Sample Code: A/D Converter for PIC

by Brandon Cerge
The code below will continuously sample a voltage on channel 4 of the A/D converter and display the 10-bit result in a 16-bit format (i.e., 0x0000 – 0x03FF) on the LCD screen. The LCD screen is hooked up to PORTA of the PIC (see UF_LCD Tutorial) and channel4 of the A/D is also on PORTA. Because the way the PIC18F4620 is set up, we must switch back and forth from digital I/O for LCD screen use and analog input for A/D use. See PIC datasheet for register information.

```c
#include <p18f4620.h>
#include <UF_LCD.h>

#pragma config OSC = INTIO67       // select internal oscillator
#pragma config WDT = OFF          // turn off watch dog timer
#pragma config LVP = OFF          // turn off low voltage program

void main(void){
    int adc_result;              // where we will store the ADC result in
    OSCCON = 0x76;               // set internal oscillator at 8MHz
    ADCON1 = 0x0F;               // PORTA for I/O
    lcd_init();                  // get the LCD up and running
    ADCON0 = 0x11;               // select channel4 and enable A/D
    ADCON2 = 0x88;               // select internal clock and acquisition time
    lcd_char('0');               // only need to write this once
    lcd_char('x');               // only need to write this once

    while (42){                  // infinite loop (with meaning)
        ADCON1 = 0x0A;            // PORTA for A/D use
        ADCON0 |= 0x02;           // set the Go bit
        while (ADCON0bits.GO == 1); // wait until the Go bit is set to 0
        adc_result = ADRES;      // grab our 10-bit answer
        ADCON1 = 0x0F;            // PORTA back to I/O for LCD use
        lcd_byte((adc_result >> 8) & 0x03); // write upper byte
        lcd_byte(adc_result & 0xFF); // write lower byte
        lcd_command(0x82);        // place cursor back at position 2
    }
}
```

*note: lcd_byte(char) is a function that must be written by the programmer!!*