

Лабораторная работа №11

Тема: Знакомство с программой Atmel Studio 6.

Цель работы: Ознакомление с интегрированной средой разработки Atmel Studio 6 для программирования микроконтроллеров AVR.

План работы:

1. Установка Atmel Studio 6.
2. Создание нового проекта.
3. Изучение интерфейса Atmel Studio 6.
4. Краткий обзор портов ATmega8.
5. Написание простой программы на C для ATmega8.
6. Сборка и загрузка проекта в Proteus.

Краткая теория.

Atmel Studio 6 - это интегрированная среда разработки (IDE) от компании **Atmel**, предназначенная для программирования микроконтроллеров **AVR** и **ARM**.

Atmel Studio 6 предоставляет инструменты для разработки программного обеспечения для микроконтроллеров AVR и ARM, включая популярные модели, такие как **ATmega** и **SAM**.

Поддерживает программирование на языке **C**, **C++**, и **ассемблере**, что позволяет разработчикам выбирать подходящий язык в зависимости от задач и предпочтений.

Предоставляет **возможности отладки**, такие как точки останова (breakpoints), отслеживание переменных, профилирование кода, что облегчает выявление и исправление ошибок.

Позволяет симулировать код без необходимости физического подключения микроконтроллера, что упрощает отладку программ.

Поддерживает проекты, созданные с использованием платформы **Arduino**, позволяя программировать Arduino в Atmel Studio.

Предоставляет стандартные **библиотеки** для взаимодействия с периферийными устройствами микроконтроллеров.

Atmel Studio 6 предоставляется **бесплатно**, что делает его доступным для широкого круга разработчиков.

ATmega8 - это 8-битный микроконтроллер, созданный фирмой Atmel. Краткое описание:

- | | |
|----|---|
| 1. | Архитектура: <ul style="list-style-type: none">• Он основан на улучшенной архитектуре AVR RISC, обеспечивающей высокую производительность при низком энергопотреблении. |
| 2. | Рабочая частота: <ul style="list-style-type: none">• Рабочая частота ATmega8 может достигать до 16 МГц, что обеспечивает достаточную производительность для различных приложений. |
| 3. | Память: <ul style="list-style-type: none">• Оборудован 8 КБ флэш-памяти для программ, 1 КБ оперативной памяти SRAM и 512 байт энергонезависимой EEPROM для хранения данных. |
| 4. | Периферийные устройства: <ul style="list-style-type: none">• Обладает разнообразными периферийными устройствами, включая аналогово-цифровой преобразователь (ADC), таймеры/счетчики, UART и другие. |
| 5. | GPIO и порты ввода-вывода: <ul style="list-style-type: none">• Предоставляет обширные возможности для ввода-вывода, с 23 программируемыми GPIO-пинами. |
| 6. | Коммуникационные интерфейсы: <ul style="list-style-type: none">• Поддерживает стандартные коммуникационные интерфейсы, такие как SPI, I2C, и UART, что делает его удобным для взаимодействия с другими устройствами. |
| 7. | Напряжение питания: <ul style="list-style-type: none">• Работает от напряжения питания от 2.7 В до 5.5 В, что обеспечивает гибкость в выборе источника питания. |
| 8. | Программирование: |

- Может быть программирован с использованием различных сред разработки, таких как Atmel Studio, и поддерживает программирование как в системе, так и через программатор.

9. Применения:

- Широко используется в различных электронных устройствах, включая промышленные контроллеры, датчики, игрушки и прочее, благодаря своей надежности и функциональности.

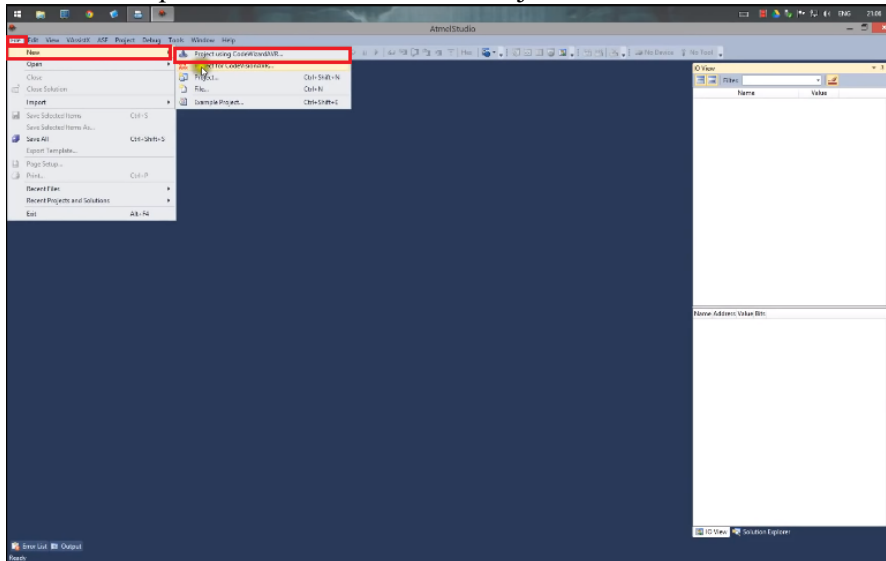
10. Бюджетность:

- АТМega8 является бюджетным и распространенным микроконтроллером, доступным для широкого круга разработчиков.

