CPE301 – SPRING 2019

Midterm 1

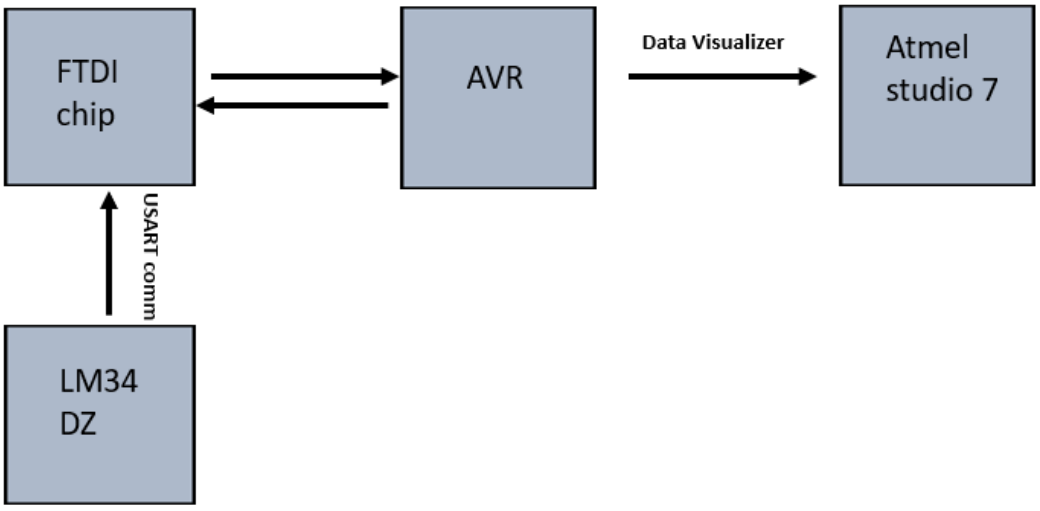
Student Name: Francisco Mata Carlos
Student #: 1012593607
Student Email: matacarl@unlv.nevada.edu
Primary Github address: https://github.com/chicosisco/da_sub.git
Directory: repository/cpe301/DesignAssignments/midterm1

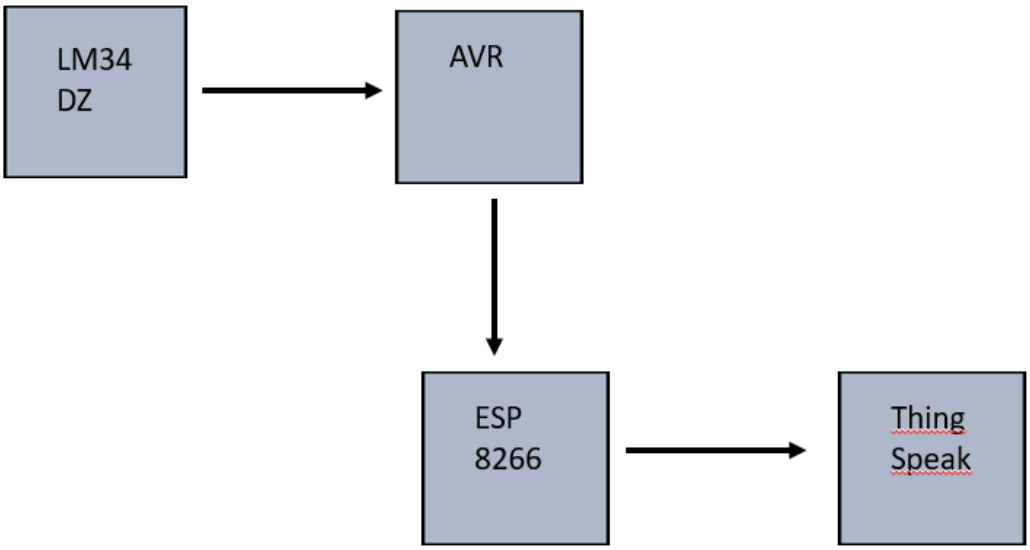
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment are the next:

a. Atmega328p Xplained Mini
b. ESP8266 module
c. ESPlorer Software
d. ESP8266 Flasher Software
e. FTDI chp
f. Atmel Studio 7

