

Exploring new ways to interact
with the Internet of Things.



By Kevin Rohling and Steve Arnold

Workshop Agenda

9:30 - 10:15: Intro and Setup

BREAK

10:30 - 11:15: Designing For The IoT

BREAK

11:30 - 12:15: Building A Capacitive Touch Sensor (Hardware)

BREAK

12:30 - 1:30: Writing the Touch Sensor Software



Who are these guys?



Kevin Rohling

VP of Software Products @ Emberlight

@kevinrohling

kevinr@emberlight.co



Steve Arnold

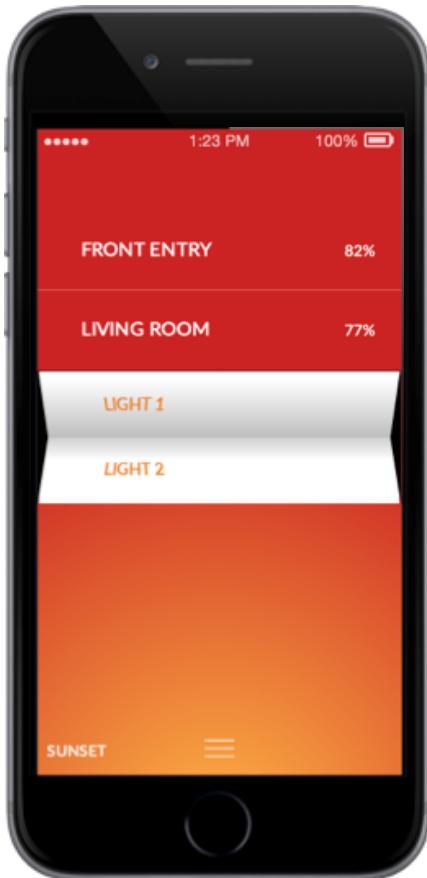
Chief Design Officer @ Emberlight

@steveonwire

steve@emberlight.co



We help make Emberlight.





Workshop Description

Our goal is to develop a new way of interacting with smart devices.



Workshop Description

To do this, we'll learn how to **turn any surface** into a **touch sensor** using Capacitive Touch.



Workshop Description

We will build our touch sensor using a **Spark Core** and **aluminum foil**.
And, we'll use the Emberlight API to control our lights.

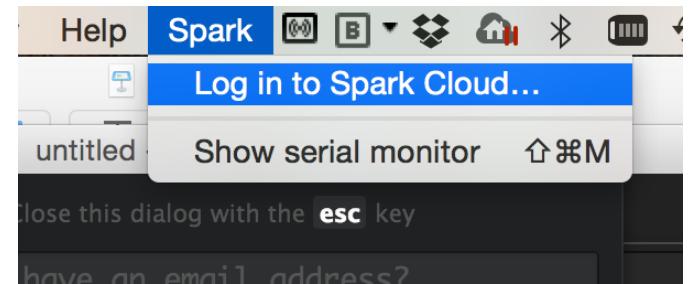


Environment Setup Part 1

GOAL: Setup Spark Dev

STEP 1: Visit <https://github.com/emberlight/mce15-workshop> and follow the steps in the README.

STEP 2: Log into Spark Dev: *Spark > Log in to Spark Cloud* using the username and password at your workstation.



Environment Setup Part 2

GOAL: Be able to upload code to your Spark Core



STEP 4: Plug the Spark Core into one of your USB Ports

STEP 5: Select a Device from the Menu: *Spark > Select device*

STEP 6: Press the  button to upload the code to your Spark Core.



Exploring new ways to interact
with the Internet of Things.

BREAK: 10:15-10:30





Designing For The IoT



Exploring new ways to interact
with the Internet of Things.

