Required Hardware & Software:
- AVR Studio 4.07, Build 240
- AVR Studio LCD Plug-In
- AVR JTAG ICE, Firmware Version 0x78
- AVR Butterfly Evaluation Kit with JTAG Header
- 12 V Power Supply for JTAG ICE
- Serial Cable for JTAG ICE
- IAR Systems Embedded Workbench AVR 2.28A

Required Source Files:
- Source Files included in HVAC_SEM.ZIP archive

Preparations:
- Extract the Source Files to a directory on the Hard Drive
- If the Demonstration has been previously held, make sure that only the original source files are located in the directory. It is important to delete all other files for the demonstration to follow the script
- If the ATmega169 has previously been used, make sure that the device has been erased and the fuses set to their correct values prior to the demonstration. The fuses should have checks in the following boxes within AVR Studio
  - Brown Out Detection Disabled [BODLEVEL = 111]
  - On-Chip Debug Enabled [OCDEN = 1]
  - JTAG Interface Enabled [JTAGEN = 1]
  - Serial Program Downloading Enabled [SPIEN = 1]
  - Internal RC, Start-up Time 6CK + 65 ms [CKSEL = 0010, SUT = 10]
- Connect the JTAG ICE to the JTAG Connector on the AVR Butterfly
- Connect the Serial Cable between the PC and the JTAG ICE
- Power the JTAG ICE
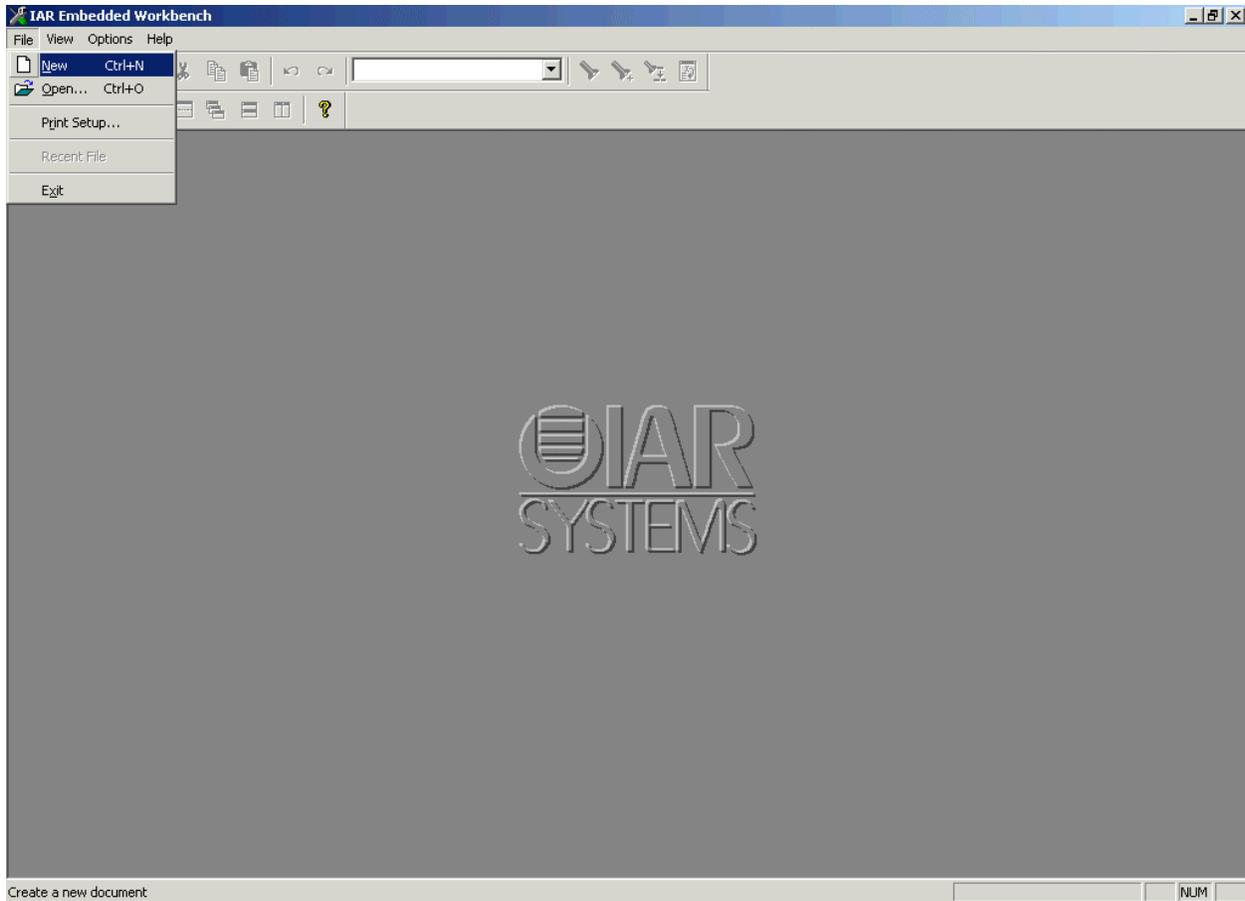
1. The IAR Embedded Workbench for AVR fully integrates the IAR C and Embedded C++ optimizing compiler, assembler, linker, librarian, text editor, and project manager
2. With its built-in chip-specific optimizer, the compiler generates very efficient, fast and reliable code for the AVR family
3. First, a New Project needs to be created

Steps to Perform:
1. Open IAR Embedded Workbench for AVR 2.28A
2. Select "New" from the "File" menu
3. Select "Project" from the following dialog and then press "OK"
4. Browse to the directory where the Seminar Demo files were extracted
5. Enter a Project Name (i.e. "HVAC_DEMO")
6. Click on "Create"
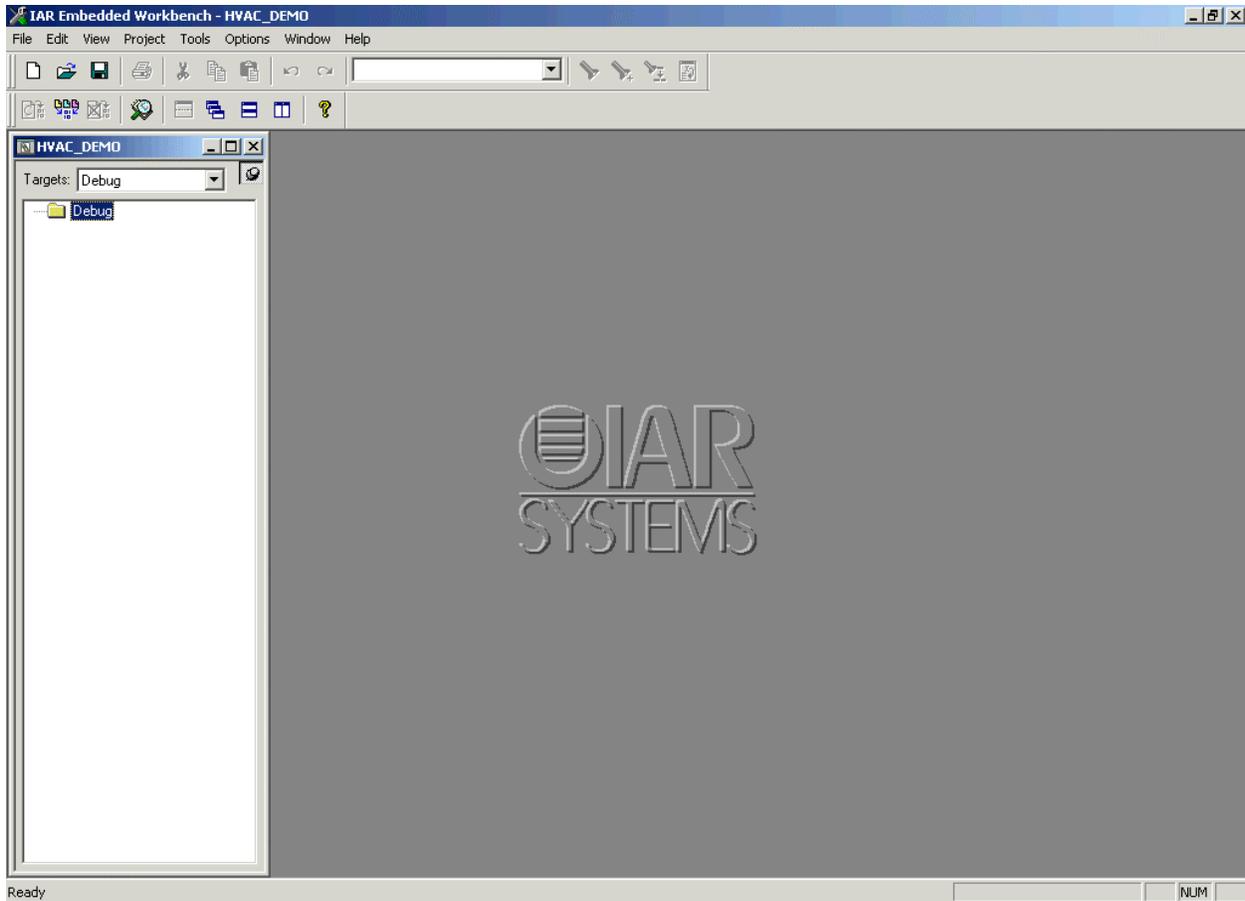
1. Now we have an empty project, we need to add source files

