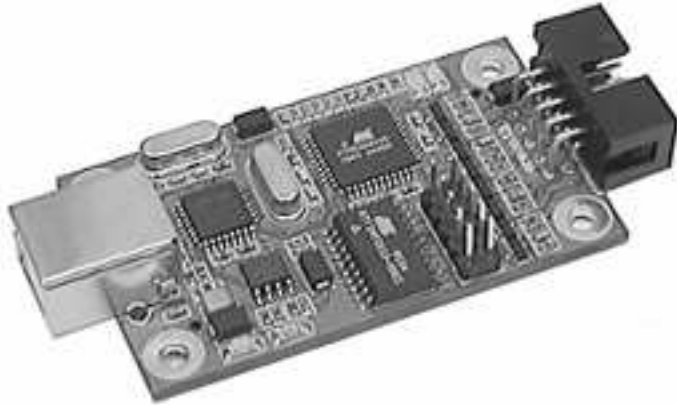


ISPAVRU1

ISP AVR Programmer through USB



Main Features

- AVR Studio Interface (AVR Studio 4.12 or later)
- Supports all AVR Device with ISP interface, refer to AVR Studio
- Programs both Flash and EEPROM
- Supports Fuse and Lock Bits Programming
- Supports RC Oscillator Calibration
- Upgradeable to Support Future Devices
- Operates at Target Voltages from 1.8V to 5V
- Adjustable programming speed
- Rescue clock – where the fuse bits do not match the hardware
- USB Interface to PC programming and Control
- Powered from PC through USB – does not require external power supply

1. Introduction

The ISPAVRU1 from ERE (www.ere.co.th) is a professional cost effective In-System Programmer covering all AVR 8-bit RISC Microcontrollers. It is designed compatible to AVRISP which is the programmer from ATMEL. But many additional functions are added for the difference from AVRISP such as a USB communication and a voltage converter for 1.8V to 5V programming. The programmer connects to a PC through USB port and draws the necessary power from PC eliminating the need for an additional power supply.

AVR Studio from ATMEL is used as front-end software, giving you highly-integrated development solution.

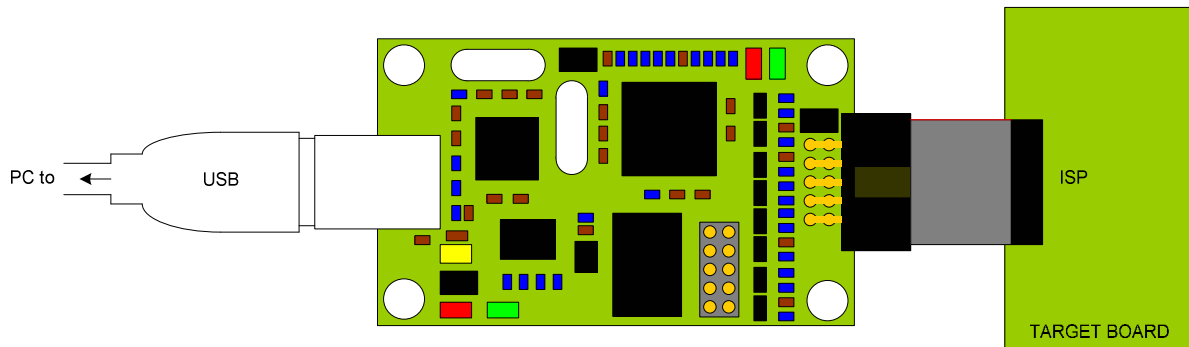


Figure 1-1 ISPAVRU1 application

The ISPAVRU1 is fully supported by AVR Studio. We recommend using the latest available version of AVR Studio (Version 4.12 when this document issued)

1.1. Device Support

ISPAVRU1 supports all AVR 8-bit RISC Microcontrollers with ISP programming option. Support for new devices will be added through new versions of AVR Studio. The latest version of AVR Studio is always available from www.atmel.com.



AVR Studio will automatically prompt the user if it detects that the firmware in the programmer is outdated.

The ISPAVRU1 supports devices operate at target voltage from 1.8V to 5V.

The ISPAVRU1 supports all different voltages and speed grades versions of the devices listed in the table below.

Classic	Tiny	Mega	Other
AT90xx1200	ATtiny12	ATmega8	AT86RF401
AT90xx2313	ATtiny13	ATmega16	AT89S51
AT90xx2323	ATtiny15	ATmega32	AT89S52
AT90xx2343	ATtiny22	ATmega48	AT90CAN128
AT90xx2333	ATtiny24	ATmega64	AT90PWM2
AT90xx4414	ATtiny26	ATmega88	AT90PWM3
AT90xx4433	ATtiny2313	ATmega103	
AT90xx4434		ATmega128	
AT90xx8515		ATmega1280	
AT90xx8535		ATmega1281	
		ATmega161	
		ATmega162	
		ATmega163	
		ATmega165	
		ATmega169	
		ATmega323	
		ATmega325	
		ATmega329	
		ATmega644	
		ATmega645	
		ATmega649	
		ATmega2560	
		ATmega2561	
		ATmega3250	
		ATmega3290	
		ATmega6450	
		ATmega6490	
		ATmega8515	
		ATmega8535	

Table 1-1 device supported

The above devices are referred from AVRISP firmware in AVR Studio version 4.12. Information about supported devices in latest versions of the firmware can be found in Online Help System in AVR Studio.

2. Getting Started

2.1. Unpacking the System

- ISPAVRU1 Programmer
- 10-pin ISP cable

Recommended products

- **I106** (10-pin to 6-pin converter)
- **LC** (lab cable, used for rescue clock wire)
- **CUSBAB** (USB cable)

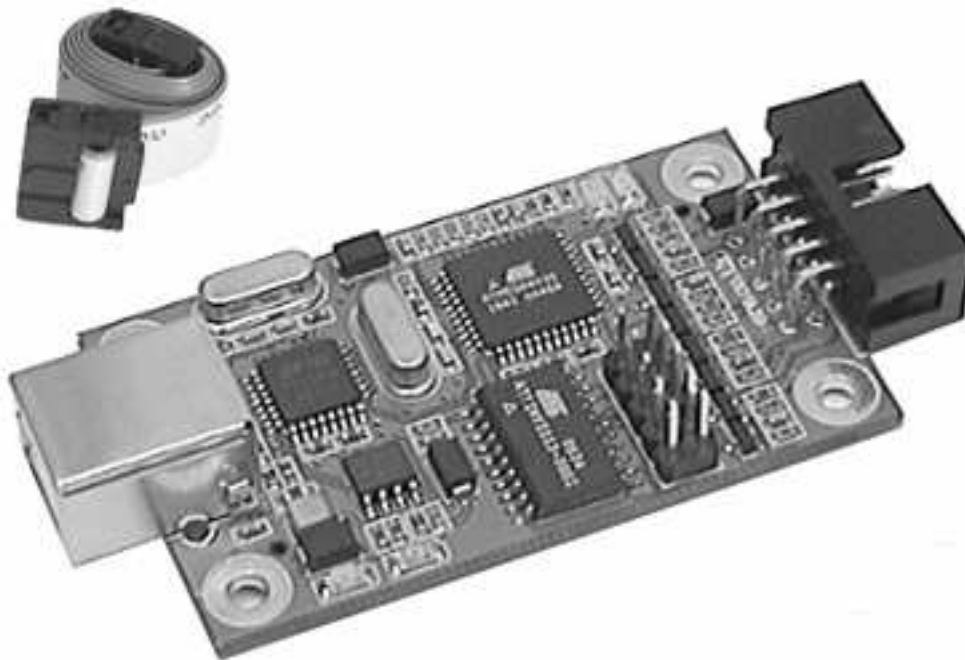


Figure 2-1 ISPAVRU1 unpacking

2.2. System Requirements

The minimum software and hardware requirements are:

- Pentium processor (Pentium II and above is recommended)
- 64 MB of RAM
- 100 MB of free hard disk space
- Windows 98/2K/XP
- USB port, bus-powered

2.3. The beginner

- Plug ISPAVRU1 to PC through USB port
- Install USB driver after new hardware is found by operating system
- Connect the programmer to target board then run the AVR Studio
- Reboot the PC or re-plug the USB if need, the AVR Studio can not find the programmer

3. Hardware Description

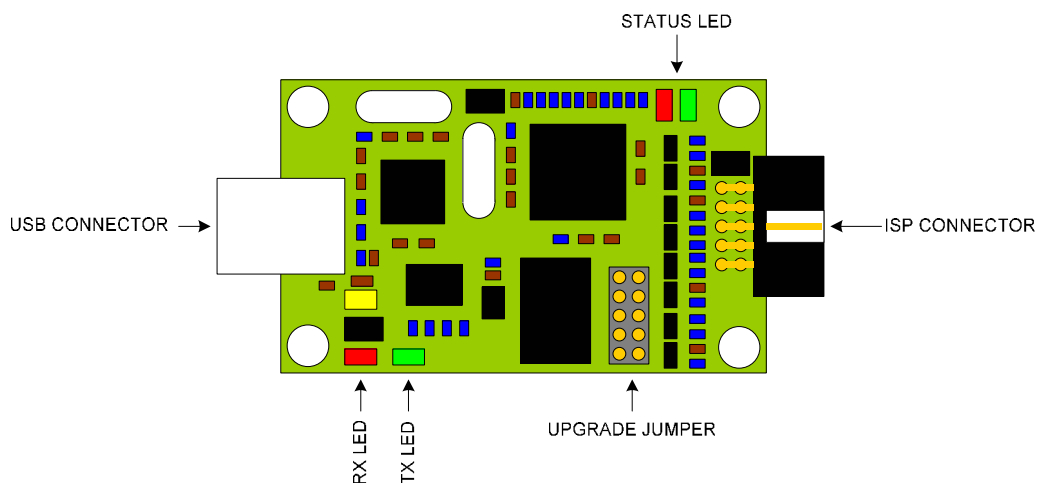


Figure 3-1 ISPAVR1 diagram

3.1. General Board Description

A block diagram of ISPAVRU1 is shown in Figure 3-2. The ISPAVRU1 can be divided into 4 sections: USB interface, Control Section, Voltage converter and ISP Cable. In this section a brief overview of different blocks will be given.

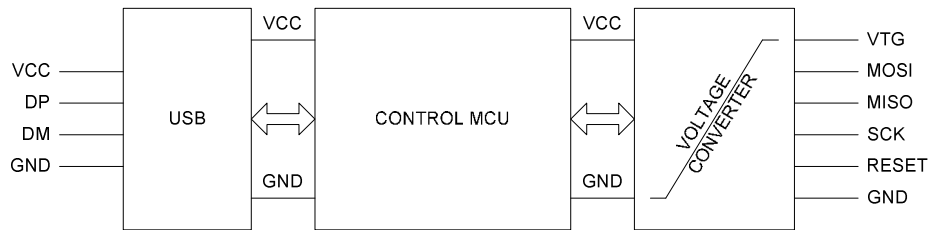


Figure 3-2 simplified ISPAVRU1 block schematics

3.1.1.USB Interface

The ISPAVRU1 uses a standard USB connector, B type. USB driver does not directly control through USB port. But it is a Com Visual Driver for RS232 communication of AVR Studio at 115200 baud.

3.1.2.Control Section

The Control MCU handles all communication between the target AVR and front-end software. The ISPAVRU1 is completely software controlled from AVR Studio. No manual configuration of ISP is needed.

LED Color	Description
Red-(Red+Green)-Green-Off-Green cycle	Power on Sequence
Red+Green	Busy-Programming
Red	Programming Failed
Green	Ready-OK

Table 3-1 LED status

3.1.3.Voltage Converter

The ISPAVRU1 operates at 5VDC under USB power supply. Also it can program the target AVR voltage from 1.8V to 5V. So the voltage converter converts the target voltage to 5V and 5V to target voltage.

