STK300 AVR Starter Kit Manual

Kit Contents

- STK300 Board
- Atmega128L device mounted on a KANMEGDEV board
- AVRISP-U USB Port In System Programmer (ISP)
- Software CD and AVR Datasheets CD
- JTAGAVR

The only extra item required is a power supply – see Power Supply section for details about suitable power supplies.

Installing Software and documentation

The CD contains an executable file – Kanda AVR.exe. Run this installer and it will copy all the documentation, sample code and software to your PC.

The default install folder for Kanda software and documentation is

C:\Program Files\Kanda AVR

You also have the option in this installer to install WinAVR C Compiler and AVR Studio development environment. We recommend that you do install these unless you already have them on your PC.

There are three main pieces of software:

- **AVRISP** – This is the programmer software, used for transferring code from the PC into the AVR device on the board. To install the software Double Click the EXE file and follow the on-screen install instructions.

AVRISP has a Documents folder that contains PDF files with useful information about AVRISP.

- **AVR Studio** - AVR Studio is a complete development system for AVR devices, supplied by Atmel. It is probably the best available for any microcontroller. This environment is where you write new programs, and assemble them for the AVR chip. It also includes a Simulator for stepping through your code to debug it. The documentation folder includes instructions on using AVR Studio.

The installer allows you to install AVR Studio. The default install path is C:\Program Files\Atmel\AVR Tools\AVR Studio

- **WINAVR -** WINAVR is a C Compiler for AVR microcontrollers. Run the installer if you want to use C language for your AVR development. Documentation is included. WinAVR will be automatically linked to AVR Studio.

Summary
The only software you need to install to start with are AVRStudio for code development, WinAVR if you want to write in C and AVRISP for programming AVR devices on the STK200 board. The other software and documentation is available if you need it. **AVRISP will be integrated into AVR Studio if AVRStudio is installed first.** The Kanda installer does this by default.

Getting Started

AVRISP is the In System Programming software, AVRStudio is the development environment and the STK300 board is the test platform. Most people prefer to connect up the hardware and program some sample files on to the board using AVRISP, before tackling writing code in AVRStudio. However, it is up to you.

There is also a book in Kanda AVR folder called Get Going with… AVR, another good place to start.

The rest of this manual describes how to connect the hardware and use AVRISP. This is followed by a description of the STK300 board and finally by a brief introduction to AVRStudio.

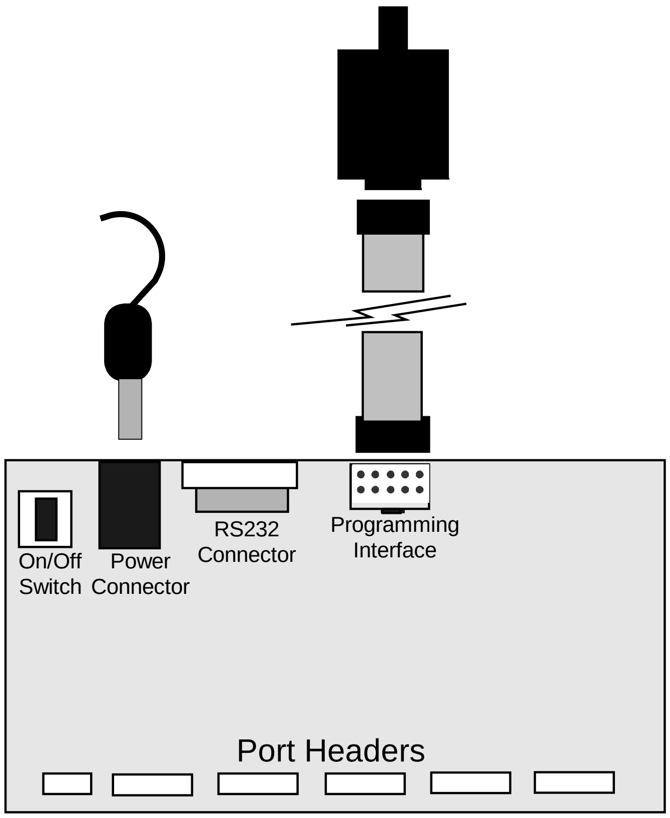
The only software you need to start with is AVRISP and AVR Studio. The rest can be used later if you want to experiment.

Connecting the hardware

Run Kanda AVR.exe from CD BEFORE connecting hardware

1) Connect the ISP dongle to the USB port on your PC. Windows should automatically install the USB drivers. If it does not install correctly, please see troubleshooting guide: C:\Program Files\Kanda AVR\AVRISP-U\ Troubleshooting USB Driver Install.pdf

2) Connect the ISP lead to the dongle
3) Connect the 10-way ISP lead to the STK300 board – see diagram.



Make sure you use the correct header on the STK300 board – it is the box header next to the 9-way D-Type connector and power input.

Power Supply

4) Connect a power supply to the board. The connector type required is 2.1mm barrel, centre negative or centre positive. It can be 7-12VAC or 9-15VDC, with a current capacity greater than 300mA.

