Lizzy's HomeHack Project

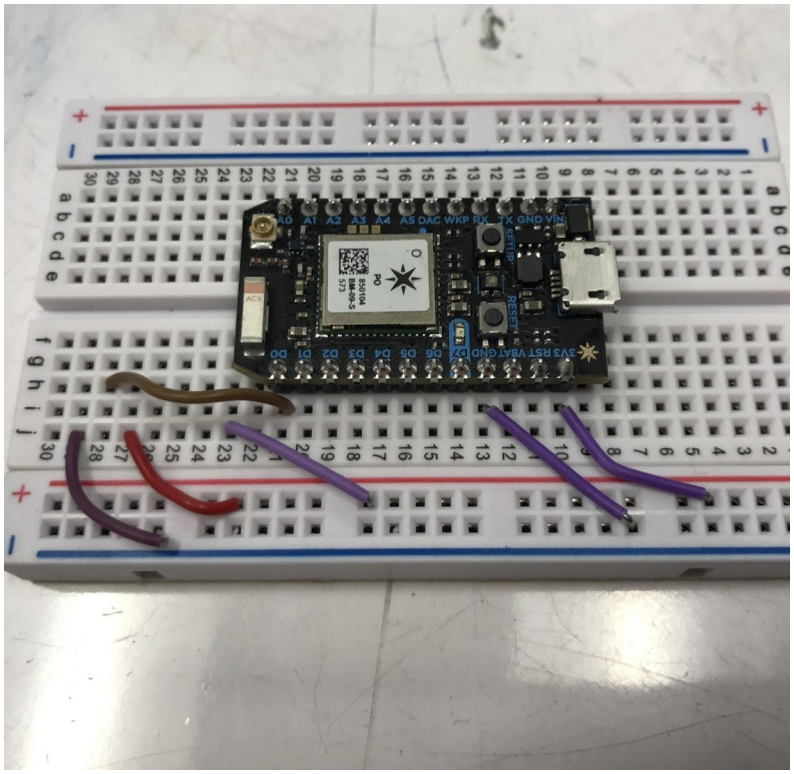
Problem Statement: My parents are tea lovers. They always desired a perfect water temperature to steep tea. They would benefit from a IoT kettle that knows when the water is ready and is able to alert them. I want to solve this problem because the appliance serves as an alert that could satisfy a lot of tea lovers.

Goal:
My goal is to connect a waterproof temperature sensor, three buttons and three LED lights onto a photon breadboard. Three buttons and three LED lights refer to the tea kinds the users need to input. Users input their choice by pressing a button and they can change their choice at anytime. When the water is ready, users get a LED alert if the water is boiled / cooled down to the right temperature for a certain kind of tea. The three tea options are 1. White/Green, 2. Olong, 3. Black/Herbal Tea.

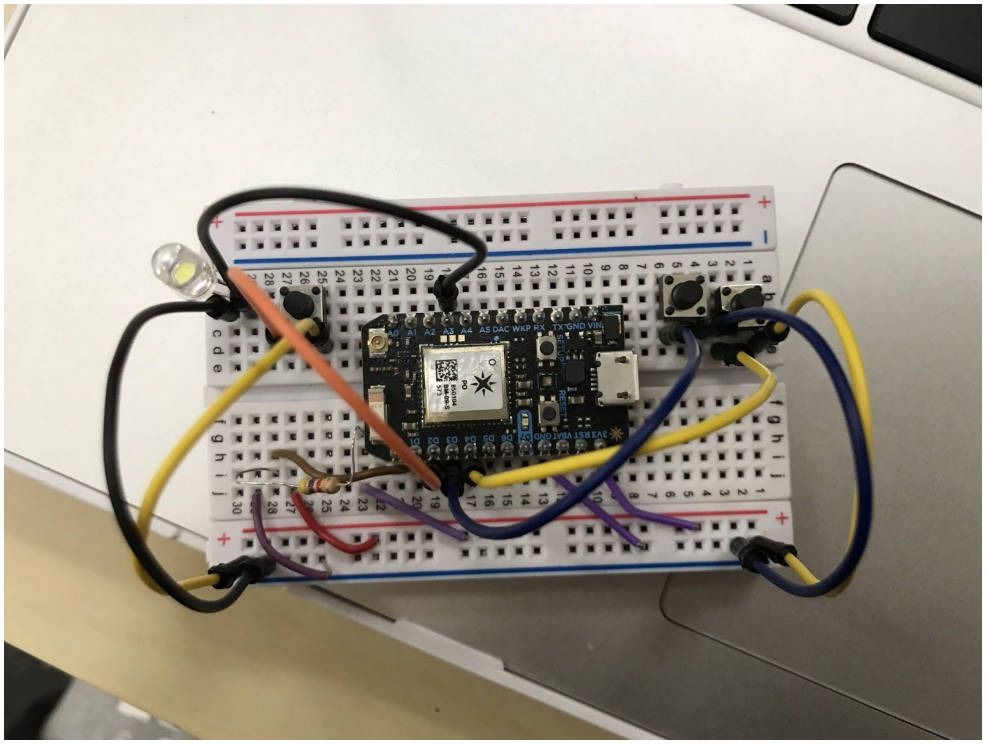
Process:

- Components used: Photon board, jumper wires, waterproof temperature sensors, 3 buttons, 3 different LED lights, resistors.

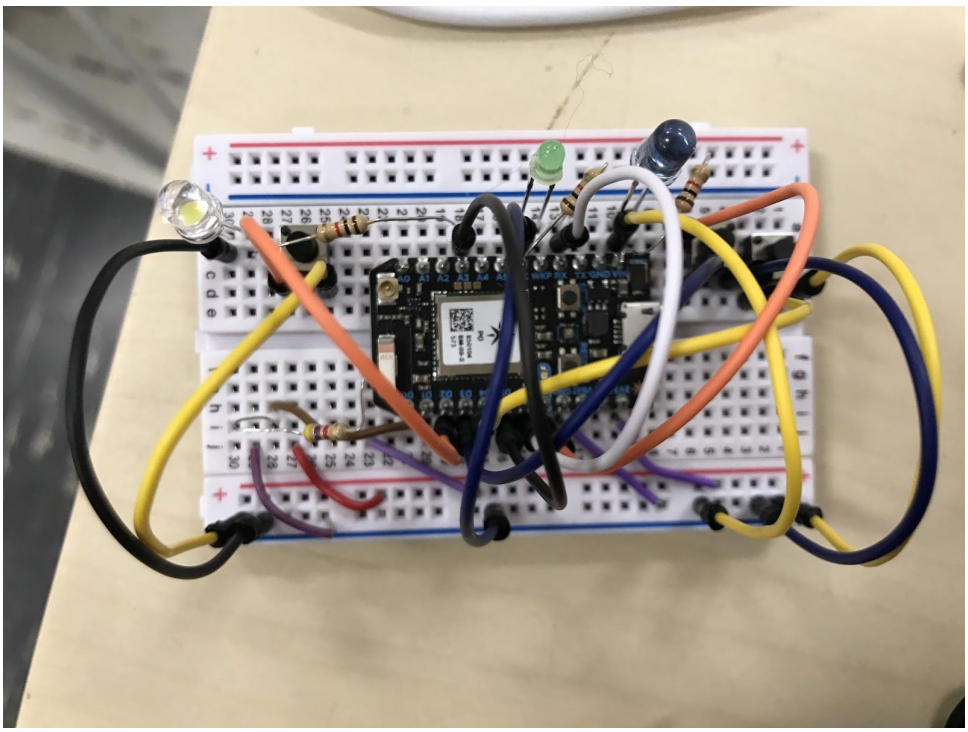
- Assembling Circuits:



