

The IBM logo is displayed in white text on a blue arrow-shaped background. The arrow points to the right and is positioned to the left of the main title.

IBM

# SPRINT 1 REPORT

SMARTFARMER – IOT ENABLED  
SMART FARMING APPLICATION

**TEAM ID – PNT2022TMID34458**

- **TEAM LEADER : MUGESHWARAN G**
- **TEAM MEMBER : ISRAVEL KEWIN CLINT P**
- **TEAM MEMBER : BLESSWIN.K.SAMUEL**
- **TEAM MEMBER : VIJAY S**

## Project Tracker

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
Sprint-1	15	5 Days	26 Oct 2022	30 Oct 2022	15	30 Oct 2022
Sprint-2	15	7 Days	31 Oct 2022	06 Nov 2022		07 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022		13 Nov 2022
Sprint-4	15	6 Days	13 Nov 2022	18 Nov 2022		18 Nov 2022 – 19 Nov 2022

<b>S.NO</b>	<b>Tools &amp; Technology Used</b>
1	Wokwi Hardware Simulation
2	Tinkercad Hardware Simulation
3	IBM Watson IOT Platform
4	IOT Monitor Board
5	Python Hardware Simulation

# Wokwi Hardware Simulation

Activities Firefox Web Browser Nov 16 2:40 PM

W esp32-dht22.ino copy Smart Farm | Tinkerc Smart Farm | Tinkerc (1) WhatsApp Sprint Delivery Plan.pdf IBM-Project-46651-1 IBM Cloud

← → ↻ https://wokwi.com/projects/348317004234490452

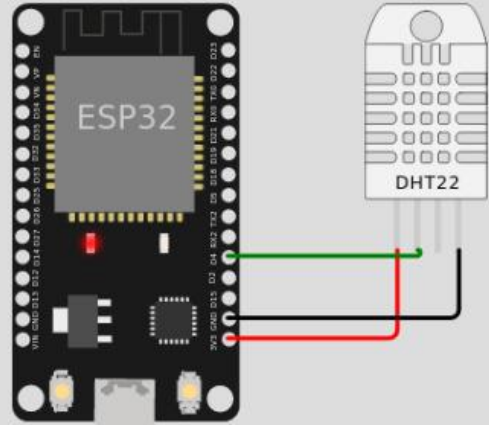
WOKWI SAVE SHARE esp32-dht22.ino copy Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 4 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define LED 5
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type
8
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "gstsmj"
14 #define DEVICE_TYPE "IOT_Device"
15 #define DEVICE_ID "Smart_Farming"
16 #define TOKEN "IOT_Device_12345"
17 String data3;
18 float h, t;
19
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
24 char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29
30 //-----
31 WiFiClient wifiClient; // creating the instance for wifi client
```

Simulation

00:02.833 19%



Temperature:80.00  
Humidity:100.00  
Sending payload: {"Temperature":80.00,"Humidity":100.00}  
Publish ok  
Temperature:80.00  
Humidity:100.00  
Sending payload: {"Temperature":80.00,"Humidity":100.00}

# Tinkercad Hardware Simulation

W esp32-dht22.ino copy x Circuit design Smart x (1) WhatsApp x Sprint Delivery Plan.pdf x IBM-Project-46651-1 x IBM Cloud x + -

← → ↻ https://www.tinkercad.com/things/eORvZ5UhoPa-smart-farm/editel ☆

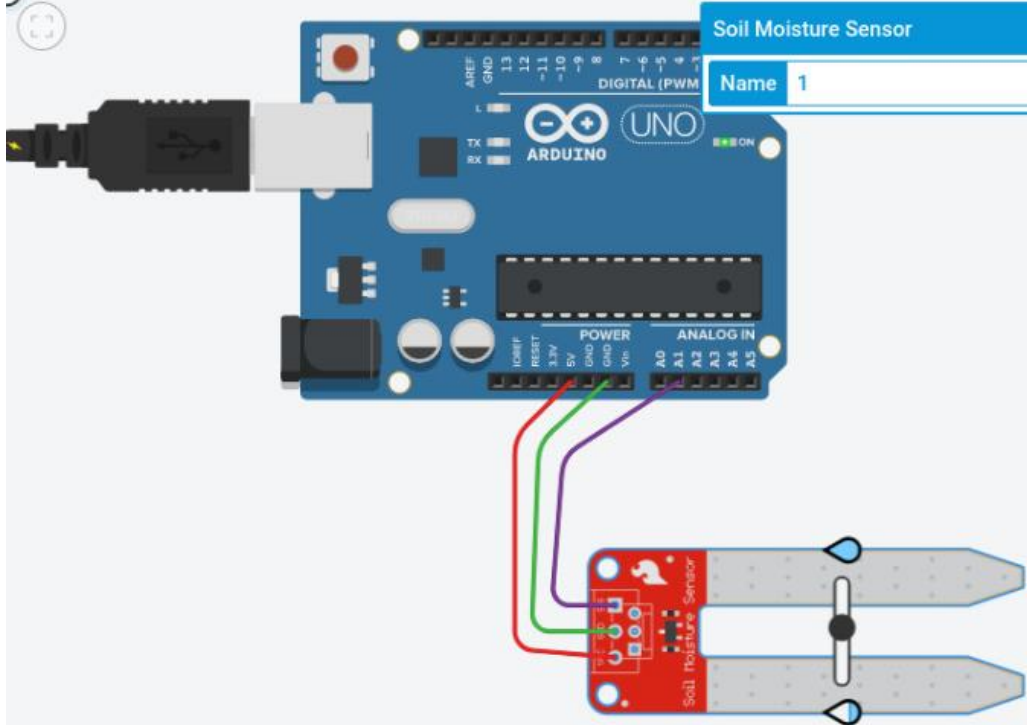
TINKERCAD Smart Farm All changes saved [Icons] [User Profile]

Simulator time: 00:00:29 [Code] [Stop Simulation] [Send To]

1 (Arduino Uno R3)

Soil Moisture Sensor

Name 1



