

Раздел 3

Программирование различных режимов таймера T0.

Теоретическая часть

В предыдущем разделе был рассмотрен лишь один режим работы таймера – «сброс при совпадении». Теперь рассмотрим возможности таймеров более подробно.

Кроме того, поскольку возможности эмулятора AVR Studio ограничены, то программы данного раздела желательно проверять на реальных микроконтроллерах с использованием программатора.

Режимы работы и регистры таймера

В состав таймера-счетчика T0 входят 3 регистра ввода/вывода:

TCNT0 – счетный регистр. В зависимости от режима работы содержимое этого регистра инкрементируется или декрементируется по каждому импульсу тактового сигнала.

OCR0 – регистр сравнения. С содержимым этого регистра сравнивается TCNT0 в случае равенства генерируется прерывание «по совпадению таймера-счетчика T0».

TCCR0 – регистр управления. Задаёт режим работы T0.

Таблица 1. Разряды регистра TCCR0

Разряд	Название	Описание		
7	FOC0	Принудительное изменение состояния вывода OC0		
3,6	WGM01:WGM00	Режим работы таймера-счетчика		
		WGM01(бит 3)	WGM00(бит 6)	Режим работы
		0	0	Normal
		0	1	ШИМ с точной фазой
		1	0	Сброс при совпадении
1	1	Быстрый ШИМ		
5,4	COM01:COM00	Режим работы блока сравнения для режимов «Normal» и «Сброс при совпадении»		
		COM01(бит5)	COM00(бит4)	Режим работы
		0	0	Таймер-счетчик отключен от вывода OC0
		0	1	Состояние вывода OC0 меняется на противоположное
		1	0	Вывод сбрасывается в 0
1	1	Вывод устанавливается в 1		
2,1,0	CS02...CS00	Управление тактовым сигналом. Задаётся режим работы делителя.		

CS02	CS02	CS02	Коэффициент деления пределителя
0	0	0	Таймер-счетчик остановлен
0	0	1	1
0	1	0	8
0	1	1	64
1	0	0	256
1	0	1	1024
1	1	0	Тактовый сигнал подается с внешнего вывода T0. Счет осуществляется по срезу.
1	1	1	Тактовый сигнал подается с внешнего вывода T0. Счет осуществляется по фронту.

1. Режим Normal

Счетный регистр функционирует, как обычный суммирующий счетчик. Генерируется прерывание по переполнению таймера-счетчика (если оно разрешено).

Кроме этого, по переполнению можно менять состояние вывода OC0. В таблице 1 описаны различные режимы работы этого вывода в зависимости от битов COM01:COM00. Например, если COM01=0, COM00=1, то при переполнении состояние вывода OC0 меняется на противоположное, таким образом на OC можем получить низкочастотный сигнал.

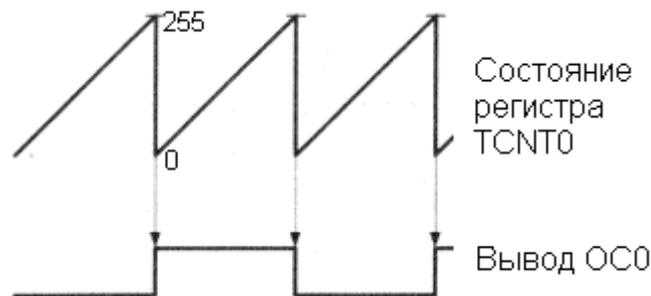


Рис 1. Генерация сигнала на выводе OC0

2. Режим сброс при совпадении

Режим работает аналогично Normal. Отличие лишь в том, что счет происходит не до 255, а до числа, указанного в регистре сравнения OCR0.

