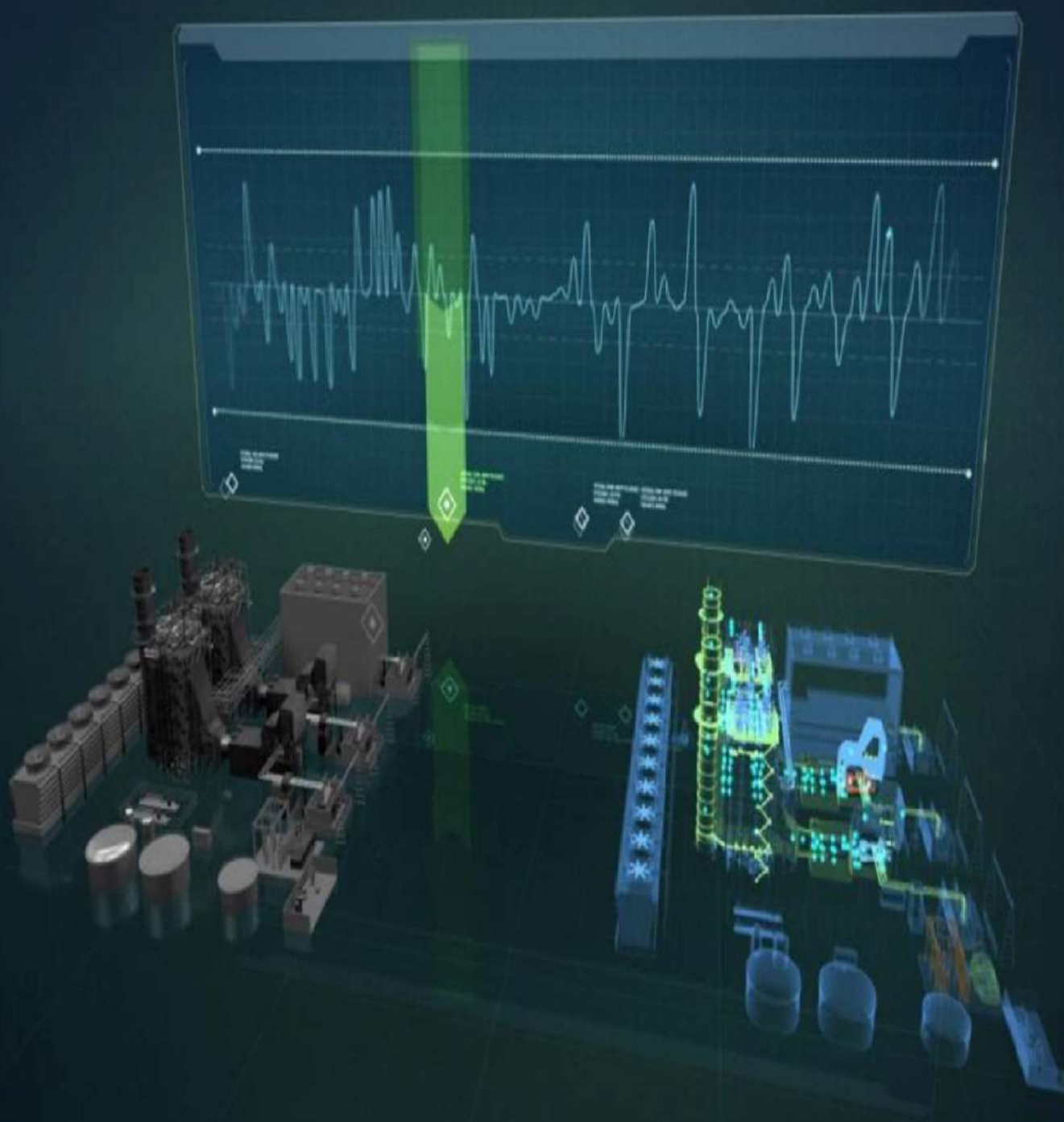


А. В. Доманов

# ЭЛЕМЕНТЫ СИСТЕМ ЭЛЕКТРОПРИВОДА (микроконтроллеры)

Методические указания



**А. В. Доманов**

# **ЭЛЕМЕНТЫ СИСТЕМ ЭЛЕКТРОПРИВОДА (микроконтроллеры)**

Методические указания

Ульяновск  
2003

УДК 621.382.2 / 3 (076)

ББК 31.291я7

Э45

Рецензент канд. техн. наук Кузнецов А. В.

Одобрено секцией методических пособий научно-методического  
совета УлГТУ

**Э45 Элементы систем электропривода (микроконтроллеры): Методические указания по курсу для студентов направления 654500 «Электротехника, электромеханика и электротехнология» / Сост. А. В. Доманов. – Ульяновск: УлГТУ, 2003. 38 с.**

Указания составлены в соответствии с программой курса «Элементы систем электропривода» для студентов направления 654500 всех форм обучения и содержат информационные материалы по структуре, работе и программированию микроконтроллеров серии Atmel AVR.

Работа подготовлена на кафедре «Электропривод и автоматизация промышленных установок».

УДК 621.382.2 / 3 (076)

ББК 31.291я7

**Учебное издание**

**ЭЛЕМЕНТЫ СИСТЕМ ЭЛЕКТРОПРИВОДА**

(микроконтроллеры)

Методические указания

Составитель ДОМАНОВ Андрей Викторович

Редактор Н. А. Евдокимова

Подписано в печать 30.11.2003. Формат 60х84/16. Бумага писчая.

Печать трафаретная. Усл.печ.л. 2,40. Уч.-изд.л. 2,0.

Тираж 100 экз. Заказ

Ульяновский государственный технический университет

432027, г. Ульяновск, Сев. Венец, 32.

Типография УлГТУ. 432027, г. Ульяновск, Сев. Венец, 32.

## Оглавление

Общие сведения	4
Регистры общего назначения	6
Обработка прерываний	9
Регистры прерываний	10
Время срабатывания прерываний	13
Аналого-цифровой преобразователь (АЦП)	14
Работа АЦП	14
Масштабирование АЦП	15
Функция шумоподавления АЦП	15
Сканирование нескольких каналов АЦП	17
Таймеры-счетчики	18
Таймер-счетчик T/C0	18
Таймер-счетчик T/C1	19
Таймер-счетчик T/C1 в ШИМ режиме	25
Таймер-счетчик T/C2	25
Таймер-счетчик T/C2 в ШИМ режиме	27
Интегрированная отладочная среда AVR Studio фирмы Atmel	28
Примеры программ	31
Список команд	35
Библиографический список	40

## Общие сведения

Во многих современных системах управления используются дискретные элементы и цифровые процессоры. Основные преимущества цифровых систем следующие: повышенная чувствительность, большая надежность, отсутствие дрейфа, более высокая устойчивость к помехам и возмущениям, меньшие габаритные размеры и масса, меньшая стоимость, удобство в программировании. Однако наиболее существенным преимуществом цифровой системы управления является ее гибкость по сравнению с аналоговыми регуляторами. Программа цифрового регулятора может быть изменена (скорректирована) или адаптирована к характеристикам объекта управления без каких-либо изменений в аппаратной схеме устройства.

Компания ATMEL один из мировых лидеров в производстве широкого спектра микросхем энергонезависимой памяти, FLASH-микроконтроллеров и микросхем программируемой логики - взяла старт по разработке RISC-микроконтроллеров в середине 90-х годов, используя все свои технические решения, накопленные к этому времени.

AVR-архитектура, на основе которой построены микроконтроллеры семейства AT90S, объединяет мощный гарвардский RISC-процессор с отдельным доступом к памяти программ и данных, 32 регистра общего назначения, каждый из которых может работать как регистр-аккумулятор, и развитую систему команд фиксированной длиной – 16 бит. Большинство команд выполняется за один машинный такт с одновременным исполнением текущей и выборкой следующей команды, что обеспечивает производительность до 1 MIPS на каждый мегагерц тактовой частоты.

32 регистра общего назначения образуют регистровый файл быстрого доступа, где каждый регистр напрямую связан с арифметическо-логическим устройством (АЛУ). За один такт из регистрового файла выбираются два операнда, выполняется операция, и результат возвращается в регистровый файл. АЛУ поддерживает арифметические и логические операции с регистрами, между регистром и константой или непосредственно с регистром.

Регистровый файл также доступен как часть памяти данных. Шесть из 32-х регистров могут использоваться как три 16-разрядных регистра-указателя для косвенной адресации. Старшие микроконтроллеры семейства AVR имеют в составе АЛУ аппаратный умножитель.

Базовый набор команд AVR содержит 120 инструкций. Инструкции битовых операций включают инструкции установки, очистки и тестирования битов.

Все микроконтроллеры AVR имеют встроенную FLASH ROM с возможностью внутрисхемного программирования через последовательный 4-проводный интерфейс.

Периферия МК AVR включает: таймеры-счетчики, широтно-импульсные модуляторы, поддержку внешних прерываний, аналоговые компараторы, 10-разрядный 8-канальный АЦП, параллельные порты (от 3 до 48 линий ввода и вывода), интерфейсы UART и SPI, сторожевой таймер и устройство сброса по включению питания. Все эти качества превращают AVR-микроконтроллеры в мощный инструмент для построения современных, высокопроизводительных и экономичных контроллеров различного назначения.

В рамках единой базовой архитектуры AVR-микроконтроллеры подразделяются на три подсемейства [1]:

- *Classic AVR* — основная линия микроконтроллеров с производительностью отдельных модификаций до 16 MIPS, FLASH ROM программ 2–8 Кбайт, EEPROM данных 64–512 байт, SRAM 128–512 байт;
- *mega AVR* с производительностью 4–16 MIPS для сложных приложений, требующих большого объема памяти, FLASH ROM программ 64–128 Кбайт, EEPROM данных 64–512 байт, SRAM 2–4 Кбайт, встроенный 10-разрядный 8-канальный АЦП, аппаратный умножитель;
- *tiny AVR* — низкостоимостные микроконтроллеры в 8-выводном исполнении имеют встроенную схему контроля напряжения питания, что позволяет обойтись без внешних супервизорных микросхем.

