ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ

УТВЕРЖДАЮ
Зам директора по
Учебно-производственной
практике
_____Пронин Г.М.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРАКТИКЕ ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ НА ЯЗЫКЕ СИ

для специальностей:

230103 Автоматизированные системы
обработки информации и управления (по отраслям)

230101 Вычислительные машины, комплексы, системы и сети

Автор Логинов Д.В.
Рецензент Малафеева Н.Б.

Рассмотрено
На заседании предметной комиссии
Специальности 230101
Протокол №_____
От «___»_____ 2007 г.

Нижний Новгород
2007 г.

Содержание

Раздел 1
Знакомство с WinAVR

Теоретическая часть

WinAVR – пакет программ, позволяющий программировать микроконтроллеры на языке высокого уровня С.

Рассмотрим преимущества использования С при написании программ.

Микроконтроллеры с RISC-архитектурой позволяют выполнять только простейшие команды, а более сложные, такие как деление, производятся с использованием определенных комбинаций простых команд. При программировании на Ассемблере это неудобно, зато при использовании языка высокого уровня данную проблему решает компилятор, освобождая программиста от написания стандартных подпрограмм.

Кроме того программы, написанные на языке высокого уровня, легко переносятся на другие микроконтроллеры, а приобретённые навыки при программировании одного типа микроконтроллеров могут найти применение и для других типов.

К недостаткам использования языка высокого уровня можно отнести неоптимальность программного кода и его большой размер. Объем памяти программ в микроконтроллерах небольшой, и в него может не поместиться сложная программа.

WinAVR позволяет преобразовать программу, написанную на языке С в программный код для микроконтроллера, который представляет собой файл с расширением hex. С помощью AVR Studio, используя этот файл, можно эмулировать работу микроконтроллера, а при наличии программатора можно ввести эту программу непосредственно в сам микроконтроллер.

Этапы разработки программы:

1. С помощью утилиты MFile[WinAVR] создается файл, содержащий параметры проекта (тип микроконтроллера, название файла программы, и др.)
2. С помощью Programmers Notepad [WinAVR] создается программа на языке С.
3. Производится компиляция программы (создается hex-файл)
4. С помощью AVR Studio выполняется эмуляция работы микроконтроллера.
5. Подключается программатор и программа вводится в микроконтроллер

Указание.

Описания операторов языка С, необходимых для выполнения лабораторных работ, приведены в приложении.

Практическая часть

1. Создание проекта

1.1. **Запустите утилиту MFile[WinAVR].**

1.2. Через меню Makefile настройте параметры:

1.2.1. Используя пункт Main file name **введите название файла** (Например, main). Запомните это название! Файл программы на языке С нужно будет назвать также.

1.2.2. В пункте MCU type **выберите микроконтроллер ATMega8515**.

1.2.3. Остальные пункты не изменяйте.

1.3. **Создайте новую папку и сохраните в неё файл настроек** с именем Makefile через меню File\Save As…
1.4. **Закройте утилиту MFile**.

2. Разработка программы на языке С
2.1. **Запустите программу Programmers Notepad** [WinAVR].
2.2. **Сохраните с расширением ".c"** в ту же папку, где находится файл с настройками. Имя файла должно соответствовать названию проекта, введенному в пункте 1.2.1 и иметь расширение .c (Если название проекта было main, то данный файл нужно сохранить с именем main.c)
2.3. **Введите текст программы**

```
#include <avr/io.h>  // Подключение библиотеки ввода-вывода
#include <stdio.h>

int main (void)  //Начало основной программы
{
char n;          //Объявляем переменную типа char
DDRB=0xFF;  //В регистр направления порта В записываем число FF,
             //означающее, что порт В будет использоваться для вывода
 for (n=0;n<10;n++) //Организовываем цикл от 0 до 10
 PORTB = n;          // Выводим в порт В текущее значение n (в цикле)
}
```

2.4. **Откомпилируйте программу**, нажав в меню Tools пункт Make All.
При отсутствии ошибок синтаксиса в нижнем окне будет сообщение

```
-------- end --------
> Process Exit Code: 0
```

