CPE301 – SPRING 2019

Midterm 1

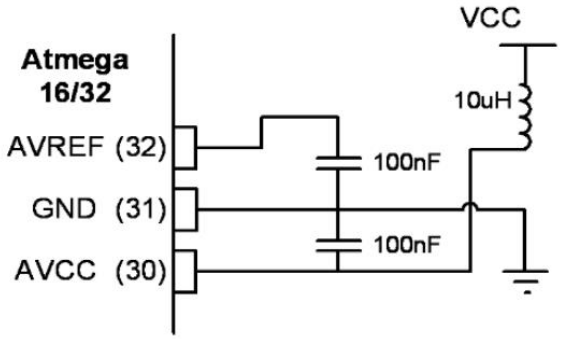
Student Name: Tyler Gardenhire
Student #: 8000450294
Student Email: gardenhi@unlv.nevada.edu
Primary Github address: gardenhi@unlv.nevada.edu
Directory: https://github.com/tylergardenhire/submission_projects.git

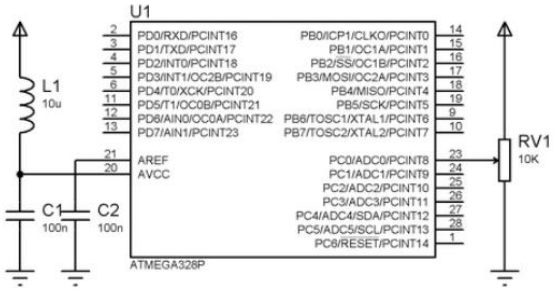
Submit the following for all Labs:

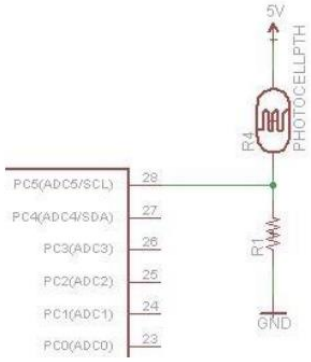
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, 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).

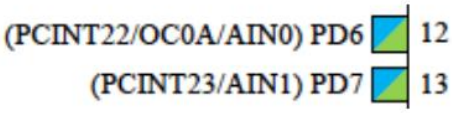
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

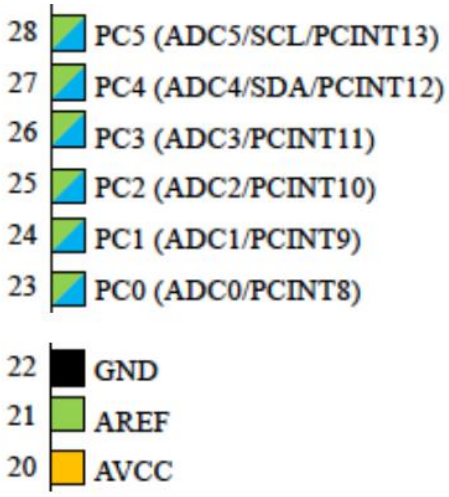
Atmel Studio 7 w/ AVR assembly, Atmega328p board, FTDI chip, and ESP module used.











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

```
/*
 * midterm1.c
 *
 * Created: 4/7/2019 12:36:13 PM
 * Author : Tyler
 */

#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 BAUDEQ 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=\"gardenhi\", \"BasketballStar23\"\r\n";
//connect to internet

volatile unsigned char ENABLE[] = "AT+CIPMUX=0\r\n"; //connected to 2.4GHz WiFi
volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n";
//tcp
volatile unsigned char CIPSEND[] = "AT+CIPSEND=45\r\n"; //length of data = 45
volatile unsigned char SEND_DATA[] = "GET /update?key=1Q0Z8SVE7A1ONWS1&field1="; //API
volatile unsigned char PAUSE[] = "\r\n\r\n"; //go to left side, skip a line

void ADCinit (void);
void UARTinit (void);
void sendAT (volatile unsigned char c[]);

int main( void )
{
	ADCinit(); //initializes ADC values
	UARTinit(); //initializes UART values

	//calls functions to enable WiFi mode, connects to the specific WiFi
	_delay_ms(1000);
	signal_AT(CWMODE);

	_delay_ms(1000);
	signal_AT(WIFI);

	_delay_ms(2000);
	signal_AT(ENABLE);

	while(1) //send values through the cloud until off
	{

		//calls functions to connect to thingspeak
		_delay_ms(1000);
		signal_AT(CIPSTART); //connects to thingspeak
```

