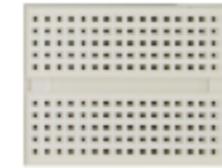


IoT Workshop
Prototype Like a Pro!



Checklist

1x Small breadboard



1x Wemos



1x Temp. & humidity sensor



1x Resistor

1x LED

1x Motion sensor



1x MicroUSB cable



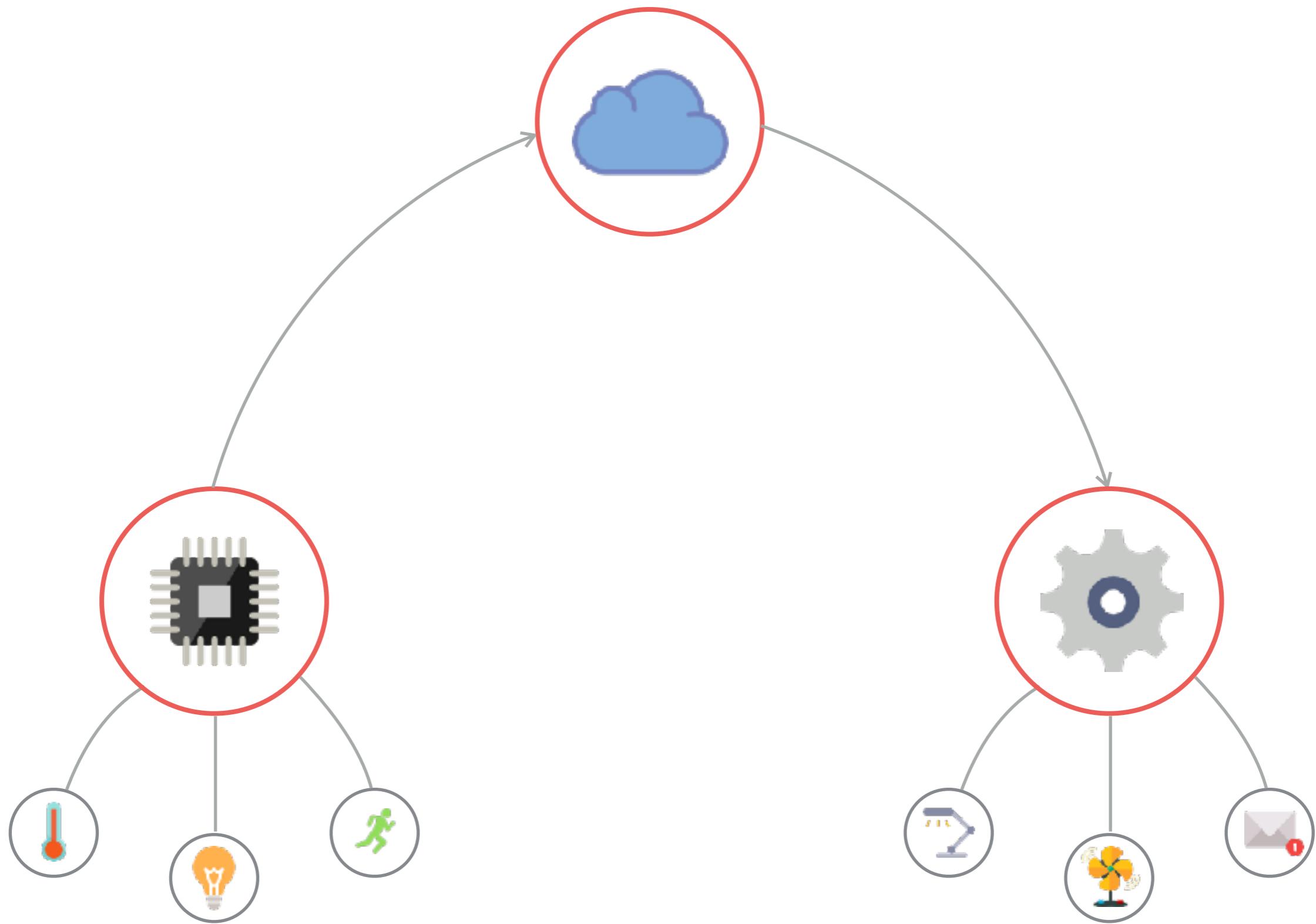
1x bunch of wires



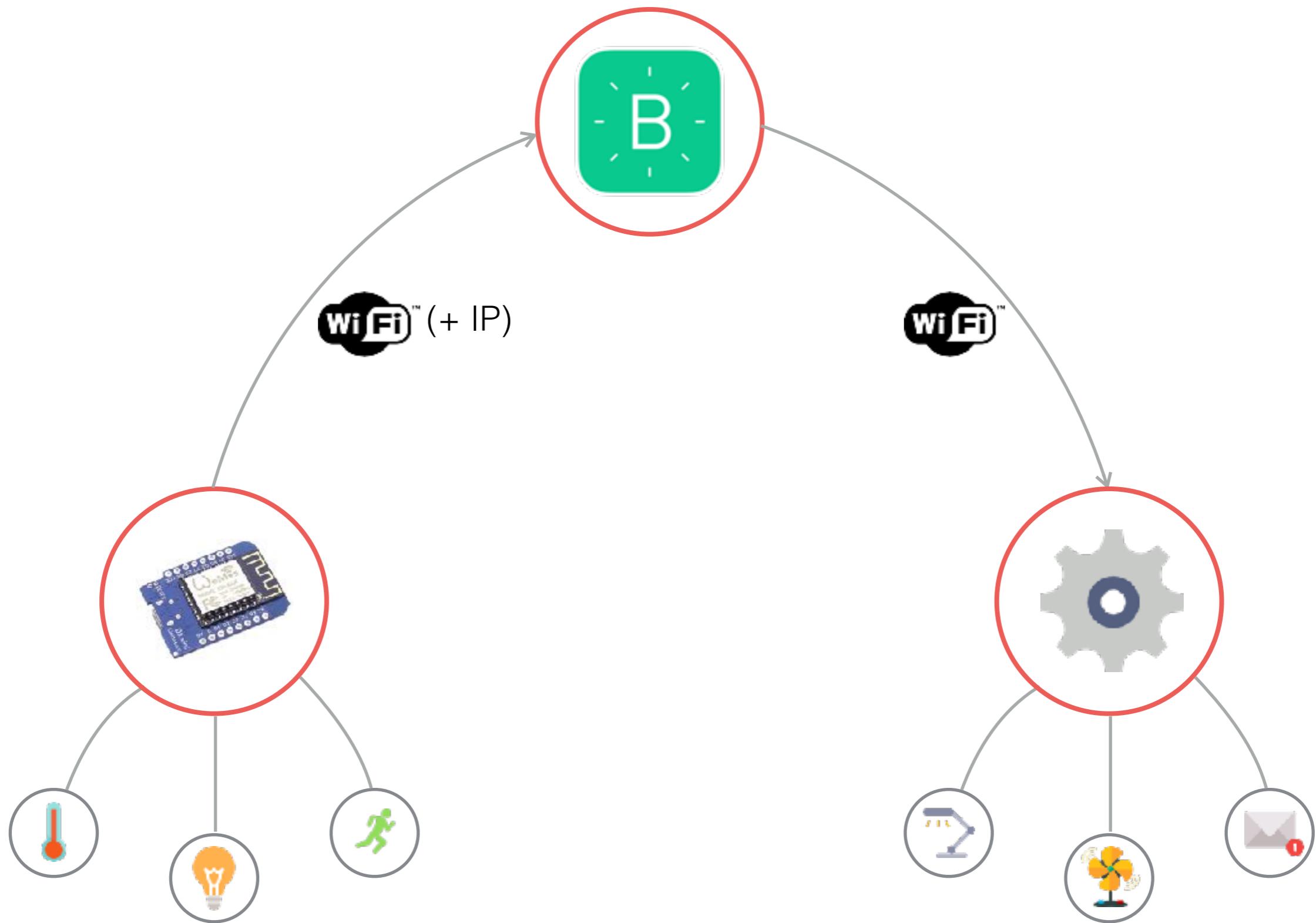
A bit of theory



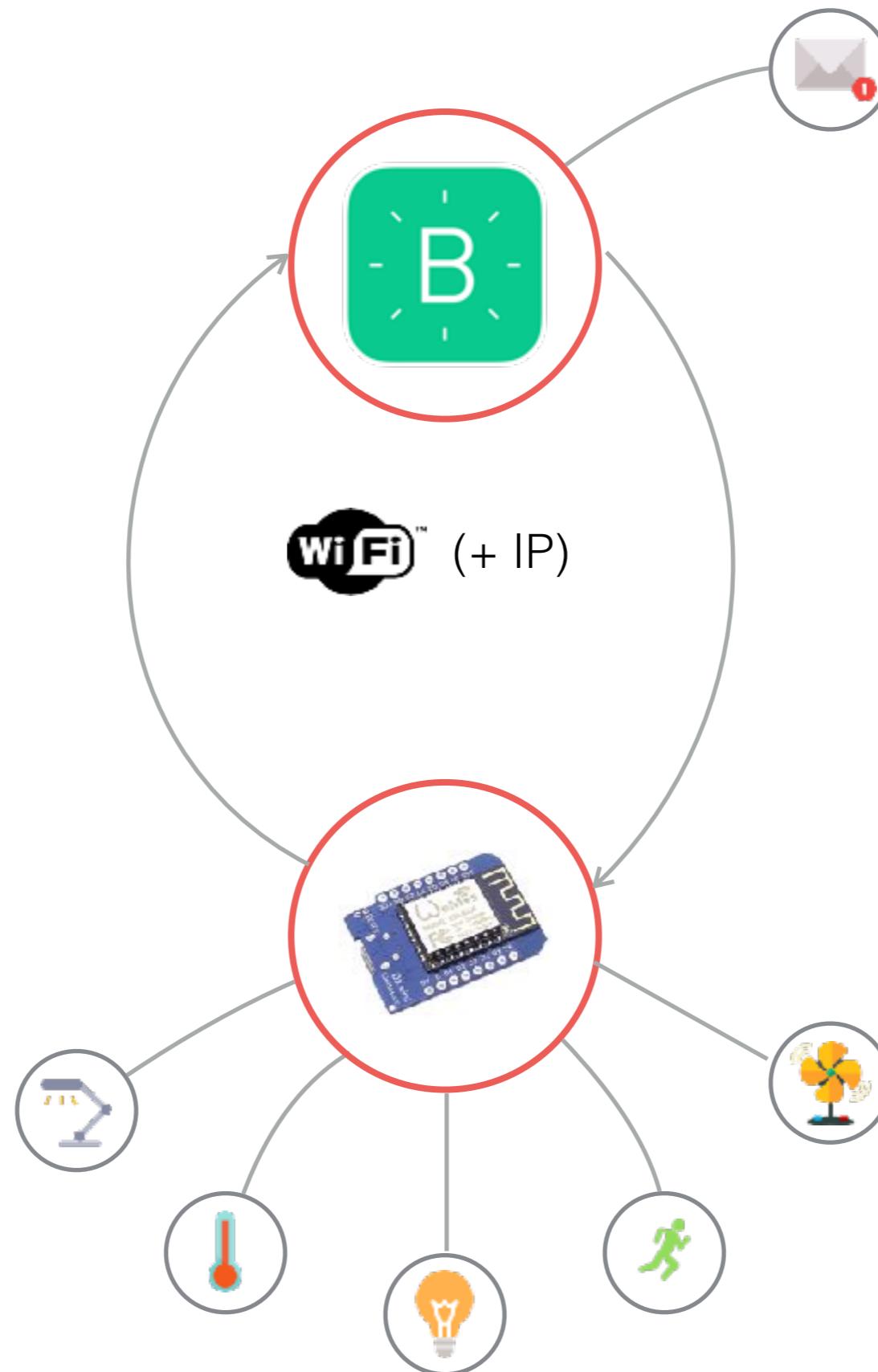
How IoT works (basically)



What we're using today



What we're using today



What we're using today

WeMos D1 mini

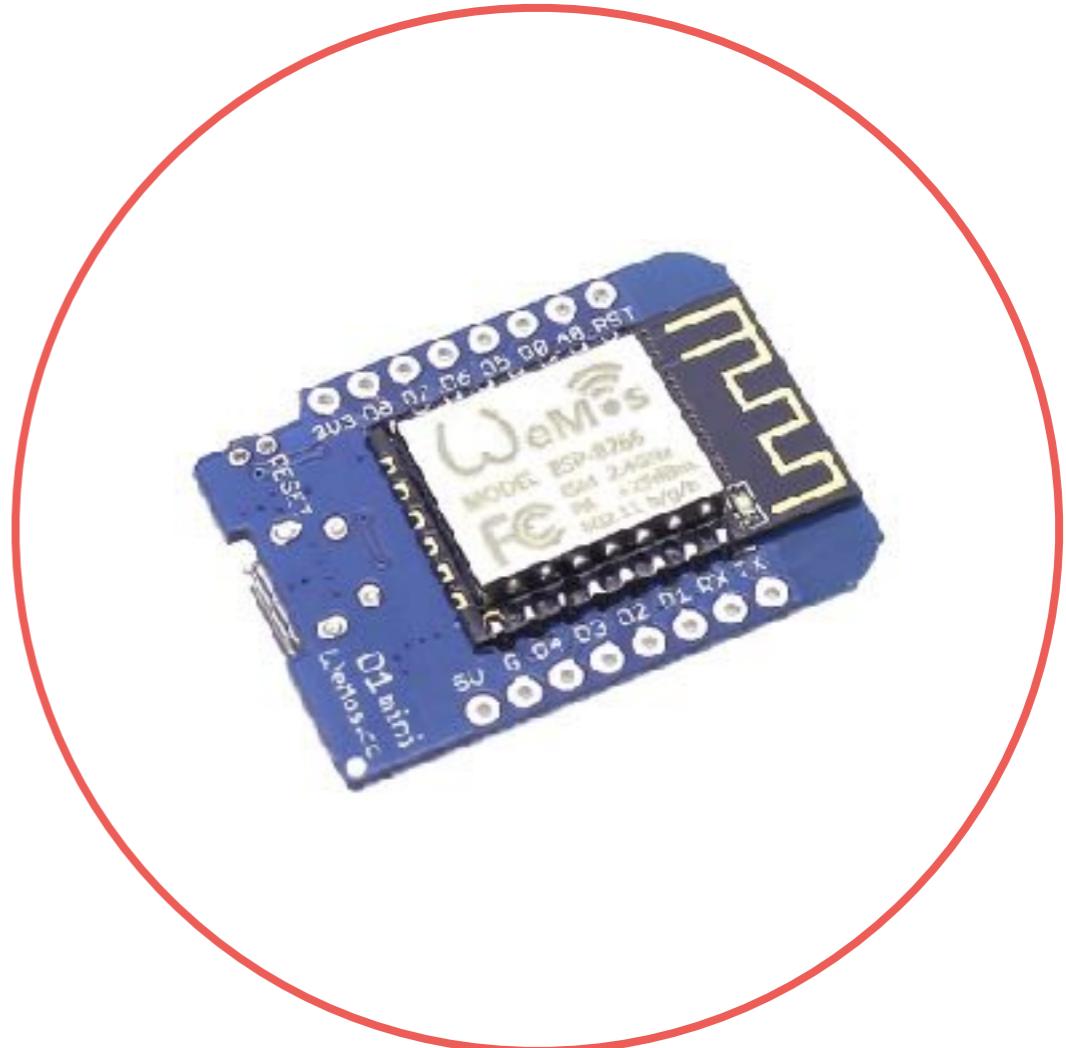
1. The *brains* of the operation.