AVRISP software

Now run the AVRISP software

- A Green Light should appear at the bottom of the screen with the message ISP Initialized : Detected Device Atmega128

If a RED light appears on the status line, then check the following

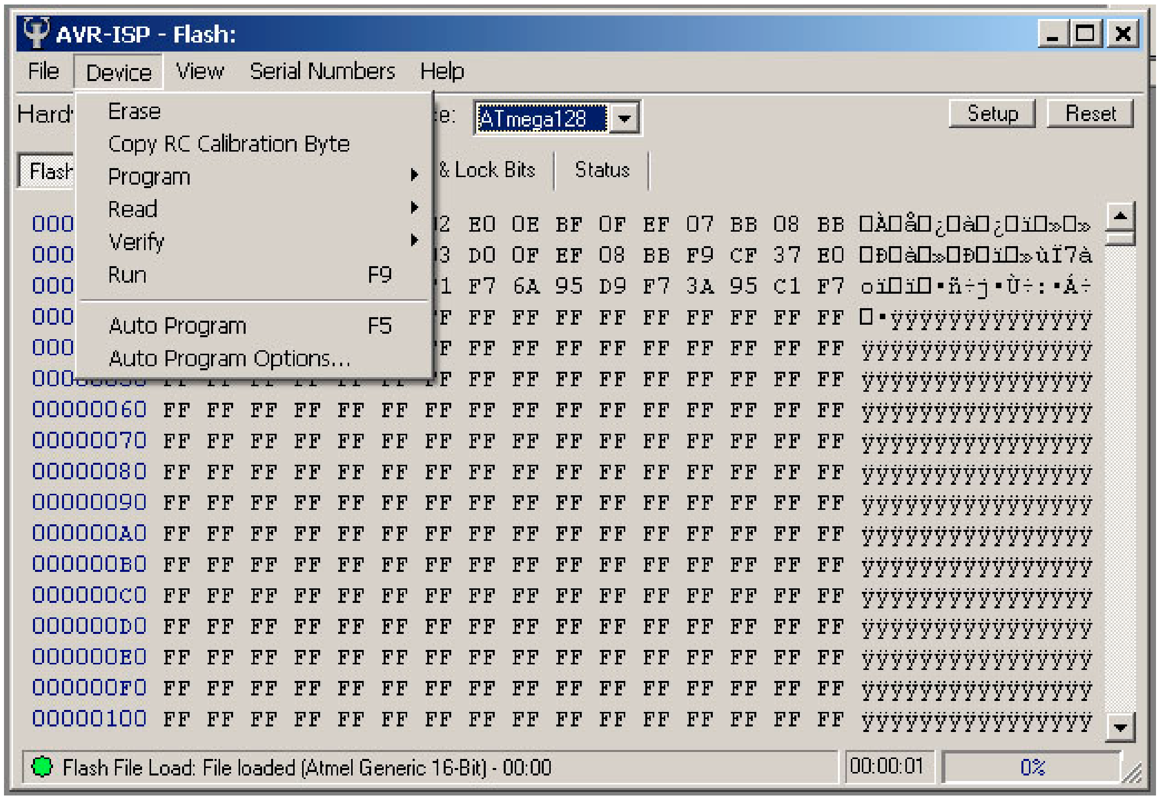
1) The AVRISP is connected to the board as shown above
2) The board is powered
3) AVR appears in Programmer box at top of screen

Load File
Once the board is connected properly, the next step is to load a file into the buffer. All operations are carried out on the data in the buffers. There are two buffers, one for code – Flash and one for data – EEPROM

Go to File Menu and choose Load > Flash, and select a file such as Samples > STK300 > LED Flash > LEDflash.hex

To load EEPROM data, follow the same procedure or type data directly into the buffer, in Hex numbers or ASCII characters.

Programming the AVR
All operations on the AVR chip are carried out from the Device Menu.



```
000          2 E0 0E BF 0F EF 07 BB 08 BB □À□å□¿□à□¿□ï□»□»
000          3 D0 0F EF 08 BB F9 CF 37 E0 □Ð□à□»□Ð□ï□»ùÏ7à
000          1 F7 6A 95 D9 F7 3A 95 C1 F7 oï□ï□•ñ÷j•Ù÷:•Á÷
000          F FF FF FF FF FF FF FF FF FF □•ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000          F FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000          F FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

