**AVR Analog**

**Example Code**

To learn about LCD interfacing, view [this](http://maxembedded.wordpress.com/2011/06/16/lcd-interfacing-with-avr/) post. You can type, compile and build it in AVR Studio 5. View [this](http://maxembedded.wordpress.com/2011/06/12/using-avr-studio-5/) page to know how. To know about the I/O port operations in AVR, view [this](http://maxembedded.wordpress.com/2011/06/10/port-operations-in-avr/) page.

[?](http://maxembedded.com/2011/06/the-adc-of-the-avr/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79 | #include <avr/io.h>  #include <util/delay.h>    #include "lcd.h"    #define LTHRES 500  #define RTHRES 500    // initialize adc  void adc\_init()  {      // AREF = AVcc      ADMUX = (1<<REFS0);        // ADC Enable and prescaler of 128      // 16000000/128 = 125000      ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);  }    // read adc value  uint16\_t adc\_read(uint8\_t ch)  {      // select the corresponding channel 0~7      // ANDing with '7' will always keep the value      // of 'ch' between 0 and 7      ch &= 0b00000111;  // AND operation with 7      ADMUX = (ADMUX & 0xF8)|ch;     // clears the bottom 3 bits before ORing        // start single conversion      // write '1' to ADSC      ADCSRA |= (1<<ADSC);        // wait for conversion to complete      // ADSC becomes '0' again      // till then, run loop continuously      while(ADCSRA & (1<<ADSC));        return (ADC);  }    int main()  {      uint16\_t adc\_result0, adc\_result1;      char int\_buffer[10];      DDRC = 0x01;           // to connect led to PC0        // initialize adc and lcd      adc\_init();      lcd\_init(LCD\_DISP\_ON\_CURSOR);        // display the labels on LCD      lcd\_puts("left  ADC = ");      lcd\_gotoxy(0,1);      lcd\_puts("right ADC = ");        \_delay\_ms(50);        while(1)      {          adc\_result0 = adc\_read(0);      // read adc value at PA0          adc\_result1 = adc\_read(1);      // read adc value at PA1            // condition for led to glow          if (adc\_result0 < LTHRES && adc\_result1 < RTHRES)              PORTC = 0x01;          else              PORTC = 0x00;            // now display on lcd          itoa(adc\_result0, int\_buffer, 10);          lcd\_gotoxy(12,0);          lcd\_puts(int\_buffer);            itoa(adc\_result1, int\_buffer, 10);          lcd\_gotoxy(12,1);          lcd\_puts(int\_buffer);          \_delay\_ms(50);      }  } |