CPE301 - SPRING 2018

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
- Sparksfun FTDI Basic Breakout
- Lots of cable jumpers

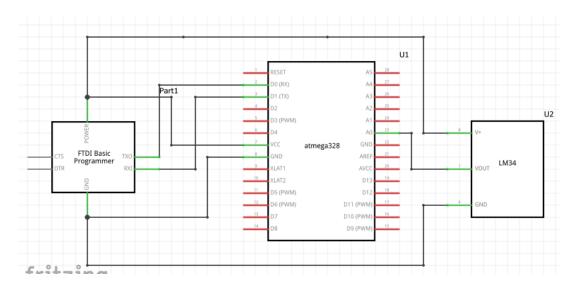
Schematic is in Number 3

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 FCPU and baudrate
   UBRR0H =0x00;
   UBRR0L =0x0C;
   // enabling TX & RX
   UCSROB = (1 << RXENO) | (1 << TXENO);
                                         // enable receive and transmit
    UCSR0A = (1 < < UDRE0) | (1 < < U2X0);
   UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);  // Set frame: 8data, 1 stop</pre>
}
void ADC init() {
   ADMUX = 0;
                             // read from port ADC0
                        // use AVcc for reference
   ADMUX = (1 < REFS0);
   ADCSRA |= (1<<ADPS2) | (1<<ADPS1); // prescalar 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</pre>
          UDR0 = *data;  // transmit next char
          data++;
                                  // move to next char
   }
}
unsigned int readADC()
   ADMUX &= \sim(1<<ADLAR);
                                     // clear the adc value
   unsigned int val = 0;
   ADCSRA = (1 << ADSC);
                                    // start adc
   while(ADCSRA & (1<<ADSC));  // wait until adc is done</pre>
```

```
val = ADC;
   val = val * 0.427;
                        // doing (5 * 100 * adc) / 1024, just simplified
   return val;
}
ISR (TIMER1 COMPA vect)
   int temp = 0;
   char out[20];
                                 // will have the string
                                  // increment counter
   count++;
                                  // get adc value
   temp = readADC();
   snprintf(out, sizeof(out), "Time(sec): %d\r\n", count);
   USART_Transmit(out);
   snprintf(out, sizeof(out), "%3d F\r\n", temp); // put string into out
   USART_Transmit(out);  // send string to be transmitted
TIFR1 |= (1<<OCF1A);  // clear flag</pre>
   TIFR1 |= (1<<0CF1A);
                                 // clear flag
}
int main (){
   count = 0;
                                 // start count at 0
                                 // initialize UART
   init_uart();
   ADC_init();
                                 // initialize ADC
   _delay_ms(250);
                                 // delay to wait for everything to catch up
   OCR1A = 3125;
                                  // set compare value
   TCCR1B |= (1 << WGM12);
                                 // set CTC mode
   TIMSK1 |= (1 << OCIE1A);
                                 // enable timer interupt
   TCCR1B |= (1<<CS12);
                                 // prescalar of 256
   sei();
                        // enable interrupts
   while(1)
          // do nothing
   return 0;
```

3. SCHEMATICS

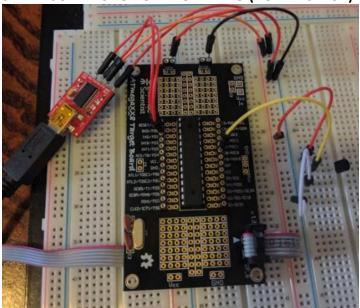


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



Format is <seconds>: Temperature. This screenshot shows it going down right after I placed a bag of ice water on top of the Im34 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