Steps to Perform:
   1. Select "Files" from the "Project" menu
   2. Select "Add All" from the subsequent dialog and press "Done"
   3. Expand the "Common Sources" option in the Project Window to view the Source Files
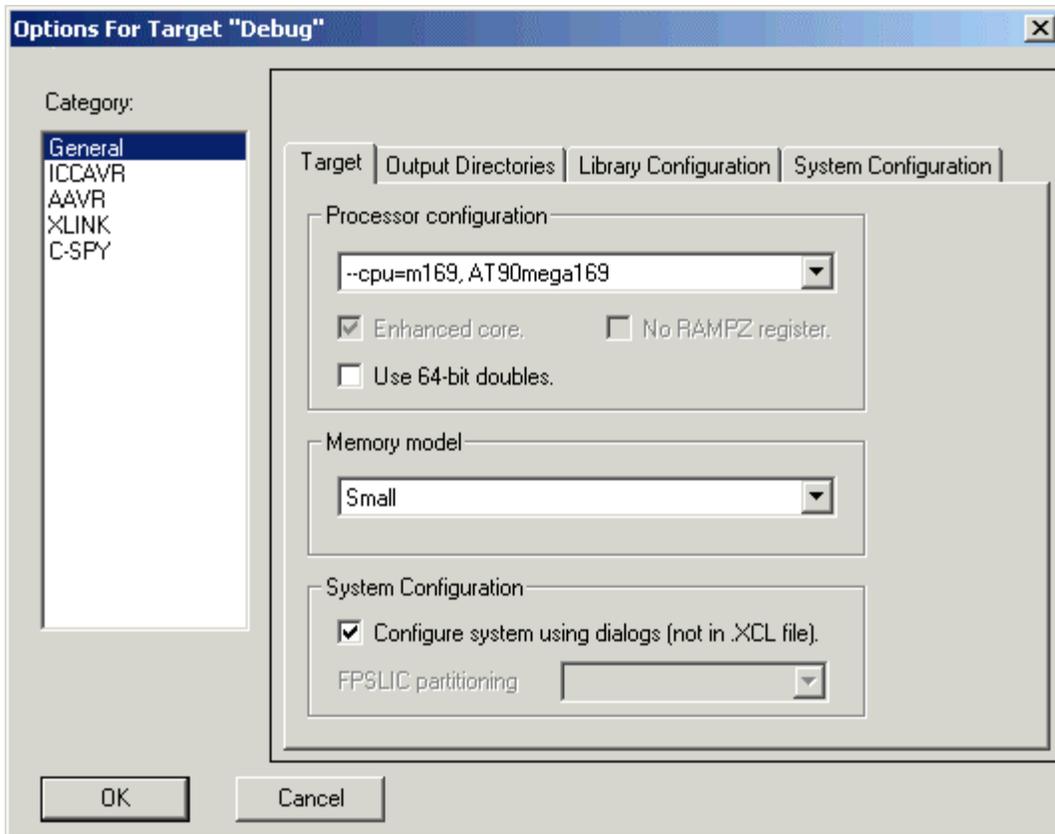
1. Select target device, targeting ATmega169
2. Select Small Memory Model as we have more that 127 Bytes of variables

Steps to Perform:
1. Select "Options…" from the "Project" menu
2. Under the "General" Category and "Target" tab,
   a. Select "—cpu=m169, AT90mega169" for "Processor Configuration"
   b. Select "Small" for "Memory Model"
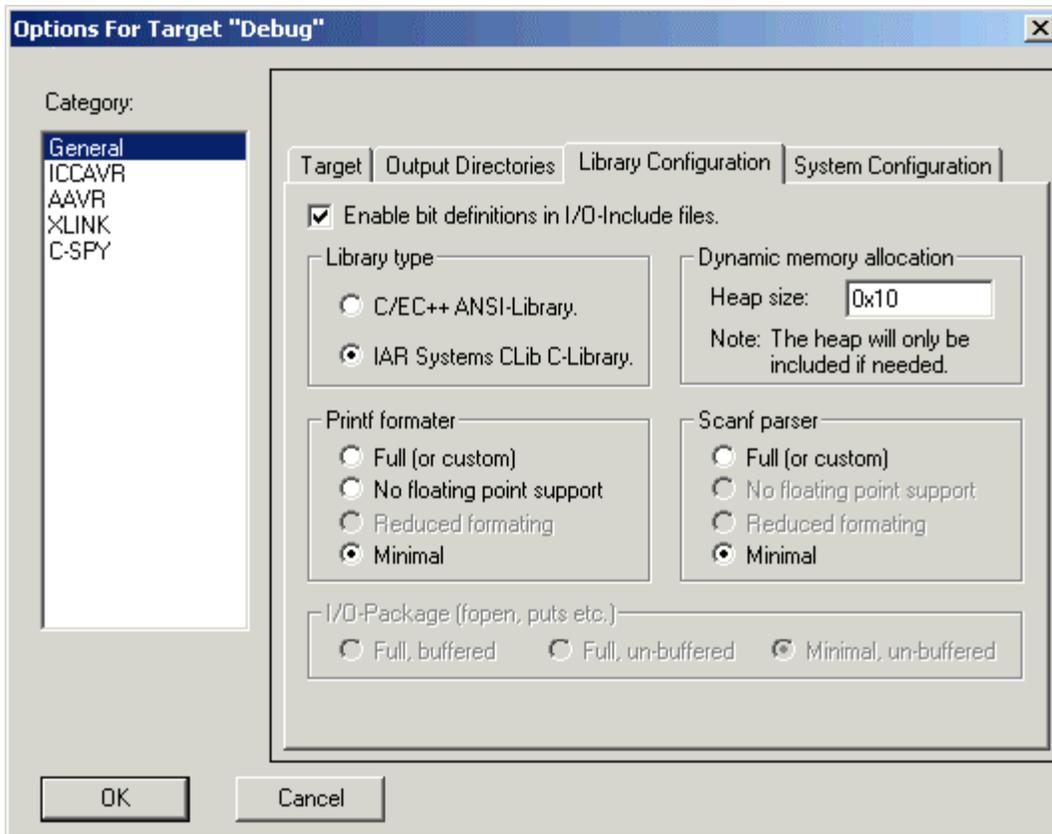
1. Enable Bit Definitions, allows for access of bits in I/O Registers by their respective bit names (i.e. PB7 in PORTB)