3.1.4.ISP Interface cable

ISPAVRU1 supports 10-pin ISP header connector pin out.

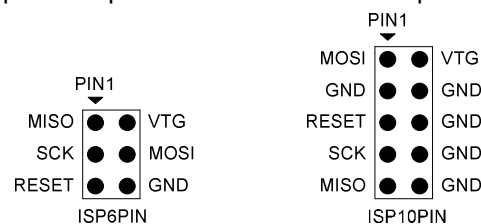


Figure 3-3 ISP connectors (top view)

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	Command and data from ISPAVRU1 to target AVR
MISO	1	9	Input	Data from target AVR to ISPAVRU1
SCK	3	7	Output	Serial Clock, Controlled by ISPAVRU1
RESET	5	5	Output	Reset. Controlled by ISPAVRU1

Table 3-2 ISP connector pin out

However the I106 can convert the 10-pin ISP header connector pin out to 6-pin ISP header connector pin out. So the I106 is advice where the target board is 6-pin header connector. The I106 is an option.

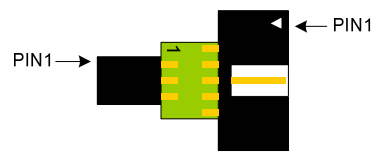


Table 3-4 I106, 10-pin to 6-pin adapter

3.2. ISPAVRU1 Power Requirements and Considerations

Event if the ISPAVRU1 draws power from USB port, however it is important that the target board is able to provide little power to voltage converter circuit.

3.3. Connecting ISPAVRU1 to Target Board

The ISPAVRU1 connects to the target board through 3 x 2 or 5 x 2 IDC socket with 2.54 mm (0.1") spacing. If there is a 6-pin or 10-pin ISP connector on the target board, the 10-pin cable or I106 has to be changed accordingly.

3.4. Handling the ISP Lines

When connecting the ISPAVRU1 to the external target some precaution should be taken. In this section a few hints and tips will be given that should assure problem free communication between the ISPAVRU1 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.

3.4.1. VCC and GND

Connect the ISPAVRU1 power lines to 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 programmer (1.8V to 5V). The target voltage is used for target voltage reference in voltage converter section, does not required to power supply for ISPAVRU1 because the ISPAVRU1 draws the necessary power from USB port.

3.4.2. SCK

The target AVR samples the clock signal generated from ISPAVRU1. To make the sampling robust, a target crystal clock, four times higher than the programming clock (SCK), is required. By selecting the correct programming speed on AVR Studio.

3.4.3. MISO/MOSI/SCK

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 show in Figure 3-5. The ISPAVRU1 should be directly connected to AVR pins, without any series resistors.

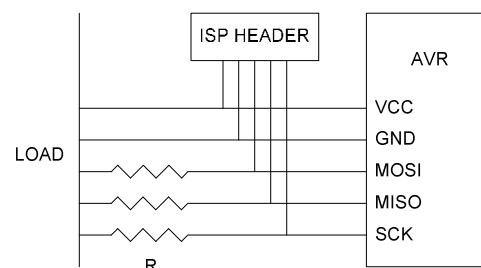


Figure 3-5 loading on the MISO/MOSI/SCK lines

3.4.4. RESET

To enter programming mode, ISPAVRU1 needed to pull reset low. It is important that the external pull-up resistor on RESET pin is not 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 10K

4. USB Driver

4.1 Installing WinXP Driver

- Plug the USB cable to PC and ISPAVRU1

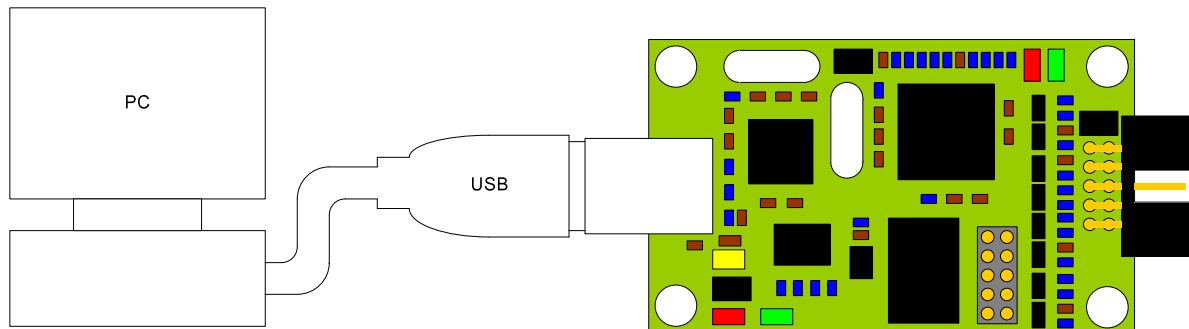


Figure 4-1 connect programmer to PC

- Wait a moment, Window will detect new USB hardware



Figure 4-2 found new hardware

- In this process the OS found new USB Visual COM driver for ISPAVRU1

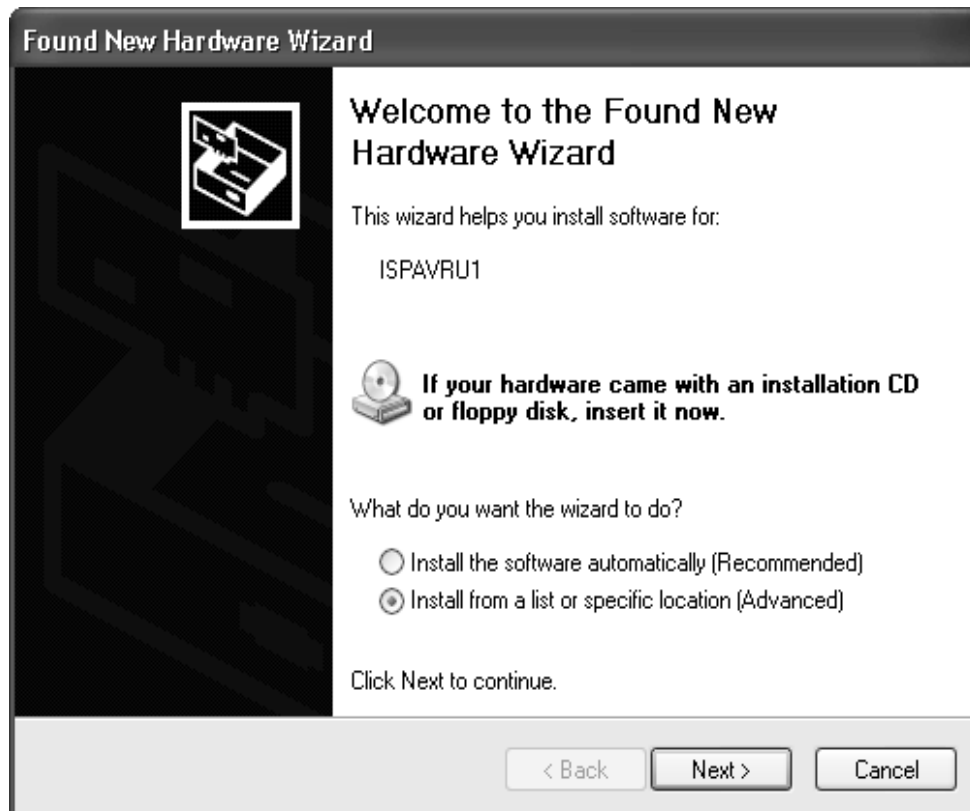


Figure 4-3 install from specific location

- Select the "Install from a list or specific location (Advanced)" radio button then click "Next" button.

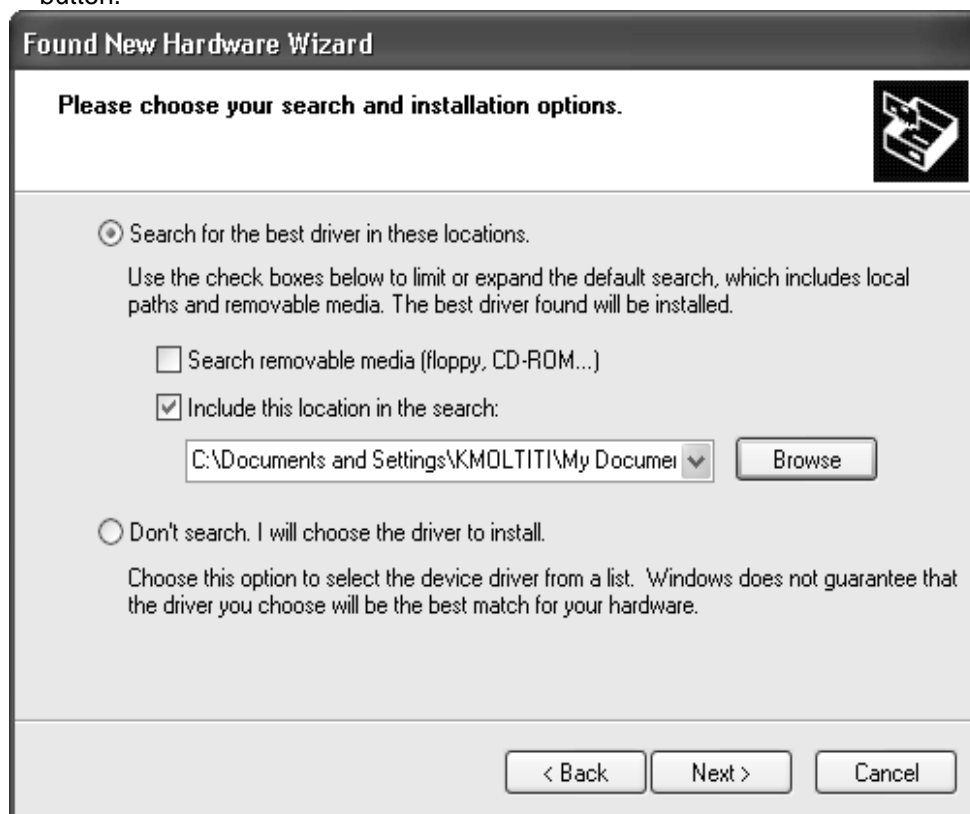


Figure 4-4 choose driver location

- Click "Brows" button to browse driver



Figure 4-5 selects the driver

- Specify the driver then click "OK" button
- If the following dialog is shown, click the "Continue Anyway" button



Figure 4-6 continue anyway

- Wait for this process

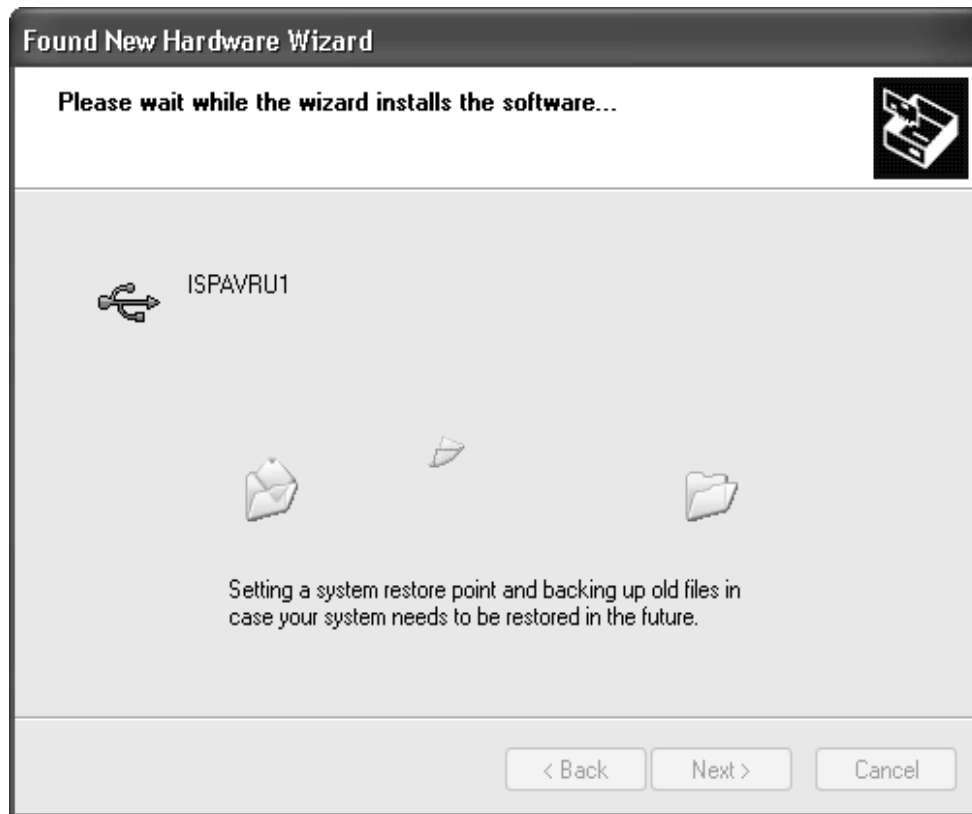


Figure 4-7 wait for setting system



Figure 4-8 completing found new hardware

- Then click "Finish" button

- The USB Visual COM Driver is installed completely
- The OS found new hardware (ISPAVRU1)



