

**AVR<sup>®</sup> ISP**

---

**User Guide**





# Table of Contents

---

## Section 1

Introduction .....	1-1
1.1 Features .....	1-1
1.2 Device Support .....	1-2

---

## Section 2

Getting Started.....	2-1
2.1 Unpacking the System .....	2-1
2.2 System Requirements.....	2-1

---

## Section 3

Hardware Description .....	3-1
3.1 General Board Description.....	3-1
3.1.1 RS-232 Serial Interface .....	3-1
3.1.2 Control Section .....	3-1
3.1.3 ISP Interface Cables.....	3-2
3.2 AVRISP Power Requirements and Considerations .....	3-2
3.3 Connecting AVRISP to Target Board.....	3-2
3.4 Handling the ISP Lines .....	3-3
3.4.1 VCC and GND .....	3-3
3.4.2 SCK .....	3-3
3.4.3 MOSI/MISO/SCK.....	3-3
3.4.4 RESET.....	3-4

---

## Section 4

Software Front-end .....	4-1
4.1 Installing AVR Studio .....	4-1
4.2 Using AVR Studio .....	4-1
4.2.1 Starting the Windows Software .....	4-1
4.2.2 Starting AVRISP Interface .....	4-1
4.2.3 Using AVRISP and STK500 Simultaneously.....	4-2
4.3 AVRISP User Interface .....	4-2
4.3.1 “Program” Settings .....	4-2
4.3.1.1 Device .....	4-3
4.3.1.2 Programming Mode.....	4-3
4.3.1.3 Flash .....	4-3



4.3.1.4	EEPROM .....	4-3
4.3.2	"Fuses" Settings .....	4-3
4.3.3	"Lock Bits" Settings.....	4-5
4.3.4	"Advanced" Settings .....	4-6
4.3.4.1	Signature Bytes.....	4-6
4.3.4.2	Oscillator Calibration Byte.....	4-6
4.3.4.3	Reading Oscillator Calibration Byte .....	4-6
4.3.4.4	Writing Oscillator Calibration Byte.....	4-6
4.3.5	"Board" Settings.....	4-6
4.3.5.1	Oscillator .....	4-7
4.3.5.2	Revision .....	4-7
4.3.6	"Auto" Settings.....	4-7
4.3.6.1	Setting up the System for Auto Programming.....	4-8
4.3.6.2	Logging the Auto Programming to a File.....	4-8
4.3.7	History Window.....	4-8
<hr/>		
<b>Section 5</b>		
	Command Line Software .....	5-1
5.1	Parameters: .....	5-1
5.1.1	Sample Usage .....	5-3
<hr/>		
<b>Section 6</b>		
	Special Considerations .....	6-1
6.1	Fuse Programming .....	6-1
6.2	RESET Used as General IO Port.....	6-1
6.3	AVR Devices with no ISP Option .....	6-1
6.4	Devices without RC Oscillator Calibration .....	6-1
<hr/>		
<b>Section 7</b>		
	Troubleshooting Guide .....	7-1
<hr/>		
<b>Section 8</b>		
	Technical Specifications .....	8-1
<hr/>		
<b>Section 9</b>		
	Technical Support.....	9-1



# Section 1

---

## Introduction

*Figure 1-1.* AVRISP In-System Programmer



The AVR<sup>®</sup> In-System Programmer (AVRISP) from Atmel<sup>®</sup> is a professional low-cost In-System Programmer covering all AVR 8-bit RISC Microcontrollers. The programmer connects to a PC through a standard RS-232 serial interface and draws the necessary power from the target board eliminating the need for an additional power supply. AVR Studio<sup>®</sup> is used as front-end software, giving you a highly-integrated development solution.

---

### 1.1 Features

- AVR Studio Interface
- ISP Programming of all AVR Devices
- Programs both Flash and EEPROM
- Supports Fuses and Lock Bit Programming
- Supports RC Oscillator Calibration
- Upgradeable to Support Future Devices
- Operates at Target Voltages from 2.7V to 5.5V
- Adjustable Speed – Supports all Target Boards Running at a Speed Higher than 8 kHz
- RS-232 Interface
- Powered from Target – No need for Additional Power Supply

The AVRISP is fully supported by AVR Studio version 3.5 or higher. For up to date information on this and other AVR tool products please read the document “avrtools.pdf”. Latest version of AVR Studio (“avrtools.pdf”) and this user guide can be found in the AVR section of the Atmel web site.

---

### 1.2 Device Support

AVRISP supports all AVR 8-bit RISC Microcontrollers with ISP programming option. Support for new devices will be added through new versions of AVR Studio.

AVR Studio will automatically prompt the user if it detects that the firmware in the programmer is outdated.

The following devices are currently supported:

- ATtiny12
- ATtiny15
- ATtiny22
- AT90S1200
- AT90S2313
- AT90S/LS2323
- AT90S/LS2343
- AT90S/LS2333
- AT90S4414
- AT90S/LS4433
- AT90S/LS4434
- AT90S8515
- AT90S/LS8535
- ATmega161(L)
- ATmega16(L)
- ATmega163(L)
- ATmega32(L)
- ATmega323(L)
- ATmega103(L)
- ATmega128(L)

Information about supported devices in latest versions of the firmware can be found in the AVR Studio Online Help System.

