Elizabeth Heider

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

- Atmega 328 P Microcontroller
- LM34 Temperature Sensor
- FTDI Chip
- Breadboard
- 5V Power Supply
- PuTTY

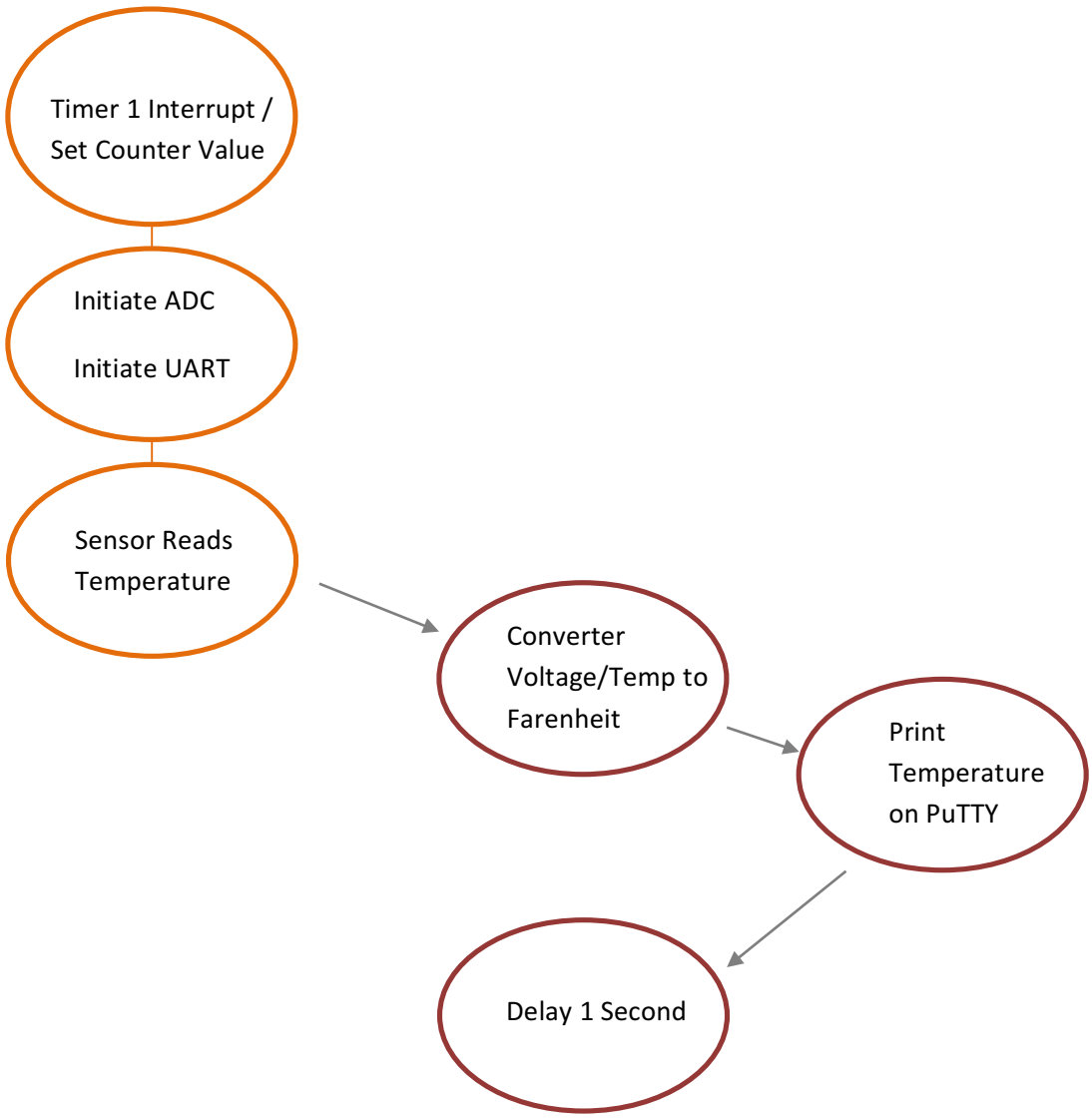
-

2. INITIAL/DEVELOPED CODE OF TASK 1/A

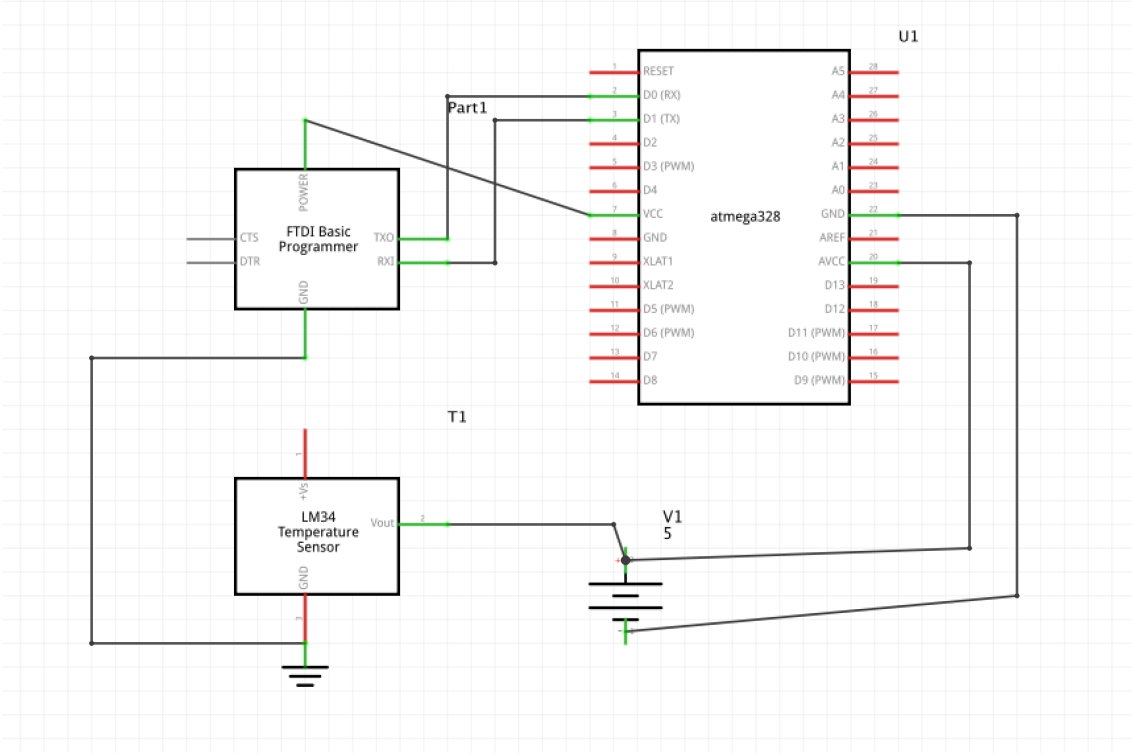
```
#define F_CPU 16000000UL //16 Mhz.
#define BAUD 9600 //Set Baud rate 9600.
#define MYUBRR F_CPU/16/BAUD-1

#include      <avr/io.h>
#include      <util/delay.h>
#include      <stdio.h>
void   read_adc(void); //   Function      Declarations
void   adc_init(void);
void   USART_init(   unsigned      int ubrr );
void   USART_tx_string(     char   *data  );
volatile      unsigned      int adc_temp;
char   outs[20];

ISR(TIMER1_OVF_vect)
{
       read_adc(); // call ADC function
       snprintf(outs,sizeof(outs),”%3d\r\n”, adc_temp); // print output
       USART_tx_string(outs);// output string
       _delay_ms(1000);
       TCNT1 = 49911; // reset timer counter
}

int main(void)       {
       adc_init(); //       Initialize    the    ADC    (Analog       /      Digital
       Converter)
       USART_init(UBRR_9600); //   Initialize    the    USART  (RS232 interface)
       USART_tx_string("Connected!\r\n"); //     we're  alive!
       _delay_ms(125); //   wait   a      bit
       while(1)
       {
              read_adc();
              // snprintf(outs,sizeof(outs),"%3d\r\n",  adc_temp);//  print  it
              //USART_tx_string(outs);
              _delay_ms(1000); //  wait   a      bit
       }
}
void   adc_init(void)
{
       /**    Setup  and    enable ADC    **/
       ADMUX  =      (0<<REFS1)| //       Reference     Selection     Bits
       (1<<REFS0)| //       AVcc - external      cap    at     AREF
       (0<<ADLAR)| //       ADC    Left   Adjust Result
       (0<<MUX2)| // ANalog Channel       Selection     Bits
