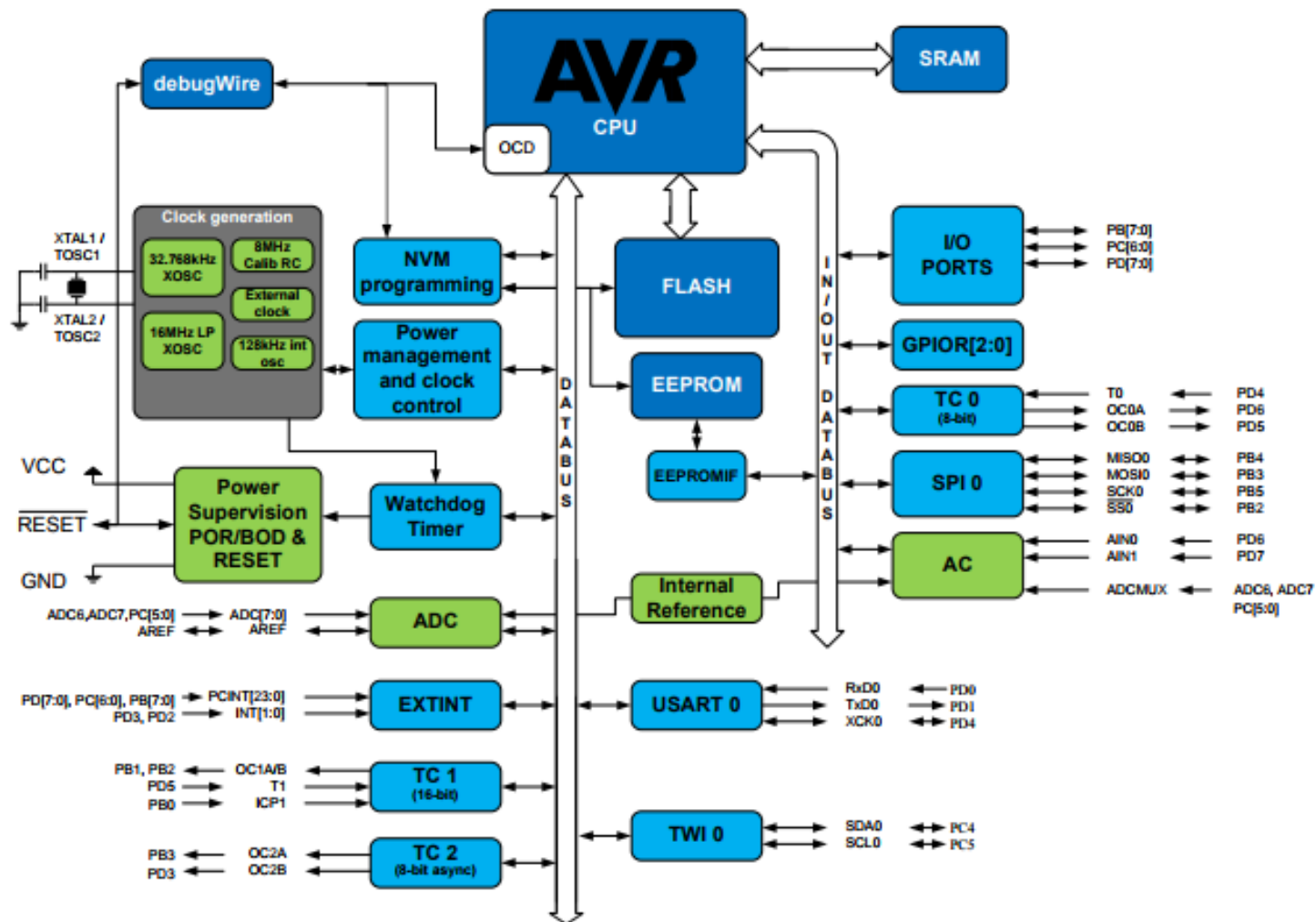


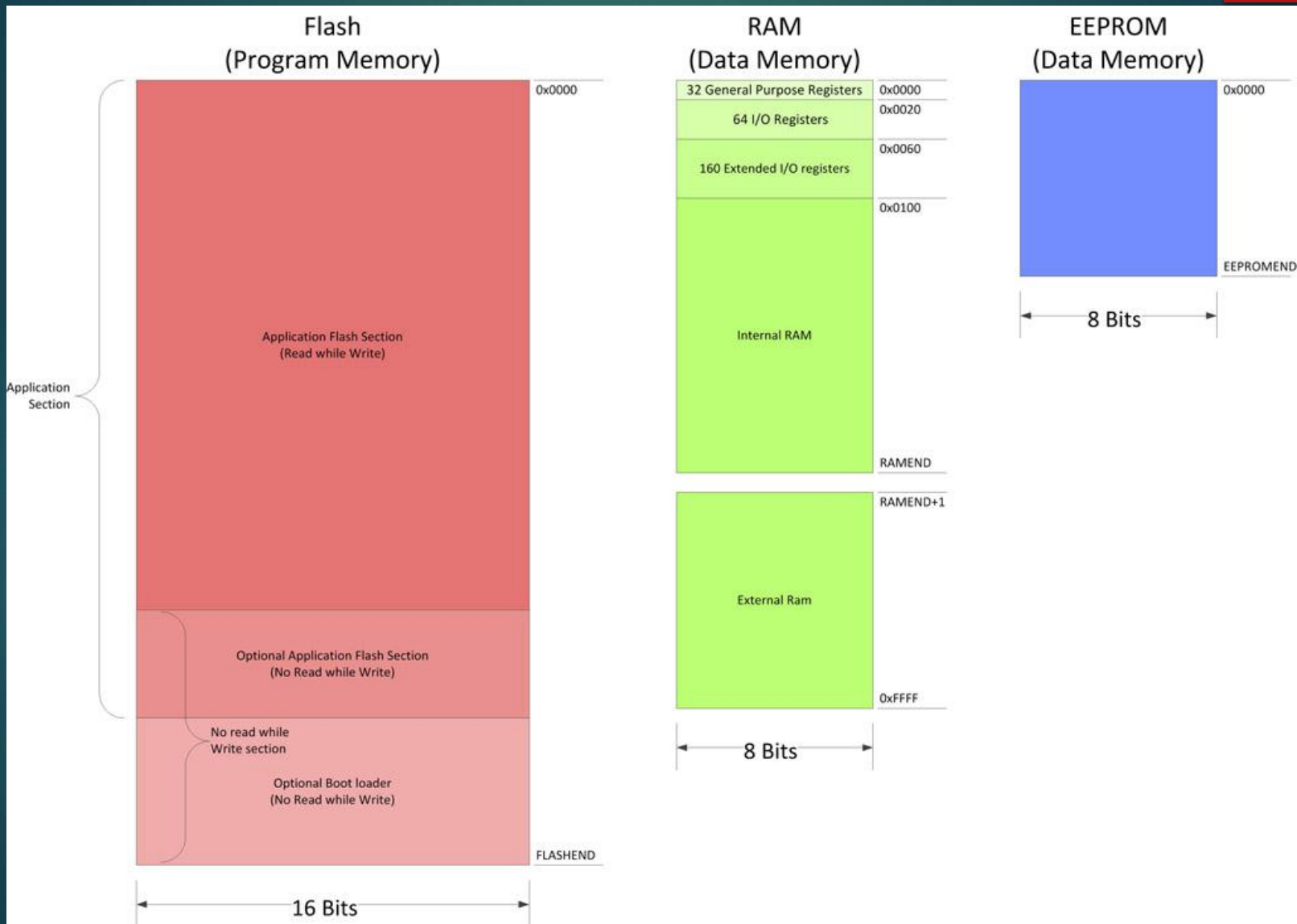
# Программирование на языке C

ЛЕКЦИЯ 3. МИКРОКОНТРОЛЛЕР ATMEGA

# Ядро AVR



# Карта памяти



# Необходимые инструменты

- ▶ Компилятор и стандартная библиотека: `avr-gcc`, `avr-libc`
- ▶ Программатор: `usbasp`
- ▶ Программа управления программатором: `avrdude`
- ▶ (Опционально) среда разработки: `eclipse` + `avr-eclipse-plugin`

# Необходимая документация

- ▶ Документация для МК на официальном сайте:  
<http://www.microchip.com/wwwproducts/en/ATmega328p>

И в первую очередь datasheet:

[http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega328\\_P%20AVR%20MCU%20with%20picoPower%20Technology%20Data%20Sheet%2040001984A.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega328_P%20AVR%20MCU%20with%20picoPower%20Technology%20Data%20Sheet%2040001984A.pdf)

(и его перевод)

<http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/index.htm>

- ▶ Документация avr-libc  
<http://www.atmel.com/webdoc/AVRLibcReferenceManual/index.html>
- ▶ Документация на avrdude, usbasp и прочее

