EN: This Datasheet/Document is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.

AVRISP mkII User Guide

See AVR Studio 4 online help for updated and complete information.

Introduction

- Introduction
- Getting Started
- Software and USB Setup
- Supported Devices

Frontend Software

- Using AVR Studio with AVRISP mkII
  - Program
  - Fuses
  - Lock Bits
  - Advanced
  - Board
  - Auto
- Command Line Software

Hardware

- Connecting AVRISP mkII
- Hardware Description

Troubleshooting

- Troubleshooting Guide
- Manual Firmware Upgrade
- Known Issues
- Technical Support

Table Of Contents

AVRISP MKII USER GUIDE 1

TABLE OF CONTENTS 2

INTRODUCTION 3

GETTING STARTED 4

USB Setup 4
Software and USB Setup 6
Install new hardware on the computer 7
Install USB driver after AVR Studio is installed 7
Frontend Software 9
Program 11
Fuses 13
Lock Bits 14
Advanced 15
Board 17
Auto 19

COMMAND LINE SOFTWARE 21

HARDWARE 24

Connecting AVRISP mkII 24
Hardware Description 25
Target Interface 26

TROUBLESHOOTING 27

MANUAL FIRMWARE UPGRADE 28

KNOWN ISSUES 29

TECHNICAL SUPPORT 30

Introduction

The AVRISP mkII combined with AVR Studio® can program all AVR® 8-bit RISC microcontrollers with ISP Interface. Click here for a list of supported devices.

AVRISP mkII features

- AVR Studio compatible (AVR Studio 4.12 or later)
- Supports all AVR devices with ISP interface
- Programs both flash and EEPROM
- Supports fuses and lock bit programming
- Upgradeable to support future devices
- Support target voltages from 1.8V to 5.5V
- Adjustable programming speed (50Hz to 8MHz SCK frequency)
- USB 2.0 compliant (full speed, 12Mbps)
- Powered from USB, does not require external power supply
- Target interface protection
- Short-circuit protection

The AVRISP mkII is supported by AVR Studio (see AVR Studio Requirements). Updated versions of AVR Studio is found on http://www.atmel.com/products/AVR/

• Please read Getting Started using the AVRISP mkII

Getting Started

Please read this section before connecting the AVRISP mkII to the computer or target.
Follow these four steps to get started using the AVRISP mkII:

1. Install AVR Studio and the USB driver
2. Connect AVRISP mkII to the computer, and auto-install new hardware (AVRISP mkII) on the computer
3. Start AVR Studio and the AVRISP mkII Programming Dialog
4. Connect AVRISP mkII to the target

USB Setup

In order to use the AVRISP mkII it is required to install the AVR Studio and USB driver first. Please do not connect the AVRISP mkII to the computer before running the USB Setup in order to follow this procedure described in Software and USB Setup.

AVRISP mkII Content

The box should contain the following items:

- AVRISP mkII
- USB cable
- Atmel Technical Library CD-ROM with datasheets, application notes and software.

System Requirements

The minimum hardware and software requirements are:

- Pentium (Pentium II and above is recommended)
- 64 MB RAM
- 100 MB Free Hard Disk Space for installation of AVR Studio 4.12 or later
- Windows® 98, Windows ME, Windows® 2000 or Windows® XP
- USB port, self-powered (200mA required)

Note: Windows 95 and Windows NT does not support USB, hence cannot be used with AVRISP mkII.

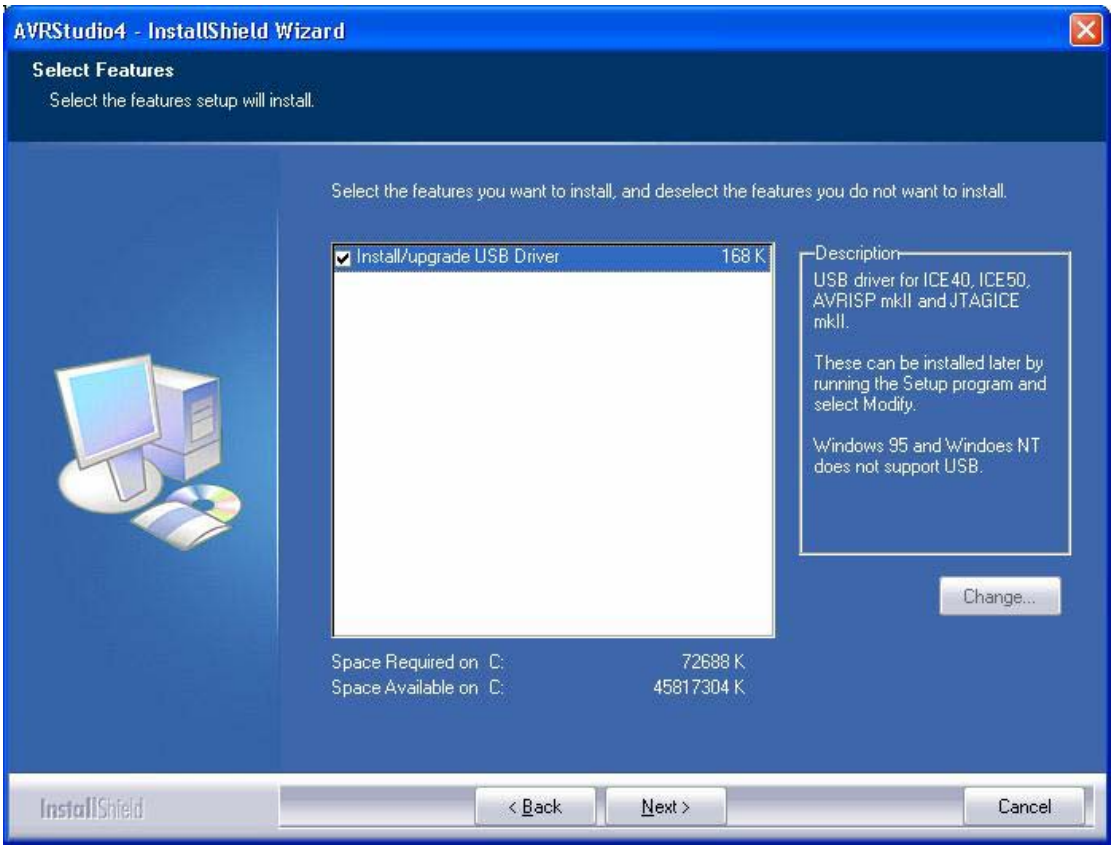
AVR Studio Requirements

It is required to use AVR Studio 4.12 or later in order to use the AVRISP mkII. Latest version of the AVR Studio can be found at: www.atmel.com/products/AVR/

Note: AVR Studio and the USB driver must be installed before the AVRISP mkII is connected to the computer.