Runs at 80 MHz, or about 20 times faster than the Apollo 11's computer 🌙

2. It's actually based on the ESP8266 (whatever)

3. It's got a tiny computer, Wi-fi and 8 digital inputs and outputs (you can read 8 different sensors or control 8 different things)

4. Cheaper than a Latte Macchiato ☕



What we're using today

Blynk

1. Neat mobile app that lets you quickly prototype with your phone

2. Lets you do a lot of cool stuff (*you'll see*)

3. Has a bunch of widgets and each widget uses 'Energy'

4. You get 2000 Energy to start, for more you need to pay up

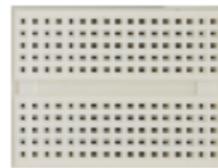


5. You won't need more than that today



What you'll learn today

1. How to use a breadboard



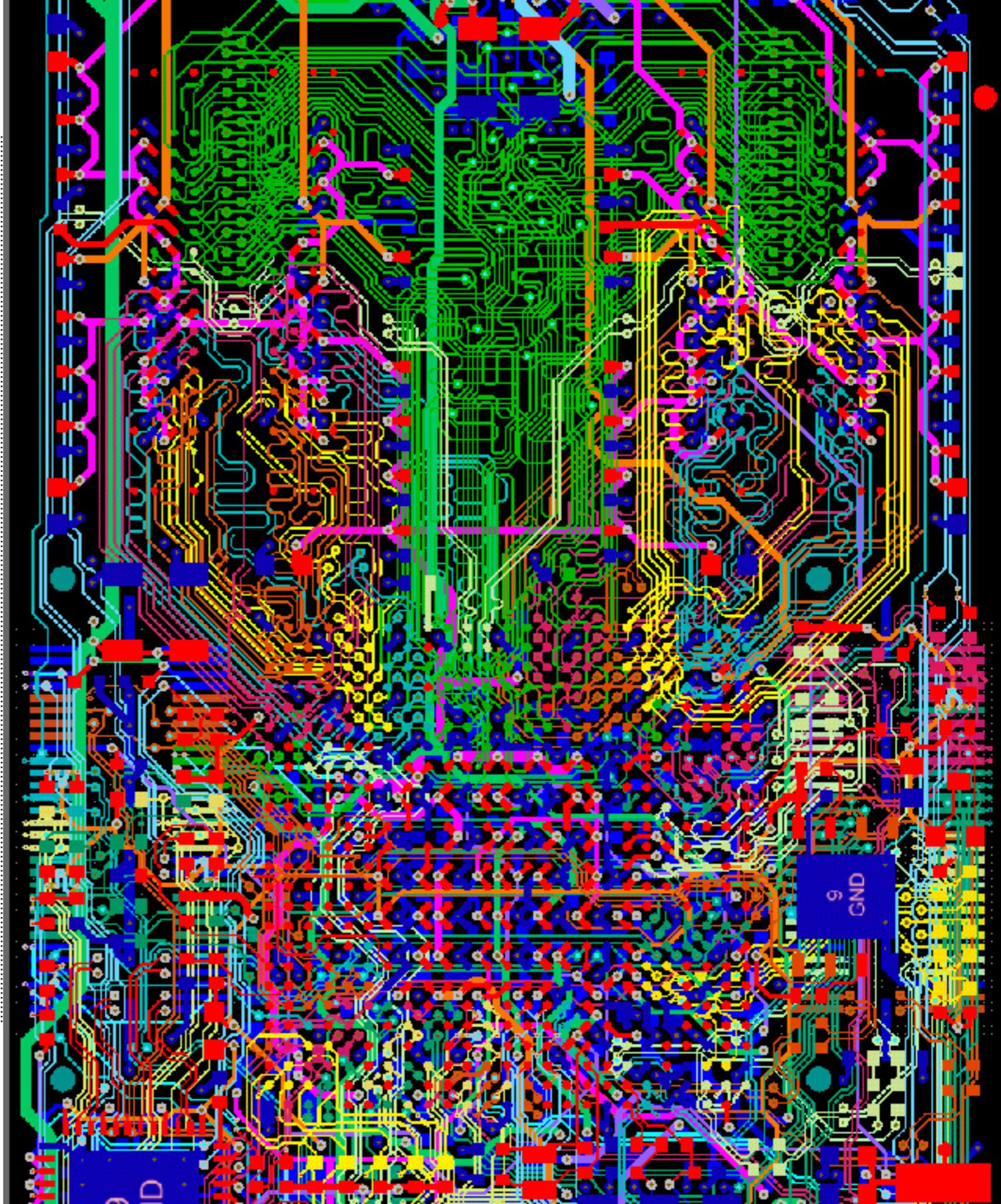
2. How to hook up and read sensors through your microcontroller



3. How to look at those data from *anywhere* (or sending it to the ☁)

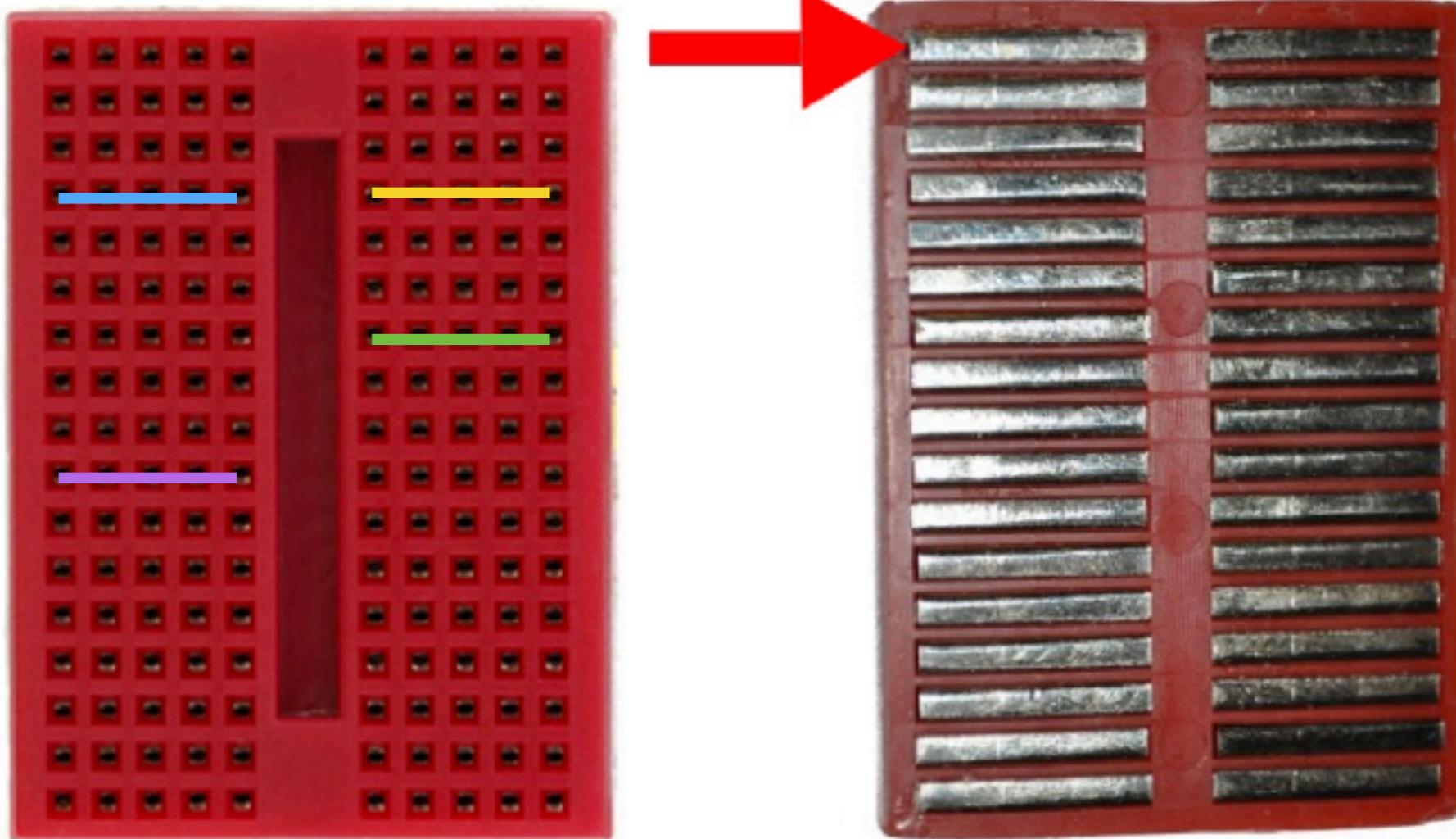
4. How to do *cool stuff* with those data and your smartphone

So it begins...

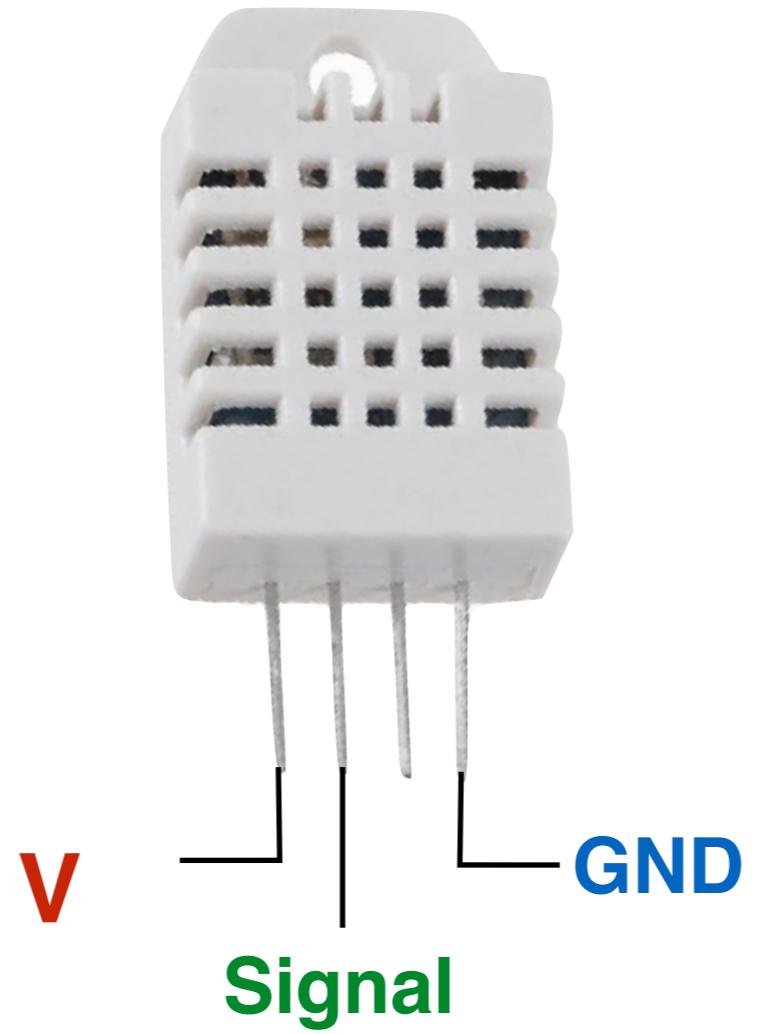


Breadboards

(pretty simple really)