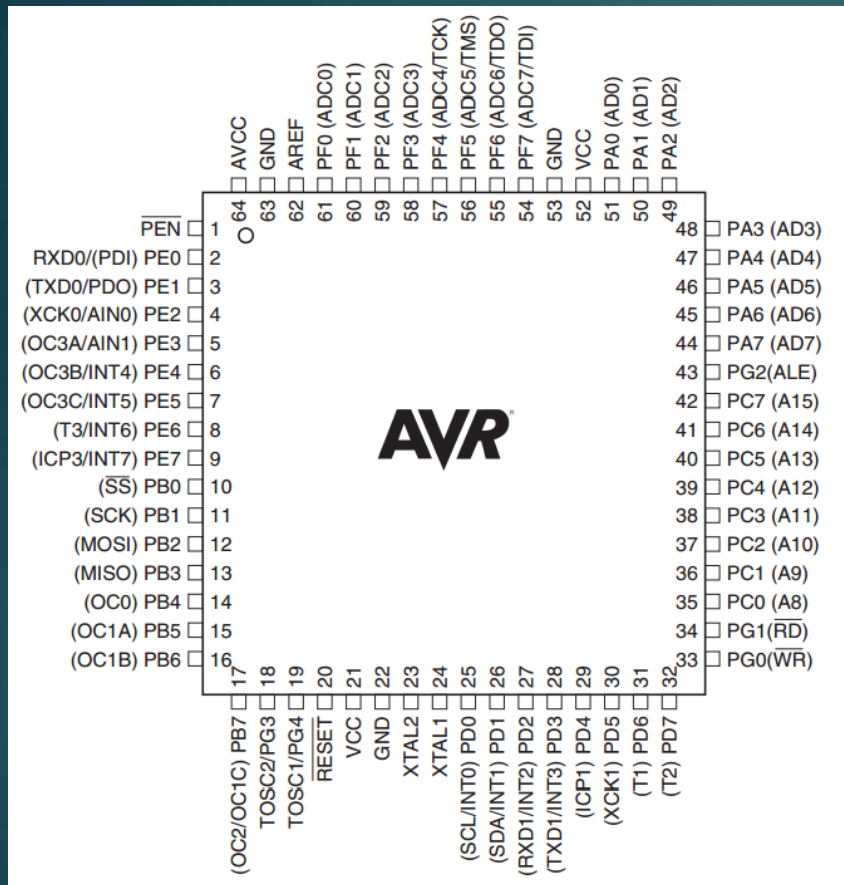
# Сравнение МК семейства atmega

| Microcontroller  | ATmega 16 | ATmega 328P | ATmega 32 | ATmega 644 | ATmega 1284/1284P | ATmega 2560P |
|------------------|-----------|-------------|-----------|------------|-------------------|--------------|
| Max Frequency    | 16MHz     | 16MHz       | 16MHz     | 20MHz      | <b>20MHz</b>      | 16MHz        |
| Digital Pins     | 32        | 23          | 32        | 32         | <b>32</b>         | 54           |
| Analog input     | 8         | 6           | 8         | 8          | <b>8</b>          | 16           |
| SRAM             | 1k        | 2k          | 2k        | 4k         | <b>16k</b>        | 8k           |
| FLASH            | 16k       | 32k         | 32k       | 64k        | <b>128k</b>       | 256k         |
| EEPROM (Bytes)   | 512       | 1024        | 1024      | 2048       | <b>4096</b>       | 4096         |
| UART             | 1         | 1           | 1         | 2          | <b>2</b>          | 4            |
| Interrupts       | 3         | 2           | 3         | 3          | <b>3</b>          | 8            |
| PCINT Interrupts | no        | 23          | no        | 32         | <b>32</b>         | 54           |
| PWM pins*        | 4         | 6           | 4         | 6          | <b>6</b>          | 16           |

