



AVR ISP USB PROGRAMMER OCT 09



Package contains:

AVR ISP USB Programmer, USB Cable, 10 Pin FRC Cable
CD containing software and documentation

Introduction:

This AVR ISP is a USB In System Programmer (ISP). With this ISP programmer AVR microcontrollers can be programmed without removing it from existing hardware. Both Slow and Fast programming modes are supported which allows a variety of devices to be programmed.

Features:

- USB compatible (No legacy RS232 required)
- Programs all AVR microcontrollers
- Supports AVR DUDE IDE.
- Target can be directly powered from the programmer.
- Works under multiple platforms like Linux, Mac OS X and Windows.
- No special controllers or smd components are needed.
- Programming speed is up to 5kBytes/sec.
- SCK option to support targets with low clock speed ($< 1,5\text{MHz}$).

Supported AVR Microcontrollers:

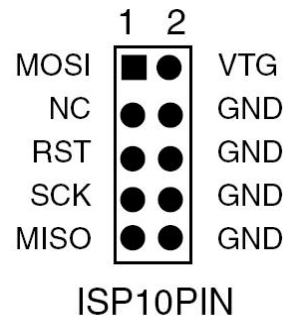
AT90CAN128	AT90PWM2	AT90PWM3	AT90S1200	AT90S2313
AT90S2333	AT90S2343 (*)	AT90S4414	AT90S4433	AT90S4434
AT90S8515	AT90S8535	ATmega103	ATmega163	ATmega1280
ATmega128	ATmega8	ATmega16	ATmega64	ATmega169
ATmega161	ATmega162	ATmega163	ATmega164	ATmega2560 (**)
ATmega2561 (**)	ATmega32	ATmega324	ATmega329	ATmega3290
ATmega48	ATmega640	ATmega644	ATmega649	ATmega6490
ATmega8515	ATmega8535	ATmega88	ATtiny12	ATtiny13
ATtiny15	ATtiny2313	ATtiny25	ATtiny26	ATtiny45
ATtiny85	ATmega1281	AT90S2323		

(*) The AT90S2323 and ATtiny22 use the same algorithm.

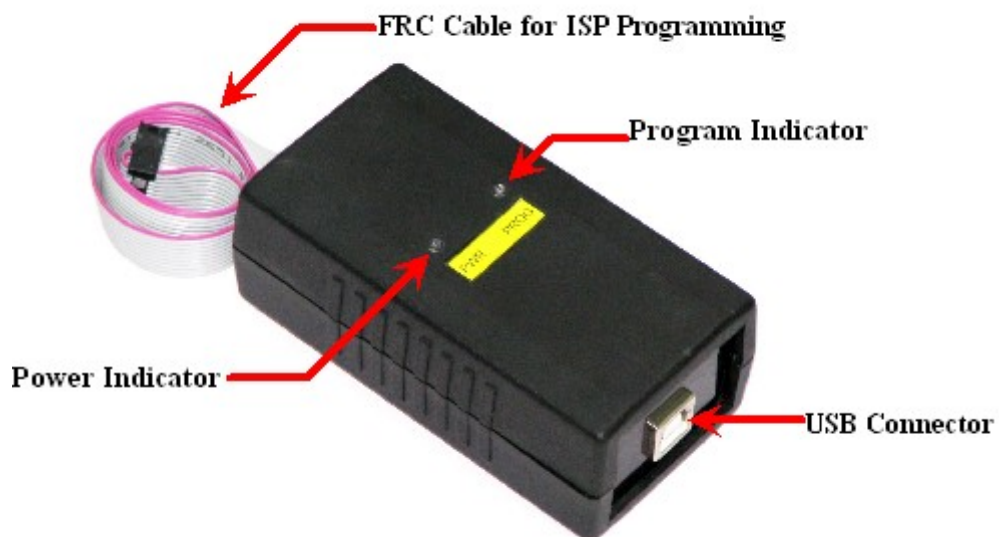
(**) Flash addressing above 128 KB is not supported by all programming hardware. Known to work are jtag2, stk500v2, and bit-bang programmers.

NOTE: This circuit can only be used for programming 5V target systems and for voltages level converter is needed.

ISP connector pin connections:

