北京郵電大學

Queen Mary
University of London

Undergraduate Project Report
2016/17

[Design and Implementation for customized mobile HTML5 template for end users]

Name:	[Hong Yangkai]
Programme:	[Telecommunications]
Class:	[2013215106]
QM Student No.	[130800662]
BUPT Student No.	[2013213085]

Date [15/05/2017]

Table of Contents

[Design and Implementation for customized mobile HTML5 template for end users]

Abstract

This project develops a smart home app that can control devices. The app mainly contains two templates, safety template and wakeup template. A template is a series of predefined operations on several devices. Users need only one key to open or close the template without having to make too many choices. Additionally, the app has a device page for users to control devices individually.

The app is developed using Cordova framework. Cordova is a HTML5 Hybrid Mobile App Framework. It combines the advantages of HTML5 app and native app. It enables the app to cross platform. At the same time, rich native APIs can be invoked with JavaScript.

For safety template, it realizes the functions of remote monitoring and remote switching alarm device. The development of safety template is divided into two parts: hardware part and app part. The hardware part mainly uses raspberry pi and related accessories. The raspberry pi automatically runs python script as web server. The safety template control page is stored on the server and the app uses a Cordova plugin InAppBrowser to access it. On the control page, click the start button, the server runs a python script to open the device. Click the stop button, the server kills the process of the script. After starting the template, users can click on the camera button to enter the monitor page.

For wakeup template, it realizes the functions of waking users up and preparing breakfast for users via Bluetooth control devices. Users only need to set a start time. At that time, Bluetooth audio plays music, Bluetooth curtain opens, Bluetooth socket for rice cooker supplies power. However, because of lacking funds and hardware, this project only has Bluetooth audio for testing.

Keywords: HTML5, Cordova, smart home

[Design and Implementation for customized mobile HTML5 template for end users]

摘要 (Chinese translation of the Abstract)

这个项目开发了一个可以控制设备的智能家居应用。这个应用主要包含了两个模板，安防模板和叫醒模板。模板指的是对几个设备进行一系列预定义好的操作。用户只需一键开启或结束模板，无需做过多的选择。另外，用户也可以在设备页对某一设备进行单独控制。

这个应用是用 Cordova 框架开发的。Cordova 是一个 HTML5 移动应用混合开发框架。混合开发结合了 HTML5 应用和原生应用的优点。它使得我的应用可以跨平台，同时又可以通过 JavaScript 调用丰富的原生 API。

安防模板实现了远程监控和远程开关警报装置的功能。安防模板的开发分为两部分：硬件部分和手机应用部分。硬件部分是用树莓派及其相关组件做的。树莓派开机自动运行一个 python 脚本成为一个 web 服务器。安防模板控制页面保存在服务器上，我的应用通过使用 Cordova 的一个名为 InAppBrowser 的插件来访问控制页面。在控制页面上，点击开始按钮，服务器会执行一个 python 脚本来开启警报装置；点击结束按钮，服务器会终止这个脚本的进程，从而实现远程开关。开启模板后，用户可以点击摄像头按钮进入监控页面。

叫醒模板实现了通过蓝牙控制设备来叫醒用户并为用户准备好早饭的功能。用户只需设定一个开始时间。到那时，蓝牙音箱播放音乐，蓝牙窗帘打开，电饭煲的蓝牙插座供电。然而由于缺乏资金和硬件，此项目只有蓝牙音箱供测试。

关键词：HTML5，Cordova，智能家居

Chapter 1: Introduction

1.1 Project

1.1.1 Motivation

Smart home is popular now. And safety system is an important constituent of the smart home. Additionally, smart home automation can greatly facilitate people's lives. So this project developed a smart home app with templates and realised the function that one button allows the device to operate automatically.

The next problem is to decide the application development mode. Web technology, including HTML, CSS and JavaScript, has the characteristic of crossing platform. It means developing once, then the project can run on different platforms, such as Android, IOS, Windows Phone and so on. However, pure web-app has many limitations (e.g. less API). So this project combined web-app and native-app by developing the app in hybrid mode using Cordova framework. By developing in hybrid mode, the app can not only cross platform but also use many native APIs.

1.1.2 Functionality to realise

According to the specification, this project should be realised with two templates, which is the core.

This project intends to implement safety template and wakeup template. For every template, software and hardware interact with each other. For safety template, suppose user is leaving and no family at home, then user can click the button in the app to start the safety scene. In fact, this operation turns on an infrared sensor. The infrared sensor should be set by the door. If the infrared sensor detects someone, it must be thief. Then the warning light turns on and buzzer rings, the hardware also sends a message to user's phone. As for wakeup template, users can set time. At that time, the Bluetooth audio plays music, the bedside lamp turns on, the socket for electric rice cooker supplies power. User can be waked up by the automatic scene and enjoy prepared breakfast.

1.1.3 Technical context

The project is based on HTML5 and Cordova to implement a hybrid phone app. Owing to the app is about smart home template. The project also contains some hardware development.

[Design and Implementation for customized mobile HTML5 template for end users]

Software requirement:

a. NodeJS and NPM

   NodeJS is the platform that Cordova development requires.

b. Android SDK

   For Android app development, an Android SDK on computer is indispensable.

c. XCode

   IOS app developing need a XCode on Mac.

d. Git

   Although git may not be used directly, it should be installed. Because Cordova uses some background processes of it.

e. Cordova

Hardware requirement

a. Bluetooth audio

b. Raspberry Pi

1.2 Report structure

a. The first part is abstract with both English and Chinese version. It aims to show the problem that this project tries to solve, the approach to solve the problem, the results obtained in this project and the impact of obtained results.

b. The first chapter is introduction. It describes the motivation to do this project, primary functionalities the templates should realise and a brief description of technical context. In addition, development environment, including software and hardware requirement, is mentioned as well.

c. The second chapter is background. It shows all the relevant background information on the project such as HTML5, JavaScript and Cordova.

d. The third part is design and implementation. It is the most important part of this report and it includes the process to build two templates and phone app.

e. The fourth chapter is results and discussions. As for implementation, the phone app needs testing by asking some users to use it. Also, feedback is an important information for me to improve the app.

f. The fifth chapter is conclusion and further work. It made a summary, shows what the project has done and valuable experience. Then it also analysed the weakness of the work. Based on the weakness, this project proposed the further work.

g. Attached are references, acknowledgement, appendix, risk assessment and environment impact assessment.

Chapter 2: Background

2.1 HTML5

2.1.1 HTML

HTML is a language to describe a web page. The full name of HTML is Hyper Text Markup Language. HTML is not a programming language, but a markup language. Markup language is a set of markup tag. HTML use markup tag to describe a web page. HTML document contains HTML tags and text content. HTML document is also called web page.

HTML5 is the latest version of HTML, made by W3C. It aims to support multi-media on mobile device. It has many improvements, such as new elements, new attributes and supporting CSS3. Using HTML5, video and audio can be played in a simple page. One important improvement is that web app can be easily developed using HTML5.

What follows is an example of html document.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=”utf-8”>
<title>Final Report</title>
</head>
<body>
    <h1>Project IP_3085</h1>
    <p>Design and implementation HTML5 mobile template for end users</p>
</body>
</html>
```

Analysis of this example:

<!DOCTYPE html> states it is an HTML5 document.

<html> is the root element of the document.

<head> contains the meta data of the document.

<title> describe the title of the document.

<body> contains the visible content of the document.

<h1> defines a big header.

<p> defines a paragraph.

2.1.2 CSS

CSS refers to Cascading Style Sheets. Style defines how to display the HTML elements. Style is usually stored in style sheet. External style sheet can extremely improve work efficiency. It is usually stored in .css file. Style solves a big problem. Editing a .css file enables changing the layout and appearance of all pages in the site. All modern browsers support CSS.

CSS3 is the latest version of CSS. CSS3 can be divided into some modules, selector, box model, background and border, text effect, 2D/3D transform, animation, multiple column layout, user interface.

What follows is an example of CSS.

```
body{
        background-color: #d0e4fe;
}
h1{
        color: orange;
        text-align: center;
}
p{
font-family: “Times New Roman”;
font-size:20px;
}
```

2.1.3 JavaScript

JavaScript is a popular scripting language on the Internet. It can be used for HTML and web, still more for server, PC, laptop, pad and phone. JavaScript is a lightweight programming language. It can be inserted into the HTML document. After inserting the JavaScript code, HTML document can run on every modern browser. Best of all, JavaScript is relatively easy to learn. JavaScript can do a lot. For example:

Write HTML output stream directly.