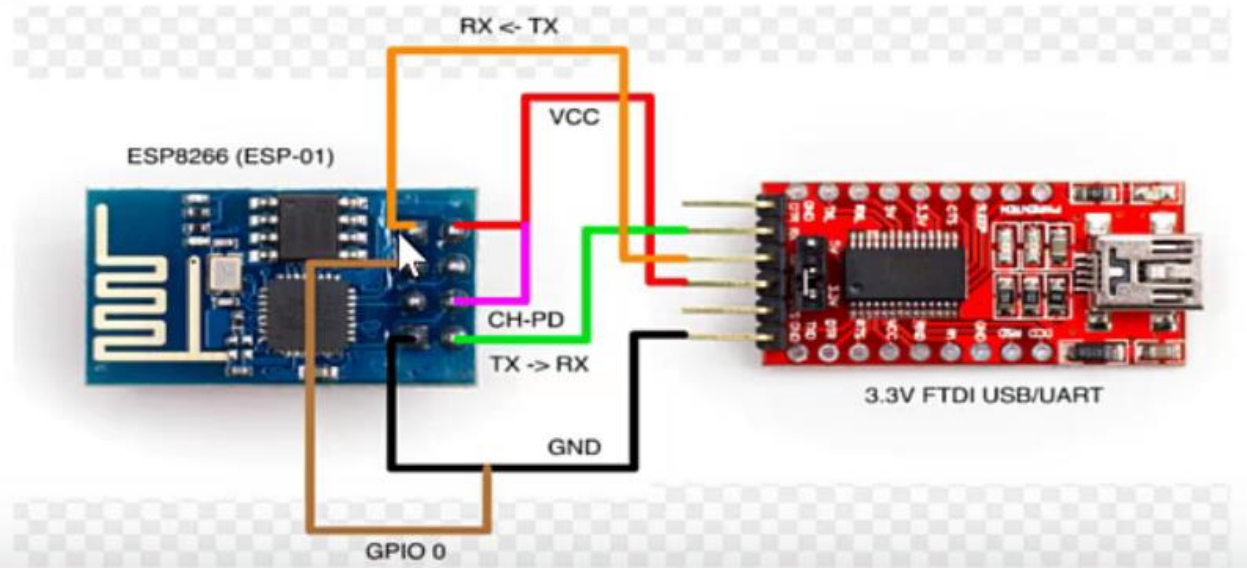
```
                _delay_ms(1000);
                signal_AT(CIPSEND); //set length of data

                _delay_ms(1000);
                signal_AT(SEND_DATA);
                signal_AT(ADCtemp); //send data

                signal_AT(PAUSE); //pause, load data
        }
}

void ADCinit (void)
{
        ADMUX = (1 << REFS0) |                          //voltage reference
        (1 << ADLAR);                                   //left adjust ADC conversion

        //ADC Control and Status Register A, ADC enable, ADC start conversion,
        //ADC auto trigger enable, ADC interrupt enable, 128 prescaler

        ADCSRA = (1 << ADEN) | (1 << ADSC) | (1 << ADATE) |
        (1 << ADIE) | (1 << ADPS2) (1 << ADPS1) (1 << ADPS0);
}

void UARTinit (void)
{
        UBRR0H = UBRREQ >> 8;               //shifts right to store upper 8 bits
        UBRR0L = UBRREQ;                    //store lower 8 bits
        UCSR0A |= (1 << U2X0);              //doubles USART transmission speed
        UCSR0B |= (1 << TXEN0);             //enables USART transmitter
        UCSR0C |= (1 << UCSZ01) |
        (1 << UCSZ00);                              // 8-bit 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
        while (j<5)                         //transfers the temp string
        {
                ADCtemp[j] = temp[j];
                j++;
        }
}


void sendAT(volatile unsigned char c[]) {
        volatile unsigned int i=0;
        volatile unsigned int j=0;

        j = 0;                                      //initialize counter
```

