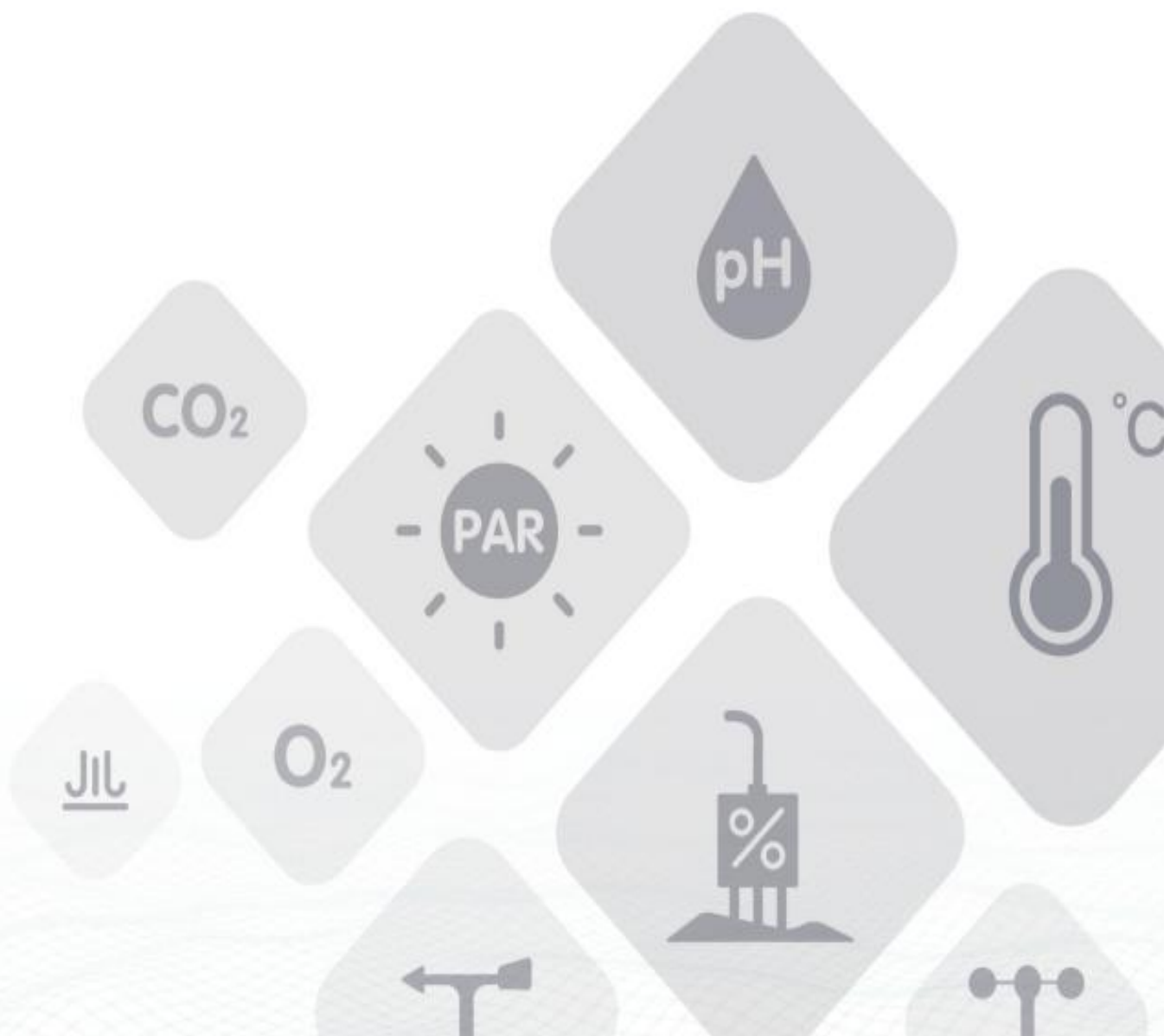


seeed studio

多功能环境监测仪

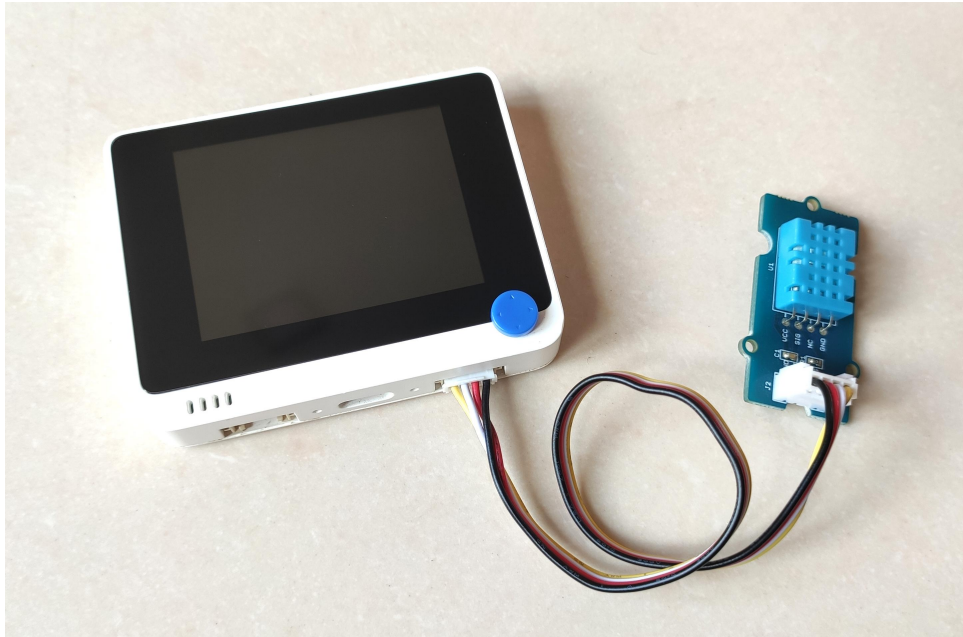
用户手册



目 录

1. 产品介绍	1
1.1. 产品清单	1
2. 产品规格	2
2.1. Wio 终端	2
2.2. Grove-温湿度传感器（DHT11）	3
2.3. 软件工具	4
3. 操作部署流程	4
4. 代码解析	8
4.1. 代码功能解析	8
1) 硬件与传感器初始化	8
2) 网络与时间同步	8
3) 核心功能架构	9
4) 报警系统	9
5) 传感器数据处理	10
6) 显示系统	10
7) 网络通信	11
8) 按键控制	11
4.2. 系统工作流程	12
1) 初始化阶段：	12
2) 主循环	12
3) 报警处理流程	13
4.3. 代码总结	14
5. 常见问题	14
6. 示例界面	15

1. 产品介绍



此设计为基于 Wio 终端的多功能环境检测仪。它集成了多个传感器（温湿度、光照、噪声、加速度）来实时监控环境参数，并通过 WiFi 将数据上传到 MQTT 服务器。系统具有本地显示屏，可以切换不同界面查看数据，并且在检测到异常（如温度超标、设备移动等）时触发声光报警。同时，系统支持 NTP 时间同步，提供日期时间显示。用户可以通过物理按键与设备交互，例如切换显示界面或关闭报警。

应用十分广泛，如温室大棚环境监测、粮仓温湿度监控、野外种植基地防盗系统、育苗室光照管理。通过多传感器融合与实时告警机制，实现农业生产环境的无人化智能监护。

1.1. 产品清单

- Wio 终端: https://wiki.seeedstudio.com/Wio_Terminal_Intro/
- Grove-温湿度传感器(DHT11):
https://wiki.seeedstudio.com/Grove-TemperatureAndHumidity_Sensor/

