

Лабораторная работа №8. Таймеры AVR Atmega 328P. Режим счета

Цель: изучение особенностей применения таймеров в микропроцессорных системах.

Оборудование на 1 рабочее место:

Компьютер с ОС Windows	1 шт
Отладочная плата МК AVR	1 шт.
Программатор USBISP	1 шт.
Логический анализатор	1 шт.

Используемое ПО:

Интегрированная среда разработки Atmel Studio 7.0 (или AVR Studio 4.19)

Программа для загрузки программного кода в микроконтроллер AVRDUDEPROG

Программа для анализа временных диаграмм Saleae Logic

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Таймеры

Таймер – это устройство, обеспечивающее подсчет импульсов. В основе работы таймеров лежит счетный регистр, значение которого увеличивается по фронту импульсов анализируемого сигнала. В качестве источника импульса обычно используется генератор синхросигнала. Подсчет количества импульсов осуществляется без участия центрального процессора. При переполнении счетного регистра таймера или при достижении определенного значения (зависит от режима работы) в системе формируется признак соответствующего события. Количество импульсов, которое подсчитывает счётчик от начального состояния до сброса, называется модулем счёта. Например, для 8 разрядного счётного регистра, модуль счёта равен 256. Для изменения модуля счёта используется регистр сравнения, по достижении значения которого будет выполнен сброс. Таймеры имеют направление счёта, т.е. могут прибавлять или вычитать единицу. Если счётчик таймера может работать в обоих направлениях, он называется реверсивный. Сброс счётчика таймера может осуществляться в двух случаях – достижение нулевого или максимального значения или достижение некоторого конкретного значения.

Таймеры используются в следующих ситуациях:

- выполнение операций через строго заданные интервалы времени: чтение данных с датчиков, изменение выходных сигналов;
- формирование широтно-импульсной модуляции;
- измерение времени между сигналами от внешних устройств (определение частоты сигнала);
- выделение кванта времени для отдельной задачи – организация псевдопараллельных вычислений.

Таймеры ATmega328P

У AVR ATmega328P три таймера (таймер/счетчик, Timer/Counter, TC): Timer0, Timer1 и Timer2 (TC0, TC1 и TC2). Также имеется сторожевой таймер WDT, который можно использовать для защиты от сбоев или как механизм программного сброса (сторожевой таймер в данной работе не рассматривается).

Timer0 и Timer2 являются двумя схожими 8-битными таймерами. Максимальное значение счета – 255. 256-й импульс вызывает сброс счетчика.

Timer1 – это 16-битный таймер. Максимальное значение счета – 65535.

Режимы работы таймеров

Соответствие режимов работы таймерам, на которых они могут быть использованы, представлено в таблице 8.1.

Таблица 8.1 – Соответствие режимов работы таймерам ATmega328P

Название режима	Таймеры ATmega328P
Обычный режим (счет импульсов до переполнения)	Timer0(TC0), Timer1(TC1), Timer2(TC2)
Режим подсчета импульсов (счет импульсов до совпадения с определенным значением и сброс)	Timer0(TC0), Timer1(TC1), Timer2(TC2)
Режим ШИМ (формирование сигнала широтно-импульсной модуляции)	Timer0(TC0), Timer1(TC1), Timer2(TC2)
Режим коррекции фазы ШИМ	Timer0(TC0), Timer1(TC1), Timer2(TC2)
Режим коррекции частоты и фазы ШИМ	Timer1(TC1)
Режим захвата	Timer1(TC1)
Асинхронный режим работы	Timer2(TC2)

Обычный режим работы. Таймер считает тактовые импульсы и при переполнении счетного регистра устанавливает флаг переполнения, который может использоваться для вызова прерывания. При этом счетный регистр сбрасывается в 0 и подсчет импульсов начинается сначала. Принцип работы показан на рис. 6.1 с использованием следующих обозначений:

- CLK – тактовый сигнал;
- T_{clk} – период тактового сигнала;
- cnt – значение счетного регистра (максимум 2_n-1 – разрядность счётного регистра);
- $T_{counter}$ – период счетчика.