Block diagrams used for this assignment





2. INITIAL/MODIFIED/DEVELOPED CODE OF Midterm 1

```
#define F_CPU 16000000UL
#define BAUD_RATE 9600
#include <avr/interrupt.h>

#include <avr/io.h>
#include <util/delay.h>
int over_flow=0;


void usart_init ();
void usart_send (unsigned char ch);

int main (void)
{
	usart_init ();

	/** Setup and enable ADC **/
	ADMUX = (0<<REFS1)|    // Reference Selection Bits
	(1<<REFS0)|    // AVcc - external cap at AREF
	(0<<ADLAR)|    // ADC Left Adjust Result
	(1<<MUX2)|     // Analog Channel Selection Bits
	(0<<MUX1)|     // ADC5 (PC5 PIN27)
	(1<<MUX0);
	ADCSRA = (1<<ADEN)|    // ADC ENable
```

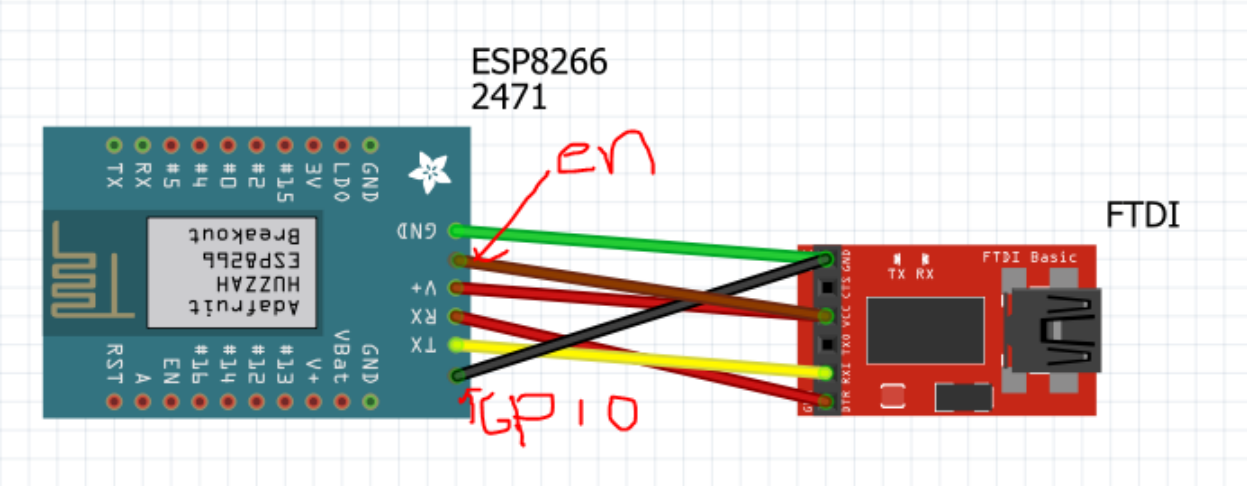
```
        (0<<ADSC)|     // ADC Start Conversion
        (0<<ADATE)|    // ADC Auto Trigger Enable
        (0<<ADIF)|     // ADC Interrupt Flag
        (0<<ADIE)|     // ADC Interrupt Enable
        (1<<ADPS2)|    // ADC Prescaler Select Bits
        (0<<ADPS1)|
        (1<<ADPS0);



////////////////////////////////////////////////////////////////////////////
        //DDRB |=(1<<DDB4);  // setting PB1 as output
        TIMSK0 |= (1<<TOIE0);
        TCNT0 = 0;                // setting initial value for counter
        sei();                    // enable global interrupts
        TCCR0B |=(1<<CS02)|(1<<CS00);  // setting prescaler to 1024


////////////////////////////////////////////////////////////////////////////

        while (1)
        {

               ADCSRA|=(1<<ADSC);  //start conversion
               while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish

               ADCSRA |= (1<<ADIF);
               if (over_flow==61)          // when TCNT0 overflows 61 times, then the
information is updated
               {
                      //functions
                      void read_adc(void);  //Read LM34 to ADC
                      void adc_init(void);  // Initialize ADC
                      void usart_init( unsigned int ubrr ); // initialize comms
                      void usart_send(char *data); //usart string
                      volatile unsigned int adc_temp; // raw temperature variable
                      volatile unsigned int temp; // Volatile Fahrenheit temperature
variable
                      char outs[256]; // String being send to usart commands
                      volatile char received_data; //receiving usart comm

                      // commands
                      //AT check
                      char AT[] = "AT\r\n";
                      //Set device mode
                      char AT_CWMODE[] = "AT+CWMODE=1\r\n";
                      // Wifi connection, SSID and Password
                      char AT_CWJAP[] = "AT+CWJAP=\"SSID\",\"Password\"\r\n";
                      //device IP Address Mode
                      char AT_CIPMUX[] = "AT+CIPMUX=0\r\n";
                      // Start connection to Thingspeak.com website, 80
                      char AT_CIPSTART[] =
"AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n";
                      // string length=100
                      char AT_CIPSEND[] = "AT+CIPSEND=100\r\n";

                      // Send commands
                      _delay_ms(200);
```

```
                        usart_send(AT);
                        _delay_ms(5000);
                        usart_send(AT_CWMODE);
                        _delay_ms(5000);
                        usart_send(AT_CWJAP);
                        _delay_ms(15000);
                        usart_send(AT_CIPMUX);
                        _delay_ms(10000);
                        usart_send(AT_CIPSTART);
                        _delay_ms(10000);
                        usart_send(AT_CIPSEND);
                        _delay_ms(5000);

                        PORTC^=(1<<5);
                        read_adc();    // Read ADC value from LM34
                        temp;

                        // Print Data to Thingspeak using the info provided below
                        snprintf(outs,sizeof(outs),"https://api.thingspeak.com/KEY\r\n",
temp);
                        usart_send(outs);//send info
                        _delay_ms(10000);


                        int a = ADCL;
                        a = a | (ADCH<<8);
                        a = (a/1024.0) * 5000/10;
                        usart_send((a/100)+'0');
                        a = a % 100;
                        usart_send((a/10)+'0');
                        a = a % 10;
                        usart_send((a)+'0');
                        usart_send('\r');
                        over_flow=0;
                }

                //_delay_ms(1000);
        }
        return 0;
 }


 // timer_0 overflow interrupt

////////////////////////////////////////////////////////////////////////////////
///////////////
 ISR(TIMER0_OVF_vect)
 {
        while (!(TIFR0 & 0X01)==0);
        TCNT0=0X00;  //resetting counter to zero
        TIFR0=0X01;  // reset the overflow flag
        over_flow++;  //increasing overflow counter

 }
```