Аналогично первому случаю возможно управление выводом OC:

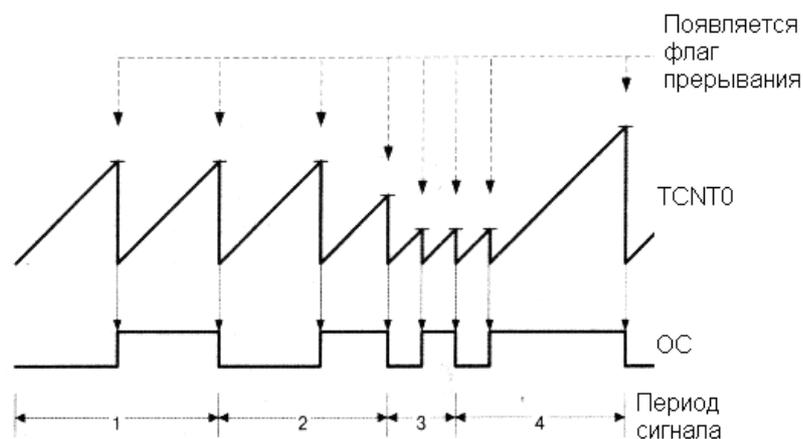


Рис 2. Работа таймера в режиме сброс при совпадении.

По рисунку видно, что возможно изменение периода следования импульсов путем изменения содержимого регистра OCR0 (период пропорционален числу, записанному в этом регистре).

3. Режим «быстродействующий ШИМ»

Данный режим позволяет генерировать сигнал с широтно-импульсной модуляцией. Понятие широтно-импульсная модуляция означает, что время нахождения сигнала в единичном состоянии определяется по некоторому закону.

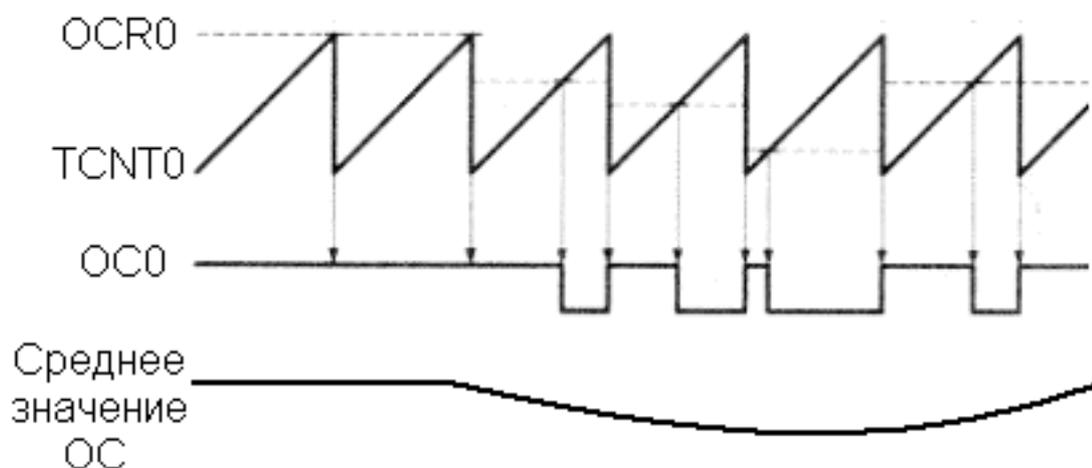


Рис 3. Работа таймера в режиме «Быстродействующий ШИМ»

Рассмотрим пример на Рис 3. В первые 2 цикла OCR0=255. Вывод OC0 находится все время в единичном состоянии. Затем OCR0 уменьшаем – появляется небольшой временной интервал, при котором OC0=0. Чем меньше число в OCR0, тем короче единичный интервал и длиннее нулевой интервал вывода OC.

Если сигнал с вывода OC0 сгладить (усреднить), например интегрирующей цепочкой, то получится непрерывный сигнал. Величина этого сигнала напрямую зависит от содержимого OCR0.

Получили цифро-аналоговый преобразователь: числа из OCR0 преобразуются в уровень сигнала после сглаживающего фильтра.

Поведение вывода OC определяется содержимым COM01:COM00

Обратите внимание, что назначение COM01:COM00 для режимов ШИМ несколько отличается от режимов Normal и сброс при совпадении.