AVR-микроконтроллеры поддерживают спящий режим и режим микропотребления. В спящем режиме останавливается центральное процессорное ядро, в то время как регистры, таймеры-счетчики, сторожевой таймер и система прерываний продолжают функционировать. В режиме микропотребления сохраняется содержимое всех регистров, останавливается тактовый генератор, запрещаются все функции микроконтроллера, пока не поступит сигнал внешнего прерывания или аппаратного сброса. В зависимости от модели, AVR-микроконтроллеры работают в диапазоне напряжений 2,7–6 В либо 4–6 В (исключение составляет ATtiny12V с напряжением питания 1,2 В).

ATMEL предлагает программную среду AVR-studio для отладки программ в режиме симуляции на программном отладчике, а также для работы непосредственно с внутрисхемным эмулятором. AVR-studio доступен с WEB-страницы ATMEL, содержит ассемблер и предназначен для работы с эмуляторами ICEPRO и MegaICE. Ряд компаний предлагают свои версии Си-компиляторов, ассемблеров, линковщиков и загрузчиков для работы с микроконтроллерами семейства AVR. Микроконтроллеры ATMEL широко

применяются в России и программируются многими отечественными программаторами. Ряд российских фирм предлагает также различные аппаратные средства отладки AVR-микроконтроллеров.

### Регистры общего назначения

Все инструкции операций с регистрами в системе команд имеют прямой доступ ко всем регистрам в течение одного такта. Единственное исключение – пять арифметических и логических команд SBCI, SUBI, CPI, ANDI и ORI между константой и регистром и LDI команда для загрузки непосредственных данных. Эти команды обращаются ко второй половине регистров в области регистров – R16 ... R31. Общие инструкции SBC, SUB, CP, AND и OR и все другие операции между двумя регистрами или с одним регистром применяются ко всем регистрам.

Каждому регистру также назначен адрес памяти данных, отображающей их непосредственно в первых 32-х адресах памяти пользователя.

#### *X-регистр, Y-регистр и Z-регистр*

Регистры R26 ... R31 имеют некоторые добавочные функции для их универсального использования. Эти регистры – указатели адреса для косвенной адресации SRAM. В различных способах адресации эти регистры адреса имеют функции фиксированного смещения, автоматического приращения и декрементации.

#### *Регистр Состояния - SREG*

Бит	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

#### • Бит 7 – I: Глобальное разрешение прерываний

Бит глобального разрешения прерываний должен быть установлен в состояние 1 для разрешения использования прерываний. Разрешение отдельных прерываний выполняется в регистрах маски прерываний – GIMSK и TIMSK. Если бит глобального разрешения прерываний обнулен, ни одно из прерываний не будет выполняться независимо от значений GIMSK и TIMSK. I-бит обнуляется аппаратными средствами после того, как произошло прерывание, и устанавливается командой RETI, чтобы дать возможность исполнения последующих прерываний.

#### • Бит 6 – T: Бит хранения копии

Команды копирования битов BLD (загрузка бита) и BST (хранение бита) используют T бит как источник и адресат для используемого бита. С помощью

команды BST бит может быть скопирован из регистра в области регистров в бит T, и бит T может быть скопирован в бит регистра в области регистров с помощью команды BLD.

- Бит 5 – H: Флаг половинной загрузки

H указывает половинную загрузку в некоторых арифметических операциях.

- Бит 4 – S: Знаковый бит

S-бит всегда является «исключающим или» между флагом отрицательности N и флагом двоичного переполнения V.

- Бит 3 – V: Флаг двоичного переполнения

Флаг двоичного переполнения V поддерживается двоично-дополнительной арифметикой.

- Бит 2 – N: Флаг отрицательности

N указывает на отрицательный результат после различных арифметических и логических операций.

- Бит 1 – Z: Флаг нуля

Z указывает на нулевой результат после различных арифметических и логических операций.

- Бит 0 – C: Флаг переноса

C указывает на перенос в арифметической или логической операции.

*Указатель стека – SP*

Общий AVR 16-разрядный указатель стека состоит из двух 8-разрядных регистров, расположенных в областях ввода-вывода. Указатель стека указывает на область стека данных SRAM, где расположены стеки подпрограмм и прерываний. Эта область стека в данных SRAM должна быть определена программой прежде, чем будут выполнены любые вызовы подпрограмм или прерываний. Указатель стека уменьшается на 1, когда данные помещаются в стек командой PUSH, и уменьшается на 2, когда данные помещаются в стек с вызовом подпрограммы или прерыванием. Указатель стека увеличивается на 1, когда данные забираются из стека командой POP, и увеличивается на 2, когда данные забираются из стека с возвратом из подпрограммы RET или возвратом из прерывания IRET.

*Регистр управления MCU - MCUCR*

Бит	7	6	5	4	3	2	1	0
	-	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R	R	R
Начальное значение	0	0	0	0	0	0	0	0



- Бит 7 – Зарезервирован
- Бит 6 – SE: Разрешение режима «сна»

SE бит должен быть установлен в состояние 1, чтобы при выполнении инструкции SLEEP MCU перешел в режим «сна». Чтобы избежать перехода MCU в режим "сна" раньше, чем это потребуется программисту, рекомендуется устанавливать бит SE непосредственно перед выполнением команды SLEEP.

- Биты 5,4 – SM1/SM0: выбор типов режима «сна»

Этот бит определяет выбор типа режима «сна» из трех доступных, как показано в следующей таблице:

SM1	SM0	Режим «сна»
0	0	Режим бездействия
0	1	Зарезервировано
1	0	Пониженное питание
1	1	Энергосбережение

- Биты 3,2 – ICS11/ICS10: Управление заданием внешнего прерывания 1

ICS11	ICS10	
0	0	Низкий уровень INT1
0	1	Зарезервировано
1	0	Фронт↓ INT1
1	1	Фронт↑ INT1

- Биты 1,0 – ICS01/ICS00: Управление заданием внешнего прерывания 1

ICS01	ICS00	
0	0	Низкий уровень INT0
0	1	Зарезервировано
1	0	Фронт↓ INT0
1	1	Фронт↑ INT0

#### Регистр состояния MCU - MCUSR

Бит	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	EXTRF	PORF
Чтение/запись	R	R	R	R	R	R	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Биты 7...2 – Зарезервированы
- Бит 1 – EXTRF: Флаг внешнего сброса

После сброса при включении этот бит не определен (X). Он будет установлен внешним сбросом. Сброс с помощью сторожевого таймера оставит этот бит неизменным.

- Бит 0 – PORF: Флаг сброса при включении

Этот бит устанавливается сбросом при включении. Сброс с помощью сторожевого таймера или внешний сброс оставит этот бит неизменным.

Чтобы подвести итог, следующая таблица показывает значение этих двух битов после трех видов сброса:

Источник сброса	PORF	EXTRF
Включение	1	Не меняется
Внешний	Не меняется	1
Сторожевой	Не меняется	Не меняется

Чтобы использовать эти биты для идентификации условия сброса, программное обеспечение пользователя должно как можно раньше в программе обнулить PORF и EXTRF биты. Проверка значений PORF И EXTRF выполняется перед обнулением битов. Если бит очищен до внешнего сброса или сброса с помощью сторожевого таймера, то источник сброса может быть определен с помощью следующей таблицы истинности:

Источник сброса	PORF	EXTRF
Включение	1	1
	1	0
Внешний	0	1
Сторожевой	0	0

### Обработка прерываний

МК имеет два отдельных 8-разрядных регистра управления маской прерываний; GIMSK – общий регистр маски прерываний и TIMSK – регистр маски прерываний таймера/счетчика. Кроме того, в регистрах управления периферией могут находиться другие биты разрешения и маски.

Когда происходит прерывание, I-бит глобального разрешения прерываний обнуляется и все прерывания запрещаются. Программное обеспечение пользователя может устанавливать I-бит внутри подпрограммы прерывания, чтобы разрешить вложенные прерывания. Команда RETI заканчивает подпрограмму прерывания и устанавливает I-бит в состояние 1.

Когда счетчик программ векторизован к фактическому вектору прерывания, чтобы выполнить прерывание, аппаратные средства очищают соответствующий флаг, который вызвал прерывание. Некоторые флаги прерываний могут также быть обнулены с помощью записи логической единицы в позицию обнуляемого флага [2].

## Регистры прерываний

### Общий регистр маски прерываний - GIMSK

Бит	7	6	5	4	3	2	1	0
	INT1	INT0	-	-	-	-	-	-
Чтение/запись	R/W	R/W	R	R	R	R	R	R
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – INT1: Разрешение внешнего прерывания 1
- Бит 6 – INT0: Разрешение внешнего прерывания 0
- Бит 5...0 – Зарезервированы

### Общий регистр флагов прерываний - GIFR

Бит	7	6	5	4	3	2	1	0
	INTF1	INTF0	-	-	-	-	-	-
Чтение/запись	R/W	R/W	R	R	R	R	R	R
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – INTF1: Флаг внешнего прерывания 1
- Бит 6 – INTF0: Флаг внешнего прерывания 0
- Бит 5...0 – Зарезервированы

### Регистр маски прерывания таймера/счетчика - TIMSK

Бит	7	6	5	4	3	2	1	0
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – OCIE2: Разрешение прерывания сравнения вывода таймера/счетчика2

Когда OCIE2 бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание сравнения таймера/счетчика2. Соответствующее прерывание (вектор \$003) выполняется если происходит совпадение при сравнении таймера/счетчика2. Флаг сравнения в таймере/счетчике2 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

- Бит 6 – TOIE2: Разрешение прерывания переполнения таймера/счетчика2

Когда TOIE2 бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание переполнения таймера/счетчика2. Соответствующее прерывание (вектор \$004) выполняется, если происходит переполнение в таймере/счетчике2. Флаг переполнения

таймера/счетчика2 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

- Бит 5 – TICIE1: Разрешение прерывания захвата входа таймера/счетчика1

Когда TICIE1 бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание захвата входа таймера/счетчика1. Соответствующее прерывание (вектор \$005) выполняется, если вызывающее захват данных событие происходит на выводе ICP. Флаг захвата входа в таймере/счетчике1 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

- Бит 4 – OCIE1A: Разрешение прерывания сравнения A выхода таймера/счетчика1

Когда OCIE1A бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание сравнения A выхода таймера/счетчика1. Соответствующее прерывание (вектор \$006) выполняется, если происходит соответствие при сравнении A в таймере/счетчике1. Флаг сравнения A в таймере/счетчике1 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

- Бит 3 – OCIE1B: Разрешение прерывания сравнения B выхода таймера/счетчика1

Когда OCIE1B бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание сравнения B выхода таймера/счетчика1. Соответствующее прерывание (вектор \$007) выполняется если, происходит соответствие при сравнении B в таймере/счетчике1. Флаг сравнения B в таймере/счетчике1 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

- Бит 2 – TOIE1: Разрешение прерывания переполнения таймера/счетчика1

Когда TOIE1 бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешается прерывание переполнения таймера/счетчика1. Соответствующее прерывание (вектор \$008) выполняется, если происходит переполнение в таймере/счетчике1. Флаг переполнения таймера/счетчика1 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR. Когда таймер/счетчик1 находится в PWM режиме, флаг переполнения таймера установлен, когда счетчик меняет направление счета в \$0000.

