**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT post, Bangalore-54

A Dissertation Report on

**BLUETOOTH BASED PRESENCE DETECTION**

Submitted by

Prithvi Sharan S 1MS12CS078

Raja Andukondan 1MS12CS083

*in partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu), **May 2015**

# Abstract

Bluetooth enabled mobile devices is an excellent and energy efficient method to detect the presence of individuals. This implementation mainly aims at using the computing power of Raspberry Pi, a powerful system on chip, in order to continuously monitor the presence of registered Bluetooth enabled mobile devices, and report the same on a webpage hosted on the local network on a timely basis.

#### Contents

***Declaration i***

***Acknowledgements ii***

***Abstract iii***

***List of Figures***

***List of Tables***

1 **INTRODUCTION 7**

* 1. General Introduction
  2. Statement of the Problem
  3. Objectives of the project
  4. Project deliverables
  5. Current Scope
  6. Future Scope

1. **PROJECT ORGANIZATION**
   1. Software Process Models
   2. Roles and Responsibilities
2. **LITERATURE SURVEY**

3.1….Introduction

3.2…Main Body

3.3 Conclusion of Survey

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**

5.1 Product Overview

5.2 External Interface Requirements

5.2.1 User Interfaces

5.2.2 Hardware Interfaces

5.2.3 Software Interfaces

5.2.4 Communication Interfaces

5.3 Functional Requirements

5.3.1 Functional Requirement 1.1

:

5.3.n Functional Requirement 1.n

5.4 Software System Attributes

5.4.1 Reliability

5.4.2 Availability

5.4.3 Security

5.4.4 Portability

5.4.5 Maintainability

5.4.6 Performance

5.5 Performance Requirements

5.6 Database Requirement

5.7 Design Constraints

5.8 Other Requirements

1. **DESIGN**
   1. Introduction
   2. Architecture Design
   3. Graphical User Interface
   4. Class Diagram and Classes (represent Inheritance, Aggregation and Association)
   5. Sequence Diagram
   6. Data flow diagram
   7. Metric calculation
2. **IMPLEMENTATION**
   1. Tools Introduction
   2. Technology Introduction
   3. Overall view of the project in terms of implementation
   4. Explanation of Algorithm and how it is been implemented
   5. Information about the implementation of Modules
3. **TESTING**

**7.1** Results and Snapshots

1. **CONCLUSION & SCOPE FOR FUTURE WORK <No Page Numbers>**
2. **REFERENCES**

**Declaration**

We, Raja Andukondan(1MS12CS083) and Prithvi Sharan S(1MS12CS078) of VII Semester B.E in Computer Science and Engineering, hereby declare that this project report titled "**BLUETOOTH BASED PRESENCE DETECTION**" submitted to M S Ramaiah Institute of Technology during the academic year of 2015-2016 is, to the best of our knowledge, a record of original work done under the guidance of Prof. K G Srinivas, Head of Department, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore. The results embodied in this paper have not been submitted to any other University or Institute for any purpose.

**Acknowledgements**

We would like to take this chance to sincerely thank our educational institution, M S Ramaiah Institute of Technology, for giving us this wonderful opportunity to work on the research paper and implementation, in lieu of course credits. We would also like to thank our mentors, Prof. K G Srinivas, Prof. Anita Kanavalli and Asst Prof. Sowmya B J for their constant support and guidance in every phase of the implementation, without which it wouldn't be possible.

**INTRODUCTION**

Bluetooth is an excellent option which can be used to improve the way to detect the way in which objects and even people can be detected. Any device that is Bluetooth enabled can be detected by another Bluetooth enabled device, and this property can be exploited to measure proximity of the two devices. The implementation aims to use Bluetooth enabled mobile device in order to detect the presence of individual based on the proximity from the central Bluetooth module interfaced on the Raspberry Pi system on chip.

