DEGREE PROJECT IN COMMUNICATION SYSTEMS, SECOND LEVEL
STOCKHOLM, SWEDEN 2015

# Radio Based User Presence

ANA R. SANTAMARÍA PORTOLÉS

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
*INFORMATION AND COMMUNICATION TECHNOLOGY*

# Radio Based User Presence

Ana R. Santamaría Portolés

2015-05-21

Master’s Thesis

Examiner and Academic adviser
Gerald Q. Maguire Jr.

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

# Abstract

While radio frequency identification tags (RFIDs) have been widely used for identifying objects (and some people), their primary use has been for access control, inventory, and other similar purposes. In this thesis we will use the detection of an RFID to indicate that a user is present in a given location. This will be used to extend a Session Initiation Protocol (SIP) system with user presence.

A typical HF RFID reader is able to read a tag from a range of 8-10 cm. In this application we wish to read tags from a range of 50 to 100 cm (the width of a doorway or narrow hallway). One of the challenges is how to increase the reading range while staying below the maximum RF power limits (for reasons of safety). Providing an RFID based presence indication can be used to automatically adjust the heating, ventilation, and air conditioning system of a room, etc.

This thesis presents several different antennas which were analyzed and simulated using FEKO to obtain a suitable antenna for this target application. The thesis shows that fractal patterns are the best for the intended application based upon the results of simulations. These antennas were implemented and tested. The prototype establishes that such antennas are a suitable design and can be used with current RFID systems to achieve long ranges. The thesis also suggests some future enhancements to these antennas.

# Sammanfattning

Identifiering av radiofrekvenstaggar (RFID) har använts i stor utsträckning för att identifiera objekt (och vissa människor).

De primära användningsområdena har varit åtkomstkontroll, inventering, och andra liknande ändamål. I denna avhandling kommer vi att använda närvaroupptäckten av en RFID-tagg för att ange att en användare befinner sig påen given plats. Detta kommer att användas för att utöka ett Session Initiation Protocol-system med användarnärvaro.

En typisk HF RFID-läsare kan läsa en tagg från ett avstånd av 8-10 cm. I denna appliceringen vill vi läsa RFID-taggar från ett avstånd av 50 till 100 cm (bredden påen dörröppning eller en smal korridor). En av utmaningarna är hur man kan öka läsavståndet och samtidigt hålla sig under de maximala RF-effektgränserna (av säkerhetsskäl). En RFID-baserad närvaroindikering kan användas för att automatiskt justera värme, ventilation och luftkonditionering av ett rum, osv.

I denna avhandling kommer vi att presentera olika antenner som analyserades och simulerades med hjälp av FEKO, för att erhålla en lämplig antenn. Avhandlingen visar att de bästa resultaten från simuleringarna för det huvudsakliga målet, är att använda fraktala mönster. Genomförandet och testerna av dessa antenner leder till en prototyp av en gynnsam antennutformning, som en bas för framtida förbättringar. Dessutom beskriver vi skälen till varför fraktalantenner ofta används i nuvarande RFID-system för att uppnålånga läsavstånd.

# Acknowledgements

First and foremost I wish to thank my supervisor and examiner at KTH professor Gerald Q. (Chip) Maguire Jr, not only for his inestimate advice during all this thesis project, but also for having had so much patience with me. His persistence and his toleration have been an incredible support throughout all this time and without them this thesis project had not been possible.

I wish to thank also all that little "family" created during my stay in Stockholm. Each of them has partially contributed to shape the person I am today. The spent time with everyone was one of the most inspiring and beneficial source of learning I could have had. The experience of my time in Stockholm will accompany me the rest of my life.

Special thanks to my beloved family, who despite the distance, has always stood by me, reinforcing and encouraging me. They are always unconditionally in every single step I take and everything I do is because of their support. Always and forever, thank you.

In addition, I wish to mention and thank all my professors at my home school of engineering, Universidad Politecnica de Valencia (UPV), whose vocation and talent are, in many cases, remarkable and respectable.

# Contents

# List of Figures

# List of Tables

# List of Acronymns and Abbreviations

| | |
|---|---|
| **CFR** | Code of Federal Regulations |
| **CMOS** | Complementary Metal–Oxide–Semiconductor |
| **EPC** | Engineering, Procurement and Construction |
| **EPC Gen2** | EPCglobal UHF Class 1 Generation 2 |
| **ETSI** | European Telecommunications Standads Institute |
| **FCC** | Federal Communication Commission |
| **HF** | High Frequency |
| **HTTP** | Hypertext Transfer Protocol |
| **IC** | Integrated Circuit |
| **ID** | Identification |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IETF** | Internet Engineering Task Force |
| **IP** | Internet Protocol |
| **ISM** | Industrial, Scientic and Medical radio bands |
| **ISO** | International Organization for Standardization |
| **LAN** | Local Area Network |
| **LF** | Low Frequency |
| **MMUSIC** | Multiparty Multimedia Session Control |
| **MoM** | Method of Moments |
| **OOK** | On-Off Keying |
| **PCB** | Printed Circuit Board |
| **PDA** | Personal Digital Assistant |
| **RF** | Radio Frequency |
| **RFC** | Requests For Comments |
| **RFID** | Radio Frequency Identification |
| **rms** | Root Mean Square |
| **RTP** | Real-time Transport Protocol |
| **SDP** | Session Description Protocol |
| **SER** | SIP Express Router |
| **SIMPLE** | Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions |
| **SIP** | Session Initiation Protocol |

| | |
|---|---|
| **SMS** | Short Message Service |
| **STCP** | Stream Transmission Control Protocol |
| **TCP** | Transmission Control Protocol |
| **TLS** | Transport Layer Security |
| **UAC** | User Agent Client |
| **UAS** | User Agent Server |
| **UDP** | User Datagram Protocol |
| **UHF** | Ultra-high Frequency |
| **ULD** | Unit Load Device |
| **URI** | Uniform Resource Identifier |
| **USA** | United States of America |
| **VoIP** | Voice over IP |
| **WLAN** | Wireless Local Area Network |

# Chapter 1

# Introduction

This chapter describes the main objectives of this thesis project and states the problem addressed. An outline of the thesis is also provided.

## 1.1 Problem Statement

In recent years, a goal of information technology has been removing some of the limitations that have been associated with the use of this technology. The first step was to provide the user with the information that the user needs. The next step focused on making this information, available all of the time and from any location, sometimes referred to as: anything, anytime, and anywhere. Both local and wide area wireless technology broke the constraint of fixed connections, imposed by wired connections.

Advances in wireless technology and integrated circuits fields made connectivity for portable devices possible almost anywhere. Today, personal digital assistants (PDAs) are generally equipped with wireless connectivity, Bluetooth® headphones can be connected to the phone sitting in a cradle in the car, smart appliances can be controlled from the Internet, etc. The Internet, consisting of the converged fixed and wireless packet networks has allowed information producers and information consumers to be anywhere in this logical network. Increasingly high data rates have shifted the main constraint of mobile devices from limited connectivity to limited battery power.

Along with the growth in wide area cellular and wireless local area network communication there has also been the growth of radio frequency identification (RFID). This technology uses signals from a reader to power a low power device (called an RFID tag) to send a signal back to the reader. This signal can range from a single bit to hundreds of bits. In practice either the readers or tags or both can be mobile. In the most frequent case tags are attached to physical objects such as packages, books, ID cards, etc. Today RFID technology is being applied in various areas, ranging from automatically tracking the movement of pallets of goods to replacing the magnetic strip on personal ID cards.

It should be noted that the need to identify objects is not a recent concern. Today the most widely used identification technology is based upon the use of barcodes. Barcode technology has been used to identify products, enabling optical scanners to read the barcode on a product as input to a cash register (for point of sales systems) or an inventory system (in warehouses, shipping terminals, etc.). However, barcodes have a number of limitations, including the limited amount of information that can be stored (in the case of 2D barcodes this can be hundreds of bits) and the difficulty in changing the barcode once it has been printed. Additionally, the barcode needs visibility to operate, i.e., the barcode must be visible to the reader - hence either the object must be positioned so that the barcode is visible to the reader, multiple readers must be used, multiple copies of the barcode applied to different sides of the object, or the object has to be moved so that the barcode faces the reader.

RFID technology overcomes most of the above limitations, since the tags can be read without a direct line of sight view (but the tag must be able to be powered by the reader and the radio signal must be able to get from the tag to the reader), re-programmable tags can be manufactured, the number of bits can be greater than that of a 2D barcode, and the RFID tag can even be integrated into the product itself (making it difficult to remove).

Nevertheless, RFID technology has some limitations. Currently, it is very difficult to read an RFID tag without it being close to the reader, therefore, existing RFID technology requires close proximity to the reader. This can be seen in the case of subway/bus access control in Stockholm (see Figure 1.1), where you have to place the travel-card close to the reader.

Unfortunately, this distance limitation imposes limitations on the use of this technology, hence users have to get out their card and place it nearly on the surface of the reader. This limitation of RFID technology violates our goal of the user not having to be aware of the technology in order to utilize it. Ideally the user should not have to be aware of the technology, i.e., the technology itself should completely transparent to the user, who simply wants to benefit from the technology without being burdened by it. While there exists battery powered RFID tags that can be read at larger distances, these tags have to include a battery which increases their cost and increases the problems of recycling the object in an appropriate way.

The goal of this thesis project is to understand if and how this limitation of RFID technology can be overcome, by proposing a solution in which the distance between the tag and the reader is significantly increased. This will enable the user to avoid having to be aware that the tag is being read, if that disregard is the user's demand.


Figure 1.1: An example of an RFID Tag. This is an Access card issued by Stockholm County's public transport authority (SL)

## 1.2 Objectives

This thesis will focus on detecting the presence and identity of a user using RFID technology, specifically detecting the presence of the user who goes through some access controlled area (for example a doorway, corridor, subway fare gate, etc.). The user's presence will be determined by detecting the presence of items which the user has with him or her, thus we can make use of the ability to detect one or more items that he or she is carrying with them, wearing, etc. More generally, these items may include a cellular phone, a Bluetooth headset, a WLAN equipped PDA or cellular phone, a personal area radio transmitter (for example, a IEEE 802.15.4 sensor), or an RFID tag.

Determining that the user is present and associating this user with some identity, could be useful for many different applications. Some of the possible scenarios include:

- Applications in sports facilities: an access control system can determine who has entered and if this user has permission to use the sport facilities. Once a user is identified the system can indicate what portion of the facilities the user is allowed to use, otherwise permission to use these facilities would be denied.

- Applications in housing developments: In such an application the access control system could monitor the entrance and allowed entry if the user lives there. For example, an RFID tag might be integrated in the key for the apartment, meaning nothing extra needs to be carried by the user. Given an accurate detection of the user, many additional applications can be developed. For example, for a resident the door might automatically unlock without the user consciously acting. While for a non-resident, they might need to ringing the bell of the person residing in the housing development they are visiting, in order for this person to let them in. The access control system could know who is home and who is not. The access control system could play a prerecorded message so that the potential visitor does not know if the user is home or not (perhaps helping to deter thieves).

- Applications in offices: In an office setting, different users might have different permissions to enter specific areas. For example, an employee might have permission to use some rooms or specific equipment, but not others. When the user is in front of a door the access control system should detect their presence and depending upon this specific user's permissions entrance should be granted or not. However, for a user who does have access to the room, they do not have to fumble for a key or position an ID card near a reader, they should simply be able to open the door (or in some cases the door should open for them).

- Applications in conference rooms. In addition to access to the room itself, the access control system could automatically configure the room based upon an earlier reservation by a speaker (see the thesis of Lidan Hu [4]).

The main objective of this thesis is to develop and evaluate an improved means of user presence detection and identification using RFID technology.

## 1.3 Thesis outline

The thesis begins with some background information about RFID systems in Chapter 2, explaining the most important points of the different components. This chapter will present the main features of RFID technology.

In addition, it will introduce some related technologies, and present some of the current solutions for detecting user presence. Chapter 3 will describe the RFID detection methodology that will be used in this thesis project in order to implement a prototype. It will also present the study and the simulations conducted in order to achieve a suitable RFID reader antenna. Testing of this prototype and analyses of these tests will be presented in Chapter 4. Chapter 5 presents come conclusions and suggestions for future work.

# Chapter 2

# Background and Related Work

In this chapter, background information about RFID systems will be introduced. It will explain the basic characteristics of the system and the devices under study. Likewise, related technologies and some current applications and solutions using RFID systems will also be presented.

## 2.1 RFID System

Radio frequency identification (RFID) is thought by many to be one of the most promising new technologies as it combines wireless communication with data communications and potentially tags can even perform some amount of computation or sensing. Today RFID technology is widely used for inventory and distribution. In fact, some large customers (such as Wal-Mart) want each pallet-level shipment from its suppliers to have an RFID [5].

### 2.1.1 What is an RFID system?

RFID is a radiofrequency identification technology. An RFID system consists of an interrogator (reader), a transponder (tag), and at least two antennas. The reader's antenna can either be integrated with the reader or connected via a cable. The tag's antenna is generally integrated with the tag in order to minimize the total cost and to allow the entire tag with antenna to be small in size.

For multi-bit tags, as will be used in this thesis project, the tag is implemented as an integrated circuit. The tag is generally attached to the object that identifies it, although the tag could also be made into the object. The reader provides power to the tag and listens to the signal from the tag, with the complete process referred to as "reading" the tag. The reader may be connected to a network or a host for storing the data and/or displaying the results. The relationship between the reader and the tag is shown in Figure 2.1.

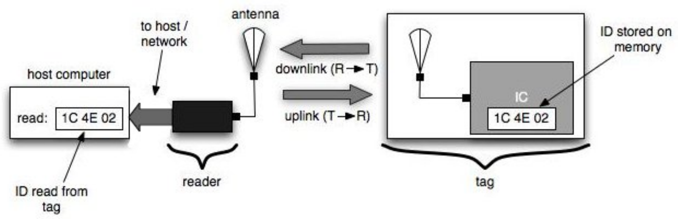



Figure 2.1: Interaction between an RFID tag and reader (Adapted from figure 2.8 in [1])

The integrated circuit can store a unique code that identifies the tag. This can in turn be used in conjunction with a database to identify the object to which the tag is attached. In some cases the tag can also locally compute some function (for example, the temperature of the tag) and send back a value that includes this measurement value. In some cases the tag has a small battery and is able to perform some local processing; for example, recording the temperature over time, so that it can report the highest or lowest temperature that it has experienced. However, tags that have large batteries are outside the scope of this thesis work. For details see [1].

As we can see the process depends upon the reader sending radio frequency waves to the tag, which are received by a small antenna, the local circuit has to extract enough power from this signal to power the microchip. The chip performs the computations that is should and transmits its unique code and possibly other values using the energy supplied by the reader.

Once the reader has received a response, it can consult a database using the unique code as a key, in order to find additional information that is associated with this unique code. For example, in the case of an single item being tagged -the database might contain a record that indicates the date of manufacture, a list of the materials the item is made of, its shipping weight, the dimensions of the package it is shipped in, the product name, the location where it was manufactured, the list of locations where it has been stored, the dates of receipt and shipment from each of these locations, etc. In the case of an RFID equipped personal card this technology can be used to look up the identity associated with this tag, to determine who is supposed to be in possession of this tag. Note that reading a specific tag does not necessarily tell us who is actually in possession of the tag, but it can give some hint (if the identity of the person who posses the tag can be established by some other means and we can ensure that this person has continuously been in control of this tag, then the identification can be stronger [6])

RFID systems use frequencies from around 100 KHz to over 5Ghz. There are three main frequency bands that RFID systems use:

- **Low frequency (LF)**:This band ranges from 125-134 KHz. These frequencies are particularly appropriated for human and animal identification. Transmissions at these frequencies are largely unaffected by the presence of water. These types of tags and readers are commonly used for access control.

- **High frequency (HF)**:This band is at 13.56 Mhz. HF tags are generally used for non-contact smart cards and for tracking/management of objects, such as in libraries for book tracking, pallet tracking, building access control, airline baggage tracking, articles of clothing, etc.

- **Ultra-high frequency (UHF)**:This band ranges from 900 MHz to 2.4 GHz, and formally ends at 3GHz (microwave). The most important advantage of this band is its potential long range. UHF RFID tags are commonly used commercially for pallet and container tracking, monitoring of trucks and trailers being used for shipments, etc.

The main problem for UHF systems is that they can not be used worldwide because there are no global regulations for their use. Additionally, even thin metallic films prevent reading at these frequencies. In North America, UHF can be used in the frequencies between 908 to 928 MHz, but there are restrictions on transmit power. The standard North American UHF (908-928 MHz) equipment is not accepted in France and Italy because it interferes with its military bands. In China and Japan there are no regulations which generally permit the use of UHF. As a result, each application for UHF RFID in these countries needs a license, which must be obtained from the local authorities, and may be revoked. In Australia and New Zealand, the range 918 to 926 MHz is available for unlicensed use, but there are restrictions on transmit power.

An important aspect of RFID tags are their ubiquity. A reader may be in an environment where there are many tags. If a reader tries to work with a set of tags, it must know the devices found within range, then either poll them one by one or make use of collision avoidance protocols. An example of a setting where there are many tags is inventorying a shipping container, when the tags are each associated with sensors collecting information about the environmental conditions as measured by each of the tags.

### 2.1.2 RFID Tag

The RFID tag or transponder consists of an antenna, a transducer radio, and a integrated circuit (a microchip). The purpose of the antenna is to allow the chip, which contains information, to transmit the identification information contained in the tag and to receive sufficient power to be able to power the microchip. The microchip has an internal memory with a capacity that depends on the model and varies from tens to thousands of bytes.

In order for the tag's integrated circuit (IC) to operate, enough power must be delivered by the tag antenna to the microchip, otherwise, the tag can not respond. Thus the first task of the tag's antenna is to couple the received radio wave to the circuitry within the tag.

For communication, the tags responds to requests or queries, generating signals by modulation of the reflection or backscatter, thus using the energy of the reader for its communication. Thus the second task of the antenna is to radiate this modulated signal. It is important to note that the signal from the tag may be very weak and must be differentiated from the signal being emitted by the reader. For details see [1].

RFID tags can be active, semi-passive (also known as semi-active or battery-assisted), or passive. We will discuss each of these types in tags in the following paragraphs.

#### Passive tags

Passive tags require no internal power source and are purely passive devices (i.e., they are only active when a reader provides them with the necessary energy). The signal from reader induces a small electrical current, which needs to be sufficient to operate the CMOS integrated circuit of the tag, so this circuit can generate and transmit a response (see Figure 2.2). Most passive tags use backscatter of the carrier, i.e., the antenna must be designed to provide the energy needed to operate the tag's circuitry and to convey the backscatter response at the same time.

As with other types of tags, a response can be any type of information, not just an identifying code. A tag may include nonvolatile memory that can be written. Today most RFID tags are passive, as they are much cheaper to manufacture.

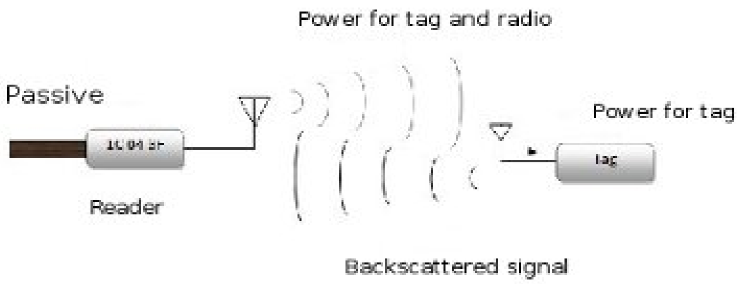



Figure 2.2: Passive Tag Power/Transmit Configuration (Adapted from figure 2.16 in [1])

### Active tags

Unlike passive tags, active tags have their own autonomous source of energy to power the integrated circuits and to generate a signal to the reader (see Figure 2.3). These tags are much more reliable (there are fewer communication errors) than passive tags, as active tags can establish communication sessions with the reader, hence they can support more complex communication protocols with handshaking, retransmission, challenges, etc. Thanks to its power source, an active tag is capable of transmitting a stronger signal than a passive tag, hence active tags are more effective in difficult radiofrequency environments, for example when faced with impairments such as water (including humans and livestock, as both consist mostly of water) or metal (containers, vehicles). Active tags are also able to operate at greater distances from the reader, and they can generate clear responses even from weak signals (in contrast to passive tags). However, they are usually considerably larger and more expensive, and their useful lifetime is generally much shorter. Today, active tags can have effective ranges of hundreds of meters and a battery life span of about 10 years. Commonly used active tags have practical ranges of ten meters and a battery life of several years. The smallest active tags are currently about the size of a coin.