**Note:** For special programming considerations please see Section 6.



## Section 2

---

# Getting Started

---

### 2.1 Unpacking the System

The AVRISP contains the following items:

- AVRISP Programmer
- AVRISP User Guide
- 10-pin ISP Cable (Connected to AVRISP)
- 6-pin ISP Cable
- 9-pin RS-232 Cable

---

### 2.2 System Requirements

The minimum hardware and software requirements are:

- 486 Processor (Pentium is recommended)
- 16 MB of RAM
- 15 MB of Free Hard Disk Space
- Windows® 95/98/2000 and Windows NT® 4.0
- 115200 Baud RS-232 Port (COM Port)

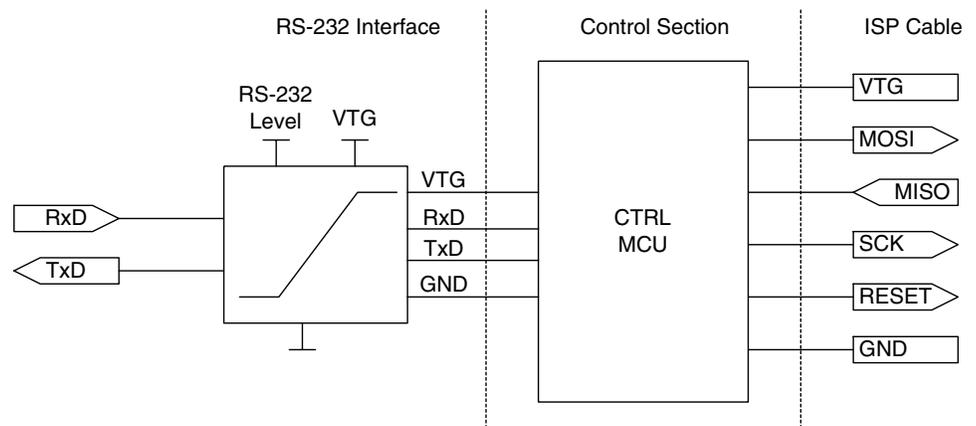


## Hardware Description

### 3.1 General Board Description

A block diagram of the AVRISP is shown in Figure 3-1. The AVRISP can be divided in three sections: The RS-232 interface, the Control section and the ISP cable. In this section a brief overview of the different blocks will be given.

**Figure 3-1.** Simplified AVRISP Block Schematics



#### 3.1.1 RS-232 Serial Interface

The AVRISP uses a standard female DSUB, RS-232 port for communication with the front-end software (AVR Studio). It supports communication of 115200 baud.

#### 3.1.2 Control Section

The control MCU handles all communication between the target AVR and the front-end software. The AVRISP is completely software controlled from AVR Studio. No manual configuration of the ISP is needed.

**Table 3-1.** Status Led

Led Color	Description
Red-Yellow-Green-Off-Green Cycle	Power on Sequence
Yellow	Busy – Programming
Red	Programming Failed
Green	Ready – OK

A 3-color LED indicates the status of the AVRISP. During programming the LED is yellow. When the target AVR is successfully programmed, the LED will turn green. If programming fails, the LED will turn red to indicate that programming (or verification)

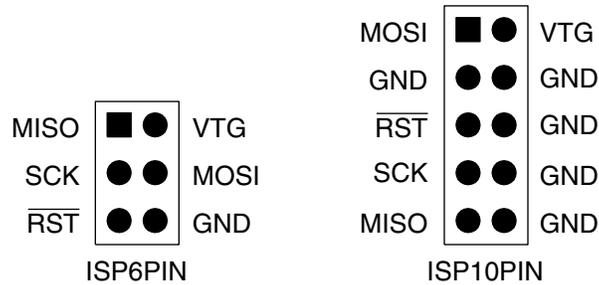
failed. If programming fails, check the “Troubleshooting Guide” on page Table 7-1. During start-up the status LED cycles through red, yellow to green to indicate that the master MCU is ready.

**3.1.3 ISP Interface Cables**

AVRISP supports both the 6-pin header connector pinout, used by the Atmel AT90ISP cable, and the 10-pin header connector used by the STK200 and STK300 development boards. Figure 3-2 shows the pinouts for the 6-pin and 10-pin ISP connectors.

AVRISP is delivered with one 6-wire and one 10-wire ISP cable. Use the one that match the pinout of the target ISP connector. However, note that only one cable should be connected, and used, at any given time. By default the 10-pin header connector is mounted.

**Figure 3-2.** AVRISP Connectors (Top View)



**Table 3-2.** AVRISP Connector pinout

Signal	6-Pin	10-Pin	I/O	Description
VTG	2	2	–	Power is delivered from the target board
GND	6	3,4,6, 8,10	–	Ground
MOSI	4	1	Output	Commands and data from AVRISP to target AVR
MISO	1	9	Input	Data from target AVR to AVRISP
SCK	3	7	Output	Serial Clock, Controlled by AVRISP
RESET	5	5	Output	Reset. Controlled by AVRISP

**3.2 AVRISP Power Requirements and Considerations**

Since the AVRISP draws power from the target, it is important that the target board is able to provide enough power to ensure correct operation. The AVRISP will draw maximum 50 mA @ 5.5V. The current is drawn through the VTG line. The target should thus be able to supply at least this amount of power in addition to the requirements of the target board itself.

See “Technical Specification” section on page 8-1 for information on power consumption.

The AVRISP is not equipped with a power switch. Power is turned on when AVRISP is connected to the target application and turned off when disconnected.

**3.3 Connecting AVRISP to Target Board**

The AVRISP connects to the target board through a 3 x 2 or 5 x 2 male header connector with 2.54 mm (0.1”) spacing (Figure 3-3). If there is a 6-pin or 10-pin ISP connector on the target board, the cable on the AVRISP has to be changed accordingly.