```
Flash File Load: File loaded (Atmel Generic 16-Bit) - 00:00    00:00:01    0%
```

Note: **AVR devices must be erased first if they have been programmed before.**

To program and run the example file we have loaded into the buffer on to the STK300,

1) Choose Device > Erase
2) Choose Device > Program > Flash
3) Choose Device > Run – Run releases the chip from ISP so it can run its code

The LEDs on the board should begin to flash. Follow this procedure to program other code into the AVR on the STK300 board.

AVR microcontroller default fuses

New AVR devices come with the Clock Source set to RC Internal Oscillator by default. The actual speed of the oscillator varies from 0.5MHz to 4MHz depending on the device. This means that programs will run slower than expected, because the **AVR is NOT running from the 8MHz clock on the STK200 board.**

To change the Clock source, go to Fuses and Lockbits tab and change CLKSEL fuses from Internal Oscillator to External Crystal/Resonator 3 – 8MHz (1110, 1111). Then go to Device menu > Program > Fuses.

Other features of AVRISP

Auto Program

It can get a bit tedious carrying out all these steps individually, especially if you want to check that the code has loaded properly (Verify) and program the EEPROM as well. This is where Device > Auto Program (F5 key) is useful.

To use Auto Program feature, first go to Device > Auto Program Options and set the operations you want to carry out. Now selecting Device > Auto Program or pressing F5 will carry out all these operations.

Fuses and Lockbits

Click on the Fuses and Lockbits Tab to view the Fuse screen. These are settings on the Atmega128 that can be altered to affect the way the device runs and interacts with the circuit.

The default fuses will work happily on the STK300 board, but you may want to alter them to make the clock run from the 8MHz crystal on the board. Leave **JTAGEN** checked to use emulator. Full descriptions of each fuse and what they do are in the device datasheet – see Documentation > AVR Datasheets folder on CD for the datasheet on Atmega128.

On the right of the Fuse screen is a second tab marked Lockbits and Boot Options. This screen displays the various security settings and block options available. Again, these are described in the Atmega128 datasheet.

Status

The Status Tab lists the operations that have been carried out during the current programming session.

Serial Numbers

Serial numbers can be added to the Code space (Flash) or Data Memory (EEPROM) in a variety of formats. If Auto Program is used, the serial number will increment with every Auto-program cycle.

View Menu

CRC Checksums

This gives the checksum calculations in different formats for the whole buffer or just the used buffer. Applies to both Flash and EEPROM

Pop-up Warnings
This displays a message box if there is a verify error. Useful if repeated Auto Program operations are carried out, when it can be easy to ignore errors.

Setup
This can also be selected by clicking the Setup Button. It covers a variety of settings, which are not needed on STK300 board, except Communications Device. This is needed if you have multiple parallel ports when you should select the correct LPT port.

The other settings are mostly for use on your own circuits, and are described on the screen.

Reset Button
This is used to Reset the AVR on the board. This is a way of resetting communications if errors occur or the programmer cannot find the chip.

STK300 Board Description

The schematics of the board are available in the Documentation > STK300 folder on the CD.

Voltage Selection
There is one jumper – JP1 – for setting the board voltage. If the jumper is on, the board operates at 5V. If it is off, the board operates at 3.3V.
Notes:
1) If the board is set to 3.3V, the brownout circuit should be set to 2.9V by removing JP2
2) The ISP interface is supplied with 5V regardless of the board voltage.
3) The Atmega128L supplied as standard can operate from 2.7-5.5V

Brownout Circuit
The STK300 has a brownout detector to reset the AVR if the voltage drops too low. The brownout level is set using JP2 and should match the voltage selected by JP1 i.e. JP1 and JP2 both off or both on.

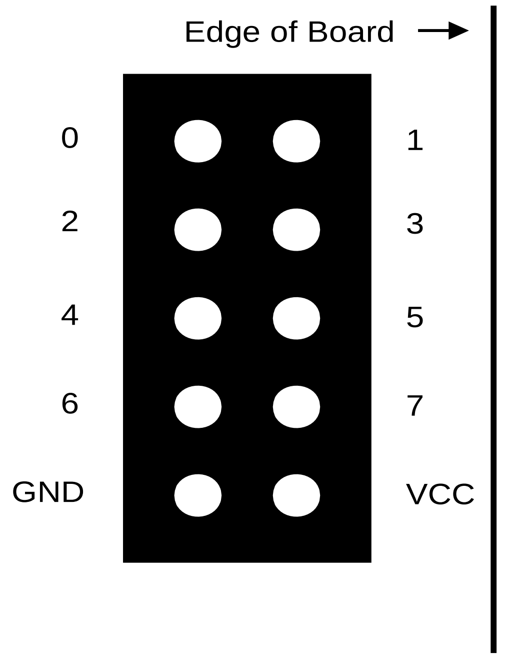
Clock Circuit
The board uses an 8MHz crystal connected to a 74HC00 external clock chip. The clock signal is connected to XTAL1 pin on Atmega128.

Serial Ports
The Atmega128 has two hardware USARTs. Both are connected through a MAX202 chip to give RS232 level signals.
USART0 (RXD0/TXD0) is connected to the female 9-way D-Type connector.
USART1 (RXD1/TXD1) is connected to a 2-way pin header, labeled COM2.

Both USARTs are also available, at TTL levels, on the expansion headers on the edge of the board (UART0 on PE0, PE1 and UART1 on PD2, PD3)
Details about Baud Rates, control and other settings are in Atmega128 datasheet.

Expansion Headers
All the device port pins are brought-out to pin headers on the edge of the board. The layout of these connectors is shown below.
All the ports are available - Port A to Port G – and all have the same layout except Port G. Port G is a smaller port and not all pins are brought-out because they are used by RTC circuit – see Port G section.

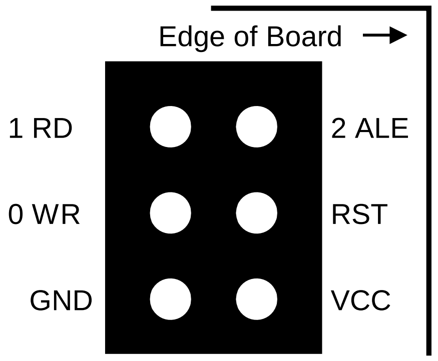
Port F is used by the ADC functions, and should be used in conjunction with the Analog header if Analog to Digital function is used.



ADC Circuit
The ADC circuit uses Port F and the 4-way Header marked Analog.
To use the ADC with the Internal Reference Pot, the jumper between AREF and POT pins on Analog connector should be on.

To use an external reference voltage, remove jumper and connect voltage to AREF and AGND pins.

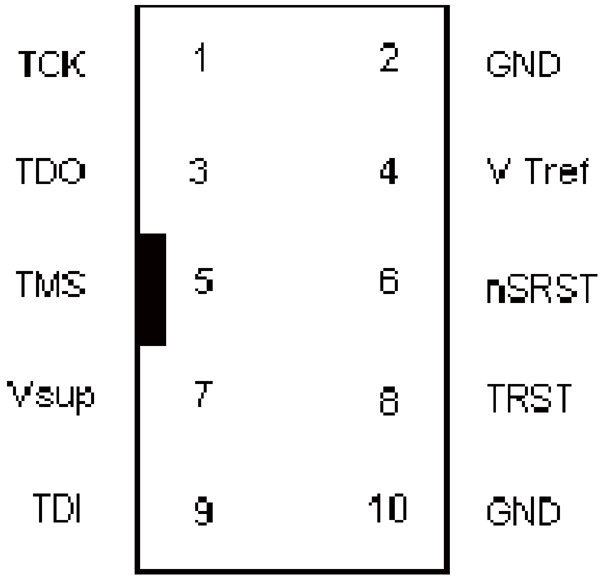
Port G header
Port G only has 5 pins. Of these, PG3 and PG4 are used by Real Time Clock (RTC) circuit so are NOT brought out to header. In addition to PG0/WR, PG1/RF and PG2/ALE, this header has Reset and a VCC and GND Pin.



