

Done at Home

Программирование микроконтроллеров avr

AVR микроконтроллеры для начинающих (урок 6) таймеры-счетчики

admin | 07.10.2013

0 Comment

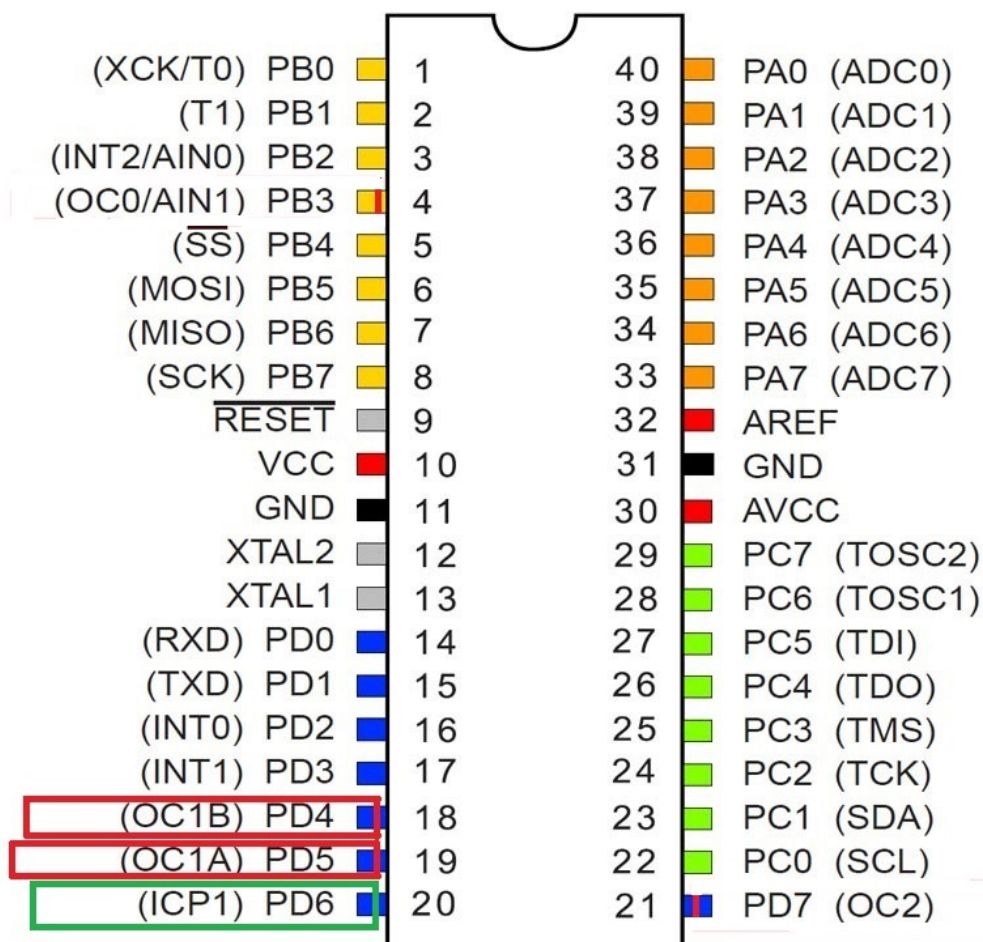
AVR микроконтроллеры для начинающих (урок 6) таймеры-счетчики

Таймеры необходимы чтобы отсчитывать или измерять определенные промежутки времени. Мы уже пользовались функцией которая делает задержку: `_delay_ms();` и `_delay_us();`

Эти функции имеют очень важный недостаток – они заставляют наш МК ничего не делать во время задержки, что уменьшает производительность. Эти две функции по сути цикл в котором переменная увеличивается на единицу каждый такт нашего МК и как только наша функция досчитает необходимое количество тактов, что эквивалентно времени, наша функция заканчивается.

