


Author	<a href="#">Dmytro Nikandrov</a>
License	

## Практика 0. Готовимся программировать ATMEL AVR используя Atmel Studio на Windows

### Первые шаги

Недавно моя коллекция устройств пополнилась платами Arduino - Uno и Mega ADK. Писать для них код в родной для Arduino среде было как-то совсем просто, и я решил, а почему бы не Assembler?

К сожалению Atmel не выпустила Studio для GNU/Linux и по-этому дальнейшие инструкции будут ориентированы на владельцев Windows 7 64bit и платы Arduino Uno.

Итак сначала нам понадобится серьёзная среда разработки от производителя чипов — Atmel Studio 6. Для того чтобы её скачать требуют пройти регистрацию [на сайте](#).

Нужно ввести ваш e-mail адрес, так как пришлют письмо и надо будет клацнуть в нём ссылку для верификации почты.

После того как вы успешно активируете ваш аккаунт, можно скачать [Atmel Studio](#).

Если вы уверены что на время установки будет в наличии хороший интернет - качайте тот образ что поменьше, если нет - то тот который имеет больший размер. Во время установки меньшего будут запрашиваться различные дополнительные пакеты для закачивания, в большой пакет они и так входят.

Чтобы загружать программы в ваш микроконтроллер (МК), можно использовать тот же метод какой использует сама Arduino IDE. Для заливки программы на чип (для общения через последовательный порт с загрузчиком вашей платы) она использует утилиту AVRDUDE:

[Приложение для Win32](#)

[Его исходные коды](#)

[Руководство](#)

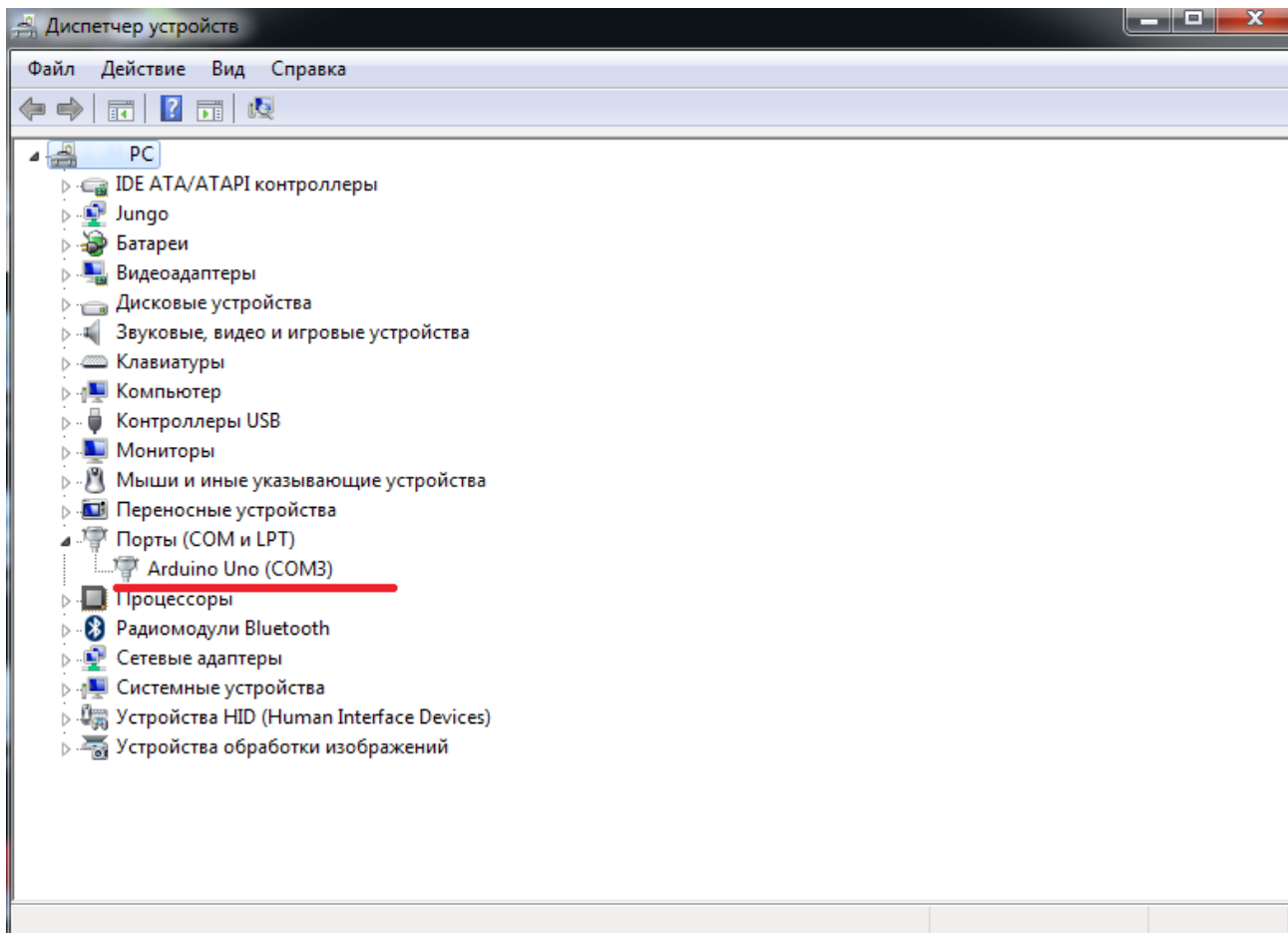
Скачайте и скопируйте её в какую-то папку с коротким путём, в идеале D:\avrdude\  
Она вообще очень привередлива к путям — лучше использовать самые короткие и без пробелов. Убедитесь что в каталоге D:\avrdude\ лежит файл libusb0.dll. Если его там нет - следуйте данному [руководству](#).

