

---

# Make a Cloud Data Logger With the Particle Photon

When building Internet of Things projects, for example with Arduino, one problem is to find the right cloud platform for your project. It can be difficult to configure, and then it can be even more complicated to write the code so your project connects to this particular platform.

Particle (ex-Spark) solved this problem for us with the Spark Core, which is a little board that automatically connects to the Particle platform, and is instantly accessible from the web. Now, they are back with a brand new board: the Photon. Compared to the Core, it's more powerful, has the WiFi along with the processor in a single module, and it's also cheaper (only \$19). The cool thing is that you can also buy the Photon chip on its own, to embed in your own products later on.

In this article, we are going to see how to use a Particle Photon to build a simple cloud data logger project. We'll connect some sensors to the Photon, connect it to the web, and then use the dashboard feature from Particle to visualise in live the measurements made by the board. Let's dive in!

## Hardware & Software Requirements

Let's first see what we need for this project. Of course, you'll need a Particle Photon board:



For the sensors, I'll be using a simple DHT11 sensor for temperature & humidity measurements, and also a photocell for light level measurements.

We'll also need a breadboard and some jumper wires.

This is the list of all the required components for this project:

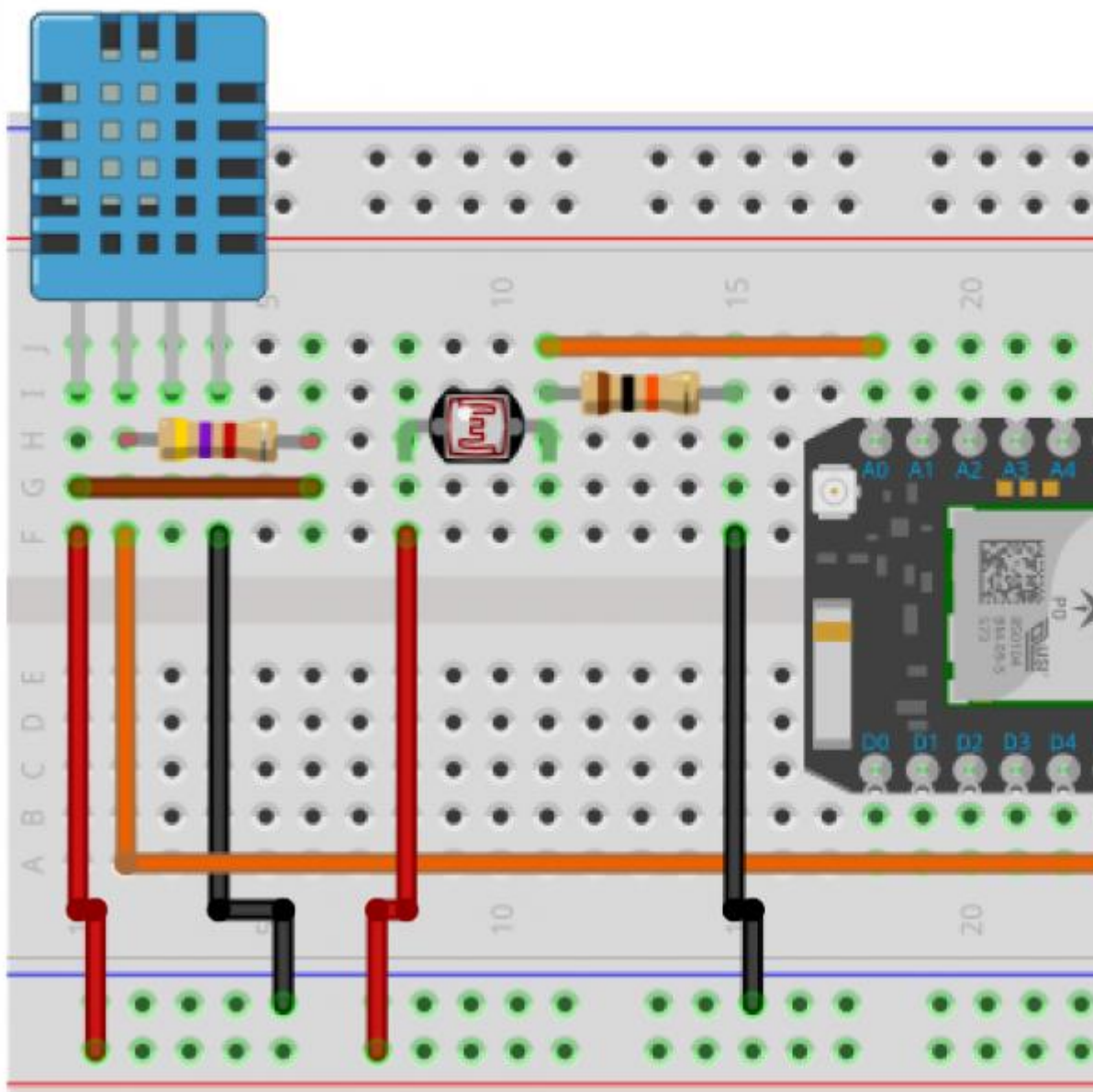
- [Particle Photon](#)
- [DHT11 sensor](#)
- [4.7K Ohm resistor](#)
- [Photocell](#)