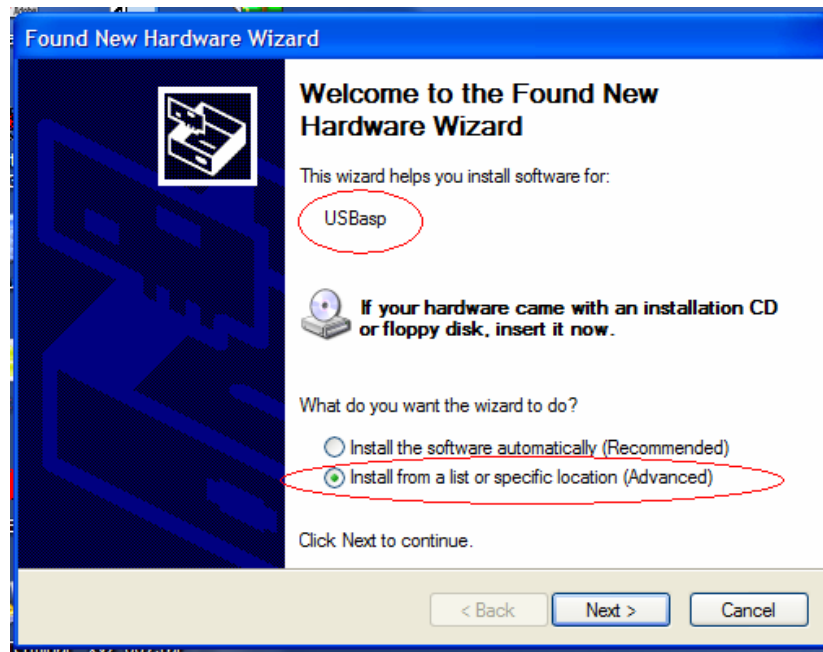


NEX USB PROGRAMMER:

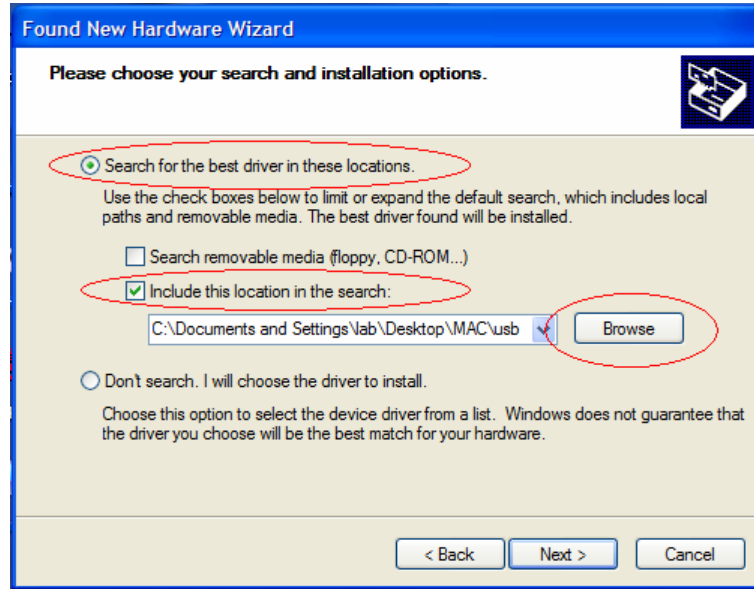


Installation of Driver for AVR ISP USB Programmer:

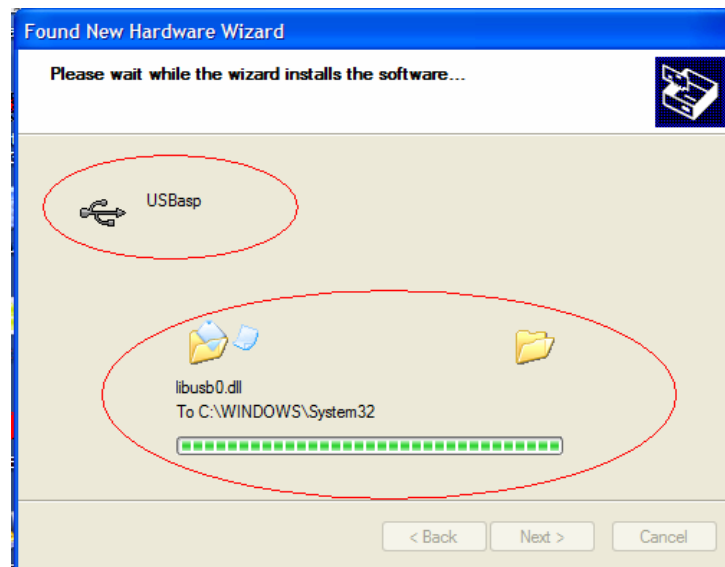
Step1: Connect your programmer to the USB port, computer shows that a new device has been found. It asks you to install the appropriate software (driver). Select "Install from a specific location".