Software and USB Setup

In order to use the AVRISP mkII it is required to install the USB driver. Please do not connect the AVRISP mkII to the computer before running the USB Setup. USB driver installation is done during the AVR Studio installation.
Note: AVRISP mkII requires AVR Studio 4.12 or later. Latest version of the AVR Studio can be found at: www.atmel.com/products/AVR/
Start the AVR Studio installation. During this installation the dialog box in the figure below will be presented to the user.



To install the USB driver, check the Install/Upgrade USB Driver checkbox, and the USB Driver will automatically be installed.

Install new hardware on the computer

When AVR Studio and USB driver installation is finished, please attach the USB cable to both PC and AVRISP mkII. (The AVRISP mkII is powered from the USB). If it is the first time the AVRISP mkII is connected to the computer, the box below will appear:

If running Windows XP you need to click "Next" a couple of times. Please wait until the installation process completes by itself. It may take from a few seconds up to a few minutes depending on the computer and operating system.
If the USB driver is correctly installed and AVRISP mkII is connect to the PC, the green LED inside the encasing next to the USB connector will be lit.
If the AVR Studio for some reason can't detect the AVRISP mkII after the USB setup, try to restart the computer in order to get the driver properly loaded.

Install USB driver after AVR Studio is installed

The USB driver can be installed even after AVR Studio have been installed by following these steps:

1. Open "Control Panel" on the PC (Windows 95 and Windows NT does not support USB)
2. Select "Add or Remove Programs"
3. Select "AVRStudio4" in the list of programs
4. Click on the "Change" button
5. Select "Modify"
6. Select "Install/upgrade USB Driver"

The USB driver is now properly installed on the PC

Note: The AVRISP mkII requires a USB port that can deliver 200mA (self-powered USB hub). • Please read about using AVR Studio with AVRISP mkII

Supported Devices

AVRISP mkII supports all AVR 8-bit RISC Micro Controllers with ISP programming interface. Support for new devices will be added with new versions of AVR Studio. The latest version of AVR Studio is always available from www.atmel.com/products/AVR/
When new firmware is available, AVR Studio will prompt the user to upgrade AVRISP mkII
The following devices are supported:

Tiny	**Mega**	**Other**
ATtiny12	ATmega48	AT86RF401
ATtiny13	ATmega8	AT89S51
ATtiny15	ATmega88	AT89S52
ATtiny22	ATmega8515	AT90PWM2
ATiny2313	ATmega8535	AT90PWM3
ATtiny24	ATmega16	AT90CAN32
ATtiny26	ATmega162	AT90CAN128
ATtiny45	ATmega165	
ATtiny861	ATmega168	
	ATmega169	
	ATmega32	
	ATmega325	
	ATmega3250	
	ATmega329	
	ATmega3290	
	ATmega64	
	ATmega640	
	ATmega644	
	ATmega645	
	ATmega6450	
	ATmega649	
	ATmega6490	
	ATmega128	
	ATmega1280	
	ATmega1281	
	ATmega2560	
	ATmega2561	

The AVRISP mkII supports all the different voltages and speed grades versions of the devices listed in the table above.

Frontend Software

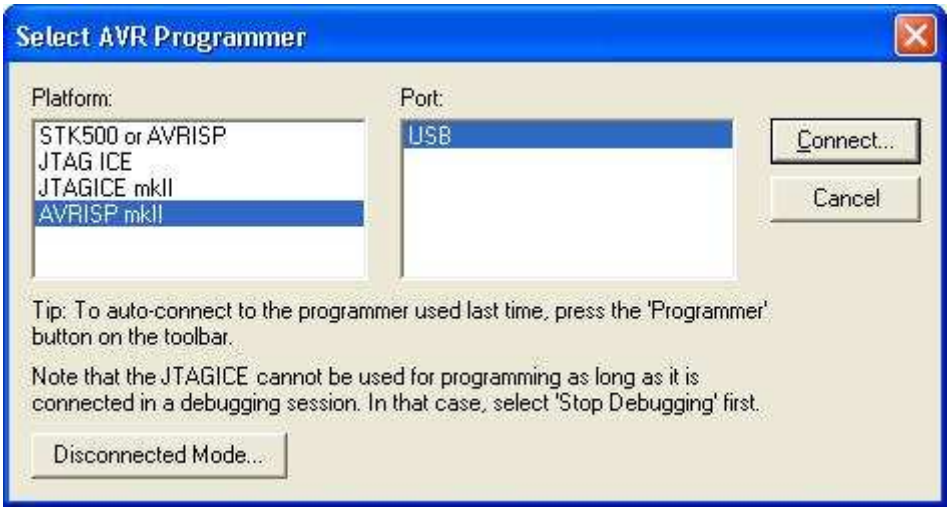
Using the AVR Studio with AVRISP mkII

AVR Studio is the user interface for the AVRISP mkII. The latest version of AVR Studio can be found at www.atmel.com/AVR/

Note: The USB driver should be installed before connecting AVRISP mkII to the PC. Read more about this in Software and USB Setup.

Connecting

Start the programming dialog by clicking on one of the two connect buttons found on the toolbar in AVR Studio . The button to the left makes it possible to select which programmer and communication port to connect to:



The button to the right will make a "direct" connect to the last chosen tool and communication port. If no tool is found with that setting, it will scan through all the other communication ports to see if any AVR tool can be found there.
If several AVRISP mkII are connected to the same PC, the user can choose which one to use by selecting the serial number. The serial number for each AVRISP mkII can be found on the bottom side of the PCB.
Note: Only one USB-connected tool can be opened for each AVR Studio session, i.e. if several AVRISP mkII are used and/or other AVR tools with USB, they must run from different AVR Studio instances.

