CPE301 – SPRING 2019

MIDTERM 1

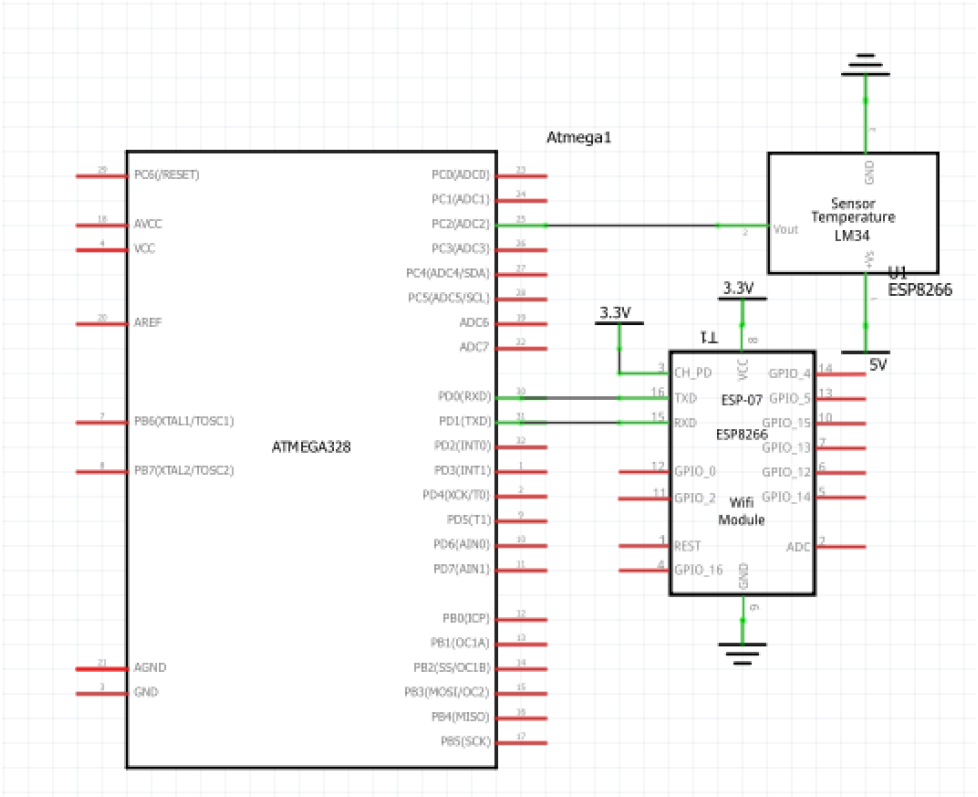
Student Name: Shaquille Regis
Student #: 2000686590
Student Email: regis@unlv.nevada.edu
Primary Github address: https://github.com/regis-shaquille/submissions-SR
Directory: https://github.com/regis-shaquille/submissions-SR/tree/master/Midterms/
Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/Midterm, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used

- ESP8266-01 Module Chip
- ATMega328P Xplained Mini
- LM35 Temperature Sensor
- Breadboard

