The Basics of AVR Studio 4

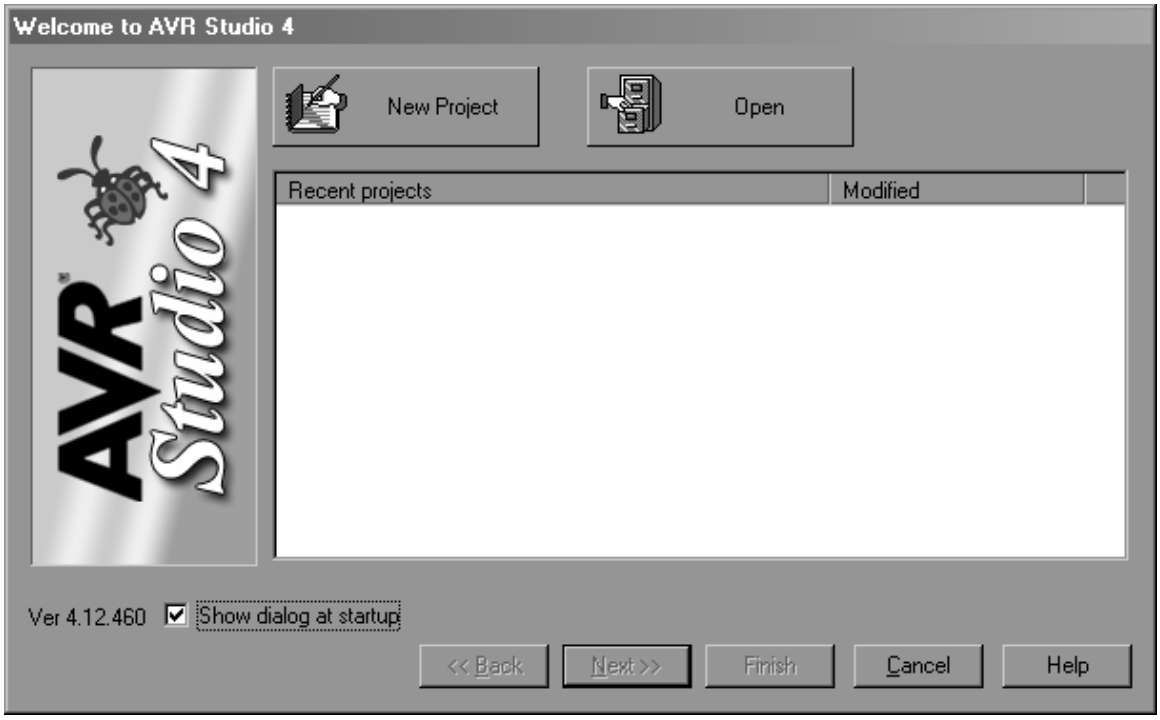
This tutorial will show you the basics of AVR C programming using AVR Studio 4. This guide will show you how to setup a project, insert some code, how to compile it, how to switch to using the JTAG, and some basics on debugging.

Setting up a project

1.) Open AVR Studio 4

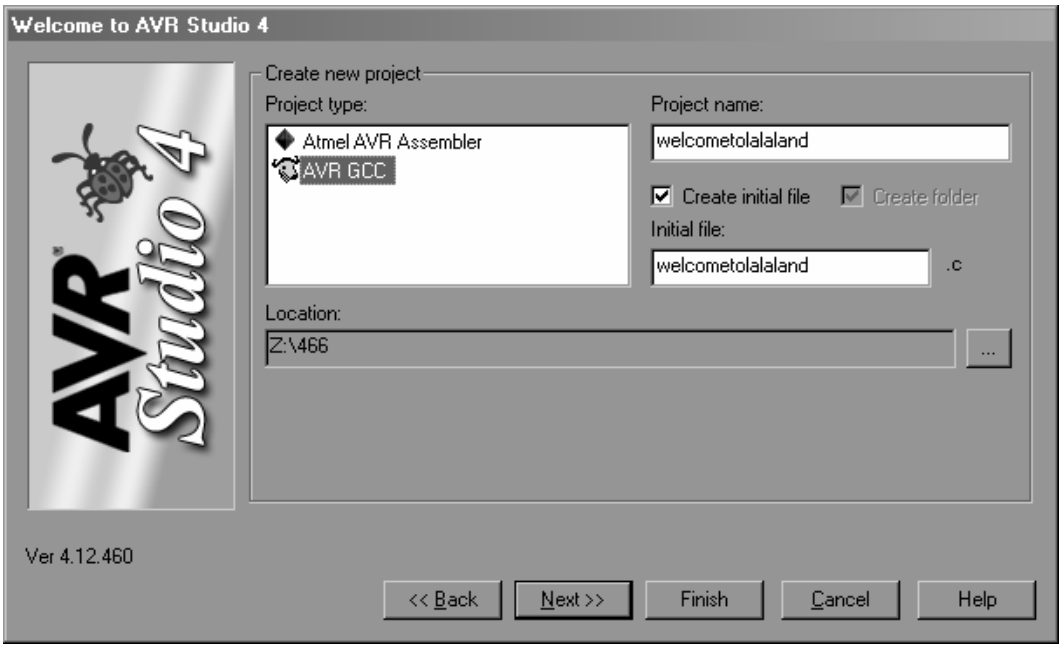
Start → Programs → Atmel AVR Tools → AVR Studio 4

