Bus Arrival Detection using Bluetooth Low Energy

Ankit Kumar, Abhinav Mathur, Akhilesh Kumar, Naveen Aggarwal
UIET, Panjab University
Chandigarh, India

***Abstract*—Making transport network of a city smarter is one of the major tasks when it comes about making a city smart. Public transport becomes a major focus area as it facilitates a major fraction of commuters in any city on a day to day scenario. So efficiently tracking the position of a public transport for e.g. tram cars, buses can give valuable information that can further be used to regulate city traffic. The aim of this paper is to track the position of a bus at a bus stop using Bluetooth Low Energy (BLE) beacons and calculate estimated time of arrival (ETA) at the subsequent bus stop based on current location traffic congestion. Bluetooth Low energy beacons will be used for bus detection and getting an accurate estimate on traffic congestion. This proposed approach is further implemented using Raspberry pi on which various algorithms keep a track of nearby beacons and determine real time traffic congestion. Also speech notification about the bus details is generated for the commuters on the bus-stop. And initial results suggests that the proposed approach can be used with a good accuracy in different environments.**

Keywords-ETA, BLE, iBeacon, RSSI.

I. Introduction

Cities are getting bigger both in area and demographic and ever-since there has been an increase in population on roads. As the primitive traffic light systems are unable to cater to the demands of present day traffic regulation, there is a need of new dynamic traffic regulatory system [1]. A better traffic regulation system can be implemented if critical real time data such as traffic congestion at various places is known. Also as cities are getting smarter and a smart transport network should involve easy availability of information to the user. All these features make tracking location of public transport such as tram cars, buses essential. As buses contribute to a big fraction of city traffic that involves maximum number of commuters, tracking a bus can provide valuable insights about traffic congestion which can be further used to regulate traffic [2]. Simultaneously the location and estimated arrival time like data can be made available to commuters on a real time basis. Several methods have been used to track a vehicle at a bus stop that includes GPS, RFID, camera etc. GPS based system suffers from high location error. In android based smartphones, this error is reduced using other sensors such as Wi-Fi and GSM. But in a system which has only GPS sensor, this error is very high. Some researchers have also proposed Vision sensor based approach to detect the arrival of bus. But this requires high installation and maintenance cost. Further vision sensor based approach do not work effectively at night and is affected by weather conditions such as rain, fog etc. RFID based solutions can be another way to provide solution for this problem. But low cost RFID systems have very low range and as range increases its cost increases exponentially. This paper focuses on using Bluetooth low energy as an alternative for detecting a bus arrival at a bus-stop. And subsequently generating notification about the bus details to commuters.

The detection process can be understood with the help of Figure 1. The picture presents top view of a single lane. With raspberry pi installed at bus stop and the scanning process scans each BLE device in range of about 75 meters. When the bus comes within this detection range it is validated from the database. Then the system keeps track of RSSI of the bus beacon. Once the bus beacon has RSSI above -50dB, details regarding the bus is announced at the bus stop. Figure 1 shows all the three stages i.e. arrival, detection and notification generation.

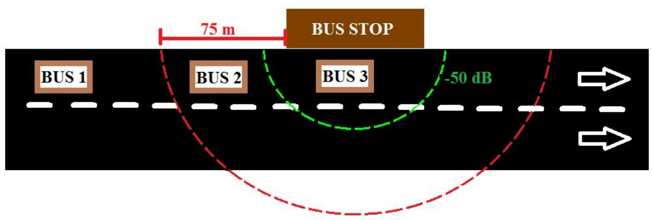

Fig. 1. System Overview

The notification generated is in the form of an audio output that announces the details about the bus stop. Also based on the rate at which RSSI level of the bus changes, congestion in the path of the bus can be determined. This can further help to establish ETA at subsequent bus-stops. Also the system is flexible for future upgrades and modifications.

The main advantage of the proposed system is simplicity, easy implementation and monitoring. Due to long BLE range there is no effect of parameters such as speed, weather conditions on bus detection. Also the same technology is being used for traffic congestion estimation reducing any additional system overhead.