```
1 #define S_sensorPin A1 // Arduino pin that connects to AOUT pin of m
2
3
4 void setup() {
5   Serial.begin(9600); // Begin serial communication at 9600 baud ra
6 }
7
8 void loop() {
9
10  int S_value = analogRead(S_sensorPin); // read the analog value f
11
12  if (S_value < 100)
13    Serial.print("The soil is DRY (");
14  else
15    Serial.print("The soil is WET (");
16
17  Serial.print(S_value);
18  Serial.println(")");
19
20  delay(1000);
21 }
```

Serial Monitor

The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)  
The soil is WET (518)

[Input Field] [Send] [Clear] [Waveform Icon]

# Ibm Watson IOT Platform

Activities Firefox Web Browser Nov 16 2:54 PM

W esp32-dht22.ino x Circuit design Sm x (1) WhatsApp x Sprint Delivery Plan x IBM-Project-466 x Service Details x IBM Watson IoT x

https://gtsmj.internetofthings.ibmcloud.com/dashboard/devices/browse

## IBM Watson IoT Platform

Browse Action Device Types Interfaces

Device ID	Status	Device Type
IOT_Device_1	Disconnected	IOT_Device
Smart_Farming	Disconnected	IOT_Device

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format
event_1	{"Temperature":19,"Humidity":11,"Soil_Moisture":3,"Motor_Pump":"OFF"}	json
event_1	{"Temperature":41,"Humidity":68,"Soil_Moisture":86,"Motor_Pump":"OFF"}	json
event_1	{"Temperature":35,"Humidity":23,"Soil_Moisture":61,"Motor_Pump":"OFF"}	json
event_1	{"Temperature":34,"Humidity":64,"Soil_Moisture":50,"Motor_Pump":"OFF"}	json
event_1	{"Temperature":42,"Humidity":37,"Soil_Moisture":80,"Motor_Pump":"OFF"}	json

Items per page 50 | 1-3 of 3 items

### Smart\_Farming

Events 1

Event type name event\_1 Frequency 20 x Every Minute Send

Payload

You can override field values in the event payload that is sent by this device. Specify the override values in the editor window.

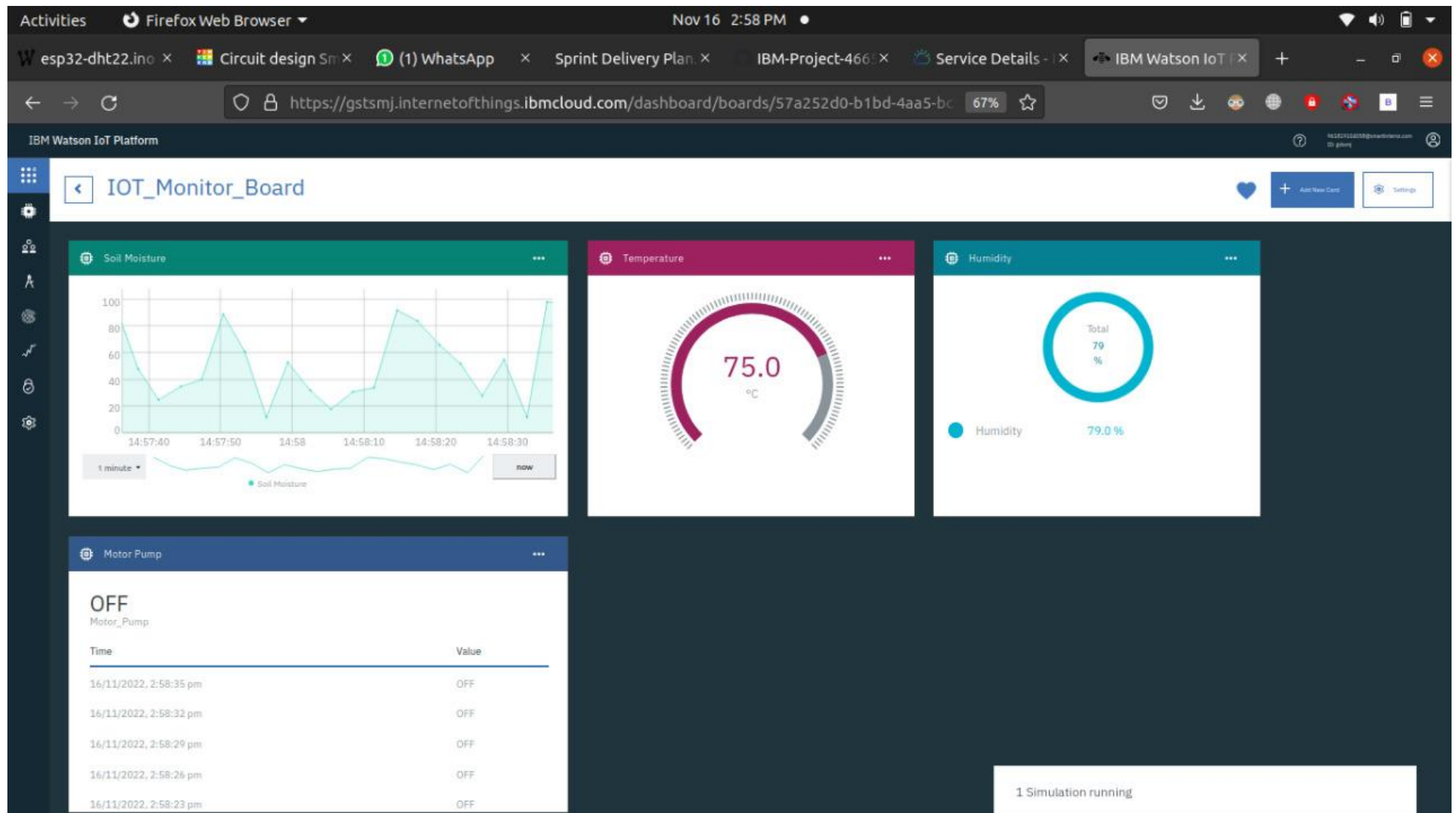
```
0 {
1   "Temperature": random(0, 100),
2   "Humidity": random(0, 100),
3   "Soil_Moisture": random(0, 100),
4   "Motor_Pump": "OFF"
5 }
6
```

What functions can I apply?

Device Nov 15, 2022 7:34 PM

1 of 1 page < 1 >

# IOT Monitor Board





# Python Hardware Simulation

The screenshot shows the Visual Studio Code interface with a Python file named `ibmtopy.py` open. The script imports `time`, `sys`, `ibmiotf.application`, `ibmiotf.device`, and `random`. It configures IBM Watson IoT credentials and device information, then initializes GPIO and sets a motor status. A `myCommandCallback` function is defined. The terminal shows the command `python3 -u "/home/nothing/Desktop/HTML-1/HTML/ibmtopy.py"` being executed, resulting in a successful connection to the IBM Watson IoT platform and a series of simulated sensor data points.

