Programming the AVR and ARM on SeatUnit v4

Doc v 3.76a Date 5 February 2011

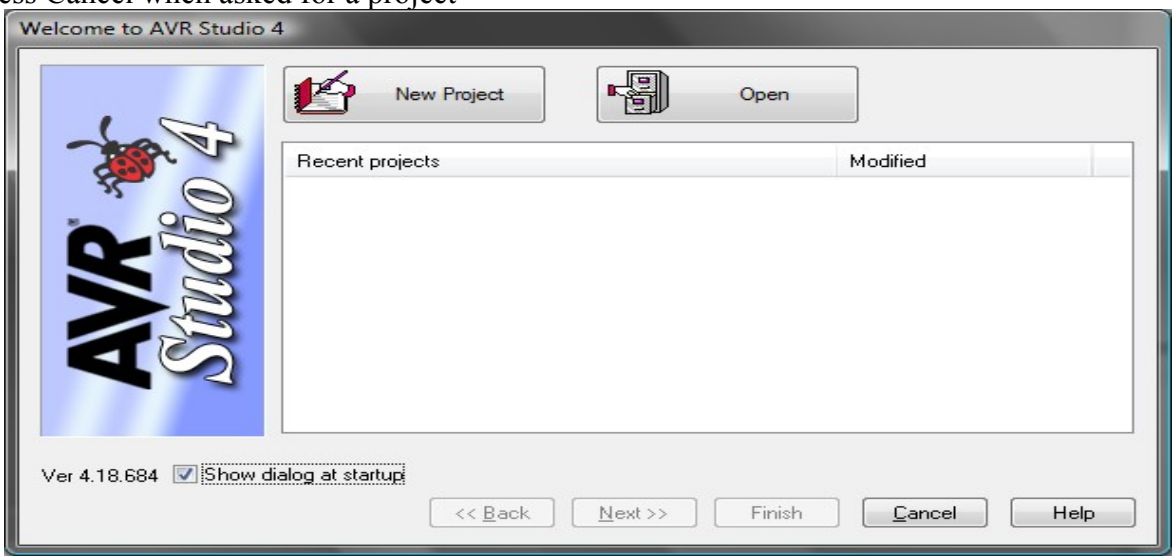
Note you must be logged in as administrator. This software was set up under Peter's name.

