0]. Значение захвата/сравнения 4 (Младший байт).

Регистр остановки

Регистр остановки **(*TIM*1_*BKR*):**

7	6	5	4	3	2	1	0
MOE	***AOE***	***BKP***	***BKE***	***OSSR***	***OSSI***	***LOCK***	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Бит 7 ***MOE***. Разрешение главного выхода.

Этот бит очищается асинхронно аппаратно, как только вход останова стал активным. Устанавливается программно или автоматически, в зависимости от бита ***AOE***. Он оказывает влияние только на каналы, которые настроены как выходы:

0: выходы ***OC*** и ***OCN*** запрещены или находятся в режиме ожидания;

1: выходы ***OC*** и ***OCN*** разрешены, если биты их разрешения установлены (биты ***CCiE*** в регистре ***TIM*1_*CCERi***).

- Бит 6 ***AOE***. Автоматическое разрешение выхода:

0: ***MOE*** может быть установлен только программно;

1: ***MOE*** может быть установлен программно или автоматически при следующем ***UEV*** (если вход останова не активен).

*Примечание: Этот бит не может быть изменен, если биты **LOCK**=01.*

- Бит 5 BKP. Полярность остановы:

0: активный уровень входа останова ***BKIN*** – низкий;

1: активный уровень входа останова ***BKIN*** – высокий.

*Примечание: Этот бит не может быть изменен, если биты **LOCK**=01.*

- Бит 4 BKE. Разрешение остановы.

0: вход останова (***BKIN***) запрещен;

1: вход останова (***BKIN***) разрешен.

*Примечание: Этот бит не может быть изменен, если биты **LOCK**=01.*

- Бит 3 OSSR. Выбор состояния выключения режима Run.

Этот бит используется, когда ***MOE***=1, для каналов с комплементарным выходом, которые настроены как выходы:

0: когда не активен, выходы ***OC***/***OCN*** запрещены (***OC***/***OCN*** разрешают выходной сигнал равный нулю);

1: когда не активен, выходы ***OC***/***OCN*** разрешены с неактивным уровнем на них, до тех пор, пока ***CCiE***=1 или ***CCiNE***=1.

*Примечание: Этот бит не может быть изменен, если биты **LOCK**=10.*

- Бит 2 ***OSSI***. Выбор состояния выключения режима ожидания.

Этот бит используется, когда ***MOE***=0, для каналов, которые настроены как выходы:

0: когда не активен, выходы ***OCi*** не активны (***OCi*** разрешает выходной сигнал равный нулю);

1: когда не активен, на выходах ***OCi*** уровень ожидания, до тех пор, пока ***CCiE***=1 (***OC*** разрешает выходной сигнал равный единице).

*Примечание: Этот бит не может быть изменен, если биты **LOCK**=10.*

- Биты 1:0 ***LOCK***[1:0]. Настройка запрета изменения битов (***lock***).

Эти биты являются защитой от программных ошибок:

00: запрет выключен – не один бит не защищен от записи;

01: запрет 1 уровня – бит ***OISi*** в регистре ***TIM1_OISR*** и биты ***BKE/BKP/AOE*** в регистре ***TIM1_BKR*** не могут быть записаны;

10: запрет 2 уровня – запрет 1 уровня + биты полярности ***CC*** (биты ***CCiP*** в регистрах ***TIM1_CCERi***, до тех пор, пока канал настроен как выход, с помощью битов ***CCiS***), а так же биты ***OSSR*** и ***OSSI*** не могут быть записаны;

11: запрет 3 уровня – запрет 2 уровня + контрольные биты ***CC*** (биты ***OCiM*** и ***OCiPE*** в регистрах ***TIM1_CCMRi***, до тех пор, пока канал настроен как выход, с помощью битов ***CCiS***) не могут быть записаны.

*Примечание: Биты **LOCK** могут быть записаны один раз после сброса. Регистр **TIM1_BKR** записывается один раз, его содержимое замораживается до следующего сброса.*

*Примечание: Так как биты **AOE**, **BKP**, **BKE**, **OSSR** и **OSSI** могут быть заблокированы для записи, в зависимости от настройки битов **LOCK**, то необходимо настроить их все во время первой записи регистра **TIM1_BKR**.*

Регистр задания паузы

Регистр задания паузы (***TIM*1_*DTR***):

7	6	5	4	3	2	1	0
DTG[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***DTG***[7:0]. Установка генератора паузы.

Это битовое поле определяет длительность паузы, вставленной между комплементарными выходами. ***DT*** соответствует этой длительности. t_{CK_PSC} является одним тактовым импульсом таймера ***TIM*1**.

DTG[7:5]=0xx=>***DT***=***DTG***[7:0] x t_{dtg}, где t_{dtg}=t_{CK_PSC} (***f1***).
DTG[7:5]=10x=>***DT***=(64 + DTG[5:0]) x t_{dtg}, где t_{dtg}= 2 x t_{CK_PSC} (***f2***).
DTG[7:5]=110=>***DT***=(32 + DTG[4:0]) x t_{dtg}, где t_{dtg}= 8 x t_{CK_PSC} (***f3***).
DTG[7:5]=111=>***DT***=(32 + DTG[4:0]) x t_{dtg}, где t_{dtg}=16 x t_{CK_PSC} (***f4***).
Пример:
Если t_{CK_PSC}= 125 ***ns*** (8 ***MHz***), то возможны следующие времена задержки:
DTG[7:0]=0 x 0 до 0 x 7***F*** от 0 до 15875 ***ns*** за 125 ***ns*** шагов
DTG[7:0]=0 x 80 до 0 x ***BF*** от 16 ***µs*** до 31750 ***ns*** за 250 ***ns*** шагов
DTG[7:0]=0 x ***C***0 до 0 x ***DF*** от 32 ***µs*** до 63 ***µs*** за ***1µs*** шаг
DTG[7:0]=0 x ***E***0 до 0 x ***FF*** от 64 ***µs*** до 126 ***µs*** за 2 ***µs*** шага
*Примечание: Это битовое поле не может быть изменено, если в регистре **TIM1_BKR** биты **LOCK** равны 01, 10, или 11.*

Выходной регистр ожидания

Выходной регистр ожидания (***TIM*1_*OISR***):

7	6	5	4	3	2	1	0
Зарезервировано	***OIS*4**	***OIS*3*N***	***OIS*3**	***OIS*2*N***	***OIS*2**	***OIS*1*N***	***OIS*1**
	rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Бит 7. Зарезервирован, аппаратно сброшен.
• Бит 6 ***OIS*4**. Состояние ожидания выхода 4 (выход ***OC*4**).
Аналогично описанию бита ***OIS*1**.
• Бит 5 ***OIS*3*N***. Состояние ожидания выхода 3 (выход ***OC*3*N***).
Аналогично описанию бита ***OIS*1*N***.
• Бит 4 ***OIS*3**. Состояние ожидания выхода 3 (выход ***OC*3**).
Аналогично описанию бита ***OIS*1**.
• Бит 3 ***OIS*2*N***. Состояние ожидания выхода 2 (выход ***OC2N***).
Аналогично описанию бита ***OIS*1*N***.
• Бит 2 ***OIS*2**. Состояние ожидания выхода 2 (выход ***OC*2**).
Аналогично описанию бита ***OIS*1**.

- Бит 1 ***OIS*1*N***. Состояние ожидания выхода 1 (выход ***OC*1*N***):
  0: ***OC*1*N***=0 после паузы, когда ***MOE***=0;
  1: ***OC*1*N***=1 после паузы, когда ***MOE***=0.
  *Примечание: Этот бит не может быть изменен, если в регистре **TIM*1*_BKR** биты **LOCK** равны 01, 10, или 11.*
- Бит 0 ***OIS*1**: Состояние ожидания выхода 1 (выход ***OC*1*)***:
  0: ***OC*1**=0 после паузы, если реализован ***OC*1*N***, когда ***MOE***=0;
  1: ***OC*1**=1 после паузы, если реализован ***OC*1*N***, когда ***MOE***=0.
  *Примечание: Этот бит не может быть изменен, если в регистре **TIM*1*_BKR** биты **LOCK** равны 01, 10, или 11.*

4.3. Примеры настройки таймера 1

Пример настройки таймера *TIM*1

В данной программе реализуется эффект мерцания светодиода по прерыванию таймера 1.

```
#include "iostm8s003k3.h"          //Подключение заголовочного файла с объявлениями регистров, масок и битов
void portD_init(void);             //Объявление подпрограммы настройки порта D
void timer_init(void);             //Объявление подпрограммы настройки таймера
void interrupt_init(void);         //Объявление подпрограммы настройки прерываний
#pragma vector=0x0D                //В таблице прерываний у таймера TIM1 номер прерывания равен 11.Прибавим 2 получим 13, в hex формате: 0x0D
__interrupt void TIM1_OVR_UIF(void);   //Объявление вектора прерывания таймера
int main( void )                   //Основная программа
{
portD_init();                      //Вызов подпрограммы настройки порта
timer_init();                      //Вызов подпрограммы настройки таймера
interrupt_init();                  //Вызов подпрограммы настройки прерываний
  for (;;)                         //Бесконечный цикл
  {
  }
}
```

```
//Подпрограмма настройки порта
void portD_init(void)
{
//Настройка нулевого бита порта D
PD_DDR_bit.DDR0=1;          //0-вход; 1-выход
PD_CR1_bit.C10=1;           //0-выход с открытым стоком; 1-выход типа Push-pull
PD_CR2_bit.C20=0;           //Скорость переключения: 0-до 2 MHz; 1-до 10MHz
PD_ODR_bit.ODR0= 0;         //Зажигаем светодиод
}

//Подпрограмма настройки таймера
void timer_init(void)
{
CLK_PCKENR2=0xff;           //Тактирование таймера
TIM1_CR1_bit.URS=1;         //Прерывание таймера только при переполнении
TIM1_CR1_bit.DIR=0;         //0-таймер считает вверх; 1-таймер считает вниз
TIM1_CR1_bit.CMS=0;         //Выбор режима счета
TIM1_IER_bit.UIE=1;         //Установка бита разрешает прерывание по переполнению
TIM1_PSCRH=0x00;            //Старший регистр предделителя
TIM1_PSCRL=0x10;            //Младший регистр предделителя
TIM1_CR1_bit.CEN=1;         //Запуск таймера
}


//Подпрограмма настройки прерываний
void interrupt_init(void)
{
asm("rim");                 //Глобальное разрешение прерываний
}

//Подпрограмма обработки прерывания таймера TIM1
__interrupt void TIM1_OVR_UIF(void)
{
TIM1_SR1_UIF=0;         //Очистка флага переполнения
//Инверсия нулевого бита порта D
PD_ODR_bit.ODR0=! PD_ODR_bit.ODR0;
```

```
TIM1_CR1_bit.CEN=1;          //Запуск таймера
}
```

Пример программы изменения частоты мерцания светодиода таймера *TIM*1

```
#include "iostm8s003k3.h"    //подключение заголовочного файла с объ-
                             явлениями регистров, масок и битов
void portD_init(void);
void timer_init(void);
void interrupt_init(void);
int i,j,a;

#pragma vector=0x0D          //в таблице прерываний у таймера TIM1
                             номер прерывания равен 11.Прибавим 2
                             получим 13, в hex формате: 0x0D
__interrupt void TIM1_OVR_UIF(void);   //объявление вектора преры-
                             вания таймера
#pragma vector=0x06
__interrupt void EXTI_PB7(void);       //имя вектора внешнего прерывания
#pragma vector=0x07
__interrupt void EXTI_PC7(void);       //имя вектора внешнего прерывания

int main( void )             //основная программа
{
ITC_SPR3=0xC0;               //приоритет прерываний таймера - высокий
ITC_SPR2=0x05;               //приоритет внешних прерываний - низкий
PB_DDR_bit.DDR7= 0;          //0-вход; 1-выход
PB_CR1_bit.C17=0;            //0-дифференциальный вход; 1-вход с под-
                             тягивающим резистором
PB_CR2_bit.C27=1;            //0-прерывания запрещены; 1-прерывания
                             разрешены

PC_DDR_bit.DDR7= 0;          //0-вход; 1-выход
PC_CR1_bit.C17=0;            //0-дифференциальный вход; 1-вход с под-
                             тягивающим резистором
PC_CR2_bit.C27=1;            //0-прерывания запрещены; 1-прерывания
                             разрешены

portD_init();                //вызов подпрограммы настройки порта D
timer_init();                //вызов подпрограммы настройки таймера
interrupt_init();            //вызов подпрограммы настройки преры-
                             ваний
```

```
  for (;;)                          //бесконечный цикл
  {
  }
}

void portD_init(void)
{
//Настройка нулевого бита порта D
PD_DDR_bit.DDR0=1;          //0-вход; 1-выход
PD_CR1_bit.C10=1;           //0-выход с открытым стоком; 1- типа
                            Push-pull
PD_CR2_bit.C20=0;           //скорость переключения: 0-2 MHz; 1-
                            10MHz
PD_ODR_bit.ODR0= 0;         //зажигаем светодиод
}

void timer_init(void)
{
//Настройка таймера
CLK_PCKENR2=0xff;           //тактирование таймера
TIM1_CR1_bit.URS=1;         //прерывание таймера только при перепол-
                            нении
TIM1_CR1_bit.DIR=0;         //0-счет вверх; 1-счет вниз
TIM1_CR1_bit.CMS=0;         //выбор режима счета
TIM1_IER_bit.UIE=1;         //установка бита разрешает прерывание по
                            переполнению
TIM1_PSCRH=0x10;            //старший регистр предделителя
TIM1_PSCRL=0x10;            //младший регистр предделителя
TIM1_ARRH=0x00;             //число до которого считает таймер
TIM1_ARRL=0x10;             //число до которого считает таймер
TIM1_CR1_bit.CEN=1;         //запуск таймера
a=TIM1_ARRL;                //переменная, отвечающая за частоту мер-
                            цания светодиода
}

void interrupt_init(void)
//Настройка прерываний
{
asm("rim");                 //глобальное разрешение прерываний
}
```

```
__interrupt void TIM1_OVR_UIF(void)        //вектор прерывания таймера
{
TIM1_SR1_UIF=0;          //очистка флага переполнения
PD_ODR_bit.ODR0=! PD_ODR_bit.ODR0;          //инверсия нулевого
                                  бито порта D
TIM1_ARRL=a;
TIM1_CR1_bit.CEN=1;          //запуск таймера
}

__interrupt void EXTI_PB7(void)  //вектор внешнего прерывания порта B
{
 if (TIM1_ARRL>0x10)
  {
 a=a - 0x10;                  //увеличиваем частоту мерцания
  }
 for (i=0;i<30000;i++);          //временная задержка
}

__interrupt void EXTI_PC7(void)    //вектор внешнего прерывания порта C
{
 if (TIM1_ARRL <0xF0)
  {
 a=a + 0x10;                  //уменьшаем частоту мерцания
  }
 for (i=0;i<30000;i++);          //временная задержка

}
```

Пример настройки ШИМ таймера *TIM*1

В данной программе реализуется изменение яркости свечения светодиода по кнопкам с использованием таймера 1, работающего в режиме ШИМ.

```
#include "iostm8s003k3.h"        //подключение заголовочного файла с объ-
                                  явлениями регистров, масок и битов
int i,j;
void interrupt_init(void);          //Объявление подпрограммы настройки
                                  прерываний
#pragma vector=0x06
__interrupt void EXTI_PB7(void);       //Имя вектора внешнего прерывания
#pragma vector=0x07
```

```
__interrupt void EXTI_PC7(void          //Имя вектора внешнего прерывания
void PWM_TIM1_CH2 (void);               //Объявление подпрограммы настройки таймера 1

int main(void)                          //Основная программа
{
PB_DDR_bit.DDR7=0;                      //0-вход; 1-выход
PB_CR1_bit.C17=0;                       //0-дифференциальный вход; 1-вход с подтягивающим резистором
PB_CR2_bit.C27=1;                       //0-прерывания запрещены; 1-прерывания разрешены
PC_DDR_bit.DDR7=0;                      //0-вход; 1-выход
PC_CR1_bit.C17=0;                       //0-дифференциальный вход; 1-вход с подтягивающим резистором
PC_CR2_bit.C27=1;                       //0-прерывания запрещены; 1-прерывания разрешены
interrupt_init();                       //Вызов подпрограммы настройки прерываний
PWM_TIM1_CH2();                         //Вызов подпрограммы настройки ШИМ Таймера1, канал2
  for (;;)
  {
  }
}

//Подпрограмма настройки прерываний
void interrupt_init(void)
{
asm("rim");                             //Глобальное разрешение прерываний
}

//Подпрограмма обработки внешнего прерывания
__interrupt void EXTI_PB7(void)
{
  if (TIM1_CCR2L!=0xFF)                 //Проверка переполнения младшего байта
  {
  TIM1_CCR2L++;                         //Увеличение времени импульса (уменьшение яркости)
  }
    for (i=0;i<300;i++);                //Временная задержка
}
```

```
//Подпрограмма обработки внешнего прерывания
__interrupt void EXTI_PC7(void)
{
 if (TIM1_CCR2L!=0x00)        //Проверка обнуления младшего байта
 {
 TIM1_CCR2L--;                //Уменьшение времени импульса
 }
   for (i=0;i<300;i++);       //Временная задержка
}

//Подпрограмма настройки таймера 1
void PWM_TIM1_CH2 (void)
{
TIM1_PSCRL=0x00;              //предварительный делитель
TIM1_PSCRH=0x0F;              //предварительный делитель
TIM1_BKR=0x80;                //разрешение каналов таймера
CLK_PCKENR2=0xff;             //тактирование таймера1
TIM1_CCMR2=0x68;              //Режим ШИМ1 и разрешение предвари-
                              тельной загрузки
TIM1_CCER1_bit.CC2P=0;        //Активный уровень высокий
TIM1_CCER1_bit.CC2E=1;        //Включение канала 2
TIM1_ARRH=0x00;               //Период ШИМ старший байт
TIM1_ARRL=0xFF;               //Период ШИМ младший байт
TIM1_CCR2H=0x00;              //Время импульса ШИМ старший байт
TIM1_CCR2L=0x01;              //Время импульса ШИМ младший байт
TIM1_CR1_bit.CEN=1;           //Запуск таймера
}
```

4.4. 16-разрядные таймеры (*TIM2, TIM3, TIM5*)

В данной главе описываются ***TIM2*** и ***TIM3***, которые являются идентичными таймерами, с одним отличием, у ***TIM2*** есть три канала, а у таймера 3 их два. ***TIM5***, в отличие от ***TIM2***, имеет два дополнительных регистра, для поддержки соединения и синхронизации между таймерами.