```
document.write(“<h1>This is a header</h1>”);
```

Response to events.

```
<button type=”button” onclick=”alert(‘welcome!’)”>Click me</button>
```

Change the content of HTML.

```
x=document.getElementById(“demo”);
x.innerHTML=”Hello JavaScript”;
```

Change HTML image.

```
function changeImage()
{
    element=document.getElementById('myimage')
    if (element.src.match("bulbon"))
    {
        element.src="/images/pic_bulboff.gif";
    }
    else
    {
        element.src="/images/pic_bulbon.gif";
    }
}
```

Change HTML style.

```
x=document.getElementById(“demo”);
x.style.color=”#ff0000”;
```

Validate input.

```
if isNaN(x) {alert(“Not a Number”)};
```

As a matter of fact, JavaScript is very different from Java, whether in concept or in design. Java is more complicated. The standard official name of JavaScript is ECMAscript. ECMA6 is the latest version.

2.1.4 jQuery

jQuery is a library of JavaScript. It simplifies the JavaScript programming extremely.

jQuery can be added to the HTML document by a simple line of code. In general, there are two ways to import it. First kind, local import. Download a version of jQuery from jquery.com. There are two versions provided, development version and production version. Development version is in the form of .js, it has notes and space so it is easy for developers to read source code. The production version is in the form of .min.js, it is compressed and used for actual website. However, they have the same functions, so putting either one into the project js folder is ok. Then write the line below in the <head> of HTML document to import jQuery.

```
<script src=”./js/jquery-3.2.1.min.js”></script>
```

Second way, import in CDN, such as Google, Baidu and Sina. For example,