Steps to Perform:
1. Under the "General" Category and "Library Configuration" tab,
   a. Select the "Enable bit definitions in I/O-Include files" option
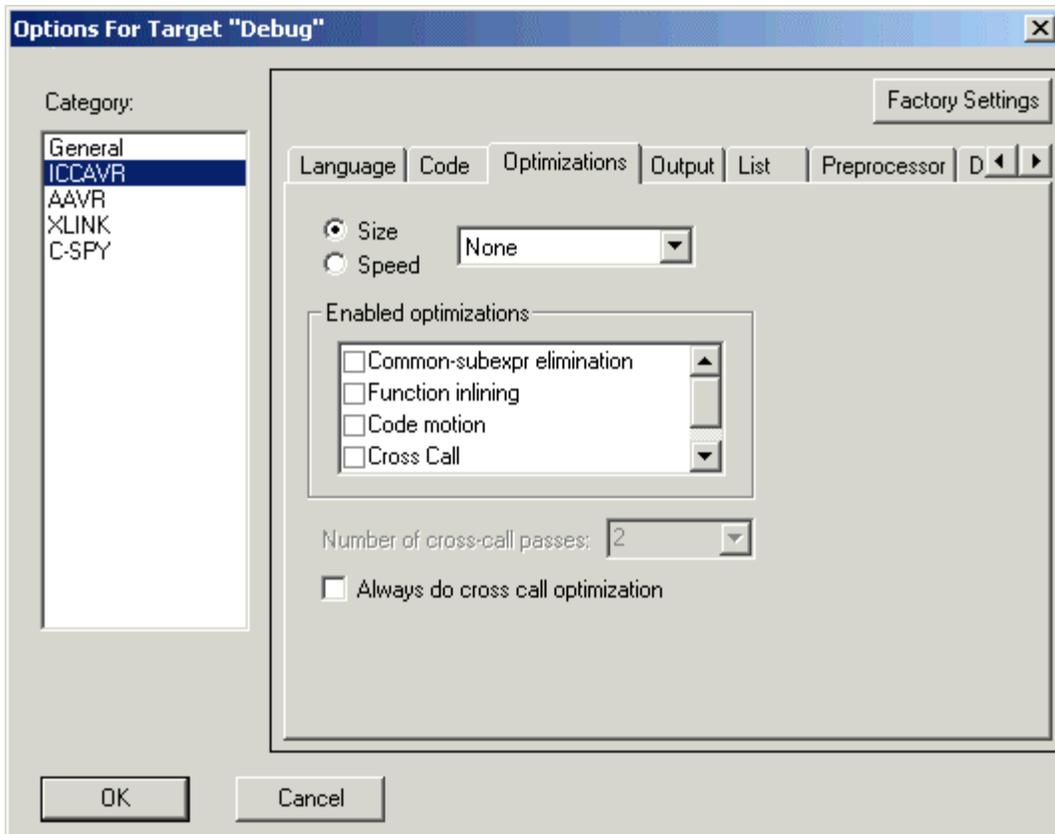
1. Disable Optimizations, will re-compile with full optimization if time permits

Steps to Perform:
1. Under the "ICCAVR" Category and "Optimization" tab,
   a. Select "None" for Optimization Level
   b. Un-check any boxes in the "Enabled Optimizations" field
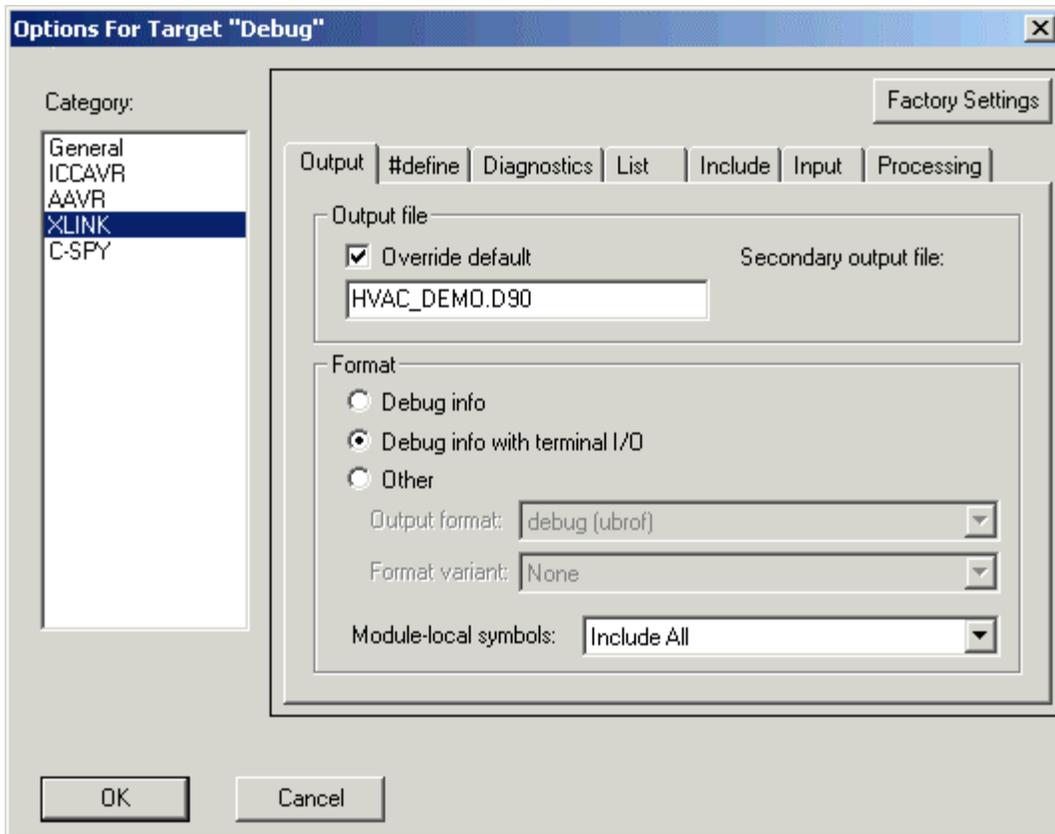
1. Specify the Name of the Output File as a .d90

What to Perform:
1. Under the "XLINK" Category and "Output" tab,
    a. Select "Override Default" under "Output File" and change the extension to ".d90"
2. Press "OK"
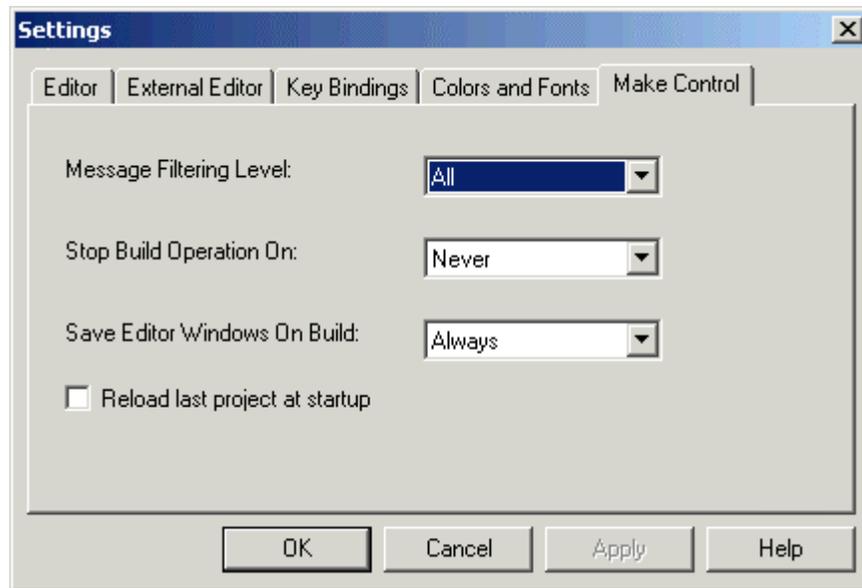
1. Adjust the "Message Filtering Level" to display all information regarding the compilation, linking, and general message information