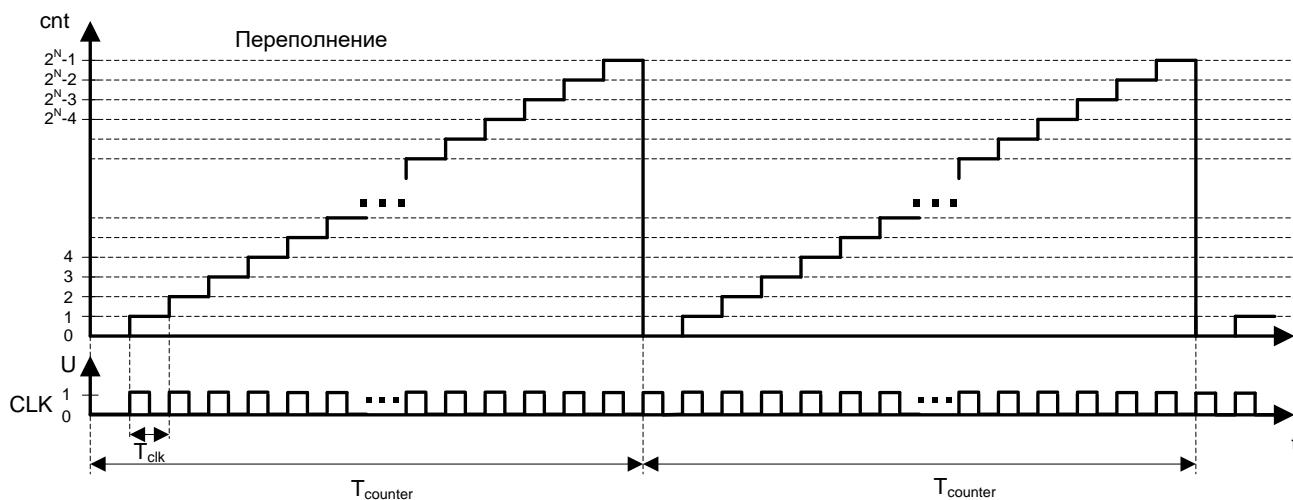


Рисунок 8.1 – Счет импульсов со сбросом при переполнении

Режим подсчета импульсов (также называется «сброс при совпадении» Clear Timer on Compare, CTC). Режим аналогичен предыдущему, но сброс счетного регистра выполняется не при переполнении, а при совпадении с определенным значением (CMP), записанном в одном из регистров сравнения. При этом формируется соответствующий флаг и может быть вызвано прерывание. Принцип работы показан на рисунке 8.2.

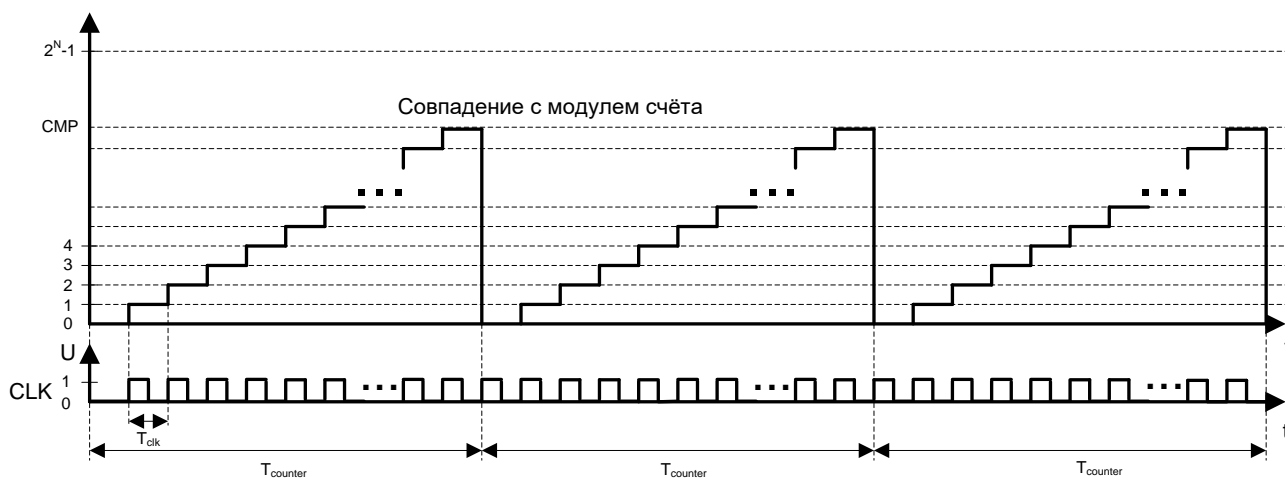


Рисунок 8.2 – Счет импульсов со сбросом при совпадении

Режим широтно-импульсной модуляции (ШИМ, PWM). В технической документации режим называется Fast PWM – быстрый ШИМ. В данном режиме изменяется ширина импульса в зависимости от значения, записанного в регистр сравнения. При этом период сигнала остается постоянным. На рисунке 8.3 представлен принцип работы счетчика в режиме ШИМ.

PWM+ – это выходной сигнал в не инверсном режиме, PWM- – выходной сигнал в инверсном режиме. PWM Count 1 и PWM Count 2 – значения, записываемые в регистр сравнения для определения ширины импульса.

