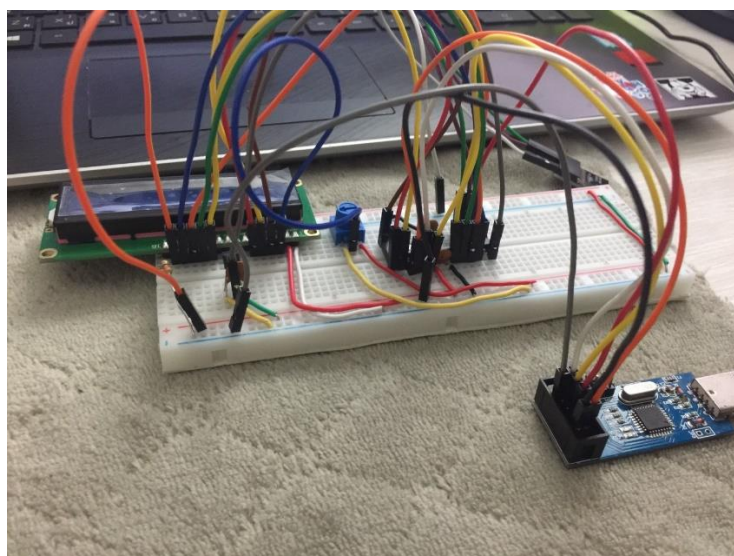
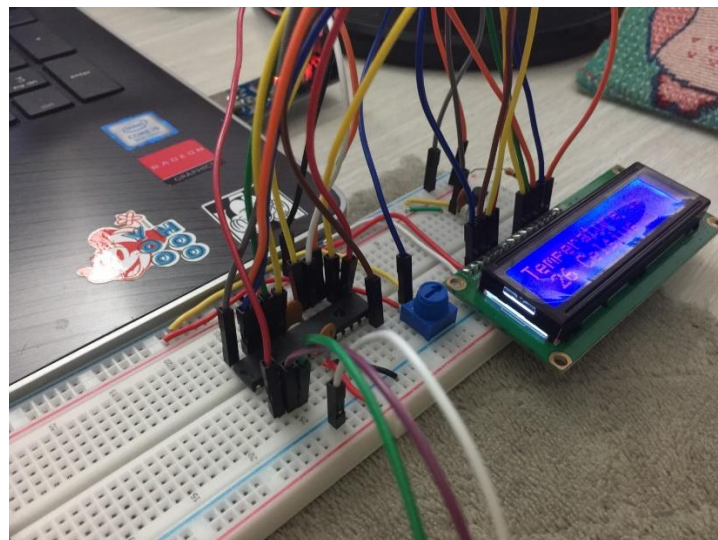
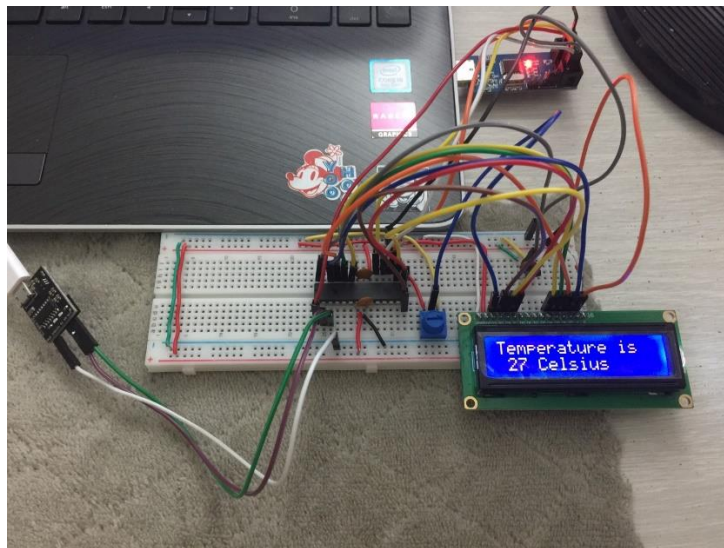