```
<script src="https://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"> </script>
```

jQuery is a library of JavaScript which helps programmer write less and do more. It contains many functions, including selecting HTML element, HTML DOM operation, CSS operation, HTML events function, JavaScript special effects and animation, AJAX and Utilities. Additionally, jQuery also provides many plug-ins, such as range, progress bar and so on. They are very useful for application development.

2.2 Hybrid app

Hybrid app is between web-app and native-app. It has good user experience as native app. At the same time, it has the advantage of crossing platform as web app. As a matter of fact, some mainstream mobile applications in the market are based on hybrid. For instance, Facebook, Baidu search Android client-side, Taobao Android client-side.

The official definition of Hybrid app is that it is a mobile application that is coded in both browser-supported language and computer language. Overall, hybrid app is closer to native app. But hybrid uses web languages meanwhile, so the development cost and difficulty are much less than native app.

According to the mixing degree of web language and native language, hybrid is usually divided into three types: Multiple views mixed type, Single view mixed type and web subject type. This project chose web subject type. This type is mainly programmed by web languages, using JavaScript to call some native API. Relatively speaking, application developed in this type has few defects in user experience, but the overall difficulties of development greatly reduce, and it can achieve crossing platform basically. The user experience of web subject application mainly depends on the bottom middleware's ability of interaction and crossing platform. There are many kinds of web subject type application middleware, such as appMobi, PhoneGap, Cordova, WeX5, AppCan, Rexsee and so on. In fact, excepting small problem in user experience, web subject type is the best solution of hybrid app. This project chose Cordova as the framework since it has many documents online. It is more convenient for me to study.

2.2.1 Cordova

Cordova is a framework for using HTML, CSS and JavaScript to build a mobile app, connecting web app to mobile native API. By default, if Cordova is not used, android API can only be called with Java code. But Java is much more complicated and cannot be inserted into HTML document. So enabling JavaScript to call native API is the thing Cordova do. It provides a bridge between web app and mobile native API. For example, mobile camera, geographical location, file and other functions can be called through JavaScript.

The official document defines Cordova as it is an open source mobile development framework. It avoids native language for each mobile platform, for instance, Java for Android, Objective-C for IOS. The application executes within the wrapper for each platform, and rely on the standard API

binding to access the sensor, data and network status of each device.

Cordova has advantages and disadvantages. As for advantages, Cordova provides a platform for building hybrid mobile applications. Theoretically, one code can be used on different mobile platforms, IOS, Android, Windows Phone, Amazon-fireos, blackberry, Firefox OS, Ubuntu and tizien. Secondly, developing a hybrid app is faster than native app, so Cordova can save a lot of development time. Thirdly, since Cordova uses JavaScript, this project does not need to use the platform specific programming language. Last, a large number of community plug-ins can be used with Cordova. Many libraries and frameworks are optimized for use.

However, Cordova has some limitations. For instance, hybrid app is slower than native app, so it is not the best choice to use Cordova for large applications that require large amounts of data and functionality. Secondly, cross browser compatibility may cause a lot of problems. When build applications for different platforms, testing and optimization can take a lot of time because it need to cover a large number of devices and operating systems. Another important limitation is that some plug-ins have compatibility problems with different devices and platforms. Additionally, there are some native APIs do not support Cordova.

For this project, it is not large so Cordova does not show performance defect. In addition, since this project only developed the android version and android supports Cordova greatly so it fits the project well.

2.3 Smart hardware

Since this project is design and implementation a smart home mobile application. There is no doubt that it need to be able to connect and control some intelligent hardware. This project used two ways to connect device, Bluetooth and Wi-Fi. After connecting, the app can send messages to control the devices.

Chapter 3: Design and Implementation

3.1 Design

This smart home application requires to implement two templates, safety template and wakeup template. Template refers to a series of well-defined operations. For example, user click a button to execute the template, then the written function A starts. After that, the output of function A is the input of function B and starts function B. And so on, until one function to stop the template.

3.1.1 Safety template

Safety template in the app aims at guarding against theft. When user and family are not at home, the thief may come to user's house. In this case, user need to know as quick as possible and call police. I think firstly, a sensor at home should detect someone closing. Then a warning light turns on and a buzzer rings. It has deterrent effect to thieves. At the same time, a camera should take a picture of the thief. Then the sensor or some other device should send a message and the picture to user. The picture can help user confirm that there is really a thief. Then user can call the police.

It makes sense, but there is another problem. User would not like the sensor always working. Because it cannot distinguish who is closing, so it will recognize user as a thief. Thus, the app should be able to turn on or off the sensor remotely. Figure1 shows the safety template flow chart.

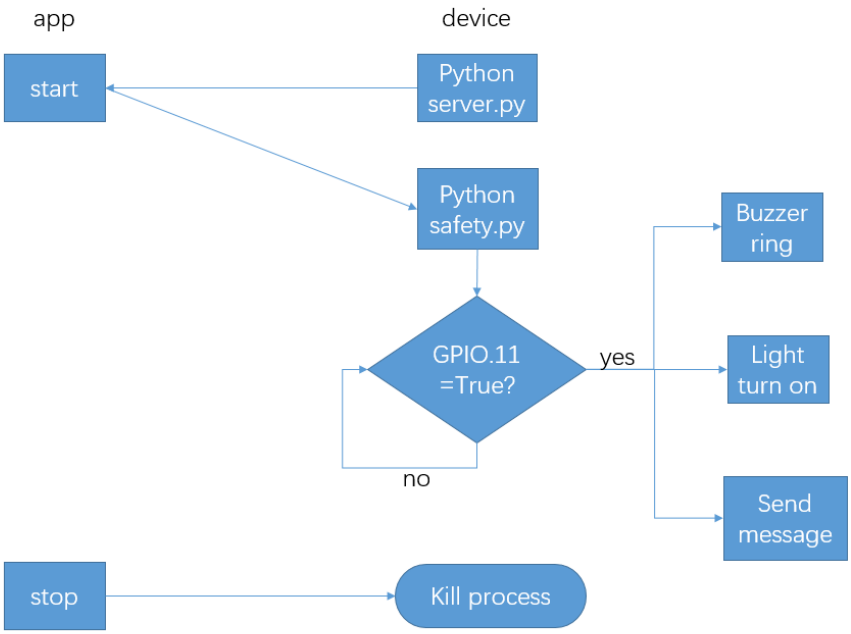

Figure 1 Safety template flow chart

3.1.2 Wakeup template

Wakeup template aims to wake user up and prepare a breakfast for him. When user starts the template, the app opens user's mobile Bluetooth and connects mobile to devices (audio, curtain and socket for rice cooker). Then user need to set a wakeup time. At that time, the audio plays music stored in user's phone. The bedside lamp turns on to help waking user up. And the socket starts supplying power for electric cooker. The socket should be switched off and the electric cooker should be set work status before user went to sleep. Figure2 shows the flow chart.

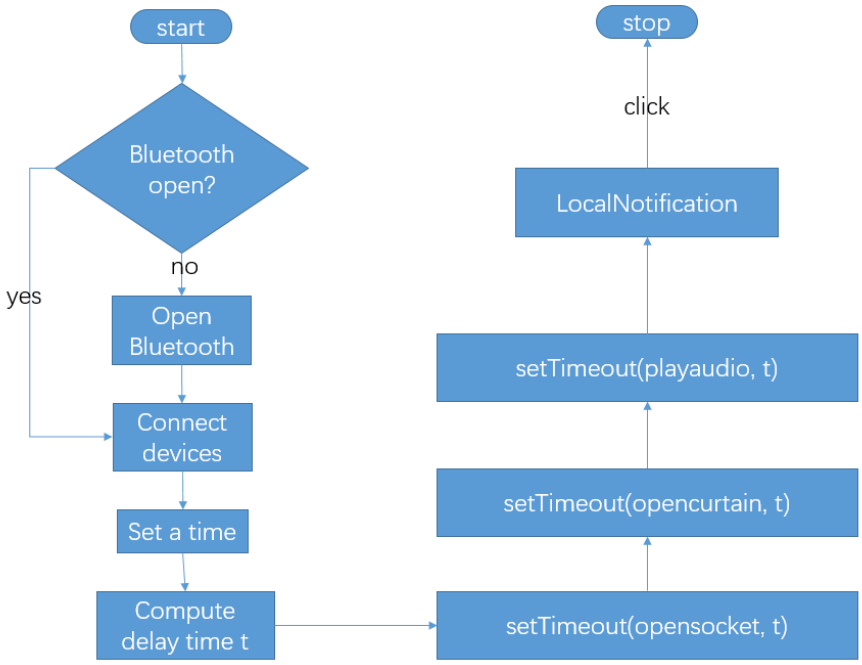

Figure 2 Wakeup template flow chart

3.2 Implementation

3.2.1 Safety template

3.2.1.1 Hardware part

I mainly used this hardware: Raspberry pi3, buzzer module, led light, camera module, infrared module, bread board, GPIO expansion board(optional) and some related accessories.

Assemble the raspberry and standard accessories, connect the bread board and the expansion board, as shown in Figure3.

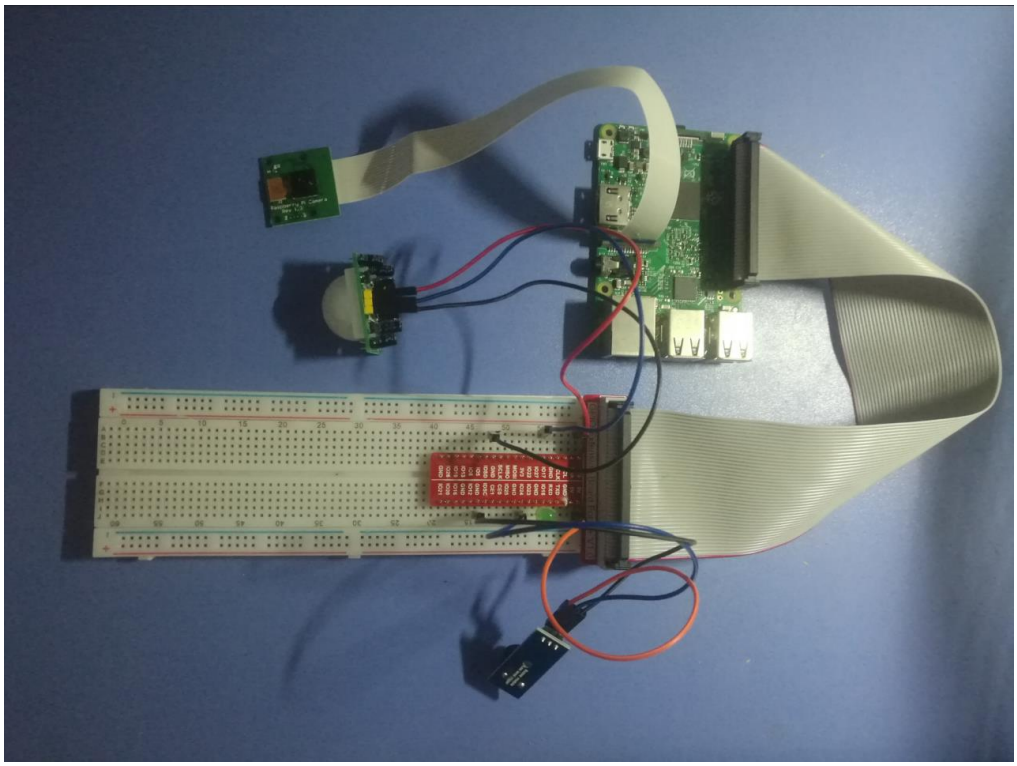

Figure 3 Alarm device

First of all, I need to introduce the Raspberry GPIO (General Purpose Input/Output). Raspberry pi is equipped with a 40 pin GPIO interface for implementing rich digital input and output functions. As shown in figure4 (JRZJ, 2016).

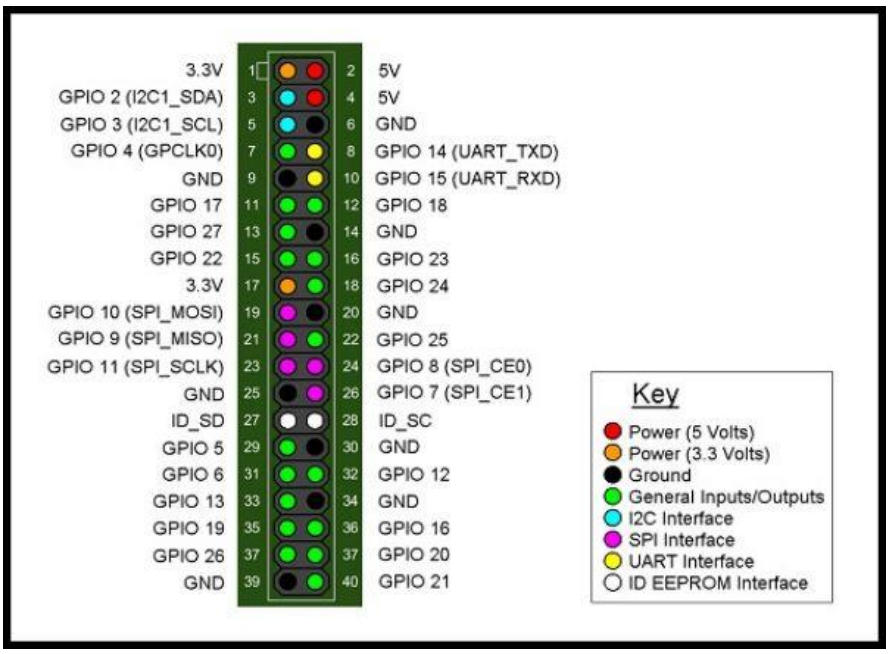

Figure 4 Raspberry GPIO

In this project, I connected the long pin of LED light to NO.12(GPIO 18) channel, short pin to NO.14(GND) channel. For infrared module, I connected its VCC pin to NO.1(3.3V) channel, GND pin to NO.25(GND) channel, OUT pin to NO.11(GPIO 17) channel. As for buzzer module, I connected its VCC pin to NO.2(5V) channel, GND pin to NO.30(GND) channel, I/O pin to NO.18(GPIO 24) channel. The camera module is connected to the CSI interface of raspberry pi. Up to this point, I have finished assembling hardware.

The next step is programming the hardware. First of all, I connected the raspberry pi to my computer with a cable. Then it got a local area network IP address. I inputted “arp –a” in DOS window to get the IP address, it was 192.168.137.188. Then I SSH the IP address of raspberry using a software named PuTTY. After inputting the default account name “pi” and default password “raspberry”, I logged into the raspberry system. I installed a Debian system, a kind of Linux, for the raspberry pi before the start.

I decided to control the hardware by using python because it is the most common way. So I create a new python file called “safety.py” by input “vim safety.py”. Before this, I installed vim. Vim is a powerful text editor in Linux. At first, I did not have vim and I used the default text editor vi. It was no use for me at all. The up and down key is not “up” and “down” and I cannot even use backspace key to delete any character. So I think a vim is necessary for text editing.

After “vim safety.py”, I started to write the code. First, infrared module need to detect people and output information. So I write this code.

