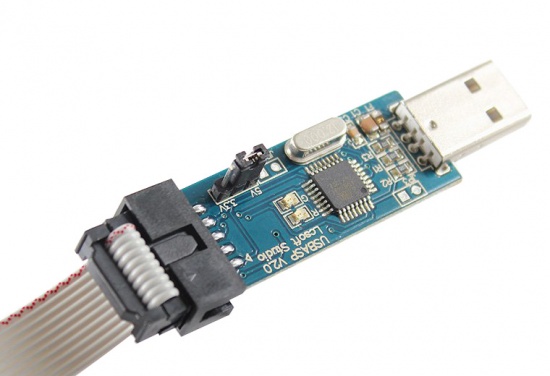
**Использование программатора USBASP в Atmel Studio**

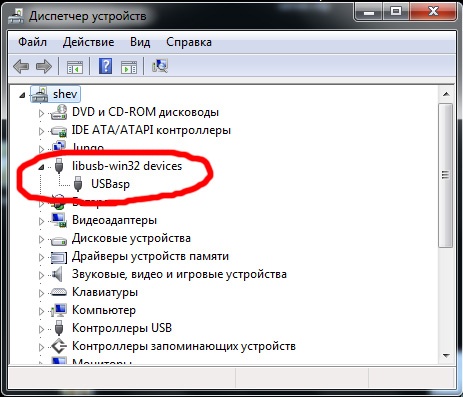


Программатор USBASP является на сегодняшний день самым дешевым программатором микроконтроллеров AVR компании ATMEL и позволяет программировать большое множество микроконтроллеров серий AVR ATTiny, AVR ATMega и других. В данной статье я расскажу вам об основных особенностях использования этого программатора из под ОС Windows 7 и как настроить его работу совместно со средой разработки программ Atmel Studio на примере версии 6.1. К сожалению, по умолчанию, Atmel Studio не поддерживает этот программатор.

**Установка драйвера программатора**

В первую очередь необходимо установить драйвер для программатора. Мы не будем подробно описывать процедуру установки драйвера, так как тут не должно возникнуть каких либо трудностей. После установки драйвера подключенный программатор отображается в диспетчере устройств Windows как устройство USBasp.

Драйвер можно скачать по [ссылке с нашего Google диска](https://drive.google.com/file/d/0B25ZKkkYCPJbZHJQekJhMDA5dmc/view?usp=sharing).



**Установка Avrdude**

Для программирования микроконтроллеров AVR будем использовать программу Avrdude. Эта программа поддерживает большое количество программаторов, в том числе и USBASP. Этой программе посвящена отдельная [страница в википедии](https://ru.wikipedia.org/wiki/Avrdude).

Скачать [Avrdude с сайта проекта](https://www.nongnu.org/avrdude/)

Программа avrdude является консольной и запускается из командной строки. Для выполнения программирования ей передается набор параметров, определяющий настройки. Ниже приведен пример командной строки для программирования контроллера при помощи программатора USBASP:

**avrdude -c usbasp -p atmega32 -U flash:w:myhexfile.hex -U lfuse:w:0x6a:m -U hfuse:w:0xff:m**

Поясним основные параметры:

* **-c usbasp** параметр определяет тип программатора, в нашем случае это usbasp;
* **-p atmega32** параметр определяет тип микроконтроллера, для примера использован ATmega32;
* **-U flash:w:myhexfile.hex** параметр определяет hex файл для записи в контроллер. Файл должен располагаться либо в каталоге программы, либо необходимо указать полный путь к файлу;
* **-U lfuse:w:0x6a:m** параметр определяет младший байт регистра Fuse;
* **-U hfuse:w:0xff:m** параметр определяет старший байт регистра Fuse;

Регистры Fuse используются для настройки режима работы микроконтроллера и для их определения обратитесь к документации соответствующего контроллера. Хочу предупредить, что некорректная установка режима контроллера через регистры Fuse может привести к невозможности дальнейшего программирования контроллера, по этому, внимательно изучите документацию. Если вы не хотите изменять регистр Fuse, вы можете не передавать эти параметры при программировании, в этом случае регистр Fuse останется в том же состоянии, что и до программирования.

Avrdude имеет много разных функций, но в рамках данной статьи нам важен один режим ее работы, а именно заливка программы в микроконтроллер. Остальные функции вы сможете изучить в документации.