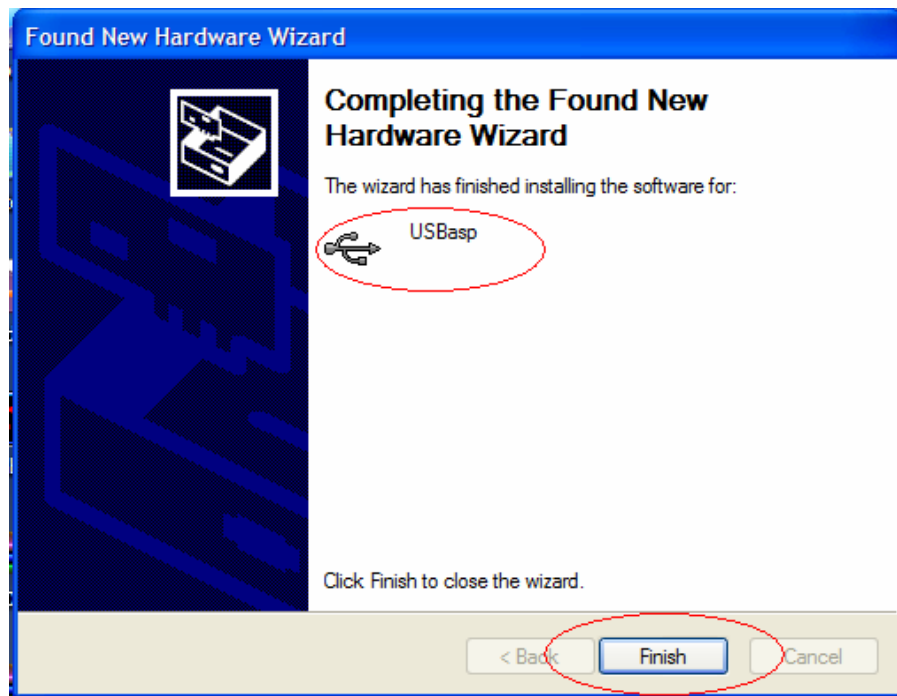
Step2: The computer asks you to select the installation path for the driver. Select the path where the driver is located. Tick "Include this location in the search" and specify the path for the driver. The driver is always located in the `usbasp.2009-02-28\bin\win-driver` folder.



Step3: The computer installs the driver for the USB programmer by copying it in its system folder.

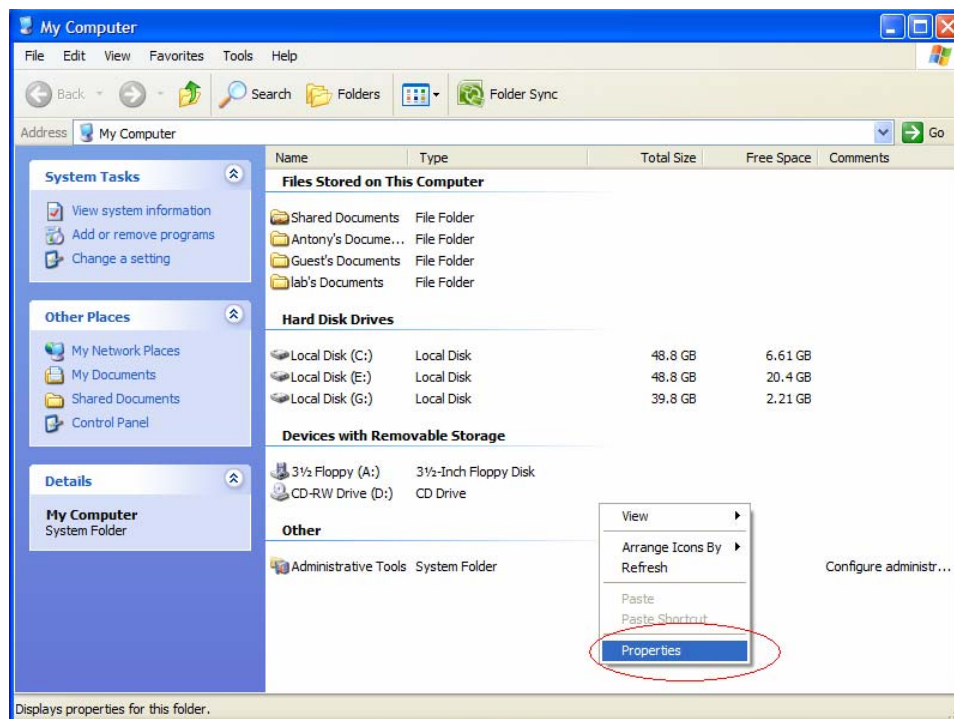


Step 4: The driver is successfully installed. Click finish to end the installation process.

