```
//ArduinoSide.ino

#include <ArduinoJson.h>
#include <SoftwareSerial.h>

unsigned long time = 0;
const int PIN_FRESH     = 2;
const int PIN_GOING_BAD = 3;
const int PIN_EXPIRED   = 4;
const int statusPins[] = {PIN_FRESH, PIN_GOING_BAD, PIN_EXPIRED};
const int NUM_STATUS_PINS = 3;

const int ALCOHOL_SENSOR_PIN = A0;
const int CH3_SENSOR_PIN     = A1;

const int THRESHOLD_BAD         = 160;
const int THRESHOLD_EXPIRED     = 500;
const int CH3_THRESHOLD_BAD     = 400;
const int CH3_THRESHOLD_EXPIRED = 600;
const int SENSOR_READ_DELAY     = 100;

SoftwareSerial NodeMCU(5, 6);

int currentFoodStatus = 0;

int determineFoodStatus() {
  int alcoholReading = analogRead(ALCOHOL_SENSOR_PIN);
  int ch3Reading = analogRead(CH3_SENSOR_PIN);

  Serial.print("Sensor Readings -> Alcohol (A0): ");
  Serial.print(alcoholReading);
  Serial.print(" | CH3 (A1): ");
  Serial.println(ch3Reading);

  int status = 0;

  if (alcoholReading >= THRESHOLD_EXPIRED || ch3Reading >= CH3_THRESHOLD_EXPIRED) {
    status = 2;
  }
  else if (alcoholReading >= THRESHOLD_BAD || ch3Reading >= CH3_THRESHOLD_BAD) {
    status = 1;
  }

  return status;
}

void turnOffStatusLeds() {
  for (int i = 0; i < NUM_STATUS_PINS; i++) {
    digitalWrite(statusPins[i], LOW);
  }
}

void updateStatusDisplay() {
  turnOffStatusLeds(); // Start clean

  switch (currentFoodStatus) {
    case 0: // FRESH (Green)
      digitalWrite(PIN_FRESH, HIGH);
      Serial.println("STATUS: FRESH");
      break;
    case 1: // GOING BAD (Blue)
      digitalWrite(PIN_GOING_BAD, HIGH);
      Serial.println("STATUS: GOING BAD (Warning)");
      break;
    case 2: // EXPIRED (Red)
      digitalWrite(PIN_EXPIRED, HIGH);
      Serial.println("STATUS: EXPIRED (Danger)");
      break;
    default:
      // Handle error state (e.g., all pins blink briefly)
      digitalWrite(PIN_EXPIRED, HIGH);
      delay(50);
      digitalWrite(PIN_EXPIRED, LOW);
      break;
  }
}

void setup()
{
  Serial.begin(9600);

  NodeMCU.begin(9600);

  const int dell = 5000;
  digitalWrite(PIN_FRESH, HIGH);
  digitalWrite(PIN_GOING_BAD, HIGH);
  digitalWrite(PIN_EXPIRED, HIGH);
  delay(dell);
  digitalWrite(PIN_FRESH, LOW);
  digitalWrite(PIN_EXPIRED, LOW);
  digitalWrite(PIN_GOING_BAD, LOW);

  Serial.println("Food Status Indicator Initialized.");

  // Set Status Pins as OUTPUTs
  for (int i = 0; i < NUM_STATUS_PINS; i++) {
    pinMode(statusPins[i], OUTPUT);
  }

  // Set Analog Pins as INPUTs (Default, but good practice)
  pinMode(ALCOHOL_SENSOR_PIN, INPUT);
  pinMode(CH3_SENSOR_PIN, INPUT);

  // Initial status check
  currentFoodStatus = determineFoodStatus();
  updateStatusDisplay();
}

void loop()
{
  // Read the current status from ze sensors
  currentFoodStatus = determineFoodStatus();

  updateStatusDisplay();

  //Sending Data to ze NodeMCU
  if (millis() - time >= 15000)
  {
    JsonDocument JsonBuff;
    JsonObject Data = JsonBuff.add<JsonObject>();
    time = millis();
    Data["A1_Stats"] = currentFoodStatus;
    if(serializeJson(Data, NodeMCU))
    {
      Serial.println("Data being written to NodeMCU");
      Serial.println("Json Buffer:\n");
      serializeJsonPretty(Data, Serial);
      Serial.println("\n");
      JsonBuff.clear();
    }
  }

  delay(SENSOR_READ_DELAY);
}
```