Real Time Clock
A 32768 Hz crystal is connected to TOSC1 and TOSC2 pins - Pin 19 and Pin 18. These pins are also Port pins PG3 and PG4, but are not connected to the expansion headers. This crystal can be used to generate a 1 second pulse by setting Timer 0 to Asynchronous Operation - see Atmega128 data sheet for details

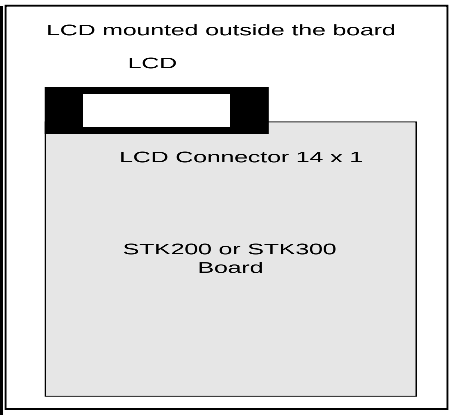
JTAG Header
The board has a second programming/debugging interface next to the mounted AVR chip. This connector should NOT be used for normal ISP operations

It has the recommended Atmel JTAG Interface pin-out and can be used with JTAG ICE tools, such as the Kanda JTAGAVR, which will plug straight in, without adapters. Kanda AVRUSB programmers, with JTAG adapter fitted, can also use this connector.



LCD Circuit
The 14 x 1 header marked LCD is for a standard 2 X 16 character LCD. The connections are shown on the board and in the following table. LCD display faces out

Connection Name	Function
0V	Ground 0V
+5	Vcc - must be 5V
Vo	Contrast
RS	Register Select- A14
Wr	Not Write
E	Enable
D0	Data Bit 0
D1	Data Bit 1
D2	Data Bit 2
D3	Data Bit 3
D4	Data Bit 4
D5	Data Bit 5
D6	Data Bit 6
D7	Data Bit 7



The board is designed for use with an LCD with a Hitachi driver chip built-in, with a contrast pin voltage of 0 to 5 Volts.   If no characters appear on LCD, then you may need to change resistors R22 and R23 to alter the contrast voltage, especially if you use a different LCD.

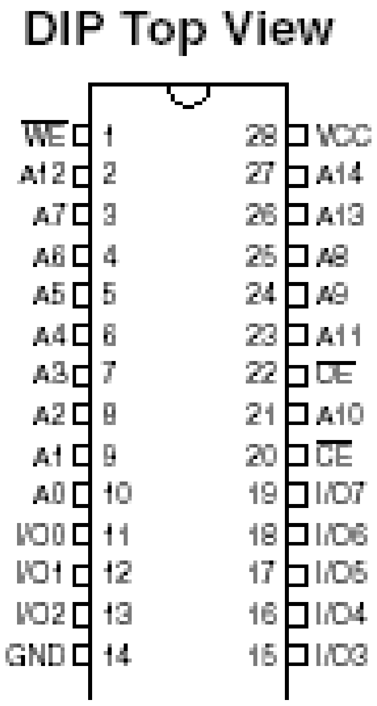
This board is not designed for use with back lit LCD modules, so the extra connections needed are not included and the power supply will not supply the current required.

The LCD can be driven in Memory Mapped or I/O Mode - see data sheet. If the crystal has been changed to a faster crystal frequency, memory mapped operations will not work.
A suitable LCD is available on the Kanda site (http://www.kanda.com/go/570910-PM)
A sample file is included on the CD in Samples directory.

External Memory
The board is fitted with sockets for an address latch (74HC573) and a Flash RAM chip (29C256).

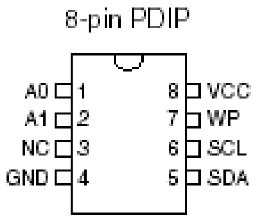
These are available as a pair from Kanda –
http://www.kanda.com/go/STK200-RAM

29C256 Pin-outs



24C EEPROM Socket
The board is fitted with a socket for 24C EEPROM Chip. This is an I^{2}C device, or in Atmel speak, a Two Wire Serial Interface (TWI).

The socket is connected to Pin 25 and Pin 26 of the Atmega128. These are SCL and SDA pins or PD0 and PD1.
The socket layout is shown below.



A full description of the TWI is given in the Atmega128 datasheet. Pull-ups must be enabled on these lines.

Switches and Bar LED
There is a 10-way bar LED fitted to the board.

- One LED is Power (labeled ON), and should be on when the board is powered.
- The next LED is labeled ISP and should be on when the board is in ISP mode.

