

Лабораторная работа №2

«Таймеры и обработка прерываний микроконтроллеров STM32»

1 Цель работы

Ознакомиться с программными средствами работы с таймерами и прерываниями микроконтроллеров STM32.

2 Теоретические сведения

2.1 Виды таймеров микроконтроллеров STM32

Встраиваемые устройства выполняют некоторые действия с учетом времени. Для достаточно простых и неточных задержек может использоваться цикл большой длительности, однако использование ядра микроконтроллера для выполнения зависимых от времени действий является плохим решением. По этой причине все микроконтроллеры предоставляют отдельную аппаратную периферию - таймеры. Таймеры являются не только генераторами временного отсчета, но и предоставляют набор дополнительных функций, используемых для взаимодействия между ядром и периферийными устройствами, как внутренними, так и внешними по отношению к микроконтроллеру.

Таймер – это автономный счетчик с частотой отсчета, составляющей часть его источника тактового сигнала. Частота отсчета может быть уменьшена с помощью отдельного делителя для каждого таймера.

В зависимости от семейства и используемого корпуса, микроконтроллеры STM32 реализуют различное количество таймеров, каждый из которых имеет определенные характеристики. В отличие от других периферийных устройств таймеры имеют практически одинаковую реализацию во всех сериях STM32 и сгруппированы в девять различных категорий. Наиболее важными из них являются:

– базовые таймеры - являются самым простым видом таймеров в микроконтроллерах STM32. Это 16-разрядные таймеры, используемые для генерации временного отсчета. Они не имеют выводов. Базовые таймеры также могут быть использованы в качестве ведущих для других таймеров.

– таймеры общего назначения - 16/32-разрядные таймеры (в зависимости от серии STM32), обеспечивающие сравнение выходного сигнала, одноимпульсного режима, захвата входного сигнала, интерфейса датчика. Может быть использован в качестве генератора временного отсчета, как и базовый таймер. Таймеры общего назначения предоставляют четыре программируемых входных/выходных канала.

– таймеры расширенного управления - являются наиболее полными в микроконтроллере STM32. В дополнение к функциям таймера общего назначения, они включают в себя несколько функций, относящихся к приложениям управления двигателем и цифрового преобразования энергии.

Таймеры общего назначения предназначены для управления продвинутыми возможностями, связанными со временем, такими как:

– режим захвата входного сигнала - позволяет вычислять частоту внешних сигналов, подаваемых на каждый из 4 каналов таймера, при этом захват выполняется независимо для каждого канала;

– режим сравнения выходного сигнала – позволяет управлять состоянием выходных каналов, когда регистр сравнения каналов (*TIMx_CCRx*) совпадает с регистром счетчика таймера (*TIMx_CNT*);

– режим генерации широтно-импульсной модуляции - позволяет формировать через каналы таймера импульсы с различными коэффициентами заполнения на заданной частоте;

– одноимпульсный режим - позволяет запускать счетчик в ответ на событие и генерировать импульс с программируемой длительностью после программируемой задержки.

Основные регистры таймеров общего назначения:

- *TIMx_CR1* (смещение 0x00) - регистр управления, задает режим работы таймера;
- *TIMx_CR2* (смещение 0x04) - регистр управления, задает режимы ведущего таймера и взаимодействия с DMA;
- *TIMx_SMCR* (смещение 0x08) - регистр управления ведомым режимом таймера;
- *TIMx_DIER* (смещение 0x0C) - регистр разрешения прерываний и событий DMA;
- *TIMx_SR* (смещение 0x10) - регистр состояния, содержит флаги прерываний;
- *TIMx_EGR* (смещение 0x14) - регистр генерации событий, позволяет искусственно формировать события;
- *TIMx_CCMR1* (смещение 0x18), *TIMx_CCMR2* (смещение 0x1C) - регистры режимов захвата или сравнения сигналов;
- *TIMx_CCER* (смещение 0x20) - регистр разрешения захвата или сравнения сигналов;
- *TIMx_CNT* (смещение 0x24) - регистр счетчика, прямое обращение к регистру счетчика необходимо только в случае отслеживания времени без использования прерывания, циклическими проверками значения таймера;
- *TIMx_PSC* (смещение 0x28) - регистр предделителя;
- *TIMx_ARR* (смещение 0x2C) - регистр перезагрузки;
- *TIMx_CCR1* (смещение 0x34), *TIMx_CCR2* (смещение 0x38), *TIMx_CCR3* (смещение 0x3C), *TIMx_CCR4* (смещение 0x40), - регистр значения захвата или сравнения сигнала для соответствующего канала таймера;
- *TIMx_DCR* (смещение 0x48) - регистр управления работой таймера с контроллером DMA;