The main challenges in implementation involves tracking the bus for a sufficiently longer distance. In the proposed approach, we used the average rate of change in attenuation level, rather than raw values to achieve better accuracy. Different tests have been performed to validate the system robustness and efficiency. These tests include

- Detection range estimation of Bluetooth beacon
- Variation of attenuation with distance Bluetooth beacon.

These tests have been performed in open and mildly crowded area.

Main contributions of the paper are:

- An approach for efficient detection of bus arrival at a bus stop in different experimental conditions.
- Optimized process for bus identification, validation and notification.

Subsequent sections of the paper is organized as follows: Section II represents the related work previously done in this field. Section III discusses about the proposed solution and the architecture of solution implemented along with the description of software and hardware components used in the prototype. It also explains communication protocol design which we are using for bus identification. Section IV discusses the experiments conducted along with the results. Finally, Section V concludes the paper.

II. RELATED WORK

There has been a lot of work done in different technologies for efficiently tracking vehicles. Researchers in these areas have shown a great deal of interest with ever advancing technology. A brief discussion on these technologies to locate a vehicle based on their cost, practicality etc. is given below.

A. Global Positioning System

The systems using GPS to track the bus positions are being widely used. The system generally consist of users and bus operators subscribing to the portal. Bus operators can manage the information about bus routes and bus stops on this and the data is managed on a server. In this system bus detection is done by GPS system and then the bus location data is sent to a server by Global Positioning Radio System in every 5-10 seconds, After analyzing the bus information and their location the users connected to the server are alerted through SMS using internet. In this way users get to know about the current location of the bus and the estimated time of arrival.

GPS systems are costly. The system is not efficient because when vehicle detection is carried out by GPS receiver is not much accurate and error in GPS detection is about 30-50 meters. Although complex algorithms and features such as incorporating GSM data along with to improve error margin, the system remains in-efficient [3].

There are high chances of error in case of multi-lane transportation when bus is on the some other lane not having the bus stop. Also various factors such as server delays impact the real time feature of this system [4].

B. Vision Sensor

In this system detection and identification of bus is done by using real time image processing. The system matches the grabbed image of bus by already uploaded images and in this way detection occurs. Using a camera is the easiest way to detect bus at bus stops but at the same time system becomes complex and demands high maintenance [5].

The orientation of the camera placed at bus stops plays an important role and when disturbed the system becomes inefficient and results in erroneous output. So there is a need of regular checking .Also the camera aperture has to be clean and proper camera calibration has to be done for the system to work fine. There are solutions for these drawbacks such as self-aligning camera system etc. which are far more complex for such a simple system. This system requires very high cost of maintenance [6] [7].

C. Radio Frequency Identification(RFID) Sensors

This system uses electromagnetic fields for automatic identification and detection of bus through the tags attached on it. The range of RFID is low for practical purpose and high range RFID are highly expensive compared to other systems discussed above [8].

The RF gives RFID its strength to this system but also act as one of the disadvantages of this system because RF fields get affected by metal yielding very poor reliability [9].

Bluetooth Low Energy (BLE), which was introduced as part of the Bluetooth 4.0 specification, is a wireless technology that gives developers unprecedented access to external hardware and provides hardware engineers with easy and reliable access to their devices. These devices can operate in two modes: either by Broadcasting or being directly connected to a central device. The broadcast mechanism is one of the big differences between Bluetooth LE and classic Bluetooth as it allows data to be sent out by the peripheral to any device in range.

Bluetooth LE peripheral device doesn't necessarily need to be paired with a central device in order to make data transfer. In Bluetooth LE, we speak of it as connected rather than paired as we did with Bluetooth 2.1. In broadcast mode, the peripheral will periodically send out advertising packets, available to any device that's looking for them, for devices acting as observers [10].

III. PROPOSED SOLUTION

Bus arrival detection using BLE consists of a Bluetooth beacon attached to each bus to be covered in the system. These are highly efficient long range beacons. This beacon will be given a unique SSID. A database that maps these SSIDs with the MAC addresses of corresponding beacons will be maintained at each of the bus stop receivers. The receiver system at the bus stops consist of a raspberry pi 3 and speakers.

These receivers at bus stops keep on simultaneously scanning for Bluetooth beacons. Once a Bluetooth beacon is available its SSID is verified with its corresponding MAC address. Next, RSSI value is monitored to have an idea about the proximity of the bus with the bus stop. Once the bus arrives closer to the bus stop an audio notification is generated at the bus stop. The basic architecture has been shown in Figure 2.

