CPE301 – SPRING 2019

MIDTERM 1

Student Name: Chris Barr

Student #: 2000682859

Student Email: barrc1@unlv.nevada.edu

Primary Github address: https://github.com/BarrChris

Directory: https://github.com/BarrChris/submission\_da.git

Submit the following for all Labs:

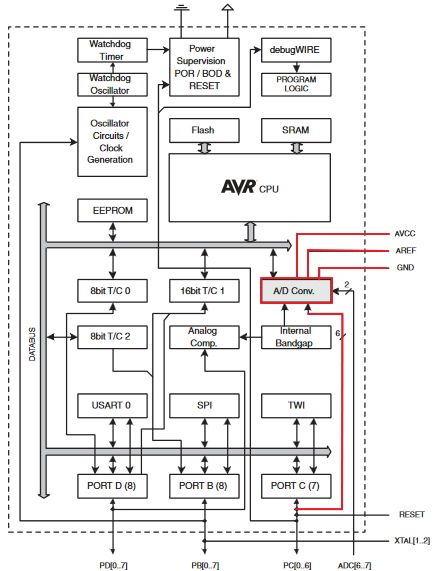
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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

* 5 M-F wires
* 4 M-M wires
* Atmega 328p
* FTDI Basic (5V)
* ESP8266

Block diagram with pins used in the Atmega328P



Used PD0 for RX (receive) and PD1 for TX (transmit)

Followed by the AVcc for 3.3V for the ESP device and relative ground

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

/\*

\* Midterm1.c

\*

\* Created: 4/4/2019 3:23:52 PM

\* Author : Chris

\*/

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdlib.h>

#include <stdint.h>

#define F\_CPU 16000000UL

#define BAUD 115200

#define FOSC 16000000

#define UBRREQ FOSC/8/BAUD -1

volatile uint8\_t ADCvalue**;**

volatile unsigned char ADCtemp**[**5**];**

volatile unsigned char CWMODE**[]** **=** "AT+CWMODE=3\r\n"**;** // WiFi mode = 3

volatile unsigned char WIFI**[]** **=** "AT+CWJAP=\"(WiFi Username)\", \"(WiFi Password)\"\r\n"**;** // Connects to local internet connection

volatile unsigned char ENABLE**[]** **=** "AT+CIPMUX=0\r\n"**;** // Connected to my 2.4GHz WiFi

volatile unsigned char CIPSTART**[]** **=** "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n"**;** //TCP, strip remote IP, port = 80

volatile unsigned char CIPSEND**[]** **=** "AT+CIPSEND=45\r\n"**;** // Length of data = 45

volatile unsigned char SEND\_DATA**[]** **=** "GET /update?key=0D46YJRKCJ7XZ3AV&field1="**;** // Write API key

volatile unsigned char PAUSE**[]** **=** "\r\n\r\n"**;** // Go to left side, skip a line, go to left side, skip a line

void ADC\_init **(**void**);**

void UART\_init **(**void**);**

void signal\_AT **(**volatile unsigned char c**[]);**

int main**(** void **)**

**{**

ADC\_init**();** // Initializes ADC values

UART\_init**();** // Initializes UART values

// ========================================================================================================

// Calls functions to enable WiFi mode, connects to the specific WiFi, establishes single connection

// ========================================================================================================

\_delay\_ms**(**1000**);**

signal\_AT**(**CWMODE**);**

\_delay\_ms**(**1000**);**

signal\_AT**(**WIFI**);**

\_delay\_ms**(**2000**);**

signal\_AT**(**ENABLE**);**

// ========================================================================================================

**while(**1**)** // Constantly send values through the cloud until device turns off

**{**

// ========================================================================================================

// Calls functions to connect to thingspeak, sets length of data to be sent, sends the data values to cloud,

// pauses till data goes through cloud

// ========================================================================================================

\_delay\_ms**(**1000**);**

signal\_AT**(**CIPSTART**);** // Connects to thingspeak

\_delay\_ms**(**1000**);**

signal\_AT**(**CIPSEND**);** // Sets length of data

\_delay\_ms**(**1000**);**

signal\_AT**(**SEND\_DATA**);**

signal\_AT**(**ADCtemp**);** // Sends data

signal\_AT**(**PAUSE**);** // Pauses to load data

// ========================================================================================================

