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 avaliable 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.

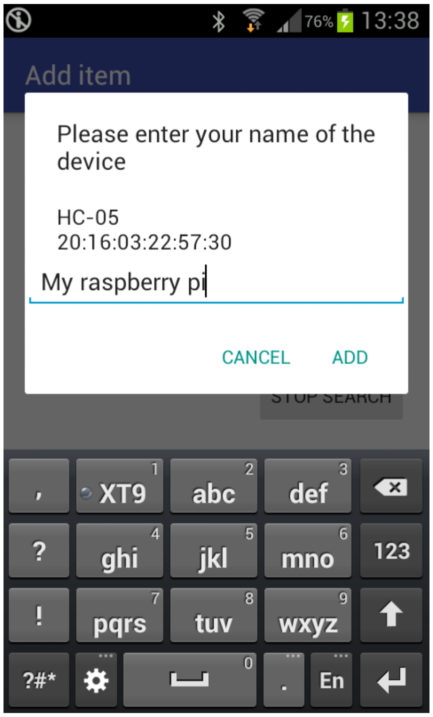
2 Main function

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

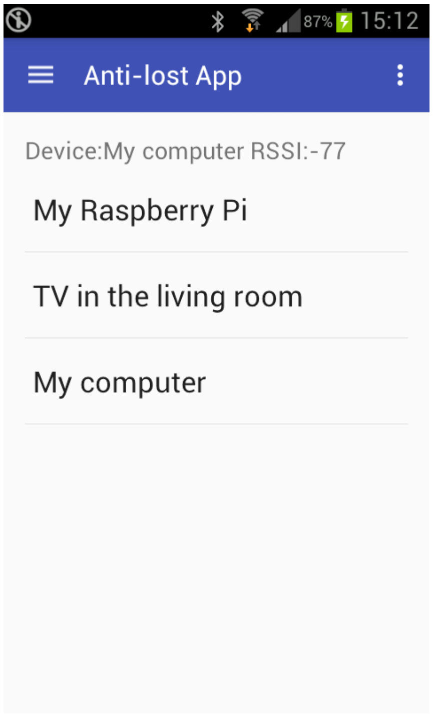
For a new user to use the system, you should keep your bluetooth on in your phone and switch on the raspberry Pi, the Python script will automatically run on the raspberry Pi. Once you finished, open the app, tap the menu in the right-up corner and select the ‘add item’ option, the page will jump to another activity. Press the scan button and youcan get all the bluetooth devices avaliable, 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.



Please enter your name of the device

HC-05
20:16:03:22:57:30

My raspberry pi



Device:My computer RSSI:-77

My Raspberry Pi

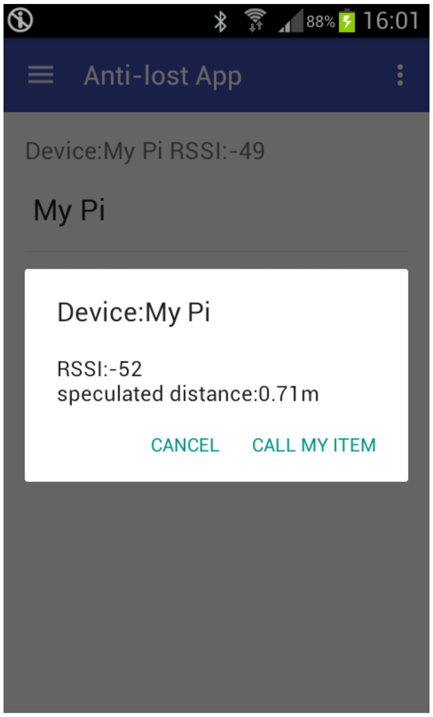
TV in the living room

My computer

3.1 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 Raspeberry 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 auto matically send a message causing both your phone and the Raspberry Pi to alarm you. Also, whether you want to recieve the alarm and the alarm distance can both be set by the user.

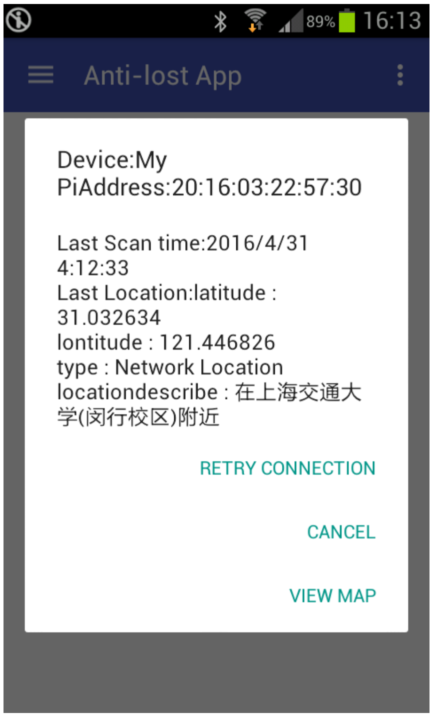


Device:My Pi RSSI:-49

My Pi

Device:My Pi

RSSI:-52
speculated distance:0.71m



Device:My PiAddress:20:16:03:22:57:30

Last Scan time:2016/4/31 4:12:33
Last Location:latitude : 31.032634
lontitude : 121.446826
type : Network Location
locationdescribe : 在上海交通大学(闵行校区)附近