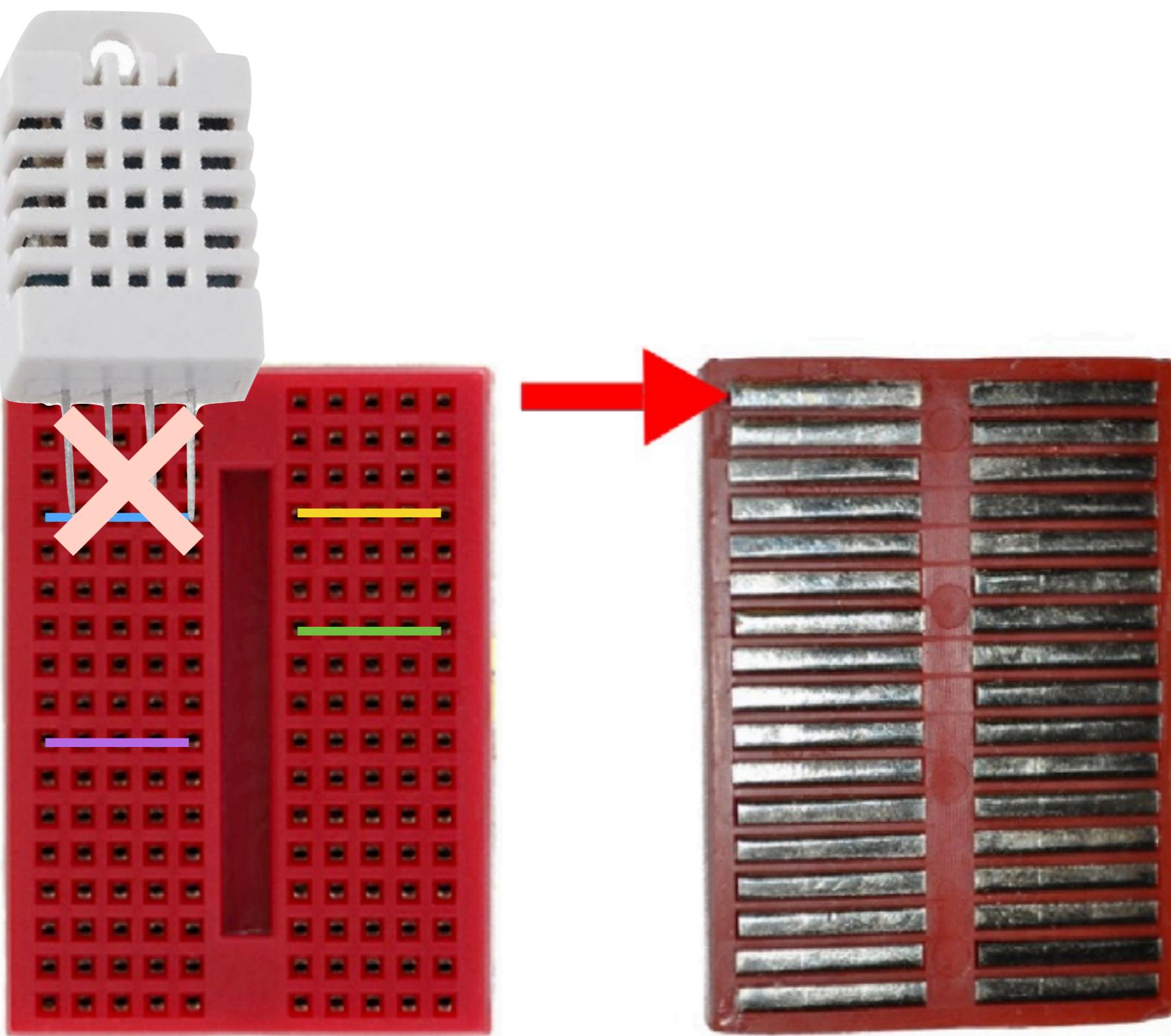


Humidity & temperature sensor



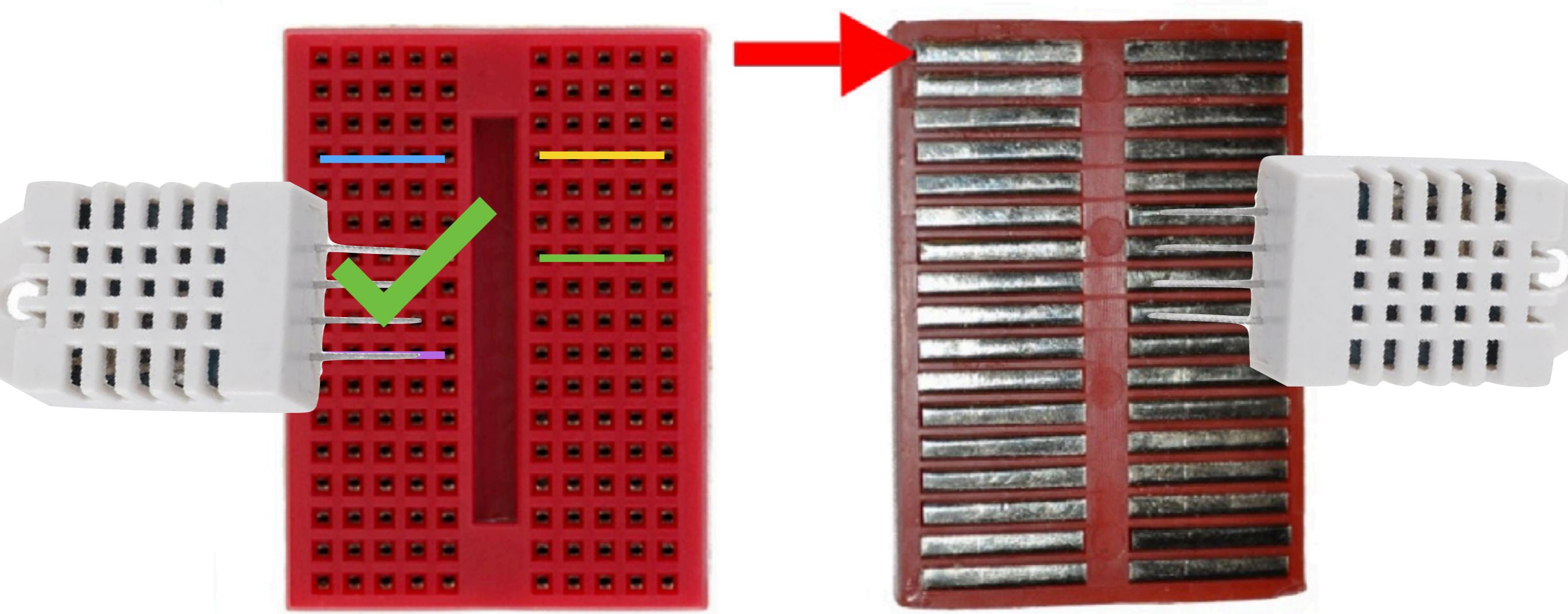
Breadboard

(pretty simple)



Breadboards

(pretty simple really)

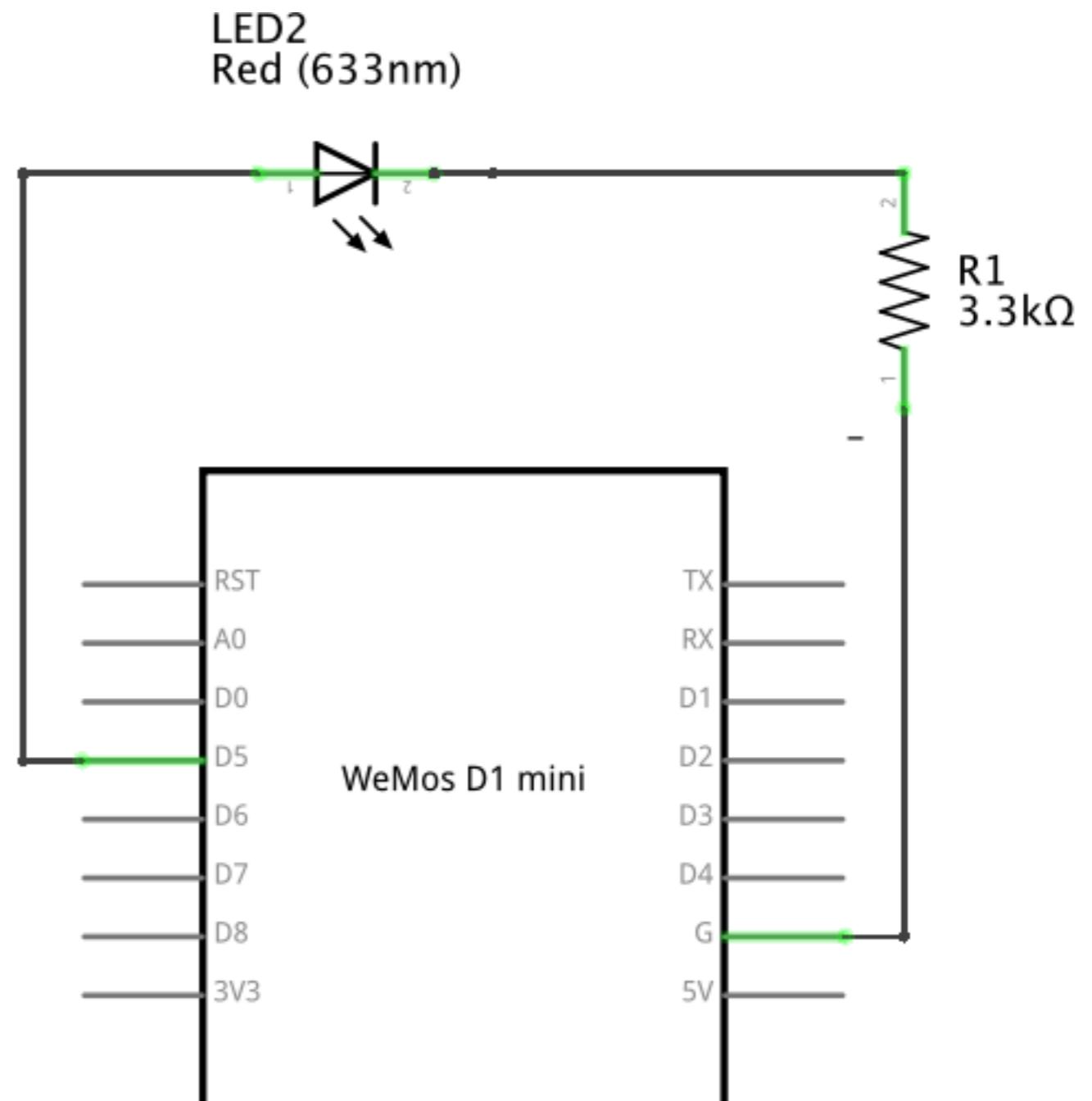


Task #1

Let there be light

Your first circuit

*Connecting the
Wemos to an LED*

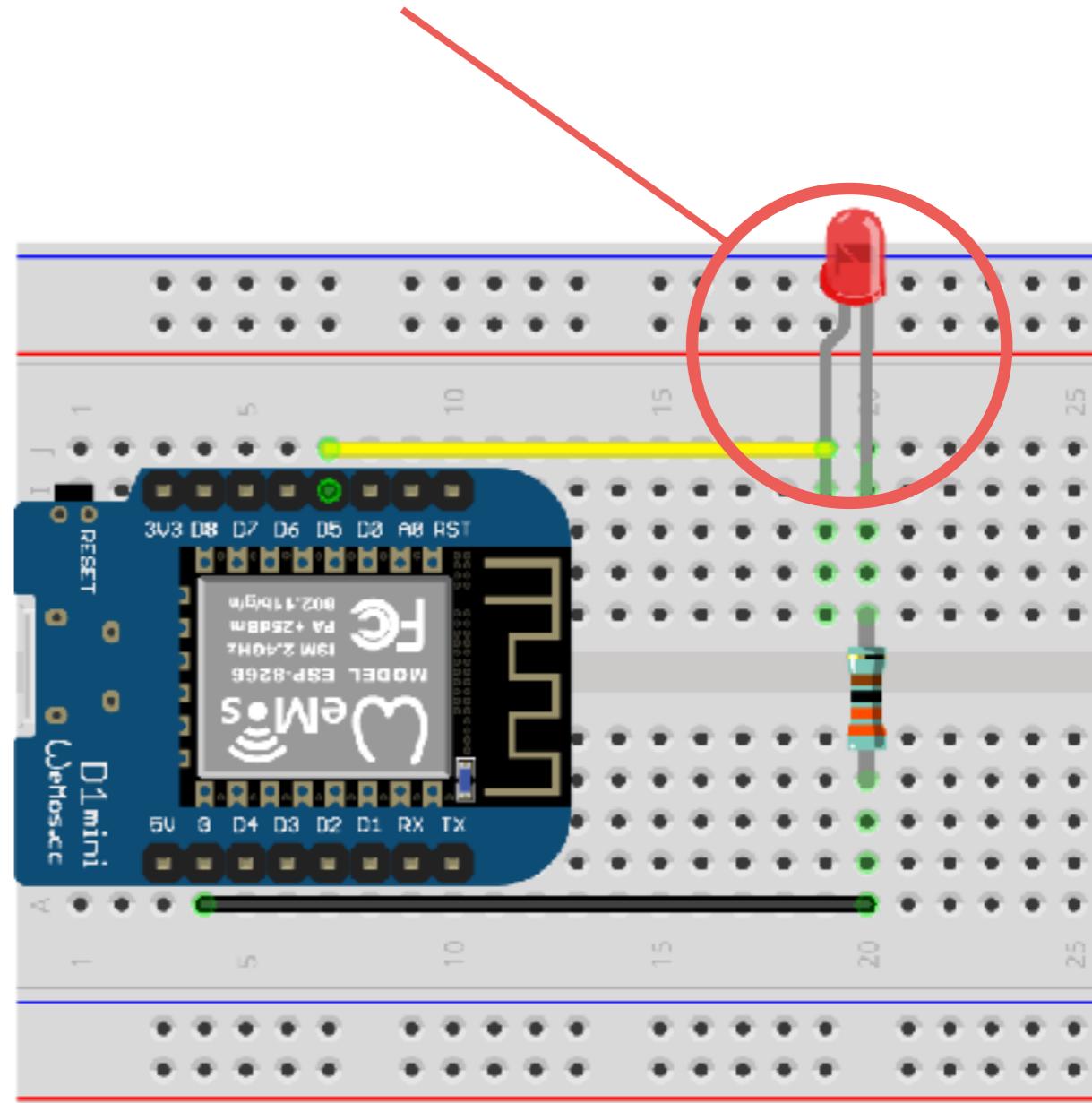


fritzing

Your first circuit

*Connecting the
Wemos to an LED*

Watch out for the LED!
It's a diode so current only flows one way.