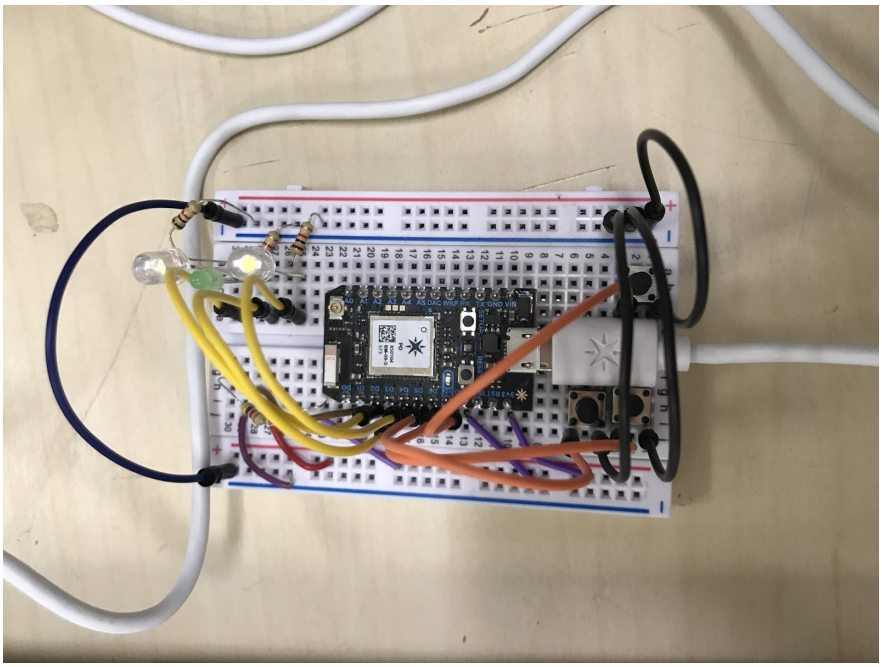
Build basic circuits based on Online Tutorial (http://diotlabs.daraghbyrne.me/3-working-with-sensors/DS18B20/)



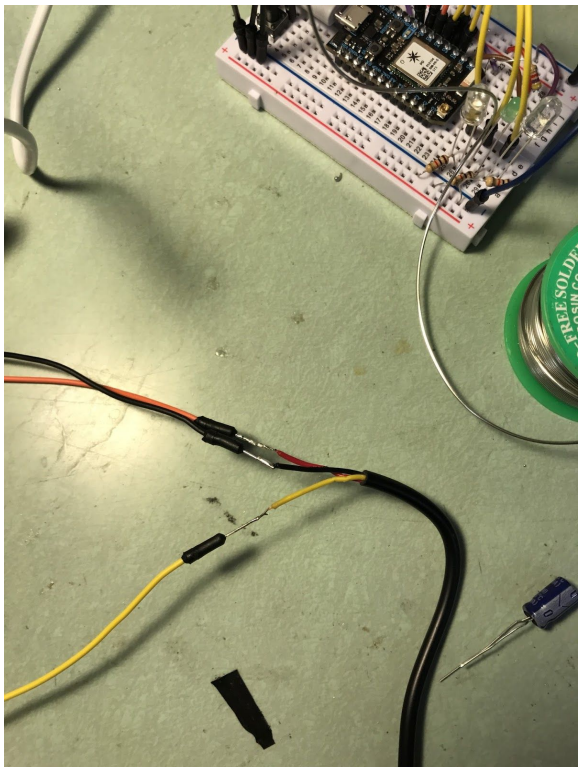
Trying one LED light.



Wiring.

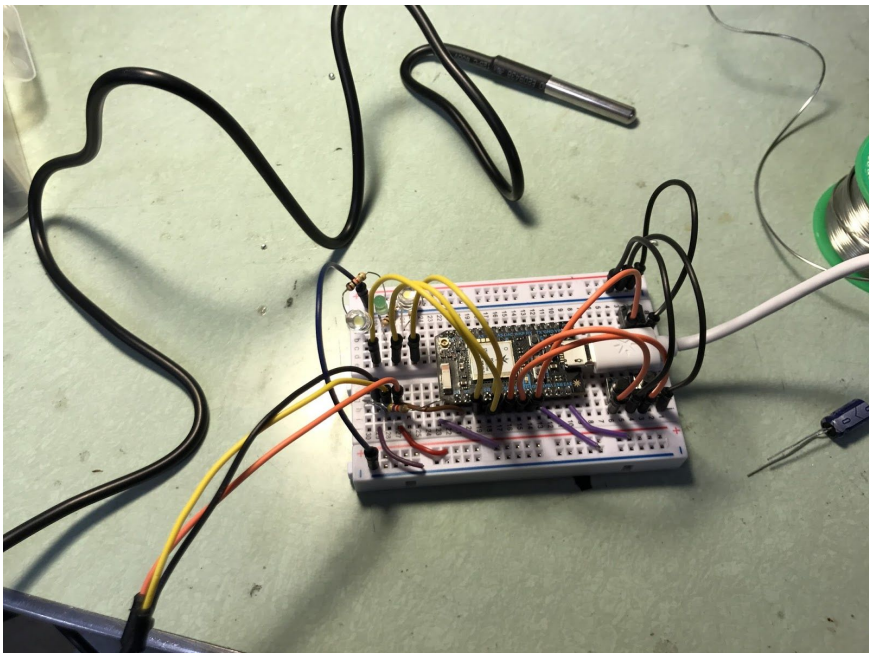


Fixing mistakes and Organizing the board with Jesse's help.