```
import RPi.GPIO as GPIO
import time
def init():
        GPIO.setwarnings(False)
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(11,GPIO.IN)
def detect():
        for i in range(1,100):
                if GPIO.input(11) == True:
                        print “Someone is closing”
```

```
                else:
                        print “Nobody”
        time.sleep(2)
time.sleep(5)
init()
detect()
GPIO.cleanup()
```

Interpretation of the code: RPi.GPIO is a module to control Raspberry Pi GPIO channels. So import it firstly. Since I do not want to detect only once, I need to import time to enable the infrared module detect once every second. GPIO.setwarnings(False) is optional, I did not want to see the warnings, so I wrote this. GPIO.setmode(GPIO.BOARD) means natural numbering (left channels are 1-39, right channels are 2-40). It is convenient for programming. GPIO.setup(11,GPIO.IN) sets the output of infrared module to be the input of other modules since I will add buzzer module later. detect() function defined 100 cycles. GPIO.input(11) equals to true means someone is closing. So print it. time.sleep(2) let the detect() function executes once every two seconds. time.sleep(5) is optional, it makes the entire script execute after five seconds. GPIO.cleanup() is important and can not be ignored. It would restore the default GPIO state after the script finished.

The programming of infrared module ended, then I programmed the buzzer module. The infrared module is input, the buzzer module should be output. So add a line to def init():

```
GPIO.setup(18,GPIO.OUT)
```

Add a new function to let the buzzer work.

```
def beep():
        while GPIO.input(11):
                GPIO.output(18,GPIO.LOW)
                time.sleep(0.5)
                GPIO.output(18,GPIO.HIGH)
                time.sleep(0.5)
```

I need a condition to execute the beep() function, so modify the code of detect() function.

```
if GPIO.input(11) == True:
        print “Someone is closing”
        beep()
else:
        print “Nobody”
        GPIO.output(18,GPIO.HIGH)
```

GPIO.output(18,GPIO.HIGH) for shutting the buzzer down when detects nobody. There is nothing wrong with it because the buzzer module works at low level and shuts down at high level.

Up to this point, the infrared and buzzer modules have been finished. Additionally, I want to add a LED light to assist the buzzer to warn thief. Also add some code to def init():

```
GPIO.setup(12,GPIO.OUT)
```

Modify the detect() function.

```
if GPIO.input(11) == True:
        GPIO.output(12,GPIO.HIGH)
else:
        GPIO.output(12,GPIO.LOW)
```

The infrared, buzzer, LED has been finished. However, detecting and warning thief are not enough, the device should let the user know that someone has broken into his house. So I achieved a sending message function. When the infrared module detects somebody, the raspberry pi sends an e-mail to me. I chose to send e-mail because it is free. I wanted to send a short message at first, but I need to buy a SMS module. So I give up the idea. I used SMTP (Simple Mail Transfer Protocol). The protocol can be used to receive and send messages with a short code. Python provides a smtplib library for SMTP operations. So I can add the e-mail function to safety.py file. First of all, import the library.

```
import smtplib
```

Then I need to set the sender and receiver. I have two e-mail boxes. 163 box is the sender, QQ box

is the receiver. Because when I receive a QQ e-mail, my WeChat app will inform me. So write this code right below “import smtplib”.

```
smtpDstAddr = “807632219@qq.com”

smtpSrcAddr = “srhongyangkai02@163.com”
```

Then set the SMTP server information. Since I choose 163 box to be the sender, I wrote the information of 163 server. The server domain name and port number can be found on its official website. smtpPwd is my smtp password. 163 box’s smtp password is different from its login password. I set it on my mailbox setting page.

```
smtpServer = “smtp.163.com”

smtpPwd = “hyk1994125”

smtpPort = 25
```

Next, set mail content.

```
mailSubject = “Warning!!!”

mailText = “Someone broke into your house!”

mailMsg = “To:” + smtpDstAddr + “\n” + \

    “From:” + smtpSrcAddr + “\n” + \

    “Subject:” + mailSubject + “\n” + \

    “\n” + mailText + “\n”
```

The content of e-mail has been finished. I need to send the e-mail when infrared module detects someone. So modify the detect() function of safety.py.

```
if GPIO.input(11) == True:

    mySMTP = smtplib.SMTP(smtpServer,smtpPort)

    mySMTP.starttls()

    mySMTP.login(smtpSrcAddr,smtpPwd)

    mySMTP.sendmail(smtpSrcAddr,smtpDstAddr,mailMsg)

    mySMTP.close
```

And the last step, camera monitoring. When I received the warning e-mail, I can not see what is going on in my house. I need to see whether there really is a thief. It is easy to achieve this function on raspberry pi. First, install a software named motion.

```
sudo apt-get install motion
```

Then open the motion daemon so that it can always run in the backend.

```
sudo vim /etc/default/motion
```

Find this line “start_motion_daemon=no”, change it to “start_motion_daemon=yes”.

Modify configuration file of motion.

```
sudo vim /etc/motion/motion.conf
```

Change “deamon off” to “deamon on”, “webcam localhost on” to “webcam localhost off”

Then run motion.

```
sudo motion
```

Now, I can access http://192.168.137.188:8081(LAN IP of raspberry pi) in Firefox browser to see the current camera shot.

At this point, the hardware part is complete.

3.2.1.2 App part

I decided to use Cordova to develop the app. Before starting to use Cordova, I installed some components. They are essential.

First of all, NodeJS and NPM. NodeJS is a platform for Cordova development. NPM is a package management tool that installed along with the NodeJS. It can solve many problems in the deployment of NodeJS code. In this project, the role of NPM is to download and install Cordova to my computer from NPM server.

Second, Android SDK. To develop Android app, I need to install Android SDK on my computer.

Third(optional), XCode. I did not install it because my computer is windows system and cannot install this. If you have a MacBook and you want to develop IOS version, you should install it.

Next, install git. Although it may not be used directly, it should be installed. Because Cordova uses

some of its background process.

At last, install Cordova. Click Run on the start menu, type CMD and press enter to bring up the DOS command window. Then write the command below.

npm install –g cordova

"-g" means the global mode. It enables me to execute Cordova command in DOS command window.

After installing, input this command to check whether the operation is a success.

cordova –v

Figure 5 Cordova version

Figure5 shows I have installed Cordova 7.0.0 version to my computer.

After installing, I created a Cordova application. It included several steps.

Step 1, at DOS command window, I installed the application in C:\Users\hyk.

C:\Users\hyk\>cordova create cp io.cordova.hellocordova Smart Home

"cp" is the name of the directory I created my application.

"io.cordova.hellocordova" is the default reverse domain value. It is not easy to understand and seems useless. But it is an essential attribute. So just write it.

"Smart Home" is the name of my application. It showed on my phone later.

Step 2, add platform. First of all, open the application directory in DOS. Since I developed the Android version, I write this command.

C:\Users\hyk\cp>cordova platform add android

Step 3, build and run. In this step, I built application for Android so that I can run it on my phone.

cordova build android

This step took me a long time. Because it need to download the necessary packages for the first time. There are so many packages that the installing process lasted for several hours. However, build did not take so many time after this.

Then I can run the application on my phone by executing this command.

cordova run android

Up to this point, I saw my first Cordova app on mobile.

Figure6 shows the Cordova project directories structure. Html, css and js files should be set in the www directory.

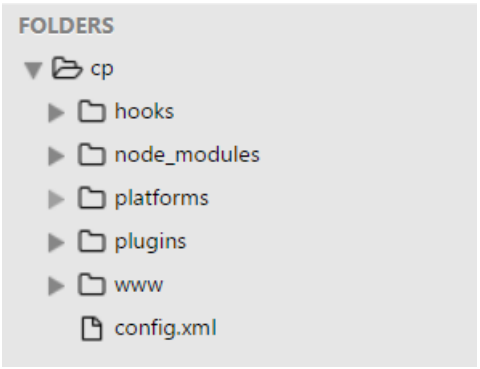

Figure 6 Cordova directory structure

Next, I modified the index.html to be home page of the app. The index.html shows the templates list, safety and wakeup. At the bottom, there is a navigation bar. It is used to jump to equipment.html and setting.html. At the home page, I can click the safety template button to jump to its control page, which is stored on raspberry pi server. This is achieved through InAppBrowser plugin for Cordova.

First, I installed this plugin in the DOS window then I could use it.

cordova plugin add cordova-plugin-inappbrowser

Next, I added event listener to safety button.

document.getElementById(“safetybt”).addEventListener(“click”,openBrowser);

Third, create openBrowser() function.

