Overview

Atmel® encourages the use of this kit to develop a prototype platform for evaluation of the intended application. The source code is included in the CD with the kit, but for the latest revision please check the Atmel web site (www.atmel.com) for new releases. This kit was designed to make modification or upgrading the firmware a very simple and quick process. The kit accepts two very common programming interfaces, ISP and JTAG. Firmware v3.0 and later will also allow re-programming via the serial port.

Tools needed for firmware upgrade:

- **AVR Studio® 4**
- **Programmer (AVRISP MKII, JTAGICE MKII, STK®500)**

Tool needed to develop firmware:

- **AVR Studio 4**
- **WinAVR**
- **Programmer (JTAGICE MKII, AVRISP MKII, STK500)**

Fuse Settings:

Fuse bits must be set properly for the kit to function. The only fuse bits that should be set are listed below.

- **JTAG Interface Enabled; (JTAGEN=0)**
- **Serial program downloading (SPI) enabled; (SPIEN=0)**
- **Boot Flash section size=1024 words Boot start address=$FC00; (BOOTSZ=10)**
- **Boot Reset vector Enabled (default address = $0000; (BootRST=0)**
- **Brown-out detection level at VCC=2.7V; (BODLEVEL=1)**
- **Int. RC OSC. 8MHz; Start-up time; 6CK + 64ms; (CKSEL=0100 SUT=10)**

All others should be unchecked

These fuses can be programmed using the tools described below. This only needs to be preformed once. Subsequent flashing of the memory does not affect the fuse settings. Fuse settings should only be modified if the change is well understood. For example, changing the clocking from Int. RC OSC to external could render the device unusable without extensive work. Or removing the SPI enabled fuse setting could make programming impossible without using a second tool.

ISP Programming

ISP is a downloading method where only six I/O lines are needed. The provided header works with the ATSTK500 as well as the AVR ISP In-System Programmer (ATAVRISP MKII). Both of these are available from Atmel or distributors for very low cost program development.