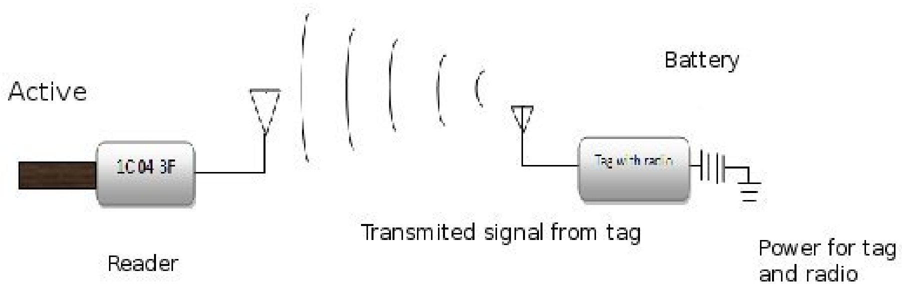



Figure 2.3: Active Tag Power/Transmit Configuration (Adapted from figure 2.16 in [1])

### Semi-Passive tags

Semi-passive tags are like active tags because they have an independent power supply, although in this case this power supply is mainly used to power the microchip and not to transmit a signal (see Figure 2.4). Like a passive tag, the energy contained in the radio-frequency signal from the reader is reflected to the reader. The battery may also be used to store energy received from the reader in order to issue a response in the future, typically using backscatter (remember that tags without batteries can only respond by reflecting carrier energy from the reader). The battery allows the integrated circuit tag to be constantly powered and eliminates the need for the antenna to couple the incoming signal power into a circuit to power the integrated circuit. As a result, the antenna can be optimized for backscattering communication methods. Semi-passive RFID tags can respond more quickly than passive tags. Semi-passive tags have a reliability comparable to active tags, but are limited to the operating range of a passive tag. Because the battery is only used to power the integrated circuit semi-passive tags have a longer battery life span than active tags.

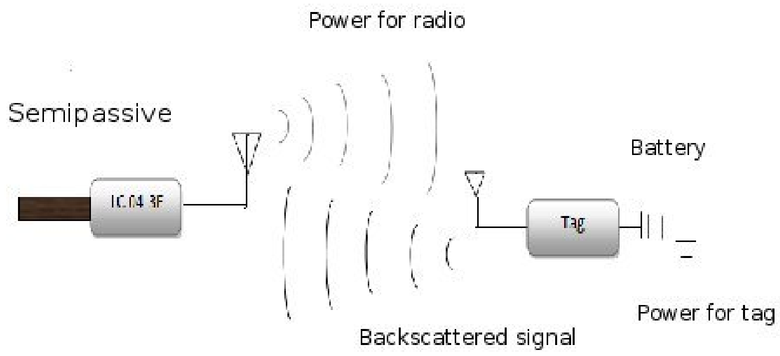



Figure 2.4: Semi-Passive Tag Power/Transmit Configuration (Adapted from figure 2.16 in [1])

**RFID Standards**

There are a wide range of tags and readers. As in other technology areas, standardization in the field of RFID is characterized by the existence of several groups with competing specifications. On one side is ISO and on the other side is the Auto-ID Center (known since October 2003 as EPCglobal, where EPC stands for Electronic Product Code).

In the HF band, ISO has developed standards for RFID for automatic identification and object management. There are several related standards, such as ISO 10536[7], ISO 14443[8], and ISO 15693[9]. The series of standards related strictly to RFID and the frequencies used in these systems start with the number 18000.

### 2.1.3 RFID Reader

The other major element needed to implement an RFID system is the reader or transceiver (a combination of transmitter and receiver). The reader will transmit a radio signal and the receiver will receive the radio signal emitted by the tags. The reader processes the received signal to extract information and sends the decoded data to the computer attached to the reader for further processing. The reader is generally equipped with an antenna, a transceiver, and a decoder.

In its normal mode of operation, a reader periodically sends a signal to see if there is a tag nearby. When a signal from a tag (which contains the identification information of this tag) is received in response, then the information is extracted from this response and passed to a data processing subsystem attached to the reader. The data processing subsystem or RFID middleware provides the common data processing and storage functions required by an application program. Typically the information will be collected and stored in a database.

Readers have two basic frequency-conversion architectures: direct conversion (homodyne) and multiple conversions (heterodyne). The difference between these two architectures is the type of conversion, as suggested by their name. Direct conversion converts the received RF signal to the baseband directly, whereas a heterodyne configuration uses an intermediate frequency in between baseband frequency and the carrier. Direct conversion is relatively simple to implement and is used in many radio systems.

Usually, readers use On-Off Keying (OOK) modulation or variants of OOK to modulate the signals which are sent to the tag. The reader transmits a continuous wave signal, providing the tag with a signal to backscatter (and potentially the rectified version of this signal is used by the tag for power). The backscattered signal is at same frequency as the transmitted signal, so we can use part of the continuous wave signal as a local oscillator to directly convert the backscattered signal from a tag into a baseband data signal. For details see [1].

An RFID transmitter performs two tasks:

**Downlink** Provides power to the tag and modulate its signal to send the tag commands and data

**Uplink** Provides an unmodulated signal that the tag can backscatter in order for the tag to return data to the receiver

Generally the reader is designed to transmit at the maximum allowed output power, so that the reader can achieve the greatest read range. In the case where the reader has to provide all of the power for the tag, this is often the most limiting aspect of a reader + antenna combination. In contrast, correctly receiving weak signals can be accomplished – at greater ranges than the range at which a passive tag can be powered.

### 2.1.4 RFID Antennas

An antenna is a device designed for the purpose of emitting or receiving electromagnetic waves. A transmitting antenna transforms changes in voltages into radiated electromagnetic waves, while a receiver antenna performs the inverse function. In the following paragraphs we will describe the distinctions between the design of the reader antenna and the tag antenna.

### Reader antenna

The reader antenna is generally implemented in one of two forms, each with advantages and disadvantages in relation to its performance. These two forms are: linear and circular polarization. Polarization refers to the way an antenna transmits an electromagnetic wave.

- **Linear Polarization**: "Linear" polarization means that the electric field of an RF signal propagates in a single plane. The benefit of linear polarization is a more focused radiation pattern, which often results in a greater range. The disadvantage is that many RFID tags are also linearly polarized, and both elements must be positioned in the same orientation to achieve optimal coupling.

- **Circular polarization**: Circular polarization means that the field component E of an RF signal turns (in the sense of clockwise or counterclockwise) while propagating away from the antenna. An advantage of this approach is that the orientation of the RFID tag is no longer important. The disadvantage is that the intensity of the circular polarization signal is directed at a larger area than the linear polarization, which limits the range.

Besides these two types of polarization, important antenna features are gain, directivity, and impedance.

**Gain** Gain is the ratio of radiated power density in the direction of maximum radiation at a distance r and the total power delivered to the antenna, divided by the area of a sphere of radius r. The efficiency of an antenna is the relationship between gain and directivity. This relationship coincides with the ratio of total radiated power and the power delivered to the antenna.

**Directivity** Directivity is the relation between power density radiated in the direction of maximum radiation at a distance r and the total radiated power divided by the area of the sphere of radius r. The directivity can be calculated from the radiation pattern. The gain of an antenna is equal to the directivity multiplied by the efficiency.

**Impedance** Impedance refers to the impedance of the antenna at its terminals. Impedance is the relation between voltage and input current, Z = V/I. The impedance is a complex number. The real part of impedance is called antenna resistance and the imaginary part is reactance. The antenna resistance is the sum of radiation resistance and loss resistance. An antenna is resonant when the entry reactance is canceled.

Although, at first sight, some obvious solutions to increase the gain, such as focusing the pattern of radiation with a narrow beam might be deployed in practice; this means that the volume in which a tag must be placed is quite small which prevents using this approach in many systems. An alternative is to use an array of antennas, each of which has a limited working volume, but the sum of which covers the volume of interest. Understanding what antenna should be used or if multiple antennas are used is an important part of this thesis project.

**Tag antenna**

The type of antenna used in a tag depends on the application for which the tag is designed and what its operating frequency will be. Low frequency (LF) tags make use of electromagnetic induction. As the induced voltage is proportional to its frequency, such an antenna can produce enough voltage to directly feed an integrated circuit using the proper number of turns (i.e., it is acting as a transformer – with the reader's antenna providing one side of the transformer and the tag antenna providing the other). There are compact LF tags (some are encapsulated in glass for human and animal identification) using an antenna at different positions (for example three sets of 100-150 turns each) around a ferrite core.

At high frequency (i.e., HF of 13.56 MHz), a flat spiral with 5-7 turns is used. This facilitates the tag antenna being used in form factor similar to a credit card. These systems achieve read distances of tens of centimeters. These antennas are cheaper to construct than an LF antenna, as they be may printed using lithography (in some cases directly printed on a substrate with conductive inks).

Passive tags at UHF frequencies and microwave frequencies frequently use a classic dipole antenna. The antenna can be constructed using a simple metal layer, reducing the cost of the antenna. However, dipole antennas do not work well with a typical IC (due to the IC's high input impedance and low capacitance).

As dipoles are coupled by radiation when their axes are orthogonal, hence the visibility of a tag using a simple dipole antenna depends on its orientation. As a result, tags with two orthogonal antennas (dual dipole tags) will be less dependent upon the orientation of the reader's antenna, but they are usually larger and more expensive than their single dipole counterparts.

UHF patch antennas provide service in the proximity of metal surfaces. However, they need a 3 to 6 mm metal thickness to achieve good bandwidth. Additionally, it is necessary to have a ground connection, which increases the cost compared with single laye structures.

HF and UHF antennas are usually made of copper or aluminum. Conductive inks have been tested for use with some antennas designs. Additionally, there have been problems with adherence to the integrated circuit which leads to intermittent connectivity, and the stability of the environment which leads to insecurity by the customer. For details see [1].

## 2.2 User Presence Detection

The goal of this thesis project is to provide user presence detection and identification using RFID. Since most of the low cost RFID tags that a user is likely to have with them are passive RFID tags, we need to first understand how we can provide enough power from the reader to the tag in order for the tag to continue to operate. Next we have to query the tag. Finally, based upon the response from the tag we want to send a Session Initiation Protocol (SIP) Notify message to a SIP proxy indicating that a given tag has been seen at a particular reader (which we will assume has a know location). In this thesis project we will use passive tags designed to operate at 13.56 MHz. The following sections provide an overview of SIP as a protocol, the elements of a SIP infrastructure, the format of SIP messages, a specific SIP protocol for presence, some additional details of a popular SIP server, and a very brief overview of the type of database that will be used to store information (associating tags with users).

### 2.2.1 SIP

The Session Initiation Protocol (SIP) was developed by Internet Engineering Task Force (IETF's) Multiparty Multimedia Session Control (MMUSIC) Working Group. It is intended to be the standard for the initiation, modification, and termination of sessions involving multimedia elements such as video, voice, instant messaging, real-time text, on-line gaming, and virtual reality. SIP is designed to be very flexible, extensible, and open. SIP enables the establishment of multimedia sessions between two or more users. SIP manages the exchange of messages between parties that want to communicate, enabling these parties to know where to send their media streams and what format the media should be in facilitates the receiver can make use of it.

SIP was designed by the IETF to re-use functions provided by other protocols, thus avoiding the need to introduce new protocols. Hence SIP focuses on the establishment, modification, and termination of sessions; while leaving the description of what the session is to the Session Description Protocol (SDP). SDP describes the multimedia content of the session, including what IP addresses, ports, and CODECs will be used during the session. The actual media is transferred using a profile of the Real-time Transport Protocol (RTP).

The syntax of SIP is similar to HTTP, a protocol used in Web services, and uses addresses similar to those used for the distribution of e-mails. This similarity is natural, since SIP was designed to enable voice over IP (VoIP) as another communication service via the Internet and this furthered the goal of building upon the experiences with these earlier protocols.

Another important concept in the design of SIP was extensibility. This means that the basic functions of the protocol (defined in the Request for Comment (RFC) 3261 [10]) can be extended by others. Purposely designing the protocol to be easily extensible has allowed SIP to evolve.

The basic functions of SIP include:

- Determining the user's location. This enables user, device, and session mobility.
- Establishing, modifying, and terminating multimedia sessions between users.

SIP adopts a client-server model and is based on transactions. The SIP client makes requests of a server. The SIP server generates one or more responses (depending on the request). Initially a SIP user will register with a SIP registrar so that they can subsequently be located to participate in a session. To initiate a session, the client sends an INVITE request, indicating to which user (or application) the client wishes to establish a session. The server responds either by rejecting or accepting this request. Each response has a status code that provides information about whether the request was resolved successfully or failed. The initial request and all the subsequent replies constitute a transaction.

SIP servers listen, by default, on port 5060 to receive requests from SIP clients. This same port number is used for traffic via several different transport protocols, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Transmission Control Protocol (STCP). For Transport Layer Security (TLS) over TCP, the TCP port number 5061 is generally used.

We have chosen to use SIP as this protocol is widely used, quite simple, and using this protocol enables us to build upon a number of earlier thesis projects at KTH. More specifically, the use of SIP Notify messages will enable this user presence detection system to integrate with a number of other projects that also are concerned with user presence [3].

### 2.2.2 SIP Messages

SIP is a textual protocol that uses similar semantics to the HTTP protocol. The UAC makes requests and the UAS returns responses to requests from clients. SIP defines two types of messages: methods and responses (with status codes). These messages use the generic message format established in RFC 2822 [11], which consists of an initial line followed by one or more header fields (headers), an empty line indicating the end of the headers, and finally, the message body, which is optional.

SIP requests are characterized by an initial line in the message, called the Request-Line, which contains the method name, the identifier of the request receiver (the Request-URI), and the SIP protocol version. There are six SIP basic methods (as defined in RFC 2543) that describe the client requests:

**INVITE** Initiates a session or modifies parameters of an existing session

**ACK** Confirms the establishment of a session

**OPTION** Request information on the capabilities of a server

**BYE** Indicates the end of a session

**CANCEL** Cancels a pending request

**REGISTER** Register the User Agent with the registrar

In Figure 2.5 we can see an example of a SIP message and the transactions between the users.

After receiving and interpreting of a SIP request message, the receiver responds with a message. This message is similar to the aforementioned one, differing in the starting line, here called the Status-Line, which contains the version of SIP, the response code (Status-Code), and a short description (a Reason-Phrase). The response code consists of three digits that can be classified into a number of different types of responses. The first digit defines the type of response. These reason codes are shown in Table 2.1.

```
             server1.com  . . . .  server2.com
          .    proxy               proxy     .
       .                                       .
User1's  . . . . . . . . . . . . . . . . . . . .  User2's
softphone                                        SIP Phone
   |                |                 |                |
   |    INVITE M1   |                 |                |
   |--------------->|    INVITE M2    |                |
   |  100 Trying M3 |---------------->|    INVITE M4   |
   |<---------------|  100 Trying M5  |--------------->|
   |                |<--------------- | 180 Ringing M6 |
   |                | 180 Ringing M7  |<---------------|
   | 180 Ringing M8 |<----------------|     200 OK M9  |
   |<---------------|    200 OK M10   |<---------------|
   |    200 OK M11  |<----------------|                |
   |<---------------|                 |                |
   |                     ACK M12                       |
   |-------------------------------------------------->|
   |                  Media Session                    |
   |<=================================================>|
   |                     BYE M13                       |
   |<--------------------------------------------------|
   |                   200 OK M14                      |
   |-------------------------------------------------->|
   |                                                   |
```

Figure 2.5: Example of SIP message (Figure 1 in [2])

Table 2.1: Status Codes

| Class Code | Purpose |
|---|---|
| 1xx | Provisional messages |
| 2xx | Successful responses |
| 3xx | Redirection Responses |
| 4xx | Failure response method |
| 5xx | Server Failure Responses |
| 6xx | Global Failure Responses |

### 2.2.3 Operation of the protocol

#### User Agent

Users, who can be humans or software applications, use what the SIP protocol calls "user agents" to establish sessions. These user agents are the endpoints of the session that will potentially be established, modified, or terminated. A hardware SIP phone (with audio and possibly video), a software client (referred to as a softphone), or any other similar device implementing a SIP user agent can be used for a SIP session. The SIP protocol does not address the interface between these devices and the end user, as SIP is only concerned with the messages these user agents generate and consume.

User agents can behave as clients (called User Agent Clients - UACs), as servers (called User Agent Servers – UAS), or both. They act as a UAC when making a request and as a UAS when receiving a request. Therefore most user agents must implement both functions, however there are devices that might only be used to initiate session and others that might only be invited to sessions by others.

#### Registrar Server

SIP enables the network location of a particular user agent to be dynamically determined. This is done by having each user agent that wishes to be reachable register its current network address. This mechanism works as follows: each user has a logical address which is independent of the user's network location. A logical SIP address has the form: 'user@domain', i.e., a SIP address is similar to an email address; but this address is in the SIP domain rather than an e-mail domain. The user agent's network address (called a "contact address") depends on where the user agent is currently connected (i.e., the user agent's IP address). When a user initializes a SIP user agent (for example by connecting his or her SIP phone via a WLAN or LAN link), the SIP user agent sends a SIP REGISTER request to the registrar Server, informing the registrar server what network address should be associated with the user's SIP address. The Registrar Server performs this association (called binding) by storing this association (a mapping) for some period of time. If this registration is not renewed before it expires, then the registrar will remove this binding. This binding may also be terminated by the user agent explicitly requesting a deregistration. How the association is stored by the registrar is not specified by the SIP protocol. However, this binding is needed by various SIP network elements in order to perform their tasks. We will examine these tasks later.

The various SIP entities in the SIP architecture are shown in Figure 2.6. Xueliang Ren describes the components of this architecture in his thesis [3] as:

- User Agent Client: The UAC is a logical SIP entity that makes a SIP request.
- User Agent Server: The UAS is a logical SIP entity which generates a response to a SIP request.
- SIP Proxy: A SIP proxy is an intermediary that forwards a request from UAC to a SIP proxy or to a UAS.

- Redirect Server: A redirect server redirects a request from a UAC or SIP proxy to another SIP entity that may be able to service this request.
- Registrar Server: A registrar server receives SIP registration requests and updates the UA's network location as stored in a location server or another database.

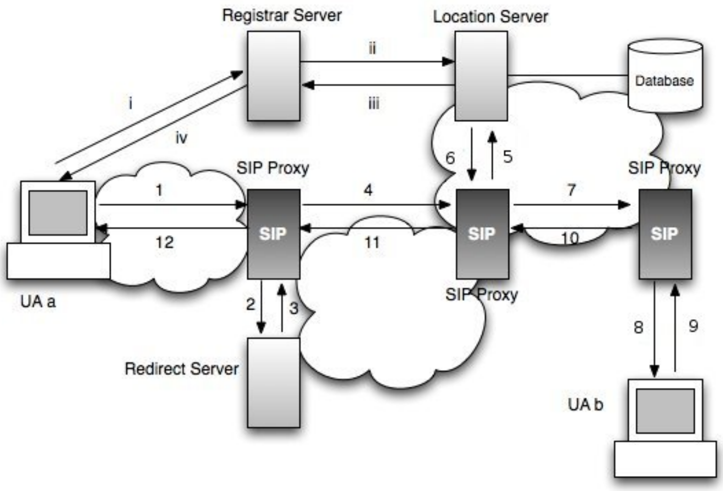



Figure 2.6: SIP Architecture and Operations (Adapted from figure 2.7 in [3])

### 2.2.4 SIMPLE

The Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) is an extension of SIP to distribute instant messaging and presence information. SIMPLE enables messages to be exchanged within a SIP session. Similar to Jabber, but different than most instant messaging protocols, SIMPLE is an open standard.

SIMPLE applies SIP to the following problems:

- Registering presence information and receiving notifications when events occur, for example when a user logs on or logs out.
- Supervise real-time messages between two or more participants.

SIMPLE was developed by the IETF SIMPLE working group. Some parts have been standardized, for example RFC 3428 [12]. Other aspects of SIMPLE, in particular instant messaging sessions, are contained in other

RFCs, such as RFC 6914 [13]. However, many implementations of SIMPLE are available, for example the one used by Microsoft Lync.