All relevant files are under c:\SeatUnit4\

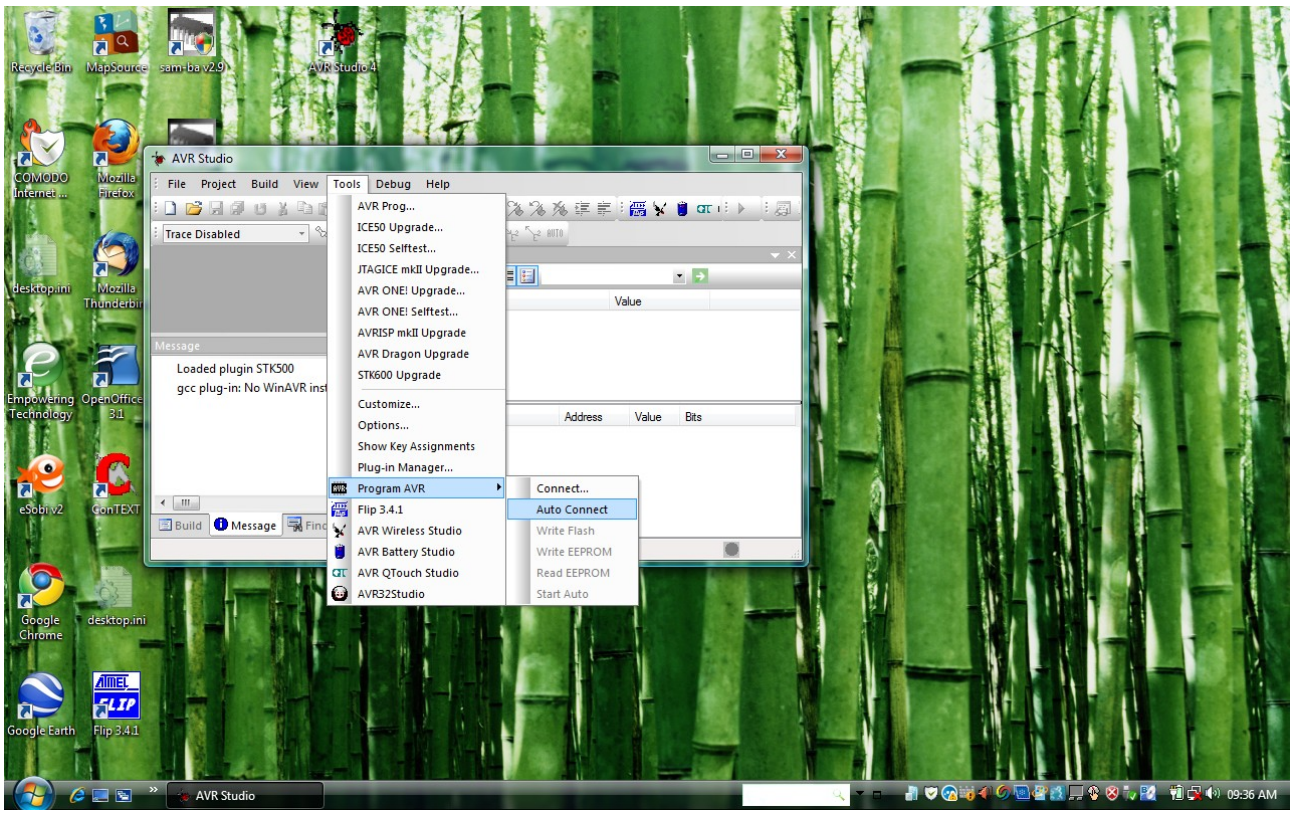
The programming connector pinouts for manufacturing is given in the Appendix.

Programming the AVR

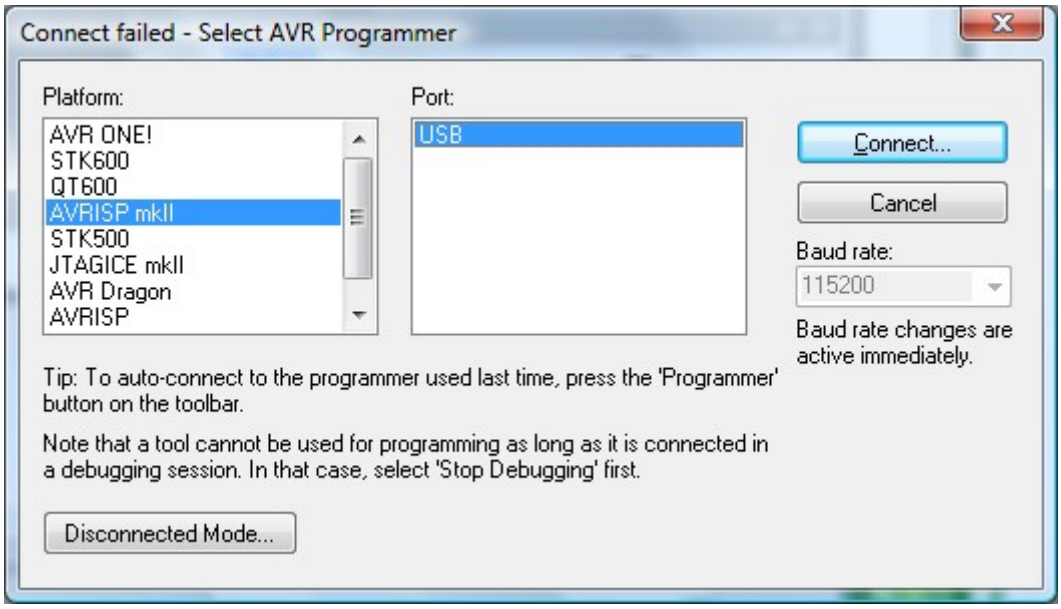
The programmer should be run as administrator. It is possible to run it as a non administrator but it will ask you for the pass word and it will give you security warnings all the time.
1.Launch AVR Stidio4 from the desktop or from the main menu.
2.Press Cancel when asked for a project