You should then see this window:

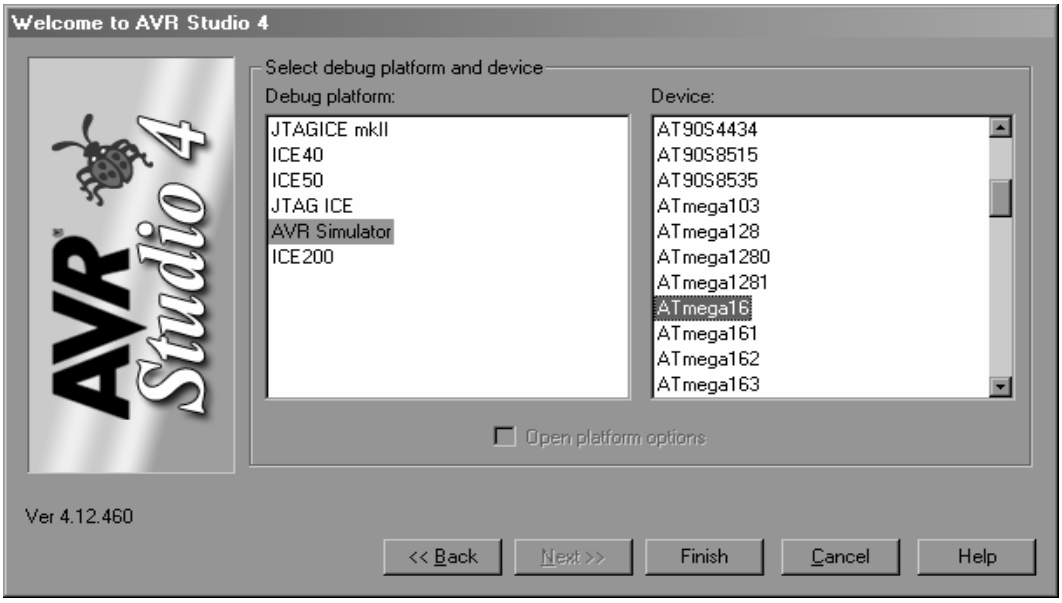


2.) Click New Project

3.) Select AVR GCC and enter a project name, placing it in an appropriate place (Z:\466 is shown here). Click Next.