### 2.2.5 SIP EXPRESS ROUTER

The SIP Express Router (SER) is a SIP server which can act as a SIP proxy, a SIP registrar, or a redirect server. It has a very high performance and can easily be configured and modified because SER is designed as open source software. SER offers services such as an SMS gateway, SIMPLE2Jabber gateway, account management and access authorization via RADIUS, server status monitoring, security, etc. SER can also be configured via web using databases. SER's performance allows it to manage events effectively such as network element outages, attacks, restarts and rapid growth of users. SER can handle IPv4 and IPv6 simultaneously in a transparent manner by providing connectivity between both. Controlling and management of SER is easier when using the MySQL modules, which allow the storage of users in a database for management.

Note that there are several implementations of SER. However, there have been some attempts to unify some of the divergent version leading to systems such as Kamailio® (a successor of OpenSER and SER) [14].

## 2.3 Related Research

RFID systems are being implemented in many applications. Additionally, there is a very large amount of research being done to decrease the cost of RFID equipment (especially tags), to better utilize the capabilities of RFID in applications, and to extend the communication between devices that are proximal by using near field communication. Here we will mention two different examples of traditional uses of RFID. Following this we will describe one of several attempts to listen to the communication between a reader and a tag. The specific example we will consider is one that attempts to read a tag at long range. Finally we will give an example of one system that has been designed for long range reading.

### 2.3.1 A low-cost extended range Skimmer

In [15], Ilan Kirschenbaum and Avishai Wool, describe how to built a low cost, extended range skimmer. An RFID skimmer can be loosely defined as a device used to gather information from RFID systems. In their paper, they show how simple it has become to "skim" the RFIDs in your office or car key.

An RFID skimmer can be used to gather legitimate information, such as for inventory control in stores, supermarkets, or pharmacies. However, an

RFID skimmer can also be used to collect illegitimate information causing a security problem. The device can be used by identity thieves as part of a relay attack system. In this type of attack, the attackers use the information of the victim to make purchases, start a car, etc.

They claim a final read range of 25cm with ISO-14443 tags using a 40 cm-diameter copper-tube antenna powered by 12 V battery, built using a budget of US$ 100. They describe a relay-attack at a range of 5 to 10 cm, hence in this attack there is no need for the tag to touch the reader. The architecture consists of a “leech” and a “ghost”, which communicate with each other. Their application skims the card being attacked and passes this data to another device (the “ghost”) which subsequently passes this data to the reader as if it were the attacked card.

### 2.3.2 Dag-System Solutions

Since 1992, the company PYGMALYON has been committed to developing an “electronic tag” (the DAG), which allows unrestricted identification of any “element” (package, luggage, pallet, files, raw materials, finished products, local) in a control process. DAG SYSTEM [16] technology was originally developed for sports applications (such as determining the time when competitors cross a finish line). They have utilized RFID tags operating at 13.56 MHz.

Their present system is capable of being used for access control in buildings and industrial traceability of products. Some of the unique features of this system are its very long read distance (up to ten meters away) and rapid unrestricted installation. To achieve these long reading ranges they utilize antennas ranging from 2.3m to 10m wide and with one or more antenna elements this size.

They offer their products for different applications. Especially, for sports, for example cycling races. The large reader antennas mean that the user does not need to show their RF card in order to be timed at a finish line (or way point). Also, they support a number of different RFID chipsets.

Another advantage of their technology is the advanced anti-collision algorithm which they use. In this way, multiple simultaneous readings of tags are possible, as the anti-collision system allows the detection of many tags at the same time.

### 2.3.3 Access Control Toll Road

S.I.C. TransCore [17] develops identification systems for use with vehicles. They implement RFID systems for the detection of vehicles on toll roads. The RF module generates a continuous signal in the 900 MHz- 2.45 GHz band, which is transmitted by the antenna. When a vehicle enters the reading area of the antenna, the tag is activated and reflects the signal to the antenna, returning the identification code stored in the tag.

The reader's antenna receives this signal and transfers it to an RF module which demodulates and preamplifies it before transmitting the signal to the reader. The reader contains a microprocessor that decodes the signal and it adds additional data, such as the time and date when the tag was read.

The module utilizes horizontal polarization, with an adjustable range from 0.9 to 9 meters and a maximum vehicle speed of 150 km / h when using tags with a battery. When using passive tags the read range is approximately 3.5 meters by increasing the output power up to 2 W. Due to the choice of frequency band, the read range varies depending on the weather.

In this system, each vehicle has a label (passive tag) attached. This avoids the need for a battery and with it any need for battery maintenance. The label contains a unique identification (ID) code. This ID is associated with a vehicle by an entry in a database. Using this system a toll authority can check if the owner has paid for access to this toll road, and if not, the owner can be invoiced for use of the toll road.

### 2.3.4 Mobile Phones: Tri-Mex International, collaboration Nokia and DHL

TRI-MEX International worked with DHL and Nokia in a study, to develop a system to track items throughout their distribution network. RFID allowed improved security and supply chain efficiency in their study of the mobile phone supply chain.

The project had two stages. The first stage, consisted of tracking cases of mobile phones tagged with active tags while in transit. These tags were placed in the mobile phone transport packaging containing several consumer mobile phone packages. The second stage involved the use of passive (ISO 15693) tags. In this second stage, at the point of manufacture an RFID tag encoded with a unique international mobile equipment identity (IMEI) was inserted into each phone. Therefore in this study RFID tags were initially incorporated into pallets and cases, and subsequently mobile phones themselves. The tags were linked with TRI-MEX satellite tracking systems via the company's international control centre and they could alert customers and law enforcement agencies when distribution was disrupted.

Fixed and mobile readers demonstrated how they could track the mobile phones (or packages) and obtain information about when each unit was processed at each location. The system operated at 13.56 MHz. By optimizing the RFID tag and antenna equipment it was possible to support read ranges of up to 2.5m. This project also showed that it was possible to read multiple tags at high speed and that it was possible to determine the contents of a plastic and metal composite Unit Load Device (ULD) aircraft container (the standardized containers used for air cargo) without opening the container.

# Chapter 3

# Goals and Implementation

In this chapter, we will explain the goals of this thesis project and the methodology selected for our implementation. Simulations, system design, and implementation are also presented in this chapter.

## 3.1 Reader Antenna Research

One of the challenges of this master thesis project is to increase the reading range of an RFID reader such that it can read the contents of an RFID tag at a distance from the reader of between about 0.5 and 1 meters. The reader will work at 13.56 MHz. In contrast, and a typical high frequency (HF) RFID reader can only be read at distance of several centimeters from the tag reader's antenna. The primary limitation is not due to difficulties in receiving the signal emitted by the tag, but rather the problem is to deliver sufficient power to the tag to energize it to emit a signal. For this reason, the first goal of the tag antenna must be to deliver power to the tag IC to turn it on: if that does not happen, nothing else matters. This is the main problem, if we do not have a suitable antenna for the reader which delivers sufficient power to the tag, we will not be able to turn the tag on and consequently we will not be able to read the tag.

The idea is to place the antennas of the reader in both sides of a doorway (as shown in Figure 3.1), thus allowing us to get sufficient power delivered to a tag being carried by a person entering or leaving the room to turn the tag IC on.

Therefore, the initial focus of this project was to design a suitable antenna for the RFID reader. Today, there are several protocols for using HF with passive tags. In this document we will use ISO 15693. This protocol indicates the constrains on the field strengths that are permissible. From ISO 15693-2 [9], we know:

- The minimum operating field is $H_{min}$ and has a value of 150 mA/m root mean square (rms).
- The maximum operating field is $H_{max}$ and has a value of 5 A/m rms.
- A vicinity coupling device shall generate a field of at least $H_{min}$ and not exceeding $H_{max}$.

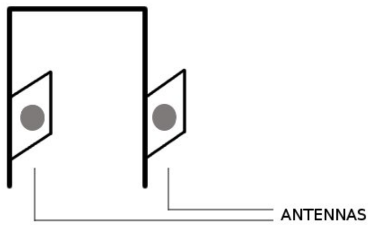



Figure 3.1: Placing antennas at a doorway

For this study, we will use the program FEKO [18] for simulation of electromagnetic fields. Using FEKO we will be able to determine the best option from among various potential antenna designs, and discard designs which are unsuitable for our desired reading range.

First I will introduce details of the problem specification, then introduce some of the potential antenna designs that could be used. These designs will be evaluated in terms of a set of criteria to determine which is the most suitable design for our problem. In addition, we will give an introduction to fractal antennas, as in our analysis they seem to be a suitable means to achieve our goal of reading RFID tags at a distance of 0.5 to 1 m.

### 3.1.1 Calculation of minimum necessary gain

First of all, we need to know the antenna gain which our reader must have. We know our reader must be able to read a tag within the range of 0.5 m up to 1m. The first step is to use the Friis equation. The Friis equation is a very convenient way to state the gain. Usually, the Friis equation (equation 3.1) is used to express the expected received power. In our case, we will calculate the reader gain and we will use typical values for the other data.

$$P_{RX,reader} = P_{TX,reader} T_b G_{reader}^2 G_{tag}^2 (\frac{\lambda}{4\pi r})^4 \tag{3.1}$$

We have the following data:

$\lambda = \frac{c}{f} = 22.12$ m
$P_{TX,reader} = 4W$
$G_{tag} = 2dB_i$
$P_{RX,reader} = -80dBm$

All these data are based upon the typical values as currently used in Europe. The emitted power is regulated by ETSI in Europe, EN 300 330, [19] and FCC in US, FCC CFR47 Part 15. Most tag antennas take the form of dipoles, so their gain is easy to estimate as about 1-2 $dB_i$. The RFID system, that we are going to consider, will use an ISO 15693 tag. There are several RFID air interface protocols for HF with passive tags: MIFARE, ISO 14443, ISO 18000-3, ISO 15693, etc. ISO 15693 systems offer a maximum read distance for 1-1.5 metres as compared to ISO 14443 that only offers a few maximum read distance of a centimeters. ISO 15693 tags usually have a $G_{tag}$ of $2dB_i$, so we will use this value for our calculations.

Using this information we can obtain:

- For 1 metre, the needed Gain is **17.4 dB**
- For 0.5 metre, the needed Gain is **11.4 dB**

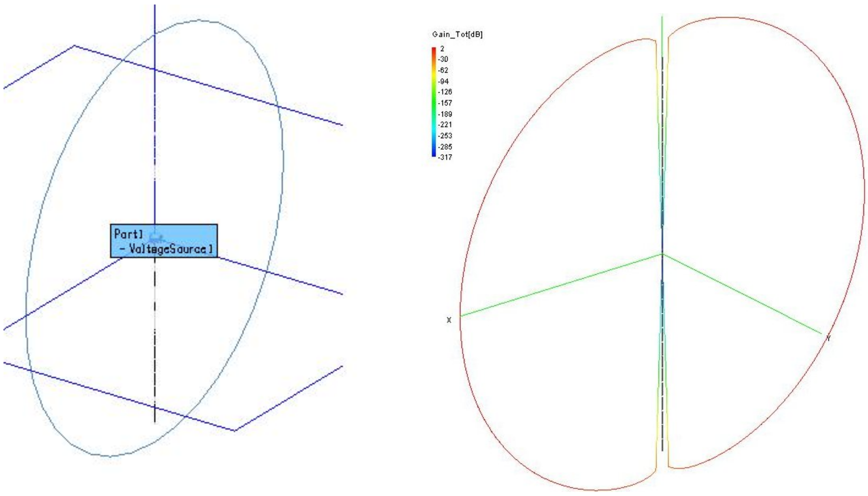
The above values gives a specification for the desired gain that would be suitable for reading a tag at the desired range. The next subsection will describe some of the antenna designs that we have considered.

### 3.1.2 Simulations of Potential Antennas

In this section we will present results of simulations for several of the antennas that have been studied. As noted earlier, the progam used for these simulations is FEKO. The FEKO software is based on the Method of Moments (MoM) integral formulation of Maxwell's equations. All the pictures that shown below were generated by the FEKO simulations. Each of the next paragraphs will describe and evaluate one of the types of antennas that have been considered. The figures show the radiation pattern of the gain and describe how well the antenna converts input power into radio waves headed in a specified direction in order to read the tag that located within the area indicated by the radiation pattern. The scale of the radiation pattern goes from blue color to red color, indicating the red color the maximum gain in the stated direction.

#### 3.1.2.1 Dipole Antenna

A dipole is a center-fed antenna used to transmit or receive radio waves. These antennas are the simplest from a theoretical point of view [20]. We knew that a dipole antenna was not the most suitable antenna for our aim, but we simulated it to provide a basis for comparing the other antennas designs to, and to build up our confidence in the simulation (since the radiation pattern for a dipole is well-known). Figure 3.2 shows the radiation pattern for a dipole antenna. The analyzed dipole has a $\lambda/16$ length. The frequency used in all of our studies is 13.56 MHz, so we could not use a longer dipole. The result of the simulation is shown in Figure 3.2, and as we expected, its gain is 2dB.




(a) Dipole Structure (b) Radiation pattern of the gain

Figure 3.2: Dipole Antenna

#### 3.1.2.2 Yagi-Uda Antenna

A Yagi antenna is a directional antenna invented by Dr. Hidetsugu Yagi of Tohoku Imperial University and his assistant, Dr. Shintaro Uda (hence the formal name of this type of antenna is Yagi-Uda). A Yagi antenna is based upon a dipole antenna with some added elements called "parasitic elements" to make it directional, usually a reflector and one or more directors. Director elements are placed in front of the antenna and reinforce the signal in the direction of emission. The reflector elements are placed behind and block the signals in the direction opposite to the desired direction of transmission. For details see [21]. The basic construction of a Yagi antenna is:

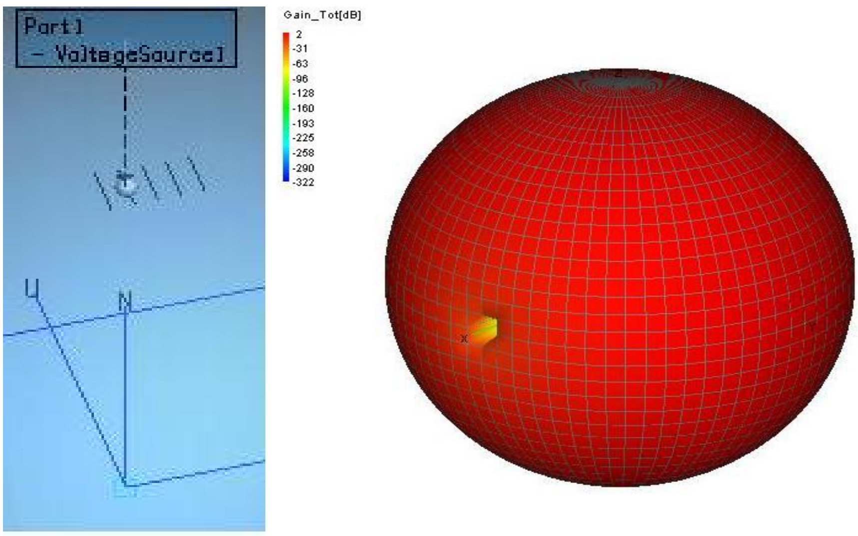
- The length of the driven element is $\lambda/2$, i.e., half the wavelength.
- The reflector element is slightly larger, for example $0.55\lambda$ (i.e., 5% more than $\lambda/2$)

- Finally each director element is 5% shorter than the active element (i.e., 0.45$\lambda$).

We can not directly create a suitable Yagi-Uda antenna for our application since the height of a door way is only approximately 2 m (our $\lambda$ is 22.12 m), so all these measures must be scaled. We analyzed a Yagi based on a active element of size $\lambda/48$, hence our parameters will be:

- The length of the directors is 0.442*$\lambda/48$
- The length of the active element is 0.451*$\lambda/48$
- The length of the reflector is 0.477*$\lambda/48$
- The spacing between elements is 0.5*$\lambda/48$

These data were obtained from a FEKO example that appears in the FEKO Example Guide [22]. After simulation with FEKO, we can see this Yagi-Uda antenna has only 2 dB of gain (as shown in Figure 3.3).

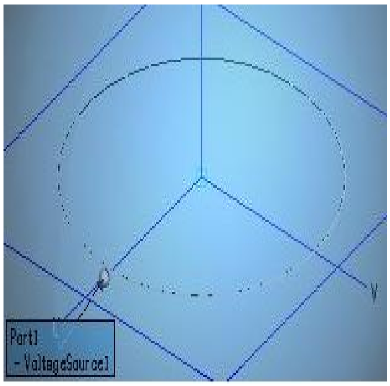



(a) Yagi-Uda Structure (b) Radiation pattern of the gain

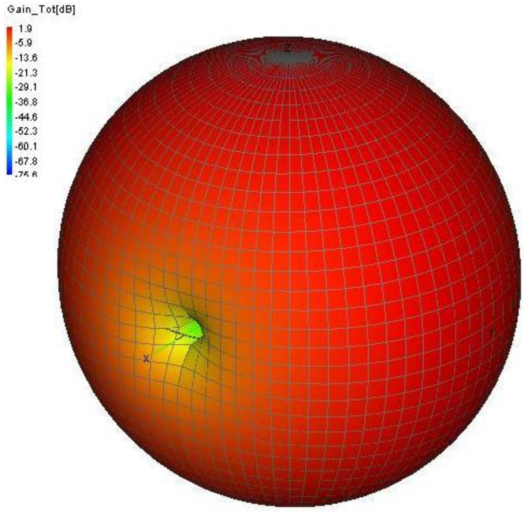
Figure 3.3: Yagi-Uda Antenna

#### 3.1.2.3 Loop Antenna

A loop antenna is an antenna composed of, at least, one loop of a conductor. This can be considered as a dipole whose arms are folded to form a circular loop. All loop antennas are directional and the privileged direction is in the plane of the loop, while the reception is minimal in the perpendicular




(a) Loop Structure




(b) Radiation pattern of the gain

Figure 3.4: Loop Antenna

direction to the plane of the loop.

Loop antennas are the most widely used antennas for HF RFID systems. This was the reason why we decided to simulate such an antenna.

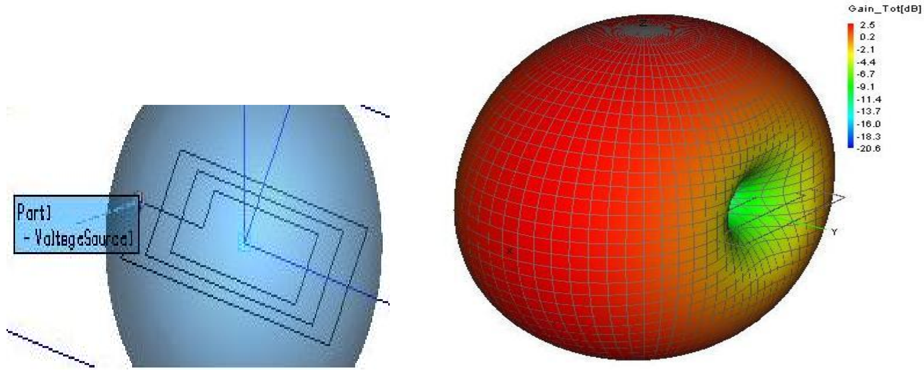
The simulated loop is a 50 cm loop. This size of loop could easily be placed at the door of a room. However, simulation shows that this sort of antenna is not useful as its gain is only 1.9 dB (Figure 3.4).

#### 3.1.2.4 Spiral Antenna

At this point of our investigation, we decided to try other antennas. One of them was an spiral. The selected spiral antenna has 1 meter of height and 50 cm of width; and decreases 20 cm of height and 10 cm of width for each turn, for a total of 3 turns. Using this antenna it is possible to increase the gain. According to our simulation we obtain 2.5 dB of gain (Figure 3.5). As this is far too small, we need to consider more complex antennas.