BREAK: 11:15-11:30





Building A Capacitive Touch Sensor



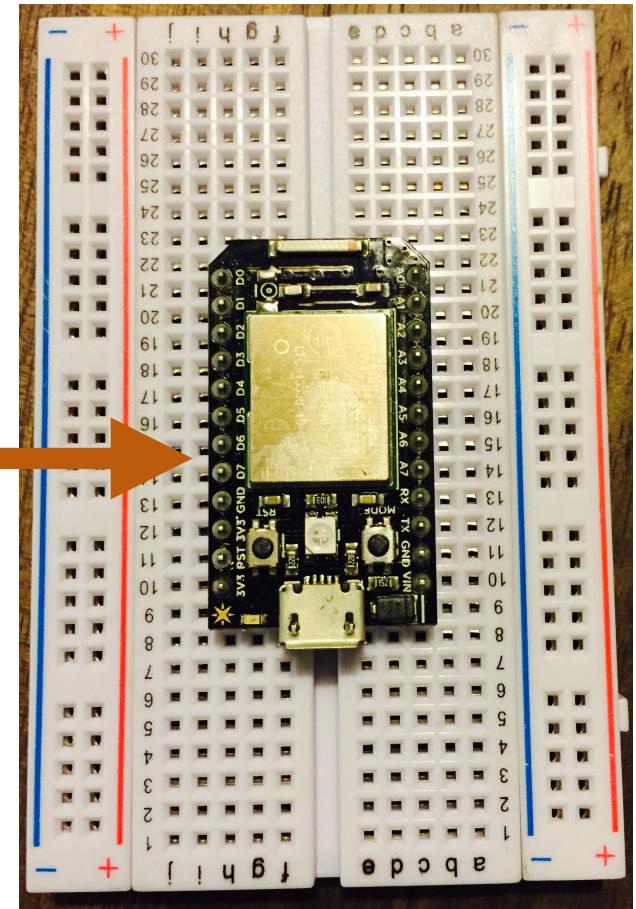


emberlight



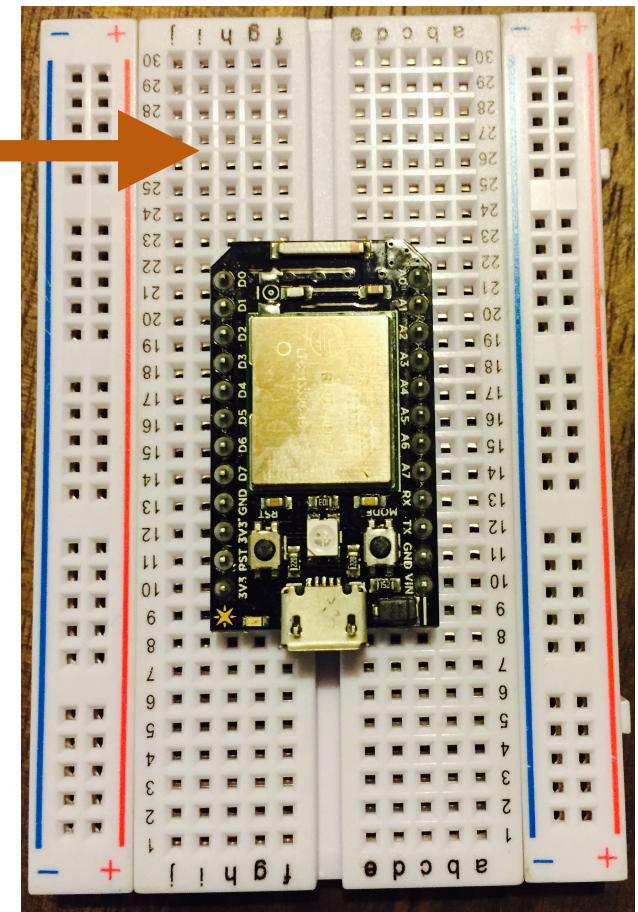
Breadboarding Intro

This thing in the middle
here is your Spark Core.



Breadboarding Intro

This big white thing with all the holes is your breadboard





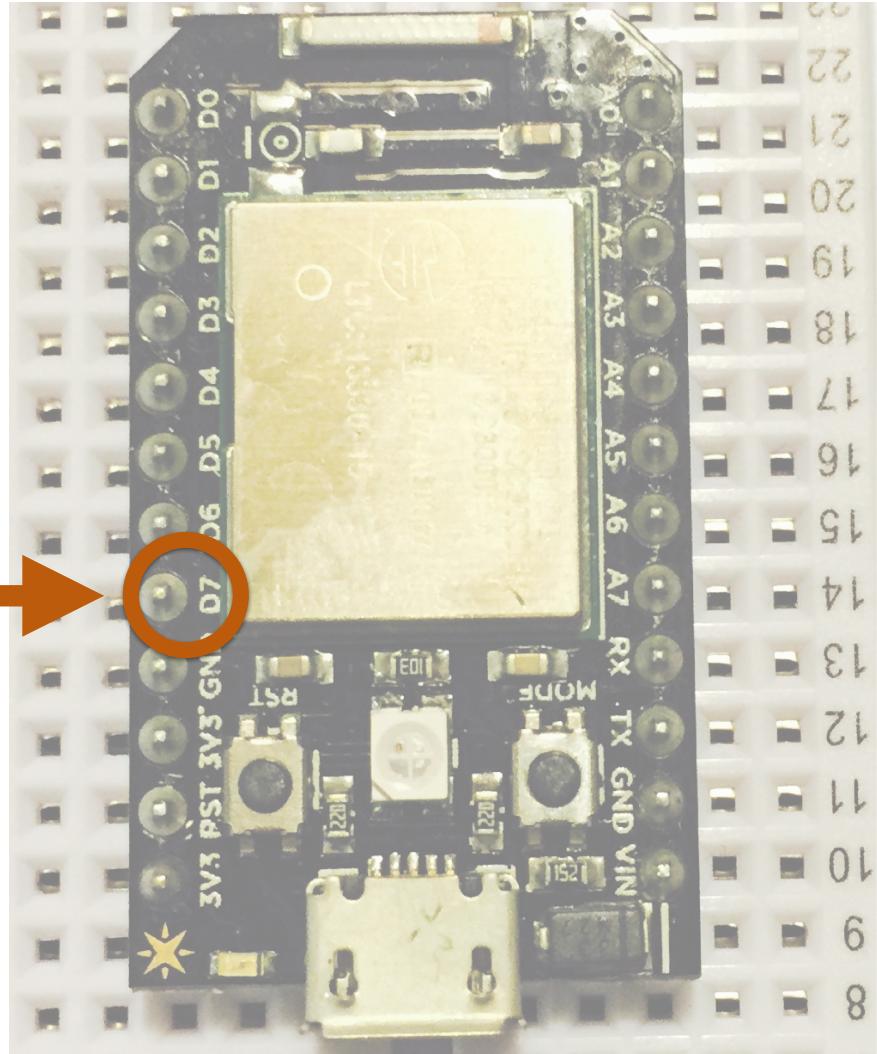
emberlight



Breadboarding Intro

To connect something to the Spark Core first find the pin label.

This, for example,
is the **D7** pin.