- 10K Ohm resistor
- Breadboard
- Jumper wires

On the software side, we are going to use the Particle Web IDE, so you don't need to install anything on your own computer. If it's not done yet, just [create an account on the Particle website](#).

## Hardware Configuration

We are now going to assemble the different elements of this project. This is a schematic to help you out:

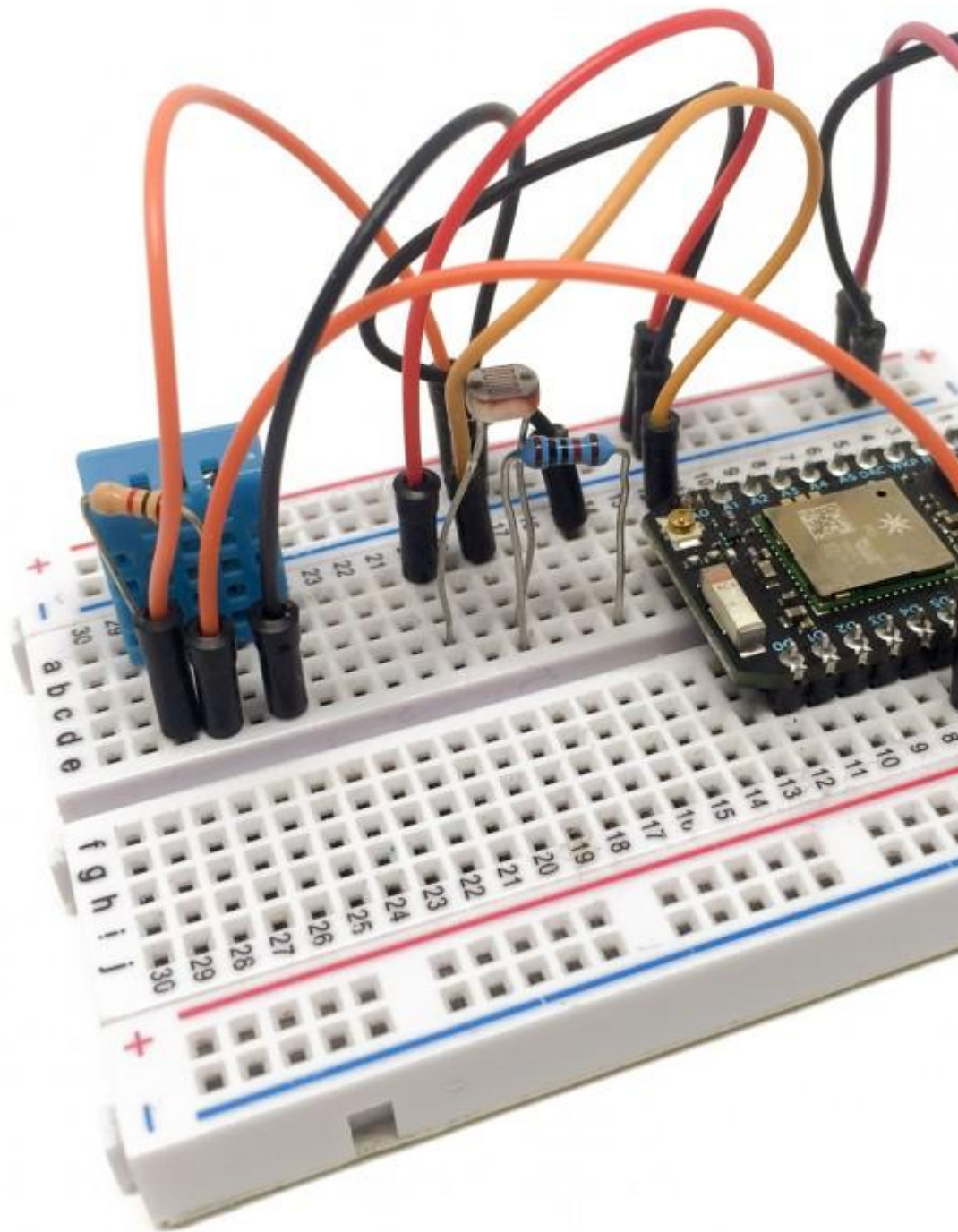


The first step is to place the Photon on the breadboard, and then connect the power from the Photon to the two power lines on the breadboard.

Then, place the DHT11 sensor on the breadboard. Connect the left pin to VCC, and the right pin to GND, like on the schematic. Then, connect pin number 2 of the sensor the pin D5 of the Photon. Finally, place the 4.7k Ohm resistor between pin 1 & 2 of the sensors.

Now, place the photocell on the breadboard, in series with a 10K Ohm resistor. Then, connect the other side of the photocell to VCC, the other side of the resistor to GND, and finally the common pin to pin A0 of the Photon.

This is a picture of the completely assembled project:



## Configuring Your Photon

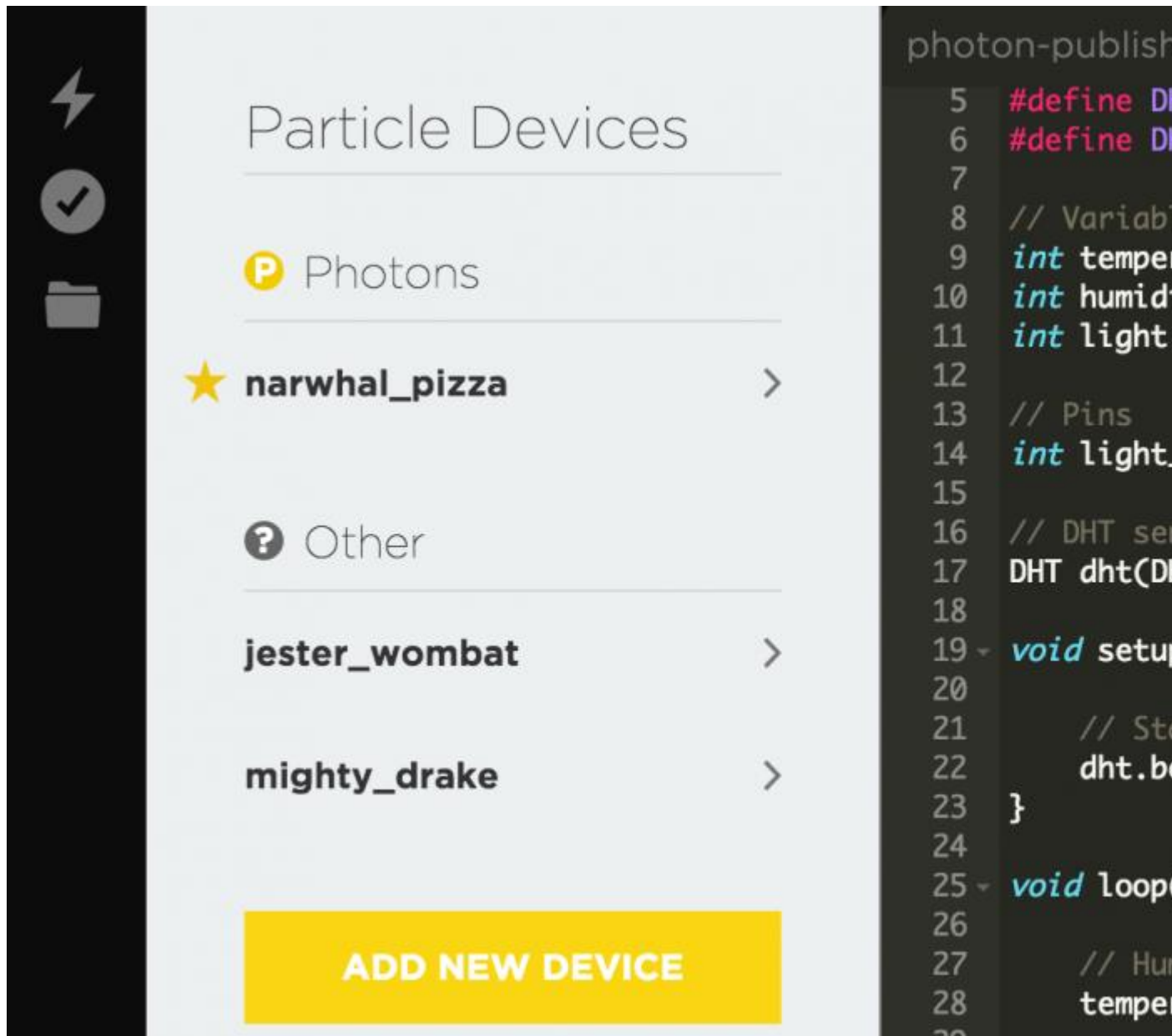
We are now going to configure your Photon for a first use. If that's already done, you can simply skip this section.



The easiest way to configure a Photon is to use the Particle mobile app, that you can find on either the App Store or the Google Play store, depending on your phone. From the app, you'll be guided so you can enter the password for your WiFi network. Once that's done, simply go to:

<https://build.particle.io/>

Click on 'devices' in the left menu, and you should see that your newly configured device is online:



Also make sure that the little start next to the Photon is selected, so it will be flashed automatically when we decide to configure the Photon with the code we'll write in a minute.

Now, an important thing at this point is to make sure your Photon is updated to the latest version. The one I received had an old firmware on it, and I just couldn't program it from the Web IDE. To do that, I recommend using the Particle-CLI tool. You can learn how to install it from:

<https://github.com/spark/particle-cli>

You also need to put your Photon in DFU mode. To do so, simply hold both buttons on the board, and then release the Reset button first. The Photon LED should be blinking yellow.

Finally, download the firmware & get the required instructions from:

<https://github.com/spark/firmware/releases>

Once your device is updated to the latest firmware version, you can move to the next step.