Каждый таймер состоит из 16 битного автоматически перезагружаемого инкрементного счетчика, управляемого программируемым предварительным делителем.

Данные таймеры можно использовать для различных целей, а именно:

- для отсчета времени;
- для измерения длительности импульса входного сигнала (вход захвата);
- для генерирования выходного сигнала различной формы;
- для генерации прерываний, при различных событиях;
- для синхронизации с другими таймерами или внешними сигналами (внешний источник тактирования, сброс, запуск и разрешение) (в устройствах с ***TIM*5**).

Основные характеристики таймеров ***TIM*2**/***TIM*3**:

- 16 битный автоматически перезагружаемый инкрементный счетчик;
- 4 битный программируемый предварительный делитель, позволяющий поделить частоту тактирования счетчика, на любое число в промежутке от 2 до 32768;
- 3 независимых канала для: входа захвата, выхода сравнения, генерации ШИМ (режим выравнивания по краю), выход в режиме одного импульса;
- генерация запросов на прерывания, по следующим событиям: обновление, переполнение счетчика, настройка счетчика (программно), вход захвата, выход сравнения.

Основные характеристики таймера ***TIM*5**:

- 16 битный автоматически перезагружаемый инкрементный счетчик;
- 4 битный программируемый предварительный делитель, позволяющий поделить частоту тактирования счетчика, на любое число в промежутке от 2 до 32768;
- 3 независимых канала для: входа захвата, выхода сравнения, генерации ШИМ (режим выравнивания по краю), выход в режиме одного импульса;
- схема синхронизации, для управления таймером внешними сигналами, для объединения нескольких таймеров;
- вход внешнего запуска ***TIM*1*_ETR*** (объединен с ***TIM*1*)***;
- генерация запросов на прерывания, по следующим событиям: обновление: переполнение счетчика, настройка счетчика (программно), вход захвата, выход сравнения

Предварительный делитель таймеров ***TIM*2**, ***TIM*3**, ***TIM*5** основан на 16 битном счетчике, управляемом с помощью 4 битного регистра (***TIMx_PSCR***). Предварительный делитель позволяет поделить частоту тактирования счетчика, на любое число в промежутке от 2 до 32768.

Частота тактирования счетчика считается следующим образом:

$$f_{CK\_CNT}=f_{CK\_PSC}/2^{(PSCR[3:0])}$$

Значение предварительного делителя загружается через предварительно загружаемый регистр. Новое значение предварительного делителя учитывается в следующем периоде (после следующего события обновления счетчика).

У таймеров ***TIM2***/***TIM3***/***TIM5*** есть 4 источника запросов прерывания:

- прерывание захвата/сравнения 3;
- прерывание захвата/сравнения 2;
- прерывание захвата/сравнения 1;
- прерывание по обновлению;
- прерывание запуска (только у таймера ***TIM5***).

Для использования прерываний, для каждого используемого канала, установите нужные биты ***CC3IE*** и/или ***CC2IE*** и/или ***CC1IE*** в регистре ***TIMx_IER***, чтобы разрешить запросы на прерывания.

Различные запросы на прерывание могут так же быть сгенерированы программно, используя соответствующие биты регистра ***TIMx_EGR***.

4.5. Регистры *TIM2*, *TIM3*, *TIM5*

Регистр контроля 1

Регистр контроля 1 **(*TIMx_CR1*)**:

7	6	5	4	3	2	1	0
ARPE	Зарезервировано			***OPM***	***URS***	***UDIS***	***CEN***
rw				*rw*	*rw*	*rw*	*rw*

- Бит 7 ***ARPE***. Разрешение авто-перезагрузки предварительной загрузки:

  0: регистр ***TIMx_ARR*** не буферизован предварительно загружаемым регистром. Он может быть непосредственно записан;

  1: регистр ***TIMx_ARR*** буферизован предварительно загружаемым регистром.
- Биты 6:4. Зарезервировано.

  Бит 3 OPM. Режим одного импульса:

  0: счетчик не останавливается после события обновления;

  1: счетчик перестает счет на следующем событии обновления (бит ***CEN*** очищается).
- Бит 2 URS. Источник запросов обновления:

  0: запрос на прерывание по обновлению посылается, как только обновятся регистры (переполнение счетчика), если прерывание разрешено;

1: запрос на прерывание по обновлению посылается, только когда счетчик переполнился, если прерывание разрешено.

- Бит 1 UDIS. Запрещение обновления:

  0: событие обновления генерируется, как только произошло переполнение счетчика или программное обновление было сгенерировано, или был сгенерирован аппаратный сброс контроллером ***clock***/***trigger***. В буферизованные регистры загружается их предварительно загружаемое значение;

  1: событие обновления не генерируется, теневые регистры хранят свои значения (***ARR***, ***PSC***, ***CCRi***). Счетчик и предварительный делитель перенастраиваются, если бит ***UG*** установлен.

- Бит 0 CEN. Разрешение счетчика:

  0: счетчик запрещен;

  1: счетчик разрешен.

Регистр контроля 2

Регистр контроля 2 **(*TIM*5*_CR*2)**:

7	6	5	4	3	2	1	0
Зарезервировано	***MMS***[2:0]			Зарезервировано			
	rw	*rw*	*rw*				

- Бит 7. Зарезервировано.
- Биты 6:4 ***MMS***[2:0]. Выбор режима ведущего.

Эти биты выбирают информацию, которая будет отправлена в режиме ведущего таймерам ***TIM*1** и ***TIM*2**, для синхронизации (***TRGO***). Возможны следующие комбинации:

000: сброс – бит ***UG*** в регистре ***TIM*5*_EGR*** используется в качестве запускающего выхода (***TRGO***). Если сброс генерируется входом запуска, то сигнал на ***TRGO*** задерживается и приравнивается к фактическому сбросу;

001: разрешение – сигнал разрешения счетчика используется в качестве выхода запуска (***TRGO***). Он используется для запуска нескольких таймеров одновременно или для контроля окна, в котором ведомый таймер разрешен. Сигнал разрешения счетчика генерируется путем логического сложения между битом ***CEN*** и входом запуска в закрытом режиме. Когда сигнал разрешения счетчика контролируется входом запуска, на ***TRGO*** присутствует задержка, если выбран режим ведущий/ведомый (см. описание битов ***MSM*** в регистре ***TIM*5*_SMCR***);

010: обновление – событие обновления выбрано в качестве выход запуска (***TRGO***);

011: зарезервировано;
100: зарезервировано;
101: зарезервировано;
110: зарезервировано;
111: зарезервировано;

- Биты 3:0 зарезервировано.

Регистр управления режимом ведомого

Регистр управления режимом ведомого (***TIM5_SMCR***):

7	6	5	4	3	2	1	0
MSM	***TS***[2:0]			Зарезервиро-вано	***SMS***[2:0]		
rw	*rw*	*rw*	*rw*		*rw*	*rw*	*rw*

*Примечание: Данный регистр доступен только в таймере **TIM5**.*

- Бит 7 ***MSM***. Режим ведущий/ведомый:

  0: ни какого действия;

  1: событие на входе запуска (***TRGI***) задерживается для улучшения синхронизации между таймерами (через ***TRGO***).

- Бит 6:4 ***TS***[2:0]. Выбор запуска.

  Это битовое поле выбирает вход запуска, который будет использоваться для синхронизации счетчика:

  000: внутренний запуск ***ITR*0** соединяется с ***TRGO*** таймера ***TIM*6**;

  001: зарезервировано;

  010: зарезервировано;

  011: внутренний запуск ***ITR*3** соединяется с ***TRGO*** таймера ***TIM*1**;

  100: TI1 детектор фронта (***TI*1*F_ED***);

  101: вход 1 таймера с фильтром (***TI1FP*1**);

  110: вход 2 таймера с фильтром (***TI2FP*2**);

  111: внешний вход запуска (***ETRF***) (с контактом ***TIM*1*_ETR***). Фильтрацией и полярностью сигнала можно управлять с помощью регистров ***TIM*5*_CCMRi*** и ***TIM*5*_CCERi***.

Примечание: Эти биты должны быть изменены только тогда, когда они не используются, чтобы избежать не правильного обнаружения уровня и перехода.

- Бит 3. Зарезервирован.
- Биты 2:0 ***SMS***[2:0]. Выбор режима ***clock***/***trigger***/***slave***.

Когда внешние сигналы выбраны, активный уровень сигнала на входе запуска (***TRGI***) сопряжен с полярностью, выбранной на внешнем входе:

000: контроллер ***clock***/***trigger*** запрещен – если ***CEN***=1, то предварительный делитель тактируется внутренним источником;

001: зарезервировано;

010: зарезервировано;
011: зарезервировано;
100: запуск в режиме сброса – передний фронт сигнала на выбранном входе запуска (***TRGI***) перенастраивает счетчик и генерирует обновление регистров;
101: закрытый режим – когда запускающий сигнал(***TRGI***) высокого уровня, тактирование счетчика разрешено. Счетчик останавливается (но не сбрасывается), как только уровень сигнала запуска становится низким. И запуск, и остановка счетчика контролируются;
110: режим запуска – счетчик начинает считать по переднему фронту запускающего сигнала ***TRGI*** (но не сбрасывается). Контролируется только старт счетчика;
111: режим внешнего источника тактирования 1 – счетчик тактируется по переднему фронту выбранного сигнала запуска (***TRGI***).

Регистр разрешения прерываний

Регистр разрешения прерываний **(*TIMx_IER*)**:

7	6	5	4	3	2	1	0
Зарезервировано	***TIE***	Зарезервировано		***CC3IE***	***CC2IE***	***CC1IE***	***UIE***
	rw			*rw*	*rw*	*rw*	*rw*

- Бит 7. Зарезервирован.
- Бит 6 ***TIE***. Разрешение прерываний запуска:
  0: прерывания запуска запрещены;
  1: прерывания запуска разрешены.
  *Примечание: В таймерах **TIM2**/**TIM3** этот бит зарезервирован.*
- Биты 5:4. Зарезервированы.
- Бит 3 ***CC3IE***. Разрешение прерываний захвата/сравнения 3:
  0: прерывания ***CC3*** запрещены;
  1: прерывания ***CC3*** разрешены.
- Бит 2 ***CC2IE***. Разрешение прерываний захвата/сравнения 2:
  0: прерывания ***CC2*** запрещены;
  1: прерывания ***CC2*** разрешены.
- Бит 1 ***CC3IE***. Разрешение прерываний захвата/сравнения 1:
  0: прерывания ***CC1*** запрещены;
  1: прерывания ***CC1*** разрешены.
- Бит 0 ***UIE***. Разрешение прерываний обновления:
  0: прерывания обновления запрещены;
  1: прерывания обновления разрешены.

Регистр статуса 1

Регистр статуса 1 (***TIMx_SR*1**):

7	6	5	4	3	2	1	0
Зарезерви-	***TIF***	Зарезерви-	***CC3IF***	***CC2IF***	***CC1IF***	***UIF***	
ровано	*rc_w0*	ровано	*rc_w0*	*rc_w0*	*rc_w0*	*rc_w0*	

- Бит 7. Зарезервирован.
- Бит 6 ***TIF***. Флаг прерывания запуска.

Этот флаг устанавливается аппаратно по событию запуска (активный фронт сигнала обнаруживается на ***TRGI***, в закрытом режиме обнаруживаются оба фронта сигнала). Бит очищается программно:

0: не произошло ни какого события запуска;

1: ожидается прерывание запуска.

*Примечание: В таймерах **TIM2**/**TIM3** этот бит зарезервирован.*

- Биты 5:4. Зарезервированы.
- Бит 3 ***CC3IF***. Флаг прерывания захвата/сравнения 3.

Описание аналогично описанию бита ***CC1IF***.

- Бит 2 ***CC2IF***. Флаг прерывания захвата/сравнения 2.

Описание аналогично описанию бита CC1IF.

- Бит 1 ***CC1IF***. Флаг прерывания захвата/сравнения 1.

Если канал ***CC*1** настроен как выход:

Этот флаг устанавливается аппаратно, когда счетчик достигает сравниваемого значения. Он очищается программно:

0: счетчик не достиг сравниваемого значения;

1: содержание регистра счетчика ***TIMx_CNT*** равно содержанию регистра ***TIMx_CCR*1**.

Если канал ***CC*1** настроен как вход:

Этот бит устанавливается аппаратно по захвату. Очищается программно или считыванием регистра ***TIMx_CCR*1*L***:

0: ни на одном входе захвата не обнаружено;

1: значение счетчика было захвачено в регистр ***TIMx_CCR*1** (на входе ***IC*1** был обнаружен фронт сигнала заданной полярности).

- Бит 0 ***UIF***. Флаг прерывания по обновлению.

Этот бит устанавливается аппаратно по событию обновления:

0: обновления не было;

1: ожидается прерывание по обновлению. Этот бит устанавливается аппаратно, когда произошло обновление регистров:

  - по переполнению, если бит ***UDIS***=0 в регистре ***TIMx_CR*1**;
  - когда ***CNT*** перенастраивается программно, используя бит ***UG*** в регистре ***TIMx_EGR***, если биты ***URS*** и ***UDIS*** в регистре ***TIMx_CR*1** равны нулю.

Регистр статуса 2

Регистр статуса 2 (***TIMx_SR2***):

7	6	5	4	3	2	1	0
Зарезервировано				***CC3OF***	***CC2OF***	***CC1OF***	Зарезервировано
				rc_w0	*rc_w0*	*rc_w0*	

- Биты 7:4 Зарезервированы.
- Бит 3 ***CC3OF***. Флаг перезахвата захвата/сравнения 3.
  Описание аналогично описанию бита ***CC1OF***.
- Бит 2 ***CC2OF***. Флаг перезахвата захвата/сравнения 2.
  Описание аналогично описанию бита ***CC1OF***.
- Бит 1 ***CC1OF***. Флаг перезахвата захвата/сравнения 1.
  Этот флаг устанавливается аппаратно, когда соответствующий канал настроен в режиме входа захвата. Очищается программно, записью в него нуля:
  0: перезахвата не было;
  1: значение счетчика было захвачено в регистр ***TIMx_CCR1***, в то время как флаг ***CC1IF*** уже был установлен.
- Бит 0. Зарезервирован.