The power of the target board should be turned off when connecting or disconnecting the header connector. Hot-swapping is not supported and might damage the programmer.

To change the cable the AVRISP box must be opened and the correct cable must be connected. See Figure 3-3 and Figure 3-4. Note that only one cable should be connected at any given time. Make sure that the cable is mounted in the correct orientation.

**Figure 3-3.** AVRISP with 10-pin ISP Connector

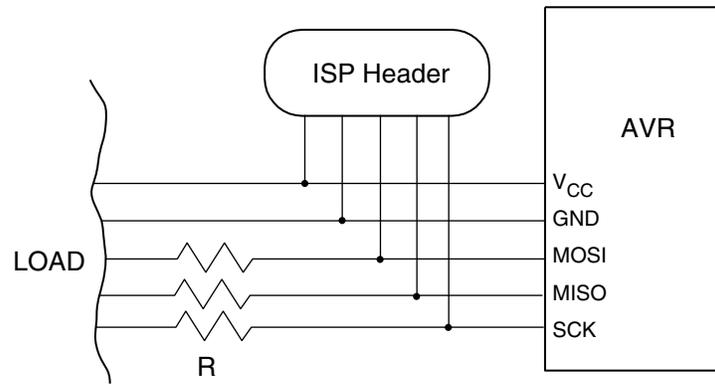


**Figure 3-4.** AVRISP with 6-pin ISP Connector



- 
- 3.4 Handling the ISP Lines** When connecting the AVRISP to an external target some precautions should be taken. In this section a few hints and tips will be given that should assure problem free communication between the AVRISP and target device.
- The part can be programmed in-system from AVR Studio with In-System Programming (ISP) in programming mode, running at the parts normal supply voltage.
- For instruction on using the AVR studio programming software, see “Using AVR Studio” on page 4.2
- 3.4.1 VCC and GND** Connect the AVRISP ISP power lines to the appropriate pins on the AVR device (preferably through a 6 or 10-pin connector on the target board). Make sure the target voltage is within specified range of the programmer (2.7V - 5.5V). Make sure that the target power supply can deliver the additional power required to power the AVRISP at the given voltage. See Section 8
- 3.4.2 SCK** The target AVR samples the clock signal generated from the AVRISP. To make the sampling robust, a target main clock, four times higher than the programming clock (SCK), is required. By selecting the correct target speed in AVR Studio, the correct SCK is automatically generated. The lowest supported target frequency is 8 kHz.
- 3.4.3 MOSI/MISO/SCK** If the MOSI, MISO or SCK lines are used as general I/Os in the application, it is recommended to use series resistors between the load and the AVR as shown in Figure 3-5. The AVRISP lines should be connected directly to the AVR pins, without any series resistors.
- Note that the ISP lines are driven directly from an AVR microcontroller inside the AVRISP. The resistors must be set according to this AVR’s maximum sink and source currents and the application load. Refer to the AVR’s datasheet for the maximum sink and source current. Datasheets can be downloaded from Atmel’s web site, [www.atmel.com](http://www.atmel.com).

**Figure 3-5.** Loading on the MISO/MOSI/SCK Lines



**3.4.4 RESET**

To enter programming mode, AVRISP needs to pull RESET low. It is important that the external pull-up resistor on RESET pin is not so strong that it forces (holds) the pin high. To avoid this problem it is recommended that the RESET pull-up resistor should be no less than 10 kΩ.

