Вариант 1

1. Перечислите основные блоки памяти, которые имеются в стенде SDK-1.1M, их назначение, объем и способ доступа к ним.

Характеристика	Type 107	Type 407	Type 427	Type 153
Вычислитель				
Модель	STM32F107 VCT6	STM32F407 VGT6	STM32F427 VIT6	STM32MP153 CAB3
Тактовая частота ядра	72 МГц	168 МГц	180 МГц	до 800 МГц
Встроенная RAM	64 КБ	192 КБ	256 КБ	708 КБ
Внешняя RAM	-	-	-	512 МБ
Встроенная FLASH-память	256 КБ	1 МБ	2 МБ	-
Внешняя FLASH-память	-	16 МБ	16 МБ	1 ГБ
Внешние интерфейсы				
Micro SD	-	да	да	да
USB	да	да	да	да

Table 4. Memory mapping vs. Boot mode/physical remap in STM32F42xxx and STM32F43xxx

Addresses	Boot/Remap in main Flash memory	Boot/Remap in embedded SRAM	Boot/Remap in System memory	Remap in FMC
0x2002 0000 - 0x2002 FFFF	SRAM3 (64 KB)	SRAM3 (64 KB)	SRAM3 (64 KB)	SRAM3 (64 KB)
0x2001 C000 - 0x2001 FFFF	SRAM2 (16 KB)	SRAM2 (16 KB)	SRAM2 (16 KB)	SRAM2 (16 KB)
0x2000 0000 - 0x2001 BFFF	SRAM1 (112 KB)	SRAM1 (112 KB)	SRAM1 (112 KB)	SRAM1 (112 KB)
0x1FFF 0000 - 0x1FFF 77FF	System memory	System memory	System memory	System memory
0x0810 0000 - 0x0FFF FFFF	Reserved	Reserved	Reserved	Reserved
0x0800 0000 - 0x081F FFFF	Flash memory	Flash memory	Flash memory	Flash memory

Table 4. Memory mapping vs. Boot mode/physical remap in STM32F42xxx and STM32F43xxx (continued)

Addresses	Boot/Remap in main Flash memory	Boot/Remap in embedded SRAM	Boot/Remap in System memory	Remap in FMC
0x0400 0000 - 0x07FF FFFF	Reserved	Reserved	Reserved	FMC bank 1 NOR/PSRAM 2 (128 MB Aliased)
0x0000 0000 - 0x001F FFFF(1)(2)	Flash (2 MB) Aliased	SRAM1 (112 KB) Aliased	System memory (30 KB) Aliased	FMC bank 1 NOR/PSRAM 1 (128 MB Aliased) or FMC SDRAM bank 1 (128 MB Aliased)

Table 6. Flash module - 2 Mbyte dual bank organization (STM32F42xxx and STM32F43xxx)

Block	Bank	Name	Block base addresses	Size
Main memory	Bank 1	Sector 0	0x0800 0000 - 0x0800 3FFF	16 Kbytes
		Sector 1	0x0800 4000 - 0x0800 7FFF	16 Kbytes
		Sector 2	0x0800 8000 - 0x0800 BFFF	16 Kbytes
		Sector 3	0x0800 C000 - 0x0800 FFFF	16 Kbyte
		Sector 4	0x0801 0000 - 0x0801 FFFF	64 Kbytes
		Sector 5	0x0802 0000 - 0x0803 FFFF	128 Kbytes
		Sector 6	0x0804 0000 - 0x0805 FFFF	128 Kbytes
		-	-	-
		-	-	-
		-	-	-
		Sector 11	0x080E 0000 - 0x080F FFFF	128 Kbytes
	Bank 2	Sector 12	0x0810 0000 - 0x0810 3FFF	16 Kbytes
		Sector 13	0x0810 4000 - 0x0810 7FFF	16 Kbytes
		Sector 14	0x0810 8000 - 0x0810 BFFF	16 Kbytes
		Sector 15	0x0810 C000 - 0x0810 FFFF	16 Kbytes
		Sector 16	0x0811 0000 - 0x0811 FFFF	64 Kbytes
		Sector 17	0x0812 0000 - 0x0813 FFFF	128 Kbytes
		Sector 18	0x0814 0000 - 0x0815 FFFF	128 Kbytes
			-	-
			-	-
			-	-
		Sector 23	0x081E 0000 - 0x081F FFFF	128 Kbytes
System memory			0x1FFF 0000 - 0x1FFF 77FF	30 Kbytes
OTP			0x1FFF 7800 - 0x1FFF 7A0F	528 bytes
Option bytes	Bank 1		0x1FFF C000 - 0x1FFF C00F	16 bytes
	Bank 2		0x1FFE C000 - 0x1FFE C00F	16 bytes