#### 3.1.2.5 Array Loop

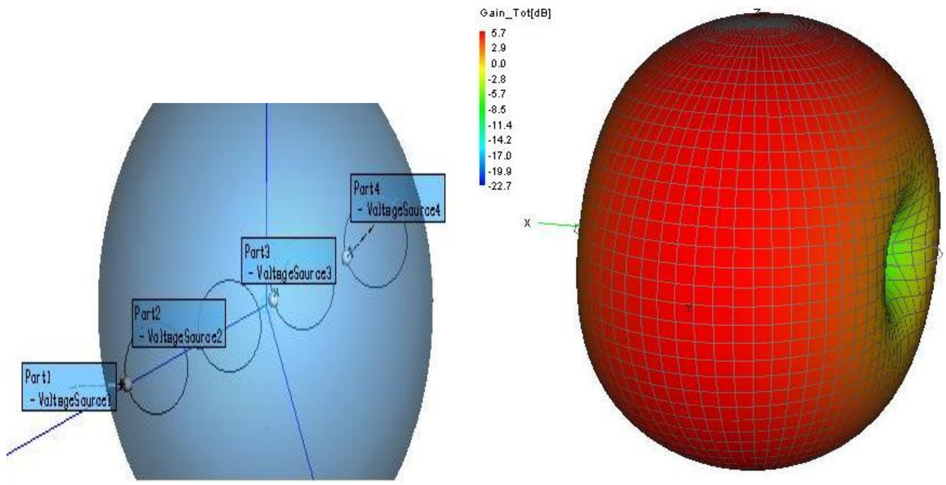
Since the simple single antennas that we have considered did not have sufficient gain, we consider an antenna based upon an array of loops. This antenna consists of four loops each 10 cm in radius and separated from each other by 5 cm. Each loop is fed independently from the others. This




(a) Spiral Structure (b) Radiation pattern of the gain

Figure 3.5: Spiral Antenna

type of antenna was inspired by the antenna installed at the Stockholms Universitetet Library. This antenna offers a gain up to 5.7 dB (Figure 3.6). This is a good result, but it is still far from our goal.



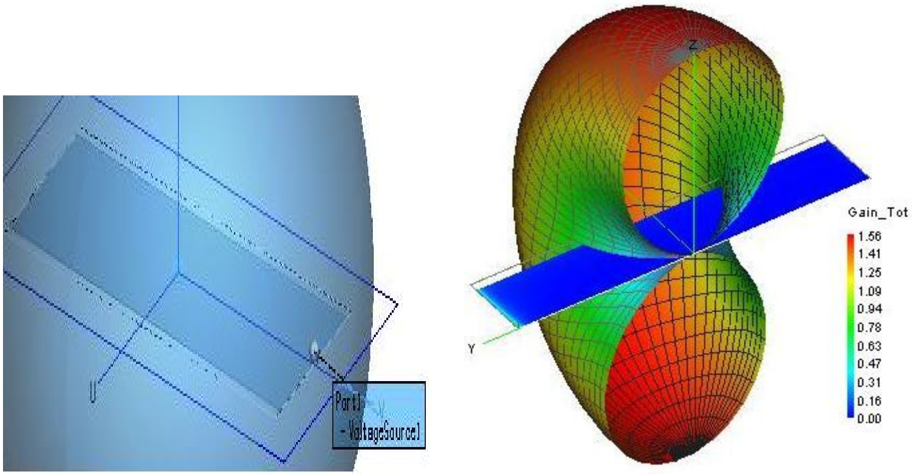

(a) Loops Structure (b) Radiation pattern of the gain

Figure 3.6: Array Loop

#### 3.1.2.6 Loop around the doorway

The next simulation shows a rectangular loop antenna around the full doorway. A typical doorway measures about 2m x 80cm, so these dimensions have been used for the simulation. In this case, we try to see if by making a big antenna we could get sufficient gain to turn the tag on. A large antenna may or may not have high gain, but it was clear that a small antenna can not. Because gain is proportional to size, we will evaluate this antenna.

After simulation with FEKO, we see this rectangular loop antenna has only 1.56 dB of gain (Figure 3.7), and as we see in the radiation pattern, the maximum gain of the antenna is in the center of the rectangle, having no gain at the ends of the rectangle.



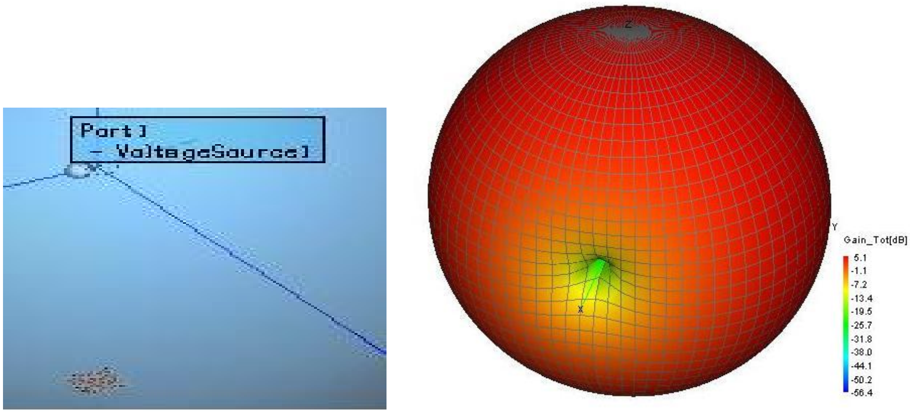

(a) Loop Door Structure (b) Radiation pattern of the gain

Figure 3.7: Loop Door

#### 3.1.2.7 Simulations with a reflector

Based upon antenna theory, we know that if we put a reflector behind an antenna we can improve the gain of the antenna. This approach works best if the spacing between the antenna and the reflector is correctly chosen. The best results occurs when this spacing is $\lambda/4$, but due to our desired operating frequency this will be a large distance (5.53 m), hence this is completely unfeasible. So, we will have to decrease this spacing to a more appropriate value. Inspired by the previous simulations, we added a reflector and repeated the simulation to see what the effect of the distance between the antenna and reflector is. In the simulation, the reflector will be a simple sheet of perfect conductor separated from the antenna. Combining this with the loop simulation with a reflector at $\lambda/4$, we obtain a gain of 5.1 dB.

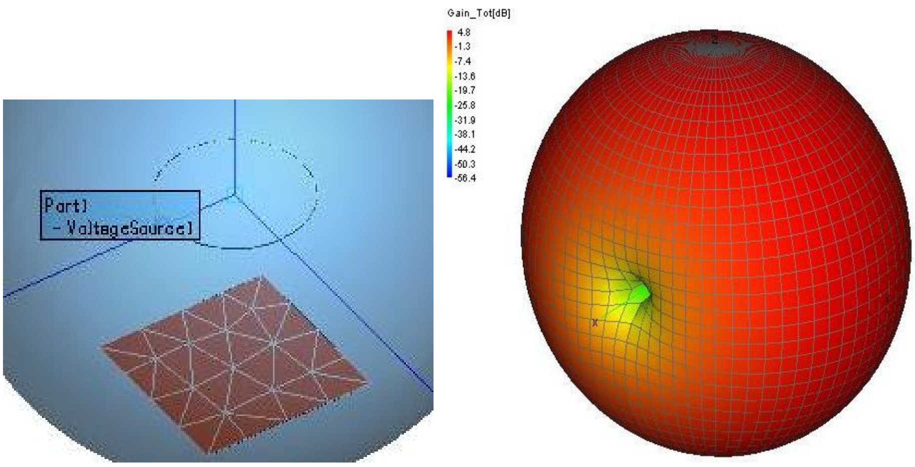
This combination lead to an improvement in gain from 1.9 dB to 5.1 dB




(a) Loop Structure with a reflector (b) Radiation pattern of the gain

Figure 3.8: Loop with a reflector at $\lambda/4$

(see Figure 3.8). If we put the reflector at $\lambda/32$ (0.69 m) we still increase the gain, but in this case the gain is 4.8 dB (see Figure 3.9), slightly less than the optimal gain. However, this new spacing is much more feasible because the antenna is separated from the reflector by a little more than half a meter.




(a) Loop Structure with a reflector (b) Radiation pattern of the gain

Figure 3.9: Loop with a reflector at $\lambda/32$

### 3.1.3 Fractal Antennas

A fractal antenna is an antenna that uses a fractal pattern to maximize the length or increase the perimeter. Fractal concepts have been applied to many branches of science and engineering, and have been extended to antenna theory and design. There have been many studies and implementations of different fractal antenna elements and arrays. The theory about fractal antennas is very extensive, thus in this document we will only touch upon the most important aspects that are relevant to our goal.

Fractals can be classified as deterministic or random. Deterministic fractals are those that are created with a generator, while random fractals contain elements of randomness; these random fractals are frequently used to simulate natural phenomena (leaves on trees, branches, ... ).

The most famous deterministic fractals are the Koch Snowflake and the Minkowski Island. The Koch Snowflake (see figure 3.10) begins with a triangle (initiator) and each of the three sides is replaced with a generator. In this case the generator is an equilateral triangle, as is shown below the following figure.

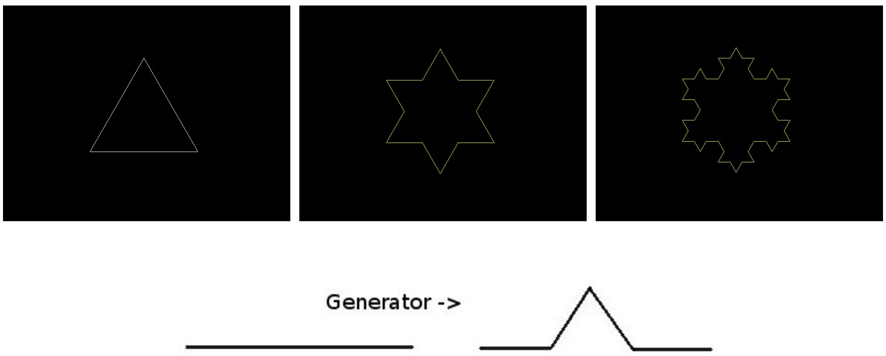



Figure 3.10: Koch Snowflake

The generator divides the segment into three equal parts, removes the central section and adds two segments of the same size. The angles correspond to an equilateral triangle. The process is repeated recursively, applying the generator to each of the resulting segments.

The Minkoswi Island (see Figure 3.11) is the same process, but begins with a square (initiator). Each of the four sides is replaced with a generator that will be another square. As with the Koch Snowflake, the first three generated iterations are displayed in the figure.

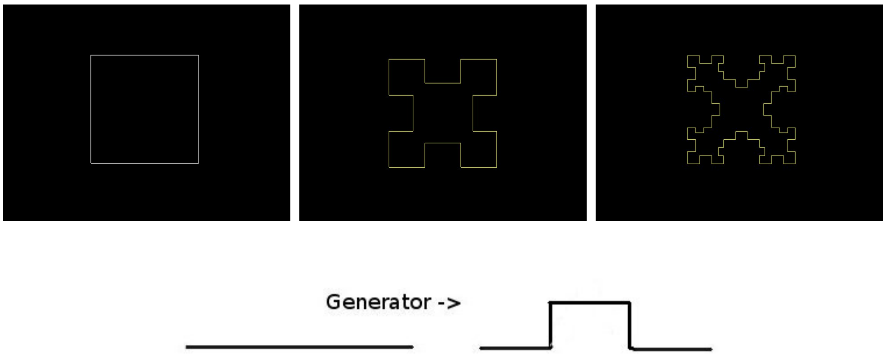



Figure 3.11: Minkowski Island

Obviously, these are only two of many possible geometries. Other geometries lead to other factals: Sierpinski carpet, Sierpinski triangle, Space-filling curve, Dragon curve, Menger sponge, etc.[23]

The trend of a fractal antenna geometry can be deduced by observing several iterations of the process. The final fractal geometry is a curve with an infinitely complicated underlying structure such that the fundamental building blocks can not be differentiated because they are scaled versions of the initiator.

Currently fractal antennas are being applied extensively with very good results in terms of efficiency, space, bandwidth, and gain. For these reasons, we decided to investigate and simulate these types of antennas.

Nowadays, there has been an increasing demand for more compact and portable systems. The goal of reducing the size of the antennas without affecting the read range has led to the study and implementation of fractal antennas. We can find many of them in academic studies, such as [24]. In this paper, Ferchichi Abdelhak shows how to use different fractal structures to reduce the size of an antenna, and in this case, a Sierpinski fractal antenna obtains a good gain result and a good reduction in the size of the antenna.

Currently, commercial fractal antennas are available for RFID. For example, Fractal Antenna Systems Inc. [25] develops, designs, and manufactures antennas for the commercial, military, and government sectors. They provide tags and readers for RFID applications emphasizing the compact and low-cost solution of these antennas.

Fractus [26] research and development work has focused on improvements in fractal technologies and the integration of fractal antennas into wireless communications devices and telecommunication operators' infrastructure. The Fractus EZConnect antenna has been specifically designed for wireless devices utilizing Zigbee, Smarthome antennas, RFID antennas, and other wireless standards and applications operating in the ISM 868/915 MHz bands.

The use of fractal antenna for RFID is growing in significance. The good results obtained with fractal technologies and the reduction of the antenna's size have lead to fractal geometries being considered for many RFID applications.[27]

### 3.1.4 Simulations of Fractal Antennas

As described in the previous subsection, fractal antennas are currently applied in many fields of technology. Some time ago, researchers from Finland's Tampere University of Technology showed that at UHF frequencies, a fractal antenna gave better results than some traditional antenna designs [28]. The read range achieved from a fractal antenna is longer than for other antennas, so this type of antenna seems to be very suitable for our problem.

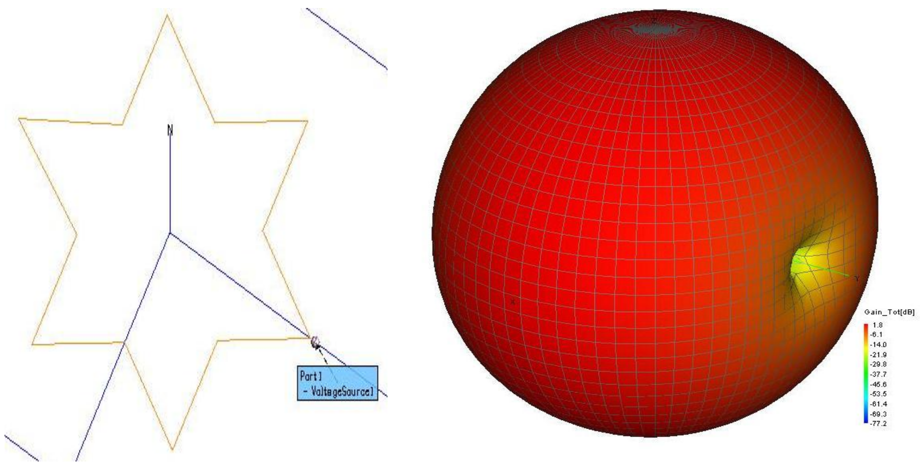
The first step was to create a program in C to compute the layout of a fractal antenna. The FEKO program needs the coordinates of the bends in the antenna. For the first iteration of a fractal antenna it is easy to calculate these points, but for the second iteration we required a program. Two programs were implemented. One to create a Koch Snowflake and the other to create a Minkowski Island (both of these fractals were shown in earlier sections).

The program computes the fractal interactively. The user can instruct the program to perform another iteration of the generator by pressing the Page_Up Key. After each iteration the program displays the pattern computed thus far. In the case of Knoch Sbowflake, it draws an equilateral triangle with side 2, and for Minkowski Island it draws a square with side 2. The reason for these choices was the ease of the subsequent calculations.

The C code for each of these is includes in an appendix: Appendix A contains the code for the Koch snowflake and Appendix B the Minkowski Island. Using the *gcc* compiler we can compile the programs, for example for Ubuntu: $gcc -o fractal_koch fractal_koch.c -lGL -lglut. Once we have calculated the points we input these coordinates into FEKO and then perform a simulation. The results of these simulations are described in the following sections.

#### 3.1.4.1 Koch Snowflake

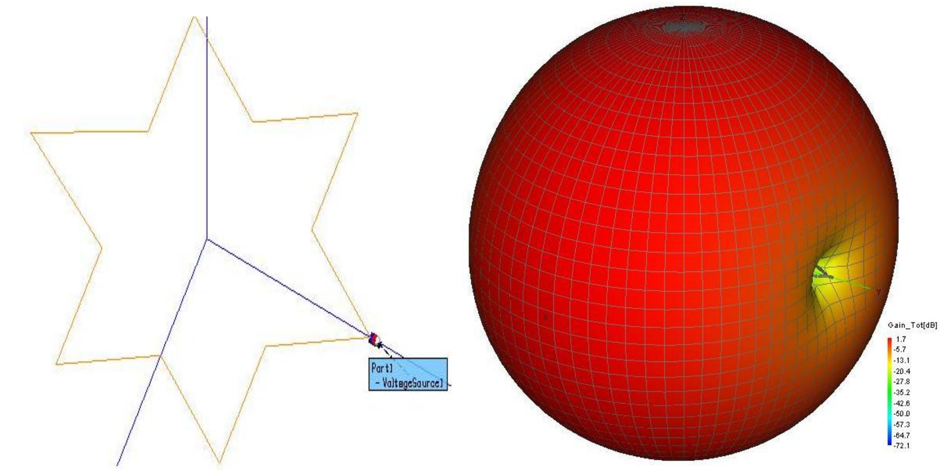
Figures 3.12 and 3.13 show the two first iterations for the Koch Snowflake for two difference sizes. Figure 3.12 shows the first iteration of a Koch Snowflake 2 meters in height and 2 meters wide. The resulting gain is 1.7 dB.




(a) Koch Snowflake Structure (b) Radiation pattern of the gain

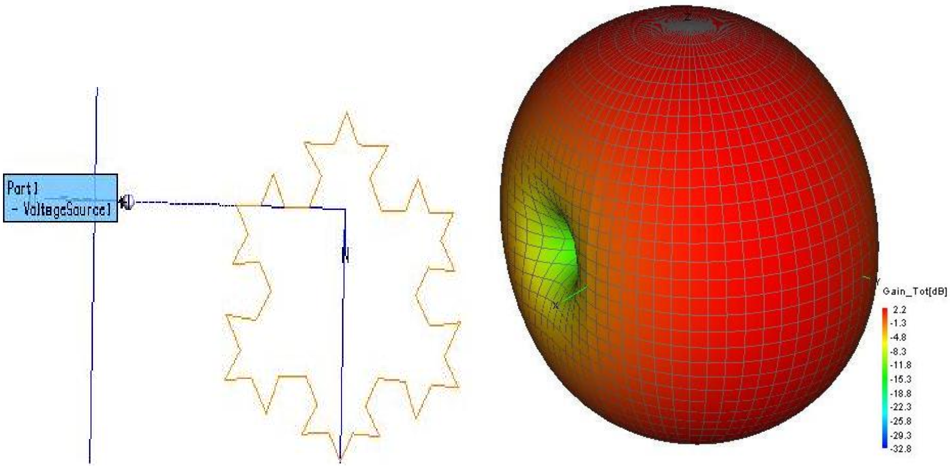
Figure 3.12: Koch Antenna first iteration, 2 m x 2 m

Scaling the antenna to 30 cm x 30 cm we still get 1.7 dB of gain (Figure 3.13), for 40 cm x 40 cm we get a 1.5 dB of gain, and for 50 cm x 50 cm just 1.2 dB. Next, we evaluate these antennas for a second iteration of the Koch Snowflake. We repeat the simulations for the second iteration with an antenna of 2 meters of height and 2 meters of width (Figure 3.14). The result is 2.2 dB of gain, for 30 cm x 30 cm 2.3 dB (Figure 3.15), 1.7 dB for 40 cm x 40 cm, and 2 dB for 50 cm x 50 cm. We can see that second iteration gives better gain, but are still not sufficient for our problem. Next we will consider the Minkowski Island.




