Getting Started with STK200 Dragon

Introduction

This guide is designed to get you up and running with main software and hardware. As you work through it, there could be lots of details you do not understand, but these are covered in books and other documentation that you can read later.

The main software package is AVRStudio, which is Atmel's Integrated Development Environment or IDE. There is also a plugin C Compiler module called WinAVR that compiles C programs within AVRStudio. The Kanda installer will install these packages and copy documentation to a folder on your hard drive.

Default install path is **C:\Program Files\STK200 Dragon**

AVRStudio has its own Atmel-AVR Tools folder in Program Files, and can be run from there or you can run from desktop icon.

WinAVR never needs to be run directly, only from AVRStudio.

The AVRStudio IDE is designed for writing source code, in C (.c files) or assembler (.asm files). These are then built or compiled into object code (.hex files) for programming into the AVR using ISP or debug files (.elf files) to step through the code.

The AVR Dragon hardware is a programmer and In Circuit Emulator (ICE) in one tool. Once source code is built, it can be programmed into the AVR using ISP, which will just run the code or it can be set in Debug mode (using DebugWire or JTAG)that allows you to examine the code operation to find bugs.

There are two Debug methods depending on the AVR device you are using

- **JTAG Mode** for AVR devices with 40-pins or more
- **DebugWire** for smaller pin-count devices.

In System Programming is common to all AVR devices, using serial programming (**ISP**). Devices with JTAG can be programmed through JTAG as well as ISP.

This guide describes the hardware setup for ISP, DebugWire and JTAG, with associated AVRStudio software operations. The default AVR supplied is an Atmega16. As this is a 40-pin device, it uses JTAG for debugging. Although it can also be programmed through JTAG, we suggest you use ISP first to familiarise yourself with standard programming.

There is a brief guide to the books and documentation at the end of this guide as well as links to Atmel site, where more information is available about AVR, AVRStudio and AVR Dragon.

Getting Started with STK200 Dragon

Creating an AVR Dragon Project

AVR Studio

AVR Studio is Atmel's development environment. The Kanda installer will install AVRStudio, service packs for AVRStudio and WinAVR C Compiler, unless you choose not to because you have them already.

This is a brief guide to using one of the sample files in AVR Studio, but more information is available in the AVRStudio Getting Started guide and at www.atmel.com

AVR Studio is project based, so you need to create a project before you can do anything. Project extension is .aps. Follow this procedure.

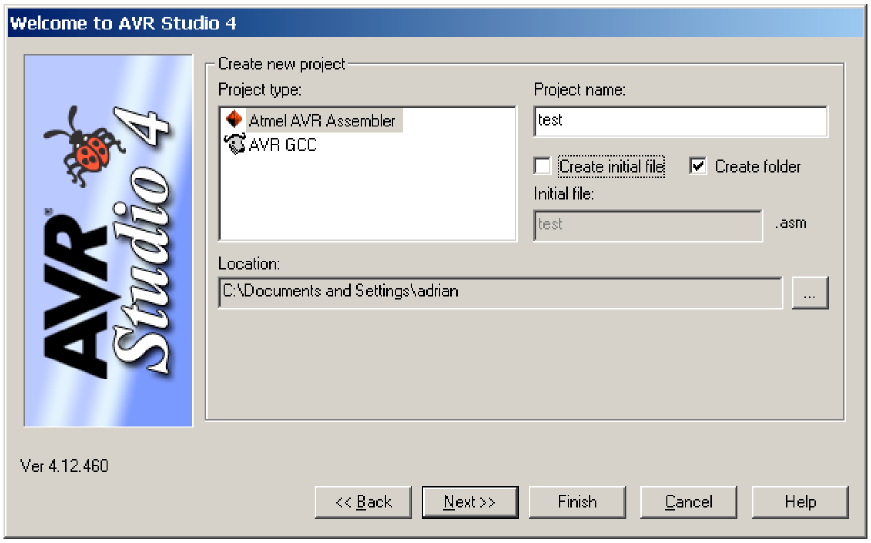
1) Run AVRStudio and select **New Project** from Welcome screen. If Welcome screen does not appear, select **Project Menu > Project Wizard**

2) Choose Atmel AVR Assembler as Project Type or AVR GCC for C Projects.