Вся память инструкций, память данных, регистры и порты ввода-вывода представлены в линейном адресном пространстве размером 4GiB.
Основная память (RAM) доступна по адресам: 0x2000 - 0x2002 FFFF
FLASH-память доступна по адресам: 0x0800 0000 - 0x081F FFFF

FLASH-память энергонезависимая - содержимое между включениями и выключениями не уничтожается. Предназначена для загрузки и хранения программы и данных.
Основная память используется во время выполнения программы.

- Память:
    - 2 Мбайта FLASH-памяти;
    - 256+4 Кбайт SRAM, включая 64 Кбайт CCM (core coupled memory);
    - контроллер внешней памяти, поддерживающий Compact Flash, SRAM, PSRAM, SDRAM/LPSDR SDRAM, NOR и NAND.

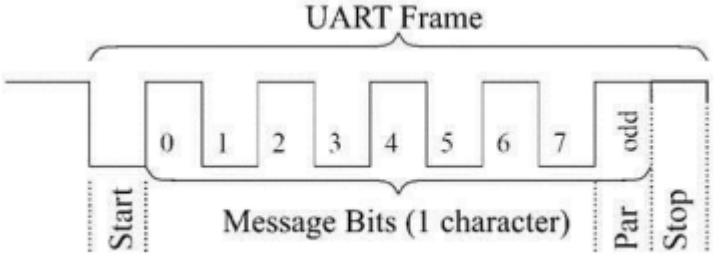
2. Что такое дребезг контактов? Где в SDK-1.1M встречается дребезг контактов? Как устранить влияние дребезга в каждом из этих случаев?

- Механические контакты подвержены явлению дребезга. Его суть в том, что при замыкании контакта нажатием на кнопку напряжение устанавливается не сразу, а в течение некоторого времени (десятки миллисекунд) «скачет», пока контакт надежно не замкнется. После того как кнопка будет отпущена, напряжение также «скачет», пока не установится на нужном уровне. Такое многократное замыкание/размыкание контактов вызвано тем, что контакты пружинят, обгорают и т. п. Поскольку процессор реагирует на события очень быстро, то он может воспринять эти скачки напряжения за несколько нажатий.
- В SDK-1.1M встречается во всех существующих кнопках (матричная клавиатура или боковая кнопка), т.к. никакая защита не предусмотрена.
- Решить данную проблему можно программно посредством отслеживания временных интервалов между замыканием и размыканием контактов. Например, можно использовать небольшую задержку перед следующим опросом кнопки после фиксации замыкания. Задержка подбирается такой, чтобы дребезг успел прекратиться к ее окончанию. Если второй опроса также показал, что контакт замкнут, можно считать, что кнопка нажата. Еще одним вариантом является постоянный периодический опрос состояния сигнала кнопки по прерыванию от таймера (будет рассмотрено в следующих разделах).

3. Опишите протокол передачи данных UART, формат кадра данных и значение его полей. Как работает программа микроконтроллера при обмене данными по UART в режиме опроса и в режиме прерывания?

- UART означает универсальный асинхронный приёмопередатчик и определяет протокол или набор правил для обмена последовательными данными между двумя устройствами. UART — очень простой протокол, в котором используется только два провода между передатчиком и приемником для передачи и приема в обоих направлениях.
- *P.S. Кадр данных лучше описать абзацем про ассинхронку ниже*
- Стандарт UART является чисто асинхронным интерфейсом, но реализующий его контроллер, как правило, может настраиваться в широких пределах и функционировать как в синхронном (USART), так и асинхронном (UART) режимах. Синхронный режим предполагает наличие средств синхронизации передатчика и приемника. Как правило, для синхронизации используют специальную линию для передачи тактовых импульсов (синхросигналов). Информация с линии данных считывается приемником только по синхросигналу.
- В асинхронном режиме при отсутствии передачи на линии данных установлена логическая «1». Перед очередным байтом информации передается специальный старт-бит, сигнализирующий о начале передачи (логический «0»). Затем следуют биты данных (их обычно восемь), за которыми может следовать дополнительный бит (его наличие зависит от режима передачи, обычно этот бит

