

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение высшего**  
**профессионального образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**  
**Филиал в г. Пятигорске**  
**Кафедра управления в технических и биомедицинских системах**

Методические указания  
к выполнению лабораторных работ по курсу  
**«Микроконтроллеры и микропроцессоры в системах управления»**  
Направление подготовки 10.03.01 Информационная безопасность  
Квалификация выпускника **бакалавр**

г. Пятигорск, 2020

Рассмотрено и утверждено на заседании кафедры управления в технических и биомедицинских системах протокол № \_\_\_\_ от \_\_\_\_\_ 202\_

Зав.кафедрой СУиИТ \_\_\_\_\_ И.М. Першин

## Оглавление

Введение

Лабораторная работа 1.

Изучение конструкции и состава микроконтроллера AVR, изучение организации систем микроконтроллера AVR 6

Лабораторная работа 2. Изучение системы команд микроконтроллера AVR 20

Лабораторная работа 3. Изучение работы и программирования портов микроконтроллера AVR 41

Лабораторная работа 4 Изучение работы и программирования блоков ШИМ микроконтроллера AVR изучение работы и программирования блоков последовательной передачи данных микроконтроллера AVR 54

Лабораторная работа 5. Инструментальные средства разработки программного обеспечения ПЛК. «Algorithm Builder» инструментальные средства разработки программного обеспечения ПЛК. «Assembler» 65

Лабораторная работа 6. Разработка программ работы с дискретными сигналами микроконтроллера AVR Разработка программ работы с аналоговыми сигналами микроконтроллера AVR 71

Лабораторная работа 7. Разработка программ работы с устройствами импульсной модуляции микроконтроллера AVR Разработка программ вывода информации на ЖКИ микроконтроллера AVR 79

Лабораторная работа 8. Разработка программ для связи микроконтроллера AVR с ПК-ЭВМ Разработка программ для связи микроконтроллера AVR как регулятора 89

## Введение

Методические указания предназначены для студентов. В лабораторный практикум по дисциплине Микроконтроллеры и микропроцессоры в системах управления включены лабораторные работы по основным разделам этой дисциплины, читаемой на кафедре «Управление в технических и биомедицинских системах». Лабораторные работы ориентированы на приобретение навыков студентами построения систем управления с использованием микроконтроллеров и микропроцессоров. Лабораторный курс ориентирован на получение практических навыков по работе с программируемыми микроконтроллерами. В работах используется лабораторный комплекс РТМЛ, который использует современный микроконтроллерный комплект AVR серии Atmega8535. В работах используются инструментальные средства разработки программного обеспечения «Algorithm Builder» и «Assembler». Приведен список рекомендуемой литературы. В приложении дан необходимый справочный материал.

Содержащиеся в практикуме сведения теории, методические указания и рекомендации по выполнению лабораторных работ позволяют использовать его в качестве дополнительного пособия для закрепления курса лекций.

Целью данного лабораторного практикума является поэтапное формирование у студентов знаний, умений и навыков создания систем управления с использованием микроконтроллеров и микропроцессоров.

Практикум предназначен для студентов Северо-Кавказского федерального университета и может быть полезным для всех желающих ознакомиться с основами компьютерной графики.

Данный вид работы играет важную роль в формировании практических навыков работы с графической информацией и способствует формированию следующих образовательных компетенций:

# Лабораторная работа №1

## Изучение конструкции, организации и состава системы микроконтроллера AVR

### Цель работы

Изучение конструкции и архитектуры микроконтроллера AVR Atmega8535, Atmega16, Atmega32 изучить основные приёмы программирования и отладки микроконтроллера.

### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контролеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

### Теоретическая часть

#### **Микроконтроллеры.**

**Вопрос 1** Конструкция МК микроконтроллера Микроконтроллер (англ. Micro Controller Unit, MCU) — микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ. По сути, это однокристалльный компьютер, способный выполнять простые задачи.

С появлением однокристалльных микро-ЭВМ связывают начало эры массового применения компьютерной автоматизации в области управления. Повидимому, это обстоятельство и определило термин «контроллер» (англ. controller— регулятор, управляющее устройство).

В связи со спадом отечественного производства и возросшим импортом техники, в том числе вычислительной, термин «микроконтроллер» (МК) вытеснил из употребления ранее использовавшийся термин «однокристалльная микро-ЭВМ».

Первый патент на однокристалльную микро-ЭВМ был выдан в 1971 году инженерам М. Кочрену и Г. Буну, сотрудникам американской Texas Instruments.

Именно они предложили на одном кристалле разместить не только процессор, но и память с устройствами ввода-вывода.

В 1976 году американская фирма Intel выпускает микроконтроллер i8048. В 1978 году фирма Motorola выпустила свой первый микроконтроллер MC6801, совместимый по системе команд с выпущенным ранее микропроцессором MC6800. Через 4 года, в 1980 году, Intel выпускает следующий микроконтроллер: i8051. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер i8051 являлся для своего времени очень сложным изделием — в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре i8086.

На сегодняшний день существует более 200 модификаций микроконтроллеров, совместимых с i8051, выпускаемых двумя десятками компаний, и большое количество микроконтроллеров других типов.

Популярность у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel, 16-битные MSP430 фирмы TI, а также 32-битные микроконтроллеры, архитектуры ARM, которую разрабатывает фирма ARM Limited и продаёт лицензии другим фирмам для их производства. Несмотря на популярность в России микроконтроллеров упомянутых выше, по данным Gartner Group от 2009 года мировой рейтинг по объёму продаж выглядит иначе: первое место с большим отрывом занимает Renesas Electronics на втором Freescale, на третьем Samsung, затем идут Microchip и TI, далее все остальные.

В СССР велись разработки оригинальных микроконтроллеров, также осваивался выпуск клонов наиболее удачных зарубежных образцов. В 1979 году в СССР НИИ ТТ разработали однокристалльную 16-разрядную ЭВМ К1801BE1, микроархитектура которой называлась «Электроника НЦ».

При проектировании микроконтроллеров приходится соблюдать баланс между размерами и стоимостью с одной стороны и гибкостью и производительностью с другой. Для разных приложений оптимальное соотношение этих и других параметров может различаться очень сильно.

Поэтому существует огромное количество типов микроконтроллеров, отличающихся архитектурой процессорного модуля, размером и типом встроенной памяти, набором периферийных устройств, типом корпуса и т. д. В отличие от обычных компьютерных микропроцессоров, в микроконтроллерах часто используется гарвардская архитектура памяти, то есть раздельное хранение данных и команд в ОЗУ и ПЗУ соответственно.

Кроме ОЗУ, микроконтроллер может иметь встроенную энергонезависимую память для хранения программы и данных. Во многих контроллерах вообще нет шин для подключения внешней памяти. Наиболее дешёвые типы памяти допускают лишь однократную запись. Такие устройства подходят для массового производства в тех случаях, когда программа контроллера не будет обновляться. Другие модификации контроллеров обладают возможностью многократной перезаписи энергонезависимой памяти.

Неполный список периферии, которая может присутствовать в микроконтроллерах, включает в себя:

- универсальные цифровые порты, которые можно настраивать как на ввод, так и на вывод;
- различные интерфейсы ввода-вывода, такие как UART, I2C, SPI, CAN, USB, IEEE 1394, Ethernet;
- аналого-цифровые и цифро-аналоговые преобразователи;

- компараторы;
- широтно-импульсные модуляторы;
- таймеры;
- контроллеры бесколлекторных двигателей;
- контроллеры дисплеев и клавиатур;
- радиочастотные приемники и передатчики;
- массивы встроенной флеш-памяти;
- встроенный тактовый генератор и сторожевой таймер;

Ограничения по цене и энергопотреблению сдерживают также рост тактовой частоты контроллеров. Хотя производители стремятся обеспечить работу своих изделий на высоких частотах, они, в то же время, предоставляют заказчикам выбор, выпуская модификации, рассчитанные на разные частоты и напряжения питания. Во многих моделях микроконтроллеров используется статическая память для ОЗУ и внутренних регистров. Это даёт контроллеру возможность работать на меньших частотах и даже не терять данные при полной остановке тактового генератора. Часто предусмотрены различные режимы энергосбережения, в которых отключается часть периферийных устройств и вычислительный модуль.

Известные семейства

- MCS 51 (Intel)
- MSP430 (TI)
- ARM (ARM Limited)
- ST Microelectronics STM32 ARM-based MCUs
- Atmel Cortex, ARM7 и ARM9-based MCUs
- Texas Instruments Stellaris MCUs
- NXP ARM-based LPC MCUs
- Toshiba ARM-based MCUs
- Analog Devices ARM7-based MCUs
- Cirrus Logic ARM7-based MCUs
- Freescale Semiconductor ARM9-based MCUs
- AVR (Atmel)
- ATmega
- ATtiny
- XMega
- PIC (Microchip)
- STM8 (STMicroelectronics)

### **Применение**

Использование в современном микроконтроллере достаточно мощного вычислительного устройства с широкими возможностями, построенного на одной микросхеме вместо целого набора, значительно снижает размеры, энергопотребление и стоимость построенных на его базе устройств. Используются в управлении различными устройствами и их отдельными блоками:

- в вычислительной технике: материнские платы, контроллеры дисководов жестких и гибких дисков, CD и DVD, калькуляторах;
- электронике и разнообразных устройствах бытовой техники, в которой используется электронные системы управления —стиральных машинах, микроволновых печах, посудомоечных машинах, телефонах и современных приборах;

В промышленности:

- устройств промышленной автоматике —от программируемого реле и встраиваемых систем до ПЛК,
- систем управления станками

В то время как 8-разрядные процессоры общего назначения полностью

вытеснены более производительными моделями, 8-разрядные микроконтроллеры продолжают широко использоваться. Это объясняется тем, что существует большое количество применений, в которых не требуется высокая производительность, но важна низкая стоимость. В то же время, есть микроконтроллеры, обладающие большими вычислительными возможностями, например цифровые сигнальные процессоры.

### Микроконтроллеры семейства AVR.

AVR — семейство восьмибитных микроконтроллеров фирмы Atmel. Год разработки — 1996. В AVR микроконтроллерах есть область (4 байта) которую называют Fuse Bits (фьюз биты), в которой хранится конфигурация микроконтроллера. У каждого AVR микроконтроллера есть свой набор Fuse Bits.

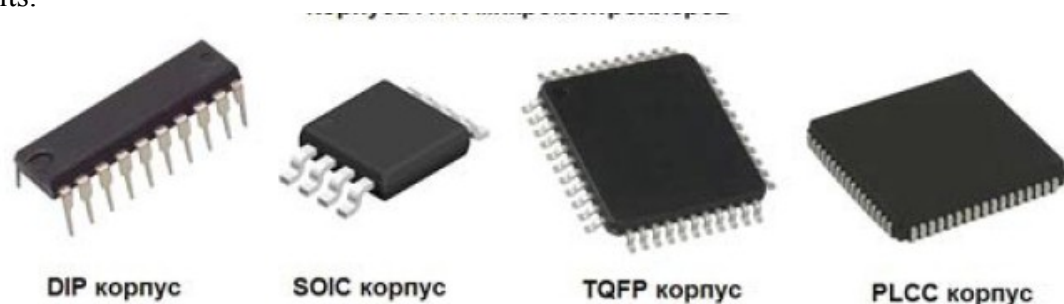


Рис. 1.1

Микроконтроллеры (МК) выпускаются в нескольких корпусах рис. 1.1. Большинство МК имеют SPI интерфейс связи, по которому можно программировать (прошивать) микроконтроллер.

Для AVR микроконтроллеров существует множество компиляторов, например: BASCOM-AVR (Basic компилятор), Code Vision AVR (C (си) компилятор), WinAVR (компилятор C (си) и ассемблера) и т.д. В данной задаче по микроконтроллерам AVR будет использоваться компилятор Algorithm Builder.

Для того чтобы запрограммировать (прошить) микроконтроллер, необходим программатор. Программатор это устройство, которое связывает микроконтроллер и компьютер вместе. Компьютер с помощью специальной программы прошивает микроконтроллер указанной программой (прошивкой).

Отличительные особенности. К особенностям микроконтроллеров AVR семейства Mega можно отнести:

- FLASH-память программ объемом от 8 до 256 Кбайт (число циклов стирания/записи не менее 10 000);
- оперативная память (статическое ОЗУ) объемом от 512 байт до 8 Кбайт;

- память данных на основе ЭСППЗУ (EEPROM) объемом от 256 байт до 4 Кбайт (число циклов стирания/записи не менее 100 000);

- возможность защиты от чтения и модификации памяти программ и данных;

- возможность программирования непосредственно в системе через последовательные интерфейсы SPI и JTAG;

- возможность самопрограммирования;

- возможность внутрисхемной отладки в соответствии со стандартом IEEE 1149.1 (JTAG), а также наличие собственного однопроводного интерфейса внутрисхемной отладки debugWire);

- разнообразные способы синхронизации: встроенный RC-генератор с внутренней или внешней времязадающей RC-цепочкой, встроенный генератор с внешним кварцевым или пьезокерамическим резонатором, внешний сигнал синхронизации;

- наличие нескольких режимов пониженного энергопотребления;



## Характеристики процессора

Основными характеристиками процессора микроконтроллеров AVR семейства Mega являются:

- полностью статическая архитектура, минимальная тактовая частота равна нулю;
- арифметико-логическое устройство (АЛУ) подключено непосредственно к регистрам общего назначения C2 регистра);
- большинство команд выполняются за один период тактового сигнала;
- векторная система прерываний, поддержка очереди прерываний;
- большое число источников прерываний (до 45 внутренних и до 32 внешних);
- наличие аппаратного умножителя.

## Периферийные устройства

Микроконтроллеры семейства Mega имеют богатый набор периферийных устройств (ПУ):

- один или два 8-битных таймера/счетчика. Во всех моделях с двумя 8-битными таймерами/счетчиками один из них может работать в качестве часов реального времени (в асинхронном режиме);
- от одного до четырех 16-битных таймеров/счетчиков;
- сторожевой таймер;
- одно- и двухканальные генераторы 8-битного ШИМ-сигнала (один из режимов работы 8-битных таймеров/счетчиков);
- двух- и трехканальные генераторы ШИМ-сигнала регулируемой разрядности (один из режимов работы 16-битных таймеров/счетчиков). Разрешение формируемого сигнала может составлять от 1 до 16 бит;
- аналоговый компаратор;
- многоканальный 10-битный АЦП последовательного приближения, имеющий как несимметричные, так и дифференциальные входы;
- последовательный синхронный интерфейс SPI;
- последовательный двухпроводный интерфейс TWI (полный аналог интерфейса I<sup>2</sup>C);
- от одного до четырех полнодуплексных универсальных синхронных/асинхронных приемо-передатчиков (USART). В ряде моделей эти приемо-передатчики могут использоваться в качестве ведущего устройства шины SPI;
- универсальный последовательный интерфейс USI, который может использоваться в качестве интерфейса SPI или I<sup>2</sup>C. Кроме того, USI может использоваться в качестве полудуплексного UART или 4/12-битного счетчика.

## Вопрос 2 Архитектура ядра

Ядро микроконтроллеров AVR семейства Mega выполнено по усовершенствованной RISC-архитектуре (enhanced RISC) (Рис. 1.2), в которой используется ряд решений, направленных на повышение быстродействия микроконтроллеров.

Арифметико-логическое устройство (АЛУ), выполняющее все вычисления, подключено непосредственно к 32 рабочим регистрам, объединенным в регистровый файл. Благодаря этому, АЛУ может выполнять одну операцию (чтение содержимого регистров, выполнение операции и запись результата обратно в регистровый файл) за

такт. Кроме того, практически каждая из команд (за исключением команд, у которых одним из операндов является 16-битный адрес) занимает одну ячейку памяти программ. В микроконтроллерах AVR реализована Гарвардская архитектура, характеризующаяся отдельной памятью программ и данных, каждая из которых имеет собственные шины доступа. Такая организация позволяет одновременно работать как с памятью программ, так и с памятью данных. Разделение информационных шин позволяет использовать для каждого типа памяти шины различной разрядности, причем способы адресации и доступа к каждому типу памяти также различаются. В сочетании с двухуровневым конвейером команд такая архитектура позволяет достичь производительности в 1 MIPS на каждый МГц тактовой частоты.

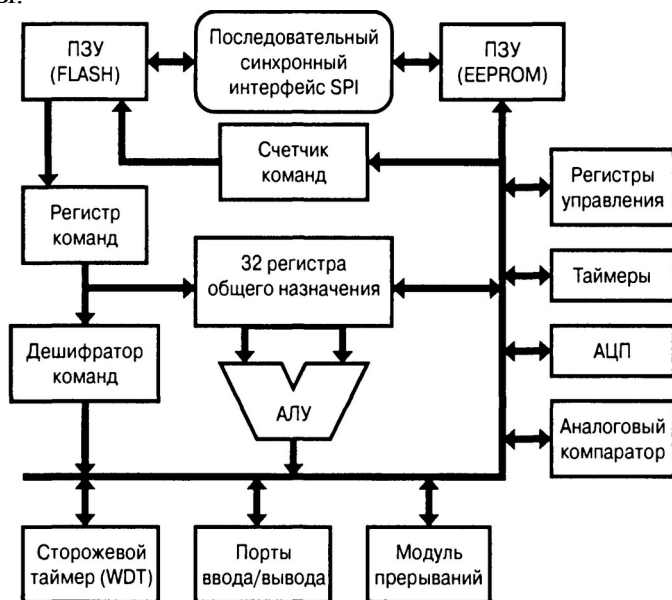


Рис. 1.2. Архитектура ядра микроконтроллеров

Микроконтроллеры ATmega8535, ATmega8535L (Рис. 1.3) — имеют FLASH-память программ объемом 8 Кбайт, ОЗУ объемом 512 байт и EEPROM-память данных объемом 512 байт. Выпускаются также в 44-выводном корпусе типа PLCC.

Описание выводов исследуемого микроконтроллера приведено в табл. 1.1.

Таблица 1.1. Описание выводов моделей ATmega8535x

Обозначение	Номер вывода			Тип вывода	Описание
	DIP	TQFP MLF	PLCC		
XTAL1	13	8	14	I	Вход тактового генератора
XTAL2	12	7	13	O	Выход тактового генератора
$\overline{\text{RESET}}$	9	4	10	I	Вход сброса
<b>Порт А. 8-битный двунаправленный порт ввода/вывода с внутренними подтягивающими резисторами</b>					
PA0 (ADC0)	40	37	43	I/O	0-й бит порта А Вход АЦП
PA1 (ADC1)	39	36	42	I/O	1-й бит порта А Вход АЦП
PA2 (ADC2)	38	35	41	I/O	2-й бит порта А Вход АЦП
PA3 (ADC3)	37	34	40	I/O	3-й бит порта А Вход АЦП
PA4 (ADC4)	36	33	39	I/O	4-й бит порта А Вход АЦП
PA5 (ADC5)	35	32	38	I/O	5-й бит порта А Вход АЦП
PA6 (ADC6)	34	31	37	I/O	6-й бит порта А Вход АЦП
PA7 (ADC7)	33	30	36	I/O	7-й бит порта А Вход АЦП

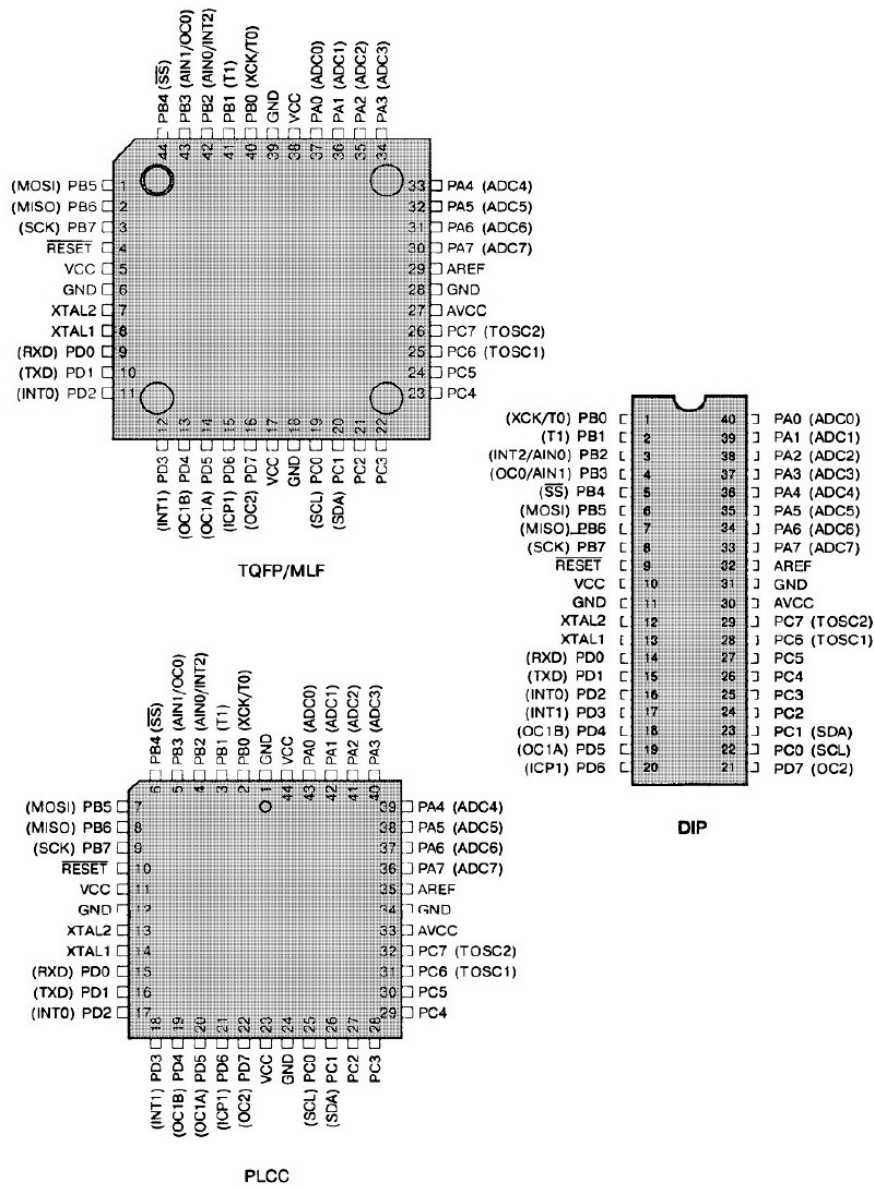


Рис. 1.3. Расположение выводов (вид сверху) моделей ATmega8535x

Таблица 1.1. Описание выводов моделей ATmega8535x

Продолжение в табл. 1.1.

Порт В. 8-битный двунаправленный порт ввода/вывода с внутренними подтягивающими резисторами					
PB0 (T0/XCK)	1	40	2	I/O	0-й бит порта В Вход внешнего тактового сигнала таймера/счетчика T0 Вход/выход внешнего тактового сигнала USART
PB1 (T1)	2	41	3	I/O	1-й бит порта В Вход внешнего тактового сигнала таймера/счетчика T1
PB2 (AIN0/INT2)	3	42	4	I/O	2-й бит порта В Неинвертирующий вход компаратора Вход внешнего прерывания
PB3 (AIN1/OC0)	4	43	5	I/O	3-й бит порта В Инвертирующий вход компаратора Выход таймера/счетчика T0

Продолжение в табл. 1.1.

Обозначение	Номер вывода			Тип вывода	Описание
	DIP	TQFP MLF	PLCC		
PB4 ( $\overline{SS}$ )	5	44	6	I/O	4-й бит порта В Выбор Slave-устройства на шине SPI
PB5 (MOSI)	6	1	7	I/O	5-й бит порта В Выход (Master) или вход (Slave) данных модуля SPI
PB6 (MISO)	7	2	8	I/O	6-й бит порта В Вход (Master) или выход (Slave) данных модуля SPI
PB7 (SCK)	8	3	9	I/O	7-й бит порта В Выход (Master) или вход (Slave) тактового сигнала модуля SPI
<b>Порт С. 8-битный двунаправленный порт ввода/вывода с внутренними подтягивающими резисторами</b>					
PC0 (SCL)	22	19	25	I/O	0-й бит порта С Вход/выход тактового сигнала модуля TWI
PC1 (SDA)	23	20	26	I/O	1-й бит порта С Вход/выход данных модуля TWI
PC2	24	21	27	I/O	2-й бит порта С
PC3	25	22	28	I/O	3-й бит порта С
PC4	26	23	29	I/O	4-й бит порта С
PC5	27	24	30	I/O	5-й бит порта С
PC6 (TOSC1)	28	25	31	I/O	6-й бит порта С Вывод для подключения резонатора к таймеру/счетчику T2
PC7 (TOSC2)	29	26	32	I/O	7-й бит порта С Вывод для подключения резонатора к таймеру/счетчику T2
<b>Порт D. 8-битный двунаправленный порт ввода/вывода с внутренними подтягивающими резисторами</b>					
PD0 (RXD)	14	9	15	I/O	0-й бит порта D Вход USART
PD1 (TXD)	15	10	16	I/O	1-й бит порта D Выход USART
PD2 (INT0)	16	11	17	I/O	2-й бит порта D Вход внешнего прерывания

Продолжение в табл. 1.1.

Обозначение	Номер вывода			Тип вывода	Описание
	DIP	TQFP MLF	PLCC		
PD3 (INT1)	17	12	18	I/O	3-й бит порта D Вход внешнего прерывания
PD4 (OC1B)	18	13	19	I/O	4-й бит порта D Выход В таймера/счетчика T1
PD5 (OC1A)	19	14	20	I/O	5-й бит порта D Выход А таймера/счетчика T1
PD6 (ICP1)	20	15	21	I/O	6-й бит порта D Вход захвата таймера/счетчика T1
PD7 (OC2)	21	16	22	I/O	7-й бит порта D Выход таймера/счетчика T2
AREF	32	29	35	P	Вход опорного напряжения для АЦП
AVCC	30	27	33	P	Вывод источника питания АЦП
VCC	10	5, 17, 38	11, 23, 44	P	Вывод источника питания
GND	11, 31	6, 18, 28, 39	12, 24, 34, 1	P	Общий вывод

### Оборудование и материалы.

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже

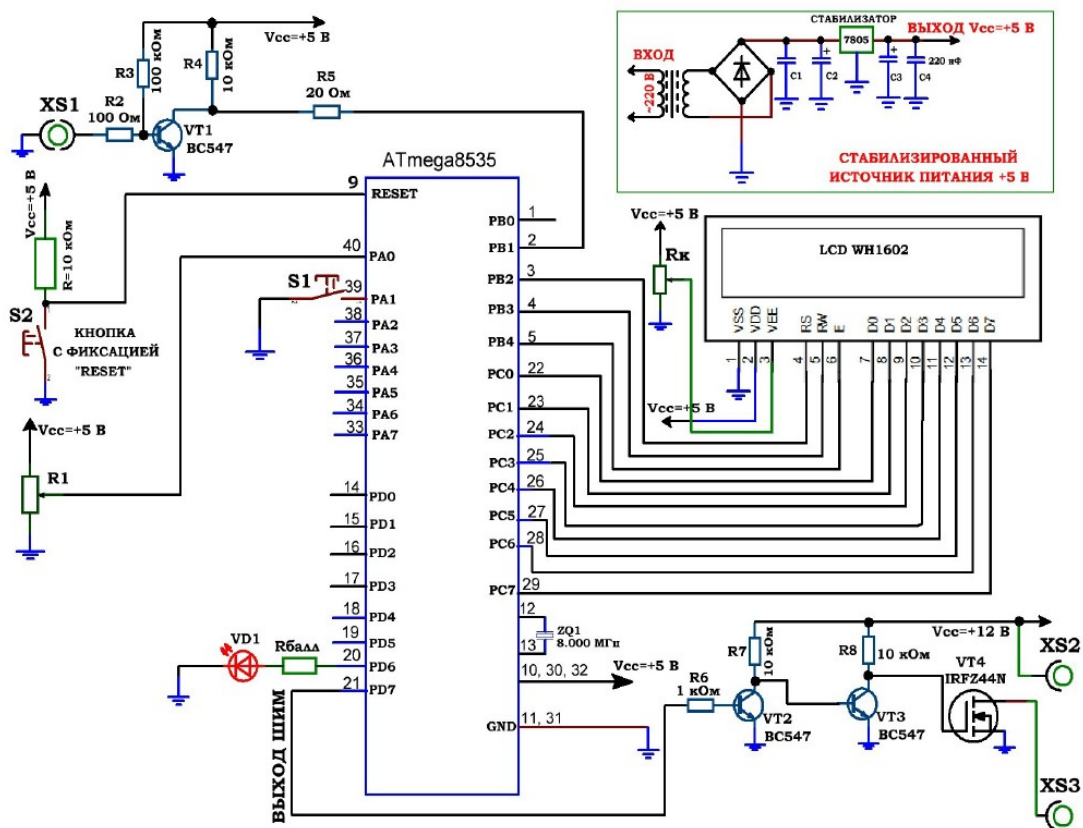


Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.



### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.
2. Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR, изучить все доступные паспорта на исследуемые микроконтроллеры, а также на подключенный LCD индикатор.
3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.
4. Загрузить операционную систему согласно стандартным процедурам загрузки.
5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder.
6. Запустить программу Algorithm Builder для работы с учебной установкой для данного эксперимента пользуясь ярлыком на рабочем столе либо другим способом, указанным лаборантом. **Для работы можно воспользоваться комплексной программной оболочкой LabVisual для РТМТЛ-1-5.**

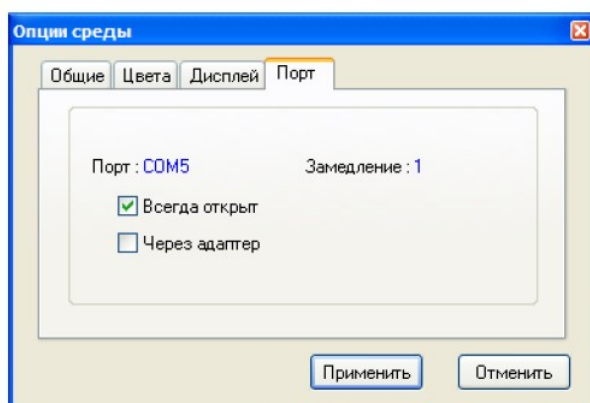


Рис. 4.1

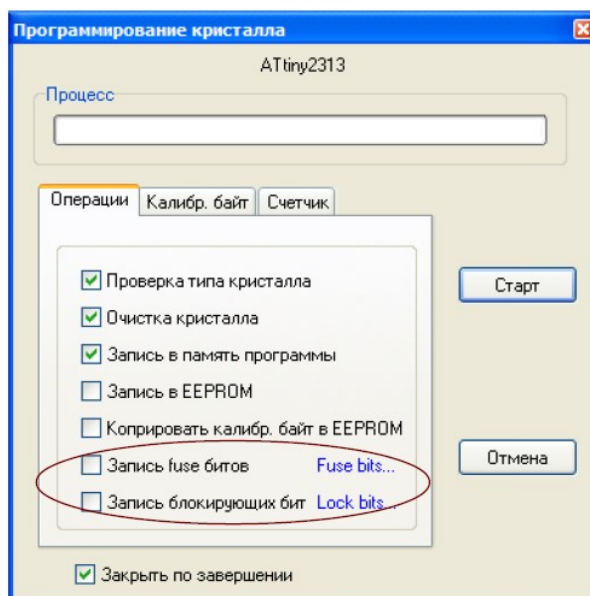


Рис. 4.2

г

6. Подключить разъём «ПРОГРАММАТОР» учебного прибора к COM – порту ПЭВМ проводом типа COM 9m/9f («мама» - «папа»)

7. Настроить программу Algorithm Builder для работы с данным COM портом ОПЦИИ-ОПЦИИ СРЕДЫ-ПОРТ рис. 4.1.

8. Для подробных инструкций следует обращаться к методическому руководству по среде Algorithm Builder.

9. Загрузить Пример 1 (папка 1) в среду Algorithm Builder (ФАЙЛ- ОТКРЫТЬ- ВЫБРАТЬ ФАЙЛ \*.alp).

10. «Прошить» тестовую программу в кристалл. Для этого нажать кнопку «ЗАПУСК С КРИСТАЛЛОМ», при этом откроется диалог программирования кристалла рис. 4.2. **Перед прошивкой следует нажать кнопку с фиксацией #RESET# на учебном приборе.**

11. Перед нажатием кнопки «СТАРТ» следует внимательно проверить положения флажков установки программатора рис. 4.2. **Особенно внимательно следует отнестись к опциям #ЗАПИСЬ Fuse bit# и #ЗАПИСЬ блокирующих бит#. ФЛАЖКИ ЭТИХ ОПЦИЙ ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ СНЯТЫ, Т. К. НЕ ПРАВИЛЬНО ПРОШИТЫЕ FUSE БИТЫ МОГУТ ПРИВЕСТИ К ДАЛЬНЕЙШЕЙ НЕВОЗМОЖНОСТИ РАБОТЫ С ДАННЫМИ КРОКОНТРОЛЛЕРОМ!**

12. **Отжать кнопку RESET и проверить работу тестовой программы.**

13. Соединить выход LINE IN звуковой карты со входом XS1 стенда проводом «тюльпан — jack» из комплекта.

14. Проверить работу программа-частотомера, подавая на вход XS1 прямоугольный сигнал звуковой частоты 20 — 9000 Гц. При подачи сигнала с линейного выхода LINE OUT ПК посредством программы «ГЕНЕРАТОР», следует установить амплитуду выходного сигнала генератора в программе не менее 80 % (рис. 3.2) и амплитуду выходного сигнала не менее 70 — 80 % непосредственно в звуковом драйвере (стандартная настройка громкости выходного сигнала). При низкой амплитуде входного сигнала, так же как и при слишком высокой амплитуде, частотомер может работать некорректно из за особенностей работы схемы однокаскадного транзисторного усилителя.

15. Загрузить пример 2 из соответствующей папки и прошить в кристалл тестовую программу.

16. Подключить к клеммам XS2 – XS3 нагрузку (вентилятор 12 В). **Обратить внимание, что вентилятор плюсовым красным выводом должен подключаться строго к клемме XS2 (соблюдать полярность).** Наблюдать работу тестовой программы.

17. Исходя из готовых примеров составить собственные программы в среде Algorithm Builder.

18. На основании Примера 1 можно составить программы, изменив, например, максимальное значение измеряемой частоты (в примере 9999 Гц), настройки TIMER1, действия, осуществляемые кнопкой S1 (включать светодиод), либо изменить вывод информации на ЖК индикатор. Также можно проводить различные операции с микроконтроллером.

19. На основании Примера 2 следует изучить работу ШИМ генератора микроконтроллера; изменить действия, осуществляемые кнопкой S1; изменить вывод информации на ЖК индикатор; поработать с потенциометром R1.

20. Для составления программ и подробного изучения работы микроконтроллеров AVR следует обращаться к учебной литературе, полному руководству к микроконтроллерам семейства AVR, а также к документации по среде Algorithm Builder.

21. По окончании работы следует закрыть программу и все открытые подпрограммы, закрыть виртуальную среду VirtualBox (при работе в среде Linux).



22. Выключить компьютер, нажав на кнопку, находящуюся в крайнем нижнем левом углу экрана. Из доступных действий выбрать «ВЫХОД»--> «ВЫКЛЮЧИТЬ КОМПЬЮТЕР».

23. Отключить установку от сети, поставив переключатели «СЕТЬ» на панели установки в положение «ВЫКЛ» и вынуть сетевые вилки из розеток. \_\_

### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы, написанные по заданным в задании алгоритмам

### **Контрольные вопросы**

1. Чем отличается микропроцессор (МП) от микроконтроллера (МК)?
2. Какие типы корпусов используются для микропроцессоров и микроконтроллеров?
3. Какие программные средства используются для программирования МП и МК?
4. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?
5. Какие функциональные блоки имеются в МК и отсутствуют в МП?
6. Каким образом используются выходы МК AVR семейства Mega?
7. Каким напряжением питается AVR семейства Mega?

### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
3. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.: Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опачий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь, 1990 (1982).
11. Большие интегральные схемы ЗУ./ Под ред. А.Ю. Гордонова, Ю.Н. Дьякова. Справочное пособие. – М: Радио и связь, 1990. – 286 с.
12. Федорков Б.Г., Телец В.А. Микросхемы ЦАП и АЦП: Функционирование, параметры, применение. – М.: Энергоатомиздат, 1990. – 320 с.
13. Гилмор Ч. Введение в микропроцессорную технику. Пер. с англ. – М.:

## Лабораторная работа 2

### Изучение системы команд микроконтроллера AVR

#### Цель работы

Изучение системы команд микроконтроллера AVR Atmega8535, Atmega16, Atmega32 и совместимого изучить основные приёмы программирования и отладки микроконтроллера..

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контроллеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

Система команд микроконтроллеров AVR семейства Mega весьма развита и насчитывает в различных моделях от 130 до 135 различных инструкций. Несмотря на то что микроконтроллеры AVR являются микроконтроллерами с RISC-архитектурой (процессор с сокращенным набором команд), по количеству реализованных инструкций и их разнообразию они больше похожи на микроконтроллеры с CISC-архитектурой (процессор с полным набором команд). Практически каждая из команд (за исключением команд, у которых одним из операндов является 16-битный адрес) занимает одну ячейку памяти программ. Причем это достигнуто не за счет сокращения числа команд процессора, а за счет увеличения разрядности памяти программ.

## 3.2. Операнды

Программа для любого микроконтроллера представляет собой последовательность команд, записанных в памяти программ. Большинство команд при выполнении изменяют содержимое одного или нескольких регистров общего назначения, регистров ввода/вывода или ячеек ОЗУ.

Для обращения к различным областям адресного пространства памяти данных используются различные команды, реализующие, в свою очередь, различные способы адресации. Подробно способы адресации памяти данных были рассмотрены в главе 2.

Доступ к регистрам ввода/вывода осуществляется по их адресам, являющихся операндами команды. Однако при написании программ гораздо удобнее обращаться к регистрам, используя вместо числовых значений адресов их стандартные, принятые в документации символические имена. Чтобы задать соответствие этих имен реальным адресам, необходимо подключить в начале программы (при помощи директивы ассемблера `.INCLUDE`) файл определения адресов регистров ввода/вывода. Помимо всего прочего, такое решение облегчит перенос программного обеспечения с одного типа кристалла на другой.

Эти файлы (для каждой модели микроконтроллеров семейства) свободно распространяются фирмой Atmel вместе с документацией на микроконтроллеры (в частности, включаемые файлы для всех выпускаемых микроконтроллеров AVR входят в комплект бесплатно распространяемой интегрированной среды AVRStudio). Для РОН, используемых в индексных регистрах, в этих файлах определяются также дополнительные символические имена (Табл. 3.1).

Таблица 3.1. Дополнительные символические имена индексных регистров

Регистр	Символическое имя
R26	XL
R27	XH
R28	YL
R29	YH
R30	ZL
R31	ZH

Названия этих включаемых файлов унифицированы и определяются следующим образом:

```
<номер_модели>def.inc
```

Например, программа для микроконтроллера ATmega128x должна содержать следующую директиву ассемблера:

```
.include "m128def.inc",
```

Необходимо только помнить, что если для обращения к регистру ввода/вывода используются команды обмена с ОЗУ, то к символическому имени требуется прибавить число \$20.

Как говорилось выше, в микроконтроллерах семейства память программ является 16-битной. Поэтому большинство команд описываются 16-битным словом, которое называется также кодом операции (КОП). Код операции — это число, расположенное в памяти программ и определяющее действие, которое необходимо произвести между источником и приемником. Некоторые команды, у которых один из операндов является 16-битным адресом, занимают две ячейки памяти программ. Соответственно, код операции таких команд является 4-байтным числом.

В ряде случаев значение операнда-источника может содержаться непосредственно в коде операции, а не в регистре. Это происходит в том случае, когда операндом-источником является константа.