Note: The Atmega128 will not run its code if the board is in ISP mode. To exit ISP, choose Device > Run command or disconnect the ISP lead from the board

- The other 8 LEDs (0-7) are for user code. They are active low (0 switches them On), and they are connected to a 10-way header marked LED'S. A short 10-way lead connects this header to the Port B header by default. Move the lead to another Port header if you want to use a different AVR Port for LED output.

The 8 switches are also active low (0 when pressed), and are connected to the 10-way header marked SWITCHES. A 10-way ribbon cable connects them to Port D by default. Again, move the cable to another header if a different port is required.

Both the LED and SWITCH headers also have GRD and VCC pins.

Note: AVR ports have three registers controlling them:
1) Port – value to write
2) DDR – direction register (0 is input)
3) Pin – value to read

So, to read switches, set DDR register to 0 and read PIN register – 0 is switch pressed. To write to LEDs, set DDR to 0xFF and write values to Port register, where 0 is on.

Summary

There is sample code on the CD, which will be copied to install folder. The board schematics are also available on the CD.

AVR Studio

AVR Studio is Atmel's development environment. Documentation is available in the AVR Studio folder in default install path.

There is a sub-folder called Documentation that contains
- Getting started guide. This explains how to set up AVR Studio and create programs.
- Instruction set guide, that lists the AVR instruction set
- Tools set up guide, which explains how to add AVRISP to AVR Studio Tools menu

Here is a brief guide to using one of the sample files in AVR Studio, but more information is available in the 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. Follow this procedure.

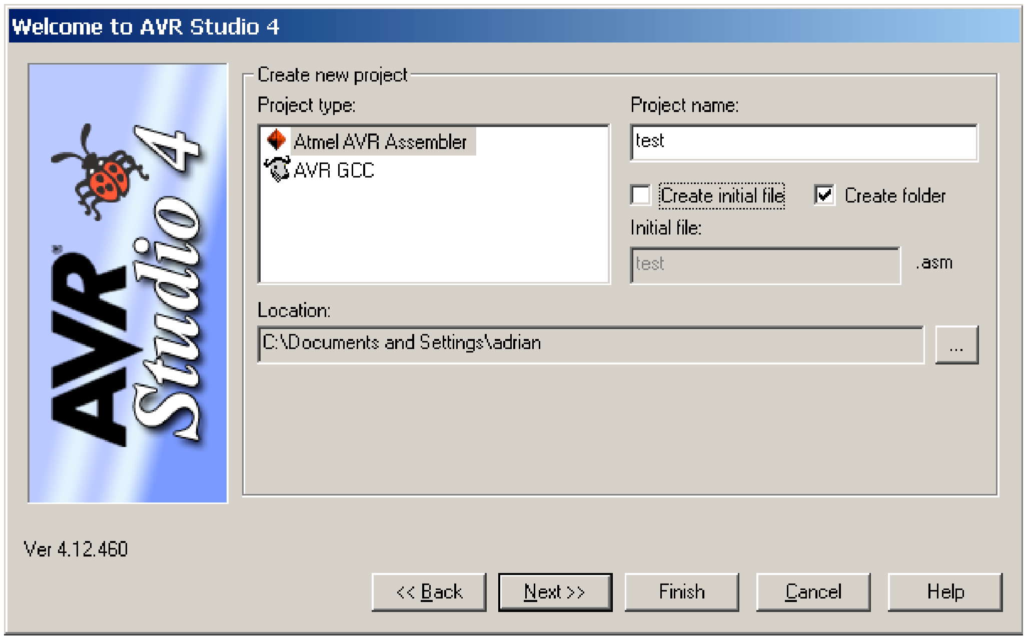
1) Run AVRStudio and select New Project from Welcome screen. If Welcome screen does not appear, select Project Menu > Project Wizard

2) Choose Atmel AVR Assembler as Project Type (see end of section for C Projects)

3) Give the Project a name

4) Uncheck Create Initial File box

5) Set location of where to save Project files



Project name: test

[ ] Create initial file [✓ Create folder]

Initial file: test .asm

Location: C:\Documents and Settings\adrian

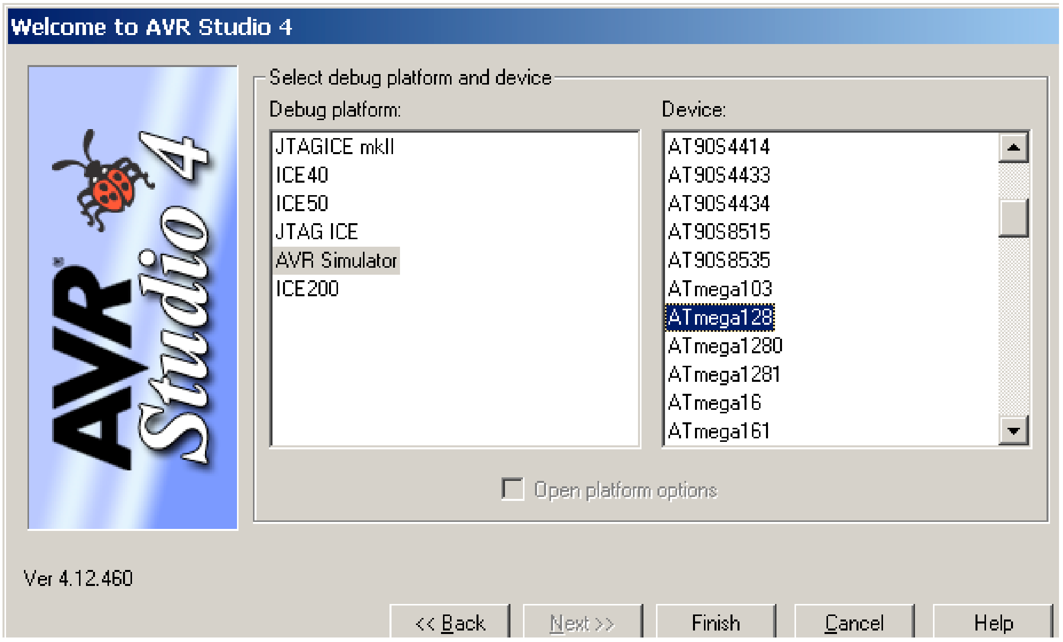
6) Click Next Button

On the next screen,

1) Set Debug platform to AVR Simulator, unless you have an In Circuit Emulator such as our JTAGAVR. In this case, choose JTAG ICE.

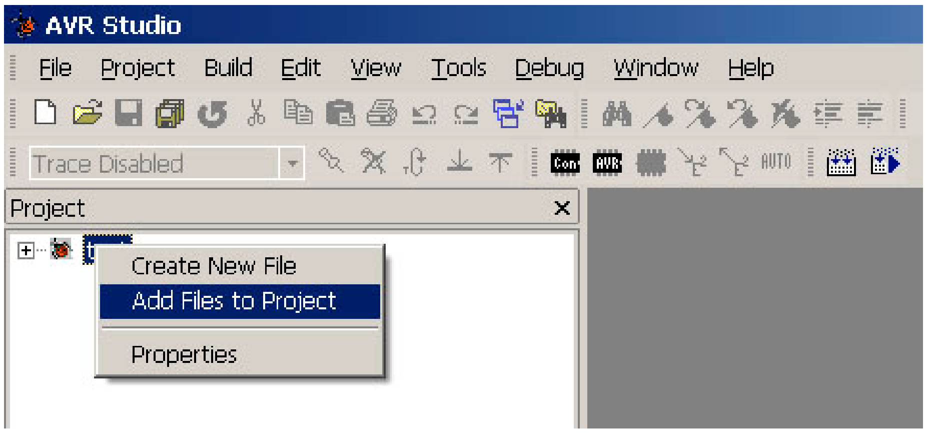
2) Set Device to Atmega128

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 assember file from Sample Code > STK300 folder on CD, such as LedFlash300.asm

3) Now go to Build Menu > Build. This will assemble the file and create object code file (.hex) for AVRISP programmer.

