

Разделы

Новости (/index.php/novosti.html)

Встраиваемые системы (/index.php/embedded-programming.html)

Программирование AVR (/index.php/programming-avr.html)

Программирование ARM (/index.php/programmirovanie-arm.html)

Инструменты/технологии (/index.php/instruments-technologies.html)

Как подключить (/index.php/how-connection.html)

Компоненты (/index.php/electronic-components.html)

RTOS (/index.php/rtos.html)

Софт (/index.php/iar-embedded-workbench.html)

Проекты (/index.php/projects-avr.html)

Ссылки (/index.php/links.html)

1wire (/index.php/programming-avr/itemlist/tag/1wire.html) arm (/index.php/programming-avr/itemlist/tag/arm.html)

avr/klindex/php/prpgramming-

avr программатор (/index.php/programming-avr/itemlist/tag/avr программатор.html) ds18b20 (/index.php/programming-avr/itemlist/tag/ds18b20.html)

eeprom (/index.php/programming-avr/itemlist/tag/eeprom.html) i2c (/index.php/programming-avr/itemlist/tag/i2c.html)

IAR (/index.php/programming-avr/itemlist/tag/IAR.html) lcd (/index.php/programming-avr/itemlist/tag/lcd.html)

tsop (/index.php/programming-avr/itemlist/tag/tsop.html) twi (/index.php/programming-avr/itemlist/tag/twi.html)

алгоритмы (/index.php/programming-avr/itemlist/tag/алгоритмы.html)

библиотеки (/index.php/programming-avr/itemlist/tag/библиотеки.html) датчик (/index.php/programming-avr/itemlist/tag/датчик.html)

драйвер (/index.php/programming-avr/itemlist/tag/драйвер.html) интерфейс (/index.php/programming-avr/itemlist/tag/днайвер.html)

 ${\it komnohehtbi (/index.php/programming-avr/itemlist/tag/komnohehtbi.html)} \quad {\it makpocbi (/index.php/programming-avr/ite$

oy (/index.php/programming-avr/itemlist/tag/oy.html)

программирование микроконтроллеров (/index.php/programming-avr/itemlist/tag/программирование микроконтроллеров.html)

pacчет (/index.php/programming-avr/itemlist/tag/pacчет.html)

семисегментный индикатор (/index.php/programming-avr/itemlist/tag/семисегментный индикатор.html) си (/index.php/programming-avr/itemlist/tag/си.html) событийная система (/index.php/programming-avr/itemlist/tag/событийная система.html)

схемотехника (/index.php/programming-avr/itemlist/tag/схемотехника.html) таймер (/index.php/programming-avr/itemlist/tag/таймер.html) управление (/index.php/programming-avr/itemlist/tag/управление.html) устройства (/index.php/programming-avr/itemlist/tag/устройства.html)

учебный курс avr (/index.php/programming-avr/itemlist/tag/учебный курс avr.html)

шим (/index.php/programming-avr/itemlist/tag/шим.html)

Учебный курс AVR. Таймер - счетчик Т0. Регистры. Ч1



14/08/2013 - 18:25 Павел Бобков

Введение

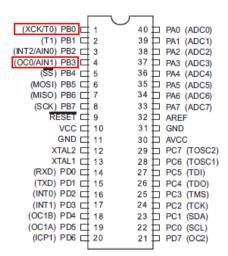
Таймер-счетчик является одним из самых ходовых ресурсов AVR микроконтроллера. Его основное назначение - отсчитывать заданные временные интервалы. Кроме того, таймеры-счетчики могут выполнять ряд дополнительных функций, как то - формирование ШИМ сигналов, подсчет длительности и количества входящих импульсов. Для этого существуют специальные режимы работы таймера-счетчика.

В зависимости от модели микроконтроллера количество таймеров и набор их функций может отличаться. Например, у микроконтроллера Atmega16 три таймера-счетчика - два 8-ми разрядных таймера-счетчика Т0 и Т2, и один 16-ти разрядный - Т1. В этой статье, на примере ATmega16, мы разберем как использовать таймер-счетчик Т0.

Используемые выводы

Таймер-счетчик T0 использует два вывода микроконтроллера ATmega16. Вывод T0 (PB0) - это вход внешнего тактового сигнала. Он может применяться, например, для подсчета импульсов. Вывод ОС0 (PB3) - это выход схемы сравнения таймера-счетчика. На этом выводе с помощью таймера может формировать меандр или ШИМ сигнал. Также он может просто менять свое состояние при срабатывании схемы сравнения, но об этом

поговорим позже.



Выводы Т0 и ОС0 задействуются только при соответствующих настройках таймера, в обычном состоянии это выводы общего назначения.