A. Architecture and Methodology

The basic architecture consists of iBeacons in Broadcast mode advertising their data and Raspberry Pi in observer mode scanning for available devices as shown in Figure 2.

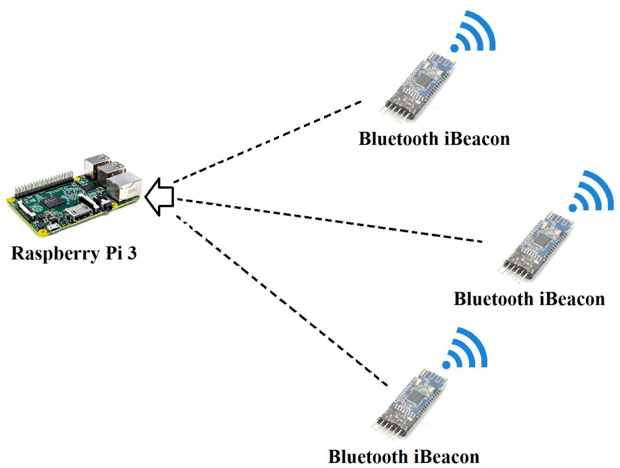

Fig. 2. Architecture

Bus arrival detection using Bluetooth is aimed to develop a precise and economical technique by which a bus arrival at the bus stop is detected.

The system will be operated with Raspberry Pi 3 and Bluetooth low energy (BLE v4.0) iBeacons due to their special broadcasting features, wide operable distance range and low power requirements.

The Raspberry Pi has an ability to verify multiple Bluetooth devices at a time. Therefore multiple buses approaching the bus stop can be uniquely identified by the system [13].

The various electronic equipment used in this prototype are:

1) Hardware Equipment

a) Raspberry Pi 3: The Raspberry Pi 3 is the third generation Raspberry Pi. It has replaced the Raspberry Pi 2 Model B in February 2016. It consists of in-build Bluetooth 4.0 which keeps on looking for a Bluetooth device nearby. The database of all the buses is stored in it. A 1.2 GHz 64-bit quad-core ARMv8 CPU speeds up the process of authentication [11].

b) Tinysine Bluetooth 4.0 BLE module: The Bluetooth module is placed at the front side of the bus. The Bluetooth module broadcast the signal in the surrounding environment. The working frequency of the module is 2.4 GHz ISM band and Gaussian Frequency Shift Keying modulation technique is used in it. The transmitted signal is highly secured by in-built encryption technique. The size of the module is very compact and the broadcast range is upto 75 meters. The power consumed by the module is very low and it is in the range of 400 uA~1.5 mA for the sleep mode and 8.5 mA for active mode.

2) Software Equipment:

a) Raspbian Jessie: Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi [14].

b) PuTTY: PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is an open source software that is available with source code and is developed and supported by a group of volunteers. The software was used for remote terminal access of Raspberry Pi.

c) NotePad++: Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GPL License. The software was used to edit codes on Raspberry Pi via SFTP (using NppFTP plugins).

The Bluetooth iBeacon installed in the bus continuously keeps advertising its various parameters. The transmitted signal includes SSID and MAC address which is unique in case of every Bluetooth iBeacon. The Raspberry Pi 3 installed at the bus stop continuously runs a program in it which scans for the available low energy Bluetooth devices present around it. The algorithm used is described in the next section with the help of flow chart. Also a pseudo code is provided in Figure 4.