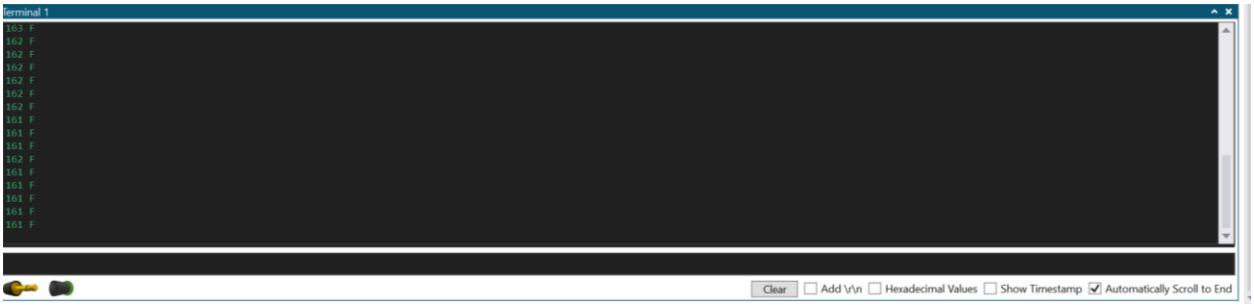


-

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

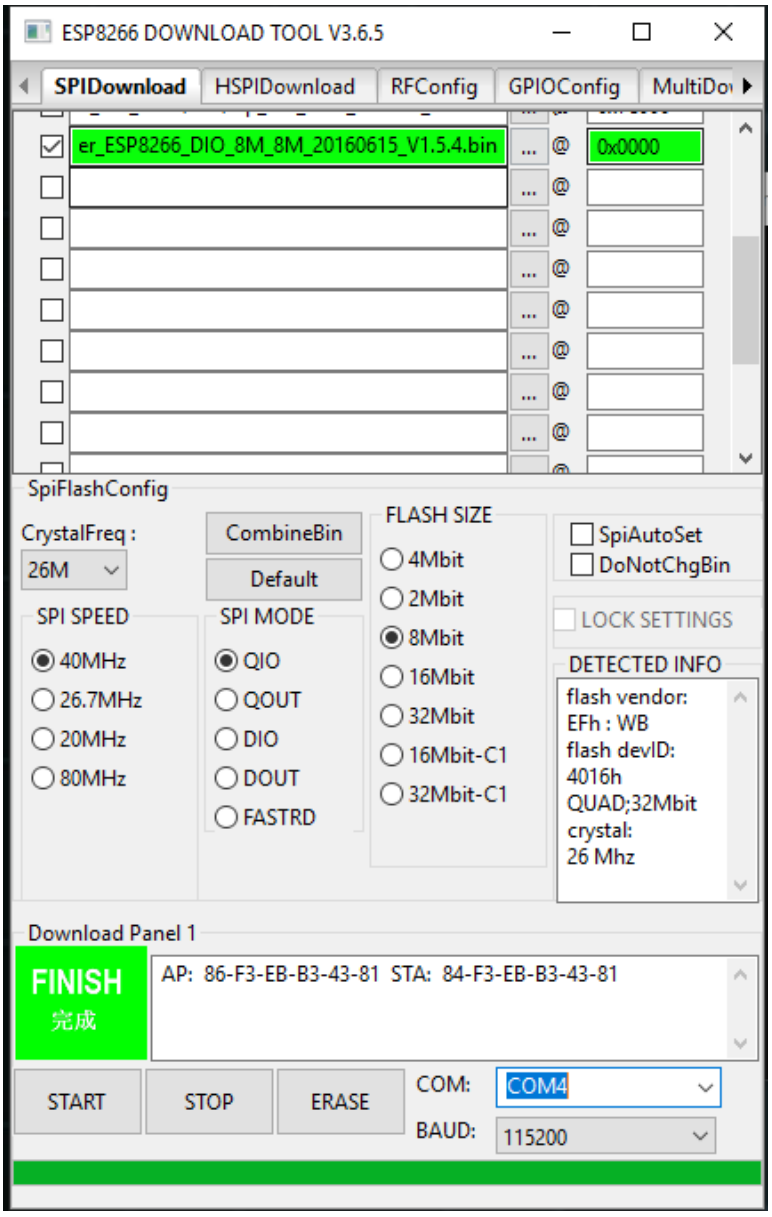
```
void read_adc(void) {
    unsigned char i =4;
    adc_temp = 0; //initialize
    while (i--) {
        ADCSRA |= (1<<ADSC);
        while(ADCSRA & (1<<ADSC));
        adc_temp+= ADC;
        _delay_ms(50);
    }
    adc_temp = adc_temp / 4; // Average a few samples

}

 /* INIT USART (RS-232) */
void USART_init( unsigned int ubrr ) {
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // Enable receiver, transmitter & RX interrupt
    UCSR0C |= (1<<UCSZ01) | (1 << UCSZ00);
}
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A



The Above COM Terminal output shows the temperature readings from the LM35 sensor.

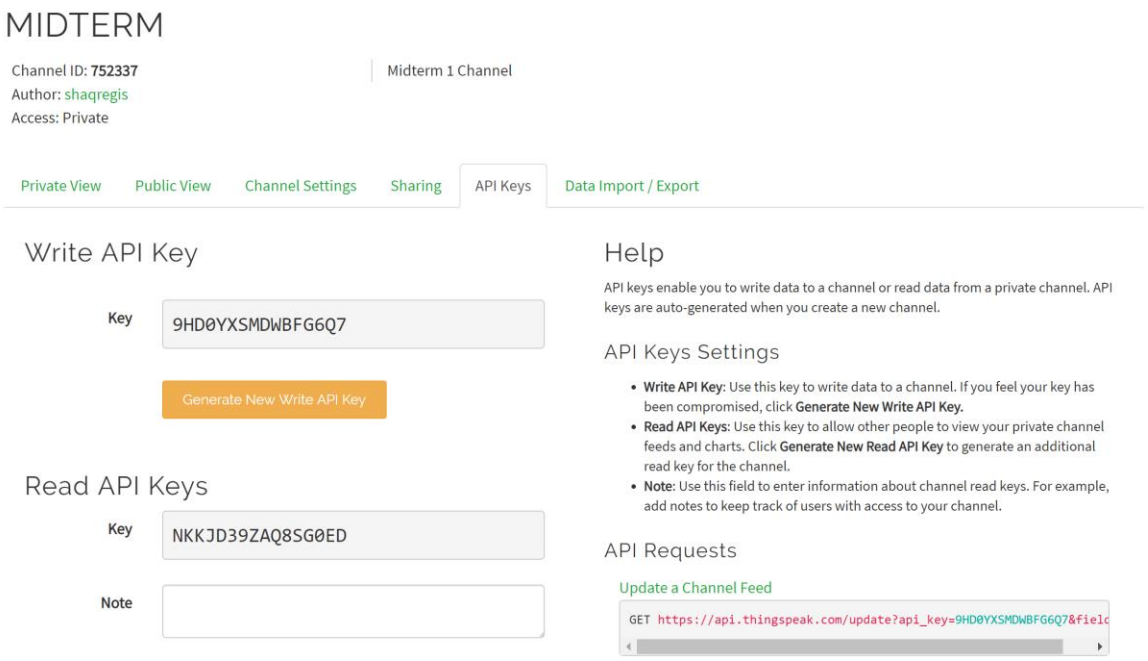
4. FIRMWARE FOR ESP8266