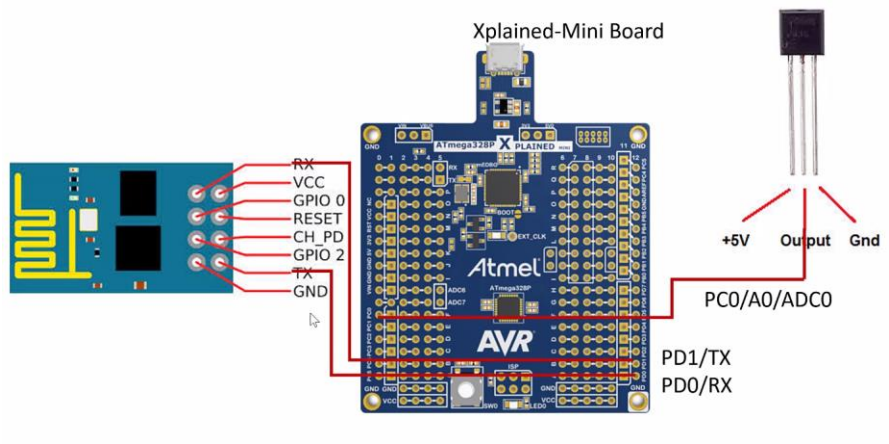
```
////////////////////////////////////////////////////////////////////////////////////////////
//////////////

 void usart_init (void)
 {
        UCSR0B = (1<<TXEN0);
        UCSR0C = (1<< UCSZ01)|(1<<UCSZ00);
        UBRR0L = F_CPU/16/BAUD_RATE-1;
 }

 void usart_send (unsigned char ch)
 {
        while (! (UCSR0A & (1<<UDRE0)));  //wait until UDR0 is empty
        UDR0 = ch;                                                  //transmit ch
 }

 void usart_print(char* str)
 {
        int i = 0;
        while(str[i] != 0)
        usart_send(str[i]);
 }
```

