

# Design Assignment X

---

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
1	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
2.	INITIAL CODE OF TASK 1/A		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
4.	SCHEMATICS		
5.	SCREENSHOTS OF EACH TASK OUTPUT		
5.	SCREENSHOT OF EACH DEMO		
6.	VIDEO LINKS OF EACH DEMO		
7.	GOOGLECODE LINK OF THE DA		

## 1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Components used:

- Atmega328P
- Lm34 temperature sensor
- Sparkfun FTDI Basic Breakout
- Lots of cable jumpers

Schematic is in Number 3

## 2. INITIAL/DEVELOPED CODE OF TASK 1/A

```
#define F_CPU 8000000
#define BAUD 9600
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include <avr/interrupt.h>

volatile int count;

void init_uart(){
    // setting the baud rate based on F_CPU and baudrate
    UBRR0H = 0x00;
    UBRR0L = 0x0C;
    // enabling TX & RX
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);           // enable receive and transmit
    UCSR0A = (1<<UDRE0)|(1<<U2X0);
    UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);    // Set frame: 8data, 1 stop
}

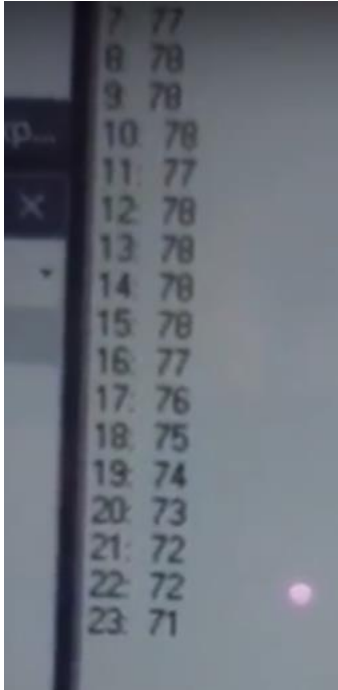
void ADC_init() {
    ADMUX = 0;                                // read from port ADC0
    ADMUX |= (1<<REFS0);                       // use AVcc for reference
    ADCSRA |= (1<<ADPS2) | (1<<ADPS1);          // prescaler of 64
    ADCSRA |= (1<<ADEN);                       // enable ADC
    ADCSRB = 0;                                // free running mode
}

void USART_Transmit( char *data)
{
    while((*data != '\0')) { // transmits all chars but null
        while(!(UCSR0A & (1<<UDRE0))); // waits for transmit flag to clear
        UDR0 = *data;                // transmit next char
        data++;                      // move to next char
    }
}

unsigned int readADC()
{
    ADMUX &= ~(1<<ADLAR);                // clear the adc value
    unsigned int val = 0;
    ADCSRA |= (1 << ADSC);                // start adc
    while(ADCSRA & (1<<ADSC));            // wait until adc is done
}
```



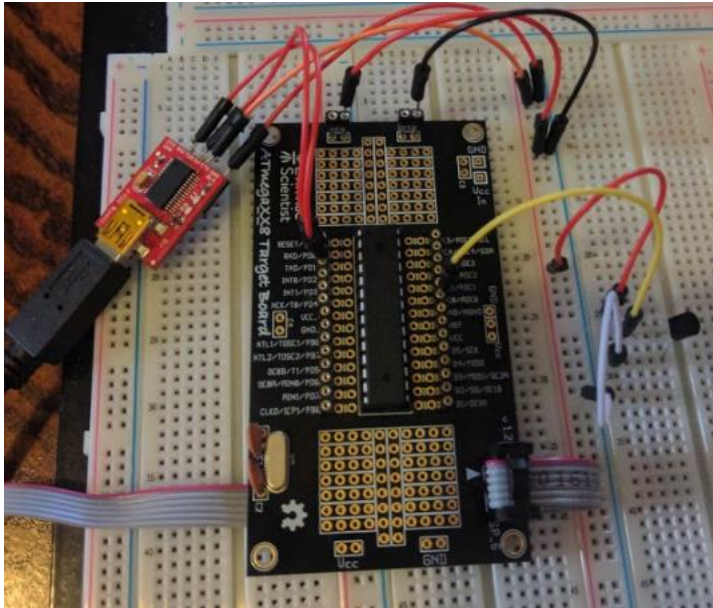
#### 4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



7	77
8	78
9	78
10	78
11	77
12	78
13	78
14	78
15	78
16	77
17	76
18	75
19	74
20	73
21	72
22	72
23	71

Format is <seconds> : Temperature. This screenshot shows it going down right after I placed a bag of ice water on top of the lm34 sensor.

#### 5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



#### 6. VIDEO LINKS OF EACH DEMO

Video showing the circuit working:

<https://www.youtube.com/watch?v=uOaTSgg3JgQ>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

*"This assignment submission is my own, original work".*

Brian Lopez