```
function openBrowser() {
        var url = 'http://192.168.43.179:8086/';
        var target = '_blank';
        var options = "location=yes;zoom=no"
}
```

192.168.43.179 is the IP address of raspberry pi after it connecting to my mobile hotspot. In this way, the app can access the control page.

The control page was stored in the raspberry pi. The raspberry pi should run a python script to be a web server. The server.py is

```
#!/usr/bin/env python3
from bottle import get,post,run,request,template
@get("/")
def index():
return template("index")
@post("/cmd")
def cmd():
print("Press the button:"+request.body.read().decode())
return "OK"
run(host="192.168.43.179",port=8086)
```

“#!/usr/bin/env python3” tells the shell this file is python source code, let bash call python3 to explain this code. “from bottle import get,post,run,request,template” imports the methods and objects that I use from the bottle framework. The following statements define 2 routes. One is “/”, it is the get type (decorated with @get). The other is “/cmd”, it is the POST type (decorated with @post). The first route read the index template (the template is a HTML) and send it to the client(browser), because “/” is the IP address of raspberry pi. Similarly, the second route “/cmd” visits http://ip/cmd. “run(host=”192.168.43.179”,port=8086)” call the run method of bottle and build a HTTP server so that I can access my interface through the browser at 8086 port.

The index.tpl was stored in the same directory with server.py. I program the index.tpl to be the safety template control page. This page shows the equipment list of this template. And I can click button to start or stop the template (execute safety.py or kill the process of safety.py). Additionally, I can click the camera button to visit 192.168.43.179:8082. This page shows the current camera shot. Up to this point, a local safety template has been finished.

3.2.1.3 Remote control

However, local safety template does not conform to the actual. So I need to achieve the remote control function. My approach is port mapping. First of all, the router in my home has a public network IP address. Suppose it is 124.127.207.175. The raspberry pi connects to its Wi-Fi and get a static local IP address, for example, 192.168.137.188. I can enter router settings and map 124.127.207.175:8086 to 192.168.137.188:8086. Then I can control the device from a remote

location. But there is another problem, the public IP address of my router is dynamic. So I set up DDNS (Dynamic Domain Name System) in router. It means no matter the public IP address of router is 124.127.207.175 or 122.122.122.122 or something, they all can be analysed to a fixed domain name. For instance, idhyk.top. In this way, I can access idhyk.top:8086 to control the device remotely. Figure7 shows the remote connection solution.

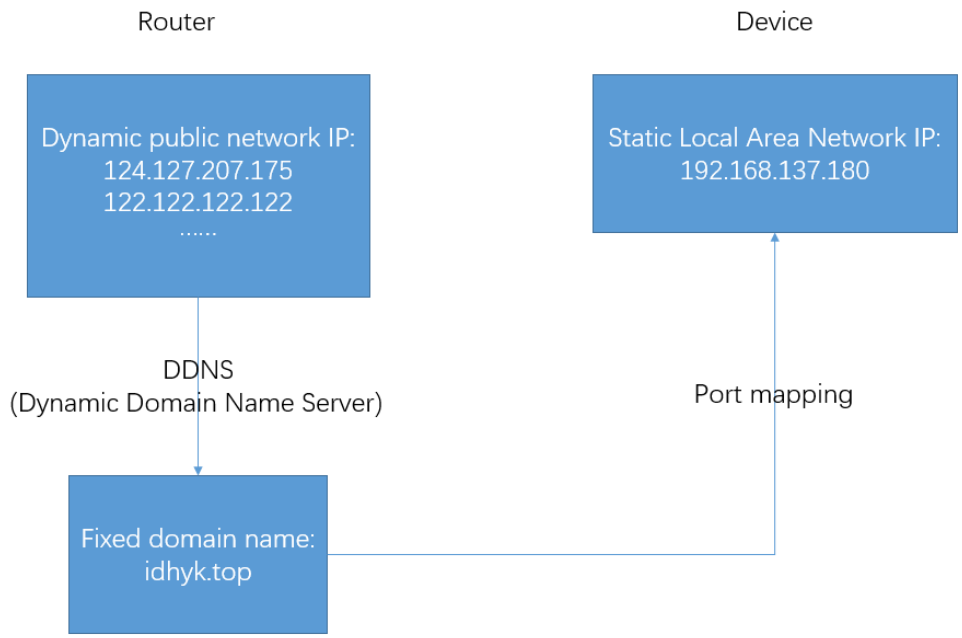

Figure 7 Remote connect

So far, the app realized the function of remote control.

3.2.2 Wakeup template

When user starts the template, the application opens the Bluetooth and start scanning. This is achieved through ble plugin for Cordova.

Again, install the plugin first.

```
cordova plugin add cordova-plugin-ble
```

Then use evothings.ble.startScan() function to open the Bluetooth and start scan.

```
document.getElementById("ble").addEventListener("click", wakeup);

function wakeup(){
```

```
        evothings.ble.startScan(onDeviceFound,onScanError);
        function onDeviceFound(device){
            console.log('Found device:' + JSON.stringify(device));
        }
        function onScanError(error)
        {
            console.log('Scan error: ' + error);
        }
    }
```

Since my mobile phone has been connected to these devices before. Bluetooth connected to devices automatically. So the next step is to have the devices work at the appointed time. First, implementing the function of setting the wake-up time. The following code implements a time selector.

```
    datePicker.show(options, onSuccess, onError);
    var options = {
        date: new Date(),
        mode: 'time',
        okText:'Ok',
        cancelText:'Cancel',
        is24Hour:true,
        androidTheme:'THEME_HOLD_DARK'
    };
```

After setting the time, the application could calculate how long it takes to get up. Play music function delayed the execution of this period of time. It is implemented through the following code.

```
    function onSuccess(date) {
        info.innerHTML = "Wake me up at: "+date.getHours()+":"+date.getMinutes();
```

```
            var date1=new Date();
            var date2=date.getTime()-date1.getTime();
            console.log(date2);
            setTimeout(playAudio,date2);
            function playAudio() {
                var src = "/android_asset/www/audio/Bad Day.mp3";
                if(myMedia === null) {
                    myMedia = new Media(src, onSuccess, onError);
                    function onSuccess() {
                        console.log("playAudio Success");
                    }
                    function onError(error) {
                        console.log("playAudio Error: " + error.code);
                    }
                }
            myMedia.play();
            }
    }
```

So the audio function is completed. The function of light and curtain remains. Since the project lack s funds and hardware, I cannot do the actual test. But the theory should be as follows.

The steps needed to connect to a BLE device and start reading/writing data are:

- Scan for the device
- Connect to the device
- Read and write characteristics

Scan and connect part has been finished. In order to control the light and curtain, I just need to do the third step. Characteristics are like commands or functions, they can be read to obtain data, or

written to control some aspect of a device.

For example, I turn the light on and off by writing to a characteristic. Using a Cordova plugin named cordova-plugin-ble. I can use function evothings.ble.getCharacteristic(service,uuid) to get the characteristic I want to write. To write a characteristic, use evothings.ble.writeCharateristic(device, characteristic,data,success,fail). Example code:

```
//UUID of light configuration characteristic (write 1 to turn ON, 0 to turn OFF).
        var LIGHT_CONFIG = 'f000aa72-0451-4000-b000-000000000000'
//Get light characteristic
        var configCharacteristic = evothings.ble.getCharacteristic(service,LGIHT_CONFIG)
//Turn light ON
    evothings.ble.writeCharacteristic(
            device,
            configCharacteristic,
            new Unit8Array([1]),
            onLightActivated,
            onLightActivatedError)
    function onLightActivated(){
            console.log('Light is ON');
    }
    function onLightActivatedError(){
            console.log('Light activate error:' +error);
    }
```

This implements the Bluetooth switch light function. I think Bluetooth controls the curtain is similar. Just different devices have different UUID and data formats.

Chapter 4: Results and Discussion

For safety template, when I clicked the start button, the application showed the working state of the device. As shown in figure8.

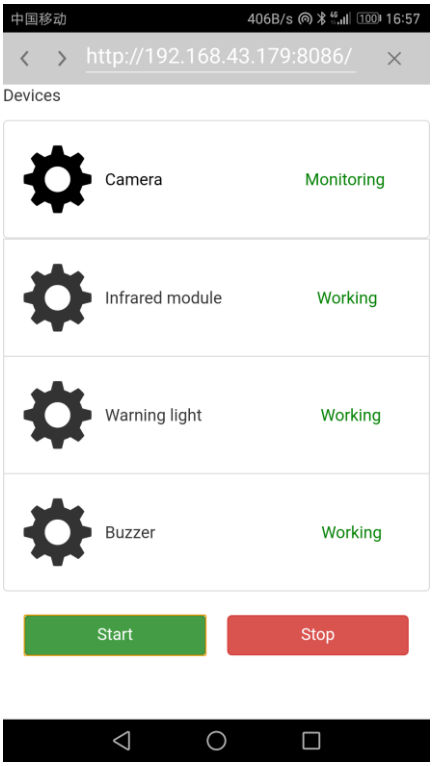

Figure 8 Safety template control page

And the device started working. Since the infrared module detected me, the buzzer rang, the warning light turned on. The devices are shown in figure9.

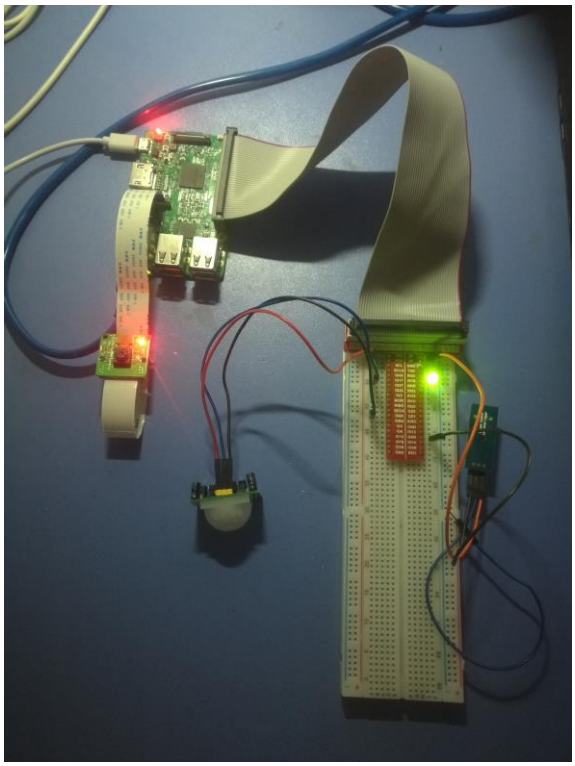

Figure 9 Device working

At the same time, I received a warning message from the device. The contents of the message are shown in figure10.

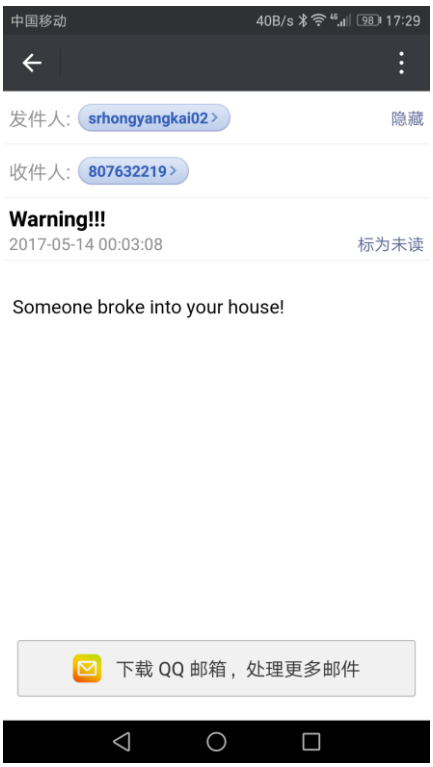

Figure 10 Warning message

Then I looked at the monitor screen in the application. It showed the picture of my apartment. Figure11 is a screenshot of the monitor page.

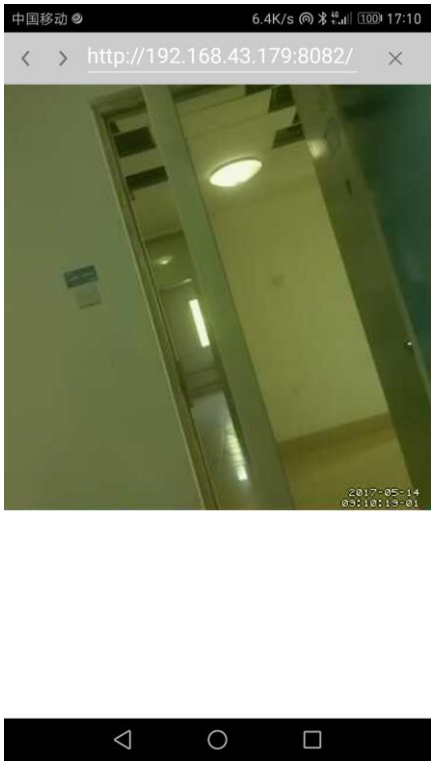

Figure 11 Monitor page

Then I clicked the stop button. The user interface changed and the device is closed. As shown in figure12.

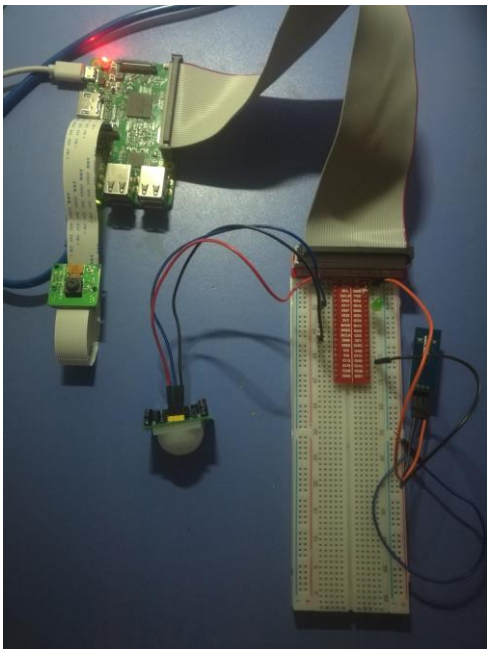

Figure 12 Device close

As for wakeup template, I clicked the start button, a time picker appeared. After I set the time, the application showed the device state and when wakes me up as figure13.

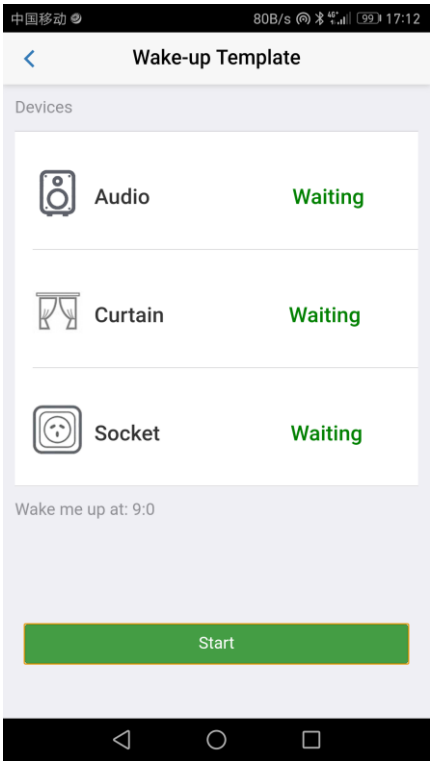

Figure 13 Wakeup template

At nine o'clock, the audio played music and waked me up.

Since this project is to develop a mobile application, only two templates are incomplete. So I added a device page to allow users to control some devices individually. The device page showed a list of available devices. Users can click on each device to enter the settings page. However, the settings page has only a user interface and does not really interact with the device. Because it is additional part and I do not have enough money and capacity to study so many hardware. After all, it is a software project. Figure14 is the screenshot of device page.

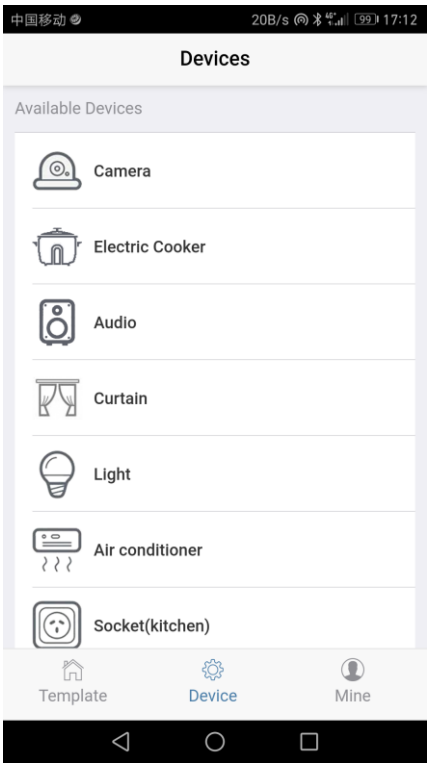

Figure 14 Device page

Chapter 5: Conclusion and Further Work

5.1 Conclusion

This project has implemented a smart home android app. It is a hybrid app developed by Cordova framework. So it can cross platform. If you want to generate IOS version of the app, you do not need to change the code. Just copy the project to your MacBook and add IOS platform to the project in Xcode, then the app can run on your iPhone. For other mobile platforms, the process is similar.

The app mainly includes two templates, safety template and wakeup template. Safety template realizes the functions of remote monitoring and switch alarming equipment. Its hardware part is implemented using raspberry pi and related components. Wakeup template realizes the function of waking user up via Bluetooth controlling devices. In the initial design, this template should control audio, curtain and socket for rice cooker. However, since lacking funds and devices, this project only tested a Bluetooth audio. But this project still put forward the complete Bluetooth control method.

To summarise, this project accomplished the main tasks and target in the specification. I built a visual HTML5 template development environment and developed two kinds of HTML5 template. This project had finally implemented a demo template and mobile app.

5.2 Further work

I think the existing smart home app on the market is different from mine on operation principle. Although I achieve the project objective, my method is not advanced. I think the reason is that I have limited knowledge of hardware and networking. Additionally, the smart home apps in the market seem to be native development. So they must be different from web app. I think web app may have some limitations in mobile platforms.

Thus, if I want to research smart home in the future, there is a lot of knowledge that I need to learn. For example, communication protocol, cloud services, native app development.

[Design and Implementation for customized mobile HTML5 template for end users]

References

[1] *Cordova environment setup*. Retrieved DEC 5, 2016, from Web site: http://www.w3cschool.cn/cordova/cordova_environment_setup.html

[2] *HTML tutorial*. Retrieved DEC 6, 2016, from Web site:

http://www.runoob.com/html/html-tutorial.html

[3] *CSS tutorial*. Retrieved DEC 13, 2016, from Web site:

http://www.runoob.com/css/css-tutorial.html

[4] *JavaScript tutorial*. Retrieved DEC 20, 2016, from Web site:

http://www.runoob.com/js/js-tutorial.html

[5] *jQuery selector*. Retrieved JAN 3, 2017, from Web site:

http://www.runoob.com/jquery/jquery-selectors.html

[6] AfternoonLeaf. (2015, Jun). *Cordova application analysis*. Retrieved JAN 10, 2017, from Web site: https://segmentfault.com/a/1190000002927103

[7] *Cordova InAppBrowser*. Retrieved JAN 17, 2017, from Web site:

http://www.w3cschool.cn/cordova/cordova_inappbrowser.html

[8] *Cordova media*. Retrieved JAN 20, 2017, from Web site:

http://www.w3cschool.cn/cordova/cordova_media.html

[9] Hu, S. T. (2016). *Start learning raspberry pi from scratch: super fun smart small hardware production book*. Beijing: Tsinghua University press. (ISBN 978-7-302-43265-4)

[10] mrr. (2016, Feb). *Simple implementation of raspberry pi WEB control using Python*. Retrieved March 1, 2017, from Web site: http://www.jb51.net/article/79673.htm

[11] Raspberry pi Lab. (2015, Feb). *Another method for enabling raspberry pi to boot Python script*. Retrieved March 8, 2017, from Web site: http://shumeipai.nxez.com/2015/02/09/run-python-script-when-the-raspberry-pi-start.html

[12] Raspberry pi Lab. (2016, Sep). *Raspberry pi + motion installation camera to achieve remote monitoring*. Retrieved March 16, 2017, from Web site:

[Design and Implementation for customized mobile HTML5 template for end users]