```
ibmtopy.py
1 import time
2 import sys
3 import ibmiotf.application
4 import ibmiotf.device
5 import random
6 #Provide your IBM Watson Device Credentials
7 organization = "gstsmj"
8 deviceType = "IOT_Device"
9 deviceId = "Smart_Farming"
10 authMethod = "token"
11 authToken = "IOT_Device_12345"
12 # Initialize GPIO
13 M_status="OFF"
14 def myCommandCallback(cmd):
```

python3 -u "/home/nothing/Desktop/HTML-1/HTML/ibmtopy.py"  
nothing@M-HP:~/Desktop/HTML-1/HTML\$ python3 -u "/home/nothing/Desktop/HTML-1/HTML/ibmtopy.py"  
2022-11-16 15:19:45,029 ibmiotf.device.Client INFO Connected successfully: d:gstsmj:IOT\_Device:Smart\_Farming  
Published Temperature = 19 C Humidity= 59 % Soil\_Moisture= 89 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 28 C Humidity= 64 % Soil\_Moisture= 42 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 76 C Humidity= 65 % Soil\_Moisture= 27 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 75 C Humidity= 63 % Soil\_Moisture= 86 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 29 C Humidity= 62 % Soil\_Moisture= 94 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 60 C Humidity= 31 % Soil\_Moisture= 65 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 54 C Humidity= 47 % Soil\_Moisture= 63 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 99 C Humidity= 25 % Soil\_Moisture= 70 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 25 C Humidity= 76 % Soil\_Moisture= 1 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 30 C Humidity= 65 % Soil\_Moisture= 100 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 73 C Humidity= 50 % Soil\_Moisture= 80 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 62 C Humidity= 92 % Soil\_Moisture= 7 % Motor\_Pump\_Status = OFF to IBM Watson  
Published Temperature = 40 C Humidity= 75 % Soil\_Moisture= 87 % Motor\_Pump\_Status = OFF to IBM Watson



**THANK YOU**