● 其他：双向端子连接线（可参考产品介绍图）、USB-C 电缆；

● 软件工具：MQTTX: <https://mqttx.app/zh/features>

Arduino IDE: <https://www.arduino.cc/en/software/#ide>

注：使用前请先准备好以上硬件设备及软件工具；

2. 产品规格

2.1. Wio 终端

核心配置	主控芯片	微控制器：ARM® Cortex®-M4
		主频：120 MHz
		存储：192KB SRAM + 4MB 外部闪存
	显示屏	尺寸：2.4 英寸
		分辨率：320×240 像素
无线连接	Wi-Fi	RTL8720DN 模块 支持 2.4GHz / 5GHz 双频 协议：802.11 a/b/g/n
	BLE	BLE 5.0（兼容 4.2）
传感器与输入	内置传感器	光传感器（LTR-553ALS） 6 轴 IMU（LSM6DS3TR-C，加速度计+陀螺仪） 红外发射器（IR 940nm）
	交互控件	5 向摇杆（方向+按下） 2 个可编程按键（侧面） 复位/电源按钮

扩展接口	Grove 生态系统	1× Grove I ² C 接口（背面） 1× Grove 数字/模拟接口（底部）
	GPIO 引脚	兼容 Raspberry Pi 40-pin（支持 UART/I ² C/SPI/ADC）
	其他接口	USB Type-C（供电/编程） MicroSD 卡槽（最大支持 32GB） 2.4G 天线接口（外接天线）
电源	输入	USB Type-C（5V/2A）
	电池扩展	支持锂电池（通过底部接口充电）
开发支持	编程环境	Arduino IDE / PlatformIO MicroPython / CircuitPython
	兼容框架	Arduino、TinyML、Edge Impulse 等
物理规格	尺寸	72mm * 57mm * 12mm
	外壳	ABS + PC

2.2. Grove-温湿度传感器 (DHT11)

温度	测量范围	0°C ~ 50°C
	精度	±2°C
	分辨率	0.1°C
湿度	测量范围	20% ~ 90% RH
	精度	±5% RH
	分辨率	0.1% RH
采样率		1 Hz (每秒采样 1 次)

接口	类型	Grove 4-pin 标准接口
	协议	数字信号 (单总线协议)
	引脚	VCC, GND, NC (未连接), SIG (信号)
供电	电压电流	5V/2A
规格	尺寸	40mm x 20mm x 12mm
	重量	约 4.5 克

2.3. 软件工具

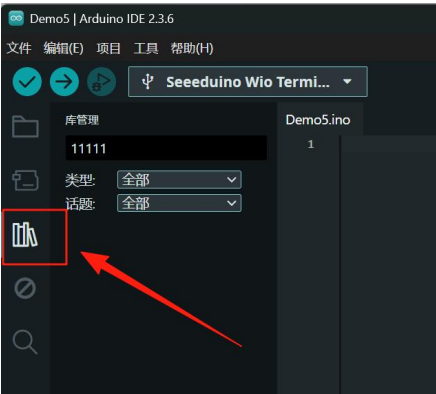
下载 MQTTX 和 Arduino IDE 第三方工具（下载链接见产品清单）

MQTTX：用于订阅 Wio 终端主题查看上行数据及下行指令；

Arduino IDE：用于给 Wio 终端烧录编程代码；

3. 操作部署流程

- 1) 准备所需软硬件设备（Wio 终端、Grove-温湿度传感器(DHT11)、双向端子连接线、USB-C 电缆、MQTTX、Arduino IDE）；
- 2) 将 Grove-温湿度传感器(DHT11)连接到 Wio 终端的 D0 端口(五向开关下方端口)；
- 3) 通过 USB-C 电缆将 Wio 终端连接到 PC；
- 4) 打开 Arduino IDE，点击如图所示图标，进入“库管理”，或者可点击“项目—



导入库—管理库” 进入“库管理”；

5) 安装运行代码所需的所有库:

TFT_eSPI

功能: 驱动 Wio Terminal 的 TFT 显示屏

安装方式: Arduino 库管理器搜索 TFT_eSPI (作者: Bodmer)