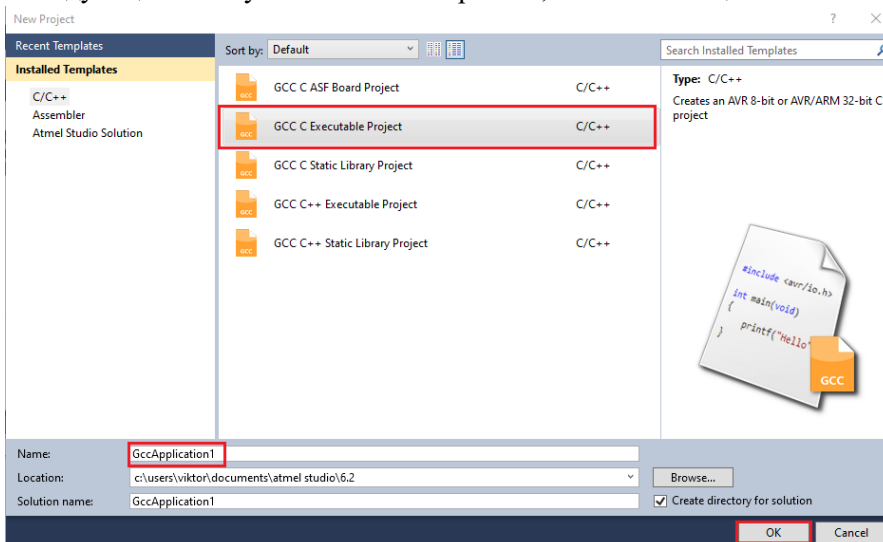
Практика.

Создание проекта

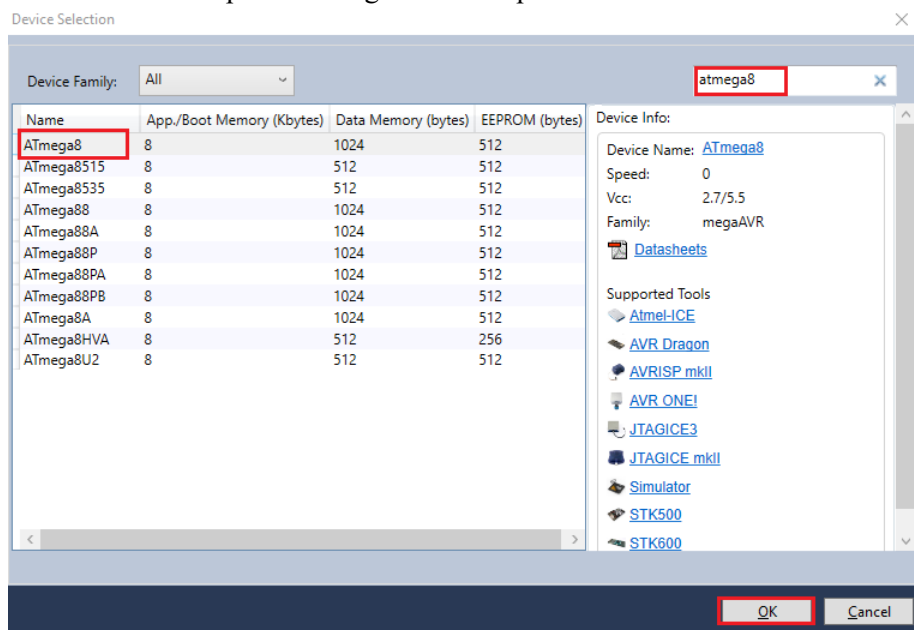
В меню выбрать File => New => New Project



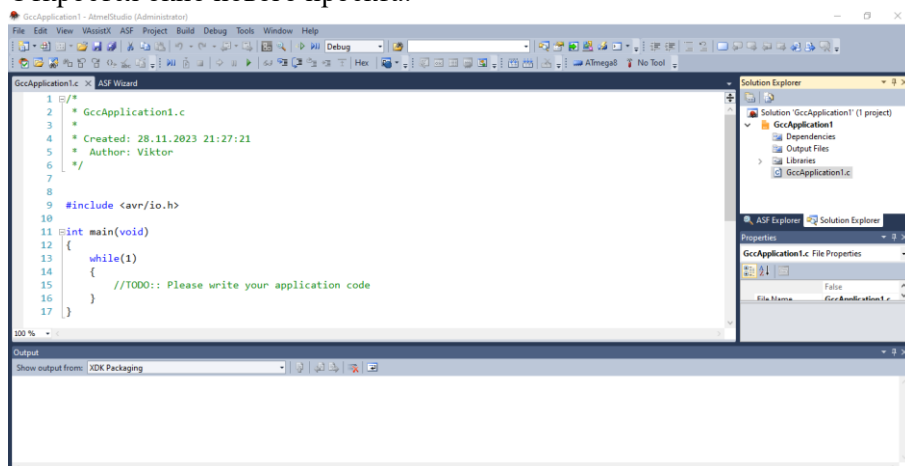
В следующем окне указываем **имя** проекта, тип компиляции: **GCC C Executable Project** и нажимаем **ОК**.