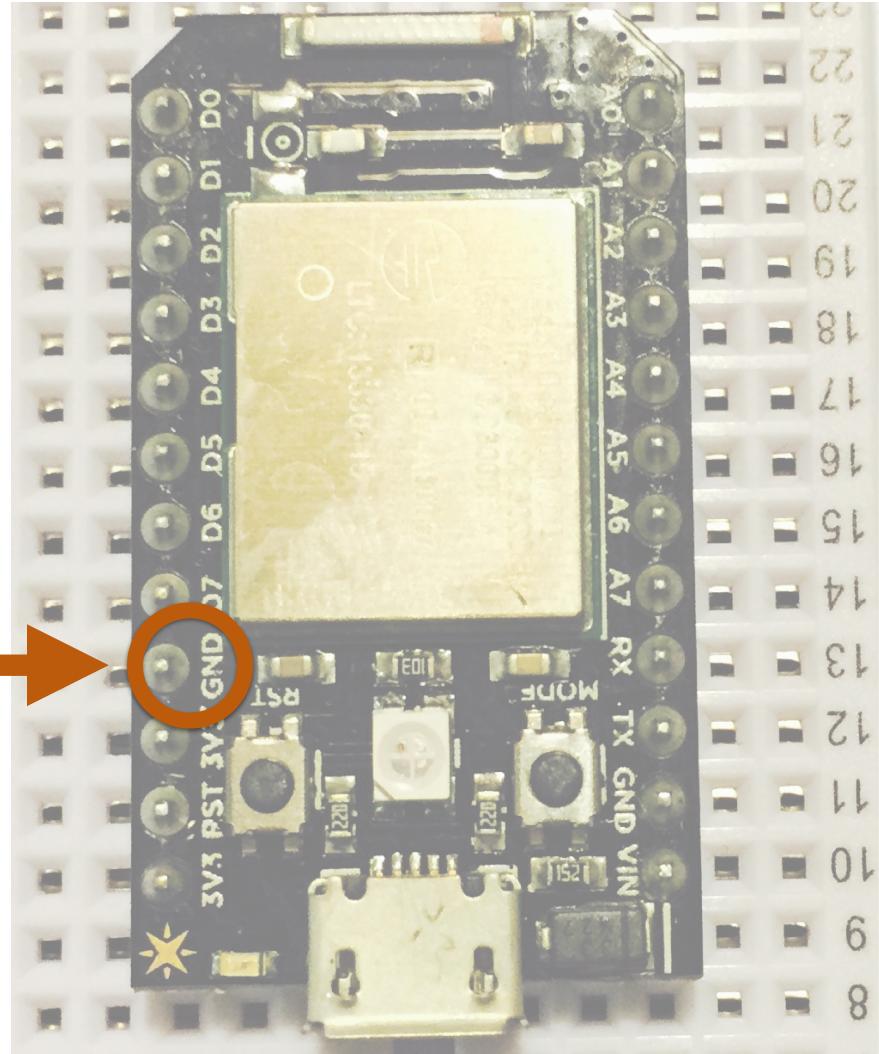
emberlight



Breadboarding Intro

To connect something to the Spark Core first find the pin label.

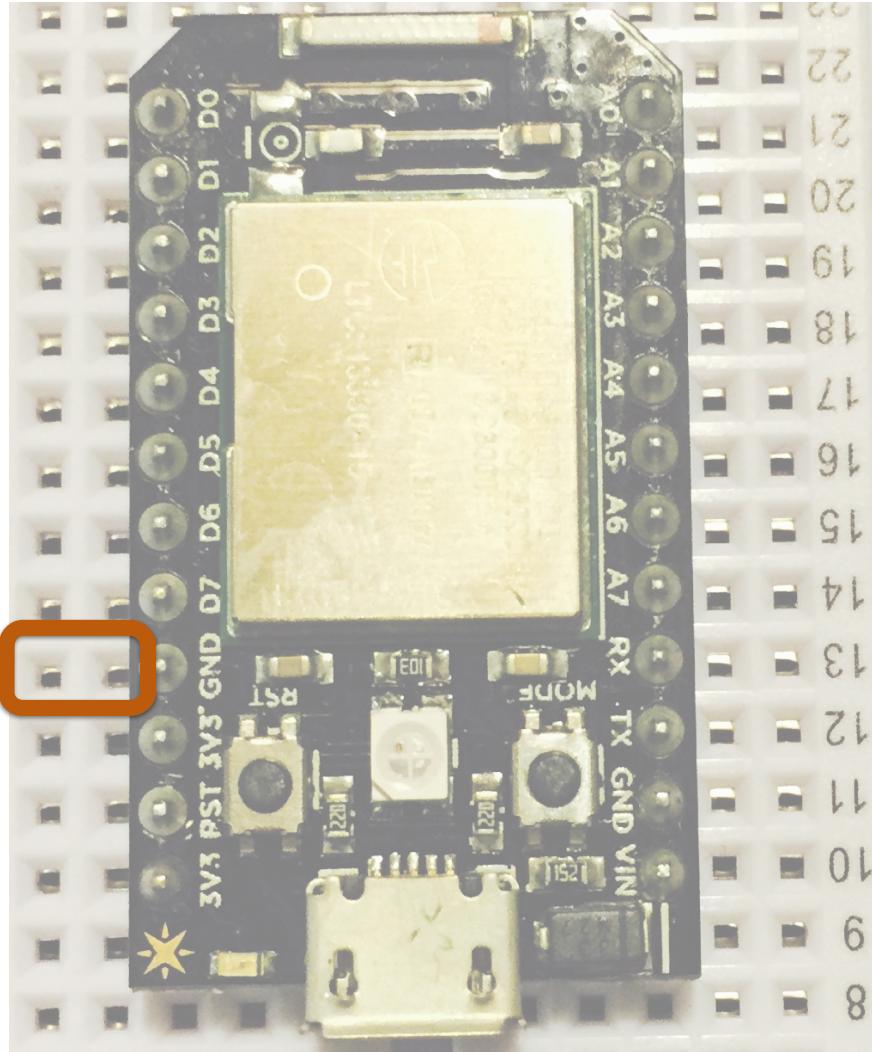
And this is
the **GND** pin.



