CPE301 – SPRING 2019

Midterm1

Student Name: Kyungseo Yun

Student #: 2001091216

Student Email: yunk93@unlv.nevada.edu

Primary Github address: <https://github.com/biscuit0x/submission_yun.git>

Directory: submission\_yun/DesignAssignments/midterm1/

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 :

atmega328

jumper wires

esp8366

serial to usb module

DHT11 humidity&temperature sensor (broke my LM35 sensor)

2. C code

#define F\_CPU 16000000UL

#define BAUDRATE 115200

#define BAUD\_PRESCALAR (((F\_CPU / (BAUDRATE \* 16UL))) - 1)

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdlib.h>

#include <util/delay.h>

void set\_timer(void); //timer initialization

void adc\_init(void); //adc initialization

void read\_adc(void); //read temperature

void AT\_init(void); //esplorer set up

void USART\_init(void); //USART functions

void USART\_sendChar(char ch);

void USART\_sendString(char\* str);

volatile int adc\_temp;

int main(void)

{

USART\_init(); // initialize USART

AT\_init(); // initialize AT setttings

set\_timer(); // initialize Timer/Interrupt

adc\_init(); // initialize ADC

while (1) // Loop forever

{

}

}

ISR (TIMER1\_OVF\_vect) {

read\_adc(); //get temp

char temp[50];

snprintf(temp,sizeof(temp),"%d\r\n",adc\_temp); //read into temp buffer

USART\_sendString("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"); //connect to thingspeak

\_delay\_ms(3000);

USART\_sendString("AT+CIPSEND=50\r\n"); //send 50 characters

\_delay\_ms(1000);

USART\_sendString("GET /update?key=W7PJEYOYV6BGU372&field1="); //channel key

USART\_sendString(temp); //send temperature

\_delay\_ms(1000);

USART\_sendString("AT+CIPCLOSE\r\n"); //end

\_delay\_ms(1000);

TCNT1 = 0; //reset

}

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; //get average

}

void set\_timer(void) {

TCNT1 = 0; //reset timer1

TIMSK1 |= (1 << TOIE0); //Enable Timer1 Interrupt

sei(); //Enable Global Interrupt

TCCR1B |= (1<<CS12)|(1<<CS10); //prescaler = 1024

}

void USART\_init( void )

{

UBRR0H = 0;

UBRR0L = 8; // BAUD 115200

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); /\* 8-bit data \*/

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0); /\* Enable RX and TX \*/

}

void USART\_sendChar(char ch) {

//wait until UDR0 is empty

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

UDR0 = ch; ; //transmit ch

}

void USART\_sendString(char\* str) {

while ((\*str != '\0')) {

while (!(UCSR0A & (1<<UDRE0))); //wait until UDR0 is empty

USART\_sendChar(\*str); //transmit ch

str++;

}

}

void adc\_init(void) {

ADMUX |= (0<<REFS1)| // Reference Select

(1<<REFS0)| // Selected AVcc

(0<<ADLAR)| // Left Adjust Result OFF

(1<<MUX2) | // Analog Channel Select

(0<<MUX1) | // 1 0 1

(1<<MUX0) ; // Channel 5 or PC5

ADCSRA |= (1<<ADEN)| // Enable ADC

(0<<ADSC)| // Do not start conversion

(0<<ADATE)|// Auto Trigger Disabled

(0<<ADIF)| // Interrupt Flag Cleared

(0<<ADIE)| // Interrupt Disabled

(1<<ADPS2)| // ADC Prescaler Select

(0<<ADPS1)| // Set to

(1<<ADPS0); // 32

}

void AT\_init(void) {

USART\_sendString("AT\r\n"); // Sends AT, expect OK

\_delay\_ms(1000);

USART\_sendString("AT+CWMODE=1\r\n"); // Sends mode set to station, expect OK

\_delay\_ms(1000);

USART\_sendString("AT+CWLAP\r\n"); // Send command to list Wifi networks, expect list

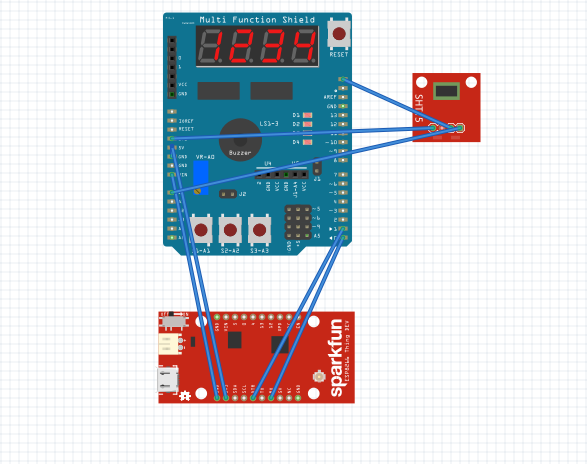
\_delay\_ms(4000);

