

```

/**
 * File: Lesson_3_Ex1_outline.c
 *
 * This program uses the AVR on-chip ADC to measure the voltage on an input
 * pin (PC3) wired to a potentiometer providing a 0..+5V signal source.
 *
 * This is just an outline. You need to fill in the gaps as annotated.
 */
#include <avr/io.h>
#define F_CPU 16000000UL // CPU runs at 16 MHz

// These header files contain definitions needed by library functions:
#include <util/delay.h>
#include <stdlib.h>
#include <string.h>
#include "lib_avrXmini.h" // def's for lib_avrXmini.a

int main(void)
{
    char strNum[20]; // number converted to string
    int adc_count; // Raw ADC conversion count (0..1023)
    int reading_mV; // Voltage reading (0..5000 mV)
    int last_reading; // previous ADC reading (last read)

    lcd_initialise(); // Initialize LCD module
    lcd_command(LCD_CLR); // Send command to clear LCD
    lcd_command(LCD_CURSOR_OFF); // Set Display ON, Cursor OFF

    // Configure I/O pin PC3 (= ADC3) as analogue input:
    ;

    lcd_cursor_posn(0, 0);
    lcd_print_string("ADC count: ");
    lcd_cursor_posn(1, 0);
    lcd_print_string("Voltage: ");

    while (TRUE) // loop forever
    {
        adc_count = ADC_ReadInput(3); // Read pot input ADC3/PC3

        if (adc_count != last_reading) // if reading has changed...
        {
            // Clear displayed data
            ;
            ;

            // Convert raw ADC reading (adc_count) to milliVolts (reading_mV)
            ;

            // Convert adc_count to string of decimal digits in array strNum
            ;

            // Display adc_count on top line of LCD
            ;
            ;

            // Convert reading_mV to string of decimal digits in array strNum
            ;

            // Display milliVolts on bottom line of LCD
            ;
            ;
            lcd_print_string(" mV");
        }

        last_reading = adc_count;
    }
}

```

```
        _delay_ms(50);  
    }  
}  
  
// end of file
```