```
AT+GMR
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
Ai-Thinker Technology Co. Ltd.
Jun 13 2016 11:29:20
OK
```

The ESP8622 was flashed with the firmware as shown from the demonstration videos. The firmware was verified using ESPlorer.

5. THINGSPEAK ACCOUNT



6. SEND DATA TO ESP8266 WITH AT COMMANDS

To verify proper operation of the ESP8266, I sent various AT commands using ESPlorer. Shown below, I was able to set the mode of operation, search for networks and connect to my home WIFI.

```
OK
AT+CWMODE=?
+CWMODE:(1-3)

OK
AT+CWMODE=1

OK
AT+CWJAP?
No AP

OK
AT+CWLAP
+CWLAP:(4,"DVW326.EC8FF0-2.4G",-79,"34:68:95:ec:8f:f0",1,-7,0)
+CWLAP:(4,"HotBitches143",-91,"9c:1e:95:67:6f:b5",1,-9,0)
+CWLAP:(3,"ComoLaFlor",-84,"70:3a:cb:a8:2a:b8",1,16,0)
+CWLAP:(4,"D2F06A",-87,"10:0d:7f:d2:f0:6a",1,-9,0)
+CWLAP:(3,"KX-HNB700_DAA655",-39,"bc:c3:42:da:a6:55",4,-21,0)
+CWLAP:(3,"ORBI89",-80,"7e:d2:94:c0:05:f4",4,23,0)
+CWLAP:(3,"NETGEAR27",-80,"10:da:43:80:c4:1d",4,-9,0)
+CWLAP:(3,"Empire2.4",-83,"38:2c:4a:5d:49:40",6,-9,0)
+CWLAP:(3,"TheNewRegis",-39,"e4:f4:c6:12:20:a7",9,-12,0)
+CWLAP:(3,"mhome2",-89,"10:7b:44:af:27:00",10,3,0)
+CWLAP:(4,"CenturyLink9259",-88,"8c:59:73:2d:da:07",11,30,0)

