

Министерство образования и науки Украины
Севастопольский национальный технический университет

Методические указания

по дисциплине

“ Микропроцессоры и микроконтроллеры”

Архитектура и периферийные устройства
микроконтроллеров семейства AVR.

Севастополь
2005

Министерство образования и науки Украины
Севастопольский национальный технический университет

Методические указания
Микропроцессоры и микроконтроллеры
Архитектура и периферийные устройства
микроконтроллеров семейства AVR.

Учебное пособие

Севастополь
2003

УДК681.3

Микропроцессоры и микроконтроллеры. Архитектура и периферийные устройства микроконтроллеров семейства AVR. / А.В.Астраханцев, Ю.И.Мошкович, А.А.Тестоедов. Учебное пособие. – Севастополь: СевНТУ, 2005. – 40с.

Целью учебного пособия является оказание помощи студентам в подготовке к выполнению лабораторных работ по дисциплине «Микропроцессоры и микроконтроллеры». Описана архитектура AVR-микроконтроллеров, организация его периферийных устройств, приведены примеры их программирования.

Учебное пособие предназначено для студентов, обучающихся по направлению «Электроника».

Учебное пособие рассмотрено и утверждено на заседании кафедры электронной техники (протокол № от 2005 г.)

Рецензент Широков И.Б. канд.техн. наук, доцент кафедры РТ.

СОДЕРЖАНИЕ

Введение	4
1. Архитектура микроконтроллера AT90S8535.....	4
2. Периферийные устройства и регистры ввода/вывода.....	6
2.1. Регистр флагов.....	6
2.2. Указатель стека.....	7
2.3. Система прерываний	7
2.4. Внешние прерывания	9
2.5. Режимы пониженного потребления	11
2.6. Порты ввода-вывода.....	13
2.7. Таймеры	14
2.7.1. Прерывания от таймеров.....	14
2.7.2. Таймер 0	15
2.7.3. Таймер 1	17
2.7.4. Таймер 2	22
2.8. Сторожевой таймер	26
2.9. EEPROM память	27
2.10. Универсальный асинхронный приемо-передатчик	29
2.10.1. Передача данных	29
2.10.2. Прием данных	30
2.10.3. Управление приемо-передатчиком.....	30
2.10.4. Скорость передачи.....	32
2.11. Последовательный периферийный интерфейс	33
2.12. Аналого-цифровой преобразователь	36
2.13. Аналоговый компаратор	40
Библиографический список	41
ПРИЛОЖЕНИЕ А	42

Введение

AVR – семейство популярных однокристальных микроконтроллеров фирмы Atmel (США). Семейство состоит из большого количества моделей, имеющих одинаковое процессорное ядро и различающихся объемом встроенной памяти и набором периферийных устройств. AVR-микроконтроллеры обеспечивают хорошую производительность (10 – 16 млн. операций /сек.), низкую себестоимости (1.5-10 USD в зависимости от модели) и невысокое потребление тока (1-2 mA). Все AVR-микроконтроллеры имеют встроенную flash-память программ с возможностью ее перепрограммирования непосредственно на целевой плате. Все это делает данную серию весьма привлекательной для построения современных электронных устройств различного назначения.

Для знакомства с AVR-микроконтроллерами в рамках курса «Микропроцессоры и микроконтроллеры» была выбрана микросхема AT90S8535, так как она обладает достаточным объемом встроенной памяти и практически полным набором периферийных устройств, используемых AVR-микроконтроллерами.

Микроконтроллер AT90S8535 имеет следующий набор устройств:

- flash-память программ емкостью 8 кБайт;
- ОЗУ данных емкостью 512 байт;
- EEPROM-память данных емкостью 512 байт;
- четыре 8-битных порта ввода-вывода;
- один 16-битный и два 8-битных таймера-счетчика со схемами захвата и сравнения;
- 8-канальный 10-битный АЦП последовательного приближения;
- последовательный асинхронный порт UART;
- последовательный синхронный порт SPI;
- аналоговый компаратор;
- программируемый сторожевой таймер;
- встроенная система сброса при включении питания.

1. Архитектура микроконтроллера AT90S8535.

Микроконтроллер AT90S8535, как и все приборы AVR-серии, базируется на Гарвардской архитектуре, подразумевающей хранение программы и данных в отдельных адресных пространствах.

Для хранения программ используется flash-память, организованная как 4096 шестнадцатиразрядных слов и занимающая адресное пространство в диапазоне 0-FFFF. Младшие адреса в диапазоне от 0 до \$10

используются как вектора прерывания. При включении питания управление передается на ячейку памяти с адресом 0.

Память данных, в отличие от памяти программ, имеет разрядность 8 бит и делится на несколько областей. Адреса в диапазоне от 0 до \$1F занимает регистровый файл, состоящий из 32-х рабочих регистров. В область памяти с адресами от \$20 до \$5F отображаются регистры, обслуживающие все периферийные устройства микроконтроллера. Адреса в диапазоне \$60 – \$25F используются для хранения данных. Здесь же располагается и программный стек. Схематическое изображение распределения памяти данных приведено на рисунке 1.



Рисунок 1. Структура памяти данных микроконтроллера.

К рабочим регистрам можно обращаться по их именам R0 – R31. В этом случае доступны все команды, поддерживаемые микроконтроллером. Кроме того, к регистрам можно обращаться как к обыкновенным ячейкам памяти, используя их абсолютные адреса. В этом случае доступны только команды пересылок с использованием прямой или косвенной адресации.

Любой из 32 регистров может использоваться как приемник и/или источник данных. Однако при использовании регистров имеются некоторые особенности. Для косвенной адресации могут использоваться только регистры R26 – R31. При этом регистровая пара **R26:R27** трактуется как регистр-указатель с именем **X**, **R28:R29** – как указатель с именем **Y** и **R30:R31** – как указатель с именем **Z**. Кроме того, команды с непосредственной адресацией поддерживаются только регистрами R16 – R31.

К регистрам ввода-вывода можно обращаться с помощью специальных команд **in** и **out**. В этом случае для них используются укороченные адреса в диапазоне 0 – \$3F. К регистрам ввода-вывода можно обращаться также и как к обычным ячейкам памяти, используя прямую или косвенную адресацию. В этом случае используются их

абсолютные адреса. Как правило, не возникает необходимость обращения к регистрам ввода/вывода как к ячейкам памяти ОЗУ. Адреса и названия всех регистров ввода/вывода микроконтроллера AT90S8535 приведены в Приложении А.

К ячейкам ОЗУ можно обращаться с помощью команд пересылок, используя прямой или косвенные способы адресации.

2. Периферийные устройства и регистры ввода/вывода.

2.1. Регистр флагов

Регистр флагов SREG имеет адрес 3F(5F) и содержит флаги процессора. Регистр доступен и для записи и для чтения. При включении питания в этот регистр аппаратно записывается 0. Структура регистра флагов приведена ниже.

	7	6	5	4	3	2	1	0
SREG	I	T	H	S	V	N	Z	C

Флаги регистра имеют следующие назначения.

- I (Interrupt). Глобальное разрешение прерывания. Запись “1” в этот бит разрешает прерывания в микроконтроллере. Логический “0” в этом бите запрещает все прерывания.
- T (Temporary). Является битовым аккумулятором и используется для пересылки битов рабочих регистров.
- H (Half Carry). Вспомогательный бит переноса. Индицирует перенос из 3-го в 4-ый бит результата некоторых арифметических операций.
- S (Sign). Всегда содержит результат операции «исключающее ИЛИ» между битами V и N регистра флагов. Используется в операциях условных переходов для знаковых чисел.
- V (oVerflow). Флаг переполнения для знаковых чисел.
- N (Negative). Флаг, указывающий, что в предыдущей арифметической или логической операции был получен отрицательный результат.
- Z (Zero). Флаг, указывающий, что в предыдущей арифметической или логической операции был получен нулевой результат.
- C (Carry). Флаг, индицирующий, что в предыдущей арифметической или логической операции был осуществлен перенос или заем.

2.2. Указатель стека

Стек микросхемы– AT90S8535 располагается во встроенном ОЗУ, а его вершина адресуется 10-ти разрядным указателем стека SP. Указатель стека состоит из двух регистров SPL и SPH – младшего и старшего байтов соответственно. Оба регистра доступны как для записи, так и для чтения. При сбросе микроконтроллера в них записывается значение 0. Структура регистров приведена ниже.

SPH	-	-	-	-	-	-	SP9	SP8
SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

При вызове подпрограммы адрес возврата записывается на вершину стека, а содержимое SP уменьшается на 2. При возврате из подпрограммы содержимое SP увеличивается на 2. В начале программы необходимо инициализировать стек, записав в него адрес последней ячейки памяти встроенного ОЗУ. Порядок инициализации указателя стека приведен в примере 1.

Пример 1.

```
ldi    R16, $5f
out     SPL, R16      ; загрузка младшего байта стека
ldi     R16, 2
out     SPH, R16      ; загрузка старшего байта стека
```

2.3. Система прерываний

Микроконтроллер AT90S8535 обеспечивает 17 различных источников прерываний. Все прерывания имеют различные программные вектора, сосредоточенные в младших адресах памяти программ. Все прерывания, за исключением сброса микроконтроллера, имеют биты индивидуального разрешения, и могут быть обработаны в том случае, если I-бит регистра флагов установлен в “1”. В таблице 1 приведен список векторов прерывания микроконтроллера AT90S8535.

