

# **Шестнадцатиразрядный таймер/счетчик T1 микроконтроллера ATtiny2313**

**Таймер/счётчик T1** позволяет формировать заданные промежутки времени (для работы в режиме реального времени) и служить генератором периодических сигналов.

### **Основные особенности:**

- два независимых модуля совпадения;
- модуль захвата;
- сброс таймера при совпадении;
- программно изменяемый период в режиме ШИМ;
- четыре независимых источника прерывания.

# Шестнадцатиразрядные регистры

Счетный регистр (TCNT1), регистры совпадения (OCR1A и OCR1B) и регистр захвата (ICR1) – 16-шестнадцати разрядные регистры, к которым центральный процессор может обращаться лишь при помощи 8-разрядной шины данных. Доступ происходит как последовательное чтение или последовательная запись двух байтов информации.

При записи в 16-разрядный регистр, сначала записывается старший байт, а затем младший.

При чтении 16-разрядного регистра сначала читается младший байт, а затем старший.

# Модуль захвата

Модуль захвата может фиксировать внешние события и присваивать им временную метку, указывающую время их возникновения.

Внешний сигнал, соответствующий началу этого события должен поступать на вход микроконтроллера ICP1.

В момент захвата 16-разрядное значение счетчика TCNT1 записывается в регистр захвата ICR1. В этом же периоде тактового сигнала устанавливается флаг захвата ICF1, сигнализируя, что значение TCNT1 скопировано в регистр ICR1. Если прерывание по захвату разрешено (установлен бит ICIE1 в регистре TIMSK), то установка флага захвата вызывает запрос на прерывание по захвату.

# Режимы работы

1. Режим «Normal».
2. Режим «CTC».
3. Режим «Fast PWM» (Быстрый ШИМ).
4. Режим «Phase Correct PWM» (ШИМ, корректный по фазе).
5. Режим «Phase and Frequency Correct PWM» (ШИМ, корректный по фазе и частоте).

# Режим сброса при совпадении

Для установки максимального значения счетчика используется регистр совпадения OCR1A ( $WGM13:0 = 4$ ) или регистр захвата ICR1 ( $WGM13:0 = 12$ ).

## Быстрый ШИМ

Счет происходит всегда в одном направлении: от 0x0000 до максимального значения (0x00FF, 0x01FF, 0x03FF, значение регистра совпадения OCR1A или регистра захвата ICR1), затем счетчик очищается и счет начинается сначала.

Флаг переполнения TOV1 будет устанавливаться каждый раз, когда счетчик будет достигать максимального значения.

Для того, чтобы таймер-счетчик считал до 0xFFFF в регистр OCR1A или регистр ICR1 должно быть записано значение 0xFFFF.

Генерация сигнала ШИМ на выводах OC1x определяется установкой бит COM1x1:COM1x0 в регистре TCCR1A.

## **ШИМ, корректный по фазе**

Счетчик периодически изменяет направление своего счета (от минимального значения 0x0000 до максимального значения (0x00FF, 0x01FF, 0x03FF, значение регистра совпадения OCR1A или регистра захвата ICR1), затем от максимального до минимального, далее направление счета снова меняется и все повторяется сначала).

Для того, чтобы таймер-счетчик считал до 0xFFFF в регистр OCR1A или регистр ICR1 должно быть записано значение 0xFFFF.

Флаг переполнения TOV1 будет устанавливаться каждый раз, когда счетчик будет достигать минимального значения.

Генерация сигнала ШИМ на выводах OC1x определяется установкой бит COM1x1:COM1A0 в регистре TCCR1A.

## **ШИМ, корректный по фазе и частоте**

Счетчик работает так же, как и в режиме «ШИМ, корректный по фазе».

Максимальное значение счетчика определяется значением регистра совпадения OCR1A или регистра захвата ICR1.

Использование данного режима предпочтительнее, если требуется менять 7 максимальное значение счетчика во время выполнения программы.

## Счетный регистр TCNT1 – TCNT1H и TCNT1L

Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
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	

## Регистр захвата ICR1 – ICR1H и ICR1L

Bit	7	6	5	4	3	2	1	0	
	ICR1[15:8]								ICR1H
	ICR1[7:0]								ICR1L
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	



## Регистр сравнения (канал A) OCR1A – OCR1AH и OCR1AL

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	

## Регистр сравнения (канал B) OCR1B – OCR1BH и OCR1BL

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	

## Регистр маски – TIMSK

Bit	7	6	5	4	3	2	1	0	
	TOIE1	OCIE1A	OCIE1B	–	ICIE1	OCIE0B	TOIE0	OCIE0A	TIMSK
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Бит 7 – TOIE1 : разрешение прерывания по переполнению.