Breadboarding Intro

To connect something to a pin, insert it into the breadboard holes next to the pin.

For example, to connect something to the **GND** pin you can use either of these two holes.

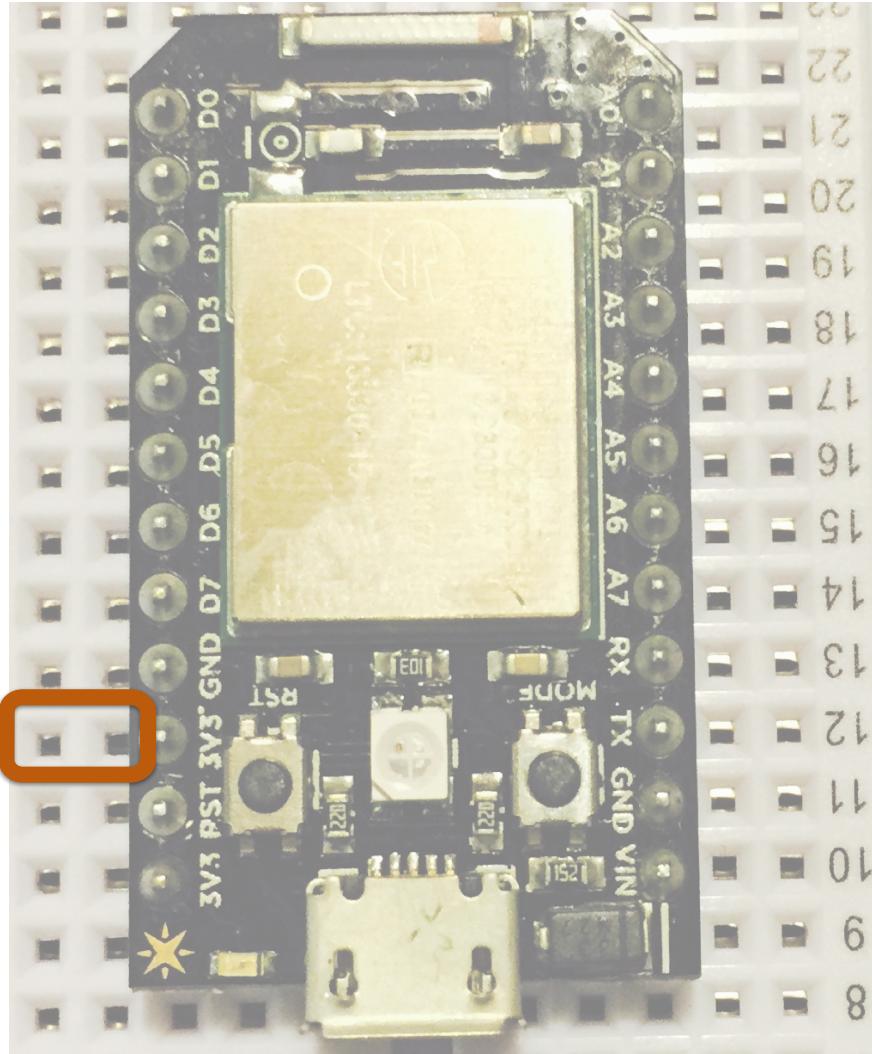




Breadboarding Intro

To connect something to a pin, insert it into the breadboard holes next to the pin.

Or to connect something to the **3V3** pin you can use either of these two holes.



Step 1: Make your touch sensor.

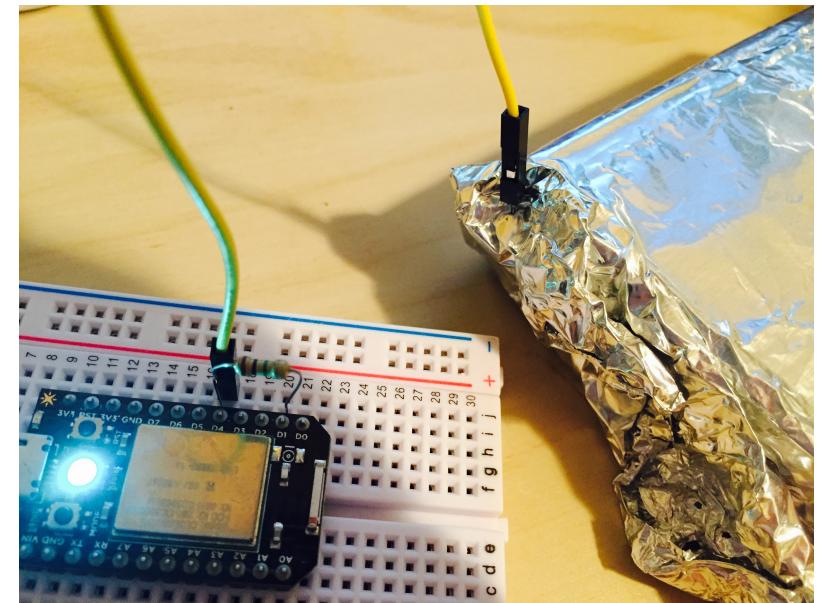
This is quite easy. Just wrap something in aluminum foil.