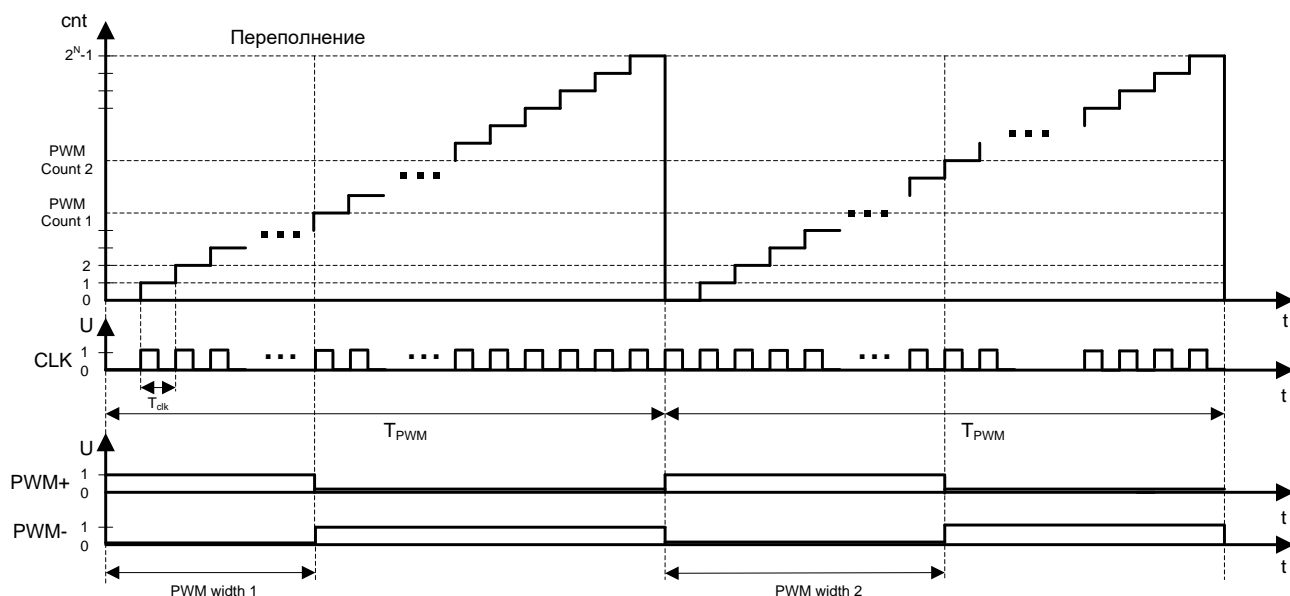


Рисунок 8.3 – Режим ШИМ

Режим коррекции фазы ШИМ. В обычном режиме ШИМ фаза выходного сигнала плавает в моменты изменения скважности (ширины импульса). В режиме коррекции фазы ШИМ (при достижении максимального значения) счетный регистр таймера/счетчика начинает декрементироваться. Это происходит циклически. При использовании данного режима частота ШИМ падает в два раза по сравнению с режимом Fast PWM.

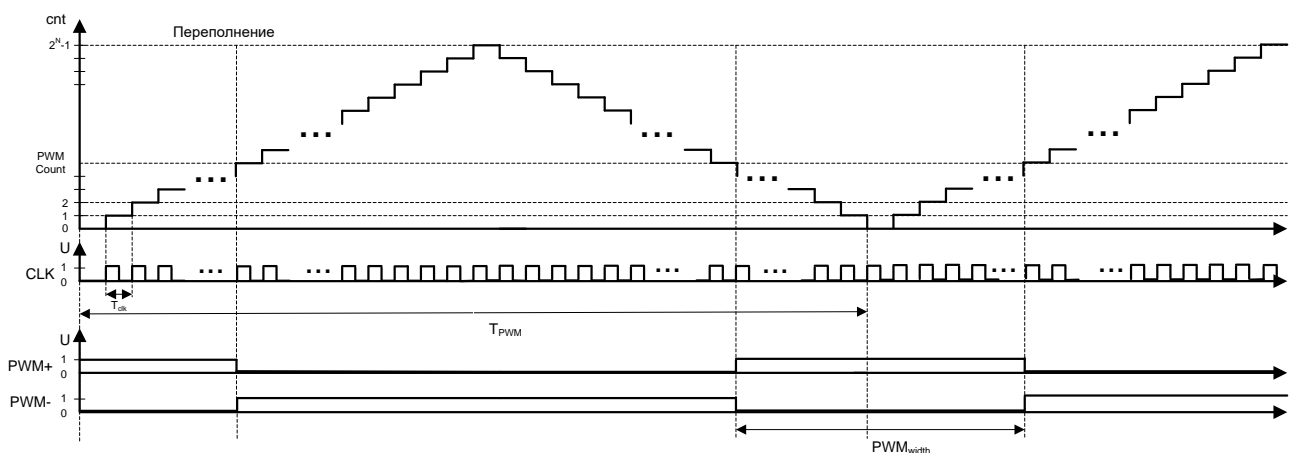


Рисунок 8.4 – Режим ШИМ с коррекцией фазы

Режим коррекции фазы и частоты ШИМ. Отличается от режима коррекции фазы ШИМ моментом обновления регистра сравнения. Регистр сравнения обновляется, когда значение счетного регистра достигает минимума.

Режим захвата. Фиксирует момент изменения сигнала на линии ICP1 (PortB0) или изменение на выходе аналогового компаратора и сохраняет значение счетного регистра в регистр захвата. Может использоваться для определения частоты входящего сигнала или измерения времени между внешними событиями.

Асинхронный режим работы. Асинхронный режим работы предполагает подключение внешнего кварцевого резонатора 32 кГц к микроконтроллеру. Чаще всего этот режим используется для работы таймера/счетчика T2 в качестве часов реального времени.

Регистры таймеров AVR Timer0 (TC0)

Регистр TCCR0A (TC0 Control Register A) регистр управления А таймера 0.