Bluetooth, a wireless technology standard for exchanging data over short distances uses short wavelength Ultra High Frequency radio waves in the band from 2.4 to 2.485 GHz from fixed and mobile devices, in order to aid building personal area networks (PANs) was invented by a telecom vendor, namely Ericsson in 1994. It was originally proposed as a wireless alternative to the existing RS-232 data cables as it can connect several devices, effectively overcoming problems of synchronization.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), that has more than at least 20,000 partner companies, mainly in the areas of telecommunication, computing, networking, and most importantly consumer electronics. IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains this standard. The Bluetooth SIG monitors the development of the specification, manages the qualification program, and protects against trademark and its violations. Manufacturers must make any device meet Bluetooth SIG standards to market it as a Bluetooth enabled device. A network of patents apply to this technology, which are licensed to individual qualifying devices and their designers.

Raspberry Pi is a series of card–sized single-board computers designed and developed in the United Kingdom by the Raspberry Pi Foundation with the aim of promoting teaching of basic computing science in high schools and developing countries. Originally Raspberry Pi and Raspberry Pi 2 are developed in several board configurations using licensed and authorised manufacturing agreements with Newark Element14 (Premier Farnell), RS Components and Egoman.

The primary essence and core used to implement the chip's hardware remains the same across all manufacturers. All Raspberry Pi's include the same VideoCore IV GPU, and

either a single-core ARMv6-compatible processor or a newer and a more advanced ARMv7-compatible quad-core one (in Pi 2); and 1 GB of RAM (in Pi 2), 512 MB, or 256 MB. They have a Secure Digital (SD) (models A and B) or MicroSD (models A+ and B+) socket for boot media and persistent storage. Recently in 2014, the Raspberry Pi Foundation launched the Compute Module, for use as a part of embedded systems for the same computing power as the original Pi. Early in February 2015, the next-generation Raspberry Pi, Raspberry Pi 2, was released. The new computer board was initially available only in one configuration (model B) that had a quad-core ARM Cortex-A7 CPU and a GB of RAM with remaining specifications being similar to those of the previous generation model B+. The Raspberry Pi 2 retains the same price point of the model B, with the cheaper model A+ remaining on sale. In November 2015, the Foundation launched the Raspberry Pi Zero, a smaller product priced at an even lower price.

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC via the RISC OS image or the Brandy Basic clone for Linux C, C++, Java, Perl, Ruby, and Squeak Smalltalk.

**Problem statement:**

* The detection of the presence of people in a finite space is a real challenge. The applications of such a system is very vast. In this project, we attempt to detect the presence of people using the Bluetooth module on their smart phones as the sensor.
* Once implemented the system data can be aggregated in a database and used for other related analytical purposes.

**Goals of project:**

* To collect data from Bluetooth sensor and process the same.
* Store this data in a database in a suitable form.
* Make the data available on-demand using a cloud platform.
* Make available this cloud and data on a mobile platform and display the relevant information to the user.
* Authentication and access control.
* Ensuring robustness and real time data availability.

**Software Requirement Specification**

1. External Interface Requirements
   1. User Interfaces - The main user interface of the current implementation is a mobile friendly website using which the presence can be detected, using the college's LAN.
   2. Hardware Interfaces - The main hardware interface used in the current implementation is bridge interface between the external Bluetooth module and the System on Chip (Raspberry Pi). The interfacing is made via a USB port found on the Raspberry Pi module.
   3. Software Interfaces - The main software interfaces used in the current implementation are -

A) To ensure that the SoC is able to correctly identify and process the unique Bluetooth addresses of the mobile devices, a Python script is used named "bluez" is used.

B) To install the necessary scripts a terminal emulator or an ssh terminal is needed on the SoC.

C) For processing and parsing the received information on the local cloud platform, suitable PHP scripts are used. These are also used to store the results in a database for future analysis and convenient access purposes.

* 1. Communication Interfaces - The main communication interface used in the current implementation is the one occurring between the Bluetooth enabled device and the SoC, which are handled by suitable Python Scripts. Another minor communication interface occurs between the SoC and the hosting platform, which is taken care of by suitable PHP scripts.

