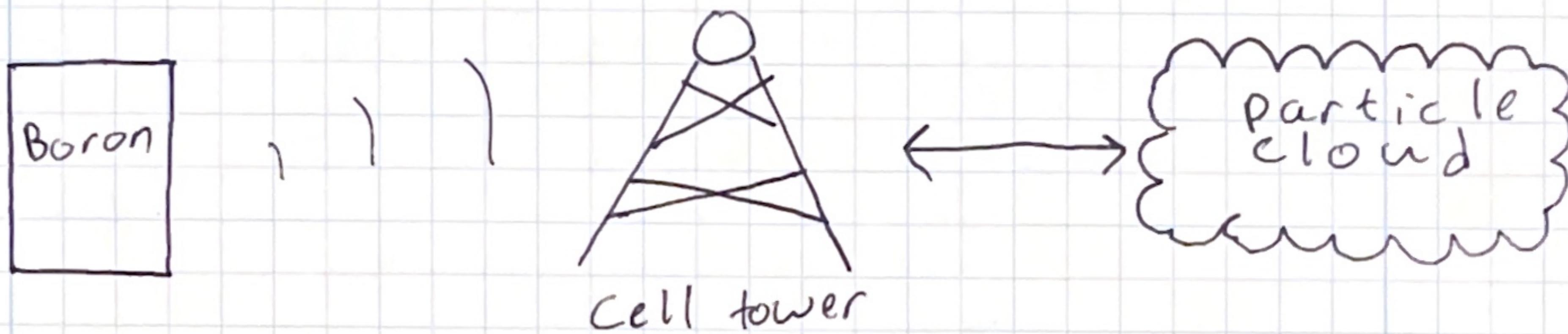


Set up

Particle as a company sells a web management interface for microcontrollers that have a cellular chip. This cellular chip allows the microcontroller to communicate over our cell towers.

We are using the particle board called "boron"



The particle cloud allows us to manage all of our devices. It even allows us to upload code entirely "OTA" (over the air).

Particle also has several other capabilities. Today, we learned about "Particle Functions" and "Particle Variables". These allow us to interact with our program over the web console.

To set up a particle board, go to Setup. Particle. ~~com~~.io

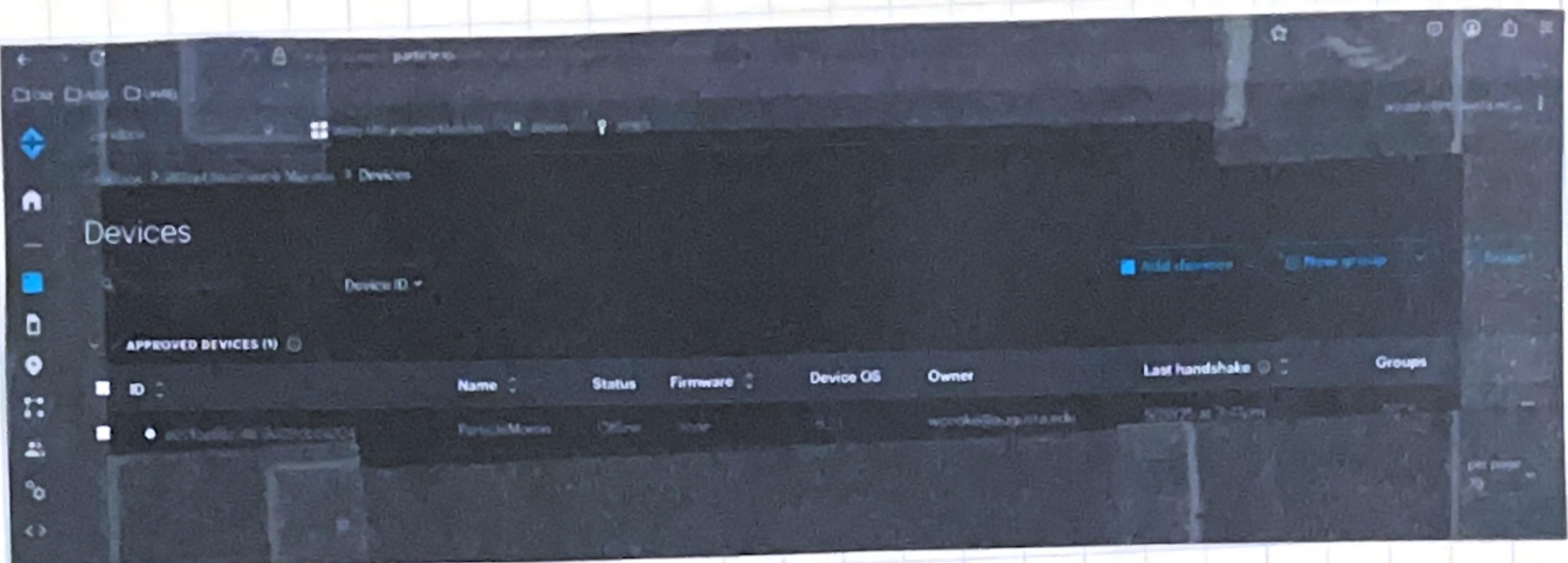
Plug the board in and follow the instructions.