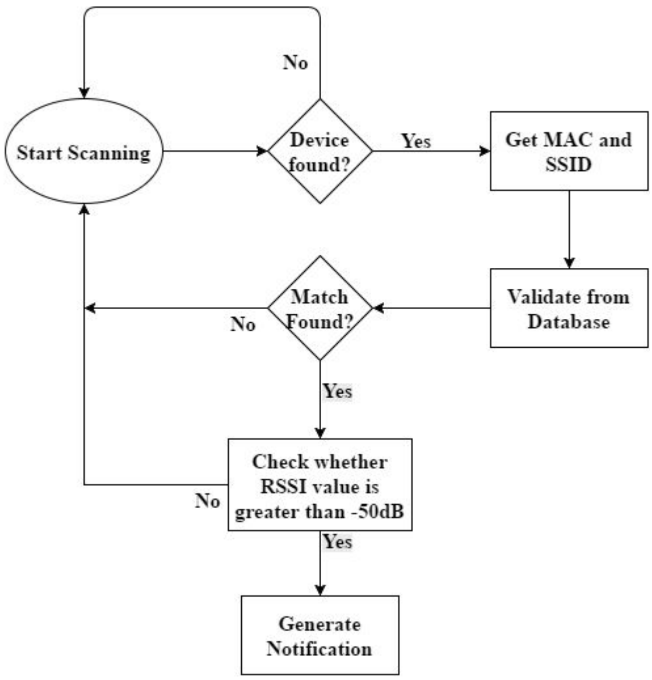

Fig. 3. Methodology

The whole process as presented in Figure 3 is explained as follows:

1) The operating system in the Raspberry Pi 3 continuously executes a program in it which checks for the available Bluetooth low energy devices present around it.

2) If a device is found, SSID and MAC are fetched from its advertising packet.

3) To ensure that the BLE device which arrived at the stop is authentic, MAC and SSID are table matched from a database maintained on Raspberry Pi.

4) The database also consists of all the information related to bus like bus number, bus route etc. After validation from the database a notification is generated at the bus stop announcing the bus details and the whole process repeats.

Algorithm 1 Bus Detection

1. **Procedure** scan_Devices()
2. **If** device_found()==0
   →go to step 1
3. **Procedure** *mac*=get_MAC()
4. **Procedure** *ssid*=get_SSID()
5. **Procedure** *val*=validate_data(mac,ssid)
6. **If** *val*==0
   →go to step 1
7. **Procedure** *rssi*=get_RSSI()
8. **If** *rssi* ≥-50 dB
   →go to step 1
9. **Procedure** generate_Notification()

Fig. 4. Algorithm of Bus Detection

B. Communication Algorithm Design

The valid detection of the bus is one of the challenging tasks in this whole process. The whole system needs to be protected from external spoofing at the same time keeping the system lite and robust. This leads to the need of an efficient protocol for the communication between iBeacon device placed in the bus and raspberry pi at the bus stop. With no protocol incorporated communication, it becomes very easy for an unauthorized person to spoof the whole system. The protocol used must not be easily decoded and should be easily upgradeable as per any future changes if required.

The protocol used here executes a set of instructions to generate a unique SSID to each iBeacon device. This is performed by applying some mathematical operations on MAC and a 4-alphanumeric digit key. This unique SSID is then transmitted along with the MAC address by the iBeacon device. In Raspberry Pi, a database is maintained which includes complete set of information for all the buses. The key attribute of every bus in the database is its unique key. As the information is received by the Raspberry Pi, again mathematical operation is performed decrypting the received MAC and SSID. This reverse operation generates the 4-alphanumeric digit key. This key is then mapped in the database. If it matches with a valid key present in the database, then an announcement will be made at the bus stop. If the key does not match with any valid key present in the database, it means that the iBeacon device is not valid. The whole process is presented as a block diagram in Figure 5.

The protocol that we used here has an ability to serve 9999 different buses, i.e., 9999 unique SSID can be generated using the encryption protocol. This protocol is easily upgradable to serve more SSIDs if required in the future. The process is pictographically represented in figure below.

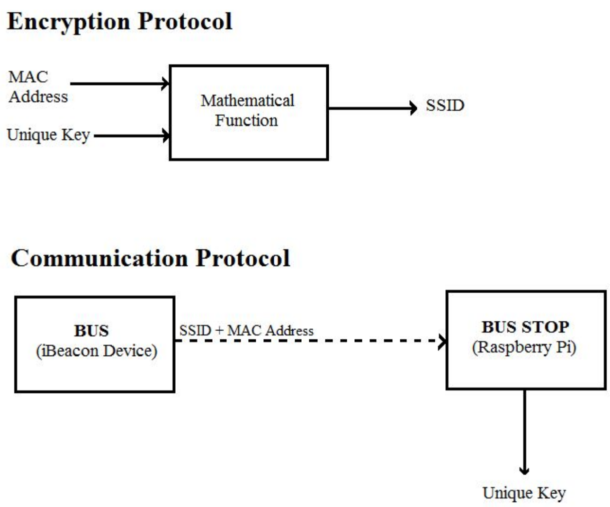

Fig. 5. Communication Protocol

IV. EXPERIMENTS

All tests has been performed on Tinysine Bluetooth 4.0 BLE module, hence results can change when module is replaced.