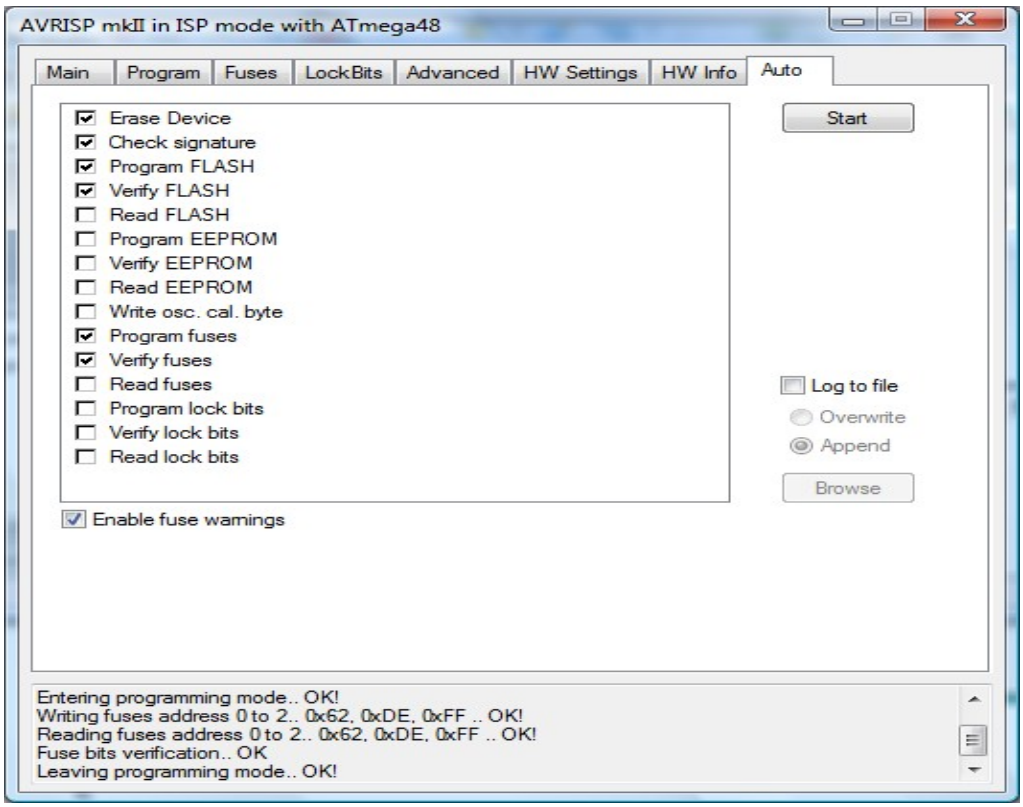
3.Select Tools/Program AVR/Auto Connect from the menus



4.Select AVRISPmkII under Platform
Select USB under connection
5.Choose Connect



6.The programmer tabs should be set as in Appendix A. This only need to be done once. Warning: Note that an incorrect setting can render the board unuseable, especially if the fuses are setup incorrectly this error can not be recovered from.
7.Select the Auto tab and setup as below

