

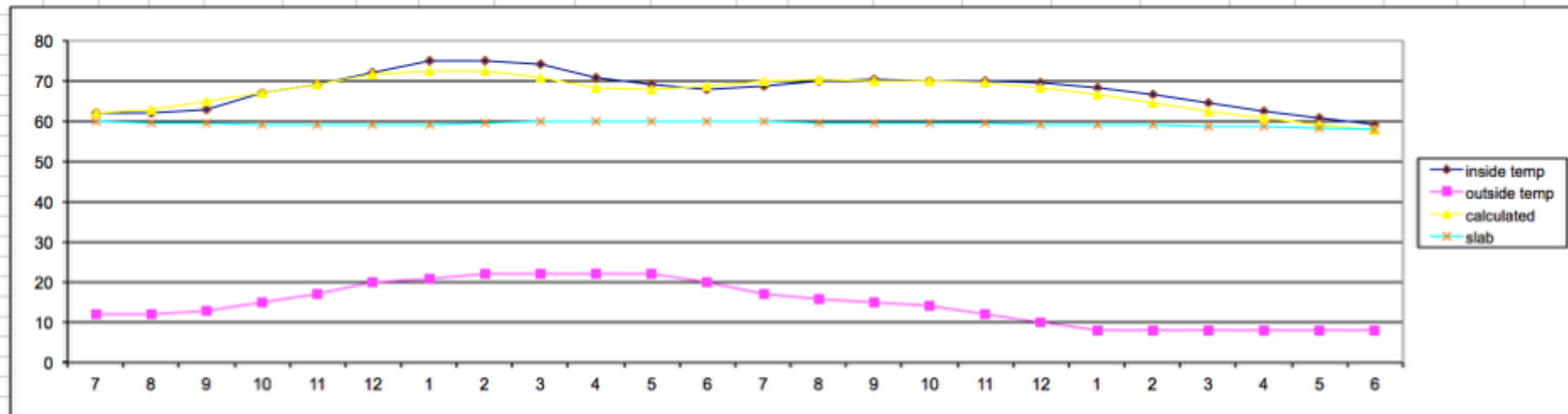
Building Wireless Sensors

Don Coleman
Chariot Solutions
Philly ETE
April 12, 2016

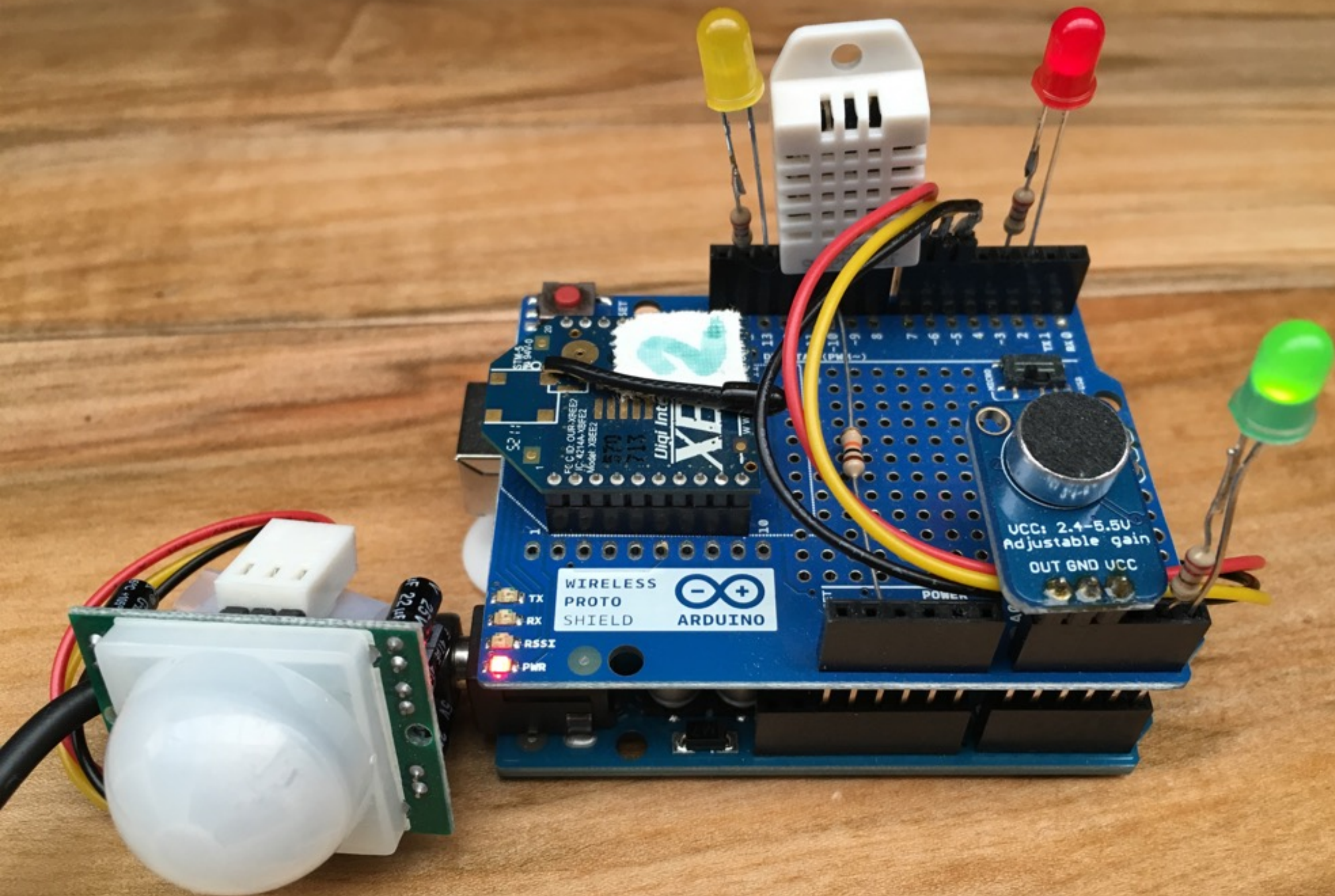




	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
2			thermo loss from house				221	btus /hr-F		thermo trans to slab					482	btu/hr-F		slab to outside		16	btu/hr-F			
3			specific heat of sheetrock and air				8239	btus/F		thermo mass of slab					25,328	btu/F		slab to grd						
4			house/outside																					
5	Time	temp	temp	delta	heat loss	make up	air/air	energy f/	energy f/	clearness	heat gain	# of	BTUs to	net heat	temp of	delta t	btu slab to	fan	heat transfe	btu slab	change	change		
6		inside	oustsi	temp	btus	air cfh	heat ex	house	slab		btus	wood	house	gain btus	slab	house-slab	house	cfm	for fan	outside	slab	house		
7	7	62	12	50	-11,047	7,500	0	-270	-6,480	0%	0	0	0	-12,281	60	2	-965	0	0	768	-0.25	-1.49	62	
8	8	62	12	50	-11,047	7,500		-303	-6,447	100%	19,905		0	7,470	59.75	2.25	-1085		0	768	-0.24	0.91	62.91	
9	9	62.9	13	49.9	-11,026	7,486	0	-458	-6,267	110%	28,726		0	15,604	59.51	3.40	-1639		0	748	-0.21	1.89	64.80	
10	10	67	15	52	-11,488	7,800	0	-1,081	-6,219	110%	33,600		0	17,314	59.30	7.70	-3716	0	0	712	-0.13	2.10	66.90	
11	11	69	17	52	-11,488	7,800	0	-1,380	-5,921	110%	36,183		0	18,572	59.17	9.83	-4742		0	677	-0.07	2.25	69.16	
12	12	72	20	52	-11,488	7,800	0	-1,812	-5,489	120%	39,601		0	20,076	59.10	12.90	-6225		0	627	0.00	2.44	71.59	
13	1	75	21	54	-11,930	8,100	0	-2,318	-5,555	120%	38,251		0	7,748	59.10	15.90	-16255	500	8585	610	0.40	0.94	72.53	
14	2	75	22	53	-11,709	7,950	0	-2,218	-5,366	100%	28,579		0	-1,196	59.50	15.50	-15848	500	8370	594	0.39	-0.15	72.39	
15	3	74	22	52	-11,488	7,800	0	-1,981	-5,320	25%	5,834	0	0	-14,443	59.89	14.11	-6807		0	600	0.04	-1.75	70.64	
16	4	71	22	49	-10,826	7,350	0	-1,465	-5,018	0%	0		0	-17,634	59.93	11.07	-5343		0	606	-0.01	-2.14	68.50	
17	5	69	22	47	-10,384	7,050	0	-1,153	-4,811	0%	0	8	11093	-4,827	59.91	9.09	-4383		0	607	-0.04	-0.59	67.91	
18	6	67.9	20	48	-10,585	7,186	0	-1,039	-5,158			8	22187	6,686	59.87	8.04	-3877	0	0	639	-0.08	0.81	68.72	
19	7	68.7	17	52	-11,427	7,758	0	-1,246	-5,977			4	27733	10,756	59.80	8.92	-4305	0	0	686	-0.09	1.31	70.03	
20	8	70	16	54	-11,936	8,104		-1,506	-6,375			4	22187	3,765	59.70	10.32	-4980		0	701	-0.08	0.46	70.48	
21	9	70.5	15	55	-12,258	8,323		-1,627	-6,685			4	16640	-2,485	59.62	10.86	-5240		0	715	-0.09	-0.30	70.18	
22	10	70.2	14	56	-12,412	8,427		-1,615	-6,907			4	16640	-2,523	59.54	10.65	-5136	0	0	730	-0.10	-0.31	69.88	
23	11	69.9	12	58	-12,787	8,681	0	-1,631	-7,413			4	16640	-2,813	59.44	10.44	-5036	0	0	761	-0.12	-0.34	69.53	
24	12	69.5	10	60	-13,153	8,930	0	-1,643	-7,927				11093	-8,633	59.31	10.22	-4931	0	0	791	-0.15	-1.05	68.49	
25	1	68.5	8	60	-13,363	9,073	0	-1,522	-8,356				5547	-13,836	59.16	9.32	-4497	0	0	821	-0.18	-1.68	66.81	
26	2	66.8	8	59	-12,992	8,821	0	-1,243	-8,094				0	-18,011	58.98	7.83	-3776	0	0	819	-0.20	-2.19	64.62	
27	3	64.6	8	57	-12,509	8,493	0	-893	-7,763				0	-16,222	58.78	5.84	-2819	0	0	816	-0.23	-1.97	62.65	
28	4	62.7	8	55	-12,074	8,198	0	-605	-7,459				0	-14,659	58.55	4.10	-1979	0	0	812	-0.25	-1.78	60.87	
29	5	60.9	8	53	-11,681	7,931	0	-367	-7,181				0	-13,289	58.30	2.57	-1241	0	0	809	-0.27	-1.61	59.26	
30	6	59.3	8	51	-11,325	7,689	0	-170	-6,925				0	-12,085	58.03	1.22	-591	0	0	805	-0.28	-1.47	57.79	
31					-282,424	0.43	0.00	-29,546	-155,112		230,679	36	149,760	-46,946			-115,414			17,219				