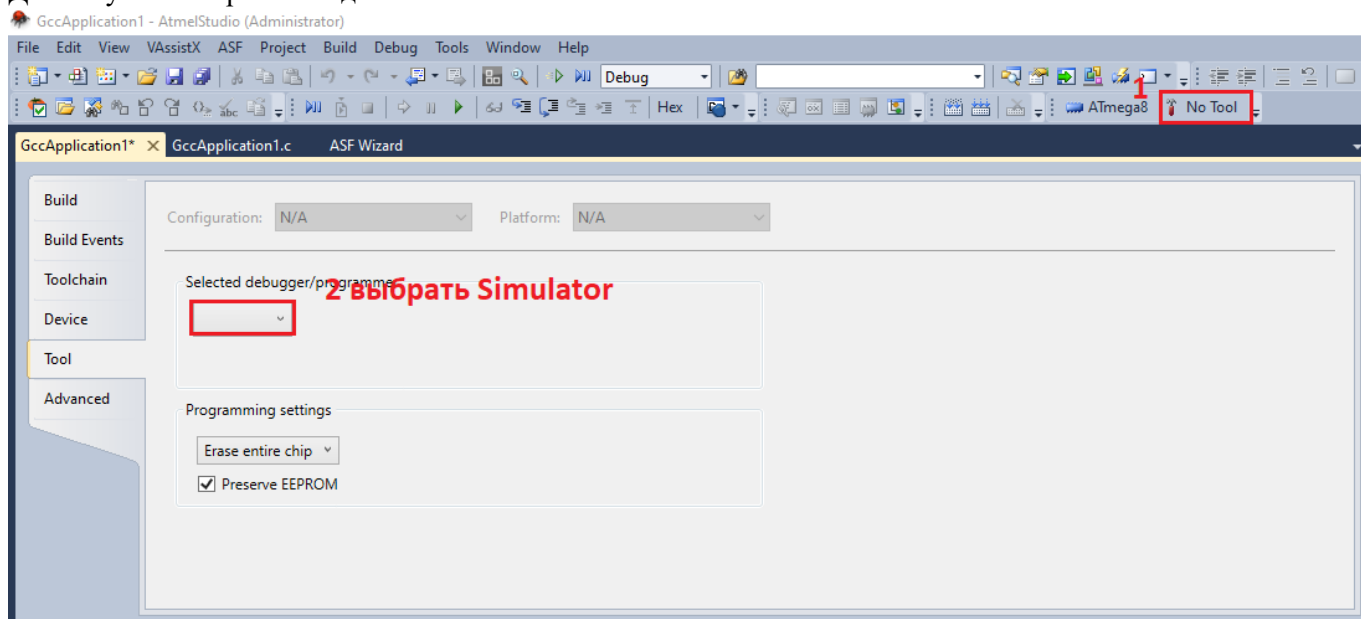
откроется окно **Device Selection**, где из обширного списка нужно выбрать конкретный микроконтроллер. В окне поиска набираем «atmega8» и выбираем его из списка:



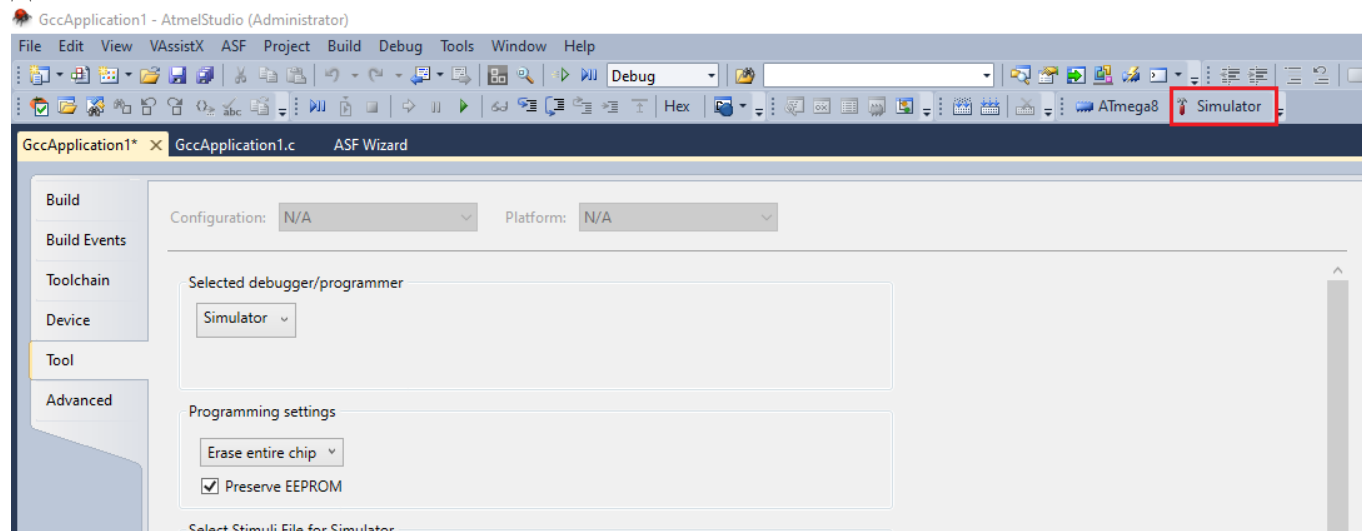
Откроется окно нового проекта:



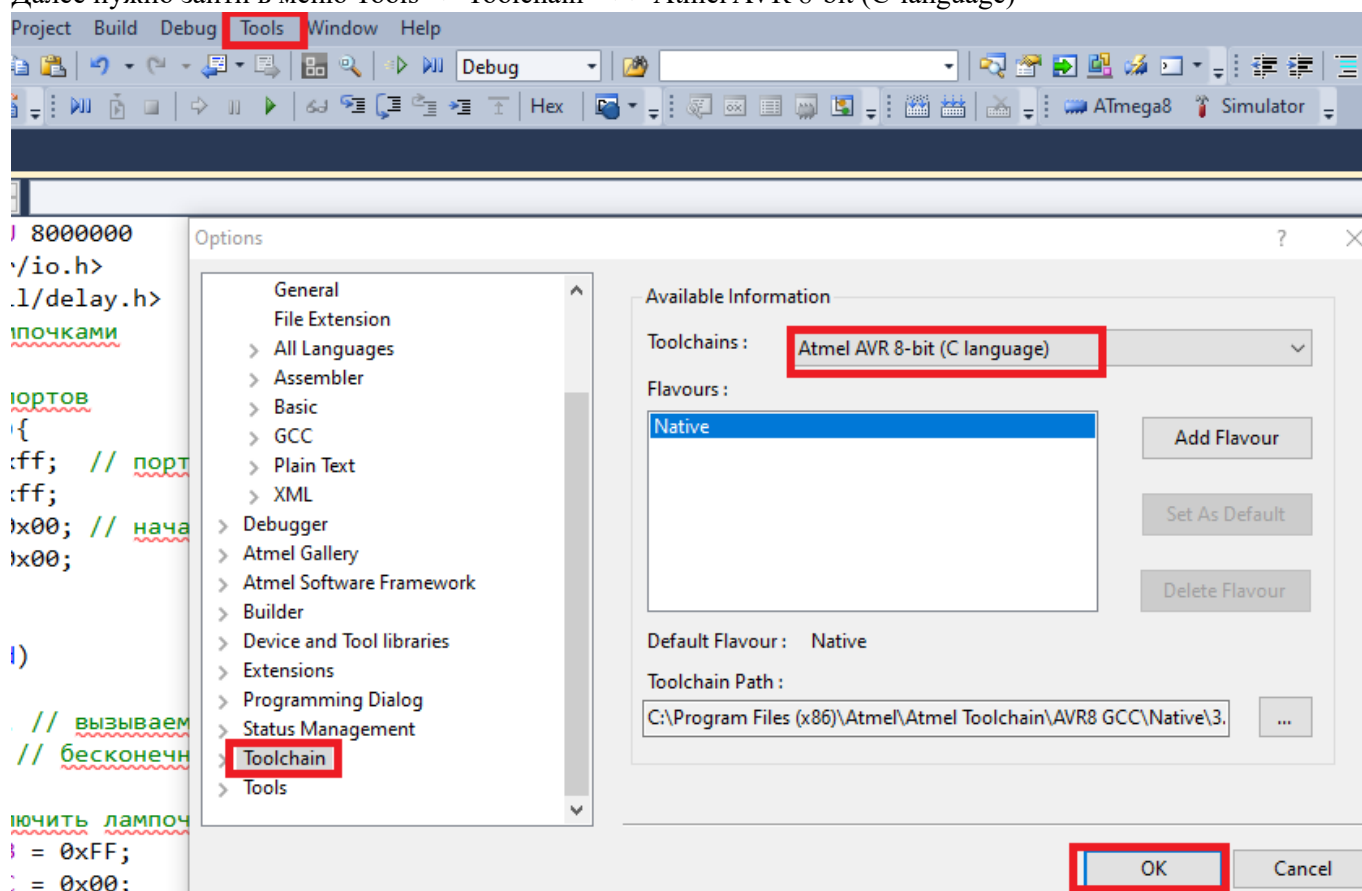
Далее нужно выбрать отладчик:



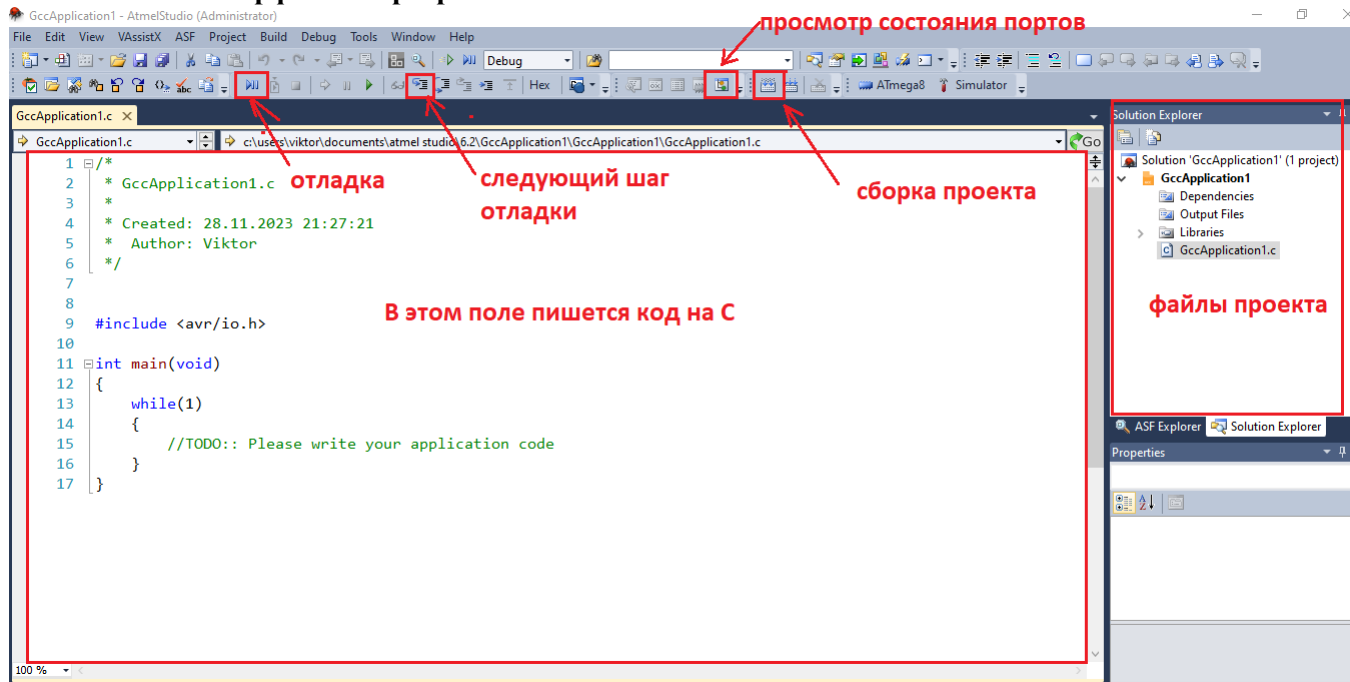
Должно стать так:



Далее нужно зайти в меню Tools => Toolchain ==> Atmel AVR 8-bit (C-language)



Знакомство с интерфейсом программы.



Справка по системам счисления. В проекте используются 10-я, 2-я и 16-я системы счисления.

Число в 10-й системе	Число в 2-й системе	Число в 16-й системе
0	0b00000000	0x00
1	0b00000001	0x01
10	0b00001010	0x0A
255	0b11111111	0xFF

Порты микроконтроллера ATmega8

Микроконтроллер ATmega8 имеет различные порты ввода-вывода (GPIO), которые предоставляют возможность подключения и управления внешними устройствами. Вот краткое описание портов микроконтроллера ATmega8:

1. Порты ввода/вывода (GPIO):

- АТmega8 оборудован 3 портами ввода-вывода: **PORTA, PORTB и PORTC**. Каждый порт может быть использован для ввода или вывода данных.

2. Регистры данных (DATA REGISTERS):

- Для каждого порта существует регистр данных (PORTx), который используется для записи данных, а также регистр ввода (PINx), через который можно читать данные с порта.

3. Регистры направления (DATA DIRECTION REGISTERS - **DDR**):

- Регистры направления (DDRx) определяют, будет ли конкретный пин порта использоваться в качестве входа или выхода.

4. Подтягивающие резисторы:

- Каждый пин порта может быть настроен как вход с подтягивающим резистором, который устанавливается в регистре **PORTx**.

5. Работа с битами:

- Микроконтроллер позволяет устанавливать и сбрасывать биты в регистрах PORTx и DDRx, что обеспечивает удобное управление отдельными пинами порта.

6. Порты аналогового ввода:

- АТmega8 также поддерживает аналоговый ввод с помощью портов ADC, позволяя подключать и считывать аналоговые сигналы.

7. Множество возможностей использования:

- Порты микроконтроллера ATmega8 могут использоваться для подключения светодиодов, кнопок, сенсоров, датчиков и других периферийных устройств.