Next go to console. Particle. io.
Here is like a main dashboard for everything.
Go to "my product" and see if your device is there.

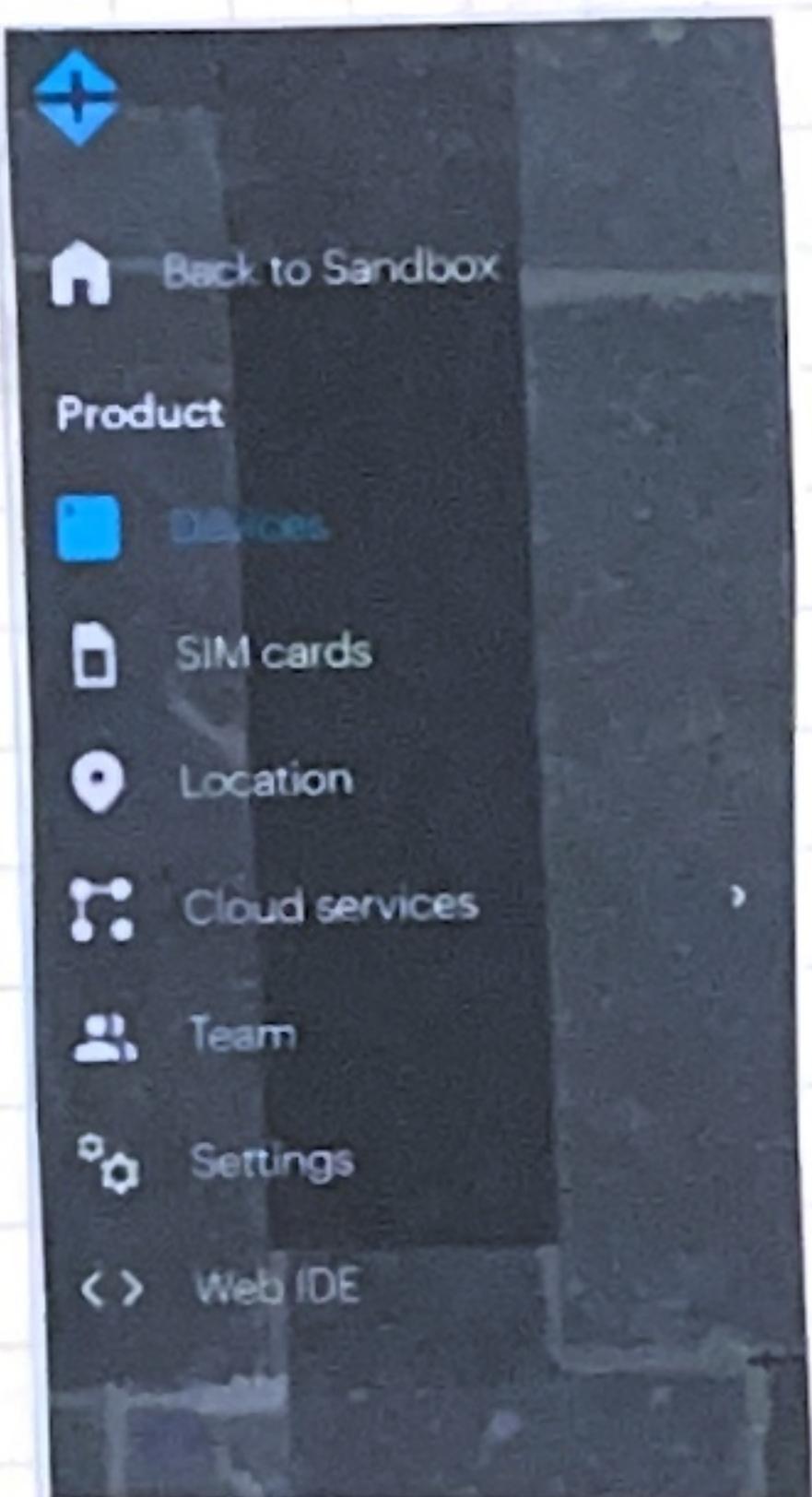
Introduction to particle Functions and Variables