Read more about the AVRISP mkII programming dialog:

- Program
- Fuses
- Lock Bits
- Advanced
- Board
- Auto

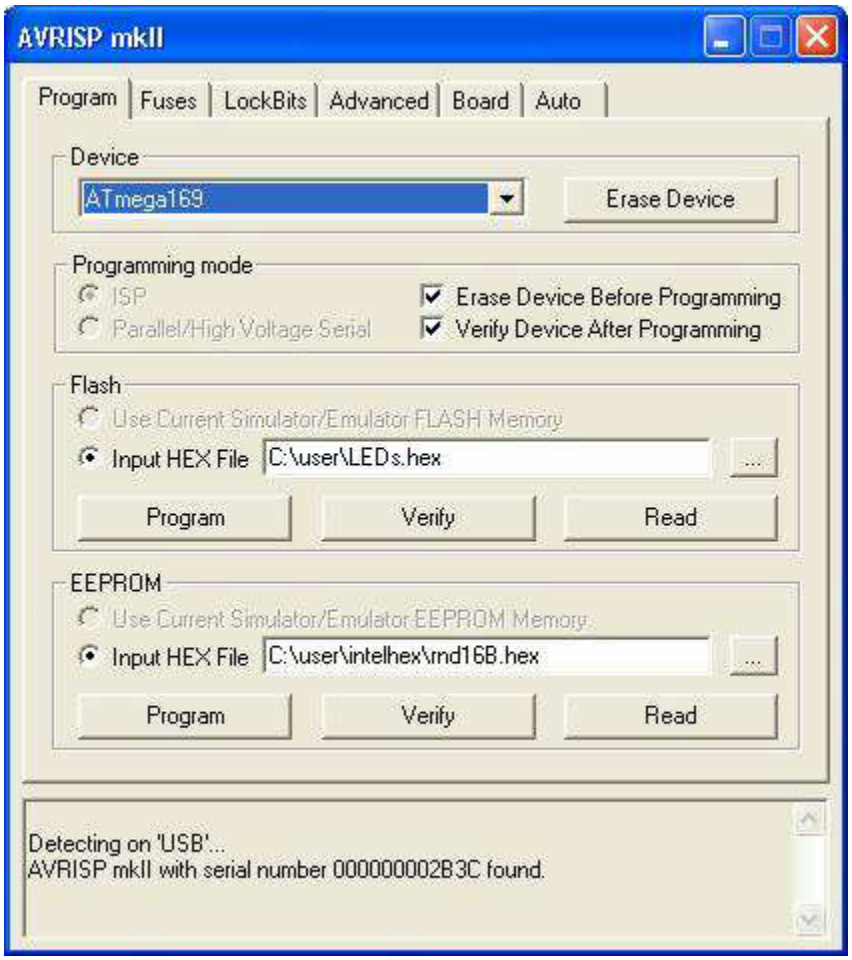
Auto-upgrade

AVR Studio will contain the latest firmware available from Atmel. AVR Studio will automatically detect if newer firmware is available and launch the upgrade program.

See how to connect to the target in the section Connecting AVRISP mkII

Program

The program settings are divided into four different sections.



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 and lock bits.

Programming Mode

This group selects programming mode. AVRISP mkII does only supports the ISP mode. Checking the "Erase Device Before Programming" will force AVRISP mkII to perform a chip-erase before programming the device. Checking "Verify Device After Programming" will force AVRISP mkII to perform a verification of the memories after programming.

Flash

If the AVRISP mkII user interface is opened without a project loaded in AVR Studio, the "Use Current Simulator/Emulator FLASH Memory" option will be disabled. 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

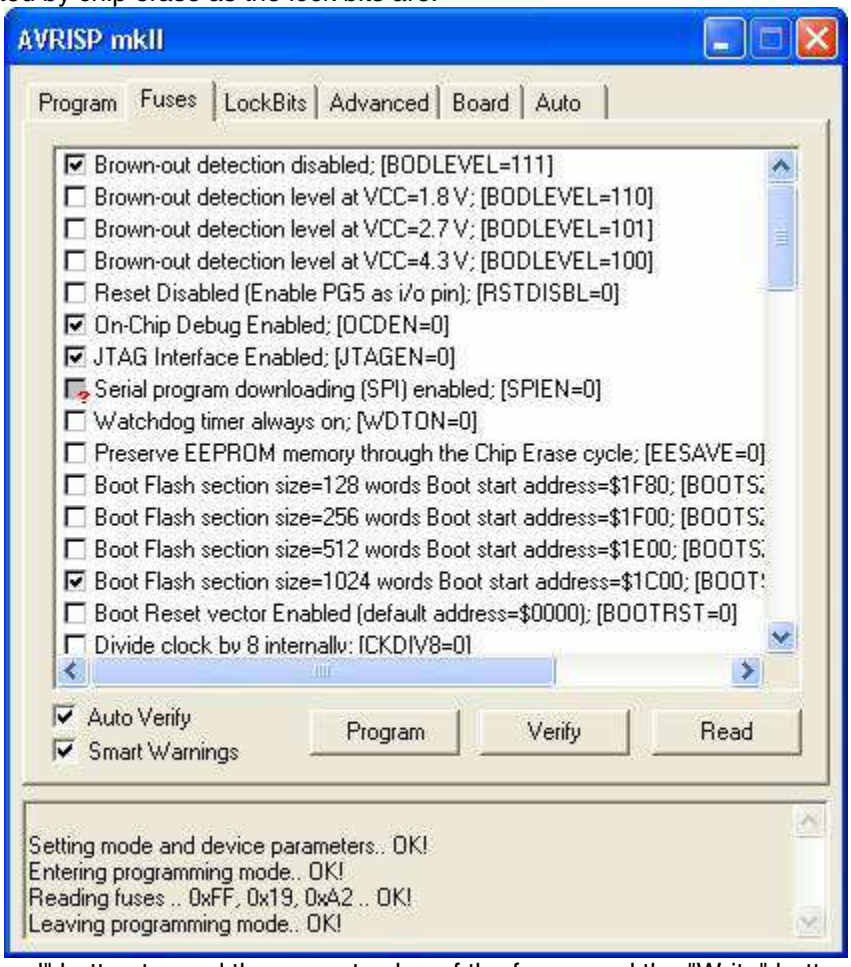
filename in the text field. The selected file must be in "Intel-hex" format or "extended Intel-hex" format.