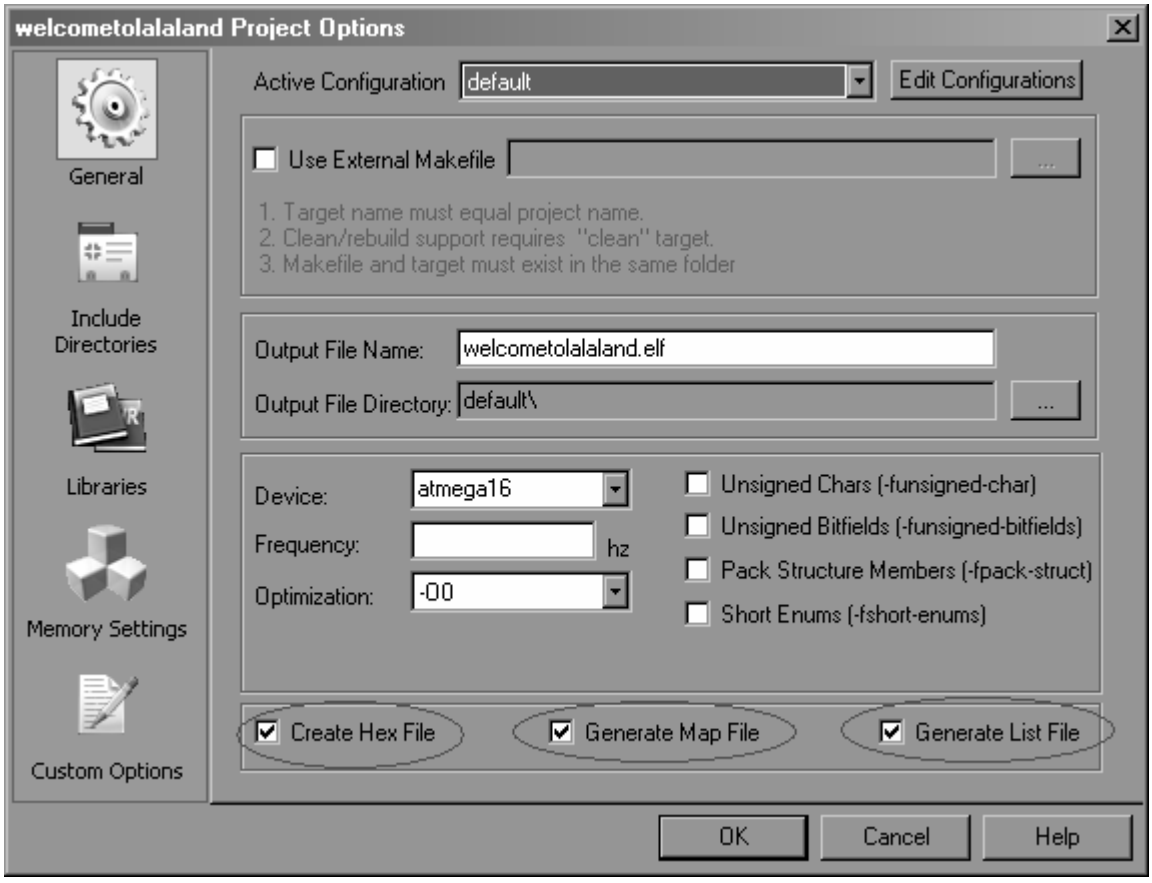


4.) Select AVR Simulator as your Debug platform and the ATmega16 as your Device. Click Finish.



