//本项目为了熟悉嵌入式驱动S操作和后端C++服务端开发。

//工具:STM32F103C8T6,

//环境：Keli5

//描述:基于STM32为下位机采集xx数据, PC为上位机服务器部署在云端，使用QT编写客户端。使用阿里云实现部署服务器，获取下住机（客户端的）数据，并给出操作信号，控制客户端的工作状态。

//挂术栈(待定): C(客户端),C++(网络)、(可能的数据处理神经网络)Python、线程池(客户端)、（客户端）多进程、（）网络通信、QT、IO多路复用、Linux系统编程, 底层数据采集:在STM32下使用SPI/IIC/单总线等协议对硬件模块进行驱动，实现数据采集以及多种对外通信方式。

上位机服务器:通过apol1+线程池架构结合多进程/共享内存/条件变量处理多客户端迷接以及线程资源管理。

阿里云中转站:使用阿里云物联网中的云产品流转等功能实现上下住机的通信，通过物模型实时监控环境系统。

可视化客户端:基于0T的Widzet、PushButton等配合Mysq|以及Socket实现与服务器继接以及数据显示和设置功能。

描述:

[RTOS 下位机] <--> [C++ 后端服务器] <--> [Qt 客户端]

(STM32 + FreeRTOS) (Linux 或云平台部署) (跨平台桌面应用)

任务调度 数据处理/API接口 图形化展示和交互

数据采集/设备控制 数据存储和分析 查询/操作/可视化

MQTT/HTTP Web API/MQTT Broker HTTP/MQTT

### ****三者角色分工****

#### ****1. RTOS（下位机）****

* **职责：**
  + 运行在嵌入式设备上（如 STM32）。
  + 管理硬件资源、实时任务调度。
  + 负责数据采集（如传感器）、设备控制、通信等底层功能。
* **开发语言：** C 或 C++。
* **典型任务：**
  + 数据采集任务（定时采集传感器数据）。
  + 数据通信任务（如通过 MQTT 或 UART 发送数据）。
  + 外设控制任务（控制电机、LED、继电器等）。

#### ****2. Qt（客户端）****

* **职责：**
  + 提供图形用户界面（GUI）作为用户与系统的交互入口。
  + 与后端或 RTOS 通信，获取数据并显示。
  + 接受用户指令并发送到后端（或通过后端转发给 RTOS）。
* **开发语言：** C++（支持跨平台开发）。
* **典型任务：**
  + 实时数据显示（如设备状态、传感器数据）。
  + 控制操作界面（如按钮操作、参数设置）。
  + 历史数据查询或可视化。

#### ****3. C++ 后端（云端服务器）****

* **职责：**
  + 作为系统的中枢，负责协调客户端和下位机的通信。
  + 提供 API 接口供客户端或其他设备访问。
  + 存储数据并处理复杂逻辑（如设备状态判断、数据分析）。
* **开发语言：** C++（如使用 Boost、Poco 框架）或其他适合后端开发的语言（Python、Node.js）。
* **典型任务：**
  + 接收 RTOS 数据，存入数据库。
  + 将指令从客户端转发到 RTOS。
  + 提供 REST API 或 WebSocket 接口给客户端调用。

### ****系统实现步骤****

#### ****1. stm32开发，先开发功能，再移植 FreeRTOS****。

* **FreeRTOS 任务：**
  1. **数据采集任务：**
     + 定时从传感器读取数据，存入缓冲区。
  2. **通信任务：**
     + 使用 MQTT 或 HTTP 协议将数据发送到后端。
  3. **设备控制任务：**
     + 接收来自后端的指令，控制硬件外设。
* **通信协议：**
  1. 推荐使用 **MQTT** 协议，轻量高效，适合嵌入式设备。

#### ****2. C++ 后端开发****

* **框架选择：**
  + 使用 **Poco** 或 **Boost.Asio** 框架开发高性能 C++ 后端。
  + 提供 API 接口（如 RESTful 或 WebSocket）供 Qt 客户端调用。
* **主要功能：**
  + 接收 RTOS 上传的传感器数据。
  + 将数据存储到数据库（如 MySQL 或 MongoDB）。
  + 处理业务逻辑（如告警、决策）。
  + 转发客户端指令到下位机。

#### ****3. Qt 客户端开发****

* **开发工具：**
  + 使用 **Qt Creator** 开发跨平台 GUI 应用。
* **主要功能：**
  1. **用户交互界面：**
     + 图表显示传感器数据（如温湿度曲线）。
     + 提供控制按钮和设置界面。
  2. **网络通信：**
     + 使用 **QNetworkAccessManager** 调用后端的 REST API。
     + 或使用 **Qt MQTT** 直接订阅后端的数据。
  3. **数据处理：**
     + 实时更新数据。
     + 保存或导出数据到本地文件。