Регистр генерации событий

Регистр генерации событий (***TIMx_EGR***):

7	6	5	4	3	2	1	0
Зарезервировано	***TG***	Зарезервировано		***CC3G***	***CC2G***	***CC1G***	***UG***
	w			*w*	*w*	*w*	*w*

- Бит 7 BG. Зарезервирован.
- Бит 6 TG. Генерация события запуска.
  Этот бит устанавливается программно и генерирует событие. Он очищается аппаратно автоматически:
  0: ни какого события;
  1: в регистре ***TIM5_SR1*** устанавливается флаг ***TIF***. Если прерывание разрешено битом ***TIE***, то оно генерируется.
  *Примечание: В таймерах **TIM2**/**TIM3** этот бит зарезервирован.*
- Биты 5:4. Зарезервированы.
- Бит 3 ***CC3G***. Генерация захвата/сравнения 3.
  Аналогично описанию ***CC1G***.
- Бит 2 ***CC2G***. Генерация захвата/сравнения 2.
  Аналогично описанию ***CC1G***.
- Бит 1 ***CC1G***. Генерация захвата/сравнения 1.

Этот бит может быть установлен программно, чтобы сгенерировать событие. Он автоматически очищается аппаратно:

0: ни какого действия;

1: событие захвата/сравнения генерируется на 1 канале:

Если канал ***CC*1** настроен в режиме выхода, то флаг ***CC*1*IF*** устанавливается и формируется соответствующее прерывание, если оно разрешено.

Если канал ***CC*1** настроен в режиме входа, то текущее значение счетчика фиксируется в регистре ***TIM*1_*CCR*1**. Флаг ***CC*1*IF*** устанавливается и, если разрешено, то посылается запрос на соответствующее прерывание. Устанавливается флаг ***CC*1*OF***, если на данный момент уже установлен флаг ***CC*1*IF***.

Бит 0 ***UG***. Генерация обновления.

Этот бит может быть установлен программно и очищается аппаратно (автоматически):

0: ни какого действия;

1: счетчик перенастраивается и генерируется обновление его регистров. Обратите внимание, что предварительный делитель счетчика также очищается.

Регистр режима захвата сравнения 1

Регистр режима захвата сравнения 1 **(*TIMx_CCMR*1):**

Этот канал может быть использован в качестве входа (режим захвата) или в качестве выхода (режим сравнения). Направление канала определяется путем настройки битов ***CC*1*S***. Все остальные биты данного регистра отвечают за различные функции в режиме входа и выхода. Каждый бит ***OCi*** описывает свою функцию, когда канал настроен на выход, ***ICi*** описывает свою функцию, когда канал настроен на вход. Поэтому следует помнить, что один и тот же бит, может иметь различные функции для режимов ввода и вывода.

Канал настроен как выход.

7	6	5	4	3	2	1	0
Зарезервировано	***OC*1*M***[2:0]			***OC*1*PE***	Зарезервировано	***CC*1*S***[1:0]	
	rw	*rw*	*rw*	*rw*		*rw*	*rw*

- Бит 7. Зарезервирован.
- Биты 6:4 ***OC*1*M***[2:0]. Режим выхода сравнения 1.

Этот бит определяет поведение выходного опорного сигнала, ***OC*1*REF***, от которого происходит ***OC*1**. Активный уровень сигнала на ***OC1REF*** – высокий уровень, в то время как, активный уровень на OC1 зависит от бита ***CC*1*P***:

000: замороженный – сравнение между выходным регистром сравнения ***TIMx_CCR*1** и регистром счетчика ***TIMx_CNT*** не имеет ни какого эффекта на выходах;

001: установить на канале 1 активный уровень – сигнал на ***OC*1*REF*** становится высоким, когда регистр счетчика ***TIMx_CNT*** соответствует регистру захвата сравнения 1 (***TIMx_CCR*1**);

010: установить на канале 1 неактивный уровень – сигнал на ***OC*1*REF*** становится низким, когда регистр счетчика ***TIMx_CNT*** соответствует регистру захвата сравнения 1 (***TIMx_CCR*1**);

011: переключение – ***OC*1*REF*** переключается, когда ***TIMx_CNT*=*TIMx_CCR*1**;

100: вынужденный неактивный уровень – сигнал на ***OC*1*REF*** принудительно низкий;

101: вынужденный активный уровень – сигнал на ***OC*1*REF*** принудительно высокий;

110: режим ШИМ 1 – в режиме счета вверх, канал 1 активен (***OC1REF***=1) до тех пор, пока ***TIMx_CNT*<*TIMx_CCR*1**, в противном случае канал неактивен (***OC*1*REF***=0). В режиме счета вниз, канал 1 неактивен (***OC*1*REF***=0) до тех пор, пока ***TIMx_CNT*>*TIMx_CCR*1**, в противном случае канал активен (***OC*1*REF***=1);

111: режим ШИМ 2 – в режиме счета вверх, канал 1 неактивен (***OC*1*REF***=0) до тех пор, пока ***TIMx_CNT*<*TIMx_CCR*1**, в противном случае канал активен (***OC*1*REF***=1).

*Примечание: В режимах ШИМ 1 или 2 уровень сигнала на **OCiREF** изменяется только когда результат сравнения изменяется или когда происходит переключение между режимами «заморожен» и «ШИМ».*

- Бит 3 ***OC*1*PE***. Разрешение предварительной загрузки выхода сравнения 1:

  0: предварительный регистр ***TIMx_CCR*1** запрещен. ***TIMx_CCR*1** может быть записан в любое время. Новое значение учитывается немедленно;

  1: предварительный регистр ***TIMx_CCR*1** разрешен. Доступны операции чтения/записи регистра предварительной загрузки. Предварительно загруженное значение регистра ***TIMx_CCR*1** загружается в теневой регистр по каждому событию обновления.

*Примечание: Для правильной работы, регистры предварительной загрузки должны быть разрешены, когда таймер работает в режиме ШИМ. Это не является обязательным в режиме одного импульса (в регистре **TIM*1*_CR*1** бит **OPM** установлен).*

- Бит 2. Зарезервирован.

• Биты 1:0 ***CC*1*S***[1:0]. Выбор захвата/сравнения 1.

Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

00: канал ***CC*1** настроен как выход;

01: канал ***CC*1** настроен как вход, ***IC*1** отображается на ***TI1FP1***;

10: канал ***CC*1** настроен как вход, ***IC*1** отображается на ***TI2FP1***;

11: канал ***CC*1** настроен как вход, ***IC*1** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM*5*_SMCR***).

*Примечание: Биты **CC*1*S** доступны только для записи, когда канал выключен (**CC*1*E**=0 в регистре **TIMx_CCER1**).*

Канал настроен как вход.

7	6	5	4	3	2	1	0
IC*1*F[3:0]				***IC*1*PSC***[1:0]		***CC*1*S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:4 ***IC*1*F***[3:0]. Фильтр входа захвата 1.

Данное битовое поле определяет частоту $f_{SAMPLING}$, частота используется в качестве образца для входа ***TI1*** и длина цифрового фильтра применяется к ***TI*1**. Цифровой фильтр выполнен из счетчика событий (***N***), необходимых для проверки перехода на выходе:

0000: фильтра нет, $f_{SAMPLING}= f_{MASTER}$;

0001: $f_{SAMPLING}=f_{MASTER}$, N=2;

0010: $f_{SAMPLING}=f_{MASTER}$, N=4;

0011: $f_{SAMPLING}=f_{MASTER}$, N=8;

0100: $f_{SAMPLING}=f_{MASTER}/2$, N=6;

0101: $f_{SAMPLING}=f_{MASTER}/2$, N=8;

0110: $f_{SAMPLING}=f_{MASTER}/4$, N=6;

0111: $f_{SAMPLING}=f_{MASTER}/4$, N=8;

1000: $f_{SAMPLING}=f_{MASTER}/8$, N=6;

1001: $f_{SAMPLING}=f_{MASTER}/8$, N=8;

1010: $f_{SAMPLING}=f_{MASTER}/16$, N=5;

1011: $f_{SAMPLING}=f_{MASTER}/16$, N=6;

1100: $f_{SAMPLING}=f_{MASTER}/16$, N=8;

1101: $f_{SAMPLING}=f_{MASTER}/32$, N=5;

1110: $f_{SAMPLING}=f_{MASTER}/32$, N=6;

1111: $f_{SAMPLING}=f_{MASTER}/32$, N=8.

*Примечание: Даже на каналах, которые имеют комплементарные выходы, это битовое поле не является предварительно загружаемым и не зависит от содержания бита **CCPC** (в регистре **TIM*1*_CR2**).*

• Биты 3:2 ***IC*1*PSC***[1:0]. Предварительный делитель входа захвата 1.

Это битовое поле определяет коэффициент делителя, действующего на входе ***CC*1** (***IC*1**). Предварительный делитель сбрасывается, как только ***CC*1*E***=0 (в регистре ***TIMx_CCER***):

00: нет предварительного делителя, захват производится каждый раз, при обнаружении фронта сигнала на входе захвата;

01: захват осуществляется 1 раз за каждые 2 события;

10: захват осуществляется 1 раз за каждые 4 события;

11: захват осуществляется 1 раз за каждые 8 событий.

*Примечание: Внутренние события счетчика не сбрасываются, когда **IC*1***PSC** изменяется на ходу. В этом случае старые значения используются, пока не произойдет следующий захват. Для того чтобы новые значения были взяты немедленно, необходимо очистить биты **CC*1***E** и заново установить их.*

- Биты 1:0 ***CC*1*S***[1:0]. Выбор захвата/сравнения 1.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC*1** настроен как выход;

  01: канал ***CC*1** настроен как вход, ***IC*1** отображается на ***TI*1*FP*1**;

  10: канал ***CC*1** настроен как вход, ***IC*1** отображается на ***TI*2*FP*1**;

  11: зарезервирован.

  *Примечание: Биты **CC*1***S** доступны только для записи, когда канал выключен (**CC*1***E**=0 в регистре **TIMx_CCER*1**).*

Регистр режима захвата сравнения 2

Регистр режима захвата сравнения 2 **(*TIMx_CCMR*2):**

Канал настроен как выход:

7	6	5	4	3	2	1	0
Зарезервировано	***OC*2*M***[2:0]			***OC*2*PE***	Зарезервировано	***CC*2*S***[1:0]	
	rw	*rw*	*rw*	*rw*		*rw*	*rw*

- Бит 7. Зарезервирован.
- Биты 6:4 ***OC*2*M***[2:0]. Режим выхода сравнения 2.
- Бит 3 ***OC*2*PE***. Разрешение предварительной загрузки выхода сравнения 2.
- Бит 2 Зарезервирован.
- Биты 1:0 ***CC*2*S***[1:0]. Выбор захвата/сравнения 2.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC*2** настроен как выход;

  01: канал ***CC*2** настроен как вод, ***IC*2** отображается на ***TI*2*FP*2**;

  10: канал ***CC*2** настроен как вход, ***IC*2** отображается на ***TI*1*FP*2**;

11: канал ***CC*2** настроен как вход, ***IC*2** отображается на ***TRC***. Этот режим работает, только если внутренний вход запуска выбран через бит ***TS*** (регистр ***TIM*5_*SMCR***).

*Примечание: Биты **CC2S** доступны только для записи, когда канал выключен (**CC2E**=0 в регистре **TIMx_CCER1**).*

Канал настроен как вход:

7	6	5	4	3	2	1	0
IC*2*F[3:0]				***IC*2*PSC***[1:0]		***CC*2*S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:4 ***IC*2*F***[3:0]. Фильтр входа захвата 2.
- Биты 3:2 ***IC*2*PSC***[1:0]. Предварительный делитель входа захвата 2.
- Биты 1:0 ***CC*2*S***[1:0]. Выбор захвата/сравнения 2.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC*2** настроен как выход;

  01: канал ***CC*2** настроен как вход, ***IC*2** отображается на ***TI*2*FP*2**;

  10: канал ***CC*2** настроен как вход, ***IC*2** отображается на ***TI*1*FP*2**;

  11: зарезервирован.

  *Примечание: Биты **CC2S** доступны только для записи, когда канал выключен (**CC2E**=0 в регистре **TIMx_CCER1**).*

Регистр режима захвата сравнения 3

Регистр режима захвата сравнения 3 **(*TIMx_CCMR*3):**

Канал настроен как выход:

7	6	5	4	3	2	1	0
Зарезервировано	***OC*3*M***[2:0]			***OC*3*PE***	Зарезервировано	***CC*3*S***[1:0]	
	rw	*rw*	*rw*	*rw*		*rw*	*rw*

- Бит 7. Зарезервирован.
- Биты 6:4 ***OC*3*M***[2:0]. Режим выхода сравнения 3.
- Бит 3 ***OC*3*PE***. Разрешение предварительной загрузки выхода сравнения 3.
- Бит 2. Зарезервирован.
- Биты 1:0 ***CC*3*S***[1:0]. Выбор захвата/сравнения 3.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC*3** настроен как выход;

  01: канал ***CC*3** настроен как вод, ***IC*3** отображается на ***TI*3*FP*3**;

  10: зарезервирован;

  11: зарезервирован.

*Примечание: Биты **CC3S** доступны только для записи, когда канал выключен (**CC3E**=0 в регистре **TIMx_CCER2**).*

Канал настроен как вход:

7	6	5	4	3	2	1	0
IC3F[3:0]				***IC3PSC***[1:0]		***CC3S***[1:0]	
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

*Примечание: В таймере **TIM3** данный регистр не доступен.*

- Биты 7:4 ***IC3F***[3:0]. Фильтр входа захвата 3.
- Биты 3:2 ***IC3PSC***[1:0]. Предварительный делитель входа захвата 3.
- Биты 1:0 ***CC3S***[1:0]. Выбор захвата/сравнения 3.

  Это битовое поле определяет направление канала (вход/выход), а так же используемый вход:

  00: канал ***CC3*** настроен как выход;

  01: канал ***CC3*** настроен как вход, ***IC3*** отображается на *TI3FP3*;

  10: зарезервирован;

  11: зарезервирован.

  *Примечание: Биты **CC3S** доступны только для записи, когда канал выключен (**CC3E**=0 в регистре **TIMx_CCER2**).*

Регистр разрешения захвата/сравнения 1

Регистр разрешения захвата/сравнения 1 **(*TIMx_CCER1*):**

7	6	5	4	3	2	1	0
Зарезервировано		***CC2P***	***CC2E***	Зарезервировано		***CC1P***	***CC1E***
		rw	***rw***			***rw***	***rw***

- Биты 7:6 Зарезервированы.
- Бит 5 ***CC2P***. Полярность выхода захвата/сравнения 2.

  Описание аналогично описанию бита ***CC1P***.
- Бит 4 ***CC2E***. Разрешение выхода захвата/сравнения 2.

  Описание аналогично описанию бита ***CC1E***.
- Биты 2:3. Зарезервированы.
- Бит 1 ***CC1P***. Полярность выхода захвата/сравнения 1.

  Канал *CC*1 настроен как выход:

  0: активный уровень ***OC1*** высокий;

  1: активный уровень ***OC1*** низкий.

  Канал *CC*1 настроен как вход для захвата:

  0: захват осуществляется по переднему фронту сигнала на ***TI1F*** или ***TI2F***;

  1: захват осуществляется по заднему фронту сигнала на ***TI1F*** или ***TI2F***.

  Бит 0 ***CC1E***. Разрешение выхода захвата/сравнения 1.

Канал *CC*1 настроен как выход:
0: выключен – ***OC*1** не активен;
1: включен – выходной сигнал ***OC*1** на соответствующей выходной ножке.
Канал *CC*1 настроен как вход:
0: захват запрещен;
1: захват разрешен.

Регистр разрешения захвата/сравнения 2

Регистр разрешения захвата/сравнения 2 **(*TIMx_CCER*2):**

7	6	5	4	3	2	1	0
Зарезервировано						***CC*3*P***	***CC*3*E***
						rw	*rw*

*Примечание: Данный регистр не доступен в таймере **TIM*3**.*

- Биты 7:2. Зарезервированы.
- Бит 1 ***CC*3*P***. Полярность выхода захвата/сравнения 3.
  Описание аналогично описанию бита ***CC*1*P***.
- Бит 0 ***CC*3*E***. Разрешение выхода захвата/сравнения 3.
  Описание аналогично описанию бита ***CC*1*E***.

Старший регистр счетчика

Старший регистр счетчика **(*TIMx_CNTRH*):**

7	6	5	4	3	2	1	0
CNT[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CNT***[15:8]: Значение счетчика (Старший байт).

Младший регистр счетчика

Младший регистр счетчика **(*TIMx_CNTRL*):**

7	6	5	4	3	2	1	0
CNT[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CNT***[7:0]. Значение счетчика (Младший байт).