3) Give the Project a name

4) Uncheck Create Initial File box

5) Set location of where to save Project files



Welcome to AVR Studio 4

Create new project

Project type: Atmel AVR Assembler, AVR GCC

Project name: test

☐ Create initial file ☑ Create folder

Initial file: test .asm

Location: C:\Documents and Settings\adrian

Ver 4.12.460

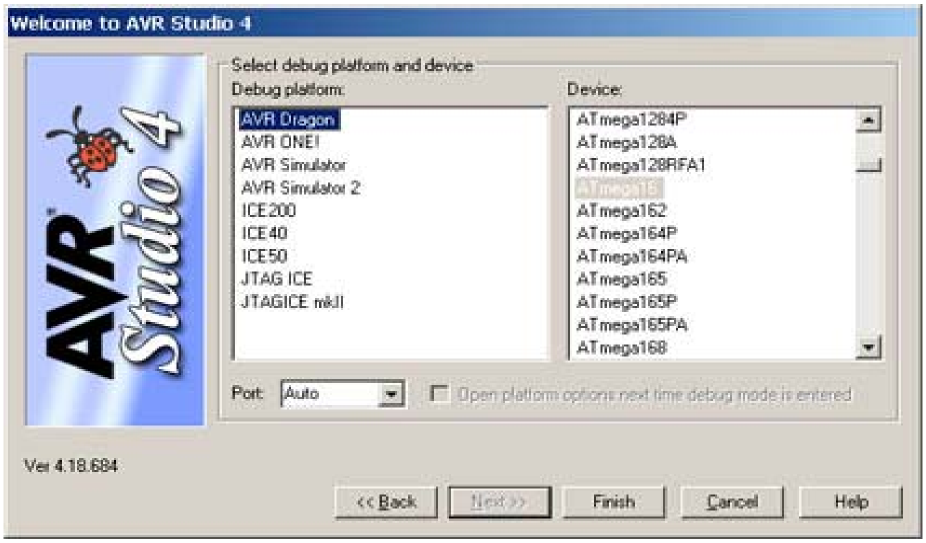
6) Click Next Button

On the next screen,

1) Set Debug platform to AVR Dragon

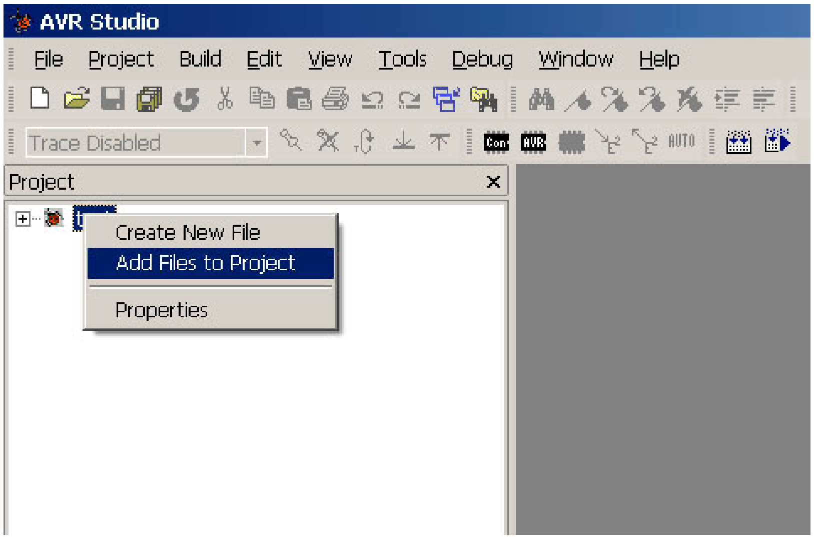
2) Set Device to ATmega162, Atmega16 or your own AVR device

3) Click Finish Button



Now the project is created, we need to add a sample file.

1) The Project is displayed at the left of the screen. Right click on the Project name and choose Add Files to Project



2) Select an assembler file from Assembler Sample Code folder, such as LedFlash200.asm. See C Code section for using C language.

3) Now go to **Build Menu > Build**. This will assemble the file and create object code file (.hex) for programmer and Elf file for debugger in Project folder (or sub-folder called Default for C Projects).

Once the source file has been built without errors, you can program it into your target AVR so it runs or use Debug Mode for debugging it. The next step is to set up the hardware for programming (ISP) and DebugWire or JTAG for debugging.

