```
/*****************************************
Project : adc_example.c
Version : v1.1 for AVR-GCC
Date    : 10/30/2003
Author  : Chris Troutner
Company : MyRobot
Chip    : ATMEGA128
Platform: Model-T Prototype
Comments: This program was the first successful test program for controlling the onboard ADC. After
            initializing the LCD, it constantly reads ADC3 (PORTF.3) which is the Battery voltage
            reference and displays the battery voltage on the LCD. This function uses the ADC
            interrupt. It also uses AVCC (PIN 64) as the voltage reference.


Clock frequency     : 16.0000 MHz
Memory model        : Small
Internal SRAM size  : 4096
External SRAM size  : 0
Data Stack size     : 1024
*****************************************/

#include <modelt.h>

//Main
    int main(void)
    {

    //Local Variables
        uint16_t    adcval=0;       //Integer used to store returned Analog-to-Digital-Conversion
                                    //(ADC) value.
        float       voltage;        //Floating point variable used to store computed voltage level
                                    //of the batteries.
        char        lcdstr[16];     //Variable to store a string we want to display on the LCD.
                                    //Makeing this a local variable is more efficient than a global
                                    //variable.

    reset();                        //Initialize the microcontroller.

    lcd_init();                     //Initialize the LCD

    ms_spin(100);                   //Wait for 100 mS to allow power to stablize.

    // Global enable interrupts
    asm volatile ("sei");

    while (1)                       //Loop Forever
          {

          adcval = getadc(3);
          voltage = (adcval * ((2*4.98)/1024.0));
          sprintf(lcdstr,"Bat: %10.2fv   ", voltage);
          line1(lcdstr);
          sprintf(lcdstr,"ADC: %11d    ",adcval);
          line2(lcdstr);
```

```
55              ms_spin(2000);
56
57              };
58      }
59
60
61
```