AVRUP V1 User Manual

Joe Pardue March 27, 2010

AVRUP V1 was used in the April 2010 Nuts&Volts Smiley's Workshop 21 column. It is based on C# code shown in earlier workshops. You can learn all the details for building code like this from my book *Virtual Serial Port Cookbook* and projects kit available from www.smileymicros.com, Nuts&Volts, and Amazon (book only).

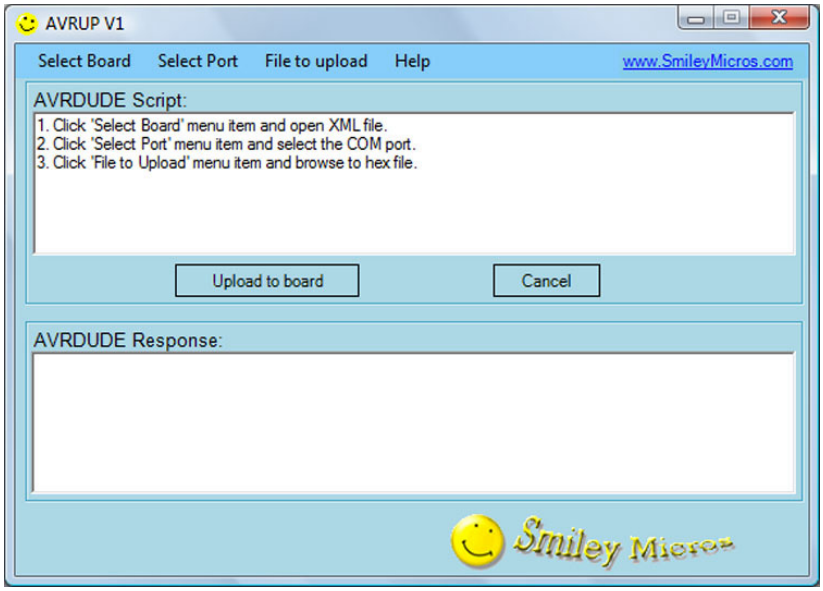
This program was written to provide an alternative to using AVRProg from AVRStudio or to using avrdude from the command line. AVRProg is no longer supported by Atmel and IMHO is somewhat flaky and provides little information if there is a problem either connecting to the AVR or uploading the program. Avrdude is a great tool, but requires using a command line window where you either type or paste in the command line. Some folks find this tedious and error prone.

It provides an autoreset feature that toggles the serial DTR and RTS lines that are used by some devices, such as the Arduino.

AVRUP V1 is available as both an executable and source code from the downloads link at www.smileymicros.com. The source code was taken form the Developer Terminal, and there are many sections of the code that are either commented out or turned off in V1 that may be activated in future versions (translation: it is junky and not well planned).

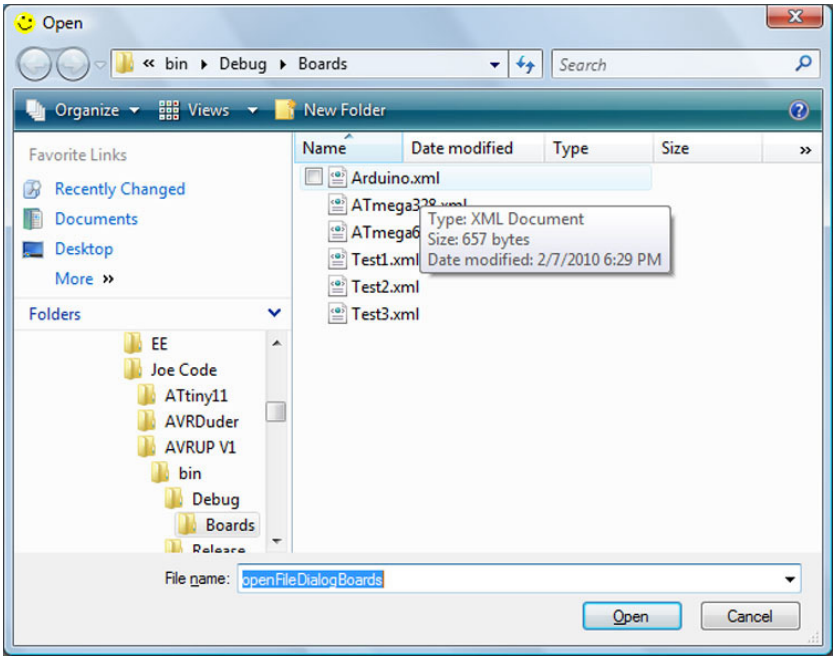
When you open AVRUP V1 the AVRDUDE Script window lists the basic instructions:

```
1. Click 'Select Board' menu item and open XML file.
2. Click 'Select Port' menu item and select the COM port.
3. Click 'File to Upload' menu item and browse to hex file.
```

