CPE301 – SPRING 2019

Mid 2/ Final

Student Name: Robert Sander

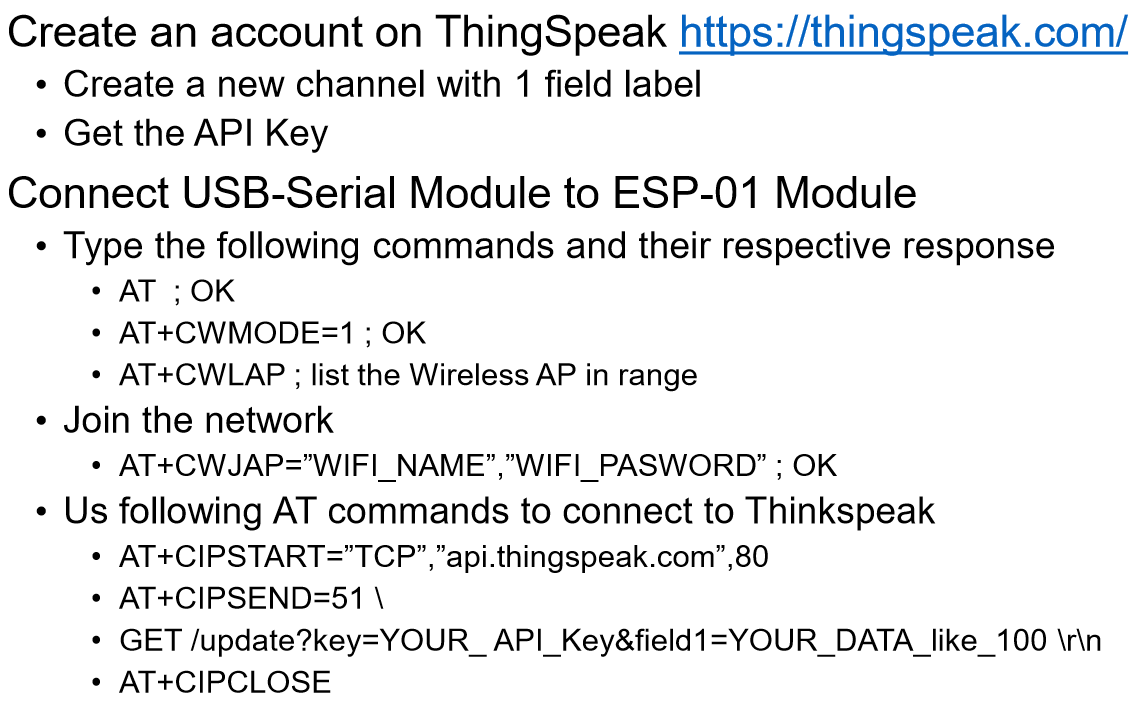
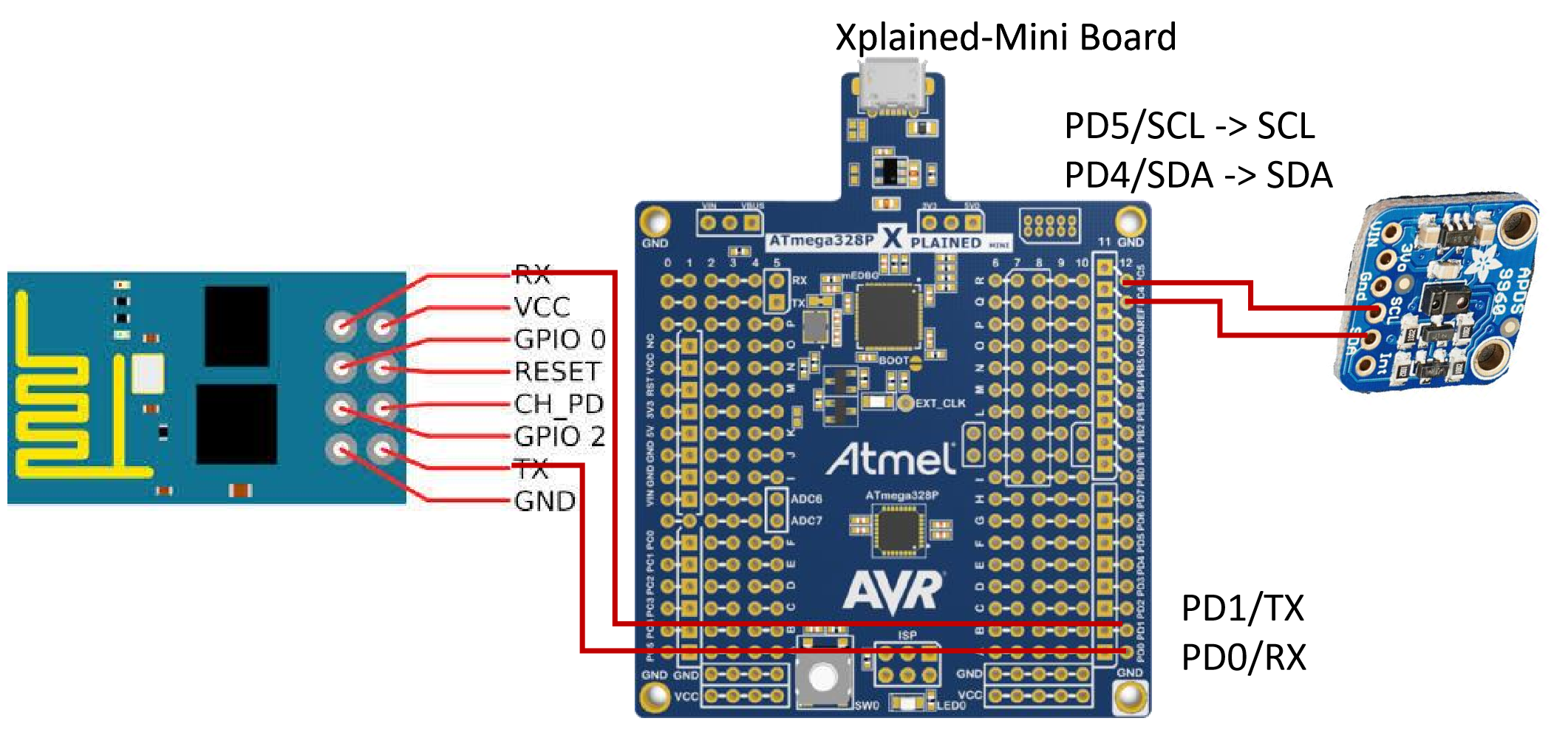
Student #: 5002102412