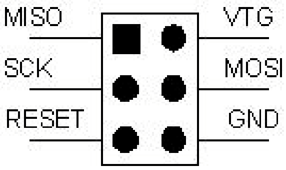
Hardware Connections for ISP and DebugWire

Using ISP

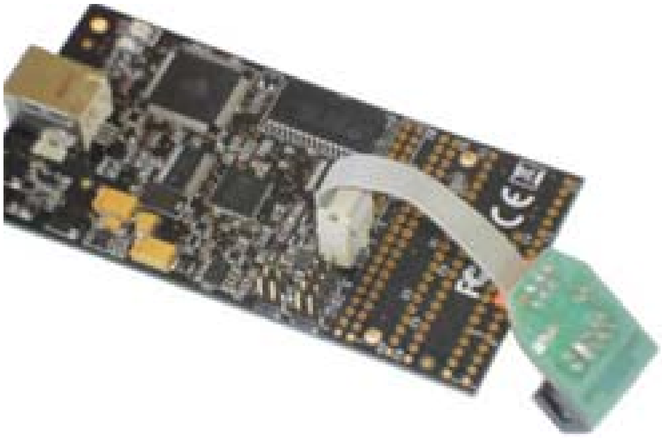
DebugWire is used for debugging smaller pin count devices from AVRStudio. ISP or In System Programming is used for programming devices from AVRStudio, or another programmer.

Connecting AVR Dragon

There is a 6-pin pin-header on AVR Dragon board labeled ISP.



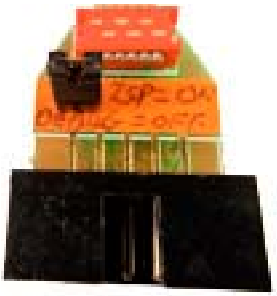
Connect the 6-way adapter to this header as shown, with key way facing prototype area.



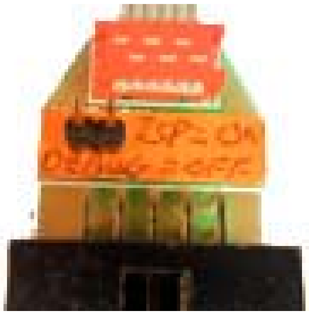
There is a jumper on adapter PCB. This should be **ON** for **ISP** and **OFF** for **DebugWire (DW).** It is supplied with jumper on.

Side Note: ISP will not work if jumper is off. DebugWire will work with jumper on, but some of PortB (LEDS) will not be available as ISP lines will be connected, on PortB, bits 5-7.

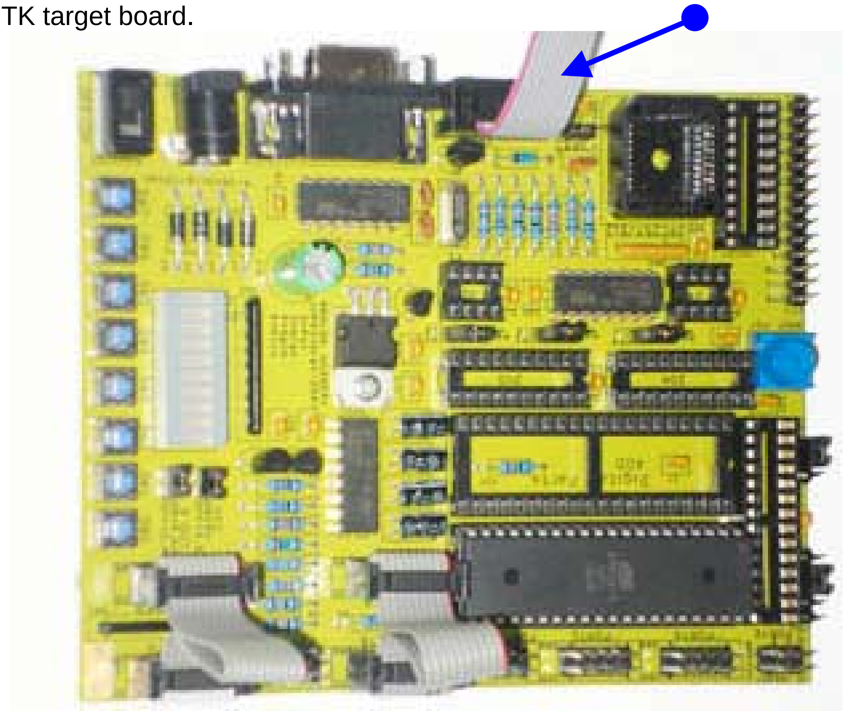
ISP : Jumper ON



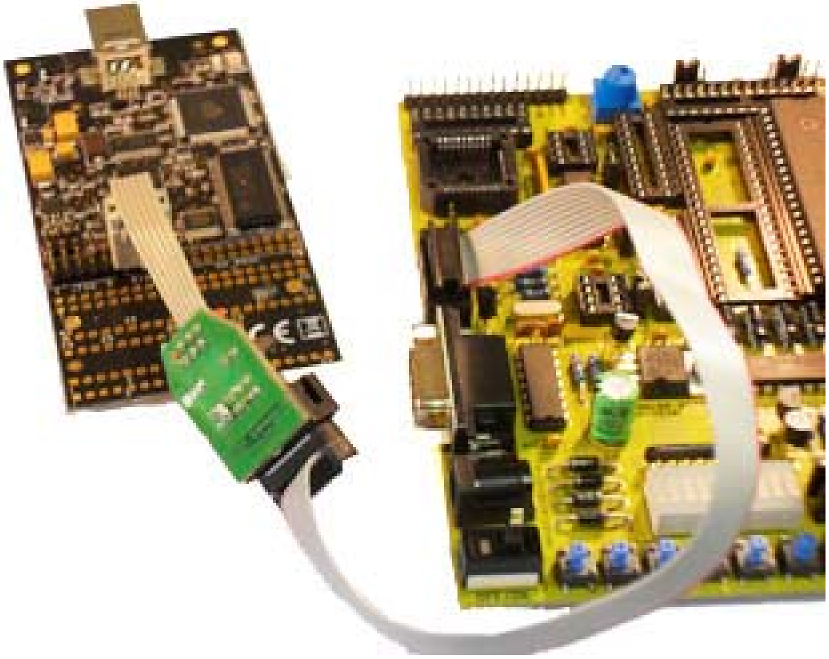
DebugWire : Jumper OFF



Connect the 10-way ribbon cable to this adapter and into the 10-way box header on STK target board.



The complete ISP or DebugWire connection looks like this:



Connect the AVR Dragon to a USB port on your computer. The New Hardware Wizard will appear, just follow instructions to install driver.

Powering STK200 Board

STK200 Board Power Connection

The AVR Dragon is self-powered from USB bus, and in most circumstance it can also be used to power the STK200 board using the two wires supplied. Unless you add extra circuitry, the STK200 board will draw less than the 100mA that the USB can supply.

If you have any doubts about your USB, then do not power the board from AVR Dragon. The alternative is to connect a power supply to the power connector on the STK200 board.

- 2.1mm barrel connector
- Centre positive or negative
- 9-15VDC or 6-12VAC

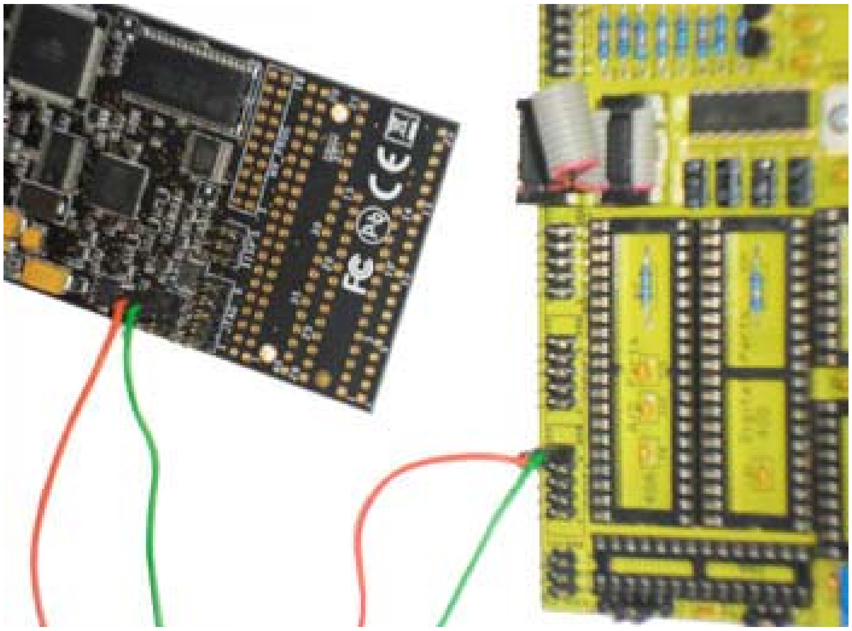
STK200 Board Power from AVR Dragon

The AVR Dragon has a 6-way connector marked VCC and GND, next to JTAG connector. It has 3 VCC and 3 GND pins. Connect the green wire supplied to one of the GND pins, and the red wire to one of the VCC pins.

All the port headers on the STK200 board also have a GND and VCC pin. It does not matter which header you use but Port A is most convenient. Connect the green wire to GND pin on Port A header (inside) and red wire to VCC pin (outside).

Side Note: There are also VCC and GND pins on 14-way LCD header on STK200 board, if it is easier to use these instead in your setup. GND wire can be omitted as JTAG or ISP headers have GND line connected already but board will only be powered if cable is left connected.

This picture shows the completed connections.

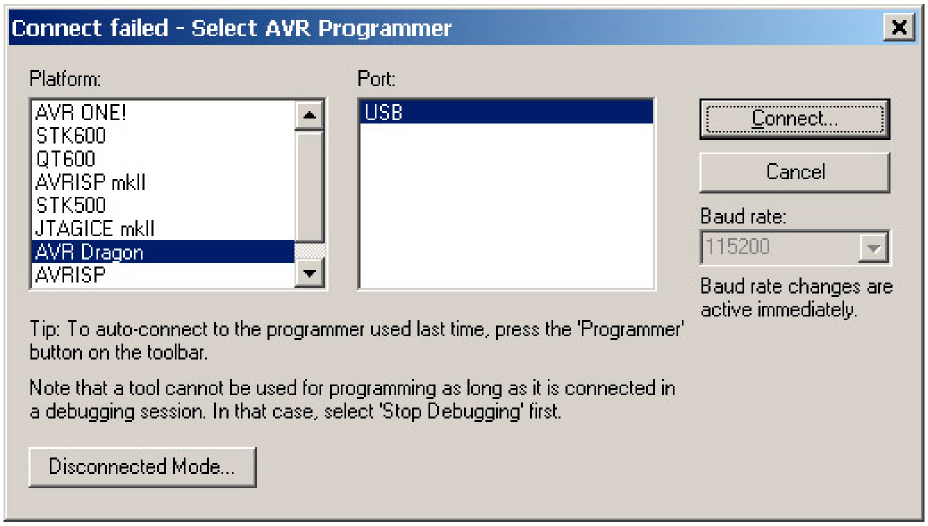


Programming from AVRStudio

Make sure hardware is connected as shown above for ISP and DebugWire.

Go to **Tools Menu > Program AVR > Auto Connect**.

This will bring up programmer dialog box. If you get a Connect Failed, then this Dialog will appear.



Baud rate changes are active immediately.

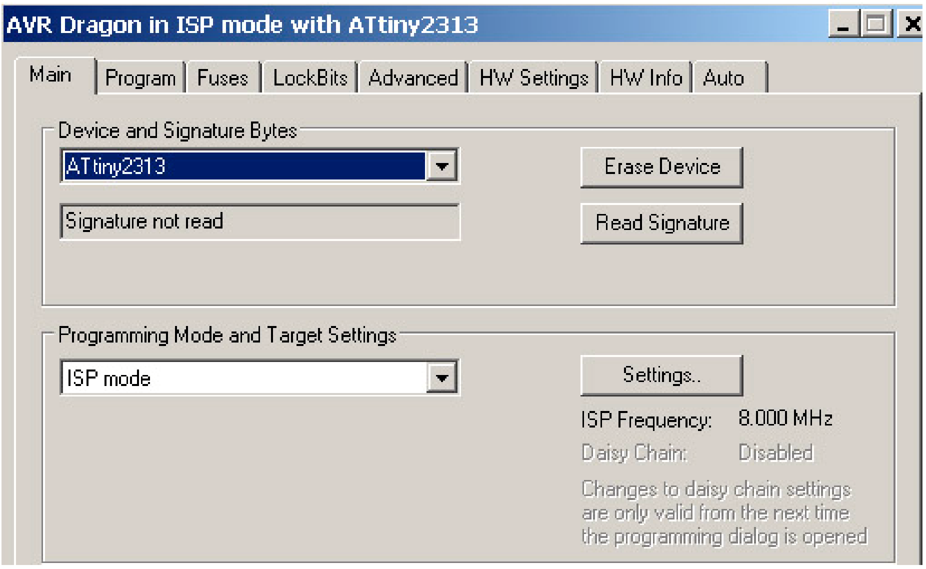
Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.

Note that a tool cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.

Make sure AVR Dragon is set as platform and your connections are all ok and try Connect button.

Once it has connected, you may get a Firmware Update message, depending on your version of AVRStudio. If you do, click Yes and continue to main programmer screen. This is described in AVRStudio Help files. There are a series of screens – **Main, Program, Fuses** and **Auto** tabs are the most important.

Main



Signature not read

ISP Frequency: 8.000 MHz

Daisy Chain: Disabled

Changes to daisy chain settings are only valid from the next time the programming dialog is opened

Make sure that **Programming Mode** is set to ISP mode. It may come up as JTAG mode on some AVR devices.
Side Note: If you want to program using JTAG, set Programming Mode to JTAG and use hardware JTAG connections – *see JTAG Connections section*

Check device type matches and **try Read Signature Byte.** If the result is 000000 or a signature mismatch, then ISP clock is probably too fast for a new AVR as default is 1MHz ISP speed. Click **Settings button** and write slower ISP speeds until it reads signature correctly.

