Лабораторная работа № 1

# Программирование микроконтроллеров AVR

Цели работы:

Ознакомление с лабораторным макетом MK8535 и средой програм- мирования микроконтроллеров CodeVisionAVR.

1. Изучение этапов программирования простейшей программы.

микроконтроллера на примере

Оборудование и программное обеспечение:

1. Лабораторный макет MK8535 в составе: модуль базовый МБ8535 +

модуль сменный МС01.

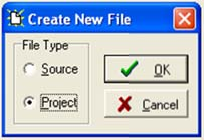
1. Кабель соединительный для программирования МК.
2. Персональный компьютер.
3. Интегрированная среда программирования CodeVisionAVR.

В ходе лабораторной работы будет подготовлена простейшая программа для МК ATmega8535 и посредством программатора записана в память программ МК.

# Подготовка проекта и файла исходного кода

Разработка программы начинается с создания проекта. Запустить на компьютере приложение CodeVisionAVR.

В меню **File** → **New** → **Create New File** выбрать **Project** (рис. 10, а).



а) б)

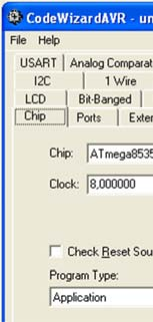
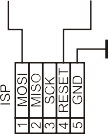
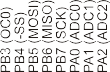
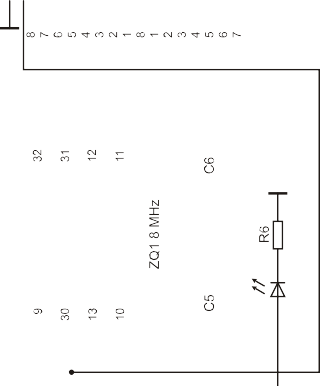
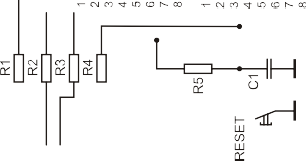
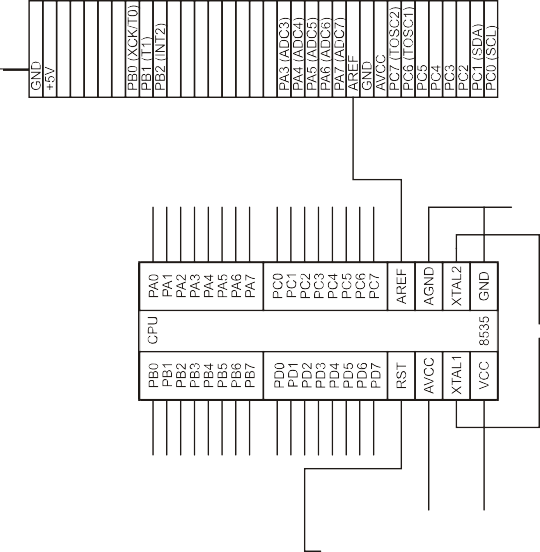
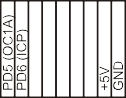


Рис. 10.

Старт проекта (а), выбор модели МК и сохранение файлов (б)

В открывшемся окне (рис. 10, б) на вкладке **Chip** выбрать модель



ATmega8535, частота – 8 Мгц, тип программы – Application.

Далее, в окне **CodeWizardAVR** (рис. 10, б) выбрать **File** → **Generate, Save and Exit** и сохранить файлы проекта под именами lab\_1.c, lab\_1.prj и lab\_1.cwp. После этого появится окно с текстом на языке Си – это и есть исходный текст программы, точнее, ее начальный вариант, со- держащий подготовительные операции для МК (см. рис. 8). Исходный текст хранится в файле lab\_1.c.

Рис. 11. Структурная схема лабораторного макета.

Для модуля МС-01 показана только та часть схемы, которая используется в данной работе

Теперь дополним эту программу нужными нам действиями. Наша программа будет осуществлять периодическое включение/выключение светодиодов, подключенных к порту A через ограничительные резисторы R1…R8 (рис. 11). Нагрузочная способность портов позволяет подклю- чать к ним светодиоды напрямую. В этой работе нам нужен только порт A, настроенный на вывод данных. В модуле МС01 светодиоды будут све- титься, когда на выходах порта A установлена 1. Длительность включе- ния светодиодов сделаем равной 300 мсек, пауза – 300 мсек. Чтобы вос- пользоваться процедурой временной задержки delay\_ms(), необходимо включить заголовочный файл delay.h.