– *TIMx_DMAR* (смещение 0x4C) - регистр адреса для последовательного режима DMA.

Структура регистра управления *TIMx_CR1* приведена на рис. 1.

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

Рис.1 - Структура регистра управления таймера *TIMx_CR1*

Флаги регистра имеют следующие значения:

- *CKD* - задает значение делителя частоты таймера;
- *ARPE* - определяет режим записи в регистр перезагрузки;
- *CMS* - определяет режим для двунаправленного счета или разрешают счет только в одном направлении;
- *DIR* - задает направление счета: прямой или реверсивный;
- *OPM* - задает режим одиночного импульса;
- *URS* - задает формирование события перезагрузки таймера программно или от ведомого таймера;
- *UDIS* - управляет разрешением автоматической перезагрузки счетчика;
- *CEN* - разрешает работу счетчика.

При работе с каналами таймеров общего назначения, необходимо выполнить настройку портов ввода-вывода, закрепленных, согласно технической спецификации микроконтроллера, за выбранным каналом таймера. Для микроконтроллеров семейства STM32F1, порты каналов таймеров используемые как входные должны быть сконфигурированы как входные - *GPIO_Mode_IPD*, порты каналов, используемые как выходные, должны быть определены как двухтактные выходы в режиме альтернативной функции - *GPIO_Mode_AF_PP*.

Номера портов ввода-вывода, закрепленных за каналами таймеров TIM2-TIM4, приведены в табл.1.

Табл.1

Порты ввода-вывода каналов таймеров

| Номер таймера | Номер канала | Номер вывода |
|---------------|--------------|--------------|
| TIM2 | CH1 | PA0 |
| | CH2 | PA1 |
| | CH3 | PA2 |
| | CH4 | PA3 |
| TIM3 | CH1 | PA6 |
| | CH2 | PA7 |
| | CH3 | PB0 |
| | CH4 | PB1 |
| TIM4 | CH1 | PB6 |
| | CH2 | PB7 |
| | CH3 | PB8 |
| | CH4 | PB9 |

2.2 Обработка прерываний в микроконтроллерах STM32

Все микроконтроллеры предоставляют функцию, называемую прерываниями. Прерывание – это асинхронное событие, которое вызывает остановку выполнения текущего кода в приоритетном порядке.

Архитектура ARM различает два типа исключений: прерывания, вызываемые аппаратным обеспечением, и исключения, вызываемые программным обеспечением. В терминологии ARM прерывание – это тип исключения. Процессоры Cortex-M предоставляют модуль, предназначенный для управления исключениями - контроллер вложенных векторных прерываний (NVIC). Контроллер NVIC – это специальный аппаратный

модуль внутри микроконтроллеров на базе Cortex-M, отвечающий за обработку исключений.

Для работы с аппаратными прерываниями необходимо:

- разрешить глобальные прерывания микроконтроллера - реализуется функцией `__enable_irq()`;
- разрешить прерывание в контроллере прерываний - реализуется функцией `NVIC_EnableIRQ`, которая в качестве параметра принимает номер прерывания - для номеров прерываний определено перечисление `IRQn` в файле `stm32f10x.h`;
- настроить устройство на формирование прерывания;
- создать функцию обработки соответствующего прерывания - объявления функций обработчиков прерываний находятся в файле `startup_stm32f10x_md.s` в секции `Default_Handler`.