Soldering jumper wires onto the temperature sensor.

Waterproof temperature sensor

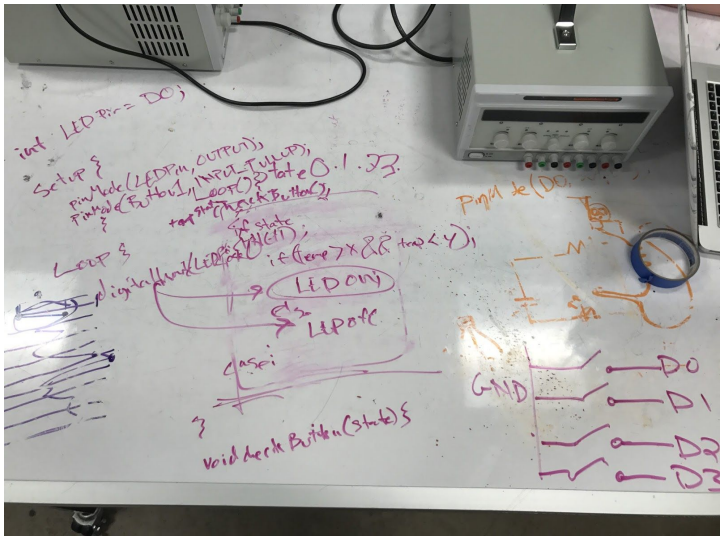


Wiring temperature sensor onto board.

- **Coding process:**

I downloaded library codes and tested the temperature sensor based on the online tutorial.
(http://diotlabs.daraghbyrne.me/3-working-with-sensors/DS18B20/**)**

First I planned my pseudocode with the help of Professor Brockmeyer.



Based on the plan, I transferred the algorithm into actual codes. I did not change the library codes for reading temperature.

TempSensor.ino

```
// This #include statement was automatically added by the Spark IDE.
#include "OneWire.h"

// This #include statement was automatically added by the Spark IDE.
#include "spark-dallas-temperature.h"

// ------------------
// Read temperature
// ------------------

// Data wire is plugged into port 0 on the Arduino
// Setup a oneWire instance to communicate with any OneWire devices (not just Ma
OneWire oneWire(D0 );

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature dallas(&oneWire);

// Create a variable that will store the temperature value
double temperature = 0.0;
double temperatureF = 0.0;
double fakeTemp = 0.0;
#include <math.h>

int LedPin1 = D2;
int LedPin2 = D3;
int LedPin3 = D4;
```

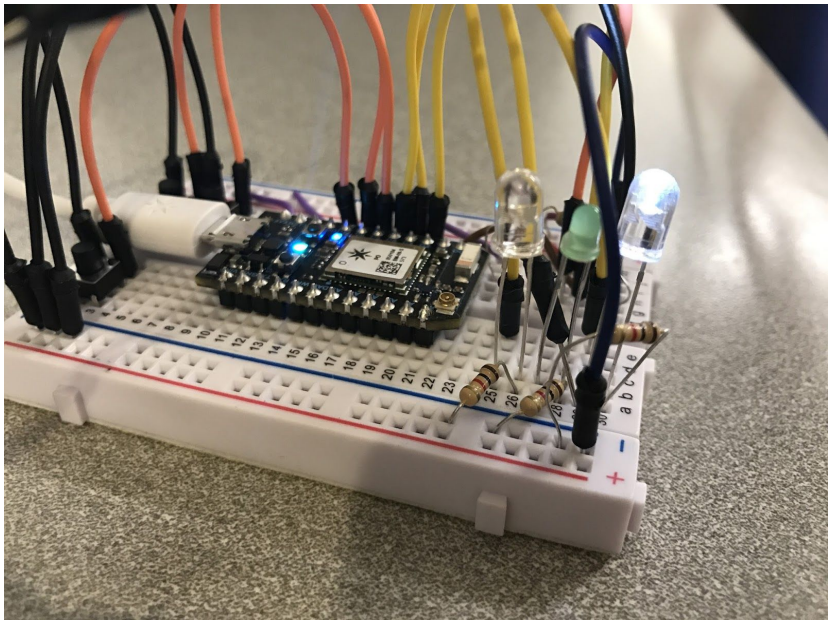
```
int Button1 = D5;
int Button2 = D6;
int Button3 = D7;
int state = 0;

int button1State = HIGH;
int button2State = HIGH;
int button3State = HIGH;

long frameCount = 0;

void setup()
{
  // Register a Particle Core variable here
  Particle.variable("temperature", &temperature, DOUBLE);
  Particle.variable("temperatureF", &temperatureF, DOUBLE);
  Particle.variable("state", state);

  Particle.function("setTemp", setTemp);

  pinMode(LedPin1, OUTPUT);
  pinMode(LedPin2, OUTPUT);
  pinMode(LedPin3, OUTPUT);
  pinMode(Button1, INPUT_PULLUP);
  pinMode(Button2, INPUT_PULLUP);
  pinMode(Button3, INPUT_PULLUP);
```

```
  button1State = digitalRead(Button1);
  button2State = digitalRead(Button2);
  button3State = digitalRead(Button3);

  if(button1State == LOW) {
    state = 1;
  }
  if(button2State == LOW) {
    state = 2;
  }
  if(button3State == LOW) {
    state = 3;
  }

  /*Serial.println(state);*/


//White/Green Tea
if (state ==1 ) {
  turnLedOff();
  if (temperatureF >= 170 && temperatureF <= 185) {
    digitalWrite(LedPin1, HIGH);
    digitalWrite(LedPin2, LOW);
    digitalWrite(LedPin3, LOW);
  }
```