Изменим начальный текст программы, созданный **CodeWizardAVR,**

так чтобы он принял следующий вид:

# #include <mega8535.h> #include <delay.h>

**void main (void)**

**{**

**DDRA = 0xff; // Настроить все линии**

**// порта A на вывод**

**while (1) // Бесконечный цикл**

**{**

**PORTA = 0xff; // Включить все светодиоды**

**delay\_ms(300); // Задержка 300 мсек**

**PORTA = 0x00; // Выключить все светодиоды**

**delay\_ms(300); // Задержка 300 мсек**

**}**

**}**

# Компиляция программы и запись в микроконтроллер

Исходный текст программы теперь нужно транслировать в ма- шинные коды модели ATmega8535, а полученную программу передать в память программ микроконтроллера. Для этой цели в CodeVisionAVR имеются компилятор языка Си, автоматический генератор машинных кодов и внутрисистемный программатор.

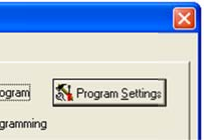
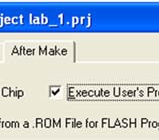
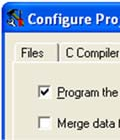
Сначала укажем компилятору модель микроконтроллера. В меню

**Project** выбрать **Configure** (рис. 12). В открывшемся окне **Configure**

для того, чтобы после компиляции разрешить работу и запуск программы в микроконтроллере.

программатора

**Project** на вкладке **C Compiler** выбрать модель чипа



ATmega8535,

Рис. 13. Разрешение программирования чипа

частота 8 МГц, тип программы – Application.

Теперь настроим программатор. В меню **Settings** выберем **Programmer**. В открывшемся окне **Programmer Settings** (рис. 14) выбрать тип программатора **STK200+/300**, порт принтера (как правило, **LPT1: 378h**).

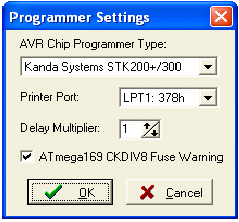
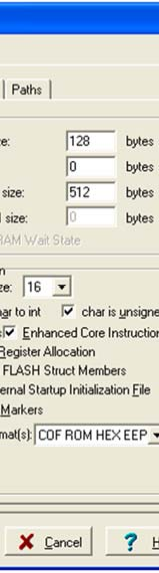
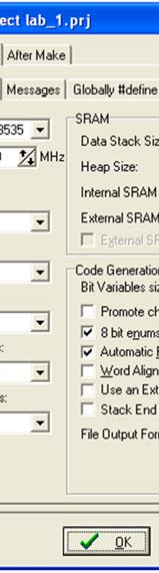
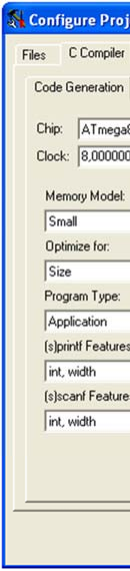


Рис. 14. Выбор типа программатора

В меню **Tools** выбрать **Chip Programmer** (рис. 15). В открывшемся

окне



# CodeVisionAVR

**Chip Programmer** выбрать чип

ATmega8535,

частота 8 МГц.

Рис. 12. Настройка компилятора

В том же окне перейдем на вкладку **After Make** (рис. 13) и вклю- чим флажки **Program the Chip** и **Execute User’s Program**. Это нужно

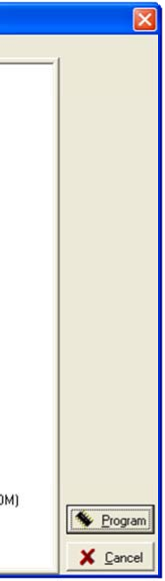
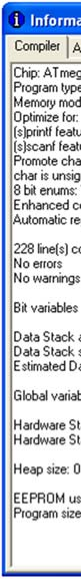


