

1) Создать новый проект для написания программного кода на языке Си

Блок, задающий время счета таймера, включает в себя:

- 16-битный суммирующий/вычитающий счетчик (TIM1_CNTRH + TIM1_CNTRL)
- 16-битный регистр автоперезагрузки (TIM1_ARRH + TIM1_ARRL)
- Счетчик повторений (TIM1_RCR)
- Предварительный делитель (TIM1_PSCRH + TIM1_PSCRL)

Таймер представляет из себя счетчик, который в зависимости от настройки увеличивает или уменьшает свое значение с каждым приходящим тактирующим импульсом. Тактовыми импульсами таймера являются импульсы тактирования микроконтроллера, проходящие через предварительный делитель. По умолчанию, после сброса микроконтроллеры семейства STM8S тактируются от внутреннего источника тактирования ($HSI/8 = 16 \text{ МГц}/8 = 2 \text{ МГц}$). Предварительный делитель таймера TIM1 настраивается с помощью 2 регистров TIM1_PSCRH и TIM1_PSCRL. Частота тактирования таймера TIM1 рассчитывается следующим образом: Частота тактирования микроконтроллера делится на (число, хранящееся в регистрах предварительного делителя + 1). Это означает, что таймер TIM1 может тактироваться импульсами с частотой тактирования микроконтроллера, деленной от 1 до 65536. После сброса, в регистрах предварительного делителя хранятся нули, соответственно частота тактирования микроконтроллера делится на 0+1 и частота тактирования таймера TIM1 составляет 2 МГц. **Обратите внимание, что для записи во все 16-ти битные регистры (TIM1_PSCR, TIM1_ARR и т.д.) необходимо сначала изменять старший байт, а затем младший!!! Это объясняется тем, что 16-ти битный регистр обновляется только после записи в младший регистр. В случае, когда изменяется сначала младший регистр, а потом старший, регистр обновится сразу после записи в младший, а изменение старшего проигнорируется.**

Направление счета таймера (суммирующий или вычитающий) настраивается с помощью бита DIR в регистре TIM1_CR1. По умолчанию там хранится логический «0», что означает, что счетчик суммирующий, с каждым тактирующим импульсом он будет увеличивать свое значение.

Порог счета таймера настраивается с помощью регистров автоперезагрузки TIM1_ARRH и TIM1_ARRL. Данные регистры содержат число, до которого будет считать таймер (в режиме суммирующего счетчика) либо число с которого он начнет счет (в режиме вычитающего счетчика). После сброса, в данных регистрах хранится число 65535 (0xFFFF).

Текущее число счета таймера TIM1 хранится в регистрах TIM1_CNTRH и TIM1_CNTRL. Включение таймера TIM1 осуществляется установкой в «1» бита CEN в регистре TIM1_CR1.

2) Настроить нулевую линию порта G в режиме двухтактного выхода с частотой тактирования 10 МГц. Включить таймер TIM1. В бесконечном цикле написать код программы, которая будет выводить сигнал логической «1» на нулевую линию порта G и ожидать, пока содержимое регистра TIM1_CNTRH меньше либо равно 0x80, затем, пока содержимое этого регистра больше чем 0x80 на нулевой линии порта G должен быть сигнал логического «0».

3) Выполнить компиляцию программы, сборку и ее загрузку в микроконтроллер. Запустить программу и с помощью осциллографа зарегистрировать сигнал на нулевой линии порта G. Посчитать частоту полученного сигнала, рассчитать теоретическую частоту этого сигнала, сравнить значения.

4) Изменить программный код таким образом, чтобы время импульса уменьшилось в 2 раза, а период импульсов не изменился. Выполнить компиляцию программы, сборку и ее загрузку в микроконтроллер. Запустить программу и с помощью осциллографа зарегистрировать сигнал на нулевой линии порта G.

5) Используя регистры предварительного делителя (TIM1_PSCRH и TIM1_PSCRL), увеличить период сигнала в 2 раза. Выполнить компиляцию программы, сборку и ее загрузку в

микроконтроллер. Запустить программу и с помощью осциллографа зарегистрировать сигнал на нулевой линии порта G. Как при этом изменилась длительность импульса и почему?