1. Functional Requirements

2.1 To provide on-the-go real time lookup for presence of the required entity.

2.2 To make the findings available on a mobile friendly web based platform.

2.3 To provide secure access to the results by ensuring that only concerned end users are given access to the findings.

2.4 To ensure that the findings are periodically relayed to a database where it is stored for future analysis.

1. Software System Attributes
   1. Reliability - Since the calculations are performed directly on the raw data, with minimal processing and filtering, the accuracy and reliability of the information is of a high degree.
   2. Availability - The findings are hosted on the campus's LAN thereby guaranteeing on-the-fly availability, provided the campus's LAN architecture is fully functional.
   3. Security - Since the website to display the findings is hosted on the campus LAN, it is virtually inaccessible to the external world, via the Internet, assuming it doesn't have a public IP address assigned to it. Also, further restrictions to intra campus usage can be made by registering MAC addresses allowed to view the webpage within the campus, but this as of now is a future enhancement.
   4. Portability - Portability is not a key feature in the current implementation due to more or less static working conditions. However, the overall implementation is easily portable to multiple locations and scenarios, thereby also giving a scalable solution.
   5. Maintainability - Low maintenance is needed for the setup. The main areas where maintenance costs may be incurred are to maintain the SoC and the external Bluetooth module. Maintenance of the website is not of great concern due to its simplistic design.

* 1. Performance - One of the key parameters of this implementation is a good performance. Using the best in class SoC and other modules and good programming techniques virtually guarantees good performance at reasonable workloads.

1. Performance Requirements -

The performance requirements for the implementation are defined using the following key parameters -

Response Time - The Bluetooth enabled mobile devices are monitored continuously and the findings are reported at periodic intervals or as soon as any change in state occurs. The response time is a good 25% extra from the time of occurrence of the event. For example, if an event occurs at time T=1, then the event will be updated on the website at T=1.25.

Workload - The performance is at its peak optimal level, assuming the number of entities monitored is 10. Anything lower, makes it economically unfeasible and anything higher may potentially dampen performance. Therefore, the key workload is 10 devices/SoC.

Scalability - The current implementational model is suitable for a small area of ideally 10 people. However, using better hardware ensures high scalability.

Platform - The main delivery platform is a mobile friendly website hosted on the campus LAN. Even if multiple instances of the sensing devices are used, they can share a common platform to deliver their findings making it a feasible solution.

Keeping in mind the above performance characteristics, the overall performance of the implementation is good.

1. Database Requirement -

Database is mainly used for an analysis purpose by storing the periodic updates. From a functionality point of view, usage of the database is not a core necessity but only a luxurious add on. Minimal permissions such as creating snapshots, making external backups and recovery options are only needed for the implementation. The database design is also simplistic as it has only a single table, therefore reducing normalization needs.

1. Design Constraints -

The hardware selected for the implementations is a Raspberry Pi 2 SoC and compatible a compatible Bluetooth module. The scripts running in the implementation are mainly written in Python whereas the scripts to

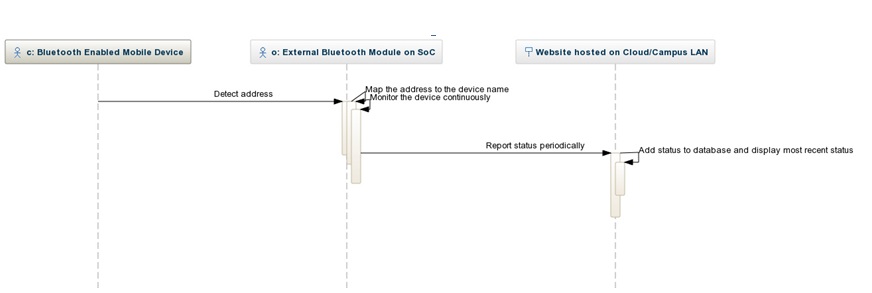
display the information in the website are written in PHP.

1. Other Requirements -

The 2 main other requirements are a unique private IP address and/or domain name to host the website of the application, and the permission to monitor the Bluetooth enables mobile device.