http://shumeipai.nxez.com/2016/09/01/raspberry-pi-motion-cameras-for-remote-monitoring.html

[13] Lantuzi. (2015, Mar). *Python to achieve SMTP send mail, Web server*. Retrieved March 24, 2017, from Web site: http://www.eefocus.com/nightseas/blog/15-03/310851_d1c6f.html

[14] linuxcom. (2013, Oct). *The dynamic DNS (DDNS) setting of the router*. Retrieved April 2, 2017, from Web site: http://jingyan.baidu.com/article/c275f6bafd7b17e33d75678b.html

[15] huangmr0610. (2016, Sep). *N methods for killing processes under Linux, including Python – killproc*. Retrieved April 10, 2017, from Web site:

http://blog.csdn.net/huangjin0507/article/details/52586854

[16] Mikael, K. (2016, Nov). *Evothings BLE API Guide*. Retrieved April 17, 2017, from Web site:

http://evothings.com/doc/tutorials/evothings-ble-api-guide.html

[17] JRZJ. (2016, Jun). *Raspberry pi operations*. Retrieved March 1, 2017, from Web site: http://www.craftsmanotes.com/42/

Acknowledgement

I would like to thank my supervisor. He gave me very important guidance and advice. I am also particularly grateful to his student Feng Yimeng for her timely help.
Thanks to family and friends for their help and support.

Appendix

北京邮电大学
本科毕业设计（论文）任务书
Project Specification Form

学院 School	International Schoo	专业 Programme	Telecommunications	班级Class	2013215106
学生姓名 Name	HONG YANGKAI	学号 BUPT student no	2013213085	学号 QM student no	130800662
设计（论文）编号 Project No.	IP_3085				
设计（论文）题目 Project Title	Design and Implementation for customized mobile HTML5 template for end users				
论文题目（中文）	为终端用户设计和实现定制的手机HTML5模板				
题目分类 Scope	Implementation	Phone apps	Software		

主要任务及目标Main tasks and target:	By
Task 1: Set up a enviroment for mobile HTML5 template development.	30 November 2016
Task 2: Set up HTML5+CSS+Javascript based customized mobile HTML5 template.	01 February 2017
Task 3: Implement two type of HTML5 templates for end users.	09 March 2017
Task 4: Implement a HTML5 template-based mobile application.	23 April 2017

Measurable outcomes
1) An visual HTML5 template development environment
2) development two kinds of HTML5 template
3) A demo template and mobile application

主要内容Project description:

The template is a kinds of software, which includes a wide variety of pre-developed and designed UI elements, remote/local interface invocation methods for mobile end users. In this project, we will design and implementation for customized mobile HTML5-based service template for end users, and eventually which can provides a kind of simple and interactive wizard to create the applications for mobile and web applications.

Project outline

This project is to design and implementation a mobile HTML5 template for end users. The template is the front end of a smart home phone application, which includes a variety of pre-developed and designed UI elements. I will tackle this project in four steps.

First of all, I will experience some smart home phone apps available in the market such as HONYAR smart home and BroadLink, then analysis their pros and cons. After that, I will seriously consider user requirements and design my own application to be different from them in some ways.

In second step, I will draw a operating flow chart to describe the logical system.

Thirdly, I will use Axure, a interactive prototyping software, to develop a rapid prototype. The prototype will show the user interface and interactions.

The last step is programming and it is the most important part of this project. In fact, the template is a html file. I need to use three programming languages, HTML, CSS and JavaScript to develop it. These three languages work together, HTML define the content of the page, CSS describe the layout and style of the page and JavaScript write the behavior of the page. I will develop this html file on PC by using Eclipse and some other web development tools. This file can run on different platforms such as Android and IOS. When this file is well completed, it will run the same as the prototype I built in the third step.

What I expect to have working at the mid-term oral

A complete Axure prototype, which shows what the application exactly looks like. And an incomplete html file, which can run on different platforms such as Android and IOS. But it will not have all the functions, just a part of them.

[Design and Implementation for customized mobile HTML5 template for end users]

Fill in the sub-tasks and select the cells to show the extent of each task

	Nov		Dec		Jan		Feb		Mar		Apr		May	
Task 1: Set up a enviroment for mobile HTML5 template development.														
Download a Eclipse and configure it														
Task 2: Set up HTML5+CSS+Javascript based customized mobile HTML5 template.														
Learn HTML5														
Learn CSS3														
Learn JavaScript														
Develop a simple template														
Task 3: Implement two type of HTML5 templates for end users.														
Use Axure to build a rapid prototype														
Programme according to the prototype														
Task 4: Implement a HTML5 template-based mobile application.														
Programme														

北 京 邮 电 大 学

BBC6521 Project 毕业设计 2016/17

Early-term Progress Report

初期进度报告

学院 School	International School	专业 Programme	Telecommunications	班级 Class	2013215106
学生姓名 Student Name	HONG YANGKAI	BUPT 学号 BUPT Student No.	2013213085	QM 学号 QM Student No.	130800662
设计（论文）编号 Project No.	IP_3085	电子邮件 Email	2013213085@bupt.edu.cn		
设计（论文）题目 Project Title	Design and Implementation for customized mobile HTML5 template for end users				

已完成工作：
Finished Work：

1. Be familiar with the development tool. I use HBuilder instead of Eclipse because I think it is more suitable for web develop. In fact, many tools can do this work. As for me, HBuilder is very fast and have some great plugins so I choose it. I may also use more tools later if necessary.
2. Be familiar with the programming languages, HTML5, CSS and JavaScript. In addition, understand some web frameworks, such as Bootstrap. And understand some JavaScript libraries, such as jQuery.
3. I have finished two simple pages and I used HBuilder to pack these files to form an Android apk. Then I installed the apk on my phone. It works as a very simple smart home application. However, it is just practice.

是否符合进度？On schedule as per GANTT chart?	[YES/NO] YES

下一步：
Next steps:

1. Design the smart home application. Think more about the scenes and equipments. Discuss my idea with supervisor assistant. Then develop some pages of the smart home application.
