

Microcontrollers and Sensors

Yatin Vadehra

Touch-Controlled Current Sensor

Project report

## Contents:

1. Abstract
2. Introduction
3. Components Used
4. Theory
5. Procedure
6. Code and Snapshots
7. Results/Outcomes
8. Industrial Applications
9. References

## Abstract

The "Touch-Controlled Current Sensor" project integrates a capacitive touch sensor and a current sensor with a NodeMCU to create a system capable of measuring electrical current based on user interactions. The project aims to demonstrate the practical application of current sensing technology and human interface devices in an IoT (Internet of Things) context. Through this project, users can measure DC or AC current by simply touching the capacitive sensor, with results displayed on a computer via a serial connection.

## Introduction

Current sensors are essential components in electrical and electronic circuits, used to monitor and manage the flow of current. The integration of a capacitive touch sensor introduces an interactive element, allowing users to engage with and control the measurement process. This project combines the ACS712 current sensor, TTP223 capacitive touch sensor, and a NodeMCU microcontroller to create an interactive current measurement system.

## Components Used:

NodeMCU (ESP8266)

ACS712 Current Sensor (20A Model)

TTP223 Capacitive Touch Sensor

Connecting Wires

### Theory

The ACS712 sensor operates on the principle of Hall-effect, which states that a magnetic field surrounding a current-carrying conductor exerts a force on moving charges, inducing a voltage across the conductor. The TTP223 touch sensor works on capacitive sensing, where the presence of a conductive object (like a human finger) alters the capacitance of the sensor's circuit. The NodeMCU reads the output from these sensors and transmits the data to a computer for visualization.

### Procedure:

Connect the components as per the schematic provided.

Load the provided code into the Arduino IDE.

Compile and upload the code to the NodeMCU.

Open the Serial Monitor to view the current measurements and touch sensor status.

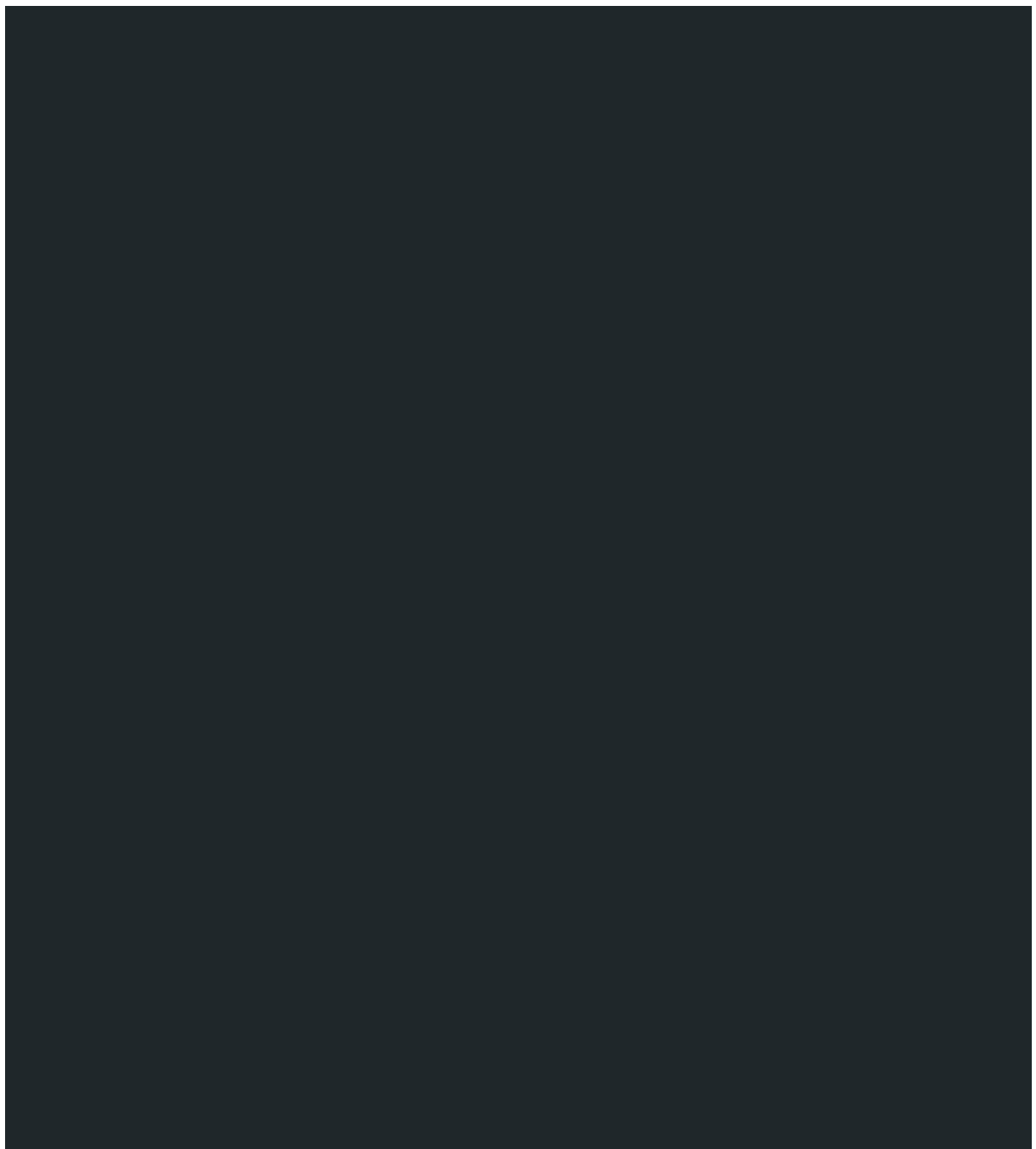
### Code and Snapshots:

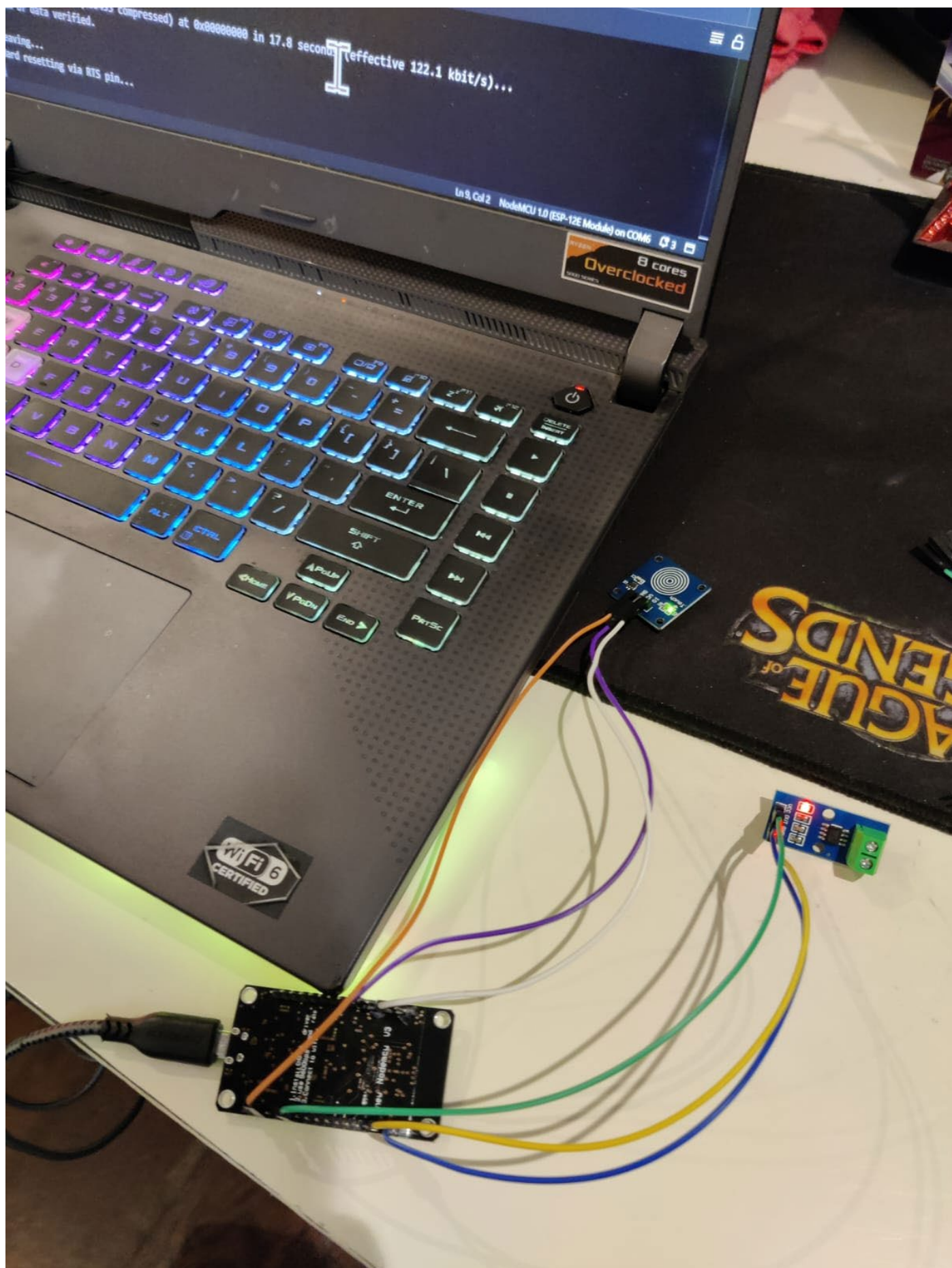
```
#include <Arduino.h>
```

```
const int touchPin = D2;
```