<http://ww1.microchip.com/downloads/en/DeviceDoc/30010135D.pdf>



# ATmega128



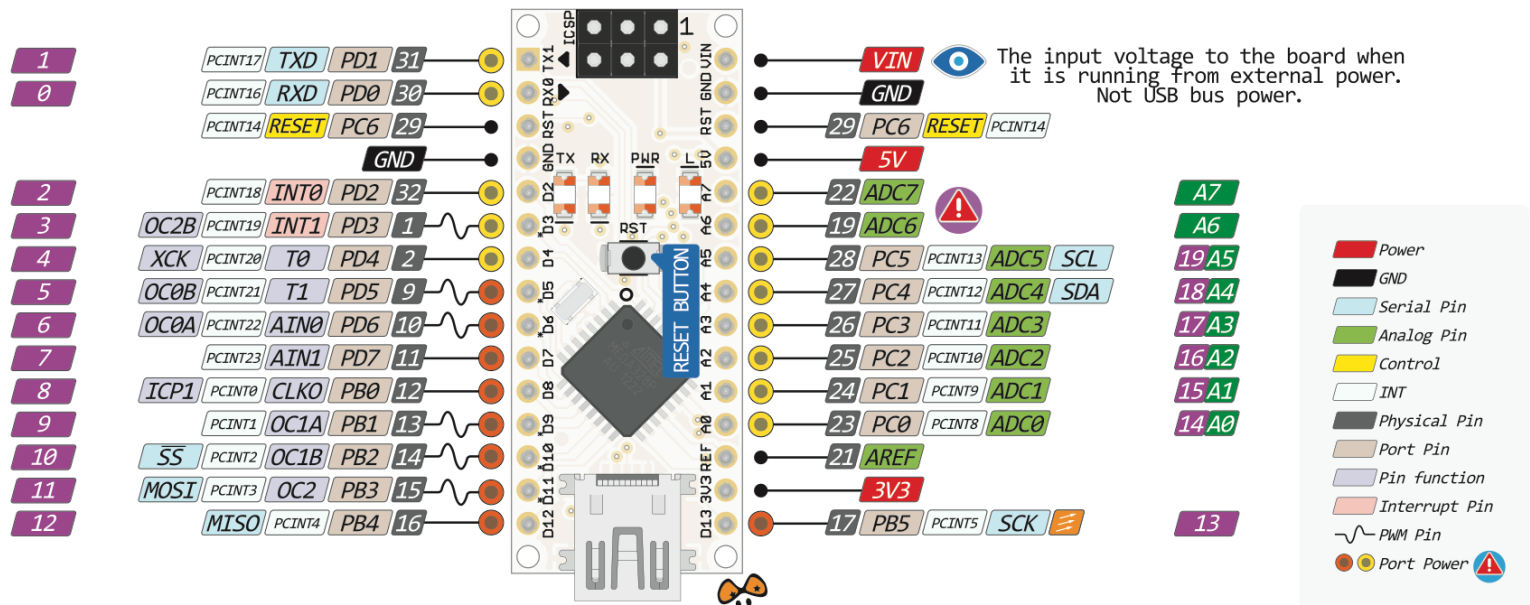
Выходы ATmega128



ATmega 128 в конструкторе cansat

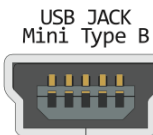
# Arduino

## NANO PINOUT



**Absolute MAX per pin 40mA**  
recommended 20mA