**DESIGN**

**Sequence Diagram:**

****

**Data Flow Diagram:**

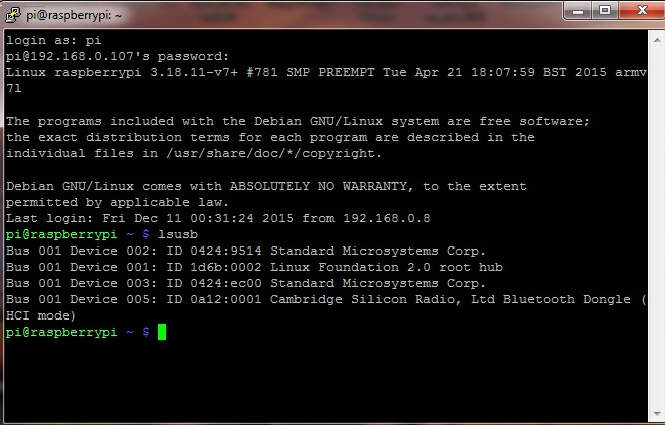
****

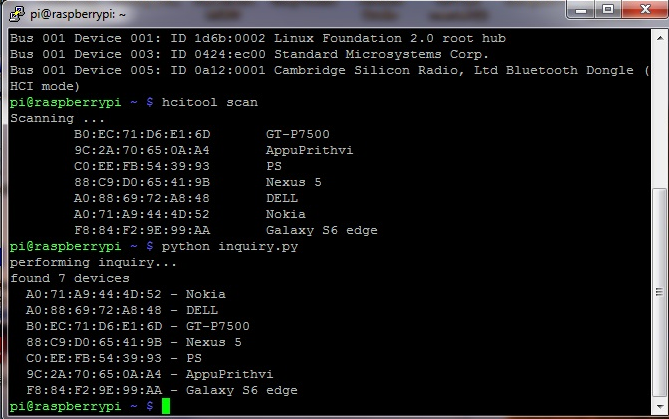
**Overall picture:**

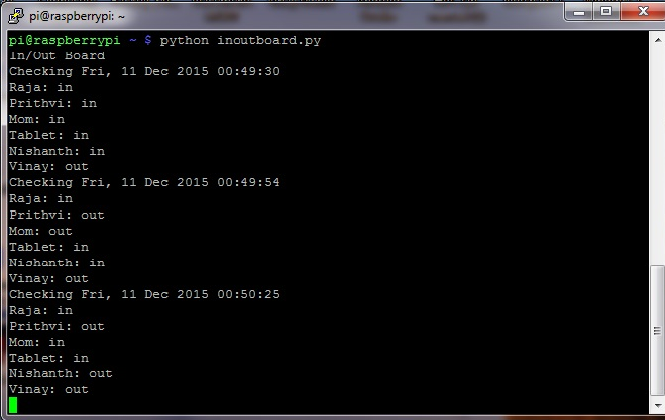
**IMPLEMENTATION**

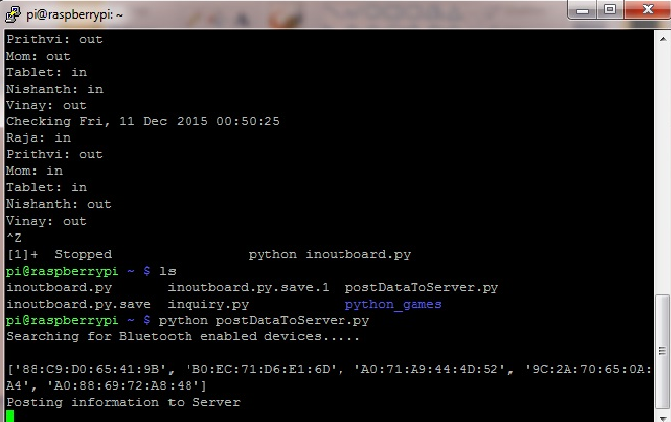
* The Bluetooth dongle was interfaced with the Raspberry Pi SoC.
* The python code to perform the scanning and computation of scan results is run on the Raspberry Pi.
* With additional python, php and json structures the addresses detected are mapped to the people and are written onto a webpage running on a server.

TESTING

**Snapshots:**

****

****

****