2.3 Базовая настройка таймера

Для обеспечения работы таймера необходимо выполнить включение тактового сигнала модуля с использованием функции `RCC_APB1PeriphClockCmd`.

Для таймера необходимо выполнить базовую настройку: определяется структура базовой инициализации таймера - `TIM_TimeBaseInitTypeDef`, выполняется начальная инициализация данной структуры - функцией `TIM_TimeBaseStructInit`, после чего производится настройка параметров таймера (значений `TIM_Prescaler`, `TIM_Period`) и выполняется базовая инициализация таймера функцией `TIM_TimeBaseInit`.

Предделитель таймера (Prescaler) задает значение, используемое для деления тактового сигнала таймера на коэффициент в диапазоне от 1 до 65535. Регистр предделителя имеет 16-разрядное разрешение. Например, если шина, к которой подключен таймер, работает на частоте 72 МГц, то значение предделителя, равное 72, понижает частоту отсчета до 1 МГц.

Значение предделителя для таймера выбирается с учетом частоты тактирования. Для получения значения предделителя при известной частоте и требуемом значении отсчета следует воспользоваться следующей формулой:

$$Prescaler = F_{TIM} \times T_{CNT} \quad (1)$$

где:

- F_{TIM} - частота тактового сигнала, Гц;
- T_{CNT} - время одного отсчета таймера, с.

Значение предделителя, записываемое в регистр таймера, должно быть на 1 меньше рассчитанного.

Для определения частоты тактового сигнала таймера при моделировании можно воспользоваться отображением параметров тактирования через модуль управления питанием, сигналами сброса и тактирования (*Power, Reset and Clock Control - PRCC*). Для доступа к данному модулю выберите пункт меню *Peripherals*, далее пункт *Power, Reset and Clock Control*. В открывшемся окне, частота тактирующего сигнала таймеров описывается параметром *TIMXCLK*.

Структуры данных и функции для работы с таймерами для микроконтроллеров семейства STM32F1, предоставляемые библиотекой SPL, представлены в файлах *stm32f10x_tim.h* и *stm32f10x_tim.c*.

2.4 Работа таймера в режиме захвата сигнала

Для задания параметров таймера в режиме захвата сигнала используется структура *TIM_ICInitTypeDef*. Поля структуры определяют:

- *TIM_Channel* - используемый канал таймера;
- *TIM_ICPolarity* - активный фронт входного сигнала;
- *TIM_ICSelection* - отображение входных каналов на каналы таймера;

– *TIM_ICPrescaler* - предделитель заданного входного канала - задает количество событий между срабатываниями захвата;

– *TIM_ICFilter* - частота отбора входного сигнала для защиты от дребезга.

Начальное заполнение структуры выполняется функцией *TIM_ICStructInit*. После инициализации структуры выполняется ее настройка. После чего выполняется инициализация режима захвата сигнала таймера с использованием функции *TIM_ICInit*, которая в качестве параметров принимает указатель на структуру таймера *TIM_TypeDef* и указатель на структуру режима *TIM_ICInitTypeDef*.

Порт ввода-вывода, закрепленный за каналом таймера, должен быть проинициализирован в режиме входа.

Для включения прерывания таймера по получении фронта сигнала необходимо определить в качестве источника прерывания *TIM_IT_CCx*, где вместо *x* указывается номер используемого канала таймера. При возникновении прерывания, в регистр *CCRx* (*x* - номер используемого канала) записывается значение регистра *CNT* в момент возникновения прерывания.

Для получения значения периода сигнала необходимо зафиксировать моменты прихода двух передних фронтов сигнала и рассчитать разницу между полученными значениями. Для повышения точности измерения можно обнулять счетный регистр перед выделением очередного периода.