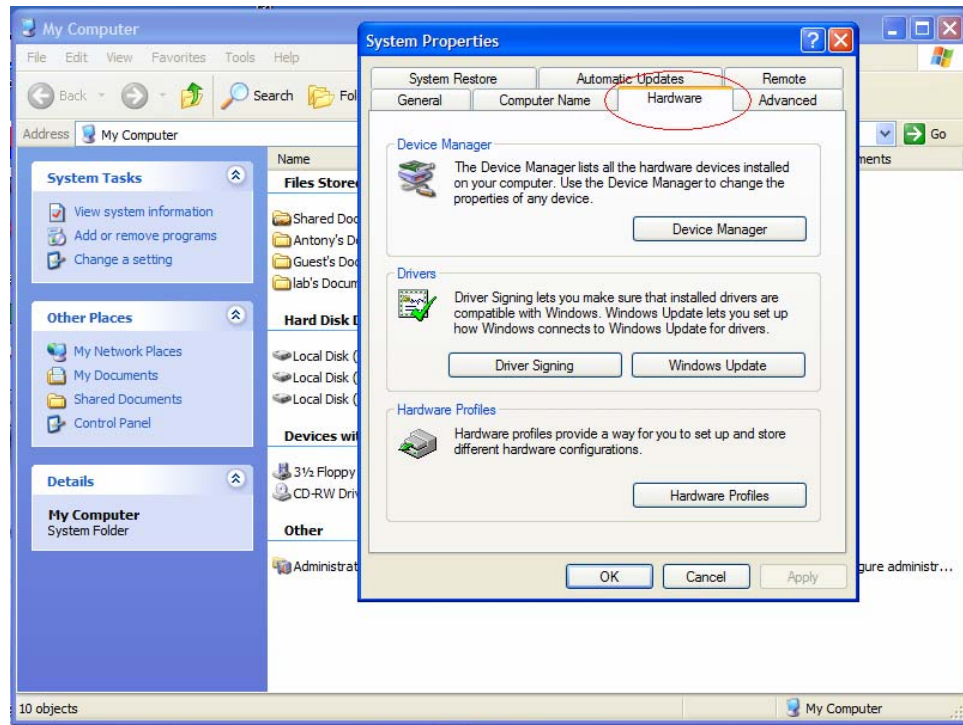


How to use AVR ISP USB Programmer?

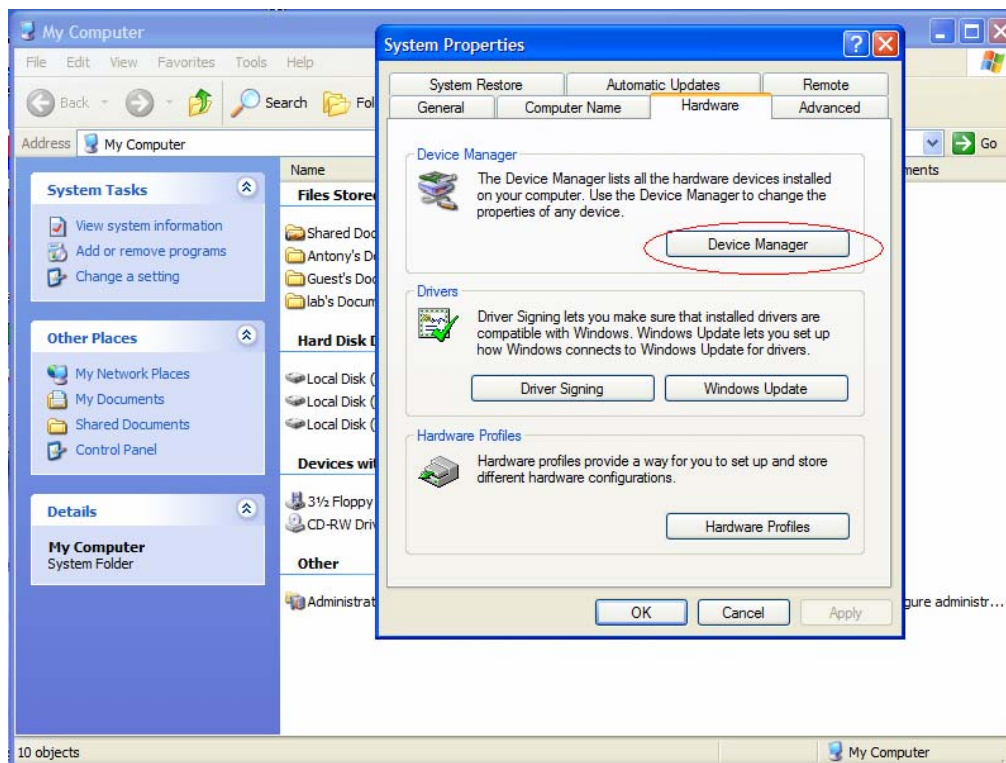
Step 1: Go to My Computer, right click and select Properties.



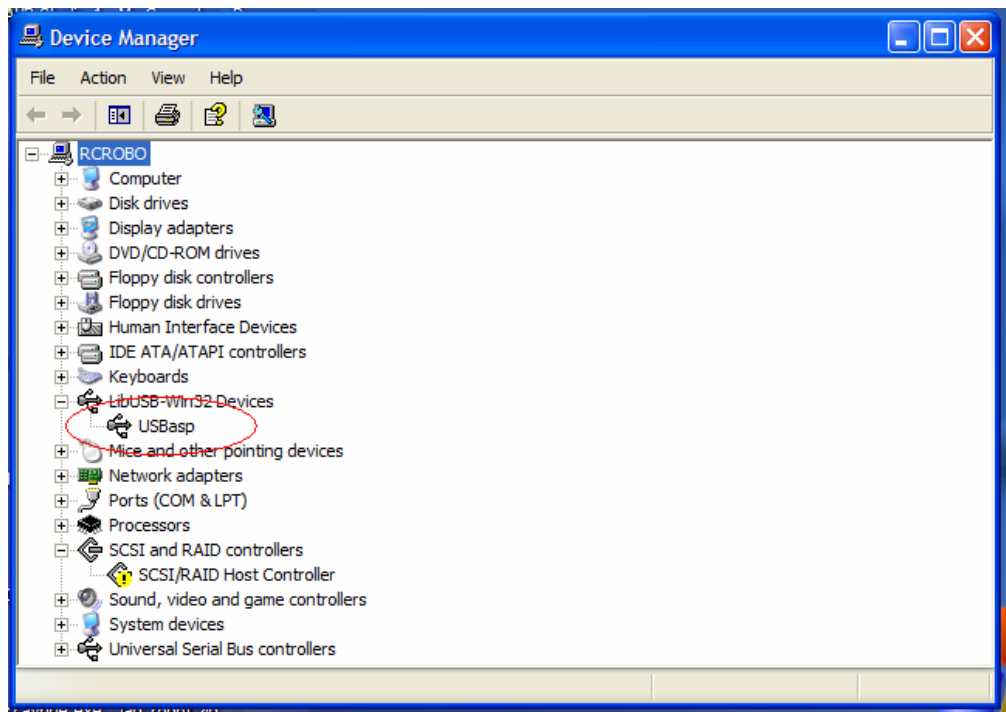
Step 2: Click on the Hardware tab.