Бит 6 - OCIE1A : разрешение прерывания по совпадению в канале A.

Бит 5 – OCIE1B : разрешение прерывания по совпадению в канале B.

Бит 3 – ICIE1 : разрешение прерывания по захвату.

## Регистр флагов – TIFR

Bit	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	–	ICF1	OCF0B	TOV0	OCF0A	TIFR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Бит 7 – TOV1 : флаг переполнения.

Бит 6 – OCF1A : флаг совпадения в канале А.

Бит 5 – OCF0B : флаг совпадения в канале В.

Бит 3 – ICF1 : флаг захвата.

## Регистр А управления таймера/счетчика T1 – TCCR1A

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

Биты 7:6 – COM1A1:0 : режимы работы схемы вывода сигнала совпадения (канал A).

Биты 5:4 – COM1B1:0 : режимы работы схемы вывода сигнала совпадения (канал B).

Биты 1:0 – WGM11:0 : режим работы таймера.

COM1A1 / COM1B1	COM1A0 / COM1B0	Описание
0	0	Стандартный режим порта. Выход OC1A / OC1B не подключен
0	1	Переключение OC1A / OC1B на противоположное в момент совпадения
1	0	Сброс OC1A / OC1B в момент совпадения
1	1	Установка OC1A / OC1B в момент совпадения

COM1A1 / COM1B1	COM1A0 / COM1B0	Описание
0	0	Стандартный режим порта. Выход OC1A / OC1B не подключен
0	1	WGM13=0: Стандартный режим порта. Выход OC1A / OC1B не подключен. WGM13=1: Переключение OC1A в момент совпадения. Для OC1B зарезервировано.
1	0	Сброс OC1A / OC1B в момент совпадения, установка OC1A / OC1B при достижении счетчиком максимального значения
1	1	Установка OC1A / OC1B в момент совпадения, сброс при достижении счетчиком максимального значения

Phase Correct PMW или  
Phase and Frequency Correct PWM

COM1A1 / COM1B1	COM1A0 / COM1B0	Описание
0	0	Стандартный режим порта. Выход OC1A / OC1B не подключен
0	1	WGM13=0: Стандартный режим порта. Выход OC1A / OC1B не подключен. WGM13=1: Переключение OC1A в момент совпадения. Для OC1B зарезервировано.
1	0	Сброс OC1A / OC1B в момент совпадения при прямом счете. Установка OC1A / OC1B в момент совпадения при обратном счете.
1	1	Установка OC1A / OC1B в момент совпадения при прямом счете. Сброс OC1A / OC1B в момент совпадения при обратном счете.

# Регистр В управления таймера/счетчика T1 – TCCR1B

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	

**Бит 7 – ICNC1 : Разрешение работы шумоподавителя на входе захвата.**

Установкой бита активизируется схема шумоподавителя на входе захвата.

**Бит 6 – ICES1 : Выбор активного фронта сигнала захвата**

Когда бит сброшен, захват происходит по заднему фронту входного сигнала. Если бит установлен – по переднему фронту.

Если происходит захват, то текущее значение счетного регистра записывается в регистр захвата ICR1. Одновременно устанавливается флаг ICF1, который может использоваться для вызова прерывания по захвату.

Если регистр захвата ICR1 используется для хранения максимального значения, то вход ICP1 отключен и функция захвата заблокирована.

**Биты 4:3 – WGM13:2 : Режим работы таймера.**

**Биты 2:0 – CS12:0 : Выбор тактовой частоты.**

WGM13	WGM12	WGM11	WGM10	Название режима	Верхний предел	Установка TOV0
0	0	0	0	Normal	0xFFFF	0xFFFF
0	0	0	1	Phase Correct PWM, 8 бит	0x00FF	0x0000
0	0	1	0	Phase Correct PWM, 9 бит	0x01FF	0x0000
0	0	1	1	Phase Correct PWM, 10 бит	0x03FF	0x0000
0	1	0	0	CTC	OCR1A	0xFFFF
0	1	0	1	Fast PWM, 8 бит	0x00FF	0x00FF
0	1	1	0	Fast PWM, 9 бит	0x01FF	0x01FF
0	1	1	1	Fast PWM, 10 бит	0x03FF	0x03FF
1	0	0	0	Phase and Frequency Correct PWM	ICR1	0x0000
1	0	0	1	Phase and Frequency Correct PWM	OCR1A	0x0000
1	0	1	0	Phase Correct PWM	ICR1	0x0000
1	0	1	1	Phase Correct PWM	OCR1A	0x0000
1	1	0	0	CTC	ICR1	0xFFFF
1	1	1	0	Fast PWM	ICR1	ICR1
1	1	1	1	Fast PWM	OCR1A	OCR1A