Для этой утилиты понадобятся параметры подключения (MCU name, programmer, COM port, overriding baud rate) к вашей плате Arduino. Если вы знаете параметры подключения, то устанавливать Arduino IDE не обязательно, достаточно лишь установить Arduino USB driver для вашей операционной системы. А если не знаете - то нужно скачать и установить последнюю стабильную версию [Arduino IDE](#) для вашей платы. Во время установки — надо включить все флажочки — т. е. устанавливаем всё вместе.

Теперь надо узнать COM-порт к которому подключена ваша Arduino плата. У каждого компьютера он может быть разным. Нажимаем кнопку Пуск, подводим курсор к Компьютер,

щелчок правой клавишей мышки - свойства. В открывшемся окне выберем Диспетчер устройств.

Оставив открытым Диспетчер устройств, отключите вашу плату Arduino если она была подключена, а затем снова подключите, в Диспетчере в секции "Порты (COM и LPT)" появится имя вашей платы и номер порта к которому она подключена.

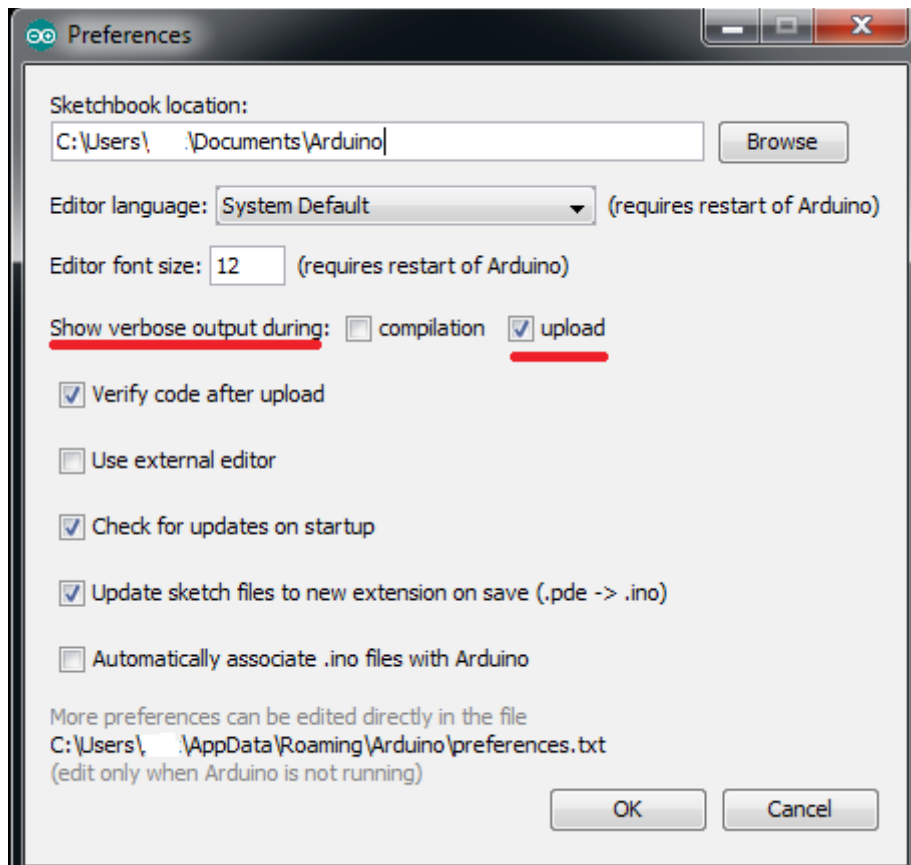


На скриншоте видно, на моём ПК этот номер 3. То есть наш COM-порт № 3.

Если номер порта меньше 10, то как параметр для утилиты avrdude его можно задать явно.