Steps to Perform:
1. Select "Settings…" from the "Options" menu
2. Under the "Make Control" tab,
    a. Select "All" for "Message Filtering Level"
    b. Un-check "Reload last project at startup"
3. Press "OK"
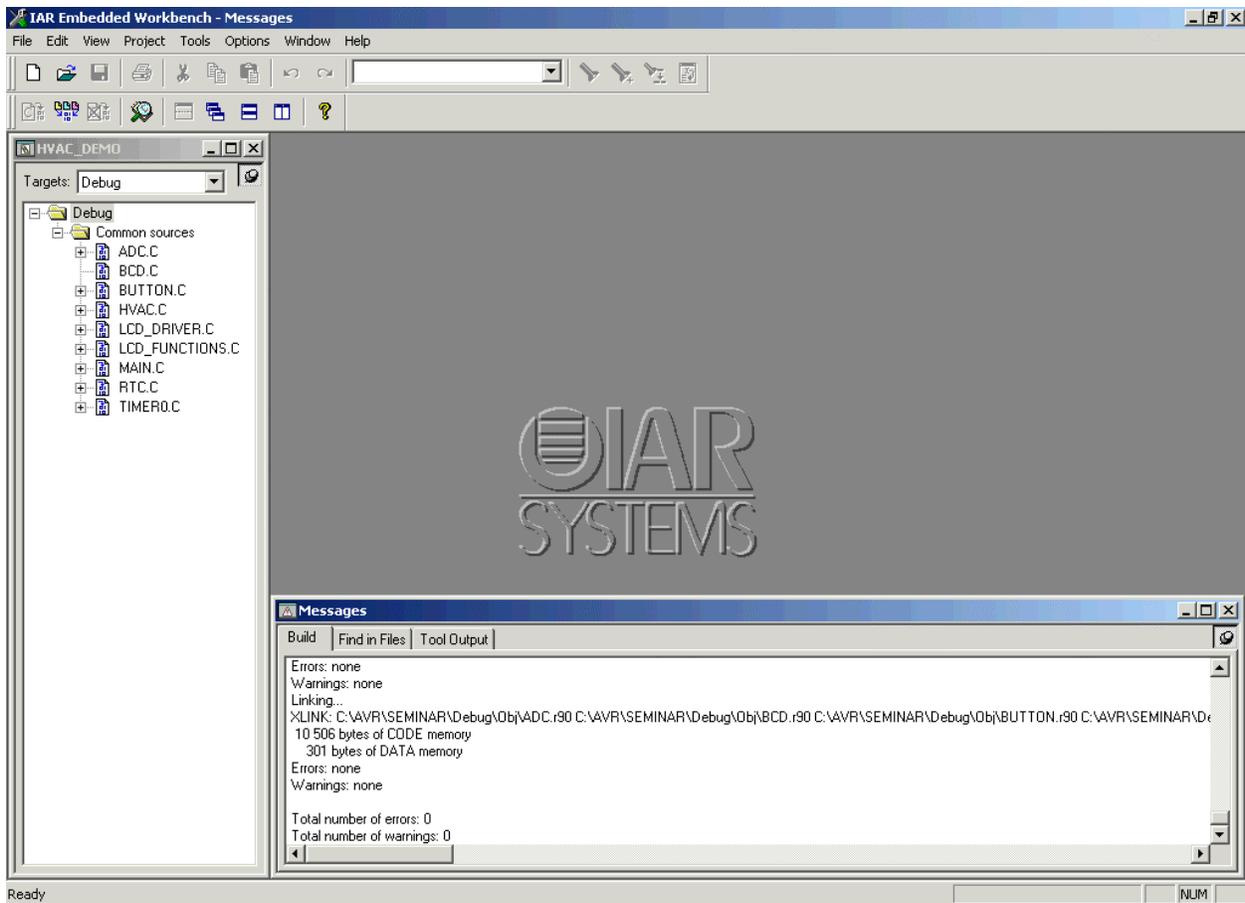
1. A successful compile – No Errors or Warnings
2. Consuming 10,506 Bytes Flash Memory and 301 Bytes of SRAM

Steps to Perform:
1. Select "Build All…" from the "Project" menu
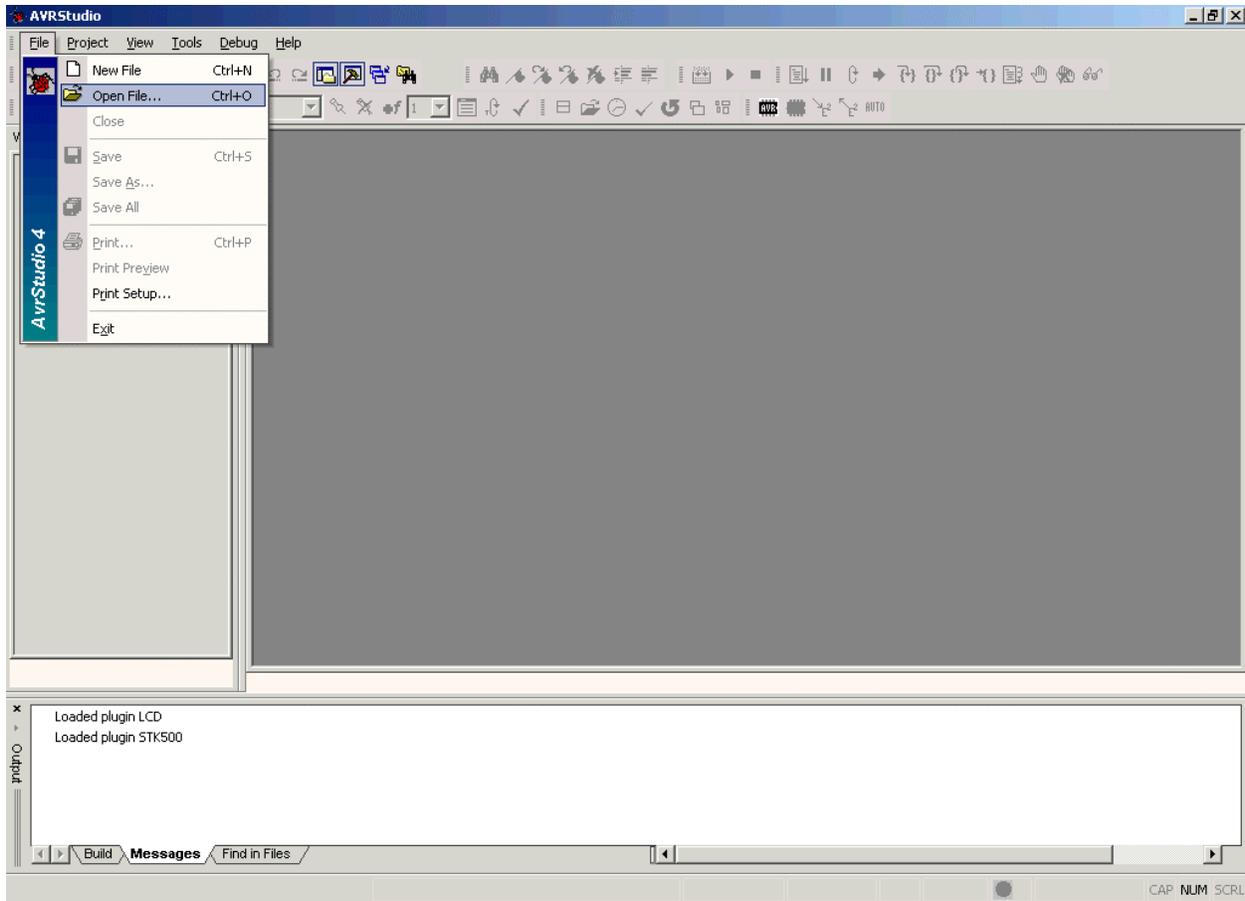
1. Object files contain the program and debug information. Debug information are references between source code and the low-level AVR instructions. There is also information about variable names, their memory locations and their types in the object file.