После компиляции будет создан файл с расширением hex.
3. Эмуляция работы микроконтроллера в AVRStudio
3.1. **Запустите приложение AVRStudio** (Пуск\Программы\Atmel AVR tools\AVR Studio4)
3.2.**Нажмите OPEN и откройте файл с расширением hex** из папки проекта.
3.3.**Выберите Микроконтроллер ATMEGA8515**

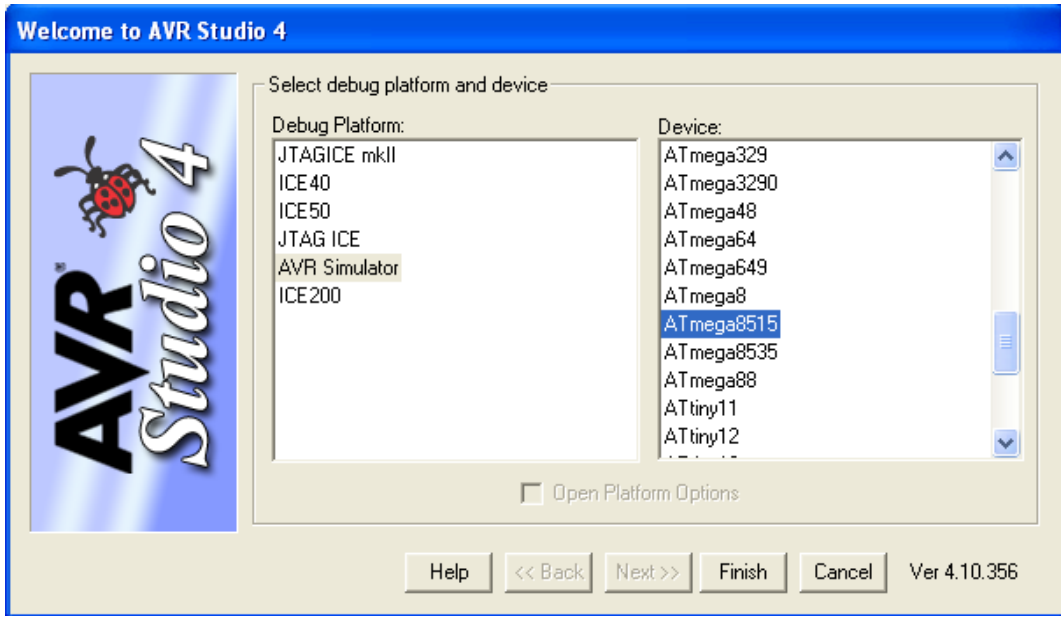


Рис. 1. Выбор микроконтроллера

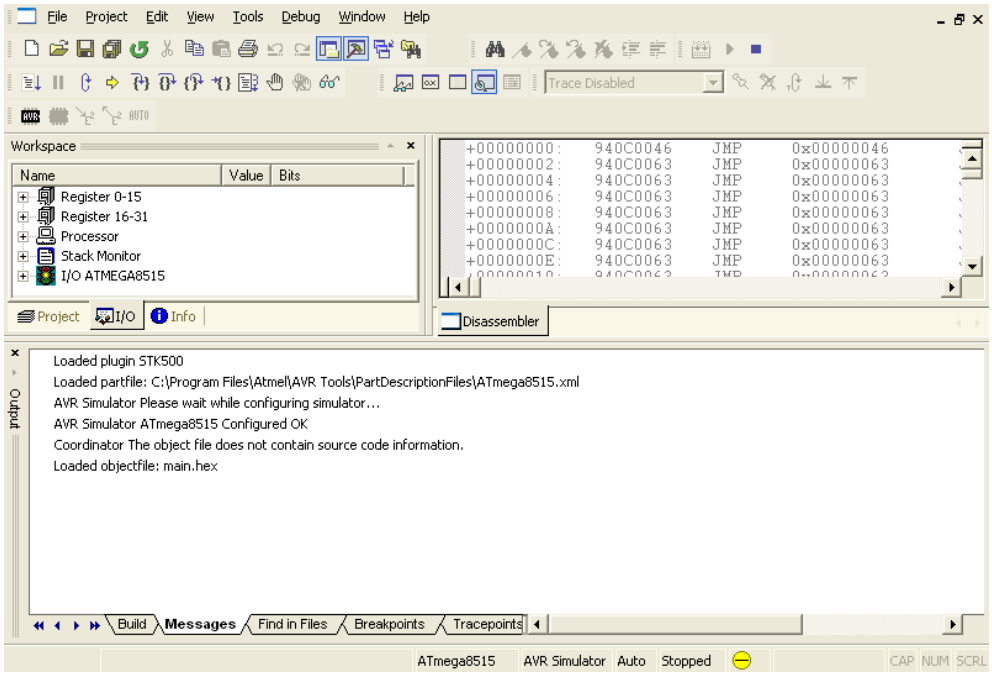


Рис 2. Окно программы AVR Studio после открытия файла

Внимание: При повторном открытии файла окно Disassembler с содержанием программы (справа) может исчезнуть. Для его появления нажмите в меню пункт View\ Disassembler.

3.4.**В окне WorkSpase выберите порт B** (он находится в пункте I/O ATmega8515):

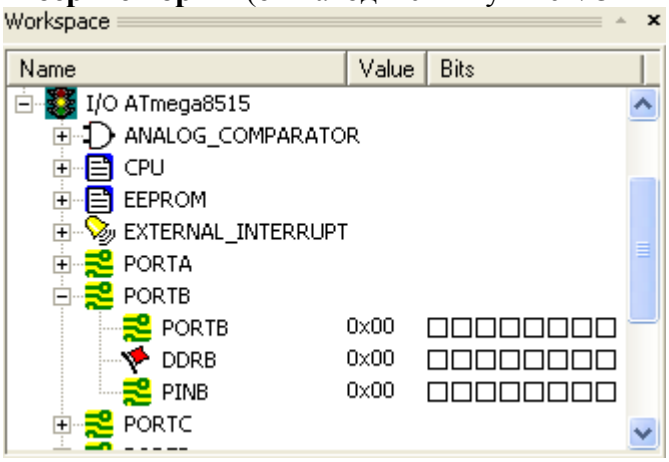


Рис 3. Окно Workspace

3.5.**Запустите пошаговое выполнение программы**, нажимая кнопку . Примерно после тридцатого нажатия в регистр DDRB будет занесено значение FF, показывающее, что порт запрограммирован на вывод.

Рис 4. Состояние порта D после программирования регистра DDRB

3.6. **Продолжайте пошаговое выполнение программы до конца**, пока в регистр PORTD не запишется число 9

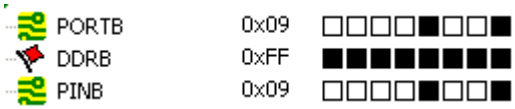


Рис 5. Состояние порта D после выполнения программы

При пошаговом выполнении видно, что программа работает верно, записывая в PORTD числа от 0 до 9, как было задано в программе на языке С.

4. Создание простейших программ на языке С.

4.1. Создайте новый проект (см. пункты 1.1-2.1). **Напишите** на С **программу, которая берет числа из портов А и В** (из регистров PINA и PINB)**, складывает их и записывает результат в PORTC.** Создайте hex-файл и проверьте его правильное выполнение.

Примечание: перед пошаговым выполнением введите с помощью мыши в порты А и В числа 4F и 6A (см рис 6 и рис 7).