Figure 4-9 found new hardware

- Select "Install from a list or specific location(Advanced)" then click "Next" button again



Figure 4-10 Install from specific location

- Click "Next" button

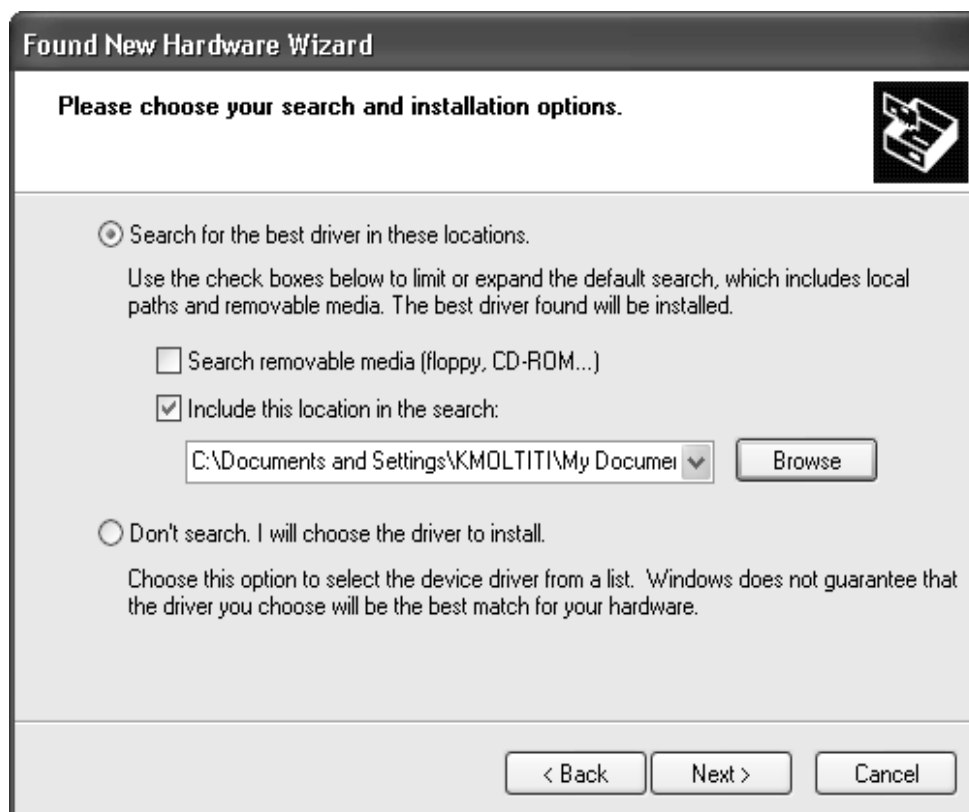


Figure 4-11 browse the driver

- Click "Browse" button to browse the driver
- In the "Browse For Folder" , Select the driver then click "OK" button



Figure 4-12 select the driver

- Click "Next" button to continue installing. And click "Continue Anyway" when a warning dialog box is shown.



Figure 4-13 driver is selected



Figure 4-14 continue anyway

- Wait for process



Figure 4-15 waiting the process

- When process is finished, click "Finish" button



Figure 4-16 finish

- The driver of the ISPAVRU1 is now installed and the programmer is ready for usage
- If need to view ISPAVRU1 in "Device Manager", please following
- The programmer is shown by select "Control Panel > Performance and Maintenance > System"
- Click "Device Manager" button on "Hardware" tab

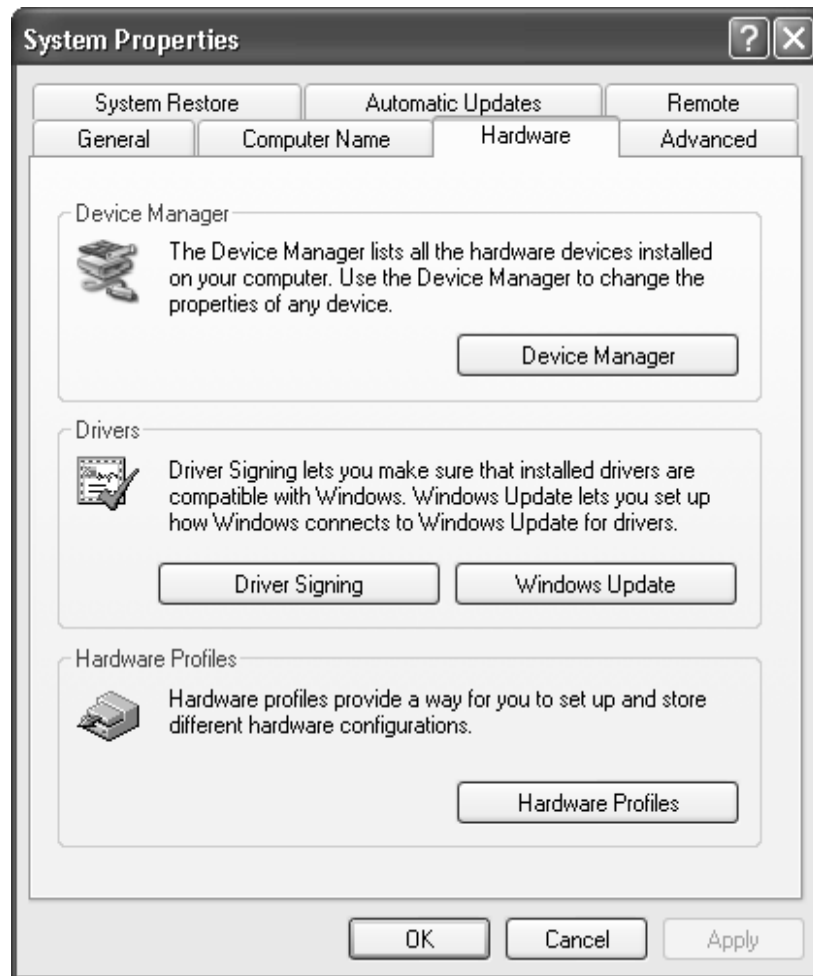


Figure 4-17 system properties

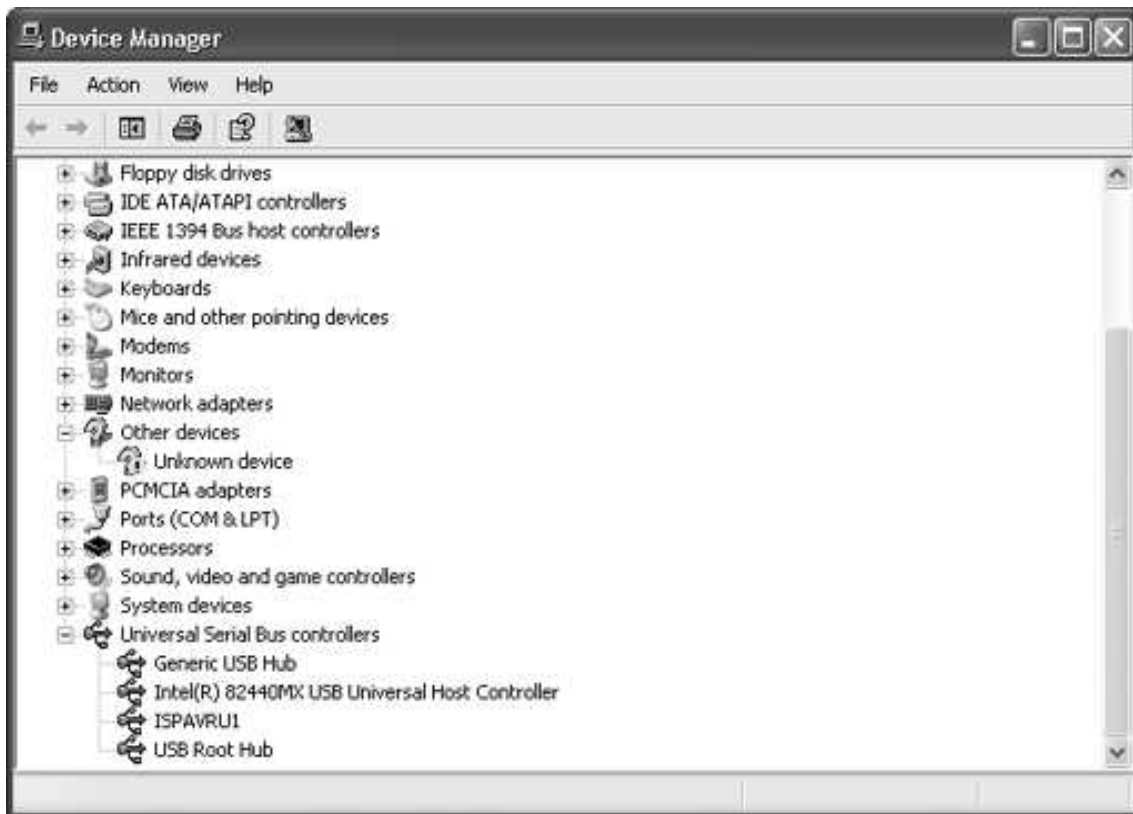


Figure 4-18 browse ISPAVRU1

5. Software Front-end

5.1. Installing AVR Studio

AVR Studio with its Integrated Development Environment (IDE) is the ideal for AVR developments. It has an editor, an assembler, a debugger and is front-end for all AVR emulators, STK500 and AVRISP In-System Programming. The ISPAVRU1 uses the same programming interface as the STK500 and the AVRISP.

To install AVR Studio download it from ATMEL website (www.atmel.com) then extracts all required files to the current directory. Execute the "Setup.exe" file. This will guide you through the setup process.

Note: AVR Studio version 4.12 or higher is required for ISPAVRU1.


AVR Studio is continuously updated to support new devices and to add functionality.

The latest AVR Studio can be downloaded from www.atmel.com


5.2. 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 the AVR Studio. The user guide covers ISPAVRU1 (STK500/AVRSP) specific topic only. In this section the supporting software for ISPAVRU1 will be presented, and an in-depth description for the available programming option given.

5.2.1. Starting the Window Software

The software used for communicating with ISPAVRU1 (STK500/AVRISP) is included in AVR Studio version 4.12 or later. Once installed, double clicking on the  icon start AVR Studio. If default install options are used, the program is located in the Windows Start Menu -> Programs -> ATMEL AVR Tools Folder.