EEPROM

If the AVRISP mkII user interface is opened without a project loaded in AVR Studio, the "Use Current Simulator/Emulator EEPROM Memory" option will be disabled. 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.

Fuses

On the "Fuses" tab an overview of accessible fuses are presented. A fuse selection may consist of a combination of setting multiple fuse bits. This is handled by AVRISP mkII user interface, and the correct fuse bits are programmed automatically for the selected fuse mode. Note that fuses are not affected by chip erase as the lock bits are.

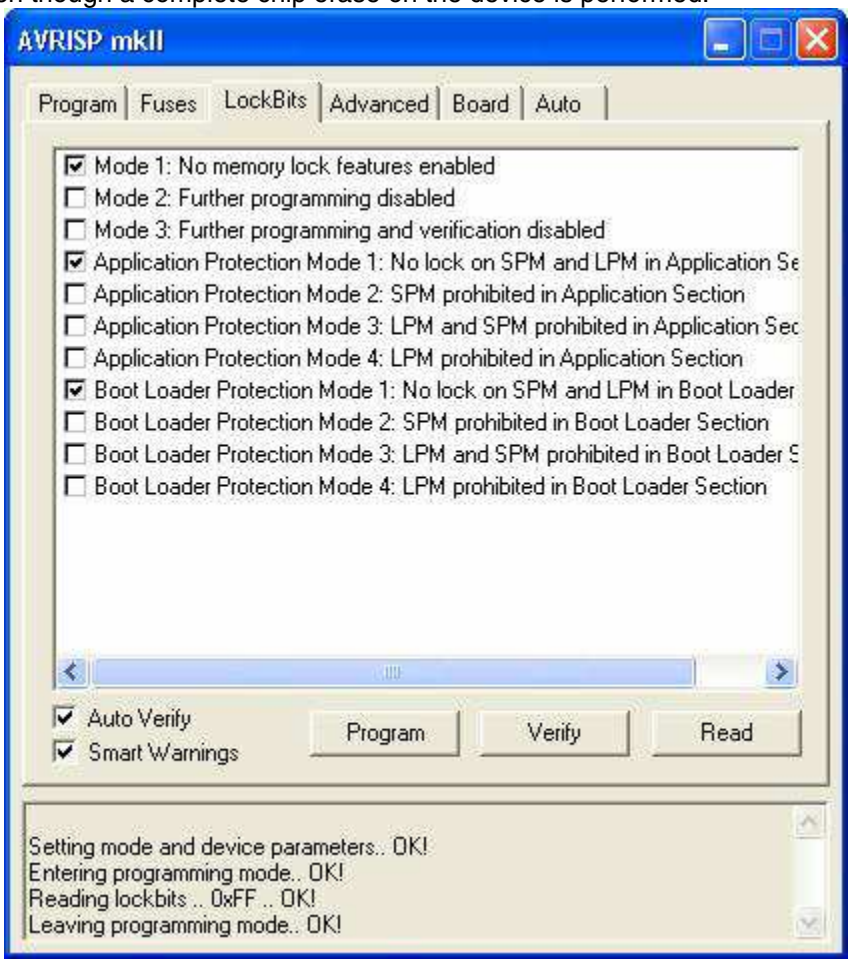


Press the "Read" button to read the current value of the fuses, and the "Write" button to write the current fuse setting to the device. Checking one of these check boxes indicates that this fuse should be enabled/programmed, which means writing a "zero" to the fuse location in the actual device. Note 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).
Detailed information about the fuses and their functions can be found in the appropriate device datasheet.
The text in brackets behind each fuse selection shows the fuse combination and their value to achieve the actual setting. Note that as described in the AVR datasheet a fuse is programmed when written to "0" and unprogrammed when written to "1".
In the bottom status box the fuse values are presented in hexadecimal values with LSB to the left

Lock Bits

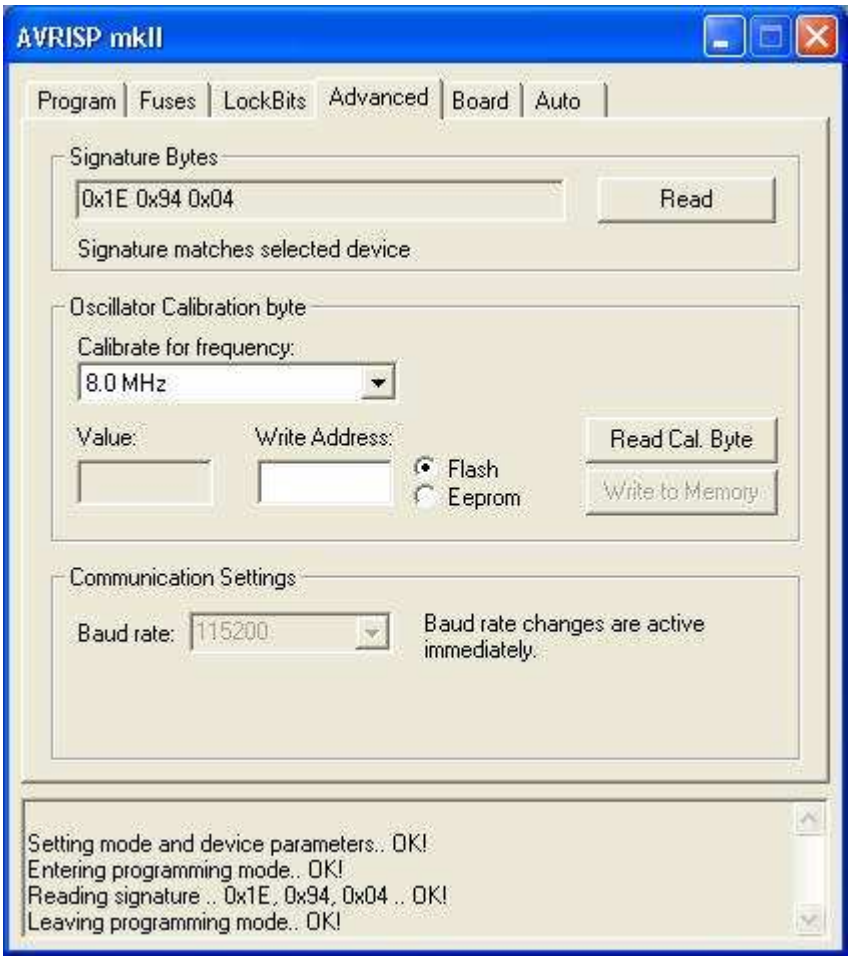
Similar to the "Fuses" settings, the "Lock Bits" tab shows which lock modes that are applicable to the selected device. A lock mode may consist of a combination of multiple lock bits. This is handled by AVRISP mkII user interface, and the correct lock bits are programmed automatically for the selected 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 by setting a different "Lock mode". The only way of removing a programmed lock bit is to perform a chip erase, erasing both program and data memories.

