

## FINAL PROJECT

<b>DATE</b>	20 <sup>th</sup> may 2023
<b>TEAM ID</b>	NM2023TMID10960
<b>PROJECT</b>	SMART BILLING SYSTEM FOR WATER SUPPLIERS

### CODE:

<https://wokwi.com/projects/365238558222190593>

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
// Define the I2C address for the LCD display
```

```
#define LCD_ADDRESS 0x27
```

```
// Define the pins for flow sensor and LCD display
```

```
#define FLOW_SENSOR_PIN 2
```

```
#define LCD_COLS 16
```

```
#define LCD_ROWS 2
```

```
// Global variables
```

```
volatile unsigned int pulseCount = 0;
```

```
float flowRate = 8.0;
```

```
float totalLiters = 1.0;

unsigned long prevMillis = 0;

float billingRate = 20.0; // Cost per liter


// LCD display object
LiquidCrystal_I2C lcd(LCD_ADDRESS, LCD_COLS, LCD_ROWS);


// Interrupt service routine for flow sensor
void pulseCounter()
{
    pulseCount++;
}


// Setup function
void setup()
{
    // Initialize LCD display
    lcd.begin(LCD_COLS, LCD_ROWS);
    lcd.print("Water Billing");
    lcd.setCursor(0, 1);
```

```
lcd.print("System");

// Attach interrupt to flow sensor pin
attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN),
pulseCounter, FALLING);

// Initialize serial communication
Serial.begin(9600);
}

// Loop function
void loop()
{
    unsigned long currentMillis = millis();
    unsigned long elapsedTime = currentMillis - prevMillis;

    // Update flow rate every second
    if (elapsedTime >= 1000)
    {
        detachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN));
```

```
    flowRate = pulseCount / (elapsedTime / 1000.0);  
    pulseCount = 0;  
    prevMillis = currentMillis;  
    attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN),  
pulseCounter, FALLING);  
}
```

```
// Calculate total liters
```

```
float liters = flowRate / 60.0;
```

```
totalLiters += liters;
```

```
// Calculate bill amount
```

```
float billAmount = totalLiters * billingRate;
```

```
// Display data on LCD
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Liters: ");
```

```
lcd.print(totalLiters);
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Bill: $");
```