Таблица 1. Векторы прерывания микроконтроллера AT90S8535

№ вект.	Адр.	Источник	Описание
1	00	RESET	Сброс и сторожевой таймер
2	01	INT0	Внешнее прерывание 0
3	02	INT1	Внешнее прерывание 1
4	03	TIMER2 COMP	Модуль сравнения таймера 2
5	04	TIMER2 OVF	Переполнение таймера 2
6	05	TIMER1 CAPT	Модуль захвата таймера 1
7	06	TIMER1 COMPA	Модуль захвата А таймера 1
8	07	TIMER1 COMPB	Модуль захвата В таймера 1
9	08	TIMER1 OVF	Переполнение таймера 1
10	09	TIMER0 OVF	Переполнение таймера 0
11	0A	SPI	Синхронный порт
12	0B	UART, RX	Асинхронный порт, готовность приемника
13	0C	UART, UDRE	Асинхрон. порт, готовность передатчика
14	0D	UART, TX	Асинхрон. порт, данные переданы
15	0E	ADC	Завершение преобразования АЦП
16	0F	EE_RDY	Данные в EEPROM записаны
17	10	ANA_COMP	Аналоговый компаратор

Вектора прерываний с меньшими адресами имеют более высокий приоритет. Таким образом, при одновременном возникновении прерываний INT1 и SPI сначала будет отработано прерывание INT1, а потом SPI.

В том случае, если в программе используются прерывания, в соответствующих адресах памяти программ должны находиться команды безусловных переходов на соответствующие подпрограммы обработки. В примере 2 приведен фрагмент программы, настраивающий вектора прерывания. В примере предполагается, что программа использует прерывания INT0 и TIMER0 OVF.

Пример 2

```

        rjmp      start      ; переход на начало программы
        rjmp      ext0       ; переход на обработку прерывания
        .org      9          ; директива установки адреса
        rjmp      tim0       ; переход на обработку прерывания

start:   ldi       R16, $5f
        .
        .
        .
        .

```

```

ext0:      .          .          . ; тело подпрограммы прерывания INT0
          reti
tim0:      .          .          . ; тело подпрограммы прерывания TIMER0
OVF
          reti

```

На адрес 0 управление передается в следующих случаях:

- сброс по включению питания;
- внешний сброс;
- сброс при срабатывании сторожевого таймера.

Иногда бывает целесообразно различать причины, вызвавшие сброс процессора. В этом случае можно воспользоваться содержимым регистра состояния микроконтроллера MCUSR. Данный регистр имеет адрес \$34(\$54) и содержит только два значащих бита. Структура регистра приведена ниже.

	7	6	5	4	3	2	1	0
MCUSR	-	-	-	-	-	-	EXTRF	PORF

Регистр содержит следующие биты:

- EXTRF – External Reset Flag – флаг внешнего сброса;
- PORF - Power On Reset Flag – флаг включения питания.

Комбинация состояний этих битов определяют причины сброса процессора. Возможные комбинации состояний приведены в таблице 2.

Таблица 2. Состояние MCUSR после сброса

PORF	EXTRF	Источник сброса
0	0	Сторожевой таймер
0	1	Внешний сброс
1	0	Включение питания
1	1	Включение питания

2.4. Внешние прерывания

Микросхема AT90S8535 имеет два вывода внешних прерываний – INT0 и INT1. Определение режимов работы и обслуживание внешних

прерываний обеспечивается с помощью трех регистров GIMSK, GIFR и MCUCR. Эти регистры доступны как для чтения, так для записи. После сброса процессора в них записывается начальное значение 0.

Регистр GIMSK (Global Interrupt Mask) имеет адрес \$3B (\$5B) и содержит биты разрешения прерываний. Структура регистра приведена ниже.

	7	6	5	4	3	2	1	0
GIMSK	INT1	INT0	-	-	-	-	-	-

Регистр содержит следующие биты:

- INT1 – бит разрешения внешнего прерывания 1. Логическая “1” в этом бите разрешает прерывание, “0” – запрещает его.
- INT0 – бит разрешения внешнего прерывания 0. Логическая “1” в этом бите разрешает прерывание, “0” – запрещает его.

Регистр GIFR (Global Interrupt Flag Register) имеет адрес \$3A (\$5A) и содержит флаги внешних прерываний. Структура регистра приведена ниже.

	7	6	5	4	3	2	1	0
GIFR	INTF1	INTF0	-	-	-	-	-	-

Регистр содержит следующие биты:

- INTF1 – флаг внешнего прерывания 1.
- INTF0 – флаг внешнего прерывания 0.

После того, как на выводах INT возникает событие, трактуемое как прерывание, соответствующие биты устанавливаются в состояние логической “1”. В том случае, если данное прерывание разрешено соответствующими битами регистра GIMSK и I-бит регистра SREG установлен в “1”, берется соответствующий вектор прерывания. В этом случае биты INTF0 и INTF1 сбрасываются в 0 автоматически. Если прерывание не берется, данные биты можно сбросить программно, записав в них логическую “1”.

Регистр MCUCR (MicroController Unit Control Register) имеет адрес \$35(\$55) и содержит биты, определяющие характер события, вызывающего внешнее прерывание. Кроме того, данный регистр содержит биты управления режимом пониженного энергопотребления микроконтроллера. Структура регистра приведена ниже.

	7	6	5	4	3	2	1	0
MCUCR	-	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00

Регистр содержит следующие биты

- SE (Sleep Enable) – разрешение режима пониженного потребления. В том случае, если бит установлен в “1”, исполнение команды **sleep** приведет переход микроконтроллера в режим пониженного потребления.
- SM1 и SM0 (Sleep Mode) – определение типа режима пониженного потребления. Комбинации состояния этих бит определяют следующие режимы

SM1	SM0	Режим
0	0	Idle
0	1	Зарезервирован
1	0	Power Down
1	1	Power Save

Описание режимов пониженного энергопотребления будет приведено ниже.

- ISC11, ISC10 (Interrupt Sense Control) – биты управления режимом срабатывания прерывания INT1. Комбинации состояния этих бит определяют следующие режимы внешнего прерывания INT1

ISC11	ISC10	Режим прерывания
0	0	Запрос прерывания генерируется низким уровнем на INT1
0	1	Зарезервирован
1	0	Запрос прерывания генерируется спадающим фронтом на INT1
1	1	Запрос прерывания генерируется нарастающим фронтом на INT1

- ISC01, ISC00 - биты управления режимом срабатывания прерывания INT0. Имеют такие же значения как ISC11, ISC10.

2.5. Режимы пониженного потребления

Микросхема AT90S8535 имеет три режима пониженного энергопотребления. После выполнения команды **sleep** микросхема

переходит в режим, определенный состоянием битов SM1и SM0 регистра MCUCR.

В режиме **Idle** (холостой ход) процессорное ядро выключается, но тактовый генератор и все периферийные устройства микроконтроллера продолжают работать. Потребление микросхемы в этом случае снижается в несколько раз. Выход из этого режима может быть осуществлен с помощью сброса или любым разрешенным прерыванием. После обслуживания прерывания управление передается команде, непосредственно следующей за командой **sleep**.

В режиме **Power Down** (экономичный режим) выключаются не только процессорное ядро, но и тактовый генератор, и все периферийные устройства. Содержимое ОЗУ микроконтроллера и состояние его портов при этом остается неизменным. Ток потребления микросхемы в этом случае падает до единиц микроампер. Выход из режима **Power Down** может быть осуществлен либо сбросом микроконтроллера, либо внешним прерыванием по уровню (биты ISCx1 и ISCx0 установлены в ноль).

Режим **Power Save** (сбережение потребления) аналогичен режиму **Power Down**, за исключением того, что таймер 2, настроенный на асинхронный режим (см. п. 2.7.4), продолжает работать.

Пример 3.

Ожидать спадающий фронт на входе INT0, используя Idle режим пониженного энергопотребления.

```

    rjmp  start                ; переход на начало программы
    rjmp  int                  ; переход на обработку прерывания
start:  .      .      .

    ldi   R16,    $40           ; на всякий случай сбрасываем
    out   TIFR,   R16          ; флаг требования прерывания

    out   TIMSK,  R16          ; разрешаем прерывание INT0

    ldi   R16,    0b01000010    ; разрешаем Idle режим и
    out   MCUCR,  R16          ; настраиваем фронт INT0

    sei                           ; разрешаем глобальные прерывания
    sleep                          ; входим в режим Idle
    cli                           ; запрещаем прерывания
    .      .      .

int:    reti                   ; пустая процедура обработки
                                ; прерывания

```

2.6. Порты ввода-вывода

Микроконтроллер AT90S8535 имеет четыре 8-битных порта – A, B, C и D. Каждая линия любого порта может быть настроена для работы в качестве собственно пользовательского ввода или вывода, или для обеспечения работы периферийных устройств. Назначение выводов портов для поддержки периферийных устройств осуществляется с помощью битов в регистрах управления данных устройств.

Любую линию каждого порта можно независимо настроить как на ввод, так и на вывод цифровой информации. В том случае, если линия настроена на ввод, имеется возможность подключить внутренние резисторы, подтягивающие данный вывод к напряжению питания.

Для обслуживания каждого порта отведено по три регистра:

- PORTA, DDRA, PINA – для порта A;
- PORTB, DDRB, PINB – для порта B;
- PORTC, DDRC, PINC – для порта C;
- PORTD, DDRD, PIND – для порта D.

Адреса всех этих регистров приведены в Приложении А.

Содержимое регистров DDRx (Data Direct Register) определяет направление передачи данных по каждой линии порта. Если бит регистра DDRx установлен в “0”, соответствующая линия порта настраивается на ввод, в противном случае – данная линия используется для вывода цифровой информации.

Если линия порта настроена на вывод, то ее состояние будет определяться содержимым соответствующего бита регистра PORTx.