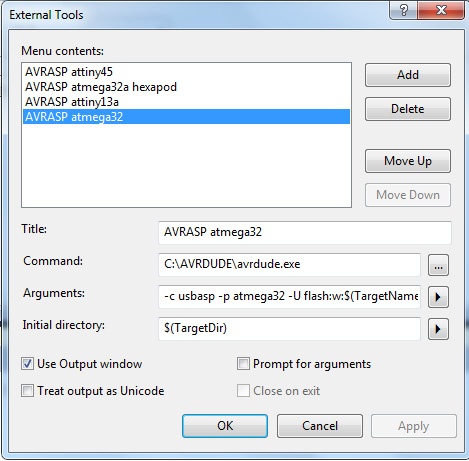
**Настройка программирования из Atmel Studio**

Как уже было упомянуто, Atmel Studio не поддерживает программатор USBASP. Однако в нее встроен механизм для запуска внешних программ для выполнения различных действий. Используя данный механизм внешних инструментов, программирование контроллера будет осуществляться одной кнопкой прямо из меню Atmel Studio.

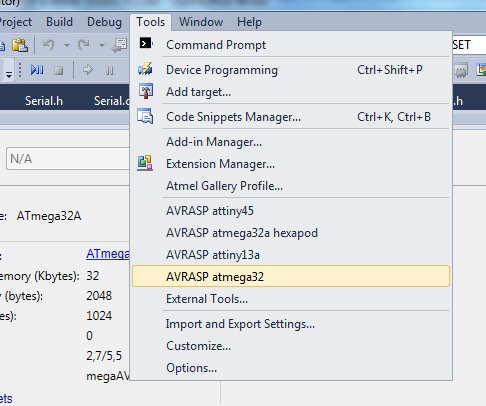
В Atmel Studio открываем пункт меню **Tools** и выбираем пункт **External tools...**

В открывшемся окне настройки инструментов нажимаем кнопку Add для добавления нового инструмента. Заполняем следующие поля:

* **Title**: произвольное имя инструмента, мы указали USBASP atmega32.
* **Command**: файл запуска программы avrdude. Мы указали C:AVRDUDEavrdude.exe, так как у нас она размещена в каталогеAVRDUDE на диске С.
* **Arguments**: параметры запуска для avrdude, которые мы рассмотрели выше. Мы задали строку -c usbasp -p atmega32 -U flash:w:$(TargetName).hex, в которой указали имя программатора, имя контроллера, и путь к файлу прошивки. Путь к файлу указан через специальные макросы и Atmel Studio подставит их сама. Мы не указали регистр Fuse в этом примере, но при необходимости вы можете добавить их самостоятельно.
* **Initial directory**: указан макрос для подстановки каталога, в котором находится файл прошивки.
* **Use Output window**: установите эту галочку для того, что бы результат работы программы показывался в окне Atmel Studio. Послу установки всех настроек нажмите кнопку OK для сохранения настроек и выхода.



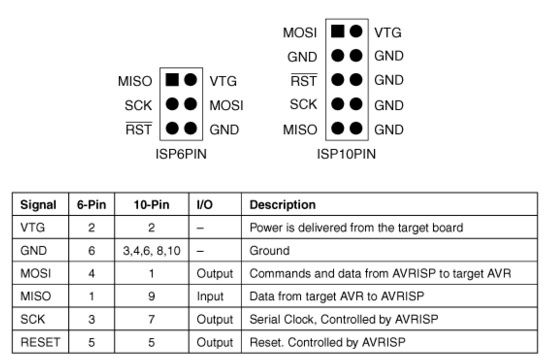
Теперь, для программирования микроконтроллера достаточно зайти в меню **Tools** и нажать команду **USBASP atmega32**, которая там должна появиться после проделанных настроек. Программирование необходимо запускать после компиляции проекта, так как файл прошивки создается в процессе компиляции проекта.



**Подключение программатора к микроконтроллеру**

[Программатор USBASP](http://mcustore.ru/store/programmatory/programmator-vnutrisxemnyj-avr-usbasp-isp/) подключается к микроконтроллеру по стандартному интерфейсу ISP. Физически на программаторе интерфейс имеет 10 контактов. Большинство контактов объединены общим проводом. Назначение используемых контактов следующее:

* MISO, MOSI, SCK, RESET — подключается к соответствующему выводу микроконтроллера;
* GND – земля, подключается к мину или GND микроконтроллера;
* VCC – используется для подачи питания на микроконтроллер.



На печатных платах модулей, где используются контроллеры AVR, разработчики очень часто располагают интерфейс SPI 6 контактов, позволяющий произвести внутрисхемное программирование контроллера. Такой интерфейс можно видеть даже на платах Arduino. Для подключения программатора к такому 6-и контактному интерфейсу можно использовать соединительные провода мама-мама или специальный [переходник ISP10 в ISP6 для программатора AVR USBASP](http://mcustore.ru/store/programmatory/perexodnik-isp10-v-isp6-dlya-programmatora-avr-usbasp/).