6) Используя регистры автоперезагрузки (TIM1_ARRH и TIM1_ARRL), уменьшить период сигнала в 2 раза. Выполнить компиляцию программы, сборку и ее загрузку в микроконтроллер. Запустить программу и с помощью осциллографа зарегистрировать сигнал на нулевой линии порта G. Как при этом изменилась длительность импульса и почему?

7) Используя флаг UIF в регистре TIM1_SR1, который устанавливается в «1» при переполнении таймера, написать программу, при выполнении которой, на нулевую линию порта G будут выводиться прямоугольные импульсы $t_i = t_p = 20$ мс. Выполнить компиляцию программы, сборку и ее загрузку в микроконтроллер. Запустить программу и с помощью осциллографа зарегистрировать сигнал на нулевой линии порта G.

Бит OPM в регистре TIM1_CR1 отвечает за настройку работы таймера в режиме одиночного импульса. Если данный бит сброшен в «0», то счетчик после переполнения продолжает счет заново. Если бит OPM установить в «1», то после переполнения счетчик остановится и начнет новый счет только после повторного запуска (установкой бита CEN в регистре TIM1_CR1).

8) Создать подпрограмму `delay_us (int a)`, которая будет осуществлять задержку в микросекундах с использованием таймера. Проверить работу данной подпрограммы формируя в бесконечном цикле прямоугольные импульсы с разными задержками:

1) $t_i = t_p = 20$ мкс; 2) $t_i = t_p = 500$ мкс; 3) $t_i = t_p = 2000$ мкс; 4) $t_i = t_p = 50000$ мкс; 5) $t_i = 500$ мкс, $t_p = 200$ мкс.

Результат показать преподавателю.

9) Создать подпрограмму `delay_ms (int a)`, которая будет осуществлять задержку в миллисекундах с использованием таймера. Проверить работу данной подпрограммы формируя в бесконечном цикле прямоугольные импульсы с разными задержками:

1) $t_i = t_p = 20$ мс; 2) $t_i = t_p = 500$ мс; 3) $t_i = 500$ мс, $t_p = 200$ мс; 4) $t_i = 10$ мс, $t_p = 200$ мкс

***Обратите внимание:** регистры TIM1_PSCRH и TIM1_PSCRL содержат значение, которое загружается в активный регистр предварительного делителя при каждом UEV (update event – событие обновления). Событие обновление формируется в следующих случаях:

При переполнении таймера;

Программно, при установке бита UG в регистре TIM1_EGR;

По событию запуска с контроллера тактирования/запуска.

Следовательно, по умолчанию, предварительный делитель не обновляется сразу после записи в регистры TIM1_PSCRH и TIM1_PSCRL, обновление происходит только после переполнения таймера. Для корректной работы необходимо, после изменения регистров предварительного делителя, программно сформировать событие обновления, установив логическую «1» в бите UG в регистре TIM1_EGR. **Следует отметить**, что по умолчанию, установка в «1» бита UG, сформирует событие обновления, что вызовет запрос на прерывание таймера (установится флаг UIF в регистре TIM1_SR1), для того чтобы этого избежать, необходимо установить в «1» бит URS в регистре TIM1_CR1.

Результат показать преподавателю.

10) Написать программу инверсии нулевого порта G по прерыванию таймера при переполнении. Для этого необходимо подключить в проект файл `main.h` в котором объявить

обработчик прерывания таймера, заменить в файле `stm8_interrupt_vector.c` имя соответствующего указателя в векторах прерываний и подключить заголовочный файл `main.h` внутри файлов `main.c` и `stm8_interrupt_vector.c` (как в 5 лабораторной работе). Разрешить прерывание таймера по переполнению в регистре `TIM1_IER` (бит `UIE`). Снять маскирование прерываний. (Обратите внимание, что в обработчике прерывания необходимо очистить флаг в регистре `TIM1_SR1`).