Student Email: sander1@unlv.nevada.edu

Primary Github address: <https://github.com/sanderUNLV/submission_DA.git>

Youtube link: <https://youtu.be/9MAIOPtxEwQ>

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**



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

// Assignment: Midterm 2/ Final

// Author : Robert Sander

/\*

\* ATmega16 Interface with MPU-6050

\* http://www.electronicwings.com

\*

\*/

#define *F\_CPU* 16000000UL /\* Define CPU clock Frequency e.g. here its 8MHz \*/

#include <avr/io.h> /\* Include AVR std. library file \*/

#include <util/delay.h> /\* Include delay header file \*/

#include <inttypes.h> /\* Include integer type header file \*/

#include <stdlib.h> /\* Include standard library file \*/

#include <stdio.h> /\* Include standard library file \*/

#include "APDS9960\_def.h" /\* Include MPU6050 register define file \*/

#include "I2C\_Master\_H\_file.h" /\* Include I2C Master header file \*/

#include "USART\_RS232\_H\_file.h" /\* Include USART header file \*/

#define APDS9960\_WRITE 0x72

#define APDS9960\_READ 0x73

float clear, red, blue, green;

void APDS9960\_Init() /\* Gyro initialization function \*/

{

*\_delay\_ms*(150); /\* Power up time >100ms \*/

//I2C\_Start\_Wait(APDS9960\_WRITE); /\* Start with device write address \*/

//I2C\_Write(APDS9960\_ENABLE); /\* Write to sample rate register \*/

//I2C\_Write(0x00); /\* 1KHz sample rate \*/

//I2C\_Stop();