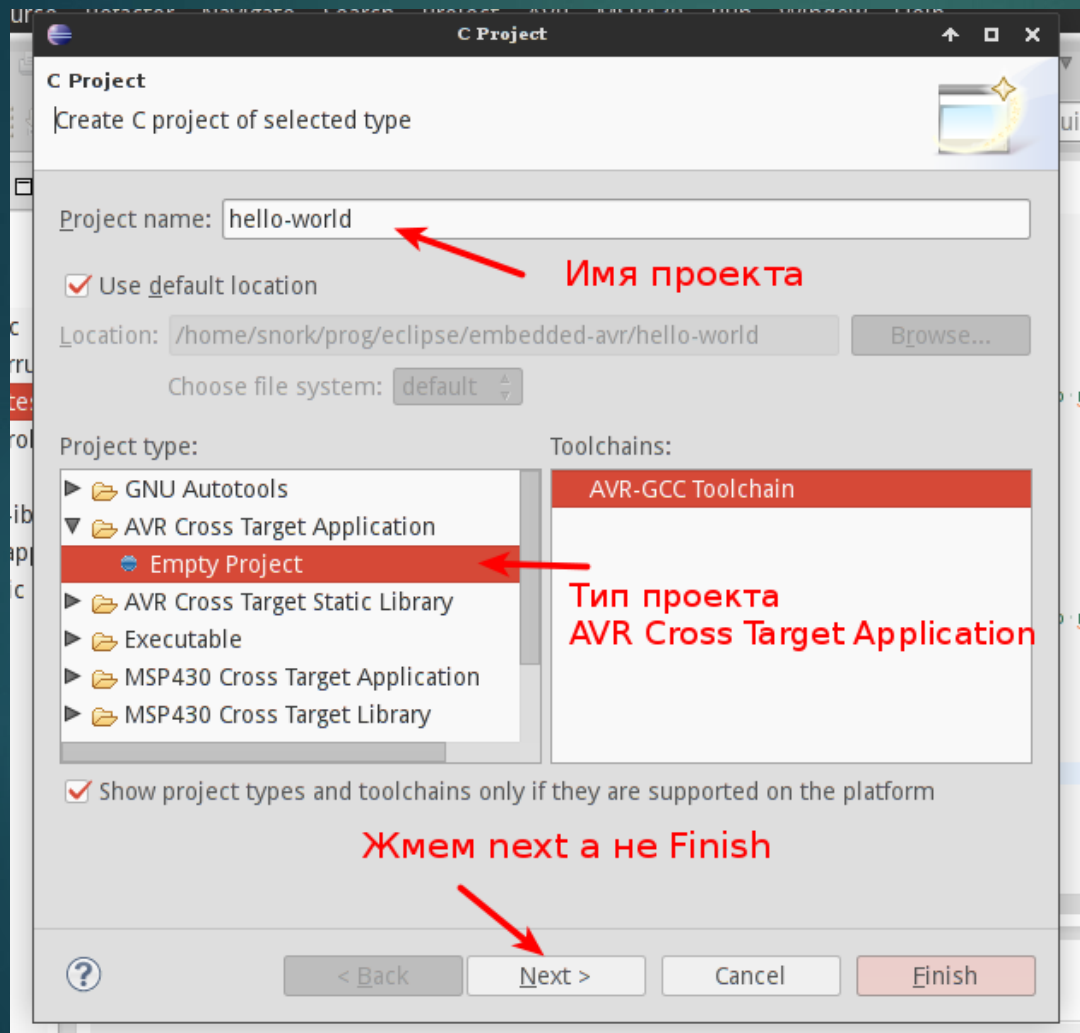
**Absolute MAX 200mA**  
for entire package



**Analog exclusively Pins**



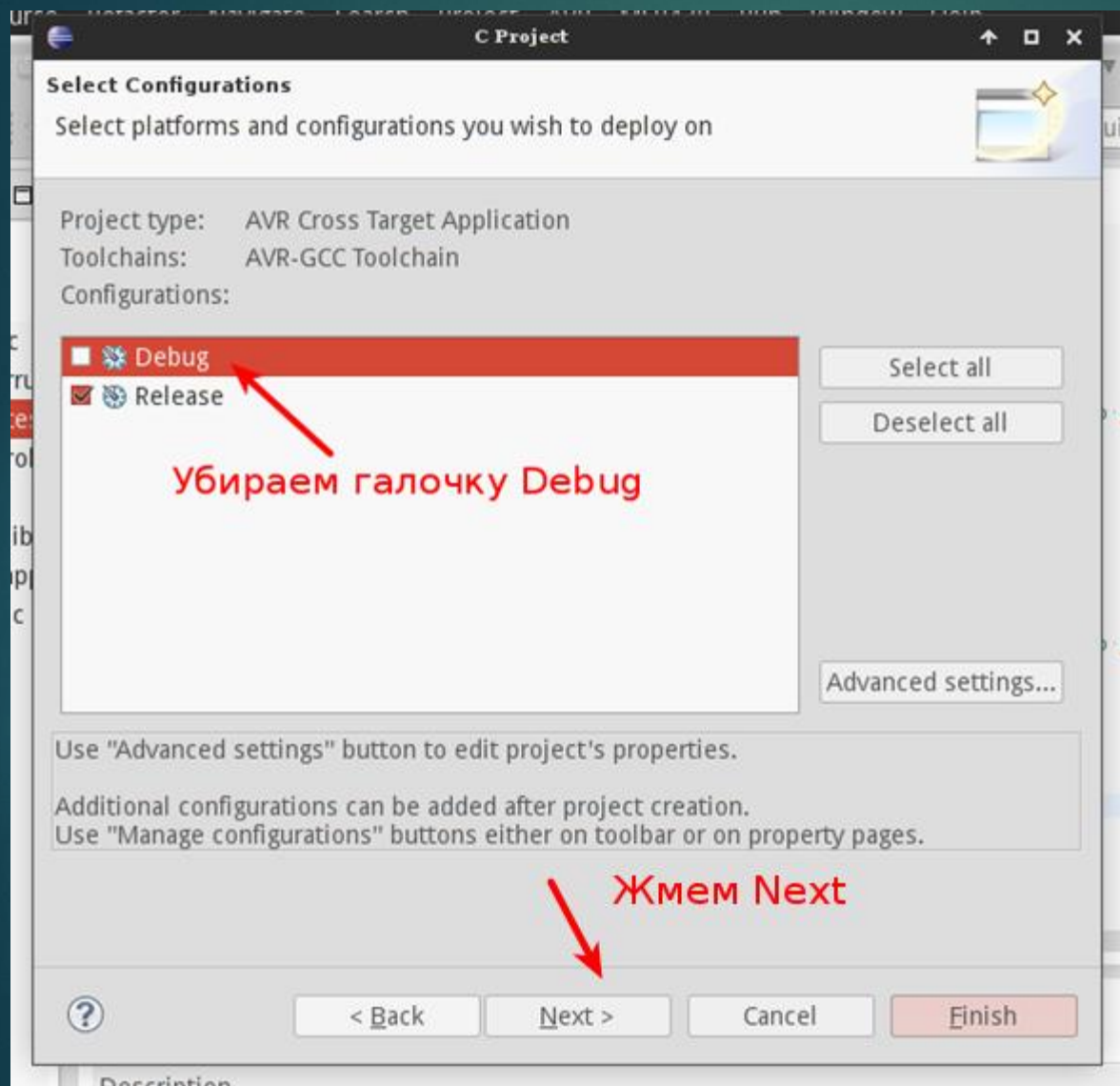
# Создание и настройка проекта AVR



Проект для микроконтроллера создается аналогично проекту для настольного компьютера

