Project Document

Smart home

A raspberry pi + mobile app based

control mechanism

COMP3432 Innovative Computing Paradigm

Term project

Submitted by

GENG XU Philip

12132031d

2015/4/7

Table of Content

1. Submission materials

2. Project lifecycle and milestones

3. Raspberry pi configuration

4. Server configuration

5. Mobile App design

6. Hardware connection

1. Submission materials:

a. Source Code:

Raspberry pi: residential program **./raspberry pi/light.py**

Run it : sudo python light.py

Mobile App: android project under folder **Smarthome**

Server:

code: under folder **server (lighton/off/status.php status.txt)**

also under url folder : [www.comp.polyu.edu.hk/~12132031d/comp3432](http://www.comp.polyu.edu.hk/~12132031d/comp3432)

b. Demo use:

ppt: **smarthome.pptx**

Contains problem and solution, architecture, mechanism and analysis, which won’t be covered in this document

Video: url <https://www.youtube.com/watch?v=Cxu-oAttzS8>

c. This project report

2. Project lifecycle and milestone:

|  |  |
| --- | --- |
| Stage 1. Idea and infrastructure | |
| Idea initiation | Feb 5 |
| Rasp-pi configuration and program | Feb 6-14 |
| Server program | Feb 8 |
| Stage 2. Control implementation | |
| Sina weibo control | Feb 21-22 |
| Mobile app design and button control | Mar 1-10 |
| Fingerprint localization with dynamic model training | Mar 12-Apr 1 |
| Stage 3. Hardware integration | |
| Purchase and welding | Mar 25-27 |
| Integration success. Video and ppt making | Mar 28-30 |
| Stage 4. Recover from damage | |
| Rasp-pi broken | Mar 29 |
| SD card broken | Apr 6 |
| New rasp-pi config and re-write python program | Apr 6 |
| Integration success | Apr 6 |
| Stage 5. Finalize |  |
| Project document finish | Apr 7 |
| Project finalized | Apr 7 |

3. Raspberry pi configuration

a. System requirement

The application runs well both on:

|  |  |
| --- | --- |
| Pi model type | OS version (debian of raspberry pi version) |
| Raspberry pi B | Wheezy Raspbian 2014-06-20 |
| Raspberry pi B+ | Wheezy Raspbian 2015-02-16 |

(2014-06-20 version don’t run on B+ because of different hardware architecture)

b. Configuration:

1. install python pip and python dev

(sudo apt-get update)

sudo apt-get install **python-pip**

sudo apt-get install **python-dev**

2. install python library for sina weibo and GPIO for raspberry pi

sudo pip install **sinaweibopy**

sudo pip install **rpi.gpio**

c. Run:

Sudo python light.py

d. Program details in pseudocode:

pass sina weibo authentication using

APP\_KEY, APP\_SECRET, USERID,PASSWORD

While true:

Every 10 second:

Read latest sina weibo post and analyze

(synchronization issue)

Change light status by GPIO.output

Every 10 second:

Read csdoor server information and analyze

(synchronization issue)

Change light status by GPIO.output

4. Server configuration

Use csdoor server of department of computing

a. lighton.php

re-write status.txt to “light on”

b. lightoff.php

re-write status.txt to “light off”

c. lightstatus.php

Check the content of status.txt

5. Mobile App Design

Test platform: HTC T328w

a. System requirement:

min android sdk version : 15

target android sdk version : 18

network connected

b. Project overview

Outlooking and frame:

MainActivity.java -- Activity\_main.xml

ObjectDrawerItem.java -- Listview\_item\_row.xml

DrawerItemCustomAdapter.java

Share data:

Data.java

Supporting class:

httpRequest.java (send http request to server)

WifiAdmin.java (detect wifi environment)

Button control fragment:

ButtonFragment.java -- Fragment\_button.xml

Auto control fragment (fingerprint localization)

AutoFragment.java -- Fragment\_auto.xml

Training fragment (training fingerprint model)

TrainFragment.java -- Fragment\_train.xml

c. button control function

outlook:



Manipulation:

Press the toggle button to turn on/off

Mechanism:

Use httpRequest.java to send out turn on/off message to background server to control the light

d. Auto control fragment

outlook:



Manipulation:

Item 1 is a list of all available models (fingerprint database). User can select one and click the start button. The wifi list information, indoor-outdoor judgement and light status will be shown.

(These information are for the convenience of developers only. For normal users, keep it running and put your mobile phone in your pocket)

Mechanism:

The model is stored in the Data.java.

Use wifiAdmin.java to detect wifi environment and compare it with the fingerprint model to decide whether the user is inside the door or outside door. Then, use httpRequest.java to send out turn on/off message.

Judgement of indoor/outdoor:

Get wifi list every 2 second

Calculate the euclidean distance with both the indoor and outdoor part of the model

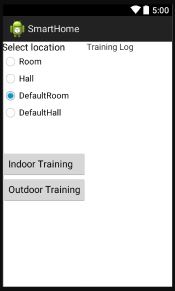
If outdoor<indoor and outdoor<threshold value, it is indoor.

Else it is outdoor.

If have 3 consecutive same answer different from current light status, send httpRequest to change it.

e. Training fragment

outlook:



Manipulation:

Select a location – a name that you want to store the model

Go indoor, press indoor training and hang around for 20 seconds

Go outdoor, press outdoor training and hang around for 20 seconds

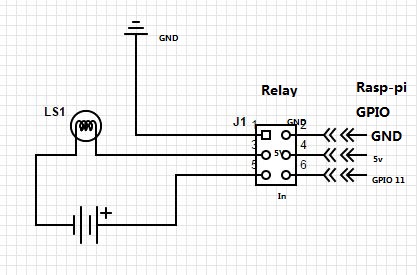
Wifi list will be shown on the right.

Mechanism:

Use Wifiadmin.java to get the wifi list

After training of each part, calculate the average and store into Data.java

6. Hardware Connection



Appliance circuit:

Bulb: 12v 1w

Power source: Batteries 1.5v \* 6 = 9v

Relay – raspberry pi connection

|  |  |
| --- | --- |
| Relay input | Rasp-pi output |
| 2 – GND | GND (port 3) |
| 4 – 5V POWER | 5v (port 0) |
| 6 – IN control signal | GPIO 11 |

Relay high-level trigger: 3-5 inter-connected.