- Бит 1 – Зарезервирован

- Бит 0 – TOIE0: Разрешение прерывания переполнения таймера/счетчика0

Когда TOIE0 бит установлен в состояние 1 и I-бит в регистре состояния также установлен в состояние 1, разрешено прерывание переполнения таймера/счетчика0. Соответствующее прерывание (вектор \$009) выполняется, если происходит переполнение в таймере/счетчике0. Флаг переполнения

таймера/счетчика0 установлен в состояние 1 в регистре флагов прерываний таймера/счетчика - TIFR.

*Регистр флагов прерываний таймера/счетчика - TIFR*

Бит	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Начальное значение	0	0	0	0	0	0	0	0

• Бит 7 – OCF2: Флаг 2 сравнения выхода

OCF2 бит установлен в состояние 1, когда происходит соответствие при сравнении таймера/счетчика2 и данными в OCR2 - регистре 2 сравнения выхода. OCF2 обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: OCF2 обнуляется путем записи в него логической единицы. Когда I-бит в SREG, OCIE2 (разрешение прерывания сравнения таймера/счетчика2) и OCF2 установлены в состояние 1, выполняется прерывание сравнения выхода таймера/счетчика2.

• Бит 6 – TOV2: Флаг переполнения таймера/счетчика2

TOV2 бит установлен в состояние 1, когда происходит переполнение в таймере/счетчике2. TOV2 обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: TOV2 обнуляется путем записи в него логической единицы. Когда I-бит в SREG, TOIE2 (разрешение прерывания переполнения таймера/счетчика2) и TOV2 установлены в состояние 1, выполняется прерывание переполнения таймера/счетчика2. В PWM режиме этот бит установлен, когда таймер/счетчик изменяет направление счета в \$ 00.

• Бит 5 – ICF1: Флаг захвата входа 1

ICF1 бит устанавливается в состояние 1, чтобы пометить событие захвата входа, указывает, что значение таймера/счетчика1 было перемещено во входной регистр сбора данных - ICR1. ICF1 обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: ICF1 обнуляется путем записи в него логической единицы.

• Бит 4 – OCF1A: Флаг сравнения выхода 1A

OCF1A бит устанавливается в состояние 1, когда происходит соответствие при сравнении таймера/счетчика1 и данными в OCR1A - регистре сравнения выхода 1A. OCF1A обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: OCF1A обнуляется путем записи в него логической единицы. Когда I-бит в SREG, OCIE1A (разрешение прерывания сравнения таймера/счетчика1) и OCF1A

установлены в состояние 1, выполняется прерывание сравнения таймера/счетчика1.

- Бит 3 – OCF1B: Флаг сравнения выхода 1B

OCF1B бит устанавливается в состояние 1, когда происходит соответствие при сравнении таймера/счетчика1 и данными в OCR1B - регистре сравнения выхода 1B. OCF1B обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: OCF1B обнуляется путем записи в него логической единицы. Когда I-бит в SREG, OCIE1B (разрешение прерывания сравнения В таймера/счетчика1) и OCF1B установлены в состояние 1, выполняется прерывание сравнения таймера/счетчика1.

- Бит 2 – TOV1: Флаг переполнения таймера/счетчика1

TOV1 бит установлен в состояние 1, когда происходит переполнение в таймере/счетчике1. TOV1 обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: TOV1 обнуляется путем записи в него логической единицы. Когда I-бит в SREG, TOIE1 (разрешение прерывания переполнения таймера/счетчика1) и TOV1 установлены в состояние 1, выполняется прерывание переполнения таймера/счетчика1. В PWM режиме этот бит установлен, когда таймер/счетчик изменяет направление счета в \$0000.

- Бит 1 – Зарезервирован

- Бит 0 – TOV0: Флаг переполнения таймера/счетчика0

TOV0 бит установлен в состояние 1, когда происходит переполнение в таймере/счетчике0. TOV0 обнуляется аппаратными средствами при выполнении соответствующего вектора прерывания. В качестве альтернативы: TOV0 обнуляется путем записи в него логической единицы. Когда I-бит в SREG, TOIE0 (разрешение прерывания переполнения таймера/счетчика0) и TOV0 установлены в состояние 1, выполняется прерывание переполнения таймера/счетчика0. В PWM режиме этот бит установлен, когда таймер/счетчик изменяет направление счета в \$00.

### **Время срабатывания прерываний**

Время отклика выполнения прерывания для всех AVR прерываний не менее четырех тактов. Через четыре такта после установки флага прерывания выполняется переход по вектору прерывания. В течение этих четырех тактов счетчик программ (2 байта) помещается в стек, и указатель стека уменьшается на 2. Вектор – это переход к подпрограмме прерывания, и этот переход занимает 3 такта. Если прерывание происходит в течение выполнения многотактовой команды, эта команда завершается прежде, чем прерывание обслуживается.

Возврат из прерывания (то же самое касается подпрограммы вызова подпрограммы) занимает четыре такта. В течение этих четырех тактов счетчик программ (2 байта) выталкивается обратно из стека, и указатель стека увеличивается на 2. Когда AVR выходит из прерывания, он всегда возвращается к основной программе и выполняет еще одну команду прежде, чем любое отложенное прерывание обслуживается.

Обратите внимание, что регистр состояния (SREG) не обрабатывается аппаратными средствами AVR ни для прерываний, ни для подпрограмм. Для прерывания, требующего хранения SREG, это должно выполняться программным обеспечением пользователя.

Для прерываний, вызванных событиями, которые могут оставаться статическими, флаг прерывания устанавливается, когда происходит событие. Если флаг прерывания обнулен и условие прерывания сохраняется, флаг не будет установлен, пока событие не произойдет в следующий раз.

### **Аналого-цифровой преобразователь (АЦП)**

Характеристики:

- 10-разрядная разрешающая способность
- 70 - 280 мкс время преобразования
- До восьми мультиплексированных входных каналов
- Режимы преобразования: постоянное или одиночное
- Прерывание при окончании преобразования
- Режим шумоподавления

Микроконтроллер имеет 10-разрядный АЦП. АЦП соединен с аналоговым мультиплексором с восемью каналами, который позволяет каждый вывод порта А использовать как вход АЦП. АЦП содержит специальный усилитель, который гарантирует, что входное напряжение АЦП удерживается на постоянном уровне в течение преобразования.

### **Работа АЦП**

АЦП может работать в двух режимах – одиночного и постоянного преобразования. В режиме одиночного преобразования каждое преобразование должно быть запущено пользователем. В режиме постоянного преобразования АЦП постоянно производит преобразование и модифицирует регистр данных АЦП. ADFR бит в ADCSR позволяет сделать выбор между двумя доступными режимами.

АЦП включается путем записи логической единицы в бит разрешения АЦП - ADEN в ADCSR. Первому преобразованию, которое будет начато после разрешения АЦП, будет предшествовать фиктивное преобразование, чтобы инициализировать АЦП. Единственное различие для пользователя будет в том,

что это преобразование будет занимать на 12 тактов генератора АЦП больше обычного.

Преобразование начинается путем записи логической единицы в бит начала преобразования АЦП - ADSC. Этот бит останется единицей в течение преобразования и обнуляется микроконтроллером при завершении преобразования.

Если во время преобразования был выбран другой канал данных, то АЦП закончит текущее преобразование перед изменением канала.

Так как АЦП генерирует 10-разрядный результат, то чтобы получить результат преобразования, должны читаться два регистра данных – ADCH и ADCL. При чтении данных сначала должен читаться ADCL. После считывания ADCL доступ к регистрам данных блокируется. Это означает, что если считан ADCL и преобразование завершается до чтения ADCH, ни один из регистров не модифицируется, и результат преобразования сохраняется. Когда считывается ADCH, доступ к регистрам ADCL и ADCH разрешается.

АЦП имеет собственное прерывание - ADIF, которое может быть вызвано при завершении преобразования. Когда доступ к регистрам данных запрещен между чтением ADCL и ADCH, прерывание будет вызываться, даже если это приведет к потере результата преобразования.

### **Масштабирование АЦП**

АЦП содержит делитель частоты, который преобразует частоту генератора МК к частоте, приемлемой для АЦП. АЦП допускает входные частоты в диапазоне 50-200 кГц. Более высокая частота может подаваться, но это приведет к снижению точности преобразования.

Для управления делителем частоты используются биты ADPS0 – ADPS2 в регистре ADCSR.

В режиме постоянного преобразования преобразование длится 13 тактов, это означает, что диапазон времени преобразования – 70-280 мкс. В режиме одиночного преобразования время преобразования – 14 тактов.

### **Функция шумоподавления АЦП**

АЦП имеет режим шумоподавления, который дает возможность проводить преобразование в течение неактивного периода, что уменьшает шум, вызванный ядром CPU. Чтобы использовать эту возможность, должна использоваться следующая процедура:

1. Выключите АЦП путем обнуления ADEN.
2. Включите АЦП и одновременно начните преобразование, установив ADEN и ADSC. Это начнет фиктивное преобразование, после которого начнется нормальное преобразование.