       (1<<MUX1)| // ADC2   (PC2   PIN25)
       (0<<MUX0);
       ADCSRA =      (1<<ADEN)| // ADC    ENable
       (0<<ADSC)| // ADC    Start  Conversion
       (0<<ADATE)| //       ADC    Auto   Trigger       Enable
       (0<<ADIF)| // ADC    Interrupt     Flag
       (0<<ADIE)| // ADC    Interrupt     Enable
       (1<<ADPS2)| //       ADC    Prescaler Select     Bits
```

```
        (0<<ADPS1)|
        (1<<ADPS0);
        //      Timer/Counter1        Interrupt     Mask    Register
        TIMSK1 |=      (1<<TOIE1); //       enable overflow      interrupt
        TCCR1B |=      (1<<CS11)|(1<<CS10);        //      native clock
        TCNT1 = 49911;
}
/*      READ    ADC     PINS    */
void    read_adc(void)
{
        unsigned        char    i =4;
        adc_temp =      0;
        while   (i--)   {
                ADCSRA |=      (1<<ADSC);
                while(ADCSRA   &       (1<<ADSC));
                adc_temp+=     ADC;
                _delay_ms(50);
        }
        adc_temp =      adc_temp /      4; //  Average       a       few     samples
        int tempf= (adc_temp*5*100)/1024;
        snprintf(outs,sizeof(outs),"%3d\r\n",     tempf);//      print   it
        USART_tx_string(outs);
}

void    USART_init(    unsigned       int ubrr )     {
        UBRR0H =       (unsigned      char)(ubrr>>8);
        UBRR0L =       (unsigned      char)ubrr;
        UCSR0B =       (1      <<     TXEN0); //     Enable receiver,      transmitter    &
        RX      interrupt
        UCSR0C =       (3      <<     UCSZ00); //asynchronous      8       N       1
}
/*      SEND    A       STRING TO      THE     RS-232 */

void    USART_tx_string(      char    *data   )       {
        while   ((*data        !=      '\0')) {
                while   (!(UCSR0A      &       (1      <<UDRE0)));
                UDR0    =      *data;
                data++;
        }
}
```