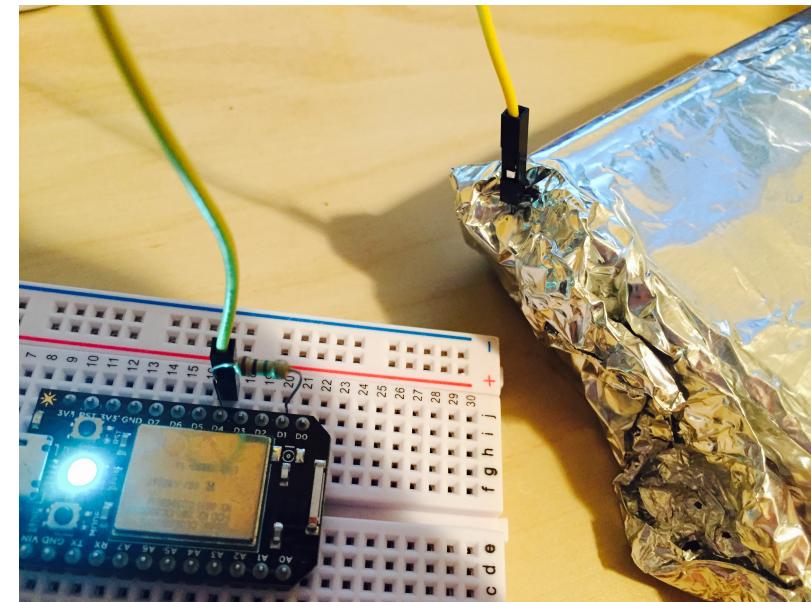
Step 2: Connect your touch sensor.

Connect a wire from D4 to your touch sensor.



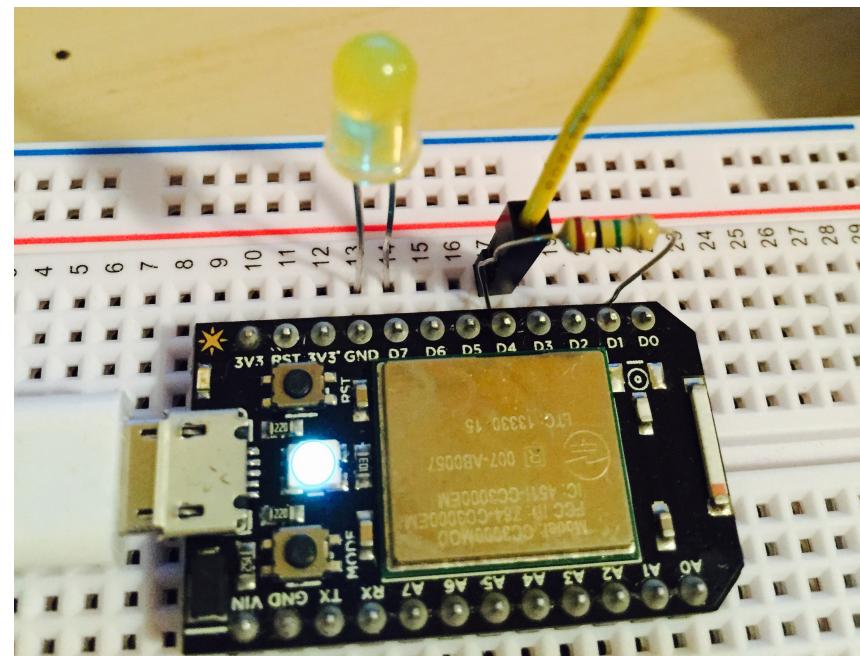
Step 3: Connect the resistor.

Connect the Resistor to pins D1 and D4 on the Spark Core.



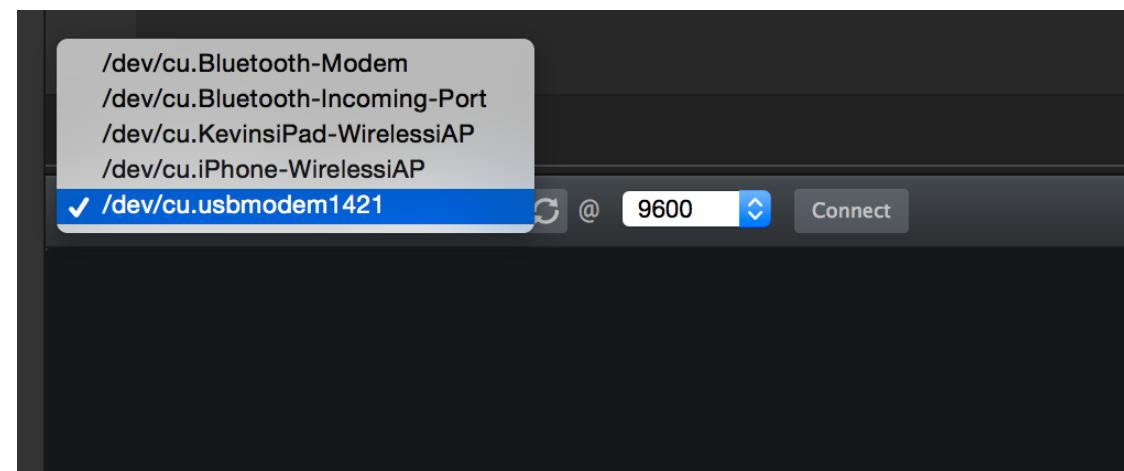
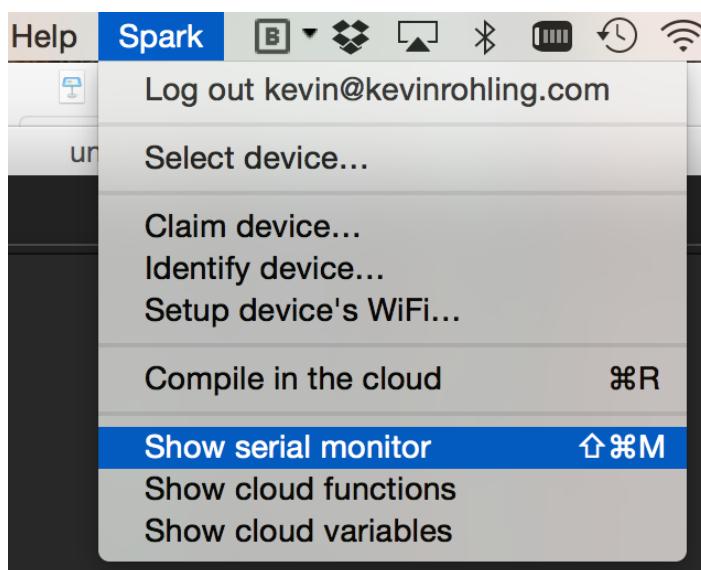
Step 4: Connect the LED.