Регистр предварительного делителя

Регистр предварительного делителя **(*TIMx_PSCR*):**

7	6	5	4	3	2	1	0
Зарезервировано				***PSC***[3:0]			
				rw	*rw*	*rw*	*rw*

• Биты 7:4. Зарезервированы.
• Биты 3:0 ***PSC***[3:0]. Значение предварительного делителя.

Значение предварительного делителя делит частоту тактирования ***CK_PSC***. Частота тактирования счетчика f_{CK_CNT} равна f_{CK_PSC} / 2(***PSCR***[3:0]). ***PSCR*** содержит значения, которые загружаются в активный регистр предварительного делителя по каждому событию обновления.

Старший авто-перезагружаемый регистр

Старший авто-перезагружаемый регистр **(*TIMx_ARRH*)**:

7	6	5	4	3	2	1	0
ARR[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***ARR***[15:8]. Авто-перезагружаемое значение (Старший байт).

ARR – это значение, которое будет загружено в авто-перезагружаемый регистр. Счетчик заблокирован, пока авто-перезагружаемое значение равно нулю.

Младший авто-перезагружаемый регистр

Младший авто-перезагружаемый регистр **(*TIMx_ARRL*):**

7	6	5	4	3	2	1	0
ARR[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***ARR***[7:0]. Авто-перезагружаемое значение (Младший байт).

Старший регистр захвата/сравнения 1

Старший регистр захвата/сравнения 1 **(*TIMx_CCR*1*H*):**

7	6	5	4	3	2	1	0
***CCR*1**[15:8]							
rw	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***

• Биты 7:0 ***CCR*1**[15:8]. Значение захвата/сравнения 1 (Старший байт).

Если канал ***CC*1** настроен как выход (биты ***CC*1*S*** в регистре ***TIMx_CCMR*1**):

Значение ***CCR*1** постоянно загружается в регистр захвата/сравнения 1, если предварительная загрузка не разрешена (бит ***OC*1*PE*** в регистре ***TIMx_CCMR*1**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 1, когда обнаружено событие обновления. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*1**.

Если канал ***CC*1** настроен как вход (биты ***CC*1*S*** в регистре ***TIM1_CCMR*1**):
Значение ***CCR*1** является значением счетчика, загружается по последнему событию на входе захвата 1 (***IC*1**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 1

Младший регистр захвата/сравнения 1 **(*TIMx_CCR*1*L*):**

7	6	5	4	3	2	1	0
CCRi[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CCR*1**[7:0]. Значение захвата/сравнения 1 (Младший байт).

Старший регистр захвата/сравнения 2

Старший регистр захвата/сравнения 2 **(*TIMx_CCR*2*H*):**

7	6	5	4	3	2	1	0
***CCR*2**[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Биты 7:0 ***CCR*2**[15:8]. Значение захвата/сравнения 2 (Старший байт).
  Если канал ***CC*2** настроен как выход (биты ***CC*2*S*** в регистре ***TIMx_CCMR*2**), то значение ***CCR*2** постоянно загружается в регистр захвата/сравнения 2, если предварительная загрузка не разрешена (бит ***OC*2*PE*** в регистре ***TIMx_CCMR*2**). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 2, когда обнаружено событие обновления. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC*2**.
  Если канал ***CC*2** настроен как вход (биты ***CC*2*S*** в регистре ***TIMx_CCMR*2**), то значение ***CCR*2** является значением счетчика, загружается по последнему событию на входе захвата 2 (***IC*2**). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 2

Младший регистр захвата/сравнения 2 **(*TIMx_CCR*2*L*):**

7	6	5	4	3	2	1	0
***CCR*2**[7:0]							
rw	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***	***rw***

- Биты 7:0 ***CCR*2**[7:0]. Значение захвата/сравнения 2 (Младший байт).

Старший регистр захвата/сравнения 3

Старший регистр захвата/сравнения 3 (***TIMx_CCR3H***):

7	6	5	4	3	2	1	0
CCR3[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

*Примечание: Данный регистр не доступен в таймере **TIM3***

- Биты 7:0 ***CCR3***[15:8]. Значение захвата/сравнения 3 (Старший байт).
  Если канал ***CC3*** настроен как выход (биты ***CC3S*** в регистре ***TIMx_CCMR3***), то значение ***CCR3*** постоянно загружается в регистр захвата/сравнения 3, если предварительная загрузка не разрешена (бит ***OC3PE*** в регистре ***TIMx_CCMR3***). В противном случае, предварительно загружаемое значение копируется в активный регистр захвата/сравнения 3, когда обнаружено событие обновления. Активный регистр захвата/сравнения содержит значение, которое сравнивается со значением регистра счетчика, ***TIMx_CNT*** и сигнализирует на выходе ***OC3***.
  Если канал ***CC3*** настроен как вход (биты ***CC3S*** в регистре ***TIMx_CCMR3***), то значение ***CCR3*** является значением счетчика, загружается по последнему событию на входе захвата 3 (***IC3***). В данном случае, эти биты доступны только для чтения.

Младший регистр захвата/сравнения 3

Младший регистр захвата/сравнения 3 (***TIMx_CCR3L***):

7	6	5	4	3	2	1	0
CCR3[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

*Примечание: Данный регистр не доступен в таймере **TIM3***

- Биты 7:0 ***CCR3***[7:0]. Значение захвата/сравнения 3 (Младший байт).

4.6. Примеры настройки таймера 2

Пример настройки ШИМ на таймере TIM2 канал 1 (канал соединен с четвертым выводом порта D)

```
#include "iostm8s003k3.h"        //подключение заголовочного файла с объявлениями регистров, масок и битов
void PWM_TIM2_CH1 (void);        //объявление подпрограммы настройки таймера
int main( void )                 //основная программа
```

```
{
PWM_TIM2_CH1();              //настройка ШИМ TIM2 канал 1 (PD4)

  for(;;)                    //бесконечный цикл
  {
  }
}

void PWM_TIM2_CH1 (void)     //подпрограмма настройки таймера 2
{
TIM2_PSCR=0x00;              //предварительный делитель
CLK_PCKENR2=0xff;            //тактирование таймера1
TIM2_CCMR1=0x68;             //режим ШИМ 1 и разрешение предвари-
                             тельной загрузки
TIM2_CCER1_bit.CC1P=0;       //активный уровень высокий
TIM2_CCER1_bit.CC1E=1;       //включение канала 1
TIM2_ARRH=0x01;              //период ШИМ старший байт
TIM2_ARRL=0xFF;              //период ШИМ младший байт
TIM2_CCR1H=0x00;             //время импульса ШИМ старший байт
TIM2_CCR1L=0xFF;             //время импульса ШИМ младший байт
TIM2_CR1_bit.CEN=1;          //запуск таймера
}
```

4.7. Базовые 8-разрядные таймеры (*TIM4*, *TIM6*)

Таймеры ***TIM*4** и ***TIM*6** состоят из 8 битных автоматически перезагружаемых инкрементных счетчиков, управляемых программируемым предварительным делителем. Они могут использоваться для отсчета времени, с генерированием прерываний по переполнению.

Таймер ***TIM*6** имеет контроллер ***clock*/*trigger***, для синхронизации и объединения с другими таймерами.

Основные характеристики таймеров ***TIM*4**:

- 8 битный авто-перезагружаемый инкрементный счетчик;
- 3 битный программируемый предварительный делитель, который позволяет поделить частоту тактирования таймера на 1, 2, 4, 8, 16, 32, 64 и 128;
- генерирование прерываний по обновлению счетчика и переполнению счетчика.

Основные характеристики таймера ***TIM*6**:

- 8 битный авто-перезагружаемый инкрементный счетчик;

- 3 битный программируемый предварительный делитель, который позволяет поделить частоту тактирования таймера на 1, 2, 4, 8, 16, 32, 64 и 128;
- схема синхронизации, для контроля таймера, с помощью внешних сигналов, а так же для объединения нескольких таймеров;
- генерирование прерываний по обновлению счетчика, переполнению счетчика и по входу запуска.

Источником тактирования для таймеров является внутренний источник (f_{MASTER}). Он соединяется, непосредственно с источником ***CK_PSC***, который питается от предварительного делителя, управляющего частотой счетчика ***CK_CNT***.

Предварительный делитель основан на 7 битном счетчике, контролируемом с помощью 3 битов (в регистре ***TIMx_PSCR***). Данный регистр буферизован, таким образом он может изменяться на ходу. Он может поделить частоту тактирования счетчика на любую степень числа 2, в промежутке от 1 до 128 (1, 2, 4, 8, 16, 32, 64, 128).

Частоты тактирования рассчитывается следующим образом:

$$f_{CK\_CNT}=f_{CK\_PSC}/2^{(PSCR[2:0])}.$$

Значение предварительного делителя загружается через регистр предварительной загрузки. Операции считывания регистра ***TIMx_PSCR*** дают доступ к регистрам предварительной загрузки, таким образом, не требуются специальные меры для их считывания.

4.8. Регистры таймеров *TIM4, TIM6*

Регистр контроля 1

Регистр контроля 1 **(*TIMx_CR1*)**.

7	6	5	4	3	2	1	0
ARPE	Зарезервировано			***OPM***	***URS***	***UDIS***	***CEN***
rw				*rw*	*rw*	*rw*	*rw*

- Бит 7 ***ARPE***. Разрешение предварительной загрузки автоперезагрузки:

  0: регистр ***TIM4_ARR*** не буферизован регистром предварительной загрузки. Доступен только для чтения;

  1: регистр ***TIM4_ARR*** буферизован регистром предварительной загрузки.
- Биты 6:4. Зарезервированы.
- Бит 3 ***OPM***. Режим одного импульса:

  0: счетчик не останавливается на событии обновления;

1: счетчик прекращает счет на следующем событии обновления (бит ***CEN*** очищается).

- Бит 2 ***URS***. Источник запроса обновления:

  0: когда разрешен, как только регистры были обновлены посылается запрос на прерывание по обновлению;

  1: когда разрешен, только когда счетчик переполнился посылается запрос на прерывание по обновлению (в режиме счета вверх или вниз).

- Бит 1 ***UDIS***. Запрещение обновления:

  0: ***UEV*** генерируется, как только было обнаружено переполнение счетчика или было сгенерировано программное обновление. Буферизованные регистры загружаются их значением предварительной загрузки;

  1: ***UEV*** не генерируется, теневые регистры хранят их значения (***ARR***, ***PSC***). Счетчик и предварительный делитель перенастраиваются, если бит ***UG*** установлен.

- Бит 0 ***CEN***. Разрешение счетчика:

  0: счетчик запрещен;

  1: счетчик разрешен.

Регистр контроля 2

Регистр контроля 2 **(*TIM*6*_CR*2)**.

7	6	5	4	3	2	1	0
Зарезервировано	***MMS***[2:0]			Зарезервировано			
	rw	*rw*	*rw*				

*Примечание: Данный регистр не доступен в таймере **TIM**4.*

- Бит 7. Зарезервирован.
- Биты 6:4 ***MMS***[2:0]. Выбор режима ведущего:

  000: сброс – бит ***UG*** в регистре ***TIM*6*_EGR*** используется как выход запуска (***TRGO***). Если запускающий вход формирует сигнал сброса, то сигнал на ***TRGO*** задерживается и сравнивается с фактическим сигналом сброса;

  001: разрешение – сигнал разрешения счетчика используется в качестве запускающего выхода (***TRGO***). Он используется для запуска нескольких таймеров одновременно или для контроля окна, в котором ведомый таймер разрешен. Сигнал разрешения счетчика генерируется путем логического сложения контрольного бита ***CEN*** и входа запуска, когда выбран закрытый режим. Когда сигнал разрешения счетчика контролируется входом запуска, на ***TRGO*** присутствует задержка, если только не выбран режим ведущий/ведомый;

010: обновление – событие обновления выбрано в качестве выхода запуска (***TRGO***);
011: зарезервировано;
100: зарезервировано;
101: зарезервировано;
110: зарезервировано;
111: зарезервировано.
Бит 3:0. Зарезервированы.

Регистр контроля режима ведомый

Регистр контроля режима ведомый **(*TIM*6_*SMCR*)**.

7	6	5	4	3	2	1	0
MSM	***TS***[2:0]			Зарезерви-рovано	***SMS***[2:0]		
rw	*rw*	*rw*	*rw*		*rw*	*rw*	*rw*

*Примечание: Данный регистр не доступен в таймере **TIM4**.*

• Бит 7 ***MSM***. Режим ведущий/ведомый:
0: ни какого действия;
1: действие события на входе запуска (***TRGI***) задерживается для улучшения синхронизации между таймерами (через ***TRGO***).

• Бит 6:4 ***TS***[2:0]. Выбор запуска.

Это битовое поле выбирает вход запуска, который будет использоваться для синхронизации счетчика:
000: зарезервировано;
001: зарезервировано;
010: внутренний запуск ***ITR2*** соединяется с ***TRGO*** таймера ***TIM5***;
011: внутренний запуск ***ITR3*** соединяется с ***TRGO*** таймера ***TIM1***;
100: зарезервировано;
101: зарезервировано;
110: зарезервировано;
111: зарезервировано.

Примечание: Эти биты должны изменяться только тогда, когда они не используются, чтобы избежать не правильного обнаружения уровня и перехода.

• Бит 3. Зарезервирован.

• Биты 2:0 ***SMS***[2:0]. Выбор режима ***clock***/***trigger***/***slave***.

Когда внешние сигналы выбраны, активный уровень сигнала запуска (***TRGI***) сопряжен с полярностью, выбранной на внешнем входе:
000: контроллер ***clock***/***trigger*** запрещен. Если ***CEN***=1, то предварительный делитель тактируется внутренним источником;
001: зарезервировано;
010: зарезервировано;

011: зарезервировано;
100: запуск в режиме сброса – передний фронт выбранного сигнала запуска (***TRGI***) перенастраивает счетчик и генерирует обновление регистров;
101: закрытый режим. Если сигнал запуска (***TRGI***) высокого уровня, то тактирование счетчика разрешено. Счетчик останавливается (но не сбрасывается), как только уровень сигнала запуска становится низким;
110: режим запуска. Счетчик начинает считать по переднему фронту сигнала запуска ***TRGI*** (но не сбрасывается). Контролируется только старт счетчика;
111: режим внешнего источника тактирования 1. Счетчик тактируется по переднему фронту выбранного сигнала запуска (***TRGI***).

Регистр разрешения прерываний

Регистр разрешения прерываний **(*TIMx_IER*)**.

7	6	5	4	3	2	1	0
Зарезервировано	***TIE***	Зарезервировано					***UIE***
	rw						*rw*

- Бит 7. Зарезервирован.
- Бит 6 ***TIE***. Разрешение прерываний запуска:
  0: прерывания запуска запрещены;
  1: прерывания запуска разрешены.
  *Примечание: В таймере **TIM4** данный бит зарезервирован.*
- Биты 5:1. Зарезервированы.
  Бит 0 ***UIE***. Разрешение прерываний по обновлению:
  0: прерывания по обновлению запрещены;
  1: прерывания по обновлению разрешены.

Регистр статуса 1

Регистр статуса 1 **(*TIMx_SR*)**.

7	6	5	4	3	2	1	0
Зарезервировано	***TIF***	Зарезервировано					***UIF***
	rw						*rw*

- Бит 7. Зарезервирован.
- Бит 6 ***TIF***. Флаг прерываний запуска.

Данный флаг устанавливается аппаратно, по событию запуска. Очищается программно:
0: событий запуска не обнаружено;

1: ожидается прерывание запуска.

*Примечание: В таймере **TIM4** данный бит зарезервирован.*

- Биты 5:1. Зарезервированы.
- Бит 0 ***UIF***. Флаг прерываний по обновлению.

Данный флаг устанавливается аппаратно, по событию обновления. Очищается программно:

0: обновление не обнаружено;

1: ожидается прерывание по обновлению. Данный бит устанавливается аппаратно при обновлении регистров:

- По переполнению, если бит ***UDIS***=0 в регистре ***TIM4_CR1***;
- Когда ***CNT*** перенастроен программно, используя бит UG в регистре ***TIM4_EGR***, если биты ***URS*** и ***UDIS*** в регистре ***TIM4_CR1*** равны 0.

Регистр генерирования событий

Регистр генерирования событий (***TIMx_EGR***).

7	6	5	4	3	2	1	0
Зарезервировано	***TG***	Зарезервировано					***UG***
	rw						*rw*

- Бит 7. Зарезервирован.
- Бит 6 ***TG***. Генерация события запуска.

Этот бит устанавливается программно и генерирует событие. Очищается автоматически аппаратно:

0: ни какого действия.

1: устанавливается флаг ***TIF*** в регистре ***TIM6_SR***. Прерывание генерируется, если разрешено битом TIE.