Note: If the target device have a programmed "EESAVE" fuse, the contents of the EEPROM will be saved even though a complete chip erase on the device is performed.



Advanced

The "Advanced" tab is divided into three sub groups.



Signature Bytes

By pressing the "Read Signature" button, the signature bytes are read from the target device. The signature bytes acts like an identifier for the device. After reading the signature bytes, the software will check if it is the correct signature according to the chosen device. Please refer to the AVR datasheets to read more about signature bytes.

Oscillator Calibration Byte

The calibration byte is a tuning value that should be written to the OSCCAL register in order to tune the internal RC Oscillators frequency. The Oscillator calibration byte is written to the device during manufacturing, and can not be erased or altered by the user.

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. Note that on some parts the calibration byte is not directly accessible during program execution and must be written to a memory location during programming if it shall be used by the program. If this option is grayed out, the selected device does not have a tunable internal RC Oscillator.

Writing Oscillator Calibration Byte

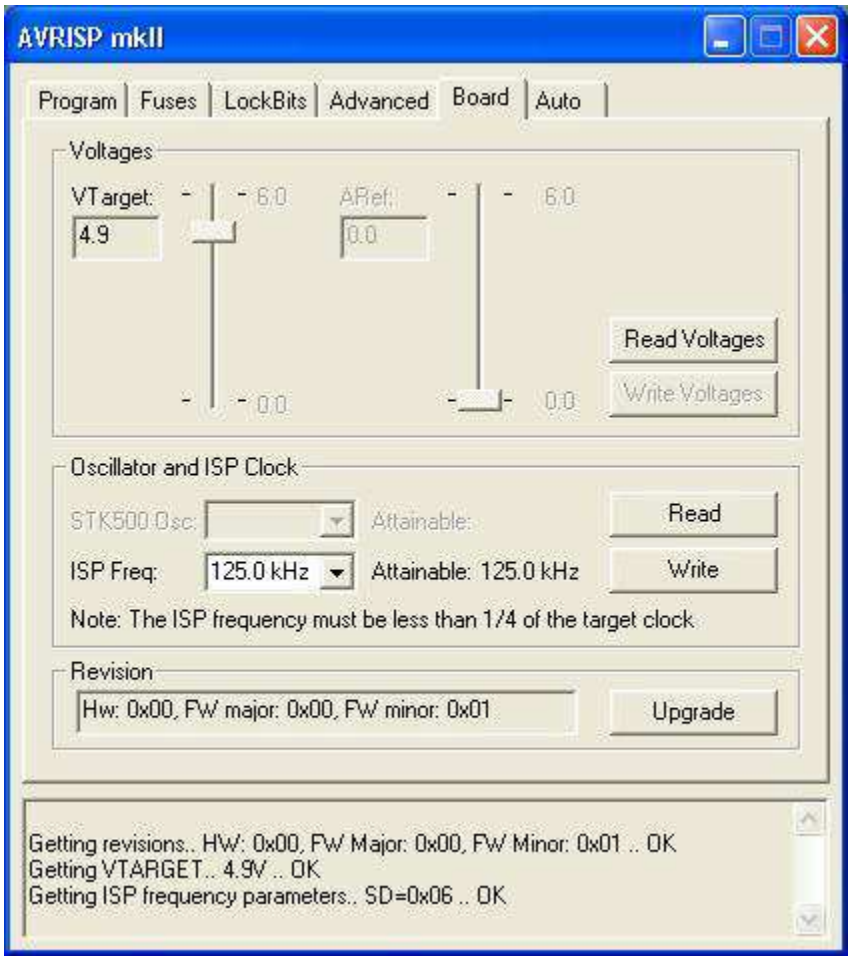
Since the calibration byte is not directly accessible during program execution on some parts, the user should write the calibration byte into a known location in Flash or EEPROM memory. Do this by writing the desired memory address in the "Write Address" text box and then pressing the "Write to Memory" button. The calibration byte is then written to the memory indicated by the "Flash" and "Eeprom" radio buttons.

Communication Settings

Communication Settings does not apply to the AVRISP mkII as it uses USB.

Board

The "Board" tab is divided into three sub groups.



Voltages

VTarget presents the Vcc on the target system.

ARef does not apply to AVRISP mkII as it does not support ARef readings.

Oscillator and ISP Clock

STK500 Osc does not apply to the AVRISP mkII, as it cannot supply the target system with a clock.

ISP Freq sets the ISP clock frequency. This is the SCK signal on the ISP interface.

For most AVR's the datasheet describes the following ISP frequency (SCK):

The minimum low and high periods for the serial clock (SCK) input are defined as follows:
Low:> 2 CPU clock cycles for fck < 12 MHz, 3 CPU clock cycles for fck 12 MHz
High:> 2 CPU clock cycles for fck < 12 MHz, 3 CPU clock cycles for fck 12 MHz

This means that for target clock less than 12MHz, the ISP frequency must be equal or less than 1/4 of the target clock.
For target clock above 12MHz, the ISP frequency must be equal or less than 1/6 of the target clock.

