LED CONTROL USING NODEMCU AND FIREBASE

CPE 331A - CPE22S1
GROUP 5
Bautista, Francisco
Cantero, Isaiah Christian
Villas, Kaith Angel

OVERVIEW

This project illustrates the implementation of a NodeMCU ESP8266 microcontroller for switching three different LEDs using a Firebase Realtime Database. Through the utilization of Wi-Fi connectivity and cloud control, the system facilitates real-time switching of LEDs from the data stored and uploaded in the database. This arrangement simulates the manner in which systems with Internet of Things (IoT) capability are able to communicate and control hardware devices remotely through cloud services.

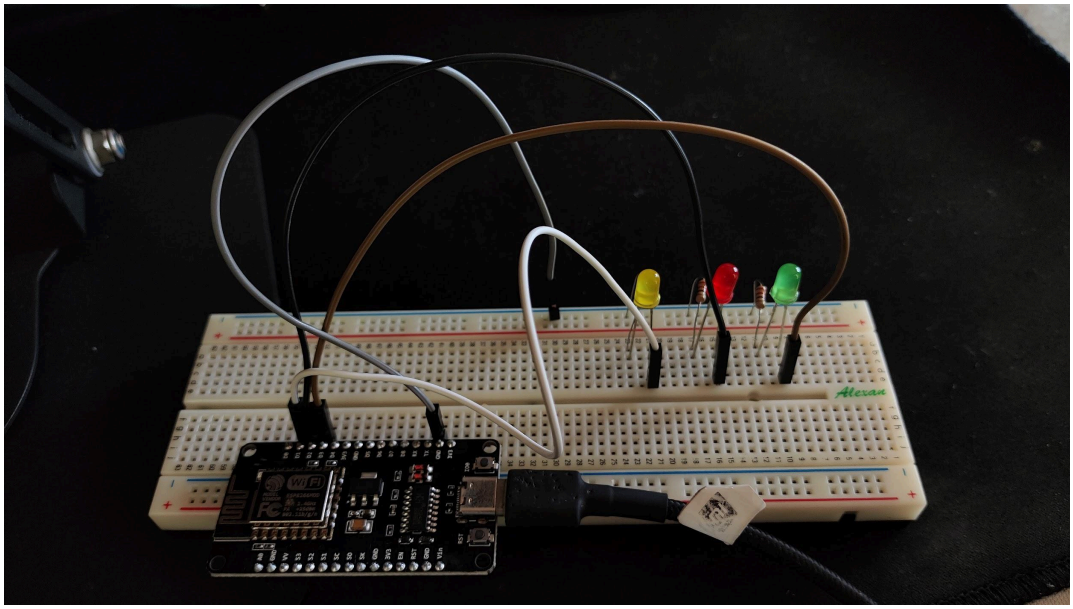
OBJECTIVE

To use the Firebase Realtime Database to control and manipulate three LEDs connected to a NodeMCU ESP8266. The goal is to create a digital switch for the 3 LEDs that can be controlled by anyone with access to the database.

MATERIALS

NodeMCU ESP8266
3 LEDs
3 Resistors (220Ω)
Breadboard and jumper wires
Firebase account
Arduino IDE
Internet connection (Wi-Fi)

BREADBOARD SETUP



FIREBASE SETUP

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app/

- LED1: false
- LED2: false
- LED3: false

```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

SERIAL MONITOR

```
LED1: 0
LED2: 0
LED3: 0
```

SOURCE CODE

nodemculedfirebase.ino

```ino
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>

// Token generation and RTDB helper functions (provided by the library addons)
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// Wi-Fi credentials (as provided)
const char* WIFI_SSID = "cow";
const char* WIFI_PASSWORD = "444444444";

// Firebase project details (as provided)
#define FIREBASE_HOST "https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app/"
#define FIREBASE_API_KEY "AIzaSyAeEArxk7wO4yvL_q_8NDD2bKkSsNAB4cw"

FirebaseData firebaseData;
FirebaseConfig firebaseConfig;
FirebaseAuth firebaseAuth;

bool signupOK = false;
const unsigned long sendInterval = 3000;  // 3-second interval between reads

// Connects to Wi-Fi with a timeout of 20 seconds
void connectWiFi() {
  Serial.print("Connecting to Wi-Fi");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  unsigned long startAttemptTime = millis();

  while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime < 20000) {
    Serial.print(".");
    delay(1000);
  }
```

nodemculedfirebase.ino

```ino
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWi-Fi connected!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nWi-Fi connection failed. Restarting...");
    ESP.restart();
  }
}

void setup() {
  Serial.begin(115200);

  // Set LED pins as outputs using NodeMCU pin labels
  pinMode(D1, OUTPUT);  // D1 typically maps to GPIO5
  pinMode(D2, OUTPUT);  // D2 typically maps to GPIO4
  pinMode(D3, OUTPUT);  // D3 typically maps to GPIO0

  // Connect to Wi-Fi
  connectWiFi();

  // Configure Firebase with your project details
  firebaseConfig.database_url = FIREBASE_HOST;
  firebaseConfig.api_key = FIREBASE_API_KEY;