### Writing the Code for the Project

We are now going to write the code for the project. The code will be in charge of measuring data from the sensors, and then publishing these data on the Particle platform. This is the complete code for the project:

```
// This #include statement was automatically added by the Particle IDE.
#include "Adafruit_DHT/Adafruit_DHT.h"

// DHT parameters
#define DHTPIN 5
#define DHTTYPE DHT11

// Variables
int temperature;
int humidity;
int light;

// Pins
int light_sensor_pin = A0;

// DHT sensor
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    // Start DHT sensor
    dht.begin();
}

void loop() {
```

```

// Humidity measurement
temperature = dht.getTempCelcius();

// Humidity measurement
humidity = dht.getHumidity();

// Light level measurement
float light_measurement = analogRead(light_sensor_pin);
light = (int)(light_measurement/4096*100);

// Publish data
Spark.publish("temperature", String(temperature) + " °C");
delay(2000);
Spark.publish("humidity", String(humidity) + "%");
delay(2000);
Spark.publish("light", String(light) + "%");
delay(2000);

}

```

Let's see the details of this code. First, we define on which pin the DHT11 sensor is connected to:

```

#define DHTPIN 5
#define DHTTYPE DHT11

```

We also define the pin of the light level sensor:

```
int light_sensor_pin = A0;
```

Then, we declare a set of variables for the measurements:

```
int temperature;
int humidity;
int light;

```

We also create an instance of the DHT sensor:

```
DHT dht(DHTPIN, DHTTYPE);
```

In the setup() function, we initialise the DHT sensor:

```
dht.begin();
```

Now, the real action happens in the loop() function of the sketch. First, we make all the measurements from the sensors:

```

// Humidity measurement
temperature = dht.getTempCelcius();

// Humidity measurement
humidity = dht.getHumidity();

// Light level measurement
float light_measurement = analogRead(light_sensor_pin);

```

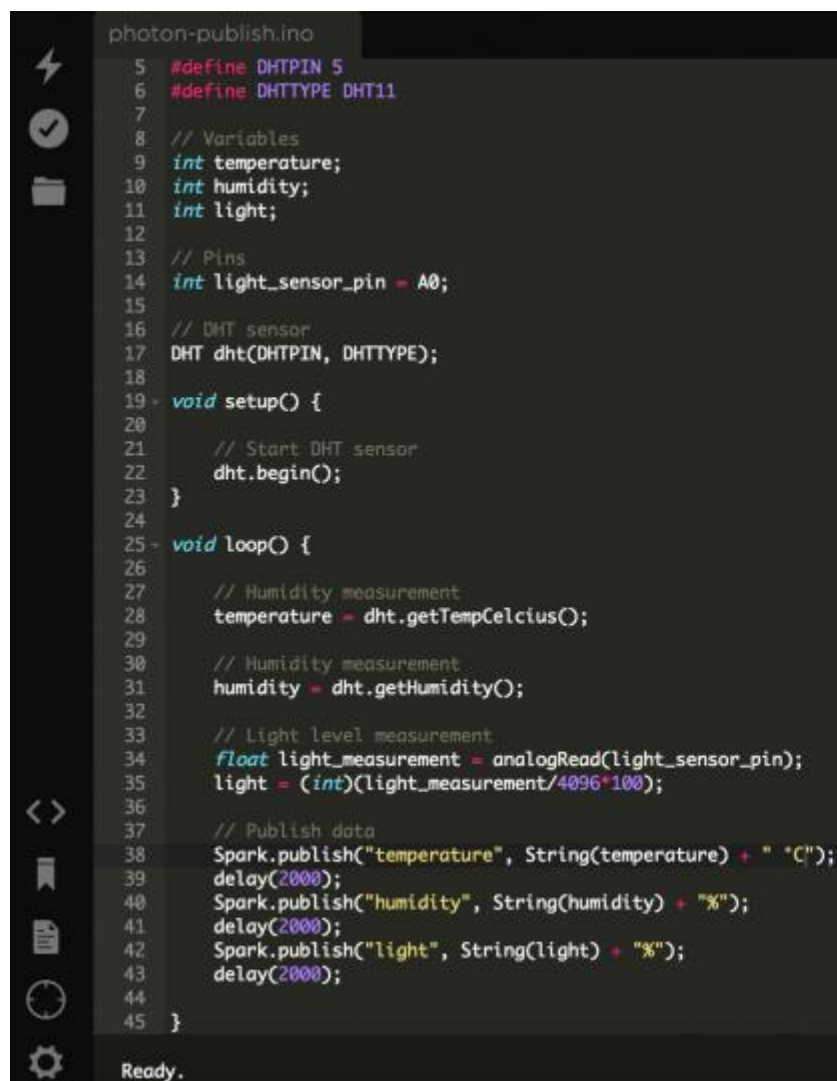


```
light = (int)(light_measurement/4096*100);
```