A. Test Parameters

•Broadcast power=6dbm (0.00398watt)
•Beacon mode: Broadcast only
•Power: 3.3v DC
•Advertising interval=100ms

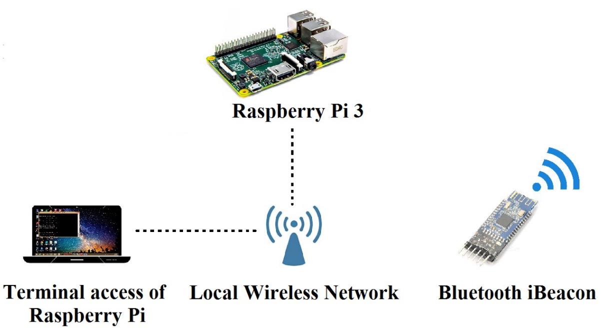

Fig. 6. Experimental Setup

All the experiments are carried out by connecting the Raspberry Pi 3 and personal computer in a local ad-hoc wireless network as shown in Figure 6. A static IP address is provided to the Raspberry Pi by the local wireless network. This IP address is used to get the terminal access to the Raspberry Pi from the personal computer using PuTTY software. All the programs running on the Raspberry Pi are monitored through this software [12].

B. Experiment 1: Range detection

The aim of this experiment was to determine the range of Bluetooth beacon being used in the project.

Setup: The beacon was kept at various regular distances from the receiver raspberry pi. The raspberry pi is continuously scanning for beacon signal. The observations were monitored using the terminal access of raspberry pi.

The experiments were repeated for different orientations and environment such as open arena, moderately crowded area and similar observations were made. Hence we conclude the experiment with the following result: "The viable distance of operation has been found to be 75m".

A. Experiment 2: Variation of Signal strength attenuation vs distance

The aim was to determine how the signal strength of the beacon varies with distance.

Setup. The beacon was kept at a known distance from the receiver and a code at the receiver raspberry pi scans and records concurrent values of RSSI. Hence we conclude the experiment with the following result: "It was observed that at for RSSI to show a variation for a radius of 10m from the receiver. After which the RSSI value saturates." A graph of the same plot has been shown in Figure 7. The plot is done between Mean, Min and Max values of RSSI attenuation vs. distance.

TABLE I: Results with variation of max, min and mean values of signal attenuation with distance.

S.No.	Parameters			
	Distance (m)	*Mean (dB)*	*Max (dB)*	*Min (dB)*
1.	0	-43.3	-34	-58
2.	5	-70.9	-64	-84
3.	10	-87.2	-74	-102
4.	15	-90.5	-78	-103
5.	20	-88.9	-80	-98
6.	25	-85.4	-77	-92
7.	30	-82.6	-76	-93
8.	35	-81.9	-77	-90
9.	40	-88.0	-81	-96
10.	45	-89.5	-76	-100
11.	50	-88.9	-81	-99