To Run the code in the Simulator, go to Debug Menu > Start Debugging. Choose commands such as Step Over to step through code or add Breakpoints and Run. These debug features are described in documentation and in AVR Studio help.

Running AVRISP from AVR Studio

AVRISP will be added to AVR Studio Tools menu, if it is installed after AVR Studio. The default installer for STK300 will install them in the correct order.

To run AVRISP from AVR Studio, go to **Tools Menu > Kanda Tools > AVRISP-U**
AVRISP will open a project with the correct files and device choice, using your AVR Studio project. It will also set Auto-program Options to

- Reload File
- Erase
- Program and Verify Flash
- Program and Verify EEPROM, if an EEPROM file exists
- Run the device

This means that you just need to use AutoProgram (F5) to load your code into the AVR on the STK200 board.

C Projects
Winavr (on CD) is a C Compiler for AVR, which will be installed by Kanda AVR installer by default.. When you run Studio you can select project type as AVR GCC – use this for WinAVR. If Winavr is not installed, AVRStudio will prompt for installation.

There are example C Projects in Sample Code folder in Kanda AVR folder. 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

Embedded C Book

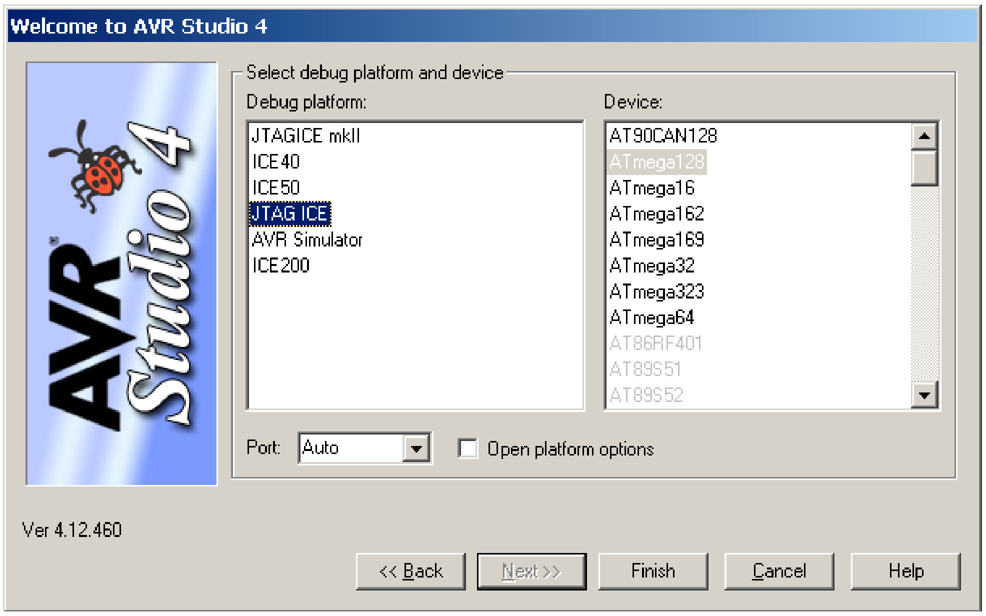
There is a PDF book on Embedded C Programming in Embedded C Book folder. It covers the basics of writing C for AVR microcontrollers but uses IAR C compiler instead of WinAVR and AVR Studio. A free 4KB limited version of the IAR compiler, called Kick Start, is available for download from IAR. It is similar to WinAVR but produces smaller code and has lots of advanced features.

Using JTAGAVR with STK300 Kit

If you have the JTAGAVR Tool as part of the package or you have purchased it separately, then it is simple to use with the ST300 board. Please note that the JTAGAVR is an In Circuit Emulator and not primarily a programmer, so AVRISP is still used as a separate tool.

Using JTAGAVR with AVR Studio

The only difference in the description of how to use AVR Studio given above is the choice of Debug Platform in the Project Wizard. On the second page, set Debug Platform to JTAG ICE not AVR Simulator.


Welcome to AVR Studio 4

Select debug platform and device

Debug platform: JTAGICE mkII, ICE40, ICE50, JTAG ICE (selected), AVR Simulator, ICE200

Device: AT90CAN128, ATmega128 (selected), ATmega16, ATmega162, ATmega169, ATmega32, ATmega323, ATmega64, AT86RF401, AT89S51, AT89S52

Port: Auto

Open platform options

Ver 4.12.460

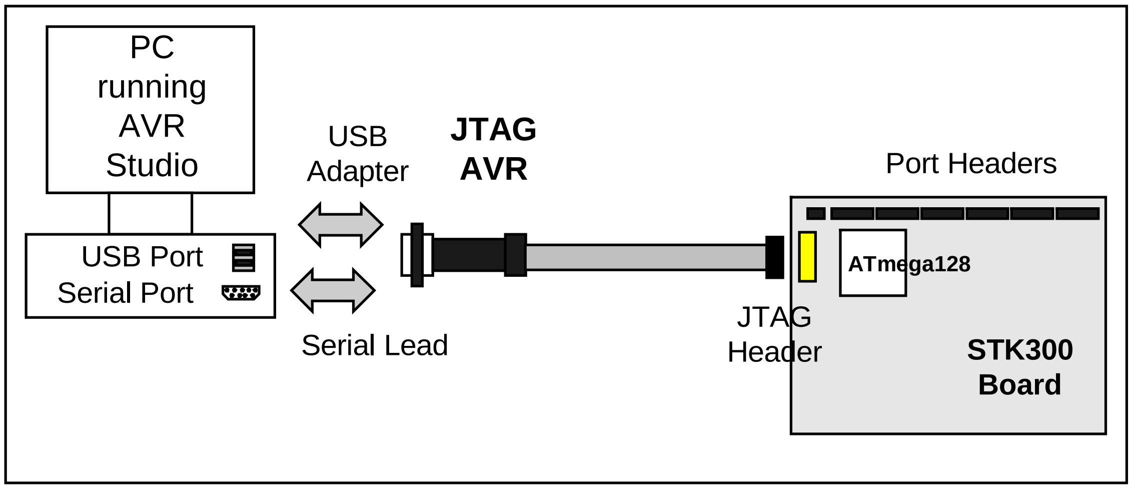
The list of available devices is much smaller, as only some AVR devices have the JTAG Interface. Choose Atmega128. For more device support and the option to use AVRStudio 6, you need AVR Dragon or ICE3.

The Port setting on this screen defaults to Auto. This means that AVR Studio will scan all Serial Ports looking for the ICE tool.

The rest of the description on debugging in AVR Studio is the same as for using AVR Simulator.

