

```

#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <RH_NRF24.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

const char* ssid = "SANDYBOY";
const char* password = "SANDYBOY";
#define M D0

String sch;
char Sch[10];
int mois = 105;
String ups;
String up1;
String up2;
String th1 = "000000";
int hrs, mins;

const long utcOffsetInSeconds = 18000;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);
WiFiClient client;
RH_NRF24 nrf24(2, 4);

void setup() {
    Serial.begin(9600);
    Serial.println("Receiver Started");

    pinMode(M, OUTPUT);
    WiFi.begin(ssid, password);
    Serial.print("Connecting to WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected");
    Serial.println(WiFi.localIP());

    timeClient.begin();
    nrf24.init();
    nrf24.setChannel(3);
    nrf24.setRF(RH_NRF24::DataRate2Mbps, RH_NRF24::TransmitPower0dBm);
}

void loop() {
    recv();
    updatetime();
}

```

```

        getsch() ;
    }

void recv() {
    Serial.println("Waiting to receive... ");
    if (nrf24.available()) {
        uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        if (nrf24.recv(buf, &len)) {
            nrf24.send((uint8_t *)"Data Received.", 14);
            nrf24.waitPacketSent();

            int node = buf[0];
            Serial.print("Node: ");
            Serial.println(node);

            if (node != 2) {
                int sm = buf[1];
                float t = buf[2];
                int h = buf[3];
                int br = buf[4];
                mois = sm;

                up1 = String(sm) + ";" + String(t);
                up2 = String(br) + ";" + String(h);
                uploadData(8, up1);
                uploadData(9, up2);

                Serial.println(up1);
                Serial.println(up2);
            } else {
                // Skip Node 2 processing or retain as needed
            }
        }
    } else {
        Serial.println("No New Message");
    }
    delay(2000);
}

void updatetime() {
    timeClient.update();
    int h5 = timeClient.getHours();
    int m5 = timeClient.getMinutes();
    int sec = timeClient.getSeconds();

    if (m5 == 30) {
        mins = 0;
        hrs = h5 + 1;
    }
}

```

```

} else if (m5 > 30) {
    mins = m5 - 30;
    hrs = h5 + 1;
} else {
    mins = m5 + 30;
    hrs = h5;
}

Serial.printf("Time: %02d:%02d:%02d\n", hrs, mins, sec);
}

void getsch() {
    WiFiClient client;
    if (!client.connect("phoenixio.atspace.cc", 80)) {
        Serial.println("Connection failed (getsch)");
        return;
    }

    client.println("GET /api/led/read_all.php?id=10 HTTP/1.0");
    client.println("Host: phoenixio.atspace.cc");
    client.println("Connection: close");
    client.println();

    while (client.connected() && !client.available()) delay(1);
    if (!client.find("\r\n\r\n")) {
        Serial.println("Invalid response");
        return;
    }

    DynamicJsonDocument doc(192);
    DeserializationError error = deserializeJson(doc, client);
    if (error) {
        Serial.println("JSON parse error");
        return;
    }

    sch = doc["led"][0]["status"].as<String>();
    Serial.println("Schedule string: " + sch);
    sch.toCharArray(Sch, 11);

    motor(Sch[0]);
    if (Sch[1] == '1') Schedule();

    if (Sch[9] == '1') {
        int th = (Sch[7] - '0') * 10 + (Sch[8] - '0');
        if (mois < th) {
            ups = "1" + sch.substring(1, 9) + "0";
            uploadData(10, ups);
            motor(Sch[2]);
        }
    }
}

```

```

        }
    }
}

void Schedule() {
    int hr = (Sch[3] - '0') * 10 + (Sch[4] - '0');
    int mi = (Sch[5] - '0') * 10 + (Sch[6] - '0');
    int change = Sch[2] - '0';

    if (hrs == hr && mins == mi) {
        ups = String(change) + "0" + sch.substring(2, 11);
        uploadData(10, ups);
        motor(Sch[2]);
    }
}

void motor(char state) {
    digitalWrite(M, state == '0' ? HIGH : LOW);
}

void uploadData(int id, String data) {
    if (!client.connect("phoenixio.atspace.cc", 80)) {
        Serial.println("Upload failed: can't connect");
        return;
    }

    String url = "GET /api/led/update.php?id=" + String(id) + "&status=" + data + "
HTTP/1.0";
    client.println(url);
    client.println("Host: phoenixio.atspace.cc");
    client.println("Connection: close");
    client.println();

    while (client.connected() && !client.available()) delay(1);
    if (!client.find("\r\n\r\n")) {
        Serial.println("Upload failed: invalid response");
    }
    client.stop();
}

```