Регистры таймера-счетчика Т0

Хоть это и скучно, но регистры - это то, без чего невозможно программировать микроконтроллеры, конечно, если вы не сидите плотно на Arduino. Так вот, таймер Т0 имеет в своем составе три регистра:

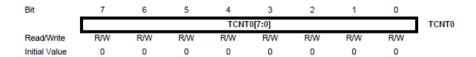
- счетный регистр TCNT0,
- регистр сравнения OCR0,
- конфигурационный регистр TCCR0.

Кроме того, есть еще три регистра, относящиеся ко всем трем таймерам ATmega16:

- конфигурационный регистр TIMSK,
- статусный регистр TIFR.
- регистр специальных функций SFIOR

Начнем с самого простого.

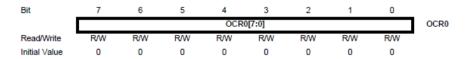
TCNT0



Это 8-ми разрядный счетный регистр. Когда таймер работает, по каждому импульсу тактового сигнала значение TCNT0 изменяется на единицу. В зависимости от режима работы таймера, счетный регистр может или увеличиваться, или уменьшаться.

Регистр TCNT0 можно как читать, так и записывать. Последнее используется когда требуется задать его начальное значение. Когда таймер работает, изменять его содержимое TCNT0 не рекомендуется, так как это блокирует схему сравнения на один такт.

OCR0



Это 8-ми разрядный регистр сравнения. Его значение постоянно сравнивается со счетным регистром TCNT0, и в случае совпадения таймер может выполнять какие-то действия - вызывать прерывание, менять состояние вывода ОС0 и т.д. в зависимости от режима работы.

Значение OCR0 можно как читать, так и записывать.

TCCR0 (Timer/Counter Control Register)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | _ |
|---------------|------|-------|-------|-------|-------|------|------|------|-------|
| | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 | TCCR0 |
| Read/Write | W | R/W | RW | R/W | RW | RW | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Это конфигурационный регистр таймера-счетчика T0, он определяет источник тактирования таймера, коэффициент предделителя, режим работы таймера-счетчика T0 и поведение вывода ОСО. По сути, самый важный регистр.

Биты **CS02**, **CS01**, **CS00** (**Clock Select**) - определяют источник тактовой частоты для таймера T0 и задают коэффициент предделителя. Все возможные состояния описаны в таблице ниже.

| CS02 | CS01 | CS00 | Описание | | |
|------|------|------|----------------------------------|--|--|
| | | 0 | Источника тактирования нет. | | |
| U | U | U | Таймер остановлен. | | |
| 0 | 0 | 1 | Тактовая частота МК | | |
| 0 | 1 | 0 | Тактовая частота МК/8 | | |
| 0 | 1 | 1 | Тактовая частота МК/64 | | |
| 1 | 0 | 0 | Тактовая частота МК/256 | | |
| 1 | 0 | 1 | Тактовая частота МК/1024 | | |
| 4 | 4 | _ | Внешний источник на выводе ТО. | | |
| 1 - | 1 1 | | Срабатывание по заднему фронту | | |
| | nial | 60 | Внешний источник на выводе ТО. | | |
| | 1 | 1 | Срабатывание по переднему фронту | | |

Как видите, таймер-счетчик может быть остановлен, может тактироваться от внутренней частоты и также может тактироваться от сигнала на выводе Т0.

Биты WGM10, WGM00 (Wave Generator Mode) - определяют режим работы таймера-счетчика Т0. Всего их может быть четыре - нормальный режим (normal), сброс таймера при совпадении (СТС), и два режима широтно-импульсной модуляции (FastPWM и Phase Correct PWM). Все возможные значения описаны в таблице ниже.

| VA/CN401 | WGM00 | Режим работы | | | |
|----------|---------|--------------------|--|--|--|
| WGM01 | WGIVIOU | таймера/счетчика | | | |
| 0 | 0 | Normal | | | |
| 0 | 1 | PWM, Phase Correct | | | |
| 1 | 0 | CTC | | | |
| 1 | 1 | Fast PWM | | | |

Более подробно будем разбирать режимы в коде. Сейчас все нюансы все равно не запомнятся.

Биты **COM01**, **COM00** (**Compare Match Output Mode**) - определяют поведение вывода ОС0. Если хоть один из этих битов установлен в 1, то вывод ОС0 перестает функционировать как обычный вывод общего назначения и подключается к схеме сравнения таймера счетчика Т0. Однако при этом он должен быть еще настроен как выход.

Поведение вывода ОС0 зависит от режима работы таймера-счетчика Т0. В режимах normal и СТС вывод ОС0 ведет себя одинаково, а вот в режимах широтно-импульсной модуляции его поведение отличается. Не будем сейчас забивать себе голову всеми этими вариантами и разбирать таблицы для каждого режима, оставим это на практическую часть.