3. В течение 14 ADC тактов установите неактивный режим.

4. Если никакие другие прерывания не произойдут прежде, чем завершится преобразование, прерывание АЦП «разбудит» MCU и выполнит подпрограмму обработки прерывания окончания преобразования АЦП.

*Регистр выбора канала мультимплексора АЦП - ADMUX*

Бит	7	6	5	4	3	2	1	0
	-	-	-	-	-	MUX2	MUX1	MUX0
Чтение/запись	R	R	R	R	R	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Биты 7 ... 3 – Зарезервированные биты

Эти биты зарезервированы и всегда равны нулю.

- Биты 2 ... 0 – MUX2 ... MUX0: Биты выбора аналогового канала 2-0

Значение этих трех битов определяет, какой аналоговый вход 7-0 соединен с АЦП.

*Регистр управления и состояния АЦП - ADCSR*

Бит	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – ADEN: Разрешение АЦП

Запись логической единицы в этот бит разрешает использование АЦП. АЦП выключается путем обнуления этого бита. Выключение АЦП во время преобразования прервет это преобразование.

- Бит 6 – ADSC: Запуск преобразования АЦП

В режиме одиночного преобразования в этот бит должна быть записана логическая единица для того, чтобы начать одно преобразование. В режиме постоянного преобразования в этот бит должна быть записана логическая единица для того, чтобы начать первое преобразование. После первой записи ADSC после разрешения АЦП или если ADSC был записан в то же время, когда был разрешен АЦП, фиктивное преобразование будет предшествовать запуску. Это фиктивное преобразование выполняет инициализацию АЦП. ADSC будет равен 1 в течение преобразования. Когда преобразование будет закончено, он перейдет в 0. Когда фиктивное преобразование предшествует реальному преобразованию, ADSC останется единицей, пока не завершается реальное преобразование.

- Бит 5 – ADFR: Выбор режима свободного хода АЦП

Когда этот бит установлен в состояние 1, АЦП функционирует в режиме постоянного преобразования. В этом режиме АЦП производит выборку и

модифицирует регистры данных непрерывно. Обнуление этого бита завершит режим постоянного преобразования.

- Бит 4 – ADIF: Флаг прерывания АЦП

Этот бит устанавливается, когда завершается преобразование АЦП и регистры данных модифицированы. Если ADIE бит и I-бит в SREG установлены, то выполняется прерывание окончания преобразования АЦП. ADIF обнуляется МК при выполнении соответствующего прерывания. В качестве альтернативы ADIF обнуляется путем записи в него логической единицы. Остерегайтесь случая, когда при выполнении действия «чтение-изменение-запись» в ADCSR отложенное прерывание может быть заблокировано. Это также касается использования команд SBI и CBI.

- Бит 3 – ADIE: Разрешение прерывания АЦП

Когда этот бит установлен в состояние 1 и I-бит в SREG также установлен в состояние 1, активизируется прерывание завершения преобразования АЦП.

- Биты 2 ... 0 – ADPS2 ... ADPS0: Биты выбора значения делителя частоты

Эти биты определяют коэффициент деления между XTAL частотой и входной частотой АЦП.

#### *Регистр данных АЦП - ADCL и ADCH*

Бит	15	14	13	12	11	10	9	8
	-	-	-	-	-	-	ADC9	ADC8
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Бит	7	6	5	4	3	2	1	0
Чтение/запись	R	R	R	R	R	R	R	R
	R	R	R	R	R	R	R	R
Начальное значение	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Когда выполнено преобразование АЦП, результат помещается в эти два регистра. В режиме постоянного преобразования необходимо, чтобы считывались оба регистра и чтобы ADCL считывался раньше ADCH.

### **Сканирование нескольких каналов АЦП**

Так как изменение аналогового канала всегда задерживается до окончания преобразования, режим постоянного преобразования может использоваться, чтобы просканировать несколько каналов без прерывания преобразователя. Обычно для сдвига канала используется прерывание окончания преобразования АЦП. Однако пользователь должен иметь в виду следующий факт:

прерывание вызывается только тогда, когда результат готов к чтению. В режиме постоянного преобразования следующее преобразование начнется через один такт АЦП после того, как будет вызвано прерывание. Если

содержание ADMUX изменено в течение одного такта АЦП после вызова прерывания, то это изменение будет учитываться при следующем преобразовании. Если ADMUX будет изменен после одного такта после вызова прерывания и после начала следующего преобразования, то будет использоваться старая установка.

### Таймеры-счетчики

Микроконтроллер имеет до трех универсальных таймеров-счетчиков (Т/С): два 8-разрядных Т/С и один 16-разрядный Т/С. Т/С2 может быть связан с внешним генератором. Этот генератор оптимизирован для использования с кристаллом 32.768 кГц, что допускает использование Т/С2 как часов реального времени (RTC). Т/С2 имеет собственный делитель частоты. Т/С0 и Т/С1 имеют индивидуальные установки масштаба от одного и того же 10-разрядного делителя частоты. Эти таймеры-счетчики могут использоваться и как таймеры с внутренним счетом времени, и как счетчики с внешним входом, который переключает счет [3].

### Таймер-счетчик Т/С0

8-разрядный Т/С0 может использовать источник тактов СК (тактовый генератор микроконтроллера), масштабированный СК или внешний вход.

Флаг состояния переполнения присутствует в регистре флагов прерываний таймеров-счетчиков – TIFR. Сигналы управления присутствуют в регистре управления таймера-счетчика – TCCR0. Установки разрешения прерываний находятся в регистре маски прерываний таймеров-счетчиков – TIMSK.

Когда Т/С0 работает от внешнего генератора, внешний сигнал синхронизируется с частотой тактового генератора CPU. Чтобы гарантировать правильное осуществление выборки внешних тактов, минимальное время между двумя внешними переключениями должно быть не менее одного такта CPU. Переходы внешнего сигнала считаются на фронте такта CPU.

Т/С0 имеет высокое разрешение и высокую точность при использовании с низкой частотой. Его целесообразно использовать для реализации медленных функций или точных временных функций с нечастыми действиями.

#### Регистр управления TCCR0

Бит	7	6	5	4	3	2	1	0
	-	-	-	-	-	CS02	CS01	CS00
Чтение/запись	R	R	R	R	R	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – Зарезервированный бит

Этот бит зарезервирован и всегда равен нулю.

- Биты 2,1,0 – CS02, CS01, CS00: Биты выбора тактов 2,1 и 0

Эти биты определяют источник тактов и масштаб для таймера/счетчика.

CS02	CS01	CS00	Описание
0	0	0	T/C0 выключен
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Внешний вход T0, фронт↓
1	1	1	Внешний вход T0, фронт↑

#### Таймер-счетчик0 - TCNT0

Бит	7	6	5	4	3	2	1	0
	MSB							LSB
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

8-разрядный регистр, содержит значение таймера-счетчика0.

#### Таймер-счетчик T/C1

16-разрядный T/C1 может использовать источник тактов СК, масштабированный СК или внешний вывод. Различные флаги состояния (переполнение, соответствие и захват) и управляющие сигналы присутствуют в регистрах управления таймера/счетчика1 – TCCR1A и TCCR1B. Разрешение прерываний для T/C1 находятся в регистре маски прерываний TIMSK.

Когда T/C1 работает от внешнего генератора, внешний сигнал синхронизируется с частотой тактового генератора CPU. Чтобы гарантировать правильное осуществление выборки внешних тактов, минимальное время между двумя внешними переключениями должно быть не менее одного такта CPU. Переходы внешнего сигнала считаются на фронте такта CPU.

T/C1 имеет высокое разрешение и высокую точность при использовании с низкой частотой. Его целесообразно использовать для реализации медленных функций или точных временных функций с нечастыми действиями.

T/C1 поддерживает две функции сравнения выхода, использующие регистры сравнения выхода OCR1A и OCR1B как источники данных, которые нужно сравнить с содержанием T/C1. Функции сравнения выхода включают

возможность очистки счетчика при соответствии А и действия на соответствующих выводах МК (OC1A, OC1B).

T/C1 может также использоваться как 8-, 9- или 10-разрядный ШИМ. В этом режиме счетчик и регистры OCR1A/OCR1B служат как двойной свободный от сбоев автономный ШИМ с центрированными импульсами.

Функция захвата входа обеспечивает захват содержания T/C1 в регистр сбора данных – ICR1, вызывается внешним событием на выводе захвата входа МК – ICP. Фактические установки события захвата данных определяются регистром управления TCCR1B. Кроме того, функция захвата входа может активизироваться аналоговым компаратором.

Если включена функция шумоподавления, фактическое условие включения события захвата данных контролируется в течение четырех выборок, прежде чем захват данных активизируется. Входной сигнал дискретизируется с частотой XTAL.

*Регистр управления А таймера-счетчика1 - TCCR1A*

Бит	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10
Чтение/запись	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Биты 7,6 – COM1A1, COM1A0: Режим сравнения выхода 1A, биты 1 и 0

COM1A1 и COM1A0 биты определяют любое действие на выводе после соответствия в T/C1. Все действия на внешнем выводе воздействуют на вывод OC1A – вывод сравнения выхода А 1. Так как это – альтернативная функция порта ввода-вывода, соответствующий служебный бит направления должен быть установлен в состояние 1, чтобы управлять выводом. Конфигурация управления приведена в таблице:

COM1x1	COM1x0	Описание
0	0	T/C1 отключен от OC1x
0	1	Переключить OC1x
1	0	Обнулить OC1x
1	1	Установить OC1x

x = А или В

- Биты 5,4 – COM1B1, COM1B0: Режим сравнения выхода 1B, биты 1 и 0

COM1B1 и COM1B0 биты определяют любое действие на выводе после соответствия в T/C1. Все это операции воздействуют на вывод OC1B – вывод сравнения выхода В. Так как это – альтернативная функция порта ввода-вывода, соответствующий служебный бит направления должен быть установлен в состояние 1, чтобы управлять выводом.

В режиме ШИМ эти биты имеют различную функцию.