(a) Koch Snowflake Structure (b) Radiation pattern of the gain

Figure 3.13: Koch Antenna first iteration, 30 cm x 30 cm

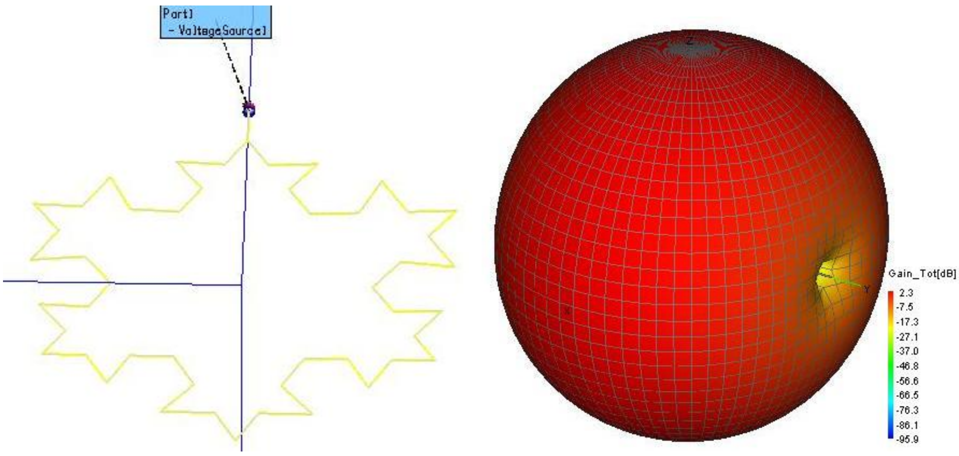



(a) Koch Snowflake Structure (b) Radiation pattern of the gain

Figure 3.14: Koch Antenna second iteration, 2 m x 2 m

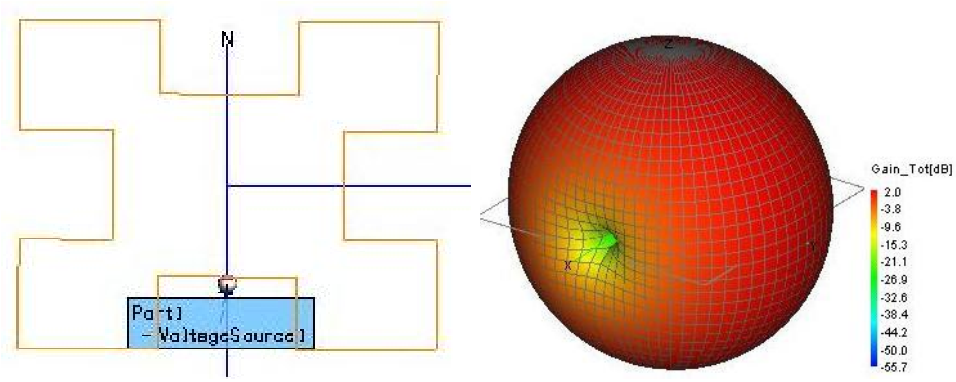
#### 3.1.4.2 Minkowski Island

As we did with the Koch Snowflake, we simulate the first and the second iteration for the Minkowski Island. The first iteration for 2 meters of height and 2 meters of width gives a 2 dB of gain(Figure 3.16), and scaling this gives very bad results, -0.6 dB for 30 cm x 30 cm (Figure 3.17), and 0.6 dB for 40 cm x 40 cm. Scaling to 50 cm x 50 cm gives 2 dB of gain, still not sufficient. This we proceed to the second iteration.




(a) Koch Snowflake Structure (b) Radiation pattern of the gain

Figure 3.15: Koch Antenna second iteration, 30 cm x 30 cm

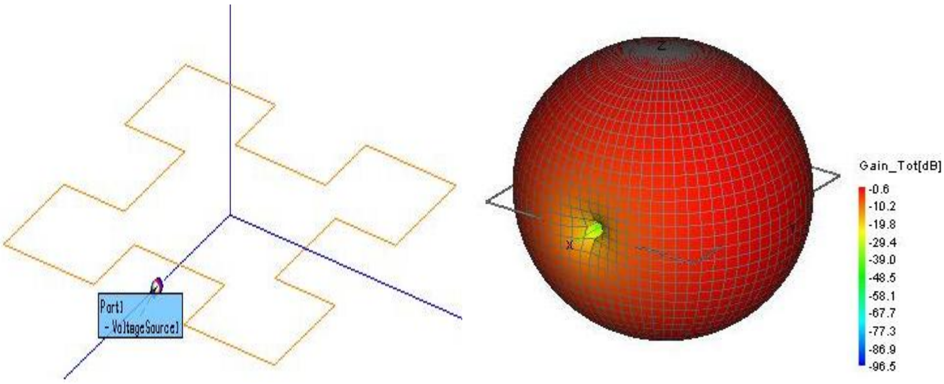



(a) Minkowski Island Structure (b) Radiation pattern of the gain

Figure 3.16: Minkowski Antenna first iteration, 2 m x 2 m

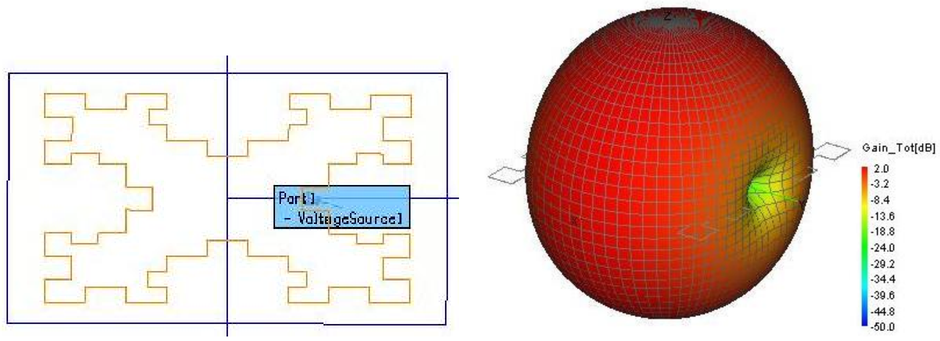
Simulating the second iteration, we obtain 2 dB of gain for 2 m x 2 m (Figure 3.18), but scaling the antenna leads to interesting results.

For 30 cm x 30 cm we get 18.7 dB of gain (Figure 3.19). This gain is sufficient to achieve our goal. However, before rushing to build this antenna we consider several different alternatives to see if we can get more gain. These results are summarized in Table 3.1.




(a) Minkowski Island Structure (b) Radiation pattern of the gain

Figure 3.17: Minkowski Antenna first iteration, 30 cm x 30 cm



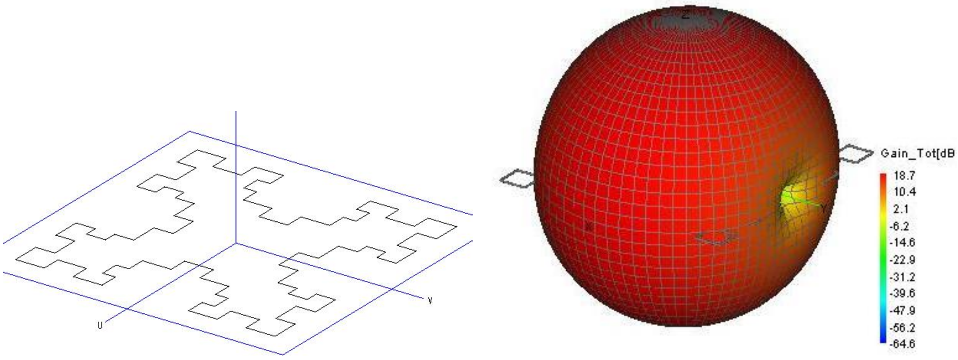

(a) Minkowski Island Structure (b) Radiation pattern of the gain

Figure 3.18: Minkowski Antenna second iteration, 2 m x 2 m

Table 3.1: Results with differents measures

| | |
|---|---|
| 30 cm x 30 cm | 18.7 dB |
| 40 cm x 40 cm | 19.4 dB |
| 50 cm x 50 cm | 20.2 dB |
| 60 cm x 60 cm | 21 dB |
| 70 cm x 70 cm | -0.9 dB |

With these results, now we know that our antenna has to be between 30 cm x 30 cm and 60 cm x 60 cm, to obtain sufficient gain. See Figure 3.19 for




(a) Minkowski Island Structure (b) Radiation pattern of the gain

Figure 3.19: Minkowski Antenna second iteration, 30 cm x 30 cm

30cm x 30cm, Figure 3.20 for 40cm x 40cm, Figure 3.21 for 50cm x 50cm, and Figure 3.22 for 60cm x 60cm. All of these results are more gain than we need for reading at 1 m (16 dB). So, we have found an antenna suitable for our needs.

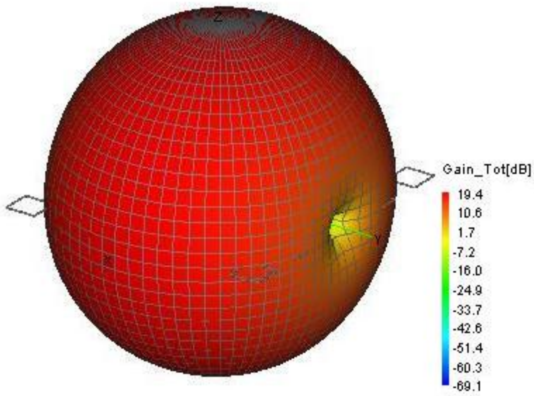



Figure 3.20: Minkowski Antenna second iteration, 40 cm x 40 cm: Radiation pattern of the gain

Now, we will try to add a reflector to see if we can further increase the gain. If we put a reflector (a simple sheet of perfect conductor) at different distances from the above antenna (where the distance is in fractions of $\lambda$) for a 60 cm x 60 cm Minkowski Antenna, we obtain the results shown in Table 3.2. In the best case ($\lambda/4$), the gain is the same as without a reflector (Figure 3.23), thus adding a reflector does not improve the gain of the Minkowski Island Antenna.

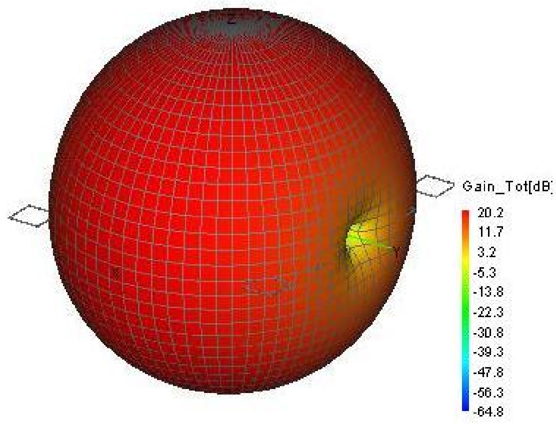



Figure 3.21: Minkowski Antenna second iteration, 50 cm x 50 cm: Radiation pattern of the gain

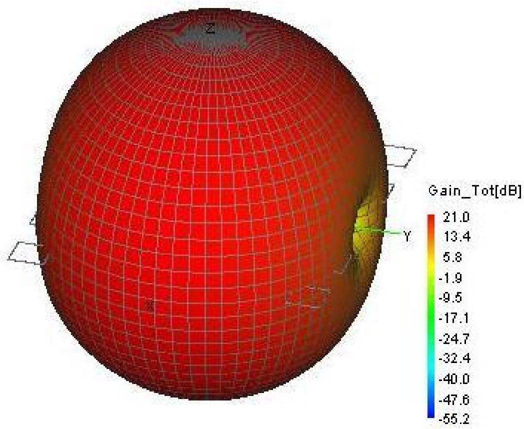
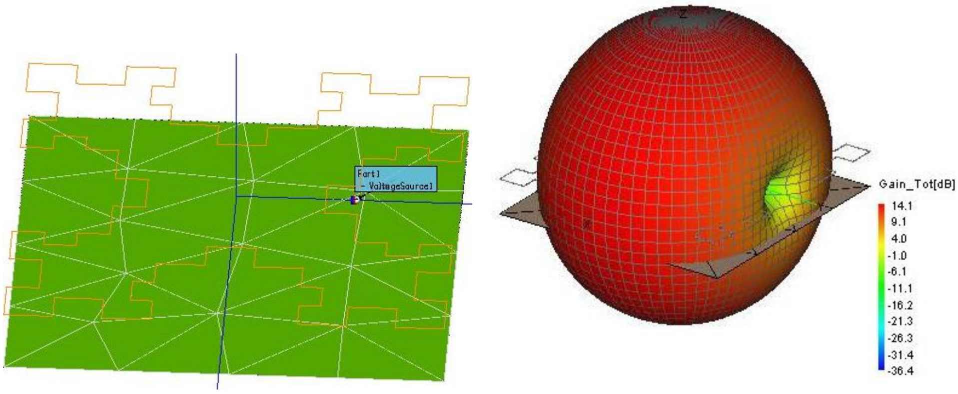



Figure 3.22: Minkowski Antenna second iteration, 60 cm x 60 cm: Radiation pattern of the gain

Table 3.2: Gain with reflector

| | |
|---|---|
| $\lambda/32$ (0.69 m) | 20.7 dB |
| $\lambda/128$ (0.17 m) | 17.2 dB |
| $\lambda/4$ (5.53 m) | 21 dB |
| $\lambda/256$ (0.08 m) | 14.1 dB |

Thus, after all the simulations and all the obtained results, we decided to test a protoype of the Minkowski Island antenna. We started building the Minkowski Island second iteration with a cooper wire, and then to reduce the




(a) Minkowski Island Structure (b) Radiation pattern of the gain

Figure 3.23: Minkowski Antenna second iteration, 60 cm x 60 cm with a reflector at $\lambda/4$

size and to test the second and third iterations we fabricated PCB antennas as described in the next chapter.

# Chapter 4

# Testing and Analysis

## 4.1 Prototype Design and Implementation

In this section we describes the implementations of the chosen fractal antennas. First, we will see the second iteration of Minkowski Island which was built with copper wire, and next, with the second and third iteration built on a Printed Circuit Board (PCB).

### 4.1.1 Minkowski Island: Copper wire antenna

Our first prototype was a Minkowski Island with a second iteration 60cm x 60cm antenna to measure if the results in practice were in keeping with the simulations. For this purpose we used a copper wire 2 mm in diameter. The prototype was built on a sheet of cardboard and the wire was fixed to the cardboard using plastic cable ties. The final antenna is showed in Figure 4.1.

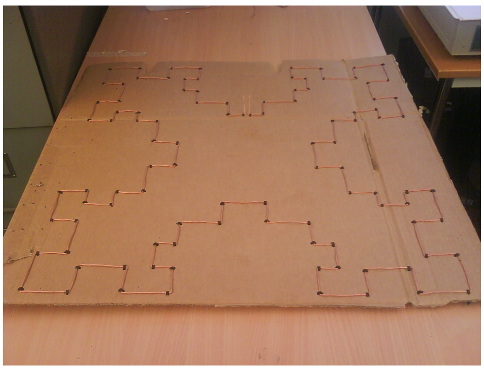


Figure 4.1: Prototype of Minkowsky Island second iteration 60cm x 60cm antenna

After building this antenna we had to carry out matching in order to connect this antenna to our RFID reader for optimum performance. The HF Antenna design Notes by Texas Intruments [29] describes how to do this matching. The first step was an inductance measurement. Using a Philips RCL meter PM 6303A we measured an inductance of 4.2 $\mu$H. Following the instructions of Texas Instruments, to lower the Quality Factor (Q) of our antenna we needed to use a "Swamping Resistor". The Q of the antenna when connected to a 50 Ohm load, i.e. the reader, should be 20 or less, so first it is necessary to calculate the total resistance of the antenna having the required Q of 20:

$$R_{par1} = 2\pi f L Q \tag{4.1}$$

For our data, 13.56 MHz and 4.2 $\mu$H, hence to have a Q of 20 we calculated that we needed a $R_{par1}$ of 7156.79 Ohm. Now, we had to assume a value for the present Q in the antenna (say 50) and repeat the calculation, obtaining a $R_{par2}$ of 17891.99 Ohm. The required resistance can be calculated using the following formula:

$$R = \frac{1}{\frac{1}{R_{par1}} - \frac{1}{R_{par2}}} \tag{4.2}$$

Finally, with the values of $R_{par1}$ and $R_{par2}$ calculated above, the swamping resistor for the antenna was 11927.97 Ohm. To perform the matching of the antenna, we had to attach the swamping resistor and two variable capacitors to the open ends of the wire antenna. One variable capacitor was a 'coarse adjustment' and had a range of about 10 to 120 pF. The other variable capacitor was for 'fine adjustment', and had a range of about 5 to 15 pF. These two capacitors were used to find the value of capacitance needed for the antenna to be resonant at 13.56 MHz. The method used for matching the antenna was T-Matching. This method taps the antenna loop for the matching points as showed in the Figure 4.2. This type of matching is 'balanced' as both screen and centre core of the coax cable are tapped and not just the core.

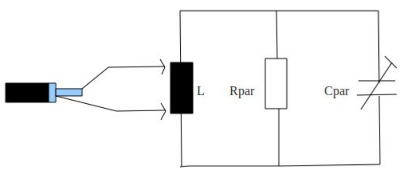



Figure 4.2: T-Matching Circuit

Using a MFJ Enterprises MFJ-259C antenna analyzer, the next step was to tune the antenna and find the matching points. The procedure to follow was to connect the antenna to the antenna analyzer and adjust the variable capacitors and the position of the T-match arms until the antenna had a characteristic impedance as close to 50+j0 as possible. After matching the antenna we connected the antenna to the reader to measure the reading range with this antenna. The first measurements were between 5 and 6 cm. Although the reader used in the lab only radiates a maximum of 0.125 Watts, this distance and the size of our antenna were not sufficient to meet our goal. At this point, we decided to reduce the size of the antenna and try to make it on a PCB.

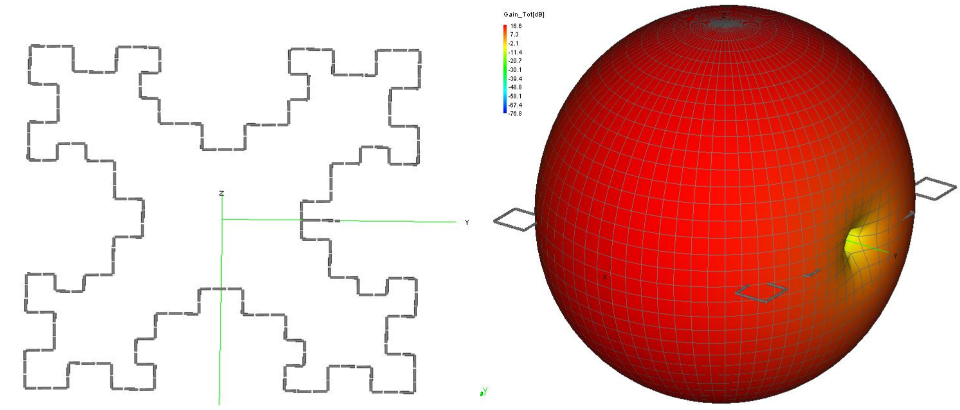
### 4.1.2 Minkowski Island: Printed Circuit Board

Once we decided to make the antenna using a PCB, we decided to build both a two and a three iteration Minkoski Island in order to compare both designs. The PCB used for the designs was FR-4. FR-4 is a composite material composed of woven fiberglass cloth with an epoxy resin binder that is flame resistant and is the most common PCB material. The copper foil thickness of the board was 35 $\mu$m.

First of all we had to simulate antennas of various sizes to know if these antennas would show a good gain to achieve long distance. The copper-clad boards used for the PCB milling machine measure 22.9 cm x 30.5 cm, so we simulated designs that could be realized on PCBs of this size. After simulating several antennas with different sizes and different radius for the tracks, the ones that showed the best gain were:

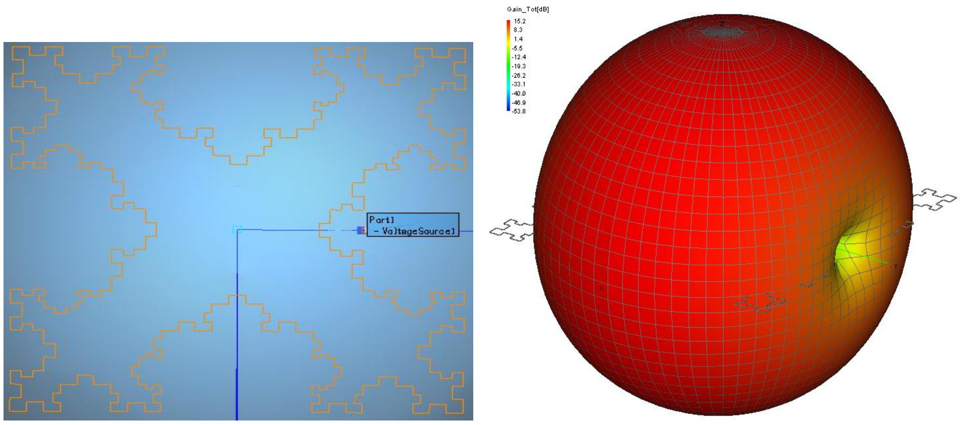
- 16.6 dB for 20 cm x 20 cm for the 2 iterations design and 0.4 mm diameter (Figure 4.3).
- 15.2 dB for 22.6 cm x 22.6 cm for the 3 iterations design and 0.2 mm diameter (Figure 4.4)

To send the design to the PCB milling machine, we had to create Gerber files for the antennas. Both of these files are included in an appendix of this thesis. Appendix C is the Gerber file for Minkowski 2 iterations and Appendix D is for 3 iterations. These files describe the images of a PCB and all the information necessary for manufacturing the PCB. These files were sent to the LPKF ProtoMat S42 milling machine by using the LKSoft programs Circuit Cam 5.2.713 and Board Master 5.0.1100.K02.1. The milling machine just cuts the traces on the copper clad board, so our next task was to remove all the remaining copper on the board. The copper clad board is composed of a copper sheet attached to a fiberglass board by using Epoxy, and heating up the board with a heat gun we could easily remove the unneeded copper, leaving just the desired traces of the design. After removing all the undesired copper, both antennas were ready to perform matching and then to test them (Figure 4.5).