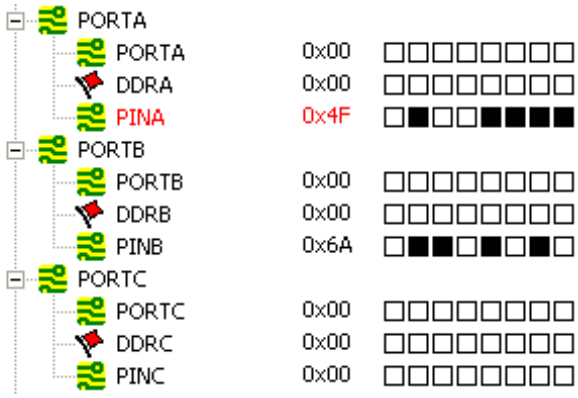


Рис 6. Состояние портов до выполнения программы

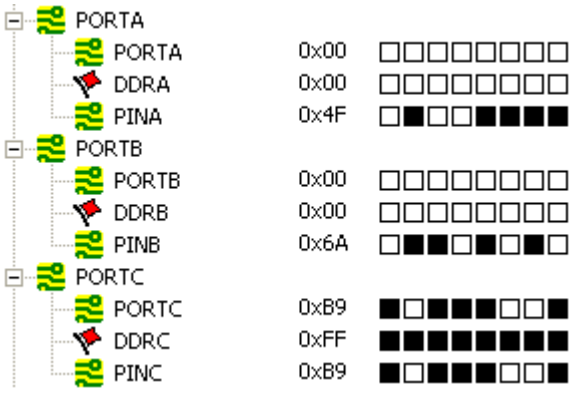
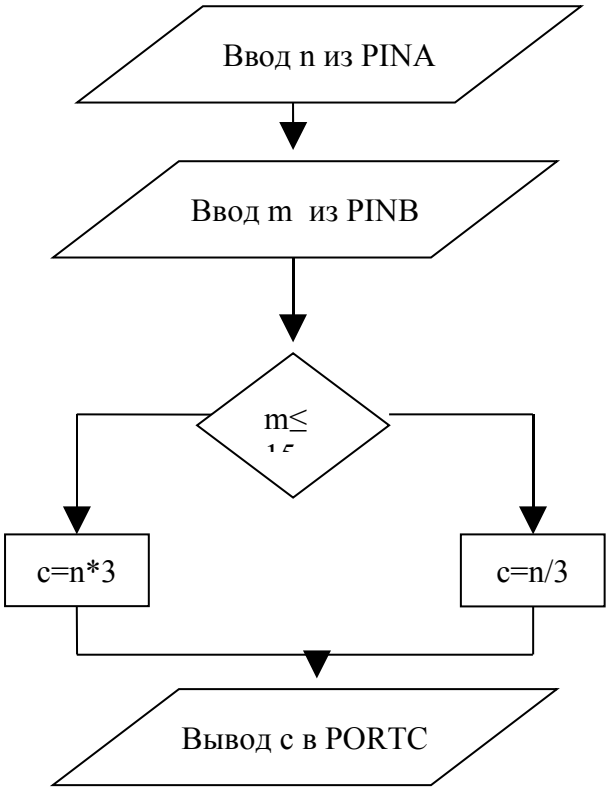


Рис 7. Состояние портов после выполнения программы

4.2. В окне Disassembler в первом столбике указывается адрес команды (номер ячейки в памяти программ). **Определите адрес последней команды, объём памяти, занимаемый программой и объем всей доступной памяти программ** в микроконтроллере, используя столбик с адресами в окне Disassembler.

