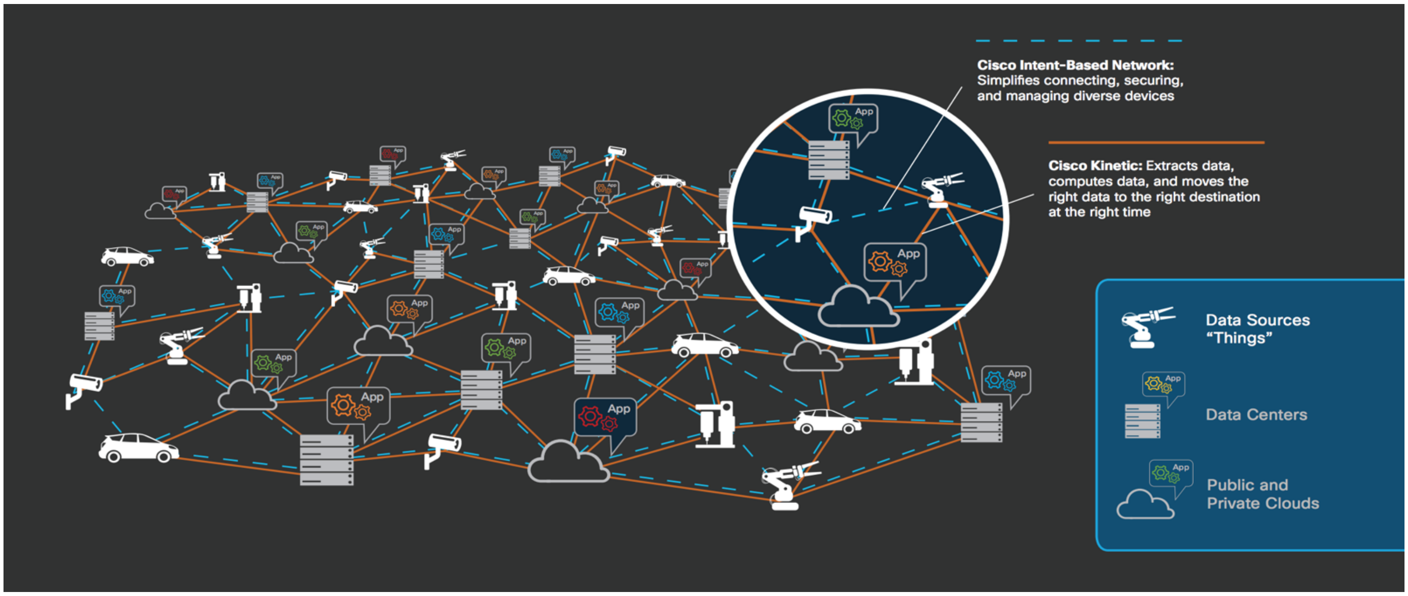
Cisco IoT Hackathon

Welcome to the Cisco Hackathon.


Cisco Intent-Based Network: Simplifies connecting, securing, and managing diverse devices
Cisco Kinetic: Extracts data, computes data, and moves the right data to the right destination at the right time
App
Data Sources "Things"
Data Centers
Public and Private Clouds

Cisco - Internet of Things (IoT) is changing the way we work, live, play, and learn.

For our Hackathon today we will be solving some basic IoT problems, developing algorithms and build prototypes of IoT solution. The content is divided into 4 sections. Each Section has a simple task that will introduce some basic IoT or Programming Concept. Once we finish the basic task you can also explore the Mission tasks if you like.

Always feel free to improve the solution with your advanced programming skills.

We hope this experience will be a lot of fun, set your imagination free & enjoy!!!