LIS3DHTR

功能: 驱动 LIS3DHTR 加速度传感器

安装方式: 搜索 LIS3DHTR (作者: Seeed Studio)

DHT Sensor Library

功能: 读取 DHT11 温湿度传感器数据

安装方式: 搜索 DHT sensor library (作者: Adafruit)

注意: 需同时安装 Adafruit Unified Sensor 库

PubSubClient

功能: MQTT 客户端通信

安装方式: 搜索 PubSubClient (作者: Nick O'Leary)

NTPClient

功能: 从 NTP 服务器同步时间

安装方式: 搜索 NTPClient (作者: Fabrice Weinberg)

WiFi (内置)

功能: Wio Terminal 的 WiFi 连接 (通过 rpcWiFi.h)

说明: 包含在 Seeed Studio SAMD Boards 板支持包中, 无需单独安装

WiFiUDP (内置)

功能: UDP 协议支持 (NTP 依赖)

说明: Arduino 核心库, 无需单独安装

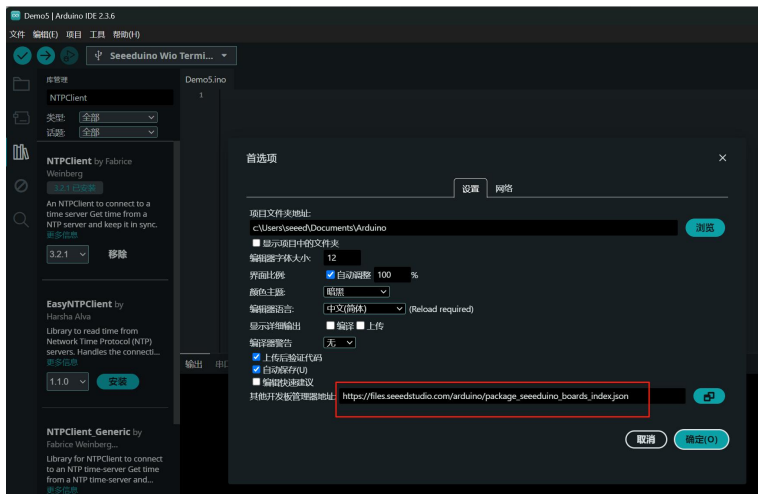
6) 安装步骤:

● 安装板支持包

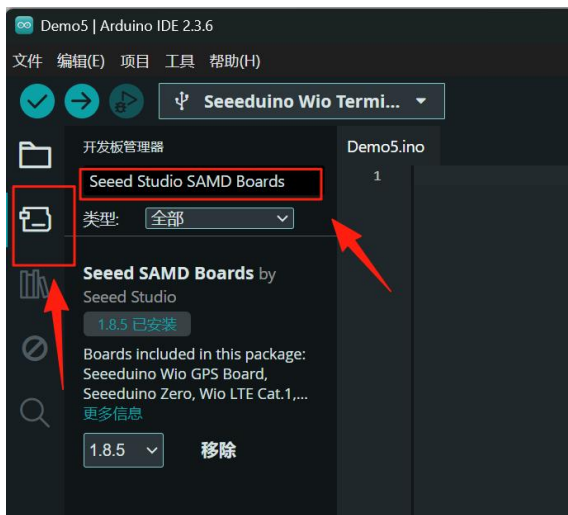
Wio Terminal 需要 Seeed Studio SAMD Boards 支持包:

① Arduino IDE → 文件 → 首选项 → 附加开发板管理器网址 添加:

https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json

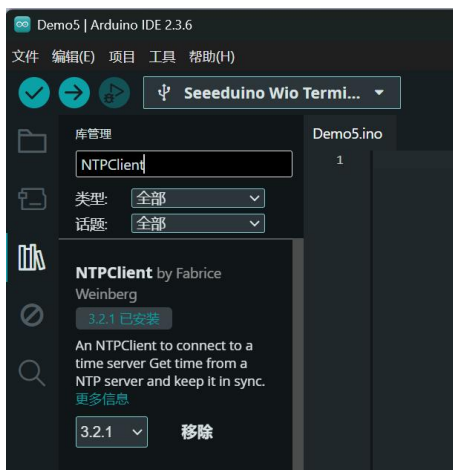


② 工具 → 开发板 → 开发板管理器 → 搜索 Seeed Studio SAMD Boards → 安装最新版。



● 通过库管理器安装依赖库

① 打开 Arduino IDE → 项目 → 加载库 → 管理库...