Таблица 2. Параметра работы блока сравнения в режиме «Быстродействующий ШИМ»

COM01 (бит5)	COM00 (бит4)	Режим работы
0	0	Таймер-счетчик отключен от вывода OC0
0	1	Зарезервировано
1	0	Сбрасывается в 0 при TCNT0=OCR0 Устанавливается в 1 при TCNT0=0
1	1	Устанавливается в 1 при TCNT0=OCR0 Сбрасывается в 0 при TCNT0=0

Практическая часть

1. Режим сброс при совпадении

- 1.1. Рассмотрим программу, которая выводит на выход OC0 (нулевой бит порта В) сигнал с частотой около 8кГц (пусть период сигнала будет = 1024 такта, тогда частота = $8\text{МГц}/1024=7,8125\text{КГц}\approx 8\text{КГц}$).

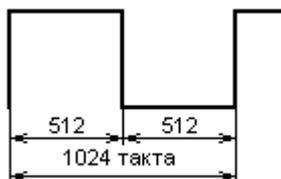


Рис 4. Сигнал на выводе OC0

- 1.1.1. Введите следующую программу:

```
#include <avr/io.h>
int main(void)
{
    TCCR0=0b10011010;
    OCR0=63;
}
```

- 1.1.2. Определите, что означает каждый бит регистра TCCR0 в программе.

Коэффициент деления = 8, значит каждые 8 тактов значение счетчика TCNT0 будет увеличиваться на 1 (8 тактов при чистоте 8 МГц выполняются за 1 микросекунду, значит каждую микросекунду значение счетчика увеличивается). Как только оно превысит число 63 ($64*8$ тактов= 512 тактов $\sim 64\text{мкс}$), состояние вывода OC0 поменяется на противоположное.

- 1.1.3. Откомпилируйте программу

- 1.1.4. Откройте hex-файл в эмуляторе AVR-Studio.

- 1.1.5. В меню Debug\AVR Simulator options выберите частоту 8МГц.

- 1.1.6. Запустите программу в эмуляторе AVRStudio в замедленном автоматическом режиме, нажав . На нулевом выводе порта В будет периодически меняться ноль и единица.

- 1.1.7. Перед строкой **TCCR0=0b10011010**; вставьте **DDRB=0x01**; Перекомпилируйте программу. Введите программу в микроконтроллер с помощью программатора К выводу B0 необходимо подключить динамик по схеме:

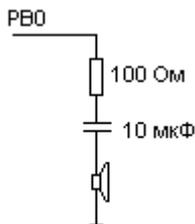


Рис 5. Подключение динамика

- 1.2. Измените написанную программу так, чтобы период сигнала менялся от 512 до $255*512$ тактов в зависимости от числа введенного в порт С. Для этого необходимо коэффициент деления установить равным 256, а число из порта С (PINС) в бесконечном цикле направлять в регистр OCR0 (как организовать бесконечный цикл смотрите в приложении в конце пятого пункта). Запрограммируйте микроконтроллер.

2. Режим «Быстродействующий ШИМ»
- 2.1. Рассмотрим программу, которая превращает микроконтроллер в ЦАП. На порт С (PINC) подается число нужно считать это число и записать его в регистр таймера, отвечающий за среднее значение сигнала на выходе ОС0. Для этого воспользуемся режимом «Быстродействующий ШИМ».

- 2.2. **Введите следующую программу:**

```
#include <avr/io.h>
int main(void)
{
  DDRB=0x01;
  TCCR0=0b11101001;
  for (;;)
  {
    OCR0=PINC;
  }
}
```

- 2.3. Используя таблицу 1 и таблицу 2 из теоретической части **определите режим работы таймера, поведение вывода ОС0 и коэффициент деления предделителя.**

- 2.4. Рассмотрим работу программы.

Например в порт С ввели число 3.

По каждому такту состояние регистра TCNT0 увеличивается на единицу. Таким образом производится отсчет от 0 до 255. При достижении значения 255 счетчик обнуляется и счет происходит заново.

