

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего  
профессионального образования**

**«Санкт-Петербургский государственный университет  
аэрокосмического приборостроения»**

---

**Методические указания  
к выполнению лабораторных работ  
по дисциплине «Программирование встроенных приложений»**

**Санкт-Петербург**

**2017**

## **Оглавление**

Введение .....	3
Настройка среды .....	6
Лабораторная работа №1 .....	10
Краткие теоретические сведения .....	10
Лабораторная работа №2 .....	11
Краткие теоретические сведения .....	11
Лабораторная работа №3 .....	14
Краткие теоретические сведения .....	15
Лабораторная работа №4 .....	15
Краткие теоретические сведения .....	15
Список использованной литературы .....	17
Приложение А.....	18
Приложение Б .....	21
Приложение В .....	24
Приложение Г .....	29

## Введение

Микроконтроллеры являются основой при реализации автоматизированных вычислительных систем для управления различными процессами. Фактически, это однокристалльный компьютер с интегрированными устройствами ввода-вывода, таймерами и возможностью подключения периферийных устройств. На сегодняшний день существует множество различных производителей микроконтроллеров и встраиваемых систем, что обусловлено ростом спроса на автоматизацию работы различных процессов и встраиваемых систем.

Первый патент на однокристалльную микроЭВМ был выдан в 1971 году инженерам М. Кочрену и Г. Буну, сотрудникам американской Texas Instruments. Именно они предложили на одном кристалле разместить не только процессор, но и память с устройствами ввода-вывода.

В 1976 году американская фирма Intel выпускает микроконтроллер i8048. В 1978 году фирма Motorola выпустила свой первый микроконтроллер MC6801, совместимый по системе команд с выпущенным ранее микропроцессором MC6800. Через 4 года, в 1980 году, Intel выпускает следующий микроконтроллер: i8051. С точки зрения технологии микроконтроллер i8051 являлся для своего времени очень сложным изделием — в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре i8086.

На сегодняшний день существует более 200 модификаций микроконтроллеров, совместимых с i8051, выпускаемых двумя десятками компаний, и большое количество микроконтроллеров других типов. Популярностью у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel, 16-битные MSP430 фирмы TI, а также 32-битные микроконтроллеры архитектуры ARM, которую разрабатывает фирма ARM Limited и продаёт лицензии другим фирмам для их производства.

Существует огромное количество типов микроконтроллеров, отличающихся архитектурой процессорного модуля, размером и типом встроенной памяти, набором периферийных устройств, типом корпуса и т. д. В отличие от обычных компьютерных микропроцессоров, в микроконтроллерах часто используется гарвардская архитектура памяти, то есть раздельное хранение данных и команд в ОЗУ и ПЗУ соответственно.

Неполный список периферийных устройств, которые могут использоваться в микроконтроллерах, включает в себя:

- универсальные цифровые порты, которые можно настраивать как на ввод, так и на вывод;
- различные интерфейсы ввода-вывода, такие, как UART, I<sup>2</sup>C, SPI, CAN, USB, IEEE 1394, Ethernet;
- аналого-цифровые и цифро-аналоговые преобразователи;
- компараторы;
- широтно-импульсные модуляторы (ШИМ-контроллер);

- таймеры;
- контроллеры бесколлекторных двигателей, в том числе шаговых;
- контроллеры дисплеев и клавиатур;
- радиочастотные приемники и передатчики;
- массивы встроенной флеш-памяти;
- встроенные тактовый генератор и сторожевой таймер.

Использование в современном микроконтроллере достаточного мощного вычислительного устройства с широкими возможностями, построенного на одной микросхеме вместо целого набора, значительно снижает размеры, энергопотребление и стоимость построенных на его базе устройств.

Используются в управлении различными устройствами и их отдельными блоками:

- в вычислительной технике: материнские платы, контроллеры дисководов жестких и гибких дисков, CD и DVD, калькуляторы;
- электронике и разнообразных устройствах бытовой техники, в которой используется электронные системы управления — стиральных машинах, микроволновых печах, посудомоечных машинах, телефонах и современных приборах, различных роботах, системах «умный дом», и др..

В промышленности:

- устройства промышленной автоматики — от программируемого реле и встраиваемых систем до ПЛК;
- систем управления станками.

Одними из самых известных семейств микроконтроллеров являются:

- MCS 51 (Intel)
- MSP430 (TI)
- ARM (ARM Limited)
- ST Microelectronics STM32 ARM-based MCUs
- Atmel Cortex, ARM7 и ARM9-based MCUs
- Texas Instruments Stellaris MCUs
- NXP ARM-based LPC MCUs
- Toshiba ARM-based MCUs
- Analog Devices ARM7-based MCUs
- Cirrus Logic ARM7-based MCUs
- Freescale Semiconductor ARM9-based MCUs

В рамках данного курса будет изучаться программирование, широко распространенного среди российских разработчиков электроники, микроконтроллера STM32 на ядре ARM Cortex-M3 от компании STMicroelectronics.

Ядро процессора Cortex-M3 построено с использованием Гарвардской архитектуры с 3-уровневым конвейером, в сочетании с рядом расширенных функций, включая одноцикловый умножитель и аппаратный делитель, обеспечивающие исключительно

высокую производительность в 1,25 DMIPS/МГц. Процессор Cortex-M3 работает также с системой команд Thumb-2, которая, в сочетании с такими функциями, как хранение невыровненных данных и побитовая обработка, обеспечивает 32-разрядную производительность при стоимости, эквивалентной стоимости современных 8- и 16-разрядных микроконтроллеров.

В семейство STM32 входят две линейки приборов:

- Access (F101xx): частота тактирования 36МГц, от 32 до 128 кбайт флэш-памяти, от 6 до 16 кбайт SRAM, до 7 коммуникационных интерфейсов. Линейка Access разработана с тем, чтобы внедрить 32-разрядную схемотехнику в критичные стоимости применения или в 16-разрядные проекты.
- Performance (F103xx): частота тактирования 72МГц, от 256 до 512 кбайт флэш-памяти, до 64 кбайт SRAM, контроллер статической памяти с поддержкой Compact Flash, SRAM, PSRAM, NOR и NAND памяти, с поддержкой LCD параллельного интерфейса (F103Vx). Микроконтроллеры имеют до 13 коммуникационных интерфейсов, в том числе USB и CAN. Линейка Performance ориентирована на применения, которым необходима одновременно повышенная производительность обработки и экономичная работа.

В текущем году семейство STM32 значительно расширилось: добавлено еще 28 новых микроконтроллеров, включая недорогие приборы, размещенные в 36-выводных корпусах, и приборы более высокого класса, размещенные в 144-выводных корпусах.

Под ARM-архитектуру существует довольно широкий выбор программных средств разработки.


Наиболее популярными (но и самыми дорогими) среди разработчиков для разработки ПО под ARM архитектуру являются инструментарии от компаний Keil и IAR Systems. Это обусловлено наиболее продвинутыми C-инструментариями с точки зрения оптимизации и компактности кода. Также, помимо лидирующих позиций в C-инструментариях, данные компании предоставляют широкие наборы дополнительного ПО — операционные системы реального времени, USB-стеки, TCP/IP-стеки и многое другое.

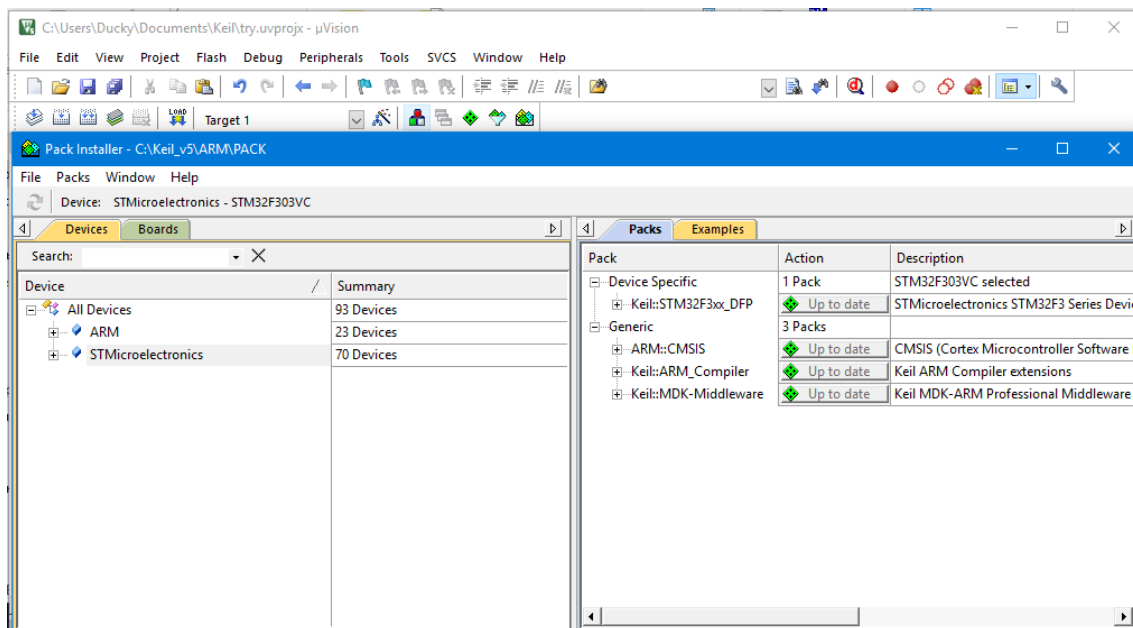
При выполнении цикла лабораторных работ на микроконтроллере серии STM32 фирмы STMicroelectronic будет использоваться интегрированная среда разработки (ИСР) (Integrated Development Environment - IDE) компании Keil Elektronik, называемая  $\mu$ Vision3 (произносится как «микровижн»). Среда разработки Keil  $\mu$ Vision3 включает в себя менеджер проектов, редактор исходных кодов, компилятор, средства отладки и утилиты для полной симуляции микроконтроллера.