7	6	5	4	3	2	1	0
COM0A1	COM0A0	COM0B1	COM0B0			WGM01	WGM00
R/W	R/W	R/W	R/W			R/W	R/W

Биты COM0A1 (7) и COM0A0 (6) влияют на то, какой сигнал появится на выводе OC0A (PORTD6) при совпадении с А (совпадение значения счетного регистра **TCNT0** с значением регистра сравнения **OCR0A**). Назначение битов представлено в таблице 8.2.

Биты COM0B1 (7) и COM0B0 (6) определяют, какой сигнал появится на выводе OC0B (PORTD5) при совпадении с В (совпадение значения счетного регистра **TCNT0** с значением регистра сравнения **OCR0B**). Назначение аналогично COM0A1, COM0A0 (таблица 8.2).

Биты WGM01 (1) , WGM00 (0) регистра TCCR0A устанавливают режим работы таймера/счетчика T0 (таблица 8.3).

Таблица 8.2 – Назначение битов COM0A1, COM0A0

COM0A1	COM0A0	Вывод OC0A
Обычный режим, режим СТС		
0	0	Вывод OC0A не функционирует.
0	1	Изменение состояния вывода OC0A на противоположное при совпадении с А.
1	0	Сброс вывода OC0A в 0 при совпадении с А.
1	1	Установка вывода OC0A в 1 при совпадении с А.
Режим ШИМ		
0	0	Вывод OC0A не функционирует.
0	1	Если бит WGM02 регистра TCCR0B установлен в 0, вывод OC0A не функционирует. Если бит WGM02 регистра TCCR0B установлен в 1, изменение состояния вывода OC0A на противоположное при совпадении с А.
1	0	Сброс вывода OC0A в 0 при совпадении с А, установка вывода OC0A в 1 если регистр TCNT0 принимает значение 0x00 (не инверсный режим).
1	1	Установка вывода OC0A в 1 при совпадении с А, установка вывода OC0A в 0 если регистр TCNT0 принимает значение 0x00 (инверсный режим).
Режим коррекции фазы ШИМ		
0	0	Вывод OC0A не функционирует.
0	1	Если бит WGM02 регистра TCCR0B установлен в 0, вывод OC0A не функционирует. Если бит WGM02 регистра TCCR0B установлен в 0, изменение состояния вывода OC0A на противоположное.
1	0	Сброс вывода OC0A в 0 при совпадении с А во время увеличения значения счетчика, установка вывода OC0A в 1 при совпадении с А во время уменьшения значения счетчика.
1	1	Установка вывода OC0A в 1 при совпадении с А во время увеличения значения счетчика, сброс вывода OC0A в 0 при совпадении с А во время уменьшения значения счетчика.

Таблица -8.3 – Назначение битов WGM01, WGM00

WGM01	WGM00	Режим
0	0	Обычный режим.
0	1	Режим коррекции фазы ШИМ.
1	0	Режим подсчета импульсов – СТС (сброс при совпадении).
1	1	Режим ШИМ.

Регистр TCCR0B (TC0 Control Register B) – регистр управления В таймера 0.

7	6	5	4	3	2	1	0
FOC0A	FOC0B			WGM02	CS02	CS01	CS00
W	W			R/W	R/W	R/W	R/W

Биты FOC0A (7) и FOC0B (6) принудительно устанавливают значения на выводах OC0A(PortD6) и OC0B(PortD5).

Бит WGM02 (3) управляет режимом работы выводов OC0A и OC0B при [COM0A1:COM0A0] = 01₂ в режимах ШИМ и коррекции фазы ШИМ (Таблица - 6.2).

Биты CS02 (2), CS01 (1), CS00 (0) определяют режим тактирования и делителя тактовой частоты таймера/счетчика T0 (PortD4), см. таблицу 8.4.

Таблица 8.4 – Назначение битов CS0[2:0]

CS02	CS01	CS00	Действие
0	0	0	Нет тактового источника (Timer/Counter остановлен).
0	0	1	Clk _{IO} /1 (без делителя, работа с частотой процессора).
0	1	0	Clk _{IO} /8 (делитель частоты, уменьшение в 8 раз).
0	1	1	Clk _{IO} /64 (делитель частоты, уменьшение в 64 раза).
1	0	0	Clk _{IO} /256 (делитель частоты, уменьшение в 256 раз).
1	0	1	Clk _{IO} /1024 (делитель частоты, уменьшение в 1024 раза).
1	1	0	Внешний тактовый источник на выводе T0. Тактирование по спаду.
1	1	1	Внешний тактовый источник на выводе T0. Тактирование по спаду.

TIMSK0 (TC0 Interrupt Mask Register) – регистр разрешения прерываний.

7	6	5	4	3	2	1	0
					OCIE0B	OCIE0A	TOIE0
					R/W	R/W	R/W

Биты OCIE0B(A) (2 и 1) – Output Compare B(A) Match Interrupt Enable – разрешают прерывания при совпадении значения счетного регистра с регистрами сравнения OCR0B(OCR0A). Когда бит OCIE0B(A) записан в единицу, и бит I в регистре состояния установлен, разрешается прерывание от Timer/Counter Compare Match B(A).