Некоторые константы, которые могут быть полезны при написании программ, определены в упомянутых включаемых файлах:

IOEND	— значение верхнего адреса области ПВВ;
SRAM_START	— значение младшего адреса ОЗУ, доступного для хранения данных;
SRAM_SIZE	— объем ОЗУ в байтах;
RAMEND	— значение верхнего адреса внутреннего ОЗУ;
XRAMEND	— значение верхнего адреса внешнего ОЗУ (для моделей, не поддерживающих подключение внешнего ОЗУ, эта константа равна 0);
E2END	— значение верхнего адреса EEPROM;
EEPROMEND	— то же, что и E2END;
EEADRBITS	— число задействованных битов регистра адреса EEPROM;

FLASHEND	— значение верхнего адреса памяти программ (в 2-байтных словах);
PAGESIZE	— размер страницы памяти программ (в 2-байтных словах);
NRWW_START_ADDR	— младший адрес области NRWW;
NRWW_STOP_ADDR	— старший адрес области NRWW;
RWW_START_ADDR	— младший адрес области RWW;
RWW_STOP_ADDR	— старший адрес области RWW;
FIRSTBOOTSTART	— наименьший размер области загрузчика;
SMALLBOOTSTART	— то же, что и FIRSTBOOTSTART;
SECONDBOOTSTART	— вторая возможная величина области загрузчика;
THIRDBOOTSTART	— третья возможная величина области загрузчика;
FOURTHBOOTSTART	— наибольший размер области загрузчика;
LARGEBOOTSTART	— то же, что и FOURTHBOOTSTART.

### 3.3. Типы команд

Все множество команд микроконтроллеров AVR семейства Mega можно разбить на несколько групп:

- команды логических операций;
- команды арифметических операций и команды сдвига;
- команды операций с битами;
- команды пересылки данных;
- команды передачи управления;
- команды управления системой.

Далее подробно описана каждая группа команд.

#### 3.3.1. Команды логических операций

Эти команды позволяют выполнять стандартные логические операции над байтами, такие как логическое умножение (И), логическое сложение (ИЛИ), операцию «Исключающее ИЛИ», а также вычисление обратного (дополнение до единицы) и дополнительного (дополнение до двух) кодов числа. К этой группе можно отнести также команды очистки/установки регистров и команду перестановки полубайтов. Операции производятся между регистрами общего назначения, либо между регистром и константой; результат сохраняется в РОН. Все команды из этой группы выполняются за один такт.

#### 3.3.2. Команды арифметических операций и команды сдвига

К данной группе относятся команды, позволяющие выполнять такие базовые операции, как сложение, вычитание, сдвиг (вправо и влево), инкрементирование, декрементирование, а также умножение. Все операции производятся только над регистрами общего назначения. При этом мик-

роконтроллеры AVR позволяют легко оперировать как знаковыми, так и беззнаковыми числами, а также работать с числами, представленными в дополнительном коде.

Почти все команды рассматриваемой группы выполняются за один такт. Команды умножения и команды, оперирующие 2-байтными значениями, выполняются за два такта.

### 3.3.3. Команды битовых операций

К данной группе относятся команды, выполняющие установку или сброс заданного бита PОН или PВВ. Причем для изменения битов регистра состояния SREG имеются отдельные команды (точнее говоря, эквивалентные мнемонические обозначения общих команд), так как проверка состояния битов именно этого регистра производится чаще всего. Условно к этой группе можно отнести также две команды передачи управления типа «проверка/пропуск», которые пропускают следующую команду в зависимости от состояния бита PОН или PВВ.

Все задействованные биты PВВ имеют свои символические имена. Определения этих имен описаны в том же включаемом файле, что и определения символических имен адресов регистров (см. раздел 3.2). Соответственно, после включения в программу указанного файла в командах вместо числовых значений номеров битов можно будет указывать их символические имена.

Следует помнить, что в командах CBR и SBR операндом является битовая маска, а не номер бита. Для получения битовой маски из номера бита следует воспользоваться ассемблерным оператором «сдвиг влево» («<<»), как показано в следующем примере:

```
sbr    r16, (1<<SE)+(1<<SM)
out    MCUCR, r16          ; Установить флаги SE и SM регистра MCUCR
```

Всем командам данной группы требуется один такт для выполнения, за исключением случаев, когда в результате проверки происходит пропуск команды. В этом случае команда выполняется за 2 или 3 такта, в зависимости от пропускаемой команды.

### 3.3.4. Команды пересылки данных

Команды этой группы предназначены для пересылки содержимого ячеек, находящихся в адресном пространстве памяти данных. Разделение адресного пространства на три части (PОН, PВВ, ОЗУ) предопределило разнообразие команд данной группы. Пересылка данных, выполняемая командами группы, может производиться в следующих направлениях:

- PОН  $\Leftrightarrow$  PОН;
- PОН  $\Leftrightarrow$  PВВ;
- PОН  $\Leftrightarrow$  память данных.

Также к данной группе можно отнести стековые команды PUSH и POP, позволяющие сохранять в стеке и восстанавливать из стека содержимое PОН.

На выполнение команд данной группы требуется от одного до трех тактов, в зависимости от команды.

### 3.3.5. Команды передачи управления

В эту группу входят команды перехода, вызова подпрограмм и возврата из них и команды типа «проверка/пропуск», пропускающие следующую за ними команду при выполнении некоторого условия. Также к этой группе относятся команды сравнения, формирующие флаги регистра SREG и предназначенные, как правило, для работы совместно с командами условного перехода.

В системе команд микроконтроллеров семейства имеются команды как безусловного, так и условного перехода. Команды относительного (RJMP), косвенного (IJMP, EIJMP) и абсолютного (JMP) безусловного перехода являются самыми простыми в этой группе. Их функция заключается только в записи нового адреса в счетчик команд. Команды условного перехода также изменяют содержимое счетчика команд, однако это изменение происходит только при выполнении некоторого условия или, точнее, при определенном состоянии различных флагов регистра SREG.

Таблица 3.2. Сводная таблица команд условного перехода

Проверка	Логическое условие	Команда	Обратная проверка	Логическое условие	Команда	Тип данных
$Rd > Rr$	$Z \cdot (N \oplus V) = 0$	BRLT <sup>1)</sup>	$Rd \leq Rr$	$Z + (N \oplus V) = 1$	BRGE <sup>1)</sup>	Со знаком
$Rd \geq Rr$	$(N \oplus V) = 0$	BRGE	$Rd < Rr$	$(N \oplus V) = 1$	BRLT	Со знаком
$Rd = Rr$	$Z = 1$	BREQ	$Rd \neq Rr$	$Z = 0$	BRNE	Со знаком
$Rd \leq Rr$	$Z + (N \oplus V) = 1$	BRGE <sup>1)</sup>	$Rd > Rr$	$Z \cdot (N \oplus V) = 0$	BRLT <sup>1)</sup>	Со знаком
$Rd < Rr$	$(N \oplus V) = 1$	BRLT	$Rd \geq Rr$	$(N \oplus V) = 0$	BRGE	Со знаком
$Rd > Rr$	$C + Z = 0$	BRLO <sup>1)</sup>	$Rd \leq Rr$	$C + Z = 1$	BRSH <sup>1)</sup>	Без знака
$Rd \geq Rr$	$C = 0$	BRHS/BRCC	$Rd < Rr$	$C = 1$	BRLO/BRCS	Без знака
$Rd = Rr$	$Z = 1$	BREQ	$Rd \neq Rr$	$Z = 0$	BRNE	Без знака
$Rd \leq Rr$	$C = Z = 1$	BRSH <sup>1)</sup>	$Rd > Rr$	$C = Z = 0$	BRLO <sup>1)</sup>	Без знака
$Rd < Rr$	$C = 1$	BRLO/BRCS	$Rd \geq Rr$	$C = 0$	BRSH/BRCC	Без знака
«Перенос»	$C = 1$	BRCS	«Нет переноса»	$C = 0$	BRCC	—
«Меньше нуля»	$N = 1$	BRMI	«Больше нуля»	$N = 0$	BRPL	—
«Переполнение»	$V = 1$	BRVS	«Нет переполнения»	$V = 0$	BRVC	—
«Ноль»	$Z = 1$	BREQ	«Не ноль»	$Z = 0$	BRNE	—

<sup>1)</sup> Для перехода по этому условию операнды предшествующей команды сравнения должны быть записаны в обратном порядке, т. е. вместо CP Rd, Rr → CP Rr, Rd.

Вообще говоря, команды, указанные в Табл. 3.2, являются всего лишь эквивалентными мнемоническими обозначениями команд BRBS *s, k* и BRBC *s, k* с определенными значениями операнда *s*. Команда BREQ *k* имеет, например, такой же код операции, что и команда BRBS *1, k*, а команда BRGE *k* — BRBC *4, k*.

Команды вызова подпрограммы (RCALL, ICALL, EICALL и CALL) работают практически так же, как и команды безусловного перехода. Отличие заключается в том, что, перед тем как выполнить переход, значение счетчика команд сохраняется в стеке. Кроме того, подпрограмма должна заканчиваться командой возврата RET, как показано в следующем примере:

```

...
rcall sp_test      ; Вызов подпрограммы «sp_test»
...               ; Текст основной программы
...
sp_test           ; Метка подпрограммы
push  r2          ; Сохранить r2 в стеке
...              ; Выполнение подпрограммы
...
pop  r2           ; Восстановить r2 из стека
ret              ; Возврат из подпрограммы

```

В приведенном выше примере команда RET заменяет адрес, находящийся в счетчике команд, адресом команды, следующей за командой CALL.

Очевидно, что команды передачи управления нарушают нормальное (линейное) выполнение основной программы. Поэтому при каждом выполнении команды этой группы (кроме команд сравнения) нормальное функционирование конвейера нарушается. Перед загрузкой в конвейер нового адреса производится остановка и очистка выполняемой последовательности команд. В результате на выполнение команд затрачивается больше одного такта. Чтобы получить более точную информацию, обратитесь к таблицам, приведенным в разделе 3.4.

### 3.3.6. Команды управления системой

В эту группу входят всего 4 команды:

- NOP — пустая команда;
- SLEEP — перевод микроконтроллера в режим пониженного энергопотребления;
- WDR — сброс сторожевого таймера;
- BREAK — команда, используемая внутрисхемным отладчиком.

Все команды этой группы, кроме последней, выполняются за один такт.

## 3.4. Сводные таблицы команд

В Табл. 3.3...3.8 указаны все команды, которыми располагают микроконтроллеры AVR семейства Mega. В каждой таблице команды сгруппированы по функциональному признаку. В таблицах приведены основные сведения о командах, такие как мнемоническое обозначение команды, ее описание, число тактов, необходимых для ее выполнения, а также флаги регистра SREG, на которые воздействует эта команда. Информация в таблицах изложена в сжатом виде, а детальное описание всех команд приведено в разделе 3.5.



**Таблица 3.4. Группа команд арифметических операций**

Мнемоника	Описание	Операция	Число тактов	Флаги
SUB Rd, Rr	Вычитание двух РОН	$Rd = Rd - Rr$	1	Z, C, N, V, H
SUBI Rd, K	Вычитание константы из РОН	$Rd = Rd - K$	1	Z, C, N, V, H
SBC Rd, Rr	Вычитание двух РОН с заемом	$Rd = Rd - Rr - C$	1	Z, C, N, V, H
SBCI Rd, K	Вычитание константы из РОН с заемом	$Rd = Rd - K - C$	1	Z, C, N, V, H
SBIW Rdh:Rdl, K	Вычитание константы из регистровой пары	$Rdh:Rdl = Rdh:Rdl - K$	2	Z, C, N, V, S
DEC Rd	Декрементирование РОН	$Rd = Rd - 1$	1	Z, N, V
INC Rd	Инкрементирование РОН	$Rd = Rd + 1$	1	Z, N, V

**Таблица 3.3. Группа команд логических операций**

Мнемоника	Описание	Операция	Число тактов	Флаги
AND Rd, Rr	«Логическое И» двух РОН	$Rd = Rd \cdot Rr$	1	Z, N, V
ANDI Rd, K	«Логическое И» РОН и константы	$Rd = Rd \cdot K$	1	Z, N, V
EOR Rd, Rr	«Исключающее ИЛИ» двух РОН	$Rd = Rd \oplus Rr$	1	Z, N, V
OR Rd, Rr	«Логическое ИЛИ» двух РОН	$Rd = Rd \vee Rr$	1	Z, N, V
ORI Rd, K	«Логическое ИЛИ» РОН и константы	$Rd = Rd \vee K$	1	Z, N, V
COM Rd	Перевод в обратный код	$Rd = \$FF - Rd$	1	Z, C, N, V
NEG Rd	Перевод в дополнительный код	$Rd = \$00 - Rd$	1	Z, C, N, V, H
CLR Rd	Сброс всех битов РОН	$Rd = Rd \oplus Rd$	1	Z, N, V
SER Rd	Установка всех битов РОН	$Rd = \$FF$	1	—
TST Rd	Проверка РОН на отрицательное или нулевое значение	$Rd \cdot Rd$	1	Z, N, V
SWAP Rd	Обмен местами полубайтов в РОН	$Rd(3...0) = Rd(7...4),$ $Rd(7...4) = Rd(3...0)$	1	—

**Таблица 3.4. Группа команд арифметических операций**

Мнемоника	Описание	Операция	Число тактов	Флаги
ADD Rd, Rr	Сложение двух РОН	$Rd = Rd + Rr$	1	Z, C, N, V, H
ADC Rd, Rr	Сложение двух РОН с переносом	$Rd = Rd + Rr + C$	1	Z, C, N, V, H
ADIW Rdh:Rdl, K	Сложение регистровой пары с константой	$Rdh:Rdl = Rdh:Rdl + K$	2	Z, C, N, V, S

ASR Rd	Арифметический сдвиг вправо	$Rd(n) = Rd(n+1), n = 0..6$	1	Z, C, N, V
LSL Rd	Логический сдвиг влево	$Rd(n+1) = Rd(n), Rd(0) = 0$	1	Z, C, N, V
LSR Rd	Логический сдвиг вправо	$Rd(n) = Rd(n+1), Rd(7) = 0$	1	Z, C, N, V
ROL Rd	Сдвиг влево через перенос	$Rd(0) = C, Rd(n+1) = Rd(n), C = Rd(7)$	1	Z, C, N, V
ROR Rd	Сдвиг вправо через перенос	$Rd(7) = C, Rd(n) = Rd(n+1), C = Rd(0)$	1	Z, C, N, V
MUL Rd, Rr	Умножение беззнаковых чисел	$R1:R0 = Rd \times Rr$	2	Z, C
MULS Rd, Rr	Умножение чисел со знаком	$R1:R0 = Rd \times Rr$	2	Z, C
MULSU Rd, Rr	Умножение беззнакового числа на число со знаком	$R1:R0 = Rd \times Rr$	2	Z, C
FMUL Rd, Rr	Умножение дробных беззнаковых чисел	$R1:R0 = (Rd \times Rr) \ll 1$	2	Z, C
FMULS Rd, Rr	Умножение дробных чисел со знаком	$R1:R0 = (Rd \times Rr) \ll 1$	2	Z, C
FMULSU Rd, Rr	Умножение дробного беззнакового числа и дробного числа со знаком	$R1:R0 = (Rd \times Rr) \ll 1$	2	Z, C

Мнемоника	Описание	Операция	Число тактов	Флаги
CBR Rd, K	Сброс бита(ов) PОН	$Rd = Rd \bullet (\$FF - K)$	1	Z, N, V
SBR Rd, K	Установка бита(ов) PОН	$Rd = Rd \vee K$	1	Z, N, V
СBI A, b	Сброс бита PВВ	$A.b = 0$	2	—
SBI A, b	Установка бита PВВ	$A.b = 1$	2	—
BCLR s	Сброс флага	$SREG.s = 0$	1	SREG.s
BSET s	Установка флага	$SREG.s = 1$	1	SREG.s
BLD Rd, b	Загрузка бита PОН из флага T (SREG)	$Rd.b = T$	1	—
BST Rr, b	Запись бита PОН в флаг T (SREG)	$T = Rr.b$	1	T
CLC	Сброс флага переноса	$C = 0$	1	C
SEC	Установка флага переноса	$C = 1$	1	C
CLN	Сброс флага отрицательного числа	$N = 0$	1	N
SEN	Установка флага отрицательного числа	$N = 1$	1	N

CLZ	Сброс флага нуля	$Z = 0$	1	Z
SEZ	Установка флага нуля	$Z = 1$	1	Z
CLI	Общее запрещение прерываний	$I = 0$	1	I
SEI	Общее разрешение прерываний	$I = 1$	1	I
CLS	Сброс флага знака	$S = 0$	1	S
SES	Установка флага знака	$S = 1$	1	S
CLV	Сброс флага переполнения дополнительного кода	$V = 0$	1	V
SEV	Установка флага переполнения дополнительного кода	$V = 1$	1	V
CLT	Сброс флага T	$T = 0$	1	T
SET	Установка флага T	$T = 1$	1	T
CLH	Сброс флага половинного переноса	$H = 0$	1	H
SEH	Установка флага половинного переноса	$H = 1$	1	H

Мнемоника	Описание	Операция	Число тактов	Флаги
MOV Rd, Rr	Пересылка между РОН	$Rd = Rr$	1	—
MOVW Rd, Rr	Пересылка 2-байтных значений	$Rd+1\ Rd = Rr+1\ Rr$	1	—
LDI Rd, K	Загрузка константы в РОН	$Rd = K$	1	—
LD Rd, X	Косвенное чтение	$Rd = [X]$	2	—
LD Rd, X+	Косвенное чтение с постинкрементом	$Rd = [X], X = X + 1$	2	—
LD Rd, -X	Косвенное чтение с преддекрементом	$X = X - 1, Rd = [X]$	2	—
LD Rd, Y	Косвенное чтение	$Rd = [Y]$	2	—
LD Rd, Y+	Косвенное чтение с постинкрементом	$Rd = [Y], Y = Y + 1$	2	—
LD Rd, -Y	Косвенное чтение с преддекрементом	$Y = Y - 1, Rd = [Y]$	2	—
LDD Rd, Y+q	Косвенное относительное чтение	$Rd = [Y+q]$	2	—
LD Rd, Z	Косвенное чтение	$Rd = [Z]$	2	—
LD Rd, Z+	Косвенное чтение с постинкрементом	$Rd = [Z], Z = Z + 1$	2	—
LD Rd, -Z	Косвенное чтение с преддекрементом	$Z = Z - 1, Rd = [Z]$	2	—
LDD Rd, Z+q	Косвенное относительное чтение	$Rd = [Z+q]$	2	—
LDS Rd, k	Непосредственное чтение из ОЗУ	$Rd = [k]$	2	—
ST X, Rr	Косвенная запись	$[X] = Rr$	2	—
ST X+, Rr	Косвенная запись с постинкрементом	$[X] = Rr, X = X + 1$	2	—
ST -X, Rr	Косвенная запись с преддекрементом	$X = X - 1, [X] = Rr$	2	—
ST Y, Rr	Косвенная запись	$[Y] = Rr$	2	—
ST Y+, Rr	Косвенная запись с постинкрементом	$[Y] = Rr, Y = Y + 1$	2	—
ST -Y, Rr	Косвенная запись с преддекрементом	$Y = Y - 1, [Y] = Rr$	2	—
STD Y+q, Rr	Косвенная относительная запись	$[Y+q] = Rr$	2	—

ST Z, Rr	Косвенная запись	$[Z] = Rr$	2	—
ST Z+, Rr	Косвенная запись с постинкрементом	$[Z] = Rr, Z = Z + 1$	2	—
ST -Z, Rr	Косвенная запись с преддекрементом	$Z = Z - 1, [Z] = Rr$	2	—
STD Z+q, Rr	Косвенная относительная запись	$[Z+q] = Rr$	2	—
STS k, Rr	Непосредственная запись в ОЗУ	$[k] = Rr$	2	—
LPM	Загрузка данных из памяти программ	$R0 = \{Z\}$	3	—
LPM Rd, Z	Загрузка данных из памяти программ	$Rv = \{Z\}$	3	—
LPM Rd, Z+	Загрузка данных из памяти программ с постинкрементом	$Rv = \{Z\}, Z = Z + 1$	3	—
ELPM	Расширенная загрузка данных из памяти программ	$R0 = \{RAMPZ Z\}$	3	—
ELPM Rd, Z	Расширенная загрузка данных из памяти программ	$Rv = \{RAMPZ Z\}$	3	—

Мнемоника	Описание	Операция	Число тактов	Флаги
ELPM Rd, Z+	Расширенная загрузка данных из памяти программ с постинкрементом	$Rv = \{RAMPZ \cdot Z\}, RAMPZ \cdot Z = RAMPZ \cdot Z + 1$	3	—
SPM	Запись в память программ	$\{Z\} = R1:R0$	—	—
IN Rd, A	Пересылка из RBV в PОН	$Rd = A$	1	—
OUT A, Rr	Пересылка из PОН в RBV	$A = Rr$	1	—
PUSH Rr	Сохранение байта в стеке	$STACK = Rr$	2	—
POP Rd	Извлечение байта из стека	$Rd = STACK$	2	—

**Таблица 3.7. Группа команд передачи управления**

Мнемоника	Описание	Операция	Число тактов	Флаги
RJMP k	Относительный безусловный переход	$PC = PC + k + 1$	2	—
IJMP	Косвенный безусловный переход	$PC = Z$	2	—
EIJMP	Расширенный косвенный безусловный переход	$PC = EIND:Z$	2	—
JMP k	Абсолютный переход	$PC = k$	3	—
RCALL k	Относительный вызов подпрограммы	$PC = PC + k + 1$	3 (4)	—
ICALL	Косвенный вызов подпрограммы	$PC = Z$	3 (4)	—
EICALL	Расширенный косвенный вызов подпрограммы	$PC = EIND.Z$	4	—
CALL k	Абсолютный вызов подпрограммы	$PC = k$	4 (5)	—
RET	Возврат из подпрограммы	$PC = STACK$	4 (5)	—
RETI	Возврат из подпрограммы обработки прерывания	$PC = STACK$	4 (5)	I
CP Rd, Rr	Сравнение PОН	$Rd - Rr$	1	Z, N, V, C, H
CPC Rd, Rr	Сравнение PОН с учетом переноса	$Rd - Rr - C$	1	Z, N, V, C, H
CPI Rd, K	Сравнение PОН с константой	$Rd - K$	1	Z, N, V, C, H

CPSE Rd, Rr	Сравнение и пропуск следующей команды при равенстве	Если Rd = Rr, то PC = PC + 2 (3)	1/2/3	—
SBRC Rr, b	Пропуск следующей команды, если бит РОН сброшен	Если Rr.b = 0, то PC = PC + 2 (3)	1/2/3	—
SBRS Rr, b	Пропуск следующей команды, если бит РОН установлен	Если Rr.b = 1, то PC = PC + 2 (3)	1/2/3	—
SBIC A, b	Пропуск следующей команды, если бит РВВ сброшен	Если A.b = 0, то PC = PC + 2 (3)	1/2/3	—
Мнемоника	Описание	Операция	Число тактов	Флаги
SBIS A, b	Пропуск следующей команды, если бит РВВ установлен	Если A.b = 1, то PC = PC + 2 (3)	1/2/3	—
BRBC s, k	Переход, если флаг s регистра SREG сброшен	Если SREG.s = 0, то PC = PC + k + 1	1/2	—
BRBS s, k	Переход, если флаг s регистра SREG установлен	Если SREG.s = 1, то PC = PC + k + 1	1/2	—
BRCS k	Переход по переносу	Если C = 1, то PC = PC + k + 1	1/2	—
BRCC k	Переход, если нет переноса	Если C = 0, то PC = PC + k + 1	1/2	—
BREQ k	Переход по «равно»	Если Z = 1, то PC = PC + k + 1	1/2	—
BRNE k	Переход по «не равно»	Если Z = 0, то PC = PC + k + 1	1/2	—
BRSR k	Переход по «больше или равно»	Если C = 0, то PC = PC + k + 1	1/2	—
BRLO k	Переход по «меньше»	Если C = 1, то PC = PC + k + 1	1/2	—
BRMI	Переход по «отрицательное значение»	Если N = 1, то PC = PC + k + 1	1/2	—
BRPL	Переход по «положительное значение»	Если N = 0, то PC = PC + k + 1	1/2	—
BRGE	Переход по «больше или равно» (числа со знаком)	Если (N ⊕ V) = 0, то PC = PC + k + 1	1/2	—
BRLT	Переход по «меньше нуля» (числа со знаком)	Если (N ⊕ V) = 1, то PC = PC + k + 1	1/2	—
BRHS	Переход по половинному переносу	Если H = 1, то PC = PC + k + 1	1/2	—
BRHC	Переход, если нет половинного переноса	Если H = 0, то PC = PC + k + 1	1/2	—
BRTS	Переход, если флаг T установлен	Если T = 1, то PC = PC + k + 1	1/2	—
BRVS	Переход по переполнению дополнительного кода	Если V = 1, то PC = PC + k + 1	1/2	—
BRVC	Переход, если нет переполнения дополнительного кода	Если V = 0, то PC = PC + k + 1	1/2	—

**Таблица 3.8. Группа команд управления системой**

Мнемоника	Описание	Операция	Число тактов	Флаги
NOP	Нет операции		1	—
SLEEP	Переход в «спящий» режим	См. соответствующие разделы	3	—
WDR	Сброс сторожевого таймера	См. соответствующие разделы	1	—
BREAK	Останов	Используется только внутри-схемным отладчиком	—	—

### 3.5. Описание команд

В этом разделе в алфавитном порядке перечислены все команды, поддерживаемые микроконтроллерами семейства Mega. Для каждой команды приводится ее детальное описание. При описании команд используются обозначения, приведенные в Табл. 3.9.

**Таблица 3.9. Обозначения, используемые при описании команд**

Обозначение, символ	Описание
<b>Регистр состояния</b>	
SREG	Регистр состояния микроконтроллера
C	Флаг переноса (0-й бит регистра SREG)
Z	Флаг нуля (1-й бит регистра SREG)
N	Флаг отрицательного значения (2-й бит регистра SREG)
V	Флаг переполнения дополнительного кода (3-й бит регистра SREG)
S	Флаг знака (4-й бит регистра SREG); $S = N \oplus V$
H	Флаг половинного переноса (5-й бит регистра SREG)
T	Пользовательский флаг (6-й бит регистра SREG)
I	Флаг общего разрешения прерываний (7-й бит регистра SREG)
<b>Регистры и операнды</b>	
Rd	Регистр-приемник (иногда также регистр-источник) в регистровом файле
Rr	Регистр-источник в регистровом файле
K	Константа (данные)
k	Адрес — константа
b	Номер бита POH или PBB (0...7)
s	Номер бита регистра состояния SREG (0...7)
X, Y, Z	Индексные регистры ( $X = R27:R26$ , $Y = R29:R28$ , $Z = R31:R30$ )

Обозначение, символ	Описание
I/O	Регистр ввода/вывода
A	Адрес в пространстве ввода/вывода
q	Смещение при относительной косвенной адресации (6-битное значение)
.	Разделитель между названием (адресом) регистра и номером бита
[XX]	Содержимое ячейки памяти данных по адресу XX
{XX}	Содержимое ячейки памяти программ по адресу XX

Операции	
–	Инверсия
•	Логическое И
v	Логическое ИЛИ
⊕	Исключающее ИЛИ
Система	
PC	Счетчик команд
STACK	Текущий уровень стека
SP	Указатель стека
Флаги	
↔	Команда воздействует на флаг
0	Флаг сбрасывается командой в 0
1	Флаг устанавливается командой в 1
–	Команда не влияет на состояние флага

### Оборудование и материалы.

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже

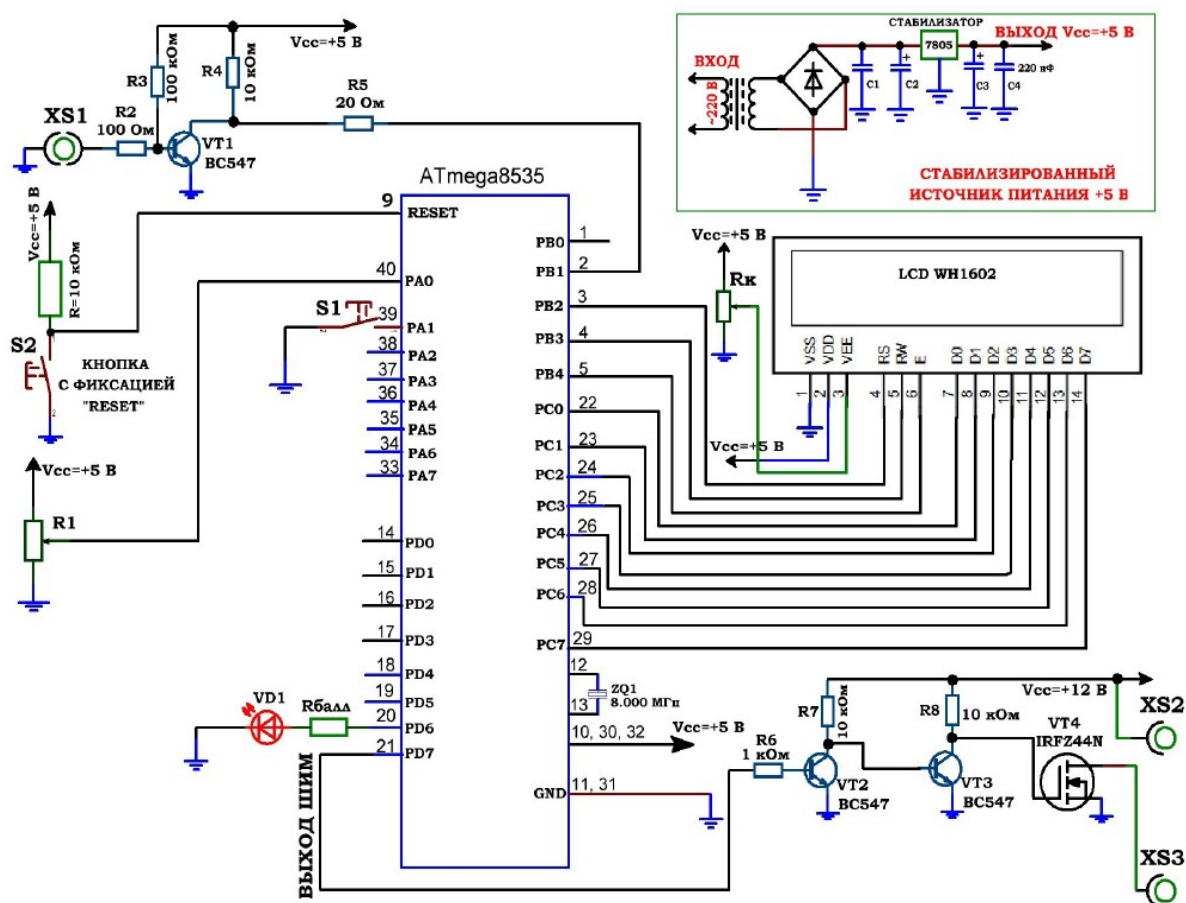


Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.

### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### **Ход работы**

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.

2. Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR, изучить все доступные паспорта на исследуемые микроконтроллеры, а также на подключенный LCD индикатор.

3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.

Задание

4. Загрузить операционную систему согласно стандартным процедурам загрузки.

5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder. папке Examples\_RTMTL-2.

6. Пример 1. По нажатию кнопки S1, подключенной к 22 выводу МК начинает светиться светодиод VD1, подключенный к выводу 1 микроконтроллера. По отжатию кнопки, светодиод перестает светиться (индикация работы кнопки);

7. Пример 2. При нажатии кнопки S1, подключенной к 22 выводу МК начинает мигать светодиод VD1, подключенный к выводу 1 микроконтроллера;

8. Пример 3. Мигание светодиода VD1, подключенного к выводу 1. 9. Пример 4. Светодиоды VD1 – VD8 мигают по очереди (по порядку подключения к выводам 1, 2, 3, 4, 18, 19, 20, 21 МК);

5) Пример 5. Светодиоды VD1 – VD8 мигают по очереди в той же последовательности с одновременной передачей по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на COM порт) номера включенного светодиода в ASCII коде (натуральное число);

### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

### **Контрольные вопросы**

1. Какие типы команд имеет микроконтроллер AVR семейства Mega?
2. Какие типы команд изменяют флаги микроконтроллера?
3. Какие виды адресации использует микроконтроллер AVR?
4. Зачем нужны команды передачи управления?
5. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?

### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988.

- 575с.

2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.

3. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.



4. 4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа №3

### Изучение работы и программирования портов микроконтроллера AVR

#### Цель работы

Изучение системы команд микроконтроллера AVR Atmega8535, Atmega16, Atmega32 и совместимого изучить основные приёмы программирования и отладки микроконтроллера..

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контролеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

## 6.2. Регистры портов ввода/вывода

Обращение к портам производится через регистры ввода/вывода. Под каждый порт в адресном пространстве ввода/вывода зарезервировано по 3 адреса, по которым размещены следующие регистры: регистр данных порта PORTx, регистр направления данных DDRx и регистр выводов порта PINx. Действительные названия регистров получаются подстановкой названия порта вместо символа x. Соответственно, регистры порта A называются PORTA, DDRA, PINA, порта B — PORTB, DDRB, PINB и т. д. Поскольку с помощью регистров PINx осуществляется доступ к физическим значениям сигналов на выводах порта, они доступны только для чтения,

тогда как остальные два регистра доступны и для чтения, и для записи. Тем не менее, в новых моделях микроконтроллеров (все модели, кроме ATmega8515x/8535x, ATmega8x/16x/32x/64x/128x и ATmega162x) запись 1 в бит регистра PINx приводит к переключению состояния соответствующего бита регистра данных PORTx.

Адреса регистров всех портов ввода/вывода приведены в Табл. 6.2.

Таблица 6.2. Регистры портов ввода/вывода

Порт	Регистр	ATmega8515x		ATmega8535x		ATmega8x		ATmega16x		ATmega162x		ATmega64x, ATmega128x		ATmega48x/88x/168x		ATmega164x/324x/644x		ATmega165x, ATmega325x/645x		ATmega3250x/6450x		ATmega1281x/2561x		ATmega640x, ATmega1280x/2560x	
A	PORTA	\$1B (\$3B)	—	\$1B (\$3B)	—	\$02 (\$22)																			
	DDRA	\$1A (\$3A)	—	\$1A (\$3A)	—	\$01 (\$21)																			
	PINA	\$19 (\$39)	—	\$19 (\$39)	—	\$00 (\$20)																			
B	PORTB	\$18 (\$38)				\$05 (\$25)																			
	DDRБ	\$17 (\$37)				\$04 (\$24)																			
	PINB	\$16 (\$36)				\$03 (\$23)																			
C	PORTC	\$15 (\$35)				\$08 (\$28)																			
	DDRC	\$14 (\$34)				\$07 (\$27)																			
	PINC	\$13 (\$33)				\$06 (\$26)																			

D	PORTD	\$12 (\$32)						\$0B (\$2B)				
	DDRD	\$11 (\$31)						\$0A (\$2A)				
	PIND	\$10 (\$30)						\$09 (\$29)				
E	PORTE	\$07 (\$27)	—	—	\$07 (\$27)	\$03 (\$23)	—	—	\$0E (\$2E)			
	DDRE	\$06 (\$26)	—	—	\$06 (\$26)	\$02 (\$22)	—	—	\$0D (\$2D)			
	PINE	\$05 (\$25)	—	—	\$05 (\$25)	\$01 (\$21)	—	—	\$0C (\$2C)			

F	PORTF	\$03 (\$23)	—	—	—	—	(\$62)	—	—	\$11 (\$31)					
	DDRF	\$02 (\$22)	—	—	—	—	(\$61)	—	—	\$10 (\$30)					
	PINF	\$01 (\$21)	—	—	—	—	\$00 (\$20)	—	—	\$0F (\$2F)					
G	PORTG	—	—	—	—	—	(\$65)	—	—	\$14 (\$34)					
	DDRG	—	—	—	—	—	(\$64)	—	—	\$13 (\$33)					
	PING	—	—	—	—	—	(\$63)	—	—	\$12 (\$32)					
H	PORTH	—	—	—	—	—	—	—	—	—	—	—	(\$DA)	—	(\$102)
	DDRH	—	—	—	—	—	—	—	—	—	—	—	(\$D9)	—	(\$101)
	PINH	—	—	—	—	—	—	—	—	—	—	—	(\$D8)	—	(\$100)
J	PORTJ	—	—	—	—	—	—	—	—	—	—	—	(\$DD)	—	(\$105)
	DDRJ	—	—	—	—	—	—	—	—	—	—	—	(\$DC)	—	(\$104)
	PINJ	—	—	—	—	—	—	—	—	—	—	—	(\$DB)	—	(\$103)

K	PORTK	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$108)
	DDRK	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$107)
	PINK	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$106)
L	PORTL	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$10B)
	DDRL	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$10A)
	PINL	—	—	—	—	—	—	—	—	—	—	—	—	—	(\$109)

### 6.3. Конфигурирование портов ввода/вывода

Упрощенная структурная схема одного из каналов порта ввода/вывода Pxi при работе его в качестве цифрового входа/выхода общего назначения приведена на Рис. 6.1.

Если же вывод функционирует как вход ( $DDx_i = 0$ ), то бит  $PORTx_i$  определяет состояние внутреннего подтягивающего резистора для данного вывода. При установке бита  $PORTx_i$  в 1 подтягивающий резистор подключается между выводом микроконтроллера и линией питания.

Вообще говоря, управление подтягивающими резисторами во всех микроконтроллерах семейства осуществляется на двух уровнях. Общее управление (для всех выводов портов) осуществляется битом PUD регистра специальных функций SFIOR или регистра управления микроконтроллера MCUCR (в зависимости от модели). В моделях ATmega64x и ATmega128x регистр SFIOR располагается по адресу \$20 (\$40), а в остальных моделях — по адресу \$30 (\$50). Регистр MCUCR располагается по адресу \$35 (\$55). Форматы этих регистров приведены на Рис. 6.2.

