环境监测系统概要设计

# 前 言

随着我国信息技术的迅猛发展以及空气调节技术的不断完善，如今已经具备了实现基于网络技术进行远程监测的硬件基础，同时在软件方面实时监测、远程监测等单元技术也有了一定的成果和经验，而且社会的信息化唤起了人们对住宅智能化的要求，智能住宅己成为中国房地产市场的主流，目前人们对室内管理智能化的需求已日趋明显，为了适应人们现如今快节奏的生活方式，设计出室内环境监控系统是非常必要而且可行的。

综上所述，笔者及其团队选择了本次的项目——办公环境监测系统。在本次项目设计中，笔者将模块分为四大部分，分别为——开发板数据交互部分、服务器后端、开发板界面部分和web前端。

目录

[前 言 1](#_Toc535328112)

[一、 项目整体构思 3](#_Toc535328113)

[二、 数据库设计 4](#_Toc535328114)

[三、 程序设计概要 5](#_Toc535328115)

[3.1 开发板数据交互设计 5](#_Toc535328116)

[3.2 服务端设计 5](#_Toc535328117)

[3.3 WEB端设计 5](#_Toc535328118)

[3.4 开发板QT界面设计 6](#_Toc535328119)

1. 项目整体构思

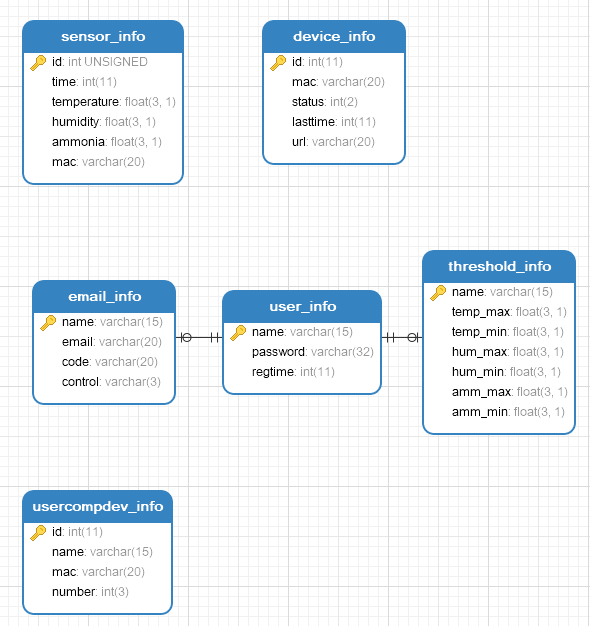
首先，对于本次项目的定位，笔者及其团队认为本项目应该是一整套完备的“服务”，而不是仅仅围绕单个开发板的程序开发。因此，我们设想用户可以拥有对多个设备（开发板及传感器等检测工具）的监控权，而一个设备也可以被多个用户监控，这也是笔者团队在本次项目设计中的“多对多”设计理念。

第二，在本项目中，开发板上将设计两个作用不同的程序——数据获取/发送程序与QT界面程序。经过深思熟虑，笔者及其团队认为，这两个程序并没有被捆绑在一起的必要，分为两个程序可以更具有灵活性，也更方便添加一些功能，同时也减少了代码量过大带来的bug和开发难度。

1. 数据库设计

与围绕单个设备进行开发相比，本项目的服务器数据库要复杂得多，因为要构建多用户—多设备的对应关系。总体来说分为六个数据表：数据表（存储所有设备得到的数据），用户表（存储用户信息），设备表（存储设备信息），用户设备对照表（指明用户与设备之间的关系），用户预警设定表（存放用户的预警设定）以及用户邮箱表（用于邮箱报警和邮箱验证）。

具体的表内容如下图：



1. 程序设计概要

## 3.1 开发板数据交互设计

本模块主要由C语言实现，设计是本项目最基础的部分。在本模块中，程序主要完成以下功能：

1. 通过MODBUS-RTU协议实现开发板与传感器的数据交互。
2. 将获取的数据存入开发板上的本地数据库。
3. 与服务器建立连接，将获取的数据发送至服务端。
4. 实现心跳机制与自动重连机制，增强程序稳定性与健壮性。
5. 设置网络错误与传感器数据错误的处理，让程序在没有网络的情况下也能继续运行（仅获取数据并存入本地数据库，可供本地QT界面查看）。

## 3.2 服务端设计

本模块由java语言实现，运用了springboot框架、JPA技术以及websocket技术，主要完成以下功能：

1. 接收开发板发送的数据并写入数据库
2. 用户模块（注册、登录、修改密码等）
3. 实时数据查询模块
4. 历史数据查询模块
5. 设备控制模块（添加、删除设备，开关视频推流）
6. 预警管理模块
7. 处理web端各类额外功能的请求

## 3.3 WEB端设计

本模块由html+js+css实现，运用了websocket、JQuery、Ajax、Echarts等技术，主要完成以下功能：

1. 实现用户的登录/注册以及修改密码的功能。
2. 能够查看用户拥有的所有设备的状态。
3. 发出实时数据请求后，使用websocket接收服务器发来的实时数据并以动态数字和动态的折线图两种方式展示出来。
4. 实时数据的数值达到预警值时，通过高亮显示或邮件提醒的方式向用户发出警告。
5. 用户可以以折线图的形式查看到历史数据（提供默认选项与自定义选项）。
6. 用户可以通过接收推流的方式查看视频监控。
7. 用户可以在网页上远程打开/关闭设备上的视频推流。
8. 用户可以通过邮箱验证码的方式找回密码。

## 3.4 开发板QT界面设计

本模块主要由C++、QML语言实现，使用QTcreator编写代码，主要完成以下功能：

1. 设计界面时考虑界面美观性与用户使用的便利性。
2. 实现用户登录功能，但是为了考虑到没有网络的情况，也可以离线使用。
3. 实现实时监测数据的展示。
4. 实时数据的数值达到预警值时，通过高亮的方式向用户发出警告。
5. 实现历史数据的查询（提供默认选项与自定义选项）。
6. 实现查看本地摄像头视频内容。
7. 实现更改用户设定的阈值（报警值）