# Выбор тактовой частоты

TCCR1B

CS12	CS11	CS10	Описание
0	0	0	Нет источника сигнала (таймер/счетчик остановлен)
0	0	1	ТЧ / 1 (нет предварительного делителя)
0	1	0	ТЧ / 8
0	1	1	ТЧ / 64
1	0	0	ТЧ / 256
1	0	1	ТЧ / 1024
1	1	0	Внешний источник сигнала, вход Т1. Синхронизация по заднему фронту.
1	1	1	Внешний источник сигнала, вход Т1. Синхронизация по переднему фронту.

ТЧ – тактовая частота микроконтроллера

```

1  #include <avr/io.h>
2
3  int main(void)
4  {
5      // настройка вывода PD0 на вывод данных
6      DDRD = 0x01;
7
8      // запуск Т/С Т1 (предварительный делитель 1024)
9      TCCR1B = 0x05;
10
11     while(1)
12     {
13         // если значение СР TCNT1 = 1000, то включаем
14         // светодиод, подключенный к выводу PD0
15         if (TCNT1 == 1000) {PORTD = 1;}
16
17         // если значение СР TCNT1 = 2000, то выключаем
18         // светодиод, подключенный к выводу PD0
19         // и сбрасываем значение счетного регистра
20         if (TCNT1 == 2000)
21         {
22             PORTD = 0;
23             TCNT1 = 0;
24         }
25     }
26 }

```

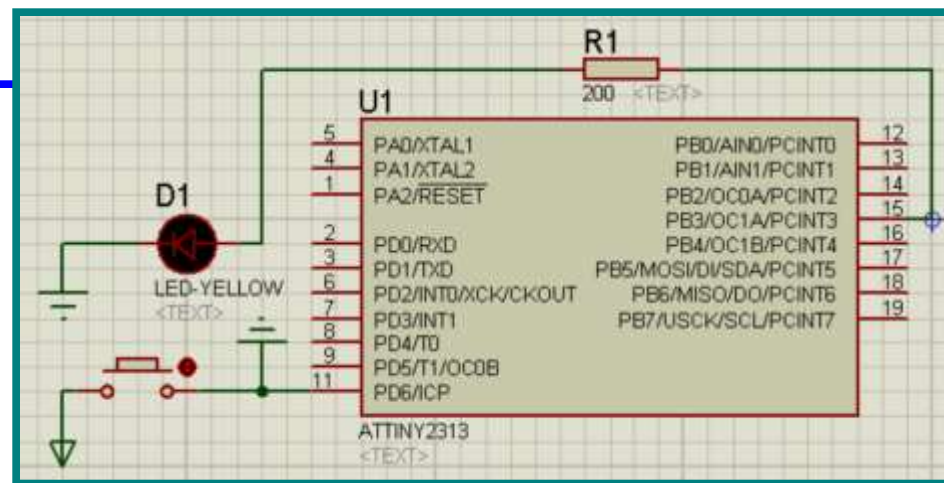
Пример программы, демонстрирующей работу Т/С Т1 в режиме «Normal»: переключение светодиода (вывод микроконтроллера PD0) через 1000 отсчетов счетчика



```

2  #include <avr/interrupt.h>
3
4  ISR(TIMER1_CAPT_vect)
5  {
6      cli();
7      // остановка таймера
8      TCCR1B = 0x00;
9      TCNT1 = 0;
10     // отключение прерывания по захвату
11     TIMSK &= ~(1 << ICIE1);
12     // изменение режима на CTC с максимальным значением в ICR1
13     TCCR1B = TCCR1B | (1 << WGM12) | (1 << WGM13);
14     // настройка схемы вывода сигнала совпадения по каналу A
15     // (переключение в момент совпадения)
16     OCR1A = ICR1;
17     TCCR1A |= (1 << COM1A0);
18     // запуск Т/С Т1 (предварительный делитель 1024)
19     TCCR1B |= (1 << CS12) | (1 << CS10);
20     sei();
21 }
22
23 int main(void)
24 {
25     DDRB = 1 << 3;
26     // разрешение прерывания по захвату
27     TIMSK = 1 << ICIE1;
28     // запуск Т/С Т1 (предварительный делитель 1024)
29     TCCR1B = 0x05;
30     sei();
31
32     while(1)
33     { }
34 }

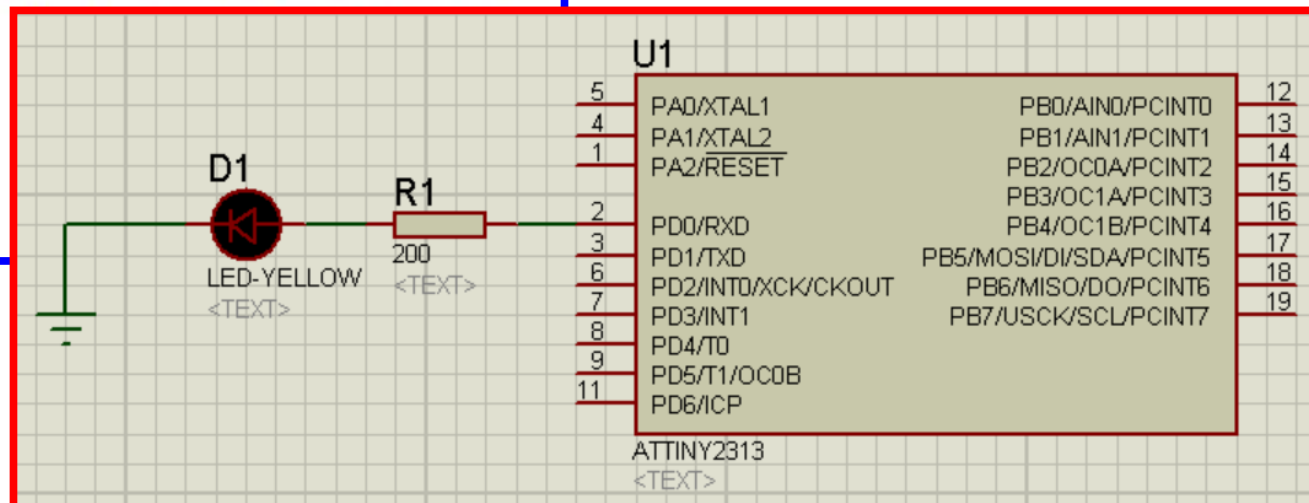
```