**}**

**}**

void ADC\_init **(**void**)**

**{**

//ADC Multiplexer Selection Register

ADMUX **=** **(**1 **<<** REFS0**)** **|** // Voltage reference during conversion, "AVcc with external capacitor at AREF pin"

**(**1 **<<** ADLAR**);** // Left adjust ADC conversion result in ADC Data Register

//ADC Control and Status Register A

ADCSRA **=** **(**1 **<<** ADEN**)** **|** // ADC enable

**(**1 **<<** ADSC**)** **|** // ADC Start Conversion

**(**1 **<<** ADATE**)** **|** // ADC Auto Trigger enable

**(**1 **<<** ADIE**)** **|** // ADC Interrupt enable

**(**1 **<<** ADPS2**)** **|**

**(**1 **<<** ADPS1**)** **|**

**(**1 **<<** ADPS0**);** // ADPS2:0 = 111 = 128 prescaler

**}**

void UART\_init **(**void**)**

**{**

//USART Baud Rate Registers

UBRR0H **=** UBRREQ **>>** 8**;** // Shifts right to store upper 8 bits

UBRR0L **=** UBRREQ**;** // Store lower 8 bits

//USART Control and Status Register (A/B/C)

UCSR0A **|=** **(**1 **<<** U2X0**);** // Doubles USART transmission speed

UCSR0B **|=** **(**1 **<<** TXEN0**);** // Enables USART transmitter

UCSR0C **|=** **(**1 **<<** UCSZ01**)** **|**

**(**1 **<<** UCSZ00**);** // 8-bit character size

sei**();** // Enable global interrupt

**}**

ISR**(**ADC\_vect**)**

**{**

volatile unsigned int j**=**0**;**

char temp**[**5**];**

ADCvalue **=** **(**ADCH **<<** 1**);** // Shifts the value left to one place

itoa**(**ADCvalue**,** temp**,** 10**);** // Converts integers to string

// Takes ADCvalue, turns it into an ASCII representation

// the ASCII representation will be stored under 'temp'

