

项目总结报告

日期：2018.9.8

组号	15	项目名称	微电网智能监控与调度系统
编程语言	Python, arduino, java, matlab, c++, javascript, html, css	开发平台和框架	Java 开发平台（后端） ReactJS 开发平台（前端） Maven 框架 Springboot 框架

软件需求特性

1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。

我们实现了全部的基本需求，额外又实现了一些新的需求。

新的需求：体感控制。温度，光照强度，测距等多元丰富的传感器

2.采用哪种架构风格？哪些设计模式？

四层 C/S 结构风格：表示层，功能层，数据持久化层，数据采集层，在物理架构上拥有前端浏览器，后端服务器，第三方云服务，数据库，本地集成控制器，硬件系统。

设计模式：抽象工厂、适配器模式、代理模式、单例模式、原型模式

3.技术方案有哪些亮点？

1、我们和硬件交互，加入了多种传感器（尤其是体感的传感器，实现了人们挥一挥手即可控制用电开关），我们通过传感器监控实时的环境变量，自动化的开启与关闭用地器，方便人们的生活

2、我们研发了 AI 小微，实现了智能家居，AI 可以帮助我们进行语音的识别，手势的识别，使我们的生活更加得智能与便捷，无聊的时候可以和我们 AI 小微聊天，也不失为一种乐趣

3、我们自行设计了非线性自回归神经网络，对太阳能发电进行了预测，我们结合了历史的数据以及实时的天气状况，对未来的太阳能发电进行了精准的预测（详细请见太阳能预测报告）

4、我们自行设计了调度算法，我们为用电及供电系统设计了模型，综合考虑了已在运行的任务，等待运行的任务，未来可能加入的任务，太阳能的发电，蓄电池的容量等等多方面的因素，智能规划每个任务的执行方案，做到最经济最绿色。（详见调度算法报告）

4. 是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？

进行了单元测试，系统功能测试，性能测试，兼容性测试。

项目组成员对项目的贡献度（%）	
<p>注：总计 100%</p> <p>冯二虎：25%</p> <p>李东起：25%</p> <p>蔡一凡：25%</p> <p>姜凡叙：25%</p>	
软件度量	
软件代码行数（不包括注解行、空行和复用代码）：	15662
复用他人代码行数：	6042
类的个数：	53
经验、教训和建议	
<p>1、做一个大型项目的时候要合理的规划好迭代的计划，找到项目风险，合理分配任务。一个大型的项目不像一般的小程序，可以直接上手写代码，只有在编程前做好详细地准备，构思好完善的软件构架，才能在编程的时候事半功倍。同时在编程完毕之后还需要进行详细的系统测试，单元测试，性能测试。在长期的编程过程中，我们时常用功能实现就是软件的全部麻痹自己，但是只有在真正的项目中才会体验到测试的重要性，软件的测试是为了保障用户的权利。</p> <p>2、在进行机器学习，神经网络的训练的过程中，调参是一件非常重要但又十分枯燥的事情，不管是语音识别的调参还是在进行太阳能的预测，神经网络的参数设置都是至关重要的，但是参数的选取是凭借经验的，所以只有一次次的尝才能找到的合适的参数。在寻参的过程我，一定要耐心坚持，可以采取逼近法等辅助的方法帮助。在我们本次的项目中为了寻找到合适的参数，我们前后花了几十个小时，最后终于找到了合适的参数，预测的效果比较的理想，可以说是功夫不负有心人。</p> <p>3、因为我们该项目工程量大的，使用的大量的语言，包括了 <code>arduino</code>，<code>python</code>，<code>java</code>，<code>mysql</code>，<code>matlab</code>，<code>javascript</code>，<code>html</code>，<code>css</code>，<code>c++</code>软件的架构中包括了前端，后端，本地集成控制台，硬件，数据库。数据需要在五方进行同步，同时这五方的通信也即为麻烦。因为为了解决不同的项目风险，不得不采用一些平时不常用的语音，这考验了项目组成员快速学习的能力。同时为了解决不同语言的兼容性，又兼顾效率，小组成员将 <code>matlab</code> 编译成了 <code>jar</code> 包，这样提高了软件的效率，同时又便于后人使用。</p> <p>4、软件的版本是需要一次次的迭代的。没有什么软件能够在第一次就做到尽善尽美，如今我们</p>	

的版本也拥有很多不足的地方需要完善。在项目的完成过程中，核心的功能调度算法总共商讨了三个方案，反复确定后才定下如今的调度算法。预测算法准备了两个 level，一个预测不精准但是对数据需求量小，一个预测精准但是对数据需求高。前端的界面拥有两个版本，后一个版本的美化程度明显高于第一个版本，所以任何让客户满意的软件都是经过多次迭代完成的，只有循序渐进才能满足客户的需求。