Т/С Т1 начинает работу в режиме «Normal» пока не будет нажата кнопка (вывод ICP); после нажатия на кнопку вызывается прерывание по захвату, в котором происходит запрет этого прерывания и включается режим CTC; в режиме CTC светодиод начинает переключаться через равные промежутки времени (промежуток времени определен захватом и равен количеству отсчетов, прошедших от момента запуска микроконтроллера до нажатия кнопки).

Пример программы на языке Си, демонстрирующей работу прерывания по переполнению в режиме «Быстрый ШИМ» (10 бит): переключение светодиода (вывод микроконтроллера PD0) через 1024 отсчета счетчика.

```
1  #include <avr/io.h>
2  #include <avr/interrupt.h>
3
4  ISR(TIMER1_OVF_vect)
5  {
6      PORTD ^= 1;
7  }
8
9  int main(void)
10 {
11     // настройка вывода
12     DDRD = 1;
13     // настройка режима Fast PWM (10 бит)
14     TCCR1A = (1 << WGM10) | (1 << WGM11);
15     // разрешение прерывания по переполнению
16     TIMSK = (1 << TOIE1);
17     // запуск T/C T1 (предварительный делитель 1024)
18     TCCR1B = 0x05;
19     sei();
20
21     while(1)
22     { }
23 }
```



Пример программы на языке Си, демонстрирующий генерацию сигнала ШИМ (работу схемы вывода сигнала совпадения (канал В) таймера/счетчика Т1 в режиме «ШИМ, корректный по фазе»): 951 отсчет счетчика светодиод (вывод микроконтроллера OC1В) выключен, 72 отсчета – включен.

```
1  #include <avr/io.h>
2  #include <avr/interrupt.h>
3
4  int main(void)
5  {
6      // настройка вывода
7      DDRB = 1 << 4;
8      // настройка режима Phase Correct PWM (9 бит)
9      TCCR1A = (1 << WGM11);
10
11     // настройка генерации сигнала ШИМ
12     OCR1B = 36;
13     TCCR1A |= (1 << COM1B1);
14
15     // запуск Т/С Т1 (предварительный делитель 1024)
16     TCCR1B = 0x05;
17     sei();
18
19     while(1)
20     { }
21 }
```