Например, если COM = 3, тогда аргумент будет -PCOM3

Если же он больше 10, тогда надо использовать формат вида -P.COMXX. Например, если COM = 22, то надо использовать -P.COM22



Нажимаем ОК.

Теперь жмём кнопку Upload. Внизу в области уведомлений появится обширный лог утилиты avrdude. Последней строчкой в нём должна быть «avrdude done. Thank you.»

Теперь проскрольте лог в самый верх. Вы увидите:

```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Done uploading.
Binary sketch size: 1,084 bytes (of a 32,256 byte maximum)
C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude -CC:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf -v -v -v -patmega328p -carduino -P\\.\COM3 -b115200 -D
-Flash:w:C:\Users\...\AppData\Local\Temp\build6361430894369757478.tmp\Blink.cpp.hex:i

avrdude: Version 5.11, compiled on Sep  2 2011 at 19:38:36
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch

System wide configuration file is "C:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf"

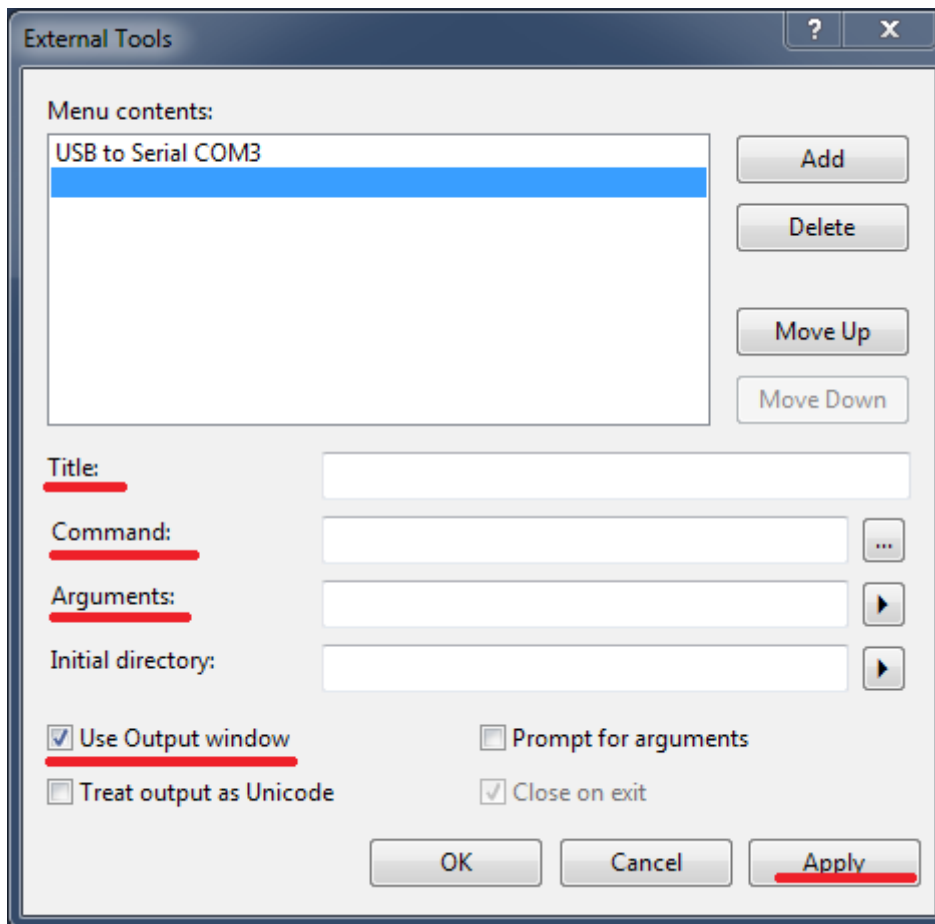
Using Port : \\.\COM3
Using Programmer : arduino
Overriding Baud Rate : 115200

avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
```