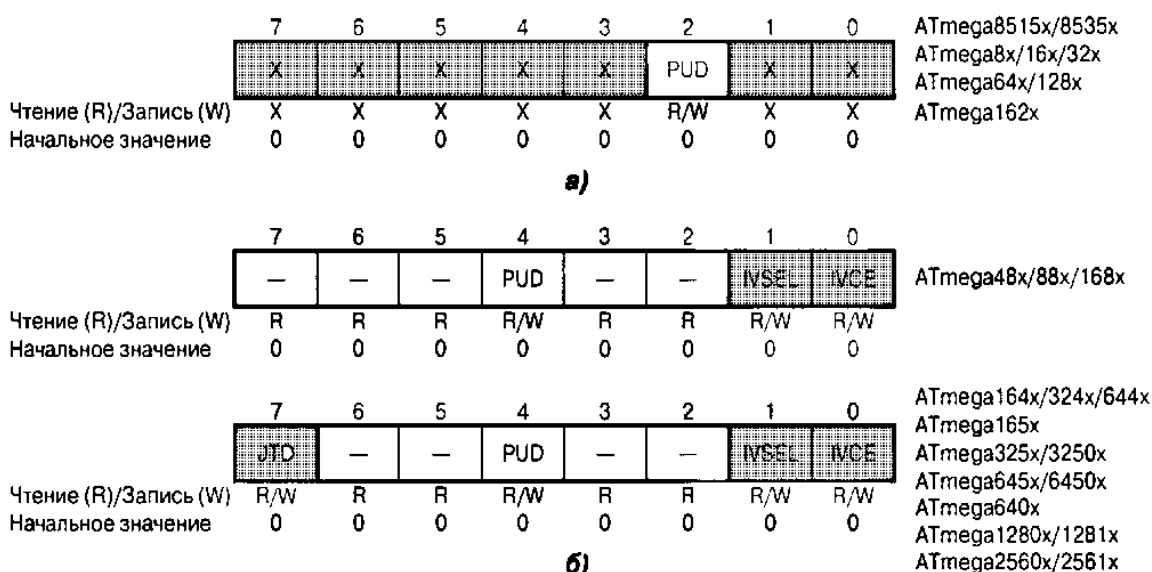
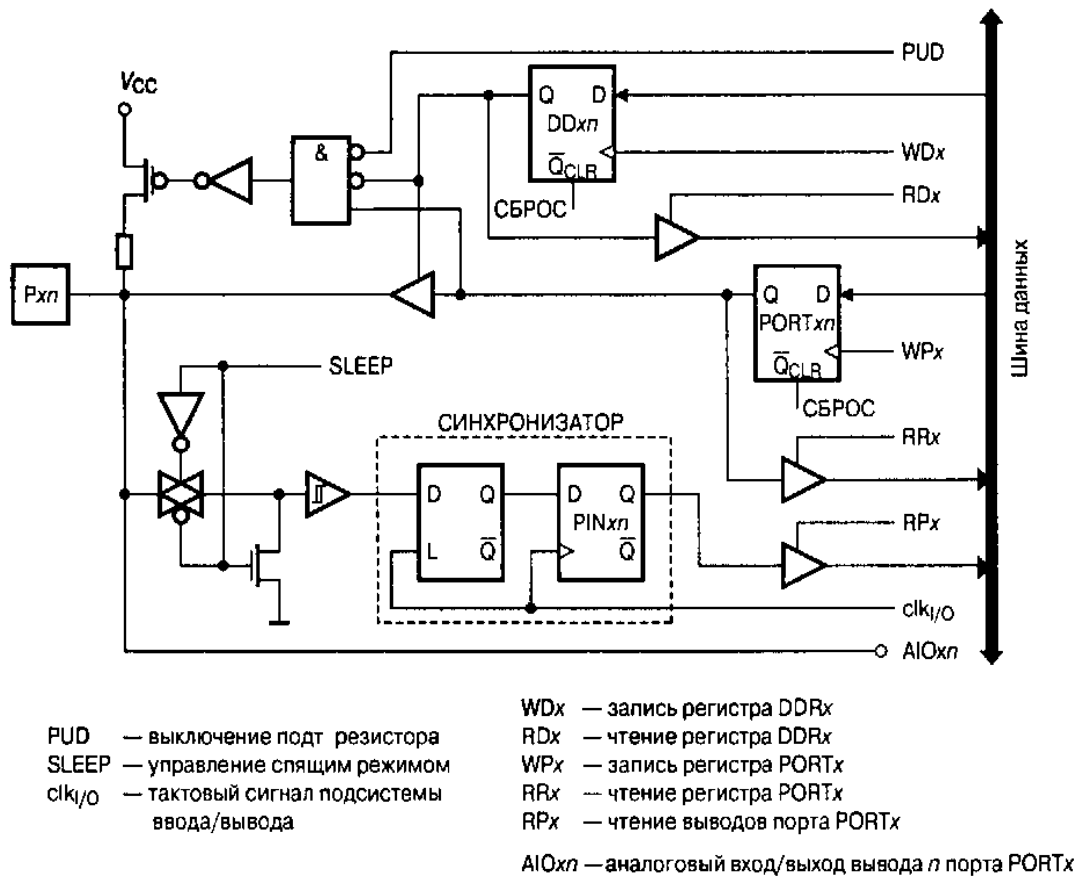


Рис 6.2 Регистры управления подтяжкой SFIOR (а) и MCUCR (б)

Если бит PUD сброшен в 0 (начальное состояние), то состояние подтягивающих резисторов будет определяться состоянием битов  $PORTx_i$  для каждого входа порта. Если же бит PUD установлен в 1, подтягивающие резисторы отключаются от всех выводов микроконтроллера.

Обратите внимание, что при переключении вывода между третьим состоянием ( $DDx_i = 0$ ,  $PORTx_i = 0$ ) и состоянием ВЫСОКОГО уровня ( $DDx_i = 1$ ,  $PORTx_i = 1$ ) происходит переход через одно из промежуточных состояний: либо включается подтягивающий резистор ( $DDx_i = 0$ ,  $PORTx_i = 1$ ), либо выход переключается в состояние НИЗКОГО уровня ( $DDx_i = 1$ ,  $PORTx_i = 0$ ). Наиболее применимым является, как правило, первый вариант, поскольку для высокоимпедансных систем безразлично, каким образом формируется ВЫСОКИЙ уровень. Если в каком-либо слу-



**Примечание.** Сигналы WP<sub>x</sub> WD<sub>x</sub> RP<sub>x</sub> RD<sub>x</sub> являются общими для всех выводов одного порта  
 сигналы clk<sub>I/O</sub> SLEEP и PUD являются общими для всех портов микроконтроллера

**Рис. 6.1.** Структурная схема канала ввода/вывода

Каждому выводу порта соответствуют три бита регистров ввода/вывода: PORT<sub>xn</sub> (регистр PORT<sub>x</sub>), DD<sub>xn</sub> (регистр DDR<sub>x</sub>) и PIN<sub>xn</sub> (регистр PIN<sub>x</sub>). Действительные названия битов регистров получаются подстановкой названия порта вместо символа *x* и номера бита вместо символа *n*. Порядковый номер вывода порта соответствует порядковому номеру бита регистров этого порта. Поэтому, если разрядность порта меньше восьми, регистрах порта используется соответствующее число младших битов. Недействующие старшие биты регистров доступны только для чтения всегда содержат 0.

Бит DD<sub>xn</sub> регистра DD<sub>x</sub> определяет направление передачи данных через контакт ввода/вывода. Если этот бит установлен в 1, то *n*-й вывод порта является выходом, если же сброшен в 0 — входом.

Бит PORT<sub>xn</sub> регистра PORT<sub>x</sub> выполняет двойную функцию. Если вывод функционирует как выход (DD<sub>xn</sub> = 1), то этот бит определяет состояние вывода порта. Если бит установлен в 1, на выводе устанавливается напряжение ВЫСОКОГО уровня. Если бит сброшен в 0, на выводе устанавливается напряжение НИЗКОГО уровня.

Если же вывод функционирует как вход ( $DDxn = 0$ ), то бит  $PORTxn$  определяет состояние внутреннего подтягивающего резистора для данного вывода. При установке бита  $PORTxn$  в 1 подтягивающий резистор подключается между выводом микроконтроллера и линией питания.

Вообще говоря, управление подтягивающими резисторами во всех микроконтроллерах семейства осуществляется на двух уровнях. Общее управление (для всех выводов портов) осуществляется битом PUD регистра специальных функций SFIOR или регистра управления микроконтроллера MCUCR (в зависимости от модели). В моделях ATmega64x и ATmega128x регистр SFIOR располагается по адресу \$20 (\$40), а в остальных моделях — по адресу \$30 (\$50). Регистр MCUCR располагается по адресу \$35 (\$55). Форматы этих регистров приведены на Рис. 6.2.

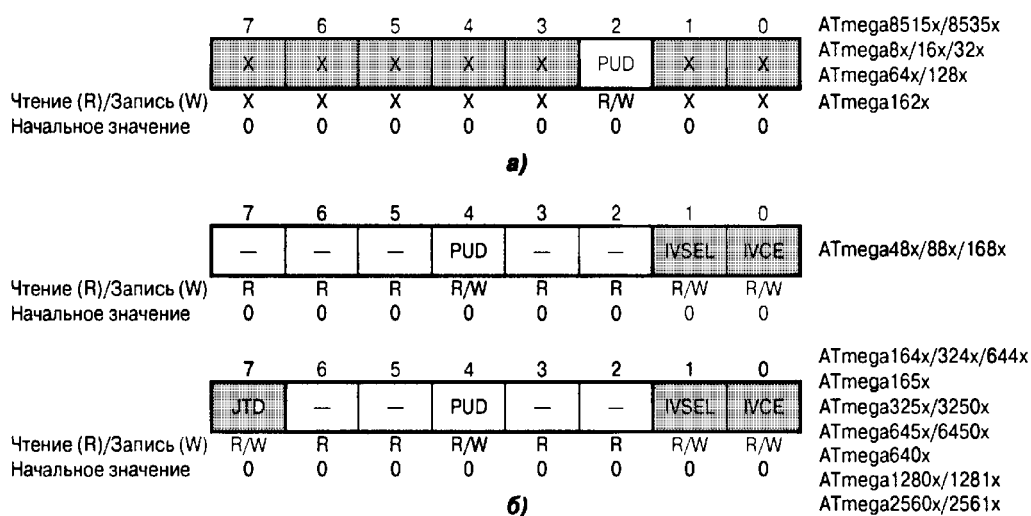


Рис. 6.2 Регистры управления подтяжкой SFIOR (а) и MCUCR (б)

Если бит PUD сброшен в 0 (начальное состояние), то состояние подтягивающих резисторов будет определяться состоянием битов  $PORTxn$  для каждого входа порта. Если же бит PUD установлен в 1, подтягивающие резисторы отключаются от всех выводов микроконтроллера.

Обратите внимание, что при переключении вывода между третьим состоянием ( $DDxn = 0$ ,  $PORTxn = 0$ ) и состоянием ВЫСОКОГО уровня ( $DDxn = 1$ ,  $PORTxn = 1$ ) происходит переход через одно из промежуточных состояний: либо включается подтягивающий резистор ( $DDxn = 0$ ,  $PORTxn = 1$ ), либо выход переключается в состояние НИЗКОГО уровня ( $DDxn = 1$ ,  $PORTxn = 0$ ). Наиболее применимым является, как правило, первый вариант, поскольку для высокоимпедансных систем безразлично, каким образом формируется ВЫСОКИЙ уровень. Если в каком-либо случае это не подходит, пользователь может отключить подтягивающие резисторы от всех портов установкой бита PUD в 1.

Аналогичная ситуация возникает и при переключении между состоянием с включенным подтягивающим резистором ( $DDx_n = 0, PORTx_n = 1$ ) и состоянием НИЗКОГО уровня ( $DDx_n = 1, PORTx_n = 0$ ). В этом случае промежуточным состоянием является либо высокоимпедансное состояние ( $DDx_n = 0, PORTx_n = 0$ ), либо состояние ВЫСОКОГО уровня ( $DDx_n = 1, PORTx_n = 1$ ).

Все возможные сочетания состояний управляющих битов и соответственно конфигурации выводов портов приведены в Табл. 6.3.

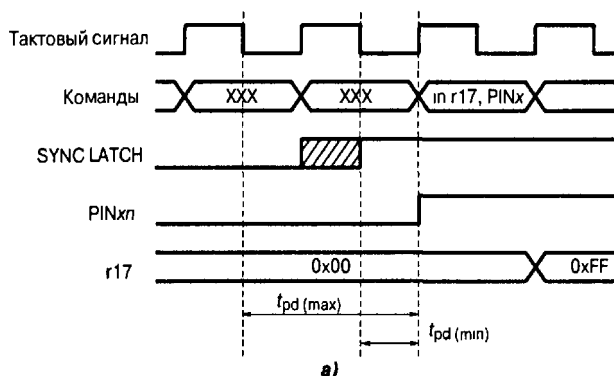
Таблица 6.3. Конфигурации выводов портов

DDx <sub>n</sub>	PORTx <sub>n</sub>	PUD	Функция вывода	Резистор	Примечание
0	0	X	Вход	Отключен	Третье состояние (Hi-Z) <sup>1)</sup>
0	1	0	Вход	Подключен	При подключении нагрузки между выводом и общим проводом вывод является источником тока
0	1	1	Вход	Отключен	Третье состояние (Hi-Z)
1	0	X	Выход	Отключен	Выход установлен в 0
1	1	X	Выход	Отключен	Выход установлен в 1

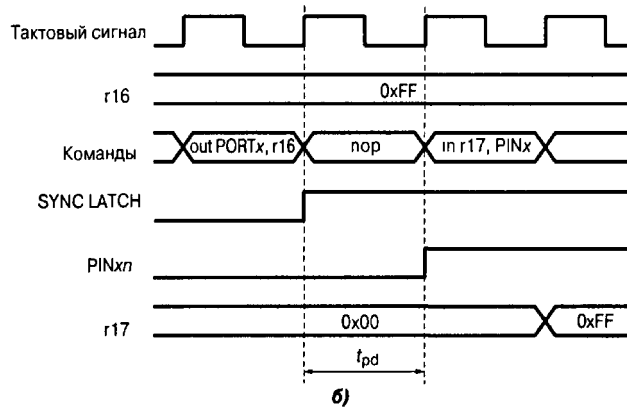
<sup>1)</sup> Состояние выводов портов при сбросе.

Состояние вывода микроконтроллера (независимо от установок бита DDx<sub>n</sub>) может быть получено путем чтения бита PINx<sub>n</sub> регистра PINx. При этом следует помнить, что между действительным изменением сигнала на выводе и изменением бита PINx<sub>n</sub> существует задержка. Эта задержка вносится узлом синхронизации, состоящим, как показано на Рис. 6.1, из бита PINx<sub>n</sub> и дополнительного триггера-защелки. Значение сигнала на выводе микроконтроллера фиксируется триггером-защелкой при НИЗКОМ уровне тактового сигнала и переписывается затем в бит PINx<sub>n</sub> по нарастающему фронту тактового сигнала. Соответственно, величина задержки может составлять от 0.5 до 1.5 периодов системного тактового сигнала, как показано на Рис. 6.3, а.

По этой же причине между операциями изменения и повторного считывания состояний вывода необходимо вставлять команду NOP. Поскольку команда OUT устанавливает сигнал «SYNC LATCH» в 1 по положительному фронту тактового сигнала, задержка в этом случае равна одному периоду тактового сигнала (Рис. 6.3, б).







**Рис. 6.3.** Синхронизация при чтении состояния вывода:  
*a* — при считывании бита  $PINxI$ ; *б* — при считывании состояния вывода, заданного программно

Далее приведен пример конфигурирования одного из портов микроконтроллера. В примере выходы 0 и 1 порта В устанавливаются в 1, выходы 2 и 3 — в 0. Выводы 4...7 порта конфигурируются как входы, при этом к выводам 6 и 7 подключаются подтягивающие резисторы.

#### Пример на ассемблере

```

...
ldi r16, (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
ldi r17, (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out  PORTB,r16                ; Задать состояние выходов и подтягивающих
                               ; резисторов
out  DDRB,r17                 ; Задать режимы работы выводов
nop                            ; для синхронизации
in   r16,PINB                 ; Считать состояние выводов порта
...

```

#### Пример на Си

```

unsigned char i;
...
/* Задать состояние выходов и подтягивающих резисторов */
/* Задать режимы работы выводов */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
_NOP();
i = PINB;
...

```

В заключение отметим, что подавляющее большинство контактов ввода/вывода всех микроконтроллеров семейства имеют дополнительные функции и могут использоваться различными периферийными устройствами микроконтроллеров. При этом возможны две ситуации. В одних случаях пользователь должен самостоятельно задавать конфигурацию вывода, а в других вывод конфигурируется автоматически при включении соответствующего периферийного устройства. Об этом будет сказано при рассмотрении соответствующих периферийных устройств.

#### Оборудование и материалы.

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже

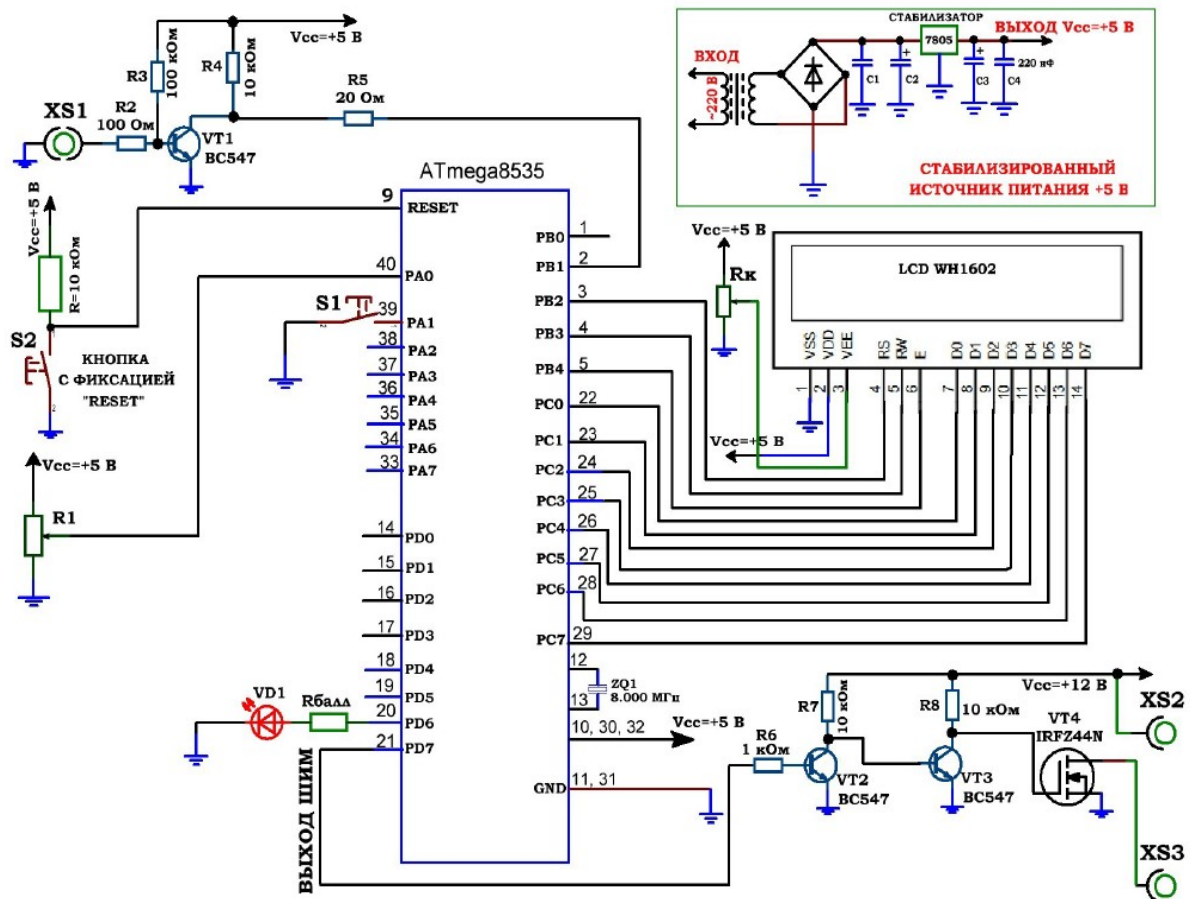


Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.

### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

Задание 1. Изучение работы АЦП микроконтроллера. Напряжение считывается с переменного резистора R1, подключённого средней точкой к 40 выводу МК (вход АЦП) в код АЦП от 0 до 63 единиц и с интервалом ~ 1 сек. передаётся по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на COM порт);

Задание 2. Приёмник RS232 с ПК. Передавая ACSII код 1 ... 8 (натуральные числа) с ПК посредством программы «ТЕРМИНАЛ» можно управлять включением светодиодов;

Задание 3. Работа с семисегментными индикаторами, RS232 и реле. Реализация таймера обратного отсчета. Текущие показания таймера выводятся на семисегментные индикаторы с общими катодами (a, b, c, d, e, f, g) и анодами A1 и A2, и одновременно передаётся по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на COM порт). При достижении конца отсчета Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535. срабатывает реле K1, подключённое через транзисторный усилитель на KT3102 к выводу 5 микроконтроллера и вторичная обмотка трансформатора TV1 (~12 В) соединяется с клеммами XS1 – XS2 (нагрузка). К выходу нагрузки можно подключать лампу накаливания 12 В, 21 Вт либо меньшей мощности. Запрещается замыкать выходы контрольных точек XS1 – XS2 (нагрузка)!;

Задание 4. Реализация вольтметра. Работа с семисегментными индикаторами и реле. Простейшая реализация вольтметра с выводом показаний на семисегментный индикатор. Напряжение считывается с переменного резистора R1, подключённого средней точкой к 40 выводу МК (вход АЦП) и выводится на семисегментный индикатор в показаниях 0-50 (0-5 Вольт, т. к. вывод h(точка dp) не подключен). При определенном пороге (определенном значении напряжения) срабатывает реле K1 и вторичная обмотка трансформатора TV1 (~12 В) соединяется с клеммами XS1 – XS2 (нагрузка). К выходу нагрузки можно подключать лампу накаливания 12 В, 21 Вт либо меньшей мощности. Запрещается замыкать выходы контрольных точек XS1 – \_\_

#### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

#### **Контрольные вопросы**

1. Какой уровень сигнала у цифровых выходов МК?
2. Какие типы команд имеет микроконтроллер AVR семейства Mega?
3. Какие типы команд изменяют флаги микроконтроллера?
4. Какие виды адресации использует микроконтроллер AVR?
5. Зачем нужны команды передачи управления?
6. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?

#### **Список литературы, рекомендуемый к использованию по данной теме:**

11. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
12. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
13. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
14. 4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
15. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
16. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
17. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
18. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
19. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
20. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа 4

### Изучение работы и программирования АЦП и ЦАП блоков ШИМ микроконтроллера AVR

#### Цель работы

Изучить работу АЦП и ЦАП блоков ШИМ микроконтроллера AVR Atmega8535, Atmega16, изучить основные приёмы программирования и отладки микроконтроллера..

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контролеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

## Общие сведения

Модуль 10-битного АЦП последовательного приближения входит в состав практически всех моделей семейства, за исключением ATmega8515x и ATmega162x. Основные параметры этого АЦП следующие:

- абсолютная погрешность:  $\pm 2 \text{ LSB}^1$ ;
- интегральная нелинейность:  $\pm 0.5 \text{ LSB}$ ;
- быстродействие: до 15 тыс. выборок/с.

На входе модуля АЦП имеется 8-канальный (в моделях ATmega640x/1280x/2560x — 16-канальный) аналоговый мультиплексор, предоставляющий в распоряжение пользователя 8 (16) каналов с несимметричными входами. Кроме того, в моделях ATmega8x, выпускаемых в корпусе DIP, доступно только 6 каналов из восьми.

В большинстве моделей входы АЦП могут объединяться попарно для формирования различного числа каналов с дифференциальным входом. При этом в некоторых каналах имеется возможность 10- и 200-кратного предварительного усиления входного сигнала. При коэффициентах усиления 1x и 10x действительная разрешающая способность АЦП по этим каналам составляет 8 бит, а при коэффициенте усиления 200x — 7 бит.

В качестве источника опорного напряжения для АЦП может использоваться как напряжение питания микроконтроллера, так и внутренний либо внешний источник опорного напряжения.

Модуль АЦП может работать в двух режимах:

- режим одиночного преобразования, когда запуск каждого преобразования инициируется пользователем;
- режим непрерывного преобразования, когда запуск преобразований выполняется непрерывно через определенные интервалы времени.

## Функционирование модуля АЦП

Обобщенная структурная схема модуля АЦП приведена на Рис. 9.1. В моделях ATmega8x и ATmega48x/88x/168x элементы, выделенные на рисунке серым цветом, и связанные с ними сигналы отсутствуют, а неинвертирующий вход компаратора выборки-хранения подключен непосредственно к выходу мультиплексора (показано пунктиром).

Регистры, используемые для управления модулем АЦП в различных моделях, приведены в Табл. 9.1.

Формат регистров ADCSRA (ADCSR) и ADMUX приведен на Рис. 9.2 и Рис. 9.3, а краткое описание функций их битов приведено в Табл. 9.2 и Табл. 9.3 соответственно.

Таблица 9.1. Регистры управления модулем АЦП

Регистр	Адрес	АТmega8535х	АТmega8х	АТmega16х/32х	АТmega64х	АТmega128х	АТmega48х/88х/168х	АТmega164х/324х/644х	АТmega165х	АТmega325х/3250х, АТmega645х/6450х	АТmega640х, АТmega1280х/1281х, АТmega2560х/2561х	Описание
ADCSR	\$06 (\$26)		•									Регистр управления и состояния
ADCSRA	\$06 (\$26)	•		•	•	•						Регистр А управления и состояния
	(\$7A)						•	•	•	•	•	
ADCSRБ	(\$8E)				•							Регистр В управления и состояния
	(\$7B)						•	•	•	•	•	
ADMUX	\$07 (\$27)	•	•	•	•	•						Регистр управления мультиплексором
	(\$7C)						•	•	•	•	•	
SFIOR	\$30 (\$50)	•	•	•								Регистр специальных функций
	\$20 (\$40)					•						

	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	АТmega8х АТmega128х
Чтение(R)/Запись(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	
	7	6	5	4	3	2	1	0	Остальные модели
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	
Чтение(R)/Запись(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	

АТmega8х — ADCSR  
Остальные модели — ADCSRA

Рис. 9.2. Формат регистра ADCSRA (ADCSR)

Таблица 9.2. Биты регистра ADCSRA (ADCSR<sup>1)</sup>)

Бит	Название	Описание
7	ADEN	Разрешение АЦП (1 — включено, 0 — выключено)
6	ADSC	Запуск преобразования (1 — начать преобразование)
5	ADATE (ADFR <sup>2)</sup> )	Выбор режима работы АЦП
4	ADIF	Флаг прерывания от компаратора
3	ADIE	Разрешение прерывания от компаратора
2...0	ADPS2:ADPS0	Выбор частоты преобразования

<sup>1)</sup> В модели АТmega8х.  
<sup>2)</sup> В моделях АТmega8х и АТmega128х.

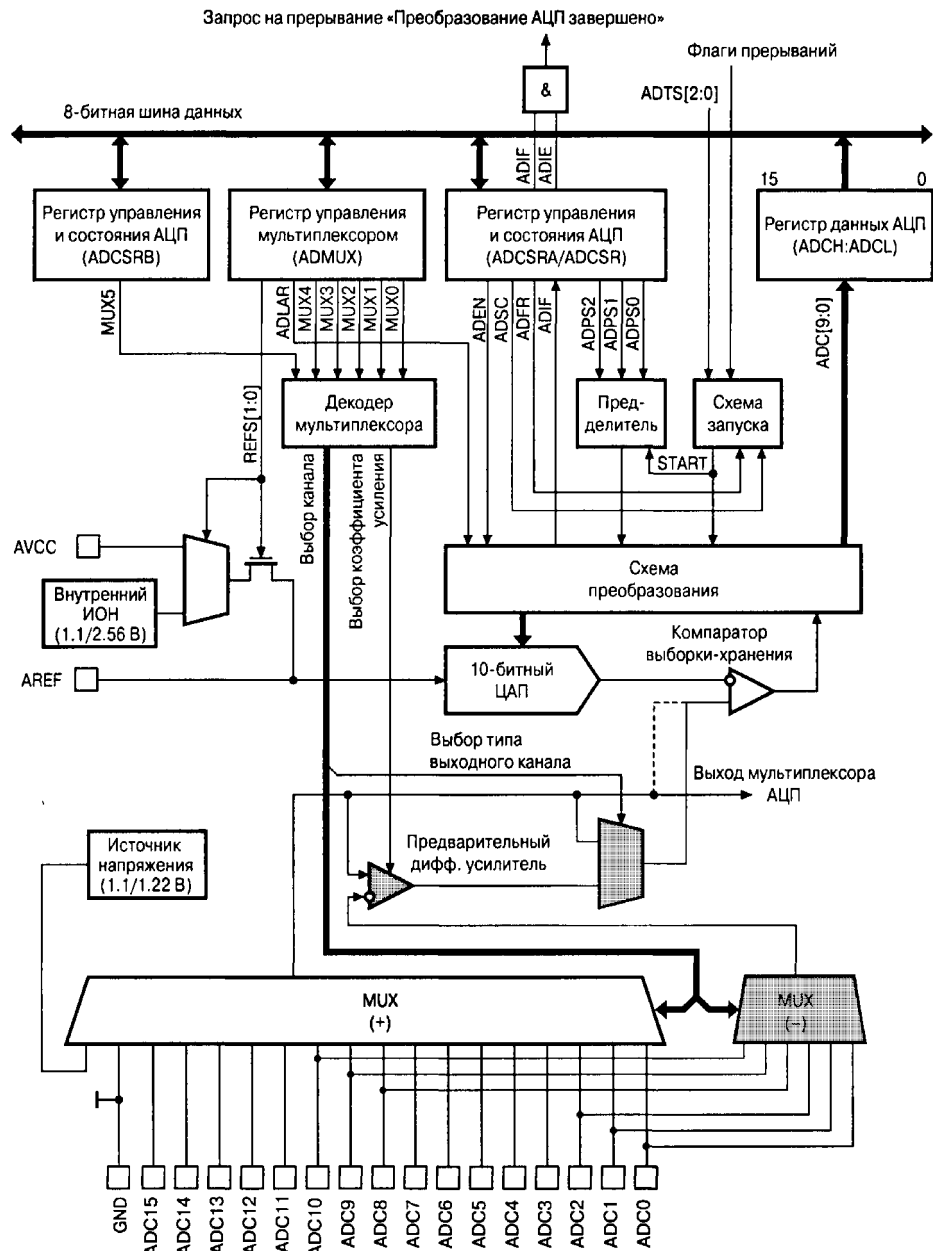


Рис. 9.1. Структурная схема модуля АЦП

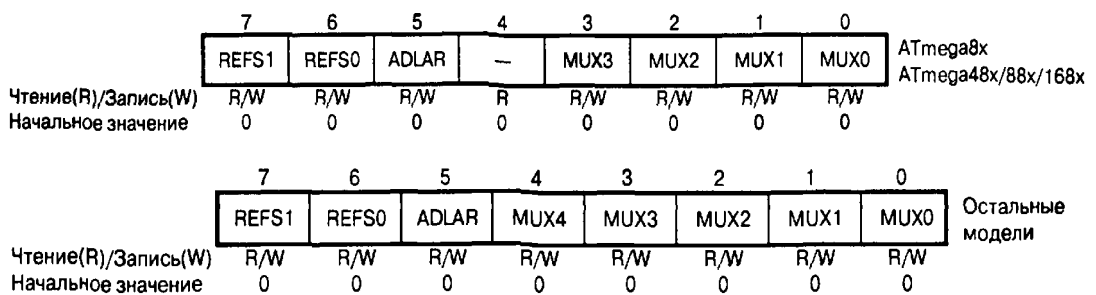


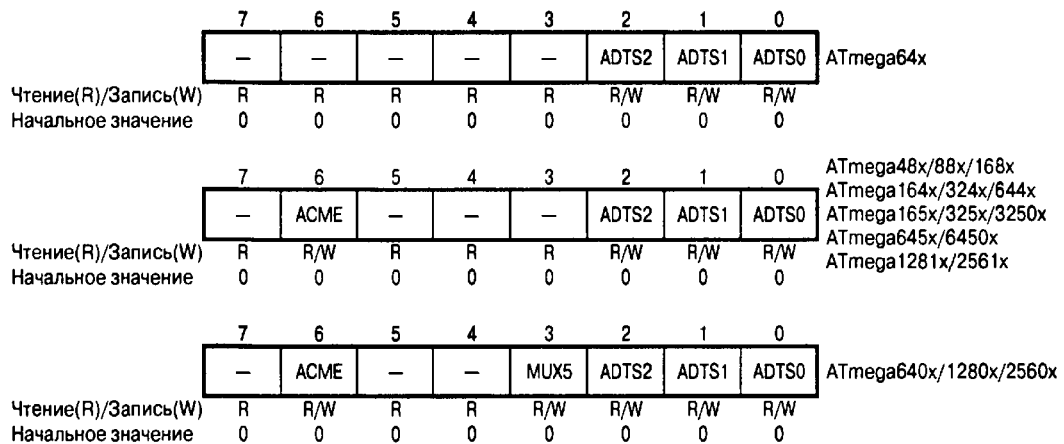
Рис. 9.3. Формат регистра ADMUX

**Таблица 9.3. Биты регистра ADMUX**

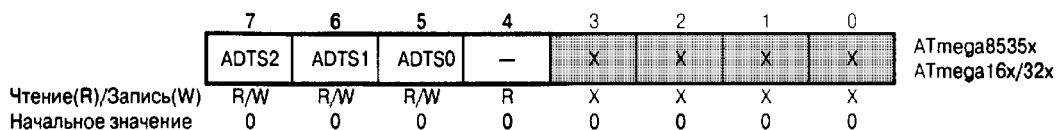
Бит	Название	Описание	Модель
7, 6	REFS1:REFS0	Выбор источника опорного напряжения	Все модели
5	ADLAR	Выравнивание результата преобразования	Все модели
4	—	Зарезервировано	ATmega8x, ATmega48x/88x/168x
	MUX4	Выбор входного канала	Остальные
3...0	MUX3...MUX0	Выбор входного канала	Все модели

Формат регистров ADCSRB и SFIOR приведен на **Рис. 9.4** и **Рис. 9.5** соответственно (неиспользуемые в данном случае биты регистра SFIOR указаны на рисунке как «X»).

Для разрешения работы АЦП необходимо записать лог. 1 в бит ADEN регистра ADCSR, а для выключения — соответственно лог. 0. Если АЦП будет выключен во время цикла преобразования, то преобразование завершено не будет (в регистре данных АЦП останется результат предыдущего преобразования).



**Рис. 9.4. Формат регистра ADCSRB**



**Рис. 9.5. Регистр SFIOR**

В моделях ATmega8x и ATmega128x режим работы АЦП определяется состоянием бита ADFR. Если он установлен в 1, АЦП работает в режиме непрерывного преобразования. В этом режиме запуск каждого следующего преобразования осуществляется автоматически после окончания текущего. Если же бит ADFR сброшен в 0, АЦП работает в режиме одиночного преобразования и запуск каждого преобразования осуществляется по команде пользователя.



В моделях ATmega8x и ATmega128x режим работы АЦП определяется состоянием бита ADFR. Если он установлен в 1, АЦП работает в режиме непрерывного преобразования. В этом режиме запуск каждого следующего преобразования осуществляется автоматически после окончания текущего. Если же бит ADFR сброшен в 0, АЦП работает в режиме одиночного преобразования и запуск каждого преобразования осуществляется по команде пользователя.

Если бит ADATE сброшен в 0, АЦП работает в режиме одиночного преобразования. Если же бит ADTAE установлен в 1, функционирование АЦП определяется содержимым битов ADTS2:0 согласно Табл. 9.4.

**Таблица 9.4. Источник сигнала для запуска преобразования**

ADTS2	ADTS1	ADTS0	Источник стартового сигнала
0	0	0	Режим непрерывного преобразования
0	0	1	Прерывание от аналогового компаратора
0	1	0	Внешнее прерывание INT0
0	1	1	Прерывание по событию «Совпадение» («Совпадение А») таймера/счетчика T0
1	0	0	Прерывание по переполнению таймера/счетчика T0
1	0	1	Прерывание по событию «Совпадение В» таймера/счетчика T1
1	1	0	Прерывание по переполнению таймера/счетчика T1
1	1	1	Прерывание по событию «Захват» таймера/счетчика T1

Запуск каждого преобразования в режиме одиночного преобразования, а также запуск первого преобразования в режиме непрерывного преобразования осуществляется установкой в 1 бита ADSC регистра ADCSRA (ADCSR). Запуск преобразования по прерыванию осуществляется при установке в 1 флага выбранного прерывания. Бит ADSC регистра ADCSRA при этом аппаратно устанавливается в 1. Запуск преобразования в этих режимах также может быть осуществлен установкой бита ADSC регистра ADCSRA в 1.

В режимах одиночного и непрерывного преобразований цикл преобразования начинается по первому нарастающему фронту тактового сигнала после установки бита ADSC. Если используется запуск по прерыванию, то цикл преобразования начинается по первому нарастающему фронту тактового сигнала после установки флага выбранного прерывания. Причем в момент установки этого флага осуществляется сброс предделителя модуля АЦП. Тем самым обеспечивается фиксированная задержка между генерацией запроса на прерывание и началом цикла преобразования. Обратите внимание, что преобразование запускается при установке соответствующего флага, т. е. даже если само прерывание запрещено.

Длительность цикла составляет 13 тактов при использовании несимметричного входа и 13 либо 14 тактов при использовании дифференциального входа (разные значения связаны с работой схемы синхронизации); выборка и запоминание входного сигнала осуществляется в течение первых 1.5 и 2.5 тактов соответственно. Через 13 (14) тактов преобразование

завершается, бит ADSC аппаратно сбрасывается в 0 (в режиме одиночного преобразования), и результат преобразования сохраняется в регистре данных АЦП. Одновременно устанавливается флаг прерывания ADIF регистра ADCSR и генерируется запрос на прерывание. Как и флаги остальных прерываний, флаг ADIF сбрасывается аппаратно при запуске подпрограммы обработки прерывания от АЦП или программно, записью в него лог. 1. Разрешение прерывания осуществляется установкой в 1 бита ADIE регистра ADCSR при установленном флаге I регистра SREG.

Если АЦП работает в режиме непрерывного преобразования, то новый цикл начнется сразу же после записи результата. В режиме одиночного преобразования новое преобразование может быть запущено сразу же после сброса бита ADSC (до сохранения результата текущего преобразования). Однако реально цикл преобразования начнется не ранее чем через один такт после окончания текущего преобразования. Временные диаграммы, иллюстрирующие сказанное, приведены на Рис. 9.6.

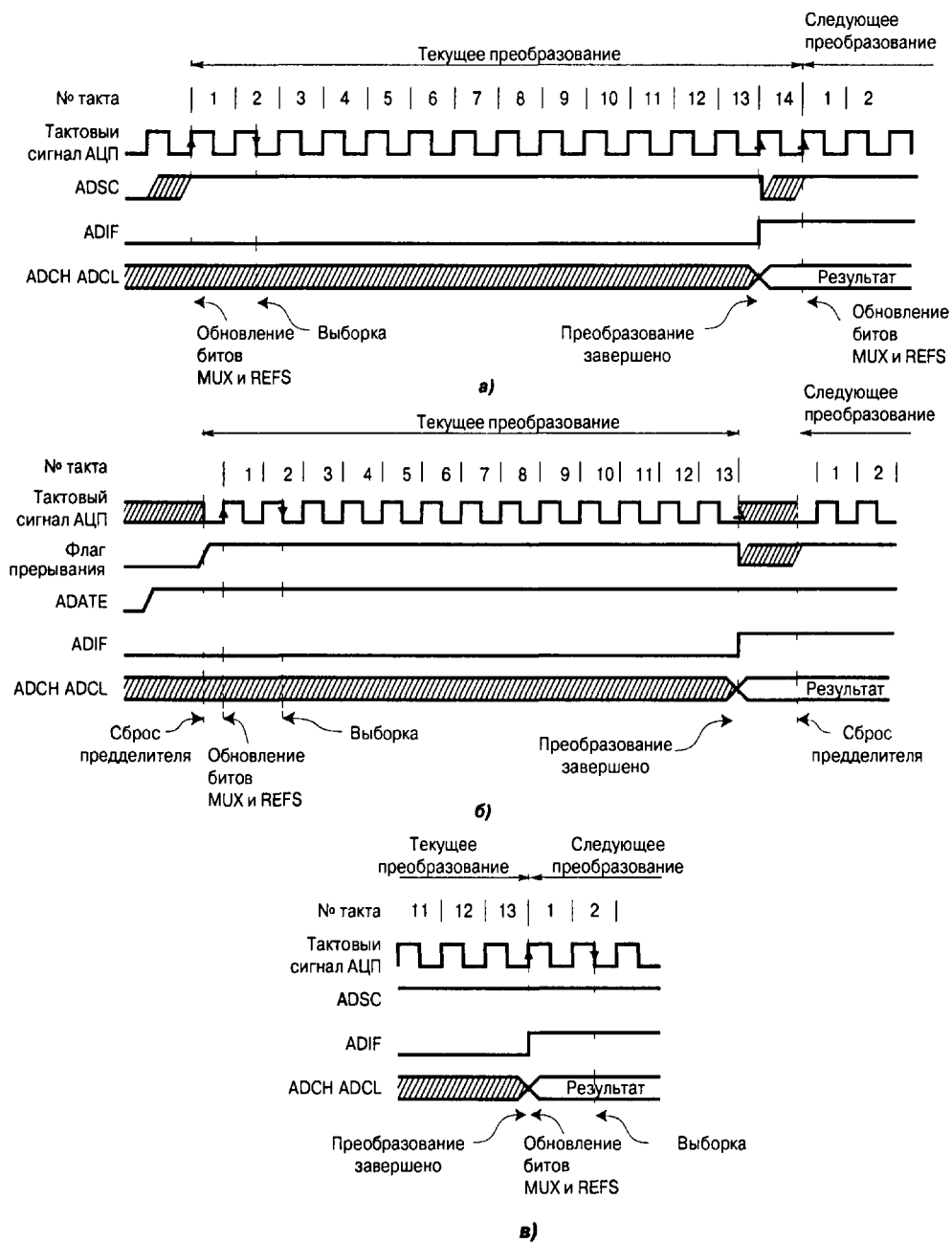
При первом запуске после включения АЦП для выполнения преобразования потребуется 25 тактов, т. е. на 12 тактов больше, чем обычно. В течение этих 12 тактов выполняется «холостое» преобразование, во время которого производится инициализация АЦП (Рис. 9.7).