В составе среды  $\mu$ Vision имеется два средства отладки. Прежде всего, после компиляции и компоновки программы ее код можно загрузить в симулятор  $\mu$ Vision3. Этот отладчик симулирует работу процессорного ядра ARM и периферийных устройств поддерживаемого микроконтроллера. Работа с симулятором — хороший способ получить представление об устройствах семейства STM32. Поскольку симулятор обеспечивает моделирование работы ядра и периферии с точностью до такта, он может оказаться очень полезным инструментом для проверки, корректно ли был инициализирован микроконтроллер и правильно ли были вычислены различные стартовые константы, например значения коэффициента деления предделителей таймеров.

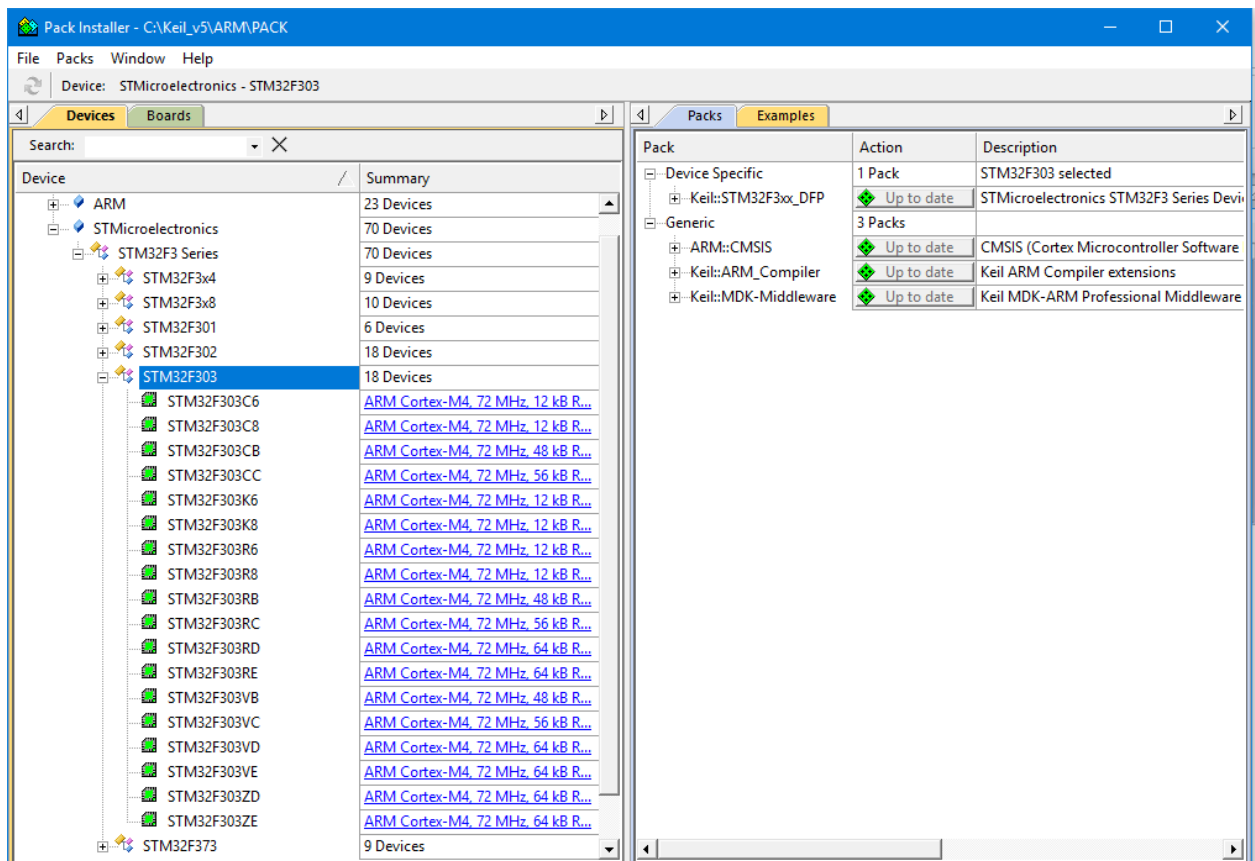
Клиентскую часть симулятора можно подключить к вашему устройству через фирменный отладчик компании Keil – ULINK. К персональному компьютеру кабель ULINK подключается через USB (universal serial bus), а к устройству – через интерфейс JTAG микроконтроллера STM32, который является отдельным модулем ядра ARM, который поддерживает команды отладки, посылаемые хостом (ПК).

## Настройка среды

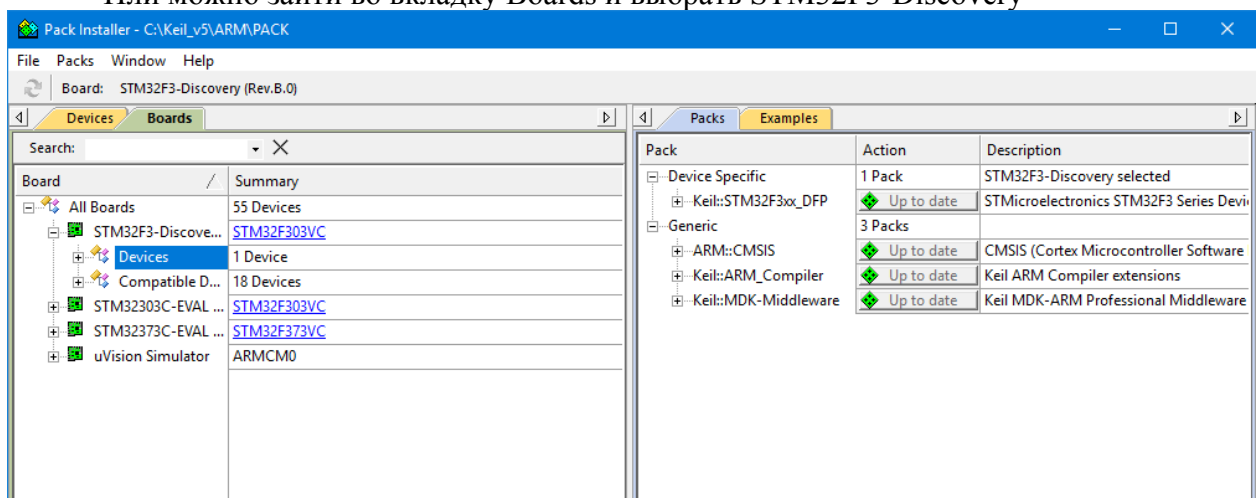
1. После установки программы Keil uVision5 в окне pack installer  импортируем необходимый для работы файл (File->import->Keil.STM32F3xx\_DFP.1.3.0.pack).



2. В этом же окне во вкладке Devices выбираем STMicroelectronic->STM32F303. Проверяем, чтобы файлы были подключены.



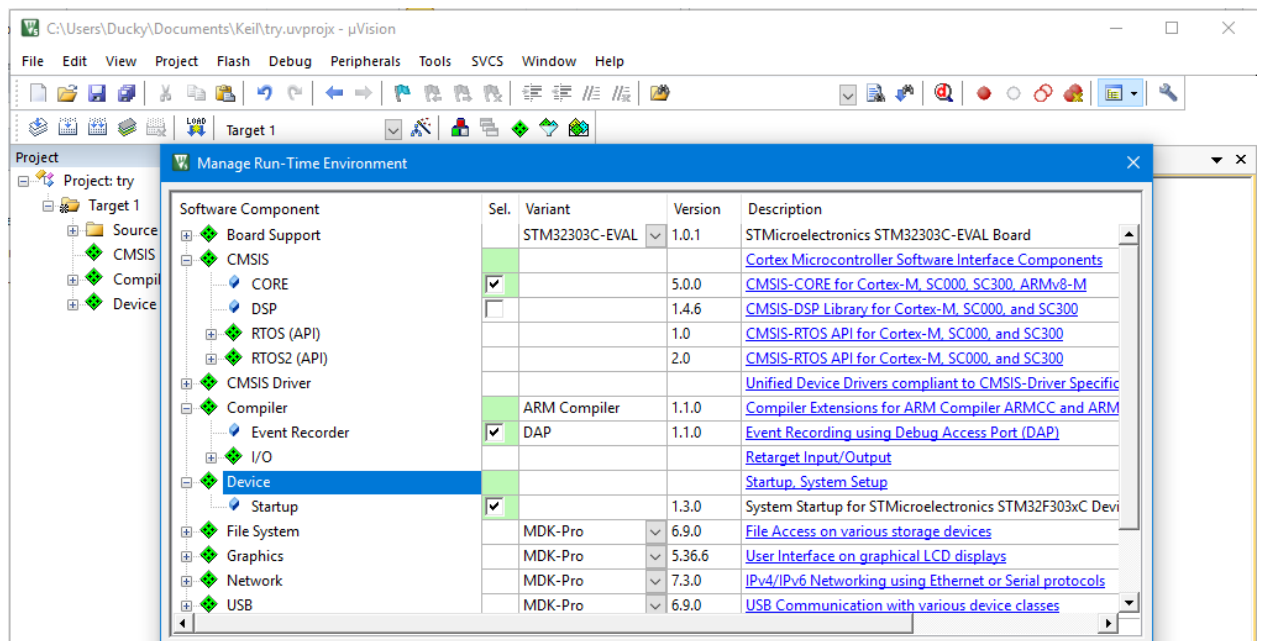
Или можно зайти во вкладку Boards и выбрать STM32F3-Discovery



Во вкладке Packs должны отображаться пакеты, отображенные на скрине.