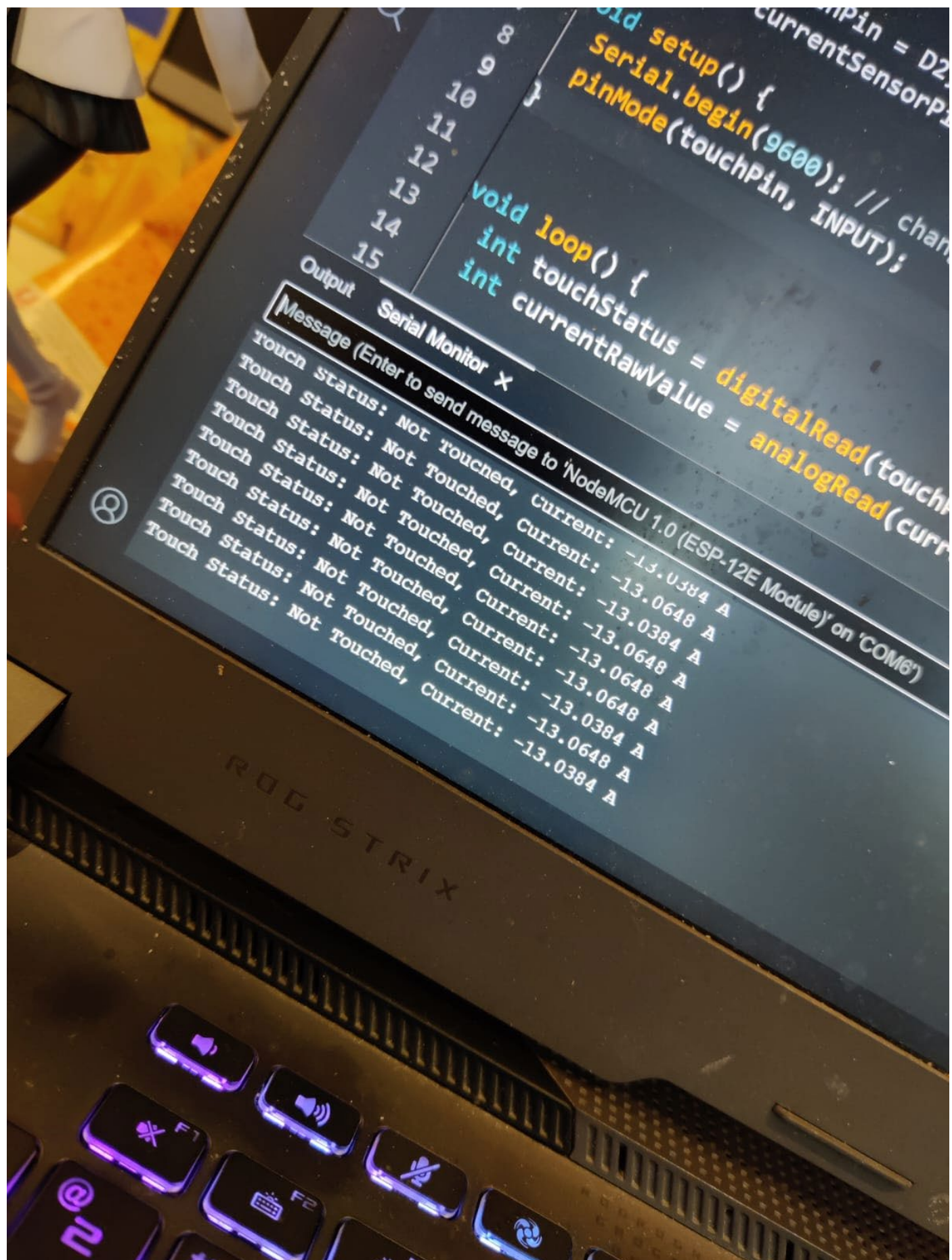
```
const int currentSensorPin = A0;
```

```
void setup() {  
  Serial.begin(115200);  
  pinMode(touchPin, INPUT);  
}  
  
void loop() {  
  int touchStatus = digitalRead(touchPin);  
  int currentRawValue = analogRead(currentSensorPin);  
  
  // Convert raw value to actual current (in Amperes)  
  float current = ((float)currentRawValue / 1024 * 5 - 2.5) / 0.185;  
  
  Serial.print("Touch Status: ");  
  Serial.print(touchStatus == HIGH ? "Touched" : "Not Touched");  
  Serial.print(", Current: ");  
  Serial.print(current, 4);  
  Serial.println(" A");  
  
  delay(500); // Delay to make the output readable  
}
```









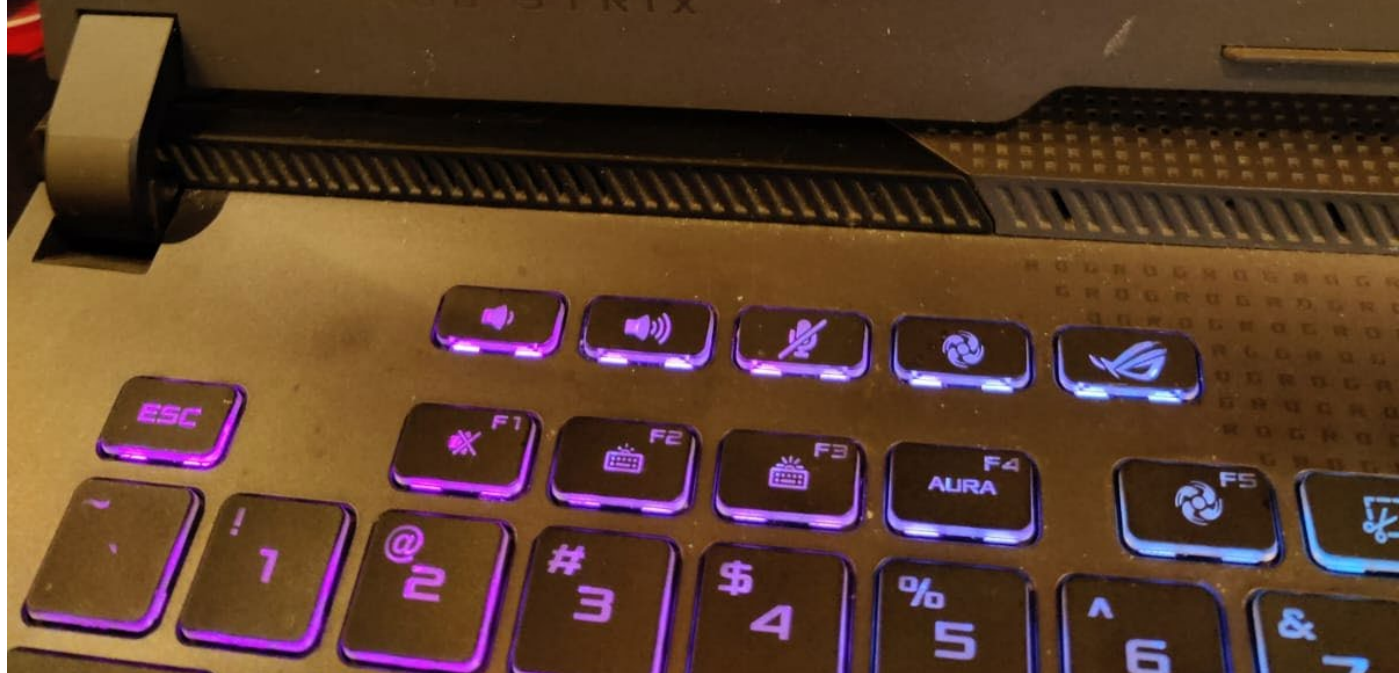


```
5  
6 void setup() {  
7   Serial.begin(9600); // change baud rate to 9600  
8   pinMode(touchPin, INPUT);  
9 }  
10  
11  
12 void loop() {  
13   int touchStatus = digitalRead(touchPin);  
14   int currentRawValue = analogRead(currentSensorPin);  
15 }
```