Getting Started

- Step1) Connect Chromebook to WiFi.
- Step2) Please note down your Pod number.
- Step3) Visit https://build.particle.io/build/new
- Step4) Login with following credentials :
    - Username: ciscoiot2018+**[N]**@gmail.com
        - (replace [N] with your Pod number. For example if you are PoD1 : N=1)
        - Username ciscoiot2018+1@gmail.com
        - Password: Cisco123
- Step5) (Optional) You can also download Particle App in you smartphone/ tablet and login with same credentials.
- Step6) Now you are ready to Program to your IoT Kit

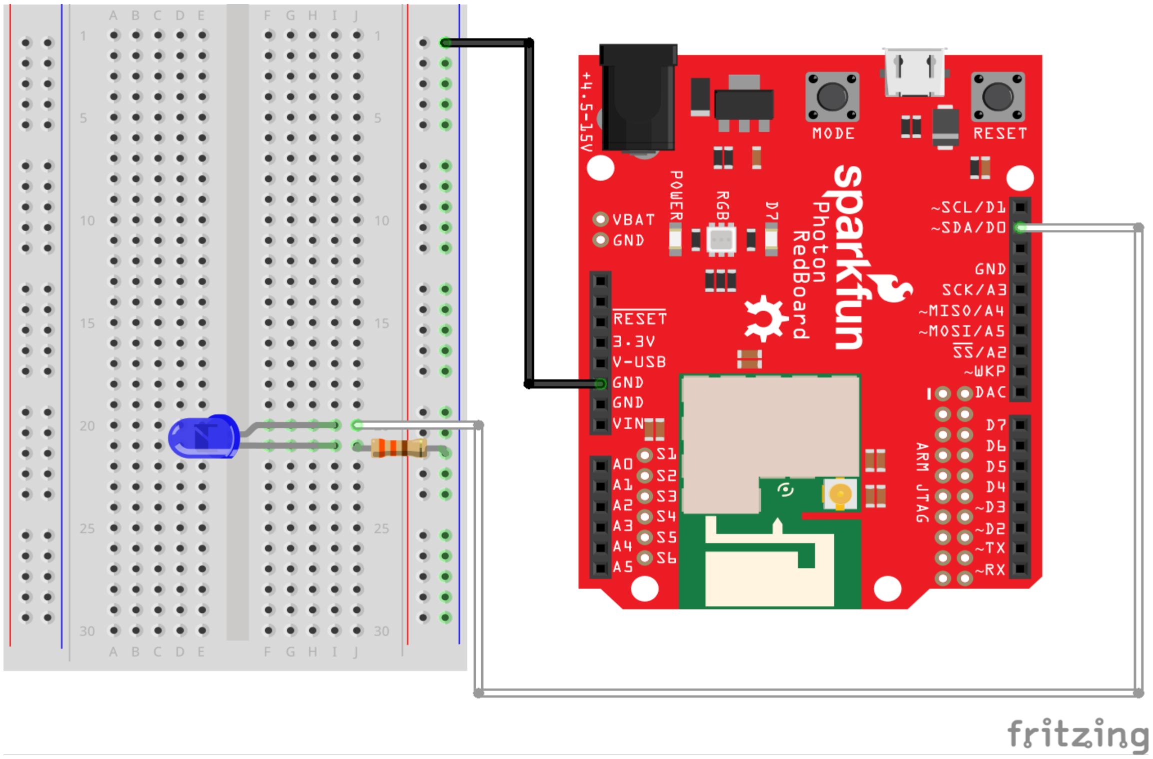
Note:
These IoT kits will be again used by students of all ages.
Please handle them with care and return them after your session. Thank you.

Section 1: Hello World, Blink an LED

In this Task we will blink an LED to make sure our is operational. Begin by connecting the circuit with LED and resistor as shown in the diagram below.

Detailed instructions can also be found here https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-for-photon-experiment-guide/experiment-1-hello-world-blink-an-led

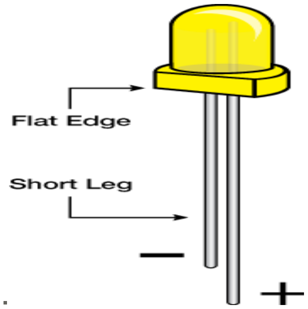
Hardware Hookup



Note:

All jumper wires work the same. They are used to connect two points together. All the experiments will show the wires with different colored insulations for clarity, but using different combinations of colors is completely acceptable.

Be sure to the polarity on the LED is correct. The longer lead should be connected to D0. You will need to slightly bend the longer leg so that both are the same length when placed in the breadboard. (refer the LED Diagram below).



You can “Power On” the IoT Kit by connecting MicroUSB cable.