Регистры портов:

DDRD = 0 // порт D настроен на прием сигнала

DDRD = 1 // порт D настроен на выход сигнала

Если DDR=0:

PORTD = 0 // подтягивающее сопротивление отключено

PORTD = 1 // подтягивающее сопротивление подключено

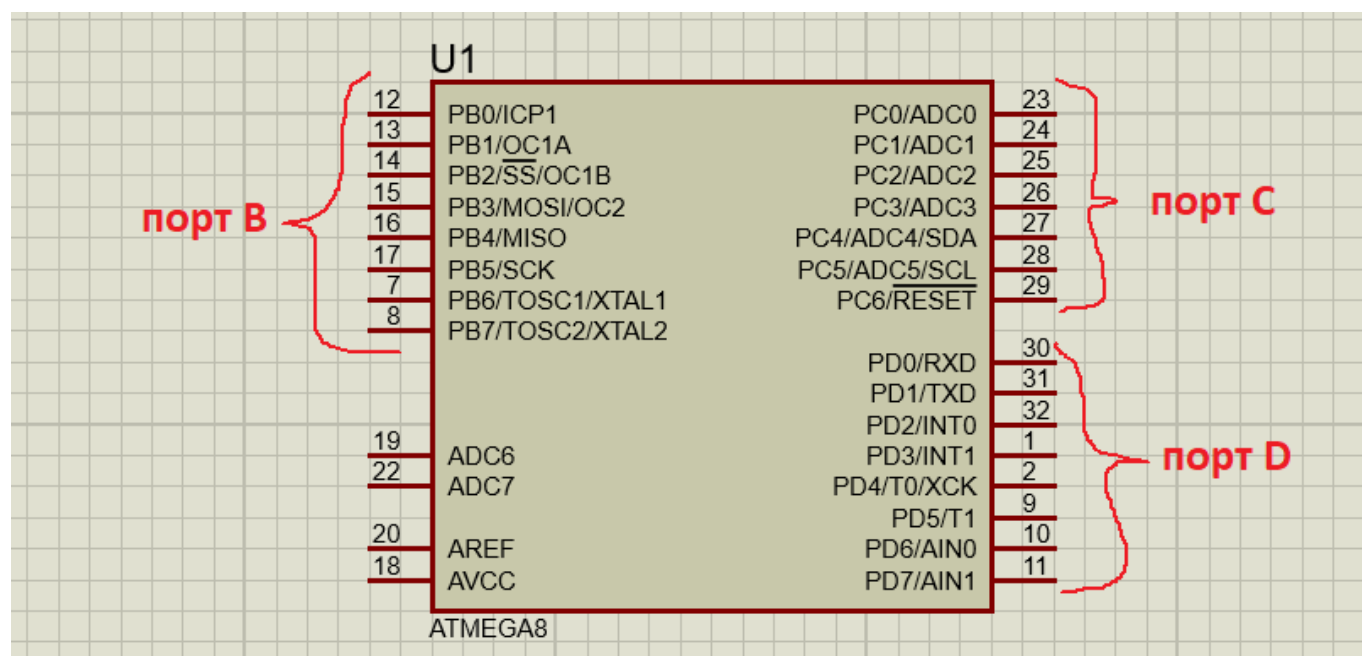
Если DDR=1:

PORTD = 0 // на выходе порта D устанавливается логический 0

PORTD = 1 // на выходе порта D устанавливается логическая 1

PIND – Чтение логических уровней разрядов порта D

Микроконтроллер ATmega8 имеет три порта ввода/вывода: PORTA, PORTB и PORTC. Каждый порт представляет собой набор GPIO (общего назначения), которые можно использовать для взаимодействия с внешними устройствами, такими как светодиоды, датчики, кнопки и другие периферийные устройства. Каждый порт состоит из 8 пинов, что дает в сумме 24 GPIO-пина для использования.



Через дробь указаны дополнительные функции пинов, например, ADC0 указывает, что этот пин PC0 также может быть использован в качестве входа аналогово-цифрового преобразователя.