// '10' represents the buffer

**while** **(**j**<**5**)** // Transfers the temp string from itoa() to ADCtemp

**{**

ADCtemp**[**j**]** **=** temp**[**j**];**

j**++;**

**}**

**}**

void signal\_AT**(**volatile unsigned char c**[])** **{**

volatile unsigned int i**=**0**;**

volatile unsigned int j**=**0**;**

j **=** 0**;** // Initializes counter

**while** **(**c**[**j**]** **!=** 0x00**)** // While not at end of string

**{**

j**++;**

**}**

**while** **(**i**<**j**)** // If UDRE0 = 1, buffer can be written to

**{**

**while(!(**UCSR0A **&** **(**1 **<<** UDRE0**)));**

UDR0 **=** c**[**i**];**

i**++;**

**}**

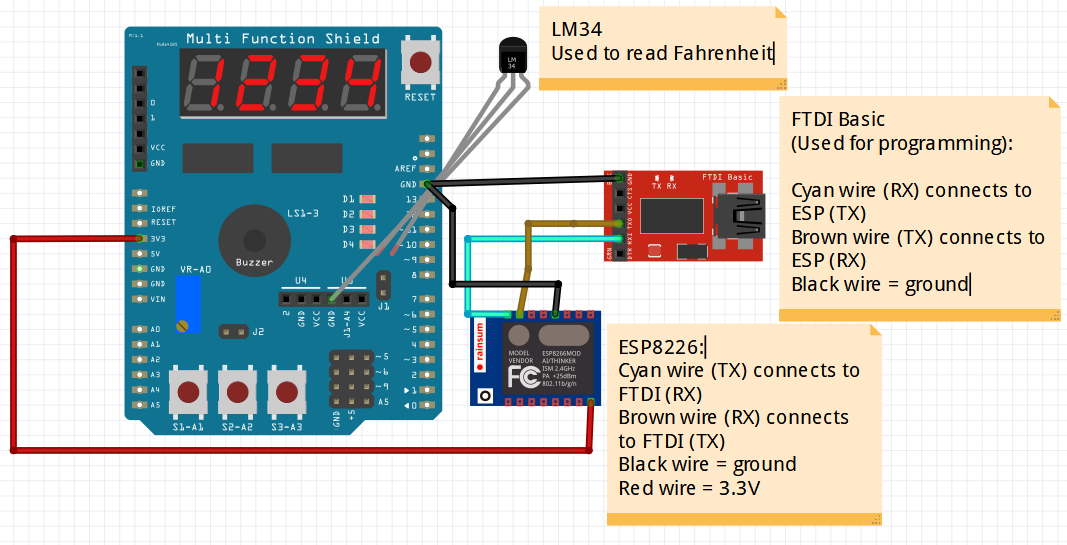
**}**

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

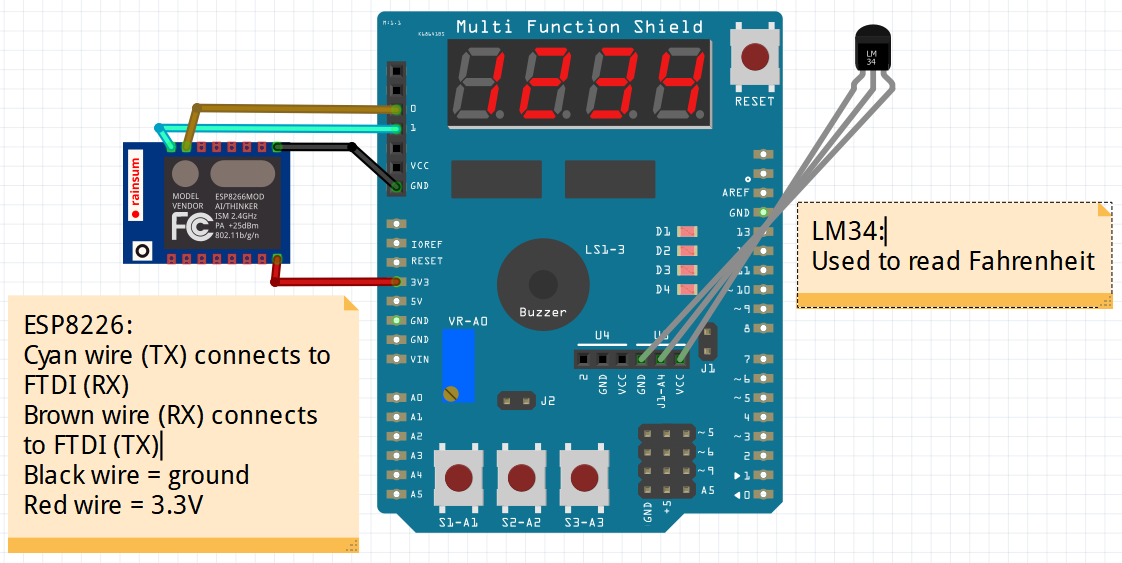
N/A

1. **SCHEMATICS**

Programming Stage:

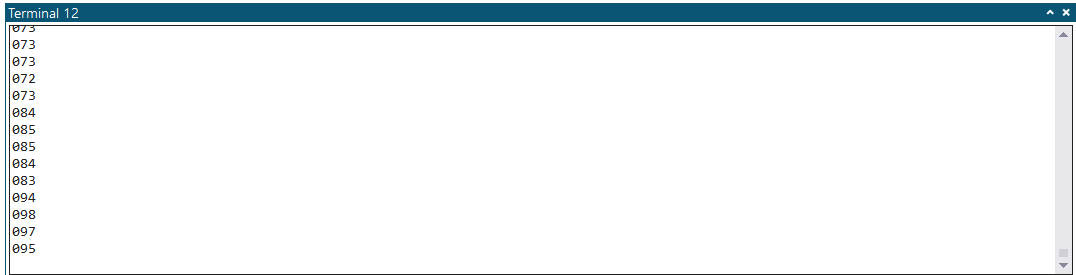


Output Stage:



1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

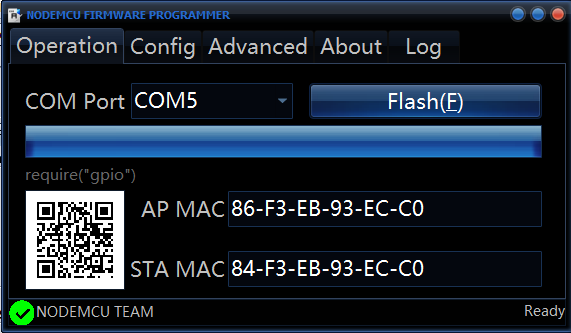
Task 1 & Task 2:



Programed the ADC to read temperature sensor. Here are the temperature readings after lighting a lighter under the LM34

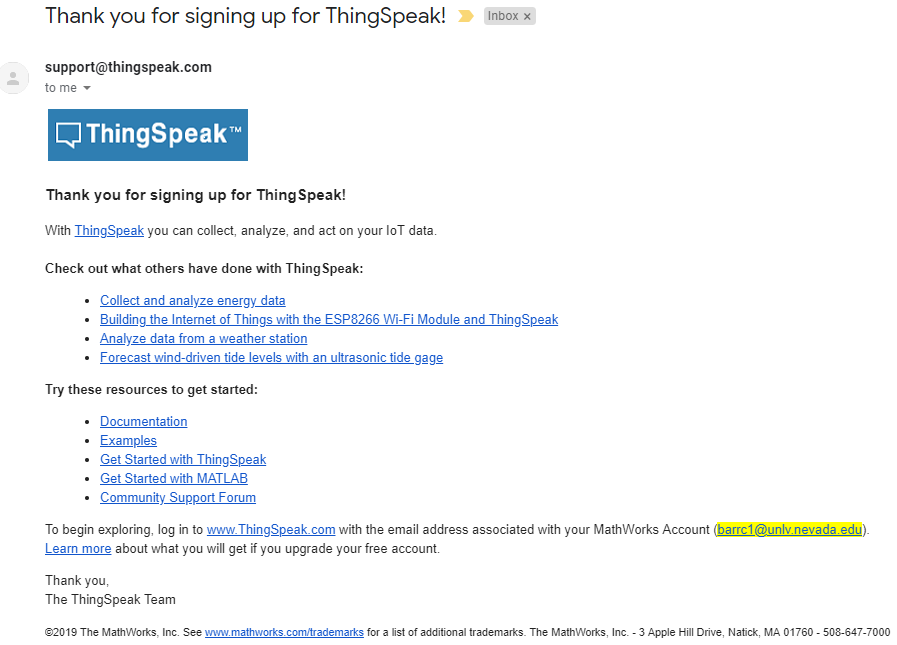
Note: Check DA3B for reference

Task 3:



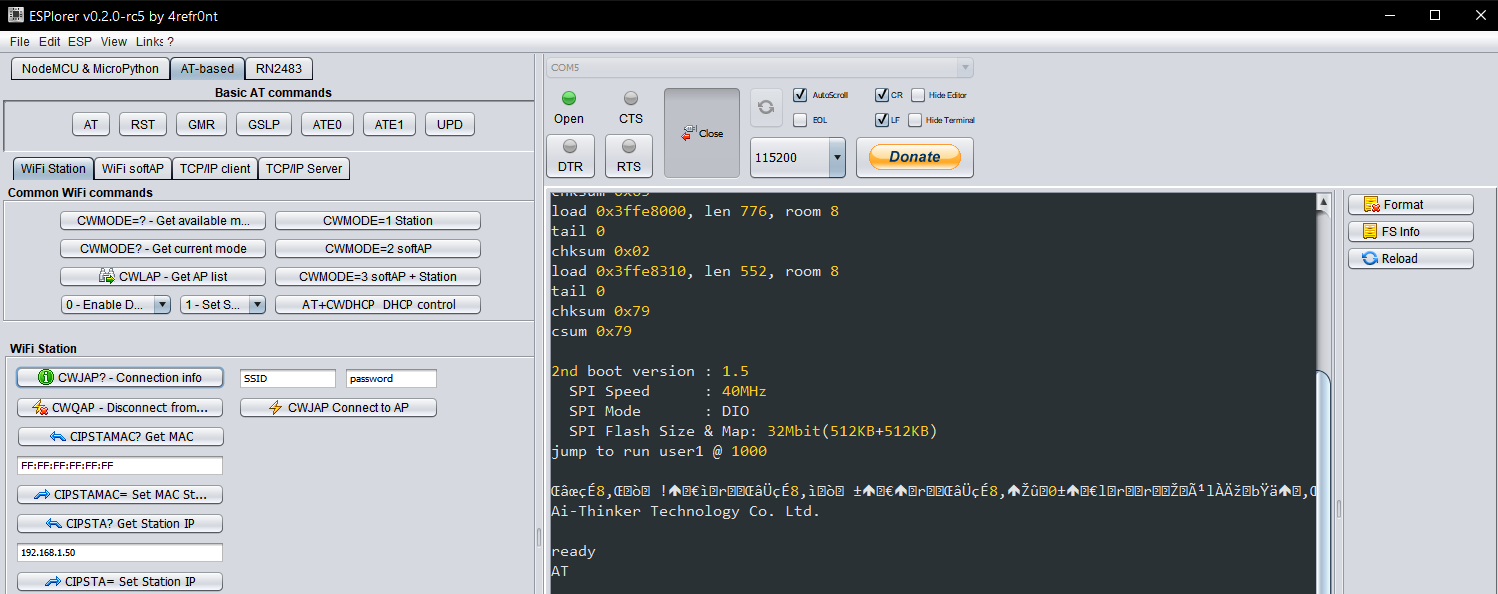
Firmware successfully installed onto ESP8226 through the FTDI Basic

Task 4:

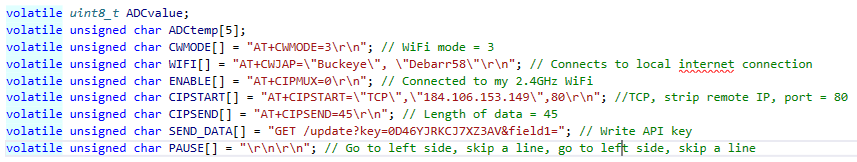
  
Registration with ThingSpeak confirmed

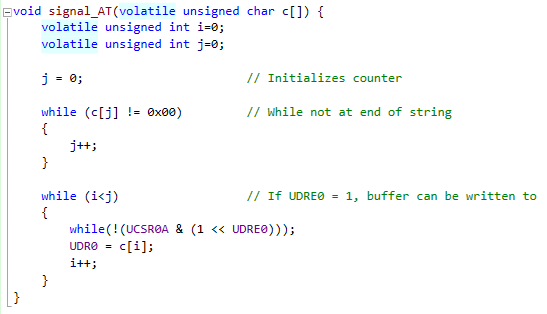
Task 5:

First, we must determine if the ESP8226 can work and connect to ThingSpeak by checking it on ESPlorer:

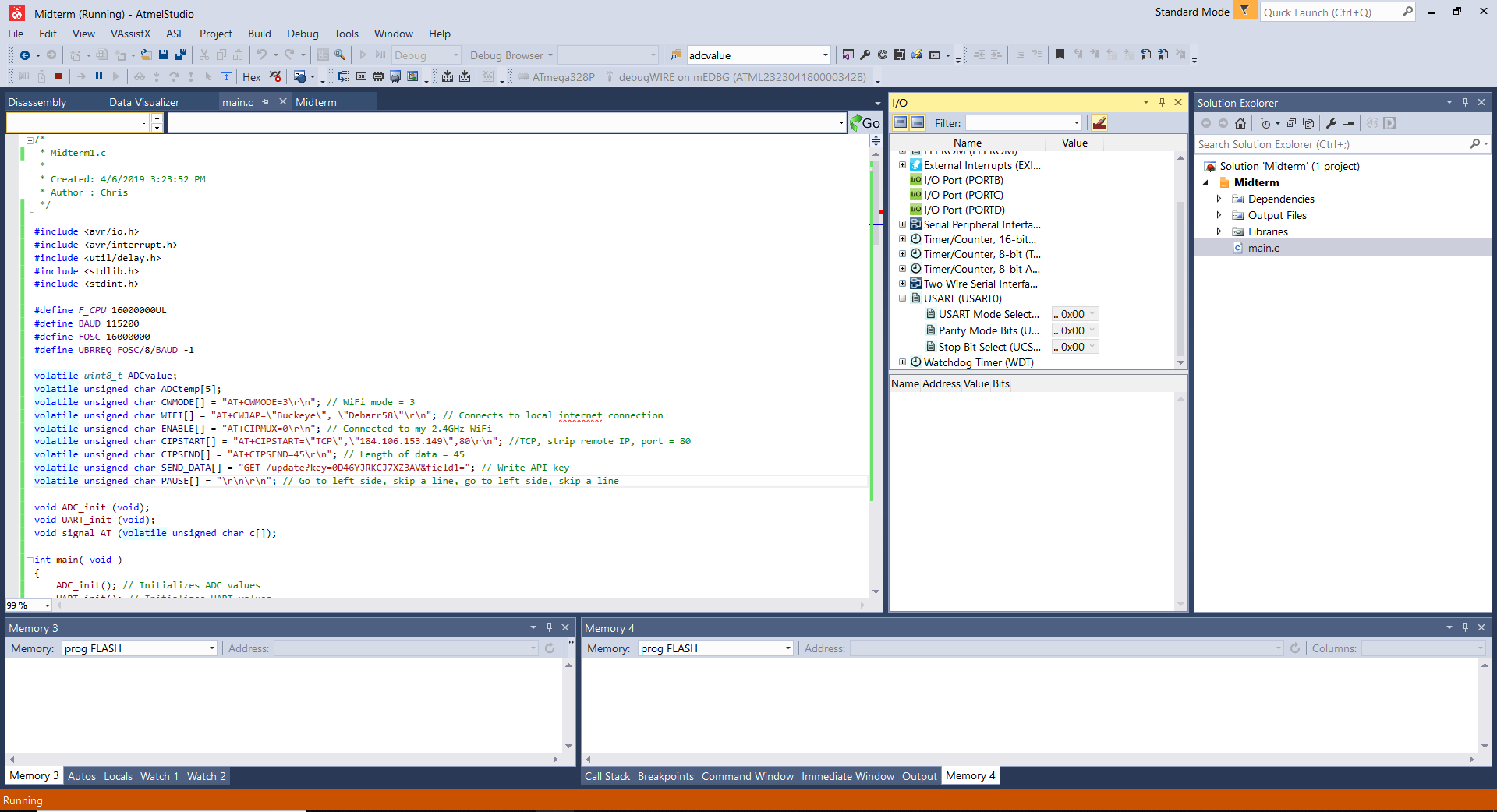


Our board connects to the ESPlorer software successfully and can use the correct ports to connect to the internet





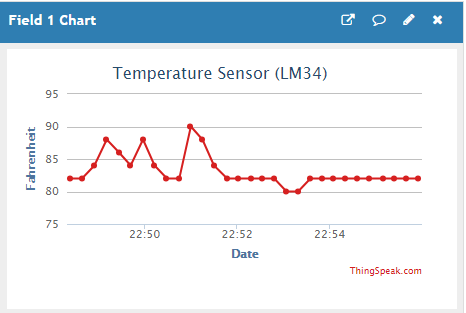
Using these AT commands and function on Atmel Studio 7



We can run the program with the Atmega328P connected to the ESP8226 and LM34

This will transmit the temperature sensor value through UART port, given the proper settings