//I2C\_Start\_Wait(APDS9960\_WRITE); /\* Start with device write address \*/

//I2C\_Write(APDS9960\_ENABLE); /\* Write to sample rate register \*/

//I2C\_Write(0x01); /\* 1KHz sample rate \*/

//I2C\_Stop();

I2C\_Start\_Wait(APDS9960\_WRITE);

I2C\_Write(APDS9960\_ATIME); /\* Write to power management register \*/

I2C\_Write(DEFAULT\_ATIME); /\* X axis gyroscope reference frequency \*/

I2C\_Stop();

I2C\_Start\_Wait(APDS9960\_WRITE);

I2C\_Write(APDS9960\_CONTROL); /\* Write to Configuration register \*/

I2C\_Write(DEFAULT\_AGAIN); /\* Fs = 8KHz \*/

I2C\_Stop();

//I2C\_Start\_Wait(APDS9960\_WRITE); /\* Start with device write address \*/

//I2C\_Write(APDS9960\_ENABLE); /\* Write to sample rate register \*/

//I2C\_Write(1<<POWER|1<< AMBIENT\_LIGHT); /\* 1KHz sample rate \*/

//I2C\_Stop();

I2C\_Start\_Wait(APDS9960\_WRITE); /\* Start with device write address \*/

I2C\_Write(APDS9960\_ENABLE); /\* Write to sample rate register \*/

I2C\_Write(0x02); /\* 1KHz sample rate \*/

I2C\_Stop();

}

void APDS\_Start\_Loc()

{

I2C\_Start\_Wait(APDS9960\_WRITE); /\* I2C start with device write address \*/

I2C\_Write(APDS9960\_RDATAH); /\* Write start location address from where to read \*/

I2C\_Repeated\_Start(APDS9960\_READ); /\* I2C start with device read address \*/

}

void Read\_RawValue()

{

APDS\_Start\_Loc(); /\* Read ambient light value \*/

clear = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

red = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

blue = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

green = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

I2C\_Stop();

}

int main()

{

char buffer[20], float\_[10];

float C, R, G, B;

I2C\_Init(); /\* Initialize I2C \*/

APDS9960\_Init(); /\* Initialize MPU6050 \*/

USART\_Init(9600); /\* Initialize USART with 9600 baud rate \*/

while(1)

{

USART\_SendString("cool");

Read\_RawValue();

C = clear;

R = red;

G = green;

B = blue; /\* Divide raw value by sensitivity scale factor to get real values \*/

*dtostrf*( C, 3, 2, float\_ ); /\* Take values in buffer to send all parameters over USART \*/

*sprintf*(buffer," CLEAR LIGHT = %s lumens ", float\_);

USART\_SendString(buffer);

*dtostrf*( R, 3, 2, float\_ ); /\* Take values in buffer to send all parameters over USART \*/

*sprintf*(buffer," RED LIGHT = %s lumens ", float\_);

USART\_SendString(buffer);

*dtostrf*( G, 3, 2, float\_ ); /\* Take values in buffer to send all parameters over USART \*/

*sprintf*(buffer," GREEN LIGHT = %s lumens ", float\_);

USART\_SendString(buffer);

*dtostrf*( B, 3, 2, float\_ ); /\* Take values in buffer to send all parameters over USART \*/

*sprintf*(buffer," BLUE LIGHT = %s lumens ", float\_);

USART\_SendString(buffer);

USART\_SendString("beans!");

}

}

//USART\_tx\_string("AT+CWJAP=\"ssid\",\"password\"");