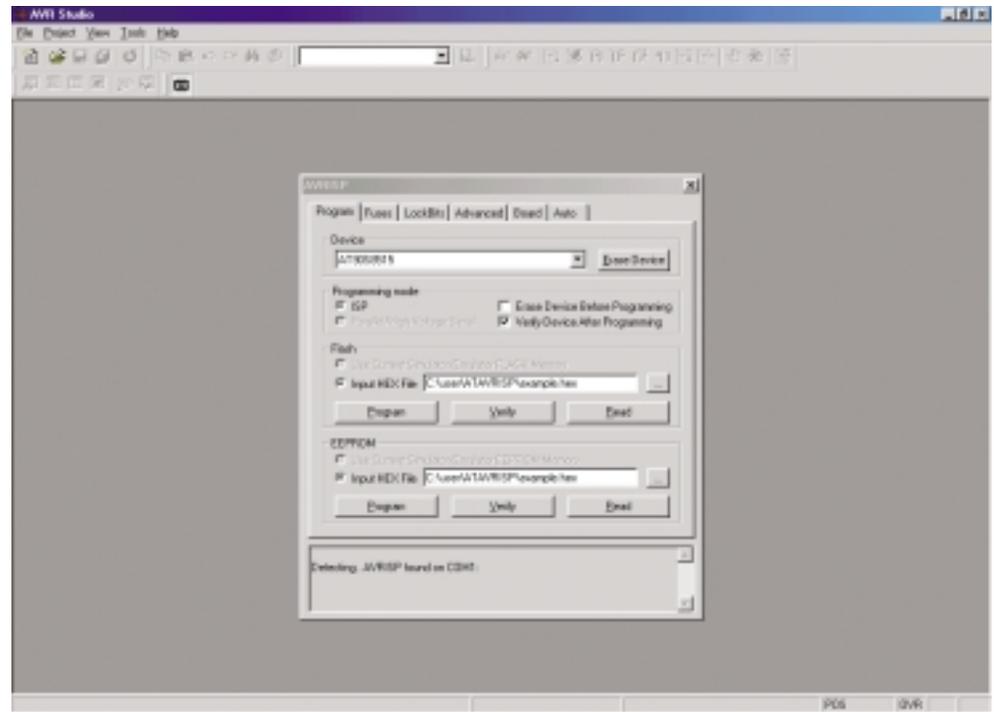


## Section 4

# Software Front-end

- 
- 4.1 Installing AVR Studio**
- AVR Studio with its Integrated Development Environment (IDE) is the ideal software for all AVR developments. It has an editor, an assembler, a debugger and is front-end for all AVR emulators, STK500 and the AVRISP In-System Programmer. AVRISP uses the same programming interface as the STK500.
- To install AVR Studio insert the supplied Atmel CD-ROM Databook in the computer and navigate to “Products → AVR 8-bit RISC → Software”. Right click with the mouse on the “avrstudio3.exe” file and select "save link as". Select an empty directory and save the file.
- Execute the “avrstudio3.exe” file, this is a self-extracting file that will extract all required files to the current directory. Execute the “Setup.exe” file, this will guide you through the setup process.
- Note: AVR Studio version 3.5 or higher is required for AVRISP support.
- AVR Studio is continuously updated to support new devices and to add functionality. The latest version of AVR Studio can be downloaded from [www.atmel.com](http://www.atmel.com).
- 
- 4.2 Using AVR Studio**
- It is assumed that the reader has general knowledge of how to use AVR Studio. For more information on general use of the program, please look in the interactive help system supplied with AVR Studio. This user guide covers AVRISP specific topics only.
- In this section the supporting software for AVRISP will be presented, and an in-depth description of the available programming options given.
- 4.2.1 Starting the Windows Software**
- The software used for communicating with the AVRISP is included in AVR Studio version 3.5 or later. Once installed, double clicking on the  icon starts AVR Studio. If default install options are used, the program is located in the Windows Start Menu → Programs → Atmel AVR Tools folder.
- 4.2.2 Starting AVRISP Interface**
- Pressing the  button on the AVR Studio toolbar will start the AVRISP user interface as shown in Figure 4-1.

Figure 4-1. AVR Studio with AVRISP User Interface



Note that the same interface is used for both STK500 and AVRISP. Since STK500 includes features that are not supported in the AVRISP, some features are not selectable when using the AVRISP interface. Only supported features are selectable.

#### 4.2.3 Using AVRISP and STK500 Simultaneously

When AVR Studio is scanning for connected devices, it searches through the COM ports in a sequential manner. The first device encountered, will gain control over the COM port. It is not possible to control both, a STK500 and an AVRISP, from AVR Studio simultaneously. To do this two instances of AVR Studio must be executed simultaneously. The title bar on the Programming interface will indicate whether it controls the AVRISP or the STK500.

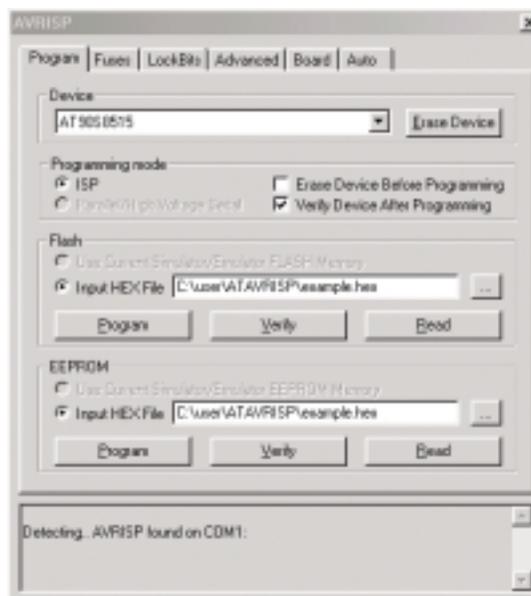
### 4.3 AVRISP User Interface

The AVRISP User Interface includes a lot of powerful features for the AVRISP In-System Programmer. The available settings are divided into six groups, each selectable by pressing on the appropriate tab. Since different devices have different features, the available options and selections will depend on which device is selected. Unavailable features are grayed out.

#### 4.3.1 “Program” Settings

The program settings are divided into four different sub groups.

Figure 4-2. Program Settings



- 4.3.1.1 Device** Device is chosen by selecting the correct device from the Pull-down menu. This group also includes a button that performs a chip-erase on the selected device, erasing both the FLASH and EEPROM memories.
- 4.3.1.2 Programming Mode** This group selects programming mode. AVRISP only supports the ISP low-voltage mode. Checking the “Erase Device Before Programming” will force AVRISP to perform a chip-erase before programming the device. Checking the “Verify Device After Programming” will force AVRISP to perform a verification of the memories after programming.
- 4.3.1.3 Flash** If the AVRISP User Interface is opened without a project loaded in AVR Studio, the “Use Current Simulator/Emulator Flash Memory” option will be grayed out. When a project is open this option allows programming of the Flash memory content currently present in the Flash Memory view of AVR Studio. For more information about AVR Studio memory views, please take a look in the AVR Studio help system.
- If no project is running, or the source code is stored in a separate HEX file, select the “Input HEX File” option. Browse to the correct file by pressing the  button, or write the complete path and filename in the text field. The selected file must be in “Intel-hex” format or “extended Intel-hex” format.
- 4.3.1.4 EEPROM** If the AVRISP User Interface is opened without a project loaded in AVR Studio, the “Use Current Simulator/Emulator EEPROM Memory” option will be grayed out. When a project is open this option allows programming of the EEPROM memory content currently present in the EEPROM Memory view. For more information about AVR Studio memory views, please take a look in the AVR Studio help system.
- If no project is running, or the source code is stored in a separate HEX file, select the “Input HEX File” option. Browse to the correct file by pressing the  button, or write the complete path and filename in the text field. The selected file must be in “Intel-hex” format or “extended Intel-hex” format.
- 4.3.2 "Fuses" Settings** On the “Fuses Settings” tab, an overview of accessible fuses are presented. Some fuses are only available during Parallel /High-voltage programming. These will be displayed, but are not accessible when operating in ISP programming mode. Press the “Read” button to read the current value of the fuses, and the “Write” button to write the

current fuse setting to the device. Checking one of these check-boxes indicates that this fuse should be enabled/programmed, which means writing a “zero” to the fuse location in the actual device. Note that the selected fuse setting is not affected by erasing the device with a chip-erase cycle (i.e., pressing “Chip Erase” button in the “Program” settings).