USART\_sendString("AT+CWJAP=\"Yun\",\"st980204\"\r\n"); // Send command to join guest wifi, expect OK

\_delay\_ms(3000);

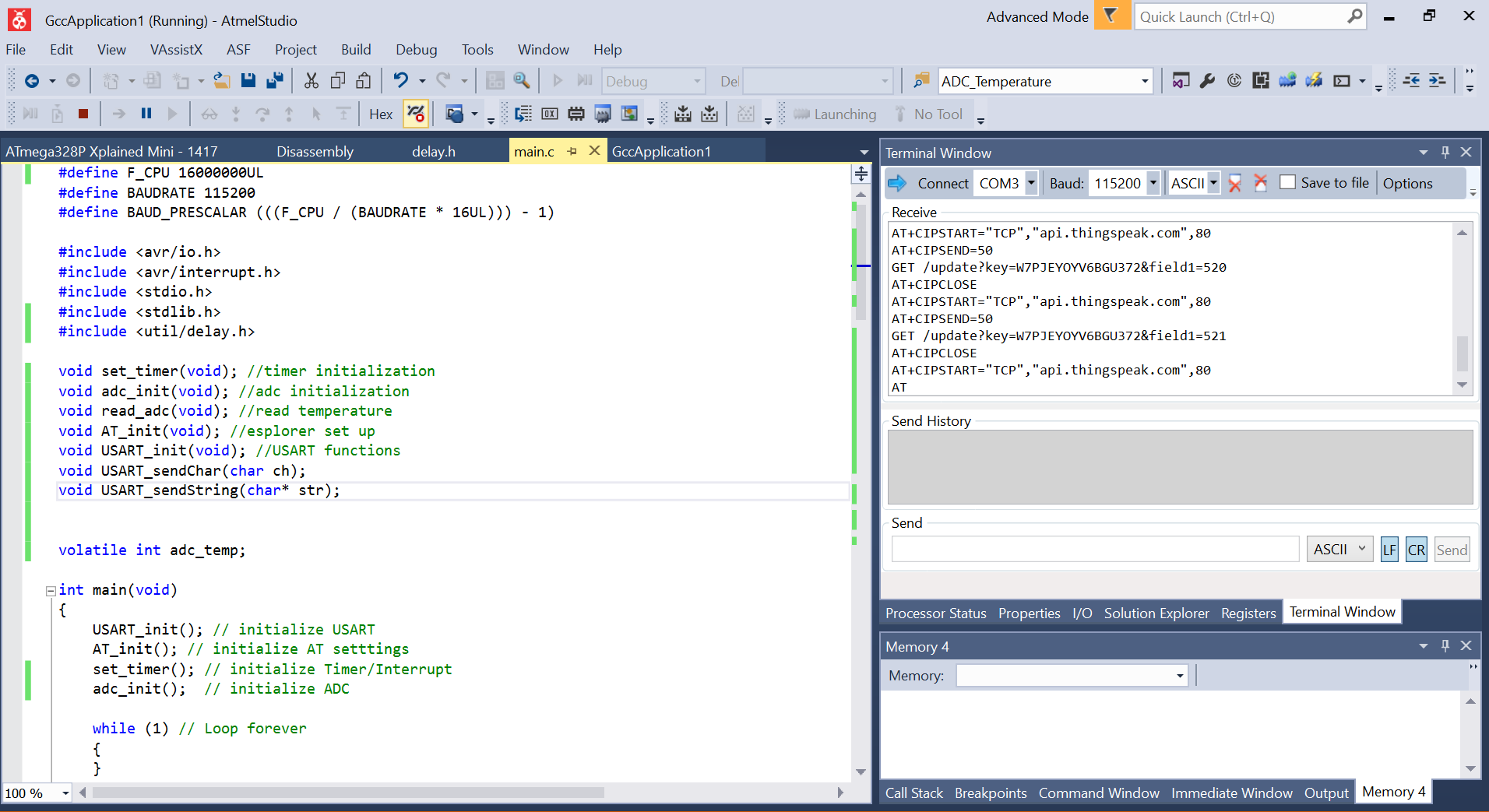