3.2 User Settings

4 Details of impletation

4.1 Java Class

I used two extra java calss to storage the user's information. The first one is the *myItem.java*, it contains the name, address, last scaned 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

When user select one item, the program will try to start a connection to the device using a bluetooth socket. The UUID is automatically set to"00001101-0000-1000-8000-00805F9B34FB", which is a universal code for all HC-05 SCM.We also give the speculated distance, which is derived from the following formula

$$d(r) = 10^{\frac{|r|-r_0}{k}}$$

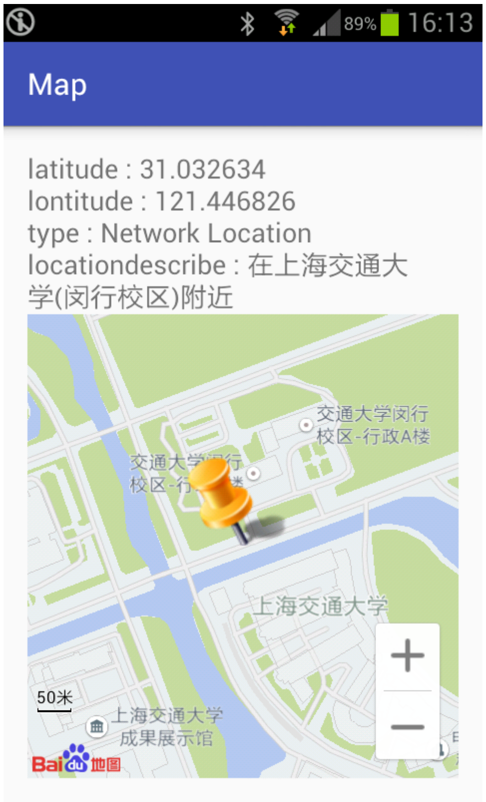
where the r_0 denotes the absolute value of RSSI when the actual distance is $1m$, and the k is the decaying factor. In practice, I measured that the RSSI around

$1m$ is about 60, and when the device is very close to the phone,e.g. about 10 cm, the RSSI value is around 35. Thus according to the two know points ,we can build the equation and solve that $k = 20$.

In the MainActivity, there is a timer task run background to scan all the available devices around. Once it discoveres a device that is in the user's item list, it will refresh the information, eg. RSSI, location and time, in the item list. Also, the scan progress also will check whether the device is too far from the user, and if true, the program will send a message to Raspberry to alert the user. 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.

May be it is proper to keep the connection in order to save time. However, the bluetooth can only have 8 device connecting at the sametime, 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 Localization Services



latitude : 31.032634
lontitude : 121.446826
type : Network Location
locationdescribe : 在上海交通大学(闵行校区)附近

I use the BaiduLocation SDK and BaiduMap SDK to finish the location and map services. In the main activity, there's a listener called *BDLocationListener* to get and recieve the location from the Baidu service. As metioned above, when any device listed in the item list is scaned, its location information will be updated according to this method. In the Map Activity, I use the Baidu Map SDK to draw a map and show the text information. The result is shown in

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 recieves the message from the FIFO, it will send a high voltage to the GPIO output to beep the buzzer.

5 Conclusion and Future Plan

The system now just do the basic functions well. However there are still lots of things to be improved in the future, including the robustness of the program and the user experience.

5.1 Online-version

n the future, I plan to make an online version. It is intrinsic to think that when many people are using this system, people can share the information that they have.A user may scans something it is not helpful for user A, but he/she can upload whatever he/she has scaned to the server, and the server can find whether somebody else is finding these things.For, example, If user A scaned an item that belongs to user B,

5.2 Combined with Indoor Localization system

It is a common knoeledge that the GPS localization is not available in the indoor environment. Thus if someone lost his/her thing in a relatively big indoor environment, e.g., supermarket or shopping mall. Even he/she record the position, it is difficult for him/her to find it. Considering that the indoor-localization is a popular topic recent years, it is very promising to combine these two system togather.