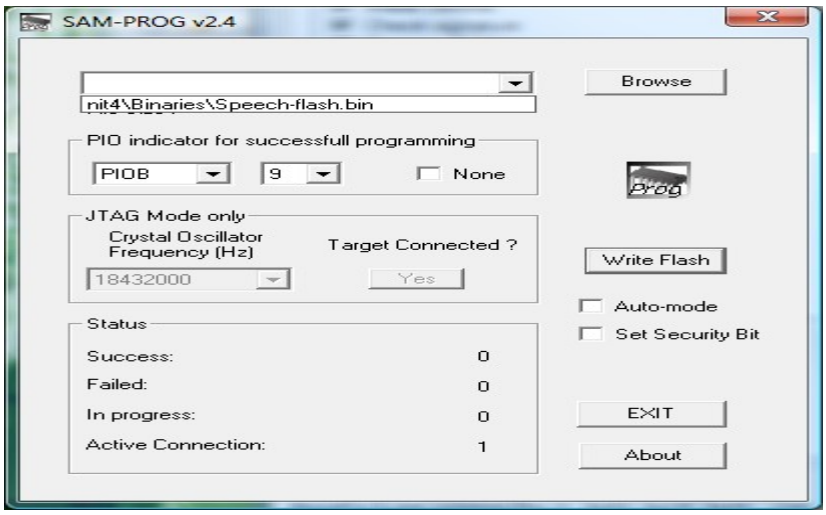


8.To program press start
9.After programming the screen should display “HEL” and “LO-” if programmed correctly.
10.Program the ARM as described in the next section.
11.The next AVR on a new board can be programmed by simply plugging in the connectors and pressing “Start” again. (NO need to restart the program as with the ARM)

Programming the ARM

The ARM can only be programmed if the FLASH memory is unprogrammed. (New ARM's are unprogrammed) One the ARM has been programmed you have to wipe the memory as described later.

1.Start the program “SAM-PROG” either from the desktop or the main windows menu.
2.Select “Browse” button if it is the first time you are running the programm. Select the file to program. (Default it is in c:\SeatUnit4\Binaries\FILE.bin) If the file has been loaded before you can simply press the down triangle/arrow next to the Browse button and select the file from the list.



3.If the board is plugged in it will show a 1 next to the “Active Connection”
4.Press the “Write Flash” button.
5.There should be briefe activity next to the “In Progress” line and the “Success” Line will flicker briefly with a 1 before returning to a 0
6.Press “EXIT” button

Testing

A quick test to see if the programming was successful is:

1. Switch off power and remove programming connctor
2. Switch on power
3. The unit should display “HELLO-” briefely
4. Next it should display “ID5” and the number of the seat. e.g. “A02”

If it does not display anything try to reprogram the AVR.
If it does no display the ID try and reprogram the ARM.
This will give you an indication that programming was successful. The mp3 decoders and amps can only be tested when playing music.

Difference between the 2 programmers

The AVR programmer only need to be opened once, after that a new board can be plugged in and programmed by simply pressing the “Start” button every time.
The SAM programmer must be restarted every time to program a new board. This due to the USB that must establish a new connection every time a USB is plugged in and out. The softwatre must therefore restart to establish a connection to the new board.

Reprogramming the AVR

No special actions need to be taken, simply plug in the old board and reprogram it.

Reprogramming the ARM

Before the ARM can be reprogrammed, it's old program must be wiped. This is done by:

1.With no power flip the programming switch to on (hold in the programming button).

