

---

# Quick Start Guide for AVR Trainer, STK200ICE and STK300ICE

---

## Introduction

This package consists of a number of different elements, each of which has its own documentation. This guide is an overview of how to setup and use the package. For details on each part, please consult the relevant documentation.

---

## Package contents

- Get Going with AVR... book on CD
- AVR Studio development software
- USB or Serial JTAGAVR In Circuit Emulator
- STK200 AVR training board (STK300 board with STK300ICE)
- AVRISP-U USB port programmer or AVRISP Parallel port Programmer
- WinAVR C Compiler
- AVR Application Builder
- Documentation, schematics and datasheets
- Embedded C Book on CD

Where this guide refers to STK200, substitute STK300 if you have STK300ICE.

---

## How these elements fit together

The core of the package around which everything else fits is Atmel's AVRStudio Integrated Development Environment or IDE. This is a platform that lets you write code and assemble it for the microcontroller. It includes a Simulator that allows you to step through your code to find basic errors.

Once your code is written, the next step is to connect up the JTAGAVR In Circuit Emulator to the STK200 board, and run the code on the microcontroller in Debug Mode. Again, this is done from the AVR Studio IDE, which allows you to step through the code but this time it is running on the AVR instead of being simulated by the PC.

The JTAGAVR will also act as a programmer to download your finished code to the AVR on the STK200 board, with some AVR devices. As an alternative the package also contains an In System Programmer or ISP. This uses the standard ISP header unlike the JTAGAVR emulator and comes into its own when you build your own circuits or want to program a wider range of AVR devices. The AVRISP software can run on its own or it can be integrated into AVR Studio.

The other elements in the package are a standalone Application Builder and a C Compiler. The application Builder is a wizard that helps set up peripherals such as the serial port without having to know all the details in the AVR device datasheet. The C Compiler is included for those who want to progress beyond assembly language but we recommend that read the book on the CD on using C on the AVR before starting with the compiler.

---

## Getting Started

The place to start is by reading the book. The first half of the book covers the basics of numbering systems, boolean algebra, microcontroller structure, types of instruction and program design. You should make sure you are happy with these concepts before starting to load software.

The second half of the book is about worked examples, so before starting on these we need to load some software.

**The steps outlined here cover a range of activities. It is easier to install all the software and come back to details later or you may want to work through each step as you come to it – it is up to you.**

---

**Run KandaAVR.exe on CD.** This will

- 1) Install AVRStudio and AVRStudio update
- 2) Install AVRISP, including AVRStudio Plugin
- 3) Copy sample files, data sheets, schematics, C Book to your hard drive
- 4) Optionally install WinAVR C Compiler and integrate it into AVR Studio

1) Select the folder **AVRStudio > Documentation** on your hard drive (default path is C:\Program Files\AVRStudio\Documentation)



Now open Getting-started.pdf, which is a guide to using AVR Studio. Follow the steps in this guide to familiarise yourself with creating projects, writing code and other basics.



There is another folder here called App Builder. Application Builder is a tool that creates initialisation code for AVR peripherals and can be a very useful aid in quickly creating projects. It is not necessary to use it at the moment, but it is available if you want to play with it. To use this tool, open this folder and copy the exe file to your hard drive.

2) Now, you have a choice. We recommend that you run some sample programs on the hardware board (STK200) using the AVRISP programmer to see something happening. Then we can come back to the In Circuit Emulator later.

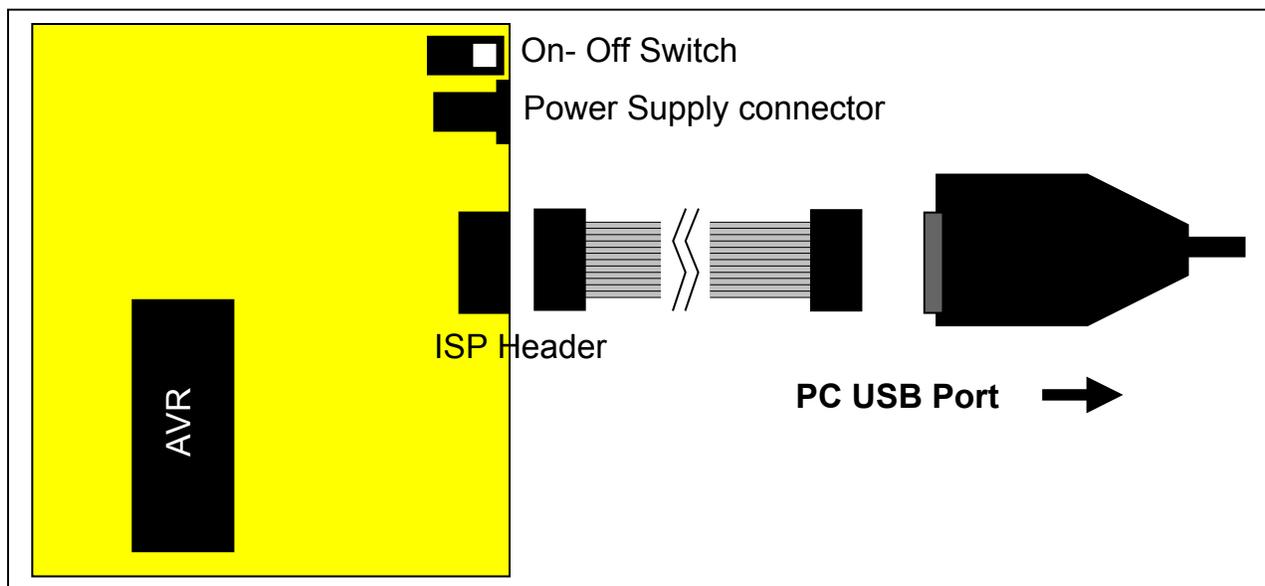
Go to **Samples >STK200 folder** in install folder. (**Samples > STK300 for STK300ICE**)