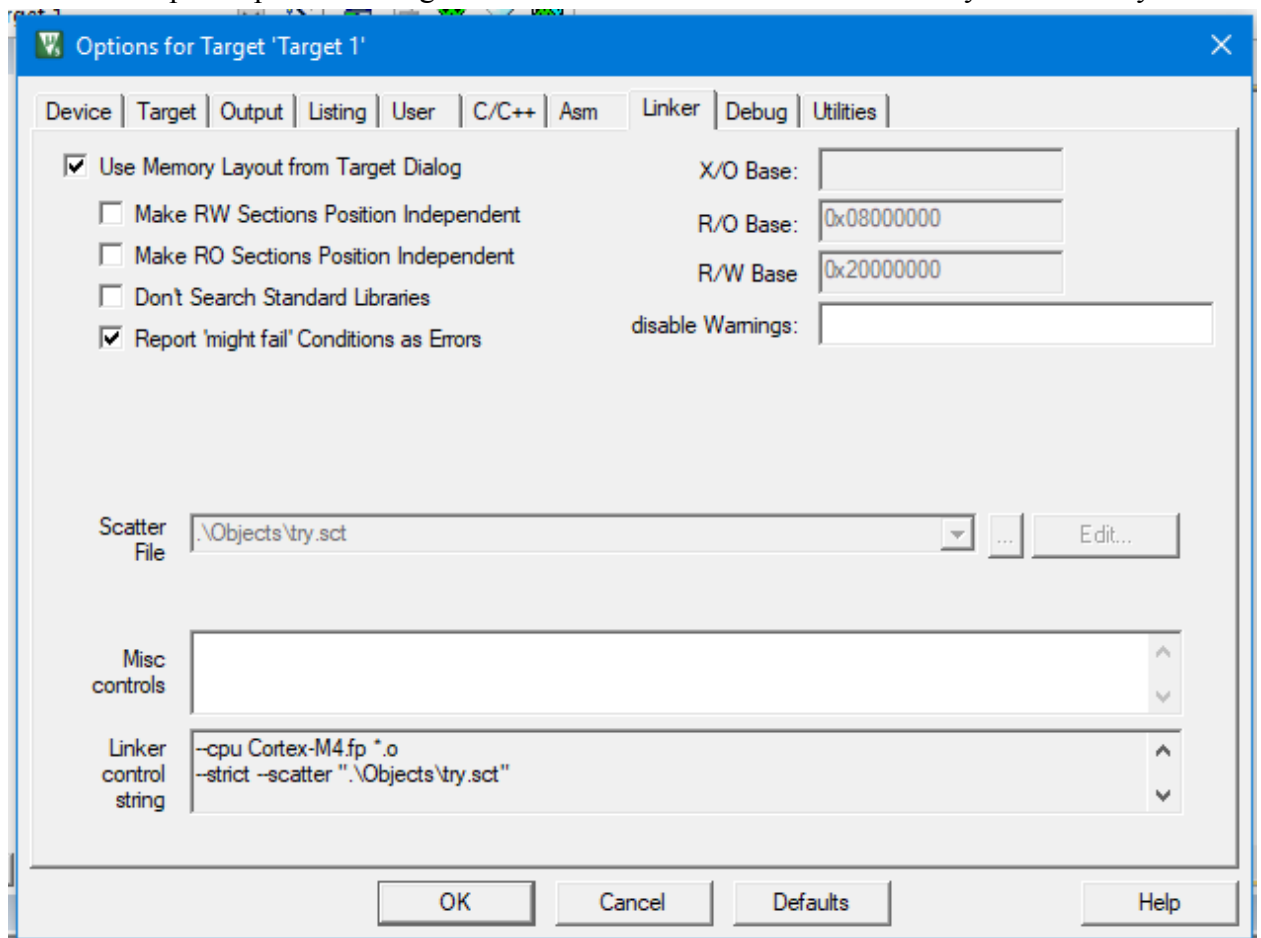
3. Создаем проект.

4. В окне Manage Run  выбираем нужные пункты



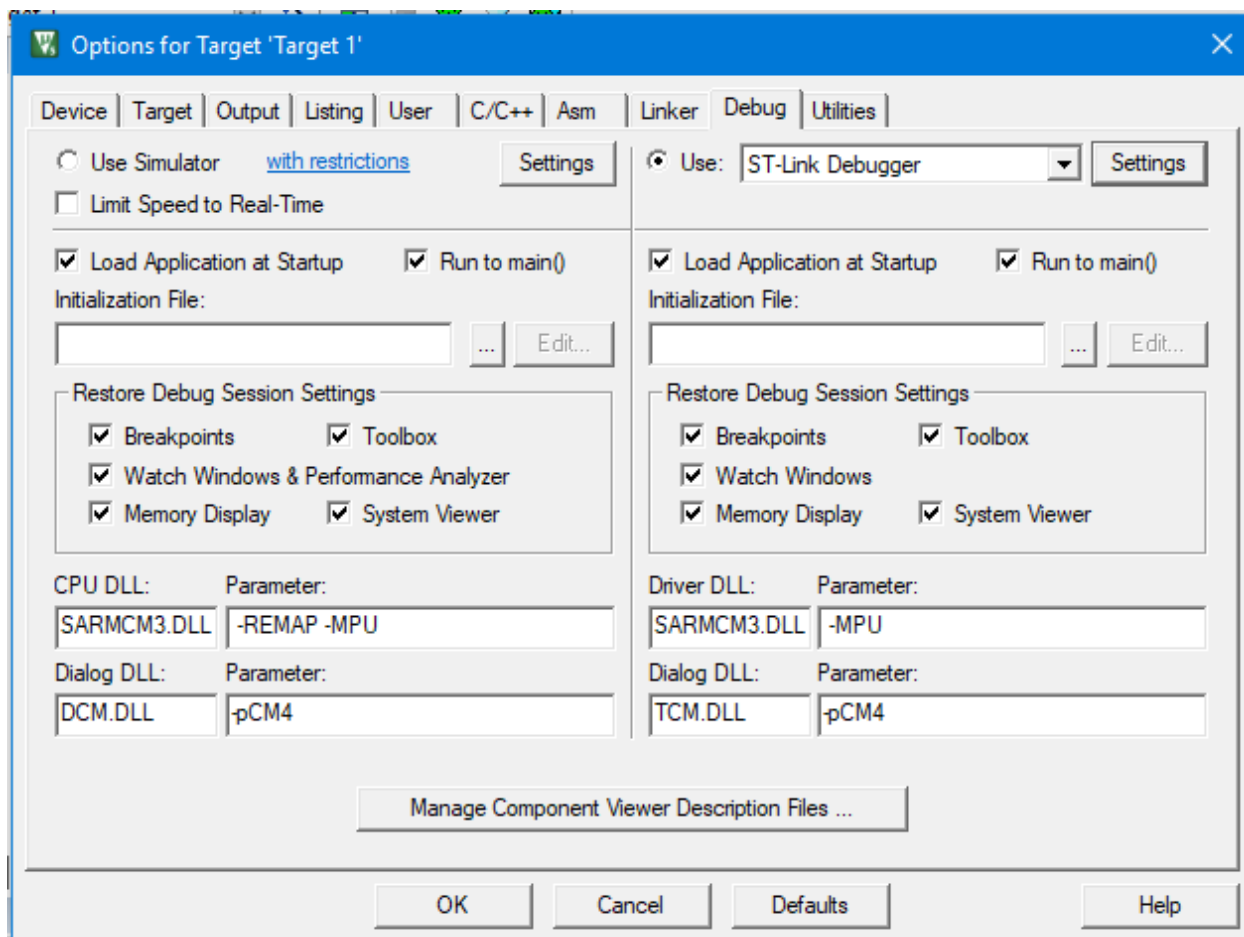
5. Устанавливаем пакет en.stsw-link009

6. Теперь в Options for target  во вкладке Linker ставим галочку Use memory...