Нужно только выбрать другой тип проекта

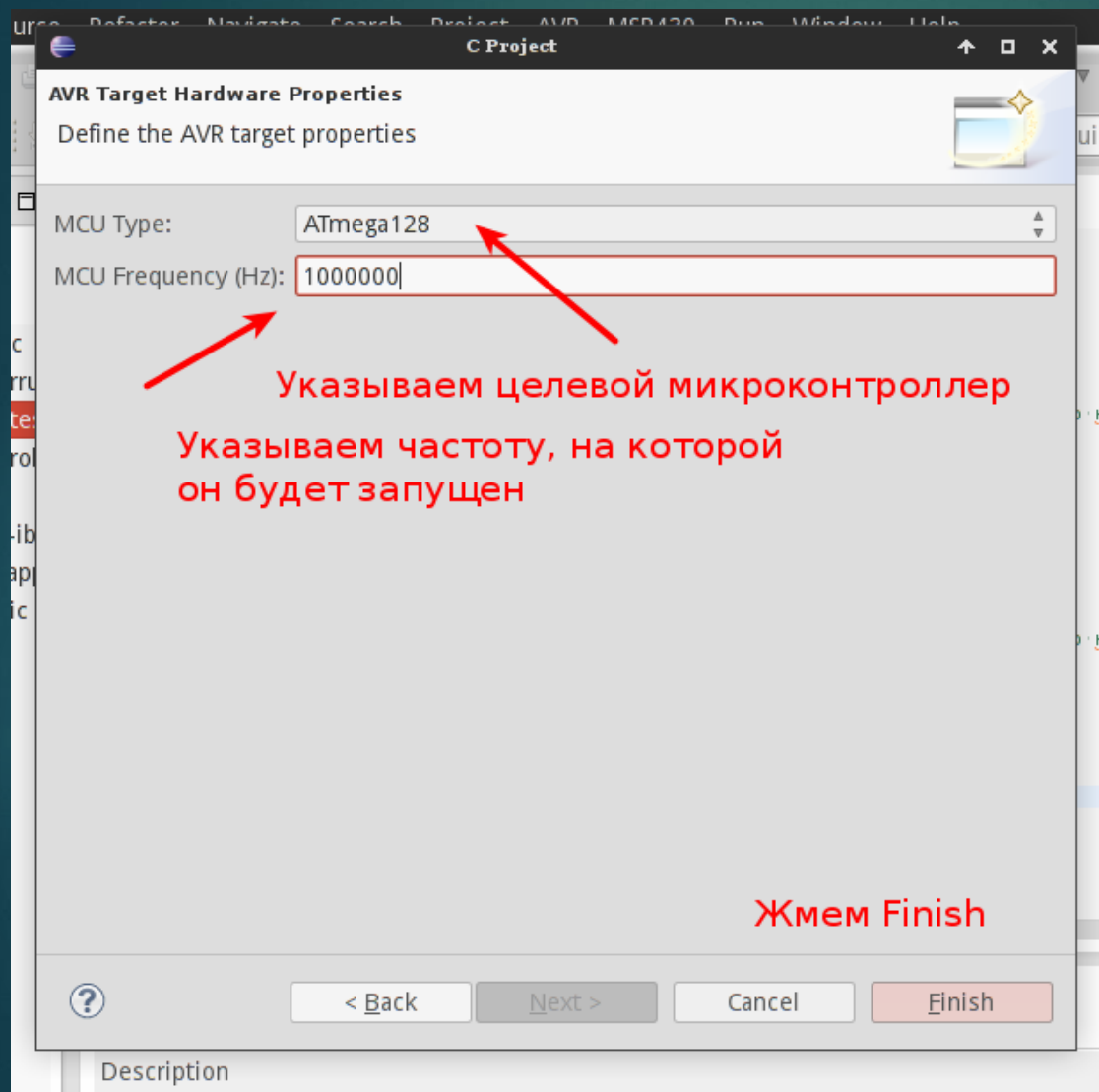
# Создание и настройка проекта AVR



Отладочную конфигурацию проекта стоит отключить. Скорее всего у нас не будет внутрисхемного отладчика, с которым она могла бы быть полезна.

Простую прошивку по кнопке отладочной конфигурации eclipse не настраивает, что может вызывать ошибки и путаницу при сборке и прошивке проекта

# Создание и настройка проекта AVR



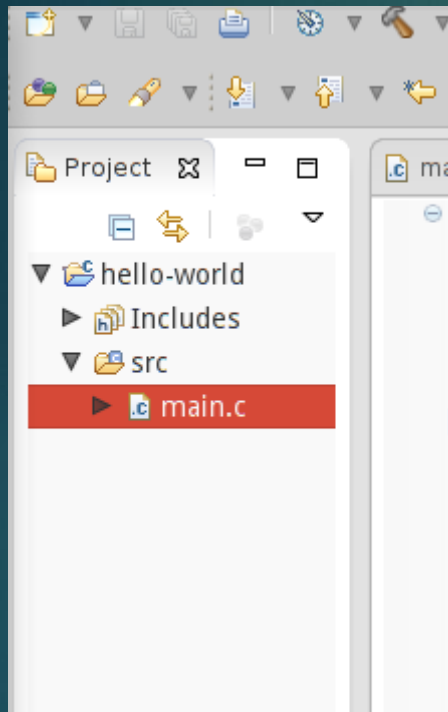
Частота – не настройка контроллера.

Этот параметр будет передан библиотеке `avr-libc`, для правильного расчета параметров, зависящих от частоты.

Фактическая частота контроллера указывается иначе.