OK
AT+CWJAP="TheNewRegis","[redacted]"
WIFI CONNECTED
WIFI GOT IP

OK
```

7. OUTPUT TO THINGSPEAK

The following shows the transmitted temperature data on thingspeak.

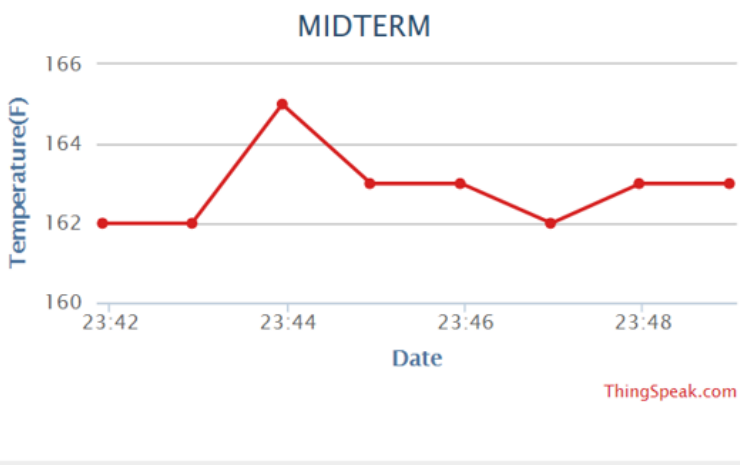
ThingSpeak™ Channels Apps Community Support

Add Visualizations | Data Export

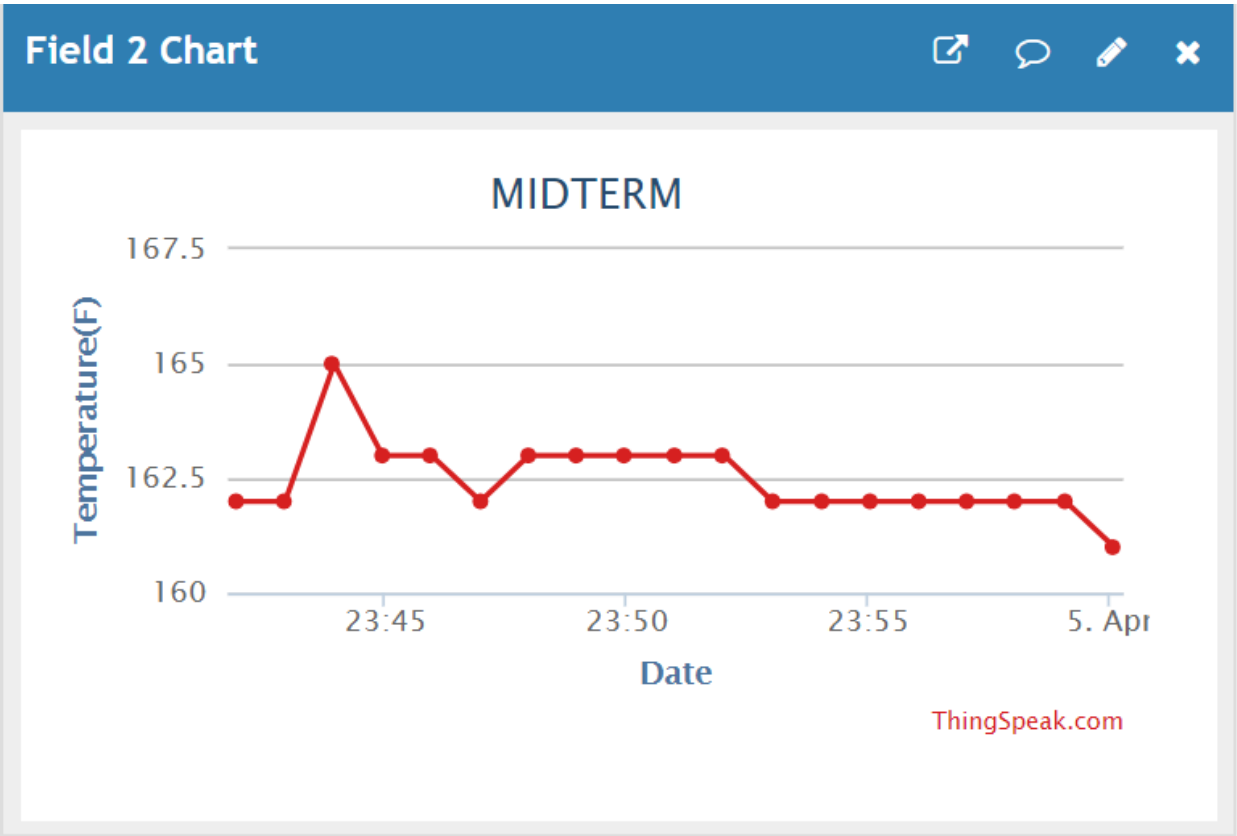
Channel Stats

Created: a day ago
Updated: 5 minutes ago
Last entry: 5 minutes ago
Entries: 51

Field 2 Chart



Add Visualizations



8. CODE