Code examples

<https://github.com/WATTx/iot-workshop/archive/ggu.zip>

Your first program

Controlling the LED



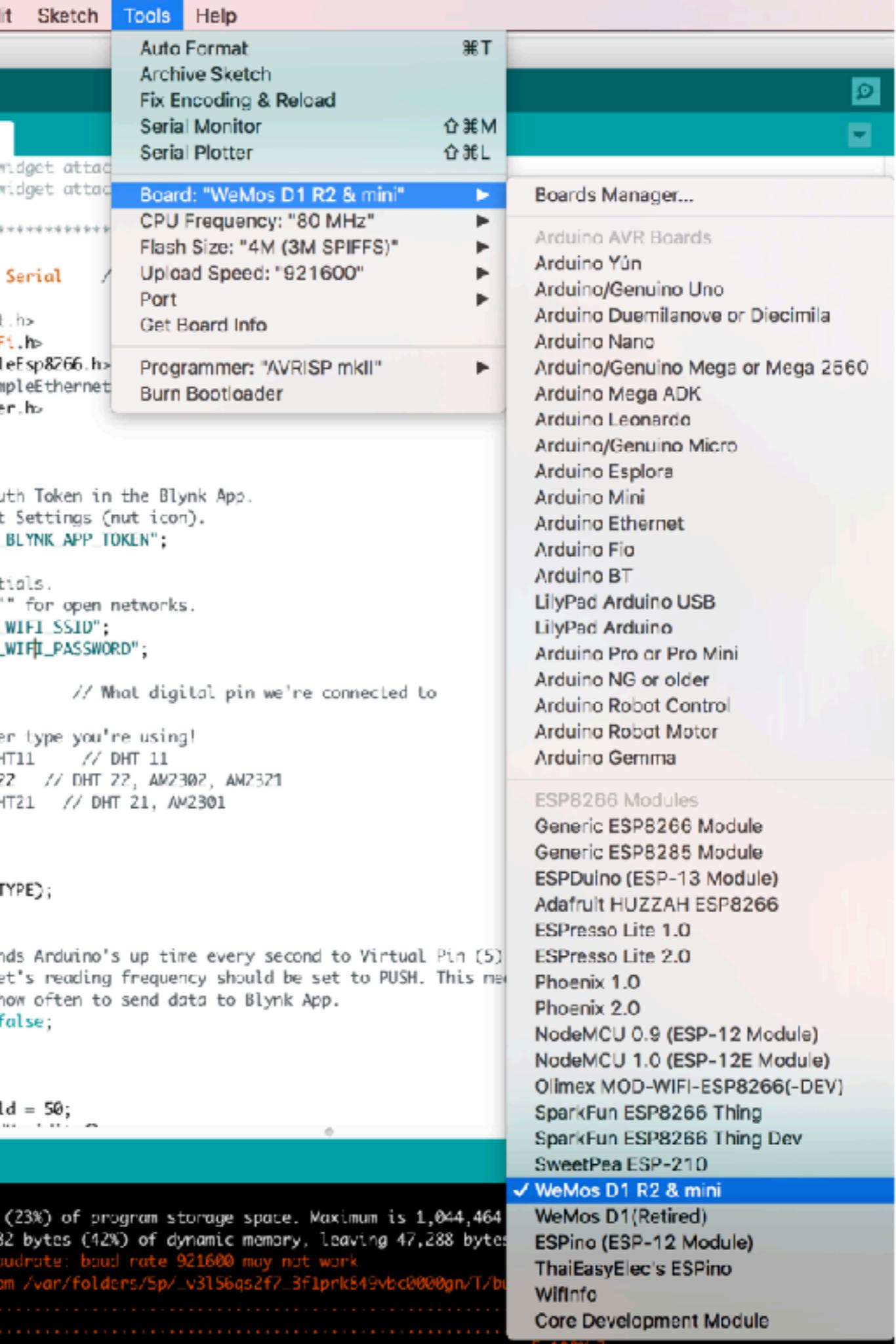
The screenshot shows the Arduino IDE interface with a sketch titled "ex1a". The code is as follows:

```
1 | 
2 int led = D5; // This is the pin where the LED is connected to
3 |
4 // the setup routine runs once when you press reset:
5 void setup() {
6     // initialize the digital pin as an output.
7     pinMode(led, OUTPUT);
8 }
9
10 // the loop routine runs over and over again forever:
11 void loop() {
12     digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
13 }
```

The status bar at the bottom indicates the board is a "WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on /dev/cu.wchusb0".

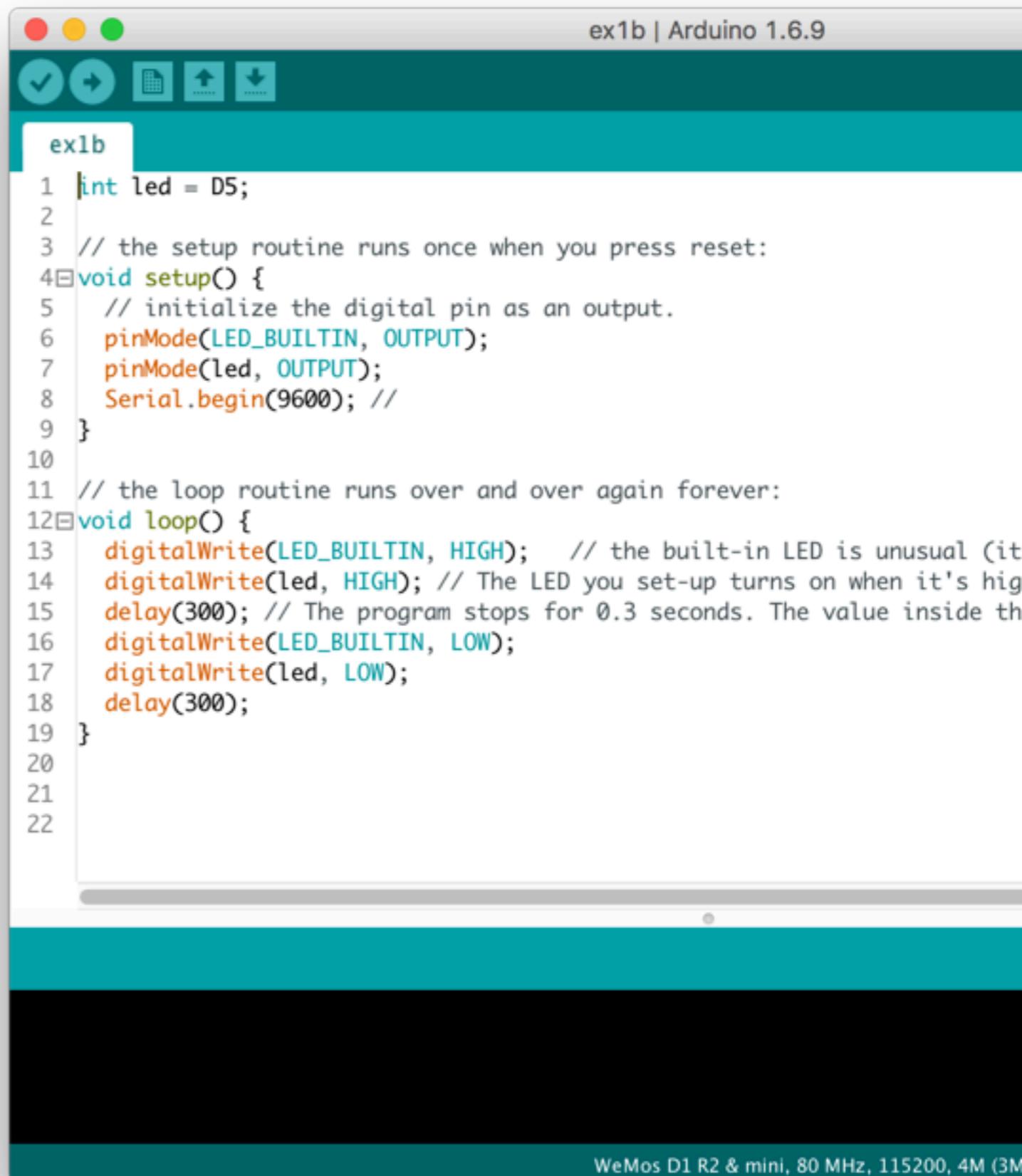
Programming it

1. Make sure the Wemos board is selected. Go to Tools > Board > and select "WeMos D1 R2 & mini"
2. Where it says **Upload Speed** select **115200**
3. On **Port** on a Mac or Linux select **/dev/cu.wchusbserial**. Windows should have something like **COM1**
4. Make sure your laptop is connected to the Wemos with a MicroUSB button and hit the Upload button. Hope it doesn't explode (*I'm kidding, it won't!*).
5. You should now have your very first program loaded onto the Wemos. Congrats!
6. Use the **Serial Monitor** with **Both NL & CR** and **9600 baud** to see the messages the Wemos sends to your computer



Your first program

The Wemos has its own LED as well



The screenshot shows the Arduino IDE interface with the title bar "ex1b | Arduino 1.6.9". The code editor window contains the following sketch:

```
int led = D5;

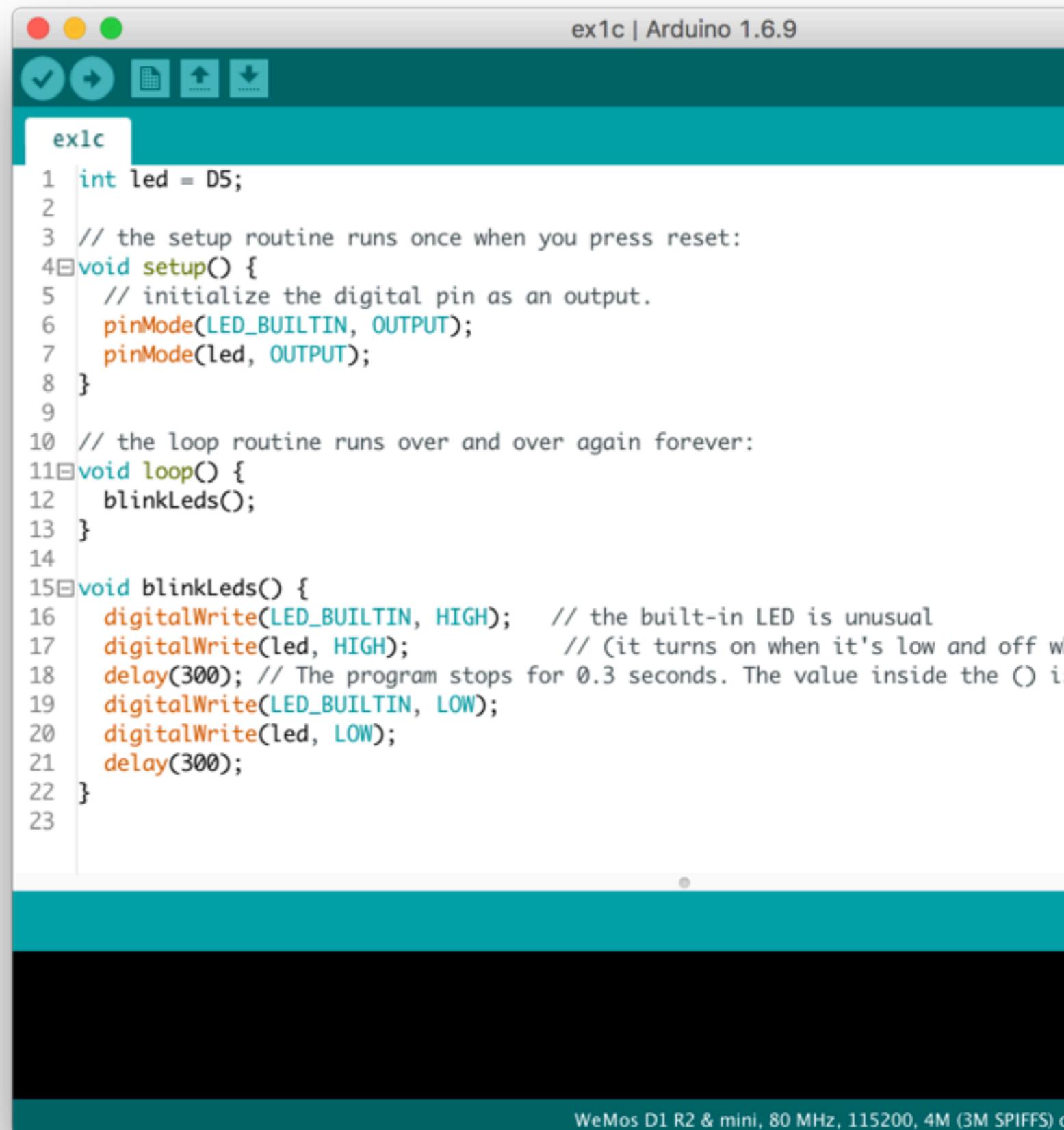
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(led, OUTPUT);
  Serial.begin(9600); // 
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // the built-in LED is unusual (it
  digitalWrite(led, HIGH); // The LED you set-up turns on when it's high
  delay(300); // The program stops for 0.3 seconds. The value inside the
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(led, LOW);
  delay(300);
}
```

The status bar at the bottom right indicates the board is "WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M)".

Functions

Let's wrap what we did in a function



```
int led = D5;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  blinkLeds();
}

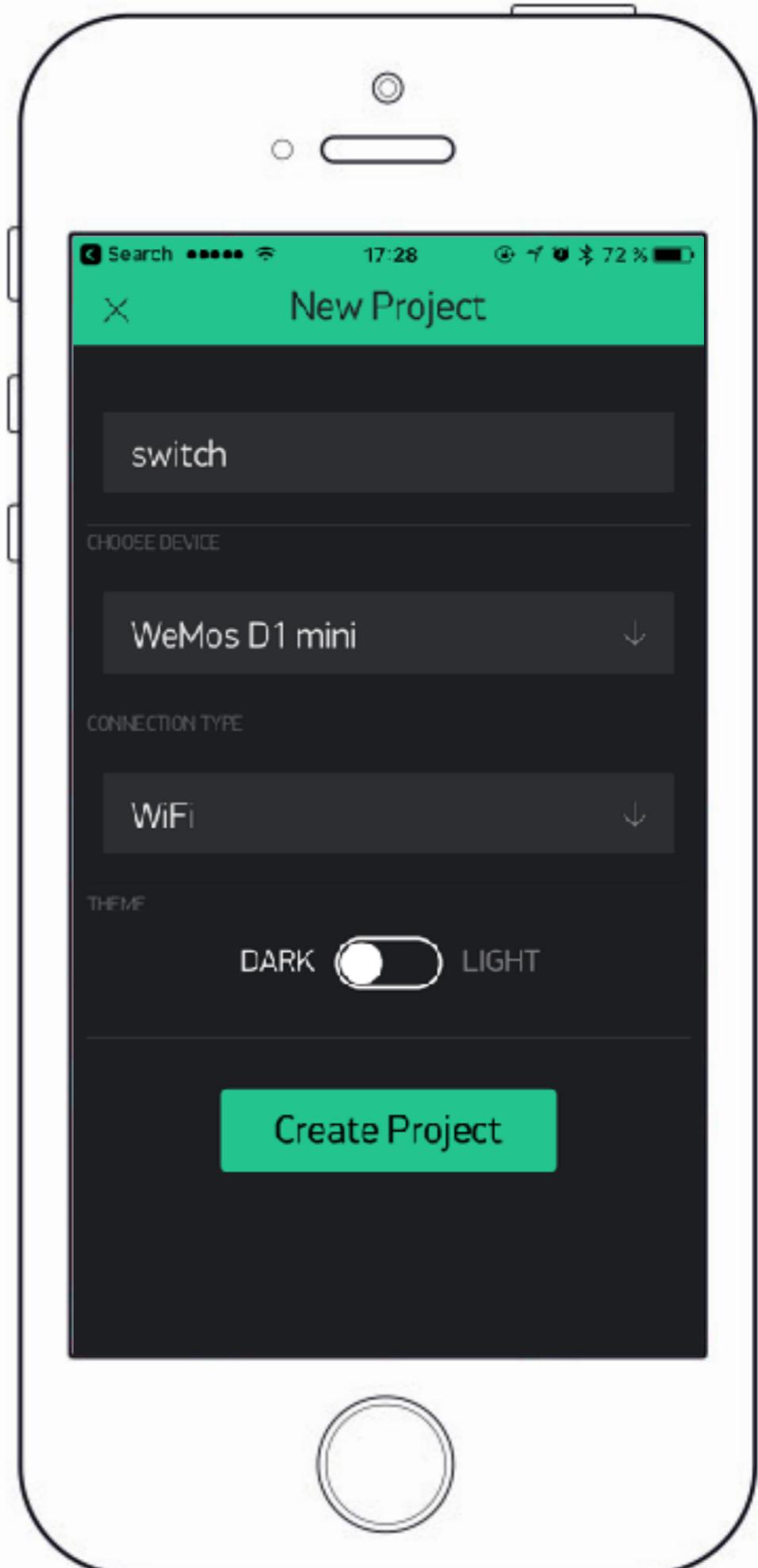
void blinkLeds() {
  digitalWrite(LED_BUILTIN, HIGH);      // the built-in LED is unusual
  digitalWrite(led, HIGH);             // (it turns on when it's low and off when it's high)
  delay(300); // The program stops for 0.3 seconds. The value inside the () is in milliseconds
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(led, LOW);
  delay(300);
}
```

WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS)

*Connecting the
light to the world.*

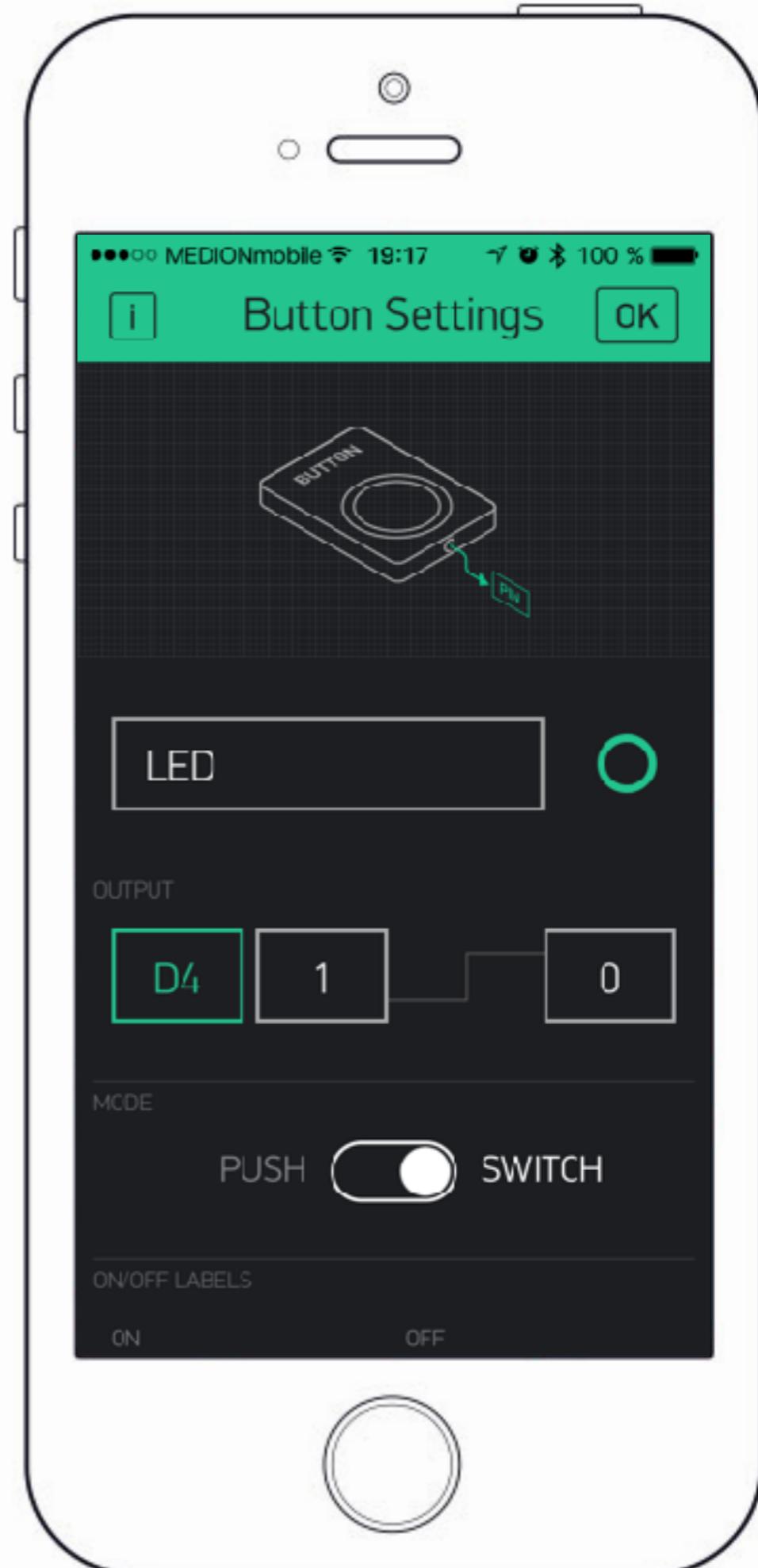
Your first Blynk project

1. Open the Blynk app and hit 'Create New Project'
2. Give it some name (here it's 'switch')
3. Tap 'Choose Device' and scroll all the way down until you find the 'WeMos D1 Mini'
4. Hit 'Create Project' and *boom*, you got your first project!
5. Your auth token has been sent to your mail. You can always check Settings to see your token



Your first Blynk project

1. You're now gonna control a Digital I/O . Add a Button and tap it
2. Because you connected the LED to D5, select D5 on the Digital Output. Tap OK.
3. The Wemos has a blue LED that's controlled by the D4 pin. Add another button and choose that pin. Name it Wemos LED (or whatever you want) Change the 0 to 1 and the 1 to 0 (like in the screenshot) cause otherwise it's just gonna be on when off and off when on, and then it just gets weird.
4. You can choose Push or Switch, it's cool either way.



Programming it

1. Fire up the Arduino app.
2. Download (or copy-paste) the **ex1d.ino** file and open it with the app.
3. Around line 27*, copy your token from the Blynk app and replace in between the “ “, in the **auth** variable.
4. Do the same for the Wifi.
5. We tell the Wemos how to connect to the Wifi using **Blynk.begin(auth, ssid, pass)** then we use **Blynk.run()** to actually connect.
6. Flash it! Congrats, you can now use your Wemos as a flashlight 
7. Don't forget to tap  in the Wemos app to start it



```
8 * Blynk community: http://community.blynk.cc
9 * Social networks: http://www.fb.com/blynkapp
10 *
11 *
12 * Blynk library is licensed under MIT license
13 * This example code is in public domain.
14 *
15 *
16 ****
17
18 #define BLYNK_PRINT Serial // Comment this out to disable prints and save space
19 #include <ESP8266WiFi.h>
20 #include <BlynkSimpleEsp8266.h>
21 #include <Wire.h>
22
23 int led = D5;
24
25 // You should get Auth Token in the Blynk App.
26 // Go to the Project Settings (nut icon).
27 char auth[] = "get_it_from_your_app_or_check_your_email";
28
29
30 // Your WiFi credentials.
31 // Set password to "" for open networks.
32 char ssid[] = "WATTx_WiFi";
33 char pass[] = "check_the_screen";
34
35
36 void setup()
37 {
38   Serial.begin(9600);
39   Blynk.begin(auth, ssid, pass);
40 }
41
42
43
44 void loop()
45 {
46   Blynk.run(); // Initiates Blynk
47 }
```

Done Saving.

Uploading 240016 bytes From /var/Folders/5p/_v3156qs2f7_3f1prk849vbc0000gn/T/buildc65a535...

[34%]
[68%]
[100%]

34 WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on /dev/c...

*Go to Arduino > Preferences and  “Display line numbers”

So you can turn on a light anywhere in
the world with your smartphone... *Cool!*

Now let's do something actually useful.

Task #2

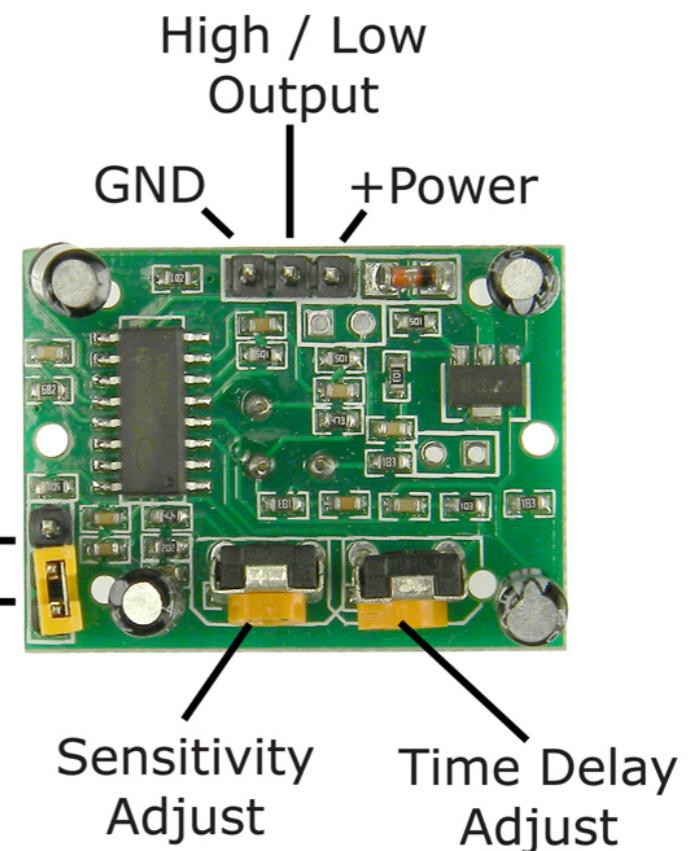
Motion detection



The motion sensor

*Uses infrared
waves to
detect people*

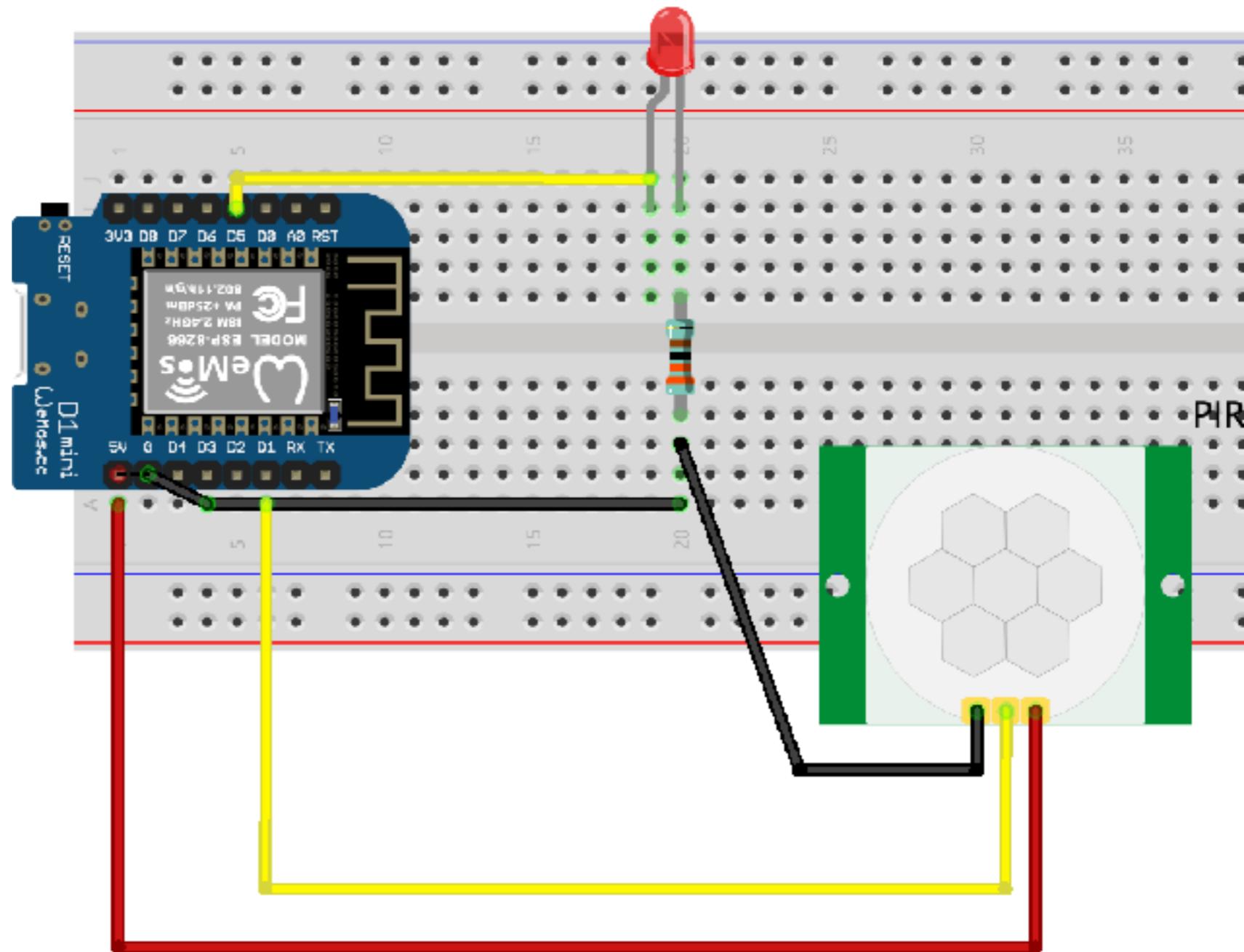
Jumper Set:
H: Repeat Trigger
L: Single Trigger



More info

Your second circuit

Adding the motion sensor



Your second program

1. So we're gonna detect people with this sensor. Cool! How?
2. Let's use an `if-else` statement to trigger a warning if movement is detected
3. We're also using `Serial` to send messages back to the computer

```
ex2a | Arduino 1.6.9

ex2a
1 int led = D5;
2 int motionSensorPin = D1;
3 int motionState;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7     // initialize the digital pin as an output.
8     pinMode(LED_BUILTIN, OUTPUT);
9     pinMode(led, OUTPUT);
10    Serial.begin(9600); // We'll use Serial to talk with the computer
11 }
12
13
14 // the loop routine runs over and over again forever:
15 void loop() {
16
17     motionState = digitalRead(motionSensorPin); // Reading the state of the sensor
18
19     if (motionState == LOW) // if value is LOW, then no motion has been detected
20     {
21         Serial.println("No motion detected");
22     }
23     else // if value is something other than LOW then it must be HIGH meaning motion
24     {
25         Serial.println("Motion detected!! Someone is here!");
26         blinkLeds();
27         digitalWrite(LED_BUILTIN, HIGH); // Turn off the built-in LED
28     }
29     delay(500);
30 }
31
32
33 void blinkLeds() {
34 }
```

Done uploading.