Connect the long end of the LED to the D7 pin and the short end to GND. The LED should turn on when you plug it in.



Step 5: Open the Spark Serial Monitor

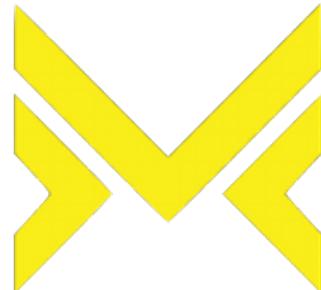
Open *Menu: Spark > Show serial monitor*, and connect to the Spark Core
You should see a stream of numbers coming in.





Exploring new ways to interact
with the Internet of Things.

BREAK: 12:15-12:30

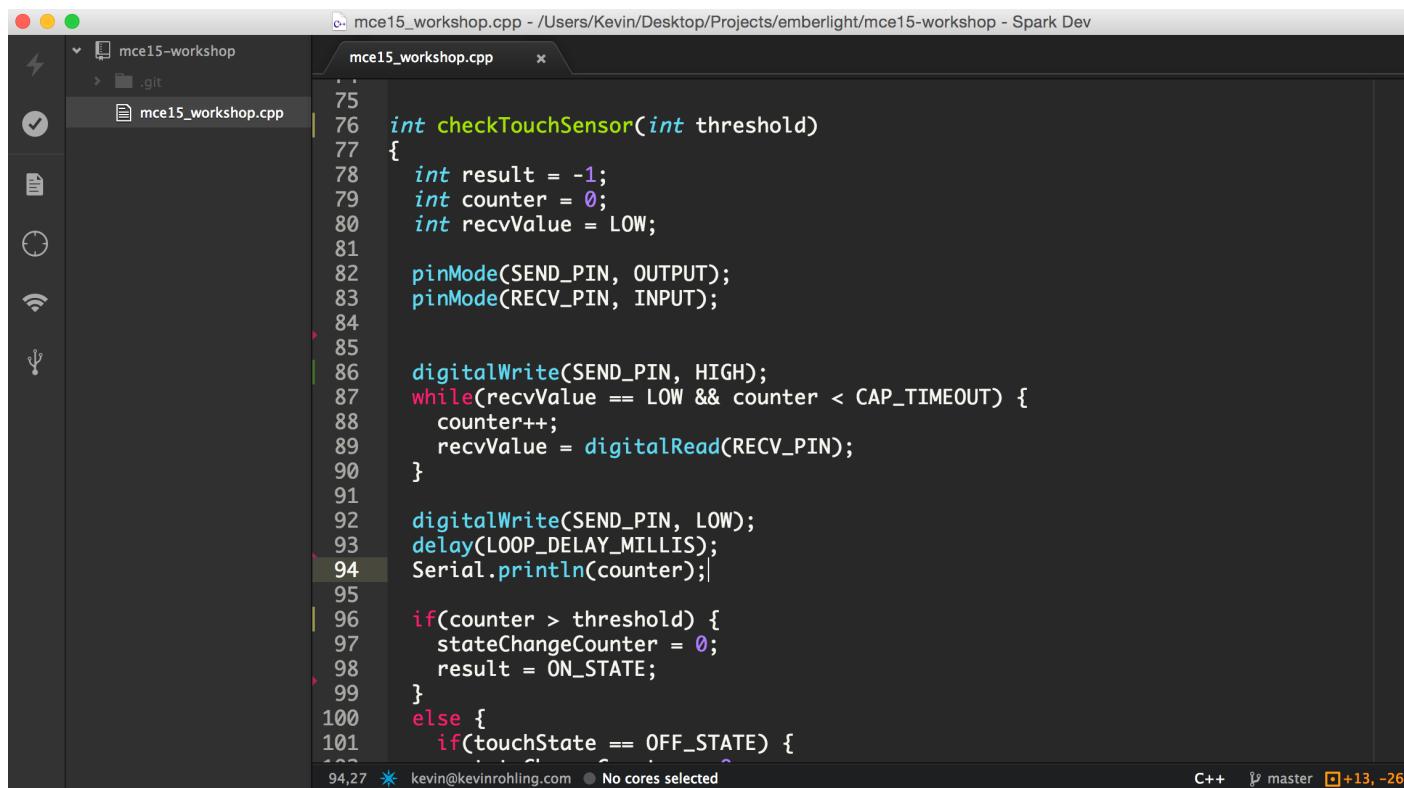




Writing the Touch Sensor Software



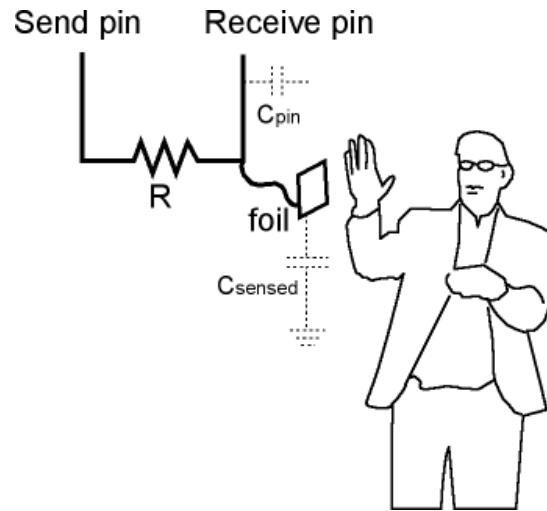
Step 1: Open the *mce_workshop.cpp* file in Spark Dev and find the *checkTouchSensor()* method.



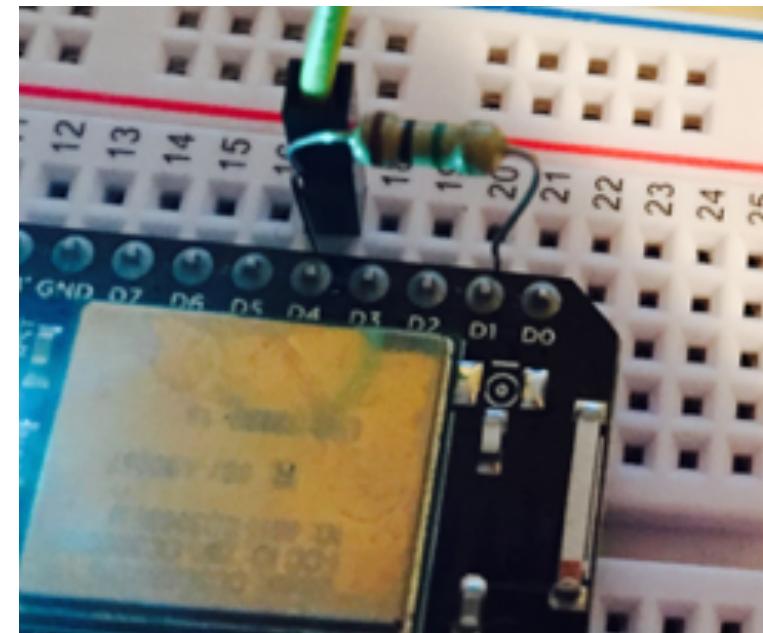
```
75
76 int checkTouchSensor(int threshold)
77 {
78     int result = -1;
79     int counter = 0;
80     int recvValue = LOW;
81
82     pinMode(SEND_PIN, OUTPUT);
83     pinMode(RECV_PIN, INPUT);
84
85
