# 1、完成情况

1.1、借鉴了哪些开源项目代码？除开开源项目代码外，项目组独立完成的内容有哪些？请展开描述

后端：EMQ，mqtt通信包

前端：iview UI组件

1.2、项目组成员在项目中参与的环节和担任的角色

|  |  |
| --- | --- |
|  |  |
| 杜瑞祥 | 前端实现，前端与后端的通信 |
| 李阳 | 后端实现，后端与硬件的通信 |

**（此处请留下项目组成员合影）**

1.3、总的用时

# 2、关键技术选取与预研过程

## 2.1、关键技术

后端：Springboot，Spring-Data，Redis，Nginx，MQTT，HTTP，EMQ

前端：

* HTML（WXML）：章节语义标签、组织内容、文字形式、嵌入内容、交互元素、事件；
* CSS（WXSS）： 元素选择器、关系选择器、属性选择器、伪类选择器、字符串函数、盒模型、网状布局、状态；
* JavaScript语言： DOM节点操作、Event对象、事件流、事件处理 /监听函数、Cookie；
* 微信小程序框架： 微信原生API、基础组件、数据双向绑定、页面管理；
* HTTP协议：请求方法、状态代码、消息报头务器的软件，发送响应的时间，回应数据的格式等信息；响应正文：就是响应的具体数据。；

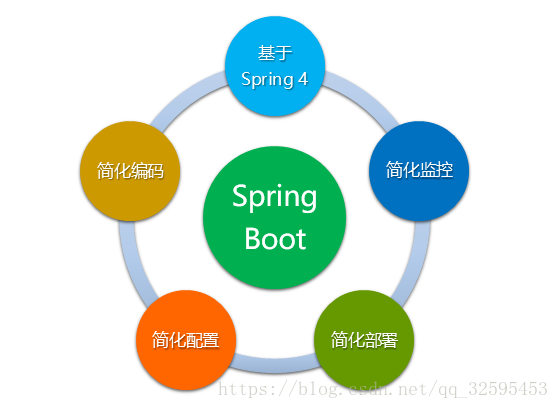
## 2.2、关键技术的预研过程

### 2.2.1、技术的主要来源

Github，Google，CSDN，各种网络博客

### 2.2.2、为什么要使用这些技术？

Nginx： 1、可以高并发连接2、内存消耗少3、成本低廉 4、配置文件非常简单5、支持Rewrite重写 6、内置的健康检查功能 7、节省带宽 8、稳定性高

Springboot：

简化编码、简化配置、简化部署、简化监控

成熟的框架，运用java编写极高的移植性，高并发性

MQTT协议：1.精简，不添加可有可无的功能。以发布/订阅（Pub/Sub）模式为中心，方便消息在传感器之间传递。2.把传输量降到最低以提高传输效率。因为MQTT协议的头很小。3.MQTT提供层级主题4.MQTT提供多个QoS选项（exact once、at least once、at most once）5.MQTT代理加上了对WebSockets的支持6.MQTT提供了多个层次的安全特性7.设置了一个Will，如果客户端异常断开链接，borker会发送一个遗嘱信息8.当执行loop时，才能对消息进行处理9.MQTT为开发者提供了大量的回调函数，实时的可以监控当前状态,可以对发生的事件进行及时的应对。10.对于只需要建立一次连接，发送大量数据的情况，MQTT提供了simple方法供开发者调用。

微信小程序框架：1、用户体验好2、流量获取易3、开发成本低4、微信生态强5、快速迭代6、跨平台共享7、提升开发体验。

HTTP协议：1、支持客户/服务器模式2、简单快速，客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。3、灵活，HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type（Content-Type是HTTP包中用来表示内容类型的标识）加以标记。4、无连接，无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。5、无状态，HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

### 2.2.3、预研过程中，遇到哪些问题？都怎样解决的？

服务器如何与硬件完成通信？

参照大量物联网设备的通信方法，最终确定了使用MQTT协议

如何搭建MQTT服务器？

可以自己实现一个简单的MQTT服务器，但在网上找到了一个很好的开源MQTT服务器EMQ，最终选择使用优秀开源服务器EMQ

如何保证与硬件通信的安全？

在服务器发送信息时将传输的信息进行签名，到达硬件端端时候进行信息验签

如何解决跨域问题？

在http头中增加允许跨域的字段

如何封装http请求？

解决方式：根据之前编写前端框架VUE的经验，编写了一个工具类，将微信小程序自带的请求API封装在该工具类中，并提供了请求成功、失败和完成的回调函数，并加入了请求失败的提示，在页面中只需引入该工具类即可，减少代码冗余。