② 搜索并安装以下库：

TFT_eSPI

LIS3DHTR

DHT sensor library（同时安装 Adafruit Unified Sensor）

PubSubClient

NTPClient

● 验证库安装

① 选择开发板：

工具 → 开发板 → Seeed Studio SAMD Boards → Wio Terminal

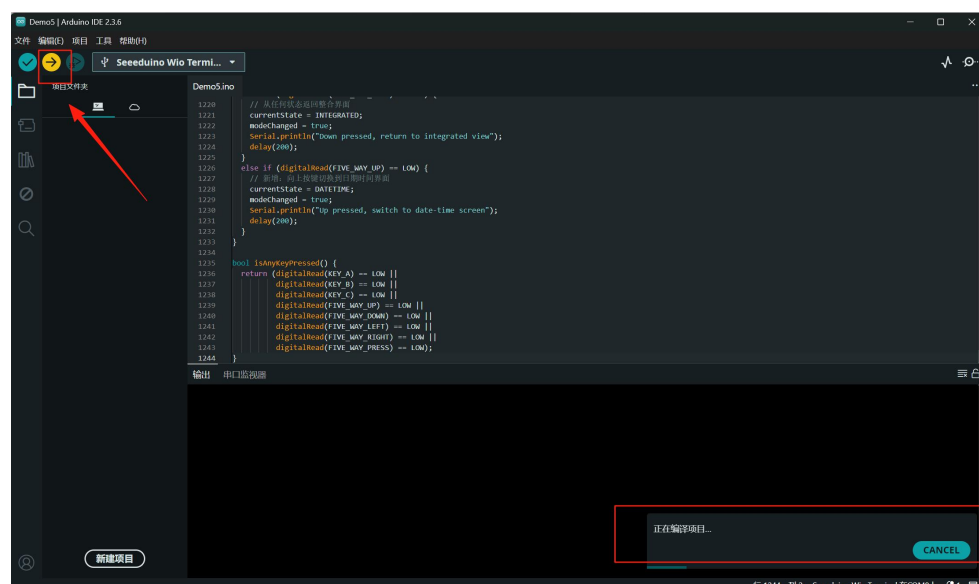
② 连接 Wio Terminal 并上传示例代码：

确保所有 `#include` 语句无报错

编译时无 `undefined reference` 错误（如有报错，可根据报错信息提示进行处理）

7) 上传代码，从以下链接直接复制到 Arduino IDE 上传即可；

<https://github.com/huakaiyang/Multifunctional-Environmental-Monitoring-Instrument/blob/main/code.txt>



4. 代码解析

4.1. 代码功能解析

1) 硬件与传感器初始化

```
// 硬件引脚定义
#define BUZZER_PIN WIO_BUZZER
#define DHTPIN 0
// ...其他引脚定义...

// 传感器对象
TFT_eSPI tft;           // TFT显示屏
LIS3DHTR<TwoWire> lis;  // 加速度计
DHT dht(DHTPIN, DHTTYPE); // 温湿度传感器
```

- 显示屏：使用 TFT_eSPI 库驱动 320x240 分辨率屏幕
- 加速度计：LIS3DHTR 传感器，配置为 4G 量程和 50Hz 采样率
- 温湿度：DHT11 传感器，通过数字引脚 0 读取

2) 网络与时间同步

```
// NTP时间客户端
NTPClient timeClient(ntpUDP, "pool.ntp.org", 8 * 3600);

// WiFi和MQTT配置
const char *ssid = "abcd";
const char *password = "123456789";
PubSubClient mqttClient(wifiClient);
```