Строки которые подчёркнуты и есть нужные нам параметры.

Запускаем Atmel Studio, в верхнем меню переходим Tools -> External tools...

Откроеется окно:



В поле Title: вводим название подключения, например «USB to Serial COM3»

В поле Command: введите путь к файлу avrdude.exe  
D:\avrdude\avrdude.exe

В поле Arguments: введите параметры которые будут переданы утилите при её запуске:  
`-C D:\avrdude\avrdude.conf -v -v -v -v -p atmega328p -c arduino  
-P\\.\COM3 -b115200 -D -Uflash:w:"$(ProjectDir)Debug\$(  
ItemFileName).hex":i`

Также поставьте флажок “Use Output window” - так вы сможете видеть результаты прошивки в виде лога. Нажимаем Apply, затем OK.

Теперь создайте новый проект File->New->Project  
слева в колонке выберите Assembler, внизу заполняете Name и прочие поля.  
Нажимаем OK. В новом окне выбираем Device Family: megaAVR, 8-bit  
а далее в списке ваш чип ATmega328P (это для Arduino Uno R2) и жмём OK.

Быстренько напишите простую assembler-программу по [статье DI HALTa](#).  
Закончили? Настало время её выполнить на реальном микроконтроллере!

Должна быть выбрана вкладка с вашим активным проектом в Studio.  
Выполните команду меню Build -> Build Solution,  
а затем Tools -> USB to Serial COM3 (как раз то что вы прописали в поле Title ранее) начнётся  
заливка в чип вашей скомпилированной программы на Assembler. При этом в Output логе

внизу будет отображаться ход событий.

Иногда можно столкнуться с такой ошибкой:

```
System wide configuration file is "C:/Program"  
avrdude.exe: can't open config file "C:/Program": No such file or directory  
avrdude.exe: error reading system wide configuration file "C:/Program"
```

Это лишь потому, что утилита avrdude не работает с путями содержащими пробелы!  
Используйте простые пути типа D:\MCU\avrdude...

===== ПАРАМЕТРЫ запуска avrdude для плат Arduino =====

Arduino Uno R2:

```
-C D:\avrdude\avrdude.conf -v -v -v -v -patmega328p -carduino  
-P\\.\COM3 -b115200 -D -Uflash:w:"$(ProjectDir)Debug\  
(ItemFileName).hex":i
```

Arduino Mega2560 ADK R3:

```
-C D:\avrdude\avrdude.conf -v -v -v -v -patmega2560 -cwiring  
-P\\.\COM4 -b115200 -D -Uflash:w:"$(ProjectDir)Debug\  
(ItemFileName).hex":i
```

C

превышает порог. По-этому будем использовать транзисторы работающие в режиме ключа на выходах индикатора выбирающих цифру.

[Подробнее про утилиту avrdude](#)

Другие руководства на эту тему:

[Лучшее](#)

[Переезд на Atmel Studio с сохранением работы кода написанного в Arduino IDE](#)

[ATMEL Studio 6 с платами Arduino - Часть 1](#)

[ATMEL Studio 6 с платами Arduino - Часть 2](#)

[ATMEL Studio 6 с платами Arduino - Часть 3](#)

Полезные утилиты:

[winavr](#)

[libusb-win32](#)