5.2.2. Start ISPAVRU1 interface

Pressing the  button on the AVR Studio toolbar will start ISPAVRU1 (STK500/AVRISP) user interface as show in Figure 5-1

Note that the same interface is used for ISPAVRU1, STK500 and AVRISP. Since STK500 includes features that are not supported in the ISPAVRU1 (AVRISP), some features are not selectable when using the ISPAVRU1 interface. Only supported features are selectable.

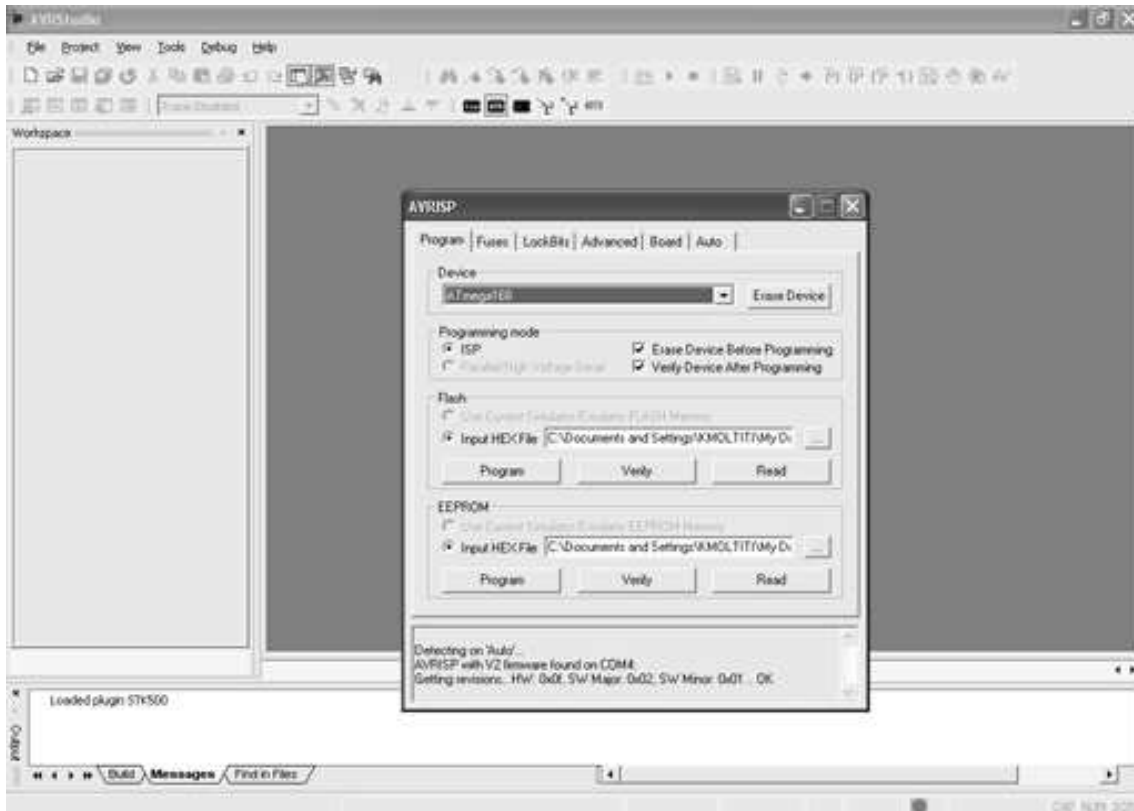


Figure 5-1 AVR Studio with ISPAVRU1 (AVRISP) user interface

5.3. ISPAVRU1 user interface

The ISPAVRU1 uses AVRISP user interface that include a lot of powerful features for the ISPAVRU1 In-System Programmer. The available settings are divided in to 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.

5.3.1. “Program” Settings

The program settings are divided into four sub groups.

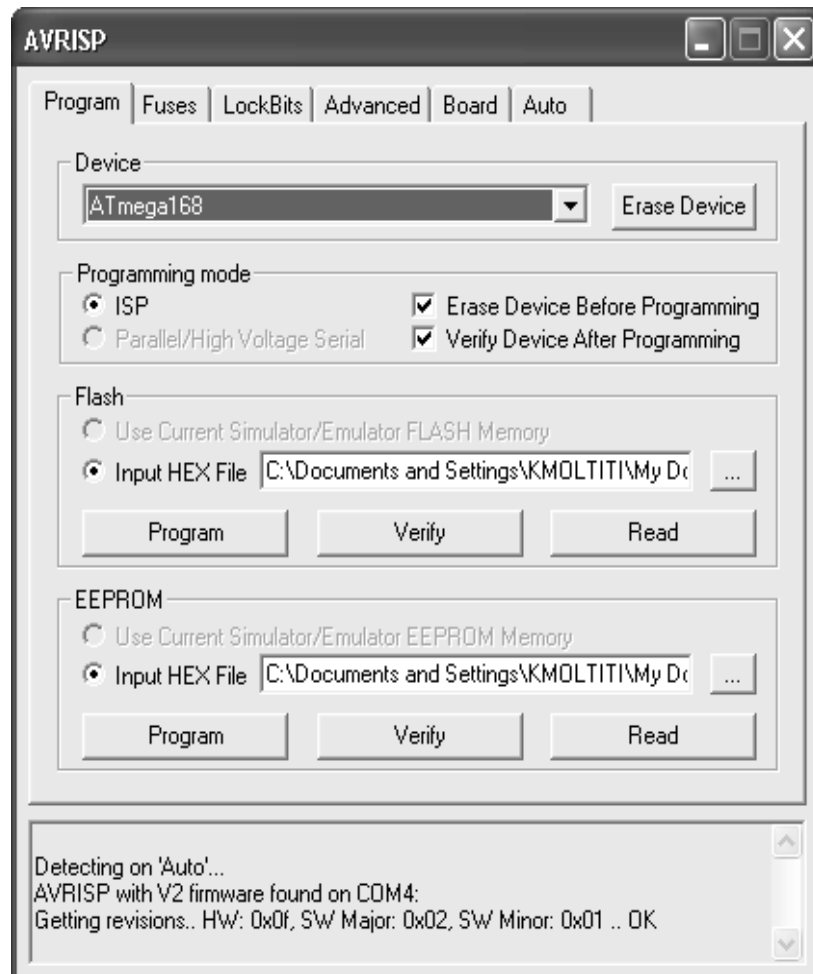


Figure 5-2 program settings

5.3.1.1. 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 FLASH and EEPROM memories.

5.3.1.2. Programming Mode

This group selects programming mode. ISPAVRU1 only supports the ISP low-voltage mode. Checking the “Erase Device Before Programming” will force ISPAVRU1 to perform a chip-erase before programming the device. Checking the “Verify Device After Programming” will force ISPAVRU1 to perform a verification of memories after programming.

5.3.1.3. Flash

If the ISPAVRU1 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.

5.3.1.4. EEPROM

If the ISPAVRU1 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.

5.3.2. "Fuses" Setting

On the "Fuses Settings" tab, overviews 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 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" setting).

Icon	Description
<input type="checkbox"/>	Un-programmed fuse or lock bit
<input checked="" type="checkbox"/>	Programmed fuse or lock bit
<input type="checkbox"/> ?	Read back of current state is impossible, but fuse or lock can be programmed. Not set to be programmed.
<input type="checkbox"/> ?	Read back of current state is impossible, but fuse or lock can be programmed. Not set to be programmed.
<input type="checkbox"/>	Read back of current value indicated un-programmed lock or fuse bit, but no access is possible. (I.e., cannot be changed in serial mode.)
<input checked="" type="checkbox"/>	Read back 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.

Table 5-1 check box description

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, verification will be automatically performed after each programming. Please see the "Special Considerations" if you plan change the RST-DISBL or SPIEN fuse.

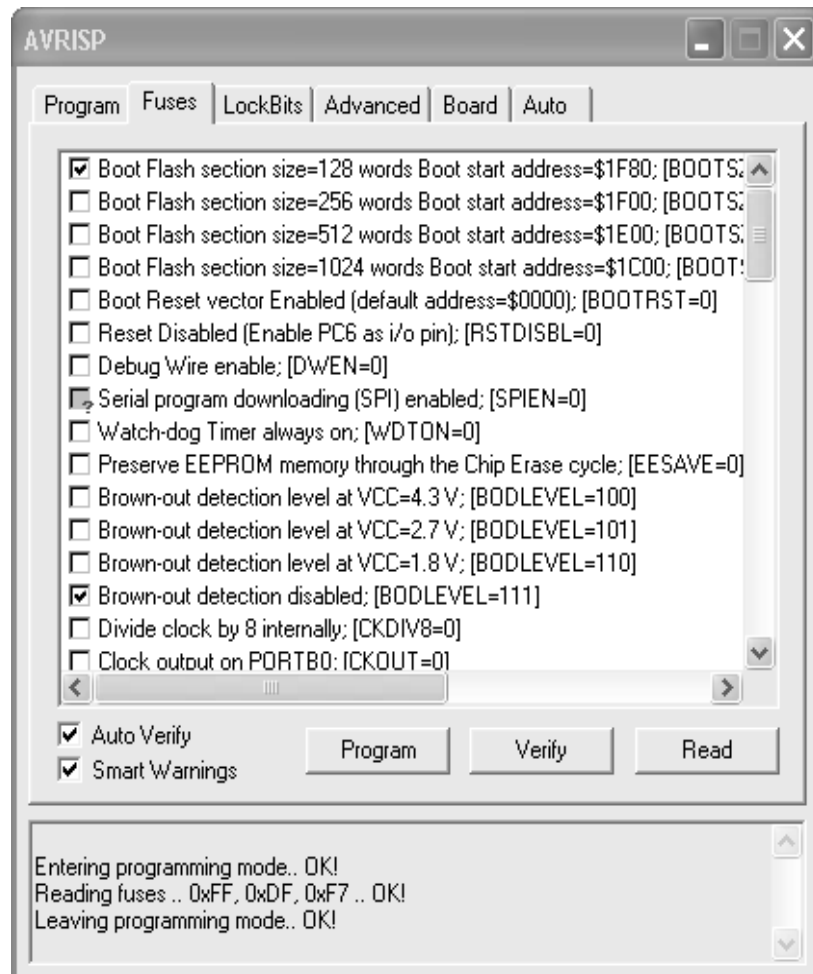


Figure 5-3 fuses setting

5.3.3. “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 ISPAVRU1 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.

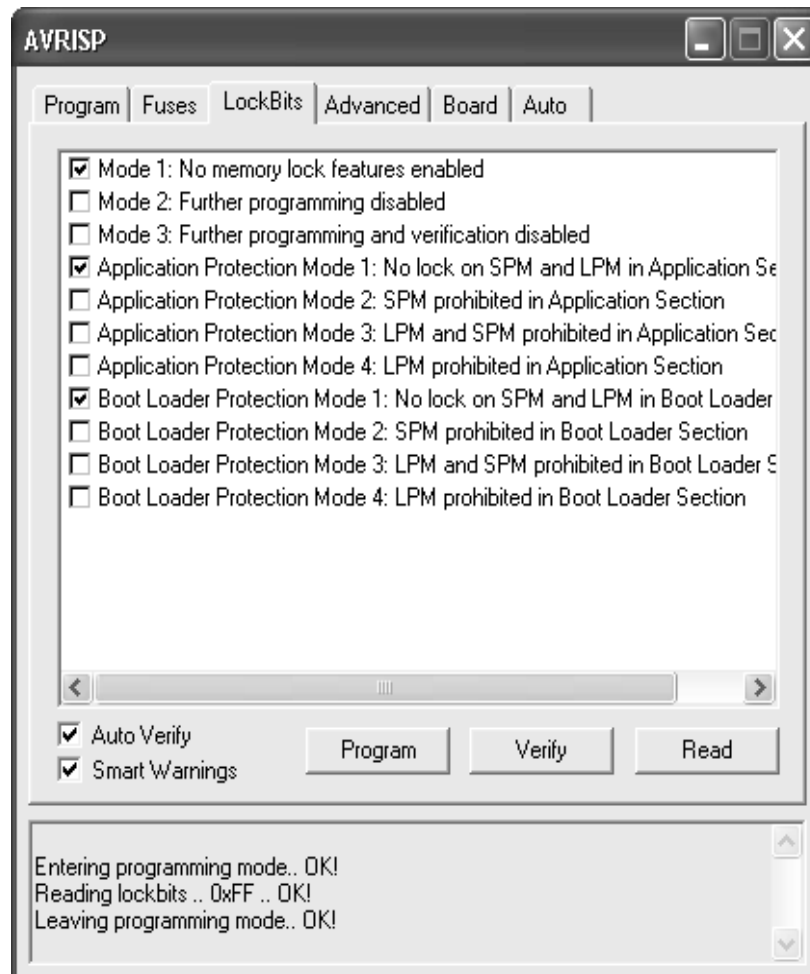


Figure 5-4 lock bits setting

5.3.4. "Advanced" Setting

The Advanced tab is currently divided into two sub groups.