Бит TOIE0 – Overflow Interrupt Enable – разрешает прерывания при переполнении счетного регистра. Когда бит TOIE0 записывается в единицу и бит I в регистре состояния установлен, прерывание по Timer/Counter Overflow разрешается.

GTCCR (General Timer/Counter Control Register) – общий регистр управления таймером/счетчиком.

7	6	5	4	3	2	1	0
TSM						PSRASY	PSRSYNC
R/W						R/W	R/W

Бит TSM (7) – режим синхронизации таймера. При установке бита TSM в единицу активируется синхронный режим работы таймера, значения PSRASY и PSRSYNC сохраняются. Когда бит TSM установлен в 0, биты PSRASY и PSRSYNC очищаются аппаратно и таймеры начинают отсчет одновременно.

Бит PSRASY (1) сбрасывает предделитель таймера T2, если установлен в 1. После этого бит сбрасывается в 0 автоматически.

Бит PSRSYNC (0) сбрасывает предделитель таймера/счетчика T0 и T1, если установить в 1. После этого бит сбрасывается в 0 автоматически.

TCNT0 (TC0 Counter Value Register) – счетный регистр таймера.

7	6	5	4	3	2	1	0
TCNT0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Биты с 7 по 0 TCNT0[7:0] (TC0 Counter Value) – значение счетчика.

Регистр обеспечивает прямой доступ, как для операций чтения, так и записи, к 8-битному счетчику. Запись в регистр TCNT0 блокирует (удаляет) сравнение совпадений на следующих временных часах.

OCR0A и **OCR0B** (Output Compare Register A и B) – регистры сравнения.

7	6	5	4	3	2	1	0
OCR0A(B)[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Биты с 7 по 0 OCR0A(B)[7:0] (Output Compare 0 A(B)) регистра сравнения выхода A(B) содержат 8-битное значение, которое непрерывно сравнивается с значением счетного регистра (TCNT0). Совпадение может использоваться для генерации прерывания Output Compare.

TIFR0 (TC0 Interrupt Flag Register) – регистр флагов прерываний.

7	6	5	4	3	2	1	0
					OCF0B	OCF0A	TOV0
					R/W	R/W	R/W

Биты OCF0B(A) (2 и 1) – флаги сравнения выходного сигнала B(A). Бит OCF0B(A) устанавливается в 1, когда происходит совпадение сравнения между таймером и данными в OCR0B(A) – регистр сравнения выходного сигнала B(A). OCF0B(A) очищается аппаратным обеспечением при выполнении соответствующего вектора обработки прерываний. Когда I-битный SREG, OCIE0B(A) и OCF0B(A) установлены, выполняется прерывание сравнения таймера.

Бит TOV0 – флаг переполнения. Бит TOV устанавливается в 1 при переполнении таймера. TOV очищается аппаратно при выполнении соответствующего вектора обработки прерываний. Когда I-бит SREG, TOIE0 и TOV0 установлены, выполняется прерывание переполнения таймера.

Timer2

Имеет все те же регистры, что и Timer0. Отличие составляет регистр **ASSR** (Asynchronous Status Register) – асинхронный регистр состояния.

7	6	5	4	3	2	1	0
	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
	R/W	R/W	R	R	R	R	R

Бит EXCLK (6) разрешает использование внешнего тактового сигнала от кварцевого резонатора 32 кГц при записи в него 1.

Бит AS2 (5) управляет способом тактирования (1 – от внешнего резонатора 32 кГц, подключенного к TOSC1 (9 ножка); 0 – от внутреннего генератора CLK).

Бит TCN2UB (4) показывает, доступен ли для записи регистр **TCNT2** (1 – недоступен; 0 – доступен).

Бит OCR2AUB (3) показывает, доступен ли для записи регистр **OCR2A** (1 – недоступен; 0 – доступен).

Бит OCR2BUB (2) показывает, доступен ли для записи регистр **OCR2B** (1 – недоступен; 0 – доступен).

Бит TCR2AUB (1) показывает, доступен ли для записи регистр **TCCR2A** (1 – недоступен; 0 – доступен).

Бит TCR2BUB (0) показывает, доступен ли для записи регистр **TCCR2B** (1 – недоступен; 0 – доступен).

Также есть разница в настройке предделителя – регистр **TCCR2B**.

7	6	5	4	3	2	1	0
FOC2A	FOC2B			WGM22	CS22	CS21	CS20
W	W			R	R	R/W	R/W

Биты FOC2A (7) и FOC2B (6) регистра **TCCR2B** принудительно устанавливают значение на выводах OC2A и OC2B.

Бит WGM22 (3) управляет режимом работы выводов OC2A и OC2B при [COM2A1:COM2A0] = 01₂ в режимах ШИМ и коррекции фазы ШИМ (аналогично Timer0 см. таблицу 8.2).

Биты CS22:CS20 (2 – 0) определяют работу предделителя тактовой частоты (см. таблицу 8.5).