*Примечание: В таймере **TIM4** данный бит зарезервирован.*

- Биты 5:1. Зарезервированы.
- Бит 0 ***UG***. Генерация события обновления.

Этот бит устанавливается программно, очищается автоматически:

0: ни какого действия.

1: перенастройка счетчика и генерация обновления регистров. Счетчик предварительного делителя так же очищается.

Регистр счетчика

Регистр счетчика (***TIMx_CNTR***).

7	6	5	4	3	2	1	0
CNT[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***CNT***[7:0]. Значение счетчика.

Регистр предварительного делителя

Регистр предварительного делителя **(*TIMx_PSCR*):**

7	6	5	4	3	2	1	0
Зарезервировано					***PSC***[2:0]		
					rw	*rw*	*rw*

• Биты 7:3. Зарезервированы.

• Биты 2:0 ***PSC***[2:0]. Значение предварительного делителя.

Предварительный делитель делит частоту тактирования ***CK_PSC***. Частота тактирования f_{CK_CNT} равна f_{CK_PSC} / 2(***PSC***[2:0]). ***PSC*** содержит значение, которое загружается в активный регистр предварительного делителя, по каждому ***UEV***.

Авто-перезагружаемый регистр

Авто-перезагружаемый регистр **(*TIMx_ARR*)**.

7	6	5	4	3	2	1	0
ARR[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***ARR***[7:0]. Авто-перезагружаемое значение.

Глава 5. Аналогово-цифровой преобразователь (АЦП)

5.1. Основные характеристики АЦП. Функционирование АЦП

Микроконтроллеры STM8S имеют два 10-битных аналогово-цифровых преобразователя последовательного приближения: АЦП 1 и АЦП 2. Они имеют до 16 мультиплексированных входных каналов. Преобразование на каждом канале может осуществляться в одиночном режиме или в непрерывном режиме.

Основные характеристики АЦП:

Данные характеристики относятся к АЦП1 и АЦП2:

- разрешение – 10 бит;
- одиночный и непрерывный режимы преобразования;
- программируемый предварительный делитель: f_{MASTER} делится на число в промежутке от 2 до 18;
- внешний запуск, используя внешние прерывания (***ADC_ETR***) или запуск таймера (***TRGO***);
- аналоговое масштабирование (в устройствах с V_{REF});
- генерация прерываний по окончанию преобразования;
- величина входного сигнала: $V_{SSA} \le V_{IN} \le V_{DDA}$.
- буферизованный непрерывный режим преобразования (АЦП 1);
- режим сканирования для одиночного и непрерывного преобразования (АЦП 1);
- аналоговый контроль с верхними и нижними порогами (АЦП 1);
- генерация прерываний по событию на аналоговом контроле (АЦП 1);

В таблице 5.1. приведено описание выводов блока АЦП микроконтроллеров ***STM8S***.

Таблица 5.1

Описание выводов АЦП

Название	Тип сигнала	Замечания
V_{DDA}	Напряжение питания	Напряжение питания. Этот вход связан с V_{DD} в устройствах, не имеющих внешнего вывода V_{DDA}.
V_{SSA}	Общий (GND)	Общий. Этот вход связан с V_{SS} в устройствах, не имеющих внешнего вывода V_{SSA}.
V_{REF-}	Отрицательное опорное напряжение	Наименьшее/отрицательное опорное напряжение для АЦП, в промежутке от V_{SSA} до (V_{SSA} + 500мВ). Этот вход связан с V_{SSA} в устройствах, не имеющих внешнего вывода V_{REF-}.

Окончание табл. 5.1

V_{REF+}	Положительное опорное напряжение	Наибольшее/положительное опорное напряжение для АЦП, в промежутке от 2.75В до V_{DDA}. Этот вход связан с V_{DDA} в устройствах, не имеющих внешнего вывода V_{REF+}.
AIN[15:0]	Аналоговые входные сигналы	До 16 аналоговых входных каналов, которые преобразуются с помощью АЦП по одному.
ADC_ETR	Цифровые входные сигналы	Внешний запуск

На рис. 5.1 и рис. 5.2 представлены функциональные схемы АЦП 1 и АЦП 2, соответственно.

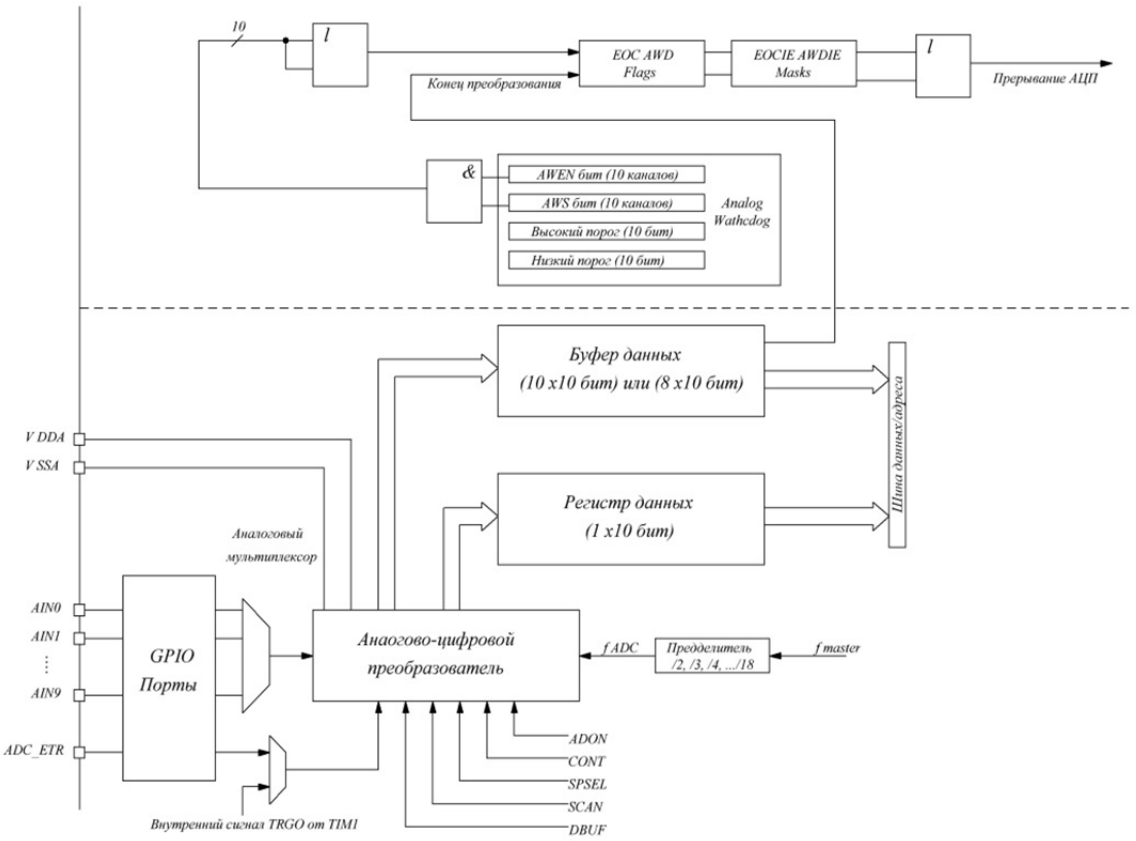

Рис. 5.1. Функциональная схема АЦП 1

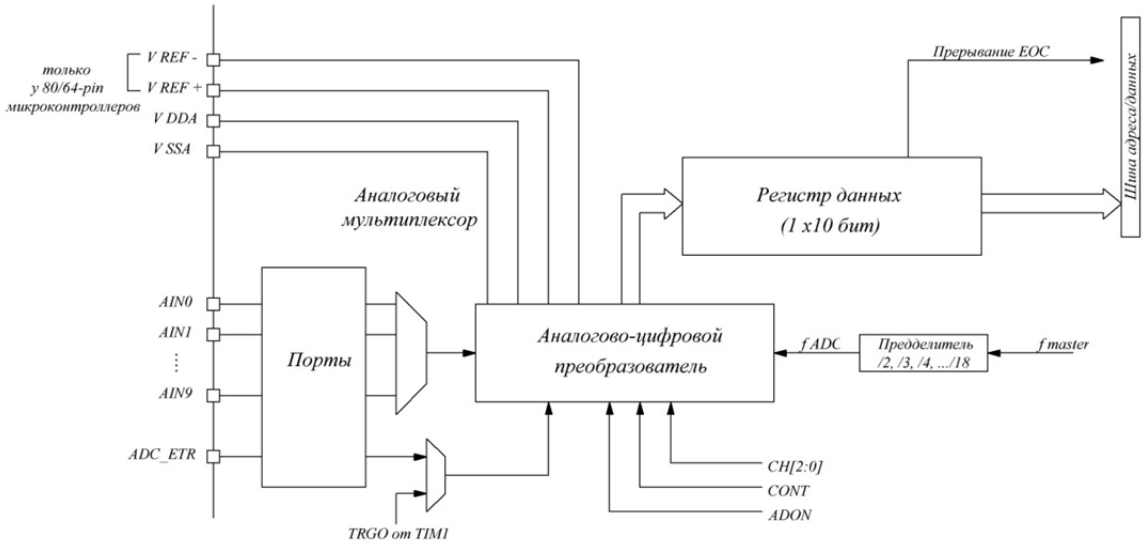

Рис. 5.2. Функциональная схема АЦП 2

Управление включением/выключением АЦП

АЦП может быть включен, установкой бита ***ADON*** в регистре ***ADC_CR*1**. Когда бит ***ADON*** установлен в первый раз, это выводит АЦП из режима с отключенным питанием. Чтобы начать преобразования, необходимо установить бит ***ADON*** в регистре ***ADC_CR*1** второй раз.

По окончанию преобразования АЦП не отключается от питания, и достаточно установить бит ***ADON*** один раз, для запуска следующего преобразования.

Если АЦП на протяжении большого промежутка времени не используется, то рекомендуется переключить его в режим с отключенным питанием, очистив бит ***ADON***.

Когда на АЦП подано напряжение питания, выходной каскад выбранного канала отключен, поэтому рекомендуется выбирать канал преобразования, перед тем как подавать напряжение питания на АЦП.

Тактирование АЦП

Источником тактирования АЦП является предварительно поделенная частота f_{MASTER}. Предварительное деление частоты зависит от битов ***SPSEL***[2:0] в регистре ***ADC_CR*1**.

Выбор канала

АЦП микроконтроллера ***STM8S*** может иметь до 16 внешних входных каналов. Число каналов зависит от размеров корпуса микроконтроллера.

Если выбор канала изменяется во время преобразования, то текущее преобразование сбрасывается и на АЦП посылается новый стартовый импульс.

Режимы преобразования

АЦП поддерживает пять режимов преобразования: одиночный режим, непрерывный режим, буферизованный непрерывный режим, режим одиночного сканирования, режим непрерывного сканирования.

В режиме одиночного преобразования АЦП выполняет одно преобразование на канале, выбранном битами ***CH***[3:0] в регистре ***ADC_CSR***. Данный режим запускается, установкой бита ***ADON*** в регистре ***ADC_CR1***, до тех пор, пока бит ***CONT***=0.

После завершения преобразования, преобразованные данные хранятся в регистре ***ADC_DR***, флаг ***EOC*** (*End of Conversion* – конец преобразования) устанавливается, и генерируется прерывание, если бит ***EOCIE*** установлен.

В режиме непрерывного преобразования АЦП запускает следующее преобразование, как только закончено предыдущее. Этот режим запускается установкой бита ***ADDON*** в регистре ***ADC_CR1***, когда бит ***CONT***=1.

- Если буферизация не включена (бит ***DBUF***=0 в регистре ***ADC_CR3***), результат преобразования хранится в регистре ***ADC_DR***, и флаг ***EOC*** устанавливается. Генерируется прерывание, если бит ***EOCIE*** установлен. Новое преобразование запускается автоматически.
- Если буферизация включена (бит ***DBUF***=1), буфер данных заполняется результатами 8 или 10 последовательных преобразований, выполненных на одном канале. Когда буфер заполнен, флаг ***EOC*** устанавливается, генерируется прерывание, если бит ***EOCIE*** установлен. Новый набор из 8 или 10 преобразований запускается автоматически. Флаг ***OVR*** устанавливается, если один из буферных регистров данных был перезаписан до того как он был считан.

Чтобы остановить непрерывное преобразование, необходимо сбросить бит ***CONT*** или сбросить бит ***ADON***, чтобы отключить питание АЦП.

Режим однократного сканирования используется для преобразования последовательности аналоговых каналов от ***AIN0*** до ***AINn***, где «***n***» это номер канала, определяемый битами ***CH***[3:0] в регистре ***ADC_CSR***. Во время преобразования последовательности биты ***CH***[3:0] обновляются аппаратно и содержат номер преобразуемого канала.

Режим однократного сканирования запускается установкой бита ***ADON***, в то время как бит ***SCAN*** уже установлен, а бит ***CONT*** сброшен.

*Примечание: В режиме сканирования не возможно использовать каналы **AIN0-AINn** в режиме выхода, так как выходной каскад каждого канала выключен, когда он был выбран аналоговым мультиплексором.*

Одиночное преобразование выполняется начиная с ***AIN0*** и результаты хранятся в буферных регистрах ***ADC_DBxR***. Когда последний канал (канал «***n***») был преобразован, при этом устанавливается флаг ***EOC***, и генерируется прерывание (если бит ***EOCIE*** установлен).

Не следует очищать бит ***SCAN***, пока выполняется последовательность преобразований. Режим однократного сканирования может быть остановлен немедленно, очищением бита ***ADON***.

Для того чтобы запустить новое ***SCAN*** преобразование, необходимо очистить бит ***EOC*** и установить бит ***ADON*** в регистре ***ADC_CR1***.

Режим непрерывного сканирования идентичен режиму однократного сканирования, за исключением того, что каждый раз, когда последний канал был преобразован, автоматически запускается новое сканирование с канала 0 до канала «***n***».

Режим непрерывного сканирования запускается установкой бита ***ADON***, в то время как бит ***CONT*** установлен.

Не следует очищать бит ***SCAN***, пока выполняется последовательность преобразований. Режим непрерывного сканирования может быть остановлен немедленно, очищением бита ***ADON***. Если во время преобразования бит ***CONT*** будет сброшен, то преобразование закончится, когда будет преобразован последний канал.

Правильным способом очистки флага ***EOC***, в режиме непрерывного сканирования, является загрузка байта в регистр ***ADC_CSR*** из переменной ***RAM***, очистка флага ***EOC*** и перезагрузка номера последнего канала для последовательности преобразования.

Флаг ошибки «*overrun*»

Флаг ошибки ***OVR*** устанавливается аппаратно, в буферизованном непрерывном режиме, режиме одиночного сканирования, режиме непрерывного сканирования. Он указывает на то, что один из десяти буферных регистров был перезаписан новым значением, до того как предыдущее значение было считано. В таком случае, рекомендуется запустить новое преобразование.

Примечание: *Установка бита **ADON** автоматически очищает флаг **OVR**.*

Контроль аналоговых значений

Контроль аналоговых значений разрешается, в режиме однократного преобразования и в режиме не буферизованного непрерывного преобразования, установкой бита ***AWDEN*** в регистре ***ADC_CSR***.

Флаг контроля аналоговых значений ***AWD*** устанавливается в том случае, если напряжение, преобразуемое АЦП, ниже нижнего порога иле выше верхнего порога. Эти пороги настраиваются в 10 битных регистрах ***ADC_HTR*** и ***ADC_LTR***. Прерывание разрешается установкой бита ***AWDIE*** в регистре ***ADC_CSR***.

Для режима сканирования, контроль аналоговых значений, может быть разрешен на выбранных каналах, используя биты ***AWENx*** в регистрах ***ADC_AWCRH*** и ***ADC_AWCRL***. Состояние контроля для каждого канала получают, считывая биты ***AWSx*** в регистрах ***ADC_AWSRH*** и ***ADC_AWSRL***. Установка любого флага ***AWS*** устанавливает флаг ***AWD***. В зависимости от бита ***AWDIE***, разрешающего прерывания, прерывание генерируется в конце последовательности ***SCAN***. В подпрограмме прерываний необходимо очистить флаг ***AWS*** и ***AWD*** в регистре ***ADC_CSR***.

Для буферизованного непрерывного режима, контроль аналоговых значений может быть доступен для выбранных буферов, и управляется так, как описано для режима сканирования, с той разницей, что буферы содержат результаты непрерывных преобразований, выполняемых на одном канале.