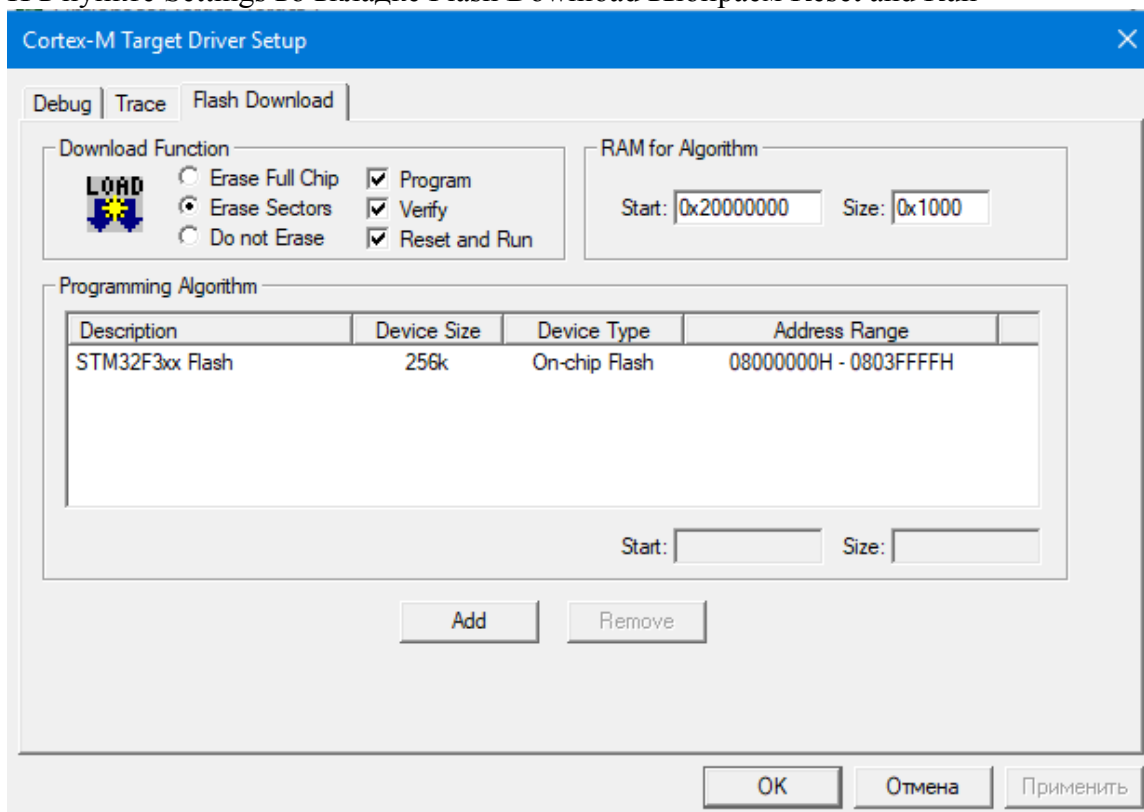


Во вкладке Debug выбрать ST-Link-Debugger





И в пункте Settings во вкладке Flash Download выбираем Reset and Run



7. Осталось добавить файл в Source Group и задать ему расширение .c

## Лабораторная работа №1

### Цель работы

Познакомиться с принципами настройки микроконтроллеров на примере STM32.

### Задание

Настроить микроконтроллер на частоту, заданную преподавателем.

Программа должна управлять тем, какие диоды горят в данный момент. При нажатии на джойстик должны загораться другие диоды. Описание регистров приведено в Приложении А.

### Краткие теоретические сведения

Написание программы для микроконтроллера – это, в первую очередь, настройка регистров, отвечающих за его отдельные элементы. Есть регистры описывающие преобразователи частот, есть регистры, отвечающие за то, на какой вывод микроконтроллера пойдет сигнал. Собственно, любое действие программируется записью информации в требуемые регистры.

Например, если требуется чтобы в ходе работы программы загорелся диод, мы должны занести требуемое значение в регистр GPIOE-> ODR. Он отвечает за то, какой конкретно диод будет гореть (или все вместе).

В общем виде для того, чтобы зажечь диод, алгоритм действий будет выглядеть так:

- разрешить тактирование необходимых устройств (регистр RCC-> AHBENR)
- настроить на необходимые режимы регистр GPIOE-> MODER
- записываем информацию в GPIOE-> ODR.

### Генератор HSI

Сигнал синхронизации HSI генерируется от внутреннего генератора 8 МГц RC и может использоваться непосредственно в виде системных часов или, будучи разделенным на 2, будет использоваться как вход PLL. Генератор автоматически запускается при появлении питания и при выходе в нормальный режим работы выставляет флаг HSIRDY в регистре RCC\_CR. Первоначально процессорное ядро запускается на тактовой частоте HSI.

### PLL

PLL может использоваться для умножения выходной частоты HSI. Конфигурирование PLL необходимо выполнять до его включения. После включения PLL эти параметры изменены уже быть не могут.

Алгоритм настройки PLL:

1. Отключить PLL, установив PLLON в 0.
2. Подождать, пока PLLRDY не будет очищен (пока полностью не остановится).
3. Изменить требуемый параметр.

4. Включите PLL снова, установив PLLON в 1.

Выходная частота PLL должна быть установлена в диапазоне 16-72 МГц.

## **Лабораторная работа №2**

### **Цель работы**

Познакомиться с принципами настройки прерываний микроконтроллеров на примере STM32.

### **Задание**

Настроить генерацию прерываний таймером.

Настроить генерацию внешнего прерывания от джойстика.

Описание регистров приведено в Приложении Б.

### **Краткие теоретические сведения**

В МК Cortex-M есть два понятия, которые часто путают Interrupt и Event. Event — это событие (аппаратное или программное), на которое могут реагировать ядро или периферийные блоки. Одним из вариантов реакции может быть — прерывание. Interrupt — это прерывание работы программы и переход управления в специализированный участок обработчик прерывания. Взаимосвязь между Event и Interrupt заключается в следующем: Каждый Interrupt вызывается Event, но не каждый Event вызывает Interrupt. Помимо прерываний, события могут активировать и другие возможности МК.

### **Контроллер вложенных векторизованных прерываний**

Контроллер вложенных векторизованных прерываний (КВВП) является стандартным блоком ядра Cortex. Это означает, что у любого Cortex-микроконтроллера будет присутствовать одна и та же структура прерываний, независимо от его производителя. Таким образом, прикладной код и операционные системы можно легко портировать с одного МК на любой другой и, при этом, программисту не потребуется изучение нового набора регистров. При разработке КВВП также учитывалось, что задержка реагирования на прерывание должна быть очень малой.

Как следует из наименования КВВП, им поддерживаются вложенные прерывания и, в частности, у МК STM32 используется 16 уровней приоритетов. Структура прерываний КВВП разработана с учетом программирования полностью на Си и

исключает потребность в написании каких-либо ассемблерных макросов или специальных, несовместимых с ANSI, директив.

Работа КВВП при входе в исключительные ситуации и выходе из них.

Если прерывание инициируется УВВ, то КВВП подключит ЦПУ Cortex к обработке прерывания. После перехода ЦПУ Cortex в режим прерывания, он помещает набор регистров в стек. Эта операция выполняется с помощью специального микрокода, что упрощает прикладной код. В процессе записи данных в стек на шине инструкций осуществляется выборка начального адреса процедуры обработки прерывания. КВВП реагирует на обработку прерывания с задержкой всего лишь 12 циклов. В них входит выполнение микрокода, который автоматически помещает набор регистров в стек

К числу помещаемых в стек данных относятся регистр статуса программы, счетчик программы и регистр связи. Благодаря этому, запоминается состояние, в котором находилось ЦПУ Cortex CPU. Помимо высокого быстродействия обработки одного прерывания, КВВП также характеризуется эффективной обработкой нескольких прерываний, что важно для быстродействующих систем реального времени. Для этого у КВВП поддерживается несколько оригинальных методов, позволяющие обрабатывать несколько запросов прерываний с минимальными задержками между прерываниями и с гарантированием приоритетности обработки прерываний.

### **Таймеры общего назначения**

Таймеры общего назначения состоят из 16-битного или 32-битного счетчика автоматической перезагрузки, управляемого программируемым предварительным делителем. Они могут использоваться для различных целей, включая измерение длительности импульсов входных сигналов (захват ввода) или формирование выходных сигналов (выходное сравнение и PWM). Длины импульсов и периоды колебаний могут быть модулированы с нескольких микросекунд до нескольких миллисекунд, используя предварительный делитель таймера и предварительные делители контроллера RCC. Таймеры полностью независимы и не разделяют никаких ресурсов.

Таймер TIMx общего назначения имеют следующие возможности:

- 16-битный (TIM3 и TIM4) или 32-разрядный (TIM2) счетчик автоматической перезагрузки;
- 16-битный программируемый предварительный делитель, используемый для деления (также «на лету») частоты счетчика с любым коэффициентом между 1 и 65535;
- До 4 независимых каналов для: захвата ввода, сравнения выхода, генерации PWM, выход в одноимпульсном режиме;

- схема синхронизации для управления таймером с помощью внешних сигналов и для соединения нескольких таймеров;

- генерация прерываний / DMA на следующих событиях:

- обновление: переполнение / переполнение счетчика, инициализация счетчика (с помощью программного обеспечения или внутреннего / внешнего триггера);

- событие триггера (запуск счетчика, остановка, инициализация или подсчет с помощью внутреннего / внешнего триггера);

- входной захват;

- сравнение выходных сигналов.

Основным блоком программируемого таймера является 16-разрядный / 32-разрядный счетчик с соответствующим регистром автозагрузки. блок включает в себя:

- регистр счетчика (TIMx->CNT);

- регистр предварительного делителя (TIMx\_PSC);

- регистр автозагрузки (TIMx\_ARR).

