

## **BLE supported indoor location**

Subtitle

**Ricardo Pacheco Pais Martins**

Thesis to obtain the Master of Science Degree in  
**Electrical and Computer Engineering**

Supervisor(s): Prof./Dr. Lorem Ipsum

### **Examination Committee**

Chairperson: Prof. Lorem  
Supervisor: Prof. Lorem Ipsum  
Co-Supervisor: Prof. Lorem Ipsum  
Members of the Committee: Dr. Lorem Ipsum  
Prof. Lorem Ipsum

**March 2017**



***Anyone who has never made a mistake has never tried anything new.***

Albert Einstein



# Acknowledgments

I would like to thank the Academy,laura, jnos, pais, pais da laura, leal , almeida etc... bla bla bla..



# Abstract

The Objective of this Work ... (English)

# Keywords

Keywords (English)





# Resumo

O objectivo deste trabalho ... (Português)

# Palavras Chave

Palavras-Chave (Português)



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	State of The Art . . . . .	2
1.2.1	Dummy Subsection A . . . . .	2
1.2.2	Dummy Subsection B . . . . .	2
1.3	Original Contributions . . . . .	2
1.4	Thesis Outline . . . . .	2
<b>2</b>	<b>A Chapter</b>	<b>3</b>
2.1	Bluetooth Low Energy . . . . .	4
2.1.1	Bluetooth Low Energy (BLE)'s Architecture . . . . .	4
2.1.1.A	Low Energy (LE) Controller Group . . . . .	5
2.1.1.B	LE Host Group . . . . .	5
2.1.2	BLE Profiles . . . . .	6
2.1.3	Communication Topology and Operation . . . . .	7
2.1.3.A	LE Piconet Topology . . . . .	9
2.1.3.B	Operational Procedures . . . . .	9
2.2	Section B . . . . .	11
2.2.1	Subsection A . . . . .	11
2.2.2	Subsection B . . . . .	11
<b>3</b>	<b>Conclusions and Future Work</b>	<b>13</b>
	<b>Bibliography</b>	<b>15</b>
	<b>Appendix A Title of AppendixA</b>	<b>A-1</b>



# List of Figures

2.1	BLE Architecture . . . . .	4
2.2	Gatt-based profile hierarchy . . . . .	7
2.3	Advertising event . . . . .	8
2.4	Connection Event . . . . .	8
2.5	LE Topology . . . . .	9



# List of Tables

2.1 Dummy Table. . . . .	11
--------------------------	----





# Abbreviations

**acro** acronym

**BLE** Bluetooth Low Energy

**LE** Low Energy

**IoT** Internet of Things

**SMP** Security Manager Protocol

**PHY** Physical

**QoS** Quality of Service

**L2CAP** Logical Link Control and Adaptation Protocol

**HCI** Host Controller Interface

**P2P** Peer-to-Peer

**ATT** Attribute Protocol The Attribute Protocol

**GATT** Generic Attribute

**GAP** Generic Access Profile

**FDMA** Frequency Division Multiple Access

**TDMA** Time Division Multiple Access



# List of Symbols



# 1

## Introduction

### Contents

1.1	Motivation	2
1.2	State of The Art	2
1.3	Original Contributions	2
1.4	Thesis Outline	2

## **1.1 Motivation**

Motivation Section.

## **1.2 State of The Art**

State of The Art Section.

### **1.2.1 Dummy Subsection A**

State of Art Subsection A

### **1.2.2 Dummy Subsection B**

State of Art Subsection B

## **1.3 Original Contributions**

Contributions Section.

## **1.4 Thesis Outline**

Outline Section.