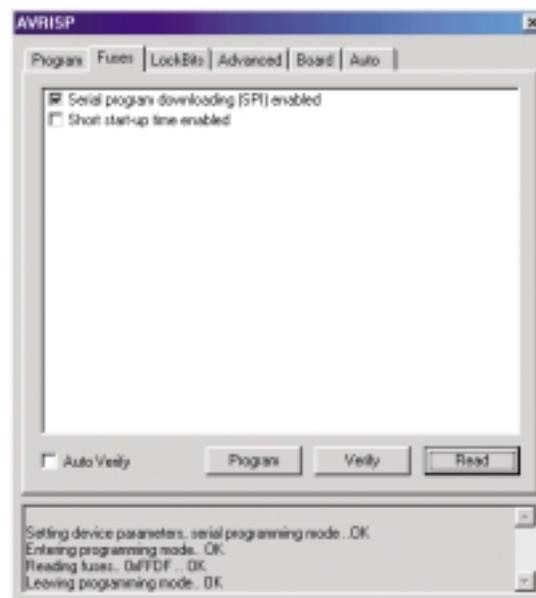
**Table 4-1.** Check Box Description

Icon	Description
<input type="checkbox"/>	Unprogrammed fuse or lockbit
<input checked="" type="checkbox"/>	Programmed fuse or lockbit
<input checked="" type="checkbox"/> 	Readback of current state is impossible, but fuse or lock can be programmed. Set to be programmed.
<input type="checkbox"/> 	Readback of current state is impossible, but fuse or lock can be programmed. Not set to be programmed.
<input type="checkbox"/>	Readback of current value indicated unprogrammed lock or fuse bit, but no access is possible. (I.e., can not be changed in serial mode.)
<input checked="" type="checkbox"/>	Readback of current value indicated programmed lock or fuse bit, but no access is possible. (I.e., cannot be changed in serial mode.)
<input type="checkbox"/> 	Fuse or lock bit is not accessible, and read back is impossible.

Detailed information on which fuses are available in the different programming modes and their functions can be found in the appropriate device datasheet. By checking the “Auto Verify” check box, a verification will be automatically performed after each programming.

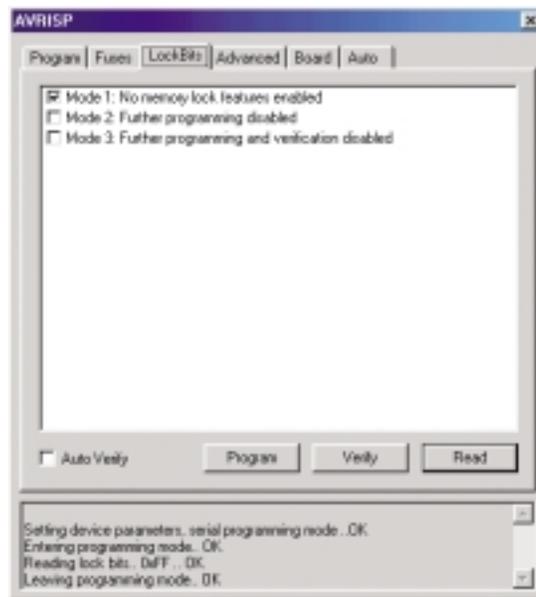
Please see the “Special Considerations” in Section 6 if you plan to change the RST-DISBL or SPIEN fuse.

**Figure 4-3.** Fuses Settings



- 4.3.3 “Lock Bits” Settings** Similar to the “Fuses” settings, the “Lock Bits” tab shows which lock modes are applicable to the selected device. All lock bits are accessible in ISP programming mode. A lock mode may consist of a combination of multiple lock bits. The AVRISP User Interface handles this, and the correct lock bits are programmed automatically for the selected Lock mode. Once a Lock mode protection level is enabled, it is not possible to lower the protection level by selecting a lower degree of protection or by setting a different Lock mode. The only way of removing a programmed lock bit is to do a complete chip-erase, erasing both Program and EEPROM memories. One exception exists: If the target device has a programmed “EESAVE” fuse, the contents of the EEPROM will be kept even when a complete chip erase on the device is performed. By checking the “Auto Verify” check box a verification will be automatically preformed after each programming.

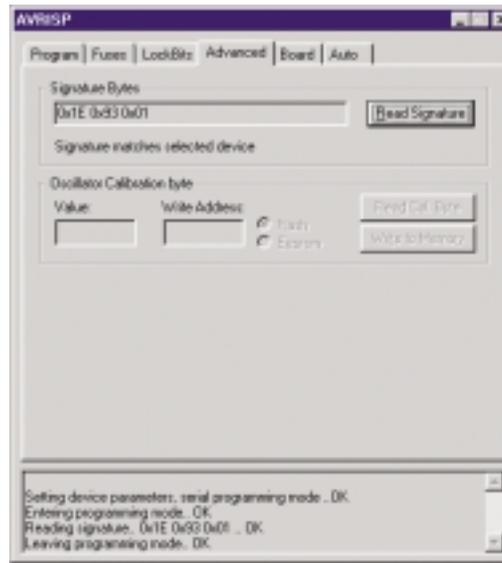
**Figure 4-4.** Lock Bits Settings



4.3.4 “Advanced” Settings

The Advanced tab is currently divided into two sub groups.