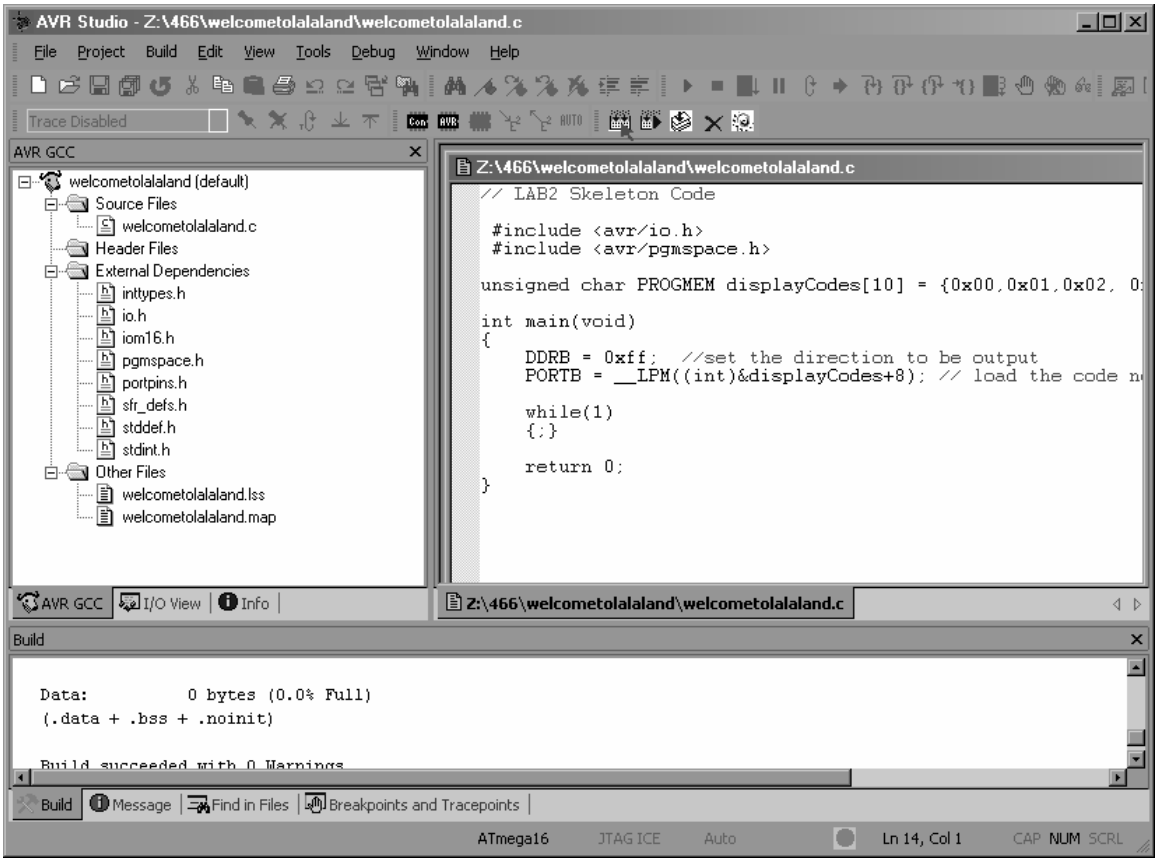
5.) Next go to Project → Configuration Options

Under the General Tab make sure that Create Hex File, Generate Map File, and Generate List File are checked



Inserting some code and compiling it

Compiling in AVR studio is easy. First enter your code then click the build button. If information on the AVR memory doesn't come up after the build, then the build probably failed.



If you want to change any of the compile options, go to

Project→ Configuration Options

Things like the optimization flags and whether chars are signed or unsigned may be found in the General Tab.
More specific compile flags may be entered in the Custom Options Tab.