Hello World Program- Blinking LED

```
int led = D0; // LED is connected to D0

// This routine runs only once upon reset
void setup()
{
  pinMode(led, OUTPUT); // Initialize D0 pin as output
}

// This routine loops forever
void loop()
{
  digitalWrite(led, HIGH);  // Turn ON the LED
  delay(1000);              // Wait for 1000mS = 1 second
  digitalWrite(led, LOW);   // Turn OFF the LED
  delay(1000);              // Wait for 1 second
}
```

After coding this program in your web page. Save, Verify and Flash in your device. Check if LED is blinks.

Mission Tasks

1) Can you modify your code increase the LED Blinking interval?
2) Blink the Onboard LED

Section 2: Building your Algorithm

This Task will focus on your programming skills. Use the Random number generator function generate a number between 40 and 80. Blink an LED every time the number if the number is above 60.

```
int led = D0; // LED is connected to D0

// This routine runs only once upon reset
void setup()
{
  pinMode(led, OUTPUT); // Initialize D0 pin as output
}

// This routine loops forever
void loop()
{
  int number= random(40,80); //Generate random number between 40 and 80
  if(number>60)
  {
        digitalWrite(led, HIGH);   // Turn ON the LED
        delay(1000);               // Wait for 1000mS = 1 second
        digitalWrite(led, LOW);    // Turn OFF the LED
        delay(1000);               // Wait for 1 second
  }
}
```