Steps to Perform:
1. Open AVR Studio 4.07
2. Select "Open File…" from the "File" menu
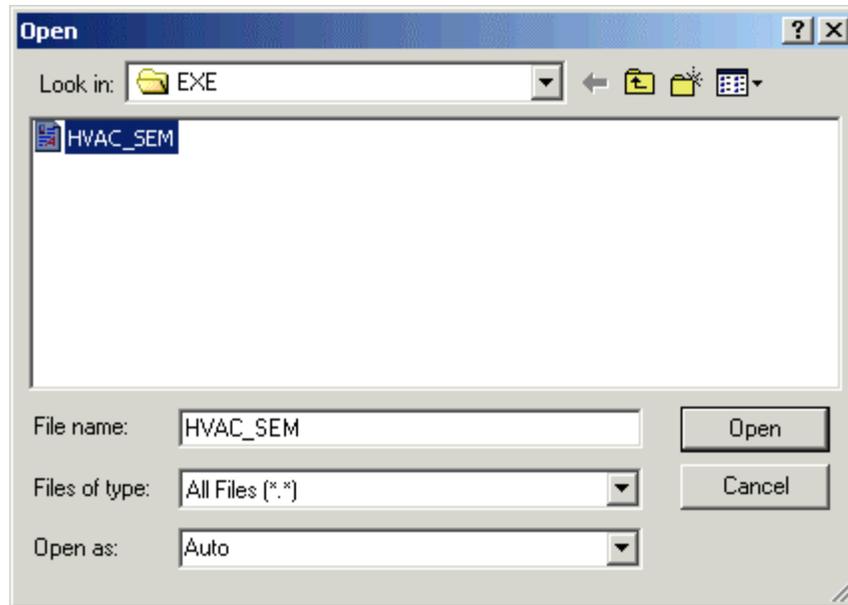
Steps to Perform:
1. Browse to the "..\DEBUG\EXE" sub-directory of your project directory
2. Select the "HVAC_SEM.D90" Object File
3. Press "Open"

Select device and debug platform

Select debug platform and device

Debug Platform:
- ICE40
- ICE50
- **JTAG ICE**
- AVR Simulator
- ICE200

Device:
- ATmega128
- ATmega16
- ATmega162
- **ATmega169**
- ATmega32
- ATmega323
- ATmega64
- ATmega8
- ATtiny26
- AT86RF401
- AT90S2313

Connect [Auto ▼]

[Help] [<< Back] [Next >>] [Finish] [Cancel]
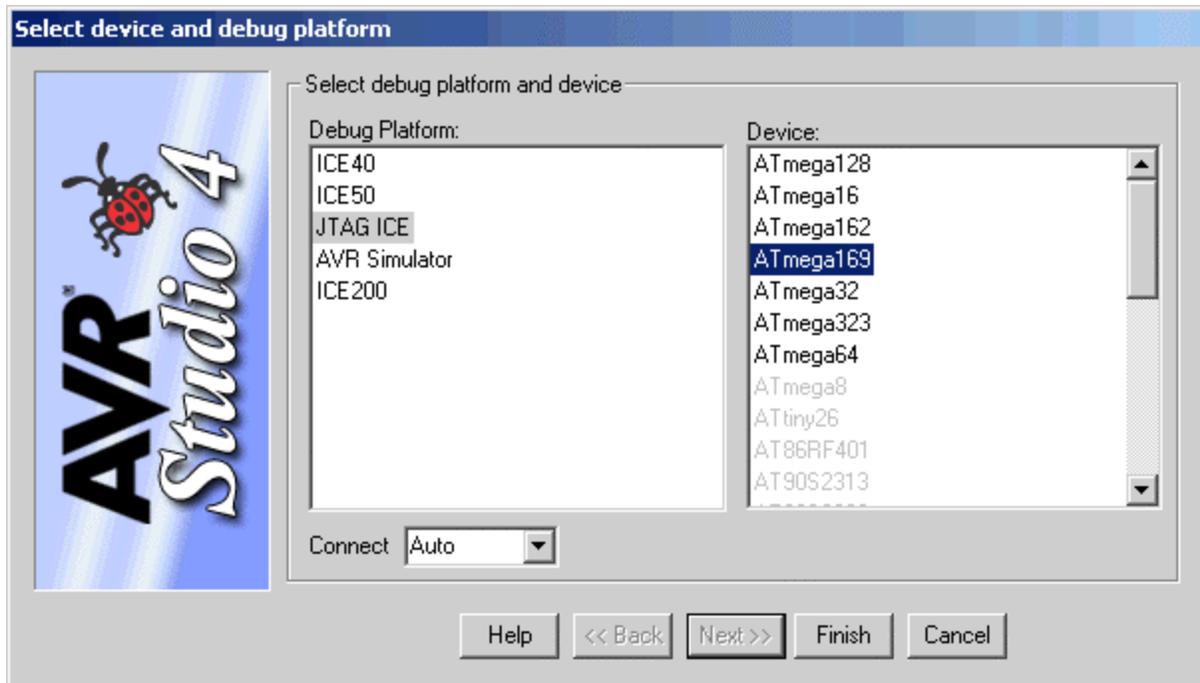
1. The platform selection tells AVR Studio what tool and where to debug project
2. The device selection informs AVR Studio what device is to be debugged

Steps to Perform:
1. Select "JTAG ICE" as "Debug Platform"
2. Select "ATmega169" as Device
3. Press "Finish"
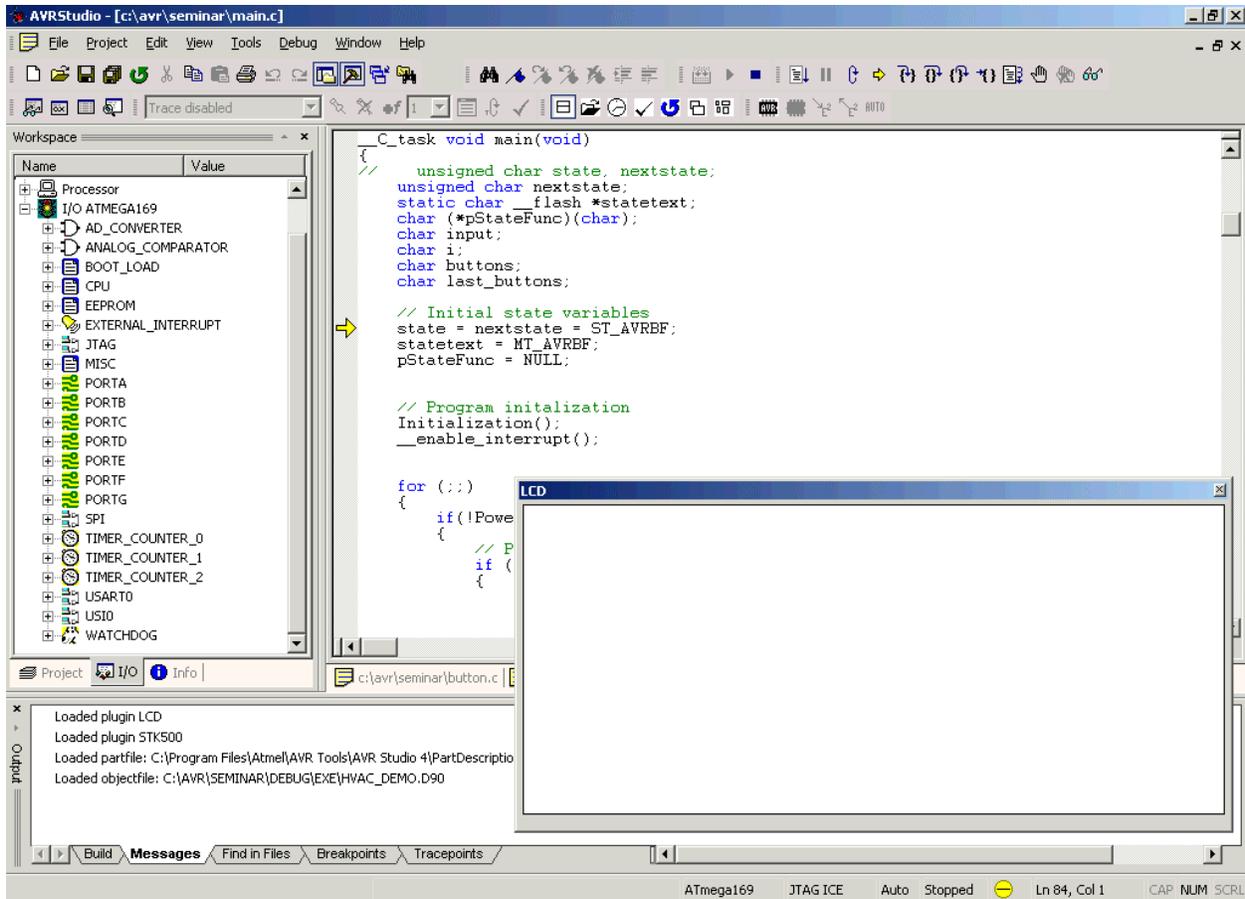
1. The program is now loaded into the ATmega169 device
2. Using the JTAG ICE, all memory and register contents are displayed in AVR Studio
3. After loading or reset, the Program Counter is at the start of the application section

Steps to Perform:
1. To open the LCD window in AVR Studio, simply click the "ADD LCD-window" button in AVR Studio
2. Configure the LCD Plug-in to support the AVR Butterfly LCD,
   a. Select "Open Display-file" from the Tools → LCD menu
   b. Browse and select "502.LCD" found in the "..\STK502" sub-directory of the LCD Plug-in installation
3. Expand the "I/O ATmega169" option in the "Workspace"
4. Set a Breakpoint on Line #137 in the MAIN.C Source File
5. Single Step (F11) and then Run (F5) to the breakpoint. Continue running, and the LCD will update scrolling text

1. Single stepping happens with one 'C' statement at a time
2. After running for a few seconds we can see the data in the LCD Data Registers (LCDDRx under MISC) changing
3. During run, we can also see the LCD on the AVR Butterfly or the LCD window in AVR Studio changing with data.

