

嵌入式大作业笔记

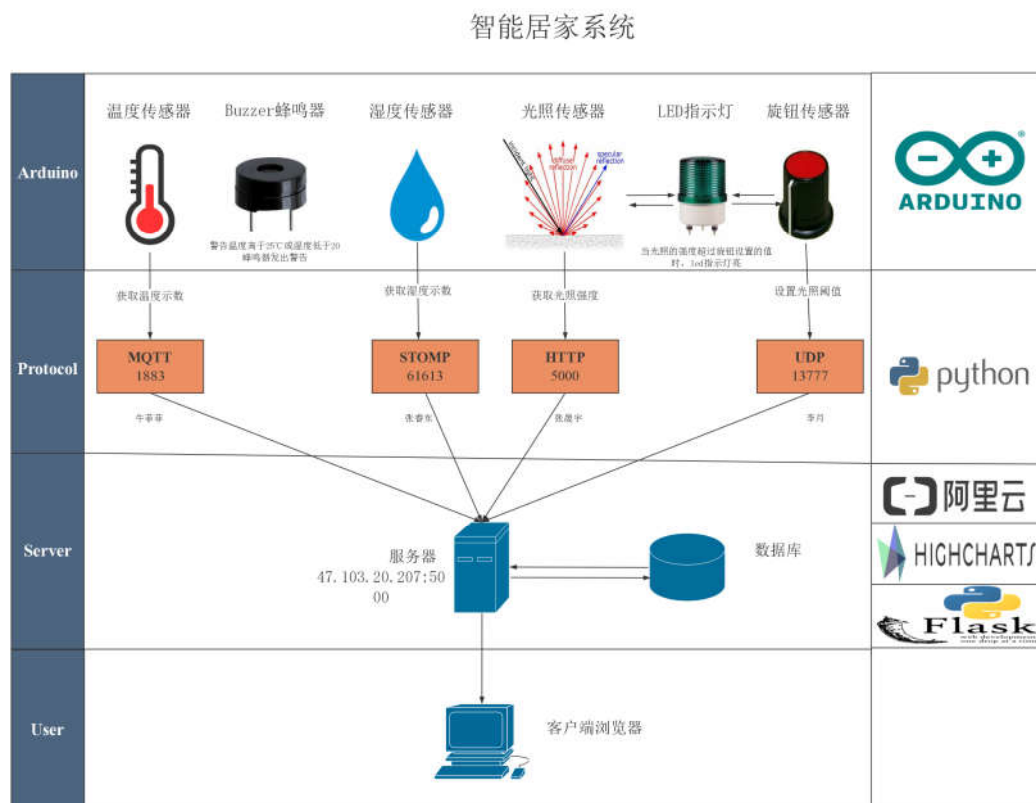
笔记本： 软件复用

创建时间： 2019/3/22 19:18

更新时间： 2019/3/30 15:59

作者： 牛菲菲

IOT通讯协议实现--智能家居系统



Team_24

小组成员及任务分配:

牛菲菲 DZ1832003 嵌入式板子编码, 实现嵌入式功能, 接收传感器数据, web service 开发, MQTT协议实现

张晟宇 MF1832241 HTTP协议实现

李月 MP1832019 UDP协议实现+系统架构图绘制

张睿东 MF1832240 STOMP协议实现

系统功能--《智能家居》

1. 温湿度传感器测量室内温度湿度

当温度超过25或者湿度低于20时, 蜂鸣器响(state=1)

2. 光照传感器检测室内光线, 并通过旋钮传感器允许用户手动设置光线的阈值, 当光线强度低于旋钮设置阈值时, led灯亮(state=1)

涉及传感器:

1. 温湿度传感器
2. 蜂鸣器传感器
3. 光线传感器
4. 旋钮传感器

5. LED灯传感器

实现技术：

1. 使用arduino对板子进行编码，实现系统功能以及边缘计算，将数据通过serial端口发送至电脑
2. 使用python serial接收端口数据，并使用四种协议MQTT, HTTP, STOMP, UDP 分别将温湿度，光线，旋钮四个数据发送至服务器的四个端口
3. 服务器端使用对应的协议接收数据并存储至数据库
4. 服务器使用python flask框架进行编写web service，前端采用high chart展示折线图
5. MQTT协议及STOMP协议基于ActiveMQ docker将数据接收并转发
6. 服务器访问地址：47.103.20.207:5000（考试周结束之前均可访问）