  // Sign up for Firebase authentication (anonymous)
  if (Firebase.signUp(&firebaseConfig, &firebaseAuth, "", "")) {
    Serial.println("Firebase Sign-up successful!");
    signupOK = true;
  } else {
    Serial.printf("Firebase Sign-up failed: %s\n", firebaseConfig.signer.signupError.message.c_str());
    delay(10000);
    ESP.restart(); // Restart if Firebase authentication fails
  }
```

nodemculedfirebase.ino

```ino
  // Set a callback to handle token status changes (for debugging)
  firebaseConfig.token_status_callback = tokenStatusCallback;

  // Initialize Firebase with the configuration and authentication details
  Firebase.begin(&firebaseConfig, &firebaseAuth);
  Firebase.reconnectWiFi(true);
}

void loop() {
  if (!signupOK) return;  // Ensure we've successfully signed in

  delay(sendInterval);  // Wait before reading again

  // Variables to hold LED states from Firebase
  bool LED1, LED2, LED3;

  // Read the state of LED1 from path "/LED1"
  if (Firebase.getBool(firebaseData, "/LED1", &LED1)) {
    Serial.print("LED1: ");
    Serial.println(LED1);
  } else {
    Serial.print("Failed to read LED1: ");
    Serial.println(firebaseData.errorReason());
  }

  // Read the state of LED2 from path "/LED2"
  if (Firebase.getBool(firebaseData, "/LED2", &LED2)) {
    Serial.print("LED2: ");
    Serial.println(LED2);
  } else {
    Serial.print("Failed to read LED2: ");
    Serial.println(firebaseData.errorReason());
  }
```

```ino
  // Read the state of LED3 from path "/LED3"
  if (Firebase.getBool(firebaseData, "/LED3", &LED3)) {
    Serial.print("LED3: ");
    Serial.println(LED3);
  } else {
    Serial.print("Failed to read LED3: ");
    Serial.println(firebaseData.errorReason());
  }

  // Control the LEDs based on Firebase data:
  // If a value is true, the corresponding LED turns ON; if false, it turns OFF.
  digitalWrite(D1, LED1 ? HIGH : LOW);
  digitalWrite(D2, LED2 ? HIGH : LOW);
  digitalWrite(D3, LED3 ? HIGH : LOW);
}
```

HOW IT WORKS

The three LEDs are wired to the NodeMCU that is connected to a WiFi network. A Firebase is set up to anonymously fetch and change the boolean values of the LEDs. The process repeats every 3 seconds for real-time updates. Using this, a user with a connection to the same WiFi and has access to the Firebase can remotely switch the LEDs on or off using their own device. This

TASK DONE BY EACH MEMBER

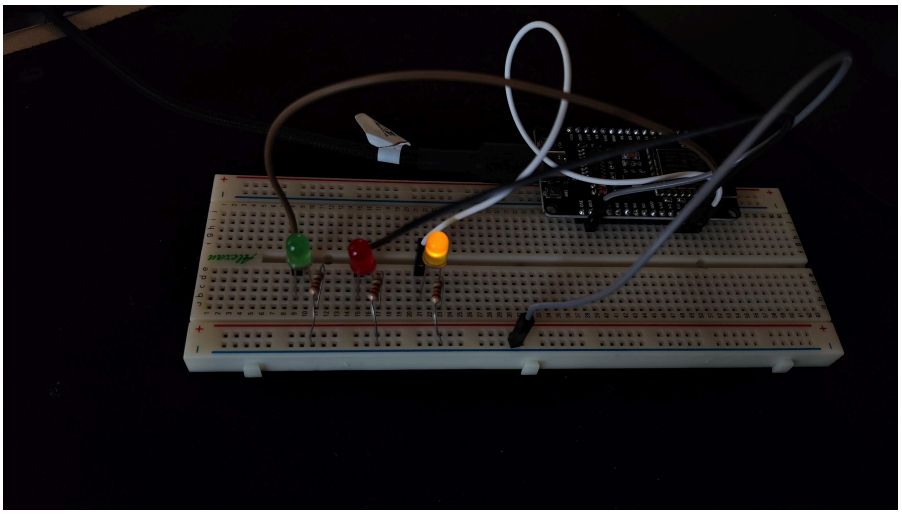
Villas, Kaith Angel (Programmer) -
Circuit Design & Assembly
Firebase Project Setup & Configuration
Arduino Code Development (with the help of Joshua Rom Pasia)
Testing and Debugging
Documentation Preparation

Bautista III, Francisco A. (Hardware)-
Documentation Contributor
Conclusion

OUTPUT

1. LED1 is on while LED2 and LED3 are off.

   BREADBOARD SETUP



FIREBASE SETUP

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app/

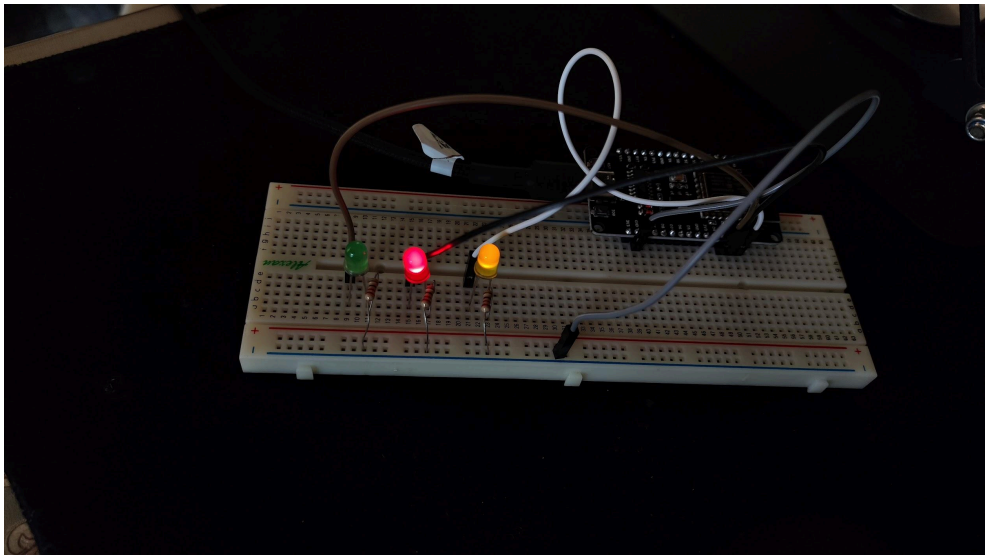
- LED1: true
- LED2: false
- LED3: false

SERIAL MONITOR