Если линия порта настроена на ввод, установка в “1” соответствующего бита в регистре PORTx приведет к подключению к этой линии подтягивающего резистора.

Состояние линий порта, в зависимости от комбинации битов регистров DDRx и PORTx, приведено в таблице 3.

Таблица 3. Режим и состояние портов

DDR _x	PORT _x	Вх./Вых.	Подтяг. Рез.	Состояние линии
0	0	Вход	Нет	Третье сост. (Hi-Z)
0	1	Вход	Да	Источник слабого тока, если на линии лог. 0
1	0	Выход	Нет	Логический 0
1	1	Выход	Нет	Логическая 1

Регистры PINx доступны только для чтения. Они содержат логические значения, присутствующие на выводах портов.

Пример 4.

Инициализировать порт В: линии 0,1,5,7 – вывод, остальные – ввод. Линии 1 и 5 установить в состояние 0, остальные - в состояние 1. На линиях 2 и 3 включить подтягивающие резисторы.

```
ldi    R16,      0b10100011
out    DDRB,     R16
ldi    R16,      0b10001101
out    PORTB,    R16
```

2.7. Таймеры

Микроконтроллер AT90S8535 имеет три пользовательских таймера:

- простой 8-разрядный таймер 0;
- 16-разрядный таймер 1 с модулем захвата и двумя модулями сравнения;
- 8-разрядный таймер 2 с модулем сравнения и возможностью работы в асинхронном режиме от дополнительного резонатора.

2.7.1. Прерывания от таймеров

Каждый таймер имеет собственные вектора прерывания по переполнению и срабатыванию модулей захвата и сравнения. Управление прерываниями от таймеров осуществляется с помощью двух общих регистров TIFR и TIMSK.

Регистр TIMSK (Timer Interrupt Mask) содержит биты разрешения прерывания от таймеров. Запись логических единиц в биты регистра TIMSK разрешает соответствующие прерывания, запись “0” – запрещает их. Структура регистра TIMSK приведена ниже.

	7	6	5	4	3	2	1	0
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0

- OCIE2 (Output Compare Interrupt Enable) – разрешение прерывание от модуля сравнения таймера 2;
- TOIE2 (Timer Overflow Interrupt Enable) – разрешение прерывания по переполнению таймера 2;
- TICIE1 (Timer Input Capture Interrupt Enable) – разрешение прерывания от модуля захвата таймера 1;

- OCIE1A – разрешение прерывания от модуля сравнения А таймера 1;
- OCIE1B – разрешение прерывания от модуля сравнения В таймера 1;
- TOIE1 – разрешение прерывания по переполнению таймера 1;
- TOIE0 – разрешение прерывания по переполнению таймера 0.

Регистр TIFR (Timer Interrupt Flag Register) содержит флаги требования прерывания таймеров. В том случае, если происходит переполнение какого-либо таймера, или срабатывание модуля захвата или сравнения, устанавливается соответствующий бит в регистре TIFR. Если разрешено глобальное прерывание и установлен соответствующий бит разрешения прерывания в регистре TIMSK, управление передается на соответствующий вектор прерывания (см. п. 2.3.). Если данное прерывание выполняется, бит регистра TIFR, вызвавший это прерывание, сбрасывается автоматически. Если прерывания не используются, то сбросить биты регистра TIFR можно, записав в них логическую “1”.

Структура регистра TIFR приведена ниже.

	7	6	5	4	3	2	1	0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0

- OCF2 (Output Compare Flag) – флаг требования прерывания модуля сравнения таймера 2;
- TOV2 (Timer Overflow) – флаг требования прерывания по переполнению таймера 2;
- ICF1 (Input Capture Flag) – флаг требования прерывания модуля захвата таймера 1;
- OCF1A – флаг требования прерывания модуля сравнения А таймера 1;
- OCF1B – флаг требования прерывания модуля сравнения В таймера 1;
- TOV1 – флаг требования прерывания по переполнению таймера 1;
- TOV0 – флаг требования прерывания по переполнению таймера 0.

2.7.2. Таймер 0

Таймер 0 является простым 8-битным инкрементирующим счетчиком. В качестве входного сигнала могут использоваться либо импульсы тактового генератора микроконтроллера, подключаемого через

программируемый предварительный делитель, либо внешние импульсы, подключаемые к входу T0 микроконтроллера, объединенного с линией порта PB0.

Текущее содержимое счетчика можно найти в регистре TCNT0. Этот регистр в любой момент времени доступен как для записи, так и для чтения. С каждым тактом входной частоты содержимое этого регистра увеличивается на единицу. Переполнение счетчика происходит при изменении кода в регистре TCNT0 из состояния \$FF в состояние 0.

Регистр TCCR0 (Timer/Counter Control Register) содержит биты управления таймера 0. Структура регистра TCCR0 приведена ниже.

	7	6	5	4	3	2	1	0
TCCR0	-	-	-	-	-	CS02	CS01	CS00

- CS02, CS01 и CS00 (Clock Select) – выбор тактирования. Комбинация состояния этих бит задает следующие режимы работы таймера и коэффициент деления предделителя. Символом СК обозначена частота тактирования микроконтроллера.

CS02	CS01	CS00	Режим тактирования таймера 0
0	0	0	Таймер 0 остановлен
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	Падающий фронт входа T0
1	1	1	Растущий фронт входа T0

Пример 5.

С помощью таймера 0 организовать задержку, равную 1680 микросекундам. Тактовая частота микроконтроллера 1 MHz.

Для организации такой задержки имеет смысл использовать предделитель СК/8. Тогда частота импульсов на входе счетчика составит $1\text{MHz} / 8 = 125\text{ kHz}$, а период следования импульсов – $8\text{ }\mu\text{s}$. Количество тактов, соответствующее 1680 μs , равно $1680 / 8 = 210$. Так как таймер 0 является инкрементирующим, в регистр TCNT0 необходимо записать значение $(256 - 210) = 46$.

```

in    R16,    TIFR      ; на всякий случай сбрасываем
cbr   R16,    1<<TOV0  ; флаг требования прерывания
out   TIFR,    R16
ldi   R16,    46        ; загружаем таймер 0
out   TCNT0,   R16
ldi   R16,    2        ; запускаем таймер 0
out   TCCR0,   R16
wait: in    R16, TIFR      ; ждем переполнения таймера 0
      sbrs   R16, TOV0
      rjmp  wait
      out   TOV0,    R16    ; сбрасываем флаг переполнения

```

2.7.3. Таймер 1

Таймер 1 является 16-битным инкрементирующим счетчиком. В качестве входного сигнала могут использоваться либо импульсы тактового генератора микроконтроллера, подключаемого через программируемый предварительный делитель, либо внешние импульсы, подключаемые к входу T1 микроконтроллера, объединенного с линией порта PB1.

На базе таймера 1 реализован модуль захвата, позволяющий аппаратно перезаписывать текущее содержимое таймера TCNT1 в регистр захвата ICR1, в ответ на перепад внешнего сигнала, подключенного на вход микроконтроллера ICP/PD6.

На базе таймера 1 реализованы два модуля сравнения, позволяющие аппаратно генерировать определенные сигналы на выводах OC1A/PD5 и OC1B/PD4 в момент равенства содержимого таймера TCNT1 и регистров OCR1A или OCR1B. Оба модуля могут работать как ШИМ-генераторы.

Так как таймер 1 является 16-разрядным, его текущее значение располагается в двух регистрах – TCNT1L (младший байт) и TCNT1H (старший байт). Оба регистра в любой момент времени доступны для записи и для чтения. Однако для обеспечения синхронной загрузки таймера необходимо при записи сначала записывать старший байт (он попадает во временный буфер), а затем младший (при записи младшего байта все 16-разрядное слово синхронно загружается в TCNT1). При чтении необходимо соблюдать обратную последовательность. Сначала читается младший байт (при этом старший байт одновременно записывается во временный буфер), а затем старший (физически осуществляется чтение из буфера). Переполнение таймера наступает в момент перехода содержимого TCNT1 из состояния \$FFFF в состояние 0.

Управление таймером 1 осуществляется с помощью двух регистров – TCCR1A и TCCR1B.

Регистр TCCR1A (Timer/Counter Control Register) имеет следующую структуру:

TCCR1A							
7	6	5	4	3	2	1	0
COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10

- COM1A1, COM1A0 (Compare Output Mode) – режим модуля сравнения А. Комбинация содержимого этих бит задает отклик вывода OC1A в ответ на совпадение содержимого регистра сравнения OCR1A и таймера TCNT1.

COM1A1	COM1A0	Режим работы модуля сравнения А
0	0	Модуль сравнения отключен от вывода OC1A
0	1	Переключение OC1A на противоположное
1	0	Сброс OC1A в 0
1	1	Установка OC1A в 1

В режиме ШИМ-генератора эти биты имеют другое назначение и будут рассмотрены ниже.

- COM1B1, COM1B0 – режим модуля сравнения В. Комбинация содержимого этих бит задает отклик вывода OC1B в ответ на совпадение содержимого регистра сравнения OCR1B и таймера TCNT1.

COM1B1	COM1B0	Режим работы модуля сравнения В
0	0	Модуль сравнения отключен от вывода OC1B
0	1	Переключение OC1B на противоположное
1	0	Сброс OC1B в 0
1	1	Установка OC1B в 1

В режиме ШИМ-генератора эти биты имеют другое назначение.

- PWM11, PWM10 (Pulse Width Modulator) – таймер 1 в режиме ШИМ-генератора. Зависимость режима работы ШИМ от состояния этих бит приведена ниже.