Пока не понятно на какой частоте будет запущен контроллер, но можно предположить, что это 16 МГц

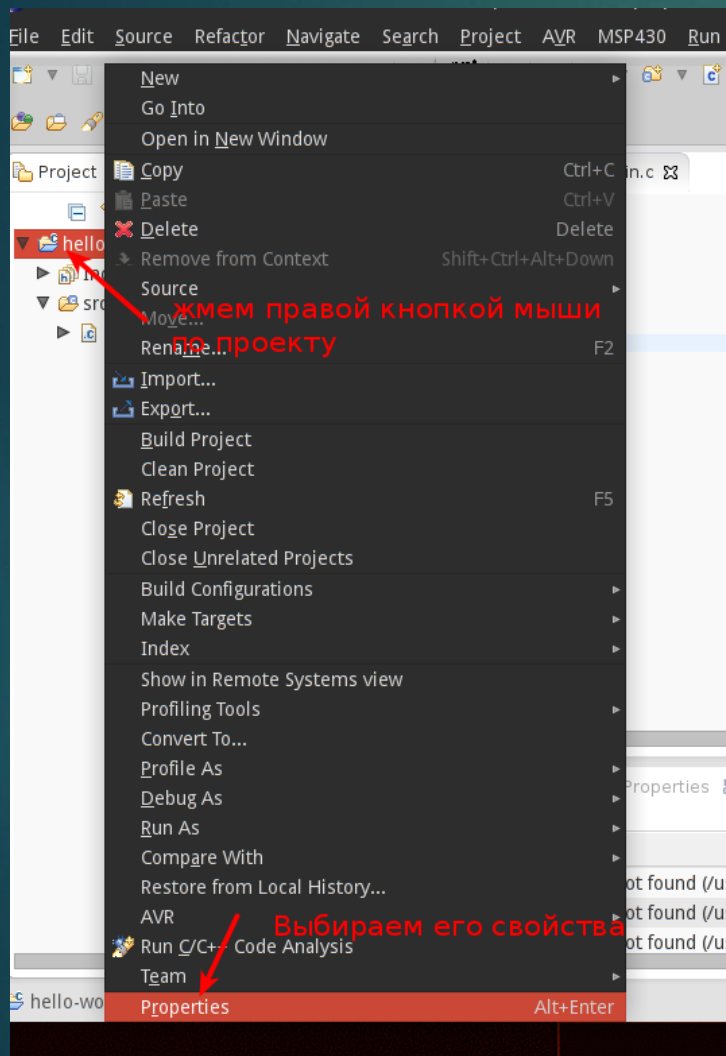
# Создание и настройка проекта AVR



Папки с исходниками  
и файлы исходников  
добавляются так же как в  
обычном проекте

Смотри лекцию 1

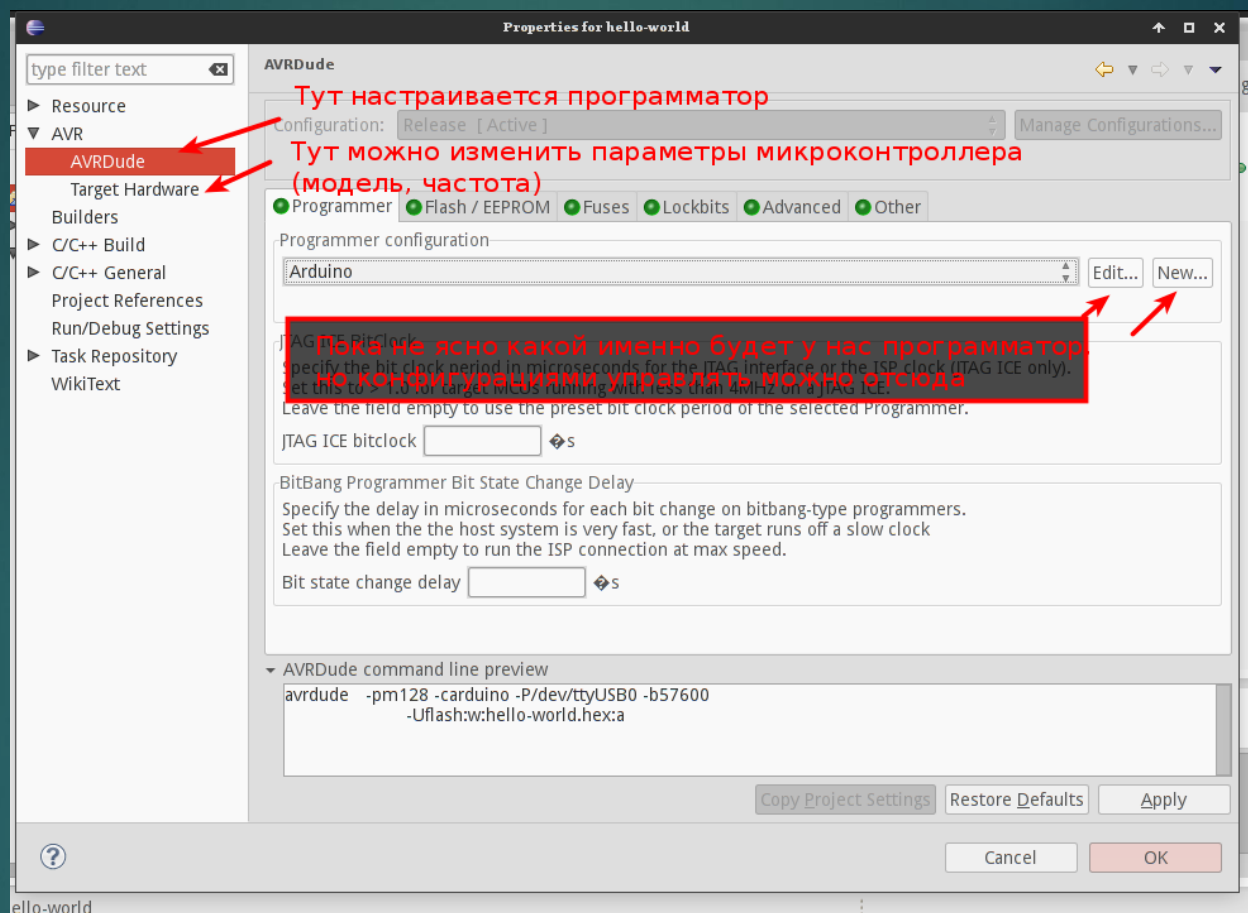
# Создание и настройка проекта AVR



Идем настраивать  
программатор.