86     digitalWrite(SEND_PIN, HIGH);
87     while(recvValue == LOW && counter < CAP_TIMEOUT) {
88         counter++;
89         recvValue = digitalRead(RECV_PIN);
90     }
91
92     digitalWrite(SEND_PIN, LOW);
93     delay(LOOP_DELAY_MILLIS);
94     Serial.println(counter);
95
96     if(counter > threshold) {
97         stateChangeCounter = 0;
98         result = ON_STATE;
99     }
100    else {
101        if(touchState == OFF_STATE) {
```

94,27 kevin@kevinrohling.com No cores selected C++ master +13, -26

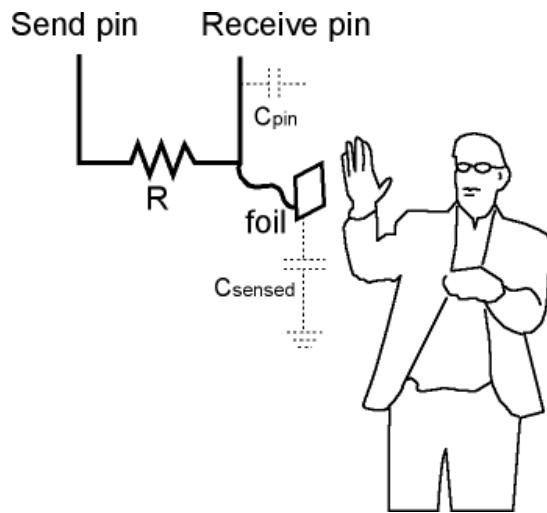
How does the software detect a touch?



The “SEND PIN” (D1) is directly connected to the “RECEIVE PIN” (D4) using a large resistor ($\sim 1M\Omega$).



How does the software detect a touch?

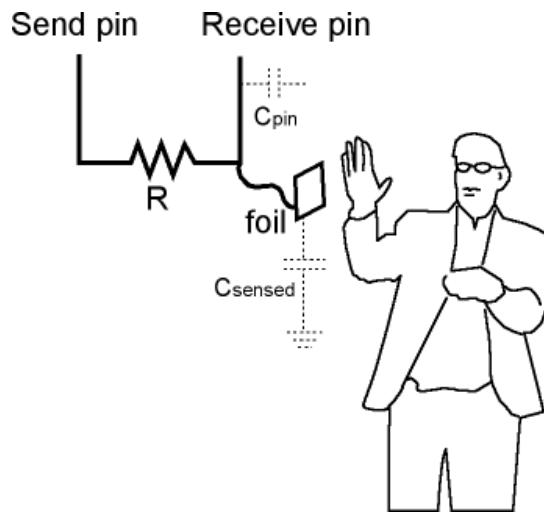


To determine if the sensor is being touched, our software will increase the voltage on the SEND PIN.