PWM11	PWM10	Режим работы ШИМ
0	0	Работа ШИМ запрещена
0	1	8-ми разрядный ШИМ
1	0	9-ти разрядный ШИМ
1	1	10-ти разрядный ШИМ

Регистр TCCR1B имеет следующую структуру:

TCCR1B

7	6	5	4	3	2	1	0
ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10

- ICNC1 (Input Capture Noise Canceler) – подавление шума входа захвата. Если этот бит сброшен, подавление шума запрещено и захват осуществляется по первому заданному (нарастающему или падающему) перепаду сигнала на входе ICP. Если бит установлен, захват осуществляется в том случае, если сигнал, вызывающий захват, удерживается на входе ICP в течение четырех тактов микроконтроллера.
- ICES1 (Input Capture Edge Select) – выбор фронта сигнала захвата. Если бит сброшен, захват осуществляется по спадающему фронту на входе ICP. Если бит установлен в “1” – захват осуществляется по нарастающему фронту.
- CTC1 (Clear Timer/Counter) – очистка таймера 1. Если бит установлен, регистр TCNT1 сбрасывается в состояние \$0000 при срабатывании модуля сравнения A.
- CS12,CS11,CS10 (Clock Select) – выбор тактирования таймера 1. Комбинация этих бит задает режимы тактирования таймера 1.

CS12	CS11	CS10	Режим тактирования таймера 1
0	0	0	Таймер 1 остановлен
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 вывод T1, падающий фронт
1	1	1	Внешний вывод T1, растущие фронт

Регистры, обеспечивающие работу модулей захвата и сравнения, являются 16-разрядными, но доступны для записи и чтения как байты. Так

регистр захвата состоит из регистров ICR1L – младший байт, и ICR1H – старший байт. Регистры сравнения доступны как OCR1AL и OCR1AH для модуля сравнения А, и OCR1BL, OCR1BH – для модуля сравнения В. Правила записи и чтения для этих регистров такие же, как и для TCNT. То есть при записи сначала записывается старший байт, затем – младший. Для чтения – сначала читается младший байт, затем старший.

Если таймер 1 работает в режиме ШИМ, таймер и модули сравнения формируют 8-ми, 9-ти или 10-разрядный непрерывный, свободный от «дрожания» и правильный по фазе сигнал, выводимый на выходы OC1A и/или OC1B. В этом случае таймер 1 работает как реверсивный счетчик, считающий от 0 до конечного значения (см. таблицу 4). При достижении конечного значения счетчик начинает считать в обратную сторону, после чего цикл повторяется. Таким образом, частота повторения импульсов ШИМ зависит от частоты, поступающей на вход таймера.

Таблица 4. Частота работы ШИМ

Разрешение ШИМ	Конечное знач.таймера	Частота ШИМ
8 бит	\$00FF (255)	Ftc1/510
9 бит	\$01FF (511)	Ftc1/1022
10 бит	\$03FF (1023)	Ftc1/2046

Длительность импульсов, генерируемых ШИМ, определяется содержимым регистров OCR1A и/или OCR1B. Форма импульсов определяется состоянием битов управления COM1x1 и COM1x0 регистра TCCR1A. Режимы работы модуля сравнения в зависимости от значения этих бит приведены в таблице 5.

Таблица 5. Режимы работы модуля сравнения таймера 1

COM1X1	COM1X0	Состояние выхода OC1x
0	0	Не подключен
0	1	Не подключен
1	0	Установлен в 1, если TCNT>OCR1x (неинвертирующий ШИМ)
1	1	Установлен в 1, если TCNT<OCR1x (инвертирующий ШИМ)

Временные диаграммы работы ШИМ приведены на рисунке 2.

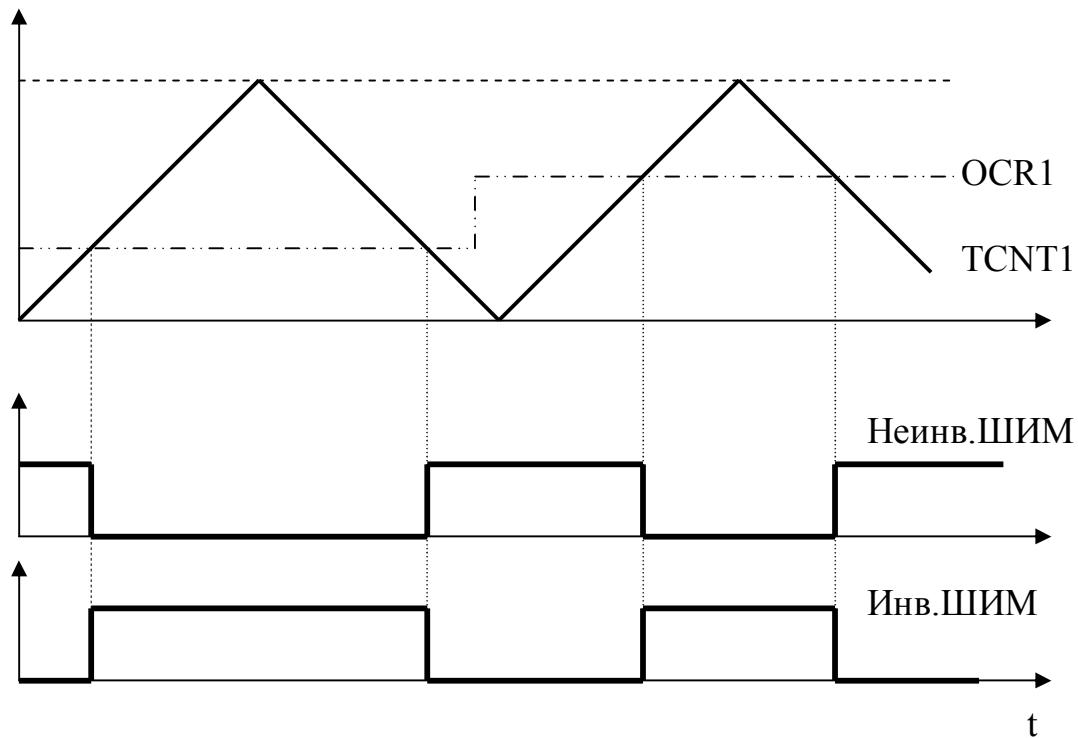


Рисунок 2. Работа ШИМ- генератора

Пример 6.

С помощью модуля захвата измерить ширину позитивного импульса в единицах количества тактов микроконтроллера. Результат занести в регистры r17:r16.

```

in      R16,    TIFR      ; на всякий случай сбрасываем
cbr     R16,    1<<ICF1   ; флаг прерывания модуля
out     TIFR,   R16       ; захвата

ldi     R16,    0b01000001 ; настраиваем модуль захвата
out     TCCR1B, R16       ; на переход 0->1 и запускаем
                                ; таймер 1

wait1:  in      R16,    TIFR      ; ожидаем захват
sbrs   R16,    ICF1
rjmp   wait1
out     TIFR,   R16       ; очищаем флаг захвата

in      R18,    ICR1L      ; читаем время перехода 0->1
in      R19,    ICR1H
ldi     R16,    1          ; настраиваем модуль захвата
out     TCCR1B, R16       ; на переход 1->0

```

```

wait0:  in      R16,      TIFR      ; ожидаем захват
        sbrs    R16,      ICF1
        rjmp   wait0
        in      R16,      ICF1L     ; читаем время перехода 1->0
        in      R17,      ICF1H
        sub     R16,      R18        ; определяем разность времен
        sbc     R17,      R19

```

Пример 7.

На выходе микроконтроллера генерировать меандр частотой 10 kHz. Тактовая частота микроконтроллера 8 MHz.

Для генерации импульсов с частотой 10 kHz необходимо, чтобы длительность импульса равнялась 50 μ s. Длительность такта контроллера равна $1/8 \text{ MHz} = 0.125 \text{ } \mu\text{s}$. Для выдержки временного интервала 50 μ s потребуется $50/0.125 = 400$ тактов.

```

clr     R16                                ; очищаем таймер1
out     TCNT1H, R16
out     TCNT1L, R16
ldi     R16, high(400) ; загружаем в регистр сравнения
out     OCR1AH, R16    ; число 400
ldi     R16, low(400)
out     OCR1AL, R16
ldi     R16, 0b01000000; назначаем режим переключения
out     TCCR1A, R16    ; вывода OC1A

ldi     R16, 0b00001001; устанавливаем режим сброса таймера
1 out     TCCR1B, R16    ; при срабатывании модуля сравнения
                           ; и запускаем таймер

```

2.7.4. Таймер 2

Таймер 2 является 8-битным инкрементирующим счетчиком. На его базе реализован один модуль сравнения, выход которого подключен к порту OC2/PD7. Кроме того, таймер 2 может работать в асинхронном режиме с дополнительным тактовым генератором, оптимизированным на подключение внешнего кварцевого резонатора 32768 Hz к выводам TOS1/PC6 и TOS1/PC7. На вход генератора TOS1 можно также подавать внешние тактовые импульсы частотой 0...256 kHz. Текущее содержимое

счетчика может быть доступно как для записи, так и для чтения через регистр TCNT2. Флаг переполнения таймера 2 TOV2 аппаратно устанавливается в “1” в ответ на переход содержимого TCNT2 из состояния \$FF в 0.

Регистр TCCR2 (Timer/Counter Control Register) управляет работой таймера 2. Его структура приведена ниже.

	7	6	5	4	3	2	1	0
TCCR2	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20

- PWM2 (Pulse Width Modulator) – если бит установлен в 1 модуль сравнения работает как ШИМ-генератор;
- COM21, COM20 (Compare Output Mode) - режим выхода модуля сравнения. Режим работы в зависимости от состояния этих бит приведен ниже.

