// Node1\_ESP8266\_Sensor\_Transmitter.ino

#include <DHT.h>

#include <SoftwareSerial.h>

// DHT11配置

#define DHTPIN D4        // DHT11数据引脚 (GPIO2)

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// MQ-2配置

#define MQ2\_AO A0        // MQ-2模拟输出

#define MQ2\_DO D5        // MQ-2数字输出 (GPIO14)

// Zigbee模块配置 (DL-20)

#define ZIGBEE\_RX D6     // GPIO12 - 连接到Zigbee的TX

#define ZIGBEE\_TX D7     // GPIO13 - 连接到Zigbee的RX

SoftwareSerial zigbeeSerial(ZIGBEE\_RX, ZIGBEE\_TX);

// 发送计数器

unsigned long sendCount = 0;

void setup() {

  // 主串口用于调试

  Serial.begin(115200);

  delay(1000);

  Serial.println("\n=== Node1 ESP8266 发送节点启动 ===");

  // 初始化Zigbee串口

  zigbeeSerial.begin(9600);

  Serial.println("Zigbee串口初始化: 9600波特率");

  Serial.printf("  RX引脚: D6 (GPIO12)\n");

  Serial.printf("  TX引脚: D7 (GPIO13)\n");

  // 初始化DHT11

  dht.begin();

  Serial.println("DHT11初始化完成");

  // 初始化MQ-2

  pinMode(MQ2\_DO, INPUT);

  Serial.println("MQ-2初始化完成");

  Serial.println("\n开始发送数据...\n");

  delay(2000);

}

void loop() {

  // 读取DHT11传感器

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // 读取MQ-2传感器

  int gasValue = analogRead(MQ2\_AO);

  int gasDigital = digitalRead(MQ2\_DO);

  // 检查DHT11读取是否成功

  if (isnan(temperature) || isnan(humidity)) {

    Serial.println("⚠ DHT11读取失败，使用默认值");

    temperature = 25.0;

    humidity = 60.0;

  }

  // 构建数据包（使用简单的分隔符格式）

  String dataPacket = "T:" + String(temperature, 1) +

                      ",H:" + String(humidity, 1) +

                      ",G:" + String(gasValue) +

                      ",A:" + String(gasDigital) + "\n";

  // 通过Zigbee发送数据

  zigbeeSerial.print(dataPacket);

  // 串口监视器显示

  sendCount++;

  Serial.printf("[%lu] 发送: %s", sendCount, dataPacket.c\_str());

  Serial.printf("     温度=%.1f°C, 湿度=%.1f%%, 气体=%d, 报警=%d\n",

                temperature, humidity, gasValue, gasDigital);

  // LED闪烁指示发送（ESP8266内置LED在GPIO2，即D4）

  digitalWrite(LED\_BUILTIN, LOW);   // 点亮（低电平有效）

  delay(100);

  digitalWrite(LED\_BUILTIN, HIGH);  // 熄灭

  delay(2000);  // 每2秒发送一次

}