11) Написать программу плавного изменения яркости свечения светодиода с использованием прерываний таймера по переполнению. Светодиод должен сначала плавно увеличивать яркость свечения, затем плавно уменьшать. **Результат показать преподавателю.**

12) Убрать из программы разрешение прерывания таймера 1 по переполнению. Настроить первый канал таймера 1 в режиме ШИМ (В документации на контроллер (STM8S207 datasheet Table 6. Pin description стр. 28 необходимо найти, какому выводу соответствует альтернативная функция Timer 1 - channel 1). Для этого необходимо:

а) В регистре `TIM1_CCMR1` необходимо настроить канал захвата/сравнения 1 на режим выхода, для этого необходимо биты `CC1S` сбросить в «0», бит `OC1FE` также необходимо сбросить в «0», чтобы состояние выхода таймера зависело только от значения счетчика и от значения регистра захвата/сравнения 1 (`TIM1_CCR1H` и `TIM1_CCR1L`). Бит `OC1PE` влияет на то, когда будет обновляться регистры сравнения: если `OC1PE` = «0», то регистр сравнения обновляет свое значение, как только в регистры `TIM1_CCR1H` и `TIM1_CCR1L` будет записано новое значение, в случае, когда `OC1PE` = «1», регистр сравнения обновляет только после события обновления. Данный бит установите в «1». С помощью битов `OC1M` задайте для выхода сравнения 1 режим ШИМ1, установив в данные биты значение «110». В данном режиме, при режиме счета таймера вверх, на канале 1 активный уровень до тех пор, пока значение регистров `TIM1_CNTRH` и `TIM1_CNTRL` будет меньше значения регистров `TIM1_CCR1H` и `TIM1_CCR1L`.

б) В регистре `TIM1_CCER1` необходимо настроить полярность сигнала с помощью бита `CC1P`. Для выбора в качестве активного уровня – высокий уровень сигнала, необходимо в данный бит записать «0». Включение первого канала осуществляется установкой в «1» бита `CC1E`.

в) В регистре `TIM1_BKR` разрешить работу каналов таймера, установив в «1» бит `MOE`.

г) Задать с помощью регистра предварительного делителя частоту тактирования таймера 1 равную 1кГц

д) Задать с помощью регистра автоперезагрузки период счета таймера равный 10 мс.

е) Задать с помощью регистра захвата/сравнения 1 время импульса 5 мс

Проверить работу данной программы. **Обратите внимание**, что таймер, при счете вверх, считает от нуля до числа, которое загружено в регистры `TIM1_ARRH` и `TIM1_ARRL`, а при следующем увеличении сбрасывается в 0 и только при этом происходит событие переполнения. Таким образом, при необходимости, чтобы таймер переполнился на n-ый счет, необходимо в регистры `TIM1_ARRH` и `TIM1_ARRL` загружать число «n-1».

13) Написать программу, в которой изначально настраивается режим ШИМ1 первого канала таймера 1. На данном канале должны формироваться прямоугольные импульсы $t_i = 1\text{ мс}$, $T = 20\text{ мс}$. При этом частота тактирования таймера должна быть 100 кГц. Проверить работу данной программы.

14) В программе из пункта 13 необходимо добавить разрешение прерывания таймера по переполнению. В обработчике данного прерывания необходимо плавно увеличивать время импульса ШИМ до 2 мс, меняя каждый раз число, загружаемое в регистры `TIM1_CCR1H` и `TIM1_CCR1L` на 1. При достижении времени импульса 2 мс, необходимо его плавно уменьшать до 1 мс. Изменение импульса от 1 до 2 мс и обратно должно происходить циклично. Проверить работу программы с использованием

15) Подключить к выводу, выполняющему роль выхода первого таймера сервопривод и проверить его работу.

16) С помощью регистра счетчика повторений TIM1_RCR можно сделать так, чтобы таймер формировал запрос на прерывание не после каждого переполнения. В данный регистр записать число 1. Проверить работу описать изменение работы, по сравнению с пунктом 15.

17) Записать в регистр TIM1_RCR число 9, а изменение регистров TIM1_CCR1H и TIM1_CCR1L в прерывании таймера осуществлять на 10.