```
/*
 * Midterm 1.c
 *
 * Created: 4/6/2019 4:32:35 PM
 * Author : regis
 */

#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1

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

void read_adc(void);                    //Read ADC
void adc_init(void);                    //initialize ADC
void USART_init( unsigned int ubrr );   //initialize USART
void USART_tx_string(char *data);       //Print String USART
volatile unsigned int adc_temp;
char outs[256]; //array
volatile char received_data;
```

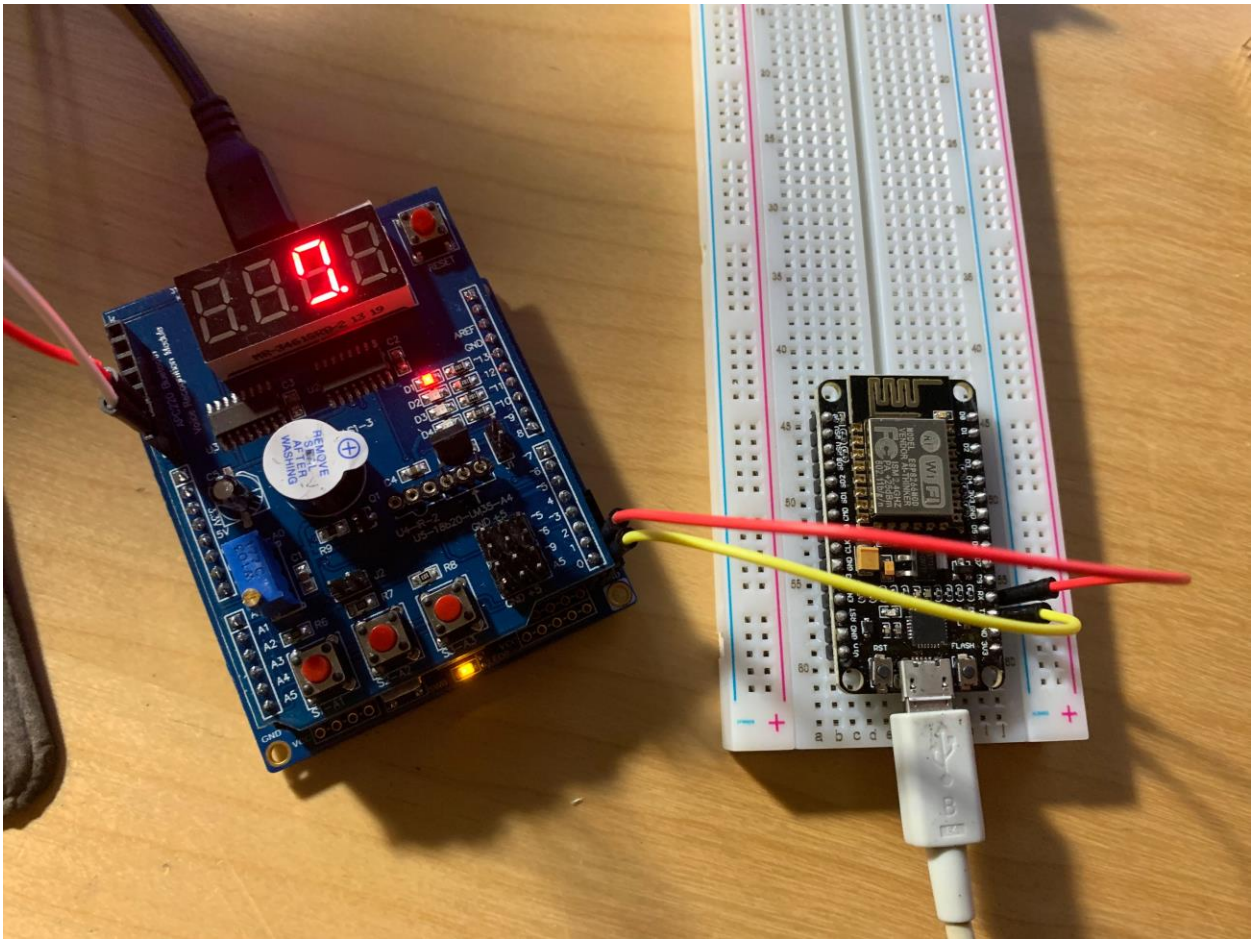
```c
void usart_send(unsigned char ascii) //send data to USART
{
    while(!(UCSR0A & (1<<UDRE0 )));
    UDR0 = ascii;