08/28/25
Wesley Cook

Console



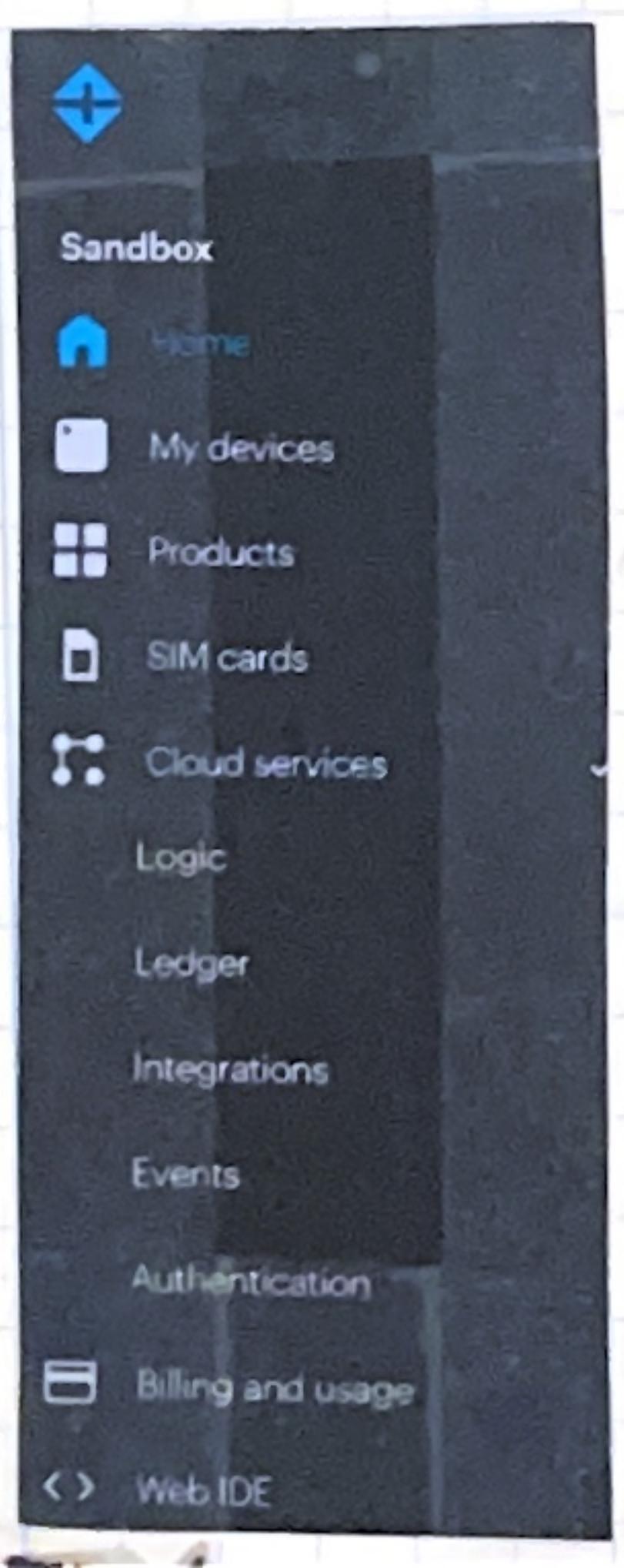
After clicking on my product, I see the device I setup earlier.



If we hover over the side menu, we get some options.