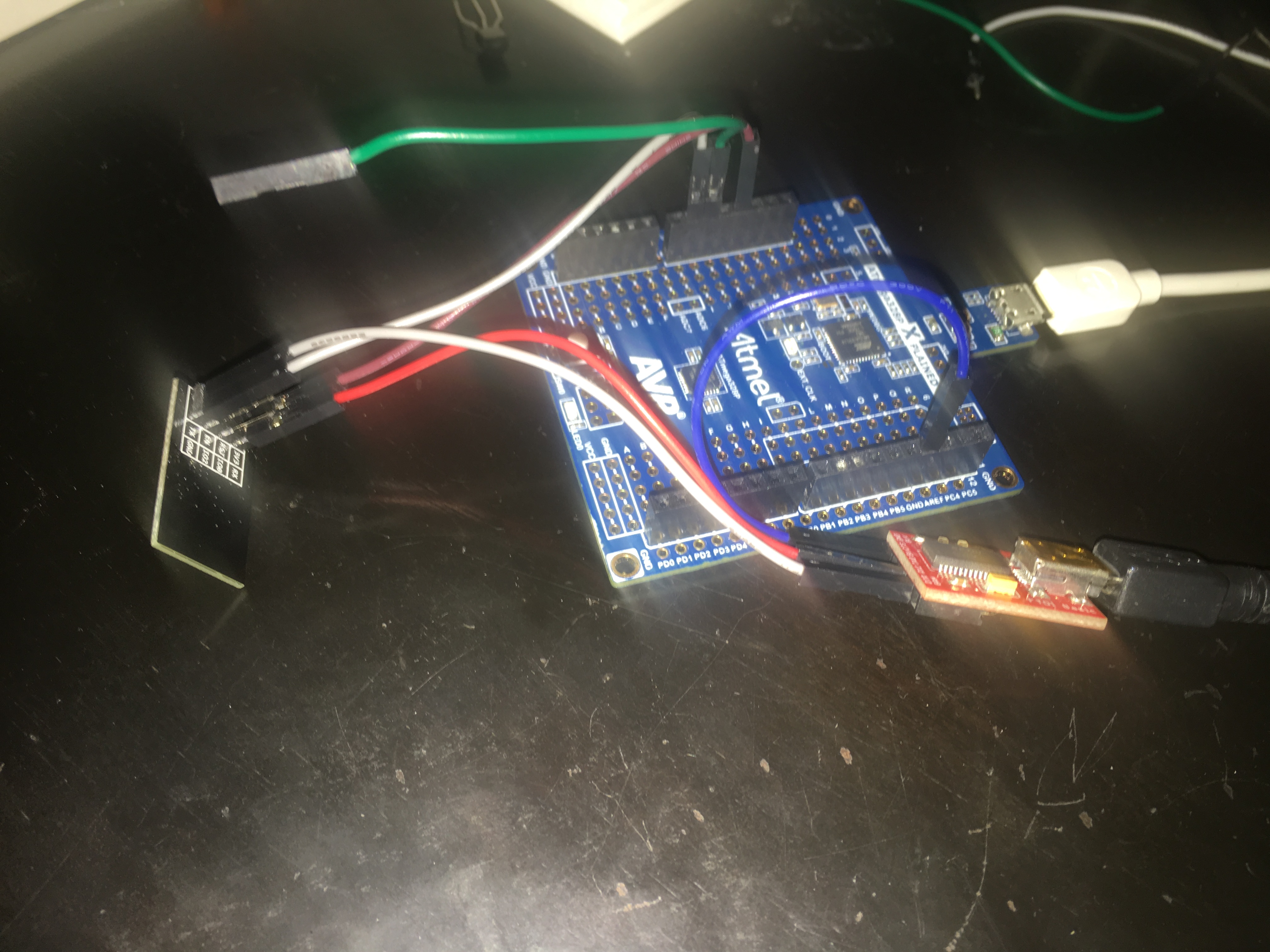
Task 6:



This is the graph that the information the temperature sensor gathered, AT commands sent the data through the cloud, and is now uploaded onto ThingSpeak