B. Test Results

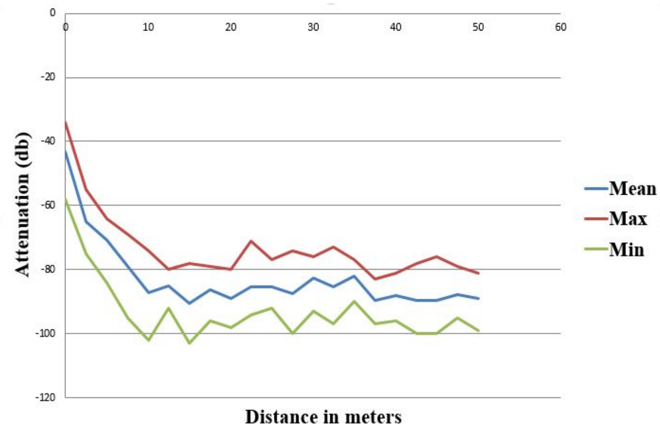

Fig. 7. Test Results

V. CONCLUSION

In this paper, we have presented an approach to detect a bus arrival at bus-stop using Bluetooth low energy. This system involves installation of Bluetooth beacons in the Buses and raspberry pi 3 as receivers at the bus-stops. Raspberry pi at each bus-stop maintains a local database containing the details of all beacons corresponding to their respective beacons. This enables notification about the arrived bus to be generated. BLE as the name indicates consumes very less power also no pairing is needed for communication which allows the system to handle as many bus at a given time, increasing the robustness of the system.

This system works fine with the arrival detection of bus at the bus stop. There has been no error in detection observed so far because of parameters such as speed of vehicle, weather conditions etc. Due to high detection range, the process of detection is rapid and its effectiveness is not deterred in different experimental conditions. Also using the BLE module mentioned above we observed a saturation in RSSI values after 10m as mentioned in Experiment 2. The plot of Attenuation (in dB) vs. Distance (in meters) is shown in Figure 4. Thus the distance is inappropriate for making an inference or comment on traffic congestion on the road. Hence the current BLE module used setup is not appropriate for ETA estimation. Future work would include experiments with different BLE modules and optimization of current algorithms for Scanning devices and communication protocol to make traffic congestion estimation possible which will further aid in algorithms for ETA.

VI. ACKNOWLEDGMENTS

This project is sponsored and supervised by Design Innovation Centre, Panjab University, Chandigarh.

REFERENCES

[1] Campos, Leonardo Barreto, Carlos Eduardo Cugnasca, André Riyuiti Hirakawa, and José Sidnei C. Martini. "Towards an IoT-based system for Smart City." In *Consumer Electronics (ISCE), 2016 IEEE International Symposium on*, pp. 129-130. IEEE, 2016.

[2] Figueiredo, Lino, Isabel Jesus, JA Tenreiro Machado, J. Ferreira, and JL Martins de Carvalho. "Towards the development of intelligent transportation systems." In *Intelligent transportation systems*, vol. 88, pp. 1206-1211. 2001.

[3] Čapkun, Srdjan, Maher Hamdi, and Jean-Pierre Hubaux. "GPS-free positioning in mobile ad hoc networks." Cluster Computing 5.2, 2002.

[4] Nagadapally, A. Srikanth Reddy, B. BalaKumar Krishnaswamy, C. Abhigna Reddy Shettap, D. Devireddy Prathika, and E. Roger Lee. "A Step Ahead To Connect Your Bus-With GPS." In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, p. 171. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.

[5] Russ, John C. The image processing handbook. CRC press, 2016.

[6] Deutscher, Jonathan, Michael Isard, and John MacCormick. "Automatic camera calibration from a single manhattan image." In *European Conference on Computer Vision*, pp. 175-188. Springer Berlin Heidelberg, 2002.

[7] Parthasarathi, V., M. Surya, B. Akshay, K. Murali Siva, and Shriram K. Vasudevan. "Smart control of traffic signal system using image processing." *Indian Journal of Science and Technology* 8, no. 16 ,2015.

[8] Bolic, Miodrag, David Simplot-Ryl, and Ivan Stojmenovic, eds. RFID systems: research trends and challenges. John Wiley & Sons, 2010.

[9] Mohd Zikrul Hakim Noor, I. Ismail and M. F. Saaid, "Bus detection device for the blind using RFID application," 2009 5th International Colloquium on Signal Processing & Its Applications, Kuala Lumpur, 2009.

[10] Gomez, Carles, Joaquim Oller, and Josep Paradells. "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology." Sensors 12.9 ,2012.

[11] Monk, Simon. Programming the Raspberry Pi: getting started with Python. McGraw Hill Professional, 2015.

[12] Maksimović, Mirjana, "Raspberry Pi as Internet of things hardware: performances and constraints." design issues 3, 2014.

[13] Newman, Nic. "Apple iBeacon technology briefing." Journal of Direct, Data and Digital Marketing Practice 15.3, 2014.

[14] Harrington, William. Learning Raspbian. Packt Publishing Ltd, 2015.