3. SCHEMATICS

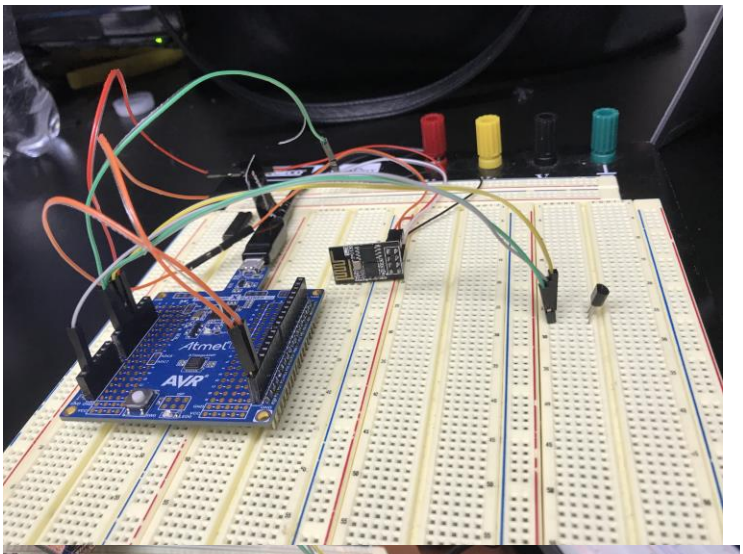
Schematic to program

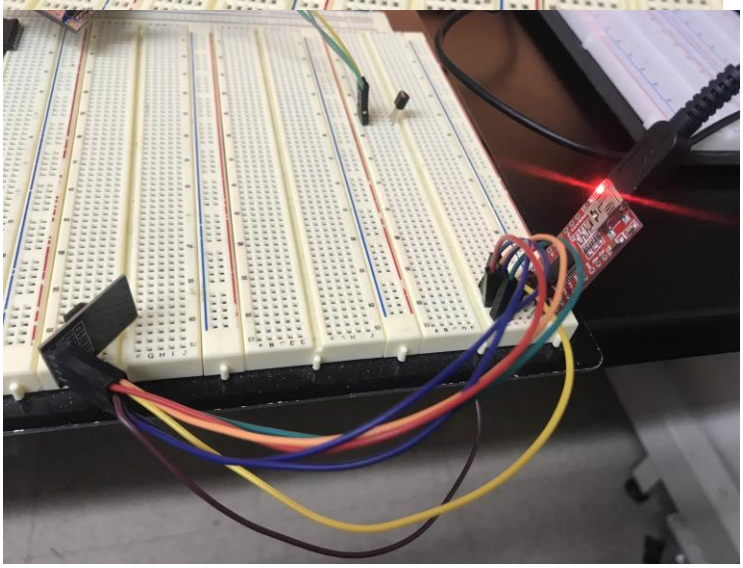


Schematic to send data to

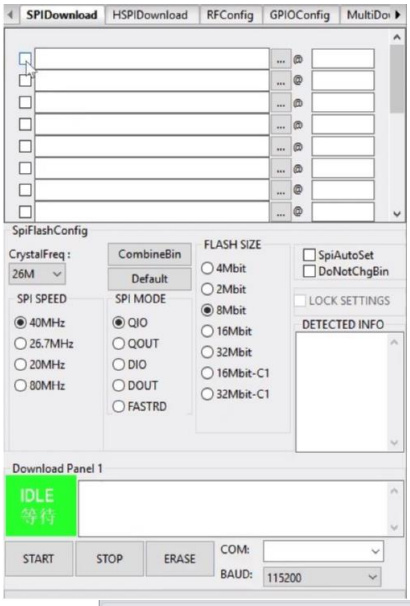


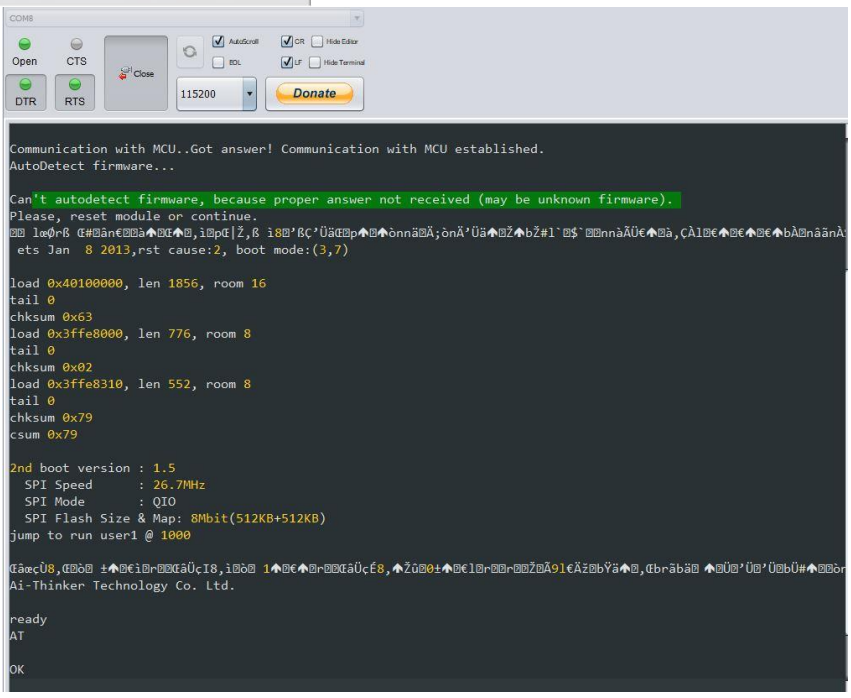
4. SCREENSHOTS OF BOARD SET UP

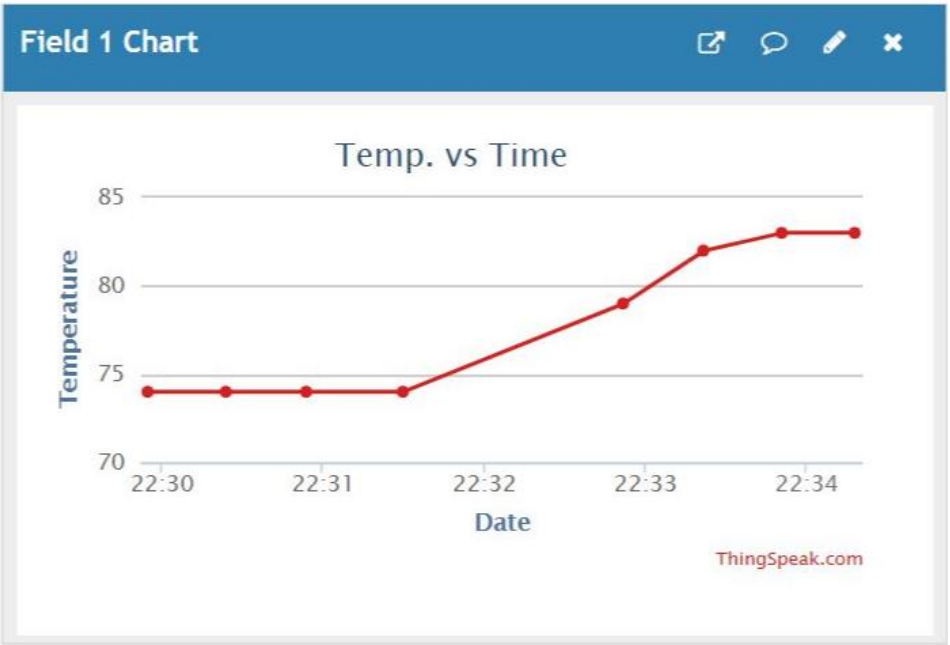




5. SCREENSHOTS OF EACH TASK OUTPUT







6. VIDEO LINKS OF EACH DEMO

https://youtu.be/VALUa6F95FM

7. GITHUB LINK OF THIS DA

https://github.com/chicosisco/da_sub.git

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

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