При изменении COM1x1/COM1x0 битов прерывания сравнения выхода 1 должны быть заблокированы путем обнуления битов их разрешения в регистре TIMSK. Иначе прерывание может произойти во время изменения битов.

- Биты 3 ... 2 – Зарезервированные биты

Эти биты зарезервированы и всегда равны нулю.

- Биты 1 ... 0 – PWM11, PWM10: Биты выбора ШИМ

Эти биты выбирают режим ШИМ T/C1.

PWM11	PWM10	Описание
0	0	Режим ШИМ выключен
0	1	8-битный ШИМ
1	0	9-битный ШИМ
1	1	10-битный ШИМ

#### Регистр управления таймера-счетчика 1 В - TCCR1B

Бит	7	6	5	4	3	2	1	0
	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 - ICNC1: Шумоподавление при захвате входа 1 (4 СК)

Когда ICNC1 бит обнулен, запрещена функция шумоподавления при захвате входных данных. Захват входных данных включается при первом фронте/спаде на входе захвата входных данных ICP. Когда ICNC1 бит установлен в состояние 1, четыре последовательных выборки считываются с ICP, и все выборки должны быть согласованы с условием захвата данных в ICES1 бите. Фактическая частота осуществления выборки – частота XTAL.

- Бит 6 - ICES1: Выбор края захвата входа 1

Если ICES1 бит обнулен, то содержание T/C1 перемещается в регистр сбора данных (ICR1) на фронте↓ сигнала на ICP. Если ICES1 бит установлен в состояние 1, содержание T/C1 перемещается в регистр сбора данных (ICR1) на фронте↑ сигнала на ICP.

- Биты 5, 4 - Res: Зарезервированные биты

Эти биты зарезервированы и всегда равны нулю.

- Бит 3 - CTC1: Очистка T/C1 при соответствии

Когда CTC1 бит установлен в состояние 1, T/C1 сбрасывается к \$0000 через один такт после соответствия сравнения А. Если CTC1 бит обнулен, T/C1 продолжает считать и не зависит от соответствия. Так как соответствие при сравнении обнаруживается через один такт CPU после соответствия, эта функция будет вести себя по-разному, когда используется масштаб больше 1. Когда используется масштаб, равный 1, и регистр сравнения А равен С, таймер будет считать следующим образом, если CTC1 установлен:

... | C-1 | C | C + 1 | 0 | 1 | ...

Когда масштаб равен 1/8, таймер будет считать так:

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | C + 1, 0, 0, 0, 0, 0, 0, 0 | ...

В режиме ШИМ этот бит ни на что не влияет.

- Биты 2,1,0 - CS12, CS11, CS10: Выбор тактового генератора 1, бит 2,1 и 0. Эти биты определяют источник масштабирования T/C1.

CS12	CS11	CS10	Описание
0	0	0	T/C1 выключен
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Внешний вход T1, фронт↓
1	1	1	Внешний вход T1, фронт↑

*Таймер/счетчик1 - TCNT1H и TCNT1L*

Бит	15	14	13	12	11	10	9	8
	MSB							
								LSB
Бит	7	6	5	4	3	2	1	0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Этот 16-разрядный регистр содержит масштабированное значение 16-разрядного T/C1. Чтобы гарантировать, что и старший, и младший байты читаются и записываются одновременно, когда CPU обращается к этим регистрам, доступ выполняется с использованием 8-разрядного временного регистра (TEMP). Этот временный регистр также используется при доступе к OCR1A, OCR1B и ICR1. Если основная программа и подпрограммы прерываний выполняют доступ к регистрам, используя TEMP, прерывания должны быть заблокированы в течение доступа из основной программы.

**TCNT1 Запись таймера-счетчика1**

Когда CPU записывает старший байт TCNT1H, данные помещаются в регистр TEMP. Затем, когда CPU записывает младший байт TCNT1L, этот байт данных объединяется с данными байта в регистре TEMP, и все 16 битов записываются в регистр TCNT1 одновременно. Следовательно, для полной 16-разрядной операции записи регистра сначала необходимо записывать

старший байт TCNT1H. При использовании T/C1 как 8-разрядного таймера достаточно записывать только младший байт.

#### TCNT1 Чтение таймера-счетчика1

Когда CPU читает младший байт TCNT1L, данные TCNT1L посылаются в CPU, и данные старшего байта TCNT1H помещаются в регистр TEMP. Когда CPU читает данные в старшем байте TCNT1H, CPU получает данные из регистра TEMP. Следовательно, для полной 16-разрядной операции чтения регистра сначала необходимо обратиться к младшему байту TCNT1L. При использовании T/C1 как 8-разрядного таймера достаточно читать только младший байт.

#### Регистр сравнения выхода таймера-счетчика1 - OCR1AH и OCR1AL

Бит	15	14	13	12	11	10	9	8
	MSB							
								LSB
Бит	7	6	5	4	3	2	1	0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

#### Регистр сравнения выхода таймера-счетчика1 - OCR1BH и OCR1BL

Бит	15	14	13	12	11	10	9	8
	MSB							
								LSB
Бит	7	6	5	4	3	2	1	0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Регистры сравнения выхода – 16-разрядные регистры чтения/записи.

Регистры сравнения выхода содержат данные, которые необходимо сравнивать с T/C1. Действия при соответствиях определены в регистре управления и статуса. Запись с помощью программного обеспечения, которая устанавливает TCNT1 и OCR1A или OCR1B равными одному и тому же значению, не генерирует соответствия.

Соответствие установит флаг прерывания сравнения через один такт CPU после события соответствия.

Так как регистры сравнения выхода (OCR1A и OCR1B) являются 16-разрядными регистрами, используется временный регистр TEMP, когда OCR1A/B записываются, что гарантирует, что оба байта модифицируются одновременно. Когда CPU записывает старший байт OCR1AH или OCR1BH,



данные временно сохраняются в регистре TEMP. Когда CPU записывает младший байт OCR1AL или OCR1BL, регистр TEMP одновременно записывается в OCR1AH или OCR1BH. Следовательно, для полной 16-разрядной операции записи регистра сначала должен быть записан старший байт OCR1AH или OCR1BH.

Регистр TEMP также используется при доступе к TCNT1 и ICR1. Если основная программа и подпрограммы прерываний выполняют доступ к регистрам, использующим TEMP, прерывания должны быть заблокированы в течение доступа из основной программы.

*Регистр сбора данных таймера-счетчика1 - ICR1H и ICR1L*

Бит	15	14	13	12	11	10	9	8
	MSB							
								LSB
Бит	7	6	5	4	3	2	1	0
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Регистр сбора данных – 16-разрядный регистр, предназначенный только для чтения.

Когда фронт<sup>↑</sup> или фронт<sup>↓</sup> (согласно установке ICES1) сигнала на выводе захвата входных данных ICP обнаружен, текущее значение T/C1 перемещается в регистр сбора данных – ICR1. В то же время устанавливается флаг захвата данных ICF1.

Так как регистр сбора данных (ICR1) является 16-разрядным регистром, используется временный регистр TEMP, когда читается ICR1, чтобы гарантировать, что оба байта читаются одновременно. Когда CPU читает младший байт ICR1L, данные посылаются в CPU, и данные старшего байта ICR1H помещаются в регистр TEMP. Когда CPU читает данные в старшем байте ICR1H, CPU получает данные регистра TEMP. Следовательно, для полной 16-разрядной операции чтения регистра сначала необходимо обратиться к младшему байту ICR1L.

Регистр TEMP также используется при доступе к TCNT1, OCR1A и OCR1B. Если основная программа и подпрограммы прерываний выполняют доступ к регистрам, использующим TEMP, прерывания должны быть заблокированы в течение доступа из основной программы.

### Таймер-счетчик T/C1 в ШИМ режиме

Когда выбран ШИМ режим T/C1, регистр сравнения выхода 1А - OCR1A и регистр сравнения выхода 1В - OCR1B формируют ШИМ (двойной 8-, 9- или 10-разрядный, свободного доступа, свободный от сбоев и с правильной фазой) с выводами OC1A и OC1B. В режиме ШИМ устанавливается флаг переполнения 1 - TOV1, когда счетчик изменяет направление в \$0000.

Прерывание переполнения таймера 1 функционирует точно так же, как в нормальном режиме таймера-счетчика, то есть выполняется, когда TOV1 установлен, если разрешены прерывание переполнения таймера 1 и глобальные прерывания. Это также относится и к флагам, и прерываниям сравнения вывода таймера 1.

COM1x1	COM1x0	OCx1
0	0	Отключен
0	1	Отключен
1	0	Прямой ШИМ
1	1	Инверсный ШИМ

### Таймер-счетчик T/C2

8-разрядный T/C2 может использовать источник тактов PCK (по умолчанию = СК) или масштабированный PCK.

Флаги состояния (переполнение и соответствие) присутствуют в регистре флагов прерываний таймеров-счетчиков – TIFR. Сигналы управления присутствуют в регистре управления таймера-счетчика – TCCR2. Установки разрешения прерываний находятся в регистре маски прерываний таймеров-счетчиков – TIMSK.

T/C2 имеет высокое разрешение и высокую точность при использовании с низкой частотой. Его целесообразно использовать для реализации медленных функций или точных временных функций с нечастыми действиями.

T/C2 поддерживает функцию сравнения выхода, использующую регистр сравнения выхода OCR2 как источник данных, которые нужно сравнить с содержанием T/C2. Функция сравнения выхода включает возможность очистки счетчика при соответствии и действия на соответствующем выводе МК (OC2).

T/C2 может также использоваться как 8-разрядный ШИМ. В этом режиме счетчик и регистр OCR2 служат как двойной свободный от сбоев автономный ШИМ с центрированными импульсами.

#### Регистр управления TCCR2

Бит	7	6	5	4	3	2	1	0
	-	PWM2	COM21	COM20	CTC2	CS02	CS01	CS00
Чтение/запись	R	R	R	R	R	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

- Бит 7 – Зарезервированный бит

Этот бит зарезервирован и всегда равен нулю.

- Бит 6 – PWM2 Бит выбора ШИМ

Этот бит выбирает режим ШИМ T/C2.