- Notice that we are in a product.
- ↗ Devices associated with this product

→ Settings of this product.



What if we click "back to sandbox"

- all devices on any product.
- Products

"Sand box" → Collection of "products"

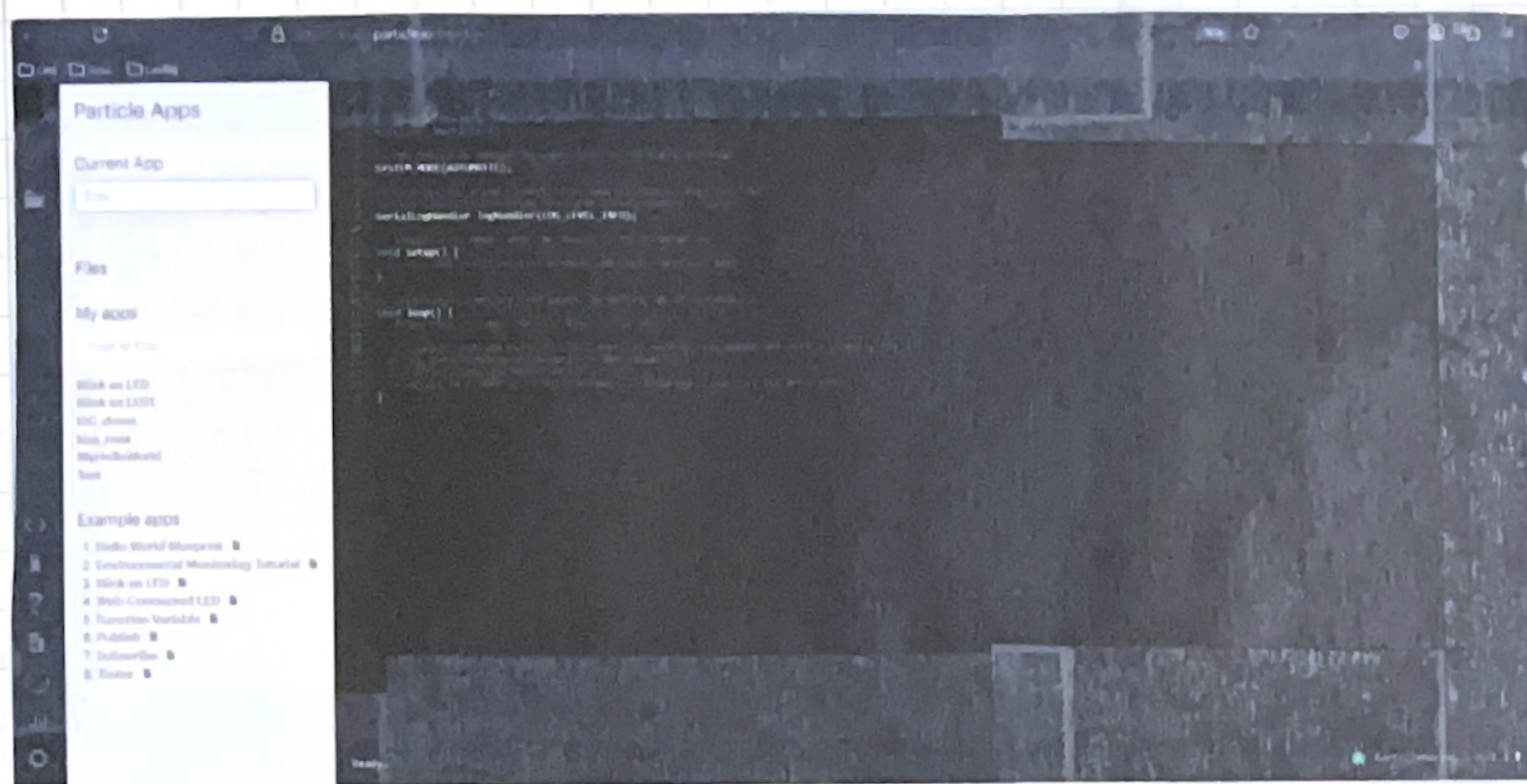
"Product" → Collection of devices

05/28/25
Wesley
code

web IDE

Introduction to Particle Functions and Variables

79



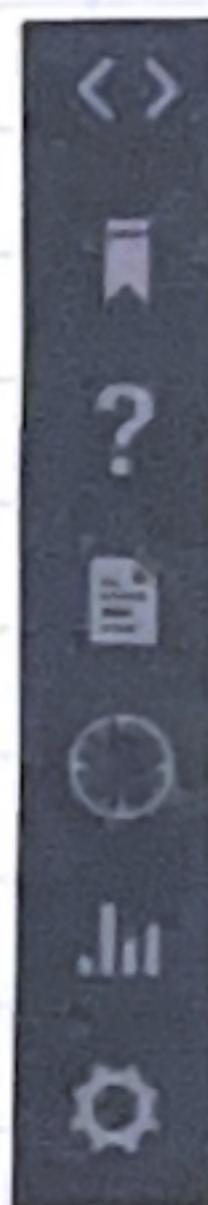
If we click the web IDE, we get this page. Here, we can write code and upload that code to the particle devices.