Program Tab
An AVR device must be erased before it is programmed again, as a separate operation, so check Erase Device before Programming and Verify boxes. If Input Hex File is blank, set it to your output Hex file. Remember that C projects put output in a sub folder called Default, and Create Hex File must be checked in Project Options. Then click Program button in Flash section.

Fuses Tab
Fuses control the operation of the device. The most important at this stage are

- **SUT_CLKSEL** – Default varies with each AVR, but will always be Internal Resonator (INT RC), running at 1MHz or less. Change this to one of the External Crystal choices (External Crystal Osc. Or Resonator), 8MHz or High Frequency, at the bottom of the list. Exact choice does not matter.

- **CLKDIV8** – Some AVRs have this fuse to divide clock frequency by 8, so disable (uncheck) for real clock speed

- **JTAGEN** – on 40-pin devices, debug is via JTAG and this fuse should be enabled for Debugging. It will stop PortC acting as normal I/O, so disable it to use PortC

- **DWEN** – On AVRs that have DebugWire, this fuse should be disabled for ISP. The Debugger will set it when required for debugging

- **RSTDISBL** – On smaller pin count AVRs, this fuse disables Reset for use as general I/O. DO NOT set this fuse or you will not be able to use ISP

- **SPIEN** – Do not disable this fuse or ISP will not work

Set up these fuses and use **Program and Verify** buttons to set and check the fuses.

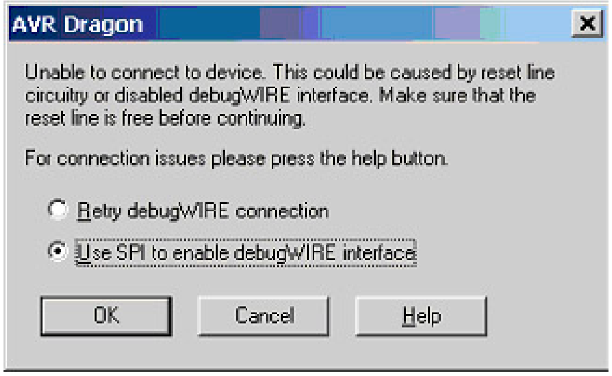
Auto
Allows you to do multiple operations by clicking Start button. Easiest way of reprogramming devices once fuses and other settings have been setup.

Once you are happy that you understand how to program a device, exit the Programmer screen. Now we will look at debugging. Depending on the AVR device in your kit, this will either be via DebugWire or JTAG. Read the appropriate section now, DebugWire for 8, 20 and 28-pin devices or JTAG for 40-pin devices.

DebugWire with AVR Dragon

DebugWire is the onboard debugger for smaller pin count AVR devices, with less than 40-pins. Hardware connections are same as already set for ISP, except jumper on adapter PCB should be OFF, *see ISP section*.

After building your project successfully, Choose **Debug > Start Debugging** or use **Build and Run**, which will launch AVR Dragon after compile. Unless you have set DWEN fuse before leaving ISP, you will get a Connection Failed dialog that prompts for setting DWEN.



Unable to connect to device. This could be caused by reset line circuitry or disabled debugWIRE interface. Make sure that the reset line is free before continuing.

For connection issues please press the help button.

○ Retry debugWIRE connection

● Use SPI to enable debugWIRE interface

This will enable DWEN and connect to your AVR. You will need to cycle the power on STK200 board before DWEN change takes effect.. A yellow arrow in source code window shows location in the code. You can then run or step though the code using Debug menu.

Important

Once you have finished debugging your code, you will need to clear **DWEN** again or you will not be able to use ISP with the device. Before exiting debug mode (Stop Debugging), you will need to disable DWEN. Choose **Debug > AVR Dragon Options** and click **Disable DebugWire** button. This will clear DWEN and exit debug mode.