COM21	COM20	Режим работы
0	0	Модуль сравнения отключен от вывода OC2
0	1	Переключение OC2 на противоположное
1	0	Сброс OC2 в 0
1	1	Установка OC2 в 1

Если включен режим ШИМ-генератора значения этих бит имеет другой смысл.

- CTC2 (Clear Timer/Counter) – сброс таймера 2. Если бит установлен, в регистр TCNT2 аппаратно записывается 0 в момент срабатывания модуля сравнения.
- CS22, CS21, CS20 (Clock Select) – выбор тактирования таймера 2. Соответствие состояния этих бит режиму тактирования таймера приведено ниже

CS22	CS21	CS20	Режим тактирования таймера 2
0	0	0	Таймер 2 остановлен
0	0	1	TCK2
0	1	0	TCK2/8
0	1	1	TCK2/32
1	0	0	TCK2/64
1	0	1	TCK2/128
1	1	0	TCK2/256
1	1	1	TCK2/1024

Работа таймера 2 в режиме ШИМ-генератора аналогична работе 8-битного ШИМ-генератора на базе таймера 1.

Длительность импульсов, генерируемых ШИМ, определяется содержимым регистра OCR2. Форма импульсов определяется состоянием битов управления COM21 и COM20 регистра TCCR2. Возможные значения их состояния приведены в таблице 6.

Таблица 6. Режимы работы модуля сравнения таймера 2

COM21	COM20	Состояние выхода OC2
0	0	Не подключен
0	1	Не подключен
1	0	Установлен в 1, если TCNT2>OCR2 (неинвертирующий ШИМ)
1	1	Установлен в 1, если TCNT2<OCR2 (инвертирующий ШИМ)

После прохождения сигнала сброса для таймера 2 аппаратно устанавливается синхронный режим работы. После перевода его в асинхронный режим все операции по загрузке регистров TCNT2, TCCR2 и OCR2 синхронизируются собственной тактовой частотой таймера. Так как тактовая частота микроконтроллера может очень сильно отличаться от собственной частоты таймера, необходим механизм, позволяющий быть уверенным, что данные, записанные в эти регистры, достигли своего назначения. Для этих целей таймер 2 имеет дополнительный статусный регистр ASSR (Asynchronous Status Register). Его структура приведена ниже.

	7	6	5	4	3	2	1	0
ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB

- AS2 (Asynchronous) – бит режима таймера 2. Установка этого бита переводит таймер 2 в асинхронный режим. При переводе таймера в асинхронный режим содержимое регистров TCNT2, OCR2 и TCCR2 может быть повреждено.
- TCN2UB (Timer/Counter2 Update Busy) – единичное значение этого бита указывает, что данные, записанные в регистр TCNT2, еще не загружены. Нулевое состояние этого бита показывает, что операция записи завершена. Бит доступен только для чтения.
- OCR2UB (Output Compare Register Update Busy) – единичное значение этого бита указывает, что данные, записанные в регистр OCR2, еще не загружены. Нулевое состояние этого бита

показывает, что операция записи завершена. Бит доступен только для чтения.

- TCR2UB (Timer Control Register Update Busy) – единичное значение этого бита указывает, что данные, записанные в регистр TCCR2, еще не загружены. Нулевое состояние этого бита показывает, что операция записи завершена. Бит доступен только для чтения.

Пример 8.

Организовать Power Down режим микроконтроллера длительностью 5 секунд. Таймер 2 работает от кварцевого резонатора 32768 Hz.

Длительность одного такта генератора составит $1/32768=30.517578 \mu s$. При использовании предварительного делителя с коэффициентом деления 1024, можно получить период следования импульсов на входе таймера, равный 31.25 ms. Тогда для временной выдержки, равной 5 s, потребуется $5000/31.25 = 160$ импульсов.

```

                                rjmp start                ; переход на начало программы

                                .org 4                    ; переход на вектор прерывания
                                rjmp tim2                  ; переполнения таймера 2
start:
                                .      .      .

                                ldi R16, 1<<AS2          ; устанавливаем асинхронный
                                out ASSR, R16              ; режим

                                ldi R16, 256-160          ; загружаем таймер 2
                                out TCNT2, R16
                                ldi R16, 0b00000111       ; запускаем таймер
                                out TCCR2, R16             ; с предделителем на 1024

wait:   in R16, ASSR          ; ждем окончания записи
        cpi R16, 1<<AS2
        brne wait

        ldi R16, 1<<TOV2     ; на всякий случай сбрасываем
        out TIFR, R16        ; флаг требования прерывания
        out TIMSK, R16       ; и разрешаем прерывание по
                                ; переполнению таймера 2

```

```

ldi R16, 0b01110000; разрешаем Power Down режим
out MCUCR, R16
sei ; разрешаем глобальное
; прерывание
sleep ; входим в Power Down режим
cli ; запрещаем прерывания

tim2: reti ; пустая процедура прерывания

```

2.8. Сторожевой таймер

Сторожевой таймер работает от отдельного встроенного генератора, работающего на частоте 1 MHz (типичное значение при питании 5 В). Управляя предварительным делителем сторожевого таймера, можно задавать интервал сброса таймера от 16 до 2048 mS. Для сброса сторожевого таймера используется команда WDR. Регистр управления сторожевым таймером WDTCR позволяет задавать различные периоды работы таймера.

Регистр управления WDTCR (Watchdog Timer Control Register) имеет следующую структуру.

	7	6	5	4	3	2	1	0
WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0

- WDTOE (Watchdog Turn-Off Enable) – разрешение выключения сторожевого таймера. Установка этого бита в 1 разрешает выключение сторожевого таймера. Бит аппаратно сбрасывается в ноль по истечению четырех тактов после его установки.
- WDE (Watchdog Enable) – разрешение сторожевого таймера. Логическая единица в этом бите разрешает работу сторожевого таймера. Для его выключения необходимо сначала одной командой записать логические единицы в биты WDTOE и WDE, а затем в течение четырех тактов процессора сбросить бит WDE в ноль.
- WDP2, WDP1, WDP0 (Watchdog Prescaler) – биты определяющие коэффициент деления предварительного делителя. Зависимость времени срабатывания сторожевого таймера от состояния этих бит приведена ниже

WDP2	WDP1	WDP0	Период времени
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

2.9. EEPROM память

Память класса EEPROM является энергонезависимой памятью, однако, в отличие от flash-памяти программ, программа может менять ее содержимое во время своей работы. Следует заметить, что время записи в EEPROM-память достаточно велико и составляет единицы миллисекунд. EEPROM-память не отображается на адресное пространство микроконтроллера и доступна через регистры EEDR, EEAR и EECR, расположенные в области ввода/вывода.

Регистр EEDR (EEPROM Data Register) является регистром данных, через который осуществляется запись и чтение EEPROM-памяти.

Так как объем EEPROM памяти превышает 256 байт, регистр EEAR (EEPROM Address Register) доступен как пара регистров – EEARL – младший байт, и EEARH – старший байт. Структура этих регистров приведена ниже.

EEARH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	EEAR9

EEARL

7	6	5	4	3	2	1	0
EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0

Регистр EECR (EEPROM Control Register) содержит управляющие биты, позволяющие читать и записывать EEPROM-память.

	7	6	5	4	3	2	1	0
EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE

- EERIE (EEPROM Ready Interrupt Enable) – бит разрешения прерывания по готовности EEPROM к новой записи в памяти. Прерывание с вектором \$0F генерируется при следующем условии: EERIE-бит и I-бит регистра SREG установлены в единицу, а бит EEWЕ очищен.
- EEMWE (EEPROM Master Write Enable) – мастер-бит записи. Этот бит определяет, будут ли записаны данные при установке бита EEWЕ. Если бит EEMWE установлен – запись возможна. Данный бит сбрасывается аппаратно по истечению четырех тактов микроконтроллера после его установки.
- EEWЕ (EEPROM Write Enable) – разрешение записи. Логическая единица, записанная в этот бит, вызывает запись данных, помещенных в регистр EEDR, в ячейку памяти EEPROM с адресом, хранящимся в регистре EEAR. Бит очищается автоматически после того, как запись будет произведена.
- EERE (EEPROM Read Enable) – разрешение чтения. Логическая единица, записанная в этот бит, вызывает пересылку содержимого ячейки памяти с адресом, хранящимся в EEAR, в регистр EEDR. После этого бит автоматически очищается.

Последовательность записи данных в EEPROM память приведена в примере 9.

Пример 9.

Записать 10 байт данных, хранящихся в буфере ОЗУ, начиная с адреса \$60, в EEPROM-память, начиная с адреса \$10.

```

ldi    R30,    $60      ; подготавливаем указатель Z
ldi    R31,    0

clr    R17      ; обнуляем старший байт
out    EEARH, R17 ; EEPROM памяти

ldi    R17,    $10      ; запоминаем начальный
                        ; адрес EEPROM

loop:  sbic    EECR,    EEWЕ ; проверяем, завершена ли
      rjmp   loop      ; прошлая запись

out    EEARL, R17      ; загружаем адрес EEPROM
ld     R16,    z+      ; читаем данные из ОЗУ и
out    EEDR,   R16      ; записываем их в регистр данных

```

```

sbi  EECR,  EEMWE ; разрешаем запись в EEPROM
sbi  EECR,  EEWE  ; записываем данные
inc  R17      ; увеличиваем адрес EEPROM
cpi  R30,    $60+10 ; проверяем, все ли данные записаны
brne loop    ; если нет - цикл

```

2.10. Универсальный асинхронный приемо-передатчик

Микроконтроллер имеет в своем составе полный дуплексный последовательный приемо-передатчик UART (Universal Asynchronous Receiver and Transmitter). Его основными характеристиками являются:

- генерация произвольных значений скорости;
- высокая скорость при низких тактовых частотах;
- 8 или 9 бит данных;
- фильтрация шума;
- определение переполнения;
- детектирование ошибки кадра;
- определение неверного стартового бита;
- три отдельных прерывания - завершение передачи, очистка регистра передачи и завершение приема.