(a) Minkowski Island Structure, 2 iterations (b) Radiation pattern of the gain

Figure 4.3: Minkowski Antenna second iteration, 20 cm x 20 cm




(a) Minkowski Island Structure, 3 iterations (b) Radiation pattern of the gain

Figure 4.4: Minskowski Antenna third iteration, 22.6 cm x 22.6 cm

## 4.2 Test of printed circuit board antenna

In order to test the new antennas, we had to repeat the matching process for both PCBs. As we did before, the first step was the inductance measurement using the RCL meter in the laboratory. The values obtained for these antennas were:

- 2 iterations antenna, 20cm x 20 cm, diameter 0.4 mm : 1.5 $\mu$H
- 3 iterations antenna, 22.6cm x 22.6 cm, diameter 0.8 mm : 1.9 $\mu$H

With these values for the inductance, we had to calculate the swamping resistors for each antenna using formulas 4.1 and 4.2, and the final results are showed in Tables 4.1 and 4.2.

Figure 4.5: Process to remove the remaining copper from the copper clad board

Table 4.1: Swamping resistor for 2 iterations antenna

| | |
|---|---|
| $R_{par1}$ | 2726.39 Ohm |
| $R_{par2}$ | 5811.99 Ohm |
| R | 3678.9 Ohm |

Table 4.2: Swamping resistor for 3 iterations antenna

| | |
|---|---|
| $R_{par1}$ | 3237.59 Ohm |
| $R_{par2}$ | 8093.99 Ohm |
| R | 5395.99 Ohm |

Again, we repeated the same process to perform matching as described in Section 4.1. To achieve an optimum performance for these antennas, we had to attach the variable capacitors and the swamping resistor calculated above. The final result is shown in Figure 4.6.

In order to carry out the matching, we used T-matching for both antennas as we did before, but for these antennas it was not so straightforward.

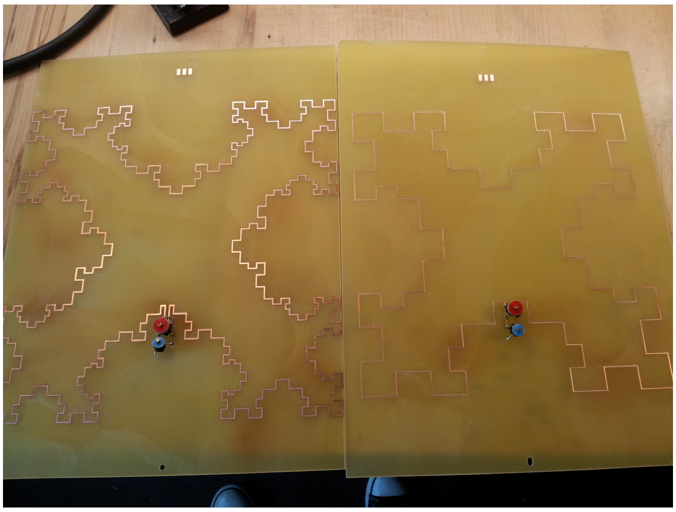


Figure 4.6: PCB Minkowski antennas for 3 and 2 iterations

Although we could not achieve perfect matching for the 2 iterations antenna, the antenna was matched fairly well, as shown in Figure 4.7.

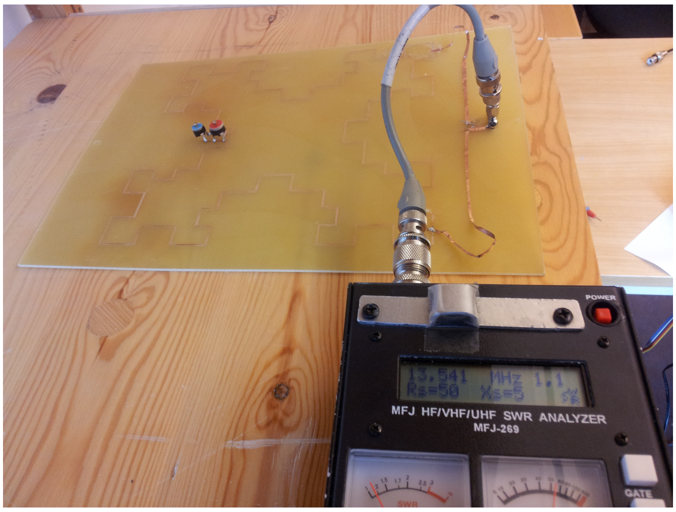



Figure 4.7: Matching for PCB Minkowski antenna 2 iterations

After performing the matching, we checked the read range of this antenna. The reader used for this project has been the Texas Instruments S6350 inductive RFID reader [30] and the tag used for the reading was an

ISO 15693 tag by Texas Instruments [31]. Using this tag, the antenna could read up to 14 cm away from the antenna.

At this point, we continued with the 3 iterations antenna, but for this antenna a correct matching was not possible. After changing the position of the T-match arms and adjusting the variable capacitors for every new position, we were not able to find a proper position. To observe the behaviour of this antenna, we decided to take different measures. We took a position for the T-match arms and we adjusted the capacitors to obtain different values for the characteristic impedance. We read the tag for every value and for several positions of the T-match arms. Some of the obtained results are showed in Table 4.3:

Table 4.3: Obtained results for the T-matching of the 3 iteration antenna

| | |
|---|---|
| Position 1: Rs=51, Xs=35 | 12.5 cm |
| Position 1: Rs=20, Xs=0 | 12 cm |
| Position 1: Rs=43, Xs=18 | 13 cm |
| Position 2: Rs=49, Xs=17 | 12.1 cm |
| Position 2: Rs=22, Xs=1 | 12.4 cm |
| Position 2: Rs=45, Xs=21 | 12 cm |
| Position 3: Rs=50, Xs=35 | 13 cm |
| Position 3: Rs=21, Xs=0 | 12.8 cm |
| Position 3: Rs=34, Xs=29 | 12.5 cm |

Although the table above does not show all the results for all the positions that we tried, we can observe that the read range of the antenna was up to 13 cm.

Next, we compared our experimental results with the earlier simulations. Using Equation 3.1, we calculated the read range we should obtain with these antennas and with the reader and tag used for the measurements.

For a maximum radiated power of the reader 0.125 Watts, the 2 iterations antenna with a gain of 16.6 dB, the simulations indicated that the antenna should have a read range of around 28 cm; and for the 3 iterations antenna with a gain of 15.2 dB the antenna should have a read range of around 23 cm. Although the results with our antennas are not the same as expected from the simulations, we have to remember that the simulations were done considering an ideal environment and an ideal antenna that is ideally adapted.

## 4.3 Session Initiation Protocol

Once the main aim of this thesis project had been developed and tested, we moved on to see how the SIP protocol could utilize the readings from the reader to give information about the user and their location.

Session Initiation Protocol (SIP) is a standard protocol that is widely used by the networking industry and by the research community. One of its features is support user mobility. The target of a SIP INVITE can be in one or more locations and these locations can dynamically change, so a SIP proxy decides where to direct a SIP request at the time of the request.

In our system, the data collected by the RFID reader is distributed to a SIP proxy server using the SIP protocol. In order to process the readings from the reader and to provide a notion of presence, a SIP for RFID management system was required. An example of such a system is described in [32] and [33].

In order to realize a SIP for RFID management system we begin by noting that as per Section 2.2 the SIP architecture model is based on SIP server(s) and SIP user agents. A user agent is an entity that sends SIP requests or returns SIP responses. A SIP server can implement on or more of the following functions: registration, redirect, and proxy.

The registration function accepts SIP REGISTER requests and registers the current location of the user agent. A redirect server responds to a SIP INVITE request by indicating the user agent server that is prepared to forward INVITE requests to a SIP proxy. The SIP proxy knows the current location of the user's SIP usage agent server. The proxy function's role is to forward SIP messages to relevant the SIP user agent servers. To do this, the proxy server uses the SIP destination represented by a Uniform Resource Identifier (URI) and a location service (that can access the locations of a SIP user agent server based upon earlier REGISTER requests) to determine the current location of the destination. Notice that a user agent must register its current location before a SIP session can be initiated by a SIP user agent client.

To realize this SIP for RFID management system a SIP user agent server sends REGISTER requests based upon the location of an RFID tag as detected by the RFID reader. The SIP Registration server interacts with the location database to record the tag's current location. The location database is responsible for managing this data. The proxy and redirect functions realize a tracking function.

The location of the tag is mainly determined by the IP address of the reader to which it is attached, i.e., each RFID tag reader has a location and this location is mapped to and from its corresponding IP address. Each RFID tag is associated with a SIP URI. The SIP URI includes the RFID tag's value. Now we can add the reader's IP address where the tag was read to the URI. This URI directly encodes the tag's value and the IP address of current reader's IP address, hence via the database we can learn the location where the RFID tag was read.

### 4.3.1 SIP registration - Location function

To perform the task described above, the RFID middleware has to behave as a SIP user agent server. When the reader detects the presence of the tag within the detection area of the reader it generates a SIP registration request for this tag. More precisely a SIP REGISTER message is sent to a SIP Registrar server indicating in its Contact field the RFID tag and the IP address associated with the reader. With this information, the SIP Registrar server updates the location database. When a tag leaves the area of the reader, a SIP REGISTER request with a field "Expires" = 0 is generated. The complete procedure can be summarized in the following steps:

**Step 1.** The tag appears in the detection zone of the reader and the tag's information is collected.

**Step 2.** The reader transmits the acquired information to the RFID middleware that performs filtering and collecting functions.

**Step 3.** The SIP user agent server running in this middleware generates a SIP REGISTER message and sends it to the SIP Registrar server. The SIP REGISTER contains the Contact field in a URI form which is the current location of the RFID tag. The registrar server registers this information in the location database.

**Step 4.** If the registration is succesful, the registrar generates a 200 OK response to the user agent. If the system is set up to only support authorized users, then the SIP registrar server generates a 407 SIP message type to request the credentials of the RFID middleware before accepting the registration.

### 4.3.2 SIP redirection - Tracking function

To carry out tracking function we used SIMPLE protocol. Primary task of SIMPLE protocol is the transport of instant messages in SIP. A message is sent to the presence server over SIMPLE protocol as a PUBLISH message, publishing all tag location changes. It means that all subscribers whose SIP session is opened, can see the list of active tags. The server agent handles requests such as information update storage or database information. The SIP server receives RFID tag information from the presence server to notify about presence of tags we are looking for. The SIP server database updates presence information when location is changed.

When a tag is read by an RFID reader, local computer with RFID application installed, generates log about tag presence. Those logs are parsed and changes of RFID tag location are sent to the Middleware. SIMPLE protocol is used for SIP extendible messaging. Three important messages are sent in the communication between the SIP server and the SIP client: PUBLISH, SUBSCRIBE, and NOTIFY.

The presence server handles SIP PUBLISH messages, which publishes the SIP address, in order to find the location of the tag. PUBLISH message allows the user agents to inform the presence server about their subscription states and presence status.

When tag changes location, a new SIP message is created with new reading information about tag changes. In order to receive tag status, clients must SUBSCRIBE to presence of tags. A NOTIFY message is received by the subscriber when the status of user changes.

# Chapter 5

# Conclusions and Future Work

In this chapter, we will conclude this master thesis and propose some suggestions for the improvement of the system along with suggestions for future work.

## 5.1 Conclusions

As described in Section 1.2, our aim was to detect the presence of a user and the user's identity using RFID technology. In this thesis, we have investigated, designed, implemented, and evaluated an antenna for an RFID reader to increase the reading range at 13.56 MHz. We use this information to provide additional information to realize SIP-based RFID location system.

Through this research, we realized it was possible to simulate the behaviour of the designed antennas. All of these simulations were done using FEKO. These simulations helped guide us to a suitable design for the reader's antenna. We concluded that using a fractal pattern gave greater and more suitable gain, hence enabling us to achieve our goal of long range reading, hence such an antenna could used for detecting the user's presence and reading the RFID to establish their identity.

During this thesis project, we built and tested several prototypes to check if the results of the simulations were correct. First we built a 2 iterations Minkowsky Island 60cm x 60cm using copper wire. However, the obtained result were far from what the simulations predicted and the size of the antenna was not appropiate to incorporate into a doorframe. We decided to reduce the size of the antenna by using printed circuit board antennas. These prototypes gave better ranges than the previous antenna prototypes.

The read range for the 2 iterations antenna was up to 14 cm and for the 3 iterations antenna was up to 13 cm. Based on this, we suggest the use of fractal antennas rather than antennas with other designs. The use of fractal antennas improved the reading range and allowed us to detect the user's presence from an interesting distance (in the sense of being able to detect and identify as user passing through a doorway). Although these results were not the same expected from the simulations, we have to bear in mind that the simulation with FEKO was for an ideal antenna that was perfectly adapted. In practice the matching of these antennas was not easy to do and for the 3 iterations antenna was impossible to do.

While the proposed reading distance was not achieved with a transmitted power of 0.125 Watts, even if we place two antennas of the reader in both sides of a doorway, an advanced prototype was presented which reaches a reading range far from what is achieved otherwise.

As to context information, we described the most important features of SIP and how SIP would be exploited by our system. Using SIP enables us to support the notion of presence, this makes SIP very suitable for our system and for our purpose. SIP was designed to be a modular, flexible component in the Internet architecture making it a powerful and efficient protocol for future uses in presence applications, such might be used in conjunction with the Internet of Things.

Evaluation of the system showed that the printed circuit board antennas obtained good results given the maximum radiated power of the reader and for the matching done with these antennas. However, in order to fulfill the specific requirements for the proposed purpose we would like to improve the range further and make use of some of the different suggestions for future work given in the next section.

## 5.2 Future Work

For a successful antenna design, theoretical analysis and numerical simulations are only part of the necessary work. The testing and measurement of the antenna's design are very important and potentially enable us to modify and improve the performance of the antenna. The benefit of a fractal antenna is that it provides a very good use of the available space and increases the range of detection of a passive RFID tag. Many studies and companies are using fractal antennas for their designs and products. There are many fractal patterns which could be simulated in order to review their behaviours. Even more iterations of the Minkowski Island could be investigated to evaluate performance of these antennas.

Some studies of the use of fractal designs for antennas to read RFID tags have been carried out. These studies have shown good performance with respect to reading ranges, providing longer read ranges than the other antennas studied. The design, analysis, and testing of a fractal antenna for an RFID reader in combination with our prototype fractal reader antenna should be supplemented with future investigations.

During our research on a reader antenna, some simulations with metamaterials [34] were done with very interesting results. The final gain of the simulated antennas with FEKO was, in some cases and at certain distances, considerably increased. Because of the fact that multiple properties of metamaterial could improve the performance even further, this is an obvious area for future research and development.

As noted in Section 4.2, the matching of the antenna for the 3 iteration Minkowski Island was impossible. Although we tested the antenna with different matching positions and it seemed that even at the matching point the read range would not change highly, there are numerous techniques to accomplish the matching. For example, one could use a Balun to realize Transformer Matching, which although more complicated might achieve matching for the 3 iteration antenna.

In our work we used the Texas Instruments S6350 inductive RFID reader with a maximum radiated power of the 0.125 Watts. Systems designers should consult their local government agencies to determine the legal limits for the RF field generated from the antenna. The proposed system meets the requirements in both Europe and USA with respect to putting 4 W into a standard 30 cm x 30 cm antenna. Using another reader that provides more radiated power, the read range of the antenna would be longer; but this reader still needs to meet the local regulations. There are several commercial power amplifiers which could be placed between the RFID reader and the antenna to provide additional radiated power from the antenna.

The most important aspect of the system was to detect the user presence at a long range. The proposed antenna has a size of 20 cm x 20 cm for the second iteration and 22.6 cm x 22.6 cm for the third iteration of the Minkowski Island. These sizes are not large enough to detect a tag which users can carry anywhere on their person. However, using an array along the doorframe of several Minkowski Island antennas tags should be detected no matter where the user is carrying the tags.

The copper-clad boards used for our prototypes had a copper foil thickness of 35 $\mu$m. Simulations with greater thickness have been completed suggesting that thicker foils would give an improvement in gain in comparison with the PCB stock that was used. The implementation of fractal antennas on PCB FR-4 with greater thickness could give an increased reading range.

Furthermore, from the report of Zhijia Wang and Karsten Becker [35] we can see the use of multiples antennas with switching between the antennas by using a double Pole Single Throw relay could be successful. This method shows a suitable way to locate antennas on both sides of the doorway. As described in this thesis, it is possible to know which antenna detected the tag, so that the tag would be detected no matter which side of the doorway the user is closest to. Using antennas with a good gain on both sides of the doorway would cover the desired read range and definitely achieve the desired range.

## 5.3 Reflections

In our days, RFID technology is used in many fields. An antenna for the RFID reader such as this prototype can help to several systems to identify users for instance, in a library, in housing developments, in a particular room of a building, etc. This type of system can give multiple advantages to the user as we saw in Section 1.2.

In this thesis we used passive tags. The fact that they do not have batteries makes them more suitables for environmental purposes. They allow us to save energy and there is no need to replace batteries. The useful life of a passive tag can be of twenty years of more which aids in environmental sustainability practices.

All these RFID systems have controversial opinions and, although RFID technology raises numerous ethical issues, the overall benefits that RFID technology provides, including increased safety and efficiency, outweigh the concerns. However, we believe that while RFID technology has enough benefits to be a useful technology, it needs more control and strong regulations to make sure it is safe from potential abuse such as stealing identity data from RFID tag.

# Bibliography

[1] Daniel Dobkin. *The RF in RFID: Passive UHF RFID in Practices*. Elsevier, UK, 2008.

[2] A. Johnston J. Peterson R. Sparks M. Handley J. Rosenberg, G. Camarillo and E. Schooler. *SIP: Session Initiation Protocol*, Internet Request for Comments, RFC 3428 (Proposed Standard), RFC Editor. https://www.ietf.org/rfc/rfc3261.txt, June 2002.

[3] Xueliang Ren. *A meeting detector to provide context to a SIP Proxy*, School of Information and Communication Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, Master Thesis. http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/081025-Xueliang_Ren-with-cover.pdf, October 2008.

[4] Hu Lidan. *An Intelligent Presentation System*, School of Information and Communication Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, Master Thesis. http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/080807-Hu_Lidan-with-cover.pdf, August 2008.

[5] SCDigest Editorial Staff. *RFID News: Looking Back at the Wal-Mart RFID Time Line*. http://www.scdigest.com/assets/On_Target/09-02-23-1.php?cid=2275&ctype=content, 2009.

[6] Mark Smith and Mat Hans. *Sensor-Enhanced Authentication Token for Dynamic Identity Management*. http://www.hpl.hp.com/techreports/2004/HPL-2004-226.pdf, December 2004.

[7] Preview of the first part of ISO 10536. http://webstore.iec.ch/preview/info_isoiec10536-1%7Bed2.0%7Den.pdf. The current edition can be found at: http://www.iso.org.

[8] Third part of ISO 14443. http://jpkc.szpt.edu.cn/2007/sznk/UploadFile/biaozhun/iso14443/14443-3.pdf. The current edition can be found at: http://www.iso.org.

[9] First edition of the second part of ISO 15693. http://jpkc.szpt.edu.cn/2007/sznk/UploadFile/biaozhun/iso15693%u534F%u8BAE/ISO%20IEC%2015693-2.pdf. The current edition can be found at: http://www.iso.org.