```
lcd.print(billAmount, 2);
```

```
// Send data to serial monitor
```

```
Serial.print("Liters: ");
```

```
Serial.print(totalLiters);
```

```
Serial.print("  Bill: $");
```

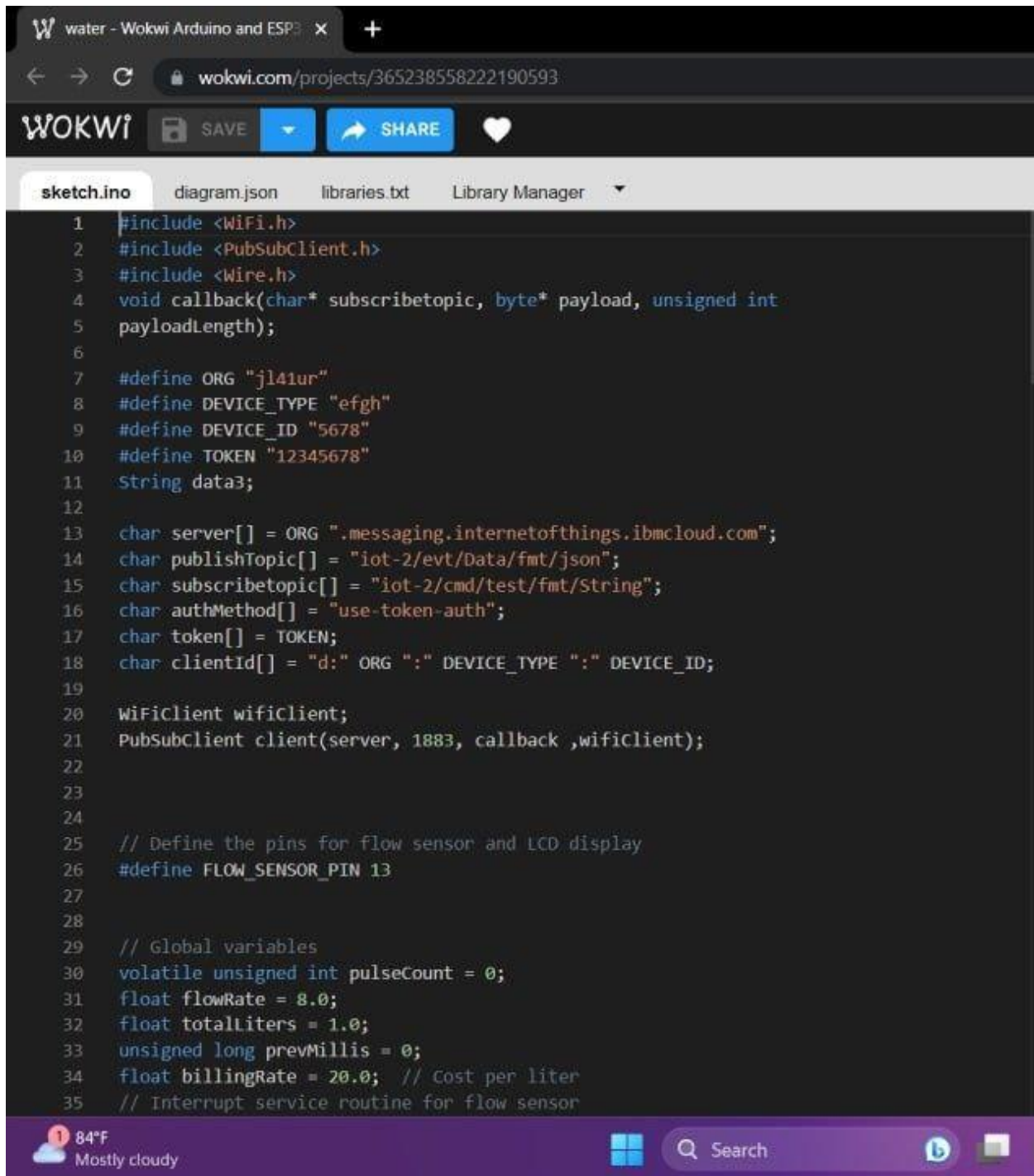
```
Serial.println(billAmount, 2);
```

```
// Wait for a second
```

```
delay(1000);
```

```
}
```

## SCHEMATIC:



The screenshot shows the Wokwi web IDE interface. The browser address bar displays `wokwi.com/projects/365238558222190593`. The Wokwi logo and navigation buttons (SAVE, SHARE, and a heart icon) are visible. The sketch editor shows the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <Wire.h>
4 void callback(char* subscribetopic, byte* payload, unsigned int
5 payloadLength);
6
7 #define ORG "jl41ur"
8 #define DEVICE_TYPE "efgh"
9 #define DEVICE_ID "5678"
10 #define TOKEN "12345678"
11 String data3;
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/Data/fmt/json";
15 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19
20 WiFiClient wificlient;
21 PubSubClient client(server, 1883, callback ,wificlient);
22
23
24
25 // Define the pins for flow sensor and LCD display
26 #define FLOW_SENSOR_PIN 13
27
28
29 // Global variables
30 volatile unsigned int pulseCount = 0;
31 float flowRate = 8.0;
32 float totalLiters = 1.0;
33 unsigned long prevMillis = 0;
34 float billingRate = 20.0; // Cost per liter
35 // Interrupt service routine for flow sensor
```

The bottom status bar shows a temperature of 84°F, weather as 'Mostly cloudy', a search bar, and social media icons for GitHub and Twitter.

W water - Wokwi Arduino and ESP3 x +

← → ↻ wokwi.com/projects/365238558222190593

WOKWI SAVE SHARE

sketch.ino diagram.json libraries.txt Library Manager

```
34 float billingRate = 20.0; // Cost per liter
35 // Interrupt service routine for flow sensor
36 void pulseCounter()
37 {
38   pulseCount++;
39 }
40
41
42 void setup() {
43   // Attach interrupt to flow sensor pin
44   attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), pulseCounter, FALLING);
45
46   // Initialize serial communication
47   Serial.begin(9600);
48
49
50   wificonnect();
51   mqttconnect();
52 }
53
54 void loop()
55 {
56   unsigned long currentMillis = millis();
57   unsigned long elapsedTime = currentMillis - prevMillis;
58
59   // Update flow rate every second
60   if (elapsedTime >= 1000)
61   {
62     detachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN));
63     flowRate = pulseCount / (elapsedTime / 1000.0);
64     pulseCount = 0;
65     prevMillis = currentMillis;
66     attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), pulseCounter, FALLING);
67   }
68   // Calculate total liters
```

84°F Mostly cloudy

Search

W water - Wokwi Arduino and ESP32 X +

← → ↻

wokwi.com/projects/365238558222190593

WOKWI

SAVE

SHARE

♥

sketch.ino

diagram.json

libraries.txt

Library Manager

```
67 }
68 // Calculate total liters
69 float liters = flowRate / 60.0;
70 totalLiters += liters;
71
72 // Calculate bill amount
73 float billAmount = totalLiters * billingRate;
74 // Send data to serial monitor
75 Serial.print("Liters: ");
76 Serial.print(totalLiters);
77 Serial.print("    Bill: $");
78 Serial.println(billAmount, 2);
79
80 // Wait for a second
81 delay(1000);
82 }
83
84 void PublishData(float dist) {
85   mqttconnect();
86   String payload = "{\"Bill\": ";
87   payload += dist;
88   payload += ", \"ALERT!!\": \"\" \"Billamount is more\"";
89   payload += "\"}";
90   Serial.print("Sending payload: ");
91   Serial.println(payload);
92   if (client.publish(publishTopic, (char*) payload.c_str())) {
93     Serial.println("Publish ok");
94   } else {
95     Serial.println("Publish failed");
96   }
97 }
98
99 void mqttconnect() {
100   if (!client.connected()) {
101     Serial.print("Reconnecting client to ");
```

84°F

Mostly cloudy




Q

Search



W water - Wokwi Arduino and ESP32 x +




← → ↻ wokwi.com/projects/365238558222190593

WOKWI  SAVE  SHARE 

sketch.ino diagram.json libraries.txt Library Manager ▾

```
100 if (!client.connected()) {
101   Serial.print("Reconnecting client to ");
102   Serial.println(server);
103   while (!client.connect(clientId, authMethod, token)) {
104     Serial.print(".");
105     delay(500);
106   } initManagedDevice();
107   Serial.println();
108 }
109 }
110
111 void wificonnect()
112 {
113   Serial.println();
114   Serial.print("Connecting to ");
115   WiFi.begin("Wokwi-GUEST", "", 6);
116   while (WiFi.status() != WL_CONNECTED) {
117     delay(500);
118     Serial.print(".");
119   }
120   Serial.println("");
121   Serial.println("WiFi connected");
122   Serial.println("IP address: ");
123   Serial.println(WiFi.localIP());
124 }
125
126 void initManagedDevice() {
127   if (client.subscribe(subscribetopic)) {
128     Serial.println((subscribetopic));
129     Serial.println("subscribe to cmd OK");
130   } else {
131     Serial.println("subscribe to cmd FAILED");
132   }
133 }
134
135 void callback(char* subscribetopic, byte* payload, unsigned int
```

84°F  
Mostly cloudy

  Search 

W water - Wokwi Arduino and ESP32 X +

← → ↻

wokwi.com/projects/365238558222190593

WOKWI

SAVE

SHARE

♥

sketch.ino

diagram.json

libraries.txt

Library Manager

```
127 if (client.subscribe(subscribetopic)) {
128   Serial.println((subscribetopic));
129   Serial.println("subscribe to cmd OK");
130 } else {
131   Serial.println("subscribe to cmd FAILED");
132 }
133 }
134
135 void callback(char* subscribetopic, byte* payload, unsigned int
136 payloadLength)
137 {
138   Serial.print("callback invoked for topic: ");
139   Serial.println(subscribetopic);
140   for (int i = 0; i < payloadLength; i++)
141   { //Serial.print((char)payload[i]);
142     data3 += (char)payload[i];
143   }
144   Serial.println("data: " + data3);
145   data3="";
146 }
147
148
149
```