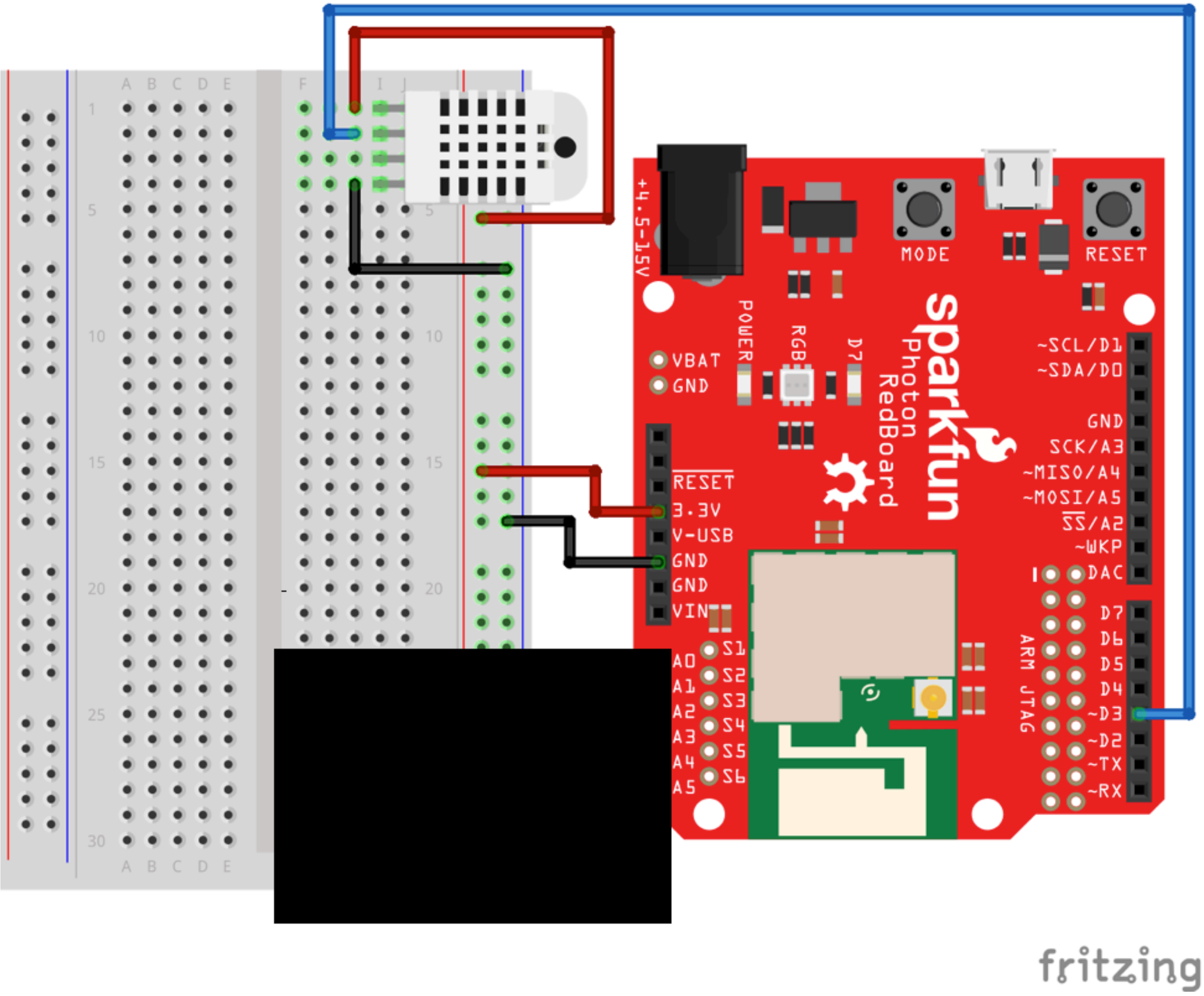
Mission Tasks

3) Change the code to generate random number between -40 to +80
4) Modify the code to blink LED for even number.

Section 3: Publish Temperature to Console

This Task will focus on measuring the Temperature using temperature sensor. After measuring this temperature the IoT will automatically publish this reading to the Particle.io Console.

Start this task by building your circuit following this Hardware hookup below



[redacted]

Start building your circuit carefully. In this circuit we will use the LED instead of photocell.

https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-for-photon-experiment-guide/experiment-6-environment-monitor

In this code below replace [N] with your Pod number to make sure your message is unique.

```
#include "SparkFunRHT03/SparkFunRHT03.h"

const int RHT03_DATA_PIN = D3; // RHT03 data pin
RHT03 rht; // This creates a RTH03 object, which we'll use to interact with the sensor

void setup()
{

    rht.begin(RHT03_DATA_PIN); // Initialize the RHT03 sensor

}

void loop()
{
    int update = rht.update(); // Read from the RHT
    if (update == 1) // If the update was successful:
    {
     float humidity = rht.humidity();// Read Relative humidity into a variable
     float tempC = rht.tempC(); // Read Celsius temperature into a variable
     float tempF = rht.tempF(); // Read farenheit temperature into a variable

     String temp_F = " The current Temperature in Farenheit in PoD [N] is : " + String(tempF);

     Particle.publish("Sparktest", temp_F, PRIVATE); // Publish message to Console

     delay(30000);
}
}
```

But wait! Don't try to upload it yet. In fact, if you try to compile, you should get an error, because we need to **add the SparkFunRHT03 library**.

Click the Libraries icon 🔖 on the left. Then click into the search bar and find **SparkFunRHT03.** Click it to get a description of the library, and options for using it.

Mission Tasks

5) In this Task we published Temperature in Farenheit. Also publish Temperature in Celcius and Relative Humidity measured.
6) Use your previous algorithm in Task (2) and send message in Spark only if temperature is above 70F.

Expert Level

Section 4: Advanced Cisco Bot & IoT Integration

This is the last section and its intended to enable advanced communications by creating web hooks between your IoT kit and Cisco Spark. Building your own Bot & more..

In the Previous Task we published the temperature to Particle.io console. Try building your own web-hooks and publish into Cisco spark room.

Hint:

a) Signup Cisco Spark & read about Bots, web-hooks integrations.

https://learninglabs.cisco.com/tracks/collab-cloud/business-messaging/collab-spark-apps/step/1

b) Create a Spark room where you want to receive the messages.
c) Use the "Incoming web-hooks App" to create web-hook-URL for this Cisco Spark

https://depot.ciscospark.com/integrations/incoming-webhooks-cisco-systems/categories/other

d) Modify the current integration in Particle to add your web-hook https://console.particle.io/integrations

Mission Task

7) Try Cisco Spark API 101 module https://learninglabs.cisco.com/tracks/collab-cloud

Hope you all enjoyed working through these tasks.

Additional resources

Cisco Spark Bot & Integrations here: https://developer.cisco.com/site/spark/

Cisco IoT www.cisco.com/go/iot