## Lab5 : AVR ADC Programming

### Circuit for lab5



## Code for lab5

```
#define F_CPU 8000000UL

#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include <string.h>

//define data and cmd port, RS, RW, E ports
#define LCD_DATA_DDR DDRC
#define LCD_DATA_PORT PORTC
#define LCD_CMD_DDR DDRB
#define LCD_CMD_PORT PORTB
#define RS PORTB0
#define E PORTB1
#define RW PORTB2

void USART_Init(unsigned int ubrr) {
    /* Set baud rate */
    UBRR0 = ubrr;
    /* Enable receiver and transmitter */
    UCSR0B |= (1 << RXEN0) | (1 << TXEN0);
    /* Set frame format: 8data*/
    UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00);
}

void USART_Transmit( unsigned char data ) {
    /* Wait for empty transmit buffer */
    while ( !( UCSR0A & (1 << UDRE0)) );
    /* Put data into buffer, sends the data */
    UDR0 = data;
}

void commit_data() {
    //pulse enable
    //enable high
    LCD_CMD_PORT |= (1 << E);
    _delay_us(1000);
    //enable low
    LCD_CMD_PORT &= ~(1 << E);
    _delay_us(1000);
}

void send_data(uint8_t data) {
    //send bit 7:4
    LCD_DATA_PORT = (data >> 4);
    commit_data();
}
```

```
//send bit 3:0
LCD_DATA_PORT = (data & 0x0F);
commit_data();
}

void send_lcd_command(uint8_t command) {
    //set to cmd mode, write mode
    LCD_CMD_PORT &= ~(1 << RS);

    //send command
    send_data(command);
}

void send_lcd_data(uint8_t data) {
    //set to data mode, write mode
    LCD_CMD_PORT |= (1 << RS);

    //send data
    send_data(data);
}

//LCD init, call once
void lcd_init() {
    LCD_CMD_DDR |= (1 << RS) | (1 << E) | (1 << RW );
    LCD_DATA_DDR = 0x0F;
    LCD_CMD_PORT &= ~(1 << RS) | (1 << E) | (1 << RW );
    LCD_DATA_PORT = 0x00;

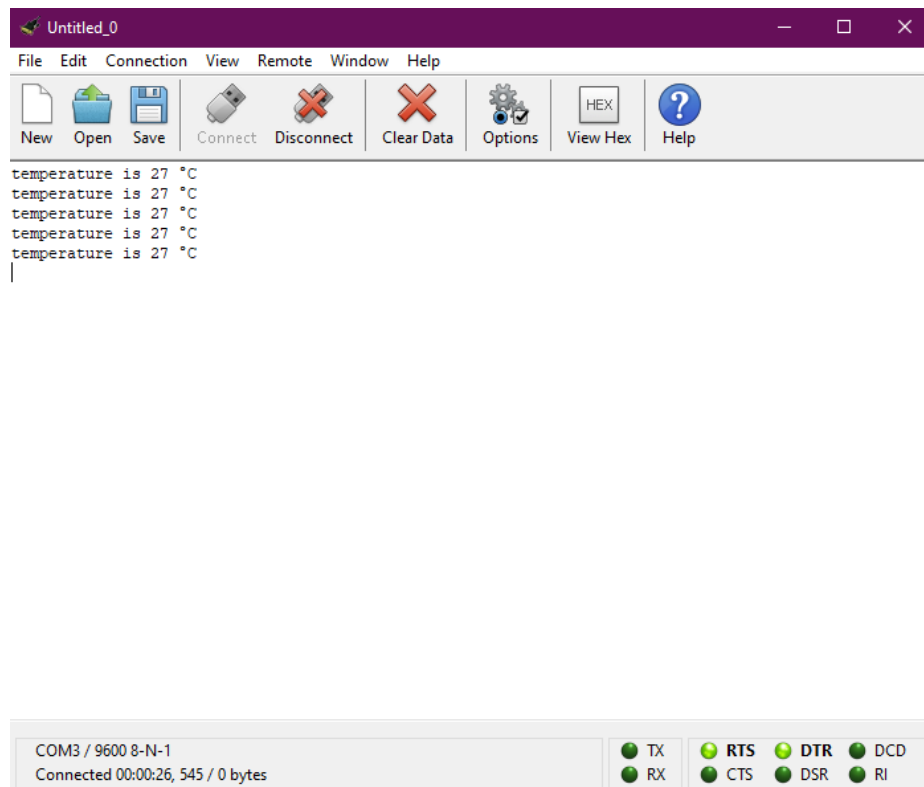
    send_lcd_command(0x03);    //4-bit mode
    send_lcd_command(0x02);