Figure 4-5. Advanced Settings



4.3.4.1 Signature Bytes

By pressing the “Read Signature” button, the signature bytes are read from the target device. The signature bytes act like an identifier for the part. Please refer to the AVR datasheets for more information about signature bytes.

4.3.4.2 Oscillator Calibration Byte

For devices with calibratable Internal RC Oscillator, the oscillator calibration byte is written to the device during manufacturing, and cannot be erased or altered by the user. The calibration byte is a tuning value that should be written to the OSCCAL register in order to tune the internal RC to specified frequency.

4.3.4.3 Reading Oscillator Calibration Byte

By pressing the “Read Cal. Byte” button, the calibration value is read from the device and shown in the “Value” text box. If this option is grayed out, the selected device does not have a tunable internal RC Oscillator. On selected devices, the RC oscillator is self-calibrating. On these devices there is no need to handle the Calibration byte manually (for more information see appropriate device datasheet).

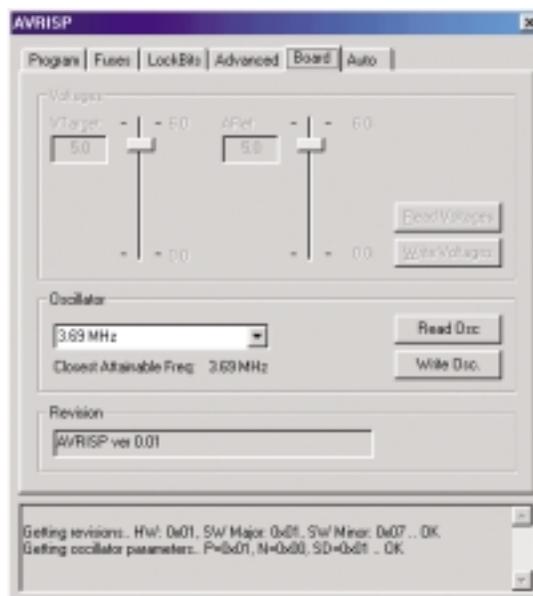
4.3.4.4 Writing Oscillator Calibration Byte

Since the calibration byte is not directly accessible during program execution on devices without automatic RC calibration, the user should write the calibration byte into a known location in Flash or EEPROM memory. Do this by writing the desired memory address in the “Write Address” text box and then press the “Write to Memory” button. The calibration byte is then written to the memory indicated by the “Flash” and “EEPROM” radio buttons.

4.3.5 “Board” Settings

The Board tab allows changing the operating conditions for the AVRISP programmer. The AVRISP allows modification of the Oscillator frequency.

Figure 4-6. Board Settings



#### 4.3.5.1 Oscillator

The AVRISP programmer uses a programmable oscillator circuit that offers a wide range of frequencies for the target device.

The frequency given here should be lower or equal to the frequency of the target AVR to be programmed. Based on this number, the AVRISP User Interface calculates the communication speed between the AVRISP and the target AVR.

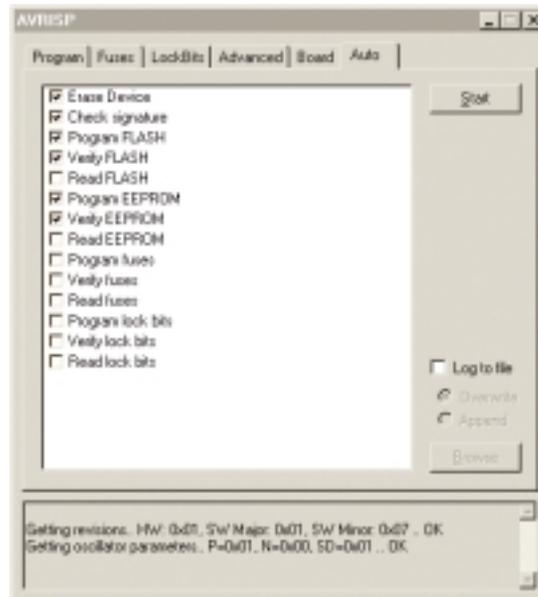
Note that the drop-down list only have a few selectable frequencies. For any other frequency, write it in the input field and press "Write Osc". The frequency will then be set to the closest attainable and displayed in the field.

#### 4.3.5.2 Revision

In the revision box the current revision AVRISP revision number is shown. If AVR Studio discovers that the AVRISP contains an older version than the one distributed with AVR Studio, it will automatically ask for permission to upgrade the Programmer.

#### 4.3.6 "Auto" Settings

When programming multiple devices with the same code, the Auto tab offers a powerful method of automatically going through a user-defined sequence of commands. The commands are listed in the order they are executed (if selected). To enable a command, the appropriate check box should be checked (e.g., if only "Program FLASH" is checked, by pressing the "Start" button the FLASH memory will be programmed with the HEX file specified in the "Program" settings). All commands depend on, and use, the settings given in the AVRISP User Interface. It is possible to log the command execution to a text file by checking the "Log to file" check box.