# 2

## A Chapter

### Contents

2.1 Bluetooth Low Energy . . . . .	4
2.2 Section B . . . . .	11

*Present the chapter content.*

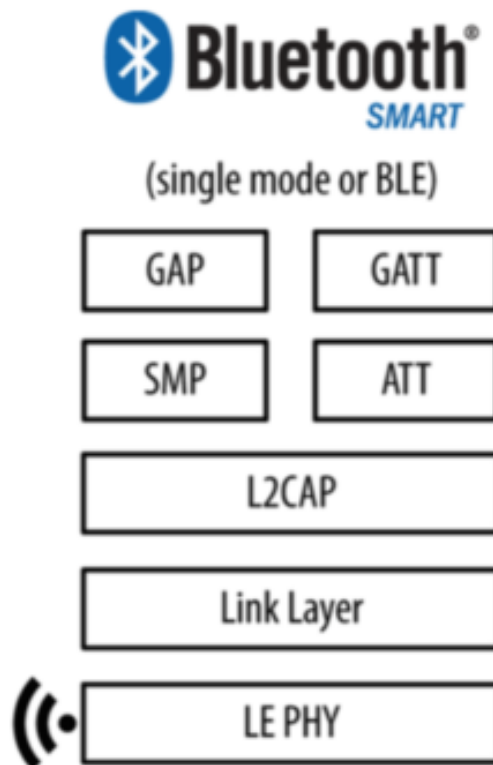
## 2.1 Bluetooth Low Energy

Bluetooth is a wireless technology that was created in 1994 with the objective of replacing cables connecting fixed or portable devices. At this point in time Bluetooth Special Interest Group is in charge of developing and managing this technology characterized by its robustness, low energy consumption and low cost.

The Bluetooth Low Energy (BLE) protocol was introduced with the Bluetooth Core Specification version 4 (also called Bluetooth Smart) circa 2010 alongside two other protocols. Out of the three, BLE stood out for its lower power consumption, lower complexity and lower cost, while allowing for device discovery, connection establishment and connection mechanisms. Due to its characteristics, the BLE protocol was utilized in various Internet of Things (IoT) applications.

### 2.1.1 BLE's Architecture

Bluetooth's Architecture is everchanging and can become very complex rather quickly with the introduction of different types of protocols. When working with BLE it's important to understand the key components of its architecture because by doing it's possible to better analyze the role of each component and how they operate and depend on each other. There are two main groups of core blocks, the Low Energy (LE) Controller and the LE Host, in 2.1.1.A and 2.1.1.B respectively, and most the most relevant of these components will now be looked at.



**Figure 2.1:** BLE architecture



### 2.1.1.A LE Controller Group

**Physical (PHY) Layer** - Architectural block responsible for all Bluetooths' communication channels on the 2,4GHz radio. Receiving and transmitting packets and supplying information crucial for controlling its timing and frequency through the baseband block.

**Link Layer** - Architectural block responsible for managing logical links between BLE devices. It can create and release connections, update connection parameters related to PHY links. It's responsible for the discovery and consequently connection procedure and also sending and receiving data.

**Device Manager** - Architectural block responsible for controlling the general behaviour of the Bluetooth device. This block is responsible for all operations that aren't directly related to data transportation. Some of its operations are: inquiring for the presence of nearby BLE devices; connecting to a BLE device; setting whether or not its local device is discoverable and/or connectable by the others; controlling device behaviour such as managing own's local name or stored keys.

**Baseband Resource Manager** - Architectural block responsible for all access to the radio medium, this means access to the PHY channels. It has two purposes, first to negotiate contracts with the entities that wish to use the medium and second to act as a scheduler on the same radio medium, granting the entities with said contracts, a time window in which they can utilize the medium. A contract is basically a commitment to deliver a certain Quality of Service (QoS) on the user application.

**Link Controller** - Architectural block responsible for the encoding and decoding of Bluetooth packets from the data payload and parameters related to the physical channel, logical transport and logical link. It also carries out the Link Layer protocol in conjunction with Baseband manager's scheduling function to communicate flow control and acknowledgement and retransmission request signals.

### 2.1.1.B LE Host Group

**Logical Link Control and Adaptation Protocol (L2CAP)** - Architectural block responsible of transmits packets to the Host Controller Interface (HCI) or directly to the Link Layer in hostless systems. It allows for higher-level protocol multiplexing, packet segmentation and reassembly, and the conveying of QoS information to higher layers.

**Channel Manager** - Architectural block responsible for creating, managing and closing L2CAP channels used in transport of service protocols and application data streams. The local Channel Manager makes use of the L2CAP protocol to communicate with a peer's Channel Manager and together create L2CAP channels and connect their endpoints to the appropriate entities.

**Security Manager Protocol (SMP)** - Architectural block responsible for implementing the Peer-to-Peer (P2P) protocol that operates over its own dedicated L2CAP channel and generates encryption keys and identity keys. This block is also in charge of storing those same keys and making them available to the controller. These keys are later used in the encryption or pairing procedures.