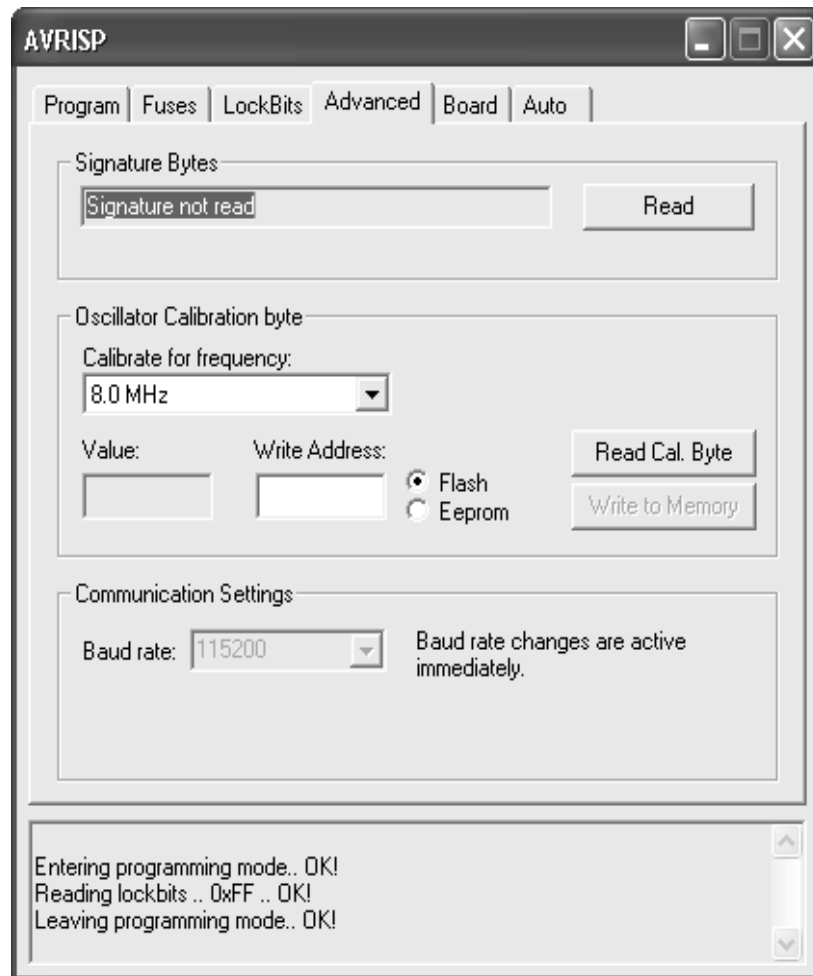


Figure 5-5 advanced setting

5.3.4.1. 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.

5.3.4.2. Oscillator Calibration Byte

For devices with calibrate-able 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.

5.3.4.3. 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.

5.3.4.4. 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.

5.3.5. "Board" Settings

The Board tab allows changing the operating conditions for the ISPAVRU1 programmer. The ISPAVRU1 allows modification of Oscillator frequency.

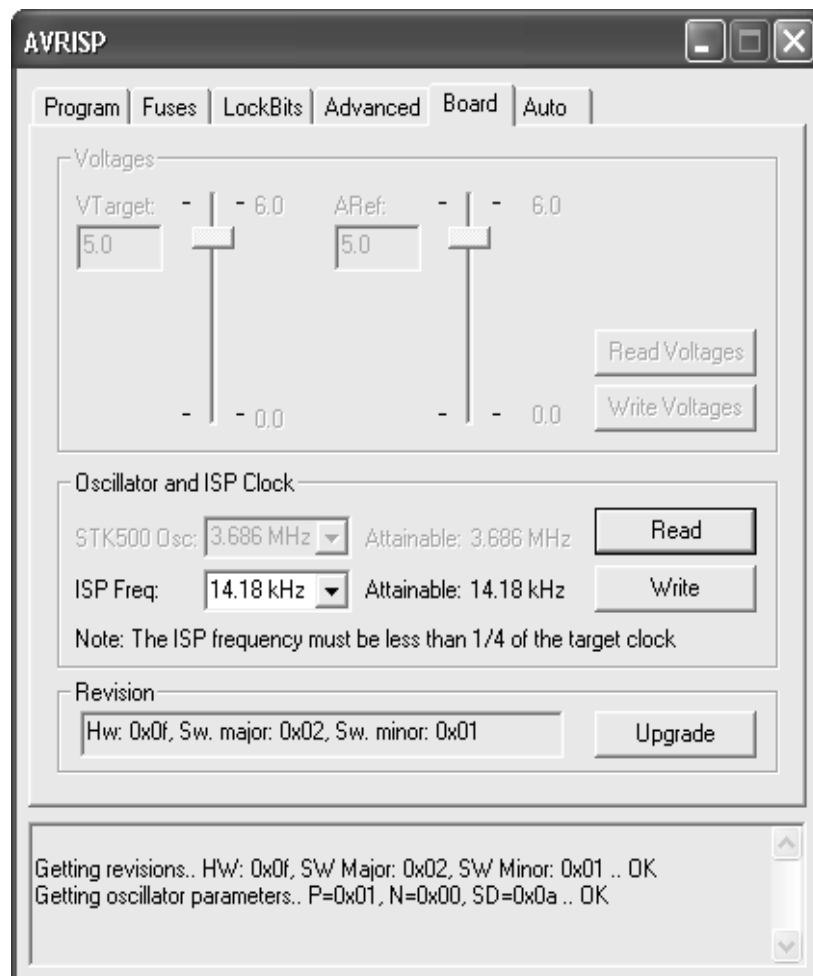


Figure 5-6 board settings

5.3.5.1. Oscillator

The ISPAVRU1 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 ISPAVRU1 User Interface calculates the communication speed between the ISPAVRU1 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.

5.3.5.2. Revision

In the revision box the current revision ISPAVRU1 revision number is shown. If AVR Studio discovers that the ISPAVRU1 contains an older version than the one distributed with AVR Studio, It will automatically ask for permission to upgrade the Programmer

5.3.6. "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 ISPAVRU1 User Interface. It is possible to log the command execution to a text file by checking the “Log to file” check box.

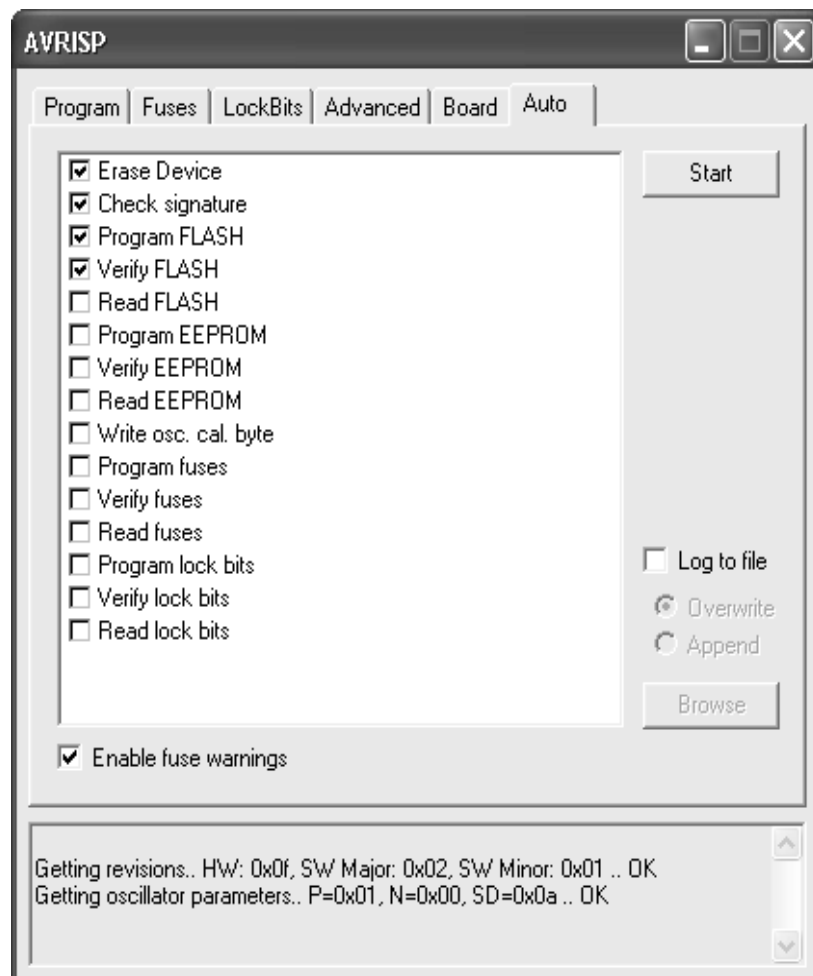


Figure 5-7 Automatic Programming Settings

5.3.6.1. Setting up the System for Auto Programming

Click on the check boxes for the commands the ISPAVRU1 User Interface should perform. A typical sequence where the device is erased and then programmed is shown in Figure. The chip is erased, both memories programmed and verified.

5.3.6.2. 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.

5.3.7. History Window

The History window is located at the bottom of the ISPAVRU1 view. In this window the dialog between AVR Studio and ISPAVRU1 is shown. For every new command performed, the old dialog is placed with the new one.

6. Rescue Clock

Where the fuse bits do not match the hardware (i.e. when the target AVR is programmed for external clock but it should be programmed for external x'tal) cause the target AVR can not run and is

not talked by the programmer. A rescue clock signal is available at the rescue clock pin on ISPAVRU1 can correct this cause by connect this pin to the XTAL1 pin of target AVR. Then uses the programmer programs correct fuse bits which match the hardware. The LC cable (our product) is advice for rescue clock wire.