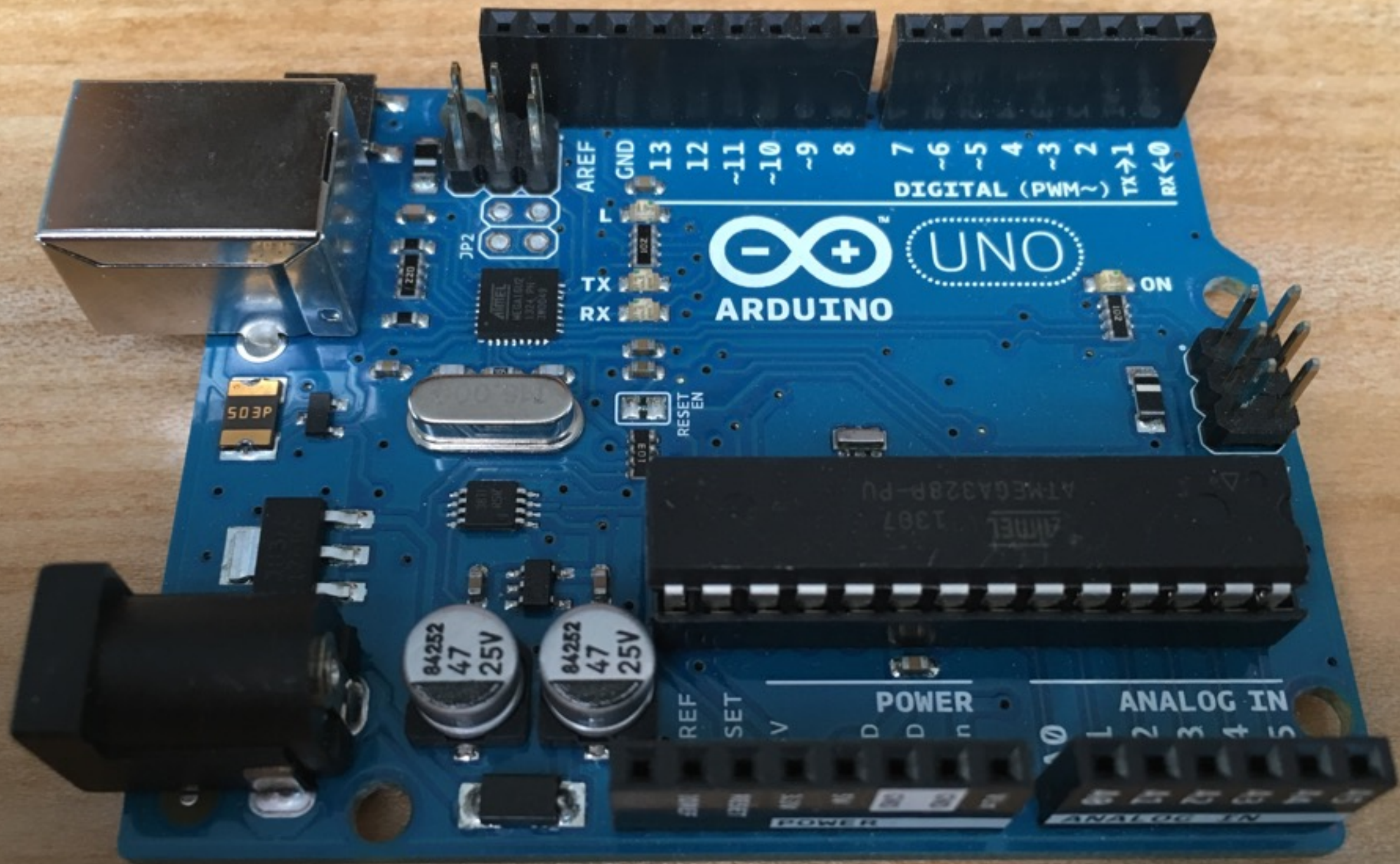


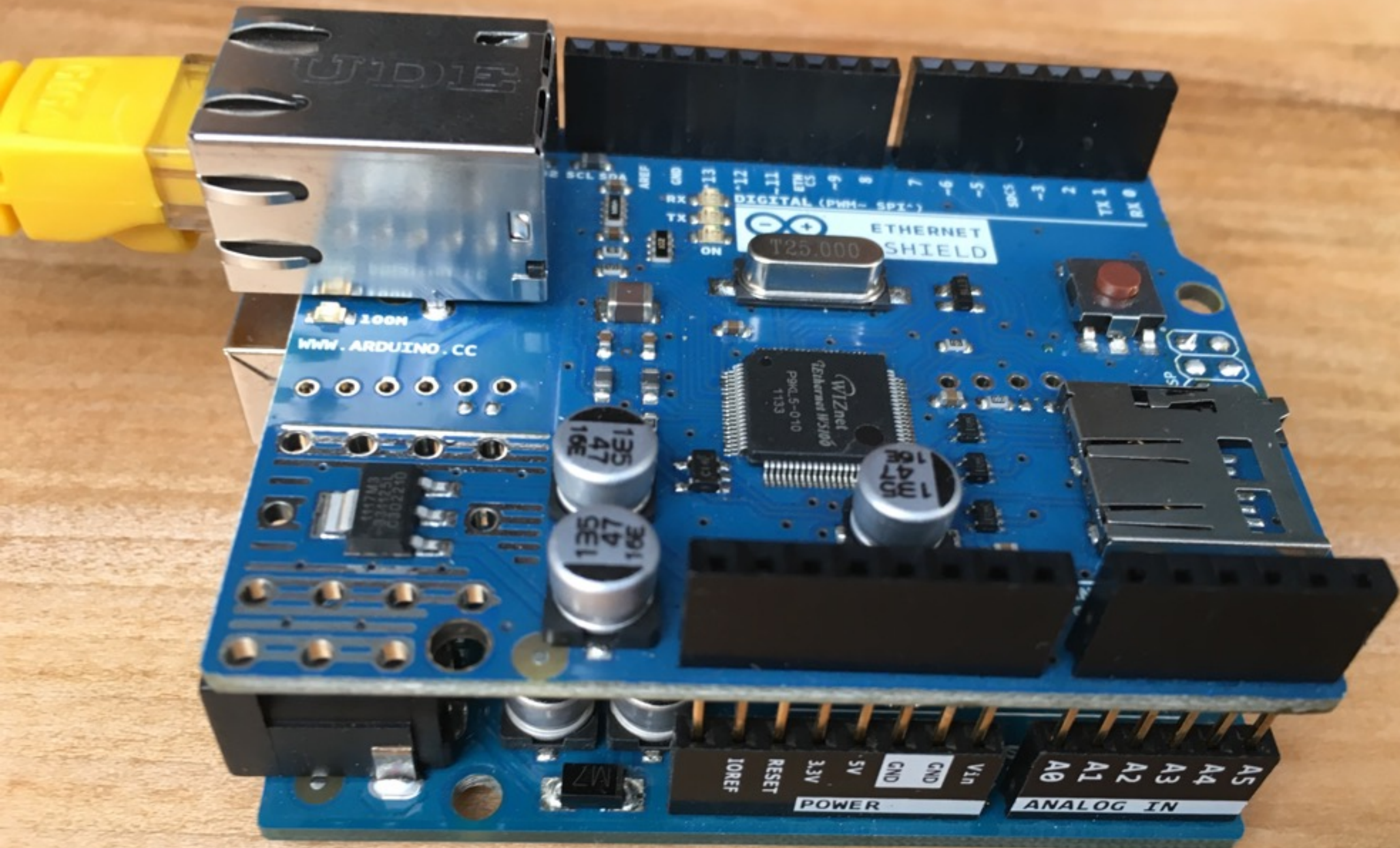
Make: Bluetooth

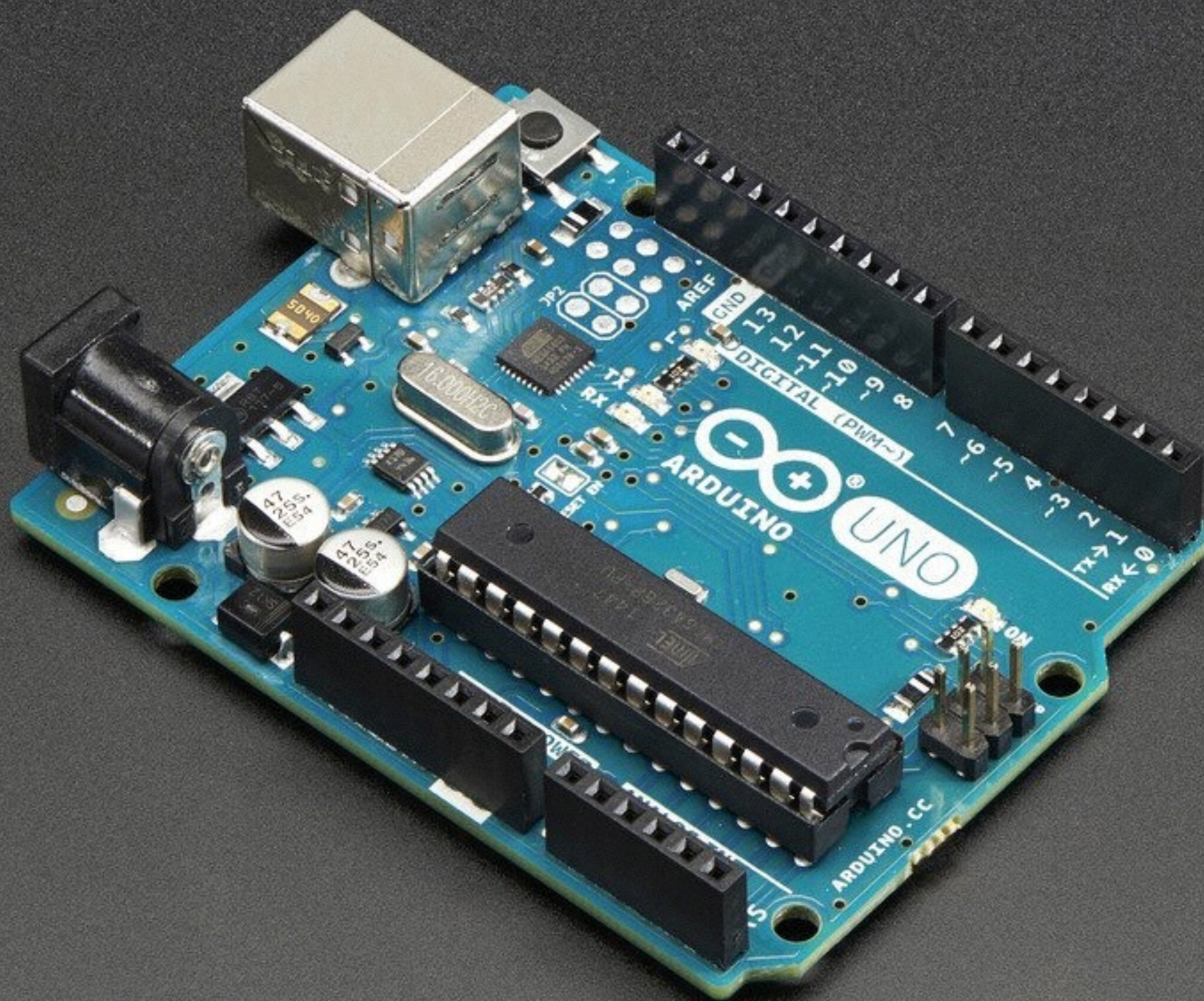


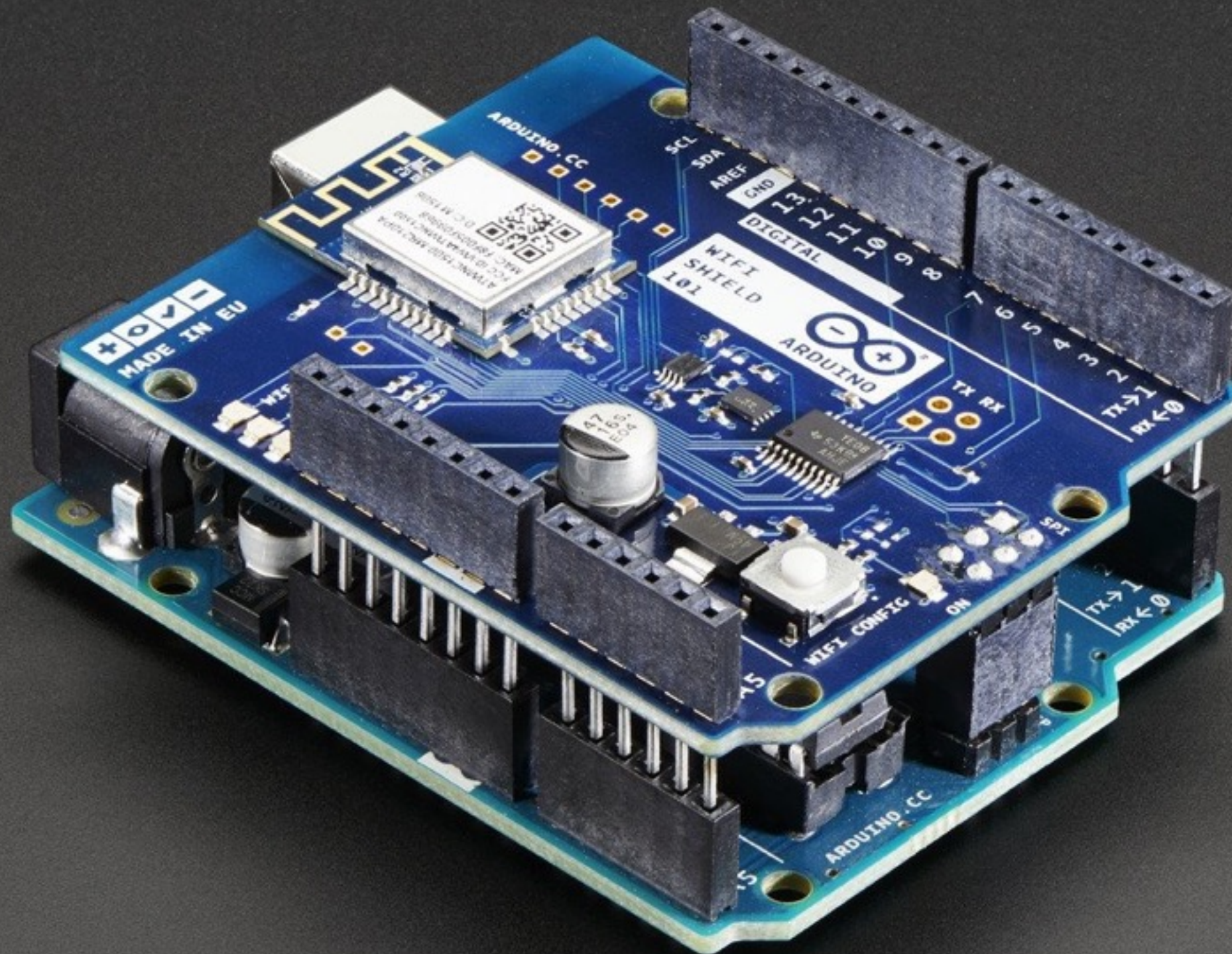
Bluetooth LE Projects for Arduino,
Raspberry Pi, and Smartphones

Alasdair Allan, Don Coleman,
Sandeep Mistry











ARDUINO.CC

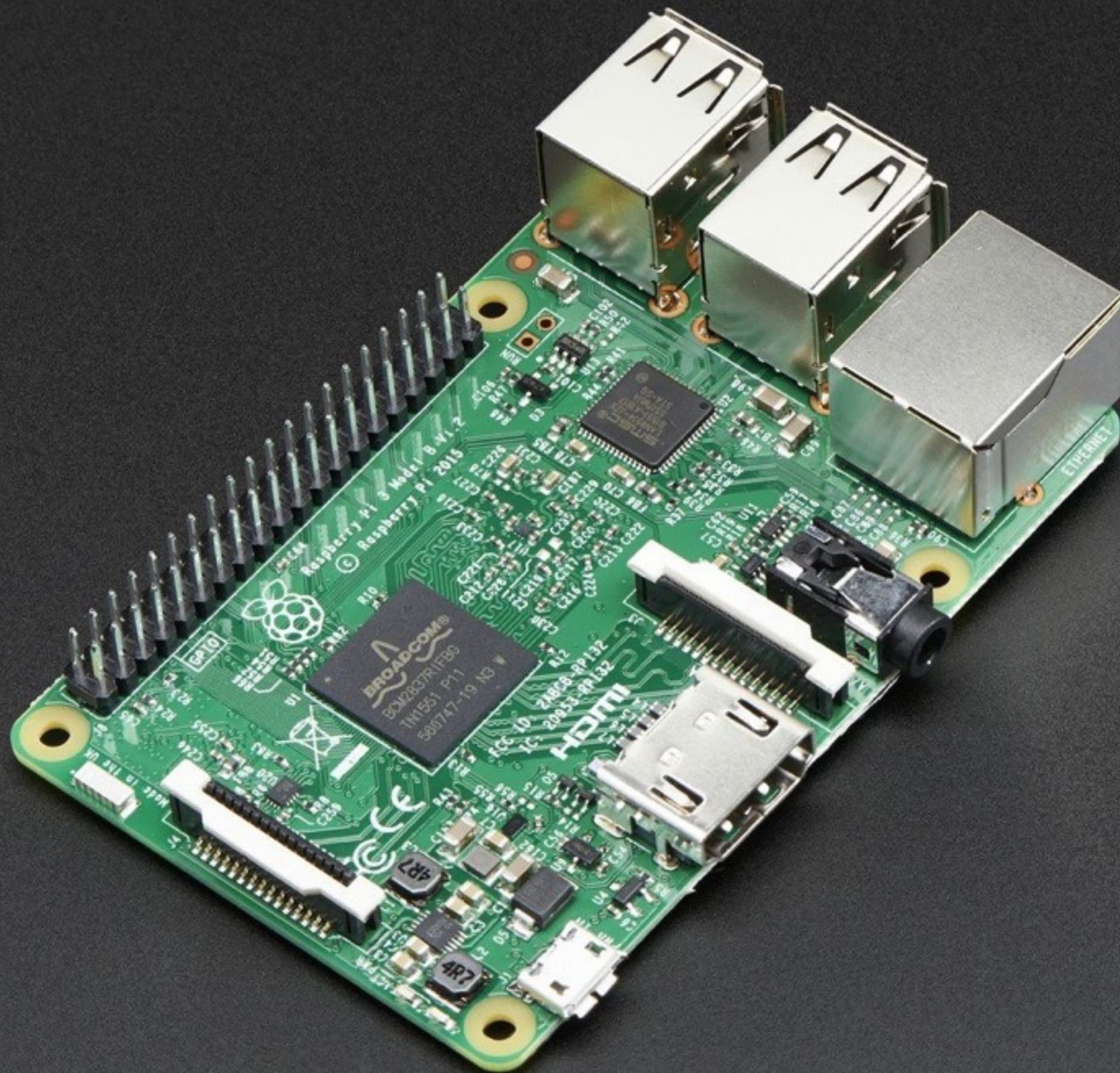
ARDUINO
YUN

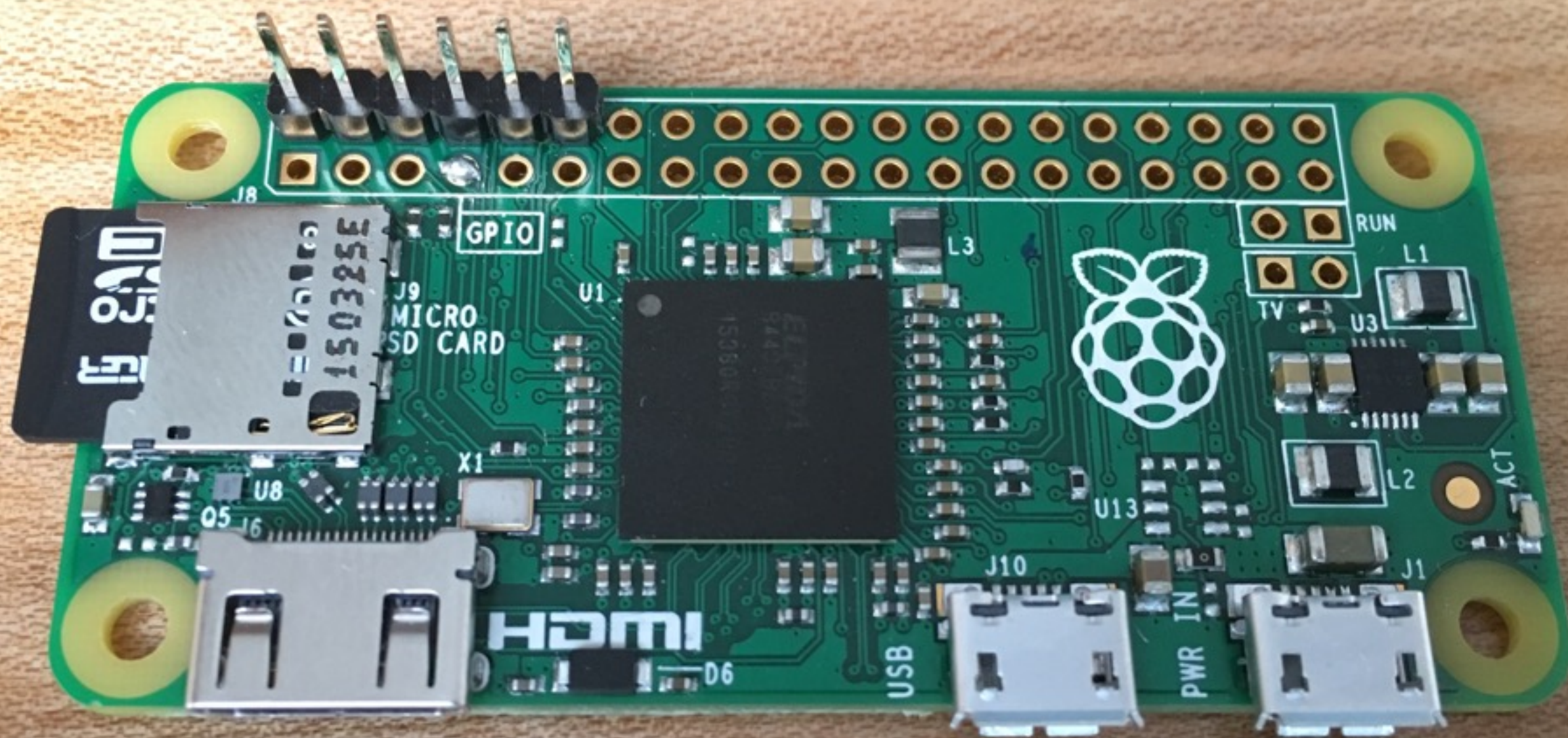
ATMEL
MEGA32U4
MH
1330E
1J6285-2

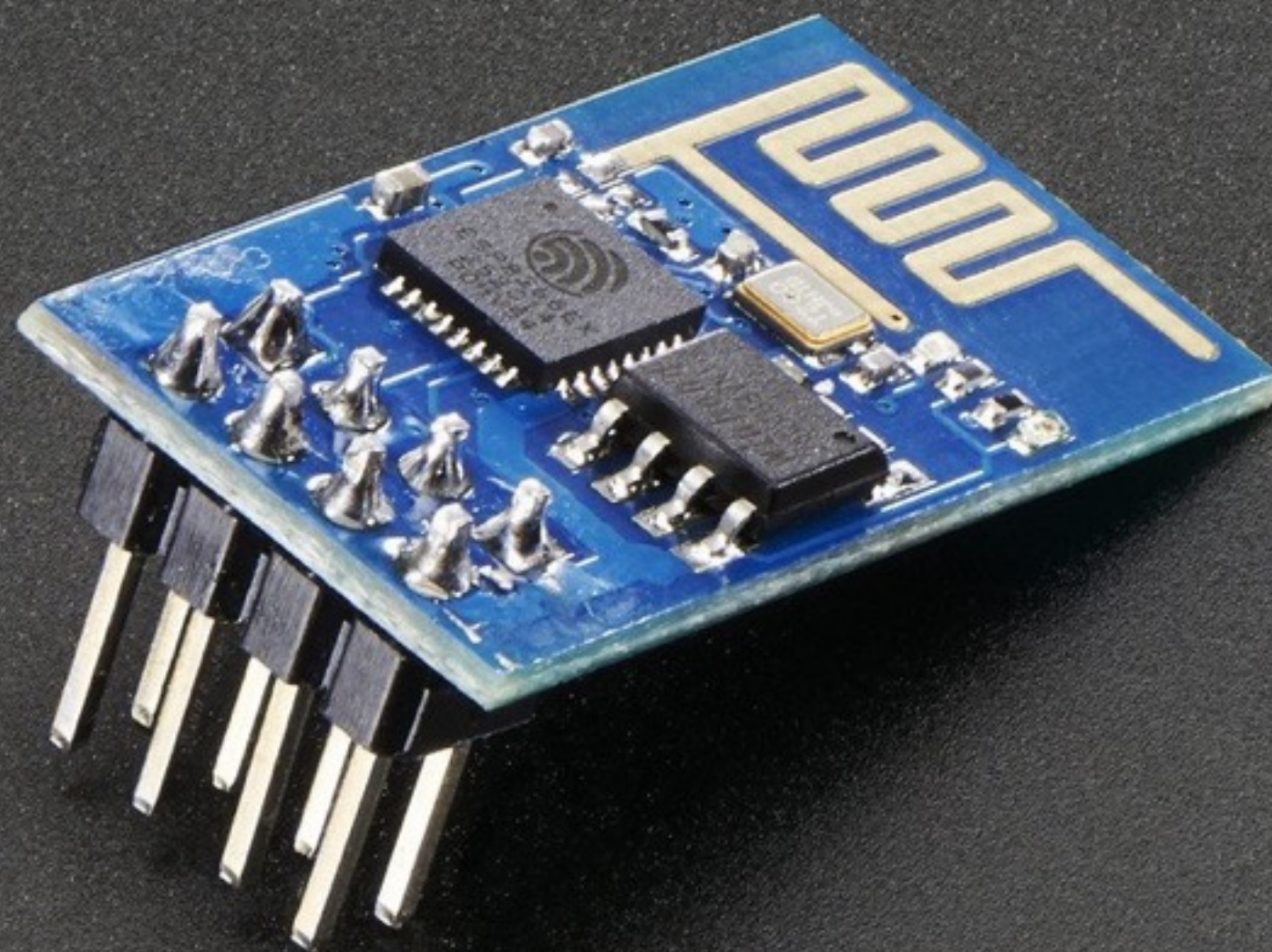
RX
TX
L13
WAN
ON
WLAN
USB
YUN RST

POWER

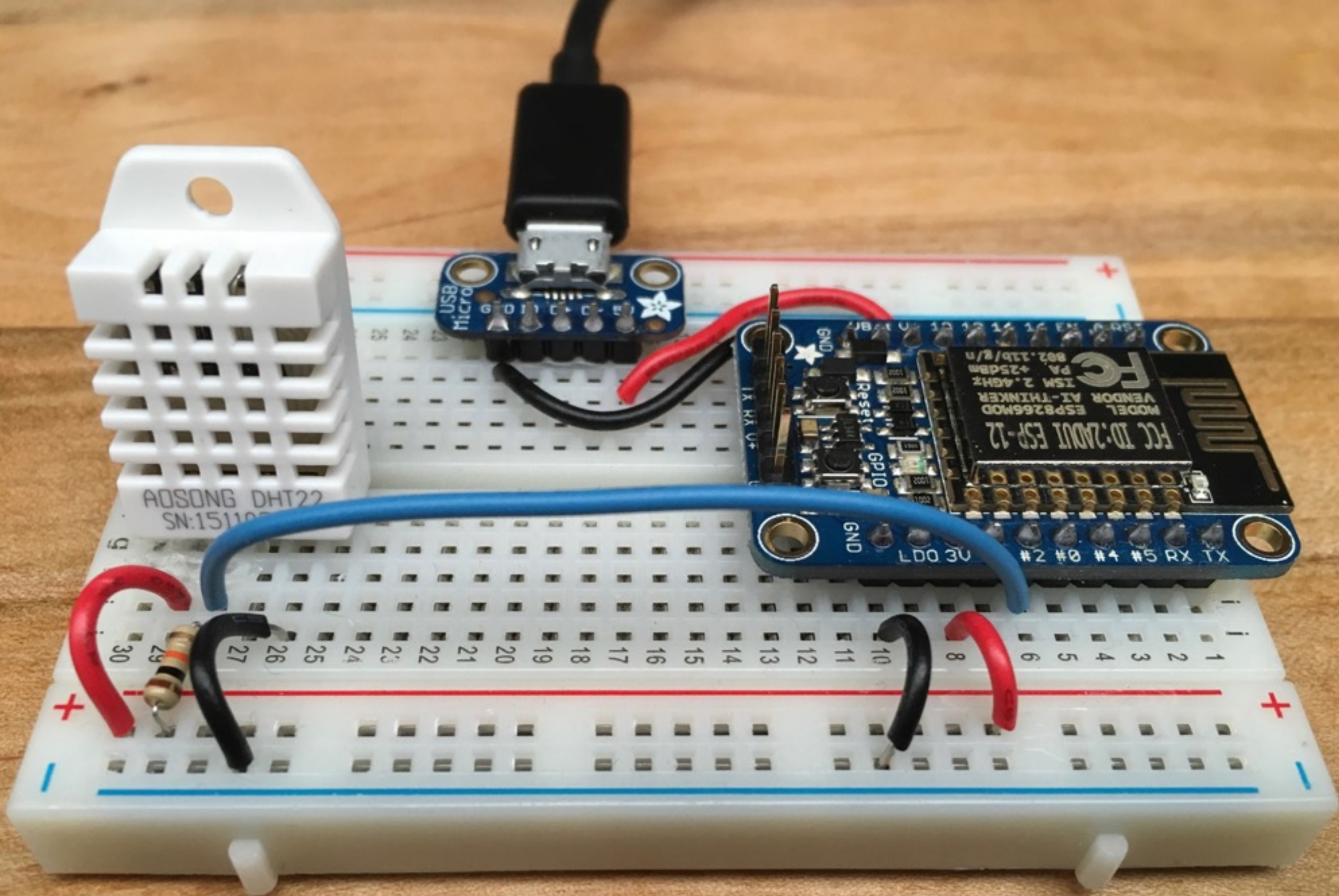
ANALOG IN

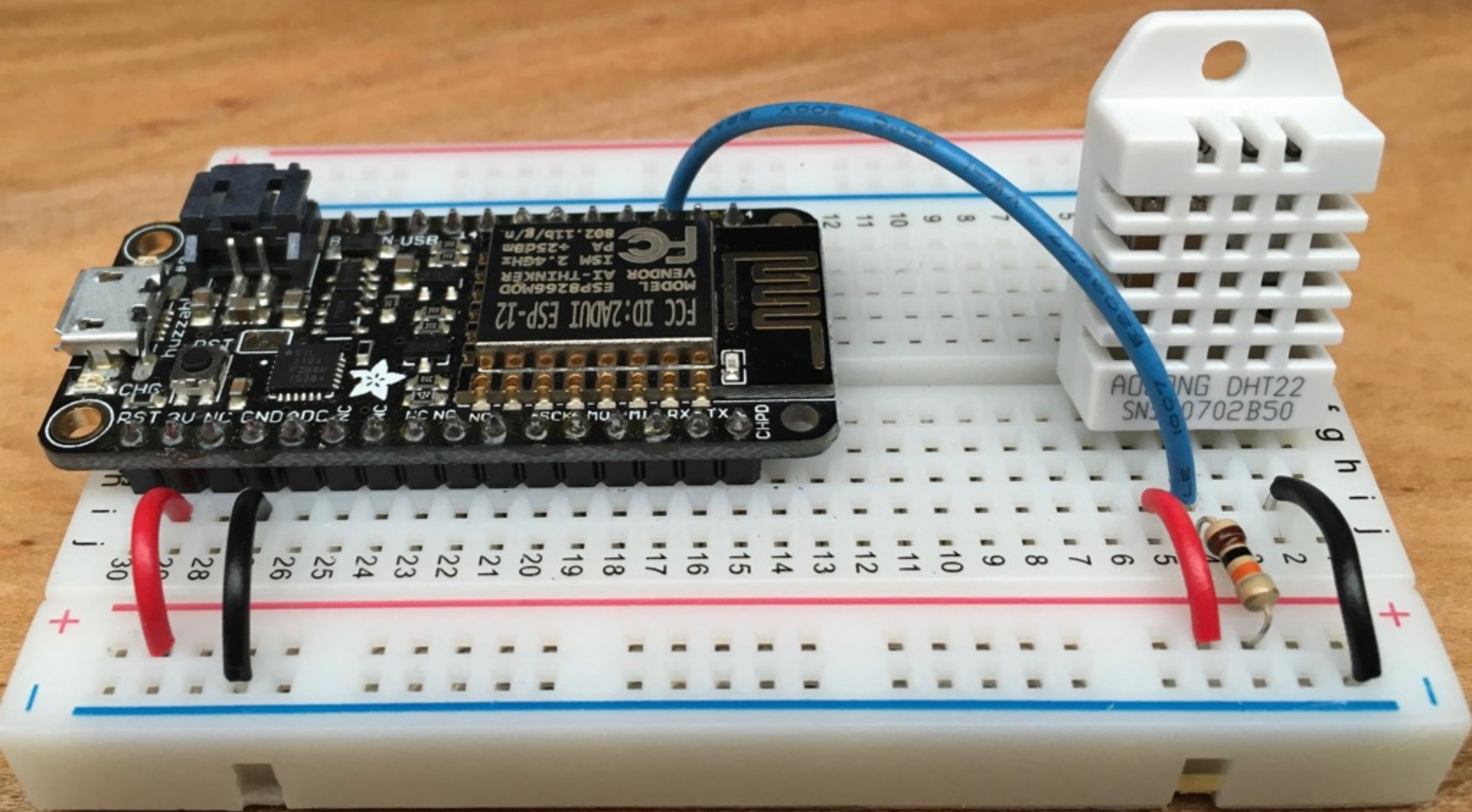














Search the Arduino Website

Home Buy Download Products Learning Forum Support Blog

DOWNLOAD

ENGLISH

Download the Arduino Software



ARDUINO 1.6.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Release Notes
Source Code
Checksums



DHT



```
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000);

  float temperature = dht.readTemperature(true); // Fahrenheit
  float humidity = dht.readHumidity();

  Serial.print(temperature);
  Serial.print(" *F ");
  Serial.print(humidity);
  Serial.println(" %");
}
```


DHT

```
#include "DHT.h"
```

```
#define DHTPIN 2
```

```
#define DHTTYPE DHT22
```

```
DHT dht
```

```
void setup
```

```
Serial
```

```
dht
```

```
}
```

```
void loop
```

```
delay
```

```
float
```

```
float
```

```
Serial
```

```
Serial
```

```
Serial
```

```
Serial
```

```
}
```

Serial Monitor window showing received data:

68.54 *F 25.70 %
68.54 *F 25.70 %
68.54 *F 25.70 %

Buttons: Send, Autoscroll, Newline, 9600 baud

Done uploading.

Sketch uses 231,993 bytes (53%) of program storage space. Maximum is 434,160
Global variables use 31,600 bytes (38%) of dynamic memory, leaving 50,320 bytes
Uploading 236144 bytes from to flash at 0x00000000

data.sparkfun.com

a place
to push
your data.

Why are you building this?

We want to bring a dose of reality to the Internet of Things hype. data.sparkfun.com is a free, robust service for use with all of your projects. The underlying engine is open source so if you don't want to use our servers you can install [phant](#) on the server of your choice.

Wait, this is totally free? What's the catch?

Create a free
data stream
immediately at
data.sparkfun.com

CREATE

Explore all of

EXPLORE

Create a Data Stream

If you need more info about creating a stream, check out the [stream creation documentation](#).

Title*

Philly ETE

Description*

Temperature and Humidity from DHT22

Show in Public Stream List?*

☒ Visible ☐ Hidden

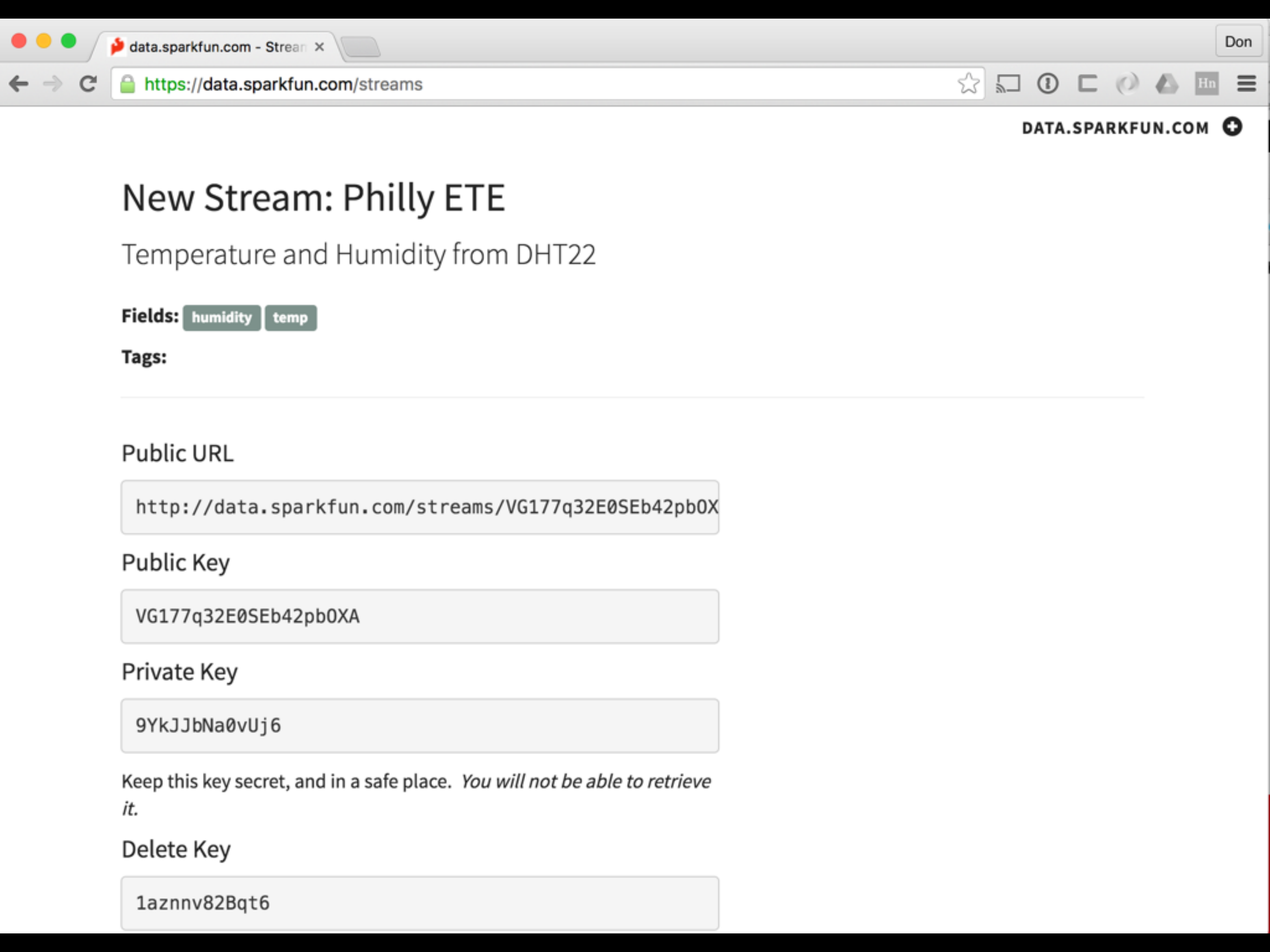
Fields*

temp x humidity x humidity, temp

Stream Alias

nist_weather

This will be used as an alias for your stream when sharing.
e.g. [http://data.sparkfun.com/](#)



New Stream: Philly ETE

Temperature and Humidity from DHT22

Fields: humidity temp

Tags:

Public URL

`http://data.sparkfun.com/streams/VG177q32E0SEb42pb0X`

Public Key

`VG177q32E0SEb42pb0XA`

Private Key

`9YkJJbNa0vUj6`

Keep this key secret, and in a safe place. *You will not be able to retrieve it.*

Delete Key

`1aznnv82Bqt6`



Sensor keys.h §

```
const char* ssid      = "dc";  
const char* password = "secret";  
  
#define HOST "data.sparkfun.com"  
int httpPort = 80;  
  
// ETE  
#define PUBLIC_KEY "VG177q32E0SEb42pb0XA"  
#define PRIVATE_KEY "9YkJJbNa0vUj6Brwe543"  
#define DELETE_KEY "1aznnv82Bqt6GNdf899d"
```



```
#include <Phant.h>
#include <ESP8266WiFi.h>

WiFiClient client;
Phant phant(HOST, PUBLIC_KEY, PRIVATE_KEY);

float temperature = dht.readTemperature(true);
float humidity = dht.readHumidity();

phant.add("humidity", humidity);
phant.add("temp", temperature);

client.connect(HOST, httpPort)
client.print(phant.post());
```


Sensor

keys.h

```
void loop() {  
  
  float temperature = dht.readTemperature(true); // true == Fahrenheit  
  float humidity = dht.readHumidity();  
  
  Serial.print(temperature);  
  Serial.print(", ");  
  Serial.println(humidity);  
  
  phant.add("humidity", humidity);  
  phant.add("temp", temperature);  
  
  WiFiClient client;  
  if (!client.connect(HOST, httpPort)) {  
    Serial.println("connection failed");  
    return;  
  }  
  
  client.print(phant.post());  
  delay(10);  
  
  // Read all the lines of the reply from server  
  while(client.available()){  
    String line = client.readStringUntil('\r');
```



A serial monitor window titled "/dev/cu.SLAB_USBtoUART" displays the output of the Arduino program. The output shows two lines of data: the first line is the IP address "192.168.1.2" preceded by two dots, and the second line is the sensor readings "69.08, 25.80". The window includes standard macOS window controls (red, yellow, green buttons) and a status bar at the bottom with "Autoscroll" checked, "Newline" selected, and a character count of "9600".

```
..  
192.168.1.2  
69.08, 25.80
```

Done uploading.

Uploading 241152 bytes from to flash at 0x00000000

Philly ETE

Temperature and Humidity Data from DHT22

Manage

Export to Analog.io

JSON

CSV

MySQL

PostgreSQL

Atom

TAGS

100% (49.98 of 50 MB) remaining.

humidity	temp	timestamp
25.6000	68.3600	2016-04-10T03:38:52.794Z
25.5000	68.3600	2016-04-10T03:38:47.040Z
25.5000	68.3600	2016-04-10T03:38:41.352Z
25.4000	68.0000	2016-04-10T03:38:36.184Z
25.6000	68.3600	2016-04-10T03:38:31.482Z
25.5000	68.3600	2016-04-10T03:38:24.891Z
25.5000	68.1800	2016-04-10T03:38:19.221Z
25.5000	68.3600	2016-04-10T03:38:13.691Z
25.5000	68.3600	2016-04-10T03:38:07.981Z
25.5000	68.3600	2016-04-10T03:38:01.347Z
25.4000	68.0000	2016-04-10T03:37:55.622Z
25.5000	68.1800	2016-04-10T03:37:50.324Z

Office ESP8266 with DHT22 and solar measurement from photovoltaic cell

Manage Export to Analog.io

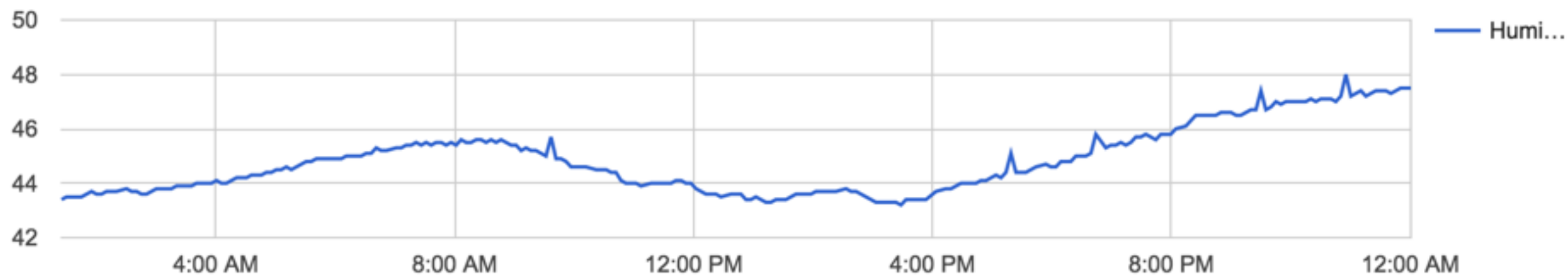
JSON CSV MySQL PostgreSQL Atom

TAGS

97% (48.30 of 50 MB) remaining.

humidity	solar	temp	timestamp
47.5000	8	51.9800	2016-03-19T03:55:35.801Z
47.5000	7	51.9800	2016-03-19T03:50:34.680Z
47.3000	8	52.1600	2016-03-19T03:40:30.649Z
47.4000	7	51.9800	2016-03-19T03:35:29.710Z
47.4000	8	52.1600	2016-03-19T03:30:28.917Z
47.4000	8	51.9800	2016-03-19T03:25:28.365Z
47.3000	8	52.1600	2016-03-19T03:20:26.146Z
47.2000	8	52.1600	2016-03-19T03:15:25.191Z
47.4000	8	52.1600	2016-03-19T03:10:24.149Z
47.2000	8	52.1600	2016-03-19T03:00:19.827Z
48.0000	9	52.3400	2016-03-19T02:55:19.250Z

Humidity



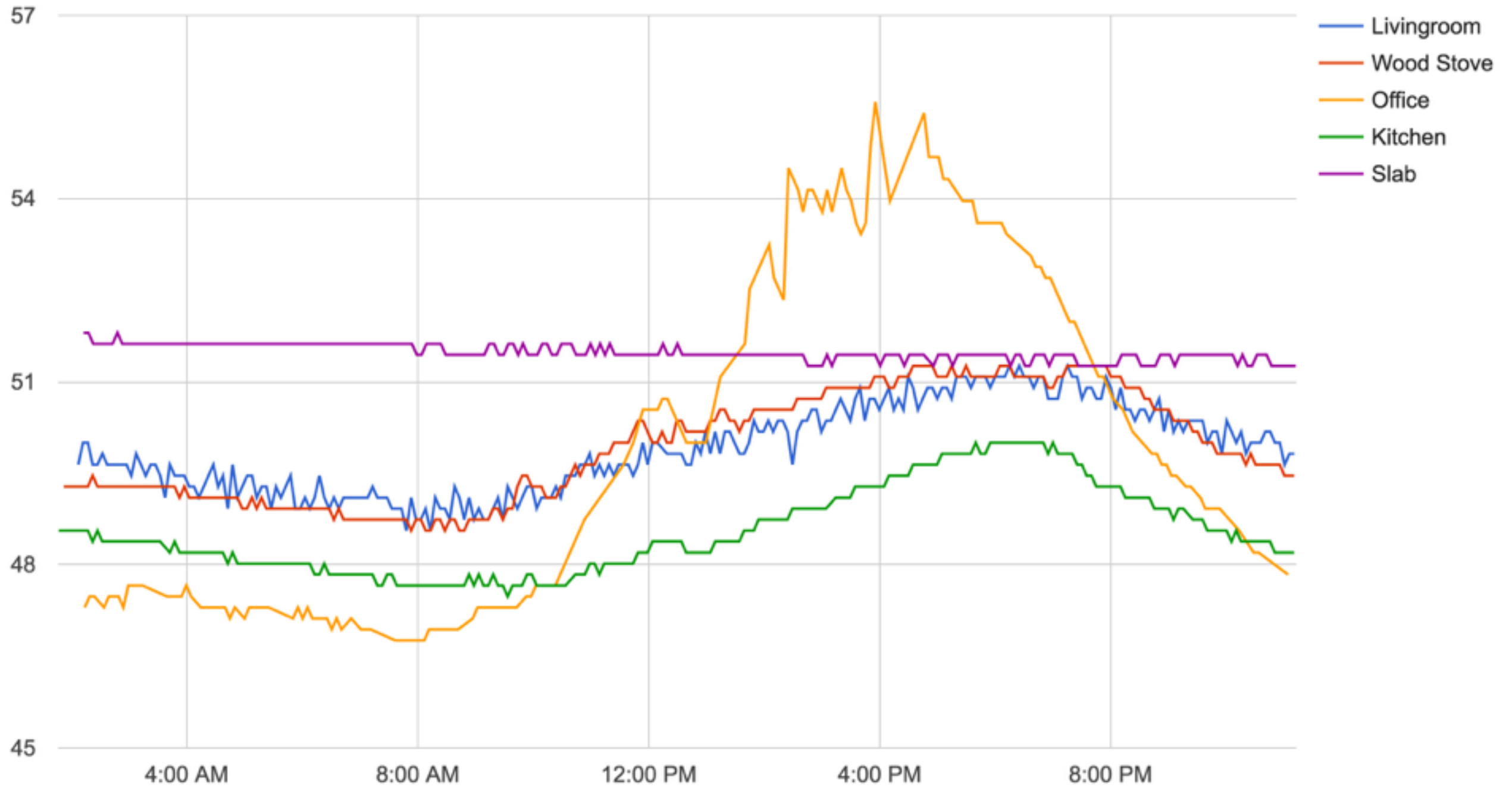
Temperature

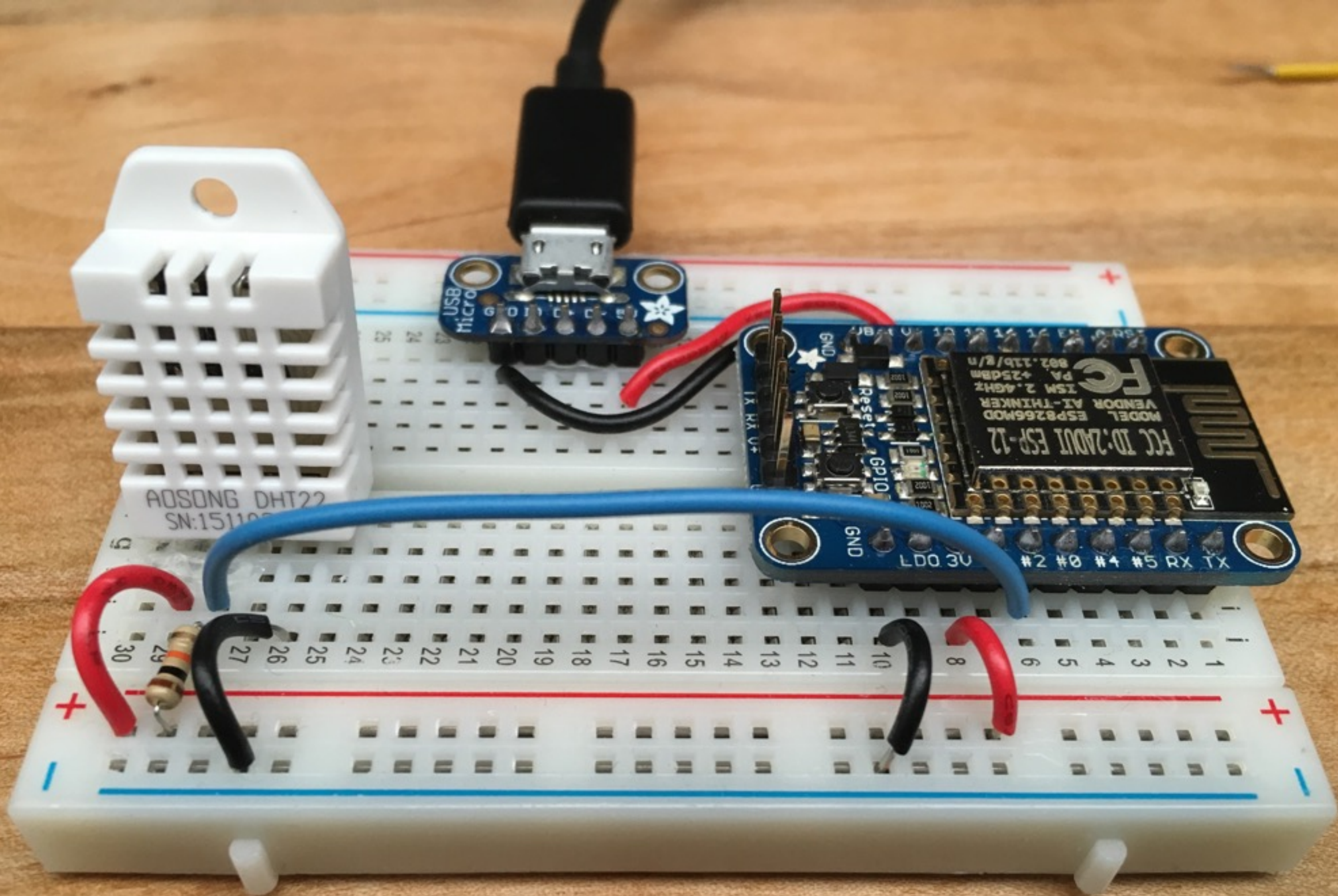


Solar



Temperature





MQTT


```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);
client.setServer(mqtt_server, 1883);

void loop() {
    client.loop();
    float temp = dht.readTemperature(true);
    client.publish("devices/ete/temperature", temp);
}
```


DHT-MQTT

ssid.h

```

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  float temperature = dht.readTemperature(true);
  float humidity = dht.readHumidity();

  Serial.print(temperature); Serial.print(", ");
  Serial.println(humidity);

  publish(TEMPERATURE_TOPIC, temperature);
  publish(HUMIDITY_TOPIC, humidity);

  delay(interval);
}

void publish(const char* topic, float value) {
  String string = String(value, 1);
  char buffer[5];
  string.toCharArray(buffer, sizeof(buffer));
  client.publish(topic, buffer);
}

```

/dev/cu.SLAB_USBtoUART

```

DHT22 Sensor

WiFi connected
192.168.1.2
Attempting MQTT connection...connected
68.36, 25.10
68.36, 25.40
68.36, 25.40
68.36, 25.30
68.36, 25.30

```

☒ Autoscroll
 Newline

Done uploading.

Sketch uses 258,228 bytes (54%) of program storage space. Maximum is 454,100 bytes.
 Global variables use 32,441 bytes (39%) of dynamic memory, leaving 49,479 bytes for local variables. Maximum is 52,000 bytes.
 Uploading 242384 bytes from to flash at 0x00000000

don — mosquitto_sub -t devices/ete/temperature — 69x22

```
[xvi:~ don$ mosquitto_sub -t devices/ete/temperature
```

68.2

68.2

68.2

68.4

74.1

72.1

70.3

69.3

69.1

69.1



don — mosquitto_sub -t devices/ete/humidity — 69×22

```
[xvi:~ don$ mosquitto_sub -t devices/ete/humidity
```

31.1

30.7

30.4

30.1

30.0

29.7

29.5

29.3

don — mosquitto_sub -t # -d — 108x24

```
[xvi:~ don$ mosquitto_sub -t '#' -d
Client mosqsub/43838-xvi.local sending CONNECT
Client mosqsub/43838-xvi.local received CONNACK
Client mosqsub/43838-xvi.local sending SUBSCRIBE (Mid: 1, Topic: #, QoS: 0)
Client mosqsub/43838-xvi.local received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/temperature', ... (4 bytes))
68.2
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/humidity', ... (4 bytes))
25.4
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/temperature', ... (4 bytes))
68.2
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/humidity', ... (4 bytes))
26.0
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/temperature', ... (4 bytes))
68.2
Client mosqsub/43838-xvi.local received PUBLISH (d0, q0, r0, m0, 'devices/ete/humidity', ... (4 bytes))
25.4
```


Sensor Graphs

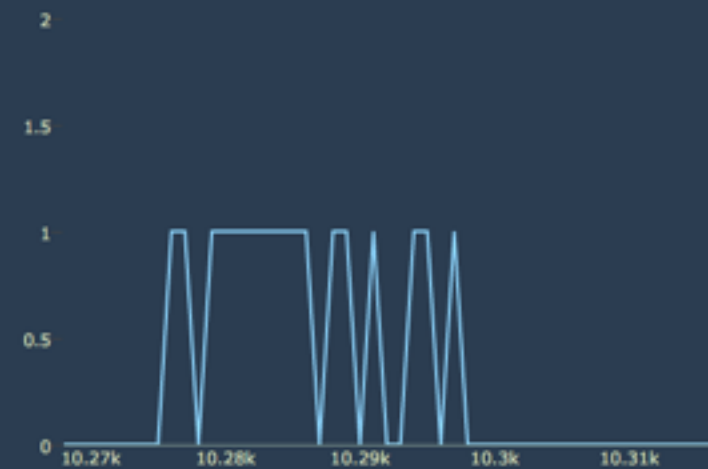
Force



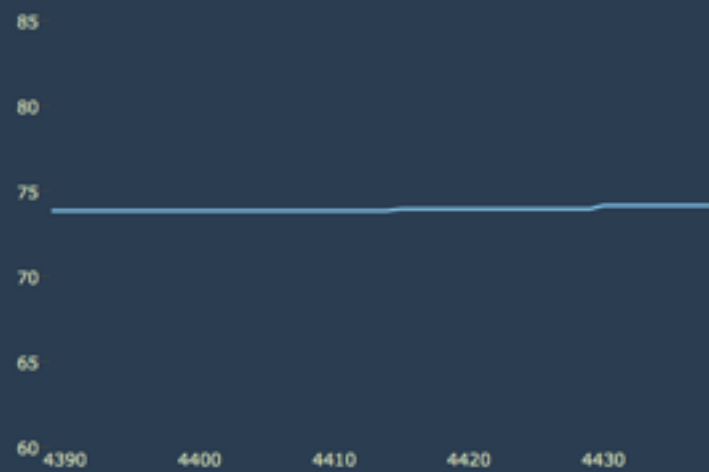
Light



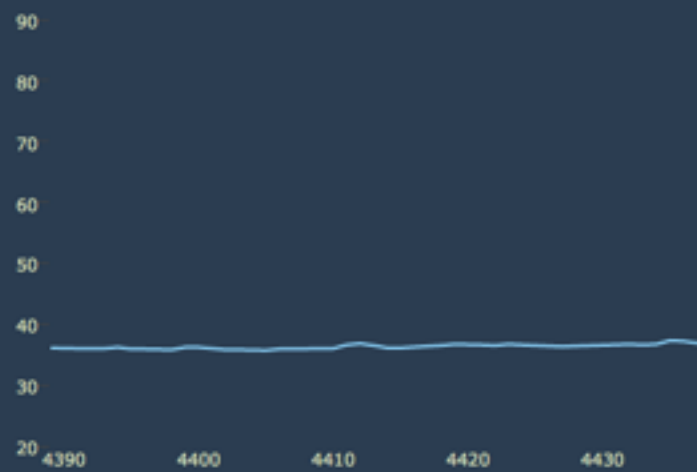
Button



Temperature

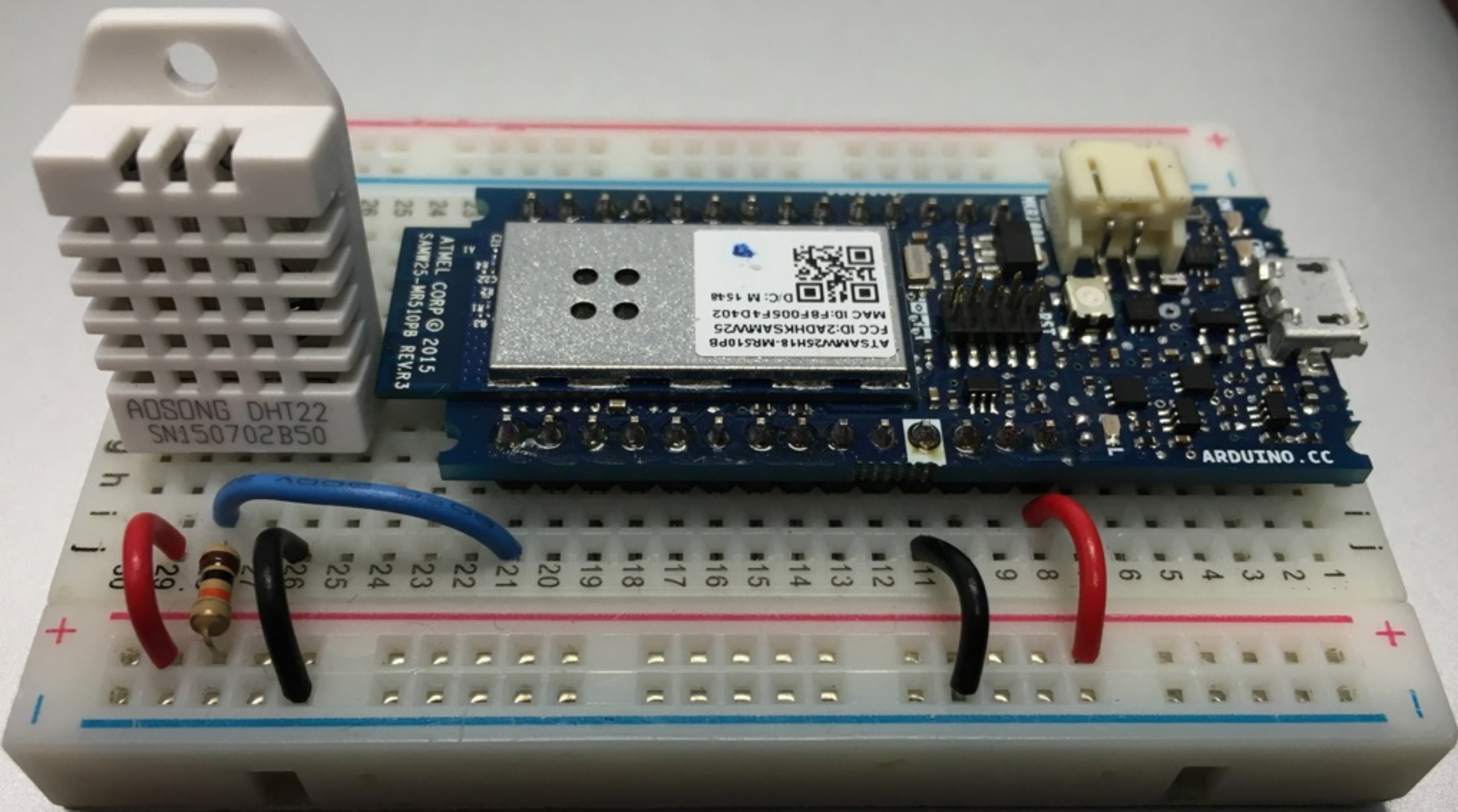


Humidity



Distance







hackster.io

Projects

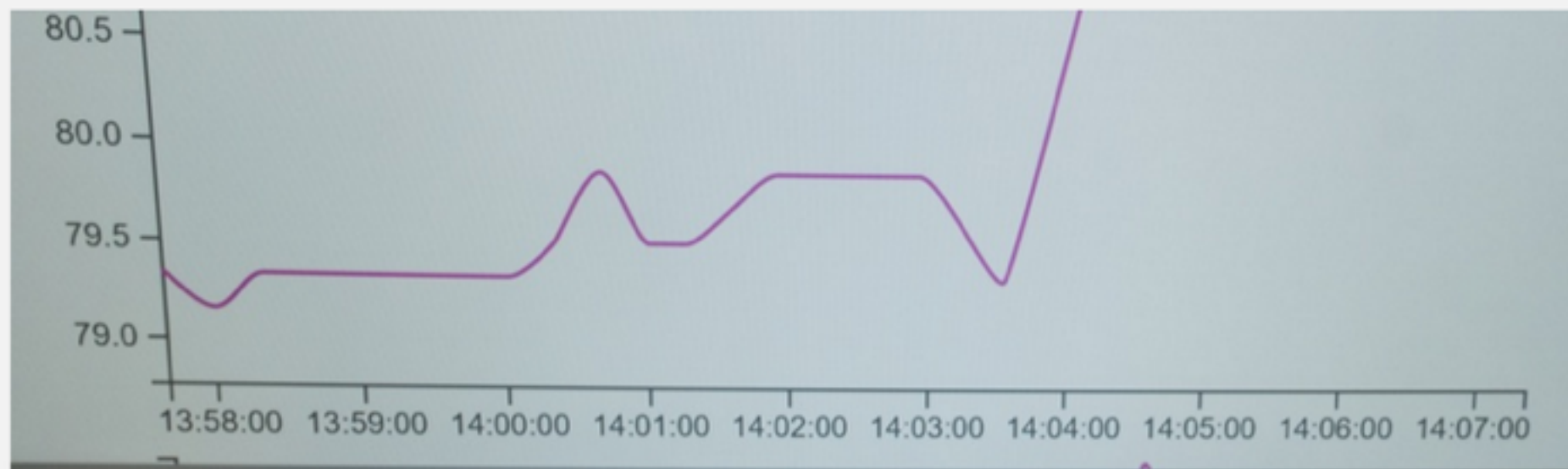
Platforms

Challenges

Live

Search

PROJECT TUTORIAL // INTERMEDIATE // 2 HOURS TO COMPLETE



Watch 1

dy

Search the web and Windows

MKR1000 Temp and Humidity Sensor

© Apache-2.0

Send temperature and humidity data from an Arduino MKR1000 to Microsoft Azure.

data collection

environmental sensing

weather

92 VIEWS 0 COMMENTS 2 RESPECTS

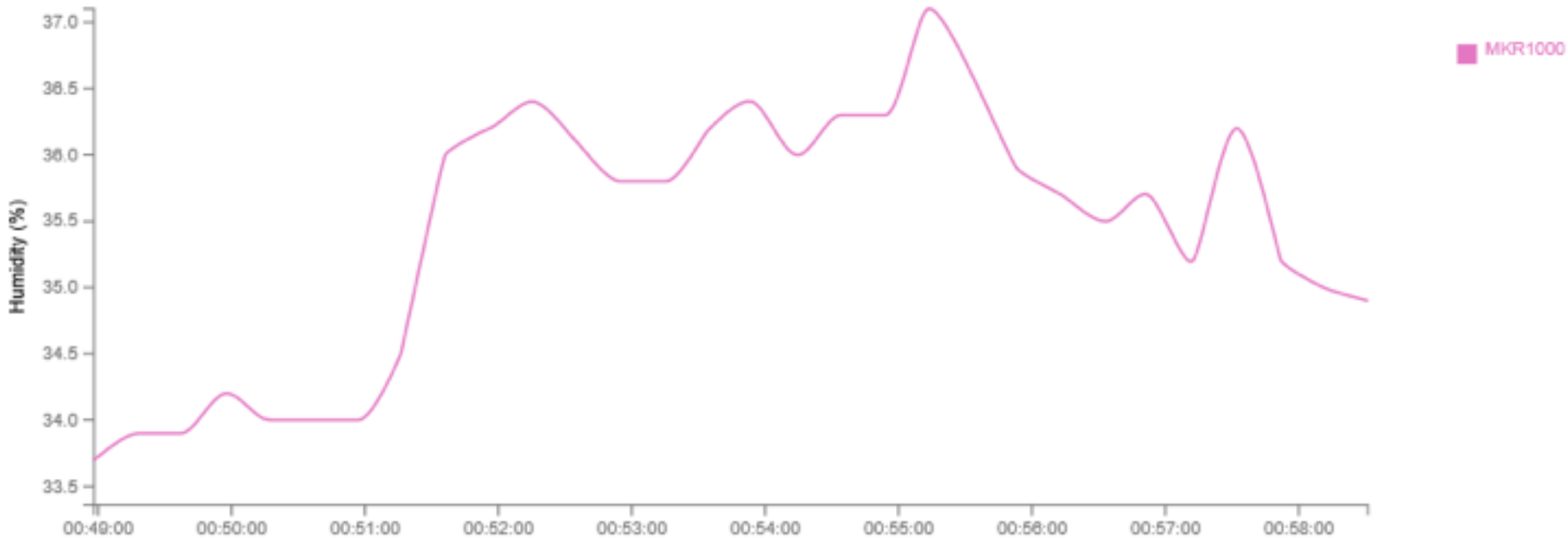


Fork me on GitHub

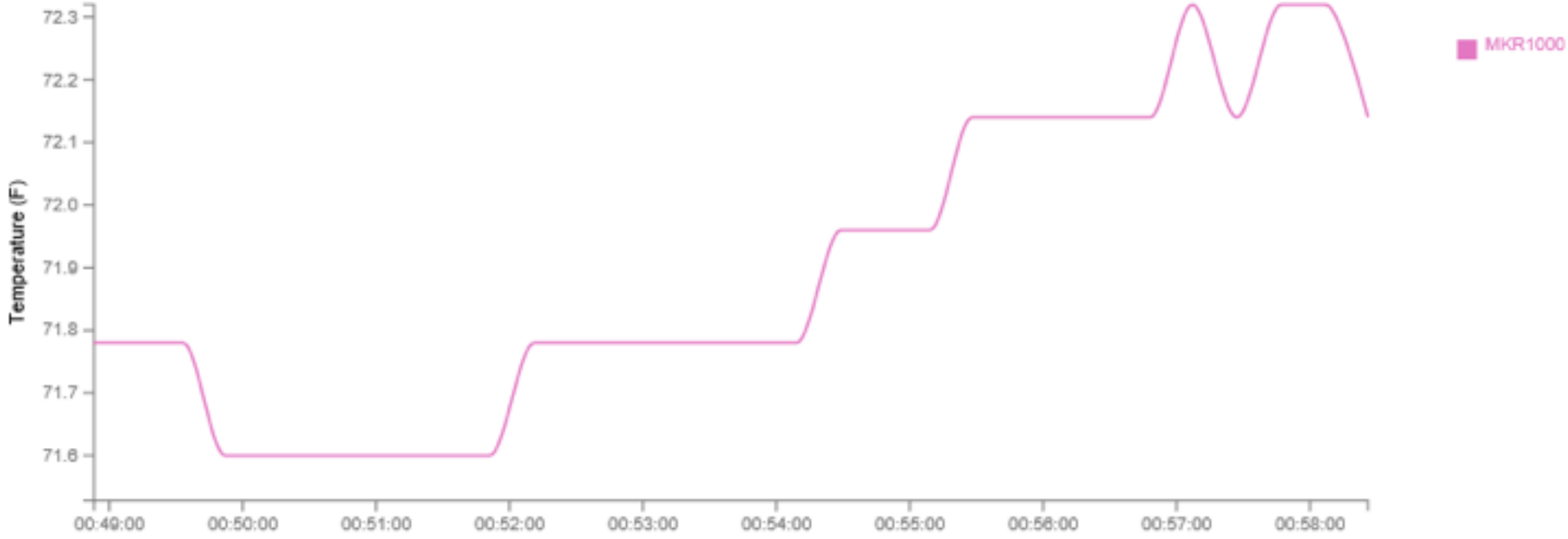
Live Sensor Data

Select Sensor/R-PI:

▶ MKR1000



▶ MKR1000





STORE

BUILD

DASHBOARD



Build your Internet of Things.

Particle is a prototype-to-production platform for developing an Internet of Things product.

[GET STARTED WITH A DEV KIT](#)



PROTOTYPE

Build a prototype in minutes with cloud-connected hardware development kits



SCALE

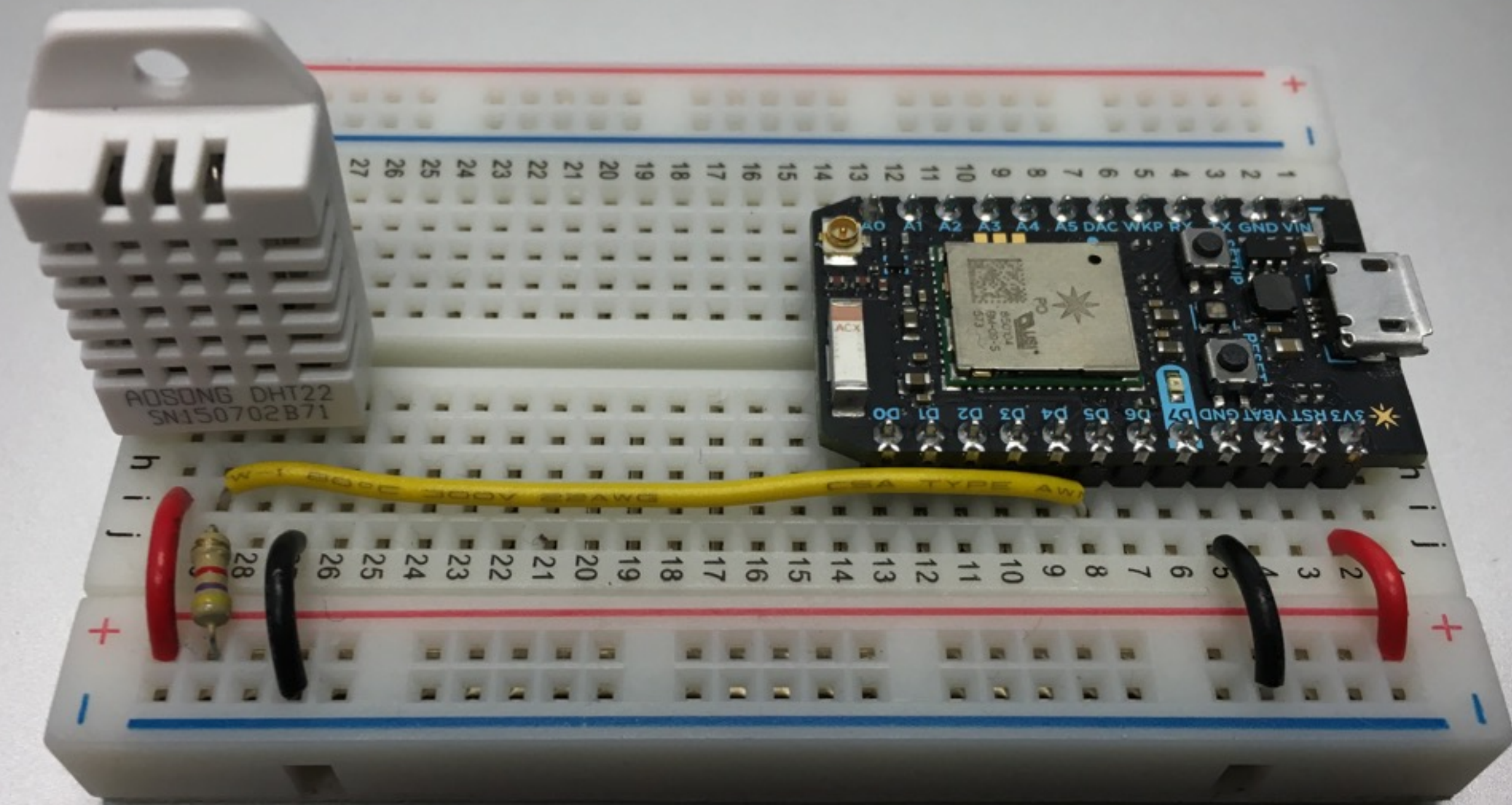
Go from a single prototype to millions of units with a cloud platform built to scale



MANAGE

Manage your products in the field: deploy firmware, collect data, and more

GO MOBILE WITH 3G!

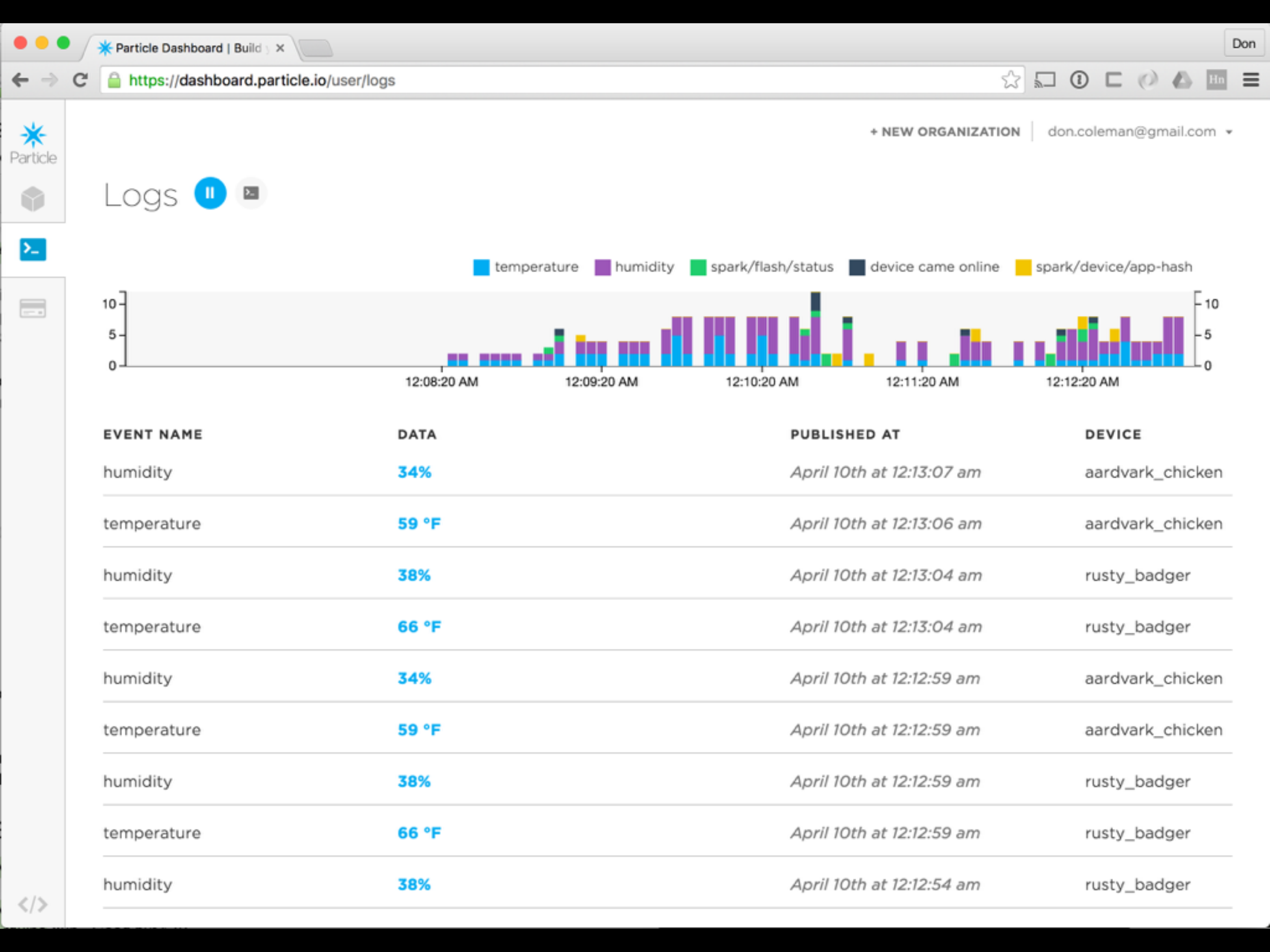




dht-temperature.ino

```
1 // This #include statement was automatically added by the Particle IDE.
2 #include "Adafruit_DHT/Adafruit_DHT.h"
3
4 // DHT parameters
5 #define DHTPIN 5
6 #define DHTTYPE DHT22
7
8 int temperature;
9 int humidity;
10
11 DHT dht(DHTPIN, DHTTYPE);
12
13 void setup() {
14     dht.begin();
15 }
16
17 void loop() {
18
19     temperature = dht.getTempFahrenheit();
20     humidity = dht.getHumidity();
21
22     // Publish data
23     Spark.publish("temperature", String(temperature) + " °F");
24     Spark.publish("humidity", String(humidity) + "%");
25     delay(5*1000);
26
27 }
```






```
[bit:~ don$ particle subscribe mine
```

```
Subscribing to all events from my personal stream (my devices only)
```

```
Listening to: /v1/devices/events
```

```
{"name":"temperature","data":"71 °F","ttl":"60","published_at":"2016-04-12T03:54:27.735Z","coreid":"1d003f001747343337363432"}
```

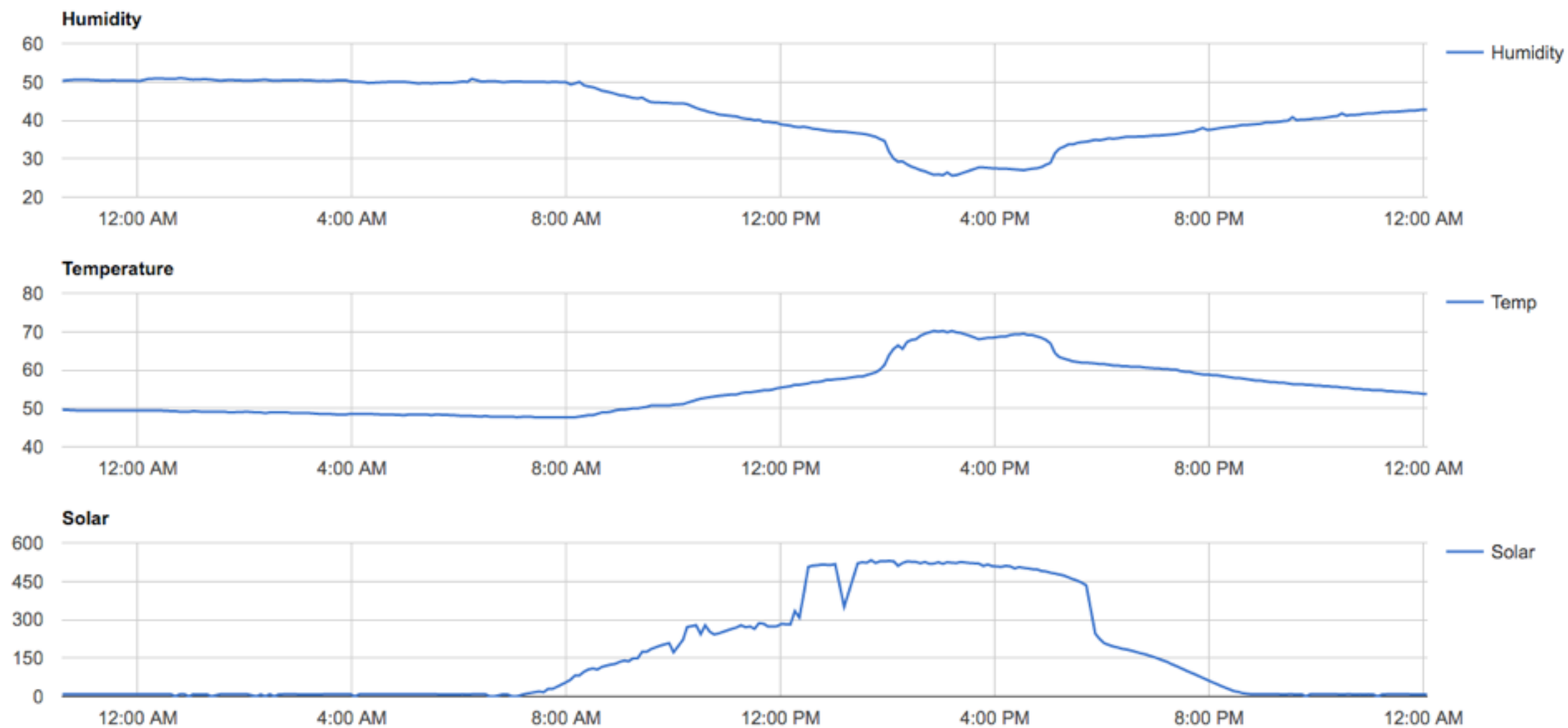
```
{"name":"humidity","data":"56%","ttl":"60","published_at":"2016-04-12T03:54:27.775Z","coreid":"1d003f001747343337363432"}
```

```
{"name":"temperature","data":"69 °F","ttl":"60","published_at":"2016-04-12T03:55:09.538Z","coreid":"2d0033000947343337373738"}
```

```
{"name":"humidity","data":"41%","ttl":"60","published_at":"2016-04-12T03:55:09.550Z","coreid":"2d0033000947343337373738"}
```

```
{"name":"temperature","data":"69 °F","ttl":"60","published_at":"2016-04-12T03:55:10.712Z","coreid":"400037000547343233323032"}
```

```
{"name":"humidity","data":"38%","ttl":"60","published_at":"2016-04-12T03:55:10.735Z","coreid":"400037000547343233323032"}
```

temperature, location=office value=72 *timestamp*


```
curl -i -XPOST \  
'http://localhost:8086/write?db=farm' \  
--data-binary 'temp,location=office value=72'
```

Query: SELECT value FROM temperature GROUP BY location LIMIT 3

Query Templates ▾

temperature

location:kitchen

time	value
2015-12-26T20:58:28.26Z	73.22
2015-12-26T20:59:28.72Z	73.04
2015-12-26T21:00:29.38Z	73.04

temperature

location:livingroom

time	value
2015-12-26T20:53:41.31Z	74.48
2015-12-26T20:54:41.79Z	74.3
2015-12-26T21:04:49.27Z	74.12

temperature

location:office

time	value
2016-01-01T19:51:25.97Z	73.58

Query: `SELECT mean(value) FROM temperature WHERE time > now() - 5d GROUP BY time(1d)`

Query Templates ▾

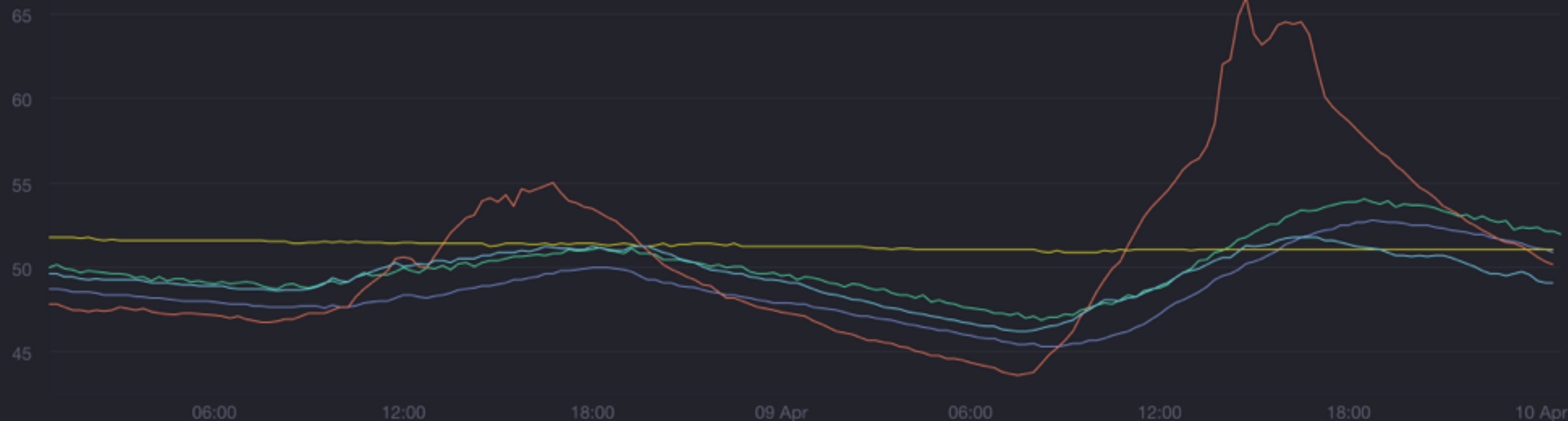
temperature

time	mean
2016-04-05T00:00:00Z	51.783532219570255
2016-04-06T00:00:00Z	51.18808035714228
2016-04-07T00:00:00Z	50.61226966292149
2016-04-08T00:00:00Z	49.98173913043446
2016-04-09T00:00:00Z	50.01026256564181
2016-04-10T00:00:00Z	52.32058823529411



Auto Refresh: 1m

Past 2 days



BUILDER



SELECT mean(value) FROM farm."default".temperature WHERE tmp1time() GROUP BY time(15m), location



FILTER BY

temperature

3

Select a tag key

EXTRACT BY

value

BY

mean

Select a field

BY

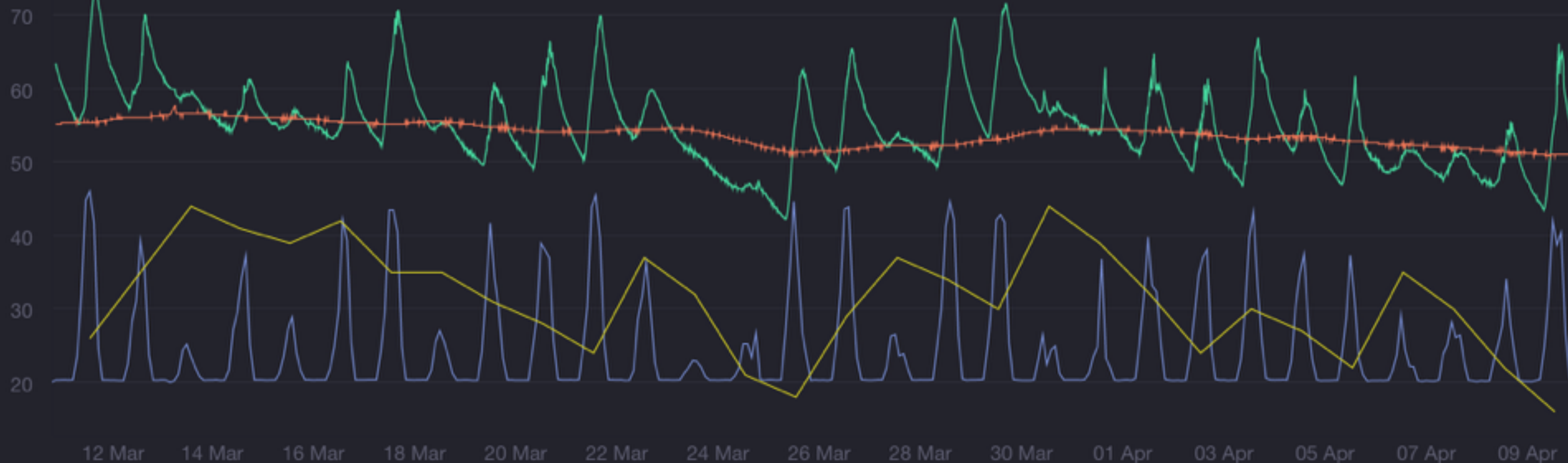
Select a function

GROUP BY



Auto Refresh: 5m

Past 30 days



BUILDER



```
SELECT sum(value)/500 + 20 FROM farm."default".solar WHERE tmplttime() GROUP BY time(2h)
```



BUILDER



```
SELECT value FROM farm."default".temperature WHERE tmplttime() AND (location = 'office' OR location = 'slab') GI
```



BUILDER



```
SELECT min FROM weather_history."default".temperature WHERE tmplttime()
```



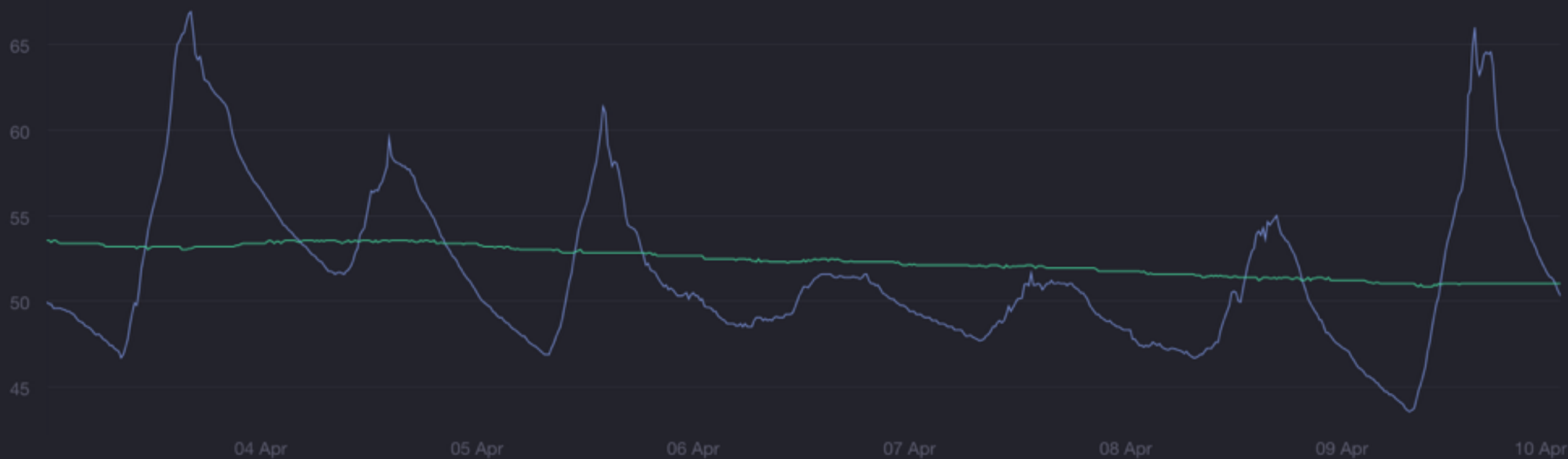
+ ADD QUERY

Add Visualization

Auto Refresh: 15s

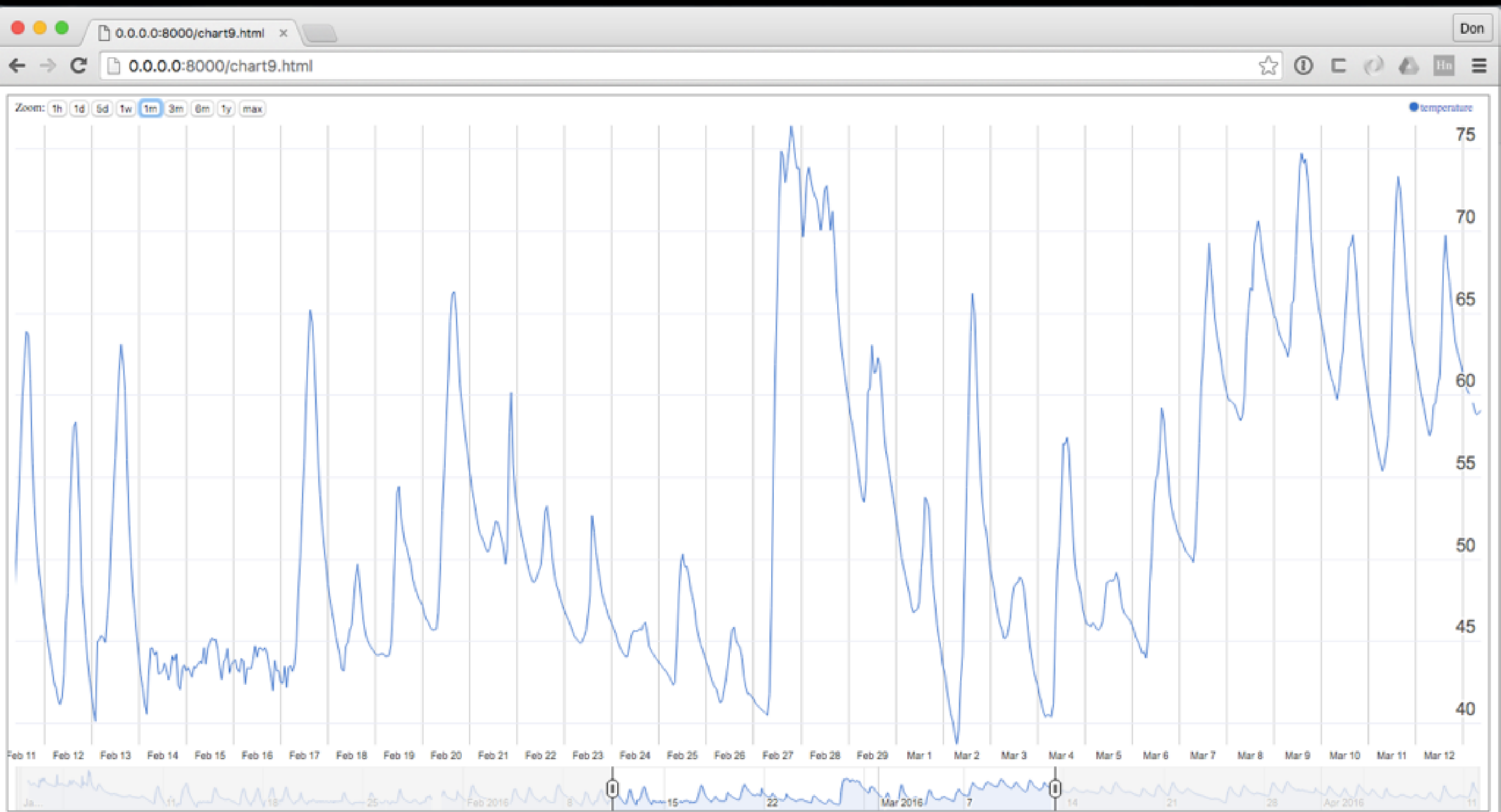
🕒 Past 7 days

Temperature (Office & Slab)



Solar







Don Coleman
Chariot Solutions
don@chariotsolutions.com
@doncoleman