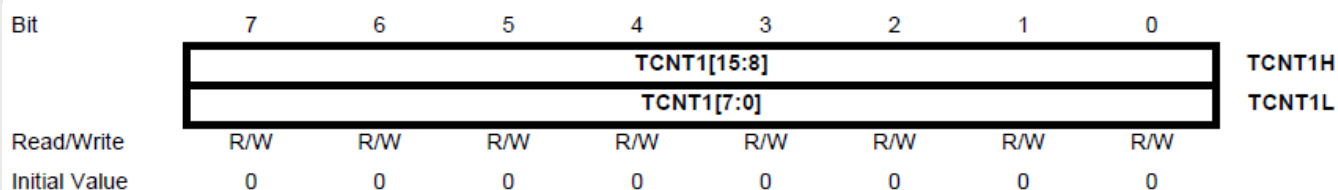
Отличаются Т/С(таймер/счетчик) друг от друга в первую очередь разрядностью: есть 8-разрядные и 16-разрядные. Разрядность Т/С определяется разрядностью счетного регистра этого Т/С.

Подробно рассмотрим 16 разрядный регистр (16-bit Timer/Counter1) в нашем МК (ATmega32A)



Это ножки Т/С1 (**OC1A**)и(**OC1B**) – ножки управляемые таймером счетчиком. (**ICP1**)-это ножка блока захвата.

Регистры таймера-счетчика T1



TCNT1(L/H) – это два 8 битных регистров которые в сумме дают один 16 битный регистр. Когда таймер работает, по каждому импульсу тактового сигнала значение **TCNT1(L/H)** изменяется на единицу. В зависимости от режима работы таймера, счетный регистр может или увеличиваться, или уменьшаться. Регистр **TCNT1(L/H)** можно как читать, так и записывать. Последнее используется когда требуется задать его начальное значение. Когда таймер работает, изменять его содержимое **TCNT1(L/H)** не рекомендуется, так как это блокирует схему сравнения на один такт.

Bit	7	6	5	4	3	2	1	0	
	OCR1A[15:8]								OCR1AH
	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR1A(L/H) - это два 8 битных регистра которые в сумме дают один 16 битный регистр. Его значение постоянно сравнивается со счетным регистром **TCNT1(L/H)**, и в случае совпадения таймер может выполнять какие-то действия – вызывать прерывание, менять состояние вывода **OC1(A/B)** и в зависимости от режима работы. Значение **OCR1A(L/H)** можно как читать, так и записывать.

Bit	7	6	5	4	3	2	1	0	
	OCR1B[15:8]								OCR1BH
	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR1B(L/H) - это два 8 битных регистра которые в сумме дают один 16 битный регистр. Его значение постоянно сравнивается со счетным регистром **TCNT1(L/H)**, и в случае совпадения таймер может выполнять какие-то действия – вызывать прерывание, менять состояние вывода **OC1B** и т.д. в зависимости от режима работы. Значение **OCR1B(L/H)** можно как читать, так и записывать.

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	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 АТмега32, он содержит флаги разрешения прерываний. Таймер T1 может вызывать прерывания при переполнении счетного регистра **TCNT1** и при совпадении счетного регистра с регистрами сравнения **OCR1(A/B)**. Соответственно для таймера T1 в регистре **TIMSK** зарезервированы три бита – это **TOIE1** и **OCIE1A** и **OCIE1B**. Остальные биты относятся к другим таймерам.

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR1A и **TCCR1B** - Это конфигурационные регистры таймера-счетчика T1, он определяет источник тактирования таймера, коэффициент делителя, режим работы таймера-счетчика T1 и поведение выводов **OC1A** и **OC1B**. В этих двух регистрах находятся биты которые почти все биты определяющие как будет работать наш T1.

Биты **ICNC1** и **ICES1**. **ICNC1** включает/выключает схему подавления помех. Если он сброшен в 0, то схема подавления помех выключена и захват производится по первому же активному фронту на выводе **ICP1**. Если бит установлен в 1, схема подавления помех включена и захват производится только в случае 4-ех одинаковых выборок, соответствующих активному фронту сигнала. **ICES1** определяет активный фронт сигнала, то есть фронт по которому блок захвата будет выполнять сохранение счетного регистра. Если

бит сброшен в 0, активным является спадающий фронт. Если установлен в 1, то активным будет нарастающий фронт.

РЕГИСТР ЗАХВАТА: **ICR1** - это регистр захвата. В нем сохраняется значение счетного регистра при сигнале на ножке **ICP1**.