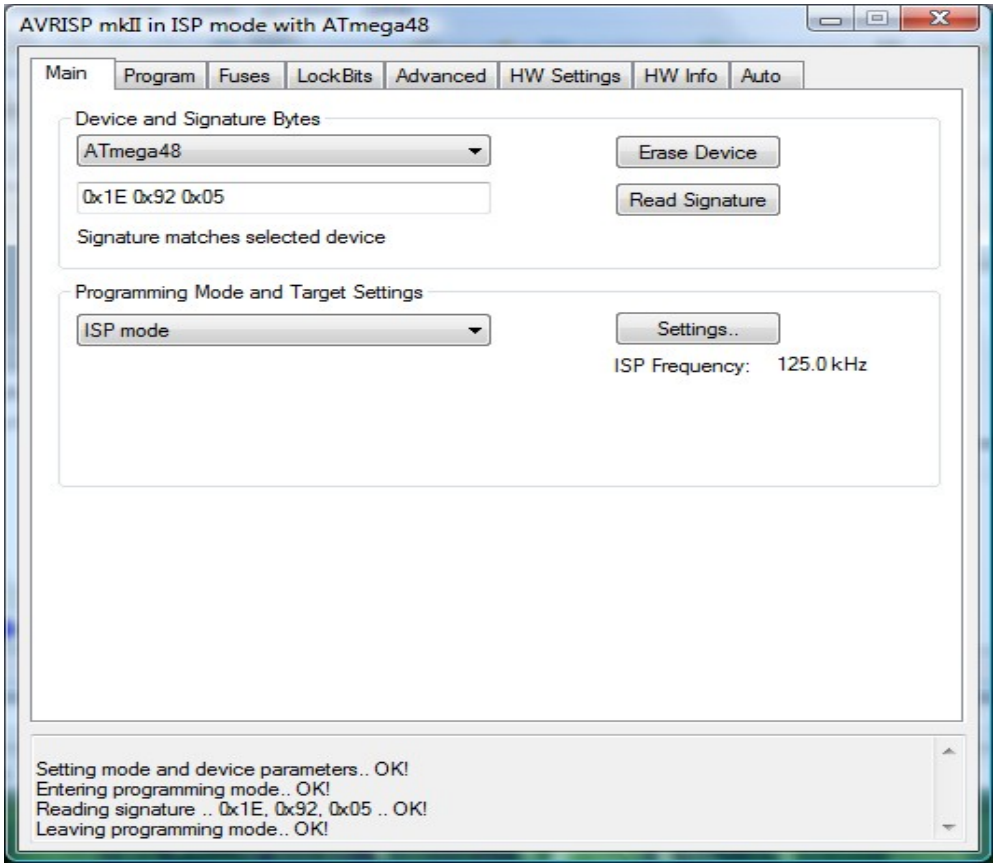
2.Power the unit for 7 to 10 seconds (with the programming button in)