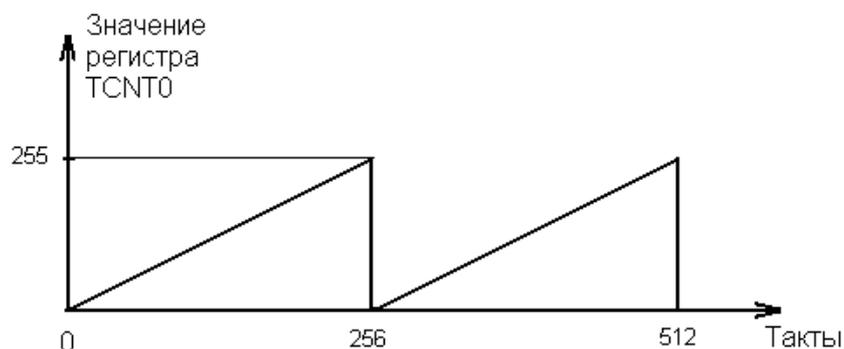


Рис 6. Рост значения регистра TCNT0.

При $TCNT0 < OCR0$ состояние на выводе ОС0=1. При $TCNT0 \geq OCR0$ состояние на выводе ОС0=0. Так как в нашем случае $OCR0=3$ (на порт С подали число 3), то вывод ОС0=1 будет лишь 3 такта. Остальные 253 такта $OCR=0$.

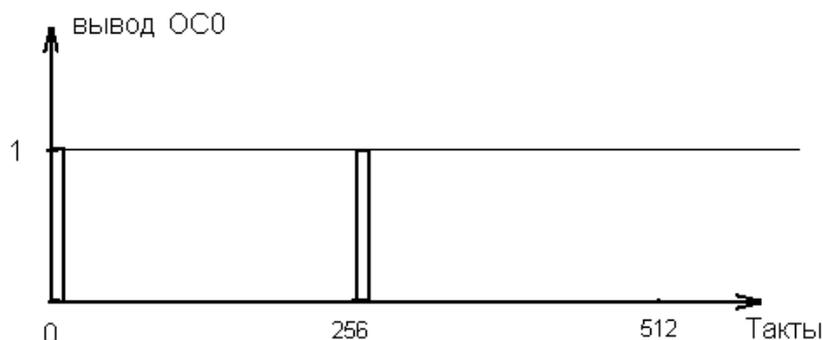


Рис 7. Сигнал на выводе ОС0.

Если к выводу ОС0 подключить светодиод, то он будет гореть лишь 3/256 часть времени и не гореть 253/256 времени. Поскольку чередование «горит/не горит» происходит очень быстро, то это будет определять яркость светодиода. В нашем случае он должен гореть с интенсивностью 3/256 от номинальной (очень слабо).

2.5. Запрограммируйте микроконтроллер

3. Творческое задание

Реализуйте один из вариантов задания (номер варианта выдаётся преподавателем либо выбирается студентом с согласия преподавателя).

1 Вариант

Напишите программу к микроконтроллеру, который будет считать пробег автомобиля. При достижении 100000 км на один из портов нужно выдать сигнал FF, что будет означать окончание гарантийного обслуживания. На другой порт выдается значение пробега в тысячах километров. С датчиков автомобиля поступает сигнал  каждый метр. Этот сигнал направляется на один из входов микроконтроллера.

2 Вариант

Напишите программу – милли-секундомер. При перепаде появлении единицы на одном из входов микроконтроллера, он начинает отсчитывать время с нуля в миллисекундах и выдает в 16-ричной форме на порт А младшую часть и на порт В старшую часть отсчитанного отрезка времени.

3 Вариант

Кодовый замок. На порт А подается число, набранное с клавиатуры в 16-ричном виде, а на порт В – тайное число открытия замка. При равенстве этих чисел на порт С подается число FF, при неравенстве – число 00. При неправильном вводе следующий ввод возможен только через определенный промежуток времени (для простоты пусть это будет 100мкс). При трех неправильных вводах микроконтроллер блокируется. Число 00, поданное на порт А игнорировать (оно означает что ничего не вводится).

4 Вариант (повышенной сложности)

Написать программу, которая бы выводила синусоидальный сигнал звуковой частоты (вместо ЦАП использовать режим ШИМ).

5 Вариант

Сформулировать свою задачу соответствующей сложности.

Контрольные вопросы

1. Каково назначение регистров TCNT0, OCR0, TCCR0 ?
2. Как, используя ШИМ, получить ЦАП?
3. Для чего нужен делитель частоты?
4. Что означает TCCR0=0b11111001;