**Generic Access Profile (GAP)** - Architectural block responsible for working in conjunction with Generic Attribute (GATT) to define the base functionality of BLE devices. The available services in this profile are: BLE device discovery, connection modes, security, authentication, association models and service discovery. GAP defines four different roles to describe a device, allowing for the controllers

to be optimized in function of the device's desired roles. **Broadcaster:** This role is optimized for transmitter-only applications. In a scenario in which a device supports this role it will make use of advertising in order to broadcast its data. The broadcaster role doesn't support for connections.

**Observer:** This role is optimized for receiver-only applications and it's complementary to the broadcaster role. It only receives broadcast data included in advertising packets and much like its counterpart, it doesn't support connections.

**Peripheral:** This role is optimized for devices that only want to support a single connection, allowing for a much less complex controller due to the fact that it only needs to support the slave role and not the master one.

**Central:** This role supports multiple connections and functions as the initiator for all of them. These connections are all made with Peripheral devices and its controller must support the master role in a connection and allow for more complex functions, in comparison to the remaining roles.

**Attribute Protocol The Attribute Protocol (ATT) Protocol** - Architectural block responsible for implementing the P2P protocol between an attribute server and client. This client/server communication happens in a dedicated fixed L2CAP channel. A server can send through this channel responses, notifications and indications, while the client can send requests, commands and confirmations. This block allows the clients to read and write values of attributes on a peer device acting as a ATT server.

**GATT Profile** - Architectural block responsible for creating a framework for the ATT, in which it is represented the functionalities of an ATT server. This profile describes the hierarchy of services, characteristics and attributes existent in the server and provides an interface for discovering, reading, writing and indicating of service characteristics and profiles. A more thorough description of profiles can be found in 2.1.2. GATT also defines two possible roles, which aren't directly tied to the GAP roles previously presented but can be specified by higher layer profiles. **Server:** A GATT server is responsible for storing data transported over the ATT and accepts ATT requests, commands and confirmations from a GATT client. It also sends responses to requests and, if implemented, send indication and notification asynchronously to a GATT client when specified events occur on the GATT server.

**Client:** A GATT client has all the functionalities presented in the GATT server description.

## 2.1.2 BLE Profiles

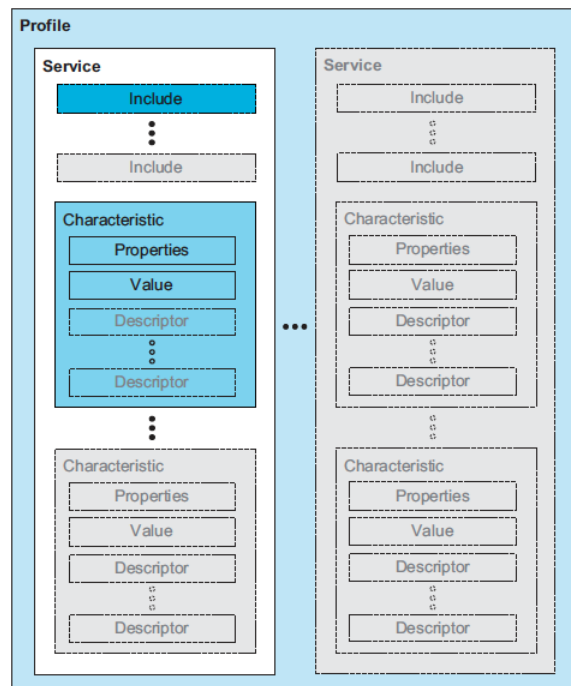
Bluetooth profiles defines the required functionalities of each layer, from the PHY to the L2CAP layer, as well as the the vertical interactions between layers and P2P interactions between device and a specific layer. Since a profile also defines application behaviour and data formats, we can say that when two devices comply with all the requirements of a Bluetooth profile, application interoperability is achieved. Each Bluetooth profiles describes its requirements necessary for devices to create a connection, to find available services and connection information required for making application level connections.

The base profile that any Bluetooth system needs to include is the GAP, already presented in 2.1.1.B. From this point, any additional profile implemented will be a superset of GAP, where GATT

is included. Among all that was already introduced about GATT in 2.1.1.B, it also specifies the profile hierarchy, or the structure in which profile data is exchanged. 2.2 shows the hierarchy in a Gatt-based profile, with the profile being the top level and services and characteristics below. The last two will now be presented individually:

**Service:** A profile is composed by one or more services. CITAÇÃO CORE PAGE 256 A service is a collection of data and associated behaviors to accomplish a particular function or feature of a device or portions of a device. It can be either primary, which provides primary functionalities of a device, or secondary, providing auxiliary functionalities of a device and is referenced from at least one primary service. A service is composed of characteristics and/or references to other services.

