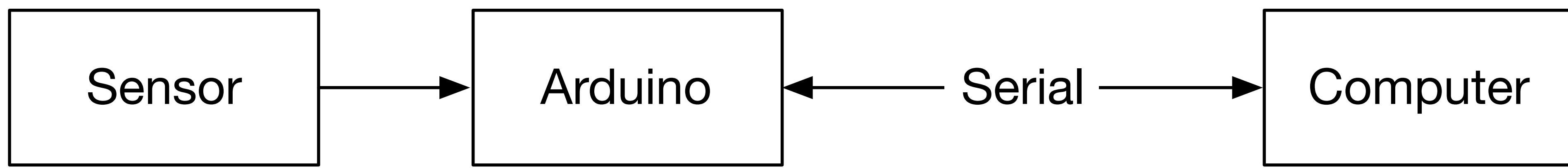
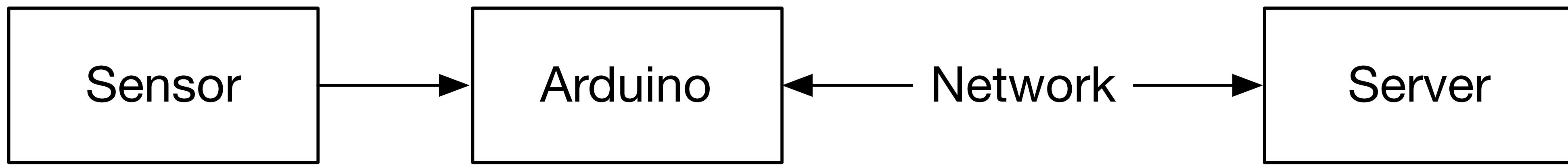
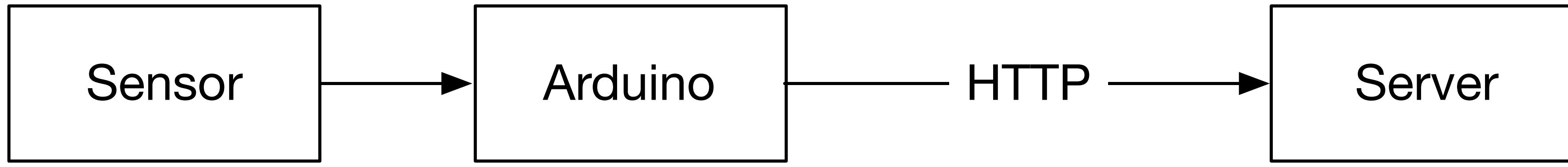


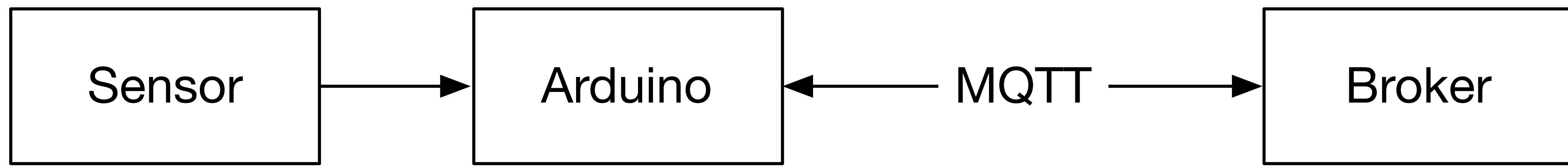
Device to Database

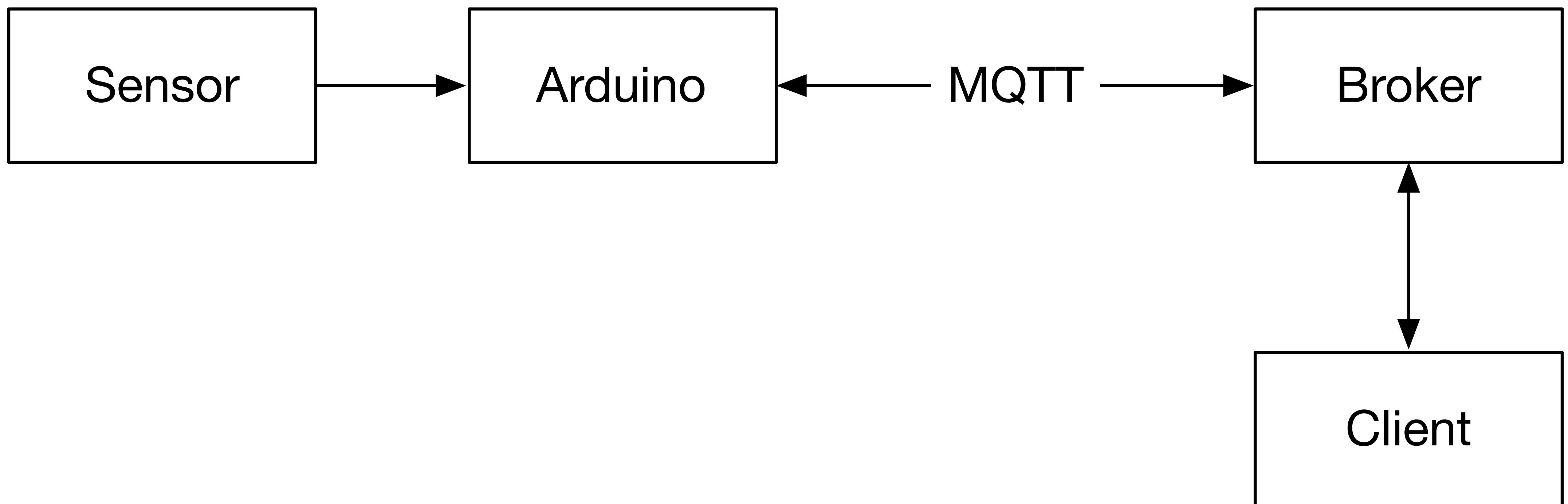
ITP Spring 2019 - Week 2

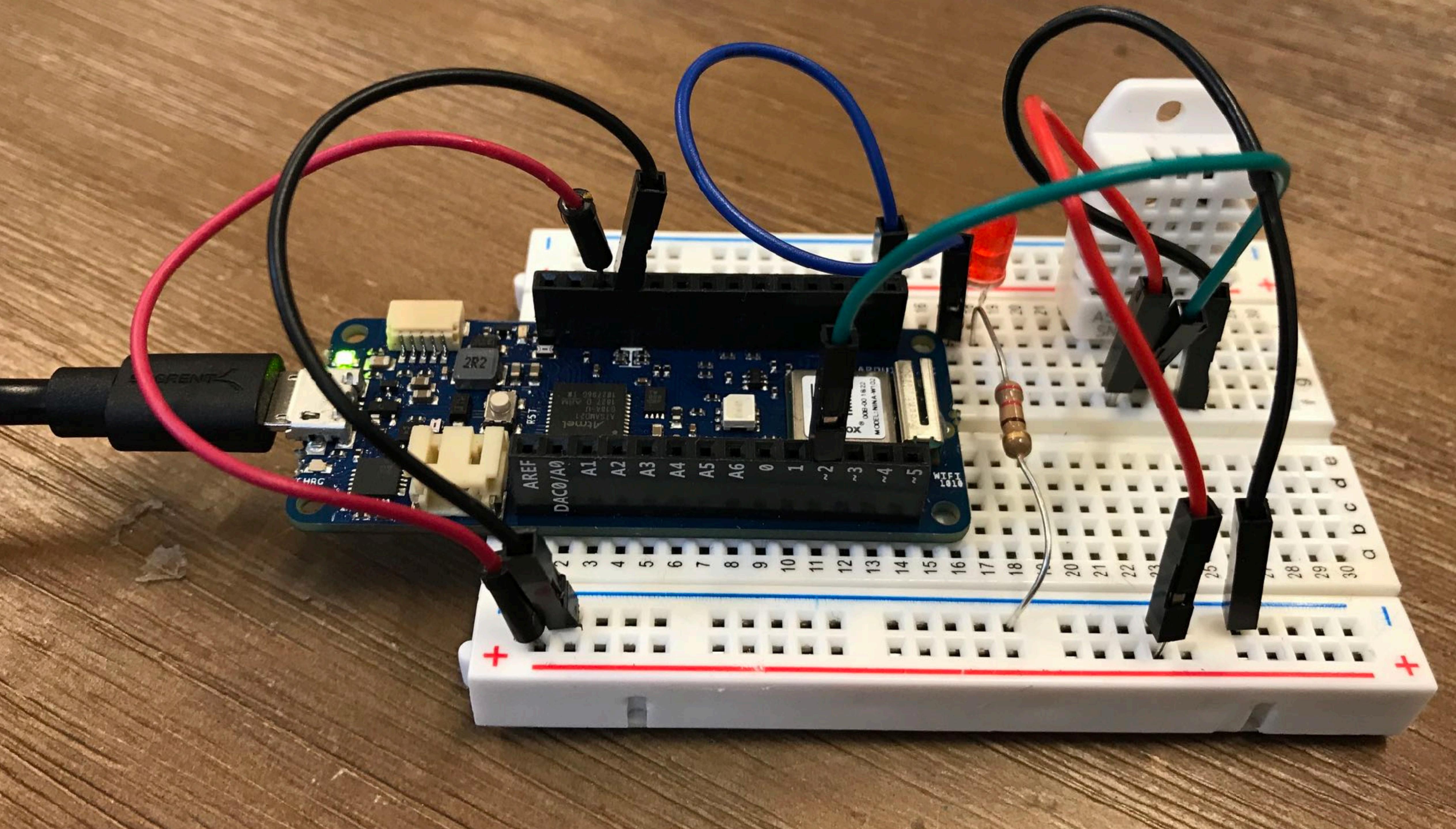


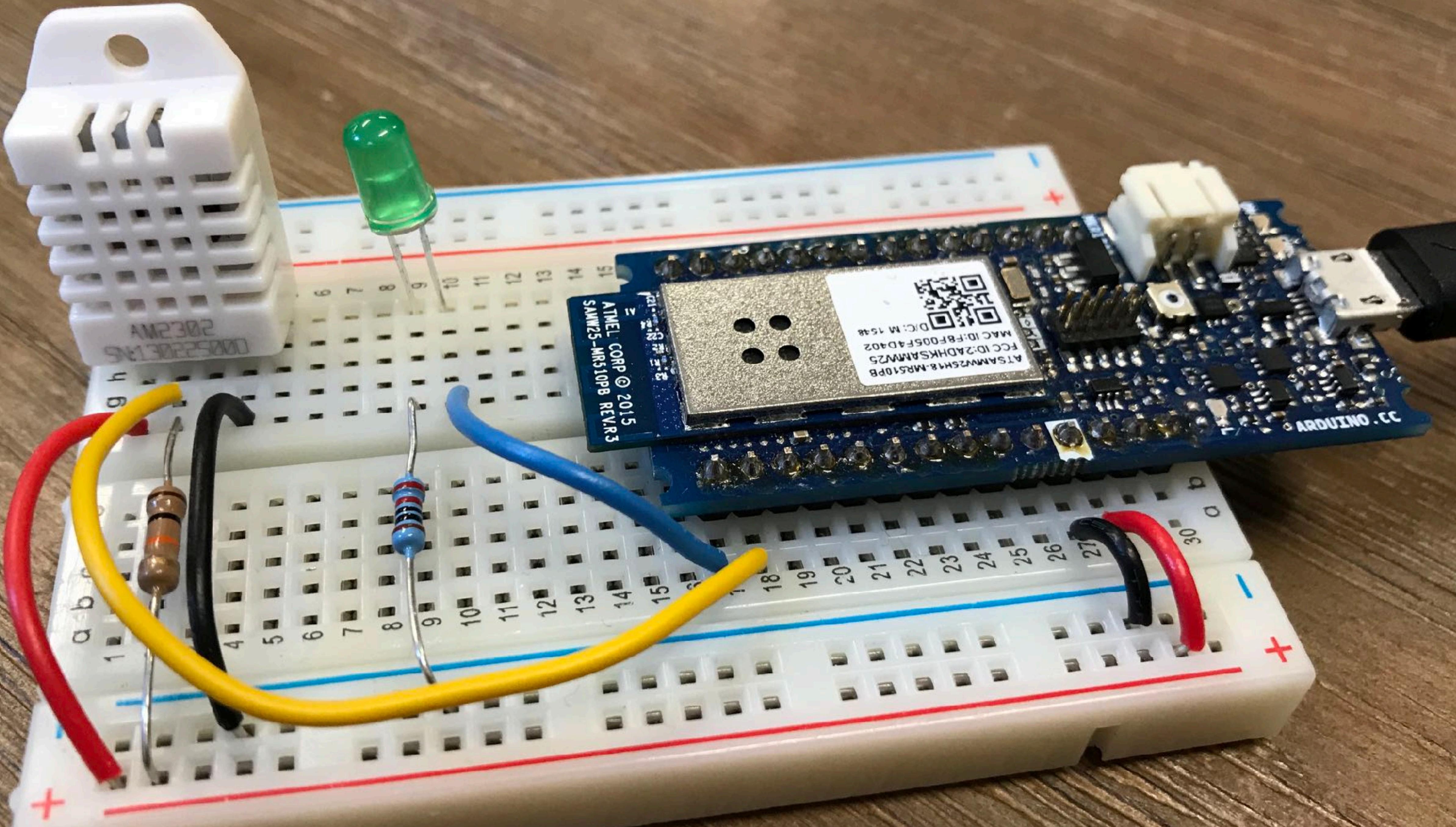


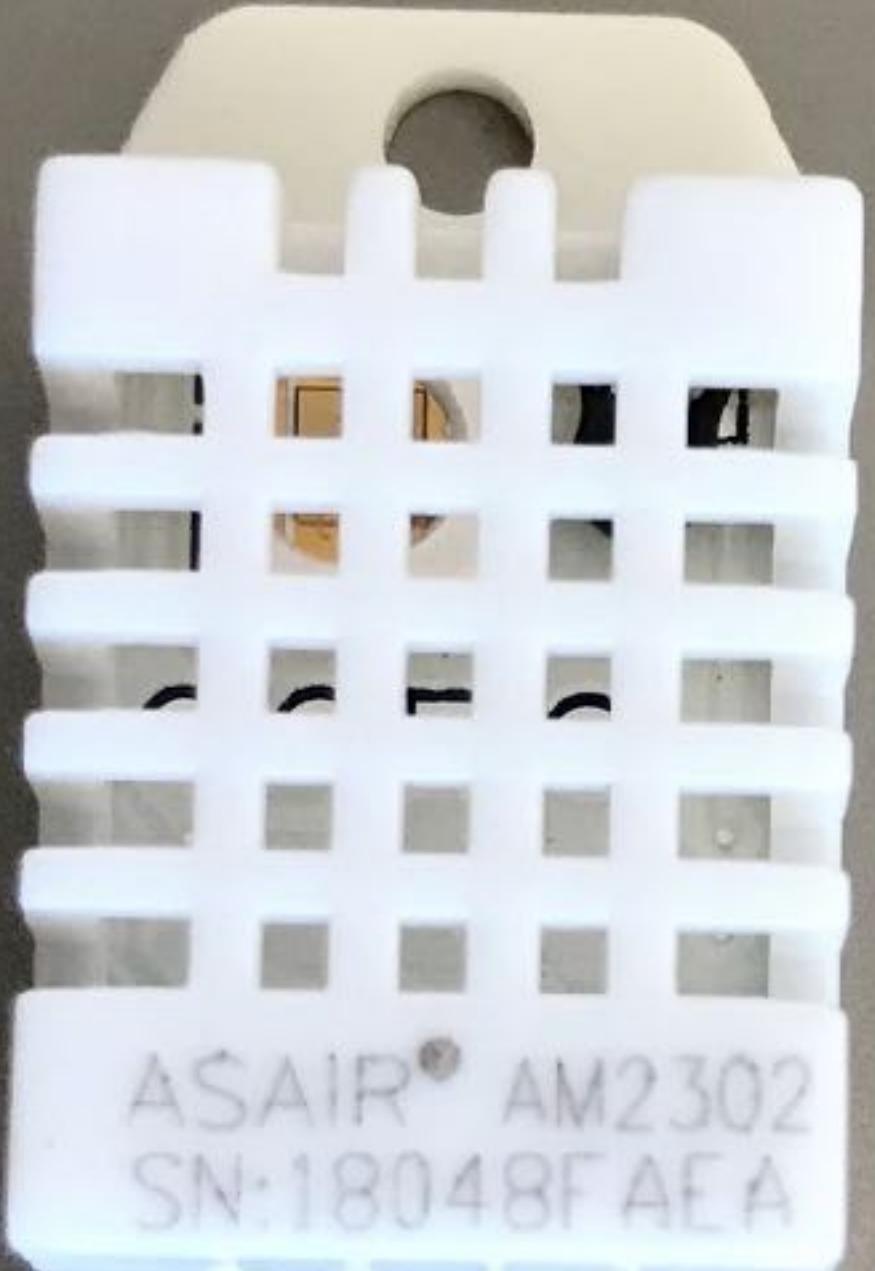




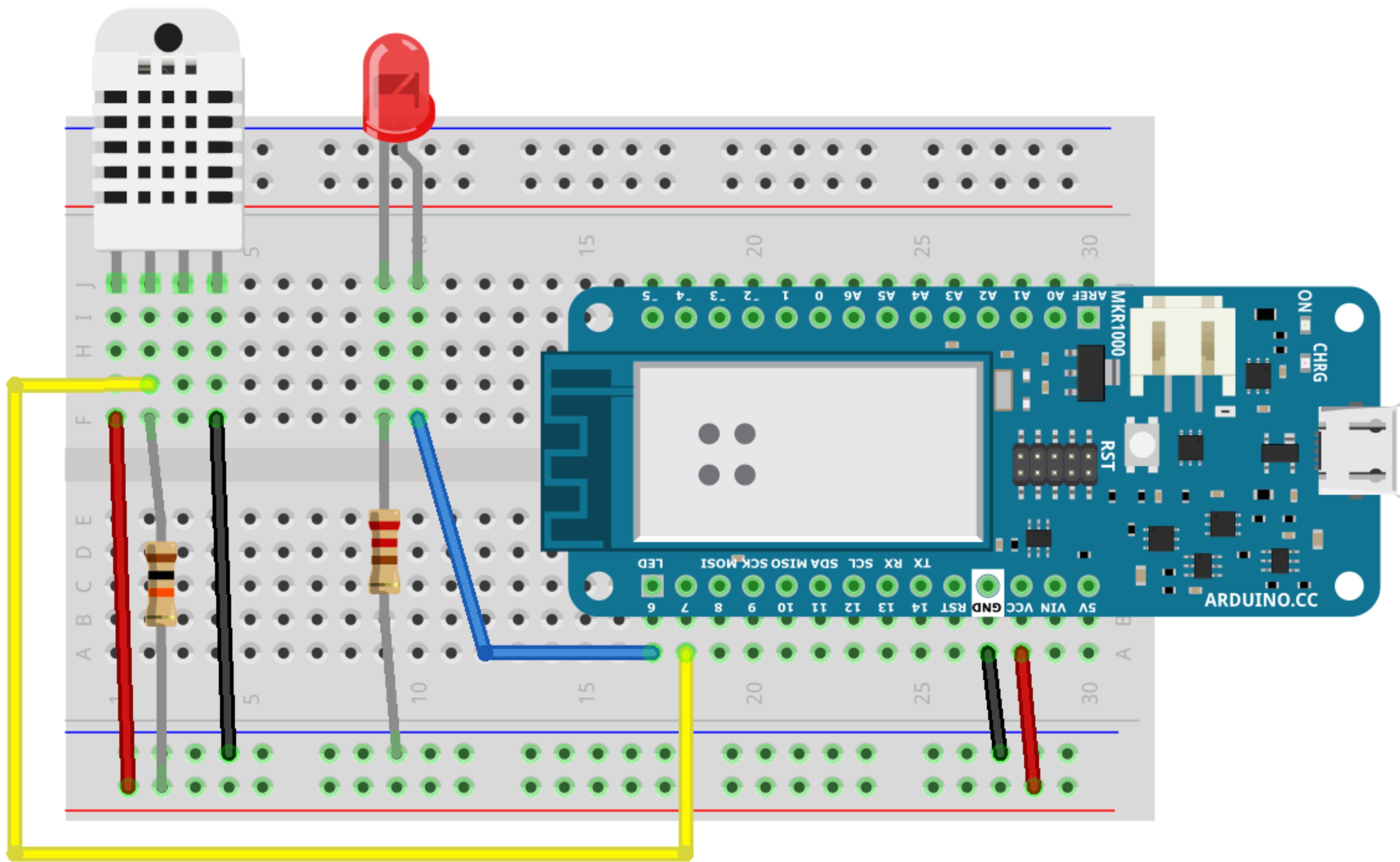








VCC Data Out Ground



fritzing

github.com/don/ITP-DeviceToDatabase

 COM3 (Arduino MKR WiFi 1010)

— □ ×

Send

81.68°F 35.20% RH
74.30°F 39.00% RH
74.12°F 38.90% RH
74.12°F 38.90% RH
74.12°F 39.00% RH
74.12°F 39.00% RH

 Autoscroll Show timestamp

Newline

9600 baud

Clear output

COM3 (Arduino MKR WiFi 1010)

— □ ×

Send

Starting connection to server...

connected to server

HTTP/1.1 200 OK

Server: nginx/1.14.0 (Ubuntu)

Date: Wed, 31 Oct 2018 02:13:42 GMT

Content-Type: text/html

Content-Length: 10

Last-Modified: Tue, 30 Oct 2018 14:36:04 GMT

Connection: close

ETag: "5bd86c54-a"

Accept-Ranges: bytes

IT WORKS!

disconnecting from server.

Autoscroll Show timestamp

Newline

9600 baud

Clear output

location/device/sensor

itp/device1/temperature

itp/device1/humidity

Write to a topic

Subscribe to a topic

Security

TemperatureHumidity.ino

1. Connect to the WiFi
2. Connect to the MQTT Server
3. Read the temperature & humidity
4. Write the data to MQTT

```
TemperatureHumidity config.h
10
11 #include <SPI.h>
12 #ifdef ARDUINO_SAMD_MKR1000
13 #include <WiFi101.h>
14 #define WL_NO_MODULE WL_NO_SHIELD
15 #else
16 #include <WiFiNINA.h>
17 #endif
18 #include <ArduinoMqttClient.h>
19
20 #include "config.h"
21
22 WiFiSSLClient net;
23 MqttClient mqtt(net);
24
```



TemperatureHumidity

config.h

```
1 const char WIFI_SSID[] = "itpsandbox";
2 const char WIFI_PASSWORD[] = "secret";
3
4 const char MQTT_BROKER[] = "mqtt.iotwork.shop";
5 const int MQTT_PORT = 8883;
6 const char MQTT_USER[] = "device_dc";
7 const char MQTT_PASSWORD[] = "secret";
8
9 const String DEVICE_ID = "device_dc";
```



TemperatureHumidity § config.h

```
24
25 // Temperature and Humidity Sensor
26 #include <DHT.h>
27 #define DHTTYPE DHT22
28 #define DHTPIN 7
29 DHT dht(DHTPIN, DHTTYPE);
30
31 String temperatureTopic = "itp/" + DEVICE_ID + "/temperature";
32 String humidityTopic = "itp/" + DEVICE_ID + "/humidity";
33
```

```
void setup() {  
    Serial.begin(9600);  
  
    // initialize temperature sensor  
    dht.begin();  
  
    Serial.println("Connecting WiFi");  
    connectWiFi();  
}
```

```
void setup() {  
    Serial.begin(9600);  
  
    // Uncomment next line to wait for a serial connection  
    // while (!Serial) { }  
  
    // initialize temperature sensor  
    dht.begin();  
  
    Serial.println("Connecting WiFi");  
    connectWiFi();  
}
```

```
void connectWiFi() {  
    while (status != WL_CONNECTED) {  
        status = WiFi.begin(wifi_ssid, wifi_password);  
        delay(3000); // wait 3 seconds  
    }  
    printWiFiStatus();  
}
```

```
void loop() {
    mqtt.poll();

    if (!mqtt.connected()) {
        connectMQTT();
    }

    if (millis() - lastMillis > publishInterval) {
        lastMillis = millis();

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

```
void connectMQTT() {
    Serial.print("Connecting MQTT...");
    mqtt.setId(clientId);
    mqtt.setUserNamePassword(username, password);

    while (!mqtt.connect(server, port)) {
        Serial.print(".");
        delay(500);
    }

    Serial.println("connected.");
}
```

```
void loop() {  
    mqtt.poll();  
  
    if (!mqtt.connected()) {  
        connectMQTT();  
    }  
  
    if (millis() - lastMillis > publishInterval) {  
        lastMillis = millis();  
  
        float temperature = dht.readTemperature(true);  
        float humidity = dht.readHumidity();  
    }  
}
```

```
if (millis() - lastMillis > publishInterval) {  
    lastMillis = millis();  
  
    float temperature = dht.readTemperature(true);  
    float humidity = dht.readHumidity();  
  
    mqtt.beginMessage(temperatureTopic);  
    mqtt.print(temperature);  
    mqtt.endMessage();  
  
    mqtt.beginMessage(humidityTopic);  
    mqtt.print(humidity);  
    mqtt.endMessage();  
}
```

Viewing Data

workshop/device00/temperature

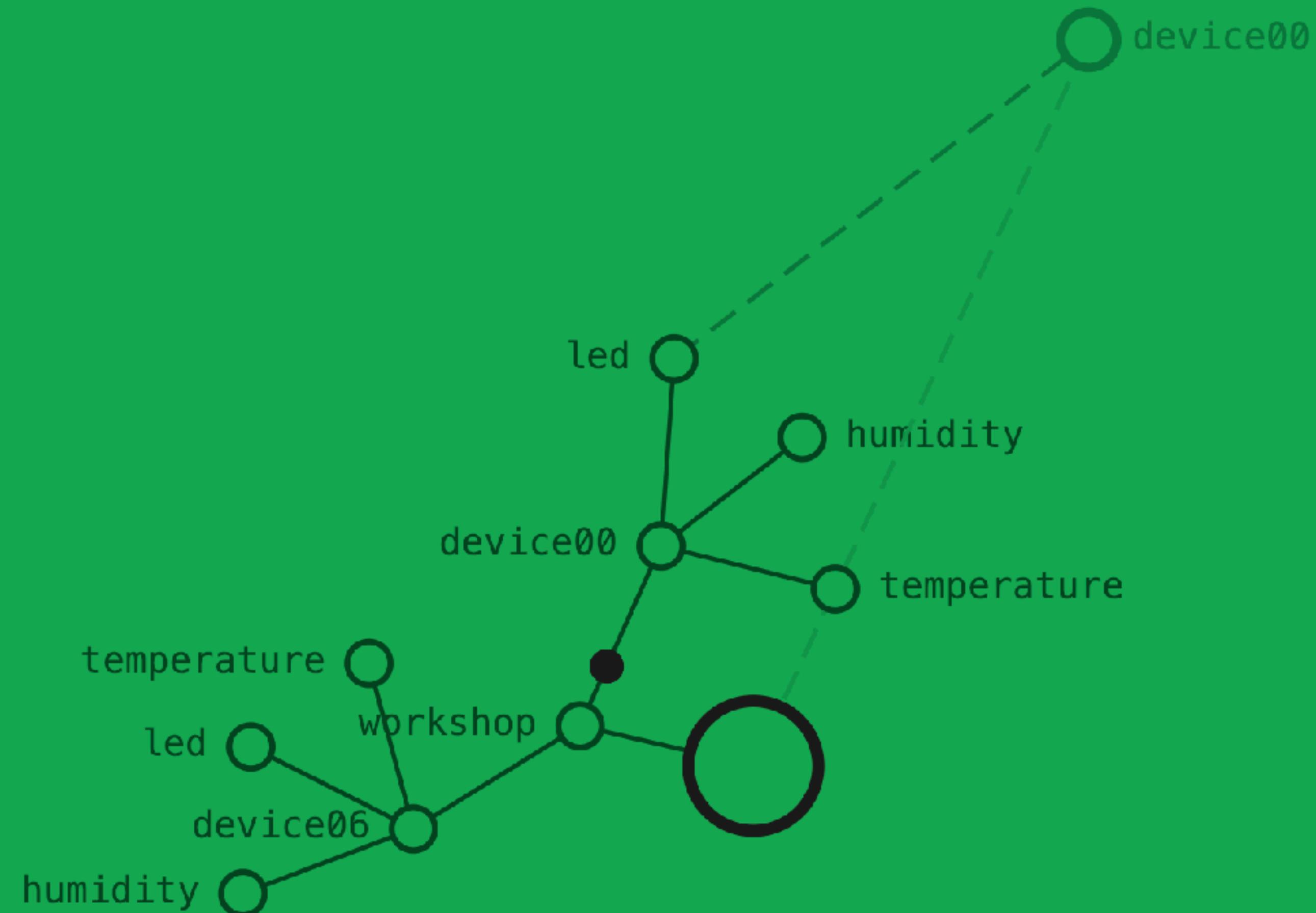
70.88

device00 - 5:41:14 PM - Q0 NR

workshop/device00/humidity

30.20

device00 - 5:41:14 PM - Q0 NR



mosquitto_sub

```
don — root@iot-workshop: ~ — ssh root@iotwork.shop — 80x24
[don@iot-workshop:~# mosquitto_sub -u user1 -P superconf18! -v -t \
workshop/device1/temperature 70.88
workshop/device1/humidity 39.30
workshop/device25/temperature 74.84
workshop/device25/humidity 39.30
workshop/device1/temperature 70.70
workshop/device1/humidity 39.10
workshop/device25/temperature 74.84
workshop/device25/humidity 39.30
workshop/device1/temperature 70.88
workshop/device1/humidity 39.30
```

www.npmjs.com/package/mqtt

Nuclear Pumpkin Mayhem

npm Enterprise Features Pricing Docs Support

Search packages

Search log in or sign up

mqtt

2.18.8 • Public • Published 2 months ago

Readme

14 Dependencies

926 Dependents

132 Versions



build passing codecov 94%



install

```
> npm i mqtt
```

weekly downloads

75,107



version

2.18.8

license

MIT

open issues

124

pull requests

8

homepage

github.com

repository

github

last publish

Subscribe to a MQTT topic

```
function connect() {
  client = mqtt.connect('wss://broker.shiftr.io',
    username: username.value,
    password: password.value
  );

  client.on('connect', function () {
    client.subscribe(topic.value);
  })

  client.on('message', function (topic, message) {
    console.log(topic, message.toString());
    pre.innerText += '\n' + topic + '\t' + message;
  })
}
```



MQTT

Username

Password

Topic

```
Wed Jan 23 2019 17:50:04 GMT-0500 (EST) workshop/device00/temperature 70.70
Wed Jan 23 2019 17:50:04 GMT-0500 (EST) workshop/device00/humidity 29.80
Wed Jan 23 2019 17:50:14 GMT-0500 (EST) workshop/device00/temperature 70.70
Wed Jan 23 2019 17:50:14 GMT-0500 (EST) workshop/device00/humidity 29.80
Wed Jan 23 2019 17:50:24 GMT-0500 (EST) workshop/device00/temperature 70.70
Wed Jan 23 2019 17:50:24 GMT-0500 (EST) workshop/device00/humidity 29.80
Wed Jan 23 2019 17:50:34 GMT-0500 (EST) workshop/device00/temperature 70.70
Wed Jan 23 2019 17:50:34 GMT-0500 (EST) workshop/device00/humidity 29.80
Wed Jan 23 2019 17:50:44 GMT-0500 (EST) workshop/device00/temperature 70.70
Wed Jan 23 2019 17:50:44 GMT-0500 (EST) workshop/device00/humidity 29.80
```

MQTT wildcards

itp/device1/+

itp/+/temperature

www.eclipse.org/paho/clients/js/utility/

eclipse paho

Components Documentation Community

Paho is an IoT.eclipse.org project

Local Storage Warning

This page uses your browser's Local Storage functionality to store MQTT messages whilst they are "inflight". This allows the Paho javascript client to ensure that QoS 1 and QoS 2 messages are successfully delivered even if the network connection or browser fails. If you do not wish for the Local Storage to be used, please do not use this client, or only use QoS 0 to send and receive messages. If you wish to see the data being stored for yourself, open up your developer console and look for the Local Storage section, as messages are sent and received, you will see entries appearing and disappearing as the messages complete their QoS 1 and QoS 2 flows. Messages are deleted as soon as they have completed their QoS flow.

More information about the Eclipse Privacy and cookie policy can be found [Here](#).

Connection - Disconnected.

| Host | Port | Client ID | Connect |
|-----------------|------|------------------|---------|
| iot.eclipse.org | 443 | js-utility-YwN1m | Connect |

| Path | Username | Password | Keepalive | Timeout | TLS | Clean Session | Automatic Reconnect |
|------|----------|----------|-----------|---------|-------------------------------------|-------------------------------------|-------------------------------------|
| /ws | | | 60 | 3 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

| Last Will Topic | QoS | Retain | Last Will Message |
|-----------------|-----|--------------------------|-------------------|
| | 0 | <input type="checkbox"/> | |

MQTT Toolbox

https://www.hivemq.com/mqtt-toolbox

HIVEMQ
ENTERPRISE MQTT BROKER

MQTT HIVEMQ RESOURCES EXTENSIONS SERVICES PRICING BLOG COMPANY

MQTT Toolbox

Talk to the experts

We use MQTT since 2012 and have questions about how MQTT and HiveMQ can help your business to evolve in a connected enterprise get in touch.

Contact us

Give HiveMQ a try

HiveMQ is the enterprise broker for enabling your business to handle massive amounts of messages and deep integrate in our existing systems.

Download HiveMQ

The best MQTT Client Tools

We gathered the authors and maintainers of all the popular MQTT utilities out there with the goal to give you an overview of all the MQTT tools available. Learn directly from the creators of the tools how to use them in your day-to-day work with MQTT.

Graphing Data

```
<body>
  <canvas id="temperatureCanvas" height="75"></canvas>
  <canvas id="humidityCanvas" height="75"></canvas>
  <script src="chart.js"></script>
  <script src="mqtt.js"></script>
</body>
```

```
client = mqtt.connect('wss://broker.shiftr.io', {  
  username: username.value,  
  password: password.value  
});  
  
client.on('connect', function () {  
  console.log('connected');  
  client.subscribe(`workshop/${deviceId.value}/temperature`);  
  client.subscribe(`workshop/${deviceId.value}/humidity`);  
});
```

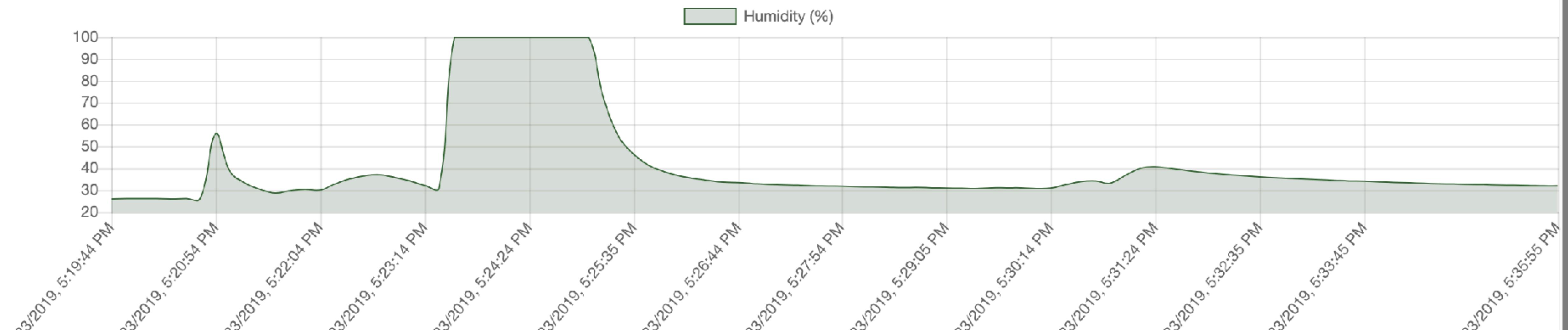
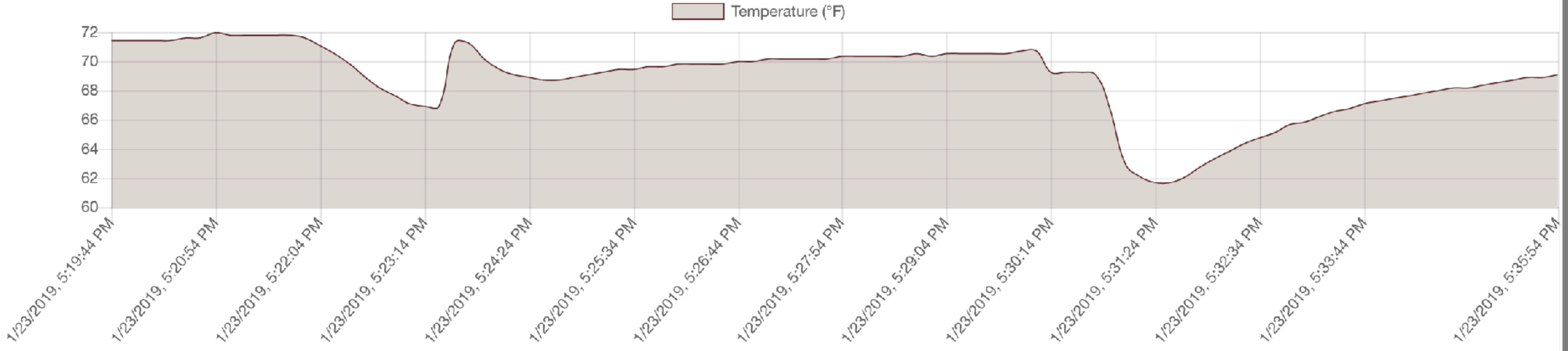
```
client.on('message', function (topic, message) {
  console.log(topic, message.toString());
  let dt = new Date().toLocaleString();

  if (topic.endsWith("temperature")) {
    temperatureData.labels.push(dt);
    temperatureData.datasets[0].data.push(parseFloat(message.toString()));
    temperatureChart.update();
  } else if (topic.endsWith("humidity")) {
    humidityData.labels.push(dt);
    humidityData.datasets[0].data.push(parseFloat(message.toString()));
    humidityChart.update();
  }
});
```

-MQTT-

| | |
|-----------|------------|
| Username | iotdevfest |
| Password | |
| Device Id | device00 |

Disconnect



Sending Data to the Arduino

LED.ino

itp/device/led

```
mqtt.subscribe(ledTopic);
```

```
mqtt.onMessage(messageReceived);
```

```
void connectMQTT() {  
    mqtt.setId(clientId);  
    mqtt.setUsernamePassword(username, password);  
  
    while (!mqtt.connect(server, port)) {  
        delay(500);  
    }  
  
    mqtt.subscribe(topic);  
}
```

```
void messageReceived(int messageSize) {  
    String payload = mqtt.readString();  
    if (payload == "ON") {  
        // turn the LED on  
        digitalWrite(LED_BUILTIN, HIGH);  
    } else if (payload == "OFF") {  
        // turn the LED off  
        digitalWrite(LED_BUILTIN, LOW);  
    }  
}
```



MQTT

Username

Password

Topic

LED

On

Off

```
function on() {
    client.publish(topic.value, 'ON');
    console.log('on');
}

function off() {
    client.publish(topic.value, 'OFF');
    console.log('off');
}
```

X.509

Securely Connecting an Arduino MKR WiFi 1010 to AWS IoT Core

<http://bit.ly/2CJVAH4>

Send

ECCX08 Serial Number = 01237F2B0D8A1817EE

The ECCX08 on your board is not locked, would you like to PERMANENTLY configure and lock it now? (y/N) [N]: Y
ECCX08 locked successfully

Hi there, in order to generate a new CSR for your board, we'll need the following information ...

Country Name (2 letter code) []:

State or Province Name (full name) []:

Locality Name (eg, city) []:

Organization Name (eg, company) []:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) [01237F2B0D8A1817EE]: device42

What slot would you like to use? (0 - 4) [0]: 0

Would you like to generate a new private key? (Y/n) [Y]: Y

Here's your CSR, enjoy!

-----BEGIN CERTIFICATE REQUEST-----

MIHNMHUCAQAwEzERMA8GA1UEAxMIZGV2aWNlNDIwMTATBgchkjOPQIBBggqhkjOPQMBBwNCAAR6
M/7tbk0UGYWJwf+318azPvjPdT4qK2dcPCRN5h/i31MTbYT0HKsb9bs+NChrbpUAXs0/lryuh/cL
i030J3xAoAAwCgYIKoZIzj0EAwIDSAAwRQIhAIx6syDeKiCKxafktzLpkLzbXQHxWYBOSwmaFMoL
qH0MAiB6q+1A24Y4Y2hJTy0XCnZcG6XCAb987duPMJnwUAMU7Q==

-----END CERTIFICATE REQUEST-----

Register thing

Certificate Signing Request

```
-----BEGIN CERTIFICATE REQUEST-----  
MIHOMHUCAQAwEzERMA8GA1UEAxMIZGV2aWNIMTcwWTATBgcqhkjOPQIBBggqhkJOPQMBwNCAASX  
iZjYMWTPg8CK5rUkkk+UEbkpO5vDpJp49IPCfRTSXIXn/Jzpmnly2pldF1K8xshYaa6Jlp7wLA+z  
CLzqa/4/oAAwCgYIKoZIzj0EAwIDSQAwRglhAJsRLjZAiMh99z9vuuoeTSn+/9/1m9C5EG8Et+A8  
sUJiAiEAxOG7atp8VZepEc00ZFwyGAvTa7IG4HHIDMiOvM3GqLU=  
-----END CERTIFICATE REQUEST-----
```

Register

Thing registered

Broker

a3jmxo5vrdefm0.iot.us-west-2.amazonaws.com

Device ID

device17

Certificate

```
-----BEGIN CERTIFICATE-----  
MIICgzCCAWugAwIBAgIUP//YHA84TPtjk/9kXebJnNjtHGwwDQYJKoZlhcNAQEL  
BQAwTTFLMEkGA1UECwxCQW1hem9uIFdIYiBTZXJ2aWNlcyBPPUfYXpvbi5jb20g  
SW5jLiBMPVNIYXR0bGUgU1Q9V2FzaGlz3RvbIBDPVVTMB4XDTE5MDEyMzIxNTk0  
N1oXDTQ5MTIzMThNTk1OVowEzERMA8GA1UEAxMIZGV2aWNIMTcwWTATBgcqhkjO  
PQIBBggqhkjOPQMBBwNCAASXiZjYMWTPg8CK5rUkkk+UEbkpO5vDpJp49IPCFRTS  
XjXn/Jzpmnly2pldF1K8xshYaa6JIp7wIA+zCLzqa/4/o2AwXjAfBgNVHSMEGDAW  
gBQwV2k0UaGqgDOJNZqKy0XiJouiYDAdBgNVHQ4EFgQUntbzfmYtYC21Kj8ChcfD  
r1SJ1zAwDAYDVR0TAQH/BAIwADAOBgNVHQ8BAf8EBAMCB4AwDQYJKoZlhcNAQEL  
BQADggEBAMJcZxuYAGjhbk2kb5lOtOD0qzTjT8X1+O8rpH/SJ3ORA+wJ2LC2BvVj  
KcEJMf4T0jqgxA36BILY1zpq0WWEfI2MyHG8HopI/Z6tWhGNT5YDfYGCScz25gk  
8Q/tIBOt5Zris8x4lnaVg2vqvdeadVJr6RNGg9SROJi+DF8mxMHVXtiP2BJfCeGH  
HvqxHMEA9LzPNr0ee6M3avwXN1cca1LKjBn4tEWicjSZxcn08Psoa4S4X+aaHZz1  
fb7QooXTPV5d9M0xUxbx/HMU/s/3HhuYvswFBnISsuIKNlROVUqxzCpstfH73zTe  
SDDNRoDb9KRDnbIUTUr8b4dLPIqURew=  
-----END CERTIFICATE-----
```

Dashboard



Register Thing

012314634AB2AB9DAF

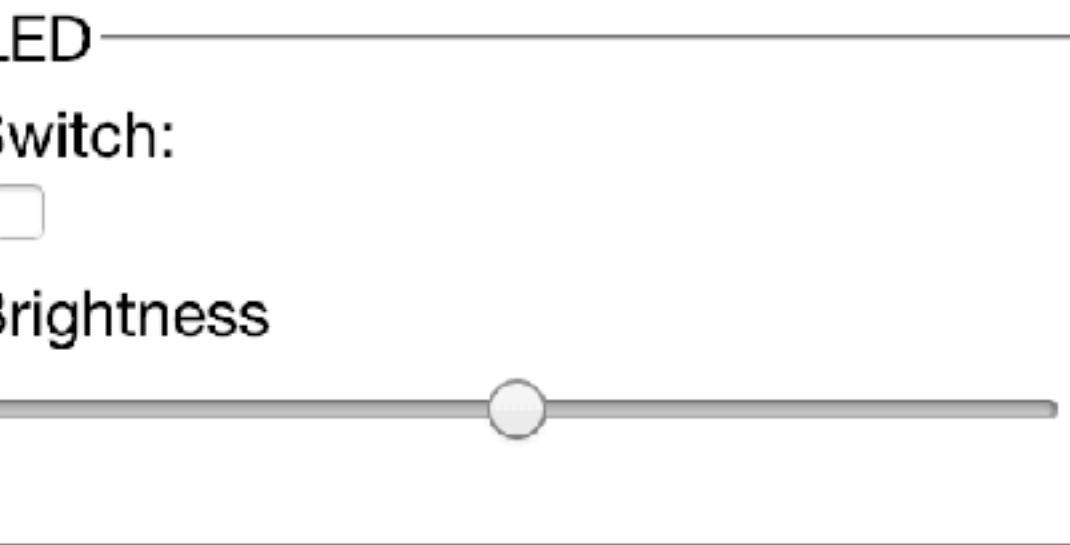
Refresh

--LED--

Switch:



Brightness



Temperature & Humidity