Be aware that for parts with CKDIV (Clock Divide) fuse and/or CLKPR (Clock Prescaler Register), the ISP Frequency must be calculated out from the divided target clock.

Note: If using Internal RC Oscillator, be sure that it is calibrated in order to use the maximum ISP frequency.

Tip: If having problems with connecting to target, try to lower the ISP Frequency.

Please consult the AVR device datasheet.

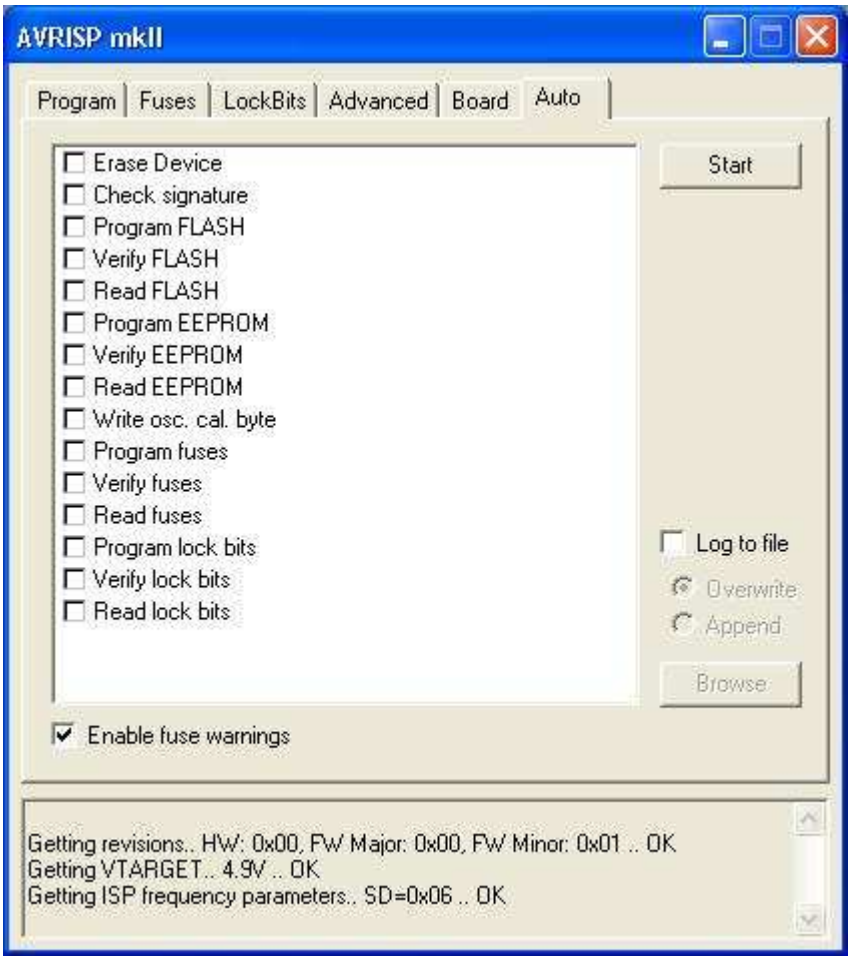
Note: The AVRISP mkII does not support Flash or EEPROM programming below 2kHz. However chip erase will work down to 51Hz. Erase the AVR if it runs to slow to be reprogrammed because of the value in CLKPR.

Revision

The revision shows the Hardware (HW), and Firmware (FW) of the AVRISP mkII.

Auto

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 must 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. The commands use the settings from the other tabs/pages in the AVRISP mkII User Interface.



It is possible to log the command execution to a text file by checking the "Log to file" check box.

Setting up the System for Auto Programming

Click on the check boxes for the commands that you want the AVRISP mkII User Interface to perform. A typical sequence where the device is erased and then programmed is shown in the figure. The chip is erased, both memories programmed and verified, and finally fuses and lock bits are programmed.

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 user errors.

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, and navigate to the location where the file is placed, or should be created. The output is saved to this file, and can be viewed and edited using a text editor.

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 it will be shown how to make simple batch files for automating the programming steps for a device.

STK500 command line programmer, v 2.1c Atmel Corp (C) 2004-2005.

Command Line Switches:

[-d device name] [-m s|p] [-if infile] [-ie infile] [-of outfile]
[-oe outfile] [-s] [-O address] [-Sf addr] [-Seaddr] [-e] [-p f|e|b]
[-r f|e|b] [-v f|e|b] [-l value] [-L value] [-y] [-f value] [-E value]
[-F value] [-G value] [-q] [-Y] [-Z address] [-c port] [-ut value]
[-Wt value] [-ua value] [-wt] [-wa] [-b h|s] [-! freq] [-t] [-I freq]
[-J] [-h] [-?]

Examples:

STK500.EXE -cUSB -dATmega128 -e -ifFlash.hex -pf -vf
STK500.EXE -cUSB -dATmega128 -fF73A -FF73A -EFF -GFF

Note that there are no spaces between switches and their parameters and that hexadecimal values do not have a '0x' prefix.