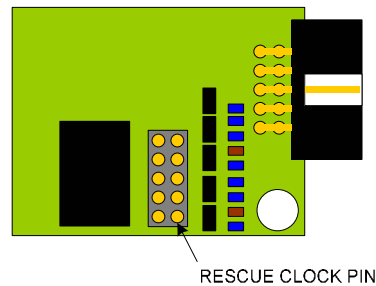


Figure 6-1 rescue clock pin

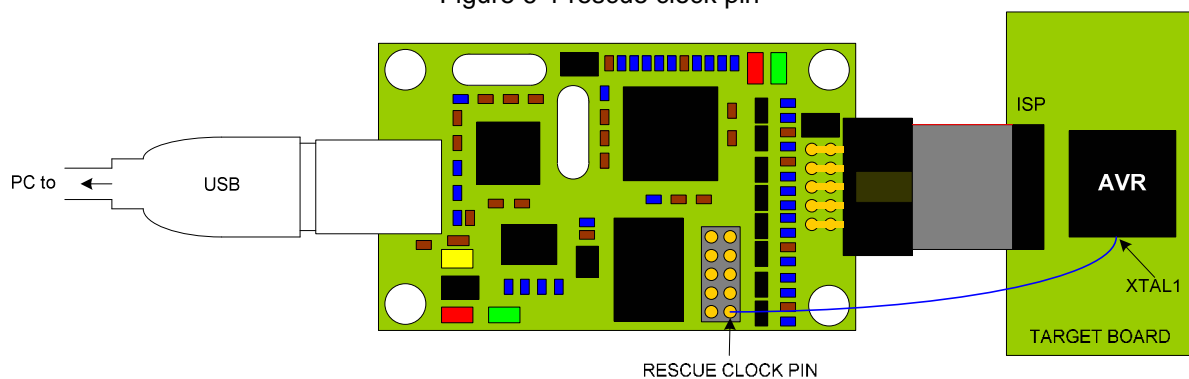


Figure 6-2 Rescue clock application

7. Special Considerations

There are a few special considerations that should be noted when using ISPAVRU1 programmer for In-System Programming of AVR devices.

7.1. Fuse Programming

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. Un-programming/programming these fuses will disable further ISP programming.

7.2. RESET Used as general I/O port

If the RESET pin is used as a general purpose I/O, 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 RSTDISBL fuse.

7.3. AVR Devices with no ISP option

Some devices do not have an ISP programming option. To program these devices, a parallel programmer is required. Only devices with low-voltage ISP programming mode are supported by the ISPAVRU1.

7.4. Devices without RC Oscillator Calibration

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.

8. ISPAVRU1 newer firmware detected

Firmware Upgrading is usually done either semi-automatically by AVR Studio, if AVR Studio detects that the Firmware distributed with AVR Studio is newer that the Firmware present in the ISPAVRU1.

- When the ISPAVRU1 is plugged to USB port on PC and the AVR Studio detects that the Firmware distributed with AVR Studio is newer than the Firmware present in the ISPAVRU1. The AVR Studio shows a newer firmware detected dialog box. Press "OK" button for progress upgrading firmware.

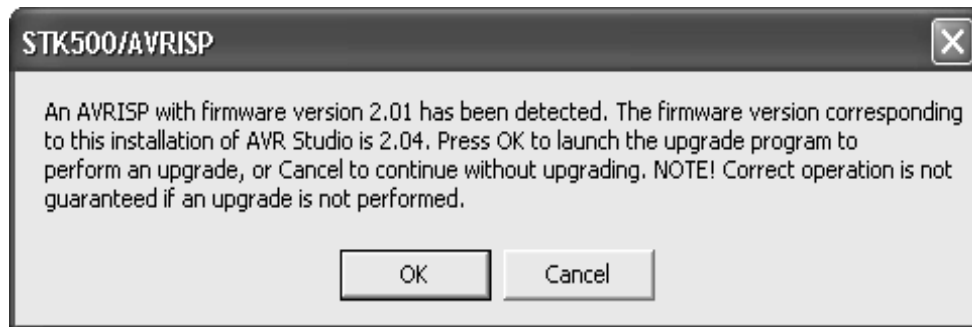


Figure 8-1 new firmware detected

- The AVR Studio shows a warning dialog box. Don't care this dialog box because the ISPAVRU1 uses power supply from USB port. It doesn't use power supply from target board. The ISPAVRU1 doesn't need connect to target board where it upgrading the firmware. Pass this process by click "OK" button.

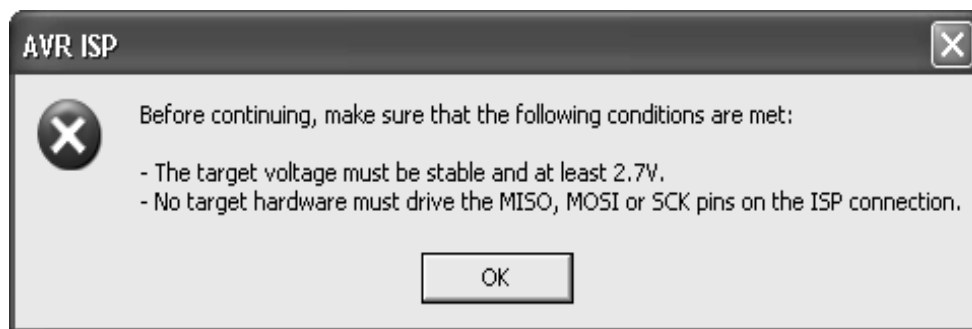


Figure 8-2 target voltage warning

- Then the AVR Studio shows "Please Wait" dialog box.



Figure 8-3 Please Wait

- After the above process is completed, The "Atmel STK500/AVRISP Upgrade" program is started than it shows "No programmer found" in status line. Don't care this process.

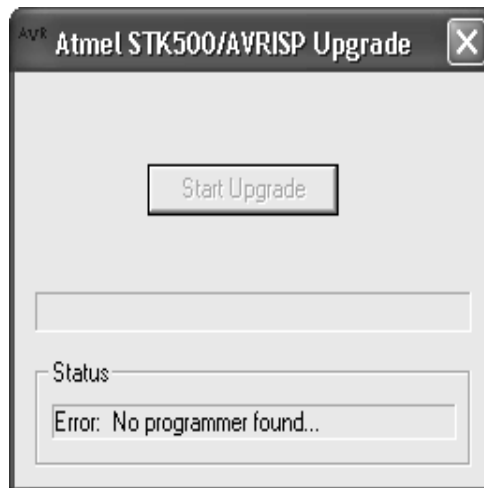


Figure 8-4 ATMEL STK500/AVRISP upgrade program

- Select “Tool > AVR Prog” in AVR Studio menu bar.

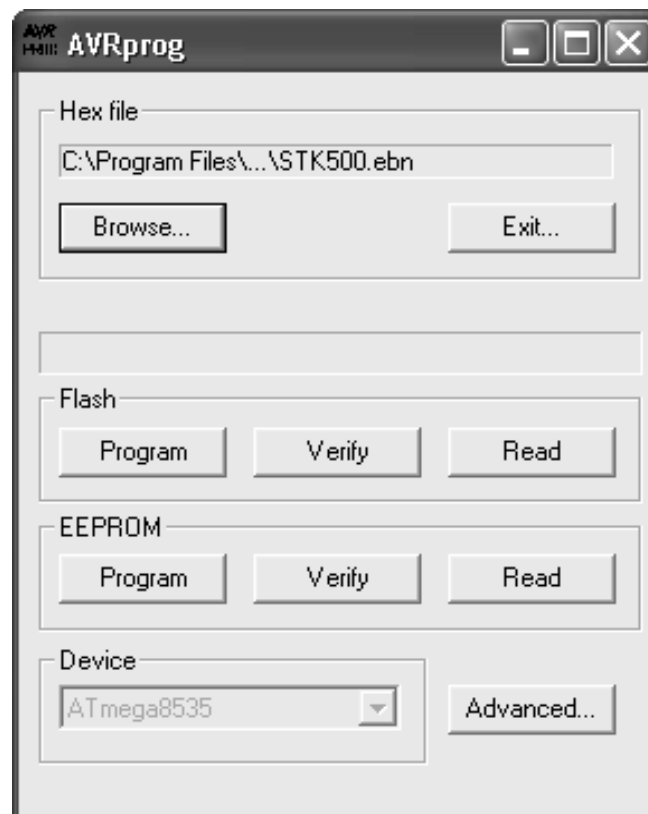


Figure 8-5 AVR Prog

- Click “Browse” button in Hex file window. Locate and select the “STK500.ebn” in the AVR Studio subfolder named STK500. Typical path is “C:\Program Files\Atmel\AVR Tools\STK500” in AVR Studio 4.x.

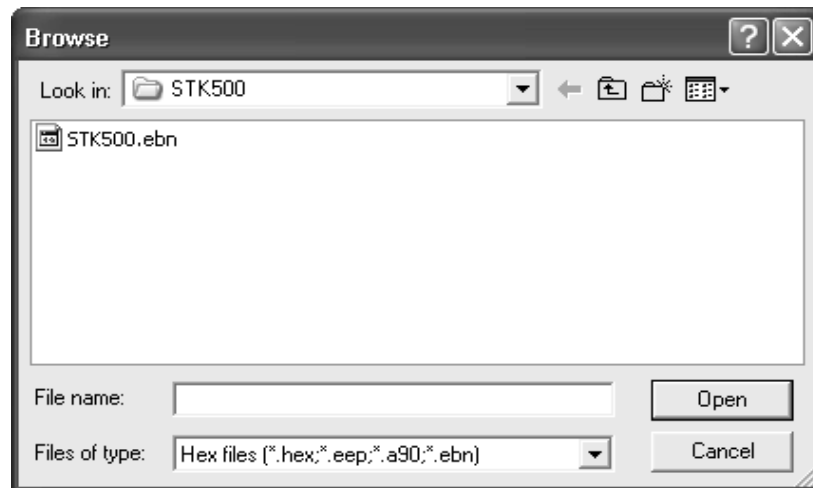


Figure 8-6 browse the STK500.ebn

- Click the "Program" button in Flash window.

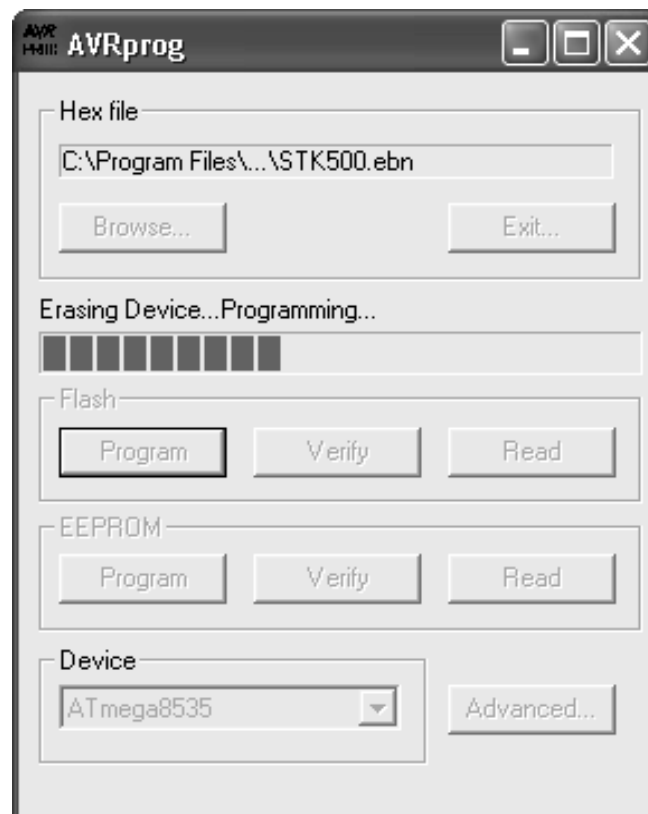


Figure 8-7 AVR Prog is programming

- Close "AVR Prog" after Programming and verification is completed.
- Close the "Atmel STK500/AVRISP Upgrade" dialog box
- Power off the ISPAVRU1 by take off the USB cable from PC.
- The Firmware of the ISPAVRU1 is now upgraded and the programmer is ready for usage.

9. ISPAVRU1 Semi-Auto Firmware Upgrade

Firmware Upgrading is usually done either semi-automatically by selecting "upgrade" in the STK500/AVRISP dialog window opened from the Tools Menu in AVR Studio.