```
LED1: 1
LED2: 0
LED3: 0
```

2. LED1 and LED2 are one while LED3 is off.

BREADBOARD SETUP



FIREBASE SETUP

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app

https://nodemculed-5133e-default-rtdb.asia-southeast1.firebasedatabase.app/

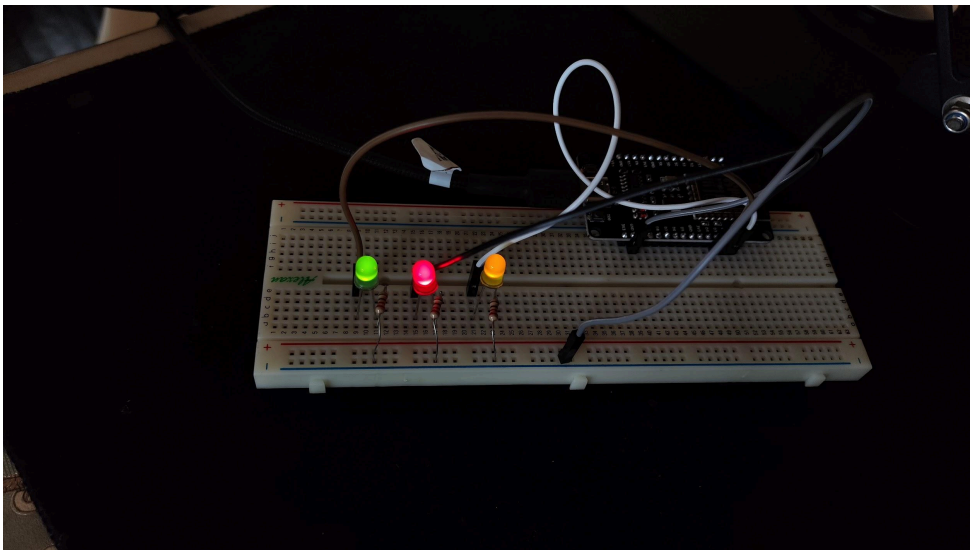
- LED1: true
- LED2: true
- LED3: false

SERIAL MONITOR

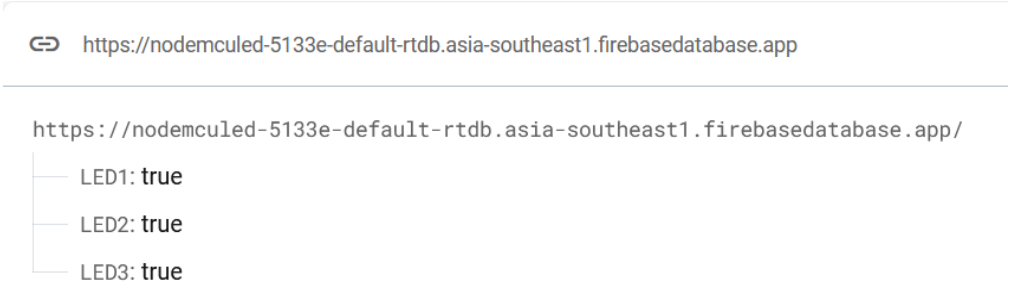
```
LED1: 1
LED2: 1
LED3: 0
```

3. LED1, LED2 and LED3 are on.

BREADBOARD SETUP



FIREBASE SETUP



SERIAL MONITOR

```
LED1: 1
LED2: 1
LED3: 1
```

CONCLUSION

The NodeMCU ESP8266 has IoT capabilities. Using cloud systems and a network connection, we can remotely manage, access, and manipulate devices and hardware wirelessly. This paves the way for other projects for more scalable and efficient automation systems.