JTAG with AVR Dragon

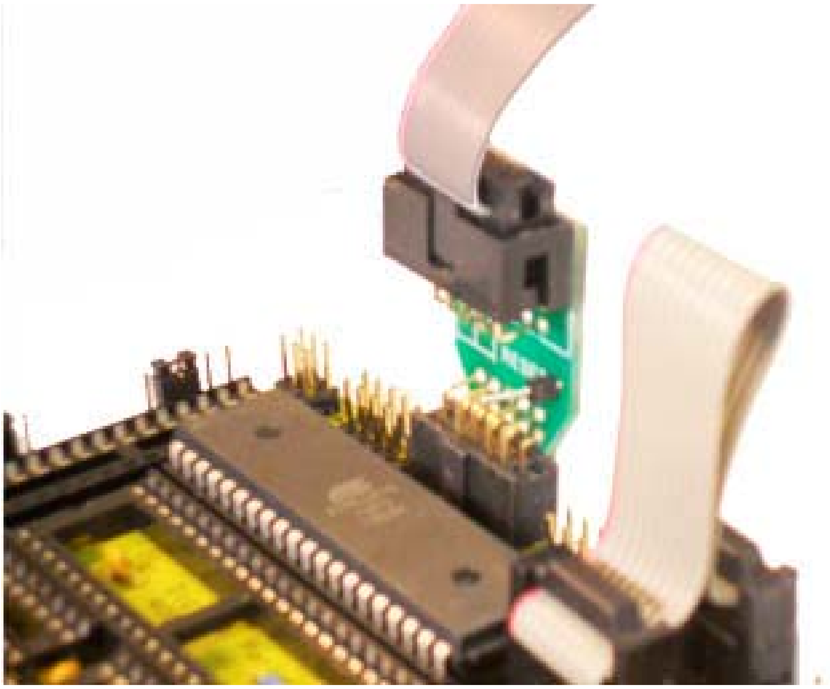
Newer AVR devices with 40-pins or more have JTAG port for On-chip Debug. Exceptions are the older Atmega8515 and Atmega8535 devices.

The board does not have a JTAG connection because there are too many sockets on it, so connections are made to the port pins of the AVR using the adapters supplied.

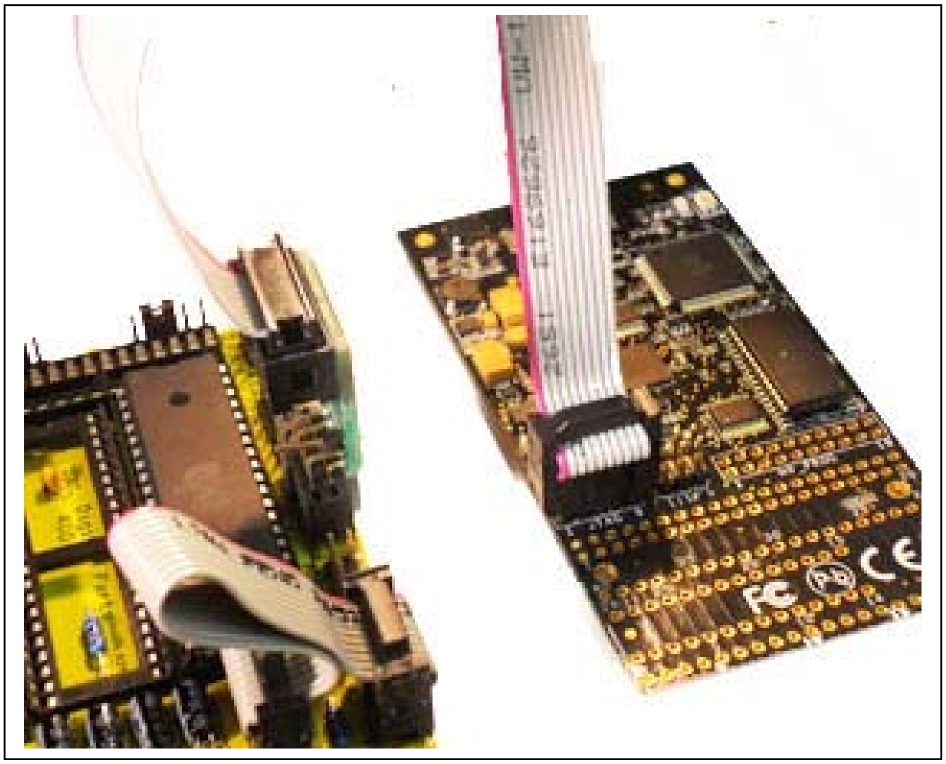
The JTAG pins on different AVR devices are in different places, so which adapter to use depends on which AVR you are using.

AVR Type	JTAG Pins	Adapter
ATmega32 ATmega323 ATmega16 Atmega164 Atmega324 Atmega644 Atmega1284	TDI - PC5 TDO - PC4 TMS - PC3 TCK - PC2	**Adapter B** to **Port C header**
Atmega162	TDI - PC7 TDO - PC6 TMS - PC5 TCK - PC4	**Adapter A** to **Port C header**

Fit the correct adapter to the correct port header, with the back of the adapter PCB facing out from the board



Fit the 10-way cable to the adapter and connect it to header marked JTAG on AVR Dragon, with keyway facing prototype area.