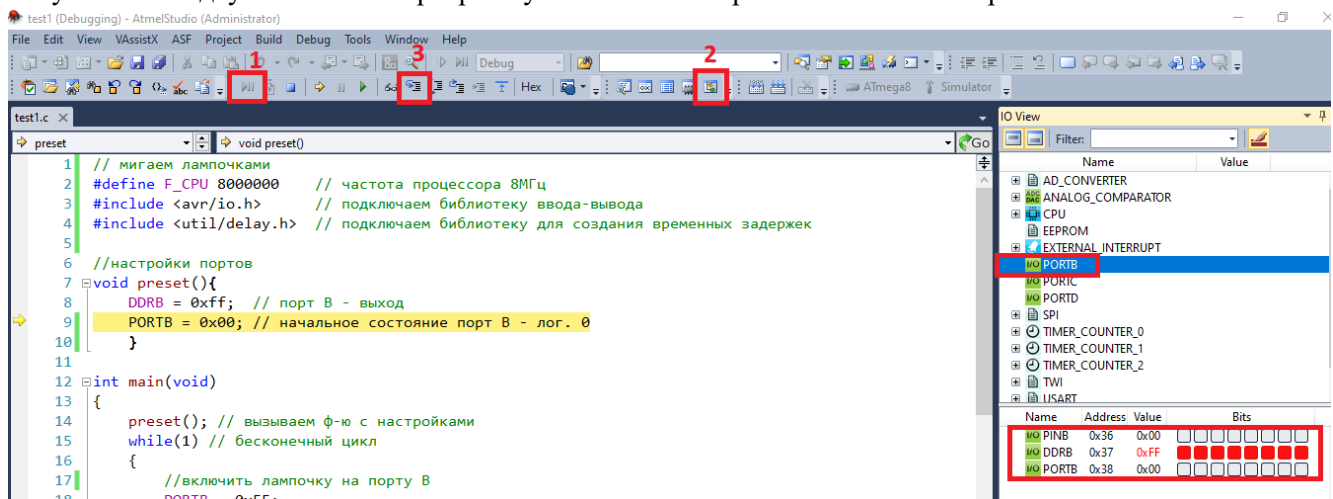
Написание простой программы на C для ATmega8.

```

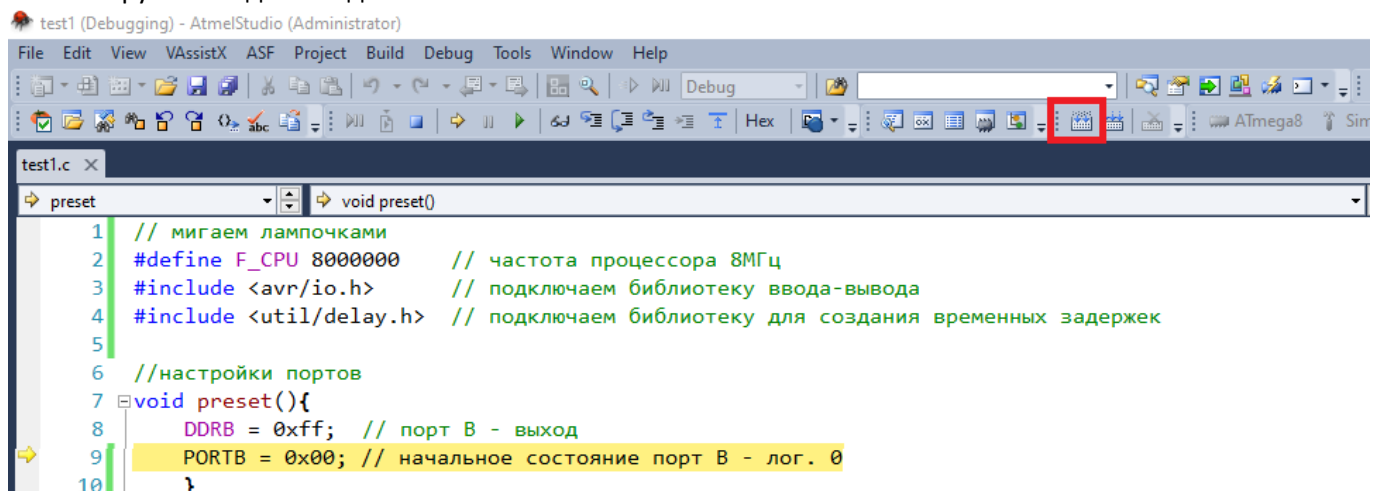
2  #define F_CPU 8000000 // частота процессора 8МГц
3  #include <avr/io.h> // подключаем библиотеку ввода-вывода
4  #include <util/delay.h> // подключаем библиотеку для создания временных задержек
5
6  //настройки портов
7  void preset(){
8      DDRB = 0xff; // порт В - выход
9      PORTB = 0x00; // начальное состояние порт В - лог. 0
10 }
11
12 int main(void)
13 {
14     preset(); // вызываем ф-ю с настройками
15     while(1) // бесконечный цикл
16     {
17         //включить лампочку на порту В
18         PORTB = 0xFF;
19         //пауза 1 сек
20         _delay_ms(1000);
21         //выключить лампочку на порту В
22         PORTB = 0x00;
23         //пауза 1 сек
24         _delay_ms(1000);
25     }
26 }

```

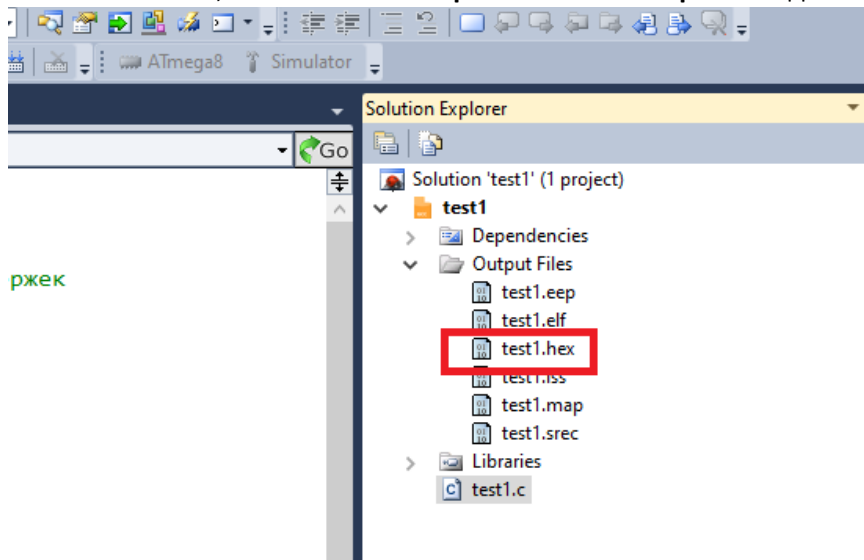
Запускаем отладку и выполняя программу пошагово смотрим на состояние порта D:



Компилируем исходный код:



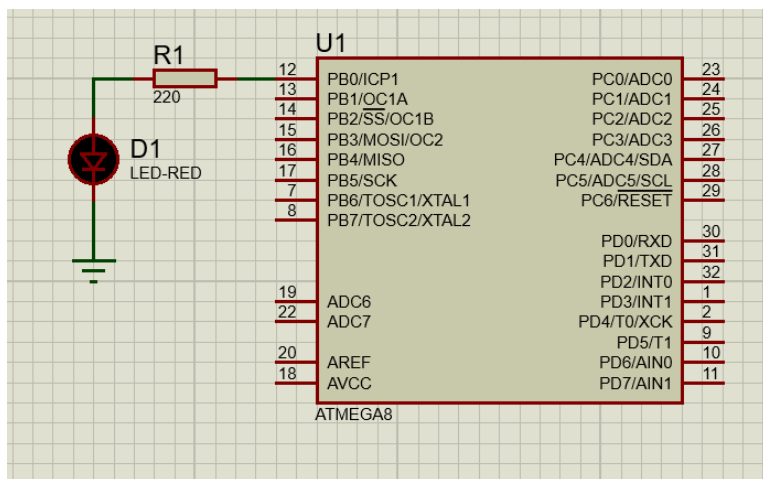
Если ошибок нет, в окне **Solution Explorer** в папке **Output Files** должен появиться HEX-файл:



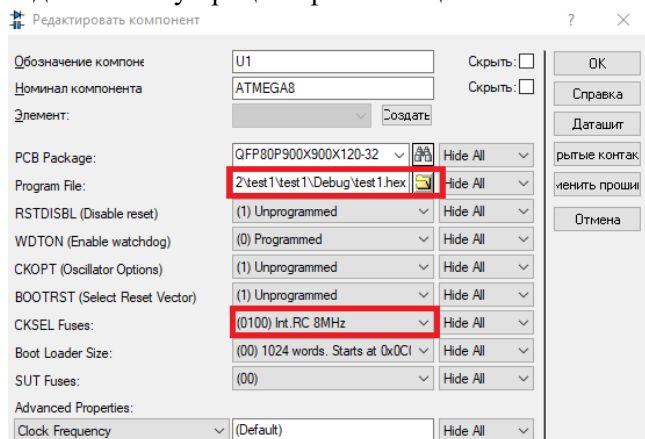
11

Создание проекта в Proteus.

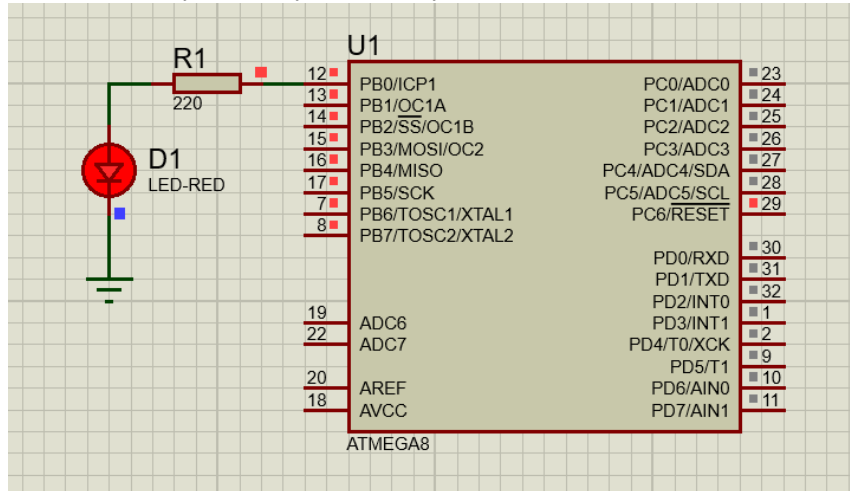
Создайте новый проект, добавьте элементы **ATmega8**, **RES**, **LED-RED** и соберите схему:



Далее кликните дважды по контроллеру, всплывет окно настроек, в нем нужно указать путь к HEX-файлу и задать частоту процессора – 8МГц:



После этого нужно запустить симуляцию, светодиод должен мигать.



Задание 1.

Добавьте в схему еще один светодиод. Подключите его к одному из пинов порта C (или D). Измените код программы так, чтобы светодиоды мигали по очереди.

Задание 2.

Измените код так, чтобы 1-й светодиод мигал 3 раза, затем 2-й один раз. Используйте цикл.

Отчет о лабораторной работе должен содержать:

- скриншот рабочего поля Proteus, с собранной схемой;
- листинг исходного кода.