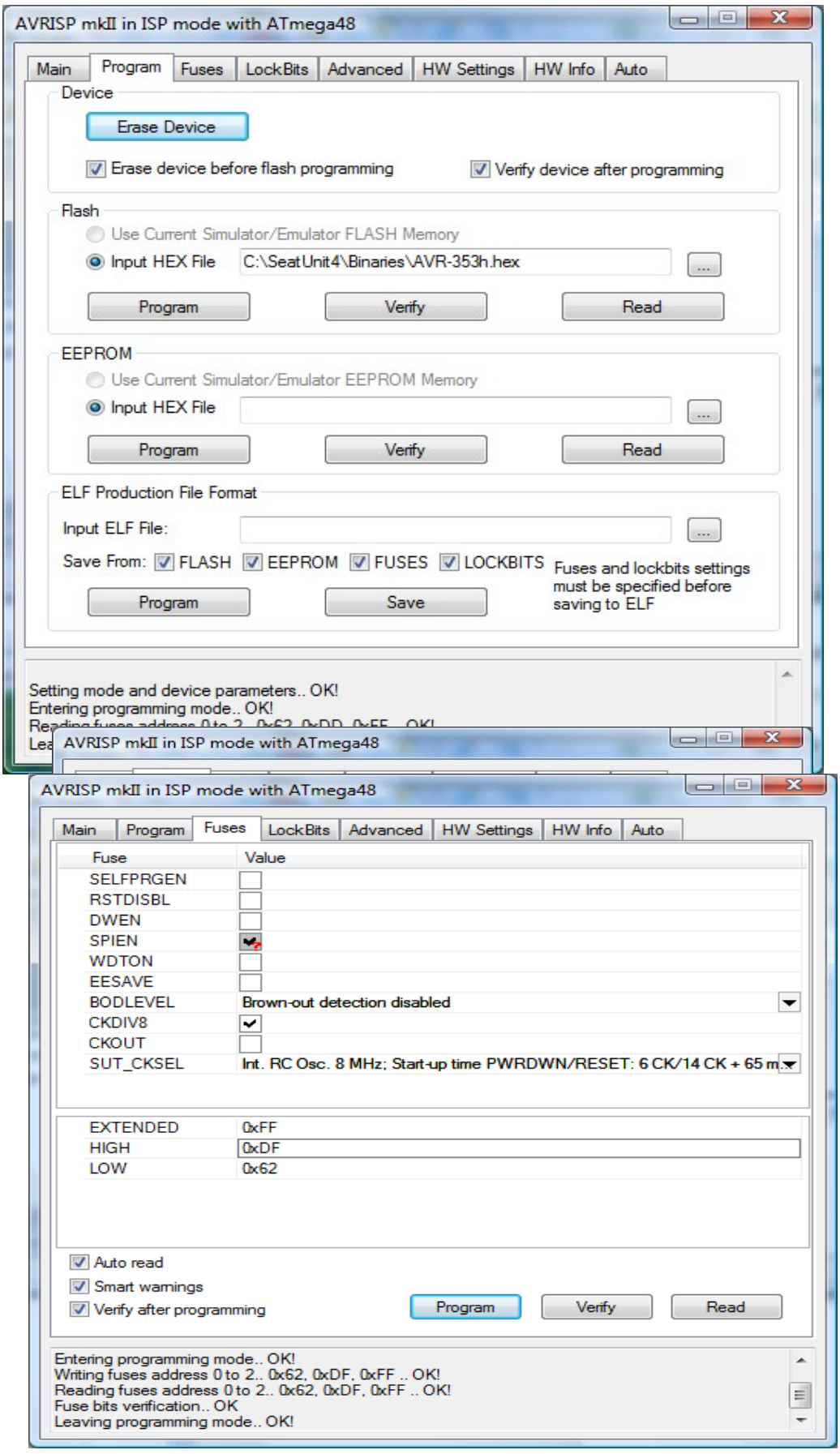
3.Remove power

4.Flip the switch to off again (release the programming button)

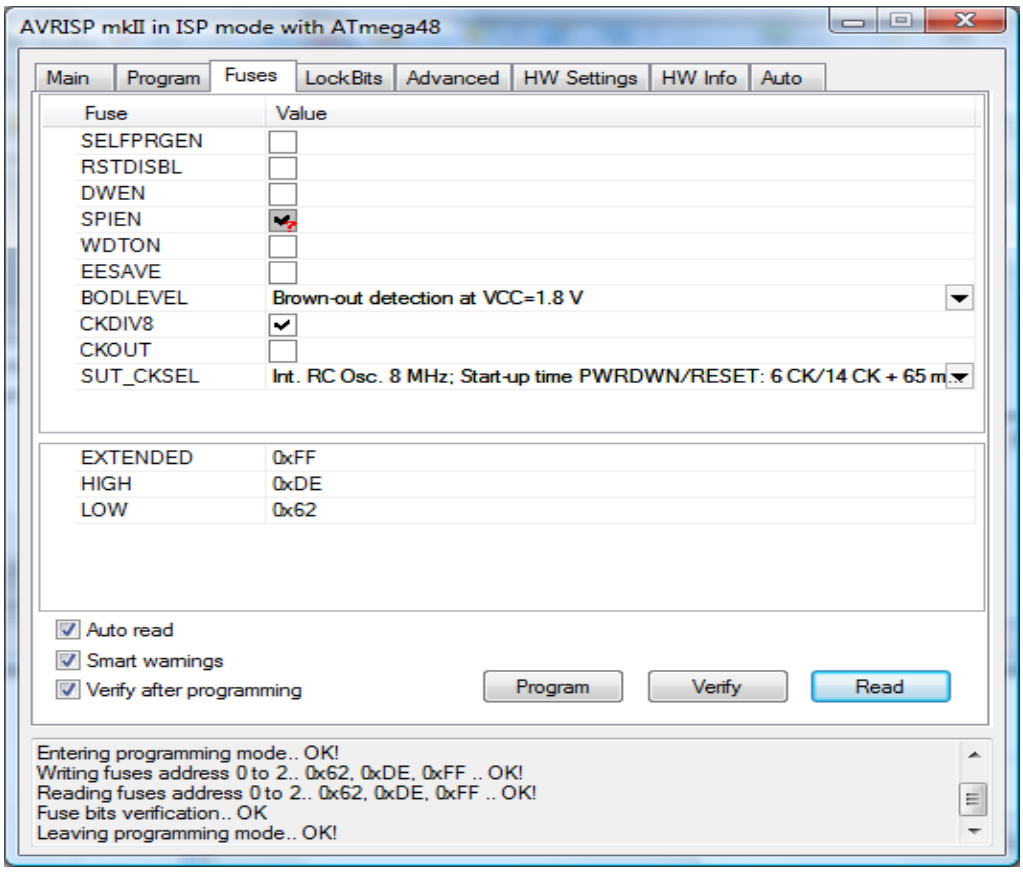
5.Continue with the normal programming sequence