Рис. 15. Настройка программатора

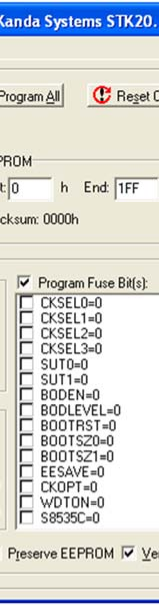
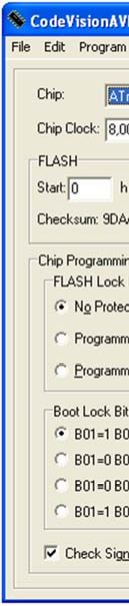


Рис. 16. Сводка результатов компиляции

# Задание

Наконец, компиляция проекта и программирование микро-

контроллера. В меню **Project** выбрать **Make**. Начнется компиляция про- екта, по завершении которой откроется окно **Information** со сводкой результатов компиляции (рис. 16). Если в программе имеются синтакси- ческие ошибки, то компиляция не завершится успешно. Нужно устра-

1. Создать проект и реализовать рассмотренный выше пример программы.
2. Изменяя длительность временной задержки, получить иные скорости переключения светодиодов.
3. Изменить код программы так, чтобы переключались только:

нить ошибки и запустить компиляцию заново.

Если компиляция завершилась успешно, кнопкой **Program** запу-

* + 4 младших светодиода;
  + 4 старших светодиода;

стим программирование микроконтроллера. По окончании процесса программирования в микроконтроллере начнет работать наша программа.

* + четные светодиоды;
  + нечетные светодиоды;
  + один старший светодиод;
  + один младший светодиод.

# ПОРТЫ ВВОДА-ВЫВОДА МИКРОКОНТРОЛЛЕРОВ AVR

**Общие сведения**

Каждый порт микроконтроллера состоит из определенного числа выводов, через которые микроконтроллер может осуществлять прием и передачу цифровых сигналов. Задание направления передачи данных через любой контакт ввода/вывода может быть произведено программно в любой момент времени.

Выходные буферы всех портов, имея симметричные нагрузочные характеристики, обеспечивают высокую нагрузочную способность при любом уровне сигнала. Нагрузочной способности, например, достаточно для непосредственного управления светодиодными индикаторами.

Входные буферы всех выводов построены по схеме триггера Шмитта. Для всех входов имеется возможность подключения внутрен- него подтягивающего резистора между входом и шиной питания *Vcc.*

Отличительной особенностью портов микроконтроллеров семей- ства Mega (как и всех микроконтроллеров AVR) при использовании их в качестве цифровых портов ввода/вывода общего назначения является реализация истинной функциональности вида «чтение-модификация- запись». Благодаря этому можно выполнять операции над любым выво- дом (с помощью команд SBI и CBI), не влияя на другие выводы порта. Это относится к изменению режима работы контакта ввода/вывода, к изменению состояния выходного буфера (для выходов) и к изменению состояния внутреннего подтягиваюшего резистора (для входов).

Микроконтроллеры различных моделей семейства имеют различ- ное количество портов и соответственно контактов ввода/вывода. Например, микроконтроллер ATmega8535 имеет четыре 8-разрядных порта ввода/вывода.

# Регистры портов ввода/вывода

Обращение к портам производится через регистры ввода/вывода. Под каждый порт в адресном пространстве ввода/вывода зарезервирова- но по 3 адреса, по которым размещены следующие регистры:

**PORT*x*** – регистр данных порта (Data Register);

**DDR*x*** – регистр направления данных (Data Direction Register);

**PIN*x*** – регистр выводов порта (Port Input Pins).

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 PIN*x* осуществляется доступ к физиче- ским значениям сигналов на выводах порта, они доступны только для чтения, тогда как остальные два регистра доступны и для чтения, и для записи.

Для примера, в табл. 2 приведены адреса регистров портов вво- да/вывода микроконтроллера ATmega8535.