2. In the process of implementation, there is no doubt that I will face some challenges. When I get into trouble, I need to learn the programming languages deeper.
3. Revise the pages according to supervisor’s suggestions.

北 京 邮 电 大 学

BBC6521 Project 毕业设计 2016/17

Mid-term Progress Report

中期进展情况报告

学院 School	International School	专业 Programme	Telecommunications	班级 Class	2013215106
学生姓名 Student Name	Hong Yangkai	BUPT 学号 BUPT Student No.	2013213085	QM 学号 QM Student No.	130800662
设计（论文）编号 Project No.	IP_3085	电子邮件 Email	2013213085@bupt.edu.cn		
设计（论文）题目 Project Title	Design and Implementation for customized mobile HTML5 template for end users				

毕业设计（论文）进展情况，字数一般不少于 1000 字
The progress on the project. Total number of words is no less than 1000

目标任务: Targets set at project initiation:

A complete Axure prototype, which shows what the application exactly looks like. And an incomplete html file, which can run on different platforms such as Android and IOS. But it will not have all the functions, just a part of them.

是否完成目标 Targets met?	[YES/NO] YES

目前已完成任务 Finished Work:

1. Set up an environment for mobile HTML5 template development. I use a HTML IDE, which called HBuilder.

2. Set up HTML5+CSS+JavaScript based customized mobile HTML5 template.

3. Implement two type of HTML5 templates for end users. I have finished a scene template and an equipment template. While implementing the equipment template, I need to realize an adding equipment function. In this function, the application will scan the equipment that are in the same WLAN with your mobile phone. It is not real detection, but simulation. I need a circular progress bar to show the progress of detecting equipment. It is hard for me at the beginning, then I found some JavaScript widgets, which have been well developed by other people before, have great performance. I tried three JS widgets, they are circle-progress.js, radialIndicator.js and progressbar.js. However, circle-progress.js and radialIndicator.js cannot meet another requirement, I need to show some hidden div when the circle reaches 100 percent. Circle-progress.js and radialIndicator.js seem not be able to get the value of the progress bar. Fortunately, progressbar.js can make it. In addition, many JS widgets' documents are full English, it takes time to learn. In a word, I spent a lot of time implementing the templates.

4. Implement a part of a HTML5 template-based mobile application.
尚需完成的任务 Work to do: 1. Optimize the codes, especially CSS codes, to make the pages look more beautiful. 2. Optimize the codes to make the pages look consistent on mobile phones of different sizes. 3. Learn more about JavaScript to know how to interact the scene page with the equipment page. For example, suppose a user has added an equipment to the application. If he wants to add the equipment to the morning scenario, the scene page need to get the information of equipment page. So I need to learn how to write the JavaScript codes. 4. Complete the functions of the smart home application to make it clear and easy to use.

能否按期完成设计（论文） Can finish the project on time or not:	[YES/NO]YES

存在问题 Problems:

1. Some pages can do the right things, but they are not very good-looking.
2. Style inconsistencies. Because I use MUI framework, bootstrap framework and something. They have different styles.
3. Layout errors. The layout may change on mobile phones of different sizes.
4. The two templates, scene template and equipment template cannot interact with each other. That is to say, I don't know how to pass value between two html files.
5. Current operation mode of the application is not very good. I need to make the application clear and easy to use.

拟采取的办法 Solutions:

1. Read more other people's codes, learn how they make the page looks good. Because a lot of work has been done in this field. The things I am doing now, many people have down them before. It's not necessary to reinvent the wheel. I can use the standards already exist. I need to find out some good teaching documents and imitate their style, rather than creating by myself. Besides, I will find out more beautiful picture materials on the Internet to make the application more good-looking.
2. Modify the CSS codes to make every element have unified style. But it is not an urgent task. I may leave it to the last.
3. Sometimes I have to write, for example, to make the div located in centre. But the 100px means that the div may not be located in centre on another mobile phone, which is bigger or smaller. First of all, I should make sure the layout on my phone is right. Then if time is enough, I will check if it is possible to make the layout correct on screens of different sizes.
4. JQuery ajax seems be able to solve this problem, I will read some articles and tutorials of this technique.
5. I will compare several smart home apps in the market, such as HiLink, mijia, BroadLink and HONYAR. Absorb their strengths and make my application clear and easy to use.

最终论文结构 Structure of the final report:

Abstract

Chapter 1: Introduction

1.1 Motivation

1.2 Functionality to realise

日期 Date: 09/03/2017	

Risk Assessment

This project develops a smart home app. The app mainly contains two templates. Safety template realizes the function of remote controlling raspberry pi. The raspberry pi automatically connects home Wi-Fi and become web server after booting so that my app can control it remotely.
Wakeup template realizes the function of Bluetooth device controlling.

Table 1 Risk Assessment

Description of risks	Description of impact	Likelihood rating	Impact rating	Preventative actions
Home Wi-Fi off	Raspberry pi server shutdown. After reconnecting to network, it cannot become a server again automatically.	1	1	After home Wi-Fi on, use a mobile SSH software to run server script on it.
Home power off	No impact. Because it will automatically become a server after it restarts.	1	0	Just wait for home power on.

Environmental Impact Assessment

This project develops a smart home app. The app mainly contains two templates. Safety template realizes the function of remote controlling raspberry pi. The raspberry pi automatically connects home Wi-Fi and become web server after booting so that my app can control it remotely.
Wakeup template realizes the function of Bluetooth device controlling.
This smart home app has a minor influence on environment. In terms of cost of manufacture, waste disposal and recycling, energy use in service and savings in energy. Devices controlled by the app cost very low energy.