Appendix A: Setup of AVR Studio 4 programming



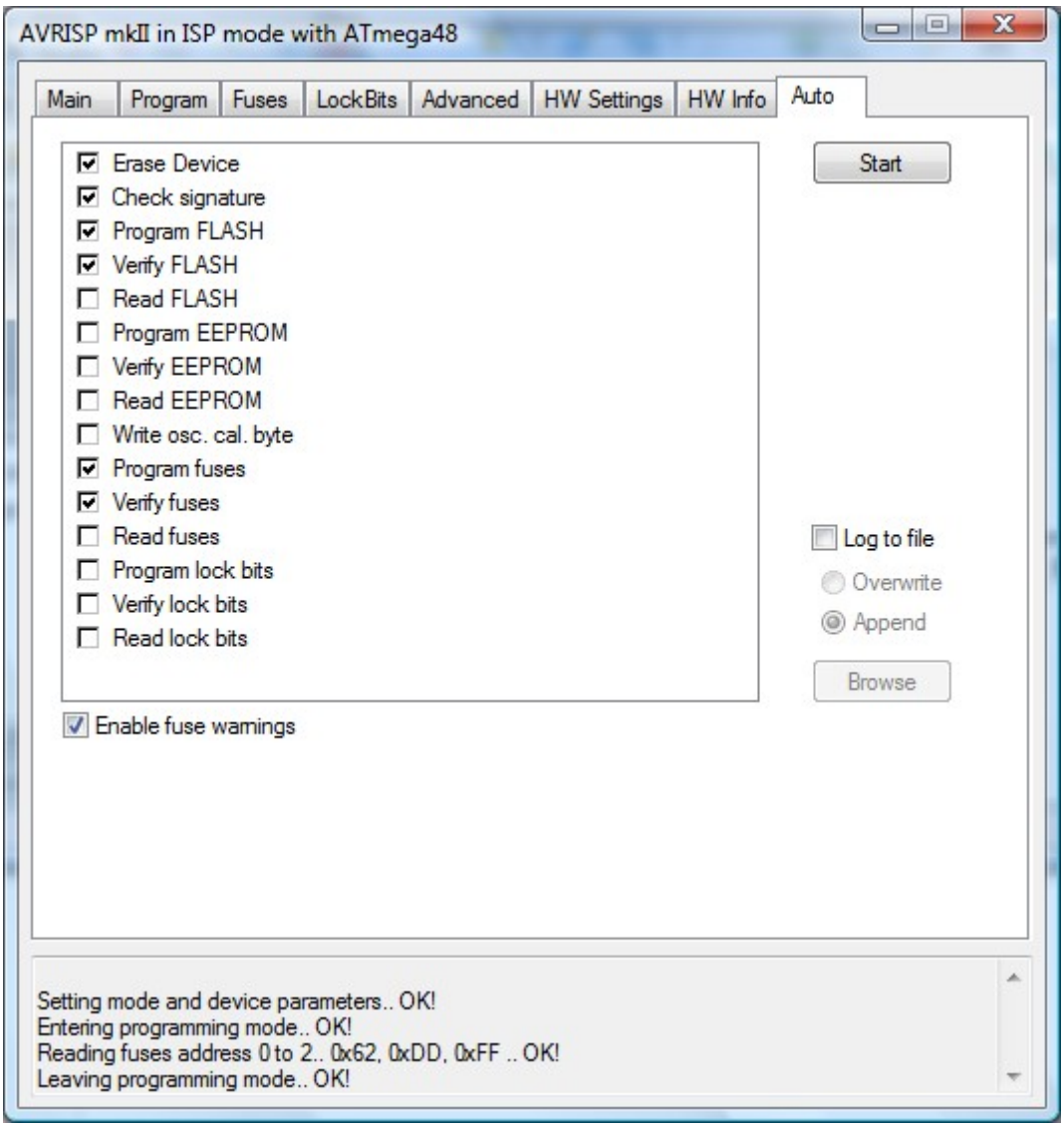

AVRISP mkII in ISP mode with ATmega48
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Device
Erase Device
Erase device before flash programming
Verify device after programming
Flash
Use Current Simulator/Emulator FLASH Memory
Input HEX File
C:\SeatUnit4\Binaries\AVR-353h.hex
Program
Verify
Read
EEPROM
Use Current Simulator/Emulator EEPROM Memory
Input HEX File
Program
Verify
Read
ELF Production File Format
Input ELF File:
Save From:
FLASH
EEPROM
FUSES
LOCKBITS
Fuses and lockbits settings must be specified before saving to ELF
Program
Save
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
AVRISP mkII in ISP mode with ATmega48
AVRISP mkII in ISP mode with ATmega48
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Fuse
Value
SELFPRGEN
RSTDISBL
DWEN
SPIEN
WDTON
EESAVE
BODLEVEL
Brown-out detection disabled
CKDIV8
CKOUT
SUT_CKSEL
Int. RC Osc. 8 MHz; Start-up time PWRDWN/RESET: 6 CK/14 CK + 65 m
EXTENDED
0xFF
HIGH
0xDF
LOW
0x62
Auto read
Smart warnings
Verify after programming
Program
Verify
Read
Entering programming mode.. OK!
Writing fuses address 0 to 2.. 0x62, 0xDF, 0xFF .. OK!
Reading fuses address 0 to 2.. 0x62, 0xDF, 0xFF .. OK!
Fuse bits verification.. OK
Leaving programming mode.. OK!

or



The following 4 tabs do not need any changes.