AVRStudio - [c:\avr\seminar\main.c]

File Project Edit View Tools Debug Window Help

Workspace

Name	Value
Processor	
I/O ATMEGA169	
AD_CONVERTER	
ANALOG_COMPARATOR	
BOOT_LOAD	
CPU	
EEPROM	
EXTERNAL_INTERRUPT	
JTAG	
MISC	
PORTA	
PORTB	
PORTC	
PORTD	
PORTE	
PORTF	
PORTG	
SPI	
TIMER_COUNTER_0	
TIMER_COUNTER_1	
TIMER_COUNTER_2	
USART0	
USI0	
WATCHDOG	

Project I/O Info

```
        {
            state = nextstate;
            for (i=0; menu_state[i].state; i++)
            {
                if (menu_state[i].state == state)
                {
                    statetext =  menu_state[i].pText;
                    pStateFunc = menu_state[i].pFunc;
```

LCD

EMPERA

```
// then activate auto press of the joystick
buttons = (~PINB) & PINB_MASK;
```

c:\avr\seminar\button.c | c:\avr\seminar\lcd_driver.c | c:\avr\seminar\main.c | c:\avr\seminar\rtc.c | c:\avr\seminar\timer0.c

Output

Loaded plugin LCD
Loaded plugin STK500
Loaded partfile: C:\Program Files\Atmel\AVR Tools\AVR Studio 4\PartDescriptionFiles\ATmega169
Loaded objectfile: C:\AVR\SEMINAR\DEBUG\EXE\HVAC_DEMO.D90

Build Messages Find in Files Breakpoints Tracepoints

ATmega169   JTAG ICE   Auto Stopped   Ln 137, Col 1   CAP NUM SCRL

1. The Butterfly is capable of measuring the temperature by using its Negative Temperature Coefficient thermistor
    a. Temperature Ranges: -15°C to 60°C and/or 0°F to 140°F

Steps to Perform:
1. Navigate to the "Temperature" menu,
    a. Press the Joystick on the AVR Butterfly "Up" towards the Atmel logo, and use the "Run" command in AVR Studio until "Temperature" is displayed on the LCD
2. Once at "Temperature",
    a. Press the Joystick on the AVR Butterfly "Left" away from the LCD, and use the "Run" command until the current temperature is displayed
    b. Toggle between Fahrenheit and Celsius by moving the joystick on the Butterfly to either "Up" or "Down" position and using the Studio "Run" command

```
                                {
                                    state = nextstate;
                                    for (i=0; menu_state[i].state; i++)
                                    {
                                        if (menu_state[i].state == state)
                                        {
                                            statetext =  menu_state[i].pText;
                                            pStateFunc = menu_state[i].pFunc;
```

LCD

HVAC

```
                            // then activate auto press of the joystick
                            buttons = (~PINB) & PINB_MASK;
```

Loaded plugin LCD
Loaded plugin STK500
Loaded partfile: C:\Program Files\Atmel\AVR Tools\AVR Studio 4\PartDescriptionFiles\ATmega169
Loaded objectfile: C:\AVR\SEMINAR\DEBUG\EXE\HVAC_DEMO.D90
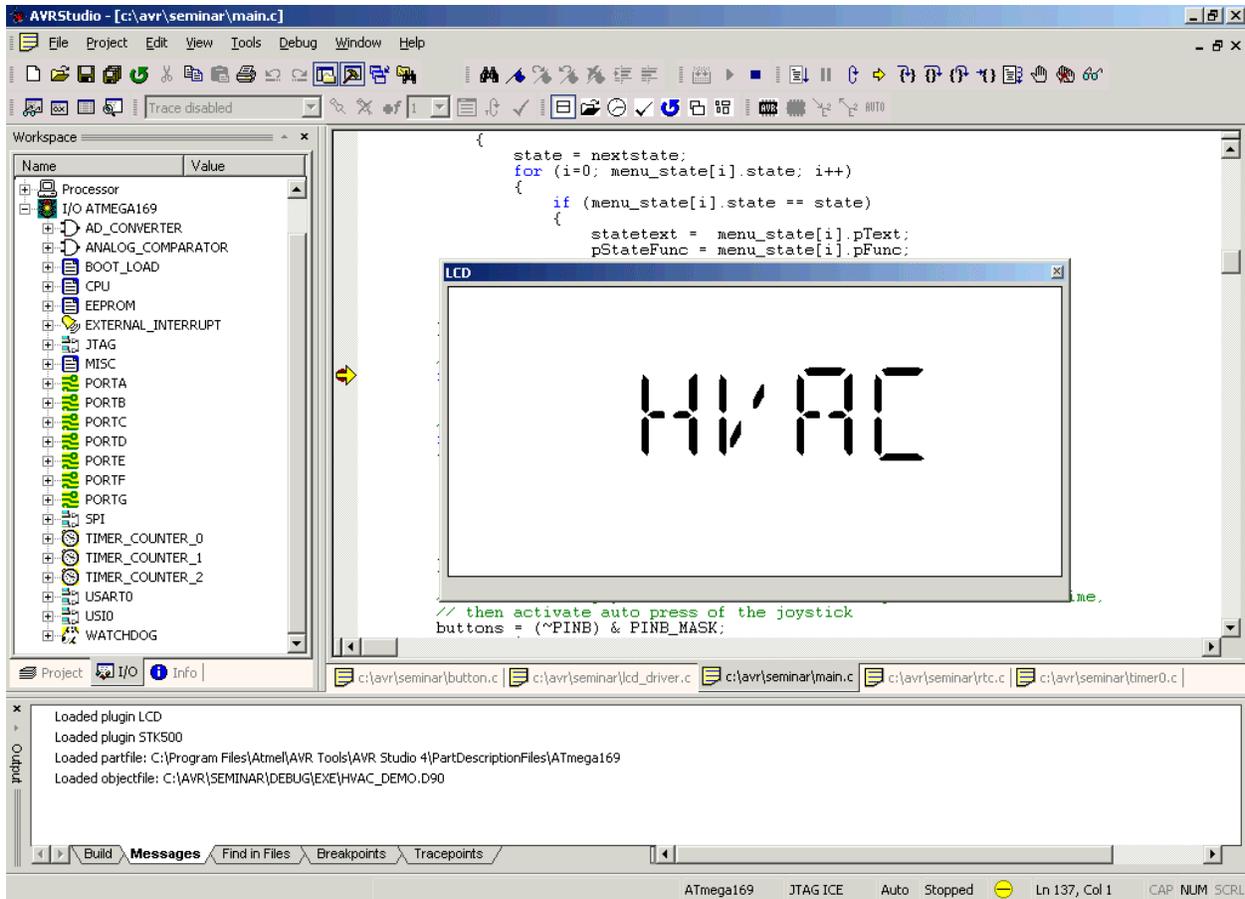
1. Now that the room temperature is known, lets move onto the Heating, Ventilation, and Air Conditioning options