- Биты 5,4 – COM21, COM20: Режим сравнения выхода, биты 1 и 0

COM21 и COM20 биты определяют любое действие на выводе после соответствия в T/C2. Все действия на выводе воздействуют на вывод OC2 – вывод сравнения выхода. Так как это – альтернативная функция порта ввода-вывода, соответствующий служебный бит направления должен быть установлен в состояние 1, чтобы управлять выводом.

В режиме ШИМ эти биты имеют различную функцию.

COM21	COM20	Описание
0	0	T/C2 отключен от OC2
0	1	Переключить OC2
1	0	Обнулить OC2
1	1	Установить OC2

- Бит 3 – CTC2: Очистка T/C2 при соответствии сравнения

Когда CTC2 бит установлен в состояние 1, T/C2 сбрасывается к \$00 через один такт после соответствия сравнения. Если CTC2 бит обнулен, T/C2 продолжает считать и не зависит от соответствия. Так как соответствие при сравнении обнаруживается через один такт CPU после соответствия, эта функция будет вести себя по-разному, когда используется масштаб больше 1. Когда используется масштаб, равный 1, и регистр сравнения равен C, таймер будет считать следующим образом, если CTC2 установлен:

... | C-2 | C-1 | C | 0 | 1 | ...

Когда масштаб равен 1/8, таймер будет считать так:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...

В режиме ШИМ этот бит ни на что не влияет.

- Биты 2,1,0 – CS22, CS21, CS20: Выбор делителя частоты тактового генератора, бит 2,1 и 0

Эти биты определяют источник масштабирования T/C2.

CS22	CS21	CS20	Описание
0	0	0	T/C2 выключен
0	0	1	PCK
0	1	0	PCK/8
0	1	1	PCK/32
1	0	0	PCK/64
1	0	1	PCK/128
1	1	0	PCK/256
1	1	1	PCK/1024

*Таймер-счетчик2 – TCNT2*

Бит	7	6	5	4	3	2	1	0
	MSB							LSB
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

8-разрядный регистр, содержит значение таймера-счетчика2.

*Регистр сравнения выхода таймера-счетчика2 – OCR2*

Бит	15	14	13	12	11	10	9	8
	MSB							LSB
Чтение/запись	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Начальное значение	0	0	0	0	0	0	0	0

Регистр сравнения выхода – 8-разрядный регистр чтения/записи.

Регистр сравнения выхода содержит данные, которые необходимо сравнивать с Т/С2. Действия при соответствиях определены в регистре управления и статуса. Запись с помощью программного обеспечения, которая устанавливает TCNT2 и OCR2 равными одному и тому же значению, не генерирует соответствия.

Соответствие установит флаг прерывания сравнения через один такт CPU после события соответствия.

**Таймер-счетчик Т/С2 в ШИМ режиме**

Когда выбран ШИМ режим, Т/С2 и регистр сравнения выхода OCR2 формируют ШИМ (8-разрядный, свободного доступа, свободный от сбоев и с правильной фазой) с выводами OC2.

COM21	COM20	OC2
0	0	Отключен
0	1	Отключен
1	0	Прямой ШИМ
1	1	Инверсный ШИМ

В режиме ШИМ устанавливается флаг переполнения TOV2, когда счетчик изменяет направление в \$00.

Прерывание переполнения таймера функционирует точно так же, как в нормальном режиме таймера-счетчика, то есть выполняется, когда TOV2 установлен, если разрешены прерывание переполнения таймера 1 и глобальные прерывания. Это также относится и к флагам, и прерываниям сравнения вывода таймера.

## **Интегрированная отладочная среда AVR Studio фирмы Atmel**

AVR Studio – это интегрированное отладочное средство для микроконтроллеров фирмы Atmel семейства AVR, включающее в себя компилятор с языка ассемблер. AVR Studio позволяет пользователю полностью контролировать выполнение программ с использованием симулятора, который поддерживает все типы микроконтроллеров AVR. Отладочная среда поддерживает выполнение программ в виде ассемблерного текста формата AVR Assembler, IAR Systems' Assembler и в формате языка C компилятора фирмы IAR Systems' ICCA90 C Compiler [4].

### ***Окно исходного текста программ***

Ключевое окно в AVR Studio – это окно исходного текста программы. Когда объектный файл открыт, автоматически создается окно исходного текста программ. В окне отображается код, который выполняется в отладочном окружении (эмуляторе или программном симуляторе), а текстовый маркер всегда находится на строке, которая будет выполнена в следующем цикле.

### ***Выполнение программ и пошаговый режим***

Пользователь может выполнять программу полностью в пошаговом режиме, трассируя блоки функций или выполняя программу до места, где стоит курсор. В дополнение можно определять неограниченное число точек останова, каждая из которых может быть включена или выключена. Точки останова сохраняются между сессиями работы.

### ***Просмотр регистров***

В окне исходного текста программы выводится информация о процессе выполнения программы. В дополнение AVR Studio имеет много других окон, которые позволяют управлять им и отображать информацию о любом элементе микроконтроллера.

### ***Список доступных окон***

Watch window: Окно показывает значения определенных символов. В этом окне пользователь может просматривать значения и адреса переменных.

Trace window: Окно показывает хронологию программы, выполняемой в настоящее время.

Register window: Окно показывает содержимое регистров. Регистры можно изменять во время остановки программы.

Memory windows: Окна показывают содержимое памяти программ, данных, портов ввода/ вывода и энергонезависимого ПЗУ. Память можно

просматривать в HEX, двоичном или десятичном форматах. Содержимое памяти можно изменять во время остановки программы.

I/O window: Показывает содержимое различных регистров ввода/вывода:

EEPROM

I/O порты

Таймеры

и т. д.

Message window: Окно показывает сообщения от AVR Studio.

Processor window: В окне отображается важная информация о ресурсах микроконтроллера, включая программный счетчик, указатель стека, регистр статуса и счетчик цикла. Эти параметры могут модифицироваться во время остановки программы.

Настройки рабочего окружения сохраняются при выходе. При первом запуске требуется настроить окна для управления и вывода необходимой информации. Во время следующей загрузки настройки автоматически восстанавливаются.

С AVR Studio также совместимы любые программаторы и отладочные средства, которые поддерживают микроконтроллеры фирмы Atmel.

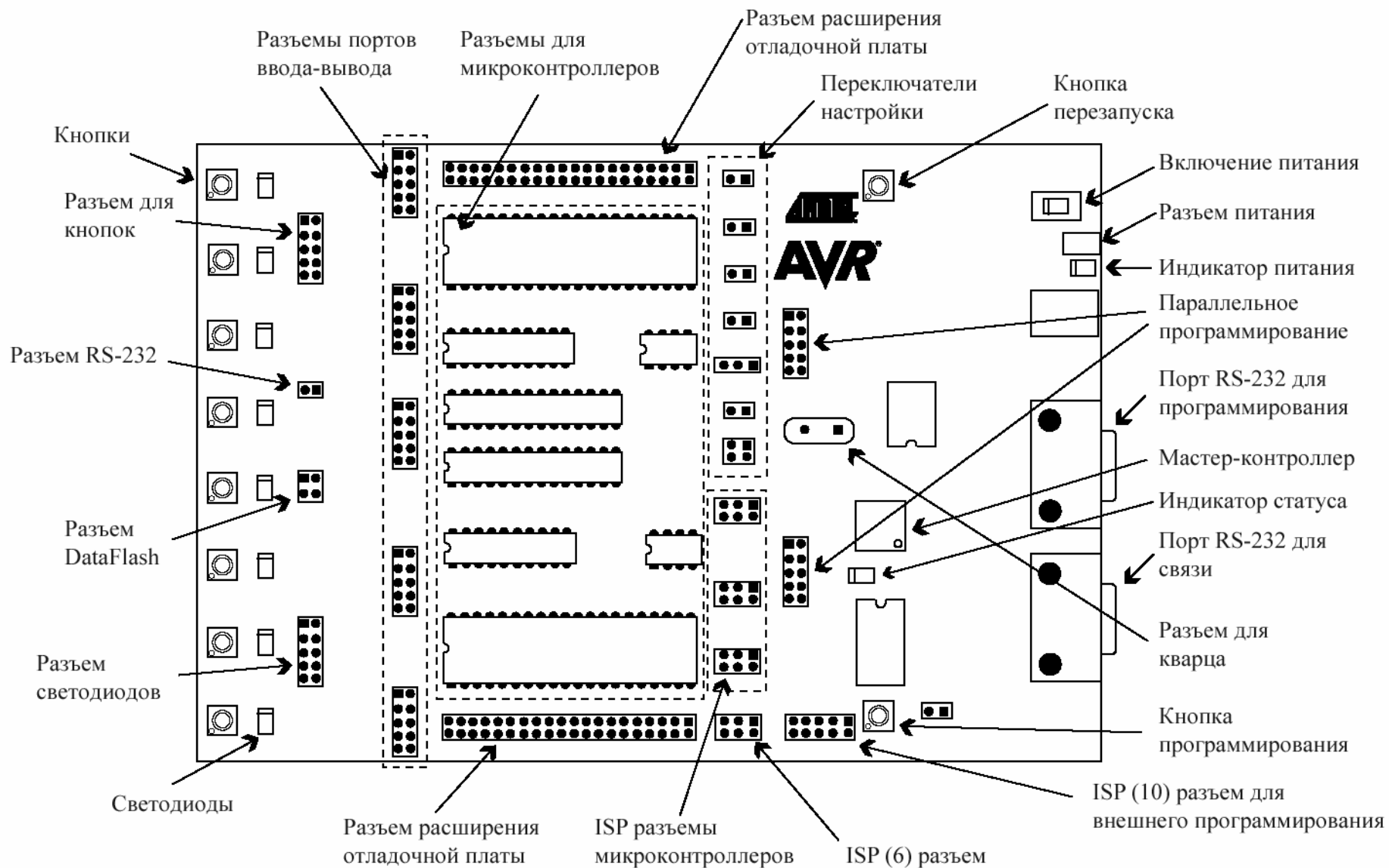


Рис. 1 Отладочная плата

## Примеры программ

### 1. Работа с портами ввода-вывода

Текст программы

---

```
.include "8535def.inc"          ; подключение файла описания МК
.device AT90S8535                ; задание конкретного МК
.def temp = r16                  ; задание временной переменной temp
.cseg                            ; установка адреса памяти программ
    rjmp RESET                  ; переход на метку RESET
RESET:
    cli                          ; глобальный запрет прерываний
    ldi temp,low(RAMEND); загрузка в temp младшего байта адреса
                                ; последней ячейки оперативной памяти МК
    out SPL,temp                ; определение указателя стека
                                ; (младший байт)
    ldi temp,high(RAMEND); загрузка в temp старшего байта адреса
                                ; последней ячейки оперативной памяти МК
    out SPH,temp                ; определение указателя стека
                                ; (старший байт)
    ldi temp,$ff                ; загрузка в temp числа $FF
    out DDRB,temp               ; запись в регистр направления порта B
                                ; значения temp ($FF – порт вывода)
    ldi temp,$00                ; загрузка в temp числа $00
    out DDRD,temp               ; запись в регистр направления порта D
                                ; значения temp ($00 – порт ввода)
LABEL:
    in temp,PIND                ; загрузка в temp значения уровней на
                                ; выводах порта D
    out PORTB,temp              ; запись в порт B переменной temp
    jmp LABEL                   ; переход на метку LABEL
```

---

Результат выполнения программы

При нажатии на кнопки SW0-SW7 на отладочной плате загораются соответствующие светодиоды LED0-LED7.