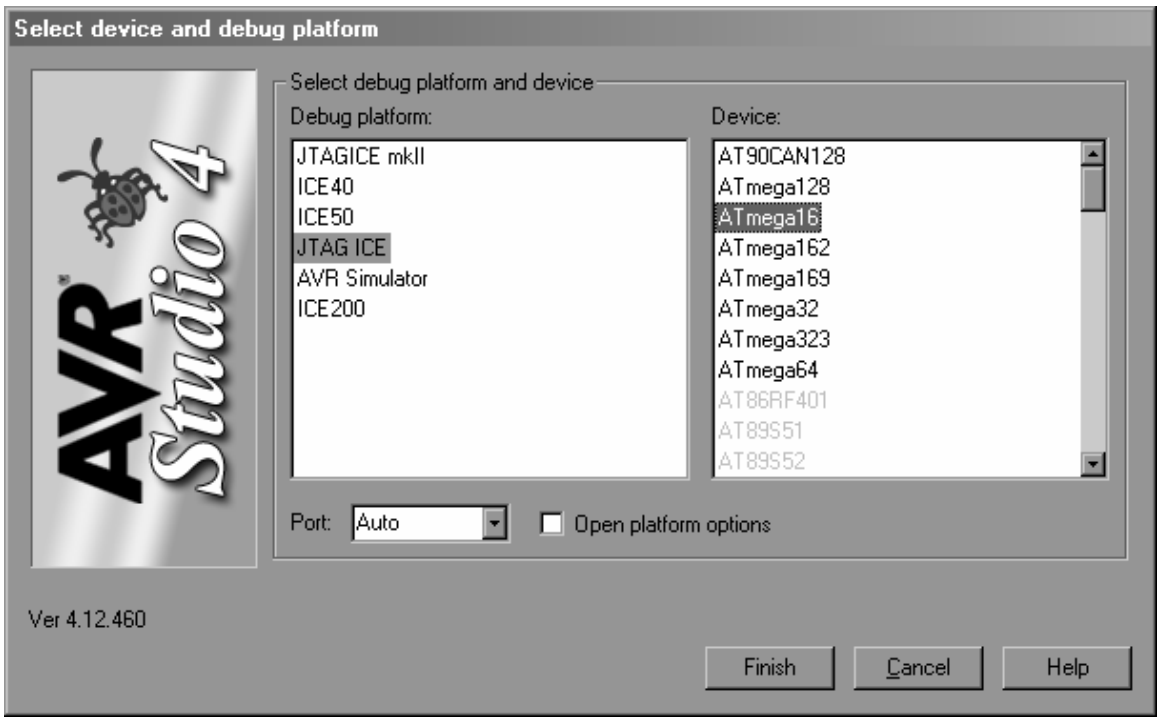
Setting up the JTAG

To switch from Simulation to the JTAG, follow these steps:

1.) From the main screen go

Debug → Select Platform and Device...

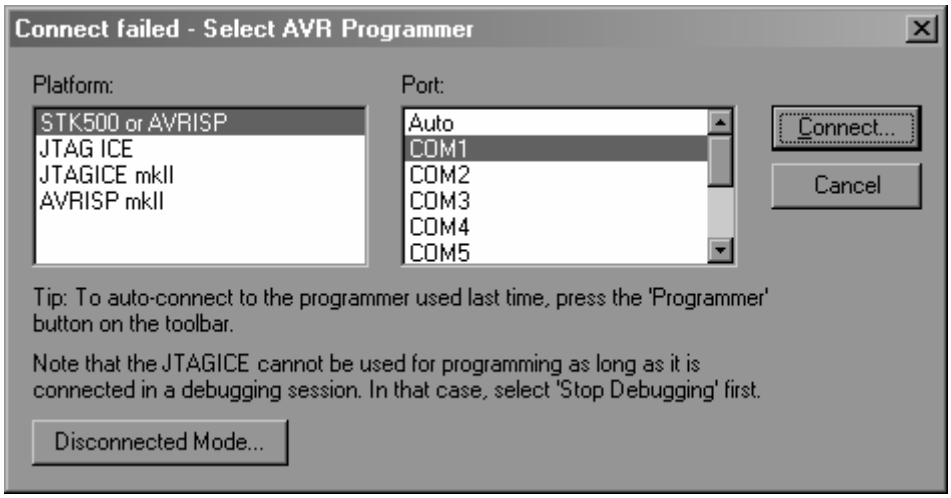
2.) Select JTAG ICE or JTAG ICE mkII depending on which device you are using (look for the label on your device) and ATmega16



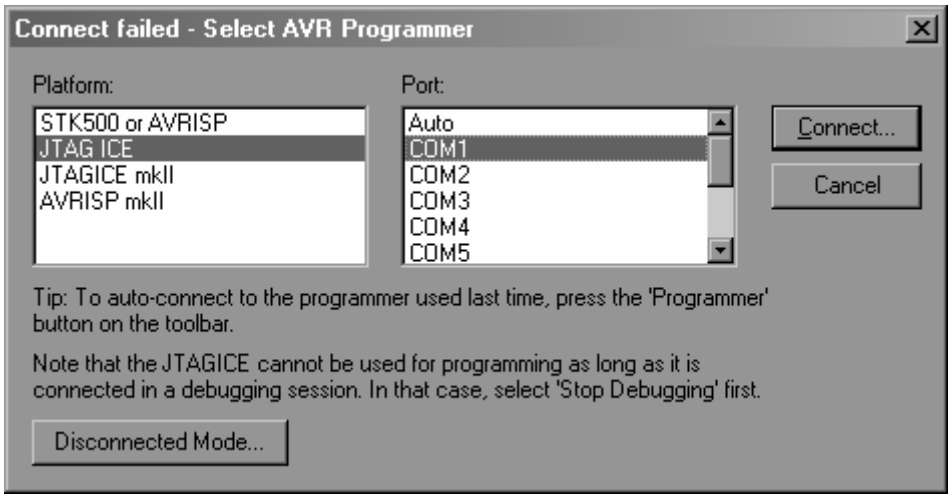
3.) Click Finish.

Using the JTAG to program the ATmega16

1.) Build your project
2.) Click the AVR button 7 buttons to the left of the Compile button. It looks like this: 
3.) This window should pop up:



Select JTAG ICE or JTAG ICE mkII and COM 1



Click Connect.