Steps to Perform:
1. Navigate to the "HVAC" menu,
   a. Press the joystick to the "Left" and use the Studio "Run" command to return to the "Temperature" menu
   b. Press the joystick "Up" and use the Studio "Run" command to navigate to the "HVAC" menu
2. Navigate to the "Set Point" option,
   a. Press the joystick to the "Left" and use the Studio "Run" command to access the HVAC options
   b. Press the joystick "Up" and use the Studio "Run" command to navigate to the "Set Point" menu
   c. Press the joystick to the "Left" and use the Studio "Run" command to display the current Set Point
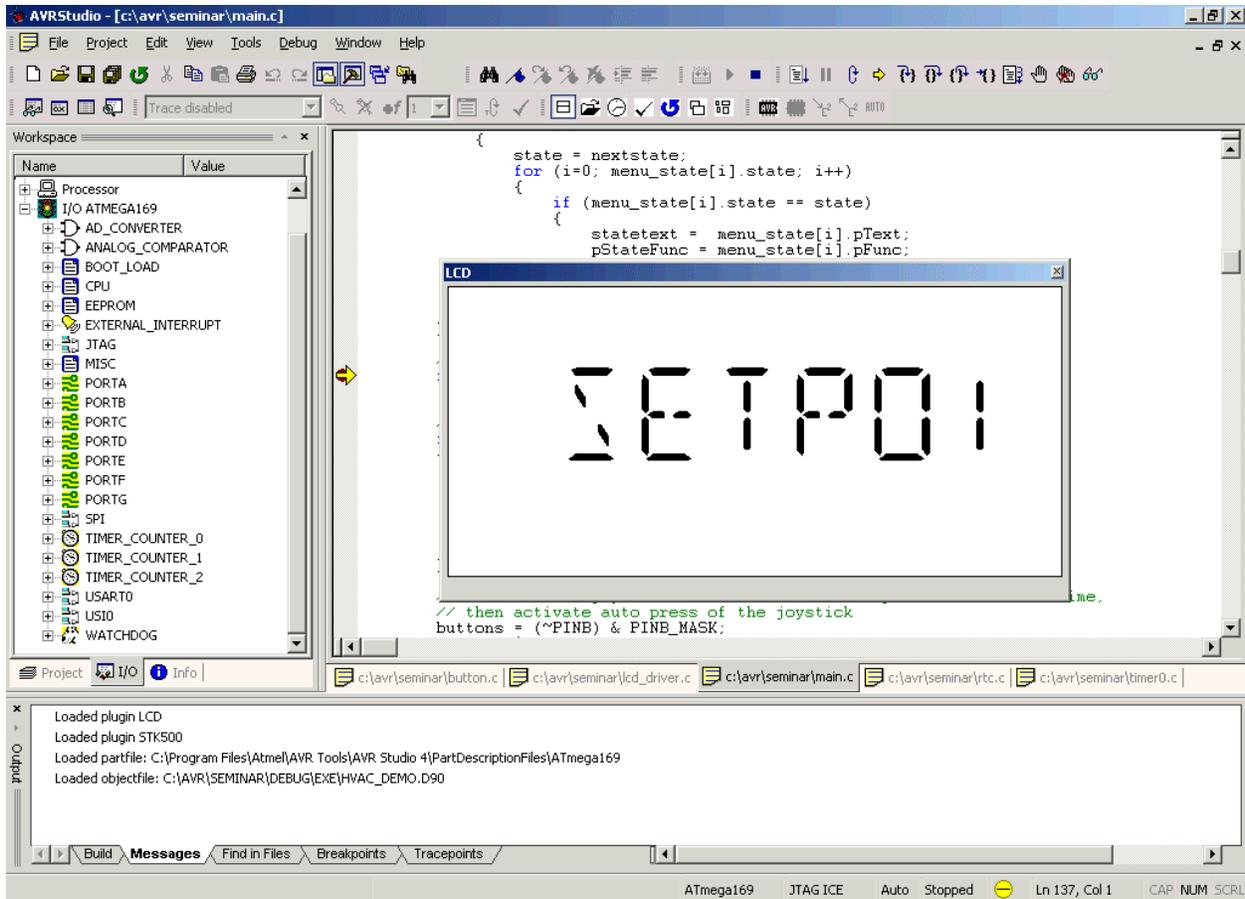
1. First we need to adjust the Set Point such that it is slightly higher then the current room temperature – This will result in the Heater being active when the HVAC system is turned on

Steps to Perform:
1. Adjust the current Set Point,
   a. Press the joystick to the "Right" and use the Studio "Run" command to display the current Set Point
   b. Press the joystick to the "Right, and use the Studio "Run" command to access the "Adjust Set Point" dialog
   c. Press the joystick to the "Right" and use the Studio "Run" command to adjust the Set Point
   d. Press the joystick either "Up" or "Down" to raise or lower the Set Point, adjust the Set Point to a couple of degrees above the room temperature
   e. Push the joystick in to set the Set Point and use the Studio "Run" command, when the Set Point has been adjusted it will return you to the "Set Point" menu

AVRStudio - [c:\avr\seminar\main.c]

File   Project   Edit   View   Tools   Debug   Window   Help

Workspace

Name	Value

- Processor
- I/O ATMEGA169
  - AD_CONVERTER
  - ANALOG_COMPARATOR
  - BOOT_LOAD
  - CPU
  - EEPROM
  - EXTERNAL_INTERRUPT
  - JTAG
  - MISC
  - PORTA
  - PORTB
  - PORTC
  - PORTD
  - PORTE
  - PORTF
  - PORTG
  - SPI
  - TIMER_COUNTER_0
  - TIMER_COUNTER_1
  - TIMER_COUNTER_2
  - USART0
  - USI0
  - WATCHDOG

Project   I/O   Info

```
                        {
        state = nextstate;
        for (i=0; menu_state[i].state; i++)
        {
            if (menu_state[i].state == state)
            {
                statetext =  menu_state[i].pText;
                pStateFunc = menu_state[i].pFunc;
```

LCD

ON/OFF

```
// then activate auto press of the joystick
buttons = (~PINB) & PINB_MASK;
```

c:\avr\seminar\button.c | c:\avr\seminar\lcd_driver.c | c:\avr\seminar\main.c | c:\avr\seminar\rtc.c | c:\avr\seminar\timer0.c

Output

Loaded plugin LCD
Loaded plugin STK500
Loaded partfile: C:\Program Files\Atmel\AVR Tools\AVR Studio 4\PartDescriptionFiles\ATmega169
Loaded objectfile: C:\AVR\SEMINAR\DEBUG\EXE\HVAC_DEMO.D90

Build   Messages   Find in Files   Breakpoints   Tracepoints

ATmega169   JTAG ICE   Auto   Stopped   Ln 137, Col 1   CAP NUM SCRL

1. After the Set Point has been adjusted we can turn on the HVAC System
2. Once the HVAC has been turned on, a digit(s) should be displayed above the text