Биты **COM1(A/B)0** и **COM1(A/B)1** - определяют поведение вывода **OC1(A/B)**. Если хоть один из этих битов установлен в 1, то вывод **OC1(A/B)** перестает функционировать как обычный вывод общего назначения и подключается к схеме сравнения таймера счетчика T1. Однако при этом он должен быть еще настроен как выход. Поведение вывода **OC1(A/B)** зависит от режима работы таймера-счетчика T1. В режимах **normal** и **СТС** вывод **OC1(A/B)** ведет себя одинаково, а вот в режимах широтноимпульсной модуляции его поведение отличается.

Table 16-2. для режимов Normal и СТС

COM1A1/COM1B1	COM1A0/COM1B0	Описание
0	0	Ножка отключена от T1
0	1	Состояние вывода меняется на противоположное
1	0	Вывод OC1(A/B) сбрасывается "0"
1	1	Вывод OC1(A/B) устанавливается в "1"

Table 16-2. для режимов Fast PWM

COM1A1/COM1B1	COM1A0/COM1B0	Описание
0	0	Ножка отключена от T1
0	1	Зарезервировано
1	0	Сбрасывается в "0" при прямом счете, устанавливается "1" при обнулении.
1	1	Устанавливается в "1" при прямом счете, сбрасывается в "0" при обнулении.

Table 16-2. для режимов Phase Correct and Phase and Frequency Correct

COM1A1/COM1B1	COM1A0/COM1B0	Описание
0	0	Ножка отключена от T1
0	1	Зарезервировано
1	0	Сбрасывается в "0" при прямом счете, устанавливается "1" при обратном.
1	1	Устанавливается в "1" при прямом счете, сбрасывается "0" при обратном.

Биты **WGM10**, **WGM11**, **WGM12**, **WGM13 (Wave Generator Mode)** - определяют режим работы таймера-счетчика T1. 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.

Режим	WGM				Режим работа таймера-счетчика	Верхний предел	Обновление OCRnx	Установка флага TOVnna:
0	0	0	0	0	Normal	0xFFFF	Сразу после записи	МАКС
1	0	0	0	1	8-битныйPhase Correct PWM	0x00FF	На вершине счета	Нижнем пределе
2	0	0	1	0	9-битныйPhase Correct PWM	0x01FF	На вершине счета	Нижнем пределе
3	0	0	1	1	10-битныйPhase Correct PWM	0x03FF	На вершине счета	Нижнем пределе
4	0	1	0	0	СТС	OCRnA	Сразу после записи	МАКС
5	0	1	0	1	8-битныйFast PWM	0x00FF	На вершине счета	На вершине счета
6	0	1	1	0	9-битныйFast PWM	0x01FF	На вершине счета	На вершине счета
7	0	1	1	1	10- битныйFast PWM	0x03FF	На вершине счета	На вершине счета
8	1	0	0	0	Phase & Frequency Correct PWM	ICRn	На нижнем пределе	Нижнем пределе
9	1	0	0	1	Phase & Frequency Correct PWM	OCRnA	На нижнем пределе	Нижнем пределе
10	1	0	1	0	Phase Correct PWM	ICRn	На вершине счета	Нижнем пределе
11	1	0	1	1	Phase Correct PWM	OCRnA	На вершине счета	Нижнем пределе
12	1	1	0	0	СТС	ICRn	Сразу после записи	МАКС.
13	1	1	0	1	(резерв)	-	-	-
14	1	1	1	0	Fast PWM	ICRn	На вершине счета	На вершине счета
15	1	1	1	1	Fast PWM	OCRnA	На вершине счета	На вершине счета

Режимы:

Режим **Normal**

Это наиболее простой режим работы таймеров/счетчиков. В этом режиме счетный регистр функционирует как обычный суммирующий счетчик. По каждому импульсу тактового сигнала $clkTn$ инкрементируется счетный регистр. При переходе через значение 0xFFFF возникает переполнение, и счет продолжается со значения 0x0000. В том же такте сигнала $clkTn$, в котором обнуляется регистр **TCNTn**. Блоки сравнения таймеров в этом режиме могут использоваться как для генерации прерываний, так и для формирования сигналов. Состояние выходов **OCnx** каждого из блоков сравнения 16-битных таймеров-счетчиков определяется содержимым битов **COMnA1...0**, **OMnB1...0**, **COMnC1...0** регистров **TCCRnA**, как показано в таблице 16-2.