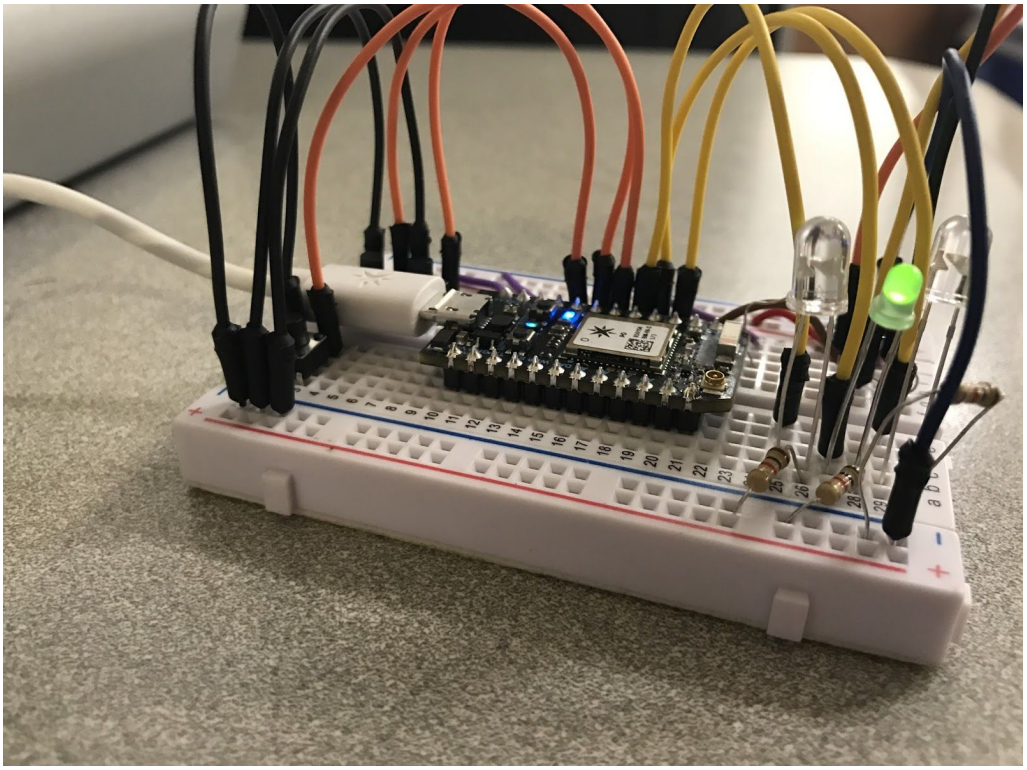
```
else {
  digitalWrite(LedPin1, LOW);
}
}

//Olong
if (state ==2){
 turnLedOff();
 if(temperatureF >= 180 && temperatureF <= 190){
   digitalWrite(LedPin2, HIGH);}

   else{
     digitalWrite(LedPin2, LOW);}}

//Black/Herbal
if (state ==3){
  turnLedOff();
  if(temperatureF >= 208 && temperatureF <= 212){
    digitalWrite(LedPin3, HIGH);
    digitalWrite(LedPin2, LOW);
    digitalWrite(LedPin1, LOW);}
   else{
     digitalWrite(LedPin3, LOW);}}
```

```
144
145  void turnLedOff(){
146    digitalWrite(LedPin3, LOW);
147    digitalWrite(LedPin2, LOW);
148    digitalWrite(LedPin1, LOW);
149
150  }
151
152  int setTemp(String input) {
153
154    fakeTemp = atoi(input);
155    return 0;
156  }
157
```