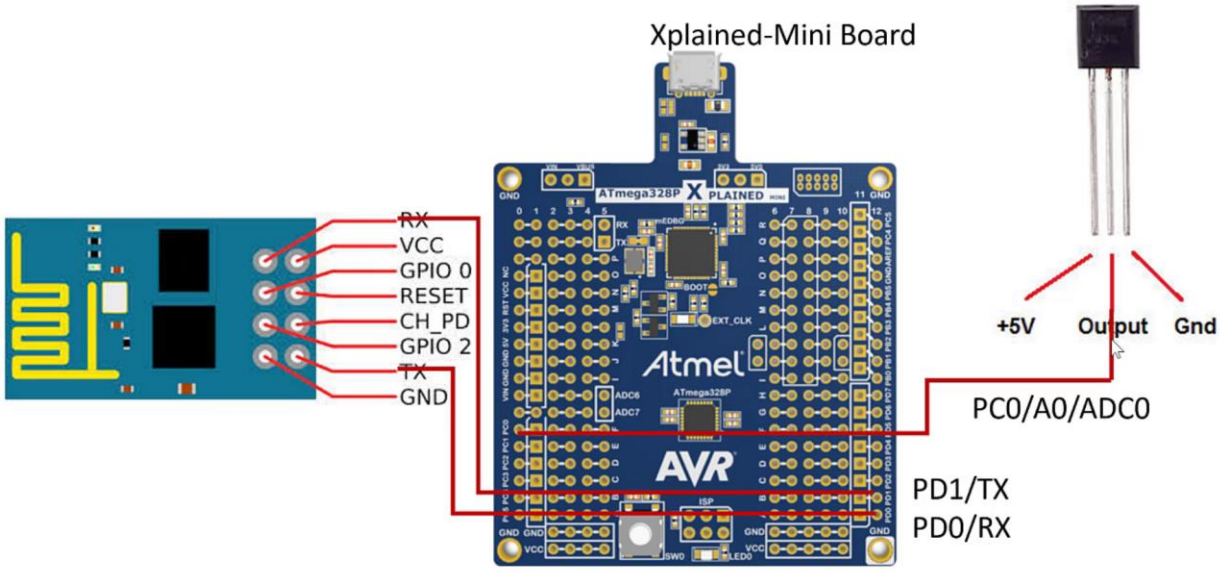
```
        while (c[j] != 0x00)                        //while not at end of string
                j++;

        while (i<j)                                 //if UDRE0 = 1, buffer can be written
        {
                while(!(UCSR0A & (1 << UDRE0)));
                UDR0 = c[i];
                i++;
        }
}
```

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

N/A

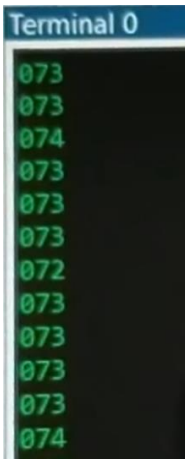
4. SCHEMATICS



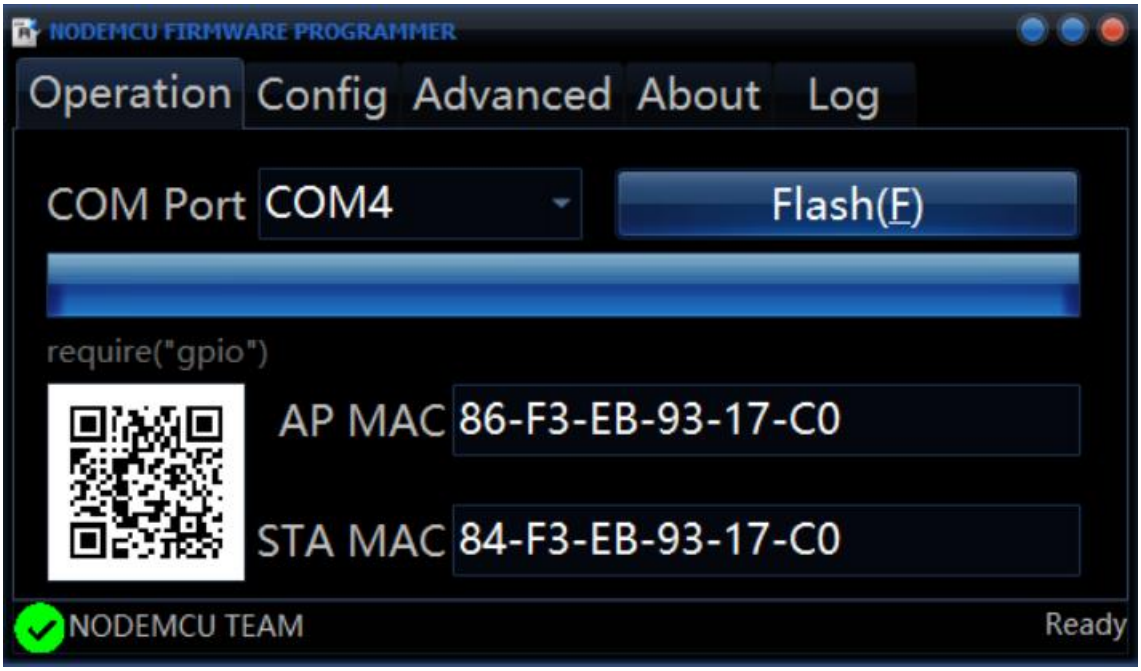


5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 2:



Task 3:



Task 4:

Thank you for signing up for ThingSpeak! Inbox ×

support@thingspeak.com 2:12 PM (0 minutes ago)
to me

ThingSpeak™

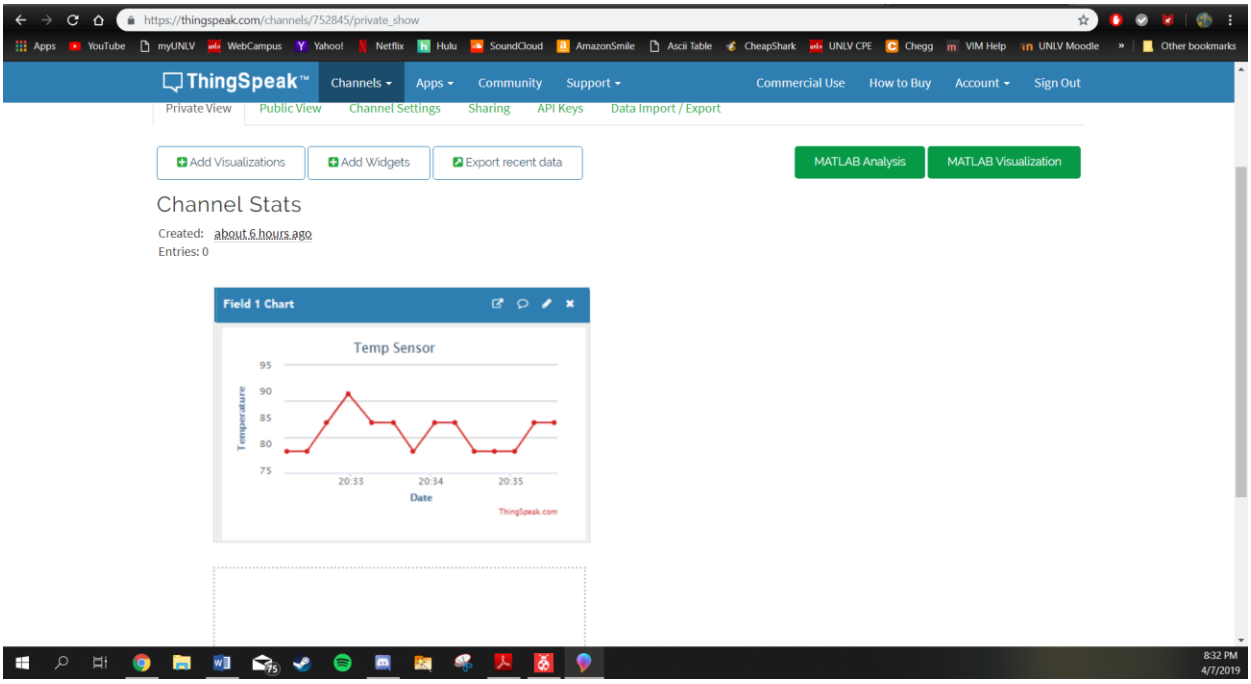
Thank you for signing up for ThingSpeak!

With ThingSpeak you can collect, analyze, and act on your IoT data.

Check out what others have done with ThingSpeak:

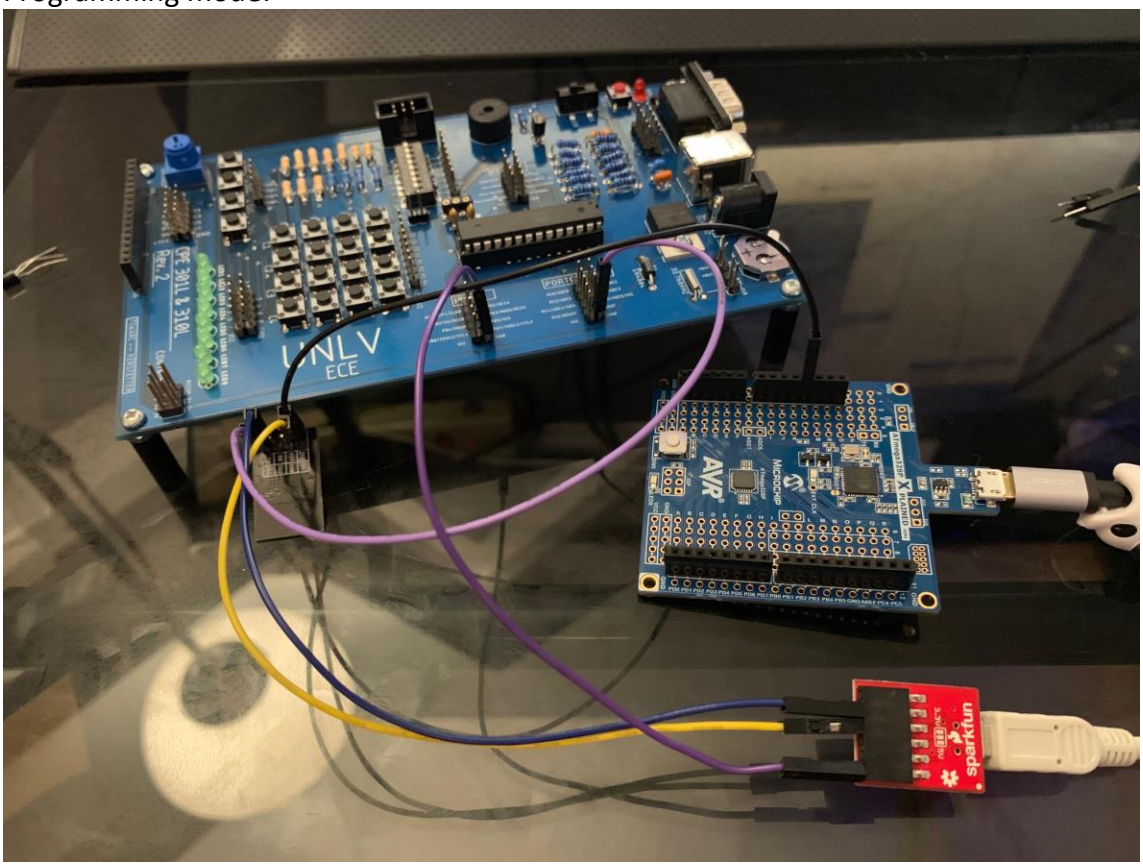
- Collect and analyze energy data
- Building the Internet of Things with the ESP8266 Wi-Fi Module and ThingSpeak
- Analyze data from a weather station
- Forecast wind-driven tide levels with an ultrasonic tide gage

Task 5:

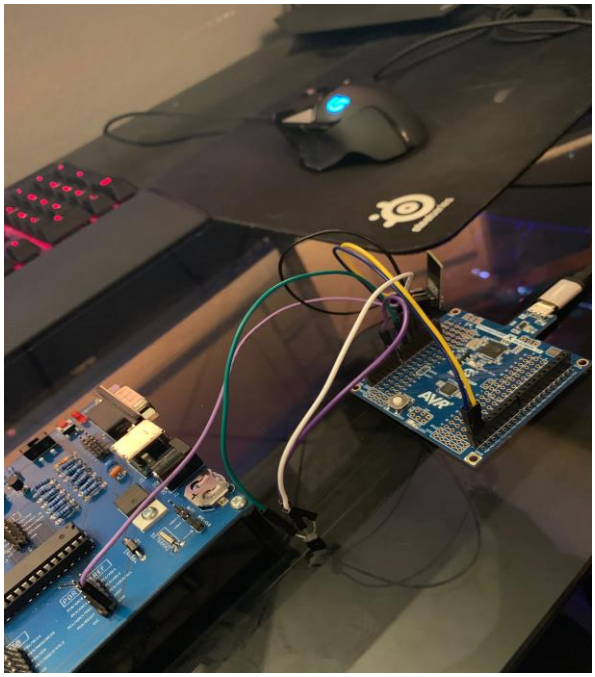


6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Programming mode:



Output Mode:



7. VIDEO LINKS OF EACH DEMO

https://youtu.be/84T_1Nl3WU4

8. GITHUB LINK OF THIS DA

https://github.com/tylergardenhire/submission_projects.git

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

"*This assignment submission is my own, original work*".
TYLER GARDENHIRE