По завершении обработки прерывания необходимо снять бит необработанного прерывания функцией *TIM_ClearITPendingBit*.

2.5 Работа таймера в режиме широтно-импульсной модуляции

Широтно-импульсная модуляция (ШИМ, Pulse-width modulation, PWM) – метод, используемый для генерации нескольких импульсов с различными коэффициентами заполнения в заданный период времени или на заданной частоте.

Основное применение ШИМ:

- управление выходным напряжением (током);
- кодирование (модулирование) сообщения на несущем сигнале.

Основными характеристиками сигналов широтно-импульсной модуляции являются период сигнала, длительность импульса и коэффициент заполнения.

Длительность импульса – это продолжительность времени внутри одного периода, когда сигнал имеет активный уровень.

Коэффициент заполнения – это процент от периода сигнала, в течение которого сигнал имеет активный уровень.

Для нахождения коэффициента заполнения используется формула:

$$D = \frac{W}{T} \times 100\% \quad (2)$$

где:

- D - коэффициент заполнения;
- W - длительность сигнала;
- T - период сигнала.

Для задания параметров таймера в режиме генерации широтно-импульсной модуляции используется структура `TIM_OCInitTypeDef`. Для режима широтно-импульсной модуляции необходимо определить следующие поля структуры:

- *`TIM_OCMode`* - режим работы: сравнение выходного сигнала или широтно-импульсная модуляция;
- *`TIM_OutputState`* - состояние для выхода - должно быть определено *`TIM_OutputState_Enable`*;
- *`TIM_Pulse`* - значение, загружаемое в регистр *`CCRx`* - для режима широтно-импульсной модуляции определяет длительность импульса.

Начальное заполнение структуры выполняется функцией *TIM_OCStructInit*. После инициализации структуры выполняется ее настройка. После чего выполняется инициализация режима с использованием функции *TIM_OC1Init*, которая в качестве параметров принимает указатель на структуру таймера *TIM_TypeDef* и указатель на структуру режима *TIM_OCInitTypeDef*.

Порт ввода-вывода, закрепленный за соответствующим каналом таймера, должен быть проинициализирован в режиме выхода альтернативной функции.

Для включения прерывания таймера по получении фронта сигнала необходимо определить в качестве источника прерывания *TIM_IT_Update*. При возникновении прерывания, для изменения длительности сигнала широтно-импульсной модуляции необходимо записать в регистр *CCRx* (x - номер используемого канала) новое значение длительности сигнала.

По завершении обработки прерывания необходимо снять бит необработанного прерывания функцией *TIM_ClearITPendingBit*.

3 Задание

Разработать программу для микроконтроллера STM32F103RB, реализующую захват входного сигнала, настройку выходного сигнала согласно параметрам входного: длительность сигнала широтно-импульсной модуляции равна периоду входного сигнала, выдачу выходного сигнала широтно-импульсной модуляции с постоянной частотой.

Выполнить формирование входного сигнала через файл сценария с сигнальной функцией. Сигнальная функция должна обеспечивать изменение периода входного сигнала для значений: 2 мс, 4 мс, 6 мс, 12 мс, 20 мс. Смену периода выполнять через фиксированное количество импульсов - 20 и более.

Период таймера широтно-импульсной модуляции задать равным 32 мс.

Параметры программы выбираются согласно варианту и приведены в табл. 2.