}

3. Schematic

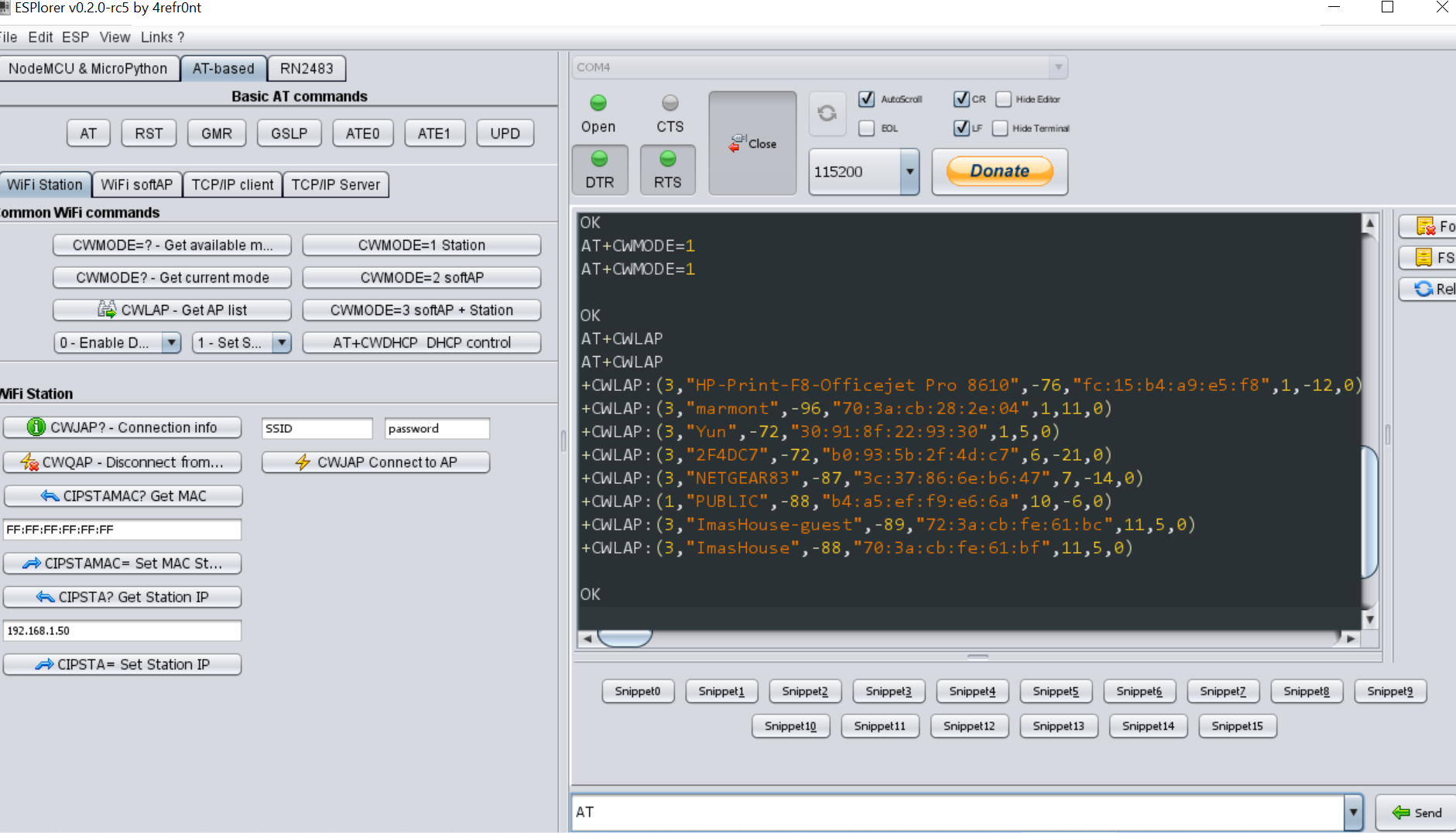


4. Screenshot

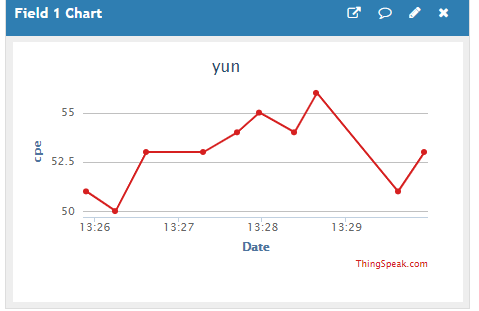
Atmel studio :