Отдельно следует сказать об использовании режима запуска по прерыванию совместно с дифференциальными каналами. В этом случае АЦП необходимо выключать между преобразованиями, чтобы избежать некорректных измерений, связанных с неопределенностью момента сброса делителя АЦП. В результате выключения и включения АЦП между преобразованиями будут выполняться только «длинные» преобразования, результаты которых всегда будут корректными.

Для формирования тактовой частоты модуля АЦП в нем имеется отдельный делитель. Коэффициент деления делителя и соответственно длительность преобразования определяются состоянием битов ADPS2...ADPS0 регистра ADCSRA/ADCSR (см. Табл. 9.5).

Наибольшая точность преобразования достигается, если тактовая частота модуля АЦП находится в диапазоне 50...200 кГц. Соответственно, коэффициент деления делителя рекомендуется выбирать таким, чтобы тактовая частота модуля АЦП находилась в указанном диапазоне. Если же точности преобразования меньше 10 битов достаточно, можно использовать более высокую частоту, увеличивая тем самым частоту выборки.

В зависимости от модели выводы микроконтроллера, подключенные к входу АЦП, определяются состоянием битов MUX3:0, MUX4:0 либо MUX5:0 (MUX5 — в ADCSRB, MUX4...MUX0 — в ADMUX). Для каналов с дифференциальным входом указанные биты определяют также коэффициент предварительного усиления входного сигнала. Соответствие установок этих битов используемым входам АЦП приведено в Табл. 9.6...9.8.



**Рис. 9.6.** Временные диаграммы работы АЦП в режиме одиночного преобразования (а), в режиме запуска по прерыванию (б) и в режиме непрерывного преобразования (в)

**Таблица 9.5.** Задание коэффициента деления делителя АЦП

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

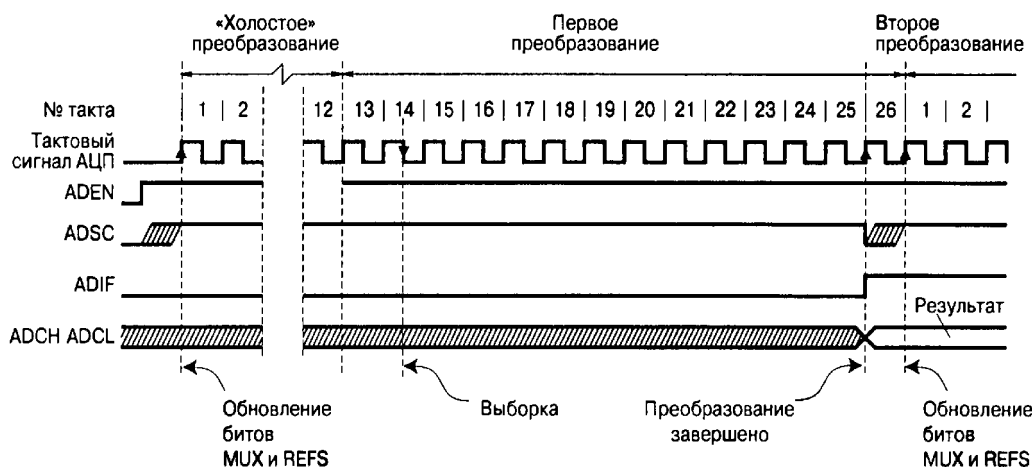


Рис. 9.7. Временные диаграммы работы АЦП при первом преобразовании (режим одиночного преобразования)

Таблица 9.6. Управление входным мультиплексором в моделях АТmega8х и АТmega48х/88х/168х

MUX3...MUX0	Несимметричный вход	MUX3...MUX0	Несимметричный вход
0000	ADC0	0110	ADC6 <sup>1)</sup>
0001	ADC1	0111	ADC7 <sup>1)</sup>
0010	ADC2	1000...1101	Зарезервировано
0011	ADC3	1110	1.22 В (1.1 В <sup>2)</sup> )
0100	ADC4	1111	0 В (GND)
0101	ADC5		

<sup>1)</sup> Только в корпусах TQFP-32 и MLF-32.  
<sup>2)</sup> В моделях АТmega48х/88х/168х.

Следует отметить, что предварительный усилитель, используемый каналами с дифференциальным входом, имеет встроенную схему коррекции смещения. Оставшаяся после коррекции величина смещения может быть учтена программным способом. Для этого следует оба входа дифференциального усилителя подключить к одному и тому же выводу микроконтроллера (Табл. 9.7 и Табл. 9.8), а затем вычитать полученное значение из результата последующих преобразований. Таким образом, ошибка смещения может быть снижена до величины, меньшей 1 LSB.

Состояние битов MUX5...MUX0 можно изменить в любой момент, однако если это будет сделано во время цикла преобразования, то смена канала произойдет только после завершения преобразования. Благодаря этому в режиме непрерывного преобразования можно легко осуществлять последовательное преобразование сигналов нескольких каналов.

Отдельно следует сказать о каналах с дифференциальным входом. После смены таких каналов первое измерение следует производить не ранее чем через 125 мкс после выбора канала. Указанное время требуется для установления значения коэффициента усиления предусилителя. Соответственно, значения, измеренные до истечения этого срока, не могут считаться достоверными. Кроме того, при переключении на канал с изменяемым коэффициентом усиления результат первого преобразования может

иметь пониженную точность из-за времени установления автоматической схемы коррекции смещения. Поэтому будет лучше считать результат первого преобразования некорректным.

В новых микроконтроллерах, а именно в ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x, ATmega325x/3250x/645x/6450x и ATmega640x/1280x/1281x/2560x/2561x, имеется возможность отключения входных цифровых буферов на выводах ADC0...ADC15 в случае, если соответствующие выводы используются только для считывания аналоговых сигналов. При отключенных цифровых буферах уменьшается общий ток потребления микроконтроллера, а соответствующие биты регистров PINx всегда читаются как 0.

Отключение цифровых буферов на входах ADC0...ADC7 осуществляется записью лог. 1 соответственно в биты ADC0D...ADC7D регистра DIDR0, расположенного по адресу (\$7E). А отключение буферов на входах ADC8...ADC15 (модели ATmega640x/1280x/2560x) осуществляется записью лог. 1 в биты ADC8D...ADC15D регистра DIDR2, расположенного по адресу (\$7D). Формат этих регистров приведен на Рис. 9.8.

Как уже было отмечено, в модуле АЦП могут использоваться различные источники опорного напряжения (ИОН). Выбор конкретного источника опорного напряжения осуществляется с помощью битов REFS1:REFS0 регистра ADMUX (см. Табл. 9.9).

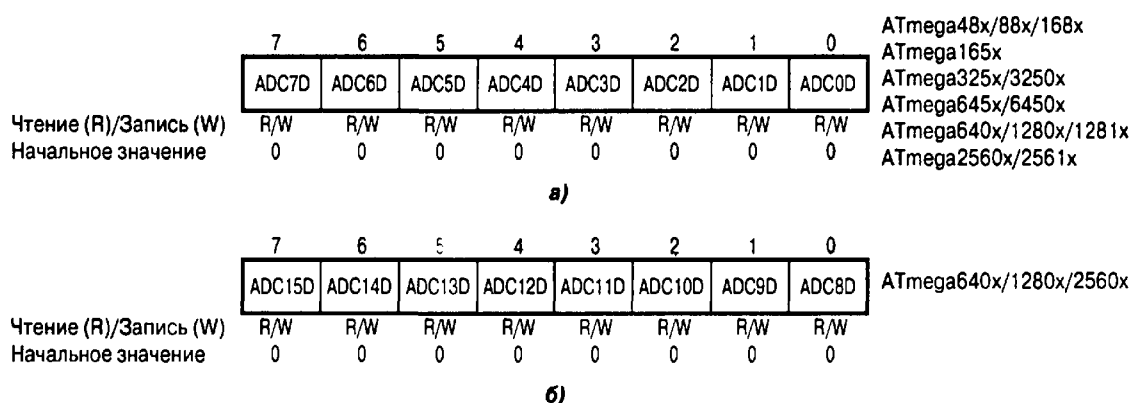


Рис. 9.8. Формат регистров DIDR0 (а) и DIDR2 (б)

Поскольку внутренний ИОН соединяется с выводом AREF микроконтроллера, при его использовании к выводу AREF желательно подключить внешний фильтрующий конденсатор для повышения помехозащищенности. Кроме того, при использовании канала с дифференциальным входом первое измерение после смены источника опорного напряжения следует производить не ранее чем через 125 мкс. Указанное время требуется для установления значения коэффициента усиления предусилителя.

**Таблица 9.9. Выбор источника опорного напряжения**

REFS1	REFS0	Источник опорного напряжения <sup>1)</sup>	Модель
0	0	Внешний ИОН, подключенный к выводу AREF; внутренний ИОН отключен	Все модели
0	1	Напряжение питания $V_{CC}$	Все модели
1	0	Внутренний ИОН напряжением 1.1 В <sup>2)</sup>	ATmega164x/324x/644x, ATmega640x/1280x/1281x, ATmega2560x/2561x
		Зарезервировано	Остальные модели
1	1	Внутренний ИОН напряжением 1.1 В <sup>2)</sup>	ATmega48x/88x/168x, ATmega165x/325x/3250x, ATmega645x/6450x
		Внутренний ИОН напряжением 2.56 В <sup>2)</sup>	Остальные модели
<sup>1)</sup> При работе с усилением 10x или 200x в качестве внутреннего ИОН можно использовать только ИОН, подключаемый при REFS1:0 = 11. <sup>2)</sup> Если к выводу AREF подключен источник напряжения, то эти варианты использоваться не могут.			

### 9.3. Результат преобразования

После завершения преобразования (при установке в 1 флага ADIF регистра ADCSR) его результат сохраняется в регистре данных АЦП. Поскольку АЦП — 10-битный, этот регистр физически размещен в двух регистрах ввода/вывода ADCH:ADCL, доступных только для чтения. Эти регистры расположены по адресам \$05:\$04 (\$25:\$24) в моделях ATmega8535x, ATmega8x/16x/32x/64x/128x и (\$79:\$78) в остальных моделях. При включении микроконтроллера в регистре данных АЦП содержится значение \$0000.

По умолчанию результат преобразования выравнивается вправо (старшие 6 битов регистра ADCH — незначащие). Однако он может выравниваться и влево (младшие 6 битов регистра ADCL — незначащие). Для управления выравниванием результата преобразования служит бит ADLAR регистра ADMUX. Если этот бит установлен в 1, результат преобразования выравнивается по левой границе 16-битного слова, если сброшен в 0 — по правой границе.

Обращение к регистрам ADCH и ADCL для получения результата преобразования должно выполняться в определенной последовательности: сначала необходимо прочитать регистр ADCL, а затем ADCH. Это требование связано с тем, что после обращения к регистру ADCL процессор блокирует доступ к регистрам данных со стороны АЦП до тех пор, пока не будет прочитан регистр ADCH. Благодаря этому можно быть уверенным, что при чтении регистров в них будут находиться составляющие одного и того же результата. Соответственно, если очередное преобразование завершится до обращения к регистру ADCH, результат преобразования будет потерян. С другой стороны, если результат преобразования выравнивается влево и достаточно 8-битной точности, то для получения результата можно прочитать только содержимое регистра ADCH.

Для каналов с несимметричным входом результат преобразования определяется выражением

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}},$$

где  $V_{IN}$  — значение входного напряжения, а  $V_{REF}$  — величина опорного напряжения.

Для каналов с дифференциальным входом результат преобразования определяется выражением

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot K \cdot 512}{V_{REF}},$$

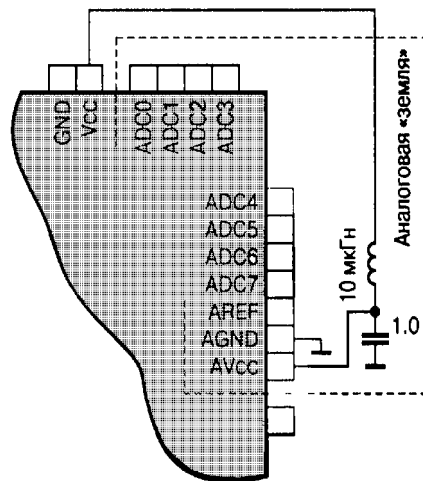
где  $V_{POS}$  — величина напряжения на положительном входе,  $V_{NEG}$  — величина напряжения на отрицательном входе, а  $K$  — коэффициент усиления. Результат преобразования представляется в этом случае в дополнительном коде, а его значение лежит в диапазоне от \$200 (–512) до \$1FF (+512).

## 9.4. Повышение точности преобразования

В этом разделе приведены некоторые рекомендации, позволяющие в наибольшей степени использовать возможности АЦП. Прежде всего, для минимизации погрешности самого АЦП необходимо правильно выбрать тактовую частоту преобразования. С этой же целью на входе АЦП рекомендуется устанавливать фильтр нижних частот. Кроме того, при разработке конструкции и разводке печатной платы следует придерживаться общих правил проектирования цифро-аналоговых устройств:

1. На печатной плате необходимо предусмотреть область сплошной металлизации под аналоговую «землю». Аналоговая часть микроконтроллера и аналоговая часть всего устройства должна располагаться над этой областью. Аналоговая и цифровая «земли» должны соединяться друг с другом только в одной точке печатной платы.
2. Проводники, по которым распространяются аналоговые сигналы, должны быть как можно короче и располагаться над аналоговой «землей». Кроме того, они должны быть размещены как можно дальше от быстродействующих цифровых цепей.
3. Вывод AVCC микроконтроллера должен подключаться к источнику питания  $V_{CC}$  через LC-фильтр, как показано на Рис. 9.9 (расположение выводов показано условно).
4. Если какие-либо выводы АЦП используются как цифровые выходы, они не должны переключаться во время преобразования.

Для сведения к минимуму электромагнитных помех, наводимых ядром процессора, во всех рассматриваемых микроконтроллерах имеется дополнительный «спящий» режим — ADC Noise Reduction (режим снижения шумов АЦП). В этом режиме из всех периферийных устройств функционируют только АЦП и сторожевой таймер. Для той же цели (но с меньшим эффектом) может быть использован режим Idle. Для использования АЦП в любом из указанных режимов необходимо выполнить следующее:



**Рис. 9.9.** Подключение цепей питания АЦП

1. Убедиться, что АЦП включен и не занят преобразованием. Затем переклЮчить АЦП в режим одиночного преобразования и разрешить прерывание от АЦП.
2. Перевести микроконтроллер в режим ADC Noise Reduction (или Idle). Сразу же после остановки процессора начнется цикл преобразования.
3. По завершении преобразования будет сгенерировано прерывание от АЦП, которое переведет микроконтроллер в рабочий режим и начнется выполнение подпрограммы обработки этого прерывания.

#### **Оборудование и материалы.**

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже лабораторных работ по исследованию программирования микроконтроллеров на различных языках программирования выполняется на учебном макетном лабораторном комплексе РТМТЛ.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микроконтроллер (Atmega8535, Atmega16, Atmega32 или совместимый); резистивно-диодный последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; набор из 8 светодиодов, подключенных к выводам 33 – 40, см. рис. 2.1; набор из 8 кнопок с фиксацией нормально разомкнутых (кнопка нажата — замкнута; отжата — разомкнута), подключенных к выводам 22 — 29; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к COM – порту ПЭВМ ТОЛЬКО проводом типа COM 9m/9f («мама» - «папа»). В данной работе на примере микроконтроллера AVR (Atmega8535, Atmega16, Atmega32 или совместимого) предлагается освоить простейшие приемы программирования на следующих языках:

- 1) Программирование микроконтроллера на языке ассемблер в интегральной среде разработки AVR Studio
- 2) Программирование микроконтроллера на языке Basic в интегральной среде разработки Vascom-AVR
- 3) Программирование микроконтроллера на языке Си в интегральной среде разработки AVR Studio с подключенным Си компилятором GCC из WinAVR
- 4) Программирование микроконтроллера в среде графического ассемблера Algorithm Builder.





### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.

**2. Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR.**

3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.

4. Загрузить операционную систему согласно стандартным процедурам загрузки.

5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder.

6. Запустить программу Algorithm Builder для работы с учебной установкой для данного эксперимента пользуясь ярлыком на рабочем столе либо другим способом, указанным лаборантом. Для работы можно воспользоваться комплексной программной оболочкой LabVisual для РТМТЛ-1-5.

6. Подключить разъём «ПРОГРАММАТОР» учебного прибора к СОМ – порту ПЭВМ проводом типа СОМ 9m/9f («мама» - «папа»)

7. Настроить программу Algorithm Builder для работы с данным СОМ портом  
**ОПЦИИ-ОПЦИИ СРЕДЫ-ПОРТ**

Задание . В данном примере кнопки с фиксацией подключены к порту А (выводы PA0-PA7), логическое состояние которого представляет собой 1 байт данных. При этом состояние нажатой кнопки это логический 0, состояние отжатой кнопки — логическая 1 соответствующего вывода. К данным порта А прибавляется константа (число 2 в среде графического ассемблера Algorithm Builder; число 3 в среде разработки AVR Studio ассемблер; число 4 в среде разработки Bascom-AVR и число 5 в среде разработки AVR Studio Си с подключенным компилятором GCC из WinAVR). Полученный результат копируется в порт С (PC0-PC7), к которому подключены светодиоды, визуализирующие логическое состояние соответствующих выводов (PC0-PC7).

### Оформление отчета

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

### Контрольные вопросы

1. Какой уровень сигнала у цифровых выходов МК?
2. Какие типы команд имеет микроконтроллер AVR семейства Mega?
3. Какие типы команд изменяют флаги микроконтроллера?
4. Какие виды адресации использует микроконтроллер AVR?
5. Зачем нужны команды передачи управления?
6. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?

### Список литературы, рекомендуемый к использованию по данной теме:

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. -

2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
3. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа 5

### Изучение работы и программирования блоков последовательной передачи данных микроконтроллера AVR с помощью инструментальных средств

#### Цель работы

Изучение работы и программирования блоков последовательной передачи данных микроконтроллера AVR с помощью инструментальных средств AVR Atmega8535, Atmega16, Atmega32 и совместимого изучить основные приёмы программирования и отладки микроконтроллера.

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контроллеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

## Общие сведения

Модуль универсального последовательного интерфейса (Universal Serial Interface — USI) реализован только в моделях ATmega165x и ATmega325x/3250x/645x/6450x. Этот модуль является своего рода «полу-фабрикатом», предоставляющим базовые аппаратные ресурсы, необходимые для осуществления обмена по последовательному каналу. Используя данный модуль, можно достичь гораздо большей скорости передачи и получить более компактный код, нежели при чисто программной реализации различных протоколов обмена.

Обращаю ваше внимание на то, что в данной главе мы будем рассматривать собственно модуль USI. Принципы обмена по трехпроводной (SPI) и двухпроводной (TWI) шинам были подробно рассмотрены в двух предыдущих главах.

Упрощенная структурная схема модуля USI приведена на **Рис. 12.1**.

В общей сложности модуль USI задействует три линии ввода/вывода микроконтроллера:

- PE6 — выход данных (DO). Используется в трехпроводном режиме (SPI);
- PE5 — вход данных (DI)/линия данных (SDA);
- PE4 — вход/выход тактового сигнала (USCK)/линия тактового сигнала (SCL).

В 8-битном сдвиговом регистре содержатся входящие и исходящие данные. Старший бит сдвигового регистра подключается, в зависимости от режима работы, к одной из двух линий данных модуля. Входящие биты всегда считываются с вывода DI, независимо от конфигурации модуля. Четырехбитный счетчик, показанный на **Рис. 12.1**, доступен как для чтения, так и для записи. Кроме того, при его переполнении может генерироваться прерывание. Поскольку сдвиговый регистр и счетчик используют один и тот же тактовый сигнал, счетчик может считать число переданных или принятых битов и сгенерировать прерывание по окончании процесса обмена. Обратите внимание, что при использовании внешнего тактового сигнала изменение состояния счетчика происходит по каждому фронту сигнала, т. е. счетчик считает количество фронтов, а не битов. Тактовый сигнал может сниматься с входа USCK, с выхода блока сравнения таймера/счетчика T0 или формироваться программно.

Блок управления тактовым сигналом используется в двухпроводном режиме и может генерировать прерывание при обнаружении на шине состояния СТАРТ. Кроме того, он может формировать на шине циклы ожидания, удерживая на линии SCL НИЗКИЙ уровень после обнаружения состояния СТАРТ или после переполнения счетчика.

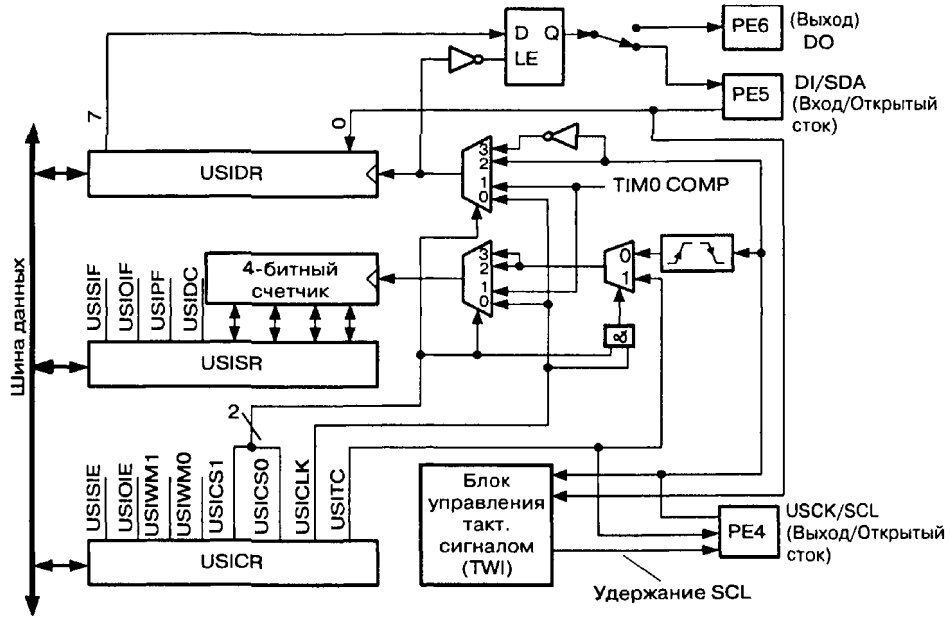


Рис. 12.1. Структурная схема модуля USI

### 13.2. Использование модулей USART

Упрощенная структурная схема одного модуля USART приведена на Рис. 13.1.

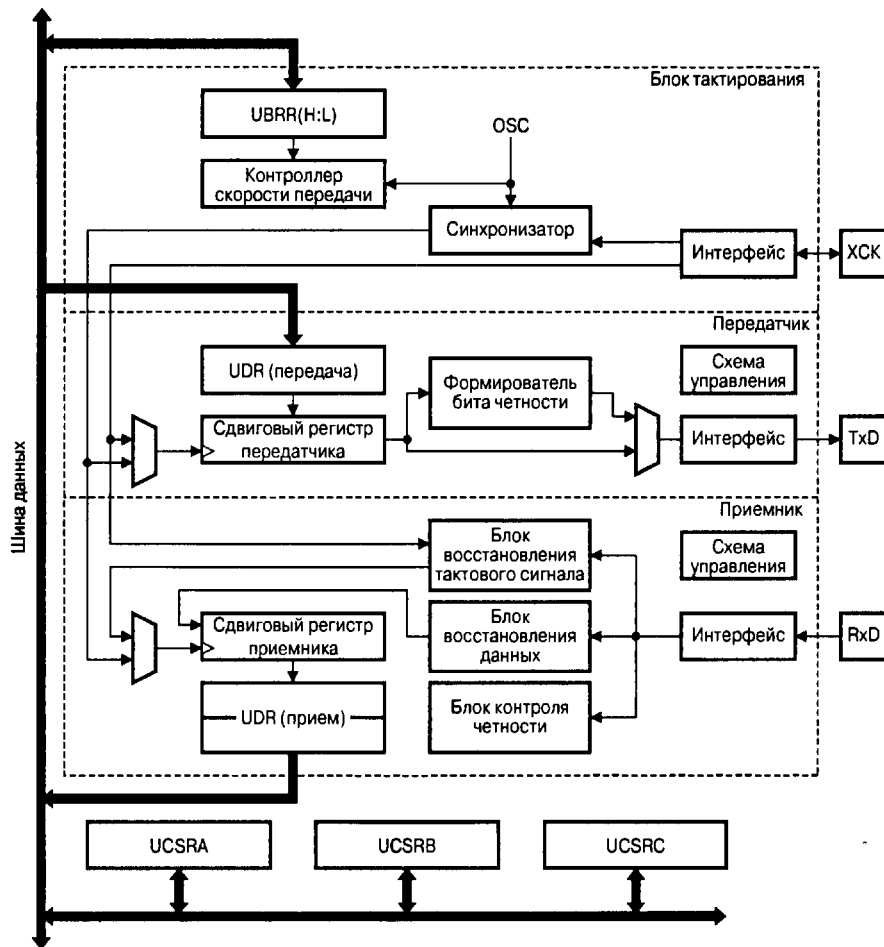


Рис. 13.1. Структурная схема модуля USART

Как показано на рисунке, модуль состоит из трех основных частей: блока тактирования, блока передатчика и блока приемника. Блок тактирования модулей USART содержит схему синхронизации, которая используется при работе в синхронном режиме, и контроллер скорости передачи.

Блок передатчика включает одноуровневый буфер, сдвиговый регистр, схему формирования бита четности и схему управления. Блок приемника, в свою очередь, содержит схемы восстановления тактового сигнала и данных, схему контроля четности, двухуровневый буфер, сдвиговый регистр, а также схему управления.

Буферные регистры приемника и передатчика располагаются по одному адресу пространства ввода/вывода и обозначаются как регистр данных UDR (UDR<sub>*n*</sub>). В этом регистре хранятся младшие 8 битов принимаемых и передаваемых данных. При чтении регистра UDR выполняется обращение к буферному регистру приемника, при записи — к буферному регистру передатчика. Размещение регистров данных UDR для различных моделей микроконтроллеров приведено в Табл. 13.3.

В модулях USART буфер приемника является двухуровневым (FIFO-буфер), изменение состояния которого происходит при любом обращении к регистру UDR. В связи с этим не следует использовать регистр UDR в качестве операндов команд типа «чтение/модификация/запись» (SBI и CBI). Кроме того, следует быть очень аккуратными при использо-

**Таблица 13.3. Размещение регистров данных модулей USART**

Регистр	Адрес	ATmega8515x/8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/3250x, ATmega645x/6450x	ATmega640x/1280x/2560x	ATmega1281x/2561x	Описание
UDR	\$0C (\$2C)	•	•								Регистр данных USART
	(\$C6)							•			
UDR0	\$0C (\$2C)			•		•					Регистр данных USART0
	(\$C6)				•		•		•	•	
UDR1	\$03 (\$23)					•					Регистр данных USART1
	(\$9C)			•							
	(\$CE)						•		•	•	
UDR2	(\$D6)								•		Регистр данных USART2
UDR3	(\$136)								•		Регистр данных USART3

Формат регистров UCSRA (UCSR<sub>nA</sub>), UCSRB (UCSR<sub>nB</sub>) и UCSRC (UCSR<sub>nC</sub>) приведен на Рис. 13.2...13.4, а значение битов этих регистров описано в Табл. 13.5...13.7 соответственно.

Регистр	Адрес	ATmega8515x/8535x	ATmega8x/16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x, ATmega325x/3250x, ATmega645x/6450x	ATmega640x/1280x/2560x	ATmega1281x/2561x	Описание
UCSR1C	\$3C (\$5C)					•					Регистр С управления USART1
	(\$9D)			•							
	(\$CA)						•		•	•	
UCSR2A	(\$D0)								•		Регистр А управления USART2
UCSR2B	(\$D1)								•		Регистр В управления USART2
UCSR2C	(\$D2)								•		Регистр С управления USART2
UCSR3A	(\$130)								•		Регистр А управления USART3
UCSR3B	(\$131)								•		Регистр В управления USART3
UCSR3C	(\$132)								•		Регистр С управления USART3

	7	6	5	4	3	2	1	0
	RXC <sub>n</sub>	TXC <sub>n</sub>	UDRE <sub>n</sub>	FE <sub>n</sub>	DOR <sub>n</sub>	UPE <sub>n</sub>	U2X <sub>n</sub>	MPCM <sub>n</sub>
Чтение(R)/Запись(W)	R	R/W	R	R	R	R	R/W	R/W
Начальное значение	0	0	1	0	0	0	0	0

Рис. 13.2. Формат регистров UCSRA (UCSR<sub>nA</sub>)



**Таблица 13.5. Биты регистров UCSRA (UCSRnA)**

Бит	Название	Описание
7	RXC (RXC <sub>n</sub> )	<b>Флаг завершения приема.</b> Флаг устанавливается в 1 при наличии непрочитанных данных в буфере приемника (регистр данных UDR). Сбрасывается аппаратно после опустошения буфера. Если бит RXCIE (RXCIE <sub>n</sub> ) регистра UCSRB (UCSRnB) установлен, то при установке флага генерируется запрос на прерывание «прием завершен»
6	TXC (TXC <sub>n</sub> )	<b>Флаг завершения передачи.</b> Флаг устанавливается в 1 после передачи всех битов посылки из сдвигового регистра передатчика при условии, что в регистр данных UDR не было загружено новое значение. Если бит TXCIE (TXCIE <sub>n</sub> ) регистра UCSRB (UCSRnB) установлен, то при установке флага генерируется прерывание «передача завершена». Флаг сбрасывается аппаратно при выполнении подпрограммы обработки прерывания или программно, записью в него лог. 1
5	UDRE (UDRE <sub>n</sub> )	<b>Флаг опустошения регистра данных.</b> Данный флаг устанавливается в 1 при пустом буфере передатчика (после пересылки байта из регистра данных UDR в сдвиговый регистр передатчика). Установленный флаг означает, что в регистр данных можно загружать новое значение. Если бит UDRIE (UDRIE <sub>n</sub> ) регистра UCSRB (UCSRnB) установлен, генерируется запрос на прерывание «регистр данных пуст». Флаг сбрасывается аппаратно, при записи в регистр данных
4	FE (FE <sub>n</sub> )	<b>Флаг ошибки кадрирования.</b> Флаг устанавливается в 1 при обнаружении ошибки кадрирования, т. е. если первый стоп-бит принятой посылки равен 0. Флаг сбрасывается при приеме стоп-бита, равного 1
3	DOR (DOR <sub>n</sub> )	<b>Флаг переполнения.</b> Флаг устанавливается в 1, если в момент обнаружения нового старт-бита в сдвиговом регистре приемника находится последнее принятое слово, а буфер приемника полон (содержит два байта). Флаг сбрасывается при пересылке принятых данных из сдвигового регистра приемника в буфер
2	UPE (UPE <sub>n</sub> )	<b>Флаг ошибки контроля четности.</b> Флаг устанавливается в 1, если в данных, находящихся в буфере приемника, выявлена ошибка контроля четности. При отключенном контроле четности этот бит постоянно сброшен в 0
1	U2X (U2X <sub>n</sub> )	<b>Удвоение скорости обмена.</b> Если этот бит установлен в 1, то коэффициент деления предделителя контроллера скорости передачи уменьшается с 16 до 8, удваивая тем самым скорость асинхронного обмена по последовательному каналу. Этот бит используется только при асинхронном режиме работы и в синхронном режиме должен быть сброшен
0	MPCM (MPCM <sub>n</sub> )	<b>Режим мультипроцессорного обмена.</b> Если этот бит установлен в 1, ведомый микроконтроллер ожидает приема кадра, содержащего адрес. Кадры, не содержащие адреса устройства, игнорируются

Примечание.  $n = 0, 1, 2$  или  $3$ .

	7	6	5	4	3	2	1	0
	RXCIE <sub>n</sub>	TXCIE <sub>n</sub>	UDRIE <sub>n</sub>	RXEN <sub>n</sub>	TXEN <sub>n</sub>	UCSZ <sub>n2</sub>	RXB8 <sub>n</sub>	TXB8 <sub>n</sub>
Чтение(R)/Запись(W)	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Начальное значение	0	0	0	0	0	0	1	0

**Рис. 13.3. Формат регистров UCSRB (UCSRnB)**

**Таблица 13.6. Биты регистров UCSRB (UCSRnB)**

Бит	Название	Описание
7	RXCIE (RXCIE $n$ )	<b>Разрешение прерывания по завершении приема.</b> Если данный бит установлен в 1, то при установке флага RXC (RXC $n$ ) регистра UCSRA (UCSR $n$ A) генерируется прерывание «прием завершен» (если флаг I регистра SREG установлен в 1)
6	TXCIE (TXCIE $n$ )	<b>Разрешение прерывания по завершении передачи.</b> Если данный бит установлен в 1, то при установке флага TXC (TXC $n$ ) регистра UCSRA (UCSR $n$ A) генерируется прерывание «передача завершена» (если флаг I регистра SREG установлен в 1)
5	UDRIE (UDRIE $n$ )	<b>Разрешение прерывания при очистке регистра данных UART.</b> Если данный бит установлен в 1, то при установке флага UDRE (UDRE $n$ ) регистра UCSRA (UCSR $n$ A) генерируется прерывание «регистр данных пуст» (если флаг I регистра SREG установлен в 1)
4	RXEN (RXEN $n$ )	<b>Разрешение приема.</b> При установке этого бита в 1 разрешается работа приемника USART и переопределяется функционирование вывода RXD (RXD $n$ ). При сбросе бита RXEN (RXEN $n$ ) работа приемника запрещается, а его буфер сбрасывается. Значения флагов TXC(TXC $n$ ), DOR (DOR $n$ ) и FE (FE $n$ ) при этом становятся недействительными
3	TXEN (TXEN $n$ )	<b>Разрешение передачи.</b> При установке этого бита в 1 разрешается работа передатчика UART и переопределяется функционирование вывода TXD (TXD $n$ ). Если бит сбрасывается в 0 во время передачи, то выключение передатчика произойдет только после завершения передачи данных, находящихся в сдвиговом регистре и буфере передатчика
2	UCSZ2 (UCSZ $n$ 2)	<b>Формат посылок.</b> Этот бит совместно с битами UCSZ1:0 (UCSZ $n$ 1:0) регистра UCSRC (UCSR $n$ C) используется для задания размера слов данных, передаваемых по последовательному каналу
1	RXB8 (RXB8 $n$ )	<b>8-й бит принимаемых данных.</b> При использовании 9-битных слов данных этот бит содержит значение старшего бита принятого слова. Содержимое этого бита должно быть считано до прочтения регистра данных UDR
0	TXB8 (TXB8 $n$ )	<b>8-й бит передаваемых данных.</b> При использовании 9-битных слов данных содержимое этого бита является старшим битом передаваемого слова. Требуемое значение должно быть занесено в этот бит до загрузки байта данных в регистр UDR

**Примечание.**  $n = 0, 1, 2$  или  $3$ .

Таблица 13.7. Биты регистров UCSRC (UCSRnC)

Бит	Название	Описание
7	UMSELn1	<b>Режим работы USART</b> (модели 48х/88х/168х, 164х/324х/644х и 640х/1280х/1281х/2560х/2561х). Совместно с битом UMSELn0 определяет режим работы модуля USART
	URSEL <sup>1)</sup> (URSELn)	<b>Выбор регистра.</b> Этот бит определяет, в какой из регистров модуля производится запись. Если бит установлен в 1, обращение производится к регистру UCSRC (UCSRnC). Если же бит сброшен в 0, обращение производится к регистру UBRRH (UBRRnH). Подробнее — см. следующий подраздел
6	UMSEL (UMSELn)	<b>Режим работы USART.</b> Если бит сброшен в 0, то модуль работает в асинхронном режиме. Если бит установлен в 1, то модуль работает в синхронном режиме
	UMSELn0	<b>Режим работы USART</b> (модели 48х/88х/168х, 164х/324х/644х и 640х/1280х/1281х/2560х/2561х). Совместно с битом UMSELn1 определяет режим работы модуля USART
5	UPM1 (UPMn1)	<b>Режим работы схемы контроля и формирования бита четности.</b> Эти биты определяют функционирование схем контроля и формирования бита четности (см. подраздел 13.2.2)
4	UPM0 (UPMn0)	
3	USBS (USBSn)	<b>Количество стоп-битов.</b> Этот бит определяет количество стоп-битов, посылаемых передатчиком. Если бит сброшен в 0, передатчик посылает 1 стоп-бит, если установлен в 1, то 2 стоп-бита. Для приемника содержимое этого бита безразлично
2	UCSZ1 (UCSZn1)	<b>Формат посылок.</b> Совместно с битом UCSZ2 (UCSZn2) регистра UCSRB (UCSRnB) эти биты определяют количество битов данных в посылках (размер слова)
1	UCSZ0 (UCSZn0)	

	7	6	5	4	3	2	1	0	
	—	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	ATmega64x ATmega128x
Чтение(R)/Запись(W)	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Начальное значение	0	0	0	0	0	1	1	0	
	7	6	5	4	3	2	1	0	
	URSELn	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	ATmega8515х/8535х ATmega8х/16х/32х ATmega162х
Чтение(R)/Запись(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Начальное значение	0	0	0	0	0	1	1	0	ATmega165х/325х/3250х ATmega645х/6450х
	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	ATmega48х/88х/168х ATmega164х/324х/644х ATmega640х/1280х/1281х ATmega2560х/2561х
Чтение(R)/Запись(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Начальное значение	0	0	0	0	0	1	1	0	

Рис. 13.4. Формат регистров UCSRC (UCSRnC)

Обратите внимание на то, что в моделях ATmega48х/88х/168х, ATmega164х/324х/644х и ATmega640х/1280х/1281х/2560х/2561х для задания режима работы используется не один бит регистра UCSRC, а два — UMSELn1:0. Зависимость режима работы модуля USART от установок этих битов приведена в Табл. 13.8.

Бит	Название	Описание		
0	UCPOL (UCPOL $n$ )	Полярность тактового сигнала. Значение этого бита определяет момент выдачи и считывания данных на выводах модуля. Бит используется только при работе в синхронном режиме. При работе в асинхронном режиме он должен быть сброшен в 0.		
		UCPOL (UCPOL $n$ )	Выдача данных на вывод TXD (TXD $n$ )	Считывание данных с вывода RXD (RXD $n$ )
		0	Спадающий фронт ХСК (ХСК $n$ )	Нарастающий фронт ХСК (ХСК $n$ )
		1	Нарастающий фронт ХСК (ХСК $n$ )	Спадающий фронт ХСК (ХСК $n$ )
<sup>1)</sup> Зарезервирован в моделях АТmega64х/128х.				

Примечание.  $n = 0, 1, 2$  или  $3$ .