Режим **СТС** (сброс при совпадении)

В этом режиме счетный регистр тоже функционирует как обычный суммирующий счетчик, инкрементирование которого осуществляется по каждому импульсу тактового сигнала $clkTn$. Однако максимально возможное значение счетного регистра и, следовательно, разрешающая способность счетчика определяются либо регистром сравнения блока А **OCRnA**, либо регистром захвата **ICRn**. После достижения максимального значения счет продолжается со значения 0x0000.

Режим **FastPWM** («Быстродействующий ШИМ»)

Режим Fast PWM позволяет генерировать высокочастотный сигнал с широтно-импульсной модуляцией. Счетный регистр в этом режиме функционирует как суммирующий счетчик, инкрементирование которого осуществляется по каждому импульсу тактового сигнала $clkTn$. Состояние

счетчика изменяется от 0×0000 до максимального значения, после чего счетный регистр сбрасывается и цикл повторяется. В зависимости от установок битов **WGMn3..0** максимальное значение счетчика (разрешение ШИМ-сигнала) либо является фиксированным значением (режимы 5, 6, 7 из таблицы 7), либо определяется содержимым определенных регистров таймера-счетчика (режимы 14 и 15).

Режим **Phase Correct PWM** («ШИМ с точной фазой»)

Режим Phase Correct PWM, как и режим Fast PWM, предназначен для генерации сигналов с широтно-импульсной модуляцией. Однако в этом режиме счетный регистр функционирует как реверсивный счетчик, состояние которого сначала изменяется от 0×0000 до максимального значения, а затем обратно до 0×0000. Соответственно, максимальная частота сигнала в этом режиме в 2 раза ниже максимальной частоты сигнала в режиме Fast PWM. В зависимости от установок битов **WGMn3...0** максимальное значение счетчика (разрешение ШИМ-сигнала) либо является фиксированным значением, либо определяется содержимым определенных регистров таймера/счетчика. Как и в режиме Fast PWM, при работе с какими-либо фиксированными значениями модуля счета для задания модуля рекомендуется использовать регистр захвата.

Режим **Phase and Frequency Correct PWM** («ШИМ с точной фазой и частотой»)

Режим Phase and Frequency Correct PWM очень похож на режим Phase Correct PWM. Единственная принципиальная разница между ними — момент обновления содержимого регистра сравнения. Максимальное значение счетчика (разрешение ШИМ-сигнала) в этом режиме может определяться только регистрами **ICRnA** или **OCRnA** таймера-счетчика. В режиме Phase and Frequency Correct PWM каждый период сигнала является полностью симметричным. Это следствие того, что обновление содержимого регистра сравнения происходит в момент достижения счетчиком минимального значения. Поэтому время прямого счета всегда равно времени обратного счета, выходные импульсы симметричны, и соответственно частота генерируемого сигнала остается постоянной.

Биты **CS10, CS11, CS12 (Clock Select)** - определяют источник тактовой частоты для таймера T1 и задают коэффициент делителя. Все возможные состояния описаны в таблице ниже.

CSn12	CSn11	CSn 10	Описание
0	0	0	Нет синхронизации. Таймер-счетчик остановлен.
0	0	1	clk/1 (без деления)
0	1	0	clk/8 (с делением)
0	1	1	clk/64 (с делением)
1	0	0	clk/256 (с делением)
1	0	1	clk/1024 (с делением)
1	1	0	Внешний тактовый источник с выв. Tn. Синхронизация по падающему фронту.
1	1	1	Внешний тактовый источник с выв. Tn. Синхронизация по нарастающему фронту.

Биты **FOC1(A/B) (Force Output Compare)**. Этот биты предназначены для принудительного изменения состояния выводов **OC1(A/B)**. Биты работают только для режимов Normal и CTC. При установке битов в единицу состояние вывода меняется соответственно значениям битов **COM1(A/B)1, COM1(A/B)0**.

FOC1(A/B) биты не вызывают прерывания и не сбрасывает таймер в CTC режиме.

Видео-экскурсия по статье:

AVR микроконтроллеры для начинающих (ур...



AVR микроконтроллеры для начинающих (урок 7) Аналого-цифровой преобразователь

Проекты с использованием таймеров/счетчиков:

- "FAST PWM" ШИМ для двигателя постоянного тока (avr+L293D)
- "normal->PWM" ШИМ на любой ножке МК
- "CTC" Подключение 7 сегментных светодиодных индикаторов (динамическая индикация)

Рубрика: Все посты AVR Уроки AVR

там