## 2. Работа с АЦП и прерываниями

## Текст программы

[illegible]

```

out ADCL ,temp      ; обнуление значения ADCL
out ADCH ,temp      ; обнуление значения ADCH
out ADMUX ,temp     ; выбор 0-го канала АЦП
ldi temp , $E8      ; загрузка в temp числа $E8
out ADCSR ,temp     ; загрузка в ADCSR значения temp
                    ; $E8 = 11101000
ldi temp , $80      ; загрузка в temp числа $80
out SREG ,temp      ; запись в регистр SREG значения temp
                    ; $80 = 10000000

LABEL:
rjmp LABEL          ; переход на метку LABEL

```

---

Результат выполнения программы

При касании 0-го вывода порта А загораются светодиоды.

### 3. Работа с таймером-счетчиком

Текст программы

---

```

.include "8535def.inc"      ; подключение файла описания МК
.device AT90S8535          ; задание конкретного МК
.def temp = r16             ; задание временной переменной temp
.def temp2 = r17           ; задание временной переменной temp2
.cseg                      ; установка адреса памяти программ
        rjmp RESET        ; переход на метку RESET
.org OC1Aaddr
        rjmp TC1CA        ; описание прерывания: при совпадении
                           ; значения таймера-счетчика с заданным
                           ; перейти на метку TC1CA

TC1CA:
        dec temp2          ; уменьшить значение temp2 на 1
        out PORTB ,temp2   ; вывести в порт В значение temp2
        reti              ; возврат из прерывания

RESET:
        cli                ; глобальный запрет прерываний
        ldi temp,low(RAMEND); загрузка в temp младшего байта адреса
                           ; последней ячейки оперативной памяти МК
        out SPL,temp       ; определение указателя стека
                           ; (младший байт)
        ldi temp,high(RAMEND); загрузка в temp старшего байта адреса

```

	; последней ячейки оперативной памяти МК
out SPH,temp	; определение указателя стека
	; (старший байт)
ldi temp,\$00	; загрузка в temp числа \$00
out TCCR1B,temp	; запись в регистр TCCR1B значения temp
out TCNT1H,temp	; запись в регистр TCNT1H значения temp
out TCNT1L,temp	; запись в регистр TCNT1L значения temp
ldi temp,\$10	; загрузка в temp числа \$10
	; \$10 = 00010000
out TIMSK,temp	; запись в регистр TIMSK значения temp
ldi temp,\$0D	; загрузка в temp числа \$0D
	; \$0D = 00001101
out TCCR1B,temp	; запись в регистр TCCR1B значения temp
ldi temp,\$10	; загрузка в temp числа \$10
out OCR1AH,temp	; запись в регистр OCR1AH значения temp
ldi temp,\$00	; загрузка в temp числа \$00
out OCR1AL,temp	; запись в регистр OCR1AL значения temp
ldi temp,\$FF	; загрузка в temp числа \$FF
out DDRB,temp	; запись в регистр направления порта В
	; значения temp (\$FF – порт вывода)
out PORTB,temp	; запись в порт В значения temp
sei	; глобальное разрешение прерываний
LABEL:	
rjmp LABEL	; переход на метку LABEL

Результат выполнения программы

Загораются светодиоды в соответствии с двоичным счетом. Время увеличения на 1 - ~1с.

# **Список команд** **Арифметические и логические операции**

Мнемокод	Операнды	Описание	Операция	Флаги	Такты
ADD	Rd, Rr	Сложение (без переноса)	$Rd \leftarrow Rd + Rr$	Z C N V S H	1
ADC	Rd, Rr	Сложение (с переносом)	$Rd \leftarrow Rd + Rr + C$	Z C N V S H	1
ADIW	Rd, K	Сложение константы и слова	$Rd+1:Rd \leftarrow Rd+1:Rd + K$	Z C N V S	2
SUB	Rd, Rr	Вычитание (без переноса)	$Rd \leftarrow Rd - Rr$	Z C N V S H	1
SUBI	Rd, K	Вычитание константы	$Rd \leftarrow Rd - K$	Z C N V S H	1
SBC	Rd, Rr	Вычитание (с переносом)	$Rd \leftarrow Rd - Rr - C$	Z C N V S H	1
SBCI	Rd, K	Вычитание константы (с переносом)	$Rd \leftarrow Rd - K - C$	Z C N V S H	1
SBIW	Rd, K	Вычитание из слова константы	$Rd+1:Rd \leftarrow Rd+1:Rd - K$	Z C N V S	2
AND	Rd, Rr	Логическое «И»	$Rd \leftarrow Rd \bullet Rr$	Z N V S	1
ANDI	Rd, K	Логическое «И» переменной и константы	$Rd \leftarrow Rd \bullet K$	Z N V S	1
OR	Rd, Rr	Логическое «ИЛИ»	$Rd \leftarrow Rd \vee Rr$	Z N V S	1
ORI	Rd, K	Логическое «ИЛИ» переменной и константы	$Rd \leftarrow Rd \vee K$	Z N V S	1
EOR	Rd, Rr	Исключающее «ИЛИ»	$Rd \leftarrow Rd \oplus Rr$	Z N V S	1
COM	Rd	Дополнение до единицы	$Rd \leftarrow \$FF - Rd$	Z C N V S	1
NEG	Rd	Инвертирование до двух	$Rd \leftarrow \$00 - Rd$	Z C N V S H	1
SBR	Rd, K	Установка бита(ов) в регистре	$Rd \leftarrow Rd \vee K$	Z N V S	1
CBR	Rd, K	Сброс бита(ов) в регистре	$Rd \leftarrow Rd \bullet (\$FF - K)$	Z N V S	1
INC	Rd	Инкрементация	$Rd \leftarrow Rd + 1$	Z N V S	1
DEC	Rd	Декрементация	$Rd \leftarrow Rd - 1$	Z N V S	1
TST	Rd	Проверка на 0 или отрицательное значение	$Rd \leftarrow Rd \bullet Rd$	Z N V S	1
CLR	Rd	Очистка регистра	$Rd \leftarrow Rd \oplus Rd$	Z N V S	1
SER	Rd	Установка регистра	$Rd \leftarrow \$FF$	-	1
MUL	Rd, Rr	Умножение незначковых	$R1:R0 \leftarrow Rd \times Rr (UU)$	Z C	2
MULS	Rd, Rr	Умножение значковых	$R1:R0 \leftarrow Rd \times Rr (SS)$	Z C	2
MULSU	Rd, Rr	Умножение значкового и незначкового	$R1:R0 \leftarrow Rd \times Rr (SU)$	Z C	2
FMUL	Rd, Rr	Умножение дробных незначковых	$R1:R0 \leftarrow (Rd \times Rr) \ll 1 (UU)$	Z C	2
FMULS	Rd, Rr	Умножение дробных значковых	$R1:R0 \leftarrow (Rd \times Rr) \ll 1 (SS)$	Z C	2
FMULSU	Rd, Rr	Умножение дробных значкового и незначкового	$R1:R0 \leftarrow (Rd \times Rr) \ll 1 (SU)$	Z C	2