- NTP 时间同步：每 60 秒同步一次 UTC+8 时间
- WiFi 连接：实现自动重连机制
- MQTT 通信：连接 test.mosquitto.org 服务器，支持数据发布/订阅

3) 核心功能架构

```
enum DisplayState {  
    INTEGRATED,      // 综合数据显示  
    TEMPERATURE,     // 温度单独显示  
    HUMIDITY,        // 湿度单独显示  
    // ...其他状态...  
    DATETIME         // 日期时间界面  
};
```

- 8 种显示模式：通过物理按键切换不同数据显示界面
- 状态管理：使用 `currentState` 变量跟踪当前显示模式

4) 报警系统

```
enum AlarmType {  
    NONE, TEMP_LOW_ALARM, TEMP_HIGH_ALARM,  
    // ...7种报警类型...  
};  
  
const float TEMP_HIGH = 28.0; // 温度高阈值  
const float ACCEL_THRESHOLD = 1.3; // 加速度变化阈值
```

- 多类型报警：温度/湿度/光照/噪声/运动等异常检测
- 阈值配置：可自定义各传感器报警阈值
- 报警处理：
 - ◆ 蜂鸣器播放高低音交替警报
 - ◆ 屏幕显示红色警报界面
 - ◆ 自动上传 MQTT 报警信息
 - ◆ 支持按键手动解除报警

5) 传感器数据处理

```
struct SensorData {  
    float temperature;  
    float humidity;  
    int lightLevel;  
    // ...其他数据...  
};  
  
void readSensors() {  
    // 读取所有传感器数据  
    sensorData.temperature = dht.readTemperature();  
    // ...其他传感器读取...  
}
```

- 结构化存储：使用 SensorData 结构体统一管理数据
- 定时采集：每秒读取一次所有传感器
- 数据滤波：对 DHT11 数据做 NaN 检查

6) 显示系统

```
void drawIntegratedScreen() {  
    // 显示所有传感器数据  
    // 网络状态  
    // 操作提示  
}  
  
void drawDateTimeScreen() {  
    // 显示Seeed Studio LOGO  
    // 当前日期和时间  
    // NTP同步状态  
}
```

- 综合界面：同时显示温度/湿度/光照/噪声/加速度数据
- 单传感器界面：大字体显示特定传感器数据
- 专业日期界面：
 - ◆ 绿色"Seeed Studio"LOGO
 - ◆ 居中显示日期(YYYY/MM/DD)和时间(HH:MM)
 - ◆ 显示下次 NTP 同步时间

7) 网络通信

```
void publishSensorData() {
    // 发布JSON格式传感器数据
    // 示例: {"temp":25.5,"humi":45.0,...}
}

void publishAlarmData(const char* alarmType) {
    // 发布报警信息含详细传感器数据
}
```

- MQTT 协议：使用 PubSubClient 库
- 数据格式：标准 JSON 格式
- 主题区分：
 - ◆ 普通数据发布到"WioTerminal"主题
 - ◆ 报警数据包含额外 alarm 字段

8) 按键控制

```
void checkButtons() {
    if (digitalRead(KEY_A)==LOW) currentState=TEMPERATURE;
    // ...其他按键检测...
}
```