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. *SIP: Session Initiation Protocol*, Internet Request for Comments, RFC 3261 (Proposed Standard), RFC Editor. http://www.rfc-editor.org/rfc/rfc3261.txt, June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141.

[11] P. Resnick. *Internet Message Format*, Internet Request for Comments, RFC 2822 (Proposed Standard), RFC Editor. http://www.rfc-editor.org/rfc/rfc2822.txt, April 2001. Obsoleted by RFC 5322, updated by RFCs 5335, 5336.

[12] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. *Session Initiation Protocol (SIP) Extension for Instant Messaging*, Internet Request for Comments, RFC 3428 (Proposed Standard), RFC Editor. http://www.rfc-editor.org/rfc/rfc3428.txt, December 2002.

[13] J. Rosenberg. SIMPLE Made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence Using the Session Initiation Protocol (SIP). *Internet Request for Comments*, RFC 6914 (Informational), April 2013.

[14] Kamailio® web page. http://www.kamailio.org/w/., 2012.

[15] Ilan Kirschenbaum and Avishai Wooly. *How to build a Low-Cost, Extended-Range RFID Skimmer*, Tel Aviv University, Israel. Master Thesis. http://eprint.iacr.org/2006/054.pdf, February 2006.

[16] Dag System: Electronic Timing and RFID Solutions. http://eu.dag-system.com/.

[17] SIC Transcore. Grupo Telepeaje S.A. http://telepeaje.net/TelepeajeTecnologia/Default.aspx, 2012.

[18] Feko web page. http://www.feko.info/, 2012.

[19] Harmonized Standards for generic Short Range Devices ETSI. http://www.etsi.org/deliver/etsi_en/300300_300399/30033002/01.03.01_30/en_30033002v010301v.pdf, 2006.

[20] Introduction to Dipole antenna. Radio-Electronics web page. http://www.radio-electronics.com/info/antennas/dipole/dipole.php.

[21] Introduction to Yagi-Uda antenna. Radio-Electronics web page. http://www.radio-electronics.com/info/antennas/yagi/yagi.php.

[22] Example Guide Feko. http://people.ee.ethz.ch/~fieldcom/pps-antenna/doc/ExampleGuide.pdf, 2005.

[23] Benoit Mandelbrot (1924-2010) Michael Frame and Nial Neger. Studies by yale university. http://classes.yale.edu/fractals/, 2012.

[24] Ferchichi Abdelhak. *A Sierpenski slot antenna as a TAG RFID Antenna*, International Journal of Communication Networks and Information Security. http://www.faqs.org/periodicals/201012/2198228761.html, December 2010.

[25] Fractal Antenna Systems INC web page. http://www.fractenna.com/, 2012.

[26] Fractus S.A. web page. http://www.fractus.com/, 2012.

[27] Nathan Cohen. *Fractals' new era in military antenna design*, Defense Electronics. http://rfdesign.com/mag/508RFDSF1.pdf, August 2005.

[28] Laurie Sullivan. *In an Academic Study, RFID Fractal Antenna Wins Out,* RFID Journal. http://www.rfidjournal.com/article/view/3334, May 2007.

[29] Texas Instruments. *HF Antenna Design Notes. Technical Application Report*. http://www.ti.com/lit/an/scba034/scba034.pdf, September 2003.

[30] Texas Instruments HF Reader System Series 6000. http://www.ti.com/lit/ds/scbs856/scbs856.pdf, 2001.

[31] Texas Instruments ISO 15693 tag. http://www.ti.com/lit/ug/scbu006b/scbu006b.pdf, 2011.

[32] Taekyoung Kwon Kideok Cho, Sangheon Pack and Yanghee Choi. *SRMS: SIP-based RFID Management System*. School of Computer Science and Engineering, University of Seoul, Korea, July 2007.

[33] Phuoc Nguyen Tran and Nadia Boukhatem. *IP-based RFID Location System*. Computer Science and Network Department, Telecom Paris Tech, September 2008.

[34] Center for Metamaterials and Integrated Plasmonics web page. http://metamaterials.duke.edu/research/metamaterials, 2015.

[35] Zhijia Wang and Karsten Becker. *Multi-tool support for multi touch.* http://karsten-becker.de/projects/MultiTouch.pdf, November 2008.

# Appendix A

# Koch Snowflake Program

Fractal Antennas: Code for **Koch Snowflake** (As a C Program)

```
#include <GL/freeglut.h>
#include <math.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define SCREEN 2.0f

typedef struct {
        float x, y, z;
        }Point;

/* Variables */
float xMax, yMax;

void initialize (void);
void redimension (int width, int heigth);
void draw (void);
void keyboard (unsigned char key, int cx, int cy);
void Skeyboard (int key, int cy, int cx);
void mouse (int button, int state, int cx, int cy);
void koch (Point p0, Point p1, int depth);

/* Variable keeps position if the vertices*/
Point vertex[3];
/* Variable number of Points*/
int numPoints;
```

```
int iter;

int main(int argc, char** argv) {

   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGBA);
   glutInitWindowSize (640, 480);
   glutInitWindowPosition (100, 105);
   glutCreateWindow (argv[0]);
   initialize();

   glutDisplayFunc(draw);
   glutReshapeFunc(redimension);
   glutKeyboardFunc(keyboard);
   glutSpecialFunc(Skeyboard);
   glutMouseFunc(mouse);

   glutMainLoop();

   return 0;
}

void initialize () {
   /*Initialize*/
   glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
   glPointSize(5.0);
   numPoints = 0;
   iter = 1;
}

void redimension (int width, int height) {
   float aspect;

   glViewport (0, 0, width, height);
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity ();

   if (height == 0){
         height = 1;
   }

   aspect = (float) width / (float) height;

   if (width <= height) {
```

```
            glOrtho(-SCREEN, SCREEN, -SCREEN/aspect,
                    SCREEN/aspect, -SCREEN, SCREEN);
            xMax = SCREEN;
            yMax = SCREEN / aspect;

    } else {
            glOrtho(-SCREEN * aspect, SCREEN * aspect,
                   -SCREEN, SCREEN, -SCREEN, SCREEN);
            xMax = SCREEN * aspect;
            yMax = SCREEN;
    }

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard (unsigned char key, int cx, int cy) {
        /*keyboard: If ESC key is pressed, the program finishes*/

        switch (key) {
                case 27:
                    exit(0);
                break;

                default: break;
        }

        glutPostRedisplay();
}

void Skeyboard (int key, int cy, int cx) {

        switch (key) {
                case GLUT_KEY_PAGE_UP:
                    iter++;
                break;

        case GLUT_KEY_PAGE_DOWN:
                    if (iter > 1) iter--;
                break;

                default: break;
        }
```

```
        glutPostRedisplay();
}

void mouse (int button, int state, int cx, int cy) {
        float x, y;


        if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {

                if (numPoints < 3) {
                   vertex[numPoints].x = x;
                   vertex[numPoints].y = y;
                   vertex[numPoints].z = 0.0;
                   ++numPoints;
            }
            else {

                numPoints = 0;
                vertex[numPoints].x = x;
                   vertex[numPoints].y = y;
                   vertex[numPoints].z = 0.0;
                   numPoints++;
            }
        }

        glutPostRedisplay();

}


void draw (void) {
        /*draw the points, we start with an equilateral triangle with
        int i;


        glClear(GL_COLOR_BUFFER_BIT);

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

         vertex[0].x = -1.0;
         vertex[0].y = 0.0;
         vertex[0].z = 0.0;
```

```
        vertex[1].x = 0.0;
        vertex[1].y = sqrt(3.0);
        vertex[1].z = 0.0;

        vertex[2].x = 1.0;
        vertex[2].y = 0.0;
        vertex[2].z = 0.0;



        if (numPoints != 3) {
            glColor3f(1.0, 1.0, 1.0);
            glBegin(GL_POINTS);
                for (i = 0; i <numPoints; i++) {
                        glVertex3f(vertex[i].x, vertex[i].y, vertex[i].z);
                }
            glEnd();
        }

        if (numPoints == 3) {
            glColor3f(1.0, 1.0, 0.0);
            glBegin(GL_LINE_STRIP);
            koch(vertex[0], vertex[1], 0);
            koch(vertex[1], vertex[2], 0);
            koch(vertex[2], vertex[0], 0);

            glEnd();
        }

        glFlush();
}

/*This function calculates the number of points of Koch snowflake*/

void koch (Point p0, Point p1, int depth) {
        Point a, b, p2;


        if (depth < iter) {
                a.x = (p1.x - p0.x) / 3.0 + p0.x;
                a.y = (p1.y - p0.y) / 3.0 + p0.y;
                a.z = 0.0;

                b.x = p1.x - (p1.x - p0.x) / 3.0;
```

```
        b.y = p1.y - (p1.y - p0.y) / 3.0;
        b.z = 0.0;


        p2.x = p0.x / 2.0 + p1.x / 2.0 + (a.y - b.y)*sqrt(3.0
        p2.y = p0.y / 2.0 + p1.y / 2.0 + (b.x - a.x)*sqrt(3.0
        p2.z = 0.0;



        /*Output: print values*/
        printf ("a.x %f\n", a.x);
        printf ("a.y %f\n", a.y);
        printf ("b.x %f\n", b.x);
        printf ("b.y %f\n", b.y);
        printf ("p2.x %f\n", p2.x);
        printf ("p2.y %f\n", p2.y);

        glVertex3f(p0.x, p0.y, p0.z);
        koch(p0, a, depth + 1);
        glVertex3f(a.x, a.y, a.z);
        koch(a, p2, depth + 1);
        glVertex3f(p2.x, p2.y, p2.z);
        koch(p2, b, depth + 1);
        glVertex3f(b.x, b.y, b.z);
        koch(b, p1, depth + 1);
        glVertex3f(p1.x, p1.y, p1.z);


    }

    return;
}
```

# Appendix B

# Minkowski Island Program

Fractal Antennas: Code for **Minkowski Island** (As a C Program)

```
#include <GL/freeglut.h>
#include <math.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define SCREEN 2.0f

typedef struct {
        float x, y, z;
        }Point;

/* Variables */
float xMax, yMax;

void initialize (void);
void redimension (int width, int height);
void draw (void);
void keyboard (unsigned char key, int cx, int cy);
void Skeyboard (int key, int cy, int cx);
void mouse (int button, int state, int cx, int cy);
void Minkowski1 (Point p0, Point p1, int depth);
void Minkowski2 (Point p0, Point p1, int depth);
/* Variable keeps position of the vertices*/
Point vertex[4];
/* Variable number of Points*/
int numPoints;
```

```
int iter;

int main(int argc, char** argv) {

   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGBA);
   glutInitWindowSize (640, 480);
   glutInitWindowPosition (100, 105);
   glutCreateWindow (argv[0]);
   initialize();

   glutDisplayFunc(draw);
   glutReshapeFunc(redimension);
   glutKeyboardFunc(keyboard);
   glutSpecialFunc(Skeyboard);
   glutMouseFunc(mouse);

   glutMainLoop();

   return 0;
}

void initialize () {
   /*Initialize*/
   glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
   glPointSize(5.0);
   numPoints = 0;
   iter = 1;
}

void redimension (int width, int height) {
   float aspect;

   glViewport (0, 0, width, height);
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity ();

   if (height == 0){
         height = 1;
   }

   aspect = (float) width / (float) height;

   if (width <= height) {
```

```
            glOrtho(-SCREEN, SCREEN, -SCREEN/aspect, SCREEN/aspect, -SCREEN, SC
            xMax = SCREEN;
            yMax = SCREEN / aspect;

    } else {
            glOrtho(-SCREEN * aspect, SCREEN * aspect, -SCREEN, SCREEN, -SCREEN
            xMax = SCREEN * aspect;
            yMax = SCREEN;
    }

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard (unsigned char key, int cx, int cy) {
        /*Keyboard: If ESC Key is pressed, the program finishes*/

        switch (key) {
                case 27:
                    exit(0);
                break;

                default: break;
        }

        glutPostRedisplay();
}

void Skeyboard (int key, int cy, int cx) {
        /*SKeyboard: If PgUp or PgDn are pressed, the iter value is increased

        switch (key) {
                case GLUT_KEY_PAGE_UP:
                    iter++;
                break;

                case GLUT_KEY_PAGE_DOWN:
                    if (iter > 1) iter--;
                break;

                default: break;
        }

        glutPostRedisplay();
```

```
}

void mouse (int button, int state, int cx, int cy) {
        float x, y;


        if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {

                if (numPoints < 4) {
                   vertex[numPoints].x = x;
                   vertex[numPoints].y = y;
                   vertex[numPoints].z = 0.0;
                   ++numPoints;
            }
            else {

                numPoints = 0;
                vertex[numPoints].x = x;
                vertex[numPoints].y = y;
                vertex[numPoints].z = 0.0;
                numPoints++;
            }
        }

        glutPostRedisplay();


}


void draw (void) {
        /* draw the points, we start with a square with coordinates (
        int i;

        glClear(GL_COLOR_BUFFER_BIT);

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

         vertex[0].x = -1.0;
         vertex[0].y = 1.0;
         vertex[0].z = 0.0;

         vertex[1].x = 1.0;
```

```
        vertex[1].y = 1.0;
        vertex[1].z = 0.0;

        vertex[2].x = 1.0;
        vertex[2].y = -1.0;
        vertex[2].z = 0.0;

        vertex[3].x = -1.0;
        vertex[3].y = -1.0;
        vertex[3].z = 0.0;


        if (numPoints < 4) {
           glColor3f(1.0, 1.0, 1.0);
           glBegin(GL_POINTS);
              for (i = 0; i <numPoints; i++) {
                     glVertex3f(vertex[i].x, vertex[i].y, vertex[i].z);
              }
           glEnd();
        }

        if (numPoints >= 4) {
              glColor3f(1.0, 1.0, 0.0);
              glBegin(GL_LINE_STRIP);
              Minkowski1(vertex[0], vertex[1], 0);
              Minkowski2(vertex[1], vertex[2], 0);
              Minkowski1(vertex[2], vertex[3], 0);
              Minkowski2(vertex[3], vertex[0], 0);

           glEnd();

        }


        glFlush();
}

/*These functions calculate the number of points of Minkowski island */

void Minkowski1 (Point p0, Point p1, int depth) {
        /*Horizontal Points*/

        Point a, b, c, d;
```

```
if (depth < iter) {

        a.x = (p1.x - p0.x) / 3.0 + p0.x;
        a.y = p0.y;
        a.z = 0.0;

        c.x = (p1.x - p0.x) / 3.0 + p0.x;
        c.y = p0.y - (p1.x -p0.x) * (2.0 / 9.0);
        c.z = 0.0;

        d.x = p1.x - (p1.x - p0.x) / 3.0;
        d.y = p0.y - (p1.x -p0.x) * (2.0 / 9.0);
        d.z = 0.0;


        b.x = p1.x - (p1.x - p0.x) / 3.0;
        b.y = p1.y;
        b.z = 0.0;

        /*Output: print values*/
        printf ("a.x %f\n", a.x);
        printf ("a.y %f\n", a.y);
        printf ("b.x %f\n", b.x);
        printf ("b.y %f\n", b.y);
        printf ("d.x %f\n", d.x);
        printf ("d.y %f\n", d.y);
        printf ("c.x %f\n", c.x);
        printf ("c.y %f\n", c.y);

        glVertex3f(p0.x, p0.y, p0.z);
        Minkowski1(p0, a, depth + 1);
        glVertex3f(a.x, a.y, a.z);
        Minkowski2(a, c, depth + 1);
        glVertex3f(c.x, c.y, c.z);
        Minkowski1(c, d, depth + 1);
        glVertex3f(d.x, d.y, d.z);
        Minkowski2 (d, b, depth +1);
        glVertex3f(b.x, b.y, b.z);
        Minkowski1(b, p1, depth + 1);
        glVertex3f(p1.x, p1.y, p1.z);



}
```

```
        return;
}
void Minkowski2 (Point p0, Point p1, int depth) {
        /*Vertical Points*/

        Point a, b, c, d;

        if (depth < iter) {


                a.x = p0.x;
                a.y = (p1.y - p0.y) / 3.0 + p0.y;
                a.z = 0.0;

                c.x = (p0.x) + (p1.y - p0.y) * (2.0 / 9.0);
                c.y = (p1.y - p0.y) / 3.0 + p0.y;
                c.z = 0.0;

                d.x = (p0.x) + (p1.y - p0.y) * (2.0 / 9.0);
                d.y = p1.y - (p1.y - p0.y) / 3.0;
                d.z = 0.0;


                b.x = p1.x;
                b.y = p1.y - (p1.y - p0.y) / 3.0;
                b.z = 0.0;

                /*Output: print values*/
                printf ("a.x %f\n", a.x);
                printf ("a.y %f\n", a.y);
                printf ("b.x %f\n", b.x);
                printf ("b.y %f\n", b.y);
                printf ("d.x %f\n", d.x);
                printf ("d.y %f\n", d.y);
                printf ("c.x %f\n", c.x);
                printf ("c.y %f\n", c.y);

                glVertex3f(p0.x, p0.y, p0.z);
                Minkowski2(p0, a, depth + 1);
                glVertex3f(a.x, a.y, a.z);
                Minkowski1(a, c, depth + 1);
                glVertex3f(c.x, c.y, c.z);
                Minkowski2(c, d, depth + 1);
```

```
                    glVertex3f(d.x, d.y, d.z);
                    Minkowski1 (d, b, depth +1);
                    glVertex3f(b.x, b.y, b.z);
                    Minkowski2(b, p1, depth + 1);
                    glVertex3f(p1.x, p1.y, p1.z);

}

          return;
}
```

# Appendix C

# Minkowski Island 2 Iterations

**Gerber File**:Minkowski Island 2 Iterations

*

%LPD*%

%LNLayer2*%

%FSLAX24Y24*%

%MOIN*%

%AD*%

%ADD10C,0.078740*%
%ADD11R,0.012748X0.012748*%
%ADD12R,0.0700X0.2000*%

G1X−037370Y037370D02*

G24D11*

G1X−030621Y037370D01*
G1X−030621Y033237D01*
G1X−021872Y033237D01*
G1X−021872Y037370D01*
G1X−013123Y037370D01*

G1X−013123Y033237D01*
G1X−017011Y033237D01*
G1X−017011Y027704D01*
G1X−013123Y027704D01*
G1X−013123Y021872D01*
G1X−004374Y021872D01*
G1X−004374Y016037D01*
G1X004374Y016037D01*
G1X004374Y021872D01*
G1X013123Y021872D01*
G1X013123Y027704D01*
G1X017011Y027704D01*
G1X017011Y033237D01*
G1X013123Y033237D01*
G1X013123Y037370D01*
G1X021872Y037370D01*
G1X021872Y033237D01*
G1X030621Y033237D01*
G1X030621Y037370D01*
G1X037370Y037370D01*
G1X037370Y030621D01*
G1X033237Y030621D01*
G1X033237Y021872D01*
G1X037370Y021872D01*
G1X037370Y013123D01*
G1X033237Y013123D01*
G1X033237Y017011D01*
G1X027704Y017011D01*
G1X027704Y013123D01*
G1X021872Y013123D01*
G1X021872Y004374D01*
G1X016037Y004374D01*
G1X016037Y−004374D01*
G1X021872Y−004374D01*
G1X021872Y−013123D01*
G1X027704Y−013123D01*
G1X027704Y−017011D01*
G1X033237Y−017011D01*
G1X033237Y−013123D01*
G1X037370Y−013123D01*
G1X037370Y−021872D01*
G1X033237Y−021872D01*
G1X033237Y−030621D01*
G1X037370Y−030621D01*

```
G1X037370Y-037370D01*
G1X030621Y-037370D01*
G1X030621Y-033237D01*
G1X021872Y-033237D01*
G1X021872Y-037370D01*
G1X013123Y-037370D01*
G1X013123Y-033237D01*
G1X017011Y-033237D01*
G1X017011Y-027704D01*
G1X013123Y-027704D01*
G1X013123Y-021872D01*
G1X004374Y-021872D01*
G1X004374Y-016037D01*
G1X001428Y-016037D01*
G1X001428Y-023713D01*

G24D10*
G1X001428Y-017344D03*
G1X001428Y-017771D03*
G1X001428Y-022278D03*
G1X-001428Y-017344D03*
G1X-001428Y-017771D03*
G1X-001428Y-022278D03*

G24D12*
G1X-001740Y020370D03*
G1X000000Y020370D03*
G1X001740Y020370D03*

G24D11*
G1X-001428Y-023713D02*
G1X-001428Y-016037D01*
G1X-004374Y-016037D01*
G1X-004374Y-021872D01*
G1X-013123Y-021872D01*
G1X-013123Y-027704D01*
G1X-017011Y-027704D01*
G1X-017011Y-033237D01*
G1X-013123Y-033237D01*
G1X-013123Y-037370D01*
G1X-021872Y-037370D01*
G1X-021872Y-033237D01*
G1X-030621Y-033237D01*
G1X-030621Y-037370D01*
```