Parameters:

d	Device name. For a list of supported devices use STK500.EXE -?
m	Select programming mode; serial (s) or parallel/high-voltage (p). Serial mode is the default mode if this parameter is ommitted.
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 callibration byte. 'address' is the address of the calibration byte as decribed in the data sheet of the device.
Sf	Write oscillator call. byte to FLASH memory. 'addr' is byte address
Se	Write oscillator call. byte to EEPROM memory. 'addr' is byte address
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.
E Set extended fuse byte. 'value' is an 8-bit hex. value describing the extend fuse settings.
F Verify fuse bytes. 'value' is a 16-bit hex. value to verify against.
G Verify extended fuse byte. 'value' is an 8-bit hex. value describing the extend fuse settings.
q Read back fuse bytes.
af FLASH address range. Specifies the address range of operations. The default is the entire FLASH. Addresses are byte oriented.
ae EEPROM address range. Specifies the address range of operations. The default is the entire EEPROM. Byte addresses.
Y Perform the oscillator calibration sequence. See appnote AVR053 for more information.
Z Loads a value from an EEPROM location prior to chip erase which then can be programmed into FLASH or EEPROM using the S option. Address is a hex value. See appnote AVR053 for more information.
c Select communication port; 'com1' to 'com8'. If this parameter is ommitted the program will scan the comm. ports for the STK500
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.
Wt Verify that VTARGET is within 5% of the given value. 'value' is a floating point value between 0.0 and 6.0.
b Get revisions; hardware revision (h) and software revision (s).
! Set oscillator frequency; 'freq' can be whole Hz, or kHz or MHz
t Get oscillator frequency.
I Set ISP frequency; 'freq' can be whole Hz, or kHz or MHz
J Get ISP frequency.
g Silent operation.
z Not used. Supported for backwards compability.
h Help information (overrides all other settings)

Supported devices:
To display supported devices type "STK500.EXE -?"

Example 1:
stk500 -cUSB -dATMEGA8515 -e -pf -ifm8515test.hex

Description:

stk500	Runs the file stk500.exe
-CUSB	Select USB as communication port
-dAT90S8515	Set device to AT90S8515
-e	Erase flash
-if8515test.hex	Selects file "8515test.hex" to be programmed into flash
-pf	Program flash (with the selected file above)

Example 2:
stk500 -cUSB:00001A52 -ut4.5 -ua4.5 -!3686000 -l460800 -dATMEGA8515 -lFF -fD9E4 -ms -ifC:\input.hex -pf -vf –e

Description:

stk500	Runs the file stk500.exe
-cUSB:00001A52	Select USB port with the specific AVRISP mkII series number*
-ut4.5	Set the target voltage (4.5V)
-ua4.5	Set the Analog reference voltage (4.5V)
-!3686000	Set STK500 clock frequency to 3.68MHz
-l460800	Set ISP frequency to 460.8kHz
-dATMEGA8515	Set device to ATmega8515. Type "stk500.exe -h" to see all supported devices
-e	Erase flash
-lFF	Set lockbits to 0xFF
-fD9E4	Set fuses to 0xD9E4, Low byte 0xD9, High Byte 0xE4
-ms	Select serial programming mode
-ifC:\input.hex	Selects file "input.hex" to be programmed into flash
-pf	Program flash (with the selected file above)
-vf	Verify flash (with the selected file above)

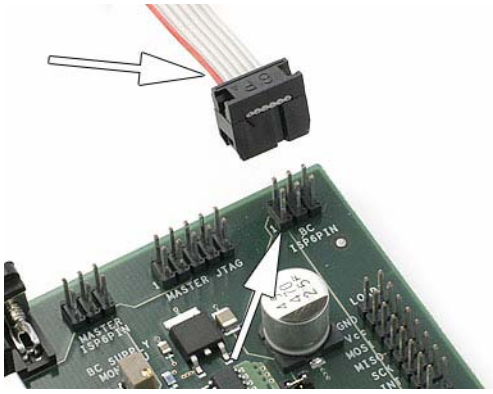
*Note that the USB port has no numbers as the COM ports. If several AVRISP mkII or other ATMEL AVR programmer tools are connected to the same PC, add the serial number to address a specific AVRISP mkII. The serial number can be found on the bottom side of the AVRISP mkII.

Hardware

Connecting AVRISP mkII

This section describes how to connect the AVRISP mkII to the host PC and the target device for correct operation. Note that AVR Studio and the USB driver must be installed. AVRISP mkII must be connected to the computer before connecting it to the target device.

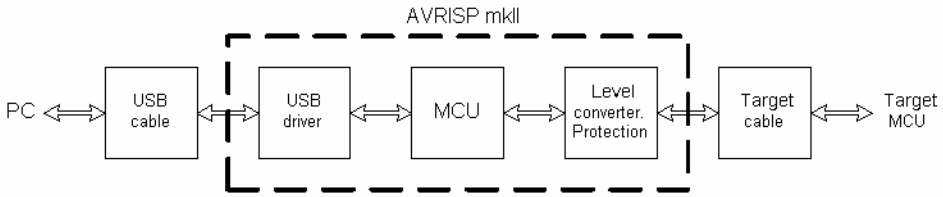
When the AVRISP mkII is connected to the PC, and if the USB driver is installed, the green LED inside the AVRISP mkII close to the USB connector will be lit. The main status LED will be red before target is detected.
After the AVRISP mkII is connected to the PC, it can be connected to the target. The red stripe on the ISP target cable marks pin 1, and this should be mated with pin 1 on the ISP connector on the target board.



When the AVRISP mkII is connected to both the PC and the target board the main status LED should be green indicating that target power has been detected. AVRISP mkII is now ready to be used with AVR Studio or the programming command line software.

Hardware Description

A block schematic of the AVRISP mkII hardware is shown below:



USB

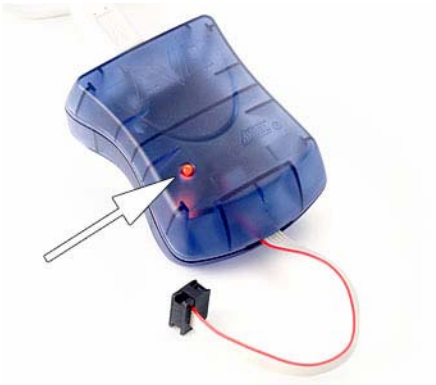
The USB interface is USB 1.1 (USB 2.0 Full Speed) 12Mbits/second.
As the AVRISP mkII is powered from the USB port it is required that the port can deliver up to 200mA. The ports on a PC, and hubs with separate power usually meets this requirement.

MCU

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

Status LED

A 3-color LED indicates the status of the AVRISP mkII. Check the Troubleshooting Guide to check for solutions if the there are any errors.