}

unsigned char usart_receive(void) //received data
{
    while(!(UCSR0A & (1<< RXC0)));
    return received_data;
}

int main(void) {

    adc_init();         // Initialize the ADC (Analog / Digital Converter)
    USART_init(MYUBRR); // Initialize the USART (RS232 interface)
    _delay_ms(500);     // wait a bit
    sei();              //interrupt


    while(1){}
}
```

```
void adc_init(void) //initialize 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<<CS12)|(1<<CS10);  // native clock
    TCNT1 = 49911;                  //((16MHz/1024)*1)-1 = 15624

}
```

```
void read_adc(void) {
    unsigned char i =4;
    adc_temp = 0; //initialize
    while (i--) {
        ADCSRA |= (1<<ADSC);
        while(ADCSRA & (1<<ADSC));
        adc_temp+= ADC;
        _delay_ms(50);
    }
    adc_temp = adc_temp / 4; // Average a few samples

}

 // INIT USART (RS-232)
void USART_init( unsigned int ubrr ) {
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // Enable receiver, transmitter & RX interrupt
    UCSR0C |= (1<<UCSZ01) | (1 << UCSZ00);
}

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

```
ISR(TIMER1_OVF_vect) //timer overflow interrupt to delay for 1 second
{
    char TEMP[256];
    unsigned char AT[] = "AT\r\n"; //AT Commands
    unsigned char CWMODE[] = "AT+CWMODE=1\r\n"; //Set MODE
    unsigned char CWJAP[] = "AT+CWJAP=\"SSID\",\"PASSWORD\"\r\n"; // Do not turn in with personal wifi/password
    unsigned char CIPMUX[] = "AT+CIPMUX=0\r\n";
    unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n";
    unsigned char CIPSEND[] = "AT+CIPSEND=100\r\n";

    _delay_ms(200);
    USART_tx_string(AT); //send commands
    _delay_ms(5000);
    USART_tx_string(CWMODE); //set mode

    _delay_ms(5000);
    USART_tx_string(CWJAP); //connect to Wifi

    _delay_ms(15000);
    USART_tx_string(CIPMUX); //select MUX

    _delay_ms(10000);
    USART_tx_string(CIPSTART);//connect TCP

    _delay_ms(10000);
    USART_tx_string(CIPSEND);//send size

    _delay_ms(5000);

    PORTC^=(1<<5);
    read_adc(); //read ADC
    snprintf(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api_key=9HD0YXSMDWBFG6Q7&field2=%3d\r\n", adc_temp);// print it
    USART_tx_string(outs);//send data
    _delay_ms(10000);
    TCNT1 = 49911; //reset

}
```

9. BOARD IMPLEMENTATION



10. GITHUB LINK OF THIS DA

https://github.com/regis-shaquille/submissions-SR/tree/master/Midterms/Midterm%201

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

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