

Anti-lost System Documentation

Li Shitao

May 31, 2016

1 Introduction

Nowadays, with quick pace of life, people frequently forget or lost their personally things. As the techonology developing significantly, it is naturally to have the idea that the high technology may help people .This anti-lost system is designed and developed to solve this problem using the Bluetooth technology. It can help people quickly find the things that they can not find currently, as well as recording the time and location that the item was last discovered. All the source code is available on <https://github.com/list12356/AntiLost-App.git>. The main functoin and overview of the system is given in Chapter 2, the step of using the system is given in Chapter 3, the details about how the system work will be explained in Chapter 4, and I also gives my future plan in Chapter 5.

2 Main function

This system consists of an Android-App and several raspberry-Pi representing the personally things of the user. The main function of this system are listed as following

- add, remove or edit the device the user want to keep listening
- quickly send a beep to the device that user want to find within the range of bluetooth communication
- alarm user when devices are too far from the user
- record the geographic information once the device is scanned

3 User Guide

3.1 Add and delete

For a new user to use the system, you should keep your bluetooth on in your phone and switch on the raspberry Pi, the Python script will automatically run

on the raspberry Pi. Once you finished, open the app, tap the menu in the right-up corner and select the 'add item' option, the page will jump to another activity. Press the scan button and you can get all the bluetooth devices available, select one you want to keep listening and give it a name (blank for its default name). Then you can get the device shown on your main item list. If you no longer want to keep the device, you can select the 'delete item' in the menu and choose the device in the delete phrase.

3.2 Find and Call the item

If you want to find your Item, select it and the program will try to connect it, if success, you can get the RSSI of the device and a speculated distance according to the RSSI. Choose the 'Call my item' button, the program will send a message to the Raspberry Pi to make it beep. If it fails, you can see the information about the last time and location that the device was scanned.

If you are too far from your device, the program will automatically send a message causing both your phone and the Raspberry Pi to alarm you. Also, whether you want to receive the alarm and the alarm distance can both be set by the user.

3.3 User Settings

4 Details of impletation

4.1 Java Class

I used two extra java classes to store the user's information. The first one is the *myItem.java*, it contains the name, address, last scanned time and location of the item. The second one is the *ItemList.java*, it contains a List data stores all the user's items and several methods including add and delete data.

4.2 The bluetooth communication

In the MainActivity, there is a timer task run background to scan all the available devices around. Once it discovers a device that is in the user's item list, it will refresh the information, eg. RSSI, location and time, in the item list. Noticing that the bluetooth discovery will automatically restart in 12 seconds, which is too long a period for the RSSI value to refresh, we set the schedule period of the task to 3 seconds. When user select one item, the program will try to start a connection to the device using a bluetooth socket. The pseudo code is in ??.

Maybe it is proper to keep the connection in order to save time. However, the bluetooth can only have 8 device connecting at the same time, and it is a waste of resources considering that in most time there is actually no communication between devices. Besides, in practice, I find that the keeping of connections will influence the quality of the discovery process in some older bluetooth devices.

4.3 The location services

I use the BaiduLocation SDK and BaiduMap SDK to finish the location and map services. In the main activity, there's a listener

4.4 The Raspberry Pi

The model I used is Raspberry Pi Model 2.0B. I used a HC-05 bluetooth adapter and a low-level triggered buzzer. I write a Python program to resolve the message recieved from the HC-05 adapter. The program will read from a FIFO file `/dev/ttyAMA0` every 0.1s, and once it recieve the message from the FIFO, it will send a high voltage to the GPIO output to beep the buzzer.

5 Conclusion and Future Plan