2.10.1. Передача данных

Передача данных инициируется записью передаваемых данных в буферный регистр ввода-вывода UART – UDR. Из буферного регистра данные перезаписываются в выходной сдвиговый регистр, который выдвигает их через вывод микроконтроллера TxD/PD1 младшим битом вперед. Кадр данных может содержать дополнительный 9-й бит данных. Кадр передаваемых данных обрамляется стартовым (логический 0) и стоповым (логическая 1) битами.

После копирования данных из буферного регистра в сдвиговый регистр, в статусном регистре UART устанавливается флаг, информирующий пользователя, что буфер передатчика способен принять следующий байт данных. После того, как последний байт данных был выдвинут сдвиговым регистром, в статусном регистре устанавливается флаг, свидетельствующий, что все данные были переданы адресату.

2.10.2. Прием данных

Прием последовательных данных производится через вывод RxD/PD0 микроконтроллера. Приемник UART ожидает перепад напряжения $1 \rightarrow 0$ на входе RxD. После того, как перепад обнаружен, передатчик включает свой тактовый генератор и выдерживает время, равное половине длительности одного бита. По прошествии этого интервала времени передатчик трижды измеряет напряжение на линии. Если хотя бы два результата измерения оказались нулевыми, принимается решение, что зафиксирован стартовый бит. Если было зарегистрировано две или более логические единицы, принимается решение, что перепад на входе RxD является следствием шумов на линии. В этом случае приемник переходит на ожидание следующего старт-бита.

В том случае, если был опознан правильный старт-бит, сдвиговый регистр приемника начинает принимать данные. Через время, равное длительности одного бита, производится троекратное измерение потенциала на линии. Решение о состоянии бита производится по мажоритарному принципу. После приема всего кадра данных производится контроль стоп-бита. Если его значение оказалось равным нулю, в статусном регистре UART выставляется флаг ошибки кадра.

После приема всего кадра, данные из сдвигового регистра переписываются в буферный регистр приемника, а в статусном регистре устанавливается флаг, сигнализирующий о принятом байте. До того как полученный байт будет вычитан из буфера, сдвиговый регистр способен принимать следующие данные. Однако если данные из буфера не будут вычитаны к моменту перезаписи байта из сдвигового регистра в буфер приемника, устанавливается соответствующий флаг в статусном регистре, сигнализируя о потере данных.

2.10.3. Управление приемо-передатчиком

Обратиться к буферу как приемника, так и передатчика можно через регистр UDR (UART Data Register). По сути это два различных регистра, имеющих один и тот же адрес. Во время операции чтения обращение производится к буферу приемника, а при записи – к буферу передатчика.

Биты, индицирующие состояние UART, сосредоточены в статусном регистре USR.

Регистр USR (UART Status Register) имеет структуру, приведенную ниже.

	7	6	5	4	3	2	1	0
USR	RXC	TXC	UDRE	FE	OR	-	-	-

- RXC (RX Complete) – прием завершен. Аппаратно устанавливается в “1” после перезаписи данных из сдвигового регистра в буфер приемника. Сбрасывается любой операцией чтения буферного регистра UDR. Бит доступен только для чтения.
- TXC (TX Complete) – передача завершена. Аппаратно устанавливается в “1”, если данные сдвигового регистра (включая стоп-бит) были переданы, а в регистр UDR новые данные записаны не были. Сбрасывается аппаратно при исполнении соответствующего прерывания, или программно, путем записи в этот бит логической “1”.
- UDRE (UART Data Register Empty) – буферный регистр передатчика пуст. Устанавливается аппаратно в момент перезаписи данных из буфера в сдвиговый регистр передатчика. Сбрасывается в ноль программно при записи новых данных в буфер передатчика.
- FE (Frame Error) - ошибка кадра. Аппаратно устанавливается в “1”, если в качестве стоп-бита был принят логический “0”. В противном случае сбрасывается. Доступен только для чтения.
- OR (OverRun) - переполнение. Аппаратно устанавливается в том случае, если данные не были прочитаны из буфера приемника к моменту приема следующего байта сдвиговым регистром. Сбрасывается аппаратно при чтении буфера приемника. Доступен только для чтения.

Биты управления UART сосредоточены в регистре UCR.

Регистр UCR (UART Control Register) имеет структуру, приведенную ниже.

	7	6	5	4	3	2	1	0
UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8

- RXCIE (RX Complete Interrupt Enable) – разрешение прерывания по приему байта. Если бит установлен в “1” и глобально прерывания разрешены, – установка бита RXC регистра USR вызовет прерывание.
- TXCIE (TX Complete Interrupt Enable) - разрешение прерывания по концу передачи. Установка этого бита в “1” разрешает прерывание от флага TXC регистра USR.
- UDRIE (UART Data Register empty Interrupt Enable) - разрешение прерывания по очистке буферного регистра. Установка этого бита в “1” разрешает прерывание от флага UDRE регистра USR.

- RXEN (RX Enable) - разрешение приемника. Если бит установлен, вывод RXD/PD0 микроконтроллера функционирует как вход приемника UART - RXD, в противном случае – как линия порта ввода-вывода PD0.
- TXEN (TX Enable) - разрешение приемника. Если бит установлен, вывод TXD/PD1 микроконтроллера функционирует как выход передатчика UART - RXD, в противном случае – как линия порта ввода-вывода PD1.
- CHR9 (Character 9) - 9-й символ. Если этот бит установлен, принимаемые и передаваемые символы имеют длину 9 бит, в противном случае – 8.
- RXB8 (RX bit 8) - 8-й бит приемника. В случае, если используется 9-битный формат, содержит 8-й бит принятого кадра. Бит доступен только для чтения.
- TXB8 (TX bit 8) - 8-й бит передатчика. В случае, если используется 9-битный формат, содержимое этого бита подставляется в передаваемый кадр в качестве дополнительного бита.

2.10.4. Скорость передачи.

Скорость приема и передачи зависит от тактовой частоты микроконтроллера и от содержимого 8-битного регистра UBRR (UART Baud Rate Register) и рассчитывается по формуле

$$\text{Baud} = F_{\text{ck}} / 16 / (\text{UBRR} + 1),$$

где F_{ck} – тактовая частота микроконтроллера;
 UBRR – содержимое регистра UBRR.

Пример 10.

Из входного потока данных выбрать 100 байт, содержащих нулевой 9-й бит и записать их в ОЗУ, начиная с адреса \$0A0. Тактовая частота микроконтроллера – 3.6864 MHz, скорость передачи 19.2 кбит/сек.

Рассчитаем код, который должен содержать UBRR

$$\text{UBRR} = 3686.4 / 16 / 19.2 = 12$$

```
ldi    R16, 12           ; задаем скорость передачи
out    UBRR, R16
```

```

ldi    R16,    0b00010100    ; прием разрешен,
out    UCR,    R16            ; 9-битный формат

ldi    R30,    $a0            ; Настроить указатель Z
ldi    R31,    0              ; на адрес $0A0

loop:  sbis    USR,    RXC      ; ждем прихода байта данных
      rjmp    loop
      in      R16,    UDR      ; читаем данные
      sbic    UCR,    RXB8     ; 9-й бит равен 0?
      rjmp    loop
      st      Z+,    R16       ; сохраняем данные в ОЗУ
      cpi    R30,    low($A0+100) ; все данные приняты?
      brne    loop            ; если нет - на начало цикла

```

2.11. Последовательный периферийный интерфейс

Последовательный периферийный интерфейс SPI (Serial Peripheral Interface) обеспечивает высокоскоростной синхронный обмен данными между микроконтроллером и другими внешними устройствами.

SPI AVR-микроконтроллеров обладает следующими характеристиками:

- полнодуплексная, трехпроводная схема передача данных;
- операции в режиме master или slave;
- передача данных старшим или младшим битами вперед;
- четыре программируемые скорости передачи;
- прерывание по завершению обмена.

Интерфейс обеспечивается тремя основными сигналами: MISO (Master Input/Slave Output), MOSI (Master Output/Slave Input) и SCK (Serial Clock).

В обмене данными всегда принимают участие два устройства – master и slave. Одноименные сигналы master и slave соединяются вместе.

Обмен данными начинается после того, как в регистр передатчика мастера были записаны данные. Мастер начинает генерировать импульсы синхронизации на линии SCK и выводить данные бит за битом по линии MOSI. Кроме того, мастер под каждый синхроимпульс принимает данные, присутствующие на линии MISO, вдвигая их в буфер приемника.

Устройство, сконфигурированное как slave, используя тактовые импульсы, генерируемые мастером, принимает данные от мастера, одновременно передавая данные своего буфера. Таким образом, после окончания передачи, данные в буферах master и slave меняются местами.

Последовательный интерфейс AVR-микроконтроллеров имеет дополнительный сигнал SS (Slave Select), совмещенный с четвертой линией порта В. Если интерфейс назначен мастером, пользователь может определить направление данных на выводе SS/PB4. Если данный вывод через регистр DDRB назначен выходом, то сигнал SS ни коим образом не влияет на работу SPI. Если вывод SS был конфигурирован как вход, на него следует подать потенциал логической единицы для подтверждения, что данный SPI находится в режиме master. Если в режиме master сигнал SS сконфигурирован как вход, но какое-либо внешнее устройство переключает потенциал на нем, задавая уровень логического нуля, SPI система интерпретирует это так, что другое устройство хочет стать мастером. В ответ на это SPI автоматически переводится в slave-режим, переключая выходы MOSI и SCK на ввод.