### **Предделитель**

Предделитель может разделить частоту счетчика с любым коэффициентом между 1 и 65536. Он основан на 16-разрядном счетчике, управляемом через 16-разрядный / 32-разрядный регистр (в регистре TIMx\_PSC). Он может изменяться «на лету», поскольку этот регистр управления буферизуется. Новый коэффициент предварительного делителя принимается во внимание при следующем событии обновления.

В режиме пересчета счетчик подсчитывает от 0 до значения автоматической перезагрузки (содержимое регистра TIMx\_ARR), затем перезапускается с 0 и генерирует событие переполнения счетчика. Когда происходит событие обновления, все регистры обновляются и устанавливается флаг обновления (бит UIF в регистре TIMx\_SR) (в зависимости от бита URS).

### **Модуль контроля прерываний**

NVIC (Nested vectored interrupt controller) – модуль контроля прерываний. Он выполняет следующие функции:

- разрешает / запрещает прерывание;

- назначает приоритет прерываний (от 0 до 15. 0 – максимальной приоритет, 15 – минимальный приоритет);

- автоматически сохраняет данные при выполнении одиночных или вложенных прерываний.

Этот модуль управляет всеми прерываниями: и внешними и прерываниями модулей самого контроллера. Прерывание с большим приоритетом может прервать выполнение обработки прерывания с меньшим приоритетом.

### **Внешние прерывания**

Внешние прерывания могут вызываться при изменении состояния логического сигнала на входе микроконтроллера. Внешние прерывания используют, когда нужно следить за изменением входного сигнала и оперативно реагировать. Когда требуется такой функционал, необходимо настроить нужный порт (pin), настраиваем внешнее прерывание (по спаду, по фронту или по обоим фронтам) и, как только произойдет смена состояния сигнала, будет вызван обработчик прерывания.

Любые линии портов ввода-вывода STM32 могут работать с внешними прерываниями. В STM32F303 всего 36 линий внешних прерываний. Для работы с пинами используются EXTI0 ... EXTI15.

### **Рекомендация к выполнению лабораторной работы**

Для выполнения данной лабораторной работы рекомендуется следовать данному алгоритму.

- разрешить тактирование необходимых устройств;
- провести необходимые настройки для генерации прерывания от таймера (CNT, ARR, PSC);
- провести необходимые настройки для генерации внешнего прерывания (IMR, RTSR, RTSR);
- написать функцию обработчик прерываний для требуемого устройства.

Для таймера:

```
void TIMx_IRQHandler(void)
{
};
```

Для генерации внешнего прерывания от джойстика:

```
void EXTI15_10_IRQHandler(void)
{
};
```

- используя регистр NVIC, установить требуемые приоритеты для прерываний.

## **Лабораторная работа №3**

### **Цель работы**

Реализовать шим-сигнал в микроконтроллерах на примере STM32.

### **Задание**

Настроить таймер для генерации ШИМ-сигнала

Описание регистров приведено в Приложении В.

### **Краткие теоретические сведения**

Широтно-Импульсная модуляция, или ШИМ - это способ управления подачей мощности к нагрузке. Управление заключается в изменении длительности импульса при постоянной частоте следования импульсов.

При разработке цифровых устройств с ШИМ следует помнить, что чем больше разрядность ШИМ, тем больше градаций импульса можно получить и тем выше частота работы счётчика.

### **Рекомендация к выполнению лабораторной работы**

Для выполнения данной лабораторной работы рекомендуется следовать данному алгоритму:

- Разрешить тактирование требуемых устройств
- Выполнить настройку таймера
- Настроить отображение результатов работы

## **Лабораторная работа №4**

Описание регистров приведено в Приложении Г.

### **Краткие теоретические сведения**

АПЦ - это аналого-цифровой преобразователь или ADC (Analog-to-Digital Converter). То есть специальное устройство, которое преобразует аналоговый сигнал в цифровой.

АЦП применяется в цифровой технике. В частности, почти все современные микроконтроллеры имеют встроенный АЦП.

Следует отметить, что существует и устройство с обратной функцией, называемое цифро-аналоговым преобразователем (ЦАП, DAC). Оно позволяет переводить цифровой сигнал в аналоговый. Например, во время проигрывания мелодии на мобильном телефоне происходит преобразование цифрового кода из MP3 файла в звук, который вы слышите у себя в наушниках.

### **Принцип работы АЦП**

1. Прежде всего, мы имеем некоторое воздействие и предмет, на которое оно производится (в нашем случае тепло воздействует на датчик). Единица измерения на этом этапе — физическая величина градусы Цельсия, Кельвина).

2. Под этим воздействием предмет — датчик — меняет свои параметры. На этом этапе мы имеем зависимость параметра датчика от воздействия.

3. У разных датчиков, разумеется, разные параметры. Поэтому на следующем этапе их всех приводят к единому «интерфейсу» — к напряжению. Другими словами, теперь мы имеем зависимость напряжения от воздействия (исключения — например, емкость может быть измерена микроконтроллером без преобразования в напряжение).

4. Далее необходимо перевести напряжение в цифровой код, который получит микроконтроллер. Этим занимается АЦП. В подавляющем большинстве ваших устройств на вход им будет подаваться напряжение в диапазоне 0 — 2.5... 5В, на выходе будет появляться число в виде, доступном микроконтроллеру.

### **Характеристики АЦП**

#### **1. Разрядность**

Разрядность АЦП характеризует количество дискретных значений, которые преобразователь может выдать на выходе. То есть количество битов в числе, которое получается на выходе АЦП.

Чем больше значение этого параметра, тем точнее работает прибор. Предположим, что у нас есть АЦП с разрядностью 1. Подавая на вход любое напряжения от 0 до 2,5 В, на выходе мы получим 0. Любое же напряжение от 2,5 до 5 В даст нам единицу. То есть одноканальный АЦП сможет распознать только два уровня напряжения. Скорость преобразования.

#### **2. Опорное напряжение**

Опорное напряжение определяет границу диапазона, с которым будет работать АЦП.

Диапазон от 0 до 5 вольт, в котором работает устройство, в общем случае этот диапазон выглядит как: от 0 до опорного напряжения. Тогда формула расчета точности АЦП будет такова: точность = опорное напряжение/1024.

#### **3. Разрешающая способность**

Разрешающая способность - величина, обратная максимальному числу кодовых комбинаций на выходе АЦП. Разрешающая способность выражается в процентах, разрядах или децибелах и характеризует потенциальные возможности АЦП с точки зрения достижимой точности.



## **Список использованной литературы**

1. Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3
2. STMicroelectronics Reference manual RM0316 (англ.) , 2016
3. STMicroelectronics User manual UM1570 (англ.) , 2016
4. Мартин М. Инсайдерское руководство по STM32
5. An Engineer's Introduction to the STM32 Series: The Insider Guide to the STM32 ARM Based Microcontroller (англ.) – Hitex Developments Tools: 2016

## Clock control register (RCC\_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	PLL RDY	PLLON	Res	Res	Res	Res	CSS ON	HSE BYP	HSE RDY	HSE ON
						r	nw					nw	nw	r	nw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res	HSI RDY	HSION
r	r	r	r	r	r	r	r	nw	nw	nw	nw	nw		r	nw

Рис.1 Регистр RCC\_CR

Bit 25 **PLL RDY**: PLL флаг готовности

Установлено аппаратными средствами, для указания состояния PLL

0: PLL заблокирован

1: PLL разблокирован

Bit 24 **PLLON**: PLL

0: PLL выключен

1: PLL включен

Bit 1 **HSI RDY**: HSI флаг готовности

Установлено аппаратными средствами, для указания состояния HSI генератора

0: HSI - генератор не готов

1: HSI - генератор готов

Bit 0 **HSION**: включение генератора HSI

0: выключить генератор HSI

1: включить генератор HSI 1

## Clock configuration register (RCC\_CFGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLNO DIV	MCOFRE[2:1]		MCOF/MCOP RED	Res	MCO[2:0]			CSRSRC	USBSPE	PLLMUL[3:0]			PLL XTPRE	PLL SRC	
nw	nw	nw	r/nw		nw	nw	nw	nw	nw	nw	nw	nw	nw	nw	nw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLSRC <sup>(1)</sup>	Res	PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]			SWS[1:0]		SW[1:0]		
nw		nw	nw	nw	nw	nw	nw	nw	nw	nw	r	r	nw	nw	

Рис.2 Регистр RCC\_CFGR

Bits 26:24 **MCO**: Выходной сигнал микроконтроллера

Устанавливаются и очищаются программно

000: Выход закрыт

001: Зарезервировано

010: Выбраны сигнал LSI

011: Выбраны сигнал LSE

100: Выбраны системные часы

101: Выбраны сигнал HSI

110: Выбраны сигнал HSE

111: Выбраны сигнал PLL clock selected (делится на 1 или 2 в зависимости от значения бита PLLNODIV).

Bits 21:18 **PLLMUL**: PLL multiplication factor

Эти биты записываются программным обеспечением для определения коэффициента умножения PLL. Эти биты могут быть записаны только тогда, когда PLL отключена. Внимание: выходная частота PLL не должна превышать 72 МГц. 0000: PLL умножение сигнала на 2

0001: PLL умножение сигнала на 3

0010: PLL умножение сигнала на 4

0011: умножение сигнала на 5

0100: умножение сигнала на 6

0101: умножение сигнала на 7

0110: умножение сигнала на 8

0111: умножение сигнала на 9

1000: умножение сигнала на 10

1001: умножение сигнала на 11

1010: умножение сигнала на 12

1011: умножение сигнала на 13

1100: умножение сигнала на 14

1101: умножение сигнала на 15

1110: умножение сигнала на 16

1111: умножение сигнала на 17

Bits 3:2 **SWS**: Состояние переключателя системных часов

Установка и очищение битов, для указания, какой источник синхронизации используется в качестве системных часов.