These files run on the STK200 or STK300 board.

---

## Connecting the AVR ISP programmer hardware

- Connect the 10-way lead labeled ISP only to the PC adapter and plug the dongle into your USB port.
- Connect the other end of the lead to the ISP header on the STK200 board
- Power the STK200 board – 9-15VDC



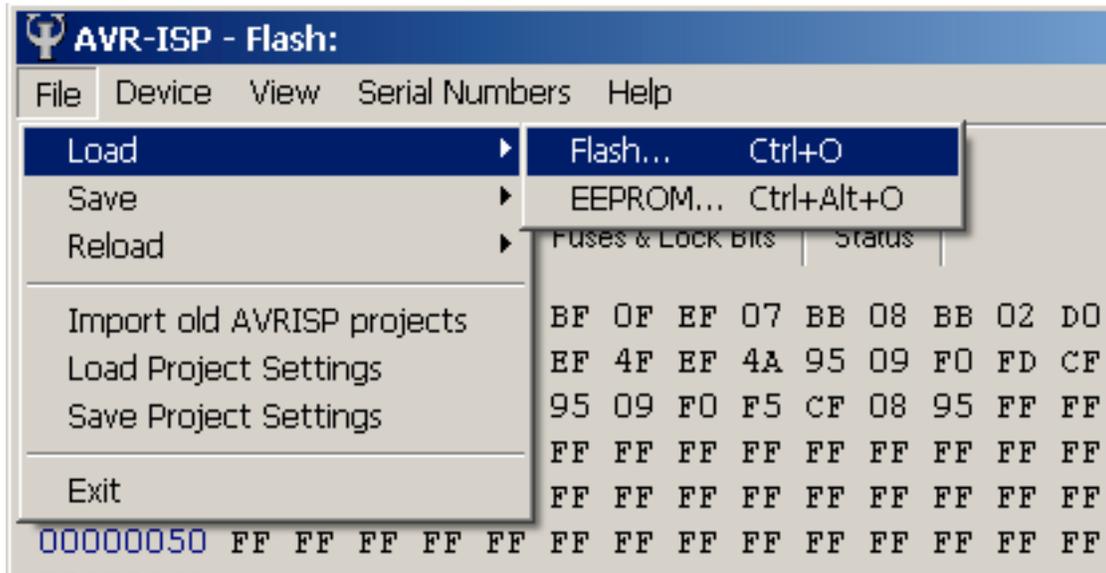
---

## Programming file into AVR

In AVR Studio, go to *Tools Menu > Kanda Tools > AVRISP-U* (or run AVRISP-U yourself from Programs or desktop icon)

In AVRISP, change the Device to Atmega16 (**or ATmega128 for STK300ICE**)

Now select *File > Load > Flash* and open `bincount200.hex` (**or `bincount300.hex`**) from the Sample code folder.



Now select *Device > Program > Flash*. Press F9 to Run or disconnect the ISP cable and the file should run on the STK200 board

### Important Notes

- 1) New AVR devices have clock source set to Internal RC Oscillator. This means they are NOT running from crystal on the board. To change to run from board crystal at 8MHz, do the following
  - Select Fuses and Lockbits tab in AVRISP
  - Change Clock Source fuses –CLKSEL- (bottom Left) to External Crystal 3-8MHz
  - On some devices, you need to uncheck CLKDIV8 fuse as this divides clock by 8
  - Go to Device menu > Program > Fuses
- 2) Once AVR devices have been programmed, they must be ERASED before they can be reprogrammed (Device menu > Erase).

### Loading a sample file into AVR Studio

Now we can run a program in AVR Studio

- Open AVRStudio
- Create a New Project in AVR Studio, and give it a name. Note the path to the project.
- Uncheck “Create Initial File”
- On next screen, select Debug Program as JTAG ICE and Device as ATmega16 (STK300ICE – Atmega128) and click finish
- Go to *Project menu > Add Existing File* and open bincount200.asm from (Samples > STK200 > Bincount) or STK300 -bincount300.asm for STK300ICE
- Press F7 or Go to *Project menu > Build* to assemble the file. The output file is a .hex file for the programmer – this will be stored in the same folder as the Project.