**AVR based 125kHz RFID Evaluation Kit (Re)Programming Guide**

ATA2270-EK1

To program the flash memory and the EEPROM, first connect the programming tool to the Main Board (MB) using a six-pin ribbon connector. The ISP header is the six pins located directly above the power plug. Make sure that proper orientation is maintained by aligning pin1 to prevent power to ground shorts.
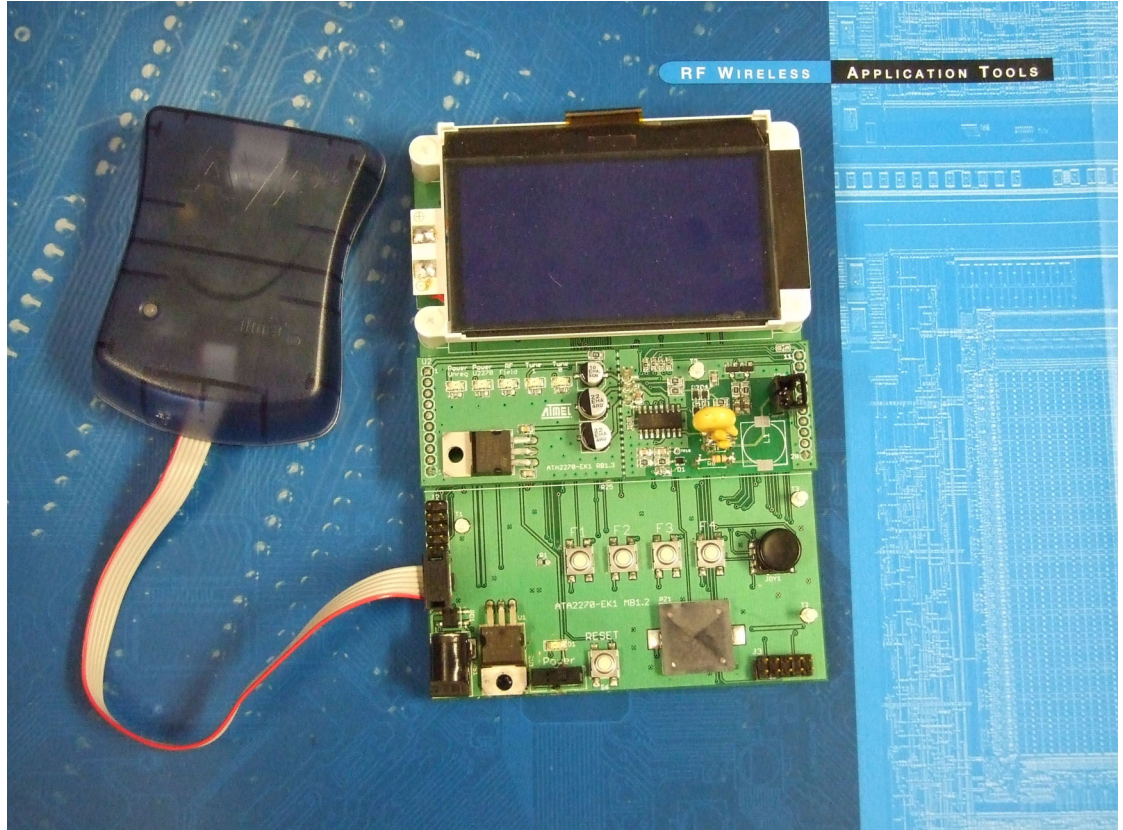
1. Open AVR Studio4 and select "Program AVR", then "Connect..." from the "Tools" menu. Cancel the welcome screen if necessary.
2. Connect to the STK500 or AVRISP MKII by selecting "STK500 or AVRISP MKII" in the Platform List. Then select "Connect...".