выполняет функцию контроля четности – «parity bit»). Завершается посылка стоп-битом (логическая «1»), длина которого (длительность единичного состояния линии) может соответствовать длительности передачи 1, 1,5 («полтора стоп-бита») или 2 бит. Стоп-бит гарантирует некоторую выдержку между соседними посылками, при этом пауза между ними может быть сколь угодно долгой (без учета «таймаута»). Если на линии данных долгое время находится логический «0», это считается ошибкой («break»)



к 7 – Временная диаграмма передачи сообщения через UART

Вариант 2

1. Перечислите основные компоненты микроконтроллера, применяемого в стенде SDK-1.1M, и их назначение.

- Процессорное ядро ARM 32-bit Cortex-M4:
    - модуль операций с плавающей запятой; ART-ускоритель, обеспечивающий мгновенное выполнение из FLASH-памяти;
    - частота до 180 МГц;
    - блок защиты памяти (MPU).
- Память:
    - 2 Мбайта FLASH-памяти;
    - 256+4 Кбайт SRAM, включая 64 Кбайт CCM (core coupled memory);
    - контроллер внешней памяти, поддерживающий Compact Flash, SRAM, PSRAM, SDRAM/LPSDR SDRAM, NOR и NAND.
- Параллельный интерфейс LCD-дисплея с режимами 8080 и 6800
- Ускоритель Chrom-ART, расширяющий функции работы с графикой (DMA2D). Управление синхронизацией, сбросом, питанием:
    - напряжение питания и ввода-вывода от 1,7 до 3,6 В;
    - сброс при включении (POR), выключении (PDR) питания, программируемый датчик напряжения (PVD);
    - внешний кварцевый генератор с частотой от 4 до 26 МГц;
    - внешний генератор 32 кГц для часов реального времени (RTC) с калибровкой;
    - встроенные RC-генераторы 16 МГц и 32 кГц;
    - режимы сна, остановки, дежурный режим.
- 3 12-битных АЦП с 2,4 млн выборок/с: до 24 каналов и 7,2 млн выборок/с в режиме тройного чередования.
- 2 12-битных ЦАП.
- Универсальный 16-канальный контроллер прямого доступа к памяти (DMA) с FIFO-буферизацией и поддержкой пакетной передачи.
- Отладочные возможности: интерфейсы JTAG и SWD (serial wire debug); Cortex-M3 Embedded Trace Macrocell.
- До 168 портов ввода-вывода с поддержкой прерываний.
- До 21 интерфейса связи:
    - 3 интерфейса I2C (SMBus/PMBus);
    - 4 USART и 4 UART; – 6 SPI (45 Мбит/с), 2 с полнодуплексным I2S;
    - SAI (serial audio interface); – 2 интерфейса CAN с 512 байт выделенной SRAM;
    - SDIO.
- Высокопроизводительные интерфейсы:
    - контроллер USB 2.0 Full Speed device/host/OTG с PHY;
    - контроллер USB 2.0 High Speed/Full Speed device/host/OTG с выделенной DMA, Full Speed PHY и ULPI;
    - 10/100 Ethernet MAC с выделенной DMA и SRAM (4 Кб), IEEE 1588v2.
- 8–14-битный параллельный интерфейс камеры со скоростью до 54 Мбайт/с.

- Аппаратный генератор случайных чисел, блок вычисления CRC, 96-битный уникальный идентификатор

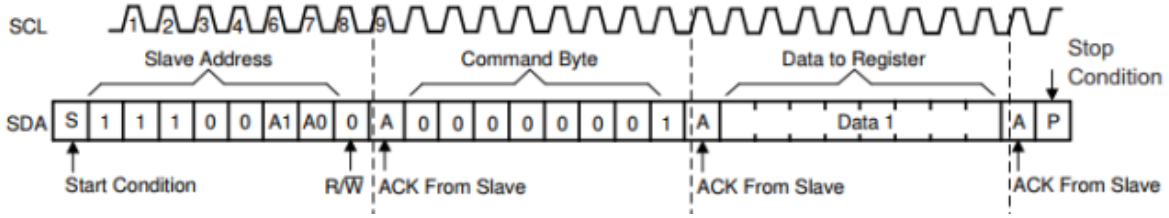
2. Основные принципы устройства подсистемы прерываний в микроконтроллерах. Способы применения прерываний в программировании микроконтроллеров.