Примечание: Для оптимизации задержки прерываний по контролю аналоговых значений в режиме сканирования и буферизованного непрерывного преобразования, рекомендуется использовать последний канал в последовательности преобразований.

Преобразования по внешнему запуску

Преобразование может быть запущено передним фронтом сигнала на ножке ***ADC_ETR*** или событием ***TRGO*** таймера. Если контрольный бит ***EXTTRIG*** установлен, то любое внешнее событие может быть использовано для запуска преобразования. Биты ***EXTSEL***[1:0] используются для выбора двух возможных источников событий, которые могут запустить преобразование.

Алгоритм для использования режима внешнего запуска:

1. АЦП находится в выключенном состоянии (***ADON***=0) и бит ***EOC*** сброшен;

2. Выбор источника запуска (***EXTSEL*** [1:0]).

3. Выбор режима внешнего запуска ***EXTTRIG***=1 (рекомендуется использовать команду установки бита, чтобы не изменять остальные биты в данном регистре);

4. Если источник запуска находится в активном состоянии (высокий уровень), АЦП включается. По этой причине, сначала проверьте, выключен ли АЦП (бит ***ADON***=0), и только потом включайте АЦП (бит ***ADON***=1).

5. Подождите в течении времени стабилизации (t_{STAB}). Если сигнал внешнего запуска будет обнаружен до того как пройдет t_{STAB}, это не вызовет ни какого эффекта.

6. Преобразование начнется, когда будет обнаружено внешнее запускающее событие.

Примечание: Если выбран режим запуска таймером (источником запуска является событие таймера, а не внешний вывод), то рекомендуется запускать таймер, когда АЦП полностью установилось, а останавливать таймер до отключения АЦП.

Временные диаграммы работы АЦП

Как показано на рис. 5.3, после включения, АЦП необходимо время стабилизации t_{STAB} (равное времени одного преобразования t_{CONV}) для того чтобы его преобразования были точными. Для последующих преобразований нет задержки стабилизации и бит ***ADON*** нужно устанавливать один раз. Время преобразования АЦП занимает 14 тактовых циклов. После преобразования устанавливается флаг ***EOC*** и, результаты преобразования загружаются в регистр АЦП. На рис. 5.3 и 5.4 представленны временные диаграммы работы АЦП в режимах одиночного и непрерывного преобразований, соответственно.

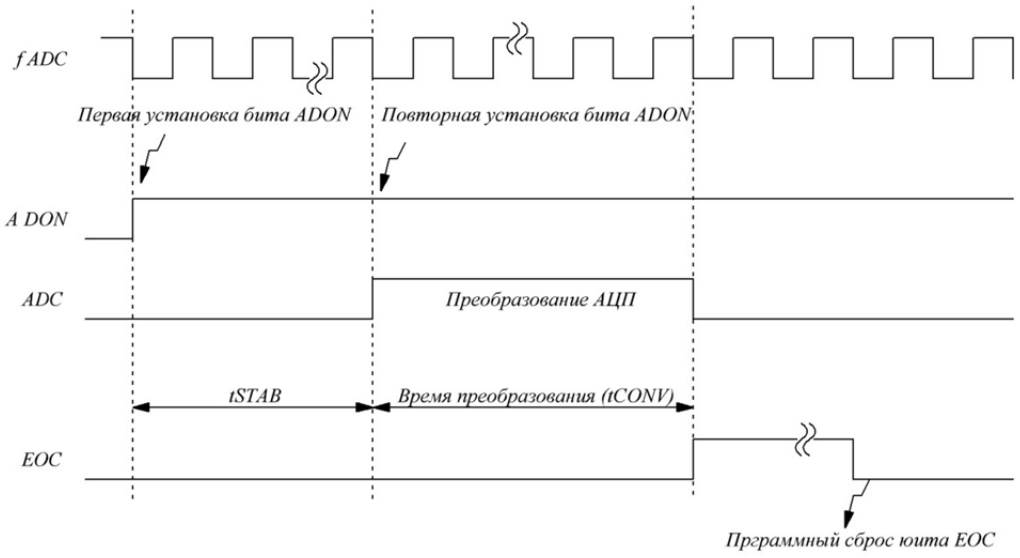

*Рис. 5.3. Временная диаграмма работы АЦП в режиме одиночного преобразования (**CONT**=0)*

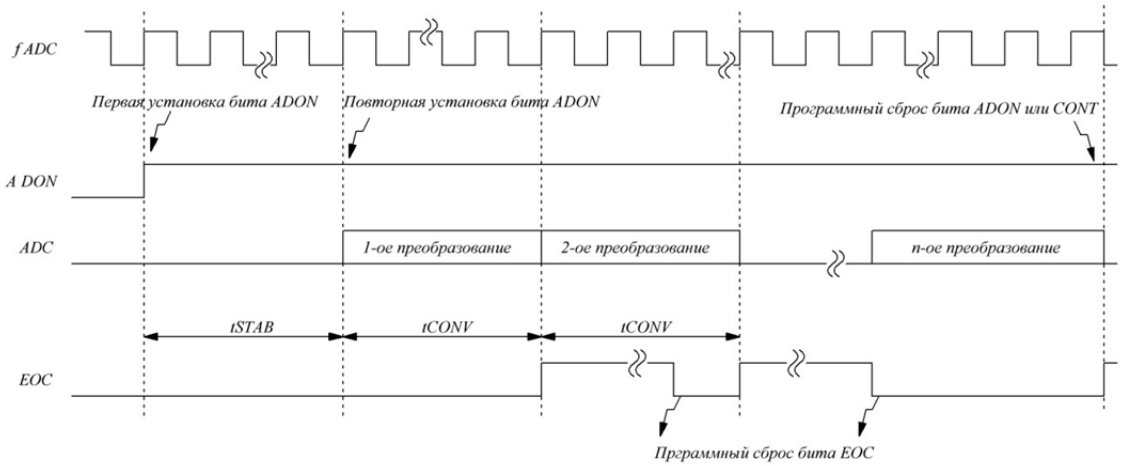

*Рис. 5.4. Временная диаграмма работы АЦП в режиме непрерывного преобразования (**CONT**=1)*

Выравнивание данных

Бит ***ALIGN*** в регистре ***ADC_CR2*** выбирает способ выравнивания результатов преобразования. Существуют следующие виды выравнивания:

• выравнивание по правому краю (рис. 5.5). В этом случе 8 младших бит записываются в регистр ***ADC_DL***, затем оставшиеся старшие биты записываются в регистр ***ADC_DH***. Сначала считывается младший байт, а за ним старший байт.

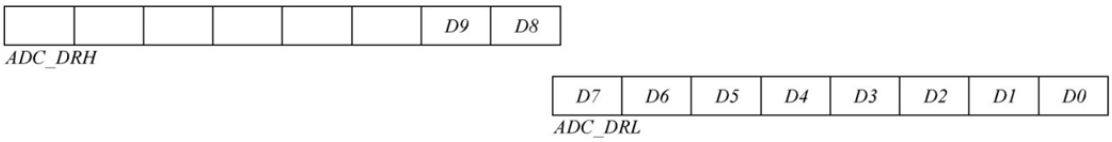

Рис. 5.5. Выравнивание данных по правому краю

Выравнивание по левому краю (рис. 5.6). В этом случае 8 старших бит записывается в регистр ***ADC_DH***, затем оставшиеся младшие биты записываются в регистр ***ADC_DL***. Сначала должен быть считан старший байт, а за ним младший байт.

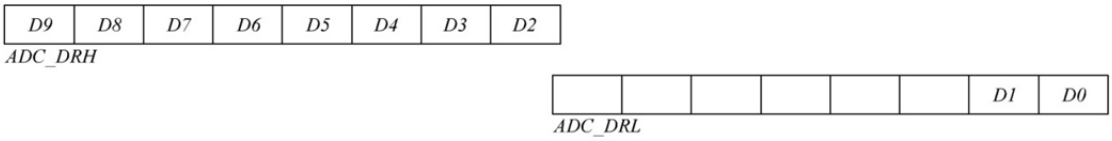

Рис. 5.6. Выравнивание данных по левому краю

5.2. Регистры АЦП

Старший буферный регистр данных АЦП

Старший буферный регистр данных АЦП (***ADC_DBxRH***), (***x***=0..7 или 0..9).

7	6	5	4	3	2	1	0
DBH[7:0]							
r	*r*	*r*	*r*	*r*	*r*	*r*	*r*

Примечание: Буферный регистр не доступен в АЦП2.

- Биты 7:0 ***DBH***[7:0]. Старшие биты данных.

Эти биты устанавливаются/сбрасываются аппаратно и доступны только для чтения. Когда АЦП работает в буферизованном непрерывном режиме или в режиме сканирования в этом регистре содержится старшие биты результатов преобразования. Выравнивание данных, по левому или по правому краю, зависит от бита ***ALIGN***.

Младший буферный регистр данных АЦП

Младший буферный регистр данных АЦП (***ADC_DBxRL***), **(x=0..7 или 0..9).**

7	6	5	4	3	2	1	0
DBL[7:0]							
r	*r*	*r*	*r*	*r*	*r*	*r*	*r*

Примечание: Буферный регистр не доступен в АЦП2.

- Биты 7:0 ***DBL***[7:0]: Младшие биты данных.

Эти биты устанавливаются/сбрасываются аппаратно и доступны только для чтения. Когда АЦП работает в буферизованном непрерывном режиме или в режиме сканирования, в этом регистре содержится младшие биты результатов преобразования. Выравнивание данных, по левому краю или по правому, зависит от бита ***ALIGN***.

Регистр управления/состояния

Регистр управления/состояния (***ADC_CSR***).

7	6	5	4	3	2	1	0
EOC	***AWD***	***EOCIE***	***AWDIE***	***CH*[3:0]**			
rw	*rc_w0*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

- Бит 7 ***EOC***. Завершение преобразования:
  0: преобразование не завершено;
  1: преобразование завершено.
- Бит 6 ***AWD***. Флаг контроля аналоговых значений:

0: нет события контроля аналоговых значений;
1: событие контроля аналоговых значений произошло.
Примечание: Данный бит недоступен в АЦП2.

- Бит 5 ***EOCIE***. Разрешение прерываний по завершению преобразования:
  0: прерывание по окончанию преобразования запрещено;
  1: прерывание по окончанию преобразования разрешено.
- Бит 4 ***AWDIE***. Разрешение прерываний от контроля аналоговых значений:
  0: прерывание от ***AWD*** запрещено;
  1: прерывание от ***AWD*** разрешено.
  Примечание: Данный бит недоступен в АЦП2.
- Биты 3:0 ***CH***[3:0]. Биты выбора канала:
  0000: канал ***AIN0***;
  0001: канал ***AIN1***;
  0010: канал ***AIN2***;
  0011: канал ***AIN3***;
  0100: канал ***AIN4***;
  0101: канал ***AIN5***;
  0110: канал ***AIN6***;
  0111: канал ***AIN7***;
  1000: канал ***AIN8***;
  1001: канал ***AIN9***;
  1010: канал ***AIN10***;
  1011: канал ***AIN11***;
  1100: канал ***AIN12***;
  1101: канал ***AIN13***;
  1110: канал ***AIN14***;
  1111: канал ***AIN15***.

Регистр конфигураций 1

Регистр конфигураций 1 **(*ADC_CR*1)**.

7	6	5	4	3	2	1	0
Зарезервировано	***SPSEL***[2:0]			Зарезервировано		***CONT***	***ADON***
	rw	***rw***	***rw***			***rw***	***rw***

- Биты 6:4 ***SPSEL***: Выбор предварительного делителя:
  000: $f_{ADC}=f_{MASTER}/2$;
  001: $f_{ADC}=f_{MASTER}/3$;
  010: $f_{ADC}=f_{MASTER}/4$;
  011: $f_{ADC}=f_{MASTER}/6$;

100: $f_{ADC}=f_{MASTER}/8$;
101: $f_{ADC}=f_{MASTER}/10$;
110: $f_{ADC}=f_{MASTER}/12$;
111: $f_{ADC}=f_{MASTER}/18$;

Примечание: Рекомендуется изменять данные биты, когда АЦП отключено. Иначе, во время изменения, может произойти сбой. Если данные биты изменить когда АЦП находится во включенном состоянии, то не следует принимать во внимание результаты первого преобразования.

• Бит 1 ***CONT***. Непрерывное преобразование.

Этот бит устанавливается и сбрасывается программно. Если он установлен, то происходит непрерывное преобразование, до того как этот бит не сбросится программно:

0: режим одиночного преобразования;

1: режим непрерывного преобразования.

• Бит 0 ***ADON***. Включение АЦП:

0: АЦП выключен;

1: АЦП включен.

Регистр конфигураций 2

Регистр конфигураций 2 **(*ADC_CR2*)**.

7	6	5	4	3	2	1	0
Зарезервировано	***EXTTRIG***	***EXTSEL*** [1:0]		***ALIGN***	Зарезервировано	***SCAN***	Зарезервировано
	rw	*rw*	*rw*	*rw*		*rw*	

• Бит 7. Зарезервирован.

• Бит 6 ***EXTTRIG***. Разрешение внешнего запуска:

0: преобразование по внешнему событию отключено;

1: преобразование по внешнему событию включено.

• Биты 5:4 ***EXTSEL***[1:0]. Выбор внешнего события.

Эти биты устанавливаются и сбрасываются программно. Они выбирают один из четырех типов событий, используемых для запуска преобразования АЦП:

00: внутреннее событие ***TIM1 TRGO***;

01: внешнее прерывание на ***ADC_ETR***;

10: зарезервировано;

11: зарезервировано.

• Бит 3 ***ALIGN***. Выравнивание данных:

0: выравнивание по левому краю (восемь ***MSB*** биты записываются в регистр ***ADC_DRH***, остальные биты ***LSB*** записываются в регистр ***ADC_DRL***). Порядок чтения – ***MSB***, а затем ***LSB***;

1: выравнивание по правому краю (восемь бит ***LSB*** записываются в регистр ***ADC_DRL***, остальные ***MSB*** биты записываются в регистр ***ADC_DH***). Порядок чтения – ***LSB***, а затем ***MSB***.

- Бит 1 ***SCAN***. Разрешение режима сканирования:

  0: режим сканирования отключен;

  1: режим сканирования включен.

Регистр конфигураций 3

Регистр конфигураций 3 (***ADC_CR3***).

7	6	5	4	3	2	1	0
DBUF	***OVR***	Зарезервировано					
rw	*rc_w0*						

Примечание: Данный регистр не доступен для АЦП2.

- Бит 7 ***DBUF***. Разрешение буфера данных.

Этот бит устанавливается и сбрасывается программно. Он используется вместе с битом ***CONT***. Когда ***DBUF*** установлен, преобразованные значения сохраняются в ***ADC_DBxRH*** и ***ADC_DBxRL*** регистрах, вместо ***ADC_DRH*** и ***ADC_DRL*** регистров:

0: буфер данных отключен;

1: буфер данных включен.

- Бит 6 ***OVR***. Флаг «***overrun***».

Этот бит устанавливается аппаратно и сбрасывается программно. Если этот бит установлен рекомендуется сбросить его и начать преобразование заново:

0: переполнения не было;

1: произошло переполнение в буферном регистре данных.

- Биты 5:0. Зарезервированы.

Старший регистр данных АЦП

Старший регистр данных АЦП (***ADC_DRH***).

7	6	5	4	3	2	1	0
DH[7:0]							
r	*r*	*r*	*r*	*r*	*r*	*r*	*r*

- Биты 7:0 ***DH***[7:0]. Старшие биты данных.

Эти биты устанавливаются/сбрасываются аппаратно и доступны только для чтения. Когда АЦП в режиме одиночного преобразования или в не буферизованном непрерывном режиме, в них хранится старшая часть результата преобразования, в зависимости от вида выравнивания (бит ***ALIGN***).

Младший регистр данных АЦП

Младший регистр данных АЦП (***ADC_DRL***).

7	6	5	4	3	2	1	0
DL[7:0]							
r	*r*	*r*	*r*	*r*	*r*	*r*	*r*

• Биты 7:0 ***DL***[7:0]. Младшие биты данных.

Эти биты устанавливаются/сбрасываются аппаратно и доступны только для чтения. Когда АЦП в режиме одиночного преобразования или в не буферизованном непрерывном режиме, в них хранится младшая часть результата преобразования, в зависимости от вида выравнивания (бит ***ALIGN***).

Старший регистр запрещения триггера Шмитта АЦП

Старший регистр запрещения триггера Шмитта АЦП (***ADC_TDRH***).

7	6	5	4	3	2	1	0
TD[15:8]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***TD***[15:8]. Старшие биты запрещения триггера Шмитта.