3. The programming dialog box should appear. Note that the system supports other programming platforms as well, like the JTAGICE mkII.
4. Next select the AVR target device (ATmega128) from the pull-down menu on the "Program" tab and locate the Intel-hex file to download. This can be the pre-compiled firmware that was downloaded from the Atmel web site (or from the CD) or code that has been customized for a specific application.
5. Select the "program" button.
6. Repeat this procedure with the file containing EEPROM initial values.
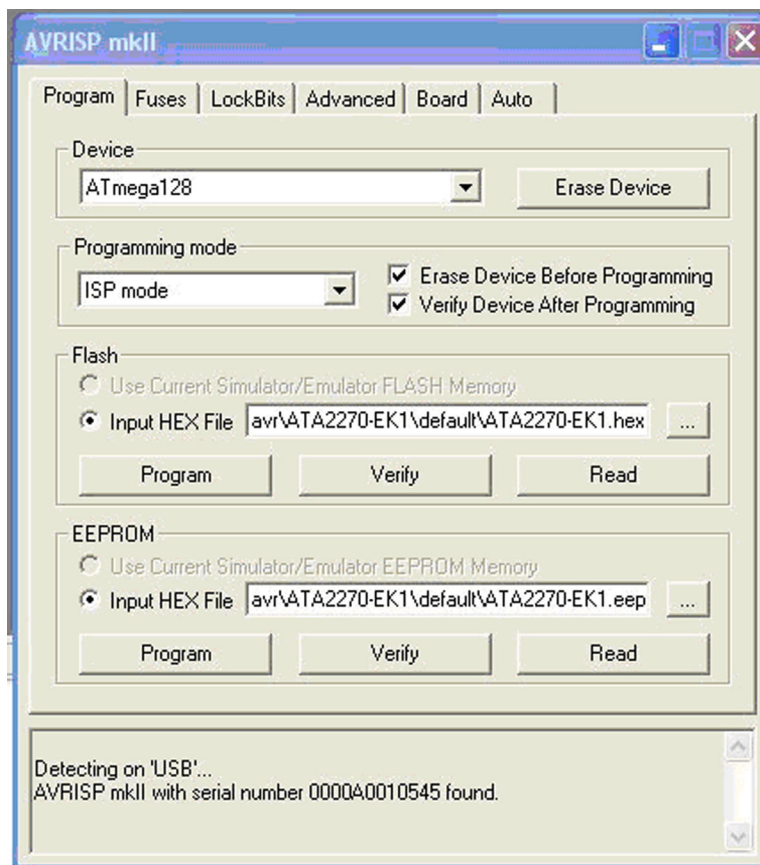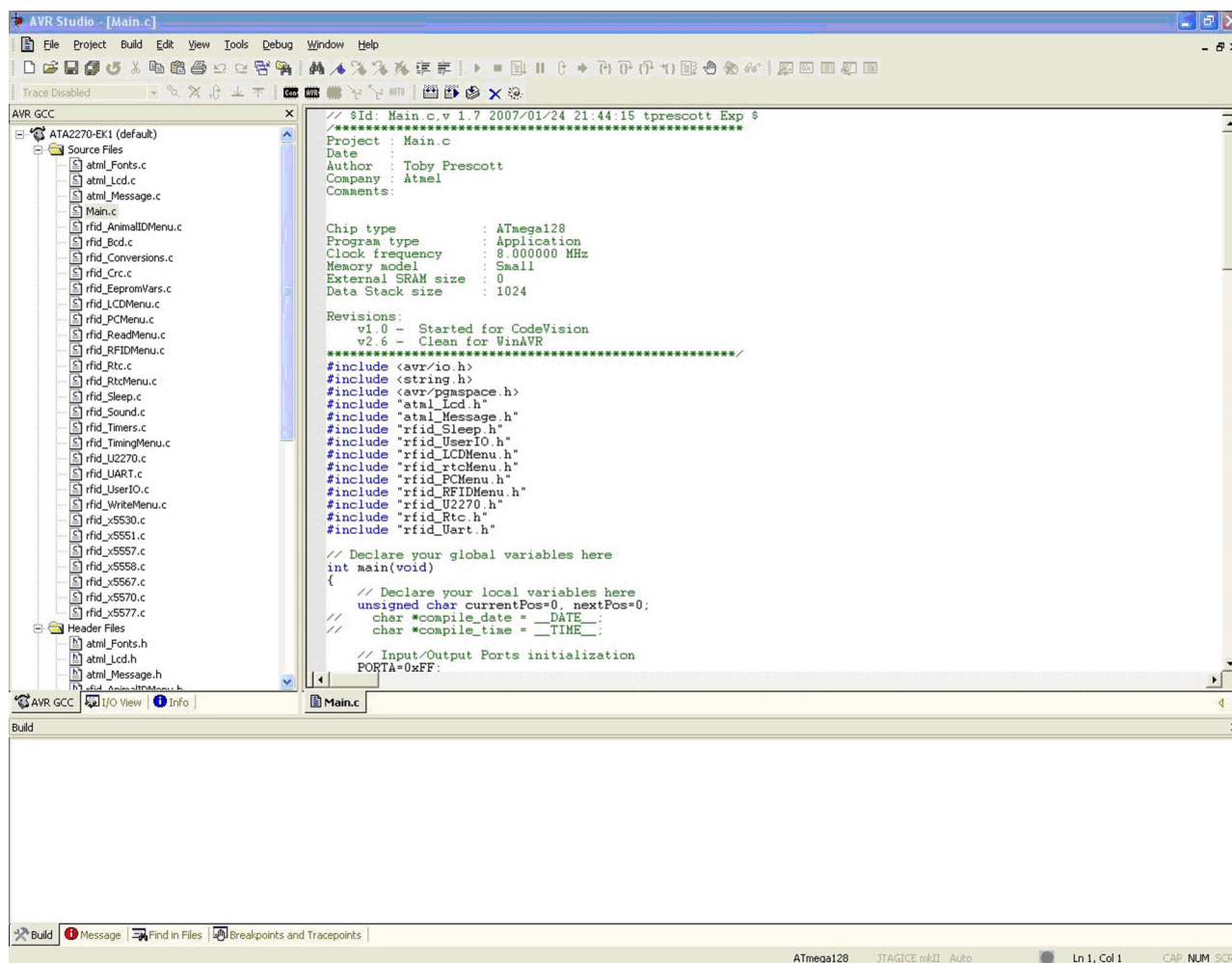
JTAG Programming/Debugging