The last tab should be set as follows:


AVRISP mkII in ISP mode with ATmega48
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Erase Device
Check signature
Program FLASH
Verify FLASH
Read FLASH
Program EEPROM
Verify EEPROM
Read EEPROM
Write osc. cal. byte
Program fuses
Verify fuses
Read fuses
Program lock bits
Verify lock bits
Read lock bits
Start
Log to file
Overwrite
Append
Browse
Enable fuse warnings
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading fuses address 0 to 2.. 0x62, 0xDD, 0xFF .. OK!
Leaving programming mode.. OK!

Appendix: Programming Connector Cable Pinout

Use a 12 way flat ribbon cable suitable for IDC connectors. Attach the ribbon cable to a 12 way dual row IDC female connector. Solder a host USB connector (also called a USB A connector), a 2.54mm 6 pin dual row header and a switch on to the ribbon cable as in the table below. Please note the polarity on the IDC connectors very carefully.

PCB IDC No.	**Name**	**ARM USB programming (colors might differ)**	**AVR programmer**	**ARM memory delete switch**
1	3V3			Connect to NO
2	ERASE ARM			Connect to common
3	USB Detect	USB +5V (red)		
4	USB DDM	USB DDM (white)		
5	USB DDP	USB DDP (green)		
6	GND	USB GND (black)		
7	MISO		1 IDC connector	
8	3V3		2 IDC connector	
9	SCK		3 IDC connector	
10	MOSI		4 IDC connector	
11	/AVR RST		5 IDC connector	
12	GND		6 IDC connector	