Данные биты устанавливаются и очищаются программно. Когда бит ***TDx*** установлен, это отключает триггер Шмитта порта ввода/вывода от соответствующего входного канала АЦП, даже если данный канал не был преобразован. Это позволяет понизить энергопотребление порта:

0: триггер Шмитта включен;

1: триггер Шмитта выключен.

Младший регистр запрещения триггера Шмитта АЦП

Младший регистр запрещения триггера Шмитта АЦП (***ADC_TDRL***).

7	6	5	4	3	2	1	0
TD[7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

• Биты 7:0 ***TD***[7:0]. Младшие биты запрещения триггера Шмитта.

Старший регистр верхнего порога АЦП

Старший регистр верхнего порога АЦП (***ADC_HTRH***).

7	6	5	4	3	2	1	0
HT[9:2]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Примечание: Данный регистр не доступен для АЦП2.

• Биты 7:0 ***HT***[9:2]. Старший байт верхнего порога напряжения аналогового контроля.

Эти биты устанавливаются и очищаются программно. Они определяют верхний порог (V_{REFH}) аналогового контроля.

Младший регистр верхнего порога АЦП

Младший регистр верхнего порога АЦП (***ADC_HTRL***).

7	6	5	4	3	2	1	0
Зарезервировано						***HT***[1:0]	
						rw	*rw*

Примечание: Данный регистр не доступен для АЦП2.

• Биты 7:2. Зарезервированы.

• Биты 1:0 ***HT***[1:0]. Младшие биты верхнего порога напряжения аналогового контроля.

Старший регистр нижнего порога АЦП

Старший регистр нижнего порога АЦП (***ADC_LTRH***).

7	6	5	4	3	2	1	0
LT[9:2]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Примечание: Данный регистр не доступен для АЦП2.

• Биты 7:0 ***LT***[9:2]. Старший байт нижнего порога напряжения аналогового контроля.

Эти биты устанавливаются и очищаются программно. Они определяют нижний порог (V_{REFL}) аналогового контроля.

Младший регистр нижнего порога АЦП

Младший регистр нижнего порога АЦП (***ADC_LTRL***).

7	6	5	4	3	2	1	0
Зарезервировано						***LT***[1:0]	
						rw	*rw*

Примечание: Данный регистр не доступен для АЦП2.

• Биты 7:2. Зарезервированы.

• Биты 1:0 ***LT***[1:0]. Младшие биты нижнего порога напряжения аналогового контроля.

Старший регистр статуса аналогового контроля АЦП

Старший регистр статуса аналогового контроля АЦП (***ADC_AWSRH***).

7	6	5	4	3	2	1	0
Зарезервировано						***AWS***[9:8]	
						rc_w0	*rc_w0*

Примечание: Данный регистр не доступен для АЦП2.

- Биты 7:2. Зарезервированы.
- Биты 1:0 ***AWS***[9:8]. Флаги статуса аналогового контроля:

  0: события от аналогового контроля в буферных регистрах не было;

  1: обнаружено событие от аналогового контроля в буферных регистрах.

Младший регистр статуса аналогового контроля АЦП

Младший регистр статуса аналогового контроля АЦП (***ADC_AWSRH***).

7	6	5	4	3	2	1	0
AWS[7:0]							
rc_w0	*rc_w0*	*rc_w0*	*rc_w0*	*rc_w0*	*rc_w0*	*rc_w0*	*rc_w0*

Примечание: Данный регистр не доступен для АЦП2.

- Биты 7:0 ***AWS***[7:0]. Флаги статуса аналогового контроля.

Старший регистр управления аналоговым контролем

Старший регистр управления аналоговым контролем (***ADC_AWCRH***).

7	6	5	4	3	2	1	0
Зарезервировано						***AWEN***[9:8]	
						rw	*rw*

Примечание: Данный регистр не доступен для АЦП2.

- Биты 7:2. Зарезервированы
- Биты 1:0 ***AWEB***[9:8]. Разрешение аналогового контроля.

Данные биты устанавливаются и сбрасываются программно. В буферизованном непрерывном режиме (***DBUF***=1, ***CONT***=1) и в режиме сканирования (***SCAN***=1) биты ***AWENx*** разрешают функцию аналогового контроля для каждого из 10 буферизованных регистров:

0: аналоговый контроль запрещен;

1: аналоговый контроль разрешен.

Младший регистр управления аналоговым контролем

Младший регистр управления аналоговым контролем (***ADC_AWCRL***).

7	6	5	4	3	2	1	0
AWEN [7:0]							
rw	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*	*rw*

Примечание: Данный регистр не доступен для АЦП2.

- Биты 7:0 ***AWEB***[7:0]. Разрешение аналогового контроля.

5.3. Пример настройки АЦП

В данной программе осуществляется настройка АЦП в режим одиночного преобразования.

```
#include "iostm8s003k3.h"          //подключение заголовочного файла с объявлениями регистров, масок и битов
void interrupt_init(void);         //объявление подпрограмм настройки прерываний
void adc_init(void);               //объявление подпрограмм настройки АЦП
int i;                             //объявление переменной i
#pragma vector=0x18                //в таблице прерываний у АЦП ADC1 номер прерывания равен 22.Прибавим 2 получим 24, в hex формате: 0x18
__interrupt void ADC_end(void);    //объявление вектора прерывания АЦП
int main( void )                   //основная программа
{
adc_init();                        //вызов подпрограммы настройки АЦП
interrupt_init();                  //вызов подпрограммы настройки прерываний
  for (;;)                         //бесконечный цикл
  {
  }
}

void adc_init(void)                //подпрограмма настройки АЦП
{
CLK_PCKENR2=0xff;                  //тактирование АЦП
ADC_CSR_bit.AWD=0;                 //0 – нет события от AWD
```

```
ADC_CSR_bit.EOCIE=1;        //прерывание по окончанию преобразования разрешено
ADC_CSR_bit.AWDIE=0;        //прерывание от сторожевого запрещено
ADC_CSR_bit.CH=0x0;         //канал AIN0
ADC_CR1_bit.SPSEL=0x0;      //Выбор частоты работы АЦП
ADC_CR1_bit.CONT=0;         //0-одиночное преобразование; 1-несколько преобразований
ADC_CR2_bit.EXTTRIG=0;      //преобразование по внешнему событию: 0-выкл; 1-вкл
ADC_CR3_bit.DBUF=0;         //результат преобразования ADC_DRH и ADC_DRL
ADC_CR3_bit.OVR=0;          //очистка флага завершения преобразования
ADC_CR1_bit.ADON=1;         //включение питания АЦП
i=0;
  do {i++;}
  while (i<100);            //временная задержка
ADC_CR1_bit.ADON=1;         //разрешить начало преобразований
}

void interrupt_init(void)   //подпрограмма настройки прерывания
{
asm("rim");                 //глобальное разрешение прерываний
}

__interrupt void ADC_end(void)
{
ADC_CSR_bit.EOC=0;          //очистка флага завершения преобразования
PD_DDR_bit.DDR0=1;          //0-вход; 1-выход
PD_CR1_bit.C10=1;           //0-выход с открытым стоком; 1-выход типа Push-pull
PD_CR2_bit.C20=0;           //скорость переключения: 0 – 2 МГц; 1 – 10 МГц
PD_ODR_bit.ODR0= 0;         //зажигаем светодиод
}
```

Приложение 1

Создание проекта в программе *IAR Embedded Workbench*

Создание проекта:

1. Запускаем среду *IAR Embedded Workbench for STMicroelectronics STM*8. На рис. П.1 представлен внешний вид стартового окна программы.

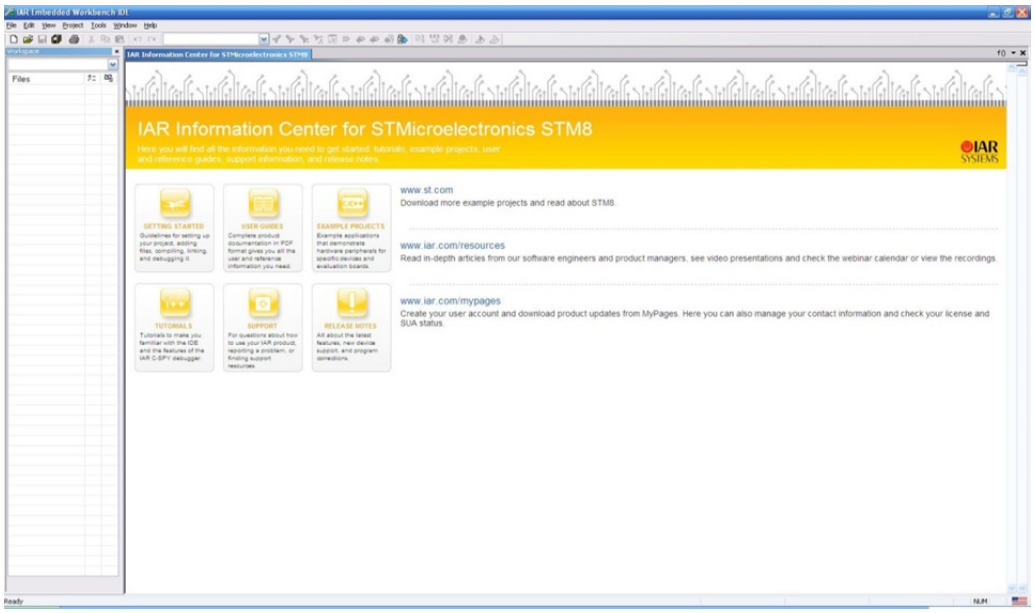

Рис П.1. Стартовое окно программы

2. Для создания нового проекта необходимо зайти в меню «*Project*» и выбрать пункт «*Create new project…*» (рис. П.1).

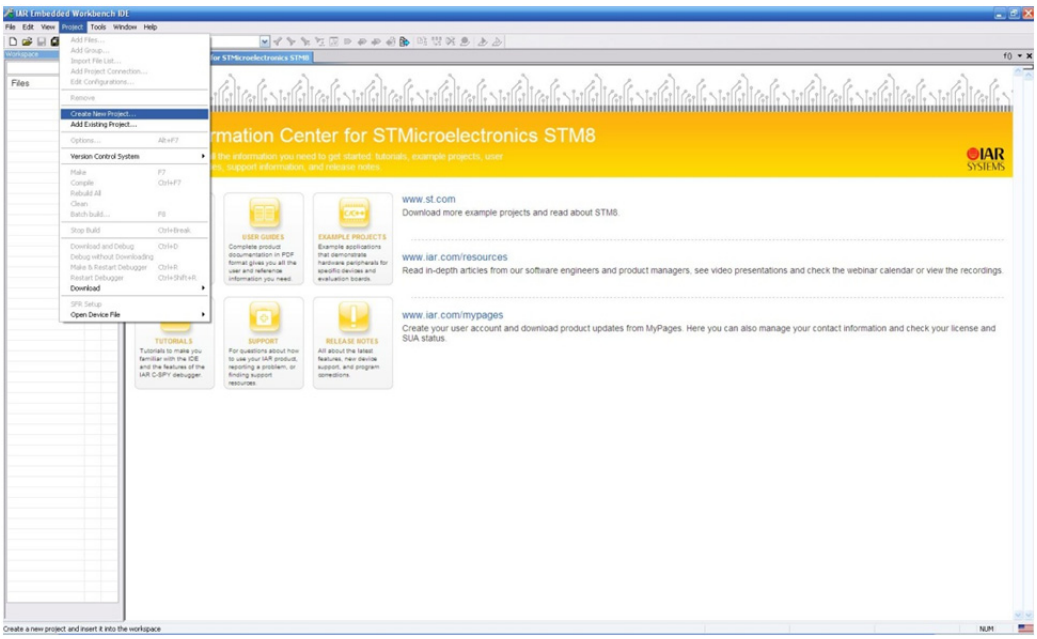

Рис П.2. Окно создания нового проекта

3. В появившемся окне (Рис. П.3) необходимо выбрать шаблон для языка Си и тип микроконтроллера, и далее сохранить рабочую область – *Workspace* (рис. П.4).

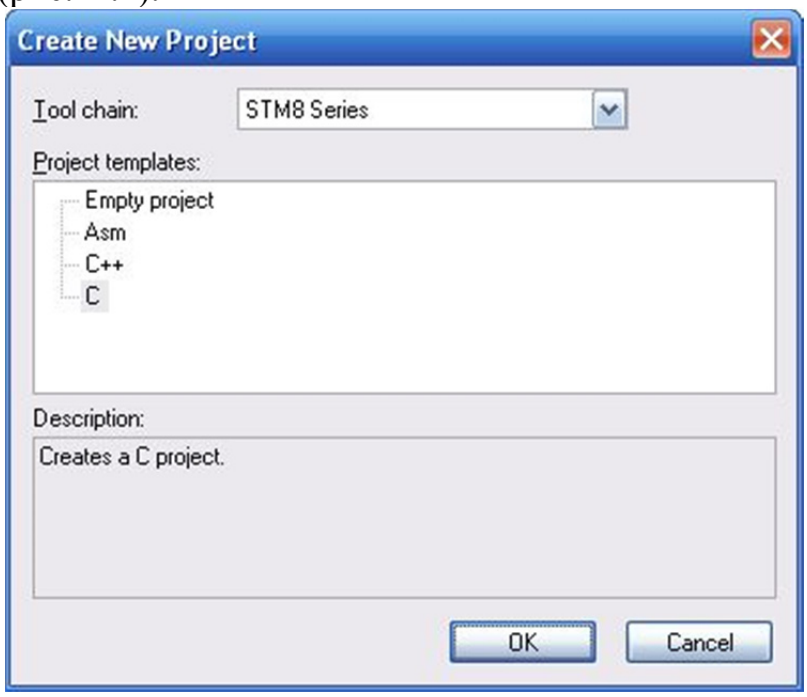

Рис П.3. Окно выбора языка программирования и микроконтроллера

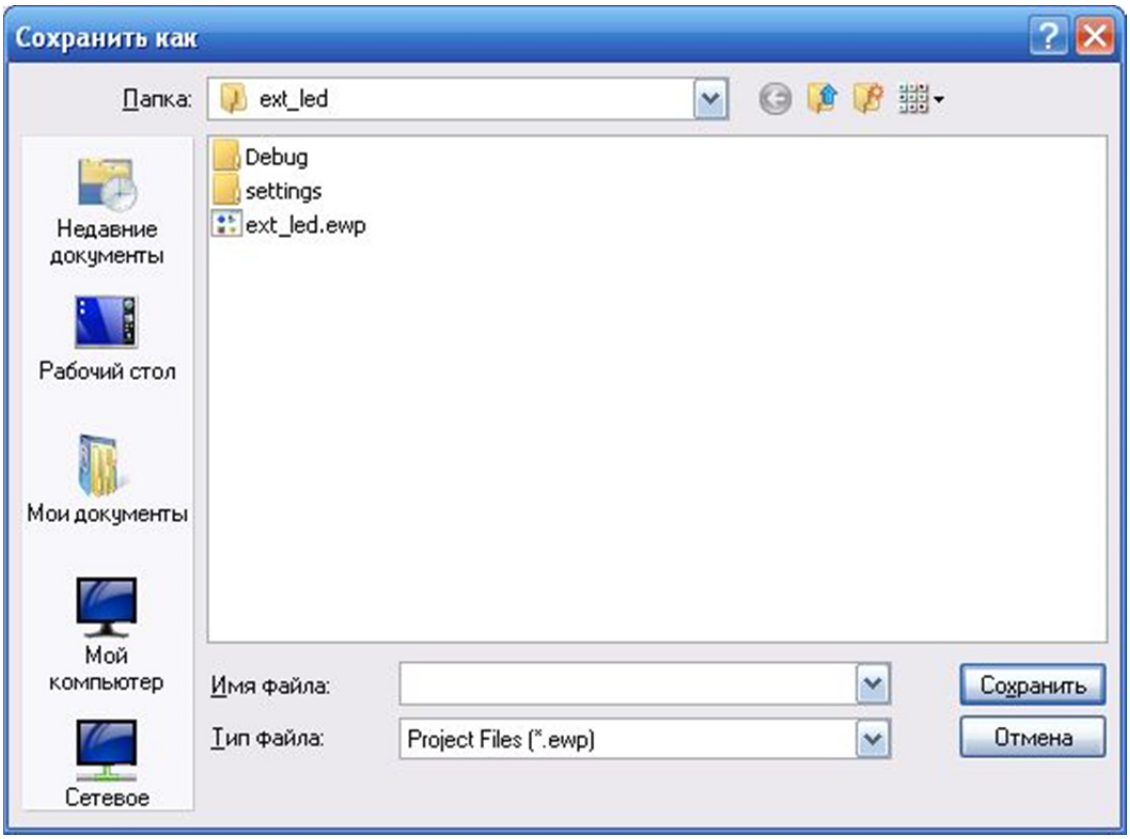

Рис П.4. Окно сохранения проекта

4. После сохранения проекта будет открыто рабочее окно проекта (рис. П.5).