The same steps apply when using the JTAGICE MKII as a simple programmer. Simply connect the JTAG 10-pin cable to the MB header, above the ISP connector, and proceed as detailed in the ISP section.

The benefit of the JTAG Platform is that it allows more control during software development by allowing real-time debugging of the source code. The JTAGICE MKII supports setting break-points and stepping through the code being executed on the target micro in real-time. This is very useful for a more complex software project such as this one.

To get started, make sure that the latest version of AVR Studio and WinAVR has been installed on the PC. Download the latest revision of the firmware for the ATA2270-EK1 kit from the Atmel web site and unzip it to a working folder. Start AVR Studio and click the "Open" button on the welcome screen. Select the "ATA2270-EK1.aps" file from the working folder.
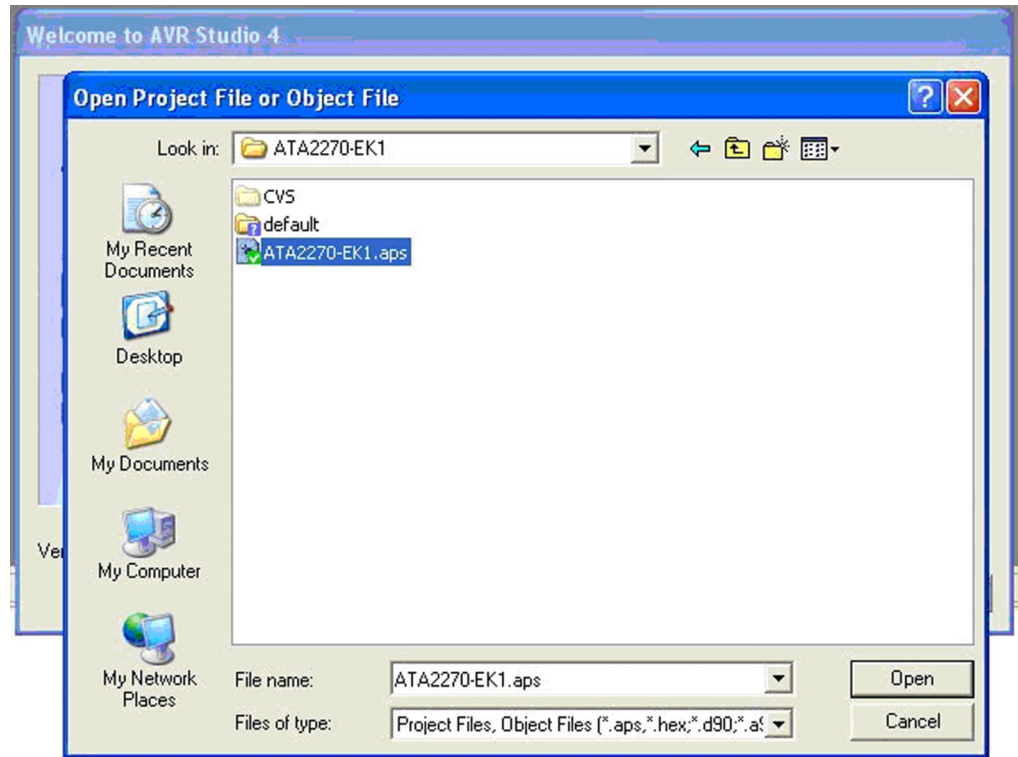
This will open up the C complier in AVR Studio and allow development and debugging of the firmware all from one simple platform. Once opened, this platform displays the source files on the left (both .c and .h) and allows editing of each file in the main screen. Once the changes have been made, real-time debugging is achieved by selecting "Build" then "Build and Run" from the menu. This will compile the source code, flash the target device, and begin code execution.

AVR Studio - [Main.c]

File   Project   Build   Edit   View   Tools   Debug   Window   Help

Trace Disabled

AVR GCC

- ATA2270-EK1 (default)
  - Source Files
    - atml_Fonts.c
    - atml_Lcd.c
    - atml_Message.c
    - Main.c
    - rfid_AnimalIDMenu.c
    - rfid_Bcd.c
    - rfid_Conversions.c
    - rfid_Crc.c
    - rfid_EepromVars.c
    - rfid_LCDMenu.c
    - rfid_PCMenu.c
    - rfid_ReadMenu.c
    - rfid_RFIDMenu.c
    - rfid_Rtc.c
    - rfid_RtcMenu.c
    - rfid_Sleep.c
    - rfid_Sound.c
    - rfid_Timers.c
    - rfid_TimingMenu.c
    - rfid_U2270.c
    - rfid_UART.c
    - rfid_UserIO.c
    - rfid_WriteMenu.c
    - rfid_x5530.c
    - rfid_x5551.c
    - rfid_x5557.c
    - rfid_x5558.c
    - rfid_x5567.c
    - rfid_x5570.c
    - rfid_x5577.c
  - Header Files
    - atml_Fonts.h
    - atml_Lcd.h
    - atml_Message.h
    - rfid_AnimalIDMenu.h

```
// $Id: Main.c,v 1.7 2007/01/24 21:44:15 tprescott Exp $
/*****************************************************
Project : Main.c
Date    :
Author  : Toby Prescott
Company : Atmel
Comments:


Chip type          : ATmega128
Program type       : Application
Clock frequency    : 8.000000 MHz
Memory model       : Small
External SRAM size : 0
Data Stack size    : 1024

Revisions:
    v1.0 -  Started for CodeVision
    v2.6 -  Clean for WinAVR
*****************************************************/
#include <avr/io.h>
#include <string.h>
#include <avr/pgmspace.h>
#include "atml_Lcd.h"
#include "atml_Message.h"
#include "rfid_Sleep.h"
#include "rfid_UserIO.h"
#include "rfid_LCDMenu.h"
#include "rfid_rtcMenu.h"
#include "rfid_PCMenu.h"
#include "rfid_RFIDMenu.h"
#include "rfid_U2270.h"
#include "rfid_Rtc.h"
#include "rfid_Uart.h"

// Declare your global variables here
int main(void)
{
    // Declare your local variables here
    unsigned char currentPos=0, nextPos=0;
//    char *compile_date = __DATE__;
//    char *compile_time = __TIME__;

    // Input/Output Ports initialization
    PORTA=0xFF;
```