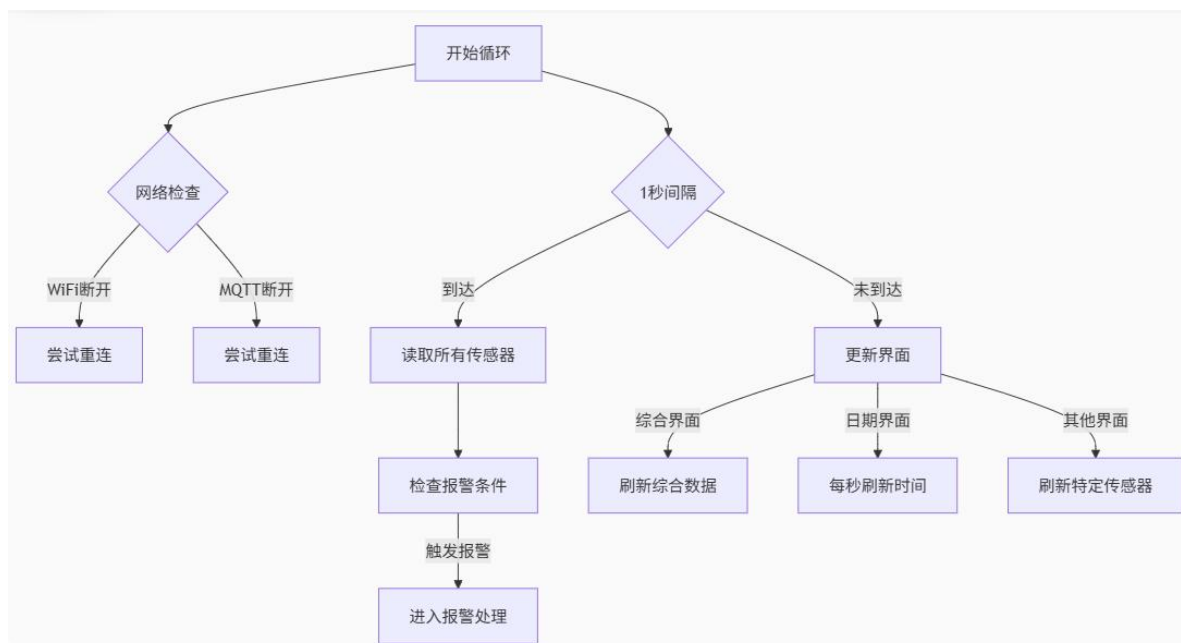
- 物理按键映射：
 - ◆ KEY_A：温度界面
 - ◆ KEY_B：湿度界面
 - ◆ KEY_C：光照界面
 - ◆ 五向键左：加速度界面
 - ◆ 五向键右：噪声界面
 - ◆ 五向键下：返回综合界面
 - ◆ 五向键上：日期时间界面

4.2. 系统工作流程

1) 初始化阶段：

- 启动屏幕显示初始化信息
- 初始化所有传感器和网络连接
- 显示综合数据界面

2) 主循环



graph TD

A[开始循环] --> B{网络检查}

B -->|WiFi 断开| C[尝试重连]

B -->|MQTT 断开| D[尝试重连]

A --> E{1 秒间隔}

E -->|到达| F[读取所有传感器]

F --> G[检查报警条件]

G -->|触发报警| H[进入报警处理]

E -->|未到达| I[更新界面]

I -->|综合界面| J[刷新综合数据]

I -->|日期界面| K[每秒刷新时间]

I -->|其他界面| L[刷新特定传感器]

3) 报警处理流程

- 检测到传感器超阈值
- 记录报警类型和开始时间
- 切换到红色警报界面
- 蜂鸣器播放警报音
- 上传 MQTT 报警信息
- 5 秒后或按键按下时解除报警

4.3. 代码总结

该代码实现了一个完整的环境监测系统，具有：

- 多传感器数据采集（温湿度/光照/噪声/加速度）
- 8 种显示界面灵活切换
- 智能阈值报警系统
- 网络时间同步功能
- MQTT 物联网通信
- 直观的用户界面和交互

5. 常见问题

- 上传代码失败？
 - ✓ 根据报错提示检查是否未按要求安装库，可复制报错信息使用 AI 工具搜索问题。
- Wio 终端屏幕卡住？
 - ✓ 向下拨动侧边按钮重启设备。
- 网络异常，未能连接上 WiFi 和 MQTT 服务器？
 - ✓ 可修改代码内 WiFi 和服务器配置，改为您需要链接的网络和服务器。

6. 示例界面