**Таблица 13.8. Управление режимом работы модулей USART моделей АТmega48х/88х/168х, АТmega164х/324х/644х и АТmega640х/1280х/1281х/2560х/2561х**

UMSEL $n$ 1	UMSEL $n$ 0	Режим работы
0	0	Асинхронный USART
0	1	Синхронный USART
1	0	Зарезервировано
1	1	Ведущий шины SPI

### 13.2.1. Скорость приема/передачи

В асинхронном режиме, а также в синхронном режиме при работе в качестве ведущего скорость приема и передачи данных задается контроллером скорости передачи, работающим как делитель системного тактового сигнала с программируемым коэффициентом деления. Коэффициент определяется содержимым регистра контроллера UBRR (UBRR $n$ ). В блок приемника сформированный сигнал поступает напрямую, а в блок передатчика — через дополнительный делитель, коэффициент деления которого (2, 8 или 16) зависит от режима работы модуля USART.

Регистр UBRR является 12-битным и физически размещается в двух регистрах ввода/вывода. Адреса и названия этих регистров для различных моделей микроконтроллеров приведены в Табл. 13.9.

**Таблица 13.9. Размещение регистров контроллера скорости передачи**

Модель	Регистры	Адрес
АТmega8515х/8535х <sup>1)</sup>	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
АТmega8х/16х/32х <sup>1)</sup>	UBRRH:UBRRL	\$20 (\$40):\$09 (\$29)
АТmega64х/128х	UBRR0H:UBRR0L	(\$90):\$09 (\$29)
	UBRR1H:UBRR1L	(\$98):(\$99)
АТmega48х/88х/168х	UBRR0H:UBRR0L	(\$C5):(\$C4)

ATmega162x <sup>1)</sup>	UBRR0H:UBRR0L	\$20 (\$40):\$09 (\$29)
	UBRR1H:UBRR1L	\$3C (\$5C):\$00 (\$20)
ATmega164x/324x/644x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
ATmega165x/325x/3250x/645x/6450x	UBRR0H:UBRR0L	(\$C5):(\$C4)
ATmega640x/1280x/2560x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
	UBRR2H:UBRR2L	(\$D5):(\$D4)
	UBRR3H:UBRR3L	(\$135):(\$134)
ATmega1281x/2561x	UBRR0H:UBRR0L	(\$C5):(\$C4)
	UBRR1H:UBRR1L	(\$CD):(\$CC)
<sup>1)</sup> В этих моделях регистр UBRRH (UBRR <sub>n</sub> H) размещается по тому же адресу, что и регистр UCSRC (UCSR <sub>n</sub> C).		

Обратите внимание на то, что в моделях ATmega8515x/8535x, ATmega8x/16x/32x и ATmega162x регистр UBRRH размещается по тому же адресу, что и регистр управления UCSRC. Поэтому при обращении по этим адресам необходимо выполнить ряд дополнительных действий для выбора конкретного регистра.

При записи регистр определяется состоянием старшего бита записываемого значения URSEL. Если этот бит сброшен в 0, изменяется содержимое регистра UBRRH. Если же старший бит значения установлен в 1, изменяется содержимое регистра управления UCSRC. Приведенные ниже фрагменты программ иллюстрируют сказанное:

#### **Оборудование и материалы.**

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже лабораторных работ по исследованию программирования микроконтроллеров на различных языках программирования выполняется на учебном макетном лабораторном комплексе РТМТЛ.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микроконтроллер (Atmega8535, Atmega16, Atmega32 или совместимый); резистивно-диодный последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; набор из 8 светодиодов, подключенных к выводам 33 – 40, см. рис. 2.1; набор из 8 кнопок с фиксацией нормально разомкнутых (кнопка нажата — замкнута; отжата — разомкнута), подключенных к выводам 22 — 29; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к COM – порту ПЭВМ ТОЛЬКО проводом типа COM 9m/9f («мама» - «папа»). В данной работе на примере микроконтроллера AVR (Atmega8535, Atmega16, Atmega32 или совместимого) предлагается освоить простейшие приемы программирования на следующих языках:

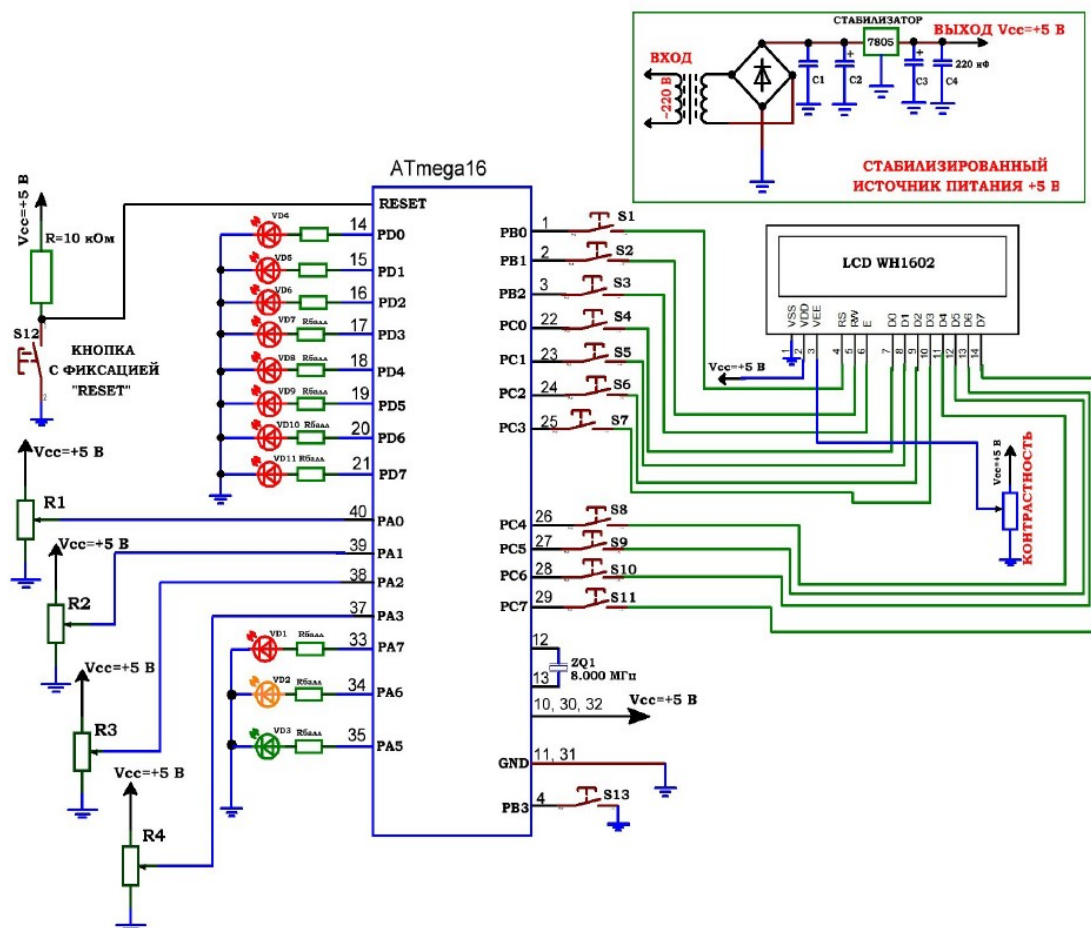


Рис. 4.1. Принципиальная электрическая схема учебного стенда для изучения работы LCD ЖК символического индикатора с микроконтроллером AVR.

### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.
2. Перед началом работы необходимо ознакомиться с документацией на исследуемый микроконтроллер, документацией на исследуемую среду программирования, а также со статьями по данным вопросам, размещенными в соответствующих папках на прилагаемом к учебной установке диске или Flash карте.
3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.
4. Загрузить операционную систему согласно стандартным процедурам загрузки.
5. Открыть демонстрационный пример (проект) в исследуемой среде программирования (AVR Studio, Bascom-AVR, WinAVR, Algorithm Builder). Откомпилировать проект в исследуемой среде и получить готовый бинарный HEX-файл.
6. Подключить разъем «ПРОГРАММАТОР» учебного прибора к COM – порту ПЭВМ проводом типа COM 9m/9f («мама» - «папа»).
7. Прошить тестовую программу (HEX-файл) в кристалл. Т. к. для программирования микроконтроллера (прошивки) используется резистивно-диодный

последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор («ПРОГРАММАТОР ГРОМОВА»), то для прошивки hex-файлов, полученных в средах программирования AVR Studio, Bascom-AVR, WinAVR следует использовать утилиту SinaPROG, поддерживающую данный резистивно-диодный последовательный программатор. При работе в среде Algorithm Builder полученный HEX-файл может быть зашит в микроконтроллер непосредственно из среды.

8. При работе с программой SinaPROG следует избегать пробелов и русских символов в полном имени HEX-файла (имя с учетом полного пути к файлу) и общей длины пути к HEX-файлу более 255 символов. Идеальным вариантом будет копирование полученного HEX-файла в корень диска перед прошивкой. 9. Перед прошивкой следует нажать кнопку с фиксацией «RESET» на учебном приборе.

10. Особенно внимательно следует отнестись к опциям «ЗАПИСЬ Fuse bit» и «ЗАПИСЬ блокирующих бит» в среде Algorithm Builder и в утилите SinaPROG. **ФЛАЖКИ ЭТИХ ОПЦИЙ ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ СНЯТЫ, И ЭТУ ОПЦИЮ В УЧЕБНЫХ ЦЕЛЯХ ИСПОЛЬЗОВАТЬ НЕ РЕКОМЕНДУЕТСЯ, Т. К. НЕ ПРАВИЛЬНО ПРОШИТЫЕ FUSE БИТЫ МОГУТ ПРИВЕСТИ К ДАЛЬНЕЙШЕЙ НЕВОЗМОЖНОСТИ РАБОТЫ С ДАННЫМ МИКРОКОНТРОЛЛЕРОМ!**

11. Отжать кнопку RESET и проверить работу тестовой программы.

12. Исходя из данного учебного примера, составить свои программы в различных средах программирования. Например, программа может прибавлять к данным порта А (РА0-РА7) любое другое произвольное число. Так-же рекомендуется поработать с действиями «умножение» и «вычитание».

13. Для составления программ и подробного изучения работы микроконтроллеров AVR следует обращаться к учебной литературе, полному руководству к микроконтроллерам семейства AVR, а также к документации по средам программирования, используемым в работе.

14. По окончании работы следует закрыть программу и все открытые подпрограммы.

15. Выключить компьютер, нажав на кнопку, находящуюся в крайнем нижнем левом углу экрана. Из доступных действий выбрать «ВЫХОД»--> «ВЫКЛЮЧИТЬ КОМПЬЮТЕР».

16. Отключить установку от сети, поставив переключатели «СЕТЬ» на панели установки в положение «ВЫКЛ» и вынуть сетевые вилки из розеток. \_\_

#### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

#### **Контрольные вопросы**

7. Какой уровень сигнала у цифровых выходов МК?
8. Какие типы команд имеет микроконтроллер AVR семейства Mega?
9. Какие типы команд изменяют флаги микроконтроллера?
10. Какие виды адресации использует микроконтроллер AVR?
11. Зачем нужны команды передачи управления?
12. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?

#### **Список литературы, рекомендуемый к использованию по данной теме:**

11. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
12. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.

13. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
14. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
15. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
16. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
17. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
18. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
19. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
20. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).



## Лабораторная работа 6

### Инструментальные средства разработки программного обеспечения ПЛК. Разработка программ работы с дискретными сигналами микроконтроллера AVR

#### Цель работы

Изучение инструментальных средств разработки программного обеспечения МК.  
Разработка программ работы с дискретными сигналами микроконтроллера AVR..

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контролеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

Учебная графическая среда для разработки программного обеспечения для микроконтроллеров с архитектурой AVR фирмы ATMEL.

Среда Algorithm Builder

Среда предназначена для производства полного цикла разработки начиная от ввода алгоритма, включая процесс отладки и заканчивая программированием кристалла рис. 2.1. Разработка программы может быть как на уровне ассемблера, так и на макроуровне с манипуляцией многобайтными величинами со знаком.

В отличие от классического ассемблера программа вводится в виде алгоритма с древовидными ветвлениями и отображается на плоскости, в двух измерениях. Сеть

условных и безусловных переходов отображается графически, в удобной векторной форме. Это к тому же освобождает программу от бесчисленных имен меток, которые в классическом ассемблере являются неизбежным балластом. Вся логическая структура программы становится наглядной.

Графические технологии раскрывают новые возможности для программистов. Визуальность логической структуры уменьшает вероятность ошибок и сокращает сроки разработки.

Для настройки периферийных устройств (таймеры, UART, ADC, SPI и т.д.) предусмотрен специальный элемент алгоритма - "настройщик" с раскрывающимся оконным интерфейсом. В нем достаточно выбрать необходимые параметры работы устройства, а набор инструкций обеспечивающих эти параметры, сформирует компилятор (в правой части окна) рис. 2.2

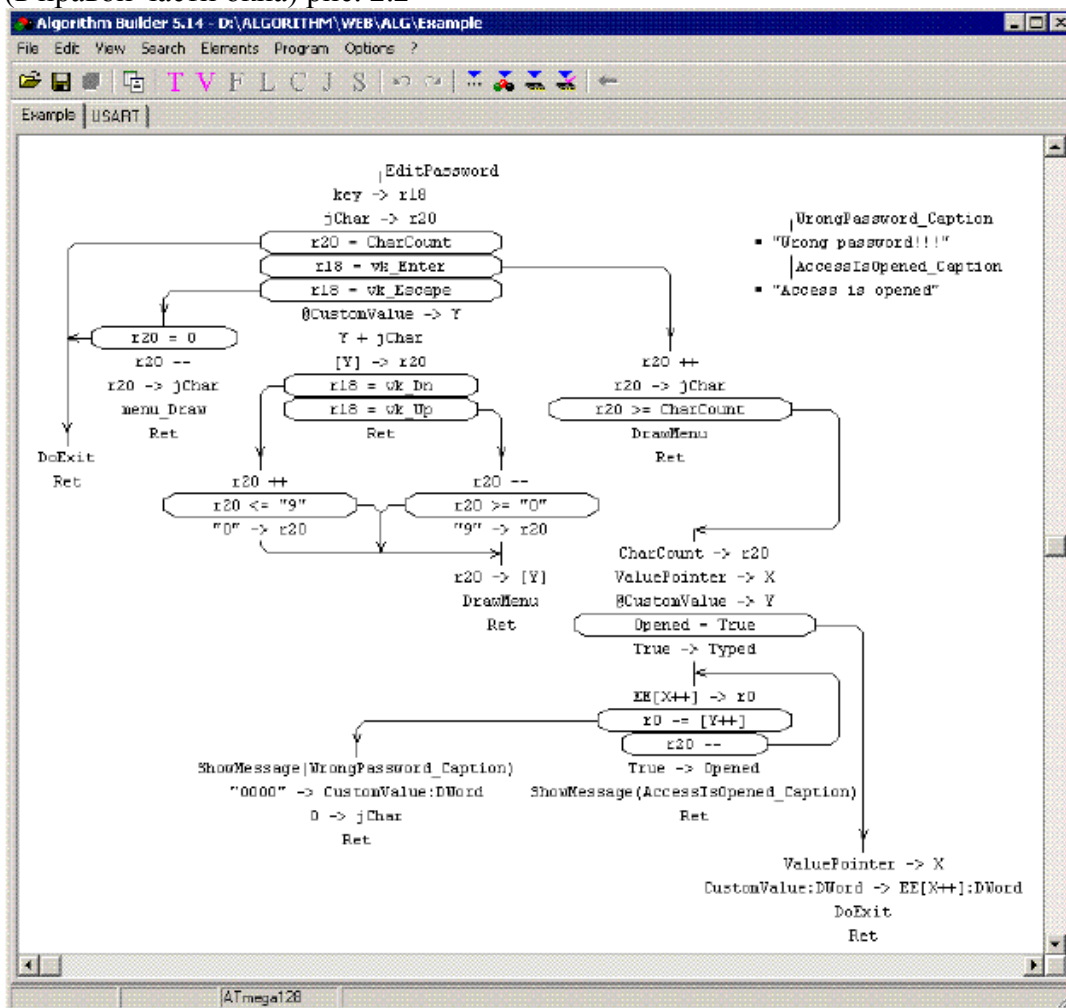


Рис. 2.1. Учебная среда программирования микроконтроллеров.

Поддерживается автоматическая перекодировка строк ANSI-кодов Windows в коды русифицированного буквенно-цифрового ЖКИ.

Среда объединяет в себе графический редактор, компилятор алгоритма, симулятор микроконтроллера, внутрисхемный программатор.

При использовании внутрисхемного программатора микроконтроллер подключается к СОМ порту компьютера через несложный адаптер (три диода и несколько резисторов). Программатор ведет подсчет числа перепрограммирований кристалла, сохраняя счетчик непосредственно в кристалле.

Algorithm Builder обеспечивает мониторинговую отладку на кристалле (On Chip debug) которая позволяет наблюдать содержимое реального кристалла в заданной точке останова. При этом, для связи микроконтроллера с компьютером используется только один вывод,

причем по выбору пользователя. Мониторная отладка может быть применена к любому типу кристалла, имеющего SRAM.

Среда предназначена для работы в ОС Windows 95-Windows 8.1 (32/64 bit).

### **Оборудование и материалы.**

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже

Комплект лабораторных модулей микропроцессорная техника РТМТЛ. Принципиальная схема лабораторного стенда приведена ниже лабораторных работ по исследованию программирования микроконтроллеров на различных языках программирования выполняется на учебном макетном лабораторном комплексе РТМТЛ.

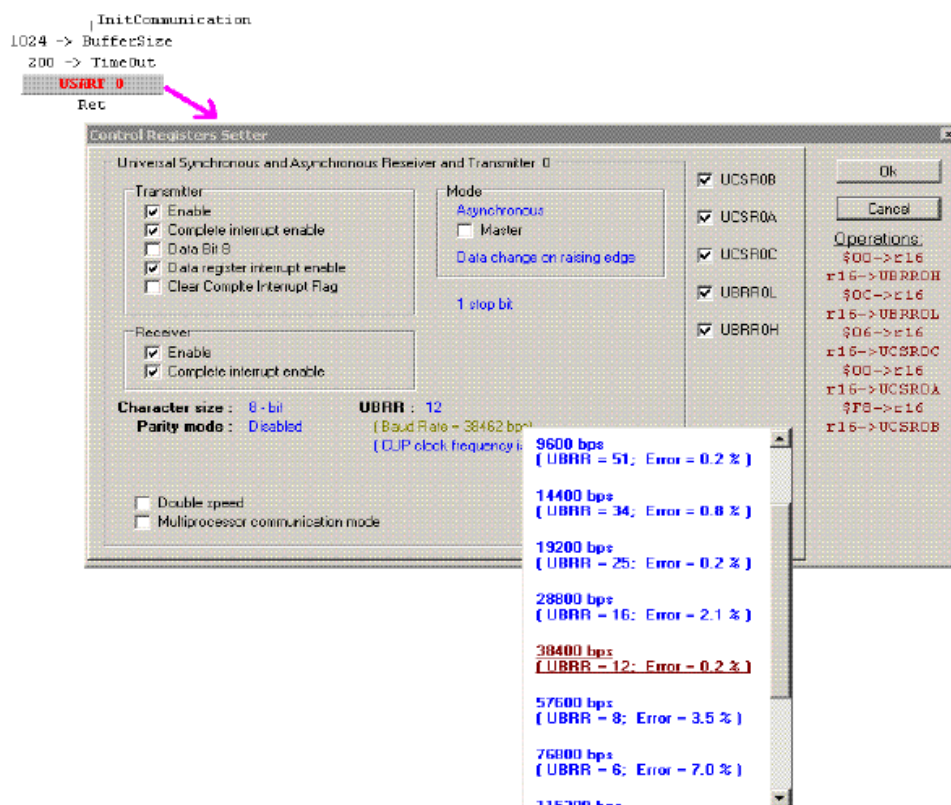


Рис. 2.2. Настройка периферийных устройств.

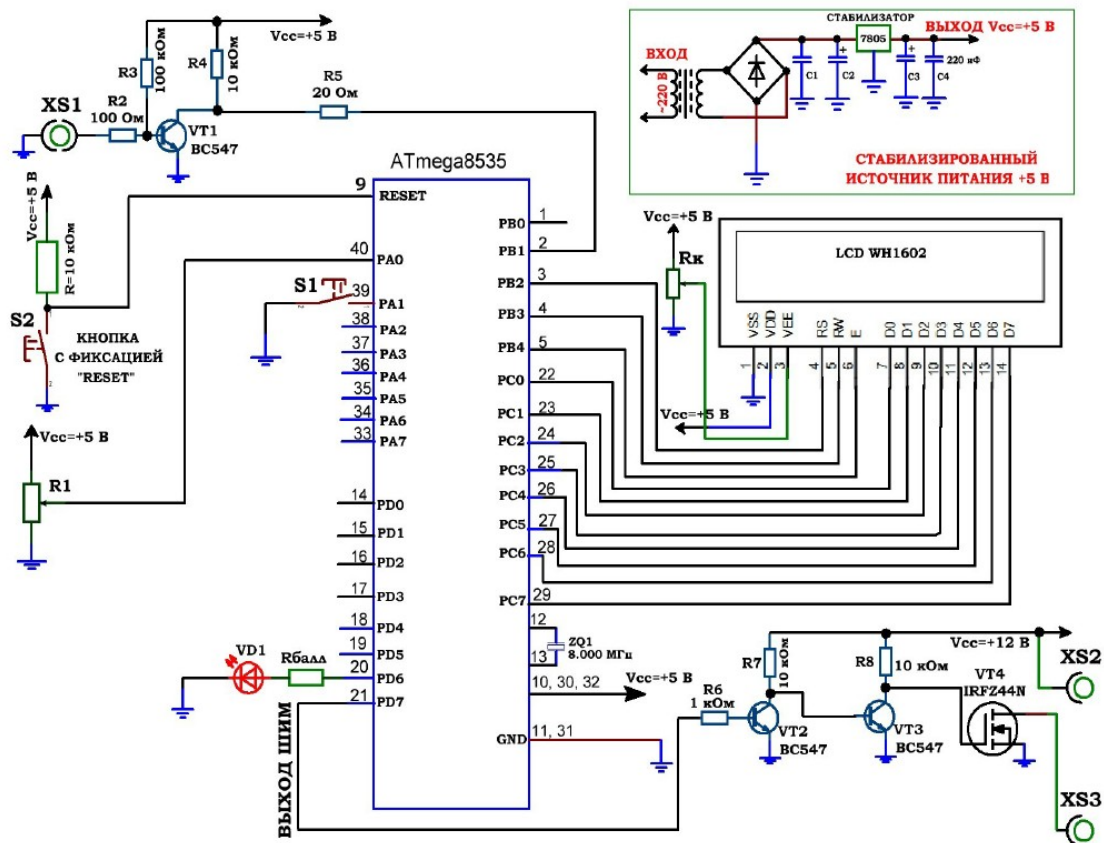


Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.

### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

Комплекс лабораторных работ по исследованию микроконтроллеров AVR выполняется на учебном макетном лабораторном комплексе РТМТЛ-1.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микропроцессор (Atmega8535); резистивно-диодный последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; 1 светодиод, подключенный к выводу 20 рис. 3.1; 1 кнопка с фиксацией нормально разомкнутая (кнопка нажата — замкнута; отжата — разомкнута), подключенная к выводу 39; 1 потенциометр (переменный резистор), подключенный средней точкой к выводу 40 микроконтроллера; выходы «НАГРУЗКА» XS2 — XS3; вход XS1 (для примера работы частотомера); подстроечный резистор Rк — контрастность, подключенный средней точкой к выводу VEE LCD индикатора; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к COM – порту ПЭВМ **ТОЛЬКО** проводом типа COM 9m/9f («мама» - «папа»).

Принципиальная электрическая схема прибора приведена на рис. 3.1.

Демонстрационные примеры готовых программ, на основе которых могут быть построены задания по программированию микроконтроллера, приведены в папке Examples\_RTMTL-1.

**Задание 1.** Работа со светодиодом, таймером, ЖК индикатором и входом PB1 микроконтроллера. Построение программы-частотомера. В данном примере на вывод 2 PB1 микроконтроллера (вход XS1) подаётся сигнал с внешнего генератора (в качестве генератора можно использовать программу «LabVisual ЗВУКОВОЙ ГЕНЕРАТОР ПК»). Сигнал предварительно усилен однострансistorным резистивным усилителем, собранном на транзисторе VT1 BC547 и резисторах R2-R5. Микроконтроллер программируется таким образом, чтобы его TIMER1 был настроен на режим синхронизации внешним сигналом, поступающим на порт PB1. Программа 1 раз в секунду выполняет подсчет входных импульсов синхронизации, поступающих на вход PB1 (макс. Значение 65535). Предусмотрен вывод показаний на LCD индикатор (максимальное значение 9999 Гц). Количество подсчитанных импульсов за 1 сек и есть частота входного сигнала в Гц.

Программа также одновременно работает с подключенным к микроконтроллеру ЖК LCD индикатором (инициализирует его, осуществляет вывод информации).

Одновременно с этим каждую секунду производится инвертирование состояние вывода PD6, к которому подключен светодиод VD1 (1 сек. Светодиод включен, 1 сек. - выключен). При отключенном от клеммы XS1 внешнем генераторе в нижней строке дисплея отображается INPUT DISCONNECT, при подключении внешнего генератора и частоте отличной от 0, в строке выводится сообщение INPUT CONNECT.

**Задание 2.** Работа переменного резистора, кнопки, ЖК индикатора и выхода ШИМ генератора. ШИМ генерируется на выводе PD7 микроконтроллера и подается на трёхкаскадный усилитель мощности, два каскада которого собраны на биполярных транзисторах с общим эмиттером VT2=VT3=BC547, а третий каскад — на полевом транзисторе IRFZ44N с общим истоком. В стоковую цепь полевого транзистора подключается нагрузка (клеммы XS2 – XS3) относительно +12 В питания (колебания напряжения питания может составлять +10 ... +12 В, т. к. данный источник не стабилизированный). В качестве нагрузки может быть использован электродвигатель вентилятора либо лампа накаливания 12 В, 5 Вт. **Обратить внимание, что вентилятор плюсовым красным выводом должен подключаться строго к клемме XS2 (соблюдать полярность).**

При запуске программы длительность импульсов плавно нарастает с определенной скоростью, в нижней строке ЖК индикатора при этом выводится SPEED=X1. Кнопка с фиксацией, подключенная к выводу PA1 в этом режиме должна быть отжата, т. к. при нажатии кнопки скорость нарастания длительности импульсов увеличивается в 4 раза и в нижней строке ЖК индикатора отображается SPEED=X4. В верхней строке ЖК индикатора выводится величина Ust=..., которая ограничивает нижний порог ШИМ и задается потенциометром R1, средней точкой подключенному к выводу 40 (PA0) микроконтроллера от 0 до 127 уровней. Отметим, что генерируемый ШИМ имеет всего 255 градаций (уровней), и, например, при Ust=127 будет меняться от 127 до 255 единиц. При достижении максимального значения, длительность импульсов ШИМ плавно убавляется.

### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

### **Контрольные вопросы**

1. Какой уровень сигнала у цифровых выходов МК?
2. Какие типы команд имеет микроконтроллер AVR семейства Mega?
3. Какие типы команд изменяют флаги микроконтроллера?
4. Какие виды адресации использует микроконтроллер AVR?
5. Зачем нужны команды передачи управления?
6. Как организована архитектура ядра микроконтроллеров AVR семейства Mega?

**Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
3. Грамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярные цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опачий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа 7

### Разработка программ работы с аналоговыми сигналами микроконтроллера AVR с устройствами импульсной модуляции микроконтроллера AVR

#### Цель работы

Изучение и разработка программ работы с аналоговыми сигналами микроконтроллера AVR с устройствами импульсной модуляции микроконтроллера AVR.

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контролеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

Модуль 10-битного АЦП последовательного приближения входит в состав практически всех моделей семейства, за исключением ATmega8515x и ATmega162x. Основные параметры этого АЦП следующие:

- абсолютная погрешность:  $\pm 2 \text{ LSB}$ );
- интегральная нелинейность:  $\pm 0.5 \text{ LSB}$ ;
- быстродействие: до 15 тыс. выборков/с.

На входе модуля АЦП имеется 8-канальный (в моделях ATmega640x/1280x/2560x — 16-канальный) аналоговый мультиплексор, предоставляющий в распоряжение пользователя 8 АБ) каналов с несимметричными входами. Кроме того, в моделях ATmega8x, выпускаемых в корпусе DIP, доступно только 6 каналов из восьми.

В большинстве моделей входы АЦП могут объединяться попарно для формирования различного числа каналов с дифференциальным входом. При этом в

некоторых каналах имеется возможность 10- и 200-кратного предварительного усиления входного сигнала. При коэффициентах усиления 1x и 10x действительная разрешающая способность АЦП по этим каналам составляет 8 бит, а при коэффициенте усиления 200x — 7 бит.

В качестве источника опорного напряжения для АЦП может использоваться как напряжение питания микроконтроллера, так и внутренний либо внешний источник опорного напряжения. Модуль АЦП может работать в двух режимах:

- режим одиночного преобразования, когда запуск каждого преобразования инициируется пользователем;
- режим непрерывного преобразования, когда запуск преобразований выполняется непрерывно через определенные интервалы времени.

## 9.2. Функционирование модуля АЦП

Обобщенная структурная схема модуля АЦП приведена на Рис. 9.1. В моделях ATmega8x и ATmega48x/88x/168x элементы, выделенные на рисунке серым цветом, и связанные с ними сигналы отсутствуют, а неинвертирующий вход компаратора выборки-хранения подключен непосредственно к выходу мультиплексора (показано пунктиром).

Регистры, используемые для управления модулем АЦП в различных моделях, приведены в Табл. 9.1.

Таблица 9.1. Регистры управления модулем АЦП

Регистр	Адрес	ATmega8/32/35x	ATmega8x	ATmega 16x/32x	ATmega64x	ATmega 128x	ATmega48x/88x/168x	ATmega 164x/324x/644x	ATmega 165x	ATmega 325x/3250x, ATmega645x/6450x	ATmega640x, ATmega 1280x/1281x, ATmega2560x/2561x	Описание
ADCSR	\$06 (\$26)		•									Регистр управления и состояния
ADCSRA	\$06 (\$26)	•		•	•	•						Регистр А управления и состояния
	(\$7A)						•	•	•	•	•	
ADCSRB	(\$8E)				•							Регистр В управления и состояния
	(\$7B)						•	•	•	•	•	
ADMUX	\$07 (\$27)	•	•	•	•	•						Регистр управления мультиплексором
	(\$7C)						•	•	•	•	•	
SFIOR	\$30 (\$50)	•	•	•								Регистр специальных функций
	\$20 (\$40)					•						



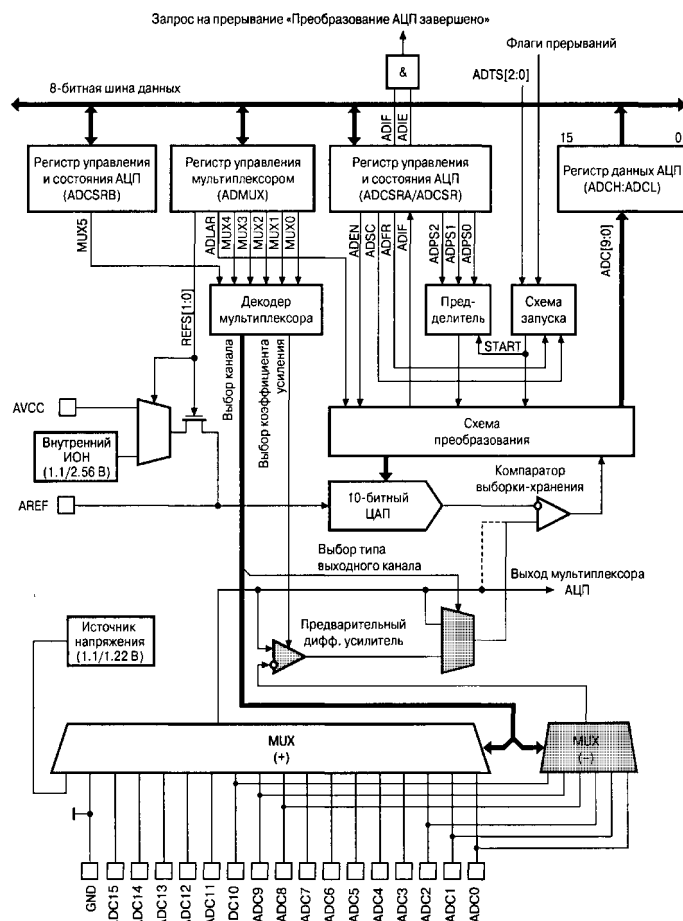


Рис. 9.1. Структурная схема модуля АЦП

Формат регистров ADCSRA (ADCSR) и ADMUX приведен на Рис. 9.2 и Рис. 9.3, а краткое описание функций их битов приведено в Табл. 9.2 и Табл. 9.3 соответственно.

	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ATmega8x ATmega128x
Чтение(R)/Запись(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	

	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	Остальные модели
Чтение(R)/Запись(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	

ATmega8x — ADCSR  
Остальные модели — ADCSRA

Рис. 9.2. Формат регистра ADCSRA (ADCSR)

Таблица 9.2. Биты регистра ADCSRA (ADCSR<sup>1)</sup>)

Бит	Название	Описание
7	ADEN	Разрешение АЦП (1 — включено, 0 — выключено)
6	ADSC	Запуск преобразования (1 — начать преобразование)
5	ADATE (ADFR <sup>2)</sup> )	Выбор режима работы АЦП
4	ADIF	Флаг прерывания от компаратора
3	ADIE	Разрешение прерывания от компаратора
2...0	ADPS2:ADPS0	Выбор частоты преобразования

<sup>1)</sup> В модели ATmega8x.  
<sup>2)</sup> В моделях ATmega8x и ATmega128x.

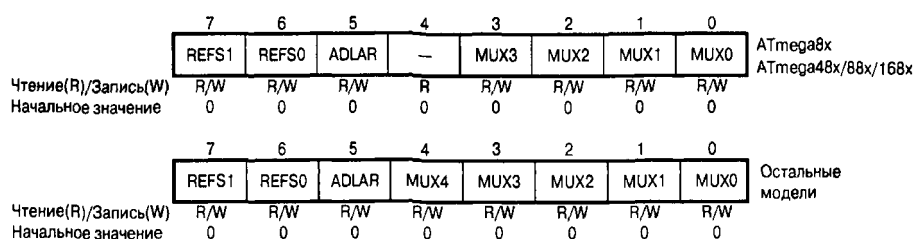


Рис. 9.3. Формат регистра ADMUX

Таблица 9.3. Биты регистра ADMUX

Бит	Название	Описание	Модель
7, 6	REFS1:REFS0	Выбор источника опорного напряжения	Все модели
5	ADLAR	Выравнивание результата преобразования	Все модели
4	—	Зарезервировано	ATmega8x, ATmega48x/88x/168x
	MUX4	Выбор входного канала	Остальные
3...0	MUX3...MUX0	Выбор входного канала	Все модели

Формат регистров ADCSRB и SFIOR приведен на Рис. 9.4 и Рис. 9.5 соответственно (неиспользуемые в данном случае биты регистра SFIOR указаны на рисунке как «X»).

Для разрешения работы АЦП необходимо записать лог. 1 в бит ADEN регистра ADCSR, а для выключения — соответственно лог. 0. Если АЦП будет выключен во время цикла преобразования, то преобразование завершено не будет (в регистре данных АЦП останется результат предыдущего преобразования).

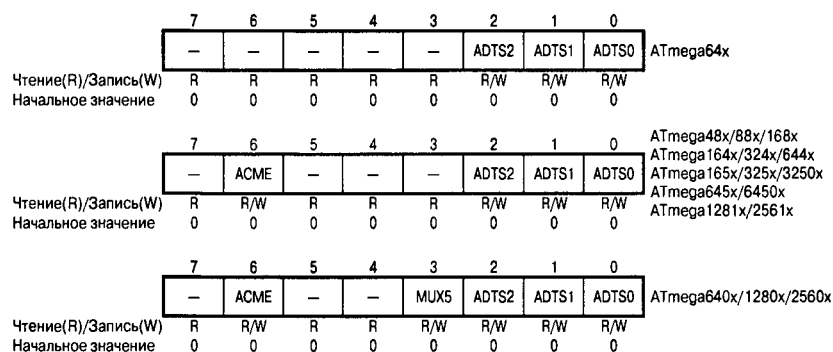


Рис. 9.4. Формат регистра ADCSRB

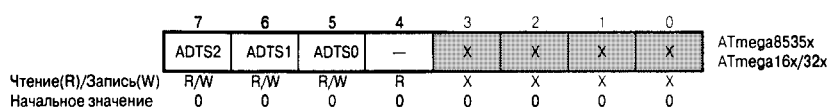


Рис. 9.5. Регистр SFIOR

В моделях ATmega8x и ATmega128x режим работы АЦП определяется состоянием бита ADFR. Если он установлен в 1, АЦП работает в режиме непрерывного преобразования. В этом режиме запуск каждого следующего преобразования осуществляется автоматически после окончания текущего. Если же бит ADFR сброшен в 0, АЦП работает в режиме одиночного преобразования и запуск каждого преобразования осуществляется по команде пользователя.

Во всех моделях, кроме ATmega8x и ATmega128x, запуск АЦП возможен не только по команде пользователя, но и по прерыванию от некоторых периферийных устройств, имеющих в составе микроконтроллера. Для выбора режима работы в этих моделях используется бит ADATE регистра ADCSRA и биты ADTS2:0 регистра SFIOR или ADCSRB (см. Табл. 9.1 и Рис. 9.4, 9.5).

В моделях ATmega8x и ATmega128x режим работы АЦП определяется состоянием бита ADFR. Если он установлен в 1, АЦП работает в режиме непрерывного преобразования. В этом режиме запуск каждого следующего преобразования осуществляется автоматически после окончания текущего.

Если же бит ADFR сброшен в 0, АЦП работает в режиме одиночного преобразования и запуск каждого преобразования осуществляется по команде пользователя.

**Таблица 9.4. Источник сигнала для запуска преобразования**

ADTS2	ADTS1	ADTS0	Источник стартового сигнала
0	0	0	Режим непрерывного преобразования
0	0	1	Прерывание от аналогового компаратора
0	1	0	Внешнее прерывание INT0
0	1	1	Прерывание по событию «Совпадение» («Совпадение А») таймера/счетчика T0
1	0	0	Прерывание по переполнению таймера/счетчика T0
1	0	1	Прерывание по событию «Совпадение В» таймера/счетчика T1
1	1	0	Прерывание по переполнению таймера/счетчика T1
1	1	1	Прерывание по событию «Захват» таймера/счетчика T1

Запуск каждого преобразования в режиме одиночного преобразования, а также запуск первого преобразования в режиме непрерывного преобразования осуществляется установкой в 1 бита ADSC регистра ADCSRA (ADCSR). Запуск преобразования по прерыванию осуществляется при установке в 1 флага выбранного прерывания. Бит ADSC регистра ADCSRA при этом аппаратно устанавливается в 1. Запуск преобразования в этих режимах также может быть осуществлен установкой бита ADSC регистра ADCSRA в 1.

В режимах одиночного и непрерывного преобразований цикл преобразования начинается по первому нарастающему фронту тактового сигнала после установки бита ADSC. Если используется запуск по прерыванию, то цикл преобразования начинается по первому нарастающему фронту тактового сигнала после установки флага выбранного прерывания. Причем в момент установки этого флага осуществляется сброс предделителя модуля АЦП. Тем самым обеспечивается фиксированная задержка между генерацией запроса на прерывание и началом цикла преобразования. Обратите внимание, что преобразование запускается при установке соответствующего флага, т. е. даже если само прерывание запрещено.