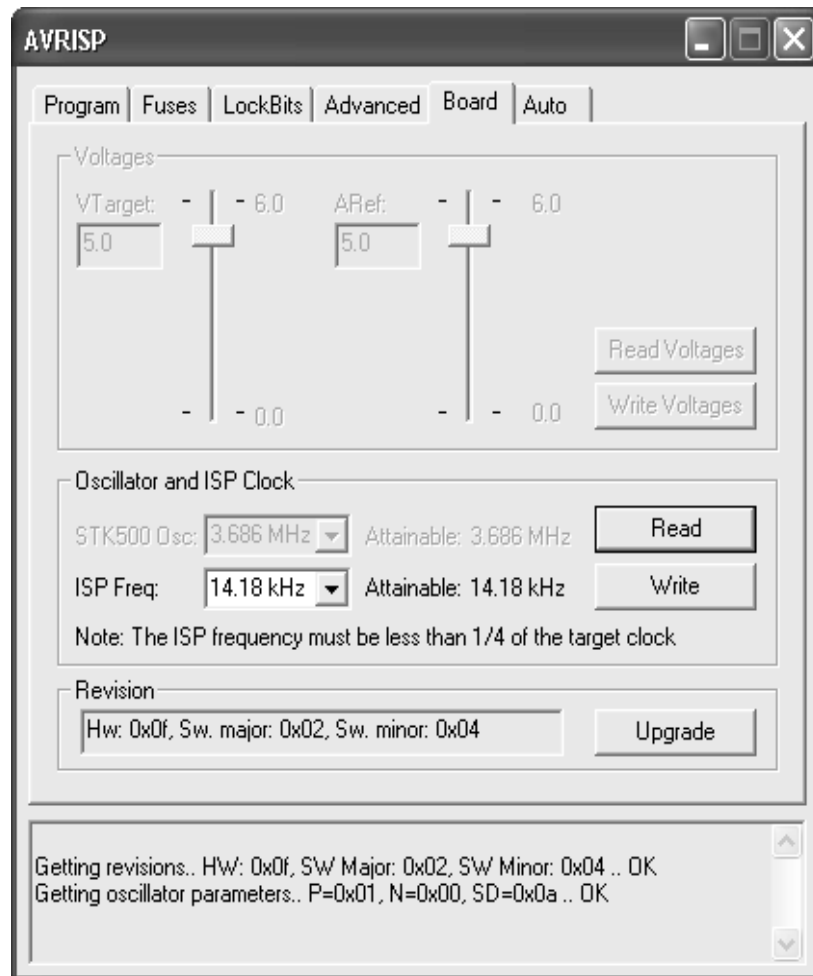


Figure 9-1 upgrade

- In “Board” tab of AVRISP program, Click “Upgrade” button for upgrading firmware.
- The AVR Studio shows a warning dialog box. Don't care this dialog box because the ISPAVRU1 uses power supply from USB port. It doesn't use power supply from target board. Pass this process by click “OK” button.

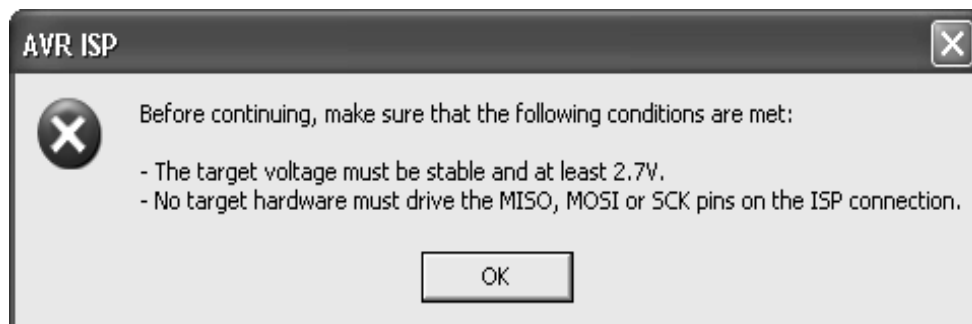


Figure 9-2 target voltage warning

- Then the AVR Studio shows “Please Wait” dialog box.



Figure 9-3 please wait

- After the above process is completed, The "Atmel STK500/AVRISP Upgrade" program is started than it shows "No programmer found" in status line. Don't care this process.

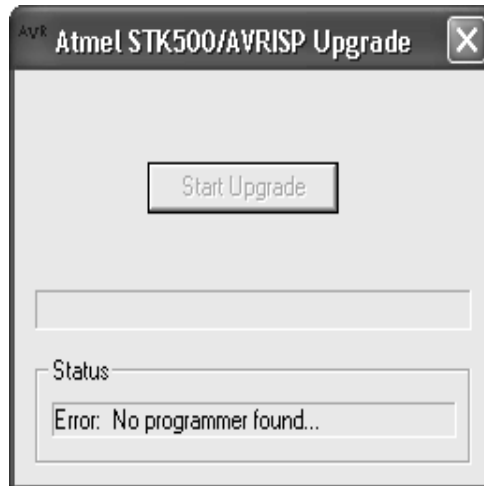


Figure 9-4 ATMEL STK500/AVRISP Upgrade program

- Select "Tool > AVR Prog" in AVR Studio menu bar.

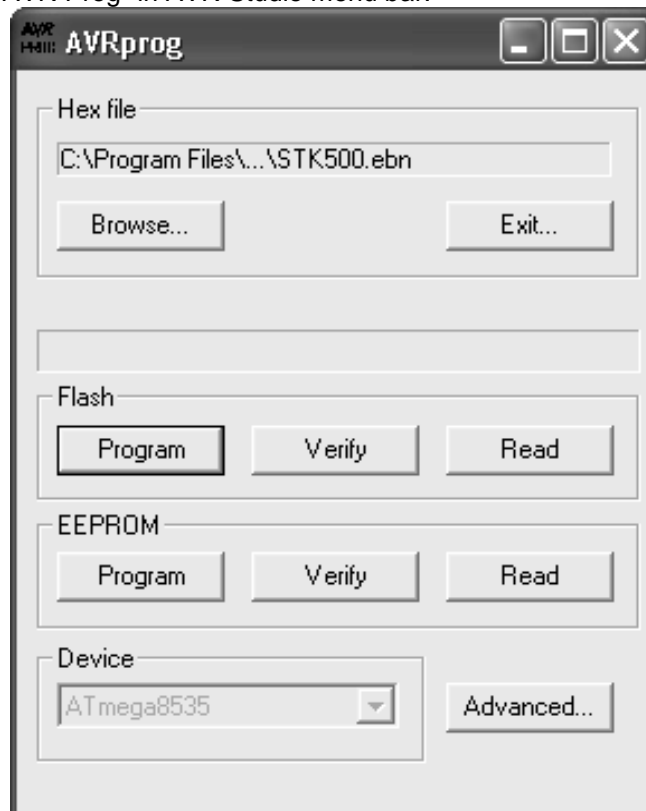


Figure 9-5 AVR Prog

- Click "Browse" button in Hex file window. Locate and select the "STK500.ebn" in the AVR Studio subfolder named STK500. typical path is "C:\Program Files\Atmel\AVR Tools\STK500" in AVR Studio 4.x.

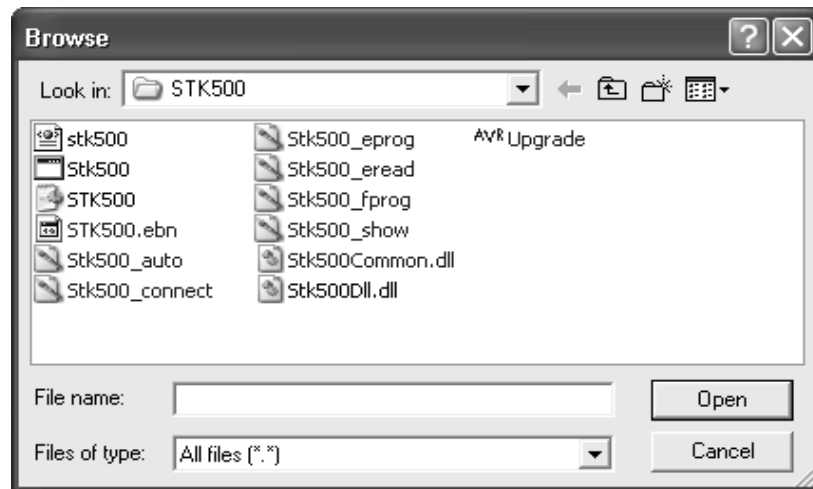


Figure 9-6 browse the STK500.ebn

- Click "Program" button in Flash window.

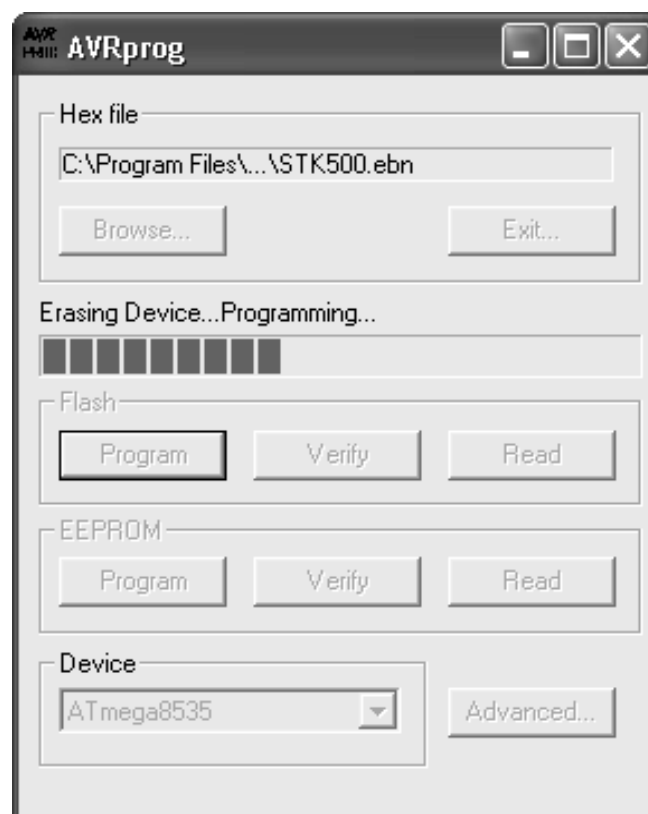


Figure 8-7 AVR Prog is programming

- Close "AVR Prog" after Programming and verification is completed.
- Close the "Atmel STK500/AVRISP Upgrade" dialog box
- Power off the ISPAVRU1 by take off the USB cable from PC.
- The Firmware of the ISPAVRU1 is now upgraded and the programmer is ready for usage.

10. ISPAVRU1 Manual Firmware Upgrade

Firmware Upgrading is usually done either semi-automatically by AVR Studio, if AVR Studio detects that the Firmware distributed with AVR Studio is newer than the Firmware present in the ISPAVRU1, or by selecting "upgrade" in the STK500/AVRISP dialog window opened from the Tools Menu in AVR Studio. However, if the communication between the ISPAVRU1 and the PC is broken during Firmware Upgrading the ISPAVRU1 stops responding, both Firmware Upgrading and Programming will not work. The ISPAVRU1 is NOT broken, but it is no longer able to automatically enter Programming mode – Programming mode have to be forced to do a firmware upgrade. Once this is done the ISPAVRU1 is up and running again.

The problem can occur if the USB cable is removed during the Programming.

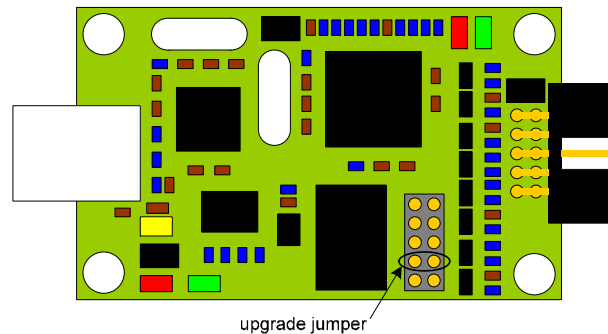


Figure 10-1 upgrade jumper mark

Procedure for manually upgrading the ISPAVRU1 Firmware:

- Power off the ISPAVRU1 by take off the USB cable from PC that supplies the power to the ISPAVRU1.