И последний бит регистра TCCR0 - это бит **FOC0 (Force Output Compare)**. Этот бит предназначен для принудительного изменения состояния вывода ОС0. Он работает только для режимов Normal и СТС. При установки бита FOC0 в единицу состояние вывода меняется соответственно значениям битов COM01, COM00. FOC0 бит не вызывает прерывания и не сбрасывает таймер в СТС режиме.

TIMSK (Timer/Counter Interrupt Mask Register)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | _ |
|---------------|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | TIMSK |
| Read/Write | RW | R/W | R/W | R/W | R/W | R/W | R/W | R/W | • |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Общий регистр для всех трех таймеров ATmega16, он содержит флаги разрешения прерываний. Таймер Т0 может вызывать прерывания при переполнении счетного регистра TCNT0 и при совпадении счетного регистра с регистром сравнения ОСR0. Соответственно для таймера Т0 в регистре TIMSK зарезервированы два бита - это TOIE0 и OCIE0. Остальные биты относятся к другим таймерам.

ТОІЕО - 0-е значение бита запрещает прерывание по событию переполнение, 1 - разрешает.

ОСІЕО - 0-е значение запрещает прерывания по событию совпадение, а 1 разрешает.

Естественно прерывания будут вызываться, только если установлен бит глобального разрешения прерываний - бит I регистра SREG.

TIFR (Timer/Counter0 Interrupt Flag Register)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|------|------|------|-------|-------|------|------|------|------|
| | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | TIFR |
| Read/Write | R/W | RW | R/W | R/W | R/W | RW | RW | R/W | • |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Общий для всех трех таймеров-счетчиков регистр. Содержит статусные флаги, которые устанавливаются при возникновении событий. Для таймера Т0 - это переполнение счетного регистра TCNT0 и совпадение счетного регистра с регистром сравнения OCR0.

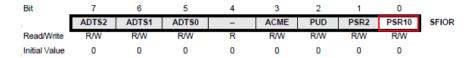
Если в эти моменты в регистре TIMSK разрешены прерывания и установлен бит I, то микроконтроллер вызовет соответствующий обработчик.

Флаги автоматически очищаются при запуске обработчика прерывания. Также это можно сделать программно, записав 1 в соответствующий флаг.

TOV0 - устанавливается в 1 при переполнении счетного регистра.

ОСF0 - устанавливается в 1 при совпадении счетного регистра с регистром сравнения

SFIOR (Special Function IO Register)



Начинающему про этот регистр в принципе можно и не знать, один из его разрядов сбросывает 10-ти разрядный двоичный счетчик, который делит входную частоту для таймера Т0 и таймера Т1.

Сброс осуществляется при установке бита PSR10 (Prescaler Reset Timer/Counter1 и Timer/Counter0) в единицу.

Заключение

Нудная часть закончена. Далее разберем как настроить таймер на определенную частоту, как таймер ведет себя в разных режимах, как генерировать ШИМ сигнал.

Твитнуть





Tagged under #avr (/index.php/programming-avr/itemlist/tag/avr.html) #учебный курс avr (/index.php/programming-avr/itemlist/tag/ учебный%20курс%20avr.html) #таймер (/index.php/programming-avr/itemlist/tag/таймер.html)

Related items

- Библиотека для опроса кнопок (/index.php/programming-avr/item/218-biblioteka-dlya-oprosa-knopok.html)
- Работа с SD картой. Воспроизведение wav файла. Ч3 (/index.php/programming-avr/item/212-rabota-s-sd-kartoy-vosproizvedenie-wav-fayla-ch3.html)
- Работа с SD картой. Подключение к микроконтроллеру. Ч1 (/index.php/programming-avr/item/209-rabota-s-sd-kartoy-podklyuchenie-k-mikrokontrolleruch1.html)
- AVR315: Использование TWI модуля в качестве ведущего I2C устройства (/index.php/programming-avr/item/208-avr315-ispolzovanie-twi-modulya-v-kachestve-veduschego-i2c-ustroystva.html)
- Учебный курс AVR. Использования TWI модуля как ведущего I2C устройства. Работа на прерываниях. Ч5 (/index.php/programming-avr/item/201-uchebnyy-kurs-avr-ispolzovaniya-twi-modulya-kak-veduschego-i2c-ustroystva-rabota-na-preryvaniyah-ch5.html)

Комментарии (/index.php/component/jcomments/feed/com_k2/171.html)

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-3082) **foxit** 14.08.2013 20:24 Ждем продолжение!

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-3091) **Pashgan** 16.08.2013 18:54 Ученые быотся над этим.