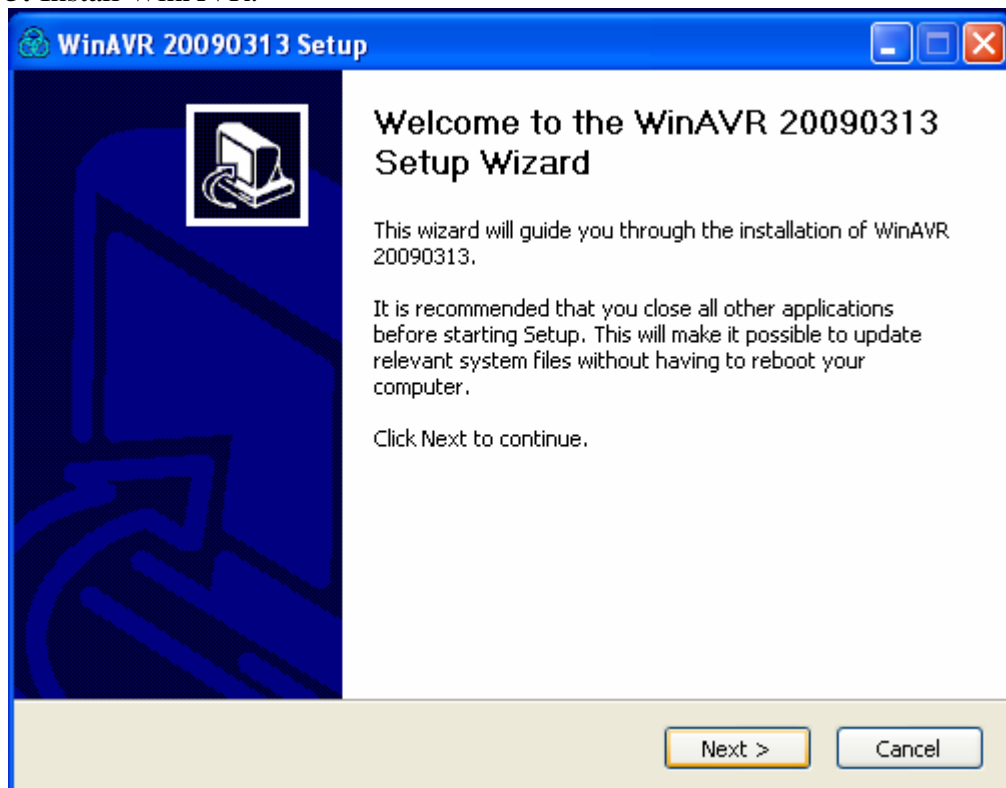
Step 3: Click Device Manager.



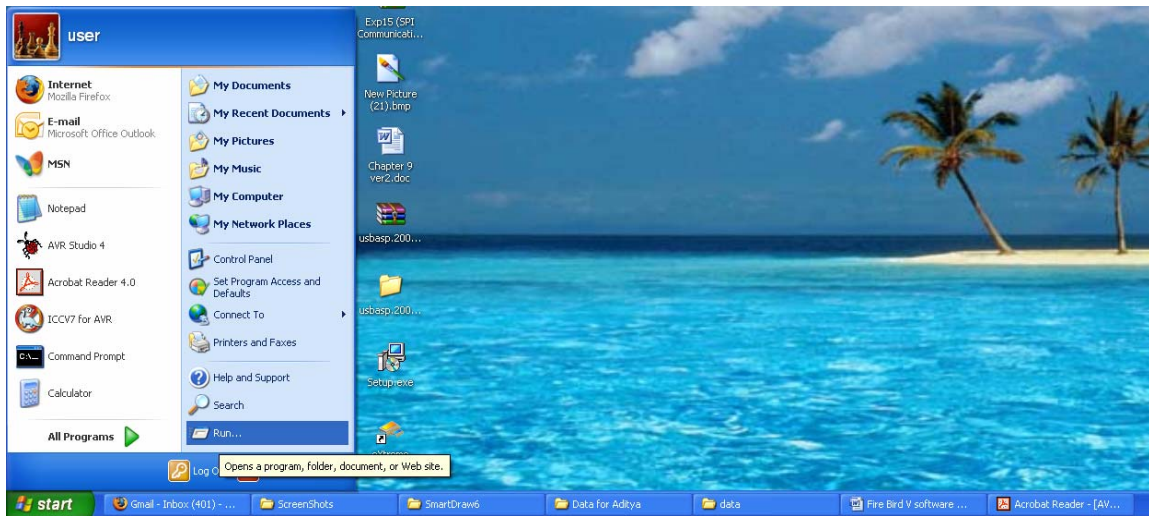
Step 4: Verify that an icon called LibUSB-Win32 Devices with USBasp appears in it.