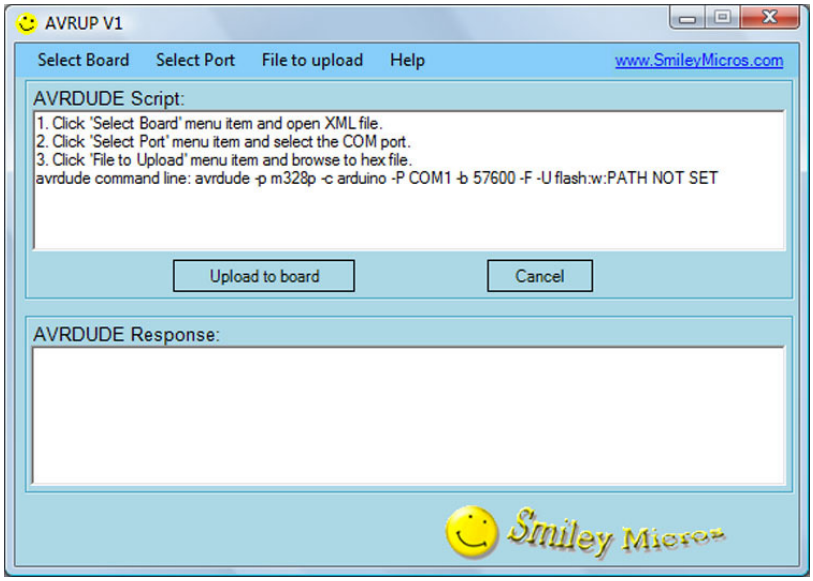


1. Click 'Select Board' menu item and open XML file.

Several XML files are included that contain data for the avrdude script. Use the file window to browse to the file corresponding to your device. If you decide to modify the XML, be sure and make a backup copy. XML files must be perfect to work and won't give any hints if you mess up a line. I recommend using Programmer's Notepad, which comes with the WinAVR installation, to edit these files.

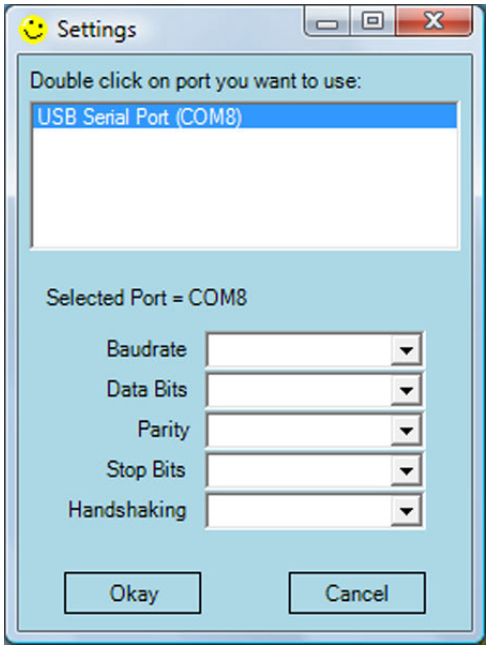


The figure below shows the avrdude command line that results from choosing the Arduino.xml file. Note that the COM1 is a default setting and probably not the correct port, and that the hex file to upload path is not set. These are taken care of in steps 2 & 3.



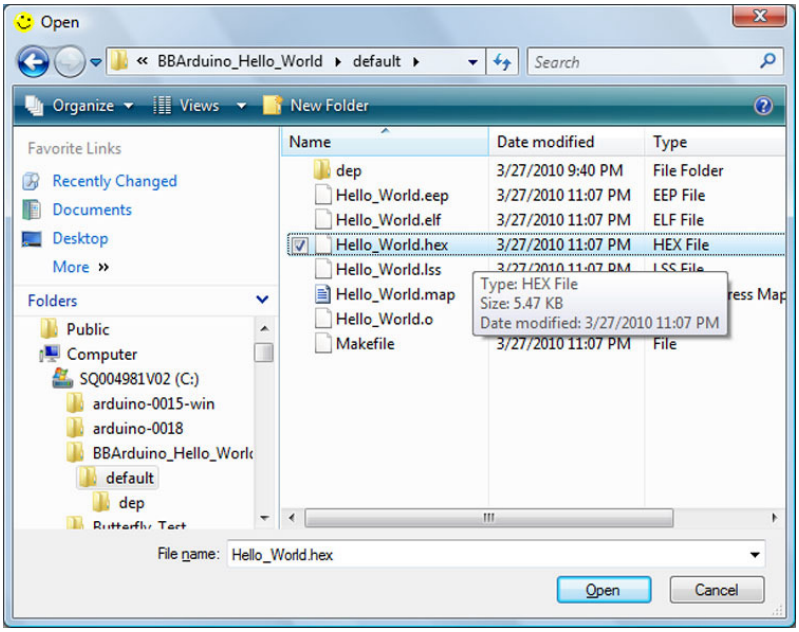
2. Click 'Select Port' menu item and select the COM port.