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4291) **D.mas** 21.09 2014 11:04
Здравствуйте, спецы :-) Я не волшебник, а только учусь, пытаюсь самостоятельно запилить хронограф для пневматики. Видел в интернете и схемы и решения, но хочется свое. С датчиками еще не определен, разбираюсь с по. Поэтому хотел бы у вас спросить.. Какое количество тактов уходит на обработку прерывания по совпадению с ОСR0? Понимаю, что зависит от длины кода в обработчике. И вообще, если установить CLK 8МГц,prescaler=1; OCR0=0x01; и в теле прерывания, допустим, глобальная переменная і будет і++, будет ли исправно отсчитываться по 1/8К секунд?

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4292) **D.mas** 21.09.2014 11:06 сорри, по 1/8М сек

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4353) **alex6441161** 02.12.2014 14:40 подскажите а обязательно значение в регистр OCR1A записывать в 16ричном формате? и обязательно ли записывать значение сначала в старший а потом младший, т.е. вот так:

OCR1AH=0x03;//записываем в регистр OCR1A 1000 OCR1AL=0xE8; или можно просто

OCR1A=0x03E8;

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4354) **_Артём_** 03.12.2014 08:09 Цитирую alex6441161:

подскажите а обязательно значение в регистр OCR1A записывать в 16ричном формате?

Нет, не обязательно. Можете хоть в десятичном, хоть в символьном.

```
Цитирую аlex6441161:

и обязательно ли записывать значение сначала в старший а потом младший, т.е. вот так:

ОСR1AH=0x03;//записываем в регистр ОСR1A 1000

ОСR1AL=0xE8;

У АВР шина 8-битная, поэтому сначала надо писать в Н, потом в L. При чтении наоборот.

Цитирую alex6441161:

ОСR1A=0x03E8;

Так тоже можно, если компилятор поддерживает (IAR и GCC позволяют так обращатся к регистру, насчёт других не знаю).

Ответить | Ответить с цитатой | Цитировать
```

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4357) **Zaprom** 04.12.2014 20:05 Здравствуйте!! По-моему есть ошибка в строке " Биты WGM10, WGM00 (Wave Generator Mode) - определяют режим работы таймера-счетчика Т0." Там должен быть бит WGM01.

Ответить | Ответить с цитатой | Цитировать

```
# (/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4400) Ваха 23.01.2015 13:13 
Цитата:
```

Биты WGM10, WGM00 (Wave Generator Mode) - определяют режим работы таймера-счетчика Т0.

думаю тут ошибка, наверное подразумевалось **WGM01** судя по таблице которая приводится следом

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4968) **goodspeedmen** 09.12.2016 18:24 Код:

```
ISR(TIMER2_OVF_vect) //программа обработки прерывания по переполнению таймера {
    PORTB = (0);
    TIMSK0 = (0b00000000);
}

int main(void) //основная программа {
    TCCR0A = (0b00000000);
    TCCR0B |= (0b10000001);
    DDRD |= (0b10000000);
    Sei(); //разрешаем глобальные прерывания

while(1) {
    IF(PIND.0 == 1) {
        TIMSK0 = (0b00000001);
        TCNT0 = (0b00000001);
        TCNT0 = (0b000000001);
        PORTB.0 = 1;
    }
}
```

Ответить | Ответить с цитатой | Цитировать

(/index.php/programming-avr/item/171-avr-timer-t0-ch1.html#comment-4969) **goodspeedmen** 09.12.2016 18:28 Цитирую goodspeedmen:

```
Код:

ISR(TIMER2_OVF_vect) //программа обработки прерывания по переполнению таймера
{
    PORTB = (0);
    TIMSK0 = (0b00000000);
}

int main(void) //ocновная программа
{
    TCCR0A = (0b00000000);
    TCCR0B |= (0b10000001);
    DDRD |= (0b10000000);
    DDRD |= (0b10000000);
    sei(); //разрешаем глобальные прерывания

while(1)
    {
    IF(PIND.0 == 1)
    {
        IIMSK0 = (0b00000000);
        PORTB.0 = 1;
    }
    }
}
```

Можно ли использовать нечто такое?

Ответить | Ответить с цитатой | Цитировать

| Обновить список ко | мментариев | | |
|--------------------|--------------------------|---|--------------------------------|
| RSS лента коммент | ариев этой записи (/inde | x.php/component/jcomments/feed/com_k2/171.html) | |
| авить комментари | Й | | |
| | Имя (обязательн | noe) | |
| | E-Mail | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| £0~ | | | |
| eggs | | | |
| Обновить | | | |
| | | | |
| Отправить | | | |
| равить | | | |
| | | | JComments (http://www.joomlatu |

© 2017 chipenable Haверх