G1X−037370Y−037370D01∗
G1X−037370Y−030621D01∗
G1X−033237Y−030621D01∗
G1X−033237Y−021872D01∗
G1X−037370Y−021872D01∗
G1X−037370Y−013123D01∗
G1X−033237Y−013123D01∗
G1X−033237Y−017011D01∗
G1X−027704Y−017011D01∗
G1X−027704Y−013123D01∗
G1X−021872Y−013123D01∗
G1X−021872Y−004374D01∗
G1X−016037Y−004374D01∗
G1X−016037Y004374D01∗
G1X−021872Y004374D01∗
G1X−021872Y013123D01∗
G1X−027704Y013123D01∗
G1X−027704Y017011D01∗
G1X−033237Y017011D01∗
G1X−033237Y013123D01∗
G1X−037370Y013123D01∗
G1X−037370Y021872D01∗
G1X−033237Y021872D01∗
G1X−033237Y030621D01∗
G1X−037370Y030621D01∗
G1X−037370Y037370D01∗

M02∗

# Appendix D

# Minkowski Island 3 Iterations

**Gerber File**: Minkowski Island 3 Iterations

```
*

%LPD*%

%LNLayer2*%

%FSLAX24Y24*%

%MOIN*%

%AD*%

%ADD10C,0.078740*%
%ADD11R,0.031476X0.031476*%
%ADD12R,0.0700X0.2000*%

G1X-044488Y044488D02*

G24D11*



G1X-041172Y044488D01*
G1X-041172Y042271D01*
G1X-037877Y042271D01*
G1X-037877Y044488D01*
```

G1X-034601Y044488D01*
G1X-034601Y042271D01*
G1X-036066Y042271D01*
G1X-036066Y040074D01*
G1X-034601Y040074D01*
G1X-034601Y037877D01*
G1X-031306Y037877D01*
G1X-031306Y032700D01*
G1X-028011Y032700D01*
G1X-028011Y037877D01*
G1X-024712Y037877D01*
G1X-024712Y040071D01*
G1X-023221Y040071D01*
G1X-023221Y042288D01*
G1X-024712Y042288D01*
G1X-024712Y044488D01*
G1X-021420Y044488D01*
G1X-021420Y042271D01*
G1X-018124Y042271D01*
G1X-018124Y044488D01*
G1X-014827Y044488D01*
G1X-014827Y042271D01*
G1X-016273Y042271D01*
G1X-016273Y040074D01*
G1X-014827Y040074D01*
G1X-014827Y037877D01*
G1X-016273Y037877D01*
G1X-016273Y038873D01*
G1X-017728Y038873D01*
G1X-017728Y037877D01*
G1X-017223Y037877D01*
G1X-017223Y032700D01*
G1X-020687Y032700D01*
G1X-020687Y033203D01*
G1X-017223Y033203D01*
G1X-017223Y031306D01*
G1X-017728Y031306D01*
G1X-017728Y030330D01*
G1X-016274Y030330D01*
G1X-016274Y031306D01*
G1X-014827Y031306D01*
G1X-014827Y027107D01*
G1X-016273Y027107D01*
G1X-016273Y026712D01*

```
G1X−014827Y026712D01∗
G1X−014827Y024712D01∗
G1X−011233Y024712D01∗
G1X−011233Y022218D01∗
G1X−008238Y022218D01∗
G1X−008238Y024712D01∗
G1X−004743Y024712D01∗
G1X−004743Y022218D01∗
G1X−006407Y022218D01∗
G1X−006407Y020322D01∗
G1X−004743Y020322D01∗
G1X−004743Y018124D01∗
G1X−001647Y018124D01∗
G1X−001647Y012727D01∗
G1X001647Y012727D01∗
G1X001647Y018124D01∗
G1X004743Y018124D01∗
G1X004743Y020322D01∗
G1X006407Y020322D01∗
G1X006407Y022218D01∗
G1X004743Y022218D01∗
G1X004743Y024712D01∗
G1X008238Y024712D01∗
G1X008238Y022218D01∗
G1X011233Y022218D01∗
G1X011233Y024712D01∗
G1X014827Y024712D01∗
G1X014827Y026712D01∗
G1X016273Y026712D01∗
G1X016273Y027107D01∗
G1X014827Y027107D01∗
G1X014827Y031306D01∗
G1X016274Y031306D01∗
G1X016274Y030330D01∗
G1X017728Y030330D01∗
G1X017728Y031306D01∗
G1X017223Y031306D01∗
G1X017223Y033203D01∗
G1X020687Y033203D01∗
G1X020687Y032700D01∗
G1X017223Y032700D01∗
G1X017223Y037877D01∗
G1X017728Y037877D01∗
G1X017728Y038873D01∗
```

G1X016273Y038873D01*
G1X016273Y037877D01*
G1X014827Y037877D01*
G1X014827Y040074D01*
G1X016273Y040074D01*
G1X016273Y042271D01*
G1X014827Y042271D01*
G1X014827Y044488D01*
G1X018124Y044488D01*
G1X018124Y042271D01*
G1X021420Y042271D01*
G1X021420Y044488D01*
G1X024712Y044488D01*
G1X024712Y042288D01*
G1X023221Y042288D01*
G1X023221Y040071D01*
G1X024712Y040071D01*
G1X024712Y037877D01*
G1X028011Y037877D01*
G1X028011Y032700D01*
G1X031306Y032700D01*
G1X031306Y037877D01*
G1X034601Y037877D01*
G1X034601Y040074D01*
G1X036066Y040074D01*
G1X036066Y042271D01*
G1X034601Y042271D01*
G1X034601Y044488D01*
G1X037877Y044488D01*
G1X037877Y042271D01*
G1X041172Y042271D01*
G1X041172Y044488D01*
G1X044488Y044488D01*
G1X044488Y041172D01*
G1X042271Y041172D01*
G1X042271Y037877D01*
G1X044488Y037877D01*
G1X044488Y034601D01*
G1X042271Y034601D01*
G1X042271Y036066D01*
G1X040074Y036066D01*
G1X040074Y034601D01*
G1X037877Y034601D01*
G1X037877Y031306D01*

```
G1X032700Y031306D01*
G1X032700Y028011D01*
G1X037877Y028011D01*
G1X037877Y024712D01*
G1X040071Y024712D01*
G1X040071Y023221D01*
G1X042288Y023221D01*
G1X042288Y024712D01*
G1X044488Y024712D01*
G1X044488Y021420D01*
G1X042271Y021420D01*
G1X042271Y018124D01*
G1X044488Y018124D01*
G1X044488Y014827D01*
G1X042271Y014827D01*
G1X042271Y016273D01*
G1X040074Y016273D01*
G1X040074Y014827D01*
G1X037877Y014827D01*
G1X037877Y016273D01*
G1X038873Y016273D01*
G1X038873Y017728D01*
G1X037877Y017728D01*
G1X037877Y017223D01*
G1X032700Y017223D01*
G1X032700Y020687D01*
G1X033203Y020687D01*
G1X033203Y017223D01*
G1X031306Y017223D01*
G1X031306Y017728D01*
G1X030330Y017728D01*
G1X030330Y016274D01*
G1X031306Y016274D01*
G1X031306Y014827D01*
G1X027107Y014827D01*
G1X027107Y016273D01*
G1X026712Y016273D01*
G1X026712Y014827D01*
G1X024712Y014827D01*
G1X024712Y011233D01*
G1X022218Y011233D01*
G1X022218Y008238D01*
G1X024712Y008238D01*
G1X024712Y004743D01*
```

G1X022218Y004743D01*
G1X022218Y006407D01*
G1X020322Y006407D01*
G1X020322Y004743D01*
G1X018124Y004743D01*
G1X018124Y001647D01*
G1X012727Y001647D01*
G1X012727Y−001647D01*
G1X018124Y−001647D01*
G1X018124Y−004743D01*
G1X020322Y−004743D01*
G1X020322Y−006407D01*
G1X022218Y−006407D01*
G1X022218Y−004743D01*
G1X024712Y−004743D01*
G1X024712Y−008238D01*
G1X022218Y−008238D01*
G1X022218Y−011233D01*
G1X024712Y−011233D01*
G1X024712Y−014827D01*
G1X026712Y−014827D01*
G1X026712Y−016273D01*
G1X027107Y−016273D01*
G1X027107Y−014827D01*
G1X031306Y−014827D01*
G1X031306Y−016274D01*
G1X030330Y−016274D01*
G1X030330Y−017728D01*
G1X031306Y−017728D01*
G1X031306Y−017223D01*
G1X033203Y−017223D01*
G1X033203Y−020687D01*
G1X032700Y−020687D01*
G1X032700Y−017223D01*
G1X037877Y−017223D01*
G1X037877Y−017728D01*
G1X038873Y−017728D01*
G1X038873Y−016273D01*
G1X037877Y−016273D01*
G1X037877Y−014827D01*
G1X040074Y−014827D01*
G1X040074Y−016273D01*
G1X042271Y−016273D01*
G1X042271Y−014827D01*

G1X044488Y−014827D01∗
G1X044488Y−018124D01∗
G1X042271Y−018124D01∗
G1X042271Y−021420D01∗
G1X044488Y−021420D01∗
G1X044488Y−024712D01∗
G1X042288Y−024712D01∗
G1X042288Y−023221D01∗
G1X040071Y−023221D01∗
G1X040071Y−024712D01∗
G1X037877Y−024712D01∗
G1X037877Y−028011D01∗
G1X032700Y−028011D01∗
G1X032700Y−031306D01∗
G1X037877Y−031306D01∗
G1X037877Y−034601D01∗
G1X040074Y−034601D01∗
G1X040074Y−036066D01∗
G1X042271Y−036066D01∗
G1X042271Y−034601D01∗
G1X044488Y−034601D01∗
G1X044488Y−037877D01∗
G1X042271Y−037877D01∗
G1X042271Y−041172D01∗
G1X044488Y−041172D01∗
G1X044488Y−044488D01∗
G1X041172Y−044488D01∗
G1X041172Y−042271D01∗
G1X037877Y−042271D01∗
G1X037877Y−044488D01∗
G1X034601Y−044488D01∗
G1X034601Y−042271D01∗
G1X036066Y−042271D01∗
G1X036066Y−040074D01∗
G1X034601Y−040074D01∗
G1X034601Y−037877D01∗
G1X031306Y−037877D01∗
G1X031306Y−032700D01∗
G1X028011Y−032700D01∗
G1X028011Y−037877D01∗
G1X024712Y−037877D01∗
G1X024712Y−040071D01∗
G1X023221Y−040071D01∗
G1X023221Y−042288D01∗

G1X024712Y−042288D01∗
G1X024712Y−044488D01∗
G1X021420Y−044488D01∗
G1X021420Y−042271D01∗
G1X018124Y−042271D01∗
G1X018124Y−044488D01∗
G1X014827Y−044488D01∗
G1X014827Y−042271D01∗
G1X016273Y−042271D01∗
G1X016273Y−040074D01∗
G1X014827Y−040074D01∗
G1X014827Y−037877D01∗
G1X016273Y−037877D01∗
G1X016273Y−038873D01∗
G1X017728Y−038873D01∗
G1X017728Y−037877D01∗
G1X017223Y−037877D01∗
G1X017223Y−032700D01∗
G1X020687Y−032700D01∗
G1X020687Y−033203D01∗
G1X017223Y−033203D01∗
G1X017223Y−031306D01∗
G1X017728Y−031306D01∗
G1X017728Y−030330D01∗
G1X016274Y−030330D01∗
G1X016274Y−031306D01∗
G1X014827Y−031306D01∗
G1X014827Y−027107D01∗
G1X016273Y−027107D01∗
G1X016273Y−026712D01∗
G1X014827Y−026712D01∗
G1X014827Y−024712D01∗
G1X011233Y−024712D01∗
G1X011233Y−022218D01∗
G1X008238Y−022218D01∗
G1X008238Y−024712D01∗
G1X004743Y−024712D01∗
G1X004743Y−022218D01∗
G1X006407Y−022218D01∗
G1X006407Y−020322D01∗
G1X004743Y−020322D01∗
G1X004743Y−018124D01∗
G1X001647Y−018124D01∗
G1X001647Y−012727D01∗

```
G1X000247Y−012727D01∗
G1X000247Y−020000D01∗
G1X001647Y−020000D01∗
G1X001647Y−027000D01∗

G24D10∗
G1X001647Y−021166D03∗
G1X001647Y−023477D03∗
G1X001647Y−022832D03∗
G1X−001647Y−021166D03∗
G1X−001647Y−023477D03∗
G1X−001647Y−022832D03∗

G24D12∗
G1X−001740Y022488D03∗
G1X000000Y022488D03∗
G1X001740Y022488D03∗


G24D11∗
G1X−001647Y−027000D02∗
G1X−001647Y−020000D01∗
G1X−000247Y−020000D01∗
G1X−000247Y−012727D01∗
G1X−001647Y−012727D01∗
G1X−001647Y−018124D01∗
G1X−004743Y−018124D01∗
G1X−004743Y−020322D01∗
G1X−006407Y−020322D01∗
G1X−006407Y−022218D01∗
G1X−004743Y−022218D01∗
G1X−004743Y−024712D01∗
G1X−008238Y−024712D01∗
G1X−008238Y−022218D01∗
G1X−011233Y−022218D01∗
G1X−011233Y−024712D01∗
G1X−014827Y−024712D01∗
G1X−014827Y−026712D01∗
G1X−016273Y−026712D01∗
G1X−016273Y−027107D01∗
G1X−014827Y−027107D01∗
G1X−014827Y−031306D01∗
G1X−016274Y−031306D01∗
G1X−016274Y−030330D01∗
```

G1X−017728Y−030330D01*
G1X−017728Y−031306D01*
G1X−017223Y−031306D01*
G1X−017223Y−033203D01*
G1X−020687Y−033203D01*
G1X−020687Y−032700D01*
G1X−017223Y−032700D01*
G1X−017223Y−037877D01*
G1X−017728Y−037877D01*
G1X−017728Y−038873D01*
G1X−016273Y−038873D01*
G1X−016273Y−037877D01*
G1X−014827Y−037877D01*
G1X−014827Y−040074D01*
G1X−016273Y−040074D01*
G1X−016273Y−042271D01*
G1X−014827Y−042271D01*
G1X−014827Y−044488D01*
G1X−018124Y−044488D01*
G1X−018124Y−042271D01*
G1X−021420Y−042271D01*
G1X−021420Y−044488D01*
G1X−024712Y−044488D01*
G1X−024712Y−042288D01*
G1X−023221Y−042288D01*
G1X−023221Y−040071D01*
G1X−024712Y−040071D01*
G1X−024712Y−037877D01*
G1X−028011Y−037877D01*
G1X−028011Y−032700D01*
G1X−031306Y−032700D01*
G1X−031306Y−037877D01*
G1X−034601Y−037877D01*
G1X−034601Y−040074D01*
G1X−036066Y−040074D01*
G1X−036066Y−042271D01*
G1X−034601Y−042271D01*
G1X−034601Y−044488D01*
G1X−037877Y−044488D01*
G1X−037877Y−042271D01*
G1X−041172Y−042271D01*
G1X−041172Y−044488D01*
G1X−044488Y−044488D01*
G1X−044488Y−041172D01*

G1X−042271Y−041172D01∗
G1X−042271Y−037877D01∗
G1X−044488Y−037877D01∗
G1X−044488Y−034601D01∗
G1X−042271Y−034601D01∗
G1X−042271Y−036066D01∗
G1X−040074Y−036066D01∗
G1X−040074Y−034601D01∗
G1X−037877Y−034601D01∗
G1X−037877Y−031306D01∗
G1X−032700Y−031306D01∗
G1X−032700Y−028011D01∗
G1X−037877Y−028011D01∗
G1X−037877Y−024712D01∗
G1X−040071Y−024712D01∗
G1X−040071Y−023221D01∗
G1X−042288Y−023221D01∗
G1X−042288Y−024712D01∗
G1X−044488Y−024712D01∗
G1X−044488Y−021420D01∗
G1X−042271Y−021420D01∗
G1X−042271Y−018124D01∗
G1X−044488Y−018124D01∗
G1X−044488Y−014827D01∗
G1X−042271Y−014827D01∗
G1X−042271Y−016273D01∗
G1X−040074Y−016273D01∗
G1X−040074Y−014827D01∗
G1X−037877Y−014827D01∗
G1X−037877Y−016273D01∗
G1X−038873Y−016273D01∗
G1X−038873Y−017728D01∗
G1X−037877Y−017728D01∗
G1X−037877Y−017223D01∗
G1X−032700Y−017223D01∗
G1X−032700Y−020687D01∗
G1X−033203Y−020687D01∗
G1X−033203Y−017223D01∗
G1X−031306Y−017223D01∗
G1X−031306Y−017728D01∗
G1X−030330Y−017728D01∗
G1X−030330Y−016274D01∗
G1X−031306Y−016274D01∗
G1X−031306Y−014827D01∗

G1X−027107Y−014827D01∗
G1X−027107Y−016273D01∗
G1X−026712Y−016273D01∗
G1X−026712Y−014827D01∗
G1X−024712Y−014827D01∗
G1X−024712Y−011233D01∗
G1X−022218Y−011233D01∗
G1X−022218Y−008238D01∗
G1X−024712Y−008238D01∗
G1X−024712Y−004743D01∗
G1X−022218Y−004743D01∗
G1X−022218Y−006407D01∗
G1X−020322Y−006407D01∗
G1X−020322Y−004743D01∗
G1X−018124Y−004743D01∗
G1X−018124Y−001647D01∗
G1X−012727Y−001647D01∗
G1X−012727Y001647D01∗
G1X−018124Y001647D01∗
G1X−018124Y004743D01∗
G1X−020322Y004743D01∗
G1X−020322Y006407D01∗
G1X−022218Y006407D01∗
G1X−022218Y004743D01∗
G1X−024712Y004743D01∗
G1X−024712Y008238D01∗
G1X−022218Y008238D01∗
G1X−022218Y011233D01∗
G1X−024712Y011233D01∗
G1X−024712Y014827D01∗
G1X−026712Y014827D01∗
G1X−026712Y016273D01∗
G1X−027107Y016273D01∗
G1X−027107Y014827D01∗
G1X−031306Y014827D01∗
G1X−031306Y016274D01∗
G1X−030330Y016274D01∗
G1X−030330Y017728D01∗
G1X−031306Y017728D01∗
G1X−031306Y017223D01∗
G1X−033203Y017223D01∗
G1X−033203Y020687D01∗
G1X−032700Y020687D01∗
G1X−032700Y017223D01∗

```
G1X-037877Y017223D01*
G1X-037877Y017728D01*
G1X-038873Y017728D01*
G1X-038873Y016273D01*
G1X-037877Y016273D01*
G1X-037877Y014827D01*
G1X-040074Y014827D01*
G1X-040074Y016273D01*
G1X-042271Y016273D01*
G1X-042271Y014827D01*
G1X-044488Y014827D01*
G1X-044488Y018124D01*
G1X-042271Y018124D01*
G1X-042271Y021420D01*
G1X-044488Y021420D01*
G1X-044488Y024712D01*
G1X-042288Y024712D01*
G1X-042288Y023221D01*
G1X-040071Y023221D01*
G1X-040071Y024712D01*
G1X-037877Y024712D01*
G1X-037877Y028011D01*
G1X-032700Y028011D01*
G1X-032700Y031306D01*
G1X-037877Y031306D01*
G1X-037877Y034601D01*
G1X-040074Y034601D01*
G1X-040074Y036066D01*
G1X-042271Y036066D01*
G1X-042271Y034601D01*
G1X-044488Y034601D01*
G1X-044488Y037877D01*
G1X-042271Y037877D01*
G1X-042271Y041172D01*
G1X-044488Y041172D01*
G1X-044488Y044488D01*

M02*
```

TRITA-ICT-EX-2015:45