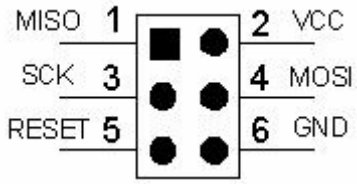


LED Color	Description
Red	Idle - No target power
Green	Idle - With target power
Orange	Busy - Programming
Orange blinking	Reversed target cable connection
Red blinking	Short-circuit on target
Red - Orange blinking	Upgrade mode

There is also a green LED inside the AVRISP mkII enclosure next to the USB connector. This LED indicates USB traffic.

Target Interface

The target connection has level converters and short-circuit protection.
Pin 1 on the connector is found on the RED side of the target cable. The target cable has the signal pinout as shown in the figure below:



Level Converter

The AVRISP mkII supports target voltages from 1.8V up to 5.5V.

Short Circuit Protection

The short circuit protection will automatically disconnect the target pins from the AVRISP mkII if the current drawn through the pin is greater than approximately 25mA.

ESD Protection

The target pins from the AVRISP mkII are protected with a transient suppressor than can withstand 8kV direct discharge.

Pull-up resistors

Pull-up resistors on MISO/MOSI/SCK connected to VCC/GND should not be stronger than 820 ohm.

Reset Line

As a part of checking if the ISP target cable is correctly mounted it will, after Vtarget is applied, check if the reset line has the correct voltage and check if it is possible to force this line low.
If there is no pull-up resistor on the line, i.e. if the AVRISP mkII detects 0V on reset, then the status LED will blink orange.
If the pull-up resistor on the reset line is to strong, the short circuit protection will trigger when the reset is forced low by the AVRISP mkII.
The Reset line should not have a stronger pull up than 4.7k ohm. Any de-coupling capacitor should not be larger than 10uF.

Note that the AT89 devices that are supported by AVRISP mkII has the opposite reset polarity. If the AVRISP mkII has been used with a AVR and then connected to a AT89, it may start to blink orange indicating error on reset. When AT89 is selected in the programming dialog and e.g. Read Signature is performed the orange blinking will occur. The same applies to the opposite situation where AT89 is used before an AVR.

Troubleshooting

Problem		Reason	Solution
1	Can't connect to AVRISP mkII from PC.	USB hub cannot provide enough power.	The AVRISP mkII requires an active/self-powered USB hub, i.e deliver 500mA.
2	Can't connect to AVRISP mkII from PC, the green USB status LED is not lit, and point 1 is OK.	USB driver is not installed properly.	Install USB driver as described in USB Setup
3	AVRISP mkII status LED is blinking orange.	ISP cable is not mounted correctly	Check that the red stripe on the cable is mating pin 1 on the ISP header. See more in Target Interface.
4	AVRISP mkII status LED is blinking orange in spite of correct ISP cable connection.	There is a problem on the reset line.	Check that the reset has a proper pull-up. Read more about Reset line.
5	AVRISP mkII reports short-circuit on the target.	The ISP cable is not mounted correctly, or some of the target pins are shorted to GND or VCC, or they are too heavily loaded.	Check point 3, and check for short circiuts. Also check that the pullup on the target lines are not to strong. See further details in Target Interface.
6	Can't detect target	The SPI interface on the target is disabled because the SPI fuse is not programmed, and/or RSTDSBL or DWEN fuse is programmed.	If the ISP interface is disabled by fuse settings, one have to use another programming interface to reset these fuses. Check the device datasheet for further details on fuse settings and programming interface. STK500 can be used for High Voltage Parallel Programming, and JTAGICE mkII can be used for JTAG programming.
7	Detects target, but can't enter prog mode or programming fails.	The ISP frequency is high.	Lower the ISP frequency. The ISP frequency is dependent on the target clock. Read more about this in the Board section.

Manual Firmware Upgrade

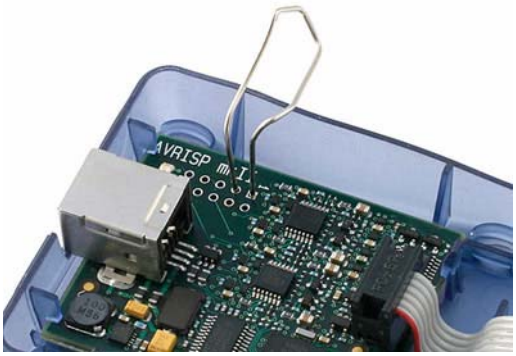
Firmware upgrading is usually handled automatically by AVR Studio, if the firmware distributed with AVR Studio is newer than the firmware loaded into the AVRISP mkII.

However, the AVRISP mkII can stop responding to firmware upgrading if the firmware on the AVRISP mkII is corrupted. Corruption of the firmware can happen if the communication between the AVRISP mkII and the PC is broken during firmware upgrading or if the firmware in the AVRISP mkII is re-programmed with the wrong file.

Follow the steps below to manually upgrade the AVRISP mkII. This procedure should work in all cases:

1. Disconnect the AVRISP mkII from target
2. Unplug the USB cable
3. Open the AVRISP mkII by pressing on the four plastic clips that holds the top and bottom parts of the enclosure together.
4. Short-circuit pin 1 and pin 3 on the pinholes next to the AVRISP mkII silk-print on the PCB. See figure below.
5. Insert the USB cable. The AVRISP mkII should now start blink red and orange.
6. Start the application "AVRISP mkII Upgrade..." located in the "Tools" menu in AVR Studio.
7. Click the "Start Upgrading" button.
8. When upgrade is finished, remove the short circuit between pin 1 and pin 3 on the AVRISP mkII.
9. Toggle AVRISP mkII power by unplug and re-insert the USB cable.
10. Press the "Done" button.

The AVRISP mkII is now upgraded and ready for use!



Known Issues

No known issues

Technical Support

For technical support please contact avr@atmel.com. When requesting technical support for AVRISP mkII please include the following information:

- Version number of AVR Studio. This can be found in AVR studio menu "Help->About"
- PC processor type and speed
- PC operating system and version
- What target AVR device is used (Complete part number)
- Fuse settings on the AVR
- Target clock frequency
- If CLKPR (Clock Prescaler Register) is used (for AVRs with this feature)
- Target voltage
- Programming speed, ISP frequency
- A detailed description of the problem, and how to recreate it.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon

Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
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

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2005. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.