Basics of Debugging

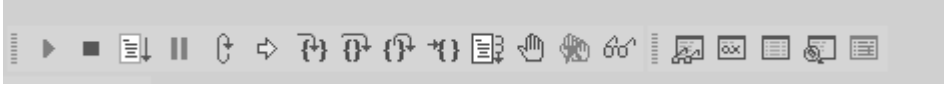
Debugging is the same whether in software or hardware. The only difference is that you must select your device as either AVR Simulator (software), or JTAG ICE (or JTAG ICE mkII) (hardware).

The interface is much the same as Visual Studio or any other debugging interface you may have used. You can set breakpoints, add variables to a watch list, etc.

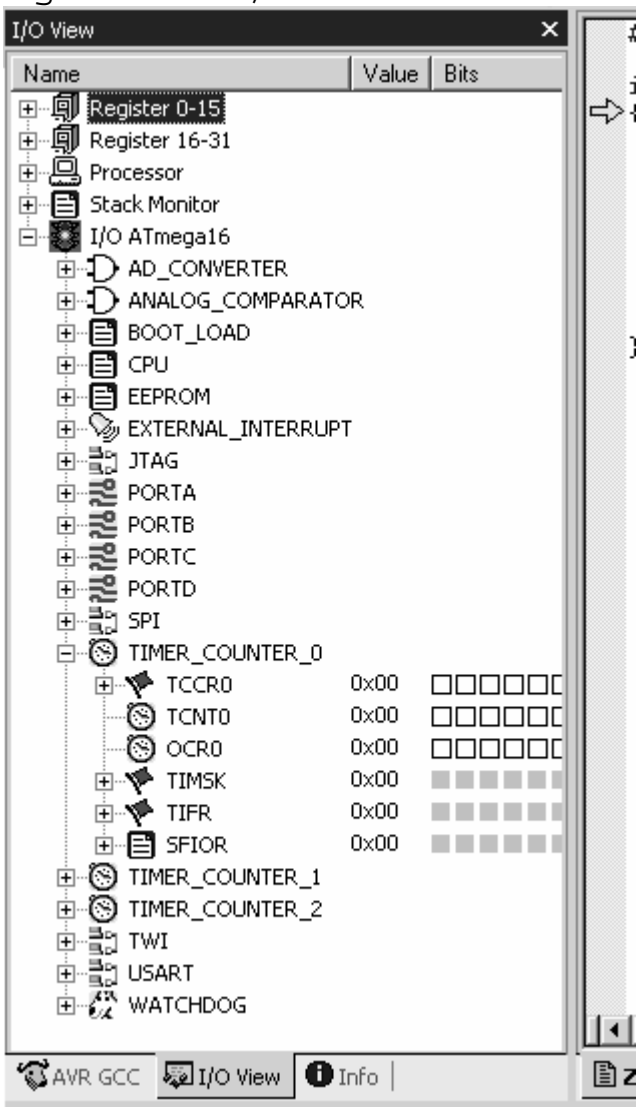
AVR Studio 4 has many interactive debugging features. For the purposes of this class the two basic methods you will use to debug are the AVR Simulator and the JTAG. At the bottom of your screen AVR Studio displays what device/chip it will interact with when debugging and which debugger it will use. Below is a picture of the Status Bar showing that the device is an "ATMega16" and the debug platform is the "AVR Simulator".

To change the target device for debugging and/or the debug platform go to the "Debug" menu and select the menu option "Select device and debug platform." In this window you can select AVR simulator if you want to debug in simulation or if you want to debug directly on your hardware choose either JTAG ICE or JTAG ICE mkII (depending on which JTAG you have at your station).

The debugger in AVR studio runs similar to other development environments. Besides the options in the debug menu there is a toolbar (shown below) to control the debugger. For more information on how to control the debugger refer to the AVR Studio Help Menu.



One of the most powerful features of the debugger is the ability to view the current states of any of the registers. You can view the current value of any register in the "I/O view" tab.



Another useful feature of the AVR Studio 4 debug system is the watch window which allows you to see what the value any variable is in the RAM. To view the current value of a variable use, add the variable to the watch window.