Apart from the hardware interface, the debug process is the same as DebugWire. Use **Build and Run** or **Debug > Start Debugging** to connect to device and then use **Run** or **Step**, or set breakpoints. A yellow arrow shows position in your code.

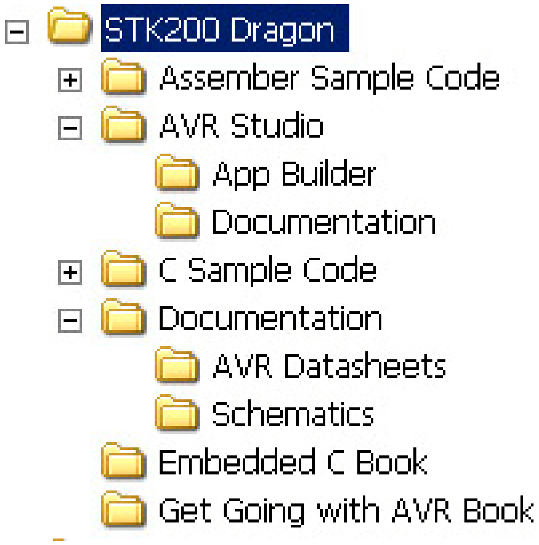
AVRStudio connection problems

AVRStudio can be a bit unstable, and if you get connection problems or peculiar behaviour, try unplugging USB cable, cycling power on STK200 board and sometimes closing and re-opening AVRStudio software.

Occasionally, you may need to restart PC to reset everything, but usually just closing AVRStudio works.

Other Documents

The Kanda installer will copy a range of documents to a folder called **STK200 Dragon** in your Program Files folder.



- STK200 Dragon
  - Assember Sample Code
  - AVR Studio
    - App Builder
    - Documentation
  - C Sample Code
  - Documentation
    - AVR Datasheets
    - Schematics
  - Embedded C Book
  - Get Going with AVR Book

Folder Description

- AVR Studio folder contains 2 folders, App Builder and Documentation. App Builder is a Wizard to help create assembler or C file templates. Documentation has PDF files on getting started with AVR Studio and AVR instruction set
- C Sample Code folder has example C Projects covering basic topics such as LED flashing, UART, LCD and Keypad
- Documentation folder has description of STK200 board, LCD details, AVR datasheets folder and STK200 board schematics
- Embedded C book has a PDF book about writing C code
- Get Going with AVR book is a good place to start if you do not know anything about AVR microcontrollers
- Assembler Sample Code folder contains Assembler example files on a range of topics.

AVRStudio Help Menu has AVR Tools User Guide. Select **AVR Dragon User Guide** from list for more information on AVR Dragon, and on its prototype area.

C Code Projects

Creating a C Project in AVR Studio is the same as assembler, except you choose C as Project Type. Create a project without creating an initial file and then add a C file as source file, just like assembler.

Alternatively, you can open a C project file (*.aps) from C Sample code folder.

Important
The C compiler is a plug-in for AVR Studio and is not directly controlled by it, so there are some important points to be aware of

1. C projects must have AVR device set correctly, unlike assembler files which are not so fussy, otherwise code will not run.

2. Changing device in AVR Studio project does NOT change device in C compiler. You need to go to **Project > Configuration Options** and change device there as well. Also on this screen, make sure **Create Hex File** is checked.

3. Output files are stored in a sub-folder called Default

There is a document called Pointers.PDF in C Sample Code folder that covers arrays and pointers if you are not familiar with C programming.

Embedded C Book

There is a PDF book on Embedded C Programming in Embedded C Book folder. It covers the basics of writing C for AVR microcontrollers but uses IAR C compiler instead of WinAVR and AVR Studio.

A free 4KB limited version of the IAR compiler, called Kick Start, is available for download. It is similar to WinAVR but produces smaller code and has lots of advanced features. AVR Dragon can be run from IAR Embedded Workbench, using their C Spy debug interface.

To setup AVR Dragon in IAR Embedded Workbench go to **Project > Options** and choose Debugging and set it to **DRAGONAVR**.

Atmel Website

Atmel who manufacture the AVR and produce AVRStudio are a very good source of information. Visit www.atmel.com for new versions of AVRStudio, application notes, new AVR devices and much more.

Contact Kanda

Contact Kanda at support@kanda.com or visit our website at www.kanda.com

Phone/fax is +44 (0)1974 261 273