→ Flash device

→ Verify code

→ Save code



→ Code, list files

→ Libraries

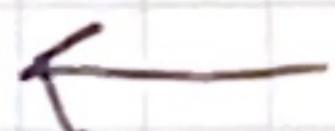
→ help?

→ DOCS

→ Devices, Select device to flash

→ console

→ Account settings



Particle functions

Particle functions allow us to interact with our code from the web IDE. They must return an integer and take in a string argument.

Syntax:

Particle.function ("Name on Console", Function Handle);

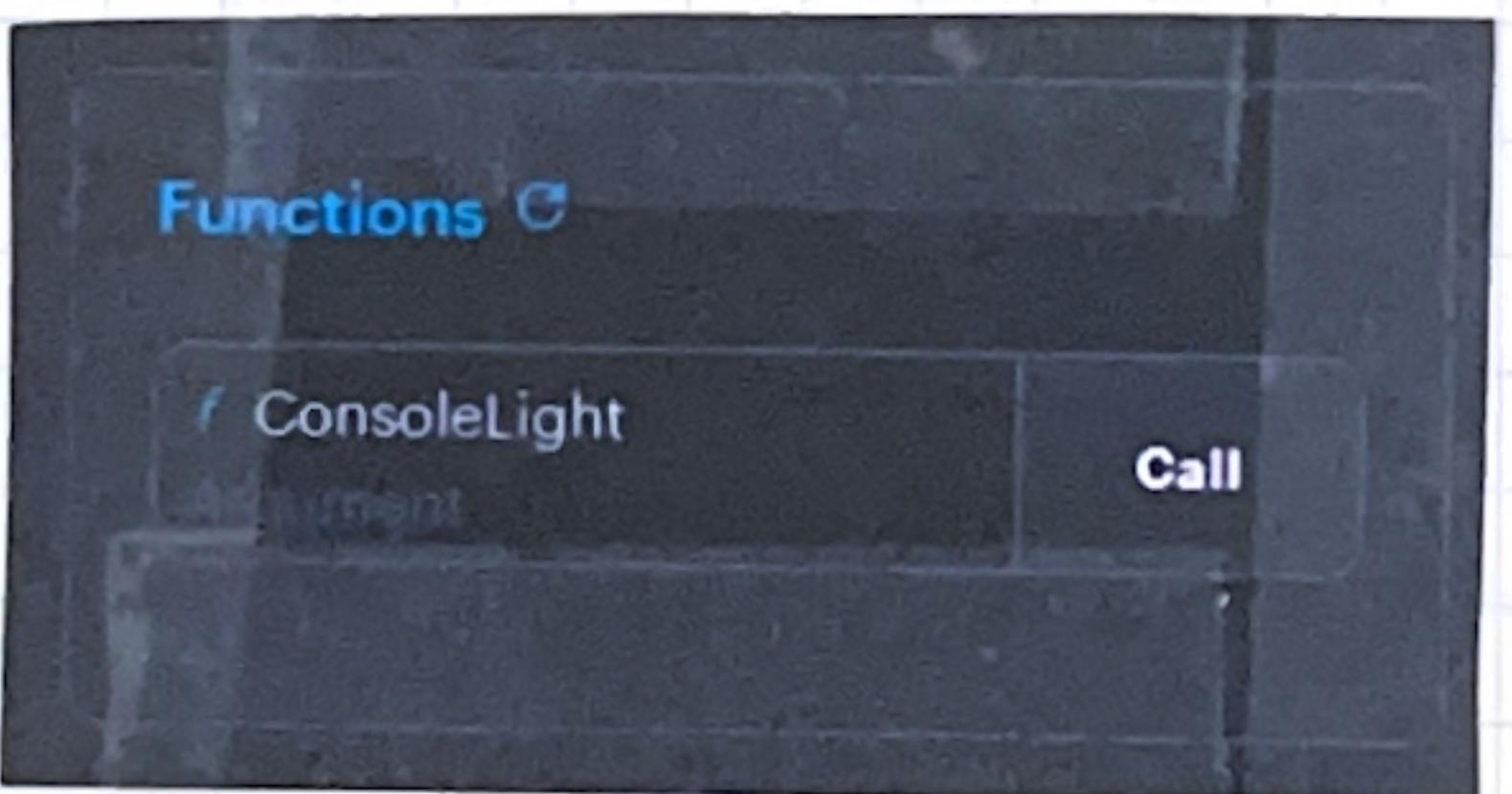
Can be anything like "Start a row"
or "console bright"