Таблица 8.5 – Назначение битов CS22:CS20

CS22	CS21	CS20	Действие
0	0	0	Нет тактового источника (Timer/Counter остановлен)
0	0	1	Clk/1
0	1	0	Clk/8
0	1	1	Clk/32
1	0	0	Clk/64
1	0	1	Clk/128
1	1	0	Clk/256
1	1	1	Clk/1024

Timer1

Регистр управления **TCCR1A**. (Функционально аналогичен TCCR0A, TCCR0B, различия касаются битов [WGM11:WGM10], см Таблица 8.6)

Регистр управления **TCCR1B**. (имеет отличия от TCCR0B, TCCR2B)

7	6	5	4	3	2	1	0
ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
R/W	R/W		R/W	R/W	R/W	R/W	R/W

Бит ICNC1 (7) регистра TCCR1B управляет схемой подавления помех блока захвата (0 – выключена; 1 – включена).

Бит ICES1 (6) регистра TCCR1B выбирает активный фронт регистра захвата (0 – по спадающему фронту сигнала; 1 – по нарастающему фронту сигнала).

Биты WGM13 (4) , WGM12 (3) регистра TCCR1B и биты WGM11 (1), WGM10 (0) регистра TCCR1A устанавливают режим работы таймера/счетчика T1 (Таблица 8.6).

Таблица 8.6 – Назначение битов WGM13:WGM10

WGM13:WGM10	Режим
0000	Обычный режим.
0001	Коррекция фазы PWM, 8 бит.
0010	Коррекция фазы PWM, 9 бит.
0011	Коррекция фазы PWM, 10 бит.
0100	Режим счета импульсов (OCR1A) (сброс при совпадении).
0101	PWM, 8 бит.
0110	PWM, 9 бит.
0111	PWM, 10 бит.
1000	Коррекция фазы и частоты PWM (ICR1).
1001	Коррекция фазы и частоты PWM (OCR1A).
1010	Коррекция фазы PWM (ICR1).
1011	Коррекция фазы и частоты PWM (OCR1A).
1100	Режим счета импульсов (ICR1) (сброс при совпадении).
1101	Резерв.
1110	PWM (ICR1).
1111	PWM (OCR1A).

Для настройки 16-битного таймера Timer1, помимо регистров контроля **TCCR1A(B)**, используется регистр **TCCR1C**.

7	6	5	4	3	2	1	0
FOC1A	FOC1B						
R/W	R/W						

Биты FOC1A (7) и FOC1B (6) регистра TCCR1C принудительно устанавливают значение на выводах OC1A и OC1B. Биты FOC1A / FOC1B активны, только когда биты WGM1[3:0] регистров TCCR1B, TCCR1A задают режим без ШИМ. При записи логического значения в бит FOC1A / FOC1B, прямое сопоставление принудительно сравнивается с осциллограммой. Выход OC1A / OC1B изменяется в соответствии с настройкой битов COM1x [1: 0].

TCNT1L и **TCNT1H** (TC1 Counter Value Low byte и TC1 Counter High byte) – младший байт и старший байт счетного регистра таймера TC1.

7	6	5	4	3	2	1	0
TCNT1H[15:8]							
TCNT1L[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Биты 7: 0 - TCNT1L(H) [7: 0]. Два байта регистра (TCNT1H и TCNT1L, объединенные TCNT1) предоставляют прямой доступ, для операций чтения, и записи, к таймеру.

ICR1L и **ICR1H** (Input Capture Register 1 Low byte и Input Capture Register 1 High byte) – входной регистр захвата Timer1, младший байт и старший байт).

7	6	5	4	3	2	1	0
ICR1H[15:8]							
ICR1L[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Биты ICR1L(H)[7:0] (7 – 0) – значение регистра обновляется значением счетчика (TCNT1) каждый раз, когда происходит событие на выводе ICP1 (или, опционально, на выходе аналогового компаратора для Timer1).

OCR1A(B)L и **OCR1A(B)H** (Output Compare Register 1 A(B) Low byte и Output Compare Register 1 A(B) High byte) – два двухбайтных регистра сравнения A и B, младший байт и старший байт.

7	6	5	4	3	2	1	0
OCR1A(B)H[15:8]							
OCR1A(B)L[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Биты OCR1A(B)L(H)[7:0] (7 – 0) – регистры сравнения счетчика. Сравнивают 16-разрядное значение. Непрерывно сравнивается со значением счетного регистра (TCNT0). Совпадение может использоваться для генерации прерывания Output Compare.

TIMSK1 (Timer/Counter 1 Interrupt Mask Register) – регистр маски прерываний таймера1.

7	6	5	4	3	2	1	0
		ICIE1			OCIE1B	OCIE1A	TOIE1
		R/W			R/W	R/W	R/W

Бит 5 ICIE1 (Input Capture Interrupt Enable) разрешает прерывания захвата входа. Когда этот бит записан в 1 и установлен I-флаг в регистре состояния (глобальные прерывания разрешены), прерывание захвата входа таймера/счетчика Timer1 включено.

Биты 2 и 1 OCIE1B(A) (Output Compare B(A) Match Interrupt Enable) – выходное сравнение B(A) прерывания совпадения разрешено). Когда этот бит

записан в 1 и установлен I-флаг в регистре состояния (глобальные прерывания включены), для таймера выходного сравнения В(А) прерывание совпадения включено.