Открываем свойства  
проекта

# Создание и настройка проекта AVR

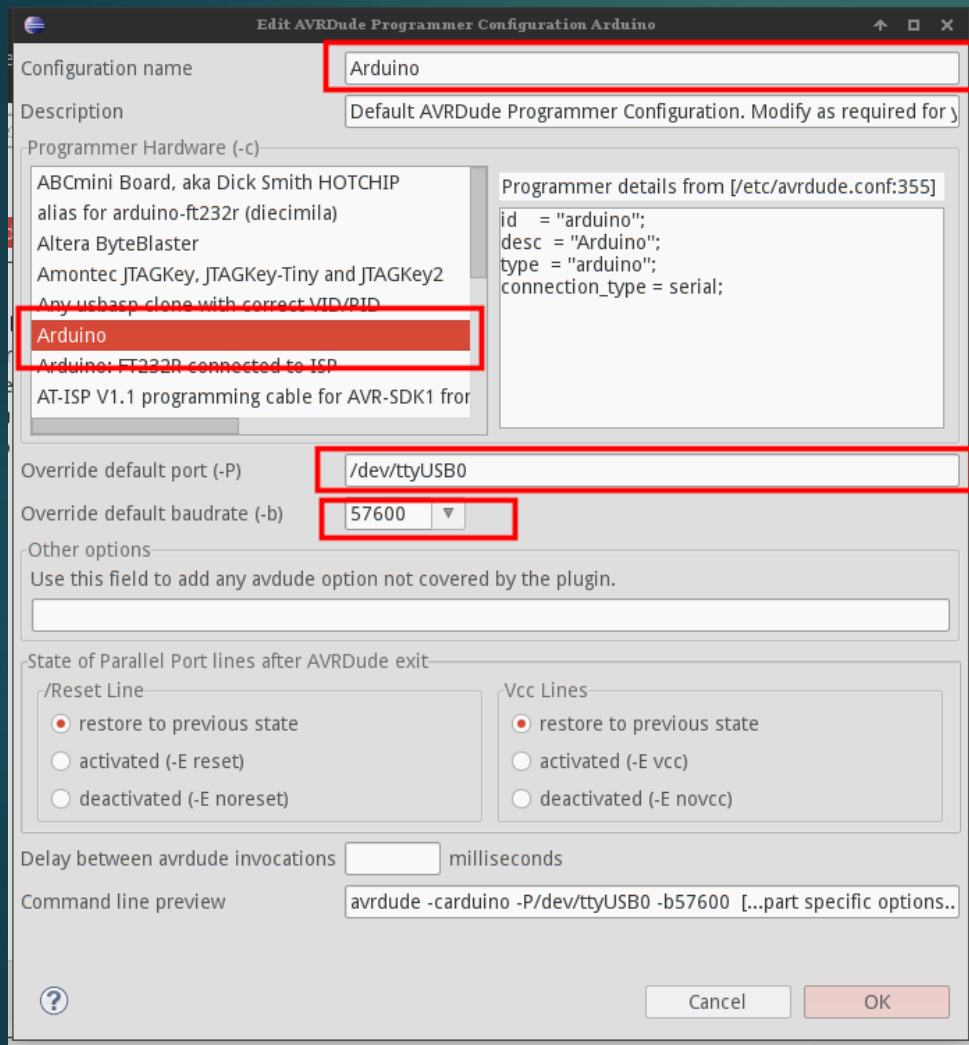


Для отправки прошивки на контроллер нужны две вещи: аппаратное устройство (программатор) и программа им управляющая.

Мы будем использовать программу avrdude, по аппаратной части пока ничего не понятно.