Then we measure how long it takes for the RECEIVE PIN to see the increase in voltage.

```
digitalWrite(SEND_PIN, HIGH);
while(recvValue == LOW && counter < CAP_TIMEOUT) {
    counter++;
    recvValue = digitalRead(RECV_PIN);
}
```

How does the software detect a touch?



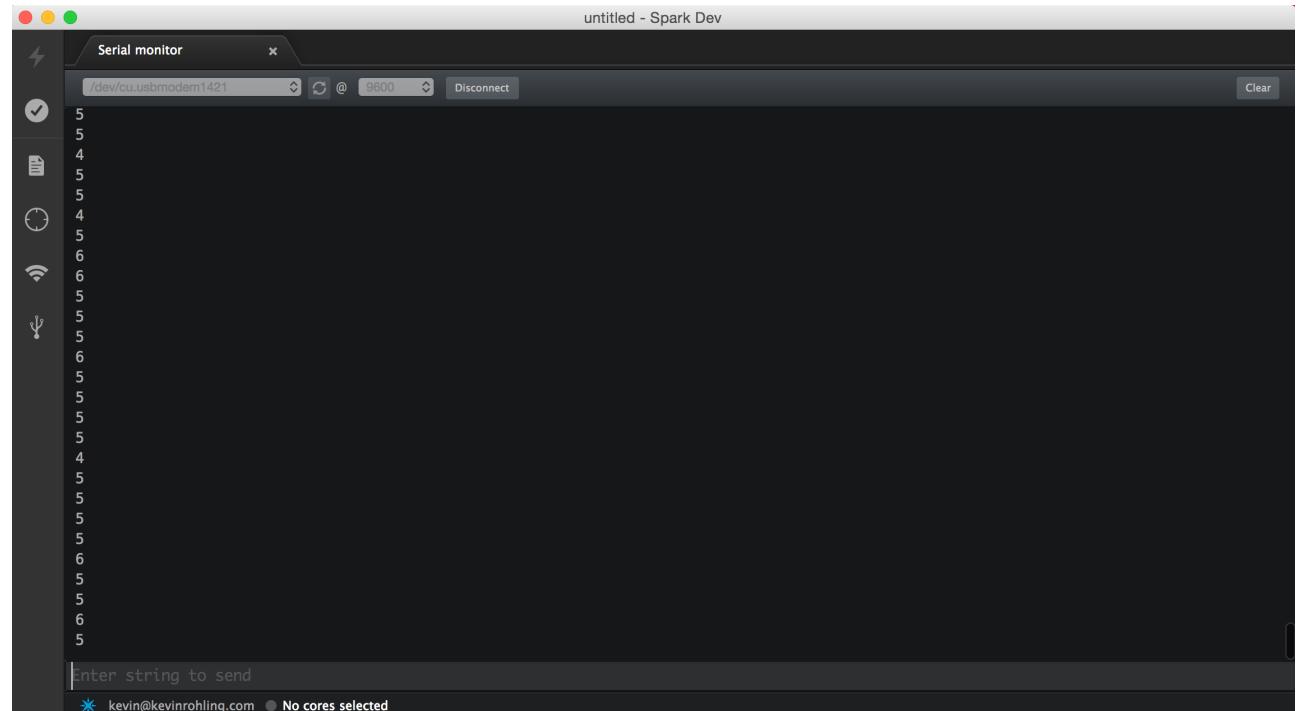
When you touch the sensor, it causes the aluminum foil to act like a capacitor and absorb charge.

This makes it take longer for the charge to travel to the RECEIVE PIN.

```
digitalWrite(SEND_PIN, HIGH);
while(recvValue == LOW && counter < CAP_TIMEOUT) {
    counter++;
    recvValue = digitalRead(RECV_PIN);
}
```

Step 2: Open the Spark Dev Serial Monitor

Notice that the numbers in the console get bigger when you are touching the sensor.





Step 3: Set the touch threshold value.

Based on the numbers you see in the Serial Monitor find a new value for the touch threshold in the loop() method.

A correct value will cause the LED to toggle on and off when you touch the sensor.

Remember to use the button to upload your code to the Spark Core.

```
void loop()
{
    //This is the number of reads necessary to trigger a 'touch'.
    //*****YOU SHOULD REPLACE THIS VALUE*****//
    int touchThreshold = -1;
    //*****YOU SHOULD REPLACE THIS VALUE*****//

    //Check the touch sensor.
    int newTouchState = checkTouchSensor(touchThreshold);

    //Check to see if our touch state just changed from OFF to ON.
    //If so, we'll toggle the LED.
    if(touchState != lastTouchState && touchState == ON_STATE) {
        toggleLed();
    }

    lastTouchState = touchState;
}
```



Step 4: Activate the Emberlight connection

After the `toggleLed()` line add `toggleEmberlight()` to the next line.

Remember to use the  button to upload your code to the Spark Core.

```
//Check to see if our touch state just changed from OFF to ON.  
//If so, we'll toggle the LED.  
if(touchState != lastTouchState && touchState == ON_STATE) {  
    toggleLed();  
    toggleEmberlight();  
}
```



We hope you've enjoyed interacting with the
Internet of Things today.

@emberlightco

<http://emberlight.co>

Download the code: <http://github.com/emberlight>