The Settings dialog is the same as the one used in Developer Terminal except that it is only used to select the COM port. The modem parameters are set in the XML files. You must click on the COM port, in the case shown below COM8 (yours will likely be different) and make sure that the 'Selected Port = COM8' shows that you have actually selected the port.

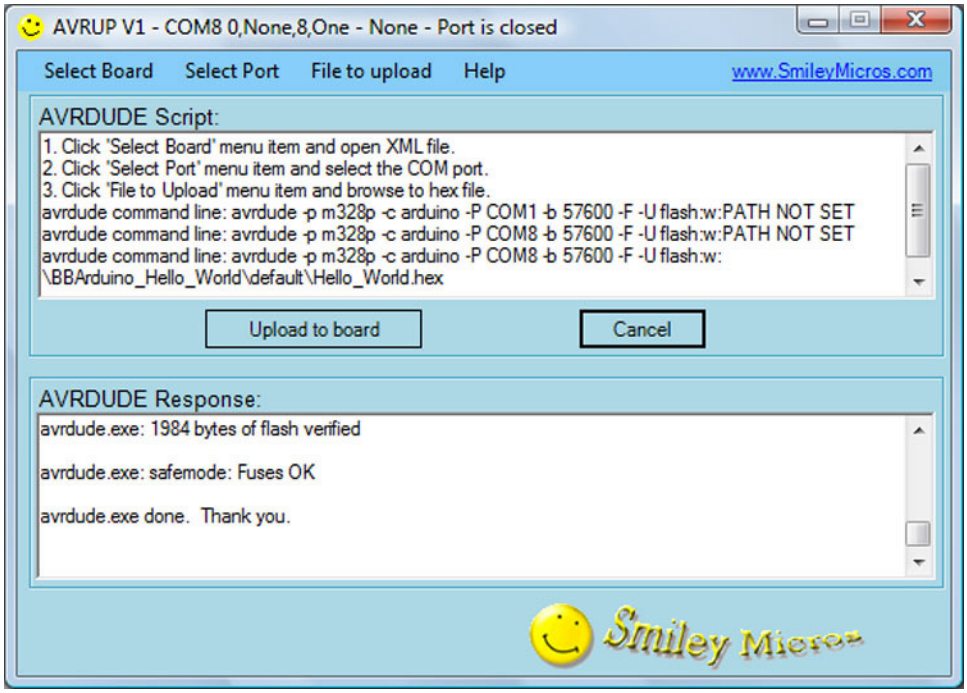


3. Click 'File to Upload' menu item and browse to hex file.

The File to Upload button opens another file browser window that lets you select the hex file to upload. Remember that the hex file must be compiled for the device you are about to upload it to. In the example below, this file was compiled for Arduino (the actual device tested is an ATmega328 on a breadboard, but it uses the standard Arduino bootloader).



Verify that the avrdude command line has all the information you want to send to avrdude.
>If you have a device that uses autoreset from either DTR or RTS, the AVRUP V1 will automatically toggle those lines for you when you click the 'Upload to board' button.
>If you are not using a device with autoreset, then you must reset the device immediately before clicking the 'Upload to board' button.



The following is a listing of an avrdude response for a successful upload:

```
Started function.  Please stand by..

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100%
0.02s

avrdude.exe: Device signature = 0x1e950f
avrdude.exe: NOTE: FLASH memory has been specified, an erase cycle will
be performed
             To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file
"\BBArduino_Hello_World\default\Hello_World.hex"
avrdude.exe: input file \BBArduino_Hello_World\default\Hello_World.hex
auto detected as Intel Hex
avrdude.exe: writing flash (1984 bytes):

Writing | ################################################## | 100%
1.01s

avrdude.exe: 1984 bytes of flash written
avrdude.exe: verifying flash memory against
\BBArduino_Hello_World\default\Hello_World.hex:
avrdude.exe: load data flash data from input file
\BBArduino_Hello_World\default\Hello_World.hex:
avrdude.exe: input file \BBArduino_Hello_World\default\Hello_World.hex
auto detected as Intel Hex
avrdude.exe: input file \BBArduino_Hello_World\default\Hello_World.hex
contains 1984 bytes
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100%
0.87s

avrdude.exe: verifying ...
avrdude.exe: 1984 bytes of flash verified

avrdude.exe: safemode: Fuses OK

avrdude.exe done.  Thank you.
```

If there is a problem then the listing may provide some helpful debugging information. You can find the avrdude manual in the doc directory or you WinAVR installation.

I tested this code on an Arduino, AVR Butterfly, and ATmega644 on a breadboard. It worked for me.

BUT REMEMBER – this is a beta release – it will have bugs – you have the source code – if you want to help the community then ask questions and provide bug fixes on www.avrfreaks.net.