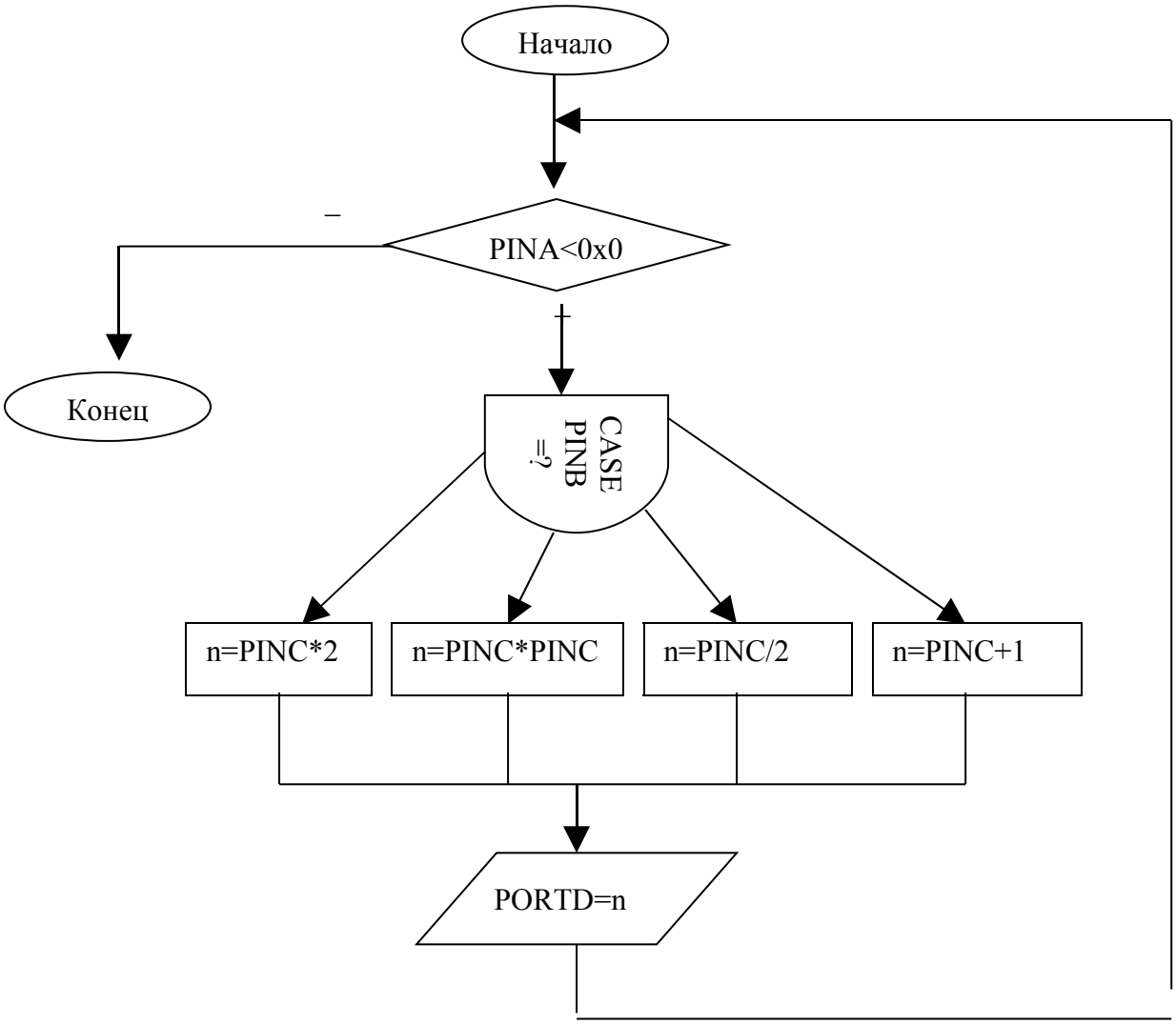
4.3. **Напишите программу**, которая бы соответствовала следующей блок-схеме (Описание оператора if см. в приложении):



4.4. **Проверьте правильность выполнения** написанной программы.

4.5. **Программу, написанную в задании 7 перекомпилируйте таким образом, чтобы она могла работать на микроконтроллере ATmega16**. Для этого необходимо изменить тип микроконтроллера в файле параметров Makefile с помощью утилиты MFile[WinAVR], перекомпилировать программу в **Programmers Notepad** [WinAVR], а также при открытии hex-файла в AVR Studio необходимо выбрать эмулятор ATmega16.

4.6. Разработайте программу, работающую по приведенному ниже алгоритму. Проверьте правильность её работы.



Контрольные вопросы

1. Какие преимущества имеет программирование на языке высокого уровня?
2. Какие недостатки имеет программирование на языке высокого уровня?
3. Что хранит hex-файл?
4. Каково назначение программных комплексов WinAVR и AVR Studio?
5. Зачем нужен Disassembler?