- Механизм прерываний реализуется аппаратными и программными средствами: контроллером прерываний и обработчиками прерываний.
- Каждое событие, требующее прерывания, сопровождается сигналом прерывания, оповещающим об этом вычислительную машину. Обработка прерывания выполняется в три основных этапа:
    - Прекращение выполнения текущей программы (сохраняем контекст программы в отдельном стеке)
    - Переход к выполнению программы обработчика (определяется приоритетный источник прерывания и соответствующий вектор прерывания, переход к программе-обработчику прерывания)
    - Возврат управления прерванной программе (восстанавливаем контекст программы)
- С помощью прерываний можно обеспечить куда более плавное и комфортное взаимодействие со стендом за счет отсутствия времени, затраченного на ожидание. Например, мы можем работать с UART6 за счет того, что обработчик прерываний UART, который использует контекст драйвера для решения задач. Команды, такие как HAL_UART_Receive_IT, не ждут конца обмена, а инициализируют обмен и обработку прерываний.

3. Опишите протокол передачи данных I2C, формат кадра данных и значение его полей. Как происходит разрешение конфликтов доступа, если к шине подключено несколько "ведущих" устройств?

- I2C – двухпроводной интерфейс для организации взаимодействия между микросхемами электронных устройств. Во время обмена данными одно устройство играет роль ведущего (master), а второе – ведомого (slave). Допускается одновременное подключение нескольких ведущих устройств к одной шине I 2C.
- Каждый байт, передаваемый по линии SDA, должен состоять из 8 бит. Количество байт, передаваемых за один сеанс связи неограничено. Каждый байт должен оканчиваться битом подтверждения. Данные передаются, начиная с наиболее значащего бита. Если приёмник не может принять еще один целый байт, пока он не выполнит какую-либо другую функцию (например, обслужит внутреннее прерывание), он может удерживать линию SCL в НИЗКОМ состоянии, переводя передатчик в состояние ожидания. Пересылка данных продолжается, когда приёмник будет готов к следующему байту и отпустит линию SCL.
- НА ВСЯКИЙ СЛУЧАЙ:

- Процедура записи регистра. Данные передаются на PCA9538 путем отправки адреса устройства и установки младшего значащего бита в «0». После адресного байта отправляется командный байт, который определяет, какой регистр получает данные, следующие за командным байтом.
- Процедура чтения регистров. Сначала ведущее устройство шины должно отправить адресный байт PCA9538 с младшим значащим битом, установленным в «0» (запись). После адреса отправляется командный байт, и определяет, к какому регистру обращаются. После этого на шине I2C формируется последовательность «повторный старт», снова отправляется адресный байт, но на этот раз младший значащий бит устанавливается в «1» (чтение). После этого данные из регистра, определенного командным байтом, читаются из PCA9538. На количество байтов данных, полученных в одной передаче чтения, ограничения нет (можно прочитать все 4 регистра подряд). Когда ведущее устройство читает последний нужный ему байт, оно не должно подтверждать данные.



- При этом для устранения конфликтов доступа к шине предусмотрен специальный механизм арбитража - Ведущий может начинать пересылку данных только если шина свободна. Два и более ведущих могут сгенерировать сигнал СТАРТ за время минимального удерживания, что ведет к определенному сигналу СТАРТ на шине. Арбитраж происходит на шине SDA, в периоды, когда шина SCL находится в ВЫСОКОМ состоянии. Если один ведущий передает на линию данных НИЗКИЙ уровень, в то время как другой - ВЫСОКИЙ, то последний отключается от линии, так как состояние SDL (НИЗКОЕ) не соответствует ВЫСОКОМУ состоянию его внутренней линии данных. Арбитраж может продолжаться на протяжении нескольких бит. Так как сначала передается адрес, а потом, то арбитраж может продолжаться до окончания адреса, а если ведущие адресуют одно и то же устройство, то в арбитраже будут участвовать и данные. Вследствие такой схемы арбитража при столкновении данные не теряются. Ведущему, проигравшему арбитраж, разрешается выдавать синхроимпульсы на шину SCL до конца байта, в течение которого был потерян доступ.