如何计算GPS直线距离？

解决方式：通过网上查询博客，了解了大概基于经纬度的距离计算，并根据其数学公式编写了JavaScript语言的距离函数。

# 3、关于成果

## 3.1、其他技术来源

## 3.2、成果定量数据

代码行数、成果类型（硬件？软件？分别描述）、运行和测试结果如何？自我满意度如何？同学互相评价如何？能否投入实际使用？

|  |  |
| --- | --- |
| 类型 | 行数 |
| 后端 |  |
| 前端 |  |
| 硬件 |  |
| 测试 |  |

服务经过一次线上虚拟测试，两次线下实际测试，已经可以投入实际环境正常使用

## 3.3、项目过程及成品展示照片：（不低于5幅）

# 4、开发感想

4.1、你有什么感想？开发过程中心情和心态的变化？这门课前后，对软件开发实践有没有突破性的改变？

李阳: 最大的收获就是在实际过程发现一个软件只是简单的完成功能与想要真正投入使用是完全不一样的，功能的实现与否可能在一个系统中占用的开发时间只有20%，而剩下的80%都是在处理其它的质量属性比如可用性，安全性，性能等。另外就是需要和其它人员沟通好需求，一个系统就算再简陋需求分析一定是要有的，当需求不确定时往往会在开发中期引起大范围的修改，甚至可能遗弃全部前期的开发，需求不明确造成的后果是十分严重的

杜瑞祥：完成一个项目并不能只着眼于眼前的需求，留意软件的可扩展性在开发中至关重要，有时候一个需求的简单变化，有可能会导致之前的代码完全作废，因此在编写代码时，应该有高瞻远瞩的心志。

4.2、你所参与的系统还有什么要改进的？下一步怎么打算？

服务器与硬件的通信效率和安全性可以进一步提高，打算在下一步使用更加适合的通信加密算法，以及优化EMQ服务器的通信效率。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 填报人 | | 李阳 | 填报时间 | 2019/4/10 | |
| 开发阶段 | | 起始 | 起止日期 | 2019/4/10-2019/4/15 | |
| 本阶段进度 | 已完成事项 | 1. 服务器的初始化 2. 软硬件通信对学习 | | | |
| 待完成事项 | 1. 如何与硬件完成通信 | | | |
| 原因及对策 | 遇到了跨域问题，在http返回头中加入允许跨域字段 | | | |
| 下阶段计划 | 完成服务器与硬件的通信 | | | | |
|
|
|
|
|
| 资源需求 | 编号 | 所需资源 | | | 备注 |
|  |  | | |  |
|  |  | | |  |
| 备注 |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 填报人 | | 李阳 | 填报时间 | 2019/4/20 | |
| 开发阶段 | | 开发 | 起止日期 | 2019/4/20-2019/4/30 | |
| 本阶段进度 | 已完成事项 | 1. 使用EMQ完成mqtt服务器的搭建 2. 选择mqtt协议进行通信 | | | |
| 待完成事项 | 1. 完成服务器与硬件的通信 2. 对服务进行线上和线下测试 | | | |
| 原因及对策 |  | | | |
| 下阶段计划 | 完成服务器与硬件的通信  对服务进行线上和线下测试 | | | | |
|
|
|
|
|
| 资源需求 | 编号 | 所需资源 | | | 备注 |
|  |  | | |  |
|  |  | | |  |
| 备注 |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 填报人 | | 李阳 | 填报时间 | 2019/5/1 | |
| 开发阶段 | | 测试 | 起止日期 | 2019/5/1-2019/5/10 | |
| 本阶段进度 | 已完成事项 | 1. 完成服务器与硬件的通信 2. 进行了服务线上与线下测试 | | | |
| 待完成事项 | 1. 对服务器和硬件的通信进行加密 | | | |
| 原因及对策 | 测试过程中发现mqtt协议qs设置错误，更改qs的设置 | | | |
| 下阶段计划 | 对服务器和硬件的通信进行加密 | | | | |
|
|
|
|
|
| 资源需求 | 编号 | 所需资源 | | | 备注 |
|  |  | | |  |
|  |  | | |  |
| 备注 |  | | | | |