[35%]
[71%]
[100%]

WeMos D1 R2 & mini 80 MHz, 115200, 4

*What about an
Internet connected
motion sensor that
warns you when
you're away?*

Great idea!

1. Let's combine our last program with the Blynk code

2. Again, add the Wifi credentials and your token to your code

3. Let's send a notification! This function is part of the Blynk library and sets off a notification on your phone:

```
Blynk.notify("Your message here");
```

4. Flash it! ⚡

5. Don't forget to add a Notification widget on your Blynk app



```
ex2b | Arduino 1.6.9

ex2b
27
28 // You should get Auth Token in the Blynk App.
29 // Go to the Project Settings (nut icon).
30 char auth[] = "get_it_from_your_app_or_check_your_email";
31
32 // Your WiFi credentials.
33 // Set password to "" for open networks.
34 char ssid[] = "WATTx_WiFi";
35 char pass[] = "check_the_screen";
36
37 void setup()
38 {
39   pinMode(LED_BUILTIN, OUTPUT);
40   pinMode(led, OUTPUT);
41   Serial.begin(9600); // We'll use Serial to talk with the computer
42   Blynk.begin(auth, ssid, pass);
43 }
44
45
46 void loop()
47 {
48   Blynk.run(); // Initiates Blynk
49   motionState = digitalRead(motionSensorPin); // Reading the state of the sensor
50
51   if (motionState == LOW) // if value is LOW, then no motion has been detected
52   {
53     Serial.println("No motion detected");
54   }
55   else // if value is something other than LOW then it must be HIGH meaning motion has been detected
56   {
57     Serial.println("Motion detected!! Someone is here!");
58     Blynk.notify("ALARM! Your motion detector has sensed someone!");
59     // This will trigger a notification on your phone
60
61     blinkLeds();
62     digitalWrite(LED_BUILTIN, HIGH); // Turn off the built-in LED
63   }
64   delay(2000);
65 }
```

Done Saving.

[34%] [68%] [100%]

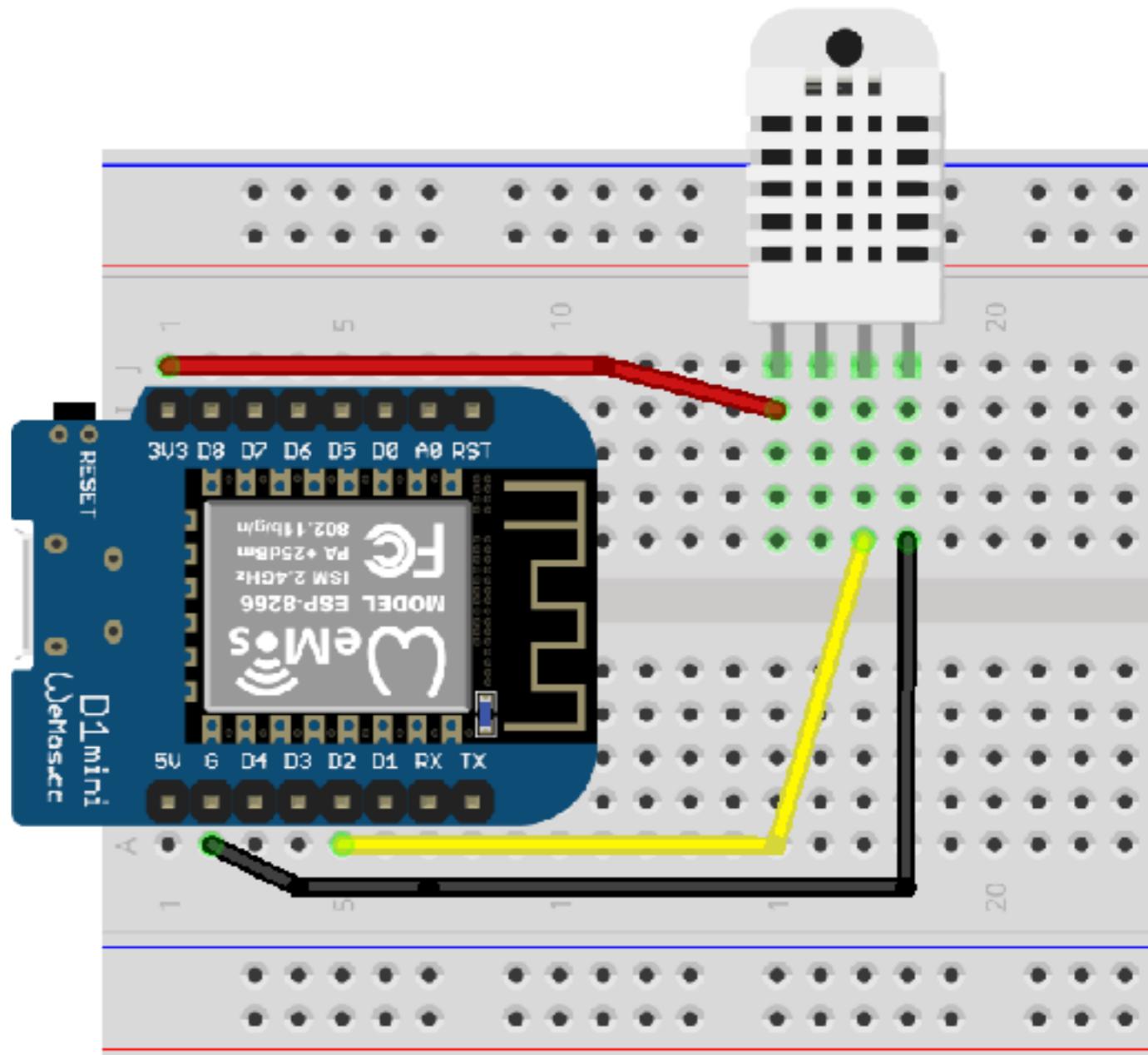
54 WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on /dev/cu.wchusbse

Task #3

*Smart temperature &
humidity sensor*

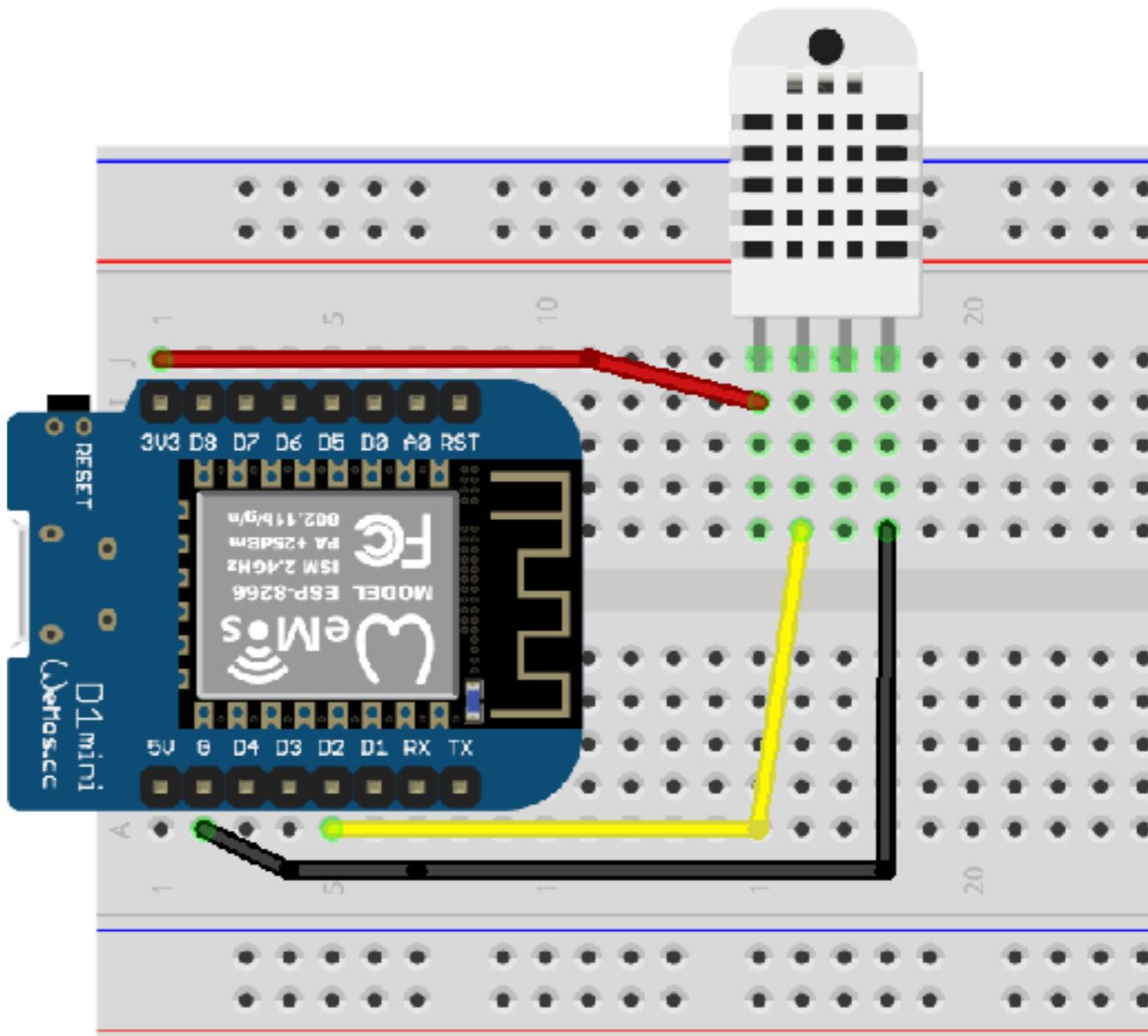
Your third circuit

Wifi Temperature & humidity sensor



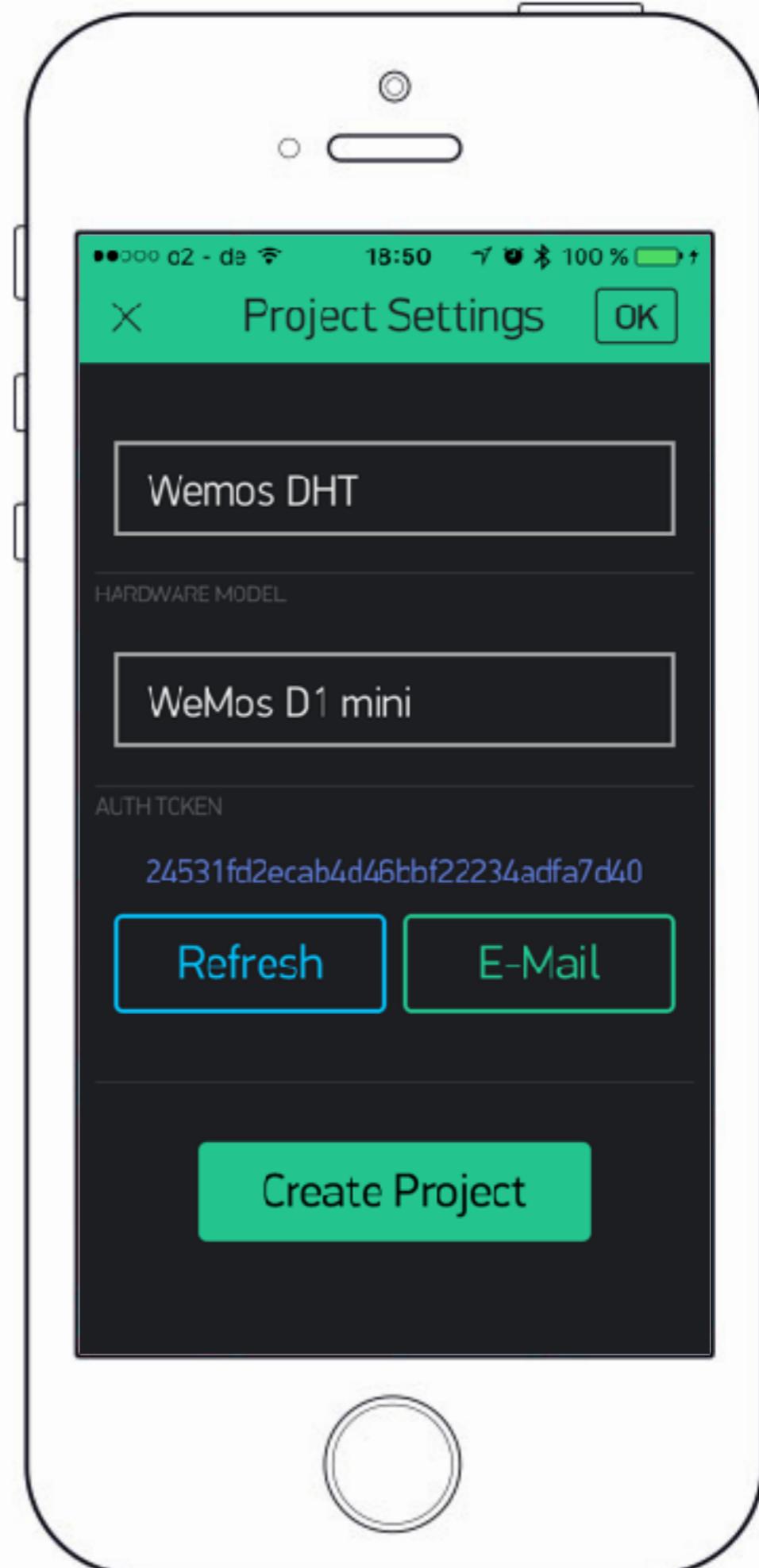
Your third circuit

1. Plug in the Wemos at the edge of the breadboard (leave a row free on each side)
2. Connect the DHT sensor somewhere on the breadboard
3. Wire'em up like shown in the diagram
4. Use a MicroUSB cable and connect the Wemos to your laptop