00: HSI-генератор используется в качестве системных часов

01: HSE-генератор используется в качестве системных часов

10: PLL используется в качестве системных часов

11: Непригодный

Bits 1:0 **SW**: Переключатель системных часов

Установка и очищение битов для выбора источника SYSCLK

00: HSI выбран как системные часы

01: HSE выбран как системные часы

10: PLL выбран как системные часы

11: Не разрешен

**GPIO port input data register (GPIOx\_IDR) (x = A..H)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Рис.3 Регистр GPIOx\_IDR

Bits 15:0 **IDRy**: Конфигурация порта ввода y (y = 0..15)

### GPIO port output data register (GPIOx\_ODR) (x = A..H)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ODR31	ODR30	ODR29	ODR28	ODR27	ODR26	ODR25	ODR24	ODR23	ODR22	ODR21	ODR20	ODR19	ODR18	ODR17	ODR16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.4 Регистр GPIOx\_ODR

Bits 15:0 **ODRy**: Конфигурация порта вывода y (y = 0..15)

### GPIO port pull-up/pull-down register (GPIOx\_PUPDR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.5 Регистр GPIOx\_PUPDR

**PUPDRy[1:0]**: Конфигурация порта y (y = 0..15)

Эти биты записываются программно, для подтягивания ножек портов вводы/вывода

00: Нет подтягивания

01: Подтягивание ножки к верху

10: Подтягивание ножки к низу

11: Зарезервировано

### GPIO port mode register (GPIOx\_MODER) (x=A..H)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.6 Регистр GPIOx\_MODER

**MODERy[1:0]**: Конфигурация порта y (y = 0..15)

Эти биты записываются программно, для конфигурации режимов портов вводы/вывода

00: Режим входа

01: Режим выхода общего назначения

10: Альтернативная функция

## AHB peripheral clock enable register (RCC\_AHBENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	ADC34 EN	ADC12EN	Res	Res	Res	TSCEN	IOPG EN <sup>(1)</sup>	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	IOPH EN <sup>(1)</sup>
		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	CRC EN	FMC EN <sup>(1)</sup>	FLITF EN	Res	SRAM EN	DMA2 EN	DMA1 EN
									rw	rw	rw		rw	rw	rw

Рис.7 Регистр RCC\_AHBENR

Bit 22 **IOPFEN**: I/O Тактирование порта F  
Устанавливаются и очищаются программно  
0: I/O тактирование выключено  
1: I/O тактирование включено

Bit 21 **IOPEEN**: I/O Тактирование порта E  
Устанавливаются и очищаются программно  
0: I/O тактирование выключено  
1: I/O тактирование включено.

## Приложение Б

## APB1 peripheral clock enable register (RCC\_APB1ENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	I2C3 EN	DAC1 EN	PWR EN	Res	DAC2 EN	CAN EN	Res	USB EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Res
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Res	Res	WWDG EN	Res	Res	Res	Res	Res	TIM7E N	TIM6EN	Res	TIM4EN	TIM3EN	TIM2 EN
rw	rw			rw						rw	rw		rw	rw	rw

Рис. 1 Регистр RCC\_APB1ENR

**TIM2EN**: тактирование таймера TIM2  
Устанавливается и очищается программно  
0: TIM2 тактирование выключено  
1: TIM2 тактирование включено

## APB2 peripheral clock enable register (RCC\_APB2ENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIM20 EN	Res	TIM17 EN	TIM16 EN	TIM15 EN
											rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4E N	USART 1EN	TIM8 EN	SPI1 EN	TIM1 EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SYS CFGEN
rw	rw	rw	rw	rw											rw

Рис. 2 Регистр RCC\_APB2ENR

Bit 0 **SYS\_CFGEN**: тактирование SYSCFG  
Устанавливается и очищается программно  
0: SYSCFG тактирование выключено  
1: SYSCFG тактирование включено

### APB2 peripheral reset register (RCC\_APB2RSTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIM20 RST	Res	TIM17 RST	TIM16 RST	TIM15 RST
													RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4RST	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SYS CFG RST
RW	RW	RW	RW	RW											RW

Рис. 3 Регистр RCC\_APB2RSTR

Bit 0 **SYSCFGRST**: Сброс компараторов и операционных усилителей  
Устанавливается и очищается программно

0: нет эффекта

1: Сброс значений регистров SYSCFG, COMP, OPAMP

### APB1 peripheral reset register (RCC\_APB1RSTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	I2C3 RST(1)	DAC1 RST	PWR RST	Res	DAC2RST	CAN RST	Res	USB RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	USART3 RST	USART2 RST	Res
		RW	RW		RW	RW		RW	RW	RW	RW	RW	RW	RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Res	Res	WWDG RST	Res	Res	Res	Res	Res	TIM7 RST	TIM6 RST	Res	TIM4 RST	TIM3 RST	TIM2 RST
RW	RW			RW						RW	RW		RW	RW	RW

Рис. 4 Регистр RCC\_APB1RSTR

Bit 0 **TIM2RST**: TIM2 сброс таймера

Устанавливается и очищается программно

0: нет эффекта

1: сброс TIM2

### TIMx status register (TIMx\_SR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	Res	Res	TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			TC_WD	TC_WD	TC_WD	TC_WD			TC_WD		TC_WD	TC_WD	TC_WD	TC_WD	TC_WD

Рис. 5 Регистр TIMx\_SR

Bit 0 **UIF**: Обновление флага прерываний

Устанавливается аппаратным обеспечением при обновлении события. Очищается программно.

This bit is set by hardware on an update event. It is cleared by software.

0: No update occurred

1: Ожидание обновления прерывания. Этот бит устанавливается аппаратным обеспечением при обновлении регистров:

- при переполнении (для TIM2 - TIM4), и, если UDIS = 0 в регистре TIMx\_CR1;
- когда CNT повторно инициализируется программным обеспечением, использующим бит UG в регистре TIMx\_EGR, если URS = 0 и UDIS = 0 в регистре TIMx\_CR1.

### TIMx control register 1 (TIMx\_CR1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE- MAP	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 6 Регистр TIMx\_CR1

Bit 0 **CEN**: Включение счетчика таймера

0: счетчик выключен

1: счетчик включен

*Примечание. Внешние часы, при стробированном режиме и режиме энкодера могут работать только в том случае, если бит CEN ранее установлен программным обеспечением. Однако, триггерный режим может автоматически установить бит CEN аппаратным обеспечением.*

*CEN автоматически очищается в одноимпульсном режиме, когда происходит событие обновления.*

### TIMx event generation register (TIMx\_EGR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

Рис. 7 Регистр TIMx\_EGR

Bit 0 **UG**: Генерация обновления

Этот бит может быть установлен программно. Очищается автоматически аппаратно.

0: Действие не задано.

1: Повторно инициализирует счетчик и обновляет регистры. Обратите внимание, что счетчик предварительного делителя также очищается (в любом случае коэффициент предварительного делителя не изменяется). Счетчик очищается, если выбран режим выравнивания по центру или DIR = 0 (пересчет), иначе он принимает значение автоматической перезагрузки (TIMx\_ARR), если DIR = 1 (при счете «вниз»).

### TIMx DMA/interrupt enable register (TIMx\_DIER)

Адрес смещения: 0x0C

Значение сброса: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Рис. 8 Регистр TIMx\_DIER

Bit 0 **UIE**: Обновление прерывания

0: Запретить генерацию прерываний.

1: Разрешить генерацию прерываний.

### TIMx prescaler (TIMx\_PSC)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 9 Регистр TIMx\_PSC

Bits 15:0 **PSC[15:0]**: Значение предделителя

Частота счетчика CK\_CNT равна частоте fCK\_PSC / (PSC[15:0] + 1).

PSC содержит значение, которое должно быть загружено в активном регистре предделителя при каждом событии обновления (в том числе, когда счетчик очищается через бит UG регистра TIMx\_EGR или через триггерный контроллер при настройке в «режиме сброса»)

### **TIMx auto-reload register (TIMx\_ARR)**

Адрес смещения: 0x2C

Значение сброса: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16] (depending on timers)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 10 Регистр TIMx\_ARR

Bits 31:16 **ARR[31:16]**: Верхнее значение перезагрузки (on TIM2 )

Bits 15:0 **ARR[15:0]**: Нижнее значение перезагрузки предделителяLow Auto-reload Prescaler value

ARR - это значение, которое должно быть загружено в настоящий регистр автоматической перезагрузки. Счетчик блокируется, когда значение автоматической перезагрузки равно нулю.

### **SYSCFG external interrupt configuration register 3 (SYSCFG\_EXTICR3)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 11 Регистр SYSCFG\_EXTICR3

Bits 11:8 **EXTI10[3:0]**: EXTI 10 биты конфигурации

x000: PA[10] pin

x001: PB[10] pin

x010: PC[10] pin

x011: PD[10] pin

x100: PE[10] pin

x101: PF[10] pin

x110: PG[10] pin

Остальные конфигурации зарезервированы.