Connecting JTAGAVR to STK300 Board

1) Connect the JTAGAVR Tool to the PC USB port (or use a USB to Serial cable).
2) Connect the 10-way ribbon cable to the JTAGAVR
3) Plug the 10-way cable into the **JTAG** header on the STK300 board



Note: The LCD, if fitted, will cover JTAG port. In this case you will have to use adapters supplied.

1) Connect Adapter A to end of JTAGAVR lead
2) Plug adapter into Port F header, with adapter headers facing into board
3) Make sure ADC port is turned fully on – 5V

JTAGAVR Documentation
JTAGAVR uses AVR Studio software, so no extra software needs installing. On the CD supplied with STK300ICE or JTAGAVR, there is documentation on the tool. It is in a Folder called ICE, in Documentation folder.

These PDF files give more detail about using the JTAGAVR tool.

Firmware Updates
If you change AVR Studio to a different version or install a new Service Pack, then it may want to do a firmware update using the built-in AVRProg software. Serial port versions of JTAGAVR will update fine but USB versions will not be found by AVRProg, because it only looks at COM1 to COM4. See **JTAGAVR Quick Start Guide** in **Documentation > ICE** folder for details on changing COM Port setting.

Conclusion

This guide has briefly described the main features of the STK300 package. There is more information about AVR Studio on the CD, in its Help file and online at www.atmel.com

AVRISP has a help file, and there are AVR datasheets and instruction set documentation on the CD.

The other packages on the CD have help and documentation included.

Further Information

More information about AVR code and programming can be found online

- www.avrfreaks.com – a forum and help site for AVR
- www.kanda.com/support – new software versions, FAQ and documents about AVR and kanda tools
- www.atmel.com/products/avr - more information about AVR microcontrollers
- Books are available at Amazon. Kanda supply a basic book at www.kanda.com/go/KB0010

Other tools and accessories

We supply add-ons for the STK300, including LCD modules, keypads, Flash RAM, 24C EEPROMs, extra mounted Atmega128 devices, ICE and a universal power supply.

- Flash RAM and Address Latch – see www.kanda.com/go/STK200-RAM
- LCD module – see www.kanda.com/go/570910-PM
- Data Entry Keypad – see www.kanda.com/go/780305-PM
- JTAGAVR ICE – see www.kanda.com/go/JTAGAVR
- Power supply – see www.kanda.com/go/PSU9V-UNI
- Mounted Atmega128 device – www.kanda.com/go/KANMEGDEV5
- Mounted Atmega128L device – www.kanda.com/go/KANMEGDEV3
- 24C Serial EEPROM – see www.kanda.com/go/EE-02

Copyright

Atmel, AVR Studio and AVR are Registered Trade Marks of Atmel Corp., San Jose, California, USA

Website: www.kanda.com

Email: sales@kanda.com

Tech support: support@kanda.com

Support Pages: www.kanda.com/support

Phone: +44 (0) 8707 446 807
Fax: +44 (0) 8707 446 807
Address
Canolafan
Llanafan
Aberystwyth
SY23 4AY
UK