This is the name you will look for on ~~the~~ [Console.particle.io](https://console.particle.io).

The function handle is the name of the function that it should be associated with in code. Similar to how we tell an interrupt what function to call.

~~on the top~~ On console.particle.io, search for your product and click on the device that has a particle function.

It will show up on the right menu;



```
1 SYSTEM_MODE(AUTOMATIC);
2 SerialLogHandler logHandler(LOG_LEVEL_INFO);
3
4 int pcbs;
5
6 const int LEDP = D0;
7
8 void setup() {
9     pinMode(LEDP, OUTPUT);
10    pinMode(D7, OUTPUT);
11    pinMode(D2, INPUT);
12
13    Particle.function("ConsoleLight", blink);
14 }
15
16 void loop() {}
17 |
18 int blink(String cmd){
19     if (cmd == "ON")
20     {
21         digitalWrite(D0, HIGH); // Turn on
22         return 1;
23     }
24     else if (cmd == "OFF")
25     {
26         digitalWrite(D0, LOW); // Turn off
27         return 0;
28     }
29     else
30     {
31         return -1;
32     }
33 }
```

So if we type "ON" into the console, it will turn on D0. Notice that we didn't have to put this in the loop!

Our particle function can be called from anywhere we have access to `console.particle.io`, assuming the board is powered and it can reach a cell tower.

particle Variables

Particle Variables can be found in the same place as particle functions: `console.particle.io` and select the specific device.

Particle Variables allow us to see the value of some variable in our program from the console. They can not be a float datatype.

Syntax:

`particle.variable("Name on Console", Variable in code);`

```

1 // Include Particle Device OS APIs
2 #include "Particle.h"
3 #include "Wire.h"
4
5 // Let Device OS manage the connection to the Particle Cloud
6 SYSTEM_MODE(AUTOMATIC);
7 SerialLogHandler logHandler(LOG_LEVEL_INFO);
8
9 const int led    = D9;
10 const int button = D2;
11 int myAdd;
12 int tempC;
13
14 // setup() runs once, when the device is first turned on
15 void setup() {
16     Wire.begin();
17     pinMode(led, OUTPUT);
18     pinMode(button, INPUT);
19     while(!Particle.connected());
20
21     Serial.begin(9600);
22     // delay(5000);
23     for (int addr = 0; addr <= 127; addr++)
24     {
25         Wire.beginTransmission(addr);
26         if (Wire.endTransmission() == 0) // Sensor sent back ACK
27         {
28             myAdd = addr;
29             Serial.print("0x"); Serial.println(addr, HEX);
30         }
31     }
32
33     Particle.variable("Temp", tempC);
34 }
35
36 // loop() runs over and over again, as quickly as it can execute.
37 void loop() {
38     Wire.beginTransmission(myAdd);
39     Wire.write(0x00);
40     Wire.endTransmission();
41     delay(100);
42
43     Wire.requestFrom(myAdd, 1);
44     tempC = Wire.read();
45
46     Serial.print("Temp: "); Serial.println(tempC);
47 }
```

↳ Using TC74