AVR GCC   I/O View   Info   Main.c

Build

Build   Message   Find in Files   Breakpoints and Tracepoints

ATmega128   JTAGICE mkII   Auto   Ln 1, Col 1   CAP NUM SCRL

Serial Port Bootloader Programming

This section applies only to kits with firmware v3.0 or later already programmed into the micro-controller. The bootloader firmware allows the kit to be upgraded using only the serial port on the kit. In order to begin, on the ATA2270-EK1 first navigate to the PC>Upgrade Firmware option and press ENTER. Press ENTER a second time to confirm the desire to enter bootloader mode.

On the PC, open AVR Studio and choose Tools>AVR Prog from the menu. Press the Browse button and choose the appropriate flash (Intel-hex) file to load. This should have a .hex extension.
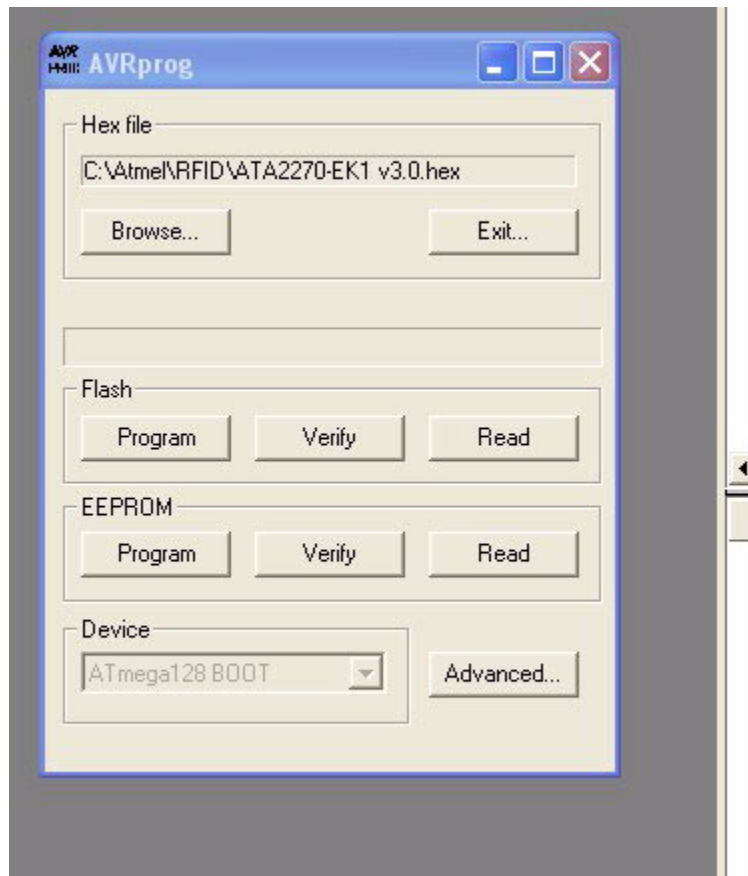
Press the Program button under the Flash section. Answer Yes to the two warning messages that will appear. These are warnings that the boot section can not be overwritten and that portion of the file will be ignored.

Again press the Browse button and locate the EEPROM file. This has a .eep extension. Press the Program button located in the EEPROM section.

Once both of these actions have been successful, pressing the Exit button will leave bootloader mode. The ATA2270-EK1 should re-start and display the main screen with the newly upgraded firmware.

The complete User Guide for the ATA2270-EK1 is available on the Atmel website along with other Application Notes and Datasheets to help with problems encountered. If you have any other questions about using the Evaluation Kit or designing with Atmel RFID, please do not hesitate to contact us:

Toby Prescott, North American RFA Applications Engineer, 719-540 6947, tprescott@cso.atmel.com

Jim Goings, North American RFA Applications Manager, 719-540-6873, jgoings@cso.atmel.com

Nicolas Schieli, North American RFID Business Development, 514-878-8113 x228, nschieli@cso.atmel.com

Revision History

Doc Rev.	Date	Comments
5241A	12/2007	Initial document release.

Headquarters

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

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel 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

Product Contact

Web Site
www.atmel.com

Technical Support
rfid@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature