Naman Srivastava
22BPS1101

CCPS EX. NO. 11
Controlling an LED remotely using ESP8266 and Blynk App

Aim:
To remotely control an LED using an ESP8266 (NodeMCU) via the Blynk mobile app and Blynk Cloud.

Requirements:
ESP8266 NodeMCU, LED, 220Ω resistor, breadboard, jumper wires, smartphone with Blynk App, Wi-Fi, Arduino IDE, Blynk library, and Blynk Cloud account.

Procedure:
1. Create a Blynk project: choose device ESP8266, connection Wi-Fi, add a Button widget mapped to digital pin D2, copy Auth Token.

2. Build circuit: LED anode → 220Ω resistor → D2; LED cathode → GND.

3. Configure Arduino IDE: install ESP8266 and Blynk libraries, set BLYNK_AUTH_TOKEN, ssid, and pass in sketch.

4. Code (setup Serial.begin(9600), Blynk.begin(auth, ssid, pass), pinMode(D2, OUTPUT); call Blynk.run() in loop).

5. Upload sketch to NodeMCU, open Serial Monitor, ensure Wi-Fi and Blynk Cloud connection.

6. Test: press the Blynk app button to toggle LED ON/OFF and observe Serial Monitor messages.

Source code:

Arduino:

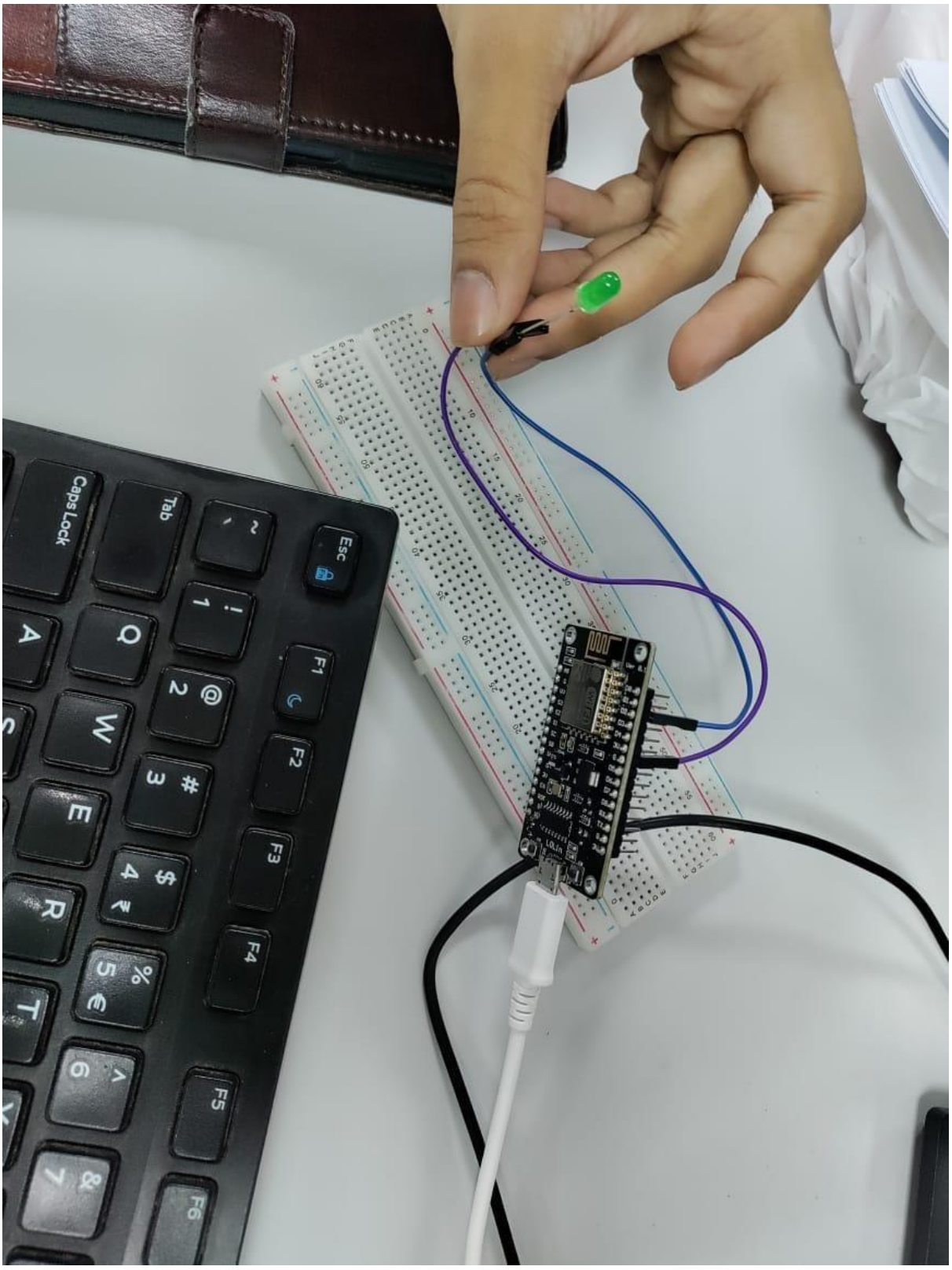
```
#define BLYNK_TEMPLATE_ID "YourTemplateID"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "YourWiFiName";
char pass[] = "YourWiFiPassword";

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(D2, OUTPUT);
}

void loop()
{
```

22BPS1101

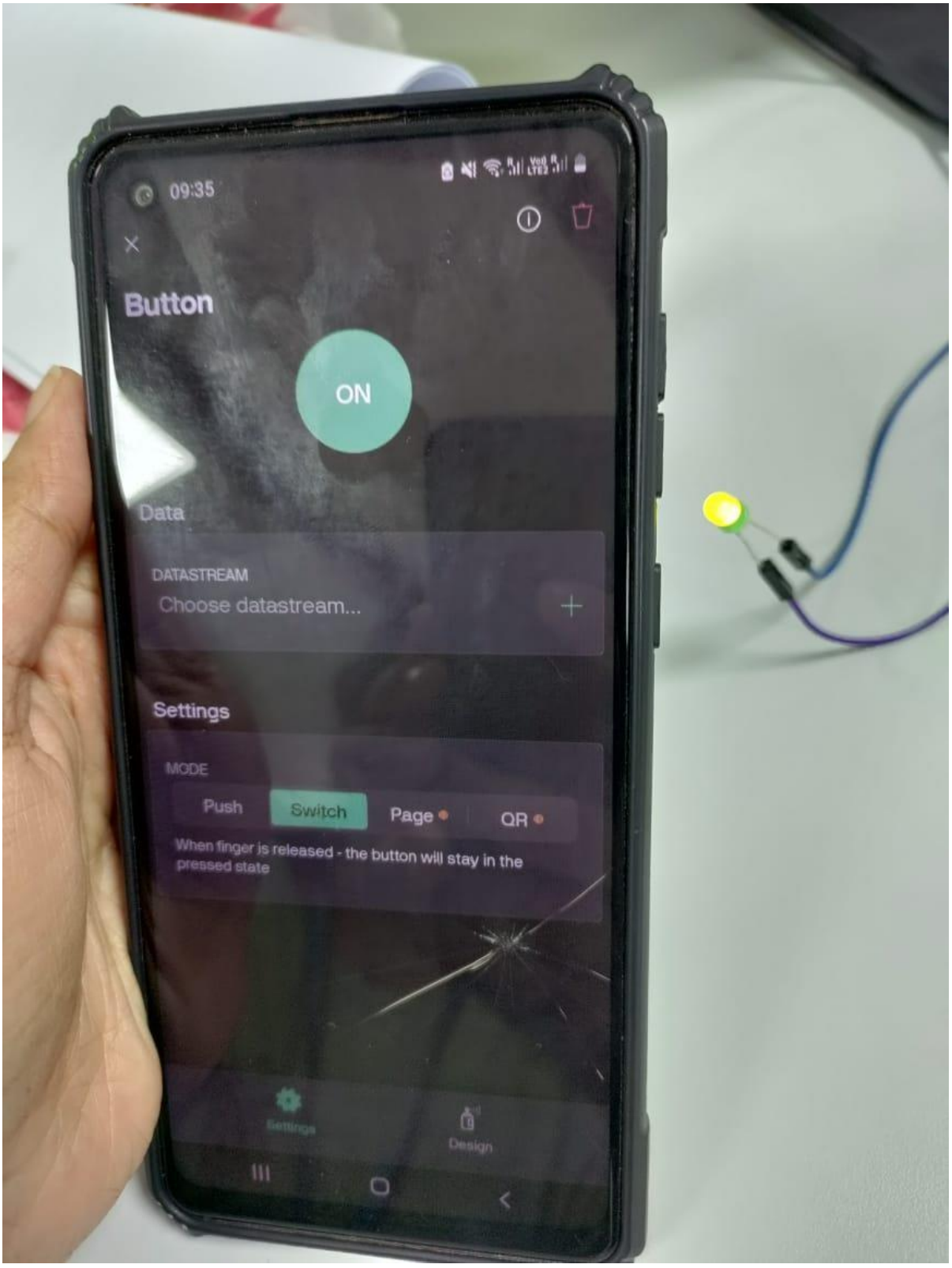
```
  Blynk.run();
}
```

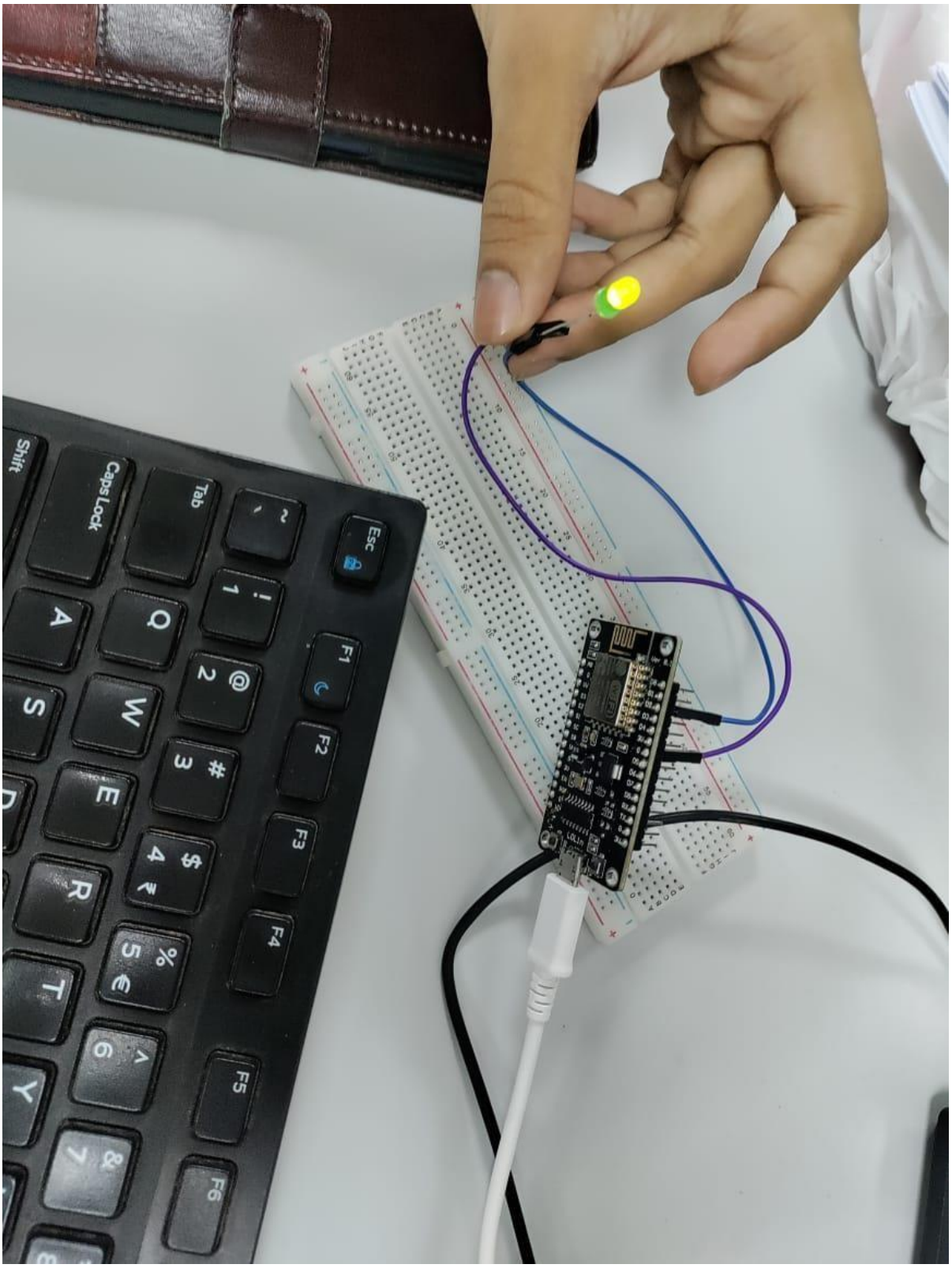
Output:

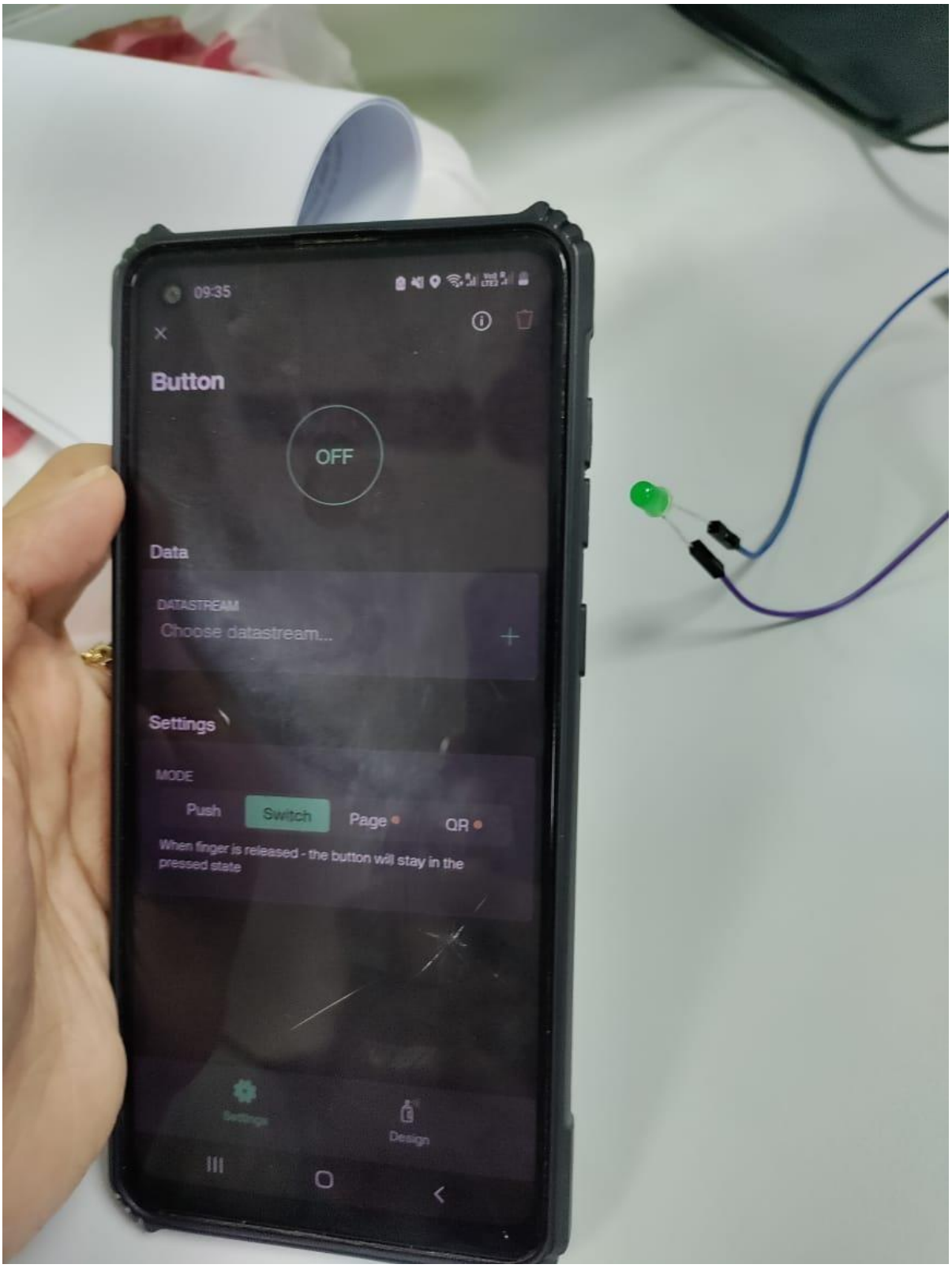
Circuit:

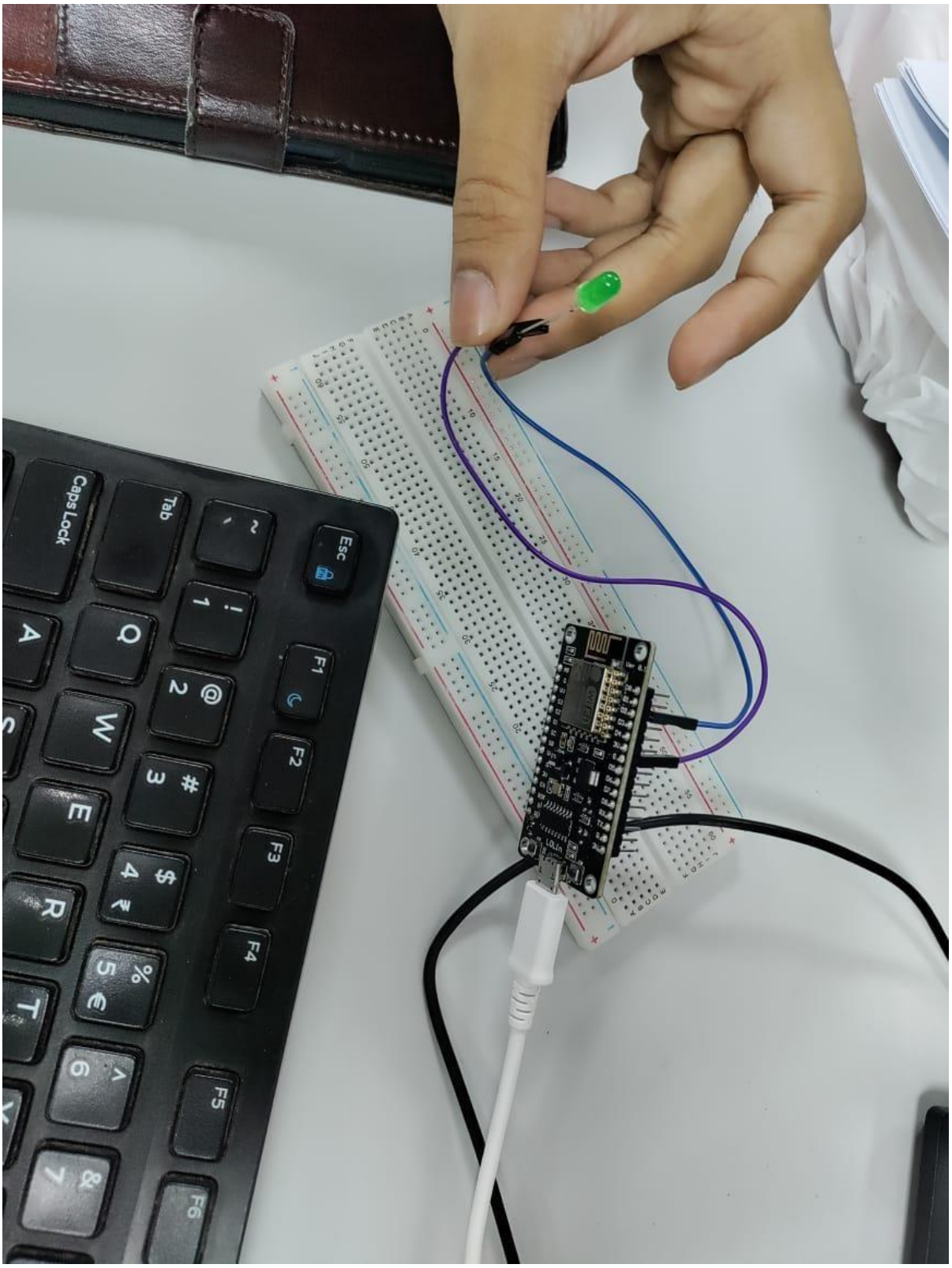


Blynk App:-


09:35
Button
ON
Data
DATASTREAM
Choose datastream...
Settings
MODE
Push
Switch
Page
QR
When finger is released - the button will stay in the pressed state
Settings
Design


Shift
Caps Lock
Tab
Esc
F1
F2
F3
F4
F5
F6


09:35
Button
OFF
Data
DATASTREAM
Choose datastream...
Settings
MODE
Push
Switch
Page
QR
When finger is released - the button will stay in the pressed state
Settings
Design



Result:
LED toggles ON/OFF from the Blynk app via ESP8266 over Wi-Fi; Serial Monitor confirms connection to Blynk Cloud and LED state changes.