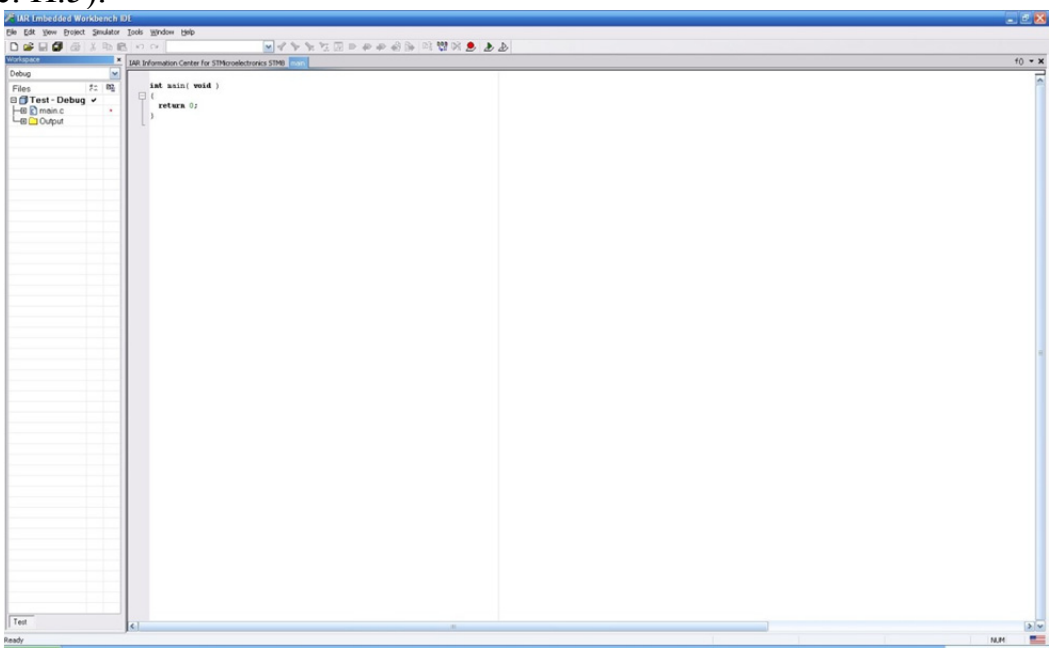

Рис П.5. Рабочее окно проекта

5. Для примера работы программы напишем следующий код и сохраним проект (Рис. П.6):

```
#include "iostm8s003k3.h"

int main( void )
{
}
```

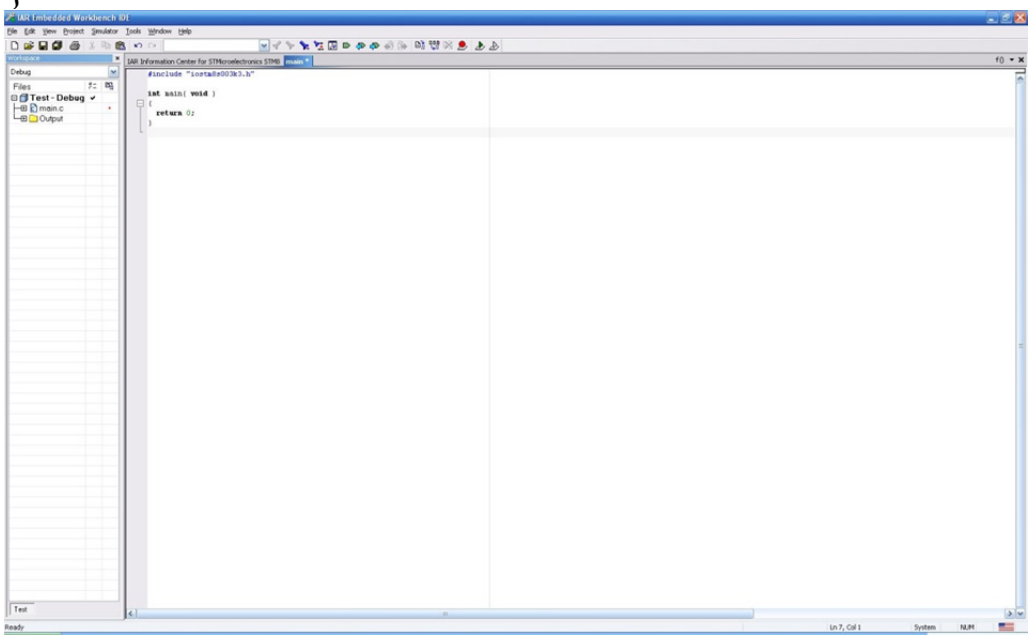

Рис П.6. Рабочее окно проекта

6. Далее необходимо настроить проект. Для этого в окне «*Workspace*» выберем пункт контекстного меню «*Options*» (Рис. П.7).

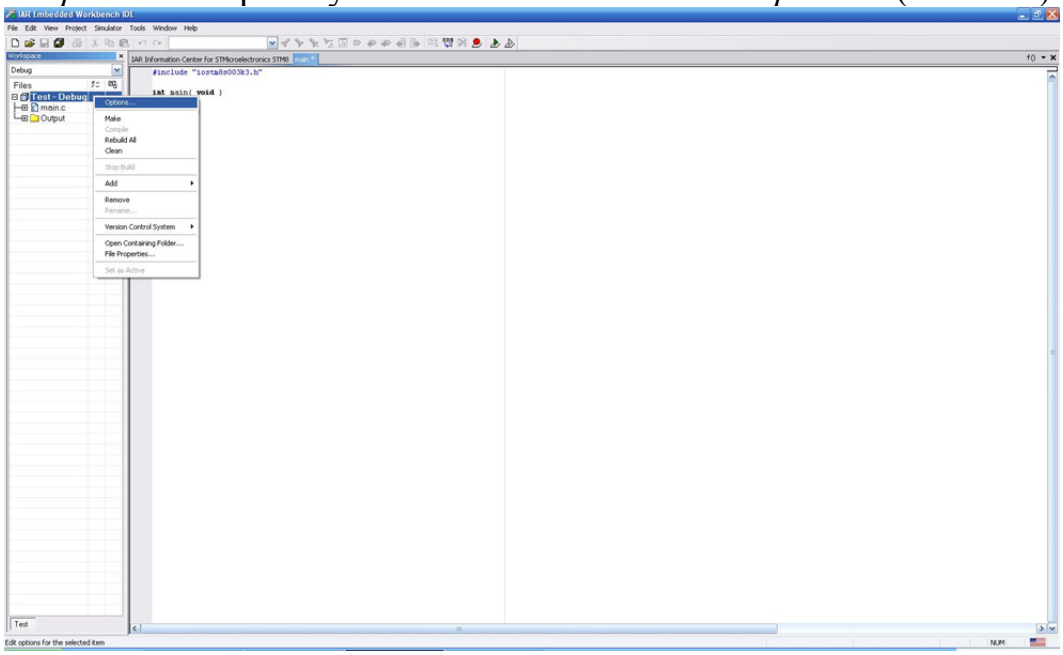

Рис П.7. Окно настройки проекта

7. На странице *General*, вкладке *Target* выбираем модель контроллера: *STM8S—>STM8S003K3* (Рис. П.8).

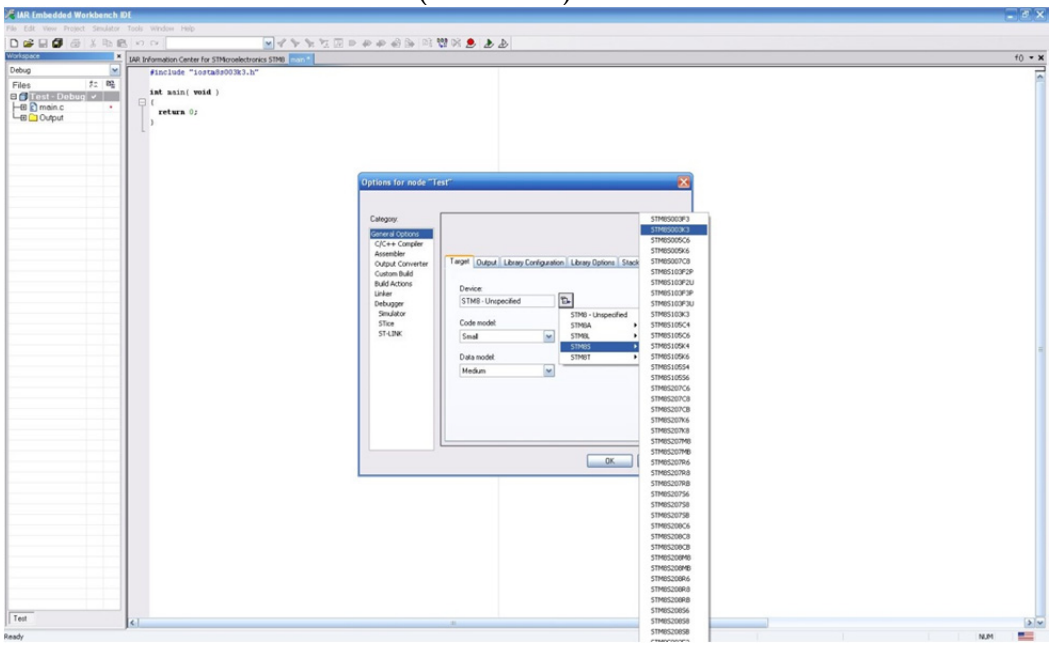

Рис П.8. Окно выбора микроконтроллера

8. На странице *Debugger*, вкладке *Setup* необходимо выбрать отладчик *ST-Link* (Рис. П.9).

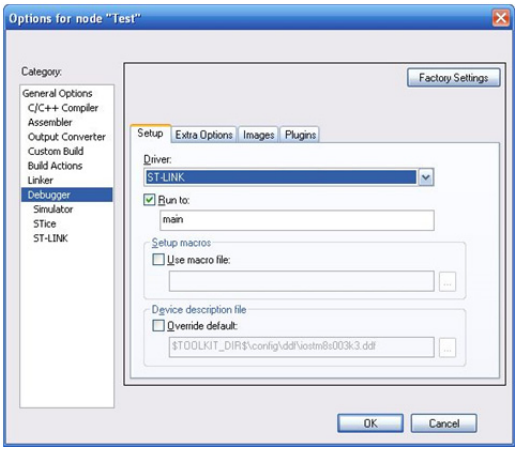

Рис П.9. Окно выбора отладчика

9. Загрузка программы в микроконтроллер осуществляется в три этапа (Рис. П.10): компиляция (*Compile*), создание (*Make*), загрузка и отладка (*Download and Debug*)

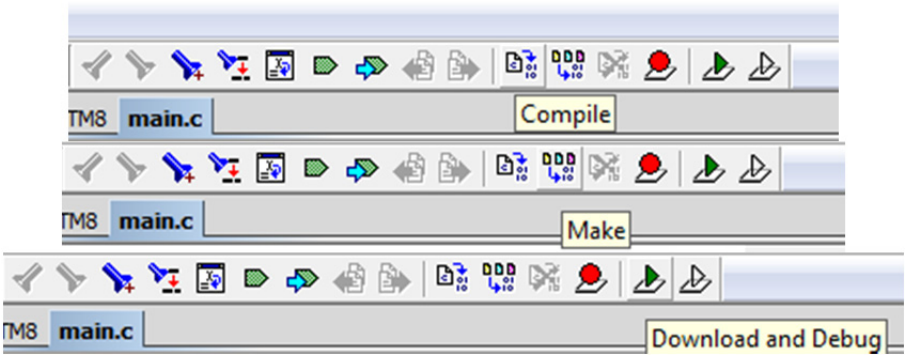

Рис П.10. Панели компиляции и загрузки программы

Далее появится окно показанное на рис. П.11.

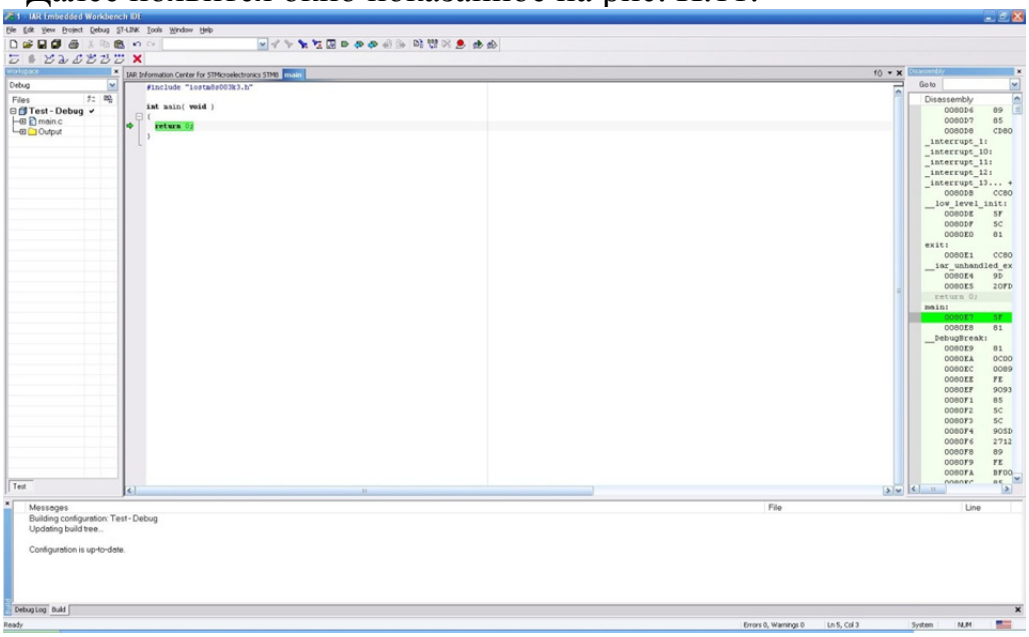

Рис П.11. Окно отладчика программы

10. Далее можно запускать программу как показано на рис. П.12.

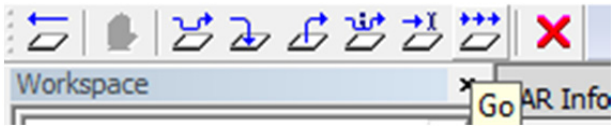

Рис П.12. Запуск программы

Список литературы

1. Ершова Н.Ю., Иващенков О.Н., Курсков С.Ю. Микропроцессоры. – Санкт-Петербург, 2002.
2. Микропроцессоры: в 3-х кн. / под ред. С.В. Преснухина. – М.: Высшая школа, 1986. – Кн.1. – 495 с. – Кн. 2. – 383 с. – Кн. 3. – 351 с.
3. Ливенцов С.Н., Вильнин А.Д., Горюнов А.Г. Основы микропроцессорной техники: учебное пособие. - Томск: ТПУ, 2007. - 118с.
4. Молодецкий В.Б., Кривенков М.В., Пахомов А.Н., Кудашев С.В. Микропроцессорная техника: учебное пособие. – Красноярск: ИПК СФУ, 2008. - 126с.
5. Sunil Mathur. Microprocessor 8085 and its interfacing. – New delhi: PHI Learning Private Limitted, 2011. – 868 p.
6. Основы микропроцессорной техники: учебное пособие / Ю.В. Новиков, П.К. Скоробогатов. – 4-е изд., испр. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. – 357 с.: ил.
7. Цифровая обработка сигналов: практический подход, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 992 с.: ил.

Учебное издание

ТОРГАЕВ Станислав Николаевич
МУСОРОВ Илья Сергеевич
ЧЕРТИХИНА Дарья Сергеевна
ТРИГУБ Максим Викторович
РЫБАКОВ Алексей Николаевич

ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ: МИКРОКОНТРОЛЛЕР STM8S

Учебное пособие

Компьютерная верстка *В.В. Михалев*

Зарегистрировано в Издательстве ТПУ
Размещено на корпоративном портале ТПУ

Национальный исследовательский Томский политехнический университет
Система менеджмента качества
Издательства Томского политехнического университета
сертифицирована в соответствии с требованиями ISO 9001:2008

ИЗДАТЕЛЬСТВО ТПУ. 634050, г. Томск, пр. Ленина, 30
Тел./факс: 8(3822)56-35-35, www.tpu.ru