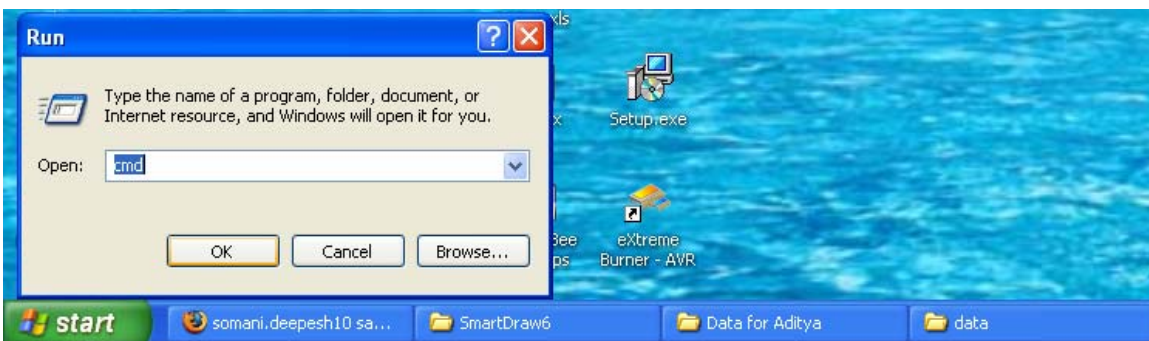
Step 5: Install WinAVR.



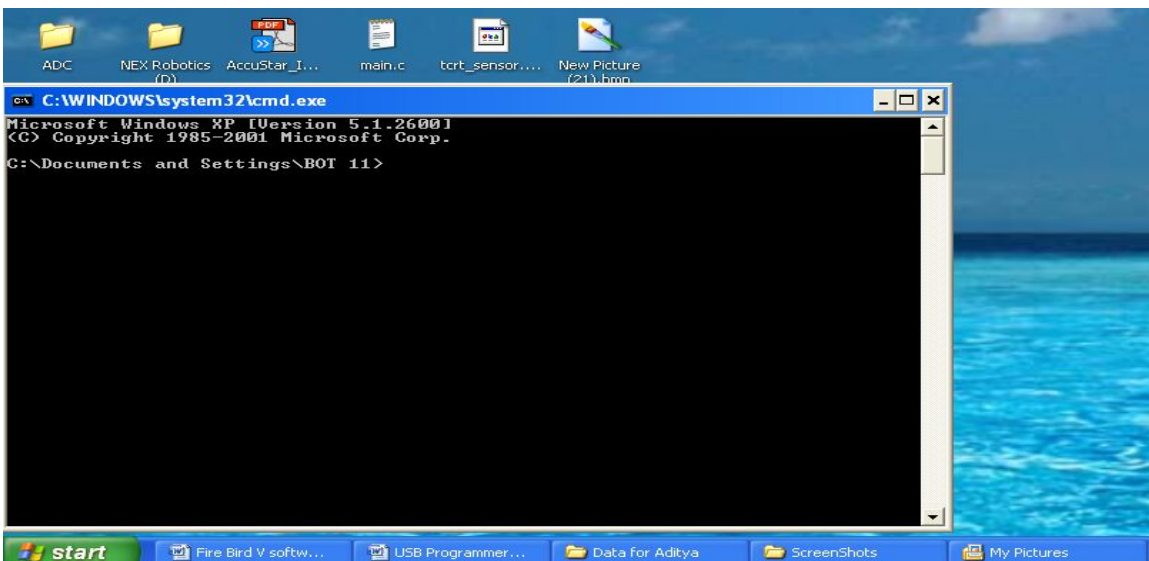
Step 6: Go to Start.