Вариант 3

1. Перечислите основные компоненты стенда SDK-1.1M и их назначение.

Расширители портов ввода-вывода
PCA9538PW PCA9538PW [9] – это 8-битный расширитель портов ввода-вывода (GPIO) с поддержкой прерываний, подключенный к процессорному модулю по интерфейсу I2C. PCA9538PW включает регистры конфигурации (настройка портов на вход или выход), входного и выходного портов, инверсии полярности. В SDK-1.1M имеется два расширителя, использующихся для подключения различных периферийных устройств и сигналов.

Часы реального времени MCP79411
MCP79411 – часы/календарь, совмещенные с 1 Кбит встроенной энергонезависимой памяти EEPROM. Часы используют внешний источник синхросигнала с частотой 32,768 кГц. Время отслеживается с использованием внутренних счетчиков часов, минут, секунд, дней, месяцев, лет, дней недели. Аппаратные прерывания (alarm) могут генерироваться по показаниям всех счетчиков вплоть до месяцев. Для использования и настройки MCP79411 поддерживает I2C со скоростью до 400 кГц.

Графический OLED-дисплей WEO012864DL
WEO012864DL – монохромный OLED-дисплей 128×64 точек с диагональю 0,96 дюйма, подключенный по интерфейсу I2C. Размеры области отображения 21,8×10,9 мм. Дисплей оборудован встроенным контроллером типа SSD1306BZ [10].

Ethernet
На стенде SDK-1.1M имеется разъем RJ-45 и приемопередатчик физического уровня Ethernet KSZ8081 [30]. Поддерживается передача данных по витой паре со скоростью 10/100 Мбит/с.

Излучатель звука HC0903A
HC0903A – электромагнитный излучатель звука (звукоизлучатель), управляемый прямоугольным периодическим сигналом.

Инерциальный измерительный модуль iNEMO LSM9DS
LSM9DS – это измерительный модуль, включающий трехмерные цифровые датчики линейного ускорения, угловой скорости и магнитного поля. Модуль подключается к процессорному модулю по последовательной шине I2C и имеет отдельные выходы прерываний.

Клавиатура
Клавиатура имеет 12 кнопок и организована в виде матрицы 3×4, подключенной к расширителю портов ввода-вывода PCA9538PW. Три бита порта соответствуют столбцам, четыре бита соответствуют рядам

2. Опишите устройство портов ввода-вывода общего назначения (GPIO) в микроконтроллере стенда SDK-1.1M. Перечислите основные режимы работы портов и их назначение.

- Контакты микроконтроллера внутри подключаются к портам ввода-вывода. Такие порты GPIO обычно объединяются в группы по 8, 16 или 32 порта с общими регистрами управления. Одни и те же порты могут выступать в роли входа или выхода в зависимости от настроек. По умолчанию обычно включен режим входа, чтобы исключить влияние порта на другие части схемы. Для работы в качестве выхода необходимо настроить соответствующий порт на выход при помощи управляющих регистров (описаны в документации микроконтроллера). Каждый контакт настраивается индивидуально – в одной группе портов входы и выходы могут чередоваться в любых комбинациях
- В стенде SDK-1.1M с процессорным модулем на базе STM32 на боковой панели имеется одна кнопка, подключѐнная к контакту PC15, и два управляемых светодиода: зеленый, подключенный к PD13, и двухцветный красный/желтый, подключенный к контактам PD14 и PD15
- Они могут работать в режимах Input, Output и Analog. Контакты GPIO могут выступать как в роли входа, так и в роли выхода

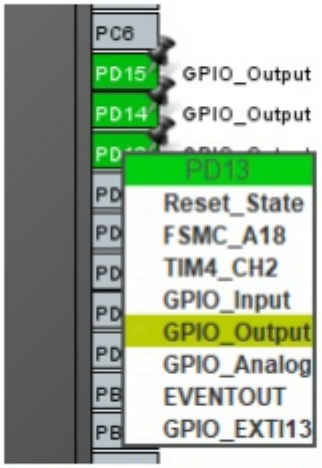


Рисунок 5 – Окно настройки портов GPIO

3. Опишите устройство и назначение таймера-счетчика на примере таймеров микроконтроллера стенда SDK-1.1M, основные режимы его работы. Как с помощью таймера можно организовать управление яркостью свечения светодиодов?