Длительность цикла составляет 13 тактов при использовании несимметричного входа и 13 либо 14 тактов при использовании дифференциального входа (разные значения связаны с работой схемы синхронизации); выборка и запоминание входного сигнала осуществляется в течение первых 1.5 и 2.5 тактов соответственно. Через 13 A4) тактов преобразование завершается, бит ADSC аппаратно сбрасывается в 0 (в режиме одиночного преобразования), и результат преобразования сохраняется в регистре данных АЦП. Одновременно устанавливается флаг прерывания ADIF регистра ADCSR и генерируется запрос на прерывание. Как и флаги остальных прерываний, флаг ADIF сбрасывается аппаратно при запуске подпрограммы обработки прерывания от АЦП или программно, записью в него лог. 1. Разрешение прерывания осуществляется установкой в 1 бита ADIE регистра ADCSR при установленном флаге I регистра SREG.

Если АЦП работает в режиме непрерывного преобразования, то новый цикл начнется сразу же после записи результата. В режиме одиночного преобразования новое преобразование может быть запущено сразу же после сброса бита ADSC (до сохранения результата текущего преобразования). Однако реально цикл преобразования начнется не

ранее чем через один такт после окончания текущего преобразования. Временные диаграммы, иллюстрирующие сказанное, приведены на Рис. 9.6.

При первом запуске после включения АЦП для выполнения преобразования потребуется 25 тактов, т. е. на 12 тактов больше, чем обычно. В течение этих 12 тактов выполняется «холостое» преобразование, во время которого производится инициализация АЦП (Рис. 9.7).

Отдельно следует сказать об использовании режима запуска по прерыванию совместно с дифференциальными каналами. В этом случае АЦП необходимо выключать между преобразованиями, чтобы избежать некорректных измерений, связанных с неопределенностью момента сброса предделителя АЦП. В результате выключения и включения АЦП между преобразованиями будут выполняться только «длинные» преобразования, результаты которых всегда будут корректными.

Для формирования тактовой частоты модуля АЦП в нем имеется отдельный предделитель. Коэффициент деления предделителя и соответственно длительность преобразования определяются состоянием битов ADPS2...ADPS0 регистра ADCSRA/ADCSR (см. Табл. 9.5).

Наибольшая точность преобразования достигается, если тактовая частота модуля АЦП находится в диапазоне 50...200 кГц. Соответственно, коэффициент деления предделителя рекомендуется выбирать таким, чтобы тактовая частота модуля АЦП находилась в указанном диапазоне. Если же точности преобразования меньше 10 битов достаточно, можно использовать более высокую частоту, увеличивая тем самым частоту выборки.

В зависимости от модели выводы микроконтроллера, подключенные к входу АЦП, определяются состоянием битов MUX3:0, MUX4:0 либо MUX5:0 (MUX5 - в ADCSRB, MUX4...MUX0 - в ADMUX). Для каналов с дифференциальным входом указанные биты определяют также коэффициент предварительного усиления входного сигнала. Соответствие установок этих битов используемым входам АЦП приведено в Табл. 9.6...9.8.

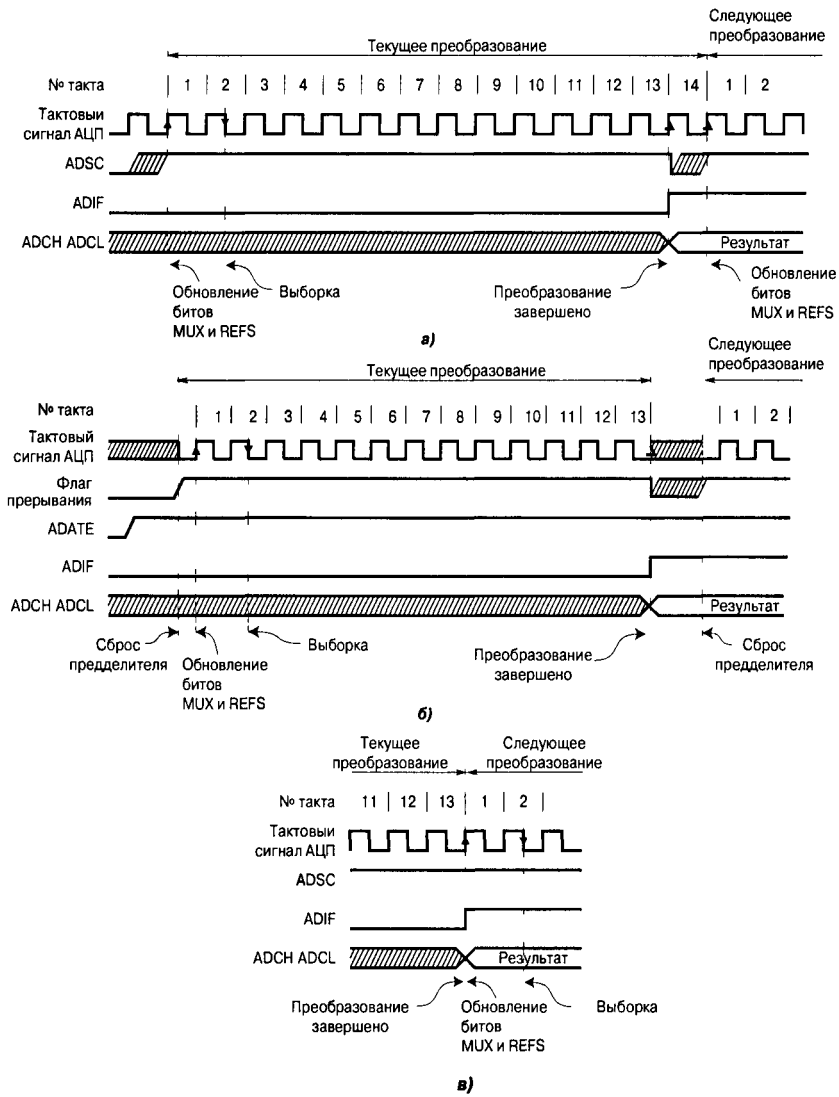


Рис. 9.6. Временные диаграммы работы АЦП в режиме одиночного преобразования (а), в режиме запуска по прерыванию (б) и в режиме непрерывного преобразования (в)

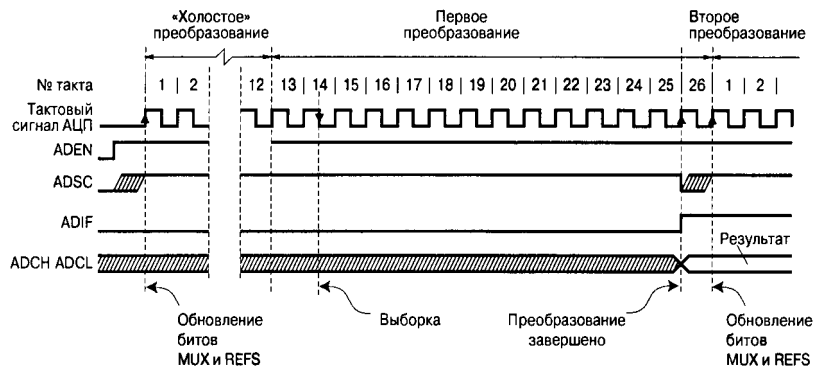


Рис. 9.7. Временные диаграммы работы АЦП при первом преобразовании (режим одиночного преобразования)

**Таблица 9.5. Задание коэффициента деления предделителя АЦП**

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

**Таблица 9.6. Управление входным мультиплексором в моделях ATmega8x и ATmega48x/88x/168x**

MUX3...MUX0	Несимметричный вход	MUX3...MUX0	Несимметричный вход
0000	ADC0	0110	ADC6 <sup>1)</sup>
0001	ADC1	0111	ADC7 <sup>1)</sup>
0010	ADC2	1000...1101	Зарезервировано
0011	ADC3	1110	1.22 В (1.1 В <sup>2)</sup> )
0100	ADC4	1111	0 В (GND)
0101	ADC5		

<sup>1)</sup> Только в корпусах TQFP-32 и MLF-32.  
<sup>2)</sup> В моделях ATmega48x/88x/168x.

**Таблица 9.7. Управление входным мультиплексором в моделях ATmega8535x, ATmega16x/32x/64x/128x, ATmega164x/324x/644x, ATmega165x/325x/3250x/645x/6450x и ATmega1281x/2561x**

MUX4...MUX0	Несимметричный вход	Дифференциальный вход		Предварительное усиление
		положительный	отрицательный	
00000	ADC0			
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000		ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x
11101		ADC5	ADC2	1x
11110	1.22 В (1.1 В <sup>1)</sup> )			
11111	0 В (GND)			

<sup>1)</sup> В моделях ATmega165x/325x/3250x/645x/6450x и ATmega1281x/2561x

**Таблица 9.8. Управление входным мультиплексором в моделях ATmega640х/1280х/2560х**

MUX5...MUX0	Несимметричный вход	Дифференциальный вход		Предварительное усиление
		положительный	отрицательный	
000000	ADC0			
000001	ADC1			
000010	ADC2			
000011	ADC3			
000100	ADC4			
000101	ADC5			
000110	ADC6			
000111	ADC7			
001000		ADC0	ADC0	10x
001001		ADC1	ADC0	10x
001010		ADC0	ADC0	200x
001011		ADC1	ADC0	200x
001100		ADC2	ADC2	10x
001101		ADC3	ADC2	10x
001110		ADC2	ADC2	200x
001111		ADC3	ADC2	200x
010000		ADC0	ADC1	1x
010001		ADC1	ADC1	1x
010010		ADC2	ADC1	1x
010011		ADC3	ADC1	1x
010100		ADC4	ADC1	1x
010101		ADC5	ADC1	1x
010110		ADC6	ADC1	1x
010111		ADC7	ADC1	1x
011000		ADC0	ADC2	1x
011001		ADC1	ADC2	1x
011010		ADC2	ADC2	1x
011011		ADC3	ADC2	1x
011100		ADC4	ADC2	1x
011101		ADC5	ADC2	1x
011110	1.1 В			
011111	0 В (GND)			

Следует отметить, что предварительный усилитель, используемый каналами с дифференциальным входом, имеет встроенную схему коррекции смещения. Оставшаяся после коррекции величина смещения может быть учтена программным способом. Для этого следует оба входа дифференциального усилителя подключить к одному и тому же выводу микроконтроллера (Табл. 9.7 и Табл. 9.8), а затем вычитать полученное значение из результата последующих преобразований. Таким образом, ошибка смещения может быть снижена а до величины, меньшей 1 LSB.

Состояние битов MUX5...MUX0 можно изменить в любой момент, однако если это будет сделано во время цикла преобразования, то смена канала произойдет только после завершения преобразования. Благодаря этому в режиме непрерывного преобразования можно легко осуществлять последовательное преобразование сигналов нескольких каналов.

Отдельно следует сказать о каналах с дифференциальным входом. После смены таких каналов первое измерение следует производить не ранее чем через 125 мкс после выбора канала. Указанное время требуется для установления значения коэффициента усиления предусилителя. Соответственно, значения, измеренные до истечения этого

срока, не могут считаться достоверными. Кроме того, при переключении на канал с изменяемым коэффициентом усиления результат первого преобразования может иметь пониженную точность из-за времени установления автоматической схемы коррекции смещения. Поэтому будет лучше считать результат первого преобразования некорректным.

В новых микроконтроллерах, а именно в ATmega48x/88x/168x, ATmega164x/324x/644x, ATmega165x, ATmega325x/3250x/645x/6450x и ATmega640x/1280x/1281x/2560x/2561x, имеется возможность отключения входных цифровых буферов на выводах ADC0...ADC15 в случае, если соответствующие выводы используются только для считывания аналоговых сигналов. При отключенных цифровых буферах уменьшается общий ток потребления микроконтроллера, а соответствующие биты регистров PINx всегда читаются как 0.

Отключение цифровых буферов на входах ADC0...ADC7 осуществляется записью лог. 1 соответственно в биты ADC0D...ADC7D регистра DIDR0, расположенного по адресу (\$7E). А отключение буферов на входах ADC8...ADC15 (модели ATmega640x/1280x/2560x) осуществляется записью лог. 1 в биты ADC8D...ADC15D регистра DIDR2, расположенного по адресу (\$7D). Формат этих регистров приведен на Рис. 9.8.

Как уже было отмечено, в модуле АЦП могут использоваться различные источники опорного напряжения (ИОН). Выбор конкретного источника опорного напряжения осуществляется с помощью битов REFS1:REFS0 регистра ADMUX (см. Табл. 9.9).

### **Оборудование и материалы.**

Комплекс лабораторных работ по исследованию микроконтроллеров AVR выполняется на учебном макетном лабораторном комплексе РТМТЛ-2.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микропроцессор (Atmega8535); резистивно-диодный последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; набор из 8 светодиодов, подключенных к выводам 1, 2, 3, 4, 18, 19, 20, 21 рис. 3.3; набор из 8 кнопок без фиксации нормально разомкнутых (кнопка нажата — замкнуто; отжата — разомкнуто); 1 потенциометр (переменный резистор), подключенный средней точкой к выводу 40 АЦП микроконтроллера; два семисегментных индикатора; выходы «НАГРУЗКА» XS — XS2; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для связи учебного стенда с ПК (передача данных) используется микросхема типа MAX232 (рис. 3.1) и разъем «ДАННЫЕ», расположенный на задней крышке прибора. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к COM – порту ПЭВМ

**ТОЛЬКО** проводом типа COM 9m/9f («мама» - «папа»), разъем «ДАННЫЕ» может быть подключен непосредственно как к COM порту ПК проводом COM 9m/9f («мама» - «папа»), так и к USB порту ПК через переходник USB-COM (RS232), собранный на микросхеме Prologic PL2303 (либо аналог.) рис. 3.2.



### Схема связи микроконтроллера AVR с платой mini-ITX через COM порт (RS-232)

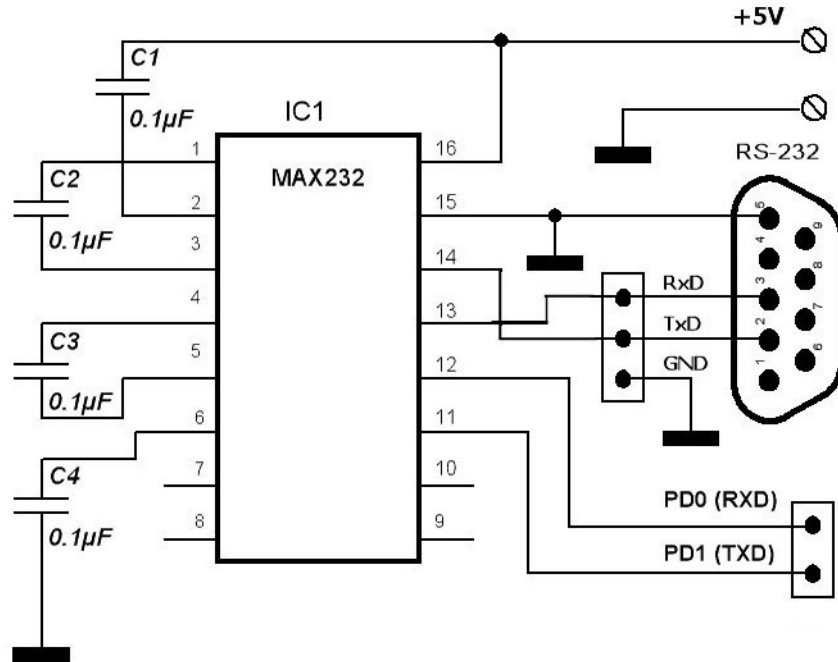


Рис. 3.1. Связь микроконтроллера с COM портом ПК через микросхему MAX232.  
**МК AVR Atmega8535 РАЗЪЁМ «ДАННЫЕ»**

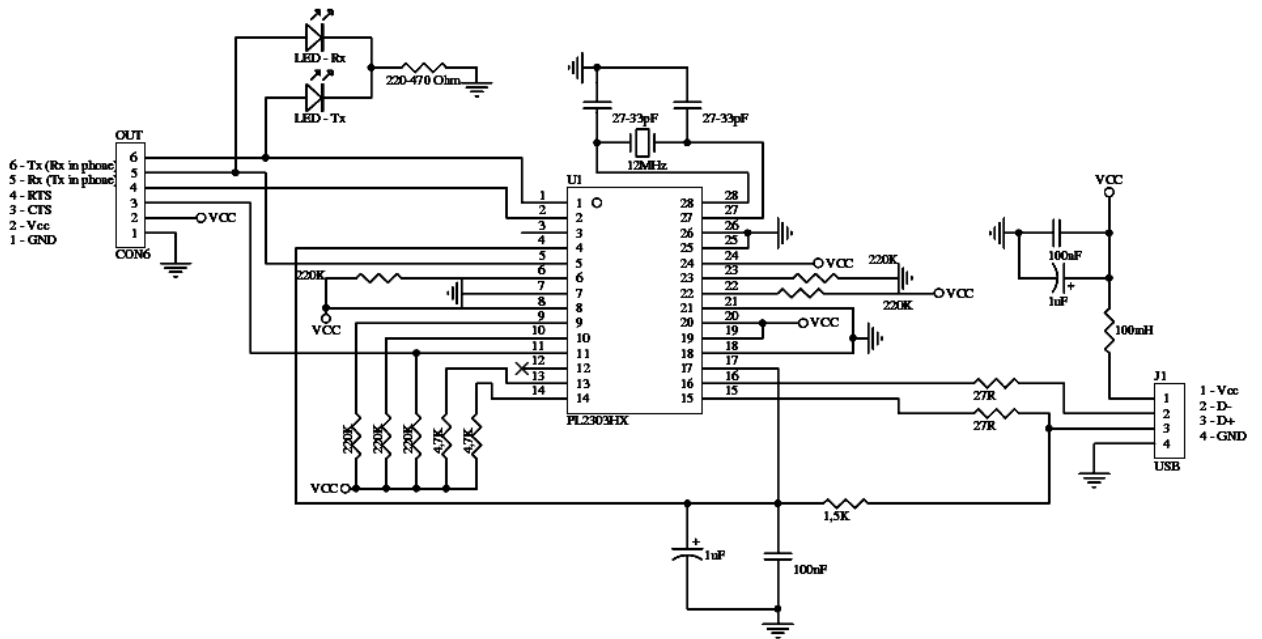


Рис. 3.2. Переходник USB-COM (RS232), собранный на микросхеме Prologic PL2303 (микросхема внутри кабеля USB – COM).

Принципиальная электрическая схема прибора приведена на рис. 3.3.

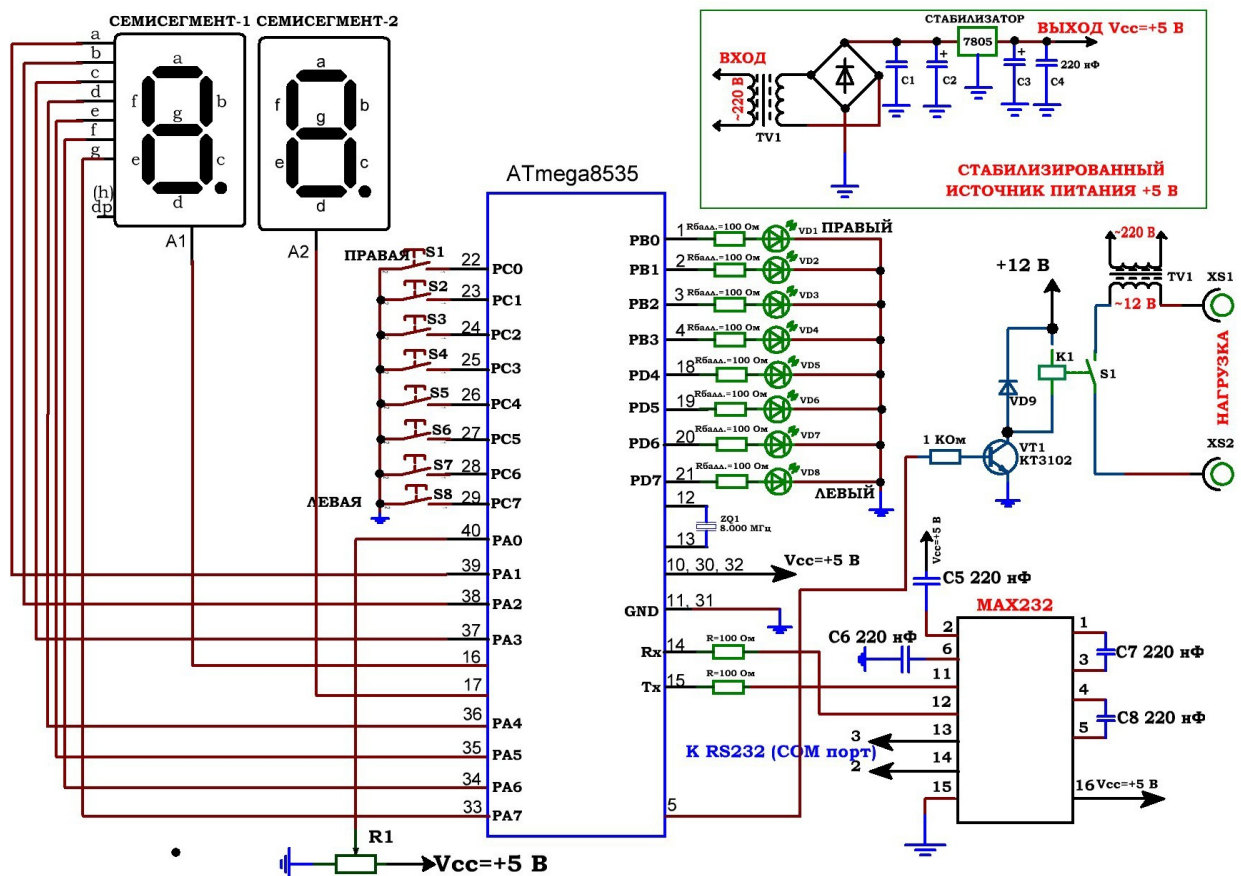


Рис. 3.3 Принципиальная электрическая схема прибора

Демонстрационные примеры готовых программ, на основе которых могут быть построены задания по программированию микроконтроллера, приведены в папке Examples\_RTMTL-2.

- 1) Пример 1. По нажатию кнопки S1, подключенной к 22 выводу МК начинает светиться светодиод VD1, подключенный к выводу 1 микроконтроллера. По отжатию кнопки, светодиод перестаёт светиться (индикация работы кнопки);
- 2) Пример 2. При нажатии кнопки S1, подключенной к 22 выводу МК начинает мигать светодиод VD1, подключенный к выводу 1 микроконтроллера;
- 3) Пример 3. Мигание светодиода VD1, подключенного к выводу 1.
- 4) Пример 4. Светодиоды VD1 – VD8 мигают по очереди (по порядку подключения к выводам 1, 2, 3, 4, 18, 19, 20, 21 МК);
- 5) Пример 5. Светодиоды VD1 – VD8 мигают по очереди в той же последовательности с одновременной передачей по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на COM порт) номера включенного светодиода в ASCII коде (натуральное число);
- 6) Пример 6. Изучение работы АЦП микроконтроллера. Напряжение считывается с переменного резистора R1, подключённого средней точкой к 40 выводу МК (вход АЦП) в код АЦП от 0 до 63 единиц и с интервалом ~ 1 сек. передаётся по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на COM порт);
- 7) Пример 7. Приёмник RS232 с ПК. Передавая ASCII код 1 ... 8 (натуральные числа) с ПК посредством программы «ТЕРМИНАЛ» можно управлять включением светодиодов;
- 8) Пример 8. Работа с семисегментными индикаторами, RS232 и реле. Реализация таймера обратного отсчета. Текущие показания таймера выводятся на семисегментные индикаторы с общими катодами (a, b, c, d, e, f, g) и анодами A1 и A2, и одновременно передаётся по USART через микросхему MAX232 на разъём «ДАННЫЕ» (выход на

COM порт). При достижении конца отсчета срабатывает реле K1, подключённое через транзисторный усилитель на KT3102 к выводу 5 микроконтроллера и вторичная обмотка трансформатора TV1 (~12 В) соединяется с клеммами XS1 – XS2 (нагрузка). К выходу нагрузки можно подключать лампу накаливания 12 В, 21 Вт либо меньшей мощности.

Запрещается замыкать выходы контрольных точек XS1 – XS2 (нагрузка)!;

9) Пример 9. Реализация вольтметра. Работа с семисегментными индикаторами и реле. Простейшая реализация вольтметра с выводом показаний на семисегментный индикатор. Напряжение считывается с переменного резистора R1, подключённого средней точкой к 40 выводу МК (вход АЦП) и выводится на семисегментный индикатор в показаниях 0-50 (0-5 Вольт, т. к. вывод h(точка dp) не подключен). При определенном пороге (определенном значении напряжения) срабатывает реле K1 и вторичная обмотка трансформатора TV1 (~12 В) соединяется с клеммами XS1 – XS2 (нагрузка). К выходу нагрузки можно подключать лампу накаливания 12 В, 21 Вт либо меньшей мощности. Запрещается замыкать выходы контрольных точек XS1 – XS2 (нагрузка)!

### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

### Ход работы

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.

2. **Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR.**

3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.

4. Загрузить операционную систему согласно стандартным процедурам загрузки.

5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder.

6. Запустить программу Algorithm Builder для работы с учебной установкой для данного эксперимента пользуясь ярлыком на рабочем столе либо другим способом, указанным лаборантом. **Для работы можно воспользоваться комплексной программной оболочкой LabVisual для РТМТЛ-1-5.**

6. Подключить разъём «ПРОГРАММАТОР» учебного прибора к COM – порту ПЭВМ проводом типа COM 9m/9f («мама» - «папа»)

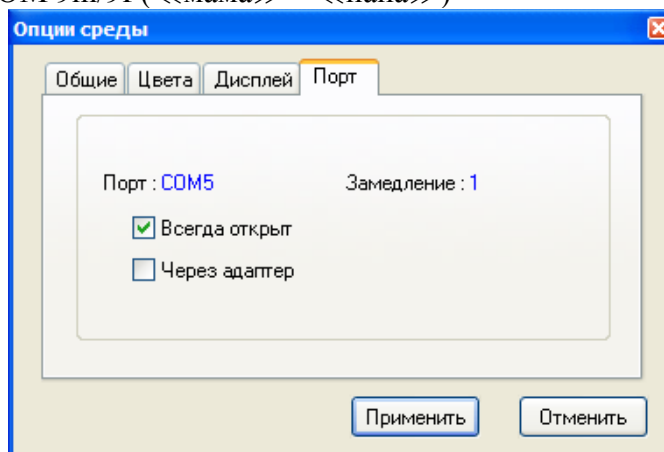


Рис. 4.1

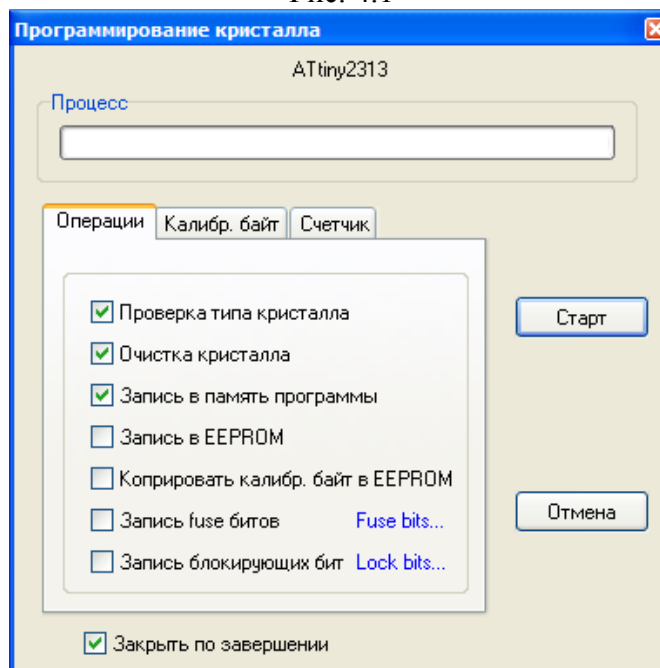


Рис. 4.2

7. Настроить программу Algorithm Builder для работы с данным COM портом ОПЦИИ-ОПЦИИ СРЕДЫ-ПОРТ рис. 4.1.

8. Для подробных инструкций следует обращаться к методическому руководству по среде Algorithm Builder.

9. Загрузить Пример 1 (папка 1) в среду Algorithm Builder (ФАЙЛ-ОТКРЫТЬ-ВЫБРАТЬ ФАЙЛ \*.alp).

10. «Прошить» тестовую программу в кристалл. Для этого нажать кнопку «ЗАПУСК С КРИСТАЛЛОМ», при этом откроется диалог программирования кристалла рис. 4.2.

**Перед прошивкой следует нажать кнопку с фиксацией #RESET# на учебном приборе.**

11. Перед нажатием кнопки «СТАРТ» следует внимательно проверить положения флажков установки программатора рис. 4.2. **Особенно внимательно следует относиться к опциям #ЗАПИСЬ Fuse bit# и #ЗАПИСЬ блокирующих бит#. ФЛАЖКИ ЭТИХ ОПЦИЙ ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ СНЯТЫ, Т. К. НЕ ПРАВИЛЬНО ПРОШИТЫЕ FUSE БИТЫ МОГУТ ПРИВЕСТИ К ДАЛЬНЕЙШЕЙ НЕВОЗМОЖНОСТИ РАБОТЫ С ДАННЫМ МИКРОКОНТРОЛЛЕРОМ!**

12. **Отжать кнопку RESET и проверить работу тестовой программы.**

13. Загрузить примеры 2 — 9 из соответствующих папок и прошить в кристалл каждую из тестовых программ.

14. При работе примеров 5 — 7 следует соединить разъём «ДАННЫЕ» с ПК.

Подключение может быть произведено непосредственно как к COM порту ПК проводом COM 9m/9f («мама» - «папа», тем же, что вы использовали для программирования), так и к USB порту ПК через переходник USBCOM (RS232).

15. При подключении разъёма «ДАННЫЕ» через переходник USB-COM (RS232) в системе в диспетчере устройств появляется виртуальный COM- порт рис. 4.3.

16. Перед дальнейшей работы следует выяснить и запомнить номер виртуального COM-порта, присвоенного системой (в примере рис. 4.3 COM33)

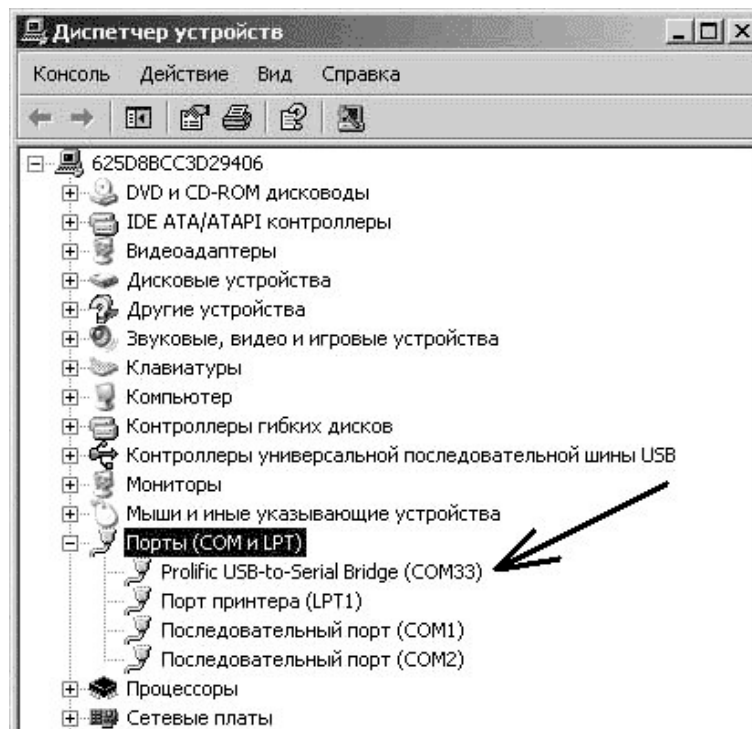


Рис. 4.3

17. Для анализа данных приходящих на СОМ порт ПК, а также для отправки байт управления на учебный стенд следует использовать программу «ТЕРМИНАЛ» для работы с СОМ – портом или аналогичную рис. 4.4.

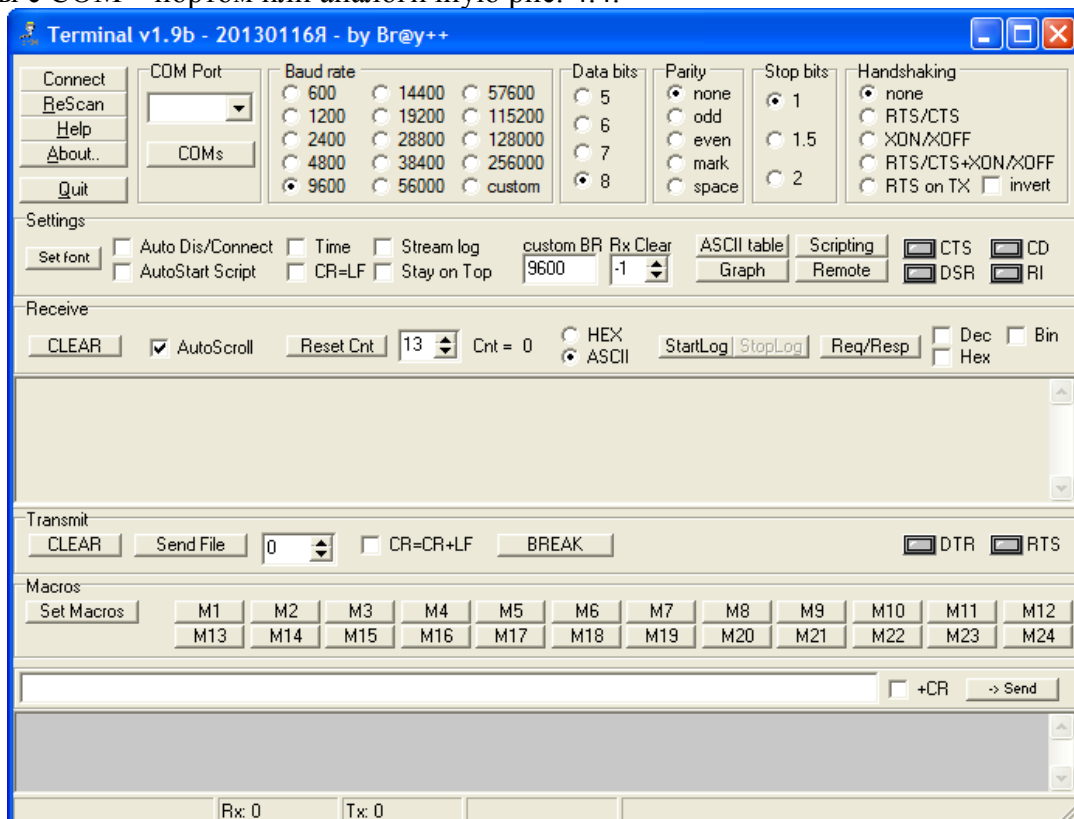


Рис. 4.4

18. В программе следует выбрать СОМ порт (либо виртуальный СОМ порт), к которому подключен разъём «ДАННЫЕ» и нажать кнопку «Connect».

19. Принятые байты данных можно наблюдать в окне терминала, для отправки байтов следует записать отправляемые данные в строку для отправки и нажать кнопку ->Send.

20. Исходя из готовых примеров составить собственные программы в среде Algorithm Builder.

21. На основании Примеров 1 — 4 можно составить программы, изменив, например, действия, осуществляемые кнопкой S1 (включать другой светодиод), либо изменить назначение других кнопок S2 – S8. Также можно изменить скорость мигания, порядок мигания светодиодов и проводить различные операции с микроконтроллером.

22. На основании Примеров 5 — 8 следует изучить работу с COM портом ПК, составляя и отлаживая различные программы для приёма — отправки данных по интерфейсу RS232.

#### **Строка отправляемых данных**

23. На основании примера 6,9 следует изучить работу АЦП микроконтроллера и реализацию динамической индикации на семисегментном индикаторе. На основании примера 8 изучить работу реле (порог срабатывания), изменить этот порог. Вывести результаты измерений по протоколу RS232 на COM порт (разъем данные).

24. На основании примера 8 изучить работу таймера, реализовать обратный отсчет таким образом, чтобы время отсчета (интервал) соответствовало строго 1 секунде (в примере время счета выбрано произвольно). Изучить работу с реле.

25. Для составления программ и подробного изучения работы микроконтроллеров AVR следует обращаться к учебной литературе, полному руководству к микроконтроллерам семейства AVR, а также к документации по среде Algorithm Builder.

26. По окончании работы следует закрыть программу и все открытые подпрограммы, закрыть виртуальную среду VirtualBox (при работе в среде Linux).

27. Выключить компьютер, нажав на кнопку, находящуюся в крайнем нижнем левом углу экрана. Из доступных действий выбрать «ВЫХОД»--> «ВЫКЛЮЧИТЬ КОМПЬЮТЕР».

28. Отключить установку от сети, поставив переключатели «СЕТЬ» на панели установки в положение «ВЫКЛ» и вынуть сетевые вилки из розеток\_\_

#### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

#### **Контрольные вопросы**

1. Какой предельный уровень сигнала у аналоговых входов МК?
2. Какие регистры и каким образом необходимо инициализировать для работы АЦП и ЦАП микроконтроллера AVR семейства Mega?
3. Каким образом происходит запуск преобразования в АЦП и ЦАП микроконтроллера AVR семейства Mega?
4. Какие выводы для АЦП использует микроконтроллер AVR семейства Mega?
5. Какие выводы для ЦАП использует микроконтроллер AVR семейства Mega?
6. Какие сигналы говорят об окончании процесса преобразования аналогового сигнала в цифровой?

#### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
3. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.

4. 4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярные цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа 8

### Разработка программ вывода информации на ЖКИ микроконтроллера AVR и программ для связи микроконтроллера AVR с ПК-ЭВМ

#### Цель работы

Изучение инструментальных средств разработки программного обеспечения МК.  
Разработка программ вывода информации на ЖКИ микроконтроллера AVR и программ для связи микроконтроллера AVR с ПК-ЭВМ.

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контроллеров (ПЛК)
ПК-11	готовностью производить установку и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

По отношению к обыкновенным 7-сегментным индикаторам (рис. 1.1), ЖКИ модули на базе контроллера HD44780 (рис. 1.2) обладают на порядок большими возможностями. Количество строк на экране у разных моделей - 1,2 или 4; число символов в строке:

8,10,16,20,24,30,32 или 40.



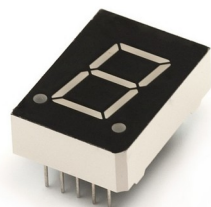


Рис 1.1

Контроллер HD44780 (а также совместимый с ним KS0066)— стандарт де-факто на контроллеры монохромных жидкокристаллических знаковсинтезирующих дисплеев с параллельным 4- или 8-битным интерфейсом.

Индикатор может иметь светодиодную или люминесцентную подсветку практически любого цвета свечения. На рис. 1.2 представлен ЖКИ индикатор производства фирмы WinSTAR, на базе контроллера HD44780. Аналогичный вид имеют и другие ЖК символные индикаторы, серийно выпускаемые промышленностью.



Рис.1.2

Напряжение питания контроллера HD44780 5В (реже 3В). Ток потребления контроллера очень мал(100...200 мкА), чего не скажешь о светодиодной подсветке. В зависимости от производителя, его величина составляет 80...120 мА.

Для работы некоторых типов ЖКИ может потребоваться дополнительный источник напряжения отрицательной полярности. Технология производства модулей подобного рода непрерывно совершенствуется, что, в целом, положительно сказывается на их размерах и электрических характеристиках.