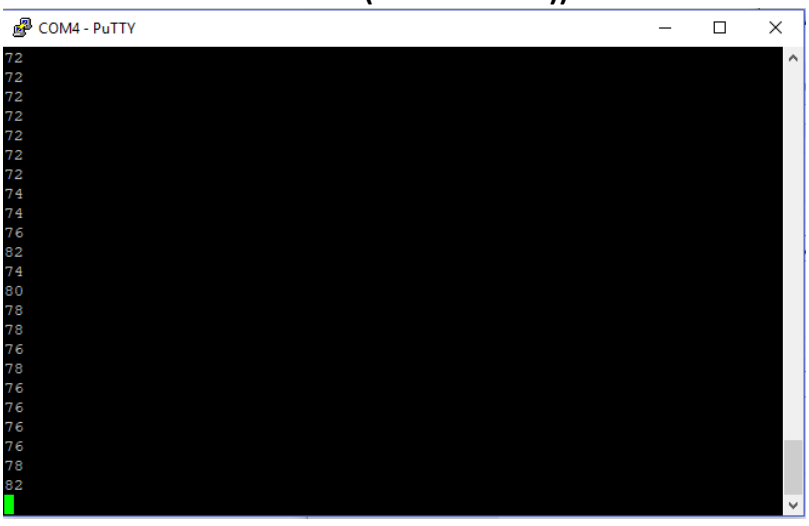
3. Flowchart of Code



4. SCHEMATICS



5. SCREENSHOTS OF OUTPUT (USING PuTTY))



Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

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