Capacitive Touch Sensor with NodeMCU and Google Sheets Integration

---

Introduction:

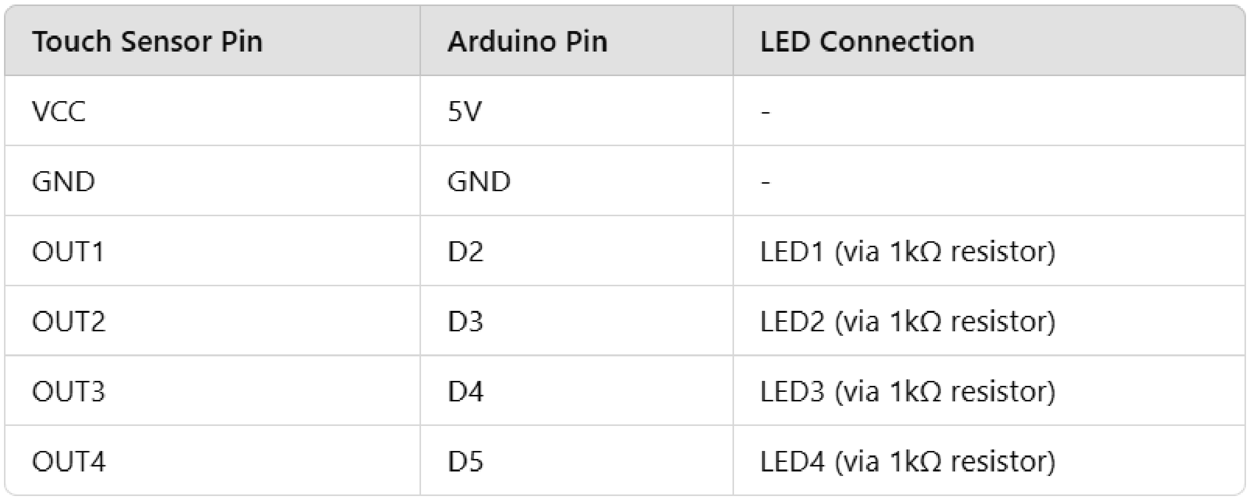
This project demonstrates how to interface a 4-channel capacitive touch sensor with the NodeMCU (ESP8266) and log real-time data into Google Sheets using Google Apps Script. Instead of traditional output components like LEDs, this setup focuses purely on sensing touch and wirelessly transmitting the data for logging and analysis. It's a practical IoT project suitable for learning about sensor integration, Wi-Fi communication, and cloud data visualization.

Components Required:

- NodeMCU ESP8266
- 4-Channel Capacitive Touch Sensor (TTP224 or similar) -1
- Jumper wires(Female to Male) - 4
- LED (White) - 4
- Breadboard (optional)

Circuit Connections:

LEDs: Connect the anode (+, long leg) of each LED to the corresponding output pin via a **1kΩ resistor**.



Touch Sensor Pin	Arduino Pin	LED Connection
VCC	5V	-
GND	GND	-
OUT1	D2	LED1 (via 1kΩ resistor)
OUT2	D3	LED2 (via 1kΩ resistor)
OUT3	D4	LED3 (via 1kΩ resistor)
OUT4	D5	LED4 (via 1kΩ resistor)

Step 2: Upload the Arduino Code

C/C++
```cpp
#define TOUCH1 2  // Touch Sensor Output 1
#define TOUCH2 3  // Touch Sensor Output 2
#define TOUCH3 4  // Touch Sensor Output 3
#define TOUCH4 5  // Touch Sensor Output 4

#define LED1 6    // LED 1
#define LED2 7    // LED 2
#define LED3 8    // LED 3
#define LED4 9    // LED 4

void setup() {
    pinMode(TOUCH1, INPUT);
    pinMode(TOUCH2, INPUT);
    pinMode(TOUCH3, INPUT);
    pinMode(TOUCH4, INPUT);

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
}

void loop() {
    digitalWrite(LED1, digitalRead(TOUCH1));
    digitalWrite(LED2, digitalRead(TOUCH2));
    digitalWrite(LED3, digitalRead(TOUCH3));
    digitalWrite(LED4, digitalRead(TOUCH4));
}
```

Step 3: Testing the Circuit

1. **Upload the code** to the Arduino using the Arduino IDE.
2. **Touch each sensor pad**:
   - If a pad is touched, the corresponding LED should turn ON.
   - If released, the LED should turn OFF.
3. **If all LEDs are blinking randomly**, add **10kΩ pull-down resistors** on OUT pins to stabilize the signal.

Troubleshooting:

Issue: All LEDs are blinking continuously

Solution: Add **10kΩ pull-down resistors** between each OUT pin and GND.

Issue: LEDs are behaving inversely (ON when not touched, OFF when touched)

Solution: Modify the code as follows:

```C/C++
digitalWrite(LED1, !digitalRead(TOUCH1));
```

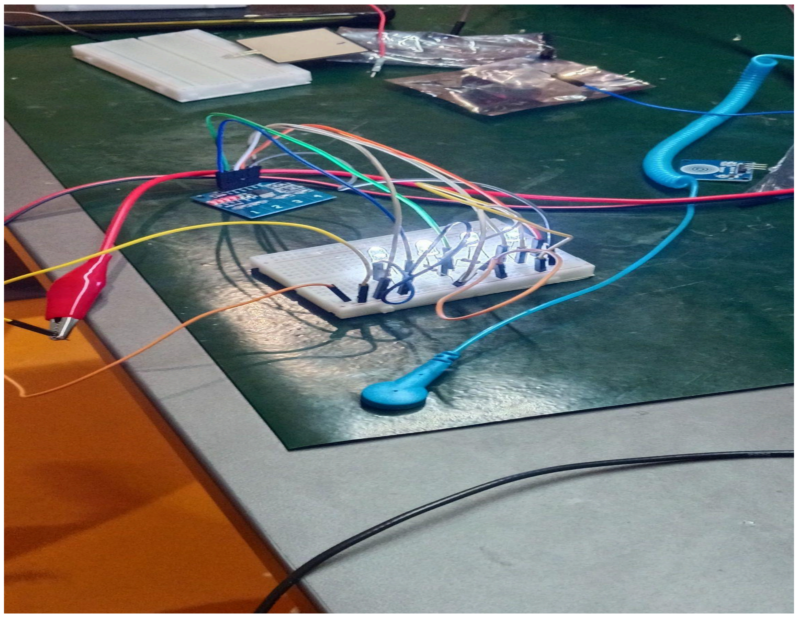
This inverts the logic, making LEDs turn ON only when touched.

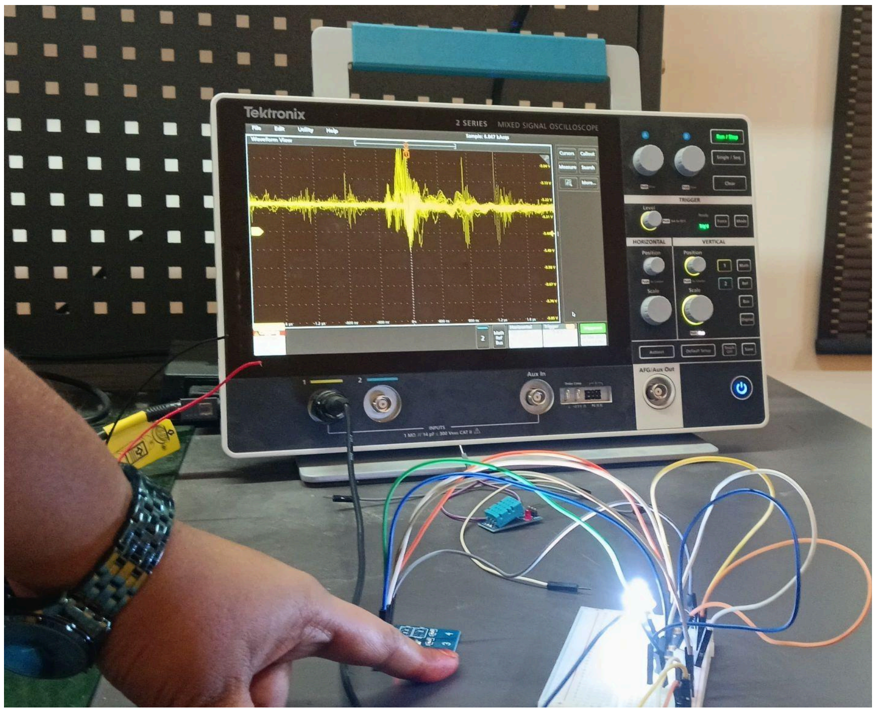
Issue: Some sensors are not responding

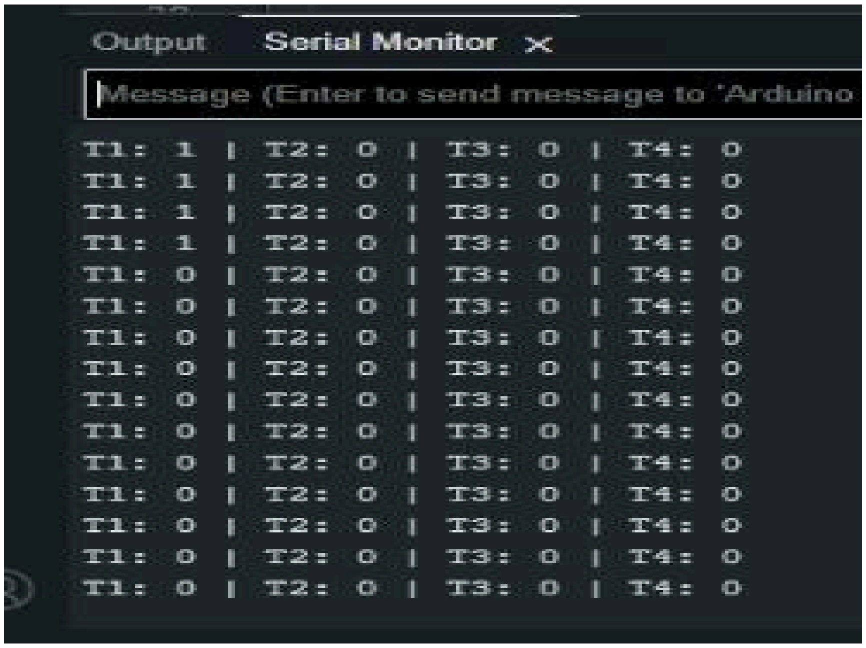
Solution:

- Ensure **proper wiring** and **firm connections**.
- If using **3.3V power instead of 5V**, check if your sensor supports it.
- Keep away from **external electromagnetic interference**.

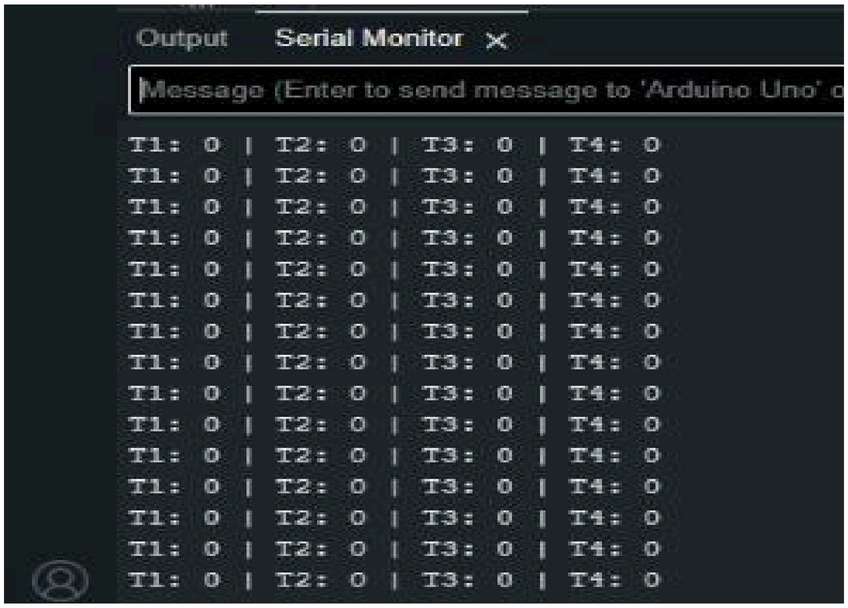
Output:



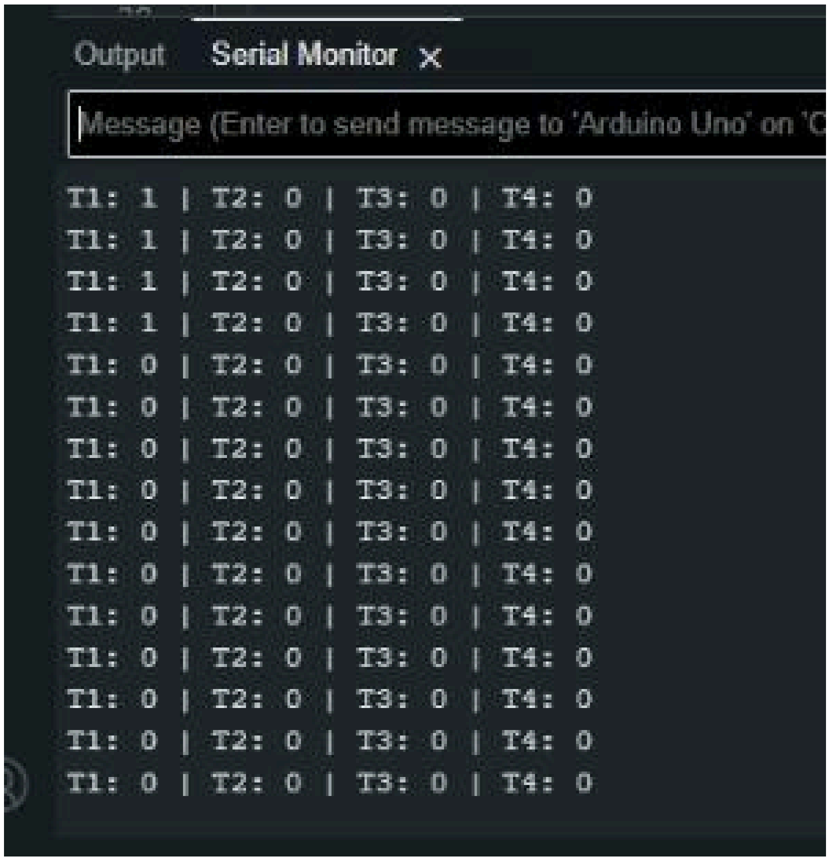




```
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
```



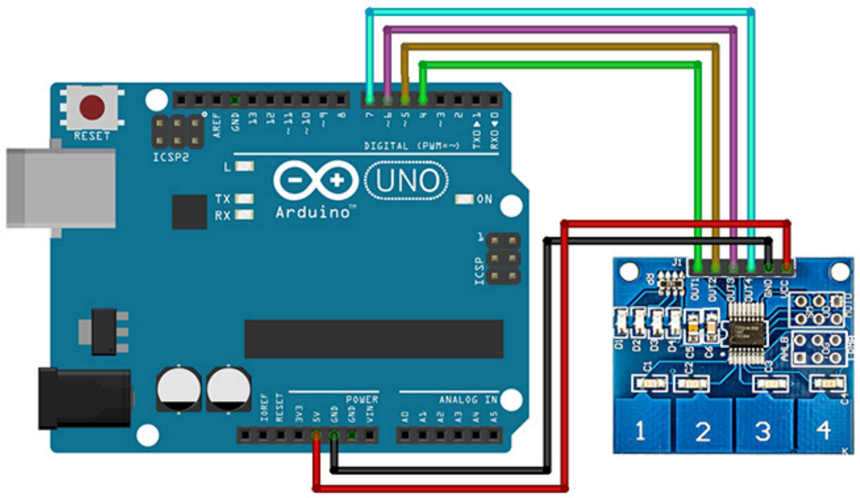
```
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
```



```
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 1 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
T1: 0 | T2: 0 | T3: 0 | T4: 0
```

Wiring Connections:

Touch Sensor Pin	**NodeMCU Pin**
OUT1	D2 (GPIO4)
OUT2	D1 (GPIO5)
OUT3	D6 (GPIO12)
OUT4	D7 (GPIO13)
VCC	3.3V
GND	GND



Step 1: Arduino Code for NodeMCU

```
C/C++
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <time.h>  // For timestamp

const char* ssid = "Redmi 10";        // Your Wi-Fi SSID
const char* password = "Nithi2004";  // Your Wi-Fi password

const char* host = "script.google.com";
const char* fullUrl =
"https://script.google.com/macros/s/AKfycby6y2y-w0Xe4zuS98E48vYSG
ROjZNjkth5mDtfE3Px4xdQeqBXT-r9CDo_6bcu0hB9W/exec";  // Google
Apps Script Web App
```

```
WiFiClientSecure client;

// Touch sensor pins
const int TOUCH_1 = 4;   // D2
const int TOUCH_2 = 5;   // D1
const int TOUCH_3 = 12;  // D6
const int TOUCH_4 = 13;  // D7

// NTP config
const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 19800; // IST = GMT + 5:30 = 19800
seconds
const int daylightOffset_sec = 0;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.print("Connecting");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi connected");
  Serial.println("IP Address: " + WiFi.localIP().toString());

  client.setInsecure();
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

  pinMode(TOUCH_1, INPUT);
  pinMode(TOUCH_2, INPUT);
  pinMode(TOUCH_3, INPUT);
  pinMode(TOUCH_4, INPUT);
}
```

```
String getTimeStamp() {
  time_t now = time(nullptr);
  struct tm* timeinfo = localtime(&now);

  char buffer[30];
  strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M:%S",
timeinfo);
  return String(buffer);
}

void loop() {
  String touch1 = digitalRead(TOUCH_1) == HIGH ? "Touched" :
"NotTouched";
  String touch2 = digitalRead(TOUCH_2) == HIGH ? "Touched" :
"NotTouched";
  String touch3 = digitalRead(TOUCH_3) == HIGH ? "Touched" :
"NotTouched";
  String touch4 = digitalRead(TOUCH_4) == HIGH ? "Touched" :
"NotTouched";

  String timestamp = getTimeStamp();
  Serial.println("Timestamp: " + timestamp);

  String url = String(fullUrl) + "?timestamp=" +
urlencode(timestamp) +
               "&touch1=" + touch1 + "&touch2=" + touch2 +
               "&touch3=" + touch3 + "&touch4=" + touch4;

  Serial.println("Requesting URL: " + url);

  if (client.connect("script.google.com", 443)) {
    Serial.println("Connected to server");

    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                  "Connection: close\r\n\r\n");

    while (client.connected()) {
      String line = client.readStringUntil('\n');
      if (line == "\r") break;
    }

    String response = client.readString();
    Serial.println("Response from server:");
    Serial.println(response);
  } else {
    Serial.println("Connection to server failed");
  }

  delay(5000); // Send every 5 seconds
}

// Simple URL encoder for spaces and colons
String urlencode(String str) {
  str.replace(" ", "%20");
  str.replace(":", "%3A");
  return str;
}
```

Step 2: Google Apps Script (Web App)

JavaScript
```
function doGet(e) {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Sheet1");
  if (!sheet) return ContentService.createTextOutput("Error:
Sheet not found");
```

```
  var timestamp = e.parameter.timestamp || new Date();
  var touch1 = e.parameter.touch1 || "";
  var touch2 = e.parameter.touch2 || "";
  var touch3 = e.parameter.touch3 || "";
  var touch4 = e.parameter.touch4 || "";

  sheet.appendRow([timestamp, touch1, touch2, touch3, touch4]);
  return ContentService.createTextOutput("Success");
}
```

After pasting the code in Apps Script Editor, deploy it as a Web App:

- Go to **Deploy** → **Manage deployments**
- Select **New deployment**, type: Web App
- Set access to **Anyone** and copy the Web App URL

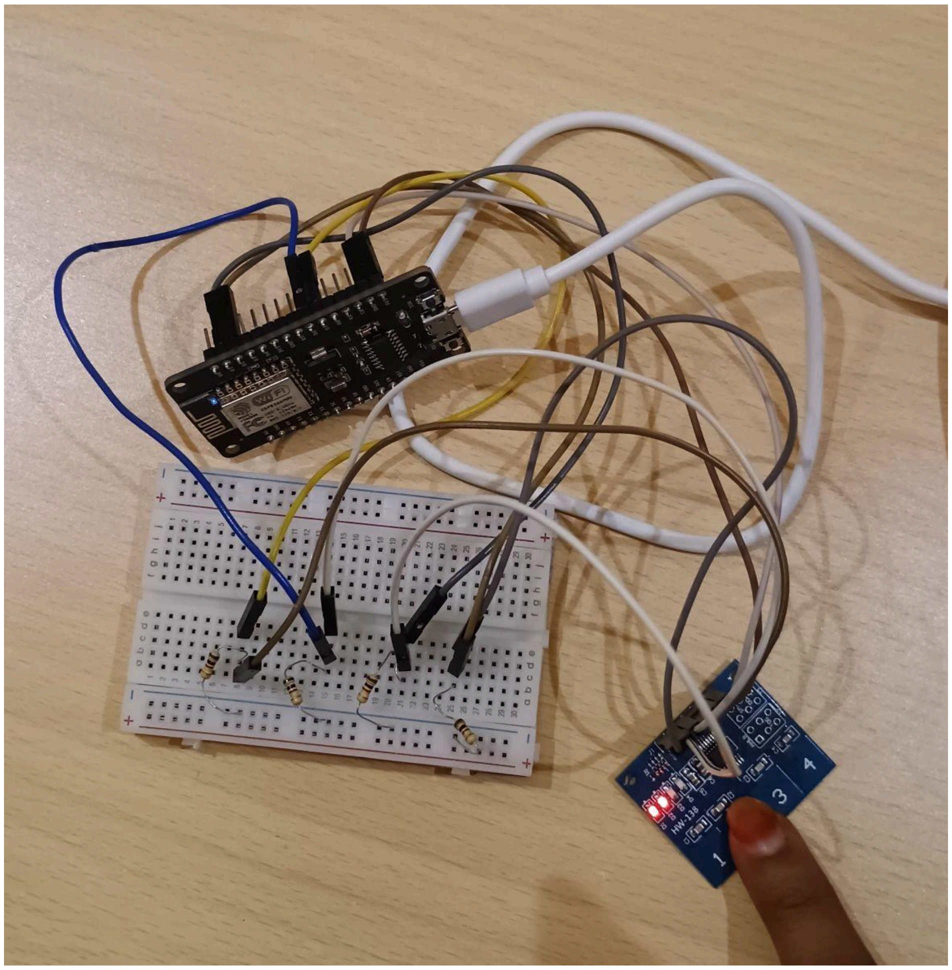
Step 3: View Data in Google Sheets

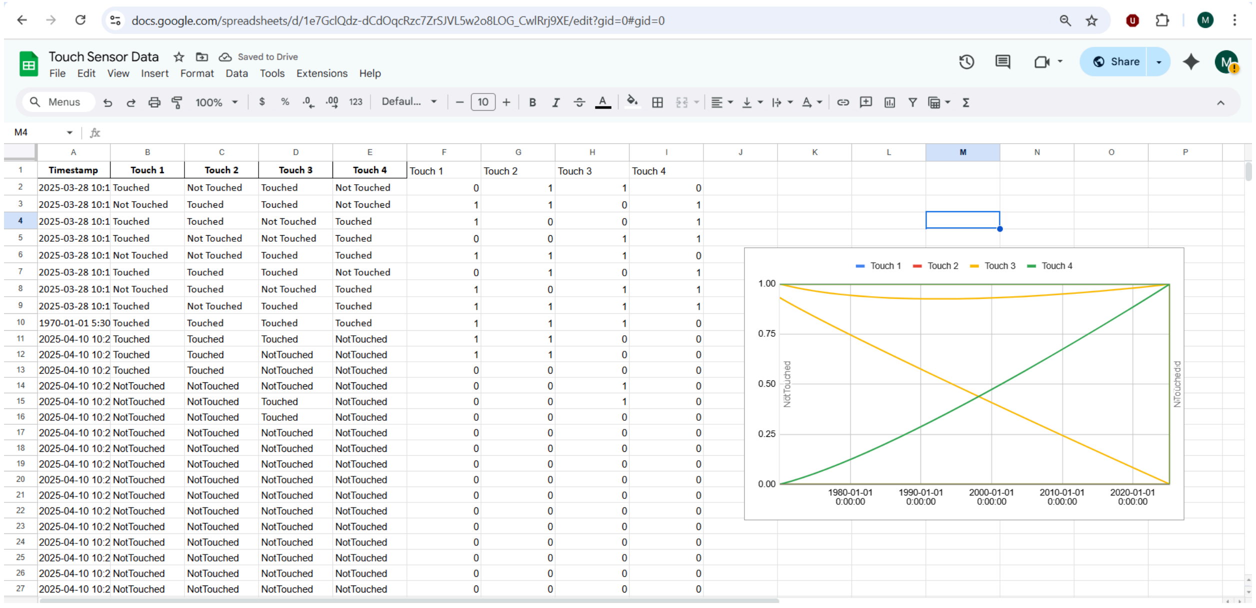
Every 5 seconds, the ESP8266 sends a timestamp and touch status of each pad. Data will automatically populate your Google Sheet.

Creating a Chart in Google Sheets

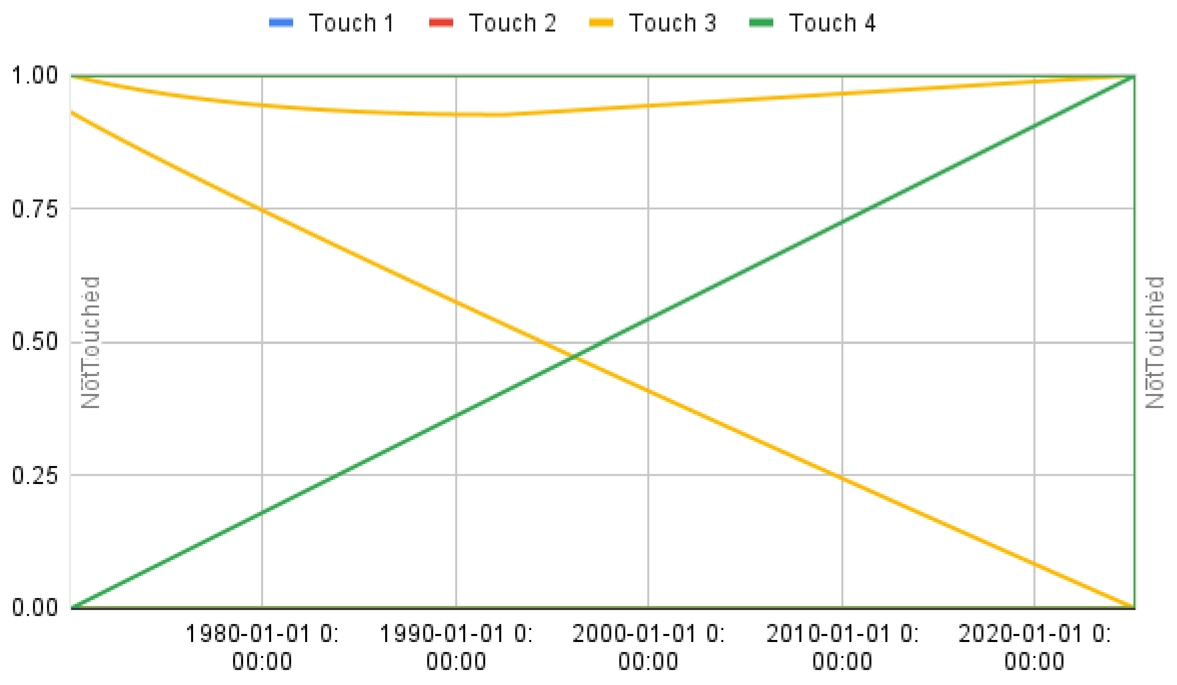
1. Highlight your data (including timestamp and touch statuses)
2. Click **Insert** → **Chart**
3. In the chart editor:
    - Chart type: Choose "Line chart" or "Column chart"
    - Use timestamp as the X-axis
    - Use touch values as series

Output:





	A	B	C	D	E	F	G	H	I
1	**Timestamp**	**Touch 1**	**Touch 2**	**Touch 3**	**Touch 4**	Touch 1	Touch 2	Touch 3	Touch 4
2	2025-03-28 10:1	Touched	Not Touched	Touched	Not Touched	0	1	1	0
3	2025-03-28 10:1	Not Touched	Touched	Touched	Not Touched	1	1	0	1
4	2025-03-28 10:1	Touched	Touched	Not Touched	Touched	1	0	0	1
5	2025-03-28 10:1	Touched	Not Touched	Not Touched	Touched	0	0	1	1
6	2025-03-28 10:1	Not Touched	Not Touched	Touched	Touched	1	1	1	0
7	2025-03-28 10:1	Touched	Touched	Touched	Not Touched	0	1	0	1
8	2025-03-28 10:1	Not Touched	Touched	Not Touched	Touched	1	0	1	1
9	2025-03-28 10:1	Touched	Not Touched	Touched	Touched	1	1	1	1
10	1970-01-01 5:30	Touched	Touched	Touched	Touched	1	1	1	0
11	2025-04-10 10:2	Touched	Touched	Touched	NotTouched	1	1	0	0
12	2025-04-10 10:2	Touched	Touched	NotTouched	NotTouched	1	1	0	0
13	2025-04-10 10:2	Touched	Touched	NotTouched	NotTouched	0	0	0	0
14	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	1	0
15	2025-04-10 10:2	NotTouched	NotTouched	Touched	NotTouched	0	0	1	0
16	2025-04-10 10:2	NotTouched	NotTouched	Touched	NotTouched	0	0	0	0
17	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
18	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
19	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
20	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
21	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
22	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
23	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
24	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
25	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
26	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0
27	2025-04-10 10:2	NotTouched	NotTouched	NotTouched	NotTouched	0	0	0	0



Sheet Link

Video Link

Conclusion:

This setup enables a wireless capacitive touch sensor system using NodeMCU, allowing real-time data logging to Google Sheets with accurate timestamps, all without additional hardware like LEDs or pull-down resistors. This is ideal for monitoring and logging touch-based inputs in various applications.