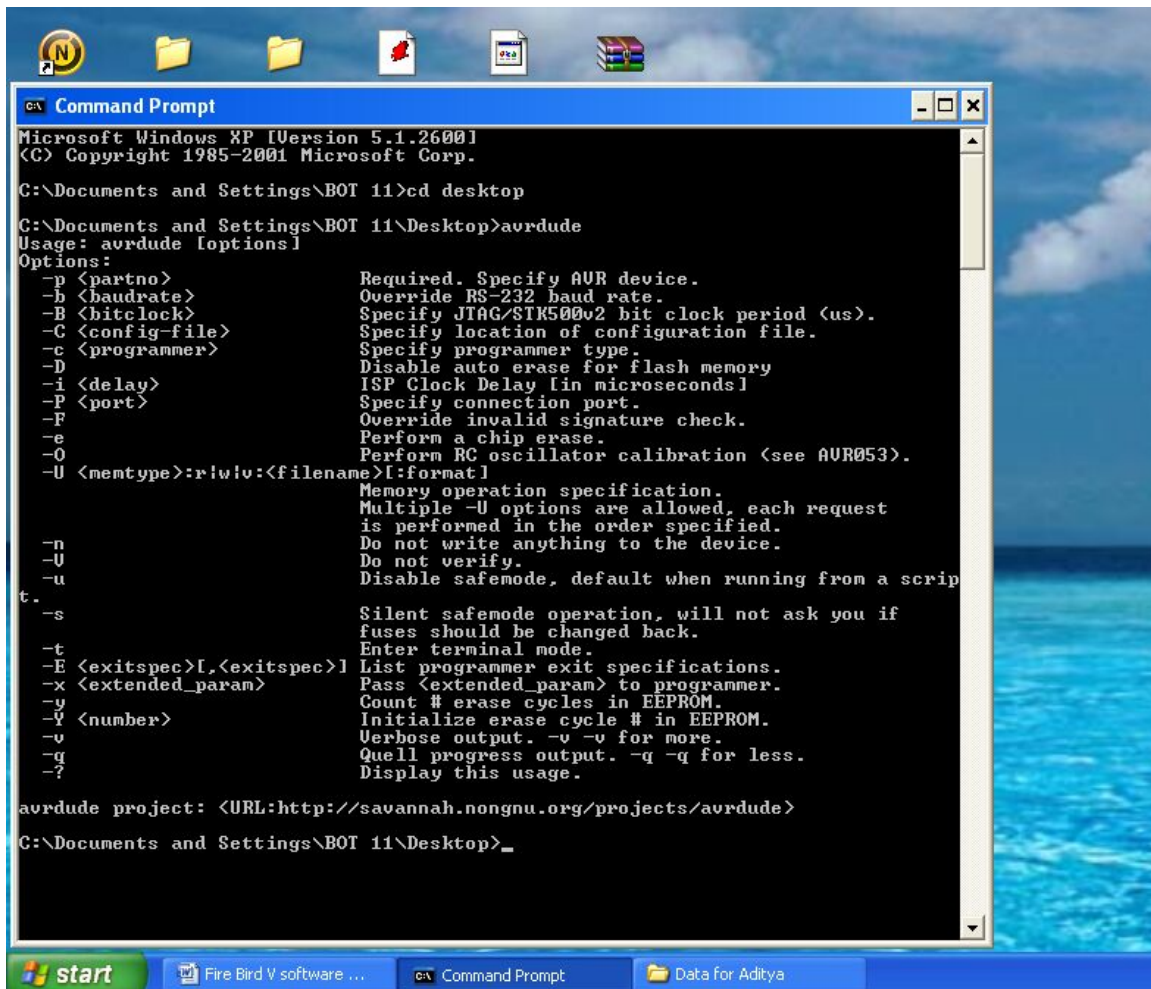
Step 7: Click run Type 'cmd' and hit enter.



Step 8: The command window appears.



Step 9: Type “avrdude” and press enter to see the list of commands available. You can also refer to the avrdude-doc-5.5.pdf file provided in the documentation CD for detailed information about various command line operations available with avrdude.