1

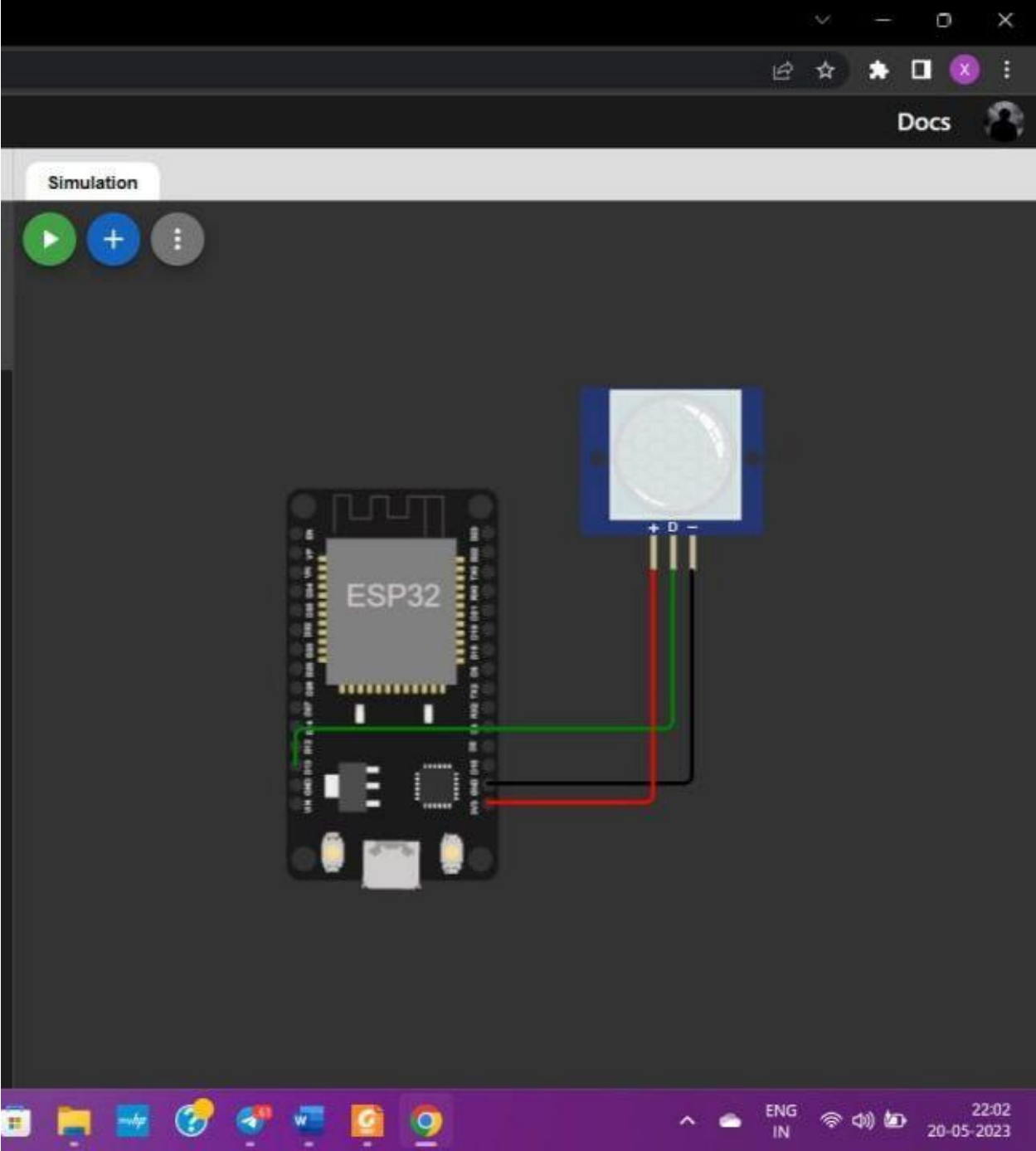
84°F  
Mostly cloudy

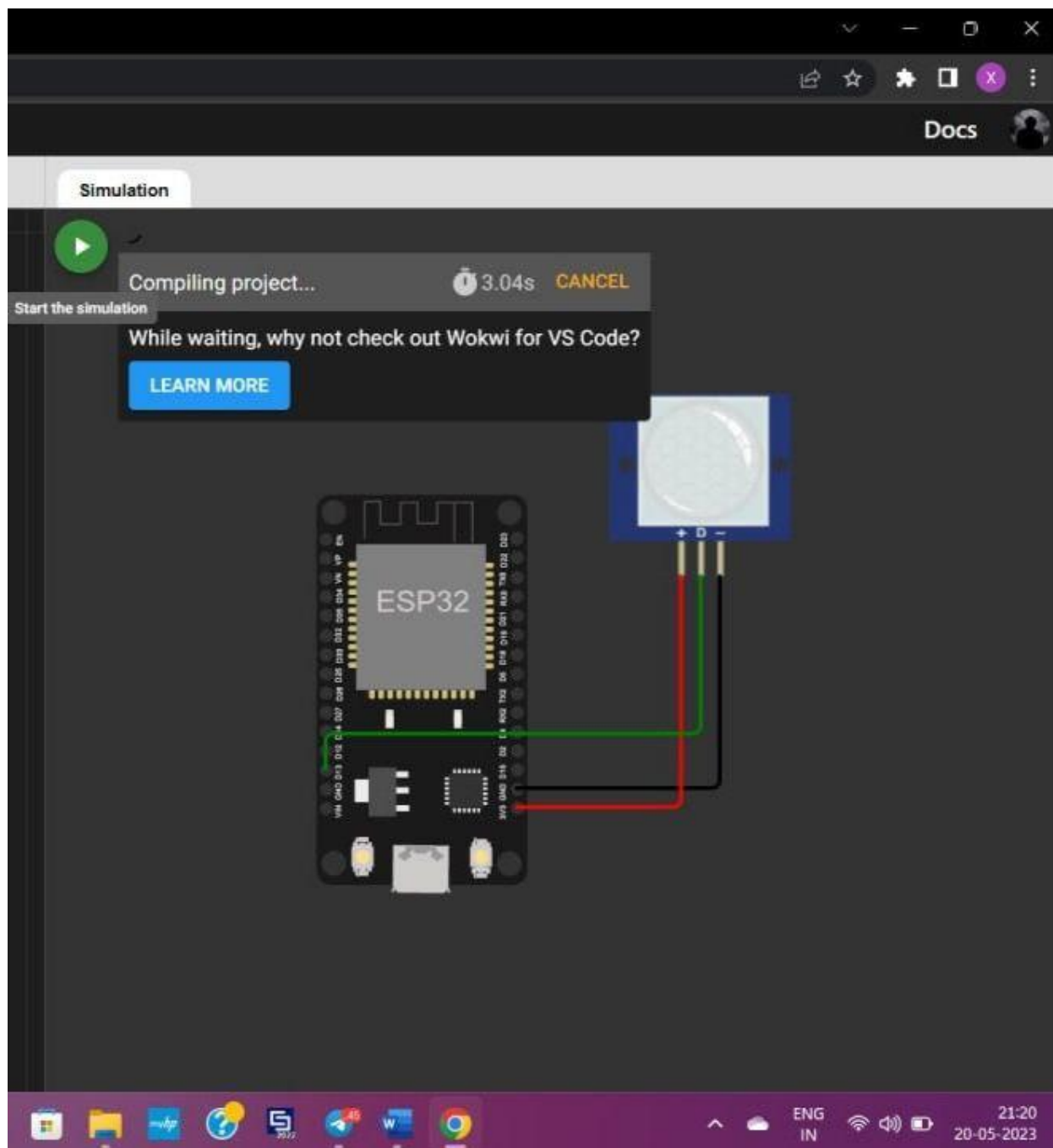
Windows logo

Search

Telegram

Taskbar icon





Simulation

00:06.607

99%

ESP32

10.10.0.2

Reconnecting client to j141ur.messaging.internetofthings.ibmcloud.com

iot-2/cmd/test/fmt/String

subscribe to cmd OK

Liters: 1.00    Bill: \$20.00

Liters: 1.00    Bill: \$20.00

ENG

IN

21:20

20-05-2023