*Таблица 2*

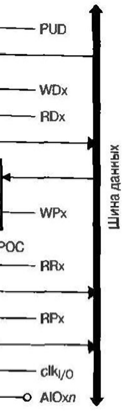
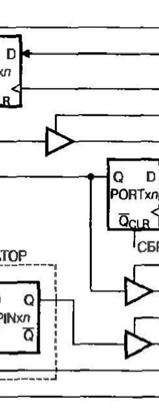
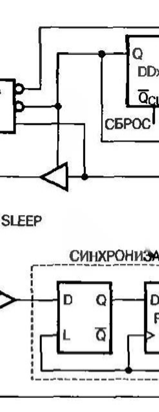
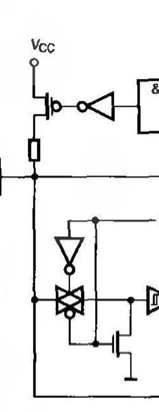
Регистры портов ввода/вывода микроконтроллера ATmega8535

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Порт A | | Порт B | | Порт C | | Порт D | |
| Регистр | Адрес | Регистр | Адрес | Регистр | Адрес | Регистр | Адрес |
| PORTA | 0x1B | PORTB | 0x18 | PORTC | 0x15 | PORTD | 0x12 |
| DDRA | 0x1A | DDRB | 0x17 | DDRC | 0x14 | DDRD | 0x11 |
| PINA | 0x19 | PINB | 0x16 | PINC | 0x13 | PIND | 0x10 |

# Конфигурирование портов ввода/вывода

Перед использованием выводы портов микроконтроллера необ- ходимо конфигурировать, т.е. задать нужный режим работы (на ввод, на вывод и т.д). Любой из выводов порта можно конфигурировать незави- симо от остальных. Структурная схема одного из каналов порта вво- да/вывода Р*xn* при работе его в качестве цифрового входа/выхода обще- го назначения приведена на рис. 17.

Каждому выводу порта соответствуют три разряда регистров вво- да/вывода: PORT*xn* (регистр PORT*x*), DD*xn* (регистр DDR*x*) и PIN*xn* (peгистр PIN*x*). Действительные названия разрядов регистров получаются подстановкой названия порта вместо символа «*x*» и номера разряда вме- сто символа «*n*». Порядковый номер вывода порта соответствует поряд- ковому номеру разряда регистров этого порта. Поэтому, если разрядность порта меньше восьми, в регистрах порта используется соответствующее число младших разрядов. Незадействованные старшие разряды регистров доступны только для чтения и всегда содержат 0.



PUD: выключение подт. рези- RPx: чтение выводов порта

Если же вывод функционирует как вход (DD*xn* = 0), разряд PORT*xn* определяет состояние внутреннего подтягивающего резистора для данного вывода. При установке разряда PORT*xn* в 1 подтягивающий резистор подключается между выводом микроконтроллера и шиной питания *Vcc*.

Вообще говоря, управление подтягивающими резисторами почти во всех микроконтроллерах семейства, за исключением моделей ATmega161x, ocyществляется на двух уровнях. Общее управление (для всех выводов портов) осуществляется разрядом PUD (2-й разряд) реги- стра специальных функций SFIOR.

Если разряд PUD сброшен в 0 (начальное состояние), состояние подтягивающих резисторов будет определяться состоянием разрядов PORT*xn* для каждого входа порта. Если же разряд PUD установлен в 1, подтягивающие резисторы отключаются от всех выводов микро- контроллера.

Обратите внимание, что при переключении вывода между третьим состоянием (DD*xn =* 0, РORT*xn* = 0) и состоянием высокого уровня (DD*xn* = 1, PORT*xn* = 1) происходит переход через одно из промежуточ-

стора

PORTx

ных состояний: либо включается подтягивающий резистор (DD*xn* = 0,

WDx: запись регистра DDRx SLEEP: управление спящим режи-

мом

RDx: чтение регистра DDRx clkI/O: тактовый сигнал подсистемы

вв/выв

PORT*xn* = 1), либо выход переключается в состояние низкого уровня (DD*xn* = 1, PORT*xn* = 0). Наиболее применимым является, как правило, первый вариант, поскольку для высокоимпедансных систем безразлично, каким образом формируется высокий уровень. Если в каком-либо случае

WPx: запись регистра PORTx AIO*xn*: аналоговый

вход/выход

это не подходит, можно отключить подтягивающие резисторы от всех

RRx: чтение регистра PORTx

вывода *n* порта PORTx

портов установкой в 1 разряда PUD регистра SFIOR.