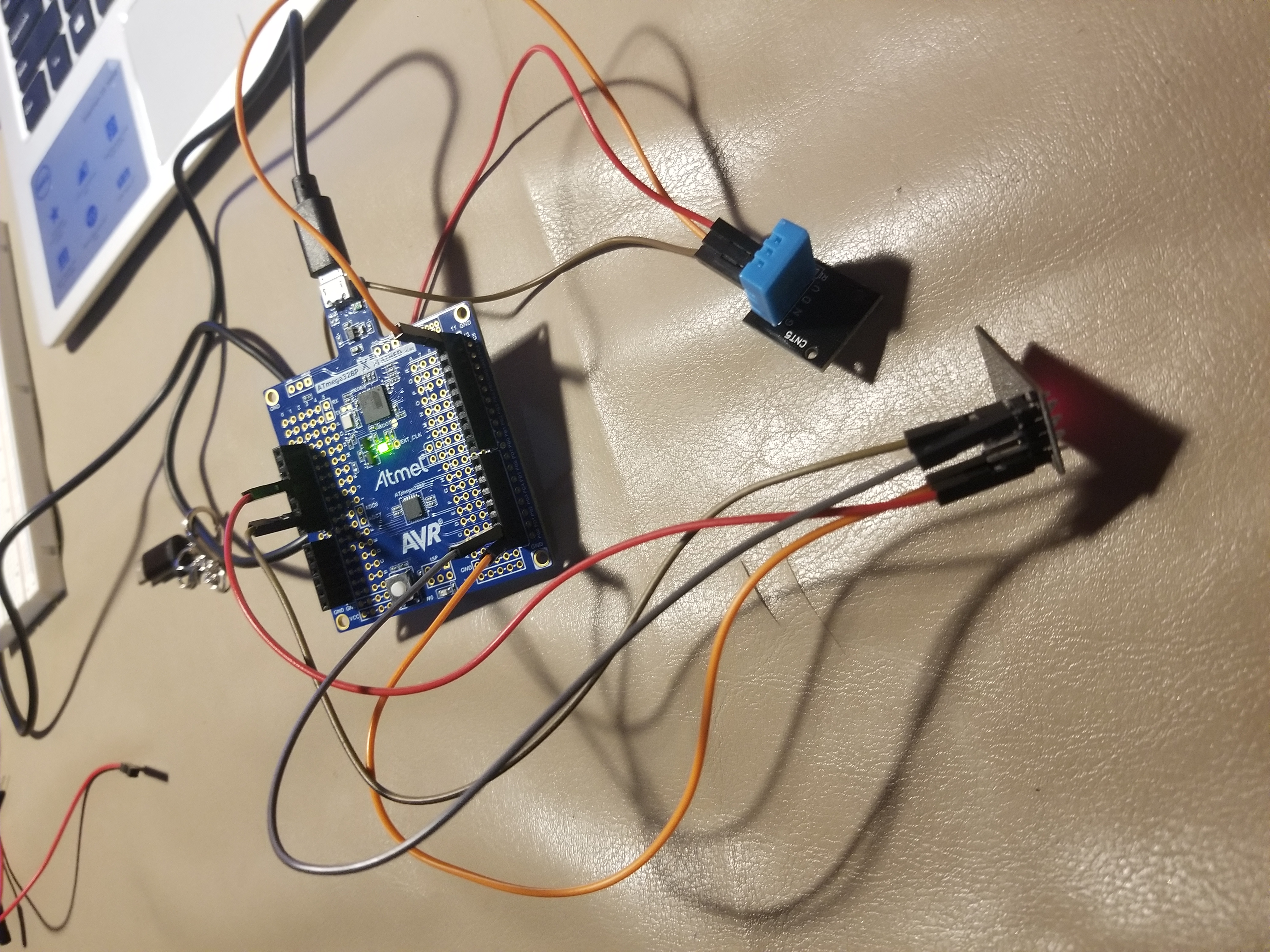


Esplore :



Thingspeak :



5. Photo / video

(broke LM35 sensor, so used DHT11 humidity&temperature sensor)

<https://youtu.be/3dKTiEkg02E>