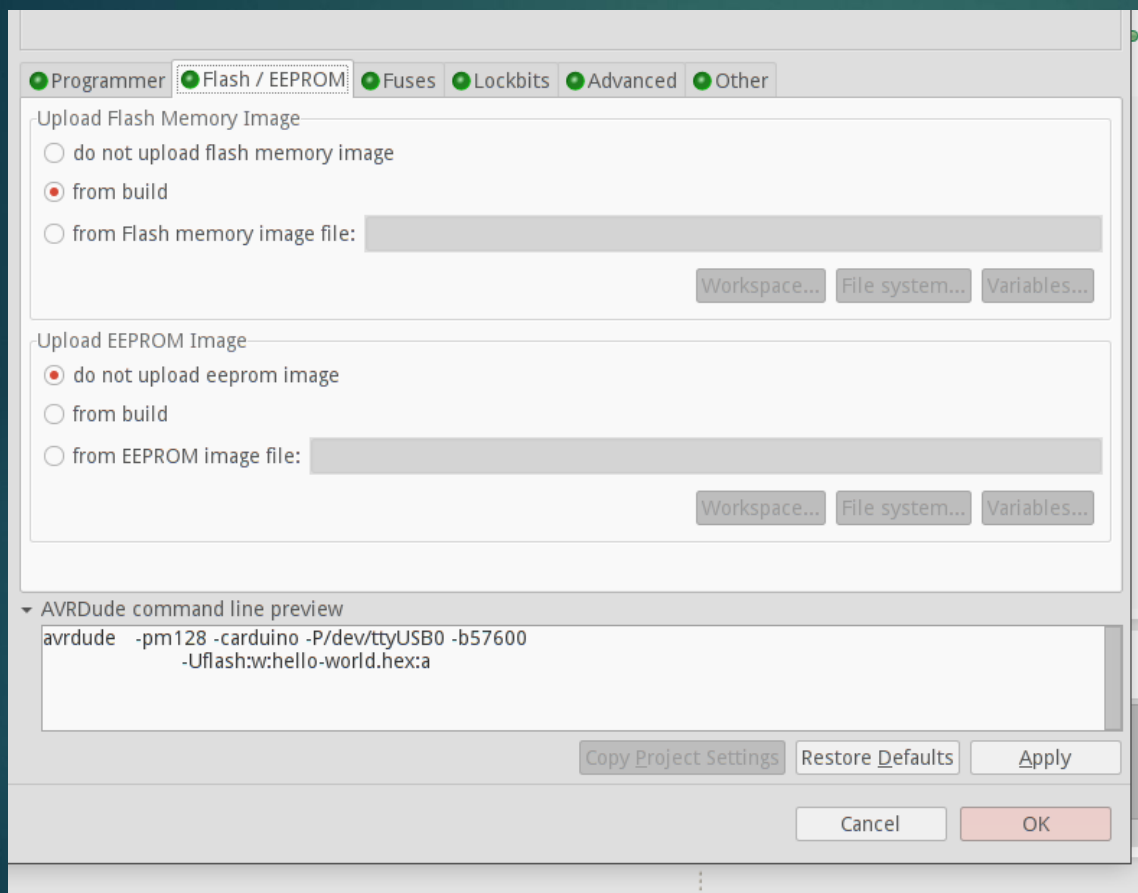
# Создание и настройка проекта AVR



Как пример конфигурация avrdude для arduino

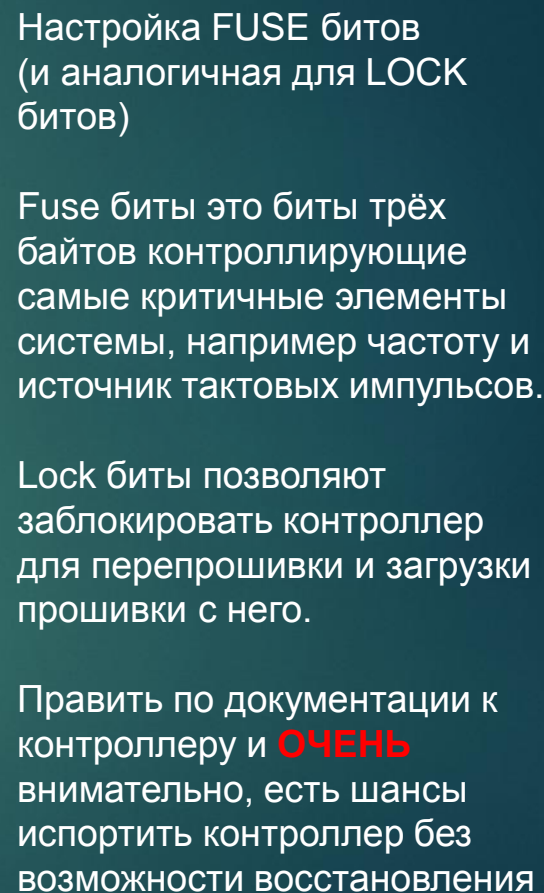
При этом нужно указать соответствующий Микроконтроллер и частоту, так как на arduino используются не 128ые атмеги

# Создание и настройка проекта AVR



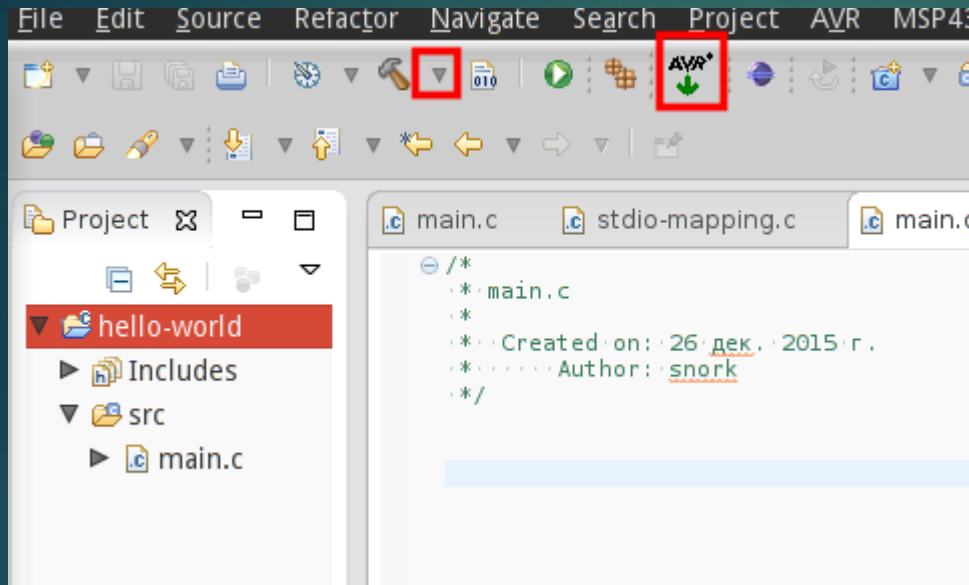
Вкладка настройки  
заливаемых на  
контроллер образов.

Тут можно включить  
прошивку EEPROM образа



<http://easyelectronics.ru/avr-uchebnyj-kurs-konfiguraciya-fuse-bit.html>

# Создание и настройка проекта AVR



При сборке проекта убедитесь, что собираете конфигурацию Release (нажав на маленькую стрелочку у молотка – иконки сборки).

После успешной сборки проекта – по кнопке со значком AVR прошивка будет залита в микроконтроллер

Следите за ошибками в консоли, они не выделяются красным.