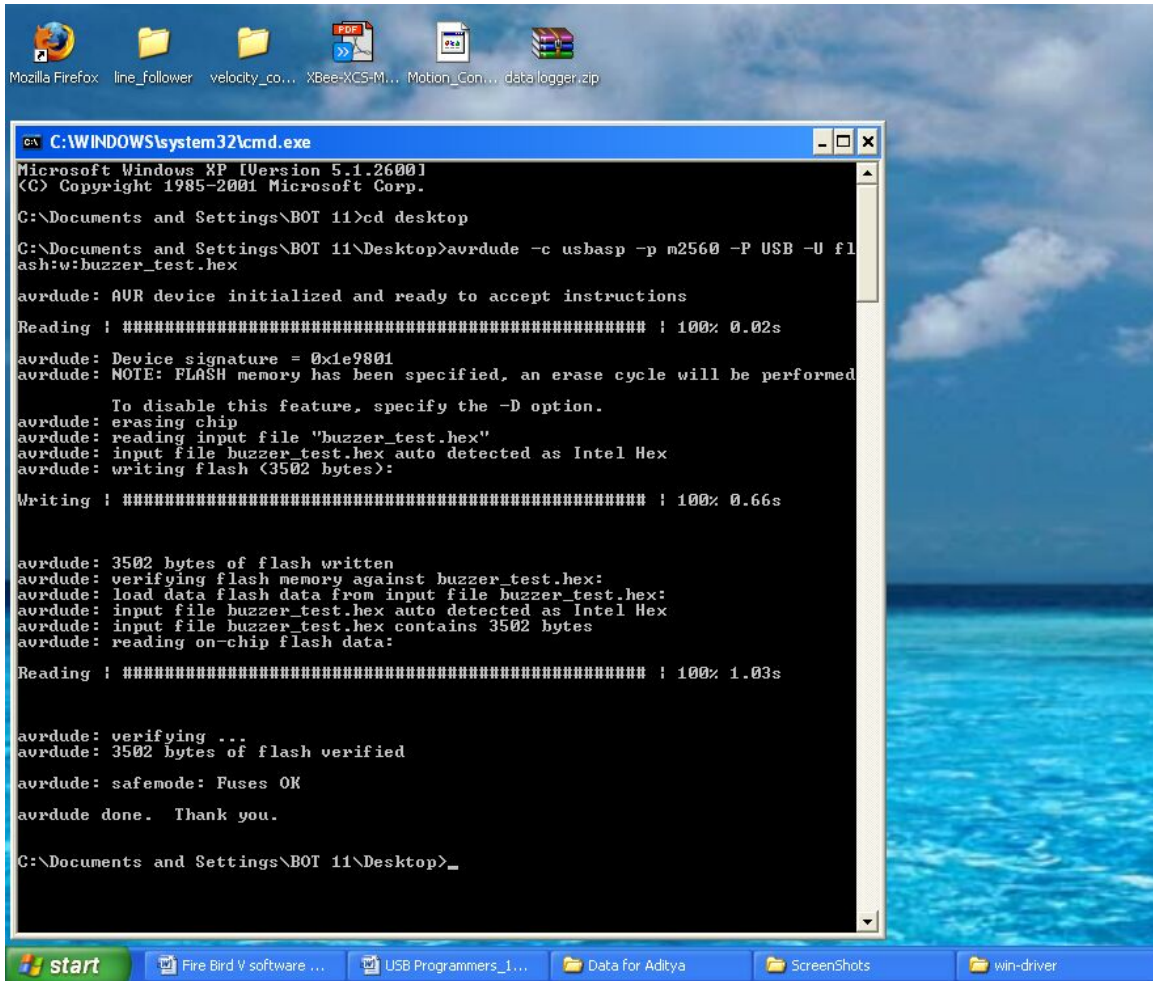


```

C:\Documents and Settings\BOT 11>cd desktop
C:\Documents and Settings\BOT 11\Desktop>avrdude
Usage: avrdude [options]
Options:
  -p <partno>           Required. Specify AVR device.
  -b <baudrate>         Override RS-232 baud rate.
  -B <bitclock>         Specify JTAG/STK500v2 bit clock period <us>.
  -C <config-file>      Specify location of configuration file.
  -c <programmer>       Specify programmer type.
  -D                    Disable auto erase for flash memory
  -i <delay>            ISP Clock Delay [in microseconds]
  -P <port>             Specify connection port.
  -F                    Override invalid signature check.
  -e                    Perform a chip erase.
  -O                    Perform RC oscillator calibration (see AVR053).
  -U <memory>[:r[:w[:v[:<filename>[:format]]]]]
                        Memory operation specification.
                        Multiple -U options are allowed, each request
                        is performed in the order specified.
  -n                    Do not write anything to the device.
  -U                    Do not verify.
  -u                    Disable safemode, default when running from a scrip
t.
  -s                    Silent safemode operation, will not ask you if
                        fuses should be changed back.
  -t                    Enter terminal mode.
  -E <exitspec>[f,<exitspec>] List programmer exit specifications.
  -x <extended_param>   Pass <extended_param> to programmer.
  -y                    Count # erase cycles in EEPROM.
  -Y <number>           Initialize erase cycle # in EEPROM.
  -v                    Verbose output. -v -v for more.
  -q                    Quell progress output. -q -q for less.
  -?                    Display this usage.

avrdude project: <URL:http://savannah.nongnu.org/projects/avrdude>
C:\Documents and Settings\BOT 11\Desktop>_
  
```

Step 10: Type commands for avrdude to program a specific device. (Refer to the examples given)



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\BOT 11>cd desktop
C:\Documents and Settings\BOT 11\Desktop>avrdude -c usbaspp -p m2560 -P USB -U flash:w:buzzer_test.hex

avrdude: AVR device initialized and ready to accept instructions

Reading : ##### : 100% 0.02s
avrdude: Device signature = 0x1e9801
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
        To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "buzzer_test.hex"
avrdude: input file buzzer_test.hex auto detected as Intel Hex
avrdude: writing flash (3502 bytes):

Writing : ##### : 100% 0.66s

avrdude: 3502 bytes of flash written
avrdude: verifying flash memory against buzzer_test.hex:
avrdude: load data flash data from input file buzzer_test.hex:
avrdude: input file buzzer_test.hex auto detected as Intel Hex
avrdude: input file buzzer_test.hex contains 3502 bytes
avrdude: reading on-chip flash data:

Reading : ##### : 100% 1.03s

avrdude: verifying ...
avrdude: 3502 bytes of flash verified

avrdude: safemode: Fuses OK
avrdude done. Thank you.

C:\Documents and Settings\BOT 11\Desktop>_
  
```

Eg 1. Transfer a file called example1.hex present on Desktop to a Mega128 device.

Type `avrdude -c usbaspp -p m128 -P usb -U flash:w:example1.hex`

and wait for the program to upload.