**Figure 4-7.** Automatic Programming Settings

#### 4.3.6.1 Setting up the System for Auto Programming

Click on the check boxes for the commands the AVRISP User Interface should perform. A typical sequence where the device is erased and then programmed is shown in Figure 4-7. The chip is erased, both memories programmed and verified.

Once configured, the same programming sequence is executed every time the “Start” button is pressed. This reduces both work and possibilities for errors due to operational errors.

#### 4.3.6.2 Logging the Auto Programming to a File

By clicking on the “Log to File” check box all output from the commands are written to a text file. The file is selected/created by pressing the “Browse” button. Navigate to the location where the file is placed, or should be created. The output is directed to this file, and can be viewed and edited using a standard text editor. An existing file will be overwritten.

#### 4.3.7 History Window

The History window is located at the bottom of the AVRISP view. In this window the dialog between AVR Studio and AVRISP is shown. For every new command performed, the old dialog is replaced with the new one.



## Section 5

# Command Line Software

The DOS command line of the programming interface is the same as for the STK500 starter kit. It is named “stk500.exe” and allows simple batch files for automatic programming.

In the following text shows how to make simple batch files for automating the programming steps for a device.

Synopsis: STK500

```
[-d device name] [-m s|p] [-if infile] [-ie infile] [-of  
outfile] [-oe outfile] [-s] [-O] [-Sf addr] [-Seaddr] [-e] [-p  
f|e|b] [-r f|e|b] [-v f|e|b] [-l value] [-L value] [-y] [-f  
value] [-F value] [-q] [-x value] [-af start,stop] [-ae  
start,stop] [-c port] [-ut value] [-ua value] [-wt] [-wa] [-j  
value] [-b h|s] [-! freq] [-&] [-t p|t] [-t] [-n] [-g] [-z] [-  
U] [-h|?]
```

---

<b>5.1</b>	<b>Parameters:</b>	d	Device name. Must be applied when programming the device. See list below.
		m	Select programming mode; serial (s) or parallel (p). Serial programming mode is the default, and is used if this parameter is not applied.
		if	Name of Flash input file. Required for programming or verification of the Flash memory. The file format is Intel Extended HEX.
		ie	Name of EEPROM input file. Required for programming or verification of the EEPROM memory. The file format is Intel Extended HEX.
		of	Name of Flash output file. Required for readout of the Flash memory. The file format is Intel Extended HEX.
		oe	Name of EEPROM output file. Required for readout of the EEPROM memory. The file format is Intel Extended HEX.
		s	Read signature bytes.
		O	Read oscillator calibration byte.
		Sf	Write oscillator calibration byte to Flash memory, “addr” is byte address. The byte is automatically verified after the write.

- Se Write oscillator calibration byte to EEPROM memory, "addr" is byte address. The byte is automatically verified after the write.
- e Erase device. If applied with another programming parameter, the device will be erased before any other programming takes place.
- p Program device; FLASH (f), EEPROM (e) or both (b). Corresponding input files are required.
- r Read out device; Flash (f), EEPROM (e) or both (b). Corresponding output files are required
- v Verify device; Flash (f), EEPROM (e) or both (b). Can be used with -p or stand-alone. Corresponding input files are required.
- l Set lock byte - "value" is an 8-bit hex. value.
- L Verify lock byte - "value" is an 8-bit hex. value to verify against.
- y Read back lock byte.
- f Set fuse bytes. 'value' is a 16-bit hex. value describing the settings for the upper and lower fuse.
- F Verify fuse bytes - "value" is a 16-bit hex. value to verify against.
- q Read back fuse bytes.
- x Fill unspecified locations with a value (0x00-0xff). The default is to not program locations not specified in the input files.
- af Flash address range. Specifies the address range of operations. The default is the entire Flash. Byte addresses.
- ae EEPROM address range. Specifies the address range of operations. The default is the entire EEPROM. Byte addresses.
- c Select communication port; "com1" to "com8". If this parameter is omitted the program will scan the communication ports for the AVRISP.
- ut Set target voltage (VTARGET) in Volts. "value" is a floating point value between 0.0 and 6.0, describing the new voltage.
- ua Set adjustable voltage (AREF) in Volts. "value" is a floating point value between 0.0 and 6.0, describing the new voltage.
- wt Get current target voltage VTARGET.
- wa Get current adjustable voltage AREF.
- b Get revisions; hardware revision (h) and software revision (s).
- ! Set oscillator frequency; "freq" is the frequency in Hz.
- & Get oscillator frequency.
- t Get currently selected device parameters.
- n Get current programming mode.

- g Silent operation.
- z No progress indicator (e.g., if piping to a file for log purposes, use this option to avoid the non-ascii characters used for the indicator.
- h|? Help information (overrides all other settings).

Since the interface is also used for the STK500 Starter Kit, not all listed switches are applicable to the AVRISP. STK500 software will give an error-message if an unsupported switch is used.

### 5.1.1 Sample Usage

Erase, program and verify the flash of an AT90S8515

```
STK500 -dAT90S8515 -ms -e -pf -vf -iftest.hex
```

Erase, program and verify the EEPROM of an AT90S/LS4433

```
STK500 -dAT90S4433 -ms -e -pe -ve -ietest.hex
```





## Section 6

---

# Special Considerations

There are a few special considerations that should be noted when using this AVRISP programmer For In-System Programing of AVR Devices.