Then, we use the `Spark.publish()` function to send the data every 2 seconds on the Particle cloud platform:

```
Spark.publish("temperature", String(temperature) + " °C");
delay(2000);
Spark.publish("humidity", String(humidity) + "%");
delay(2000);
Spark.publish("light", String(light) + "%");
delay(2000);
```

You can now copy the whole code in a new project inside the Particle Web IDE. There is one more thing we have to do here: adding the DHT library in the project. You can do that from the ‘libraries’ section inside the left menu. From there, select “ADAFRUIT\_DHT”, and add it to your current project. This is how it should look like:



```
photon-publish.ino
5  #define DHTPIN 5
6  #define DHTTYPE DHT11
7
8  // Variables
9  int temperature;
10 int humidity;
11 int light;
12
13 // Pins
14 int light_sensor_pin = A0;
15
16 // DHT sensor
17 DHT dht(DHTPIN, DHTTYPE);
18
19 void setup() {
20
21     // Start DHT sensor
22     dht.begin();
23 }
24
25 void loop() {
26
27     // Humidity measurement
28     temperature = dht.getTempCelcius();
29
30     // Humidity measurement
31     humidity = dht.getHumidity();
32
33     // Light level measurement
34     float light_measurement = analogRead(light_sensor_pin);
35     light = (int)(light_measurement/4096*100);
36
37     // Publish data
38     Spark.publish("temperature", String(temperature) + " °C");
39     delay(2000);
40     Spark.publish("humidity", String(humidity) + "%");
41     delay(2000);
42     Spark.publish("light", String(light) + "%");
43     delay(2000);
44 }
45
Ready.
```

You can now use the 'Flash' button from the left menu to send the code to your Photon. If it works correctly, you should see the LED on the board flashing magenta during the operation, and then the board should reboot.

## Visualising the Live Data Stream

Remember that we used the `publish()` function inside our code? Well, this means that the measurements will be sent directly to the Particle cloud platform. You can now go to the Dashboard section of the Particle website:

<https://dashboard.particle.io/>

You should immediately see the live data coming from your board:

Logs  

2  
0

EVENT NAME	DATA	PUBLISHED AT
light	65%	August 29th
humidity	32%	August 29th
temperature	27 °C	August 29th
light	65%	August 29th
humidity	32%	August 29th
temperature	27 °C	August 29th
light	65%	August 29th

You can also click on a given piece of data, and see the details of the JSON data that was sent to the platform:

temperature 27 °C August 29th

```
{"data": "27 °C", "ttl": "60", "published_at": "2015-08-29T15:25:16.748Z", "coreid": "22001f000247343138333038", "name": "temperature"}
```

From there, you can use this page to see the measurements done by your photon in real time. You can also for example create your own app and use this data to make a dashboard, or record them into your own database. In the future, Particle is going to integrate graphical dashboard features, so you can create your own Internet of Things dashboard directly. As soon as this feature becomes available, this article will be updated.

## How to Go Further

In this article, we built a data logger based on the Particle photon, that automatically send measurements to the Particle cloud platform. There are of course many ways you can use what you can learn in this article & build more complex applications. You can for example use many Photon boards, with the same code, that will all send data to the Particle cloud platform. You can then monitor them all from a single page. You could also have other sensors that will trigger a `publish()` function on a given condition, for example when motion is detected by a motion sensor.