    send_lcd_command(0x28);    //4-bit comm, 2 lines, 5x8 font
    send_lcd_command(0x0C);    //display ON, cursor OFF, blink OFF
    send_lcd_command(0x01);    //clear screen
    send_lcd_command(0x80);    //cursor go to top left corner
    _delay_ms(1);
}

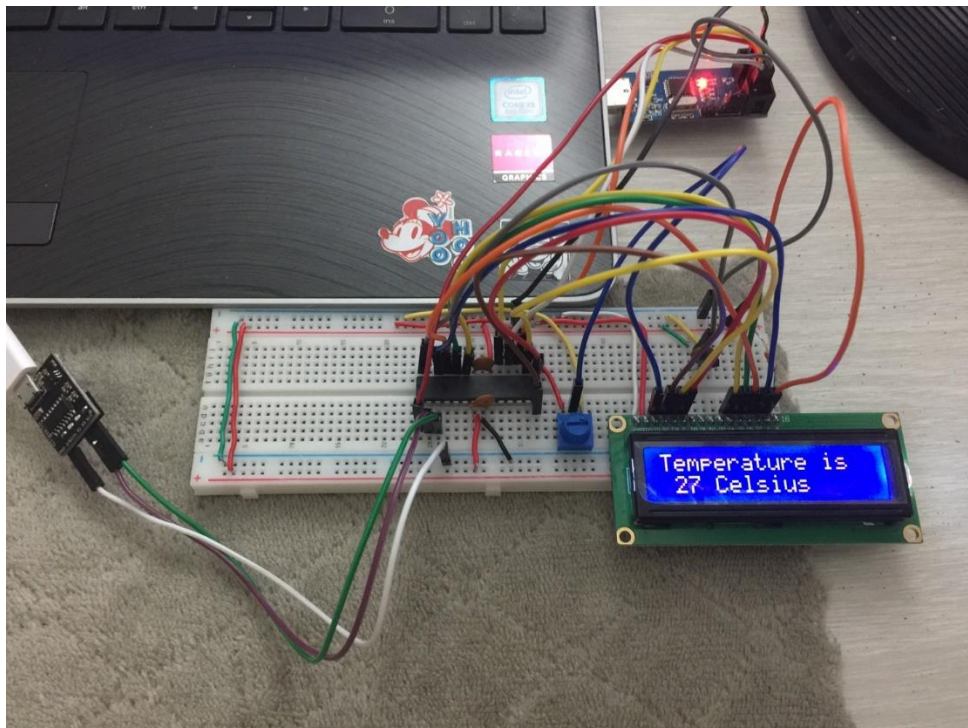
void adc_init() {
    //set input channel, set AREF
    ADMUX = 0x05; // input pin ADC5
    ADMUX |= (1 << REFS0) | (1 << MUX2) | (1 << MUX0); // set AREF
    //enable ADC, set prescaler so the output freq is around 50-200kHz
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (0 << ADPS0); // divide by 64
    //disable digital buffer of that ADC pin
    DIDR0 |= (1 << ADC5D);
}
```

```
int main(void) {
    lcd_init();
    adc_init();
    USART_Init(51);
    char str[40] = "Temperature is ", temp_char[40];
    int temp, i;
    while (1) {
        //start conversion
        ADCSRA |= (1 << ADSC);
        //wait for ADIF
        while(!(ADCSRA & (1 << ADIF)));
        //copy data out
        uint16_t data = ADC;
        //reset ADIF flag by writing 1
        ADCSRA |= (1 << ADIF);
        temp = (((data / 1024.0) * 5) - 0.5) * 100;
        //send data to serial
        sprintf(temp_char, "temperature is %d °C \n", temp);
        for(i = 0; i < strlen(temp_char); ++i){
            USART_Transmit(temp_char[i]);
        }
        //send data to lcd
        send_lcd_command(0x80);
        for(i = 0; i < strlen(str); ++i){
            send_lcd_data(str[i]);
        }
        sprintf(temp_char, " %d Celsius ", temp);
        send_lcd_command(0xC0);
        for(i = 0; i < strlen(temp_char); ++i){
            send_lcd_data(temp_char[i]);
        }
        _delay_ms(1000);
    }
}
```

## Result for lab5



รูปที่ 1 : Result in Coolterm



รูปที่ 2 : Result on LCD