- 
- |            |  |  |
|------------|--|--|
| <b>6.1</b> | <b>Fuse Programming</b>                          | Some devices have fuses not accessible in ISP mode. To program these fuses a parallel programmer is needed. Some of the tinyAVR™ devices allow access to the SPIEN and RSTDISBL fuses. Unprogramming/programming these fuses will disable further ISP programming. |
| <b>6.2</b> | <b>RESET Used as General IO Port</b>             | If the RESET pin is used as a general purpose IO, In-System Programming is not possible. The reason is that the RESET pin must pulled to 12V to enter High Voltage Serial Programming mode (HVSP). HVSP must be used to change the RSTDSBL fuse.                   |
| <b>6.3</b> | <b>AVR Devices with no ISP Option</b>            | Some devices do not have an ISP programming option (e.g., ATtiny28). To program these devices, a parallel programmer is required. (e.g STK500 Starter Kit). Only devices with low-voltage ISP programming mode are supported by the AVRISP.                        |
| <b>6.4</b> | <b>Devices without RC Oscillator Calibration</b> | Not all devices with internal RC clock option feature Oscillator calibration. For these devices the internal RC will run at the default speed as indicated by the appropriate datasheet.   |
-





## Section 7

# Troubleshooting Guide

**Table 7-1.** Troubleshooting Guide

Problem	Reason	Solution
The LED is not lit	AVRISP is not connected to target	Connect ISP cable to target board
	ISP pinout is not correct	Verify pinout on target ISP header connector
	Target does not provide enough power	Verify that the target power supply can deliver enough power
Can't get any communication with target device	ISP pinout is not correct	Verify pinout on target ISP header connector
	Device does not support ISP programming mode	Verify that device supports ISP mode, and that correct IO pins are connected
	Heavy loading on ISP pins	Connect series resistors between load and IO pins as shown in Figure 3-2
	Too strong pullup on RESET pin	Reset pullup resistor should be more than 10 k $\Omega$
	Target frequency set wrong in AVR Studio	Reduce the frequency in AVR Studio to match the target board frequency
	Target does not provide enough power	Verify that the target power supply can deliver enough power to source both application and AVRISP
	SPIEN fuse disabled	Enable SPIEN fuse with a Parallel/High Voltage Serial programmer
	Reset used as general IO	Use a High Voltage Serial Programmer /Parallel programmer to Change the RSTDISBL fuse
AVR Studio does not find AVRISP	Old version of AVR Studio	Download AVR studio version 3.5 or higher from Atmel's web site
	Other Device or service controls the COM port	Disable Mouse Drivers IRDA drivers or other devices that takes control of the COM port





## Atmel Headquarters

*Corporate Headquarters*  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### *Europe*

Atmel SarL  
Route des Arsenaux 41  
Casa Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

### *Asia*

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *Japan*

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Operations

*Atmel Colorado Springs*  
1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

*Atmel Rousset*  
Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

*Atmel Smart Card ICs*  
Scottish Enterprise Technology Park  
East Kilbride, Scotland G75 0QR  
TEL (44) 1355-357-000  
FAX (44) 1355-242-743

*Atmel Grenoble*  
Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex  
France  
TEL (33) 4-7658-3243  
FAX (33) 4-7658-3320

---

### *Fax-on-Demand*

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

*e-mail*  
literature@atmel.com

*Web Site*  
<http://www.atmel.com>

*BBS*  
1-(408) 436-4309

### © Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.





## Section 8

---

# Technical Specifications

### System Unit

Physical Dimensions . . . . . 60 x 75 x 27 mm

Weight . . . . . 50 g.

### Operating Conditions

Voltage Range . . . . . 2.7V - 5.5V

Target Freq Range . . . . . higher than 8 kHz

Max I @ 5.5V target voltage . . . . . 50 mA

Max I @ 3.3V target voltage . . . . . 15 mA

### Connections

Serial Connector . . . . . 9-pin D-SUB Female

Serial Communications Speed . . . . . 115200 bits/second

ISP connector . . . . . 3 x 2 and 5 x Male connector 2.54 mm spacing





## Section 9

---

# Technical Support

For Technical support, please contact [avr@atmel.com](mailto:avr@atmel.com). When requesting technical support for AVRISP, please include the following information:

- Version number of AVR Studio (this can be found in the AVR Studio help menu).
- PC processor type and speed
- PC operating system and version
- What target AVR device is used (complete part number)
- Programming Voltage and speed
- A detailed description of the problem





## Atmel Headquarters

*Corporate Headquarters*  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### *Europe*

Atmel SarL  
Route des Arsenaux 41  
Casa Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

### *Asia*

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *Japan*

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Product Operations

### *Atmel Colorado Springs*

1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### *Atmel Grenoble*

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-7658-3000  
FAX (33) 4-7658-3480

### *Atmel Heilbronn*

Theresienstrasse 2  
POB 3535  
D-74025 Heilbronn, Germany  
TEL (49) 71 31 67 25 94  
FAX (49) 71 31 67 24 23

### *Atmel Nantes*

La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
TEL (33) 0 2 40 18 18 18  
FAX (33) 0 2 40 18 19 60

### *Atmel Rousset*

Zone Industrielle  
13106 Rousset Cedex, France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

### *Atmel Smart Card ICs*

Scottish Enterprise Technology Park  
East Kilbride, Scotland G75 0QR  
TEL (44) 1355-357-000  
FAX (44) 1355-242-743

---

*e-mail*  
[literature@atmel.com](mailto:literature@atmel.com)

*Web Site*  
<http://www.atmel.com>

*BBS*  
1-(408) 436-4309

### © Atmel Corporation 2001.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Atmel®, AVR® and AVR Studio® are the registered trademarks of Atmel. ICE10™ is the trademark of Atmel.

Windows® 95/98/2000 and Windows NT® are registered trademarks of Microsoft Corporation. Terms and product names may be trademarks of others.



Printed on recycled paper.