Аналогичная ситуация возникает и при переключении между

Сигналы WPx, WDx, RPx, RDx являются общими для всех выводов одного порта.

Cигналы clkI/O, SLEEP и PUD являются общими для всех портов микро- контроллера.

Рис. 17. Структ рная схема одного канала порта ввода/вывода

у

Разряд DD*xn* регистра DD*x* оп еделяет направление передачи данных через контакт ввода/вывода. Если этот разряд установлен в 1, то *n*-й вывод порта является выходом, если же сброшен в 0 – входом.

р

Разряд PORT*xn* регистра PORT*x* выполняет двойную функцию.

состоянием с включенным подтягивающим резистором (DD*xn* = 0, PORT*xn* = 1) и состоянием низкого уровня (DD*xn* = 1, PORT*xn* = 0). В этом случае промежуточным состоянием является либо высокоимпе- дансное состояние (DD*xn* = 0, PORT*xn* = 0), либо состояние высокого уровня (DD*xn* = 1, PORT*xn* = 1).

Все возможные сочетания состояний управляющих разрядов и соответственно конфигурации выводов портов сведены в табл. 3.

Если вывод функционирует как выход ( D*xn* = 1), этот разряд определяет

D

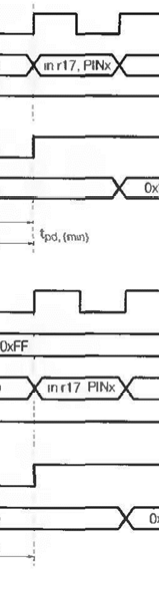
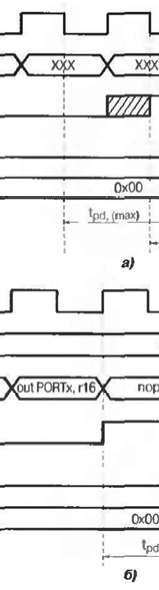
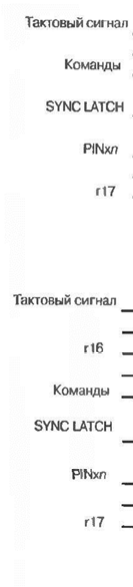
состояние вывода порта. Если разряд установлен в 1, на выводе устанав- лива тся напряжение высокого уровня. Если разряд сброшен в 0, на вы-

е

воде устанавливается апряжение низкого уровня.

н

*Таблица 3*



Конфигурации выводов портов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DD*xn* | PORT*xn* | PUD  (B SFIOR) | Функция вывода | Резистор | Примечания |
| 0 | 0 | х | Вход | Отключен | Третье состояние (Hi-Z) |
| 0 | 1 | 0 | Вход | Подключен | При подключении нагруз- ки между выводом и об- щим проводом, вывод является источником тока |
| 0 | 1 | 1 | Вход | Отключен | Третье состояние (Hi-Z) |
| 1 | 0 | х | Выход | Отключен | Выход установлен в 0 |
| 1 | 1 | х | Выход | Отключен | Выход установлен в 1 |

Состояние вывода микроконтроллера (независимо от установок разряда DD*xn*) может быть получено путем чтения разряда PIN*xn* реги- стра PIN*x*. При этом следует помнить, что между действительным изме- нением сигнала на выводе и изменением разряда PIN*xn* существует за- держка. Эта задержка вносится узлом синхронизации, состоящего, как показано на рис. 17, из разряда PIN*xn* и дополнительного триггера- защелки. Значение сигнала на выводе микроконтроллера фиксируется триггером-защелкой при низком уровне тактового сигнала и переписы- вается затем в разряд PIN*xn* по нарастающему фронту тактового сигнала. Соответственно величина задержки может составлять от 0,5 до 1,5 пери- ода системного тактового сигнала, как показано на рис. 18, 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 рав- на одному периоду тактового сигнала (рис. 18, б).

Рис. 18. Синхронизация при чтении состояния вывода:

а – при считывании состояния разряда PIN*xn*, б – при повторном считывании

Следует отметить, что контакты портов микроконтроллера ATmega8535 многофункциональны и кроме ввода-вывода цифровых сигналов используются для работы с аналоговыми сигналами (входы АЦП, компаратора).