- Основным назначением таймера является увеличение или уменьшение значения в счетном регистре по каждому входному импульсу. Границы счета зависят от настроек. Таймеры имеют режим автоперезагрузки (autoreload), что позволяет не настраивать их после каждого окончания счета.
- **Устройство базового таймера.** Основа базового таймера - 16-битовый суммирующий счётчик и связанный с ним регистр автоматической перезагрузки.

Частота поступающего на вход счётчика сигнала делится с помощью предварительного делителя частоты таймера.

- Основные блоки таймера имеют соответствующие им регистры, отображаемые в память микроконтроллера (здесь и далее в статье в именах регистров используется обозначение x, x=6 или x=7 для соответствующих таймеров TIM6, TIM7):
    - **TIMx_CNT** (Counter Register) - регистр счётчика; содержит количество импульсов, подсчитанных счётчиком с момента последней реинициализации (переполнения);
    - **TIMx_PSC** (Prescaler Register) - регистр прескалера; определяет коэффициент деления k прескалера, k=TIMx_PSC+1;
    - **TIMx_ARR** (Auto-Reload Register) - регистр авто-перезагрузки; когда значение в счётчике достигает величины, записанной в этом регистре, следующий импульс сбрасывает счётчик в 0, при этом генерируется сигнал переполнения счётчика, который используется как update event - событие обновления. Если в TIMx_ARR записано значение 0, то счётчик таймера останавливается.
    - Содержимое счётчика, регистра автоматической перезагрузки и регистр предварительного делителя (прескалера) программно доступны для чтения и записи в любой момент, в том числе и в процессе счёта.
- Таймер может работать самостоятельно, а может и совместно с другими таймерами (в режиме главного или подчинённого устройства). Возможность совместной работы и множество выполняемых функций позволяют реализовать достаточно сложное устройство на одних лишь таймерах!
- Рассмотрим пример использования двух таймеров в программе для стенда SDK-1.1M. Один таймер будет задавать яркость зеленого светодиода путем подачи на него ШИМ-сигнала с необходимыми характеристиками, а второй таймер – отсчитывать периоды времени горения зеленого светодиода с заданной скважностью.
- 13-й сигнал порта GPIOD, к которому подключен зеленый светодиод, в качестве одной из альтернативных функций имеет 2-й канал таймера TIM4. Следовательно, мы можем использовать аппаратные возможности TIM4 для генерации ШИМ-сигнала. Таблица альтернативных функций приведена в datasheet микроконтроллера
- Чтобы настроить таймер TIM4, вначале определим, какова частота его внутреннего синхросигнала. По схеме в datasheet найдем, к какой шине подключен таймер. Это шина APB1. В конфигураторе микроконтроллера проверим установленную частоту синхросигнала для таймеров на этой шине. В нашем примере она составляет 90 МГц
- Теперь выберем для PD13 функцию TIM4_CH2 и настроим параметры таймера TIM4, как показано на рис. 14. Включено тактирование таймера от внутреннего источника, включена генерация ШИМ на канале 2, предделитель (prescaler) таймера будет делить входную частоту на 90, то есть счетный регистр будет тактироваться частотой 1 МГц. Таймер считает от нуля «вверх». Период счета (значение регистра автоперезагрузки) установлено в 999, чтобы регистр счета сбрасывался каждые 1000 тактов счетчика, то есть с периодом в 1 мс.
- Для ШИМ на канале 2 установлен режим 1 и длительность импульса в 500 тактов счетчика. В результате каждую миллисекунду будет генерироваться

импульс в 500 мкс, что соответствует коэффициенту заполнения ШИМ-сигнала в 50 % или скважности 2.

- Светодиод, на который подается ШИМ-сигнал с коэффициентом заполнения 50 %, будет гореть лишь половину времени, а из-за очень малого периода включения-выключения визуально будет казаться, что он равномерно горит с половиной максимальной яркости.
- Дополним программу, добавив использование второго таймера, с помощью которого реализуем автоматическое изменение яркости с некоторой периодичностью. Для этого не нужны каналы ввода-вывода сигналов, поэтому возьмем самый простой таймер TIM6. Он тоже тактируется от шины APB1. Настроим его для работы в режиме автоперезагрузки с генерацией прерываний по каждому переполнению. Период переполнения установим в 1.