Если устройство сконфигурировано как slave – вывод SS обязательно становится входом. Для обеспечения нормальной работы устройства в slave-режиме необходимо поддерживать на этом выводе сигнал низкого уровня. В противном случае устройство выключается вовсе: не принимает входные данные по тактам синхронизации, выход MISO переводится в третье состояние.

Доступ к данным, принимаемым и передаваемым SPI, можно получить через регистр SPDR (Serial Port Data Register).

Биты, индицирующие состояние SPI, находятся в статусном регистре SPSR (Serial Peripheral Status Register). Регистр содержит два значащих бита и доступен только для чтения. Его структура приведена ниже.

	7	6	5	4	3	2	1	0
SPSR	SPIF	WCOL	-	-	-	-	-	-

- SPIF (SPI Interrupt Flag) – флаг прерывания. Устанавливается по окончании обмена данными. Если вывод SS конфигурирован как вход, а SPI находится в master-режиме, то низкий потенциал на входе SS также устанавливает данный флаг. Флаг сбрасывается автоматически, если берется соответствующее прерывание, либо программно при обращении к регистру данных SPDR, следующим за прочтением SPSR с установленным SPIF.
- WCOL (Write Collision flag) – флаг коллизии записи. Устанавливается при попытке записи данных в регистр SPDR до завершения предыдущего обмена. Сбрасывается при доступе к регистру SPDR, следующим за прочтением SPSR с установленным WCOL.

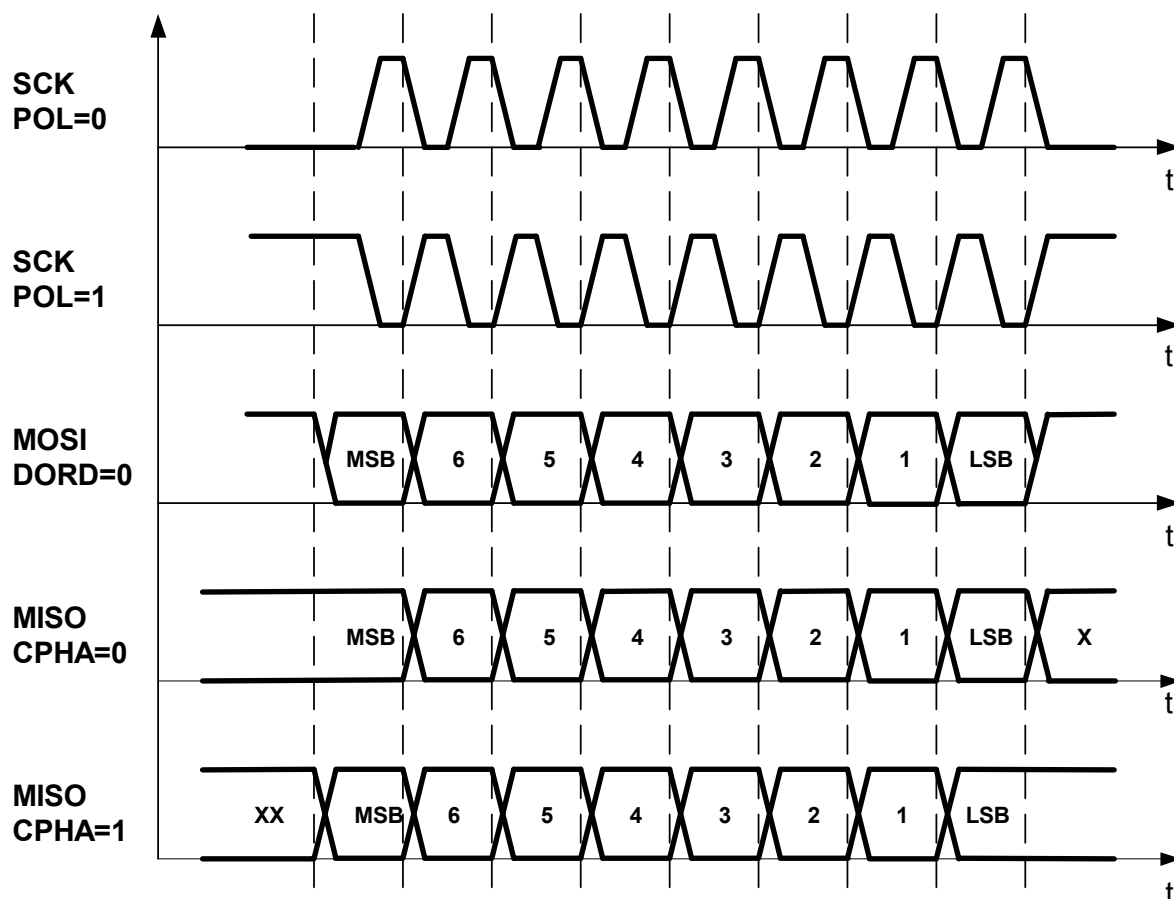
Биты управления SPI сосредоточены в регистре SPCR (Serial Peripheral Control Register). Структура SPCR представлена ниже.

	7	6	5	4	3	2	1	0
SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

- SPIE (SPI Interrupt Enable) – разрешение прерывания. При установленном бите SPIE и, в случае глобального разрешения прерывания, установка бита SPIF вызывает соответствующее прерывание.
- SPE (SPI Enable) – разрешение SPI. Если данный бит установлен, SPI разрешен и выводы микроконтроллера MOSI/PB5, MISO/PB6 и SCK/PB7 работают как сигналы интерфейса. В противном случае, они могут использоваться пользовательского ввода/вывода.
- DORD (Data Order) – порядок передачи данных. Если данный бит установлен, данные передаются младшим битом вперед. В противном случае – старшим.
- MSTR (Master) – выбор master-режима. Установка этого бита задает master-режим интерфейса. Сброс этого бита задает slave-режим. Если интерфейс конфигурирован как мастер, использующий вывод SS, то переключение SS в состояние логического нуля автоматически сбрасывает этот бит.
- CPOL (Clock Polarity) – полярность тактирования. Влияние этого бита на характеристики интерфейса показано на рисунке 3.
- CPHA (Clock Phase) – фаза тактирования. Влияние этого бита на характеристики интерфейса показано на рисунке 3.
- SPR1, SPR0 (SPI Rate) – выбор скорости передачи данных. Зависимость скорости передачи от состояния этих бит приведена ниже

SPR1	SPR0	Частота тактирования
0	0	$F_{cl} / 4$
0	1	$F_{cl} / 16$
1	0	$F_{cl} / 64$
1	1	$F_{cl} / 128$

где F_{cl} – частота тактирования микроконтроллера.



x – старший бит только что принятого символа;
 xx – младший бит только что переданного символа.

Рис. 3. Временная диаграмма работы SPI.

2.12. Аналого-цифровой преобразователь

Микроконтроллер AT90S8535 содержит в своем составе 10-разрядный 8-канальный аналого-цифровой преобразователь. АЦП имеет встроенное устройство выборки-хранения (УВХ). Питание АЦП осуществляется с помощью выделенных выводов – AGND и AVCC. Напряжение на выводе AVCC не должно отличаться от основного питания более чем на 0.3 В. АЦП не имеет встроенного источника опорного напряжения. Поэтому на вывод Aref необходимо подать стабильное опорное напряжение в диапазоне от AGND до AVCC. Время

преобразования АЦП программируется и должно находиться в пределах 65 – 260 μ s.

АЦП может функционировать в двух различных режимах – однократного измерения и непрерывного измерения. В режиме однократного измерения на весь цикл преобразования тратится 14 внутренних тактов АЦП, для непрерывного преобразования требуется 13 тактов. В режиме однократного измерения каждое преобразование должно инициироваться пользователем. В непрерывном режиме АЦП постоянно измеряет входное напряжение и автоматически заменяет выходной результирующий код.

Выходной код АЦП сохраняется в регистре результата ADC. Так как разрядность АЦП составляет 10 бит, доступ к регистру результата осуществляется через регистровую пару – ADCL – младший байт, и ADCH – старший байт. Эти регистры доступны только для чтения. Вычитывать эти регистры необходимо в порядке, принятом для всех регистровых пар: сначала читается младший байт, потом старший.

АЦП содержит 8-ми канальный аналоговый мультиплексор, входы которого совмещены с портом А. С помощью регистра ADMUX (ADC Multiplexer), содержащего три младших значащих бита, производится выбор входного канала. Код, записываемый в регистр ADMUX, соответствует индексу выбранной линии порта А – PA0..PA7.

Регистр ADCSR (ADC Control and Status Register) содержит управляющие биты, и биты индицирующие состояние АЦП. Его структура приведена ниже.

	7	6	5	4	3	2	1	0
ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

- ADEN (ADC Enable) – разрешение АЦП. Если бит установлен – АЦП разрешен. Выключение АЦП приводит к сокращению тока потребления.
- ADSC (ADC Start Conversion) – старт преобразования. Запись логической “1” в этот бит вызывает начало преобразования. Логическая единица удерживается в этом бите на все время преобразования. После окончания преобразования этот бит аппаратно сбрасывается.
- ADFR (ADC Free Run) – АЦП в режиме непрерывного измерения. Установка этого бита вызывает перевод АЦП в режим непрерывного измерения. Однажды запущенное установкой бита ADSC преобразование будет непрерывно повторяться, с постоянной заменой выходных данных в регистрах ADCL, ADCH. Запись логического нуля в этот бит завершит режим непрерывного преобразования.