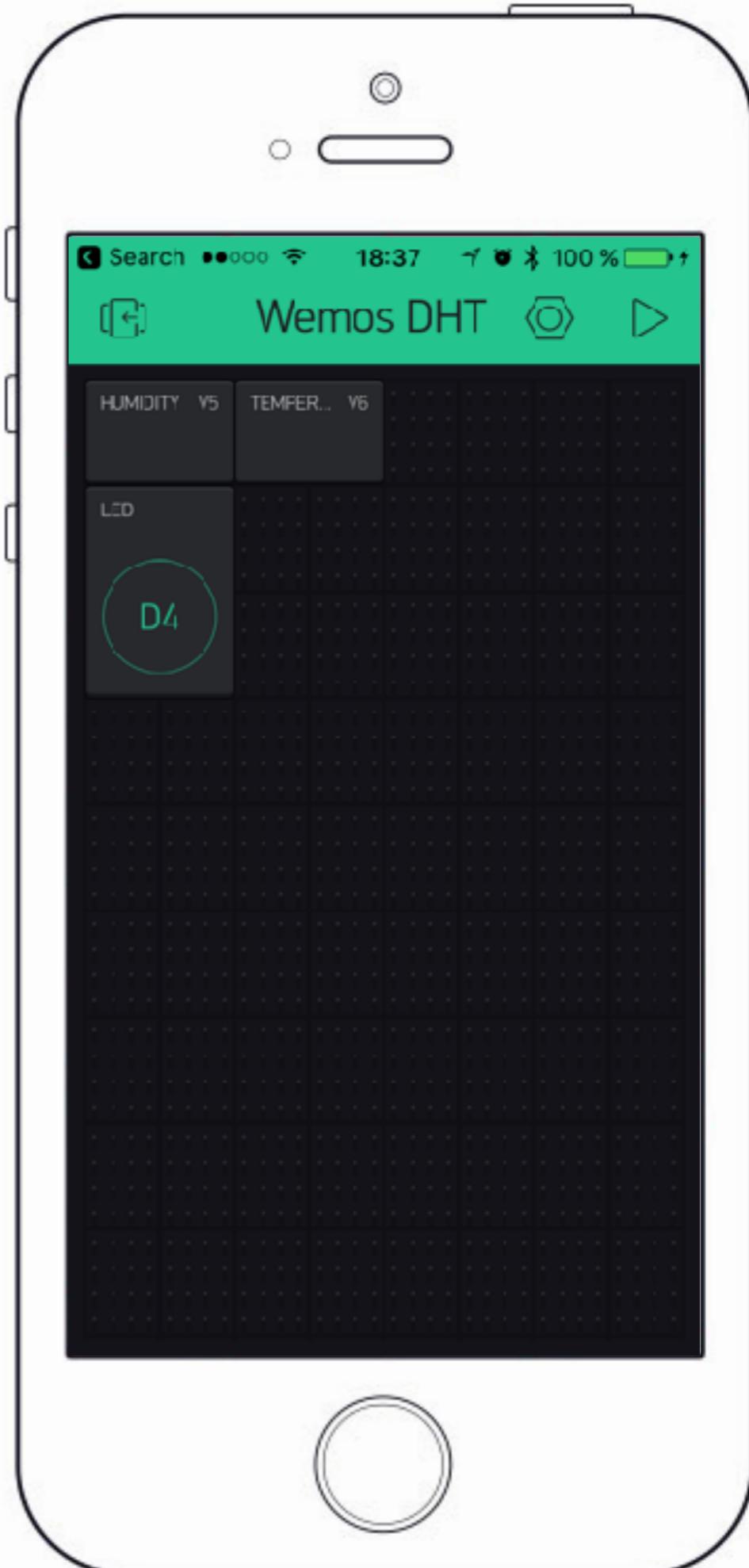
Your second Blynk project

1. Open the Blynk app and hit 'Create New Project'
2. Let's give it a new name
3. Same thing, tap the 'Hardware Model' and scroll all the way down until you find the 'WeMos D1 Mini'
4. You know the drill



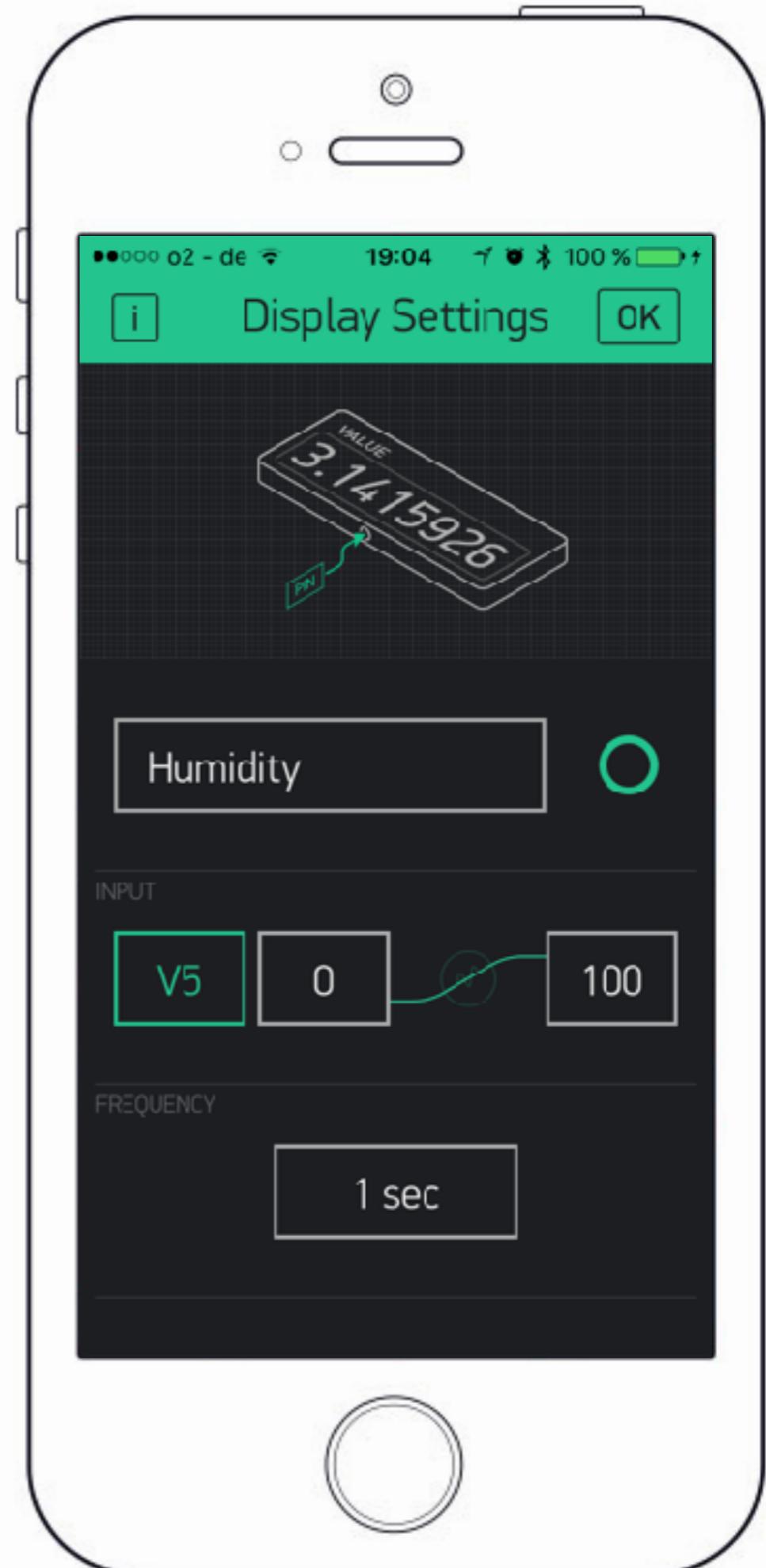
Your second Blynk project

1. Tap anywhere on the black mat
2. You'll see a bunch of components. Scroll down and tap Value Display (costs 200 ⚡)
3. Get another one of those



Your second Blynk project

1. Tap the first Value Display.
2. You're gonna read Humidity with this one. So change the name to Humidity 😊
3. On the Input, go to Virtual Pins and select V5. This is where your Wemos will write its humidity values.
4. Because humidity varies between 0 and 100%, change 1024 to 100 😊
5. Do the same thing for Temperature, and use V6, with whatever values you like



Your new program

1. This time we're sending values to the Blynk app.

- ## 2. We're using

`Blynk.virtualWrite()` to write temperature and humidity measurements to Blynk's virtual variables that we set up on the app.

3. If your sensor is blue, change where it says DHT22 (line 50) to DHT11

4. Flash the Wemos with the new code.



The screenshot shows the Arduino IDE interface with the title bar "ex3a | Arduino 1.6.9". The main area displays the following C++ code for a Blynk DHT example:

```
47
48
49 #define DHTPIN D2          // What digital pin we're connected to
50 #define DHTTYPE DHT22    // DHT 22, AM2302, AM2321
51
52 DHT dht(DHTPIN, DHTTYPE);
53 SimpleTimer timer;
54
55
56 // This function sends Arduino's up time every second to Virtual Pin (5).
57 // In the app, Widget's reading frequency should be set to PUSH. This means
58 // that you define how often to send data to Blynk App.
59 void sendSensor()
60 {
61     float h = dht.readHumidity();
62     float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
63     if (isnan(h) || isnan(t)) {
64         Serial.println("Failed to read from DHT sensor!");
65         return;
66     }
67
68     // You can send any value at any time.
69     // Please don't send more than 10 values per second.
70     Blynk.virtualWrite(V5, h);
71     Blynk.virtualWrite(V6, t);
72     Serial.println(h);
73     Serial.println(t);
74
75 }
76
77 void setup()
78 {
79     Serial.begin(9600); // See the connection status in Serial Monitor
80     Blynk.begin(auth, ssid, pass);
81
82     dht.begin();
83     // Setup a function to be called every second
84     timer.setInterval(2000L, sendSensor);
85
86 }
```

At the bottom of the screen, a message "Done Saving." is displayed.

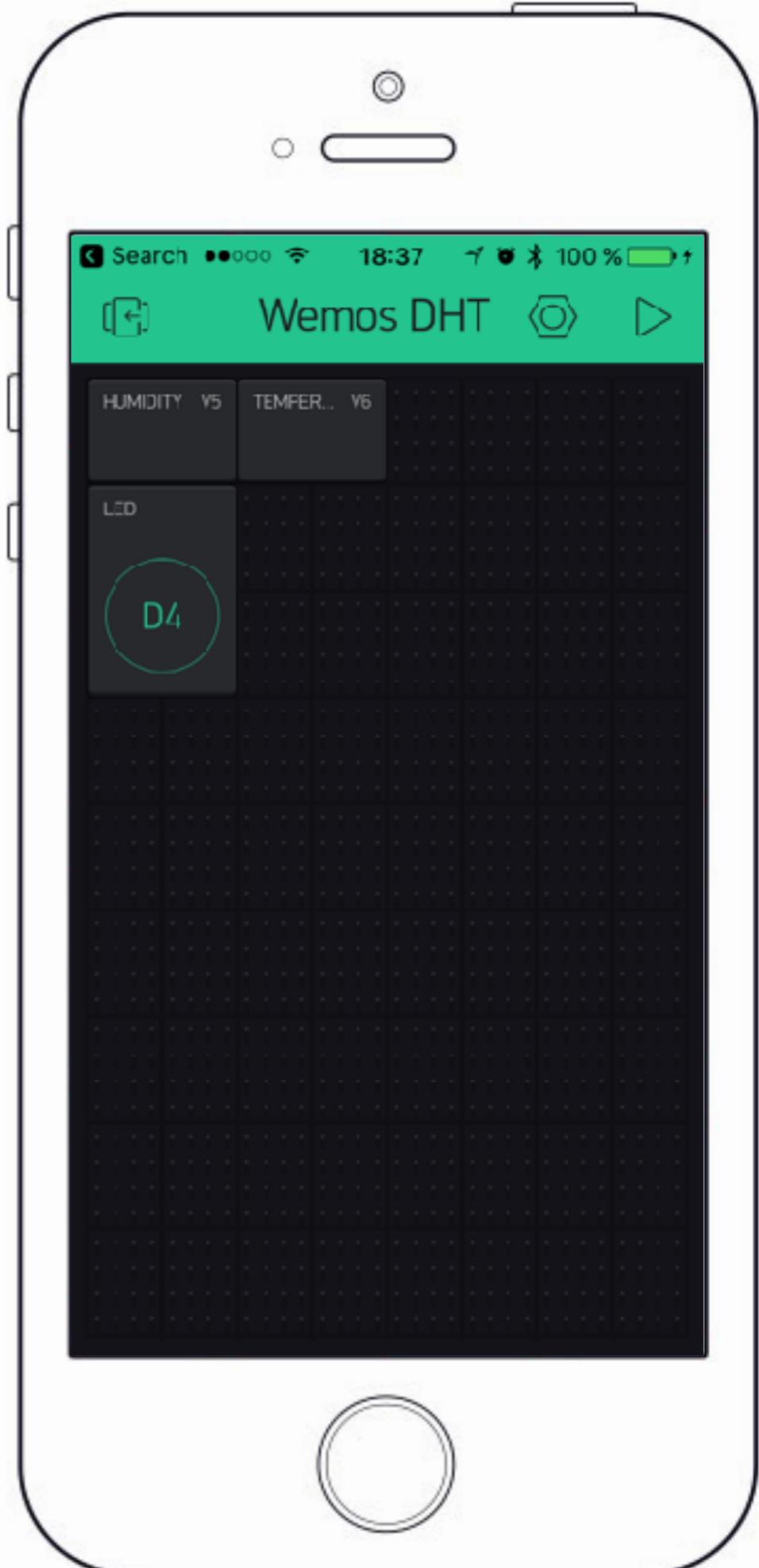
Back in the app

1. Hit the  button

2. You now have your very own connected & super smart humidity & temperature sensor.

Awesome job!