**Characteristic:** A Characteristic is a value that is used in a service that has properties and configuration information that describe how the value should be accessed as well as information on how to display the value. A characteristic is defined by its declaration, its properties, its value and may also be defined by its descriptor, which describes the value or permit configuration of the server relative to the value.

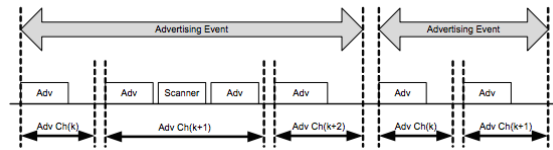


**Figure 2.2:** Gatt-based profile hierarchy

### 2.1.3 Communication Topology and Operation

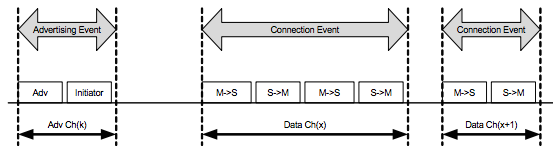
The LE radio operates at the 2.4GHz band and employs a frequency hopping transceiver to combat interference and fading. LE also employs two multiple access schemes: Frequency Division Multiple Access (FDMA) used to separate the 40 available PHY channels, 37 of them are used as data channels and the remaining as advertising channels and Time Division Multiple Access (TDMA) in a polling scheme that is used when one device transmits a packet at a predetermined time and a corresponding device responds with a packet after a predetermined interval.

The PHY channels are sub-divided into time units known as events and these can be of two types according to which type of channel they belong, either advertising events or data events.



**Figure 2.3:** Advertising event

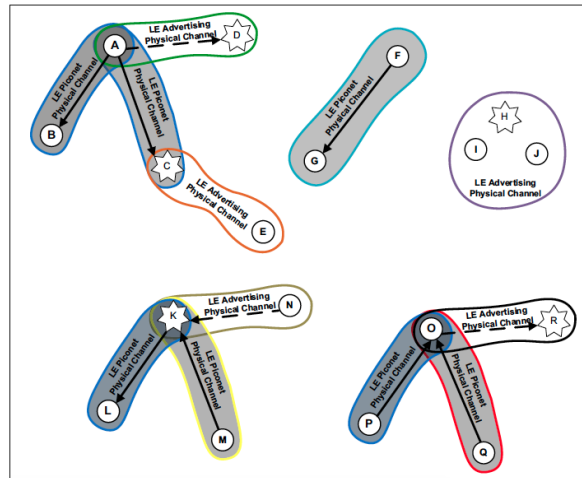
**Advertising events:** There are three roles that can be used to describe a device in function of their utilization of the channel: **advertisers**, are those that transmit advertising packets; **scanners** are devices that receive advertising packets without the intention of connecting with the advertising device; **initiators** are devices that listen for connectable advertising packets in order to later initiate a connection. Transmissions in the advertising channels occur in advertising events which always start with an advertiser sending a packet. Depending on the type of advertising packet, a scanner device may make a request to the advertiser which may be followed by a response from the advertiser, always on the same advertising PHY channel. The advertising PHY channel changes when the advertiser sends a new advertising packet. An advertising event can be terminated whenever the advertiser wants and when a new advertising event is created it will occur in the first advertising PHY channel. The whole process can be visualized in 2.3.



**Figure 2.4:** Connection event

**Connection events:** When an advertiser is using a connectable advertising event an initiator may request a connection on the same PHY channel. If the advertiser accepts the connection request, the advertising event ends and a connection event starts in order to establish the connection. Once it's established the initiator takes the master role and the advertised, the slave role. These events are used to transmit data between eachother and they always begin with a message from the master. During a connection event master and slave alternate send data packets on the same packet. The master is responsible for ending the end whenever he pleases and for the creation of new event channel hopping is required. The whole connection event can be visualized in 2.4.