Бит 0 TOIE1 (Overflow Interrupt Enable) разрешает прерывания по переполнению. Когда этот бит записан в 1 и установлен I-флаг в регистре состояния (глобальные прерывания включены), прерывание по переполнению *Timer1* включено.

TIFR1 (TC1 Interrupt Flag Register) – регистр флагов прерываний TC1.

7	6	5	4	3	2	1	0
		ICF1			OCF1B	OCF1A	TOV1
		R/W			R/W	R/W	R/W

Бит 5 ICF1 (Input Capture Flag) – флаг захвата входа. Флаг устанавливается, когда на выводе ICP1 происходит событие захвата. ICF1 автоматически очищается при выполнении вектора прерывания захвата ввода.

Биты 2 и 1 OCF1B(A) (Output Compare B(A) Match Flag) – флаг сравнения выходных данных B(A). Этот флаг устанавливается в тактовом цикле таймера после того, как значение счетчика (TCNT1) соответствует выходному регистру сравнения B(A) (OCR1B(A)). OCF1B(A) автоматически очищается, когда выполняется вектор прерывания совпадения выходного совпадения B(A).

Бит 0 TOV1 (Overflow Flag) – флаг переполнения. Установка этого флага зависит от установки битов WGM1[3:0] регистров TCCR1B, TCCR1A. В режимах Normal и CTC этот флаг устанавливается при переполнении таймера. TOV1 автоматически очищается при выполнении вектора прерывания переполнения *Timer1*.

Сводная таблица описанных регистров представлена в таблице 8.7.

Таблица 8.7 – Сводная таблица регистров таймеров

[illegible]

0x8A		OCR1BL	OCR1B[7:0]							
0x8B		OCR1BH	OCR1B[15:8]							
0xB0		TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0			WGM21	WGM20
0xB1		TCCR2B	FOC2A	FOC2B			WGM22	CS22	CS21	CS20
0xB2		TCNT2	TCNT2[7:0]							
0xB3		OCR2A	OCR2A[7:0]							
0xB4		OCR2B	OCR2B[7:0]							
0xB6		ASSR		EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB

Для работы с регистрами TIFR0, TIFR1, TIFR2, GTCCR, TCCR0A, TCCR0B, TCNT0, OCR0A, OCR0B, используются команды IN/OUT. Они расположены в адресном пространстве регистров ввода/вывода.

Для работы с регистрами TIMSK0, TIMSK1, TIMSK2, TCCR1A, TCCR1B, TCCR1C, TCNT1L, TCNT1H, ICR1L, ICR1H, OCR1AL, OCR1AH, OCR1BL, OCR1BH, TCCR2A, TCCR2B, TCNT2, OCR2A, OCR2B, ASSR используются команды sts, lds. Они расположены в адресном пространстве дополнительных регистров ввода/вывода.

Сводная таблица векторов прерываний таймеров микроконтроллера ATmega328P представлена в таблице 8.8.

Таблица 8.8 – Прерывания таймеров AVR ATmega328P

№	Адрес вектора	Имя вектора	Событие вызывающее прерывание	Имя вектора в процедуре ISR()
8	0x000E	TIMER2_COMPA	Совпадение А таймера/счетчика T2	TIMER2_COMPA_vect
9	0x0010	TIMER2_COMPB	Совпадение В таймера/счетчика T2	TIMER2_COMPB_vect
10	0x0012	TIMER2_OVF	Переполнение таймера/счетчика T2	TIMER2_OVF_vect
11	0x0014	TIMER1_CAPT	Захват таймера/счетчика T1	TIMER1_CAPT_vect
12	0x0016	TIMER1_COMPA	Совпадение А таймера/счетчика T1	TIMER1_COMPA_vect
13	0x0018	TIMER1_COMPB	Совпадение В таймера/счетчика T1	TIMER1_COMPB_vect
14	0x001A	TIMER1_OVF	Переполнение таймера/счетчика T1	TIMER1_OVF_vect
15	0x001C	TIMER0_COMPA	Совпадение А таймера/счетчика T0	TIMER0_COMPA_vect
16	0x001E	TIMER0_COMPB	Совпадение В таймера/счетчика T0	TIMER0_COMPB_vect
17	0x0020	TIMER0_OVF	Переполнение таймера/счетчика T0	TIMER0_OVF_vect

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Представлена программа, которая обеспечивает изменение состояния диода на противоположное при переполнении таймера TC1. Ниже представлены программы на языках Assembler и C. Комментарии к коду представлены в тексте программы.

Выполните анализ представленных программ, загрузите каждую из них в микроконтроллер, проанализируйте ее работу. Просчитайте точное время изменения состояния светодиода (частота тактирования контроллера $F_{clk} - 16\text{МГц}$, делитель частоты – $F_{clk} / 64$, разрядность счетчика 16 бит – 65536 отсчетов до переполнения).