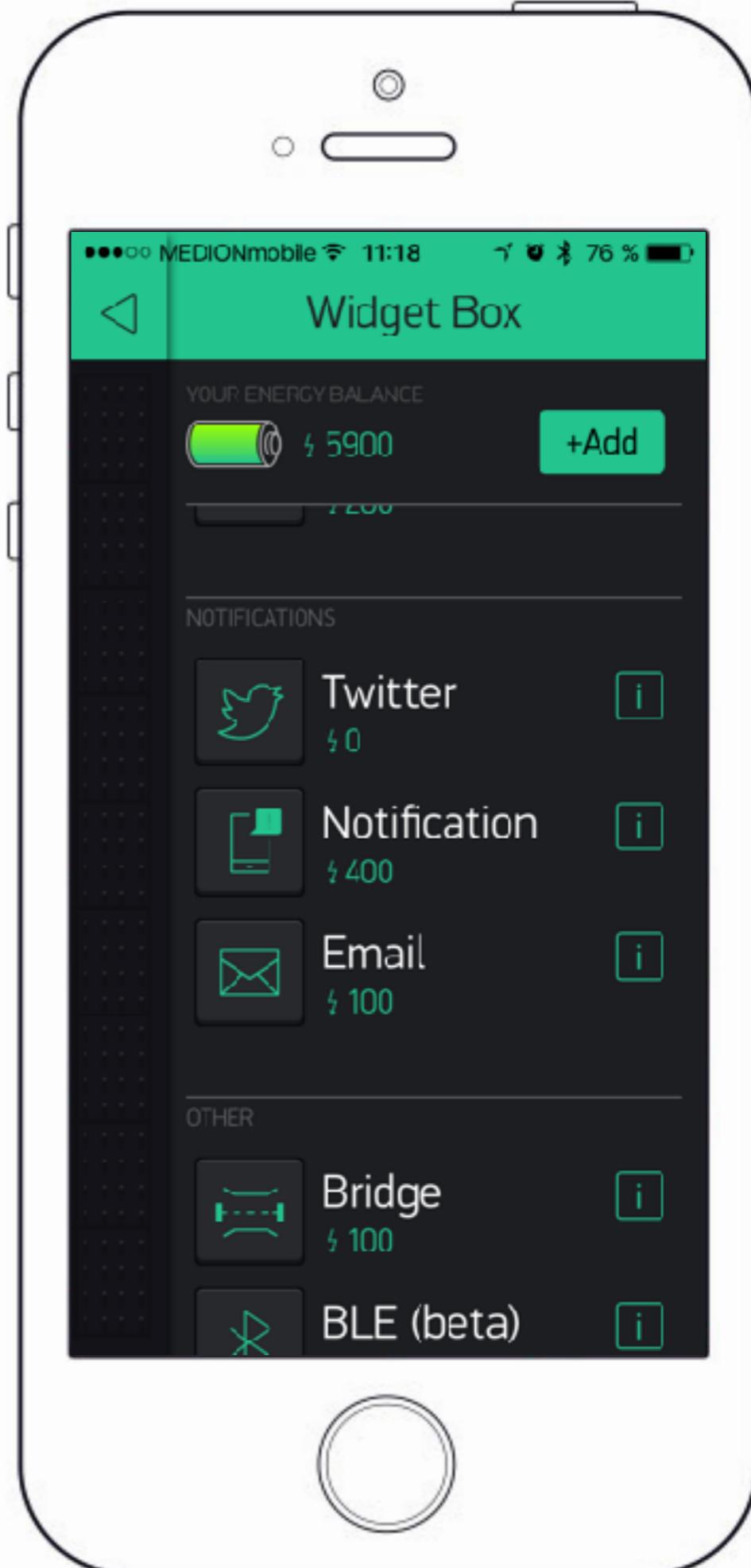
3. Check your app! You should be getting some readings by now



More notifications!

1. Let's set up a new notification widget

2. What to do next? Maybe make something useful with the humidity measurements somehow?



Back to the Arduino app

1. So you wanna get notified when humidity gets too high? *Smart!*

2. Go to line 69

3. *Time to code:* let's use an `if` to send a notification if humidity is higher than 60% then...

4. Let's use Blynk's `notify` function again to trigger a notification on your phone: `Blynk.notify("Your message here");`

5. Remind yourself to open a window or something 💨

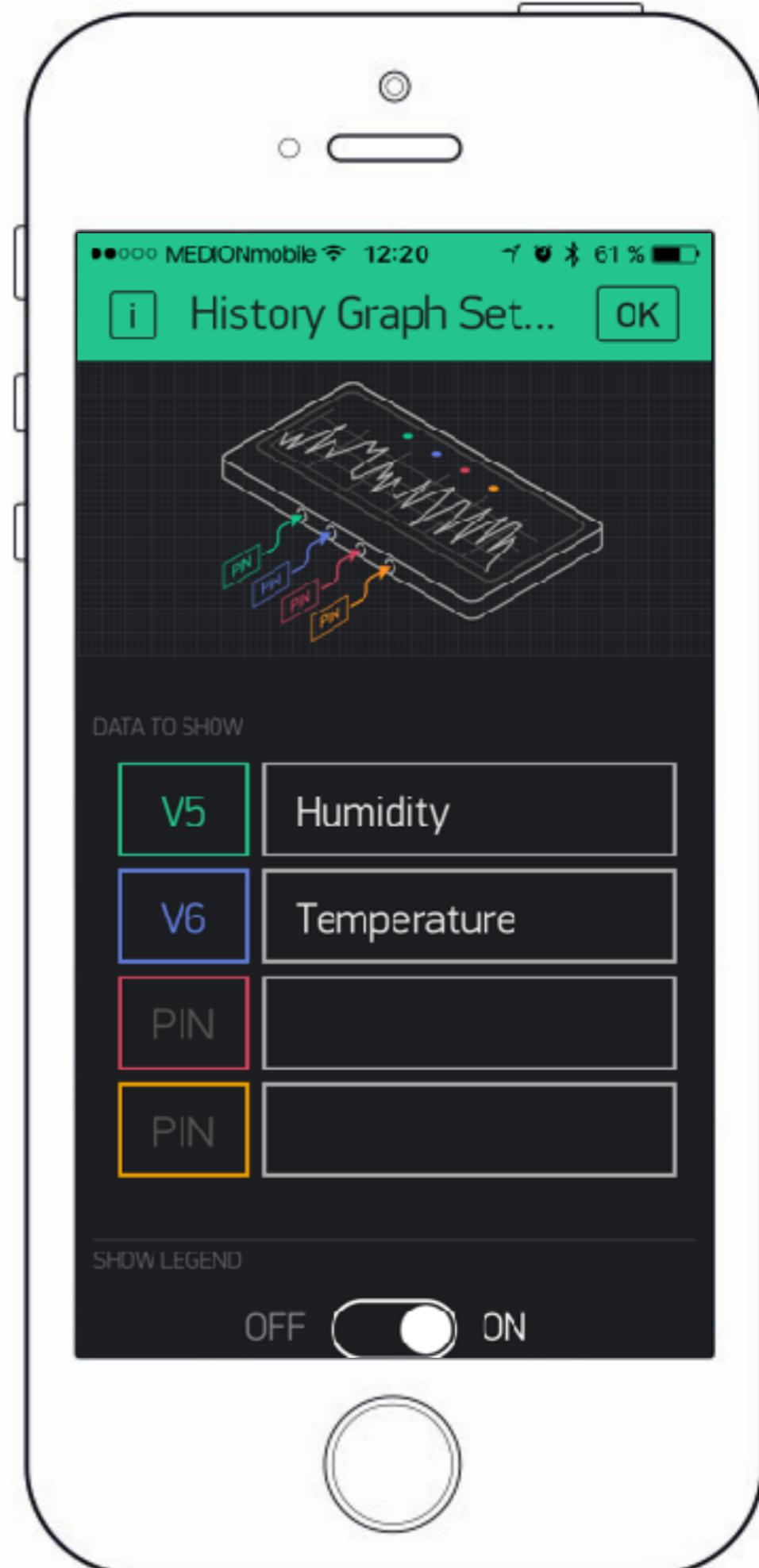
6. Flash it! ⚡

```

blynk_DHT.ino
30 #include "DHT.h" // DHT22, AM2302, AM2321
51
52 DHT dht(DHTPIN, DHTTYPE);
53 SimpleTimer timer;
54
55
56 // This function sends Arduino's up time every second to Virtual Pin (5).
57 // In the app, Widget's reading frequency should be set to PUSH. This means
58 // that you define how often to send data to Blynk App.
59 void sendSensor()
60 {
61     float h = dht.readHumidity();
62     float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
63     if (isnan(h) || isnan(t)) {
64         Serial.println("Failed to read from DHT sensor!");
65         return;
66     }
67
68 // Uncomment to play around with notifications
69 // if (h > 60) {
70 //     Blynk.notify("Your humidity is pretty high. Consider opening a window.");
71 // }
72
73 // You can send any value at any time.
74 // Please don't send more than 10 values per second.
75 Blynk.virtualWrite(V5, h);
76 Blynk.virtualWrite(V6, t);
77 Serial.println(h);
78 Serial.println(t);
79
80
81 }
82
83 void setup()
84 {
85     Serial.begin(9600); // See the connection status in Serial Monitor
86     Blynk.begin(auth, ssid, pass);
87
88     dht.begin();
89     // Setup a function to be called every second
90     timer.setInterval(2000L, sendSensor);
91
92 }
93
94
95 void loop()
96 {
97     Blynk.run(); // Initiates Blynk
98     timer.run(); // Initiates SimpleTimer
99 }
```

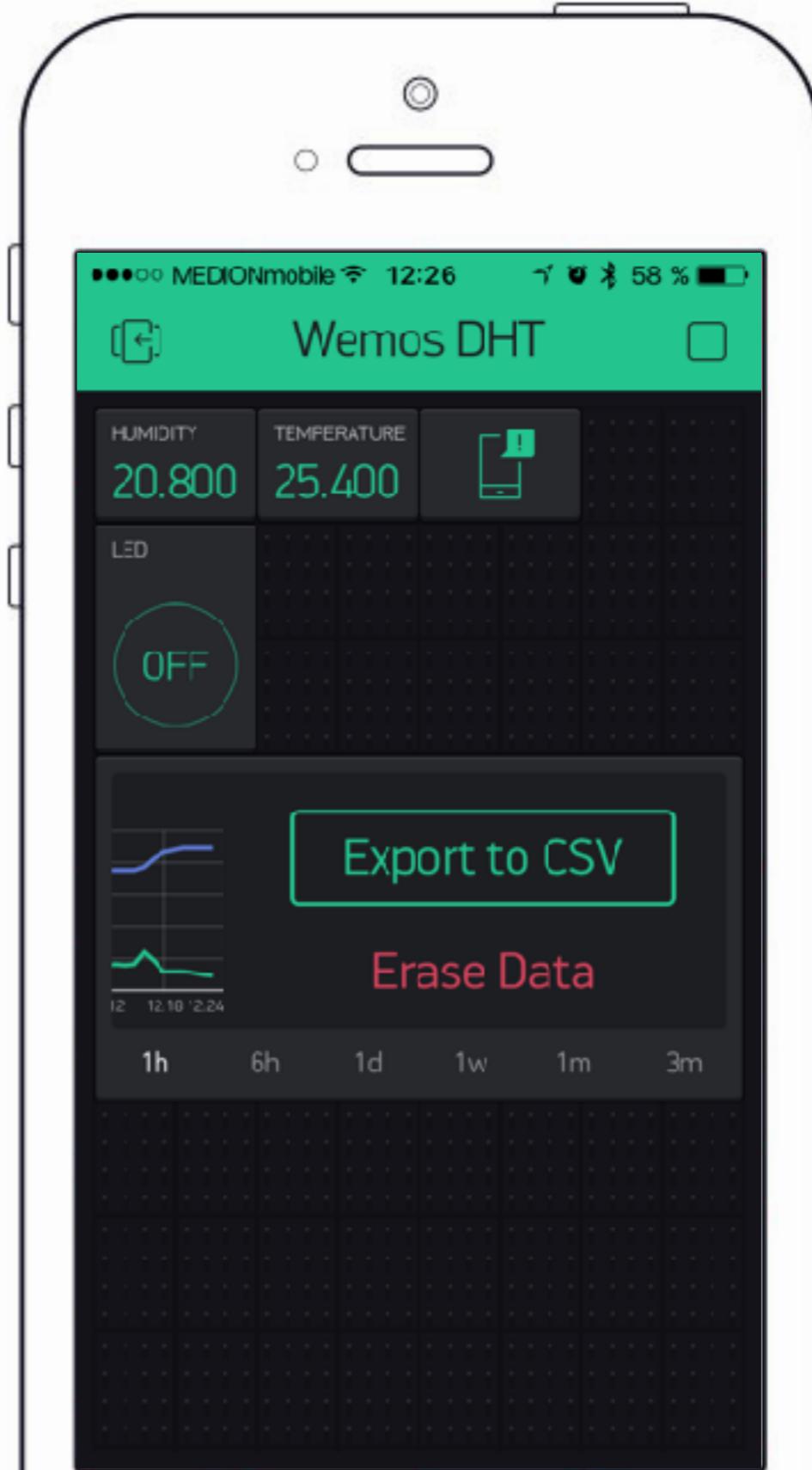
So, where's my data?

1. Yeah, cool, but how can I go back check all my measurements? Go back to the Blynk app.
2. Tap the mat, scroll down and drag a History Graph (pricey, I know)
3. Tap on it and select V5 and V6 on the first two boxes, like in the screenshot.
4. OK and then  and get ready to be amazed!



Data, Graphs & Beyond

1. Whoa! Nice graph. Thanks! You can do more though
2. Swipe left on the graph.
3. You can now get all your data sent in CSV format to your email!
4. The .csv file can be opened in a lot of different ways, including Excel. Try it out.



Thank you!

WATT*x*