作业提交列表：

1. AnalogReadSerial: 基于Arduino的嵌入式板子端代码
2. Communications Protocols: 包含MQTT, HTTP, STOMP以及UDP四种协议的发送数据以及服务器接收数据的代码，其中HTTP服务端接收数据代码位于flask网站代码中
3. flask: 基于python flask的web service代码
4. 网站截图
5. serial_reader: 基于python接收电脑serial端口数据，并将数据通过协议的代码函数发送至服务器
6. 框架图：本次作业整体的实现架构

以下为本次作业经验以及采坑笔记：

python

收

python接收serial端口数据

python安装serial，以及pyserial包，然后接收端口数据

发

将接收的数据，通过四种协议传送至服务器的各个端口

mqtt->温度 1883

stomp->湿度 61613

http->light path:5000/sender/light

udp->rotation 13777

服务端：

阿里云服务器

1.安装python3.6，系统自带python2，自己装python3

2.添加python3的环境变量

python3安装路径为/usr/local/python3.6

3.使用pip安装包，pip3 install paho-mqtt（pip3安装后路径为python3.6/bin/lib/site-packages）

服务器安装activemq：

activemq支持多种协议，

协议与端口：

ws:61614

mqtt:1883

stomp: 61613

amqp: 5672/ 5671

openwire:61616

首先安装jdk，<https://www.cnblogs.com/stulzq/p/9286878.html>

然后安装activemq，直接下载解压，然后启

动 <https://www.jianshu.com/p/9e5f2057cb9c>

前台进程启动ActiveMQ

```
./bin/activemq console
# 后台进程启动ActiveMQ
./bin/activemq start

# 停止ActiveMQ
./bin/activemq stop
```

启动后，可以访问查看连接状态<http://47.103.20.207:8161>

各种协议可能涉及端口的打开问题

放开端口：https://blog.csdn.net/qg_24232123/article/details/79781527

开启防火墙，*# systemctl start firewalld 没有任何提示即开启成功*

开启防火墙 *# service firewalld start*

关闭防火墙 *# systemctl stop firewalld*

查看想开的端口是否已开 # firewall-cmd --query-port=666/tcp 提示no表示未开

开永久端口号 firewall-cmd --add-port=666/tcp --permanent 提示 success 表示成功

重新载入配置 # firewall-cmd --reload 比如添加规则之后，需要执行此命令

再次查看想开的端口是否已开 # firewall-cmd --query-port=666/tcp 提示yes表示成功

若移除端口 # firewall-cmd --permanent --remove-port=666/tcp

```
firewall-cmd --zone=public --add-port=8161/tcp --permanent
```

http协议实现

使用python的request，基于网站的/sender/light路径进行传输数据

udp协议实现：

注意服务端监听要是""，监听所有，不能监听本地，localhost

开启对应的UDP端口

firewall-cmd --add-port=13777/udp --permanent

mqtt

开启端口 1883

安装EMQ，写客户端订阅和发布

emq启动：cmd -> D:\emqtttd\bin> .\emqtttd.cmd start(stop 停止/status状态)

在服务器端安装EMQ

启动EMQ

./bin/emqtttd start (path: ~/emqtt/)

python中pip install paho-mqtt包

stomp可以通过activemq进行通信

安装stomp.py

未实现：

xmpp:

install xmpppy

AMQP协议实现，

在服务器上安装rabbitMQ，yum installxxx

运行AMQP_1，报错缺少_ssl,重新引入ssl: <https://www.jianshu.com/p/3ec24f563b81>

重新配置python环境后，在Py36文件夹中存在python环境，需要进入文件

夹，./bin/python36 amqp/AMQP_1.py 进行运行

xx找activemq中的example，安装python-qpid-proton

COAP

安装包

aiocoap

CoaPthon

CoaPthon3

网站

python+flask

flask本地运行通之后，在服务器上部署flask环境：

<https://zhuanlan.zhihu.com/p/21262280>

服务器上新建flask项目，运行：gunicorn -w 4 -b 0.0.0.0:5000 app:app

python安装flask后启动报错：没有_ssl,重新安装gunicorn环境，重新运行就可以