Once the file has been built, it can be debugged using JTAGAVR tool

---

## Using JTAGAVR ICE Tool

The next tool to look at is the In Circuit Emulator. We selected the JTAG AVR as the debug tool when we created the project, so the AVR Studio is ready to accept it. No software needs to be installed but there is a lot of documentation about using the emulator in the install folder in *Documentation > ICE* folder.

This documentation covers breakpoints, emulator use and much more. If you have any questions about the JTAG AVR, the answer should be in these files.

## Connecting the JTAG AVR hardware.

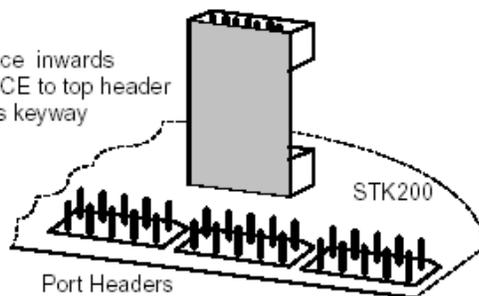
- Connect the USB lead to the PC and the JTAGAVR main board. Windows will now install drivers. The path for the USB drivers for the JTAG AVR is

**C:\Program Files\Kanda AVR\USB Drivers**

- Connect adapter to the JTAGAVR using 10-way lead – not lead marked ISP only.
- On STK200 and AVR-TRAIN, with Atmega16 devices, use adapter B (the one with the big letter B on it) and connect to Port C header, with headers facing inwards
- **On older STK300, use Adapter A and connect to Port F Analog Port**, with the connectors facing inwards.
- On new STK300, connect JTAG lead directly into JTAG header, without adapter
- Power the board

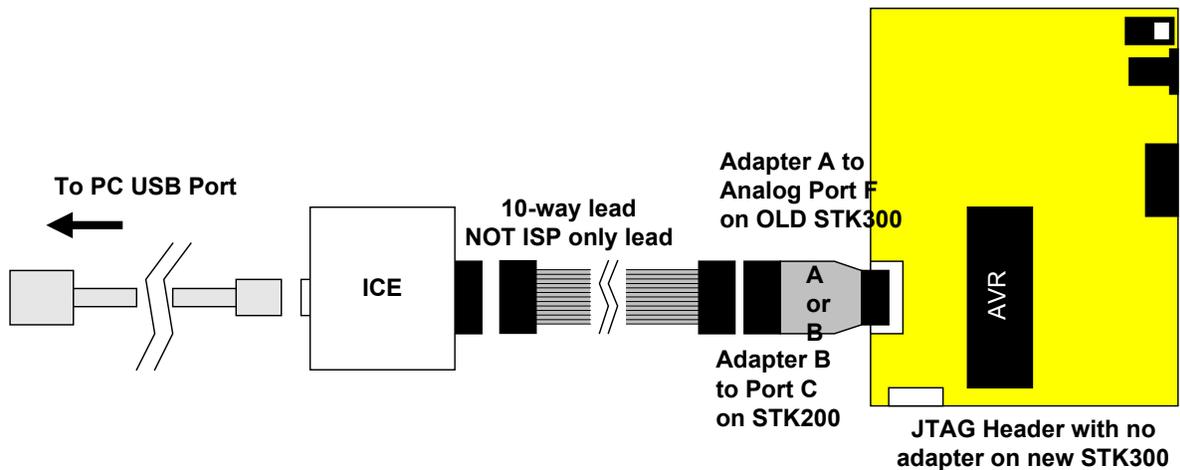
### Adapter:

Box Headers face inwards  
Connect JTAGICE to top header  
Top Header has keyway



### Old STK300ICE Only

The AREF Pot should be turned fully anticlockwise, as the JTAG pins are on the Analog header. This only has AREF pin not Vcc, so turning pot gives 5V on pin.



## Running JTAGAVR ICE from AVR Studio

This should be really easy. Reopen the Bincount project if you have closed it.

Build the Project.

Now go to *Debug menu > Start Debugging*

You can use Step, breakpoints and run to navigate through the code.

## Using JTAGAVR as a Programmer in AVRStudio

Go to Tools menu > Program AVR > Connect. Select platform as JTAGICE, and Port as COM1 to COM9.

This is just a brief introduction to the ICE. There is more information in the ICE directory, and in AVR Studio documentation and help.

---

## Next Steps

We have covered how to use the tools in this guide. The next step is to return to the book and begin to work through the examples.

Good Luck

---

## FURTHER INFORMATION

Please contact [support@kanda.com](mailto:support@kanda.com) for technical support or go to our website support pages for latest software.

See [www.kanda.com/support](http://www.kanda.com/support)

**Contact details**

**Website:** [www.kanda.com](http://www.kanda.com)

**email:** [sales@kanda.com](mailto:sales@kanda.com)

**Phone/Fax:** +44 (0)1974 261 273

---



*Atmel, AVR, AVR Studio are registered trade marks of Atmel Corp. San Jose, CA, USA*