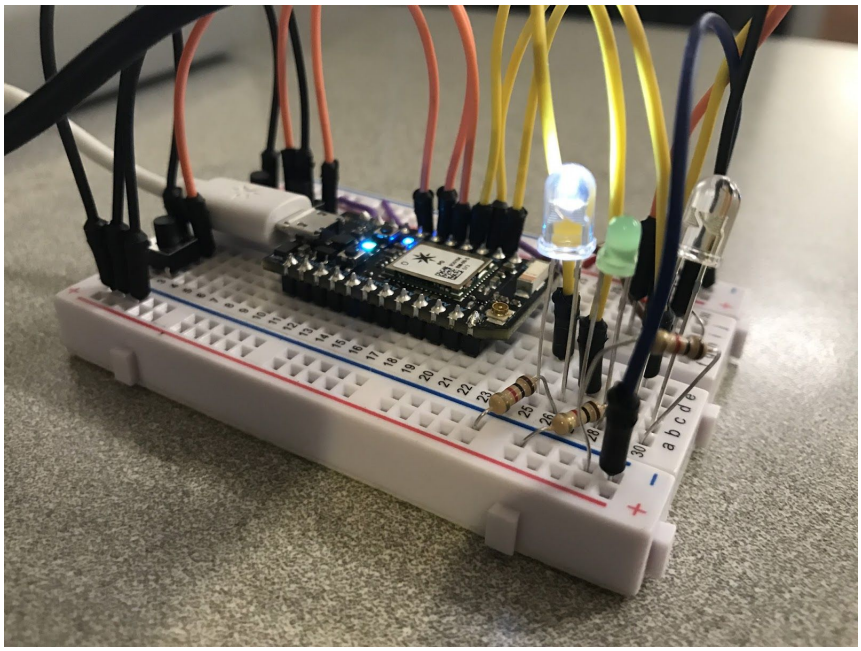
Testings:



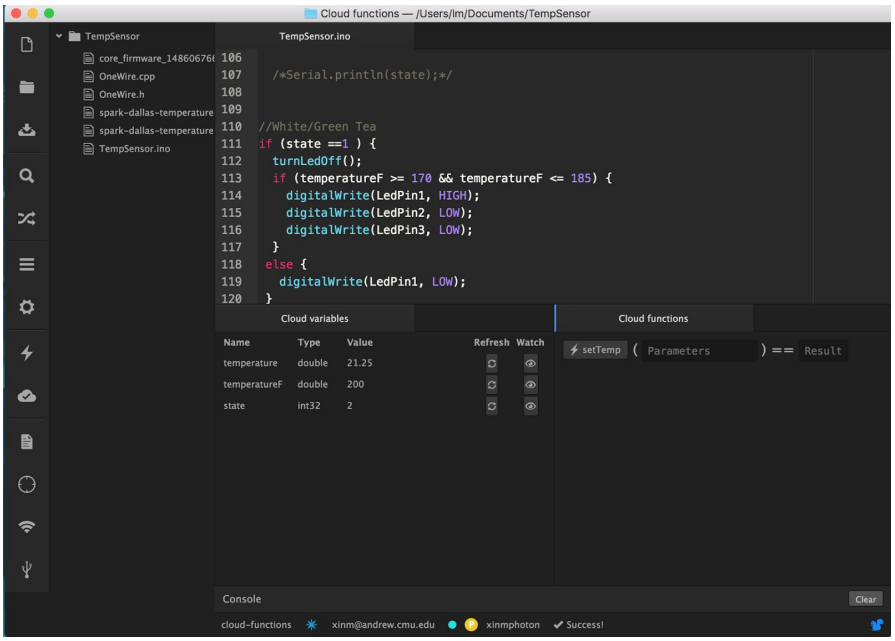
Testing LED 1: White/Green Tea



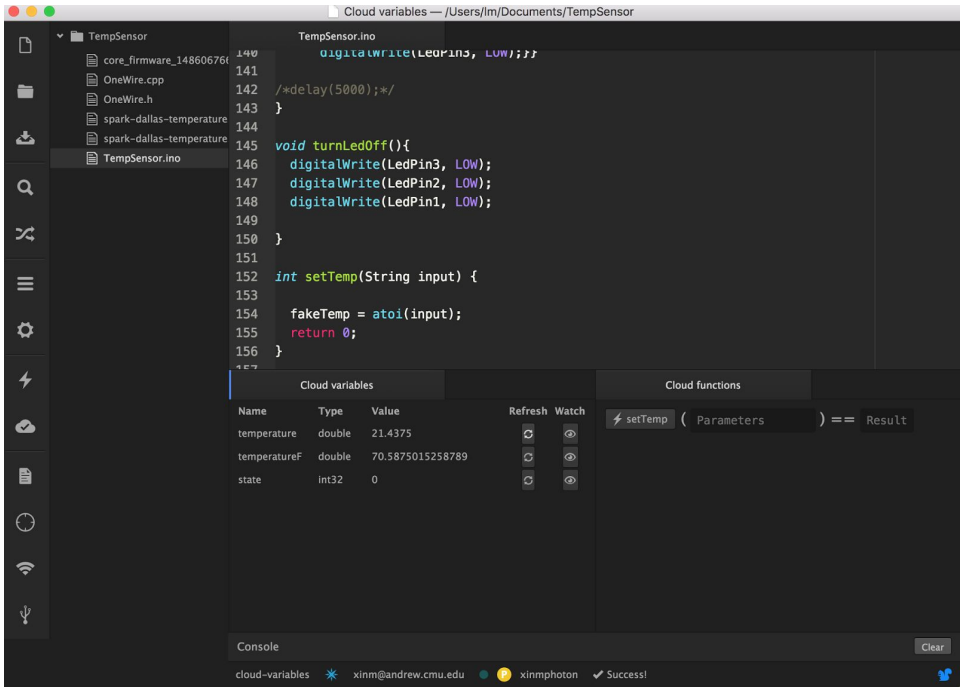
Testing LED 2: Olong Tea



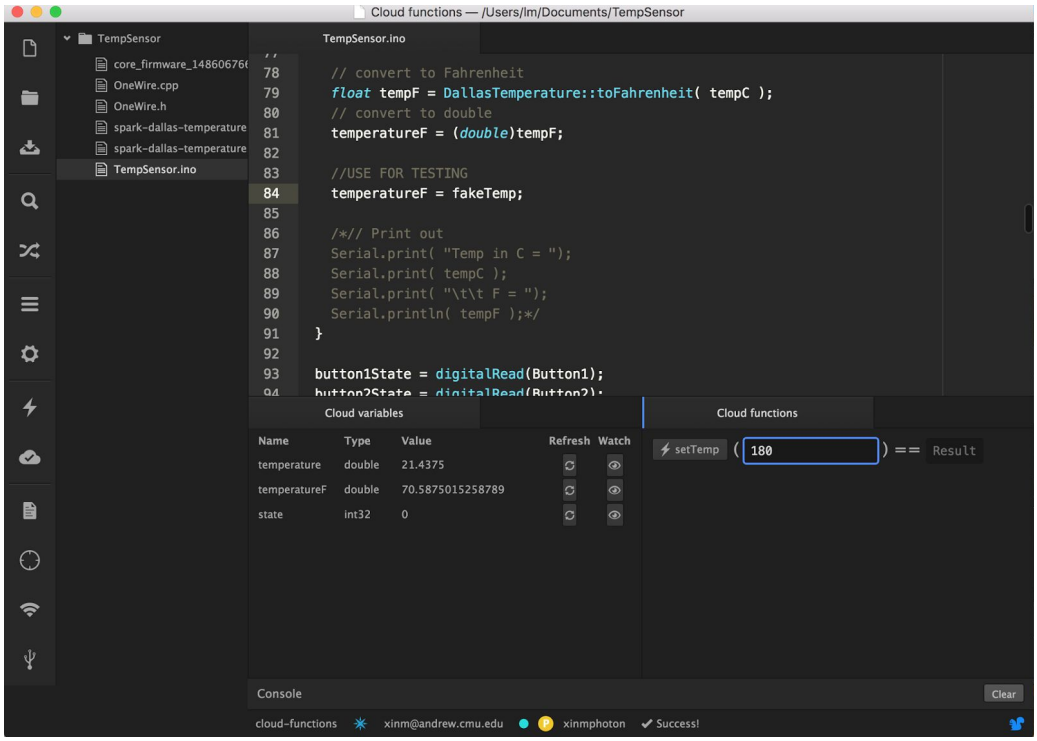
Testing LED 3: Black/Herbal Tea



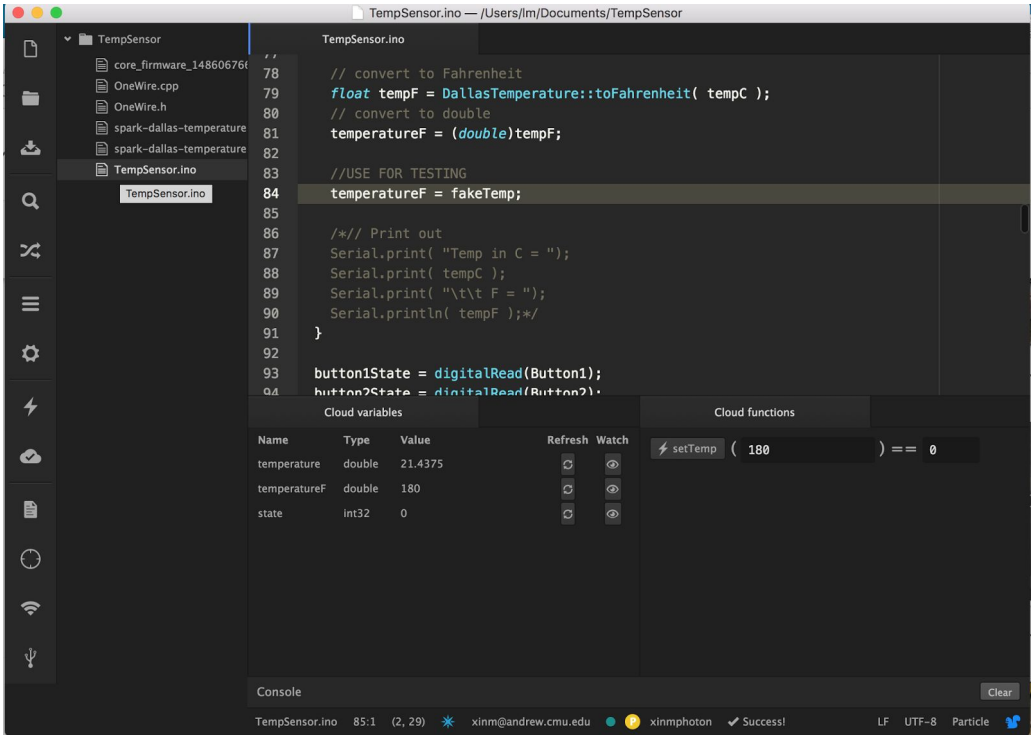
Testing buttons(as variable state 1,2,3)



Testing the accuracy of the temperature Sensor.



Using setTemp to simulate a real temperature of boiled water.



After setTemp(180), temperatureF is changed to 180. The corresponding LED is lit.

Challenges encountered in the process and help received:

I did not have any experience with circuits. So when I had the plan to connect the sensors and buttons onto the board. I did not know how they are supposed to connect to each other. I even broke a sensor due to erroneous circuiting. Jesse was very helpful that he taught me how basic circuit work and other things like organizing board. When I first wired the circuits myself, there was a short circuit issue. Jesse also helped me to pinpoint the problem in order to correct it.

For the programming part, although had experience in programming, I did not know how to use cloud functions, variables, and translate algorithm that would work correctly on a photon board. Through the learning process, I learned those skills.

Outcome: The projected satisfied my original goal that LED lights would inform the user whether the water is ready or not. Things I want to implement are:1. LED lights that can inform the water is too hot or too cold. 2. Alert message sent to phones / sound alert.

Reflection: I learned a lot from this project. I learned the basics of a circuit, soldering, basic coding on Particle Dev, organizing board and so on. I am satisfied about what I have learned; however, I wish I had practiced with the tutorials more before building on my own. In order to implement more advanced features, I need to learn how to connect this appliance to other IoT such as computers and phones, or other cookwares. I also need to know different components better in order to program them in a correct and concise manner.