- ADIF (ADC Interrupt Flag) – флаг прерывания. Аппаратно устанавливается в “1” после того, как преобразование завершилось и результат преобразования был записан в регистр результата ADC. Сбрасывается аппаратно при вхождении в прерывание или программно, путем записи в этот бит логической “1”.
- ADIE (ADC Interrupt Enable) – разрешение прерывания. Установка этого бита разрешает соответствующее прерывание, логический “0” – запрещает его.
- ADPS2, ADPS1, ADPS0 (ADC Prescaler Select) – выбор делителя частоты для работы АЦП. Тактовая частота АЦП должна находиться в пределах 50 – 200 kHz. При слишком высокой частоте могут не сработать аналого-цифровые схемы устройства. Чрезмерное снижение частоты приведет к увеличению ошибки измерения из-за разряда запоминающего конденсатора UBX во время преобразования. Выбор коэффициента деления позволяет подобрать нужную частоту тактирования АЦП в зависимости от собственной частоты микроконтроллера. Соответствие состояния этих бит коэффициентам деления частоты приведены ниже

ADPS2	ADPS1	ADPS0	Коэффициент деления
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Если АЦП находится в режиме непрерывного преобразования, следующее преобразование начинается сразу же после установки флага ADIF. Преобразование начинается с подключения канала, номер которого находится в регистре ADMUX, на вход UBX и выборки мгновенного значения входного напряжения. Таким образом, в ответ на установку флага требования прерывания пользователь может изменить содержимое регистра ADMUX. При этом канала с выбранным номером будет обслужен в следующем цикле преобразования.

Для улучшения точности измерения входного напряжения в АЦП предусмотрен специальный метод измерения в малопотребляющем, и

следовательно, малому режиму, при котором производится одновременный запуск измерения и перевод микроконтроллера в Idle-режим. Для этого необходимо выполнить следующую последовательность действий.

- 1) Убедиться, что АЦП включен и не занят текущим циклом преобразования, то есть $ADEN = 1$, $ADSC = 0$.
- 2) Выбрать режим однократного преобразования и разрешить прерывания, то есть $ADFR = 0$, $ADIE = 1$.
- 3) Войти в Idle-режим пониженного потребления, что вызовет автоматический старт АЦП.

Прерывание, генерируемое по завершению преобразования, выведет процессор из Idle-режима.

Пример 11.

Используя режим непрерывного преобразования, оцифровать все 8 входных каналов, результаты измерения записать в память, начиная с адреса \$60. Тактовая частота процессора 4MHz.

Коэффициент деления частоты должен быть 32, так как $4000 \text{ kHz} / 32 = 125 \text{ kHz}$.

```

ldi    R30,    $60           ; настраиваем указатель Z
ldi    R31,    0
ldi    R16,    0b10111101    ; АЦП включить, Free Run – режим,
out    ADCSR, R16           ; коэффициент деления 32
clr    R17           ; установить нулевой канал
out    ADMUX, R17
sbi    ADCSR, ADSC           ; запустить АЦП
loop:  inc    r17           ; установить следующий канал
out    ADMUX, R17
sbis   ADCSR, ADIF           ; готово ?
rjmp   pc-1           ; переход на предыдущую команду
sbi    ADCSR, ADIF           ; очищаем флаг
in     R16,     ADCL           ; читаем и сохраняем
st     Z+,     R16           ; младший байт

in     R16,     ADCH           ; читаем и сохраняем
st     Z+,     R16           ; старший байт

cpi    R30,     $60+7         ; седьмое измерение?
brne   free

```



```

        cbi   ADCSR,  ADFR           ; выключить режим Free Run
free:  cpi    R31,     $60+8         ; последнее измерение,
        brne  loop                  ; если нет – цикл.

```

Пример 12.

Используя малозащумляющий Idle-режим, измерить напряжение текущего канала. Данные сохранить в регистрах R1:R0. Тактовая частота микроконтроллера 4 MHz.

```

ldi    R16,      0b10011101 ; режим однократного измерения,
out    ADCSR,    R16         ; прерывания разрешены
sei                                ; глобальное разрешение прерывания
ldi    R16,      0b01000000 ; разрешение Idle-режима
out    MCUCR,    R16
sleep                                ; войти в Idle-режим
in     R0,       ADCL         ; вычитать младший байт
in     R1,       ADCH         ; вычитать старший байт

```

2.13. Аналоговый компаратор

Микроконтроллер AT90S8535 имеет в своем составе аналоговый компаратор. Неинвертирующий вход компаратора совмещен с портом PB2, а инвертирующий – с портом PB3. Работа аналогового компаратора поддерживается одним регистром ACSR. Регистр ACSR (Analog Comparator Control and Status Register) содержит следующие биты.

	7	6	5	4	3	2	1	0
ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0

- ACD (Analog Comparator Disable) – запрещение компаратора. Установка этого бита выключает компаратор. Это приводит к снижению потребления микроконтроллера.
- ACO (Analog Comparator Output) – выход компаратора. Этот бит соединен с аппаратным выходом компаратора. Доступен только для чтения.
- ACI (Analog Comparator Interrupt flag) – флаг требования прерывания. Устанавливается аппаратно при срабатывании компаратора. Сбрасывается аппаратно при вхождении в прерывание, или программно, при записи в этот бит логической “1”.

- ACIE (Analog Comparator Interrupt Enable) – разрешение прерывания. Установка этого бита разрешает прерывания аналогового компаратора.
- ACIC (Analog Comparator Input Capture enable) – установка данного бита разрешает захват текущего значения таймера1 (содержимого регистра TCNT1) в регистр захвата ICP1 при срабатывании компаратора.
- ACIS1, ACIS0 (Analog Comparator Interrupt mode Select) – выбор режима прерывания. Соответствие состояния этих бит режиму прерывания аналогового компаратора приведено ниже

ACIS1	ACIS0	Режим прерывания
0	0	Прерывания при любом переключении
0	1	Зарезервировано
1	0	Прерывание по спадающему переходу
1	1	Прерывание по нарастающему переходу

При переключении этих бит необходимо запретить прерывание, так как это может привести к его несанкционированной генерации.

Библиографический список

1. AVR. Atmel Corporation 8-Bit RISC microcontrollers. Data Book. –1997.
2. www.atmel.com.

ПРИЛОЖЕНИЕ А

Регистры периферийных устройств микроконтроллера AT90S8535

Адрес	Имя	Назначение регистра	Стр.
\$3F/\$5F	SREG	Регистр флагов процессора	6
\$3E/\$5E	SPH	Указатель стека	6
\$3D/\$5D	SPL		
\$3B/\$5B	GIMSK	Маска внешних прерываний	10
\$3A/\$5A	GIFR	Флаги внешних прерываний	10
\$39/\$59	TIMSK	Маска прерывания таймеров	14
\$38/\$58	TIFR	Флаги прерывания таймеров	15
\$35/\$54	MCUCR	Управление микроконтроллера	11
\$34/\$54	MCUSR	Статус микроконтроллера	9
\$33/\$53	TCCR0	Управление таймера 0	16
\$32/\$52	TCNT0	Таймер 0	16
\$2F/\$4F	TCCR1A	Управление А таймера 1	18
\$2E/\$4E	TCCR1B	Управление В таймера 1	19
\$2D/\$4D	TCNT1H	Таймер 1	17
\$2C/\$4C	TCNT1L		
\$2B/\$4B	OCR1AH	Регистр сравнения А таймера 1	17
\$2A/\$4A	OCR1AL		
\$29/\$49	OCR1BH	Регистр сравнения В таймера 1	17
\$28/\$48	OCR1BL		
\$27/\$47	ICR1H	Регистр захвата таймера 1	17
\$26/\$46	ICR1L		
\$25/\$45	TCCR2	Управление таймера 2	23
\$24/\$44	TCNT2	Таймер 2	23
\$23/\$43	OCR2	Регистр сравнения таймера 2	24
\$22/\$42	ASSR	Статус асинхронного режима таймера 2	24
\$21/\$41	WDTCR	Управление сторожевым таймером	26
\$1F/\$3F	EEARH	Регистр адреса EEPROM памяти	27
\$1E/\$3E	EEARL		
\$1D/\$3D	EEDR	Регистр данных EEPROM памяти	27
\$1C/\$3C	EECR	Регистр управления EEPROM памяти	27
\$1B/\$3B	PORTA	Управление портом А	13
\$1A/\$3A	DDRA	Направление порта А	13
\$19/\$39	PINA	Состояние порта А	13
\$18/\$38	PORTB	Управление портом В	13
\$17/\$37	DDRB	Направление порта В	13
\$16/\$36	PINB	Состояние порта В	13
\$15/\$35	PORTC	Управление портом С	13

\$14/\$34	DDRC	Направление порта C	13
\$13/\$33	PINC	Состояние порта C	13
\$12/\$32	PORTD	Управление портом D	13
\$11/\$31	DDRD	Направление порта D	13
\$10/\$30	PIND	Состояние порта D	13
\$0F/\$2F	SPDR	Регистр данных SPI	34
\$0E/\$2E	SPSR	Регистр статуса SPI	34
\$0D/\$2D	SPCR	Управление SPI	35
\$0C/\$2C	UDR	Регистр данных UART	30
\$0B/\$2B	USR	Статус UART	30
\$0A/\$2A	UCR	Управление UART	31
\$09/\$29	UBRR	Регистр задания скорости UART	32
\$08/\$28	ACSR	Управление и статус аналогового компаратора	40
\$07/\$27	ADMUX	Управление мультиплексором АЦП	37
\$06/\$26	ADCSR	Управление и статус АЦП	37
\$05/\$25	ADCH	Регистр результата преобразования АЦП	37
\$04/\$24	ADCL		