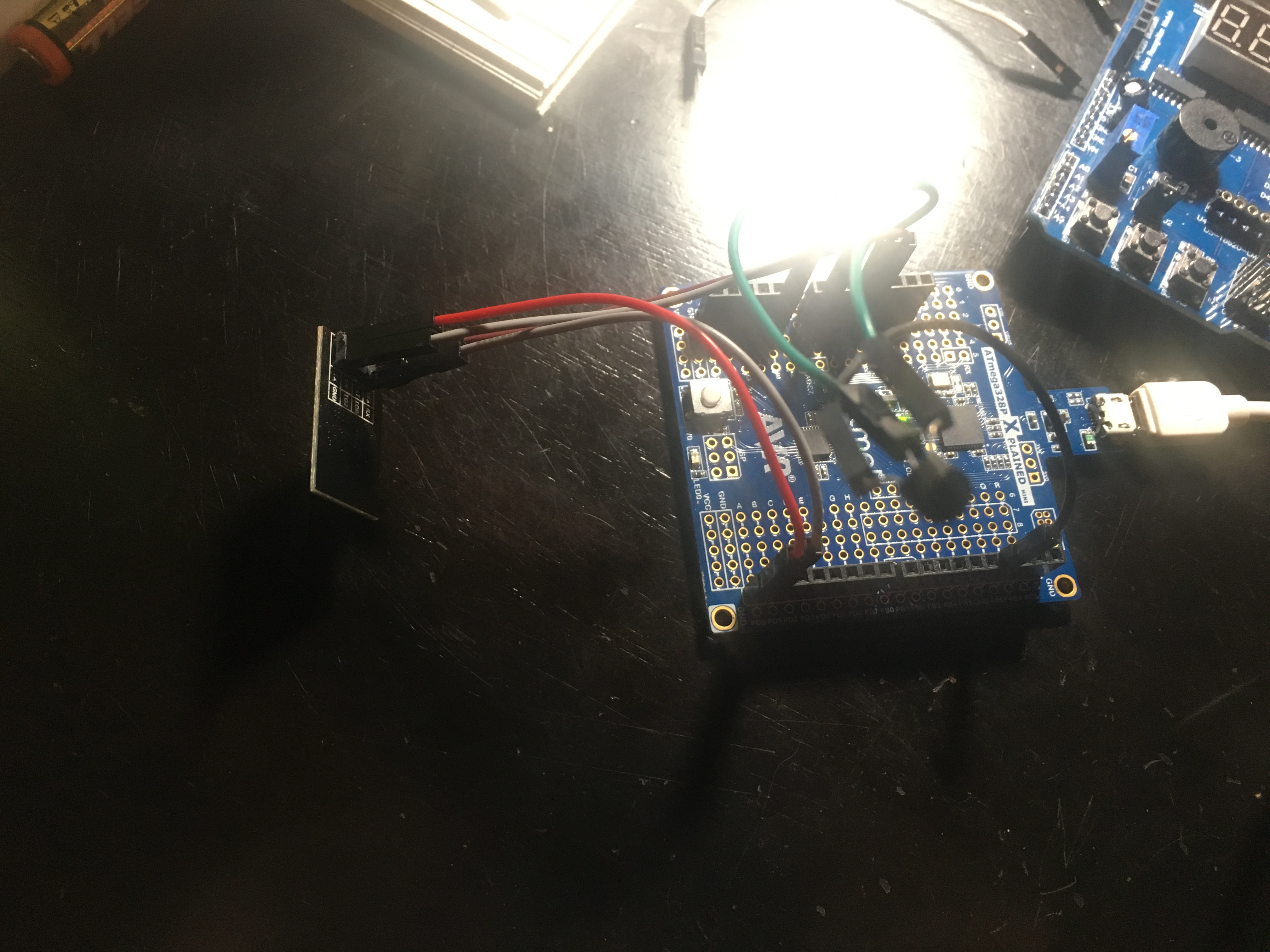
1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Programming Stage:



The ESP8226 and FTDI basic is connected to the ATmega328P

Output Stage:



Only the ESP8226 and LM34 is connected to the ATmega328P

1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=u3gDQam7OOc&t=18s>

1. **GITHUB LINK OF THIS DA**

<https://github.com/BarrChris/submission_da/tree/master/Midterms/Midterm1/Midterm1>

**Student Academic Misconduct Policy**

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

“This assignment submission is my own, original work”.

Chris Barr