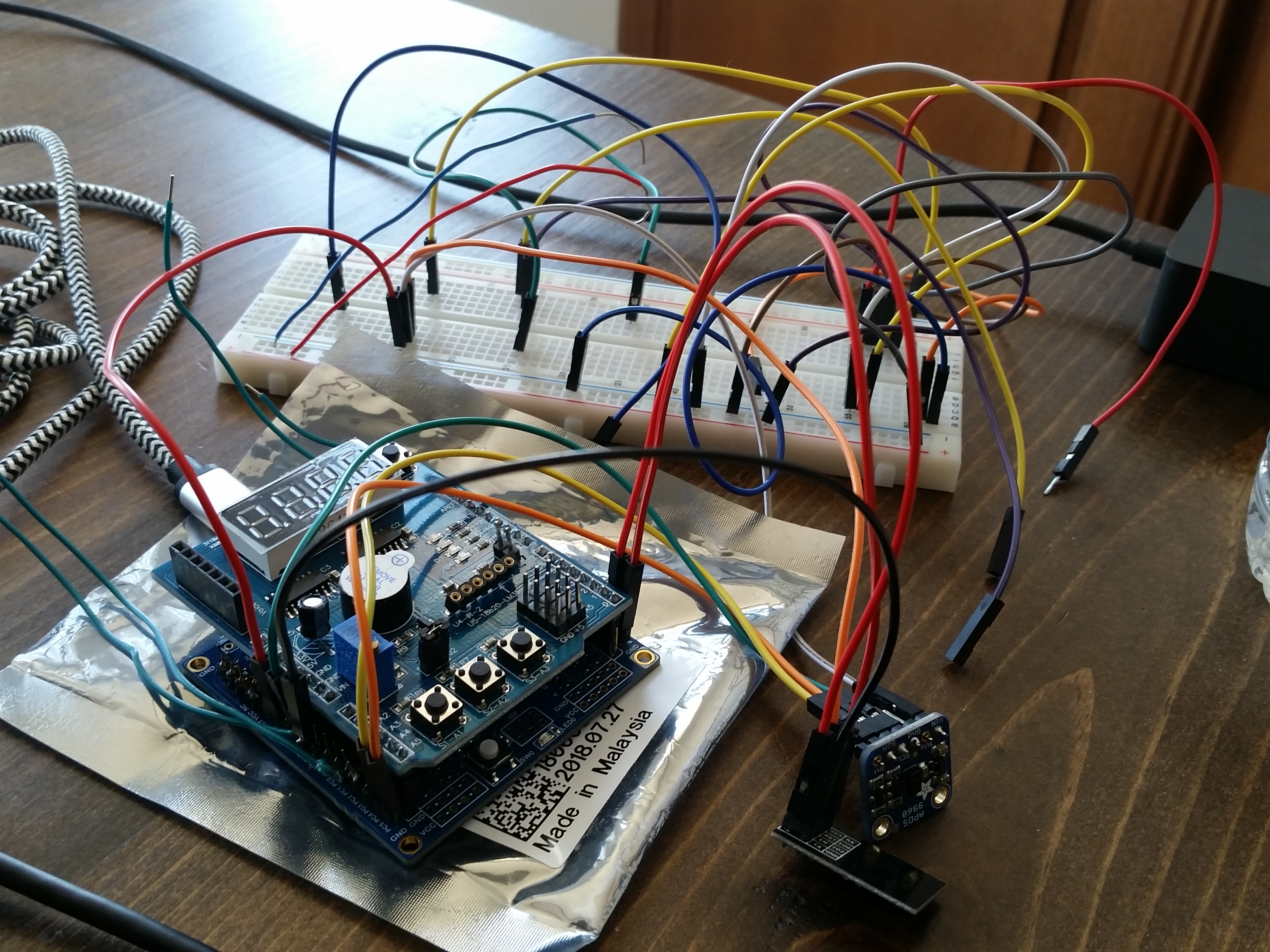
//USART\_tx\_string("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80");

//USART\_tx\_string("AT+CIPSEND=51");

//USART\_tx\_string("GET /update?key=KEY&field1=100");

//USART\_tx\_string("GET /update?key=key&field1=100");

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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/9MAIOPtxEwQ>

1. **GITHUB LINK OF THIS DA**

<https://github.com/sanderUNLV/submission_DA.git>

**Student Academic Misconduct Policy**

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

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

-Robert Sander