Steps to Perform:
1. Navigate to the "On/Off" option,
   a. Press the joystick either "Up" or "Down" and use the Studio "Run" command to navigate to the "On/Off" menu
   b. Press the joystick to the "Right" and use the Studio "Run" command to display the current state of the HVAC
   c. Press the joystick either "Up" or "Down" to turn the HVAC "On" and use the Studio "Run" command
   d. Once "On" is displayed, Press the joystick to the "Left" and use the Studio "Run" command to return to the "On/Off" option
   e. Press the joystick to the "Left" and use the Studio "Run" command to return to the "HVAC" menu
   f. Press the joystick to the "Left" and use the Studio "Run" command to return to the initial screen
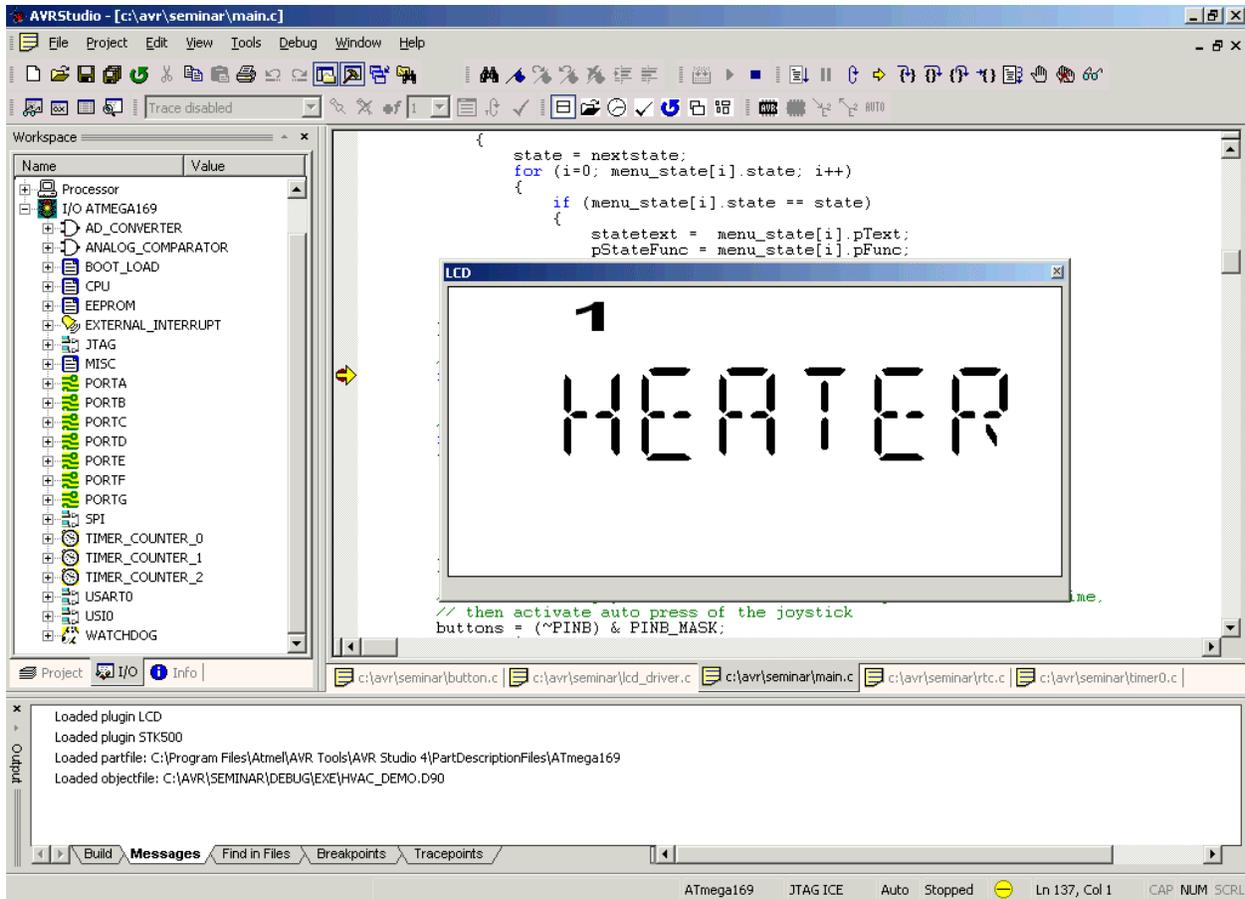
1. Prior to turning on the HVAC the room temperature was … The Set Point is currently set at … The Set Point is … degrees higher than room temperature, thus the heater needs to be turned on
2. The numbers up top display the intensity of the heater/cooler (1…5, Low to High), as the difference between the Set Point and Room Temperature increases, so does the intensity.

Steps to Perform:
1. Place the Butterfly near a heat source (i.e. the fan output of the LCD Projector) and use the Studio "Run" command to update the LCD display
2. In the ShowDisplay function in MAIN.C add the variable "counter" to the Watch Window
   a. Edit "counter" and change the value to 125 (Decimal) for quicker update of the display text. When the temperature becomes higher then the Set Point, the text will change to "Cooler"
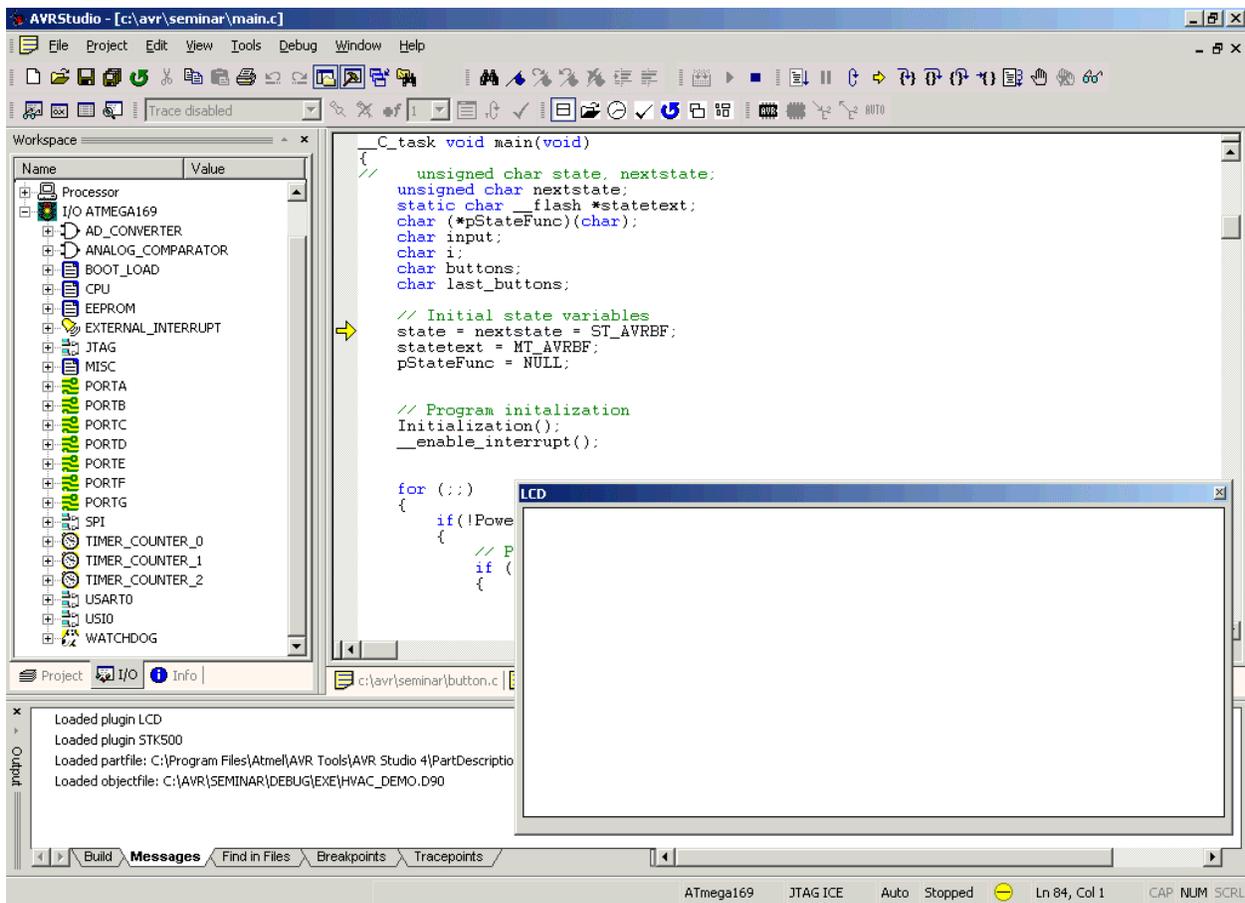
Steps to Perform:

1. Turn off the HVAC System using the Butterfly joystick and the AVR Studio "Run" command
2. Navigate back to the main dialog, "AVR World Tour 2003" should now be displayed
3. If time permits, perform a few of the selected other tasks found at the end of this document

If Time Permits:
1. Re-compile the Source Code with Compression Enabled
   a. A savings of up to 20% can be achieved
2. Adjust the Set Point to Higher/Lower Values
   a. Set Points can Range from –15 to 60°C and 0 to 140°F
   b. View changes in the Intensity of the Heater/Cooler
3. Show the OSCCAL_calibration() function
   a. Self Calibrate the Internal 8 MHz Oscillator using the 32 kHz oscillator
4. Change the contents of the LCDDRx Registers – View the update on the Butterfly and the LCD Plug-in
5. Demonstrate the JTAG ICE as a Programmer