## Операции переходов

Мнемокод	Операнды	Описание	Операция	Флаги	Такты
RJMP	k	Относительный переход	$PC \leftarrow PC + k + 1$	-	2
IJMP		Непрямой переход к Z	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	-	2
EIJMP		Дальний непрямой переход к Z	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	-	2
JMP	k	Переход	$PC \leftarrow k$	-	3
RCALL	k	Относительный вызов процедуры	$PC \leftarrow PC + k + 1$	-	3 / 4
ICALL		Непрямой вызов Z	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	-	3 / 4
EICALL		Дальний непрямой вызов к Z	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	-	4
CALL	k	Вызов процедуры	$PC \leftarrow k$	-	4 / 5
RET		Возврат из процедуры	$PC \leftarrow STACK$	-	4 / 5
RETI		Возврат из прерывания	$PC \leftarrow STACK$	I	4 / 5
CPSE	Rd, Rr	Сравнить, пропустить, если равно	If (Rd=Rr) $PC \leftarrow PC + 2$ или 3	-	1 / 2 / 3
CP	Rd, Rr	Сравнить	$Rd - Rr$	Z C N V S H	1
CPC	Rd, Rr	Сравнить с учетом переноса	$Rd - Rr - C$	Z C N V S H	1
CPI	Rd, K	Сравнить с константой	$Rd - K$	Z C N V S H	1
SBRC	Rr, b	Пропустить, если бит в регистре равен 0	If (Rr(b)=0) $PC \leftarrow PC + 2$ или 3	-	1 / 2 / 3
SBRB	Rr, b	Пропустить, если бит в регистре равен 1	If (Rr(b)=1) $PC \leftarrow PC + 2$ или 3	-	1 / 2 / 3
SBIC	A, b	Пропустить, если бит в регистре В/В равен 0	If (A(b)=0) $PC \leftarrow PC + 2$ или 3	-	1 / 2 / 3
SBIS	A, b	Пропустить, если бит в регистре В/В равен 1	If (A(b)=1) $PC \leftarrow PC + 2$ или 3	-	1 / 2 / 3
BRBS	s, k	Перейти, если установлен флаг в регистре статуса	If (SREG(s)=1) $PC \leftarrow PC + k + 1$	-	1 / 2
BRBC	s, k	Перейти, если сброшен флаг в регистре статуса	If (SREG(s)=0) $PC \leftarrow PC + k + 1$	-	1 / 2
BREQ	k	Перейти, если равно	If (Z=1) $PC \leftarrow PC + k + 1$	-	1 / 2
BRNE	k	Перейти, если не равно	If (Z=0) $PC \leftarrow PC + k + 1$	-	1 / 2
BRCS	k	Перейти, если установлен флаг переноса	If (C=1) $PC \leftarrow PC + k + 1$	-	1 / 2
BRCC	k	Перейти, если сброшен флаг переноса	If (C=0) $PC \leftarrow PC + k + 1$	-	1 / 2
BRSH	k	Перейти, если равно или больше	If (C=0) $PC \leftarrow PC + k + 1$	-	1 / 2
BRLO	k	Перейти, если меньше	If (C=1) $PC \leftarrow PC + k + 1$	-	1 / 2
BRMI	k	Перейти, если отрицательно	If (N=1) $PC \leftarrow PC + k + 1$	-	1 / 2
BRPL	k	Перейти, если положительно	If (N=0) $PC \leftarrow PC + k + 1$	-	1 / 2
BRGE	k	Перейти, если равно или больше, с учетом знака	If ( $N \oplus V = 0$ ) $PC \leftarrow PC + k + 1$	-	1 / 2

BRLT	k	Перейти, если меньше, с учетом знака	If ( $N \oplus V = 1$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRHS	k	Перейти, если установлен флаг Н	If ( $H=1$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRHC	k	Перейти, если сброшен флаг Н	If ( $H=0$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRTS	k	Перейти, если установлен флаг Т	If ( $T=1$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRTC	k	Перейти, если сброшен флаг Т	If ( $T=0$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRVS	k	Перейти, если установлен флаг переполнения	If ( $V=1$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRVC	k	Перейти, если сброшен флаг переполнения	If ( $V=0$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRIE	k	Перейти, если разрешены прерывания	If ( $I=1$ ) $PC \leftarrow PC + k + 1$	-	1 / 2
BRID	k	Перейти, если запрещены прерывания	If ( $I=0$ ) $PC \leftarrow PC + k + 1$	-	1 / 2

### Операции с данными

Мнемокод	Операнды	Описание	Операция	Флаги	Такты
MOV	Rd, Rr	Копировать регистр	$Rd \leftarrow Rr$	-	1
MOVW	Rd, Rr	Копировать пару регистров	$Rd + 1 : Rd \leftarrow Rr + 1 : Rr$	-	1
LDI	Rd, K	Загрузить константу	$Rd \leftarrow K$	-	1
LDS	Rd, k	Загрузить напрямую из области данных	$Rd \leftarrow (k)$	-	2
LD	Rd, X	Непрямая загрузка	$Rd \leftarrow (X)$	-	2
LD	Rd, X+	Непрямая загрузка с инкрементацией	$Rd \leftarrow (X), X \leftarrow X + 1$	-	2
LD	Rd, X-	Непрямая загрузка с декрементацией	$X \leftarrow X - 1, Rd \leftarrow (X)$	-	2
LD	Rd, Y	Непрямая загрузка	$Rd \leftarrow (Y)$	-	2
LD	Rd, Y+	Непрямая загрузка с инкрементацией	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	-	2
LD	Rd, Y-	Непрямая загрузка с декрементацией	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	-	2
LD	Rd, Y+q	Непрямая загрузка со смещением	$Rd \leftarrow (Y+q)$	-	2
LD	Rd, Z	Непрямая загрузка	$Rd \leftarrow (Z)$	-	2
LD	Rd, Z+	Непрямая загрузка с инкрементацией	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	-	2
LD	Rd, Z-	Непрямая загрузка с декрементацией	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	-	2
LD	Rd, Z+q	Непрямая загрузка со смещением	$Rd \leftarrow (Z+q)$	-	2
STS	k, Rr	Запись напрямую в область данных	$(k) \leftarrow Rr$	-	2
ST	X, Rr	Непрямая запись	$(X) \leftarrow Rr$	-	2
ST	X+, Rr	Непрямая запись с инкрементацией	$(X) \leftarrow Rr, X \leftarrow X + 1$	-	2
ST	-X, Rr	Непрямая запись с декрементацией	$X \leftarrow X - 1, (X) \leftarrow Rr$	-	2

					ОКОНЧАНИЕ
ST	Y, Rr	Непрямая запись	$(Y) \leftarrow Rr$	-	2
ST	Y+, Rr	Непрямая запись с инкрементацией	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	-	2
ST	-Y, Rr	Непрямая запись с декрементацией	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	-	2
STD	Y+q, Rr	Непрямая запись со смещением	$(Y+q) \leftarrow Rr$	-	2
ST	Z, Rr	Непрямая запись	$(Z) \leftarrow Rr$	-	2
ST	Z+, Rr	Непрямая запись с инкрементацией	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	-	2
ST	-Z, Rr	Непрямая запись с декрементацией	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	-	2
STD	Z+q, Rr	Непрямая запись со смещением	$(Z+q) \leftarrow Rr$	-	2
LPM		Загрузить из памяти команд	$R0 \leftarrow (Z)$	-	3
LPM	Rd, Z	Загрузить из памяти команд	$Rd \leftarrow (Z)$	-	3
LPM	Rd, Z+	Загрузить из памяти команд с инкрементацией	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	-	3
ELPM		Удаленная загрузка из памяти команд	$R0 \leftarrow (RAMPZ:Z)$	-	3
ELPM	Rd, Z	Удаленная загрузка из памяти команд	$Rd \leftarrow (RAMPZ:Z)$	-	3
ELPM	Rd, Z+	Удаленная загрузка из памяти команд с инкрементацией	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	-	3
SPM		Загрузка памяти команд	$(Z) \leftarrow R1 : R0$	-	-
IN	Rd, A	Считать данные из В/В	$Rd \leftarrow I/O(A)$	-	1
OUT	A, Rr	Вывести данные в В/В	$I/O(A) \leftarrow Rr$	-	1
PUSH	Rr	Положить регистр в стек	$STACK \leftarrow Rr$	-	2
POP	Rd	Извлечь регистр из стека	$Rd \leftarrow STACK$	-	2

### Операции управления микроконтроллером

Мнемокод	Операнды	Описание	Операция	Флаги	Такты
BREAK		Выход из текущей процедуры		-	1
NOP		Нет операции		-	1
SLEEP		Перейти в спящий режим		-	1
WDR		Перезапуск сторожевого таймера		-	1

## Операции с битами

Мнемокод	Операнды	Описание	Операция	Флаги	Такты
LSL	Rd	Логический сдвиг влево	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0, C \leftarrow Rd(7)$	Z C N V H	1
LSR	Rd	Логический сдвиг вправо	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0, C \leftarrow Rd(0)$	Z C N V	1
ROL	Rd	Вращение влево с переносом	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z C N V H	1
ROR	Rd	Вращение вправо с переносом	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z C N V	1
ASR	Rd	Арифметический сдвиг вправо	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z C N V	1
SWAP	Rd	Перестановка	$Rd(3..0) \leftrightarrow Rd(7..4)$	-	1
BSET	s	Установить флаг	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Сбросить флаг	$SREG(s) \leftarrow 0$	SREG(s)	1
SBI	A, b	Установить бит в регистре B/B	$I/O(A,b) \leftarrow 1$	-	2
CBI	A, b	Обнулить бит в регистре B/B	$I/O(A,b) \leftarrow 0$	-	2
BST	Rr, b	Поместить бит из регистра в T	$T \leftarrow Rr(b)$	T	1
BLD	Rr, b	Загрузить бит из T в регистр	$Rr(b) \leftarrow T$	-	1
SEC		Установит флаг переноса	$C \leftarrow 1$	C	1
CLC		Сбросить флаг переноса	$C \leftarrow 0$	C	1
SEN		Установить флаг отрицательного	$N \leftarrow 1$	N	1
CLN		Сбросить флаг отрицательного	$N \leftarrow 0$	N	1
SEZ		Установить флаг нуля	$Z \leftarrow 1$	Z	1
CLZ		Сбросить флаг нуля	$Z \leftarrow 0$	Z	1
SEI		Глобальное разрешение прерываний	$I \leftarrow 1$	I	1
CLI		Глобальное запрещение прерываний	$I \leftarrow 0$	I	1
SES		Установить флаг проверки знаковости	$S \leftarrow 1$	S	1
CLS		Сбросить флаг проверки знаковости	$S \leftarrow 0$	S	1
SEV		Установить переполнение дополнения до 2	$V \leftarrow 1$	V	1
CLV		Сбросить переполнение дополнения до 2	$V \leftarrow 0$	V	1
SET		Установить T в SREG	$T \leftarrow 1$	T	1
CLT		Сбросить T в SREG	$T \leftarrow 0$	T	1
SEH		Установить флаг полупереноса в SREG	$H \leftarrow 1$	H	1
CLH		Сбросить флаг полупереноса в SREG	$H \leftarrow 0$	H	1



**Библиографический список**

1. Atmel Corporation AVR RISC Microcontrollers Data Book. August 1999.
2. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel./ В. В. Гребнев – М.: ИП РадиоСофт, 2002. – 176 с.
3. Водовозов А. М. Микроконтроллеры для систем автоматики: Учебное пособие./А. М. Водовозов – Вологда: ВоГТУ, 2002. – 123 с.
4. Предко М. Руководство по микроконтроллерам. Т. 1,2./ М. Предко – М.: Постмаркет, 2001. – 416, 488 с.