Output Serial Monitor X

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM6')

Touch Status: Not Touched, Current: -13.0648 A  
Touch Status: Not Touched, Current: -13.0648 A  
Touch Status: Not Touched, Current: -13.0384 A  
Touch Status: Not Touched, Current: -13.0384 A  
Touch Status: Not Touched, Current: -13.0648 A  
Touch Status: Touched, Current: -13.0384 A  
Touch Status: Touched, Current: -13.0384 A  
Touch Status: Touched, Current: -13.0384 A  
Touch Status: Touched, Current: -13.0648 A







## Results/Outcomes:

The system successfully demonstrates the ability to measure current based on user interaction. The Serial Monitor provides real-time feedback, showing the status of the touch sensor and the current measurement.

Such a system could find applications in industrial settings for safety and monitoring, where a non-invasive and interactive method for measuring current could enhance operational efficiency and safety protocols.

## References

1. NodeMCU Documentation: [NodeMCU Documentation](#)
2. ACS712 Datasheet: [ACS712 Datasheet](#)
3. TTP223 Datasheet: [TTP223 Datasheet](#)
4. Arduino IDE Software: [Arduino Software](#)
5. Getting Started with NodeMCU (ESP8266): [NodeMCU Getting Started Guide](#)
6. ACS712 Current Sensor Interfacing Tutorial: [ACS712 Arduino Tutorial](#)
7. Capacitive Touch Sensor Interfacing with Arduino: [Capacitive Touch Arduino Tutorial](#)