Варианты заданий к лабораторной работе №2

| Номер варианта | Таймер захвата сигнала | | | Таймер ШИМ | | |
|-------------------|------------------------|-----------------|---------------------|------------------|-----------------|---------------------|
| | Номер таймера | Номер канала | Значение отсчета | Номер таймера | Номер канала | Значение отсчета |
| 1 | TIM4 | CH2 | 1 мкс | TIM3 | CH1 | 10 мкс |
| 2 | TIM4 | CH4 | 50 мкс | TIM3 | CH3 | 100 мкс |
| 3 | TIM4 | CH3 | 5 мкс | TIM3 | CH1 | 20 мкс |
| 4 | TIM4 | CH1 | 10 мкс | TIM3 | CH3 | 50 мкс |
| 5 | TIM3 | CH4 | 1 мкс | TIM2 | CH1 | 10 мкс |
| 6 | TIM3 | CH3 | 10 мкс | TIM2 | CH2 | 100 мкс |
| 7 | TIM3 | CH2 | 1 мкс | TIM2 | CH3 | 10 мкс |
| 8 | TIM3 | CH1 | 5 мкс | TIM2 | CH4 | 100 мкс |
| 9 | TIM4 | CH1 | 10 мкс | TIM2 | CH1 | 100 мкс |
| 10 | TIM4 | CH2 | 5 мкс | TIM2 | CH2 | 10 мкс |
| 11 | TIM4 | CH3 | 1 мкс | TIM2 | CH1 | 2 мкс |
| 12 | TIM4 | CH4 | 20 мкс | TIM2 | CH2 | 50 мкс |
| 13 | TIM3 | CH1 | 2 мкс | TIM4 | CH1 | 10 мкс |
| 14 | TIM3 | CH2 | 20 мкс | TIM4 | CH2 | 100 мкс |
| 15 | TIM3 | CH2 | 10 мкс | TIM4 | CH3 | 100 мкс |
| 16 | TIM3 | CH1 | 1 мкс | TIM4 | CH4 | 5 мкс |
| 17 | TIM2 | CH1 | 1 мкс | TIM3 | CH1 | 10 мкс |
| 18 | TIM2 | CH2 | 1 мкс | TIM3 | CH3 | 2 мкс |
| 19 | TIM2 | CH3 | 50 мкс | TIM3 | CH1 | 100 мкс |
| 20 | TIM2 | CH4 | 10 мкс | TIM3 | CH3 | 20 мкс |

4 Порядок выполнения работы

4.1 Получить вариант задания у преподавателя.

4.2 Рассчитать значение предделителя каждого таймера.

4.3 Создать проект в среде *Keil uVision5* для микроконтроллера *STM32F103RB*.

4.4 Выбрать программные компоненты:

- *CMSIS/Core*;
- *Device/Startup*;
- *Device/StdPeriph Drivers/Framework*;
- *Device/StdPeriph Drivers/GPIO*;
- *Device/StdPeriph Drivers/RCC*;
- *Device/StdPeriph Drivers/TIM*.

4.5 Выполнить настройку режима отладки для проекта.

4.6 Разработать файл сценария.

4.7 Разработать программу согласно варианту задания, реализующую:

- инициализацию портов ввода-вывода;
- инициализацию таймера в режиме захвата сигнала;
- инициализацию таймера в режиме широтно-импульсной модуляции;
- обработку прерываний таймеров.

4.8 Выполнить симуляцию разработанной программы с использованием функций отладки. Зафиксировать параметры обрабатываемых и формируемых сигналов.

4.9 Рассчитать коэффициент заполнения по полученным сигналам широтно-импульсной модуляции.

4.10 Сделать выводы по проделанной работе и оформить отчет.

5 Содержание отчета

Отчет по результатам работы должен включать:

- используемые периферийные модули микроконтроллера;
- словесное описание алгоритма работы разработанной программы;
- расчет значения предделителя каждого таймера;
- текст файла сценария имитации входного сигнала;

- текст разработанной программы;
- результаты работы разработанной программы в режиме отладки в симуляторе с временными характеристиками сигналов;
- расчет коэффициентов заполнения для полученных сигналов широтно-импульсной модуляции.

6 Контрольные вопросы

1. Основные виды таймеров в микроконтроллерах STM32?
2. Назначение предделителя таймера?
3. Режимы работы таймеров общего назначения?
4. Назначение и принцип работы режима захвата сигнала таймера?
5. Назначение и принцип работы режима широтно-импульсной модуляции таймера?
6. Как определить коэффициент заполнения для сигнала широтно-импульсной модуляции?