### 2.1.3.A LE Piconet Topology



**Figure 2.5:** Examples of LE topology

As opposed to regular Bluetooth piconets, in LE, slaves can't share a PHY channel and as such each must have his own PHY channel to communicate with a master device. In order to best understand 2.5, solid arrows always point from master to slave, dashed arrows indicate a connection initiation where the arrow points from the initiator to the advertised using a connection event and devices that are advertising are indicated with a star. With these notes the different types of topologies can be analyzed:

- In piconet A, that which contains device A, there are multiple types of topologies such as , device A having connections with multiple slaves and initiating a new one with device D which was advertising. It is also worth noting that the device E is functioning as a scanner listening to the advertiser device C.
- In piconet F there is a simple master slave connection with device F as master and G as slave.
- In Group H there are more than two devices in a single PHY channel, such thing occurs since multiple devices can listen for advertisements on the same channel. In this case device H is functioning as an advertiser while I and J are scanners.
- In scatternet K there is an example where device K functions as master in the connection with device L and as a slave in the connection with device M, at the same time.
- In scatternet O there is an example where device O is functioning as a slave in both his connections with devices P and Q but it is still capable of forming a connection with R, where O will be the master.

### 2.1.3.B Operational Procedures

The most common operational mode of a Bluetooth device is when he is connected and exchanging data with another Bluetooth device. Since Bluetooth is an ad-hoc wireless communications tech-

nology, i.e. decentralized type of wireless network, there are a number of operational procedures that enable piconets to be formed so that the subsequent communications can take place.

**Device Filtering Procedure** Method used by controllers to reduce the number of devices requiring communication responses through the use of a "white list" located in the Link Layer that enumerates the devices that are allowed to communicate with the local device. This procedure allows the device reduce power consumption since it reduces the number of transmissions that it needs to make.

**Advertising Procedure** An advertiser utilizes this procedure to perform unidirectional broadcasts to devices in the vicinity. The broadcast occurs without any need of connections and can be utilized to establish connections with nearby devices or to simply broadcast information to nearby scanner devices. This procedure includes the operations already described in advertising events in 2.1.3.

**Scanning Procedure** A scanner device utilizes this procedure to listen to unidirectional broadcasts of user data sent by advertising devices. It is also capable of requesting additional user data by making a scan request as an answer on the same PHY channel of the first broadcast. This procedure can be utilized while the device is connected to other LE devices for as long as its connections requirements are maintained.

**Discovering Procedure** Bluetooth devices use the advertising procedure and scanning procedure to discover nearby devices, or to be discovered by devices in a given area, as such the discovery procedure is asymmetrical. A Bluetooth device that tries to find other devices in the vicinity can be called as a discovering device and will listen for devices advertising scannable advertising events. Devices that are available to be found and actively broadcast scannable advertising events are called discoverable devices.

**Connecting Procedure** The connecting procedure is asymmetrical as it requires one of the devices to be utilize the advertising procedure and the other one the scanning procedure. The advertising procedure has the capability of being targeted which allows only the chosen device to respond. The scanning procedure also has the capability of being target if the device discovers an advertising device and from there on out only listens for its advertisements. Upon receiving a connectable advertising event, it can initiate the connection by answering with a connection request.

[Connected Mode] Once the Connecting procedure is over the devices are physically connected to each other within a piconet. While in connected mode there is the possibility of changing the connection's properties such as data packet's length and for the device to utilize advertising, scanning or discovery procedures.

acronym (acro)

acronym (acro)

acro

acronym

acros

As seen in [1]. *Enfatizar*

Remember you can change the reference style. Another dummy citation [2].

## 2.2 Section B

### 2.2.1 Subsection A

The model described can also be represented as

$$\dot{\mathbf{x}}(t) = \mathbf{T}\mathbf{z}(y), \mathbf{y}(0) = \mathbf{y}_0, z \geq 0 \quad (2.1)$$

where

$$\mathbf{A} = \begin{bmatrix} -(a_{12} + a_{10}) & a_{21} \\ a_{12} & -(a_{21} + a_{20}) \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.2)$$

### 2.2.2 Subsection B

**Table 2.1:** Dummy Table.

Vendor Name	Short Name	Commercial Name	Manufacturer
Text in Multiple Row	ABC	ABC <sup>®</sup>	ABC SA
	DEF	DEF <sup>®</sup>	DEF SA
	GHF	GHF <sup>®</sup>	GHF SA
Text in Single Row	IJK	IJK <sup>®</sup>	IJK SA
Frescos SA	LMN	LMN <sup>®</sup>	LMN SA
Carros Lda.	Text in Multiple Column		





# 3

## **Conclusions and Future Work**



# Bibliography

[1] "Wikipedia," <http://www.wikipedia.org>, 2011.

[2] R. Dummy, "How to write a Latex Article," <http://www.biopsychiatry.com/misc/genetic-defects.html>, August 2011.





## **Title of AppendixA**