## **Приложение В**

### **GPIO alternate function low register (GPIOx\_AFRL) (x = A..H)**

Смещение адреса: 0x20

Значение сброса: 0x0000 0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.1 Регистр GPIOx\_AFRL

Биты 31:0 **AFRy[3:0]**: Альтернативный выбор функции для порта x pin y (y = 0..7).

Эти биты записываются программным обеспечением для настройки альтернативных функций ввода / вывода.

Выбор AFRy:

0000: AF0

0001: AF1

0010: AF2

0011: AF3

0100: AF4

0101: AF5

0110: AF6

0111: AF7

1000: AF8

1001: AF9

1010: AF10

1011: AF11

1100: AF12

1101: AF13

1110: AF14

1111: AF15

### GPIO alternate function high register (GPIOx\_AFRH) (x = A..H)

Смещение адреса: 0x24

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.2 Регистр GPIOx\_AFRH

Биты 31:0 **AFRy[3:0]**: Альтернативный выбор функции для порта x pin y (y = 8..15)

Эти биты записываются программным обеспечением для настройки альтернативных функций ввода / вывода.

Выбор AFRy:

0000: AF0

0001: AF1

0010: AF2

0011: AF3  
 0100: AF4  
 0101: AF5  
 0110: AF6  
 0111: AF7  
 1000: AF8  
 1001: AF9  
 1010: AF10  
 1011: AF11  
 1100: AF12  
 1101: AF13  
 1110: AF14  
 1111: AF15

### Capture/compare mode register 1 (TIMx\_CCMR1)

Смещение адреса: 0x18

Значение сброса: 0x0000 0000

Каналы могут использоваться при входе (режиме захвата) или при выводе (режим сравнения). Направление канала определяется путем конфигурирования соответствующих бит CCxS. Все остальные биты этого регистра имеют различную функцию во входном и выходном режимах. Для данного бита, OCxх описывает свою функцию, когда канал настроен на выходе, ICxх описывает его когда канал настроен на вход. Поэтому вы должны проследить, чтобы один и тот же бит мог иметь другое значение при входе и выходе.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

Рис.3 Регистр TIMx\_CCMR1

Биты 6:4 **OC1M[2:0]**: Выходное сравнение 1 режима (биты от 2 до 0)

Эти биты определяют поведение сигнала выходного OC1REF, из которого получены OC1 и OC1N. OC1REF является активным, в то время как активный уровень OC1 и OC1N зависит от бит CC1P и CC1NP.

0000: Frozen - Сравнение между выходным регистром сравнения TIMx\_CCR1 и счетчик TIMx\_CNT не влияет на выходы.

0001: Установите канал 1 на активный уровень при совпадении. Сигнал OC1REF принудительно увеличивается, когда счетчик TIMx\_CNT соответствует регистру присвоения / сравнения 1 (TIMx\_CCR1).

0010: Установите канал 1 на неактивный уровень при совпадении. Сигнал OC1REF принудительно уменьшается, когда счетчик TIMx\_CNT соответствует регистру присвоения / сравнения 1 (TIMx\_CCR1).

0011: Переключатель - OC1REF переключается, когда TIMx\_CNT = TIMx\_CCR1.

0100: Принудительно неактивный уровень - OC1REF принудительно разряжен.

0101: Принудительно активный уровень - OC1REF принудительно поднят.

0110: Режим PWM 1 - Канал 1 активен, пока TIMx\_CNT <TIMx\_CCR1 иначе неактивен.

0111: Режим PWM 2 - Канал 1 неактивен, пока TIMx\_CNT <TIMx\_CCR1 иначе активен.

Все остальные значения: зарезервированы.

*Примечание: 1: эти биты не могут быть изменены, пока запрограммирован 3 уровень блокировки LOCK (LOCK биты в TIMx\_BDTR регистре) и CC1S='00' (канал сконфигурирован на выходе).*

*Примечание: 2: В режиме PWM 1 или 2 уровни OCREF изменяется только тогда, когда результат*

*Сравнения изменяется или когда режим сравнения на выходе переключается с «замороженного» режима в режим «PWM».*

### Capture/compare register 1 (TIMx\_CCR1)

Смещение адреса: 0x34

Значение сброса: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.4 Регистр TIMx\_CCR1

Биты 15:0 CCR1[15:0]: Присвоение / Сравнение 1 значения

**Если канал CC1 сконфигурирован как выход:** CCR1 - это значение, которое должно быть загружено в данный регистр присвоения / сравнения 1 (значение предварительной загрузки).

Он загружается постоянно, если функция предварительной загрузки не выбрана в регистре TIMx\_CCMR1 (бит OC1PE). Кроме того, значение предварительной загрузки копируется в активный регистр присвоения / сравнения 1 при возникновении события обновления.

Активный регистр присвоения / сравнения содержит значение, которое будет сравниваться со счетчиком TIMx\_CNT и появляться на выходе OC1.

**Если канал CC1 сконфигурирован как вход:** CR1 - значение счетчика, переданное последним входным событием 1 (IC1).

### Capture/compare enable register (TIMx\_CCER)

Смещение адреса: 0x20

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.5 Регистр TIMx\_CCER

Бит 3 **CC1NP**: Присвоить/сравнить 1 дополнительную выходную полярность

Канал **CC1**, настроенный на выход:

0: **OC1N** высокоактивный

1: **OC1N** малоактивный

Канал **CC1**, настроенный на вход:

Этот бит используется в сочетании с **CC1P** для определения полярности **TI1FP1** и **TI2FP1**. Сммотри описание **CC1P**.

*Примечание: 1. Этот бит не доступен для записи, пока LOCK уровень 2 или 3 запрограммирован (LOCK бит в регистре TIMx\_BDTR) и CC1S = "00" (канал сконфигурирован на выходе). 2. В каналах с дополнительным выходом этот бит предварительно загружается. Если бит **CCPC** установлен в регистре TIMx\_CR2, тогда активный бит **CC1NP** принимает новое значение из предварительно загруженного бита только тогда, когда сгенерируется событие коммутации.*

Бит 1 **CC1P**: Присвоение / Сравнить 1 выходная полярность

**Канал CC1, настроенный как выход:**

0: активный уровень **OC1**

1: малоактивный уровень **OC1**

**Канал CC1, настроенный как вход:**

Биты **CC1NP** / **CC1P** выбирают полярность **TI1FP1** и **TI2FP1** для запуска или присвоения операции.

00: Не инвертируемый/растущий фронт. Схема чувствительна к нарастающему фронту **TIxFP1** (захват или триггерные операции в режиме сброса, внешний тактовый или триггерный режим), **TIxFP1** не инвертируется (запуск триггера в закрытом режиме).

01: Инвертируемый / задний фронт. Схема чувствительна к фронту падения **TIxFP1** (операции захвата или триггера при сбросе, внешний тактовый или триггерный режим), **TIxFP1** инвертирован (операция триггера в закрытом режиме).

10: Зарезервирован, не используйте эту конфигурацию.

1: Не инвертируемый / оба фронта. Схема чувствительна к восходящим и спадающим фронтам **TIxFP1** (операции захвата или триггера при сбросе, внешнем такте или режиме триггера), **TIxFP1** не инвертируется (запуск триггера в закрытом режиме).

*Примечание: 1. Этот бит не доступен для записи, когда LOCK на уровне 2 или 3 запрограммирован (бит LOCK в регистре TIMx\_BDTR).*

*Примечание: 2. В каналах с дополнительным выходом этот бит предварительно загружается. Если бит **CCPC** устанавливается в регистре TIMx\_CR2, тогда активный бит **CC1P** принимает новое значение из предварительно загруженного бита, только когда генерируется событие коммутации.*

## Break and dead-time register (TIMx\_BDTR)

Адрес смещения: 0x44

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.6 Регистр TIMx\_BDTR

*Примечание: 1. Поскольку биты BK2P, BK2E, BK2F [3: 0], BKF [3: 0], AOE, BKP, BKE, OSSI, OSSR и DTG [7: 0] могут быть заблокированы в зависимости от конфигурации LOCK, может потребоваться настройка во время первого доступа на запись в регистр TIMx\_BDTR.*

Бит 15 **МОЕ**: Включение главного выхода

Этот бит очищается асинхронно с помощью аппаратного обеспечения, как только вход прерывания активен. Он устанавливается программным обеспечением или автоматически в зависимости от бита AOE. Он действует только на каналы, настроенные на выходе.

0: выходы OC и OCN отключены или принудительно переведены в режим ожидания в зависимости от бит OSSI.

1: выходы OC и OCN активируются, если установлены их соответствующие разрешающие биты (CCxE, CCxNE в регистре TIMx\_CCER)

### **GPIO port input data register (GPIOx\_IDR) (x = A..H)**

Адрес смещения: 0x10

Значение сброса: 0x0000 XXXX (где X означает не определён)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Рис.7 Регистр GPIOx\_IDR

Биты 31:16 Заняты, должны сохраняться при сбросе.

Биты 15:0 **IDRy**: **Бит входных данных порта** (y = 0..15)

Эти биты доступны только для чтения. Они содержат входное значение соответствующего порта ввода-вывода.

## **Приложение Г**

### **ADC sample time register 2 (ADCx\_SMPR2, x=1..4)**

Смещение адреса: 0x18

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0		SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.1 Регистр ADCx\_SMPR2

Биты 31:27 Зарезервированы, должны храниться при сбросе значения

Биты 26:0 **SMPx[2:0]**: Канал x времени выборки

Эти биты написаны ПО для выбора времени выборки отдельно для каждого канала. Во время дискреционных циклов, биты выборки каналов должны оставаться неизменными.