#### **Символьный ЖКИ с контроллером HD44780 (KS0066). Интерфейс.**

Символьный ЖКИ представляется матрицей из точек, разделенной на строки и поля символов (рис. 2.1):

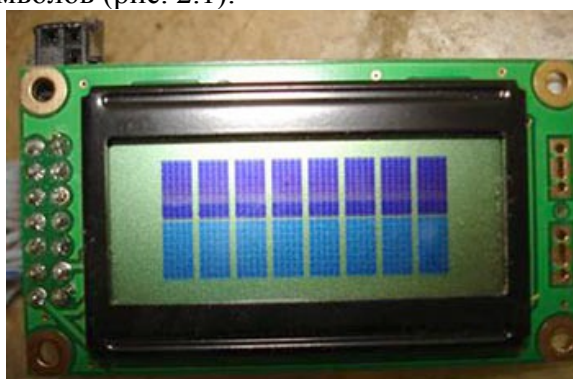


Рис. 2.1

Для управления этой матрицей и вывода собственно символов используется специальный контроллер HD44780.

Контролер ЖКИ оперирует 3-мя блоками памяти:

Для вывода символа контроллер использует память **DDRAM** (Display Data RAM), где хранятся ASCII-коды символов, которые мы хотим видеть на ЖКИ.

Под нее отведено 80 ячеек памяти. Понятно, что на ЖКИ мы увидим лишь часть символов, которые находятся в DDRAM - если наш ЖКИ 1 или 2- строчный и отображает 8 символов в строке, то так:

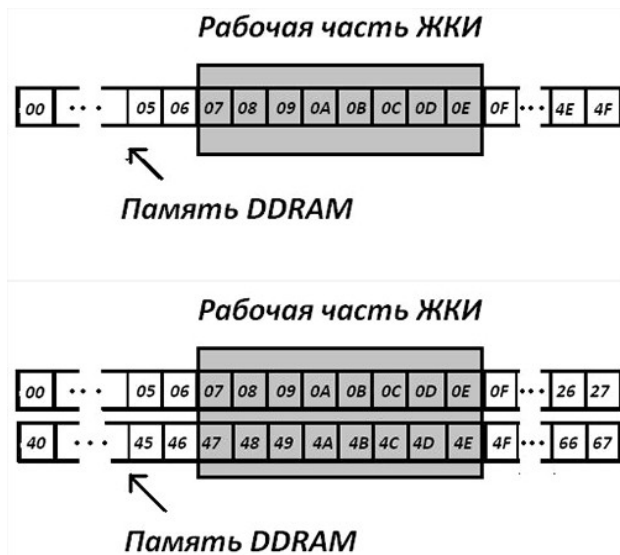


Рис. 2.2

Рабочую область дисплея, как видно, можно смещать по ячейкам DDRAM (получается эффект бегущей строки).

2. Шаблоны самих символов контроллер берет из CGROM (Character Generator ROM) – памяти знакогенератора. Таблицу символов можно посмотреть в спецификации на HD44780.

3. Для хранения пользовательских символов (их шаблонов) предусмотрена память CGRAM (Character Generator RAM).

Также, контроллер в зависимости от некоторых условий распределяет пришедшие в него данные в регистр инструкций или регистр данных.

Контрастность изображения на ЖКИ можно изменять, подключив дополнительно построечный резистор на 10 кОм по схеме рис. 2.3.

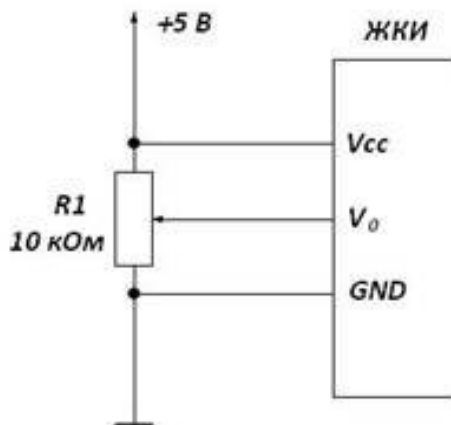


Рис. 2.3

Но следует смотреть в спецификацию на свой контроллер (например у ЖКИ K1sn10294v-0 на чипе KS0066 1-Vcc, а 2-GND). Подача питания подсветки может различаться от модели к модели в зависимости от её типа. Подсветка питается от 5 вольт, токоограничительный резистор (50-100 Ом) обычно не обязателен.

#### **Назначение выводов R/S, R/W, E.**

При переходе E с высокого логического уровня на низкий данные, которые уже «висят» на выводах DB0..DB7, записываются в память контроллера ЖКИ для последующей обработки.

При высоком лог. уровне на R/S(Register Select) контроллер ЖКИ воспринимает этот набор битов как данные(код символа), а при низком – как инструкцию и направляет их в соответствующий регистр.

R/W определяет направление работы выводов DB0..DB7 – если на R/W «0», то мы можем только писать в порт DB, а если R/W = «1», то можем прочитать с него (например

узнать занят контроллер или свободен для приема новых данных). Если мы не будем читать данные из ЖКИ, то можно «посадить» R/W на землю.

#### **Набор команд HD44780**

Для того чтобы начать выводить информацию на ЖКИ, его контроллер надо проинициализировать (сообщить ему об интерфейсе, шрифте, смещениях и т.д.). Контроллер может воспринимать всего 11 команд, приведенных в таблице 1.

Название инструкции	Состояние выводов										Прим.	Время исполнения $f_{\text{раб.}} = 270$ кГц	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear display	0	0	0	0	0	0	0	0	0	1		Очистка Всего ЖКИ установка адреса DDRAM в 0	1.52 мс
Return home	0	0	0	0	0	0	0	0	0	1	X	Установка текущего адреса DDRAM в 0 (курсор – домой) Данные DDRAM не меняются	1.52 мс
Entry mode set	0	0	0	0	0	0	0	1	I/D	S		Установка направления движения курсора (I/D) и смещения дисплея (S) при выводе данных	37 мкс
Display on/off control	0	0	0	0	0	0	1	D	C	B		Вкл/выкл. дисплей(D), курсор(C) и его мерцание(B)	37 мкс
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	X	X		Двигает курсор и смещает дисплей по DDRAM	37 мкс
Function set	0	0	0	0	1	DL	N	F	X	X		Установка интерфейса(DL), число строк(N) и шрифт символов(F)	37 мкс
Set CGRAM address	0	0	0	1	ACG5	ACG4	ACG3	ACG2	ACG1	ACG0		Установка счетчика адреса CGRAM. После этого можно записывать данные в CGRAM	37 мкс
Set DDRAM address	0	0	1	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0		Установка счетчика адреса DDRAM	37 мкс
Read busy flag & address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Если BF = 1 то контроллер ЖКИ выполняет внутреннюю операцию (занят). AC6-AC0 – текущее значение адреса DDRAM	0 мкс
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0		Запись данных в RAM	37 мкс
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0		Чтение данных из RAM	37 мкс

I/D = 1: адрес DDRAM увеличивается;

I/D = 0: уменьшается;

S = 1: сдвиг рабочей области дисплея по DDRAM разрешен;  
D = 1: дисплей (изображение) включен;  
C = 1: курсор включен;  
V = 1: мерцание курсора включено;  
S/C = 1: сдвинуть дисплей;  
S/C = 0: переместить курсор  
R/L = 1: вправо R/L = 0: влево;  
DL = 1: 8 bit; DL = 0: 4 bits;  
N = 1: 2 lines N = 0: 1 line;  
F = 1: 5x10; F = 0: 5x8;  
ACG: CGRAM address;  
ADD: DDRAM address (адрес курсора);  
AC: Address counter DD и CGRAM адресов

### **Инициализация ЖКИ.**

Есть 2 способа инициализации контроллера ЖКИ:

1. Через внутреннюю схему сброса.

2. В ручном режиме (через посылку в него ряда команд, которыми мы задаем режим работы ЖКИ) Внутренняя схема сброса контроллера начинает работать сразу после включения питания. В этом есть один минус – если питание у нас «ползет» до рабочего уровня медленно (медленнее, чем за 10 мс), то самоинициализация контроллера может пройти некорректно. При этом способе инициализации контроллер сам исполняет следующие команды:

1. Display clear

2. Function set:

DL = 1; 8-bit interface data

N = 0; 1-line display

F = 0; 5x8 dot character font

3. Display on/off control:

D = 0; Display off

C = 0; Cursor off

V = 0; Blinking off

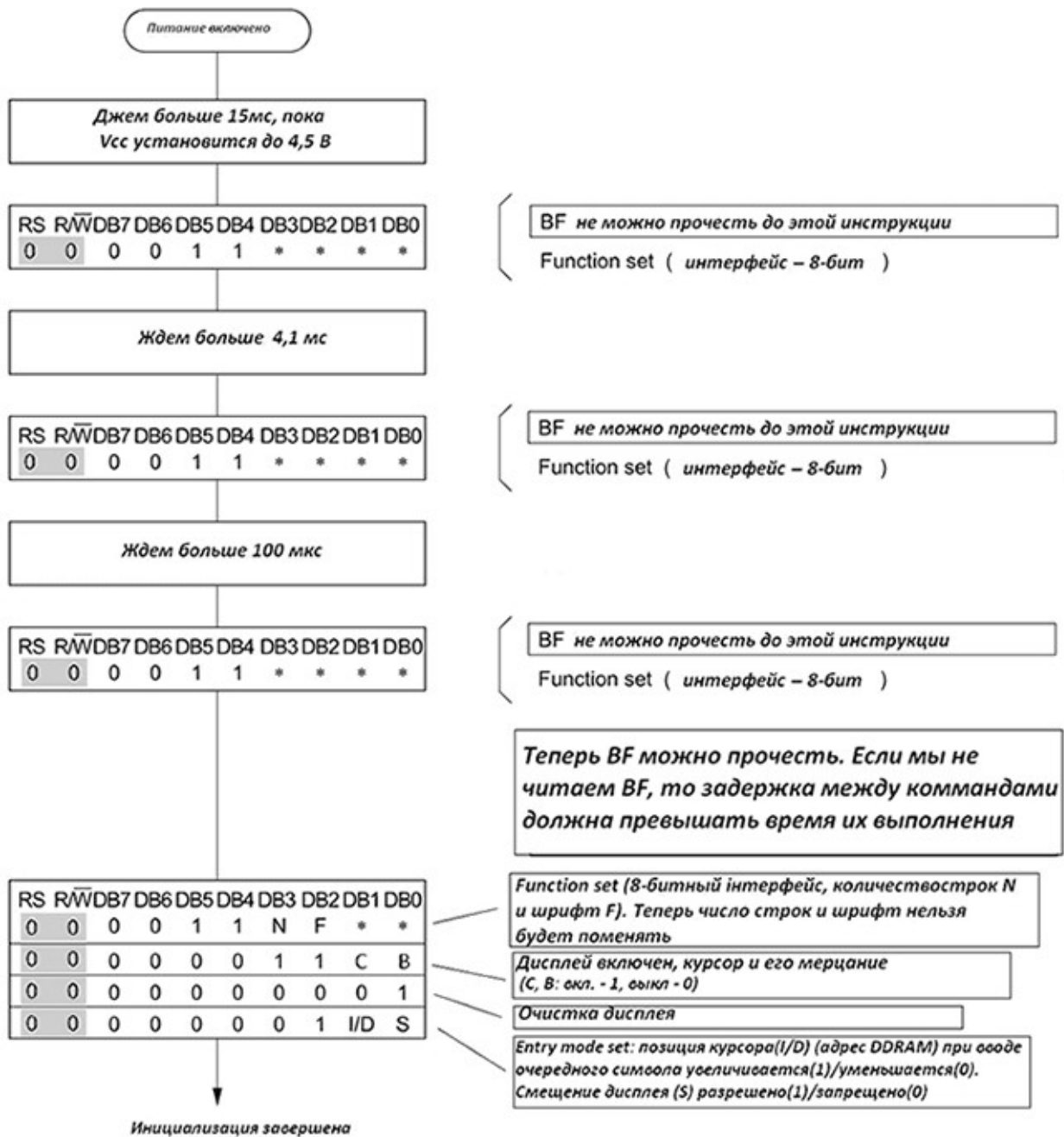
4. Entry mode set:

I/D = 1; Increment by 1

S = 0; No shift

Второй способ исключает зависимость схемы от источника питания. Для инициализации контроллера ЖКИ в ручном режиме необходимо исполнить следующий алгоритм:

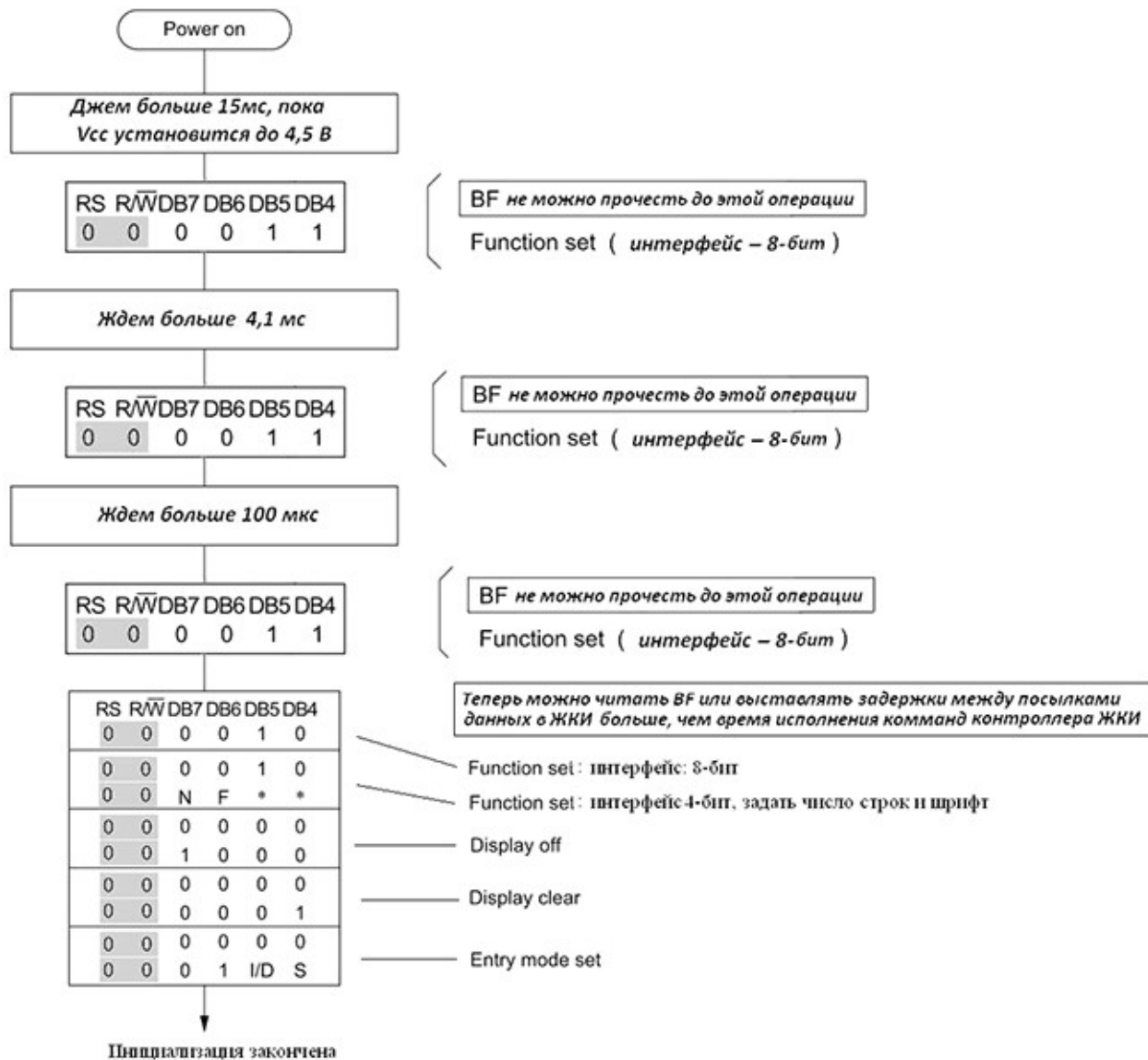
*8-битный параллельный интерфейс:*



#### 4-битный параллельный интерфейс:

Как видно, здесь нет ничего сложного: посылаем в ЖКИ команду за командой, учитывая время их исполнения (около 40 мкс) или проверяя флаг занятости контроллера ЖКИ (тогда надо посадить пин RW на лапку микроконтроллера и выставлять его в «1», когда хотим узнать, занят ЖКИ или нет).

#### 4-битный параллельный интерфейс:



### Принцип работы ЖК модуля.

Типичный внешний вид ЖК символьного модуля приведен на рис. 3.1.

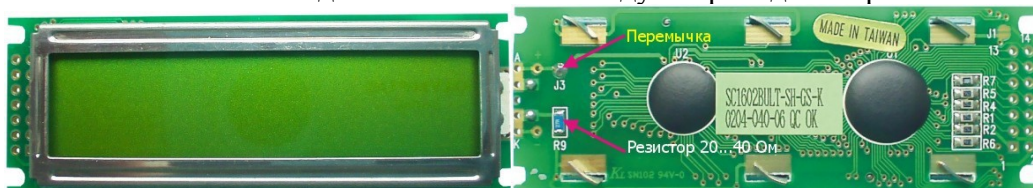


Рис. 3.1. Вид спереди и вид сзади ЖК модуля.

Как уже говорилось выше, изначально HD44780 имеет predeterminedную таблицу символов, размещенную в ОЗУ знакогенератора CGRAM (Character Generator RAM). Для отображения любого из них программа микроконтроллера должна передать координаты позиции и, непосредственно за ними, сам адрес символа из CGRAM. Пример таблицы CGRAM приведен на рис. 3.2.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1			!	1	A	Q	a	q			Г	Я	Ш	Щ	Ц	Ч	Ш	Щ
2			"	2	B	R	b	r			ё	б	ъ	ш	щ	ц	ч	ш
3			#	3	C	S	c	s			Ж	В	Ы	!!	д	з	д	з
4			\$	4	D	T	d	t			Э	Г	Ь	Ъ	Ф	Ф	Ф	Ф
5			%	5	E	U	e	u			И	ё	э	х	ц	ц	ц	ц
6			&	6	F	V	f	v			Й	ж	ю	ъ	щ	щ	щ	щ
7			'	7	G	W	g	w			Л	э	я	І	'	Е	Е	Е
8			(	8	H	X	h	x			П	и	«	И	"	К	К	К
9			)	9	I	Y	i	y			У	й	»	↑	~	М	М	М
A			*	:	J	Z	j	z			Ф	к	«	↓	é	Н	Н	Н
B			+	:	K	[	k	]			Ч	л	"	Н	Ф	К	К	К
C			,	<	L	φ	l	φ			Ш	м	№	Н	і	У	У	У
D			-	=	M	]	m	]			Ь	н	с	Н	Ф	Ф	Ф	Ф
E			.	>	N	^	n	^			Ы	п	f	Ъ	∅	Ф	Ф	Ф
F			/	?	O	_	o	_			Э	т	é	•	о	■	■	■

Рис3.2

Заглавные и прописные буквы латинского алфавита, числовые знаки, а также большинство знаков препинания совпадают в ней с кодами ASCII. Набор символов, размещенных по адресам 0xA0...0xFF, содержит национальный алфавит (в данном случае кириллицу) того региона, где предполагается его использование. Первые 16 ячеек CGRAM имеют особое значение. При желании, в них могут быть записаны любые пользовательские символы, которых нет таблице (сразу после включения модуля в них находится случайная информация).

Типичная нумерация и функциональное назначение выводов ЖКИ приведены в табл.2. Кроме напряжения питания контроллера VCC, модуль имеет вход регулировки контрастности изображения V0. Питание подсветки (если таковая имеется) подается на выводы А и К. HD44780 взаимодействует с AVR через 8-битную двунаправленную шину команд/данных DB7:DB0. Временная диаграмма работы шины показана на рис. 3.3а. В момент записи информации в ЖКИ ведущий микроконтроллер выставляет на линиях DB7...DB0 8-разрядный код, после чего формирует на выводе Е стробирующий импульс (активный фронт – задний). По окончании импульса должна быть выдержана пауза до начала новой транзакции. Признаком записи команды/ данных является состояние линии RS. При RS=0 происходит запись команды, при RS=1 – данных. Когда необходимо считать данные из индикатора, то выводы порта DB7:DB0 микроконтроллера настраиваются на ввод. Затем следует импульс подтверждения на линии Е и байт данных переписывается во внутренний регистр для дальнейшей обработки. Направление передачи данных определяет уровень на линии R/W (R/W =1 – чтение из индикатора, R/W =0 – запись в индикатор). В реальных приложениях, как правило, нет необходимости в чтение данных. Поэтому вывод R/W всегда соединяют с общим проводом.

Таблица 2. Функциональное назначение выводов символьного ЖКИ на базе HD44780:



Номер вывода	Название выводов	Функциональное назначение
1	GND	Общий вывод
2	VCC	Напряжение питания
3	V0	Напряжение управления контрастностью
4	RS	Выбор записи команды/данные
5	R/W	Выбор направления передачи данных запись/чтение
6	E	Вход тактовых импульсов
7-14	BD7-DB0	Шина данных
15	A	Анод светодиодной подсветки
16	K	Катод светодиодной подсветки

Одна из типовых схем подключения микроконтроллера типа AVR к ЖК индикатору типа A162-D приведена на рис.3.4а.

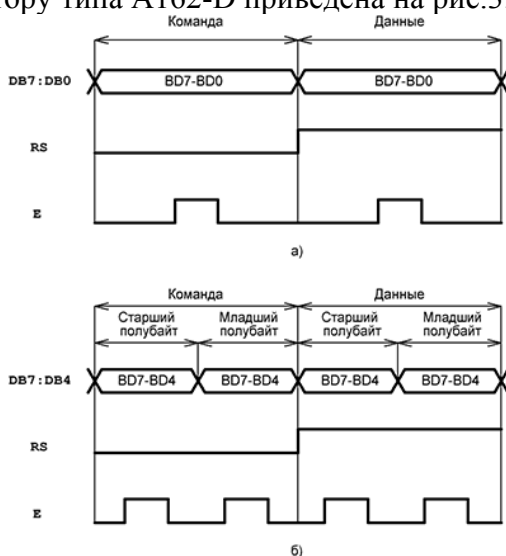


Рис. 3.3. Последовательность передачи данных в HD44780: а - по 8-разрядной шине команд/данных; б - по 4- разрядной шине команд/данных.

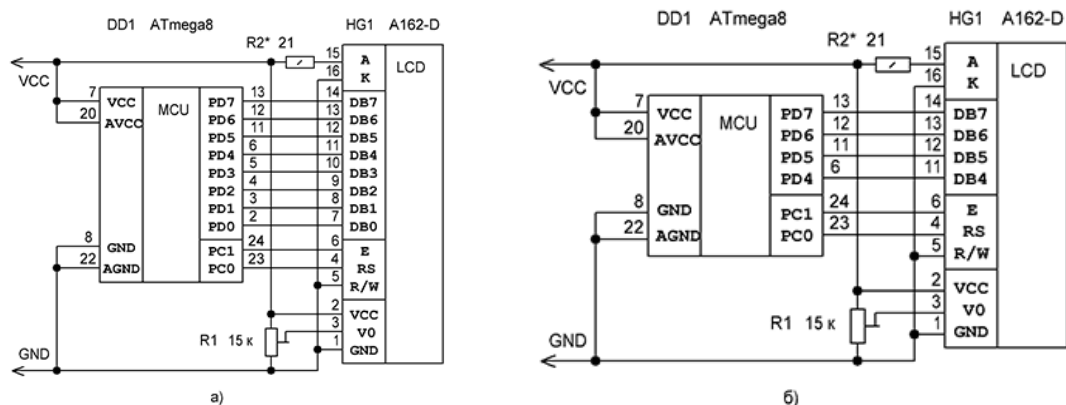


Рис. 3.4. Схема подключения символического ЖКИ к микроконтроллеру: а – при использовании 8-разрядной шины команд/данных; б - при использовании 4- разрядной шины команд/данных.

Для управления ЖКИ может быть использован также 4-проводный интерфейс (см. схему подключения на рис. 3.4б), что позволяет сэкономить 4 линии ввода-вывода, при незначительном усложнении программы.

В этом случае 4-разрядную шину команд/данных формируют линии DB7... DB4 (линии DB3...DB0 остаются незадействованными). Скорость записи снижается в 2 раза, но это, обычно, не вызывает ни каких проблем во время работы.

Последовательность передачи данных показана на рис. 3.3б. Команды/ данные передается за два такта. Первым следует старший полубайт, вторым – младший.

Каждая тетрада, естественно, должна быть зафиксирована импульсом на линии E.

Контроллер HD44780 имеет буфер видеопамати DDRAM (Display Data RAM), из которой символы переносятся на дисплей. Объем DDRAM зависит от числа строк и позиций на экране. Адреса ячеек видеопамати первой строки 0x80...0xA8, второй 0xC0...0xE8. В текущий момент времени в окно дисплея попадают только 16 символов из DDRAM рис. 2.2 (положение окна можно изменять программно).\_\_

### **Оборудование и материалы.**

Комплекс лабораторных работ по исследованию работы ЖК символьных индикаторов с микроконтроллером типа AVR выполняется на учебном макетном лабораторном комплексе РТМТЛ-3.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микропроцессор (Atmega8535, Atmega16 или совместимый); резистивно-диодный последовательный (работающий по интерфейсу RS232 через СОМ-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; набор из 11 светодиодов, подключенных к выводам 14 – 21, 33 – 35 микроконтроллера см. рис. 4.1; набор из 11 кнопок с фиксацией нормально разомкнутых (кнопка нажата — замкнута; отжата — разомкнута); 1 кнопку без фиксации нормально разомкнутую; 1 потенциометр (переменный резистор КОНТРАСТНОСТЬ), подключенный подключенный средней точкой к выводу VEE исследуемого LCD индикатора, для измерения напряжения контрастности выведены клеммы типа «RCA тюльпан» для подключения мультиметра; 4 потенциометра, подключенные к выводам 40, 39, 38, 37; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к СОМ –порту ПЭВМ **ТОЛЬКО** проводом типа СОМ 9m/9f («мама» - «папа»).

Принципиальная электрическая схема прибора приведена на рис. 4.1. Демонстрационные примеры готовых программ, на основе которых могут быть построены задания по программированию микроконтроллера, приведены в папке Examples\_RTMTL-3.

**Задание 1.** Реализация 8 битного режима работы LCD ЖК символьного индикатора. При этом с помощью кнопок S1 – S11 можно моделировать неисправности ЖК индикатора при обрыве того или иного провода, соединяющего LCD дисплей с микроконтроллером.

**Задание 2.** Реализация 4 битного режима работы LCD ЖК символьного индикатора.

**Задание 3.** Реализация 8 битного режима работы LCD индикатора и работа со светодиодами. При этом 8 светодиодов повторяют логическое состояние порта данных дисплея, а три светодиода — состояние выводов RS, RW и E. Можно наглядно наблюдать команды управления и данные, приходящие на ЖК индикатор.

**Задание 4.** Работа с потенциометрами, подключенными к портам PA0-PA3 микроконтроллера. Каждый из потенциометров подключен средней точкой к соответствующему выводу микроконтроллера, который является входом АЦП. Измерения выводятся на ЖК индикатор.

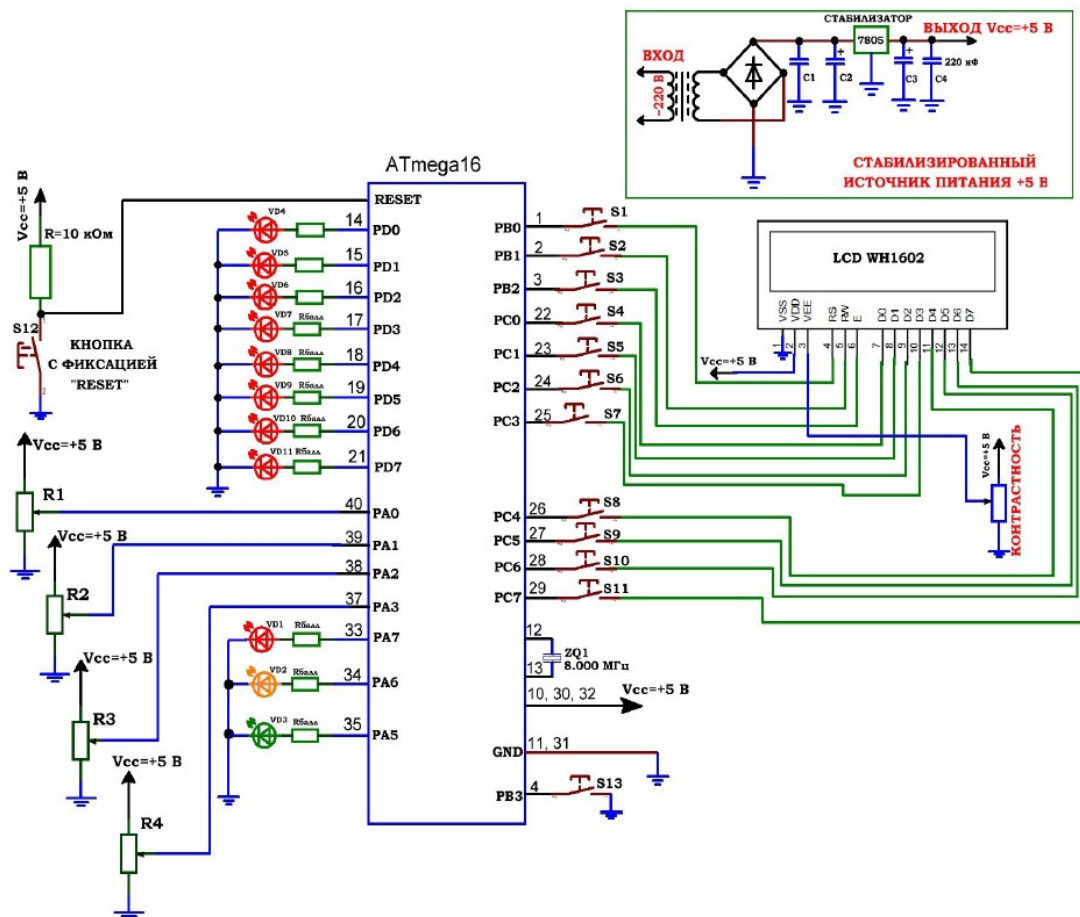


Рис. 4.1. Принципиальная электрическая схема учебного стенда для изучения работы ЖК символического индикатора с микроконтроллером AVR.

**Задание 5.** Работа с потенциометрами и кнопкой S13, подключенной к порту PB3. Кнопка настроена на переключение каналов АЦП. Измерения выводятся на ЖК индикатор.

**Задание 6.** Переменный резистор R1, подключен средней точкой ко входу 40 АЦП микроконтроллера. Измеренное значение напряжения в кодах АЦП выводится на ЖК индикатор с одновременной визуализацией условного уровня с помощью 8 светодиодов VD4 – VD11. Количество зажженных светодиодов пропорционально коду АЦП. Светодиоды VD1 – VD3 при этом мигают \_

#### Указания по технике безопасности

Соответствуют технике безопасности по работе с компьютерной техникой.

#### Ход работы

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.

2. Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR. Ознакомиться с кратким методическим руководством по программированию микроконтроллеров, приведенных в руководстве для учебного стенда РТМТЛ-2. Ознакомиться со статьей «Практическое объяснение работы ЖКИ модулей с HD44780» (файл Kea\_45\_1.html). Ознакомиться с паспортными данными на исследуемый ЖК индикатор (файл lcd\_hd44780.pdf).

3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.

4. Загрузить операционную систему согласно стандартным процедурам загрузки.

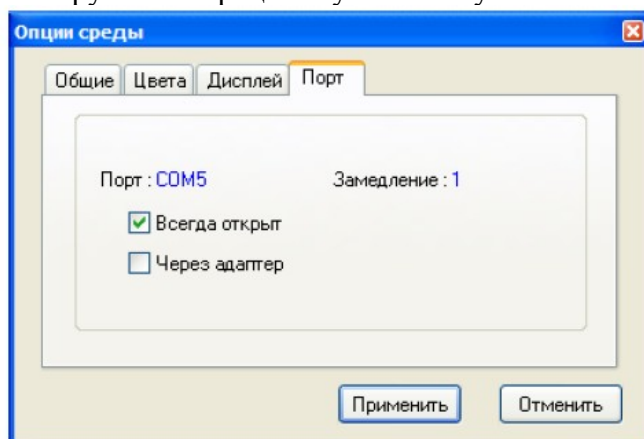


Рис. 5.1

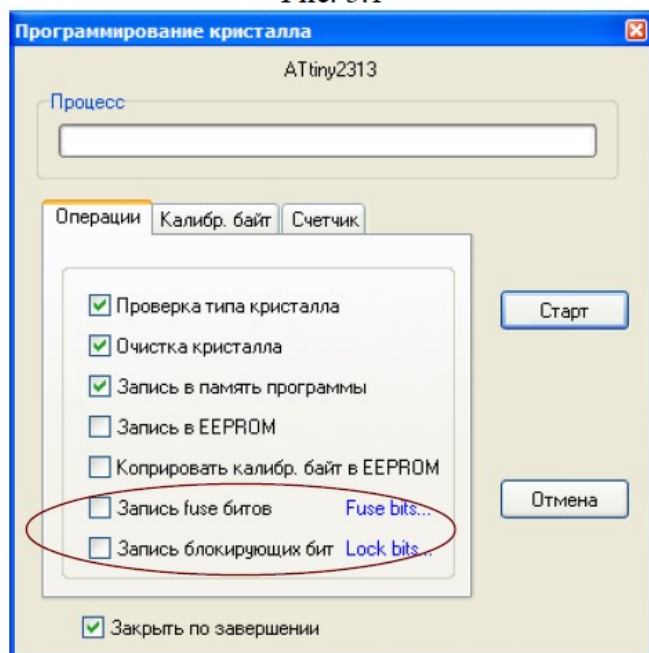


Рис. 5.2

5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder.

6. Запустить программу Algorithm Builder для работы с учебной установкой для данного эксперимента пользуясь ярлыком на рабочем столе либо другим способом, указанным лаборантом. Для запуска можно воспользоваться комплексной программной оболочкой LabVisual для РТМТЛ-1-5.

7. Подключить разъем «ПРОГРАММАТОР» учебного прибора к COM – порту ПЭВМ проводом типа COM 9m/9f («мама» - «папа»)

8. Настроить программу Algorithm Builder для работы с данным COM портом ОПЦИИ-ОПЦИИ СРЕДЫ-ПОРТ рис. 5.1.

9. Для подробных инструкций следует обращаться к методическому руководству по среде Algorithm Builder.

10. Загрузить Пример 1 (папка 1) в среду Algorithm Builder (ФАЙЛ-ОТКРЫТЬ-ВЫБРАТЬ ФАЙЛ \*.alp).

11. «Прошить» тестовую программу в кристалл. Для этого нажать кнопку «ЗАПУСК С КРИСТАЛЛОМ», при этом откроется диалог программирования кристалла

рис. 5.2. **Перед прошивкой следует нажать кнопку с фиксацией «RESET» на учебном приборе.**

12. Перед нажатием кнопки «СТАРТ» следует внимательно проверить положения флажков установки программатора рис. 5.2. Особенно внимательно следует отнестись к опциям «ЗАПИСЬ Fuse bit» и «ЗАПИСЬ блокирующих бит». **ФЛАЖКИ ЭТИХ ОПЦИЙ ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ СНЯТЫ, Т. К. НЕ ПРАВИЛЬНО ПРОШИТЫЕ FUSE БИТЫ МОГУТ ПРИВЕСТИ К ДАЛЬНЕЙШЕЙ НЕВОЗМОЖНОСТИ РАБОТЫ С ДАННЫМ МИКРОКОНТРОЛЛЕРОМ!**

13. **Отжать кнопку RESET и проверить работу тестовой программы.**

14. Загрузить примеры 2 — 6 из соответствующих папок и прошить в кристалл каждую из тестовых программ.

15. На основании данных примеров составить собственные программы, реализующие различные задачи (изменить вывод на ЖК индикатор, мигание светодиодов, опрос кнопок и потенциометров). Составить программы для реализации задач как для 8-битного, так и для 4-битного режима работы ЖК индикатора.

16. Отключая соответствующие выходы ЖК индикатора от микроконтроллера с помощью кнопок управления, моделировать неисправности, которые могут возникнуть при плохом соединении или обрыве связи LCD – микроконтроллер.

17. Измерить с помощью мультиметра напряжение, подаваемое на вход VEE (контрастность) дисплея при наилучшем визуальном отображении символов.

18. Для составления программ и подробного изучения работы микроконтроллеров AVR следует обращаться к учебной литературе, полному руководству к микроконтроллерам семейства AVR, а также к документации по среде Algorithm Builder и документацией на ЖК индикатор.

19. По окончании работы следует закрыть программу и все открытые подпрограммы, закрыть виртуальную среду VirtualBox (при работе в среде Linux).

20. Выключить компьютер, нажав на кнопку, находящуюся в крайнем нижнем левом углу экрана. Из доступных действий выбрать «ВЫХОД»---> «ВЫКЛЮЧИТЬ КОМПЬЮТЕР».

21. Отключить установку от сети, поставив переключатели «СЕТЬ» на панели установки в положение «ВЫКЛ» и вынуть сетевые вилки из розеток

### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

### **Контрольные вопросы**

1. Какие способы взаимодействия имеет микроконтроллера AVR с ЖКИ модулем на базе контроллера HD44780
2. Сколько строк имеет ЖКИ модуль на базе контроллера HD44780?
3. Сколько символов в строке воспроизводит ЖКИ модуль на базе контроллера HD44780?
4. Какие регистры имеет ЖКИ модуль на базе контроллера HD44780 для управления ?
5. Какое количество символов и каким образом воспроизводит ЖКИ модуль на базе контроллера HD44780?
6. Каким сигналом синхронизируется запись кода символа в ЖКИ модуль на базе контроллера HD44780?
7. Каким сигналом синхронизируется запись кода команды в ЖКИ модуль на базе контроллера HD44780?

### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах.
3. Грамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.
4. 4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).

## Лабораторная работа 9

### Разработка программ для связи микроконтроллера AVR как регулятора

#### Цель работы

Изучение инструментальных средств разработки программного обеспечения МК.  
Разработка программ вывода информации на ЖКИ микроконтроллера AVR и программ для связи микроконтроллера AVR с ПК-ЭВМ.

#### Компетенции:

Индекс	Формулировка:
ОПК-3	способностью решать задачи анализа и расчета характеристик электрических цепей
ОПК-7	способностью учитывать современные тенденции развития теории автоматического управления, электронного оборудования, измерительной и вычислительной техники, информационных технологий в профессиональной деятельности
ПК-1	способностью выполнять эксперименты на действующих объектах по заданным методикам, оценивать динамические характеристики рассматриваемых объектов и идентифицировать передаточные функции объектов с применением современных информационных технологий и технических средств
ПК-7	готовностью участвовать в разработке и изготовлении стендов для комплексной отладки и испытаний программно-аппаратных управляющих комплексов
ПК-8	готовностью к внедрению результатов разработок средств и систем автоматизации и управления в производстве с использованием современных программируемых логических контроллеров (ПЛК)
ПК-11	готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления
ПК-14	способностью разрабатывать электромеханические системы и использовать современную элементную базу при проектировании средств и систем управления

#### Теоретическая часть

Микроконтроллеры семейства в зависимости от модели имеют в своем составе от двух до шести таймеров/счетчиков общего назначения  
(Табл. 7.1).

**Таблица 7.1. Таймеры/счетчики общего назначения**

Таймер/счетчик	ATmega8515x	ATmega8535x	ATmega8x, ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x	ATmega325x/3250x, ATmega645x/6450x	ATmega640x, ATmega1280x/1281x, ATmega2560x/2561x
Таймер/счетчик T0 (8-битный)	•	•	•	• <sup>1)</sup>	•	•	•	•	•	•
Таймер/счетчик T1 (16-битный)	•	•	•	•	•	•	•	•	•	•
Таймер/счетчик T2 (8-битный)		• <sup>1)</sup>	• <sup>1)</sup>	•	• <sup>1)</sup>	• <sup>1)</sup>	• <sup>1)</sup>	• <sup>1)</sup>	• <sup>1)</sup>	• <sup>1)</sup>
Таймер/счетчик T3 (16-битный)				•		•				•
Таймер/счетчик T4 (16-битный)										•
Таймер/счетчик T5 (16-битный)										•
<sup>1)</sup> Асинхронный таймер/счетчик.										