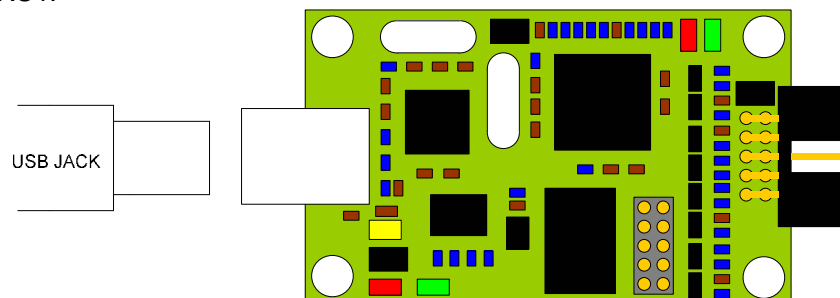


Figure 10-2 power off the ISPAVRU1

- Short the jumper marked on Figure 10-3.

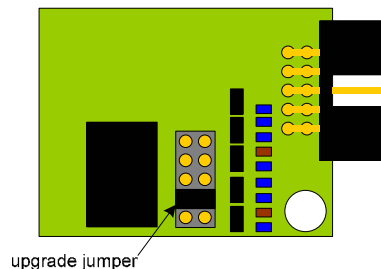


Figure 10-3 short the upgrade jumper

- Power on the ISPAVRU1 by put on the USB cable to PC that supplies the power to the ISPAVRU1.

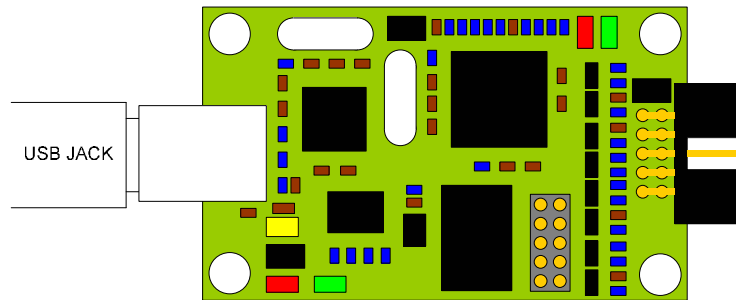


Figure 10-4 power on the ISPAVRU1

- Wait for a moment.
- Start AVR Studio.

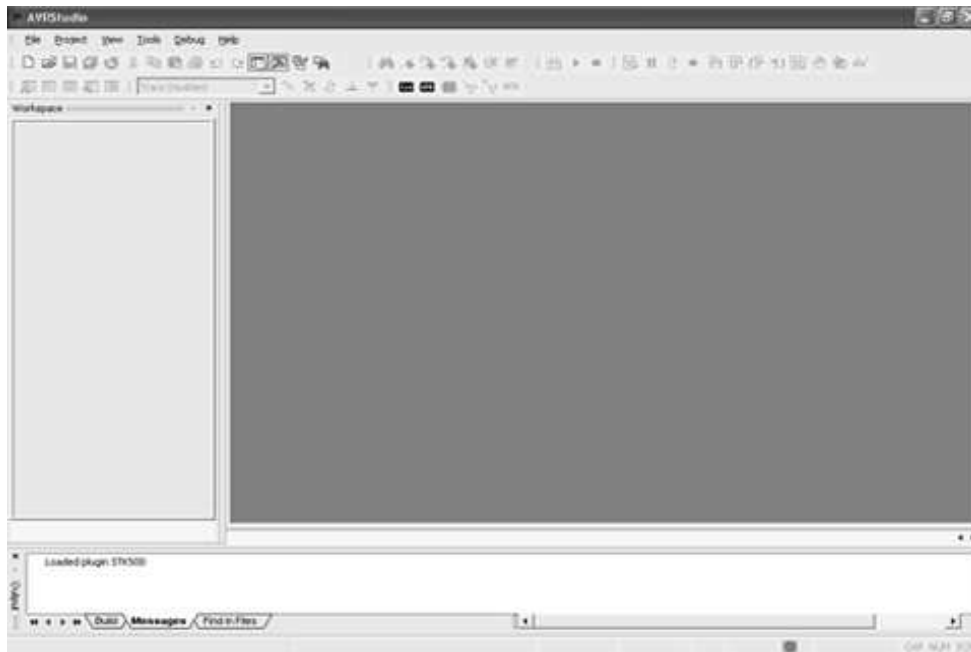


Figure 10-5 AVR Studio

- Start the application "AVR Prog" located in the Tools menu.

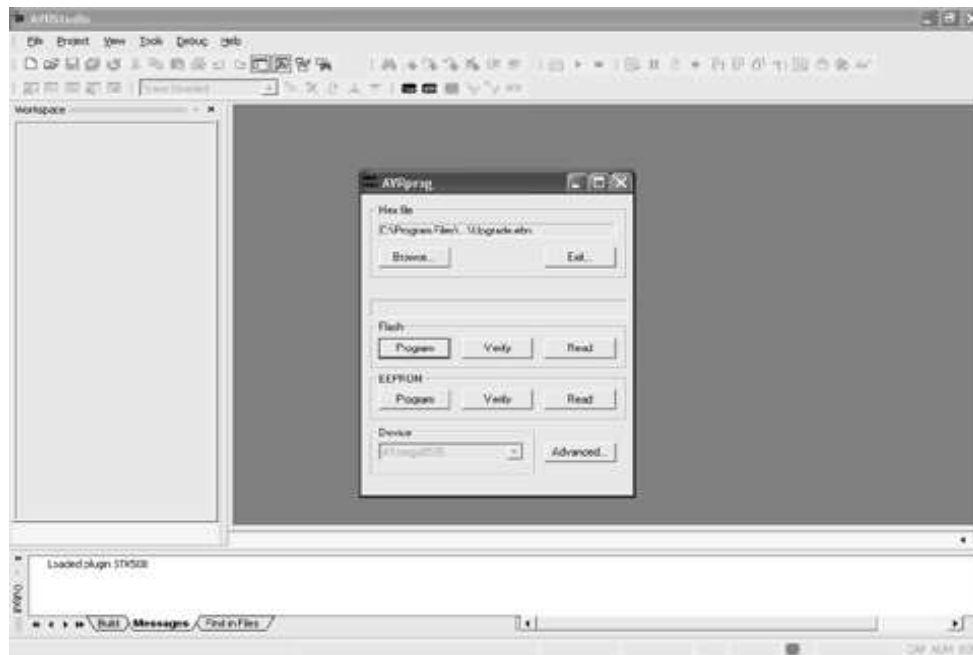


Figure 10-6 start the AVR Prog

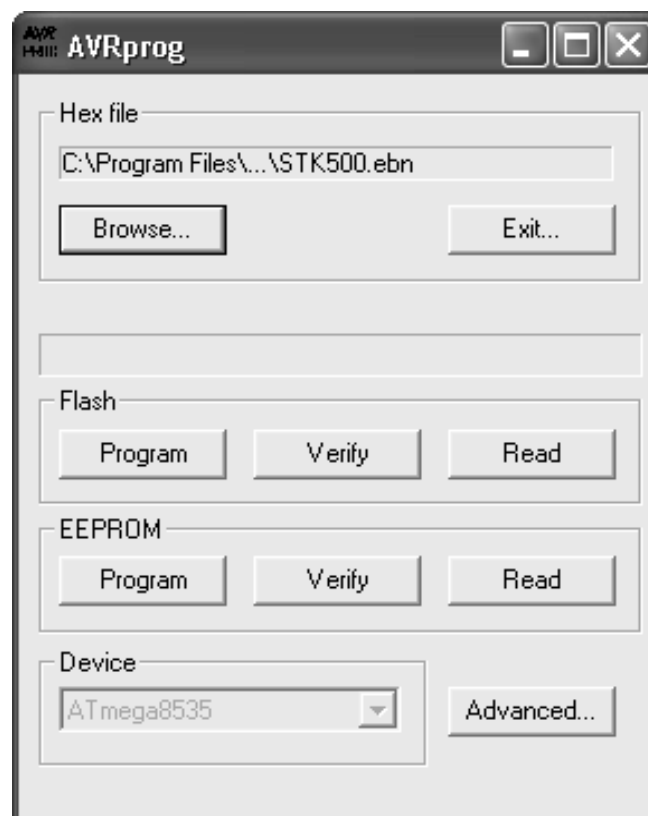


Figure 10-7 AVR Prog

- Click "Browse" button in Hex file window. Locate and select the "STK500.ebn" in the AVR Studio subfolder named STK500. Typical path is "C:\Program Files\Atmel\AVR Tools\STK500" in AVR Studio 4.x.

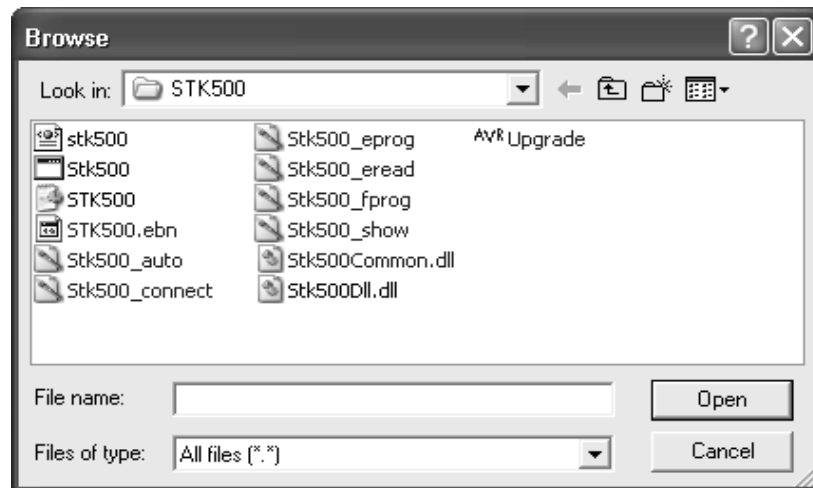


Figure 10-8 browse the STK500.ebn

- Click "Program" button in Flash window.
- Close "AVR Prog" after Programming and verification is completed.
- Power off the ISPAVRU1 by take off the USB cable from PC.
- Remove the short.

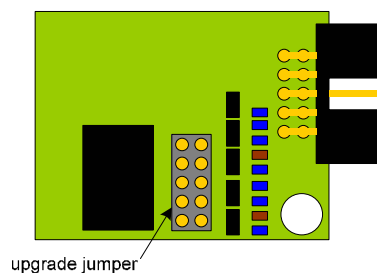


Figure 10-9 removes the short

- The Firmware of the ISPAVRU1 is now upgraded and the programmer is ready for usage.

11. Troubleshooting Guide

Problem	Reason	Solution
The LED is not lit	ISPAVRU1 is not connected to USB port	Connect USB cable to PC
	ISP pin-out is not correct	Verify pin-out on target ISP header connector
	USB does not provide enough power	Verify that the USB port can deliver enough power.
Can't get any communication with target device	ISP pin-out is not correct	Verify pin-out 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 resistor between load and IO pins
	Too strong pull-up on RESET pin	Reset pull-up resistor should be more than 10K
	ISP Freq set wrong in AVR Studio	Reduce the ISP frequency in AVR Studio to match the target board frequency
USB does not provide enough power	Verify that the USB power supply can deliver enough power to source ISPAVRU1	

	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 ISPAVRU1	Old version of AVR Studio	Download AVR Studio newer version from Atmel's website
	Other Device or service controls the USB port	Disable other drivers.

Table 11-1 Troubleshooting Guide