000: 1.5 АЦП тактовые циклы

001: 2.5 АЦП тактовые циклы

010: 4.5 АЦП тактовые циклы

011: 7.5 ADC тактовые циклы

100: 19.5 ADC тактовые циклы

101: 61.5 ADC тактовые циклы

110: 181.5 ADC тактовые циклы

111: 601.5 ADC тактовые циклы

*Примечание: 1. ПО разрешено записать значения в эти биты только тогда, когда ADSTART=0 и JADSTART=0 (которые гарантируют, что никакое преобразование не будет продолжаться).*

#### ADC configuration register (ADCx\_CFGR, x=1..4)

Смещение адреса: 0x0C

Значение сброса: 0x0000 00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWD1CH[4:0]					JAUTO	JAWD1 EN	AWD1 EN	AWD1S GL	JQM	JDISC EN	DISCNUM[2:0]		DISC EN	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AUT DLY	CONT	OVR MOD	EXTEN[1:0]		EXTSEL[3:0]				ALIGN	RES[1:0]		Res.	DMA CFG	DMA EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

Рис.2 Регистр ADCx\_CFGR

Биты 9:6 **EXTSEL[3:0]**: Выбор внешнего триггера для стандартных групп. Эти биты выбирают как внешнее событие будет использовано для запуска начала

преобразований стандартной группы:

0000: Событие 0

0001: Событие 1

0010: Событие 2

0011: Событие 3

0100: Событие 4

0101: Событие 5

0110: Событие 6

0111: Событие 7

...

1111: Событие 15

*Примечание: 1. ПО разрешено записывать значения в эти биты только тогда, когда ADSTART=0 (который гарантирует, что преобразование не будет продолжаться).*

Бит 13 **CONT**: Режим однократного/ непрерывного преобразования для стандартных конверсий.

Этот бит может быть установлен и очищен только с помощью ПО. Если он установлен, преобразование происходит непрерывно, пока оно не будет очищено.

0: режим однократного преобразования

1: Режим непрерывного преобразования

*Примечание: 1. Невозможно включить как прерывистый режим, так и непрерывный режим: запрещено устанавливать оба параметра DISCEN = 1 и CONT = 1.*

*Примечание: 2. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечание: 3. Когда активирован двойной режим (бит DUAL регистра ADCx\_CCR не равен нулю), бит CONT подчиненного АЦП больше не доступен для записи, а его значение эквивалентно значению биту CONT основного АЦП.*

Бит 16 **DISCEN** : Режим прерывания для стандартных каналов

Этот бит устанавливается и очищается ПО для включения / выключения режима прерывания для стандартных каналов.

0: прерывистый режим для обычных каналов отключен

1: прерывистый режим для обычных каналов включен

*Примечание: 1. Невозможно включить как прерывистый режим, так и непрерывный режим: запрещено устанавливать оба параметра  $DISCEN = 1$  и  $CONT = 1$ .*

*Примечание: 2. Бит  $DISCEN$  и  $JDISCEN$  должны быть очищены программным обеспечением, если установлен  $JAUTO$ .*

*Примечание: 3. ПО разрешено записывать значения в этот бит только тогда, когда  $ADSTART=0$  (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечание: 4. Когда активирован двойной режим (бит  $DUAL$  регистра  $ADCx\_CCR$  не равен нулю), бит  $DISCEN$  подчиненного АЦП больше не доступен для записи, а его значение эквивалентно значению биту  $DISCEN$  основного АЦП.*

**Бит 23:  $AWD1EN$ :** Аналоговый сторожевой таймер 1 активируется на стандартных каналах

Этот бит устанавливается и очищается ПО.

0: Аналоговый сторожевой таймер 1 отключен на обычных каналах

1: Аналоговый сторожевой таймер 1 включен на обычных каналах

*Примечание: 1. ПО разрешено записывать значения в этот бит только тогда, когда  $ADSTART=0$  (который гарантирует, что регулярное преобразование не будет продолжаться).*

**Биты 30:26  $AWD1CH[4:0]$ :** Выбор канала аналогового сторожевого таймера 1

Эти биты устанавливаются и очищаются программным обеспечением. Они выбирают входной канал, который должен охранять аналоговый сторожевой таймер.

00000: зарезервировано (аналоговый входной канал 0 не отображается)

00001: аналоговый входной канал-1 АЦП, контролируемый  $AWD1$

...

10010: аналоговый входной канал-18 АЦП, контролируемый  $AWD1$

остальные: зарезервированы, не должны использоваться

**ADC regular sequence register 1 ( $ADCx\_SQR1$ ,  $x=1..4$ )**

Смещение адреса: 0x30

Значение сброса: 0x0000 0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2[3:0]				Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw	rw	rw	rw

Рис.3 Регистр ADCx\_SQR1

Биты 31:29 Зарезервировано, должно храниться при значении сброса.

Биты 28:24 **SQ4[4:0]** : Четвёртое преобразование в последовательности

Эти биты записываются ПО с номером канала (1...18), в качестве четвертого преобразования в последовательности преобразований.

*Примечание: 1. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечание: 2. Аналоговый входной канал 0 не отображается: значение «00000» не должно использоваться*

Бит 23 Зарезервировано, должно храниться при значении сброса.

Биты 22:18 **SQ3[4:0]** : Третье преобразование в последовательности

Эти биты записываются ПО с номером канала (1...18), в качестве третьего преобразования в последовательности преобразований.

*Примечание: 1. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечани: 2. Аналоговый входной канал 0 не отображается: значение «00000» не должно использоваться*

Бит 17 Зарезервировано, должно храниться при значении сброса.

Биты 16:12 **SQ2[4:0]** : Второе преобразование в последовательности

Эти биты записываются ПО с номером канала (1...18), в качестве второго преобразования в последовательности преобразований.

*Примечание: 1. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечание: 2. Аналоговый входной канал 0 не отображается: значение «00000» не должно использоваться*

Бит 11 Зарезервировано, должно храниться при значении сброса.

Бит 10:6 **SQ1[4:0]** : Первое преобразование в последовательности

Эти биты записываются ПО с номером канала (1...18), в качестве первого преобразования в последовательности преобразований.

*Примечание: 1. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

*Примечание: 2. Аналоговый входной канал 0 не отображается: значение «00000» не должно использоваться*

Биты 5:4 Зарезервировано, должно храниться при значении сброса.

Биты 3:0 **L[3:0]** : Постоянная длина последовательности каналов

Эти биты записываются программным обеспечением для определения общего количества преобразований в преобразованиях канала.

0000: 1 конверсия

0001: 2 конверсия

...

1111: 16 конверсия

*Примечание.: 1. ПО разрешено записывать значения в этот бит только тогда, когда ADSTART=0 (который гарантирует, что регулярное преобразование не будет продолжаться).*

Каналы SQR2, SQR3, SQR4 данные регистры необходимо обнулить в 4-ой лабораторной работе, чтобы работать непосредственно с каналом SQR1.

#### **ADC common control register (ADCx\_CCR, x=12 or 34)**

Адрес смещения: 0x08 (этот адрес смещения относится к базовому адресу основного АЦП + 0x300)

Значение сброса: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBAT EN	TS EN	VREF EN	Res.	Res.	Res.	Res.	CKMODE[1:0]	
							rw	rw	rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDMA[1:0]		DMA CFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
rw	rw	rw		rw	rw	rw	rw				rw	rw	rw	rw	rw

Рис.4 Регистр ADCx\_CCR

Бит 23 **TSEN**: Датчик температуры

Этот бит устанавливается и очищается программным обеспечением для включения

/ выключения канала датчика температуры.

0: канал датчика температуры отключен

1: канал датчика температуры включен

*Примечание: 1. Программе разрешено записывать этот бит только тогда, когда АЦП отключено ( $ADCAL = 0$ ,  $JADSTART = 0$ ,  $ADSTART = 0$ ,  $ADSTP = 0$ ,  $ADDIS = 0$  и  $ADEN = 0$ ).*

#### ADC control register (ADCx\_CR, x=1..4)

Адрес смещения: 0x08

Значение сброса: 0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD CAL	ADCA LDIF	ADVREGEN[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JAD STP	AD STP	JAD START	AD START	AD DIS	AD EN
										rs	rs	rs	rs	rs	rs

Рис.5 Регистр ADCx\_CR

Бит 0: **ADEN**: Управление АЦП

Этот бит устанавливается программным обеспечением для включения АЦП. АЦП будет готов к работе после того, как установлен флаг ADRDY.

Бит очищается аппаратным обеспечением, когда АЦП отключен после выполнения команды ADDIS.

0: АЦП отключен (состояние ВЫКЛ)

1: Напишите 1, чтобы включить АЦП

*Примечание: 1. Программному обеспечению разрешено устанавливать ADEN только тогда, когда все биты регистров ADCx\_CR равны 0 ( $ADCAL = 0$ ,  $JADSTART = 0$ ,  $ADSTART = 0$ ,  $ADSTP = 0$ ,  $ADDIS = 0$  и  $ADEN = 0$ ), за исключением бита ADVREGEN, который должен быть равен 1 (и ПО должно ожидать запуск регулятора напряжения).*