Код на языке Assembler

```
1  .include "m328Pdef.inc" ;библиотека с логическими именами портов ввода/вывода
2  .def tmp = r16          ;задание логического имени tmp - регистру r16
3  .def msk = r17          ;задание логического имени msk - регистру r17
4  .org 0                  ;инициализация вектора reset
5      jmp Reset           ;
6  .org 0x001A             ;инициализация вектора переполнения TC1
7      jmp Timer1OVF       ;
8  Reset:                 ;метка вектора reset
9      ldi tmp, 0x20        ;настройка PORTB5 на вывод информации
10     out DDRB, tmp        ;
11     ldi tmp, 0x00        ;обычный режим без использования выводов OC1A и OC1B
12     sts TCCR1A,tmp       ;
13     ldi tmp, 0x03        ;запуск таймера, предделитель clk/64
14     sts TCCR1B,tmp       ;
15     ldi tmp, 0x01        ;разрешение прерываний по переполнению счётчика TC1
16     sts TIMSK1,tmp       ;
17     ldi msk,0x20         ;инициализация msk значением 0x20
18     sei                  ;глобальное разрешение прерываний
19 p:                      ;метка для организации бесконечного цикла
20     jmp p                ;безусловный переход к метке p
21
22 Timer1OVF:              ;метка вектора переполнения TC1
23     in tmp,PORTB         ;загрузка текущего состояния вывода PORTB
24     eor tmp, msk         ;инверсия значения PORTB5 с помощью маски 0x20
25     out PORTB,tmp       ;вывод нового значения в PORTB5
26     reti                ;возврат из обработчика прерываний
```

Код на языке C

```
1  #include <avr/io.h>      //библиотека с логическими именами портов ввода/вывода
2  #include <avr/interrupt.h> //библиотека работы с прерываниями
3  ISR (TIMER1_OVF_vect)    //обработчик прерываний по вектору переполнения Timer1
4  {
5      PORTB ^= 0x20;        //смена состояния вывода PORTB5 на противоположное
6      (светодиод)
7  }
8  int main(void)           //главная процедура, точка входа в программу
9  {
10     DDRB = 0x20;          //настройка PORTB5 на вывод информации
11     TCCR1A = 0x00;        //обычный режим без использования выводов OC1A и OC1B
12     TCCR1B = (1<<CS11)|(1<<CS10); //запуск таймера, предделитель clk/64
13     TIMSK1 = (1<<TOIE1);  //разрешение прерываний по переполнению счётчика TC1
14     sei();                //глобальное разрешение прерываний
15     while (1)             //бесконечный цикл
16     {
17     }
18 }
```

Задание 2. Измените программу из примера 1 таким образом, чтобы светодиод менял состояние каждую секунду. Для выполнения используйте TC1 в режиме CTC. Чтобы задать точное время – подберите такое количество тактов, при котором будет отсчитан необходимый интервал времени. Для этого измените значение регистра OCR1A (или OCR1B), выберите и разрешите соответствующее прерывание, опишите обработчик прерывания. При необходимости сконфигурируйте значение делителя тактовой частоты.

Соберите проект, загрузите программу в контроллер. Проанализируйте работу устройства.

Задание 3. Измените программу из задания 2 так, чтобы вывод PORTB0 менял своё состояние на противоположное через интервал времени, заданный по варианту (см. таблицу 8.9)

Таблица 8.9 – варианты для задания 3

Вариант №	1	2	3	4	5	6	7	8	9	10
Время, мс	20	0.5	0.25	0.75	2	0.9	10	12	5	0.01

Соберите проект, загрузите программу в контроллер. Проанализируйте работу устройства с помощью логического анализатора. Зафиксируйте в отчёте полученное реальное значение временной задержки.

Задание 4. Измените программу из задания 2 таким образом, чтобы меняли состояния все три светодиода, подключенные к плате. За изменение отсчёт времени для каждого светодиода должен отвечать свой таймер.

Соберите проект, загрузите программу в контроллер. Проанализируйте работу устройства.

Светодиод	RX	TX	L
Интервал, мс	1500	600	2000

Задание 5. Измените программу из задания 4 таким образом, чтобы светодиоды меняли состояния через заданные интервалы времени с использованием одного таймера.

Соберите проект, загрузите программу в контроллер. Проанализируйте работу устройства.

Задание 6. Используя программы, полученные в работах 7 и 8 разработайте проект, в котором АЦП преобразует значения через заданные интервалы времени (с заданной частотой дискретизации) по таймеру и выводит среднее из накопленных за секунду по UART.

Вопросы для самоконтроля

1. Что такое таймер?
2. Зачем в таймерах используется делитель частоты?
3. В каких режимах могут работать таймеры Atmega328p?
4. Зачем может быть использован режим захвата?
5. Что такое ШИМ?

6. Какие действия необходимо выполнить, чтобы настроить таймер TC1 в режиме ШИМ?
7. Поясните различия между режимами ШИМ.
8. Определите максимальный период, через который таймер TC0 может формировать прерывание, при частоте тактирования микроконтроллера 16МГц.
9. Какие события могут вызывать прерывания от таймеров.
10. Определите время, через которое таймер TC1 в режиме CTC будет сброшен в состояние 0, если модуль счета равен 500, предделитель стоит в режиме CLK/256, частота тактирования 16МГц.