MGIMSS 第一次迭代评估报告

1、迭代计划中第一次迭代所需要完成任务如下：

- 系统版本 1 (R1) 主要实现技术架构的搭建，具体包括：
 - 硬件接口的连接、数据的传输
 - 调度算法的设计与实现
 - 预测用电量的机器学习模型的搭建
 - 人机交互的设计与识别

2、实际完成情况

- 硬件接口的连接、数据的传输：采用了 B 计划，通过 py 模拟器代替所有的硬件，向系统发送接收数据，其中接口和硬件传感器相同，可以再在之后的版本中升级为硬件传感器。（经济风险）完成度：100%
- 调度算法的设计与实现：自行设计的基于机器学习用电器调度算法，基本实现调度的最优解。完成度：100%
- 预测用电量的机器学习模型的搭建：采用了 Nonlinear autoregressive neural network with external input 神经网络，对于季节性时间序列数据有较好的预测结果，基于我们所掌握的数据和所需的预测精度，我们采用了两个时间预测模型，一个模型预测结果精确，一个模型所需的数据源数据较少。
完成度：100%
- 人机交互的设计与识别：手势的识别采用 face++ 的 webapi，语音的识别采用讯飞 web 的 api，经过简单地 demo 测试。给出了前端的接口。完成度：100%

3、超额完成

- 完成了调度部分的后台实现（包括算法，和模拟器的交互，和数据库的交互）
- 完成了后端所有接口，实体类的设计，数据库的设计，完整的后端框架设计。

- 前端初步搭建完成。

4、迭代计划的修改部分

- 在制定迭代计划时，由于预估不足，第一次迭代没法实现系统测试和获得用户反馈，现在讲该步骤合并到第二次迭代计划之中。

MGIMSS 第二次迭代评估报告

1、迭代计划中第二次迭代所需要完成任务如下：

- 系统版本 2 (R2) 对主要功能进行完善，具体包括：

- 用户界面的设计与实现
- 用电量的可视化
- R1 中各功能模块的链接
- 用户登录以及查询等基本功能

- 迭代一时加入的任务，具体包括：

- 对系统进行系统测试
- 进行调度的评估

5、实际完成情况

- 用户界面的设计与实现：完成页面主题的风格的设置，增修查删用电器，增修查删 job 的界面，用户个人信息的界面。(完成度 90% 存在一些模板没有删除，主页还没有设计好)

- 用电量的可视化：具体可视化的有：（按日按月）查询用电器的用电量，每家用户（按月按年）的总用电量，用户的用电花费以及用电收益，未来发电的预测结果，实时的用电器的电流电压。（完成度 90%，可视化永无止境，未来可以再添加）
- R1 中各功能模块的链接：整合了调度算法，实现完整了的前后端的调度的策略；加入人机交互，前后端加入了 AI 小微，可以实现手势和语音的控制，前端显示预测结果，前端后端 python 模拟器实现通信以及同步。（完成度 90%之后要是先与真正的传感器进行通行）
- 用户登录以及查询等基本功能：实现用户的登录，以及查询的功能，具体情况可以见用户界面的设计与实现（完成度 90%）
- 对系统进行系统测试：初步完成，正常操作不会产生 bug，并且对每个类和函数进行了单元测试（完成度 50%）
- 调度算法的评估：简单的评估，大致效果符合，具体要在之后进行试验（完成度 30%）

Modify	Delete
id:	9887
Appliance:	测试用例
Status:	Pending
Start after:	2018/07/27 08:00
Finish by:	2018/07/27 10:00
Duration:	10
Scheduled at:	2018-07-27 08:59:53
Power:	440

Modify	Delete
id:	9878
Appliance:	风扇
Status:	Pending
Start after:	2018/07/27 08:00
Finish by:	2018/07/27 08:40
Duration:	30
Scheduled at:	2018-07-27 08:08:13
Power:	394

这是一个典型的调度规划：

两个用电器，考虑到尽量多的使用太阳能，我们要两个用电器精良的错开使用，同时，因为 9 点的太阳能比 8 点多的多，所以第二的用电器尽量在 9 开始

使用。以上的规划符合我们预期，如果第二个用电器一开始就运行，需要电费 4814360，按照调度计划执行，需要电费 3977360，节省了 17%的电费，效果显著。

id:

10751

Appliance:

风扇

Status:

Pending

Start after:

2018/07/27 13:00

Finish by:

2018/07/27 15:00

Duration:

30

Scheduled at:

2018-07-27 13:33:10

Power:

395

id:

10780

Appliance:

测试用例

Status:

Pending

Start after:

2018/07/27 13:00

Finish by:

2018/07/27 15:00

Duration:

40

Scheduled at:

2018-07-27 14:04:50

Power:

443

这次测试发现两个任务并不是连续开始的，中间需要给出时间让蓄电池充满，如果连续调度用电器所需电费 8002420，给充足的时间让蓄电池充电，再进行用电器调度我们的电费为 7772770，节约电费 2.8%

id:

11936

Appliance:

测试用例

Status:

Pending

Start after:

2018/07/27 15:00

Finish by:

2018/07/27 17:00

Duration:

30

Scheduled at:

2018-07-27 16:00:03

Power:

443

下面比较了电价和新能源发电情况对用电器的影响，可以发现 15 点电价贵，但是太阳能发电充足，16 点电价便宜但是太阳能发电量少，16 点开始花费电费 1051455，15 点开始花费电费 5756955，节约花费 81.7%

6、超额完成

无

7、迭代计划的修改部分

无

第三次迭代评估报告

评估日期：

组号	15 组	项目名称	微电网智能监控与调度系统
迭代名称	第三次迭代	实际起止日期	7.29---9.1
<p>任务达成情况：（完成的任务、实现的功能、进度、质量等）</p> <p>原迭代计划：</p> <ul style="list-style-type: none">系统版本 3 (R3) 对高级功能进行实现，具体包括：<ul style="list-style-type: none">图像识别、语音识别等复杂人机交互的实现更高级、大规模的数据可视化优化机器学习模型，提高预测准确率进行系统测试，对调度方案进行评估修复缺陷 <p>实际完成：</p> <ol style="list-style-type: none">获取了更多的历史真实数据，对太阳能发电的预测算法进行了优化，加入了干扰项使的预测的结果更加符合真实的情况美化了前端界面，加上了大量的动画效果，修复了前端显示的 bug对系统进行了单元测试系统测试，并且通过了性能测试，压力测试丰富了硬件连接，加入了多元化的传感器（新加入的需求）改进了图像识别和语音识别的精度 <p>完成情况：</p> <ol style="list-style-type: none">（完成度一般）在原迭代计划中，对图像识别以及语音识别改进有限。在语音识别方面已经较为精准，而在图像识别上分析了识别失败的原因，主要为光照不足，因为时间关系没有进一步的改善（在光照充足的情况下识别精度较高）（完成度较高）对于机器学习方面因为获得了大量的真实数据所以进展喜人，预测经过更加的真实可信。（新增）对前端的进一步美化，加入大量动画效果（新增）增加硬件传感器			

问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）

问题：

- 1、神经网络的调参，需要大量的重复试验。
- 2、对于硬件的编程没有基础，需要重头开始学习
- 3、前端动画的不兼容性

变更：

- 1、见上一项

返工：

无

经验和教训：

- 1、这一次的迭代都是需要花比较长时间才能出成果的，特别是对神经网络的调参，因为调参基本上靠经验，所以找到适合的参数比较困难，需要一遍遍反复的尝试，需要耐得住性子。
- 2、对于硬件的调试，特别是体感传感器的调试，我们花费了大量的时间，但是功夫不负有心人，最后我们得到的结果非常令人满意，如果中途放弃了，我们就缺少了项目的其中一个亮点。

第四次迭代评估报告

评估日期：

组号	15 组	项目名称	微电网智能监控与调度系统
迭代名称	第四次迭代	实际起止日期	9.1---9.10
<p>任务达成情况：（完成的任务、实现的功能、进度、质量等）</p> <p>原迭代计划：</p> <p>安装 R3，撰写用户手册，系统上线运行，并根据系统运行反馈进行缺陷修复与改进，准备验收与答辩</p> <p>实际完成：</p> <p>1、修复大小 bug，保障系统的可靠性</p> <p>2、撰写测试报告，项目总结报告，迭代评估报告，测试用例，预测算法报告，调度算法报告。</p> <p>3、制作演示视频，ppt，用户手册</p> <p>完成情况：</p> <p>1、系统测试和缺陷修复基本完成（余下少量未修复但不影响功能的小 bug）</p> <p>2、修改了部署至服务器，改为本地部署（项目需求不高）</p> <p>3、完成相关文档的撰写</p>			
<p>问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）</p> <p>问题：</p> <p>1、系统并不兼容所有的浏览器，主要原因是不是所有的浏览器都拥有调用及其摄像头和麦克风的接口。</p> <p>变更：</p> <p>1、部署到本地非服务器（部署服务器代价太高，且需求低）</p> <p>返工：</p> <p>无</p>			

经验和教训：

- 1、第一次较为全面的进行系统的测试，包括单元测试，系统测试时，算法测试，兼容性测试，性能测试，通过测试我们发现了很多 bug，有些影响系统等功能，有一些不影响功能但会降低用户体验，测试能够很好的帮组我们评估我们的做的系统的质量，每一次的测试结果都是下一次迭代开发的基础
- 2、作为一个用户体验良好的系统,需要保证软件的易用性。在准备演示 ppt 以及视频的时候，我们需要考虑如何在最短的时间内将我们软件的特点告诉给用户，让用户快速上手使用我们的软件，另外良好的用户手册是提高软件的易用性的良好手段。
- 3、做一个大型的项目需要善始善终，不能辛辛苦苦写代码，草草率率收工演示，演示和编程过程同样重要，只有获得了用户的认可，软件才能获得最终的成功。