Как видно из таблицы, во всех моделях микроконтроллеров семейства присутствуют как минимум два таймера/счетчика — T0 и T1. Таймер/счетчик T0 имеет минимальный набор функций, зависящий, тем не менее, от модели микроконтроллера. В одних моделях он может использоваться только для отсчета и измерения временных интервалов или как счетчик внешних событий. В других моделях к этим функциям добавляется возможность генерации сигналов с широтно-импульсной модуляцией (ШИМ) фиксированной разрядности (один или два канала), а также возможность работать в асинхронном режиме в качестве часов реального времени (в моделях ATmega64x/128x).

Таймер/счетчик T1 тоже может использоваться для отсчета временных интервалов и как счетчик внешних событий. Кроме того, он может осуществлять запоминание своего состояния по внешнему сигналу. Как и таймер/счетчик T0, он может работать в качестве 2- или 3-канального широтно-импульсного модулятора, но уже переменной разрядности. Количество каналов ШИМ зависит от модели.

Таймер/счетчик T2 практически полностью аналогичен таймеру/счетчику T0. Во всех моделях, кроме ATmega64x/128x, таймер/счетчик T2 может работать в асинхронном режиме.

Таймеры/счетчики T3...T5 по функциональным возможностям идентичны таймеру/счетчику T1.

В составе всех микроконтроллеров семейства имеется также сторожевой таймер, являющийся неременным атрибутом всех современных микроконтроллеров. Этот таймер позволяет избежать несанкционированного закливания программы, возникающего по тем или иным причинам.

## 7.2. Назначение выводов таймеров/счетчиков

Каждый таймер/счетчик использует один или более выводов микроконтроллера. Как правило, эти выводы — линии портов ввода/вывода общего назначения, а функции, реализуемые этими выводами при работе совместно с таймерами/счетчиками, являются их альтернативными функциями.

Все выводы микроконтроллеров, используемые таймерами/счетчиками общего назначения, приведены в Табл. 7.2. Там же указаны функции этих выводов.



Таблица 7.2. Выводы, используемые таймерами/счетчиками общего назначения

Название	Микроконтроллеры												Описание
	ATmega8515x	ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x/325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x/1280x/2560x	
T0	PB0	PB0	PD4	PB0	—	PD4	PB0	PB0	PG4	PG4	PD7	PD7	Вход внешнего сигнала таймера T0
OC0	PB0	PB3	—	PB3	PB4	—	PB0	—	—	—	—	—	Выход схемы сравнения таймера T0
OC0A	—	—	—	—	—	PD6	—	PB3	PB4	PB4	PB7	PB7	
OC0B	—	—	—	—	—	PD5	—	PB4	—	—	PG5	PG5	
T1	PB1	PB1	PD5	PB1	PD6	PD5	PB1	PB1	PG3	PG3	PD6	PD6	Вход внешнего сигнала таймера T1
ICP	PE0	—	—	—	—	—	—	—	—	—	—	—	Вход захвата таймера T1
ICP1	—	PD6	PB0	PD6	PD4	PB0	PE0	PD6	PD0	PD0	PD4	PD4	Вход захвата таймера T1
OC1A	PD5	PD5	PB1	PD5	PB5	PB1	PD5	PD5	PB5	PB5	PB5	PB5	Выход схемы сравнения таймера T1
OC1B	PE2	PD4	PB2	PD4	PB6	PB2	PE2	PD4	PB6	PB6	PB6	PB6	
OC1C	—	—	—	—	PB7	—	—	—	—	—	PB7	PB7	
T2	—	—	—	—	PD7	—	—	—	—	—	—	—	Вход внешнего сигнала таймера T2
OC2	—	—	PB3	PD7	PB7	—	PB1	—	—	—	—	—	Выход схемы сравнения таймера T2
OC2A	—	—	—	—	—	PB3	—	PD7	PB7	PB7	PB4	PB4	
OC2B	—	—	—	—	—	PD3	—	PD6	—	—	—	PH6	
T3	—	—	—	—	PE6	—	—	—	—	—	PE6	PE6	Вход внешнего сигнала таймера T3
ICP3	—	—	—	—	PE7	—	PD3	—	—	—	PE7	PE7	Вход захвата таймера T3

(продолжение)

Название	АТмега												Описание
	8515x	8535x	8x	16x/32x	64x/128x	48x/88x/168x	162x	164x/324x/644x	165x/325x/645x	3250x/6450x	1281x/2561x	640x/1280x/2560x	
OC3A	—	—	—	—	PE3	—	PD4	—	—	—	PE3	PE3	Выход схемы сравнения таймера T3
OC3B	—	—	—	—	PE4	—	PB4	—	—	—	PE4	PE4	
OC3C	—	—	—	—	PE5	—	—	—	—	—	PE5	PE5	
T4	—	—	—	—	—	—	—	—	—	—	—	PH7	Вход внешнего сигнала таймера T4
ICP4	—	—	—	—	—	—	—	—	—	—	—	PL0	Вход захвата таймера T4
OC4A	—	—	—	—	—	—	—	—	—	—	—	PH3	Выход схемы сравнения таймера T4
OC4B	—	—	—	—	—	—	—	—	—	—	—	PH4	
OC4C	—	—	—	—	—	—	—	—	—	—	—	PH5	
T5	—	—	—	—	—	—	—	—	—	—	—	PL2	Вход внешнего сигнала таймера T5
ICP5	—	—	—	—	—	—	—	—	—	—	—	PL1	Вход захвата таймера T5
OC5A	—	—	—	—	—	—	—	—	—	—	—	PL3	Выход схемы сравнения таймера T5
OC5B	—	—	—	—	—	—	—	—	—	—	—	PL4	
OC5C	—	—	—	—	—	—	—	—	—	—	—	PL5	
TOSC1	—	PC6	PB6	PC6	PG4	PB6	PD4	PC6	— <sup>1)</sup>	— <sup>1)</sup>	PG4	PG4	Вход для подключения резонатора
TOSC2	—	PC7	PB7	PC7	PG3	PB7	PD5	PC7	— <sup>1)</sup>	— <sup>1)</sup>	PG3	PG3	Выход для подключения резонатора

<sup>1)</sup> Отдельные контакты ввода/вывода, совмещенные с выводами XTAL1/XTAL2.

Не забывайте о том, что при использовании альтернативных функций линий портов ввода/вывода необходимо, как правило, самостоятельно сконфигурировать эти выходы в соответствии с их функциональным назначением.

### 7.3. Прерывания от таймеров/счетчиков

В старых моделях для разрешения/запрещения прерываний от таймеров/счетчиков использовалось от одного до двух регистров ввода/вывода. В новых моделях число таких регистров С...6) равно числу счетчиков в конкретной модели. Точно так же дело обстоит и с регистрами, содержащими флаги прерываний. Названия и адреса всех этих регистров приведены в Табл. 7.3.

(продолжение)

Название	ATmega851x												ATmega8535x	ATmega8x	ATmega16x/32x	ATmega64x/128x	ATmega48x/88x/168x	ATmega162x	ATmega164x/324x/644x	ATmega165x/325x/645x	ATmega3250x/6450x	ATmega1281x/2561x	ATmega640x/1280x/2560x	Описание
	—	—	—	—	PE3	—	PD4	—	—	—	PE3	PE3												
OC3A	—	—	—	—	PE3	—	PD4	—	—	—	PE3	PE3	Выход схемы сравнения таймера T3											
OC3B	—	—	—	—	PE4	—	PB4	—	—	—	PE4	PE4												
OC3C	—	—	—	—	PE5	—	—	—	—	—	PE5	PE5												
T4	—	—	—	—	—	—	—	—	—	—	—	PH7	Вход внешнего сигнала таймера T4											
ICP4	—	—	—	—	—	—	—	—	—	—	—	PL0	Вход захвата таймера T4											
OC4A	—	—	—	—	—	—	—	—	—	—	—	PH3	Выход схемы сравнения таймера T4											
OC4B	—	—	—	—	—	—	—	—	—	—	—	PH4												
OC4C	—	—	—	—	—	—	—	—	—	—	—	PH5												
T5	—	—	—	—	—	—	—	—	—	—	—	PL2	Вход внешнего сигнала таймера T5											
ICP5	—	—	—	—	—	—	—	—	—	—	—	PL1	Вход захвата таймера T5											
OC5A	—	—	—	—	—	—	—	—	—	—	—	PL3	Выход схемы сравнения таймера T5											
OC5B	—	—	—	—	—	—	—	—	—	—	—	PL4												
OC5C	—	—	—	—	—	—	—	—	—	—	—	PL5												
TOSC1	—	PC6	PB6	PC6	PG4	PB6	PD4	PC6	— <sup>1)</sup>	— <sup>1)</sup>	PG4	PG4	Вход для подключения резонатора											
TOSC2	—	PC7	PB7	PC7	PG3	PB7	PD5	PC7	— <sup>1)</sup>	— <sup>1)</sup>	PG3	PG3	Выход для подключения резонатора											

<sup>1)</sup> Отдельные контакты ввода/вывода, совмещенные с выводами XTAL1/XTAL2.

Форматы регистров, используемых для разрешения/запрещения прерываний от таймеров/счетчиков, показаны на Рис. 7.1...7.2, а описание их битов приведено в Табл. 7.4.

Для разрешения какого-либо прерывания от таймера/счетчика необходимо установить в 1 соответствующий бит регистра TIMSK (TIMSI)/ETIMSK и, разумеется, флаг I регистра SREG.

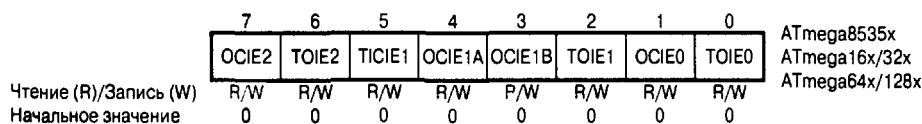


Рис. 7.1. Формат регистров TIMSK (а) и ETIMSK (б)

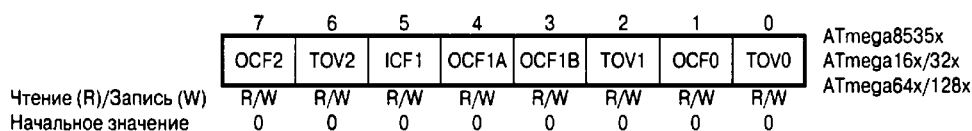
**Таблица 7.4. Биты регистров TIMSK, ETIMSK и TIMSK0...TIMSK5**

Название бита	Описание
TOIE <sub>n</sub>	Флаг разрешения прерывания по переполнению таймера/счетчика T <sub>n</sub> (n = 0...5)
OCIE <sub>n</sub>	Флаг разрешения прерывания по событию «Совпадение» таймера/счетчика T <sub>n</sub> (n = 0, 2)
OCIE <sub>nA</sub>	Флаг разрешения прерывания по событию «Совпадение А» таймера/счетчика T <sub>n</sub> (n = 0...5)
OCIE <sub>nB</sub>	Флаг разрешения прерывания по событию «Совпадение В» таймера/счетчика T <sub>n</sub> (n = 0...5)
OCIE <sub>nC</sub>	Флаг разрешения прерывания по событию «Совпадение С» таймера/счетчика T <sub>n</sub> (n = 1, 3...5)
TCIE <sub>n</sub>	Флаг разрешения прерывания по событию «Захват» таймера/счетчика T <sub>n</sub> (n = 1, 3)
ICIE <sub>n</sub>	Флаг разрешения прерывания по событию «Захват» таймера/счетчика T <sub>n</sub> (n = 1, 3...5)

Форматы регистров, используемых для индикации наступления прерываний от таймеров/счетчиков, показаны на Рис. 7.3 и Рис. 7.4, а описание их битов приведено в Табл. 7.5.

**Таблица 7.5. Биты регистра TIFR**

Название бита	Описание
TOV <sub>n</sub>	Флаг прерывания по переполнению таймера /счетчика T <sub>n</sub> (n = 0...5)
OCF <sub>n</sub>	Флаг прерывания по событию «Совпадение» таймера/счетчика T <sub>n</sub> (n = 0, 2)
OCF <sub>nA</sub>	Флаг прерывания по событию «Совпадение А» таймера/счетчика T <sub>n</sub> (n = 0...5)
OCF <sub>nB</sub>	Флаг прерывания по событию «Совпадение В» таймера/счетчика T <sub>n</sub> (n = 0...5)
OCF <sub>nC</sub>	Флаг прерывания по событию «Совпадение С» таймера/счетчика T <sub>n</sub> (n = 1, 3...5)
ICF <sub>n</sub>	Флаг прерывания по событию «Захват» таймера/счетчика T <sub>n</sub> (n = 1, 3...5)



**Рис. 7.3. Формат регистров TIFR (а) и ETIFR (б)**

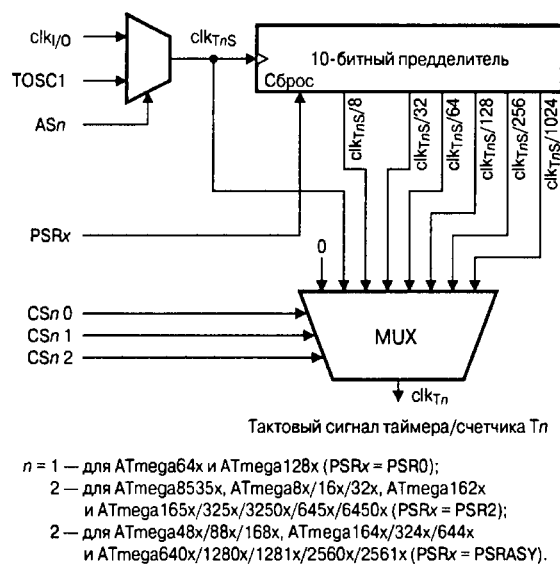
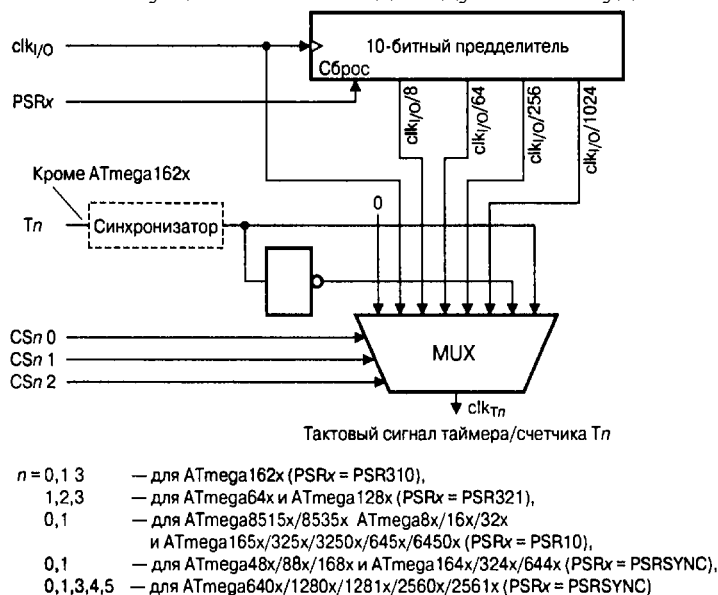
При наступлении какого-либо события соответствующий флаг регистра TIFR (TIFR<sub>rt</sub>)/ETIFR устанавливается в 1. При запуске подпрограммы обработки прерывания он аппаратно сбрасывается в 0. Любой флаг может быть также сброшен программно, записью в него лог. 1.

#### 7.4. Пределители таймеров/счетчиков

Блоки пределителей предназначены для формирования тактовых сигналов таймеров/счетчиков clkT<sub>0</sub>, clkT<sub>1</sub>, clkT<sub>2</sub>, clkT<sub>3</sub>. Упрощенная структурная схема блока пределителя таймеров/счетчиков, не имеющих асинхронного режима работы, приведена на Рис. 7.5, а. Структурная схема блока пределителя таймеров/счетчиков, имеющих возможность работы в асинхронном режиме, приведена на Рис. 7.5, б.

Как показано на рисунке, в состав каждого блока входят собственно 10-битный пределитель, выходной мультиплексор (селектор тактового сигнала), а для таймеров, имеющих возможность работы в асинхронном режиме, — еще и входной мультиплексор исходного тактового сигнала. По последней схеме выполнен пределитель таймера/счетчика TO моделей ATmega64x/128x и таймера/счетчика T2 остальных моделей. Следует иметь в виду, что все таймеры/счетчики каждой модели семейства, не имеющие асинхронного режима работы, используют один и тот же 10-

битный предделитель. При этом управление тактовым сигналом каждого таймера/счетчика осуществляется индивидуально и будет описано при их рассмотрении.



б)

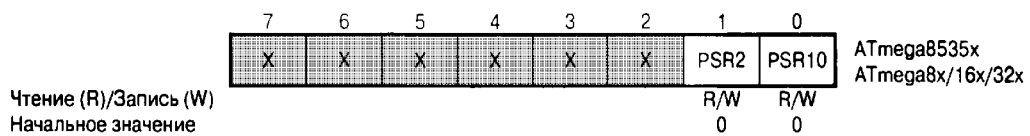
**Рис. 7.5.** Блок предделителя таймеров/счетчиков:

*a* — без асинхронного режима; *б* — с асинхронным режимом

Следует понимать, что предделители работают независимо от таймеров/счетчиков. Следствием этого является, в частности, неопределенный промежуток времени  $A \dots 7N+1$  тактов исходного сигнала, где  $N$  — коэффициент деления предделителя) между разрешением таймера/счетчика и первым его отсчетом при работе совместно с предделителем. Чтобы уйти от этой неопределенности, можно воспользоваться средствами, описанными в следующем подразделе.

#### 7.4.1. Управление предделителями

Помимо управления тактовым сигналом таймера/счетчика, все микроконтроллеры семейства позволяют осуществлять сброс предделителей, а отдельные модели позволяют также осуществлять их остановку. Для этого используется либо регистр специальных функций SFIOR, либо (в новых моделях) регистр управления таймеров/счетчиков GTCCR, расположенный по адресу \$23 (\$43). Формат этого регистра для различных моделей микроконтроллеров приведен на Рис. 7.6 (биты, не используемые для управления предделителями таймеров/счетчиков, указаны на рисунке как X).



**Рис. 7.6.** Управление пределителями таймеров/счетчиков — регистры SFIOR (а) и GTCCR (б)

Для сброса пределителей таймеров/счетчиков используются биты PSR<sub>jc</sub> (PSRSYNC/PSRASY) регистра. При записи в эти биты лог. 1 пределители соответствующих таймеров/счетчиков переводятся в исходное состояние. Биты сбрасываются в 0 аппаратно после выполнения операции сброса. Напоминаю, что один пределитель, как правило, используется несколькими таймерами/счетчиками, и соответственно сброс пределителя повлияет на все таймеры/счетчики, которые его используют.

Остановка всех пределителей микроконтроллера осуществляется записью лог. 1 в бит TSM регистра SFIOR или GTCCR. Последующий запуск пределителей осуществляется записью в бит TSM лог. 0. Указанная функция может использоваться, в частности, для синхронизации таймеров/счетчиков. После установки бита TSM и битов PSR<sub>jc</sub> (PSRSYNC/PSRASY) соответствующие таймеры/счетчики останавливаются и могут быть проинициализированы требуемыми значениями. После сброса бита TSM биты PSR<sub>x</sub> (PSRSYNC/PSRASY) аппаратно сбрасываются и все таймеры/счетчики начинают работать одновременно.

#### **Оборудование и материалы.**

Комплекс лабораторных работ по исследованию микроконтроллеров AVR выполняется на учебном макетном лабораторном комплексе РТМТЛ-1.

На передней крышке прибора закреплена отладочная плата, содержащая исследуемый микропроцессор (Atmega8535); резистивно-диодный последовательный (работающий по интерфейсу RS232 через COM-порт ПК) программатор; обвязку микроконтроллера согласно паспортным данным; 1 светодиод, подключенный к выводу 20 рис. 3.1; 1 кнопка с фиксацией нормально разомкнутая (кнопка нажата — замкнута; отжата — разомкнута), подключенная к выводу 39; 1 потенциометр (переменный резистор), подключенный средней точкой к выводу 40 микроконтроллера; выходы «НАГРУЗКА» XS2 — XS3; вход XS1 (для примера работы частотомера); подстроечный резистор R<sub>k</sub> — контрастность, подключенный средней точкой к выводу VEE LCD индикатора; а также кнопку с фиксацией «RESET», при нажатии которой вывод RESET кристалла соединяется с корпусом схемы. Для программирования прибора используется разъем «ПРОГРАММАТОР». При этом «ПРОГРАММАТОР» следует подключать к COM – порту ПЭВМ **ТОЛЬКО** проводом типа COM 9m/9f («мама» - «папа»).

Принципиальная электрическая схема прибора приведена на рис. 3.1.

Демонстрационные примеры готовых программ, на основе которых могут быть построены задания по программированию микроконтроллера, приведены в папке Examples\_RTMTL-1.

**Пример 1.** Работа со светодиодом, таймером, ЖК индикатором и входом PB1 микроконтроллера. Построение программы-частотомера. В данном примере на вывод 2 PB1 микроконтроллера (вход XS1) подаётся сигнал с внешнего генератора (в качестве генератора можно использовать программу «LabVisual ЗВУКОВОЙ ГЕНЕРАТОР ПК»). Сигнал предварительно усилен однотранзисторным резистивным усилителем, собранном на транзисторе VT1 BC547 и резисторах R2-R5. Микроконтроллер программируется таким образом, чтобы его TIMER1 был настроен на режим синхронизации внешним сигналом, поступающим на порт PB1. Программа 1 раз в секунду выполняет подсчет входных импульсов синхронизации, поступающих на вход PB1 (макс. Значение 65535).

Предусмотрен вывод показаний на LCD индикатор (максимальное значение 9999 Гц). Количество подсчитанных импульсов за 1 сек и есть частота входного сигнала в Гц.

Программа также одновременно работает с подключенным к микроконтроллеру ЖК LCD индикатором (инициализирует его, осуществляет вывод информации).

Одновременно с этим каждую секунду производится инвертирование состояние вывода PD6, к которому подключен светодиод VD1 (1 сек. Светодиод включен, 1 сек. - выключен). При отключенном от клеммы XS1 внешнем генераторе в нижней строке дисплея отображается INPUT DISCONNECT, при подключении внешнего генератора и частоте отличной от 0, в строке выводится сообщение INPUT CONNECT.

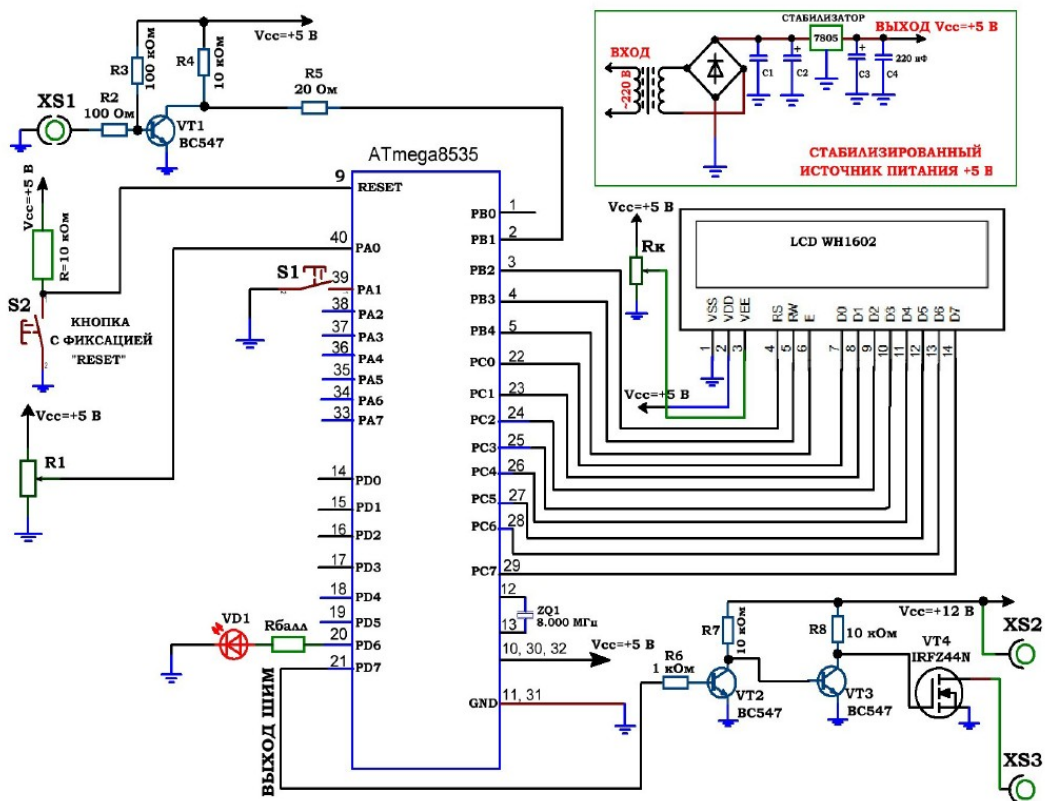


Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.

Рис. 3.1. Принципиальная электрическая схема учебного стенда для изучения микроконтроллеров серии Atmega8535.

2) **Пример 2.** Работа переменного резистора, кнопки, ЖК индикатора и выхода ШИМ генератора. ШИМ генерируется на выводе PD7 микроконтроллера и подается на трёхкаскадный усилитель мощности, два каскада которого собраны на биполярных транзисторах с общим эмиттером  $VT2=VT3=BC547$ , а третий каскад — на полевом транзисторе IRFZ44N с общим истоком. В стоковую цепь полевого транзистора подключается нагрузка (клеммы XS2 – XS3) относительно +12 В питания (колебания напряжения питания может составлять +10 ... +12 В, т. к. данный источник не стабилизированный). В качестве нагрузки может быть использован электродвигатель вентилятора либо лампа накаливания 12 В, 5 Вт. **Обратить внимание, что вентилятор плюсовым красным выводом должен подключаться строго к клемме XS2 (соблюдать полярность).**

При запуске программы длительность импульсов плавно нарастает с определенной скоростью, в нижней строке ЖК индикатора при этом выводится SPEED=X1. Кнопка с фиксацией, подключенная к выводу PA1 в этом режиме должна быть отжата, т. к. при нажатии кнопки скорость нарастания длительности импульсов

увеличивается в 4 раза и в нижней строке ЖК индикатора отображается  $SPEED=X4$ . В верхней строке ЖК индикатора выводится величина  $Ust=...$ , которая ограничивает нижний порог ШИМ и задается потенциометром R1, средней точкой подключенному к выводу 40 (PA0) микроконтроллера от 0 до 127 уровней. Отметим, что генерируемый ШИМ имеет всего 255 градаций (уровней), и, например, при  $Ust=127$  будет меняться от 127 до 255 единиц. При достижении максимального значения, длительность импульсов ШИМ плавно убавляется.

Для получения гармонического сигнала используется свободно распространяемый компонент LabVisual Генератор для ПК рис. 3.2, представляющий из себя низкочастотный комбинированный генератор функций. Выходные сигналы генератора могут поступать на два канала стерео выхода звуковой карты компьютера.

Генератор вырабатывает сигналы следующего вида:

- синусоидальный,
- прямоугольный (меандр),
- треугольный симметричный,
- пилообразный нарастающий,
- пилообразный спадающий.

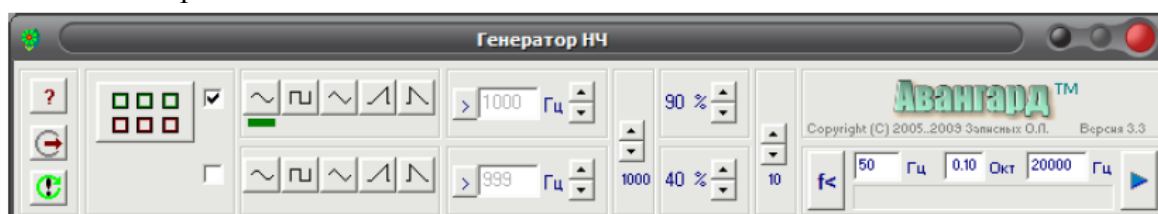



Рис. 3.2. Компонент «ГЕНЕРАТОР НЧ» для получения гармонических сигналов.

Оптимальные настройки.

Компонент может работать в различных режимах. В постоянном независимом режиме форма, частота и амплитуда сигналов по двум каналам устанавливаются отдельно. В постоянном зависимом режиме форма, частота и амплитуда сигналов по обоим каналам одинаковые, сигнал второго (правого) канала может быть произвольно сдвинут по фазе относительно сигнала первого (левого) канала. В режиме нарастания частоты сигнал по первому (левому) каналу меняется по частоте в заданных пределах с установленной скоростью. Сигнал второго канала выключен. В режиме нарастания амплитуды сигнал по первому (левому) каналу меняется по амплитуде в заданных пределах с установленной скоростью. Сигнал второго канала выключен. [Экранные органы управления снабжены всплывающими комментариями.](#)

**Частота** сигналов может устанавливаться двумя способами. Установка **прямым**

**вводом** производится в окне вида . Для редактирования следует нажать кнопку слева от окна, ввести новое значение, затем нажать клавишу **Enter**. Второй способ - **ступенчатое изменение** частоты с заданным шагом. Для этого имеются переключатели "выше-ниже". Двумя изменяется частота, третьим переключается шаг этого изменения. Дискретность установки частоты 1 Гц.


**Амплитуда** задается ступенчатым изменением с заданным шагом. Для этого имеются переключатели "выше-ниже". Двумя изменяется частота, третьим переключается шаг этого изменения. Дискретность установки амплитуды 1 % от максимальной.

**Сдвиг фаз** между сигналами **в зависимом режиме** устанавливается прямым вводом значений в градусах. Для редактирования нажмите кнопку слева от окна, введите новое значение, затем нажмите клавишу **Enter**. Дискретность установки сдвига 1 градус.

Ввиду большого размера буферов обмена звуковой карты управление прибором несколько замедленно. Реакция на переключения составляет примерно 1 секунду.



В случае возникновения пауз выходных сигналов воспользуйтесь экранной

кнопкой перезапуска (  ).

Программа запоминает установки и настройки, и восстанавливает их при следующем включении. \_\_

### **Указания по технике безопасности**

Соответствуют технике безопасности по работе с компьютерной техникой.

### **Ход работы**

1. Перед включением установки в сеть проверить целостность всех соединительных сигнальных и сетевых проводов. Все работы по подключению комплекса к компьютеру следует выполнять только при отключенных от сети приборах. Разобраться с принципиальными блок-схемами опытов, в назначении кнопок, переключателей и ручек прибора.

2. Перед выполнением работы следует изучить инструкцию по работе с учебной средой программирования Algorithm Builder, а также ознакомиться с полным методическим руководством по микроконтроллерам семейства AVR, изучить все доступные паспорта на исследуемые микроконтроллеры, а также на подключенный LCD индикатор.

3. Соединить монитор с системным блоком ПЭВМ, подключить клавиатуру и мышь к системному блоку используя стандартные провода для подключения. Подключить системный блок ПЭВМ и монитор к сети ~220 В.

4. Загрузить операционную систему согласно стандартным процедурам загрузки.

5. При необходимости, настроить компьютер для работы с учебной установкой и программной средой Algorithm Builder.

6. Запустить программу Algorithm Builder для работы с учебной установкой для данного эксперимента пользуясь ярлыком на рабочем столе либо другим способом, указанным лаборантом. Для работы можно воспользоваться комплексной программной оболочкой LabVisual для РТМТЛ-1-5-

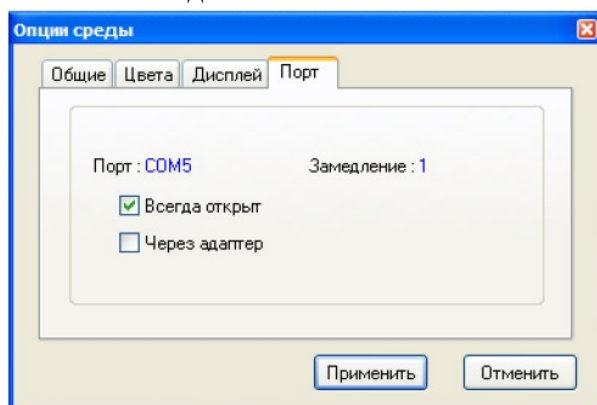


Рис. 4.1

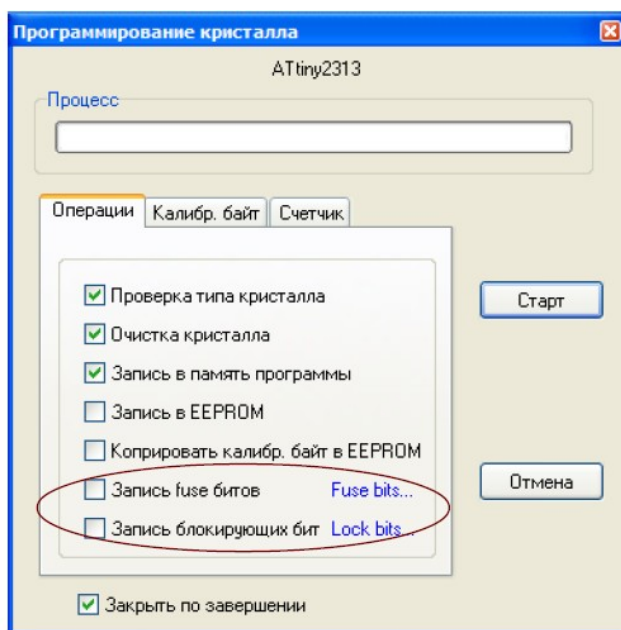


Рис. 4.2

6. Подключить разъём «ПРОГРАММАТОР» учебного прибора к СОМ – порту ПЭВМ проводом типа СОМ 9m/9f («мама» - «папа»)

7. Настроить программу Algorithm Builder для работы с данным СОМ портом ОПЦИИ-ОПЦИИ СРЕДЫ-ПОРТ рис. 4.1.

8. Для подробных инструкций следует обращаться к методическому руководству по среде Algorithm Builder.

9. Загрузить Пример 1 (папка 1) в среду Algorithm Builder (ФАЙЛ-ОТКРЫТЬ-ВЫБРАТЬ ФАЙЛ \*.alp).

10. «Прошить» тестовую программу в кристалл. Для этого нажать кнопку «ЗАПУСК С КРИСТАЛЛОМ», при этом откроется диалог программирования кристалла рис. 4.2. **Перед прошивкой следует нажать кнопку с фиксацией #RESET# на учебном приборе.**

11. Перед нажатием кнопки «СТАРТ» следует внимательно проверить положения флажков установки программатора рис. 4.2. **Особенно внимательно следует отнестись к опциям #ЗАПИСЬ Fuse bit# и #ЗАПИСЬ блокирующих бит#. ФЛАЖКИ ЭТИХ ОПЦИЙ ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ СНЯТЫ, Т. К. НЕ ПРАВИЛЬНО ПРОШИТЫЕ FUSE БИТЫ МОГУТ ПРИВЕСТИ К ДАЛЬНЕЙШЕЙ НЕВОЗМОЖНОСТИ РАБОТЫ С ДАННЫМ МИКРОКОНТРОЛЛЕРОМ!**

12. **Отжать кнопку RESET и проверить работу тестовой программы.**

13. Соединить выход LINE IN звуковой карты со входом XS1 стенда проводом «тюльпан — jack» из комплекта.

14. Проверить работу программа-частотомера, подавая на вход XS1 прямоугольный сигнал звуковой частоты 20 — 9000 Гц. При подачи сигнала с линейного выхода LINE OUT ПК посредством программы

«ГЕНЕРАТОР», следует установить амплитуду выходного сигнала генератора в программе не менее 80 % (рис. 3.2) и амплитуду выходного сигнала не менее 70 — 80 % непосредственно в звуковом драйвере (стандартная настройка громкости выходного сигнала). При низкой амплитуде входного сигнала, так же как и при слишком высокой амплитуде, частотомер может работать некорректно из за особенностей работы схемы однокаскадного транзисторного усилителя.

15. Загрузить пример 2 из соответствующей папки и прошить в кристалл тестовую программу.

16. Подключить к клеммам XS2 – XS3 нагрузку (вентилятор 12 В). **Обратить внимание, что вентилятор плюсовым красным выводом должен подключаться строго к клемме XS2 (соблюдать полярность).** Наблюдать работу тестовой программы.

17. Исходя из готовых примеров составить собственные программы в среде Algorithm Builder.

18. На основании Примера 1 можно составить программы, изменив, например, максимальное значение измеряемой частоты (в примере 9999 Гц), настройки TIMER1, действия, осуществляемые кнопкой S1 (включать светодиод), либо изменить вывод информации на ЖК индикатор. Также можно проводить различные операции с микроконтроллером.

19. На основании Примера 2 следует изучить работу ШИМ генератора микроконтроллера; изменить действия, осуществляемые кнопкой S1; изменить вывод информации на ЖК индикатор; поработать с потенциометром R1.

20. Для составления программ и подробного изучения работы микроконтроллеров AVR следует обращаться к учебной литературе, полному руководству к микроконтроллерам семейства AVR, а также к документации по среде Algorithm Builder.

21. По окончании работы следует закрыть программу и все открытые подпрограммы, закрыть виртуальную среду VirtualBox (при работе в среде Linux).

22. Выключить компьютер, нажав на кнопку, находящуюся в крайнем нижнем левом углу экрана. Из доступных действий выбрать «ВЫХОД»--> «ВЫКЛЮЧИТЬ КОМПЬЮТЕР».

23. Отключить установку от сети, поставив переключатели «СЕТЬ» на панели установки в положение «ВЫКЛ» и вынуть сетевые вилки из розеток. \_\_

### **Оформление отчета**

В отчете должны содержаться описание команд использованных в выполнении заданий и программы написанные по заданным в задании алгоритмам

### **Контрольные вопросы**

1. Какие способы взаимодействия имеет микроконтроллера AVR с ЖКИ модулем на базе контроллера HD44780
2. Сколько строк имеет ЖКИ модуль на базе контроллера HD44780?
3. Сколько символов в строке воспроизводит ЖКИ модуль на базе контроллера HD44780?
4. Какие регистры имеет ЖКИ модуль на базе контроллера HD44780 для управления ?
5. Какое количество символов и каким образом воспроизводит ЖКИ модуль на базе контроллера HD44780?
6. Каким сигналом синхронизируется запись кода символа в ЖКИ модуль на базе контроллера HD44780?
7. Каким сигналом синхронизируется запись кода команды в ЖКИ модуль на базе контроллера HD44780?

### **Список литературы, рекомендуемый к использованию по данной теме:**

1. Клингман Э. Проектирование микропроцессорных систем. М.: Мир. 1988. - 575с.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.
3. Трамперт В. AVR-RISK микроконтроллеры.: Пер. с нем. – К.: «МК-ПРЕСС», 2006.-464с., ил.

4. 4. Ю.А.Шпак. Программирование на языке С для AVR и PIC микроконтроллеров. – К.: «МК-ПРЕСС», 2006.-400с., ил.
5. Якубовский С.В. Цифровые и аналоговые интегральные микросхемы. М.:Радио и связь. 1990.
6. Токхейм Р. Основы цифровой электроники. Пер. с англ. - М.: Мир, 1988. – 390 с.
7. Шило В.Л. Популярныe цифровые микросхемы. – М.: Радио и связь, 1990.– 350 с.
8. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах - М.: Радио и связь. 1992 (1996).
9. Опадчий Ю.Ф., Глудкин О.П., Гуров А.И. Аналоговая и цифровая электроника. Полный курс: учебник для вузов. - М.: Горячая линия, 1999 (2000, 2005). – 768 с.
10. Алексенко А.В., Шагуров И.И. Микросхемотехника.– М.: Радио и связь,1990 (1982).